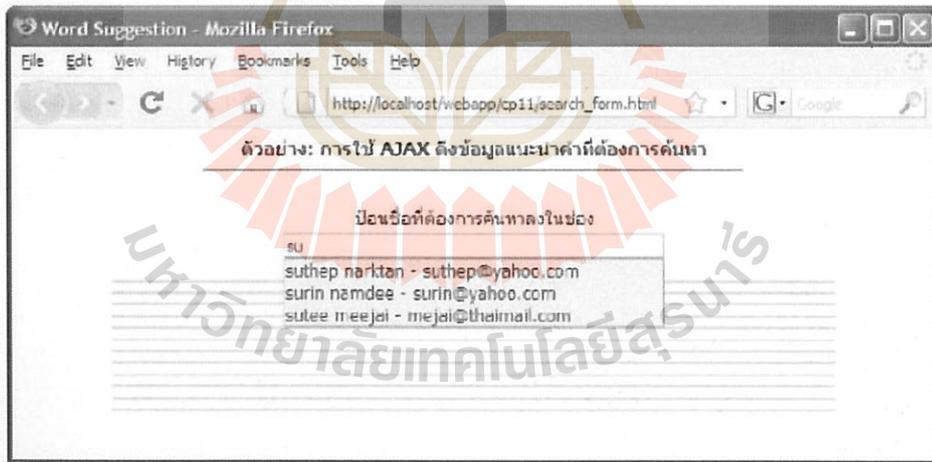




โปรแกรมประยุกต์บนเว็บสำหรับผู้เริ่มต้น (Web Application for Beginner)



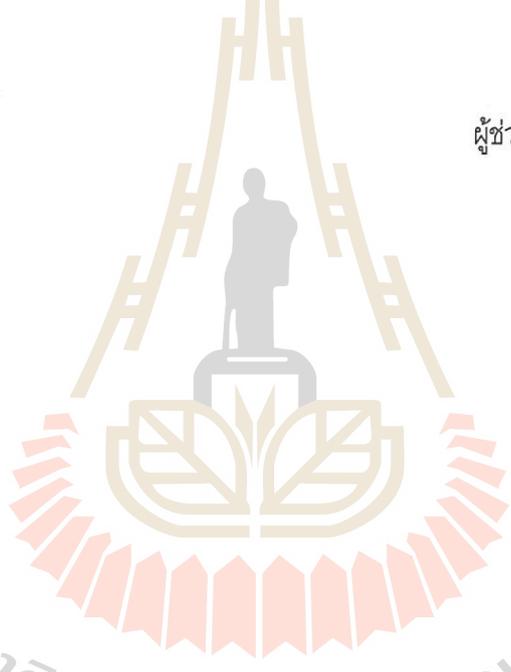
ผู้ช่วยศาสตราจารย์ ดร. ศษา ชาณศิลป์
สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีสุรนารี



กิตติกรรมประกาศ

หนังสือเล่มนี้ถูกเขียนขึ้นมาจนสำเร็จลุล่วงไปได้ด้วยดี ก็เพราะการสนับสนุนของนักศึกษา สาขาวิชา วิศวกรรมคอมพิวเตอร์ รุ่นที่ 9, 10 และ 11 มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้ทดลองใช้เป็นเอกสาร คำสอน ประกอบการเรียนในรายวิชา 423401 Web Application ในภาคการศึกษาที่ 1/2550, 1/2551 และ 1/2552 ตามลำดับ โดยช่วยตรวจสอบการทำงานของตัวอย่างการใช้คำสั่งต่าง ๆ แก้ไข คำผิด และเสนอแนะการเพิ่มเติมข้อมูลบางส่วน เพื่อให้หนังสือเล่มนี้สมบูรณ์และครอบคลุมการใช้งานมากยิ่งขึ้น และผู้ช่วยศาสตราจารย์สมพันธ์ ชาญศิลป์ หัวหน้าหน่วยวิจัยและพัฒนาไอเพนซอร์ส มหาวิทยาลัย เทคโนโลยีสุรนารี ที่ช่วยตรวจสอบการทำงานของคำสั่งต่าง ๆ เสนอแนะเนื้อหา ตัวอย่าง และการจัดทำ แผ่นวีดิทัศน์บรรจุซอร์สโค้ดทั้งหมดและเซิร์ฟเวอร์ SUTinsServer 5207 พร้อมใช้ นอกจากนี้ยังมีผู้ที่มีส่วน เกี่ยวข้องที่ไม่ได้เอ่ยนาม ผู้เขียนจึงใคร่ขอขอบคุณทุกท่านมา ณ โอกาสนี้

ผู้ช่วยศาสตราจารย์ ดร. คະชา ชาญศิลป์



มหาวิทยาลัยเทคโนโลยีสุรนารี

คำนำ

ข้อมูลข่าวสารต่าง ๆ ที่สามารถส่งถึงผู้รับได้ด้วยความรวดเร็วและแม่นยำ ก็คงหนีไม่พ้นการจัดส่งโดยผ่านทางอินเทอร์เน็ต รวมไปถึงการสั่งซื้อสินค้า การสั่งจองตัวเครื่องบินและที่พัก การจัดการทางการเงินการธนาคาร เป็นต้น ซึ่งต่างก็ต้องมีโปรแกรมเมอร์ที่มีความรู้ความชำนาญในการเขียนโปรแกรมเพื่อจัดการกับสิ่งที่ต้องการ เช่น สร้างแบบฟอร์มเพื่อให้ลูกค้าป้อนข้อมูลที่ต้องการจัดเก็บ จัดเก็บข้อมูลต่าง ๆ ติดต่อบริบบนฐานข้อมูลเพื่อประมวลผล แสดงผลที่ได้ให้ลูกค้าดู เป็นต้น โดยผ่านทางหน้าเว็บ

หนังสือเล่มนี้ จึงได้ออกแบบเนื้อหาเพื่อสนองความต้องการของนักโปรแกรมเมอร์ที่ต้องการศึกษาการเขียนโปรแกรมที่ทำงานบนเว็บ (Web Application) ตั้งแต่เริ่มต้นในเรื่องของ HTML, การกำหนดรูปแบบการแสดงผลโดยใช้ CSS (Cascading Style Sheet), สคริปต์ PHP ที่ใช้ในการติดต่อกับเครื่องเซิร์ฟเวอร์, การใช้คำสั่ง SQL ในการจัดการกับฐานข้อมูล MySQL, การใช้ตัวแปรแบบคูกี้และเซสชัน และการใช้เทคโนโลยีเอแจ็กมาเป็นส่วนเสริมในการเขียนโปรแกรม โดยท้ายบทของแต่ละเนื้อหานี้ยังมีแบบฝึกหัดเพื่อให้ทดลองปฏิบัติ อีกทั้งยังได้รวมคู่มือการติดตั้ง AppServ, คู่มือการติดตั้ง SUTinsServer 5207 และคู่มือการใช้งาน phpMyAdmin ไว้ในภาคผนวก ซึ่งมาพร้อมกับแผ่นดีวีดีที่บรรจุระบบแม่ข่าย (Server) ที่ทำงานบนระบบปฏิบัติการลินุกซ์และซอร์สโค้ดของตัวอย่างทั้งหมดในหนังสือเล่มนี้ จึงเหมาะสำหรับใช้ประกอบการเรียนการสอนในรายวิชาการเขียนโปรแกรมประยุกต์บนเว็บ ในหลักสูตรระดับบัณฑิตศึกษาระดับปริญญาโท สาขาวิชาเทคโนโลยีสารสนเทศ สาขาวิชาวิทยาการคอมพิวเตอร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ และสาขาวิชาอื่น ๆ ที่เกี่ยวข้อง

ผู้เขียนจึงหวังว่า หนังสือเล่มนี้คงมีประโยชน์สำหรับนักเรียน นักศึกษา และผู้ที่สนใจในการเขียนโปรแกรมที่ทำงานบนเว็บไม่มากนักน้อย ถ้ามีข้อเสนอแนะประการใด ผู้เขียนขอน้อมรับด้วยความยินดี และถ้าพบข้อผิดพลาดประการใด ผู้เขียนขออภัยมา ณ ที่นี้ด้วย

ผู้ช่วยศาสตราจารย์ ดร. คະชา ชาญศิลป์

สารบัญ

กิตติกรรมประกาศ.....	ก
คำนำ.....	ค
สารบัญ.....	จ
สารบัญตาราง.....	ฉ
สารบัญรูปภาพ.....	ช
บทที่ 1 แนะนำภาษา HTML <Introduction to HTML>.....	1
บทนำ.....	1
วัตถุประสงค์.....	1
เนื้อหา.....	1
1.1 ประวัติและความเป็นมาของภาษา HTML.....	1
1.2 แท็กต่าง ๆ ในภาษา HTML.....	2
1.3 รูปแบบโครงสร้างของภาษา HTML.....	3
1.3.1 แท็กเดี่ยว.....	4
1.3.2 แท็กคู่.....	5
1.3.3 แท็กทั่วไป.....	5
1.3.4 แท็กตาราง.....	8
1.3.5 แท็กแบบฟอร์ม.....	11
1.3.6 แท็กเฟรม.....	16
1.3.7 แท็กไอเฟรม.....	19
สรุป.....	22
แบบฝึกหัด.....	23
บทที่ 2 แนะนำ CSS <Introduction to Cascading Style Sheet>.....	31
บทนำ.....	31
วัตถุประสงค์.....	31
เนื้อหา.....	31
2.1 CSS คืออะไร?.....	31
2.2 การกำหนดรูปแบบของ CSS.....	33
2.2.1 แบบในบรรทัด (Inline Styles).....	33

2.2.2	แบบภายใน (Internal Style Sheet).....	34
2.2.3	แบบภายนอก (External Style Sheet).....	34
2.3	ตัวเลือกแบบคลาส (Class Selector).....	35
2.4	ตัวเลือกแบบไอดี (ID Selector).....	38
2.5	รหัสเทียมแบบคลาส (Pseudo-classes) และรหัสเทียมแบบอีลีเมนต์ (Pseudo-elements)	39
2.6	คุณสมบัติต่าง ๆ ใน CSS.....	42
	สรุป.....	44
	แบบฝึกหัด.....	45
บทที่ 3 แนะนำภาษา PHP <Introduction to PHP>.....		47
	บทนำ.....	47
	วัตถุประสงค์.....	47
	เนื้อหา.....	47
3.1	ประวัติและความเป็นมาของภาษา PHP.....	47
3.2	รูปแบบโครงสร้างและหลักการทำงานของภาษา PHP.....	48
3.3	ชนิดข้อมูล (Data Types).....	49
3.4	ตัวแปร (Variable).....	53
3.4.1	ตัวแปรระบบ (Server Variables).....	55
3.4.2	ตัวแปรแวดล้อม (Environment Variables).....	56
3.4.3	ตัวแปรคุกกี้ (HTTP Cookies).....	56
3.4.4	ตัวแปรแบบ GET (HTTP GET Variables).....	56
3.4.5	ตัวแปรแบบ POST (HTTP POST Variables).....	56
3.4.6	ตัวแปรแบบ HTTP File Upload Variables.....	56
3.4.7	ตัวแปรแบบ Request (Request Variables).....	56
3.4.8	ตัวแปรแบบเซสชัน (Session Variables).....	56
3.4.9	ตัวแปรแบบ Global (Global Variables).....	57
3.4.10	ตัวแปรแบบ The previous error message.....	57
3.4.11	ขอบเขตของตัวแปร (Variable Scope).....	57
3.5	ค่าคงที่ (Constant).....	58
3.6	คำสั่งเบื้องต้นของภาษา PHP.....	58
3.7	แบบฟอร์มในภาษา HTML.....	60
	สรุป.....	65
	แบบฝึกหัด.....	66
บทที่ 4 ตัวดำเนินการและการควบคุม <Operators and Controls>.....		67
	บทนำ.....	67

วัตถุประสงค์	67
เนื้อหา	67
4.1 ตัวดำเนินการ (Operators).....	67
4.1.1 ตัวดำเนินการคณิตศาสตร์ (Arithmetic Operators).....	68
4.1.2 ตัวดำเนินการกำหนดค่า (Assignment Operators).....	68
4.1.3 ตัวดำเนินการสตริง (String Operators).....	68
4.1.4 ตัวดำเนินการเพิ่มค่าและลดค่า (Increment and Decrement Operators).....	69
4.1.5 ตัวดำเนินการเปรียบเทียบ (Comparison Operators).....	70
4.1.6 ตัวดำเนินการเปรียบเทียบบิต (Bitwise Operators).....	70
4.1.7 ตัวดำเนินการตรรกศาสตร์ (Logical Operators).....	75
4.1.8 ตัวดำเนินการควบคุมข้อผิดพลาด (Error Control Operators).....	75
4.1.9 ตัวดำเนินการจัดการคำสั่ง (Execution Operators).....	76
4.1.10 ตัวดำเนินการอะเรย์ (Array Operators).....	76
4.1.11 ตัวดำเนินการชนิด (Type Operators).....	77
4.1.12 ตัวดำเนินการแบบเงื่อนไข (Conditional Operators).....	77
4.1.13 ลำดับความสำคัญของตัวดำเนินการ (Operator Precedence).....	78
4.2 การควบคุม (Controls).....	79
4.2.1 การควบคุมแบบทางเลือก (Branching).....	80
4.2.2 การควบคุมแบบวนลูป (Looping).....	84
สรุป	89
แบบฝึกหัด	90
บทที่ 5 ฟังก์ชัน <Functions>.....	95
บทนำ.....	95
วัตถุประสงค์	95
เนื้อหา.....	95
5.1 ฟังก์ชันที่มีอยู่ในตัวโปรแกรม (Internal or Built-in Functions).....	95
5.1.1 ฟังก์ชันที่เกี่ยวกับวันและเวลา (Date and Time Functions).....	96
5.1.2 ฟังก์ชันที่เกี่ยวกับอะเรย์ (Array Functions).....	97
5.1.3 ฟังก์ชันที่เกี่ยวกับชนิดตัวอักษร (Character Type Functions).....	103
5.1.4 ฟังก์ชันเกี่ยวกับไดเรคทอรี (Directory Functions).....	104
5.1.5 ฟังก์ชันเกี่ยวกับสตริง (String Functions).....	105
5.1.6 ฟังก์ชันเกี่ยวกับคณิตศาสตร์ (Mathematical Functions).....	109
5.1.7 ฟังก์ชันเกี่ยวกับเมล (Mail Functions).....	111
5.1.8 ฟังก์ชันเกี่ยวกับระบบไฟล์ข้อมูล (Filesystem Functions).....	113
5.2 ฟังก์ชันที่สร้างขึ้นเอง (User-defined Functions).....	117
5.2.1 ฟังก์ชันอาร์กิวเมนต์ (Function Arguments).....	117
5.2.2 ตัวแปรฟังก์ชัน (Variable Functions).....	120

สรุป.....	121
แบบฝึกหัด.....	122
บทที่ 6 เพิ่มข้อมูล <Data Files>	125
บทนำ.....	125
วัตถุประสงค์.....	125
เนื้อหา.....	125
6.1 ฟังก์ชันเกี่ยวกับการอ่านและเขียนเพิ่มข้อมูล.....	126
6.1.1 การเปิดและปิดเพิ่มข้อมูล.....	126
6.1.2 การเขียนข้อมูลลงเพิ่มข้อมูล.....	129
6.1.3 การอ่านข้อมูลจากเพิ่มข้อมูล.....	133
6.1.4 ฟังก์ชันที่เื้ออำนวยการจัดการเพิ่มข้อมูล.....	143
6.1.5 ปัญหาในการจัดการกับเพิ่มข้อมูล.....	145
สรุป.....	146
แบบฝึกหัด.....	147
บทที่ 7 การรวมเพิ่มข้อมูลและฟังก์ชันวาดรูป <Including Files and Drawing Functions>	151
บทนำ.....	151
วัตถุประสงค์.....	151
เนื้อหา.....	151
7.1 คำสั่ง require() และ include() ในการรวมเพิ่มข้อมูล.....	152
7.2 ฟังก์ชันเกี่ยวกับการวาดรูป.....	155
สรุป.....	173
แบบฝึกหัด.....	174
บทที่ 8 แนะนำ MySQL <Introduction to MySQL>.....	177
บทนำ.....	177
วัตถุประสงค์.....	177
เนื้อหา.....	177
8.1 แนวคิดเกี่ยวกับฐานข้อมูลเชิงสัมพันธ์.....	178
8.2 การติดต่อระบบฐานข้อมูล MySQL ผ่าน Command Prompt.....	180
8.3 ภาษาสอบถามข้อมูลเชิงโครงสร้าง.....	182
8.3.1 ภาษากำหนดนิยามข้อมูล (Data Definition Languages: DDL).....	182
8.3.2 ภาษาปรับแต่งข้อมูล (Data Manipulation Languages: DML).....	189
8.4 ชนิดข้อมูลของคอลัมน์.....	199

8.4.1	ชนิดตัวเลข (Numeric Type).....	200
8.4.2	ชนิดวันและเวลา (Date and Time Type).....	200
8.4.3	ชนิดข้อความ (String Type).....	201
8.4.4	ตัวดำเนินการในการเปรียบเทียบสำหรับประโยค WHERE	202
8.4.5	การขอความช่วยเหลือจาก MySQL.....	202
	สรุป	203
	แบบฝึกหัด	204
บทที่ 9 การใช้ PHP และ MySQL <Using PHP & MySQL>		207
	บทนำ.....	207
	วัตถุประสงค์	207
	เนื้อหา.....	207
9.1	ฟังก์ชันเกี่ยวกับฐานข้อมูล MySQL.....	207
9.2	ตัวอย่างการใช้งานฐานข้อมูล MySQL.....	209
	สรุป	226
	แบบฝึกหัด	227
บทที่ 10 คุกกี้และเซสชัน <Cookie and Session>.....		229
	บทนำ.....	229
	วัตถุประสงค์	229
	เนื้อหา.....	229
10.1	คุกกี้คืออะไร.....	229
10.2	การลบคุกกี้.....	235
10.3	เซสชันคืออะไร.....	238
	สรุป	257
	แบบฝึกหัด	258
บทที่ 11 แนะนำเอแจ็ก <Introduction to AJAX>		261
	บทนำ.....	261
	วัตถุประสงค์	261
	เนื้อหา.....	261
11.1	เอแจ็กคืออะไร.....	261
11.2	อ็อบเจกต์ XMLHttpRequest คืออะไร.....	264
11.2.1	เมทอดของ XMLHttpRequest (XMLHttpRequest Methods).....	264
11.2.2	คุณสมบัติของ XMLHttpRequest (XMLHttpRequest Properties).....	264

11.3 การเริ่มต้นใช้งาน XMLHttpRequest	267
11.4 การใช้งาน XMLHttpRequest กับ XML.....	277
11.5 การใช้งาน XMLHttpRequest กับฐานข้อมูล.....	283
สรุป.....	285
แบบฝึกหัด.....	286
ภาคผนวก ก การติดตั้ง AppServ <AppServ Installation>	289
ภาคผนวก ข การติดตั้ง SUTinsServer <SUTinsServer Installation>	297
ภาคผนวก ค การใช้งานโปรแกรม phpMyAdmin <phpMyAdmin's Manual>	307
บรรณานุกรม <Bibliography>.....	325
อภิธานศัพท์ <Glossary>	327
ดัชนี <Index>	329
ประวัติผู้เขียน <Curriculum Vitae>.....	333

สารบัญตาราง

ตารางที่ 1.1:	ตัวอย่างแท็กเดี่ยวและการใช้งาน.....	4
ตารางที่ 1.2:	ตัวอย่างแท็กคู่และการใช้งาน.....	5
ตารางที่ 1.3:	ตัวอย่างแท็กที่ใช้ในการกำหนดตาราง แถว และคอลัมน์.....	8
ตารางที่ 1.4:	ตัวอย่างแท็กที่ใช้ในการสร้างแบบฟอร์ม.....	11
ตารางที่ 1.5:	ตัวอย่างแท็กที่ใช้ในการสร้างเฟรม.....	19
ตารางที่ 2.1:	คุณสมบัติของตัวอักษร.....	42
ตารางที่ 2.2:	คุณสมบัติของสีและพื้นหลัง.....	42
ตารางที่ 2.3:	คุณสมบัติของข้อความ.....	42
ตารางที่ 2.4:	คุณสมบัติของกล่องข้อความ.....	43
ตารางที่ 2.5:	คุณสมบัติของการแบ่งแยก.....	43
ตารางที่ 2.6:	คุณสมบัติของหน่วยวัด.....	43
ตารางที่ 3.1:	อักขระหลีก (Escaped Characters).....	50
ตารางที่ 3.2:	ตัวแปรระบบ.....	55
ตารางที่ 4.1:	ตัวดำเนินการทางคณิตศาสตร์.....	68
ตารางที่ 4.2:	ตัวดำเนินการกำหนดค่า.....	68
ตารางที่ 4.3:	ตัวดำเนินการเพิ่มค่าและลดค่า.....	69
ตารางที่ 4.4:	ตัวดำเนินการเปรียบเทียบ.....	70
ตารางที่ 4.5:	การแทนค่าในรูปแบบของเลขฐานสอง.....	71
ตารางที่ 4.6:	ตัวดำเนินการเปรียบเทียบบิต.....	71
ตารางที่ 4.7:	การเปรียบเทียบบิต $a \& b$	72
ตารางที่ 4.8:	การเปรียบเทียบบิต $a b$	72
ตารางที่ 4.9:	การเปรียบเทียบบิต $a \wedge b$	72
ตารางที่ 4.10:	การเปรียบเทียบบิต $\sim a \& b$	72
ตารางที่ 4.11:	การเปรียบเทียบบิต $a \& \sim b$	73
ตารางที่ 4.12:	การเปรียบเทียบบิต $\sim a \& \sim b$	73
ตารางที่ 4.13:	การเปรียบเทียบบิต $\sim a$	73
ตารางที่ 4.14:	การเปรียบเทียบบิต $\sim b$	74
ตารางที่ 4.15:	การเปรียบเทียบบิต $\sim c$	74
ตารางที่ 4.16:	การเปรียบเทียบบิต $a \ll 2$	75
ตารางที่ 4.17:	การเปรียบเทียบบิต $a \gg 2$	75
ตารางที่ 4.18:	การใช้ตัวดำเนินการในการกำหนดค่าร่วมกับตัวดำเนินการทางคณิตศาสตร์.....	75
ตารางที่ 4.19:	ตัวดำเนินการอะเรย์.....	77
ตารางที่ 4.20:	ลำดับความสำคัญและวิธีการทำงานของตัวดำเนินการ.....	78
ตารางที่ 5.1:	ฟังก์ชันที่เกี่ยวกับวันและเวลา.....	96
ตารางที่ 5.2:	ฟังก์ชันที่เกี่ยวกับอะเรย์.....	97
ตารางที่ 5.3:	ฟังก์ชันที่เกี่ยวกับชนิดตัวอักษร.....	103
ตารางที่ 5.4:	ฟังก์ชันที่เกี่ยวกับไดเรคทอรี.....	104
ตารางที่ 5.5:	ฟังก์ชันที่เกี่ยวกับสตริง.....	105

ตารางที่ 5.6:	ฟังก์ชันที่เกี่ยวกับคณิตศาสตร์.....	109
ตารางที่ 5.7:	ฟังก์ชันที่เกี่ยวกับเมล.....	111
ตารางที่ 5.8:	ฟังก์ชันที่เกี่ยวกับระบบไฟล์ข้อมูล.....	113
ตารางที่ 6.1:	โหมดเกี่ยวกับการอ่านและเขียนเพิ่มข้อมูล.....	126
ตารางที่ 6.2:	การล๊อคเพิ่มข้อมูล	130
ตารางที่ 7.1:	ฟังก์ชันต่าง ๆ ในการวาดรูปพร้อมกับรูปแบบและความหมาย.....	155
ตารางที่ 8.1:	สิทธิ์ต่าง ๆ สำหรับผู้ใช้งาน (User).....	187
ตารางที่ 8.2:	สิทธิ์ต่าง ๆ สำหรับผู้ดูแลระบบ (Administrator).....	188
ตารางที่ 8.3:	สิทธิ์พิเศษ.....	188
ตารางที่ 8.4:	ชนิดข้อมูลแบบเลขจำนวนเต็ม	200
ตารางที่ 8.5:	ชนิดข้อมูลแบบเลขจำนวนจริง.....	200
ตารางที่ 8.6:	ชนิดข้อมูลแบบวันเวลา	200
ตารางที่ 8.7:	การแสดงผลของ TIMESTAMP.....	200
ตารางที่ 8.8:	ชนิดข้อความแบบ CHAR และ VARCHAR.....	201
ตารางที่ 8.9:	ชนิดข้อความแบบ TEXT และ BLOB.....	201
ตารางที่ 8.10:	ชนิดข้อความแบบ SET และ ENUM.....	201
ตารางที่ 8.11:	ตัวดำเนินการในการเปรียบเทียบสำหรับอนุประโยค WHERE.....	202
ตารางที่ 9.1:	ฟังก์ชันต่าง ๆ เกี่ยวกับฐานข้อมูล MySQL.....	208
ตารางที่ 10.1:	พารามิเตอร์ต่าง ๆ ของฟังก์ชัน setcookie().....	230
ตารางที่ 10.2:	ฟังก์ชันเกี่ยวกับเซสชัน	238
ตารางที่ 10.3:	ไฟล์ข้อมูลตรวจสอบการทำงานแบบเซสชัน.....	239
ตารางที่ 11.1:	เมตอดสำหรับอ็อบเจกต์ XMLHttpRequest.....	264

สารบัญรูปภาพ

รูปที่ 1.1:	แท็กที่สำคัญเบื้องต้นใน HTML	3
รูปที่ 1.2:	การใช้แท็กเพื่อการแสดงผลในตำแหน่งต่าง ๆ ของเว็บเบราว์เซอร์	3
รูปที่ 1.3:	การแสดงผลของไฟล์ HTML ในรูปที่ 1.2	4
รูปที่ 1.4:	การใช้แท็ก <hr> และ <p>	4
รูปที่ 1.5:	การแสดงผลของไฟล์ HTML ในรูปที่ 1.4	5
รูปที่ 1.6:	การใช้แท็กเดี่ยวและแท็กคู่ในการแสดงผลในรูปแบบต่าง ๆ	6
รูปที่ 1.7:	การแสดงผลของไฟล์ HTML ในรูปที่ 1.6	6
รูปที่ 1.8:	การแสดงผลของรายการย่อยในรูปแบบต่าง ๆ	7
รูปที่ 1.9:	การแสดงผลของไฟล์ html ในรูปที่ 1.8	8
รูปที่ 1.10:	การใช้แท็กในการสร้างตาราง	9
รูปที่ 1.11:	การแสดงผลของไฟล์ HTML ในรูปที่ 1.10	10
รูปที่ 1.12:	การใช้แท็กเพื่อสร้างปุ่ม submit และ reset	11
รูปที่ 1.13:	การแสดงผลส่วนหนึ่งของไฟล์ HTML ในรูปที่ 1.12	11
รูปที่ 1.14:	การใช้แท็กในการสร้างฟอร์มในรูปแบบต่าง ๆ	12
รูปที่ 1.15:	การแสดงผลของไฟล์ HTML ในรูปที่ 1.14	12
รูปที่ 1.16:	แบบฟอร์มการส่งข้อมูลแบบ post	15
รูปที่ 1.17:	การซ่อนข้อมูลในช่อง URL เมื่อส่งข้อมูลแบบ post	15
รูปที่ 1.18:	แบบฟอร์มการส่งข้อมูลแบบ get	16
รูปที่ 1.19:	การแสดงผลข้อมูลในช่อง URL เมื่อส่งข้อมูลแบบ get	16
รูปที่ 1.20:	การใช้แท็กในการสร้างเฟรม	16
รูปที่ 1.21:	การใช้แท็กในไฟล์ link.html	17
รูปที่ 1.22:	การแสดงผลของไฟล์ link.html	17
รูปที่ 1.23:	การแสดงผลของจุดเชื่อมต่อแบบไม่มีเส้นใต้	17
รูปที่ 1.24:	การใช้แท็กในไฟล์ welcome.html	18
รูปที่ 1.25:	การแสดงผลของไฟล์ welcome.html	18
รูปที่ 1.26:	การแสดงผลข้อมูลของไฟล์ index.html	18
รูปที่ 1.27:	รูปแบบของการกำหนดไอเฟรมในไฟล์ index_iframe.html	19
รูปที่ 1.28:	การแสดงผลของไฟล์ index_iframe.html	20
รูปที่ 1.29:	รูปแบบของการกำหนดการแสดงผลข้อมูลในไฟล์ info.html	20
รูปที่ 1.30:	การแสดงผลของไฟล์ info.html	20
รูปที่ 1.31:	รูปแบบของการกำหนดการแสดงผลข้อมูลในไฟล์ iframe.html	21
รูปที่ 1.32:	การแสดงผลของไฟล์ iframe.html	21
รูปที่ 1.33:	รูปแบบของการกำหนดการแสดงรายชื่อชนิดกีฬาที่ชอบในไฟล์ sports.html	21
รูปที่ 1.34:	การแสดงผลของไฟล์ sports.html	21
รูปที่ 1.35:	การแสดงผลของไฟล์ iframe.html ในตำแหน่งเป้าหมายที่ชื่อว่า display	22
รูปที่ 1.36:	การแสดงผลของไฟล์ home1.html	23
รูปที่ 1.37:	การแสดงผลของไฟล์ office_hour1.html	23
รูปที่ 1.38:	การแสดงผลของไฟล์ like_dislike1.html	24

รูปที่ 1.39:	การแสดงผลของไฟล์ images1.html.....	24
รูปที่ 1.40:	การแสดงผลของไฟล์ links1.html.....	25
รูปที่ 1.41:	การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ home.html.....	25
รูปที่ 1.42:	การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ office_hour.html.....	26
รูปที่ 1.43:	การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ like_dislike.html.....	26
รูปที่ 1.44:	การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ images.html.....	27
รูปที่ 1.45:	การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ links.html.....	27
รูปที่ 1.46:	การแสดงผลของไฟล์ guestbook.html.....	28
รูปที่ 1.47:	การแสดงผลของไฟล์ thanks.html.....	28
รูปที่ 1.48:	การแสดงผลของไฟล์ f1.html.....	29
รูปที่ 1.49:	การแสดงผลของไฟล์ f2.html.....	29
รูปที่ 2.1:	การใช้ CSS กำหนดรูปแบบให้กับแท็ก <p>.....	32
รูปที่ 2.2:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.1.....	33
รูปที่ 2.3:	การใช้ CSS แบบในบรรทัด (Inline Styles).....	33
รูปที่ 2.4:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.3.....	33
รูปที่ 2.5:	การใช้ CSS แบบภายใน (Internal Style Sheet).....	34
รูปที่ 2.6:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.5.....	34
รูปที่ 2.7:	การสร้างไฟล์ CSS.....	35
รูปที่ 2.8:	การใช้ CSS แบบภายนอก (External Style Sheet).....	35
รูปที่ 2.9:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.8.....	35
รูปที่ 2.10:	การแบ่งคลาสของ Selector หรือแท็ก.....	36
รูปที่ 2.11:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.10.....	37
รูปที่ 2.12:	การกำหนดคลาสแบบทั่วไปสำหรับแท็กต่าง ๆ.....	37
รูปที่ 2.13:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.12.....	38
รูปที่ 2.14:	การกำหนดไอดีแบบทั่วไปสำหรับแท็กต่าง ๆ.....	39
รูปที่ 2.15:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.14.....	39
รูปที่ 2.16:	การกำหนด Pseudo-classes ให้กับอ็อบเจกต์ a.....	40
รูปที่ 2.17:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.16.....	40
รูปที่ 2.18:	คำอธิบายของการแสดงตัวอักษร Times New Roman.....	41
รูปที่ 2.19:	การกำหนด Pseudo-element ให้กับ p.....	41
รูปที่ 2.20:	การแสดงผลของไฟล์ HTML ในรูปที่ 2.19.....	42
รูปที่ 2.21:	สร้าง CSS แบบ Inline Styles จากโปรแกรม Notepad.....	45
รูปที่ 2.22:	ผลที่ได้จากคำสั่งในรูปที่ 2.21.....	45
รูปที่ 2.23:	สร้าง CSS แบบ Internal Style Sheet จากโปรแกรม Notepad.....	45
รูปที่ 2.24:	ผลที่ได้จากคำสั่งในรูปที่ 2.23.....	46
รูปที่ 2.25:	ผลที่ได้จากการใช้คำสั่งแบบ External Style Sheet.....	46
รูปที่ 3.1:	การประมวลผลภาษา PHP ทางฝั่งเซิร์ฟเวอร์และส่งข้อมูล HTML กลับไปที่เครื่องไคลเอนท์.....	49
รูปที่ 3.2:	การกำหนดสตริงแบบ Heredoc.....	50
รูปที่ 3.3:	การแสดงผลตัวอักษรที่เป็น Escaped Characters.....	51
รูปที่ 3.4:	การประกาศตัวแปรแบบ array และ associative array.....	52
รูปที่ 3.5:	การกำหนดชนิดข้อมูลให้กับตัวแปรแบบ Object.....	52

รูปที่ 3.6:	การกำหนดชื่อตัวแปรและการใช้ . (period) เชื่อมตัวแปร.....	53
รูปที่ 3.7:	การประกาศตัวแปรและการกำหนดค่าชนิดต่าง ๆ ให้กับตัวแปรตัวเดียว.....	54
รูปที่ 3.8:	การประกาศตัวแปรและการกำหนดค่าชนิดต่าง ๆ ให้กับตัวแปรหลาย ๆ ตัว.....	54
รูปที่ 3.9:	ตัวอย่างการใช้ตัวแปรระบบ.....	55
รูปที่ 3.10:	ตัวอย่างตัวแปรแบบ global.....	57
รูปที่ 3.11:	ตัวอย่างการเข้าถึงตัวแปรแบบ global จากฟังก์ชัน.....	57
รูปที่ 3.12:	ตัวอย่างตัวแปรแบบ local.....	57
รูปที่ 3.13:	ตัวอย่างค่าคงที่ที่ถูกกำหนดโดย PHP.....	58
รูปที่ 3.14:	ตัวอย่างค่าคงที่ที่ถูกกำหนดโดยผู้ใช้งาน.....	58
รูปที่ 3.15:	การใช้เครื่องหมาย ‘...’ และ “...” ในคำสั่ง echo.....	59
รูปที่ 3.16:	การใช้คำสั่ง printf.....	59
รูปที่ 3.17:	การใส่ข้อสังเกตในภาษา PHP.....	59
รูปที่ 3.18:	การเรียกใช้ฟังก์ชัน date().....	60
รูปที่ 3.19:	การสร้างฟอร์มในไฟล์ form1.html เพื่อรับข้อมูลจากผู้ใช้งาน.....	61
รูปที่ 3.20:	การแสดงผลของไฟล์ form1.html ในรูปที่ 3.19.....	61
รูปที่ 3.21:	การป้อนข้อมูลในแบบฟอร์มของไฟล์ form1.html.....	63
รูปที่ 3.22:	คำสั่งต่าง ๆ ในการแสดงผลของไฟล์ process.php.....	64
รูปที่ 3.23:	การแสดงผลของไฟล์ process.php ที่รับข้อมูลมาจากไฟล์ form1.html ที่ส่งมาแบบ post.....	64
รูปที่ 3.24:	การแสดงผลของไฟล์ process.php ที่รับข้อมูลมาจากไฟล์ form1.html ที่ส่งมาแบบ get.....	65
รูปที่ 3.25:	ผลการแสดงของไฟล์ restaurant_survey.html.....	66
รูปที่ 3.26:	การแสดงผลของไฟล์ thank.php.....	66
รูปที่ 4.1:	การแสดงผลของข้อมูลที่อยู่ในตัวแปรหลาย ๆ ตัวโดยใช้เครื่องหมายจุดทศนิยม (.).....	68
รูปที่ 4.2:	การใช้เครื่องหมายจุดทศนิยม (.) ร่วมกับเครื่องหมายเท่ากับ (=).....	69
รูปที่ 4.3:	การเพิ่มค่าให้กับตัวแปรแบบ Postfix.....	69
รูปที่ 4.4:	การเพิ่มค่าให้กับตัวแปรแบบ Prefix.....	69
รูปที่ 4.5:	การนำค่าไปใช้เมื่อมีการเพิ่มค่าและลดค่าให้กับตัวแปรแบบ Prefix และ Postfix.....	70
รูปที่ 4.6:	ผลของการใช้ตัวดำเนินการเปรียบเทียบ.....	71
รูปที่ 4.7:	การแสดงผลของการใช้ตัวดำเนินการคำสั่ง.....	76
รูปที่ 4.8:	การแสดงผลของการใช้ตัวดำเนินการคำสั่ง.....	76
รูปที่ 4.9:	การแสดงผลของการใช้ตัวดำเนินการจัดการคำสั่ง.....	76
รูปที่ 4.10:	การแสดงผลของการใช้ตัวดำเนินการอระเรย์.....	77
รูปที่ 4.11:	การแสดงผลของการใช้ตัวดำเนินการชนิด.....	77
รูปที่ 4.12:	การแสดงผลของการใช้ตัวดำเนินการแบบเงื่อนไข.....	78
รูปที่ 4.13:	การแสดงผลของการใช้ตัวดำเนินการแบบเงื่อนไข.....	79
รูปที่ 4.14:	การแสดงผลของการใช้ if (เมื่อเป็นจริง).....	80
รูปที่ 4.15:	การแสดงผลของการใช้ if (เมื่อเป็นเท็จ).....	80
รูปที่ 4.16:	การแสดงผลของการใช้ if...else (เมื่อเป็นจริง).....	81
รูปที่ 4.17:	การแสดงผลของการใช้ if...else (เมื่อเป็นเท็จ).....	81
รูปที่ 4.18:	การแสดงผลของการใช้ if...elseif แบบที่ 1.....	82
รูปที่ 4.19:	การแสดงผลของการใช้ if...elseif แบบที่ 2.....	82
รูปที่ 4.20:	การแสดงผลของการใช้ if...elseif แบบที่ 3.....	83

รูปที่ 4.21:	การแสดงผลของการใช้ switch.....	84
รูปที่ 4.22:	การแสดงผลของการใช้ลูป for.....	85
รูปที่ 4.23:	การแสดงผลของการใช้ลูป while.....	86
รูปที่ 4.24:	การแสดงผลของการใช้ลูป do...while.....	86
รูปที่ 4.25:	การแสดงผลของการใช้ลูป foreach แบบที่ 1.....	87
รูปที่ 4.26:	การแสดงผลของการใช้ลูป foreach แบบที่ 2.....	87
รูปที่ 4.27:	การใช้คำสั่ง break;.....	88
รูปที่ 4.28:	การใช้คำสั่ง break; แบบกำหนดขั้น.....	88
รูปที่ 4.29:	การใช้คำสั่ง continue.....	89
รูปที่ 4.30:	ผลการแสดงของไฟล์ calculate.html.....	90
รูปที่ 4.31:	การใส่ข้อมูลในแบบฟอร์มของไฟล์ calculate.html.....	90
รูปที่ 4.32:	ผลการคำนวณและแสดงผลของไฟล์ calculate.php.....	90
รูปที่ 4.33:	ผลการแสดงของไฟล์ minmax.html.....	91
รูปที่ 4.34:	การใส่ข้อมูลในแบบฟอร์มของไฟล์ minmax.html.....	91
รูปที่ 4.35:	ผลการคำนวณและแสดงผลของไฟล์ minmax.php.....	91
รูปที่ 4.36:	ผลการแสดงของไฟล์ loop.html.....	92
รูปที่ 4.37:	การใส่ข้อมูลในแบบฟอร์มของไฟล์ loop.html.....	92
รูปที่ 4.38:	ผลการคำนวณและแสดงผลของไฟล์ loop.php.....	92
รูปที่ 4.39:	การใส่ข้อมูลในแบบฟอร์มของไฟล์ loop.html โดยการป้อนค่า num1 มากกว่า num2.....	93
รูปที่ 4.40:	ผลการคำนวณและแสดงผลของไฟล์ loop.php เมื่อค่าของ num1 มากกว่า num2.....	93
รูปที่ 5.1:	การใช้ฟังก์ชันเกี่ยวกับวันเวลา.....	96
รูปที่ 5.2:	การใช้ฟังก์ชันเกี่ยวกับอะเรย์ในการเก็บค่า คีย์ และแสดงผล.....	99
รูปที่ 5.3:	การใช้ฟังก์ชันเกี่ยวกับอะเรย์ในการเพิ่มค่า ตัดค่า และแสดงผล.....	101
รูปที่ 5.4:	การใช้ฟังก์ชันเกี่ยวกับชนิดตัวอักษร.....	103
รูปที่ 5.5:	การใช้ฟังก์ชันเกี่ยวกับโดเรคทอรี.....	104
รูปที่ 5.6:	การใช้ฟังก์ชันเกี่ยวกับสตริง.....	108
รูปที่ 5.7:	การใช้ฟังก์ชันเกี่ยวกับคณิตศาสตร์.....	110
รูปที่ 5.8:	แบบฟอร์มการส่งเมล.....	112
รูปที่ 5.9:	การส่งค่าตัวแปรต่าง ๆ ในแบบฟอร์ม.....	113
รูปที่ 5.10:	การใช้ฟังก์ชันเกี่ยวกับเพิ่มข้อมูล.....	115
รูปที่ 5.11:	การส่งแบบการผ่านค่า.....	117
รูปที่ 5.12:	การส่งแบบการผ่านตัวอ้างอิง.....	118
รูปที่ 5.13:	การส่งแบบการกำหนดค่าปริยาย.....	119
รูปที่ 5.14:	การทำงานของตัวแปรฟังก์ชัน.....	120
รูปที่ 5.15:	การแสดงผลวัน เดือน และปี เป็นภาษาไทย.....	122
รูปที่ 5.16:	แบบฟอร์มของไฟล์ form_info.php.....	122
รูปที่ 5.17:	การแสดงผลของไฟล์ display_form_info.php.....	123
รูปที่ 5.18:	ข้อมูลในไฟล์ Excel.....	123
รูปที่ 5.19:	แบบฟอร์มของไฟล์ input_text.html.....	123
รูปที่ 5.20:	การแสดงผลของไฟล์ column3.php.....	124
รูปที่ 6.1:	การใช้ฟังก์ชัน fopen() และ fclose().....	128

รูปที่ 6.2:	การใช้ตัวดำเนินการควบคุมข้อผิดพลาด @.....	128
รูปที่ 6.3:	การใช้ฟังก์ชัน flock().....	130
รูปที่ 6.4:	การแสดงผลของข้อมูลที่อยู่ในแฟ้ม data2.txt โดยโปรแกรม Notepad.....	131
รูปที่ 6.5:	การบันทึกข้อมูลเป็นสกุล CSV.....	132
รูปที่ 6.6:	การแสดงผลของแฟ้มข้อมูล data2_csv.csv โดยโปรแกรม Notepad.....	132
รูปที่ 6.7:	การแสดงผลของแฟ้มข้อมูล data2_csv.csv โดยโปรแกรม Microsoft Excel.....	133
รูปที่ 6.8:	การใช้ลูป while กับฟังก์ชัน feof() เพื่ออ่านแฟ้มข้อมูลและแสดงผล.....	133
รูปที่ 6.9:	ข้อมูลจากแฟ้มข้อมูลหนึ่งที่ถูกแสดงออกโดยโปรแกรม Notepad.....	134
รูปที่ 6.10:	ข้อมูลจากแฟ้มข้อมูลหนึ่งที่ถูกแสดงออกโดยฟังก์ชัน fgets().....	134
รูปที่ 6.11:	ข้อมูลจากแฟ้มข้อมูลหนึ่งที่ถูกแสดงออกโดยฟังก์ชัน fgets() และยกเว้นแท็ก <i>.....	135
รูปที่ 6.12:	การใช้ฟังก์ชัน fgetsv() เพื่อแยกข้อมูลออกเป็นส่วน ๆ.....	135
รูปที่ 6.13:	การใช้ฟังก์ชัน fgetsv() เพื่อแยกข้อมูลออกเป็นส่วน ๆ แล้วแสดงผลในตาราง.....	136
รูปที่ 6.14:	การใช้ฟังก์ชัน readfile().....	137
รูปที่ 6.15:	การใช้ฟังก์ชัน fpassthru().....	138
รูปที่ 6.16:	การใช้ฟังก์ชัน file().....	139
รูปที่ 6.17:	การใช้ฟังก์ชัน file_get_contents().....	140
รูปที่ 6.18:	การใช้ฟังก์ชัน fread().....	141
รูปที่ 6.19:	การใช้ฟังก์ชัน fgetc().....	142
รูปที่ 6.20:	การใช้ฟังก์ชัน file_exists(), filesize(), ftell(), fseek() และ rewind().....	144
รูปที่ 6.21:	เพิ่มจุดเชื่อม View Guestbook.....	147
รูปที่ 6.22:	การแสดงความของไฟล์ process.php.....	147
รูปที่ 6.23:	หน้าต่างของไฟล์ guestbook.html.....	148
รูปที่ 6.24:	การดึงข้อมูลจากแฟ้มข้อมูลเพื่อแสดงผลของไฟล์ view_guestbook.php.....	148
รูปที่ 6.25:	การดึงข้อมูลจากแฟ้มข้อมูลเพื่อแสดงผลของไฟล์ view_guestbook_table.php.....	149
รูปที่ 7.1:	การแสดงผลของแฟ้มข้อมูล calendar.php.....	152
รูปที่ 7.2:	การแสดงผลของแฟ้มข้อมูล image.php.....	152
รูปที่ 7.3:	ข้อมูลในแฟ้ม info.inc.....	153
รูปที่ 7.4:	การแสดงผลข้อผิดพลาดจากการใช้คำสั่ง require() ดึงแฟ้มข้อมูล.....	154
รูปที่ 7.5:	การแสดงผลค่าเตือนจากการใช้คำสั่ง include() ดึงแฟ้มข้อมูล.....	154
รูปที่ 7.6:	การแสดงผลของการใช้คำสั่ง require().....	155
รูปที่ 7.7:	การแสดงผลของการใช้คำสั่ง include().....	155
รูปที่ 7.8:	การทดสอบเวอร์ชัน PHP และสกุลของรูปภาพชนิดต่าง ๆ ที่ใช้งานได้.....	158
รูปที่ 7.9:	วาดรูปสี่เหลี่ยมผืนผ้าสีชมพูขนาด 500 x 150 พิกเซล.....	160
รูปที่ 7.10:	วาดรูปวงกลมสีแดงขนาด 150 พิกเซลอยู่ตำแหน่งตรงกลางของพื้นที่วาดรูป.....	162
รูปที่ 7.11:	วาดรูปสี่เหลี่ยมจัตุรัสสีดำขนาด 200 พิกเซลและวงกลมสีแดงขนาด 150 พิกเซล เส้นขอบสีขาวขนาด 5 พิกเซลอยู่ตำแหน่งตรงกลาง.....	163
รูปที่ 7.12:	วาดรูปสี่เหลี่ยมจัตุรัสสีดำขนาด 200 พิกเซลและเส้นสีขาวตามแนวตั้งและแนวนอน โดยเว้นระยะห่าง 10 พิกเซล.....	165
รูปที่ 7.13:	วาดรูปสตรึงจากตัวอักษรระบบกลุ่มที่ 1 ถึง 5 และชนิดและขนาดของตัวอักษรที่กำหนดขึ้นเอง.....	167
รูปที่ 7.14:	วาดรูปโพลีกอนและสตรึงในแนวนอนและตั้ง.....	169
รูปที่ 7.15:	การสร้างและปรับแต่งรูปภาพจากแฟ้มรูปภาพ.....	171

รูปที่ 7.16:	การแสดงผลผิดพลาดเมื่อไม่สามารถเปิดเพิ่มรูปภาพได้.....	172
รูปที่ 7.17:	วาดรูปธงชาติไทย.....	174
รูปที่ 7.18:	วาดรูปลายเส้น.....	174
รูปที่ 7.19:	วาดรูปตารางหมากรุก.....	175
รูปที่ 7.20:	วาดรูปกราฟแท่ง.....	175
รูปที่ 8.1:	ข้อมูลในตาราง student.....	178
รูปที่ 8.2:	ข้อมูลในตาราง student และ register.....	179
รูปที่ 8.3:	ความสัมพันธ์แบบ One-to-One.....	179
รูปที่ 8.4:	ความสัมพันธ์แบบ One-to-Many.....	180
รูปที่ 8.5:	หน้าต่างของ Command Prompt.....	180
รูปที่ 8.6:	หน้าต่างของการเปลี่ยนไดเรกทอรีไปที่ MySQL.....	181
รูปที่ 8.7:	หน้าต่างของการแสดงผลผิดพลาดเมื่อเข้าสู่ระบบฐานข้อมูล.....	181
รูปที่ 8.8:	หน้าต่างของการแสดงผลเมื่อสามารถเข้าสู่ระบบฐานข้อมูล MySQL.....	181
รูปที่ 8.9:	หน้าต่างของการแสดงผลเมื่อสร้างฐานข้อมูล webapp150.....	185
รูปที่ 8.10:	หน้าต่างของการแสดงฐานข้อมูลทั้งหมดในระบบ.....	185
รูปที่ 8.11:	การเรียกใช้ฐานข้อมูล webapp150.....	185
รูปที่ 8.12:	การสร้างและกำหนดรูปแบบตาราง student.....	186
รูปที่ 8.13:	การสร้างและกำหนดรูปแบบตาราง register.....	186
รูปที่ 8.14:	การแสดงผลตารางทั้งหมดในฐานข้อมูล webapp150.....	186
รูปที่ 8.15:	รายละเอียดโครงสร้างตาราง student.....	187
รูปที่ 8.16:	การใช้คำสั่ง describe ดูรายละเอียดโครงสร้างตาราง register.....	187
รูปที่ 8.17:	การเพิ่มข้อมูลเข้าไปในตาราง student.....	190
รูปที่ 8.18:	การเพิ่มข้อมูลเข้าไปในตาราง register.....	190
รูปที่ 8.19:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student.....	191
รูปที่ 8.20:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง register.....	191
รูปที่ 8.21:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student ใช้คอลัมน์ earn เป็นหลัก เรียงลำดับจากน้อยไปมาก (Ascending Order).....	191
รูปที่ 8.22:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student ใช้คอลัมน์ earn เป็นหลัก เรียงลำดับจากมากไปน้อย (Descending Order).....	192
รูปที่ 8.23:	การใช้คำสั่ง update เพื่อเปลี่ยนแปลงข้อมูลในตาราง register.....	192
รูปที่ 8.24:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง register หลังมีการเปลี่ยนแปลง.....	192
รูปที่ 8.25:	การใช้คำสั่ง delete เพื่อลบแถวข้อมูล.....	193
รูปที่ 8.26:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง register หลังมีการลบแถวข้อมูล.....	193
รูปที่ 8.27:	การใช้คำสั่ง insert ในการเพิ่มข้อมูลลงในตาราง register และใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง.....	193
รูปที่ 8.28:	การสร้างตัวชี้ชื่อ xid ที่ไม่ซ้ำกัน (Unique) ให้กับคอลัมน์ studentid ในตาราง register.....	194
รูปที่ 8.29:	การใช้คำสั่ง describe ดูรายละเอียดโครงสร้างตาราง register หลังจากการสร้างตัวชี้ให้กับคอลัมน์ studentid.....	194
รูปที่ 8.30:	การลบตัวชี้ xid ออกจากตาราง register.....	195
รูปที่ 8.31:	การใช้คำสั่ง describe ดูรายละเอียดโครงสร้างตาราง register หลังจากการลบตัวชี้ xid ออกจากตาราง register.....	195

รูปที่ 8.32:	การใช้คำสั่ง alter เปลี่ยนชื่อตาราง	196
รูปที่ 8.33:	การใช้คำสั่ง alter เพิ่มคอลัมน์ในตาราง	196
รูปที่ 8.34:	การใช้คำสั่ง desc ดูรายละเอียดโครงสร้างตาราง student_list หลังการเพิ่มคอลัมน์ telephone.....	196
รูปที่ 8.35:	การใช้คำสั่ง alter เพิ่มเปลี่ยนชื่อคอลัมน์จาก earn เป็น earned ในตาราง student_list.....	197
รูปที่ 8.36:	การใช้คำสั่ง desc ดูรายละเอียดโครงสร้างตาราง student_list หลังจากเปลี่ยนชื่อคอลัมน์.....	197
รูปที่ 8.37:	การใช้คำสั่ง update เพื่อแก้ไขข้อมูลเบอร์โทรศัพท์ในคอลัมน์ telephone ของตาราง student_list.....	197
รูปที่ 8.38:	การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student_list หลังจากมีการปรับปรุงข้อมูล เบอร์โทรศัพท์.....	198
รูปที่ 8.39:	การใช้คำสั่ง create table สร้างตารางชื่อว่า tb_test.....	198
รูปที่ 8.40:	การใช้คำสั่ง show tables เพื่อแสดงชื่อตารางทั้งหมด.....	198
รูปที่ 8.41:	การใช้คำสั่ง drop table เพื่อลบตาราง tb_test และคำสั่ง show tables เพื่อแสดงชื่อตารางทั้งหมด	199
รูปที่ 8.42:	การเปิดแฟ้มข้อมูลที่ถูกเขียนลงโดยคำสั่ง OUTFILE	199
รูปที่ 8.43:	แบบฟอร์ม guestbook.....	204
รูปที่ 8.44:	แบบฟอร์ม student_info.....	204
รูปที่ 9.1:	การใช้คำสั่ง PHP เชื่อมต่อระบบฐานข้อมูล.....	209
รูปที่ 9.2:	การแสดงผลผิดพลาดในการเชื่อมต่อระบบฐานข้อมูล.....	210
รูปที่ 9.3:	แฟ้มข้อมูลสกุล .inc เพื่อเชื่อมต่อระบบฐานข้อมูล.....	211
รูปที่ 9.4:	การรวมแฟ้มข้อมูลสกุล .inc ในการติดต่อระบบฐานข้อมูลแล้วใช้ PHP ในการแสดงฐานข้อมูลทั้งหมด ในระบบ	211
รูปที่ 9.5:	การลบฐานข้อมูล webapp15000 ออกจากระบบฐานข้อมูล	212
รูปที่ 9.6:	การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อตารางทั้งหมด.....	213
รูปที่ 9.7:	การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์ ชนิด และขนาดในตาราง register	215
รูปที่ 9.8:	การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์โดยใช้ภาษาสอบถาม SQL.....	216
รูปที่ 9.9:	การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์และรายละเอียดต่าง ๆ ในตาราง register.....	218
รูปที่ 9.10:	การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์และรายละเอียดต่าง ๆ ในตาราง student_list.....	220
รูปที่ 9.11:	การเรียกใช้ฐานข้อมูลชื่อ webapp150 เพื่อปรับปรุงข้อมูลในคอลัมน์ telephone ในตาราง student_list.....	223
รูปที่ 9.12:	แบบฟอร์มสำหรับรับข้อมูลแล้วทำการบันทึกลงในตาราง student_list.....	224
รูปที่ 9.13:	การแสดงผลเมื่อทำการบันทึกข้อมูลลงในตาราง student_list ได้สำเร็จ.....	225
รูปที่ 9.14:	แสดงข้อความ Invalid ID เมื่อผู้ใช้ป้อน id ซ้ำกับข้อมูลที่มีอยู่ในตาราง student_list	226
รูปที่ 9.15:	สร้างฐานข้อมูล guestbook สำเร็จ	227
รูปที่ 9.16:	สร้างตาราง record ในฐานข้อมูล guestbook สำเร็จ.....	227
รูปที่ 9.17:	แบบฟอร์มการรับข้อมูลเพื่อบันทึกลงในตาราง record ในฐานข้อมูล guestbook.....	227
รูปที่ 9.18:	บันทึกข้อมูลลงในตาราง record ในฐานข้อมูล guestbook สำเร็จ.....	228
รูปที่ 9.19:	แสดงผลข้อมูลในตาราง record.....	228
รูปที่ 10.1:	การตั้งค่าและบันทึกข้อมูลแบบคุกกี้.....	231
รูปที่ 10.2:	การแสดงผลของการใช้คำสั่งจากรูปที่ 10.1.....	231
รูปที่ 10.3:	การตรวจสอบค่าและข้อมูลแบบคุกกี้.....	232
รูปที่ 10.4:	การแสดงผลของการใช้คำสั่งจากรูปที่ 10.3.....	233

รูปที่ 10.5:	การแสดงผลของการเรียกไฟล์ last_visit2.php อีกครั้ง.....	233
รูปที่ 10.6:	การเพิ่มคำสั่งในการปรับปรุงข้อมูลของวันและเวลาครั้งล่าสุดที่เข้าเว็บ.....	233
รูปที่ 10.7:	การแสดงผลของการใช้คำสั่งจากรูปที่ 10.6.....	234
รูปที่ 10.8:	การใช้คำสั่งในการปรับปรุงข้อมูลของวันและเวลาครั้งล่าสุดที่เข้าเว็บให้อยู่ในส่วนหัว.....	234
รูปที่ 10.9:	การแสดงผลของการใช้ไฟล์ last_visit4.php ครั้งที่ 1.....	234
รูปที่ 10.10:	การแสดงผลของการใช้ไฟล์ last_visit4.php ครั้งที่ 2.....	235
รูปที่ 10.11:	การแสดงผลของการใช้ไฟล์ last_visit4.php ครั้งที่ 3.....	235
รูปที่ 10.12:	การเลือกเมนู Tools แล้วตามด้วย Options.....	236
รูปที่ 10.13:	การเลือกเมนู Privacy แล้วกดปุ่ม Show Cookies.....	236
รูปที่ 10.14:	การเลือกคุกกี้ที่ต้องการลบทิ้ง.....	237
รูปที่ 10.15:	การเลือกเมนู Tools แล้วตามด้วย Clear Private Data.....	237
รูปที่ 10.16:	การเลือกหัวข้อต่าง ๆ ที่ต้องการลบในหน้าต่าง Clear Private Data.....	238
รูปที่ 10.17:	คำสั่งต่าง ๆ ในไฟล์ index.php.....	240
รูปที่ 10.18:	ผลการแสดงของไฟล์ index.php.....	241
รูปที่ 10.19:	คำสั่งต่าง ๆ ในไฟล์ login.php.....	242
รูปที่ 10.20:	การป้อนข้อมูลเพื่อเข้าสู่ระบบ.....	245
รูปที่ 10.21:	แสดงข้อความเมื่อชื่อบริษัทผ่านไม่ถูกต้อง.....	245
รูปที่ 10.22:	แสดงข้อความเมื่อชื่อบริษัทผ่านถูกต้อง.....	245
รูปที่ 10.23:	แสดงหน้า home.php.....	246
รูปที่ 10.24:	คำสั่งต่าง ๆ ในไฟล์ home.php.....	246
รูปที่ 10.25:	คำสั่งต่าง ๆ ในไฟล์ header.php.....	247
รูปที่ 10.26:	คำสั่งต่าง ๆ ในไฟล์ error.php.....	248
รูปที่ 10.27:	การแสดงผลของไฟล์ error.php.....	248
รูปที่ 10.28:	คำสั่งต่าง ๆ ในไฟล์ products.php.....	249
รูปที่ 10.29:	การแสดงผลของไฟล์ products.php.....	250
รูปที่ 10.30:	คำสั่งต่าง ๆ ในไฟล์ products.php.....	251
รูปที่ 10.31:	การแสดงผลของไฟล์ about.php.....	252
รูปที่ 10.32:	คำสั่งต่าง ๆ ในไฟล์ logout.php.....	252
รูปที่ 10.33:	การแสดงผลของไฟล์ logout.php.....	252
รูปที่ 10.34:	คำสั่งต่าง ๆ ในไฟล์ signin_form.php.....	253
รูปที่ 10.35:	การแสดงผลของไฟล์ signin_form.php.....	254
รูปที่ 10.36:	การแสดงผลของไฟล์ signin.php เมื่อป้อนข้อมูลไม่ครบถ้วน.....	254
รูปที่ 10.37:	คำสั่งต่าง ๆ ในไฟล์ signin_form.php.....	255
รูปที่ 10.38:	การแสดงผลของไฟล์ index.php และรหัสลับ FZE.....	258
รูปที่ 10.39:	แสดงการป้อนข้อมูลไม่ครบถ้วน.....	258
รูปที่ 10.40:	แสดงข้อความ “รหัสลับไม่ถูกต้อง”.....	259
รูปที่ 10.41:	การแสดงผลของไฟล์ index.php พร้อมกับสร้างรหัสลับขึ้นมาใหม่คือ UKB.....	259
รูปที่ 10.42:	แสดงการป้อนข้อมูลครบถ้วนและถูกต้อง.....	260
รูปที่ 10.43:	แสดงข้อความ “ยินดีต้อนรับ” ตามด้วยชื่อ.....	260
รูปที่ 10.44:	การแสดงผลของไฟล์ welcome.php.....	260
รูปที่ 11.1:	หน้าต่างกลุ่มของคำแนะนำเมื่อผู้ใช้ป้อนตัวอักษร c.....	262

รูปที่ 11.2:	หน้าต่างกลุ่มของคำแนะนำเมื่อผู้ใช้ป้อนตัวอักษร ca.....	262
รูปที่ 11.3:	รูปแบบการเปรียบเทียบการส่งข้อมูลของเว็บแอปพลิเคชันแบบดั้งเดิมและแบบเอแจ็ก	263
รูปที่ 11.4:	รูปแบบเปรียบเทียบการติดต่อระหว่างเว็บเบราว์เซอร์ทางฝั่งผู้ใช้กับเว็บเซิร์ฟเวอร์แบบ Synchronous และ Asynchronous.....	263
รูปที่ 11.5:	การกำหนด instance ของคลาสสำหรับเว็บเบราว์เซอร์ Internet Explorer ไฟล์ create_ie.html.....	267
รูปที่ 11.6:	การแสดงผลของไฟล์ create_ie.html ในรูปที่ 11.5 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Internet Explorer.....	268
รูปที่ 11.7:	การแสดงผลของไฟล์ create_ie.html ในรูปที่ 11.5 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Firefox	268
รูปที่ 11.8:	การกำหนด instance ของคลาสสำหรับเว็บเบราว์เซอร์ Firefox ในไฟล์ create_firefox.html	269
รูปที่ 11.9:	การแสดงผลของไฟล์ create_firefox.html ในรูปที่ 11.8 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Firefox	270
รูปที่ 11.10:	การแสดงผลของไฟล์ create_firefox.html ในรูปที่ 11.8 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Internet Explorer.....	270
รูปที่ 11.11:	การกำหนด instance ของคลาสสำหรับเว็บเบราว์เซอร์ IE7, Firefox, Safari, Opera ในไฟล์ create.html.....	271
รูปที่ 11.12:	คำสั่งในไฟล์ converter_form.html	272
รูปที่ 11.13:	คำสั่งในไฟล์ convert.php.....	274
รูปที่ 11.14:	การแสดงผลของไฟล์ converter_form.html	275
รูปที่ 11.15:	การแสดงผลจากแบบฟอร์มของไฟล์ converter_form.html เมื่อป้อนข้อมูล.....	275
รูปที่ 11.16:	คำสั่งในไฟล์ converter_form_post.html ที่มีวิธีการส่งข้อมูลแบบ POST	276
รูปที่ 11.17:	คำสั่งในไฟล์ convert_post.php ที่มีวิธีการส่งข้อมูลแบบ POST	277
รูปที่ 11.18:	ข้อมูลในไฟล์ note.xml.....	278
รูปที่ 11.19:	ข้อมูลในไฟล์ note.js.....	278
รูปที่ 11.20:	ข้อมูลในไฟล์ note_form.html	279
รูปที่ 11.21:	คำสั่งต่าง ๆ ในไฟล์ note_display.php.....	280
รูปที่ 11.22:	การแสดงผลของไฟล์ note_form.html	282
รูปที่ 11.23:	แสดงข้อมูลตัวเลือกของไฟล์ note_form.html.....	282
รูปที่ 11.24:	แสดงผลจากการเลือกข้อมูลของไฟล์ note_form.html.....	282
รูปที่ 11.25:	ข้อมูลตาราง contact ในฐานะข้อมูล friends.....	283
รูปที่ 11.26:	คำสั่งต่าง ๆ ในไฟล์ contact_form.html	283
รูปที่ 11.27:	การแสดงผลจากไฟล์ contact_form.html.....	283
รูปที่ 11.28:	คำสั่งต่าง ๆ ในไฟล์ contact.js.....	284
รูปที่ 11.29:	คำสั่งต่าง ๆ ในไฟล์ contact.php.....	284
รูปที่ 11.30:	แสดงผลของข้อมูลทั้งหมดที่เริ่มต้นด้วยตัวอักษร s	285
รูปที่ 11.31:	แสดงผลของข้อมูลทั้งหมดที่เริ่มต้นด้วยตัวอักษร so.....	285
รูปที่ 11.32:	ข้อมูลในตาราง contact.....	286
รูปที่ 11.33:	แสดงผลของไฟล์ search_form.html.....	286
รูปที่ 11.34:	แสดงข้อมูลเมื่อผู้ใช้ป้อนตัวอักษร s.....	287
รูปที่ 11.35:	แสดงข้อมูลเมื่อผู้ใช้ป้อนตัวอักษร su.....	287
รูปที่ 11:	หน้าต่างเริ่มต้นการติดตั้งโปรแกรม AppServ.....	289

รูปที่ ก2:	หน้าต่างแสดงลิขสิทธิ์.....	290
รูปที่ ก3:	หน้าต่างแสดงตำแหน่งการติดตั้งโปรแกรม.....	290
รูปที่ ก4:	หน้าต่างแสดงตัวเลือกในการติดตั้ง.....	291
รูปที่ ก5:	หน้าต่างแสดงรายละเอียดเกี่ยวกับ HTTP Server.....	291
รูปที่ ก6:	หน้าต่างแสดงการตั้งค่าการเข้าถึงระบบฐานข้อมูล.....	292
รูปที่ ก7:	หน้าต่างแสดงการทำงานขณะทำการติดตั้ง.....	292
รูปที่ ก8:	หน้าต่างแสดงการติดตั้งสำเร็จ.....	293
รูปที่ ก9:	หน้าต่างแสดงข้อมูลหลังการติดตั้ง.....	293
รูปที่ ก10:	หน้าต่างแสดงรายละเอียดของโปรแกรมต่าง ๆ ที่ติดตั้งลงไป.....	294
รูปที่ ก11:	หน้าต่างแสดงตำแหน่งที่อยู่ของสคริปต์ .php.....	294
รูปที่ ก12:	หน้าต่างแสดงรายละเอียดของโปรแกรม PHP.....	295
รูปที่ ข1:	หน้าต่างแรกเมื่อบูตจากแผ่น SUTinsServer.....	301
รูปที่ ข2:	หน้าต่างแสดงตัวเลือกกราฟิก.....	301
รูปที่ ข3:	หน้าต่างแสดงการเลือกปุ่ม “ตรวจสอบแผ่น”.....	302
รูปที่ ข4:	หน้าต่างแสดงการตรวจสอบแผ่นสำเร็จ.....	302
รูปที่ ข5:	หน้าต่างแสดงปุ่มเลือกขนาดของไฟล์ swap.....	302
รูปที่ ข6:	หน้าต่างแสดงปุ่มให้เลือกระบบที่เป็น 32 หรือ 64 บิต.....	303
รูปที่ ข7:	หน้าต่างแสดงตำแหน่งและรูปแบบที่ต้องการลงระบบ SUTinsServer.....	303
รูปที่ ข8:	หน้าต่างแสดงรายละเอียดและตำแหน่งที่เลือกลงระบบ SUTinsServer.....	304
รูปที่ ข9:	หน้าต่างแสดงขณะกำลังติดตั้งระบบ SUTinsServer.....	304
รูปที่ ข10:	หน้าต่างแสดงการติดตั้งระบบ SUTinsServer เสร็จเรียบร้อย.....	304
รูปที่ ข11:	หน้าต่างแสดงตัวเลือกเข้าสู่ระบบต่าง ๆ.....	305
รูปที่ ข12:	หน้าต่างล็อกอินเข้าสู่ระบบปฏิบัติการลินุกซ์เซิร์ฟเวอร์.....	306
รูปที่ ข13:	หน้าต่างแรกของลินุกซ์เซิร์ฟเวอร์.....	306
รูปที่ ค1:	หน้าต่างเริ่มต้นการใช้งาน phpMyAdmin.....	307
รูปที่ ค2:	หน้าต่างป้อนชื่อและรหัสผ่านเข้าสู่ระบบการจัดการฐานข้อมูล.....	308
รูปที่ ค3:	หน้าต่างเริ่มต้นของโปรแกรมการจัดการฐานข้อมูล phpMyAdmin.....	308
รูปที่ ค4:	หน้าต่างแสดงการเปลี่ยนการแสดงผลจากภาษาอังกฤษเป็นภาษาไทย.....	309
รูปที่ ค5:	หน้าต่างแสดงรายละเอียดและคำสั่งต่าง ๆ เป็นภาษาไทย.....	309
รูปที่ ค6:	หน้าต่างแสดงฐานข้อมูลทั้งหมดที่มีอยู่ในระบบ.....	310
รูปที่ ค7:	หน้าต่างแสดงการสร้างฐานข้อมูลใหม่ชื่อว่า “bookstore”.....	311
รูปที่ ค8:	หน้าต่างแสดงฐานข้อมูล “bookstore”.....	311
รูปที่ ค9:	หน้าต่างแสดงการลบฐานข้อมูล “bookstore”.....	312
รูปที่ ค10:	หน้าต่างยืนยันการลบฐานข้อมูล “bookstore”.....	312
รูปที่ ค11:	หน้าต่างแสดงการสร้างตาราง “book_info”.....	313
รูปที่ ค12:	หน้าต่างตาราง “book_info”.....	313
รูปที่ ค13:	หน้าต่างตาราง “book_info”.....	314
รูปที่ ค14:	หน้าต่างตาราง “book_info” เมื่อป้อนรายละเอียดแล้วกดปุ่ม “บันทึก”.....	314
รูปที่ ค15:	หน้าต่างการเพิ่มฟิลด์ในตาราง “book_info”.....	315
รูปที่ ค16:	หน้าต่างฟิลด์ใหม่ในตาราง “book_info”.....	315
รูปที่ ค17:	หน้าต่างตาราง “book_info” ที่มีฟิลด์ “price” เพิ่มขึ้น.....	315

รูปที่ ค18:	หน้าต่างตาราง “book_info” เมื่อป้อนรายละเอียดของฟิลด์ “price” แล้วกดปุ่ม “บันทึก” 316
รูปที่ ค19:	แทรกข้อมูลลงในตาราง “book_info” 316
รูปที่ ค20:	หน้าต่างช่องข้อมูลต่าง ๆ ในตาราง “book_info” 317
รูปที่ ค21:	หน้าต่างแสดงการป้อนข้อมูลต่าง ๆ ในตาราง “book_info” 317
รูปที่ ค22:	หน้าต่างแสดงการป้อนข้อมูลต่าง ๆ ในตาราง “book_info” 318
รูปที่ ค23:	หน้าต่างแสดงการกดปุ่ม “เปิดดู” 319
รูปที่ ค24:	หน้าต่างแสดงการเปิดดูข้อมูลทั้งหมดในตาราง “book_info” 319
รูปที่ ค25:	หน้าต่างแสดงการแทรกข้อมูลเข้าในตารางอีก 2 ข้อมูลในตาราง “book_info” 320
รูปที่ ค26:	หน้าต่างแสดงการเปิดดูข้อมูลทั้งหมดในตาราง “book_info” หลังจากแทรกข้อมูลเข้าไปอีก 2 ข้อมูล. 321
รูปที่ ค27:	หน้าต่างแสดงการแก้ไขข้อมูล..... 322
รูปที่ ค28:	หน้าต่างแสดงข้อมูลที่เลือกแก้ไข..... 322
รูปที่ ค29:	หน้าต่างแสดงการเปลี่ยนแปลงข้อมูล 323
รูปที่ ค30:	หน้าต่างแสดงข้อมูลทั้งหมดหลังจากการเปลี่ยนแปลงข้อมูล 323
รูปที่ ค31:	หน้าต่างแสดงการเลือกข้อมูลที่ 2 ที่ต้องการลบ..... 323
รูปที่ ค32:	หน้าต่างถามย้ำต้องการลบ..... 324
รูปที่ ค33:	หน้าต่างแสดงข้อมูลที่เหลือหลังจากการลบข้อมูลที่ 2..... 324



บทที่ 1 แนะนำภาษา HTML

<Introduction to HTML>

บทนำ

การส่งข้อมูลหรือข่าวสารต่าง ๆ ได้มีการเปลี่ยนแปลงวิธีการส่งและรับซึ่งจากเดิมนั้นต้องใช้การส่งทางจดหมายหรือโทรสาร ต่อมาเทคโนโลยีคอมพิวเตอร์ทางด้านซอฟต์แวร์และฮาร์ดแวร์ได้มีการพัฒนาอย่างต่อเนื่อง ทำให้มีคุณภาพดีขึ้นและราคาต่ำลง ก่อให้เกิดการขยายตัวของตลาดทางด้านคอมพิวเตอร์สูงขึ้น การติดต่อสื่อสารผ่านระบบอินเทอร์เน็ต ช่วยให้การส่งข่าวสารหรือจดหมายในลักษณะของข้อมูลแบบดิจิทัล ไปถึงจุดหมายปลายทางได้อย่างสะดวก รวดเร็ว และประหยัดอย่างเห็นได้ชัด อินเทอร์เน็ตจึงเป็นสิ่งที่เอื้ออำนวยให้ทั้งผู้นำเสนอข้อมูลหรือข่าวสารและผู้บริโภคได้ใช้ประโยชน์อย่างเต็มที่ เป็นสื่อกลางในการแลกเปลี่ยนข้อมูล โดยอาศัยภาษาที่เป็นมาตรฐานสากลเป็นตัวช่วย นั่นก็คือภาษา HTML ผ่านโปรแกรมเว็บเบราว์เซอร์ ดังนั้น จึงจำเป็นต้องเข้าใจหลักการทำงานของภาษา HTML เพื่อนำมาประยุกต์และใช้ในการออกแบบการนำเสนอข้อมูลหรือรับข้อมูลต่าง ๆ ได้อย่างเต็มที่

วัตถุประสงค์

1. เพื่อให้รู้ประวัติความเป็นมาของภาษา HTML
2. เพื่อให้เข้าใจหลักการใช้แท็กต่าง ๆ ในภาษา HTML
3. เพื่อฝึกเขียนและทดสอบการเขียนโค้ด HTML

เนื้อหา

1.1 ประวัติและความเป็นมาของภาษา HTML

HTML ย่อมาจาก HyperText Markup Language เป็นภาษาที่ใช้ในการแสดงข้อมูลต่าง ๆ บนเว็บเบราว์เซอร์ ถูกคิดค้นขึ้นโดย ทิม เบอร์เนิร์ส-ลี (Tim Berners-Lee) ขณะทำงานอยู่ที่ CERN (Centre Europeen de Recherche Nucleaire) ในกรุงเจนีวา ประเทศสวิตเซอร์แลนด์ ในปี ค.ศ. 1990 ทิม เบอร์เนิร์ส-ลี เป็นนักวิจัยผู้มีประสบการณ์ทางด้าน การประมวลผลตัวอักษร (text processing), การติดต่อสื่อสาร (communications), และซอฟต์แวร์แบบเวลาจริง (real-time software) เขาได้เขียนข้อเสนอโครงการชื่อว่า Hypertext เพื่อพัฒนาการติดต่อสื่อสารในชุมชนที่เรียกว่า High Energy Physics Community

ทิม เบอร์เนิร์ส-ลี ได้รวมเทคโนโลยีสมัยใหม่ 3 อย่างในข้อเสนอโครงการคือ

1. HTML (HyperText Markup Language) ใช้ในการเขียนข้อมูลข่าวสารบนเว็บ
2. HTTP (HyperText Transfer Protocol) ใช้ในการส่งข้อมูลข่าวสาร
3. Web Browser Client Software โปรแกรมเว็บเบราว์เซอร์ฝั่งลูกข่ายเพื่อใช้ในการรับ แปลง และแสดงผลของข้อมูล

โดยให้ความสำคัญอยู่ที่ การเข้าถึงข้อมูลของลูกค้าโดยใช้อินเทอร์เน็ตเป็นสื่อกลางนั้น จะต้องเข้าได้จากเครื่องคอมพิวเตอร์ทุกเครื่อง ทุกระบบปฏิบัติการ

ในปี ค.ศ. 1991 ได้มีการทดลองส่งข้อมูลโดยใช้เทคโนโลยีทั้ง 3 แบบ (HTML, HTTP และ Web Browser Client Software ซึ่งได้ตั้งชื่อเป็น WWW) ที่ CERN พร้อมทั้งมีบริการส่งไฟล์ไปยังเครื่องลูกค้า ซึ่งนี่คือที่มาของคำว่า Web Server และเป็น Web Server เครื่องแรกของโลก ต่อมาในปี ค.ศ. 1992 ได้มี Web Server มากกว่า 50 เครื่อง ซึ่งส่วนใหญ่แล้วเครื่องเซิร์ฟเวอร์เหล่านี้จะตั้งอยู่ในสถาบันการศึกษาและศูนย์วิจัยต่าง ๆ ในกลางปี ค.ศ. 1992 จำนวนเครื่องเซิร์ฟเวอร์มีมากถึง 720,000 เครื่อง และเกินกว่า 24 ล้านเครื่องในเดือนเมษายน ค.ศ. 2001

ในกลางปี ค.ศ. 1993 ทิม เบอร์เนิร์ส-ลีและเดฟ แร็กเกตต์ (Dave Raggett) ได้ออก HTML เวอร์ชัน 1.0 และได้มีการเพิ่มประสิทธิภาพในการนำเสนอข้อมูลบนเว็บมากขึ้น เช่นการเพิ่มแบบฟอร์ม และการส่งจุดภาพของรูปภาพไปที่เครื่องเซิร์ฟเวอร์ (server-side image mapping) ต่อมาได้ถูกพัฒนาโดยแดน คอนนอลลี (Dan Connolly) และได้ออกเป็น HTML เวอร์ชัน 2.0 ในเดือนกันยายน ปี ค.ศ. 1995 ซึ่งก่อนหน้านี้ ในปลายปี ค.ศ. 1993 ดอน คอนนอลลี ได้พัฒนาเวอร์ชันที่เรียกว่า HTML+ แต่ไม่ได้ถูกนำมาใช้อย่างจริงจัง แต่เป็นการก่อร่างขึ้นของเวอร์ชัน 3.0 ซึ่งออกมาในปลายปี ค.ศ. 1995 โดยได้เพิ่มการสนับสนุนการแสดงผลตาราง (table) ตัวอักษรรอบรูปภาพ (text flow around images) และการจัดวางตำแหน่งของกลุ่มข้อมูล (page alignment for block structuring elements) เป็นต้น

ต้นปี ค.ศ. 1997 สมาคมดับเบิ้ลยู-ทีริ-ซี (W3C: World Wide Web Consortium) ผู้ส่งเสริมสนับสนุน และกำหนดมาตรฐานของภาษา HTML ให้มีขีดความสามารถในการแลกเปลี่ยนข้อมูลข่าวสารบนเว็บเป็นไปอย่างกว้างขวาง ได้แนะนำ HTML เวอร์ชัน 3.2 ซึ่งเป็นการรวมประสิทธิภาพของ HTML เวอร์ชัน 2.0 และ 3.0 เข้าด้วยกัน และยังได้เพิ่มลักษณะที่สำคัญคือการสนับสนุน Style Sheets, Frames, Scripting, Math และ Internationalization ต่อมาปลายปี ค.ศ. 1997 สมาคมดับเบิ้ลยู-ทีริ-ซี ก็ได้แนะนำ HTML เวอร์ชัน 4.0 ที่มีการใช้ Unicode เป็นตัวอักษรพื้นฐาน ทำให้มีขีดความสามารถในการแสดงผลโดยใช้ตัวอักษรหลากหลายชนิดมากขึ้น

1.2 แท็กต่าง ๆ ในภาษา HTML

การแสดงผลข้อมูลใด ๆ บนโปรแกรมเว็บเบราว์เซอร์นั้น จำเป็นต้องมีตัวควบคุมการแสดงผล เพื่อทำการสื่อสารกับโปรแกรมเว็บเบราว์เซอร์ว่าการแสดงผลเหล่านั้นแบบไหน เช่น ต้องการแสดงผลตัวอักษรให้เป็นแบบตัวหนา ตัวเอียง ชิดเส้นใต้ หัวข้อสำคัญ รายการ หรือ การแสดงผลตัวอักษรเป็นสีต่าง ๆ ตัวควบคุมการแสดงผลเหล่านี้เรียกว่าแท็ก โดยแท็กต่าง ๆ เหล่านี้จะอยู่ภายในเครื่องหมาย < (less than) และ > (greater than) เช่น แท็กตาราง ซึ่งสามารถเขียนได้คือ <table> เราเรียกแท็กนี้ว่า แท็กเปิด เมื่อมีแท็กเปิดแล้วก็ต้องมีแท็กปิดด้วย ซึ่งแท็กปิดจะแตกต่างจากแท็กเปิดก็คือ จะมีเครื่องหมาย / (slash) นำหน้าชื่อแท็กนั้น ๆ เช่น </table> เป็นต้น แท็กเกือบทั้งหมดในภาษา HTML จำเป็นต้องมีทั้งแท็กเปิดและปิดคู่กัน แต่ก็มีแท็กบางชนิดที่มีเฉพาะแท็กเปิด เช่น <p>,
 และ <hr> เป็นต้น

1.3 รูปแบบโครงสร้างของภาษา HTML

ไฟล์ HTML เป็นเพียงแค่เท็กซ์ไฟล์ธรรมดา มีแท็กหรือเครื่องหมายคำสั่งทำหน้าที่ควบคุมการแสดงผล ซึ่งโดยส่วนใหญ่แล้วจะประกอบไปด้วย 2 ส่วนคือ ส่วนที่เป็นข้อมูลหรือเนื้อหาที่ต้องการแสดงออก เรียกว่า คอนเทนต์ (content) และอีกส่วนหนึ่งเป็นส่วนของตัวควบคุมการแสดงผลของข้อมูลนั้น ๆ เรียกว่า แท็ก (tag)

1.	<html>
2.	<head>
3.	<title>PHP & MySQL</title>
4.	</head>
5.	<body>
6.	...
7.	...
8.	</body>
9.	</html>

รูปที่ 1.1: แท็กที่สำคัญเบื้องต้นใน HTML

การสร้างไฟล์ HTML นั้น ส่วนใหญ่แล้วจะประกอบไปด้วยแท็ก 4 แท็กเป็นอย่างต่ำ ดังแสดงในรูปที่ 1.1 โดยมีแท็กเปิด <html> ในบรรทัดที่ 1 และปิดในบรรทัดที่ 9 เป็นแท็กที่ใช้คลุมแท็กอื่น ๆ ส่วนแท็กที่อยู่ภายในก็จะประกอบไปด้วยแท็กใหญ่ ๆ 2 แท็กคือ แท็ก <head> และแท็ก <body> แท็ก <head> ใช้เป็นส่วนที่ต้องการแสดงในส่วนหัว ซึ่งประกอบไปด้วยแท็ก <title> โดยข้อความที่อยู่ในส่วนของแท็ก <title> นี้จะถูกแสดงออกในส่วนบนของโปรแกรมเว็บเบราว์เซอร์ โดยส่วนใหญ่จะเป็นชื่อของเว็บเพจนั้น ๆ ส่วนข้อมูลต่าง ๆ ที่ต้องการแสดงออกบนหน้าเว็บจะอยู่ในแท็ก <body> ดังตัวอย่างในรูปที่ 1.2

1.	<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
2.	<html>
3.	<head>
4.	<title> A Sample Page </title>
5.	<meta name="generator" content="editplus">
6.	<meta name="author" content="Kacha Chansilp">
7.	<meta name="keywords" content="PHP MySQL Web Application">
8.	<meta name="description" content="This Web site will help you learn
9.	the basics of HTML and Web page design">
10.	<meta http-equiv="content-type" content="text/html; charset=tis-620">
11.	</head>
12.	<body>
13.	<hr width="60%">
14.	<p>This is a sample page.
15.	<hr>
16.	</body>
17.	</html>

รูปที่ 1.2: การใช้แท็กเพื่อแสดงผลในตำแหน่งต่าง ๆ ของเว็บเบราว์เซอร์

บรรทัดที่ 1 เป็นส่วนของการประกาศชนิดของคำสั่ง คุณสมบัติต่าง ๆ และเวอร์ชันของ HTML
บรรทัดที่ 2 เป็นการใส่แท็กเปิด <html> และแท็กปิด </html> ในบรรทัดที่ 17
บรรทัดที่ 3 เป็นการใส่แท็กเปิด <head> และแท็กปิด </head> ในบรรทัดที่ 11
บรรทัดที่ 4 เป็นการใส่แท็กเปิด <title> และแท็กปิด </title> ในบรรทัดเดียวกัน ซึ่งข้อมูลที่อยู่ในแท็กนี้ จะปรากฏอยู่ตรงส่วนบนสุดของโปรแกรมเว็บเบราว์เซอร์

บรรทัดที่ 5 – 10 เป็นการใส่แท็ก <meta> ซึ่งเป็นส่วนที่เครื่องมือค้นหา หรือที่เรียกว่า Search Engine นั้น จะค้นหาข้อมูลบนเว็บโดยดูคุณสมบัติ (Attribute) ที่ผู้เขียนไฟล์ html ใส่ไว้ในส่วนนี้ คุณสมบัติที่สำคัญมีอยู่ 2 คุณสมบัติคือ name และ content ส่วนมากนิยมใส่แท็ก <meta> 3 ครั้งด้วยกัน

คือ ส่วนที่ name มีค่าเท่ากับ author, keywords, และ description (บรรทัดที่ 6, 7 และ 8) และอีก
หนึ่งคุณสมบัติซึ่งเป็นส่วนที่จะสามารถแสดงผลให้เป็นภาษาไทยโดยอัตโนมัติ ในกรณีที่ผู้ใช้อย่างไม่ได้ตั้งค่าและ
คุณสมบัติของโปรแกรมเว็บเบราว์เซอร์ นั่นคือส่วนที่อยู่ในบรรทัดที่ 10

บรรทัดที่ 12 เป็นการใส่แท็กเปิด <body> และแท็กปิด </body> ในบรรทัดที่ 16 ข้อมูลที่อยู่ใน
ในส่วนนี้จะเป็นข้อมูลที่จะถูกแสดงออกที่เว็บ

บรรทัดที่ 13 เป็นการใส่แท็ก <hr> โดยมีการกำหนดคุณสมบัติคือ ความกว้าง ให้มีค่าเท่ากับ
60% นั่นคือการสร้างเส้นให้มีความยาว 60% ของความกว้างของหน้าต่างเว็บเบราว์เซอร์ที่เปิดขึ้นมา
ตำแหน่งของเส้นจะอยู่ที่กึ่งกลางหน้าต่าง ซึ่งเป็นค่าปริยาย (Default)

บรรทัดที่ 14 เป็นการใส่แท็ก <p> ซึ่งเป็นการขึ้นย่อหน้าใหม่แล้วแสดงข้อความว่า “This is a
sample page.”

บรรทัดที่ 15 เป็นการใส่แท็ก <hr> อีกครั้ง แต่ครั้งนี้ไม่มีการกำหนดคุณสมบัติให้ ดังนั้น ค่า
ปริยายที่จะแสดงออกคือ เส้นตรงที่มีความยาวเต็มหน้าต่างเว็บเบราว์เซอร์ ดังแสดงในรูปที่ 1.3



รูปที่ 1.3: การแสดงผลของไฟล์ HTML ในรูปที่ 1.2

1.3.1 แท็กเดี่ยว คือแท็กที่เป็นแท็กเปิดอย่างเดียว ไม่ต้องมีแท็กปิดก็ทำงานได้ เช่น <p>, <hr>, และ
 เป็นต้น

ตารางที่ 1.1: ตัวอย่างแท็กเดี่ยวและการใช้งาน

แท็ก	การใช้งาน
<p>	ใช้เมื่อต้องการขึ้นพารากราฟใหม่
 	ใช้เมื่อต้องการขึ้นบรรทัดใหม่
<hr>	ใช้เมื่อต้องการสร้างเส้นตรง
	ใช้เมื่อต้องการใส่รูปภาพ ซึ่ง xxx คือชื่อไฟล์รูปภาพ

```

1. <!doctype html public "-//w3c//dtd html 4.0 transitional//en">
2. <html>
3.   <head>
4.     <title>Sample of using hr and p tags.</title>
5.     <meta name="author" content="Kacha Chansilp">
6.     <meta name="keywords" content="">
7.     <meta name="description" content="">
8.     <meta http-equiv="content-type" content="text/html; charset=tis-620">
9.   </head>
10.  <body>
11.    <hr>
12.    Sample text 1 followed by hr tag<hr width="80%">
13.    Sample text 2 followed by p tag<p><hr>
14.    Sample text 3 followed by hr tag<hr>
15.  </body>
16. </html>

```

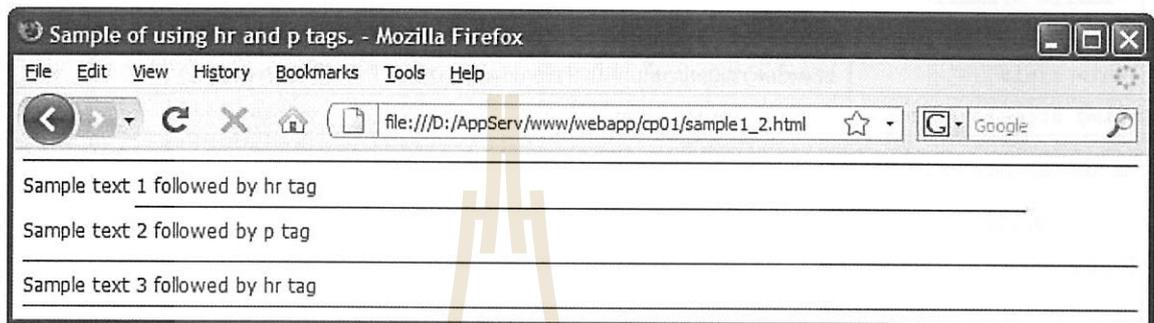
รูปที่ 1.4: การใช้แท็ก <hr> และ <p>

บรรทัดที่ 11 เป็นการใช้แท็ก `<hr>` โดยไม่มีการกำหนดคุณสมบัติให้ ดังนั้นค่าปริยายที่จะแสดงออกคือ เส้นตรงที่มีความยาวเต็มหน้าตาเว็บเบราว์เซอร์

บรรทัดที่ 12 เป็นการแสดงข้อความ "Sample text 1 followed by hr tag" แล้วใช้แท็ก `<hr>` โดยกำหนดคุณสมบัติให้มีความกว้างเท่ากับ 80% ของหน้าเว็บ

บรรทัดที่ 13 เป็นการแสดงข้อความ "Sample text 2 followed by p tag" แล้วใช้แท็ก `<p>` และ `<hr>` โดยไม่มีการกำหนดคุณสมบัติใด ๆ

บรรทัดที่ 14 เป็นการแสดงข้อความ "Sample text 3 followed by hr tag" แล้วใช้แท็ก `<hr>` ผลที่ได้แสดงในรูปที่ 1.5



รูปที่ 1.5: การแสดงผลของไฟล์ HTML ในรูปที่ 1.4

จะสังเกตเห็นว่า เมื่อใดที่มีการใช้แท็ก `<hr>` เส้นตรงจะถูกสร้างขึ้นในบรรทัดใหม่และสิ่งที่ตามมาจะปรากฏอยู่ในบรรทัดต่อไปโดยอัตโนมัติ ซึ่งมีช่องไฟ (ระยะห่างระหว่างบรรทัด) เท่ากับการใช้แท็ก `
` แต่ถ้ามีการใช้แท็ก `<p>` ช่องไฟที่เกิดขึ้นจะมีระยะห่างมากกว่าแท็ก `
` (ดังในรูปที่ 1.5 นั้น จะสังเกตเห็นว่าช่องไฟระหว่างเส้นที่ถูกสร้างขึ้นจากแท็ก `<hr>` เส้นที่ 3 กับข้อความ Sample text 2 followed by p tag จะมากกว่าช่องไฟระหว่างเส้นที่ถูกสร้างขึ้นจากแท็ก `<hr>` เส้นสุดท้ายกับข้อความ Sample text 3 followed by hr tag)

1.3.2 แท็กคู่ คือแท็กที่เป็นแท็กเปิดและปิดคู่กันเช่น ` ... `, `<i> ... </i>` และ `<a> ... ` เป็นต้น ซึ่งในเบื้องต้นนี้จะกล่าวถึงแท็กที่จำเป็นในการแสดงผลที่เห็นกันอยู่โดยทั่ว ๆ ไป ส่วนรายละเอียดและคุณสมบัติต่าง ๆ ของแท็กทั้งหมดจะมีอยู่ในภาคผนวก ข.

1.3.3 แท็กทั่วไป เป็นแท็กที่ใช้ในการแสดงข้อมูลต่าง ๆ ในรูปแบบที่ต้องการ เช่น แสดงตัวอักษรหรือข้อความให้เป็นตัวเอียง ตัวหนา ตัวห้อย ตัวยก หรือแสดงตัวอักษรหรือข้อความโดยให้เลื่อนจากขวาไปซ้าย เป็นต้น

ตารางที่ 1.2: ตัวอย่างแท็กคู่และการใช้งาน

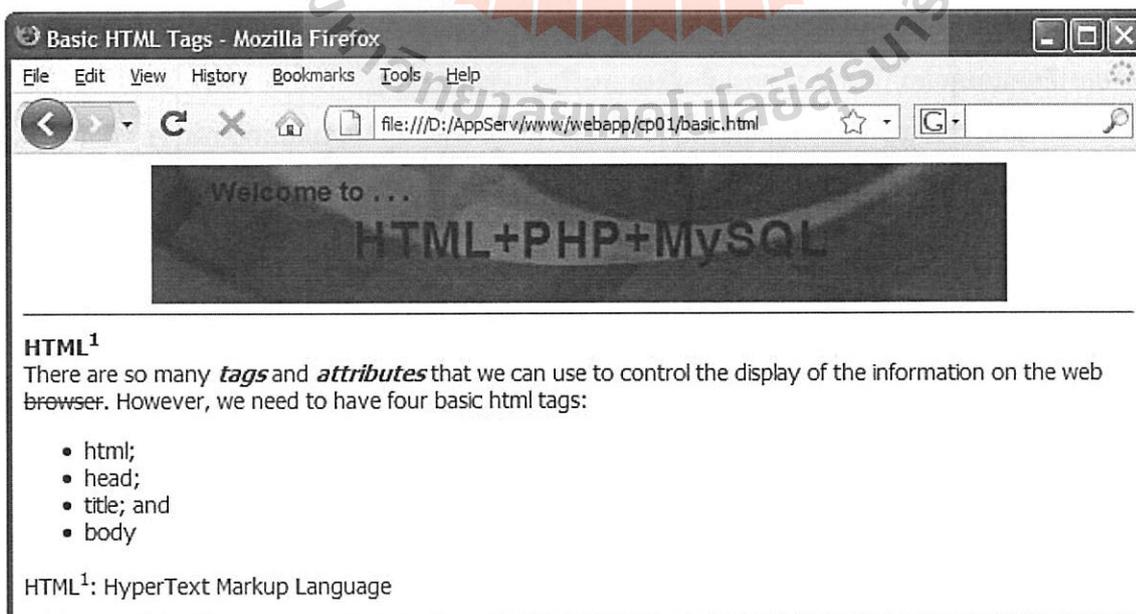
แท็ก (เปิดและปิด)	การใช้งาน
<code> </code>	แสดงข้อความที่อยู่ภายในให้เป็นตัวหนา (Bold)
<code><i> </i></code>	แสดงข้อความที่อยู่ภายในให้เป็นตัวเอียง (Italic)
<code><u> </u></code>	แสดงข้อความที่อยู่ภายในโดยขีดเส้นใต้ (Underline)
<code><sub> </sub></code>	แสดงข้อความที่อยู่ภายในให้เป็นดรรชนีล่างหรือตัวห้อย (Subscript)
<code><sup> </sup></code>	แสดงข้อความที่อยู่ภายในให้เป็นดรรชนีบนหรือตัวยก (Superscript)
<code><strike> </strike></code>	แสดงข้อความที่อยู่ภายในโดยขีดเส้นทับ (Strike Through)
<code> </code>	แสดงรายการที่จะแสดงออกให้เป็นแบบเรียงลำดับ (Ordered List)
<code> </code>	แสดงรายการที่จะแสดงออกให้เป็นแบบไม่เรียงลำดับ (Unordered List)

แท็ก (เปิดและปิด)	การใช้งาน
 	แสดงรายการโดยใช้คู่กับรายการแบบเรียงลำดับและรายการแบบไม่เรียงลำดับ
<a> 	กำหนดจุดเชื่อมต่อ
<center> </center>	แสดงข้อความหรือสิ่งที่อยู่ภายในแท็กนี้ให้อยู่กึ่งกลางของหน้าเว็บ
<div> </div>	ใช้แบ่งรูปแบบการแสดงผล โดยบรรจุแท็กอื่นไว้ภายใน เช่น align, width, height, style เป็นต้น
 	มีการทำงานคล้ายกับแท็ก <div> แต่นิยมใช้กับข้อความสั้น ๆ
<pre> </pre>	แสดงข้อความภายในแท็กนี้บนหน้าเว็บเหมือนกับที่กำลังแสดงอยู่ในโปรแกรมสร้างไฟล์ html
<iframe> </iframe>	แสดงเฟรมแบบลอยตัว
 	แสดงรูปแบบของตัวอักษร
<marquee> </marquee>	แสดงข้อความที่อยู่ภายในให้เคลื่อนที่ (ตัวอักษรวิ่ง)
<small> </small>	แสดงข้อความที่อยู่ภายในให้ขนาดของตัวอักษรเล็กลง
<big> </big>	แสดงข้อความที่อยู่ภายในให้ขนาดของตัวอักษรใหญ่ขึ้น
<h1> </h1>	แสดงข้อความที่อยู่ภายในให้เป็นหัวข้อ (Heading) ระดับที่ 1 (มีทั้งหมด 6 ระดับ)
	กำหนดการแสดงผลรูปภาพชื่อว่า logo.jpg ในหน้าเว็บ

ตัวอย่าง การแสดงผลของข้อมูลในรูปแบบต่าง ๆ

1.	<html>
2.	<head><title> Basic HTML Tags </title>
3.	</head>
4.	<body>
5.	<center></center>
6.	<hr>HTML¹</hr>
7.	There are so many <i>tags</i> and <i>attributes</i> that we can use to
8.	control the display of the information on the web browser. However, we need
9.	to have four basic html tags:
10.	
11.	html;
12.	head;
13.	title; and
14.	body
15.	
16.	HTML¹: HyperText Markup Language
17.	</body>
18.	</html>

รูปที่ 1.6: การใช้แท็กเดี่ยวและแท็กคู่ในการแสดงผลในรูปแบบต่าง ๆ

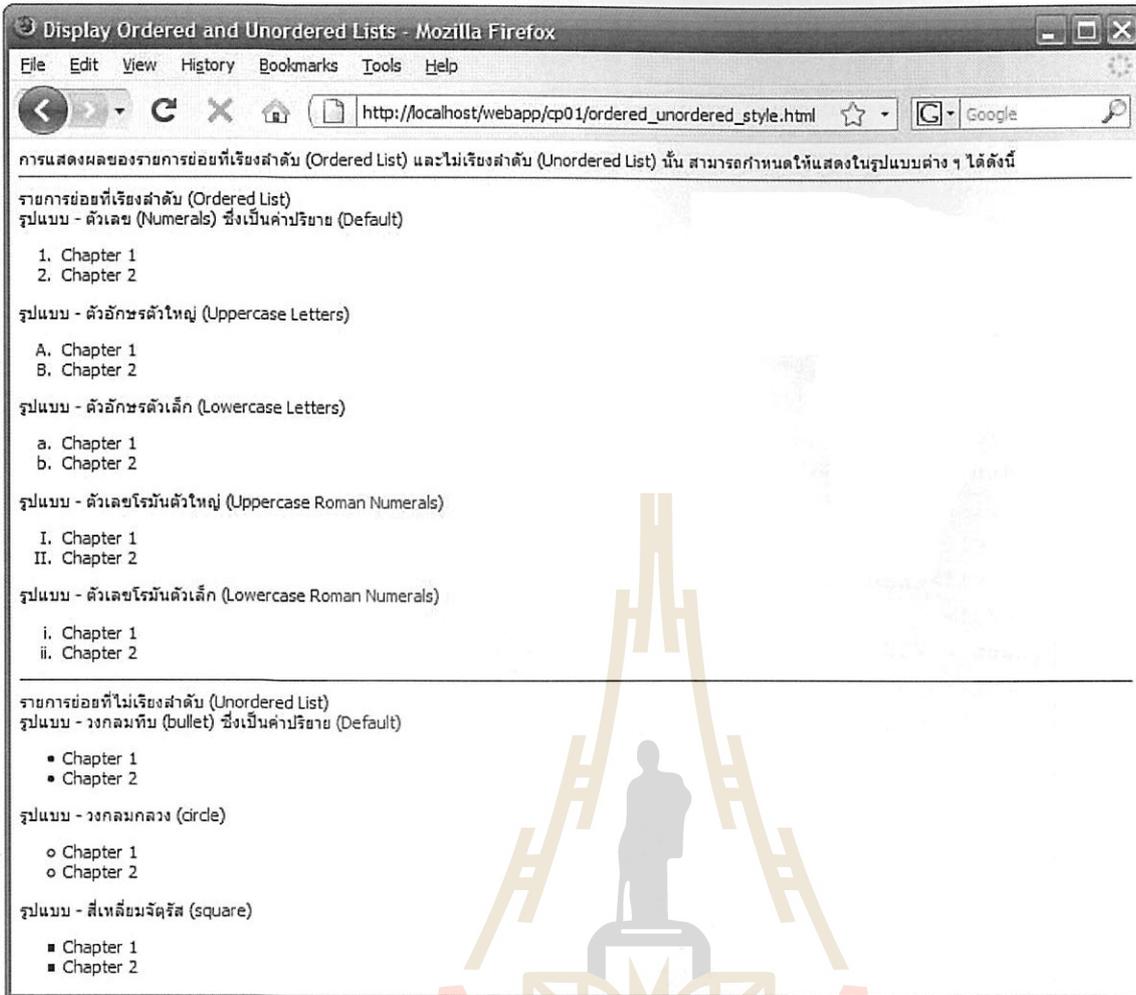


รูปที่ 1.7: การแสดงผลของไฟล์ HTML ในรูปที่ 1.6

ตัวอย่าง การแสดงผลของรายการย่อย (List Items) ในรูปแบบต่าง ๆ

1.	<html>
2.	<head><title>Display Ordered and Unordered Lists</title></head>
3.	<body>การแสดงผลของรายการย่อยที่เรียงลำดับ (Ordered List) และไม่เรียงลำดับ (Unordered List) นั้น สามารถกำหนดให้แสดงในรูปแบบต่าง ๆ ได้ดังนี้ <hr>
4.	รายการย่อยที่เรียงลำดับ (Ordered List)
5.	รูปแบบ - ตัวเลข (Numerals) ซึ่งเป็นค่าปริยาย (Default)
6.	
7.	Chapter 1
8.	Chapter 2
9.	
10.	รูปแบบ - ตัวอักษรตัวใหญ่ (Uppercase Letters)
11.	<ol type="A">
12.	Chapter 1
13.	Chapter 2
14.	
15.	รูปแบบ - ตัวอักษรตัวเล็ก (Lowercase Letters)
16.	<ol type="a">
17.	Chapter 1
18.	Chapter 2
19.	
20.	รูปแบบ - ตัวเลขโรมันตัวใหญ่ (Uppercase Roman Numerals)
21.	<ol type="I">
22.	Chapter 1
23.	Chapter 2
24.	
25.	รูปแบบ - ตัวเลขโรมันตัวเล็ก (Lowercase Roman Numerals)
26.	<ol type="i">
27.	Chapter 1
28.	Chapter 2
29.	<hr>
30.	รายการย่อยที่ไม่เรียงลำดับ (Unordered List)
31.	รูปแบบ - วงกลมทึบ (bullet) ซึ่งเป็นค่าปริยาย (Default)
32.	<ul type="disc">
33.	Chapter 1
34.	Chapter 2
35.	
36.	รูปแบบ - วงกลมกลวง (circle)
37.	<ul type = "circle">
38.	Chapter 1
39.	Chapter 2
40.	
41.	รูปแบบ - สี่เหลี่ยมจัตุรัส (square)
42.	<ul type = "square">
43.	Chapter 1
44.	Chapter 2
45.	
46.	</body></html>

รูปที่ 1.8: การแสดงผลของรายการย่อยในรูปแบบต่าง ๆ



รูปที่ 1.9: การแสดงผลของไฟล์ html ในรูปที่ 1.8

1.3.4 **แท็กตาราง** เป็นแท็กที่ใช้ในการสร้างตารางการนำเสนอข้อมูลต่าง ๆ โดยใช้แท็กเปิด `<table>` และแท็กปิด `</table>` เป็นตัวกำหนดการสร้างตาราง ซึ่งมีแท็กที่ใช้ในการกำหนดจำนวนแถว และจำนวนคอลัมน์ดังในตารางดังต่อไปนี้

ตารางที่ 1.3: ตัวอย่างแท็กที่ใช้ในการกำหนดตาราง แถว และคอลัมน์

แท็ก (เปิดและปิด)	การใช้งาน
<code><table></code> <code></table></code>	กำหนดการสร้างตาราง (Table)
<code><th></code> <code></th></code>	กำหนดหัวข้อในตาราง (Table Heading)
<code><tr></code> <code></tr></code>	กำหนดแถวในตาราง (Table Row)
<code><td></code> <code></td></code>	แสดงข้อมูลในตาราง (Table Data)
<code><caption></code> <code></caption></code>	แสดงชื่อของตาราง

ตัวอย่าง การสร้างตารางเพื่อแสดงข้อมูลในรูปแบบต่าง ๆ

```

1. <html>
2.   <head>
3.     <title> Sample: Table</title>
4.   </head>
5.   <body>
6.     <center>
7.       <table border="1" width="60%">
8.         <caption align="bottom">Table 1: Simple Table</caption>
9.         <tr bgcolor="yellow"><th>Column 1</th><th>Column 2</th><th>Column 3</th>
10.        <th>Column 4</th></tr>
11.        <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
12.        <tr><td>one</td><td>two</td><td>three</td><td>four</td></tr>
13.        <tr><td bgcolor="pink" colspan="2">one + two</td>
14.          <td>three</td><td colspan="2" align="center">four</td></tr>
15.        <tr><td>first</td><td>second</td><td>third</td></tr>
16.      </table>
17.    </center>
18.  </body>
19. </html>

```

รูปที่ 1.10: การใช้แท็กในการสร้างตาราง

บรรทัดที่ 6 เป็นการใส่แท็กเปิด `<center>` และแท็กปิด `</center>` ในบรรทัดที่ 17 นั้นหมายถึงสิ่งที่อยู่ระหว่างแท็กคู่นี้จะถูกจัดอยู่ตรงกึ่งกลางของหน้าเว็บ ซึ่งก็คือตารางนั่นเอง

บรรทัดที่ 7 เป็นการใส่แท็กเปิด `<table>` ในการสร้างตารางและแท็กปิด `</table>` ในบรรทัดที่ 16 โดยในบรรทัดที่ 7 นั้นมีการกำหนดคุณสมบัติให้กับตารางคือ

`border="1"` คือการกำหนดให้ขอบของตารางมีขนาด 1 พิกเซล

`width="60%"` คือการกำหนดให้ตารางมีความกว้าง 60% ของหน้าต่างเว็บ

บรรทัดที่ 8 เป็นการใส่แท็กเปิด `<caption>` และแท็กปิด `</caption>` ในการกำหนดชื่อของตาราง โดยมีการกำหนดคุณสมบัติให้กับชื่อของตารางนี้คือ

`align="bottom"` คือการกำหนดให้ชื่อตารางอยู่ใต้ตาราง โดยคำปரியายของแท็กนี้ก็คือข้อความที่อยู่ในแท็กนี้จะอยู่ด้านล่างและกึ่งกลางของตาราง

บรรทัดที่ 9 เป็นการใส่แท็กเปิด `<tr>` หรือเรียกอีกอย่างหนึ่งว่า table row นั้นคือการกำหนดแถวในตาราง โดยมีการกำหนดคุณสมบัติให้กับแถวนี้คือ

`bgcolor="yellow"` คือการกำหนดให้สีพื้นหลังของแถวนี้เป็นสีเหลือง และต่อจากนั้นมีการใส่แท็กเปิด `<th>` และแท็กปิด `</th>` ย่อมาจาก table heading ทั้งหมด 4 คู่ นั่นคือการกำหนดหัวข้อทั้งหมด 4 หัวข้อหรือ 4 คอลัมน์นั่นเองแล้วปิดท้ายด้วยแท็กปิด `</tr>` โดยคำปரியายของการใช้แท็ก `<th>` นี้ก็คือ ข้อความที่อยู่ในแท็กชนิดนี้จะเป็นตัวหนาและอยู่กึ่งกลางของคอลัมน์นั้น ๆ

บรรทัดที่ 11 เป็นการใส่แท็กเปิด `<tr>` แล้วตามด้วยแท็กเปิด `<td>` และแท็กปิด `</td>` ย่อมาจาก table data อีก 4 คู่ ในแต่ละคู่นั้นจะมีเลข 1, 2, 3, และ 4 ตามลำดับ แล้วปิดท้ายด้วยแท็กปิด `</tr>` จะสังเกตเห็นว่าเลขเหล่านั้นจะถูกแสดงออกเป็นสีดำและอยู่ชิดซ้ายของคอลัมน์นั้น ๆ นี่คือการปரியาย

บรรทัดที่ 12 ก็เป็นลักษณะเช่นเดียวกันกับบรรทัดที่ 11

บรรทัดที่ 13-14 เป็นการใส่แท็กเปิด `<tr>` แล้วตามด้วยแท็กเปิด `<td>` และแท็กปิด `</td>` อีก 3 คู่ โดยคู่แรกของแท็ก `<td>` นั้นมีการกำหนดคุณสมบัติให้กับข้อมูลที่อยู่ในแถวและคอลัมน์นี้คือ

`bgcolor="pink"` คือการกำหนดสีพื้นหลังของแถวนี้เป็นสีชมพู

`colspan="2"` คือการกำหนดให้ยึดคอลัมน์นั้นให้มีความกว้างเท่ากับ 2 คอลัมน์ แล้วรวมเป็นคอลัมน์เดียวกัน นั่นคือการรวมคอลัมน์ที่ 1 และ 2 ของแถวที่ 4 เข้าด้วยกันนั่นเอง ส่วนคู่ที่สองนั้นเป็นการแสดงข้อมูลที่อยู่ในคอลัมน์นั้นโดยไม่มีการกำหนดคุณสมบัติใด ๆ ส่วนคู่ที่ 3 นั้นมีการกำหนดคุณสมบัติให้กับข้อมูลที่อยู่ในแถวและคอลัมน์นี้คือ

`rowspan="2"` คือการกำหนดให้ยึดแถวนั้นให้มีความสูงเท่ากับ 2 แถว แล้วรวมเป็นแถวเดียวกัน นั่นคือการรวมแถวที่ 4 และ 5 ในคอลัมน์ที่ 4 เข้าด้วยกันนั่นเอง

`align="center"` คือการกำหนดให้แสดงข้อความตรงกึ่งกลางของคอลัมน์

บรรทัดที่ 15 ก็เป็นลักษณะเช่นเดียวกันกับบรรทัดที่ 12 แต่มีการใช้แท็ก `<td>` แค่ 3 คู่ เพราะว่าบรรทัดที่ 14 มีการรวมแถว (แถวที่ 5 คอลัมน์ที่ 4) ไปแล้ว ผลที่ได้แสดงดังในรูปที่ 1.11

Column 1	Column 2	Column 3	Column 4
1	2	3	4
one	two	three	four
one + two		three	four
first	second	third	

Table 1: Simple Table

รูปที่ 1.11: การแสดงผลของไฟล์ HTML ในรูปที่ 1.10

การแสดงผลในรูปแบบของตารางนั้น ยังสามารถกำหนดคุณสมบัติต่าง ๆ ได้ดังต่อไปนี้ คือ

- คุณสมบัติ **frame** ซึ่งสามารถกำหนดการแสดงผลของเส้นขอบในตาราง ได้ดังนี้
 - `void`: ไม่แสดงเส้นขอบ
 - `above`: แสดงเส้นขอบที่อยู่ด้านบนสุดเท่านั้น
 - `below`: แสดงเส้นขอบที่อยู่ด้านล่างสุดเท่านั้น
 - `hsides`: แสดงเส้นขอบที่อยู่ด้านบนและล่างสุดเท่านั้น
 - `vsides`: แสดงเส้นขอบที่อยู่ด้านข้างเท่านั้น
 - `lhs`: แสดงเส้นขอบที่อยู่ด้านซ้ายเท่านั้น
 - `rhs`: แสดงเส้นขอบที่อยู่ด้านขวาเท่านั้น
 - `box`: แสดงเส้นขอบทั้ง 4 ด้าน
 - `border`: แสดงเส้นขอบทั้ง 4 ด้าน
- คุณสมบัติ **rules** ซึ่งสามารถกำหนดการแสดงผลของเส้นขอบในตาราง ได้ดังนี้
 - `none`: ไม่กำหนดกฎ (ค่าปริยาย)
 - `groups`: แสดงเฉพาะกลุ่มของแถวและคอลัมน์เท่านั้น
 - `rows`: แสดงเฉพาะข้อมูลที่อยู่ระหว่างแถวเท่านั้น
 - `cols`: แสดงเฉพาะข้อมูลที่อยู่ระหว่างคอลัมน์เท่านั้น

ที่อยู่ในส่วนนี้จะจะเป็น CGI Script (Common Gateway Interface Script) นั่นคือการส่งแบบมาตรฐานของข้อมูลระหว่างเครื่องเซิร์ฟเวอร์กับ Script เช่น ASP, Perl, หรือ PHP เป็นต้น โดยสคริปต์จะทำการรันที่เครื่องเซิร์ฟเวอร์ ดังนั้น ในการทดสอบหรือรันสคริปต์ต่าง ๆ เหล่านี้จึงจำเป็นต้องมีเครื่องเซิร์ฟเวอร์ แต่ในบางครั้งอาจเรียกไฟล์สกุลอื่นได้ เช่น ไฟล์สกุล html เป็นต้น ทั้งนี้ขึ้นอยู่กับความต้องการของผู้ออกแบบ

method="post" คือการส่งข้อมูลในแบบฟอร์มไปให้กับไฟล์ process.php แบบ post (รายละเอียดของการส่งข้อมูลแบบ post และ get จะกล่าวถึงในตอนท้าย)

บรรทัดที่ 5 เป็นการใช้แท็กเปิด <fieldset> ในการจัดกลุ่มรูปแบบฟอร์มและแท็กปิด </fieldset> ในบรรทัดที่ 26 โดยในบรรทัดที่ 5 นั้นมีการใช้แท็กเปิด <legend><i> ตามด้วยข้อความว่า "Personal Information" และแท็กปิด </i></legend> จะสังเกตเห็นว่าเมื่อมีการใช้แท็กเปิดที่ซ้อนกันแล้ว เมื่อต้องการปิดแท็กเหล่านั้น จะต้องปิดแท็กที่เปิดตัวสุดท้ายของแท็กเปิดทั้งหมดก่อน ในกรณีนี้แท็กที่เปิดตัวสุดท้ายคือแท็กเปิด <i> นั่นเอง

บรรทัดที่ 6 มีข้อความว่า name: แล้วตามด้วยแท็ก <input> ซึ่งมีการกำหนดคุณสมบัติให้ดังนี้

type="text" คือการกำหนดให้ชนิดข้อมูลเข้าเป็นแบบข้อความ

name="name" คือการกำหนดตัวแปรชื่อว่า name เพื่อเก็บข้อมูลเข้า

size="30" คือการกำหนดขนาดจำนวนของข้อมูลเข้ามีความยาว 30 ตัวอักษร

แล้วหลังจากนั้นมีการใช้คำสั่ง (new blank space) สองตัว ก็คือกำหนดช่องว่าง 2 ช่อง เพราะในภาษา html นั้น ถ้ามีช่องว่างระหว่างข้อความมากกว่า 1 ช่องจะแสดงช่องว่างเพียง 1 ช่องเท่านั้น ถ้าต้องการให้มีช่องว่างมากกว่านั้น ต้องใช้คำสั่ง ช่วยในการจัดรูปแบบ

บรรทัดที่ 7 มีข้อความว่า E-mail: แล้วตามด้วยแท็ก <input> ซึ่งมีการกำหนดคุณสมบัติให้ดังนี้

type="text" คือการกำหนดให้ชนิดข้อมูลเข้าเป็นแบบข้อความ

name="email" คือการกำหนดตัวแปรชื่อว่า email เพื่อเก็บข้อมูลเข้า

size="30" คือการกำหนดขนาดจำนวนของข้อมูลเข้ามีความยาว 30 ตัวอักษรแล้ว

หลังจากนั้นมีการแท็ก
 เพื่อขึ้นบรรทัดใหม่

บรรทัดที่ 8 มีข้อความว่า Age: แล้วตามด้วยแท็ก <input> ซึ่งมีการกำหนดคุณสมบัติให้ดังนี้

type="radio" คือการกำหนดให้ชนิดข้อมูลเข้าเป็นตัวเลือกแบบปุ่มข้อความ

name="age" คือการกำหนดตัวแปรชื่อว่า age เพื่อเก็บข้อมูลเข้า

value="1" คือการกำหนดค่าให้กับตัวแปร age

บรรทัดที่ 9-11 เป็นตัวเลือกแบบปุ่มตัวเลือกเช่นเดียวกับบรรทัดที่ 8 จะสังเกตเห็นว่าค่าของ name ทั้ง 4 บรรทัดนั้นมีค่าเช่นเดียวกันคือ age ส่วนค่าของ value จะเปลี่ยนไป ตัวเลือกแบบปุ่มตัวเลือกนี้สามารถเลือกได้เพียง 1 ปุ่มตัวเลือกเท่านั้น หลังจากนั้นมีการใช้แท็ก
 เพื่อขึ้นบรรทัดใหม่

บรรทัดที่ 12 มีข้อความว่า What language can you speak?

บรรทัดที่ 13 เป็นการใช้แท็ก <input> ซึ่งมีการกำหนดคุณสมบัติให้ดังนี้

`type="checkbox"` คือการกำหนดให้ชนิดข้อมูลเข้าเป็นตัวเลือกแบบกล่องตัวเลือก ซึ่งผู้ใช้สามารถเลือกได้มากกว่า 1 กล่องตัวเลือก โดยค่าของกล่องตัวเลือกที่ผู้ใช้เลือกทั้งหมดจะถูกส่งไปยังไฟล์ที่ถูกกำหนดไว้ให้กับ `action` ในแท็กเปิด `<form>` ซึ่งในที่นี้ก็คือไฟล์ `process.php` นั่นเอง

`name="language"` คือการกำหนดตัวแปรชื่อว่า `language` เพื่อเก็บข้อมูลเข้า

`value="Chinese"` คือการกำหนดค่าให้เป็น `Chinese` แล้วเก็บไว้ที่ตัวแปร `language` เมื่อผู้ใช้เลือกกล่องตัวเลือกนี้

บรรทัดที่ 14-17 เป็นตัวเลือกแบบกล่องตัวเลือก เช่นเดียวกับบรรทัดที่ 13

บรรทัดที่ 18 มีข้อความว่า `Please select your most favourite food:`

บรรทัดที่ 19 เป็นการใช้แท็กเปิด `<select>` เพื่อกำหนดตัวเลือกแบบเมนู และแท็กปิด `</select>` ในบรรทัดที่ 25 โดยในบรรทัดที่ 19 นั้นมีการกำหนดคุณสมบัติให้ดังนี้

`name="food"` คือการกำหนดตัวแปรชื่อว่า `food` เพื่อเก็บข้อมูลเข้า

บรรทัดที่ 20 มีการกำหนดคุณสมบัติให้ดังนี้

`<option value="American">American` ค่าที่อยู่ในเครื่องหมายคำพูด ที่ถูกกำหนดให้กับ `value` คือ `American` ซึ่งถ้าผู้ใช้เลือกตัวเลือกนี้ `American` จะเป็นข้อมูลเข้าและถูกเก็บไว้ที่ตัวแปร `food`

บรรทัดที่ 21-24 เป็นการกำหนดตัวเลือกที่มีลักษณะเดียวกับบรรทัดที่ 22

บรรทัดที่ 27 เป็นการใช้แท็กเปิด `<fieldset>` ในการจัดกลุ่มรูปแบบฟอร์มและแท็กปิด `</fieldset>` ในบรรทัดที่ 30 โดยในบรรทัดที่ 27 นั้นมีการใช้แท็กเปิด `<legend align="right"><i>` ตามด้วยข้อความว่า `"Comments"` และแท็กปิด `</i></legend>` ดังนั้น ข้อความ `Comments` จะถูกแสดงออกเป็นตัวเอียงและหนาในตำแหน่งขีดขวาของหน้าเว็บ

บรรทัดที่ 28 มีข้อความว่า `Please give us your comments:` แล้วตามด้วยแท็ก `
` เพื่อขึ้นบรรทัดใหม่

บรรทัดที่ 29 เป็นการใช้แท็กเปิด `<textarea>` และแท็กปิด `</textarea>` โดยมีการกำหนดคุณสมบัติให้ดังนี้

`name="comments"` คือการกำหนดตัวแปรชื่อว่า `comments`

`cols="50"` คือการกำหนดจำนวนคอลัมน์ให้มีค่าเท่ากับ 50 หรือ 50 ตัวอักษรนั่นเอง

`rows="5"` คือการกำหนดจำนวนแถวให้มีค่าเท่ากับ 5

บรรทัดที่ 31 เป็นการใช้แท็กเปิด `<center>` และแท็กปิด `</center>` ในบรรทัดที่ 34 นั้นหมายถึงสิ่งที่อยู่ระหว่างแท็กนี้ คือบรรทัดที่ 32 และ 33 จะถูกจัดอยู่ตรงกึ่งกลางของหน้าเว็บ

บรรทัดที่ 32 เป็นการใช้แท็ก `<input>` โดยมีการกำหนดคุณสมบัติให้ดังนี้

`type="submit"` คือการกำหนดชนิดข้อมูลให้เป็นแบบปุ่ม `submit` เพื่อส่งข้อมูลทั้งหมดที่ถูกเก็บไว้ในตัวแปรต่าง ๆ ไปที่เครื่องเซิร์ฟเวอร์

`name="submit"` คือการกำหนดตัวแปรชื่อว่า `submit` ซึ่งถ้ามีการใช้ปุ่มแบบ `submit` นี้เพียงปุ่มเดียว การกำหนดตัวแปรก็ไม่จำเป็นต้องมีก็ได้

`value="Submit Information"` คือการกำหนดค่าที่จะแสดงออกบนปุ่ม

บรรทัดที่ 33 เป็นการใช้แท็ก `<input>` โดยมีการกำหนดคุณสมบัติให้ดังนี้

`type="reset"` คือการกำหนดชนิดข้อมูลให้เป็นแบบปุ่ม reset เพื่อทำการลบข้อมูลทั้งหมดที่ผู้ใช้ป้อนเข้ามา

`name="reset"` คือการกำหนดตัวแปรชื่อว่า reset

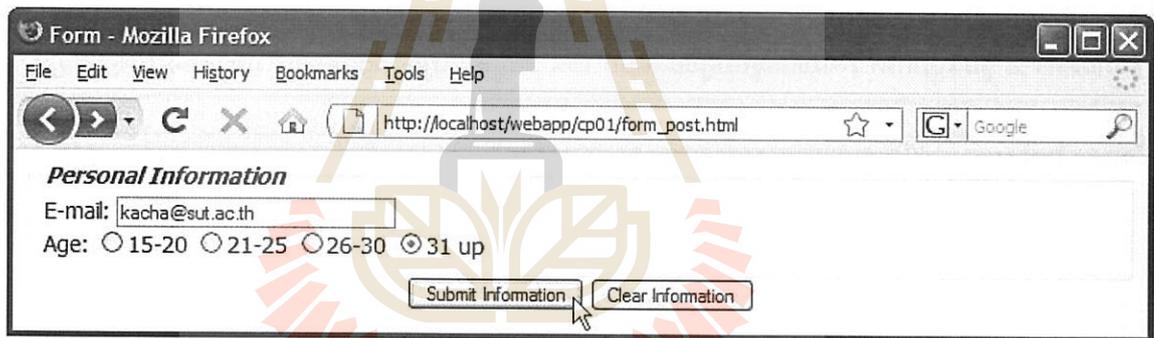
`value="Clear Information"` คือการกำหนดค่าที่จะแสดงออกบนปุ่ม

จากการกำหนดแบบฟอร์มที่ผ่านมา จะสังเกตเห็นว่าการกำหนดคุณสมบัติให้กับแท็กเปิด `<form>` นั้นจำเป็นต้องมีการกำหนดคุณสมบัติอย่างน้อย 2 คุณสมบัติคือ action และ method ซึ่งการกำหนดค่าให้กับ action นั้นเป็นการบอกถึงแหล่งปลายทางหรือไฟล์ที่ข้อมูลที่ถูกเก็บอยู่ในตัวแปรต่าง ๆ จะถูกส่งไปถึง ส่วนการกำหนดค่าให้กับ method นั้นจะเป็นการบ่งบอกถึงลักษณะของการส่งข้อมูลเข้าไปยังไฟล์ที่กำหนดใน action มีอยู่ 2 ลักษณะคือ post และ get ซึ่งมีลักษณะแตกต่างกันดังนี้

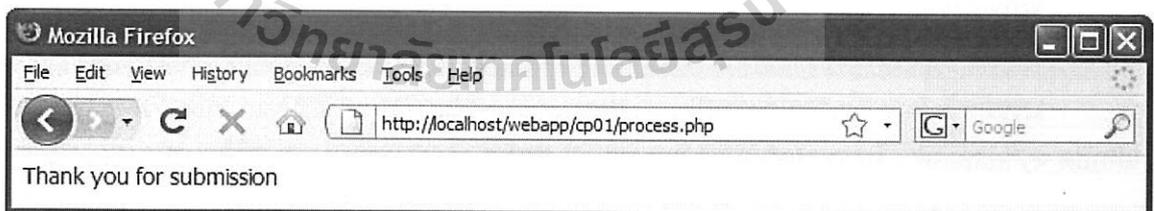
post คือการส่งข้อมูลไปยังแฟ้มปลายทางโดยซ่อนค่าต่าง ๆ ที่ส่งไป

get คือการส่งข้อมูลไปยังแฟ้มปลายทางโดยแสดงค่าที่ส่งไปในช่อง URL

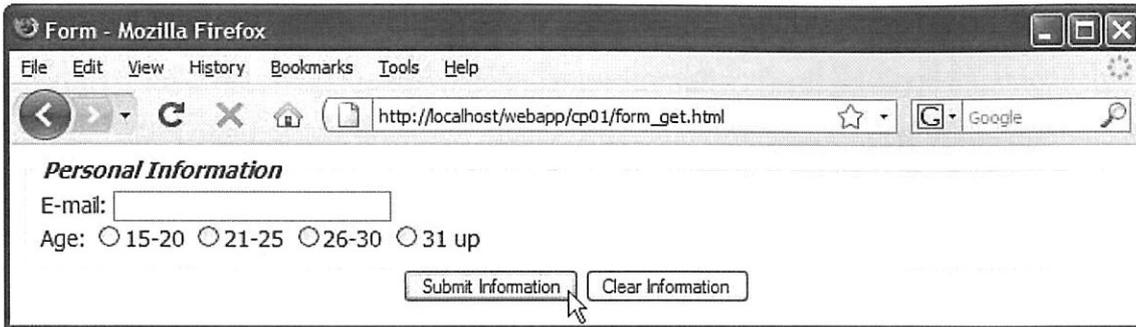
การส่งข้อมูลไปยังแฟ้มปลายทางแบบ post เป็นแบบที่นิยมมากที่สุด เพราะว่าการส่งข้อมูลแบบ post นี้จะไม่แสดงข้อมูลในช่องที่แสดง URL ดังแสดงในรูปที่ 1.16 ถึง 1.19 เป็นการแสดงให้เห็นถึงข้อแตกต่างระหว่างการซ่อนและแสดงข้อมูลในช่อง URL



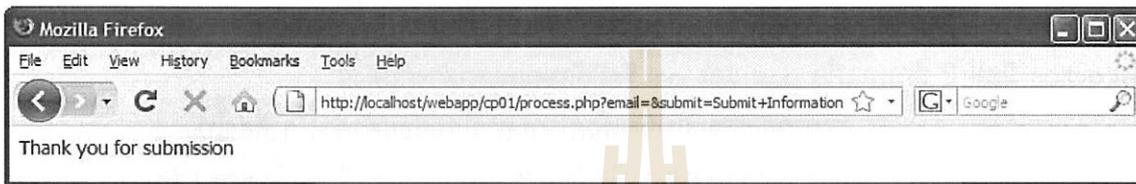
รูปที่ 1.16: แบบฟอร์มการส่งข้อมูลแบบ post



รูปที่ 1.17: การซ่อนข้อมูลในช่อง URL เมื่อส่งข้อมูลแบบ post



รูปที่ 1.18: แบบฟอร์มการส่งข้อมูลแบบ get



รูปที่ 1.19: การแสดงข้อมูลในช่อง URL เมื่อส่งข้อมูลแบบ get

1.3.6 แท็กเฟรม เป็นแท็กที่ใช้ในการกำหนดรูปแบบของการนำเสนอข้อมูลต่าง ๆ โดยแบ่งเป็นเฟรม ใช้แท็กเปิด `<frameset>` และแท็กปิด `</frameset>` เป็นตัวกำหนดการแบ่งเฟรม แล้วกำหนดคุณสมบัติที่ต้องการแบ่ง เช่น แบ่งเป็นคอลัมน์หรือแบ่งเป็นแถว เป็นต้น ซึ่งการกำหนดคุณสมบัตินี้สามารถกำหนดทั้ง 2 รูปแบบก็ได้ ทั้งนี้ขึ้นอยู่กับผู้ออกแบบ เช่น ข้อมูลในไฟล์ `index.html` มีดังต่อไปนี้

1.	<code><html></code>
2.	<code><head></code>
3.	<code><title>Frame Control Web Page</title></code>
4.	<code></head></code>
5.	<code><frameset cols="150,*"></code>
6.	<code><frame name="link" src="link.html"></code>
7.	<code><frame name="display" src="welcome.html"></code>
8.	<code></frameset></code>
9.	<code></html></code>

รูปที่ 1.20: การใช้แท็กในการสร้างเฟรม

จะสังเกตเห็นว่าเมื่อใช้แท็ก `<frameset>` แล้วจะไม่มีแท็ก `<body>` เพราะว่าเป็นขั้นตอนของการกำหนดรูปแบบของการแบ่งเฟรม

บรรทัดที่ 5 เป็นการใส่แท็กเปิด `<frameset>` ในการกำหนดรูปแบบของการแบ่งเฟรมและแท็กปิด `</frameset>` ในบรรทัดที่ 8 โดยในบรรทัดที่ 5 นั้นมีการกำหนดคุณสมบัติให้กับเฟรมคือ

`cols="150,*"` คือการแบ่งเฟรมออกเป็น 2 คอลัมน์ โดยคอลัมน์ที่หนึ่งมีความกว้างเท่ากับ 150 พิกเซล ส่วนคอลัมน์ที่สองนั้นมีความกว้างเท่ากับจำนวนพิกเซลที่เหลือทั้งหมดของหน้าโปรแกรมเว็บเบราว์เซอร์ เมื่อมีการแบ่งเฟรมออกเป็น 2 คอลัมน์แล้ว จึงมีการใช้แท็ก `<frame>` 2 ครั้ง (ในบรรทัดที่ 6 และ 7)

บรรทัดที่ 6 เป็นการใส่แท็ก `<frame>` โดยมีการกำหนดคุณสมบัติให้กับเฟรมแรกหรือคอลัมน์ที่หนึ่งดังนี้

`name="link"` คือการตั้งชื่อให้กับคอลัมน์นี้ชื่อว่า link

`src="link.html"` คือการนำไฟล์ link.html มาแสดงในคอลัมน์นี้

บรรทัดที่ 7 เป็นการใช้แท็ก `<frame>` อีกครั้งโดยมีการกำหนดคุณสมบัติให้กับเฟรมที่สองหรือคอลัมน์ที่สองดังนี้

`name="display"` คือการตั้งชื่อให้กับคอลัมน์นี้ชื่อว่า display

`src="welcome.html"` คือการนำไฟล์ welcome.html มาแสดงในคอลัมน์นี้

สมมุติว่าไฟล์ link.html มีรายละเอียดดังในรูปที่ 1.21 และแสดงออกที่หน้าเว็บดังในรูปที่ 1.22

```

1. <html>
2.   <head>
3.     <title>Link</title>
4.   </head>
5.   <body>
6.     <a href="home.html" target="display">home</a><br>
7.     <a href="course.html" target="display">course</a><br>
8.     <a href="download.html" target="display">download</a><br>
9.   </body>
10. </html>
    
```

รูปที่ 1.21: การใช้แท็กในไฟล์ link.html

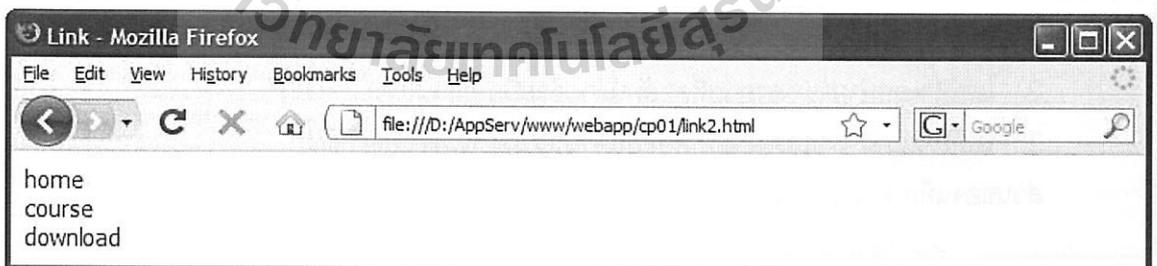


รูปที่ 1.22: การแสดงผลของไฟล์ link.html

โดยปกติแล้ว ค่าปริยายของตัวอักษรที่เป็นจุดเชื่อมต่อจะเป็นสีน้ำเงินและขีดเส้นใต้ เราสามารถกำหนดรูปแบบให้จุดเชื่อมต่อไม่มีเส้นใต้ได้โดยแทรกคำสั่งต่อไปนี้ก่อนแท็ก `<body>` และจะปรากฏผลดังในรูปที่ 1.23

```

<style type="text/css">
  <!-- a {text-decoration:none} -->
</style>
    
```



รูปที่ 1.23: การแสดงผลของจุดเชื่อมต่อแบบไม่มีเส้นใต้

สมมุติว่าไฟล์ welcome.html มีรายละเอียดดังในรูปที่ 1.24 และแสดงออกที่หน้าเว็บดังในรูปที่ 1.25

```

1. <html>
2.   <head>
3.     <title>Welcome</title>
4.   </head>
5.   <body bgcolor="#99FF99">Welcome to <b>HTML - PHP - MySQL</b>
6. </body>
7. </html>
    
```

รูปที่ 1.24: การใช้แท็กในไฟล์ welcome.html



รูปที่ 1.25: การแสดงผลของไฟล์ welcome.html

ดังนั้น การแสดงข้อมูลของไฟล์ index.html ซึ่งเป็นไฟล์ที่กำหนดรูปแบบของเฟรม จะปรากฏดังในรูปที่ 1.26



รูปที่ 1.26: การแสดงข้อมูลของไฟล์ index.html

จะสังเกตเห็นว่า การกำหนดรูปแบบของเฟรมที่ประกอบไปด้วยคอลัมน์ 2 คอลัมน์นั้น ประกอบไปด้วยไฟล์ 3 ไฟล์ด้วยกันคือ

1. ไฟล์ที่กำหนดรูปแบบของเฟรม (index.html)
2. ไฟล์ที่จะถูกนำมาแสดงออกที่คอลัมน์ด้านซ้ายมือ (link.html)
3. ไฟล์ที่จะถูกนำมาแสดงออกที่คอลัมน์ด้านขวามือ (welcome.html)

ส่วนของแท็ก body ในไฟล์ link.html นั้น มีการใช้แท็ก a เพื่อเชื่อมต่อไปยังไฟล์ต่าง ๆ ดังนี้

```

<a href="home.html" target="display">home</a><br>
<a href="course.html" target="display">course</a><br>
<a href="download.html" target="display">download</a><br>
    
```

เมื่อผู้ใช้เลือกจุดเชื่อมต่อ home, course หรือ download (ตำแหน่งตัวหนังสือสีน้ำเงินที่ขีดเส้นใต้ในเฟรมซ้ายมือของรูปที่ 1.26) ไฟล์ home.html, course.html หรือ download.html จะปรากฏในเฟรมด้านขวามือตามจุดเชื่อมต่อที่ผู้ใช้เลือก เพราะมีการกำหนดคุณสมบัติคือ target="display" ซึ่งเฟรมทางด้านขวามือมีชื่อว่า display

ตารางต่อไปนี้จะแสดงการใช้แท็ก frameset โดยกำหนดคุณสมบัติทั้งแบบคอลัมน์และแบบแถว

ตารางที่ 1.5: ตัวอย่างแท็กที่ใช้ในการสร้างเฟรม

แท็ก	ความหมาย
<code><frameset cols="20%,*"></code>	แบ่งเฟรมออกเป็น 2 คอลัมน์ โดยคอลัมน์ที่หนึ่งมีความกว้าง 20 เปอร์เซ็นต์ของหน้าเว็บ ส่วนคอลัมน์ที่สองนั้นมีความกว้างเท่ากับจำนวนเปอร์เซ็นต์ที่เหลือทั้งหมดของหน้าเว็บ ซึ่งก็คือ 80 เปอร์เซ็นต์นั่นเอง
<code><frameset cols="10%,*,20%"></code>	แบ่งเฟรมออกเป็น 3 คอลัมน์ โดยคอลัมน์ที่หนึ่งมีความกว้าง 10 เปอร์เซ็นต์ของหน้าเว็บ คอลัมน์ที่สองนั้นมีความกว้างเท่ากับจำนวนเปอร์เซ็นต์ที่เหลือทั้งหมดจากคอลัมน์ที่หนึ่งและสามรวมกัน (70 เปอร์เซ็นต์) ส่วนคอลัมน์ที่สามนั้นมีความกว้าง 20 เปอร์เซ็นต์ของหน้าเว็บ
<code><frameset rows="10%,*"></code>	แบ่งเฟรมออกเป็น 2 แถว โดยแถวที่หนึ่งมีความสูง 10 เปอร์เซ็นต์ของหน้าเว็บ ส่วนแถวที่สองนั้นมีความสูงเท่ากับจำนวนเปอร์เซ็นต์ที่เหลือทั้งหมดของหน้าเว็บ ซึ่งก็คือ 90 เปอร์เซ็นต์นั่นเอง
<pre> <frameset rows="10%,*" <frame name="logo" src="logo.html"> <frameset cols="20%,*" <frame name="link" src="link.html"> <frame name="display" src= "welcome.html"> </frameset> </frameset> </pre>	แบ่งเฟรมออกเป็น 2 แถว โดยแถวที่หนึ่งมีความสูง 10 เปอร์เซ็นต์ของหน้าเว็บ ส่วนแถวที่สองนั้นมีความกว้างเท่ากับจำนวนเปอร์เซ็นต์ที่เหลือทั้งหมดของหน้าเว็บ ซึ่งก็คือ 90 เปอร์เซ็นต์นั่นเอง โดยแถวที่ 1 มีชื่อว่า "logo" แสดงไฟล์ logo.html ส่วนแถวที่ 2 ซึ่งมีความสูง 90 เปอร์เซ็นต์นั้น ยังถูกแบ่งออกเป็น 2 คอลัมน์ โดยคอลัมน์ที่ 1 มีความกว้าง 20 เปอร์เซ็นต์ของหน้าเว็บ คอลัมน์นี้มีชื่อว่า "link" แสดงไฟล์ link.html ส่วนคอลัมน์ที่ 2 นั้นมีความกว้างเท่ากับจำนวนเปอร์เซ็นต์ที่เหลือทั้งหมดของหน้าเว็บซึ่งก็คือ 80 เปอร์เซ็นต์นั่นเอง คอลัมน์นี้มีชื่อว่า "display" แสดงไฟล์ welcome.html

1.3.7 แท็กไอเฟรม (อินไลน์เฟรม) เป็นแท็กที่ใช้ในการกำหนดเอกสารชิ้นหนึ่งให้แทรกอยู่ในเอกสารอีกชิ้นหนึ่ง โดยเอกสารทั้ง 2 นั้นไม่ได้เป็นส่วนหนึ่งส่วนใดของกันและกัน แต่จะแยกจากกันโดยสิ้นเชิง สมมุติว่ามีไฟล์เอกสารสกุล HTML อยู่ 4 ไฟล์ดังต่อไปนี้

1. ไฟล์ index_iframe.html เป็นไฟล์เริ่มต้น ที่มีการกำหนดรูปแบบของ iframe ดังในรูปที่ 1.27

..	..
..	..
10.	<body>
11.	<center>
12.	<h3>... Welcome ...</h3>
13.	What is an iframe? My Sports
14.	<hr>
15.	<iframe src="info.html" name="display" width="80%" height = "110px"> </iframe>
16.	</center>
17.	</body>
18.	..

รูปที่ 1.27: รูปแบบของการกำหนดไอเฟรมในไฟล์ index_iframe.html

บรรทัดที่ 13 เป็นการกำหนดจุดเชื่อมต่อ 2 จุด โดยจุดแรกมีชื่อว่า What is an iframe? เมื่อผู้ใช้กดจุดเชื่อมต่อนี้แล้ว จะทำการเปิดไฟล์ html ชื่อว่า iframe.html และจุดที่สองมีชื่อว่า sports เมื่อผู้ใช้กดจุดเชื่อมต่อนี้แล้ว จะทำการเปิดไฟล์ html ชื่อว่า sports.html โดยทั้ง 2 จุดเชื่อมต่อ ได้มีการกำหนดตำแหน่งเป้าหมายในการแสดงเป็น target="display" ซึ่งหมายถึงการเปิดไฟล์ html ที่เกี่ยวข้องในตำแหน่งเป้าหมายที่มีชื่อว่า display

บรรทัดที่ 14 เป็นการสร้างเส้นโดยใช้แท็ก <hr>

บรรทัดที่ 15 เป็นการกำหนดไอเฟรม โดยให้เอาข้อมูลที่อยู่ในไฟล์ info.html มาแสดง พร้อมกับกำหนดให้ตำแหน่งของไอเฟรมนี้มีชื่อว่า display มีขนาดความกว้างเท่ากับ 80 เปอร์เซ็นต์ของหน้าจอ มีขนาดความสูงเท่ากับ 110 พิกเซล ดังแสดงในรูปที่ 1.28

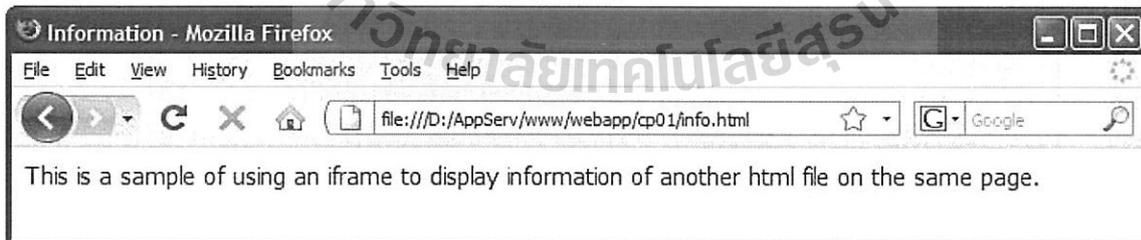


รูปที่ 1.28: การแสดงผลของไฟล์ index_iframe.html

2. ไฟล์ info.html เป็นไฟล์ที่แสดงข้อมูลชนิดหนึ่ง ดังในรูปที่ 1.29 ซึ่งจะแสดงผลดังในรูปที่ 1.30



รูปที่ 1.29: รูปแบบของการกำหนดการแสดงผลข้อมูลในไฟล์ info.html



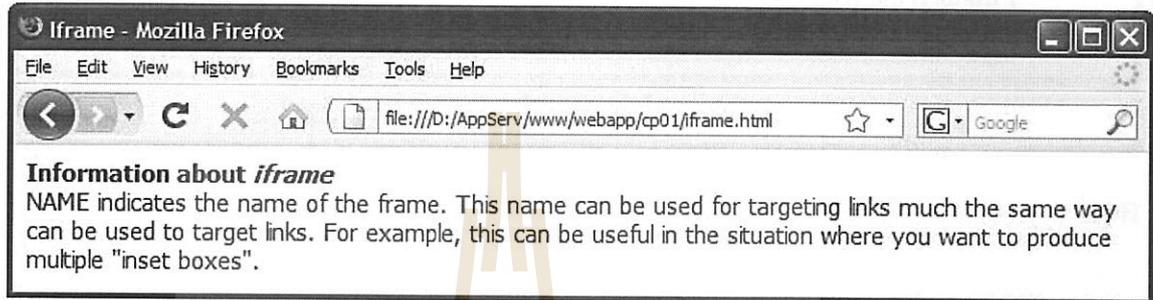
รูปที่ 1.30: การแสดงผลของไฟล์ info.html

3. ไฟล์ `iframe.html` เป็นไฟล์ที่แสดงข้อมูลชนิดหนึ่ง ดังในรูปที่ 1.31 ซึ่งจะแสดงผลดังในรูปที่ 1.32

```

..      ..
..      ..
10.     <body>
11.     <b>Information about <i>iframe</i></b><br>NAME indicates the name of the frame. This name
      can be used for targeting links much the same way can be used to target links. For example,
      this can be useful in the situation where you want to produce multiple "inset boxes".
12.     </body>
13.     ..
    
```

รูปที่ 1.31: รูปแบบของการกำหนดการแสดงผลข้อมูลในไฟล์ `iframe.html`



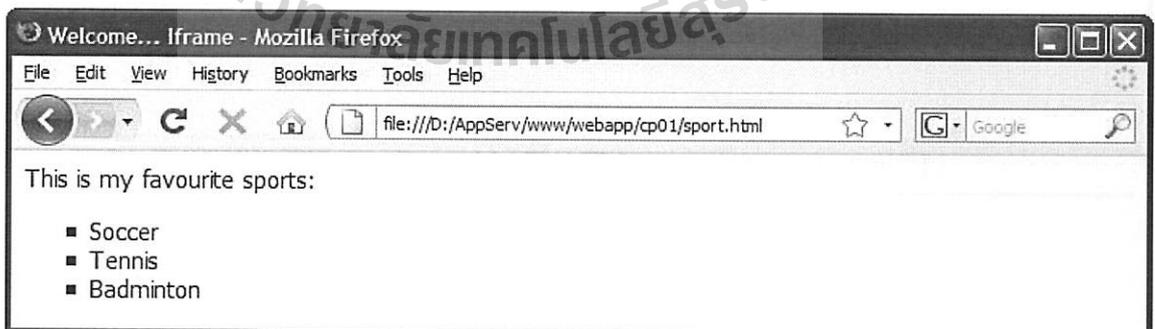
รูปที่ 1.32: การแสดงผลของไฟล์ `iframe.html`

4. ไฟล์ `sports.html` เป็นไฟล์ที่แสดงรายชื่อชนิดกีฬาที่ชอบ ดังในรูปที่ 1.33 ซึ่งจะแสดงผลดังในรูปที่ 1.34

```

..      ..
..      ..
10.     <body>
11.     This is my favourite sports:<br>
12.     <ul type="square">
13.         <li>Soccer</li>
14.         <li>Tennis</li>
15.         <li>Badminton</li>
16.     </ul>
17.     </body>
18.     ..
    
```

รูปที่ 1.33: รูปแบบของการกำหนดการแสดงผลรายชื่อชนิดกีฬาที่ชอบในไฟล์ `sports.html`



รูปที่ 1.34: การแสดงผลของไฟล์ `sports.html`

เมื่อเปิดไฟล์ `index_iframe.html` ในโปรแกรมเว็บเบราว์เซอร์แล้ว ผู้ใช้กดจุดเชื่อมต่อ `what is iframe?` จะมีการเปิดไฟล์ชื่อว่า `iframe.html` โดยนำเอาข้อมูลที่อยู่ในไฟล์ `iframe.html` มาแสดงออกที่ตำแหน่งเป้าหมายที่ชื่อว่า `display` ดังในรูปที่ 1.35



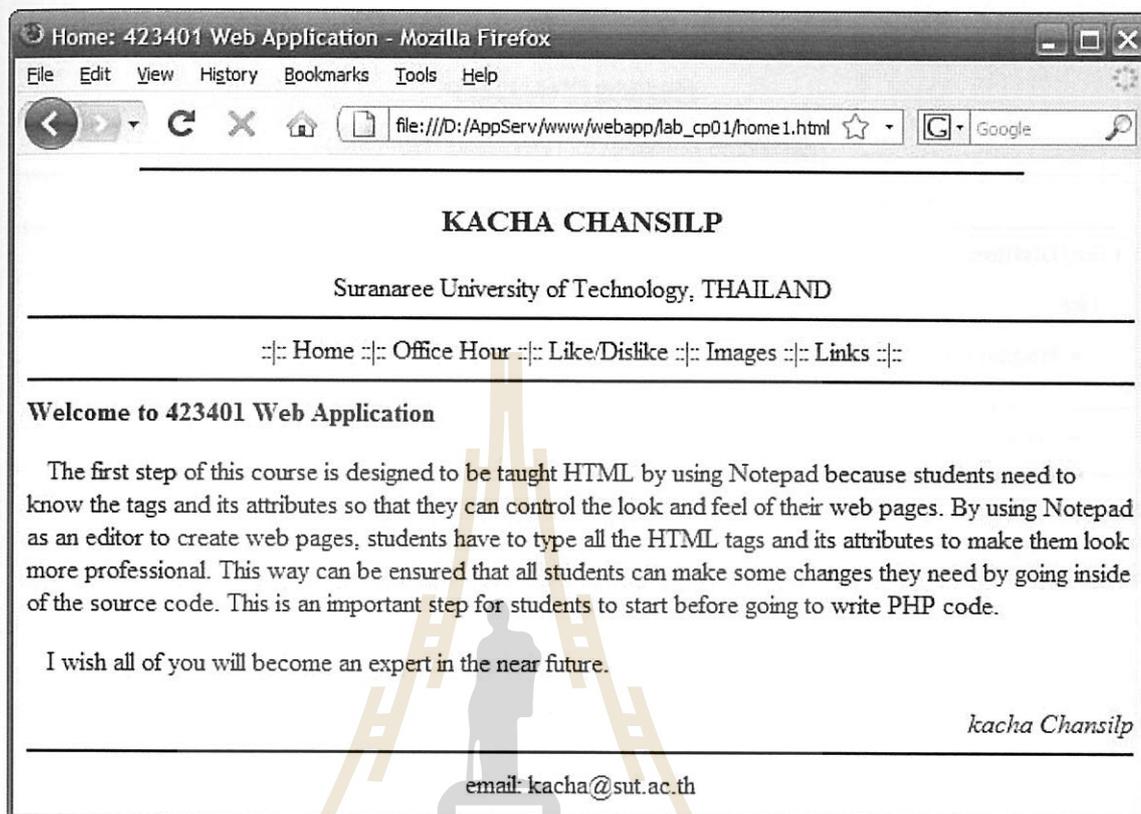
รูปที่ 1.35: การแสดงผลของไฟล์ iframe.html ในตำแหน่งเป้าหมายที่ชื่อว่า display

สรุป

ภาษา HTML ใช้สำหรับการนำเสนอข้อมูล ข่าวสาร และสื่อประสมต่าง ๆ ผ่านโปรแกรมเว็บเบราว์เซอร์ โดยใช้แท็กต่าง ๆ ในการจัดรูปแบบและควบคุมการนำเสนอ ทั้งยังสามารถสร้างแบบฟอร์มเพื่อรับข้อมูลจากผู้ใช้ และเก็บข้อมูลเหล่านั้นไว้สำหรับการประมวลผล ซึ่งจะได้กล่าวถึงการจัดเก็บข้อมูลลงระบบฐานข้อมูลหรือส่งข้อมูลต่อไปยังเซิร์ฟเวอร์เพื่อทำการประมวลผลในบทต่อไป

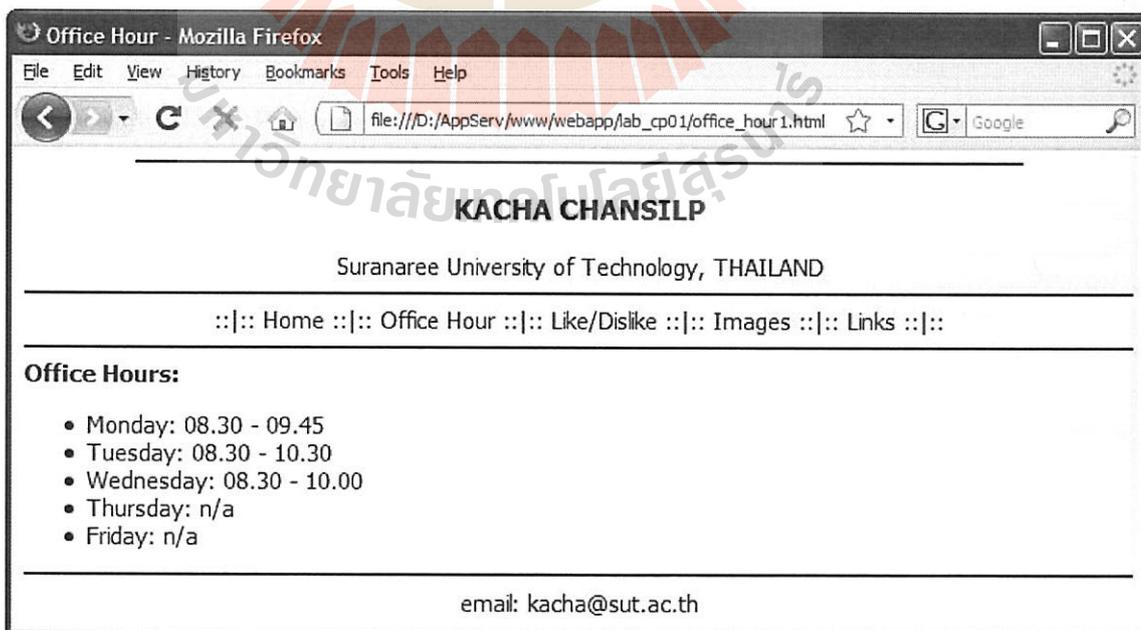
แบบฝึกหัด

1. ใช้โปรแกรม Notepad สร้างเว็บเพจชื่อว่า “home1.html” ดังรูปที่ 1.36



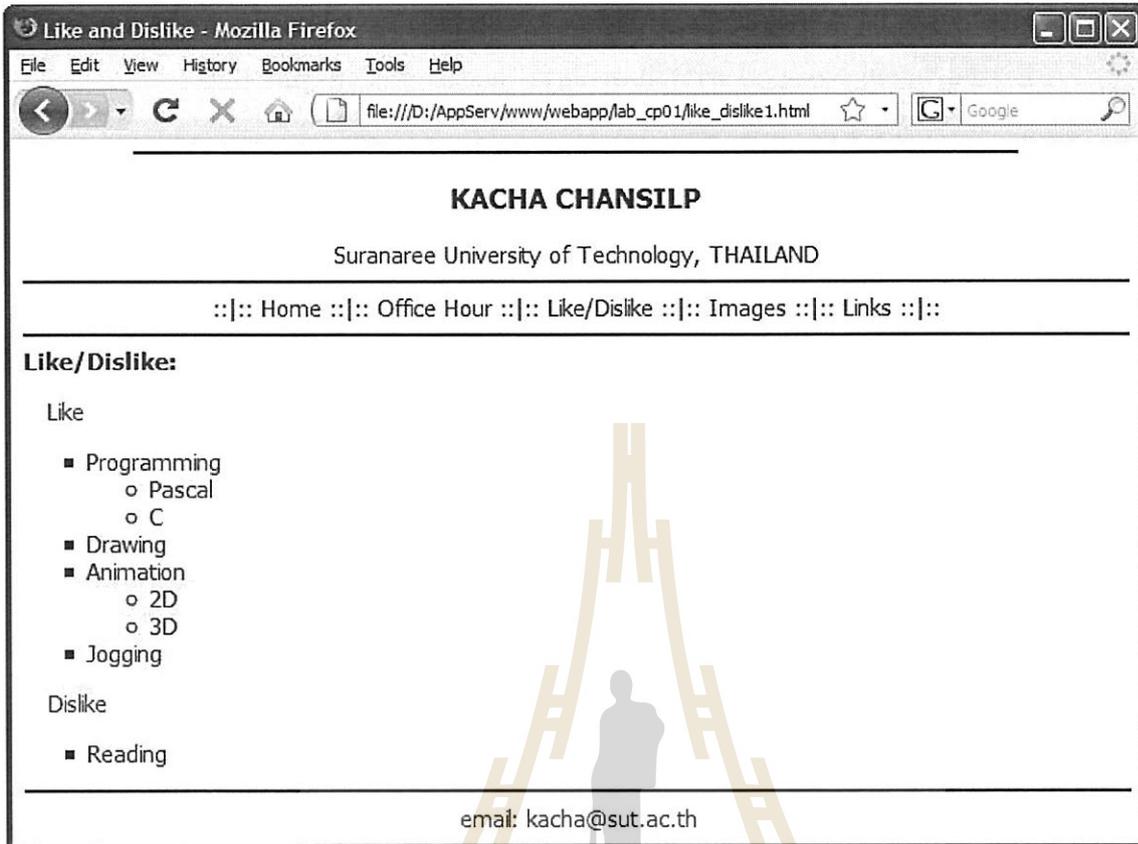
รูปที่ 1.36: การแสดงผลของไฟล์ home1.html

2. ใช้โปรแกรม Notepad สร้างเว็บเพจชื่อว่า “office_hour1.html” ดังรูปที่ 1.37



รูปที่ 1.37: การแสดงผลของไฟล์ office_hour1.html

3. ใช้โปรแกรม Notepad สร้างเว็บเพจชื่อว่า “like_dislike1.html” ดังรูปที่ 1.38



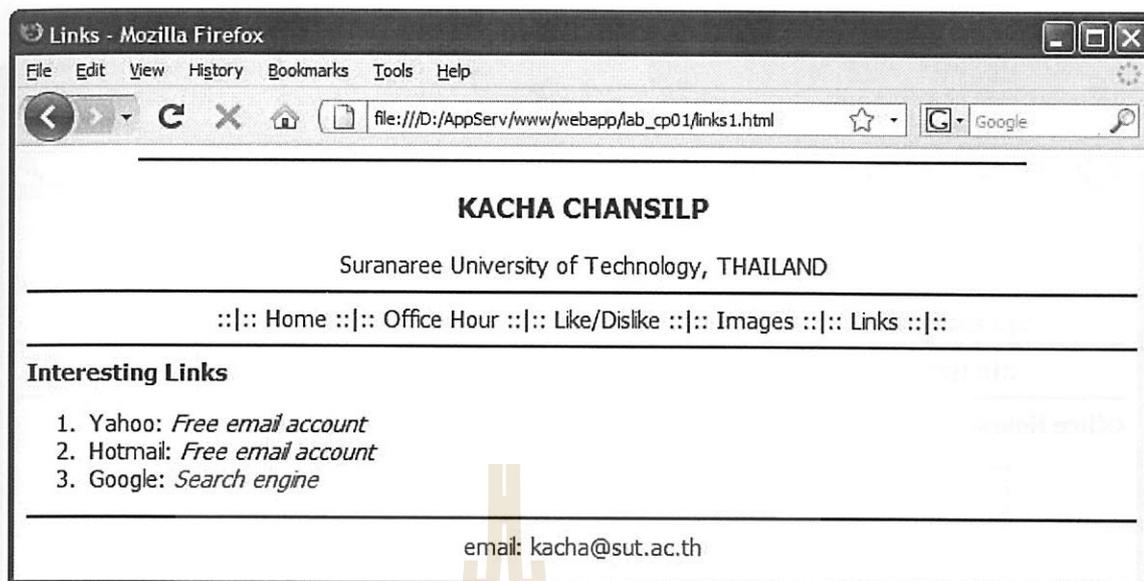
รูปที่ 1.38: การแสดงผลของไฟล์ like_dislike1.html

4. ใช้โปรแกรม Notepad สร้างเว็บเพจชื่อว่า “images1.html” ดังรูปที่ 1.39



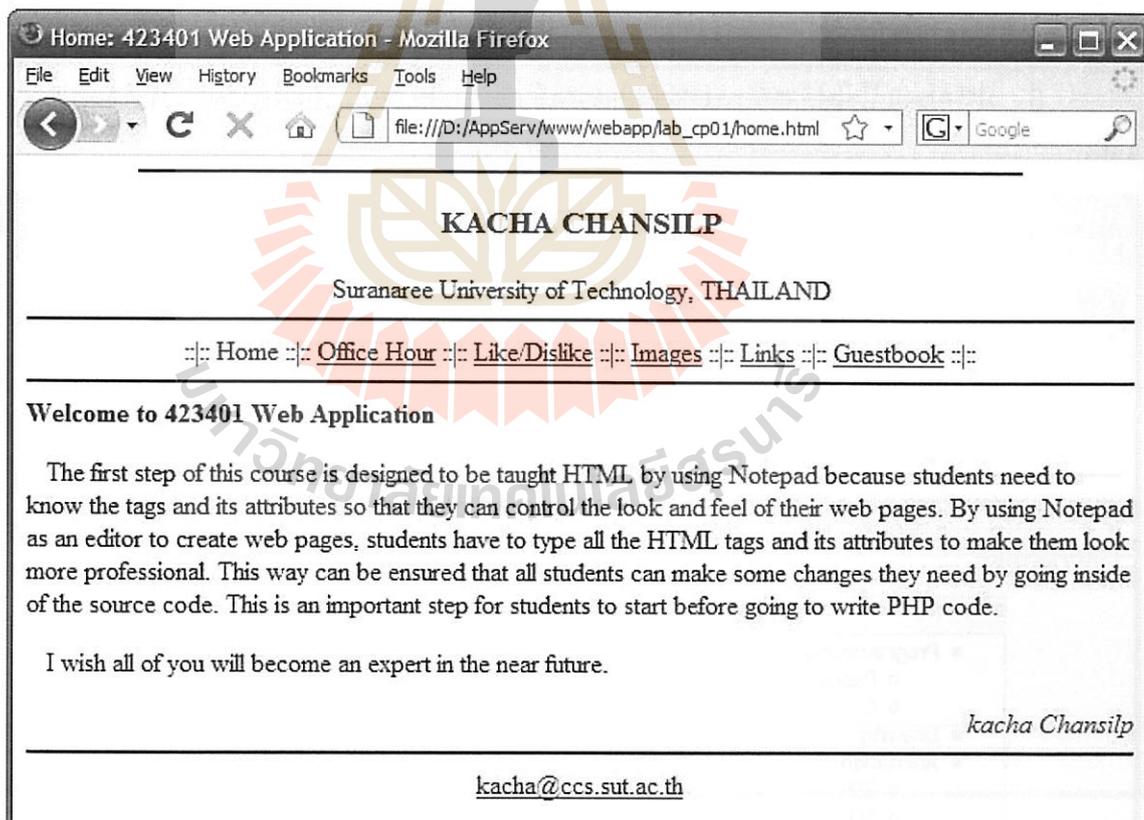
รูปที่ 1.39: การแสดงผลของไฟล์ images1.html

5. ใช้โปรแกรม Notepad สร้างเว็บเพจชื่อว่า “links1.html” ดังรูปที่ 1.40



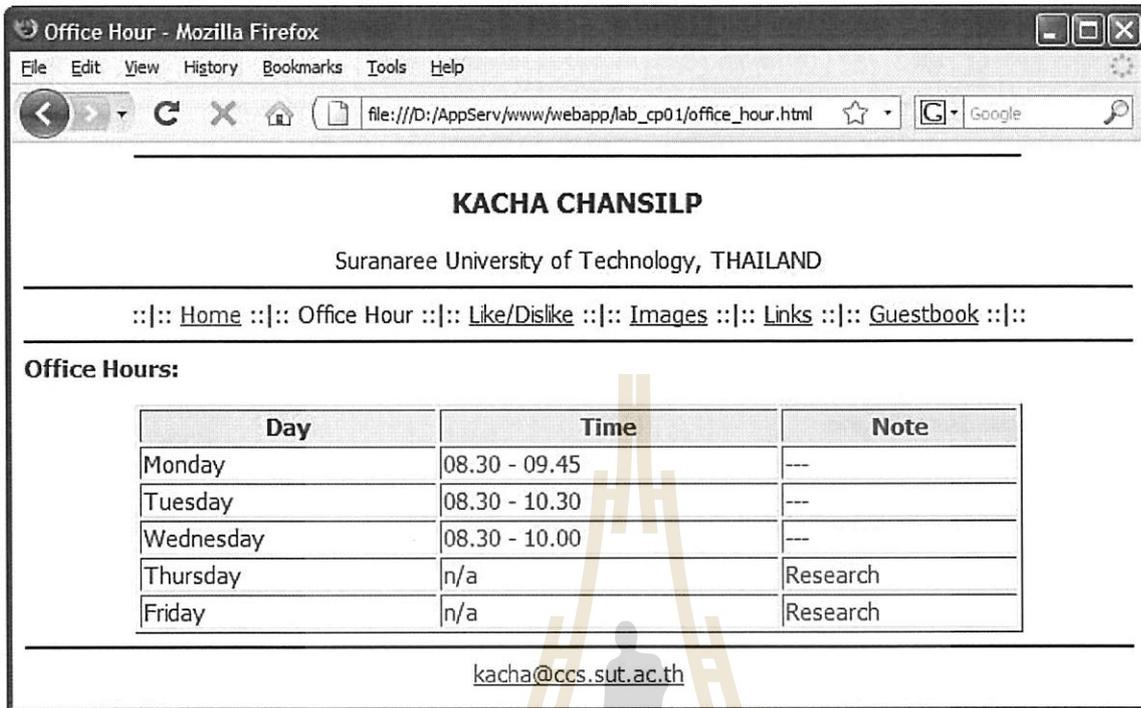
รูปที่ 1.40: การแสดงผลของไฟล์ links1.html

6. ปรับปรุงแก้ไขไฟล์ “home1.html” แล้วบันทึกเป็น “home.html” โดยเพิ่มจุดเชื่อมต่อต่าง ๆ ดังรูปที่ 1.41



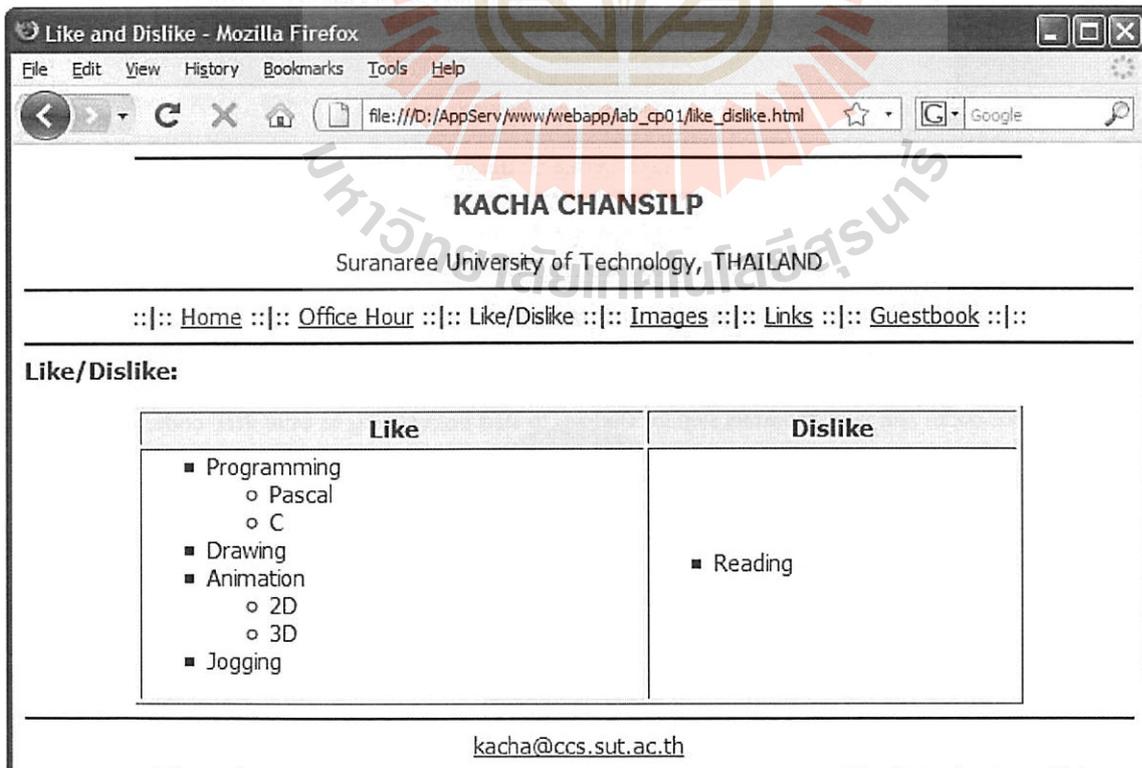
รูปที่ 1.41: การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ home.html

7. ปรับปรุงแก้ไขไฟล์ “office_hour1.html” แล้วบันทึกเป็น “office_hour.html” ให้ได้ดังรูปที่ 1.42



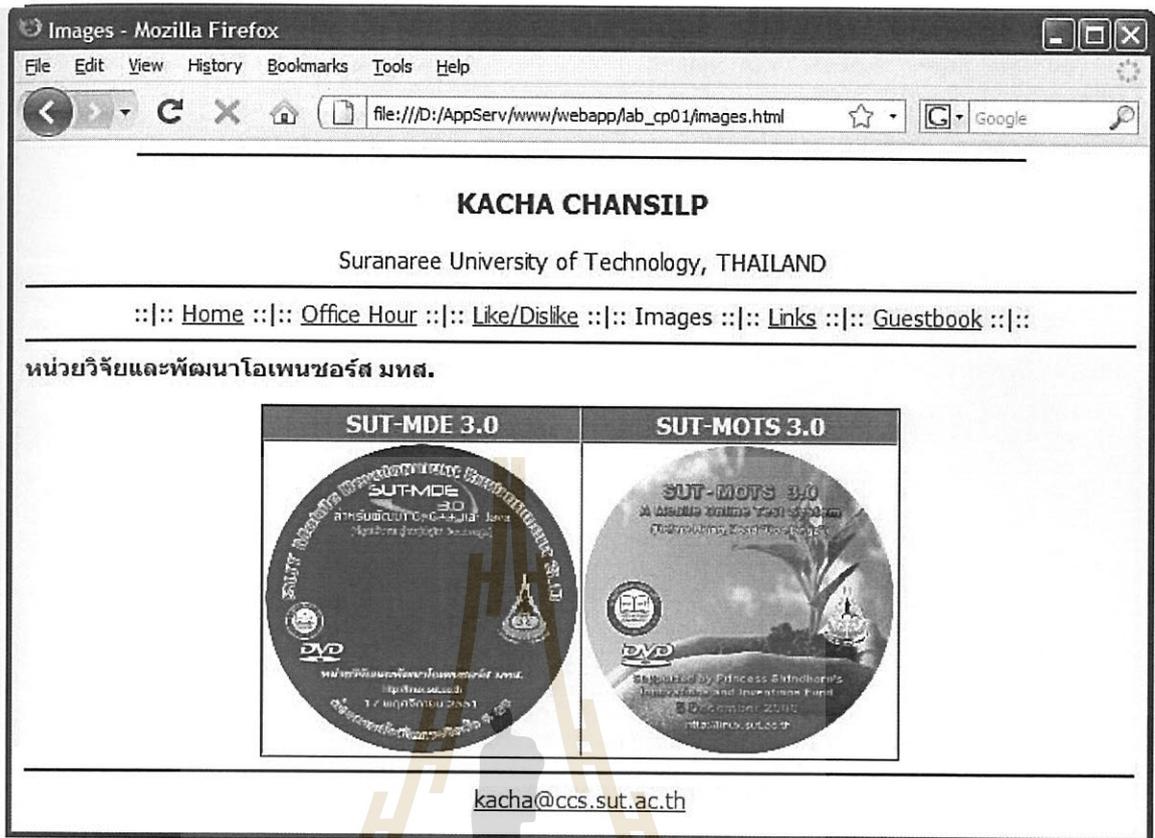
รูปที่ 1.42: การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ office_hour.html

8. ปรับปรุงแก้ไขไฟล์ “like_dislike1.html” แล้วบันทึกเป็น “like_dislike.html” ให้ได้ดังรูปที่ 1.43



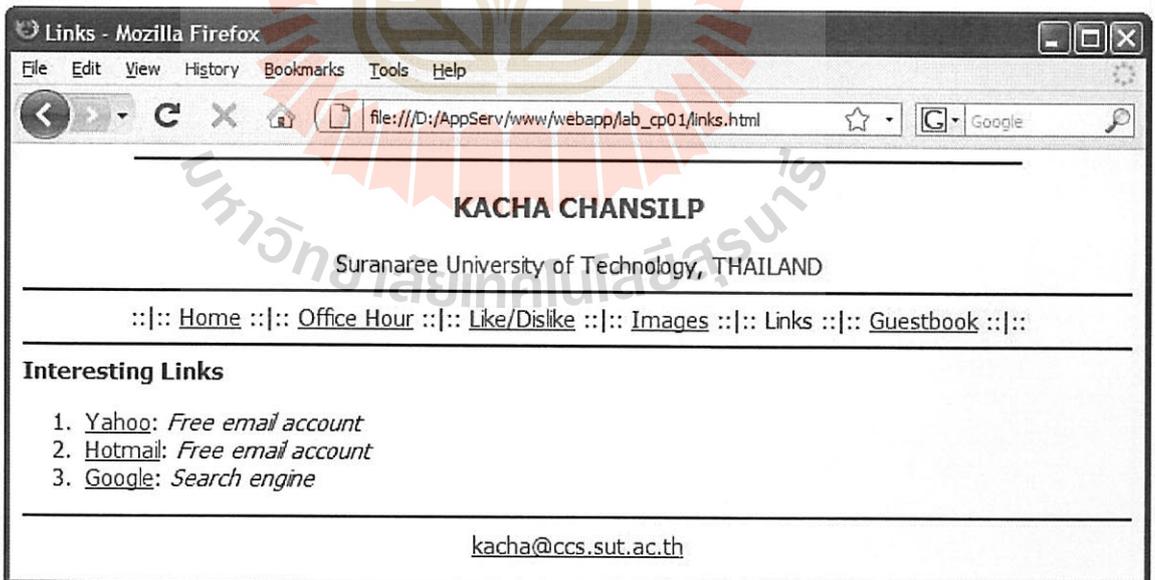
รูปที่ 1.43: การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ like_dislike.html

9. ปรับปรุงแก้ไขไฟล์ “images1.html” แล้วบันทึกเป็น “images.html” ให้ได้ดังรูปที่ 1.44



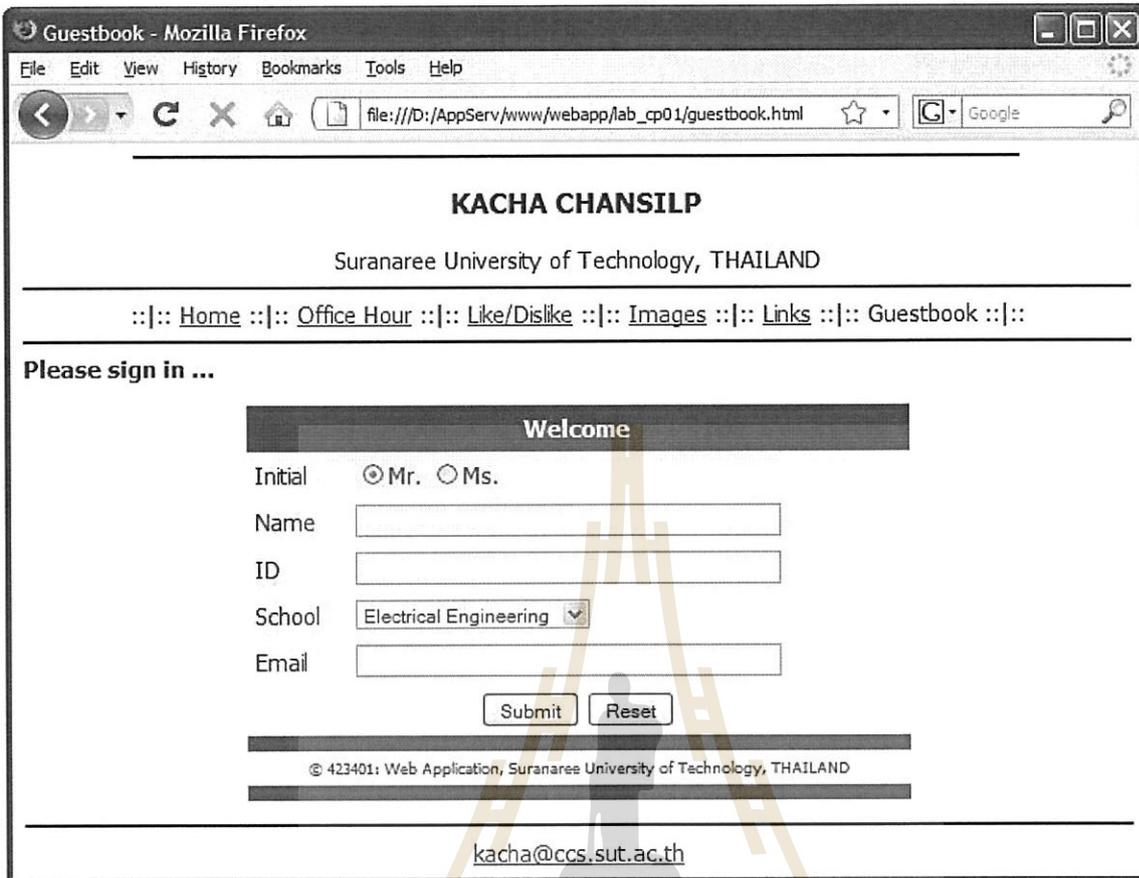
รูปที่ 1.44: การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ images.html

10. ปรับปรุงแก้ไขไฟล์ “links1.html” แล้วบันทึกเป็น “links.html” ให้ได้ดังรูปที่ 1.45



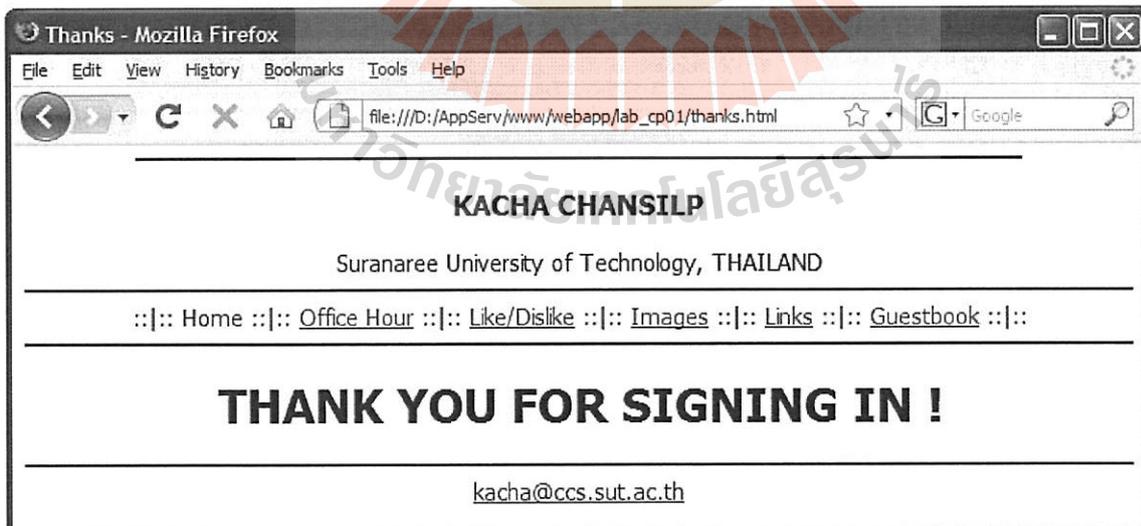
รูปที่ 1.45: การแสดงผลเมื่อปรับปรุงแก้ไขไฟล์ links.html

11. ใช้โปรแกรม Notepad สร้างเว็บเพจชื่อว่า “guestbook.html” ดังรูปที่ 1.46



รูปที่ 1.46: การแสดงผลของไฟล์ guestbook.html

เมื่อผู้ใช้กดปุ่ม Submit Query จะเปิดหน้าต่างใหม่ขึ้นมาดังรูปที่ 1.47



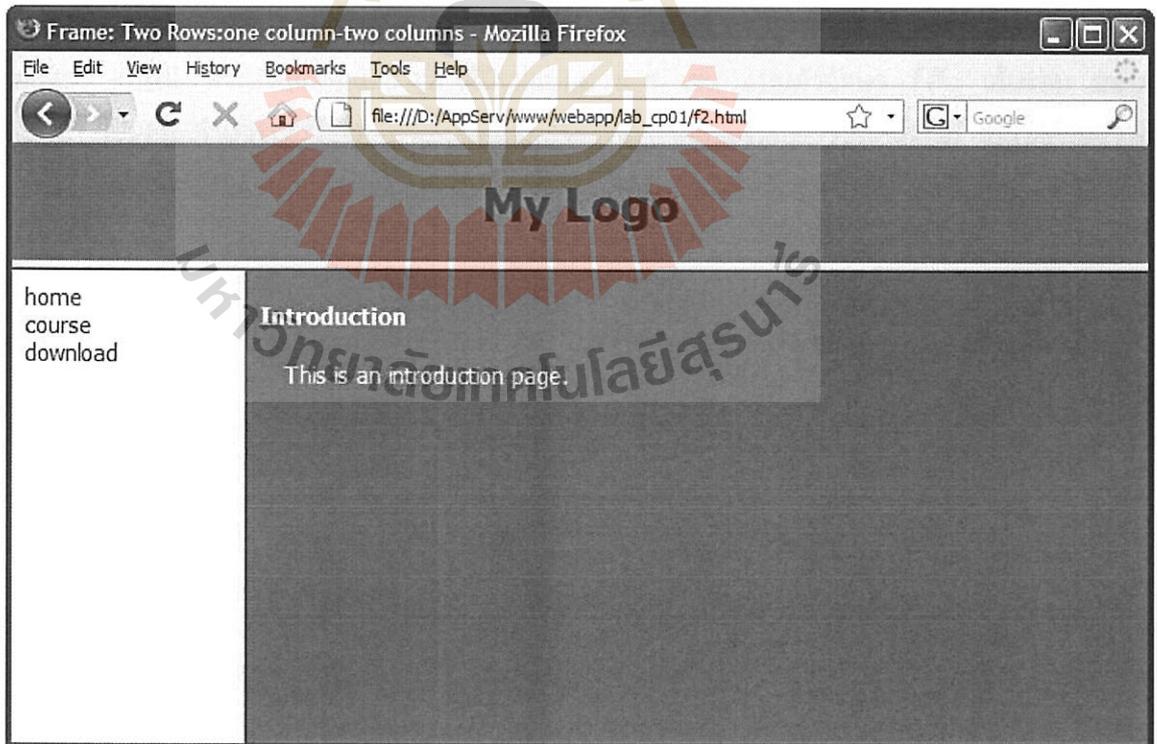
รูปที่ 1.47: การแสดงผลของไฟล์ thanks.html

12. ใช้โปรแกรม Notepad สร้างเฟรมดังรูปที่ 1.48



รูปที่ 1.48: การแสดงผลของไฟล์ f1.html

13. ใช้โปรแกรม Notepad สร้างเฟรมดังรูปที่ 1.49



รูปที่ 1.49: การแสดงผลของไฟล์ f2.html

บทที่ 2 แนะนำ CSS

<Introduction to Cascading Style Sheet>

บทนำ

การใช้คำสั่งต่าง ๆ ในภาษา HTML เพื่อตกแต่งเอกสารหรือข้อมูลที่ต้องการแสดงบนหน้าเว็บนั้น มีความสลับซับซ้อนมาก บางคำสั่งมีการเรียกใช้บ่อยครั้ง ซึ่งต้องแทรกคำสั่งเหล่านั้นในตำแหน่งที่ต้องการ ทุก ๆ ครั้ง ทำให้การปรับแต่งรูปแบบของการนำเสนอเป็นไปด้วยความยุ่งยากและเสียเวลามาก ทั้งยังทำให้เอกสาร HTML ของเรานั้นมีขนาดใหญ่และสลับซับซ้อนมากขึ้น ดังนั้น จึงได้มีการพัฒนาภาษา CSS ขึ้นมา เพื่อเป็นตัวช่วยในการจัดรูปแบบของภาษา HTML ให้ง่ายและสะดวกยิ่งขึ้น

วัตถุประสงค์

1. เพื่อให้รู้และเข้าใจในการกำหนดรูปแบบของเอกสารโดยใช้ CSS
2. เพื่อทดลองกำหนดรูปแบบต่าง ๆ โดยใช้ CSS

เนื้อหา

2.1 CSS คืออะไร?

CSS ย่อมาจาก Cascading Style Sheet เป็นภาษา ๆ หนึ่ง ที่ใช้ช่วยในการกำหนดรูปแบบของเอกสาร HTML ที่จะนำเสนอบนหน้าเว็บให้มีรูปแบบต่าง ๆ เช่น แบบตัวอักษร สี ลี้นขอบ ขนาด พื้นหลัง เป็นต้น ทำให้การปรับแต่งหรือเปลี่ยนแปลงการแสดงผลของเอกสารบนหน้าเว็บ เป็นไปด้วยความสะดวกและรวดเร็วยิ่งขึ้น ซึ่งประกอบไปด้วย 3 ส่วนคือ selector, property และ value มีลักษณะการเขียนคำสั่งดังนี้

```
selector{  
    property: value;  
}
```

1. Selector คือแท็กในภาษา HTML ที่เราต้องการกำหนดรูปแบบ เช่น p, h1, table, tr, td, ol เป็นต้น
2. Property คือคุณสมบัติที่เราต้องการกำหนดให้กับ selector นั้น ๆ เช่น font-style, font-size, color, text-align เป็นต้น
3. Value คือ ค่าที่ต้องการตั้งให้กับคุณสมบัตินั้น ๆ เช่น font-size: 12pt; ก็คือการกำหนดค่า 12 พอยต์ให้กับขนาดของตัวอักษร

ตัว selector สามารถกำหนดให้มีหลายตัวได้โดยใช้เครื่องหมายจุลภาค (,) กั้น เช่น ถ้าต้องการกำหนดให้ h1, h2, และ h3 ให้มีสีแดง สามารถทำได้โดยใช้คำสั่งดังนี้

```
h1, h2, h3{
    color: red;
}
```

ตัวอย่าง การกำหนดรูปแบบของย่อหน้า (paragraph) ให้อยู่ในตำแหน่งตรงกลาง ตัวอักษร arial สีดำ ขนาด 10 พอยต์ ความสูงของแถวข้อความ 30 พอยต์ สามารถกำหนดได้ดังนี้

```
p{
    text-align: center;
    font-style: italic;
    font-family: arial;
    font-size: 10pt;
    line-height: 30pt;
    color: black;
}
```

จะสังเกตเห็นว่า การกำหนดคุณสมบัติของ font นั้นมีทั้งหมด 3 คุณสมบัติด้วยกันคือ font-family, font-size และ line-height ซึ่งคุณสมบัติทั้ง 3 นี้สามารถเขียนคำสั่งแบบสั้นได้ดังนี้

```
p{
    text-align: center;
    font: italic 10pt/30pt arial;
    color: black;
}
```

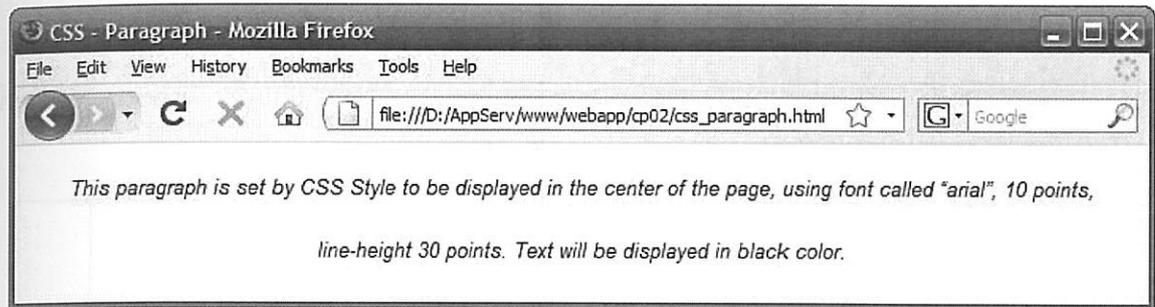
ข้อควรระวัง ถ้ามีช่องว่างระหว่างขนาดของตัวอักษรและหน่วยวัด เช่น font-size: 10 pt; รูปแบบที่กำหนดนี้จะแสดงผลได้เฉพาะบน IE เท่านั้น ดังนั้น ควรเขียนติดกันเป็น font-size: 10pt; เพื่อให้รูปแบบที่กำหนดนั้นทำงานได้ปกติทั้ง IE และ Mozilla/Firefox

1.	<html>
2.	<head>
3.	<title>CSS - Paragraph</title>
4.	<style type="text/css">
5.	p{
6.	text-align: center;
7.	font-style: italic;
8.	font-family: arial;
9.	font-size: 10pt;
10.	line-height: 30pt;
11.	color: black;
12.	}
13.	</style>
14.	</head>
15.	<body>
16.	<p>This paragraph is set by CSS Style to be displayed in the center of the page, using
17.	font called "arial", 10 points, line-height 30 points. Text will be displayed in black
18.	color.</p>
19.	</body>
20.	</html>

รูปที่ 2.1: การใช้ CSS กำหนดรูปแบบให้กับแท็ก <p>

บรรทัดที่ 4 – 13 เป็นการกำหนดรูปแบบให้กับตัวเลือก (selector) ที่ชื่อว่า p ซึ่งก็คือคำสั่งของการขึ้นย่อหน้าใหม่ (paragraph) ในภาษา HTML ให้มีการแสดงผลในตำแหน่งตรงกลาง โดยใช้ตัวอักษรชื่อว่า arial สีดำ ขนาด 10 พอยต์ มีระยะห่างระหว่างบรรทัด 30 พอยต์

บรรทัดที่ 16 – 18 เป็นการเรียกใช้แท็ก <p> และมีข้อความต่าง ๆ อยู่ภายใน ซึ่งจะถูกแสดงผลในรูปแบบที่กำหนดไว้ในบรรทัดที่ 4-13



รูปที่ 2.2: การแสดงผลของไฟล์ HTML ในรูปที่ 2.1

2.2 การกำหนดรูปแบบของ CSS

การกำหนดรูปแบบการแสดงผลของไฟล์ HTML บนหน้าเว็บ โดยใช้ CSS นั้น สามารถกำหนดได้ 3 ลักษณะด้วยกันคือ

2.2.1 แบบในบรรทัด (Inline Styles)

การกำหนดรูปแบบในบรรทัดไม่เป็นที่นิยมนัก เพราะว่ายุ่งยากในการจัดการ ปรับแต่ง หรือเปลี่ยนแปลง แต่ก็ยังเป็นทางเลือกที่สามารถใช้ได้ การกำหนดแบบนี้สามารถทำได้โดยใช้ attribute ชื่อว่า style ในการกำหนดรูปแบบให้กับแท็กแต่ละแท็ก

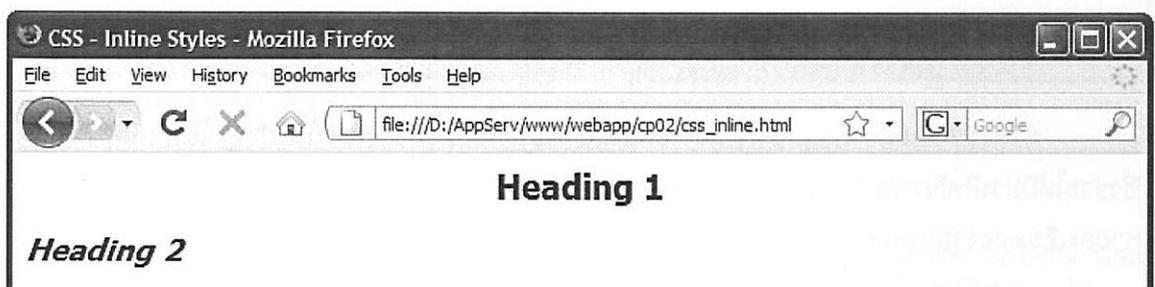
```
<tagname style="attribute:value; attribute:value;">...</tagname>
```

1.	<html>
2.	<head>
3.	<title>CSS - Inline Styles</title>
4.	</head>
5.	<body>
6.	<h1 style="font-size: 16pt; text-align: center">Heading 1</h1>
7.	<h2 style="font-size: 14pt; font-style: italic">Heading 2</h2>
8.	</body>
9.	</html>

รูปที่ 2.3: การใช้ CSS แบบในบรรทัด (Inline Styles)

บรรทัดที่ 6 เป็นการกำหนดรูปแบบในบรรทัดให้กับแท็ก <h1> โดยใช้ attribute ชื่อว่า style และกำหนดให้ขนาดตัวอักษรเป็น 16 พอยต์ แสดงตรงกลาง

บรรทัดที่ 7 เป็นการกำหนดรูปแบบในบรรทัดให้กับแท็ก <h2> โดยใช้ attribute ชื่อว่า style และกำหนดให้ขนาดตัวอักษรเป็น 14 พอยต์ ตัวเอียง



รูปที่ 2.4: การแสดงผลของไฟล์ HTML ในรูปที่ 2.3

2.2.2 แบบภายใน (Internal Style Sheet)

การกำหนด CSS ให้เป็นแบบภายในคือ การกำหนดรูปแบบให้อยู่ภายในแท็ก <head> ของไฟล์ HTML ซึ่งสามารถทำได้โดยใช้คำสั่ง

```
<style type="text/css">
  selector{
    property: value;
  }
  selector{
    property: value;
  }
</style>
```

ตัวอย่าง การกำหนดรูปแบบในส่วนของแท็ก <body> และ <h1> แบบภายใน

<ol style="list-style-type: none"> 1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.18.19.20. 	<pre><html> <head> <title>CSS: Internal Style Sheet</title> <meta http-equiv="content-type" content="text/html; charset=tis-620"> <style type="text/css"> body{ background: black; color: white; } h1{ font-size: 16pt; font-family: courier new, arial; text-align: center; } </style> </head> <body> <h1>This is a sample of using CSS: Internal Style Sheet.</h1> </body> </html></pre>
--	--

รูปที่ 2.5: การใช้ CSS แบบภายใน (Internal Style Sheet)



รูปที่ 2.6: การแสดงผลของไฟล์ HTML ในรูปที่ 2.5

2.2.3 แบบภายนอก (External Style Sheet)

การนำเสนอที่มีจำนวนหน้าเว็บหลายหน้านั้น ควรกำหนดรูปแบบโดยใช้ CSS ให้เป็นแบบภายนอก ซึ่งช่วยให้มีประสิทธิภาพมากยิ่งขึ้น สามารถทำได้โดยการสร้างไฟล์ขึ้นมาใหม่ไฟล์หนึ่ง ซึ่งมีการกำหนดรายละเอียดของรูปแบบต่าง ๆ ที่เราต้องการ แล้วบันทึกให้มีสกุลเป็น .css เท่านั้น สมมุติไฟล์นั้นมีชื่อว่า mystyle.css หลังจากนั้นในทุก ๆ ไฟล์ HTML เราก็ใช้คำสั่งดังต่อไปนี้ใส่ไว้ภายในแท็ก <head>

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

คำสั่งดังกล่าว เป็นการดึงข้อมูลจากไฟล์ mystyle.css เมื่อต้องการเปลี่ยนแปลงรูปแบบการนำเสนอ ก็เพียงแต่เข้าไปแก้ไขไฟล์ mystyle.css ไฟล์เดียวเท่านั้น ทำให้การบริหารจัดการเว็บไซต์มีความสะดวกและรวดเร็วยิ่งขึ้น

		mystyle.css
1.	body{	
2.		background: yellow;
3.		color: white;
4.	}	
5.		
6.	h1{	
7.		font-size: 20pt;
8.		font-family: courier new, arial;
9.		text-align: center;
10.	}	
11.	p{	
12.		font-size: 10pt;
13.		font-family: arial;
14.		color: black;
15.		font-style: italic
16.	}	

รูปที่ 2.7: การสร้างไฟล์ CSS

		css_external.html
1.	<html>	
2.	<head>	
3.	<title>CSS: External Style Sheet</title>	
4.	<link rel="stylesheet" type="text/css" href="mystyle.css">	
5.	</head>	
6.	<body>	
7.	<h1>External Style Sheet</h1>	
8.	<p>Cascading Style Sheets (CSS) are the most flexible way to create attractive,	
9.	standards-compliant websites. All your pages read off of one template. Wouldn't it be great	
10.	if you could set all the font face and size commands at once? It is very simple and easy to	
11.	update.</p>	
12.	</body>	
13.	</html>	

รูปที่ 2.8: การใช้ CSS แบบภายนอก (External Style Sheet)



รูปที่ 2.9: การแสดงผลของไฟล์ HTML ในรูปที่ 2.8

2.3 ตัวเลือกแบบคลาส (Class Selector)

วิธีการกำหนดรูปแบบให้กับ selector หรือแท็กหนึ่ง ๆ นั้น เราเรียกว่า Element Selector เมื่อมีการเรียกใช้แท็กนั้น ๆ ผลที่แสดงออกก็เป็นไปตามรูปแบบที่กำหนดไว้ เช่น

```
p{
    font-size: 10pt;
    font-family: arial;
    color: black;
}
```

เป็นการกำหนดการแสดงผลให้กับย่อหน้า (paragraph) ทุกครั้งที่มีการใช้แท็ก <p> ข้อความที่อยู่ระหว่างแท็กเปิด <p> และแท็กปิด </p> จะถูกแสดงออกเป็นตัวอักษร arial ขนาด 10 พอยต์ สีดำเสมอ ซึ่งการกำหนดแบบ Element Selector นี้จึงถือว่ามีข้อจำกัด เนื่องจากเมื่อไรที่มีการเรียกใช้แท็ก <p> ผลที่จะถูกแสดงออกก็จะเป็นลักษณะดังกล่าวทุกครั้งเสมอ ดังนั้น จึงควรแบ่ง selector หรือแท็กให้เป็นคลาสหรือชั้น ซึ่งเรียกว่า Class Selector

การแบ่ง selector ให้เป็นคลาสต่าง ๆ นั้น ทำให้สามารถกำหนดรูปแบบในการแสดงผลของ selector หรือแท็กหนึ่ง ๆ ได้หลายรูปแบบตามต้องการ เช่น

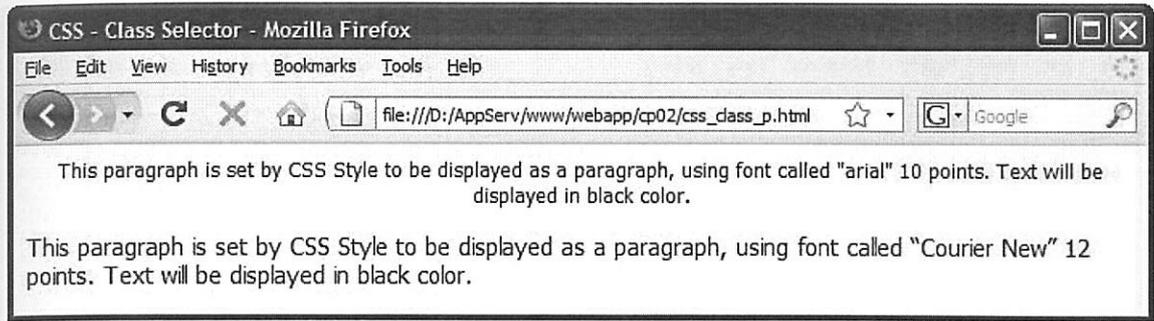
```
p.ten{
    font-size: 10pt;
    font-family: arial;
    color: black;
    text-align: center;
}
p.twelve{
    font-size: 12pt;
    font-family: Courier New;
    color: black;
}
```

จะสังเกตเห็นว่า selector p หรือแท็ก <p> ได้ถูกกำหนดให้มี 2 คลาสหรือเรียกอีกอย่างหนึ่งว่า 2 รูปแบบคือ

- รูปแบบที่ 1 มีชื่อว่า ten ซึ่งจะแสดงผลเป็นตัวอักษรชื่อว่า arial ขนาด 10 พอยต์ สีดำตำแหน่งตรงกลาง
- รูปแบบที่ 2 มีชื่อว่า twelve ซึ่งจะแสดงผลเป็นตัวอักษรชื่อว่า Courier New ขนาด 12 พอยต์ สีดำ

1.	<html>
2.	<head>
3.	<title>CSS - Class Selector</title>
4.	<style type="text/css">
5.	p.ten{
6.	font-size:10pt;
7.	font-family: arial;
8.	color: black;
9.	text-align: center;
10.	}
11.	p.twelve{
12.	font-size: 12pt;
13.	font-family: Courier New;
14.	color: black;
15.	}
16.	</style>
17.	</head>
18.	<body>
19.	<p class="ten">This paragraph is set by CSS Style to be displayed as a paragraph, using
20.	font called "arial" 10 points. Text will be displayed in black color.</p>
21.	<p class="twelve">This paragraph is set by CCS Style to be displayed as a paragraph,
22.	using font called "Courier New" 12 points. Text will be displayed in black color.</p>
	</body>
	</html>

รูปที่ 2.10: การแบ่งคลาสของ Selector หรือแท็ก



รูปที่ 2.11: การแสดงผลของไฟล์ HTML ในรูปที่ 2.10

การแบ่งคลาสดังกล่าวนี้ เป็นการแบ่งคลาสให้เฉพาะแท็ก <p> เท่านั้น ถ้าแท็กอื่นใดต้องการให้แสดงผลเป็นลักษณะเดียวกันกับคลาส twelve ต้องทำการกำหนดใหม่ให้กับแท็กนั้น ๆ ถ้าต้องการกำหนดคลาสเพื่อให้แท็กต่าง ๆ สามารถเรียกใช้ได้ ต้องกำหนดโดยไม่ผูกกับแท็กใด ๆ นั่นคือไม่ต้องระบุแท็กหน้าจุด เช่น

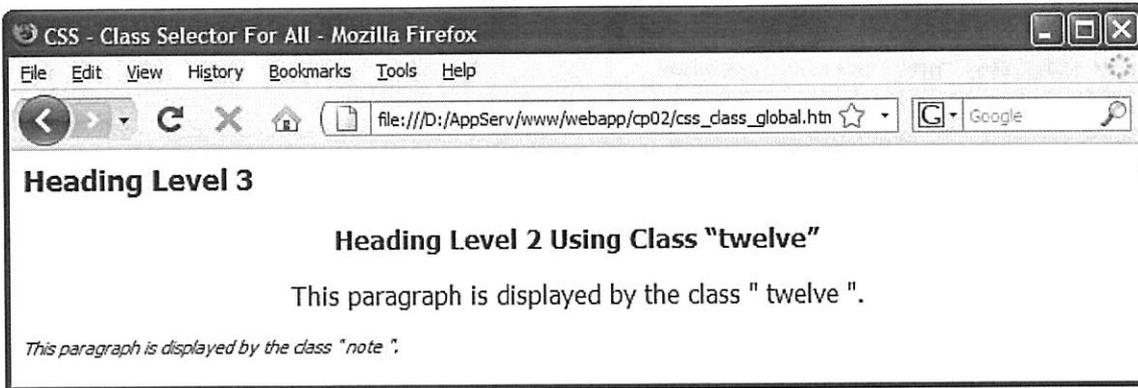
```
.note{
    font-size: 8pt;
    font-style: italic;
}
.twelve{
    font-size: 12pt;
    font-family: Courier New;
    text-align: center;
}
```

ข้อควรระวัง การตั้งชื่อคลาสนั้นจะต้องไม่ซ้ำกันและไม่ขึ้นต้นด้วยตัวเลข เนื่องจาก Mozilla/Firefox ไม่สนับสนุน

ตัวอย่าง การกำหนดคลาสชื่อ note และ twelve ที่แท็กต่าง ๆ สามารถเรียกใช้ได้

```
1. <html>
2. <head>
3. <title>CSS - Class Selector For All</title>
4. <style type="text/css">
5.     .note{
6.         font-size: 8pt;
7.         font-style: italic;
8.     }
9.     .twelve{
10.        font-size: 12pt;
11.        font-family: Courier New;
12.        text-align: center;
13.    }
14. </style>
15. </head>
16. <body>
17. <h3>Heading Level 3</h3>
18. <h2 class="twelve">Heading Level 2 Using Class "twelve"</h2>
19. <p class="twelve">This paragraph is displayed by the class " twelve ".</p>
20. <p class="note">This paragraph is displayed by the class " note ".</p>
21. </body>
22. </html>
```

รูปที่ 2.12: การกำหนดคลาสแบบทั่วไปสำหรับแท็กต่าง ๆ



รูปที่ 2.13: การแสดงผลของไฟล์ HTML ในรูปที่ 2.12

สมมุติว่ามีการกำหนดคลาส 2 คลาสดังต่อไปนี้

```
.center{
    text-align: center;
}
.red{
    color: red;
}
```

ถ้าต้องการกำหนดคลาส center และคลาส red ให้กับแท็ก p สามารถทำได้ดังนี้

```
<p class="center red">This is a paragraph.</p>
```

2.4 ตัวเลือกแบบไอดี (ID Selector)

การกำหนดรูปแบบอีกชนิดหนึ่งเรียกว่า ID Selector ซึ่งใช้เครื่องหมาย # นำหน้าชื่อ มีลักษณะการทำงานคล้ายกับแบบ Class Selector แต่สามารถกำหนดให้ใช้ได้กับทุก ๆ แท็กที่ต้องการเรียกใช้ หรือจะกำหนดให้ใช้ได้กับเฉพาะแท็ก ๆ เดียว ก็ได้ตามความต้องการของผู้ออกแบบ เช่น ถ้าต้องการกำหนดรูปแบบเป็นแบบ ID Selector ให้เฉพาะแท็ก <p> สามารถกำหนดได้ดังนี้

```
p#note{
    font-size: 8pt;
    font-family: arial;
    font-style: italic;
}
```

เมื่อไรที่มีการใช้แท็ก <p> และมี id คือ note จะแสดงผลเป็นตัวอักษรชื่อว่า arial ขนาด 8 พอยต์ ตัวเอียง แต่ถ้าต้องการกำหนด ID Selector ให้ใช้ได้กับทุก ๆ แท็กที่ต้องการเรียกใช้ ต้องกำหนดรูปแบบโดยไม่ผูกกับแท็กใด ๆ นั่นคือ ไม่ต้องระบุแท็กหน้าเครื่องหมาย # เช่น

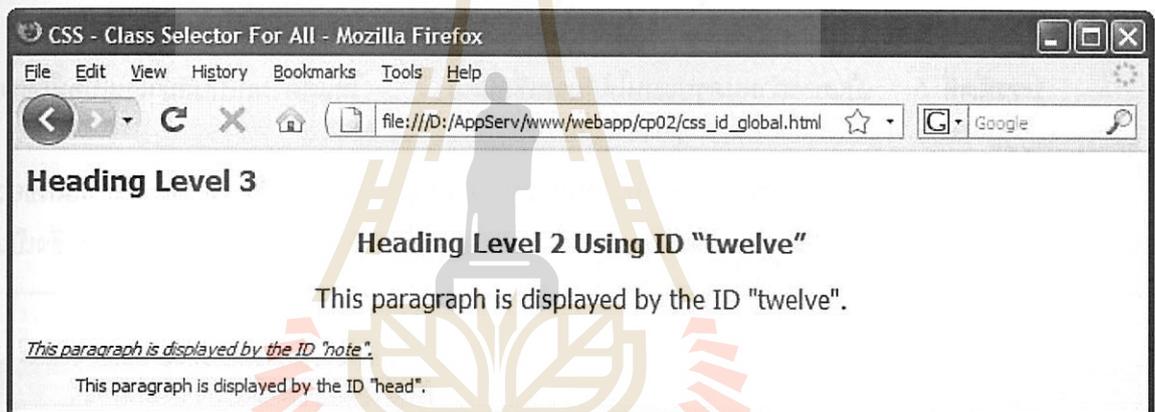
```
#note{
    font-size: 8pt;
    font-style: italic;
}
```

```

1. <html>
2.   <head><title>CSS - Class Selector For All</title>
3.     <style type="text/css">
4.       #note{
5.         font-size: 8pt;
6.         font-style: italic;
7.         text-decoration: underline;
8.       }
9.       #twelve{
10.        font-size: 12pt;
11.        font-family: Courier New;
12.        text-align: center;
13.      }
14.      p#head{
15.        font-size: 8pt;
16.        font-family: Courier New;
17.        text-indent: 3em;
18.      }
19.    </style>
20.  </head>
21.  <body>
22.    <h3>Heading Level 3</h3>
23.    <h2 id="twelve">Heading Level 2 Using ID "twelve"</h2>
24.    <p id="twelve">This paragraph is displayed by the ID "twelve".</p>
25.    <p id="note">This paragraph is displayed by the ID "note".</p>
26.    <p id="head">This paragraph is displayed by the ID "head".</p>
27.  </body>
28. </html>

```

รูปที่ 2.14: การกำหนดไอดีแบบทั่วไปสำหรับแท็กต่าง ๆ



รูปที่ 2.15: การแสดงผลของไฟล์ HTML ในรูปที่ 2.14

ข้อแนะนำ ควรกำหนดรูปแบบให้เป็นแบบ ID Selector เมื่อมีการเรียกใช้ 1 ครั้งบนหน้าเว็บ แต่ถ้ามีการเรียกใช้บ่อยครั้ง ควรเปลี่ยนเป็นแบบ Class Selector

2.5 รหัสเทียมแบบคลาส (Pseudo-classes) และรหัสเทียมแบบอีลีเมนต์ (Pseudo-elements)

Pseudo-classes และ Pseudo-elements เป็นคลาสและอีลีเมนต์พิเศษที่โปรแกรมท่องเว็บ (IE, Mozilla/Firefox) รู้จักและสามารถแสดงผลได้ตามที่กำหนด เป็นรูปแบบอีกชนิดหนึ่งที่ผู้ออกแบบเว็บไซต์สามารถนำไปเป็นตัวช่วยในการนำเสนอข้อมูลข่าวสารบนหน้าเว็บ

รูปแบบ Pseudo-classes

```

selector:pseudo-class{
    property: value;
}

```

ตัวอย่าง การกำหนดรูปแบบให้กับ Pseudo-classes อีริเมนต์ a

```
a:link{color: red; font-family: Courier New;}
a:active{color: blue;}
a:visited{color: green; text-decoration: none;}
```

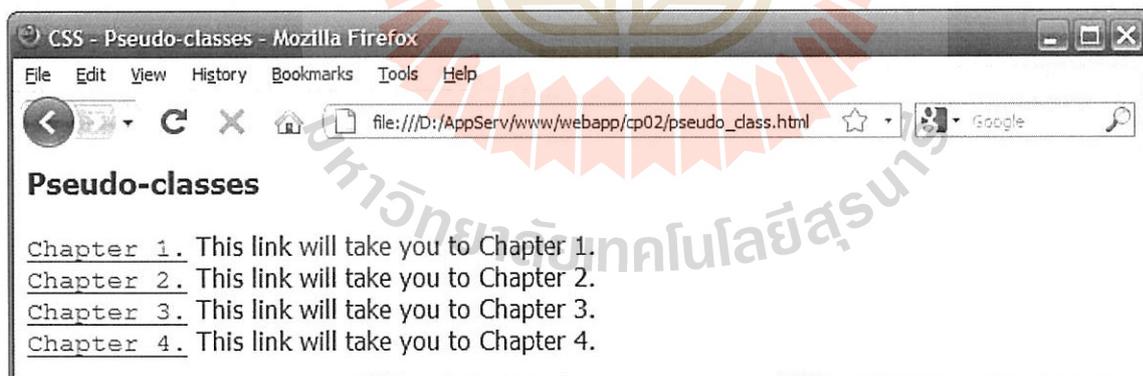
```
1. <html>
2.   <head>
3.     <title>CSS - Pseudo-classes</title>
4.     <style type="text/css">
5.       a:link{color : red; font-family: Courier New;}
6.       a:active{color: blue;}
7.       a:visited{color: green; text-decoration: none;}
8.     </style>
9.   </head>
10.  <body>
11.    <h3>Pseudo-classes</h3>
12.    <a href="chapter1.html">Chapter 1.</a> This link will take you to Chapter 1.<br>
13.    <a href="chapter2.html">Chapter 2.</a> This link will take you to Chapter 2.<br>
14.    <a href="chapter3.html">Chapter 3.</a> This link will take you to Chapter 3.<br>
15.    <a href="chapter4.html">Chapter 4.</a> This link will take you to Chapter 4.<br>
16.  </body>
17. </html>
```

รูปที่ 2.16: การกำหนด Pseudo-classes ให้กับอีริเมนต์ a

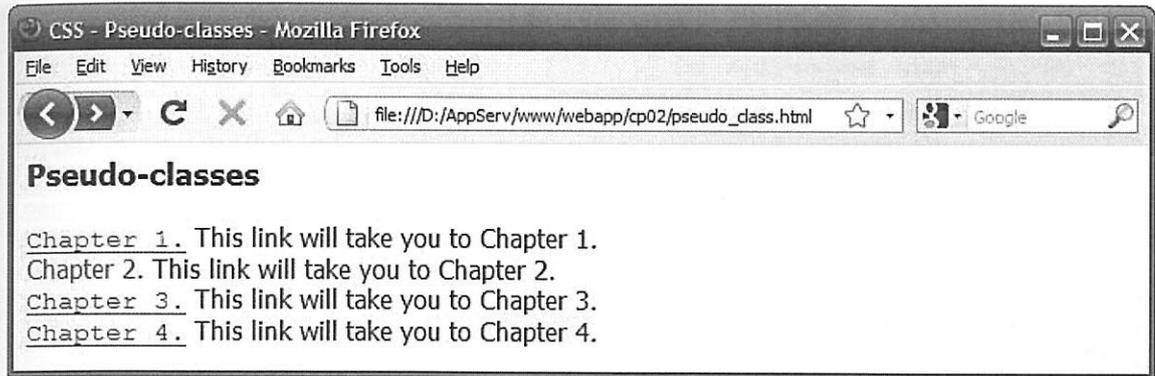
บรรทัดที่ 5 เป็นการกำหนดรูปแบบให้กับจุดเชื่อมต่อ ให้แสดงผลเป็นตัวอักษร Courier New สีแดง

บรรทัดที่ 6 เป็นการกำหนดรูปแบบให้กับจุดเชื่อมต่อ ให้แสดงผลเป็นสีน้ำเงินเมื่อผู้ใช้งานจุดเชื่อมต่อนั้น

บรรทัดที่ 7 เป็นการกำหนดรูปแบบให้กับจุดเชื่อมต่อ ที่ผู้ใช้ได้กดไปแล้ว ให้แสดงผลเป็นสีเขียว และไม่มีเส้นใต้ จะสังเกตเห็นว่าตัวอักษรก็จะเปลี่ยนจาก Courier New เป็น Times New Roman ซึ่งเป็นคำปริยาย ดังแสดงในรูปที่ 2.18



รูปที่ 2.17: การแสดงผลของไฟล์ HTML ในรูปที่ 2.16



รูปที่ 2.18: คำอธิบายของการแสดงตัวอักษร Times New Roman

รูปแบบ Pseudo-elements

```
selector:pseudo-element{
    property: value;
}
```

ตัวอย่าง การกำหนดรูปแบบให้กับ Pseudo-elements p

```
p{text-indent: 3ex}
p:first-line{color: red; font-family: Courier New;
font-weight: bold;}
p:first-letter{font-size: 300%; color: blue;
font-weight: bold; font-style: italic}
```

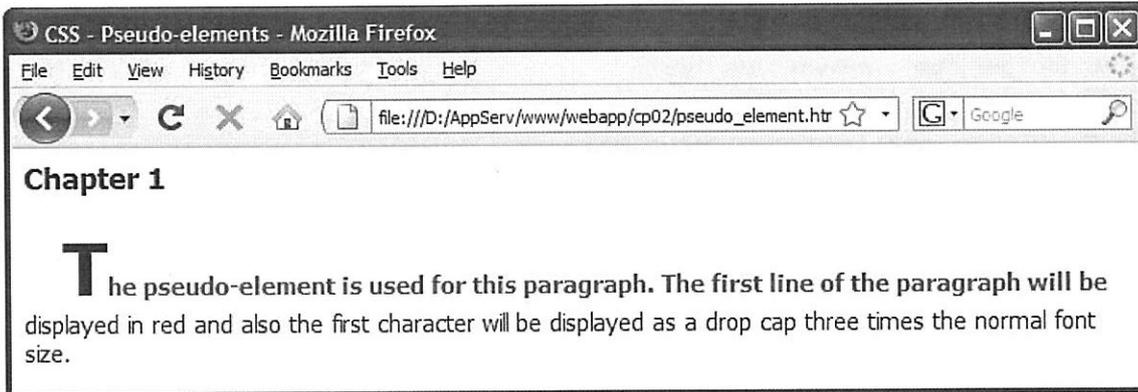
<ol style="list-style-type: none"> 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 	<pre><html> <head> <title>CSS - Pseudo-elements</title> <style type="text/css"> p{text-indent: 3ex} p:first-line{color: red; font-family: Courier New; font-weight: bold;} p:first-letter{font-size: 300%; color: blue; font-weight: bold; font- style: italic} </style> </head> <body> <h3>Chapter 1</h3> <p>The pseudo-element is used for this paragraph. The first line of the paragraph will be displayed in red and also the first character will be displayed as a drop cap three times the normal font size.</p> </body> </html></pre>
--	--

รูปที่ 2.19: การกำหนด Pseudo-element ให้กับ p

บรรทัดที่ 5 เป็นการกำหนดรูปแบบให้กับย่อหน้า (paragraph) ให้ตัวอักษรเริ่มต้นที่ตำแหน่งห่างจากขอบเป็นระยะเท่ากับ 3 เท่าของความสูงของตัวอักษรในแท็กนี้

บรรทัดที่ 6 เป็นการกำหนดรูปแบบแถวแรกของย่อหน้า ให้แสดงผลโดยใช้ตัวอักษร Courier New สีแดง ตัวหนา

บรรทัดที่ 7 เป็นการกำหนดรูปแบบตัวอักษรแรกของย่อหน้า ให้แสดงผลเป็นตัวอักษรสีน้ำเงิน ตัวหนาและเอียง มีขนาดเป็น 3 เท่าของตัวอักษรปกติ



รูปที่ 2.20: การแสดงผลของไฟล์ HTML ในรูปที่ 2.19

2.6 คุณสมบัติต่าง ๆ ใน CSS

คุณสมบัติต่าง ๆ ที่ควรรู้ ในการกำหนดรูปแบบการนำเสนอบนหน้าเว็บ ซึ่งสามารถรวมกลุ่มได้ทั้งหมด 6 กลุ่มด้วยกันคือ

1. คุณสมบัติของตัวอักษร (Font Properties)

ตารางที่ 2.1: คุณสมบัติของตัวอักษร

คุณสมบัติ	ค่า
font-family	arial, courier new
font-style	normal, italic, oblique
font-variant	normal, small-caps
font-weight	normal, bold, bolder, lighter
font-size	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %

2. คุณสมบัติของสีและพื้นหลัง (Color and Background Properties)

ตารางที่ 2.2: คุณสมบัติของสีและพื้นหลัง

คุณสมบัติ	ค่า
color	name, #rgb, #rrggbb, rgb(n,n,n), rgb(x%,x%,x%)
background-color	color, transparent
background-image	url, none
background-repeat	repeat, repeat-x, repeat-y, no-repeat
background-attachment	scroll, fixed
background-position	top, right, bottom, left, center,

3. คุณสมบัติของข้อความ (Text Properties)

ตารางที่ 2.3: คุณสมบัติของข้อความ

คุณสมบัติ	ค่า
word-spacing	normal, length
letter-spacing	normal, length
text-decoration	none, underline, overline, line-through, blink
vertical-align	baseline, sub, super, top, text-top, middle, bottom, text-bottom, percentage
text-transform	none, capitalize, uppercase, lowercase
text-align	left, right, center, justify
text-indent	length, percentage
line-height	normal, number, length, percentage
first-letter	font-size, font-weight, float, color

4. คุณสมบัติของกล่องข้อความ (Box Properties)

ตารางที่ 2.4: คุณสมบัติของกล่องข้อความ

คุณสมบัติ	ค่า
margin-top	length, percentage
margin-right	length, percentage
margin-bottom	length, percentage
margin-left	length, percentage
margin	length, percentage
padding-top	length, percentage
padding-right	length, percentage
padding-bottom	length, percentage
padding-left	length, percentage
padding	length, percentage
border-top-width	thin, medium, thick, length
border-right-width	thin, medium, thick, length
border-bottom-width	thin, medium, thick, length
border-left-width	thin, medium, thick, length
border-width	thin, medium, thick, length
border-color	color
border-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border-top-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border-bottom-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border-left-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border-right-style	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border	thin, medium, thick, length, none, dotted, dashed, solid, double, groove, ridge, inset, outset, color
width	length, percentage, auto
height	length, auto
float	none, left, right
clear	none, left, right, both

5. คุณสมบัติของการแบ่งแยก (Classification Properties)

ตารางที่ 2.5: คุณสมบัติของการแบ่งแยก

คุณสมบัติ	ค่า
display	block, inline, list-item, none
white-space	normal, pre, nowrap
list-style-type	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none
list-style-image	url, none
list-style-position	inside, outside

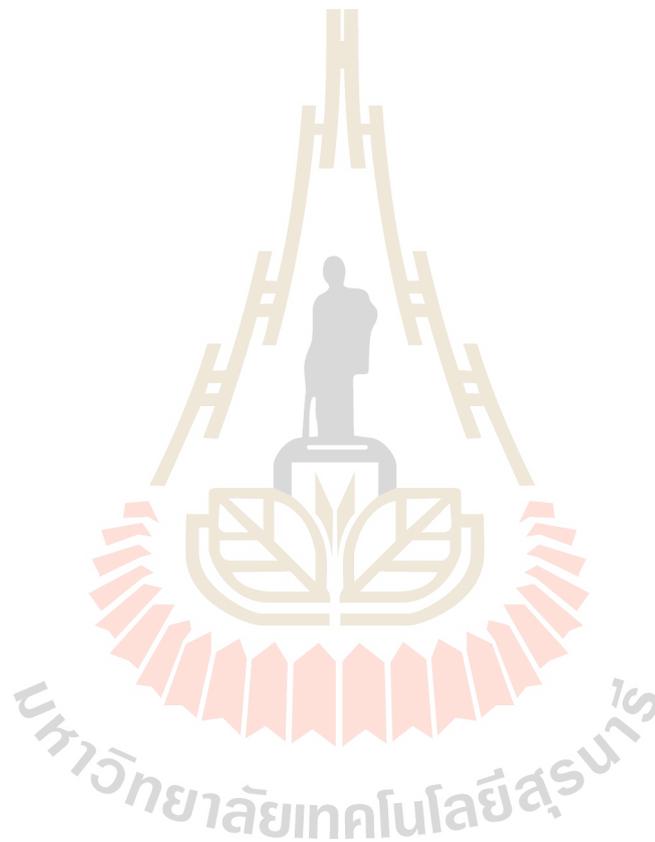
6. คุณสมบัติของหน่วยวัด (Unit Properties)

ตารางที่ 2.6: คุณสมบัติของหน่วยวัด

คุณสมบัติ	ค่า
length	em, ex, px, in, cm, mm, pt, pc
percentage	%

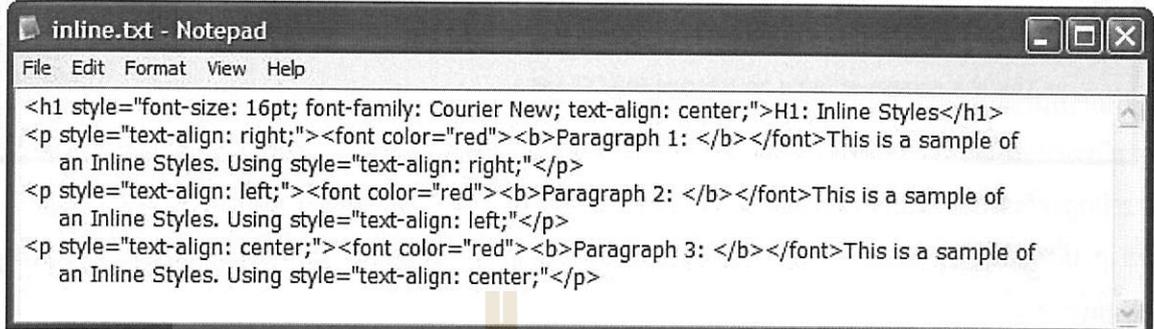
สรุป

การใช้ CSS ช่วยในการออกแบบหน้าเว็บ ทำให้ผู้ออกแบบเว็บสามารถกำหนดตำแหน่งและรูปแบบของการนำเสนอได้อย่างอิสระ สะดวก รวดเร็ว และง่ายต่อการปรับแต่ง โดยสามารถกำหนดลักษณะการแสดงผลของข้อความ การจัดตำแหน่งย่อหน้า (ขอบซ้าย ขวา บน ล่าง) สีของหน้าเว็บ ภาพ ฉากหลัง กรอบ ตาราง ซึ่ง CSS สามารถนำไปแทรกให้อยู่ในส่วนต้นของไฟล์ HTML หรือตำแหน่งอื่น ๆ ก็ได้ตามความต้องการของผู้ออกแบบ การกำหนด CSS ในไฟล์เอกสาร HTML จะทำให้มีผลกับการแสดงผลทั้งหมดในหน้านั้น ๆ แต่ถ้ากำหนด CSS โดยแยกเป็นไฟล์ไว้ต่างหากจากไฟล์เอกสาร HTML จะสามารถนำไฟล์นั้นไปใช้ร่วมกับเอกสารไฟล์ HTML อื่น ๆ ได้อีกด้วย



แบบฝึกหัด

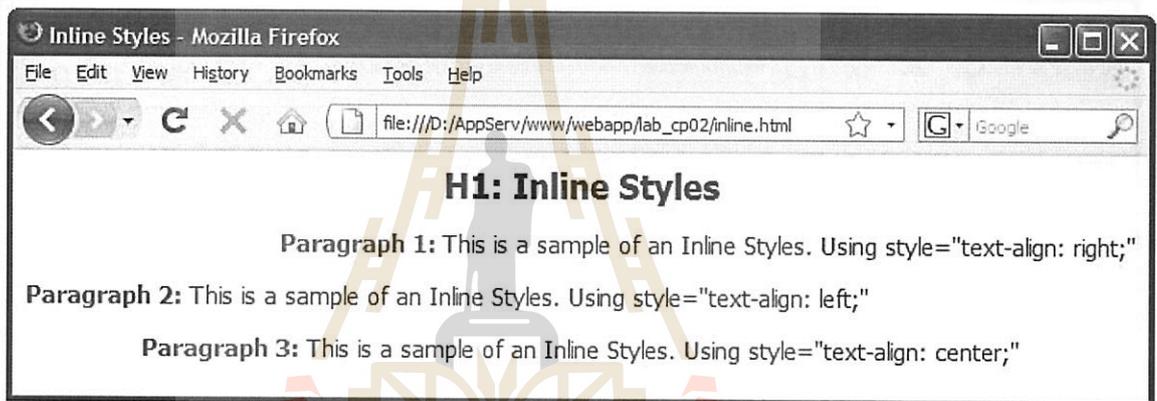
1. ใช้โปรแกรม Notepad สร้าง CSS แบบ Inline Styles (รูปที่ 2.21) เพื่อให้ได้ผลดังในรูปที่ 2.22



```

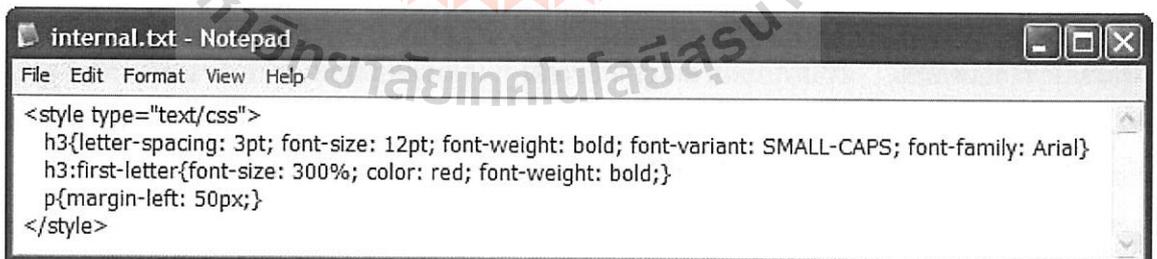
inline.txt - Notepad
File Edit Format View Help
<h1 style="font-size: 16pt; font-family: Courier New; text-align: center;">H1: Inline Styles</h1>
<p style="text-align: right;"><font color="red"><b>Paragraph 1: </b></font>This is a sample of
an Inline Styles. Using style="text-align: right;"</p>
<p style="text-align: left;"><font color="red"><b>Paragraph 2: </b></font>This is a sample of
an Inline Styles. Using style="text-align: left;"</p>
<p style="text-align: center;"><font color="red"><b>Paragraph 3: </b></font>This is a sample of
an Inline Styles. Using style="text-align: center;"</p>
    
```

รูปที่ 2.21: สร้าง CSS แบบ Inline Styles จากโปรแกรม Notepad



รูปที่ 2.22: ผลที่ได้จากคำสั่งในรูปที่ 2.21

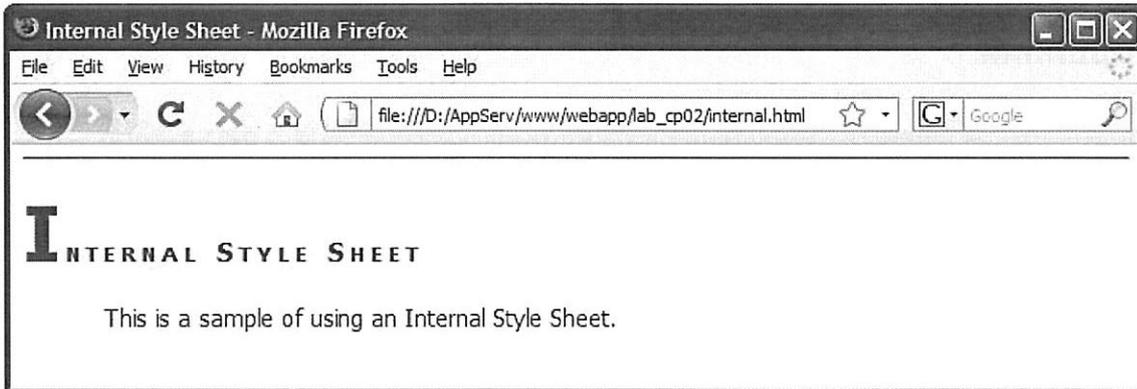
2. ใช้โปรแกรม Notepad สร้าง CSS แบบ Internal Style Sheet (รูปที่ 2.23) เพื่อให้ได้ผลดังในรูปที่ 2.24



```

internal.txt - Notepad
File Edit Format View Help
<style type="text/css">
  h3{letter-spacing: 3pt; font-size: 12pt; font-weight: bold; font-variant: SMALL-CAPS; font-family: Arial}
  h3:first-letter{font-size: 300%; color: red; font-weight: bold;}
  p{margin-left: 50px;}
</style>
    
```

รูปที่ 2.23: สร้าง CSS แบบ Internal Style Sheet จากโปรแกรม Notepad



รูปที่ 2.24: ผลที่ได้จากคำสั่งในรูปที่ 2.23

3. ใช้โปรแกรม Notepad สร้าง CSS แบบ External Style Sheet เพื่อให้ได้ผลดังในรูปที่ 2.25



รูปที่ 2.25: ผลที่ได้จากการใช้คำสั่งแบบ External Style Sheet

4. ติดตั้งโปรแกรม SUTinsServer (ดาวน์โหลด SUTinsServer ได้ที่ <http://linux.sut.ac.th/>) และทดสอบการทำงานของไฟล์ HTML ต่าง ๆ

บทที่ 3 แนะนำภาษา PHP

<Introduction to PHP>

บทนำ

การนำเสนอข้อมูลข่าวสารผ่านอินเทอร์เน็ต โดยใช้ภาษา HTML เป็นตัวนำเสนอผ่านโปรแกรมเว็บเบราว์เซอร์นั้นมีขีดจำกัด เนื่องจากข้อมูลเหล่านั้นเป็นข้อมูลที่คงที่ (หรืออาจเรียกอีกอย่างหนึ่งว่า Static) ซึ่งถูกสร้างขึ้นมาก่อนการนำเสนอ เมื่อมีการเรียกหน้าเว็บนั้นขึ้นมาดู ข้อมูลเหล่านั้นก็จะถูกนำมาเสนอ ถ้าต้องการปรับเปลี่ยนข้อมูลบางอย่าง ผู้ออกแบบเว็บก็ต้องเข้าไปทำการแก้ไขไฟล์ข้อมูลนั้น ๆ ซึ่งเป็นอุปสรรคอย่างหนึ่งที่ไม่สามารถนำเสนอข้อมูลบางอย่างให้ทันต่อเหตุการณ์ ผู้พัฒนาเว็บต่างก็ค้นหาคำตอบที่จะช่วยแก้ไขปัญหาดังกล่าว คำตอบก็มีอยู่หลายตัวเลือก และหนึ่งในนั้นก็คือ ภาษา PHP ซึ่งเป็นภาษาที่นิยมมากสำหรับนักพัฒนาโปรแกรมบนเว็บ (Web Application)

วัตถุประสงค์

1. เพื่อให้รู้ประวัติความเป็นมาของภาษา PHP
2. เพื่อให้เข้าใจหลักการทำงานและการใช้คำสั่งเบื้องต้นของภาษา PHP
3. เพื่อให้เข้าใจความสัมพันธ์ระหว่างแบบฟอร์มในไฟล์ HTML และการนำเสนอข้อมูลในภาษา

PHP

เนื้อหา

3.1 ประวัติและความเป็นมาของภาษา PHP

ภาษา PHP เป็นภาษาที่เกิดมาจากภาษา PHP/FI ซึ่งเป็นกลุ่มของภาษาเพิร์ล สร้างโดย รัสมัส เลอร์ดอร์ฟ (Rasmus Lerdorf) ในปี ค.ศ. 1995 เขาสร้างสคริปต์นี้ขึ้นมาเพื่อใช้ในการตรวจสอบจำนวนการเข้าดูชีวประวัติของเขาที่ได้นำเสนอบนเว็บไว้ โดยเรียกกลุ่มสคริปต์เหล่านี้ว่า เครื่องมือสำหรับโฮมเพจส่วนตัว (Personal Home Page Tool) และต่อมาได้มีการพัฒนาคุณสมบัติในส่วนต่าง ๆ ของเครื่องมือนี้เพิ่มขึ้น โดยใช้ภาษาซีเป็นภาษาหลักในการพัฒนาเพื่อใช้ในการติดต่อกับฐานข้อมูล ซึ่งเอื้ออำนวยให้นักพัฒนาโปรแกรมสามารถสร้างโปรแกรมบนเว็บที่มีลักษณะแตกต่างไปจากเดิม คือจากข้อมูลที่แสดงบนเว็บแบบคงที่ (Static) มาเป็นแบบเคลื่อนไหว (Dynamic) แทน หลังจากนั้น รัสมัส เลอร์ดอร์ฟ ก็ได้เปิดเผยซอร์สโค้ดแก่สาธารณชน เพื่อนำไปใช้ ปรับปรุง แก้ไขและพัฒนาต่อไป หลังจากที่มีการพัฒนาภาษา PHP/FI อย่างต่อเนื่อง เวอร์ชันทดลอง (Beta Version) ต่าง ๆ ได้ออกมาอย่างแพร่หลาย จนกระทั่งในเดือนพฤศจิกายน ปี ค.ศ. 1997 ก็ได้มีการออก PHP/FI 2.0 อย่างเป็นทางการ ซึ่งเป็นรากฐานให้เกิด PHP 3.0 โดยมีผู้พัฒนาร่วมอีก 2 คนคือ แอนดี กัทแมนส์ (Andi Gutmans) และ ซีฟ ซูร์สกี (Zeev Suraski) และนั่นก็คือการหยุดพัฒนา PHP/FI 2.0 โดยสิ้นเชิง

PHP 3.0 มีข้อดีตรงที่มีความหลากหลายในการติดต่อกับฐานข้อมูล (Databases) ชนิดต่าง ๆ รูปแบบการรับ-ส่งข้อมูล (Protocols) และโปรแกรมในการเชื่อมต่อ (Application Program Interface) ต่าง ๆ ได้เป็นอย่างดี ซึ่งต่อมาได้เปลี่ยนความหมายของ PHP จากเดิม Personal Home Page Tool ที่มีข้อจำกัดเกินไป ไปเป็น PHP Hypertext Preprocessor แทน หลังจากได้มีการทดลองใช้ PHP 3.0 อยู่หลายเดือน PHP 3.0 ก็ได้ออกสู่ตลาดอย่างเป็นทางการในเดือนมิถุนายน ปี ค.ศ. 1998 ซึ่งในช่วงนั้นประมาณ 10 เปอร์เซนต์ของเว็บไซต์ต่าง ๆ ที่เชื่อมต่ออินเทอร์เน็ตได้ลง PHP 3.0

หลังจากที่ PHP 3.0 ออกมาได้ไม่นาน ก็ได้มีการพัฒนาเครื่องมือชื่อว่า Zend Engine ซึ่งเป็นชื่อที่มาจากมารวมเอาชื่อของ ซีฟ ซูร์สกี (Zeev Suraski) กับ แอนดี กัทแมนส์ (Andi Gutmans) เข้าด้วยกัน เป็นเครื่องมือที่ช่วยเสริมให้ PHP มีคุณสมบัติเด่น ๆ มากขึ้น เหมาะที่จะใช้เป็นภาษาในการพัฒนาโปรแกรมบนเว็บ ต่อมาในเดือนพฤษภาคม ปี ค.ศ. 2000 PHP 4.0 ก็ได้ออกมาอย่างเป็นทางการ ได้รวม Zend Engine และเพิ่มคุณสมบัติต่าง ๆ ที่สามารถรองรับเซิร์ฟเวอร์ชนิดต่าง ๆ HTTP session, บัฟเฟอร์ส่งออก, เพิ่มประสิทธิภาพและการป้องกันในการรองรับข้อมูลและโครงสร้างของภาษาใหม่ ๆ จากผู้ใช้ ซึ่งต่อมาในเดือนกรกฎาคม ปี ค.ศ. 2000 PHP 5.0 ที่มีการรวม Zend Engine 2.0 ก็ได้ออกสู่ตลาด

3.2 รูปแบบโครงสร้างและหลักการทำงานของภาษา PHP

ภาษา PHP มีลักษณะคล้ายกับภาษาซี ที่คำสั่งแต่ละคำสั่งนั้นจะต้องปิดท้ายด้วยเครื่องหมาย ; (Semicolon) โดยสามารถสอดแทรกกลุ่มของภาษา PHP ไว้ในภาษา HTML ได้ในทุก ๆ ส่วนหรือตำแหน่งที่ต้องการ หรือจะเขียนเฉพาะคำสั่งภาษา PHP อย่างเดียวโดยไม่ต้องมีคำสั่ง HTML เลยก็ได้ แล้วบันทึกเป็นสกุล .php ซึ่งภาษา PHP สามารถเขียนได้ในรูปแบบที่แตกต่างกันถึง 4 รูปแบบดังต่อไปนี้

1. รูปแบบ XML (XML Style)

```
<?php
    PHP Code In Here
?>
```

2. รูปแบบสคริปต์ (Script Style)

```
<script language="php">
    PHP Code In Here
</script>
```

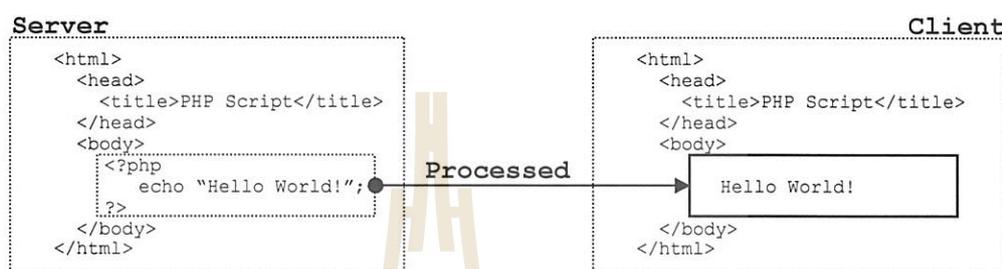
3. รูปแบบสั้น (Short Style)

```
<?
    PHP Code In Here
?>
```

4. รูปแบบ ASP (ASP Style)

```
<%
    PHP Code In Here
%>
```

จากรูปแบบทั้ง 4 นั้น รูปแบบ XML เป็นรูปแบบที่นิยมใช้มากที่สุด และบ่อยครั้งที่ภาษา PHP จะถูกใช้ควบคู่กับภาษา HTML โดยสอดแทรกภาษา PHP ในตำแหน่งที่ต้องการในไฟล์ข้อมูล HTML เมื่อโปรแกรมเว็บเบราว์เซอร์ (Web Browser) จากฝั่งไคลเอนท์ (Client) ส่งไฟล์สกุล .php ไปที่เว็บเซิร์ฟเวอร์ (Web Server) แล้ว เว็บเซิร์ฟเวอร์ก็จะส่งต่อไปที่ PHP Pre-processor เพื่อทำการประมวลผลคำสั่ง PHP ที่พบ แล้วก็จะส่งผลลัพธ์กลับให้กับเว็บเซิร์ฟเวอร์ ตัวเว็บเซิร์ฟเวอร์ก็จะส่งผลลัพธ์นั้นกลับไปเว็บเบราว์เซอร์ในรูปแบบข้อมูล HTML เพื่อแสดงผลในฝั่งไคลเอนท์ ซึ่งสามารถแสดงขั้นตอนการทำงานดังกล่าวได้ดังในรูปที่ 3.1



รูปที่ 3.1: การประมวลผลภาษา PHP ทางฝั่งเซิร์ฟเวอร์และส่งข้อมูล HTML กลับไปที่เครื่องไคลเอนท์

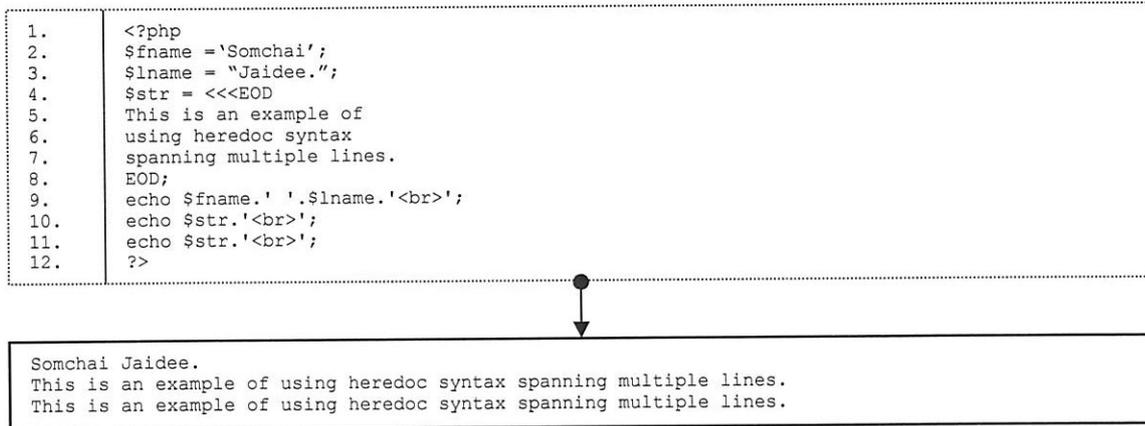
3.3 ชนิดข้อมูล (Data Types)

ภาษา PHP แบ่งชนิดข้อมูลออกเป็น 3 ประเภทด้วยกันคือ

1. ชนิดข้อมูลแบบ Scalar Type ซึ่งมีอยู่ 4 รูปแบบคือ

- Boolean เป็นชนิดข้อมูลที่จะส่งค่ากลับเป็น จริง (True) หรือ เท็จ (False)
- Integer เป็นชนิดข้อมูลของเลขจำนวนเต็ม ในรูปแบบของเลขฐานสิบ (Decimal) เลขฐานแปด (Octal) หรือเลขฐานสิบหก (Hexadecimal)
- Float เป็นชนิดข้อมูลของเลขจำนวนจริง เช่น 3.75, 1.5e2 หรือ 9E-10 เป็นต้น
- String เป็นชนิดข้อมูลของกลุ่มตัวอักษร ที่ไม่มีขอบเขตจำกัด นั่นคือความยาวของสตริงหรือจำนวนของตัวอักษรไม่ถูกจำกัดจำนวน สามารถกำหนดได้ใน 3 รูปแบบคือ

- Single Quoted คือการกำหนดสตริงภายในเครื่องหมาย Single Quote เช่น `$name = 'Jack';` เป็นต้น
- Double Quoted คือการกำหนดสตริงภายในเครื่องหมาย Double Quote เช่น `$str = "ABC";` เป็นต้น
- Heredoc คือการกำหนดสตริงโดยใช้ Heredoc Syntax (“<<<”) ซึ่งมีการทำงานคล้ายกับการใช้ Double Quoted วิธีการกำหนดชนิด Heredoc คือ เริ่มต้นด้วยเครื่องหมาย <<< แล้วตามด้วยชื่อตัวระบุ (Identifier) ตามด้วยข้อความ และปิดท้ายด้วยการขึ้นบรรทัดใหม่แล้วใส่ชื่อตัวระบุและเครื่องหมาย ; (Semicolon) ในคอลัมน์แรก ดังตัวอย่างในรูปที่ 3.2



รูปที่ 3.2: การกำหนดสตริงแบบ Heredoc

บรรทัดที่ 2 เป็นการใช้ Single Quoted ในการกำหนดสตริง Somchai ให้กับตัวแปร \$fname

บรรทัดที่ 3 เป็นการใช้ Double Quoted ในการกำหนดสตริง Jaidee ให้กับตัวแปร \$lname

บรรทัดที่ 4 เป็นการใช้ Heredoc โดยจะเริ่มต้นด้วยเครื่องหมาย <<< แล้วตามด้วยชื่อตัวระบุ (Identifier) ซึ่งในที่นี้ได้ตั้งชื่อว่า EOD

บรรทัดที่ 5-7 เป็นสตริงหรือกลุ่มของข้อความ ที่สามารถเขียนโดยแยกบรรทัดได้ แต่เมื่อแสดงผลจะนำข้อความในบรรทัดใหม่ไปต่อกับบรรทัดก่อนหน้านั้น โดยเพิ่มช่องว่าง 1 ช่อง

บรรทัดที่ 8 เป็นการใช้สิ้นสุด Heredoc โดยจะใช้ชื่อตัวระบุ (Identifier) ซึ่งในที่นี้ได้ตั้งชื่อว่า EOD ตามด้วยเครื่องหมาย ; (Semicolon) โดยชื่อตัวระบุจะต้องอยู่ในตำแหน่งคอลัมน์ที่ 1 เท่านั้น

ชนิดข้อมูลแบบข้อความนี้ ตัวอักษรบางตัวต้องใช้อักขระควบคุมพิเศษในการแสดงผล ตัวอักษรเหล่านี้เรียกว่า Escaped Characters ซึ่งใช้เครื่องหมาย \ (Backslash) นำหน้า ดังตัวอย่างในตารางที่ 3.1

ตารางที่ 3.1: อักขระหลีก (Escaped Characters)

ตัวอักษร	ความหมาย
\n	ใช้เมื่อต้องการขึ้นบรรทัดใหม่ มีผลเมื่อดูที่ซอร์สโค้ดแต่จะไม่มีผลเมื่อแสดงผลบนเว็บเบราว์เซอร์
\r	ใช้เมื่อต้องการให้ Cursor เลื่อนกลับไปตำแหน่งแรก
\t	ใช้เมื่อต้องการให้ Cursor เลื่อนไปอีก 1 แท็บ
\\	ใช้เมื่อต้องการพิมพ์เครื่องหมาย Backslash (\)
\\$	ใช้เมื่อต้องการพิมพ์เครื่องหมาย Dollar (\$)
\"	ใช้เมื่อต้องการพิมพ์เครื่องหมาย Double-Quote (")
\'	ใช้เมื่อต้องการพิมพ์เครื่องหมาย Single-Quote (')
\[0-7]{1,3}	ใช้เมื่อต้องการพิมพ์เลขฐานแปด
\x[0-9A-Fa-f]{1,2}	ใช้เมื่อต้องการพิมพ์เลขฐานสิบหก

ตัวอย่าง การใช้เครื่องหมาย \ (Backslash) นำหน้าตัวอักษรที่เป็น Escaped Characters

```

1. <?php
2. $s1 = 'Students\' Record';
3. $s2 = 'The first "IMPORTANT" statement.';
4. $s3 = 'The second \'IMPORTANT\' statement.';
5. $s4 = "The third \"IMPORTANT\" statement.";
6. echo $s1.'  
';
7. echo $s2.'  
';
8. echo $s3.'  
';
9. echo $s4.'  
';
10. ?>
    
```

```

Students' Record
The first "IMPORTANT" statement.
The second \'IMPORTANT\' statement.
The third "IMPORTANT" statement.
    
```

รูปที่ 3.3: การแสดงตัวอักษรที่เป็น Escaped Characters

บรรทัดที่ 2 เป็นการกำหนดสตริงให้กับตัวแปร \$s1 โดยกำหนดให้สตริงนั้นอยู่ในเครื่องหมาย Single Quoted ซึ่งในสตริงนั้นจะมีเครื่องหมาย Single Quoted อยู่ด้วย ดังนั้น จึงต้องมีการใช้ Escaped Sequence

บรรทัดที่ 3 เป็นการกำหนดสตริงให้กับตัวแปร \$s2 โดยกำหนดให้สตริงนั้นอยู่ในเครื่องหมาย Single Quoted ซึ่งในสตริงนั้นจะมีเครื่องหมาย Double Quoted อยู่ด้วย แต่เมื่อได้กำหนดให้สตริงนั้นอยู่ในเครื่องหมาย Single Quoted ดังนั้น จึงไม่ต้องใช้เครื่องหมาย Backslash นำหน้าเครื่องหมาย Double Quoted

บรรทัดที่ 4 เป็นการกำหนดสตริงให้กับตัวแปร \$s3 โดยกำหนดให้สตริงนั้นอยู่ในเครื่องหมาย Single Quoted ซึ่งในสตริงนั้นจะมีเครื่องหมาย Double Quoted อยู่ด้วย แต่ทดลองใช้ Escaped Sequence โดยใส่เครื่องหมาย Backslash นำหน้าเครื่องหมาย Double Quoted ผลที่ได้ก็คือ เครื่องหมาย Backslash ก็ถูกพิมพ์ออกด้วย

บรรทัดที่ 5 เป็นการกำหนดสตริงให้กับตัวแปร \$s4 โดยกำหนดให้สตริงนั้นอยู่ในเครื่องหมาย Double Quoted ซึ่งในสตริงนั้นจะมีเครื่องหมาย Double Quoted อยู่ด้วย ดังนั้น จึงต้องมีการใช้ Escaped Sequence

2. ชนิดข้อมูลแบบ Compound ซึ่งมีอยู่ 2 รูปแบบคือ

- Array เป็นชนิดข้อมูลที่ถูกจัดเก็บเป็นกลุ่ม โดยใช้ฟังก์ชัน array() เป็นตัวช่วยในการกำหนด เช่น \$parts = array("Leg","Wing","Breast"); ซึ่งจะเกิดตัวแปรทั้งหมด 3 ตัวคือ \$parts[0] เก็บสตริง "Leg", \$parts[1] เก็บสตริง "Wing" และ \$parts[2] เก็บสตริง "Breast" ตามลำดับ ถ้าต้องการกำหนดชนิดข้อมูลที่เป็นช่วงข้อมูล เช่น เลขจำนวนเต็ม 1 ถึง 10 ให้กับตัวแปร \$n สามารถใช้ฟังก์ชัน range() ช่วยในการกำหนด เช่น \$n = range(1,10); ซึ่งมีความหมายเช่นเดียวกันกับ \$n = array(1,2,3,4,5,6,7,8,9,10); ะเรย์อีกรูปแบบหนึ่งเรียกว่าอะเรย์สัมพันธ์ (Associative Array) ซึ่งมีรูปแบบในการกำหนดคือ variable = array(key=>value); เช่น \$prices = array("Leg"=>30, "Wing"=>20, "Breast"=>60); ดังตัวอย่างในรูปที่ 3.4

```

1. <?php
2. $num = array(1,2,3,4,5);
3. $n = range(1,5);
4. $prices = array("Leg"=>30,"Wing"=>20,"Breast"=>60);
5. echo "$num[0]<br>";
6. echo "$n[3]<br>";
7. echo "$prices[Leg]<br>";
8. echo "$prices[Wing]<br>";
9. echo "$prices[Breast]<br>";
10. ?>
    
```

1
4
30
20
60

รูปที่ 3.4: การประกาศตัวแปรแบบ array และ associative array

- Object เป็นการกำหนดชนิดข้อมูลแบบ Object โดยใช้คำว่า new นำหน้าชื่อตัวแปรที่ต้องการกำหนดให้เป็นแบบ Object ดังตัวอย่างในรูปที่ 3.5

```

1. <?php
2. class msg{
3.     function msg1(){
4.         echo "This is message 1.<br>";
5.     }
6.     function msg2(){
7.         echo "This is message 2.<br>";
8.     }
9. }
10. $text = new msg;
11. $text->msg1();
12. $text->msg2();
13. ?>
    
```

This is message 1.
This is message 2.

รูปที่ 3.5: การกำหนดชนิดข้อมูลให้กับตัวแปรแบบ Object

3. ชนิดข้อมูลแบบพิเศษ (Special) ซึ่งมีอยู่ 2 รูปแบบคือ

- Resource เป็นตัวแปรชนิดพิเศษที่เก็บการอ้างอิงถึงแหล่งข้อมูลภายนอก โดยใช้ฟังก์ชันพิเศษเป็นตัวช่วย เช่น `$conn = mysql_connect("localhost","root","meroot");` เป็นต้น
- NULL ตัวแปรที่ถือว่าเป็นชนิด NULL คือตัวแปรที่ถูกกำหนดค่าให้เป็น NULL เช่น `$temp = NULL;` เป็นตัวแปรที่ยังไม่ได้กำหนดค่า เช่น `$num` หรือเป็นตัวแปรที่ถูกใช้ในฟังก์ชัน `unset();` เช่น `unset($total);`

ข้อสังเกต ในการเขียนโปรแกรมภาษาต่าง ๆ นั้น ตัวแปรตัวหนึ่งสามารถเก็บค่าได้เพียงชนิดเดียวเท่านั้น เช่นจะเก็บค่าของเลขจำนวนเต็ม ก็จะเก็บได้เฉพาะเลขจำนวนเต็มเท่านั้น จะเก็บสตริง ก็จะเก็บได้เฉพาะสตริงเท่านั้น แต่ภาษา PHP แตกต่างจากนั้นโดยสิ้นเชิง เพราะตัวแปรในภาษา PHP จะเก็บค่าอะไรก็ขึ้นอยู่กับค่าที่จะกำหนดให้ เช่น

```

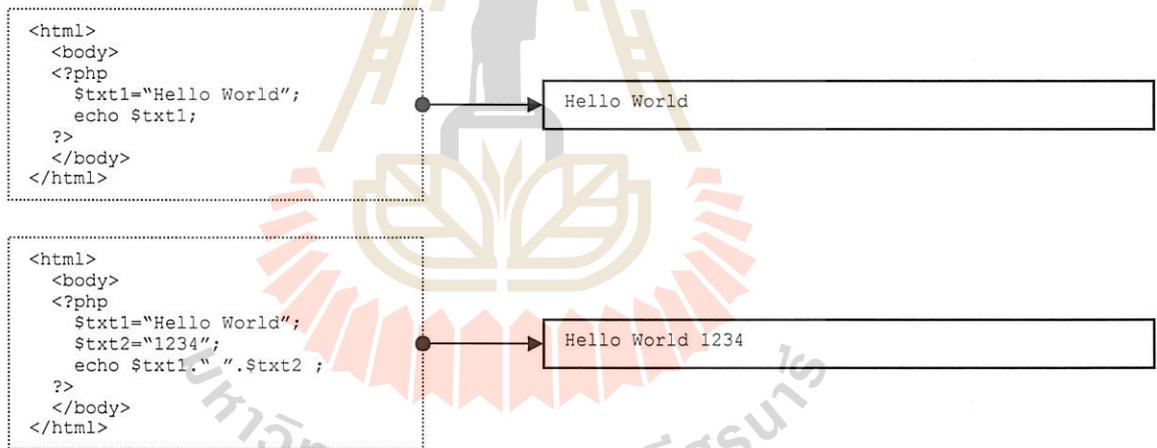
$data = 25;
$data = "SUT-Thailand";
    
```

จะเห็นว่าตัวแปร `$data` ในตอนแรกนั้นเก็บค่า 25 ซึ่งเป็นเลขจำนวนเต็ม ส่วนในบรรทัดต่อไปเก็บค่า SUT-Thailand ซึ่งเป็นสตริง ดังนั้น ตัวแปรใน PHP จะเป็นตัวแปรที่จะเก็บค่าประเภทใดนั้นขึ้นอยู่กับค่าที่มันเก็บอยู่

3.4 ตัวแปร (Variable)

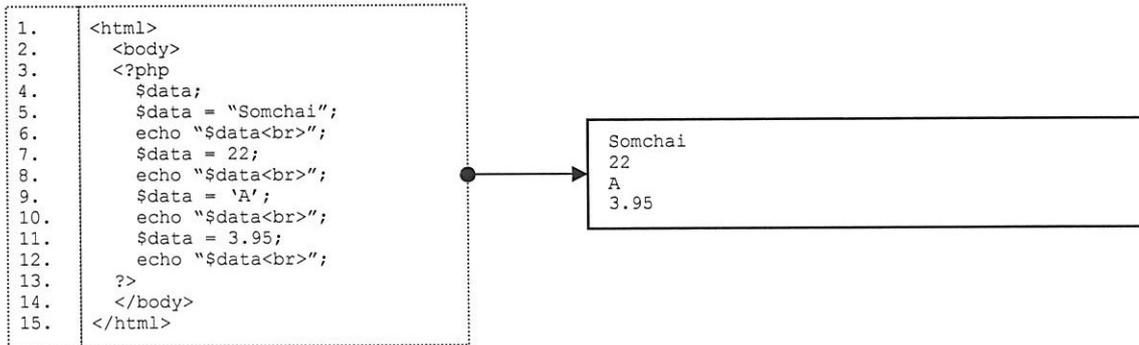
ตัวแปร (Variable) ในภาษา PHP มีข้อกำหนดดังนี้คือ

1. ตัวแปรจะต้องขึ้นต้นด้วยเครื่องหมาย \$ แล้วตามด้วยชื่อตัวแปร เช่น \$name หรือ \$student_age เป็นต้น
2. ชื่อของตัวแปรจะต้องมีความยาวไม่เกิน 255 ตัวอักษร ตัวอักษรตัวแรกจะต้องเป็นตัวอักษร a-z หรือ A-Z หรือเครื่องหมาย _ (Underscore) เขียนติดกันโดยห้ามมีช่องว่างหรือจุดทศนิยม (โดยทั่วไปแล้วไม่นิยมใช้เครื่องหมาย _ (Underscore) เป็นตัวอักษรตัวแรก เนื่องจากคำสงวนหลายตัวที่ใช้เครื่องหมาย _ (Underscore) นำหน้า)
3. ชื่อของตัวแปรจะต้องไม่ตรงกับคำสงวน (Keyword)
4. ชื่อของตัวแปรที่เป็นตัวอักษรตัวเล็ก จะมีชื่อแตกต่างกับชื่อตัวแปรที่เป็นตัวอักษรตัวใหญ่ เช่น ตัวแปร \$name กับตัวแปร \$Name จะเป็นตัวแปรคนละตัวกัน
5. ตัวแปรตัวหนึ่ง สามารถเก็บข้อมูลชนิดใดก็ได้ ขึ้นอยู่กับค่าที่กำหนดให้ เช่น \$name = "Somsri" หรือ \$student_age = 19 เป็นต้น
6. เมื่อต้องการแสดงข้อมูลของตัวแปร 2 ตัวหรือมากกว่า ให้ใช้เครื่องหมาย . (Period) เป็นตัวเชื่อม ดังแสดงในรูปที่ 3.6



รูปที่ 3.6: การกำหนดชื่อตัวแปรและการใช้ . (period) เชื่อมตัวแปร

การประกาศตัวแปรในภาษา PHP นั้น ไม่ต้องระบุชนิดของข้อมูลที่ตัวแปรนั้น ๆ จะเก็บ ตัวแปรที่ถูกประกาศขึ้นมาสามารถเก็บข้อมูลชนิดใดก็ได้ ทั้งนี้ขึ้นอยู่กับวิธีการออกแบบของนักเขียนโปรแกรม ดังตัวอย่างในรูปที่ 3.7



รูปที่ 3.7: การประกาศตัวแปรและการกำหนดค่าชนิดต่าง ๆ ให้กับตัวแปรตัวเดียว

บรรทัดที่ 4 มีการประกาศตัวแปร \$data

บรรทัดที่ 5 มีการกำหนดค่ากลุ่มตัวอักษรหรือสตริงให้กับตัวแปร \$data (ค่าของสตริงจะต้องครอบคลุมด้วยเครื่องหมาย “ ” (Double Quotation Marks)

บรรทัดที่ 6 แสดงค่าของตัวแปร \$data ออกที่จอภาพแล้วขึ้นบรรทัดใหม่

บรรทัดที่ 7 มีการกำหนดค่าจำนวนเต็ม 22 ให้กับตัวแปร \$data

บรรทัดที่ 8 แสดงค่าของตัวแปร \$data ออกที่จอภาพแล้วขึ้นบรรทัดใหม่

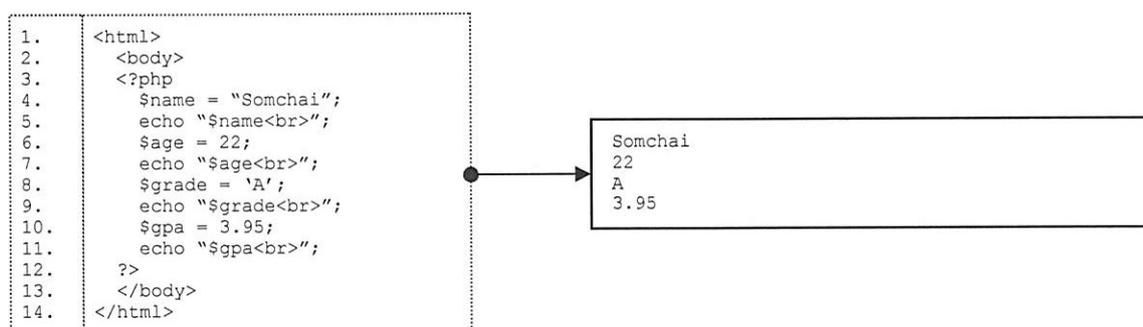
บรรทัดที่ 9 มีการกำหนดค่าตัวอักษร 1 ตัวคือ A ให้กับตัวแปร \$data (ค่าของตัวอักษร 1 ตัวจะต้องครอบคลุมด้วยเครื่องหมาย ‘ ’ (Single Quotation Marks)

บรรทัดที่ 10 แสดงค่าของตัวแปร \$data ออกที่จอภาพแล้วขึ้นบรรทัดใหม่

บรรทัดที่ 11 มีการกำหนดค่าจำนวนจริง 3.95 ให้กับตัวแปร \$data

บรรทัดที่ 12 แสดงค่าของตัวแปร \$data ออกที่จอภาพแล้วขึ้นบรรทัดใหม่

จะสังเกตเห็นว่าตัวแปร \$data สามารถเก็บค่าชนิดข้อมูลแบบใดก็ได้ แต่ไม่ควรใช้ตัวแปรเดียวในการเก็บข้อมูลที่ต่างชนิดกัน เพราะจะทำให้สับสน และการตั้งชื่อตัวแปรนั้นควรจะต้องสอดคล้องกับข้อมูลที่จัดเก็บด้วย ดังแสดงในรูปที่ 3.8



รูปที่ 3.8: การประกาศตัวแปรและการกำหนดค่าชนิดต่าง ๆ ให้กับตัวแปรหลาย ๆ ตัว

ตัวแปรที่ถูกกำหนดไว้ใน PHP เวอร์ชัน 5.0.0 มีหลายชนิด ซึ่งจะขอกล่าวอย่างสังเขปดังต่อไปนี้

3.4.1 ตัวแปรระบบ (Server Variables)

เป็นตัวแปรที่ระบบได้กำหนดไว้ ดังตัวอย่างในตารางที่ 3.2 ซึ่งเป็นเพียงส่วนหนึ่งของตัวแปรระบบเท่านั้น ในเวอร์ชัน 4.1.0 ใช้ \$HTTP_SERVER_VARS แต่ในเวอร์ชัน 5.0.0 ได้เปลี่ยนเป็น \$_SERVER ซึ่งเป็นอะเรย์ที่เก็บรายละเอียดข้อมูลส่วนหัว (Header) เส้นทาง (Path) และตำแหน่งของสคริปต์ต่าง ๆ ซึ่งข้อมูลเหล่านี้จะถูกสร้างขึ้นโดยเว็บเบราว์เซอร์ เป็นตัวแปรแบบ Superglobal และแบบ Global อัตโนมัติที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์ โดยเซิร์ฟเวอร์แต่ละตัวก็อาจจะมีรายละเอียดที่แตกต่างกันออกไป

ตารางที่ 3.2: ตัวแปรระบบ

ตัวแปร	ความหมาย
SERVER_PROTOCOL	ใช้เมื่อต้องการให้แสดงชื่อโปรโตคอล
SERVER_SOFTWARE	ใช้เมื่อต้องการให้แสดงชื่อของซอฟต์แวร์ที่ใช้ในการบริการเว็บ
SERVER_NAME	ใช้เมื่อต้องการให้แสดงชื่อของเซิร์ฟเวอร์
SERVER_PORT	ใช้เมื่อต้องการให้แสดงหมายเลขพอร์ตที่ใช้ใช้ติดต่อกับเซิร์ฟเวอร์
REQUEST_METHOD	ใช้เมื่อต้องการให้แสดงรูปแบบการรับ-ส่ง (GET หรือ POST)
DOCUMENT_ROOT	ใช้เมื่อต้องการให้แสดงเส้นทาง (Path) ของตำแหน่งต้นทางของเอกสาร
HTTP_ACCEPT_CHARSET	ใช้เมื่อต้องการให้แสดงกลุ่มของตัวอักษรที่รองรับ
HTTP_ACCEPT_LANGUAGE	ใช้เมื่อต้องการให้แสดงภาษาที่รองรับ
HTTP_CONNECTION	ใช้เมื่อต้องการให้แสดงสถานะภาพในการติดต่อ
REMOTE_ADDR	ใช้เมื่อต้องการให้แสดงที่อยู่เป็นหมายเลข ip ของเซิร์ฟเวอร์

ตัวอย่าง การใช้ตัวแปรระบบ

```

1.  <?php
2.     echo 'SERVER_PROTOCOL = ' . $SERVER_PROTOCOL . '<br>';
3.     echo 'SERVER_SOFTWARE = ' . $SERVER_SOFTWARE . '<br>';
4.     echo 'SERVER_NAME = ' . $SERVER_NAME . '<br>';
5.     echo 'SERVER_PORT = ' . $SERVER_PORT . '<br>';
6.     echo 'REQUEST_METHOD = ' . $REQUEST_METHOD . '<br>';
7.     echo 'DOCUMENT_ROOT = ' . $DOCUMENT_ROOT . '<br>';
8.     echo 'HTTP_ACCEPT_CHARSET = ' . $HTTP_ACCEPT_CHARSET . '<br>';
9.     echo 'HTTP_ACCEPT_LANGUAGE = ' . $HTTP_ACCEPT_LANGUAGE . '<br>';
10.    echo 'REMOTE_ADDR = ' . $REMOTE_ADDR . '<br>';
11.  ?>
    
```

```

SERVER_PROTOCOL = HTTP/1.1
SERVER_SOFTWARE = Apache/2.2.4 (Win32) PHP/5.2.1
SERVER_NAME = localhost
SERVER_PORT = 80
REQUEST_METHOD = GET
DOCUMENT_ROOT = D:/AppServ/www
HTTP_ACCEPT_CHARSET = TIS-620,utf-8;q=0.7,*;q=0.7
HTTP_ACCEPT_LANGUAGE = en-us,en;q=0.7,th;q=0.3
HTTP_CONNECTION = keep-alive
REMOTE_ADDR = 127.0.0.1
    
```

รูปที่ 3.9: ตัวอย่างการใช้ตัวแปรระบบ

3.4.2 ตัวแปรแวดล้อม (Environment Variables)

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 ซึ่งใช้ `$HTTP_ENV_VARS` แต่ในเวอร์ชัน 5 ได้เปลี่ยนเป็น `$_ENV` เป็นตัวแปรจาก PHP Parser ที่กำลังทำงานอยู่ ให้ถูกบรรจุอยู่ในพื้นที่ที่เก็บชื่อของตัวแปรแบบ Global ของ PHP ซึ่งเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.3 ตัวแปรคุกกี้ (HTTP Cookies)

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 ซึ่งใช้ `$HTTP_COOKIE_VARS` แต่ในเวอร์ชัน 5 ได้เปลี่ยนเป็น `$_COOKIE` เป็นตัวแปรแบบอะเรย์สัมพันธ์ (Associative Array) ที่เก็บตัวแปรที่มีการส่งในสคริปต์นั้น ๆ ในรูปแบบ HTTP Cookies โดยเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.4 ตัวแปรแบบ GET (HTTP GET Variables)

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 ซึ่งใช้ `$HTTP_GET_VARS` แต่ในเวอร์ชัน 5 ได้เปลี่ยนเป็น `$_GET` เป็นตัวแปรแบบอะเรย์สัมพันธ์ ที่เก็บตัวแปรที่มีการส่งในสคริปต์นั้น ๆ ในรูปแบบ HTTP GET โดยเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.5 ตัวแปรแบบ POST (HTTP POST Variables)

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 ซึ่งใช้ `$HTTP_POST_VARS` แต่ในเวอร์ชัน 5 ได้เปลี่ยนเป็น `$_POST` เป็นตัวแปรแบบอะเรย์สัมพันธ์ ที่เก็บตัวแปรที่มีการส่งในสคริปต์นั้น ๆ ในรูปแบบ HTTP POST โดยเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.6 ตัวแปรแบบ HTTP File Upload Variables

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 ซึ่งใช้ `$HTTP_POST_FILES` แต่ในเวอร์ชัน 5 ได้เปลี่ยนเป็น `$_FILES` เป็นตัวแปรแบบอะเรย์สัมพันธ์ ที่ประกอบไปด้วยชื่อหรือหัวข้อที่มีการอัปโหลดในสคริปต์นั้น ๆ ในรูปแบบ HTTP POST โดยเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.7 ตัวแปรแบบ Request (Request Variables)

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 โดยใช้ `$_REQUEST` เป็นตัวแปรแบบอะเรย์สัมพันธ์ ที่ประกอบไปด้วยข้อความที่เกี่ยวข้องกับ `$_GET`, `$_POST` และ `$_COOKIE` โดยเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.8 ตัวแปรแบบเซสชัน (Session Variables)

เป็นตัวแปรที่ถูกแนะนำในเวอร์ชัน 4.1.0 ซึ่งใช้ `$HTTP_SESSION_VARS` แต่ในเวอร์ชัน 5 ได้เปลี่ยนเป็น `$_SESSION` เป็นตัวแปรแบบอะเรย์สัมพันธ์ที่เก็บตัวแปรแบบเซสชัน เพื่อใช้ในสคริปต์นั้น ๆ โดยเป็นแบบ Superglobal และแบบ Global อัดโนมิตีที่สามารถเข้าถึงได้จากทุก ๆ สคริปต์

3.4.9 ตัวแปรแบบ Global (Global Variables)

เป็นตัวแปรที่ถูกแนะนำตั้งแต่เวอร์ชัน 3.0.0 แบบอะเรย์สัมพันธ์ ที่เก็บแหล่งอ้างอิงของตัวแปรที่มีขอบเขตแบบ Global

3.4.10 ตัวแปรแบบ The previous error message

เป็นตัวแปร ๆ หนึ่งชื่อว่า \$php_errormsg ที่เก็บข้อความการผิดพลาดที่สร้างขึ้นจาก PHP

3.4.11 ขอบเขตของตัวแปร (Variable Scope)

การประกาศตัวแปรในสคริปต์ PHP นั้นถูกแบ่งออกเป็น 2 แบบคือ แบบ global และแบบ local ซึ่งมีข้อแตกต่างดังต่อไปนี้

- แบบ global เป็นการประกาศตัวแปรที่อยู่ภายในสคริปต์

```
<?php
$a = 9; /* global scope*/
echo $a;
?>
```

รูปที่ 3.10: ตัวอย่างตัวแปรแบบ global

ถ้ามีการสร้างฟังก์ชันขึ้นมา การที่จะเข้าถึงข้อมูลตัวแปรแบบ global ได้นั้น ภายในฟังก์ชันเองจะต้องทำการประกาศตัวแปร โดยมีคำสั่ง global นำหน้าชื่อตัวแปร และชื่อตัวแปรที่กำหนดขึ้นมานั้นจะต้องเป็นชื่อเดียวกันด้วย

```
<?php
$a = 9; /* global scope */
function test(){
    global $a; /* reference to global scope*/
    echo $a;
}
test();
?>
```

รูปที่ 3.11: ตัวอย่างการเข้าถึงตัวแปรแบบ global จากฟังก์ชัน

- แบบ local เป็นการประกาศตัวแปรที่อยู่ภายในฟังก์ชัน สามารถเข้าถึงข้อมูลได้เฉพาะเมื่ออยู่ในฟังก์ชันเท่านั้น

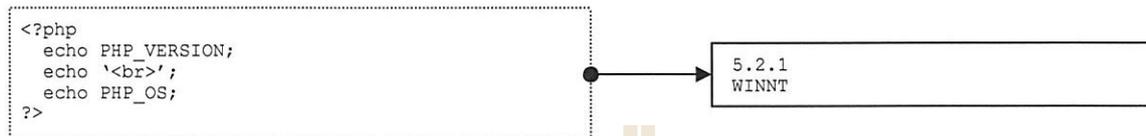
```
<?php
function test(){
    $a = 20; /* local scope */
    echo "Inside function: $a<br>";
}
test();
echo "Outside function: $a";
?>
```

รูปที่ 3.12: ตัวอย่างตัวแปรแบบ local

3.5 ค่าคงที่ (Constant)

ค่าคงที่ คือ ชื่อหรือตัวที่ระบุถึงค่าหนึ่ง ๆ ซึ่งเป็นค่าที่ไม่สามารถเปลี่ยนแปลงได้ในขณะที่โปรแกรมกำลังทำงาน สามารถแบ่งออกเป็น 2 ประเภทคือ

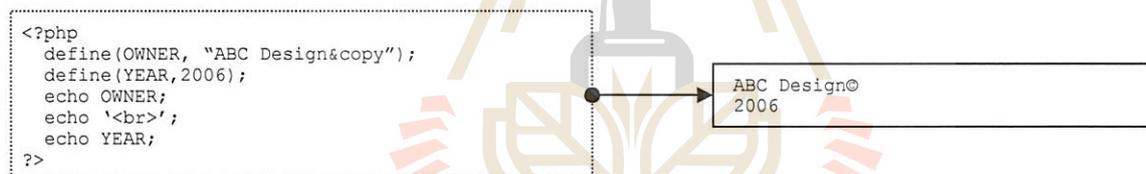
1. ค่าคงที่ที่ถูกกำหนดให้โดย PHP (Predefined Constants) เป็นค่าคงที่ที่สามารถนำมาใช้ได้เลย เช่น PHP_VERSION จะทำการแสดงชื่อเวอร์ชันของ PHP ที่กำลังใช้งานอยู่ และ PHP_OS จะทำการแสดงชื่อระบบปฏิบัติการที่ใช้ PHP เป็นต้น ดังตัวอย่างในรูปที่ 3.13



รูปที่ 3.13: ตัวอย่างค่าคงที่ที่ถูกกำหนดโดย PHP

2. ค่าคงที่ที่ผู้ใช้กำหนดเอง (User-defined Constants) เป็นค่าคงที่ที่ผู้เขียนโปรแกรมสามารถกำหนดขึ้นได้ โดยใช้ฟังก์ชัน define() การเขียนโปรแกรมส่วนใหญ่แล้วมักนิยมการกำหนดค่าคงที่ไว้ในส่วนหัวของโปรแกรมและมักจะนิยมตั้งชื่อที่จะเก็บค่าคงที่นั้นโดยใช้ตัวอักษรใหญ่ทั้งหมด มีรูปแบบในการกำหนดค่าคงที่ดังนี้

define(name, value);

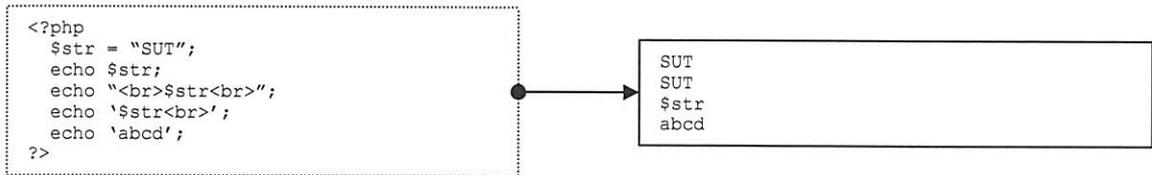


รูปที่ 3.14: ตัวอย่างค่าคงที่ที่ถูกกำหนดโดยผู้ใช้งาน

3.6 คำสั่งเบื้องต้นของภาษา PHP

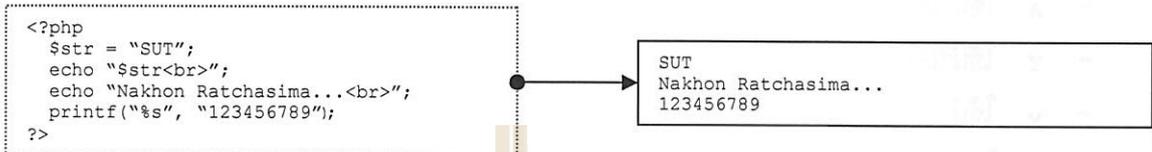
คำสั่งเบื้องต้นที่ควรทำความรู้จักมีดังต่อไปนี้

1. คำสั่งแต่ละคำสั่งจะสิ้นสุดด้วยเครื่องหมาย ; (Semicolon) เสมอ
2. คำสั่ง echo เป็นคำสั่งที่ใช้ในการแสดงข้อมูลออกที่หน้าจอภาพ ถ้าข้อมูลนั้นเป็นข้อความทั่วไปหรือเป็นแท็ก HTML สามารถใช้เครื่องหมาย ‘..’ (Single Quotation Marks) ครอบข้อความหรือแท็ก HTML ได้ แต่ถ้าต้องการแสดงข้อมูลที่ถูกเก็บอยู่ในตัวแปร ต้องใช้เครื่องหมาย “...” (Double Quotation Marks) ครอบข้อความตัวแปรนั้น ๆ ซึ่งจะมีการประมวลผลก่อนการแสดงผลที่หน้าจอภาพ ดังนั้นวิธีการหลีกเลี่ยงข้อผิดพลาดในการแสดงผล จึงควรใช้เครื่องหมาย “...” (Double Quotation Marks) แทนเครื่องหมาย ‘..’ (Single Quotation Marks) ดังตัวอย่างในรูปที่ 3.15



รูปที่ 3.15: การใช้เครื่องหมาย ‘...’ และ “...” ในคำสั่ง echo

3. คำสั่ง printf() เป็นคำสั่งที่ใช้ในการแสดงข้อมูลออกที่หน้าจอภาพเช่นเดียวกับคำสั่ง echo ดังตัวอย่างในรูปที่ 3.16



รูปที่ 3.16: การใช้คำสั่ง printf

4. คำสั่ง // ใช้เมื่อต้องการเขียนข้อสั่งเกต (Comment) บรรทัดเดียว โดยใช้เครื่องหมาย // (Two Slash Signs) นำหน้าข้อสั่งเกตนั้น ๆ แต่ถ้าต้องการเขียนข้อสั่งเกตที่มีความยาวหลายบรรทัด เราสามารถใช้เครื่องหมาย /* (Slash and Star Signs) ตามด้วยข้อสั่งเกต และปิดท้ายด้วยเครื่องหมาย */ (Star Slash Signs) ในบรรทัดสุดท้าย ดังตัวอย่างในรูปที่ 3.17



รูปที่ 3.17: การใช้ข้อสั่งเกตในภาษา PHP

5. ฟังก์ชัน date()

การเรียกใช้ฟังก์ชัน date() นั้นจะช่วยให้การนำเสนอและจัดเก็บข้อมูลเป็นลักษณะแบบ Dynamic ซึ่งมีตัวเลือกหลายตัวที่สามารถเลือกให้เหมาะสมกับความต้องการของผู้ออกแบบได้ เช่น

- h ให้นำเสนอชั่วโมงให้เป็นรูปแบบ 12 ชั่วโมง โดยมีเลขศูนย์นำหน้า (01 ถึง 12)
- H ให้นำเสนอชั่วโมงให้เป็นรูปแบบ 24 ชั่วโมง โดยมีเลขศูนย์นำหน้า (00 ถึง 23)
- i ให้นำเสนอนาที โดยมีเลขศูนย์นำหน้า (00 ถึง 59)
- s ให้นำเสนอวินาที โดยมีเลขศูนย์นำหน้า (00 ถึง 59)
- j ให้นำเสนอวันที่ของเดือน โดยไม่มีเลขศูนย์นำหน้า (1 ถึง 31)
- d ให้นำเสนอวันที่ของเดือน โดยมีเลขศูนย์นำหน้า (01 ถึง 31)
- D ให้นำเสนอวันของสัปดาห์ 3 ตัวอักษร เช่น Mon, Tue, ..., และ Sun
- l (ตัวอักษรแอลเล็ก) ให้นำเสนอวันของสัปดาห์แบบเต็ม เช่น Monday, Tuesday เป็นต้น

- s ให้นำเสนอตัวต่อท้าย (st, nd, rd หรือ th) ของวันที่ของเดือน เช่น 1st, 2nd, 3rd, 4th เป็นต้น
- F ให้นำเสนอชื่อเดือนแบบเต็ม เช่น January, February, ..., และ December
- M ให้นำเสนอชื่อเดือนแบบย่อ 3 ตัวอักษร เช่น Jan, Feb, ..., และ Dec
- m ให้นำเสนอชื่อเดือนแบบตัวเลขโดยมีเลขศูนย์นำหน้า (01 ถึง 12)
- n ให้นำเสนอชื่อเดือนแบบตัวเลขโดยไม่มีเลขศูนย์นำหน้า (1 ถึง 12)
- a ให้นำเสนอช่วงเวลาแบบตัวอักษรตัวเล็ก (am หรือ pm)
- A ให้นำเสนอช่วงเวลาแบบตัวอักษรตัวใหญ่ (AM หรือ PM)
- Y ให้นำเสนอปีค.ศ. 4 ตัว เช่น 2006, 2007, 2008 เป็นต้น
- y ให้นำเสนอปีค.ศ. 2 ตัวหลัง เช่น 06, 07, 08 เป็นต้น
- Z ให้นำเสนอวันของปี เริ่มต้นจาก 0 ถึง 365
- t ให้นำเสนอจำนวนวันของเดือน
- w ให้นำเสนอตัวเลขของวันในสัปดาห์ (0-6)

<pre><?php echo "Order Processed...
"; echo date("H:i, jS F, Y."); ?></pre>		<pre>Order Processed... 05:37, 12th February, 2007.</pre>
--	--	---

รูปที่ 3.18: การเรียกใช้ฟังก์ชัน date()

3.7 แบบฟอร์มในภาษา HTML

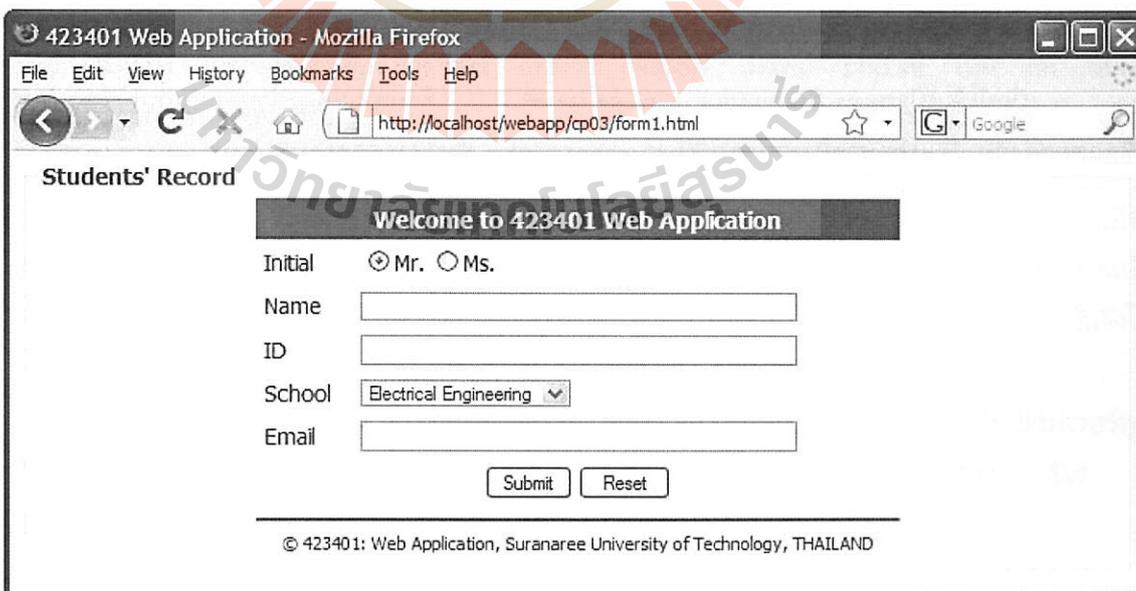
การรับข้อมูลจากผู้ใช้ (User) บนหน้าเว็บเพจนั้นต้องอาศัยแบบฟอร์มในภาษา HTML เป็นตัวกลาง ซึ่งผู้ออกแบบสามารถกำหนดรูปแบบและขนาดของข้อมูล ชื่อตัวแปรที่จะเก็บข้อมูลนั้น ๆ รวมไปถึงชนิดของการส่งข้อมูลไปยังเซิร์ฟเวอร์ด้วย ดังตัวอย่างในรูปที่ 3.19

```

1. <html>
2.   <head>
3.     <title>423401 Web Application</title>
4.     <meta http-equiv="content-type" content="text/html; charset=tis-620">
5.   </head>
6.   <body><center>
7.     <form action="process.php" method="post">
8.       <fieldset><legend><font color="#0000FF"><b>Students' Record</b></font> </legend>
9.       <table width="60%" border="0" cellpadding="5" cellspacing="0">
10.        <tr bgcolor= "#0066ff">
11.          <td colspan="2" align="center"><font color="white"><b>Welcome to 423401 Web
12. Application</b></font></td></tr>
13.        <tr>
14.          <td>Initial</td>
15.          <td align="left">
16.            <input type="radio" name="initial" value="Mr" checked>Mr.
17.            <input type="radio" name="initial" value="Ms">Ms.</td></tr>
18.        <tr>
19.          <td>Name</td>
20.          <td align="left">
21.            <input type="text" name="name" size="50" maxlength="50"></td></tr>
22.        <tr>
23.          <td>ID</td>
24.          <td align="left">
25.            <input type="text" name="id" size="50" maxlength="50"></td></tr>
26.        <tr>
27.          <td>School</td>
28.          <td><select name="school">
29.            <option value="1">Electrical Engineering
30.            <option value="2">Computer Engineering
31.            <option value="3">Civil Engineering</select></td></tr>
32.        <tr>
33.          <td>Email</td>
34.          <td align="left">
35.            <input type="text" name="email" size="50" maxlength="50"></td></tr>
36.        <tr>
37.          <td colspan = "2" align="center">
38.            <input type="submit" name="submit" value ="Submit">
39.            <input type="reset" name="reset" value="&nbsp;Reset&nbsp;"></td></tr>
40.        </table>
41.        <hr width="60%" color="red">
42.        <font size="-1">&copy; 423401: Web Application, Suranaree University of Technology,
43. THAILAND</font>
44.      </fieldset></form></center>
45.   </body>
46. </html>

```

รูปที่ 3.19: การสร้างฟอร์มในไฟล์ form1.html เพื่อรับข้อมูลจากผู้ใช้



รูปที่ 3.20: การแสดงผลของไฟล์ form1.html ในรูปที่ 3.19

บรรทัดที่ 7 เป็นการกำหนดแบบฟอร์ม โดยมีการกำหนดคุณสมบัติดังนี้

`action="process.php"` คือการเรียกไฟล์ `process.php` เมื่อผู้ใช้กดปุ่ม `submit`
`method="post"` คือการส่งข้อมูลในแบบฟอร์มไปที่ไฟล์ `process.php` แบบ `post` ซึ่ง
 การส่งในลักษณะนี้จะไม่แสดงชื่อตัวแปรและค่าในหน้าต่าง `url` ส่วนอีกลักษณะหนึ่งคือแบบ `get` ซึ่งจะแสดง
 ชื่อตัวแปรและค่าทั้งหมดในหน้าต่าง `url`

บรรทัดที่ 16 เป็นการกำหนดลักษณะของข้อมูลหรือที่เรียกว่า `input` โดยมีการกำหนดคุณสมบัติ
 ดังนี้

`type="radio"` คือการกำหนดข้อมูลมีลักษณะเป็นตัวเลือกแบบปุ่มวงกลม
`name="initial"` คือการกำหนดตัวแปรชื่อว่า `initial` ที่จะเก็บค่าที่ผู้ใช้เลือก
`value="Mr"` คือการกำหนดตัวเลือกนี้ให้มีค่าเท่ากับ `Mr`
`checked` คือการกำหนดให้ตัวเลือกนี้ได้ถูกเลือกไว้เป็นค่าปริยาย (`default`)

บรรทัดที่ 21 เป็นการกำหนดลักษณะของข้อมูลหรือที่เรียกว่า `input` โดยมีการกำหนดคุณสมบัติ
 ดังนี้

`type="text"` คือการกำหนดข้อมูลมีลักษณะเป็นข้อความ
`name="name"` คือการกำหนดตัวแปรชื่อว่า `name`
`size="50"` คือการกำหนดขนาดกล่องให้มีขนาดความยาวเท่ากับ 50 ตัวอักษร
`maxlength="50"` คือการกำหนดขนาดความยาวสูงสุดของจำนวนตัวอักษรไว้ที่ 50

ตัวอักษร

บรรทัดที่ 28 เป็นการกำหนดลักษณะของข้อมูลแบบตัวเลือก โดยมีการกำหนดคุณสมบัติดังนี้

`name="school"` คือการกำหนดตัวแปรชื่อว่า `school` ที่จะเก็บค่าที่ผู้ใช้เลือกตัวใดตัว
 หนึ่ง โดยตัวแปร `school` จะเก็บค่า 1 ถ้าผู้ใช้เลือก `Electrical Engineering` จะเก็บค่า 2 ถ้าผู้ใช้
 เลือก `Computer Engineering` หรือจะเก็บค่า 3 ถ้าผู้ใช้เลือก `Civil Engineering`

เมื่อผู้ใช้ได้ป้อนและเลือกข้อมูลต่าง ๆ ดังในรูปที่ 3.21 ข้อมูลในแต่ละส่วนจะถูกเก็บไว้ที่ตัวแปรที่มี
 ชื่อแตกต่างกัน ตัวแปรเหล่านี้จะถูกส่งไปที่ไฟล์ `process.php` ทันทีที่ผู้ใช้กดปุ่ม `Submit`

The screenshot shows a Mozilla Firefox browser window titled "423401 Web Application - Mozilla Firefox". The address bar shows "http://localhost/webapp/cp03/form1.html". The page content includes a header "Welcome to 423401 Web Application" and a form with the following fields: "Initial" with radio buttons for "Mr." (selected) and "Ms."; "Name" with a text input containing "Somchai Jaidee"; "ID" with a text input containing "B4871234"; "School" with a dropdown menu showing "Computer Engineering"; and "Email" with a text input containing "somchai@yahoo.com". Below the form are "Submit" and "Reset" buttons. At the bottom, it says "© 423401: Web Application, Suranaree University of Technology, THAILAND".

รูปที่ 3.21: การป้อนข้อมูลในแบบฟอร์มของไฟล์ form1.html

การรับข้อมูลที่ถูกเก็บไว้ในตัวแปรต่าง ๆ ในแบบฟอร์มนั้นมี 3 วิธีด้วยกัน (ทั้งนี้ขึ้นอยู่กับเวอร์ชันของ PHP และการตั้งค่าในแฟ้มข้อมูล php.ini) โดยวิธีทั้ง 3 นั้นไม่มีชื่ออย่างเป็นทางการ ดังนั้น จึงขอเรียกว่า แบบสั้น (Short Style), แบบกลาง (Medium Style) และแบบยาว (Long Style) สมมติว่าในแบบฟอร์มมีการกำหนดรูปแบบในการรับรหัสประจำตัวนักศึกษา ดังนี้

```
<input type="text" name="id" size="50" maxlength="50">
```

การรับข้อมูลจากแบบฟอร์มนี้สามารถทำได้โดย

1. แบบสั้น (Short Style) เป็นแบบที่สามารถเรียกใช้ได้อย่างสะดวกที่สุด นั่นคือสามารถกำหนดตัวแปรขึ้นมาให้มีชื่อเช่นเดียวกับการกำหนดชื่อในแบบฟอร์ม ก็จะได้ค่าหรือข้อมูลที่ผู้ใช้ได้ป้อนเข้ามาโดยอัตโนมัติ นั่นคือสามารถนำค่าที่อยู่ในตัวแปรนั้นไปแสดงผลหรือประมวลผลได้ตามต้องการ เช่น echo 'ID: '.\$id; เป็นต้น วิธีนี้จะใช้ได้ก็ต่อเมื่อค่าของ register_globals ในแฟ้ม php.ini เป็น on นั่นคือ register_globals = on ซึ่งวิธีการเช่นนี้อาจทำให้เกิดข้อผิดพลาดแล้วเป็นเหตุให้โค้ดนั้นไม่ปลอดภัยเสี่ยงต่อการถูกดัดข้อมูลได้

2. แบบกลาง (Medium Style) เป็นแบบที่สามารถเรียกใช้ได้อย่างสะดวกเช่นกัน โดยก่อนที่จะเรียกใช้ตัวแปรใด ๆ นั้น จะต้องใช้คำสั่งในการรับค่าของตัวแปรเข้ามาก่อน เช่น ต้องการรับค่าให้กับตัวแปร \$id ซึ่งมีการส่งเข้ามาแบบ Post สามารถทำได้โดยการใช้คำสั่ง \$id = \$_POST['id']; หลังจากนั้นจึงใช้ตัวแปร \$id ตามต้องการ รูปแบบนี้ใช้ได้กับ PHP ตั้งแต่เวอร์ชัน 4.1.0 เป็นต้นไป

3. แบบยาว (Long Style) เป็นแบบที่ปลอดภัยที่สุดที่ควรใช้ เพราะทำงานได้กับเซิร์ฟเวอร์ทุกชนิด โดยไม่ต้องห่วงเรื่องการตั้งค่าของเซิร์ฟเวอร์ต่าง ๆ สามารถทำได้โดยใช้คำสั่ง \$id = \$_HTTP_POST_VARS['id'];

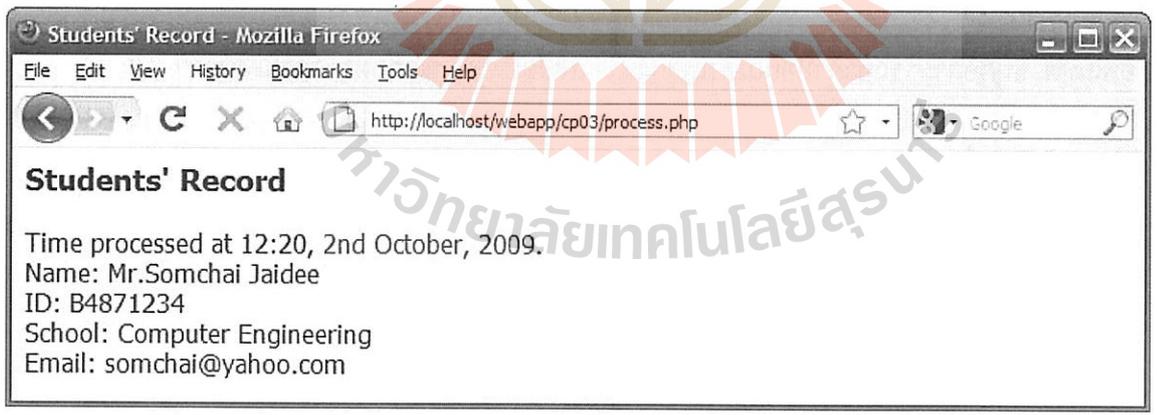
ไฟล์ process.php (ดังในรูปที่ 3.22) จะทำการรับข้อมูลต่าง ๆ ที่ส่งมาจากแบบฟอร์ม (form1.html) แล้วแสดงข้อมูลต่าง ๆ เหล่านั้นออกที่จอภาพ จะสังเกตเห็นว่าในช่อง url นั้นแสดง http://localhost/php_book/cp3/process.php เท่านั้น (ดังในรูปที่ 3.23) แต่ถ้าใน

แบบฟอร์มได้เปลี่ยนจาก method="post" เป็น method="get" ผลที่จะแสดงออกที่จอภาพจะเหมือนเดิม แต่ในช่อง url นั้นจะแสดง

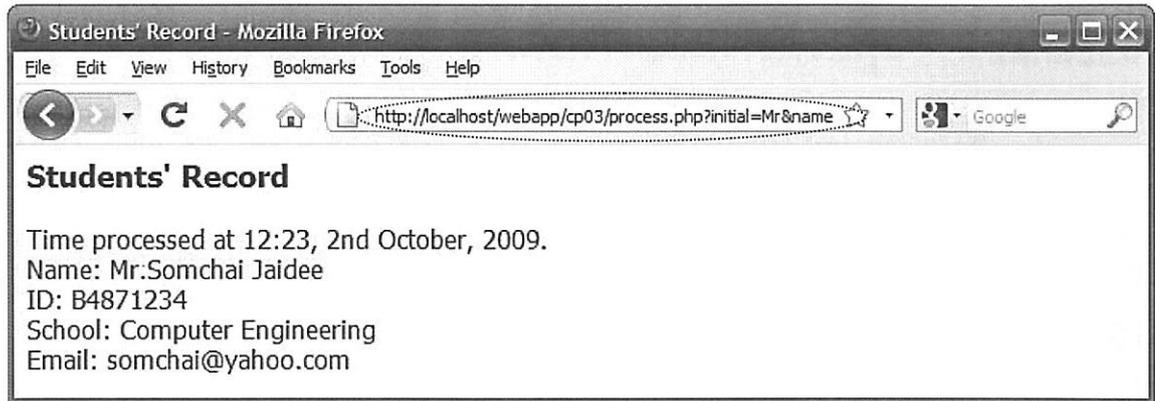
http://localhost/webapp/cp03/process.php?initial=Mr&name=Somchai+Jaidee&id=B4871234&school=2&email=somchai@yahoo.com&submit=Submit (ดังในรูปที่ 3.24)

```
<html>
<head>
  <title>Students' Record</title>
</head>
<body>
<h3>Students' Record</h3>
<?php
  $initial = $_POST['initial'];
  $name = $_POST['name'];
  $id = $_POST['id'];
  $s = $_POST['school'];
  $email = $_POST['email'];
  $date = date("H:i, jS F, Y.");
  switch ($school){
    case '1' :
      $s = "Electrical Engineering";
      break;
    case '2' :
      $s = "Computer Engineering";
      break;
    case '3' :
      $s = "Civil Engineering";
      break;
  }
  echo "Time processed at ";
  echo "$date<br>";
  echo "Name: $initial.$name<br>";
  echo "ID: $id<br>";
  echo "School: $s<br>";
  echo "Email: $email";
?>
</body>
</html>
```

รูปที่ 3.22: คำสั่งต่าง ๆ ในการแสดงผลของไฟล์ process.php



รูปที่ 3.23: การแสดงผลของไฟล์ process.php ที่รับข้อมูลมาจากไฟล์ form1.html ที่ส่งมาแบบ post



รูปที่ 3.24: การแสดงผลของไฟล์ process.php ที่รับข้อมูลมาจากไฟล์ form1.html ที่ส่งมาแบบ get

ไฟล์ process.php ได้แสดงข้อมูลต่าง ๆ ที่รับมาจากแบบฟอร์มในไฟล์ form1.html ออกที่จอภาพ ซึ่งโดยส่วนใหญ่แล้ว เมื่อรับข้อมูลเข้ามาก็จะทำการบันทึกลงแฟ้มข้อมูลหรือในระบบฐานข้อมูล ซึ่งสามารถนำมาประมวลผลในภายหลังได้ แต่ในเบื้องต้นนี้ ได้เน้นความสำคัญเกี่ยวกับการใช้งานของคำสั่งต่าง ๆ ในภาษา PHP และการบันทึกข้อมูลลงแฟ้มข้อมูล ซึ่งในบทท้าย ๆ เมื่อกล่าวถึงระบบฐานข้อมูล จะมีการใช้คำสั่งในภาษา PHP เพื่อบันทึกข้อมูลลงระบบฐานข้อมูล เข้าสู่ระบบฐานข้อมูลเพื่อทำการประมวลผลและนำเสนอผลต่าง ๆ

สรุป

คำสั่งในภาษา PHP ใช้ในการรับข้อมูลจากแบบฟอร์ม แล้วทำการประมวลผลและแสดงออกที่จอภาพ ที่สามารถแทรกอยู่ในตำแหน่งต่าง ๆ ของไฟล์ HTML ได้ตามต้องการ หรือจะเขียนเฉพาะคำสั่งภาษา PHP อย่างเดียวโดยไม่ต้องมีคำสั่ง HTML เลยก็ได้ แต่จะต้องทำการบันทึกไฟล์นั้นเป็นสกุล .php ซึ่งไฟล์ PHP นี้จะทำการประมวลผลต่าง ๆ ที่เว็บเซิร์ฟเวอร์ (Web Server) แล้วเว็บเซิร์ฟเวอร์ก็จะส่งต่อไปที่ PHP Pre-processor เพื่อทำการประมวลผลคำสั่ง PHP ที่พบ แล้วก็ส่งผลลัพธ์กลับไปให้กับเว็บเซิร์ฟเวอร์ ตัวเว็บเซิร์ฟเวอร์ก็จะส่งผลลัพธ์นั้นกลับไปเว็บเบราว์เซอร์ในรูปแบบข้อมูล HTML เพื่อแสดงผลในฝั่งไคลเอนท์

แบบฝึกหัด

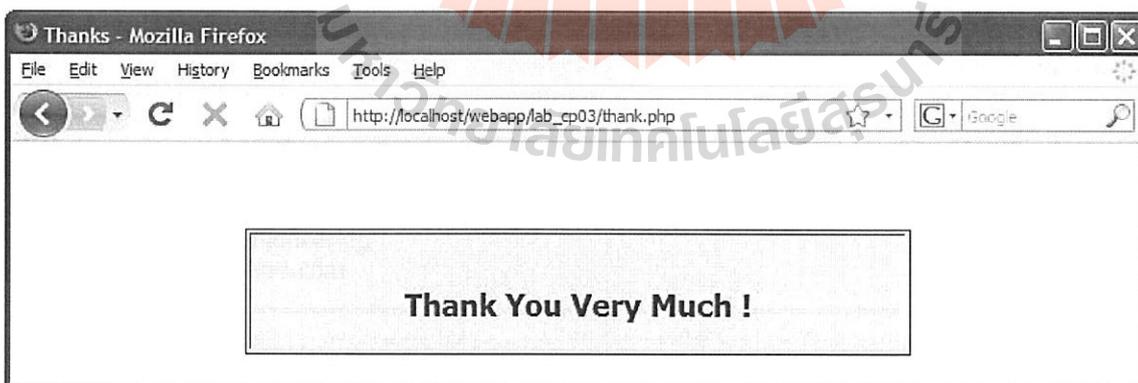
1. ใช้โปรแกรม Text Editor สร้างไฟล์ html ชื่อว่า "restaurant_survey.html" เช่นตัวอย่างดังรูปที่ 3.25

Item	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	N/A
The restaurant was clean	<input type="radio"/>					
The food was reasonably priced	<input type="radio"/>					
The menu offered foods that I like	<input type="radio"/>					
The quality of the food was good	<input type="radio"/>					
The quality of the service was good	<input type="radio"/>					
The staff was polite	<input type="radio"/>					

Submit

รูปที่ 3.25: ผลการแสดงผลของไฟล์ restaurant_survey.html

- เมื่อผู้ใช้ป้อนข้อมูลและกดปุ่ม Submit ก็จะมีส่งข้อมูลทั้งหมดไปที่ไฟล์ thank.php เพื่อแสดงผลดังในรูปที่ 3.26



รูปที่ 3.26: การแสดงผลของไฟล์ thank.php

2. ทดลองออกแบบโดยใช้โปรแกรม Text Editor สร้างฟอร์มในรูปแบบต่าง ๆ ที่มี input เป็น text, radio, checkbox, password, file, hidden, submit, reset, image และ button

บทที่ 4 ตัวดำเนินการและการควบคุม

<Operators and Controls>

บทนำ

ในการเขียนโปรแกรมหนึ่ง ๆ นั้น สิ่งที่เราคาดไม่ถึงก็คือการกำหนดตัวแปรเพื่อเก็บค่าต่าง ๆ แล้วนำไปประมวลผล ซึ่งในการประมวลผลนั้นต้องอาศัยตัวดำเนินการเป็นตัวช่วยเพื่อให้ได้ผลตามที่ต้องการ ตัวดำเนินการนั้นมีอยู่หลายชนิด แต่ละชนิดมีลำดับความสำคัญไม่เท่ากัน จึงจำเป็นต้องศึกษาเพื่อให้เข้าใจถึงหลักการทำงานและลำดับความสำคัญของตัวดำเนินการต่าง ๆ ทั้งนี้เพื่อให้ได้ผลลัพธ์ที่ถูกต้องและเชื่อถือได้อีกทั้งยังต้องศึกษาในเรื่องของการควบคุม ไม่ว่าจะเป็นการควบคุมการนำเสนอข้อมูลหรือการควบคุมการประมวลผลข้อมูล ซึ่งการควบคุมจะช่วยให้อำนวยความสะดวกให้ตัวโปรแกรมทำงานได้อย่างมีประสิทธิภาพ

วัตถุประสงค์

1. เพื่อให้รู้ชนิดและหลักการทำงานของตัวดำเนินการ
2. เพื่อให้เข้าใจรูปแบบและหลักการทำงานของการควบคุม
3. สามารถใช้ตัวดำเนินการและการควบคุมได้

เนื้อหา

4.1 ตัวดำเนินการ (Operators)

ตัวดำเนินการในภาษา PHP ถูกแบ่งออกเป็น 3 ชนิดใหญ่ ๆ คือ

1. ตัวดำเนินการที่ต้องการตัวที่ถูกกระทำ (Operand) เพียง 1 ตัวเท่านั้น เรียกว่า Unary Operators เช่น $\$total++$, $\$k--$, และ $-\$num$; เป็นต้น
2. ตัวดำเนินการที่ต้องการตัวที่ถูกกระทำ (Operand) 2 ตัว เรียกว่า Binary Operators เช่น $\$sum = \$a + \$b$ และ $\$area = (\$base * \$height) / 2$ เป็นต้น
3. ตัวดำเนินการที่ต้องการตัวที่ถูกกระทำ (Operand) 3 ตัว เรียกว่า Ternary Operators ซึ่งเป็นการกำหนดค่าให้กับตัวแปรที่อยู่ทางด้านซ้ายมือของเครื่องหมายเท่ากับโดยมีเงื่อนไข บางครั้งเรียกว่า Conditional Operators เช่น $\$max = (\$x > \$y) ? \$x : \$y$ เป็นต้น

4.1.1 ตัวดำเนินการคณิตศาสตร์ (Arithmetic Operators)

ตารางที่ 4.1: ตัวดำเนินการทางคณิตศาสตร์

กำหนดให้ \$n1 = 19; \$n2 = 4;		
ตัวกระทำ	ตัวอย่าง	ความหมาย
+	\$ans = \$n1+\$n2;	เอาค่า n1 มาบวกค่า n2 แล้วนำผลลัพธ์ที่ได้คือ 23 ไปเก็บไว้ที่ตัวแปร ans
-	\$ans = \$n1-\$n2;	เอาค่า n1 ตั้งลบด้วยค่า n2 แล้วนำผลลัพธ์ที่ได้คือ 15 ไปเก็บไว้ที่ตัวแปร ans
*	\$ans = \$n1*\$n2;	เอาค่า n1 คูณด้วยค่า n2 แล้วนำผลลัพธ์ที่ได้คือ 76 ไปเก็บไว้ที่ตัวแปร ans
/	\$ans = \$n1/\$n2;	เอาค่า n1 ตั้งหารด้วยค่า n2 แล้วนำผลลัพธ์ที่ได้คือ 4.75 ไปเก็บไว้ที่ตัวแปร ans
%	\$ans = \$n1%n2;	เอาค่า n1 ตั้งหารด้วยค่า n2 แล้วนำเศษที่เหลือคือ 3 ไปเก็บไว้ที่ตัวแปร ans

4.1.2 ตัวดำเนินการกำหนดค่า (Assignment Operators)

ตัวดำเนินการกำหนดค่าใช้เพื่อกำหนดค่าให้กับตัวแปรต่าง ๆ สามารถทำได้โดยใช้เครื่องหมาย = (Equal Sign) เช่น \$a = 5; คือการกำหนดค่าที่อยู่ทางด้านขวามือของเครื่องหมายเท่ากับ ซึ่งในที่นี้ก็คือค่า 5 ให้กับตัวแปรที่อยู่ทางด้านซ้ายมือของเครื่องหมายเท่ากับ ซึ่งในที่นี้ก็คือตัวแปร \$a นั่นเอง ตัวดำเนินการในการกำหนดค่านี้ยังสามารถนำไปใช้ร่วมกับตัวดำเนินการทางคณิตศาสตร์ได้ ดังแสดงในตารางที่ 4.2

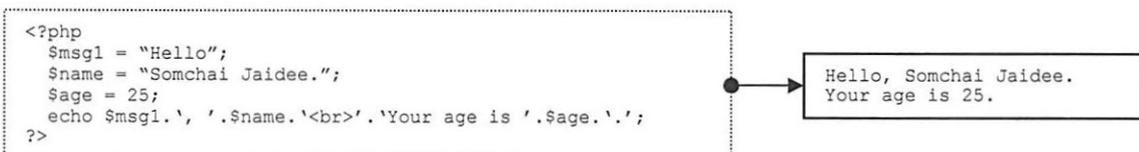
ตารางที่ 4.2: ตัวดำเนินการกำหนดค่า

ตัวกระทำ	ตัวอย่าง	ความหมาย
=	\$num = 5;	กำหนดค่า 5 ให้กับตัวแปรทางด้านซ้ายมือของเครื่องหมาย = ซึ่งก็คือตัวแปร num นั่นเอง
+=	\$num += 5;	มีความหมายเช่นเดียวกันกับ \$num = \$num+5; คือเอาค่าที่อยู่ในตัวแปร num มาบวกกับ 5 แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ที่ตัวแปรทางด้านซ้ายมือของเครื่องหมาย = ซึ่งก็คือตัวแปร num นั่นเอง
-=	\$num -= 5;	มีความหมายเช่นเดียวกันกับ \$num = \$num-5; คือเอาค่าที่อยู่ในตัวแปร num มาลบด้วย 5 แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ที่ตัวแปรทางด้านซ้ายมือของเครื่องหมาย = ซึ่งก็คือตัวแปร num นั่นเอง
*=	\$num *= 5;	มีความหมายเช่นเดียวกันกับ \$num = \$num*5; คือเอาค่าที่อยู่ในตัวแปร num มาคูณกับ 5 แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ที่ตัวแปรทางด้านซ้ายมือของเครื่องหมาย = ซึ่งก็คือตัวแปร num นั่นเอง
/=	\$num /= 5;	มีความหมายเช่นเดียวกันกับ \$num = \$num/5; คือเอาค่าที่อยู่ในตัวแปร num มาหารด้วย 5 แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ที่ตัวแปรทางด้านซ้ายมือของเครื่องหมาย = ซึ่งก็คือตัวแปร num นั่นเอง
%=	\$num %= 5;	มีความหมายเช่นเดียวกันกับ \$num = \$num%5; คือเอาค่าที่อยู่ในตัวแปร num มาหารด้วย 5 แล้วนำเศษที่เหลือไปเก็บไว้ที่ตัวแปรทางด้านซ้ายมือของเครื่องหมาย = ซึ่งก็คือตัวแปร num นั่นเอง

4.1.3 ตัวดำเนินการสตริง (String Operators)

การใช้คำสั่ง echo ในการแสดงผลข้อมูลของตัวแปรนั้น โดยปกติแล้วจะทำการแสดงผลของตัวแปรเดี่ยว ๆ เท่านั้น เช่น echo \$a; หรือ echo "The answer is \$result"; เป็นต้น ถ้าต้องการแสดงผลข้อมูลของตัวแปรหลาย ๆ ตัว ก็ต้องใช้คำสั่ง echo หลาย ๆ ครั้ง ซึ่งทำให้เสียเวลา ดังนั้นตัวดำเนินการสตริงจะเป็นตัวช่วยในการจัดการได้เป็นอย่างดี โดยแบ่งออกเป็น 2 ลักษณะคือ

1. ตัวเชื่อมต่อสตริง ใช้เพื่อแสดงผลข้อมูลของหลาย ๆ ตัวแปรในคราวเดียว โดยใช้เครื่องหมาย . (Period) เป็นตัวเชื่อมต่อ ดังตัวอย่างในรูปที่ 4.1



รูปที่ 4.1: การแสดงผลของข้อมูลที่อยู่ในตัวแปรหลาย ๆ ตัวโดยใช้เครื่องหมายจุดทศนิยม (.)

2. ตัวเชื่อมต่อสตริงแล้วกำหนดค่าสตริงที่เชื่อมเข้าด้วยกันนั้นให้กับตัวแปร โดยใช้เครื่องหมาย (Period) และเครื่องหมาย = (Equal Sign) ติดกัน ดังตัวอย่างในรูปที่ 4.2

```
<?php
$msg1 = "Hello";
$name = "Somchai Jaidee.";
$age = 25;
echo $msg1.' , '.$name.'  
'.'Your age is '.$age.'  
';
$output = $msg1;
$output .= ',';
$output .= $name;
$output .= '<br>';
$output .= 'Your age is ';
$output .= $age;
$output .= '.';
echo $output;
?>
```

Hello, Somchai Jaidee.
Your age is 25.
Hello, Somchai Jaidee.
Your age is 25.

รูปที่ 4.2: การใช้เครื่องหมายจุดทศนิยม (.) ร่วมกับเครื่องหมายเท่ากับ (=)

4.1.4 ตัวดำเนินการเพิ่มค่าและลดค่า (Increment and Decrement Operators)

ตัวดำเนินการที่ใช้ในการเพิ่มค่า (++) และลดค่า (--) ใช้เมื่อต้องการเพิ่มค่าให้กับตัวแปรอีก 1 หรือลดค่าให้กับตัวแปรลง 1 โดยตัวดำเนินการเพิ่มค่าและลดค่านี้สามารถวางอยู่ในตำแหน่งหน้าหรือหลังตัวแปรก็ได้ ถ้าวางอยู่หน้าตัวแปรเรียกว่า Prefix ถ้าวางอยู่หลังตัวแปรเรียกว่า Postfix

ตารางที่ 4.3: ตัวดำเนินการเพิ่มค่าและลดค่า

คำสั่ง	ความหมาย
\$num = 5; \$num++;	เป็น Postfix มีความหมายเช่นเดียวกับ \$num = \$num + 1; ทำให้ตัวแปร num มีค่าเท่ากับ 6
\$num = 5; ++\$num;	เป็น Prefix มีความหมายเช่นเดียวกับ \$num = \$num + 1; ทำให้ตัวแปร num มีค่าเท่ากับ 6
\$num = 5; \$num--;	เป็น Postfix มีความหมายเช่นเดียวกับ \$num = \$num - 1; ทำให้ตัวแปร num มีค่าเท่ากับ 4
\$num = 5; --\$num;	เป็น Prefix มีความหมายเช่นเดียวกับ \$num = \$num - 1; ทำให้ตัวแปร num มีค่าเท่ากับ 4

จากตัวอย่างในตารางที่ 4.3 นั้น จะสังเกตเห็นว่าการเพิ่มค่าโดยใช้เครื่องหมาย ++ ไว้ข้างหน้าหรือข้างหลังตัวแปร ผลที่ได้ก็เท่ากัน หรือการลดค่าโดยใช้เครื่องหมาย -- ไว้ข้างหน้าหรือข้างหลังตัวแปร ผลที่ได้ก็เท่ากัน ดังแสดงในรูปที่ 4.3 และ 4.4

```
<?php
$a = 5;
$b = 10;
echo '$a = '.$a.', and $b = '.$b.'  
';
$a++;
$b--;
echo 'After $a++; and $b--;<br>';
echo '$a = '.$a.', and $b = '.$b.'  
';
?>
```

\$a = 5, and \$b = 10.
After \$a++; and \$b--;
\$a = 6, and \$b = 9.

รูปที่ 4.3: การเพิ่มค่าให้กับตัวแปรแบบ Postfix

```
<?php
$a = 5;
$b = 10;
echo '$a = '.$a.', and $b = '.$b.'  
';
++$a;
--$b;
echo 'After ++$a; and --$b;<br>';
echo '$a = '.$a.', and $b = '.$b.'  
';
?>
```

\$a = 5, and \$b = 10.
After ++\$a; and --\$b;
\$a = 6, and \$b = 9.

รูปที่ 4.4: การเพิ่มค่าให้กับตัวแปรแบบ Prefix

ตอนนี้ทดลองสังเกตผลการทำงานของคำสั่งในรูปที่ 4.5 ซึ่งค่าของตัวแปร \$sum มีค่าเท่ากับ 16 สาเหตุที่เป็นเช่นนั้นเพราะว่า

บรรทัดที่ 5 มีการเพิ่มค่าให้กับตัวแปร \$a แบบ Prefix นั่นคือทำการเพิ่มค่าทันที ค่าของตัวแปร \$a จึงกลายเป็น 6 ส่วนการลดค่าให้กับตัวแปร \$b นั้นเป็นแบบ Postfix นั่นคือจะทำการลดค่าหลังจากที่เอาค่าดั้งเดิมไปประมวลผลก่อน ซึ่งก็คือค่า 10 นั่นเอง หลังจากทำการประมวลผลแล้ว ค่าของตัวแปร \$b จึงกลายเป็น 9

```

1. <?php
2.     $a = 5;
3.     $b = 10;
4.     echo '$a = ' . $a . ', and $b = ' . $b . ' . <br>';
5.     $sum = ++$a + $b--;
6.     echo '$sum = ' . $sum . ' . <br>';
7.     echo '$a = ' . $a . ', and $b = ' . $b . ' . <br>';
8. ?>
    
```

\$a = 5, and \$b = 10.
\$sum = 16
\$a = 6, and \$b = 9.

รูปที่ 4.5: การนำค่าไปใช้เมื่อมีการเพิ่มค่าและลดค่าให้กับตัวแปรแบบ Prefix และ Postfix

ดังนั้นจึงสามารถสรุปได้ว่า การเพิ่มค่าหรือลดค่าให้กับตัวแปรใด ๆ โดยที่ยังไม่ได้เอาค่าของตัวแปรนั้น ๆ ไปใช้ ไม่ว่าจะแบบ Prefix หรือ Postfix จะมีผลเหมือนกัน ส่วนการเพิ่มค่าหรือลดค่าให้กับตัวแปรใด ๆ ที่จะต้องเอาค่าของตัวแปรนั้น ๆ ไปประมวลผล ถ้าเป็นแบบ Prefix (อยู่ข้างหน้าตัวแปร) จะทำการเพิ่มค่าหรือลดค่าทันทีก่อนที่จะนำค่านั้นไปประมวลผล แต่ถ้าเป็นแบบ Postfix (อยู่ข้างหลังตัวแปร) จะเอาค่านั้นไปประมวลผลก่อนแล้วจึงทำการเพิ่มค่าหรือลดค่า

4.1.5 ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

ตัวดำเนินการเปรียบเทียบใช้เพื่อเปรียบเทียบค่าระหว่างตัวแปรหรือนิพจน์ ซึ่งค่าที่ได้จะเป็นจริง (1) หรือเท็จ (0) เท่านั้น

ตารางที่ 4.4: ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	การใช้งาน	ผลที่ได้
==	\$a == \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าเท่ากับ \$b
===	\$a === \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าเท่ากับ \$b และเป็นตัวแปรชนิดเดียวกัน
!=	\$a != \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าไม่เท่ากับ \$b
<>	\$a <> \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าไม่เท่ากับ \$b
!==	\$a !== \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าไม่เท่ากับ \$b หรือเป็นตัวแปรต่างชนิดกัน
<	\$a < \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าน้อยกว่า \$b
>	\$a > \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่ามากกว่า \$b
<=	\$a <= \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่าน้อยกว่าหรือเท่ากับ \$b
>=	\$a >= \$b	จะเป็นจริงก็ต่อเมื่อ ค่าของ \$a มีค่ามากกว่าหรือเท่ากับ \$b

4.1.6 ตัวดำเนินการเปรียบเทียบบิต (Bitwise Operators)

ตัวดำเนินการเปรียบเทียบบิตใช้เพื่อเปรียบเทียบหรือประมวลผลตัวเลขจำนวนเต็มในระดับบิต โดยการเปลี่ยนค่าตัวเลขจำนวนเต็มที่ถูกกำหนดให้กับตัวแปร (ซึ่งโดยปกติแล้วตัวแปรต่าง ๆ จะถูกกำหนดให้เก็บค่าตัวเลขที่เป็นตัวเลขฐานสิบ) ให้อยู่ในรูปแบบของเลขฐานสองแล้วจึงนำมาเปรียบเทียบ เช่น กำหนดค่าให้กับตัวแปร \$a มีค่าเท่ากับ 9 และกำหนดค่าให้กับตัวแปร \$b มีค่าเท่ากับ 10 จะได้รูปแบบ

ของเลขฐานสองดังตารางที่ 4.5 แต่ถ้าเป็นสตริง การเปรียบเทียบจะทำการนำเอาค่าแอสกีของอักขรเหล่านั้นมาเปรียบเทียบกัน

ตารางที่ 4.5: การแทนค่าในรูปแบบของเลขฐานสอง

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	9	0	0	0	0	1	0	0	1
\$b	10	0	0	0	0	1	0	1	0

ตัวดำเนินการเปรียบเทียบบิตมีทั้งหมด 6 ตัวดำเนินการ ดังแสดงในตารางที่ 4.6

ตารางที่ 4.6: ตัวดำเนินการเปรียบเทียบบิต

ตัวดำเนินการ	การใช้งาน	ผลที่ได้
&	\$a & \$b	บิตที่เป็นเซตของทั้ง \$a และ \$b
	\$a \$b	บิตที่เป็นเซตของ \$a หรือ \$b
^	\$a ^ \$b	บิตที่เป็นเซตของ \$a หรือ \$b อันใดอันหนึ่งเท่านั้น
~	~\$a	บิตที่ไม่เป็นเซตของ \$a
<<	\$a << \$b	ขยับบิต \$a ไปทางซ้ายมือจำนวน \$b ครั้ง
>>	\$a >> \$b	ขยับบิต \$a ไปทางขวามือจำนวน \$b ครั้ง

เมื่อเปรียบเทียบค่าของตัวแปร \$a และ \$b โดยใช้ตัวดำเนินการเปรียบเทียบบิตต่าง ๆ จะได้ผลลัพธ์ดังในรูปที่ 4.6

```

1. <?php
2.   $a = 9; $b = 10; $c = -20;
3.   echo '$a = ' . $a . ', $b = ' . $b . ', $c = ' . $c . '<br>';
4.   echo '$a & $b = ' . ($a & $b) . '<br>';
5.   echo '$a | $b = ' . ($a | $b) . '<br>';
6.   echo '$a ^ $b = ' . ($a ^ $b) . '<br>';
7.   echo '~$a & $b = ' . (~$a & $b) . '<br>';
8.   echo '$a & ~$b = ' . ($a & ~$b) . '<br>';
9.   echo '~$a & ~$b = ' . (~( $a ) & (~$b)) . '<br>';
10.  echo '~$a = ' . (~$a) . '<br>';
11.  echo '~$b = ' . (~$b) . '<br>';
12.  echo '~$c = ' . (~$c) . '<br>';
13.  echo '$a << 2 = ' . ($a << 2) . '<br>';
14.  echo '$a >> 2 = ' . ($a >> 2) . '<br>';
15.  ?>
                
```

```

$a = 9, $b = 10, $c = -20
$a & $b = 8
$a | $b = 11
$a ^ $b = 3
~$a & $b = 2
$a & ~$b = 1
~$a & ~$b = -12
~$a = -10
~$b = -11
~$c = 19
$a << 2 = 36
$a >> 2 = 2
                
```

รูปที่ 4.6: ผลของการใช้ตัวดำเนินการเปรียบเทียบบิต

บรรทัดที่ 3 เป็นการแสดงผลของค่าตัวแปร \$a, \$b และ \$c ออกที่จอภาพ ซึ่งก็คือค่า 9, 10 และ -20 ตามลำดับ

บรรทัดที่ 4 เป็นการแสดงผลของ \$a & \$b นั่นคือค่าของบิตที่เป็นเซตของทั้ง \$a และ \$b ซึ่งเมื่อเขียนค่าของตัวแปรทั้ง 2 ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.7

ตารางที่ 4.7: การเปรียบเทียบบิต \$a & \$b

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	9	0	0	0	0	1	0	0	1
\$b	10	0	0	0	0	1	0	1	0
\$a & \$b	8	0	0	0	0	1	0	0	0

ผลลัพธ์ที่ได้คือ 00001000 เป็นค่าบวก นั่นคือ $(1 * 2^3) = 8$

บรรทัดที่ 5 เป็นการแสดงผลของ \$a | \$b นั่นคือค่าของบิตที่เป็นเซตของ \$a หรือ \$b ซึ่งเมื่อเขียนค่าของตัวแปรทั้ง 2 ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.8

ตารางที่ 4.8: การเปรียบเทียบบิต \$a | \$b

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	9	0	0	0	0	1	0	0	1
\$b	10	0	0	0	0	1	0	1	0
\$a \$b	11	0	0	0	0	1	0	1	1

ผลลัพธ์ที่ได้คือ 00001011 เป็นค่าบวก นั่นคือ $(1 * 2^3) + (1 * 2^1) + (1 * 2^0) = 8 + 2 + 1 = 11$

บรรทัดที่ 6 เป็นการแสดงผลของ \$a ^ \$b นั่นคือค่าของบิตที่เป็นเซตของ \$a หรือ \$b อันใดอันหนึ่งเท่านั้น ซึ่งเมื่อเขียนค่าของตัวแปรทั้ง 2 ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.9

ตารางที่ 4.9: การเปรียบเทียบบิต \$a ^ \$b

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	9	0	0	0	0	1	0	0	1
\$b	10	0	0	0	0	1	0	1	0
\$a ^ \$b	3	0	0	0	0	0	0	1	1

ผลลัพธ์ที่ได้คือ 00000011 เป็นค่าบวก นั่นคือ $(1 * 2^1) + (1 * 2^0) = 2 + 1 = 3$

บรรทัดที่ 7 เป็นการแสดงผลของ ~\$a & \$b นั่นคือค่าของบิตที่ไม่เป็นเซตของ \$a แต่เป็นเซตของ \$b ซึ่งเมื่อเขียนค่าของตัวแปรทั้ง 2 ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.10

ตารางที่ 4.10: การเปรียบเทียบบิต ~\$a & \$b

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	9	0	0	0	0	1	0	0	1
~\$a	~9	1	1	1	1	0	1	1	0
\$b	10	0	0	0	0	1	0	1	0
~\$a & \$b	2	0	0	0	0	0	0	1	0

ผลลัพธ์ที่ได้คือ 00000010 เป็นค่าบวก นั่นคือ $(1 * 2^1) = 2$

บรรทัดที่ 8 เป็นการแสดงผลของ $\$a \& \sim \b นั่นคือค่าของบิตที่เป็นเซตของ $\$a$ แต่ไม่เป็นเซตของ $\$b$ ซึ่งเมื่อเขียนค่าของตัวแปรทั้ง 2 ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.11

ตารางที่ 4.11: การเปรียบเทียบบิต $\$a \& \sim \b

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
$\$a$	9	0	0	0	0	1	0	0	1
$\$b$	10	0	0	0	0	1	0	1	0
$\sim \$b$	~ 10	1	1	1	1	0	1	0	1
$\$a \& \sim \b	1	0	0	0	0	0	0	0	1

ผลลัพธ์ที่ได้คือ 00000001 เป็นค่าบวก นั่นคือ $(1 * 2^0) = 1$

บรรทัดที่ 9 เป็นการแสดงผลของ $\sim \$a \& \sim \b นั่นคือค่าของบิตที่ไม่เป็นเซตของ $\$a$ และค่าของบิตที่ไม่เป็นเซตของ $\$b$ ซึ่งเมื่อเขียนค่าของตัวแปรทั้ง 2 ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.12

ตารางที่ 4.12: การเปรียบเทียบบิต $\sim \$a \& \sim \b

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
$\$a$	9	0	0	0	0	1	0	0	1
$\$b$	10	0	0	0	0	1	0	1	0
$\sim \$a$	~ 9	1	1	1	1	0	1	1	0
$\sim \$b$	~ 10	1	1	1	1	0	1	0	1
$\sim \$a \& \sim \b	-12	1	1	1	1	0	1	0	0

จะสังเกตเห็นว่าผลลัพธ์ที่ได้คือ 11110100 ซึ่งเป็นค่าติดลบ มีวิธีการหาค่าของเลขฐานสองของ 11110100 ดังนี้คือ

1. หา One's Complement ของเลขฐานสอง 11110100
 $11110100 \rightarrow 00001011$
2. นำผลลัพธ์ที่ได้จากข้อ 1 มาบวกกับ 1
 $00001011 + 1 \rightarrow 00001100 \rightarrow 12$
3. เนื่องจากเลขฐานสอง 11110110 เป็นเลขติดลบ ดังนั้นผลที่ได้จึงเป็น -12

บรรทัดที่ 10 เป็นการแสดงผลของ $\sim \$a$ นั่นคือค่าของบิตที่ไม่เป็นเซตของ $\$a$ ซึ่งเมื่อเขียนค่าของตัวแปร $\$a$ ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.13

ตารางที่ 4.13: การเปรียบเทียบบิต $\sim \$a$

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
$\$a$	9	0	0	0	0	1	0	0	1
$\sim \$a$	-10	1	1	1	1	0	1	1	0

จะสังเกตเห็นว่าผลลัพธ์ที่ได้คือ 11110110 ซึ่งเป็นค่าติดลบ ซึ่งมีวิธีการหาค่าของเลขฐานสองของ 11110110 ดังนี้คือ

1. หา One's Complement ของเลขฐานสอง 11110110
 $11110110 \rightarrow 00001001$
2. นำผลลัพธ์ที่ได้จากข้อ 1 มาบวกกับ 1
 $00001001 + 1 \rightarrow 00001010 \rightarrow 10$
3. เนื่องจากเลขฐานสอง 11110110 เป็นเลขติดลบ ดังนั้นผลที่ได้จึงเป็น -10

บรรทัดที่ 11 เป็นการแสดงผลของ $\sim \$b$ นั่นคือค่าของบิตที่ไม่เป็นเซตของ $\$b$ ซึ่ง $\$b$ มีค่าเท่ากับ 10 เมื่อนำมาเขียนให้อยู่ในรูปของเลขฐานสอง 8 บิต จะได้ดังในตารางที่ 4.14

ตารางที่ 4.14: การเปรียบเทียบบิต $\sim \$b$

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2^7)	64 (2^6)	32 (2^5)	16 (2^4)	8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
$\$b$	10	0	0	0	0	1	0	1	0
$\sim \$b$	-11	1	1	1	1	0	1	0	1

จะสังเกตเห็นว่าผลลัพธ์ที่ได้คือ 11110101 ซึ่งเป็นค่าติดลบ ซึ่งมีวิธีการหาค่าของเลขฐานสองของ 11110101 ดังนี้คือ

วิธีการหาค่าของเลขฐานสองของ 11110101 ทำดังนี้คือ

1. หา One's Complement ของเลขฐานสอง 11110101
 $11110101 \rightarrow 00001010$
2. นำผลที่ได้จากข้อ 1 มาบวกกับ 1
 $00001010 + 1 \rightarrow 00001011 \rightarrow 11$
3. เนื่องจากเลขฐานสอง 11110101 เป็นเลขติดลบ ดังนั้นผลที่ได้จึงเป็น -11

บรรทัดที่ 12 เป็นการแสดงผลของ $\sim \$c$ นั่นคือค่าของบิตที่ไม่เป็นเซตของ $\$c$ ซึ่ง $\$c$ มีค่าเท่ากับ -20 เมื่อนำมาเขียนให้อยู่ในรูปของเลขฐานสอง 8 บิต เนื่องจากเป็นค่าติดลบ จึงต้องใช้วิธี Two's Complement จะได้ดังในตารางที่ 4.15 โดยมีวิธีการทำดังนี้คือ

1. หาค่าสัมบูรณ์ (Absolute Value) ของ $\$c$ ในรูปแบบของเลขฐานสอง
 $|-20| \rightarrow 20 \rightarrow 00010100$
2. หา One's Complement:
 $00010100 \rightarrow 11101011$
3. นำผลลัพธ์ที่ได้จากข้อ 2 มาบวกกับ 1
 $11101011 + 1 \rightarrow 11101100$

ตารางที่ 4.15: การเปรียบเทียบบิต $\sim \$c$

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$c	-20	1	1	1	0	1	1	0	0
~\$c	19	0	0	0	1	0	0	1	1

ผลลัพธ์ที่ได้คือ 00010011 เป็นค่าบวก นั่นคือ $(1*2^4)+(1*2^1)+(1*2^0) = 16+2+1 = 19$

บรรทัดที่ 13 เป็นการแสดงผลของ $\$a << 2$ นั่นคือเลื่อนค่าของบิต \$a ไปทางซ้ายเป็นจำนวน 2 ครั้ง ซึ่งเมื่อเขียนค่าของตัวแปร \$a ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.16

ตารางที่ 4.16: การเปรียบเทียบบิต $\$a << 2$

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	9	0	0	0	0	1	0	0	1
\$a << 2	36	0	0	1	0	0	1	0	0

ผลลัพธ์ที่ได้คือ 00100100 เป็นค่าบวก นั่นคือ $(1*2^5)+(1*2^2) = 32+4 = 36$

บรรทัดที่ 14 เป็นการแสดงผลของ $\$a >> 2$ นั่นคือเลื่อนค่าของบิต \$a ไปทางขวาเป็นจำนวน 2 ครั้ง ซึ่งเมื่อเขียนค่าของตัวแปร \$a ให้อยู่ในเลขฐานสองแล้วจะได้ผลดังในตารางที่ 4.17

ตารางที่ 4.17: การเปรียบเทียบบิต $\$a >> 2$

ตัวแปร	ค่าในเลขฐานสิบ	ค่าในเลขฐานสอง (8 บิต)							
		128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
\$a	-20	0	0	0	0	1	0	0	1
\$a >> 2	2	0	0	0	0	0	0	1	0

ผลลัพธ์ที่ได้คือ 00000010 เป็นค่าบวก นั่นคือ $(1*2^1) = 2$

4.1.7 ตัวดำเนินการตรรกศาสตร์ (Logical Operators)

ตารางที่ 4.18: การใช้ตัวดำเนินการในการกำหนดค่าร่วมกับตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	การใช้งาน	ผลที่ได้
&&	$\$a \ \&\& \ \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าของทั้ง 2 เป็นจริงเท่านั้น
and	$\$a \ \text{and} \ \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าของทั้ง 2 เป็นจริงเท่านั้น
	$\$a \ \ \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าใดค่าหนึ่งเป็นจริง หรือค่าทั้ง 2 เป็นจริง
or	$\$a \ \text{or} \ \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าใดค่าหนึ่งเป็นจริง หรือค่าทั้ง 2 เป็นจริง
xor	$\$a \ \text{xor} \ \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าใดค่าหนึ่งเป็นจริงเท่านั้น (ถ้าค่าทั้ง 2 เป็นจริงผลที่ได้จะเป็นเท็จ)
!	$!\$a$	จะเป็นจริงถ้าค่าของ \$a เป็นเท็จ (จะเป็นเท็จถ้าค่าของ \$a เป็นจริง)

4.1.8 ตัวดำเนินการควบคุมข้อผิดพลาด (Error Control Operators)

คำสั่งบางคำสั่งในภาษา PHP อาจเกิดข้อผิดพลาดได้ ซึ่งจะมีข้อความบอกถึงข้อผิดพลาดนั้น ๆ แต่ถ้าไม่สนใจในข้อผิดพลาดที่จะเกิดขึ้นกับคำสั่งนั้น ๆ สามารถทำได้โดยใช้ตัวดำเนินการควบคุมข้อผิดพลาด

คือเครื่องหมาย at sign (@) นำหน้านิพจน์ ดังตัวอย่างการไม่ใช้ตัวดำเนินการควบคุมข้อผิดพลาดในรูปแบบที่ 4.7 และการใช้ตัวดำเนินการควบคุมข้อผิดพลาดในรูปแบบที่ 4.8

```
<?php
$result = 25/0;
echo $result;
echo "Bye Bye";
?>
```

Warning: Division by zero in D:\AppServ\www\php_book\cp4\at1.php on line 2
Bye Bye

รูปที่ 4.7: การแสดงผลของการใช้ตัวดำเนินการคำสั่ง

```
<?php
$result = @(25/0);
echo $result;
echo "Bye Bye";
?>
```

Bye Bye

รูปที่ 4.8: การแสดงผลของการใช้ตัวดำเนินการคำสั่ง

4.1.9 ตัวดำเนินการจัดการคำสั่ง (Execution Operators)

ภาษา PHP เอื้ออำนวยให้ผู้เขียนโปรแกรมสามารถใช้คำสั่งลักษณะ Shell Command โดยคำสั่งเหล่านั้นจะต้องอยู่ภายในเครื่องหมาย Backticks (` `) ปกติอยู่ที่แป้นพิมพ์เดียวกันกับเครื่องหมาย Tilde (~) ซึ่งแตกต่างจากเครื่องหมาย Single Quotation (‘ ’) และโดยส่วนใหญ่แล้วผลของการทำงานของคำสั่งนั้น ๆ จะถูกเก็บไว้ที่ตัวแปรก่อน ก่อนที่จะส่งไปแสดงผลโดยตรง เนื่องจากผลที่ได้จากการใช้คำสั่งเหล่านี้จะเป็นข้อมูลที่สลับซับซ้อน ดังตัวอย่างในรูปแบบที่ 4.9

```
<?php
$output = `dir D:`;
echo '<pre>'.$output.'</pre>';
?>
```

```
Volume in drive D is KC_1
Volume Serial Number is 3C67-CB71
Directory of D:\AppServ\www\php_book\cp4

04/26/2007  09:33 AM
.
..
02/25/2007  01:19 PM    836 loop.php
02/25/2007  01:13 PM    822 loop.html
04/25/2007  10:39 AM    219 operator1.php
04/25/2007  10:36 AM    306 operator2.php
04/26/2007  10:55 AM    134 dot.php
04/26/2007  11:06 AM    324 dot2.php3
04/26/2007  11:06 AM    324 dot2.php
04/26/2007  04:48 PM    502 bit.php
04/27/2007  09:44 AM     63 backtick1.php
04/27/2007  09:44 AM     63 backtick2.php
10 File(s)          3,593 bytes
 2 Dir(s)           9,876,938,752 bytes free
```

รูปที่ 4.9: การแสดงผลของการใช้ตัวดำเนินการจัดการคำสั่ง

4.1.10 ตัวดำเนินการอะเรย์ (Array Operators)

ตัวดำเนินการอะเรย์ส่วนใหญ่แล้ว จะทำหน้าที่เปรียบเทียบอะเรย์ทั้งสองและส่งผลลัพธ์เป็นจริงหรือเท็จ แต่ตัวดำเนินการอะเรย์ + จะทำการรวมข้อมูลของอะเรย์ทั้งสองแบบยูเนียน (Union) นั่นคือถ้าคีย์เหมือนกัน ค่าของอะเรย์ตัวแรกจะทับค่าของอะเรย์ตัวที่สอง ดังแสดงในรูปแบบที่ 4.10

ตารางที่ 4.19: ตัวดำเนินการอระเรย์

ตัวดำเนินการ	การใช้งาน	ผลที่ได้
+	$\$a + \$b;$	จะทำการรวมแบบยูเนียน (Union) อระเรย์ของทั้ง 2
==	$\$a == \$b;$	จะเป็นจริงก็ต่อเมื่อ คีย์/ค่า (key/value) ของทั้ง 2 คู่เหมือนกัน
===	$\$a === \$b;$	จะเป็นจริงก็ต่อเมื่อ คีย์/ค่า (key/value) ของทั้ง 2 คู่เหมือนกันและเป็นชนิดเดียวกัน
!=	$\$a != \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าของ $\$a$ ไม่เท่ากับค่าของ $\$b$
<>	$\$a <> \$b;$	จะเป็นจริงก็ต่อเมื่อ ค่าของ $\$a$ ไม่เท่ากับค่าของ $\$b$
!==	$\$a !== \$b;$	จะเป็นจริงก็ต่อเมื่อ คีย์/ค่า (key/value) ของทั้ง 2 คู่ไม่เหมือนกันหรือไม่เป็นชนิดเดียวกัน

```
<?php
$x = array("a"=>"apple", "b"=>"grape");
$y = array("a"=>"apple", "b"=>"orange", "c"=>"mango");
$z = $x+$y;
echo "Union of \$x and \$y:<br>";
var_dump($z);
echo "<br>";
$z = $y+$x;
echo "Union of \$y and \$x:<br>";
var_dump($z);
?>
```

↓

```
Union of $x and $y:
array(3) { ["a"]=> string(5) "apple" ["b"]=> string(5) "grape" ["c"]=> string(5) "mango" }
Union of $y and $x:
array(3) { ["a"]=> string(5) "apple" ["b"]=> string(6) "orange" ["c"]=> string(5) "mango" }
```

รูปที่ 4.10: การแสดงผลของการใช้ตัวดำเนินการอระเรย์

4.1.11 ตัวดำเนินการชนิด (Type Operators)

ตัวดำเนินการชนิดมี 1 ตัวคือ instanceof ใช้เพื่อตรวจสอบว่าตัวแปรหรืออ็อบเจกต์หนึ่ง ๆ เป็นอ็อบเจกต์คลาสที่ระบุหรือไม่ ดังแสดงในรูปที่ 4.11

```
<?php
class Apple { }
class Banana { }
$temp = new Apple;
if ($temp instanceof Apple) {
    echo 'Apple.';
}
if ($thing instanceof Banana) {
    echo 'Banana.';
}
?>
```

Apple.

รูปที่ 4.11: การแสดงผลของการใช้ตัวดำเนินการชนิด

4.1.12 ตัวดำเนินการแบบเงื่อนไข (Conditional Operators)

ตัวดำเนินการแบบเงื่อนไข คือตัวดำเนินการที่ต้องทำการเปรียบเทียบเงื่อนไขก่อน ก่อนทำการกำหนดค่าให้กับตัวแปรที่อยู่ทางซ้ายมือของเครื่องหมายเท่ากับ ซึ่งมีรูปแบบดังนี้คือ

$variable = (expression1) ? (expression 2) : (expression 3);$

ซึ่งขั้นตอนการทำงานของตัวดำเนินการแบบเงื่อนไขก็คือ

1. เปรียบเทียบ expression 1 ว่าเป็นจริงหรือเท็จ

2. ถ้าผลที่ได้จากข้อ 1 เป็นจริง ให้เอาค่าที่ได้จาก expression 2 ไปให้กับตัวแปรที่อยู่ทางด้านซ้ายมือของเครื่องหมายเท่ากับ มิฉะนั้นแล้วให้เอาค่าที่ได้จาก expression 3 ไปแทน

```
<?php
$x = 5;
$y = 10;
$min = ($x < $y) ? $x : $y;
$max = ($x > $y) ? $x : $y;
echo '$x = '.$x.', $y = '.$y.'<br>';
echo '$max = '.$max.', $min = '.$min.'<br>';
?>
```

\$x = 5, \$y = 10
\$max = 10, \$min = 5

รูปที่ 4.12: การแสดงผลของการใช้ตัวดำเนินการแบบเงื่อนไข

4.1.13 ลำดับความสำคัญของตัวดำเนินการ (Operator Precedence)

ตัวดำเนินการแต่ละตัวต่างก็มีลำดับความสำคัญและวิธีการทำงานที่แตกต่างกัน จึงจำเป็นต้องทำความเข้าใจและรู้ถึงขั้นตอนในการทำงานของตัวดำเนินการต่าง ๆ ถ้าตัวดำเนินการใดมีความสำคัญเท่ากัน วิธีการทำงานก็จะขึ้นอยู่กับความสัมพันธ์ของตัวดำเนินการนั้น ๆ เช่น ทำจากซ้ายไปขวาหรือขวาไปซ้าย แต่ถ้ามีความสำคัญต่างกัน ให้ทำตัวดำเนินการที่มีความสำคัญมากกว่าก่อน แล้วจึงทำตัวดำเนินการที่มีความสำคัญรองลงมา ซึ่งสามารถสรุปได้ดังในตารางที่ 4.20

ตารางที่ 4.20: ลำดับความสำคัญและวิธีการทำงานของตัวดำเนินการ

ความสัมพันธ์	ตัวดำเนินการ	ลำดับความสำคัญ
ไม่มี	new	↑ สูงสุด ↓ ต่ำสุด
ซ้ายไปขวา	[
ไม่มี	++ --	
ไม่มี	~ - (int) (float) (string) (array) (object) @	
ไม่มี	instanceof	
ขวาไปซ้าย	!	
ซ้ายไปขวา	* / %	
ซ้ายไปขวา	+ - .	
ซ้ายไปขวา	<< >>	
ไม่มี	< <= > >=	
ไม่มี	== != === !==	
ซ้ายไปขวา	&	
ซ้ายไปขวา	^	
ซ้ายไปขวา		
ซ้ายไปขวา	&&	
ซ้ายไปขวา		
ซ้ายไปขวา	? :	
ขวาไปซ้าย	= += -= *= /= %= .= &= \= ^= <<= >>=	
ซ้ายไปขวา	and	
ซ้ายไปขวา	xor	
ซ้ายไปขวา	Or	
ซ้ายไปขวา	,	

ตัวอย่าง กำหนดให้ $a = 5$; $b = 7$; $c = 2$; จงหาค่า $answer$ จากนิพจน์ต่อไปนี้
 $answer = a + b * c$;

จากนิพจน์ที่กำหนดให้ นั้น จะสังเกตเห็นว่ามีตัวดำเนินการอยู่ 3 ตัวคือ $=$, $+$ และ $*$ ซึ่งเมื่อเรียงลำดับความสำคัญของตัวดำเนินการทั้ง 3 จากความสำคัญสูงสุดไปต่ำสุด จะได้ $*$, $+$ และ $=$ ดังนั้นจึงต้องทำตัวดำเนินการ $*$ ก่อน ตามด้วยตัวดำเนินการ $+$ แล้วจึงทำตัวดำเนินการ $=$ ดังขั้นตอนต่อไปนี้

1. ตัวดำเนินการ $*$ มีความสัมพันธ์คือทำงานจากซ้ายไปขวา จะได้ $b * c$ ซึ่งเท่ากับ $7 * 2$ ผลลัพธ์ที่ได้คือ 14

2. ตัวดำเนินการ $+$ มีความสัมพันธ์คือทำงานจากซ้ายไปขวา จะได้ $a +$ ผลลัพธ์ที่ได้จากข้อที่ 1 ซึ่งเท่ากับ 14 นั่นคือ $5 + 14$ ผลลัพธ์ที่ได้คือ 19

3. ตัวดำเนินการ $=$ มีความสัมพันธ์คือทำงานจากขวาไปซ้าย นั่นคือเอาผลลัพธ์ที่ได้จากทางด้านขวามือของเครื่องหมายเท่ากับ $(a + b * c)$ ไปกำหนดค่าให้กับตัวแปรที่อยู่ทางด้านซ้ายมือของเครื่องหมายเท่ากับ $(answer)$ ผลลัพธ์ที่ได้คือ ตัวแปร $answer$ จะมีค่าเป็น 19

ถ้าตัวดำเนินการไม่มีความสัมพันธ์และลำดับความสำคัญแล้ว ผลลัพธ์ที่ได้จะมีข้อผิดพลาด เช่น จากตัวอย่างที่กล่าวมาแล้วนี้ ผลลัพธ์ที่ได้อาจจะเป็น 24 ก็ได้ (เอา a บวก b ก่อน แล้วคูณด้วย c)

4.2 การควบคุม (Controls)

ในบทต้น ๆ นั้นจะสังเกตเห็นว่า การทำงานของคำสั่งแต่ละคำสั่งนั้น จะถูกกระทำแบบตามลำดับ (Sequential) นั่นคือทำจากคำสั่งในบรรทัดแรกไปเรื่อย ๆ จนถึงบรรทัดสุดท้าย ซึ่งการทำงานของคำสั่งในลักษณะนี้ยังขาดความยืดหยุ่นและประสิทธิภาพในการทำงาน เนื่องจากในบางครั้งมีความจำเป็นที่จะต้องทดสอบเงื่อนไขต่าง ๆ ว่าเป็นไปในทิศทางใด เช่น ถ้าการทดสอบเงื่อนไขเป็นจริงจะทำคำสั่งใดบ้าง หรือถ้าเป็นเท็จแล้วจะให้โปรแกรมทำงานอย่างไร เป็นต้น ตัวอย่างเช่น นักเขียนโปรแกรมสามารถเขียนคำสั่งให้แสดงผลออกที่จอภาพ 3 บรรทัด โดยในบรรทัดที่ 1 นั้นให้แสดงเลขจำนวนเต็มจาก 1 ถึง 10 บรรทัดที่ 2 ให้แสดงตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กโดยเรียงลำดับจาก a ถึง z ส่วนบรรทัดที่ 3 นั้นให้แสดงตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่โดยเรียงลำดับจาก Z ถึง A ได้ดังในรูปที่ 4.13

```

<?php
echo 'Line 1: 1 2 3 4 5 6 7 8 9 10<br>';
echo 'Line 2: a b c d e f g h i j k l m n o p q r s t u v w x y z<br>';
echo 'Line 3: Z Y X W V U T S R Q P O N M L K J I H G F E D C B A<br>';
?>

```



```

Line 1: 1 2 3 4 5 6 7 8 9 10
Line 2: a b c d e f g h i j k l m n o p q r s t u v w x y z
Line 3: Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

```

รูปที่ 4.13: การแสดงผลของการใช้ตัวดำเนินการแบบเงื่อนไข

ถ้าต้องการแสดงข้อมูลเฉพาะที่ผู้ใช้ต้องการ เช่น ผู้ใช้ต้องการดูการแสดงผลของเลขจำนวนเต็มจาก 1 ถึง 10 หรือต้องการดูการแสดงผลของตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่โดยเรียงลำดับจาก Z ถึง A เท่านั้น นักเขียนโปรแกรมจะอย่างไร? คำตอบก็คือ นักเขียนโปรแกรมจะต้องทำการทดสอบโดยมีการกำหนดเงื่อนไขว่าผู้ใช้ต้องการให้โปรแกรมแสดงผลอะไร? โดยเงื่อนไขนี้จะถูกออกแบบโดยใช้แบบฟอร์ม

เพื่อให้ผู้ใช้ได้เลือกสิ่งที่ต้องการ เมื่อผู้ใช้เลือกสิ่งที่ต้องการแล้ว ให้กดปุ่ม Submit ซึ่งจะทำการส่งข้อมูลที่ผู้ใช้เลือกไปที่ไฟล์สกุล .php ไฟล์หนึ่งที่จะทำการรับข้อมูลจากแบบฟอร์ม ทดสอบเงื่อนไข และแสดงผลตามเงื่อนไขที่ได้รับ หรืออาจจะมีเงื่อนไขที่ผู้ใช้สามารถเลือกได้มากกว่าการแสดงผลของทั้ง 3 รูปแบบที่กล่าวมาแล้วข้างต้น เช่น อาจจะไม่เลือกการแสดงตัวอักษรตัวพิมพ์ใหญ่โดยเรียงลำดับจากน้อยไปหามาก (A – Z) โดยเริ่มจากตัวอักษรตัวใดตัวหนึ่งก็ได้ เป็นต้น ซึ่งการควบคุมสามารถแบ่งออกเป็น 2 ลักษณะใหญ่ ๆ คือ

1. การควบคุมแบบทางเลือก (Branching)
2. การควบคุมแบบวนลูป (Looping)

4.2.1 การควบคุมแบบทางเลือก (Branching)

ผู้เขียนโปรแกรมสามารถกำหนดเงื่อนไขในการทดสอบโดยการควบคุมแบบทางเลือกได้ เช่น ถ้าเงื่อนไขที่ทดสอบเป็นจริงจะให้โปรแกรมทำงานทางหนึ่ง แต่ถ้าเป็นเท็จจะให้ทำงานอีกทางหนึ่ง เป็นต้น คำสั่งที่ใช้ในการควบคุมแบบทางเลือก มี 4 คำสั่งด้วยกันคือ if, if...else, if...elseif (หรือ else if), และ switch

4.2.1.1 คำสั่ง if

รูปแบบของคำสั่ง if คือ

```
if (condition) {
    statement1;
    statement2;
}
```

การทำงานของคำสั่ง if คือจะทดสอบเงื่อนไข (Condition) ก่อน ถ้าเงื่อนไขเป็นจริง ก็จะทำคำสั่งที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด แต่ถ้าเป็นเท็จ ก็ให้ไปทำคำสั่งที่อยู่ต่อจากเครื่องหมายปีกกาปิด เช่นดังตัวอย่างในรูปที่ 4.14

<pre>1. <?php 2. \$score = 80; 3. if(\$score >= 50){ 4. echo 'You have passed the course.
'; 5. echo 'Good Luck !!!
'; 6. } 7. echo 'Program Terminated!'; 8. ?></pre>	<p>→</p> <pre>You have passed the course. Good Luck !!! Program Terminated!</pre>
---	---

รูปที่ 4.14: การแสดงผลของการใช้ if (เมื่อเป็นจริง)

จากคำสั่งในรูปที่ 4.14 บรรทัดที่ 2 นั้น ถ้าเปลี่ยนค่าของ \$score จาก 80 เป็น 45 ผลที่ได้จะแตกต่างกันไป เนื่องจากเมื่อทำการทดสอบเงื่อนไขแล้ว ผลการทดสอบที่ได้นั้นจะเป็นเท็จ เพราะว่า 45 น้อยกว่า 50 ดังนั้น คำสั่งที่อยู่ภายในปีกกาเปิดและปิด (บรรทัดที่ 4 และ 5) จึงไม่ถูกกระทำ แต่จะไปทำคำสั่งในบรรทัดที่ 7 ซึ่งเป็นคำสั่งถัดไป ดังแสดงในรูปที่ 4.15

<pre>1. <?php 2. \$score = 45; 3. if(\$score >= 50){ 4. echo 'You have passed the course.
'; 5. echo 'Good Luck !!!
'; 6. } 7. echo 'Program Terminated!'; 8. ?></pre>	<p>→</p> <pre>Program Terminated!</pre>
---	---

รูปที่ 4.15: การแสดงผลของการใช้ if (เมื่อเป็นเท็จ)

4.2.1.2 คำสั่ง if...else

รูปแบบของคำสั่ง if...else คือ

```
if (condition) {
    statement1;
    statement2;
}else{
    statement3;
    statement4;
}
```

การทำงานของคำสั่ง if...else คือจะทดสอบเงื่อนไข (Condition) ก่อน ถ้าเงื่อนไขเป็นจริง ก็จะทำคำสั่งที่อยู่ในเครื่องหมายปีกกาเปิดและปิด (statement1 และ statement2) แต่ถ้าเป็นเท็จให้ไปทำคำสั่งที่อยู่ในช่วงของ else ที่อยู่ในเครื่องหมายปีกกาเปิดและปิด (statement3 และ statement4) แทน เช่น ดังตัวอย่างในรูปแบบที่ 4.16

<pre>1. <?php 2. \$score = 80; 3. if(\$score >= 50){ 4. echo 'You have passed the course.
'; 5. echo 'Good Luck !!!
'; 6. }else{ 7. echo 'Failed
'; 8. echo 'Try Again !
'; 9. } 10. echo 'Program Terminated!'; 11. ?></pre>		<pre>You have passed the course. Good Luck !!! Program Terminated!</pre>
--	---	--

รูปที่ 4.16: การแสดงผลของการใช้ if...else (เมื่อเป็นจริง)

จากคำสั่งในรูปแบบที่ 4.16 บรรทัดที่ 2 นั้น ถ้าเปลี่ยนค่าของ \$score จาก 80 เป็น 45 ผลที่ได้จะแตกต่างกันไป เนื่องจากเมื่อทำการทดสอบเงื่อนไขแล้ว ผลการทดสอบที่ได้นั้นจะเป็นเท็จ เพราะว่า 45 น้อยกว่า 50 ดังนั้นคำสั่งที่อยู่ในปีกกาเปิดและปิด (บรรทัดที่ 4 และ 5) จึงไม่ถูกกระทำ แต่จะไปทำคำสั่งที่อยู่ในช่วงของ else แทน (บรรทัดที่ 7 และ 8) เสร็จแล้วจึงไปทำคำสั่งในบรรทัดที่ 10 ดังแสดงในรูปแบบที่ 4.17

<pre>1. <?php 2. \$score = 45; 3. if(\$score >= 50){ 4. echo 'You have passed the course.
'; 5. echo 'Good Luck !!!
'; 6. }else{ 7. echo 'Failed
'; 8. echo 'Try Again !
'; 9. } 10. echo 'Program Terminated!'; 11. ?></pre>		<pre>Failed Try Again ! Program Terminated!</pre>
--	---	---

รูปที่ 4.17: การแสดงผลของการใช้ if...else (เมื่อเป็นเท็จ)

4.2.1.3 คำสั่ง if...elseif (หรือ else if)

รูปแบบของคำสั่ง if...elseif หรือ if...else if คือ

```
if(condition) {
    statement1;
    statement2;
}elseif(condition) {
    statement3;
    statement4;
}else{
    statement5;
    statement6;
}
```

การทำงานของคำสั่ง if...elseif (หรือ if...else if) คือจะทดสอบเงื่อนไข (Condition) ก่อน ถ้าเงื่อนไขเป็นจริง ก็จะทำคำสั่งที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด (statement1 และ statement2) แต่ถ้าเป็นเท็จก็จะไปทำการทดสอบเงื่อนไข (Condition) ที่อยู่ภายในวงเล็บหลัง elseif ถ้าเงื่อนไขเป็นจริง ก็จะทำคำสั่งที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด (statement3 และ statement4) แต่ถ้าเป็นเท็จก็จะให้ไปทำคำสั่งที่อยู่ในช่วงของ else ที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด (statement5 และ statement6) แทน เช่นดังตัวอย่างในรูปที่ 4.18

<pre> 1. <?php 2. \$score = 80; 3. if(\$score >= 80){ 4. echo 'Very Good
'; 5. echo 'Your score is very high.
'; 6. }elseif(\$score >= 50){ 7. echo 'You have passed the course.
'; 8. echo 'Good Luck !!!
'; 9. }else{ 10. 11. echo 'Failed
'; 12. echo 'Try Again !
'; 13. } 14. echo 'Program Terminated!'; 15. ?></pre>		<pre> Very Good Your score is very high. Program Terminated!</pre>
---	--	--

รูปที่ 4.18: การแสดงผลของการใช้ if...elseif แบบที่ 1

จากคำสั่งในรูปที่ 4.18 บรรทัดที่ 2 นั้น ถ้าเปลี่ยนค่าของ \$score จาก 80 เป็น 55 ผลที่ได้จะแตกต่างกันไป เนื่องจากเมื่อทำการทดสอบเงื่อนไขแล้ว ผลการทดสอบที่ได้นั้นจะเป็นเท็จ เพราะว่า 55 น้อยกว่า 80 ดังนั้นคำสั่งที่อยู่ภายในปีกกาเปิดและปิด (บรรทัดที่ 4 และ 5) จึงไม่ถูกกระทำ แต่จะไปทำการทดสอบในบรรทัดที่ 6 แทน ซึ่งผลจากการทดสอบเป็นจริง เนื่องจาก 55 มากกว่า 50 ดังนั้นจึงทำคำสั่งในบรรทัดที่ 7 และ 8 แทน เสร็จแล้วก็โดดไปทำคำสั่งในบรรทัดที่ 13 ต่อไป ดังแสดงในรูปที่ 4.19

<pre> 1. <?php 2. \$score = 55; 3. if(\$score >= 80){ 4. echo 'Very Good
'; 5. echo 'Your score is very high.
'; 6. }elseif(\$score >= 50){ 7. echo 'You have passed the course.
'; 8. echo 'Good Luck !!!
'; 9. }else{ 10. 11. echo 'Failed
'; 12. echo 'Try Again !
'; 13. } 14. echo 'Program Terminated!'; 15. ?></pre>		<pre> You have passed the course. Good Luck !!! Program Terminated!</pre>
---	--	---

รูปที่ 4.19: การแสดงผลของการใช้ if...elseif แบบที่ 2

จากคำสั่งในรูปที่ 4.19 บรรทัดที่ 2 นั้น ถ้าเปลี่ยนค่าของ \$score จาก 55 เป็น 30 ผลที่ได้จะแตกต่างกันไป เนื่องจากเมื่อทำการทดสอบเงื่อนไขแล้ว ผลการทดสอบที่ได้นั้นจะเป็นเท็จ เพราะว่า 30 น้อยกว่า 80 ดังนั้นคำสั่งที่อยู่ภายในปีกกาเปิดและปิด (บรรทัดที่ 4 และ 5) จึงไม่ถูกกระทำ แต่จะไปทำการทดสอบในบรรทัดที่ 6 แทน ซึ่งผลจากการทดสอบก็ยังเป็นเท็จอีก เนื่องจาก 30 น้อยกว่า 50 ดังนั้นจึงทำคำสั่งในส่วนของ else แทน (บรรทัดที่ 10 และ 11) เสร็จแล้วจึงไปทำคำสั่งในบรรทัดที่ 13 ต่อไป ดังแสดงในรูปที่ 4.20

<pre> 1. <?php 2. \$score = 30; 3. if(\$score >= 80){ 4. echo 'Very Good
'; 5. echo 'Your score is very high.
'; 6. }elseif(\$score >= 50){ 7. echo 'You have passed the course.
'; 8. echo 'Good Luck !!!
'; 9. }else{ 10. echo 'Failed
'; 11. echo 'Try Again !
'; 12. } 13. echo 'Program Terminated!'; 14. ?> </pre>		<pre> Failed Try Again ! Program Terminated! </pre>
---	--	---

รูปที่ 4.20: การแสดงผลของการใช้ if...elseif แบบที่ 3

4.2.1.4 คำสั่ง switch

รูปแบบของคำสั่ง switch คือ

```

switch(variable){
  case constant:
    statement1;
    statement2;
    break;
  case constant:
    statement3;
    statement4;
    break;
  ...
  ...
  ...
  case constant:
    statement11;
    statement12;
    break;
  default:
    statement13;
}

```

การทำงานของคำสั่ง switch คือจะเอาค่าที่อยู่ในตัวแปร (ที่อยู่ภายในวงเล็บเปิดปิด ต่อจากคำสั่ง switch) ไปทดสอบเงื่อนไขในแต่ละกรณี (Case) กับค่าคงที่ (Constant) ถ้าตรงกับกรณีใด ก็จะทำคำสั่งที่อยู่ในส่วนของกรณีนั้น ๆ แต่ถ้าไม่ตรงกับกรณีใด ๆ ก็จะไปทำในส่วนของ default แทน ดังแสดงในรูปที่ 4.21

```

1.  <?php
2.  $choice = 5;
3.  switch($choice){
4.      case 1:
5.          $day = 'Sunday';
6.          $color = 'Red';
7.          break;
8.      case 2:
9.          $day = 'Monday';
10.         $color = 'Yellow';
11.         break;
12.      case 3:
13.         $day = 'Tuesday';
14.         $color = 'Pink';
15.         break;
16.      case 4:
17.         $day = 'Wednesday';
18.         $color = 'Green';
19.         break;
20.      case 5:
21.         $day = 'Thursday';
22.         $color = 'Orange';
23.         break;
24.      case 6:
25.         $day = 'Friday';
26.         $color = 'Blue';
27.         break;
28.      case 7:
29.         $day = 'Saturday';
30.         $color = 'Purple';
31.         break;
32.      default:
33.         echo 'Invalid Input';
34.     }
35.     if($choice >= 1 && $choice <= 7){
36.         echo 'You were born on '.$day.'  
';
37.         echo 'Your color is '.$color.'  
';
38.     }
39.     echo 'Program Terminated!';
40. }?>

```

You were born on Thursday.
Your color is Orange.
Program Terminated!

รูปที่ 4.21: การแสดงผลของการใช้ switch

จะสังเกตเห็นว่าการใช้คำสั่งในการควบคุมแบบทางเลือกนั้น ยังไม่ยืดหยุ่นเพียงพอ เนื่องจากมีการทดสอบเงื่อนไข แล้วทำคำสั่งหรือชุดคำสั่งตามที่เงื่อนไขนั้น ๆ กำหนด แล้วจึงไปทำคำสั่งต่อ ๆ ไปจนกระทั่งสิ้นสุดโปรแกรม โดยไม่มีการย้อนกลับไป ณ ตำแหน่งใดตำแหน่งหนึ่งของชุดคำสั่งนั้น ๆ ซ้ำอีก ซึ่งส่วนใหญ่แล้วผู้เขียนโปรแกรมต้องการเขียนชุดคำสั่งให้มีความยืดหยุ่นมากกว่านี้ ซึ่งเรียกว่าการทำงานแบบวนลูป ดังจะได้กล่าวต่อไป

4.2.2 การควบคุมแบบวนลูป (Looping)

การควบคุมแบบวนลูปมีหลายรูปแบบ ทั้งนี้ขึ้นอยู่กับผู้เขียนโปรแกรมว่าจะเลือกใช้แบบไหนที่สะดวกและเหมาะสมกับการควบคุมการทำงานของส่วนนั้น ๆ ซึ่งมีรายละเอียดดังนี้

4.2.2.1 ลูป for

รูปแบบของลูป for คือ

```

for(initialization; condition; increment/decrement){
    statement1;
    statement2;
}
statement3;

```

หลักการการทำงานของลูป for นั้น คือ

1. การกำหนดค่าเริ่มต้น (Initialization)
2. ทดสอบเงื่อนไข (Condition)

3. ถ้าเงื่อนไขเป็นเท็จจะกระโดดไปทำคำสั่งที่ 3 (statement3) ซึ่งคือการหลุดออกจากหลุม
4. ถ้าเงื่อนไขเป็นจริงก็จะเข้าไปทำกลุ่มของคำสั่งที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด (นั่นคือคำสั่ง statement1 และ 2)
5. เพิ่มค่าหรือลดค่า (Increment/Decrement)
6. กลับไปทำข้อที่ 2 ใหม่

ตัวอย่าง รูปที่ 4.22 แสดงถึงการใช้หลุม for ในการแสดงผลของเลขจำนวนเต็มจาก 11 ถึง 50 โดยแสดงแถวละ 10 จำนวน

<ol style="list-style-type: none"> 1.2.3.4.5.6.7.8.9.10.11. 	<pre> <?php \$count=0; for(\$i=11; \$i<=50; \$i++){ echo "\$i "; \$count++; if(\$count == 10){ echo '
'; \$count=0; } } ?> </pre>		<pre> 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 </pre>
--	---	--	--

รูปที่ 4.22: การแสดงผลของการใช้หลุม for

4.2.2.2 หลุม while

รูปแบบของหลุม while คือ

```

initialization;
while (condition) {
    statement1;
    statement2;
    increment/decrement;
}
statement3;
    
```

หลักการการทำงานของหลุม while นั้น คือ

1. การกำหนดค่าเริ่มต้น (Initialization)
2. ทดสอบเงื่อนไข (Condition)
3. ถ้าเงื่อนไขเป็นเท็จจะกระโดดไปทำคำสั่งที่ 3 (statement3) ซึ่งคือการหลุดออกจากหลุม
4. ถ้าเงื่อนไขเป็นจริงก็จะเข้าไปทำกลุ่มของคำสั่งที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด (นั่นคือคำสั่ง statement1, statement2 และ เพิ่มค่าหรือลดค่า (Increment/Decrement) แล้วแต่กรณี)
5. กลับไปทำข้อที่ 2 ใหม่

ตัวอย่าง รูปที่ 4.23 แสดงถึงการใช้ลูป while ในการแสดงผลของเลขจำนวนเต็มจาก 11 ถึง 50 โดยแสดงแถวละ 10 จำนวน

<pre> 1. <?php 2. \$count=0; 3. \$i=11; 4. while(\$i<=50){ 5. echo "\$i "; 6. \$count++; 7. if(\$count == 10){ 8. echo '
'; 9. \$count=0; 10. } 11. \$i++; 12. } 13. ?> </pre>		<pre> 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 </pre>
---	---	--

รูปที่ 4.23: การแสดงผลของการใช้ลูป while

4.2.2.3 ลูป do...while

รูปแบบของลูป do...while คือ

```

do{
    statement1;
    statement2;
}while (condition) ;
statement3;
                
```

หลักการทำงานของลูป do...while นั้น คือ

1. ทำคำสั่ง statement 1 และ 2
2. ทดสอบเงื่อนไข (Condition)
3. ถ้าเงื่อนไขเป็นจริง กลับไปทำข้อที่ 1 ใหม่ ถ้าเงื่อนไขเป็นเท็จจะกระโดดไปทำคำสั่งที่ 3

(statement3) ซึ่งคือการหลุดออกจากลูป

ตัวอย่าง รูปที่ 4.24 แสดงถึงการใช้ลูป do...while ในการแสดงผลของเลขจำนวนเต็มจาก 11 ถึง 50 โดยแสดงแถวละ 10 จำนวน

<pre> 1. <?php 2. \$count=0; 3. \$i=11; 4. do{ 5. echo "\$i "; 6. \$count++; 7. if(\$count == 10){ 8. echo '
'; 9. \$count=0; 10. } 11. \$i++; 12. }while(\$i<=50); 13. ?> </pre>		<pre> 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 </pre>
--	---	--

รูปที่ 4.24: การแสดงผลของการใช้ลูป do...while

4.2.2.4 หลูป foreach

หลูป foreach ช่วยให้การจัดการกับตัวแปรที่เก็บข้อมูลอะเรย์ได้เป็นอย่างดี ทำงานกับตัวแปรอะเรย์เท่านั้น ถ้าใช้หลูป foreach กับตัวแปรประเภทอื่นหรือตัวแปรที่ยังไม่ได้กำหนดค่า ระบบจะแจ้งข้อผิดพลาด ซึ่งไม่สามารถใช้ตัวดำเนินการควบคุมข้อผิดพลาด (@) ทำการควบคุมเพื่อไม่ให้แสดงข้อผิดพลาดได้ มีรูปแบบการทำงาน 2 รูปแบบดังนี้ คือ

รูปแบบที่ 1

```
foreach(array_expression as $value) {
    statement1;
    statement2;
}
```

เมื่อเริ่มต้นเข้าสู่หลูป foreach ตัวชี้ตำแหน่งภายในอะเรย์จะทำการชี้ที่ข้อมูลแรกในอะเรย์โดยอัตโนมัติ แล้วในแต่ละหลูปนั้น ค่าที่กำหนดโดย array_expression จะถูกกำหนดให้กับตัวแปร \$value แล้วตัวชี้ตำแหน่งภายในอะเรย์จะทำการเลื่อนไปชี้ที่ข้อมูลถัดไปโดยอัตโนมัติ (นั่นคือ ในหลูปถัดไปก็คือข้อมูลถัดไปนั่นเอง)

<ol style="list-style-type: none"> 1. 2. 3. 4. 5. 6. 	<pre><?php \$fruit=array("apple", "banana", "orange"); foreach (\$fruit as \$value){ echo "Value: \$value
"; } ?></pre>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Value: apple Value: banana Value: orange </div>
--	--	--

รูปที่ 4.25: การแสดงผลของการใช้หลูป foreach แบบที่ 1

รูปแบบที่ 2

```
foreach(array_expression as $key => $value){
    statement1;
    statement2;
}
```

ในรูปแบบที่ 2 นี้ ก็มีลักษณะการทำงานเช่นเดียวกับแบบที่ 1 เพียงแต่ตำแหน่งที่ตัวชี้ตำแหน่งภายในอะเรย์ที่ชื่ออยู่ในขณะนั้นจะถูกกำหนดให้กับตัวแปร \$key

<ol style="list-style-type: none"> 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 	<pre><?php \$fruit=array("apple", "banana", "orange"); foreach(\$fruit as \$key => \$value){ echo "Key: \$key; Value: \$value
"; } echo 'Another demonstration...
'; foreach(\$fruit as \$key => \$value) { echo '\$fruit['.\$key.'] = '.\$value.'
'; } ?></pre>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Key: 0; Value: apple Key: 1; Value: banana Key: 2; Value: orange Another demonstration... \$fruit[0] = apple \$fruit[1] = banana \$fruit[2] = orange </div>
---	---	--

รูปที่ 4.26: การแสดงผลของการใช้หลูป foreach แบบที่ 2

4.2.2.5 คำสั่ง break;

คำสั่ง break; เป็นคำสั่งที่ใช้เพื่อหยุดการทำงานของลูป for, while, do...while, foreach และ switch เช่นดังตัวอย่างในรูปที่ 4.27

```

1.  <?php
2.  $fruit=array("apple", "banana", "orange");
3.  foreach ($fruit as $value){
4.      if($value == "banana"){
5.          break;
6.      }
7.      echo "Value: $value<br>";
8.  }
9.  echo "Bye Bye!";
10. ?>
    
```

Value: apple
Bye Bye!

รูปที่ 4.27: การใช้คำสั่ง break;

จะสังเกตเห็นว่าในบรรทัดที่ 4 เป็นการทดสอบเงื่อนไขว่าค่าที่ถูกเก็บในตัวแปร \$value มีค่าเท่ากับสตริง "banana" หรือไม่ ถ้าเท่ากับ ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 5 ซึ่งเป็นการใช้คำสั่ง break;; ก็จะทำให้หลุดออกจากลูป foreach ซึ่งเป็นลูปที่ครอบคลุมคำสั่ง break; อยู่ 1 ชั้น แล้วไปทำคำสั่งในบรรทัดที่ 9 แล้วก็จบโปรแกรม แต่การใช้คำสั่ง break; ยังมีส่วนเสริมที่สามารถกำหนดจำนวนชั้นที่ต้องการให้หยุดทำงานได้ ดังแสดงในรูปที่ 4.28

```

1.  <?php
2.  $i=1;
3.  $count=0;
4.  while($i<10){
5.      for($ch='A'; $ch<='Z'; $ch++){
6.          echo "$ch ";
7.          $count++;
8.          if($ch == 'F'){
9.              break; //Exit only for loop
10.         }
11.         if($count == 10){
12.             break 2; // Exit for and while loop
13.         }
14.     }
15.     $i++;
16. }
17. echo "<br>". '$count = ' . $count;
18. echo "<br>". '$i = ' . $i;
19. echo "<br>Bye Bye!";
20. ?>
    
```

A B C D E F A B C D
\$count = 10
\$i = 2
Bye Bye!

รูปที่ 4.28: การใช้คำสั่ง break; แบบกำหนดชั้น

- บรรทัดที่ 2** เป็นการกำหนดค่า 1 ให้กับตัวแปร \$i
- บรรทัดที่ 3** เป็นการกำหนดค่า 0 ให้กับตัวแปร \$count
- บรรทัดที่ 4** เริ่มต้นเข้าสู่ลูป while โดยทำการทดสอบเงื่อนไขว่าค่าของ \$i น้อยกว่า 10 หรือไม่? ผลที่ได้คือจริง ดังนั้นจึงเข้าไปทำคำสั่งที่อยู่ในลูป while โดยเริ่มจากบรรทัดที่ 5
- บรรทัดที่ 5** เป็นการใช้ลูป for โดยกำหนดค่าเริ่มต้นคือตัวอักษร A ให้กับตัวแปร \$ch แล้วทำการทดสอบเงื่อนไขว่าค่าของ \$ch น้อยกว่าหรือเท่ากับตัวอักษร Z หรือไม่? ผลที่ได้คือจริง ดังนั้นจึงเข้าไปทำคำสั่งที่อยู่ในลูป for โดยเริ่มจากบรรทัดที่ 6
- บรรทัดที่ 6** เป็นการแสดงค่าที่อยู่ในตัวแปร \$ch (ตัวอักษร A) พร้อมกับช่องว่าง 1 ช่อง ออกที่จอภาพ

บรรทัดที่ 7 เป็นการเพิ่มค่าให้กับตัวแปร \$count อีก 1

บรรทัดที่ 8 เป็นการทดสอบเงื่อนไขว่าค่าของ \$ch เท่ากับอักษร F หรือไม่? ผลที่ได้คือเท็จ ดังนั้นจึงไปทำคำสั่งในบรรทัดที่ 11 ซึ่งก็เป็นการทดสอบเงื่อนไขว่าค่าของ \$count เท่ากับ 10 หรือไม่? ผลที่ได้คือเท็จ ดังนั้นจึงกลับไปบรรทัดที่ 5 เพื่อทำการเพิ่มค่าให้กับตัวแปร \$ch อีก 1 ค่าที่ถูกเก็บอยู่ในตัวแปร \$ch ก็กลายเป็นตัวอักษร B

ดังนั้นผลที่แสดงออกที่จอภาพคือ A B C D E F ซึ่งเมื่อแสดงผลถึงตัวอักษร F แล้ว จะทำให้ตัวแปร \$count มีค่าเป็น 6 และทำให้การทดสอบเงื่อนไขในบรรทัดที่ 8 เป็นจริง จึงไปทำคำสั่งในบรรทัดที่ 9 คือคำสั่ง break; จึงทำให้หลุดออกจากหลุม for 1 ชั้น แล้วไปทำคำสั่งในบรรทัดที่ 15

บรรทัดที่ 15 เป็นการเพิ่มค่าให้กับตัวแปร \$i อีก 1 ตัวแปร \$i จึงมีค่าเป็น 2 แล้ววนหลุมกลับขึ้นไปบรรทัดที่ 4 เพื่อทำการทดสอบเงื่อนไขว่าค่าของ \$i น้อยกว่า 10 หรือไม่? ผลที่ได้คือจริง ดังนั้นจึงเข้าไปทำคำสั่งที่อยู่ในหลุม while โดยเริ่มจากบรรทัดที่ 5 ใหม่

ผลที่ได้จะกลายเป็น A B C D E F A B C D ซึ่งเมื่อแสดงผลถึงตัวอักษร D แล้ว จะทำให้ตัวแปร \$count มีค่าเป็น 10 และทำให้การทดสอบเงื่อนไขในบรรทัดที่ 11 เป็นจริง จึงไปทำคำสั่งในบรรทัดที่ 12 คือคำสั่ง break 2; จึงทำให้หลุดออกจากหลุม for (ชั้นที่ 1) และหลุม while (ชั้นที่ 2) แล้วไปทำคำสั่งในบรรทัดที่ 17

บรรทัดที่ 17 เป็นการขึ้นบรรทัดใหม่แล้วแสดงค่าของตัวแปร \$count ซึ่งมีค่าเท่ากับ 10 ออกที่จอภาพ

บรรทัดที่ 18 เป็นการขึ้นบรรทัดใหม่แล้วแสดงค่าของตัวแปร \$i ซึ่งมีค่าเท่ากับ 2 ออกที่จอภาพ

บรรทัดที่ 19 เป็นการขึ้นบรรทัดใหม่แล้วแสดงสตริง "Bye Bye!" ออกที่จอภาพ

4.2.2.6 คำสั่ง continue;

คำสั่ง continue; ใช้ในโครงสร้างของหลุมเพื่อหยุดการทำงานของคำสั่งที่เหลือ โดยไปเริ่มทำงานในหลุมรอบต่อไป ดังแสดงในรูปที่ 4.29

<ol style="list-style-type: none"> 1. 2. 3. 4. 5. 6. 7. 8. 9. 	<pre><?php for (\$i=1; \$i<6; \$i++){ if (\$i == 3) continue; echo "\$i"; echo '+'; echo '- ' ; } ?></pre>		1+- 2+- 4+- 5+-
--	---	--	-----------------

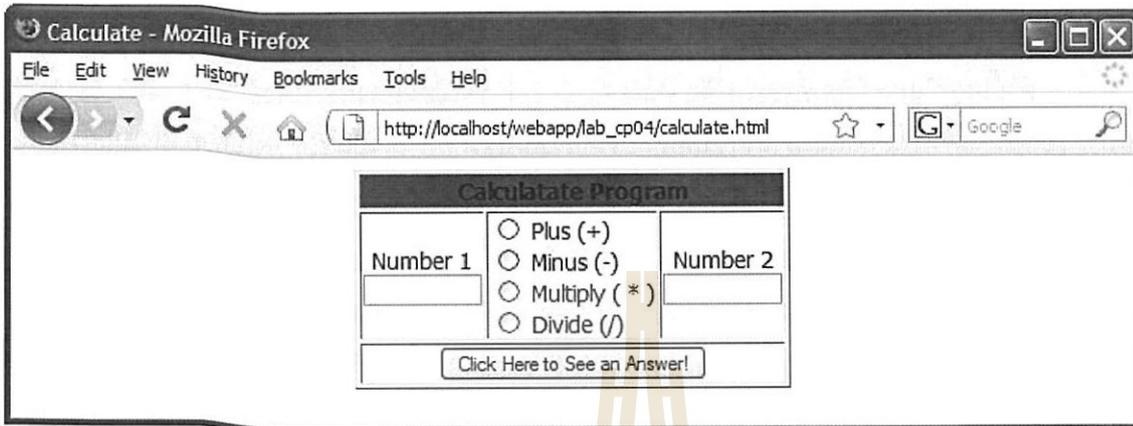
รูปที่ 4.29: การใช้คำสั่ง continue

สรุป

ตัวดำเนินการแต่ละตัวต่างก็มีลักษณะการใช้งานและลำดับความสำคัญที่แตกต่างกัน ดังนั้นจึงมีความจำเป็นที่จะต้องศึกษาและทำความเข้าใจในกฎเกณฑ์ต่าง ๆ เพื่อให้ได้ผลลัพธ์ที่ถูกต้องและเชื่อถือได้ รวมไปถึงเรื่องของการออกแบบและเขียนโปรแกรมโดยใช้คำสั่งในการควบคุม เพื่อให้โปรแกรมทำงานได้อย่างมีประสิทธิภาพดังที่ต้องการ

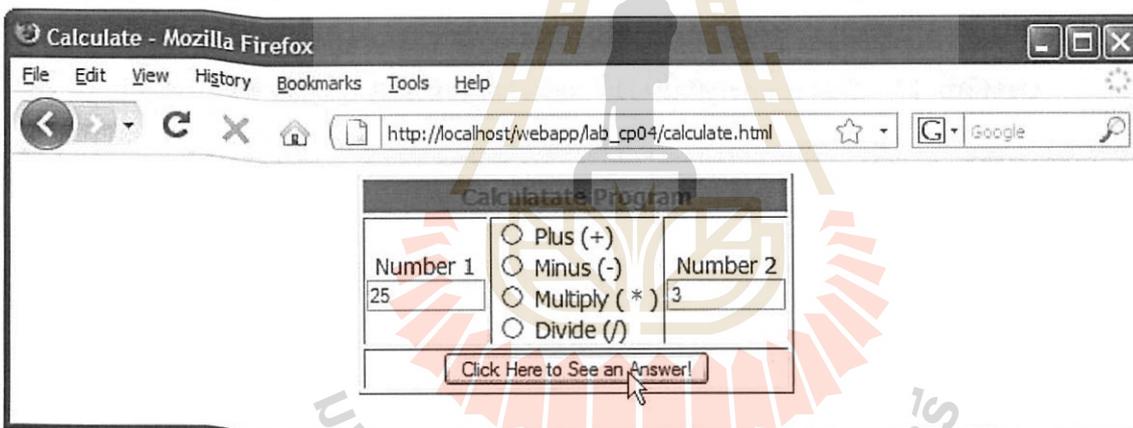
แบบฝึกหัด

1. ใช้โปรแกรม Text Editor สร้างไฟล์ HTML ชื่อว่า “calculate.html” เช่นตัวอย่างดังรูปที่ 4.30



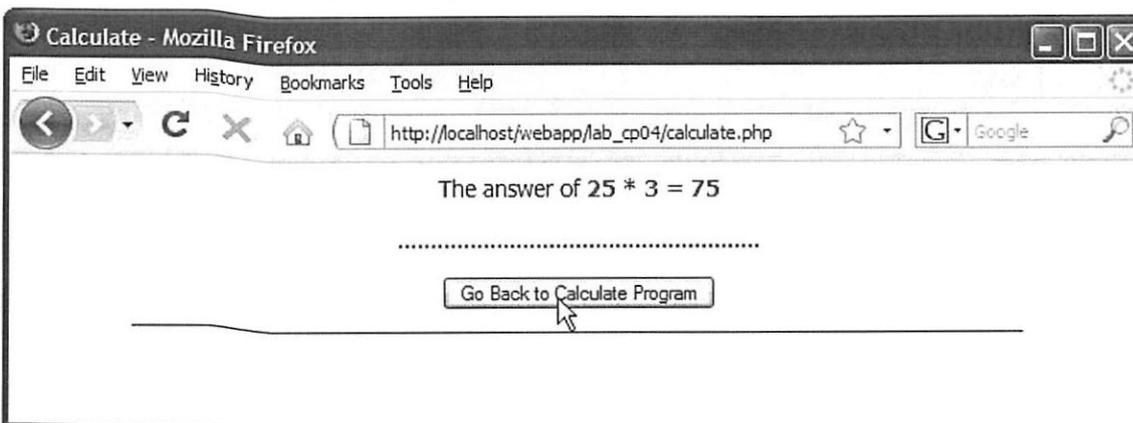
รูปที่ 4.30: ผลการแสดงผลของไฟล์ calculate.html

เมื่อผู้ใช้ป้อนข้อมูลและกดปุ่ม Click Here to See an Answer ดังในรูปที่ 4.31



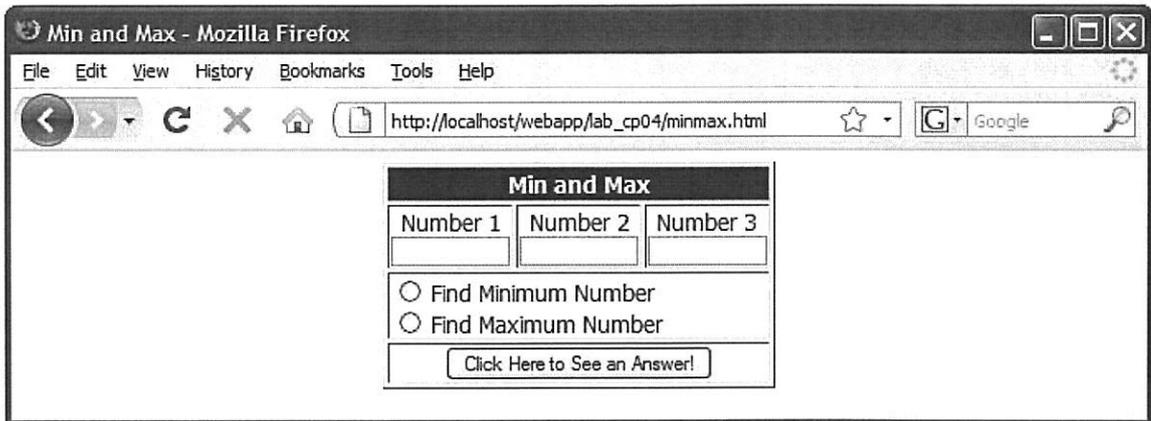
รูปที่ 4.31: การใส่ข้อมูลในแบบฟอร์มของไฟล์ calculate.html

จะส่งข้อมูลทั้งหมดไปที่ไฟล์ “calculate.php” และแสดงผลลัพธ์ออกดังในรูปที่ 4.32 ซึ่งจะมีปุ่ม Go Back to Calculate Program โดยปุ่มนี้จะเรียกไฟล์ “calculate.html” ขึ้นมาใหม่ เช่นดังรูปที่ 4.30



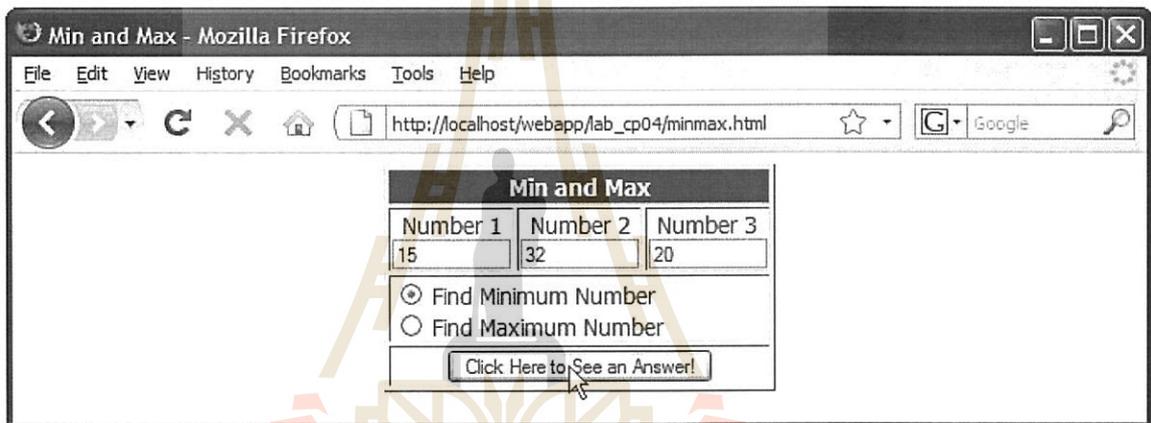
รูปที่ 4.32: ผลการคำนวณและแสดงผลของไฟล์ calculate.php

2. ใช้โปรแกรม Text Editor สร้างไฟล์ html ชื่อว่า “minmax.html” เช่นตัวอย่างดังรูปที่ 4.33



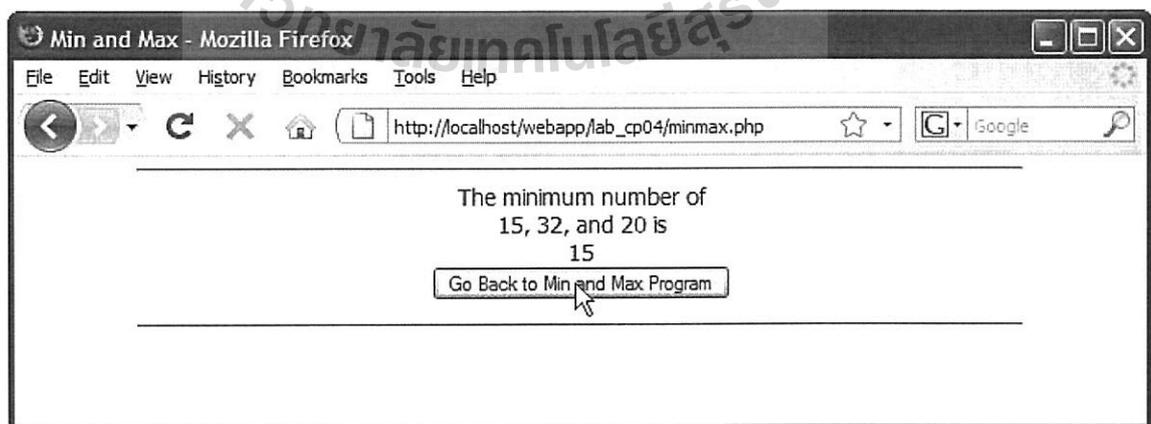
รูปที่ 4.33: ผลการแสดงผลของไฟล์ minmax.html

เมื่อผู้ใช้ป้อนข้อมูลและกดปุ่ม Click Here to See an Answer ดังในรูปที่ 4.34



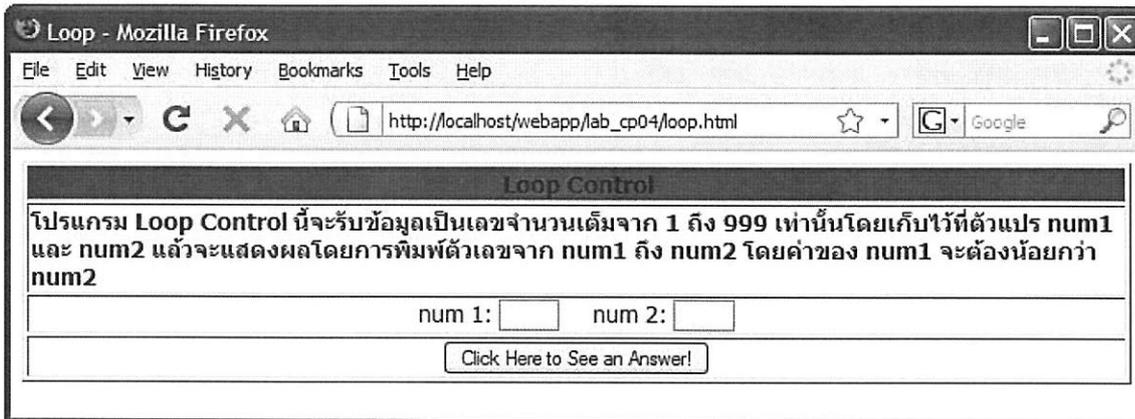
รูปที่ 4.34: การใส่ข้อมูลในแบบฟอร์มของไฟล์ minmax.html

จะส่งข้อมูลทั้งหมดไปที่ไฟล์ “minmax.php” และแสดงผลลัพธ์ออกดังในรูปที่ 4.35 ซึ่งจะมีปุ่ม Go Back to Min and Max Program โดยปุ่มนี้จะเรียกไฟล์ “minmax.html” ขึ้นมาใหม่ เช่นดังรูปที่ 4.33



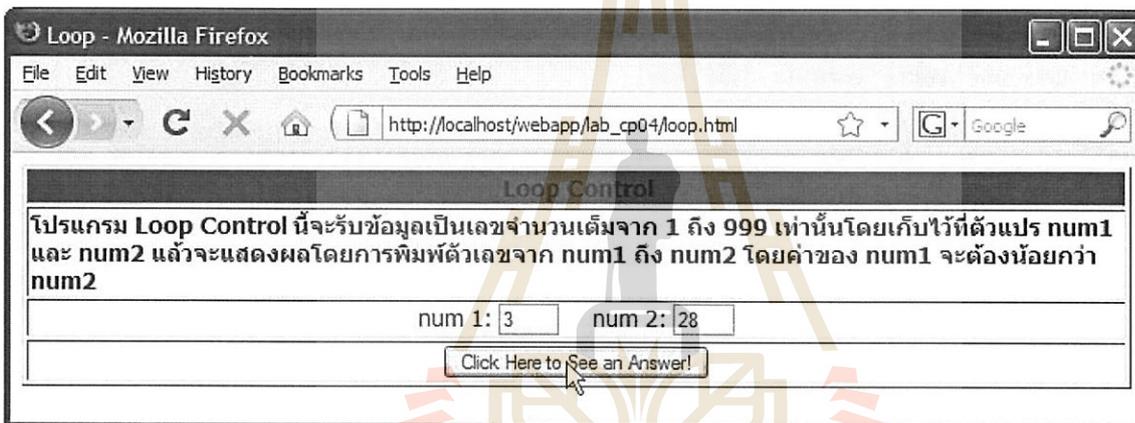
รูปที่ 4.35: ผลการคำนวณและแสดงผลของไฟล์ minmax.php

3. ใช้โปรแกรม Text Editor สร้างไฟล์ html ชื่อว่า "loop.html" เช่นตัวอย่างดังรูปที่ 4.36



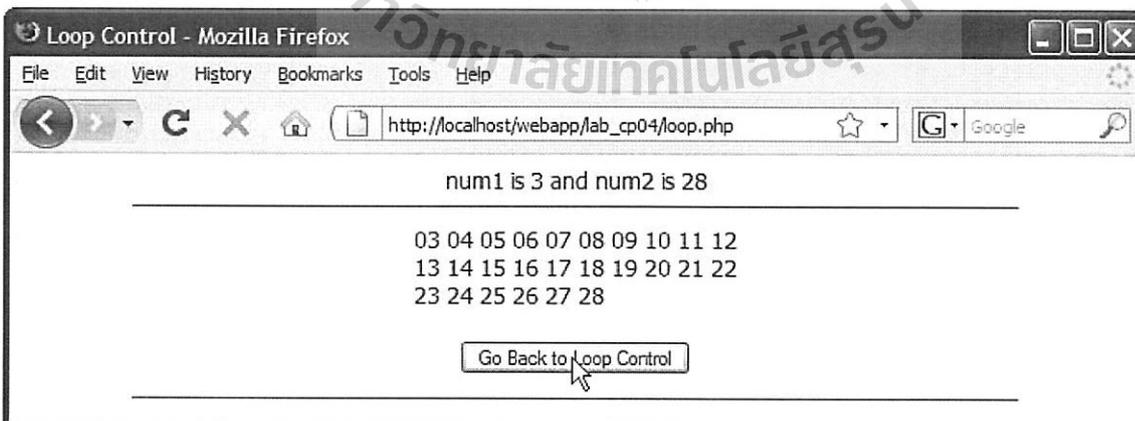
รูปที่ 4.36: ผลการแสดงผลของไฟล์ loop.html

เมื่อผู้ใช้ป้อนข้อมูลและกดปุ่ม Click Here to See an Answer ดังในรูปที่ 4.37



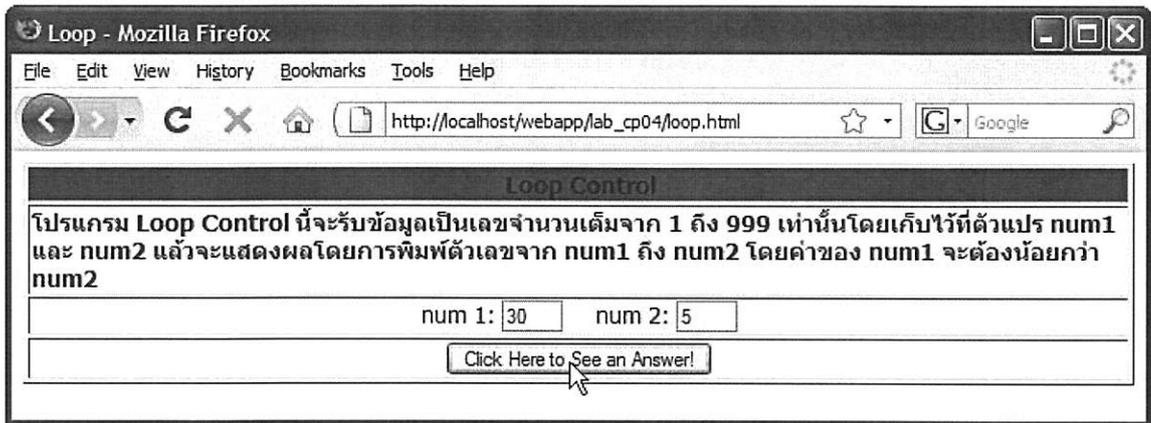
รูปที่ 4.37: การใส่ข้อมูลในแบบฟอร์มของไฟล์ loop.html

จะส่งข้อมูลทั้งหมดไปที่ไฟล์ "loop.php" และแสดงผลลัพธ์ออกดังในรูปที่ 4.38 ซึ่งจะมีปุ่ม Go Back to Loop Control โดยปุ่มนี้จะเรียกไฟล์ "loop.html" ขึ้นมาใหม่ เช่นดังรูปที่ 4.36

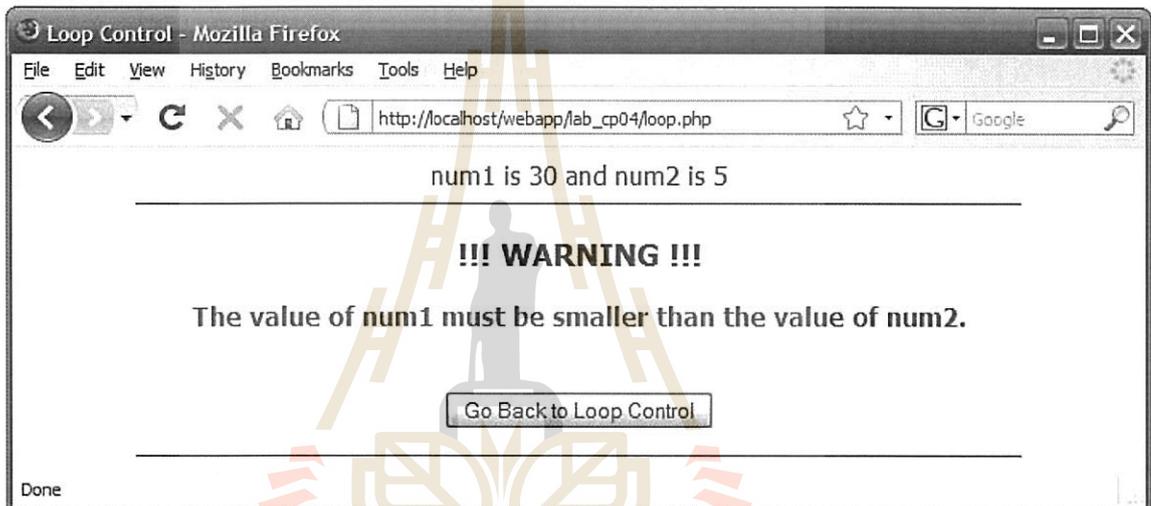


รูปที่ 4.38: ผลการคำนวณและแสดงผลของไฟล์ loop.php

ถ้าค่าของ num1 มากกว่า num2 ดั้งการป้อนข้อมูลในรูปที่ 4.39 จะแสดงผลดังในรูปที่ 4.40



รูปที่ 4.39: การใส่ข้อมูลในแบบฟอร์มของไฟล์ loop.html โดยการป้อนค่า num1 มากกว่า num2



รูปที่ 4.40: ผลการคำนวณและแสดงผลของไฟล์ loop.php เมื่อค่าของ num1 มากกว่า num2



บทที่ 5 ฟังก์ชัน

<Functions>

บทนำ

จากบทที่ผ่านมา จะสังเกตเห็นว่าผู้เขียนโปรแกรมสามารถเขียนคำสั่งต่าง ๆ ในโปรแกรมหนึ่ง ๆ เพื่อให้ประมวลผลหรือแสดงผลตามที่ต้องการ ในคำสั่งนั้น ๆ จะมีการเรียกใช้ฟังก์ชันที่ภาษา PHP ได้สร้าง และกำหนดรูปแบบไว้ให้แล้ว เช่น เมื่อต้องการแสดงเลขจำนวนเต็มจาก 1 ถึง 10 ก็สามารถทำได้โดยใช้คำสั่งดังต่อไปนี้

```
echo '1 2 3 4 5 6 7 8 9 10';
```

หรือถ้าต้องการประกาศตัวแปร \$num ให้เก็บเลขจำนวนเต็ม 1 ถึง 10 ก็สามารถทำได้โดยการเรียกใช้ฟังก์ชัน array() หรือฟังก์ชัน range() นั่นคือ

```
$num = array(1,2,3,4,5,6,7,8,9,10); หรือ
```

```
$num = range(1,10); เป็นต้น
```

การใช้คำสั่งหรือฟังก์ชันต่าง ๆ นั้น จำเป็นต้องเข้าใจในโครงสร้างของคำสั่งหรือรูปแบบของฟังก์ชันนั้น ๆ ซึ่งในภาษา PHP ได้มีฟังก์ชันไว้ให้เลือกใช้มากมาย แต่บางครั้งผู้เขียนโปรแกรมต้องการสร้างฟังก์ชันใหม่ขึ้นมาเอง เพื่อให้สอดคล้องและตรงกับงานมากที่สุด จะได้ประหยัดเวลาและง่ายต่อการจัดการ ปรับปรุง และแก้ไขในภายหลัง ดังนั้นจึงจำเป็นต้องศึกษารูปแบบของคำสั่งและฟังก์ชันต่าง ๆ ที่เกี่ยวข้อง รวมไปถึงวิธีการในการกำหนดรูปแบบฟังก์ชันขึ้นเพื่อใช้งาน

วัตถุประสงค์

1. เพื่อศึกษารูปแบบการทำงานของคำสั่งและฟังก์ชันต่าง ๆ
2. เพื่อทดลองเรียกใช้ฟังก์ชันต่าง ๆ
3. เพื่อให้เข้าใจหลักการและรูปแบบในการสร้างฟังก์ชันใหม่

เนื้อหา

ฟังก์ชันคือกลุ่มของคำสั่งหรือโปรแกรมย่อยโปรแกรมหนึ่ง ที่สามารถทำงานได้ภายในตัวของมันเอง ตามรูปแบบที่ถูกกำหนด โดยจะส่งค่ากลับให้กับฟังก์ชันหลักหรือไม่ก็ขึ้นอยู่กับข้อกำหนดรูปแบบของฟังก์ชันนั้น ๆ ถูกแบ่งออกเป็น 2 ลักษณะใหญ่ ๆ คือ

5.1 ฟังก์ชันที่มีอยู่ในตัวโปรแกรม (Internal or Built-in Functions)

ฟังก์ชันที่มีอยู่ในตัวโปรแกรมภาษา PHP มีมากมาย แต่จะขอกล่าวถึงฟังก์ชันที่พบเห็นและมีการเรียกใช้งานเป็นประจำ ซึ่งในฟังก์ชันหนึ่ง ๆ นั้นจะมีรูปแบบการทำงาน (Function Prototype) ที่แตกต่างกัน

5.1.1 ฟังก์ชันที่เกี่ยวกับวันและเวลา (Date and Time Functions)

ตารางที่ 5.1: ฟังก์ชันที่เกี่ยวกับวันและเวลา

ฟังก์ชัน	รูปแบบ	ความหมาย
checkdate	bool checkdate (int month, int day, int year)	ตรวจสอบวันและเวลา
date_sunrise	mixed date_sunrise (int timestamp [,int format [,float latitude [,float longitude [, float zenith [,float gmt_offset]]]])	ส่งค่ากลับเป็นเวลาพระอาทิตย์ขึ้นของวันและสถานที่ที่ระบุ
date_sunset	mixed date_sunset (int timestamp [,int format [, float latitude [,float longitude [,float zenith [,float gmt_offset]]]])	ส่งค่ากลับเป็นเวลาพระอาทิตย์ตกของวันและสถานที่ที่ระบุ
date	string date (string format [,int timestamp])	ส่งค่ากลับเป็นวัน/เวลา ตามรูปแบบที่ระบุ
getdate	array getdate ((int timestamp))	ดึงข้อมูลรายละเอียดของวัน/เวลา
gettimeofday	mixed gettimeofday ((bool return_float))	ดึงข้อมูลเวลาปัจจุบัน
gmdate	string gmdate (string format [,int timestamp])	วัน/เวลา ในรูปแบบ GMT (Greenwich Mean Time)
gmmktime	int gmmktime ((int hour [,int minute [,int second [,int month [,int day [,int year [,int is_dst]]]])	ดึงข้อมูลรายละเอียดสำหรับวันแบบ GMT
gmstrftime	string gmstrftime (string format [,int timestamp])	จัดรูปแบบวัน/เวลา GMT หรือ UTC ตามวัน/เวลาท้องถิ่น
idate	int idate (string format [,int timestamp])	ส่งค่ากลับเป็นเลขจำนวนเต็มจัดรูปแบบวัน/เวลาท้องถิ่น
localtime	array localtime ((int timestamp [,bool is_associative]))	ดึงข้อมูลรายละเอียดของวัน/เวลาท้องถิ่น
microtime	mixed microtime ((bool get_as_float))	ส่งค่ากลับเป็นไมโครวินาที (1/1000000 วินาที)
mktime	int mktime ((int hour [,int minute [,int second [,int month [,int day [,int year [,int is_dst]]]])	ดึงข้อมูลรายละเอียดสำหรับวัน
strftime	string strftime (string format [,int timestamp])	จัดรูปแบบวัน/เวลาท้องถิ่น โดยใช้ควบคู่กับ setlocale()
time	int time (void)	ส่งค่ากลับเป็นเวลา

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับวันและเวลา

```

1. <?php
2. echo "Today is ".date("l \t\h\e jS \o\f E, Y.")."<br>";
3. echo "Now, it is ".date("H:i:s A")."<br>";
4. echo "December 5, 2007 is on ".date("l", mktime(0, 0, 0, 12, 5, 2007))."<br>";
5. ?>
    
```

```

Today is Saturday the 5th of May, 2007.
Now, it is 18:27:36 PM.
December 5, 2007 is on Wednesday.
    
```

รูปที่ 5.1: การใช้ฟังก์ชันเกี่ยวกับวันเวลา

บรรทัดที่ 2 เป็นการแสดงข้อมูลสตริง "Today is " แล้วต่อท้ายด้วยการเรียกใช้ฟังก์ชัน date() ซึ่งมีตัวเลือกที่เลือกใช้ดังนี้คือ

- | ใช้นำเสนอวันของสัปดาห์แบบเต็ม เช่น Monday เป็นต้น
- \\t\\h\\e ใช้เครื่องหมาย \ (Backslash) เป็น escape character เพื่อแสดงข้อความสตริง "the"
- j ใช้นำเสนอวันที่ของเดือน โดยไม่มีเลขศูนย์นำหน้า เช่น 5, 12 หรือ 25 เป็นต้น

S ใช้นำเสนอตัวต่อท้าย (th, st, nd, หรือ rd) ของวันที่ของเดือน เช่น 1st, 2nd, 3rd, 4th เป็นต้น

\o\f ใช้เครื่องหมาย \ (Backslash) เป็น escape character เพื่อแสดงข้อความสตริง “of”

F ใช้นำเสนอชื่อเดือนแบบเต็ม เช่น January, February หรือ December เป็นต้น

Y ใช้นำเสนอปี ค.ศ. 4 ตัว เช่น 2006, 2007 หรือ 2008 เป็นต้น

บรรทัดที่ 3 เป็นการแสดงข้อมูลสตริง “Now, it is ” แล้วต่อท้ายด้วยการเรียกใช้ฟังก์ชัน date() ซึ่งมีตัวเลือกที่เลือกใช้ดังนี้คือ

H ใช้นำเสนอชั่วโมงในรูปแบบ 24 ชั่วโมง โดยมีเลขศูนย์นำหน้า (00 ถึง 23)

i ใช้นำเสนอนาที โดยมีเลขศูนย์นำหน้า (00 ถึง 59)

s ใช้นำเสนอวินาที โดยมีเลขศูนย์นำหน้า (00 ถึง 59)

A ใช้นำเสนอช่วงเวลาแบบตัวอักษรตัวใหญ่ (AM หรือ PM)

บรรทัดที่ 4 เป็นการแสดงข้อมูลสตริง “December 5, 2007 is on ” แล้วต่อท้ายด้วยการเรียกใช้ฟังก์ชัน date() ซึ่งมีตัวเลือกที่เลือกใช้ดังนี้คือ

l ใช้นำเสนอวันของสัปดาห์แบบเต็ม เช่น Monday เป็นต้น แต่ไม่ใช่วัน ณ ปัจจุบัน เป็นวันที่ต้องประมวลผลมาจากฟังก์ชัน mktime() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ (0, 0, 0, 12, 5, 2007) นั่นคือ เลข 0 (ศูนย์) สามตัวแรกเป็นค่าของ ชั่วโมง นาที และวินาทีตามลำดับ เลข 12 นั้นเป็นค่าของเดือน เลข 5 นั้นเป็นค่าของวัน และเลข 2007 นั้นเป็นค่าของปี ฟังก์ชัน mktime() ทำการประมวลผลโดยตรวจสอบ เดือน 12 วันที่ 5 ปี 2007 แล้วส่งค่าเป็นวันของสัปดาห์แบบเต็ม ตามที่ต้องการ

5.1.2 ฟังก์ชันที่เกี่ยวกับอะเรย์ (Array Functions)

ตารางที่ 5.2: ฟังก์ชันที่เกี่ยวกับอะเรย์

ฟังก์ชัน	รูปแบบ	ความหมาย
array_change_key_case	array array_change_key_case (array input [,int case])	ส่งค่ากลับเป็นอะเรย์ที่มีคีย์เป็นตัวอักษรพิมพ์เล็ก/ใหญ่
array_chunk	array array_chunk (array input,int size [,bool preserve_keys])	แบ่งอะเรย์ออกเป็นส่วน ๆ
array_combine	array array_combine (array keys, array values)	สร้างอะเรย์ขึ้นใหม่โดยการรวมเอาอะเรย์หนึ่งอะเรย์เป็นคีย์และอีกหนึ่งอะเรย์เป็นค่า
array_count_values	array array_count_values (array input)	นับค่าทั้งหมดในอะเรย์
array_diff	array array_diff (array array1, array array2 [,array ...])	ประมวลผลหาความแตกต่างของอะเรย์
array_fill	array array_fill (int start_index, int num, mixed value)	เติมค่าต่าง ๆ ให้กับอะเรย์
array_flip	array array_flip (array trans)	สลับคีย์กับค่าที่สัมพันธ์กันทั้งหมดในอะเรย์
array_merge	array array_merge (array array1 [,array array2 [,array ...]])	ทำการรวมอะเรย์เข้าด้วยกัน
array_multisort	bool array_multisort (array ar1 [,mixed arg [,mixed ... [,array ...]])	เรียงลำดับอะเรย์หลาย ๆ อะเรย์ หรือหลาย ๆ มิติ
array_pad	array array_pad (array input, int pad_size, mixed pad_value)	เพิ่มข้อมูลตามจำนวนและค่าที่ระบุ
array_pop	mixed array_pop (array &array)	ลบข้อมูลตัวสุดท้ายในอะเรย์ทิ้ง
array_push	int array_push (array &array, mixed var [,mixed ...])	ใส่ข้อมูลใหม่เพิ่มในอะเรย์โดยการใส่ต่อท้ายข้อมูลเดิม
array_rand	mixed array_rand (array input [, int num_req])	สุ่มเลือกค่าใดค่าหนึ่งออกจากอะเรย์
array_reduce	mixed array_reduce (array input, callback function [, int initial])	ทำการรวมข้อมูลทั้งหมดในอะเรย์ให้เป็นค่าเดียว

ฟังก์ชัน	รูปแบบ	ความหมาย
array_reverse	array array_reverse (array array [, bool preserve_keys])	เรียงลำดับข้อมูลในอะเรย์แบบย้อนกลับ
array_search	mixed array_search (mixed needle, array haystack [, bool strict])	ค้นหาข้อมูลในอะเรย์จากค่าที่กำหนดให้แล้วส่งคืนกลับถ้าค้นพบ
array_shift	mixed array_shift (array &array)	ลบข้อมูลตัวแรกในอะเรย์ทิ้ง
array_slice	array array_slice (array array, int offset [, int length [, bool preserve_keys]])	ตัดทอนข้อมูลในอะเรย์ตามตำแหน่งและจำนวนที่จะระบุ
array_splice	array array_splice (array &input, int offset [, int length [, array replacement]])	ลบข้อมูลบางส่วนในอะเรย์ทิ้งแล้วทับด้วยข้อมูลใหม่
array_sum	number array_sum (array array)	หาผลบวกของค่าทั้งหมดในอะเรย์
array	array array ([mixed ...])	สร้างข้อมูลอะเรย์
arsort	bool arsort (array &array [, int sort_flags])	เรียงลำดับข้อมูลในอะเรย์แบบย้อนกลับโดยให้ตัวชี้ (index) อยู่ในตำแหน่งเดิม
asort	bool asort (array &array [, int sort_flags])	เรียงลำดับข้อมูลในอะเรย์โดยให้ตัวชี้ (index) อยู่ในตำแหน่งเดิม
compact	array compact (mixed varname [, mixed ...])	สร้างอะเรย์ประกอบไปด้วยตัวแปรและค่าของมัน
count	int count (mixed var [, int mode])	นับจำนวนในอะเรย์
current	mixed current (array &array)	ส่งกลับค่าปัจจุบันในอะเรย์
each	array each (array &array)	ส่งกลับคีย์และค่าปัจจุบันในอะเรย์และเลื่อนตัวชี้อะเรย์ไปข้างหน้า
end	mixed end (array &array)	ตั้งค่าของตัวชี้ภายในอะเรย์ให้ชี้ไปที่ตำแหน่งสุดท้ายของข้อมูล
extract	int extract (array var_array [, int extract_type [, string prefix]])	ส่งตัวแปรเข้าในตารางสัญลักษณ์ปัจจุบันจากอะเรย์
in_array	bool in_array (mixed needle, array haystack [, bool strict])	ตรวจสอบค่าในอะเรย์ว่ามีอยู่หรือไม่?
key	mixed key (array &array)	ดึงคีย์จากอะเรย์สัมพันธ์
ksort	bool ksort (array &array [, int sort_flags])	เรียงลำดับข้อมูลในอะเรย์โดยคีย์แบบย้อนกลับ
ksort	bool ksort (array &array [, int sort_flags])	เรียงลำดับข้อมูลในอะเรย์โดยคีย์
list	void list (mixed varname, mixed ...)	กำหนดชื่อตัวแปรให้กับข้อมูลในอะเรย์
natcasesort	bool natcasesort (array &array)	เรียงลำดับข้อมูลในอะเรย์แบบธรรมชาติ โดยไม่สนใจตัวเล็ก-ใหญ่
natsort	bool natsort (array &array)	เรียงลำดับข้อมูลในอะเรย์แบบธรรมชาติ
next	mixed next (array &array)	เลื่อนตำแหน่งตัวชี้ข้อมูลในอะเรย์ไปข้างหน้า
pos	mixed pos (array &array)	ส่งกลับค่าปัจจุบันในอะเรย์
prev	mixed prev (array &array)	เลื่อนตำแหน่งตัวชี้ข้อมูลในอะเรย์ไปข้างหลัง (ตำแหน่งที่ผ่านมา)
range	array range (mixed low, mixed high [, number step])	สร้างอะเรย์ที่มีข้อมูลอยู่ในช่วงที่กำหนด
reset	mixed reset (array &array)	กำหนดให้ตำแหน่งตัวชี้ข้อมูลในอะเรย์ไปอยู่ที่จุดเริ่มต้น
rsort	bool rsort (array &array [, int sort_flags])	เรียงลำดับข้อมูลในอะเรย์แบบย้อนกลับ
shuffle	bool shuffle (array &array)	สุ่มข้อมูลในอะเรย์
sizeof	int sizeof (mixed var [, int mode])	นับจำนวนในอะเรย์ (มีการทำงานเหมือนกับ count)
sort	bool sort (array &array [, int sort_flags])	เรียงลำดับข้อมูลในอะเรย์
uasort	bool uasort (array &array, callback cmp_function)	เรียงลำดับข้อมูลในอะเรย์โดยผู้ใช้กำหนดการเปรียบเทียบและยึดความสัมพันธ์ตัวเดิม
uksort	bool uksort (array &array, callback cmp_function)	เรียงลำดับข้อมูลในอะเรย์โดยผู้ใช้กำหนดการเปรียบเทียบ
usort	bool usort (array &array, callback cmp_function)	เรียงลำดับข้อมูลในอะเรย์โดยค่า ซึ่งผู้ใช้กำหนดการเปรียบเทียบ

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับอะเรย์ในการเก็บค่า คีย์ และแสดงผล

```

1. <?php
2.   $a = array(1, 3, 5, 7, 9);
3.   echo 'Sum of all values in $a = '.array_sum($a).'\n';
4.   $b = array("a">1.2, "b">1.3, "c">1.5);
5.   echo 'Sum of all values in $b = '.array_sum($b).'\n';
6.   echo 'Array $a in original order: ';
7.   foreach($a as $val){
8.     echo "$val ";
9.   }
10.  $a = array_reverse($a);
11.  echo '\nArray $a in reversed order: ';
12.  foreach($a as $val){
13.    echo "$val ";
14.  }
15.  $key = array_search(3, $a);
16.  echo '\nFound number 3 at index '.$key;
17.
18.  echo '\nDisplay "key" of array $b: ';
19.  print_r(array_keys($b));
20.  echo '\nDisplay "value" of array $b: ';
21.  print_r(array_values($b));
22.
23.  echo '\nDisplay shuffle numbers from 1 to 10: ';
24.  $numbers = range(1,10);
25.  shuffle($numbers);
26.  foreach ($numbers as $number) {
27.    echo "$number ";
28.  }
29.  ?>

```

```

Sum of all values in $a = 25
Sum of all values in $b = 4
Array $a in original order: 1 3 5 7 9
Array $a in reversed order: 9 7 5 3 1
Found number 3 at index 3
Display "key" of array $b: Array ( [0] => a [1] => b [2] => c )
Display "value" of array $b: Array ( [0] => 1.2 [1] => 1.3 [2] => 1.5 )
Display shuffle numbers from 1 to 10: 7 3 2 8 9 6 4 5 1 10

```

รูปที่ 5.2: การใช้ฟังก์ชันเกี่ยวกับอะเรย์ในการเก็บค่า คีย์ และแสดงผล

บรรทัดที่ 2 เป็นการกำหนดข้อมูลให้กับตัวแปร \$a ที่เป็นอะเรย์ ประกอบไปด้วย 1, 3, 5, 7 และ 9 โดยผ่านฟังก์ชัน array()

บรรทัดที่ 3 เป็นการแสดงข้อมูลสตริง "Sum of all values in \$a = " แล้วต่อท้ายด้วยการเรียกใช้ฟังก์ชัน array_sum() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ ตัวแปร \$a ฟังก์ชัน array_sum() ก็ทำการประมวลผลโดยหาผลบวกของข้อมูลทั้งหมดในตัวแปร \$a ผลที่ได้คือ 25 ดังนั้น 25 จึงถูกแสดงออกต่อท้ายข้อความดังกล่าวแล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 4 เป็นการกำหนดข้อมูลให้กับตัวแปร \$b ที่เป็นอะเรย์สัมพันธ์ ประกอบไปด้วย คีย์ "a" มีค่าเท่ากับ 1.2 คีย์ "b" มีค่าเท่ากับ 1.3 และคีย์ "c" มีค่าเท่ากับ 1.5

บรรทัดที่ 5 เป็นการแสดงข้อมูลสตริง "Sum of all values in \$b = " แล้วต่อท้ายด้วยการเรียกใช้ฟังก์ชัน array_sum() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ ตัวแปร \$b ฟังก์ชัน array_sum() ก็ทำการประมวลผลโดยหาผลบวกของข้อมูลทั้งหมดในตัวแปร \$b ผลที่ได้คือ 4 ดังนั้น 4 จึงถูกแสดงออกต่อท้ายข้อความดังกล่าวแล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 6 เป็นการแสดงข้อมูลสตริง "Array \$a in original order: "

บรรทัดที่ 7 - 9 เป็นการใช้ฟังก์ชัน foreach() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ \$a as \$val ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ \$a ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ \$a แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร \$val แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 8 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร \$val ออกที่จอภาพ แล้วกลับไปบรรทัดที่ 7 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 7 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 7 - 9 คือ 1 3 5 7 9

บรรทัดที่ 10 เป็นการกำหนดค่าให้กับตัวแปร \$a โดยค่านั้นมาจากการเรียกใช้ฟังก์ชัน array_reverse() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ \$a ทำให้ข้อมูลที่อยู่ในอะเรย์ \$a ถูกเรียงลำดับแบบย้อนกลับ จึงกลายเป็น 9 7 5 3 1 ดังนั้นตอนนี้ตัวแปร \$a จึงเก็บค่า 9 7 5 3 1

บรรทัดที่ 11 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ แล้วแสดงข้อมูลสตริง "Array \$a in reversed order: " ออกที่จอภาพ

บรรทัดที่ 12 - 14 มีการทำงานเหมือนกับบรรทัดที่ 7 - 9 คือวนลูปเพื่อแสดงข้อมูลที่อยู่ในตัวแปร \$a นั่นคือ 9 7 5 3 1 ออกที่จอภาพ

บรรทัดที่ 15 เป็นการกำหนดค่าให้กับตัวแปร \$key โดยค่านั้นมาจากการเรียกใช้ฟังก์ชัน array_search() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ 3 และตัวแปร \$a ทำให้เกิดการค้นหาตำแหน่งของเลข 3 ในอะเรย์ \$a ซึ่งคือ 3 นั่นเอง (9 อยู่ที่ตำแหน่ง 0, 7 อยู่ที่ตำแหน่ง 1, 5 อยู่ที่ตำแหน่ง 2, 3 อยู่ที่ตำแหน่ง 3 และ 1 อยู่ที่ตำแหน่ง 4)

บรรทัดที่ 16 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ แล้วแสดงข้อมูลสตริง "Found number 3 at index " ตามด้วยข้อมูลที่อยู่ในตัวแปร \$key นั่นคือ 3 ออกที่จอภาพ

บรรทัดที่ 18 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ แล้วแสดงข้อมูลสตริง "Display "key" of array \$b: " ออกที่จอภาพ

บรรทัดที่ 19 สิ่งแรกที่จะถูกทำคือสิ่งที่อยู่ภายในฟังก์ชัน print_r() นั่นคือ array_keys(\$b) ซึ่งจะทำการประมวลผลหาคีย์ทั้งหมดในอะเรย์ \$b (คีย์ a, b และ c) ต่อจากนั้นฟังก์ชัน print_r() ก็ทำการแสดงผลโดยแสดงในรูปแบบของอะเรย์สัมพันธ์ ซึ่งมีคีย์เริ่มจาก 0 และค่าที่จะถูกแสดงออกคือค่าของคีย์ที่ได้จาก array_keys(\$b) ดังนั้นผลที่ถูกแสดงออกคือ Array ([0] => a [1] => b [2] => c)

บรรทัดที่ 20 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ แล้วแสดงข้อมูลสตริง "Display "value" of array \$b: " ออกที่จอภาพ

บรรทัดที่ 21 สิ่งแรกที่จะถูกทำคือสิ่งที่อยู่ภายในฟังก์ชัน print_r() นั่นคือ array_values(\$b) ซึ่งจะทำการประมวลผลหาค่าทั้งหมดในอะเรย์ \$b (ค่า 1.2, 1.3 และ 1.5) ต่อจากนั้นฟังก์ชัน print_r() ก็ทำการแสดงผลโดยแสดงในรูปแบบของอะเรย์สัมพันธ์ ซึ่งมีคีย์เริ่มจาก 0 และค่าที่จะถูกแสดงออกคือค่าของค่าที่ได้จาก array_values(\$b) ดังนั้นผลที่ถูกแสดงออกคือ Array ([0] => 1.2 [1] => 1.3 [2] => 1.5)

บรรทัดที่ 23 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ แล้วแสดงข้อมูลสตริง "Display shuffle numbers from 1 to 10: " ออกที่จอภาพ

บรรทัดที่ 24 เป็นการกำหนดค่าให้กับตัวแปร \$numbers โดยค่านั้นมาจากการเรียกใช้ฟังก์ชัน range(1,10) ซึ่งคือค่า 1, 2, 3, 4, 5, 6, 7, 8, 9 และ 10 ตามลำดับ

บรรทัดที่ 25 เป็นการเรียกใช้ฟังก์ชัน shuffle() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยตัวแปร \$numbers ทำให้เกิดการสลับเปลี่ยนตำแหน่งข้อมูลที่อยู่ในอะเรย์ \$numbers

บรรทัดที่ 26 – 28 เป็นการเรียกใช้ฟังก์ชัน `foreach()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ `$numbers as $number` ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ `$numbers` ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ `$numbers` แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร `$number` แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 27 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร `$number` ออกที่จอภาพ แล้วกลับไปบรรทัดที่ 26 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 26 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 26 – 28 คือ 7 3 2 8 9 6 4 5 1 10 (ผลที่ได้นี้จะมีค่าที่แตกต่างกันออกไปทุกครั้งที่เราไปทำงาน)

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับอะเรย์ในการเพิ่มค่า ตัดค่า และแสดงผล

<pre> 1. <?php 2. \$a = array(1, 3, 5); 3. \$result = array_pad(\$a, 5, 0); 4. foreach(\$result as \$item){ 5. echo "\$item "; 6. } 7. echo '
'; 8. \$result = array_pad(\$a, -5, 0); 9. foreach(\$result as \$item){ 10. echo "\$item "; 11. } 12. echo '
'; 13. array_push(\$a, 9); 14. foreach(\$a as \$item){ 15. echo "\$item "; 16. } 17. echo '
'; 18. \$str = array("One", "Two", "Three", "Four"); 19. array_shift(\$str); 20. foreach(\$str as \$item){ 21. echo "\$item "; 22. } 23. ?></pre>		<pre> 1 3 5 0 0 0 0 1 3 5 1 3 5 9 Two Three Four</pre>
--	--	--

รูปที่ 5.3: การใช้ฟังก์ชันเกี่ยวกับอะเรย์ในการเพิ่มค่า ตัดค่า และแสดงผล

บรรทัดที่ 2 เป็นการกำหนดข้อมูลให้กับตัวแปร `$a` ที่เป็นอะเรย์ ประกอบด้วย 1, 3 และ 5 โดยผ่านฟังก์ชัน `array()`

บรรทัดที่ 3 เป็นการกำหนดค่าให้กับตัวแปร `$result` โดยค่านั้นมาจากการเรียกใช้ฟังก์ชัน `array_pad($a, 5, 0)` ซึ่งก็คือการนำค่าที่อยู่ในตัวแปร `$a` (คือค่า 1, 3, และ 5) มายึดออกจากเดิม 3 ไปทางขวามือของข้อมูลเดิมเป็น 5 (เนื่องจากเป็นค่านบวก) โดยเอาค่า 0 ไปใส่ในตำแหน่งที่ยึดออกไป ค่าที่ตัวแปร `$result` เก็บก็คือ 1, 3, 5, 0, และ 0 ตามลำดับ

บรรทัดที่ 4 – 6 เป็นการเรียกใช้ฟังก์ชัน `foreach()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ `$result as $item` ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ `$result` ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ `$result` แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร `$item` แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 5 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร `$item` ออกที่จอภาพ แล้วกลับไปบรรทัดที่ 4 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 4 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 4 – 6 คือ 1 3 5 0 0

บรรทัดที่ 7 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 8 เป็นการกำหนดค่าให้กับตัวแปร `$result` โดยค่านั้นมาจากการเรียกใช้ฟังก์ชัน `array_pad($a, -5, 0)` ซึ่งก็คือการนำค่าที่อยู่ในตัวแปร `$a` (คือค่า 1, 3, และ 5) มายึดออกจากเดิม 3

ไปทางซ้ายมือของข้อมูลเดิมเป็น 5 (เนื่องจากเป็นค่าลบ) โดยเอาค่า 0 ไปใส่ในตำแหน่งที่ยึดออกไป ค่าที่ตัวแปร \$result เก็บก็คือ 0, 0, 1, 3 และ 5 ตามลำดับ

บรรทัดที่ 9 – 11 เป็นการเรียกใช้ฟังก์ชัน `foreach()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ `$result as $item` ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ `$result` ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ `$result` แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร `$item` แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 10 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร `$item` ออกที่จอภาพ แล้วกลับไปบรรทัดที่ 9 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 9 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 9 – 11 คือ 0 0 1 3 5

บรรทัดที่ 12 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 13 เป็นการเรียกใช้ฟังก์ชัน `array_push()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ ตัวแปร `$a` และ 9 ทำให้เกิดการเพิ่มข้อมูลคือเลข 9 ต่อท้ายข้อมูลที่มีอยู่ในตัวแปร `$a` ทำให้ตัวแปร `$a` เก็บค่า 1, 3, 5, และ 9 ตามลำดับ

บรรทัดที่ 14 – 16 เป็นการเรียกใช้ฟังก์ชัน `foreach()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ `$a as $item` ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ `$a` ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ `$a` แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร `$item` แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 15 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร `$item` ออกที่จอภาพ แล้วกลับไปบรรทัดที่ 14 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 14 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 14 – 16 คือ 1 3 5 9

บรรทัดที่ 17 เป็นการเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 18 เป็นการกำหนดข้อมูลให้กับตัวแปร `$str` ที่เป็นอะเรย์ ประกอบด้วยสตริง “One”, “Two”, “Three” และ “Four” ตามลำดับโดยผ่านฟังก์ชัน `array()`

บรรทัดที่ 19 เป็นการเรียกใช้ฟังก์ชัน `array_shift()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ ตัวแปร `$str` ทำให้เกิดการตัดข้อมูลตัวแรกในอะเรย์ `$str` ทิ้ง ดังนั้นข้อมูลในอะเรย์ `$str` จึงกลายเป็น “Two”, “Three” และ “Four” ตามลำดับ

บรรทัดที่ 20 – 22 เป็นการเรียกใช้ฟังก์ชัน `foreach()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ `$str as $item` ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ `$str` ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ `$str` แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร `$item` แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 21 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร `$item` ออกที่จอภาพ แล้วกลับไปบรรทัดที่ 20 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 20 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 20 – 22 คือ Two Three Four

5.1.3 ฟังก์ชันที่เกี่ยวกับชนิดตัวอักษร (Character Type Functions)

ตารางที่ 5.3: ฟังก์ชันที่เกี่ยวกับชนิดตัวอักษร

ฟังก์ชัน	รูปแบบ	ความหมาย
ctype_alnum	bool ctype_alnum (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษร (a-z, A-Z) หรือตัวเลข?
ctype_alpha	bool ctype_alpha (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษร (a-z, A-Z)?
ctype_cntrl	bool ctype_cntrl (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรควบคุม (control)?
ctype_digit	bool ctype_digit (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวเลข (0 - 9)?
ctype_graph	bool ctype_graph (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรที่สามารถพิมพ์ได้ (ไม่รวมช่องว่าง)?
ctype_lower	bool ctype_lower (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรเล็ก (a-z)?
ctype_print	bool ctype_print (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรที่สามารถพิมพ์ได้ (รวมช่องว่าง)?
ctype_punct	bool ctype_punct (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรที่สามารถพิมพ์ได้ที่ไม่ใช่ตัวอักษรหรือตัวเลข?
ctype_space	bool ctype_space (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรที่สร้างช่องว่าง (space)?
ctype_upper	bool ctype_upper (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรใหญ่ (A-Z)?
ctype_xdigit	bool ctype_xdigit (string text)	ตรวจสอบว่าตัวอักษรทุกตัวในสตริงเป็นตัวอักษรของเลขฐานสิบหก (0-9, a-f หรือ A-F)?

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับชนิดตัวอักษร

```

1.  <?php
2.  $str = array("SUT Korat 2007", "Yahoo1999DotCom");
3.  foreach ($str as $s){
4.      echo "The string \"$s\" ";
5.      if(ctype_alnum($s)){
6.          echo "consists of all letters or digits.<br>";
7.      }else{
8.          echo "does not consist of all letters or digits.<br>";
9.      }
10. }
11. ?>
    
```

The string "SUT Korat 2007" does not consist of all letters or digits.
The string "Yahoo1999DotCom" consists of all letters or digits.

รูปที่ 5.4: การใช้ฟังก์ชันเกี่ยวกับชนิดตัวอักษร

บรรทัดที่ 2 เป็นการกำหนดข้อมูลให้กับตัวแปร \$str ที่เป็นอะเรย์ ประกอบไปด้วยสตริง "SUT Korat 2007" และ "Yahoo1999DotCom" โดยผ่านฟังก์ชัน array()

บรรทัดที่ 3 - 10 เป็นการเรียกใช้ฟังก์ชัน foreach() ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ \$str as \$s ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ \$str ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ \$str แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร \$s แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 4 แสดงข้อมูลสตริง "The string" ตามด้วยเครื่องหมาย " (Open Double Quotation) ตามด้วยข้อมูลที่อยู่ในตัวแปร \$s นั่นคือสตริง SUT Korat 2007 แล้วตามด้วยเครื่องหมาย " (Close Double Quotation) ออกที่จอภาพ

บรรทัดที่ 5 เป็นการใช้นิพจน์ if เพื่อทดสอบตัวอักษรทุกตัวที่อยู่ในตัวแปร \$s ด้วยฟังก์ชัน ctype_alnum() นั่นคือทดสอบว่าตัวอักษรทุกตัวที่อยู่ในตัวแปร \$s นั้นเป็นตัวอักษร (a-z หรือ A-Z) หรือเป็นตัวเลขเท่านั้น โดยไม่เป็นตัวอักษรอื่นใด ผลการทดสอบจึงเป็นเท็จ เนื่องจากมีตัวอักษรที่เป็นช่องว่างอยู่

ด้วย ดังนั้นจึงไปทำคำสั่งในบรรทัดที่ 8 ซึ่งแสดงข้อความว่าเป็นสตริง “does not consist of all letters or digits.” แล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ หลังจากนั้นก็จะกลับไปทำงานที่บรรทัดที่ 3 ทำให้ตัวแปร \$s เก็บข้อมูลสตริง Yahoo1999DotCom พอเลื่อนไปที่บรรทัดที่ 4 แสดงข้อมูลสตริง “The string “” ตามด้วยข้อมูลที่อยู่ในตัวแปร \$s นั่นคือสตริง Yahoo1999DotCom แล้วตามด้วยเครื่องหมาย ” (Close Double Quotation) ออกที่จอภาพ เมื่อไปถึงบรรทัดที่ 5 ก็จะทำการทดสอบอีกครั้ง คราวนี้ผลที่ได้เป็นจริง เนื่องจากตัวอักษรทุกตัวเป็นตัวอักษร (a-z หรือ A-Z) หรือเป็นตัวเลข ดังนั้นจึงไปทำคำสั่งในบรรทัดที่ 6 ซึ่งแสดงข้อความว่าเป็นสตริง “consists of all letters or digits.” แล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

5.1.4 ฟังก์ชันเกี่ยวกับไดเรกทอรี (Directory Functions)

ตารางที่ 5.4: ฟังก์ชันเกี่ยวกับไดเรกทอรี

ฟังก์ชัน	รูปแบบ	ความหมาย
chdir	bool chdir (string directory)	เปลี่ยนไดเรกทอรีที่กำลังทำงานอยู่ให้เป็นไดเรกทอรีที่กำหนด
chroot	bool chroot (string directory)	เปลี่ยนรูทไดเรกทอรีที่กำลังทำงานอยู่ให้เป็นไดเรกทอรีที่กำหนด
dir	class dir (string director)	ชั้นของไดเรกทอรี
closedir	void closedir (resource dir_handle)	ปิดไดเรกทอรี
getcwd	string getcwd (void)	แสดงไดเรกทอรีที่กำลังทำงานอยู่
opendir	resource opendir (string path [,resource context])	เปิดไดเรกทอรี
readdir	string readdir (resource dir_handle)	อ่านไดเรกทอรี
rewinddir	void rewinddir (resource dir_handle)	เลื่อนตัวชี้กลับไปจุดเริ่มต้นของไดเรกทอรี
scandir	array scandir (string directory [,int sorting_order [,resource context]])	แสดงไฟล์และไดเรกทอรีภายในเส้นทางที่กำหนด

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับไดเรกทอรี

```

1. <?php
2.     echo getcwd().'\<br>';
3.     $dir = 'D:\AppServ\www\php_book';
4.     $files = scandir($dir);
5.     foreach($files as $value){
6.         echo "$value<br>";
7.     }
8.     echo getcwd().'\<br>';
9.     chdir('../cp4');
10.    echo getcwd().'\<br>';
11.    ?>

```

D:\AppServ\www\php_book\cp5
 .
 ..
 cp3
 cp4
 cp5
 D:\AppServ\www\php_book\cp5
 D:\AppServ\www\php_book\cp4

รูปที่ 5.5: การใช้ฟังก์ชันเกี่ยวกับไดเรกทอรี

บรรทัดที่ 2 เป็นการเรียกใช้ฟังก์ชัน `getcwd()` ซึ่งเป็นการแสดงไดเรกทอรีที่กำลังทำงานอยู่ ออกที่จอภาพ

บรรทัดที่ 3 เป็นการกำหนดค่าที่เป็นสตริง “D:\AppServ\www\php_book” ให้กับตัวแปร `$dir`

บรรทัดที่ 4 เป็นการกำหนดข้อมูลให้กับตัวแปร `$files` โดยการเรียกใช้ฟังก์ชัน `scandir()` แล้วส่งอาร์กิวเมนต์คือตัวแปร `$dir`

บรรทัดที่ 5 – 7 เป็นการเรียกใช้ฟังก์ชัน `foreach()` ซึ่งมีการส่งอาร์กิวเมนต์ไปด้วยคือ `$files as $value` ทำให้ตัวชี้ตำแหน่งภายในอะเรย์ `$files` ชี้ไปที่ตำแหน่งแรกของข้อมูลในอะเรย์ `$files` แล้วนำข้อมูลนั้นมาเก็บไว้ที่ตัวแปร `$value` แล้วเลื่อนตัวชี้ตำแหน่งไปยังตำแหน่งถัดไป

บรรทัดที่ 6 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร `$value` ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่ แล้วกลับไปบรรทัดที่ 5 ทำเช่นนี้ไปเรื่อย ๆ จนกว่าอาร์กิวเมนต์ในบรรทัดที่ 5 จะเป็นเท็จ ผลที่ได้จากการทำงานของคำสั่งในบรรทัดที่ 5 – 7 คือ

```

.
.
cp3
cp4
cp5
    
```

บรรทัดที่ 8 เป็นการเรียกใช้ฟังก์ชัน `getcwd()` ซึ่งเป็นการแสดงไดเรกทอรีที่กำลังทำงานอยู่ ออกที่จอภาพ ผลที่ได้ก็ยังคงเป็นไดเรกทอรีเดิมคือ `D:\AppServ\www\php_book\cp5` เนื่องจากยังไม่มีการเปลี่ยนไดเรกทอรีไปที่อื่น พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 9 เป็นการเรียกใช้ฟังก์ชัน `chdir()` พร้อมกับส่งอาร์กิวเมนต์คือสตริง `“..\cp4”` ซึ่งเป็นการเปลี่ยนไดเรกทอรีไปเป็น `cp4`

บรรทัดที่ 10 เป็นการเรียกใช้ฟังก์ชัน `getcwd()` ซึ่งเป็นการแสดงไดเรกทอรีที่กำลังทำงานอยู่ ออกที่จอภาพ ผลที่ได้ก็คือ `D:\AppServ\www\php_book\cp4` (เนื่องจากผลของการทำงานในบรรทัดที่ 9) พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

5.1.5 ฟังก์ชันเกี่ยวกับสตริง (String Functions)

ตารางที่ 5.5: ฟังก์ชันเกี่ยวกับสตริง

ฟังก์ชัน	รูปแบบ	ความหมาย
<code>addslashes</code>	<code>string addslashes (string str, string charlist)</code>	เพิ่มเครื่องหมาย \ (Backslash) แบบภาษาซี
<code>addslashes</code>	<code>string addslashes (string str)</code>	เพิ่มเครื่องหมาย \ (Backslash)
<code>bin2hex</code>	<code>string bin2hex (string str)</code>	แปลงค่าของเลขฐานสองให้เป็นฐานสิบหก
<code>chop</code>	<code>string chop (string str [, string charlist])</code>	ตัดช่องว่างหรือตัวอักษรอื่นที่ระบุออกจากจุดสิ้นสุดของสตริง
<code>chr</code>	<code>string chr (int ascii)</code>	ส่งค่ากลับเป็นตัวอักษรที่มีค่าแอสกีที่ระบุ
<code>chunk_split</code>	<code>string chunk_split (string body [, int chunklen [, string end]])</code>	แยกข้อมูลในสตริงออกเป็นส่วย่อย ๆ
<code>count_chars</code>	<code>mixed count_chars (string string [, int mode])</code>	ส่งค่ากลับเป็นรายละเอียดของตัวอักษรตามที่ระบุในสตริง
<code>crypt</code>	<code>string crypt (string str [, string salt])</code>	เปลี่ยนข้อมูลในสตริงโดยการเข้ารหัสให้เป็นโค้ด ๆ หนึ่ง
<code>echo</code>	<code>void echo (string arg1 [, string ...])</code>	แสดงสตริงออกที่จอภาพ
<code>explode</code>	<code>array explode (string delimiter, string string [, int limit])</code>	แยกข้อมูลในสตริงออกจากกันโดยใช้ตัวแบ่ง (Delimiter) เป็นตัวแยก
<code>fprintf</code>	<code>int fprintf (resource handle, string format [, mixed args [, mixed ...]])</code>	เขียนสตริงที่มีการจัดรูปแบบไปที่สตรีมที่ระบุ โดยรับตัวแปรระบุจำนวนของอาร์กิวเมนต์
<code>htmlspecialchars</code>	<code>string htmlspecialchars (string string [, int quote_style [, string charset]])</code>	แปลงตัวอักษรพิเศษเป็นกลุ่มตัวอักษรของ HTML
<code>html_entity_decode</code>	<code>string html_entity_decode (string string [, int quote_style [, string charset]])</code>	แปลงกลุ่มตัวอักษรของ HTML ทุกตัวเป็นตัวอักษรปกติ
<code>htmlentities</code>	<code>string htmlentities (string string [, int quote_style [, string charset]])</code>	แปลงตัวอักษรปกติเป็นกลุ่มตัวอักษรของ HTML
<code>htmlspecialchars_decode</code>	<code>string htmlspecialchars_decode (string string [, int quote_style])</code>	แปลงกลุ่มตัวอักษรพิเศษของ HTML เป็นตัวอักษรปกติ
<code>implode</code>	<code>string implode (string glue, array pieces)</code>	รวมอะเรย์เข้าด้วยกันโดยมีตัวเชื่อม (glue) กั้น

ฟังก์ชัน	รูปแบบ	ความหมาย
join	string join (string glue, array pieces)	รวมอะเรย์เข้าด้วยกันโดยมีตัวเชื่อม (glue) กั้น
ltrim	string ltrim (string str [, string charlist])	ตัดช่องว่างหรือตัวอักษรอื่นที่ระบุออกจากจุดเริ่มต้นของสตริง
md5_file	string md5_file (string filename [, bool raw_output])	คำนวณค่า md5 แแฮช ของไฟล์
md5	string md5 (string str [, bool raw_output])	คำนวณค่า md5 แแฮช ของสตริง
money_format	string money_format (string format, float number)	จัดรูปแบบตัวเลขให้อยู่ในรูปแบบของสตริงที่เป็นเงินตรา
nl2br	string nl2br (string string)	ใส่แท็ก ก่อนเริ่มต้นขึ้นบรรทัดใหม่ของสตริง
number_format	string number_format (float number [, int decimals [, string dec_point, string thousands_sep]])	จัดรูปแบบตัวเลขให้เป็นกลุ่มโดยแยกหลักพัน
ord	int ord (string string)	หาค่าแอสกีของตัวอักษร
print	int print (string arg)	แสดงสตริงออกที่จอภาพ
printf	int printf (string format [, mixed args [, mixed ...]])	แสดงสตริงที่มีการจัดรูปแบบออกที่จอภาพ
rtrim	string rtrim (string str [, string charlist])	ตัดช่องว่างหรือตัวอักษรอื่นที่ระบุออกจากจุดสิ้นสุดของสตริง
similar_text	int similar_text (string first, string second [, float &percent])	คำนวณหาความเหมือนของสตริงทั้งสอง
sprintf	string sprintf (string format [, mixed args [, mixed ...]])	ส่งค่ากลับเป็นสตริงที่มีการจัดรูปแบบ
sscanf	mixed sscanf (string str, string format [, mixed &...])	กระจายข้อมูลเข้าจากสตริงที่มีการจัดรูปแบบ
str_ireplace	mixed str_ireplace (mixed search, mixed replace, mixed subject [, int &count])	ค้นหาสตริงแล้วเปลี่ยนเป็นสตริงที่ระบุ โดยไม่สนใจตัวเล็ก-ใหญ่
str_pad	string str_pad (string input, int pad_length [, string pad_string [, int pad_type]])	ขยายสตริงตามจำนวนที่ระบุ
str_repeat	string str_repeat (string input, int multiplier)	ทำซ้ำสตริงนั้นตามจำนวนครั้งที่ระบุ
str_replace	mixed str_replace (mixed search, mixed replace, mixed subject [, int &count])	ค้นหาสตริงแล้วเปลี่ยนเป็นสตริงที่ระบุ
str_shuffle	string str_shuffle (string str)	สลับตำแหน่งสตริง
str_split	array str_split (string string [, int split_length])	แปลงสตริงให้เป็นอะเรย์
str_word_count	mixed str_word_count (string string [, int format [, string charlist]])	นับจำนวนคำในสตริงตามรูปแบบที่ระบุ
strcasecmp	int strcasecmp (string str1, string str2)	เปรียบเทียบสตริง 2 สตริง โดยไม่สนใจตัวเล็ก-ใหญ่
strchr	string strchr (string haystack, string needle)	ส่งค่ากลับเป็นสตริงที่เริ่มต้นด้วยสตริงที่ระบุ
strcmp	int strcmp (string str1, string str2)	เปรียบเทียบสตริง 2 สตริง
strip_tags	string strip_tags (string str [, string allowable_tags])	ตัดคำที่เป็นแท็ก HTML หรือ PHP ออกจากสตริง
stripcslashes	string stripcslashes (string str)	ตัดเครื่องหมาย \ (Backslash) ออกแบบภาษาซี
stripos	int stripos (string haystack, string needle [, int offset])	ค้นหาตำแหน่งแรกของสตริงที่ระบุ (needle) ในสตริงที่กำหนด (haystack) ไม่สนใจตัวเล็ก-ใหญ่
stripslashes	string stripslashes (string str)	ตัดเครื่องหมาย \ (Backslash) ออก
stristr	string stristr (string haystack, string needle)	ส่งค่ากลับเป็นสตริงที่เริ่มต้นด้วยสตริงที่ระบุ โดยไม่สนใจตัวเล็ก-ใหญ่
strlen	int strlen (string string)	หาความยาว (จำนวนตัวอักษร) ของสตริง
strnatcasecmp	int strnatcasecmp (string str1, string str2)	เปรียบเทียบสตริง 2 สตริง โดยเรียงลำดับแบบธรรมชาติ ไม่สนใจตัวเล็ก-ใหญ่
strnatcmp	int strnatcmp (string str1, string str2)	เปรียบเทียบสตริง 2 สตริง โดยเรียงลำดับแบบธรรมชาติ
strncasecmp	int strncasecmp (string str1, string str2, int len)	เปรียบเทียบสตริง 2 สตริง โดยกำหนดจำนวนตัวอักษรที่ต้องการเปรียบเทียบ ไม่สนใจตัวเล็ก-ใหญ่
strncmp	int strncmp (string str1, string str2, int len)	เปรียบเทียบสตริง 2 สตริง โดยกำหนดจำนวนตัวอักษรที่ต้องการเปรียบเทียบ
strpbrk	string strpbrk (string haystack, string char_list)	ส่งค่ากลับเป็นสตริงที่มีตัวอักษรเหมือนกับกลุ่มของ

ฟังก์ชัน	รูปแบบ	ความหมาย
		ตัวอักษรที่ระบุ
strpos	int strpos (string haystack, mixed needle [, int offset])	ค้นหาตำแหน่งแรกของสตริงที่ระบุ (needle) ในสตริงที่กำหนด (haystack)
strrchr	string strrchr (string haystack, string needle)	ค้นหาตัวอักษรตัวสุดท้ายในสตริงที่กำหนด (haystack)
strrev	string strrev (string string)	ทำการเรียงลำดับข้อมูลในสตริงแบบย้อนกลับ
stripos	int stripos (string haystack, string needle [, int offset])	ค้นหาตำแหน่งสุดท้ายของตัวอักษรที่ระบุ (needle) ในสตริงที่กำหนด (haystack) ไม่สนใจตัวเล็ก-ใหญ่
strrpos	int strrpos (string haystack, string needle [, int offset])	ค้นหาตำแหน่งสุดท้ายของตัวอักษรที่ระบุ (needle) ในสตริงที่กำหนด (haystack)
strstr	string strstr (string haystack, string needle)	ส่งค่ากลับเป็นสตริงที่เริ่มต้นด้วยสตริงที่ระบุ
strtok	string strtok (string str, string token)	แยกสตริงออกเป็นสตริงย่อยเมื่อพบตัวอักษรใด ๆ ที่ระบุ (token)
strtolower	string strtolower (string str)	เปลี่ยนตัวอักษรทั้งหมดในสตริงให้เป็นตัวเล็ก
strtoupper	string strtoupper (string string)	เปลี่ยนตัวอักษรทั้งหมดในสตริงให้เป็นตัวใหญ่
substr_compare	int substr_compare (string main_str, string str, int offset [, int length [, bool case_insensitivity]])	เปรียบเทียบส่วนของสตริง 2 สตริง โดยสามารถระบุจุดเริ่มต้นและจำนวนตัวอักษร
substr_count	int substr_count (string haystack, string needle [, int offset [, int length]])	นับจำนวนส่วนของสตริงที่พบ
substr_replace	mixed substr_replace (mixed string, string replacement, int start [, int length])	เปลี่ยนข้อความตัวอักษรในสตริงด้วยส่วนของสตริงที่ระบุ
substr	string substr (string string, int start [, int length])	ส่งค่ากลับเป็นสตริงใหม่ที่เริ่มจากจุดเริ่มต้นที่ระบุและจำนวนความยาวของสตริง
trim	string trim (string str [, string charlist])	ตัดช่องว่างหรือตัวอักษรอื่นที่ระบุออกจากจุดเริ่มต้นและสิ้นสุดของสตริง
ucfirst	string ucfirst (string str)	เปลี่ยนตัวอักษรตัวแรกของสตริงให้เป็นตัวใหญ่
ucwords	string ucwords (string str)	เปลี่ยนตัวอักษรตัวแรกของคำแต่ละคำในสตริงให้เป็นตัวใหญ่
fprintf	int fprintf (resource handle, string format, array args)	เขียนสตริงที่มีการจัดรูปแบบไปที่สตรีมที่ระบุ โดยรับอะเรย์ของอาร์กิวเมนต์
vprintf	int vprintf (string format, array args)	แสดงสตริงที่มีการจัดรูปแบบออกที่จอภาพ โดยรับอะเรย์ของอาร์กิวเมนต์
vsprintf	string vsprintf (string format, array args)	ส่งค่ากลับเป็นสตริงที่มีการจัดรูปแบบ โดยรับอะเรย์ของอาร์กิวเมนต์
wordwrap	string wordwrap (string str [, int width [, string break [, bool cut]])	ตัดคำในสตริงตามจำนวนตัวเลขที่ระบุ

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับสตริง

```

1. <?php
2.     $data1 = "SUT 2007";
3.     foreach(count_chars($data1, 1) as $i => $val){
4.         echo "There were $val instance(s) of \" ".chr($i)." \" in the string.<br>";
5.     }
6.     $data2 = "This is a sample of using string functions.";
7.     echo "$data2<br>";
8.     $data2_out1 = ucwords($data2);
9.     echo "$data2_out1<br>";
10.    $data2_out2 = str_replace("a sample of","the",$data2);
11.    echo "$data2_out2<br>";
12.    $ans = strcmp("somsak","sombat");
13.    echo "The comparison between \"somsak\" and \"sombat\" is $ans.<br>";
14.    ?>

```

```

There were 1 instance(s) of " " in the string.
There were 2 instance(s) of "0" in the string.
There were 1 instance(s) of "2" in the string.
There were 1 instance(s) of "7" in the string.
There were 1 instance(s) of "S" in the string.
There were 1 instance(s) of "T" in the string.
There were 1 instance(s) of "U" in the string.
This is a sample of using string functions.
This Is A Sample Of Using String Functions.
This is the using string functions.
The comparison between "somsak" and "sombat" is 1.
    
```

รูปที่ 5.6: การใช้ฟังก์ชันเกี่ยวกับสตริง

บรรทัดที่ 2 เป็นการกำหนดค่าที่เป็นสตริง "SUT 2007" ให้กับตัวแปร \$data1

บรรทัดที่ 3 – 5 เป็นการเรียกใช้ฟังก์ชัน foreach() ซึ่งมีการส่งอาร์กิวเมนต์ที่เป็นฟังก์ชันประกอบไปด้วย count_chars(\$data1, 1) as \$i => \$val เป็นฟังก์ชันที่ใช้นับจำนวนตัวอักษรที่มีอยู่ในตัวแปร \$data1 โดยเอาตัวอักษรแต่ละตัวนั้นไปเก็บไว้ที่ตัวแปร \$i แล้วเอาจำนวนที่นับได้เก็บไว้ที่ตัวแปร \$val

บรรทัดที่ 6 เป็นการกำหนดค่าที่เป็นสตริง "This is a sample of using string functions." ให้กับตัวแปร \$data2

บรรทัดที่ 7 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร \$data2 ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 8 เป็นการกำหนดค่าให้กับตัวแปร \$data2_out1 โดยผ่านฟังก์ชัน ucwords() ซึ่งส่งอาร์กิวเมนต์คือตัวแปร \$data2 นั่นคือเปลี่ยนตัวอักษรตัวแรกของคำแต่ละคำที่อยู่ในตัวแปร \$data2 ให้เป็นตัวใหญ่แล้วเอาไปเก็บไว้ที่ตัวแปร \$data2_out1

บรรทัดที่ 9 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร \$data2_out1 ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 10 เป็นการกำหนดค่าให้กับตัวแปร \$data2_out2 โดยผ่านฟังก์ชัน str_replace() ซึ่งส่งอาร์กิวเมนต์คือ ("a sample of", "the", \$data2) นั่นคือการค้นหาสตริง "a sample of" ค่าที่อยู่ในตัวแปร \$data2 แล้วเปลี่ยนให้เป็น "the" แทนแล้วเอาไปเก็บไว้ที่ตัวแปร \$data2_out2

บรรทัดที่ 11 เป็นการแสดงข้อมูลที่อยู่ในตัวแปร \$data2_out2 ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 12 เป็นการกำหนดค่าให้กับตัวแปร \$ans โดยผ่านฟังก์ชัน strcmp() ซึ่งส่งอาร์กิวเมนต์เป็นสตริง 2 สตริงคือ "somsak" และ "sombat" นั่นคือการเปรียบเทียบสตริงทั้ง 2 สตริง ผลที่ได้คือ 1 (เนื่องจากสตริง "somsak" มีค่ามากกว่าสตริง "sombat")

บรรทัดที่ 13 เป็นการแสดงข้อมูลสตริง "The comparison between "somsak" and "sombat" is " ตามด้วยข้อมูลที่อยู่ในตัวแปร \$ans นั่นคือ 1 แล้วตามด้วยเครื่องหมาย . (Period) ออกที่จอภาพ

5.1.6 ฟังก์ชันเกี่ยวกับคณิตศาสตร์ (Mathematical Functions)

ตารางที่ 5.6: ฟังก์ชันเกี่ยวกับคณิตศาสตร์

ฟังก์ชัน	รูปแบบ	ความหมาย
abs	number abs (mixed number)	คำนวณหาค่าสัมบูรณ์
acos	float acos (float arg)	คำนวณหาค่า arc cosine
asin	float asin (float arg)	คำนวณหาค่า arc sine
atan	float atan (float arg)	คำนวณหาค่า arc tangent
bindec	number bindec (string binary_string)	แปลงเลขฐานสองให้เป็นเลขฐานสิบ
ceil	float ceil (float value)	คำนวณหาค่าสูงกว่าและใกล้ที่สุดที่เป็นเลขจำนวนเต็ม
cos	float cos (float arg)	คำนวณหาค่า cosine
decbin	string decbin (int number)	แปลงเลขฐานสิบให้เป็นเลขฐานสอง
dechex	string dechex (int number)	แปลงเลขฐานสิบให้เป็นเลขฐานสิบหก
decoct	string decoct (int number)	แปลงเลขฐานสิบให้เป็นเลขฐานแปด
deg2rad	float deg2rad (float number)	แปลงองศาให้เป็นเรเดียน
exp	float exp (float arg)	คำนวณหาเอ็กโพเนนทของค่าที่ระบุ
floor	float floor (float value)	คำนวณหาค่าต่ำกว่าและใกล้ที่สุดที่เป็นเลขจำนวนเต็ม
fmod	float fmod (float x, float y)	คำนวณหาค่าเศษที่เหลือจากการหารของเลขจำนวนจริง
getrandmax	int getrandmax (void)	แสดงค่าที่มากที่สุดที่เป็นไปได้ในการสุ่ม
hexdec	number hexdec (string hex_string)	แปลงเลขฐานสิบหกให้เป็นเลขฐานสิบ
is_finite	bool is_finite (float val)	ทดสอบค่าว่าเป็นเลขที่มีขอบเขตจำกัด?
is_infinite	bool is_infinite (float val)	ทดสอบค่าว่าเป็นเลขที่ไม่มีขอบเขตจำกัด?
is_nan	bool is_nan (float val)	ทดสอบค่าว่าไม่เป็นตัวเลข?
log10	float log10 (float arg)	คำนวณหาค่า logarithm ฐานสิบ
log	float log (float arg [, float base])	คำนวณหาค่า logarithm ฐานธรรมชาติ
max	mixed max (number arg1, number arg2 [, number ...]) mixed max (array numbers)	หาค่าที่มากที่สุด
min	mixed min (number arg1, number arg2 [, number ...]) mixed min (array numbers)	หาค่าน้อยที่สุด
mt_getrandmax	int mt_getrandmax (void)	แสดงค่าที่มากที่สุดที่เป็นไปได้ในการสุ่ม
mt_rand	int mt_rand ([int min, int max])	คำนวณหาเลขสุ่มแบบพิเศษที่ดีกว่า rand()
octdec	number octdec (string octal_string)	แปลงเลขฐานแปดให้เป็นเลขฐานสิบ
pi	float pi (void)	คำนวณหาค่า pi
pow	number pow (number base, number exp)	คำนวณหาค่าเลขยกกำลัง
rad2deg	float rad2deg (float number)	แปลงเรเดียนให้เป็นองศา
rand	int rand ([int min, int max])	คำนวณหาเลขสุ่ม
round	float round (float val [, int precision])	คำนวณหาค่าต่ำกว่าหรือมากกว่าที่ใกล้ที่สุดที่เป็นเลขจำนวนเต็มของเลขจำนวนจริงที่ระบุ
sin	float sin (float arg)	คำนวณหาค่า sine
sqrt	float sqrt (float arg)	คำนวณหาค่ารากที่สองของเลขที่ระบุ
tan	float tan (float arg)	คำนวณหาค่า tangent

ตัวอย่าง การใช้ฟังก์ชันเกี่ยวกับคณิตศาสตร์

```

1.  <?php
2.      echo 'abs(-25) = ' .abs(-25) .'\<br>';
3.      echo 'max(8,9,3,4,1,2,7) = ' .max(8,9,3,4,1,2,7) .'\<br>';
4.      echo 'min(8,9,3,4,1,2,7) = ' .min(8,9,3,4,1,2,7) .'\<br>';
5.      echo 'decoct(9) = ' .decoct(9) .'\<br>';
6.      echo 'decbin(9) = ' .decbin(9) .'\<br>';
7.      echo 'bindec(1001) = ' .bindec(1001) .'\<br>';
8.      echo 'ceil(3.25) = ' .ceil(3.25) .'\<br>';
9.      echo 'floor(5.95) = ' .floor(5.95) .'\<br>';
10.     echo 'sqrt(9) = ' .sqrt(9) .'\<br>';
11.     echo 'deg2rad(45) = ' .deg2rad(45) .'\<br>';
12.     echo 'pow(2,4) = ' .pow(2,4) .'\<br>';
13.     echo 'rand(1, 30) = ' .rand(1,30) .'\<br>';
14.  ?>
    
```

```

abs(-25) = 25
max(8,9,3,4,1,2,7) = 9
min(8,9,3,4,1,2,7) = 1
decoct(9) = 11
decbin(9) = 1001
bindec(1001) = 9
ceil(3.25) = 4
floor(5.95) = 5
sqrt(9) = 3
deg2rad(45) = 0.785398163397
pow(2,4) = 16
rand(1, 30) = 5
    
```

รูปที่ 5.7: การใช้ฟังก์ชันเกี่ยวกับคณิตศาสตร์

บรรทัดที่ 2 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน abs() โดยส่งอาร์กิวเมนต์คือ -25 ซึ่งเป็นการหาค่าสัมบูรณ์ ผลที่ได้คือ 25 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 3 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน max() โดยส่งอาร์กิวเมนต์คือ 8, 9, 3, 4, 1, 2 และ 7 ซึ่งเป็นการหาค่าที่มากที่สุด ผลที่ได้คือ 9 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 4 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน min() โดยส่งอาร์กิวเมนต์คือ 8, 9, 3, 4, 1, 2 และ 7 ซึ่งเป็นการหาค่าที่น้อยที่สุด ผลที่ได้คือ 1 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 5 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน decoct() โดยส่งอาร์กิวเมนต์คือ 9 ซึ่งเป็นการแปลงค่าของเลขฐานสิบเป็นเลขฐานแปด ผลที่ได้คือ 11 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 6 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน decbin() โดยส่งอาร์กิวเมนต์คือ 9 ซึ่งเป็นการแปลงค่าของเลขฐานสิบเป็นเลขฐานสอง ผลที่ได้คือ 1001 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 7 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน bindec () โดยส่งอาร์กิวเมนต์คือ 1001 ซึ่งเป็นการแปลงค่าของเลขฐานสองเป็นเลขฐานสิบผลที่ได้คือ 9 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 8 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน ceil() โดยส่งอาร์กิวเมนต์คือ 3.25 ซึ่งเป็นการหาค่าสูงกว่าและใกล้ที่สุดที่เป็นเลขจำนวนเต็ม ผลที่ได้คือ 4 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 9 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน floor() โดยส่งอาร์กิวเมนต์คือ 5.95 ซึ่งเป็นการหาค่าต่ำกว่าและใกล้ที่สุดที่เป็นเลขจำนวนเต็ม ผลที่ได้คือ 5 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 10 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน sqrt() โดยส่งอาร์กิวเมนต์คือ 9 ซึ่งเป็นการหาค่ารากที่สอง ผลที่ได้คือ 3 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 11 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน deg2rad() โดยส่งอาร์กิวเมนต์คือ 45 ซึ่งเป็นการแปลงองศาให้เป็นเรเดียน ผลที่ได้คือ 0.785398163397 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 12 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน pow() โดยส่งอาร์กิวเมนต์คือ 2, 4 ซึ่งเป็นการหาค่าเลขยกกำลัง ผลที่ได้คือ 16 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 13 เป็นการแสดงผลของการเรียกใช้ฟังก์ชัน rand() โดยส่งอาร์กิวเมนต์คือ 1, 30 ซึ่งเป็นการหาเลขสุ่มจาก 1 ถึง 30 ผลที่ได้คือ 5 พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

5.1.7 ฟังก์ชันเกี่ยวกับเมล (Mail Functions)

ตารางที่ 5.7: ฟังก์ชันเกี่ยวกับเมล

ฟังก์ชัน	รูปแบบ	ความหมาย
mail	bool mail (string to, string subject, string message [, string additional_headers [, string additional_parameters]])	การส่งเมล

พารามิเตอร์ที่สำคัญในฟังก์ชันเมลมีดังนี้คือ

to คือผู้รับจดหมาย เช่น somchai@yahoo.com

subject คือหัวเรื่องของจดหมาย เช่น Happy Birthday

message คือข้อความที่ต้องการส่ง เช่น Happy Birthday To You... Mr.Somchai

headers คือส่วนหัวที่เป็นทางเลือกที่จะใส่หรือไม่ใส่ก็ได้ เช่น from, reply-to, cc และ

bcc เป็นต้น

ตัวอย่าง การสร้างแบบฟอร์มในการส่งจดหมาย

```
<html>
<head>
  <title>An Example of Sending Mail</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
</head>
<center>
<body>
  <table width="200px"><tr><td>
    <form method="post" action="mail.php">
      <fieldset style="border-style: ridge; border-color: red"><legend><i><b>An Example of
Sending Mail</b></i></legend>
      <table>
        <tr>
          <td colspan="2" align="center"><hr></td></tr>
        <tr>
          <td align="right"><b>To: </b></td>
          <td><input type="text" name="to" size="40"></td></tr>
        <tr>
          <td align="right"><b>Subject: </b></td>
          <td><input type="text" name="subject" size="40"></td></tr>
        <tr>
          <td align="right"><b>Message: </b></td>
          <td><textarea name="msg" rows="5" cols="30"></textarea></td></tr>
        <tr>
          <td align="right"><b>From: </b></td>
          <td><input type="text" name="from" size="40"></td></tr>
        <tr>
          <td colspan="2" align="center"><hr><input type="submit" value="Submit"
style="width:80px">&nbsp;&nbsp;&nbsp;<input type="reset" value="Reset" style="width:80px"></td></tr>
      </table></fieldset></form>
    </td></tr></table>
</center>
</body>
</html>
```



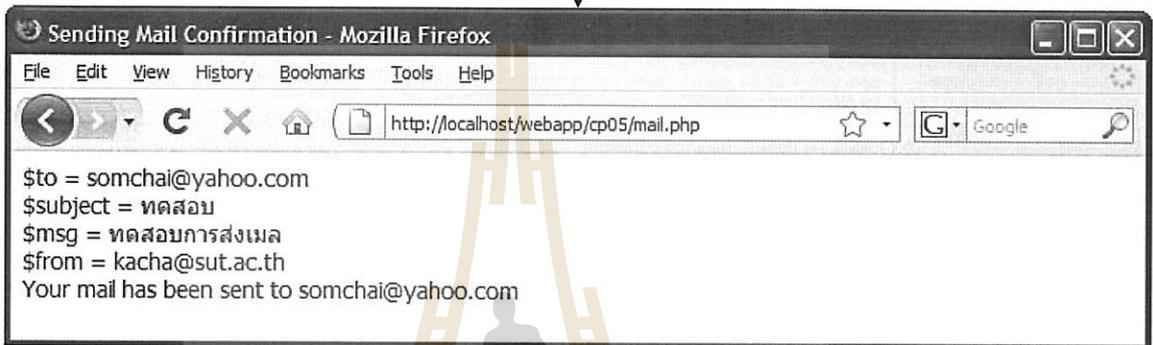
รูปที่ 5.8: แบบฟอร์มการส่งเมล

เมื่อผู้ใช้ป้อนข้อมูลทั้งหมดแล้วกดปุ่ม "Submit" ตัวแปร \$to, \$subject, \$msg และ \$from ที่เก็บข้อมูลเหล่านั้นก็จะถูกส่งไปที่ไฟล์ mail.php ซึ่งรูปแบบของการเขียนคำสั่งในไฟล์ mail.php มีรายละเอียดดังภาพที่ 5.9 (สมมุติว่าในไฟล์ php.ini นั้น กำหนดค่า register_globals = On)

```

1. <html>
2. <head>
3.   <title>Sending Mail Confirmation</title>
4.   <meta http-equiv="content-type" content="text/html; charset=utf-8">
5. </head>
6. <body>
7. <?php
8.   echo '$to = '.$to.'<br>';
9.   echo '$subject = '.$subject.'<br>';
10.  echo '$msg = '.$msg.'<br>';
11.  echo '$from = '.$from.'<br>';
12.  $headers = "From: ".$from;
13.  if(mail($to, $subject, $msg, $headers))
14.    echo "Your mail has been sent to $to<br>";
15.  else
16.    echo "Sorry, Please try again, later!!!<br>";
17.  ?>
18. </body>
19. </html>

```



รูปที่ 5.9: การส่งค่าตัวแปรต่าง ๆ ในแบบฟอร์ม

บรรทัดที่ 8 – 11 เป็นเพียงการทดสอบเพื่อแสดงข้อมูลของตัวแปร \$to, \$subject, \$msg และ \$from ที่ถูกส่งมาจากไฟล์ “mail_form.html” ซึ่งผลที่แสดงออกก็เป็นข้อมูลที่ถูกต้อง

บรรทัดที่ 12 เป็นการรวมสตริง “From: ” กับข้อมูลที่อยู่ในตัวแปร \$from เข้าด้วยกันแล้วเก็บไว้ที่ตัวแปร \$headers

บรรทัดที่ 13 เป็นการเรียกใช้ฟังก์ชัน mail() พร้อมกับทดสอบว่าสามารถส่งเมลได้หรือไม่? ถ้าส่งได้ให้แสดงข้อความว่า “Your mail has been sent to somchai@yahoo.com” มิฉะนั้นแล้วให้แสดงข้อความ “Sorry, Please try again, later!!!” แล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

5.1.8 ฟังก์ชันเกี่ยวกับระบบไฟล์ข้อมูล (Filesystem Functions)

ตารางที่ 5.8: ฟังก์ชันที่เกี่ยวกับระบบไฟล์ข้อมูล

ฟังก์ชัน	รูปแบบ	ความหมาย
basename	string basename (string path [, string suffix])	ส่งค่ากลับเป็นชื่อไฟล์ที่เป็นส่วนหนึ่งของเส้นทาง
chgrp	bool chgrp (string filename, mixed group)	เปลี่ยนกลุ่มของไฟล์
chmod	bool chmod (string filename, int mode)	เปลี่ยนโหมดของไฟล์
chown	bool chown (string filename, mixed user)	เปลี่ยนเจ้าของของไฟล์
clearstatcache	void clearstatcache (void)	ลบข้อมูลรายละเอียดของไฟล์ในแคช
copy	bool copy (string source, string dest)	ทำสำเนาไฟล์
delete	void delete (string file)	ลบไฟล์
dirname	string dirname (string path)	แสดงชื่อไดเรกทอรีของเส้นทาง
disk_free_space	float disk_free_space (string directory)	คำนวณหาเนื้อที่ว่างในไดเรกทอรี

ฟังก์ชัน	รูปแบบ	ความหมาย
disk_total_space	float disk_total_space (string directory)	คำนวณหาพื้นที่ของไดเรกทอรี
diskfreespace	float diskfreespace (string directory)	คำนวณหาเนื้อที่ว่างในไดเรกทอรี
fclose	bool fclose (resource handle)	ปิดไฟล์ที่พอยน์เตอร์ชี้อยู่
feof	bool feof (resource handle)	ทดสอบจุดสิ้นสุดของไฟล์
fflush	bool fflush (resource handle)	ส่งผลลัพธ์ไปที่ไฟล์
fgetc	string fgetc (resource handle)	รับตัวอักษรหนึ่งตัวจากไฟล์ที่พอยน์เตอร์ชี้อยู่
fgetcsv	array fgetcsv (resource handle [, int length [, string delimiter [, string enclosure]])	รับแถวจากไฟล์ที่พอยน์เตอร์ชี้อยู่แล้วส่งกลับเป็นอะเรย์สำหรับ CSV ฟิลด์
fgets	string fgets (resource handle [, int length])	รับตัวอักษรหนึ่งแถวจากไฟล์ที่พอยน์เตอร์ชี้อยู่
fgetss	string fgetss (resource handle [, int length [, string allowable_tags]])	รับตัวอักษรหนึ่งแถวจากไฟล์ที่พอยน์เตอร์ชี้อยู่ ยกเว้นแท็ก HTML
file_exists	bool file_exists (string filename)	ทดสอบว่ามีไฟล์หรือไดเรกทอรีที่ระบุ?
file_get_contents	string file_get_contents (string filename [, bool use_include_path [, resource context [, int offset [, int maxlen]])	อ่านข้อมูลทั้งหมดในไฟล์แล้วเก็บใส่สตริง 1 สตริง
file_put_contents	int file_put_contents (string filename, mixed data [, int flags [, resource context]])	เขียนสตริงลงไฟล์
file	array file (string filename [, int use_include_path [, resource context]])	อ่านข้อมูลทั้งหมดในไฟล์แล้วเก็บใส่สตริง 1 สตริง
fileatime	int fileatime (string filename)	แสดงเวลาครั้งสุดท้ายที่มีการจัดการกับไฟล์
filectime	int filectime (string filename)	แสดงเวลาที่ไฟล์ถูกเปลี่ยนแปลงครั้งล่าสุด
filegroup	int filegroup (string filename)	แสดงกลุ่มของไฟล์
fileinode	int fileinode (string filename)	แสดง inode ของไฟล์
filemtime	int filemtime (string filename)	แสดงเวลาครั้งสุดท้ายที่มีการเปลี่ยนแปลงไฟล์
fileowner	int fileowner (string filename)	แสดงเจ้าของไฟล์
fileperms	int fileperms (string filename)	แสดงการกำหนดสิทธิ์ของไฟล์
filesize	int filesize (string filename)	แสดงขนาดของไฟล์
filetype	string filetype (string filename)	แสดงชนิดของไฟล์
flock	bool flock (resource handle, int operation [, int &wouldblock])	กำหนดรูปแบบในการล็อกไฟล์
fopen	resource fopen (string filename, string mode [, bool use_include_path [, resource zcontext]])	เปิดไฟล์ที่พอยน์เตอร์ชี้อยู่
fpassthru	int fpassthru (resource handle)	แสดงข้อมูลที่เหลือทั้งหมดเริ่มต้นที่ไฟล์พอยน์เตอร์ชี้อยู่
fputcsv	int fputcsv (resource handle [, array fields [, string delimiter [, string enclosure]])	จัดรูปแบบแถวให้เป็นแบบ CSV แล้วเขียนไปที่ไฟล์พอยน์เตอร์ชี้อยู่
fputs	int fputs (resource handle, string string [, int length])	เขียนไฟล์ที่พอยน์เตอร์ชี้อยู่
fread	string fread (resource handle, int length)	อ่านไฟล์ที่พอยน์เตอร์ชี้อยู่
fscanf	mixed fscanf (resource handle, string format [, mixed &...])	อ่านข้อมูลแบบมีรูปแบบจากไฟล์
fseek	int fseek (resource handle, int offset [, int whence])	เลื่อนพอยน์เตอร์ไปยังตำแหน่งที่ต้องการ
fstat	array fstat (resource handle)	แสดงรายละเอียดของไฟล์ที่ใช้พอยน์เตอร์ในการเปิดไฟล์
ftell	int ftell (resource handle)	แสดงตำแหน่งในการอ่าน/เขียนของไฟล์
ftruncate	bool ftruncate (resource handle, int size)	ตัดไฟล์บางส่วนทิ้งตามจำนวนที่ระบุ
fwrite	int fwrite (resource handle, string string [, int length])	เขียนไฟล์ที่พอยน์เตอร์ชี้อยู่
is_dir	bool is_dir (string filename)	ทดสอบว่าชื่อไฟล์นั้นเป็นไดเรกทอรี?
is_executable	bool is_executable (string filename)	ทดสอบว่าไฟล์นั้นสามารถกระทำการได้ (execute)?
is_file	bool is_file (string filename)	ทดสอบว่าชื่อไฟล์นั้นเป็นไฟล์ปกติ?
is_link	bool is_link (string filename)	ทดสอบว่าชื่อไฟล์นั้นเป็นจุดเชื่อมต่อ?
is_readable	bool is_readable (string filename)	ทดสอบว่าชื่อไฟล์นั้นสามารถอ่านได้?
is_uploaded_file	bool is_uploaded_file (string filename)	ทดสอบว่าไฟล์นั้นได้อัพโหลดผ่านทาง HTTP POST?

- บรรทัดที่ 2** เป็นการกำหนดข้อมูลสตริง “D:” ให้กับตัวแปร \$dir
- บรรทัดที่ 3** เป็นการแสดงข้อมูลที่อยู่ในตัวแปร \$dir ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่
- บรรทัดที่ 4** เป็นการกำหนดข้อมูลเลขจำนวนเต็ม 0 ให้กับตัวแปร \$c_file เพื่อใช้ในการเก็บค่าจำนวนไฟล์
- บรรทัดที่ 5** เป็นการกำหนดข้อมูลเลขจำนวนเต็ม 0 ให้กับตัวแปร \$c_dir เพื่อใช้ในการเก็บค่าจำนวนไดเรคทอรี
- บรรทัดที่ 6** เป็นการแสดงข้อมูลของเนื้อที่ทั้งหมดของไดรว์ D: ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่
- บรรทัดที่ 7** เป็นการแสดงข้อมูลของเนื้อที่ที่ว่างของไดรว์ D: ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่
- บรรทัดที่ 8** เป็นการแสดงสตริง “The following information are files and directory in drive ” ตามด้วยค่าที่อยู่ในตัวแปร \$dir นั่นคือ D: ออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่
- บรรทัดที่ 9** เป็นการทดสอบการเปิดไดเรคทอรีโดยเรียกใช้ฟังก์ชัน opendir() ซึ่งส่งอาร์กิวเมนต์เป็นตัวแปร \$dir แล้วให้เก็บผลไว้ที่ตัวแปร \$dh
- บรรทัดที่ 10 – 17** เป็นการใช้ลูป while โดยเรียกใช้ฟังก์ชัน readdir() ในการอ่านข้อมูลทั้งหมดที่อยู่ในตัวแปร \$dh ทีละข้อมูล แล้วเก็บไว้ที่ตัวแปร \$name
- บรรทัดที่ 11** เป็นการกำหนดค่าให้กับตัวแปร \$type โดยผ่านฟังก์ชัน filetype() ซึ่งส่งอาร์กิวเมนต์เป็นตัวแปร \$dir.\$name นั่นคือเส้นทางของไฟล์
- บรรทัดที่ 12** เป็นการแสดงชื่อและชนิดของข้อมูลออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่
- บรรทัดที่ 13 – 17** เป็นการทดสอบชนิดข้อมูลว่า ถ้าเป็นไดเรคทอรี จะทำการเพิ่มค่าอีก 1 ให้กับตัวแปร \$c_dir มิฉะนั้นแล้วจะทำการเพิ่มค่าอีก 1 ให้กับตัวแปร \$c_file แทน
- บรรทัดที่ 18** เป็นการเรียกใช้ฟังก์ชัน closedir() ซึ่งส่งอาร์กิวเมนต์เป็นตัวแปร \$dh นั่นคือการปิดไดเรคทอรี ที่มีการเปิดโดยการใช้ฟังก์ชัน opendir() กับอาร์กิวเมนต์เป็นตัวแปร \$dh
- บรรทัดที่ 20** เป็นการแสดงผลของการนับจำนวนไฟล์และไดเรคทอรีออกที่จอภาพ พร้อมกับเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

5.2 ฟังก์ชันที่สร้างขึ้นเอง (User-defined Functions)

ฟังก์ชันที่ผู้พัฒนาโปรแกรมสามารถสร้างขึ้นใช้เองนั้น มีรูปแบบดังนี้

5.2.1 ฟังก์ชันอาร์กิวเมนต์ (Function Arguments) คือรายละเอียดต่าง ๆ ที่ส่งถึงฟังก์ชัน ในรูปแบบของตัวแปรหรือค่าคงที่มี 3 ลักษณะคือ

- การส่งแบบการผ่านค่า (Pass by value)

การส่งแบบการผ่านค่ามีรูปแบบดังนี้

```
function function_name(argument) {
    statement;
}

function_name(variable);
```

ตัวอย่าง การส่งแบบการผ่านค่า

<pre>1. <?php 2. function summation(\$num1,\$num2){ 3. \$sum = \$num1 + \$num2; 4. return \$sum; 5. } 6. \$n1 = 5; 7. \$n2 = 20; 8. echo "\$n1 + \$n2 = ".summation(\$n1,\$n2); 9. ?></pre>	
---	--

รูปที่ 5.11: การส่งแบบการผ่านค่า

บรรทัดที่ 2 – 5 เป็นการกำหนดรูปแบบการทำงานของฟังก์ชัน summation() ซึ่งมีรายละเอียดดังนี้

บรรทัดที่ 2 function คือ คำสำคัญที่บ่งบอกถึงจุดเริ่มต้นของฟังก์ชัน summation คือ ชื่อฟังก์ชัน \$num1 และ \$num2 เป็นอาร์กิวเมนต์อยู่ในรูปของตัวแปรที่จะรับค่าจากการเรียกใช้ฟังก์ชันนี้

บรรทัดที่ 3 \$sum เป็นตัวแปรที่เก็บค่าผลบวกระหว่าง \$num1 กับ \$num2

บรรทัดที่ 4 return \$sum; คำว่า return เป็นคำสำคัญที่บ่งบอกถึงการส่งค่ากลับไปจุดเรียกใช้ ซึ่งตามด้วยตัวแปร \$sum นั่นคือการส่งค่าที่อยู่ในตัวแปร \$sum กลับไปที่จุดเรียกใช้

บรรทัดที่ 6 เป็นการกำหนดค่า 5 ให้กับตัวแปร \$n1

บรรทัดที่ 7 เป็นการกำหนดค่า 20 ให้กับตัวแปร \$n2

บรรทัดที่ 8 เป็นการนำค่าที่อยู่ในตัวแปร \$n1 แสดงออกที่จอภาพ (ค่า 5 ถูกแสดงออก) ตามด้วยเครื่องหมาย + (Plus) ตามด้วยค่าที่อยู่ในตัวแปร \$n2 (ค่า 20 ถูกแสดงออก) ตามด้วยเครื่องหมาย = (Equal) แล้วตามด้วยการใช้คำสั่ง summation(\$n1,\$n2) ซึ่งคือการเรียกใช้ฟังก์ชัน summation() โดยส่งอาร์กิวเมนต์ไปที่ฟังก์ชันนั้นด้วยคือ ตัวแปร \$n1 และ \$n2 เมื่อไปที่ฟังก์ชัน summation() แล้วค่าของตัวแปร \$n1 จะถูกทำสำเนาแล้วเก็บไว้ที่ตัวแปร \$num1 และค่าของตัวแปร \$n2 จะถูกทำสำเนาแล้วเก็บไว้ที่ตัวแปร \$num2 ตามลำดับ ดังนั้นค่าที่ส่งกลับมาที่จุดเรียกใช้คือ 25

● **การส่งแบบการผ่านตัวอ้างอิง (Pass by reference)**

โดยปกติแล้วการส่งอาร์กิวเมนต์ไปที่ฟังก์ชันนั้นจะเป็นการส่งแบบการผ่านค่า ซึ่งนั่นคือถ้ามีการเปลี่ยนแปลงค่าในฟังก์ชัน ก็จะไม่กระทบกับค่าที่ส่งเข้ามา แต่สำหรับการส่งแบบการผ่านตัวอ้างอิงแล้ว ถ้ามีการเปลี่ยนแปลงค่าในฟังก์ชัน ก็จะกระทบกับค่าเดิมด้วย การส่งแบบการผ่านตัวอ้างอิงทำได้โดยใส่เครื่องหมาย & (Ampersand) นำหน้าตัวแปร มีรูปแบบดังนี้

```
function function_name(argument) {
    statement;
}

function_name(&variable);
```

ตัวอย่าง การส่งแบบการผ่านตัวอ้างอิง

```
1. <?php
2.     function new_value(&$num){
3.         $num = $num * $num;
4.     }
5.     $n1 = 20;
6.     echo 'Before calling the function: $n1 = ' . $n1 . '<br>';
7.     new_value($n1);
8.     echo 'After calling the function: $n1 = ' . $n1 . '<br>';
9. ?>
```

หรือ

```
1. <?php
2.     function new_value($num){
3.         $num = $num * $num;
4.     }
5.     $n1 = 20;
6.     echo 'Before calling the function: $n1 = ' . $n1 . '<br>';
7.     new_value(&$n1);
8.     echo 'After calling the function: $n1 = ' . $n1 . '<br>';
9. ?>
```

หรือ

```
1. <?php
2.     function new_value(&$num){
3.         $num = $num * $num;
4.     }
5.     $n1 = 20;
6.     echo 'Before calling the function: $n1 = ' . $n1 . '<br>';
7.     new_value(&$n1);
8.     echo 'After calling the function: $n1 = ' . $n1 . '<br>';
9. ?>
```

```
Before calling the function: $n1 = 20
After calling the function: $n1 = 400
```

รูปที่ 5.12: การส่งแบบการผ่านตัวอ้างอิง

บรรทัดที่ 2 – 4 เป็นการกำหนดรูปแบบการทำงานของฟังก์ชัน new_value() ซึ่งมีรายละเอียดดังนี้

บรรทัดที่ 2 function คือ คำสำคัญที่บ่งบอกถึงจุดเริ่มต้นของฟังก์ชัน
 new_value คือ ชื่อฟังก์ชัน

- \$num เป็นอาร์กิวเมนต์อยู่ในรูปของตัวแปรที่จะรับค่าจากการเรียกใช้ฟังก์ชันนี้
- บรรทัดที่ 3** \$num เป็นตัวแปรที่เก็บค่าผลคูณระหว่าง \$num กับ \$num
- บรรทัดที่ 5** เป็นการกำหนดค่า 20 ให้กับตัวแปร \$n1
- บรรทัดที่ 6** เป็นการแสดงค่าของตัวแปร \$n1 การเรียกใช้ฟังก์ชัน new_value() ซึ่งค่าที่แสดงออกคือ 20
- บรรทัดที่ 7** เป็นการเรียกใช้ฟังก์ชัน new_value() โดยส่งอาร์กิวเมนต์ไปที่ฟังก์ชันนั้นด้วยคือที่อยู่ของตัวแปร \$n1 เมื่อไปที่ฟังก์ชัน new_value() แล้ว ฟังก์ชัน new_value() จะเก็บที่อยู่ของตัวแปร \$n1 ไว้ที่ตัวแปร \$num นั่นคือ ถ้ามีการเปลี่ยนแปลงค่าของตัวแปร \$num ภายในฟังก์ชัน new_value() จะทำให้ค่าที่อยู่ในตัวแปร \$n1 เปลี่ยนไปด้วย
- บรรทัดที่ 8** เป็นการแสดงค่าของตัวแปร \$n1 หลังการเรียกใช้ฟังก์ชัน new_value() ซึ่งค่าที่แสดงออกคือ 400 ที่เป็นเช่นนี้เนื่องจากมีการเปลี่ยนแปลงค่าของตัวแปร \$num ในบรรทัดที่ 3

● **การส่งแบบการกำหนดค่าปริยาย (Default argument values)**

การส่งแบบการกำหนดค่าปริยายนี้ เป็นการกำหนดค่าปริยายในฟังก์ชันนั้น ๆ เลย สามารถเรียกใช้ฟังก์ชันนั้น ๆ ได้ 2 ลักษณะคือ

1. การเรียกใช้ฟังก์ชันโดยไม่ส่งอาร์กิวเมนต์ใด ๆ ไปที่ฟังก์ชัน การทำงานของฟังก์ชันจะถือเอาค่าปริยายที่มีการกำหนดในส่วนของอาร์กิวเมนต์
2. การเรียกใช้โดยส่งอาร์กิวเมนต์ไปที่ฟังก์ชัน การทำงานของฟังก์ชันจะถือเอาที่ถูกส่งมาแทนค่าปริยายที่มีการกำหนดในส่วนของอาร์กิวเมนต์

```
function function_name(argument){
    statement;
}

function_name(); // function_name(variable);
```

ตัวอย่าง การส่งแบบการกำหนดค่าปริยาย

<pre>1. <?php 2. function food(\$type = "steak"){ 3. return "I like to eat \$type.
"; 4. } 5. echo food(); 6. \$str = "noodle"; 7. echo food(\$str); 8. echo food(); 9. ?></pre>		<pre>I like to eat steak. I like to eat noodle. I like to eat steak.</pre>
---	---	--

รูปที่ 5.13: การส่งแบบการกำหนดค่าปริยาย

- บรรทัดที่ 2 – 4** เป็นการกำหนดรูปแบบการทำงานของฟังก์ชัน food() ซึ่งมีรายละเอียดดังนี้
- บรรทัดที่ 2** function คือ คำสำคัญที่บ่งบอกถึงจุดเริ่มต้นของฟังก์ชัน
- food คือ ชื่อฟังก์ชัน
- \$type = "steak" เป็นอาร์กิวเมนต์ในรูปของตัวแปรที่มีการกำหนดค่าเป็นสตริง "steak" ให้ ซึ่งคือค่าปริยาย (default) นั่นเอง

บรรทัดที่ 3 มีการส่งค่ากลับเป็นสตริงที่ประกอบไปด้วยข้อมูล “I like to eat” และข้อมูลที่อยู่ในตัวแปร \$type

บรรทัดที่ 5 เป็นการเรียกใช้ฟังก์ชัน food() โดยไม่ส่งอาร์กิวเมนต์ใด ๆ ไปที่ฟังก์ชัน ดังนั้นฟังก์ชัน food() ก็จะนำเอาค่าปริยาย (\$type = “steak”) ออกมาประมวลผล นั่นคือการส่งค่ากลับเป็นสตริง “I like to eat steak.”

บรรทัดที่ 6 เป็นการกำหนดค่าสตริง “noodle” ให้กับตัวแปร \$str

บรรทัดที่ 7 เป็นการเรียกใช้ฟังก์ชัน food() โดยส่งอาร์กิวเมนต์เป็นตัวแปร \$str ซึ่งเก็บข้อมูลเป็นสตริง “noodle” ไปที่ฟังก์ชัน ดังนั้นฟังก์ชัน food() ก็จะนำเอาสตริง “noodle” ไปเก็บไว้ที่ตัวแปร \$type แล้วส่งค่ากลับเป็นสตริง “I like to eat noodle.”

บรรทัดที่ 8 เป็นการเรียกใช้ฟังก์ชัน food() อีกครั้งโดยไม่ส่งอาร์กิวเมนต์ใด ๆ ไปที่ฟังก์ชัน ดังนั้นฟังก์ชัน food() ก็จะนำเอาค่าปริยาย (\$type = “steak”) ออกมาประมวลผล นั่นคือการส่งค่ากลับเป็นสตริง “I like to eat steak.”

5.2.2 ตัวแปรฟังก์ชัน (Variable Functions) ในภาษา PHP นี้ ถ้าตัวแปรใด ๆ มีเครื่องหมาย () (Parentheses) ต่อท้าย จะทำการค้นหาและประมวลผลฟังก์ชันที่มีชื่อเหมือนกับตัวแปรนั้น ดังตัวอย่างในรูปที่ 5.14

<pre> 1. <?php 2. function abc(){ 3. echo "This is in function abc(
"; 4. } 5. function message(\$year = "2004"){ 6. echo "Function message(): \$year
"; 7. } 8. \$func = "abc"; 9. \$func(); 10. \$func = "message"; 11. \$func(); 12. \$func(2007); 13. ?> </pre>		<pre> This is in function abc() Function message(): 2004 Function message(): 2007 </pre>
--	--	--

รูปที่ 5.14: การทำงานของตัวแปรฟังก์ชัน

บรรทัดที่ 2 – 4 เป็นการกำหนดรูปแบบการทำงานของฟังก์ชัน abc() ซึ่งมีรายละเอียดดังนี้

บรรทัดที่ 2 function คือ คำสำคัญที่บ่งบอกถึงจุดเริ่มต้นของฟังก์ชัน
abc คือ ชื่อฟังก์ชัน

บรรทัดที่ 3 เป็นการแสดงข้อความที่เป็นสตริง “This is in function abc()” ออกที่จอภาพ แล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 5 – 7 เป็นการกำหนดรูปแบบการทำงานของฟังก์ชัน message() ซึ่งมีรายละเอียดดังนี้

บรรทัดที่ 5 function คือ คำสำคัญที่บ่งบอกถึงจุดเริ่มต้นของฟังก์ชัน
message คือ ชื่อฟังก์ชัน

\$year = “2004” เป็นอาร์กิวเมนต์ในรูปของตัวแปรที่มีการกำหนดค่าเป็นสตริง “2004” ให้ ซึ่งคือค่าปริยาย (default) นั่นเอง

บรรทัดที่ 6 เป็นการแสดงข้อความที่เป็นสตริง “Function message(): ” แล้วต่อด้วยข้อมูลที่อยู่ในตัวแปร \$year ออกที่จอภาพแล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 8 เป็นการกำหนดค่าสตริง “abc” ให้กับตัวแปร \$func

บรรทัดที่ 9 เป็นการเรียกใช้ฟังก์ชัน func() โดยไม่มีการส่งอาร์กิวเมนต์ใด ๆ ไป จะสังเกตเห็นว่าไม่มีการกำหนดรูปแบบและการทำงานของฟังก์ชัน func() ในตัวโปรแกรมเลย แต่ในบรรทัดที่ 8 มีการกำหนดค่าให้กับตัวแปร \$func เป็นสตริง “abc” ดังนั้นคำสั่งในบรรทัดนี้จึงทำการค้นหาและประมวลผลฟังก์ชันที่มีชื่อเหมือนกับค่าที่อยู่ในตัวแปร \$func นั่นคือฟังก์ชัน abc() จึงมีการแสดงผลเป็นสตริง “This is in function abc()” ออกที่จอภาพแล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

บรรทัดที่ 10 เป็นการกำหนดค่าสตริง “message” ให้กับตัวแปร \$func

บรรทัดที่ 11 เป็นการเรียกใช้ฟังก์ชัน func() โดยไม่มีการส่งอาร์กิวเมนต์ใด ๆ ไป จะสังเกตเห็นว่าไม่มีการกำหนดรูปแบบและการทำงานของฟังก์ชัน func() ในตัวโปรแกรมเลย แต่ในบรรทัดที่ 10 มีการกำหนดค่าให้กับตัวแปร \$func เป็นสตริง “message” ดังนั้นคำสั่งในบรรทัดนี้จึงทำการค้นหาและประมวลผลฟังก์ชันที่มีชื่อเหมือนกับค่าที่อยู่ในตัวแปร \$func นั่นคือฟังก์ชัน message() จึงมีการแสดงผลเป็นสตริง “Function message(): 2004” ออกที่จอภาพแล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

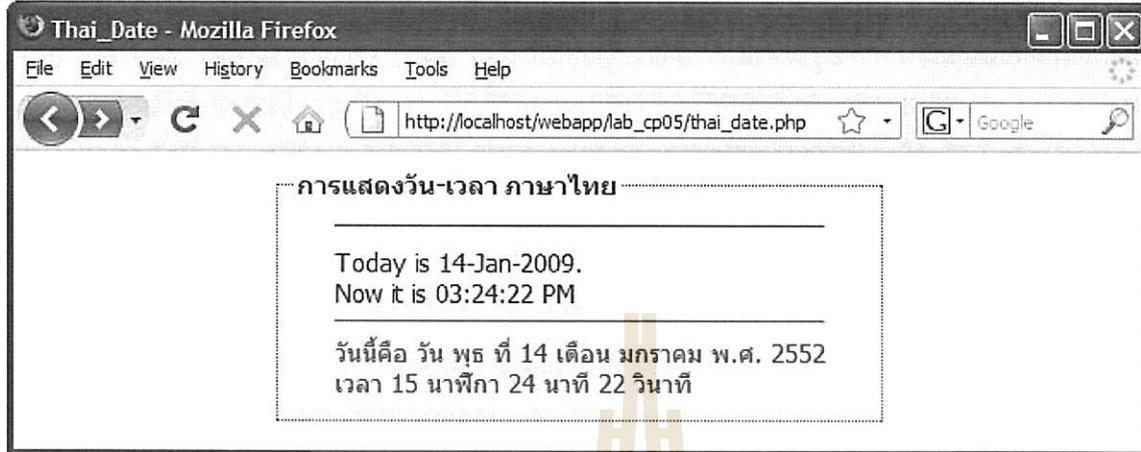
บรรทัดที่ 12 เป็นการเรียกใช้ฟังก์ชัน func() อีกครั้งโดยส่งอาร์กิวเมนต์เป็นสตริง “2007” ไป จะสังเกตเห็นว่าไม่มีการกำหนดรูปแบบและการทำงานของฟังก์ชัน func() ในตัวโปรแกรมเลย แต่ในบรรทัดที่ 10 มีการกำหนดค่าให้กับตัวแปร \$func เป็นสตริง “message” ดังนั้นคำสั่งในบรรทัดนี้จึงทำการค้นหาและประมวลผลฟังก์ชันที่มีชื่อเหมือนกับค่าที่อยู่ในตัวแปร \$func นั่นคือฟังก์ชัน message() แล้วนำสตริง “2007” ไปเก็บไว้ที่ตัวแปร \$year ดังนั้นจึงมีการแสดงผลเป็นสตริง “Function message(): 2007” ออกที่จอภาพแล้วเลื่อนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) ไปบรรทัดใหม่

สรุป

ฟังก์ชันต่าง ๆ ที่มีอยู่ในตัวโปรแกรม (Internal or Built-in Functions) นั้นต่างก็เื้อออำนวยความสะดวกให้กับนักพัฒนาโปรแกรมเป็นอย่างมาก แต่บางครั้งผู้พัฒนาโปรแกรมจำเป็นต้องสร้างฟังก์ชันขึ้นมาใช้งานเอง (User-defined Functions) ทั้งนี้เพื่อให้เหมาะสมและสอดคล้องกับความต้องการในการแก้ปัญหาต่าง ๆ โดยฟังก์ชันต่าง ๆ ที่สร้างขึ้นมานี้ สามารถเรียกใช้ได้ตามความต้องการ ซึ่งง่ายและสะดวกต่อการปรับปรุง ทดสอบ และแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นได้ ณ จุดใดจุดหนึ่งของการทำงานของโปรแกรม

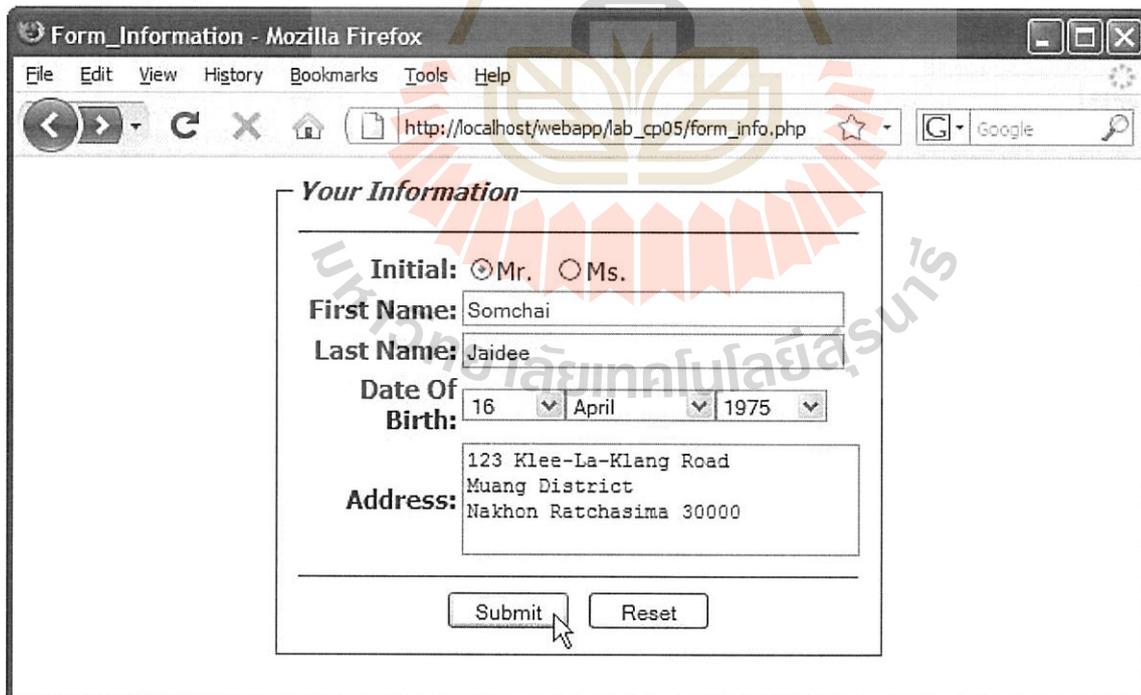
แบบฝึกหัด

1. สร้างไฟล์ thai_date.php เพื่อแสดงข้อมูล วัน เดือน และ ปี เป็นภาษาไทย ดังรูปที่ 5.15



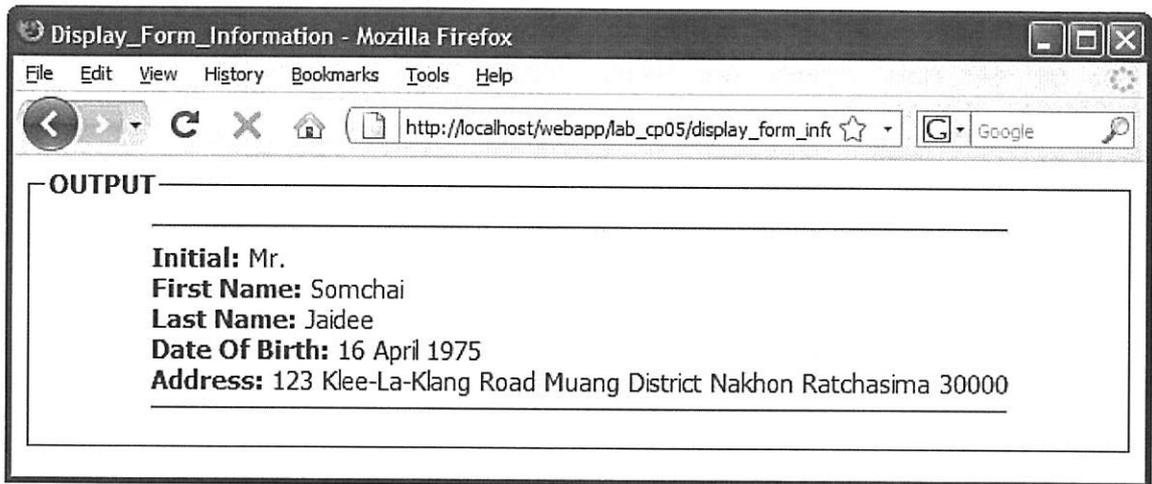
รูปที่ 5.15: การแสดงผลวัน เดือน และปี เป็นภาษาไทย

2. สร้างแบบฟอร์มในการรับข้อมูล ซึ่งประกอบไปด้วย คำนำหน้า ชื่อ สกุล วันเดือนปีเกิด และที่อยู่ แล้วบันทึกเป็น form_info.php ดังรูปที่ 5.16 โดยปุ่มวันให้ใช้คำสั่งในการวนลูปจาก วันที่ 1 – 31 ในการแสดงตัวเลือก ปุ่มเดือนให้กำหนดข้อมูลแบบอะเรย์ที่เก็บเดือนทั้ง 12 เดือน แล้วใช้ foreach ในการแสดงตัวเลือก ปุ่มปีให้ใช้คำสั่งในการวนลูปจาก ปี 1937-1995 ในการแสดงตัวเลือก



รูปที่ 5.16: แบบฟอร์มของไฟล์ form_info.php

เมื่อผู้ใช้กดปุ่ม Submit ก็จะมีการส่งข้อมูลทั้งหมดไปที่ไฟล์ display_form_info.php เพื่อแสดงผลดังรูปที่ 5.17

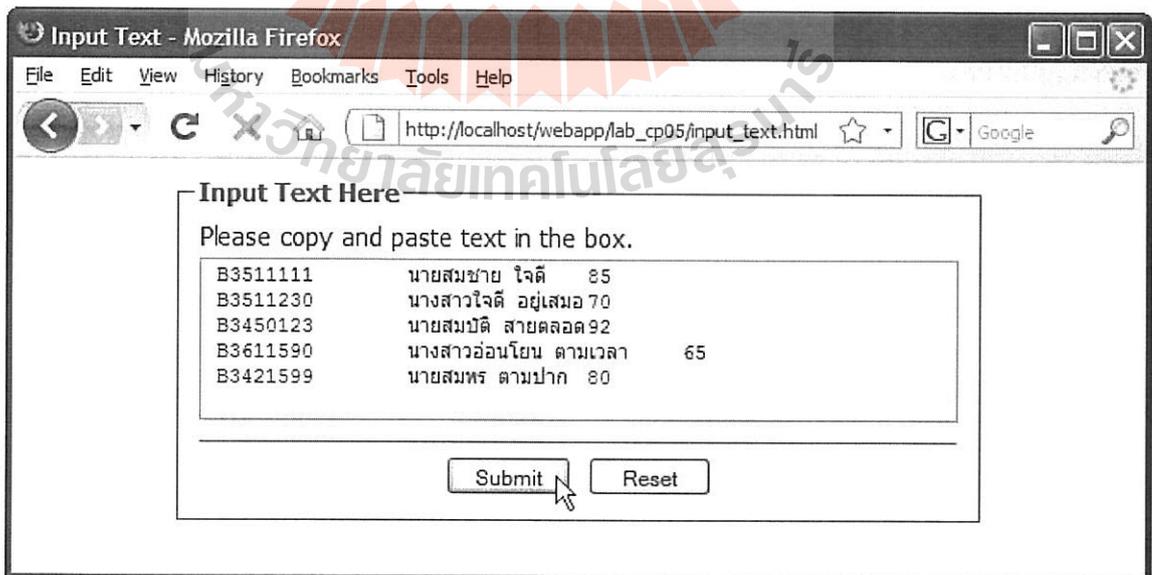


รูปที่ 5.17: การแสดงผลของไฟล์ display_form_info.php

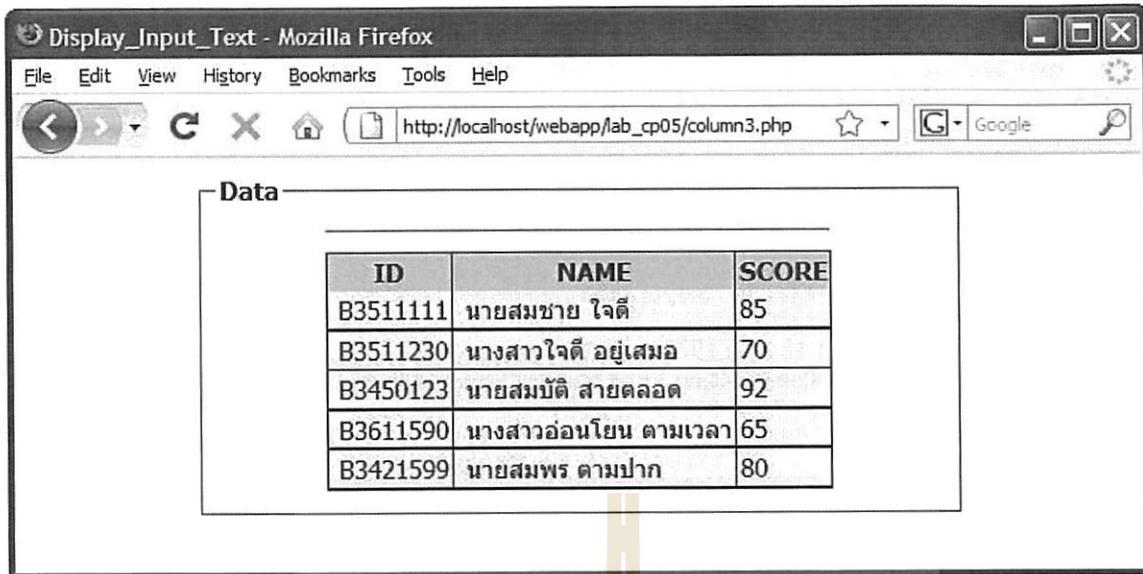
3. สมมติว่ามีข้อมูลที่ถูกเก็บอยู่ในไฟล์ Excel (ดังแสดงในรูปที่ 5.18) ให้ทำการคัดลอก (copy) ข้อมูลดังกล่าวแล้วแปะ (paste) ลงในแบบฟอร์ม input_text.html (ดังแสดงในรูปที่ 5.19) หลังจากนั้นเมื่อกดปุ่ม Submit ก็จะทำให้การส่งข้อมูลทั้งหมดไปที่ไฟล์ column3.php เพื่อทำการแสดงผลดังในรูปที่ 5.20

	A	B	C
1	B3511111	นายสมชาย ใจดี	85
2	B3511230	นางสาวใจดี อยู่เสมอ	70
3	B3450123	นายสมบัติ สายตลอด	92
4	B3611590	นางสาวอ่อนโยน ตามเวลา	65
5	B3421599	นายสมพร ตามปาก	80

รูปที่ 5.18: ข้อมูลในไฟล์ Excel



รูปที่ 5.19: แบบฟอร์มของไฟล์ input_text.html



รูปที่ 5.20: การแสดงผลของไฟล์ column3.php

บทที่ 6 เพิ่มข้อมูล <Data Files>

บทนำ

การรับข้อมูลจากแบบฟอร์มแล้วส่งต่อไปยังสคริปต์เพื่อทำการประมวลผลแล้วแสดงออกที่จอภาพ นั้น ยังไม่เพียงพอสำหรับการจัดการกับข้อมูลเหล่านั้น เนื่องจากเป็นข้อมูลหรือกลุ่มของข้อมูลเพียงชุดเดียว เท่านั้น แล้วข้อมูลก่อนหน้านี้สามารถหาความสัมพันธ์กันได้หรือไม่ คำตอบก็คือ **ไม่ได้** เนื่องจากข้อมูลก่อนหน้านี้ได้หายไปแล้ว ดังนั้นอาจมีความจำเป็นต้องเก็บข้อมูลเหล่านั้นเพื่อการจัดการต่อไป เช่น นำข้อมูลไปประมวลผลแล้วแสดงออกในรูปแบบต่าง ๆ ตามที่ต้องการ นั้นหมายความว่า นอกจากจะทำการประมวลผลข้อมูลที่ใช้ป้อนเข้ามาในแต่ละครั้งแล้ว ยังจะต้องทำการบันทึกข้อมูลเหล่านั้นด้วย ซึ่งในตอนนี้จะขอแนะนำ การเก็บข้อมูลลงในแฟ้มข้อมูลแบบเท็กซ์ (Text File) ก่อน เมื่อก้าวถึงเรื่องของระบบฐานข้อมูลในบทต่อไป ก็สามารถเลือกรูปแบบได้ว่าจะเก็บข้อมูลโดยวิธีการเขียนลงแฟ้มข้อมูลหรือลงระบบฐานข้อมูล ทั้งนี้ก็ขึ้นอยู่กับความเหมาะสมและต้องการของนักเขียนโปรแกรมเอง

วัตถุประสงค์

1. เพื่อศึกษารูปแบบการทำงานของคำสั่งและฟังก์ชันต่าง ๆ ในการเขียนและอ่านแฟ้มข้อมูล
2. เพื่อทดลองเรียกใช้ฟังก์ชันต่าง ๆ ในการเขียนและอ่านแฟ้มข้อมูล
3. เพื่อประยุกต์การใช้งานฟังก์ชันต่าง ๆ ในการแสดงผล

เนื้อหา

การบันทึกหรือเขียนข้อมูลลงแฟ้มข้อมูลนั้น มีขั้นตอนที่สำคัญอยู่ 3 ขั้นตอนคือ

1. เปิดแฟ้มข้อมูลที่ต้องการ ถ้าแฟ้มข้อมูลไม่มี จำเป็นต้องสร้างขึ้นใหม่ แต่ถ้ามีอยู่แล้วจะทำการเขียนทับหรือเขียนต่อทำก็ขึ้นอยู่กับความต้องการของนักเขียนโปรแกรม
2. เขียนข้อมูลลงแฟ้มข้อมูล
3. ปิดแฟ้มข้อมูล

ส่วนการอ่านข้อมูลจากแฟ้มข้อมูลนั้นก็มีลักษณะคล้ายกันกับการเขียนคือ

1. เปิดแฟ้มข้อมูลที่ต้องการ ถ้าแฟ้มข้อมูลไม่มี แจ้งข้อผิดพลาดให้ผู้ใช้ พร้อมหยุดการทำงาน
2. อ่านข้อมูลจากแฟ้มข้อมูล
3. ปิดแฟ้มข้อมูล

6.1 ฟังก์ชันเกี่ยวกับการอ่านและเขียนแฟ้มข้อมูล

6.1.1 การเปิดและปิดแฟ้มข้อมูล

ก่อนทำการอ่านหรือเขียนแฟ้มข้อมูลนั้น ต้องมีการเปิดแฟ้มข้อมูลพร้อมกับระบุว่าต้องการเปิดแฟ้มข้อมูลนั้น ๆ ขึ้นมาเพื่อทำอะไร? เช่น เพื่อเขียน? เขียนต่อท้ายหรือเขียนทับ? หรือเพื่ออ่าน? เป็นต้น ฟังก์ชันที่ใช้ในการเปิดแฟ้มข้อมูลคือฟังก์ชัน `fopen()` ซึ่งมีรูปแบบดังนี้

```
resource fopen (string filename, string mode[, bool
                use_include_path[, resource zcontext]])
```

พารามิเตอร์ที่สำคัญมีอยู่ 2 พารามิเตอร์คือ

`string filename` เป็นสตริงของชื่อแฟ้มข้อมูลที่ต้องการเปิด โดยระบุเส้นทางหรือตำแหน่งที่อยู่ของแฟ้มนั้น

`string mode` เป็นสตริงของโหมด โดยระบุถึงวัตถุประสงค์ที่ต้องการในการเปิดแฟ้มข้อมูลนั้น ดังแสดงในตารางที่ 6.1

ตารางที่ 6.1: โหมดเกี่ยวกับการอ่านและเขียนแฟ้มข้อมูล

โหมด	ความหมาย
"r"	เปิดแฟ้มข้อมูลเพื่อการอ่านเท่านั้น โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งเริ่มต้นของข้อมูล
"r+"	เปิดแฟ้มข้อมูลเพื่อการอ่านและเขียน โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งเริ่มต้นของข้อมูล
"w"	เปิดแฟ้มข้อมูลเพื่อการเขียนเท่านั้น โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งเริ่มต้นของข้อมูลและตัดความยาวของแฟ้มข้อมูลให้เป็น 0 (ศูนย์) นั่นคือถ้ามีแฟ้มข้อมูลนั้นอยู่แล้วจะทำการเขียนทับของเดิม แต่ถ้ายังไม่มีก็จะทำการสร้างแฟ้มข้อมูลขึ้นมาใหม่
"w+"	เปิดแฟ้มข้อมูลเพื่อการอ่านและเขียน โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งเริ่มต้นของข้อมูลและตัดความยาวของแฟ้มข้อมูลให้เป็น 0 (ศูนย์) นั่นคือถ้ามีแฟ้มข้อมูลนั้นอยู่แล้วจะทำการเขียนทับของเดิม แต่ถ้ายังไม่มีก็จะทำการสร้างแฟ้มข้อมูลขึ้นมาใหม่
"a"	เปิดแฟ้มข้อมูลเพื่อการเขียนเท่านั้น โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งสุดท้ายของข้อมูล นั่นคือถ้ามีแฟ้มข้อมูลนั้นอยู่แล้วจะทำการเขียนต่อท้ายข้อมูลเดิม แต่ถ้ายังไม่มีก็จะทำการสร้างแฟ้มข้อมูลขึ้นมาใหม่แล้วเริ่มการเขียน
"a+"	เปิดแฟ้มข้อมูลเพื่อการอ่านและเขียน โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งสุดท้ายของข้อมูล นั่นคือถ้ามีแฟ้มข้อมูลนั้นอยู่แล้วจะทำการเขียนต่อท้ายข้อมูลเดิม แต่ถ้ายังไม่มีก็จะทำการสร้างแฟ้มข้อมูลขึ้นมาใหม่แล้วเริ่มการเขียน
"x"	สร้างและเปิดแฟ้มข้อมูลเพื่อการเขียนเท่านั้น โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งเริ่มต้นของข้อมูล ถ้ามีแฟ้มข้อมูลอยู่แล้ว เมื่อใช้ฟังก์ชัน <code>fopen()</code> จะเกิดข้อผิดพลาด โดยส่งค่ากลับเป็น <code>False</code> ใช้ได้เฉพาะกับแฟ้มข้อมูลที่เป็นแบบ <code>Local</code> เท่านั้น
"x+"	สร้างและเปิดแฟ้มข้อมูลเพื่อการอ่านและเขียน โดยกำหนดให้ตัวชี้แฟ้มข้อมูลอยู่ที่ตำแหน่งเริ่มต้นของข้อมูล ถ้ามีแฟ้มข้อมูลอยู่แล้ว เมื่อใช้ฟังก์ชัน <code>fopen()</code> จะเกิดข้อผิดพลาด โดยส่งค่ากลับเป็น <code>False</code> ใช้ได้เฉพาะกับแฟ้มข้อมูลที่เป็นแบบ <code>Local</code> เท่านั้น
t	ใช้ร่วมกับโหมดอื่น ๆ สำหรับแฟ้มข้อมูลที่เป็นเท็กซ์ เช่น "wt" หรือ "a+t" เป็นต้น
b	ใช้ร่วมกับโหมดอื่น ๆ สำหรับแฟ้มข้อมูลที่เป็นไบนารี เช่น "wb" หรือ "rb" เป็นต้น

ระบบปฏิบัติการต่าง ๆ ได้มีการจัดการการสิ้นสุดของข้อมูลในแต่ละบรรทัด โดยใช้ตัวอักษรเป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัดนั้น ๆ เช่น ระบบปฏิบัติการยูนิกซ์ใช้ตัวอักษร `\n` เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด ส่วนระบบปฏิบัติการวินโดวส์ใช้ตัวอักษร `\r\n` เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด แต่ระบบปฏิบัติการแมคอินทอชใช้ตัวอักษร `\r` เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด ดังนั้นเมื่อต้องการบันทึกข้อมูลลงแฟ้มข้อมูลแบบเท็กซ์และต้องการให้เกิดการขึ้นบรรทัดใหม่ในตำแหน่งต่าง ๆ ที่ต้องการนั้น

จึงมีความจำเป็นต้องใช้ตัวอักษรให้ตรงกับระบบ มิฉะนั้นแล้วการแสดงผลอาจดูผิดปกติไปจากรูปแบบที่กำหนดไว้

ระบบปฏิบัติการวินโดวส์ได้กำหนดตัวอักษร t เพื่อเปิดแฟ้มข้อมูลขึ้นมาเป็นแบบเท็กซ์ และตัวอักษร b เพื่อเปิดแฟ้มข้อมูลขึ้นมาเป็นแบบไบนารี โดยสามารถใส่ตัวอักษรดังกล่าวต่อจากตัวอักษรใหม่ นั่นหมายความว่าเมื่อมีการอ่านหรือเขียนข้อมูลลงบนระบบปฏิบัติการวินโดวส์ ไม่ว่าจะเขียนแบบเท็กซ์หรือแบบไบนารีก็ตาม ควรจะมีการระบุชนิดของข้อมูล (t หรือ b) และใช้ \n เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัดทุกครั้ง เช่น ถ้าต้องการเขียนคำสั่งในการเปิดแฟ้มข้อมูล data.txt ขึ้นมาเพื่อเขียนแบบเท็กซ์ คำสั่งดังกล่าวก็คือ

```
$fp = fopen("data.txt", "wt");
```

การใช้คำสั่งใน PHP เพื่อเปิดแฟ้มข้อมูลหนึ่ง ๆ ขึ้นมานั้น แฟ้มข้อมูลนั้นไม่ควรอยู่ที่เดียวกันกับแฟ้ม PHP ที่ใช้คำสั่งในการเปิดแฟ้ม เนื่องจากง่ายต่อการโจมตีหรือทำลายจากผู้ไม่ประสงค์ดี แฟ้มข้อมูลดังกล่าวควรอยู่ที่อื่น สามารถทำได้โดยการเรียกใช้ตัวแปรของระบบของ PHP คือ ตัวแปร \$DOCUMENT_ROOT เพื่อแสดงตำแหน่งเส้นทางของเอกสาร ซึ่งในที่นี้จะกำหนดให้แฟ้มข้อมูลที่ชื่อว่า data.txt อยู่ในโฟลเดอร์ชื่อว่า data โดยโฟลเดอร์ data นี้จะอยู่ที่ต้นทางของเซิร์ฟเวอร์ โดยใช้คำสั่งดังต่อไปนี้

```
$DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
$fp = fopen("$DOCUMENT_ROOT/./data/data.txt", "wt");
```

ในบรรทัดแรกนั้นเป็นการกำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off แล้วในบรรทัดต่อไปเป็นการเปิดแฟ้มข้อมูลชื่อว่า data.txt อยู่ในโฟลเดอร์ชื่อว่า data ซึ่งโฟลเดอร์นี้จะอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก (เครื่องหมาย . . เป็นตัวบ่งบอกถึงต้นทางของแฟ้มข้อมูลหลัก) วิธีการเช่นนี้สามารถช่วยป้องกันการโจรกรรมข้อมูลจากผู้ไม่ประสงค์ดีได้ระดับหนึ่ง

หมายเหตุ เครื่องหมาย / (Forward Slash) ใช้ได้ในระบบปฏิบัติการ UNIX และระบบปฏิบัติการวินโดวส์ได้ ส่วนเครื่องหมาย \ (Back Slash) นั้นใช้ได้เฉพาะกับระบบปฏิบัติการวินโดวส์เท่านั้น และถ้าต้องการใช้เครื่องหมาย \ จะต้องใช้ตัวอักษรควมพิเศษ (Escape Character) โดยใส่ตัวอักษร \ นำหน้าเครื่องหมาย \ นั้น ๆ เช่น

```
$fp = fopen("$DOCUMENT_ROOT\\..\\data\\data.txt", "wt");
```

เมื่อเปิดแฟ้มข้อมูลได้แล้ว สมมุติว่าต้องการบันทึกข้อความ 2 บรรทัดลงแฟ้มข้อมูล โดยข้อมูลของบรรทัดแรกคือข้อความ Suranaree University of Technology ส่วนบรรทัดที่ 2 คือข้อความ Nakhon Ratchasima เวลาที่จะเขียนข้อมูลลงแฟ้มข้อมูลนั้น เมื่อสิ้นสุดข้อความของข้อมูลในบรรทัดแรกแล้ว ต้องขึ้นบรรทัดใหม่โดยใช้ \n เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัดแล้วจึงตามด้วยข้อความต่อไป ซึ่งสามารถใช้คำสั่งได้ดังต่อไปนี้

```
fwrite($fp, "Suranaree University of Technology\n");
fwrite($fp, "Nakhon Ratchasima\n");
```

เมื่อเปิดแฟ้มและเขียนข้อมูลลงแฟ้มเรียบร้อยแล้ว ก่อนจบโปรแกรมต้องทำการปิดแฟ้มข้อมูลที่ถูกเปิดขึ้นมา โดยใช้ฟังก์ชัน `fclose()` ซึ่งมีรูปแบบการทำงานดังนี้

```
bool fclose (resource handle)
```

รูปแบบของคำสั่งในการเปิดแฟ้มข้อมูลชื่อว่า `data.txt` เพื่อทำการอ่านและปิดแฟ้มข้อมูลนั้น

```
1. <?php
2.   $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.   $fp = fopen("$DOCUMENT_ROOT/../data/data.txt","rt");
4.   echo "Opening file \"data.txt\" for reading only!";
5.   fclose($fp);
6.   ?>
```

```
Warning: fopen(D:/AppServ/www/../data/data.txt) [function.fopen]: failed to open stream: No such file or directory in D:\AppServ\www\php_book\cp6\open1.php on line 3
Opening file \"data.txt\" for reading only!
Warning: fclose(): supplied argument is not a valid stream resource in D:\AppServ\www\php_book\cp6\open1.php on line 5
```

รูปที่ 6.1: การใช้ฟังก์ชัน `fopen()` และ `fclose()`

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร `$DOCUMENT_ROOT` ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ `register_globals` ว่าเป็น `on` หรือ `off` โดยตัวแปร `$DOCUMENT_ROOT` เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 เปิดแฟ้มข้อมูลชื่อว่า `data.txt` ในโฟลเดอร์ `data` ซึ่งอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ (t คือแบบเท็กซ์ ส่วน b คือแบบไบนารี) เพื่ออ่านเท่านั้น โดยให้ตัวแปร `$fp` เป็นตัวชี้แฟ้มไปยังแฟ้มข้อมูลที่เปิดขึ้นมา

บรรทัดที่ 4 แสดงข้อความ "Opening file "data.txt" for reading only!" ออกที่จอภาพ

บรรทัดที่ 5 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม `$fp` ชี้อยู่ ซึ่งก็คือแฟ้มข้อมูลชื่อว่า `data.txt` นั้นเอง

จะสังเกตเห็นว่ามีคำเตือนในการใช้ฟังก์ชัน `fopen()` และ `fclose()` ทั้งนี้เนื่องจากไม่มีแฟ้มข้อมูลดังที่ระบุ ตรงนี้นักเขียนโปรแกรมสามารถใช้ตัวดำเนินการควบคุมข้อผิดพลาด `@` นำหน้าตัวแปร `$fp` เพื่อควบคุมไม่ให้ข้อความเตือนดังกล่าวแสดงออกที่จอภาพ เนื่องจากข้อความเตือนดังกล่าวไม่ได้สื่อความหมายที่เข้าใจง่ายกับผู้ใช้ทั่วไป ดังนั้นเมื่อใช้แล้ว สามารถกำหนดข้อความที่ผู้ใช้ทั่วไปอ่านแล้วเข้าใจง่าย เช่นดังตัวอย่างในรูปที่ 6.2

```
1. <?php
2.   $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.   @ $fp = fopen("$DOCUMENT_ROOT/../data/data.txt","rt");
4.   if(!$fp){
5.     echo "Cannot open file \"data.txt\" at this time, please try again later...";
6.     exit;
7.   }
8.   echo "Opening file \"data.txt\" for reading only!";
9.   fclose($fp);
10.  ?>
```

```
Cannot open file "data.txt" at this time, please try again later...
```

รูปที่ 6.2: การใช้ตัวดำเนินการควบคุมข้อผิดพลาด `@`

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ตัวดำเนินการควบคุมข้อผิดพลาด @ นำหน้าตัวแปร \$fp ที่เป็นตัวชี้แฟ้มไปยังแฟ้มข้อมูลชื่อว่า data.txt ในโฟลเดอร์ data ซึ่งอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์เพื่ออ่านเท่านั้น

บรรทัดที่ 4 ทดสอบดูว่าแฟ้มข้อมูลที่ระบุนั้นได้ถูกเปิดสำเร็จหรือไม่ ถ้าเปิดได้สำเร็จก็จะกระโดดไปบรรทัดที่ 8 แต่ถ้าไม่สำเร็จก็จะเข้าไปทำบรรทัดที่ 5 และ 6 ตามลำดับ ซึ่งไม่สามารถเปิดแฟ้มข้อมูลได้สำเร็จเนื่องจากไม่มีแฟ้มข้อมูลดังที่ระบุ จึงเข้าไปทำคำสั่งในบรรทัดที่ 5

บรรทัดที่ 5 แสดงข้อความ “Cannot open file “data.txt” at this time, please try again later...”

บรรทัดที่ 6 คำสั่ง exit ทำให้ออกจากโปรแกรม นั่นคือจบการทำงานของโปรแกรม

บรรทัดที่ 8 แสดงข้อความ “Opening file “data.txt” for reading only!” ออกที่จอภาพ

บรรทัดที่ 9 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม \$fp ชี้อยู่ ซึ่งก็คือแฟ้มข้อมูลชื่อว่า data.txt นั่นเอง

6.1.2 การเขียนข้อมูลลงแฟ้มข้อมูล

การจัดการแฟ้มข้อมูลในแต่ละครั้งนั้น ควรมีการป้องกันโดยทำการล็อกแฟ้มข้อมูลนั้น ๆ ทั้งนี้เพื่อป้องกันข้อผิดพลาดที่อาจจะเกิดขึ้นกับข้อมูลเดิมที่อยู่ในแฟ้มข้อมูล หรือข้อมูลใหม่ที่กำลังจะถูกรสร้างขึ้น ตัวอย่างเช่น ถ้าเกิดมีการสั่งซื้อสินค้าผ่านทางเว็บไซต์เกิดขึ้นพร้อมกันของลูกค้า 2 คน ทั้ง 2 คนก็พยายามเปิดแฟ้มข้อมูลโดยใช้ฟังก์ชัน fopen() เพื่อต้องการเขียนรายการสินค้าที่ตัวเองต้องการ คำถามที่จะเกิดขึ้นก็คือ

1. ข้อมูลที่จะถูกบันทึกนั้นจะเป็นอย่างไร?
2. ข้อมูลของคนที่ 1 หรือคนที่ 2 จะถูกบันทึก?
3. จะทำการบันทึกข้อมูลของคนที่ 1 แล้วตามด้วยคนที่ 2 หรือคนที่ 2 แล้วตามด้วยคนที่ 1?
4. จะเกิดการสั่งของเพียงรายการเดียวหรือทั้ง 2 รายการ?
5. การบันทึกข้อมูลของการสั่งของนั้นอาจจะเป็นข้อมูลที่ไม่มีประโยชน์หรือไม่ เนื่องจากข้อมูลดังกล่าวอาจเกิดการสอดแทรกกันขึ้นในขณะที่ทำการบันทึก?

คำถามเหล่านั้นจำเป็นต้องหาคำตอบให้ชัดเจนว่าจะเกิดอะไรขึ้น แล้วจะรู้ได้อย่างไรว่าสิ่งที่เกิดขึ้นคืออะไร ทั้งนี้ขึ้นอยู่กับระบบปฏิบัติการของเว็บไซต์ แต่ไม่สามารถที่จะระบุให้แน่ชัดได้ว่าสิ่งนั้นหรือสิ่งนี้จะเกิดขึ้น ซึ่งโดยปกติแล้วเหตุการณ์ที่คน 2 คน (หรือมากกว่า) ต้องการสั่งของพร้อมกันนั้นจะไม่เกิดขึ้น แต่ถ้าสินค้าชิ้นนั้นเกิดเป็นสินค้าที่ผู้ใช้นิยมใช้กันมาก การสั่งซื้อสินค้าก็จะมากตามไปด้วย แล้วเหตุการณ์ที่ไม่คาดคิดก็อาจจะเกิดขึ้นได้ ดังนั้นวิธีป้องกันที่ดีที่สุดก็คือ เมื่อมีการเปิดแฟ้มข้อมูลขึ้นมาแล้ว (ไม่ว่าคนที่ 1 หรือคนที่ 2 จะเปิดก่อน) ให้ทำการล็อกแฟ้มข้อมูลนั้นเลยก่อนทำการอ่านหรือเขียน โดยใช้ฟังก์ชัน flock() ซึ่งมีรูปแบบดังนี้

```
bool flock (resource handle, int operation[,int &wouldblock])
```

ฟังก์ชัน flock() จะส่งค่ากลับเป็นจริง (True) ก็ต่อเมื่อทำการล็อกเพิ่มข้อมูลได้สำเร็จ มิฉะนั้นแล้ว จะส่งค่ากลับเป็นเท็จ (False) พารามิเตอร์ที่สำคัญมีอยู่ 2 พารามิเตอร์คือ

resource handle เป็นตัวแปรที่จะชี้ไปที่เพิ่มข้อมูล

int operation เป็นคำสั่งที่ใช้ระบุการกระทำ เช่น ล็อกเพิ่มข้อมูลเพื่ออ่านหรือ ล็อกเพิ่มข้อมูลเพื่อเขียน เป็นต้น ดังแสดงในตารางที่ 6.2

ตารางที่ 6.2: การล็อกเพิ่มข้อมูล

โหมด	ความหมาย
LOCK_SH (formerly 1)	ล็อกเพิ่มข้อมูลเพื่อการอ่าน ซึ่งสามารถให้ผู้อื่นอ่านได้ด้วย
LOCK_EX (formerly 2)	ล็อกเพิ่มข้อมูลเพื่อการเขียน ซึ่งผู้อื่นไม่สามารถเขียนได้
LOCK_UN (formerly 3)	ปลดล็อกเพิ่มข้อมูลที่ถูกล็อก
LOCK_NB (formerly 4)	เมื่อไม่ต้องการให้ฟังก์ชัน flock() ปิดขวางการทำงานของสคริปต์อื่น ๆ ระหว่างที่ล็อก

หมายเหตุ บนระบบปฏิบัติการ UNIX นั้นจะอนุญาตให้การล็อกเพิ่มข้อมูลขึ้นอยู่กับความต้องการของผู้เขียนโปรแกรม แต่บนระบบปฏิบัติการวินโดวส์นั้น ระบบจะทำการล็อกไฟล์โดยอัตโนมัติ

การเขียนข้อมูลลงเพิ่มข้อมูลแบบเท็กซ์นั้นสะดวกและง่ายต่อการจัดการ สามารถใช้ฟังก์ชัน fwrite() หรือ fputs() ได้ โดยฟังก์ชันทั้งสองนี้มีรูปแบบการใช้งานเหมือนกัน ในที่นี้จะเรียกใช้ฟังก์ชัน fwrite() คือ

```
int fwrite (resource handle, string string [, int length])
```

ฟังก์ชัน fwrite() จะส่งค่ากลับเป็นจำนวนไบต์ที่เขียน ซึ่งสามารถระบุได้ว่าต้องการเขียนกี่ไบต์ ฟังก์ชันนี้จะหยุดการเขียนก็ต่อเมื่อพบว่าได้เขียนถึงจำนวนไบต์ที่ระบุแล้วหรือสิ้นสุดข้อความที่จะเขียนแล้ว

รูปแบบการใช้ฟังก์ชัน flock() กับการเปิดเพิ่มข้อมูลชื่อว่า data.txt แบบเท็กซ์เพื่อทำการเขียน และปิดเพิ่มข้อมูลโดยใช้ \t เป็นตัวกั้นข้อมูลแต่ละสตริง และใช้ \n เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด

```

1. <?php
2. $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3. @ $fp = fopen("$DOCUMENT_ROOT/../data/data2.txt", "wt");
4. if (!$fp) {
5.     echo "Cannot open file \"data2.txt\" at this time, please try again later...";
6.     exit;
7. }
8. echo "Opening file \"data2.txt\" for writing only!";
9. if (flock($fp, LOCK_EX)) { // do an exclusive lock
10.     $output1 = "Computer Engineering\tEngineering\tSUT\tNakhon Ratchasima\n";
11.     $output2 = "Information Technology\tSocial Technology\tSUT\tNakhon Ratchasima\n";
12.     fwrite($fp, $output1);
13.     fputs($fp, $output2);
14.     flock($fp, LOCK_UN); // release the lock
15. } else {
16.     echo "Couldn't lock the file !";
17. }
18. fclose($fp);
19. ?>
    
```

Opening file "data2.txt" for writing only!

รูปที่ 6.3: การใช้ฟังก์ชัน flock()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ตัวดำเนินการควบคุมข้อผิดพลาด @ นำหน้าตัวแปร \$fp ที่เป็นตัวชี้แฟ้มไปยังแฟ้มข้อมูลชื่อว่า data2.txt ในโฟลเดอร์ data ซึ่งอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ เพื่อเขียนเท่านั้น

บรรทัดที่ 4 ทดสอบดูว่าแฟ้มข้อมูลที่ระบุนั้นได้ถูกเปิดสำเร็จหรือไม่ ถ้าเปิดได้สำเร็จก็จะกระโดดไปบรรทัดที่ 8 แต่ถ้าไม่สำเร็จก็จะเข้าไปทำบรรทัดที่ 5 และ 6 ตามลำดับ ซึ่งสามารถเปิดแฟ้มข้อมูลได้สำเร็จเนื่องจากมีแฟ้มข้อมูลดังที่ระบุ จึงกระโดดไปบรรทัดที่ 8

บรรทัดที่ 8 แสดงข้อความ "Opening file "data2.txt" for writing only!" ออกที่จอภาพ

บรรทัดที่ 9 ทดสอบดูว่าสามารถทำการล็อกแฟ้มข้อมูลโดยใช้ฟังก์ชัน flock() แบบ LOCK_EX ที่ถูกเปิดขึ้นมาได้หรือไม่ ถ้าล็อกได้สำเร็จก็จะไปทำคำสั่งในบรรทัดที่ 10 ถึง 14 แต่ถ้าไม่สำเร็จก็จะกระโดดไปทำคำสั่งในบรรทัดที่ 16 ที่จะแสดงข้อความ "Couldn't lock the file !" ซึ่งเป็นส่วนของ else แทน ในกรณีนี้สามารถล็อกแฟ้มข้อมูลได้สำเร็จ จึงเข้าไปทำคำสั่งในบรรทัดที่ 10

บรรทัดที่ 10 กำหนดข้อมูลที่เป็นสตริง 4 สตริงให้กับตัวแปร \$output1 โดยใช้ตัวอักษร \t เป็นตัวกั้นระหว่างสตริงเหล่านั้นและใช้ตัวอักษร \n เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด

บรรทัดที่ 11 กำหนดข้อมูลที่เป็นสตริง 4 สตริงให้กับตัวแปร \$output2 โดยใช้ตัวอักษร \t เป็นตัวกั้นระหว่างสตริงเหล่านั้นและใช้ตัวอักษร \n เป็นตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด

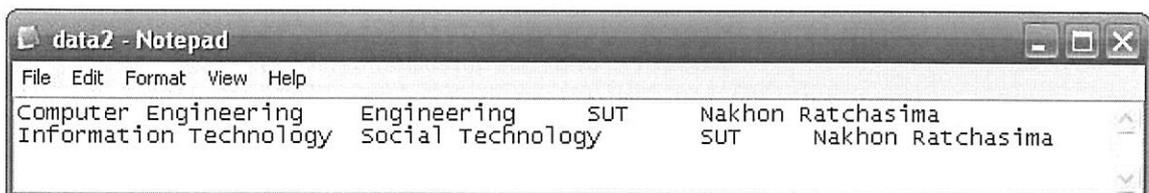
บรรทัดที่ 12 ใช้ฟังก์ชัน fwrite() ในการเขียนสตริงหรือข้อความที่อยู่ในตัวแปร \$output1 โดยแต่ละสตริงจะถูกกั้นด้วยเครื่องหมาย \t เมื่อสิ้นสุดการเขียนสตริงตัวสุดท้ายแล้ว จะเลื่อน Cursor ไปที่ตำแหน่งแรกของบรรทัดใหม่

บรรทัดที่ 13 ใช้ฟังก์ชัน fputs() ในการเขียนข้อความที่อยู่ในตัวแปร \$output2 โดยแต่ละสตริงจะถูกกั้นด้วยเครื่องหมาย \t เมื่อสิ้นสุดการเขียนสตริงตัวสุดท้ายแล้ว จะเลื่อน Cursor ไปที่ตำแหน่งแรกของบรรทัดใหม่

บรรทัดที่ 14 ใช้ฟังก์ชัน flock() อีกครั้งแบบ LOCK_UN คือการปลดล็อกแฟ้มข้อมูลที่ถูกเปิดขึ้นมา

บรรทัดที่ 18 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม \$fp ซึ่งก็คือแฟ้มข้อมูลชื่อว่า data2.txt นั่นเอง

เมื่อใช้โปรแกรม Notepad เปิดแฟ้มข้อมูล data2.txt ข้อมูลที่ปรากฏจะเป็นดังแสดงในรูปที่ 6.4



รูปที่ 6.4: การแสดงผลของข้อมูลที่อยู่ในแฟ้ม data2.txt โดยโปรแกรม Notepad

หมายเหตุ ฟังก์ชัน flock() นั้นไม่สามารถใช้ร่วมกับแฟ้มข้อมูลแบบ NFS (Network File System) หรือแฟ้มระบบเครือข่ายแบบอื่น ๆ ได้ รวมไปถึงแฟ้มระบบแบบเก่า เช่น FAT ที่ใช้กันอย่าง

แพร่หลายในระบบปฏิบัติการวินโดวส์ 95 และ 98 ด้วย แต่ทำงานได้ดีกับเพิ่มข้อมูลแบบ NTFS และ FAT32 ดังนั้นจึงควรระวังการใช้ฟังก์ชัน flock() เนื่องจากผลที่คาดหวังอาจคลาดเคลื่อนได้

การเขียนข้อมูลลงเพิ่มข้อมูลอีกลักษณะหนึ่งคือการเขียนแบบ CSV ซึ่งย่อมาจาก Comma Separated Version เป็นเพิ่มข้อมูลที่แต่ละบรรทัดคือ 1 เรคอร์ด และใน 1 เรคอร์ด นั้นแบ่งเป็นฟิลด์หลายฟิลด์ โดยแต่ละฟิลด์จะถูกคั่นด้วยเครื่องหมาย , (Comma) ข้อมูลแบบนี้จะสะดวกในการเปิดด้วยโปรแกรมไมโครซอฟต์เอ็กเซล อีกทั้งยังสามารถเชื่อมต่อเข้ายังระบบฐานข้อมูลได้อีกด้วย ดังตัวอย่างในรูปที่ 6.5

```

1.  <?php
2.  $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.  @ $fp = fopen("$DOCUMENT_ROOT/./data/data2.csv","wt");
4.  if(!$fp){
5.      echo "Cannot open file \"data2.csv\" at this time, please try again later...";
6.      exit;
7.  }
8.  echo "Opening file \"data2.csv\" for writing only!";
9.  if(flock($fp, LOCK_EX)) { // do an exclusive lock
10.     $output1 = "Computer Engineering,Engineering,SUT,Nakhon Ratchasima\n";
11.     $output2 = "Information Technology,Social Technology,SUT,Nakhon Ratchasima\n";
12.     fwrite($fp, $output1);
13.     fputs($fp, $output2);
14.     flock($fp, LOCK_UN); // release the lock
15. }else{
16.     echo "Couldn't lock the file !";
17. }
18. fclose($fp);
19. ?>

```

Opening file "data2.csv" for writing only!

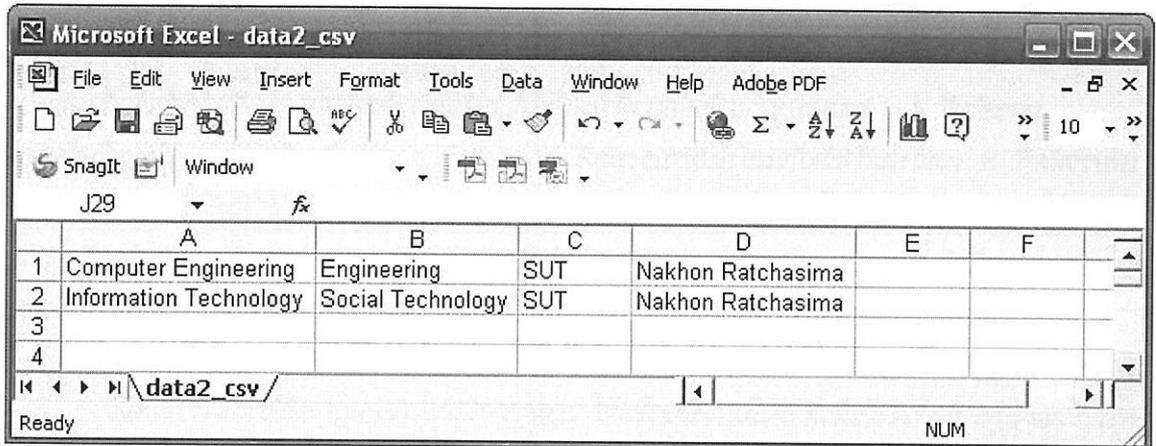
รูปที่ 6.5: การบันทึกข้อมูลเป็นสกุล CSV

รูปที่ 6.6 แสดงถึงการเปิดเพิ่มข้อมูล data2.csv ด้วยโปรแกรม Notepad



รูปที่ 6.6: การแสดงข้อมูลของเพิ่มข้อมูล data2_csv.csv โดยโปรแกรม Notepad

รูปที่ 6.7 แสดงถึงการเปิดแฟ้มข้อมูล data2_csv.csv ด้วยโปรแกรม Microsoft Excel



รูปที่ 6.7: การแสดงข้อมูลของแฟ้มข้อมูล data2_csv.csv โดยโปรแกรม Microsoft Excel

6.1.3 การอ่านข้อมูลจากแฟ้มข้อมูล

การอ่านข้อมูลแบบเท็กซ์จากแฟ้มข้อมูลนั้นสามารถทำได้โดยใช้โปรแกรม Notepad เปิดแฟ้มข้อมูลนั้น ๆ ขึ้นมา แต่ในที่นี้จะนำเสนอการเขียนคำสั่งที่จะทำการเปิดแฟ้มข้อมูลแบบเท็กซ์ขึ้นมาเพื่ออ่านอย่างเดียว แล้วแสดงผลข้อมูลทั้งหมดออกที่จอภาพ สามารถทำได้โดยใช้คำสั่งในการอ่านข้อมูลจากแฟ้มข้อมูลไปเรื่อย ๆ จนกว่าจะถึงจุดสุดท้ายของข้อมูล โดยใช้ฟังก์ชัน feof() ในการทดสอบ เช่น

```
while(!feof($fp))
```

ฟังก์ชัน feof() รับตัวแปร \$fp ที่เป็นตัวชี้แฟ้มข้อมูล จะส่งค่ากลับเป็นจริงถ้าตัวชี้แฟ้มข้อมูลอยู่ตำแหน่งสุดท้ายของข้อมูล ดังตัวอย่างการอ่านข้อมูลในรูปที่ 6.8

```

1. <?php
2. $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3. @ $fp = fopen("$DOCUMENT_ROOT/../data/data2.txt","rt");
4. if(!$fp){
5.     echo "Cannot open file \"data2.txt\" at this time, please try again later...";
6.     exit;
7. }
8. echo "Opening file \"data2.txt\" for reading only!";
9. while (!feof($fp)){
10.     $output= fgets($fp, 999);
11.     echo $output.'<br>';
12. }
13. fclose($fp);
14. ?>
    
```

```

Opening file "data2.txt" for reading only!
Computer Engineering Engineering SUT Nakhon Ratchasima
Information Technology Social Technology SUT Nakhon Ratchasima
    
```

รูปที่ 6.8: การใช้ลูป while กับฟังก์ชัน feof() เพื่ออ่านแฟ้มข้อมูลและแสดงผล

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่อง การตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปร ระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ตัวดำเนินการควบคุมข้อผิดพลาด @ นำหน้าตัวแปร \$fp ที่เป็นตัวชี้แฟ้มไปยังแฟ้มข้อมูลชื่อว่า data2.txt ในโฟลเดอร์ data ซึ่งอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ เพื่ออ่านเท่านั้น

บรรทัดที่ 4 ทดสอบดูว่าแฟ้มข้อมูลที่ระบุนั้นได้ถูกเปิดสำเร็จหรือไม่ ถ้าเปิดได้สำเร็จก็จะกระโดดไปบรรทัดที่ 8 แต่ถ้าไม่สำเร็จก็จะเข้าไปทำบรรทัดที่ 4 และ 5 ตามลำดับ ซึ่งสามารถเปิดแฟ้มข้อมูลได้สำเร็จเนื่องจากมีแฟ้มข้อมูลดังที่ระบุ จึงกระโดดไปบรรทัดที่ 8

บรรทัดที่ 8 แสดงข้อความ “Opening file “data2.txt” for reading only!” ออกที่จอภาพ

บรรทัดที่ 9 ใช้ลูป while ในการทดสอบว่าตัวชี้แฟ้มข้อมูลไม่ได้ชี้อยู่ที่ตำแหน่งสุดท้ายของข้อมูล ถ้าจริงจะเข้าไปทำคำสั่งในบรรทัดที่ 10 ถึง 11 แต่ถ้าไม่ใช่จะกระโดดไปทำคำสั่งในบรรทัดที่ 13 ในกรณีนี้คือ **จริง** ดังนั้นจึงไปทำคำสั่งในบรรทัดที่ 10

บรรทัดที่ 10 ใช้ฟังก์ชัน fgets() โดยมีพารามิเตอร์ 2 ตัวคือ \$fp และ 999 คือการรับสตริงเข้ามาที่ละบรรทัด จะหยุดรับข้อมูลก็ต่อเมื่อเจอตัวบ่งบอกตำแหน่งสิ้นสุดของบรรทัด \n หรือเมื่อสิ้นสุดข้อมูลในแฟ้มข้อมูล หรือเมื่อรับสตริงเข้ามาเป็นจำนวนไบต์ที่ระบุ - 1 นั่นคือ 998 ไบต์จากแฟ้มข้อมูล โดยข้อมูลที่รับเข้ามานั้นจะถูกเก็บไว้ที่ตัวแปร \$output (ฟังก์ชัน fgets() จะส่งค่ากลับเป็นสตริงที่มีความยาวเท่ากับค่าของความยาวที่ระบุ - 1)

บรรทัดที่ 11 แสดงข้อมูลที่อยู่ในตัวแปร \$output ออกที่จอภาพพร้อมกับเลื่อนตำแหน่ง Cursor ไปที่จุดเริ่มต้นของบรรทัดใหม่

บรรทัดที่ 13 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม \$fp ชี้อยู่ ซึ่งคือแฟ้มข้อมูล data2.txt นั่นเอง

ฟังก์ชัน fgets() ก็มีการทำงานคล้ายกันกับฟังก์ชัน fgetss() ที่แตกต่างกันก็คือว่า ฟังก์ชัน fgetss() นั้นจะไม่สนใจแท็ก HTML หรือ PHP ที่พบในสตริงนั้น ๆ ซึ่งมีรูปแบบดังต่อไปนี้

```
string fgetss(resource handle [,int length [,string allowable_tags]])
```

สมมติว่า แฟ้มข้อมูลหนึ่งเมื่อถูกเปิดโดยโปรแกรม Notepad ข้อมูลที่ถูกแสดงออก ปรากฏดังแสดงในรูปที่ 6.9

```
<h1>Computer Engineering Engineering SUT Nakhon Ratchasima<h1>
<i>Information Technology Social Technology SUT Nakhon Ratchasima<i>
```

รูปที่ 6.9: ข้อมูลจากแฟ้มข้อมูลหนึ่งที่ถูกแสดงออกโดยโปรแกรม Notepad

แล้วมีการใช้ส่วนของคำสั่งต่อไปนี้ เพื่อดึงข้อมูลออกโดยใช้ฟังก์ชัน fgetss()

```
$output = fgetss($fp, 999);
echo $output. '<br>';
```

ผลที่จะถูกแสดงออกคือ

```
Computer Engineering Engineering SUT Nakhon Ratchasima
Information Technology Social Technology SUT Nakhon Ratchasima
```

รูปที่ 6.10: ข้อมูลจากแฟ้มข้อมูลหนึ่งที่ถูกแสดงออกโดยฟังก์ชัน fgets()

ที่เป็นเช่นนั้นเนื่องมาจากการใช้ฟังก์ชัน `fgetss()` ซึ่งจะไม่สนใจแท็ก HTML หรือ PHP ที่พบในสตริงนั้น ๆ แต่สามารถใส่ตัวเลือกที่ต้องการยกเว้นแท็กนั้น ๆ ให้แสดงออกได้ เช่น การใช้คำสั่ง

```
$output = fgetss($fp, 999, "<i>");
echo $output. '<br>';
```

ผลที่จะถูกแสดงออกคือ

Computer Engineering Engineering SUT Nakhon Ratchasima
Information Technology Social Technology SUT Nakhon Ratchasima

รูปที่ 6.11: ข้อมูลจากเพิ่มข้อมูลหนึ่งที่ถูกแสดงออกโดยฟังก์ชัน `fgetss()` และยกเว้นแท็ก `<i>`

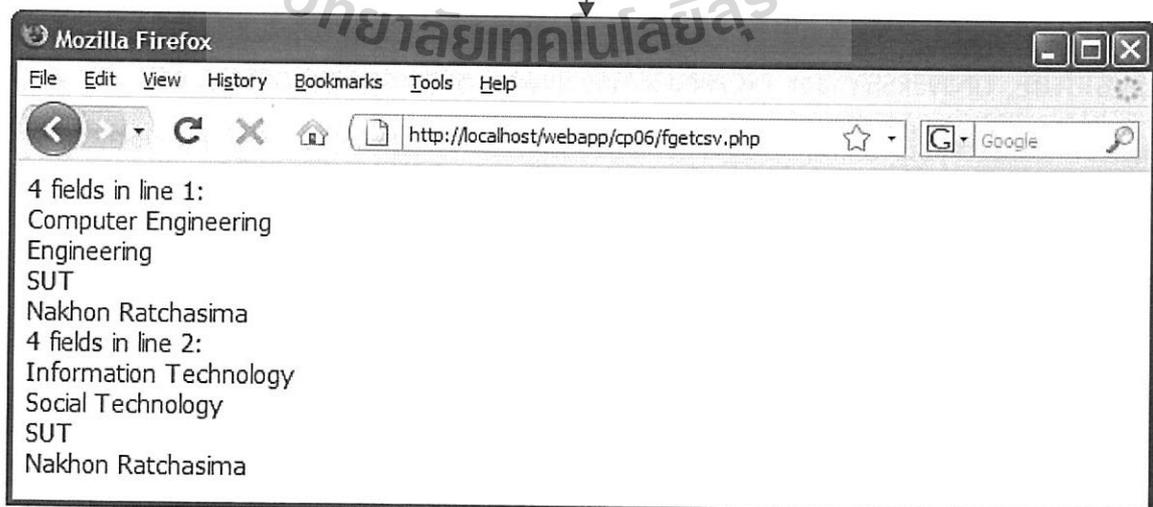
นั่นคือการใช้ฟังก์ชัน `fgetss()` โดยมีการระบุว่าจะให้แท็ก `<i>` แท็กเดียวที่สามารถแสดงผลได้

ฟังก์ชันที่มีการทำงานคล้ายกันกับฟังก์ชัน `fgets()` ก็คือฟังก์ชัน `fgetcsv()` ซึ่งเหมาะสำหรับการรับข้อมูลจากเพิ่มข้อมูลที่มีการใช้ตัวกั้น (Delimiter) ซึ่งอาจจะเป็นตัวอักษร `\t` หรือเครื่องหมาย `,` (Comma) เป็นต้น ซึ่งทำให้ข้อมูลที่รับเข้ามานั้นสะดวกและง่ายต่อการนำไปประมวลผลหรือแสดงผลในรูปแบบของตารางมีรูปแบบการทำงานดังนี้

```
array fgetcsv(int fp, int length[, string delimiter
[, string enclosure]])
```

ตัวอย่าง การใช้ฟังก์ชัน `fgetcsv()` เพื่อรับข้อมูลจากเพิ่มข้อมูลที่มีตัวอักษร `\t` เป็นตัวกั้น

```
1. <?php
2.     $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.     $fp = fopen("$DOCUMENT_ROOT/./data/data2.txt", "rt");
4.     $row = 1;
5.     while(($data = fgetcsv($fp, 999, "\t")) != FALSE) {
6.         $num = count($data);
7.         echo "$num fields in line $row: <br>";
8.         $row++;
9.         for ($c=0; $c < $num; $c++) {
10.            echo $data[$c] . "<br>";
11.        }
12.    }
13.    fclose($fp);
14.    ?>
```

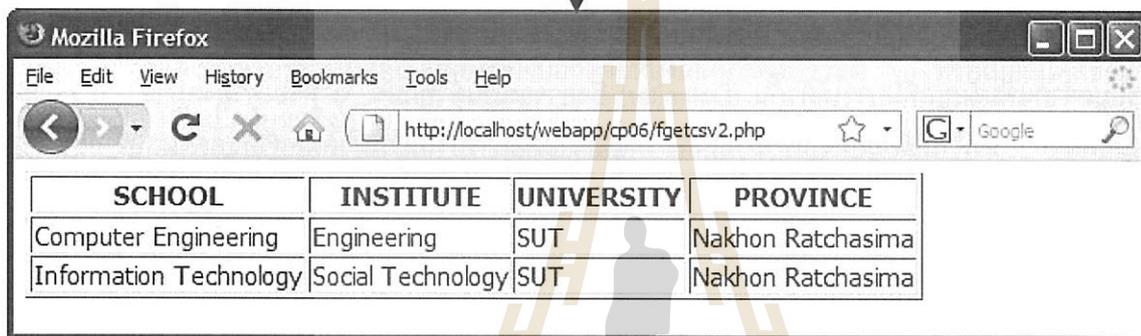


รูปที่ 6.12: การใช้ฟังก์ชัน `fgetcsv()` เพื่อแยกข้อมูลออกเป็นส่วน ๆ

ตัวอย่าง การใช้ฟังก์ชัน fgetcsv() เพื่อรับข้อมูลจากเพิ่มข้อมูลที่มีตัวอักษร \t เป็นตัวกั้นแล้ว แสดงผลของข้อมูลเหล่านั้นในรูปแบบตาราง

```

1. <?php
2. $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3. $fp = fopen("$DOCUMENT_ROOT/./data/data2.txt", "rt");
4. echo "<table border='1'>";
5. echo "<tr><th>SCHOOL</th><th>INSTITUTE</th><th>UNIVERSITY</th><th>PROVINCE</th></tr>";
6. while(($data = fgetcsv($fp, 999, "\t")) != FALSE){
7.     $num = count($data);
8.     echo "<tr>";
9.     for($i=0; $i<$num; $i++) {
10.         echo "<td>$data[$i]</td>";
11.     }
12.     echo "</tr>";
13. }
14. echo "</table>";
15. fclose($fp);
16. ?>
    
```



รูปที่ 6.13: การใช้ฟังก์ชัน fgetcsv() เพื่อแยกข้อมูลออกเป็นส่วน ๆ แล้วแสดงผลในตาราง

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของเพิ่มข้อมูล

บรรทัดที่ 3 กำหนดให้ตัวแปร \$fp ที่เป็นตัวชี้เพิ่มไปยังเพิ่มข้อมูลชื่อว่า data2.txt ในไฟล์เดอร์ data ซึ่งอยู่ที่ต้นทางของเพิ่มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ เพื่ออ่านเท่านั้น

บรรทัดที่ 4 กำหนดการสร้างตารางขึ้น โดยกำหนดให้มีเส้นขอบเท่ากับ 1

บรรทัดที่ 5 เป็นการกำหนดเปิดแถว โดยใช้แท็ก <tr> เนื่องจากเราเป็นผู้เขียนข้อมูลและรู้ว่าในแต่ละแถวนั้นมี 4 คอลัมน์ ดังนั้นในบรรทัดนี้จึงได้กำหนดหัวของตารางขึ้นมา 4 หัวด้วยกันคือ SCHOOL, INSTITUTE, UNIVERSITY และ PROVINCE โดยใช้แท็ก <th> จะสังเกตเห็นว่าข้อความที่อยู่ภายในแท็ก <th> จะถูกแสดงออกเป็นตัวหนาและอยู่ตำแหน่งกึ่งกลาง แล้วตามด้วยแท็กปิด </tr>

บรรทัดที่ 6 เป็นการใช้ลูป while วนรับข้อมูลจากเพิ่มข้อมูลที่ \$fp ซ้ำๆ (นั่นคือ data2.txt) เพื่อทดสอบว่าข้อมูลที่รับเข้ามานั้นยังไม่สิ้นสุด หรือยังไม่ถึงจำนวน 998 ไบต์ ซึ่งรับข้อมูลเข้ามาทีละ 1 บรรทัด แล้วแยกข้อมูลในบรรทัดนั้นออกเป็นสตริงย่อย ๆ โดยใช้ตัวอักษร \t เป็นตัวแบ่ง สตริงเหล่านั้นเป็นข้อมูลแบบอะเรย์จะถูกเก็บไว้ที่ตัวแปร \$data

บรรทัดที่ 7 เป็นการใช้ฟังก์ชัน count() ในการนับจำนวนข้อมูลในตัวแปร \$data แล้วเก็บจำนวนนั้นไว้ที่ตัวแปร \$num ซึ่งค่านั้นก็คือ 4

บรรทัดที่ 8 เป็นการกำหนดเปิดแถว โดยใช้แท็ก <tr>

บรรทัดที่ 9 เป็นการใช้ลูป for วนรอบเท่ากับจำนวนของค่าที่อยู่ในตัวแปร \$num ซึ่งก็คือ 4

บรรทัดที่ 10 เป็นการกำหนดคอลัมน์โดยใช้แท็ก <td> แสดงข้อมูลที่ถูกเก็บไว้ในตัวแปร \$data แล้วปิดคอลัมน์ด้วยแท็ก </td>

บรรทัดที่ 12 เป็นการปิดแถว โดยใช้แท็ก </tr>

บรรทัดที่ 14 เป็นการปิดตาราง โดยใช้แท็ก </table>

บรรทัดที่ 15 ปิดเพิ่มข้อมูลที่ตัวชี้เพิ่ม \$fp ซ้ำอยู่ ซึ่งก็คือเพิ่มข้อมูลชื่อว่า data2.txt นั่นเอง

การอ่านเพิ่มข้อมูลเข้ามาทีละ 1 บรรทัดนั้น อาจจะไม่สะดวกหรือเหมาะสมกับการประมวลผลในบางกรณี ดังนั้นในบางครั้งอาจต้องการอ่านข้อมูลทั้งหมดในเพิ่มข้อมูลเข้ามาในคราวเดียว โดยมีวิธีการทั้งหมด 5 วิธีคือ

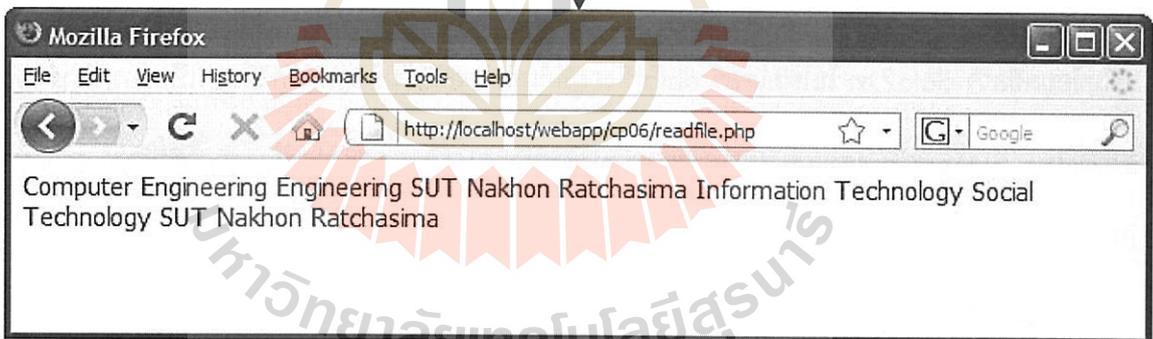
1. ใช้ฟังก์ชัน readfile() ซึ่งมีรูปแบบของการทำงานดังต่อไปนี้

```
int readfile (string filename [,bool use_include_path
[,resource context]])
```

ฟังก์ชัน readfile() นี้จะทำการอ่านเพิ่มข้อมูลทั้งหมดแล้วแสดงออกที่จอภาพ โดยโปรแกรมเว็บเบราว์เซอร์ และส่งค่ากลับเป็นจำนวนไบต์ที่อ่านมาจากเพิ่มข้อมูลนั้น ๆ

ตัวอย่าง การใช้ฟังก์ชัน readfile()

```
1. <?php
2. $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3. readfile("$DOCUMENT_ROOT/../data/data2.txt");
4. ?>
```



รูปที่ 6.14: การใช้ฟังก์ชัน readfile()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของเพิ่มข้อมูล

บรรทัดที่ 3 ใช้ฟังก์ชัน readfile() ในการอ่านเพิ่มข้อมูลชื่อว่า data2.txt ในโฟลเดอร์ data ซึ่งอยู่ที่ต้นทางของเพิ่มข้อมูลหลัก แล้วแสดงผลออกที่เว็บเบราว์เซอร์

2. ใช้ฟังก์ชัน fpassthru() ซึ่งมีรูปแบบของการทำงานดังต่อไปนี้

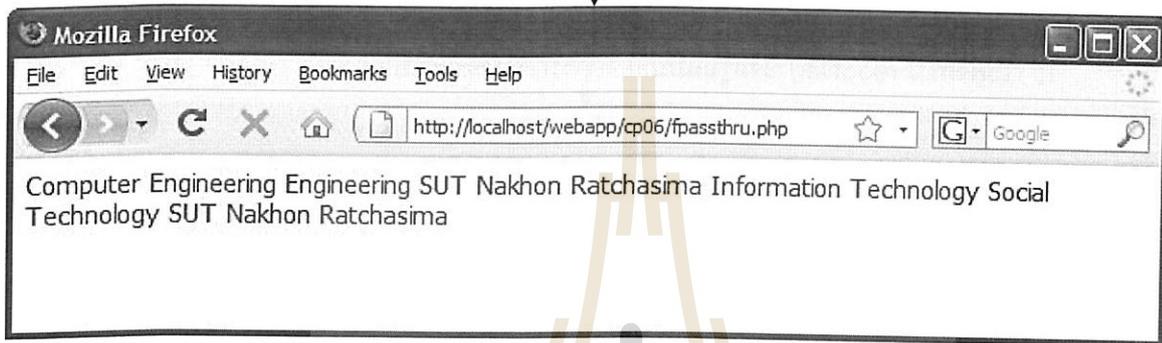
```
int fpassthru (resource handle)
```

ฟังก์ชัน `fpasssthru()` นี้จะส่งค่ากลับเป็นจริงเมื่อสามารถอ่านแฟ้มข้อมูลดังกล่าวได้ มิฉะนั้นแล้วจะส่งค่ากลับเป็นเท็จ ฟังก์ชันนี้จะอ่านข้อมูลทั้งหมดจากแฟ้มข้อมูลจนสุดข้อมูล (EOF) แล้วแสดงข้อมูลนั้นออกที่เว็บเบราว์เซอร์

ตัวอย่าง การใช้ฟังก์ชัน `fpasssthru()`

```

1. <?php
2. $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3. @ $fp = fopen("$DOCUMENT_ROOT/./data/data2.txt", "rt");
4. fpasssthru($fp);
5. fclose($fp);
6. ?>
    
```



รูปที่ 6.15: การใช้ฟังก์ชัน `fpasssthru()`

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร `$DOCUMENT_ROOT` ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ `register_globals` ว่าเป็น `on` หรือ `off` โดยตัวแปร `$DOCUMENT_ROOT` เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ตัวดำเนินการควบคุมข้อผิดพลาด `@` นำหน้าตัวแปร `$fp` ที่เป็นตัวชี้แฟ้มไปยังแฟ้มข้อมูลชื่อว่า `data2.txt` ในโฟลเดอร์ `data` ซึ่งอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ เพื่ออ่านเท่านั้น

บรรทัดที่ 4 ใช้ฟังก์ชัน `fpasssthru()` ในการรับข้อมูลมาจากแฟ้มข้อมูลที่ตัวชี้แฟ้ม `$fp` ชี้อยู่ ซึ่งก็คือแฟ้มข้อมูล `data2.txt` นั่นเอง แล้วแสดงผลออกที่เว็บเบราว์เซอร์

3. ใช้ฟังก์ชัน `file()` ซึ่งมีรูปแบบของการทำงานดังต่อไปนี้

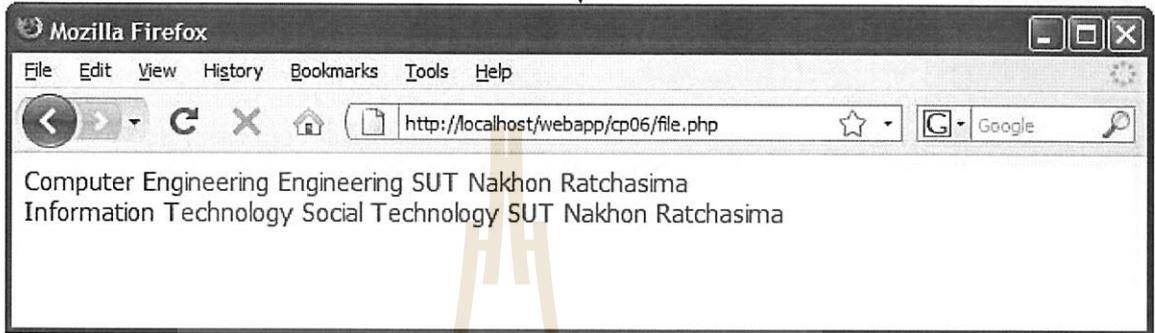
```
array file ($fp);
```

ฟังก์ชัน `file()` นี้จะมีการทำงานเหมือนกันกับฟังก์ชัน `readfile()` แต่สิ่งที่แตกต่างกันคือ ฟังก์ชัน `file()` จะไม่แสดงข้อมูลออกที่จอภาพ แต่จะส่งค่ากลับเป็นข้อมูลแบบอะเรย์ โดยข้อมูลหนึ่ง ๆ ในอะเรย์ก็คือข้อมูลทั้งหมดที่อยู่ในบรรทัดหนึ่ง ๆ ในแฟ้มข้อมูล แต่การใช้ฟังก์ชันนี้ไม่ปลอดภัยสำหรับแฟ้มข้อมูลแบบไบนารี

ตัวอย่าง การใช้ฟังก์ชัน file()

```

1.  <?php
2.  $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.  $lines = file("$DOCUMENT_ROOT/../data/data2.txt");
4.  if($lines == FALSE){
5.      echo "Cannot open file \"data2.txt\" at this time, please try again later...";
6.      exit;
7.  }else{
8.      foreach($lines as $value){
9.          echo "$value<br>";
10.     }
11. }
12. ?>
    
```



รูปที่ 6.16: การใช้ฟังก์ชัน file()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ฟังก์ชัน file() ในการอ่านข้อมูลทั้งหมดในแฟ้มข้อมูล data2.txt แล้วส่งข้อมูลเหล่านั้นเป็นแบบอะเรย์ไปเก็บไว้ที่ตัวแปร \$lines โดยข้อมูลหนึ่ง ๆ ในอะเรย์ก็คือข้อมูลทั้งหมดที่อยู่ในบรรทัดหนึ่ง ๆ ในแฟ้มข้อมูล

บรรทัดที่ 4 ทดสอบดูว่าการทำงานของฟังก์ชัน file() นั้นทำงานได้ถูกต้องหรือไม่ ถ้าทำงานได้ถูกต้อง นั่นคืออ่านได้สำเร็จ ก็จะกระโดดไปบรรทัดที่ 8 ซึ่งเป็นส่วนของคำสั่ง else แต่ถ้าไม่สำเร็จก็จะเข้าไปทำบรรทัดที่ 5 และ 6 ตามลำดับ ซึ่งสามารถอ่านแฟ้มข้อมูลได้สำเร็จเนื่องจากมีแฟ้มข้อมูลดังที่ระบุ จึงกระโดดไปบรรทัดที่ 8

บรรทัดที่ 8 ใช้ฟังก์ชัน foreach() ในการวนลูปให้ค่าที่อยู่ในตัวแปร \$lines ไปเก็บไว้ที่ตัวแปร \$value

บรรทัดที่ 9 แสดงข้อมูลที่ถูกเก็บอยู่ในตัวแปร \$value ออกที่จอภาพ

4. ใช้ฟังก์ชัน file_get_contents() ซึ่งมีรูปแบบของการทำงานดังต่อไปนี้

```

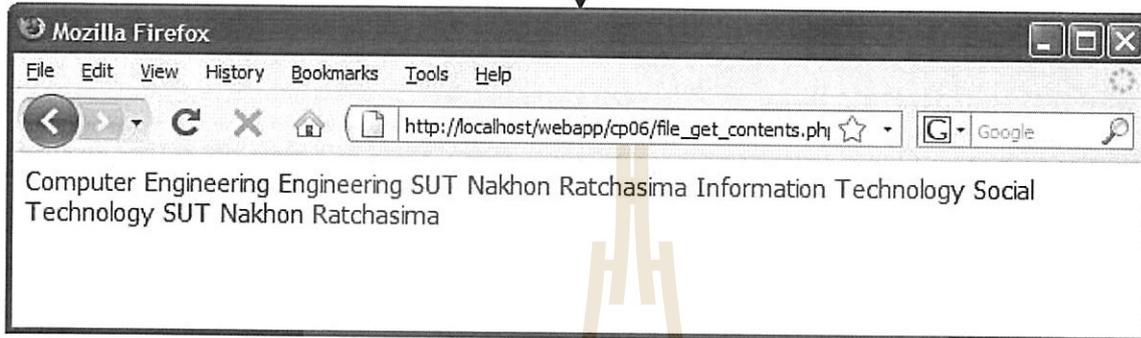
string file_get_contents (string filename [,bool
use_include_path [,resource context]])
    
```

ฟังก์ชัน file_get_contents() นี้จะมีการทำงานเหมือนกันกับฟังก์ชัน file() แต่สิ่งที่แตกต่างกันคือฟังก์ชัน file_get_contents() จะส่งค่าทั้งหมดกลับเป็นสตริง

ตัวอย่าง การใช้ฟังก์ชัน file_get_contents()

```

1.  <?php
2.  $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.  $str = file_get_contents("$DOCUMENT_ROOT/../data/data2.txt");
4.  if($str == FALSE){
5.      echo "Cannot open file \"data2.txt\" at this time, please try again later...";
6.      exit;
7.  }else{
8.      echo "$str<br>";
9.  }
10. ?>
    
```



รูปที่ 6.17: การใช้ฟังก์ชัน file_get_contents()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่อง การตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปร ระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ฟังก์ชัน file_get_contents() ในการอ่านข้อมูลทั้งหมดในแฟ้มข้อมูล data2.txt แล้วส่งข้อมูลสตริงเหล่านั้นไปเก็บไว้ที่ตัวแปร \$str

บรรทัดที่ 4 ทดสอบดูว่าการทำงานของฟังก์ชัน file_get_contents() นั้นทำงานได้ถูกต้อง หรือไม่ ถ้าทำงานได้ถูกต้อง นั่นคืออ่านได้สำเร็จ ก็จะกระโดดไปบรรทัดที่ 8 ซึ่งเป็นส่วนของคำสั่ง else แต่ถ้าไม่สำเร็จก็จะเข้าไปทำบรรทัดที่ 5 และ 6 ตามลำดับ ซึ่งสามารถอ่านแฟ้มข้อมูลได้สำเร็จเนื่องจากมี แฟ้มข้อมูลดังที่ระบุ จึงกระโดดไปบรรทัดที่ 8

บรรทัดที่ 8 แสดงข้อมูลที่ถูเก็บอยู่ในตัวแปร \$str ออกที่จอภาพ

5. ใช้ฟังก์ชัน fread() ซึ่งมีรูปแบบของการทำงานดังต่อไปนี้

string fread (resource handle, int length)

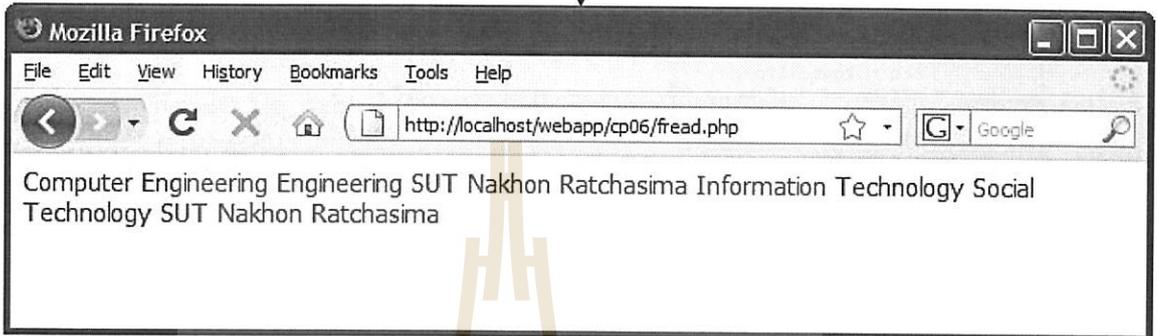
ฟังก์ชัน fread() นี้จะทำการอ่านข้อมูลจากแฟ้มข้อมูลที่ตัวชี้แฟ้มข้อมูลซึ่งอยู่เป็นจำนวนไบต์ที่ระบุ จะหยุดอ่านข้อมูลก็ต่อเมื่อตรงกับประเด็นใดประเด็นหนึ่งดังต่อไปนี้คือ เมื่ออ่านครบจำนวนไบต์ที่ระบุแล้ว หรืออ่านจนจบข้อมูลแล้ว หรืออ่านจนครบจำนวน 8192 ไบต์แล้ว ฟังก์ชัน fread() จะส่งค่าทั้งหมดกลับ เป็นสตริง

ตัวอย่าง การใช้ฟังก์ชัน fread()

```

1. <?php
2.   $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.   @ $fp = fopen("$DOCUMENT_ROOT/../data/data2.txt", "rt");
4.   if (!$fp) {
5.     echo 'Could not open file "data2.txt"<br>';
6.     exit;
7.   }
8.   $str = fread($fp, 999);
9.   echo "$str<br>";
10.  fclose($fp);
11. ?>

```



รูปที่ 6.18: การใช้ฟังก์ชัน fread()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่องการตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปรระบบเก็บค่าที่เป็นต้นทางของแฟ้มข้อมูล

บรรทัดที่ 3 ใช้ตัวดำเนินการควบคุมข้อผิดพลาด @ นำหน้าตัวแปร \$fp ที่เป็นตัวชี้แฟ้มไปยังแฟ้มข้อมูลชื่อว่า data2.txt ในโฟลเดอร์ data ซึ่งอยู่ที่ต้นทางของแฟ้มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ เพื่ออ่านเท่านั้น

บรรทัดที่ 4 ทดสอบดูว่าแฟ้มข้อมูลที่เราจะเปิดได้ถูกเปิดสำเร็จหรือไม่ ถ้าเปิดได้สำเร็จจะกระโดดไปบรรทัดที่ 8 แต่ถ้าไม่สำเร็จจะเข้าไปทำบรรทัดที่ 5 และ 6 ตามลำดับ ซึ่งสามารถเปิดแฟ้มข้อมูลได้สำเร็จเนื่องจากมีแฟ้มข้อมูลดังที่ระบุ จึงกระโดดไปบรรทัดที่ 8

บรรทัดที่ 8 ใช้ฟังก์ชัน fread() ในการอ่านข้อมูลจากแฟ้มข้อมูลที่ตัวชี้แฟ้มข้อมูล \$fp ชี้อยู่ ซึ่งคือแฟ้ม data2.txt นั่นเอง จะหยุดอ่านข้อมูลก็ต่อเมื่อ ได้อ่านเข้ามาเป็นจำนวน 999 ไบต์ หรือจนจบข้อมูล ทั้งนี้ขึ้นอยู่กับจำนวนของข้อมูล แล้วเอาข้อมูลที่อ่านเข้ามานั้นไปเก็บไว้ที่ตัวแปร \$str เป็นสตริง

บรรทัดที่ 9 แสดงข้อมูลที่ถูกเก็บอยู่ในตัวแปร \$str ออกที่จอภาพ แล้วเลื่อนตำแหน่ง Cursor ไปยังตำแหน่งแรกของบรรทัดใหม่

บรรทัดที่ 10 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม \$fp ชี้อยู่ ซึ่งคือแฟ้มข้อมูลชื่อว่า data2.txt นั่นเอง

จากตัวอย่างการใช้ฟังก์ชันต่าง ๆ ในการอ่านแฟ้มข้อมูลเข้ามานั้น จะสังเกตเห็นว่า ในการอ่านนั้น จะอ่านข้อมูลทั้งหมดเข้ามาทีเดียวเลย แต่ในบางครั้งอาจต้องการอ่านข้อมูลเข้ามาทีละ 1 ตัวอักษร เพื่อประมวลผลในรูปแบบต่าง ๆ ที่ต้องการ โดยใช้ฟังก์ชัน fgets() ซึ่งมีรูปแบบการทำงานดังนี้

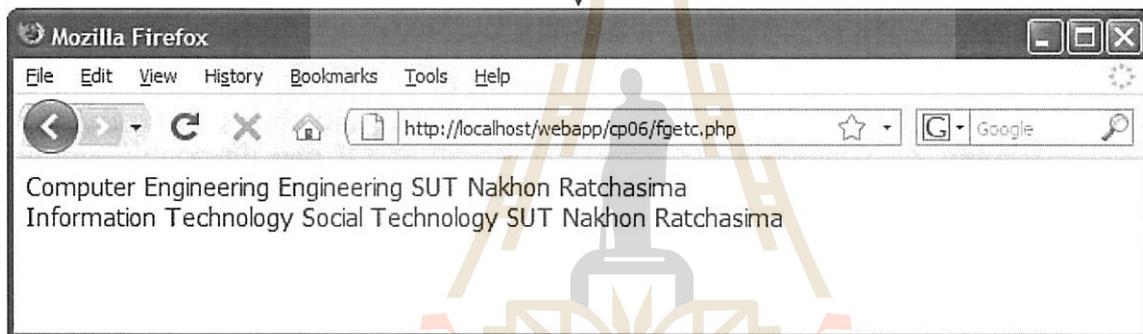
```
string fgets(resource handle)
```

ฟังก์ชัน fgetc() นี้จะส่งค่ากลับเป็น สตริง ที่ประกอบไปด้วยตัวอักษร 1 ตัว จะส่งค่ากลับเป็นเท็จก็ต่อเมื่อสิ้นสุดเพิ่มข้อมูล ซึ่งโดยปกติแล้วมันจะไม่นิยามอ่านข้อมูลจากเพิ่มข้อมูลเข้ามาทีละ 1 ตัวอักษร ยกเว้นในบางกรณีที่ต้องการทดสอบหรือประมวลผลตัวอักษรทุก ๆ ตัวในเพิ่มข้อมูลนั้น ๆ

ตัวอย่าง การใช้ฟังก์ชัน fgetc()

```

1.  <?php
2.  $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3.  @ $fp = fopen("$DOCUMENT_ROOT/./data/data2.txt", "rt");
4.  if(!$fp){
5.      echo 'Could not open file "data2.txt"<br>';
6.      exit;
7.  }
8.  while(!feof($fp)){
9.      $ch = fgetc($fp);
10.     if($ch == "\n")
11.         echo '<br>';
12.     else
13.         echo $ch;
14.     }
15.     fclose($fp);
16.  ?>
    
```



รูปที่ 6.19: การใช้ฟังก์ชัน fgetc()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่อง การตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปร ระบบเก็บค่าที่เป็นต้นทางของเพิ่มข้อมูล

บรรทัดที่ 3 ใช้ตัวดำเนินการควบคุมข้อผิดพลาด @ นำหน้าตัวแปร \$fp ที่เป็นตัวชี้เพิ่มไปยัง เพิ่มข้อมูลชื่อว่า data2.txt ในโฟลเดอร์ data ซึ่งอยู่ที่ต้นทางของเพิ่มข้อมูลหลัก ขึ้นมาเป็นแบบเท็กซ์ เพื่อ อ่านเท่านั้น

บรรทัดที่ 4 ทดสอบดูว่าเพิ่มข้อมูลที่เราจะเปิดได้ถูกเปิดสำเร็จหรือไม่ ถ้าเปิดได้สำเร็จจะกระโดด ไปบรรทัดที่ 8 แต่ถ้าไม่สำเร็จจะเข้าไปทำบรรทัดที่ 5 และ 6 ตามลำดับ ซึ่งสามารถเปิดเพิ่มข้อมูลได้สำเร็จ เนื่องจากมีเพิ่มข้อมูลดังที่ระบุ จึงกระโดดไปบรรทัดที่ 8

บรรทัดที่ 8 ใช้ลูป while ในการทดสอบว่าตัวชี้เพิ่มข้อมูลไม่ได้ชี้อยู่ที่ตำแหน่งสุดท้ายของ ข้อมูลจริงหรือไม่ ถ้าจริงจะเข้าไปทำคำสั่งในบรรทัดที่ 9 ถึง 13 แต่ถ้าไม่ใช่จะกระโดดไปทำคำสั่งในบรรทัดที่ 15 ในกรณีนี้คือ **จริง** ดังนั้นจึงไปทำคำสั่งในบรรทัดที่ 9

บรรทัดที่ 9 ใช้ฟังก์ชัน fgetc() รับตัวอักษร 1 ตัวจากเพิ่มข้อมูล แล้วเอาไปเก็บไว้ที่ตัวแปร \$ch

บรรทัดที่ 10 ทดสอบดูว่าตัวอักษรที่อยู่ในตัวแปร \$ch มีค่าเท่ากับตัวอักษร '\n' หรือไม่ ถ้าใช่ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 11 ซึ่งเป็นคำสั่งที่เลื่อนตำแหน่ง Cursor ไปยังตำแหน่งแรกของบรรทัดใหม่ มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 13 ซึ่งเป็นการแสดงตัวอักษรที่อยู่ในตัวแปร \$ch ออกที่จอภาพ

บรรทัดที่ 15 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม \$fp ซ้ำอยู่ ซึ่งคือแฟ้มข้อมูลชื่อว่า data2.txt นั่นเอง

6.1.4 ฟังก์ชันที่เอื้ออำนวยความสะดวกในการจัดการแฟ้มข้อมูล

ฟังก์ชันที่เอื้ออำนวยความสะดวกในการจัดการกับแฟ้มข้อมูลมีอยู่หลายฟังก์ชันด้วยกัน แต่ในที่นี้จะขอยกตัวอย่างของฟังก์ชันที่พบเห็นว่ามีการใช้งานอยู่บ่อย ๆ เช่น

- ฟังก์ชัน `file_exists()` ใช้ทดสอบว่ามีแฟ้มข้อมูลนั้นอยู่หรือไม่ โดยไม่ได้เปิดแฟ้มข้อมูลนั้น มีรูปแบบการทำงานดังนี้

```
bool file_exists (string filename)
```

- ฟังก์ชัน `filesize()` ใช้หาขนาดของแฟ้มข้อมูล ค่าที่ส่งกลับจะมีหน่วยเป็นไบต์ มีรูปแบบการทำงานดังนี้

```
int filesize (string filename)
```

- ฟังก์ชัน `unlink()` ใช้เมื่อต้องการลบแฟ้มข้อมูล ค่าที่ส่งกลับจะเป็น true หรือ false มีรูปแบบการทำงานดังนี้

```
bool unlink (string filename [,resource context])
```

- ฟังก์ชัน `rewind()` ใช้เมื่อต้องการเลื่อนตำแหน่งตัวชี้แฟ้มข้อมูลกลับไปยังตำแหน่งเริ่มต้นของข้อมูล ทั้งนี้ต้องมีตัวชี้แฟ้มข้อมูลชี้ไปที่แฟ้มข้อมูลและแฟ้มข้อมูลนั้นได้ถูกเปิดไว้แล้ว ค่าที่ส่งกลับจะเป็น true หรือ false มีรูปแบบการทำงานดังนี้

```
bool rewind (resource handle)
```

- ฟังก์ชัน `ftell()` ใช้เมื่อต้องการระบุตำแหน่งตัวชี้แฟ้มข้อมูลว่าอยู่ที่ตำแหน่งใด ทั้งนี้ต้องมีตัวชี้แฟ้มข้อมูลชี้ไปที่แฟ้มข้อมูลและแฟ้มข้อมูลนั้นได้ถูกเปิดไว้แล้ว ค่าที่ส่งกลับจะเป็นตำแหน่งไบต์ที่ชี้อยู่ ถ้าเกิดข้อผิดพลาดจะส่งค่ากลับเป็น false มีรูปแบบการทำงานดังนี้

```
int ftell (resource handle)
```

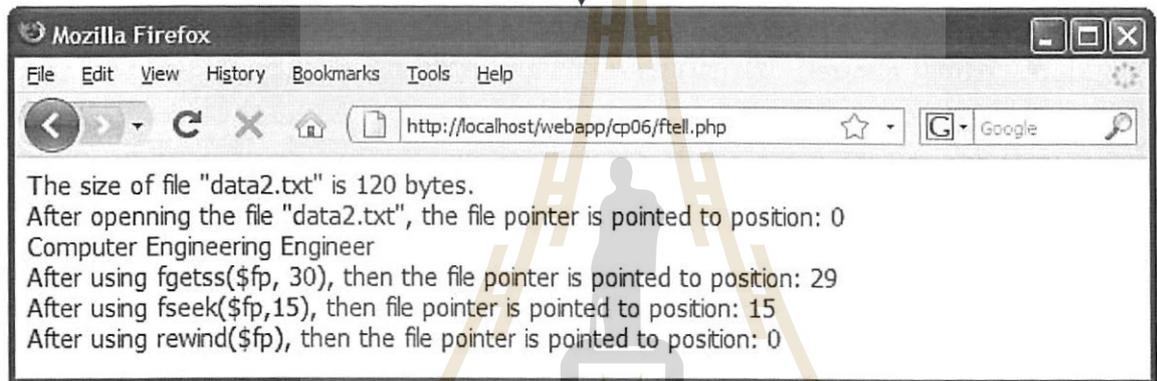
- ฟังก์ชัน `fseek()` ใช้เมื่อต้องการระบุตำแหน่งให้ตัวชี้แฟ้มข้อมูลชี้ไปที่ตำแหน่งนั้น ๆ ทั้งนี้ต้องมีตัวชี้แฟ้มข้อมูลชี้ไปที่แฟ้มข้อมูลและแฟ้มข้อมูลนั้นได้ถูกเปิดไว้แล้ว ค่าที่ส่งกลับจะเป็น 0 เมื่อสามารถชี้ไปที่ตำแหน่งนั้นได้ มิฉะนั้นแล้วจะส่งค่ากลับเป็น -1 แทน มีรูปแบบการทำงานดังนี้

```
int fseek (resource handle, int offset [, int whence])
```

ตัวอย่าง การใช้ฟังก์ชัน file_exists(), filesize(), ftell(), fseek() และ rewind()

```

1. <?php
2. $DOCUMENT_ROOT = $HTTP_SERVER_VARS["DOCUMENT_ROOT"];
3. $filename = "$DOCUMENT_ROOT/./data/data2.txt";
4. if(file_exists("$filename")){
5.     $size = filesize("$filename");
6.     echo 'The size of file "data2.txt" is '.$size.' bytes.<br>';
7.     $fp = fopen("$filename","rt");
8.     echo 'After opening the file "data2.txt", the file pointer is pointed to position:
9.     '.ftell($fp).'<br>';
10.    $output= fgets($fp, 30);
11.    echo $output.<br>';
12.    echo 'After using fgets($fp, 30), then the file pointer is pointed to position:
13.    '.ftell($fp).'<br>';
14.    fseek($fp,15);
15.    echo 'After using fseek($fp,15), then file pointer is pointed to position:
16.    '.ftell($fp).'<br>';
17.    rewind($fp);
18.    echo 'After using rewind($fp), then the file pointer is pointed to position:
19.    '.ftell($fp).'<br>';
20.    fclose($fp);
21. }
22. ?>
    
```



รูปที่ 6.20: การใช้ฟังก์ชัน file_exists(), filesize(), ftell(), fseek() และ rewind()

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$DOCUMENT_ROOT ก่อน เนื่องจากไม่ต้องห่วงเรื่อง การตั้งค่าของ register_globals ว่าเป็น on หรือ off โดยตัวแปร \$DOCUMENT_ROOT เป็นตัวแปร ระบบเก็บค่าที่เป็นต้นทางของเพิ่มข้อมูล

บรรทัดที่ 3 กำหนดให้ตัวแปร \$filename ให้เก็บสตริงที่เส้นทางเพิ่มข้อมูล data2.txt

บรรทัดที่ 4 ทดสอบว่ามีเพิ่มข้อมูล data2.txt อยู่หรือไม่ ถ้ามีอยู่ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 5 ถึง 15 ในกรณีนี้มีเพิ่มข้อมูล data2.txt จริง ดังนั้นจึงเข้าไปเริ่มทำคำสั่งในบรรทัดที่ 5

บรรทัดที่ 5 หาขนาดของเพิ่มข้อมูล data2.txt โดยใช้ฟังก์ชัน filesize() แล้วเก็บผลลัพธ์ซึ่งมี หน่วยเป็นไบต์ไว้ที่ตัวแปร \$size

บรรทัดที่ 6 แสดงข้อมูลที่ถูกเก็บไว้ในตัวแปร \$size ออกที่จอภาพ แล้วเลื่อนตำแหน่ง Cursor ไปที่จุดเริ่มต้นของบรรทัดใหม่

บรรทัดที่ 7 เปิดเพิ่มข้อมูลชื่อว่า data2.txt ขึ้นมาแบบเท็กซ์เพื่ออ่านเท่านั้น โดยกำหนดตัวแปร \$fp เป็นตัวชี้ไปที่เพิ่มข้อมูลนั้น

บรรทัดที่ 8 แสดงตำแหน่งของตัวชี้เพิ่มข้อมูลว่าขึ้นอยู่กับ ตำแหน่งไบต์ที่เท่าไร ในกรณีนี้คือ ตำแหน่งเริ่มต้นคือ 0 นั่นเอง แล้วเลื่อนตำแหน่ง Cursor ไปที่จุดเริ่มต้นของบรรทัดใหม่

บรรทัดที่ 9 กำหนดตัวแปร \$output ให้เก็บสตริงที่ใช้ฟังก์ชัน fgets() รับข้อมูลที่ตัวชี้แฟ้มข้อมูล \$fp ซึ่งอยู่เป็นจำนวน 30 ไบต์ ซึ่งค่าที่ส่งกลับคือ จำนวนไบต์ - 1 แล้วเลื่อนตำแหน่ง Cursor ไปที่จุดเริ่มต้นของบรรทัดใหม่

บรรทัดที่ 10 แสดงข้อมูลที่อยู่ในตัวแปร \$output ซึ่งคือสตริงที่มีความยาว 29 ไบต์ นั่นคือข้อความ Computer Engineering Engineer ซึ่งรวมทั้งหมดมี 29 ตัวอักษร

บรรทัดที่ 11 แสดงตำแหน่งของตัวชี้แฟ้มข้อมูลว่าอยู่ที่ ณ ตำแหน่งไบต์ที่เท่าไร ในกรณีนี้คือตำแหน่งที่ 29 แล้วเลื่อนตำแหน่ง Cursor ไปที่จุดเริ่มต้นของบรรทัดใหม่

บรรทัดที่ 12 ใช้คำสั่ง fseek() โดยระบุตำแหน่งที่ต้องการให้ตัวชี้แฟ้มข้อมูลชี้ไปที่ไบต์ที่ 15

บรรทัดที่ 13 แสดงตำแหน่งของตัวชี้แฟ้มข้อมูลว่าอยู่ที่ ณ ตำแหน่งไบต์ที่เท่าไร ในกรณีนี้คือตำแหน่งที่ 15 แล้วเลื่อนตำแหน่ง Cursor ไปที่จุดเริ่มต้นของบรรทัดใหม่

บรรทัดที่ 14 ใช้คำสั่ง rewind() ทำให้ตัวชี้แฟ้มข้อมูลกลับไปชี้อยู่ที่ตำแหน่งแรกคือ 0 นั่นเอง

บรรทัดที่ 15 แสดงตำแหน่งของตัวชี้แฟ้มข้อมูลว่าอยู่ที่ ณ ตำแหน่งไบต์ที่เท่าไร ในกรณีนี้คือตำแหน่งแรกคือ 0 นั่นเอง

บรรทัดที่ 17 ปิดแฟ้มข้อมูลที่ตัวชี้แฟ้ม \$fp ซึ่งก็คือแฟ้มข้อมูลชื่อว่า data2.txt นั่นเอง

6.1.5 ปัญหาในการจัดการกับแฟ้มข้อมูล

แฟ้มข้อมูลแบบเท็กซ์ (Text File) นี้อาจก่อให้เกิดปัญหาและความยุ่งยากต่าง ๆ ในการจัดการซึ่งพอสรุปได้ดังต่อไปนี้

1. เมื่อแฟ้มข้อมูลนั้น ๆ มีขนาดใหญ่ อาจใช้เวลาในการประมวลผลเพิ่มขึ้น
2. การค้นหาข้อมูลในแฟ้มข้อมูลจะเกิดความยุ่งยาก เนื่องจากต้องทำการค้นหาในทุก ๆ ตำแหน่งที่มีการเก็บข้อมูล
3. ถ้ามีผู้ใช้แฟ้มข้อมูลหลายคนในเวลาเดียวกัน อาจเกิดความล่าช้าจนทำให้ผู้ใช้บางคนไม่ได้เข้าใช้แฟ้มข้อมูลนั้น ๆ อาจเนื่องมาจากมีเวลาจำกัดเป็นต้น
4. เนื่องจากการจัดการแฟ้มข้อมูลเป็นไปในรูปแบบเรียงลำดับ จากต้นข้อมูลไปเรื่อย ๆ จนถึงข้อมูลตัวสุดท้าย ถ้าต้องการลบข้อมูลหนึ่งทิ้ง หรือต้องการเพิ่มข้อมูลหนึ่งลงในส่วนตรงกลางของแฟ้มข้อมูล จะเกิดความยุ่งยากและใช้เวลาในการจัดการมาก เนื่องจากต้องทำการอ่านแฟ้มข้อมูลทั้งหมดเข้ามาเก็บไว้ในหน่วยความจำชั่วคราวก่อน หลังจากนั้นจึงทำการเปลี่ยนแปลงข้อมูล แล้วจึงเขียนข้อมูลที่มีการเปลี่ยนแปลงแล้วลงแฟ้มข้อมูลอีกครั้ง ถ้าเป็นแฟ้มข้อมูลที่มีขนาดใหญ่ จะทำให้เสียเวลาในการจัดการมากเพิ่มขึ้น

อย่างไรก็ดี ถ้าแฟ้มข้อมูลมีขนาดเล็ก ไม่ต้องมีการเปลี่ยนแปลง หรือปรับปรุงข้อมูลในแฟ้มข้อมูลบ่อย ๆ แฟ้มข้อมูลแบบเท็กซ์ก็เป็นทางเลือกที่ดี ซึ่งสะดวกและรวดเร็วในการจัดการ แต่ถ้าแฟ้มข้อมูลมีขนาดใหญ่ มีการเปลี่ยนแปลงข้อมูลบ่อยครั้ง จำเป็นต้องหาวิธีการจัดการที่เหมาะสมและดีกว่า นั่นคือระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ ที่จะกล่าวในบทต่อไป

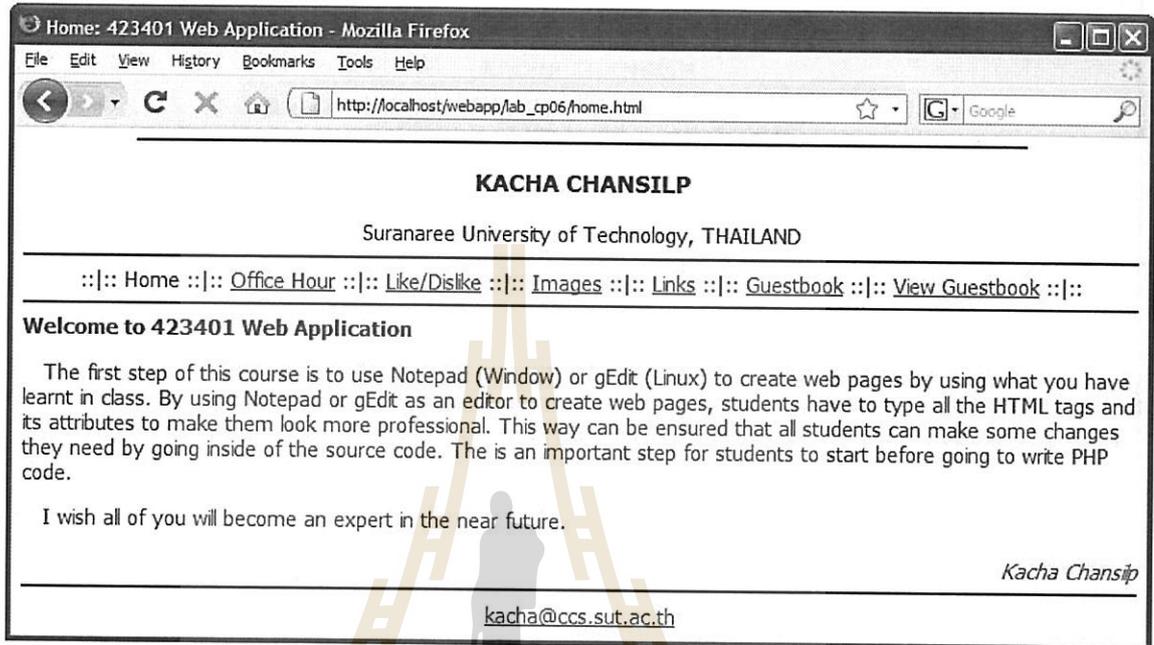
สรุป

การใช้ฟังก์ชันต่าง ๆ ในการจัดการกับข้อมูลที่ได้รับจากแบบฟอร์มที่ผู้ใช้ป้อนเข้ามานั้น มีหลายวิธีและหลายรูปแบบ ทั้งนี้ขึ้นอยู่กับวัตถุประสงค์ที่ต้องการ โดยมีขั้นตอนการทำงาน 3 ขั้นตอนคือ เปิดแฟ้มข้อมูล ประมวลผล และปิดแฟ้มข้อมูล โดยในบทเรียนนี้ได้เสนอเฉพาะการเขียนข้อมูลลงแฟ้มข้อมูล (Text File) และการอ่านข้อมูลจากแฟ้มข้อมูลเพื่อนำเสนอในรูปแบบต่าง ๆ ซึ่งเป็นทางเลือกทางหนึ่งในการจัดเก็บข้อมูล



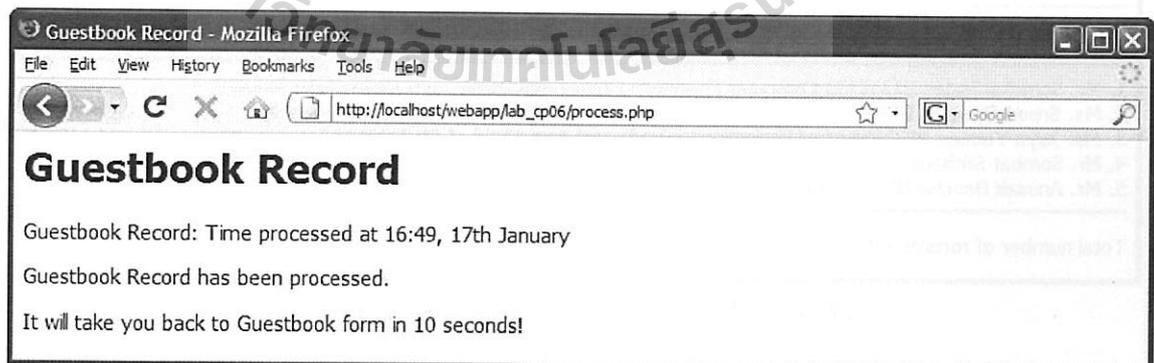
แบบฝึกหัด

1. ปรับปรุงไฟล์ html ทุก ๆ ไฟล์ในปฏิบัติการครั้งที่ 2 (home.html, office_hour.html, like_dislike.html, images.html, links.html และ guestbook.html) โดยเพิ่มจุดเชื่อมต่ออีก 1 จุดชื่อว่า "View Guestbook" ต่อท้าย ดังแสดงในรูปที่ 6.21

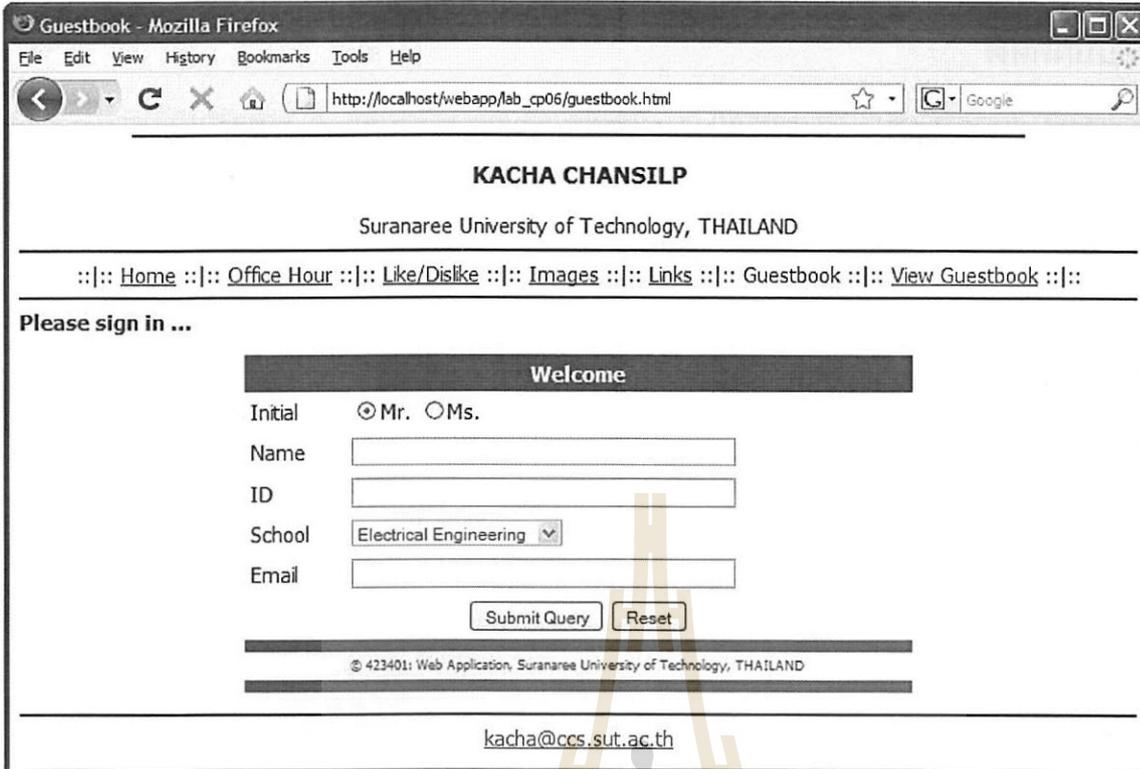


รูปที่ 6.21: เพิ่มจุดเชื่อม View Guestbook

2. จากปฏิบัติการครั้งที่ 2 เมื่อผู้ใช้กดปุ่ม Submit Query จะเรียกไฟล์ "thanks.html" ซึ่งจะแสดงข้อความว่า "THANKS FOR YOUR SIGNING IN" นั้น ให้เปลี่ยนคำสั่งในฟอร์มจาก action="thanks.html" เป็น action="process.php" ซึ่งไฟล์ process.php จะทำการเขียนข้อมูลที่ผู้ใช้ป้อนเข้ามาในฟอร์มลงที่ไฟล์ gbook.txt แล้วแสดงข้อความดังรูปที่ 6.22 ซึ่งจะแสดงอยู่ประมาณ 10 วินาที แล้วจะกลับไปสู่หน้าต่างของ guestbook.html ดังแสดงในรูปที่ 6.23

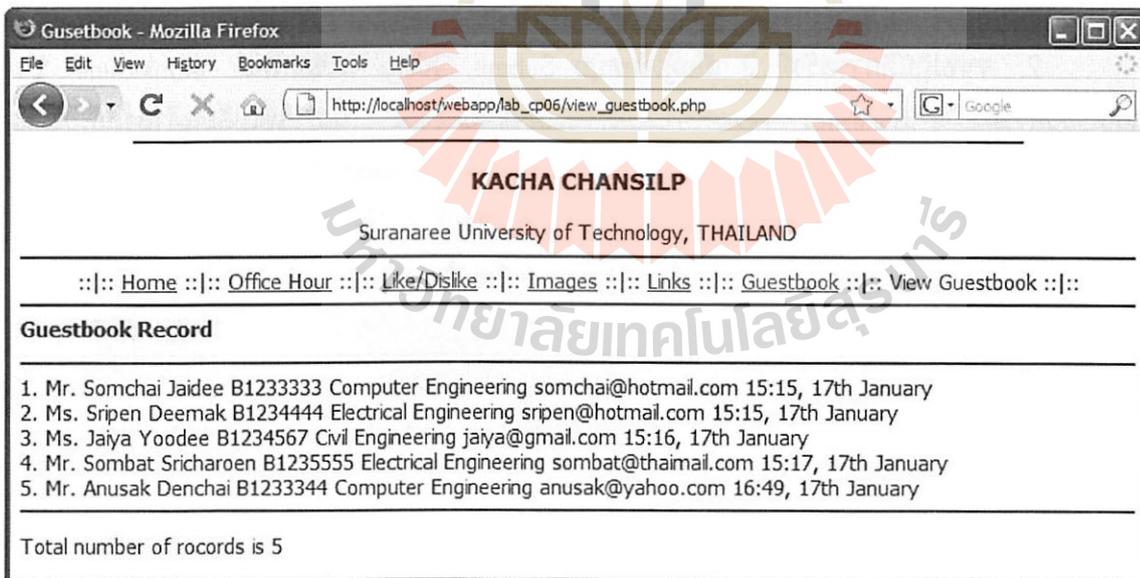


รูปที่ 6.22: การแสดงข้อความของไฟล์ process.php



รูปที่ 6.23: หน้าต่างของไฟล์ guestbook.html

3. เมื่อผู้ใช้กดปุ่ม “View Guestbook” ก็จะเรียกไฟล์ “view_guestbook.php” ซึ่งจะทำการเรียกข้อมูลจากไฟล์ gbook.txt แล้วแสดงข้อมูลที่มีอยู่ใน gbook.txt ออก ดังแสดงในรูปที่ 6.24



รูปที่ 6.24: การดึงข้อมูลจากเพิ่มข้อมูลเพื่อแสดงผลของไฟล์ view_guestbook.php

4. ปรับแต่งไฟล์ “view_guestbook.php” แล้วบันทึกเป็น “view_guestbook_table.php” โดยเพิ่มการนำเสนอข้อมูลให้อยู่ในตาราง ดังแสดงในรูปที่ 6.25 ดังนั้นเมื่อผู้ใช้งานปุ่ม “View Guestbook” ก็จะเรียกไฟล์ “view_guestbook_table.php”

KACHA CHANSILP
Suranaree University of Technology, THAILAND

[Home](#) [Office Hour](#) [Like/Dislike](#) [Images](#) [Links](#) [Guestbook](#) [View Guestbook](#)

Guestbook Record

Record #1: Mr. Somchai Jaidee B1233333 Computer Engineering somchai@hotmail.com 15:15, 17th January
 Record #2: Ms. Sripen Deemak B1234444 Electrical Engineering sripen@hotmail.com 15:15, 17th January
 Record #3: Ms. Jaiya Yoodee B1234567 Civil Engineering jaiya@gmail.com 15:16, 17th January
 Record #4: Mr. Sombat Sricharoen B1235555 Electrical Engineering sombat@thaimail.com 15:17, 17th January
 Record #5: Mr. Anusak Denchai B1233344 Computer Engineering anusak@yahoo.com 16:49, 17th January

Total number of student is 5

Putting Guestbook Records into a Table

NO	Initial	NAME	ID	SCHOOL	EMAIL	TIME
1	Mr.	Somchai Jaidee	B1233333	Computer Engineering	somchai@hotmail.com	15:15, 17th January
2	Ms.	Sripen Deemak	B1234444	Electrical Engineering	sripen@hotmail.com	15:15, 17th January
3	Ms.	Jaiya Yoodee	B1234567	Civil Engineering	jaiya@gmail.com	15:16, 17th January
4	Mr.	Sombat Sricharoen	B1235555	Electrical Engineering	sombat@thaimail.com	15:17, 17th January
5	Mr.	Anusak Denchai	B1233344	Computer Engineering	anusak@yahoo.com	16:49, 17th January

รูปที่ 6.25: การดึงข้อมูลจากเพิ่มข้อมูลเพื่อแสดงผลของไฟล์ view_guestbook_table.php



บทที่ 7 การรวมเพิ่มข้อมูลและฟังก์ชันวาดรูป <Including Files and Drawing Functions>

บทนำ

เว็บไซต์หนึ่ง ๆ นั้นประกอบไปด้วยแฟ้มสกุลต่าง ๆ เช่น สกุล .html, สกุล .php, สกุล .txt หรือ สกุล .jpg เป็นต้น ข้อมูลบางอย่างอาจจะมีอยู่ในแฟ้มทุกแฟ้มก็ได้เช่น จุดเชื่อมต่อที่จะส่งอีเมลไปถึงผู้ดูแลเว็บไซต์ หรือ สถานที่ติดต่อของผู้ดูแลเว็บไซต์ เป็นต้น เมื่อต้องการปรับปรุงแก้ไขข้อมูลนั้น ๆ ก็จะต้องเข้าไปจัดการกับแฟ้มข้อมูลในแต่ละแฟ้ม ถ้าสมมุติว่ามี 10 แฟ้มข้อมูล นั้นหมายความว่า จะต้องทำการปรับปรุงแก้ไขแฟ้มข้อมูลทั้ง 10 แฟ้ม วิธีการเช่นนี้ทำให้เกิดความยุ่งยากในการจัดการกับผู้ดูแลเว็บไซต์ ซึ่งในภาษา PHP ได้มีฟังก์ชันที่เอื้ออำนวยความสะดวกในการจัดการกับข้อมูลที่ซ้ำ ๆ กัน โดยแยกข้อมูลที่ซ้ำกันนั้นออกไปเก็บเป็นแฟ้มข้อมูลหนึ่ง แล้วแฟ้มข้อมูลต่าง ๆ ที่ต้องการข้อมูลนั้น ก็สามารถทำได้โดยการดึงข้อมูลจากแฟ้มข้อมูลนั้นเข้าไปได้เลย โดยใช้คำสั่งพิเศษเพียงบรรทัดเดียว (ซึ่งจะกล่าวถึงในตอนต่อไป) เมื่อมีการปรับปรุงแก้ไขในส่วนที่ซ้ำกัน ก็เพียงแคไปจัดการที่แฟ้มข้อมูลนั้นแฟ้มเดียว นอกจากนั้นแล้ว ในภาษา PHP ยังมีฟังก์ชันต่าง ๆ ที่ช่วยในการวาดรูป เช่น การวาดรูปสี่เหลี่ยมผืนผ้า สี่เหลี่ยมจัตุรัส วงกลม หรือวงรี รวมไปถึงการใส่สีต่าง ๆ ให้กับเส้นขอบและพื้นหลังของรูป

วัตถุประสงค์

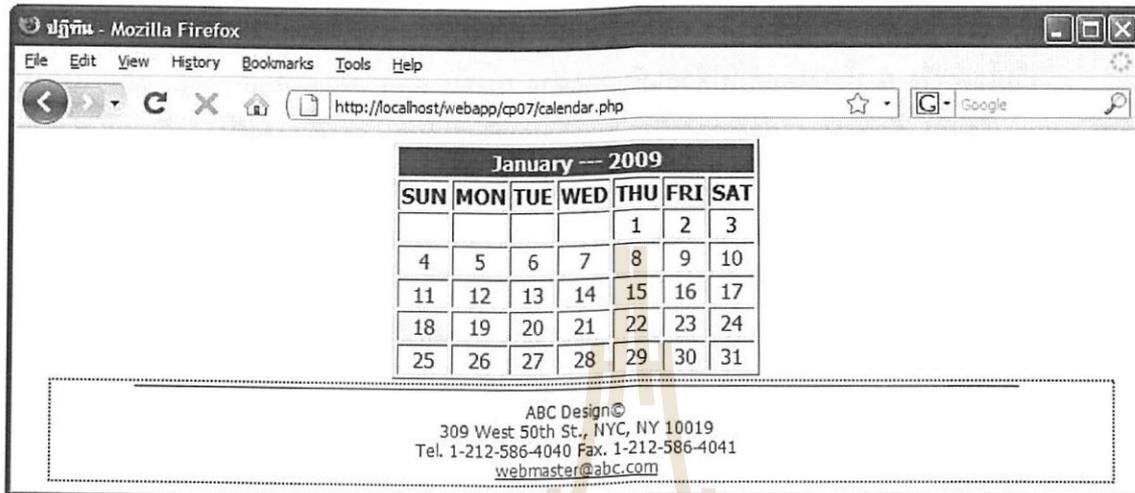
1. เพื่อศึกษารูปแบบการทำงานของคำสั่งและฟังก์ชันต่าง ๆ ในการจัดการเกี่ยวกับข้อมูลที่ซ้ำและการวาดรูป
2. เพื่อทดลองเรียกใช้ฟังก์ชันในการแทรกข้อมูลที่ซ้ำและการวาดรูป
3. เพื่อประยุกต์การใช้ฟังก์ชันในการแทรกข้อมูลที่ซ้ำและการวาดรูป

เนื้อหา

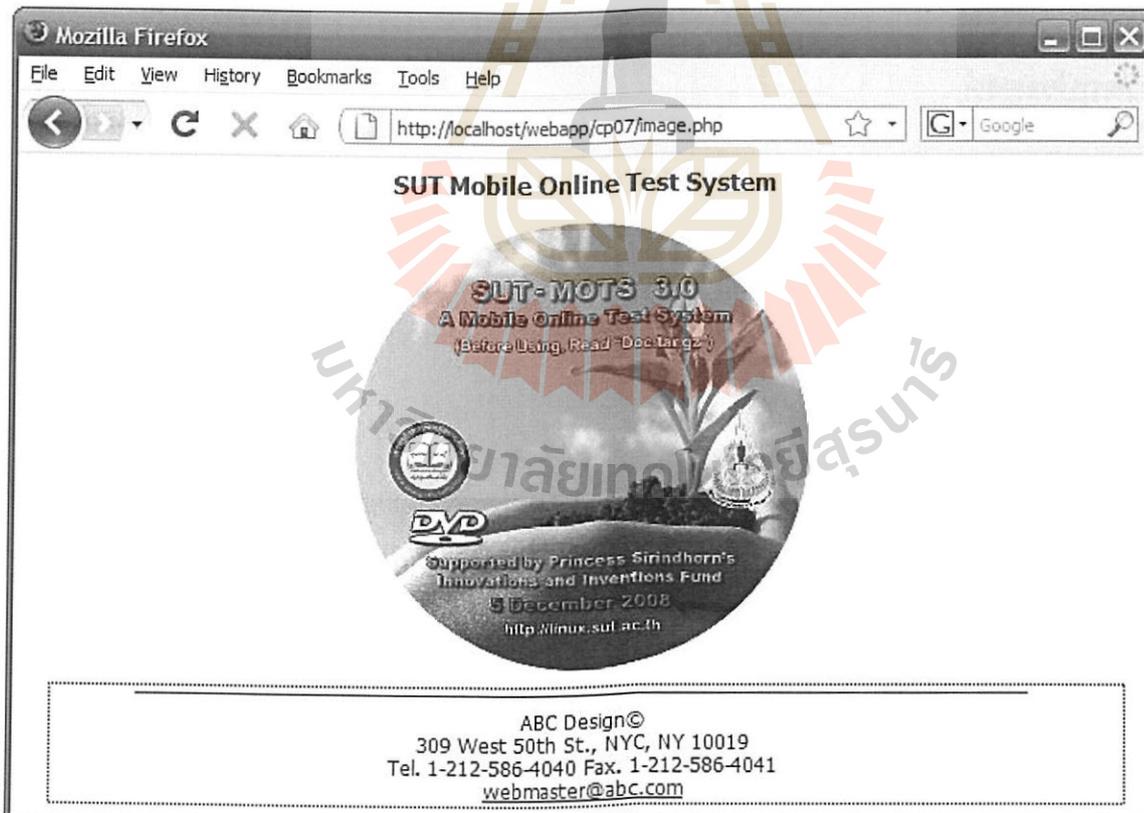
ในส่วนแรกของบทนี้ จะกล่าวถึงคำสั่งในการรวมเพิ่มข้อมูลเข้าด้วยกัน เพื่อนำเสนอเนื้อหาหรือข้อมูลบนเว็บ ถ้าเป็นข้อมูลทั่ว ๆ ไป จะเป็นแฟ้มข้อมูลที่เป็นสกุล .html แต่ถ้าข้อมูลเหล่านั้นมีการประมวลผล โดยอาจจะดึงข้อมูลมาจากแฟ้มข้อมูลอื่นแล้วนำมาประมวลผลหรือดึงข้อมูลมาจากระบบฐานข้อมูลเพื่อนำมาประมวลผลแล้วนำเสนอสิ่งที่ต้องการ แฟ้มข้อมูลเหล่านี้จะต้องเป็นแฟ้มข้อมูลที่สามารถติดต่อกับแฟ้มอื่น ๆ หรือติดต่อกับระบบฐานข้อมูลได้ เช่น แฟ้มข้อมูลที่เป็นสกุล .asp หรือสกุล .php เป็นต้น ในการนำเสนอเนื้อหาหรือข้อมูลเหล่านี้ มักจะประกอบไปด้วยส่วนของข้อมูลที่เว็บไซต์นั้น ๆ ต้องการให้มีเสมอ เช่น ข้อมูลที่อยู่ของเว็บไซต์ จุดเชื่อมต่อที่เป็นอีเมลของผู้ดูแลระบบ หรือตัวแปรที่แฟ้มข้อมูลต่าง ๆ ต้องการใช้ร่วมกัน เป็นต้น ดังนั้นส่วนของข้อมูลที่มีซ้ำในทุก ๆ หน้าของเว็บหรือค่าของตัวแปรที่แฟ้มข้อมูลต่าง ๆ ต้องการใช้ร่วมกันนั้น ควรแยกออกมาเก็บเป็นแฟ้มข้อมูลแฟ้มหนึ่ง (นิยมตั้งให้เป็นสกุล .inc) แล้วใช้คำสั่งเรียกแฟ้มข้อมูลนี้ไปแสดงในตำแหน่งที่ต้องการได้ และในส่วนที่สองจะกล่าวถึงฟังก์ชันต่าง ๆ ที่ช่วยในการจัดการในการวาดรูป กำหนดสีของเส้นขอบ และสีพื้นหลัง

7.1 คำสั่ง require() และ include() ในการรวมแฟ้มข้อมูล

สมมุติว่ามีแฟ้มข้อมูลอยู่ 2 แฟ้มชื่อว่า calendar.php และ image.php ซึ่งทั้ง 2 แฟ้มนั้นต่างก็มีข้อมูลที่เหมือนกันคือ ส่วนท้ายของการแสดงผล นั่นคือที่อยู่ของเว็บไซต์และจุดเชื่อมต่ออีเมลที่จะส่งถึงผู้ดูแลเว็บไซต์ ดังแสดงในภาพที่ 7.1 และ 7.2 ตามลำดับ



รูปที่ 7.1: การแสดงผลของแฟ้มข้อมูล calendar.php



รูปที่ 7.2: การแสดงผลของแฟ้มข้อมูล image.php

จากรูปที่ 2 นั้นจะเห็นว่า ส่วนที่เหมือนกัน (บริเวณที่มีเส้นประ) ซึ่งมีคำสั่งดังต่อไปนี้อยู่ในแฟ้ม calendar.php และ image.php คือ

```
echo '<center>':
echo '<hr width="80%"><br>';
echo '<font size="2">ABC Design&copy<br>';
echo '309 West 50th St., NYC, NY 10019<br>';
echo 'Tel. 1-212-586-4040 Fax. 1-212-586-4041<br>';
echo '<a href="mailto:webmaster@abc.com">webmaster@abc.com</a>';
echo '</font><br>';
echo '</center>';
```

ดังนั้นเราสามารถตัดคำสั่งเหล่านั้นออกจากแฟ้มทั้ง 2 แล้วนำไปเก็บไว้ที่แฟ้มข้อมูลหนึ่งชื่อว่า address.inc ส่วนแฟ้มข้อมูลทั้ง 2 แฟ้มนั้น ให้ใช้คำสั่งเรียกแฟ้มข้อมูล address.inc เข้าไปตรงตำแหน่งที่ตัดคำสั่งเหล่านั้นทิ้ง ซึ่งคำสั่งที่ใช้ในการเรียกแฟ้มข้อมูลเข้าที่นิยมใช้มีอยู่ 4 คำสั่งด้วยกันคือ

1. คำสั่ง require()
2. คำสั่ง require_once()
3. คำสั่ง include()
4. คำสั่ง include_once()

คำสั่ง require() และ include() นั้นมีลักษณะการทำงานที่คล้ายกันคือ ใช้เมื่อต้องการดึงแฟ้มข้อมูลเข้า แต่ที่แตกต่างกันคือการจัดการข้อผิดพลาดเกี่ยวกับแฟ้มข้อมูลที่ถูกดึงเข้ามา โดยคำสั่ง require() จะแสดงข้อผิดพลาด (Fatal Error) และหยุดการทำงานของสคริปต์ที่เหลืออยู่ที่ส่วนคำสั่ง include() นั้นจะแสดงคำเตือน (Warning) แต่ไม่หยุดการทำงานของสคริปต์ นั่นคือสคริปต์ที่เหลือก็ยังทำงานปกติ

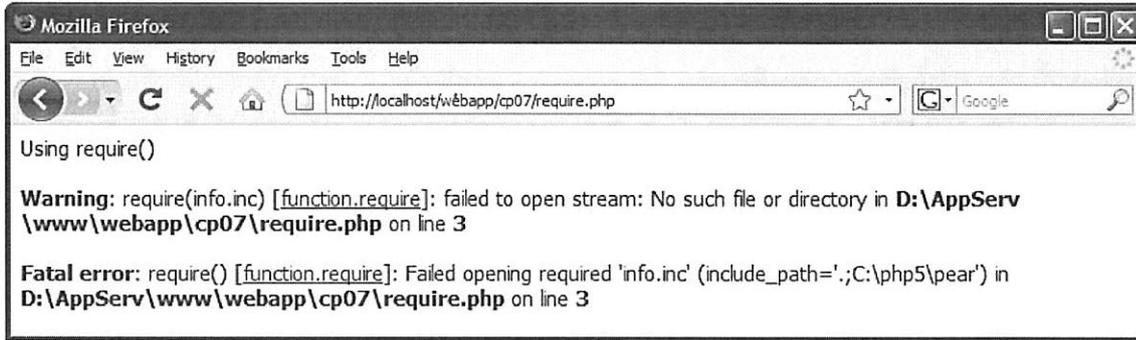
ตัวอย่าง แฟ้มข้อมูล info.inc

1.	<?php
2.	\$color = 'Red';
3.	\$text = 'Dragon';
4.	echo 'Coming Soon...';
5.	?>

รูปที่ 7.3: ข้อมูลในแฟ้ม info.inc

ตัวอย่าง การใช้คำสั่ง require() แต่ไม่มีแฟ้มข้อมูล info.inc

1.	<?php
2.	echo "Using require() ";
3.	require("info.inc");
4.	echo "The \$color \$text !!!";
5.	echo " Bye Bye";
6.	?>



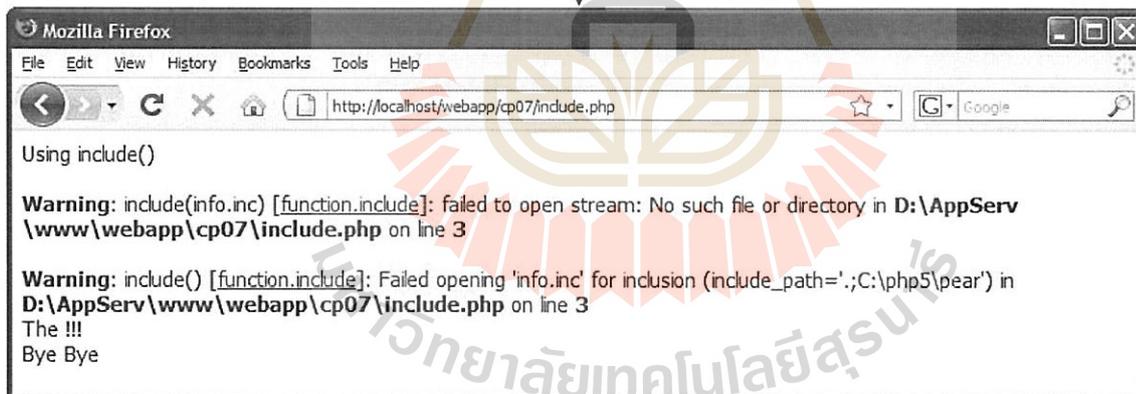
รูปที่ 7.4: การแสดงผลข้อผิดพลาดจากการใช้คำสั่ง require() ดึงแฟ้มข้อมูล

จะสังเกตเห็นว่า มีการทำงานของคำสั่งในบรรทัดที่ 2 ที่แสดงข้อความ “Using require()” แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดใหม่ พอเจอคำสั่งในบรรทัดที่ 3 ซึ่งเป็นการดึงแฟ้มข้อมูลชื่อว่า info.inc โดยใช้คำสั่ง require() แต่เนื่องจากไม่มีแฟ้มข้อมูล info.inc อยู่ จึงแจ้งข้อผิดพลาด แล้วหยุดการทำงานทันที ดังนั้นคำสั่งในบรรทัดที่ 4 และ 5 จึงไม่ถูกแสดงออก

ตัวอย่าง การใช้คำสั่ง include() แต่ไม่มีแฟ้มข้อมูล info.inc

```

1. <?php
2.   echo "Using include(<br>";
3.   include("info.inc");
4.   echo "The $color $text !!!";
5.   echo "<br>Bye Bye";
6.   ?>
    
```



รูปที่ 7.5: การแสดงผลคำเตือนจากการใช้คำสั่ง include() ดึงแฟ้มข้อมูล

ส่วนการใช้คำสั่ง include() นั้น จะสังเกตเห็นว่ามีการทำงานของคำสั่งในบรรทัดที่ 2 ที่แสดงข้อความ “Using include()” แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดใหม่ พอเจอคำสั่งในบรรทัดที่ 3 ซึ่งเป็นการดึงแฟ้มข้อมูลชื่อว่า info.inc โดยใช้คำสั่ง include() แต่เนื่องจากไม่มีแฟ้มข้อมูล info.inc อยู่ จึงแจ้งข้อผิดพลาด แล้วไปทำงานตามคำสั่งในบรรทัดที่ 4 ที่แสดงเฉพาะข้อความ The !!! แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดใหม่ แต่ไม่ได้แสดงค่าของตัวแปร \$color และ \$text เนื่องจากตัวแปรทั้ง 2 อยู่ในแฟ้มข้อมูล info.inc ที่หาไม่พบ ส่วนบรรทัดที่ 5 นั้นก็แสดงข้อความ “Bye Bye” แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดใหม่

ตัวอย่าง การแสดงผลของการใช้คำสั่ง require() เมื่อมีแฟ้มข้อมูล info.inc



รูปที่ 7.6: การแสดงผลของการใช้คำสั่ง require()

ตัวอย่าง การแสดงผลของการใช้คำสั่ง include() เมื่อมีแฟ้มข้อมูล info.inc



รูปที่ 7.7: การแสดงผลของการใช้คำสั่ง include()

ส่วนการใช้คำสั่ง require_once() และ include_once() นั้น ก็มีลักษณะการทำงานเช่นเดียวกับกับคำสั่ง require() และ include() ตามลำดับ แต่ที่แตกต่างกันคือการใช้คำสั่ง require_once() หรือ include_once() จะทำการดึงแฟ้มข้อมูลเข้ามาเพียงแค่ครั้งเดียวเท่านั้น เหมาะสำหรับใช้ในกรณีที่จะมีการดึงแฟ้มข้อมูลเข้ามามากกว่า 1 ครั้ง ทั้งนี้เพื่อป้องกันข้อผิดพลาดที่อาจเกิดขึ้นได้เช่น การประกาศรูปแบบของฟังก์ชันหรือการกำหนดค่าให้กับตัวแปรต่าง ๆ ที่เกี่ยวข้องกับแฟ้มข้อมูล เมื่อมีการดึงแฟ้มข้อมูลเข้ามาใหม่ ค่าของตัวแปรอาจเปลี่ยนไป เป็นต้น

7.2 ฟังก์ชันเกี่ยวกับการวาดรูป

ภาษา PHP มีฟังก์ชันที่ช่วยในการจัดการรูปภาพมากมาย เช่น ช่วยในการวาดรูปสี่เหลี่ยม วงกลม วงรี เป็นต้น ดังแสดงในตารางที่ 7.1 พร้อมกับรูปแบบการใช้งานและความหมาย

ตารางที่ 7.1: ฟังก์ชันต่าง ๆ ในการวาดรูปพร้อมทั้งรูปแบบและความหมาย

ฟังก์ชัน	รูปแบบ	ความหมาย
image2wbmp	int image2wbmp (resource image [, string filename [, int threshold]])	แสดงรูปภาพสกุล .wbmp ที่เบราว์เซอร์หรือเขียนลงไฟล์
image_type_to_extension	string image_type_to_extension (int imagetype [, bool include_dot])	หาสกุลของชนิดรูปภาพที่กำหนด
imagealphablending	bool imagealphablending (resource image, bool blendmode)	ตั้งค่าโหมดในการผสมผสานสำหรับรูปภาพ
imageantialias	bool imageantialias (resource image, bool on)	ทดสอบว่าการใช้ฟังก์ชัน antialias ควรใช้หรือไม่
imagearc	bool imagearc (resource image, int cx, int cy, int w, int h, int s, int e, int color)	วาดส่วนของวงรี
imagechar	bool imagechar (resource image, int font, int x, int y, string c, int color)	วาดตัวอักษรตามแนวนอน
imagecharup	bool imagecharup (resource image, int font, int x, int y, string c, int color)	วาดตัวอักษรตามแนวตั้ง
imagecolorallocate	int imagecolorallocate (resource image, int red, int green, int blue)	กำหนดสีสำหรับรูปภาพ
imagecolorallocatealpha	int imagecolorallocatealpha (resource image, int red, int green, int blue, int alpha)	กำหนดสีและระดับอัลฟาสำหรับรูปภาพ

ฟังก์ชัน	รูปแบบ	ความหมาย
imagecolorat	int imagecolorat (resource image, int x, int y)	หาตัวชี้ของสีของพิกเซลที่กำหนด
imagecolorclosest	int imagecolorclosest (resource image, int red, int green, int blue)	หาตัวชี้ของสีที่ใกล้เคียงที่สุดของสีที่กำหนด
imagecolorclosestalpha	int imagecolorclosestalpha (resource image, int red, int green, int blue, int alpha)	หาตัวชี้ของสีที่ใกล้เคียงที่สุดของสีที่กำหนดและระดับของอัลฟา
imagecolordeallocate	bool imagecolordeallocate (resource image, int color)	ยกเลิกการกำหนดสีสำหรับรูปภาพ
imagecolorexact	int imagecolorexact (resource image, int red, int green, int blue)	หาตัวชี้ของสีที่กำหนด
imagecolorexactalpha	int imagecolorexactalpha (resource image, int red, int green, int blue, int alpha)	หาตัวชี้และอัลฟาของสีที่กำหนด
imagecolormatch	bool imagecolormatch (resource image1, resource image2)	ทำให้สีของแผงสี (palette) ของรูปภาพให้คล้ายกับสีจริง (true color) มากที่สุด
imagecolorresolve	int imagecolorresolve (resource image, int red, int green, int blue)	หาตัวชี้ของสีที่กำหนดหรือสีที่ใกล้เคียงที่สุด
imagecolorresolvealpha	int imagecolorresolvealpha (resource image, int red, int green, int blue, int alpha)	หาตัวชี้ของสีที่กำหนดและอัลฟาหรือสีที่ใกล้เคียงที่สุด
imagecolorset	void imagecolorset (resource image, int index, int red, int green, int blue)	ตั้งค่าสำหรับตัวชี้ที่แผงสี
imagecolorsforindex	array imagecolorsforindex (resource image, int index)	หาสีสำหรับเป็นตัวชี้ (index)
imagecolorstotal	int imagecolorstotal (resource image)	หาจำนวนสีในรูปภาพที่ใช้แผงสี
imagecolortransparent	int imagecolortransparent (resource image [, int color])	กำหนดสีที่เป็น transparent
imagecopy	bool imagecopy (resource dst_im, resource src_im, int dst_x, int dst_y, int src_x, int src_y, int src_w, int src_h)	สำเนาส่วนของรูปภาพ
imagecopymerge	bool imagecopymerge (resource dst_im, resource src_im, int dst_x, int dst_y, int src_x, int src_y, int src_w, int src_h, int pct)	สำเนาและรวมส่วนของรูปภาพ
imagecopymergegray	bool imagecopymergegray (resource dst_im, resource src_im, int dst_x, int dst_y, int src_x, int src_y, int src_w, int src_h, int pct)	สำเนาและรวมส่วนของรูปภาพแบบสเกลสีเทา (gray scale)
imagecopyresampled	bool imagecopyresampled (resource dst_image, resource src_image, int dst_x, int dst_y, int src_x, int src_y, int dst_w, int dst_h, int src_w, int src_h)	สำเนาและปรับขนาดของรูปภาพด้วยการสุ่มตัวอย่างซ้ำ (resampling)
imagecopyresized	bool imagecopyresized (resource dst_image, resource src_image, int dst_x, int dst_y, int src_x, int src_y, int dst_w, int dst_h, int src_w, int src_h)	สำเนาและปรับส่วนของรูปภาพ
imagecreate	resource imagecreate (int x_size, int y_size)	สร้างรูปภาพที่มีแผงสี (palette)
imagecreatefromgd	resource imagecreatefromgd (string filename)	สร้างรูปภาพใหม่จากแฟ้ม GD หรือ URL
imagecreatefromgd2	resource imagecreatefromgd2 (string filename)	สร้างรูปภาพใหม่จากแฟ้ม GD2 หรือ URL
imagecreatefromgd2part	resource imagecreatefromgd2part (string filename, int srcX, int srcY, int width, int height)	สร้างรูปภาพใหม่จากส่วนหนึ่งของแฟ้ม GD2 หรือ URL
imagecreatefromgif	resource imagecreatefromgif (string filename)	สร้างรูปภาพใหม่สกุล .gif จากแฟ้มข้อมูลหรือ URL
imagecreatefromjpeg	resource imagecreatefromjpeg (string filename)	สร้างรูปภาพใหม่สกุล .jpeg จากแฟ้มข้อมูลหรือ URL
imagecreatefrompng	resource imagecreatefrompng (string filename)	สร้างรูปภาพใหม่สกุล .png จากแฟ้มข้อมูลหรือ URL
imagecreatefromstring	resource imagecreatefromstring (string image)	สร้างรูปภาพใหม่จากสตริงของรูปภาพในสตรีมที่กำหนด
imagecreatefromwbmp	resource imagecreatefromwbmp (string filename)	สร้างรูปภาพใหม่จากแฟ้มข้อมูลหรือ URL
imagecreatefromxbm	resource imagecreatefromxbm (string filename)	สร้างรูปภาพใหม่จากแฟ้มข้อมูลหรือ URL
imagecreatefromxpm	resource imagecreatefromxpm (string filename)	สร้างรูปภาพใหม่จากแฟ้มข้อมูลหรือ URL
imagecreatetruecolor	resource imagecreatetruecolor (int x_size, int y_size)	สร้างรูปภาพใหม่แบบสีจริง (True Color)
imagedashedline	bool imagedashedline (resource image, int x1, int y1, int x2, int y2, int color)	วาดเส้นประ
imagedestroy	bool imagedestroy (resource image)	ทำลายรูปภาพ
imageellipse	bool imageellipse (resource image, int cx, int cy, int w, int h, int color)	วาดรูปวงรี

ฟังก์ชัน	รูปแบบ	ความหมาย
imagefill	bool imagefill (resource image, int x, int y, int color)	เติมสีให้กับพื้นที่ ๆ กำหนด
imagefilledarc	bool imagefilledarc (resource image, int cx, int cy, int w, int h, int s, int e, int color, int style)	วาดเส้นของรูปวงรีและเติมสีให้กับส่วนนั้น
imagefilledellipse	bool imagefilledellipse (resource image, int cx, int cy, int w, int h, int color)	วาดรูปวงรีและเติมสี
imagefilledpolygon	bool imagefilledpolygon (resource image, array points, int num_points, int color)	วาดรูปเหลี่ยม (ที่มากกว่า 4 ด้าน) และเติมสี
imagefilledrectangle	bool imagefilledrectangle (resource image, int x1, int y1, int x2, int y2, int color)	วาดรูปสี่เหลี่ยมผืนผ้าและเติมสี
imagefilltoborder	bool imagefilltoborder (resource image, int x, int y, int border, int color)	เติมสีที่กำหนดให้กับพื้นที่
imagefilter	bool imagefilter (resource src_im, int filtertype [, int arg1 [, int arg2 [, int arg3]])	ใช้ฟิลเตอร์กับรูปภาพ
imagefontheight	int imagefontheight (int font)	หาขนาดความสูงของตัวอักษร
imagefontwidth	int imagefontwidth (int font)	หาขนาดความกว้างของตัวอักษร
imageftbbox	array imageftbbox (float size, float angle, string font_file, string text [, array extrainfo])	ขนาดพื้นที่รูปสี่เหลี่ยมผืนผ้ารอบตัวอักษรแบบ freetype2
imagegammacorrect	bool imagegammacorrect (resource image, float inputgamma, float outputgamma)	ใช้การปรับแต่ง gamma กับรูปภาพแบบ GD
imagegd	bool imagegd (resource image [, string filename])	แสดงรูปภาพ GD ที่เบราว์เซอร์หรือเขียนลงไฟล์
imagegd2	bool imagegd2 (resource image [, string filename [, int chunk_size [, int type]])	แสดงรูปภาพ GD2 ที่เบราว์เซอร์หรือเขียนลงไฟล์
imagegif	bool imagegif (resource image [, string filename])	แสดงรูปภาพสกุล .gif ที่เบราว์เซอร์หรือเขียนลงไฟล์
imageinterlace	int imageinterlace (resource image [, int interlace])	เปิดหรือปิดการอนุญาตให้รูปภาพไขว้กัน
imageistruecolor	bool imageistruecolor (resource image)	ทดสอบว่ารูปภาพนั้นเป็นแบบสีจริงหรือไม่
imagejpeg	bool imagejpeg (resource image [, string filename [, int quality]])	แสดงรูปภาพสกุล .jpeg ที่เบราว์เซอร์หรือเขียนลงไฟล์
imageline	bool imageline (resource image, int x1, int y1, int x2, int y2, int color)	วาดเส้นตรง
imageloadfont	int imageloadfont (string file)	โหลดตัวอักษรแบบใหม่
imagepalettecopy	void imagepalettecopy (resource destination, resource source)	สำเนาแผงสี (palette) จากรูปภาพหนึ่งไปยังอีกรูปภาพหนึ่ง
imagepng	bool imagepng (resource image [, string filename])	แสดงรูปภาพสกุล .png ที่เบราว์เซอร์หรือเขียนลงไฟล์
imagepolygon	bool imagepolygon (resource image, array points, int num_points, int color)	วาดรูปโพลิگون
imagespsbbox	array imagespsbbox (string text, int font, int size [, int space, int tightness, float angle])	ขนาดพื้นที่รูปสี่เหลี่ยมผืนผ้ารอบตัวอักษรแบบ PostScript Type 1
imagespsencodefont	bool imagespsencodefont (resource font_index, string encodingfile)	เปลี่ยนลักษณะการเข้ารหัสตัวอักษรเวกเตอร์
imagespsextendfont	bool imagespsextendfont (int font_index, float extend)	ยืดหรือหดตัวอักษร
imagespsfreefont	bool imagespsfreefont (resource fontindex)	คืนหน่วยความจำที่ใช้โดยตัวอักษร PostScript Type 1
imagesploadfont	resource imagesploadfont (string filename)	โหลดตัวอักษรชนิด PostScript Type 1 จากแฟ้มข้อมูล
imagespslantfont	bool imagespslantfont (resource font_index, float slant)	ทำตัวอักษรเอียง
imagepsstext	array imagepsstext (resource image, string text, resource font, int size, int foreground, int background, int x, int y [, int space, int tightness, float angle, int antialias_steps])	วาดกลุ่มตัวอักษรใช้ตัวอักษรแบบ PostScript Type 1 บนรูปภาพ
imagerectangle	bool imagerectangle (resource image, int x1, int y1, int x2, int y2, int col)	วาดรูปสี่เหลี่ยมผืนผ้า
imagerotate	resource imagerotate (resource src_im, float angle, int bgd_color [, int ignore_transparent])	หมุนรูปภาพตามมุมองศาที่กำหนด
imagesavealpha	bool imagesavealpha (resource image, bool saveflag)	บันทึกรายละเอียดของระดับอัลฟา เมื่อทำการบันทึกเป็นแฟ้มรูปภาพสกุล .png
imagesetbrush	bool imagesetbrush (resource image, resource brush)	กำหนดรูปภาพสำหรับแปรงในการวาดเส้น

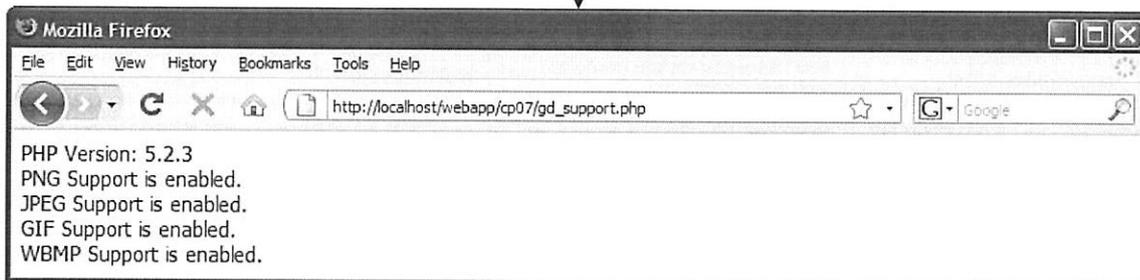
ฟังก์ชัน	รูปแบบ	ความหมาย
imagepixel	bool imagepixel (resource image, int x, int y, int color)	ตั้งค่าพิกเซลหนึ่ง ๆ
imagestyle	bool imagestyle (resource image, array style)	กำหนดรูปแบบของการวาดเส้น
imagesthickness	bool imagesthickness (resource image, int thickness)	กำหนดความหนาของการวาดเส้น
imagesttile	bool imagesttile (resource image, resource tile)	กำหนดการเติมรูปภาพแบบซ้ำให้กับพื้นหลัง
imagestring	bool imagestring (resource image, int font, int x, int y, string s, int col)	วาดกลุ่มข้อความตามแนวนอน
imagestringup	bool imagestringup (resource image, int font, int x, int y, string s, int col)	วาดกลุ่มข้อความตามแนวตั้ง
imagesx	int imagesx (resource image)	หาความกว้างของรูปภาพ
imagesy	int imagesy (resource image)	หาความสูงของรูปภาพ
imagetruecolortopalette	bool imagetruecolortopalette (resource image, bool dither, int ncolors)	แปลงรูปภาพที่เป็นสีจริง (true color) เป็นแผงสี (palette)
imageftbbox	array imageftbbox (float size, float angle, string fontfile, string text)	ขนาดพื้นที่รูปสี่เหลี่ยมผืนผ้ารอบตัวอักษรแบบ TrueType
imagefttext	array imagefttext (resource image, float size, float angle, int x, int y, int color, string fontfile, string text)	เขียนตัวอักษรที่รูปภาพโดยใช้ตัวอักษรแบบ TrueType
imagetypes	int imagetypes (void)	ส่งกลับชนิดรูปภาพที่สนับสนุนด้วย PHP
imagewbmp	bool imagewbmp (resource image [, string filename [, int foreground]])	แสดงรูปภาพสกุล .wbmp ที่เบราว์เซอร์หรือเขียนลงไฟล์
imagexbm	bool imagexbm (resource image, string filename [, int foreground])	แสดงรูปภาพสกุล .xbm ที่เบราว์เซอร์หรือเขียนลงไฟล์

ใน PHP รุ่นก่อนเวอร์ชัน 4.3 ต้องเรียกใช้ไลบรารี GD หรือไมก็ GD2 เมื่อต้องการใช้ฟังก์ชันในการวาดรูป (แนะนำให้ใช้ GD2 เนื่องจากมีความเสถียรมากกว่า) ส่วนในเวอร์ชัน 4.3 เป็นต้นมานั้น PHP ได้รวมคุณสมบัติเกี่ยวกับไลบรารี GD ไว้เรียบร้อยแล้ว สามารถใช้งานได้เลย ซึ่งชนิดหรือสกุลของรูปภาพที่ภาษา PHP สนับสนุนนั้น สามารถทดสอบโดยการใส่คำสั่งดังในรูปที่ 7.8

```

1.  <?php
2.      echo "PHP Version: ".PHP_VERSION."<br>";
3.      if(imagetypes() & IMG_PNG) {
4.          echo "PNG Support is enabled.<br>";
5.      }
6.      if(imagetypes() & IMG_JPG) {
7.          echo "JPEG Support is enabled.<br>";
8.      }
9.      if(imagetypes() & IMG_GIF) {
10.         echo "GIF Support is enabled.<br>";
11.     }
12.     if(imagetypes() & IMG_WBMP) {
13.         echo "WBMP Support is enabled.<br>";
14.     }
15.     if (imagetypes() & IMG_XPM) {
16.         echo "XPM Support is enabled.<br>";
17.     }
18.     ?>

```



รูปที่ 7.8: การทดสอบเวอร์ชัน PHP และสกุลของรูปภาพชนิดต่าง ๆ ที่ใช้งานได้

หมายเหตุ ไฟล์รูปภาพสกุล .GIF นั้นได้ถูกลบออกจาก GD ตั้งแต่เวอร์ชัน 1.6 เป็นต้นมา เนื่องจากปัญหาทางด้านลิขสิทธิ์ จนกระทั่งเวอร์ชัน 2.0.28 ถึงได้รวมเอาสกุล .GIF เข้าไปอีกครั้ง

รูปแบบของรูปภาพต่าง ๆ นั้นมีลักษณะแตกต่างกัน ซึ่งสรุปได้ดังต่อไปนี้

1. **รูปแบบ JPEG** ออกเสียงเป็น “เจ-เพ็ก” เป็นคำย่อมาจาก Joint Photographic Experts Group ใช้ในการเก็บข้อมูลรูปภาพที่มีสีหรือเฉดสีมาก ๆ โดยใช้การบีบอัดแบบ Lossy Compression ซึ่งเป็นการบีบอัดข้อมูลให้มีขนาดเล็กลงโดยการตัดสีที่คล้ายกันทิ้ง ทำให้ข้อมูลบางส่วนขาดหายไป แต่สายตาโดยปกติไม่สามารถมองเห็นความแตกต่างได้ รูปแบบนี้ไม่เหมาะกับรูปภาพลายเส้น รูปภาพตัวอักษรหรือรูปภาพที่ใช้สีน้อย ๆ

2. **รูปแบบ GIF** ย่อมาจาก Graphics Interchange Format ใช้การบีบอัดแบบ Lossless Compression ซึ่งเป็นการบีบอัดที่ไม่ทำลายข้อมูลทิ้ง จึงทำให้การบีบอัดเช่นนี้สามารถลดขนาดของแฟ้มข้อมูลได้น้อย แต่เป็นที่นิยมสำหรับการจัดเก็บรูปภาพที่เป็นตัวอักษร ลายเส้น รูปภาพที่ใช้สีน้อย และรูปภาพเคลื่อนไหว รูปแบบ GIF มาตรฐานใช้รูปแบบการบีบอัดที่เรียกว่า LZW (Lempel Ziv Welch) ซึ่งจดทะเบียนสิทธิบัตรโดย UNISYS ดังนั้นโปรแกรมใดที่สามารถอ่านหรือเขียนแฟ้มข้อมูลในรูปแบบ GIF ได้ จะต้องเสียค่าลิขสิทธิ์ให้กับ UNISYS เช่น โปรแกรม PhotoShop ของบริษัท Adobe เป็นต้น นำเสียดายที่รูปแบบ GIF ใช้กันอย่างแพร่หลายบนเว็บก่อนที่ UNISYS จะยกเรื่องลิขสิทธิ์ขึ้นมาฟ้อง

3. **รูปแบบ PNG** ออกเสียงเป็น “ปิง” เป็นคำย่อมาจาก Portable Network Graphics ใช้การบีบอัดแบบ Lossless Compression เช่นเดียวกับ GIF รูปแบบ ดังนั้นรูปแบบนี้จึงถูกนำมาใช้แทนที่ GIF แต่ไม่สามารถบันทึกรูปภาพเคลื่อนไหวได้

4. **รูปแบบ WBMP** ย่อมาจาก Wireless Bitmap ได้ถูกออกแบบขึ้นมาใช้กับอุปกรณ์ต่าง ๆ แบบไร้สายโดยเฉพาะ

การวาดรูปโดยใช้ฟังก์ชัน PHP นั้น มี 4 ลำดับขั้นตอนคือ

1. กำหนดพื้นที่และสีสำหรับการวาดรูปโดยใช้ฟังก์ชัน `imagecreate()` ในการกำหนดพื้นที่ และฟังก์ชัน `imagecolorallocate()` ในการกำหนดสี ซึ่งมีรูปแบบการใช้งานดังนี้

```
resource imagecreate (int x_size, int y_size)
int imagecolorallocate (resource image, int red, int green, int blue)
```

2. วาดรูปหรือตัวอักษรบนพื้นที่ที่กำหนด โดยใช้ฟังก์ชัน `imagefill()` เพื่อเติมสี, ฟังก์ชัน `imageline()` ในการวาดเส้น หรือ `imagestring()` ในการวาดกลุ่มตัวอักษร ซึ่งมีรูปแบบการใช้งานดังนี้

```
bool imagefill (resource image, int x, int y, int color)
bool imageline (resource image, int x1, int y1, int x2, int y2, int color)
bool imagestring (resource image, int font, int x, int y, string s, int color)
```

3. แสดงรูปภาพออกที่จอภาพหรือบันทึกเป็นแฟ้มสกุลต่าง ๆ โดยใช้ฟังก์ชัน `header()` และ `imagepng()` เป็นต้น ซึ่งมีรูปแบบการใช้งานดังนี้

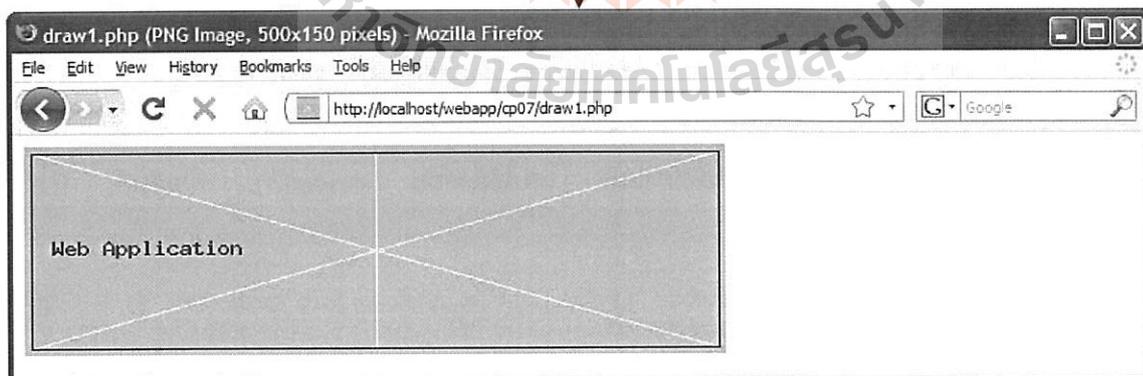
```
void header (string string [, bool replace [, int
http_response_code]])
bool imagepng (resource image [, string filename])
```

4. คืนหน่วยความจำที่ใช้ในการวาดรูปให้กับระบบ โดยใช้ฟังก์ชัน imagedestroy () ซึ่งมีรูปแบบการใช้งานดังนี้

```
bool imagedestroy (resource image)
```

ตัวอย่างที่ 1 การใช้ฟังก์ชันในการวาดรูปสี่เหลี่ยมผืนผ้าเส้นสีดำบนพื้นที่ ขนาด 500 x 150 พิกเซล และวาดเส้นทแยงมุมสีขาวจากมุมซ้ายล่างไปยังมุมขวาบน มุมซ้ายบนไปยังมุมขวาล่าง และเส้นแบ่งครึ่ง พร้อมกับแสดงข้อความสีดำว่า “Web Application” ใช้ตัวอักษรกลุ่มที่ 5 ในตำแหน่งที่ห่างจากขอบด้านซ้าย 20 พิกเซล และขอบด้านบน 65 พิกเซล

```
1. <?php
2. // create a canvas 500*150
3. $height = 150;
4. $width = 500;
5. $img = imagecreate($width, $height)
6.     or die("Cannot Initialize new GD image stream");
7.
8. // setup color
9. $white = imagecolorallocate ($img, 255, 255, 255);
10. $black = imagecolorallocate ($img, 0, 0, 0);
11. $pink = imagecolorallocate ($img, 250, 150, 250);
12.
13. // draw on image
14. imagefill($img, 0, 0, $pink);
15. imageline($img, 5, 145, 495, 5, $white);
16. imageline($img, 5, 5, 495, 145, $white);
17. imageline($img, 250, 5, 250,145, $white);
18. imagerectangle($img, 5, 5, 495, 145, $black);
19. imagestring($img, 5, 20, 65, "Web Application", $black);
20.
21. // output image: ouput the image to the browser
22. header ("Content-type: image/png");
23. imagepng ($img);
24.
25. // free memory
26. imagedestroy($img);
27. ?>
```



รูปที่ 7.9: วาดรูปสี่เหลี่ยมผืนผ้าสีชมพูขนาด 500 x 150 พิกเซล

ช่วงบรรทัดที่ 3-9 นั้นเป็นการกำหนดพื้นที่และสีสำหรับการวาดรูป

บรรทัดที่ 3 กำหนดค่า 150 ให้กับตัวแปร \$height

บรรทัดที่ 4 กำหนดค่า 500 ให้กับตัวแปร \$width

บรรทัดที่ 5-6 ใช้ฟังก์ชัน `imagecreate()` สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความสูงมีขนาดเท่ากับ 150 พิกเซล และความกว้างมีขนาดเท่ากับ 500 พิกเซล แล้วให้ตัวแปร `$img` เป็นตัวชี้หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้น ว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง `die()` เพื่อแสดงข้อความว่า “Cannot Initialize new GD image stream” ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

ช่วงบรรทัดที่ 9-17 นั้นเป็นการวาดรูป

บรรทัดที่ 9 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$white`

บรรทัดที่ 10 กำหนดสีดำที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$black`

บรรทัดที่ 11 กำหนดสีชมพูที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$pink`

ช่วงบรรทัดที่ 14-19 นั้นเป็นการวาดรูป

บรรทัดที่ 14 เรียกใช้ฟังก์ชัน `imagefill()` เพื่อใส่สีพื้นเป็นสีชมพู โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 0 (ศูนย์) บนแกน x และจุดพิกัด 0 (ศูนย์) บนแกน y

บรรทัดที่ 15 เรียกใช้ฟังก์ชัน `imageline()` เพื่อวาดเส้นสีขาว โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 5 บนแกน x และ จุดพิกัด 145 บนแกน y และไปสิ้นสุดที่ตำแหน่ง 495 บนแกน x และ 5 บนแกน y

บรรทัดที่ 16 เรียกใช้ฟังก์ชัน `imageline()` เพื่อวาดเส้นสีขาว โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 5 บนแกน x และ จุดพิกัด 5 บนแกน y และไปสิ้นสุดที่ตำแหน่งจุดพิกัด 495 บนแกน x และจุดพิกัด 145 บนแกน y

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน `imageline()` เพื่อวาดเส้นสีขาว โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 250 บนแกน x และจุดพิกัด 5 บนแกน y และไปสิ้นสุดที่ตำแหน่งจุดพิกัด 250 บนแกน x และจุดพิกัด 145 บนแกน y

บรรทัดที่ 18 เรียกใช้ฟังก์ชัน `imagerectangle()` เพื่อวาดรูปสี่เหลี่ยมผืนผ้าเส้นขอบสีดำ โดยตำแหน่งบนซ้ายคือจุดพิกัด 5 บนแกน x และจุดพิกัด 5 บนแกน y และตำแหน่งล่างขวาคือจุดพิกัด 495 บนแกน x และจุดพิกัด 145 บนแกน y

บรรทัดที่ 19 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร Web Application กลุ่มที่ 5 สีดำ (กลุ่มมีทั้งหมด 5 กลุ่มคือ 1 ถึง 5) โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 65 บนแกน y

ช่วงบรรทัดที่ 22-23 นั้นเป็นการแสดงผลออกที่จอภาพ

บรรทัดที่ 22 เรียกใช้ฟังก์ชัน `header()` เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล .png

บรรทัดที่ 23 เรียกใช้ฟังก์ชัน `imagepng()` เพื่อแสดงรูปภาพสกุล .png ที่สร้างขึ้นโดยมีตัวชี้คือ `$img` ออกที่เว็บเบราว์เซอร์

ช่วงบรรทัดที่ 26 นั้นเป็นการทำลายหรือคืนพื้นที่ในการวาดรูปให้กับระบบ

บรรทัดที่ 26 เรียกใช้ฟังก์ชัน `imagedestroy()` เพื่อคืนพื้นที่ที่ตัวแปร `$img` ให้อยู่ให้กับระบบ

ตัวอย่างที่ 2 การใช้ฟังก์ชันในการวาดรูปวงกลมขนาด 150 พิกเซลมีสีพื้นเป็นสีแดง บนพื้นที่วาดรูปขนาด 200 x 200 พิกเซล โดยให้จุดศูนย์กลางอยู่ที่ตำแหน่งกึ่งกลางของพื้นที่วาดรูป

```

1. <?php
2. // create a canvas 200*200
3. $img = imagecreate(200, 200);
4.     or die("Cannot Initialize new GD image stream");
5. // allocate some color
6. $white = imagecolorallocate($img, 255, 255, 255);
7. $red = imagecolorallocate ($img, 255, 0, 0);
8.
9. // fill in red circle
10. imagefilledellipse($img, 100, 100, 150, 150, $red);
11.
12. // output image: output image to the browser
13. header("Content-type: image/png");
14. imagepng($img);
15.
16. // free memory
17. imagedestroy($img);
18. ?>
    
```



รูปที่ 7.10: วาดรูปวงกลมสีแดงขนาด 150 พิกเซลอยู่ตำแหน่งตรงกลางของพื้นที่วาดรูป

บรรทัดที่ 3-4 ใช้ฟังก์ชัน `imagecreate()` สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้างและความสูงมีขนาดเท่ากับ 200 พิกเซล แล้วให้ตัวแปร `$img` เป็นตัวชี้ หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้นว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง `die()` เพื่อแสดงข้อความว่า “Cannot Initialize new GD image stream” ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

บรรทัดที่ 6 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$white`

บรรทัดที่ 7 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$red`

บรรทัดที่ 10 เรียกใช้ฟังก์ชัน `imagefilledellipse()` เพื่อวาดวงรีให้มีสีพื้นเป็นสีแดง โดยมีจุดศูนย์กลางอยู่ที่ตำแหน่งจุดพิกัด 100 บนแกน x และจุดพิกัด 100 บนแกน y ให้มีความกว้างเท่ากับ 150 พิกเซล และความสูงเท่ากับ 150 พิกเซล (เมื่อความกว้างและความสูงมีค่าเท่ากัน ทำให้รูปที่ได้เป็นวงกลม)

บรรทัดที่ 13 เรียกใช้ฟังก์ชัน `header()` เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล `.png`

บรรทัดที่ 14 เรียกใช้ฟังก์ชัน `imagepng()` เพื่อแสดงรูปภาพสกุล `.png` ที่สร้างขึ้นมาโดยมีตัวชี้คือ `$img` ออกที่เว็บเบราว์เซอร์

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน `imagedestroy()` เพื่อคืนพื้นที่ที่ตัวแปร `$img` ครอบไว้ให้กับระบบ

ตัวอย่างที่ 3 การใช้ฟังก์ชันในการวาดรูปสี่เหลี่ยมจัตุรัสสีดำ ขนาด 200 x 200 พิกเซล และวาดรูปวงกลมสีแดงขนาด 150 พิกเซลด้วยเส้นขอบสีขาวขนาด 5 พิกเซล โดยให้จุดศูนย์กลางอยู่ที่ตำแหน่งกึ่งกลางของพื้นที่สี่เหลี่ยมจัตุรัส

```

1. <?php
2. // create a canvas 200*200
3. $height = 200;
4. $width = 200;
5. $img = imagecreate($width, $height)
6. or die("Cannot Initialize new GD image stream");
7. $white = imagecolorallocate($img, 255, 255, 255);
8. $black = imagecolorallocate($img, 0, 0, 0);
9. $red = imagecolorallocate($img, 255, 0, 0);
10.
11. // draw on image: fill the background in Black color
12. imagefill($img, 0, 0, $black);
13.
14. // draw the White and Red ellipses
15. imagefilledellipse($img, 100, 100, 150, 150, $white);
16. imagefilledellipse($img, 100, 100, 145, 145, $red);
17.
18. // output image: output image to the browser
19. header("Content-type: image/png");
20. imagepng($img);
21.
22. // free memory
23. imagedestroy($img);
24. ?>
    
```



รูปที่ 7.11: วาดรูปสี่เหลี่ยมจัตุรัสสีดำขนาด 200 พิกเซลและวงกลมสีแดงขนาด 150 พิกเซลเส้นขอบสีขาวขนาด 5 พิกเซลอยู่ตำแหน่งตรงกลาง

ช่วงบรรทัดที่ 3-9 นั้นเป็นการกำหนดพื้นที่และสีสำหรับการวาดรูป

บรรทัดที่ 3 กำหนดค่า 200 ให้กับตัวแปร \$height

บรรทัดที่ 4 กำหนดค่า 200 ให้กับตัวแปร \$width

บรรทัดที่ 5-6 ใช้ฟังก์ชัน imagecreate() สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้าง 200 พิกเซลและความสูงมีขนาดเท่ากับ 200 พิกเซล แล้วให้ตัวแปร \$img เป็นตัวชี้ หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้นว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง die() เพื่อแสดงข้อความว่า “Cannot Initialize new GD image stream” ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

บรรทัดที่ 7 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร \$white

บรรทัดที่ 8 กำหนดสีดำที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร \$black

บรรทัดที่ 9 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร \$red

ช่วงบรรทัดที่ 12 นั้นเป็นการเติมสีให้กับพื้นที่ที่จะวาดรูป

บรรทัดที่ 12 เรียกใช้ฟังก์ชัน `imagefill()` เพื่อใส่สีพื้นเป็นสีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 0 (ศูนย์) บนแกน x และจุดพิกัด 0 (ศูนย์) บนแกน y

ช่วงบรรทัดที่ 15-16 นั้นเป็นการวาดรูป

บรรทัดที่ 15 เรียกใช้ฟังก์ชัน `imagefilledellipse()` เพื่อวาดวงรีให้มีสีพื้นเป็นสีขาว โดยมีจุดศูนย์กลางอยู่ที่ตำแหน่งจุดพิกัด 100 บนแกน x และจุดพิกัด 100 บนแกน y ให้มีความกว้างเท่ากับ 150 พิกเซล และความสูงเท่ากับ 150 พิกเซล (เมื่อความกว้างและความสูงมีค่าเท่ากัน ทำให้รูปที่ได้เป็นวงกลม)

บรรทัดที่ 16 เรียกใช้ฟังก์ชัน `imagefilledellipse()` เพื่อวาดวงรีให้มีสีพื้นเป็นสีแดง โดยมีจุดศูนย์กลางอยู่ที่จุดพิกัด 100 บนแกน x และจุดพิกัด 100 บนแกน y ให้มีความกว้างเท่ากับ 145 พิกเซล และความสูงเท่ากับ 145 พิกเซล (เมื่อความกว้างและความสูงมีค่าเท่ากัน ทำให้รูปที่ได้เป็นวงกลม)

ช่วงบรรทัดที่ 19-20 นั้นเป็นการแสดงผลออกที่จอภาพ

บรรทัดที่ 19 เรียกใช้ฟังก์ชัน `header()` เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล .png

บรรทัดที่ 20 เรียกใช้ฟังก์ชัน `imagepng()` เพื่อแสดงรูปภาพสกุล .png ที่สร้างขึ้นโดยมีตัวชี้คือ `$img` ออกที่เว็บเบราว์เซอร์

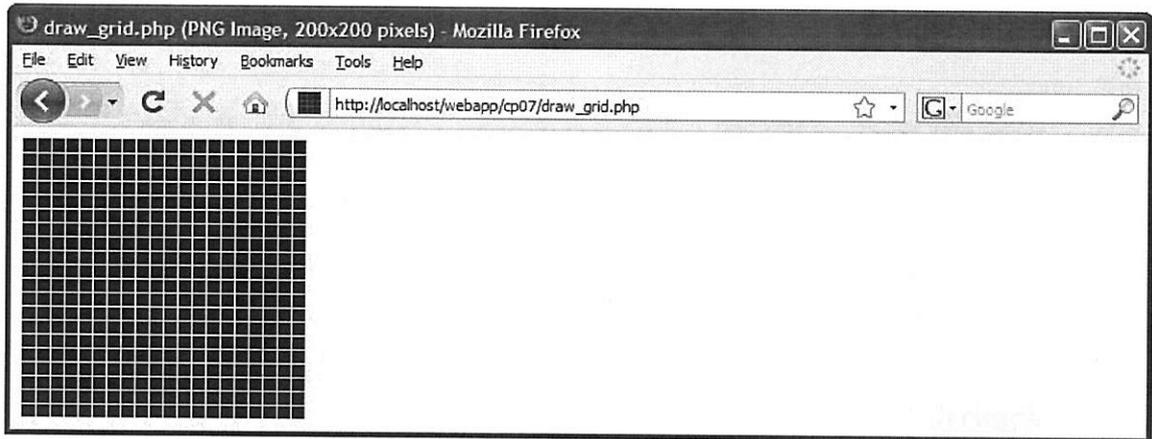
ช่วงบรรทัดที่ 23 นั้นเป็นการทำลายหรือคืนพื้นที่ในการวาดรูปให้กับระบบ

บรรทัดที่ 23 เรียกใช้ฟังก์ชัน `imagedestroy()` เพื่อคืนพื้นที่ที่ตัวแปร `$img` ซ้ำอยู่ให้กับระบบ

ตัวอย่างที่ 4 การใช้ฟังก์ชันในการวาดรูปสี่เหลี่ยมจัตุรัสสีดำ ขนาด 200 x 200 พิกเซล และวาดเส้นสีขาวตามแนวนอนและแนวตั้ง โดยให้มีระยะห่างแต่ละช่วงเท่ากับ 10 พิกเซล

```

1. <?php
2. // create a canvas 200*200
3. $height = 200;
4. $width = 200;
5. $img = imagecreate($width, $height)
6.     or die("Cannot Initialize new GD image stream");
7. $white = imagecolorallocate($img, 255, 255, 255);
8. $black = imagecolorallocate($img, 0, 0, 0);
9. $red = imagecolorallocate($img, 255, 0, 0);
10.
11. // draw on image: fill the background in Black color
12. imagefill($img, 0, 0, $black);
13.
14. // draw on image: draw the White and Red ellipses
15. for($i=10, $j=10; $i<=200; $i += 10, $j += 10){
16.     imageline($img, $i, 0, $i, $height, $white);
17.     imageline($img, 0, $j, $width, $j, $white);
18. }
19.
20. // output image: output image to the browser
21. header("Content-type: image/png");
22. imagepng($img);
23.
24. // free memory
25. imagedestroy($img);
26. ?>
    
```



รูปที่ 7.12: วาดรูปสี่เหลี่ยมจัตุรัสสีดำขนาด 200 พิกเซลและเส้นสีขาวตามแนวตั้งและแนวนอน โดยเว้นระยะห่าง 10 พิกเซล

ช่วงบรรทัดที่ 3-9 นั้นเป็นการกำหนดพื้นที่และสีสำหรับการวาดรูป

บรรทัดที่ 3 กำหนดค่า 200 ให้กับตัวแปร \$height

บรรทัดที่ 4 กำหนดค่า 200 ให้กับตัวแปร \$width

บรรทัดที่ 5-6 ใช้ฟังก์ชัน imagecreate() สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้าง 200 พิกเซลและความสูงมีขนาดเท่ากับ 200 พิกเซล แล้วให้ตัวแปร \$img เป็นตัวชี้ หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้นว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง die() เพื่อแสดงข้อความว่า “Cannot Initialize new GD image stream” ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

บรรทัดที่ 7 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร \$white

บรรทัดที่ 8 กำหนดสีดำที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร \$black

บรรทัดที่ 9 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร \$red

ช่วงบรรทัดที่ 12 นั้นเป็นการเติมสีให้กับพื้นที่ที่จะวาดรูป

บรรทัดที่ 12 เรียกใช้ฟังก์ชัน imagefill() เพื่อใส่สีพื้นเป็นสีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 0 (ศูนย์) บนแกน x และจุดพิกัด 0 (ศูนย์) บนแกน y

ช่วงบรรทัดที่ 15-18 นั้นเป็นการวาดรูปโดยใช้ลูป for วนรอบไปเรื่อย ๆ จนกว่าการทดสอบจะเป็นเท็จ

บรรทัดที่ 15 กำหนดค่าเริ่มต้น 10 ให้กับตัวแปร \$i ซึ่งจะใช้เป็นจุดพิกัดเริ่มต้นของตำแหน่งในแกน x และกำหนดค่าเริ่มต้น 10 ให้กับตัวแปร \$j ซึ่งจะใช้เป็นจุดพิกัดเริ่มต้นของตำแหน่งในแกน y

บรรทัดที่ 16 เรียกใช้ฟังก์ชัน imageline() เพื่อวาดเส้นสีขาวในแนวตั้ง โดยมีจุดพิกัดเริ่มต้นที่ค่าที่อยู่ในตัวแปร \$i และ 0 นั่นคือ (10, 0) และจุดพิกัดสุดท้ายอยู่ที่ค่าที่อยู่ในตัวแปร \$i และค่าที่อยู่ในตัวแปร \$height นั่นคือ (10, 200)

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน imageline() เพื่อวาดเส้นสีขาวในแนวนอน โดยมีจุดพิกัดเริ่มต้นที่ 0 และค่าที่อยู่ในตัวแปร \$j นั่นคือ (0, 10) และจุดพิกัดสุดท้ายอยู่ที่ค่าที่อยู่ในตัวแปร \$width และค่าที่อยู่ในตัวแปร \$j นั่นคือ (200, 10)

ช่วงบรรทัดที่ 21-22 นั้นเป็นการแสดงผลออกที่จอภาพ

บรรทัดที่ 21 เรียกใช้ฟังก์ชัน `header()` เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล `.png`

บรรทัดที่ 22 เรียกใช้ฟังก์ชัน `imagepng()` เพื่อแสดงรูปภาพสกุล `.png` ที่สร้างขึ้นโดยมีตัวชี้คือ `$img` ออกที่เว็บเบราว์เซอร์

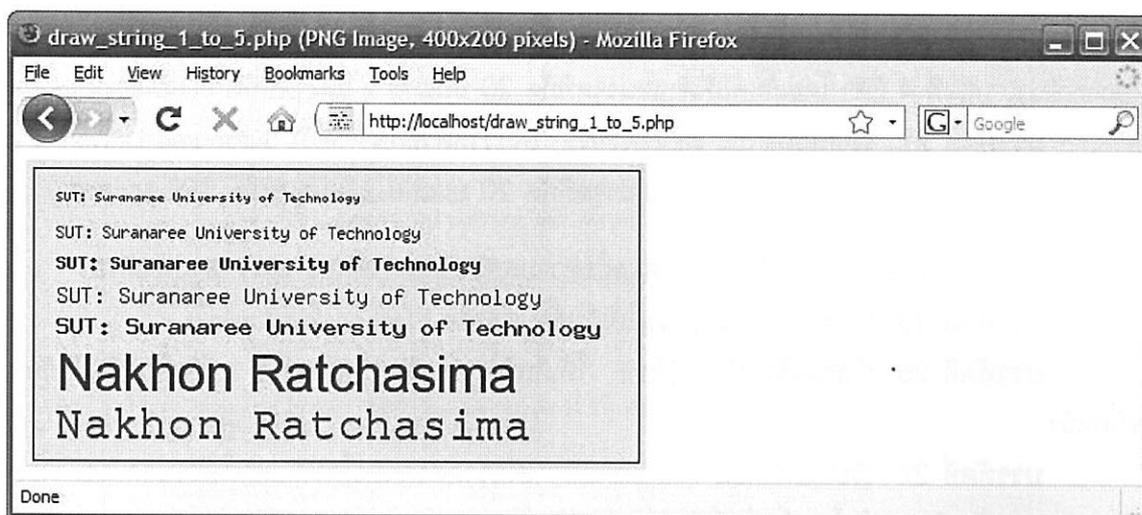
ช่วงบรรทัดที่ 25 นั้นเป็นการทำลายหรือคืนพื้นที่ในการวาดรูปให้กับระบบ

บรรทัดที่ 25 เรียกใช้ฟังก์ชัน `imagedestroy()` เพื่อคืนพื้นที่ที่ตัวแปร `$img` ใช้อยู่ให้กับระบบ

ตัวอย่างที่ 5 การใช้ฟังก์ชันในการวาดสตริงใช้ตัวอักษรระบบจากกลุ่มที่ 1 ถึง 5 และรูปแบบตัวอักษรและขนาดที่กำหนดเอง

```

1.  <?php
2.  // create a canvas 400*200
3.  $img = imagecreate(400,200)
4.      or die("Cannot Initialize new GD image stream");
5.
6.  // setup a color for the polygon
7.  $white = imagecolorallocate($img, 255, 255, 255);
8.  $black = imagecolorallocate($img, 0, 0, 0);
9.  $red = imagecolorallocate($img, 255, 0, 0);
10. $yellow = imagecolorallocate($img, 255, 255, 0);
11.
12. // draw on image: fill the background in Yellow color
13. imagefill($img, 0, 0,$yellow);
14.
15. // draw on image: draw the rectangle and string from 1 to 5
16. imagerectangle($img, 5, 5, 395, 195, $black);
17. imagestring($img, 1, 20, 20, "SUT: Suranaree University of Technology", $black);
18. imagestring($img, 2, 20, 40, "SUT: Suranaree University of Technology", $black);
19. imagestring($img, 3, 20, 60, "SUT: Suranaree University of Technology", $black);
20. imagestring($img, 4, 20, 80, "SUT: Suranaree University of Technology", $black);
21. imagestring($img, 5, 20, 100, "SUT: Suranaree University of Technology", $black);
22. // The text to draw
23. $text = "Nakhon Ratchasima";
24. // Replace path by your own font path
25. $font = "ARIAL.TTF";
26. // Add the text
27. imagettftext($img, 25, 0, 20, 150, $black, $font, $text);
28.
29. $font = "COUR.ttf";
30. // Add some shadow to the text
31. imagettftext($img, 22, 0, 20, 181, $red, $font, $text);
32. // Add the text
33. imagettftext($img, 22, 0, 20, 180, $black, $font, $text);
34.
35. // output image: output the image to the browser
36. header("Content-type: image/png");
37. imagepng($img);
38.
39. // free memory
40. imagedestroy($img);
41. ?>
    
```



รูปที่ 7.13: วาดรูปสตริงจากตัวอักษรระบบกลุ่มที่ 1 ถึง 5 และชนิดและขนาดของตัวอักษรที่กำหนดขึ้นเอง

ช่วงบรรทัดที่ 3-4 นั้นเป็นการกำหนดพื้นที่สำหรับการวาดรูป

บรรทัดที่ 3-4 ใช้ฟังก์ชัน `imagecreate()` สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้าง 400 พิกเซลและความสูงมีขนาดเท่ากับ 200 พิกเซล แล้วให้ตัวแปร `$img` เป็นตัวชี้ หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้น ว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง `die()` เพื่อแสดงข้อความว่า “Cannot Initialize new GD image stream” ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

ช่วงบรรทัดที่ 7-10 นั้นเป็นการกำหนดสีที่ใช้ในการวาดรูป

บรรทัดที่ 7 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$white`

บรรทัดที่ 8 กำหนดสีดำที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$black`

บรรทัดที่ 9 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$red`

บรรทัดที่ 10 กำหนดสีเหลืองที่ต้องการใช้ในการวาดรูปให้กับแฉงสี เก็บไว้ที่ตัวแปร `$yellow`

ช่วงบรรทัดที่ 13 นั้นเป็นการเติมสีให้กับพื้นที่ที่จะวาดรูป

บรรทัดที่ 13 เรียกใช้ฟังก์ชัน `imagefill()` เพื่อใส่สีพื้นเป็นสีเหลือง โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 0 (ศูนย์) บนแกน x และจุดพิกัด 0 (ศูนย์) บนแกน y

ช่วงบรรทัดที่ 16-21 นั้นเป็นการวาดรูปสี่เหลี่ยมและสตริงกลุ่มที่ 1 ถึง 5

บรรทัดที่ 16 เรียกใช้ฟังก์ชัน `imagerectangle()` เพื่อวาดรูปสี่เหลี่ยมผืนผ้า โดยตำแหน่งบนซ้ายคือจุดพิกัด 5 บนแกน x และจุดพิกัด 5 บนแกน y และตำแหน่งล่างขวาคือจุดพิกัด 395 บนแกน x และจุดพิกัด 195 บนแกน y

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร SUT: Suranaree University of Technology กลุ่มที่ 1 สีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 20 บนแกน y

บรรทัดที่ 18 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร SUT: Suranaree University of Technology กลุ่มที่ 2 สีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 40 บนแกน y

บรรทัดที่ 19 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร SUT: Suranaree University of Technology กลุ่มที่ 3 สีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 60 บนแกน y

บรรทัดที่ 20 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร SUT: Suranaree University of Technology กลุ่มที่ 4 สีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 80 บนแกน y

บรรทัดที่ 21 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร SUT: Suranaree University of Technology กลุ่มที่ 5 สีดำ โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 100 บนแกน y

ช่วงบรรทัดที่ 23-33 นั้นเป็นการกำหนดข้อความสตริง ชนิดตัวอักษร และวาดรูปตัวอักษร

บรรทัดที่ 23 กำหนดตัวแปร `$text` ให้เก็บข้อมูลสตริง Nakhon Ratchasima

บรรทัดที่ 25 กำหนดตัวแปร `$font` ให้เก็บข้อมูลสตริง ARIAL.TTF ซึ่งเป็นชนิดตัวอักษรชนิดหนึ่ง

บรรทัดที่ 27 เรียกใช้ฟังก์ชัน `imagefttext()` เพื่อวาดตัวอักษรโดยกำหนดให้มีขนาด 25 พอยต์ มีมุมมองค่าเท่ากับ 0 (ศูนย์) โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 150 บนแกน y แล้ววาดตัวอักษรสีดำมีชนิดตัวอักษรเป็นแบบ ARIAL.TTF โดยเอาตัวอักษรที่อยู่ในตัวแปร `$text` มาวาด (นั่นคือตัวอักษร Nakhon Ratchasima)

บรรทัดที่ 29 กำหนดตัวแปร `$font` ให้เก็บข้อมูลสตริง COUR.TTF ซึ่งเป็นชนิดตัวอักษรชนิดหนึ่ง

บรรทัดที่ 31 เรียกใช้ฟังก์ชัน `imagefttext()` เพื่อวาดตัวอักษรโดยกำหนดให้มีขนาด 22 พอยต์ มีมุมมองค่าเท่ากับ 0 (ศูนย์) โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 181 บนแกน y แล้ววาดตัวอักษรสีแดงมีชนิดตัวอักษรเป็นแบบ COUR.TTF โดยเอาตัวอักษรที่อยู่ในตัวแปร `$text` มาวาด (นั่นคือตัวอักษร Nakhon Ratchasima) ซึ่งเป็นการวาดเงาให้กับตัวอักษรที่จะถูกวาดขึ้นโดยคำสั่งในบรรทัดที่ 33

บรรทัดที่ 33 เรียกใช้ฟังก์ชัน `imagefttext()` เพื่อวาดตัวอักษรโดยกำหนดให้มีขนาด 22 พอยต์ มีมุมมองค่าเท่ากับ 0 (ศูนย์) โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 20 บนแกน x และจุดพิกัด 180 บนแกน y แล้ววาดตัวอักษรสีดำมีชนิดตัวอักษรเป็นแบบ COUR.TTF โดยเอาตัวอักษรที่อยู่ในตัวแปร `$text` มาวาด (นั่นคือตัวอักษร Nakhon Ratchasima) ซึ่งเป็นการวาดตัวอักษรวางทับบนเงาที่ถูกวาดขึ้นโดยคำสั่งในบรรทัดที่ 31 โดยเลื่อนตำแหน่งมา 1 พิกัดบนแกน y (จาก 181 เป็น 180)

ช่วงบรรทัดที่ 36-37 นั้นเป็นการแสดงผลออกที่จอภาพ

บรรทัดที่ 36 เรียกใช้ฟังก์ชัน `header()` เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล .png

บรรทัดที่ 37 เรียกใช้ฟังก์ชัน `imagepng()` เพื่อแสดงรูปภาพสกุล .png ที่สร้างขึ้นโดยมีตัวชี้คือ `$img` ออกที่เว็บเบราว์เซอร์

ช่วงบรรทัดที่ 40 นั้นเป็นการทำลายหรือคืนพื้นที่ในการวาดรูปให้กับระบบ

บรรทัดที่ 40 เรียกใช้ฟังก์ชัน `imagedestroy()` เพื่อคืนพื้นที่ที่ตัวแปร `$img` ซึ้อยู่ให้กับระบบ

ตัวอย่างที่ 6 การใช้ฟังก์ชันในการวาดรูปโพลีกอน (จำนวนด้านมากกว่า 3 ด้าน) และสตริงในแนวนอนและตั้ง

```

1. <?php
2. // create a blank image
3. $img = imagecreate(400,200)
4.     or die("Cannot Initialize new GD image stream");
5.
6. // setup color for the polygon
7. $white = imagecolorallocate($img, 255, 255, 255);
8. $black = imagecolorallocate($img, 0, 0, 0);
9. $green = imagecolorallocate($img, 0, 255, 0);
10. $red = imagecolorallocate($img, 255, 0, 0);
11. $yellow = imagecolorallocate($img, 255, 255, 0);
12.
13. // draw on image: fill the background in White color
14. imagefill($img, 0, 0, $white);
15.
16. // draw on image: draw the rectangle and polygon
17. imagerectangle($img, 0, 0, 399, 199, $black);
18. $draw4 = array (10,10,100,100,300,100,390,10);
19. $draw4_in = array (100,100,300,100,390,10,190,10);
20. imagepolygon($img, $draw4, 4, $red);
21. imagefilledpolygon($img, $draw4_in, 4, $yellow);
22. imagepolygon($img, $draw4_in, 4, $green);
23.
24. // draw on image: draw string in various direction and line
25. imagestring($img, 5, 10, 170, "SUT --- THAILAND", $black);
26. imageline($img, 10, 190, 150, 190, $black);
27. imagestringup($img, 5, 360, 190, "SURANAREE", $black);
28. imagestringup($img, 3, 380, 190, "University of Technology", $black);
29.
30. // output image: output image to the browser
31. header("Content-type: image/png");
32. imagepng($img);
33.
34. // free memory
35. imagedestroy($img);
36. ?>
    
```



รูปที่ 7.14: วาดรูปโพลีกอนและสตริงในแนวนอนและตั้ง

ช่วงบรรทัดที่ 3-4 นั้นเป็นการกำหนดพื้นที่สำหรับการวาดรูป

บรรทัดที่ 3-4 ใช้ฟังก์ชัน `imagecreate()` สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้าง 400 พิกเซลและความสูงมีขนาดเท่ากับ 200 พิกเซล แล้วให้ตัวแปร `$img` เป็นตัวชี้ หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้นว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง `die()` เพื่อแสดงข้อความว่า "Cannot Initialize new GD image stream" ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

ช่วงบรรทัดที่ 7-11 นั้นเป็นการกำหนดสีที่ใช้ในการวาดรูป

บรรทัดที่ 7 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแฟงสี เก็บไว้ที่ตัวแปร `$white`

บรรทัดที่ 8 กำหนดสีดำที่ต้องการใช้ในการวาดรูปให้กับแผงสี เก็บไว้ที่ตัวแปร \$black

บรรทัดที่ 9 กำหนดสีเขียวที่ต้องการใช้ในการวาดรูปให้กับแผงสี เก็บไว้ที่ตัวแปร \$green

บรรทัดที่ 10 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแผงสี เก็บไว้ที่ตัวแปร \$red

บรรทัดที่ 11 กำหนดสีเหลืองที่ต้องการใช้ในการวาดรูปให้กับแผงสี เก็บไว้ที่ตัวแปร \$yellow

ช่วงบรรทัดที่ 14 นั้นเป็นการเติมสีให้กับพื้นที่ที่จะวาดรูป

บรรทัดที่ 14 เรียกใช้ฟังก์ชัน imagefill() เพื่อใส่สีพื้นเป็นสีขาว โดยเริ่มต้นที่ตำแหน่ง 0 (ศูนย์) บนแกน x และ 0 (ศูนย์) บนแกน y

ช่วงบรรทัดที่ 17-22 นั้นเป็นการวาดรูปสี่เหลี่ยมและโพลีกอน

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน imagerectangle() เพื่อวาดรูปสี่เหลี่ยมผืนผ้า โดยตำแหน่งบนซ้ายคือจุดพิกัด 0 บนแกน x และจุดพิกัด 0 บนแกน y และตำแหน่งล่างขวา คือจุดพิกัด 399 บนแกน x และจุดพิกัด 199 บนแกน y

บรรทัดที่ 18 กำหนดตัวแปร \$draw4 ให้เก็บข้อมูลอะเรย์ ประกอบไปด้วย 10, 10, 100, 100, 300, 100, 390 และ 10 ตามลำดับ เพื่อใช้ในการกำหนดจุดพิกัดในการวาดรูปโพลีกอน

บรรทัดที่ 19 กำหนดตัวแปร \$draw4_in ให้เก็บข้อมูลอะเรย์ ประกอบไปด้วย 100, 100, 300, 100, 390, 10, 190 และ 10 ตามลำดับ เพื่อใช้ในการกำหนดจุดพิกัดในการวาดรูปโพลีกอน

บรรทัดที่ 20 เรียกใช้ฟังก์ชัน imagepolygon() เพื่อวาดรูปโพลีกอนเส้นสีแดงประกอบไปด้วยด้าน 4 ด้าน โดยมีจุดพิกัดตำแหน่งต่าง ๆ อยู่ที่ตัวแปร \$draw4

บรรทัดที่ 21 เรียกใช้ฟังก์ชัน imagefilledpolygon() เพื่อเติมสีให้กับรูปโพลีกอนประกอบไปด้วยด้าน 4 ด้าน โดยมีจุดพิกัดตำแหน่งต่าง ๆ อยู่ที่ตัวแปร \$draw4_in แล้วเติมสีพื้นหลังให้เป็นสีเหลือง

บรรทัดที่ 22 เรียกใช้ฟังก์ชัน imagepolygon() เพื่อวาดรูปโพลีกอนเส้นสีเขียวประกอบไปด้วยด้าน 4 ด้าน โดยมีจุดพิกัดตำแหน่งต่าง ๆ อยู่ที่ตัวแปร \$draw4_in

ช่วงบรรทัดที่ 25-28 นั้นเป็นการวาดสตริงและเส้น

บรรทัดที่ 25 เรียกใช้ฟังก์ชัน imagestring() เพื่อวาดตัวอักษร SUT --- THAILAND กลุ่มที่ 5 สีดำ (กลุ่มมีทั้งหมด 5 กลุ่มคือ 1 ถึง 5) โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 10 บนแกน x และจุดพิกัด 170 บนแกน y

บรรทัดที่ 26 เรียกใช้ฟังก์ชัน imageline() เพื่อวาดเส้นสีดำในแนวอน โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 10 บนแกน x และจุดพิกัด 190 บนแกน y และไปสิ้นสุดที่ตำแหน่งจุดพิกัด 150 บนแกน x และจุดพิกัด 190 บนแกน y

บรรทัดที่ 27 เรียกใช้ฟังก์ชัน imagestringup() เพื่อวาดตัวอักษร SURANAREE กลุ่มที่ 5 สีดำในแนวตั้ง โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 360 บนแกน x และจุดพิกัด 190 บนแกน y

บรรทัดที่ 28 เรียกใช้ฟังก์ชัน imagestringup() เพื่อวาดตัวอักษร University of Technology กลุ่มที่ 3 สีดำในแนวตั้ง โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 380 บนแกน x และจุดพิกัด 190 บนแกน y

ช่วงบรรทัดที่ 31-32 นั้นเป็นการแสดงผลออกที่จอภาพ

บรรทัดที่ 31 เรียกใช้ฟังก์ชัน `header()` เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล `.png`

บรรทัดที่ 32 เรียกใช้ฟังก์ชัน `imagepng()` เพื่อแสดงรูปภาพสกุล `.png` ที่สร้างขึ้นโดยมีตัวชี้คือ `$img` ออกที่เว็บเบราว์เซอร์

ช่วงบรรทัดที่ 35 นั้นเป็นการทำลายหรือคืนพื้นที่ในการวาดรูปให้กับระบบ

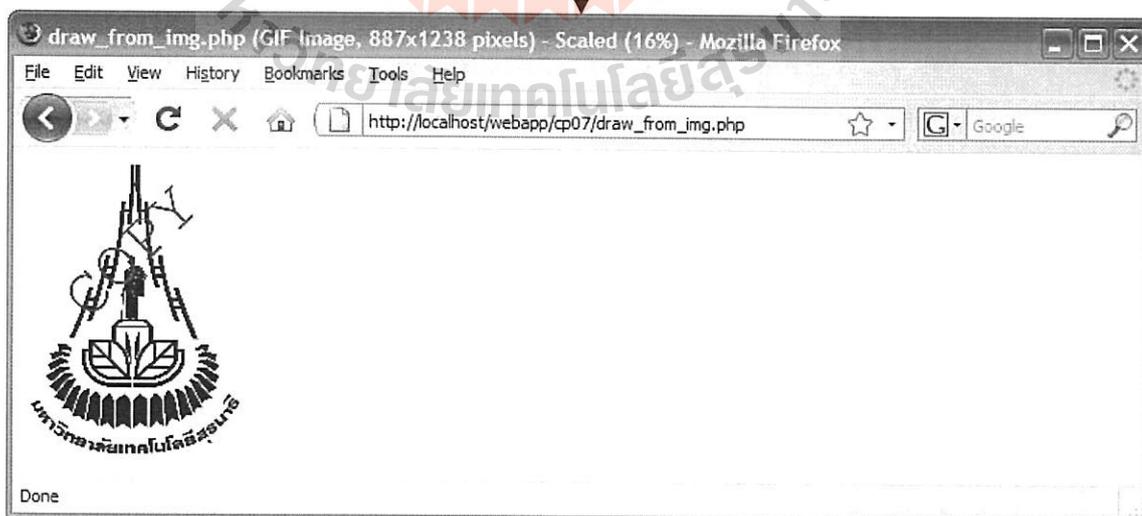
บรรทัดที่ 35 เรียกใช้ฟังก์ชัน `imagedestroy()` เพื่อคืนพื้นที่ที่ตัวแปร `$img` ซึ่งอยู่ให้กับระบบ

ตัวอย่างที่ 7 การสร้างและปรับแต่งรูปภาพจากแฟ้มรูปภาพ

```

1.  <?php
2.  // create a blank image
3.  $temp = imagecreate(10,10)
4.  or die("Cannot Initialize new GD image stream");
5.
6.  // open an image
7.  $img = @imagecreatefromgif ("logo_bw.gif");
8.  if (!$img) { /* See if it failed */
9.    $img = imagecreate(250, 20);
10.   $white = imagecolorallocate($img, 255, 255, 255);
11.   $red = imagecolorallocate($img, 255, 0, 0);
12.   imagestring($img, 5, 5, 5, "Error loading image !!!", $red);
13.  }
14.
15.  // setup a color for the polygon
16.  $red = imagecolorallocate($img, 255, 0, 0);
17.
18.  // the text to draw
19.  $text = "COPY";
20.  // Replace path by your own font path
21.
22.  $font = "COUR.TTF";
23.
24.  // draw text
25.  imagefttext($img, 200, 45, 250, 650, $red, $font, $text);
26.
27.  // output image: ouput the image to the browser and save it to file
28.  header("Content-type: image/gif");
29.  imagegif($img);
30.  imagegif($img, "logo_copy.gif");
31.
32.  // free memory
33.  imagedestroy($img);
34.  ?>

```



รูปที่ 7.15: การสร้างและปรับแต่งรูปภาพจากแฟ้มรูปภาพ

ช่วงบรรทัดที่ 3-4 นั้นเป็นการกำหนดพื้นที่สำหรับการวาดรูป

บรรทัดที่ 3-4 เรียกใช้ฟังก์ชัน `imagecreate()` สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้าง 10 พิกเซลและความสูงมีขนาดเท่ากับ 10 พิกเซล แล้วให้ตัวแปร `$temp` เป็นตัวชี้ หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้นว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว อีกทั้งยังมีการใช้คำสั่ง `die()` เพื่อแสดงข้อความว่า “Cannot Initialize new GD image stream” ในกรณีที่ไม่สามารถสร้างพื้นที่ได้

หมายเหตุ ต้องทำการกำหนดพื้นที่ในการวาดก่อน แล้วจึงสามารถเปิดรูปที่ต้องการจัดการได้

ช่วงบรรทัดที่ 7-13 นั้นเป็นการเปิดเพิ่มรูปภาพและกำหนดสีที่ใช้ในการวาดสติริงแจ้งข้อผิดพลาด

บรรทัดที่ 7 เรียกใช้ฟังก์ชัน `imagecreatefromgif` เพื่อเปิดเพิ่มรูปภาพชื่อว่า `logo_bw.gif`

บรรทัดที่ 8 ทดสอบว่าเปิดเพิ่มรูปภาพดังกล่าวได้สำเร็จหรือไม่ ถ้าเปิดไม่ได้จะเข้าไปทำคำสั่งในบรรทัดที่ 9-12 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 16

บรรทัดที่ 9 เรียกใช้ฟังก์ชัน `imagecreate()` สร้างพื้นที่ที่จะวาดรูป โดยกำหนดให้ความกว้าง 250 พิกเซลและความสูงมีขนาดเท่ากับ 20 พิกเซล แล้วให้ตัวแปร `$img` เป็นตัวชี้หรืออาจเรียกอีกอย่างหนึ่งเพื่อให้เข้าใจได้ง่ายขึ้นว่าเป็นตัวเก็บรายละเอียดของข้อมูลดังกล่าว

บรรทัดที่ 10 กำหนดสีขาวที่ต้องการใช้ในการวาดรูปให้กับแอสกี เก็บไว้ที่ตัวแปร `$white` เพื่อให้เป็นสีพื้นหลังของพื้นที่วาดรูปที่กำหนด

หมายเหตุ คำสั่งแรกที่มีการกำหนดสีนั้น สีที่กำหนดขึ้นมานั้นจะกลายเป็นสีพื้นหลังให้กับพื้นที่โดยอัตโนมัติ

บรรทัดที่ 11 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแอสกี เก็บไว้ที่ตัวแปร `$red`

บรรทัดที่ 12 เรียกใช้ฟังก์ชัน `imagestring()` เพื่อวาดตัวอักษร Error loading image !!! กลุ่มที่ 5 สีแดงในแนวนอน (กลุ่มมีทั้งหมด 5 กลุ่มคือ 1 ถึง 5) โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 5 บนแกน x และจุดพิกัด 5 บนแกน y

หมายเหตุ ถ้าไม่สามารถเปิดเพิ่มรูปภาพได้จะปรากฏดังในภาพที่ 7.16



รูปที่ 7.16: การแสดงข้อผิดพลาดเมื่อไม่สามารถเปิดเพิ่มรูปภาพได้

ช่วงบรรทัดที่ 16-22 นั้นเป็นการกำหนดสี กลุ่มข้อความ ชนิดตัวอักษรและวาดตัวอักษร

บรรทัดที่ 16 กำหนดสีแดงที่ต้องการใช้ในการวาดรูปให้กับแอสกี เก็บไว้ที่ตัวแปร `$red`

บรรทัดที่ 19 กำหนดตัวแปร `$text` ให้เก็บข้อมูลสติริง COPY

บรรทัดที่ 22 กำหนดตัวแปร `$font` ให้เก็บข้อมูลสติริง COUR.TTF ซึ่งเป็นชนิดตัวอักษรชนิดหนึ่ง

บรรทัดที่ 25 เรียกใช้ฟังก์ชัน `imagefttext()` เพื่อวาดตัวอักษรโดยกำหนดให้มีขนาด 200 พอยต์ มีมุมมองเท่ากับ 45 องศา โดยเริ่มต้นที่ตำแหน่งจุดพิกัด 250 บนแกน x และจุดพิกัด 650 บนแกน y

แล้ววาดตัวอักษรสีแดงมีชนิดตัวอักษรเป็นแบบ COUR.TTF โดยเอาตัวอักษรที่อยู่ในตัวแปร \$text มาวาด (นั่นคือตัวอักษร COPY)

ช่วงบรรทัดที่ 28-30 นั้นเป็นการแสดงผลออกที่จอภาพและบันทึกลงแฟ้ม

บรรทัดที่ 28 เรียกใช้ฟังก์ชัน header() เพื่อบอกเว็บเบราว์เซอร์ว่าข้อมูลต่อไปจะเป็นรูปภาพสกุล .gif

บรรทัดที่ 29 เรียกใช้ฟังก์ชัน imagegif() เพื่อแสดงรูปภาพสกุล .gif ที่สร้างขึ้นโดยมีตัวชี้คือ \$img ออกที่เว็บเบราว์เซอร์

บรรทัดที่ 30 เรียกใช้ฟังก์ชัน imagegif() อีกครั้งเพื่อทำการบันทึกรูปภาพที่สร้างขึ้นโดยมีตัวชี้คือ \$img ลงแฟ้มข้อมูลชื่อว่า logo_copy.gif

หมายเหตุ ถ้าไม่มีคำสั่งบรรทัดที่ 29 (มีเฉพาะคำสั่งในบรรทัดที่ 30) รูปภาพจะไม่ถูกแสดงออกที่เบราว์เซอร์

ช่วงบรรทัดที่ 33 นั้นเป็นการทำลายหรือคืนพื้นที่ในการวาดรูปให้กับระบบ

บรรทัดที่ 33 เรียกใช้ฟังก์ชัน imagedestroy() เพื่อคืนพื้นที่ที่ตัวแปร \$img ซ้ำให้กับระบบ

สรุป

การใช้ฟังก์ชันในการรวมแฟ้มข้อมูลต่าง ๆ เข้าด้วยกัน ไม่ว่าจะเป็นฟังก์ชัน include(), include_once, require() หรือ require_once() ต่างก็ทำให้สะดวกต่อการแก้ไขและปรับปรุงข้อมูล ส่วนฟังก์ชันที่ช่วยในการวาดรูปก็ให้อำนวยความสะดวกให้นักเขียนโปรแกรมสามารถประยุกต์และนำไปใช้ในการแสดงผลในรูปแบบต่าง ๆ ได้ เช่น แผนภูมิ แผนภาพ เป็นต้น โดยการวาดรูปแต่ละครั้งนั้น จะมีลำดับขั้นตอนที่สำคัญอยู่ 4 ขั้นตอนคือ

ขั้นตอนที่ 1 กำหนดพื้นที่ (Canvas) และสีสำหรับการวาดรูป โดยสีที่กำหนดเป็นสีแรกจะถูกนำไปเป็นสีพื้นหลังโดยอัตโนมัติ

ขั้นตอนที่ 2 วาดรูปตามที่ต้องการ รูปภาพต่าง ๆ ที่วาดออกมาจะถูกวางทับกันเป็นชั้น ๆ บนพื้นที่วาดรูป นั่นคือรูปภาพแรกจะอยู่ล่างสุด ส่วนรูปสุดท้ายจะอยู่บนสุด

ขั้นตอนที่ 3 แสดงรูปภาพออกที่จอภาพหรือบันทึกเป็นแฟ้มสกุลต่าง ๆ

ขั้นตอนที่ 4 คืนหน่วยความจำที่ใช้ในการวาดรูปให้กับระบบ

แบบฝึกหัด

1. ปรับปรุงไฟล์ html ทุก ๆ ไฟล์ (home.html, office_hour.html, like_dislike.html, images.html, links.html, guestbook.html และ View Guestbook) โดยตัดส่วนที่เหมือนกันออกแล้วนำไปบันทึกเป็นแฟ้ม menu.inc หลังจากนั้นให้ทำการรวมแฟ้มข้อมูล menu.inc เข้ากับแฟ้มข้อมูล html เดิม (อย่าลืมบันทึกแฟ้มสกุล .html ที่มีการรวมเอาแฟ้ม menu.inc เข้าไปให้เป็นสกุล .php)

2. เขียนโปรแกรมในการวาดรูปธงชาติของประเทศไทย ขนาดกว้าง 200 และยาว 120 พิกเซล ดังแสดงในรูปที่ 7.17



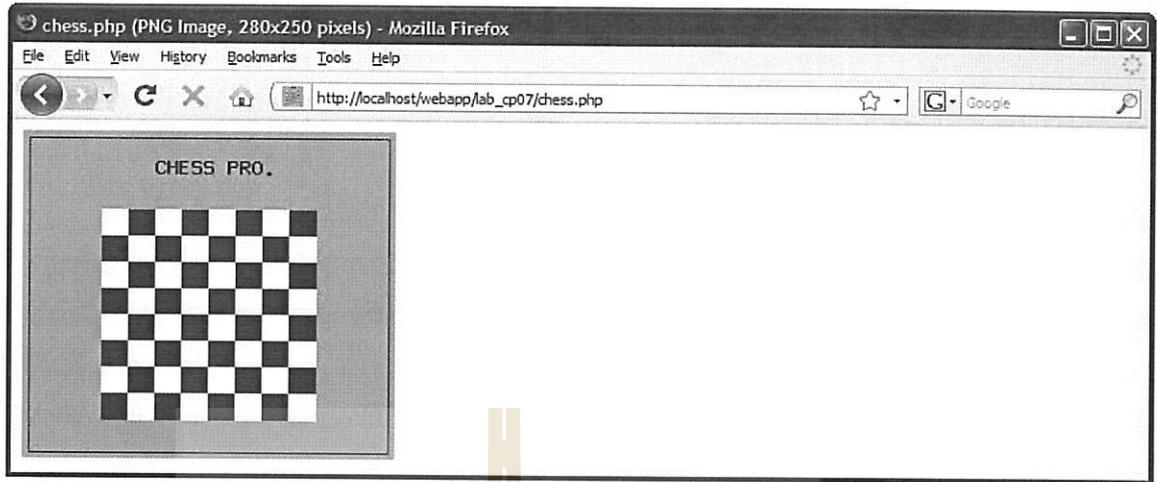
รูปที่ 7.17: วาดรูปธงชาติไทย

3. เขียนโปรแกรมในการวาดลายเส้นบนพื้นที่วาดรูปขนาด 201 x 201 พิกเซล ให้ได้ผลลัพธ์ดังแสดงในรูปที่ 7.18



รูปที่ 7.18: วาดรูปลายเส้น

4. เขียนโปรแกรมในการวาดรูปของกระดานหมากรุก ให้ช่องทางเดินแต่ละช่องมีสีแดงสลับสีขาวขนาด 20 x 20 พิกเซล บนพื้นที่วาดรูปสี่เหลี่ยมขนาด 280 x 250 พิกเซล ดังแสดงในรูปที่ 7.19



รูปที่ 7.19: วาดรูปตารางหมากรุก

5. คะแนนของนักศึกษา 5 คนมีดังนี้คือ นาย A ได้ 90, นาย B ได้ 80, นาย C ได้ 95, นาย D ได้ 73 และนาย E ได้ 84 จงเขียนโปรแกรมในการวาดรูปกราฟแท่งแสดงกราฟของนักศึกษาทั้ง 5 และเส้นคะแนนเฉลี่ยของบนพื้นที่วาดรูปขนาด 250 x 200 พิกเซล ให้ได้ผลลัพธ์ดังแสดงในรูปที่ 7.20



รูปที่ 7.20: วาดรูปกราฟแท่ง

บทที่ 8 แนะนำ MySQL

<Introduction to MySQL>

บทนำ

การจัดเก็บข้อมูลลงแฟ้มข้อมูลที่ได้อีกแล้วในบทที่ผ่านมา เป็นวิธีหนึ่งที่สามารถจัดเก็บข้อมูลที่ต้องการได้ แต่อีกวิธีการหนึ่งคือ การจัดเก็บข้อมูลลงระบบฐานข้อมูล (Database System) ซึ่งวิธีการนี้จะช่วยให้การบริหารและจัดการกับข้อมูล เช่น การค้นหา การเรียงลำดับ รวมไปถึงการดึงข้อมูลเพื่อประมวลผลหรือแสดงผล ต่างก็มีประสิทธิภาพมากยิ่งขึ้น ระบบฐานข้อมูลเป็นระบบสำหรับผู้ใช้หลายคน (Multi-user) โดยใช้ภาษาสอบถามที่เป็นโครงสร้างเรียกว่า SQL (Structured Query Language) ซึ่งเป็นภาษาสำหรับใช้ในการสอบถามของระบบฐานข้อมูลที่นิยมใช้กันอย่างแพร่หลาย ดังนั้นในส่วนของบทนี้ จึงขอแนะนำการใช้งานเบื้องต้นของระบบฐานข้อมูล MySQL และภาษาสอบถาม SQL

วัตถุประสงค์

1. เพื่อศึกษารูปแบบการทำงานของระบบฐานข้อมูล MySQL และคำสั่งต่าง ๆ ในการติดต่อและสอบถามกับระบบ
2. เพื่อทดลองใช้คำสั่งต่าง ๆ ในการติดต่อกับระบบฐานข้อมูล สร้างฐานข้อมูล และรายละเอียดต่าง ๆ
3. เพื่อประยุกต์การใช้คำสั่งสอบถามในรูปแบบต่าง ๆ

เนื้อหา

ระบบฐานข้อมูล MySQL เป็นระบบฐานข้อมูลที่นิยมใช้กันอย่างแพร่หลาย มีลักษณะเป็นฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database) ที่มีการจัดเก็บข้อมูลให้ในรูปแบบของตารางแทนการจัดเก็บลงแฟ้มข้อมูลแฟ้มเดียว โดยตารางต่าง ๆ สามารถติดต่อหรือเชื่อมโยงกันได้ ทำให้การเข้าถึงข้อมูลมีความรวดเร็วขึ้น มีการดึงข้อมูลที่ถูกสอบถามได้ง่าย เนื่องจากมีการเข้าถึงข้อมูลแบบสุ่ม มีระบบการจัดการการเข้าถึงข้อมูลของผู้ใช้แต่ละคนได้อย่างมีประสิทธิภาพ โดยใช้ภาษาสอบถามที่เป็นโครงสร้างข้อมูลเรียกว่า SQL (Structured Query Language) เป็นตัวกลาง

เมื่อลงโปรแกรม Appserv แล้ว จะมีโปรแกรมช่วยในการจัดการระบบฐานข้อมูล MySQL เรียกว่า phpMyAdmin ซึ่งเป็นโปรแกรมลักษณะกราฟิกที่ใช้งานง่าย เพียงแต่กดปุ่มหรือพิมพ์ข้อมูลลงไปเท่านั้น ไม่จำเป็นต้องมีความรู้ในเรื่องของการใช้คำสั่งต่าง ๆ เช่น คำสั่งในการสร้างฐานข้อมูล คำสั่งในการกำหนดชนิดข้อมูลในตาราง เป็นต้น แต่การใช้โปรแกรม phpMyAdmin นี้เหมาะสำหรับผู้ที่มีความชำนาญในการใช้ภาษาสอบถามแล้ว จึงไม่เหมาะสมกับผู้ที่เริ่มต้นใช้หรือเริ่มฝึกการเขียนโปรแกรม PHP เพื่อติดต่อฐานข้อมูล เนื่องจากมีความจำเป็นต้องฝึกเขียนและใช้คำสั่งต่าง ๆ ก่อน จะได้นำประสบการณ์เหล่านั้นไปพัฒนาต่อได้

ง่ายและสะดวกขึ้น ดังนั้นในบทเรียนนี้จึงขอแนะนำการเข้าถึงระบบฐานข้อมูลผ่าน Command Prompt ซึ่งจะกล่าวในหัวข้อที่ 8.2

8.1 แนวคิดเกี่ยวกับฐานข้อมูลเชิงสัมพันธ์

การเก็บข้อมูลความหมายของคำสำคัญต่าง ๆ ที่เกี่ยวข้องกับฐานข้อมูลเชิงสัมพันธ์ ประกอบไปด้วยตาราง (Table), คอลัมน์ (Column), แถว (Row), ค่า (Value), คีย์ (Key), แบบแผน (Schemas), และความสัมพันธ์ (Relationship) โดยมีรายละเอียดดังต่อไปนี้

ตาราง มีลักษณะคล้ายกับตารางในโปรแกรม Microsoft Excel ซึ่งใช้ในการเก็บข้อมูลต่าง ๆ เช่นดังตัวอย่างในรูปที่ 8.1 ที่ตารางมีชื่อว่า student แต่ละคอลัมน์ก็เก็บข้อมูลต่าง ๆ ส่วนแต่ละแถวนั้นก็เป็นข้อมูลของนักศึกษาคนหนึ่ง

student

studentid	name	age	earn
0001	Somchai Jaidee	21	85
0002	Jaidee Yoosamor	20	78
0003	Sombat Saitalord	20	88
0004	Onyon Tamvela	22	81
0005	Somporn Tampark	21	80

รูปที่ 8.1: ข้อมูลในตาราง student

คอลัมน์ (Column) แต่ละคอลัมน์ในตารางนั้นจะมีชื่อเฉพาะและมีข้อมูลที่แตกต่างกัน เช่นดังในรูปที่ 8.1 นั้นมีทั้งหมด 4 คอลัมน์ ประกอบไปด้วย studentid, name, age และ earn ในคอลัมน์ studentid เก็บข้อมูลเป็นเลขจำนวนเต็ม ส่วนคอลัมน์ name เก็บสตริง คอลัมน์ age และคอลัมน์ earn เก็บข้อมูลเป็นเลขจำนวนเต็ม บางครั้งคอลัมน์อาจถูกเรียกว่า ฟิลด์หรือแอททริบิวต์

แถว (Row) แต่ละแถวในตารางนั้นเรียกว่าเรคอร์ด ซึ่งเก็บข้อมูลของนักศึกษาคนหนึ่ง

ค่า (Value) ข้อมูลที่ถูกเก็บในแถวหนึ่ง ๆ นั้นมีหลายค่า ซึ่งค่าแต่ละค่านี้จะขึ้นอยู่กับชนิดของคอลัมน์ เช่น คอลัมน์นั้นเป็นสตริง ค่าในแถวที่อยู่ตรงกับคอลัมน์นั้นก็จะเป็นสตริงด้วย

คีย์ (Keys) ค่าที่ถูกเก็บอยู่ในคอลัมน์ name ซึ่งคือชื่อของนักศึกษานั้น บางครั้งอาจเหมือนกัน อีกทั้งยังมีความเป็นไปได้ว่า นักศึกษาทั้ง 2 ที่มีชื่อเหมือนกันนั้น ต่างก็มีอายุ 20 ปีเช่นเดียวกัน หรือยิ่งไปกว่านั้นคือ ต่างก็มีหน่วยกิตเท่ากันอีก เมื่อเป็นเช่นนั้นแล้ว การแยกแยะข้อมูลของนักศึกษาเพื่อนำไปประมวลผลต่อแล้วให้ได้ผลลัพธ์ที่ถูกต้องและเป็นจริง คงทำได้ยากมาก วิธีการที่จะช่วยแยกแยะนักศึกษาแต่ละคนออกจากกันโดยสิ้นเชิงนั้นคือการกำหนดคีย์ให้กับนักศึกษาทุก ๆ คน ซึ่งคีย์เหล่านั้นอาจจะเป็นเลขจำนวนเต็มหรือสตริงก็ได้ ขึ้นอยู่กับผู้ออกแบบ โดยค่าคีย์แต่ละคีย์นั้นจะต้องเป็นค่าที่ไม่ซ้ำกัน ค่าคีย์นี้เรียกว่า คีย์หลัก (Primary Key) ในฐานข้อมูลนั้นส่วนใหญ่แล้วจะประกอบไปด้วยตารางหลาย ๆ ตาราง แล้วใช้คีย์หรือคีย์หลักในการเชื่อมต่อข้อมูลต่าง ๆ เช่นดังในรูปที่ 8.2 ได้กำหนดตารางเพิ่มขึ้นอีก 1 ตาราง ชื่อว่า register ซึ่งประกอบไปด้วยคอลัมน์ 5 คอลัมน์ คือ no, studentid, register, date และ paid โดยแถวแต่ละแถวในตาราง register เก็บข้อมูลของนักศึกษาเกี่ยวกับการจำนวนหน่วยกิตที่ลงทะเบียน วันที่ลงทะเบียน และการจ่ายเงิน (0 ยังไม่จ่ายเงิน ส่วน 1 คือจ่ายเงินแล้ว) จะสังเกตเห็นว่าสามารถเชื่อมต่อข้อมูลทั้ง 2 ตารางเข้าหากันได้ โดยการเปรียบเทียบคีย์หลักในตาราง student คือ studentid กับคีย์รอง

(Foreign Key) ในตาราง คือ studentid เช่น สามารถบอกได้ว่านักศึกษารหัส 0002 ชื่อ Jaidee Yoosamor อายุ 20 ปี มีหน่วยกิตสะสมเท่ากับ 78 หน่วยกิต ตอนนี้อยู่ลงทะเบียนเมื่อวันที่ 15 มิถุนายน 2550 จำนวน 18 หน่วยกิต แต่ยังไม่ได้จ่ายค่าหน่วยกิตที่ลงทะเบียน

student

studentid	name	age	earn
0001	Somchai Jaidee	21	85
0002	Jaidee Yoosamor	20	78
0003	Sombat Saitalord	20	88
0004	Onyon Tamvela	22	81
0005	Somporn Tampark	21	80

register

no	studentid	register	date	paid
1	0002	18	2007-06-15	0
2	0005	18	2007-06-16	0
3	0001	21	2007-06-16	1
4	0003	15	2007-06-18	1
5	0004	16	2007-06-20	0

รูปที่ 8.2: ข้อมูลในตาราง student และ register

แบบแผน (Schemas) แบบโครงสร้างของตารางทั้งหมดที่จะใช้ในระบบฐานข้อมูล มีลักษณะคล้ายกับพิมพ์เขียว ในทุก ๆ ตารางจะมีการระบุคีย์หลักและคีย์รอง รวมไปถึงชนิดข้อมูลในการจัดเก็บข้อมูลของแต่ละคอลัมน์ในตารางนั้น ๆ อาจใช้ไดอะแกรมในการนำเสนอหรือรูปแบบของข้อความ เช่น

```
student(studentid, name, age, earn)
register(no, studentid, register, date, paid)
```

ความสัมพันธ์ (Relationship) คีย์รองเป็นตัวบ่งบอกถึงความสัมพันธ์ระหว่างตาราง เช่น แถวในตาราง Register มีความสัมพันธ์กับแถวในตาราง student โดยความสัมพันธ์ในฐานข้อมูลเชิงสัมพันธ์นั้นมี 3 ลักษณะด้วยกันคือ

1. **ความสัมพันธ์แบบ One-to-One** เป็นความสัมพันธ์ที่เกิดขึ้นระหว่างแถวข้อมูลจากตาราง หนึ่งกับแถวข้อมูลในอีกตาราง เช่น ข้อมูลของการลงทะเบียนของนักศึกษาในตาราง register มีความสัมพันธ์แบบ One-to-One กับข้อมูลในตาราง student โดยใช้ studentid ในตาราง register เป็นคีย์สำรอง ดังแสดงในรูปที่ 8.3 นั้น

student

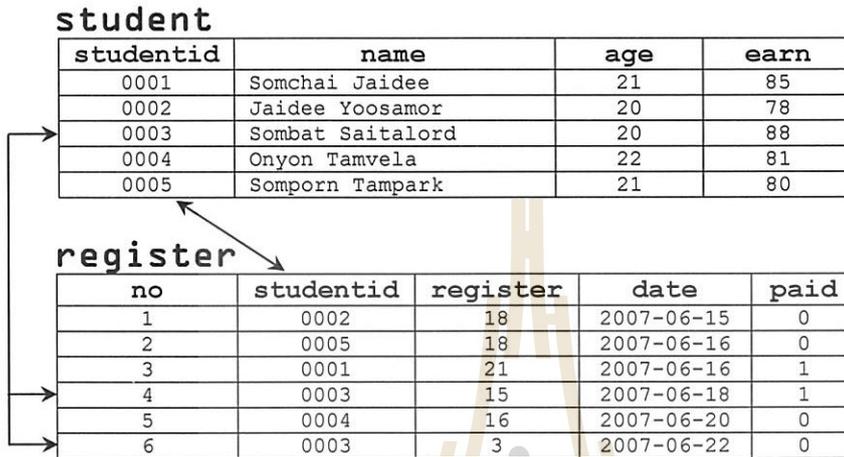
studentid	name	age	earn
0001	Somchai Jaidee	21	85
0002	Jaidee Yoosamor	20	78
0003	Sombat Saitalord	20	88
0004	Onyon Tamvela	22	81
0005	Somporn Tampark	21	80

register

no	studentid	register	date	paid
1	0002	18	2007-06-15	0
2	0005	18	2007-06-16	0
3	0001	21	2007-06-16	1
4	0003	15	2007-06-18	1
5	0004	16	2007-06-20	0

รูปที่ 8.3: ความสัมพันธ์แบบ One-to-One

2. ความสัมพันธ์แบบ One-to-Many เป็นความสัมพันธ์ที่เกิดขึ้นระหว่างแถวข้อมูลจากตาราง ๑ หนึ่งกับแถวข้อมูลหลายแถวในอีกตาราง เช่น จากตาราง register ในรูปที่ 8.3 นั้น นักศึกษาที่มี studentid เท่ากับ 0003 ได้ลงทะเบียน 15 หน่วยกิตในวันที่ 18 มิถุนายน 2550 ต่อมาในวันที่ 22 ได้ลงทะเบียนอีก 3 หน่วยกิต แต่ยังไม่จ่ายค่าหน่วยกิตที่ลงทะเบียนเพิ่มในครั้งนี้ ผลที่เกิดขึ้นจะได้อ้างอิงในรูปที่ 8.4

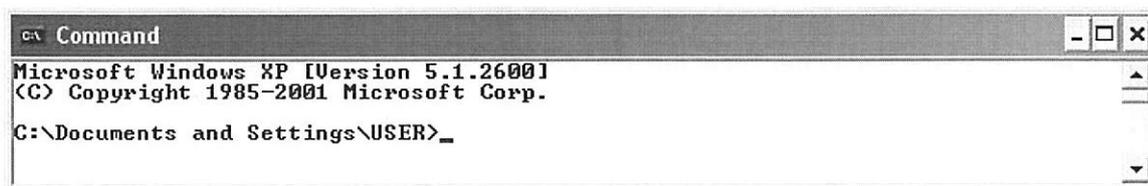


รูปที่ 8.4: ความสัมพันธ์แบบ One-to-Many

3. ความสัมพันธ์แบบ Many-to-Many เป็นความสัมพันธ์ที่เกิดจากแถวหลาย ๆ แถวในตารางหนึ่ง กับแถวหลาย ๆ แถวในอีกตาราง เช่น ตารางที่เก็บชื่อนักแต่งหนังสือ และตารางที่เก็บชื่อหนังสือ สมมุติว่านาย A, B และ C เขียนหนังสือร่วมกันเล่มหนึ่ง ซึ่งแต่ละคนก็ยังมีหนังสือที่เขียนคนเดียวหรือเขียน 2 คนอีกหลายเล่ม ดังนั้นความสัมพันธ์ทั้ง 2 ตารางจึงเป็นแบบ Many-to-Many เนื่องจากมีแถวหลาย ๆ แถวในตารางที่เก็บชื่อนักแต่งหนังสือ มีความสัมพันธ์กันกับแถวหลาย ๆ แถวในตารางที่เก็บชื่อหนังสือ

8.2 การติดต่อระบบฐานข้อมูล MySQL ผ่าน Command Prompt

โปรแกรม Appserv ได้ถูกเลือกให้ลงในไดรว์ D: และกำหนดชื่อผู้ใช้งานข้อมูลคือ root และรหัสผ่านคือ meroot ดังนั้นเมื่อเปิด Console (โดยการกดปุ่ม Start → Programs → Accessories → Command Prompt) ขึ้นมาจะปรากฏดังรูปที่ 8.5



รูปที่ 8.5: หน้าต่างของ Command Prompt

ต่อไปต้องพิมพ์คำสั่งเปลี่ยนไดรว์จาก C เป็น D โดยพิมพ์ d: แล้วกดปุ่ม Enter จะทำให้เปลี่ยนสถานะการใช้งานจากไดรว์ C เป็น D ต่อจากนั้นก็เปลี่ยนไดเรกทอรีไปที่ MySQL โดยพิมพ์คำสั่ง ดังแสดงในรูปที่ 8.6

```

c:\ Command
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\USER>d:
D:\>cd appserv\mysql
D:\AppServ\MySQL>_
    
```

รูปที่ 8.6: หน้าต่างของการเปลี่ยนไดเรกทอรีไปที่ MySQL

เมื่ออยู่ที่ MySQL แล้ว ทดลองพิมพ์คำสั่ง `mysql` เพื่อเข้าสู่ระบบฐานข้อมูล จะปรากฏข้อความ ERROR ขึ้นดังแสดงในรูปที่ 8.7 เนื่องจากตอนลงโปรแกรม Appserv ได้กำหนดชื่อผู้ใช้คือ `root` รหัสผ่านคือ `meroot`

```

c:\ Command
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\USER>d:
D:\>cd appserv\mysql
D:\AppServ\MySQL>mysql
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: N
O)
D:\AppServ\MySQL>_
    
```

รูปที่ 8.7: หน้าต่างของการแสดงข้อผิดพลาดเมื่อเข้าสู่ระบบฐานข้อมูล

ดังนั้นต้องใช้คำสั่ง `mysql -u root -p` แล้วกดปุ่ม Enter ก็จะถามให้ใส่ Password ให้ใส่คำว่า `meroot` แล้วกดปุ่ม Enter ก็จะสามารถเข้าสู่ระบบฐานข้อมูล MySQL ได้แล้ว ดังแสดงผลในรูปที่ 8.8

หมายเหตุ การเข้าสู่ระบบฐานข้อมูล MySQL นั้น สามารถเข้าจากตำแหน่งใดหรือใครก็ได้

```

c:\ Command - mysql -u root -p
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\USER>d:
D:\>cd appserv\mysql
D:\AppServ\MySQL>mysql
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: N
O)
D:\AppServ\MySQL>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 5.0.27-community-nt-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> _
    
```

รูปที่ 8.8: หน้าต่างของการแสดงผลเมื่อสามารถเข้าสู่ระบบฐานข้อมูล MySQL

8.3 ภาษาสอบถามข้อมูลเชิงโครงสร้าง

ภาษาสอบถามข้อมูลเชิงโครงสร้างหรือที่เรียกว่า SQL (Structured Query Language) นั้น เป็นภาษามาตรฐานที่นิยมใช้กันอย่างแพร่หลายกับระบบฐานข้อมูลเชิงสัมพันธ์ เช่น MySQL, Oracle, PostgreSQL, Sybase, Microsoft SQL Server เป็นต้น เพื่อทำการบันทึก ปรับปรุง และดึงข้อมูลจาก ฐานข้อมูล สามารถรองรับการทำงานแบบเป็นภาษากำหนดนิยาม (Data Definition Languages: DDL) และภาษาปรับแต่งข้อมูล (Data Manipulation Languages: DML) ได้

8.3.1 ภาษากำหนดนิยามข้อมูล (Data Definition Languages: DDL)

ภาษากำหนดนิยามข้อมูลใช้ในการสร้างฐานข้อมูลและกำหนดรูปแบบของโครงสร้างของข้อมูล ประกอบไปด้วยการกำหนดรูปแบบของตารางต่าง ๆ เช่น ชื่อคอลัมน์ ชนิดข้อมูล เป็นต้น รวมไปถึงการ กำหนดสิทธิ์ในการใช้ฐานข้อมูลด้วย

การสร้างฐานข้อมูล การสร้างฐานข้อมูลสามารถทำได้หลายวิธี เช่น อาจใช้โปรแกรม phpMyAdmin หรือใช้คำสั่งใน PHP หรือใช้ MySQL Console ซึ่งในช่วงนี้ขอแนะนำการใช้ MySQL Console ก่อน โดยใช้คำสั่ง SQL ในการสร้างฐานข้อมูลและแสดงฐานข้อมูลทั้งหมดในระบบ ดังนี้คือ

- คำสั่ง **CREATE** ใช้สร้างฐานข้อมูลหรือตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

การสร้างฐานข้อมูล

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_specification [, create_specification] ...]
create_specification:
    [DEFAULT] CHARACTER SET charset_name
    | [DEFAULT] COLLATE collation_name;
```

การสร้างตาราง

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_option ...]
```

หรือ

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_option ...]
    select_statement
```

หรือ

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    { LIKE old_tbl_name | (LIKE old_tbl_name) }
```

ทางเลือกต่าง ๆ ที่สามารถใช้ในคำสั่ง CREATE ได้ มีดังต่อไปนี้

นิยามการสร้าง

```
create_definition:
    column_definition
    | [CONSTRAINT [symbol]] PRIMARY KEY [index_type]
        (index_col_name,...)
    | {INDEX|KEY} [index_name] [index_type]
        (index_col_name,...)
    | [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
        [index_name] [index_type] (index_col_name,...)
    | {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name]
```

```
(index_col_name, ...)
| [CONSTRAINT [symbol]] FOREIGN KEY
    [index_name] (index_col_name, ...)
    [reference_definition]
| CHECK (expr)
```

นิยามของคอลัมน์

```
column_definition:
    col_name data_type [NOT NULL | NULL]
    [DEFAULT default_value]
    [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
    [COMMENT 'string'] [reference_definition]
```

ชนิดข้อมูล

```
data_type:
    TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length, decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length, decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length, decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL(length, decimals) [UNSIGNED] [ZEROFILL]
| NUMERIC(length, decimals) [UNSIGNED] [ZEROFILL]
| DATE
| TIME
| TIMESTAMP
| DATETIME
| YEAR
| CHAR(length) [BINARY | ASCII | UNICODE]
| VARCHAR(length) [BINARY]
| BINARY(length)
| VARBINARY(length)
| TINYBLOB
| BLOB
| MEDIUMBLOB
| LONGBLOB
| TINYTEXT [BINARY]
| TEXT [BINARY]
| MEDIUMTEXT [BINARY]
| LONGTEXT [BINARY]
| ENUM(value1, value2, value3, ...)
| SET(value1, value2, value3, ...)
| spatial_type
```

ชื่อตัวชี้คอลัมน์

```
index_col_name:
    col_name [(length)] [ASC | DESC]
```

นิยามตัวอ้างอิง

```
reference_definition:
    REFERENCES tbl_name [(index_col_name, ...)]
    [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
    [ON DELETE reference_option]
    [ON UPDATE reference_option]
```

ทางเลือกตัวอ้างอิง

```
reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION
```

ทางเลือกตาราง

```
table_option:
    {ENGINE|TYPE} [=] engine_name
    | AUTO_INCREMENT [=] value
    | AVG_ROW_LENGTH [=] value
    | [DEFAULT] CHARACTER SET charset_name
    | CHECKSUM [=] {0 | 1}
    | COLLATE collation_name
    | COMMENT [=] 'string'
    | CONNECTION [=] 'connect_string'
    | DATA DIRECTORY [=] 'absolute path to directory'
    | DELAY_KEY_WRITE [=] {0 | 1}
    | INDEX DIRECTORY [=] 'absolute path to directory'
    | INSERT_METHOD [=] { NO | FIRST | LAST }
    | MAX_ROWS [=] value
    | MIN_ROWS [=] value
    | PACK_KEYS [=] {0 | 1 | DEFAULT}
    | PASSWORD [=] 'string'
    | ROW_FORMAT [=] {DEFAULT | DYNAMIC | FIXED |
        COMPRESSED | REDUNDANT | COMPACT}
    | UNION [=] (tbl_name[,tbl_name]...)
```

- คำสั่ง SHOW ใช้แสดงข้อมูลของฐานข้อมูลหรือตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

```
SHOW [FULL] COLUMNS FROM tbl_name [FROM db_name] [LIKE
    'pattern']
SHOW CREATE DATABASE db_name
SHOW CREATE FUNCTION funcname
SHOW CREATE PROCEDURE procname
SHOW CREATE TABLE tbl_name
SHOW DATABASES [LIKE 'pattern']
SHOW ENGINE engine_name {LOGS | STATUS }
SHOW [STORAGE] ENGINES
SHOW ERRORS [LIMIT [offset,] row_count]
SHOW FUNCTION STATUS [LIKE 'pattern']
SHOW GRANTS FOR user
SHOW INDEX FROM tbl_name [FROM db_name]
SHOW INNODB STATUS
SHOW PROCEDURE STATUS [LIKE 'pattern']
SHOW [BDB] LOGS
SHOW PRIVILEGES
SHOW [FULL] PROCESSLIST
SHOW [GLOBAL | SESSION] STATUS [LIKE 'pattern']
SHOW TABLE STATUS [FROM db_name] [[LIKE 'pattern']
SHOW [OPEN] TABLES [FROM db_name] [LIKE 'pattern']
SHOW TRIGGERS
SHOW [GLOBAL | SESSION] VARIABLES [LIKE 'pattern']
SHOW WARNINGS [LIMIT [offset,] row_count]
```

แสดงข้อมูลของฐานข้อมูล
SHOW DATABASES ;

แสดงข้อมูลของตาราง
SHOW TABLES ;

- คำสั่ง DESCRIBE หรือ DESC ใช้แสดงรายละเอียดของโครงสร้างตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

{DESCRIBE | DESC} tbl_name [col_name | wild]

ตัวอย่าง การใช้คำสั่ง CREATE DATABASE สร้างฐานข้อมูลชื่อว่า webapp150 แล้วแสดงฐานข้อมูลทั้งหมดในระบบ

```

C:\ Command - mysql -u root -p
D:\AppServ\MySQL>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.27-community-nt-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database webapp150;
Query OK, 1 row affected (0.03 sec)

mysql> _
    
```

รูปที่ 8.9: หน้าต่างของการแสดงผลเมื่อสร้างฐานข้อมูล webapp150

```

C:\ Command - mysql -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| phpmyadmin |
| sutnots |
| webapp150 |
+-----+
5 rows in set (0.05 sec)

mysql> _
    
```

รูปที่ 8.10: หน้าต่างของการแสดงฐานข้อมูลทั้งหมดในระบบ

จากรูปที่ 8.10 นั้นแสดงให้เห็นว่ามีฐานข้อมูลทั้งหมด 5 ฐานข้อมูล ซึ่งตอนนี้ต้องการเรียกใช้ฐานข้อมูล webapp150 เพื่อสร้างตาราง student และตาราง register พร้อมกับป้อนข้อมูลให้ได้ผลดังแสดงในรูปที่ 8.2 จึงต้องใช้คำสั่ง use webapp150; ดังแสดงในรูปที่ 8.11

```

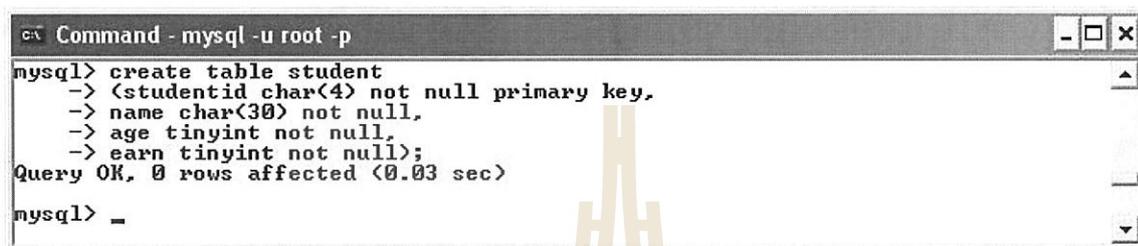
C:\ Command - mysql -u root -p
mysql> use webapp150;
Database changed
mysql> _
    
```

รูปที่ 8.11: การเรียกใช้ฐานข้อมูล webapp150

ตอนนี้กำลังใช้งานฐานข้อมูล webapp150 อยู่ ต่อไปใช้คำสั่งในการสร้างและกำหนดรูปแบบตารางชื่อว่า student ได้ดังนี้

```
create table student
(studentid char(4) not null primary key,
name char(30) not null,
age tinyint not null,
earn tinyint not null);
```

เมื่อบ้อนคำสั่งดังกล่าวแล้ว ถ้าสร้างตารางได้สำเร็จก็จะปรากฏผลดังแสดงในรูปที่ 8.12



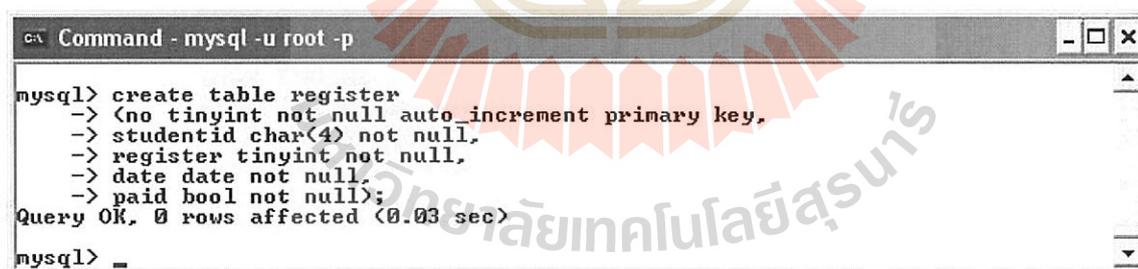
```
c:\ Command - mysql -u root -p
mysql> create table student
-> (studentid char(4) not null primary key,
-> name char(30) not null,
-> age tinyint not null,
-> earn tinyint not null);
Query OK, 0 rows affected (0.03 sec)
mysql> _
```

รูปที่ 8.12: การสร้างและกำหนดรูปแบบตาราง student

ต่อไปใช้คำสั่งในการสร้างตารางชื่อว่า register ได้ดังนี้

```
create table register
(no int not null auto_increment primary key,
studentid char(4) not null,
register tinyint not null,
date date not null,
paid bool not null);
```

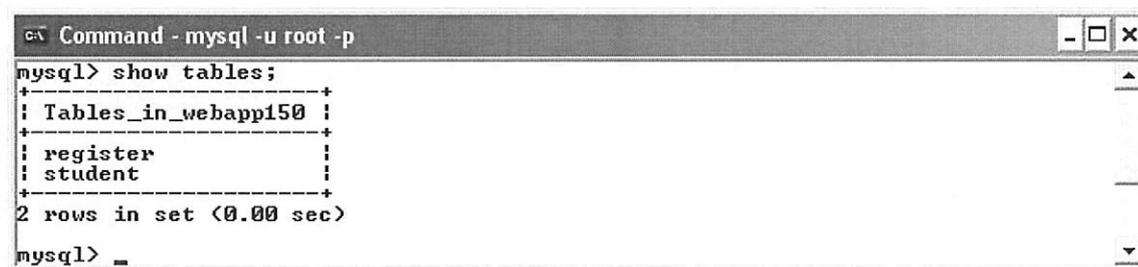
เมื่อบ้อนคำสั่งดังกล่าวแล้ว ถ้าสร้างตารางได้สำเร็จก็จะปรากฏผลดังแสดงในรูปที่ 8.13



```
c:\ Command - mysql -u root -p
mysql> create table register
-> (no tinyint not null auto_increment primary key,
-> studentid char(4) not null,
-> register tinyint not null,
-> date date not null,
-> paid bool not null);
Query OK, 0 rows affected (0.03 sec)
mysql> _
```

รูปที่ 8.13: การสร้างและกำหนดรูปแบบตาราง register

คำสั่งที่ใช้ในการแสดงตารางทั้งหมดคือ show tables; ดังแสดงในรูปที่ 8.14



```
c:\ Command - mysql -u root -p
mysql> show tables;
+-----+
| Tables_in_webapp150 |
+-----+
| register             |
| student              |
+-----+
2 rows in set (0.00 sec)
mysql> _
```

รูปที่ 8.14: การแสดงตารางทั้งหมดในฐานข้อมูล webapp150

จากรูปที่ 8.13 นั้นจะเห็นได้ว่าในฐานข้อมูล webapp150 นั้น มีตารางอยู่ 2 ตารางคือ register และ student ซึ่งตอนนี้ทั้ง 2 ตารางยังไม่มีข้อมูลใด ๆ ทั้งสิ้น นอกจากคำสั่งที่ใช้ในการสร้างตารางแล้ว ยังมีคำสั่ง describe ที่ใช้ดูรายละเอียดของโครงสร้างตาราง ดังแสดงในรูปที่ 8.15 และ 8.16

```

c:\ Command - mysql -u root -p
mysql> describe student;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| studentid  | char(4)       | NO   | PRI  |          |       |
| name       | char(30)      | NO   |      |          |       |
| age        | tinyint(4)    | NO   |      |          |       |
| earn       | tinyint(4)    | NO   |      |          |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
mysql> _
    
```

รูปที่ 8.15: รายละเอียดโครงสร้างตาราง student

```

c:\ Command - mysql -u root -p
mysql> describe register;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no         | tinyint(4)    | NO   | PRI  | NULL    | auto_increment |
| studentid  | char(4)       | NO   |      |          |       |
| register   | tinyint(4)    | NO   |      |          |       |
| date       | date          | NO   |      |          |       |
| paid       | tinyint(1)    | NO   |      |          |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> _
    
```

รูปที่ 8.16: การใช้คำสั่ง describe ดูรายละเอียดโครงสร้างตาราง register

การกำหนดสิทธิ์ ฐานข้อมูลที่ถูกสร้างขึ้นมานั้น ควรมีการกำหนดสิทธิ์ในการใช้งาน เช่น กำหนดสิทธิ์ให้นาย ก. สามารถทำทุกอย่างกับฐานข้อมูลนี้ได้ ส่วนนาย ข. นั้นให้สิทธิ์เฉพาะการเพิ่มและแก้ไขข้อมูลได้เท่านั้น เป็นต้น ซึ่งการกำหนดสิทธิ์ต่าง ๆ นั้นได้แบ่งออกเป็น สิทธิ์สำหรับผู้ใช้งาน (User) สิทธิ์สำหรับผู้ดูแลระบบ (Administrator) และสิทธิ์พิเศษ ดังแสดงในตารางที่ 8.1, 8.2 และ 8.3 ตามลำดับ

ตารางที่ 8.1: สิทธิ์ต่าง ๆ สำหรับผู้ใช้งาน (User)

สิทธิ์	ใช้กับ	ความหมาย
SELECT	ตารางและแถว	ผู้ใช้งานสามารถเลือกแถวหรือเรคคอร์ดจากตารางได้
INSERT	ตารางและแถว	ผู้ใช้งานสามารถเพิ่มแถวลงในตารางได้
UPDATE	ตารางและแถว	ผู้ใช้งานสามารถแก้ไขค่าที่อยู่ในคอลัมน์ของแถวในตารางได้
DELETE	ตาราง	ผู้ใช้งานสามารถลบแถวข้อมูลในตารางได้
INDEX	ตาราง	ผู้ใช้งานสามารถสร้างหรือลบตัวชี้ (index) บนตารางได้ โดยใช้ร่วมกับ CREATE หรือ DROP
ALTER	ตาราง	ผู้ใช้งานสามารถดัดแปลงโครงสร้างของตารางได้ เช่น เพิ่มคอลัมน์ เปลี่ยนชนิดข้อมูลของคอลัมน์ หรือเปลี่ยนชื่อคอลัมน์หรือตารางได้
CREATE	ฐานข้อมูลและตาราง	ผู้ใช้งานสามารถสร้างฐานข้อมูลหรือตารางใหม่ได้
TRUNCATE	ตาราง	ผู้ใช้งานสามารถลบข้อมูลทั้งหมดในตาราง โดยตารางยังคงอยู่
DROP	ฐานข้อมูลและตาราง	ผู้ใช้งานสามารถลบตารางหรือฐานข้อมูลได้

ตารางที่ 8.2: สิทธิ์ต่าง ๆ สำหรับผู้ดูแลระบบ (Administrator)

สิทธิ์	ความหมาย
RELOAD	ผู้ดูแลระบบสามารถโหลดสิทธิ์ให้ใหม่
SHUTDOWN	ผู้ดูแลระบบสามารถปิดระบบฐานข้อมูล MySQL ได้
PROCESS	ผู้ดูแลระบบสามารถดูกระบวนการทำงานของเครื่องแม่ข่ายและสามารถหยุดการทำงานของกระบวนการทำงานนั้น ๆ ได้
FILE	สามารถอ่านข้อมูลจากแฟ้มข้อมูลเข้าสู่ตาราง หรืออ่านจากตารางแล้วบันทึกลงแฟ้มข้อมูล

ตารางที่ 8.3: สิทธิ์พิเศษ

สิทธิ์	ความหมาย
ALL	ให้สิทธิ์ทั้งหมดในตารางที่ 8.1 และ 8.2
USAGE	กำหนดผู้ใช้งาน (User) แต่ไม่ให้สิทธิ์อะไรเลย

- คำสั่ง GRANT ใช้ในการให้สิทธิ์ต่าง ๆ แก่ผู้ใช้ ซึ่งมีรูปแบบของคำสั่งดังนี้

```
GRANT priv_type[(column_list)][,priv_type
[(column_list)]].
ON [object_type] {tbl_name | * | *.* | db_name.*}
TO user [IDENTIFIED BY [PASSWORD] 'password'
[, user [IDENTIFIED BY [PASSWORD] 'password']] ...
[REQUIRE
NONE |
[SSL | X509]]
[CIPHER 'cipher' [AND]]
[ISSUER 'issuer' [AND]]
[SUBJECT 'subject']]
[WITH with_option [with_option] ...]

object_type =
TABLE
| FUNCTION
| PROCEDURE

with_option =
GRANT OPTION
| MAX_QUERIES_PER_HOUR count
| MAX_UPDATES_PER_HOUR count
| MAX_CONNECTIONS_PER_HOUR count
| MAX_USER_CONNECTIONS count
```

ตัวอย่าง การใช้คำสั่ง GRANT ในการกำหนดสิทธิ์ทุกอย่าง (all) ทุกระบบข้อมูล (on *) ให้กับ somchai ที่ใช้รหัสผ่านเป็น sc1234 พร้อมทั้งอนุญาตให้ somchai กำหนดสิทธิ์ให้กับคนอื่น ๆ ได้

```
mysql> grant all
-> on *
-> to somchai identified by 'sc1234'
-> with grant option;
```

ตัวอย่าง การใช้คำสั่ง GRANT ในการกำหนดสิทธิ์ทุกอย่าง (all) ในการจัดการตารางทุกตารางของฐานข้อมูลชื่อว่า webapp150 ให้กับ anucha ที่ใช้รหัสผ่านเป็น ac1999

```
mysql> grant all
-> on webapp150.*
-> to anucha identified by 'ac1999';
```

- คำสั่ง **REVOKE** ใช้ในการยกเลิกสิทธิ์ต่าง ๆ ที่ได้ให้แก่ผู้ใช้ ซึ่งมีรูปแบบของคำสั่งดังนี้

```
REVOKE priv_type [(column_list)] [, priv_type
  [(column_list)]] ...
ON [object_type] {tbl_name | * | *.* | db_name.*}
FROM user [, user] ...
```

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM user [, user] ...
```

ตัวอย่าง การใช้คำสั่ง **REVOKE** ในการยกเลิกสิทธิ์ต่าง ๆ ที่ให้กับ somchai

```
mysql> revoke grant option
-> on *
-> from somchai;
```

ตัวอย่าง การใช้คำสั่ง **REVOKE** ในการยกเลิกสิทธิ์ต่าง ๆ ที่ให้กับ anucha

```
mysql> revoke all
-> on webapp150.*
-> from anucha;
```

8.3.2 ภาษาปรับแต่งข้อมูล (Data Manipulation Languages: DML)

ภาษาปรับแต่งข้อมูล (Data Manipulation Languages: DML) ใช้ในการเรียกดูข้อมูล แก้ไขเพิ่มเติม และลบข้อมูลในฐานข้อมูลได้ โดยสามารถเรียกดูข้อมูลจากตารางเดียว หรือจากหลาย ๆ ตาราง โดยกำหนดเงื่อนไขตามต้องการได้ จากหัวข้อที่ผ่านมาได้ ได้สร้างตารางขึ้นมา 2 ตารางเรียบร้อยแล้ว ตอนนี้จะใช้คำสั่ง **INSERT** ในการเพิ่มข้อมูลเข้าไปในตาราง student และตาราง register หลังจากนั้นจะใช้คำสั่ง **SELECT** เรียกดูข้อมูลทั้งหมดในตาราง student และตาราง register ตามลำดับ

- คำสั่ง **INSERT** เป็นคำสั่งที่ใช้ในการเพิ่มข้อมูลเข้าไปในตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name [(col_name,...)]
VALUES ({expr | DEFAULT},...), (...),...
[ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

หรือ

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name
SET col_name={expr | DEFAULT}, ...
[ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

หรือ

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name [(col_name,...)]
SELECT ...
[ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

```

c:\ Command - mysql -u root -p
mysql> insert into student values
-> ('0001',"Somchai Jaidee",21,85),
-> ('0002',"Jaidee Yoosamor",20,78),
-> ('0003',"Sombat Saitalord",20,88),
-> ('0004',"Onyon Tamvela",22,81),
-> ('0005',"Somporn Tampark",21,80);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> _
    
```

รูปที่ 8.17: การเพิ่มข้อมูลเข้าไปในตาราง student

```

c:\ Command - mysql -u root -p
mysql> insert into register (studentid,register,date,paid) values
-> ('0002",18,"20070615",0),
-> ('0005",18,"20070616",0),
-> ('0001",21,"20070616",1),
-> ('0003",15,"20070618",1),
-> ('0004",16,"20070620",0);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> _
    
```

รูปที่ 8.18: การเพิ่มข้อมูลเข้าไปในตาราง register

- คำสั่ง **SELECT** เป็นคำสั่งที่ใช้ในการดูข้อมูลที่อยู่ในคอลัมน์จากตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

```

SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  [HIGH PRIORITY]
  [STRAIGHT JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT]
  [SQL_BUFFER_RESULT]
  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr, ...
  [FROM table_references
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | position}
  [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
  [PROCEDURE procedure_name(argument_list)]
  [INTO OUTFILE 'file_name' export_options
  | INTO DUMPFILE 'file_name']
  [FOR UPDATE | LOCK IN SHARE MODE]]
    
```

หมายเหตุ ใช้เครื่องหมาย * ตรงตำแหน่ง column_name ในการเลือกตารางทุกตาราง

```

c:\ Command - mysql -u root -p
mysql> select * from student;
+----+-----+-----+-----+
| studentid | name           | age | earn |
+----+-----+-----+-----+
| 0001      | Somchai Jaidee | 21  | 85   |
| 0002      | Jaidee Yoosamor | 20  | 78   |
| 0003      | Sombat Saitalord | 20  | 88   |
| 0004      | Onyon Tamvela  | 22  | 81   |
| 0005      | Somporn Tampark | 21  | 80   |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.19: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student

```

c:\ Command - mysql -u root -p
mysql> select * from register;
+----+-----+-----+-----+-----+
| no | studentid | register | date       | paid |
+----+-----+-----+-----+-----+
| 1  | 0002      | 18       | 2007-06-15 | 0    |
| 2  | 0005      | 18       | 2007-06-16 | 0    |
| 3  | 0001      | 21       | 2007-06-16 | 1    |
| 4  | 0003      | 15       | 2007-06-18 | 1    |
| 5  | 0004      | 16       | 2007-06-20 | 0    |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.20: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง register

ตัวอย่าง การใช้คำสั่ง SELECT เพื่อเรียกดูข้อมูลทั้งหมดในตาราง student ใช้คอลัมน์ earn เป็นหลักโดยเรียงลำดับจากน้อยไปมาก (Ascending Order)

```

c:\ Command - mysql -u root -p
mysql> select * from student order by earn;
+----+-----+-----+-----+
| studentid | name           | age | earn |
+----+-----+-----+-----+
| 0002      | Jaidee Yoosamor | 20  | 78   |
| 0005      | Somporn Tampark | 21  | 80   |
| 0004      | Onyon Tamvela  | 22  | 81   |
| 0001      | Somchai Jaidee | 21  | 85   |
| 0003      | Sombat Saitalord | 20  | 88   |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.21: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student ใช้คอลัมน์ earn เป็นหลักเรียงลำดับจากน้อยไปมาก (Ascending Order)

ตัวอย่าง การใช้คำสั่ง SELECT เพื่อเรียกดูข้อมูลทั้งหมดในตาราง student ใช้คอลัมน์ earn เป็นหลักโดยเรียงลำดับจากมากไปน้อย (Descending Order)

```

Command - mysql -u root -p
mysql> select * from student order by earn desc;
+-----+-----+-----+-----+
| studentid | name           | age | earn |
+-----+-----+-----+-----+
| 0003      | Sombat Saitalord | 20  | 88   |
| 0001      | Sonchai Jaidee   | 21  | 85   |
| 0004      | Onyon Tamvela    | 22  | 81   |
| 0005      | Somporn Tampark  | 21  | 80   |
| 0002      | Jaidee Yoosamor  | 20  | 78   |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.22: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student ใช้คอลัมน์ earn เป็นหลักเรียงลำดับจากมากไปน้อย (Descending Order)

- คำสั่ง UPDATE เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงข้อมูล ซึ่งมีรูปแบบของคำสั่งดังนี้

```

UPDATE [LOW_PRIORITY] [IGNORE] tbl_name
      SET col_name1=expr1 [, col_name2=expr2 ...]
      [WHERE where_condition]
      [ORDER BY ...]
      [LIMIT row_count]
    
```

จากรูปที่ 8.20 นั้น ถ้าต้องการเปลี่ยนวันที่ที่ลงทะเบียนของนักเรียนที่มี studentid เท่ากับ 0004 จากเดิมวันที่ 20 ให้เป็นวันที่ 19 สามารถทำได้โดยใช้คำสั่ง UPDATE ดังแสดงในรูปที่ 8.23 หลังจากนั้นใช้คำสั่ง SELECT เพื่อดูข้อมูลที่ถูกละเปลี่ยนไป ดังแสดงในรูปที่ 8.24

```

Command - mysql -u root -p
mysql> update register
-> set date="20070619" where studentid="0004";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> _
    
```

รูปที่ 8.23: การใช้คำสั่ง update เพื่อเปลี่ยนแปลงข้อมูลในตาราง register

```

Command - mysql -u root -p
mysql> select * from register;
+----+-----+-----+-----+-----+
| no | studentid | register | date       | paid |
+----+-----+-----+-----+-----+
| 1  | 0002      | 18      | 2007-06-15 | 0    |
| 2  | 0005      | 18      | 2007-06-16 | 0    |
| 3  | 0001      | 21      | 2007-06-16 | 1    |
| 4  | 0003      | 15      | 2007-06-18 | 1    |
| 5  | 0004      | 16      | 2007-06-19 | 0    |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.24: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง register หลังมีการเปลี่ยนแปลง

- คำสั่ง DELETE ใช้ในการลบแถวข้อมูลในตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
FROM tbl_name[.*] [, tbl_name[.*]] ...
USING table_references
[WHERE where_condition]
```

ตัวอย่าง การใช้คำสั่ง DELETE ลบแถวข้อมูลแถวสุดท้าย (no = 5) ในตาราง register

```
Command - mysql -u root -p
mysql> delete from register
-> where no=5;
Query OK, 1 row affected (0.02 sec)
mysql> _
```

รูปที่ 8.25: การใช้คำสั่ง delete เพื่อลบแถวข้อมูล

```
Command - mysql -u root -p
mysql> select * from register;
+----+-----+-----+-----+-----+
| no | studentid | register | date       | paid |
+----+-----+-----+-----+-----+
| 1  | 0002      | 18      | 2007-06-15 | 0    |
| 2  | 0005      | 18      | 2007-06-16 | 0    |
| 3  | 0001      | 21      | 2007-06-16 | 1    |
| 4  | 0003      | 15      | 2007-06-18 | 1    |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
mysql> _
```

รูปที่ 8.26: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง register หลังมีการลบแถวข้อมูล

ตอนนี้ทดลองใส่ข้อมูลที่เพิ่งลบทิ้งกลับคืนแล้วดูข้อมูลที่อยู่ในตาราง register โดยใช้คำสั่งดังแสดงในรูปที่ 8.27

```
Command - mysql -u root -p
mysql> insert into register (studentid,register,date,paid) values
-> (<"0004",16,"20070619",0);
Query OK, 1 row affected (0.00 sec)
mysql> select * from register;
+----+-----+-----+-----+-----+
| no | studentid | register | date       | paid |
+----+-----+-----+-----+-----+
| 1  | 0002      | 18      | 2007-06-15 | 0    |
| 2  | 0005      | 18      | 2007-06-16 | 0    |
| 3  | 0001      | 21      | 2007-06-16 | 1    |
| 4  | 0003      | 15      | 2007-06-18 | 1    |
| 6  | 0004      | 16      | 2007-06-19 | 0    |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> _
```

รูปที่ 8.27: การใช้คำสั่ง insert ในการเพิ่มข้อมูลลงในตาราง register และใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง

จะสังเกตเห็นข้อมูลที่ใส่เพิ่มเข้าไปใหม่ ตรงคอลัมน์ no (ซึ่งมีชนิดข้อมูลเป็นแบบ Tinyint และ auto_increment) นั้นแสดงค่าเป็น 6 เนื่องจากเดิมมีข้อมูลอยู่ 5 แถว ถึงแม้ว่าจะมีการลบแถวข้อมูลทิ้งไป 1 แถวกก็ตาม ตัว auto_increment ก็ยังจำค่าเดิมคือจำนวนครั้งที่มีการเพิ่มข้อมูลเข้ามา แล้วทำการเพิ่มค่าจากเดิมซึ่งเคยมีอยู่ถึง 5 แถวข้อมูล ไปเป็น 6 โดยอัตโนมัติ ตัวเลขในคอลัมน์ no นี้ไม่สามารถนำไปเป็น

ตัวนับจำนวนแถวหรือจำนวนข้อมูลที่มีอยู่ในตารางได้ เพียงแต่เป็นตัวบ่งบอกว่ามีการเพิ่มข้อมูลเข้าไปทั้งหมดกี่ครั้งเท่านั้นเอง

- คำสั่ง INDEX ใช้เพื่อเป็นตัวชี้ไปยังคอลัมน์ ทำให้การเข้าถึงข้อมูลทำได้เร็วขึ้น มักใช้กับคอลัมน์ที่เป็นคีย์หลักหรือคีย์รอง ซึ่งมีรูปแบบของคำสั่งดังนี้

```
CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name
      [USING type_name]
      ON tbl_name (index_col_name,...)
```

ชื่อตัวชี้คอลัมน์

```
index_col_name:
      col_name [(length)] [ASC | DESC]
```

การสร้างตัวชี้

```
CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name
      ON table_name (index_col_name,...);
```

การลบตัวชี้

```
DROP INDEX index_name ON table_name;
```

ตัวอย่าง การใช้คำสั่ง CREATE UNIQUE INDEX สร้างตัวชี้ชื่อว่า xid ที่ไม่ซ้ำกัน (Unique) ให้กับคอลัมน์ studentid ในตาราง register

```
mysql> create unique index xid
-> on register(studentid);
Query OK, 5 rows affected (0.03 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> _
```

รูปที่ 8.28: การสร้างตัวชี้ชื่อว่า xid ที่ไม่ซ้ำกัน (Unique) ให้กับคอลัมน์ studentid ในตาราง register

```
mysql> describe register;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| no | tinyint(4) | NO | PRI | NULL | auto_increment |
| studentid | char(4) | NO | UNI | | |
| register | tinyint(4) | NO | | | |
| date | date | NO | | | |
| paid | tinyint(1) | NO | | | |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> _
```

รูปที่ 8.29: การใช้คำสั่ง describe ดูรายละเอียดโครงสร้างตาราง register หลังจากการสร้างตัวชี้ให้กับคอลัมน์ studentid

การกำหนดตัวชี้ให้กับคอลัมน์จะช่วยให้การค้นหาข้อมูลในคอลัมน์นั้น ๆ มีความรวดเร็วยิ่งขึ้น จะเห็นผลได้ชัดเจนยิ่งขึ้นเมื่อมีจำนวนข้อมูลมาก ๆ

ตัวอย่าง การใช้คำสั่ง DROP INDEX ลบตัวชี้ชื่อว่า xid ออกจากตาราง register

```

c:\ Command - mysql -u root -p
mysql> drop index xid on register;
Query OK, 5 rows affected (0.03 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> _
    
```

รูปที่ 8.30: การลบตัวชี้ xid ออกจากตาราง register

```

c:\ Command - mysql -u root -p
mysql> describe register;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| no         | tinyint(4)    | NO   | PRI | NULL     | auto_increment |
| studentid | char(4)       | NO   |     |          |                |
| register   | tinyint(4)    | NO   |     |          |                |
| date       | date          | NO   |     |          |                |
| paid       | tinyint(1)    | NO   |     |          |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> _
    
```

รูปที่ 8.31: การใช้คำสั่ง describe ดูรายละเอียดโครงสร้างตาราง register หลังจากการลบตัวชี้ xid ออกจากตาราง register

- คำสั่ง ALTER ใช้ในการดัดแปลงโครงสร้างของตาราง เช่น เพิ่มคอลัมน์ เปลี่ยนชนิดข้อมูลของคอลัมน์ เปลี่ยนชื่อคอลัมน์หรือตาราง ซึ่งมีรูปแบบของคำสั่งดังนี้

```
ALTER [IGNORE] TABLE tbl_name
    alter_specification [, alter_specification] ...
```

ข้อกำหนดในคำสั่ง ALTER

alter_specification:

```

ADD [COLUMN] column_definition [FIRST|AFTER col_name]
| ADD [COLUMN] (column_definition,...)
| ADD {INDEX|KEY} [index_name] [index_type]
    (index_col_name,...)
| ADD [CONSTRAINT [symbol]]
    PRIMARY KEY [index_type] (index_col_name,...)
| ADD [CONSTRAINT [symbol]]
    UNIQUE [INDEX|KEY] [index_name] [index_type]
    (index_col_name,...)
| ADD [FULLTEXT|SPATIAL] [INDEX|KEY] [index_name]
    (index_col_name,...)
| ADD [CONSTRAINT [symbol]]
    FOREIGN KEY [index_name] (index_col_name,...)
    [reference_definition]
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP
    DEFAULT}
| CHANGE [COLUMN] old_col_name column_definition
    [FIRST|AFTER col_name]
| MODIFY [COLUMN] column_definition [FIRST | AFTER
    col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO] new_tbl_name
| ORDER BY col_name
    
```

```
| CONVERT TO CHARACTER SET charset_name [COLLATE
| collation_name]
| [DEFAULT] CHARACTER SET charset_name [COLLATE
| collation_name]
| DISCARD TABLESPACE
| IMPORT TABLESPACE
| table_option ...
```

ตัวอย่าง การใช้คำสั่ง ALTER เพื่อเปลี่ยนชื่อตารางจาก student เป็น student_list

```
Command - mysql -u root -p
mysql> alter table student
-> rename student_list;
Query OK, 0 rows affected (0.01 sec)
mysql> _
```

รูปที่ 8.32: การใช้คำสั่ง alter เปลี่ยนชื่อตาราง

ตัวอย่าง การใช้คำสั่ง ALTER เพื่อเพิ่มคอลัมน์ชื่อ telephone ในตาราง student_list ดังแสดงในรูปที่ 8.33 แล้วใช้คำสั่ง desc หรือ describe เพื่อดูโครงสร้างของตาราง student_list ดังแสดงในรูปที่ 8.34

```
Command - mysql -u root -p
mysql> alter table student_list
-> add (telephone char(10));
Query OK, 5 rows affected (0.03 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> _
```

รูปที่ 8.33: การใช้คำสั่ง alter เพิ่มคอลัมน์ในตาราง

```
Command - mysql -u root -p
mysql> desc student_list;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| studentid  | char(4)   | NO   | PRI |          |       |
| name       | char(30)  | NO   |     |          |       |
| age        | tinyint(4)| NO   |     |          |       |
| earn       | tinyint(4)| NO   |     |          |       |
| telephone  | char(10)  | YES  |     |          |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> _
```

รูปที่ 8.34: การใช้คำสั่ง desc ดูรายละเอียดโครงสร้างตาราง student_list หลังการเพิ่มคอลัมน์ telephone

ตัวอย่าง การใช้คำสั่ง ALTER เพื่อเปลี่ยนชื่อคอลัมน์ในตาราง student_list จาก earn เป็น earned ดังแสดงในรูปที่ 8.35 แล้วใช้คำสั่ง desc หรือ describe เพื่อดูโครงสร้างของตาราง student_list ดังแสดงในรูปที่ 8.36

```

Command - mysql -u root -p
mysql> alter table student_list
-> change earn earned tinyint not null;
Query OK, 5 rows affected (0.05 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> _
    
```

รูปที่ 8.35: การใช้คำสั่ง alter เพิ่มเปลี่ยนชื่อคอลัมน์จาก earn เป็น earned ในตาราง student_list

```

Command - mysql -u root -p
mysql> desc student_list;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| studentid | char(4)       | NO   | PRI |          |       |
| name       | char(30)     | NO   |     |          |       |
| age        | tinyint(4)   | NO   |     |          |       |
| earned     | tinyint(4)   | NO   |     |          |       |
| telephone | char(10)     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql> _
    
```

รูปที่ 8.36: การใช้คำสั่ง desc ดูรายละเอียดโครงสร้างตาราง student_list หลังจากเปลี่ยนชื่อคอลัมน์

เราได้เพิ่มตาราง telephone ในตาราง student_list เรียบร้อยแล้ว (ดังแสดงในรูปที่ 8.33) แต่ยังไม่ได้ใส่ข้อมูลเบอร์โทรศัพท์ ดังนั้นในตอนนี้จะใช้คำสั่ง UPDATE เพื่อทำการปรับปรุงข้อมูลใหม่โดยการใส่เบอร์โทรศัพท์

```

Command Prompt - mysql -u root -p
mysql> update student_list
-> set telephone="0894251199" where studentid="0001";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> update student_list
-> set telephone="0894251155" where studentid="0002";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> update student_list
-> set telephone="0894251111" where studentid="0003";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> update student_list
-> set telephone="0894251122" where studentid="0004";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> update student_list
-> set telephone="0894251177" where studentid="0005";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> _
    
```

รูปที่ 8.37: การใช้คำสั่ง update เพื่อแก้ไขข้อมูลเบอร์โทรศัพท์ในคอลัมน์ telephone ของตาราง student_list

ทดลองใช้คำสั่ง SELECT เพื่อดูข้อมูลทั้งหมดในตาราง student_list

```

C:\ Command Prompt - mysql -u root -p
mysql> select * from student_list;
+-----+-----+-----+-----+-----+
| studentid | name           | age | earned | telephone |
+-----+-----+-----+-----+-----+
| 0001      | Somchai Jaidee | 21  | 85     | 0894251199 |
| 0002      | Jaidee Yoosamor | 20  | 78     | 0894251155 |
| 0003      | Sombat Saitalord | 20  | 88     | 0894251111 |
| 0004      | Onyon Tamvela  | 22  | 81     | 0894251122 |
| 0005      | Somporn Tampark | 21  | 80     | 0894251177 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.38: การใช้คำสั่ง select เพื่อดูข้อมูลทั้งหมดในตาราง student_list หลังจากมีการปรับปรุงข้อมูลเบอร์โทรศัพท์

- คำสั่ง TRUNCATE ใช้ในการลบข้อมูลทั้งหมดในตาราง แต่ชื่อตารางยังคงอยู่ ซึ่งมีรูปแบบของคำสั่งดังนี้

```
TRUNCATE TABLE table_name;
```

- คำสั่ง DROP ใช้ในการลบตารางหรือฐานข้อมูล ซึ่งมีรูปแบบของคำสั่งดังนี้

การลบฐานข้อมูล

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

การลบตาราง

```
DROP [TEMPORARY] TABLE [IF EXISTS]
tbl_name [, tbl_name] ...
[RESTRICT | CASCADE]
```

ตัวอย่าง การใช้คำสั่ง CREATE TABLE ชื่อว่า tb_test หลังจากนั้นจะใช้คำสั่ง show tables เพื่อเช็คดูตารางทั้งหมด แล้วจะใช้คำสั่ง DROP TABLE เพื่อทำการลบตารางที่ชื่อว่า tb_test

```

C:\ Command - mysql -u root -p
mysql> create table tb_test
-> (col char(10));
Query OK, 0 rows affected (0.03 sec)

mysql> _
    
```

รูปที่ 8.39: การใช้คำสั่ง create table สร้างตารางชื่อว่า tb_test

```

C:\ Command - mysql -u root -p
mysql> show tables;
+-----+
| Tables_in_webapp150 |
+-----+
| register            |
| student_list        |
| tb_test              |
+-----+
3 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.40: การใช้คำสั่ง show tables เพื่อแสดงชื่อตารางทั้งหมด

```

c:\ Command - mysql -u root -p
mysql> drop table tb_test;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_webapp150 |
+-----+
| register             |
| student_list        |
+-----+
2 rows in set (0.00 sec)

mysql> _
    
```

รูปที่ 8.41: การใช้คำสั่ง drop table เพื่อลบตาราง tb_test และคำสั่ง show tables เพื่อแสดงชื่อตารางทั้งหมด

- คำสั่ง RELOAD เป็นคำสั่งที่ยอมให้ผู้ดูแลระบบ โหลดสถิติให้ใหม่
- คำสั่ง SHUTDOWN เป็นคำสั่งที่ยอมให้ผู้ดูแลระบบปิด ระบบฐานข้อมูล MySQL
- คำสั่ง PROCESS เป็นคำสั่งที่ยอมให้ผู้ดูแลระบบดูกระบวนการของการทำงาน
- คำสั่ง FILE เป็นคำสั่งที่ยอมให้อ่านข้อมูลเข้าจากแฟ้มข้อมูล หรือส่งข้อมูลออกไปที่แฟ้มข้อมูลได้

ตัวอย่าง การดึงข้อมูลจากตารางแล้วบันทึกลงแฟ้มข้อมูล

```
mysql> SELECT * FROM register INTO OUTFILE 'c:/register.txt';
```

File	Edit	Format	View	Help
1	0002	18	2007-06-15	02 0005 18 2007-06-16 0
3	0001	21	2007-06-16	14 0003 15 2007-06-18 1
6	0004	16	2007-06-19	0

รูปที่ 8.42: การเปิดแฟ้มข้อมูลที่ถูกเขียนลงโดยคำสั่ง OUTFILE

ตัวอย่าง การดึงข้อมูลจากแฟ้มข้อมูลแล้วบันทึกลงตาราง

```
mysql> LOAD DATA INFILE 'c:/register.txt' INTO TABLE register;
```

8.4 ชนิดข้อมูลของคอลัมน์

ชนิดข้อมูลที่จะกำหนดให้กับคอลัมน์หนึ่ง ๆ ในระบบฐานข้อมูล MySQL นั้นถูกแบ่งออกเป็น 3 ชนิด ได้แก่ ชนิดตัวเลข ชนิดวันเวลา และชนิดข้อความ โดยแต่ละชนิดต่างก็มีขอบเขตในการเก็บข้อมูลและขนาดในการใช้พื้นที่ในหน่วยความจำต่างกัน ดังนั้นจึงมีความจำเป็นที่จะต้องทำความเข้าใจชนิดข้อมูลต่าง ๆ เพื่อให้สามารถกำหนดชนิดข้อมูลได้ถูกต้องและใช้เนื้อที่น้อยที่สุด ทั้งนี้ เพื่อให้การทำงานของระบบเร็วและมีประสิทธิภาพมากยิ่งขึ้น จึงขอกล่าวถึงรายละเอียดของชนิดต่าง ๆ ดังต่อไปนี้

8.4.1 ชนิดตัวเลข (Numeric Type)

ชนิดตัวเลข ประกอบไปด้วยเลขจำนวนเต็มและเลขจำนวนจริง ดังแสดงในตารางที่ 8.4 และ 8.5 ตามลำดับ

ตารางที่ 8.4: ชนิดข้อมูลแบบเลขจำนวนเต็ม

ชนิด	ช่วงข้อมูล	ใช้เนื้อที่	ความหมาย
TINYINT	0...255 หรือ -128...127	1	เก็บเลขจำนวนเต็มขนาดเล็กมาก
SMALLINT	0...65535 หรือ -32768...32767	2	เก็บเลขจำนวนเต็มขนาดเล็ก
MEDIUMINT	0...16777215 หรือ -8388608...8388607	3	เก็บเลขจำนวนเต็มขนาดกลาง
INT	0...4294967295 หรือ -2147483648...2147483647	4	เก็บเลขจำนวนเต็มขนาดปกติ
BIGINT	0...18446744073709551615 หรือ -9223372036854775808...9223372036854775807	8	เก็บเลขจำนวนเต็มขนาดใหญ่

ตารางที่ 8.5: ชนิดข้อมูลแบบเลขจำนวนจริง

ชนิด	ช่วงข้อมูล	ใช้เนื้อที่	ความหมาย
FLOAT	±1.175494351E-38...±3.402823466E+38	4	เก็บเลขจำนวนจริงขนาดปกติ
DOUBLE	±2.2250738585072014E-308... ±1.7976931348623157E+308	8	เก็บเลขจำนวนจริงขนาดใหญ่

8.4.2 ชนิดวันและเวลา (Date and Time Type)

ชนิดข้อมูลสามารถกำหนดให้เป็น วันหรือ/และเวลาได้ ดังแสดงในตารางที่ 8.6

ตารางที่ 8.6: ชนิดข้อมูลแบบวันเวลา

ชนิด	ช่วงข้อมูล	ใช้เนื้อที่	ความหมาย
DATE	1000-01-01...9999-12-31	3	เก็บข้อมูลเป็น ค.ศ-เดือน-วัน (YYYY-MM-DD)
TIME	-838:59:59...838:59:59	3	เก็บข้อมูลเป็น ชั่วโมง:นาที:วินาที (HH:MM:SS)
DATETIME	1000-01-01 00:00:00 ... 9999-12-31 23:59:59	8	เก็บข้อมูลเป็น ค.ศ-เดือน-วัน ชั่วโมง:นาที:วินาที (YYYY-MM-DD HH:MM:SS)
TIMESTAMP [M]	19700101000000...20371231235959	4	เก็บข้อมูลเป็น ค.ศ-เดือน-วัน ชั่วโมง:นาที:วินาที ส่วนรูปแบบนั้นขึ้นอยู่กับค่า M (ดูรายละเอียดค่า M ในตารางที่ 8.7)
YEAR [(2 4)]	70-69 หรือ 1970-2069	1	เก็บข้อมูลเป็นปี (ระบุเป็น 2 จะได้ 70-69 ระบุ 4 จะได้ 1970-2069)

การแสดงผลของ TIMESTAMP แสดงในตารางที่ 8.7

ตารางที่ 8.7: การแสดงผลของ TIMESTAMP

ชนิดที่กำหนด	การแสดงผล
TIMESTAMP	YYYYMMDDHHMMSS
TIMESTAMP (14)	YYYYMMDDHHMMSS
TIMESTAMP (12)	YYMMDDHHMMSS
TIMESTAMP (10)	YYMMDDHHMM
TIMESTAMP (8)	YYYYMMDD
TIMESTAMP (6)	YYMMDD
TIMESTAMP (4)	YYMM
TIMESTAMP (2)	YY

8.4.3 ชนิดข้อความ (String Type)

ชนิดข้อความถูกแบ่งออกเป็น 3 กลุ่มด้วยกันคือ

1. ชนิดข้อความที่สั้น มี 2 แบบคือ CHAR ที่ระบุจำนวนตัวอักษรที่สามารถเก็บได้ (Fixed length character) ถ้าข้อมูลไม่ถึงจำนวนที่ระบุ จะทำการใส่ช่องว่างในส่วนที่เหลือ เมื่อมีการดึงข้อมูล MySQL ไม่สนใจช่องว่างเหล่านั้น ซึ่งการกำหนดแบบนี้จะทำให้เปลืองเนื้อที่แต่การประมวลผลเร็วกว่าแบบที่ 2 คือ VARCHAR ที่ไม่ระบุจำนวนตัวอักษร (Variable length character) ใช้เนื้อที่เท่ากับจำนวนตัวอักษร นั่นคือเมื่อข้อความมีจำนวนตัวอักษรมากก็ใช้เนื้อที่มาก แต่เมื่อมีจำนวนตัวอักษรน้อยก็ใช้เนื้อที่น้อยตามไปด้วย แต่การประมวลผลจะช้ากว่าแบบ CHAR

ตารางที่ 8.8: ชนิดข้อความแบบ CHAR และ VARCHAR

ชนิด	ความยาวสูงสุด	ความหมาย
CHAR	1	สามารถเก็บจำนวนตัวอักษรได้จำนวน 1 ตัวอักษร
CHAR (n)	255	สามารถเก็บจำนวนตัวอักษรได้จำนวน n ตัวตัวอักษร โดยมีความยาวคงที่ (n)
VARCHAR	65535	สามารถเก็บจำนวนตัวอักษร โดยมีความยาวไม่คงที่

2. ชนิดข้อความที่มีจำนวนตัวอักษรมาก ๆ มี 2 แบบคือ TEXT ที่เก็บข้อความยาว ๆ และ BLOB ย่อมาจาก Binary Large Objects ชนิดข้อความทั้ง 2 แบบนั้นมีลักษณะการทำงานเหมือนกัน เพียงแต่แบบ TEXT นั้นเป็นแบบ Case Sensitive

ตารางที่ 8.9: ชนิดข้อความแบบ TEXT และ BLOB

ชนิด	ความยาวสูงสุด	ความหมาย
TINYBLOB	$2^8 - 1$ หรือ 255	แบบ BLOB ขนาดเล็ก
TINYTEXT	$2^8 - 1$ หรือ 255	แบบ TEXT ขนาดเล็ก
BLOB	$2^{16} - 1$ หรือ 65,535	แบบ BLOB ขนาดปกติ
TEXT	$2^{16} - 1$ หรือ 65,535	แบบ TEXT ขนาดปกติ
MEDIUMBLOB	$2^{24} - 1$ หรือ 16,777,215	แบบ BLOB ขนาดกลาง
MEDIUMTEXT	$2^{24} - 1$ หรือ 16,777,215	แบบ TEXT ขนาดกลาง
LOB	$2^{32} - 1$ หรือ 4,294,967,295	แบบ BLOB ขนาดยาว
LONGTEXT	$2^{32} - 1$ หรือ 4,294,967,295	แบบ TEXT ขนาดยาว

3. ชนิดข้อความแบบพิเศษ มี 2 แบบคือ SET คือการกำหนดให้ค่าในคอลัมน์นั้นต้องมาจากกลุ่มของค่าที่ระบุเท่านั้น ค่าที่อยู่ในคอลัมน์นั้นอาจมีมากกว่า 1 ค่าได้ ส่วนกลุ่มของค่าที่จะระบุนั้นสามารถมีได้ถึง 64 ค่า และ ENUM มีลักษณะการทำงานคล้ายกับ SET เพียงแต่ค่าที่อยู่ในคอลัมน์นั้นจะมีได้เพียงค่าเดียวเท่านั้น ส่วนกลุ่มของค่าที่จะระบุนั้นสามารถมีได้ถึง 65,535 ค่า

ตารางที่ 8.10: ชนิดข้อความแบบ SET และ ENUM

ชนิด	จำนวนค่าสูงสุด	ความหมาย
SET('value1', value2, ...)	64	คอลัมน์ชนิดนี้สามารถเก็บค่าได้หลายค่าจากกลุ่มข้อความที่ระบุ
ENUM('value1', value2, ...)	65,535	คอลัมน์ชนิดนี้สามารถเก็บค่าได้ค่าเดียวจากกลุ่มข้อความที่ระบุ

8.4.4 ตัวดำเนินการในการเปรียบเทียบสำหรับอนุประโยค WHERE

การแสดงผลข้อมูลจากตารางต่าง ๆ นั้น ส่วนใหญ่แล้วจะต้องใช้อนุประโยค WHERE ในการเปรียบเทียบเพื่อให้ได้ผลลัพธ์ตามเงื่อนไขที่ต้องการ ซึ่งตัวเปรียบเทียบต่าง ๆ เหล่านี้ ได้จัดอยู่ในตารางที่ 8.11 พร้อมทั้งตัวอย่างและความหมายของตัวดำเนินการแต่ละตัว

ตารางที่ 8.11: ตัวดำเนินการในการเปรียบเทียบสำหรับอนุประโยค WHERE

ตัวกระทำ	ชื่อ	ตัวอย่าง	ความหมาย
=	เท่ากับ	id = 3	ทดสอบว่า id เท่ากับ 3 หรือไม่
>	มากกว่า	total > 10	ทดสอบว่า total มากกว่า 10 หรือไม่
<	น้อยกว่า	total < 10	ทดสอบว่า total น้อยกว่า 10 หรือไม่
>=	มากกว่าหรือเท่ากับ	total >= 10	ทดสอบว่า total มากกว่าหรือเท่ากับ 10 หรือไม่
<=	น้อยกว่าหรือเท่ากับ	total <= 10	ทดสอบว่า total น้อยกว่าหรือเท่ากับ 10 หรือไม่
!= or <>	ไม่เท่ากับ	total != 10	ทดสอบว่า total ไม่เท่ากับ 10 หรือไม่
IS NOT NULL	-	total is not null	ทดสอบว่าฟิลด์ total มีค่าอยู่หรือไม่
IS NULL	-	total is null	ทดสอบว่าฟิลด์ total ไม่มีค่าอยู่หรือไม่
BETWEEN	-	total between 0 and 10	ทดสอบว่า total มีค่าตั้งแต่ 0 ถึง 10 หรือไม่
IN	-	name in ("Jack", "Kay")	ทดสอบค่าว่าอยู่ในกลุ่มหรือไม่
NOT IN	-	name not in ("Jack", "Kay")	ทดสอบค่าว่าไม่อยู่ในกลุ่มหรือไม่
LIKE	รูปแบบคล้าย	name like ("Fred %")	ทดสอบว่า name มีรูปแบบคล้าย FRED หรือไม่
NOT LIKE	รูปแบบคล้าย	name not like ("Fred %")	ทดสอบว่า name มีรูปแบบไม่คล้าย FRED
REGEXP	รูปแบบคล้าย	"apa" REGEXP "pilapa"	ทดสอบว่า "apa" มีรูปแบบคล้าย regular expression ผลที่ได้คือ จริง (1)

8.4.5 การขอความช่วยเหลือจาก MySQL

การขอความช่วยเหลือจาก MySQL ทำได้โดยพิมพ์คำว่า help แล้วกดปุ่ม Enter (ไม่มีเครื่องหมาย ; ต่อท้าย help) แต่ถ้าอยู่ที่เครื่องเซิร์ฟเวอร์สามารถดูข้อมูลลึกลงไปถึงเนื้อหาหรือรูปแบบการใช้งานคำสั่งต่าง ๆ ได้ โดยพิมพ์คำว่า help content เช่น ต้องการดูรูปแบบการทำงานของคำสั่ง GRANT ก็พิมพ์คำว่า help grant เป็นต้น

- รูปแบบการแสดงผลเมื่อพิมพ์คำว่า help

```
mysql> help
```

```
For information about MySQL products and services, visit:
  http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Network Support, training, or other products, visit:
  https://shop.mysql.com/
```

```
List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?          (\?) Synonym for 'help'.
clear      (\c) Clear command.
connect    (\r) Reconnect to the server. Optional arguments are db and
           host.
delimiter (\d) Set statement delimiter. NOTE: Takes the rest of the line
           as new delimiter.
ego        (\G) Send command to mysql server, display result vertically.
```

```

exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to mysql server.
help      (\h) Display this help.
notee     (\t) Don't write into outfile.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source    (\.) Execute an SQL script file. Takes a file name as an
           argument.

status    (\s) Get status information from the server.
tee        (\T) Set outfile [to_outfile]. Append everything into given
           outfile.

use        (\u) Use another database. Takes database name as argument.
charset   (\C) Switch to another charset. Might be needed for processing
           binlog with multi-byte charsets.
warnings  (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.
    
```

For server side help, type 'help contents'

- รูปแบบการแสดงผลเมื่อพิมพ์คำว่า help describe

```
mysql> help describe
```

```

Name: 'DESCRIBE'
Description:
Syntax:
      {DESCRIBE | DESC} tbl_name [col_name | wild]
    
```

DESCRIBE provides information about the columns in a table. It is a shortcut for SHOW COLUMNS FROM. As of MySQL 5.0.1, these statements also display information for views. (See [show-columns].)

สรุป

การเข้าถึงระบบฐานข้อมูล MySQL สามารถทำได้โดยผ่าน Command Prompt ที่สามารถสร้างฐานข้อมูลและตาราง (CREATE) แสดงฐานข้อมูลหรือตารางทั้งหมด (SHOW) ใส่ค่าในตาราง (INSERT) เปลี่ยนชื่อคอลัมน์และตาราง (ALTER) เรียกดูโครงสร้างตาราง (DESCRIBE หรือ DESC) เรียกดูข้อมูลจากตาราง (SELECT) แก้ไขข้อมูล (UPDATE) ลบข้อมูลทั้งหมดในตารางทิ้ง (TRUNCATE) กำหนดสิทธิ์ (GRANT) ถอนสิทธิ์ (REVOKE) รวมไปถึงการลบตารางหรือฐานข้อมูลทิ้ง (DROP) ซึ่งการใช้คำสั่งต่าง ๆ เหล่านี้ ผู้ใช้ต้องพิมพ์คำสั่งเอง อาจทำให้เกิดความล่าช้าหรือยุ่งยาก แต่วิธีการเช่นนี้จะช่วยให้ผู้ใช้ได้ฝึกทักษะในการใช้คำสั่งสอบถามข้อมูลที่เรียกว่า SQL ซึ่งในอนาคตต่อไปจะใช้คำสั่ง PHP ควบคู่กับการใช้คำสั่งสอบถาม SQL ในการติดต่อกับระบบฐานข้อมูล

แบบฝึกหัด

1. เข้าสู่ระบบฐานข้อมูลโดยใช้ Command Prompt แล้วสร้างฐานข้อมูลชื่อว่า "WAbxxxxxx" (นั่นคือตัวอักษร WA แล้วตามด้วยรหัสนักศึกษา) แล้วสร้างตารางชื่อว่า guestbook โดยออกแบบชนิดข้อมูลให้สอดคล้องกับข้อมูลที่รับในฟอร์มจากปฏิบัติการที่ผ่านมาชื่อว่า guestbook.html

รูปที่ 8.43: แบบฟอร์ม guestbook

2. เรียกดูโครงสร้างของตาราง guestbook (คำสั่ง DESC หรือ DESCRIBE)
3. ใส่ข้อมูลต่อไปนี้ลงในตาราง guestbook (โดยเปลี่ยนเวลาให้อยู่ในรูปแบบของ DATETIME)

```
Mr.Somchai Jaidee B1233333 Computer Engineering somchai@hotmail.com 12:23, 20th May
Ms.Sripen Deemak B1234444 Electrical Engineering sripen@hotmail.com 12:24, 20th May
Ms.Jaiya Yoodee B1234567 Civil Engineering jaiya@gmail.com 12:25, 20th May
Mr.Sombat Sricharoen B1235555 Electrical Engineering sombat@thaimail.com 16:28, 31st May
```

4. เรียกดูข้อมูลทั้งหมดในตาราง guestbook (คำสั่ง SELECT)
5. สร้างตารางชื่อว่า student_info โดยออกแบบชนิดข้อมูลให้สอดคล้องกับข้อมูลที่รับในแบบฟอร์มจากปฏิบัติการที่ผ่านมาชื่อว่า form_info.php

รูปที่ 8.44: แบบฟอร์ม student_info

6. ใส่ข้อมูล 4 ข้อมูลในตาราง student_info (ชื่อและนามสกุลทั้ง 4 เหมือนกับข้อ 3 ส่วนวันเดือน ปีเกิด และที่อยู่ให้กำหนดขึ้นเอง)
7. เรียกดูข้อมูลทั้งหมดในตาราง guestbook และใช้เงื่อนไขต่าง ๆ เช่น เฉพาะสาขา Computer Engineering เป็นต้น (คำสั่ง SELECT และ WHERE)
8. เรียกดูข้อมูลทั้งหมดในตาราง student_info และใช้เงื่อนไขต่าง ๆ เช่น เฉพาะคนที่เกิดก่อนวันที่ 15 เป็นต้น (คำสั่ง SELECT และ WHERE)
9. เรียกดูข้อมูลจาก 2 ตาราง โดยใช้เงื่อนไข เช่น เฉพาะสาขา Computer Engineering และเกิดก่อนวันที่ 15 เป็นต้น (คำสั่ง SELECT และ WHERE)
10. ทดลองแก้ไขข้อมูลในตารางทั้ง 2 เช่น เปลี่ยนข้อมูลในตาราง เปลี่ยนชื่อตาราง ลบข้อมูลทั้งหมดในตาราง เป็นต้น (คำสั่ง UPDATE)



บทที่ 9 การใช้ PHP และ MySQL

<Using PHP & MySQL>

บทนำ

การใช้ภาษาสอบถาม SQL เพื่อเข้าถึงระบบฐานข้อมูลผ่าน Command Prompt เป็นทางเลือกทางหนึ่งที่มีความสำคัญ เนื่องจากเป็นการฝึกฝนเพื่อให้เกิดทักษะในการเรียกใช้คำสั่งต่าง ๆ แต่ในบทเรียนนี้จะนำเสนอการติดต่อฐานข้อมูลและใช้คำสั่งต่าง ๆ ผ่านหน้าเว็บ ซึ่งใช้สคริปต์ PHP เป็นเครื่องมือช่วยในการติดต่อ ไม่ว่าจะเป็นการลงทะเบียนเข้าสู่ระบบ การสร้างฐานข้อมูล สร้างตาราง กำหนดสิทธิ์ บันทึกข้อมูล หรือประมวลผลข้อมูล เป็นต้น การติดต่อฐานข้อมูลในลักษณะนี้ช่วยให้ผู้ใช้งานในทุกระดับมีความสะดวกมากยิ่งขึ้น เนื่องจากระบบจะทำการประมวลผลผ่านทางเว็บ ดังนั้นผู้ใช้งานสามารถเข้าถึงระบบฐานข้อมูลได้จากทุก ๆ แห่งที่มีการเชื่อมต่อระบบอินเทอร์เน็ต

วัตถุประสงค์

1. เพื่อศึกษาการใช้คำสั่งและฟังก์ชันต่าง ๆ ของสคริปต์ PHP ในการติดต่อระบบฐานข้อมูล MySQL และคำสั่งสอบถาม SQL
2. เพื่อทดลองใช้คำสั่งและฟังก์ชันต่าง ๆ ในการติดต่อกับระบบฐานข้อมูล สร้างฐานข้อมูล ประมวลผล และนำเสนอข้อมูลรายละเอียดต่าง ๆ ผ่านทางเว็บ
3. เพื่อประยุกต์ใช้คำสั่งและฟังก์ชันต่าง ๆ ของสคริปต์ PHP กับคำสั่งสอบถาม ในรูปแบบต่าง ๆ ผ่านทางเว็บ

เนื้อหา

ระบบฐานข้อมูลที่มีการเชื่อมต่อทางเว็บนั้น จำเป็นที่จะต้องมีการกำหนดชื่อฐานข้อมูล รหัสผู้ใช้งาน รหัสผ่าน พร้อมทั้งสิทธิ์ต่าง ๆ ของผู้ใช้งานด้วย เพื่อความปลอดภัยของระบบ ซึ่งในบทนี้จะแนะนำถึงการใช้คำสั่งต่าง ๆ ของ PHP ในการติดต่อฐานข้อมูลผ่านทางเว็บร่วมกับคำสั่งภาษาสอบถาม SQL และแนะนำรูปแบบฟังก์ชันต่าง ๆ ของ PHP ที่มีส่วนเกี่ยวข้องกับระบบฐานข้อมูล

9.1 ฟังก์ชันเกี่ยวกับฐานข้อมูล MySQL

ฟังก์ชันต่าง ๆ ของ PHP ที่เกี่ยวข้องกับการติดต่อระบบฐานข้อมูล MySQL นั้นมีมาก แต่จะขอสรุปและยกตัวอย่างเฉพาะฟังก์ชันที่นิยมและถูกเรียกใช้งานมาก ดังในตารางที่ 9.1

ตารางที่ 9.1: ฟังก์ชันต่าง ๆ เกี่ยวกับฐานข้อมูล MySQL

ฟังก์ชัน	รูปแบบ	ความหมาย
mysql_affected_rows	int mysql_affected_rows ([resource link_identifier])	ดึงข้อมูลตัวเลขของจำนวนแถวจากการใช้คำสั่งของ MySQL
mysql_change_user	int mysql_change_user (string user, string password [, string database [, resource link_identifier]])	เปลี่ยนชื่อผู้ใช้ในขณะที่มีการเชื่อมต่อกับระบบอยู่
mysql_client_encoding	string mysql_client_encoding ([resource link_identifier])	แสดงชื่อของกลุ่มชนิดตัวอักษรทางฝั่ง client
mysql_close	bool mysql_close ([resource link_identifier])	ปิดการติดต่อระบบฐานข้อมูล
mysql_connect	resource mysql_connect ([string server [, string username [, string password [, bool new_link [, int client_flags]]]])	เปิดการเชื่อมต่อระบบฐานข้อมูล MySQL
mysql_create_db	bool mysql_create_db (string database_name [, resource link_identifier])	สร้างฐานข้อมูล
mysql_data_seek	bool mysql_data_seek (resource result, int row_number)	เลื่อนตัวชี้ผลลัพธ์ภายใน
mysql_db_name	string mysql_db_name (resource result, int row [, mixed field])	แสดงชื่อฐานข้อมูล
mysql_db_query	resource mysql_db_query (string database, string query [, resource link_identifier])	ส่งคำถาม (query) ไปที่ MySQL
mysql_drop_db	bool mysql_drop_db (string database_name [, resource link_identifier])	ลบฐานข้อมูลทิ้ง
mysql_errno	int mysql_errno ([resource link_identifier])	แสดงค่าจำนวนตัวเลขข้อผิดพลาดจากคำสั่งที่ผ่านมา
mysql_error	string mysql_error ([resource link_identifier])	แสดงข้อความข้อผิดพลาดจากคำสั่งที่ผ่านมา
mysql_escape_string	string mysql_escape_string (string unescaped_string)	แสดงข้อความแบบ escape string
mysql_fetch_array	array mysql_fetch_array (resource result [, int result_type])	ดึงข้อมูลแถวออกมาเป็นอะเรย์สัมพันธ์ อะเรย์ หรือทั้ง 2 แบบ แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป (แถวใหม่)
mysql_fetch_assoc	array mysql_fetch_assoc (resource result)	ดึงข้อมูลจำนวนแถวออกมาเป็นอะเรย์สัมพันธ์
mysql_fetch_field	object mysql_fetch_field (resource result [, int field_offset])	ดึงข้อมูลฟิลด์หรือคอลัมน์ออกเป็นอ็อบเจกต์
mysql_fetch_lengths	array mysql_fetch_lengths (resource result)	ดึงข้อมูลความยาวของแต่ละฟิลด์หรือคอลัมน์ออกเป็นอะเรย์
mysql_fetch_object	object mysql_fetch_object (resource result [, string class_name [, array params]])	ดึงข้อมูลจำนวนแถวออกมาเป็นอ็อบเจกต์
mysql_fetch_row	array mysql_fetch_row (resource result)	ดึงข้อมูลจำนวนแถวออกมาเป็นอะเรย์
mysql_field_len	int mysql_field_len (resource result, int field_offset)	แสดงค่าความยาวของฟิลด์ที่ระบุจำนวน offset
mysql_field_name	string mysql_field_name (resource result, int field_offset)	แสดงชื่อของฟิลด์หรือคอลัมน์ที่ระบุจำนวน offset
mysql_field_seek	bool mysql_field_seek (resource result, int field_offset)	ตั้งตัวชี้ตำแหน่งฟิลด์โดยระบุจำนวน offset
mysql_field_table	string mysql_field_table (resource result, int field_offset)	แสดงชื่อของตารางที่ฟิลด์นั้นถูกระบุอยู่
mysql_field_type	string mysql_field_type (resource result, int field_offset)	แสดงชนิดของฟิลด์ที่ระบุ
mysql_free_result	bool mysql_free_result (resource result)	คืนหน่วยความจำให้กับระบบ
mysql_get_client_info	string mysql_get_client_info (void)	ดึงข้อมูลรายละเอียดจาก client
mysql_get_host_info	string mysql_get_host_info ([resource link_identifier])	ดึงข้อมูลรายละเอียดจาก host
mysql_get_server_info	string mysql_get_server_info ([resource link_identifier])	ดึงข้อมูลรายละเอียดของ MySQL Server
mysql_info	string mysql_info ([resource link_identifier])	ดึงข้อมูลเกี่ยวกับคำถาม (query) ที่มีการใช้ครั้งล่าสุด
mysql_list_dbs	resource mysql_list_dbs ([resource link_identifier])	แสดงชื่อฐานข้อมูลในระบบ MySQL
mysql_list_fields	resource mysql_list_fields (string database_name, string table_name [, resource link_identifier])	แสดงชื่อของฟิลด์หรือคอลัมน์
mysql_list_processes	resource mysql_list_processes ([resource link_identifier])	แสดงชื่อรายการของการดำเนินการ (Processes)
mysql_list_tables	resource mysql_list_tables (string database [, resource link_identifier])	แสดงชื่อของตาราง
mysql_num_fields	int mysql_num_fields (resource result)	แสดงจำนวนของฟิลด์หรือคอลัมน์

ฟังก์ชัน	รูปแบบ	ความหมาย
mysql_num_rows	int mysql_num_rows (resource result)	แสดงจำนวนแถวข้อมูล
mysql_pconnect	resource mysql_pconnect ([string server [, string username [, string password [, int client_flags]]]])	เปิดการเชื่อมต่อระบบฐานข้อมูล MySQL แบบ Persistent
mysql_ping	bool mysql_ping ([resource link_identifier])	ใช้คำสั่ง Ping เพื่อทดสอบการติดต่อระบบ
mysql_query	resource mysql_query (string query [, resource link_identifier])	ส่งภาษาสอบถามไปที่ MySQL
mysql_real_escape_string	string mysql_real_escape_string (string unescaped_string [, resource link_identifier])	แสดงข้อความแบบ escape string ที่ใช้ในภาษาสอบถาม SQL
mysql_result	string mysql_result (resource result, int row [, mixed field])	แสดงผลลัพธ์
mysql_select_db	bool mysql_select_db (string database_name [, resource link_identifier])	เลือกฐานข้อมูล
mysql_stat	string mysql_stat ([resource link_identifier])	แสดงสถานะของระบบในปัจจุบัน

9.2 ตัวอย่างการใช้งานฐานข้อมูล MySQL

จากบทที่ผ่านมา เราได้สร้างฐานข้อมูลชื่อว่า webapp150 ขึ้นมา ซึ่งการเข้าใช้ระบบฐานข้อมูลได้กำหนดชื่อผู้ใช้คือ root รหัสผ่านคือ meroot ดังนั้นเราจะทดลองติดต่อฐานข้อมูลนี้ แล้วดูรายละเอียดต่าง ๆ ดังในตัวอย่างต่อไปนี้

ตัวอย่าง การใช้คำสั่ง PHP เชื่อมต่อระบบฐานข้อมูล

```

1. <?php
2.   $host = "localhost";
3.   $user = "root";
4.   $password = "meroot";
5.
6.   $link = mysql_connect($host,$user,$password);
7.   if (!$link) {
8.     die('Could not connect: ' .mysql_error());
9.   }
10.  echo "Connected successfully";
11.  mysql_close($link);
12.  ?>

```



รูปที่ 9.1: การใช้คำสั่ง PHP เชื่อมต่อระบบฐานข้อมูล

- บรรทัดที่ 2 กำหนดค่าสตริง localhost ให้กับตัวแปร \$host
- บรรทัดที่ 3 กำหนดค่าสตริง root ให้กับตัวแปร \$user
- บรรทัดที่ 4 กำหนดค่าสตริง meroot ให้กับตัวแปร \$password
- บรรทัดที่ 6 เรียกใช้ฟังก์ชัน mysql_connect() เพื่อทำการเชื่อมต่อระบบฐานข้อมูลที่ถูกเก็บอยู่ในตัวแปร \$host โดยมีรหัสผู้ใช้คือค่าที่ถูกเก็บอยู่ในตัวแปร \$user และรหัสผ่านคือค่าที่ถูกเก็บอยู่ในตัวแปร \$password แล้วให้เก็บผลที่ได้ไว้ที่ตัวแปร \$link

บรรทัดที่ 7 ทดสอบการเชื่อมต่อระบบฐานข้อมูล ถ้าเชื่อมต่อระบบฐานข้อมูลไม่ได้ ให้ไปทำคำสั่งในบรรทัดที่ 8 มิฉะนั้นแล้วให้ข้ามไปทำคำสั่งในบรรทัดที่ 10 แทน

บรรทัดที่ 8 เรียกใช้ฟังก์ชัน die() (มีการทำงานเหมือนกับฟังก์ชัน exit()) คือแสดงข้อความผิดพลาดแล้วหยุดการทำงานของสคริปต์) เพื่อแสดงข้อความ “Could not connect:” แล้วตามด้วยการเรียกใช้ฟังก์ชัน mysql_error() เพื่อแสดงข้อผิดพลาดออกที่จอภาพแล้วหยุดการทำงานของสคริปต์ (เช่น ดังแสดงในรูปที่ 9.2)

บรรทัดที่ 10 แสดงข้อความ “Connected successfully” ออกที่จอภาพ

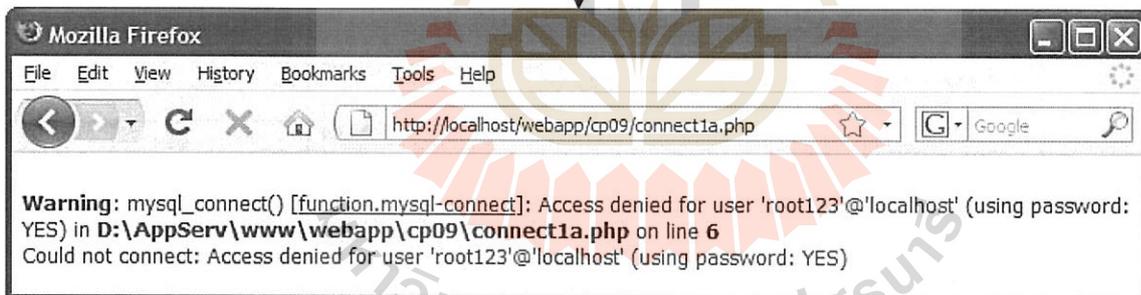
บรรทัดที่ 11 เรียกใช้ฟังก์ชัน mysql_close() เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ถูกเก็บอยู่ในตัวแปร \$link

ตัวอย่าง การใช้คำสั่ง PHP เชื่อมต่อระบบฐานข้อมูล โดยการเปลี่ยนชื่อผู้ใช้จากเดิมคือ root เป็น root123 เพื่อทดสอบการแสดงผลข้อผิดพลาดในการเชื่อมต่อ เนื่องจากไม่มีชื่อผู้ใช้ root123

```

1. <?php
2.     $host = "localhost";
3.     $user = "root123";
4.     $password = "meroot";
5.
6.     $link = mysql_connect($host,$user,$password);
7.     if (!$link) {
8.         die('Could not connect: ' .mysql_error());
9.     }
10.    echo "Connected successfully";
11.    mysql_close($link);
12.    ?>

```



รูปที่ 9.2: การแสดงข้อผิดพลาดในการเชื่อมต่อระบบฐานข้อมูล

หมายเหตุ การเชื่อมต่อระบบฐานข้อมูลในสคริปต์หนึ่ง ๆ เมื่อการทำงานของสคริปต์นั้นสิ้นสุดลง จะทำให้การเชื่อมต่อสิ้นสุดลงด้วย ถึงแม้จะไม่มีคำสั่งเรียกใช้ฟังก์ชัน mysql_close() เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลก็ตาม ดังนั้นจึงมีความจำเป็นที่จะต้องเปิดการเชื่อมต่อในทุก ๆ สคริปต์ที่ต้องการใช้ระบบฐานข้อมูล วิธีที่สะดวกที่สุดคือการบันทึกคำสั่งต่าง ๆ ในการเชื่อมต่อระบบฐานข้อมูลเป็นแฟ้มข้อมูลสกุล .inc แล้วใช้คำสั่ง include() ในการรวมแฟ้มข้อมูลนั้นเข้าไป

จากรูปที่ 9.1 นั้น เราสามารถสร้างแฟ้มข้อมูลเป็นสกุล .inc เพื่อเชื่อมต่อระบบฐานข้อมูลได้ดังแสดงในรูปที่ 9.3

```

1. <?php
2.   $host = "localhost";
3.   $user = "root";
4.   $password = "meroot";
5.
6.   $link = mysql_connect($host,$user,$password);
7.   if (!$link) {
8.     die('Could not connect: ' .mysql_error());
9.   }
10.  ?>
    
```

รูปที่ 9.3: แฟ้มข้อมูลสกุล .inc เพื่อเชื่อมต่อระบบฐานข้อมูล

ตัวอย่าง การรวมแฟ้มสกุล .inc เพื่อใช้ในการเชื่อมต่อระบบฐานข้อมูลโดยใช้คำสั่ง include() หลังจากนั้นจะใช้คำสั่งสอบถาม show databases; แสดงฐานข้อมูลทั้งหมดที่มีอยู่ในระบบ

```

1. <?php
2.   include("connect.inc");
3.   $sql = "show databases";
4.   $db_list = mysql_query($sql);
5.   while ($row = mysql_fetch_object($db_list)) {
6.     echo $row->Database . "<br>";
7.   }
8.   mysql_close($link);
9.  ?>
    
```



รูปที่ 9.4: การรวมแฟ้มข้อมูลสกุล .inc ในการติดต่อระบบฐานข้อมูลแล้วใช้ PHP ในการแสดงฐานข้อมูลทั้งหมดในระบบ

- บรรทัดที่ 2 ใช้คำสั่ง include() เพื่อรวมแฟ้มข้อมูล connect.inc เข้ามา
- บรรทัดที่ 3 กำหนดค่าสตริง show databases ให้กับตัวแปร \$sql
- บรรทัดที่ 4 เรียกใช้ฟังก์ชัน mysql_query() เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร \$sql แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$db_list
- บรรทัดที่ 5 ใช้ลูป while ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะมีการใช้ฟังก์ชัน mysql_fetch_object() เพื่อทำการดึงข้อมูลที่อยู่ในตัวแปร \$db_list ออกมาเป็นอ็อบเจกต์แล้วนำไปเก็บไว้ในตัวแปร \$row
- บรรทัดที่ 6 แสดงข้อมูลอ็อบเจกต์ที่อยู่ในตัวแปร \$row โดยใช้ชื่อฟิลด์หรือคอลัมน์ Database แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป

บรรทัดที่ 8 เรียกใช้ฟังก์ชัน `mysql_close()` เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร `$link` ซี่งอยู่

หมายเหตุ การเรียกใช้ฟังก์ชัน `mysql_fetch_object()` ในบรรทัดที่ 5 นั้นสามารถเปลี่ยนเป็นฟังก์ชัน `mysql_fetch_row()` ได้ แล้วค่าที่ได้จะเป็นอะเรย์ เมื่อต้องการแสดงผลข้อมูลที่อยู่ในตัวแปร `$row` ต้องใช้อินเด็กซ์ที่เป็นตัวเลข ซึ่งในกรณีนี้คือ 0 เท่านั้น เนื่องจากในแต่ละแถวมีข้อมูลเดียว ดังรูปแบบต่อไปนี้

```
while($row = mysql_fetch_row($db_list)){
    echo $row[0]. "<br>";
}
```

ตัวอย่าง การลบฐานข้อมูลชื่อ `webapp15000` ออกจากระบบฐานข้อมูล

```
1. <?php
2.     include("connect.inc");
3.     $sql = "drop database webapp15000";
4.     $result = mysql_query($sql);
5.     if(!$result){
6.         die('Could not delete: ' . mysql_error());
7.     }
8.     echo "Deleted successfully !!!<br>";
9.     $sql = "show databases";
10.    $db_list = mysql_query($sql);
11.    while ($row = mysql_fetch_row($db_list)) {
12.        echo $row[0] . "<br>";
13.    }
14.    mysql_close($link);
15. ?>
```



รูปที่ 9.5: การลบฐานข้อมูล `webapp15000` ออกจากระบบฐานข้อมูล

บรรทัดที่ 2 ใช้คำสั่ง `include()` เพื่อรวมแฟ้มข้อมูล `connect.inc` เข้ามา

บรรทัดที่ 3 กำหนดคำสั่งจริง `drop databases webapp15000` ให้กับตัวแปร `$sql`

บรรทัดที่ 4 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$result`

บรรทัดที่ 5 ตรวจสอบผลที่อยู่ในตัวแปร `$result` ว่าทำงานได้ถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะเข้าไปทำคำสั่งในบรรทัดที่ 6 แต่ถ้าถูกต้องก็จะข้ามไปทำคำสั่งในบรรทัดที่ 8 แทน

บรรทัดที่ 6 เรียกใช้ฟังก์ชัน `die()` ซึ่งมีการทำงานเหมือนกันกับฟังก์ชัน `exit()` เพื่อแสดงข้อความว่า “Could not delete:” แล้วตามด้วยการเรียกใช้ฟังก์ชัน `mysql_error()` เพื่อแสดงข้อผิดพลาดที่เกี่ยวข้องแล้วหยุดการทำงานของสคริปต์

บรรทัดที่ 8 แสดงข้อความ “Deleted successfully !!!” แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดต่อไป

บรรทัดที่ 9 กำหนดคำสั่ง `show databases` ให้กับตัวแปร `$sql`

บรรทัดที่ 10 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$db_list`

บรรทัดที่ 11 ใช้ลูป `while` ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะมีการใช้ฟังก์ชัน `mysql_fetch_row()` เพื่อทำการดึงข้อมูลที่อยู่ในตัวแปร `$db_list` ออกมาเป็นอะเรย์แล้วนำไปเก็บไว้ในตัวแปร `$row`

บรรทัดที่ 12 แสดงข้อมูลอะเรย์ที่อยู่ในตัวแปร `$row` โดยใช้อินเด็กซ์ที่เป็นตัวเลข ซึ่งในกรณีนี้คือ 0 เท่านั้น เนื่องจากในแต่ละแถวมีข้อมูลเดียว แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป

บรรทัดที่ 14 เรียกใช้ฟังก์ชัน `mysql_close()` เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร `$link` ใช้อยู่

หมายเหตุ ถ้าไม่สามารถลบฐานข้อมูลได้จะแสดงผลดังนี้

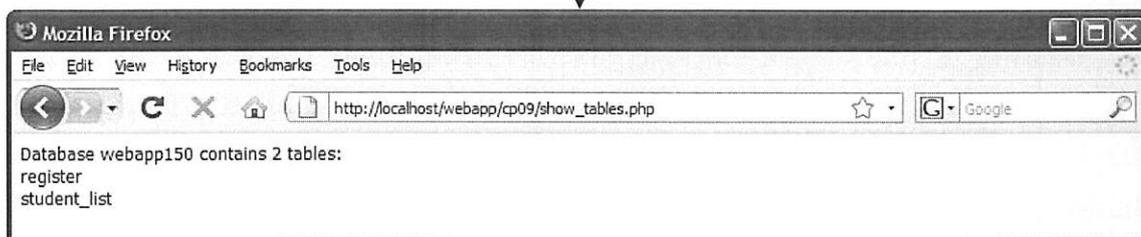
```
Could not delete: Can't drop database 'webapp15000';
database doesn't exist
```

ตัวอย่าง การเรียกใช้ฐานข้อมูลชื่อ `webapp150` พร้อมทั้งแสดงชื่อตารางทั้งหมด โดยใช้คำสั่งสอบถาม SQL

```

1. <?php
2.     include("connect.inc");
3.     $db = "webapp150";
4.     $sql = "show tables from $db";
5.     $result = mysql_query($sql);
6.     if(!$result){
7.         die('Could not find database called webapp150: '. mysql_error());
8.     }
9.     $list_table = mysql_list_tables($db);
10.    $num_rows = mysql_num_rows($list_table);
11.    echo "Database $db contains $num_rows tables:<br>";
12.    while($row = mysql_fetch_row($result)){
13.        echo "$row[0]<br>";
14.    }
15.    mysql_close($link);
16.    ?>

```



รูปที่ 9.6: การเรียกใช้ฐานข้อมูลชื่อ `webapp150` พร้อมทั้งแสดงชื่อตารางทั้งหมด

บรรทัดที่ 2 ใช้คำสั่ง include() เพื่อรวมแฟ้มข้อมูล connect.inc เข้ามา

บรรทัดที่ 3 กำหนดค่าสตริง webapp150 ให้กับตัวแปร \$db

บรรทัดที่ 4 กำหนดค่าสตริง show tables from \$db ให้กับตัวแปร \$sql

บรรทัดที่ 5 เรียกใช้ฟังก์ชัน mysql_query() เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร \$sql แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result

บรรทัดที่ 6 ตรวจสอบผลที่อยู่ในตัวแปร \$result ว่าทำงานได้ถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะเข้าไปทำคำสั่งในบรรทัดที่ 7 แต่ถ้าถูกต้องก็จะข้ามไปทำคำสั่งในบรรทัดที่ 9 แทน

บรรทัดที่ 7 เรียกใช้ฟังก์ชัน die() ซึ่งมีการทำงานเหมือนกันกับฟังก์ชัน exit() เพื่อแสดงข้อความว่า "Could not find database called webapp150:" แล้วตามด้วยการเรียกใช้ฟังก์ชัน mysql_error() เพื่อแสดงข้อผิดพลาดที่เกี่ยวข้องแล้วหยุดการทำงานของสคริปต์

บรรทัดที่ 9 เรียกใช้ฟังก์ชัน mysql_list_tables() เพื่อดึงข้อมูลที่เป็นชื่อตารางมาจากฐานข้อมูล webapp150 แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$list_table

บรรทัดที่ 10 เรียกใช้ฟังก์ชัน mysql_num_rows() เพื่อดึงข้อมูลจำนวนแถวที่มีอยู่ในตัวแปร \$list_table แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$num_rows

บรรทัดที่ 11 แสดงข้อความบอกว่าฐานข้อมูลนั้นมีกี่ตาราง แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดต่อไป

บรรทัดที่ 12 ใช้ลูป while ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะมีการใช้ฟังก์ชัน mysql_fetch_row() เพื่อทำการดึงข้อมูลที่อยู่ในตัวแปร \$result ออกมาเป็นอะเรย์แล้วนำไปเก็บไว้ในตัวแปร \$row

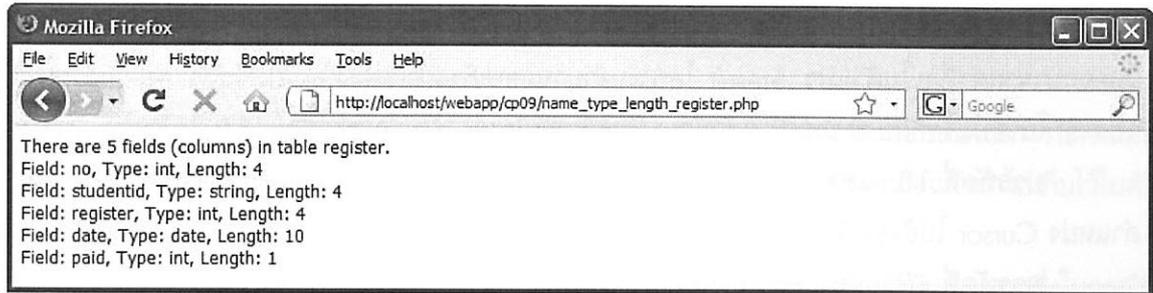
บรรทัดที่ 13 แสดงข้อมูลอะเรย์ที่อยู่ในตัวแปร \$row โดยใช้อินเด็กซ์ที่เป็นตัวเลข ซึ่งในกรณีนี้คือ 0 เท่านั้น เนื่องจากในแต่ละแถวมีข้อมูลเดียว แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป

บรรทัดที่ 15 เรียกใช้ฟังก์ชัน mysql_close() เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร \$link ซ้ำอยู่

ตัวอย่าง การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์ ชนิด และขนาดในตาราง register

```

1.  <?php
2.      include("connect.inc");
3.      $db = "webapp150";
4.      $tb = "register";
5.      $result = mysql_select_db($db); // mysql_query("use $db");
6.      if(!$result){
7.          die('Could not find database called webapp150: ' . mysql_error());
8.      }
9.      $sql = "select * from $tb";
10.     $result = mysql_query($sql);
11.     $num_fields = mysql_num_fields($result);
12.     echo "There are $num_fields fields (columns) in table $tb.<br>";
13.     $i = 0;
14.     while($i<$num_fields){
15.         $name[$i] = mysql_field_name($result,$i);
16.         $type[$i] = mysql_field_type($result,$i);
17.         $len[$i] = mysql_field_len($result,$i);
18.         echo "Field: $name[$i], Type: $type[$i], Length: $len[$i]<br>";
19.         $i++;
20.     }
21.     mysql_close($link);
22.  ?>
    
```



รูปที่ 9.7: การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์ ชนิด และขนาดในตาราง register

- บรรทัดที่ 2 ใช้คำสั่ง include() เพื่อรวมแฟ้มข้อมูล connect.inc เข้ามา
- บรรทัดที่ 3 กำหนดค่าสตริง webapp150 ให้กับตัวแปร \$db
- บรรทัดที่ 4 กำหนดค่าสตริง register ให้กับตัวแปร \$tb
- บรรทัดที่ 5 เรียกใช้ฟังก์ชัน mysql_select_db() เพื่อเลือกใช้ฐานข้อมูลชื่อว่า webapp150 (ค่าที่อยู่ในตัวแปร \$db) แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result
- บรรทัดที่ 6 ตรวจสอบผลที่อยู่ในตัวแปร \$result ว่าทำงานได้ถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะเข้าไปทำคำสั่งในบรรทัดที่ 7 แต่ถ้าถูกต้องก็จะข้ามไปทำคำสั่งในบรรทัดที่ 9 แทน
- บรรทัดที่ 7 เรียกใช้ฟังก์ชัน die() ซึ่งมีการทำงานเหมือนกันกับฟังก์ชัน exit() เพื่อแสดงข้อความว่า "Could not find database called webapp150:" แล้วตามด้วยการเรียกใช้ฟังก์ชัน mysql_error() เพื่อแสดงข้อผิดพลาดที่เกี่ยวข้องแล้วหยุดการทำงานของสคริปต์
- บรรทัดที่ 9 กำหนดค่าสตริง select * from \$tb ให้กับตัวแปร \$sql ซึ่งเป็นคำสั่ง sql หมายถึงการเลือกข้อมูลทั้งหมดในตารางที่ชื่อว่า register
- บรรทัดที่ 10 เรียกใช้ฟังก์ชัน mysql_query() เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร \$sql แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result
- บรรทัดที่ 11 เรียกใช้ฟังก์ชัน mysql_num_fields() เพื่อดึงข้อมูลจำนวนฟิลด์หรือจำนวนคอลัมน์ แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$num_fields
- บรรทัดที่ 12 แสดงข้อความบอกว่ามีกี่ฟิลด์หรือคอลัมน์ในตารางที่ระบุ แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดต่อไป
- บรรทัดที่ 13 กำหนดค่า 0 (ศูนย์) ให้กับตัวแปร \$i
- บรรทัดที่ 14 ใช้ลูป while ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะตรวจสอบดูว่าค่าที่อยู่ในตัวแปร \$i น้อยกว่าค่าที่อยู่ในตัวแปร \$num_fields จริงหรือไม่ ถ้าจริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 15 - 19 แต่ถ้าไม่จริงก็จะข้ามไปทำคำสั่งในบรรทัดที่ 21 แทน
- บรรทัดที่ 15 เรียกใช้ฟังก์ชัน mysql_field_name() เพื่อดึงข้อมูลชื่อของฟิลด์หรือคอลัมน์นั้น ออกมาจากค่าที่อยู่ในตัวแปร \$result โดยระบุจำนวนออฟเซตเท่ากับค่าที่อยู่ในตัวแปร \$i ซึ่งค่าของตัวแปร \$i ในรอบแรกก็คือ 0 นั่นเอง แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$name[\$i]
- บรรทัดที่ 16 เรียกใช้ฟังก์ชัน mysql_field_type() เพื่อดึงข้อมูลชนิดของฟิลด์หรือคอลัมน์นั้น ออกมาจากค่าที่อยู่ในตัวแปร \$result โดยระบุจำนวนออฟเซตเท่ากับค่าที่อยู่ในตัวแปร \$i ซึ่งค่าของตัวแปร \$i ในรอบแรกก็คือ 0 นั่นเอง แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$type[\$i]

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน `mysql_field_len()` เพื่อดึงข้อมูลความยาวของฟิลด์หรือคอลัมน์นั้นออกมาจากค่าที่อยู่ในตัวแปร `$result` โดยระบุจำนวนออฟเซตเท่ากับค่าที่อยู่ในตัวแปร `$i` ซึ่งค่าของตัวแปร `$i` ในรอบแรกก็คือ 0 นั่นเอง แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$len[$i]`

บรรทัดที่ 18 แสดงข้อความบอกชื่อฟิลด์หรือคอลัมน์ บอกชนิด และบอกความยาว แล้วเลื่อนตำแหน่ง Cursor ไปยังจุดเริ่มต้นของบรรทัดต่อไป

บรรทัดที่ 19 เพิ่มค่าให้กับตัวแปร `$i` อีก 1

บรรทัดที่ 21 เรียกใช้ฟังก์ชัน `mysql_close()` เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร `$link` ใช้อยู่

ตัวอย่าง การเรียกใช้ฐานข้อมูลชื่อ `webapp150` พร้อมทั้งแสดงชื่อคอลัมน์โดยใช้ภาษาสอบถาม SQL

```

1.  <?php
2.  include("connect.inc");
3.  $db = "webapp150";
4.  $sql = "use $db";
5.  $result = mysql_query($sql);
6.  if(!$result){
7.      die('Could not find database called webapp150: '.mysql_error());
8.  }
9.  $sql = "show columns from register";
10. $result = mysql_query($sql);
11. while($row = mysql_fetch_row($result)) {
12.     echo "$row[0] | ";
13. }
14. mysql_close($link);
15. ?>
    
```



รูปที่ 9.8: การเรียกใช้ฐานข้อมูลชื่อ `webapp150` พร้อมทั้งแสดงชื่อคอลัมน์โดยใช้ภาษาสอบถาม SQL

บรรทัดที่ 2 ใช้คำสั่ง `include()` เพื่อรวมแฟ้มข้อมูล `connect.inc` เข้ามา

บรรทัดที่ 3 กำหนดค่าสตริง `webapp150` ให้กับตัวแปร `$db`

บรรทัดที่ 4 กำหนดค่าสตริง `use $db` ให้กับตัวแปร `$sql`

บรรทัดที่ 5 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$result`

บรรทัดที่ 6 ตรวจสอบผลที่อยู่ในตัวแปร `$result` ว่าทำงานได้ถูกต้องหรือไม่ ถ้าไม่ถูกต้องจะเข้าไปทำคำสั่งในบรรทัดที่ 7 แต่ถ้าถูกต้องก็จะข้ามไปทำคำสั่งในบรรทัดที่ 9 แทน

บรรทัดที่ 7 เรียกใช้ฟังก์ชัน `die()` ซึ่งมีการทำงานเหมือนกับฟังก์ชัน `exit()` เพื่อแสดงข้อความว่า "Could not find database called webapp150:" แล้วตามด้วยการเรียกใช้ฟังก์ชัน `mysql_error()` เพื่อแสดงข้อผิดพลาดที่เกี่ยวข้องแล้วหยุดการทำงานของสคริปต์

บรรทัดที่ 9 กำหนดค่าสตริง `show columns from register` ให้กับตัวแปร `$sql` ซึ่งเป็นคำสั่ง `sql` หมายถึงการแสดงชื่อฟิลด์หรือคอลัมน์ทั้งหมดที่อยู่ในตารางที่ชื่อว่า `register`

บรรทัดที่ 10 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$result`

บรรทัดที่ 11 ใช้หุ้บ `while` ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะมีการใช้ฟังก์ชัน `mysql_fetch_row()` เพื่อทำการดึงข้อมูลที่อยู่ในตัวแปร `$result` ออกมาเป็นอะเรย์แล้วนำไปเก็บไว้ในตัวแปร `$row`

บรรทัดที่ 12 แสดงข้อมูลอะเรย์ที่อยู่ในตัวแปร `$row` โดยใช้อินเด็กซ์ที่เป็นตัวเลข ซึ่งในกรณีนี้คือ 0 เท่านั้น เนื่องจากในแต่ละแถวมีข้อมูลเดียว แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป

บรรทัดที่ 14 เรียกใช้ฟังก์ชัน `mysql_close()` เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร `$link` ซึ้อยู่

ตัวอย่าง การเรียกใช้ฐานข้อมูลชื่อ `webapp150` พร้อมทั้งแสดงชื่อคอลัมน์และรายละเอียดต่าง ๆ ในตาราง `register` แล้วแสดงออกที่จอภาพในรูปแบบของตาราง

```

1. <?php
2.   include("connect.inc");
3.   $db = "webapp150";
4.   $tb = "register";
5.   $result = mysql_select_db($db);
6.   if(!$result){
7.       die('Could not find database called webapp150: ' . mysql_error());
8.   }
9.   $sql = "select * from $tb";
10.  $result = mysql_query($sql);
11.  $num_rows = mysql_num_rows($result);
12.  $num_fields = mysql_num_fields($result);
13.
14.  echo "<center>";
15.  echo "<table width=500px border=1px><tr>";
16.
17.  $sql = "show columns from register";
18.  $col = mysql_query($sql);
19.  while($row = mysql_fetch_row($col)){
20.      echo "<th>$row[0]</th>";
21.  }
22.  echo "</tr>";
23.
24.  $i = 0;
25.  while($i<$num_rows){
26.      $data = mysql_fetch_array($result);
27.      echo "<tr align=\"center\">";
28.      for($j=0; $j<$num_fields; $j++){
29.          echo "<td>$data[$j]</td>";
30.      }
31.      echo "</tr>";
32.      $i++;
33.  }
34.  echo "</table></center>";
35.  mysql_close($link);
36.  ?>

```

no	studentid	register	date	paid
1	0002	18	2007-06-15	0
2	0005	18	2007-06-16	0
3	0001	21	2007-06-16	1
4	0003	15	2007-06-18	1
6	0004	16	2007-06-19	0

รูปที่ 9.9: การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์และรายละเอียดต่าง ๆ ในตาราง register

บรรทัดที่ 2 ใช้คำสั่ง include() เพื่อรวมแฟ้มข้อมูล connect.inc เข้ามา

บรรทัดที่ 3 กำหนดค่าสตริง webapp150 ให้กับตัวแปร \$db

บรรทัดที่ 4 กำหนดค่าสตริง register ให้กับตัวแปร \$tb

บรรทัดที่ 5 เรียกใช้ฟังก์ชัน mysql_select_db() เพื่อเลือกใช้ฐานข้อมูลชื่อว่า webapp150 (ค่าที่อยู่ในตัวแปร \$db) แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result

บรรทัดที่ 6 ตรวจสอบผลที่อยู่ในตัวแปร \$result ว่าทำงานได้ถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะเข้าไปทำคำสั่งในบรรทัดที่ 7 แต่ถ้าถูกต้องก็จะข้ามไปทำคำสั่งในบรรทัดที่ 9 แทน

บรรทัดที่ 7 เรียกใช้ฟังก์ชัน die() ซึ่งมีการทำงานเหมือนกันกับฟังก์ชัน exit() เพื่อแสดงข้อความว่า "Could not find database called webapp150:" แล้วตามด้วยการเรียกใช้ฟังก์ชัน mysql_error() เพื่อแสดงข้อผิดพลาดที่เกี่ยวข้องแล้วหยุดการทำงานของสคริปต์

บรรทัดที่ 9 กำหนดค่าสตริง select * from \$tb ให้กับตัวแปร \$sql ซึ่งเป็นคำสั่ง sql หมายถึงการเลือกข้อมูลทั้งหมดในตารางที่ชื่อว่า register

บรรทัดที่ 10 เรียกใช้ฟังก์ชัน mysql_query() เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร \$sql แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result

บรรทัดที่ 11 เรียกใช้ฟังก์ชัน mysql_num_rows() เพื่อดึงข้อมูลจำนวนแถว แล้วนำผลลัพธ์ที่ได้ นั้นไปเก็บไว้ในตัวแปร \$num_rows

บรรทัดที่ 12 เรียกใช้ฟังก์ชัน mysql_num_fields() เพื่อดึงข้อมูลจำนวนฟิลด์หรือจำนวนคอลัมน์ แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$num_fields

บรรทัดที่ 14 ใช้แท็กเปิด <center> เพื่อกำหนดการแสดงผลต่อไปนี้อยู่ที่ตำแหน่งตรงกลางของหน้าจอ

บรรทัดที่ 15 ใช้แท็กเปิด <table> เพื่อสร้างตาราง โดยกำหนดค่าความกว้างของตารางเท่ากับ 500 พิกเซลมีขอบหนาเท่ากับ 1 พิกเซล แล้วตามด้วยแท็กเปิด<tr> ทิ้งไว้

บรรทัดที่ 17 กำหนดค่าสตริง show columns from register ให้กับตัวแปร \$sql ซึ่งเป็นคำสั่ง sql หมายถึงการแสดงชื่อฟิลด์หรือคอลัมน์ทั้งหมดที่อยู่ในตารางที่ชื่อว่า register

บรรทัดที่ 18 เรียกใช้ฟังก์ชัน mysql_query() เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร \$sql แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$col

บรรทัดที่ 19 ใช้ลูป `while` ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะมีการใช้ฟังก์ชัน `mysql_fetch_row()` เพื่อทำการดึงข้อมูลที่อยู่ในตัวแปร `$col` ออกมาเป็นอะเรย์แล้วนำไปเก็บไว้ในตัวแปร `$row`

บรรทัดที่ 20 ใช้แท็กเปิด `<th>` แล้วแสดงข้อมูลอะเรย์ที่อยู่ในตัวแปร `$row` โดยใช้อินเด็กซ์ที่เป็นตัวเลข ซึ่งในกรณีนี้ก็คือ 0 เท่านั้น เนื่องจากในแต่ละแถวมีข้อมูลเดียว ตามด้วยแท็กปิด `</th>` แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป

บรรทัดที่ 22 ใช้แท็กปิด `</tr>` เพื่อสิ้นสุดการแสดงผลข้อมูลในแถว

บรรทัดที่ 24 กำหนดค่า 0 (ศูนย์) ให้กับตัวแปร `$i`

บรรทัดที่ 25 ใช้ลูป `while` ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะตรวจสอบดูว่าค่าที่อยู่ในตัวแปร `$i` น้อยกว่าค่าที่อยู่ในตัวแปร `$num_rows` จริงหรือไม่ ถ้าจริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 26 - 32 แต่ถ้าไม่จริงก็จะข้ามไปทำคำสั่งในบรรทัดที่ 34 แทน

บรรทัดที่ 26 เรียกใช้ฟังก์ชัน `mysql_fetch_arrays()` เพื่อดึงข้อมูลจำนวนแถวออกมาเป็นอะเรย์ แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$data`

บรรทัดที่ 27 ใช้แท็กเปิด `<tr>` เพื่อเริ่มต้นการใส่ข้อมูลในแถวใหม่ โดยให้ข้อมูลนั้น ๆ อยู่ในตำแหน่งตรงกลางของคอลัมน์

บรรทัดที่ 28 - 30 ใช้ลูป `for` เพื่อวนรอบในการดึงข้อมูลแต่ละตัวออกมาแสดงในคอลัมน์ต่าง ๆ

บรรทัดที่ 31 ใช้แท็กปิด `</tr>` เพื่อสิ้นสุดการแสดงผลข้อมูลในแถว

บรรทัดที่ 32 เพิ่มค่าให้กับตัวแปร `$i` อีก 1

บรรทัดที่ 34 ใช้แท็กปิด `</table>` และ `</center>` เพื่อสิ้นสุดการสร้างตารางและการกำหนดตำแหน่งตรงกลาง

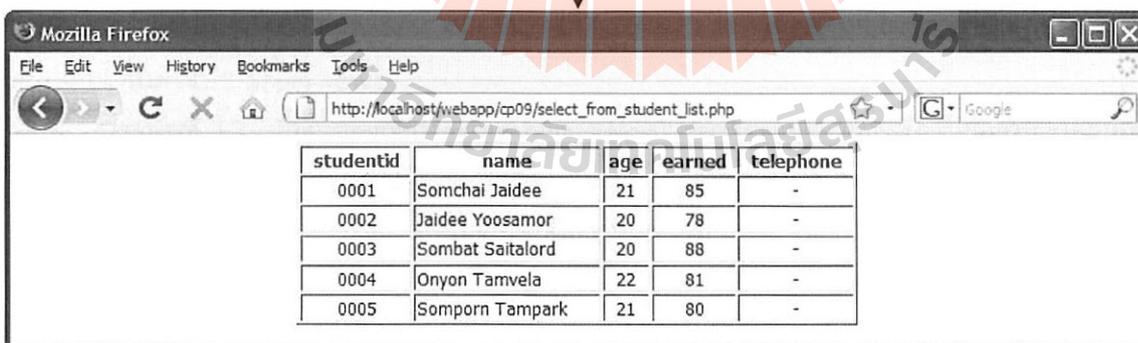
บรรทัดที่ 35 เรียกใช้ฟังก์ชัน `mysql_close()` เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร `$link` ใช้อยู่

ตัวอย่าง การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์และรายละเอียดต่าง ๆ ในตาราง student_list แล้วแสดงออกที่จอภาพในรูปแบบของตาราง

```

1. <?php
2.   include("connect.inc");
3.   $db = "webapp150";
4.   $tb = "student_list";
5.   $result = mysql_select_db($db);
6.   if(!$result){
7.     die('Could not find database called webapp150: '. mysql_error());
8.   }
9.
10.  $sql = "select * from $tb";
11.  $result = mysql_query($sql);
12.  $num_rows = mysql_num_rows($result);
13.  $num_fields = mysql_num_fields($result);
14.
15.  echo "<center>";
16.  echo "<table width=500px border=1px><tr>";
17.
18.  $sql = "show columns from student_list";
19.  $scol = mysql_query($sql);
20.  while($row = mysql_fetch_row($scol)) {
21.    echo "<th>$row[0]</th>";
22.  }
23.  echo "</tr>";
24.
25.  $i = 0;
26.  while($i<$num_rows){
27.    $data = mysql_fetch_array($result);
28.    echo "<tr align=\"center\">";
29.    for($j=0; $j<$num_fields; $j++){
30.      if($data[$j] == NULL)
31.        echo "<td>-</td>";
32.      else{
33.        if($j==1)
34.          echo "<td align=\"left\">$data[$j]</td>";
35.        else
36.          echo "<td>$data[$j]</td>";
37.      }
38.    }
39.    echo "</tr>";
40.    $i++;
41.  }
42.  echo "</table></center>";
43.  mysql_close($link);
44.  ?>

```



รูปที่ 9.10: การเรียกใช้ฐานข้อมูลชื่อ webapp150 พร้อมทั้งแสดงชื่อคอลัมน์และรายละเอียดต่าง ๆ ในตาราง student_list

- บรรทัดที่ 2 ใช้คำสั่ง include() เพื่อรวมแฟ้มข้อมูล connect.inc เข้ามา
- บรรทัดที่ 3 กำหนดค่าสตริง webapp150 ให้กับตัวแปร \$db
- บรรทัดที่ 4 กำหนดค่าสตริง student_list ให้กับตัวแปร \$tb
- บรรทัดที่ 5 เรียกใช้ฟังก์ชัน mysql_select_db() เพื่อเลือกใช้ฐานข้อมูลชื่อว่า webapp150 (ค่าที่อยู่ในตัวแปร \$db) แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result

บรรทัดที่ 6 ตรวจสอบผลที่อยู่ในตัวแปร `$result` ว่าทำงานได้ถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็จะเข้าไปทำคำสั่งในบรรทัดที่ 7 แต่ถ้าถูกต้องก็จะข้ามไปทำคำสั่งในบรรทัดที่ 10 แทน

บรรทัดที่ 7 เรียกใช้ฟังก์ชัน `die()` ซึ่งมีการทำงานเหมือนกันกับฟังก์ชัน `exit()` เพื่อแสดงข้อความว่า “Could not find database called webapp150:” แล้วตามด้วยการเรียกใช้ฟังก์ชัน `mysql_error()` เพื่อแสดงข้อผิดพลาดที่เกี่ยวข้องแล้วหยุดการทำงานของสคริปต์

บรรทัดที่ 10 กำหนดคำสั่งจริง `select * from $tb` ให้กับตัวแปร `$sql` ซึ่งเป็นคำสั่ง `sql` หมายถึงการเลือกข้อมูลทั้งหมดในตารางที่ชื่อว่า `student_list`

บรรทัดที่ 11 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$result`

บรรทัดที่ 12 เรียกใช้ฟังก์ชัน `mysql_num_rows()` เพื่อดึงข้อมูลจำนวนแถว แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$num_rows`

บรรทัดที่ 13 เรียกใช้ฟังก์ชัน `mysql_num_fields()` เพื่อดึงข้อมูลจำนวนฟิลด์หรือจำนวนคอลัมน์ แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$num_fields`

บรรทัดที่ 15 ใช้แท็กเปิด `<center>` เพื่อกำหนดการแสดงผลต่อไปนี้อยู่ที่ตำแหน่งตรงกลางของหน้าจอ

บรรทัดที่ 16 ใช้แท็กเปิด `<table>` เพื่อสร้างตาราง โดยกำหนดค่าความกว้างของตารางเท่ากับ 500 พิกเซลมีขอบหนาเท่ากับ 1 พิกเซล แล้วตามด้วยแท็กเปิด `<tr>` ทิ้งไว้

บรรทัดที่ 18 กำหนดคำสั่งจริง `show columns from student_list` ให้กับตัวแปร `$sql` ซึ่งเป็นคำสั่ง `sql` หมายถึงการแสดงชื่อฟิลด์หรือคอลัมน์ทั้งหมดที่อยู่ในตารางที่ชื่อว่า `student_list`

บรรทัดที่ 19 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$col`

บรรทัดที่ 20 ใช้ลูป `while` ในการวนรอบ ซึ่งในแต่ละรอบนั้น จะมีการใช้ฟังก์ชัน `mysql_fetch_row()` เพื่อทำการดึงข้อมูลที่อยู่ในตัวแปร `$col` ออกมาเป็นอะเรย์แล้วนำไปเก็บไว้ในตัวแปร `$row`

บรรทัดที่ 21 ใช้แท็กเปิด `<th>` แล้วแสดงข้อมูลอะเรย์ที่อยู่ในตัวแปร `$row` โดยใช้อินเด็กซ์ที่เป็นตัวเลข ซึ่งในกรณีนี้ก็คือ 0 เท่านั้น เนื่องจากในแต่ละแถวมีข้อมูลเดียว ตามด้วยแท็กปิด `</th>` แล้วเลื่อนตัวชี้ภายในไปยังตำแหน่งถัดไป

บรรทัดที่ 23 ใช้แท็กปิด `</tr>` เพื่อสิ้นสุดการแสดงข้อมูลในแถว

บรรทัดที่ 25 กำหนดค่า 0 (ศูนย์) ให้กับตัวแปร `$i`

บรรทัดที่ 26 ใช้ลูป `while` ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะตรวจสอบดูว่าค่าที่อยู่ในตัวแปร `$i` น้อยกว่าค่าที่อยู่ในตัวแปร `$num_rows` จริงหรือไม่ ถ้าจริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 27 - 40 แต่ถ้าไม่จริงก็จะข้ามไปทำคำสั่งในบรรทัดที่ 42 แทน

บรรทัดที่ 27 เรียกใช้ฟังก์ชัน `mysql_fetch_arrays()` เพื่อดึงข้อมูลจำนวนแถวออกมาเป็นอะเรย์ แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร `$data`

บรรทัดที่ 28 ใช้แท็กเปิด `<tr>` เพื่อเริ่มต้นการใส่ข้อมูลในแถวใหม่ โดยให้ข้อมูลนั้น ๆ อยู่ในตำแหน่งตรงกลางของคอลัมน์

บรรทัดที่ 29 – 38 ใช้ลูป for เพื่อวนรอบในการดึงข้อมูลแต่ละตัวออกมาแสดงในคอลัมน์ต่าง ๆ ซึ่งในบรรทัดที่ 30 นั้นเป็นการตรวจสอบว่าข้อมูลที่จะถูกแสดงออกนั้นมีค่าเท่ากับ NULL จริงหรือไม่ ถ้าจริงก็ให้แสดงตัวอักษร '-' แต่ถ้าไม่จริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 33 ซึ่งคำสั่งในบรรทัดที่ 33 นั้นเป็นการตรวจสอบว่าข้อมูลที่จะถูกแสดงออกนั้นมีตัวชี้หรืออินเด็กซ์เท่ากับ 1 จริงหรือไม่ ถ้าจริงก็จะแสดงข้อมูลนั้น (ชื่อ) ในตำแหน่งขีดซ้าย แต่ถ้าไม่จริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 36 ซึ่งจะแสดงข้อมูลนั้นในตำแหน่งกึ่งกลาง

บรรทัดที่ 39 ใช้แท็กปิด </tr> เพื่อสิ้นสุดการแสดงผลในแถว

บรรทัดที่ 40 เพิ่มค่าให้กับตัวแปร \$i อีก 1

บรรทัดที่ 42 ใช้แท็กปิด </table> และ </center> เพื่อสิ้นสุดการสร้างตารางและการกำหนดตำแหน่งตรงกลาง

บรรทัดที่ 43 เรียกใช้ฟังก์ชัน mysql_close() เพื่อปิดการเชื่อมต่อระบบฐานข้อมูลที่ตัวแปร \$link ซี่อยู่

จากรูปที่ 9.10 นั้น จะเห็นว่าข้อมูลทั้งหมด 5 เรคอร์ด (5 แถว) ซึ่งในแต่ละเรคอร์ดนั้นไม่มีข้อมูลในคอลัมน์ telephone เลย ดังนั้นเราสามารถปรับปรุงข้อมูลในคอลัมน์ telephone โดยเพิ่มคำสั่งที่อยู่ในกล่องสี่เหลี่ยม (บรรทัดที่ 9-14) ดังแสดงในรูปที่ 9.11

```

1. <?php
2. include("connect.inc");
3. $db = "webapp150";
4. $tb = "student_list";
5. $result = mysql_select_db($db);
6. if(!$result){
7.     die('Could not find database called webapp150: '.mysql_error());
8. }
9.
10. $stel = array("0898881234", "0898881111", "0898881122", "0899997777", "0898880000");
11. for($i=0; $i<5; $i++){
12.     $sid = "000".($i+1);
13.     $sql = "update student_list set telephone=\"$stel[$i]\" where studentid=$sid";
14.     $result = mysql_query($sql);
15. }
16.
17. $sql = "select * from $tb";
18. $result = mysql_query($sql);
19. $num_rows = mysql_num_rows($result);
20. $num_fields = mysql_num_fields($result);
21. echo "<center>";
22. echo "<table width=500px border=1px><tr>";
23. $sql = "show columns from student_list";
24. $col = mysql_query($sql);
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49. echo "</table></center>";
50. mysql_close($link);
51. ?>

```

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://localhost/webapp/cp09/update_student_list_telephone.php`. The main content area displays a table with the following data:

studentid	name	age	earned	telephone
0001	Somchai Jaidee	21	85	0898881234
0002	Jaidee Yoosamor	20	78	0898881111
0003	Sombat Saitalord	20	88	0898881122
0004	Onyon Tamvela	22	81	0899997777
0005	Somporn Tampark	21	80	0898880000

รูปที่ 9.11: การเรียกใช้ฐานข้อมูลชื่อ webapp150 เพื่อปรับปรุงข้อมูลในคอลัมน์ telephone ในตาราง student_list

คำสั่งในบรรทัดที่ 1-8 และ 17-51 เหมือนกับคำสั่งในรูปที่ 9.10 ดังนั้นจึงขออธิบายเฉพาะคำสั่งที่ถูกแทรกเข้ามา (บรรทัดที่ 10 -15) เท่านั้น

บรรทัดที่ 10 กำหนดค่าอะเรย์ที่ประกอบไปด้วยสตริง 0898881234, 0898881111, 0898881122, 0899997777 และ 0898880000 ให้กับตัวแปร \$tel

บรรทัดที่ 11 ใช้ลูป for เพื่อวนรอบเป็นจำนวน 5 รอบ

บรรทัดที่ 12 กำหนดค่าสตริง 000 และตามด้วยค่าที่อยู่ในตัวแปร \$i บวกด้วย 1 ให้กับตัวแปร \$id ซึ่งในรอบที่ 1 นั้น ตัวแปร \$id จะเก็บสตริง 0001 แล้วในรอบต่อ ๆ ไปก็จะเป็น 0002, 0003, 0004 และ 0005 ตามลำดับ

บรรทัดที่ 13 กำหนดค่าสตริง `update student_list set telephone = \"$tel[$i]\" where studentid=$id` ให้กับตัวแปร \$sql ซึ่งในแต่ละรอบนั้น ตัวแปร \$sql จะเก็บสตริง ดังนี้

รอบที่ 1 เก็บ `update student_list set telephone = "0898881234" where studentid="0001"`

รอบที่ 2 เก็บ `update student_list set telephone = "0898881111" where studentid="0002"`

รอบที่ 3 เก็บ `update student_list set telephone = "0898881122" where studentid="0003"`

รอบที่ 4 เก็บ `update student_list set telephone = "0899997777" where studentid="0004"`

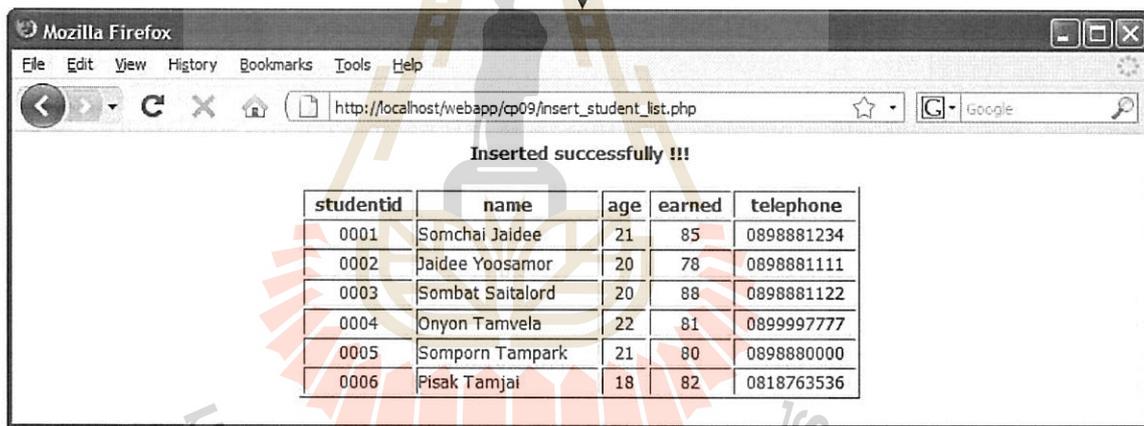
รอบที่ 5 เก็บ `update student_list set telephone = "0898880000" where studentid="0005"`

บรรทัดที่ 14 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร \$sql แล้วนำผลลัพธ์ที่ได้นั้นไปเก็บไว้ในตัวแปร \$result


```

1. <?php
2.   include("connect.inc");
3.   $db = "webapp150";
4.   $tb = "student_list";
5.   $result = mysql_select_db($db);
6.   if(!$result){
7.     die('Could not find database called webapp150: '. mysql_error());
8.   }
9.
10.  $sid = $_POST["id"];
11.  $name = $_POST["name"];
12.  $age = $_POST["age"];
13.  $earned = $_POST["earned"];
14.  $stel = $_POST["tel"];
15.
16.  $test_id = "select studentid from $tb where studentid=$id";
17.  $tmp = mysql_query($test_id);
18.  $rows = mysql_num_rows($tmp);
19.  if($rows!=0){
20.    echo "Invalid ID";
21.    exit;
22.  }else{
23.    $sql = "insert into $tb values (\\"$sid\\",\\"$name\\",\\"$age\\",\\"$earned\\",\\"$stel\\")";
24.    $result = mysql_query($sql);
25.    echo "<center><b>Inserted successfully !!!</b></center><br>";
26.  }
...
...
...
...
61.   echo "</table></center>";
62.   mysql_close($link);
63. >?

```



รูปที่ 9.13: การแสดงผลเมื่อทำการบันทึกข้อมูลลงในตาราง student_list ได้สำเร็จ

คำสั่งในบรรทัดที่ 1-8 และ 27-63 เหมือนกับคำสั่งในรูปที่ 9.10 ดังนั้นจึงขออธิบายเฉพาะคำสั่งที่ถูกแทรกเข้ามา (บรรทัดที่ 10 -26) เท่านั้น

- บรรทัดที่ 10 กำหนดค่าที่ส่งมาจากแบบฟอร์มที่มีชื่อว่า id ให้กับตัวแปร \$id
- บรรทัดที่ 11 กำหนดค่าที่ส่งมาจากแบบฟอร์มที่มีชื่อว่า name ให้กับตัวแปร \$name
- บรรทัดที่ 12 กำหนดค่าที่ส่งมาจากแบบฟอร์มที่มีชื่อว่า age ให้กับตัวแปร \$age
- บรรทัดที่ 13 กำหนดค่าที่ส่งมาจากแบบฟอร์มที่มีชื่อว่า earned ให้กับตัวแปร \$earned
- บรรทัดที่ 14 กำหนดค่าที่ส่งมาจากแบบฟอร์มที่มีชื่อว่า tel ให้กับตัวแปร \$tel
- บรรทัดที่ 16 กำหนดคำสั่งจริง select studentid from \$tb where studentid=\$id ให้กับตัวแปร \$test_id

บรรทัดที่ 17 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$test_id` แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ในตัวแปร `$tmp`

บรรทัดที่ 18 เรียกใช้ฟังก์ชัน `mysql_num_rows()` เพื่อดึงข้อมูลจำนวนแถว แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ในตัวแปร `$rows`

บรรทัดที่ 19 ตรวจสอบผลที่อยู่ในตัวแปร `$rows` ว่ามีค่าไม่เท่ากับ 0 จริงหรือไม่ ถ้าจริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 20-21 (ซึ่งก็หมายความว่าค่าที่อยู่ในตัวแปร `$id` ที่ผู้ใช้ป้อนเข้ามาในแบบฟอร์มนั้นมีค่าเท่ากับค่าที่อยู่ในคอลัมน์ `studentid` ในตาราง จึงทำให้การทดสอบนี้เป็นจริง) แต่ถ้าไม่จริงก็จะข้ามไปทำคำสั่งในบรรทัดที่ 23-25 แทน

บรรทัดที่ 20 แสดงข้อความ “Invalid ID” ออกที่จอภาพ

บรรทัดที่ 21 จบการทำงานของสคริปต์

บรรทัดที่ 23 กำหนดคำสั่งจริง `insert into $tb values (\“$id\”, \“$name\”, \“$age\”, \“$earned\”, \“$tel\”)` ให้กับตัวแปร `$sql`

บรรทัดที่ 24 เรียกใช้ฟังก์ชัน `mysql_query()` เพื่อส่งภาษาสอบถามที่อยู่ในตัวแปร `$sql` แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ในตัวแปร `$result`

บรรทัดที่ 25 แสดงข้อความ “Inserted successfully !!!” เป็นตัวหนาอยู่ตำแหน่งตรงกลางออกที่จอภาพ

ถ้าค่าที่อยู่ในตัวแปร `$id` ที่ผู้ใช้ป้อนเข้ามาในแบบฟอร์มนั้นมีค่าเท่ากับค่าที่อยู่ในคอลัมน์ `studentid` ในตาราง จะทำให้การทดสอบในบรรทัดที่ 19 นั้นเป็นจริง จะแสดงข้อความว่า “Invalid ID” แล้วจบโปรแกรม ดังแสดงในรูปที่ 9.14



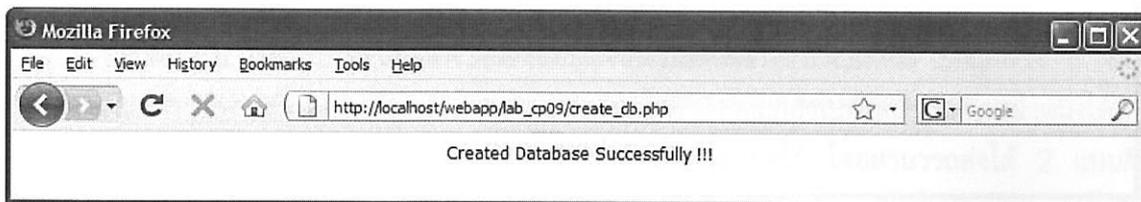
รูปที่ 9.14: แสดงข้อความ Invalid ID เมื่อผู้ใช้ป้อน id ซ้ำกับข้อมูลที่มีอยู่ในตาราง student_list

สรุป

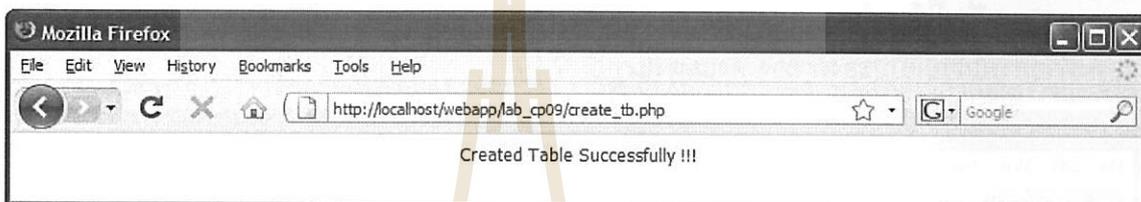
การติดต่อฐานข้อมูลและใช้คำสั่งต่าง ๆ ผ่านหน้าเว็บ โดยใช้สคริปต์ PHP เป็นเครื่องมือช่วยในการติดต่อพร้อมกับใช้ภาษาสอบถาม SQL และฟังก์ชันใน PHP ที่เกี่ยวกับการจัดการฐานข้อมูล ทำให้การติดต่อฐานข้อมูลในลักษณะนี้มีความสะดวกมากยิ่งขึ้นทั้งทางผู้ออกแบบระบบเองและผู้ใช้งาน เนื่องจากระบบจะทำการประมวลผลผ่านทางเว็บ ผู้ใช้งานระบบสามารถเข้าสู่ระบบฐานข้อมูลได้จากทุก ๆ แห่งที่มีการเชื่อมต่อระบบอินเทอร์เน็ต

แบบฝึกหัด

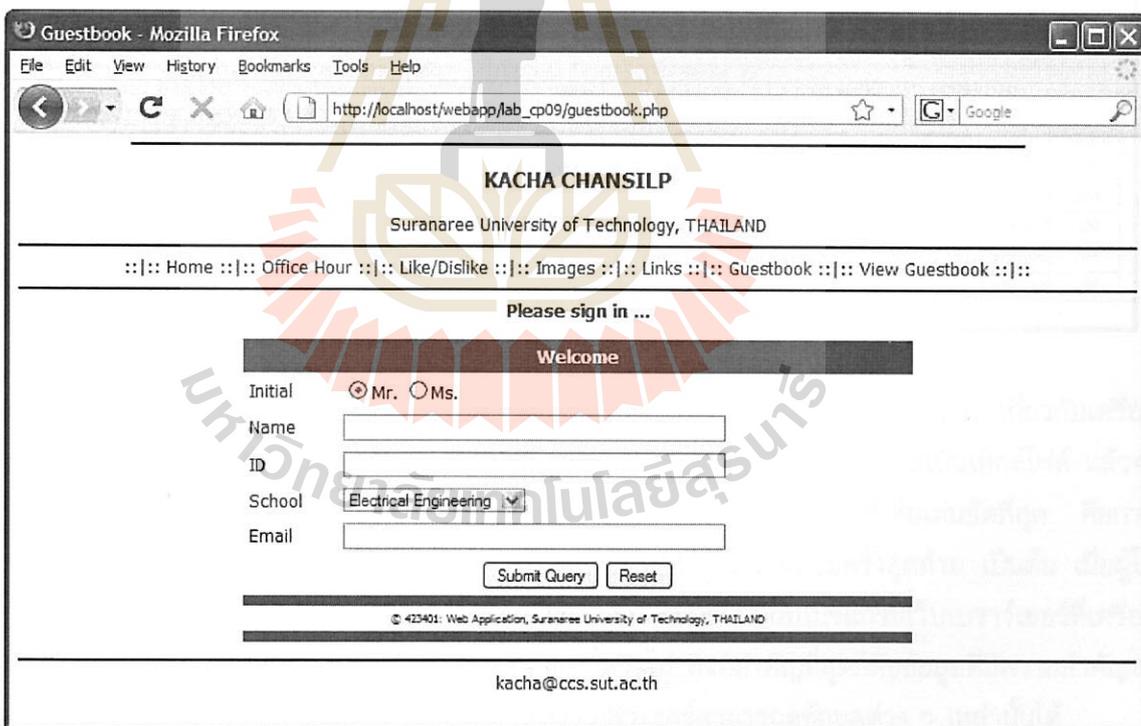
1. เขียนคำสั่ง PHP เพื่อสร้างฐานข้อมูลชื่อว่า guestbook ประกอบด้วยตาราง 1 ตาราง ชื่อว่า record ซึ่งในตาราง record นี้ก็จะประกอบด้วยฟิลด์หรือคอลัมน์ 5 คอลัมน์ เพื่อรองรับข้อมูลจากแบบฟอร์มดังแสดงในรูปที่ 9.17



รูปที่ 9.15: สร้างฐานข้อมูล guestbook สำเร็จ

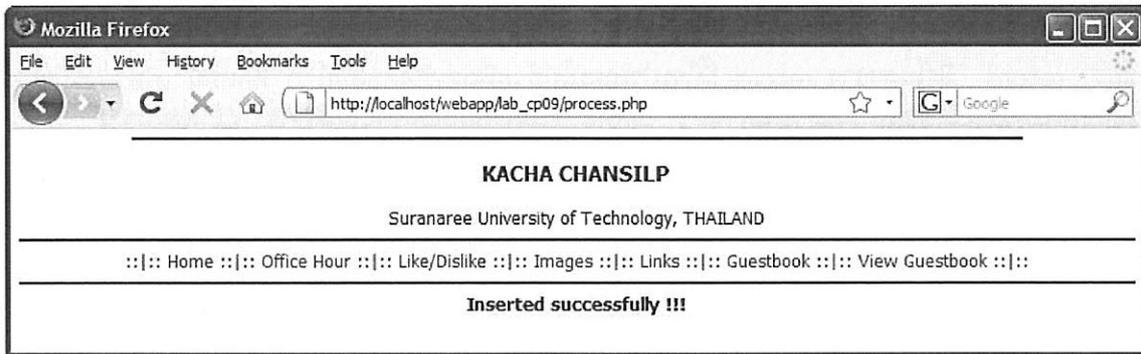


รูปที่ 9.16: สร้างตาราง record ในฐานข้อมูล guestbook สำเร็จ



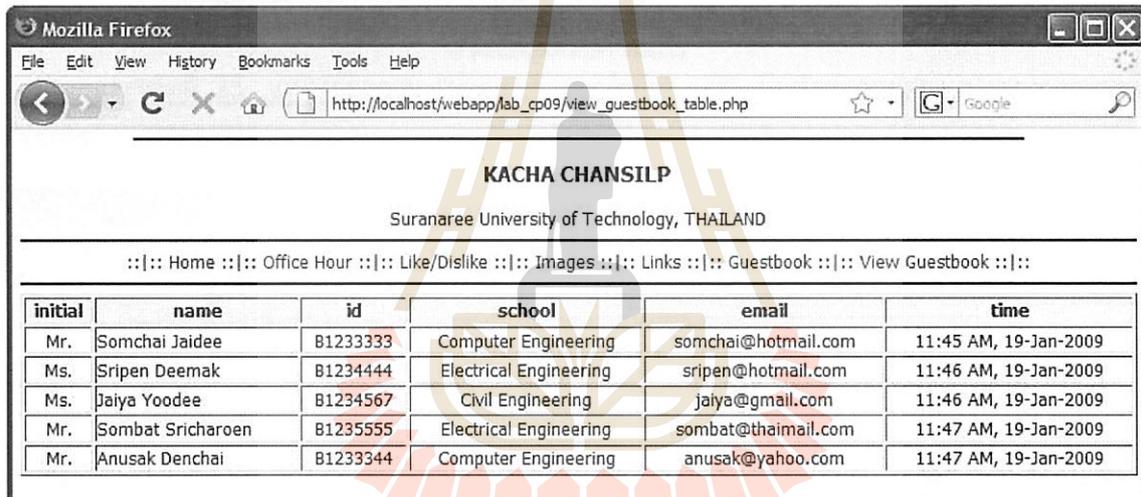
รูปที่ 9.17: แบบฟอร์มการรับข้อมูลเพื่อบันทึกลงในตาราง record ในฐานข้อมูล guestbook

2. จากแบบฟอร์มในรูปที่ 9.17 นั้น ให้ป้อนข้อมูลเข้าไปแล้วบันทึกลงในตาราง record ดังแสดงในรูปที่ 9.18 (ประมาณ 5 ข้อมูล)



รูปที่ 9.18: บันทึกข้อมูลลงในตาราง record ในฐานะข้อมูล guestbook สำเร็จ

3. เมื่อผู้ใช้กดปุ่ม View Guestbook ก็จะทำให้การดึงข้อมูลทั้งหมดจากตาราง record ออกมาแสดงที่จอภาพในรูปแบบของตาราง ดังแสดงในรูปที่ 9.19



รูปที่ 9.19: แสดงผลข้อมูลในตาราง record

บทที่ 10 คุกกี้และเซสชัน <Cookie and Session>

บทนำ

ที่ผ่านมาแล้ว เมื่อผู้เข้าชมเว็บไซต์ได้ป้อนข้อมูลจากแบบฟอร์มในหน้าเว็บหนึ่ง ข้อมูลต่าง ๆ จะถูกเก็บไว้ที่ตัวแปร ซึ่งค่าหรือข้อมูลที่ถูกเก็บไว้ที่ตัวแปรเหล่านั้นสามารถถูกส่งไปยังหน้าเว็บอีกหน้าหนึ่งได้เพื่อแสดงผลออกที่หน้าเว็บหรือบันทึกข้อมูลต่าง ๆ เหล่านี้ลงในฐานข้อมูลได้ โดยสามารถส่งได้ 2 แบบคือแบบ post และแบบ get เมื่อข้อมูลถึงหน้าเว็บที่ต้องการแล้ว ค่าหรือข้อมูลที่อยู่ในตัวแปรเหล่านั้นก็ไม่สามารถนำไปใช้ในหน้าเว็บอื่น ๆ ได้อีกต่อไป ทำให้เกิดปัญหาในบางกรณีได้ เช่น การตรวจสอบสถานะของผู้ใช้งาน ระบบจะรู้ได้อย่างไรว่าผู้ที่เข้ามาใช้งานนั้นเป็นใคร มีสิทธิ์ที่จะทำอะไรในระบบได้บ้าง เป็นต้น นี่จึงเป็นสาเหตุที่ต้องมีการจัดการกับข้อมูลบางอย่าง ที่จะต้องมีการตรวจสอบตลอดเวลาก่อนการแสดงผลของหน้าเว็บต่าง ๆ เหล่านี้ ข้อมูลที่จำเป็นเหล่านั้นสามารถถูกเก็บไว้ที่เครื่องไคลเอนท์หรือที่เครื่องเซิร์ฟเวอร์ก็ได้ตามความเหมาะสมที่ผู้ดูแลระบบต้องการโดยใช้คุกกี้ (Cookie) หรือเซสชัน (Session)

วัตถุประสงค์

1. เพื่อให้เข้าใจหลักการเก็บข้อมูลให้คงอยู่ในหน้าเว็บต่าง ๆ
2. เพื่อให้รู้ถึงวิธีการเก็บข้อมูลให้คงอยู่กับหน้าเว็บต่าง ๆ
3. เพื่อฝึกเขียนและทดสอบการเก็บข้อมูลและตรวจสอบข้อมูลในหน้าเว็บต่าง ๆ

เนื้อหา

10.1 คุกกี้คืออะไร

คุกกี้คือสิ่งที่เอื้ออำนวยให้ผู้ดูแลเว็บ (Webmaster) สามารถจัดเก็บข้อมูลต่าง ๆ เกี่ยวกับเครื่องของผู้เข้ามาเยี่ยมชมเว็บไซต์ไว้ที่เครื่องของผู้เข้าชมเอง โดยข้อมูลเหล่านั้นจะถูกจัดเก็บเป็นเท็กซ์ไฟล์ แล้วจะถูกเรียกขึ้นมาใช้งานอีกเมื่อผู้ใช้งานเข้ามาที่เว็บไซต์อีก ตัวอย่างของการใช้คุกกี้ที่เห็นเด่นชัดที่สุด คือการบันทึกชื่อเข้าสู่ระบบ (Username) รหัสผ่าน (Password) การเข้ามาเยี่ยมชมครั้งสุดท้าย เป็นต้น เมื่อผู้ใช้เข้าสู่เว็บไซต์ในครั้งต่อ ๆ ไปก็ไม่ต้องป้อนข้อมูลดังกล่าวอีก ทั้งนี้ขึ้นอยู่กับโปรแกรมเว็บเบราว์เซอร์ที่เครื่องของผู้ใช้งานเองว่าจะเปิดหรือปิดการเก็บข้อมูลแบบคุกกี้หรือไม่ สิ่งที่สำคัญที่สุดก็คือข้อมูลที่มีความสำคัญไม่ควรถูกเก็บไว้โดยคุกกี้ เนื่องจากคนอื่นที่มาใช้เครื่องคอมพิวเตอร์สามารถดูข้อมูลต่าง ๆ เหล่านี้ได้

ฟังก์ชันใน PHP ที่ใช้สำหรับการกำหนดการจัดเก็บข้อมูลแบบคุกกี้มีรูปแบบดังต่อไปนี้

```
bool setcookie (string name [,string value [,int expire [,string path [,string domain [,int secure]]]]);
```

โดยในสคริปต์นั้น ๆ จะต้องมีการเรียกใช้ฟังก์ชันนี้ในส่วนหัวของสคริปต์ก่อนมีการใช้คำสั่งใด ๆ ในการส่งผลต่าง ๆ ไปแสดงที่หน้าเว็บ มิฉะนั้นแล้วการกำหนดคุกกี้ก็จะไม่สามารถทำได้ แต่ใน PHP 4 สามารถส่งผลการแสดงก่อนเรียกใช้ฟังก์ชัน `setcookie()` ได้โดยส่งข้อมูลเหล่านั้นไปเก็บไว้ที่บัฟเฟอร์ภายใน (Internal Buffer) ก่อนโดยเรียกใช้ฟังก์ชัน `ob_start()` เพื่อเปิดการทำงานของบัฟเฟอร์แสดงผลมีรูปแบบดังต่อไปนี้

```
bool ob_start ([callback output_callback]);
```

ซึ่งเมื่อเปิดการทำงานของบัฟเฟอร์แสดงผลแล้ว การแสดงผลต่าง ๆ จะไม่ถูกนำไปแสดงยังหน้าเว็บแต่จะถูกเก็บไว้ที่บัฟเฟอร์ภายในแทน เมื่อต้องการแสดงข้อมูลที่อยู่ในบัฟเฟอร์ภายในออกที่หน้าเว็บก็สามารถทำได้โดยการเรียกใช้ฟังก์ชัน `ob_end_flush()` ซึ่งมีรูปแบบดังต่อไปนี้

```
bool ob_end_flush (void);
```

หรือถ้าต้องการเก็บข้อมูลที่อยู่ในบัฟเฟอร์ภายใน ก็สามารถทำได้โดยการดึงข้อมูลเหล่านั้นออกไปเป็นสตริงแล้วนำไปเก็บไว้ที่ตัวแปรได้ โดยการเรียกใช้ฟังก์ชัน `ob_get_contents()` ซึ่งมีรูปแบบดังต่อไปนี้

```
string ob_get_contents (void);
```

เมื่อต้องการลบข้อมูลที่อยู่ในบัฟเฟอร์ภายใน ก็สามารถทำได้โดยการเรียกใช้ฟังก์ชัน `ob_end_flush()` ซึ่งมีรูปแบบดังต่อไปนี้

```
bool ob_end_flush (void);
```

โดยทั่ว ๆ ไปแล้ว ถ้าต้องการเพียงแค่เปิดการทำงานของคุกกี้ก็เรียกใช้ฟังก์ชัน `setcookie()` ซึ่งรายละเอียดของพารามิเตอร์ที่เกี่ยวข้องได้แสดงในตารางที่ 10.1

ตารางที่ 10.1: พารามิเตอร์ต่าง ๆ ของฟังก์ชัน `setcookie()`

พารามิเตอร์	ความหมาย	ตัวอย่าง
name	ชื่อของตัวแปรคุกกี้	ชื่อ 'cookienam' ถูกเรียกได้โดยใช้คำสั่ง <code>\$COOKIE['cookienam']</code>
value	ค่าของตัวแปรคุกกี้จะถูกเก็บไว้ที่เครื่องไคลเอนท์ จึงไม่ควรเก็บข้อมูลที่สำคัญ	ค่าของตัวแปรคุกกี้ที่ชื่อ 'cookienam' สามารถดึงมาได้โดยใช้คำสั่ง <code>\$_COOKIE['cookienam']</code>
expire	เวลาหมดอายุของตัวแปรคุกกี้ มีหน่วยเป็นวินาที ซึ่งสามารถทำได้โดยการเรียกใช้ฟังก์ชัน <code>time()</code> บวกกับจำนวนวินาทีที่ต้องการให้หมดอายุ	<code>time()+60*60*24*30</code> คือการตั้งค่าหมดอายุ 30 วัน ถ้าไม่ตั้งค่าใด ๆ ตัวแปรคุกกี้จะหมดอายุเมื่อโปรแกรมเว็บเบราว์เซอร์ถูกปิดลง
path	เส้นทางในเครื่องเซิร์ฟเวอร์ที่สามารถเข้าถึงตัวแปรคุกกี้ได้	เมื่อตั้งค่าเป็น '/' ตัวแปรคุกกี้สามารถใช้งานได้ทั้งหมดในโดเมน แต่ถ้าตั้งค่าเป็น '/abc/' เฉพาะไดเรกทอรี abc และไดเรกทอรีที่อยู่ภายในทั้งหมดของ abc เท่านั้นที่สามารถใช้งานได้
domain	โดเมนที่สามารถใช้งานตัวแปรคุกกี้ได้	กำหนดตัวแปรคุกกี้ให้สามารถถูกเรียกใช้งานได้โดยโดเมนภายในทั้งหมดที่อยู่ในโดเมน abc.com ทำได้โดยกำหนด

พารามิเตอร์	ความหมาย	ตัวอย่าง
		'abc.com' ซึ่งเครื่องหมาย '.' ก่อน abc นั้นไม่จำเป็นต้องใส่ก็ได้ แต่ที่ใส่ไว้ก็เพราะต้องการให้ใช้งานได้โปรแกรมเว็บเบราว์เซอร์ต่าง ๆ ได้มากยิ่งขึ้น
secure	ตัวกำหนดว่าตัวแปรคุกกี้ควรจะถูกส่งได้เฉพาะการเชื่อมต่อแบบ https เท่านั้น (0 หรือ 1)	กำหนดค่าเป็น 1 เมื่อต้องการส่งตัวแปรคุกกี้เฉพาะการเชื่อมต่อแบบ https เท่านั้น ค่าปริยายคือ 0

ตัวอย่าง การตั้งค่าและบันทึกข้อมูลวันและเวลาที่เครื่องผู้ใช้งาน

```

1. <?php
2.     $month = time()+2592000;
3.     $time_now = date("F jS, Y - g:i a");
4.     setcookie(LastVisit, $time_now, $month);
5.     echo "Hi.....<br>";
6.     echo "Today is $time";
7. ?>
    
```

รูปที่ 10.1: การตั้งค่าและบันทึกข้อมูลแบบคุกกี้

บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP

บรรทัดที่ 2 เป็นการกำหนดค่าให้กับตัวแปร \$month เก็บจำนวนวินาทีในเวลา 1 เดือน นั่นคือ วัน x ชั่วโมง x นาที x วินาที ซึ่งก็คือ $30 \times 24 \times 60 \times 60 = 2592000$ วินาที

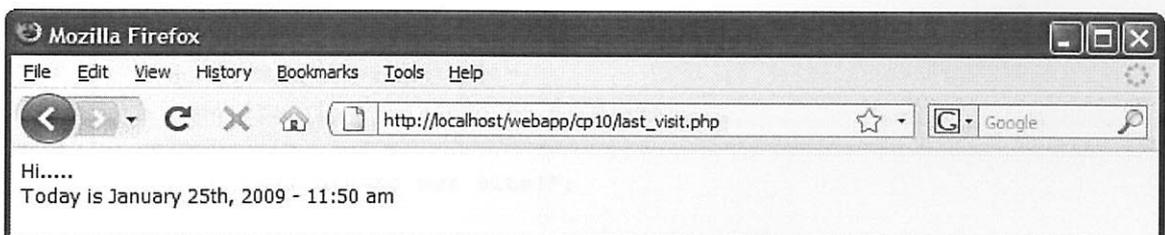
บรรทัดที่ 3 เป็นการกำหนดค่าให้กับตัวแปร \$time_now เก็บค่าที่ได้จากการเรียกใช้ฟังก์ชัน date()

บรรทัดที่ 4 เป็นการตั้งชื่อตัวแปรแบบคุกกี้ใหม่ให้มีชื่อว่า LastVisit เก็บค่าที่ได้จากตัวแปร \$time_now โดยเก็บไว้เป็นเวลา 30 วัน

บรรทัดที่ 5 แสดงข้อความ "Hi....." ออกที่หน้าเว็บ

บรรทัดที่ 6 แสดงข้อความ "Today is" พร้อมกับค่าที่ถูกเก็บไว้ในตัวแปร \$time ออกที่หน้าเว็บ ดังแสดงในรูปที่ 10.2

บรรทัดที่ 7 จบการทำงาน



รูปที่ 10.2: การแสดงผลของการใช้คำสั่งจากรูปที่ 10.1

หลังจากที่มีการบันทึกข้อมูลลงที่เครื่องผู้ใช้งานแล้ว คำสั่งที่จะใช้ในการตรวจสอบค่าเหล่านั้นมีดังแสดงในรูปที่ 10.3

```

1.  <?php
2.     $time_now = date("F jS, Y - g:i a");
3.     echo "Hi.....<br>";
4.     echo "Today is $time_now";
5.     if(isset($_COOKIE[ 'LastVisit' ])){
6.         $last = $_COOKIE[ 'LastVisit' ];
7.         echo "Welcome back!<br>";
8.         echo "You last visited on ". $last;
9.     }else{
10.        echo "Welcome to our website!";
11.    }
12. ?>

```

รูปที่ 10.3: การตรวจสอบค่าและข้อมูลแบบคุกกี้

บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP

บรรทัดที่ 2 เป็นการกำหนดค่าให้กับตัวแปร \$time_now เก็บค่าที่ได้จากการเรียกใช้ฟังก์ชัน date()

บรรทัดที่ 3 แสดงข้อความ “Hi.....” ออกที่หน้าเว็บ

บรรทัดที่ 4 แสดงข้อความ “Today is” พร้อมกับค่าที่ถูกเก็บไว้ในตัวแปร \$time_now ออกที่หน้าเว็บ

บรรทัดที่ 5 เป็นการตรวจสอบว่ามีค่าของตัวแปรแบบคุกกี้ที่มีชื่อว่า LastVisit หรือไม่ โดยใช้ฟังก์ชัน isset() ถ้ามีก็จะเข้าไปทำในบรรทัดที่ 6 ถึง 8 มิฉะนั้นแล้วจะไปทำคำสั่งในบรรทัดที่ 10 แทน

บรรทัดที่ 6 เป็นการกำหนดค่าให้กับตัวแปร \$last โดยให้มีค่าเท่ากับข้อมูลที่ถูเก็บไว้ในตัวแปรแบบคุกกี้ที่ชื่อว่า LastVisit

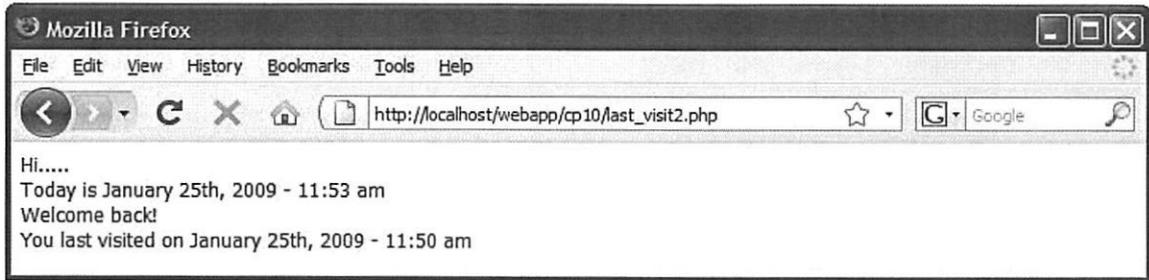
บรรทัดที่ 7 แสดงข้อความ “Welcome back!” ออกที่หน้าเว็บ

บรรทัดที่ 8 แสดงข้อความ “You last visited on” พร้อมกับค่าที่ถูกเก็บไว้ในตัวแปร \$last ออกที่หน้าเว็บ ดังแสดงในรูปที่ 10.4

บรรทัดที่ 10 ถ้าการตรวจสอบในบรรทัดที่ 5 เป็นเท็จ ก็จะเข้ามาทำคำสั่งในบรรทัดนี้ คือแสดงข้อความ “Welcome to our website!” ออกที่หน้าเว็บแทน

บรรทัดที่ 11 จบการทำงานของ else

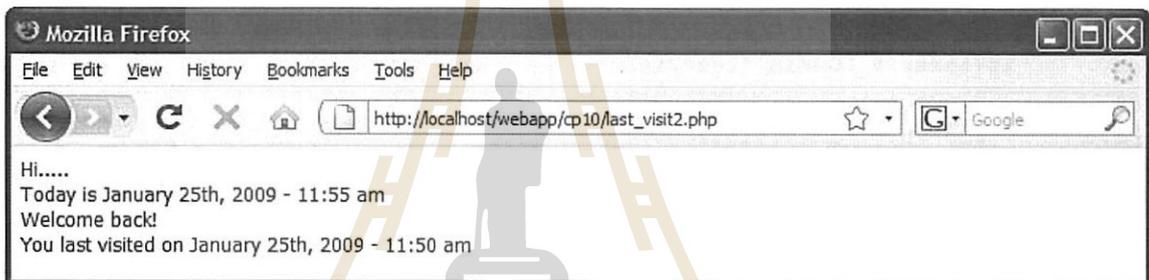
บรรทัดที่ 12 จบการทำงานของไฟล์ last_visit.php



รูปที่ 10.4: การแสดงผลของการใช้คำสั่งจากรูปที่ 10.3

จากรูปที่ 10.4 แสดงให้เห็นถึงเวลาเมื่อเรียกไฟล์ last_visit2.php นั่นคือเวลา 11.53 am และเวลาครั้งสุดท้ายที่เข้าชมเว็บไซต์นี้ (เวลา 11.50 am) ซึ่งก็คือค่าที่เรียกมาจากตัวแปร \$LastVisit นั่นเอง

ตอนนี้ทดลองเรียกไฟล์ last_visit2.php อีกครั้ง จะสังเกตเห็นว่าเวลาที่เรียกไฟล์ last_visit2.php นั่นคือเวลา 11.55 am และเวลาครั้งสุดท้ายที่เข้าชมเว็บไซต์นี้ควรจะเป็นเวลา 11.53 am แต่ยังคงเป็นเวลาที่เดิมนั่นคือเวลา 11.50 am ซึ่งค่าของตัวแปร \$LastVisit ไม่ถูกเปลี่ยนแปลงเลย นั่นคือจะเก็บค่าตั้งแต่เข้ามาเยี่ยมชมเว็บในครั้งแรกเท่านั้น ดังแสดงในรูปที่ 10.5



รูปที่ 10.5: การแสดงผลของการเรียกไฟล์ last_visit2.php อีกครั้ง

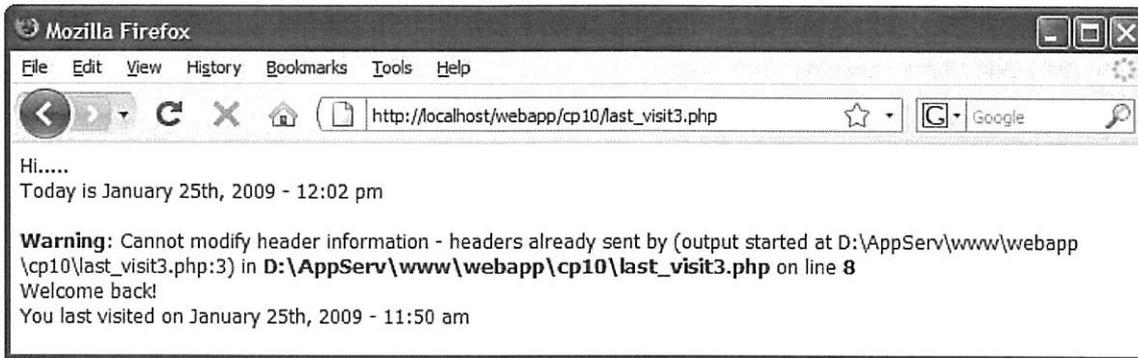
ดังนั้นจึงมีความจำเป็นที่จะต้องปรับปรุงข้อมูลของวันและเวลาที่เข้าชมเว็บไซต์ครั้งล่าสุดด้วยการเพิ่มคำสั่งดังแสดงในรูปที่ 10.6

```

1.  <?php
2.      $time_now = date("F jS, Y - g:i a");
3.      echo "<br>Hi.....<br>";
4.      echo "Today is $time_now<br>";
5.      if(isset($_COOKIE['LastVisit'])){
6.          $last = $_COOKIE['LastVisit'];
7.          $month = time()+2592000;
8.          setcookie('LastVisit', $time_now, $month);
9.          echo "Welcome back!<br>";
10.         echo "You last visited on ". $last;
11.     } else{
12.         echo "Welcome to our site!";
13.     }
14.  ?>

```

รูปที่ 10.6: การเพิ่มคำสั่งในการปรับปรุงข้อมูลของวันและเวลาครั้งสุดท้ายที่เข้าเว็บ



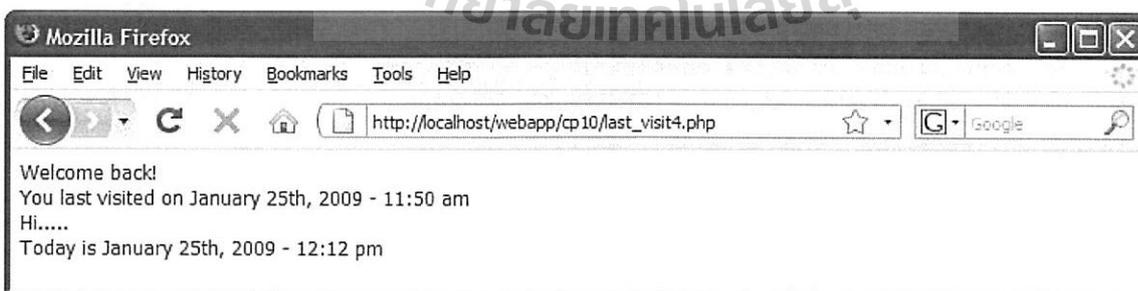
รูปที่ 10.7: การแสดงผลของการใช้คำสั่งจากรูปที่ 10.6

จากรูปที่ 10.7 แสดงค่าเตือนของการใช้คุกกี้ในบรรทัดที่ 8 ซึ่งไม่สามารถปรับเปลี่ยนข้อมูลในคุกกี้ได้ เนื่องจากส่วนหัวของไฟล์นี้ได้ถูกส่งไปที่เครื่องเซิร์ฟเวอร์แล้ว ดังนั้นจึงต้องใช้คำสั่งในบรรทัดที่ 8 ก่อนที่จะมีการส่งส่วนใด ๆ ไปที่เครื่องเซิร์ฟเวอร์ โดยปรับคำสั่งต่าง ๆ จากรูปที่ 10.6 ให้เป็นดังแสดงในรูปที่ 10.8

```

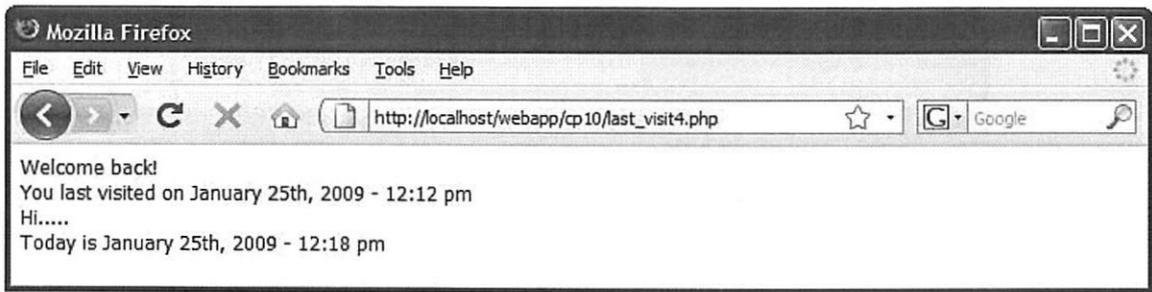
1.  <?php
2.      $time_now = date("F jS, Y - g:i a");
3.      if(isset($_COOKIE['LastVisit'])){
4.          $last = $_COOKIE['LastVisit'];
5.          $month = time()+2592000;
6.          setcookie('LastVisit', $time_now, $month);
7.          echo "Welcome back!<br>";
8.          echo "You last visited on ". $last;
9.      } else{
10.         echo "Welcome to our site!";
11.     }
12.     echo "<br>Hi.....<br>";
13.     echo "Today is $time_now<br>";
14.  ?>
    
```

รูปที่ 10.8: การใช้คำสั่งในการปรับปรุงข้อมูลของวันและเวลาครั้งล่าสุดที่เข้าเว็บให้อยู่ในส่วนหัว



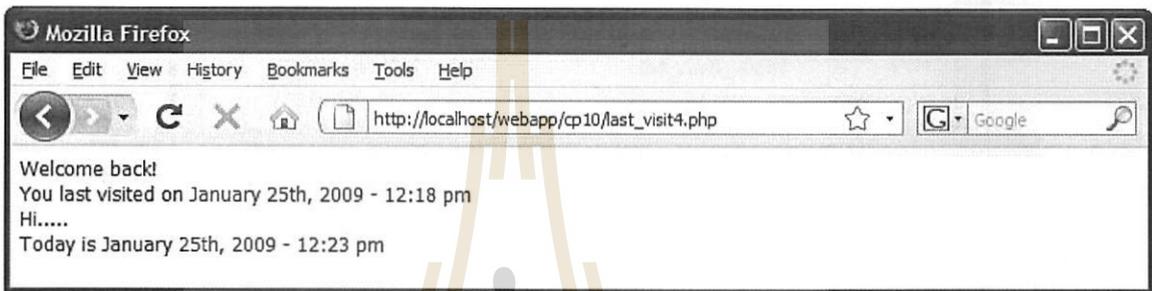
รูปที่ 10.9: การแสดงผลของการใช้ไฟล์ last_visit4.php ครั้งที่ 1

ตอนนี้ทดลองเรียกไฟล์ last_visit4.php อีกครั้ง จะสังเกตเห็นว่าเวลาที่เรียกไฟล์ last_visit4.php ในครั้งนี้คือเวลา 12.12 pm และเวลาครั้งสุดท้ายที่เข้าชมเว็บไซต์นี้ควรจะเป็นเวลา 11.50 am เวลาที่ถูกเก็บไว้ที่ตัวแปร \$LastVisit จะถูกเปลี่ยนไปจากเวลาเดิมคือ 11.50 am ไปเป็นเวลา 12.12 pm สามารถทดสอบได้โดยการเรียกไฟล์ last_visit4.php อีกครั้ง ดังแสดงในรูปที่ 10.10



รูปที่ 10.10: การแสดงผลของการใช้ไฟล์ last_visit4.php ครั้งที่ 2

ถ้าเรียกไฟล์ last_visit4.php อีกครั้ง เวลาครั้งสุดท้ายที่เข้าชมเว็บไซต์นี้จะเป็นเวลา 12.18 pm ดังแสดงในรูปที่ 10.11



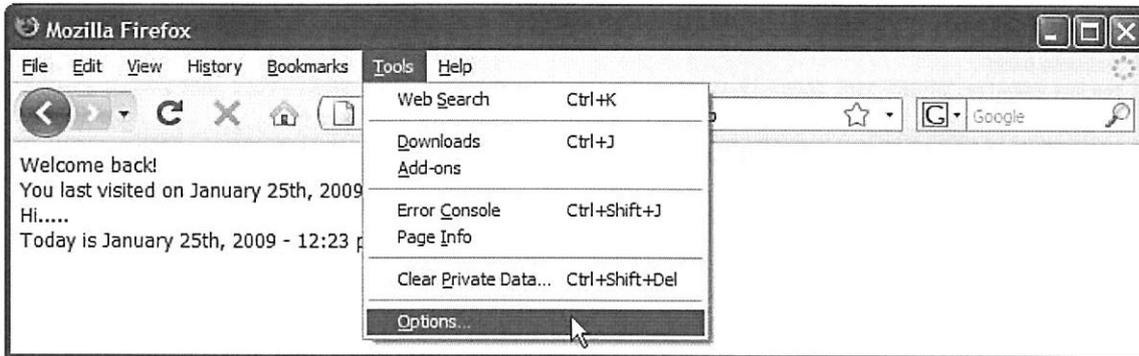
รูปที่ 10.11: การแสดงผลของการใช้ไฟล์ last_visit4.php ครั้งที่ 3

หมายเหตุ ตัวแปรคุกกี้จะหมดอายุหรือใช้งานไม่ได้เมื่อถึงเวลาที่กำหนดโดยผู้พัฒนาระบบ ส่วนไฟล์ที่จะเรียกใช้ตัวแปรคุกกี้จะต้องอยู่ในโฟลเดอร์เดียวกันกับไฟล์ที่สร้างตัวแปรคุกกี้

10.2 การลบคุกกี้

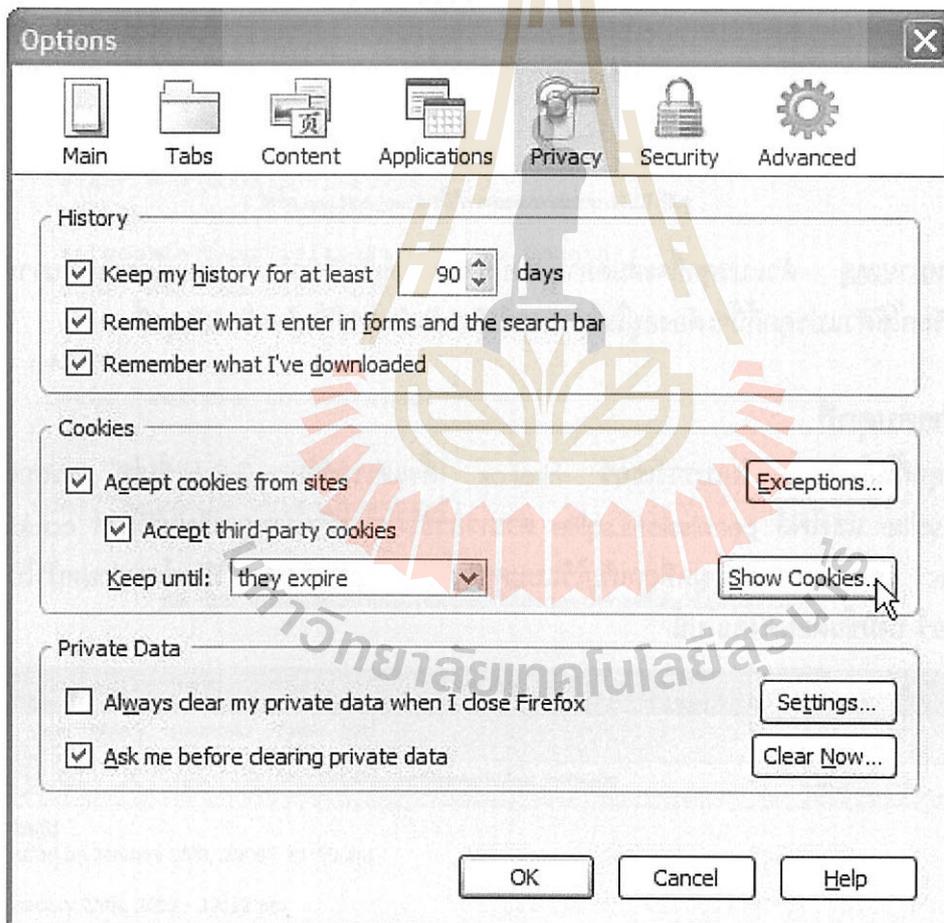
คุกกี้ในโปรแกรมเว็บเบราว์เซอร์ Firefox ตั้งแต่เวอร์ชัน 3.0 ขึ้นไป จะถูกเก็บไว้ที่ไฟล์ cookies.sqlite และไฟล์ permissions.sqlite ส่วนเวอร์ชันก่อนหน้านั้นจะถูกเก็บไว้ที่ไฟล์ cookies.txt และ hostperm.1 แต่วิธีการลบข้อมูลที่ถูเก็บไว้แบบคุกกี้สามารถทำได้ 2 วิธี โดยผ่านหน้าโปรแกรมเว็บเบราว์เซอร์ ซึ่งมีขั้นตอนดังต่อไปนี้

วิธีที่ 1 เลือกเมนู Tools แล้วเลือก Options... ดังแสดงในรูปที่ 10.12



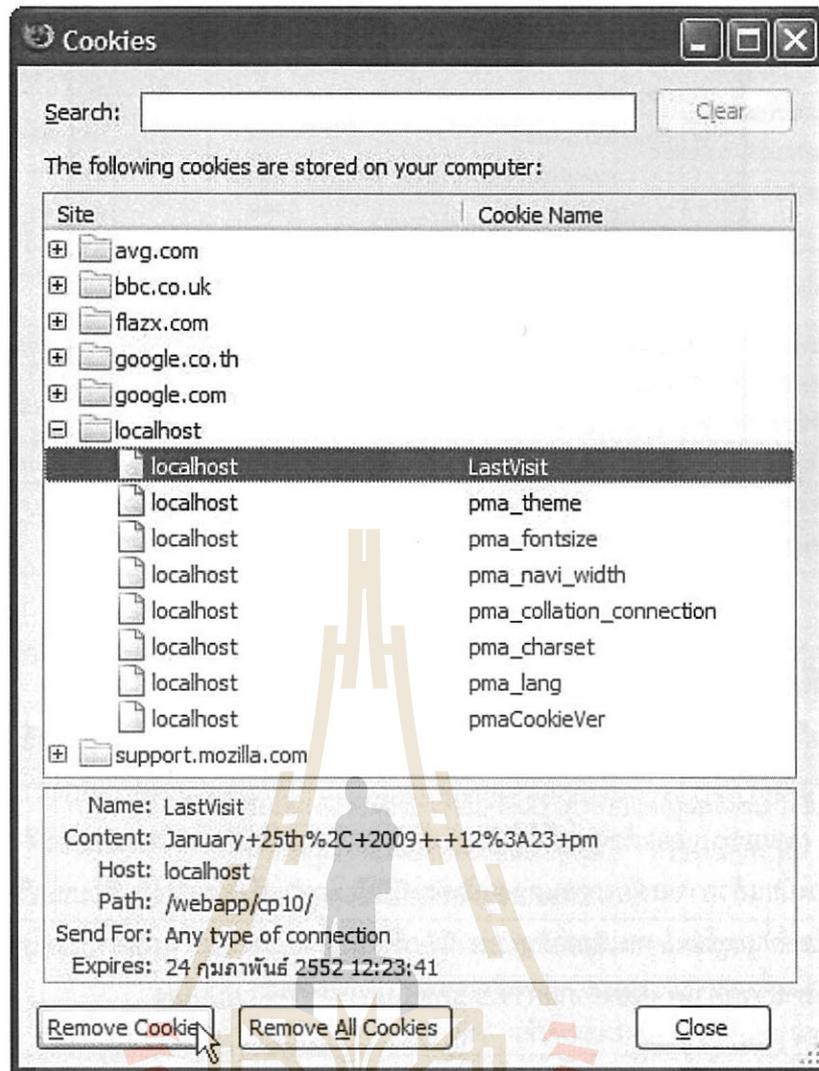
รูปที่ 10.12: การเลือกเมนู Tools แล้วตามด้วย Options

เลือก Privacy แล้วกดปุ่ม Show Cookies... ดังแสดงในรูปที่ 10.13



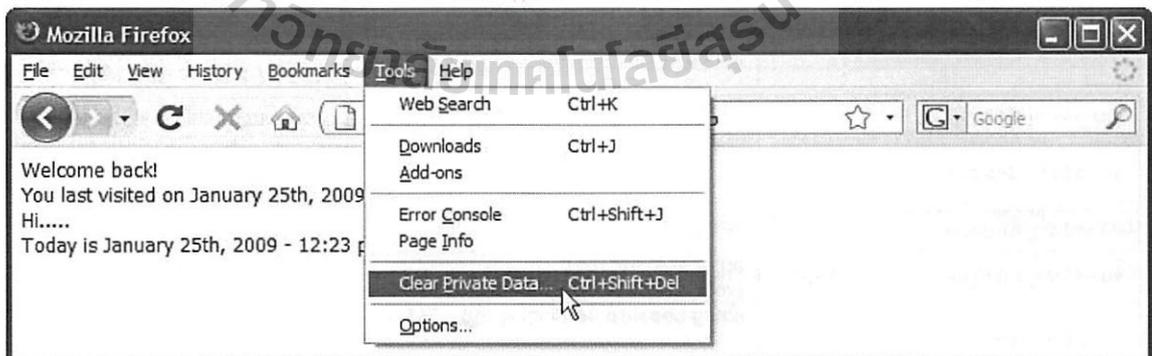
รูปที่ 10.13: การเลือกเมนู Privacy แล้วกดปุ่ม Show Cookies...

เมื่อกดปุ่ม Show Cookies... แล้วหน้าต่างดังรูปที่ 10.14 จะแสดงให้เห็นถึงคุกกี้ต่าง ๆ ที่มีการจัดเก็บไว้ที่เครื่องผู้ใช้งานเองหรือเรียกว่าเครื่องโคลเออนท์ ให้เลือกคุกกี้ที่ต้องการลบทิ้งแล้วกดปุ่ม Remove Cookie ดังแสดงในรูปที่ 10.14



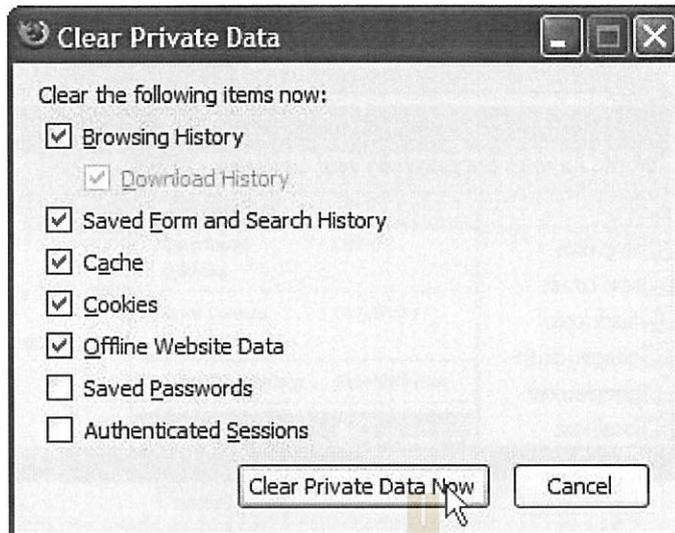
รูปที่ 10.14: การเลือกคุกกี้ที่ต้องการลบทิ้ง

วิธีที่ 2 เลือกเมนู Tools แล้วเลือก Clear Private Data... ดังแสดงในรูปที่ 10.15



รูปที่ 10.15: การเลือกเมนู Tools แล้วตามด้วย Clear Private Data...

จะปรากฏหน้าต่าง Clear Private Data ดังแสดงในรูปที่ 10.16 แล้วให้เลือกหัวข้อต่าง ๆ ที่ต้องการลบทิ้ง



รูปที่ 10.16: การเลือกหัวข้อต่าง ๆ ที่ต้องการลบในหน้าต่าง Clear Private Data

10.3 เซสชันคืออะไร

เซสชันก็มีลักษณะการทำงานที่คล้ายกันกับคุกกี้ เพียงแต่ข้อมูลจะถูกเก็บที่เครื่องเซิร์ฟเวอร์แทนซึ่งมีประโยชน์มากในการตรวจสอบข้อมูลและสถานะของผู้เข้ามาเยี่ยมชมเว็บไซต์ เนื่องจากหน้าต่างเว็บไซต์บางส่วนนั้น ผู้ดูแลระบบอาจต้องการให้ผู้เข้ามาเยี่ยมชมเว็บไซต์ได้ลงทะเบียนเสียก่อนจึงจะสามารถเข้าใช้งานระบบได้ มิฉะนั้นแล้วอาจจะต้องการแสดงหน้าต่างที่มีข้อความแจ้งให้ผู้เข้ามาเยี่ยมชมเว็บไซต์ได้รู้ว่าผู้เข้ามาเยี่ยมชมเว็บไซต์นั้นจะต้องลงทะเบียนเสียก่อน ถึงแม้ว่าผู้เข้ามาเยี่ยมชมเว็บไซต์จะรู้ชื่อไฟล์หรือ URL ที่จะเข้าโดยตรงก็ไม่สามารถทำได้ เนื่องจากมีการใช้เซสชันในการตรวจสอบนั่นเอง

ฟังก์ชันต่าง ๆ ที่เกี่ยวกับเซสชันใน PHP มีรายละเอียดดังแสดงในตารางที่ 10.2

ตารางที่ 10.2: ฟังก์ชันเกี่ยวกับเซสชัน

ฟังก์ชัน	รูปแบบ	ความหมาย
<code>session_cache_expire()</code>	<code>int session_cache_expire ([int new_cache_expire])</code>	ส่งกลับค่าหมดเวลาของแคชปัจจุบัน (มีหน่วยเป็นนาทีและค่าปริยายคือ 180)
<code>session_cache_limiter()</code>	<code>string session_cache_limiter ([string cache_limiter])</code>	ดึง และ/หรือ กำหนดค่าตัวควบคุมแคชปัจจุบัน (current cache limiter)
<code>session_decode()</code>	<code>bool session_decode (string data)</code>	ถอดรหัสข้อมูลเซสชันปัจจุบันจากสตริง
<code>session_destroy()</code>	<code>bool session_destroy (void)</code>	ทำลายข้อมูลทั้งหมดที่ได้ลงทะเบียนกับเซสชันแต่ไม่ได้ยกเลิกเซสชัน
<code>session_encode()</code>	<code>string session_encode (void)</code>	เข้ารหัสข้อมูลเซสชันปัจจุบันเป็นสตริง
<code>session_get_cookie_params()</code>	<code>array session_get_cookie_params (void)</code>	ดึงเซสชันคุกกี้พารามิเตอร์
<code>session_id()</code>	<code>string session_id ([string id])</code>	ดึง และ/หรือ กำหนดค่าไอดีเซสชันปัจจุบัน
<code>session_is_registered()</code>	<code>bool session_is_registered (string name)</code>	ตรวจสอบการลงทะเบียนของตัวแปรแบบโกลบอลในเซสชัน
<code>session_module_name()</code>	<code>string session_module_name ((string module))</code>	ดึง และ/หรือ กำหนดค่าโมดูลเซสชันปัจจุบัน
<code>session_name()</code>	<code>string session_name ((string name))</code>	ดึง และ/หรือ กำหนดค่าชื่อเซสชันปัจจุบัน

ฟังก์ชัน	รูปแบบ	ความหมาย
session_regenerate_id()	bool session_regenerate_id (void)	ปรับปรุงไอดีเซสชันปัจจุบันด้วย
session_register()	bool session_register (mixed name [, mixed ...])	ลงทะเบียนตัวแปรแบบโกลบอลหนึ่งตัว หรือมากกว่ากับเซสชันปัจจุบัน
session_save_path()	string session_save_path ([string path])	ตั้ง และ/หรือ กำหนดค่าของเส้นทางที่บันทึกเซสชันปัจจุบัน
session_set_cookie_params()	void session_set_cookie_params (int lifetime [, string path [, string domain [, bool secure]])	กำหนดค่าเซสชันคุกกี้พารามิเตอร์
session_start()	bool session_start (void)	กำหนดการเปิดใช้เซสชัน
session_unregister()	bool session_unregister (string name)	ยกเลิกการลงทะเบียนของตัวแปรแบบโกลบอลจากเซสชันปัจจุบัน
session_unset()	void session_unset (void)	ยกเลิกตัวแปรเซสชันทั้งหมด
session_write_close()	void session_write_close (void)	เขียนข้อมูลเซสชันแล้วจบเซสชัน
unset()	void unset (mixed var [, mixed var [, mixed ...]])	ยกเลิกตัวแปรเซสชันที่ระบุ

สมมุติว่ามีไฟล์สกุล .php อยู่ 10 ไฟล์ ซึ่งแต่ละไฟล์จะมีหน้าที่ต่าง ๆ ดังแสดงในตารางที่ 10.3

ตารางที่ 10.3: ไฟล์ข้อมูลตรวจสอบการทำงานแบบเซสชัน

ไฟล์	หน้าที่
index.php	จะแสดงแบบฟอร์มการล็อกอินเข้าสู่ระบบและปุ่มสมัครเข้าเป็นสมาชิกใหม่ ในกรณีที่ผู้ใช้งานยังไม่ได้ล็อกอินเข้าสู่ระบบ หรือจะแสดงปุ่ม “ออกจากระบบ” เมื่อผู้ใช้งานได้ล็อกอินเข้าสู่ระบบแล้ว
login.php	ตรวจสอบชื่อและรหัสผ่านกับฐานข้อมูล ถ้าข้อมูลถูกต้องก็จะแสดงข้อความ “ยินดีต้อนรับ” แล้วจะแสดงหน้า home.php มิฉะนั้นแล้วจะแสดงข้อความ “ชื่อหรือรหัสผ่านไม่ถูกต้อง” แล้วจะแสดงหน้า index.php แทน
header.php	แสดงจุดเชื่อมต่อไปยังเว็บหน้าอื่น
home.php	แสดงหน้าหลักและรายละเอียดต่าง ๆ
products.php	แสดงหน้าเกี่ยวกับสินค้าซอฟต์แวร์
about.php	แสดงหน้าเกี่ยวกับเว็บไซต์
logout.php	จะแสดงข้อความ “โอกาสหน้า เชิญใหม่นะคะ” แล้วจะแสดงหน้า index.php
signin_form.php	จะแสดงแบบฟอร์มเพื่อถามชื่อ รหัสผ่าน และอีเมล
signin.php	จะตรวจสอบข้อมูลที่ผู้ใช้ป้อนเข้ามาว่าครบทุกรายการหรือไม่ ถ้าครบก็จะทำการบันทึกข้อมูลเหล่านั้นลงฐานข้อมูล มิฉะนั้นแล้วจะแสดงข้อความ “กรอกข้อมูลไม่ครบถ้วน” แล้วจะแสดงหน้า index.php
error.php	จะแสดงข้อความ “คุณยังไม่ได้ลงทะเบียนเข้าสู่ระบบ” ในกรณีที่ผู้ใช้พยายามเข้าหน้าเว็บต่าง ๆ (home.php, products.php หรือ about.php) โดยพิมพ์ชื่อไฟล์ลงใน URL โดยตรง หลังจากนั้นจะแสดงหน้า index.php

ไฟล์ index.html มีคำสั่งต่าง ๆ ดังในรูปที่ 10.17 และผลการแสดงดังในรูปที่ 10.18

```

1. <?php
2.     session_start();
3.     echo "<html><head><title>Session</title></head>";
4.     if($_SESSION['login']=='yes'){
5.         echo "<center><body><table width='80%'>";
6.         echo "<form method='post' action='logout.php'>";
7.         echo "<tr valign='middle'><td><center><input type='submit'
value='ออกจากระบบ' style='width:250px'></center></td></tr>";
8.         echo "</form></table></center>";
9.     } else{
10.        echo "<center><body><table width='300px'><tr><td>";
11.        echo "<form method='post' action='login.php'><fieldset
style='border-style: ridge; border-color: red'><legend><b>[ เข้าสู่ระบบ]
</b></legend>";
12.        echo "<center><table><tr><td colspan='2' align='center'><br><hr>
</td></tr>";
13.        echo "<tr><td align='right'><b>ชื่อ: </b></td><td><input type='text'
name='name' size='20'></td></tr>";
14.        echo "<tr><td align='right'><b>รหัสผ่าน: </b></td><td><input
type='password' name='password' size='20'></td></tr>";
15.        echo "<tr><td colspan='2' align='center'><input type='submit'
value=' เข้าสู่ระบบ' style='width:80px'>&nbsp;&nbsp;&nbsp;<input type='reset'
value=' ล้างข้อมูล' style='width:80px'><hr><p><br></td></tr></table>
</center></fieldset></form>";
16.        echo "<form method='post' action='signin_form.php'>";
17.        echo "<center><table><tr><td colspan='2' align='center'><b><font
color='blue'>สมัครสมาชิกใหม่</font></b></td></tr>";
18.        echo "<tr><td><input type='submit' value='สมัครสมาชิก'
style='width:250px'></td></tr></table></center></form></td></tr>
</table></center></body></html>";
19.    }
20. ?>

```

รูปที่ 10.17: คำสั่งต่าง ๆ ในไฟล์ index.php

บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP

บรรทัดที่ 2 คำสั่งเริ่มต้นการทำงานของตัวแปรแบบเซสชัน

บรรทัดที่ 3 เปิดแท็ก <html>

บรรทัดที่ 4 ตรวจสอบว่ามีตัวแปรเซสชันที่ชื่อว่า login มีค่าเท่ากับ yes หรือไม่ ถ้าใช่ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 5 ถึง 8 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 9 แทน

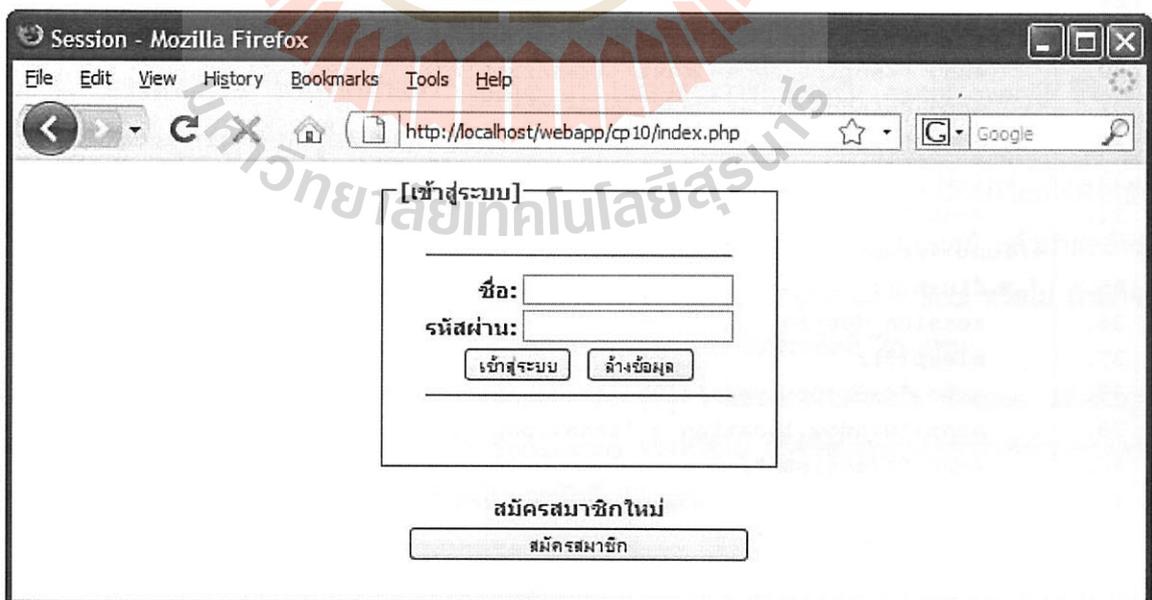
บรรทัดที่ 5 แสดงตารางในตำแหน่งตรงกลางขนาด 80 เปอร์เซ็นต์ของหน้าเว็บ

บรรทัดที่ 6 กำหนดแบบฟอร์มที่มีการส่งข้อมูลแบบ post ไปที่ไฟล์ logout.php

บรรทัดที่ 7 กำหนดแบบฟอร์มให้การป้อนข้อมูลเป็นแบบ submit ตรงตำแหน่งกึ่งกลางของคอลัมน์

บรรทัดที่ 8 ปิดแท็ก </form>, </table> และ</center>

- บรรทัดที่ 9** เข้าทำงานในส่วนของ else ถ้าการตรวจสอบในบรรทัดที่ 4 เป็นเท็จ
- บรรทัดที่ 10** แสดงตารางในตำแหน่งตรงกลางขนาด 300 พิกเซล
- บรรทัดที่ 11** กำหนดแบบฟอร์มที่มีการส่งข้อมูลแบบ post ไปที่ไฟล์ login.php พร้อมกับใช้แท็ก <fieldset> ในการกำหนดรูปแบบ
- บรรทัดที่ 12** แสดงตารางในตำแหน่งตรงกลาง โดยรวมคอลัมน์ทั้งสองเข้าด้วยกันแล้วแสดงเส้นแนวนอน
- บรรทัดที่ 13** แสดงข้อความ “ชื่อ:” ในคอลัมน์ที่ 1 และกำหนดแบบฟอร์มให้การป้อนข้อมูลเป็นแบบเท็กซ์ ชื่อว่า name ขนาด 20 ตัวอักษร
- บรรทัดที่ 14** แสดงข้อความ “รหัสผ่าน:” ในคอลัมน์ที่ 1 และกำหนดแบบฟอร์มให้การป้อนข้อมูลเป็นแบบรหัสผ่าน ชื่อว่า password ขนาด 20 ตัวอักษร
- บรรทัดที่ 15** รวมคอลัมน์ทั้งสองเข้าด้วยกัน แล้วแสดงปุ่มแบบ submit ให้มีค่าเป็น “เข้าสู่ระบบ” และปุ่มแบบ reset ให้มีค่าเป็น “ล้างข้อมูล” ตรงตำแหน่งกึ่งกลางพร้อมกับแสดงเส้นแนวนอนแล้วปิดแท็ก </td>, </tr>, </table>, </center>, </fieldset> และ </form>
- บรรทัดที่ 16** กำหนดแบบฟอร์มที่มีการส่งข้อมูลแบบ post ไปที่ไฟล์ signin_form.php
- บรรทัดที่ 17** แสดงตารางในตำแหน่งตรงกลาง โดยรวมคอลัมน์ทั้งสองเข้าด้วยกันแล้วแสดงข้อความ “สมัครสมาชิกใหม่” เป็นสีน้ำเงิน
- บรรทัดที่ 18** รวมคอลัมน์ทั้งสองเข้าด้วยกัน แล้วแสดงปุ่มแบบ submit ให้มีค่าเป็น “สมัครสมาชิก” ตรงตำแหน่งกึ่งกลางแล้วปิดแท็ก </td>, </tr>, </table>, </center>, </form>, </td>, </tr>, </table>, </center>, </body> และ </html>
- บรรทัดที่ 19** จบการทำงานของ else
- บรรทัดที่ 20** จบการทำงานของไฟล์ index.php



รูปที่ 10.18: ผลการแสดงผลของไฟล์ index.php

ไฟล์ login.php มีคำสั่งต่าง ๆ ดังในรูปที่ 10.19

```

1.  <?php
2.      session_start();
3.      $name = $_POST[ 'name' ] ;
4.      $password = $_POST[ 'password' ] ;
5.      $link = mysql_connect("localhost","root","meroot");
6.      if (!$link) {
7.          die("Could not connect: ".mysql_error());
8.      }
9.      $sql = mysql_select_db('session_test');
10.     if(!$sql){
11.         echo "Error: Database \"session_test\" cannot be found<br>";
12.         exit();
13.     }
14.     $sql = "select * from user";
15.     $result = mysql_query($sql);
16.     while($col = mysql_fetch_array($result)){
17.         if($col[ 0]==$name && $col[ 1]==$password){
18.             $_SESSION[ 'name' ]=$name;
19.             $_SESSION[ 'password' ]=$password;
20.             $valid='yes';
21.             break;
22.         } else{
23.             $valid='no';
24.         }
25.     }
26.     if($valid=='yes'){
27.         echo "<center><p><p><b>ยินดีต้อนรับ <font
color='blue'>".$_SESSION[ 'name' ] . "</font></b></center>";
28.         flush();
29.         sleep(3);
30.         echo "<script type='text/javascript'>";
31.         echo "window.location = 'home.php'";
32.         echo "</script>";
33.     } else{
34.         echo "<center><p><p><font color='red'><b>ชื่อที่กรอกผิดพลาดไม่ถูกต้อง</b>
</font></center>";
35.         flush();
36.         session_destroy();
37.         sleep(3);
38.         echo "<script type='text/javascript'>";
39.         echo "window.location = 'index.php'";
40.         echo "</script>";
41.     }
42.     mysql_close($link);
43.     ?>

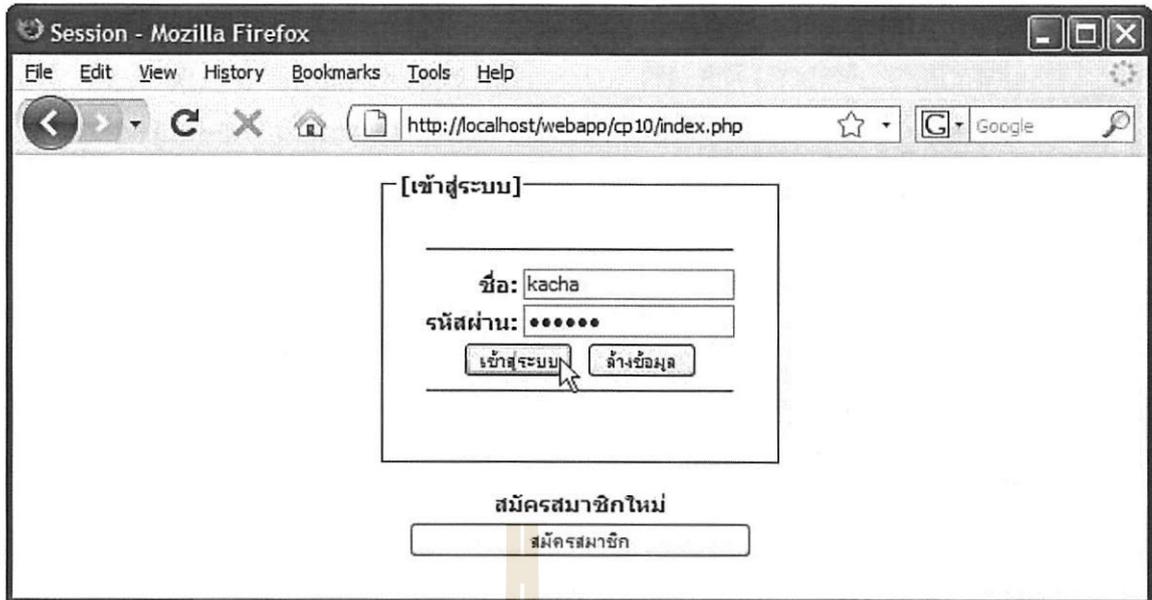
```

รูปที่ 10.19: คำสั่งต่าง ๆ ในไฟล์ login.php

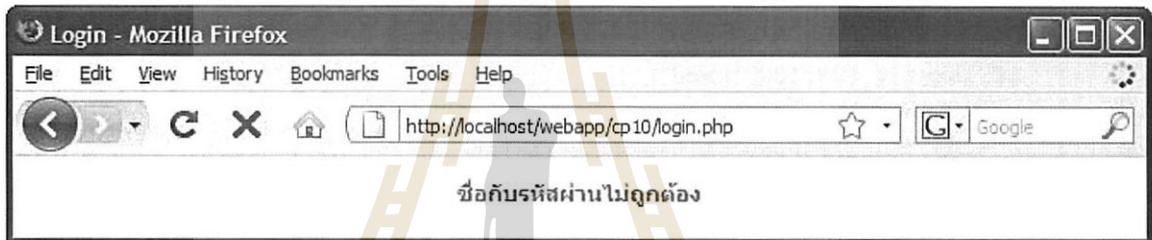
- บรรทัดที่ 1** คำสั่งเปิดแท็ก PHP
- บรรทัดที่ 2** คำสั่งเริ่มต้นการทำงานของตัวแปรแบบเซสชัน
- บรรทัดที่ 3** ประกาศตัวแปร \$name ให้เก็บข้อมูลที่ส่งมาจากแบบฟอร์มแบบ POST ที่มีชื่อว่า name
- บรรทัดที่ 4** ประกาศตัวแปร \$password ให้เก็บข้อมูลที่ส่งมาจากแบบฟอร์มแบบ POST ที่มีชื่อว่า password
- บรรทัดที่ 5** ประกาศตัวแปร \$link ให้เก็บข้อมูลในการติดต่อกับเซิร์ฟเวอร์โดยใช้ฟังก์ชัน mysql_connect() ติดต่อไปที่เซิร์ฟเวอร์ชื่อว่า localhost โดยผู้ใช้คือ root และรหัสผ่านคือ meroot
- บรรทัดที่ 6** ตรวจสอบการเชื่อมต่อกับเซิร์ฟเวอร์ ถ้าติดต่อไม่ได้ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 7 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 9 แทน
- บรรทัดที่ 7** ฟังก์ชัน die() ซึ่งจะแสดงข้อความ “Could not connect: ” และฟังก์ชัน mysql_error() จะแสดงข้อผิดพลาดจากการเรียกใช้คำสั่ง mysql ที่ผ่านมาล่าสุด แล้วจบการทำงานของสคริปต์นี้ทันที
- บรรทัดที่ 8** จบการทำงานของคำสั่ง if
- บรรทัดที่ 9** ประกาศตัวแปร \$sql ให้เก็บข้อมูลในการใช้ฟังก์ชัน mysql_select_db() ซึ่งจะทำการเลือกฐานข้อมูลชื่อว่า session_test
- บรรทัดที่ 10** ตรวจสอบการเลือกฐานข้อมูล ถ้าเลือกฐานข้อมูลดังกล่าวไม่ได้ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 11 ถึง 12 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 14 แทน
- บรรทัดที่ 11** แสดงข้อความ “Error: Database "session_test" cannot be found”
- บรรทัดที่ 12** เรียกใช้ฟังก์ชัน exit() เพื่อจบการทำงานของสคริปต์
- บรรทัดที่ 13** จบการทำงานของคำสั่ง if
- บรรทัดที่ 14** ประกาศตัวแปร \$sql ให้เก็บข้อมูล “select * from user” ซึ่งเป็นคำสั่ง sql ในการเลือกข้อมูลทั้งหมดจากตารางชื่อว่า user
- บรรทัดที่ 15** ประกาศตัวแปร \$result ให้เก็บข้อมูลในการใช้ฟังก์ชัน mysql_query() ซึ่งจะทำการ query ข้อมูลที่ถูกเก็บอยู่ในตัวแปร \$sql
- บรรทัดที่ 16** ใช้ลูป while ในการวนรอบ ซึ่งในแต่ละรอบนั้นจะเรียกใช้ฟังก์ชัน mysql_fetch_arrays() เพื่อดึงข้อมูลจากตัวแปร \$result เป็นจำนวนแถวออกมาเป็นอะเรย์ แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ในตัวแปร \$col โดยจะตรวจสอบว่ามีค่าที่ถูกนำมาเก็บอยู่ที่ตัวแปร \$col หรือไม่ ถ้ามีก็จะเข้าไปทำคำสั่งในบรรทัดที่ 17 – 25 แต่ถ้าไม่จริงก็จะข้ามไปทำคำสั่งในบรรทัดที่ 26 แทน
- บรรทัดที่ 17** ตรวจสอบข้อมูลของตัวแปร \$col[0] ว่ามีค่าเท่ากับตัวแปร \$name และข้อมูลของตัวแปร \$col[1] ว่ามีค่าเท่ากับตัวแปร \$password จริงหรือไม่ ถ้าจริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 18 ถึง 21 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 22 แทน
- บรรทัดที่ 18** กำหนดตัวแปรแบบเซสชันชื่อว่า name ให้เก็บค่าที่อยู่ในตัวแปร \$name
- บรรทัดที่ 19** กำหนดตัวแปรแบบเซสชันชื่อว่า password ให้เก็บค่าที่อยู่ในตัวแปร \$password
- บรรทัดที่ 20** กำหนดตัวแปร \$valid ให้เก็บค่า “yes”
- บรรทัดที่ 21** ใช้คำสั่ง break; เพื่อหลุดออกจากลูป while

- บรรทัดที่ 22 จบการทำงานของ if แล้วเริ่มเข้าสู่การทำงานในส่วนของ else
- บรรทัดที่ 23 กำหนดตัวแปร \$valid ให้เก็บค่า “no”
- บรรทัดที่ 24 จบการทำงานของ else
- บรรทัดที่ 25 จบการทำงานของลูป while
- บรรทัดที่ 26 ตรวจสอบข้อมูลของตัวแปร \$valid ว่ามีค่าเท่ากับ “yes” จริงหรือไม่ ถ้าจริงจะเข้าไปทำคำสั่งในบรรทัดที่ 27 ถึง 32 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 34 แทน
- บรรทัดที่ 27 แสดงข้อความ “ยินดีต้อนรับ” พร้อมกับแสดงค่าของตัวแปรแบบเซสชันที่ชื่อว่า name ออกที่หน้าเว็บเป็นตัวหนาสีน้ำเงิน
- บรรทัดที่ 28 ฟังก์ชัน flush() จะแสดงข้อมูลที่มีอยู่ในบัฟเฟอร์ออกที่หน้าเว็บ
- บรรทัดที่ 29 ฟังก์ชัน sleep() จะทำการหน่วงเวลา (หน่วยเป็นวินาที)
- บรรทัดที่ 30 – 32 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ home.php
- บรรทัดที่ 33 เริ่มเข้าสู่การทำงานในส่วนของ else
- บรรทัดที่ 34 แสดงข้อความ “ชื่อกรหัสผ่านไม่ถูกต้อง” ออกที่หน้าเว็บเป็นตัวหนาสีแดง
- บรรทัดที่ 35 ฟังก์ชัน flush() จะแสดงข้อมูลที่มีอยู่ในบัฟเฟอร์ออกที่หน้าเว็บ
- บรรทัดที่ 36 ฟังก์ชัน session_destroy() จะลบตัวแปรแบบเซสชันทั้งหมดทิ้ง
- บรรทัดที่ 37 ฟังก์ชัน sleep() จะทำการหน่วงเวลา (หน่วยเป็นวินาที)
- บรรทัดที่ 38 – 40 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ index.php
- บรรทัดที่ 41 จบการทำงานของ else
- บรรทัดที่ 42 เรียกใช้ฟังก์ชัน mysql_close() เพื่อทำการปิดการเชื่อมต่อกับฐานข้อมูลที่ตัวแปร \$link ได้เชื่อมต่อเอาไว้
- บรรทัดที่ 43 จบการทำงานของไฟล์ index.php

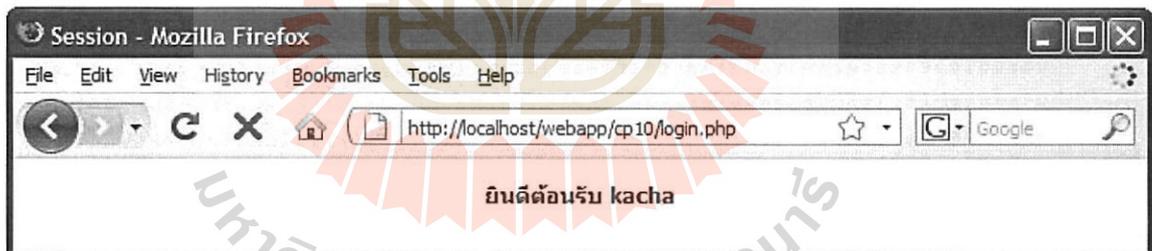
เมื่อผู้ใช้ป้อนข้อมูล (ชื่อและรหัสผ่าน) เพื่อเข้าสู่ระบบดังแสดงในรูปที่ 10.20 แล้วกดปุ่ม “เข้าสู่ระบบ” ไฟล์ login.php จะทำการติดต่อกับฐานข้อมูลชื่อว่า session_test แล้วตรวจสอบข้อมูลดังกล่าวกับข้อมูลที่อยู่ในตารางชื่อว่า user ถ้าข้อมูลไม่ตรงก็จะแสดงข้อความดังในรูปที่ 10.21 แล้วกลับไปหน้าจอ index.php แต่ถ้าข้อมูลตรงกันก็จะแสดงข้อความดังในรูปที่ 10.22 แล้วเปิดหน้า home.php ดังแสดงในรูปที่ 10.23



รูปที่ 10.20: การป้อนข้อมูลเพื่อเข้าสู่ระบบ



รูปที่ 10.21: แสดงข้อความเมื่อชื่อกับรหัสผ่านไม่ถูกต้อง



รูปที่ 10.22: แสดงข้อความเมื่อชื่อกับรหัสผ่านถูกต้อง

บรรทัดที่ 8 แสดงข้อความ “logout” เพื่อเป็นจุดเชื่อมต่อไปยังไฟล์ logout.php พร้อมช่องว่าง 3 ช่องและข้อความ “: :” และเส้นตรงแนวนอน

บรรทัดที่ 9 จบการทำงานของ if และเข้าสู่การทำงานในส่วนของ else ถ้าการตรวจสอบในบรรทัดที่ 3 เป็นเท็จ

บรรทัดที่ 10 – 12 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ error.php

บรรทัดที่ 13 จบการทำงานของ else

บรรทัดที่ 14 จบการทำงานของไฟล์ header.php

ไฟล์ error.php มีคำสั่งต่าง ๆ ดังในรูปที่ 10.26

```

1. <?php
2.     session_start();
3.     echo "<center><p><p><font color='red'><b>คุณยังไม่ได้ลงทะเบียนเข้าสู่ระบบ
   </b></font></center>";
4.     flush();
5.     session_destroy();
6.     sleep(3);
7.     echo "<script type='text/javascript'>";
8.     echo "window.location = 'index.php'";
9.     echo "</script>";
10. ?>
    
```

รูปที่ 10.26: คำสั่งต่าง ๆ ในไฟล์ error.php

บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP

บรรทัดที่ 2 คำสั่งเริ่มต้นการทำงานของตัวแปรแบบเซสชัน

บรรทัดที่ 3 แสดงข้อความ “คุณยังไม่ได้ลงทะเบียนเข้าสู่ระบบ” แบบตัวหนาสีแดงตำแหน่งตรงกลาง

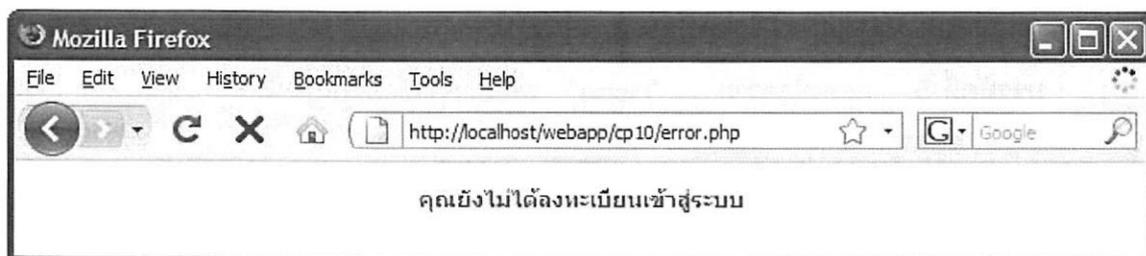
บรรทัดที่ 4 ฟังก์ชัน flush() จะแสดงข้อมูลที่มีอยู่ในบัฟเฟอร์ออกที่หน้าเว็บ

บรรทัดที่ 5 ฟังก์ชัน session_destroy() จะลบตัวแปรแบบเซสชันทั้งหมดทิ้ง

บรรทัดที่ 6 ฟังก์ชัน sleep() จะทำการหน่วงเวลา (หน่วยเป็นวินาที)

บรรทัดที่ 7 – 9 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ index.php

บรรทัดที่ 10 จบการทำงานของไฟล์ error.php



รูปที่ 10.27: การแสดงผลของไฟล์ error.php

ไฟล์ products.php มีคำสั่งต่าง ๆ ดังในรูปที่ 10.28

```

1. <?php
2.     session_start();
3.     if(isset($_SESSION['name'])) {
4.         include('header.php');
5.         echo "<html><head><title>Products</title>";
6.         echo "<body><br><b>ยินดีต้อนรับ... </b>";
7.         echo "<font color='blue'><b>".$_SESSION['name']."</b></font><p>";
8.         echo "<center><table border='1'>";
9.         echo "<th>SUT-MDE 3.0</th>";
10.        echo "<th>SUT-MOTS 3.0</th>";
11.        echo "<th>SUTinsServer 5201</th>";
12.        echo "<th>SUTinsDesktop 5201</th>";
13.        echo "<tr>";
14.        echo "<td><img src='dvd_SUT-MDE30.jpg' width='180px' height='180px'
15.        alt='dvd_SUT-MDE30'></td>";
16.        echo "<td><img src='dvd_SUT-MOTS30.jpg' width='180px'
17.        height='180px' alt='dvd_SUT-MOTS30'></td>";
18.        echo "<td><img src='dvd_SUTinsServer_5201.jpg' width='180px'
19.        height='180px' alt='dvd_SUTinsServer_5201'></td>";
20.        echo "<td><img src='dvd_SUTinsDesktop_5201.jpg' width='180px'
21.        height='180px' alt='dvd_SUTinsDesktop_5201'></td>";
22.        echo "</tr></table></center></html>";
23.    } else {
24.        echo "<script type='text/javascript'>";
25.        echo "window.location = 'error.php'";
26.        echo "</script>";
27.    }
28. ?>

```

รูปที่ 10.28: คำสั่งต่าง ๆ ในไฟล์ products.php

- บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP
- บรรทัดที่ 2 คำสั่งเริ่มต้นการทำงานของตัวแปรแบบเซสชัน
- บรรทัดที่ 3 ตรวจสอบว่ามีตัวแปรเซสชันที่ชื่อว่า name หรือไม่ ถ้ามีก็จะเข้าไปทำคำสั่งในบรรทัดที่ 4 ถึง 22 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 23 แทน
- บรรทัดที่ 4 รวมไฟล์ header.php
- บรรทัดที่ 5 เปิดแท็ก <html>
- บรรทัดที่ 6 แสดงข้อความ “ยินดีต้อนรับ...” แบบตัวหนา
- บรรทัดที่ 7 แสดงค่าของตัวแปรแบบเซสชันที่ชื่อว่า name ออกที่หน้าเว็บเป็นตัวหนาสีน้ำเงิน
- บรรทัดที่ 8 แสดงตารางที่มีเส้นขอบหนาเท่ากับ 1 ตรงกลางหน้าเว็บ
- บรรทัดที่ 9 แสดงส่วนหัวของตาราง (heading) เป็นข้อความ “SUT-MDE 3.0”
- บรรทัดที่ 10 แสดงส่วนหัวของตาราง (heading) เป็นข้อความ “SUT-MOTS 3.0”
- บรรทัดที่ 11 แสดงส่วนหัวของตาราง (heading) เป็นข้อความ “SUTinsServer 5201”
- บรรทัดที่ 12 แสดงส่วนหัวของตาราง (heading) เป็นข้อความ “SUTinsDesktop 5201”

บรรทัดที่ 13 เปิดแท็ก <tr>

บรรทัดที่ 14 แสดงรูปภาพชื่อว่า “dvd_SUT-MDE30.jpg” ในคอลัมน์ที่ 1

บรรทัดที่ 15 แสดงรูปภาพชื่อว่า “dvd_SUT-MOTS30.jpg” ในคอลัมน์ที่ 2

บรรทัดที่ 16 แสดงรูปภาพชื่อว่า “dvd_SUTinsServer_5201.jpg” ในคอลัมน์ที่ 3

บรรทัดที่ 17 แสดงรูปภาพชื่อว่า “dvd_SUTinsDesktop_5201.jpg” ในคอลัมน์ที่ 4

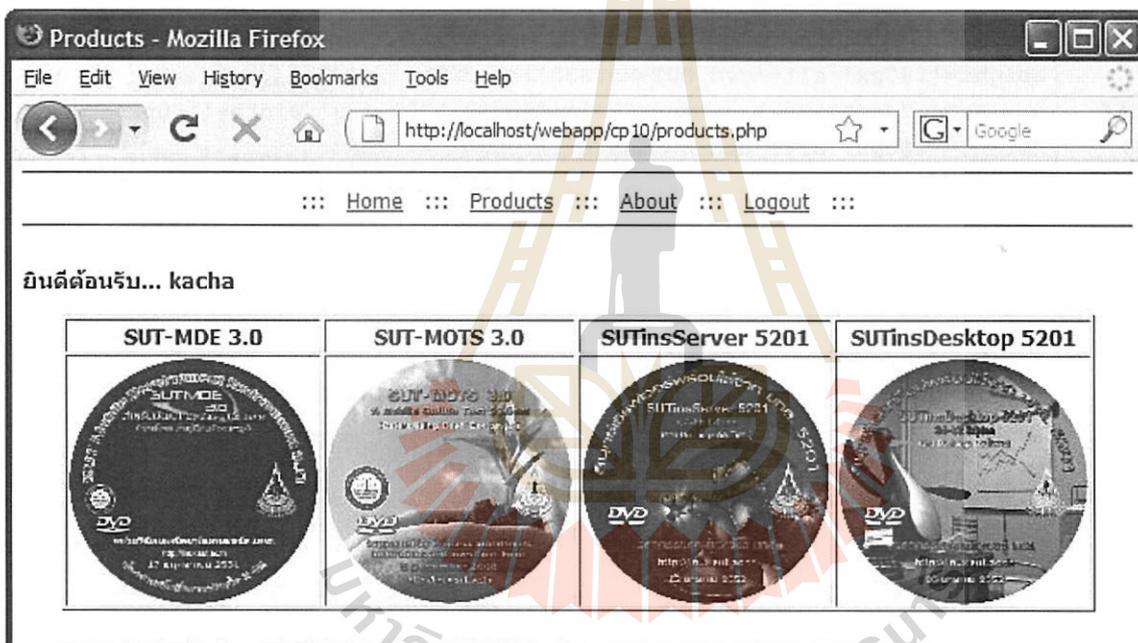
บรรทัดที่ 18 ปิดแท็ก </tr>, </table>, </center> และ </html>

บรรทัดที่ 19 จบการทำงานของ if แล้วเข้าสู่การทำงานในส่วนของ else ถ้าการตรวจสอบในบรรทัดที่ 3 เป็นเท็จ

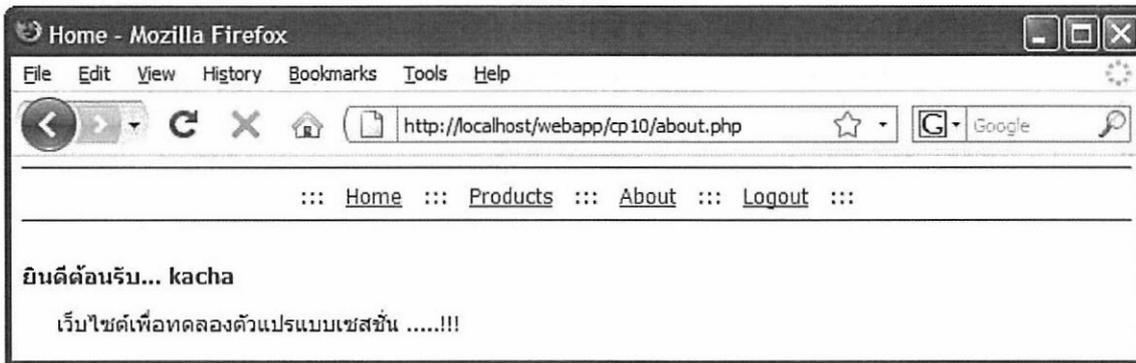
บรรทัดที่ 20 – 22 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ error.php

บรรทัดที่ 23 จบการทำงานของ else

บรรทัดที่ 24 จบการทำงานของไฟล์ products.php



รูปที่ 10.29: การแสดงผลของไฟล์ products.php



รูปที่ 10.31: การแสดงผลของไฟล์ about.php

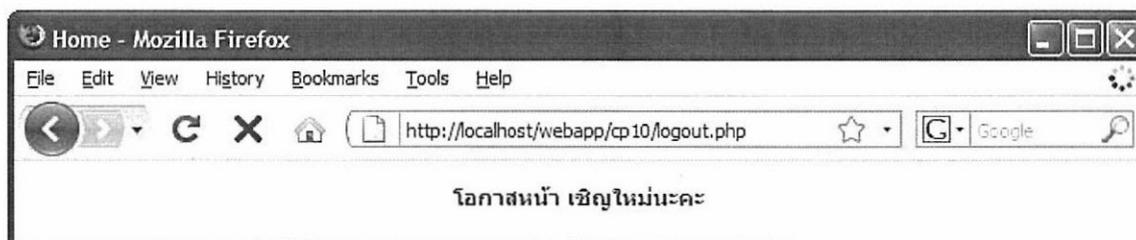
ไฟล์ logout.php มีคำสั่งต่าง ๆ ดังในรูปที่ 10.32

```

1.  <?php
2.     session_start();
3.     echo "<center><p><p><font color='blue'><b>โอกาสหน้า เชิญใหม่ นะคะ
</b></font></center>";
4.     flush();
5.     session_destroy();
6.     sleep(3);
7.     echo "<script type='text/javascript'>";
8.     echo "window.location = 'index.php'";
9.     echo "</script>";
10. ?>
    
```

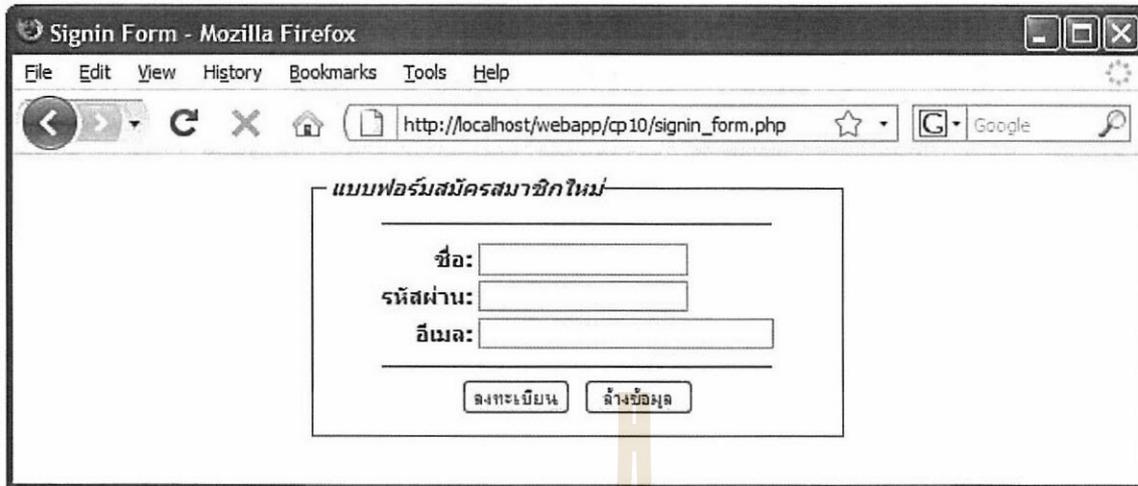
รูปที่ 10.32: คำสั่งต่าง ๆ ในไฟล์ logout.php

- บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP
- บรรทัดที่ 2 คำสั่งเริ่มต้นการทำงานของตัวแปรแบบเซสชัน
- บรรทัดที่ 3 แสดงข้อความ “โอกาสหน้า เชิญใหม่ นะคะ” แบบตัวหนาสีน้ำเงินตำแหน่งตรงกลาง
- บรรทัดที่ 4 ฟังก์ชัน flush() จะแสดงข้อมูลที่มีอยู่ในบัฟเฟอร์ออกที่หน้าเว็บ
- บรรทัดที่ 5 ฟังก์ชัน session_destroy() จะลบตัวแปรแบบเซสชันทั้งหมดทิ้ง
- บรรทัดที่ 6 ฟังก์ชัน sleep() จะทำการหน่วงเวลา (หน่วยเป็นวินาที)
- บรรทัดที่ 7 – 9 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ index.php
- บรรทัดที่ 10 จบการทำงานของไฟล์ logout.php



รูปที่ 10.33: การแสดงผลของไฟล์ logout.php

บรรทัดที่ 13 จบการทำงานของไฟล์ signin_form.php



รูปที่ 10.35: การแสดงผลของไฟล์ signin_form.php

จากภาพที่ 10.35 นั้น เมื่อผู้ใช้กดปุ่ม “ลงทะเบียน” ข้อมูลต่าง ๆ ในแบบฟอร์มจะถูกส่งไปยังไฟล์ signin.php โดยไฟล์ signin.php จะทำการตรวจสอบว่าข้อมูลทั้งหมด (ชื่อ รหัสผ่าน และอีเมล) มีครบหรือไม่ ถ้ามีข้อมูลไม่ครบถ้วนจะแสดงข้อความ “กรอกข้อมูลไม่ครบถ้วน” ดังแสดงในรูปที่ 10.36 แล้วแสดงไฟล์ index.php แต่ถ้ามีข้อมูลครบถ้วนจะทำการเชื่อมต่อเซิร์ฟเวอร์แล้วเลือกฐานข้อมูลพร้อมกับบันทึกข้อมูลเหล่านั้นลงในตารางแล้วจึงแสดงไฟล์ home.php



รูปที่ 10.36: การแสดงผลของไฟล์ signin.php เมื่อป้อนข้อมูลไม่ครบถ้วน

ไฟล์ signin.php มีคำสั่งต่าง ๆ ดังในรูปที่ 10.37

```

1. <?php
2.     session_start();
3.     $name = $_POST[ "name" ] ;
4.     $password = $_POST[ "password" ] ;
5.     $email = $_POST[ "email" ] ;
6.     if($name==" " || $password==" " || $email==""){
7.         echo "<center><p><p><font color='red'><b>กรอกข้อมูลไม่ครบถ้วน
</b></font></p></center>";
8.         flush();
9.         session_destroy();
10.        sleep(3);
11.        echo "<script type='text/javascript'>";
12.        echo "window.location = 'index.php'";
13.        echo "</script>";
14.    }
15.    $link = mysql_connect("localhost","root","meroot");
16.    if(!$link){
17.        die("Could not connect: ".mysql_error());
18.    }
19.    $sql = mysql_select_db('session_test');
20.    if(!$sql){
21.        echo "Error: Database \"session_test\" cannot be found<br>";
22.        exit();
23.    }
24.    $sql = "INSERT INTO user VALUES
25.    (\\"$name\\",\\"$password\\",\\"$email\\")";
26.    if(!mysql_query($sql)){
27.        echo "Cannot insert into table<br>";
28.    }else{
29.        echo "<center><p><p><b>ยินดีต้อนรับ <font color='blue'>
30.        $name</font></b></center>";
31.        $_SESSION[ 'name' ] = $name;
32.        $_SESSION[ 'password' ] = $password;
33.        flush();
34.        sleep(3);
35.        echo "<script type='text/javascript'>";
36.        echo "window.location = 'home.php'";
37.        echo "</script>";
38.    }
39.    mysql_close($link);
40. ?>

```

รูปที่ 10.37: คำสั่งต่าง ๆ ในไฟล์ signin_form.php

- บรรทัดที่ 1 คำสั่งเปิดแท็ก PHP
- บรรทัดที่ 2 คำสั่งเริ่มต้นการทำงานของตัวแปรแบบเซสชัน
- บรรทัดที่ 3 ประกาศตัวแปร \$name ให้เก็บข้อมูลที่ส่งมาจากแบบฟอร์มแบบ POST ที่มีชื่อว่า

name

บรรทัดที่ 4 ประกาศตัวแปร `$password` ให้เก็บข้อมูลที่ส่งมาจากแบบฟอร์มแบบ POST ที่มีชื่อว่า `password`

บรรทัดที่ 5 ประกาศตัวแปร `$email` ให้เก็บข้อมูลที่ส่งมาจากแบบฟอร์มแบบ POST ที่มีชื่อว่า `email`

บรรทัดที่ 6 ตรวจสอบตัวแปร `$name` มีค่าเท่ากับช่องว่าง หรือตัวแปร `$password` มีค่าเท่ากับช่องว่างหรือตัวแปร `$email` มีค่าเท่ากับช่องว่างหรือไม่ ถ้าใช่จะเข้าไปทำคำสั่งในบรรทัดที่ 7 ถึง 13 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 15 แทน

บรรทัดที่ 7 แสดงข้อความ “กรอกข้อมูลไม่ครบถ้วน” ออกที่หน้าเว็บเป็นตัวหนาสีแดงตำแหน่งตรงกลาง

บรรทัดที่ 8 ฟังก์ชัน `flush()` จะแสดงข้อมูลที่มีอยู่ในบัฟเฟอร์ออกที่หน้าเว็บ

บรรทัดที่ 9 ฟังก์ชัน `session_destroy()` จะลบตัวแปรแบบเซสชันทั้งหมดทิ้ง

บรรทัดที่ 10 ฟังก์ชัน `sleep()` จะทำการหน่วงเวลา (หน่วยเป็นวินาที)

บรรทัดที่ 11 – 13 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ `index.php`

บรรทัดที่ 14 จบการทำงานของ `if`

บรรทัดที่ 15 ประกาศตัวแปร `$link` ให้เก็บข้อมูลในการติดต่อกับเซิร์ฟเวอร์โดยใช้ฟังก์ชัน `mysql_connect()` ติดต่อไปที่เซิร์ฟเวอร์ชื่อว่า `localhost` โดยผู้ใช้คือ `root` และรหัสผ่านคือ `meroot`

บรรทัดที่ 16 ตรวจสอบการเชื่อมต่อกับเซิร์ฟเวอร์ ถ้าติดต่อไม่ได้ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 17 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 19 แทน

บรรทัดที่ 17 ฟังก์ชัน `die()` ซึ่งจะแสดงข้อความ “Could not connect: ” และฟังก์ชัน `mysql_error()` จะแสดงข้อผิดพลาดจากการเรียกใช้คำสั่ง `mysql` ที่ผ่านมาล่าสุด แล้วจบการทำงานของสคริปต์นี้ทันที

บรรทัดที่ 18 จบการทำงานของคำสั่ง `if`

บรรทัดที่ 19 ประกาศตัวแปร `$sql` ให้เก็บข้อมูลในการใช้ฟังก์ชัน `mysql_select_db()` ซึ่งจะทำให้การเลือกฐานข้อมูลชื่อว่า `session_test`

บรรทัดที่ 20 ตรวจสอบการเลือกฐานข้อมูล ถ้าเลือกฐานข้อมูลดังกล่าวไม่ได้ก็จะเข้าไปทำคำสั่งในบรรทัดที่ 21 ถึง 22 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 24 แทน

บรรทัดที่ 21 แสดงข้อความ “Error: Database "session_test" cannot be found”

บรรทัดที่ 22 เรียกใช้ฟังก์ชัน `exit()` เพื่อจบการทำงานของสคริปต์

บรรทัดที่ 23 จบการทำงานของคำสั่ง `if`

บรรทัดที่ 24 ประกาศตัวแปร `$sql` ให้เก็บข้อมูล “INSERT INTO user VALUES (\"\$name\", \"\$password\", \"\$email\")” ซึ่งเป็นคำสั่ง `sql` ในการใส่ข้อมูลลงในตารางที่ชื่อว่า `user` โดยข้อมูลต่าง ๆ คือค่าที่ถูกเก็บไว้ในตัวแปร `$name`, `$password` และ `$email` ตามลำดับ

บรรทัดที่ 25 ตรวจสอบการทำงานของการทำงานของการเรียกใช้ฟังก์ชัน `mysql_query()` ซึ่งจะทำให้การ `query` ข้อมูลที่ถูกเก็บอยู่ในตัวแปร `$sql` ถ้าไม่สามารถทำได้จริงจะเข้าไปทำคำสั่งในบรรทัดที่ 26 มิฉะนั้นแล้วจะกระโดดไปทำคำสั่งในบรรทัดที่ 27 ถึง 36 แทน

บรรทัดที่ 26 แสดงข้อความ “Cannot insert into table”

บรรทัดที่ 27 จบการทำงานของ if แล้วเข้าสู่การทำงานในส่วนของ else ถ้าการตรวจสอบในบรรทัดที่ 25 เป็นเท็จ

บรรทัดที่ 28 แสดงข้อความ “ยินดีต้อนรับ” พร้อมกับแสดงค่าของตัวแปร \$name ออกที่หน้าเว็บเป็นตัวหนาสีน้ำเงินตำแหน่งตรงกลาง

บรรทัดที่ 29 กำหนดตัวแปรแบบเซสชันชื่อว่า name ให้เก็บค่าที่อยู่ในตัวแปร \$name

บรรทัดที่ 30 กำหนดตัวแปรแบบเซสชันชื่อว่า password ให้เก็บค่าที่อยู่ในตัวแปร \$password

บรรทัดที่ 31 ฟังก์ชัน flush() จะแสดงข้อมูลที่มีอยู่ในบัฟเฟอร์ออกที่หน้าเว็บ

บรรทัดที่ 32 ฟังก์ชัน sleep() จะทำการหน่วงเวลา (หน่วยเป็นวินาที)

บรรทัดที่ 33 – 35 เป็นคำสั่ง JavaScript เพื่อเปิดไฟล์ home.php

บรรทัดที่ 36 จบการทำงานของ else

บรรทัดที่ 37 เรียกใช้ฟังก์ชัน mysql_close() เพื่อทำการปิดการเชื่อมต่อกับฐานข้อมูลที่ตัวแปร \$link ได้เชื่อมต่อเอาไว้

บรรทัดที่ 38 จบการทำงานของไฟล์ signin.php

หมายเหตุ ตัวแปรเซสชันจะหมดอายุเมื่อโปรแกรมเว็บเบราว์เซอร์ปิดลง ส่วนไฟล์ที่จะเรียกใช้ตัวแปรเซสชันนี้จะอยู่ในไดเรกทอรีใดก็ได้

สรุป

ข้อมูลจากหน้าเว็บหนึ่ง ๆ สามารถส่งต่อไปยังหน้าเว็บอื่น ๆ ได้โดยใช้ตัวแปรแบบคุกกี้ซึ่งจะถูกเก็บไว้ในเครื่องไคลเอนท์หรือตัวแปรแบบเซสชันซึ่งจะถูกเก็บไว้ในเครื่องเซิร์ฟเวอร์ ตัวแปรทั้ง 2 แบบนั้นต่างก็มีประโยชน์ในการจดจำข้อมูลและตรวจสอบสถานะต่าง ๆ ของผู้ที่เข้ามาเยี่ยมชมเว็บ ซึ่งผู้ดูแลเว็บสามารถออกแบบและเก็บข้อมูลต่าง ๆ ได้ตามต้องการ ทำให้การบริหารจัดการเว็บไซต์เป็นไปได้ด้วยความสะดวก ที่สำคัญที่สุดก็คือ ผู้พัฒนาระบบจะต้องระมัดระวังในการใช้ตัวแปรทั้งสองแบบ เนื่องจากตัวแปรแบบคุกกี้จะหมดอายุหรือใช้งานไม่ได้เมื่อถึงเวลาที่กำหนดโดยผู้พัฒนาระบบ ส่วนไฟล์ที่จะเรียกใช้ตัวแปรคุกกี้ก็ต้องอยู่ในโฟลเดอร์เดียวกันกับไฟล์ที่สร้างตัวแปรคุกกี้ ส่วนตัวแปรแบบเซสชันนั้นจะหมดอายุเมื่อโปรแกรมเว็บเบราว์เซอร์ถูกปิดลง ส่วนไฟล์ที่จะเรียกใช้ตัวแปรเซสชันนี้จะอยู่ในโฟลเดอร์ใดก็ได้

แบบฝึกหัด

1. ให้สร้างฐานข้อมูลขึ้นมาหนึ่งฐานข้อมูลชื่อว่า session โดยมีตารางชื่อว่า user เก็บข้อมูล ชื่อ และรหัสผ่านจำนวน 2 เรคคอร์ด แล้วเขียนไฟล์ index.php เพื่อแสดงการรับข้อมูลซึ่งประกอบไปด้วย ชื่อ รหัสผ่าน และรหัสลับ (ดังรูปที่ 10.38)

รูปที่ 10.38: การแสดงผลของไฟล์ index.php และรหัสลับ FZE

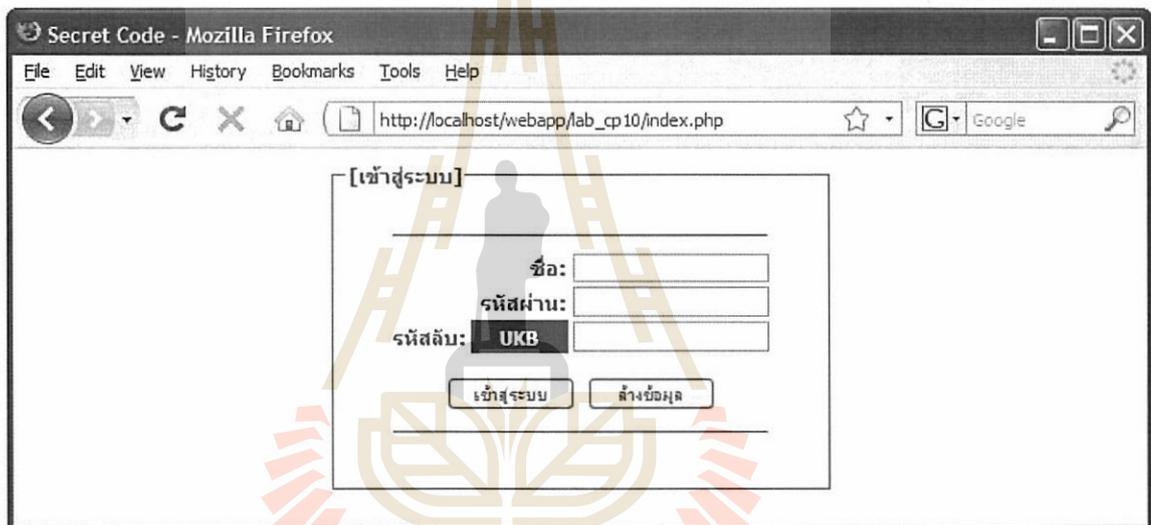
ซึ่งรหัสลับนั้นจะถูกสุ่มขึ้นมาใหม่ทุกครั้งไฟล์ index.php ถูกเรียกใช้ เมื่อผู้ใช้กดปุ่ม “เข้าสู่ระบบ” ข้อมูลทั้งหมดจะถูกส่งไปยังไฟล์ login.php เพื่อทำการตรวจสอบข้อมูลทั้งหมดว่าถูกต้องหรือไม่ (ชื่อและรหัสผ่าน ตรวจสอบกับข้อมูลในฐานข้อมูล ส่วนรหัสลับที่ผู้ใช้ป้อนเข้ามานั้นให้ตรวจสอบกับรหัสลับที่ถูกสร้างขึ้นมา) สมมุติว่า ผู้ใช้ป้อนชื่อและรหัสผ่านแต่ไม่ได้ป้อนรหัสลับ แล้วกดปุ่ม “เข้าสู่ระบบ” (ดังรูปที่ 10.39)

รูปที่ 10.39: แสดงการป้อนข้อมูลไม่ครบถ้วน

ไฟล์ login.php ก็ จะแสดงข้อความ “รหัสลับไม่ถูกต้อง” (ดังรูปที่ 10.40) แล้วเปิดหน้า index.php พร้อมกับสร้างรหัสลับขึ้นมาใหม่ (ดังรูปที่ 10.41)

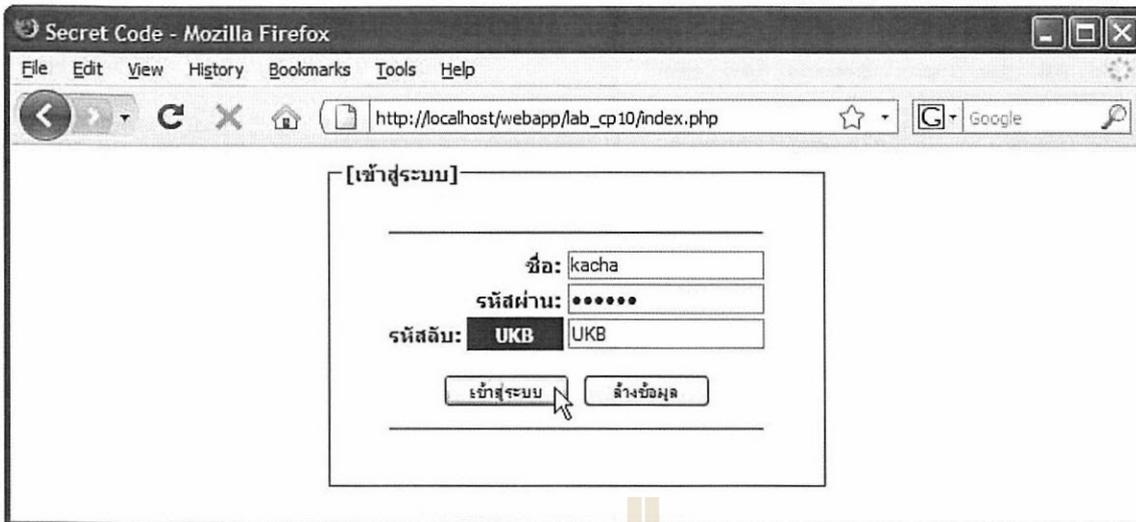


รูปที่ 10.40: แสดงข้อความ "รหัสลับไม่ถูกต้อง"

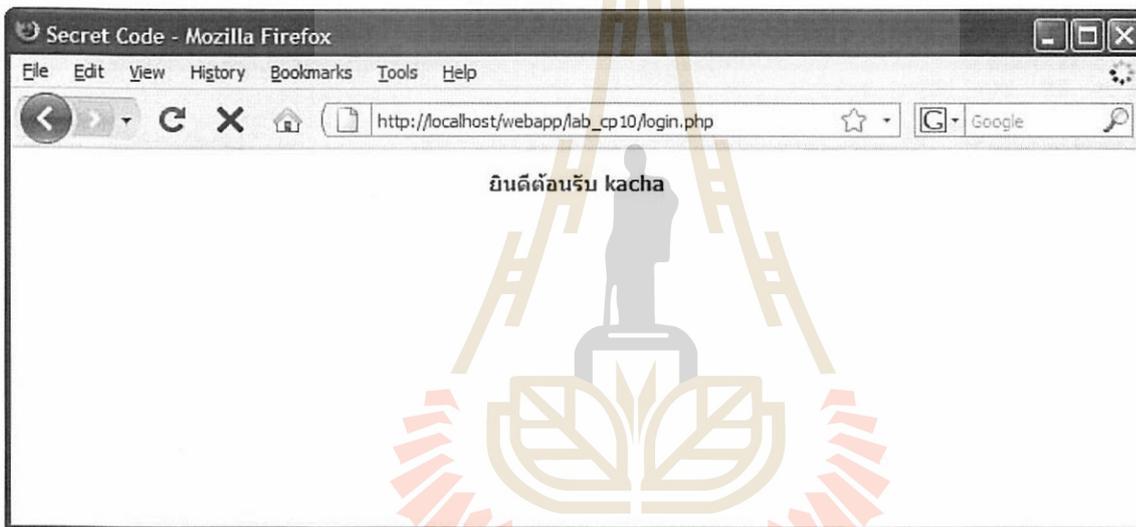


รูปที่ 10.41: การแสดงผลของไฟล์ index.php พร้อมกับสร้างรหัสลับขึ้นมาใหม่คือ UKB

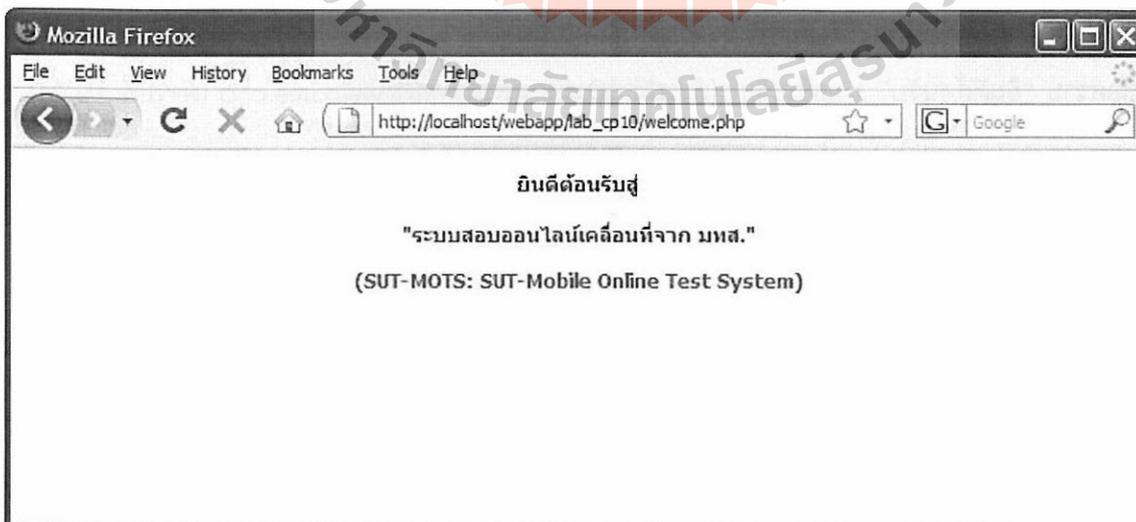
ถ้าผู้ใช้ป้อนข้อมูลครบถ้วนและถูกต้องแล้วกดปุ่ม "เข้าสู่ระบบ" (ดังรูปที่ 10.42) ไฟล์ login.php จะแสดงข้อความ "ยินดีต้อนรับ" ตามด้วยชื่อ (ดังรูปที่ 10.43) แล้วเปิดหน้า welcome.php (ดังรูปที่ 10.44)



รูปที่ 10.42: แสดงการป้อนข้อมูลครบถ้วนและถูกต้อง



รูปที่ 10.43: แสดงข้อความ "ยินดีต้อนรับ" ตามด้วยชื่อ



รูปที่ 10.44: การแสดงผลของไฟล์ welcome.php

บทที่ 11 แนะนำเอแจ็ก

<Introduction to AJAX>

บทนำ

การแสดงผลหรือข้อมูลต่าง ๆ บนหน้าเว็บเพจแต่ละหน้านั้น ไม่ว่าจะมีการติดต่อกับระบบฐานข้อมูลหรือไม่ ต่างก็ต้องทำการกดปุ่มเชื่อมต่อ (Link) หรือกดปุ่มส่งข้อมูล (Submit) ในแบบฟอร์มเพื่อส่งข้อมูลไปที่เว็บเซิร์ฟเวอร์ แล้วรอหน้าเว็บใหม่เพื่อแสดงผลข้อมูล โดยทำการโหลดข้อมูลทั้งหมดของหน้านั้น ๆ ออกมาแสดง ซึ่งในบางครั้งข้อมูลที่จำเป็นต้องโหลดมาแสดงใหม่นี้ มีความแตกต่างกับข้อมูลในหน้าเดิมเฉพาะบางส่วนเท่านั้น แต่ก็ต้องทำการโหลดข้อมูลทั้งหมดใหม่และในบางครั้งอาจจะสังเกตเห็นว่าหน้าจอกเกิดการกระพริบขึ้น เนื่องจากข้อมูลใหม่ได้ถูกโหลดเข้ามาเพื่อแสดงผล ทำให้เกิดความล่าช้า จึงมีการคิดค้นที่จะนำเสนอข้อมูลใหม่ในเฉพาะส่วนที่แตกต่างจากของเดิมเท่านั้น ส่วนข้อมูลไหนที่ยังเหมือนเดิมก็ไม่ต้องโหลดเข้ามาใหม่ ซึ่งจะทำให้แสดงผลได้เร็วขึ้น เกิดการโต้ตอบกับผู้ใช้งานได้ดีขึ้น หน้าจอทั้งหมดไม่กระพริบ นี่จึงเป็นที่มาของคำว่า “เอแจ็ก”

วัตถุประสงค์

1. เพื่อให้เข้าใจหลักการทำงานของเอแจ็ก
2. เพื่อให้รู้ถึงวิธีการแสดงข้อมูลโดยใช้เอแจ็ก
3. เพื่อฝึกเขียนและทดสอบการแสดงข้อมูลโดยใช้เอแจ็ก

เนื้อหา

11.1 เอแจ็กคืออะไร

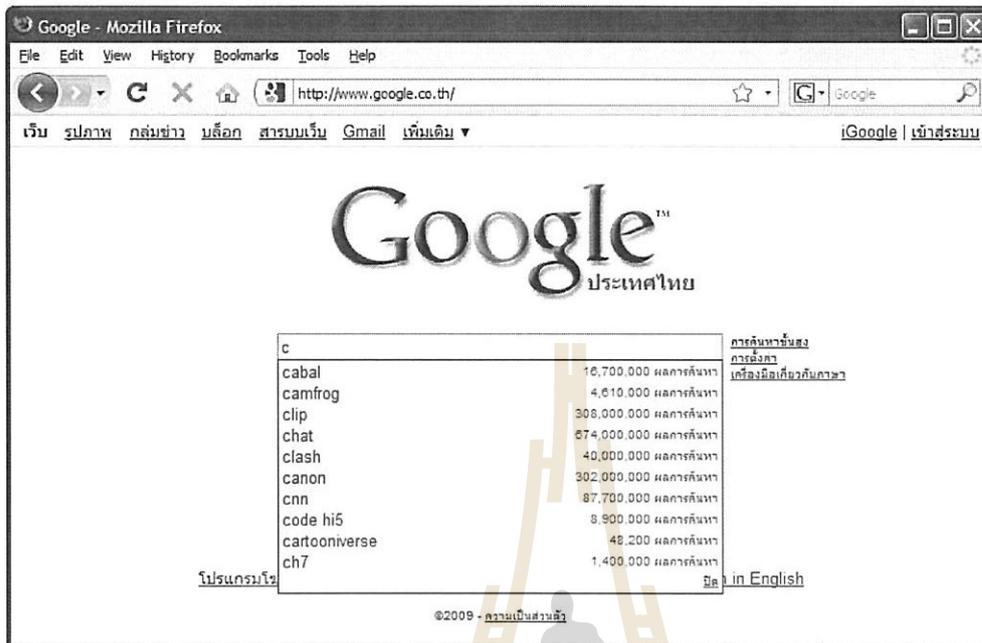
AJAX เป็นคำย่อมาจากคำว่า Asynchronous JavaScript And XML ซึ่งไม่ใช่ภาษาใหม่ในการเขียนโปรแกรม แต่เป็นเพียงเทคนิคใหม่ในการเขียนโปรแกรมเท่านั้น ที่ใช้อ็อบเจกต์ XMLHttpRequest ของจาวาสคริปต์ในการแลกเปลี่ยนข้อมูลบางส่วนกับเว็บเซิร์ฟเวอร์ที่จะนำมาแสดงบนหน้าเว็บ จึงโหลดเฉพาะข้อมูลบางส่วนเท่านั้น ไม่ต้องโหลดข้อมูลเก่าเข้ามาอีก ทำให้การแสดงผลเร็วขึ้น หน้าจอจะกระพริบเฉพาะในส่วนของข้อมูลที่ถูกโหลดเข้ามาใหม่เท่านั้น

เอแจ็กเป็นเทคนิคในการพัฒนาเว็บแอปพลิเคชันที่ถูกคิดค้นโดย Jesse James Garrett ในปี ค.ศ. 2005 โดยรวมเอาเทคนิคต่าง ๆ เข้าด้วยกัน ประกอบไปด้วย

- XHTML และ CSS สำหรับการแสดงผล
- DOM (Document Object Model) สำหรับการแสดงผลโต้ตอบแบบไดนามิก
- XML และ XSLT สำหรับการแลกเปลี่ยน แก้ไข และแสดงข้อมูล
- XMLHttpRequest สำหรับการติดต่อแบบ Asynchronous กับเซิร์ฟเวอร์
- JavaScript สำหรับรวมเทคโนโลยีที่กล่าวมาแล้วข้างต้นเข้าด้วยกัน

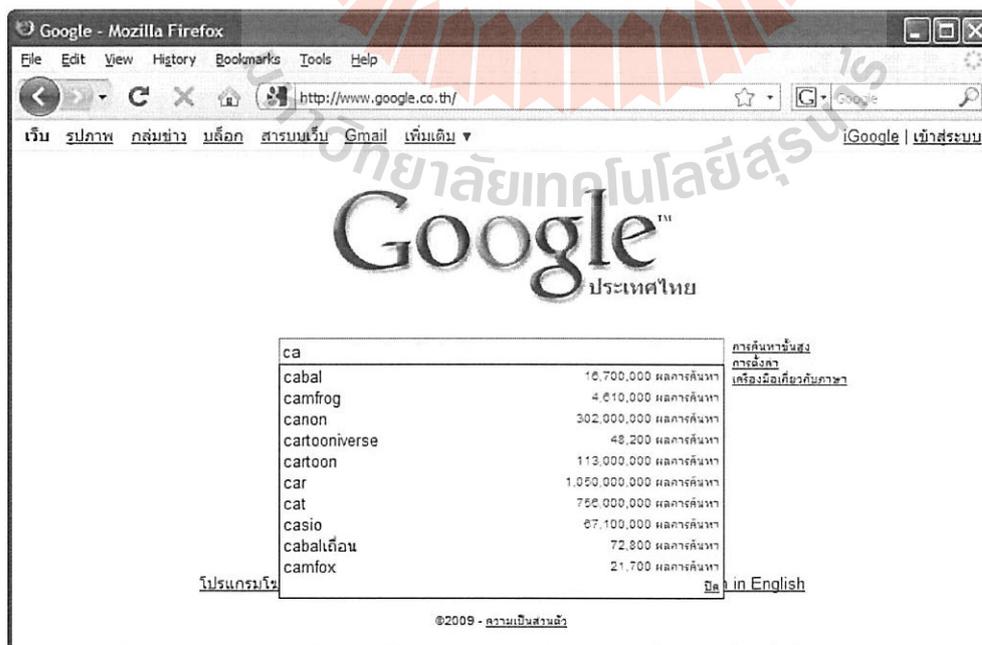
ตัวอย่าง การใช้เทคนิคเอแจ็กในการแนะนำคำที่ต้องการค้นหาใน Google ดังแสดงในรูปที่

11.1



รูปที่ 11.1: หน้าต่างกลุ่มของคำแนะนำเมื่อผู้ใช้ป้อนตัวอักษร c

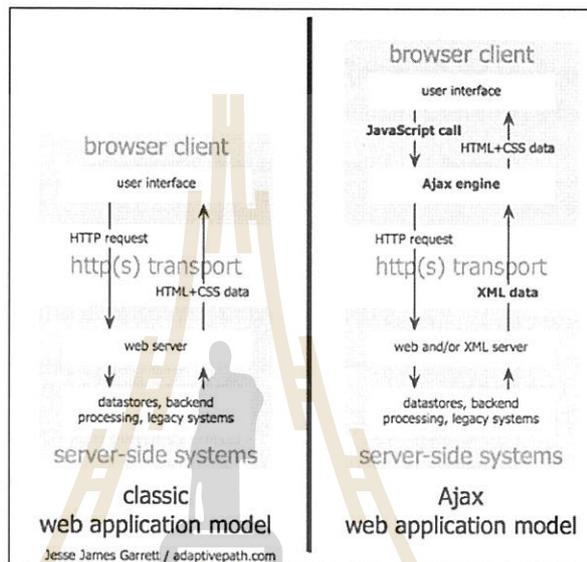
จากรูปที่ 11.1 นั้น ผู้ใช้เพียงป้อนตัวอักษร c เข้าไปในช่องค้นหา ก็จะปรากฏกลุ่มของคำแนะนำที่เริ่มต้นด้วยตัวอักษร c ขึ้นมาให้ ซึ่งทาง Google ได้ใช้เทคนิคเอแจ็กช่วยในการแสดงผล และเมื่อผู้ใช้ป้อนตัวอักษร d (ต่อจากตัวอักษร c) หน้าต่างกลุ่มของคำแนะนำก็เปลี่ยนไป โดยกลุ่มของคำแนะนำต่าง ๆ เหล่านี้จะเริ่มต้นด้วยตัวอักษร cd ดังแสดงในรูปที่ 11.2



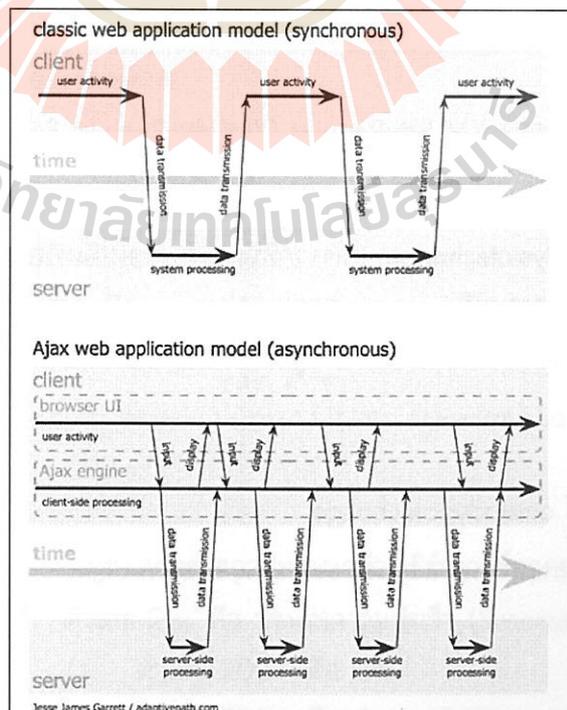
รูปที่ 11.2: หน้าต่างกลุ่มของคำแนะนำเมื่อผู้ใช้ป้อนตัวอักษร ca

จากรูปที่ 11.2 นั้น จะสังเกตเห็นว่าการใช้เทคนิคเอแจ็กเป็นส่วนเสริมในการออกแบบเว็บไซต์ ทำให้การแสดงผลในหน้าเว็บเร็วขึ้น เนื่องจากเนื้อหาทั้งหมดในหน้าต่างยังคงเหมือนเดิม แต่ส่วนที่เปลี่ยนไปก็มีเพียงเฉพาะส่วนที่แสดงกลุ่มของคำแนะนำเท่านั้น จึงไม่จำเป็นต้องโหลดข้อมูลทั้งหมดใหม่ แต่โหลดเฉพาะส่วนที่มีการเปลี่ยนแปลง

รูปที่ 11.3 แสดงถึงรูปแบบการเปรียบเทียบการส่งข้อมูลของเว็บแอปพลิเคชันแบบทั่ว ๆ ไปกับเว็บแอปพลิเคชันที่ใช้เทคโนโลยีเอแจ็ก ส่วนรูปที่ 11.4 จะแสดงถึงการติดต่อระหว่างเว็บเบราว์เซอร์ทางฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์แบบ Synchronous และ Asynchronous



รูปที่ 11.3: รูปแบบการเปรียบเทียบการส่งข้อมูลของเว็บแอปพลิเคชันแบบดั้งเดิมและแบบเอแจ็ก



รูปที่ 11.4: รูปแบบเปรียบเทียบการติดต่อระหว่างเว็บเบราว์เซอร์ทางฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์แบบ Synchronous และ Asynchronous

11.2 อ็อบเจกต์ XMLHttpRequest คืออะไร

อ็อบเจกต์ XMLHttpRequest มีตัวย่อว่า XHR คือกลุ่มของ API (Application Program Interface) ที่สามารถถูกเรียกมาใช้งานผ่านเว็บเบราว์เซอร์จากเครื่องผู้ใช้โดยสคริปต์ต่าง ๆ เช่น จาวาสคริปต์ วิบีสคริปต์ เป็นต้น เพื่อสร้างช่องทางในการติดต่อสื่อสารระหว่างหน้าเว็บทางฝั่งผู้ใช้งานกับทางฝั่งเซิร์ฟเวอร์แล้วถ่ายโอนข้อมูลที่เป็น XML, HTML หรือข้อมูลที่เป็นเท็กซ์ไปยังหรือมาจากเว็บเซิร์ฟเวอร์ซึ่งเป็นการทำงานในเบื้องหลัง โดยการใช้เกณฑ์วิธีถ่ายโอนข้อมูลไฮเปอร์เท็กซ์หรือที่เรียกว่า HTTP (HyperText Transfer Protocol)

11.2.1 เมตอดของ XMLHttpRequest (XMLHttpRequest Methods)

อ็อบเจกต์ XMLHttpRequest มีเมตอดที่สามารถเรียกใช้ได้ 6 เมตอด ดังแสดงในตารางที่ 11.1

ตารางที่ 11.1: เมตอดสำหรับอ็อบเจกต์ XMLHttpRequest

เมตอด	ความหมาย
abort ()	ยกเลิกการร้องขอ
getAllResponseHeaders ()	รับค่าส่วนหัวที่ร้องขอทั้งหมดเป็นคู่คือ คีย์และค่า (key/value pair)
getResponseHeader (header)	รับค่าส่วนหัวที่ร้องขอตามที่ระบุ
open (method, url, [isAsynchronous])	เปิดการติดต่อกับเว็บเซิร์ฟเวอร์ ซึ่งเมตอดที่ใช้อาจจะเป็น POST, GET หรือ PUT, URL นั้นสามารถเป็นแบบ Relative หรือ Absolute ก็ได้ ส่วนพารามิเตอร์ตัวสุดท้ายเป็นการเชื่อมต่อแบบ Asynchronous หรือไม่ (true หรือ false) ค่าปริยายคือ true
send (body)	ส่งการร้องขอไปยังเซิร์ฟเวอร์
setRequestHeader (header, value)	การกำหนดประเภทของข้อมูลที่จะมีการส่ง/รับ

11.2.2 คุณสมบัติของ XMLHttpRequest (XMLHttpRequest Properties)

อ็อบเจกต์ XMLHttpRequest มีคุณสมบัติต่าง ๆ (Properties) ดังต่อไปนี้

- **onreadystatechange** หลังจากที่มีการร้องขอไปที่เครื่องเซิร์ฟเวอร์แล้ว จำเป็นที่จะต้องมีการฟังที่ซ่อนอยู่ในคุณสมบัติ onreadystatechange เพื่อทำหน้าที่ในการจัดการกับข้อมูลที่ถูกส่งกลับมาจากเครื่องเซิร์ฟเวอร์ ซึ่งฟังก์ชันนี้จะถูกเรียกใช้อัตโนมัติเมื่อมีการเปลี่ยนแปลงสถานะของอ็อบเจกต์ XMLHttpRequest

ตัวอย่าง การใช้ onreadystatechange

```
xmlHttp.onreadystatechange=function(){
    // เขียนคำสั่งต่าง ๆ
}
```

- **readyState** เป็นค่าตัวเลขที่แสดงถึงสถานะการตอบกลับจากเครื่องเซิร์ฟเวอร์ของการร้องขอจากการใช้อ็อบเจกต์ XMLHttpRequest ซึ่งมีค่าต่าง ๆ ดังนี้

- 0 คือ การร้องขอยังไม่ถูกกำหนดขึ้น
- 1 คือ การร้องขอได้ถูกกำหนดขึ้น
- 2 คือ การร้องขอได้ถูกส่งไป
- 3 คือ การร้องขอกำลังทำงานอยู่
- 4 คือ การร้องขอได้เสร็จสิ้นการทำงาน

ตัวอย่าง การใช้ readyState

```
xmlHttp.onreadystatechange=function(){
    if(xmlHttp.readyState==4){
        // รับค่าที่ส่งกลับมาจากเซิร์ฟเวอร์
    }
}
```

- **responseText** เป็นข้อมูลแบบเท็กซ์ที่ถูกส่งกลับมาจากเครื่องเซิร์ฟเวอร์ ซึ่งเหมาะสมกับสถานการณ์ที่ต้องการค่าที่ส่งกลับจากเซิร์ฟเวอร์เพียงค่าเดียว

ตัวอย่าง การใช้ responseText

```
xmlHttp.onreadystatechange=function(){
    if(xmlHttp.readyState==4){
        alert(xmlHttp.responseText);
    }
}
```

- **responseXML** เป็นข้อมูลแบบ XML ที่ถูกส่งกลับมาจากเครื่องเซิร์ฟเวอร์ ซึ่งเหมาะสมกับสถานการณ์ที่ต้องการค่าที่ส่งกลับจากเซิร์ฟเวอร์หลายค่า

ตัวอย่าง การใช้ responseXML

```
xmlHttp.onreadystatechange=function(){
    if(xmlHttp.readyState==4){
        var response =
xmlHttp.responseXML.documentElement;
        var n =
response.getElementsByTagName('name')[0].firstChild.nodeValue;
        var e =
response.getElementsByTagName('email')[0].firstChild.nodeValue;
    }
}
```

- **status** เป็นข้อมูลแสดงค่าสถานะของเครื่องเซิร์ฟเวอร์ออกเป็นตัวเลข เช่น

- ตัวเลข 2xx คือ ประสบความสำเร็จ (Success)
 - ตัวเลข 200 คือ OK
 - ตัวเลข 201 คือ Created

- ตัวเลข 202 คือ Accepted
- ตัวเลข 203 คือ Source unknown
- ตัวเลข 204 คือ No new content
- ตัวเลข 205 คือ Reset content
- ตัวเลข 206 คือ Request only partially filled
- ตัวเลข 3xx คือ การเปลี่ยนทิศข้อมูล (Redirection)
 - ตัวเลข 300 คือ Multiple Choices
 - ตัวเลข 301 คือ Moved
 - ตัวเลข 302 คือ Found
 - ตัวเลข 303 คือ Method
 - ตัวเลข 304 คือ Not Modified
- ตัวเลข 4xx คือ ข้อผิดพลาดทางฝั่งผู้ใช้งาน (Client Error)
 - ตัวเลข 400 คือ Bad Request
 - ตัวเลข 401 คือ Unauthorized
 - ตัวเลข 402 คือ Payment Required
 - ตัวเลข 403 คือ Forbidden
 - ตัวเลข 404 คือ Not Found
- ตัวเลข 5xx คือ ข้อผิดพลาดทางฝั่งเซิร์ฟเวอร์ (Server Error)
 - ตัวเลข 500 คือ Internal Error
 - ตัวเลข 501 คือ Not Implemented
 - ตัวเลข 502 คือ Service Temporarily Overloaded
 - ตัวเลข 503 คือ Gateway Timeout

ตัวอย่าง การใช้ status

```
xmlHttpRequest.onreadystatechange=function(){
    if(xmlHttpRequest.readyState==4){
        if (xmlHttpRequest.status == 200){
            alert(xmlHttpRequest.responseText);
        } else{
            alert('There was a problem with
the request.');
```

- **statusText** แสดงค่าสถานะของเครื่องเซิร์ฟเวอร์ออกเป็นข้อความ เช่น Not Found แทนตัวเลขคือ 404

11.3 การเริ่มต้นใช้งาน XMLHttpRequest

การที่จะเรียกใช้ XMLHttpRequest ได้นั้น เราต้องกำหนด instance ของคลาสที่เว็บเบราว์เซอร์สามารถเรียกใช้งานได้ ซึ่งคลาสที่ใช้ผ่านเว็บเบราว์เซอร์ Internet Explorer เรียกว่า ActiveXObject ส่วนเว็บเบราว์เซอร์ตัวอื่น ๆ เช่น Firefox หรือ Safari เรียกว่า XMLHttpRequest ดังนั้นนักพัฒนาเว็บไซต์จึงจำเป็นต้องตรวจสอบว่า ผู้ใช้ที่ติดต่อเข้ามาที่เครื่องเซิร์ฟเวอร์นั้นใช้เว็บเบราว์เซอร์ตัวไหน จะได้จัดการเรื่องคลาสได้ถูกต้อง ซึ่งในปัจจุบันนี้โปรแกรมเว็บเบราว์เซอร์ส่วนใหญ่ที่นิยมใช้กันมากที่สุดก็คือ Internet Explorer ลำดับต่อมาก็คือ Firefox ซึ่งจะขอแนะนำตัวอย่างในการกำหนด instance ของคลาสสำหรับโปรแกรมเว็บเบราว์เซอร์ทั้งสอง ดังต่อไปนี้

ตัวอย่าง การกำหนด instance ของคลาสเพื่อใช้งานใน Internet Explorer

```

1. <html><head><title>Create XMLHttpRequest for IE</title></head>
2. <body>
3. <script language="javascript" type="text/javascript">
4.     var xmlhttp = null;
5.     if(window.ActiveXObject){
6.         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
7.         alert('You are using Internet Explorer. ');
8.     }else{
9.         alert('Giving up :( Cannot create an XMLHTTP instance');
10.    }
11. </script>
12. </body></html>

```

รูปที่ 11.5: การกำหนด instance ของคลาสสำหรับเว็บเบราว์เซอร์ Internet Explorer ไฟล์ create_ie.html

บรรทัดที่ 1 เปิดแท็ก <html>, <head>, <title> แล้วแสดงข้อความ “Create XMLHttpRequest for IE” ในส่วนของ title และปิดแท็ก </title> และ </head>

บรรทัดที่ 2 เปิดแท็ก <body>

บรรทัดที่ 3 เริ่มต้นการเขียนคำสั่งจาวาสคริปต์

บรรทัดที่ 4 กำหนดตัวแปร xmlhttp ให้มีค่าเท่ากับ null

บรรทัดที่ 5 ตรวจสอบว่าถ้าหน้าต่างเว็บเบราว์เซอร์ ที่เปิดใช้งานอยู่นี้เป็นคลาสแบบ ActiveXObject (นั่นหมายถึงหน้าต่างที่เปิดอยู่คือ Internet Explorer) ถ้าจริงก็จะไปทำคำสั่งในบรรทัดที่ 6-7 มิฉะนั้นแล้วก็จะกระโดดไปทำคำสั่งในบรรทัดที่ 9 แทน

บรรทัดที่ 6 สร้างคลาสใหม่ที่เป็น ActiveXObject ให้กับตัวแปร xmlhttp

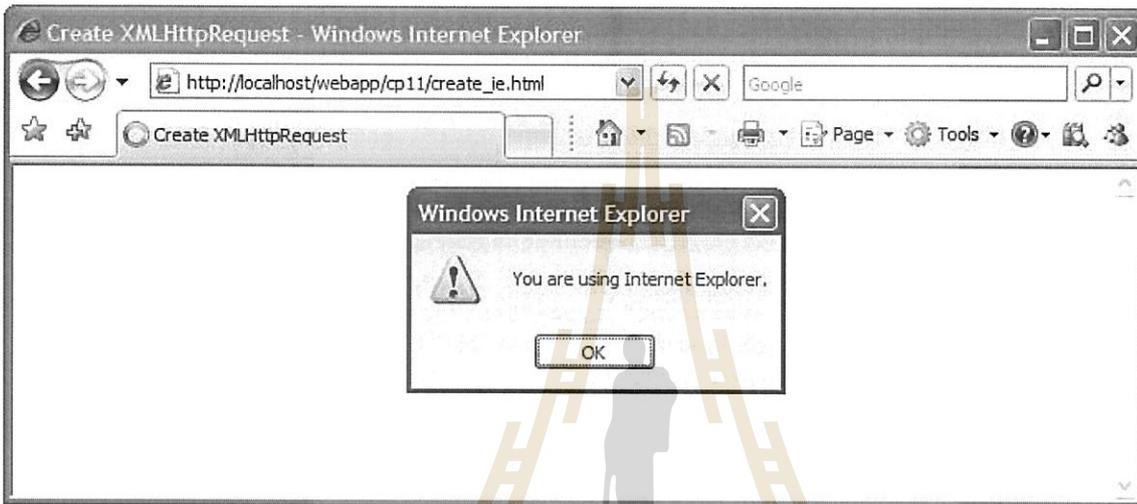
บรรทัดที่ 7 เปิดหน้าต่างเว็บเบราว์เซอร์ขึ้นใหม่และแสดงข้อความ “You are using Internet Explorer.”

บรรทัดที่ 8 จบการทำงานของคำสั่ง if แล้วเข้าสู่การทำงานของ else

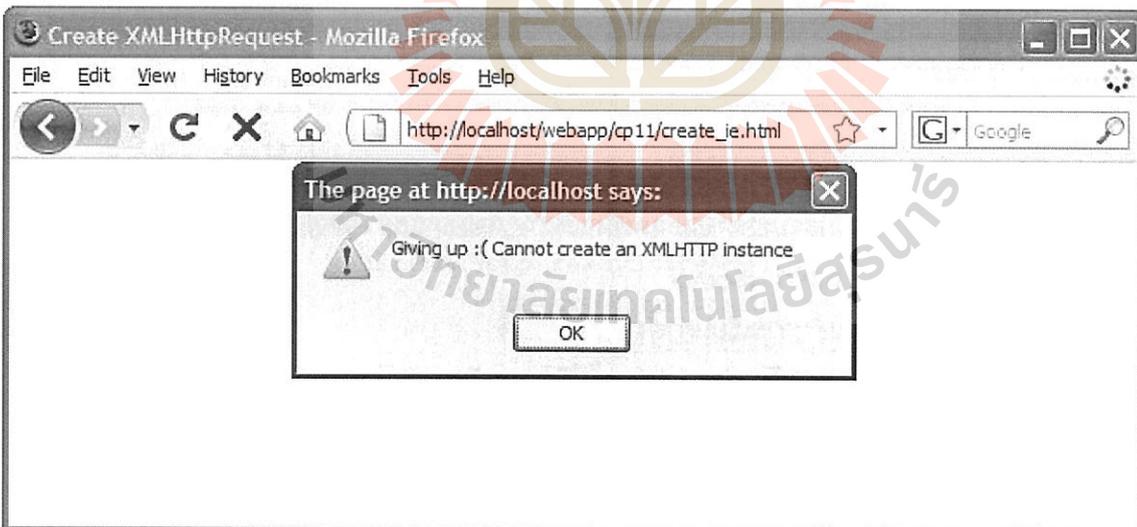
บรรทัดที่ 9 เปิดหน้าต่างเว็บเบราว์เซอร์ขึ้นใหม่และแสดงข้อความ “Giving up :(Cannot create an XMLHTTP instance”

- บรรทัดที่ 10 จบการทำงานของคำสั่ง else
- บรรทัดที่ 11 จบการทำงานของจาวาสคริปต์
- บรรทัดที่ 12 ปิดแท็ก </body> และ </html>

จากคำสั่งดังแสดงในรูปที่ 11.5 นั้น เมื่อเปิดไฟล์ create_ie.html ด้วยเว็บเบราว์เซอร์ Internet Explorer จะปรากฏหน้าต่างดังแสดงในรูปที่ 11.6 แต่ถ้าเปิดด้วยเว็บเบราว์เซอร์ Firefox จะปรากฏหน้าต่างดังแสดงในรูปที่ 11.7



รูปที่ 11.6: การแสดงผลของไฟล์ create_ie.html ในรูปที่ 11.5 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Internet Explorer



รูปที่ 11.7: การแสดงผลของไฟล์ create_ie.html ในรูปที่ 11.5 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Firefox

ตัวอย่าง การกำหนด instance ของคลาสเพื่อใช้งานใน Firefox

```

1. <html><head><title>Create XMLHttpRequest for None-IE</title></head>
2. <body>
3. <script language="javascript" type="text/javascript">
4.     var xmlhttp = null;
5.     if(window.XMLHttpRequest){
6.         xmlhttp = new XMLHttpRequest();
7.         alert('Welcome to Ajax World !!!');
8.     }else{
9.         alert('Giving up :( Cannot create an XMLHTTP instance');
10.    }
11. </script>
12. </body></html>

```

รูปที่ 11.8: การกำหนด instance ของคลาสสำหรับเว็บเบราว์เซอร์ Firefox ในไฟล์ create_firefox.html

บรรทัดที่ 1 เปิดแท็ก <html>, <head>, <title> แล้วแสดงข้อความ “Create XMLHttpRequest for None-IE” ในส่วนของ title และปิดแท็ก </title> และ </head>

บรรทัดที่ 2 เปิดแท็ก <body>

บรรทัดที่ 3 เริ่มต้นการเขียนคำสั่งจาวาสคริปต์

บรรทัดที่ 4 กำหนดตัวแปร xmlhttp ให้มีค่าเท่ากับ null

บรรทัดที่ 5 ตรวจสอบว่าถ้าหน้าต่างเว็บเบราว์เซอร์ ที่เปิดใช้งานอยู่นี้เป็นคลาสแบบ XMLHttpRequest (นั่นหมายถึงหน้าต่างที่เปิดอยู่ไม่ใช่ Internet Explorer) ถ้าจริงก็จะไปทำคำสั่งในบรรทัดที่ 6-7 มิฉะนั้นแล้วก็จะกระโดดไปทำคำสั่งในบรรทัดที่ 9 แทน

บรรทัดที่ 6 สร้างคลาสใหม่ที่เป็น XMLHttpRequest ให้กับตัวแปร xmlhttp

บรรทัดที่ 7 เปิดหน้าต่างเว็บเบราว์เซอร์ขึ้นใหม่และแสดงข้อความ “Welcome to Ajax World !!!”

บรรทัดที่ 8 จบการทำงานของคำสั่ง if แล้วเข้าสู่การทำงานของ else

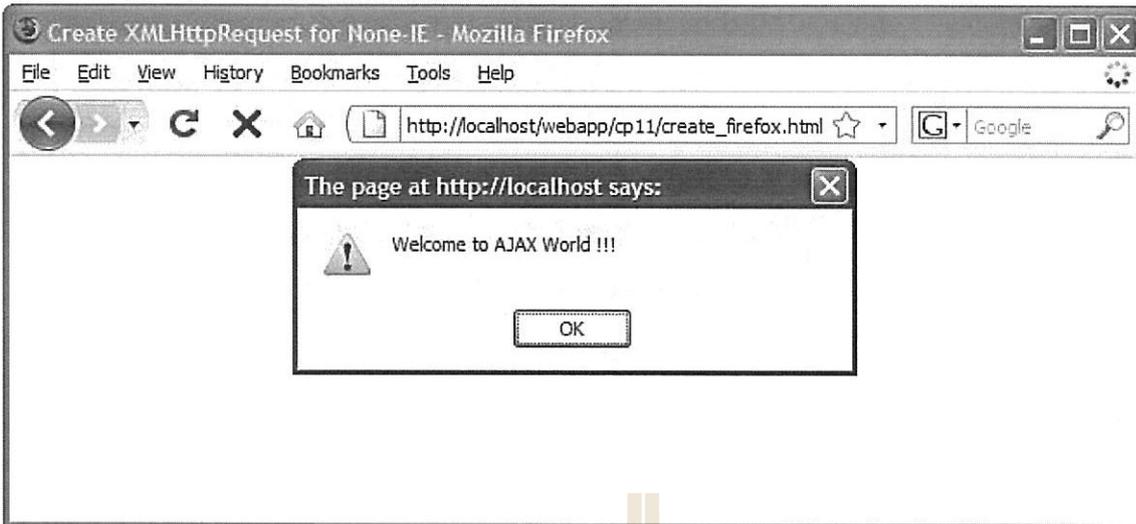
บรรทัดที่ 9 เปิดหน้าต่างเว็บเบราว์เซอร์ขึ้นใหม่และแสดงข้อความ “Giving up :(Cannot create an XMLHTTP instance”

บรรทัดที่ 10 จบการทำงานของคำสั่ง else

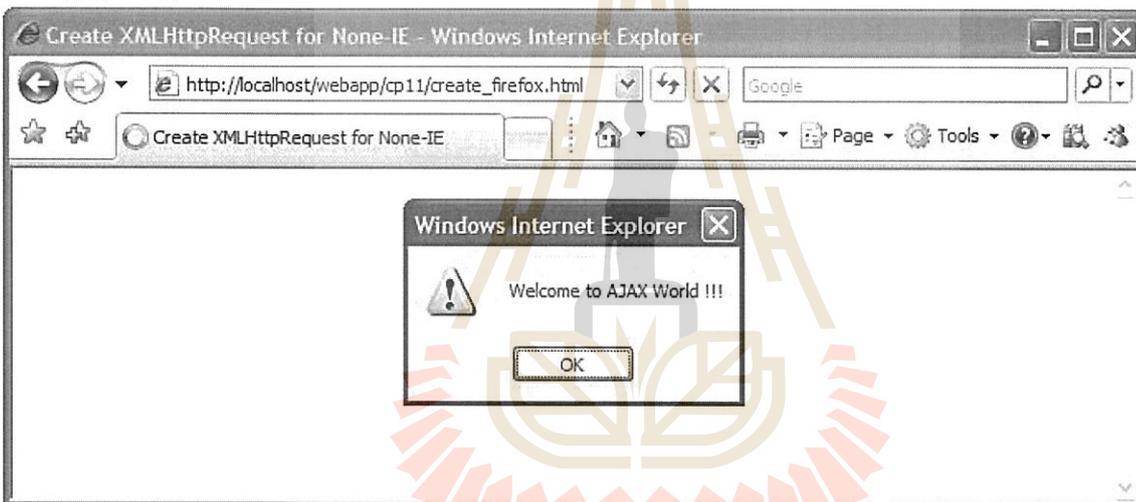
บรรทัดที่ 11 จบการทำงานของจาวาสคริปต์

บรรทัดที่ 12 ปิดแท็ก </body> และ </html>

จากคำสั่งดังแสดงในรูปที่ 11.8 นั้น เมื่อเปิดไฟล์ create_firefox.html ด้วยเว็บเบราว์เซอร์ Firefox จะปรากฏหน้าต่างดังแสดงในรูปที่ 11.9 แต่ถ้าเปิดด้วยเว็บเบราว์เซอร์ Internet Explorer จะปรากฏหน้าต่างดังแสดงในรูปที่ 11.10



รูปที่ 11.9: การแสดงผลของไฟล์ create_firefox.html ในรูปที่ 11.8 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Firefox



รูปที่ 11.10: การแสดงผลของไฟล์ create_firefox.html ในรูปที่ 11.8 เมื่อเปิดด้วยโปรแกรมเว็บเบราว์เซอร์ Internet Explorer

จะสังเกตเห็นว่าเมื่อเปิดไฟล์ create_firefox.html ด้วยโปรแกรมเว็บเบราว์เซอร์ Firefox หรือ Internet Explorer ต่างก็แสดงผลเหมือนกัน เนื่องจากโปรแกรมเว็บเบราว์เซอร์ Internet Explorer ที่ใช้เป็นเวอร์ชัน 7 นั้นหมายถึงว่า คลาส XMLHttpRequest ได้ถูกกำหนดให้ใช้ได้ ใน IE 7 นั้นเอง แต่อย่างไรก็ดี ผู้พัฒนาระบบก็ควรตรวจสอบการกำหนด instance ของคลาสที่สามารถเรียกใช้งานได้ก่อน ทั้งนี้ก็เพื่อป้องกันข้อผิดพลาด เนื่องจากโปรแกรมเว็บเบราว์เซอร์มีหลายโปรแกรมและหลายเวอร์ชันด้วยกันที่ยังคงแพร่หลายและใช้กันอยู่ในปัจจุบัน ดังนั้นไฟล์ create_ie.html และ create_firefox.html สามารถเขียนรวมกันเพื่อตรวจสอบการกำหนด instance ของคลาสได้ดังแสดงในรูปที่ 11.11

```

1. <html><head><title>Create XMLHttpRequest</title></head>
2. <body>
3. <script language="javascript" type="text/javascript">
4.     var xmlhttp = null;
5.     if(window.XMLHttpRequest){
6.         xmlhttp = new XMLHttpRequest();
7.         alert('Welcome to AJAX World !!!');
8.     }else if(window.ActiveXObject){
9.         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
10.        alert('You are using Internet Explorer. ');
11.    }else{
12.        alert('Giving up :( Cannot create an XMLHTTP instance');
13.    }
14. </script>
15. </body></html>

```

รูปที่ 11.11: การกำหนด instance ของคลาสสำหรับเว็บเบราว์เซอร์ IE7, Firefox, Safari, Opera ในไฟล์ create.html

เมื่อรู้และเข้าใจถึงขั้นตอนของการกำหนด instance ของคลาส XMLHttpRequest แล้ว ในลำดับต่อไปนี้จะเป็นตัวอย่งการเขียนโปรแกรมแสดงแบบฟอร์ม ซึ่งไฟล์ converter_form.html (รูปที่ 11.12) จะแสดงแบบฟอร์มเพื่อรับข้อมูลเป็นเท็กซ์และเพื่อแสดงผลข้อมูลออกเป็นเท็กซ์เช่นเดียวกัน เมื่อป้อนข้อมูลเข้ามาที่ช่องรับข้อมูลแล้ว โปรแกรมจะใช้เทคโนโลยีเอ็จกซ์ช่วยในการแสดงผลคือเมื่อผู้ใ้กดปุ่มป้อนข้อมูลที่แป้นพิมพ์แล้วปล่อยปุ่มนั้น โปรแกรมจะทำการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ แล้วส่งข้อมูลตัวนั้นไปที่เว็บเซิร์ฟเวอร์เพื่อทำการประมวลผลในพื้นหลัง (Background) ถ้าข้อมูลตัวนั้นเป็นตัวพิมพ์เล็ก (a-z) จะทำการเปลี่ยนให้เป็นตัวพิมพ์ใหญ่ (A-Z), ถ้าข้อมูลตัวนั้นเป็นตัวพิมพ์ใหญ่ (A-Z) จะทำการเปลี่ยนให้เป็นตัวพิมพ์เล็ก (a-z) หรือถ้าเป็นตัวอักษรอื่น ๆ ก็ทำการเปลี่ยนให้เป็นเครื่องหมายบวก (+) แล้วส่งกลับมาแสดงผลในช่องแสดงผลที่เครื่องผู้ใ้



บรรทัดที่ 8 ตรวจสอบหน้าต่างเว็บเบราว์เซอร์ที่เปิดใช้งานอยู่นี้เป็นคลาสแบบ XMLHttpRequest (นั่นหมายถึงหน้าต่างที่เปิดอยู่เป็น Internet Explorer) ถ้าจริงก็จะไปทำคำสั่งในบรรทัดที่ 9 มิฉะนั้นแล้วก็จะกระโดดไปทำคำสั่งในบรรทัดที่ 10 แทน

บรรทัดที่ 9 สร้างคลาสใหม่ที่เป็น XMLHttpRequest ให้กับตัวแปร xmlhttp

บรรทัดที่ 10 ตรวจสอบหน้าต่างเว็บเบราว์เซอร์ที่เปิดใช้งานอยู่นี้เป็นคลาสแบบ XMLHttpRequest (นั่นหมายถึงหน้าต่างที่เปิดอยู่ไม่ใช่ Internet Explorer) ถ้าจริงก็จะไปทำคำสั่งในบรรทัดที่ 11 มิฉะนั้นแล้วก็จะกระโดดไปทำคำสั่งในบรรทัดที่ 12 แทน

บรรทัดที่ 11 สร้างคลาสใหม่ที่เป็น XMLHttpRequest ให้กับตัวแปร xmlhttp

บรรทัดที่ 12 เริ่มทำคำสั่ง else

บรรทัดที่ 13 เปิดหน้าต่างเว็บเบราว์เซอร์ขึ้นใหม่และแสดงข้อความ “Your browser does not support AJAX.”

บรรทัดที่ 14 เป็นข้อสั่งเกต การเริ่มต้นเข้าสู่ฟังก์ชัน processData()

บรรทัดที่ 15 เริ่มต้นเข้าสู่ฟังก์ชัน processData()

บรรทัดที่ 16 ตรวจสอบว่าค่าของตัวแปร xmlhttp ไม่เท่ากับ null จริงหรือไม่? ถ้าจริงก็จะเข้าไปทำคำสั่งในบรรทัดที่ 17 – 24 มิฉะนั้นแล้วจะไปทำคำสั่งในบรรทัดที่ 25 แทน

บรรทัดที่ 17 เรียกใช้เมทอด open เพื่อเปิดการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ซึ่งใช้เมทอดการส่งข้อมูลแบบ GET ไปที่ไฟล์ convert.php โดยส่งตัวแปรที่ชื่อว่า input ซึ่งมีค่าเท่ากับข้อมูลที่ผู้ใช้ป้อนเข้ามาในแบบฟอร์มนี้ที่มีไอดีคือ inputID และเป็นการเชื่อมต่อแบบ Asynchronous

บรรทัดที่ 18 เรียกใช้เมทอด send เพื่อส่งข้อมูลเป็น null ไปที่เว็บเซิร์ฟเวอร์ (ซึ่งในที่นี้ก็คือไม่ได้ส่งอะไรไปเลย ทั้งนี้เนื่องจากการส่งข้อมูลแบบ GET ไปแล้วในคำสั่งบรรทัดที่ 17 แต่คำสั่งในบรรทัดนี้จำเป็นต้องมีอยู่)

บรรทัดที่ 19 เรียกใช้คุณสมบัติ onreadystatechange โดยกำหนดให้มีค่าเท่ากับการทำงานของฟังก์ชันหนึ่ง ซึ่งจะทำหน้าที่จัดการกับข้อมูลที่ถูกส่งกลับมาจากเครื่องเซิร์ฟเวอร์ ฟังก์ชันนี้จะถูกเรียกใช้โดยอัตโนมัติเมื่อมีการเปลี่ยนสถานะของอ็อบเจกต์ XMLHttpRequest

บรรทัดที่ 20 เรียกใช้คุณสมบัติ readyState เพื่อตรวจสอบดูว่ามีค่าเท่ากับ 4 หรือไม่ (ค่า 4 หมายถึงการร้องขอได้เสร็จสิ้นการทำงาน)

บรรทัดที่ 21 ข้อมูลที่เว็บเซิร์ฟเวอร์ส่งกลับมาให้เป็นเท็กซ์นั้น ให้นำไปแสดงในหน้าเว็บนี้ตรงตำแหน่งที่มีไอดีคือ outputID

บรรทัดที่ 22 จบการทำงานของ if ในบรรทัดที่ 20

บรรทัดที่ 23 จบการทำงานของฟังก์ชันในบรรทัดที่ 19

บรรทัดที่ 24 จบการทำงานของ if ในบรรทัดที่ 16

บรรทัดที่ 25 จบการทำงานของฟังก์ชัน processData()

บรรทัดที่ 26 จบการทำงานของจาวาสคริปต์

บรรทัดที่ 27 เปิดแท็ก <center> แสดงข้อความ “Lowercase-to-Uppercase Converter” พร้อมกับขึ้นบรรทัดใหม่ 2 บรรทัด

บรรทัดที่ 28 เปิดแท็ก <form> เพื่อสร้างกล่องข้อความ 2 กล่อง โดยมีลักษณะดังนี้

กล่องข้อความที่ 1 ให้แสดงข้อความ “ป้อนข้อมูล :” แล้วตามด้วยกล่องข้อความชนิดเท็กซ์มีชื่อว่า input มีไอดีคือ inputID และมีการเรียกใช้เหตุการณ์ (event) ที่เกิดจากการกดปุ่มแล้วปล่อยปุ่ม (onkeyup) ให้ทำการเรียกใช้ฟังก์ชัน processData()

ใช้คำสั่ง จำนวน 6 ตัว เพื่อให้เกิดช่องว่าง 6 ช่อง

กล่องข้อความที่ 2 ให้แสดงข้อความ “แสดงผล :” แล้วตามด้วยกล่องข้อความชนิดเท็กซ์มีชื่อว่า output มีไอดีคือ outputID

บรรทัดที่ 29 ปิดแท็ก </form>, </center>, </body> และ </html>

ส่วนคำสั่งต่าง ๆ ในไฟล์ convert.php นั้น มีดังแสดงในรูปที่ 11.13

```

1.  <?php
2.      if (isset($_GET['input'])) {
3.          $txt = trim($_GET['input']);
4.          $len = strlen($txt);
5.          for($i=0; $i<$len; $i++){
6.              if ctype_lower($txt[$i])
7.                  $txt[$i]=strtoupper($txt[$i]);
8.              else if(ctype_upper($txt[$i])
9.                  $txt[$i]=strtolower($txt[$i]);
10.             else
11.                 $txt[$i]='+';
12.             }
13.             echo $txt;
14.         }
15.     ?>

```

รูปที่ 11.13: คำสั่งในไฟล์ convert.php

บรรทัดที่ 1 เริ่มการเขียนคำสั่ง PHP

บรรทัดที่ 2 ตรวจสอบว่ามีตัวแปรที่รับเข้ามาแบบ GET ชื่อว่า input หรือไม่ ถ้ามีก็จะเข้าไปทำคำสั่งในบรรทัดที่ 3-14 มิฉะนั้นแล้วก็จะไปทำคำสั่งในบรรทัดที่ 15 แทน

บรรทัดที่ 3 ใช้ฟังก์ชัน trim() เพื่อทำการตัดช่องว่างที่อาจจะมีก่อนและหลังข้อมูลที่อยู่ในตัวแปร input แล้วเก็บไว้ที่ตัวแปร \$txt

บรรทัดที่ 4 ใช้ฟังก์ชัน strlen() เพื่อทำการหาความยาวของข้อมูลที่อยู่ในตัวแปร \$txt แล้วเก็บไว้ที่ตัวแปร \$len

บรรทัดที่ 5 ใช้ for หลูปในการวนรอบเป็นจำนวนเท่ากับค่าที่อยู่ในตัวแปร \$len

บรรทัดที่ 6 ตรวจสอบข้อมูลที่อยู่ในตัวแปร \$txt[\$i] ถ้าเป็นตัวพิมพ์เล็กก็จะไปทำคำสั่งในบรรทัดที่ 7 มิฉะนั้นแล้วจะไปทำคำสั่งในบรรทัดที่ 8 แทน

บรรทัดที่ 7 เปลี่ยนข้อมูลที่อยู่ในตัวแปร \$txt[\$i] ให้เป็นตัวพิมพ์ใหญ่ แล้วนำไปเก็บไว้ที่ตัวแปรเดิมคือตัวแปร \$txt[\$i]

บรรทัดที่ 8 ตรวจสอบข้อมูลที่อยู่ในตัวแปร \$txt[\$i] ถ้าเป็นตัวพิมพ์ใหญ่ก็จะไปทำคำสั่งในบรรทัดที่ 9 มิฉะนั้นแล้วจะไปทำคำสั่งในบรรทัดที่ 10 แทน

บรรทัดที่ 9 เปลี่ยนข้อมูลที่อยู่ในตัวแปร `$txt[$i]` ให้เป็นตัวพิมพ์เล็ก แล้วนำไปเก็บไว้ที่ตัวแปรเดิมคือตัวแปร `$txt[$i]`

บรรทัดที่ 10 เริ่มการทำงานของคำสั่ง `else`

บรรทัดที่ 11 เปลี่ยนข้อมูลที่อยู่ในตัวแปร `$txt[$i]` ให้เป็นเครื่องหมายบวก (+)

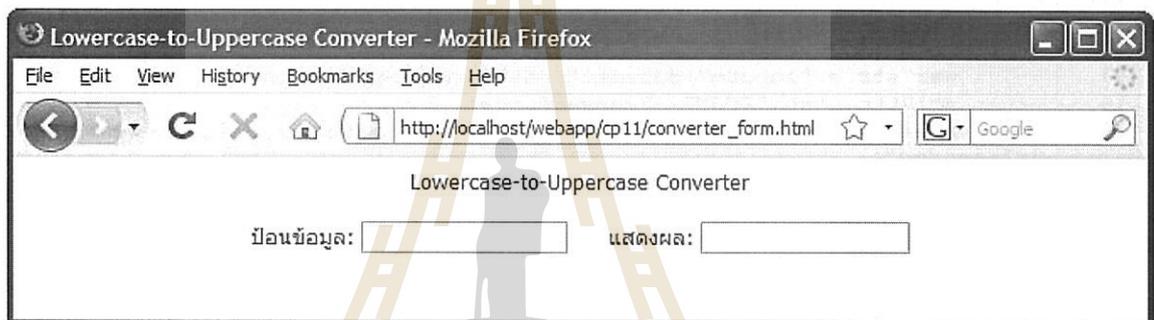
บรรทัดที่ 12 จบการทำงานของ `for` หลุด

บรรทัดที่ 13 แสดงข้อมูลที่อยู่ในตัวแปร `$txt` ออกที่จอภาพ

บรรทัดที่ 14 จบการทำงานของ `if`

บรรทัดที่ 15 จบการทำงานของคำสั่ง PHP

เมื่อเปิดไฟล์ `convert_form.html` ด้วยโปรแกรมเว็บเบราว์เซอร์แล้ว จะปรากฏดังแสดงในรูปที่ 11.14 และเมื่อป้อนข้อมูลลงในแบบฟอร์มในช่อง “ป้อนข้อมูล :” แล้วจะปรากฏการเปลี่ยนแปลงขึ้นในช่อง “แสดงผล :” ดังแสดงในรูปที่ 11.15



รูปที่ 11.14: การแสดงผลของไฟล์ `convert_form.html`



รูปที่ 11.15: การแสดงผลจากแบบฟอร์มของไฟล์ `convert_form.html` เมื่อป้อนข้อมูล

จากรูปที่ 11.15 นั้นจะสังเกตเห็นว่าเมื่อกดปุ่มตัวอักษร `a` ที่แป้นพิมพ์เพื่อป้อนข้อมูลเข้ามา 1 ตัว แล้วปล่อยปุ่มดังกล่าว ข้อมูลตัวอักษร `a` นั้นก็จะถูกส่งไปที่เครื่องเซิร์ฟเวอร์เพื่อทำการเปลี่ยนข้อมูลนั้นให้เป็นตัวพิมพ์ใหญ่แล้วนำมาแสดงในช่องแสดงผล เมื่อป้อนข้อมูลตัวต่อไปในแต่ละครั้งที่ปล่อยปุ่มก็จะทำการส่งข้อมูลเหล่านั้นไปที่เครื่องเซิร์ฟเวอร์เพื่อทำการเปลี่ยนข้อมูลนั้น แล้วนำมาแสดงในช่องแสดงผลโดยอัตโนมัติ

บรรทัดที่ 20 กำหนดรูปแบบการร้องขอของส่วนหัวที่จะส่งไปที่เว็บเซิร์ฟเวอร์ โดยกำหนดให้ความยาวของข้อมูลเป็นความยาวของค่าที่ถูกเก็บไว้ที่ตัวแปร `str` ซึ่งก็คือข้อมูลที่ป้อนลงในช่องป้อนข้อมูลนั่นเอง

บรรทัดที่ 21 กำหนดรูปแบบการร้องขอของส่วนหัวที่จะส่งไปที่เว็บเซิร์ฟเวอร์ โดยให้ทำการปิดการร้องขอของส่วนหัว

บรรทัดที่ 27 เรียกใช้เมทอด `send` เพื่อส่งข้อมูลที่ถูกเก็บไว้ที่ตัวแปร `str` ไปที่เว็บเซิร์ฟเวอร์ ส่วนไฟล์ `convert_post.php` มีคำสั่งดังแสดงในรูปที่ 11.17

```

1. <?php
2.     if (isset($_POST['input'])) {
3.         $txt = trim($_POST['input']);
4.         $len = strlen($txt);
5.         for($i=0; $i<$len; $i++){
6.             if ctype_lower($txt[$i])
7.                 $txt[$i]=strtoupper($txt[$i]);
8.             else if(ctype_upper($txt[$i]))
9.                 $txt[$i]=strtolower($txt[$i]);
10.            else
11.                $txt[$i]='+';
12.        }
13.        echo $txt;
14.    }
15. ?>

```

รูปที่ 11.17: คำสั่งในไฟล์ `convert_post.php` ที่มีการส่งข้อมูลแบบ POST

11.4 การใช้งาน XMLHttpRequest กับ XML

การใช้งาน XMLHttpRequest ในการเชื่อมต่อกับเว็บเซิร์ฟเวอร์โดยใช้เทคโนโลยีเอแจ็กที่ผ่านมานั้น ได้ทำการรับข้อมูลจากแบบฟอร์มแล้วส่งข้อมูลนั้นไปยังเว็บเซิร์ฟเวอร์เพื่อทำการประมวลผล แล้วส่งกลับไปแสดงผลที่เครื่องผู้ใช้งาน ซึ่งในบางครั้งอาจมีความจำเป็นที่จะต้องทำการดึงข้อมูลจากแฟ้มข้อมูลมาทำการเปรียบเทียบหรือประมวลผล แฟ้มข้อมูลชนิดหนึ่งที่ถูกออกแบบมาเพื่อใช้ในการแลกเปลี่ยนหรือถ่ายโอนข้อมูลและมีโครงสร้างและรูปแบบตามมาตรฐานที่ W3C กำหนดไว้คือ แฟ้มข้อมูล XML

XML (ย่อมาจาก Extensive Markup Language) เป็นภาษาหนึ่งที่ใช้ในการเก็บข้อมูลที่ให้ความชัดเจนในรายละเอียดเกี่ยวกับข้อมูลต่าง ๆ ซึ่งมีการใช้แท็กเปิดและแท็กปิดคล้ายกับภาษา HTML แต่มีความยืดหยุ่นและสะดวกมากกว่าภาษา HTML เนื่องจากภาษา XML ไม่ได้กำหนดชื่อแท็กไว้ก่อน แต่ปล่อยให้ผู้ใช้เป็นคนกำหนดชื่อแท็กขึ้นมาใช้งานได้เองตามความต้องการ โดยในส่วนนี้จะขอยกตัวอย่างในการใช้ XMLHttpRequest กับเทคโนโลยีเอแจ็กในการจัดการข้อมูลที่เป็น XML ชื่อว่า `note.xml` ดังแสดงในรูปที่ 11.18

```

1.  <?xml version="1.0" encoding="TIS-620" ?>
2.  <NOTE>
3.      <PERSON>
4.          <NAME>Kacha Chansilp</NAME>
5.          <MESSAGE>Don't Worry, Be Happy</MESSAGE>
6.          <EMAIL>kacha@sut.ac.th</EMAIL>
7.      </PERSON>
8.      <PERSON>
9.          <NAME>Somsak Jaidee</NAME>
10.         <MESSAGE>Forget Me Not</MESSAGE>
11.         <EMAIL>somsak@yahoo.com</EMAIL>
12.     </PERSON>
13.     <PERSON>
14.         <NAME>Somsri Boonmak</NAME>
15.         <MESSAGE>Love me, Love my dog !!!</MESSAGE>
16.         <EMAIL>somsri@gmail.com</EMAIL>
17.     </PERSON>
18.     <PERSON>
19.         <NAME>Yoodee Meesuk</NAME>
20.         <MESSAGE>So Far, So Good</MESSAGE>
21.         <EMAIL>yoodee_meesuk@yahoo.com</EMAIL>
22.     </PERSON>
23. </NOTE>

```

รูปที่ 11.18: ข้อมูลในไฟล์ note.xml

ในลำดับต่อไปเป็นไฟล์จาวาสคริปต์ที่ใช้สร้างอ็อบเจกต์ XMLHttpRequest เพื่อเชื่อมต่อกับเว็บเซิร์ฟเวอร์ชื่อว่า note.js ซึ่งมีคำสั่งต่าง ๆ ดังแสดงในรูปที่ 11.19

```

1.  var xmlhttp = null;
2.  if(window.ActiveXObject)
3.      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
4.  else if(window.XMLHttpRequest)
5.      xmlhttp = new XMLHttpRequest();
6.  else
7.      alert("Your browser does not support AJAX.");
8.  function processData(str){
9.      if(xmlhttp != null){
10.         xmlhttp.open("GET", "note_display.php?input="+
11.         +str+"&id="+Math.random(), true);
12.         xmlhttp.send(null);
13.         xmlhttp.onreadystatechange = function(){
14.             if(xmlhttp.readyState == 4){
15.                 document.getElementById("displayID").innerHTML=
16.                 xmlhttp.responseText;
17.             }
18.         }
19.     }
20. }

```

รูปที่ 11.19: ข้อมูลในไฟล์ note.js

คำสั่งต่าง ๆ ในรูปที่ 11.19 นั้น ส่วนใหญ่แล้วจะเหมือนกับคำสั่งในรูปที่ 11.12 ซึ่งได้อธิบายถึงการทำงานของคำสั่งเหล่านั้นไปแล้ว แต่ส่วนที่แตกต่างก็คือการใช้คำสั่งในบรรทัดที่ 8, 10 และ 14 เท่านั้น จึงขออธิบายเฉพาะการใช้คำสั่งในบรรทัดดังกล่าว

บรรทัดที่ 8 เริ่มต้นเข้าสู่ฟังก์ชัน processData() พร้อมกับประกาศตัวแปรชื่อว่า str เพื่อรับข้อมูลที่ส่งมา

บรรทัดที่ 10 เรียกใช้เมทอด open เพื่อเปิดการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ซึ่งใช้เมทอดการส่งข้อมูลแบบ GET ไปที่ไฟล์ note_display.php โดยส่งตัวแปรชื่อว่า input ที่มีค่าเท่ากับข้อมูลที่ผู้ใช้เลือกจากแบบฟอร์มในไฟล์ note_form.html และตัวแปรชื่อว่า id ที่มีค่าเท่ากับการสุ่มตัวเลขของการใช้อ็อบเจกต์ Math กับเมทอด random (ค่าที่ได้คือเลขจำนวนจริงที่มีค่าระหว่าง 0.0 ถึง 1.0) สาเหตุที่ส่งค่าของตัวแปร id มาด้วยก็เพราะว่าต้องการป้องกันเว็บเซิร์ฟเวอร์ใช้แคชไฟล์) และเป็นารเชื่อมต่อแบบ Asynchronous

บรรทัดที่ 14 ข้อมูลที่เว็บเซิร์ฟเวอร์ส่งกลับมาให้เป็นเท็กซ์นั้น ให้นำไปแทรกลงในหน้าเว็บตรงตำแหน่งที่มีไอดีคือ displayed

ไฟล์ note_form.html มีคำสั่งต่าง ๆ ดังแสดงในรูปที่ 11.20

```

1. <html><head><title>Note From Friends</title></head><body>
2. <script src="note.js"></script>
3. <center><b>ตัวอย่าง: การใช้ AJAX ช่วยในการค้นหาข้อมูลจากแฟ้มข้อมูล XML</b><hr
width="60%"><br>
4. <form> เลือกข้อมูลที่ต้องการ<br><br>
5. <select name="names" onchange="processData(this.value);">
6. <option>-- Please Select One --</option>
7. <option value="Kacha Chansilp">Kacha Chansilp</option>
8. <option value="Somsak Jaidee">Somsak Jaidee</option>
9. <option value="Somsri Boonmak">Somsri Boonmak</option>
10. <option value="Yoodee Meesuk">Yoodee Meesuk</option>
11. </select></form><hr width="60%">
12. <p><div id="displayID"><b>ข้อมูลจะถูกนำมาแสดงตรงนี้</b></div></p>
13. </body></html>

```

รูปที่ 11.20: ข้อมูลในไฟล์ note_form.html

บรรทัดที่ 1 เปิดแท็ก <html>, <head>, <title> แสดงข้อความ “Note From Friends” ในส่วนของ title และปิดแท็ก </title> และ </head> แล้วเปิดแท็ก <body>

บรรทัดที่ 2 เริ่มการเขียนคำสั่งจาวาสคริปต์ โดยดึงคำสั่งต่าง ๆ มาจากไฟล์ note.js แล้วจบการทำงานของจาวาสคริปต์

บรรทัดที่ 3 แสดงข้อความ “ตัวอย่าง: การใช้ AJAX ช่วยในการค้นหาข้อมูลจากแฟ้มข้อมูล XML” เป็นตัวหนาอยู่ตรงกลางของหน้าเว็บพร้อมกับแสดงเส้นแนวนอนมีขนาดความยาว 60 เปอร์เซนต์ของหน้าเว็บแล้วขึ้นบรรทัดใหม่

บรรทัดที่ 4 เปิดแท็ก <form> แสดงข้อความ “เลือกข้อมูลที่ต้องการ” แล้วขึ้นบรรทัดใหม่ 2 บรรทัด

บรรทัดที่ 5 เปิดแท็ก <select> ชื่อว่า names และมีการเรียกใช้เหตุการณ์ (event) เมื่อเกิดการเปลี่ยนค่าหรือสถานะ (onchange) จากตัวเลือกที่มีให้ ให้ทำการเรียกใช้ฟังก์ชัน processData() พร้อมกับส่งค่าที่เปลี่ยนไปด้วย

บรรทัดที่ 6 ตัวเลือกที่ 1 ให้แสดงข้อความ "-- Please Select One --"

บรรทัดที่ 7 ตัวเลือกที่ 2 ให้แสดงข้อความ "Kacha Chansilp" มีค่าเท่ากับ Kacha Chansilp

บรรทัดที่ 8 ตัวเลือกที่ 3 ให้แสดงข้อความ "Somsak Jaidee" มีค่าเท่ากับ Somsak Jaidee

บรรทัดที่ 9 ตัวเลือกที่ 4 ให้แสดงข้อความ "Somsri Boonmak" มีค่าเท่ากับ Somsri Boonmak

บรรทัดที่ 10 ตัวเลือกที่ 5 ให้แสดงข้อความ "Yoodee Meesuk" มีค่าเท่ากับ Yoodee Meesuk

บรรทัดที่ 11 ปิดแท็ก </select> และ </form> แล้วแสดงเส้นแนวนอนมีขนาดความยาว 60 เปอร์เซ็นต์ของหน้าเว็บ

บรรทัดที่ 12 ขึ้นย่อหน้าใหม่แล้วเปิดแท็ก <div> พร้อมกับกำหนด id ให้มีค่าเท่ากับ displayID และแสดงข้อความ "ข้อมูลจะถูกนำมาแสดงตรงนี้" เป็นตัวหนา แล้วตามด้วยปิดแท็ก </div> และ </p>

บรรทัดที่ 13 ปิดแท็ก </body> และ </html>

ไฟล์ note_display.php มีคำสั่งต่าง ๆ ดังแสดงในรูปที่ 11.21

```

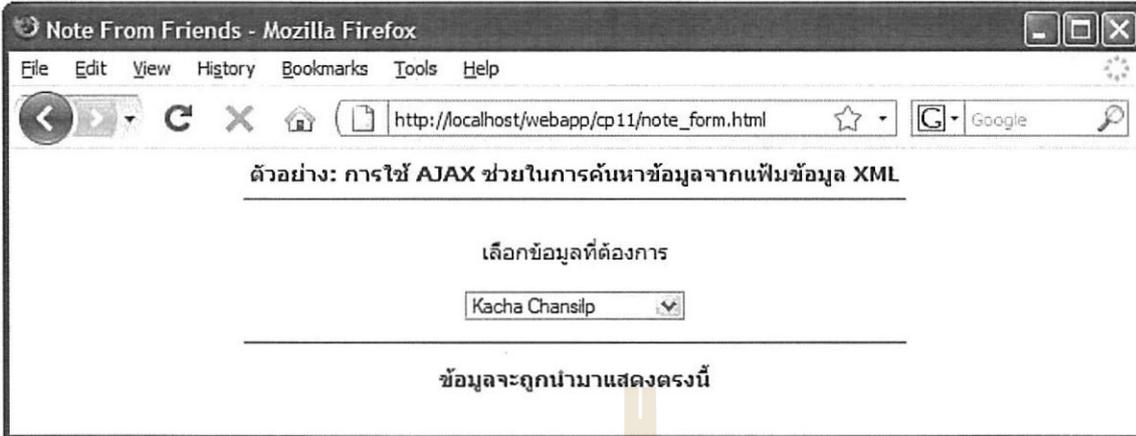
1.  <?php
2.      $txt=$_GET["input"];
3.      $xmlDoc = new DOMDocument();
4.      $xmlDoc->load("note.xml");
5.      $n=$xmlDoc->getElementsByTagName('NAME');
6.      for($i=0; $i<$n->length; $i++){
7.          if($n->item($i)->nodeType==1){
8.              if($n->item($i)->childNodes->item(0)->nodeValue == $txt){
9.                  $p=($n->item($i)->parentNode);
10.                 break;
11.             }
12.         }
13.     }
14.     $person=($p->childNodes);
15.     for($i=0;$i<$person->length;$i++){
16.         if($person->item($i)->nodeType==1){
17.             echo ($person->item($i)->nodeName).": ";
18.             echo ($person->item($i)->nodeValue)."<br>";
19.         }
20.     }
21.     ?>

```

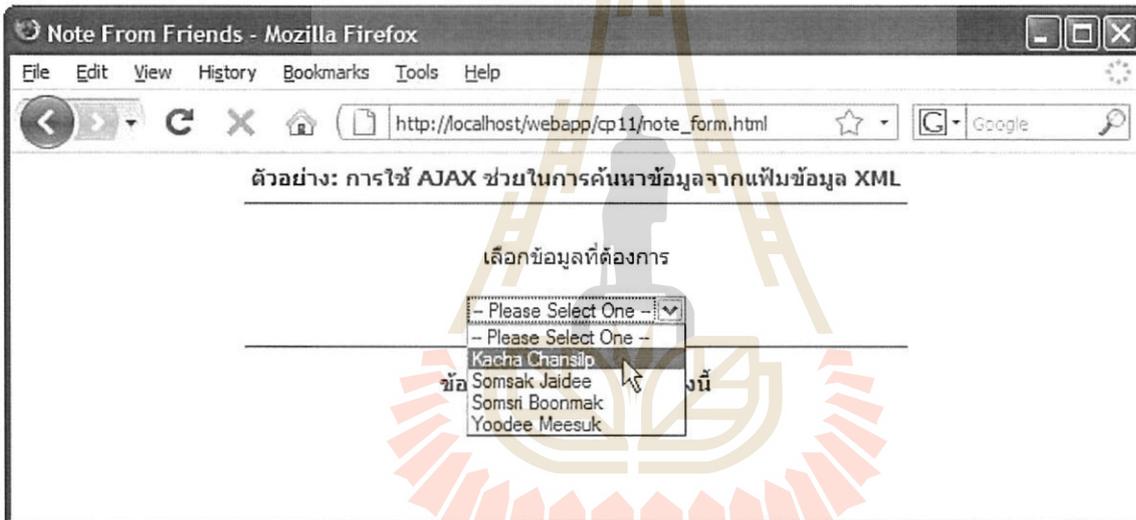
รูปที่ 11.21: คำสั่งต่าง ๆ ในไฟล์ note_display.php

- บรรทัดที่ 1 เริ่มการเขียนคำสั่ง PHP
- บรรทัดที่ 2 ประกาศตัวแปร `$txt` ให้เก็บค่าที่ส่งมาแบบ GET จากตัวแปร `$input`
- บรรทัดที่ 3 สร้างอ็อบเจกต์ใหม่แบบ DOM (Document Object Model) ซึ่งจะใช้สำหรับการเข้าถึงแฟ้มข้อมูล XML ให้กับตัวแปร `$xmlDoc`
- บรรทัดที่ 4 โหลดข้อมูลจากไฟล์ `note.xml` แล้วเก็บไว้ที่ตัวแปร `$xmlDoc`
- บรรทัดที่ 5 ดึงข้อมูลจากตัวแปร `$xmlDoc` โดยเรียกใช้ `getElementsByTagName` ที่มีชื่อเท่ากับ NAME ให้กับตัวแปร `$n`
- บรรทัดที่ 6 ใช้ `for` หลบทำการวนเป็นจำนวนเท่ากับจำนวนข้อมูล (ความยาว) ในตัวแปร `$n`
- บรรทัดที่ 7 ตรวจสอบชนิดข้อมูล `$n->item($i)->nodeType` ว่าเป็น element (มีค่าเท่ากับ 1) หรือไม่ ถ้าใช่ก็จะไปทำคำสั่งในบรรทัดที่ 8 มิฉะนั้นแล้วจะไปวนหลูปใหม่
- บรรทัดที่ 8 ตรวจสอบค่าของ `$n->item($i)->childNodes->item(0)->nodeValue` ว่ามีค่าเท่ากับข้อมูลของตัวแปร `$txt` หรือไม่ ถ้าใช่ก็จะไปทำคำสั่งในบรรทัดที่ 9-10 มิฉะนั้นแล้วจะไปวนหลูปใหม่
- บรรทัดที่ 9 กำหนดค่า `$n->item($i)->parentNode` ให้กับตัวแปร `$p`
- บรรทัดที่ 10 เรียกใช้คำสั่ง `break;` เพื่อให้หลุดออกจาก `for` หลูป
- บรรทัดที่ 11 จบการทำงานของ `if` ในบรรทัดที่ 8
- บรรทัดที่ 12 จบการทำงานของ `if` ในบรรทัดที่ 7
- บรรทัดที่ 13 จบการทำงานของ `for` หลูปในบรรทัดที่ 6
- บรรทัดที่ 14 กำหนดค่า `$p->childNodes` ให้กับตัวแปร `$person`
- บรรทัดที่ 15 ใช้ `for` หลบทำการวนเป็นจำนวนเท่ากับจำนวนข้อมูล (ความยาว) ในตัวแปร `$person`
- บรรทัดที่ 16 ตรวจสอบชนิดข้อมูล `$person->item($i)->nodeType` ว่าเป็น element (มีค่าเท่ากับ 1) หรือไม่ ถ้าใช่ก็จะไปทำคำสั่งในบรรทัดที่ 17 มิฉะนั้นแล้วจะไปวนหลูปใหม่
- บรรทัดที่ 17 แสดงข้อมูล `$person->item($i)->nodeName` พร้อมกับเครื่องหมาย “:” ออกที่จอภาพ
- บรรทัดที่ 18 แสดงข้อมูล `$person->item($i)->nodeValue` ออกที่จอภาพแล้วขึ้นบรรทัดใหม่
- บรรทัดที่ 19 จบการทำงานของ `if` ในบรรทัดที่ 16
- บรรทัดที่ 20 จบการทำงานของ `for` หลูปในบรรทัดที่ 15
- บรรทัดที่ 21 จบการทำงานของคำสั่ง PHP

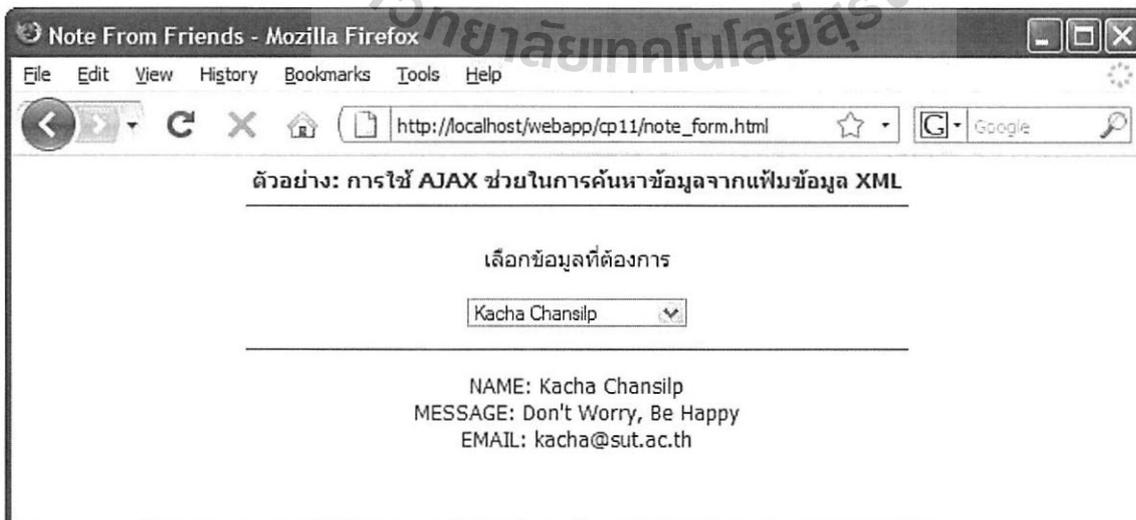
รูปที่ 11.22 – 11.24 แสดงถึงผลการทำงานการใช้ XMLHttpRequest กับไฟล์ XML



รูปที่ 11.22: การแสดงผลของไฟล์ note_form.html



รูปที่ 11.23: แสดงข้อมูลตัวเลือกของไฟล์ note_form.html



รูปที่ 11.24: แสดงผลจากการเลือกข้อมูลของไฟล์ note_form.html

11.5 การใช้งาน XMLHttpRequest กับฐานข้อมูล

การใช้งาน XMLHttpRequest กับฐานข้อมูลก็มีลักษณะคล้ายกันกับ XML เพียงแต่เพิ่มการเชื่อมต่อกับระบบฐานข้อมูลเข้ามาเท่านั้น โดยจะขอยกตัวอย่างในการเชื่อมต่อเว็บเซิร์ฟเวอร์ ติดต่อระบบฐานข้อมูล เปรียบเทียบแล้วนำผลที่ต้องการมาแสดงออกที่หน้าเว็บ ซึ่งมีข้อมูลอยู่ในตารางชื่อว่า contact ในฐานข้อมูล ชื่อว่า friends ดังแสดงในรูปที่ 11.25 และไฟล์อีก 3 ไฟล์ ดังรายละเอียดต่อไปนี้

← T →			name ▲	email
<input type="checkbox"/>			anucha yordmak	anucha_y@gmail.com
<input type="checkbox"/>			boondej dechawoot	b_dechawoot@yahoo.com
<input type="checkbox"/>			kacha chansilp	kacha@sut.ac.th
<input type="checkbox"/>			sombat yoocharoen	sombat77@hotmail.com
<input type="checkbox"/>			somsak jaidee	somsak@yahoo.com
<input type="checkbox"/>			suthep narktan	suthep@yahoo.com

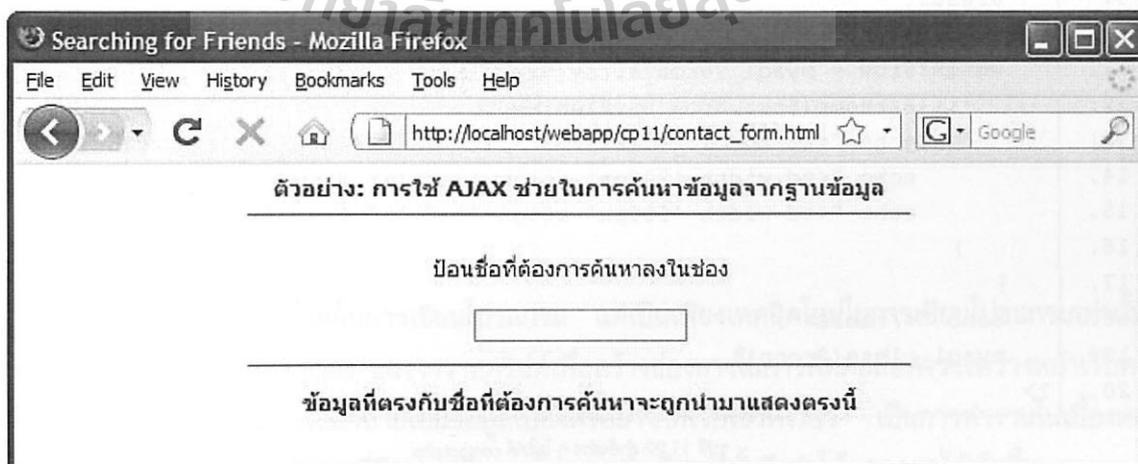
รูปที่ 11.25: ข้อมูลตาราง contact ในฐานข้อมูล friends

1. ไฟล์ contact_form.html มีคำสั่งต่าง ๆ ดังแสดงในรูปที่ 11.26 และแสดงผลดังรูปที่ 11.27

```

1. <html><head><title>Searching for Friends</title></head><body>
2. <script src="contact.js"></script>
3. <center><b>ตัวอย่าง: การใช้ AJAX ช่วยในการค้นหาข้อมูลจากฐานข้อมูล</b>
4. <hr width="60%"><br>
5. <form> ป้อนชื่อที่ต้องการค้นหาลงในช่อง<br><br>
6. <input type="text" onkeyup="processData();" name="input" id="inputID">
7. </form>
8. <hr width="60%">
9. <p><div id="displayID"><b>ข้อมูลที่ตรงกับชื่อที่ต้องการค้นหาจะถูกนำมาแสดงตรงนี้
10. </b></div></p>
11. </body>
12. </html>
    
```

รูปที่ 11.26: คำสั่งต่าง ๆ ในไฟล์ contact_form.html



รูปที่ 11.27: การแสดงผลจากไฟล์ contact_form.html

2. ไฟล์ contact.js มีคำสั่งต่าง ๆ ดังแสดงในรูปที่ 11.28

```

1. var xmlhttp = null;
2. if(window.ActiveXObject)
3.     xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); //IE5.x & IE6
4. else if(window.XMLHttpRequest)
5.     xmlhttp = new XMLHttpRequest();
6. else
7.     alert("Your browser does not support AJAX.");
8. function processData(){
9.     if(xmlhttp != null){
10.        xmlhttp.open("GET", "contact_display.php?input="+
+document.getElementById('inputID').value, true);
11.        xmlhttp.send(null);
12.        xmlhttp.onreadystatechange = function(){
13.            if(xmlhttp.readyState == 4){
14.                document.getElementById("displayID").innerHTML=
xmlhttp.responseText;
15.            }
16.        }
17.    }
18. }
    
```

รูปที่ 11.28: คำสั่งต่าง ๆ ในไฟล์ contact.js

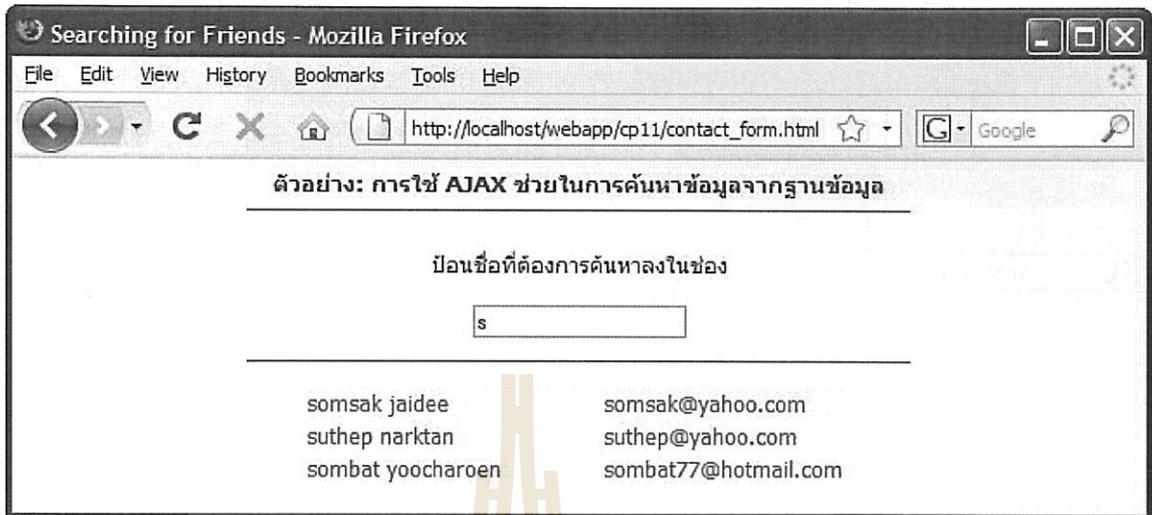
3. ไฟล์ contact.php มีคำสั่งต่าง ๆ ดังแสดงในรูปที่ 11.29

```

1. <?php
2.     $txt=$_GET["input"];
3.     $len = strlen($txt);
4.     $conn = mysql_connect('localhost', 'root', 'meroot');
5.     if(!$conn)
6.         die('Could not connect: ' . mysql_error());
7.     mysql_select_db("friends", $conn);
8.     $sql="SELECT * FROM contact";
9.     $result = mysql_query($sql);
10.    echo "<table border='0' width='400px'>";
11.    while($row = mysql_fetch_array($result)){
12.        if(strncmp($txt,$row[0],$len)==0){
13.            echo "<tr bgcolor='#FFFF99'>";
14.            echo "<td width='200px' align='left'>".$row[0]."</td>";
15.            echo "<td width='150px' align='left'>".$row[1]."</td></tr>";
16.        }
17.    }
18.    echo "</table>";
19.    mysql_close($conn);
20. ?>
    
```

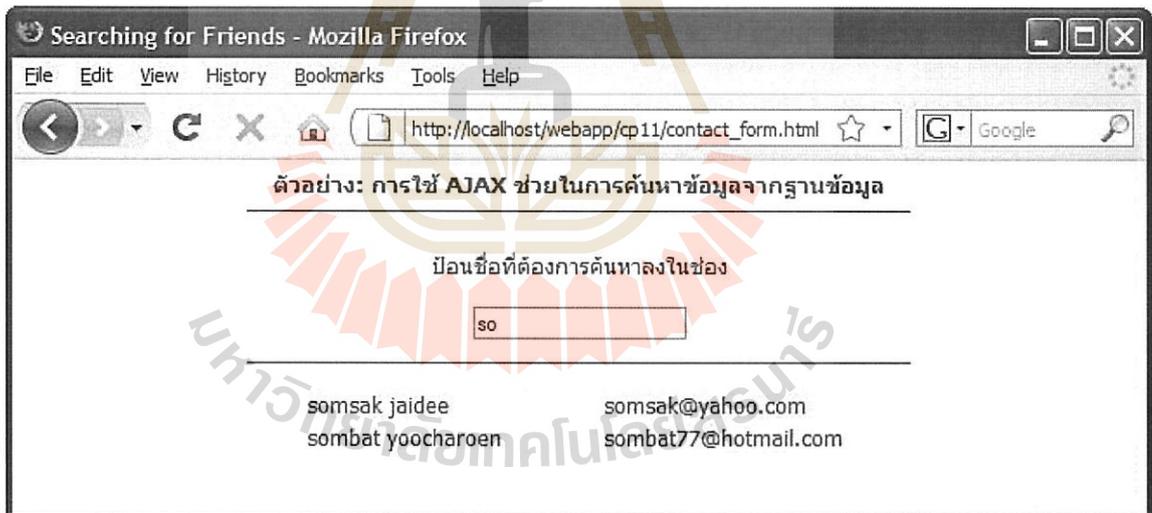
รูปที่ 11.29: คำสั่งต่าง ๆ ในไฟล์ contact.php

รูปที่ 11.30 เป็นการแสดงผลเมื่อผู้ใช้ป้อนตัวอักษร s เข้าไป ข้อมูลที่ขึ้นต้นด้วยตัวอักษร s จะถูกนำมาแสดงทั้งหมด



รูปที่ 11.30: แสดงผลของข้อมูลทั้งหมดที่เริ่มต้นด้วยตัวอักษร s

รูปที่ 11.31 เป็นการแสดงผลเมื่อผู้ใช้ป้อนตัวอักษร so เข้าไป ข้อมูลที่ขึ้นต้นด้วยตัวอักษร so จะถูกนำมาแสดงทั้งหมด



รูปที่ 11.31: แสดงผลของข้อมูลทั้งหมดที่เริ่มต้นด้วยตัวอักษร so

สรุป

เอแจ็กไม่ใช่ภาษาใหม่ในการเขียนโปรแกรม แต่เป็นเพียงเทคนิคใหม่ในการเขียนโปรแกรมเท่านั้นที่ใช้อ็อบเจกต์ XMLHttpRequest ของจาวาสคริปต์เพื่อสร้างช่องทางในการติดต่อสื่อสารระหว่างหน้าเว็บทางฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์แล้วถ่ายโอนข้อมูลไปยังหรือมาจากเว็บเซิร์ฟเวอร์ เป็นการทำงานในเบื้องหลังโดยจัดการเฉพาะส่วนของข้อมูลที่มีการเปลี่ยนแปลงบนหน้าเว็บเท่านั้นจึงทำให้แสดงผลได้เร็วขึ้น

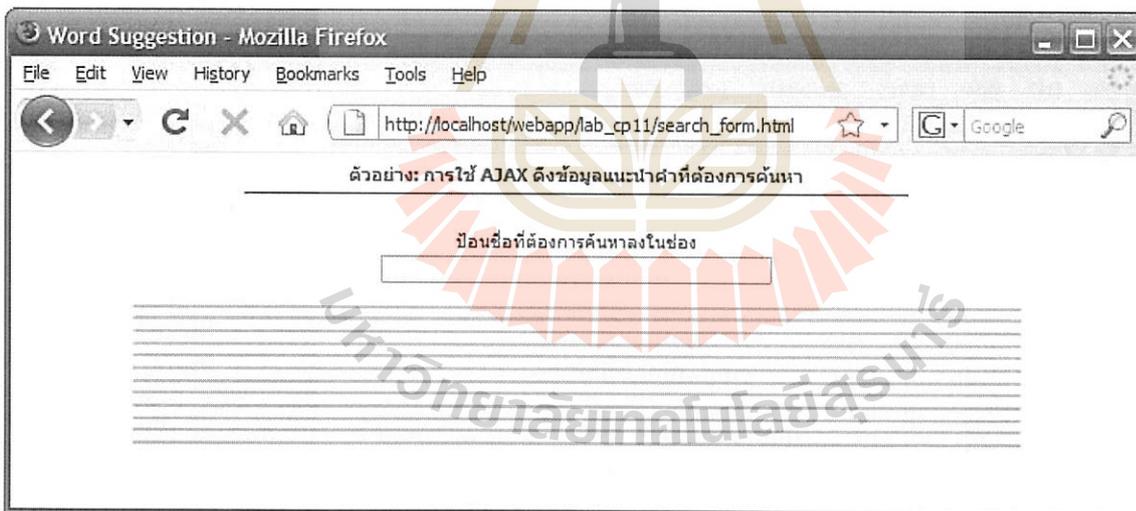
แบบฝึกหัด

1. สร้างฐานข้อมูลชื่อว่า friends มีตารางชื่อว่า contact และมีข้อมูลดังแสดงในรูปที่ 11.32

			name ▲	email
<input type="checkbox"/>			anucha yordmak	anucha_y@gmail.com
<input type="checkbox"/>			boondej dechawoot	b_dechawoot@yahoo.com
<input type="checkbox"/>			kacha chansilp	kacha@sut.ac.th
<input type="checkbox"/>			sandee sanden	s_sanden@gmail.com
<input type="checkbox"/>			seeda danthong	seeda@hotmail.com
<input type="checkbox"/>			sombat yoocharoen	sombat77@hotmail.com
<input type="checkbox"/>			somsak jaidee	somsak@yahoo.com
<input type="checkbox"/>			surin namdee	surin@yahoo.com
<input type="checkbox"/>			sutee meejai	mejai@thaimail.com
<input type="checkbox"/>			suthep naritan	suthep@yahoo.com

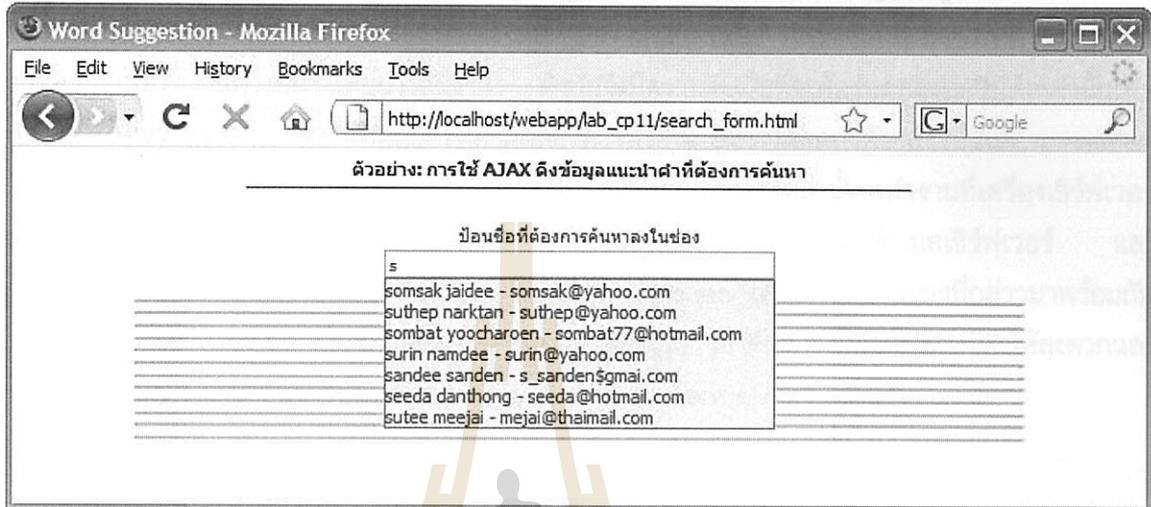
รูปที่ 11.32: ข้อมูลในตาราง contact

2. สร้างแบบฟอร์มชื่อว่า search_form.html ดังแสดงในรูปที่ 11.33

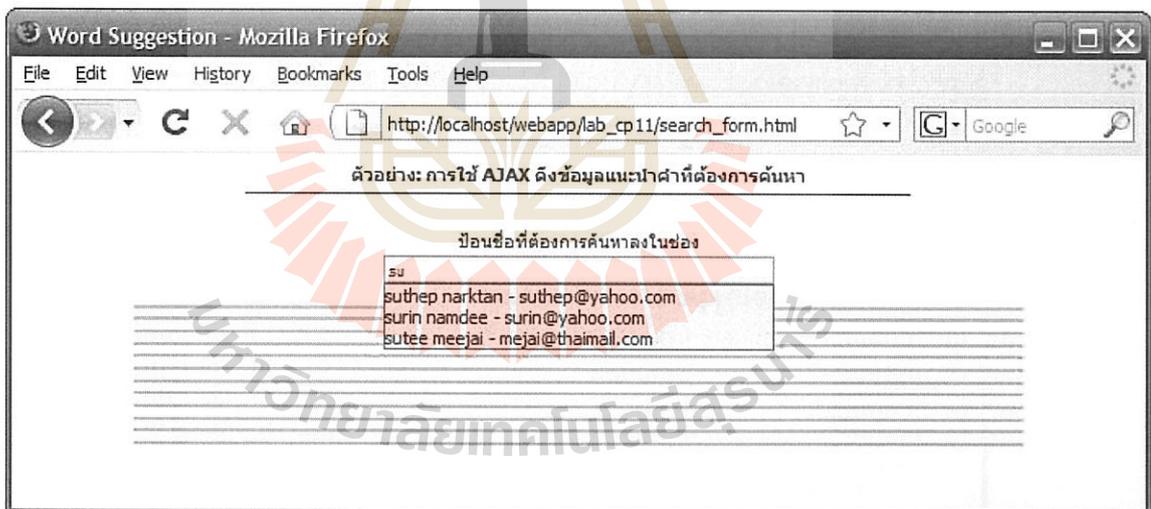


รูปที่ 11.33: แสดงผลของไฟล์ search_form.html

3. ใช้เทคโนโลยีเอแจ็กเพื่อดึงข้อมูลจากฐานข้อมูลมาแสดงผล เช่น ผู้ใช้ใส่ตัวอักษร s เข้ามา โปรแกรมจะทำการส่งข้อมูลนั้นไปที่เว็บเซิร์ฟเวอร์ แล้วทำการเชื่อมต่อกับฐานข้อมูลเพื่อแสดงผลของชื่อที่เริ่มต้นด้วยตัวอักษร s ออกที่หน้าจอที่เครื่องผู้ใช้ ดังแสดงในรูปที่ 11.34 จะสังเกตเห็นว่ามีข้อมูลคือชื่อและอีเมลถูกนำมาแสดงออกที่หน้าจอทั้งหมด 7 ชื่อ และเมื่อผู้ใช้ป้อนตัวอักษรตัวต่อไปคือตัว u ข้อมูลที่ถูกแสดงออกที่หน้าจอจะเหลืออยู่แค่ 3 ชื่อ นั่นคือข้อมูลที่เริ่มต้นด้วยตัวอักษร su ดังแสดงในรูปที่ 11.35



รูปที่ 11.34: แสดงข้อมูลเมื่อผู้ใช้ป้อนตัวอักษร s



รูปที่ 11.35: แสดงข้อมูลเมื่อผู้ใช้ป้อนตัวอักษร su

ภาคผนวก ก

การติดตั้ง AppServ <AppServ Installation>

การพัฒนาเว็บไซต์ที่ใช้เฉพาะแท็ก HTML สามารถทดลองการแสดงผลของไฟล์นั้น ๆ ได้โดยผ่านโปรแกรมเว็บเบราว์เซอร์บนเครื่องคอมพิวเตอร์ของผู้พัฒนาเอง แต่โดยส่วนใหญ่แล้วผู้พัฒนาจะใช้สคริปต์ต่าง ๆ เช่น ASP, PHP เป็นต้น ในการพัฒนาเว็บไซต์ เพื่อรับข้อมูล ประมวลผล แสดงผล หรือติดต่อฐานข้อมูลเพื่อดำเนินการอย่างใดอย่างหนึ่ง ดังนั้นจึงมีความจำเป็นที่จะต้องทดลองสคริปต์เหล่านั้นเพื่อทดสอบระบบการทำงานว่าทำงานได้ถูกต้องตามที่ต้องการหรือไม่ ซึ่งการทดลองดังกล่าวนี้ไม่สามารถทำได้โดยเรียกผ่านทางโปรแกรมเว็บเบราว์เซอร์แบบปกติ เนื่องจากสคริปต์เหล่านั้นจะทำงานที่เครื่องเซิร์ฟเวอร์ ดังนั้นในเครื่องของผู้พัฒนาเว็บไซต์เองจะต้องทำการติดตั้งเว็บเซิร์ฟเวอร์ ดาต้าเบสเซิร์ฟเวอร์ และโปรแกรมประมวลผลการทำงานของสคริปต์ ซึ่งโปรแกรม AppServ ได้รวมทั้ง 3 อย่างที่กล่าวมาพร้อมกับโปรแกรม phpMyAdmin (โปรแกรมในการจัดการฐานข้อมูล) เข้าด้วยกันเป็นแพ็คเกจ ทำให้สะดวกและง่ายต่อการติดตั้ง สามารถดาวน์โหลดได้ที่ <http://www.appservnetwork.com/>

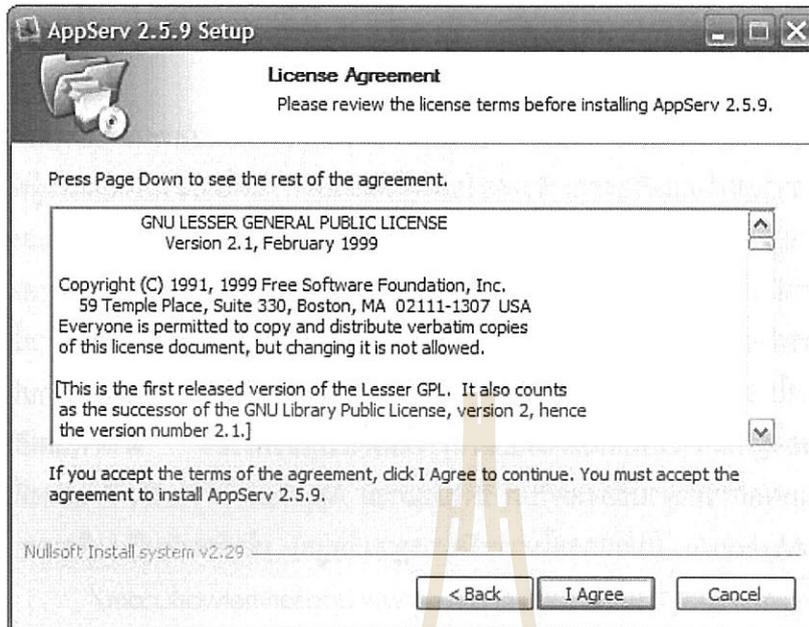
AppServ ที่แนะนำเสนอเป็นตัวอย่างในการติดตั้งครั้งนี้เป็นเวอร์ชัน 2.5.9 ซึ่งได้รวม เว็บเซิร์ฟเวอร์ Apache เวอร์ชัน 2.2.4, โปรแกรมประมวลผลการทำงานของ PHP สคริปต์ เวอร์ชัน 5.2.3, ระบบฐานข้อมูล MySQL เวอร์ชัน 5.0.45 และโปรแกรมจัดการระบบฐานข้อมูล phpMyAdmin เวอร์ชัน 2.10.2 ซึ่งมีขั้นตอนในการติดตั้งดังต่อไปนี้

1. ดาวน์โหลดไฟล์โปรแกรมจากเว็บไซต์ <http://www.appservnetwork.com/>
2. ดับเบิลคลิกที่ไฟล์โปรแกรม จะปรากฏหน้าต่างดังรูปที่ ก1



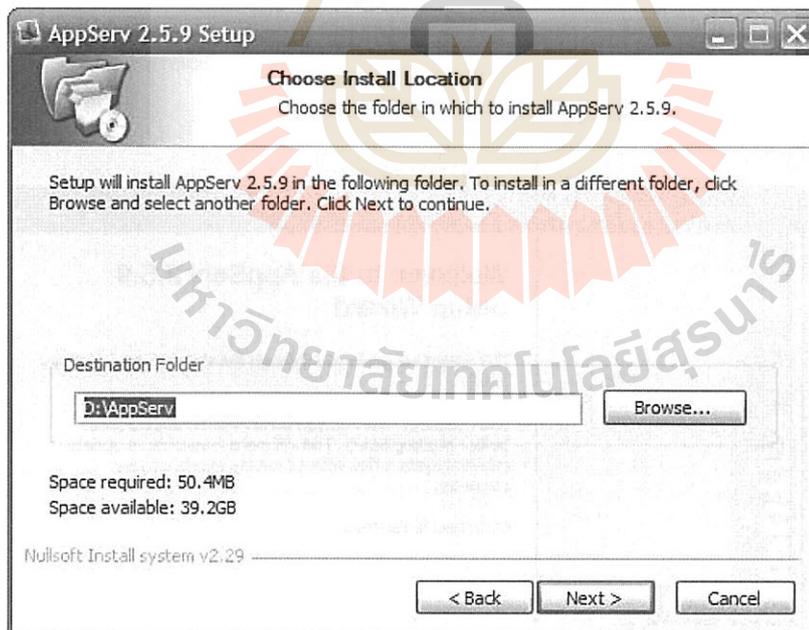
รูปที่ ก1: หน้าต่างเริ่มต้นการติดตั้งโปรแกรม AppServ

3. กดปุ่ม “Next >” จะปรากฏหน้าต่างดังรูปที่ ก2



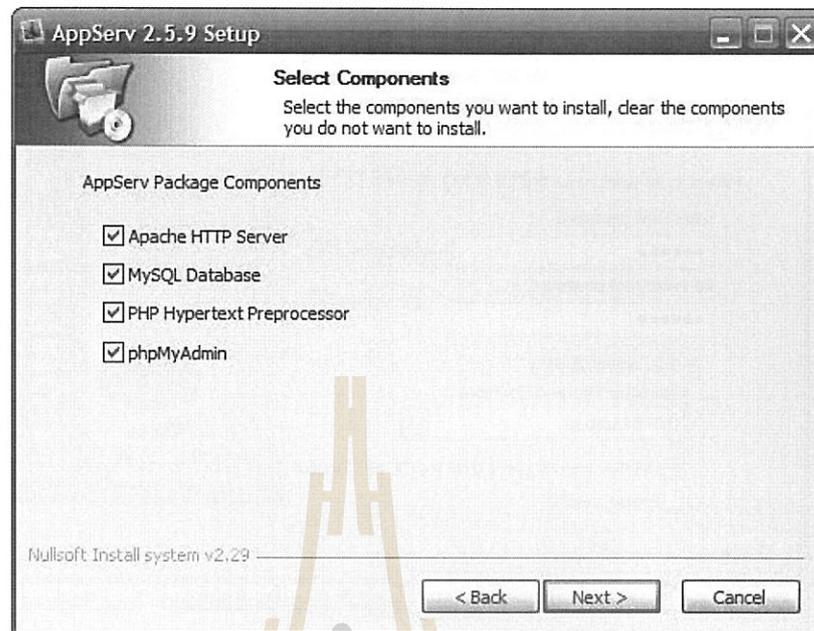
รูปที่ ก2: หน้าต่างแสดงลิขสิทธิ์

4. กดปุ่ม “I Agree” จะปรากฏหน้าต่างดังรูปที่ ก3



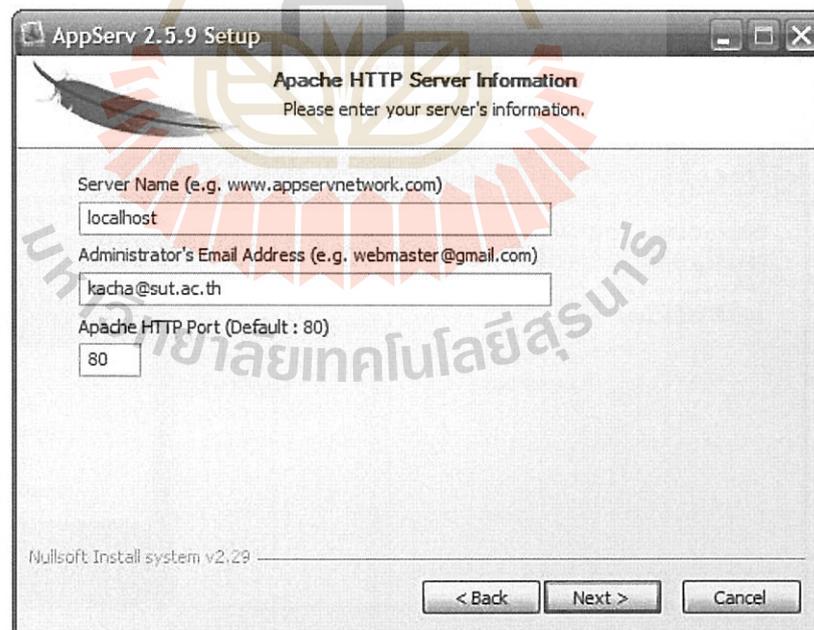
รูปที่ ก3: หน้าต่างแสดงตำแหน่งการติดตั้งโปรแกรม

5. ให้เลือกตำแหน่งที่ต้องการติดตั้ง (ในที่นี้ขอเลือกติดตั้งในไดรฟ์ D) แล้วกดปุ่ม “Next >” จะปรากฏหน้าต่างดังรูปที่ ก4



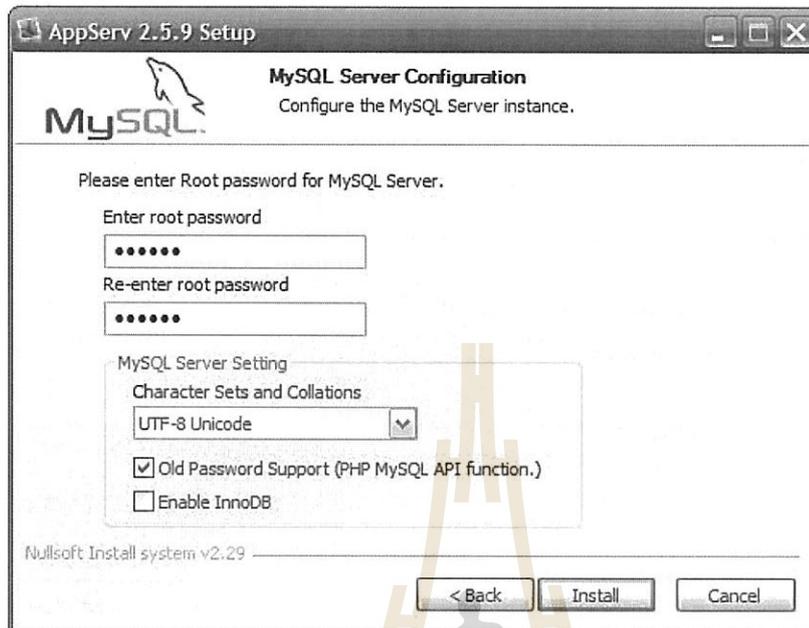
รูปที่ ก4: หน้าต่างแสดงตัวเลือกในการติดตั้ง

6. ให้เลือกติดตั้งทั้ง 4 ตัวเลือก แล้วกดปุ่ม “Next >” จะปรากฏหน้าต่างดังรูปที่ ก5



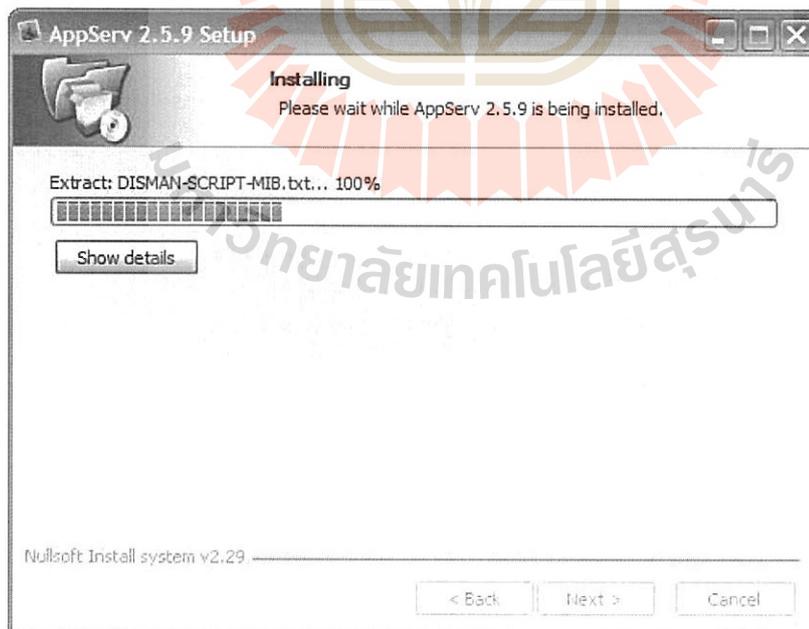
รูปที่ ก5: หน้าต่างแสดงรายละเอียดเกี่ยวกับ HTTP Server

7. ใส่ข้อมูลชื่อของเว็บเซิร์ฟเวอร์ อีเมลของผู้ดูแลเว็บ และ HTTP Port (ค่าปริยายคือ 80) แล้วกดปุ่ม “Next >” จะปรากฏหน้าต่างดังรูปที่ ก6



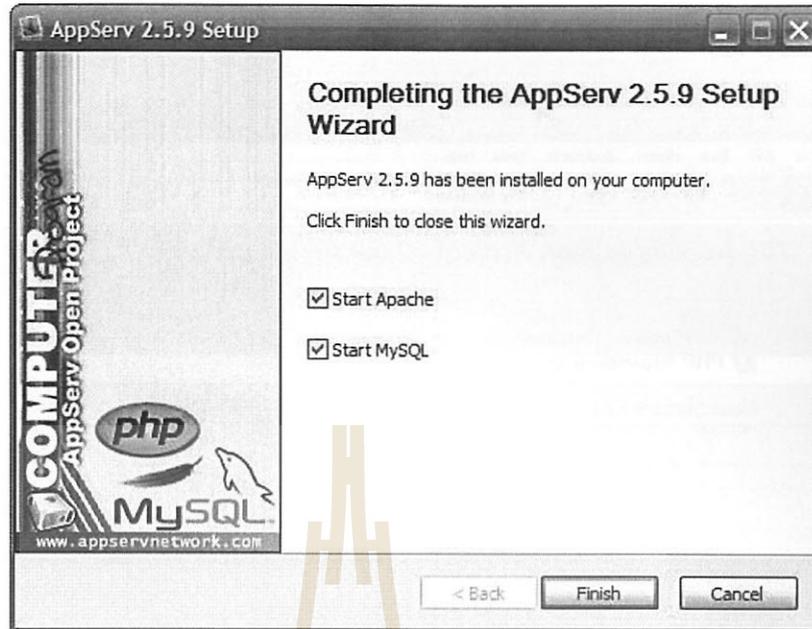
รูปที่ ก6: หน้าต่างแสดงการตั้งค่าการเข้าถึงระบบฐานข้อมูล

8. ใส่รหัสสำหรับการเข้าฐานข้อมูล MySQL เซิร์ฟเวอร์ แล้วกดปุ่ม “Install” จะปรากฏหน้าต่างดังรูปที่ ก7



รูปที่ ก7: หน้าต่างแสดงการทำงานขณะทำการติดตั้ง

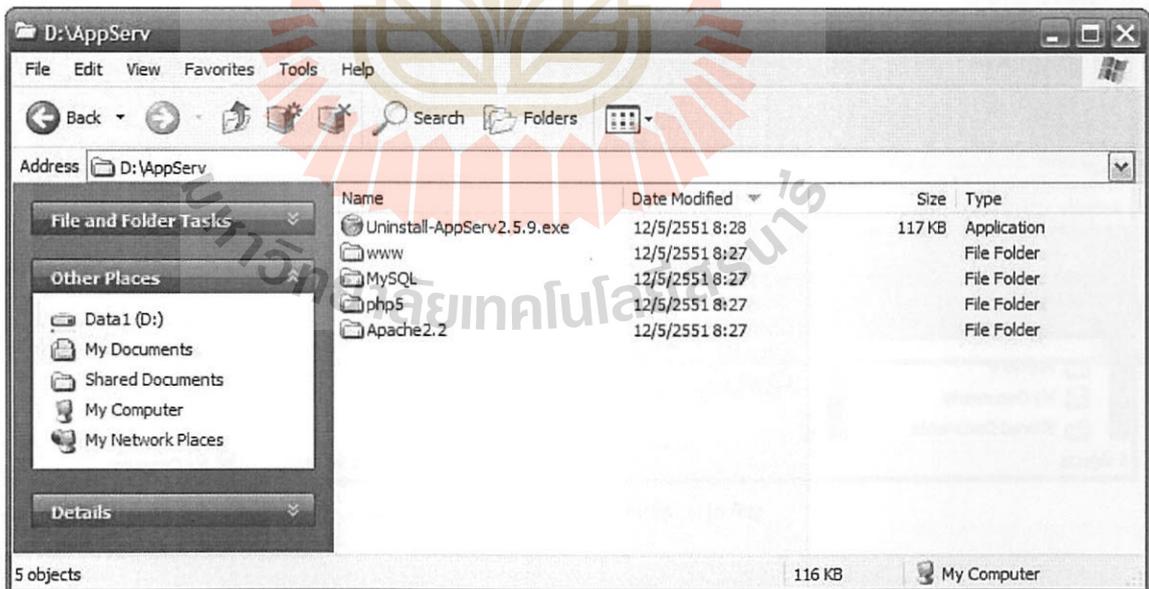
9. เมื่อติดตั้งเสร็จเรียบร้อย จะปรากฏหน้าต่างดังรูปที่ ก8



รูปที่ ก8: หน้าต่างแสดงการติดตั้งสำเร็จ

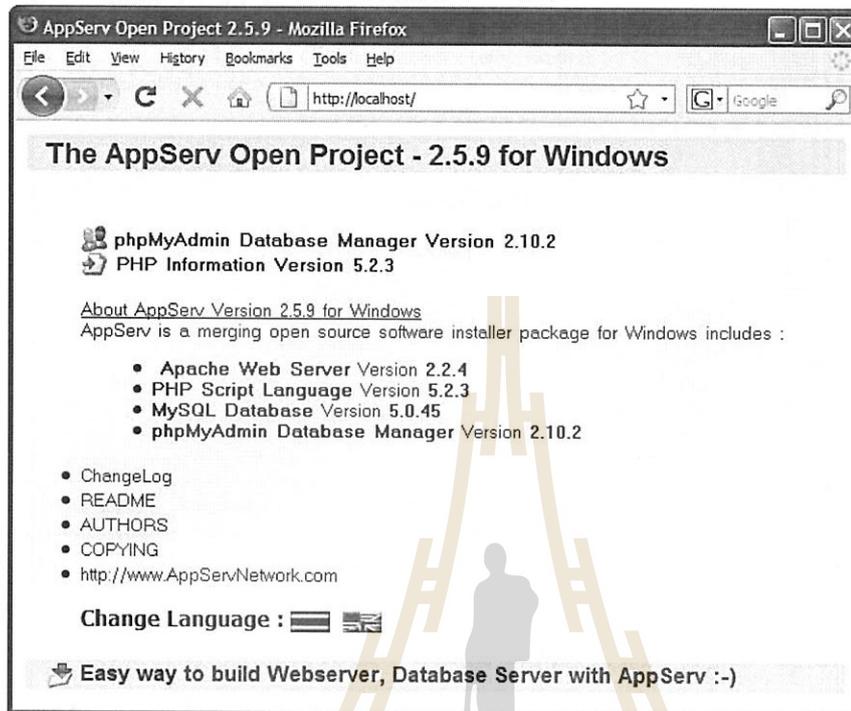
10. กดปุ่ม “Finish” เพื่อเสร็จสิ้นการติดตั้ง

เมื่อเปิดดูตำแหน่งที่ติดตั้งระบบดังกล่าวแล้ว จะเห็นข้อมูลดังแสดงในรูปที่ ก9



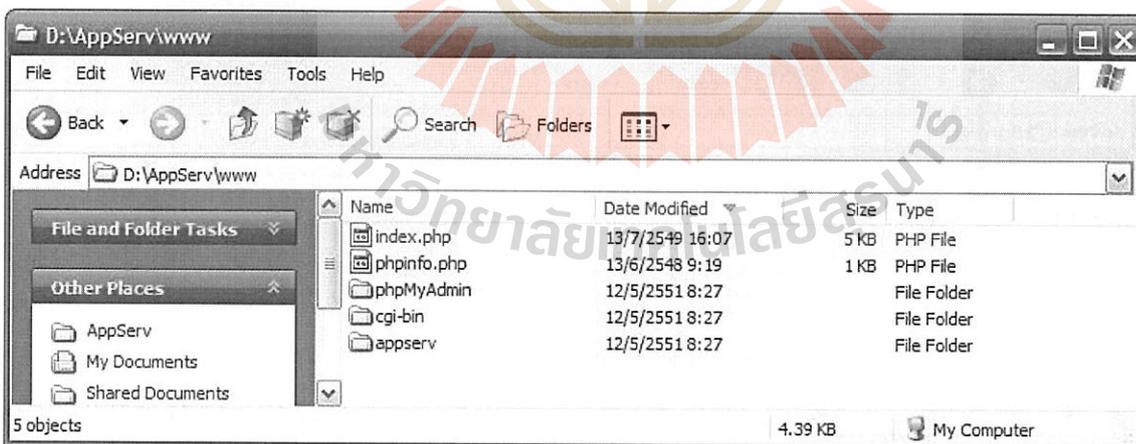
รูปที่ ก9: หน้าต่างแสดงข้อมูลหลังการติดตั้ง

เมื่อเปิดโปรแกรมเว็บเบราว์เซอร์ แล้วพิมพ์ <http://localhost/> หรือ <http://127.0.0.1> ในช่อง URL จะปรากฏรายละเอียดของโปรแกรมต่าง ๆ ที่ได้ติดตั้งลงไป ดังแสดงในรูปที่ ก10



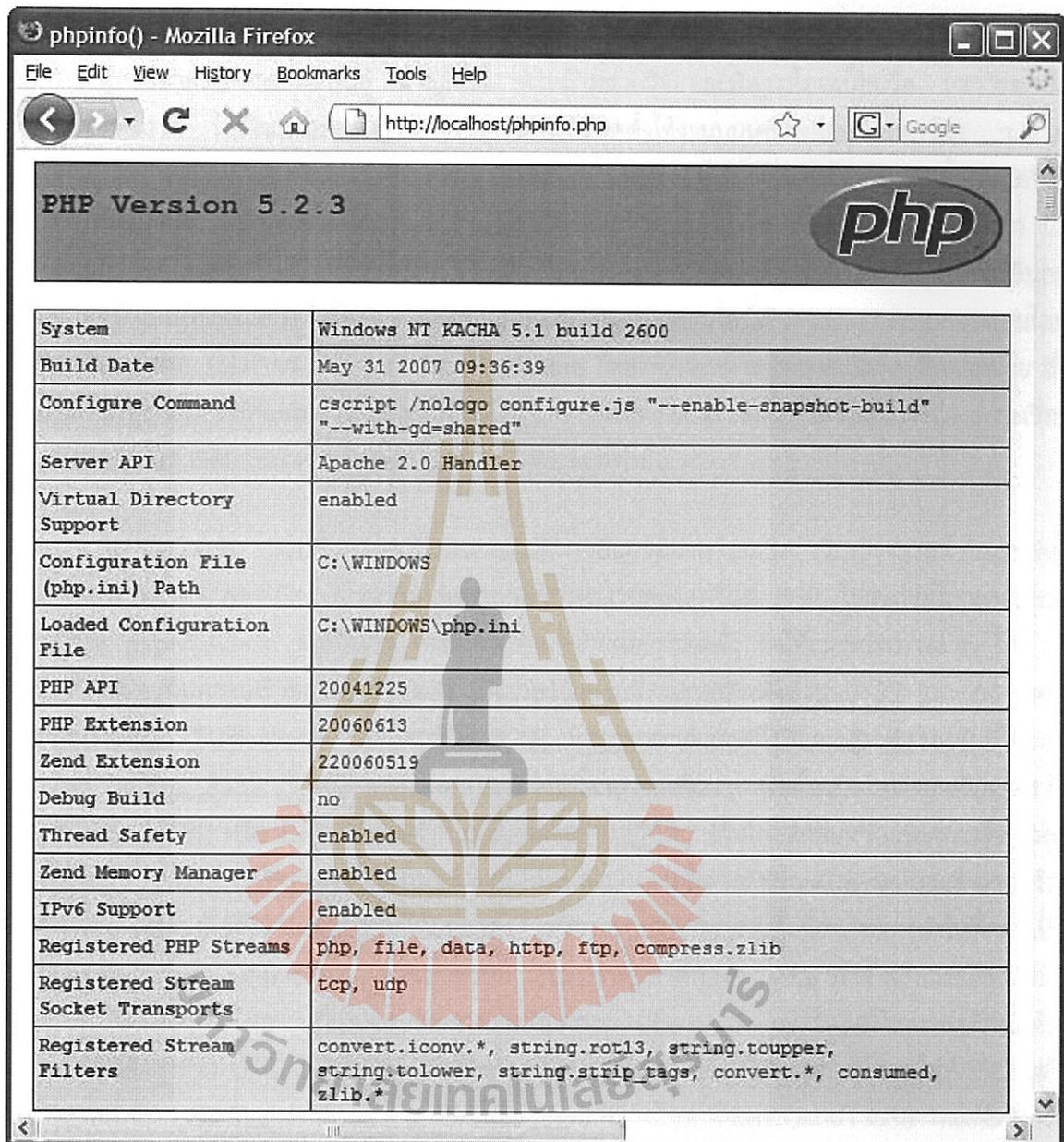
รูปที่ ก10: หน้าต่างแสดงรายละเอียดของโปรแกรมต่าง ๆ ที่ติดตั้งลงไป

ไฟล์สกุล .php จะต้องอยู่ในไดเรกทอรี `D:\AppServ\www\` ดังแสดงในรูปที่ ก11



รูปที่ ก11: หน้าต่างแสดงตำแหน่งที่อยู่ของสคริปต์ .php

ทดลองการทำงานของเซิร์ฟเวอร์โดยพิมพ์ <http://localhost/phpinfo.php> ในช่อง URL จะปรากฏรายละเอียดของ PHP ดังแสดงในรูปที่ ก12



รูปที่ ก12: หน้าต่างแสดงรายละเอียดของโปรแกรม PHP

ภาคผนวก ข

การติดตั้ง SUTinsServer <SUTinsServer Installation>

ผู้ช่วยศาสตราจารย์สมพันธ์ ชาญศิลป์ หัวหน้าหน่วยวิจัยและพัฒนาโอเพนซอร์ส มหาวิทยาลัยเทคโนโลยีสุรนารี ได้พัฒนาระบบปฏิบัติการลินุกซ์สำหรับติดตั้งใช้งานบนเครื่องคอมพิวเตอร์ทั่ว ๆ ไป เรียกว่า SUTinsDesktop (SUT Instant Desktop), ระบบปฏิบัติการลินุกซ์สำหรับติดตั้งเป็นเครื่องเซิร์ฟเวอร์ เรียกว่า SUTinsServer (SUT Instant Server), ระบบปฏิบัติการลินุกซ์สำหรับนักพัฒนา C, C++ และ Java เรียกว่า SUT-MDE (SUT Mobile Development Environment), ระบบปฏิบัติการลินุกซ์สำหรับติดตั้งเป็นเครื่องเซิร์ฟเวอร์ ที่มาพร้อมกับโปรแกรมสอบออนไลน์เรียกว่า ระบบสอบออนไลน์เคลื่อนที่จาก มทส. (SUT-MOTS: SUT Mobile Online Test System) และผลงานอื่น ๆ อีกมากมาย ซึ่งสามารถเข้าดูรายละเอียดและดาวน์โหลดได้ที่ <http://linux.sut.ac.th> แต่ในที่นี้จะขอกล่าวถึงเฉพาะเรื่องของระบบปฏิบัติการลินุกซ์สำหรับติดตั้งเป็นเครื่องเซิร์ฟเวอร์เท่านั้น

SUTinsServer 5207 เป็นระบบปฏิบัติการลินุกซ์ที่มาพร้อมกับ เว็บเซิร์ฟเวอร์ Apache, ดาต้าเบสเซิร์ฟเวอร์ MySQL, โปรแกรมประมวลผลการทำงานของสคริปต์ PHP, โปรแกรมในการจัดการฐานข้อมูล phpMyAdmin และอื่น ๆ ทำให้สะดวกและง่ายต่อการติดตั้ง ซึ่งใช้เวลาประมาณ 5 ถึง 10 นาที ทั้งนี้ขึ้นอยู่กับความเร็วของคอมพิวเตอร์ เป็นเวอร์ชันแรกที่มีการติดตั้งทั้งแบบ 32 และ 64 บิตไว้ด้วยกันในตระกูล SUTinsServer เลขเวอร์ชัน 5207 มีความหมายแสดงถึงปีและเดือนที่เวอร์ชันนี้ได้ออกเผยแพร่ ซึ่งก็คือปี 2552 เดือน 7 (กรกฎาคม) เวอร์ชันนี้สามารถเลือกการติดตั้งแบบไดเรคทอรีบนไดรฟ์ที่เป็น NTFS, FAT32 หรือ Ext3 โดยไม่จำเป็นต้องลงระบบปฏิบัติการวินโดวส์ก่อนหรือเลือกการติดตั้งลงบนทั้งไดรฟ์ โดยไดรฟ์ที่ถูกเลือกจะถูกฟอร์แมตให้เป็น Ext3 ถ้าไม่มีพาร์ติชัน swap ระบบจะทำการสร้างไฟล์ swap ให้โดยผู้ใช้เป็นคนเลือกว่าจะให้มีขนาด 2 หรือ 4 GB. ผลงานชิ้นนี้เหมาะการนำไปติดตั้งเป็นเครื่องเซิร์ฟเวอร์เพื่อใช้งานจริง และสำหรับนักเรียน-นักศึกษาที่ต้องการศึกษาการใช้งานเซิร์ฟเวอร์บนระบบปฏิบัติการลินุกซ์ ผลงานนี้เป็นการพัฒนาต่อยอดมาจาก Ubuntu Server 9.04 สามารถติดตั้งผ่านทางแผ่นดีวีดีและทาง USB Thumb Drive (ที่มีขนาดตั้งแต่ 4 GB) ซึ่งผู้ใช้เพียงดาวน์โหลดไฟล์ ISO แล้วนำไปเขียนใส่ใน USB Thumb Drive ตามคู่มือที่เขียนบอกไว้ที่เว็บไซต์ ก็จะสามารถนำ USB Thumb Drive นั้นไปบูตและทำการติดตั้งระบบบนเครื่องต่าง ๆ ที่ต้องการทำเป็นเซิร์ฟเวอร์

เวอร์ชัน 5207 นี้ใช้ grub เป็นตัวบูตระบบ ซึ่งในตอนติดตั้งจะสร้างเมนูบูตระบบใหม่ขึ้น 2 เมนูคือเมนูแรกจะบูตเข้า GUI โหมด ส่วนเมนูที่สองจะบูตเข้าสู่ Text โหมด ถ้าเข้า Text โหมดจะใช้ RAM น้อยกว่าโหมด GUI ประมาณ 200 MB. เนื่องจากผู้ใช้เกือบทั้งหมดได้ติดตั้งระบบปฏิบัติการวินโดวส์อยู่ก่อนแล้ว จึงได้กำหนดการบูตเข้าสู่ระบบปฏิบัติการวินโดวส์เป็นค่าปริยาย ดังนั้นในเมนูแรกของการบูตจึงชี้ไปที่ระบบปฏิบัติการวินโดวส์ ซึ่งสามารถเปลี่ยนการบูตได้ที่ไฟล์ boot/grub/grub.cfg

สำหรับเครื่องคอมพิวเตอร์รุ่นใหม่ ๆ คือรุ่นที่มี CPU ของ Intel ตั้งแต่รุ่น Pentium 4 ความเร็ว 3.0 GHz ขึ้นไปหรือผลิตรหลังจากนั้น เช่น CPU ที่เป็น Dual Core และ Core 2 Dual เป็นต้น สามารถติดตั้งระบบที่เป็น 64 บิตได้ ทั้งนี้โปรแกรมติดตั้งจะตรวจสอบและให้ท่านเลือกการติดตั้งที่ต้องการและจากประสบการณ์พบว่าระบบที่เป็น 64 บิตมีสมรรถนะดีกว่าแบบ 32 บิต

การเริ่มต้นใช้งาน

1. การติดตั้งต้องการเนื้อที่ประมาณ 6 GB. ทั้งนี้ขึ้นอยู่กับโหมดการติดตั้งและมี swap พาร์ทิชันแล้วหรือไม่
2. บูตเครื่องด้วยแผ่น SUTinsServer แล้วเข้าสู่โปรแกรมติดตั้ง
3. ก่อนที่จะติดตั้งระบบ ควรใช้เมนู “ตรวจสอบแผ่น” ก่อน
4. ใช้เมนู “ติดตั้งระบบ”
5. เมื่อต้องการเลิกใช้งาน สามารถใช้เมนู “ถอนระบบ” โดยการบูตแผ่นติดตั้งนี้ ถ้าติดตั้งแบบทั้งไดรฟ์เป็น Ext3 โปรแกรมจะทำการฟอร์แมตไดรฟ์นั้นกลับคืน เพื่อให้สามารถใช้ในระบบปฏิบัติการวินโดวส์ได้ต่อไป

สิ่งที่ควรทราบคือ

1. เว็บอยู่ที่ /var/www/html
2. สำหรับผู้ชำนาญ การกำหนดให้ start/stop โปรแกรมที่รันอัตโนมัติพร้อมกับการบูตระบบสามารถกำหนดโดยใช้คำสั่ง `sudo sysv-rc-conf` ในเทอร์มินัล
3. เมื่อต้องการเปลี่ยนสิทธิ์ให้โปรแกรมบนเว็บอ่านและเขียนไฟล์ได้ให้ใช้คำสั่ง `sudo chmod 777 -R dirname` ในเทอร์มินัล
4. ไฟล์ config ของ Apache อยู่ที่ /etc/apache2/apache2.conf และ /etc/apache2/sites-enabled/000-default
5. ไฟล์ config ของ PHP อยู่ที่ /etc/php5/apache2/php.ini
6. เมื่อต้องการ start, stop หรือ restart โปรแกรมในขณะนั้น ๆ เช่นเมื่อต้องการ restart โปรแกรม Apache ให้ใช้คำสั่ง `sudo service apache2 restart` เป็นต้น
7. ถ้าต้องการให้ FTP Server ทำงาน ให้ใช้คำสั่ง `sudo service vsftpd start`
8. ถ้าต้องการดู IP ของเครื่อง ให้ใช้คำสั่ง `ifconfig`
9. ล็อกอินเข้าสู่ระบบเป็นผู้ใช้งาน ชื่อผู้ใช้คือ sut รหัสผ่านคือ a
10. ถ้าต้องการเปลี่ยนรหัสผ่านของผู้ใช้ sut ให้ทำดังนี้
 - เปิดเทอร์มินัล
 - รันคำสั่ง `sudo su -`
 - ป้อนรหัสผ่านเดิมของผู้ใช้ sut
 - รันคำสั่ง `passwd sut`
 - ป้อนรหัสผ่านใหม่ของผู้ใช้ sut สองครั้ง
11. การเปลี่ยนภาษาของคีย์บอร์ดใช้ Alt+Shift

12. ใช้ webalizer เพื่อดูสถิติการเข้าชมเว็บ โดยรันคำสั่ง `sudo webalizer` ในเทอร์มินัล จากนั้นใช้เบราว์เซอร์ต่อเข้ามาที่ `http://localhost/webalizer/` ส่วนไฟล์ config ของ webalizer คือ `/etc/webalizer/webalizer.conf` ถ้าต้องการเปลี่ยนการเข้าถึง ให้แก้ไขไฟล์ `/etc/apache2/sites-enabled/000-default`

13. สำหรับโปรแกรม phpMyAdmin อยู่ที่ `/usr/share/myadmin` สามารถรันโดยใช้เมนูบน Firefox ที่ Utilities --> phpMyAdmin - 2.11.8.1deb1 แล้วล็อกอินชื่อผู้ใช้คือ root รหัสผ่านคือ meroot

14. เพื่อความปลอดภัย เมื่อต้องการกำหนดให้เรียกใช้โปรแกรมจากเครื่องเซิร์ฟเวอร์เท่านั้น คือที่ localhost หรือ 127.0.0.1 ให้เพิ่มคำสั่งต่อไปนี้เข้าไปในไฟล์

`/etc/apache2/conf.d/phpmyadmin.conf` ในส่วน

```
<Directory .....
    ## added by sompan
    Order deny,allow
    Deny from all
    Allow from localhost 127.0.0.0/255.0.0.0 ::1/128
    ## added by sompan
```

จากนั้นต้องรันโปรแกรมใหม่ด้วยคำสั่ง `sudo service apache2 restart`

สำหรับการรักษาความปลอดภัยของระบบ

1. อ่านไฟล์ `/etc/sudoers`
2. ตามคำปรียายที่ตั้งไว้บนไฟล์วอลล์จะยังไม่ทำงาน การเริ่มต้นใช้งานต้องใช้คำสั่ง `sudo ufw enable`

จากนั้นจึงเพิ่มกฎต่าง ๆ เข้าไปตามรูปแบบซึ่งดูได้จาก `man ufw`. โดยทั้งนี้มียกข้อยกเว้นคือ ทุกครั้งที่เปลี่ยนการเชื่อมต่ออินเทอร์เน็ตเวิร์กการ์ดใหม่ จะต้องใช้คำสั่ง `sudo ufw enable` ทุกครั้งหรืออาจใช้โปรแกรม `setup firewall` ก็ได้ โดยไปที่

เมนูหลัก --> System --> Administration --> Firewall

คู่มือการใช้งาน

คู่มือการใช้งานการเชื่อมต่อและตั้งค่าต่าง ๆ อยู่ในไฟล์ `Doc.tar.gz` ซึ่งประกอบไปด้วย

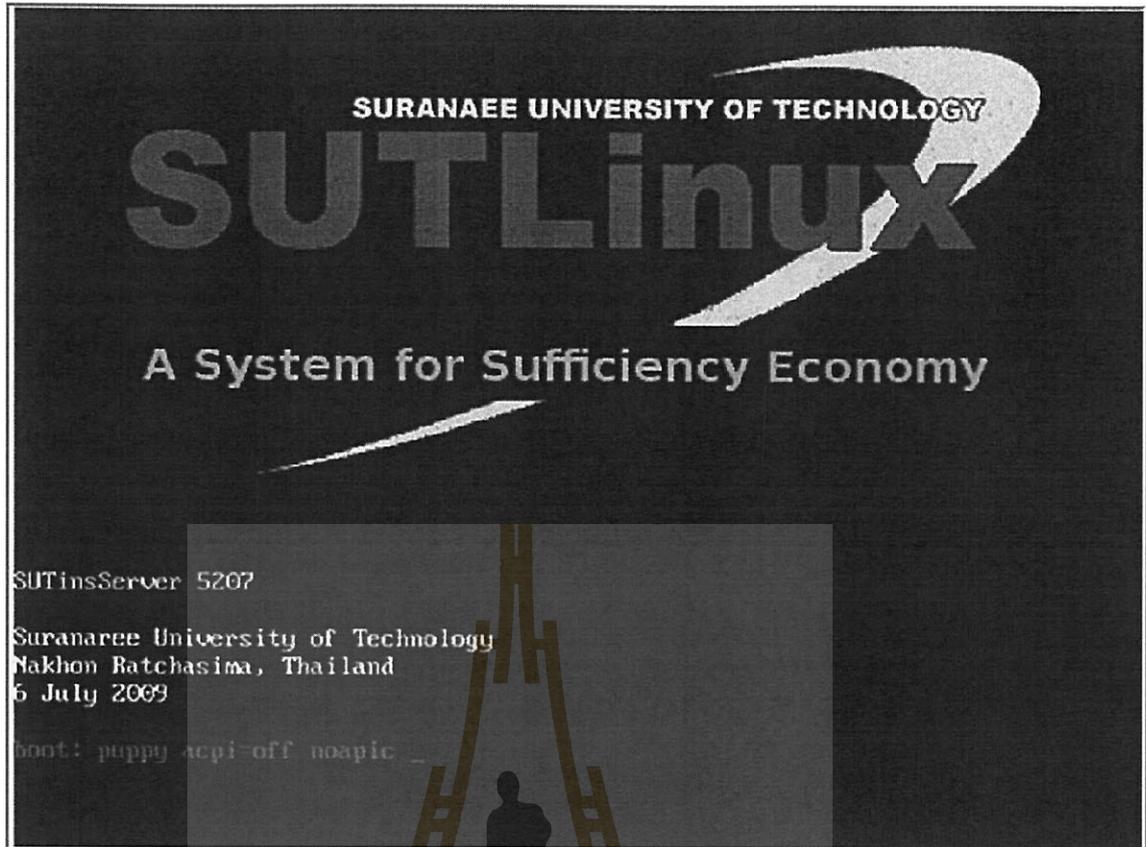
1. การเชื่อมต่อ VMware
2. การเชื่อมต่อ SSH
3. การเชื่อมต่อ FTP
4. การเชื่อมต่อ DHCP
5. การเชื่อมต่อ DNS
6. การเชื่อมต่อ Vhost ใน Apache
7. การเชื่อมต่อ Postfix/Dovecot และ Webmail
8. การเชื่อมต่อ Proxy Server
9. การเชื่อมต่อ IP

10. การทำ Remote Admin
11. การเซ็ตอัฟ Firewall
12. การติดต่อกับอุปกรณ์ถอดเสียบ
13. การทำแผ่น CD/DVD เป็นของตนเอง
14. การเริ่มต้นใช้งาน
15. การทำ MultiWeb
16. การตรวจสอบแผ่นเมื่อบูตจากแผ่นและการตรวจสอบแผ่นเมื่ออยู่บนระบบปฏิบัติการลินุกซ์
17. การเขียนแผ่น CD/DVD
18. การอัฟโหลดไฟล์ไปยังเซิร์ฟเวอร์
19. การเซ็ตอัฟ SSL
20. การติดตั้ง USB Flash Drive

การติดตั้ง SUTinsServer 5207

เมื่อมีแผ่นโปรแกรม SUTinsServer 5207 แล้วต้องการเริ่มต้นติดตั้ง ก็สามารถทำได้ง่าย โดยสามารถจัดการกับระบบต่าง ๆ เพื่อเตรียมความพร้อมให้กับระบบในการที่จะติดตั้ง SUTinsServer ต่อไปนี้ ซึ่งขั้นตอนและวิธีการติดตั้งที่จะนำเสนอต่อไปนี้ เป็นข้อมูลและรูปภาพประกอบที่ดาวน์โหลดมาจากเว็บไซต์ <http://linux.sut.ac.th> ซึ่งจะมีข้อแตกต่างจากการติดตั้งในเครื่องอื่น ๆ บ้างเล็กน้อย

1. เมื่อได้พื้นที่สำหรับการติดตั้ง SUTinsServer เป็นที่เรียบร้อยแล้ว ก็จะมาถึงขั้นตอนการติดตั้ง ซึ่งสามารถทำได้โดยใส่แผ่นดีวีดี (ซึ่งมาพร้อมกับหนังสือเล่มนี้ แต่ถ้ายังไม่มีให้ไปดาวน์โหลดได้จาก linux.sut.ac.th เมื่อได้ไฟล์ iso มาแล้วให้ใช้โปรแกรมเขียนบนแผ่นดีวีดีโดยใช้โหมดเขียน Disc image) แล้วบูตระบบขึ้นใหม่โดยเลือกให้บูตจากแผ่นดีวีดี ก็จะสามารถเข้าสู่โปรแกรมการติดตั้ง SUTinsServer 5207 ได้ เมื่อบูตแผ่นแล้ว จะปรากฏหน้าต่างดังแสดงในรูปที่ ข1 จากรูปจะเห็นว่าได้บ่อนข้อความ puppy acpi=off noapic ต่อจาก boot: ซึ่งจะทำเช่นนี้เฉพาะเครื่องที่มีปัญหาการบูตเท่านั้น ถ้าเครื่องท่านไม่มีปัญหาไม่จำเป็นต้องบ่อนข้อความดังกล่าว



รูปที่ ข1: หน้าต่างแรกเมื่อบูตจากแผ่น SUTinsServer

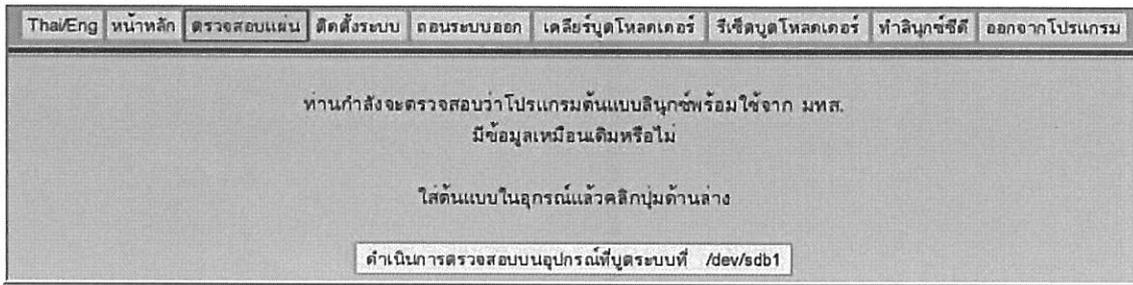
2. เมื่อบูตสำเร็จจะเข้าสู่หน้าต่างดังแสดงในรูปที่ ข2



รูปที่ ข2: หน้าต่างแสดงตัวเลือกกราฟิก

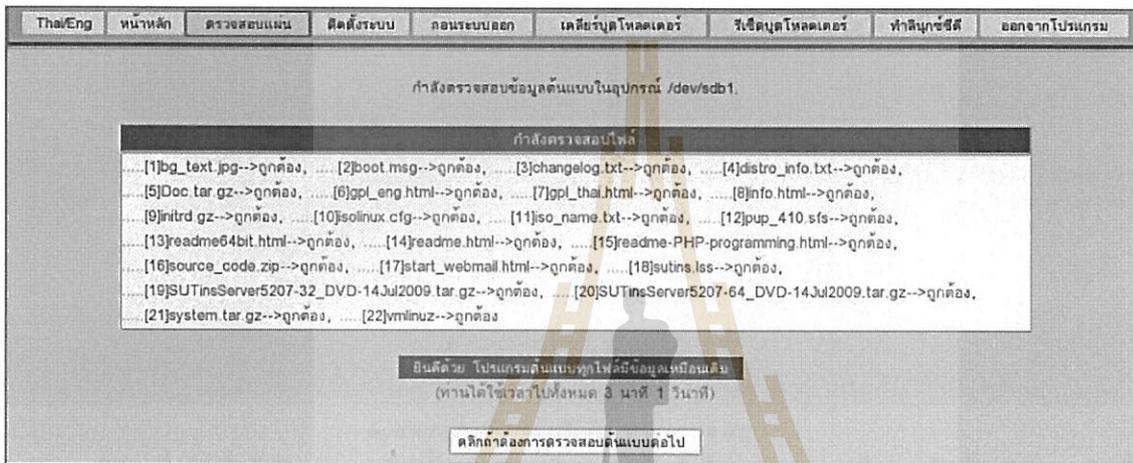
3. เลือกเข้าระบบโดยเลือก Xorg ถ้าเข้าไม่ได้ให้ reboot แล้วเลือกเข้า Xvesa แทน ต่อจากนั้นจะปรากฏหน้าต่างดังแสดงในรูปที่ ข3 (ในขั้นตอนก่อนที่จะเข้าสู่โปรแกรมหลักที่อยู่ในโหมดกราฟิก จะมีการตรวจสอบอุปกรณ์กราฟิกการ์ดและจอภาพ ซึ่งบางครั้งใช้เวลานานมาก โดยเฉพาะจอภาพแบบ LCD จะใช้เวลานานเป็นพิเศษ อาจถึง 2-3 นาที)

4. ทุกครั้งก่อนที่จะทำการติดตั้งระบบ ควรทำการตรวจสอบแผ่นก่อน โดยให้เลือกที่ “ตรวจสอบแผ่น” แล้วกดปุ่ม “ดำเนินการตรวจสอบบนอุปกรณ์ซีดีรอมที่ /dev/sdb1” ที่อยู่ด้านล่าง



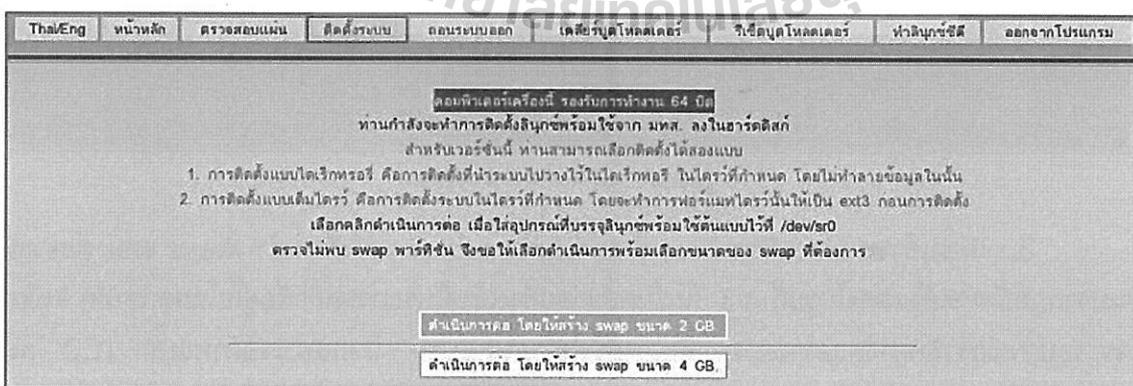
รูปที่ ข3: หน้าต่างแสดงการเลือกปุ่ม “ตรวจสอบแผ่น”

เมื่อตรวจสอบแผ่นเรียบร้อยแล้วจะปรากฏหน้าต่างดังแสดงในรูปที่ ข4

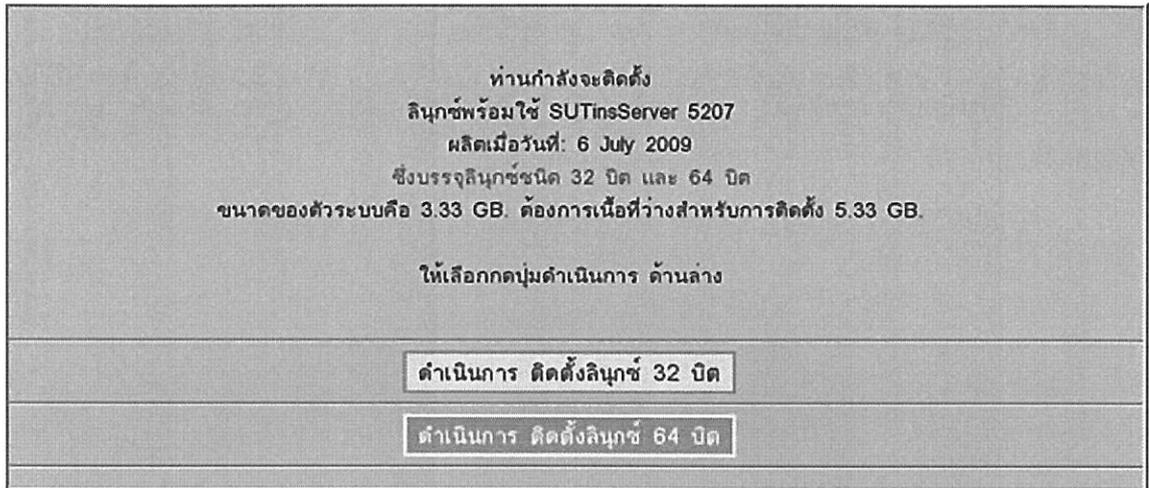


รูปที่ ข4: หน้าต่างแสดงการตรวจสอบแผ่นสำเร็จ

5. จากนั้นจะเข้าสู่กระบวนการติดตั้ง โดยให้เลือกที่ “ติดตั้งระบบ” (ดังแสดงในรูปที่ ข5) ส่วนแรกจะแสดงข้อความรายละเอียดต่าง ๆ โดยส่วนบนสุดจะบอกว่าคอมพิวเตอร์ของท่านสามารถรับคำสั่ง 64 บิตได้หรือไม่ และส่วนล่างสุด ท่านต้องเลือกว่าจะให้ตัวติดตั้งสร้าง swap ให้มีขนาดเท่าใด สำหรับการทดลองใช้ 2GB ก็เพียงพอ แต่ถ้าต้องการนำไปใช้งานเป็นเครื่องเซิร์ฟเวอร์จริงๆ ควรเลือก 4GB จากนั้นจะปรากฏหน้าต่างดังแสดงในรูปที่ ข6 ซึ่งท่านต้องเลือกว่าจะติดตั้งระบบที่เป็น 32 หรือ 64 บิต

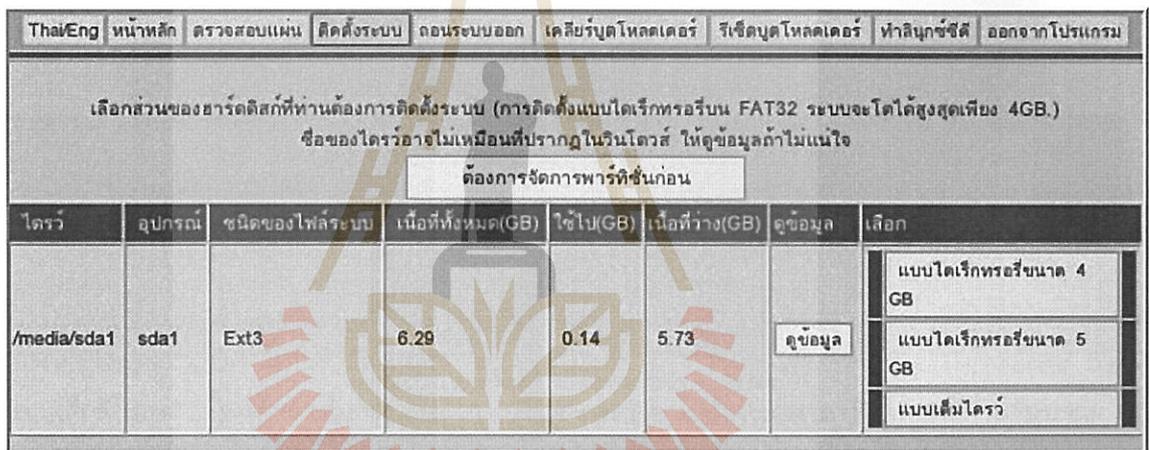


รูปที่ ข5: หน้าต่างแสดงปุ่มเลือกขนาดของไฟล์ swap



รูปที่ ๖: หน้าต่างแสดงปุ่มให้เลือกระบบที่เป็น 32 หรือ 64 บิต

6. จากนั้นโปรแกรมจะทำการตรวจสอบหาเนื้อที่ว่างเพื่อจะติดตั้งต่อไป ซึ่งจะปรากฏหน้าต่างดังแสดงในรูปที่ ๗



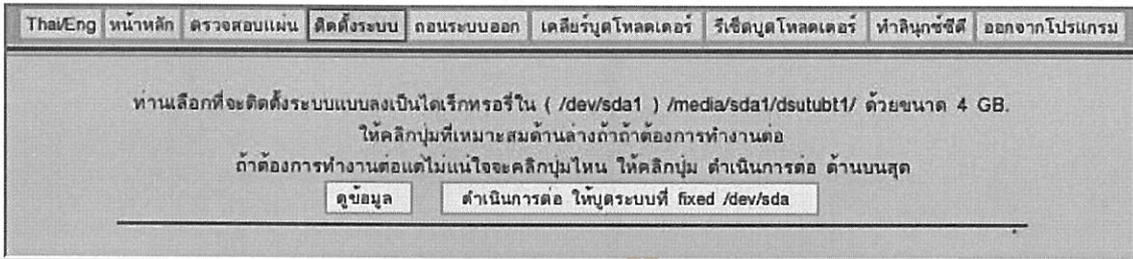
รูปที่ ๗: หน้าต่างแสดงตำแหน่งและรูปแบบที่ต้องการลงระบบ SUTinsServer

7. ในรูปที่ ๗ นั้น (จะสังเกตเห็นว่าในเครื่องคอมพิวเตอร์ที่กำลังจะติดตั้งนี้ มีไดรว์เพียงไดรว์เดียวแสดงให้เห็นซึ่งเป็น Ext3 ที่เป็นเช่นนี้เพราะว่า การจับหน้าจอของการลงระบบ SUTinsServer ต้องใช้โปรแกรมพิเศษช่วยและได้จำลองเนื้อที่บางส่วนมาทดลองทำการติดตั้ง เพื่อให้แสดงหน้าต่างของการติดตั้งในทุกขั้นตอนคล้ายกับของจริงมากที่สุด ดังนั้น เมื่อท่านติดตั้งระบบดังกล่าวในเครื่องของท่านเอง หน้าต่างในรูปที่ ๗ นี้จะแตกต่างออกไป เช่น ถ้าเครื่องคอมพิวเตอร์ของท่านมี 3 พาร์ติชัน ก็จะมีตัวเลือกทั้ง 3 พาร์ติชัน เป็นต้น) มีปุ่ม “ต้องการจัดการพาร์ติชันก่อน” เพื่อช่วยในการจัดการพาร์ติชัน ส่วนทางด้านซ้ายมือ นั้นจะมีปุ่มตัวเลือกการติดตั้ง 3 แบบคือ

- ปุ่ม “แบบไดเรกทอรี 4 GB” จะติดตั้งอยู่ในไดเรกทอรีบนระบบปฏิบัติการวินโดวส์ใช้เนื้อที่ 4 GB
- ปุ่ม “แบบไดเรกทอรี 5 GB” จะติดตั้งอยู่ในไดเรกทอรีบนระบบปฏิบัติการวินโดวส์ใช้เนื้อที่ 5 GB โดยเพิ่มเนื้อที่จากแบบ 4 GB มาอีก 1 GB ทั้งนี้ก็เพื่อผู้ใช้งานที่ต้องการดาวน์โหลดหรืออัปเดตโปรแกรมต่าง ๆ จะได้มีเนื้อที่พอเพียง

- ปุ่ม “แบบเต็มไดรว์” จะทำการ Format ไดรว์ที่ต้องการติดตั้งให้เป็น Ext3 แล้วลงระบบ

ในที่นี้จะขอเลือกการติดตั้ง “แบบไดเร็คทอรี 4 GB” ซึ่งเมื่อกดปุ่มดังกล่าวแล้วจะปรากฏหน้าต่างดังแสดงในรูปที่ ๗8



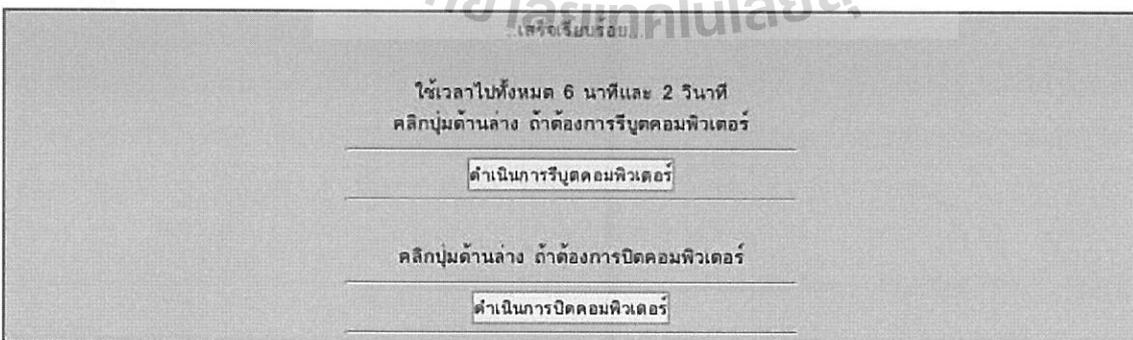
รูปที่ ๗8: หน้าต่างแสดงรายละเอียดและตำแหน่งที่เลือกลงระบบ SUTinsServer

8. ในรูปที่ ๗8 นั้น โปรแกรมจะอธิบายรายละเอียดต่าง ๆ ของระบบ พร้อมกับปุ่ม “ดูข้อมูล” เพื่อดูข้อมูลในไดรว์ที่เลือกติดตั้งและปุ่ม “ดำเนินการต่อ ให้บูตระบบที่ fixed /dev/sda” เพื่อดำเนินการติดตั้งต่อไป เมื่อกดปุ่ม “ดำเนินการต่อ ให้บูตระบบที่ fixed /dev/sda” จะปรากฏหน้าต่างดังแสดงในรูปที่ ๗9



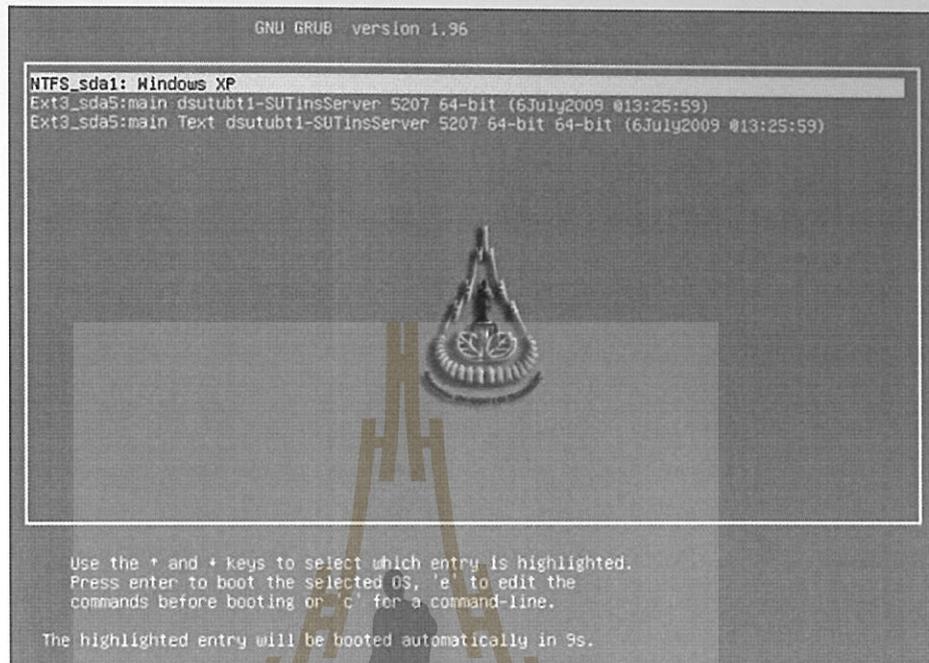
รูปที่ ๗9: หน้าต่างแสดงขณะกำลังติดตั้งระบบ SUTinsServer

9. ในรูปที่ ๗9 นั้น โปรแกรมจะแสดงรายละเอียดต่าง ๆ ในการติดตั้งแล้วจะปรากฏหน้าต่างดังแสดงในรูปที่ ๗10



รูปที่ ๗10: หน้าต่างแสดงการติดตั้งระบบ SUTinsServer เสร็จเรียบร้อยแล้ว

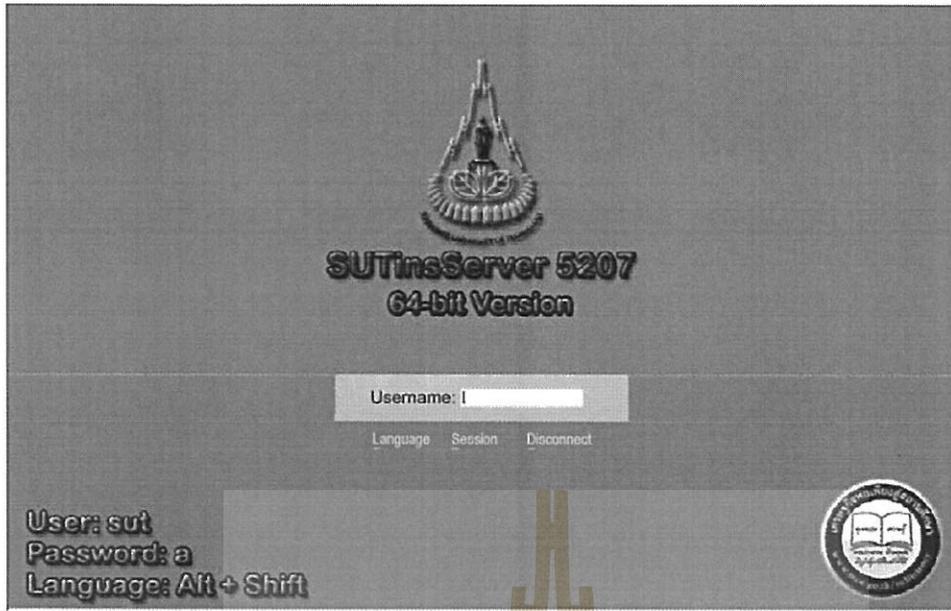
10. ระบบจะแสดงเวลาทั้งหมดที่ใช้ในการติดตั้ง ให้เอาแผ่นดีวีดีออกจาก DVD-ROM ก่อน เพื่อไม่ให้มีการรันเข้าสู่แผ่นอีกครั้ง แล้วให้เลือกกดปุ่ม “ดำเนินการรีบูตคอมพิวเตอร์” เพื่อทำการรีบูตเครื่องคอมพิวเตอร์ ก็จะปรากฏหน้าต่างดังแสดงในรูปที่ ข11



รูปที่ ข11: หน้าต่างแสดงตัวเลือกเข้าสู่ระบบต่าง ๆ

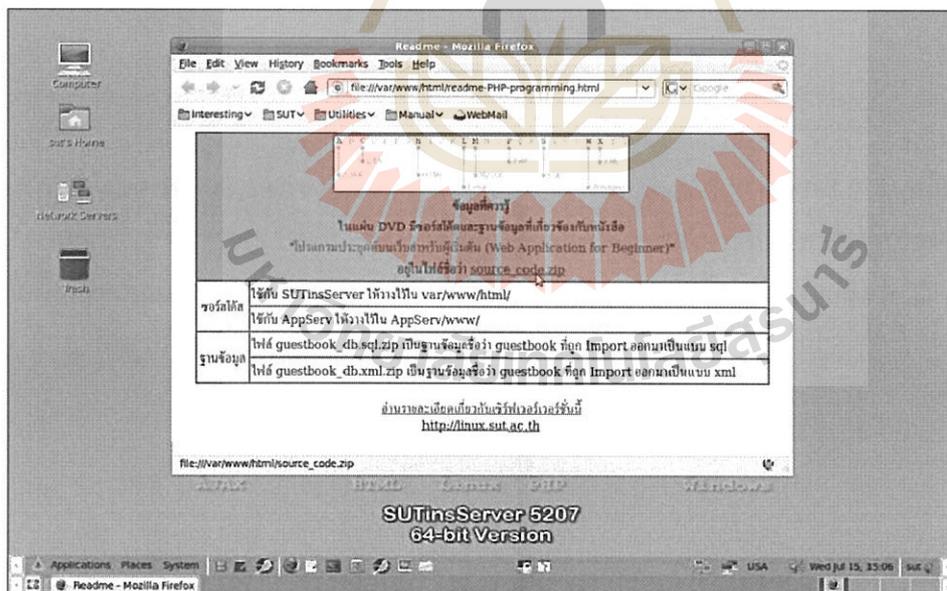
11. ในรูปที่ ข11 นั้นจะเห็นได้ว่ามีตัวเลือกการใช้งานระบบเพียง 3 ตัวเลือกเท่านั้น คือตัวเลือกที่ 1 “การเลือกเข้าสู่ระบบ Windows” ซึ่งถูกเลือกไว้เป็นค่าปริยาย ตัวเลือกที่ 2 “การเลือกเข้าสู่ระบบ SUTinsServer 5207 แบบกราฟิกโหมด” ส่วนตัวเลือกที่ 3 “การเลือกเข้าสู่ระบบ SUTinsServer 5207 แบบเท็กซ์โหมด” โดยถ้าท่านไม่กดปุ่มใด ๆ เพื่อทำการเลือกในช่วงระยะเวลาที่กำหนดไว้ (ประมาณ 10 วินาที) ก็จะเข้าสู่ระบบปฏิบัติการวินโดวส์เองโดยอัตโนมัติ

12. ถ้าเลือกตัวเลือกที่ 2 “การเลือกเข้าสู่ระบบ SUTinsServer 5207 แบบกราฟิกโหมด” จากรูปที่ ข11 จะเข้าสู่ระบบปฏิบัติการลินุกซ์เซิร์ฟเวอร์ซึ่งมีหน้าต่างให้ล็อกอิน ดังแสดงในรูปที่ ข12 โดยมี user name คือ sut และ password คือ a



รูปที่ ข12: หน้าต่างล็อกอินเข้าสู่ระบบปฏิบัติการลินุกซ์เซิร์ฟเวอร์

13. เมื่อล็อกอินได้สำเร็จจะเข้าสู่หน้าต่างแรกของลินุกซ์เซิร์ฟเวอร์ 5207 ดังแสดงในรูปที่ ข13 ซึ่งโปรแกรมเว็บเบราว์เซอร์ Firefox จะแสดงข้อมูลต่าง ๆ ที่ผู้ใช้ควรรู้พร้อมกับจุดเชื่อมต่อที่สามารถดาวน์โหลดซอร์สโค้ดทั้งหมดเกี่ยวกับหนังสือเล่มนี้ จุดเชื่อมต่อไปยังเว็บไซต์ของลินุกซ์ มทส. และจุดเชื่อมต่อไปดูรายละเอียดเกี่ยวกับเซิร์ฟเวอร์ 5207 พร้อมคู่มือการใช้งานด้านต่าง ๆ

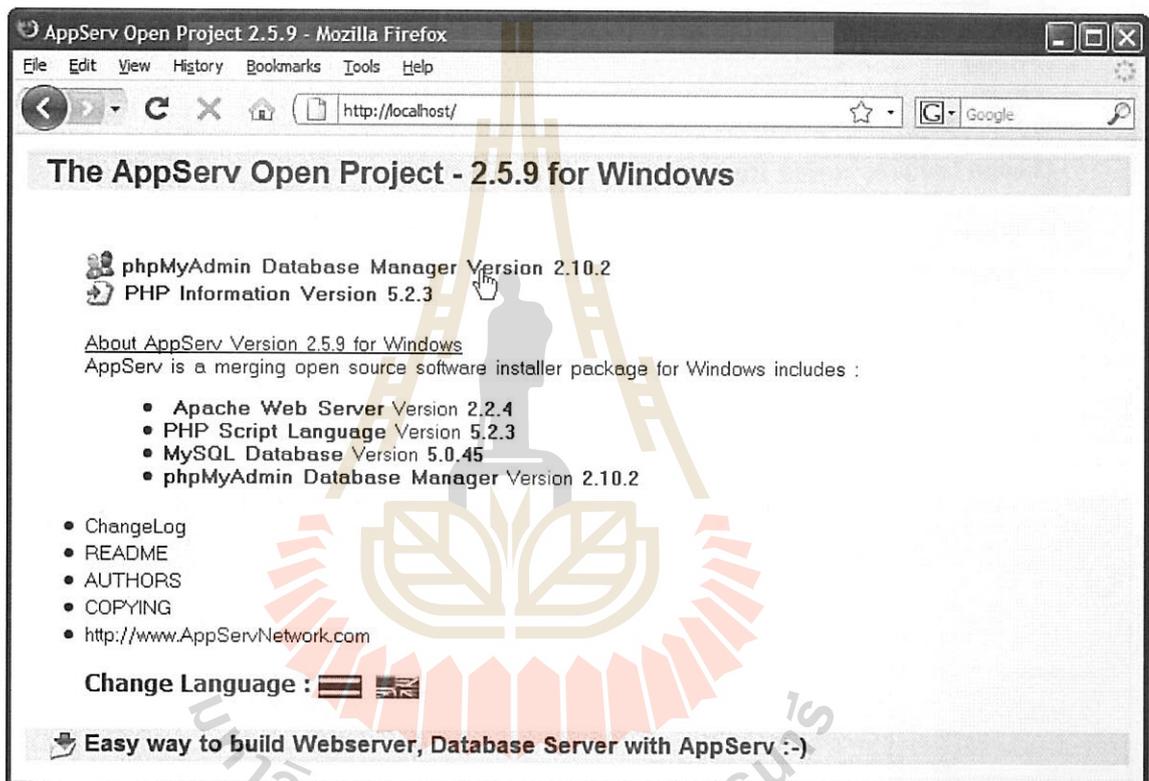


รูปที่ ข13: หน้าต่างแรกของลินุกซ์เซิร์ฟเวอร์

ภาคผนวก ค

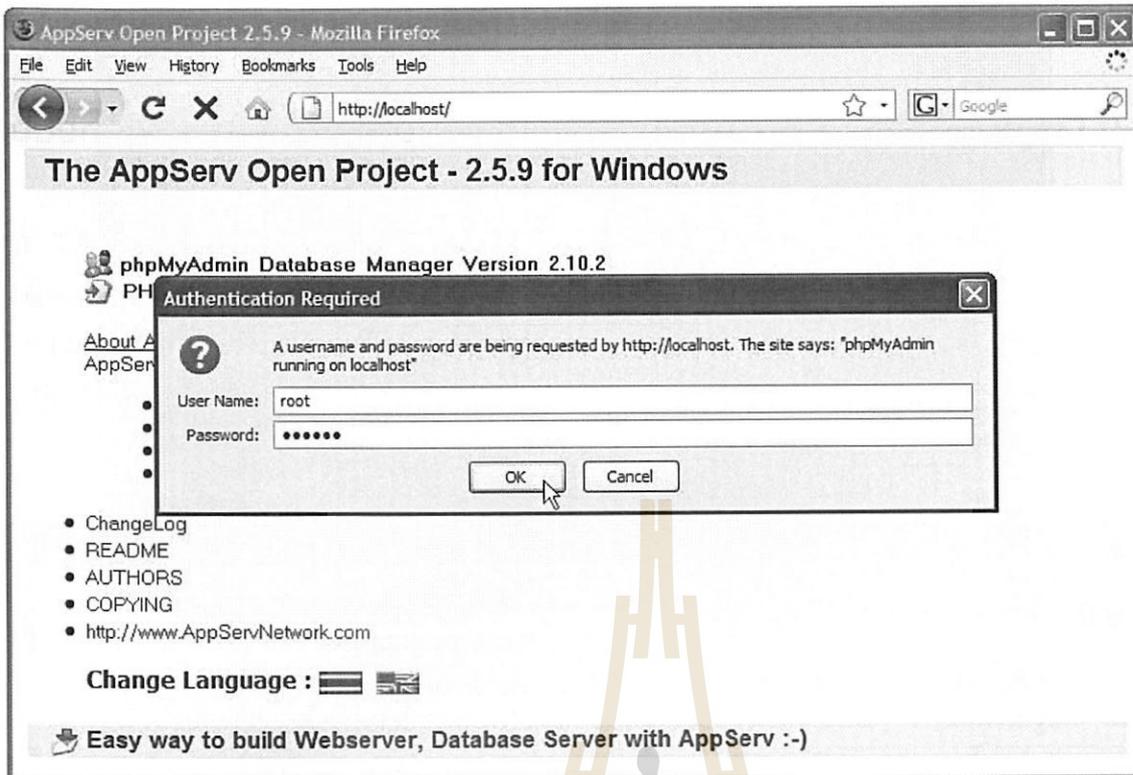
การใช้งานโปรแกรม phpMyAdmin <phpMyAdmin's Manual>

การจัดการฐานข้อมูลสามารถทำได้โดยผ่าน Command Prompt (ดังที่กล่าวในบทที่ 8.2 การติดตั้งระบบฐานข้อมูล MySQL ผ่าน Command Prompt) และผ่านโปรแกรมสำเร็จรูป phpMyAdmin ที่มาพร้อมกับ AppServ 2.5.9 และ SUTinsServer 5112 มีลักษณะการทำงานได้ตอบกับผู้ใช้แบบกราฟิก (GUI: Graphical User Interface) ซึ่งง่ายและสะดวกต่อการจัดการ โดยเรียกใช้งานผ่านหน้าเว็บ (http://localhost) ดังแสดงในรูปที่ ค1



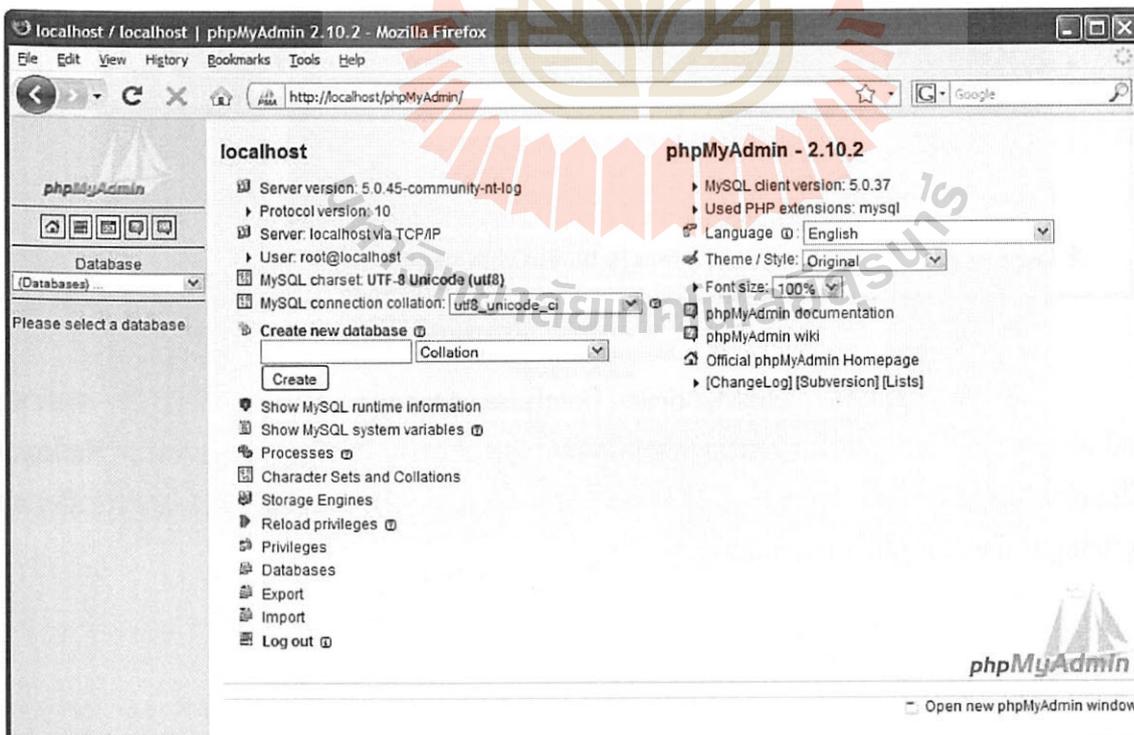
รูปที่ ค1: หน้าต่างเริ่มต้นการใช้งาน phpMyAdmin

เมื่อกดที่จุดเชื่อมต่อ “phpMyAdmin Database Manager Version 2.10.2” จะปรากฏหน้าต่างถามชื่อผู้ใช้และรหัสผ่านเพื่อเข้าสู่ระบบการจัดการฐานข้อมูล ซึ่งชื่อผู้ใช้และรหัสผ่านก็คือข้อมูลที่ป้อนเข้าไปในตอนติดตั้ง AppServ (ดังแสดงในรูปที่ ก6: หน้าต่างแสดงการตั้งค่าการเข้าถึงระบบฐานข้อมูล) ก็จะปรากฏหน้าต่างดังแสดงในรูปที่ ค2



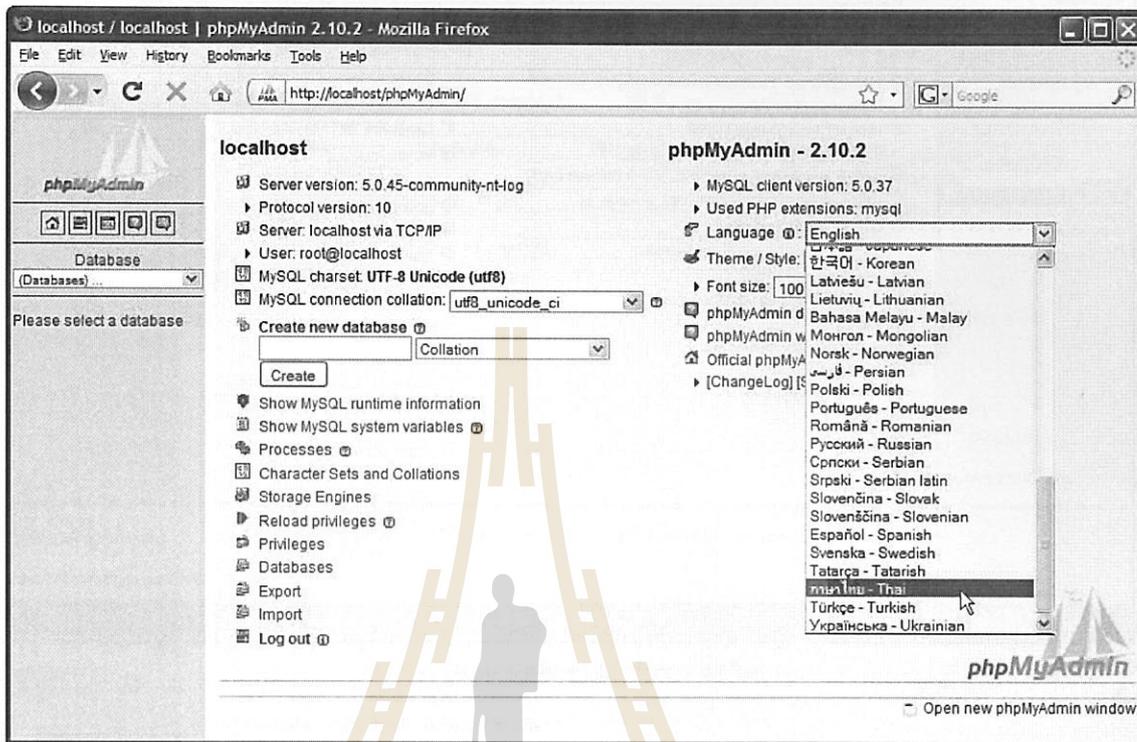
รูปที่ ค2: หน้าต่างป้อนชื่อและรหัสผ่านเข้าสู่ระบบการจัดการฐานข้อมูล

เมื่อป้อนชื่อและรหัสผ่านแล้วกดปุ่ม “OK” จะปรากฏหน้าต่างดังแสดงในรูปที่ ค3 แต่ถ้าชื่อหรือรหัสผ่านไม่ถูกต้อง หน้าต่างป้อนชื่อและรหัสผ่านก็จะปรากฏขึ้นมาให้ใส่ข้อมูลใหม่

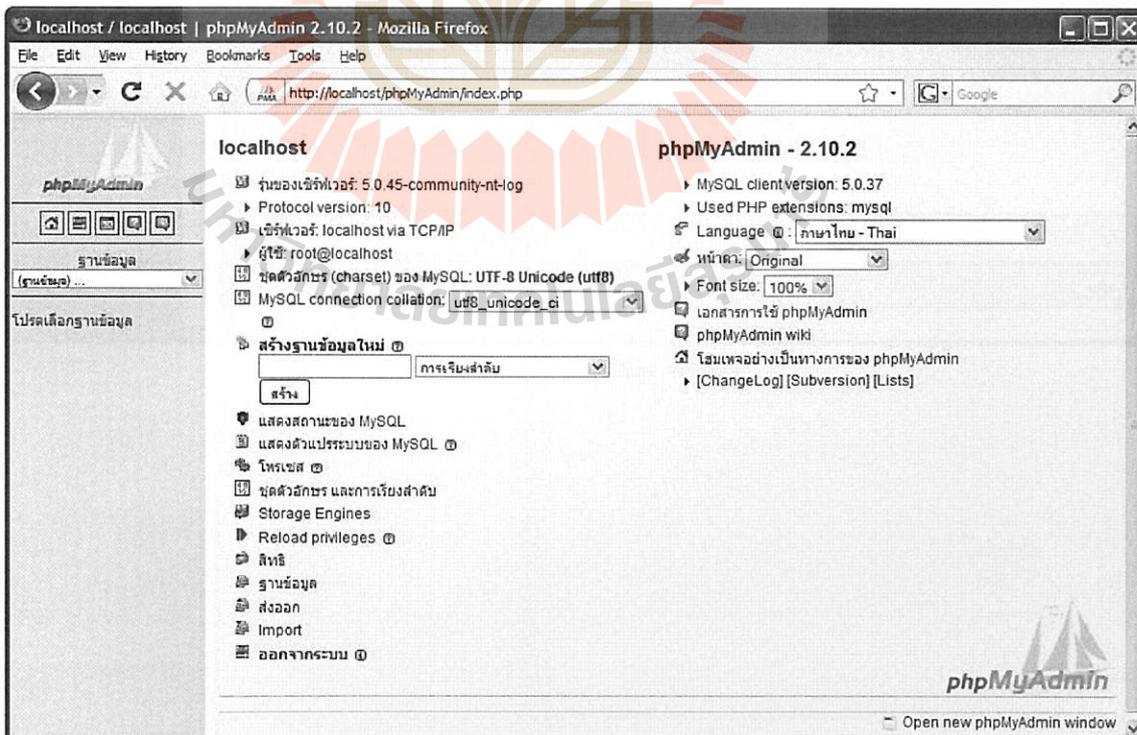


รูปที่ ค3: หน้าต่างเริ่มต้นของโปรแกรมการจัดการฐานข้อมูล phpMyAdmin

จะสังเกตเห็นว่าการแสดงผลของรายละเอียดและคำสั่งต่าง ๆ ถูกแสดงออกเป็นภาษาอังกฤษ ซึ่งสามารถเปลี่ยนให้เป็นภาษาไทยได้ โดยการกดปุ่มทำข้อความ “Language” แล้วเลือก “ภาษาไทย” ดังแสดงในรูปที่ ค4 ก็จะมีปรากฏหน้าต่างแสดงเป็นภาษาไทย ดังแสดงในรูปที่ ค5

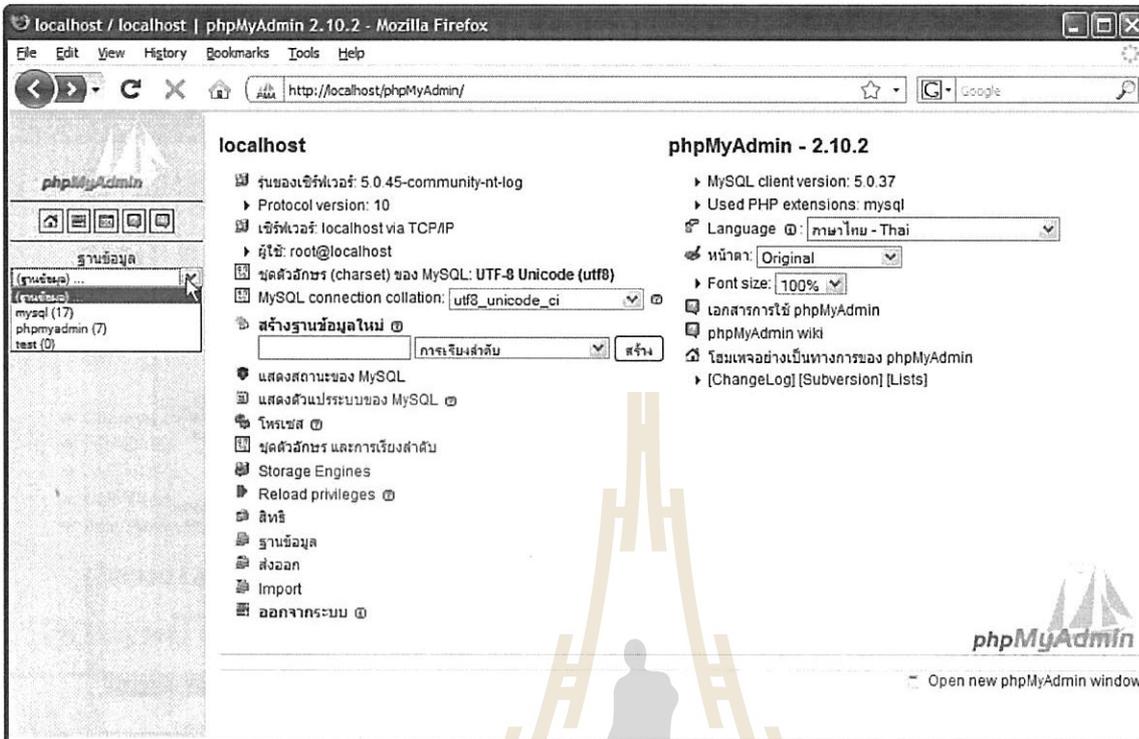


รูปที่ ค4: หน้าต่างแสดงการเปลี่ยนการแสดงผลจากภาษาอังกฤษเป็นภาษาไทย



รูปที่ ค5: หน้าต่างแสดงรายละเอียดและคำสั่งต่าง ๆ เป็นภาษาไทย

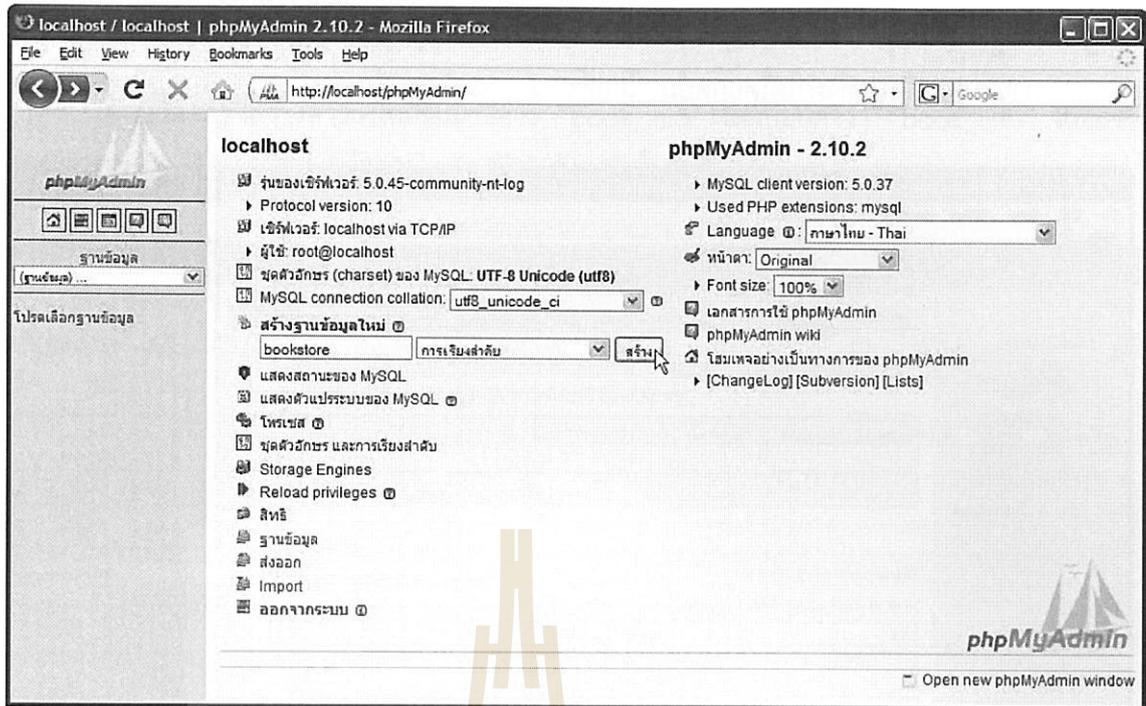
เมื่อกดปุ่มทางด้านขวาของฐานข้อมูล ดังแสดงในรูปที่ ค6 ก็จะมีฐานข้อมูลที่มีอยู่ในระบบ



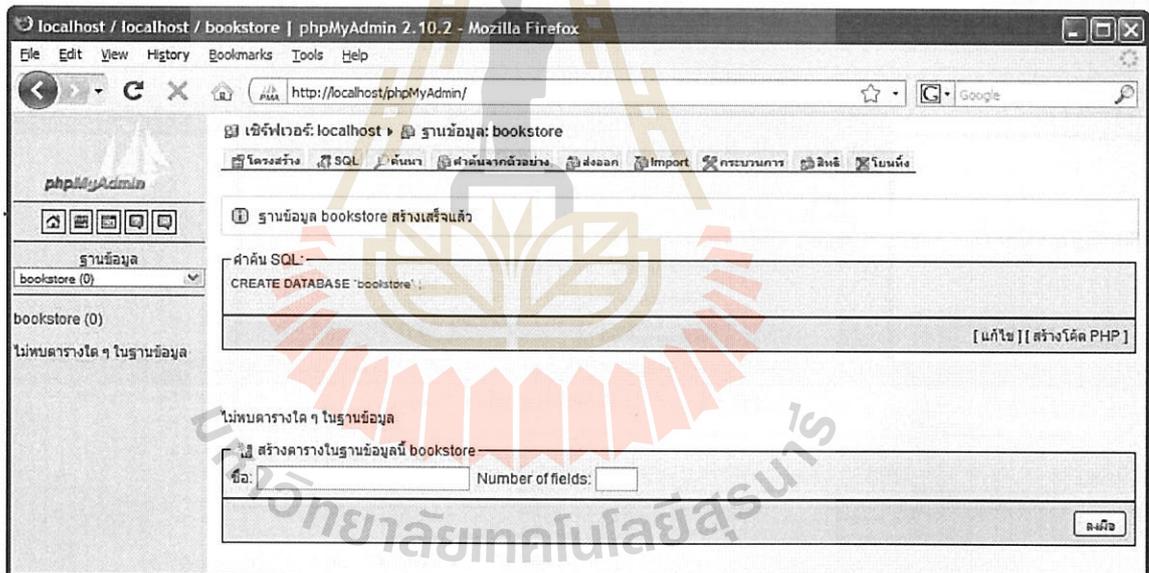
รูปที่ ค6: หน้าต่างแสดงฐานข้อมูลทั้งหมดที่มีอยู่ในระบบ

สร้างฐานข้อมูลใหม่ (Create Database)

ตอนนี้ทดลองสร้างฐานข้อมูลใหม่ชื่อว่า “bookstore” โดยป้อนคำว่า “bookstore” ในช่องสร้างฐานข้อมูลใหม่ แล้วกดปุ่ม “สร้าง” ดังแสดงในรูปที่ ค7 ก็จะปรากฏหน้าต่างฐานข้อมูล “bookstore” ขึ้นมา ดังแสดงในรูปที่ ค8



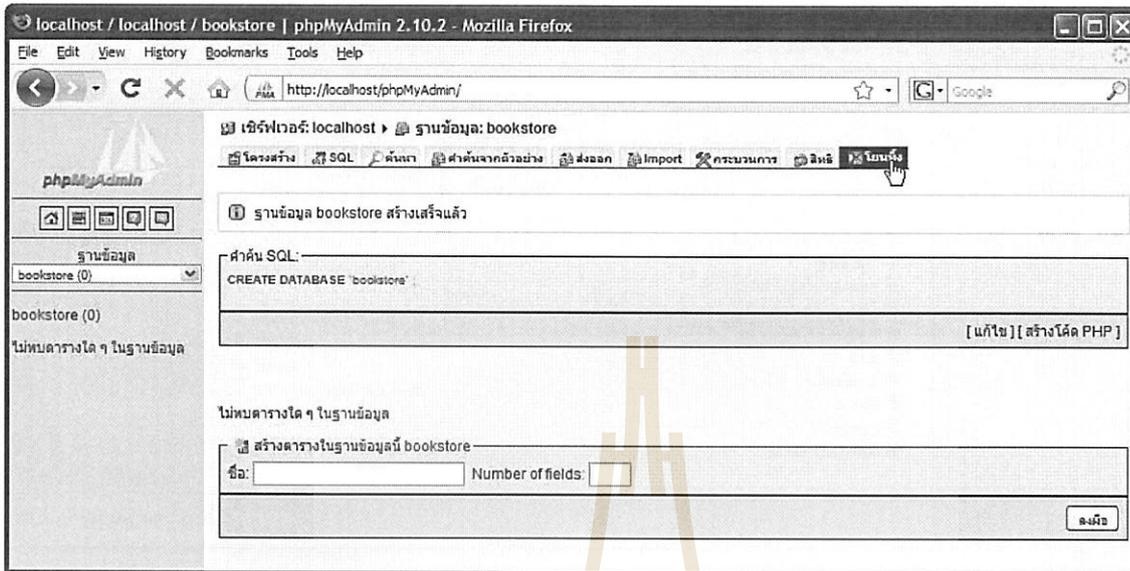
รูปที่ ค7: หน้าต่างแสดงการสร้างฐานข้อมูลใหม่ชื่อว่า "bookstore"



รูปที่ ค8: หน้าต่างแสดงฐานข้อมูล "bookstore"

ลบฐานข้อมูล (Drop Database)

การลบฐานข้อมูลทั้งทำได้โดยกดปุ่ม “โยนทิ้ง” ดังแสดงในรูปที่ ค9



รูปที่ ค9: หน้าต่างแสดงการลบฐานข้อมูล “bookstore”

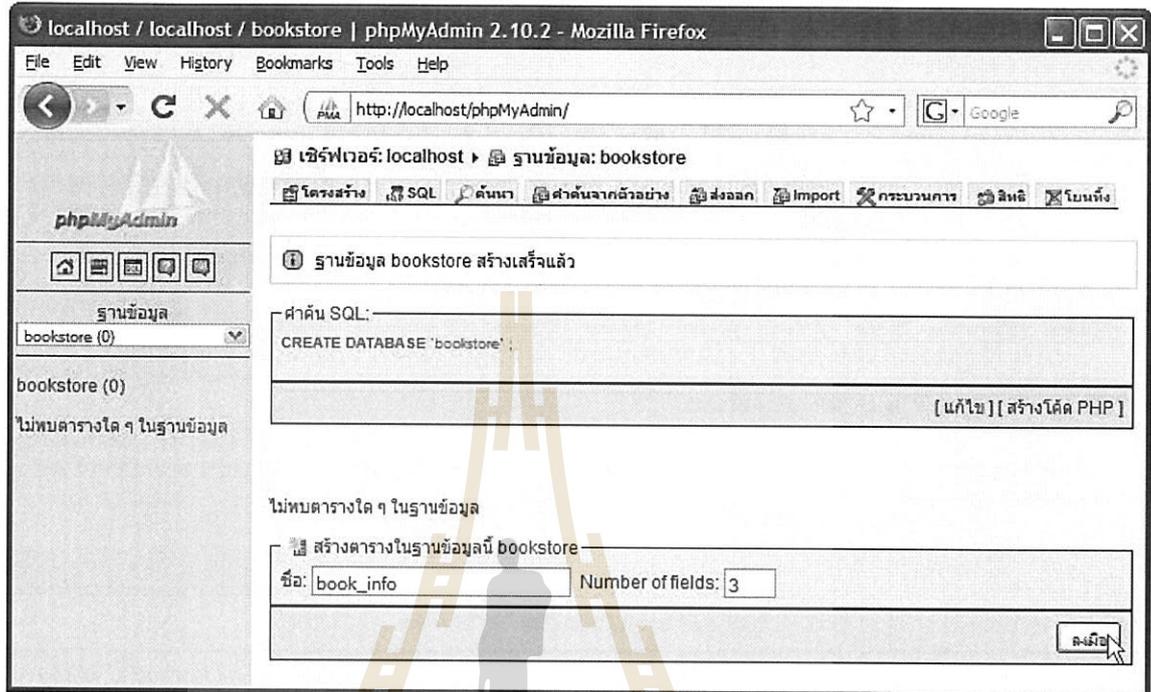
เมื่อกดปุ่ม “โยนทิ้ง” แล้ว หน้าต่างยืนยันการลบฐานข้อมูลจะปรากฏขึ้นดังแสดงในรูปที่ ค10



รูปที่ ค10: หน้าต่างยืนยันการลบฐานข้อมูล “bookstore”

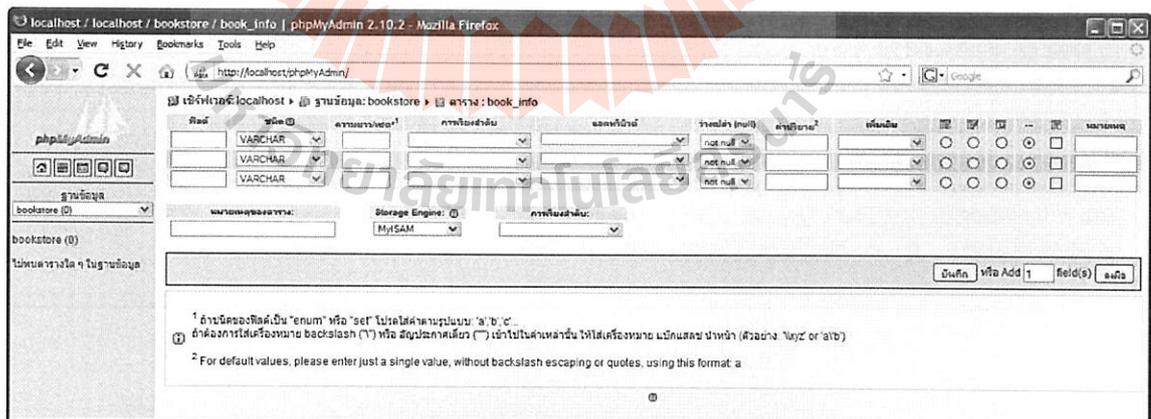
สร้างตารางใหม่ (Create Table)

ตอนนี้ทดลองสร้างตารางใหม่ขึ้นมาชื่อว่า “book_info” โดยป้อนคำว่า “book_info” ในช่องชื่อ และมีจำนวนฟิลด์คือ 3 ฟิลด์ แล้วกดปุ่ม “ลงมือ” ดังแสดงในรูปที่ ค11



รูปที่ ค11: หน้าต่างแสดงการสร้างตาราง “book_info”

เมื่อกดปุ่ม “ลงมือ” แล้ว หน้าต่างแสดงรายละเอียดของตาราง “book_info” ก็จะปรากฏขึ้นดังแสดงในรูปที่ ค12



รูปที่ ค12: หน้าต่างตาราง “book_info”

ทดลองป้อนข้อมูลต่าง ๆ ลงในตาราง “book_info” แล้วกดปุ่ม “บันทึก” ดังแสดงในรูปที่ ค13

ฟิลด์	ชนิด	ความยาว/เขต	การเรียงลำดับ	แอดทริบิวต์	วางเปล่า (null)	ค่าปริยาย	เพิ่มเติม	การตั้งค่า	หมายเหตุ
no	TINYINT	4			not null		auto_increment	<input type="checkbox"/>	
isbn	VARCHAR	20			not null			<input type="checkbox"/>	
name	VARCHAR	50			not null			<input type="checkbox"/>	

Storage Engine: MyISAM

ปุ่ม: บันทึก หรือ Add 1 field(s) สมมติ

รูปที่ ค13: หน้าต่างตาราง “book_info”

เมื่อกดปุ่ม “บันทึก” แล้ว หน้าต่างตาราง “book_info” ก็จะปรากฏขึ้นดังแสดงในรูปที่ ค14

ตาราง 'bookstore`.`book_info' สร้างเสร็จแล้ว

คำสั่ง SQL:

```
CREATE TABLE 'book_info' (
  'no' TINYINT( 4 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  'isbn' VARCHAR( 20 ) NOT NULL ,
  'name' VARCHAR( 50 ) NOT NULL ,
  INDEX ( 'isbn' )
) ENGINE = MYISAM ;
```

[แก้ไข] [สร้างโค้ด PHP]

ฟิลด์	ชนิด	การเรียงลำดับ	แอดทริบิวต์	วางเปล่า (null)	ค่าปริยาย	เพิ่มเติม	การตั้งค่า
no	tinyint(4)			ไม่		auto_increment	<input type="checkbox"/>
isbn	varchar(20)	utf8_general_ci		ไม่			<input type="checkbox"/>
name	varchar(50)	utf8_general_ci		ไม่			<input type="checkbox"/>

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:

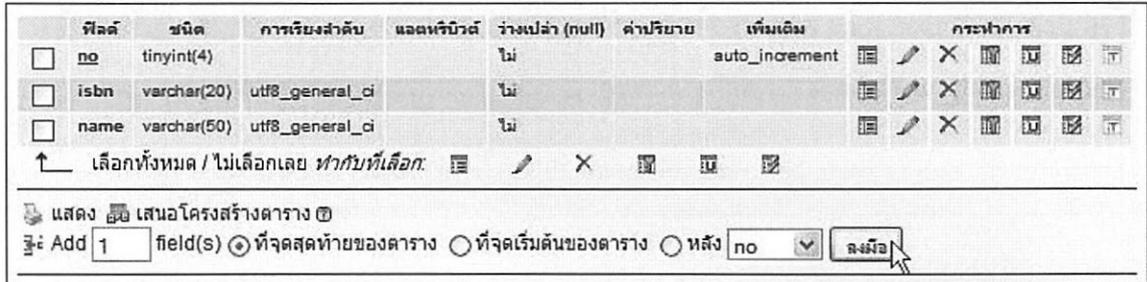
แสดง เสนอโครงสร้างตาราง

เพิ่ม: Add 1 field(s) ที่จุดสุดท้ายของตาราง ที่จุดเริ่มต้นของตาราง หลัง no

ชื่อคีย์	ชนิด	Cardinality	การตั้งค่า	ฟิลด์	ชนิด	ใช้งาน	สถิติของแถว
PRIMARY	PRIMARY	0	<input type="checkbox"/> <input type="checkbox"/>	no	ข้อมูล	0	ไม่
isbn	INDEX	ไม่มี	<input type="checkbox"/> <input type="checkbox"/>	isbn	ดัชนี	0	ไม่
สร้างดัชนีโดยอัตโนมัติ		1	<input type="button" value="สมมติ"/>	รวม		0	ไม่

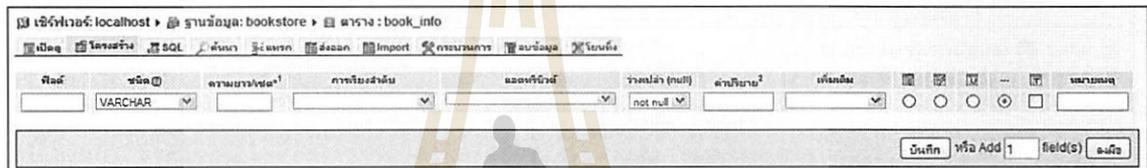
รูปที่ ค14: หน้าต่างตาราง “book_info” เมื่อป้อนรายละเอียดแล้วกดปุ่ม “บันทึก”

เมื่อต้องการเพิ่มฟิลด์อีก 1 ฟิลด์ตรงจุดสุดท้ายของตาราง สามารถทำได้โดยใส่เลขจำนวนฟิลด์ที่ต้องการเพิ่ม ในที่นี้คือ 1 ในช่อง Add แล้วเลือก “ที่จุดสุดท้ายของตาราง” แล้วกดปุ่ม “ลงมือ” ดังแสดงในรูปที่ ค15



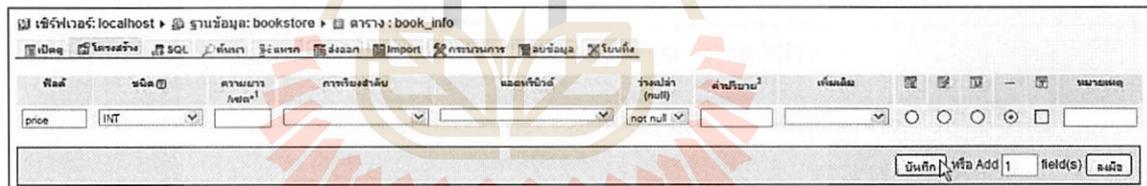
รูปที่ ค15: หน้าต่างการเพิ่มฟิลด์ในตาราง “book_info”

เมื่อกดปุ่ม “ลงมือ” แล้ว ก็จะปรากฏหน้าต่างฟิลด์ใหม่ขึ้นมา 1 ฟิลด์ ดังแสดงในรูปที่ ค16



รูปที่ ค16: หน้าต่างฟิลด์ใหม่ในตาราง “book_info”

ทดลองป้อนข้อมูลต่าง ๆ ลงในตาราง “book_info” แล้วกดปุ่ม “บันทึก” ดังแสดงในรูปที่ ค17



รูปที่ ค17: หน้าต่างตาราง “book_info” ที่มีฟิลด์ “price” เพิ่มขึ้น

เมื่อกดปุ่ม “บันทึก” แล้ว หน้าต่างตาราง “book_info” ก็จะปรากฏขึ้นดังแสดงในรูปที่ ค18

เบราว์เซอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู โครงสร้าง SQL ค้นหา แทรก ส่งออก Import กระบวนการ ลบข้อมูล โอนทั้ง

ตาราง book_info เปลี่ยนเสร็จแล้ว

คำสั่ง SQL: ALTER TABLE `book_info` ADD `price` INT NOT NULL ;

[แก้ไข] [สร้างโค้ด PHP]

ฟิลด์	ชนิด	การเรียงลำดับ	แอตทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย	เพิ่มเติม	การกระทำ
<input type="checkbox"/> no	tinyint(4)			ไม่		auto_increment	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> isbn	varchar(20)	utf8_general_ci		ไม่			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> name	varchar(50)	utf8_general_ci		ไม่			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> price	int(11)			ไม่			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

เลือกทั้งหมด / ไม่เลือกเลย *ทำกับที่เลือก:*

แสดง เสนอโครงสร้างตาราง

เพิ่ม Add field(s) ที่จุดสุดท้ายของตาราง ที่จุดเริ่มต้นของตาราง หลัง ลงมือ

ดัชนี: (7)				เนื้อที่ที่ใช้		สถิติของแถว	
ชื่อดัชนี	ชนิด	Cardinality	การกระทำ	ฟิลด์	ชนิด	ใช้งาน	ค่าเฉลี่ย
PRIMARY	PRIMARY	1	<input type="checkbox"/> <input type="checkbox"/>	no	ข้อมูล	44 ไบต์	รูปแบบ
isbn	INDEX	ไม่มี	<input type="checkbox"/> <input type="checkbox"/>	isbn	ดัชนี	3,072 ไบต์	การเรียงลำดับ
สร้างดัชนีโดยคอมป์ไรซ์	1	ลงมือ			รวม	3,116 ไบต์	utf8_general_ci
							แถว
							ความยาวแถว
							ขนาดแถว
							ต่อไป Autoindex
							สร้างเมื่อ
							ปรับปรุงครั้งสุดท้ายเมื่อ

รูปที่ ค18: หน้าต่างตาราง “book_info” เมื่อป้อนรายละเอียดของฟิลด์ “price” แล้วกดปุ่ม “บันทึก”

จากรูปที่ ค18 นั้น จะเห็นว่าฟิลด์ใหม่เพิ่มขึ้นมาอีก 1 ฟิลด์ชื่อว่า “price” ซึ่งในตอนนี ตาราง “book_info” มีฟิลด์ทั้งหมดรวม 4 ฟิลด์ด้วยกัน ถ้าต้องการเปลี่ยนแปลงหรือแก้ไขฟิลด์ต่าง ๆ ก็สามารถทำได้โดยกดปุ่มที่รูปดินสอในฟิลด์นั้น ๆ และถ้าต้องการลบฟิลด์ใดทิ้ง ก็กดปุ่มที่รูปกากบาทสีแดงในฟิลด์นั้น ๆ ได้เช่นกัน

การแทรกข้อมูลลงในตาราง (Insert)

เมื่อตาราง “book_info” ถูกสร้างขึ้นมาเรียบร้อยแล้ว ต่อไปให้ทดลองใส่ข้อมูลต่าง ๆ ลงในตารางดังกล่าว ซึ่งทำได้โดยกดปุ่ม “แทรก” ก็จะปรากฏหน้าต่างขึ้นดังแสดงในรูปที่ ค19

เบราว์เซอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู โครงสร้าง SQL ค้นหา **แทรก** ส่งออก Import กระบวนการ ลบข้อมูล โอนทั้ง

รูปที่ ค19: แทรกข้อมูลลงในตาราง “book_info”

เมื่อกดปุ่ม “แทรก” แล้ว จะปรากฏหน้าต่างขึ้นมาดังแสดงในรูปที่ ค20

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู
 โครงสร้าง
 SQL
 ค้นหา
 แทรก
 ส่งออก
 Import
 กระบวนการ
 ลบข้อมูล
 โหมดทั้ง

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
isbn	varchar(20)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
name	varchar(50)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
price	int(11)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>

ไม่สนใจ

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
isbn	varchar(20)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
name	varchar(50)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
price	int(11)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>

แทรกเป็นแถวใหม่ ▼ and then

รูปที่ ค20: หน้าต่างช่องข้อมูลต่าง ๆ ในตาราง “book_info”

ทดลองป้อนข้อมูลดังแสดงในรูปที่ ค21

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู
 โครงสร้าง
 SQL
 ค้นหา
 แทรก
 ส่งออก
 Import
 กระบวนการ
 ลบข้อมูล
 โหมดทั้ง

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
isbn	varchar(20)	<input type="text"/>	<input type="checkbox"/>	876-253-1234-09-1
name	varchar(50)	<input type="text"/>	<input type="checkbox"/>	Computer Programming
price	int(11)	<input type="text"/>	<input type="checkbox"/>	290

ไม่สนใจ

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
isbn	varchar(20)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
name	varchar(50)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
price	int(11)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>

แทรกเป็นแถวใหม่ ▼ and then

รูปที่ ค21: หน้าต่างแสดงการป้อนข้อมูลต่าง ๆ ในตาราง “book_info”

เมื่อกดปุ่ม “ลงมือ” แล้ว จะปรากฏหน้าต่างขึ้นมาดังแสดงในรูปที่ ค22

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิด ▶ โครงสร้าง ▶ SQL ▶ ค้นหา ▶ ฐานแทรก ▶ ส่งออก ▶ Import ▶ กระบวนการ ▶ ลบข้อมูล ▶ โยนทิ้ง

ไอ ไอ แลที่ถูกแทรก: 1

คำสั่ง SQL:

```
INSERT INTO `bookstore`.`book_info` (
  `no`,
  `isbn`,
  `name`,
  `price`
)
VALUES (
  '1', '876-253-1234-09-1', 'Computer Programming', '290'
);
```

[แก้ไข] [สร้างโค้ด PHP]

ฟิลด์	ชนิด	การเรียงลำดับ	แฉ ตหริ บิวร์ด	ว่าง เปล่า (null)	ค่า ปริยาย	เพิ่มเติม	กระทำการ
<input type="checkbox"/> no	tinyint(4)			ไม่		auto_increment	แก้ไข ✕ ลบข้อมูล โยนทิ้ง
<input type="checkbox"/> isbn	varchar(20)	utf8_general_ci		ไม่			แก้ไข ✕ ลบข้อมูล โยนทิ้ง
<input type="checkbox"/> name	varchar(50)	utf8_general_ci		ไม่			แก้ไข ✕ ลบข้อมูล โยนทิ้ง
<input type="checkbox"/> price	int(11)			ไม่			แก้ไข ✕ ลบข้อมูล โยนทิ้ง

เลือกทั้งหมด / ไม่เลือกเลย *เท่ากับที่เลือก:* แก้ไข ✕ ลบข้อมูล โยนทิ้ง

แสดง ไอ ไอ เสนอโครงสร้างตาราง ไอ

เพิ่ม Add 1 field(s) ที่จุดสุดท้ายของตาราง ที่จุดเริ่มต้นของตาราง ที่หลัง no ลงมือ

ดัชนี: ๓		เนื้อที่ที่ใช้		สถิติของแถว				
ชื่อคีย์	ชนิด	Cardinality	กระทำการ	ฟิลด์	ชนิด	ใช้งาน	ค่าตั้ง	ค่า
PRIMARY	PRIMARY	1	✕	no	ชนิด	ข้อมูล 64 ไบต์	รูปแบบ	ไม่คงที่
isbn	INDEX	ไม่มี	✕	isbn	ดัชนี	3,072 ไบต์	การเรียงลำดับ	utf8_general_ci
สร้างดัชนีโดยคอลัมน์ 1 ลงมือ				รวม	3,136 ไบต์	แถว	ความยาวแถว ๑	1
							ขนาดแถว ๑	64
							ต่อไป Autoindex	3,136 ไบต์
							สร้างเมื่อ	5
							ปรับปรุงครั้งสุดท้ายเมื่อ	09 ก.พ. 2009 น.
								09 ก.พ. 2009 น.

รูปที่ ค22: หน้าต่างแสดงการป้อนข้อมูลต่าง ๆ ในตาราง “book_info”

ตอนนี้ทดลองดูข้อมูลที่แทรกเข้าไป โดยกดปุ่ม “เปิดดู” ดังแสดงในรูปที่ ค23

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู | โครงสร้าง | SQL | ค้นหา | ฟิลด์แรก | ส่งออก | Import | กระบวนการ | ลบข้อมูล | โหมด

ฟิลด์	ชนิด	การเรียงลำดับ	แฉ คหรี บิวส์	ว่าง เปล่า (null)	ค่า ปริยาย	เพิ่มเติม	กระทำ
<input type="checkbox"/> no	tinyint(4)			ไม่		auto_increment	แก้ไข X ลบ เพิ่ม ลด
<input type="checkbox"/> isbn	varchar(20)	utf8_general_ci		ไม่			แก้ไข X ลบ เพิ่ม ลด
<input type="checkbox"/> name	varchar(50)	utf8_general_ci		ไม่			แก้ไข X ลบ เพิ่ม ลด
<input type="checkbox"/> price	int(11)			ไม่			แก้ไข X ลบ เพิ่ม ลด

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก: [แก้ไข] [ลบ] [เพิ่ม] [ลด]

รูปที่ ค23: หน้าต่างแสดงการกดปุ่ม “เปิดดู”

เมื่อกดปุ่ม “เปิดดู” แล้ว จะปรากฏหน้าต่างขึ้นมาดังแสดงในรูปที่ ค24

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู | โครงสร้าง | SQL | ค้นหา | ฟิลด์แรก | ส่งออก | Import | กระบวนการ | ลบข้อมูล | โหมด

แสดงระเบียบที่ 0 - 0 (1 ทั้งหมด, ค่าค้นใช้เวลา 0.0002 วินาที)

คำสั่ง SQL:

```
SELECT *
FROM 'book_info'
LIMIT 0 , 30
```

[แก้ไข] [อธิบาย SQL] [สร้างโค้ด PHP] [เรียกใหม่]

Query results operations

แสดง | Print view (with full texts) | ส่งออก

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน: แนวนอน และข้ามแถวแถวทุกๆ 100 เซลล์

	no	isbn	name	price
<input type="checkbox"/>	1	878-253-1234-09-1	Computer Programming	290

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก: [แก้ไข] [ลบ] [เพิ่ม] [ลด]

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน: แนวนอน และข้ามแถวแถวทุกๆ 100 เซลล์

รูปที่ ค24: หน้าต่างแสดงการเปิดดูข้อมูลทั้งหมดในตาราง “book_info”

สมมุติว่ามีการแทรกข้อมูลเพิ่มขึ้นอีก 2 ข้อมูล ดังแสดงในรูปที่ ค25

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู โครงสร้าง SQL ค้นหา แทรก ส่งออก Import การบวกรวม ลบข้อมูล โยนทิ้ง

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)			
isbn	varchar(20)			555-123-2200-21-1
name	varchar(50)			Web Application
price	int(11)			350

ไม่สนใจ

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)			
isbn	varchar(20)			345-222-1111-22-6
name	varchar(50)			Computer Security
price	int(11)			320

แทรกเป็นแถวใหม่ and then ส่งกลับ ลงมือ เริ่มใหม่

รูปที่ ค25: หน้าต่างแสดงการแทรกข้อมูลเข้าในตารางอีก 2 ข้อมูลในตาราง "book_info"



เมื่อกดปุ่ม “เปิดดู” แล้ว จะปรากฏหน้าต่างขึ้นมาดังแสดงในรูปที่ ค26

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู โครงสร้าง SQL ค้นหา แทรก ส่งออก Import กระบวนการ ลบข้อมูล โยนทิ้ง

แสดงระเบียบที่ 0 - 2 (3 ทั้งหมด, ค่าคืนใช้เวลา 0.0002 วินาที)

คำสั่ง SQL:

```
SELECT *
FROM `book_info`
ORDER BY `book_info`.`no` ASC
LIMIT 0 , 30
```

[แก้ไข] [อธิบาย SQL] [สร้างโค้ด PHP] [เรียกใหม่]

Query results operations

แสดง Print view (with full texts) ส่งออก

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน: แนวนอน และนำหัวแถวทุกๆ 100 เซลล์

เรียงโดยคีย์: ไม่มี ลงมือ

		no	isbn	name	price	
<input type="checkbox"/>			1	876-253-1234-09-1	Computer Programming	290
<input type="checkbox"/>			2	555-123-2200-21-1	Web Application	350
<input type="checkbox"/>			3	345-222-1111-22-8	Computer Security	320

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน: แนวนอน และนำหัวแถวทุกๆ 100 เซลล์

รูปที่ ค26: หน้าต่างแสดงการเปิดดูข้อมูลทั้งหมดในตาราง “book_info” หลังจากแทรกข้อมูลเข้าไปอีก 2 ข้อมูล



การแก้ไขข้อมูลในตาราง (Update)

ถ้าต้องการแก้ไขข้อมูลใด ๆ ในตาราง ให้เลือก checkbox หน้าข้อมูลนั้น ๆ แล้วกดปุ่มตรงเครื่องหมาย “ดินสอ” เช่น ต้องการแก้ไขราคาหนังสือ Computer Programming จาก 290 บาทเป็น 250 บาท สามารถทำได้ดังแสดงในรูปที่ ค27

	no	isbn	name	price
<input checked="" type="checkbox"/>	1	876-253-1234-09-1	Computer Programming	290
<input type="checkbox"/>	2	555-123-2200-21-1	Web Application	350
<input type="checkbox"/>	3	345-222-1111-22-8	Computer Security	320

เลือกทั้งหมด / ไม่เลือกเลย *เท่ากับที่เลือก:*

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน และเข้าหัวแถวทุกๆ เซลล์

รูปที่ ค27: หน้าต่างแสดงการแก้ไขข้อมูล

เมื่อกดปุ่ม “ดินสอ” จะปรากฏหน้าต่างดังแสดงในรูปที่ ค28

เซิร์ฟเวอร์: localhost > ฐานข้อมูล: bookstore > ตาราง: book_info

เปิดดู โครงสร้าง SQL ค้นหา ระบุแพ็คเกจ ส่งออก Import กระบวนการ ลบข้อมูล โยนทิ้ง

ฟิลด์	ชนิด	ฟังก์ชัน	วางเปล่า (null)	ค่า
no	tinyint(4)			1
isbn	varchar(20)			876-253-1234-09-1
name	varchar(50)			Computer Programming
price	int(11)			290

บันทึก and then ส่งกลับ

รูปที่ ค28: หน้าต่างแสดงข้อมูลที่เลือกแก้ไข

ทำการเปลี่ยนราคาจาก 290 บาทเป็น 250 บาท แล้วกดปุ่ม “ลงมือ” ดังแสดงในรูปที่ ค29

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bookstore ▶ ตาราง : book_info

เปิดดู โครงสร้าง SQL ค้นหา ระบุแถวๆ ส่งออก Import กระบวนการ ลบข้อมูล โบบัง

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
no	tinyint(4)		1	
isbn	varchar(20)		876-253-1234-09-1	
name	varchar(50)		Computer Programming	
price	int(11)		250	

บันทึก and then ส่งกลับ

ลงมือ เริ่มใหม่

รูปที่ ค29: หน้าต่างแสดงการเปลี่ยนแปลงข้อมูล

เมื่อกดปุ่ม “ลงมือ” จะปรากฏหน้าต่างดังแสดงในรูปที่ ค30

	no	isbn	name	price
<input type="checkbox"/> ✎ ✕	1	876-253-1234-09-1	Computer Programming	250
<input type="checkbox"/> ✎ ✕	2	555-123-2200-21-1	Web Application	350
<input type="checkbox"/> ✎ ✕	3	345-222-1111-22-8	Computer Security	320

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก: ✎ ✕

รูปที่ ค30: หน้าต่างแสดงข้อมูลทั้งหมดหลังจากการเปลี่ยนแปลงข้อมูล

การลบข้อมูลในตาราง (Delete)

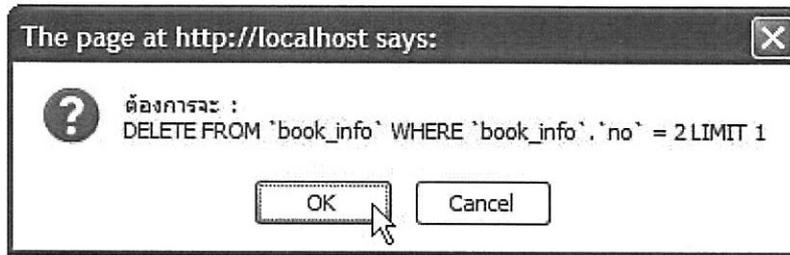
จากรูปที่ ค30 แสดงให้เห็นว่ามีข้อมูลอยู่ทั้งหมด 3 ข้อมูลด้วยกัน ถ้าต้องการลบข้อมูลที่ 2 ที่สามารถทำได้โดยเลือก checkbox หน้าข้อมูลที่ 2 แล้วกดปุ่มตรงเครื่องหมาย “กากบาท” ดังแสดงในรูปที่ ค31

	no	isbn	name	price
<input type="checkbox"/> ✎ ✕	1	876-253-1234-09-1	Computer Programming	250
<input checked="" type="checkbox"/> ✎ ✕	2	555-123-2200-21-1	Web Application	350
<input type="checkbox"/> ✎ ✕	3	345-222-1111-22-8	Computer Security	320

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก: ✎ ✕

รูปที่ ค31: หน้าต่างแสดงการเลือกข้อมูลที่ 2 ที่ต้องการลบ

เมื่อกดปุ่ม “กากบาท” แล้วจะปรากฏหน้าต่างถามย้ำอีกครั้งว่าต้องการลบข้อมูลดังกล่าวหรือไม่ ดังแสดงในรูปที่ ค32



รูปที่ ค32: หน้าต่างถามย้ำต้องการลบ

เมื่อกดปุ่ม “OK” จะปรากฏหน้าต่างดังแสดงในรูปที่ ค33

	no	isbn	name	price
<input type="checkbox"/>	1	876-253-1234-09-1	Computer Programming	250
<input type="checkbox"/>	3	345-222-1111-22-6	Computer Security	320

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก: [edit] [delete] [refresh]

รูปที่ ค33: หน้าต่างแสดงข้อมูลที่เหลือหลังจากการลบข้อมูลที่ 2

บรรณานุกรม <Bibliography>

กิตติ ภัคดีวัฒน์กุล และ อังศุมาลิน เวชนารายณ์. (2546) คัมภีร์ PHP, พิมพ์ครั้งที่ 3. กรุงเทพฯ : เคทีพี คอมพิวเตอร์คอนซัลท์.

สุพิน วรรณาน. (2543) PHP Web Application Development, กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

อดิศักดิ์ จันทร์มิน. (2549) สร้าง Web Application อย่างมืออาชีพด้วย PHP ฉบับ Workshop, กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

อดิศักดิ์ จันทร์มิน. (2549) สร้าง Web Application อย่างมืออาชีพด้วย PHP ฉบับ Workshop เล่ม 2, กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

อภิธานศัพท์ (ออนไลน์)

- <http://en.wikipedia.org/>
- <http://th.wikipedia.org/>

พร้อมเลิศ หล่อวิจิตร. (2550) คู่มือเรียน PHP และ MySQL สำหรับผู้เริ่มต้น, โปรวิชั่น, กรุงเทพฯ
เอแจ็ก (ออนไลน์)

- <http://www.adaptivepath.com/ideas/essays/archives/000385.php>
- http://www.dynamicajax.com/fr/XMLHttpRequest_Properties-.html
- <http://www.webopedia.com/TERM/X/XMLHttpRequest.html>
- https://developer.mozilla.org/En/AJAX:Getting_Started
- <http://www.w3.org/Protocols/HTTP/HTRESP.html>
- <http://www.ajaxf1.com/tutorial/ajax-php.html>
- <http://keelypavan.blogspot.com/2006/01/using-ajax.html>
- http://www.openjs.com/articles/ajax_xmlhttp_using_post.php
- http://www.w3schools.com/PHP/php_ajax_xml.asp

H. M. Deitel, P. J. Deitel and T. R. Neito, (2002). *Internet & World Wide Web: How to Program (2nd Edition)*, Prentice-Hall, Inc. New Jersey 07458

Ian Graham, (2000), *Introduction to HTML*, Available at <http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/>

Luke Welling and Laura Thompson, (2003). *PHP and MySQL Web Development (2nd Edition)*, Sams Publishing, Indianapolis, Indiana 46290

PHP Documentation Group, (2008). *PHP Manual*, Available at <http://www.php.net/download-docs.php>

Quentin Zervaas, (2007). *Practical Web 2.0 Application with PHP*, Apress, CA 94705

Robert W. Sebesta, (2003). *Programming the World Wide Web (2nd Edition)*, Addison Wesley

Sun Microsystems, (2008). *MySQL Reference Manual*, Available at <http://dev.mysql.com/doc/>

อภิธานศัพท์

<Glossary>

Administrator: ผู้ดูแลระบบ

AJAX (Asynchronous JavaScript And XML): เทคนิคในการพัฒนาโปรแกรมประยุกต์สำหรับเว็บเพื่อให้เกิดความสามารถในการโต้ตอบกับผู้ใช้ได้ดีขึ้น

Appserv: โปรแกรมจำลองการทำงานเป็นเครื่องแม่ข่ายที่ประกอบไปด้วย ระบบฐานข้อมูล MySQL, เว็บเซิร์ฟเวอร์ Apache, สคริปต์ PHP และ โปรแกรมช่วยจัดการฐานข้อมูล phpMyAdmin ทั้งหมดทำงานบนระบบปฏิบัติการวินโดวส์

ASP (Active Server Pages): ภาษาที่ทำงานบนฝั่งเซิร์ฟเวอร์ของบริษัทไมโครซอฟท์

Asynchronous: เกิดขึ้นในเวลาที่แตกต่างกัน

CERN (Centre Europeen de Recherche Nucleaire): ย่อมาจาก Centre Europeen de Recherche Nucleaire เป็นศูนย์วิจัยทางด้านอนุภาคทางด้านฟิสิกส์ในประเทศสวิตเซอร์แลนด์

CGI (Common Gateway Interface) Script: สคริปต์ทั่ว ๆ เช่น ASP, Perl, หรือ PHP เป็นต้น ที่ใช้ในการส่ง - รับข้อมูลระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย

Command Prompt: ตัวแปรคำสั่งในระบบปฏิบัติการวินโดวส์

CSS (Cascading Style Sheet): ภาษา ๆ หนึ่ง ที่ใช้ช่วยในการกำหนดรูปแบบของเอกสาร HTML ที่จะนำเสนอบนหน้าเว็บ ให้มีรูปแบบต่าง ๆ

DDL (Data Definition Language): ภาษาที่ใช้ในการกำหนดนิยามข้อมูล

DDM (Data Manipulation Language): ภาษาที่ใช้ในการปรับแต่งข้อมูล

DOM (Document Object Model): รูปแบบการจัดเรียงตัวของข้อมูลต่างๆ แบบเชิงวัตถุในเอกสาร HTML หรือ XML

Firefox: โปรแกรมท่องเว็บที่ใช้ได้ฟรีและเปิดเผยแพร่สโตร์โค้ดของมูลนิธิ Mozilla

HTML (HyperText Markup Language): ภาษาที่ใช้ในการแสดงข้อมูลต่าง ๆ บนโปรแกรมเว็บเบราว์เซอร์

HTTP (HyperText Transfer Protocol): เกณฑ์วิธีในการรับ - ส่งข้อมูล

Internet Explorer: โปรแกรมท่องเว็บของบริษัทไมโครซอฟท์

JavaScript: ภาษาสคริปต์ที่ใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน

Localhost: เครื่องคอมพิวเตอร์ที่ใช้อยู่ในขณะนี้ เป็นเครื่องแม่ข่าย มีที่อยู่ของไอพีคือ 127.0.0.1

MySQL: ระบบฐานข้อมูลโดยใช้ภาษา SQL ปัจจุบันบริษัทซันไมโครซิสเต็มส์

Perl (Practical Extraction and Report Language): ภาษาโปรแกรมแบบไดนามิก พัฒนาโดยนายแลร์รี่ วอลล์ (Larry Wall) ในปี ค.ศ. 1987

PHP (PHP Hypertext Preprocessor): ภาษาสคริปต์หนึ่งทำงานในฝั่งเซิร์ฟเวอร์ คิดค้นขึ้นโดย รัสมัส เลอร์ดอร์ฟ (Rasmus Lerdorf) ในปี ค.ศ. 1995

phpMyAdmin: โปรแกรมช่วยจัดการฐานข้อมูล

Search Engine: เครื่องมือช่วยในการค้นหาข้อมูลบนอินเทอร์เน็ต

SQL (Structured Query Language): ภาษาที่ใช้ในการสอบถามและจัดการข้อมูลของระบบฐานข้อมูล

SUTinsServer (SUTintServer): เป็นระบบแม่ข่ายที่ประกอบไปด้วย ระบบฐานข้อมูล MySQL, เว็บเซิร์ฟเวอร์ Apache, สคริปต์ PHP และ โปรแกรมช่วยจัดการฐานข้อมูล phpMyAdmin ที่ทำงานบนระบบปฏิบัติการลินุกซ์

Synchronous: เกิดขึ้นในเวลาเดียวกัน

Unicode: มาตรฐานอุตสาหกรรมที่ช่วยให้คอมพิวเตอร์แสดงผลและจัดการข้อความตัวอักษรที่ใช้ในระบบการเขียนของภาษาส่วนใหญ่ได้อย่างสอดคล้องกัน

URL (Uniform Resource Locator): ตำแหน่งระบุที่อยู่ของทรัพยากร

- W3C (World Wide Web Consortium): สมาคมนานาชาติที่ทำงานและประสานกับหน่วยงานต่าง ๆ เพื่อกำหนดรูปแบบมาตรฐานในการนำเสนอข้อมูลข่าวสารบนอินเทอร์เน็ต
- WWW (World Wide Web): ระบบที่มีจุดเชื่อมต่อของข้อมูลต่าง ๆ ผ่านทางอินเทอร์เน็ตโดยใช้โปรแกรมเว็บเบราว์เซอร์
- XHTML (Extensible HyperText Markup Language): ภาษาที่มีความคล้ายคลึงกับภาษา HTML แต่จะมีความเข้มงวดในเรื่องโครงสร้างภาษามากกว่า
- XML (Extensible Markup Language): ภาษาที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่แตกต่างกัน
- XSLT (Extensible Stylesheet Language Transformations): ภาษาที่มีรากฐานมาจาก XML ใช้แปลงข้อมูล XML ไปเป็นเอกสาร XML หรือ ภาษาอื่น ๆ ที่อ่านเข้าใจได้
- Zend Engine: เครื่องมือเสริมสำหรับ PHP คิดค้นขึ้นโดย Andi Gutmans และ Zeev Suraski
- ไคลเอนท์ (Client): เครื่องลูกข่าย
- ซอร์สโค้ดหรือรหัสต้นฉบับ (Source Code): ข้อความที่ถูกเขียนขึ้นตามรูปแบบของสคริปต์หรือภาษาในการเขียนโปรแกรม
- ซีฟ ซูร์สกี (Zeev Suraski): ผู้ร่วมพัฒนา PHP 3.0 และพัฒนาเครื่องมือเสริมใน PHP เรียกว่า Zend Engine
- เซิร์ฟเวอร์ (Server): เครื่องแม่ข่าย
- เบราว์เซอร์ (Browser): โปรแกรมท่องเว็บ เช่น Internet Explorer, Firefox เป็นต้น
- รัสมึส เลอร์ดอร์ฟ (Rasmus Lerdorf): ผู้คิดค้นภาษาสคริปต์ PHP
- ลินุกซ์ (Linux): ระบบปฏิบัติการชนิดหนึ่งที่สามารถใช้งานได้ฟรี
- วินโดวส์ (Windows): ระบบปฏิบัติการชนิดหนึ่งที่ต้องเสียค่าใช้จ่ายซื้อลิขสิทธิ์ของบริษัทไมโครซอฟท์
- เว็บเซิร์ฟเวอร์ (Web Server): เครื่องคอมพิวเตอร์ที่ให้บริการเว็บไซต์ ผู้ใช้เรียกชมหน้าเว็บไซต์ได้โดยใช้โพรโทคอล HTTP ผ่านทางโปรแกรมเว็บเบราว์เซอร์
- สคริปต์ (Script): ภาษาหรือชุดคำสั่งที่ใช้ในการประมวลผลหรือติดต่อกับฐานข้อมูล เช่น สคริปต์ PHP เป็นต้น
- แอนดี กัทแมนส์ (Andi Gutmans): ผู้ร่วมพัฒนา PHP 3.0 และพัฒนาเครื่องมือเสริมใน PHP เรียกว่า Zend Engine

ดัชนี

<Index>

-	68	<p>	4
!=	77	<pre>	6
!= or <>	202	<select>	11
!==	77	<small>	6
"	51		6
%	68	<strike>	5
&&, and	75	<sub>	5
*	68	<sup>	5
*=	68	<table>	8
/	68	<td>	8
//	59	<textarea>	11
, or	75	<th>	8
+	68, 77	<tr>	8
+=	68	<u>	5
<	202		5
<=	202	=	68, 202
<>	77	=	68
<a>	6	==	70, 77
	5	===	70, 77
<big>	6	>	202
 	4	>=	202
<caption>	8	ActiveXObject	267
<center>	6	ALL	188
<div>	6	ALTER	187, 195
<fieldset>	11	Apache	289
	6	Appserv	177, 180
<form>	11	AppServ	289, 307
<h1>	6	Asynchronous	263
<i>	5	Asynchronous	261
<iframe>	6	BETWEEN	202
	4	BIGINT	200
<input>	11	BLOB	201
<legend>	11	Boolean	49
	6	break	88
<marquee>	6	CHAR	201
	5	CHAR(n)	201

Command Prompt.....	307	HTTP (HyperText Transfer Protocol).....	264
continue.....	89	if	80
CREATE.....	182, 187	if...else	81
CSS.....	31, 261	if...elseif.....	82
DATE	200	imagecreate()	159
date().....	59	imagedestroy ().....	160
DATETIME	200	imagefill().....	159
DELETE	187, 193	imageline()	159
DESCRIBE.....	185	imagepng().....	159
do...while.....	86	imagestring()	159
DOM (Document Object Model).....	261	IN	202
DOUBLE.....	200	include().....	153
DROP	187, 198	include_once().....	155
echo	58	INDEX.....	187, 194
ENUM	201	INSERT	187, 189
fclose().....	128	instance.....	267, 269, 271
feof().....	133	INT.....	200
fgetc().....	142	Integer	49
fgetcsv()	135	Internet Explorer.....	267
fgetss().....	134	Internet Explorer	267
FILE.....	188, 199	IS NOT NULL.....	202
file().....	138	IS NULL.....	202
file_exists().....	143	JavaScript.....	261
file_get_contents().....	139	Jesse James Garrett	261
filesize().....	143	LIKE	202
Firefox	267, 269	LOB.....	201
Float.....	49	LONGTEXT.....	201
FLOAT	200	MEDIUMBLOB	201
flock().....	129	MEDIUMINT.....	200
fopen().....	126	MEDIUMTEXT	201
for	84	MySQL.....	177, 289, 307
foreach.....	87	mysql_close.....	208
fpass thru()	137	mysql_connect.....	208
fread()	140	mysql_create_db.....	208
fseek().....	143	mysql_drop_db	208
ftell().....	143	mysql_error.....	208
fwrite().....	130	mysql_fetch_array.....	208
GET	276	mysql_fetch_row	208
GRANT.....	188	mysql_info	208
header().....	159	mysql_list_dbs.....	208
HTML.....	1	mysql_list_fields.....	208

mysql_list_tables	208	TIMESTAMP	200
mysql_pconnect	209	TINYBLOB	201
mysql_query	209	TINYINT	200
mysql> help	202	TINYTEXT	201
NOT IN	202	TRUNCATE	187, 198
NOT LIKE	202	unlink()	143
ob_end_flush()	230	UPDATE	187, 192
ob_get_contents()	230	USAGE	188
ob_start()	230	VARCHAR	201
onreadystatechange	264	while	85
PHP	47, 289	XHTML	261
phpMyAdmin	177, 307	XML	261, 264, 277
POST	276	XMLHttpRequest	261, 264, 267
printf()	59	XMLHttpRequest กับ XML	277
PROCESS	188, 199	XMLHttpRequest กับฐานข้อมูล	283
readfile()	137	XSLT	261
readyState	265	YEAR	200
REGEXP	202	การแก้ไขข้อมูลในตาราง (Update)	322
RELOAD	188, 199	การแทรกข้อมูลลงในตาราง (Insert)	316
require()	153	การลบข้อมูลในตาราง (Delete)	323
require_once()	155	การส่งแบบการกำหนดค่าปรีายาย	119
responseText	265	การส่งแบบการผานคา	117
responseXML	265	การส่งแบบการผานตัวอ้างอิง	118
REVOKE	189	ขอบเขตของตัวแปร (Variable Scope)	57
rewind()	143	แบบ global	57
Safari	267	แบบ local	57
SELECT	187, 190	ความสัมพันธ์แบบ Many-to-Many	180
SET	201	ความสัมพันธ์แบบ One-to-Many	180
setcookie()	230	ความสัมพันธ์แบบ One-to-One	179
SHOW	184	คอลัมน์ (Column)	178
SHUTDOWN	188, 199	ค่า (Value)	178
SMALLINT	200	ค่าคงที่ (Constant)	58
SQL	177, 182	คีย์ (Keys)	178
status	265	คูกี้	229
statusText	266	คุณสมบัติต่าง ๆ ใน CSS	42
String	49, 68, 105, 201	คุณสมบัติของกล่องข้อความ (Box Properties)	43
SUTinsServer	297	ชนิดข้อมูล (Data Types)	49
switch	83	เซสชัน	229, 238
Synchronous	263	ตัวแปร (Variable)	53
TEXT	201	ตัวแปรคูกี้ (HTTP Cookies)	56
TIME	200	ตัวแปรแบบ GET (HTTP GET Variables)	56

ตัวแปรแบบ Global (Global Variables).....	57	ฟังก์ชันเกี่ยวกับฐานข้อมูล MySQL.....	207
ตัวแปรแบบ HTTP File Upload Variables	56	ฟังก์ชันเกี่ยวกับไดเรคทอรี	104
ตัวแปรแบบ POST (HTTP POST Variables).....	56	ฟังก์ชันเกี่ยวกับเมล	111
ตัวแปรแบบ Request (Request Variables).....	56	ฟังก์ชันเกี่ยวกับระบบไฟล์ข้อมูล	113
ตัวแปรแบบ Session (Session Variables).....	56	ฟังก์ชันเกี่ยวกับสตริง	105
ตัวแปรแบบ The previous error message	57	ฟังก์ชันที่เกี่ยวกับชนิดตัวอักษร	103
ตัวแปรระบบ (Server Variables).....	55	ฟังก์ชันที่เกี่ยวกับวันและเวลา.....	96
ตัวแปรแวดล้อม (Environment Variables).....	56	ฟังก์ชันที่เกี่ยวกับอะเรย์.....	97
ตัวเลือกแบบคลาส (Class Selector).....	35	รหัสเทียมแบบคลาส (Pseudo-classes)	39
ตัวเลือกแบบไอดี (ID Selector).....	38	รหัสเทียมแบบอีลีเมนต์ (Pseudo-elements).....	39
แถว (Row).....	178	รูปแบบของ CSS.....	33
ทิม เบอร์เนอรส์-ลี	1	แบบในบรรทัด (Inline Styles).....	33
แท็กตาราง	8	แบบภายนอก (External Style Sheet).....	34
แท็กแบบฟอร์ม.....	11	แบบภายใน (Internal Style Sheet).....	34
แท็กเฟรม	16	ลบฐานข้อมูล (Drop Database).....	312
แท็กไอเฟรม	19	สร้างฐานข้อมูลใหม่ (Create Database).....	310
ฟังก์ชันเกี่ยวกับการวาดรูป	155	สร้างตารางใหม่ (Create Table).....	313
ฟังก์ชันเกี่ยวกับคณิตศาสตร์.....	109	เอแจ็ก.....	261

ประวัติผู้เขียน

<Curriculum Vitae>

ชื่อ ผู้ช่วยศาสตราจารย์ ดร. คชะ ชาญศิลป์ (Asst. Prof. Kacha Chansilp, PhD.)

วันเดือนปีเกิด 5 พฤษภาคม 2506

ตำแหน่งปัจจุบัน ผู้ช่วยศาสตราจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์

สถานที่ทำงาน สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
111 ถนนมหาวิทยาลัย ตำบลสุรนารี อำเภอเมืองนครราชสีมา
จังหวัดนครราชสีมา 30000
โทรศัพท์ 0-4422-4237, 0-4422-4422 โทรสาร 0-4422-4602
อีเมล kacha@sut.ac.th

ประวัติการศึกษา

2542 - 2546 PhD (Interactive Multimedia Technologies),
Edith Cowan University, Perth, Western Australia

2536 - 2538 MA (Graphics Design),
New York Institute of Technology, New York, USA

2532 - 2537 BA (Computer Science),
Queen College, New York, USA

ผลงานทางวิชาการ

1. Chansilp, K. & Oliver, R. (2002). *"Using Multimedia to Develop Students' Programming Concepts"*, Paper presented at the Proceeding of the EDU – 2002 (Higher Education without Borders Sustainable Development in Higher Education), Nov. 25-27, Khon Kaen, THAILAND.
2. Chansilp, K (2003). *"Development, Implementation and Evaluation of an Interactive Multimedia Instructional Model: A Teaching and Learning Programming Approach"*, Unpublished doctoral dissertation, Edith Cowan University, Perth, Western Australia.
3. Chansilp, K. & Oliver, R. (2004). *"Students' responses to the use of a multimedia tool for learning computer programming"*, In L. Cantoni & C. McLoughlin (Eds). Proceedings of Ed-Media 2004, World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, Lugano, Switzerland. (pp. 1739-1746). Norfolk, USA: Association for the Advancement of Computing in Education.

4. Chansilp, K. & Mukviboonchai, S. (2004). *"The Conceptual Framework of Dynamic Interactive Visualization Tool in Teaching Data Structure"*, Proceeding of the EDU-COM 2004 (New Challenges for Sustainability and Growth in Higher Education), Nov. 24-26, Khon Kaen, THAILAND.

5. Chansilp, K. & Mukviboonchai, S. (2005). *"The Design and Development of Dynamic Interactive Visualization Tool in Teaching Data Structure"*, Paper presented at the Seventh International Conference on Information Integration and Web Based Applications & Services (iiWAS2005), Sept. 19-21, Kuala Lumpur, Malaysia. (Volume 1: pp. 277-286) Austrain Computer Society.

6. Chansilp, K. & Oliver, R. (2006). *"Reusable and Sharable Learning Objects Supporting Students' Learning of Data Structures in University Courses"*, Proceeding of the EDU-COM 2006 Engagement and Empowerment: New Opportunities for Growth in Higher Education, Nov. 22-24, Nong Khai Campus of Khon Kaen University, Nong Khai, THAILAND (pp. 105-113) (in CD-Rom format).

7. สำราญ ขอบใจและคะชา ชาญศิลป์. (2549). *การพัฒนาบบบริหารการจัดการเว็บไซต์กองบินของกองทัพอากาศไทย* การประชุมวิชาการระดับชาติ ด้านวิทยาศาสตร์และเทคโนโลยี 2549 (National Conference Science & Technology 2006) มหาวิทยาลัยราชภัฏสุรินทร์, สุรินทร์, วันที่ 14-15 ธันวาคม 2549.

8. ธนินทร์ ระเบียบโพธิ์, คะชา ชาญศิลป์ และ สมพันธ์ ชาญศิลป์. (2550). *วิธีการค้นคืนข้อมูลอย่างมีประสิทธิภาพสำหรับระบบจัดการการเรียนการสอน* การประชุมวิชาการทางคอมพิวเตอร์และเทคโนโลยีสารสนเทศ (CIT2007: Conference on Computer Information Technologies 2007) มหาวิทยาลัยเทคโนโลยีราชมงคลวิทยาเขตศรีวิชัย, จังหวัดตรัง วันที่ 31 ม.ค. – 3 ก.พ. 2550.

9. ชลดา พรหมสุข, คะชา ชาญศิลป์ และ สมพันธ์ ชาญศิลป์. (2550). *เครื่องมือสำหรับสร้างระบบจัดการการประชุม* การประชุมทางวิชาการระดับชาติ ครั้งที่ 3 ด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ (NCCIT' 07 : The 3rd National Conference on Computing and Information Technology) สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, กรุงเทพฯ วันที่ 25-26 พฤษภาคม 2550.

10. คะชา ชาญศิลป์. (2551). *ซอฟต์แวร์ภาพเคลื่อนไหวสื่อหลายมิติในการเรียนรู้โดยใช้ปัญหาเป็นหลักสำหรับรายวิชาการโปรแกรมคอมพิวเตอร์ภาษาซี* การประชุมทางวิชาการ คุรุศาสตร์อุตสาหกรรมแห่งชาติ ครั้งที่ 2 เรื่อง "การสอนวิทยาศาสตร์และเทคโนโลยีในอาชีวศึกษาตามแนวทางเศรษฐกิจพอเพียง" (S&T Teaching in Vocational Education based on Sufficient Economy) มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ, กรุงเทพฯ, วันที่ 24-25 มกราคม 2551.

11. Tanatorn Tanantong and Kacha Chansilp. (2008). *"Development of Learning Management System with Weblog Technology"*, The 2nd National Conference on Information Technology 2008, Grand Mercure Fortune Bangkok Hotel, THAILAND. Nov. 6-7, (pp. 334-337).

แหล่งทุนและรางวัลที่เคยได้รับ

1. ทุนพัฒนาอาจารย์ มหาวิทยาลัยเทคโนโลยีสุรนารี และทุนมหาวิทยาลัยอิตีทีโคแวน เพื่อศึกษาต่อระดับปริญญาเอก ที่มหาวิทยาลัยอิตีทีโคแวน เมืองเพิร์ท ประเทศออสเตรเลีย ในปี 1999

2. ทุน 2006 Endeavour Australia Cheung Kong Awards for Asian Scholars จากประเทศออสเตรเลีย เพื่อทำวิจัยระยะสั้น (4-6 เดือน) ที่ประเทศออสเตรเลีย

3. รางวัลชมเชย การประกวดผลงานสิ่งประดิษฐ์ ประจำปี 2551 ในงานวันนักประดิษฐ์และวันนักประดิษฐ์นานาชาติ ครั้งที่ 1 ประจำปี 2551 สำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) เรื่อง "ซอฟต์แวร์ภาพเคลื่อนไหวสื่อหลาย

มิติในการเรียนรู้โดยใช้ปัญหาเป็นหลักสำหรับรายวิชาการโปรแกรมคอมพิวเตอร์ภาษาซี (Hypermedia Animation Software in Problem-Based Learning for C Programming)” สาขาการศึกษา ด้านพัฒนาลังคมและวัฒนธรรม

4. รางวัลระดับดี ในการประกวดเพื่อรับรางวัล Inventor Award ในงานวันนักประดิษฐ์และวันนักประดิษฐ์นานาชาติ ครั้งที่ 2 ประจำปี 2552 สำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) เรื่อง “ระบบสอบออนไลน์เคลื่อนที่จาก มทส. เวอร์ชัน 2.0 (SUT-MOTS: SUT Mobile Online Test System Version 2.0)” สาขาเทคโนโลยีสารสนเทศและนิเทศศาสตร์ ด้านวิทยาศาสตร์และเทคโนโลยี

5. พนักงานดีเด่น ด้านผลงานสิ่งประดิษฐ์ ประจำปี 2552 มหาวิทยาลัยเทคโนโลยีสุรนารี

6. รางวัลที่ 1 ประเภทซอฟต์แวร์ รุ่นทั่วไป เรื่อง “วัตถุการเรียนรู้เรื่องโครงสร้างข้อมูลและขั้นตอนวิธี (LODSA: Learning Object in Data Structures and Algorithms)” ในการประกวดสิ่งประดิษฐ์ ครั้งที่ 4 ประจำปี 2552 สถาบันวิจัยและพัฒนา มหาวิทยาลัยเทคโนโลยีสุรนารี ในงานสัปดาห์วิทยาศาสตร์และเทคโนโลยี ณ อาคารสุรพัฒน์ 2 มหาวิทยาลัยเทคโนโลยีสุรนารี วันที่ 18-19 สิงหาคม 2522

