



การเขียนโปรแกรมบนเว็บด้วย PHP ร่วมกับฐานข้อมูล MySQL

ปริญญา น้อยdon尉

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

2556



การเขียนโปรแกรมบนเว็บด้วย PHP ร่วมกับฐานข้อมูล MySQL

ปริญญา น้อยดอนไพร

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

ISBN::978-974-306-469-2

2556

การเขียนโปรแกรมบนเว็บด้วย PHP ร่วมกับฐานข้อมูล MySQL

พิมพ์ครั้งที่ 1 (ฉบับแก้ไขเพิ่มเติม พฤศจิกายน 2558)

พฤษภาคม 2556

จำนวน 100 เล่ม

ส่วนลด 10%

ข้อมูลทางบรรณานุกรม

ปริญญา น้อยดอนไพร. (2556). การเขียนโปรแกรมบนเว็บด้วย PHP ร่วมกับฐานข้อมูล MySQL. สุราษฎร์ธานี : มหาวิทยาลัยราชภัฏสุราษฎร์ธานี, 322 หน้า

ISBN 978-974-306-469-2

จัดพิมพ์โดย

มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

272 หมู่ที่ 9 ตำบลชุมแพ อำเภอเมือง

จังหวัดสุราษฎร์ธานี 84100

โทรศัพท์ 077 913333 โทรสาร 077 355468

www.sru.ac.th

คำนำ

ในปัจจุบันเทคโนโลยีสารสนเทศมีบทบาทในชีวิตประจำวันมากขึ้น การที่จะสร้างหรือพัฒนาระบบงาน ที่ใช้ในองค์กรในลักษณะของเว็บแอ��พลิเคชันหรือการเขียนโปรแกรมบนเว็บเพื่อจัดเก็บข้อมูล และสืบค้น ข้อมูลในระบบฐานข้อมูล การที่จะพัฒนาได้นั้น จะเป็นต้องมีผู้ชำนาญในการพัฒนาระบบจริงจะทำให้ระบบนั้น มีประสิทธิภาพในการใช้งาน ดังนั้น เป้าหมายสำคัญของหนังสือวิชาการเล่มนี้ ต้องการเสริมสร้างทักษะความรู้ ให้กับนักศึกษาหรือผู้ที่สนใจ เริ่มจากการเขียนเว็บแอฟพลิเคชันตั้งแต่เริ่มต้น จนถึงประยุกต์ใช้ในระดับมืออาชีพต่อไป

การเขียนเว็บแอฟพลิเคชันนั้น มีภาษาที่ใช้สำหรับการพัฒนามีอยู่ด้วยกันหลายภาษา เช่น ASP, Perl, JSP และ PHP เป็นต้น แต่ภาษาที่ได้รับความนิยมอย่างแพร่หลาย โดยสังเกตได้จากเว็บไซต์ส่วนใหญ่ เช่น facebook.com และเว็บไซต์หน่วยงานภาครัฐและเอกชน โดยภาษา PHP นั้น เป็นภาษาสคริปต์ที่ทำงานใน ฝั่งของเครื่องแม่ข่าย เพื่อประมวลผลแล้วส่งกลับมายังเครื่องที่ร้องขอในลักษณะของภาษา HTML ลักษณะ สำคัญของภาษา PHP จะมีโครงสร้างที่ง่ายแก่การทำความเข้าใจ คล้ายกับภาษาซี หมายเหตุเริ่มต้นพัฒนา และยังสามารถเขียนโปรแกรมในลักษณะของออบเจ็คได้อีกด้วย หมายเหตุสำหรับการย่อสคริปต์คำสั่งให้สั้นลง ฐานข้อมูลก็เป็นส่วนหนึ่งที่มีความสำคัญ สำหรับการเขียนเว็บแอฟพลิเคชันด้วยภาษา PHP โดยภาษา PHP สามารถติดต่อกับฐานข้อมูลหลายชนิด เช่น Oracle, Progress, MS-Access และ MySQL เป็นต้น ในหนังสือ เล่มนี้จะใช้ฐานข้อมูล MySQL สำหรับเรียนรู้และฝึกปฏิบัติ

หนังสือเล่มนี้เขียนขึ้นเพื่อถ่ายทอดความรู้และประสบการณ์ของผู้เขียน (จากประสบการณ์ สู่ภาคปฏิบัติ) ประสบการณ์ของผู้เขียนนั้นได้มาจากงานสอน และพัฒนาเว็บแอฟพลิเคชันด้วยภาษา PHP และ MySQL มาโดยตลอด เป็นเวลานานไม่น้อยกว่า 10 ปี และใช้ประกอบการเรียนการสอนรายวิชาการเขียน โปรแกรมบนเว็บ หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสุราษฎร์ ธานี หนังสือเล่มนี้ยังหมายเหตุกับผู้ที่สนใจในด้านการเขียนโปรแกรมหรือพัฒนาเว็บแอฟพลิเคชัน นำไปศึกษา เรียนรู้ และประยุกต์ใช้งานที่หลากหลายตามลักษณะของงานที่แตกต่างกัน ดังนั้นผู้เขียนจึงหวังว่า หลังจากที่ ได้ศึกษาเนื้หาจากหนังสือเล่มนี้ทั้งหมดแล้ว ผู้อ่านจะสามารถนำความรู้ที่ได้ไปประยุกต์ใช้ในการเขียน โปรแกรมบนเว็บหรือพัฒนาเว็บแอฟพลิเคชันต่อไป

สารบัญ

	หน้าที่
คำนำ	(1)
สารบัญ	(3)
สารบัญภาพ	(9)
สารบัญตาราง	(13)
บทที่ 1 บทนำ	1
1.1 ประเภทของภาษาสคริปต์	1
1.2 ความสามารถของภาษา PHP.....	1
1.3 ประวัติความเป็นมาของภาษา PHP	2
1.4 หลักการทำงานของภาษา PHP	3
1.5 MySQL	5
1.6 ชนิดของข้อมูลที่ MySQL สนับสนุน	6
สรุป	8
คำถาມท้ายบท	8
บทที่ 2 PHP เป็งตัน	9
2.1 การแทรกสคริปต์ PHP ในเอกสาร HTML	9
2.2 องค์ประกอบพื้นฐานของการเขียนโปรแกรมด้วยภาษา PHP	12
2.3 การแสดงผลข้อมูลบนเว็บбраузอร์	13
2.4 ชนิดข้อมูลที่รองรับในภาษา PHP	18
2.5 การแปลงชนิดข้อมูล	22
2.6 การปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ	25
2.7 พังก์ชันที่เกี่ยวข้องกับชนิดข้อมูล	26
2.8 คำส่วนของภาษา PHP	31
สรุป	32
คำถาມท้ายบท	32
บทที่ 3 ตัวแปร ตัวดำเนินการและนิพจน์.....	33
3.1 ตัวแปร	33
3.2 การกำหนดค่าตัวแปร	33
3.3 ขอบเขตของตัวแปร	39

สารบัญ (ต่อ)

	หน้าที่
3.4 การตรวจสอบและยกเลิกตัวแปร.....	46
3.5 ตัวดำเนินการ และนิพจน์	49
สรุป	59
คำถามท้ายบท	59
บทที่ 4 โครงสร้างควบคุม	61
4.1 โครงสร้างเงื่อนไข	61
4.2 โครงสร้างเงื่อนไขทำซ้ำ	68
4.3 คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ	77
สรุป	81
คำถามท้ายบท	81
บทที่ 5 พังก์ชัน	83
5.1 พังก์ชันมาตรฐาน	83
5.2 พังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง	84
5.3 พังก์ชันพารามิเตอร์	87
5.4 การส่งค่ากลับจากพังก์ชันด้วยคำสั่ง return	90
5.5 พังก์ชันแบบเรียกตัวเอง	92
5.6 พังก์ชันไลบรารี	94
สรุป	96
คำถามท้ายบท	96
บทที่ 6 อาร์เรย์	99
6.1 การสร้างอาร์เรย์	100
6.2 การเข้าถึงข้อมูลภายในอาร์เรย์	105
6.3 พังก์ชันอื่นๆ ที่เกี่ยวกับอาร์เรย์	107
สรุป	129
คำถามท้ายบท	129
บทที่ 7 ข้อความ ตัวเลข วันและเวลา	131
7.1 พังก์ชันที่ใช้จัดการข้อความ	131
7.2 พังก์ชันที่ใช้จัดการจำนวนและตัวเลข	145

สารบัญ (ต่อ)

	หน้าที่
7.3 พังก์ชันที่ใช้จัดการวันและเวลา	147
สรุป	161
คำถมท้ายบท	161
บทที่ 8 การจัดการไฟล์และไดเรกทอรี	163
8.1 การอ้างอิงตำแหน่งไฟล์และไดเรกทอรี	163
8.2 การจัดการกับไฟล์	164
8.3 การจัดการกับไดเรกทอรี	178
8.4 การอัพโหลดไฟล์	184
สรุป	187
คำถมท้ายบท	187
บทที่ 9 การจัดการข้อมูลจากฟอร์ม	189
9.1 กำหนดขอบเขตของฟอร์ม	189
9.2 การใช้งานส่วนนำเข้าข้อมูล	190
9.3 การทดสอบการแสดงผลฟอร์ม	197
9.4 การจัดเก็บข้อมูลลงไฟล์	199
สรุป	201
คำถมท้ายบท	201
บทที่ 10 การเชื่อมโยงระหว่างเพจ คุกกี้ และเซสชัน	203
10.1 เยดเดอร์	203
10.2 การส่งข้อมูลระหว่างเว็บเพจแบบ Query String	205
10.3 การจัดเก็บข้อมูลแบบคุกกี้	207
10.4 การจัดเก็บข้อมูลแบบเซสชัน	207
10.5 ขั้นตอนการประยุกต์ใช้เซสชัน เพื่อเป็นแนวทางในการพัฒนาระบบ Login	216
สรุป	219
คำถมท้ายบท	220
บทที่ 11 การเขียนโปรแกรมเชิงวัตถุ	221
11.1 คลาส	221
11.2 การนำคลาสมาใช้ในสคริปต์ PHP	222
11.3 ออบเจกต์ และอินสแตนซ์	222

สารบัญ (ต่อ)

	หน้าที่
11.4 เมรอด	223
11.5 โมดิไฟล์เอกสาร แบบ public และ private	226
11.6 พร้อมเพอร์ตี้ และฟิลด์	227
11.7 ค่าคงที่	229
11.8 คอนสตรัคเตอร์	230
11.9 โมดิไฟล์เอกสาร Static	231
11.10 การสืบทอด	233
สรุป	235
คำถาวรท้ายบท	235
บทที่ 12 การใช้เทคนิค AJAX ร่วมกับ PHP	237
12.1 AJAX	237
12.2 ประวัติความเป็นมา	238
12.3 ที่มาของปัญหา	238
12.4 โครงสร้างของ AJAX	239
12.5 การทำงานของ AJAX	240
12.6 การใช้ AJAX Framework	242
12.7 ฟังก์ชันที่ใช้จัดการข้อมูล	245
12.8 แนวทางการพัฒนา AJAX Application	245
12.9 ปัญหาภาษาไทยใน AJAX	247
12.10 การส่งผลลัพธ์กลับมาเป็น JavaScript	247
12.11 การอัปเดโทินพุต Select ด้วยเทคนิค AJAX	251
12.12 อีเวนต์	255
12.13 การกำหนดอีเวนต์	256
12.14 ออกแบบเจ็กต์อีเวนต์	258
12.15 อีเวนต์เกี่ยวกับมาส์	259
12.16 อีเวนต์เกี่ยวกับคีย์บอร์ด	260
สรุป	267
คำถาวรท้ายบท	267

สารบัญ (ต่อ)

	หน้าที่
บทที่ 13 ฐานข้อมูล MySQL และการใช้งาน phpMyAdmin	269
13.1 คำสอ柏ถามหรือคำสั่งเพื่อนฐานของ MySQL	269
13.2 องค์ประกอบของฐานข้อมูล	271
13.3 ชนิดข้อมูลสำหรับกำหนดให้กับขอบเขตข้อมูล	272
13.4 แอพทริบิวต์สำหรับกำหนดให้กับขอบเขตข้อมูล	274
13.5 การสร้างฐานข้อมูล	275
13.6 การสร้างตาราง.....	276
13.7 การแทรก ปรับปรุง ลบ และเรียกดูข้อมูลในตาราง	281
13.8 จัดการฐานข้อมูลด้วย phpMyAdmin	291
13.9 การลบฐานข้อมูล และตาราง	298
13.10 ส่วนของการใช้คำสอ柏ถามแบบกำหนดเอง	298
13.11 ส่วนของการค้นหา	299
13.12 ส่วนของการกำหนดสีที่	299
13.13 เทคนิคการนำเข้าข้อมูลจากไฟล์ Excel	300
สรุป	304
คำถາມท้ายบท	304
บทที่ 14 การใช้ PHP ร่วมกับ MySQL	305
14.1 พังก์ชันเชื่อมต่อ/ยกเลิกการเชื่อมต่อระหว่างภาษา PHP และระบบฐานข้อมูล MySQL 305	305
14.2 พังก์ชันสำหรับการเลือกฐานข้อมูล	308
14.3 การส่งคำสอ柏ถามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL	309
14.4 การอ่านข้อมูลผลลัพธ์	312
14.5 การเปลี่ยนแปลงข้อมูล	317
14.6 การแก้ไขข้อมูลภาษาไทยกลายเป็น "?????"	326
14.7 การตรวจสอบการเปลี่ยนแปลงข้อมูล	326
14.8 การตรวจสอบข้อผิดพลาดระหว่างการทำงานร่วมกับ MySQL	329
14.9 การตรวจสอบชื่อฐานข้อมูลและตาราง	329
14.10 การตรวจสอบโครงสร้างของขอบเขตข้อมูล	331
สรุป	335
คำถາມท้ายบท	335

สารบัญ (ต่อ)

	หน้าที่
บรรณานุกรม	337
ภาคผนวก	343
ภาคผนวก ก เครื่องมือที่ใช้สำหรับเขียนโปรแกรมบนเว็บ	345
ภาคผนวก ข กรณีศึกษา : ระบบบันทึกการทำกิจกรรมของนักศึกษา	363
ดัชนี	417

สารบัญภาพ

ภาพที่	หน้าที่
1.1 ผู้ร่วมพัฒนา PHP	2
1.2 รูปแบบการติดต่อสื่อสารระหว่างเว็บไซต์และเว็บเซิร์ฟเวอร์ผ่านโปรโตคอล HTTP	4
1.3 รูปแบบการติดต่อสื่อสารระหว่างเว็บไซต์และเว็บเซิร์ฟเวอร์เพื่อเรียกเว็บเพจรวมด้วย 1.4 การร้องขอการใช้งานจากเว็บไซต์ไปยังเว็บเซิร์ฟเวอร์ที่ได้ติดตั้ง Engine ของภาษา PHP ..5	4
1.5 ทีมพัฒนาฐานข้อมูล MySQL	6
3.1 แสดงตัวอย่างการส่งค่าแบบ Query String	45
3.2 แสดงองค์ประกอบของการคำนวณทางคณิตศาสตร์	50
4.1 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if	62
4.2 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if ... else	63
4.3 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if ... elseif	64
4.4 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ while	69
4.5 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ do ... while	70
4.6 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ for	72
4.7 แสดงลักษณะการวนรอบของโครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ	73
4.8 แสดงลักษณะการวนรอบของโครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข	74
5.1 แสดงการส่งผ่านค่าพารามิเตอร์เข้าไปประมวลผลภายในฟังก์ชัน	86
5.2 แสดงการเรียกใช้ฟังก์ชันภายนอกฟังก์ชันที่กำลังทำงานในปัจจุบัน	87
6.1 ตัวอย่างการจัดเก็บตัวแปรอาร์เรย์ \$Province ในหน่วยความจำ	99
6.2 แสดงหน้าเว็บเพจสำหรับเริ่มต้นอัปโหลดไฟล์	128
7.1 แสดงตัวอย่างผลลัพธ์การสร้างปฏิทินออนไลน์	160
8.1 แสดงการเปิดไฟล์รูปภาพ	175
9.1 แสดงตัวอย่างการสร้างฟอร์มเพื่อนำเข้าข้อมูล	191
9.2 แสดงตัวอย่างการใช้งานปุ่มตัวเลือกแบบปุ่มกลม	192
9.3 แสดงตัวอย่างการใช้งานรายการตัวเลือกแบบ Menu	193
9.4 แสดงตัวอย่างการใช้งานตัวเลือกแบบปุ่มเปลี่ยน	194
9.5 แสดงตัวอย่างการใช้งานปุ่มส่งค่าข้อมูล และล้างข้อมูล	195
9.6 แสดงตัวอย่างการใช้งานปุ่มส่งค่าข้อมูลแบบรูปภาพ	195
9.7 แสดงตัวอย่างการใช้งานส่วนนำเข้าข้อมูลจากไฟล์	197
9.8 แสดงตัวอย่างหน้าเว็บเพจเมื่อเรียกดูไฟล์ regist.php บนเว็บบราวเซอร์	198

สารบัญภาพ (ต่อ)

ภาคที่	หน้าที่
9.9 แสดงตัวอย่างหน้าเว็บเพจเมื่อคลิก Send เพื่อส่งข้อมูลไปประมวลผลที่ member.php	199
9.10 แสดงหน้าเพิ่มข้อมูลสมาชิก และหน้ารายงานผลข้อมูลสมาชิก	201
10.1 แสดงตัวอย่างรูปแบบของ Query String	205
10.2 สร้างฟอร์มสำหรับการ Login กรณีศึกษาและประยุกต์ใช้งานเช斯ชั้น	217
12.1 เปรียบเทียบการทำงานแบบเดิม กับ AJAX	239
12.2 โครงสร้างของ AJAX ในปัจจุบัน	240
12.3 หลักการทำงานของ AJAX	241
12.4 ตัวอย่างการสร้าง AJAX dictionary จาก longdo ด้วย jQuery	266
13.1 แสดงลักษณะการทำงานและผลลัพธ์เมื่อใช้ CREATE TABLE ร่วมกับ SELECT...FROM	280
13.2 แสดงลักษณะการทำงานและผลลัพธ์เมื่อใช้การใช้ CREATE TABLE ร่วมกับ SELECT ... FROM แบบมีเงื่อนไข	281
13.3 แสดงความสัมพันธ์ของข้อมูล ในรูปแบบ ER-Diagram สำหรับเก็บข้อมูลนักศึกษา.....	288
13.4 หน้าล็อกอินเพื่อเข้าสู่ระบบของ phpMyAdmin เวอร์ชัน 3.5.4	292
13.5 หน้าเพจหลังจากล็อกอินเข้าสู่ระบบ phpMyAdmin เวอร์ชัน 3.5.4	292
13.6 แสดงขั้นตอนการสร้างฐานข้อมูลใหม่	293
13.7 แสดงขั้นตอนการสร้างตารางใหม่	294
13.8 แสดงขั้นตอนการสร้างขอบเขตข้อมูลในตาราง	294
13.9 แสดงขั้นตอนการสร้างตารางและกำหนดขอบเขตข้อมูลในตาราง	295
13.10 แสดงขั้นตอนการแทรกข้อมูลลงในตาราง	295
13.11 แสดงตัวอย่างรายการสำหรับการแทรกข้อมูลลงในตาราง	295
13.12 แสดงขั้นตอนการอ่านข้อมูลในตาราง	296
13.13 แสดงขั้นตอนการเลือกข้อมูลเพื่อแก้ไขข้อมูล	297
13.14 แสดงขั้นตอนการเลือกข้อมูลเพื่อลบข้อมูล	297
13.15 แสดงตัวอย่างการเลือกข้อมูลเพื่อแก้ไขโครงสร้างข้อมูล	298
13.16 แสดงตัวอย่างการลบข้อมูลในตารางและลบตาราง	298
13.17 แสดงตัวอย่างการใช้คำสอบถามหรือคำสั่งแบบกำหนดเองผ่าน "Run SQL Query"	299
13.18 แสดงตัวอย่างการใช้คำสอบถามผ่าน "Run SQL Query" เพื่อค้นหาข้อมูล	299
13.19 แสดงตัวอย่างการใช้คำสั่งผ่าน "Run SQL Query" เพื่อกำหนดสิทธิ์	300
13.20 แสดงตัวอย่างข้อมูลในไฟล์ Excel	300

สารบัญภาพ (ต่อ)

ภาพที่	หน้าที่
13.21 แสดงตัวอย่างการบันทึกไฟล์ให้เป็นรูปแบบอื่นๆ จากรายการ	301
13.22 แสดงตัวอย่างการบันทึกไฟล์ให้เป็นรูปแบบ CSV (Comma Delimited)	301
13.23 แสดงตัวอย่างการบันทึกไฟล์ให้เป็นรูปแบบ CSV และกำหนด UTF-8 ใน Notepad	302
13.24 แสดงตัวอย่างข้อมูลในตาราง student เป็นผลมาจากการนำเข้ามาจากไฟล์ CVS	303
14.1 หน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูลลูกค้า	320
14.2 แสดงหน้าเว็บเพจรายงานข้อมูลลูกค้าทั้งหมด	322
14.3 หน้าเว็บเพจแสดงข้อมูลลูกค้าที่ต้องการแก้ไข	325
ก-1 ภาระพิธี ปัญญาดี ผู้พัฒนา AppServ	348
ก-2 เริ่มการติดตั้งโปรแกรม AppServ	348
ก-3 แสดงรายละเอียดเงื่อนไขของ GNU License	349
ก-4 เลือกตำแหน่งปลายทางที่เก็บไฟล์ติดตั้งโปรแกรม AppServ	349
ก-5 เลือกชุดติดตั้งที่ต้องการใช้งาน	350
ก-6 แสดงการกำหนดค่าคอนฟิกสำหรับ Apache	350
ก-7 แสดงการกำหนดค่าคอนฟิกสำหรับฐานข้อมูล MySQL	351
ก-8 แสดงการติดตั้งโปรแกรม AppServ และชุดโปรแกรมต่างๆ ตามที่ระบุ	351
ก-9 แสดงขั้นตอนสุดท้ายของการติดตั้งโปรแกรมและตัวเลือกการทำงาน	352
ก-10 แสดงการโหลดโปรแกรม Apache และ MySQL เข้าสู่หน่วยความจำ	352
ก-11 แสดงผลการทำงานของ AppServ บนหน้าเว็บบราวเซอร์	352
ก-12 แสดงไฟล์ และโฟลเดอร์ต่างๆ ของโปรแกรม AppServ	353
ก-13 Kai ‘Oswald’ Seidler ผู้พัฒนา XAMPP	354
ก-14 เริ่มการติดตั้ง XAMPP	354
ก-15 ขั้นตอนการติดตั้ง XAMPP สอบ sama ต้องการสร้าง Shortcuts หรือไม่	354
ก-16 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการทวนถามความถูกต้องของตำแหน่งติดตั้ง	355
ก-17 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการติดตั้ง portable แบบ without drive letters ..	355
ก-18 ขั้นตอนการติดตั้ง XAMPP แสดงผลการดำเนินการติดตั้ง XAMPP แบบปกติ	356
ก-19 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการปรับโฉนเวลาในไฟล์ php.ini และ my.ini	356
ก-20 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการปรับเปลี่ยนการใช้งานเพิ่มเติม	356
ก-21 แสดงขั้นตอนการเปิด XAMPP Control Panel	357
ก-22 แสดงหน้าจอ XAMPP Control Panel เพื่อควบคุมโปรแกรมต่างๆ ที่เกี่ยวข้อง	357

สารบัญภาพ (ต่อ)

ภาคี	หน้าที่
ก-23 แสดงผลการทำงานของ XAMPP บนหน้าเว็บбраузอร์	357
ก-24 แสดงการสร้างโฟลเดอร์เก็บเอกสารเว็บเพจของ XAMPP	358
ข-1 แสดงความสัมพันธ์ของข้อมูล ในรูปแบบ ER-Diagram ระบบบันทึกการทำกิจกรรมของนักศึกษา	365
ข-2 แสดงความสัมพันธ์ของข้อมูลระบบบันทึกการทำกิจกรรมของนักศึกษา	365
ข-3 แสดงหน้าแรกของเว็บไซต์ ใช้ระบุชื่อผู้ใช้และรหัสผ่านเพื่อเข้าสู่ระบบ	370
ข-4 หน้าเว็บเพจหลักสำหรับเจ้าหน้าที่ใช้เชื่อมโยงหน้าจัดการระบบทั้งหมด	373
ข-5 หน้าเว็บเพจสำหรับจัดการข้อมูลนักศึกษา	375
ข-6 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลนักศึกษาใหม่	375
ข-7 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลนักศึกษา	376
ข-8 หน้าเว็บเพจแสดงข้อมูลการทำกิจกรรมของนักศึกษา (รายบุคคล)	376
ข-9 หน้าเว็บเพจแสดงตัวอย่างการเพิ่มข้อมูลการทำกิจกรรม	377
ข-10 หน้าเว็บเพจสำหรับจัดการข้อมูลสาขาวิชา	394
ข-11 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลสาขาวิชาใหม่	394
ข-12 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลสาขาวิชา	395
ข-13 หน้าเว็บเพจสำหรับจัดการข้อมูลคณะ	403
ข-14 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลคณะ	404
ข-15 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลคณะ	404
ข-16 หน้าเว็บเพจสำหรับจัดการข้อมูลโครงการ	410
ข-17 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลโครงการใหม่	411
ข-18 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลโครงการ	411

สารบัญตาราง

ตารางที่	หน้าที่
2.1 แสดงตัวกำหนดชนิดการแสดงผลที่ใช้ร่วมกับคำสั่ง printf	16
2.2 ตัวดำเนินการแปลงชนิดของข้อมูล	22
2.3 พังค์ชันที่ใช้สำหรับตรวจสอบชนิดของข้อมูล.....	28
2.4 คำส่วนของภาษา PHP	31
3.1 แสดงชื่ออวิลิเมนท์ของตัวแปรพิเศษ \$_SERVER ที่สำคัญ และใช้งานบ่อย	44
3.2 ตัวดำเนินการทางคณิตศาสตร์	51
3.3 ตัวดำเนินการกำหนดค่า	53
3.4 ตัวดำเนินการสำหรับเพิ่ม และลดค่า	53
3.5 ตัวดำเนินการเปรียบเทียบ	54
3.6 ตัวดำเนินการเปรียบเทียบทางตระกกะ	55
6.1 แสดงความสัมพันธ์ระหว่างอินเด็กซ์กับค่าของข้อมูลที่กำหนดให้ตัวแปร \$province	99
6.2 แสดงผลสรุปฟังค์ชันที่ใช้สำหรับเรียงลำดับข้อมูลในอาร์เรย์โดยใช้คีย์อาร์เรย์	120
6.3 สรุปฟังค์ชันที่ใช้สำหรับการจัดการตัวชี้ตำแหน่งสมาชิกในอาร์เรย์	121
6.4 แสดงค่าคีย์ของอาร์เรย์ \$_FILE	126
7.1 พังค์ชันที่ใช้จัดการรหัสAESกี	131
7.2 พังค์ชันในการตัดช่องว่าง	143
7.3 พังค์ชันประมาณค่า	145
7.4 พังค์ชันเบรียบเทียบจำนวน	146
7.5 พังค์ชันการตรวจสอบ และจัดรูปแบบตัวเลข	146
7.6 พังค์ชันอื่นๆ ที่เกี่ยวข้องกับตัวเลข	147
7.7 อักษรที่ใช้แทนความหมายของวันและสัปดาห์ สำหรับฟังค์ชัน date ()	150
7.8 อักษรที่ใช้แทนความหมายของเดือน สำหรับฟังค์ชัน date ()	151
7.9 อักษรที่ใช้แทนความหมายของปี สำหรับฟังค์ชัน date ()	151
7.10 อักษรที่ใช้แทนความหมายของเวลา สำหรับฟังค์ชัน date ()	151
7.11 คีย์และค่าของอาร์เรย์ที่ได้จากการใช้ฟังค์ชัน getdate ()	153
8.1 ขั้นตอนการเขียนและอ่านข้อมูลในไฟล์	164
8.2 โหมดไฟล์	165
8.3 พังค์ชันที่ใช้ตรวจสอบเกี่ยวกับไฟล์	177
8.4 พังค์ชันที่ใช้ตรวจสอบเวลาที่เกี่ยวข้องกับไฟล์	178

สารบัญตาราง (ต่อ)

ตารางที่	หน้าที่
8.5 ค่าคีย์ของตัวแปร \$_FILES	184
9.1 แสดงลักษณะการใช้งานวิธีการส่งข้อมูลแบบ POST และ GET	189
9.2 ความสัมพันธ์ระหว่างวิธีการส่งและรูปแบบการเรียกใช้ข้อมูล	197
10.1 สัญลักษณ์ที่ใช้ร่วมกับ Query String เพื่อการส่งค่าระหว่างเว็บเพจ	206
12.1 แสดงอีเวนต์ของ JavaScript สำหรับกำหนดให้กับอัลิเมนต์	255
12.2 แสดงอีเวนต์ของ JavaScript แบบพร็อกอปเพอร์ตี้	258
12.3 แสดงตัวอย่างรหัสที่ใช้แทนปุ่มคีย์บอร์ด	261
13.1 แสดงชนิดข้อมูลของ MySQL ชนิดตัวเลข	272
13.2 แสดงชนิดข้อมูลของ MySQL ชนิดข้อความ	273
13.3 แสดงชนิดข้อมูลของ MySQL ชนิด BLOB	273
13.4 แสดงชนิดข้อมูลของ MySQL ชนิด SET และ ENUM	274
13.5 แสดงชนิดข้อมูลของ MySQL ชนิดวันเวลา	274
13.6 แสดงแอทริบิวต์สำหรับกำหนดให้กับขอบเขตข้อมูลใน MySQL	274
13.7 ตัวอย่างโครงสร้างตาราง user	276
13.8 ตัวอย่างโครงสร้างตาราง employees	277
13.9 แสดงโอเปอร์เรเตอร์ที่ใช้งานร่วมกับ WHERE	285
13.10 แสดงฟังก์ชัน Aggregate สำหรับใช้ร่วมกับ GROUP BY	286
13.11 ตัวอย่างตารางข้อมูลนักศึกษา	289
13.12 ตัวอย่างตารางข้อมูลสาขาวิชา	289
13.13 ตัวอย่างตารางข้อมูลคณะ	289
13.14 แสดงตัวอย่างข้อมูลในตารางข้อมูลนักศึกษา	289
13.15 แสดงตัวอย่างข้อมูลในตารางข้อมูลสาขาวิชา	290
13.16 แสดงตัวอย่างข้อมูลในตารางข้อมูลคณะ	290
14.1 ตัวอย่างโครงสร้างตาราง customer	317
ข-1 ตารางข้อมูลนักศึกษา (student)	366
ข-2 ตารางข้อมูลสาขาวิชา (program)	366
ข-3 ตารางข้อมูลคณะ (faculty)	366
ข-4 ตารางข้อมูลโครงการ (project)	366

สารบัญตาราง (ต่อ)

ตารางที่	หน้าที่
ข-5 ตารางข้อมูลการทำกิจกรรม (activity)	367
ข-6 แสดงรายละเอียดของไฟล์ที่เกี่ยวข้องในระบบบันทึกการทำกิจกรรมของนักศึกษา	368

ចំណី

A

abs, 155
Absolute, 163, 195
Addition, 54
Adobe Dreamweaver, 9, 37
AJAX Engine, 237, 240
AJAX Framework, 242, 245
AJAX, 237, 238, 239, 240, 241,
 242, 243, 244, 245, 246,
 247, 248, 249, 250, 251,
 252, 253, 254, 255, 264,
 265, 268, 270
AJAXCall, 246, 247, 250, 251, 252,
 255, 256
AJAXLoad, 245, 246, 247, 250,
 251, 255
AppServ, 163, 164, 167, 169
argument, 83
Arithmetic Operators, 53
Array Data Types, 22
array pointer, 79
array, 99, 100, 102, 103, 104, 107,
 108, 109, 110, 111, 113,
 114, 115, 116, 117, 118,
 119, 120, 121, 122, 123,
 124, 125, 126, 128, 129,
 130, 131, 132, 133, 135,
 137, 141
array_combine, 128, 131
array_count_values, 121
array_diff, 137
array_intersect, 135
array_key_exists, 133
array_merge, 128, 129, 130

array_merge_recursive, 128, 129,
 130
array_pad, 115, 116
array_pop, 118, 119
array_push, 113, 115
array_search, 133, 135
array_shift, 119
array_slice, 117
array_splice, 117
array_unique, 113
array_unshift, 116
arsort, 124, 125, 126
asort, 124, 125, 126
ASP Style, 11, 37
Assignment, 57
Asynchronous JavaScript and
 XML, 237
Asynchronous, 237, 238, 241
Attribute, 275
AUTO_INCREMENT, 276, 279, 331
AVG, 291

B

BIGINT, 8
Binary Large Object, 9
BINARY, 9, 276
BIT, 7, 9
Bitwise Operators, 60
BLOB, 9
break, 68, 69, 76, 77, 80, 81, 82
Built-in Functions, 83

C

\$_COOKIE, 45, 46, 47, 210, 211
ceil, 153, 169
CHAR, 8
Checkbox, 194

checkdate, 165, 166
 chmod, 186
 chr, 131, 133
 Class, 221, 222, 235, 236
 Client Side script, 1
 Client, 1, 2, 5
 Client-Server, 5
 closedir, 182, 183
 Code, 3, 4
 Comment, 12
 Comparison, 58
 Compound Data Types, 22
 Conditional Structures, 61
 Constants variables, 34
 Constructor, 231
 continue, 80, 82, 83
 Control Structures, 61
 Converting Types Functions, 31
 Cookie, 209
 copy, 138, 140, 189
 count, 106, 107, 108, 119, 120,
 121, 141
 COUNT, 291
 COUNT_RECURSIVE, 120
 CREATE DATABASE, 277
 CREATE TABLE, 321
 current, 128
 CVS, 307, 308, 310

D

Data Sets, 99
 Database, 269, 270, 273, 278, 308
 DATABASE, 277, 298
 date, 156, 157, 158, 159, 161,
 162, 163, 164, 165, 166,
 167, 169, 170
 date_default_timezone_set, 156,
 162, 164
 Decimal, 20

Default Parameter, 89
 Default Syntax, 9, 12, 37
 DEFAULT, 276
 default, 68, 69, 82
 DESC, 270, 271, 272, 281, 292
 DESCRIBE, 270, 271, 272
 Directory, 163, 182, 195
 dirname, 184, 185, 186, 188
 Division, 54
 Document Root, 163, 164, 169
 Dollar Sign, 33
 Double Quotes, 21, 34, 35
 DOUBLE, 8
 DROP, 272
 Dynamic Web page, 1

E

\$_ENV, 45, 46
 each, 107, 108, 109, 128
 echo, 9, 10, 11, 12, 14, 15, 16, 19,
 22, 34, 35, 37
 EditPlus, 9, 37
 EDIUMBLOB, 9
 Element, 45, 99
 empty, 48, 49, 50
 end, 128
 endfor, 61
 endforeach, 61
 endif, 61
 endswitch, 61
 endwhile, 61
 Engine, 4, 6, 14
 ENUM, 9
 Error suppression Operator, 61
 Execute Command Operator, 62
 EXIT, 272
 explode, 139, 140, 173
 Expressions, 52
 Extensible Markup Language, 241



F

\$_FILES, 137, 138, 140, 191, 192,
 193, 194
 Fatal Error, 98, 99
 fclose, 168, 169, 170, 172, 173,
 175, 177, 180, 196
 feof, 173, 174, 175, 177
 fgetc, 173, 174
 fgets, 175
 fgetss, 175, 177
 Field, 270, 272, 309
 File Field, 197
 File Mode, 165
 File Pointer, 166
 file, 167, 168, 169, 170, 171, 172,
 173, 178, 179, 180, 181,
 188, 189, 190, 191, 192,
 193, 194, 195, 196
 file_exists, 170, 177, 188, 196
 file_get_contents, 173, 179, 180
 file_put_contents, 168, 172, 173,
 177
 fileatime, 182
 filectime, 182
 filemtime, 182
 filesize, 175, 178, 179, 191, 194,
 195
 filetype, 183, 195
 Flag, 275
 Flat File, 164
 FLOAT, 8
 floatval, 154
 floor, 154, 171, 172
 fopen, 165, 166, 167, 168, 169,
 170, 171, 172, 173, 174,
 175, 177, 178, 179, 180,
 196

foreach, 80, 81, 103, 104, 123,
 124
 Form, 3
 Formatted String, 16
 fpassthru, 178, 179
 fputs, 168, 170, 171, 172, 196
 fread, 174, 175
 FTP, 167
 Function Libraries, 96
 Function parameters, 39
 Function, 3
 fwrite, 168, 169, 170, 172, 180

G

\$_GET, 45, 46, 199, 207
 GET, 192
 getdate, 163, 164
 getFormData, 246, 250, 251, 255
 gettype, 25, 26, 27, 28, 29, 31, 34,
 35
 Global variables, 39
 GLOBAL, 42, 43
 gmmktime, 166, 167
 GMT, 156, 157, 161, 166
 Grave Accent, 63
 Greenwich Mean Time, 156, 166
 GROUP BY, 288, 291

H

HAVING, 288
 header, 164, 178, 179, 203, 204,
 205, 206, 208, 223, 224,
 225, 226, 248, 249, 251,
 253, 257, 265, 324, 329,
 330
 Hexadecimal, 20
 Hidden Field, 197
 HTML, 1, 4, 5, 6, 9, 11, 12, 14, 37,
 38



HTTP, 3, 4, 5, 46
 HTTP_REFERER, 45
 Hypertext Markup Language, 1
 Hypertext Transfer Protocol, 3

I

iconv, 248
 if ... else Statement, 63
 if ... elseif Statement, 65
 IF NOT EXISTS, 277
 if Statement, 61
 if, 34, 35, 36, 37, 140, 169, 170, 171, 193, 196
 IMAP, 2
 implode, 140, 173, 179
 in_array, 133, 135
 include, 97, 98, 99, 222, 225, 227, 229, 234
 include_once, 97, 98
 Increment & Decrement, 57
 INDEX, 276, 277, 281, 282, 297, 300
 Inheritance, 235
 INSERT INTO, 284, 285, 308
 Instance, 223
 INT, 8
 Internet, 2, 5
 intval, 154, 170
 IP Address, 45
 is_array, 33
 is_bool, 33, 34
 is_float, 33, 155
 is_int, 33, 35, 155
 is_integer, 33
 is_null, 33
 is_numeric, 33, 155
 is_object, 33
 is_readable, 181
 is_real, 33

is_resource, 33, 36
 is_scalar, 33
 is_string, 33
 is_upload_file, 192
 is_uploaded_file, 192
 is_writable, 181
 iset, 48, 49, 51, 210, 211, 214, 215, 224

J

JavaScript, 237, 238, 241, 242, 246, 248, 249, 250, 251, 252, 253, 254, 256, 257, 260, 270

join, 140

K

Key, 99, 104, 127
 ksort, 124, 125, 126, 127

L

LDAP, 2
 LIMIT, 285, 286, 289, 292
 list, 107, 109
 Local Mean Time, 156
 Local variables, 39
 localhost, 163, 305, 307, 308, 309, 310, 311, 312, 313, 315, 316, 317, 318, 319, 324, 326, 329, 330, 331, 332, 335, 339, 340, 342

LONGBLOB, 9

LONGTEXT, 9

Loop Structures, 61

ltrim, 148, 149

M

max, 154
 MAX, 291
 MEDIUMINT, 8
 MEDIUMTEXT, 9



Method, 221, 223
min, 154
MIN, 291
mkdir, 186, 188
mktime, 162, 164, 165, 166, 171
Modifier, 226
Modulus, 54, 57
move_uploaded_file, 138, 140, 192, 194
Multiple-Line C Syntax, 14
Multiplication, 54
MySQL link identifier, 305, 306, 307, 308, 310, 312, 313, 330, 331, 333, 335, 336, 338
MySQL, 2, 3, 7, 9, 10, 269, 272, 273, 274, 275, 276, 281, 284, 296, 297, 298, 30, 306, 307, 308, 309, 310, 311, 312, 313, 320, 321, 323, 326, 329, 330, 331, 333, 335, 336, 338, 339, 341
mysql_affected_rows, 330, 331
mysql_close, 306, 307, 308, 309, 310
mysql_connect, 305, 306, 307, 308, 309, 330, 331, 341
mysql_data_seek, 332, 333
mysql_error, 309, 310, 311, 312, 315, 316, 317, 319, 320, 324, 326, 329, 331, 332, 333, 336, 337, 338, 339, 340, 342
mysql_fetch_array, 318, 319, 335
mysql_fetch_assoc, 312, 314, 325, 326, 333, 335, 338, 342

mysql_fetch_object, 319, 320, 335
mysql_fetch_row, 316, 317, 335, 336
mysql_free_result, 313, 314, 315, 316, 317, 319, 320, 326, 328, 333, 336, 341, 342
mysql_insert_id, 331, 332
mysql_list dbs, 334, 335
mysql_list_fields, 337, 338
mysql_num_rows, 315, 333, 340
mysql_pconnect, 305, 306, 307, 309, 310, 311, 312, 313, 315, 316, 317, 318, 320, 324, 326, 329, 332, 335, 336, 339, 340, 341, 342
mysql_query, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 324, 326, 329, 330, 331, 332, 335, 336, 337, 338, 339, 340, 341, 342
mysql_select_db, 310, 311, 312, 313, 315, 316, 317, 318, 320, 324, 326, 329, 331, 332, 339, 340, 341, 342

N

next, 128
nl2br, 151
NOT NULL, 276, 278, 279, 280, 281, 282, 289
Notepad, 9, 37
Notepad++, 9, 37
NULL, 40, 47, 48, 49, 50, 51
number_format, 155
Numeric Data types, 7

O

Object Data Types, 24



Object Oriented Programming, 2, 221
 Octal, 20
 Open source, 2
 opendir, 182, 183
 Operands, 52
 Operating System, 2
 Operators, 52, 53, 60
 ord, 131, 133
 ORDER BY, 289, 292, 295, 296

P

\$_POST, 45, 46, 198, 199, 200, 201, 202, 214, 223, 224
 page description language, 6
 Parameter, 15
 parse_str, 208, 209
 Passing Parameter by Value and Reference, 91
 pathinfo, 184, 185, 186
 Personal Home Page, 3
 PHP Hypertext Preprocessor, 1, 4
 PHP_SELF, 45
 phpMyAdmin, 269, 296, 297, 298
 PHP-Tools, 3, 4
 POP3, 2
 POST, 191
 pow, 156, 173
 prev, 128
 PRIMARY KEY, 277, 279, 280, 281, 282
 print, 14, 15, 16, 37
 print_r, 100, 102, 103, 105, 106, 110, 111, 113, 115, 116, 117, 118, 119, 121, 130, 131, 132, 135, 137, 141
 printf, 14, 16, 17, 18, 25, 26, 27, 28, 29, 30, 31, 38
 Procedural, 221

Protocol, 2

Q

Query String, 45, 46, 206, 207, 208, 209, 212, 226, 243, 246
 Query, 269, 305, 306
 QUERY_STRING, 45
 QUIT, 272

R

Radio Button, 193
 rand, 155
 range, 100, 103, 104, 105
 readdir, 183
 readfile, 177, 178, 179, 192
 Recursive Functions, 94
 Reference assignment, 34
 Relational Database Management System, 7, 269
 REMOTE_ADDR, 45
 rename, 189, 190
 require, 97, 98, 99, 222
 require_once, 97, 98
 Reserved Words in PHP, 36
 Reset Button, 196
 reset, 128
 Resource, 165, 168, 170
 Retrieving Types Functions, 30
 return, 41, 64, 92, 93, 94, 95, 96
 Returning Multiple Values, 93
 Returning Single Value, 92
 rewinddir, 183
 rmdir, 188
 round, 154
 rsort, 122, 123, 124
 rtrim, 148, 149

S

\$_SERVER, 44, 45, 208



`$_SESSION`, 45, 47, 214, 215, 217, 219, 220, 224, 225, 226
Scalar Data Types, 19
Script Language, 1
Script Style, 10, ๒
SELECT, 282, 283, 284, 286, 287, 288, 289, 290, 291, 292, 293, 295, 296, 312, 314, 315, 316, 317, 318, 320, 324, 326, 332, 337, 339, 340, 342
Semicolon, 12, 37
Server Side script, 1
Server, 1, 2, 5
`session_decode`, 217, 218, 219
`session_destroy`, 215, 221, 226
`session_encode`, 217, 219
`session_id`, 216, 217
`session_is_registered`, 220, 221
`session_regenerate_id`, 217
`session_register`, 219, 220, 221, 223, 226
`session_start`, 47, 213, 214, 215, 217, 219, 220, 221, 223, 224, 226
`session_unregister`, 219, 220
`session_unset`, 220
SET, 9
settype, 31, 32
SGML Style, 10, 37
Shell Syntax, 13
Short Tags, 10, 37
SHOW DATABASES, 269, 277
SHOW TABLES, 270
SID, 212, 216, 217
Single Quotes, 21, 34, 35
Single-Line C++ Syntax, 13
`sizeof`, 120, 121, 141

SMALLINT, 8
SNMP, 2
sort, 113, 122, 123, 124, 125
SORT_FLAG_CASE, 122, 123
SORT_NATURAL, 122, 123
SORT_NUMERIC, 113, 122
SORT_REGULAR, 113, 122, 124, 125
SORT_STRING, 113, 122
Source Code, 4
`sprintf`, 14, 18, 19, 312, 313, 314, 315, 316, 317, 318, 320, 324, 326, 329
`sqrt`, 156
Statement, 1, 2
Static variables, 39
Static Web page, 1
STATIC, 44
STD, 291
STDDEV, 291
`str_ireplace`, 146, 147
`str_pad`, 149, 151, 169
STR_PAD_BOTH, 150, 151
STR_PAD_LEFT, 150, 151, 169
STR_PAD_RIGHT, 149, 150
`str_replace`, 144, 146, 147
`str_word_count`, 134, 135
String Concatenation, 55
String, 131
`strpos`, 144
`strlen`, 133, 134, 143, 149
`strpos`, 143, 144
`strrstr`, 140, 141, 142
`strtolower`, 135, 137, 139
`strtotime`, 157, 158, 159, 162, 169, 174
`strtoupper`, 136, 137
`strr`, 147



Structure Query Language, 269
 Structure Query Language, 7
 Submit Button, 196
 substr, 142
 substr_count, 142, 143
 Subtraction, 54
 SUM, 291
 Super Global variables, 39
 switch Statement, 67

T

Table, 273, 299
 Ternary Operator, 61
 Text File, 164
 TEXT, 9
 time, 157, 158, 159, 163, 164,
 166, 169, 171
 TIME, 8
 Timestamp, 164, 182
 Timezone, 156
 TINYBLOB, 9
 TINYINT, 8
 TINYTEXT, 9
 tis-620, 248, 249, 251, 253, 257,
 265
 trim, 148, 149
 Type-Related Functions, 30

U

ucfirst, 137
 ucwords, 137, 139, 173
 UNIQUE, 276, 279, 280, 281, 282
 Unix Timestamp, 157, 164
 unlink, 180
 unset, 48, 50, 51, 117, 118, 215,
 220
 UPDATE, 285, 286, 314, 320, 329,
 330
 Upload, 9

URL, 45, 46, 163, 167, 203, 204,
 206, 207, 208, 209, 212
 USE, 270, 278
 User Defined Functions, 83
 UTC, 156
 utf-8, 248

V

Value assignment, 33
 Value, 99, 126
 var_dump, 102
 VARBINARY, 9
 VARCHAR, 8
 Variable Character Field, 8
 Variable variables assignment, 34
 Variable, 33, 34, 39
 Version, 4
 vi, 9, 37

W

Web Browser, 1
 Web Client, 5
 web content, 206
 Web page, 1
 Web Site, 1
 Webpage Redirect, 203, 204
 WHERE, 282, 283, 285, 286, 288,
 289, 290, 295, 296
 whitespace, 149

wordwrap, 153

X

XML Style, 9, 12, 37
 XML, 237, 238, 240, 241, 242

Y

YEAR, 8

ก

การกำหนดตัวแปรค่าคงที่, 37
 การเข้าถึงข้อมูลภายในอาร์เรย์, 106
 การเขียนข้อมูลลงไฟล์, 168



- ก**
- การเขียนโปรแกรมเชิงวัตถุ, 24, 221
 - การจัดการกับไฟล์, 164
 - การปิดไฟล์, 167
 - การลบไฟล์, 180
 - การส่งผ่านค่าพารามิเตอร์แบบส่งผ่านค่าแบบปกติ, 91
 - การส่งผ่านค่าพารามิเตอร์แบบอ้างอิง, 92
 - การสืบทอด, 235
 - การอ้างทำาหน่งของอินเด็กซ์, 106
 - การอ่านข้อมูลจากไฟล์, 173
 - กำหนดค่าตัวแปร, 33
 - กำหนดค่าตัวแปรช้อนตัวแปร, 34, 36
 - กำหนดค่าตัวแปรโดยการอ้างอิง, 34, 36
 - กำหนดค่าตัวแปรปกติ, 33
- ข**
- ข้อความ, 131, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 151, 153, 154, 155, 157, 158, 168, 169, 170, 171, 172, 173, 175, 179, 208, 209, , 253, 257, 258, 265, 267, 273, 274, 302, 303, 304, 308, 309, 312, 313, 314, 316, 336
 - ขอบเขตของตัวแปร, 39
 - ขอบเขตของฟอร์ม, 189
 - ขอบเขตข้อมูล, 270, 273
 - ข้อมูลชนิดข้อความ, 9, 21
 - ข้อมูลชนิดตัวเลข, 7
 - ข้อมูลชนิดบูลีน, 19, ดู
 - ข้อมูลชนิดไบนารี, 9
 - ข้อมูลชนิดเลขจำนวนจริง, 21
 - ข้อมูลชนิดเลขจำนวนเต็ม, 20
 - ข้อมูลชนิดเลขฐานสอง, 9
 - ข้อมูลที่เป็นกลุ่มของข้อมูล, 9
- ค**
- คลาส, 24, 25, 96, 98, 99, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 233, 234, 235, 236, 237, 238
 - คอนสตรัคเตอร์, 231
 - คัดลอกไฟล์, 189
 - คำนวนอายุ, 171
 - คำส่วนของภาษา PHP, 36
 - คำสอบถามหรือคำสั่ง, 269
 - คำอธิบาย, 12, 13, 14
 - คีย์, 22, 24, 78, 79, 99, 102, 105, 106, 107, 108, 109, 110, 111, 112, 121, 122, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 137, 138, 140, 147, 163, 184, 191, 210, 214, 314, 318, 321
 - คุกกี้, 203, 209, 210, 211, 212, 214, 226
 - คูณ, 40, 41, 42, 43, 52, 54, 55, 57
 - เครื่องหมายวงเล็บก้ามปู, 100, 101, 102
 - โครงสร้างควบคุม, 20, 34, 61, 70, 84
 - โครงสร้างเงื่อนไข if ... else, 63, 64, 65
 - โครงสร้างเงื่อนไข if ... elseif, 65, 66
 - โครงสร้างเงื่อนไข if, 34, 61, 63, 64, 65, 67, 69, 70, 167, 169, 171, 221, 224
 - โครงสร้างเงื่อนไข switch, 67, 68, 69, 70, 82
 - โครงสร้างเงื่อนไขทำซ้ำ do ... while, 72, 73
 - โครงสร้างเงื่อนไขทำซ้ำ for, 73, 74, 75, 76, 77, 78, 108



โครงสร้างเงื่อนไขทำซ้ำ foreach, 79, 81
 โครงสร้างเงื่อนไขทำซ้ำ while, 70, 71, 72, 82, 83, 174, 314
 โคลอนต์, 1, 2, 10, 137, 237, 238, 239, 240
 โคลอนต์ไซด์สคริปต์, 1

ก

จัดการกับไดเรกทอรี, 182
 จดรูปแบบตัวเลข, 155

ข

ชนิดข้อมูลแบบชุด, 22
 ชนิดข้อมูลพื้นฐาน, 19
 ชนิดข้อมูลวัตถุ, 24
 ชนิดข้อมูลสำหรับกำหนดให้กับขอบเขต
 ข้อมูล, 273
 ช่องรับข้อความบรรทัดเดียว, 190
 ช่องรับข้อความรหัสผ่าน, 191
 ช่องรับข้อความหลายบรรทัด, 190

ย

ชอร์สโค้ด, 4
 เชสชัน, 47, 203, 212, 214
 เชิร์ฟเวอร์,
 เชิร์ฟเวอร์, 1, 2, 3, 4, 5, 6, 10, 167,
 191, 192, 194, 195, 237,
 238, 239, 240, 242, 243,
 245, 246, 247, 248, 249,
 250, 251, 252, 254, 257,
 258
 เชิร์ฟเวอร์ไซด์สคริปต์, 1, 2

ฉ

ฐานข้อมูล, 269, 270, 272, 273, 276,
 277, 278, 293, 296, 298,
 299, 301, 302, 304, 306,
 308, 309, 310, 311

ด

ดาวน์โหลด, 4

岱นามิกเว็บเพจ, 1, 6, 10
 ไดเรกทอรี, 163, 164, 165, 167, 182,
 183, 184, 185, 186, 188,
 189, 191, 192, 195, 196

ต

ตรวจสอบและยกเลิกตัวแปร, 48
 ตัวกำหนดชนิดการแสดงผลที่ใช้ร่วมกับ
 คำสั่ง printf, 17
 ตัวชี้ตำแหน่งข้อมูลในไฟล์, 166, 171
 ตัวชี้ตำแหน่งสมาชิกในอาร์เรย์, 79
 ตัวดำเนินการ, 33, 52, 53
 ตัวดำเนินการกำหนดค่า, 53, 57
 ตัวดำเนินการเชื่อมต่อข้อมูล, 53, 55
 ตัวดำเนินการใช้เงื่อนไขในการ
 กำหนดค่า, 61, 62
 ตัวดำเนินการทางคณิตศาสตร์, 53, 54
 ตัวดำเนินการเปรียบเทียบ, 53, 58
 ตัวดำเนินการเปรียบเทียบทางตระกูล,
 53, 60
 ตัวดำเนินการแปลงชนิดของข้อมูล, 25
 ตัวดำเนินการเพิ่ม และลดค่า, 53, 57
 ตัวดำเนินการรับการเกิดข้อผิดพลาด,
 61, 62, 63
 ตัวดำเนินการเรียกใช้คำสั่งของ
 ระบบปฏิบัติการ, 62
 ตัวถูกกระทำ, 52
 ตัวแปร, 33, 49, 57, 63, 64, 65, 67,
 68, 69, 70, 71, 73, 74
 ตัวแปรคงที่, 39, 43, 44
 ตัวแปรเฉพาะที่, 39, 40, 44
 ตัวแปรที่มีค่า NULL, 39, 47
 ตัวแปรพิเศษ, 39, 43, 44, 45, 46, 47,
 200
 ตัวแปรฟังก์ชันพารามิเตอร์, 39, 40, 41
 ตัวแปรสาธารณะ, 39, 42, 43
 ตัวเลข, 131, 149, 166, 172
 ตัวเลือกที่เลือกตอบได้เพียงข้อเดียว,
 193



- ตัวเลือกแบบ Menu, 193
- ตาราง, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 288, 289, 290, 291, 293, 294, 295, 296, 298, 299, 300, 301, 302, 303, 304, 305, 308, 310, 311
- ท**
- เท็กซ์ไฟล์,
เท็กซ์ไฟล์, 100, 105, 106, 164, 165, 166, 169, 171, 202
เที่ยบเวลาวินาที, 172
แทรกข้อมูลลงในตาราง, 284
- น**
- นำเข้าข้อมูลจากไฟล์, 197
นำเข้าข้อมูลแบบช่อง, 197
นิพจน์, 33, 51, 52, 53, 54, 58, 60, 61, 83, 93, 94, 206
- บ**
- บวก, 36, 52, 54, 55, 57, 58
บันทึกข้อมูลลงไฟล์, 165
- ป**
- ปฏิทินออนไลน์, 169, 171
ปรับปรุงข้อมูลในตาราง, 285
ปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ, 29
- ปีกติสุรทิน, 154
ปีอธิกสุรทิน, 154
ปุ่มตัวเลือกแบบปุ่มกลม, 193
ปุ่มตัวเลือกแบบปุ่มเหลี่ยม, 194
ปุ่มล้างข้อมูล, 196
ปุ่มส่งข้อมูล, 196
ปุ่มส่งค่าข้อมูลแบบรูปภาพ, 197
ปุ่มสঁงงาน, 195
เปลี่ยนแปลงชื่อไฟล์, 189
- เปิดไฟล์, 165
เปิดไฟล์สำหรับการเขียน, 165
เปิดไฟล์สำหรับการเขียนและอ่าน, 166
เปิดไฟล์สำหรับการเพิ่ม, 166
เปิดไฟล์สำหรับการอ่าน, 165
เปิดไฟล์สำหรับการอ่านและเขียน, 165
แปลงค่าข้อมูลชนิดเลขจำนวนจริงเป็นเลขจำนวนเต็ม, 26
แปลงค่าข้อมูลปกติเป็นข้อมูลชนิดวัตถุ, 28
- แปลงค่าข้อมูลปกติเป็นข้อมูลชนิดอาร์เรย์, 28
โปรแกรมเชิงวัตถุ, 2, 3
ໂປຣໂຕຄອລ, 2, 4, 5, 46
- พ**
- พร็อปเพอร์ตี้, 221
พารามิเตอร์, 15, 17, 33, 39, 40, 41, 80, 81, 82, 83, 84, 85, 88, 89, 90, 91, 95, 99, 104, 113, 116, 120, 122, 124, 125, 135, 147, 159, 163, 164, 165, 166, 169, 171, 186, 204, 205, 208, 210, 225, 228, 231, 292, 308, 313
พารามิเตอร์แบบกำหนดค่าเริ่มต้น, 89
พารามิเตอร์แบบส่งผ่านค่าโดยการอ้างอิง, 91
- ฟ**
- ฟอร์ม, 3, 9, 46, 138, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 202, 207, 221, 222, 223, 243, 245, 246, 257, 258, 301, 303, 323
ฟังก์ชันค้นหาข้อมูล, 143



ฟังก์ชันที่เกี่ยวกับการคำนวนทางคณิตศาสตร์, 83
 ฟังก์ชันที่เกี่ยวกับการจัดการข้อมูล,
 83
 ฟังก์ชันที่เกี่ยวกับการติดต่อกับฐานข้อมูล, 83
 ฟังก์ชันที่เกี่ยวกับวันที่และเวลา, 83
 ฟังก์ชันที่เกี่ยวข้องกับชนิดข้อมูล, 30
 ฟังก์ชันที่ใช้จัดการข้อมูล, 131
 ฟังก์ชันที่ใช้จัดการจำนวนและตัวเลข,
 153
 ฟังก์ชันที่ใช้จัดการรหัสแอสกี, 131
 ฟังก์ชันที่ใช้จัดการวันและเวลา, 156
 ฟังก์ชันที่ใช้จัดการอักษรพิเศษของ
 HTML, 151
 ฟังก์ชันที่ใช้สำหรับตรวจสอบชนิดของ
 ข้อมูล, 33
 ฟังก์ชันที่ใช้สำหรับตัดช่องว่างและเติม
 ข้อความ, 148
 ฟังก์ชันที่ใช้สำหรับแยกและรวม
 ข้อความ, 139
 ฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง, 83,
 85, 86
 ฟังก์ชันที่มีการส่งผ่านค่าพารามิเตอร์,
 85
 ฟังก์ชันที่ไม่มีการส่งผ่านค่าพารามิเตอร์
 , 85
 ฟังก์ชันแทนที่ข้อความ, 144
 ฟังก์ชันแบบเรียกตัวเอง, 94
 ฟังก์ชันแปลงชนิดข้อมูล, 31
 ฟังก์ชันพารามิเตอร์, 89
 ฟังก์ชันมาตรฐาน, 83, 85
 ฟังก์ชันเรียกดูชนิดของข้อมูล, 30
 ฟังก์ชันไลบรารี, 96
 ฟังก์ชันสำหรับการค้นหาอินเด็กซ์, 133
 ฟังก์ชันสำหรับการรวมอาร์เรย์, 128
 แฟล็ก, 275

ก

ภาษา C, 2, 3, 12
 ภาษา HTML, 1, 2, 3, 4, 6
 ภาษา Perl, 2
 ภาษา SQL, 7
 ภาษา XML, 12
 ภาษาสคริปต์, 1, 2, 10

ม

เมอร์อด, 223
 โมดิไฟล์เออร์ แบบ public และ
 private, 226
 โมดิไฟล์เออร์, 226
 โมดิไฟล์เออร์แบบ static, 233

ย

ย้ายไดเรกทอรี, 189
 ย้ายหน้าเว็บเพจ, 204
 ย้ายหน้าเว็บเพจแบบ refresh, 205

ร

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์, 7
 ระบบปฏิบัติการ, 2, 3

ล

ลบ, 54, 55
 ลบข้อมูลในตาราง, 286
 ไลบรารีฟังก์ชัน, 83

ว

วันและเวลา, 131
 วางแผนดำเนินการ, 58
 เว็บ, 5
 เว็บแแคช, 206
 เว็บไซต์เด็นท์, 5
 เว็บเซิร์ฟเวอร์, 2, 4, 5, 6, 9, 44, 137,
 189, 203, 212
 เว็บไซต์, 1, 3, 4, 5, 9, 46, 47
 เว็บบราวเซอร์, 1, 2, 4, 5, 9, 10, 11,
 14, 47, 189, 200



- เว็บเพจ, 1, 4, 5, 6, 9, 10, 12, 45, 46, 93, 97, 99, 139, 164, 200, 203, 222, 224, 242, 323
 เว็บเพจแบบคงที่, 1, 6
 เว็บแอพพลิเคชัน, 189, 225, 241, 306, 330, 341
 เวอร์ชัน, 4
- ศ**
 เชษฐากรหาร, 54
- ส**
 ส่งค่ากลับแบบหลายค่า, 93
 ส่งค่ากลับหนึ่งค่า, 92
 สร้างฐานข้อมูล, 277
 สร้างตาราง, 277
 สร้างและลบไดเรกทอรี, 186
 สอบความเพื่อเรียกดูข้อมูลในตาราง, 286
 แสดงข้อผิดพลาด, 98
- ห**
 หาร, 54, 55, 57
 ใหม่ไฟล์, 165
- อ**
 ขอบเจ็กต์, 221, 223, 231, 233, 235, 237
 อักษรที่ใช้แทนความหมายของปี, 161
 อักษรที่ใช้แทนความหมายของเวลา, 161
 อัปโหลด, 9, 137, 138, 139, 140, 191, 192, 194
 อกิมเมนต์, 83
 อ้างอิงตำแหน่งไฟล์และไดเรกทอรี, 163
 อ้างอิงพาราเบน Absolute, 163, 164
 อ้างอิงพาราเบน Relative, 164
 อ่านข้อมูลจากไฟล์, 165
- อาร์เรย์, 19, 22, 24, 25, 28, 32, 33, 43, 45, 78, 79, 80, 81, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 133, 134, 135, 137, 139, 140, 141, 145, 147, 154, 163, 171, 172, 179, 184, 185, 186, 210, 214, 308, 309, 317, 318
 อินเด็กซ์, 99, 100, 101, 103, 104, 105, 106, 107, 121, 123, 127, 128, 130, 131, 133, 134, 140, 186, 317, 318
 อินเทอร์เน็ต, 5
 อินสแตนซ์, 223, 224, 229, 230, 233, 234, 236, 237
 อิลิเมนต์, 45, 99, 140
 อีเวนต์, 257
 อีเวนต์เกี่ยวกับคีย์บอร์ด, 262
 อีเวนต์เกี่ยวกับมาส์, 261
 แอททริบิวต์, 191, 275, 278
 โอเพ่นซอร์ส, 2, 7
- ย**
 เยดเดอร์, 203



บทที่ 1

บทนำ

ในช่วงแรกๆ ภาษาที่นิยมใช้ในการพัฒนาเว็บไซต์ (Web Site) คือ ภาษา HTML (Hypertext Markup Language) โดยภายในหน้าเว็บเพจ (Web page) จะประกอบไปด้วย ตัวอักษร ภาพ หรืออื่นๆ ที่ไม่สามารถเปลี่ยนแปลงข้อมูลได้ด้วยตัวเอง ไม่สามารถคำนวณ ประมวลผลข้อมูล หรือเชื่อมต่อกับฐานข้อมูลได้ เป็นแค่ส่วนของการแสดงผลของหน้าเว็บเพจเท่านั้น หรือเรียกว่า หน้าเว็บเพจแบบคงที่ (Static Web page) ต่อมาได้พัฒนาภาษาที่ข้อมูลจะถูกเปลี่ยนแปลงโดยอัตโนมัติตามเงื่อนไขต่างๆ ที่ผู้เขียนกำหนดไว้ หรือเรียกว่า ไนามิกเว็บเพจ (Dynamic Web page) ในปัจจุบันมีหลายภาษาที่ถูกพัฒนามา มีลักษณะเป็นภาษาสคริปต์ (Script Language) ที่สามารถติดต่อกับผู้ใช้ ฐานข้อมูล คำนวณ และประมวลผลตามเงื่อนไขต่างๆ ได้ และหนึ่งในภาษาสคริปต์ที่ได้รับความนิยมในปัจจุบันก็คือ ภาษา PHP (PHP Hypertext Preprocessor)

1.1 ประเภทของภาษาสคริปต์

ภาษาสคริปต์ มี 2 ประเภท ประกอบด้วย

1.1.1 คลาเอนต์ไซด์สคริปต์ (Client Side script)

คลาเอนต์ไซด์สคริปต์ คือ เทคโนโลยีที่สคริปต์มีการประมวลผลและแปลคำสั่ง (Statement Translation) ในฝั่งของเครื่องผู้ใช้ เช่น JAVA สคริปต์ ส่วนมากจะใช้วิธีการเขียนสคริปต์ แทรกลงไประบบภาษา HTML และถูกประมวลผลโดยเว็บбраузอร์ (Web Browser) เพื่อโടိตอบกับผู้ใช้งาน เช่น การแสดงข้อความเตือน การตรวจสอบข้อมูลที่ผู้ใช้ป้อน เป็นต้น

1.1.2 เซิร์ฟเวอร์ไซด์สคริปต์ (Server Side script)

เซิร์ฟเวอร์ไซด์สคริปต์ คือ เทคโนโลยีที่สคริปต์ที่มีการประมวลผลและแปลคำสั่งในฝั่งของเครื่องเซิร์ฟเวอร์ (Server) แล้วแปลงเป็นเอกสารในรูปแบบของภาษา HTML แล้วส่งผลลัพธ์ที่ได้กลับไปให้เว็บбраузอร์ในฝั่งคลาเอนต์ (Client)

1.2 ความสามารถของภาษา PHP

PHP เป็นภาษาสคริปต์ที่มีการประมวลผลและแปลคำสั่งประเภทเซิร์ฟเวอร์ไซด์สคริปต์ มีรูปแบบในการเขียนคำสั่ง (Statement) การทำงานคล้ายกับภาษา Perl หรือภาษา C และสามารถทำงานร่วมกับภาษา HTML ได้เป็นอย่างดี ความสามารถของภาษา PHP มีรายละเอียด ดังต่อไปนี้

1.2.1 เป็นตัวแปลภาษาที่มีลักษณะ ที่เรียกว่า โอเพ่นซอร์ส (Open source) คือ ผู้ใช้สามารถดาวน์โหลดและติดตั้งบนเว็บเซิร์ฟเวอร์ (Web Server) ได้โดยไม่เสียค่าใช้จ่าย

1.2.2 เป็นภาษาสคริปต์ประเภทเซิร์ฟเวอร์ใช้สคริปต์ มีการประมวลผลและแปลคำสั่งในฝั่งของเว็บเซิร์ฟเวอร์ แล้วแปลงเป็นเอกสารในรูปแบบของภาษา HTML แล้วส่งผลลัพธ์ที่ได้กลับไปให้เว็บบราวเซอร์ในฝั่งไคลเอนต์

1.2.3 สามารถทำงานบนระบบปฏิบัติการ (Operating System) ที่หลากหลาย เช่น Unix, MS-Windows, Mac OS หรือ Risc OS อย่างมีประสิทธิภาพ

1.2.4 สามารถทำงานร่วมกับเว็บเซิร์ฟเวอร์ เช่น Personal Web Server (PWS), Apache, OmniHttpd และ Internet Information Service (IIS) เป็นต้น

1.2.5 สามารถเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้

1.2.6 สามารถทำงานร่วมกับโปรแกรมระบบจัดการฐานข้อมูลที่หลากหลาย เช่น Oracle, MySQL, FilePro, Solid, FrontBase, mSQL และ MS SQL Server เป็นต้น

1.2.7 สามารถติดต่อและทำงานร่วมกับโปรโตคอล (Protocol) ชนิดต่างๆ ได้ เช่น LDAP, IMAP, SNMP, POP3 และ HTTP (Hypertext Transfer Protocol) เป็นต้น

1.3 ประวัติความเป็นมาของภาษา PHP



Rasmus Lerdorf

Zeev Suraski

Andi Gutmans

ภาพที่ 1.1 ผู้ร่วมพัฒนา PHP

ที่มา: Rasmus. (2011)., Markbuchner. (2010). และ Peter Judge. (2010).

ภาษา PHP ถูกสร้างขึ้นในปี ค.ศ.1994 โดย Rasmus Lerdorf เป็นผู้เริ่มต้นพัฒนา จุดเริ่มต้นมาจากการต้องการที่จะบันทึกข้อมูลผู้ที่เข้ามาเยี่ยมชมเว็บไซต์ส่วนตัวที่สร้างขึ้น โดยแนวคิดคือ การเขียนโปรแกรมด้วยภาษา C แต่ต้องการแยกส่วนที่เป็นภาษา HTML ออกจากภาษา C และเป็นเหตุผลสำคัญทำให้สร้างโค้ด (Code) ของภาษา HTML ขึ้นมาใหม่ และตั้งชื่อว่า Personal Home Page (PHP-Tools)

หลังจากพัฒนาภาษา PHP ขึ้นมาแล้ว ก็เริ่มแจกลงให้บุคคลทั่วไปใช้งาน และในช่วงแรกภาษา PHP ยังไม่มีความสามารถมากนัก ต่อมาในปี ค.ศ.1995 มีการพัฒนาเพิ่มขีดความสามารถให้กับภาษา PHP ให้สามารถรับข้อมูลที่ส่งมาจากฟอร์ม (Form) ของภาษา HTML รวมทั้งสามารถติดต่อกับฐานข้อมูล MySQL ได้อีกด้วย

ในปี ค.ศ.1997 ได้เปิดให้ผู้สนใจเข้าร่วมพัฒนาภาษา PHP และได้ผู้ร่วมพัฒนาเพิ่มอีก 2 คน คือ Zeev Suraski และ Andi Gutmans ได้ช่วยกันปรับปรุงและพัฒนาโค้ดขึ้นมาใหม่ให้ดีขึ้นในหลายๆ ด้าน ทั้งแก้ไขข้อบกพร่อง เพิ่มประสิทธิภาพ และเพิ่มฟังก์ชัน (Function) ใหม่ๆ มากขึ้น และสามารถเขียนโปรแกรมเชิงวัตถุที่สมบูรณ์แบบ สามารถทำงานร่วมกับเว็บเซิร์ฟเวอร์ บนระบบปฏิบัติการหลายชนิด จนเป็นที่นิยมในปัจจุบัน และเปลี่ยนชื่อใหม่จากเดิม Personal Home Page (PHP-Tools) เป็น PHP Hypertext Preprocessor

ปัจจุบันภาษา PHP ได้ผ่านการพัฒนามากหลายเวอร์ชัน (Version) มีเว็บไซต์เพื่อให้ผู้ใช้หรือผู้พัฒนาเว็บไซต์สามารถเข้าไปดาวน์โหลดโปรแกรมเวอร์ชันล่าสุด รวมทั้งค้นหาข้อมูล วิธีการใช้งาน ฟังก์ชัน และสนทนากันระหว่างผู้ใช้ด้วยกัน เว็บไซต์ต้องอยู่ที่ <http://php.net>

1.4 หลักการทำงานของภาษา PHP

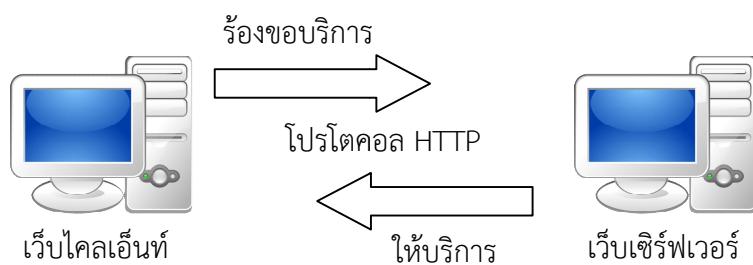
ภาษา PHP นั้น ทำงานอยู่ในฝั่งเว็บเซิร์ฟเวอร์ จะไม่สามารถเห็นสคริปต์ของภาษา PHP ที่เครื่องคอมพิวเตอร์ในฝั่งไคลเอ็นท์ เพราะระหว่างที่ใช้งานเว็บбраузरเรียกชมเว็บไซต์นั้น ในฝั่งของ เว็บเซิร์ฟเวอร์ จะทำการประมวลผลภาษา PHP และส่งผลลัพธ์ที่ได้ เป็นภาษา HTML หลังจากนั้นจะส่ง HTML ที่ได้จากการประมวลผลแล้วนั้นมาจังเว็บбраузரของไคลเอ็นท์ แล้วแสดงผลให้เห็นเป็นรูปร่าง ของเว็บไซต์ และเมื่อต้องรหัสโคด (Source Code) ในฝั่งไคลเอ็นท์ จะเห็นเฉพาะ HTML tags ของภาษา HTML เพียงอย่างเดียว ไม่เห็นสคริปต์ PHP สามารถสรุปขั้นตอนและหลักการทำงานระหว่างไคลเอ็นท์ และเว็บเซิร์ฟเวอร์ ได้ดังต่อไปนี้

1.4.1 เว็บเซิร์ฟเวอร์ คือ โปรแกรมที่อยู่และทำงานบนเครื่องฝั่งเซิร์ฟเวอร์ ทำหน้าที่ในการรับคำสั่งจากการร้องขอของไคลเอ็นท์ (โดยผู้ใช้จะเรียกชื่อหน้าเว็บเพจได้โดยใช้โปรโตคอล HTTP ผ่านทาง เว็บбраузอร์) และประมวลผลการทำงานจากการร้องขอตั้งกล่าว แล้วส่งข้อมูลกลับไปยังเครื่องของ ไคลเอ็นท์ที่ร้องขอ โดยสรุปคือ เว็บเซิร์ฟเวอร์จะคอยให้บริการแก่ ไคลเอ็นท์ ที่ร้องขอข้อมูลเข้ามาโดยผ่าน เว็บбраузอร์ หรือ จากการร้องขอผ่านโปรโตคอล HTTP โดยเว็บเพจที่เขียนก็สามารถเขียนได้หลาย ภาษา หลายรูปแบบ การเขียนสคริปต์ PHP ระบบจะสามารถทำงานได้นั้นจำเป็นต้องมีเว็บเซิร์ฟเวอร์ เป็น ตัวเรียกใช้งาน Engine ของภาษา PHP ดังนั้นถ้าต้องการให้คอมพิวเตอร์สามารถประมวลผลและ แปลภาษา PHP ได้นั้น จำเป็นต้องมีเว็บเซิร์ฟเวอร์หรือกรณีศึกษานี้สามารถจำลองเครื่องคอมพิวเตอร์ ธรรมดาให้ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ (วิธีการติดตั้งสามารถศึกษาเพิ่มเติมได้ในภาคผนวก) รวมถึงรูปแบบ



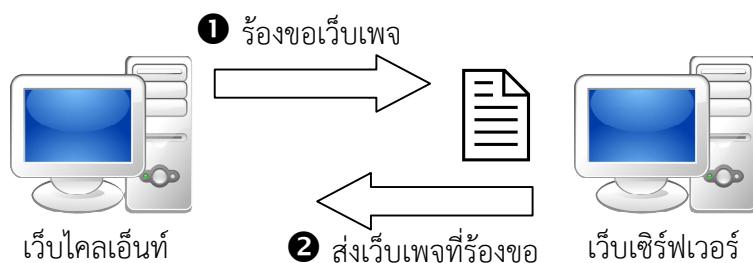
และวิธีการ เข้าถึงข้อมูล จะเรียกใช้ผ่านบริการจากโปรโตคอล HTTP เพื่อให้ผู้ใช้สามารถอ่านข้อมูล ภาพเสียง หรือข้อมูลอื่นๆ ตามต้องการ

1.4.2 หลักการทำงานของเว็บเซิร์ฟเวอร์ มีลักษณะเช่นเดียวกันกับบริการอื่นๆ บนระบบอินเทอร์เน็ต (Internet) คือ อยู่ในรูปแบบโคลเลอืนท์-เซิร์ฟเวอร์ (Client-Server) โดยมีโปรแกรมเว็บโคลเลอืนท์ (Web Client) ทำหน้าที่เป็นผู้ร้องขอบริการ และมีโปรแกรมเว็บเซิร์ฟเวอร์ ทำหน้าที่เป็นผู้ให้บริการ โปรแกรมเว็บโคลเลอืนท์ ก็คือโปรแกรมเว็บбраเวอร์ในเครื่องของผู้ใช้ สำหรับโปรแกรมเว็บเซิร์ฟเวอร์นั้นจะถูกติดตั้งไว้ในเครื่องของผู้ให้บริการเว็บไซต์ (มักเรียกว่า "เว็บเซิร์ฟเวอร์") การติดต่อระหว่างโปรแกรมเว็บราเวอร์กับโปรแกรมเว็บเซิร์ฟเวอร์จะกระทำการผ่านโปรโตคอล HTTP ดังภาพที่ 1.2



ภาพที่ 1.2 รูปแบบการติดต่อสื่อสารระหว่างเว็บโคลเลอืนท์และเว็บเซิร์ฟเวอร์ผ่านโปรโตคอล HTTP

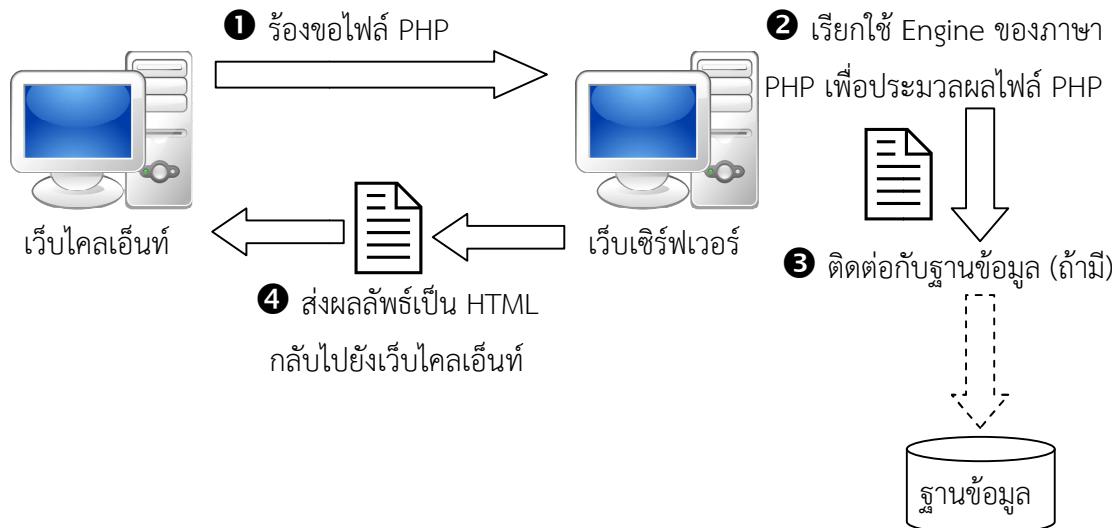
1.4.3 กลไกการทำงานเมื่อร้องขอเว็บเพจธรรมดा โดยปกติจะมีนามสกุลของเอกสารเป็น .htm หรือ .html นั้น เมื่อใช้เว็บโคลเลอืนท์ เปิดดูหรือเรียกดูเว็บเพจใด เว็บเซิร์ฟเวอร์ก็จะส่งเว็บเพจนั้นกลับมายังเว็บโคลเลอืนท์ จากนั้นเว็บโคลเลอืนท์จะแสดงผลตามคำสั่งภาษา HTML ที่อยู่ในเอกสารนั้นๆ



ภาพที่ 1.3 รูปแบบการติดต่อสื่อสารระหว่างเว็บโคลเลอืนท์และเว็บเซิร์ฟเวอร์เพื่อเรียกเว็บเพจธรรมดา

จะเห็นได้ว่าเว็บเพจ จากภาพที่ 1.3 เป็นเว็บเพจแบบคงที่ กล่าวคือผู้ใช้จะพบกับเว็บเพจหน้าตาเดิมๆ ทุกครั้ง จนกว่าผู้ดูแลเว็บจะทำการปรับปรุงเว็บเพจนั้นใหม่ ภาษา HTML เป็นภาษาที่ใช้อธิบายหน้าตาของเว็บเพจ (ภาษา HTML จัดเป็นภาษาในกลุ่มที่เรียกว่า page description language) หรือกล่าวอีกนัยหนึ่งคือ ภาษา HTML สามารถกำหนดให้เว็บเพจมีหน้าตาอย่างที่ต้องการได้ ในปัจจุบันมีการพัฒนาโปรแกรมที่ใช้ในการจัดการเว็บเพจ ให้มีความน่าสนใจมากยิ่งขึ้น

1.4.4 กลไกการทำงานเมื่อร้องขอเว็บเพจที่ใช้ภาษา PHP จะมีรูปแบบการประมวลผลและแสดงผลเป็นแบบไดนามิกเว็บเพจ สามารถติดต่อกับผู้ใช้ ฐานข้อมูล คำนวณ และประมวลผลตามเงื่อนไขต่างๆ ได้ ลักษณะการประมวลผล ดังภาพที่ 1.4

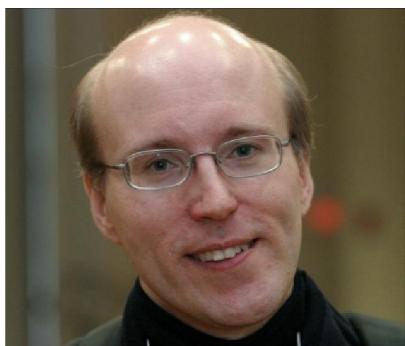


ภาพที่ 1.4 การร้องขอการใช้งานจากเร็ปโคลอี้นท์ไปยังเร็ปเชิร์ฟเวอร์ที่ได้ติดตั้ง Engine ของภาษา PHP

จากภาพที่ 1.4 เป็นกระบวนการประมวลผลภาษา PHP โดยเริ่มต้นเมื่อเว็บไคลเอนท์ร้องขอไฟล์ หรือเอกสารที่ใช้ภาษา PHP ในการพัฒนาร่วมกับภาษา HTML นำยังเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์จะเรียกใช้ Engine ของภาษา PHP หรือตัวแปลคำสั่งของภาษา PHP กรณีที่มีการใช้คำสั่งในการเชื่อมต่อฐานข้อมูลระบบก็จะติดต่อไปยังฐานข้อมูลตามการร้องขอ มีการตรวจสอบสิทธิ์การเข้าถึงฐานข้อมูลและอื่นๆ ตามลำดับขั้นตอนนี้ ลักษณะการแปลคำสั่งของภาษา PHP นั้นจะแปลและเปลี่ยนคำสั่งจากภาษา PHP เป็นภาษา HTML และส่งกลับไปแสดงผลลัพธ์ที่เว็บไคลเอนท์ที่ร้องขอ

1.5 MySQL

MySQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) ชนิดหนึ่ง โดยใช้ภาษา SQL (Structure Query Language) เป็นฐานในการพัฒนา MySQL ถูกพัฒนาขึ้นมา ประมาณปี ค.ศ.1995 MySQL จัดอยู่ในกลุ่มของโอเพ่นซอร์ส แต่แตกต่างจากโอเพ่นซอร์สโดยทั่วไป เนื่องจากถูกพัฒนาภายใต้บริษัท MySQL AB ในประเทศไทย ผลิตภัณฑ์ของบริษัทฯ มีทั้งแบบที่อนญาตให้ใช้งานได้ฟรี และแบบที่ใช้ในเชิงธุรกิจ ทีมพัฒนา MySQL ประกอบด้วย Monty Widenius, David Axmark และ Allan Larsson ทีมงานของชื่อ MySQL มาจากชื่อลูกสาวของ Monty Widenius ชื่อ My และใช้สัญลักษณ์ปลาโลมาเป็นสัญลักษณ์ของ MySQL



David Axmark



Allan Larsson



Monty Widenius

ภาพที่ 1.5 ทีมพัฒนาฐานข้อมูล MySQL

ที่มา: Declan McCullagh. (2007)., Week of Innovative Regions in Europe. (2007).
และ Helsinki, Finland. (2009).

ปัจจุบันบริษัทชั้นไมโครซิสเต็มส์ (Sun Microsystems, Inc.) ได้ซื้อกิจการทั้งหมดของบริษัท MySQL AB ดังนั้นผลิตภัณฑ์ทั้งหมดที่ถูกพัฒนาโดยบริษัท MySQL AB จึงเป็นกรรมสิทธิ์ของบริษัทชั้นไมโครซิสเต็มส์

1.6 ชนิดของข้อมูลที่ MySQL สนับสนุน

1.6.1 ข้อมูลชนิดตัวเลข (Numeric Data types)

1) BIT ใช้สำหรับเก็บข้อมูลชนิดเลขฐานสองที่มีสัญลักษณ์เพียงสองค่า ประกอบด้วย 0 (ศูนย์) กับ 1 (หนึ่ง) บางครั้งอาจหมายถึงค่าของโฉกทางเลือก เช่น ปิดกับเปิด ไม่ใช่กับใช่ เท็จกับจริง ซ้ายกับขวา เป็นต้น

2) TINYINT ใช้สำหรับจัดเก็บข้อมูลชนิดตัวเลขจำนวนเต็ม ขนาด 8 บิต

3) SMALLINT ใช้สำหรับเก็บข้อมูลชนิดตัวเลขจำนวนเต็ม ขนาด 16 บิต

4) MEDIUMINT ใช้สำหรับเก็บข้อมูลชนิดตัวเลขจำนวนเต็ม ขนาด 24 บิต

5) INT ใช้สำหรับเก็บข้อมูลชนิดตัวเลขจำนวนเต็ม ขนาด 32 บิต

6) BIGINT ใช้สำหรับเก็บข้อมูลชนิดตัวเลขจำนวนเต็ม ขนาด 64 บิต

7) FLOAT ใช้สำหรับเก็บข้อมูลชนิดตัวเลขจำนวนจริง (เลขที่มีเศษทศนิยม) ขนาด 32 บิต

8) DOUBLE ใช้สำหรับเก็บข้อมูลชนิดตัวเลขจำนวนจริง เช่นเดียวกับ FLOAT ขนาด 64 บิต

1.6.2 ข้อมูลชนิดวันที่และเวลา (Date and Time Data types)

1) DATETIME ใช้สำหรับจัดเก็บข้อมูลวันเดือนปีและเวลา ขนาด 64 บิต แสดงผลในรูปแบบ YYYY-MM-DD HH:MM:SS

2) DATE ใช้สำหรับจัดเก็บข้อมูลเฉพาะข้อมูลวันเดือนปี ขนาด 24 บิต แสดงผลในรูปแบบ YYYY-MM-DD

3) TIME ใช้สำหรับจัดเก็บข้อมูลเฉพาะช่วงเวลา ขนาด 24 บิต แสดงผลในรูปแบบ HHH:MM:SS โดยที่ HHH จะคืนค่าจาก -838 ถึง 838

4) TIMESTAMP ใช้สำหรับเก็บข้อมูลชนิดวันที่และเวลา เช่นเดียวกับ DATETIME แต่จะเก็บในรูปแบบของ YYYYMMDDHHMMSS หรือ YYMMDDHHMMSS หรือ YYYYMMDD หรือ YYMMDD และแต่ละวันจะระบุค่าแอ็ตทริบิวต์ TIMESTAMP[(M)] โดยกำหนด M เป็น 14 หรือ 12 หรือ 8

5) YEAR ใช้สำหรับเก็บข้อมูลปีเท่านั้น ขนาด 8 บิต แสดงผลในรูปแบบ YYYY มีค่าตั้งแต่ 1901 - 2155 เท่านั้น

1.6.3 ข้อมูลชนิดข้อความ (String Data Types)

1) CHAR ใช้สำหรับเก็บข้อมูลชนิดข้อความ ขนาดสูงสุดได้ไม่เกิน 255 อักขระ ถ้าอักขระที่ใช้ยาวไม่เท่ากับที่กำหนดไว้ อักขระที่เหลือ จะถูกแทนที่ด้วยช่องว่าง

2) VARCHAR (ย่อมาจาก Variable Character Field หมายถึง กลุ่มข้อมูลตัวอักขระที่ไม่สามารถระบุความยาวได้) ใช้สำหรับเก็บข้อมูลชนิดข้อความ ชนิดข้อมูลประเภท varchar สามารถเก็บข้อมูลตัวอักขระขนาดเท่าใดก็ได้ที่ไม่เกินความยาวที่จำกัดไว้ (เก็บข้อมูลตามที่รับมาจริงเท่านั้น) ขนาดสูงสุดได้ไม่เกิน 255 อักขระ

3) BINARY ใช้สำหรับเก็บข้อมูลชนิดเลขฐานสองที่มีสัญลักษณ์เพียงสองตัว เมื่อเป็น BIT ประกอบด้วย 0 (ศูนย์) กับ 1 (หนึ่ง) บางครั้งอาจหมายถึงการที่มีโอกาสเลือกได้เพียง 2 ทาง เช่น ปิดกับเปิด ไม่ใช่กับใช่ เทียบกับจริง ซ้ายกับขวา เป็นต้น

4) VARBINARY ใช้สำหรับเก็บข้อมูลชนิดข้อความ ขนาดสูงสุดไม่เกิน 8,000 อักขระ มีลักษณะการเก็บคล้าย VARCHAR คือ การเก็บข้อมูลตามที่รับมาจริงเท่านั้น

5) TINYTEXT ใช้สำหรับเก็บข้อมูลชนิดข้อความ ขนาดสูงสุดไม่เกิน 255 อักขระ สามารถใช้งานฟีเจอร์ FULL TEXT SEARCH ของ MySQL

6) TEXT ใช้สำหรับเก็บข้อมูลชนิดข้อความ เช่นเดียวกับ TINYTEXT ขนาดสูงสุดไม่เกิน 65,535 อักขระ (64 KB) เหมาะสำหรับเก็บข้อมูลจำนวนมากเนื้อหาที่มีความยาวหรือจำนวนมาก

7) MEDIUMTEXT ใช้สำหรับเก็บข้อมูลชนิดข้อความ เช่นเดียวกับ TINYTEXT ขนาดสูงสุดไม่เกิน 16,777,215 อักขระ (16 MB)

8) LONGTEXT ใช้สำหรับเก็บข้อมูลชนิดข้อความ เช่นเดียวกับ TINYTEXT ขนาดสูงสุดไม่เกิน 4,294,967,295 อักขระ (4 GB)

9) ENUM ใช้สำหรับเก็บเซตของข้อมูลชุดหนึ่งที่มีจำนวนสมาชิกที่กำหนดไว้แน่นอน และทราบค่าทุกตัว มักจะเป็นข้อมูลที่มีลักษณะคงที่ ขนาดสูงสุดไม่เกิน 65,535 อักขระ

10)SET ใช้สำหรับเก็บข้อมูลที่เป็นกลุ่มของข้อมูลที่ยอมให้เลือกได้ 1 ค่าหรือหลายๆ ค่า สามารถกำหนดได้ถึง 64 ค่า

1.6.4 ข้อมูลชนิด BLOB (Binary Large Object)



- 1) TINYBLOB ใช้สำหรับเก็บข้อมูลชนิดไบนาเรีย ได้แก่ ไฟล์ข้อมูลต่างๆ เช่น ไฟล์รูปภาพ ไฟล์มัลติมีเดีย เป็นต้น คือ ไฟล์อะไรก์ตามที่อัปโหลด (Upload) ผ่านฟอร์มอัปโหลดไฟล์ในภาษา HTML ขนาดสูงสุดไม่เกิน 256 ไบต์
- 2) BLOB ใช้สำหรับเก็บข้อมูลชนิดไบนาเรีย เช่นเดียวกับ TINYBLOB ขนาดสูงสุดได้ไม่เกิน 64,000 ไบต์
- 3) EDIUMBLOB ใช้สำหรับเก็บข้อมูลชนิดไบนาเรีย เช่นเดียวกับ TINYBLOB ขนาดสูงสุดได้ไม่เกิน 16 เมกะไบต์ (16 MB)
- 4) LONGBLOB ใช้สำหรับเก็บข้อมูลชนิดไบนาเรีย เช่นเดียวกับ TINYBLOB ขนาดสูงสุดได้ไม่เกิน 4 กิกะไบต์ (4 GB)

สรุป

ภาษา PHP เป็นภาษาประยุกต์เฟอร์ฟิลด์สคริปต์ คือ ประมวลผลและแปลงภาษาในฝั่งของเว็บเซิร์ฟเวอร์ และส่งกลับมายังเว็บбраузரเพื่อแสดงผลในฝั่งของโคลอئินท์ ภาษา PHP เป็นภาษาสคริปต์ที่นิยมใช้ในปัจจุบันและมีแนวโน้มการใช้งานเพิ่มขึ้นในอนาคต สำหรับการพัฒนาด้านมิกเว็บเพจเนื่องจากโครงสร้างภาษา PHP มีความยืดหยุ่นในการเขียนสคริปต์ มีฟังก์ชันการทำงานที่หลากหลายสามารถประยุกต์ใช้งานได้ง่ายสำหรับการติดต่อกับผู้ใช้งานระบบ การประมวลผลตามเงื่อนไข การติดต่อกับระบบไฟล์และไดเรกทอรี และมีความสามารถในการเชื่อมต่อกับระบบฐานข้อมูลที่หลากหลาย แต่ที่ได้รับความนิยมมากที่สุด คือ ฐานข้อมูล MySQL

คำถามท้ายบท

1. จงอธิบายหลักการทำงานของภาษาสคริปต์ประเภทโคลอئินท์สคริปต์
2. จงอธิบายหลักการทำงานของภาษาสคริปต์ประเภทเฟอร์ฟิลด์สคริปต์
3. จงความแตกต่างระหว่างเว็บเพจแบบคงที่ และแบบดینามิก
4. จงอธิบายหลักการทำงานของภาษา PHP เมื่อเว็บโคลอئินท์ร้องขอเว็บเพจที่พัฒนาด้วยภาษา PHP อย่างไรโดยสังเขป
5. จงอธิบายชนิดของข้อมูลที่ MySQL สนับสนุน พร้อมยกตัวอย่างประกอบ

บทที่ 2

PHP เปื้องต้น

เครื่องมือสำหรับสร้างเว็บไซต์เป็นส่วนประกอบสำคัญอย่างหนึ่งในการเขียนสคริปต์ PHP รูปแบบประโยชน์คำสั่งที่ใช้เขียนจะเรียกว่าสคริปต์คำสั่ง (Statement) ลงในเว็บเพจก่อนอพโหลดเว็บเพจทั้งหมดไปยังเซิร์ฟเวอร์ และเรียกดูเว็บเพจที่เครื่องคอมพิวเตอร์ด้วยโปรแกรมเว็บบราวเซอร์ เครื่องมือที่ใช้สำหรับเขียนสคริปต์ PHP สามารถเลือกใช้ได้หลายโปรแกรม เช่น Notepad, Notepad++, EditPlus, vi และ Adobe Dreamweaver เป็นต้น รูปแบบการเขียนสคริปต์ PHP นั้น จะใช้เทคนิควิธีการแทรกประโยชน์คำสั่งร่วมกับ HTML tags และเอกสารตั้งกล่าวจำเป็นต้องบันทึกเป็นชนิด PHP เท่านั้น ตัวอย่างเช่น index.php เป็นต้น

2.1 การแทรกสคริปต์ PHP ในเอกสาร HTML

การแทรกสคริปต์ PHP เพื่อปั่งบอกให้รู้ว่าส่วนที่ระบุนั้นๆ เป็นประโยชน์คำสั่งของภาษา PHP ส่วนใดคือจุดเริ่มต้น และจุดสิ้นสุดของสคริปต์ PHP ที่อยู่ภายใต้เอกสาร HTML การแทรกสคริปต์ PHP ลงในเอกสาร HTML นั้น สามารถแทรกได้ 4 รูปแบบ ดังต่อไปนี้

2.1.1 XML Style หรือ Default Syntax

รูปแบบ

```
<?php  
    คำสั่งที่ 1;  
    คำสั่งที่ 2;  
    ...  
    คำสั่งที่ N;  
?>
```

ตัวอย่างที่ 2.1 การแทรกสคริปต์ PHP แบบ XML Style

```
1 <?php  
2     echo "Hello World ! <br>";  
3     echo "I am PHP";  
4 ?>
```

จากตัวอย่างที่ 2.1 จุดเริ่มต้นของสคริปต์ PHP คือ <?php และตามด้วยชุดคำสั่งที่ต้องการ และจุดสิ้นสุดของสคริปต์ PHP คือ ?> หากไม่ระบุจุดเริ่มต้นและจุดสิ้นสุด โปรแกรมจะไม่สามารถทำงานได้

2.1.2 SGML Style หรือ Short Tags

รูปแบบ

```
<?
    คำสั่งที่ 1;
    คำสั่งที่ 2;
    ...
    คำสั่งที่ N;
?>
```

ตัวอย่างที่ 2.2 การแทรกสคริปต์ PHP แบบ SGML Style

```
1   <?
2       echo "Hello World ! <br>";
3       echo "I am PHP";
4   ?>
```

จากตัวอย่างที่ 2.2 จุดเริ่มต้นของสคริปต์ PHP คือ <? และตามด้วยชุดคำสั่งที่ต้องการ และจุดสิ้นสุดของสคริปต์ PHP คือ ?>

2.1.3 Script Style

รูปแบบ

```
<script language="php">
    คำสั่งที่ 1;
    คำสั่งที่ 2;
    ...
    คำสั่งที่ N;
</script>
```

ตัวอย่างที่ 2.3 การแทรกสคริปต์ PHP แบบ Script Style

```
1   <script language="php">
2       echo "Hello World ! <br>";
3       echo "I am PHP";
4   </script>
```

จากตัวอย่างที่ 2.3 จุดเริ่มต้นของสคริปต์ PHP คือ <script language="php"> และตามด้วยชุดคำสั่งที่ต้องการ และจุดสิ้นสุดของสคริปต์ PHP คือ </script>

2.1.4 ASP Style

รูปแบบ

```
<%  
    คำสั่งที่ 1;  
    คำสั่งที่ 2;  
    ...  
    คำสั่งที่ N;  
%>
```

ตัวอย่างที่ 2.4 การแทรกสคริปต์ PHP แบบ ASP Style

```
1 <%  
2     echo "Hello World ! <br>";  
3     echo "I am PHP";  
4 %>
```

จากตัวอย่างที่ 2.4 จุดเริ่มต้นของสคริปต์ PHP คือ <%> และตามด้วยชุดคำสั่งที่ต้องการ และจุดสิ้นสุดของสคริปต์ PHP คือ %>

การแสดงผลลัพธ์ของทั้ง 4 รูปแบบตามตัวอย่างที่ 2.1-2.4 นั้นจะมีผลลัพธ์เหมือนกัน oriwayดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 แสดงผลคำว่า Hello World !

บรรทัดที่ 3 แสดงผลคำว่า I am PHP

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

เมื่อ
 เป็น HTML tag ใช้สำหรับขึ้นบรรทัดใหม่ ส่วน echo เป็นคำสั่งของภาษา PHP ใช้สำหรับการแสดงผลข้อมูลบนเว็บบราวเซอร์ (รายละเอียดการใช้งานคำสั่ง echo จะแนะนำเป็นลำดับต่อไป) หรืออาจจะแทรกสลับไปมาระหว่างสคริปต์ PHP กับ HTML tags ดังตัวอย่างที่ 2.5

ตัวอย่างที่ 2.5 การเรียกใช้สคริปต์ PHP ร่วมกับ HTML tags

```
1 <html>  
2     <head>  
3         <title>My Homepage</title>  
4     </head>  
5     <body>
```



```
6 <?php  
7     echo "Hello World ! <br>";  
8     echo "I am PHP";  
9 ?>  
10    </body>  
11 </html>
```

การแทรกสคริปต์ PHP ในเอกสาร HTML ในหนังสือเล่มนี้ทั้งเล่มจะใช้รูปแบบ XML style หรือ Default Syntax เนื่องจากสามารถทำงานได้กับทุกเว็บเซิร์ฟเวอร์ ง่ายต่อการอ่านสคริปต์ และสอดคล้อง กับไวยากรณ์ของภาษา XML

2.2 องค์ประกอบพื้นฐานของการเขียนโปรแกรมด้วยภาษา PHP

ภาษา PHP นั้นจะใช้โครงสร้างทางภาษาในรูปแบบเดียวกับภาษา C ดังนั้นแนวทางในการเขียน จึงมีลักษณะคล้ายคลึงกัน ทั้งนี้มีองค์ประกอบพื้นฐานบางส่วนที่ควรศึกษาและทำความเข้าใจ เพื่อเป็น พื้นฐานที่ดีสำหรับการเขียนและพัฒนาเว็บเพจต่อไป ดังนี้

2.2.1 เครื่องหมายสิ้นสุดคำสั่ง

ภาษา PHP จะใช้เครื่องหมาย ; (Semicolon) เป็นตัวแสดงจุดสิ้นสุดในแต่ละคำสั่ง เช่น

\$x = 10;

\$z = "abc";

หลังใส่เครื่องหมาย ; เพื่อสิ้นสุดในแต่ละประโยคคำสั่งแล้ว สามารถนำคำสั่งอื่นมาต่อท้าย ได้ทันที แต่การเขียนสคริปต์ลักษณะนี้จะอ่านสคริปต์ได้ยาก ไม่นิยมเขียนในรูปแบบดังกล่าว ตัวอย่างเช่น

\$x = 10; \$y = x + 10 : \$z = "abc";

2.2.2 คำอธิบาย (Comment)

คำอธิบาย ใช้ในการอธิบายเกี่ยวกับสคริปต์ที่เขียน เพื่อช่วยให้อ่านสคริปต์ได้ง่ายขึ้น แต่โปรแกรมจะไม่นำส่วนที่เป็นคำอธิบายไปประมวลผล สามารถเขียนคำอธิบายได้ 2 รูปแบบ มีรายละเอียด ดังต่อไปนี้

1) Single-Line C++ Syntax หรือ Shell Syntax

เป็นการเขียนคำอธิบายบรรทัดเดียว โดยใช้เครื่องหมาย // หรือ # โปรแกรมจะถือว่า ตั้งแต่สัญลักษณ์เป็นต้นไปที่ระบุเป็นคำอธิบายสคริปต์ และจะไม่นำบรรทัดที่ระบุนั้นไปประมวลผล ดังนี้

ตัวอย่างที่ 2.6 การใช้เครื่องหมาย // หรือ # เพื่อเขียนคำอธิบายแบบบรรทัดเดียว

//Author : Parinya Noidonprai

#All Rights Reserved

2) Multiple-Line C Syntax

เป็นการเขียนคำอธิบายแบบหลายๆ บรรทัด โดยใช้สัญลักษณ์ /* */ โปรแกรมจะถือว่า ตั้งแต่สัญลักษณ์ /* เป็นต้นไปเป็นคำอธิบาย จนกว่าจะเจอสัญลักษณ์ */ จึงจะถือว่าสิ้นสุดคำอธิบาย ตัวอย่าง ดังนี้

ตัวอย่างที่ 2.7 การใช้เครื่องหมาย /* */ เพื่อเขียนคำอธิบายแบบหลายบรรทัด

```
/* Author : Parinya Noidonprai
```

```
All Rights Reserved */
```

2.3 การแสดงผลข้อมูลบนเว็บบราวเซอร์

การแสดงผลข้อมูลบนเว็บบราวเซอร์ คือ การประมวลผลของเว็บเซิร์ฟเวอร์ร่วมกับ Engine ของภาษา PHP และส่งผลลัพธ์กลับเป็นภาษา HTML ไปแสดงผลที่เว็บบราวเซอร์ที่ร้องขอเอกสาร PHP สำหรับในภาษา PHP มี 4 คำสั่งที่สำคัญ คือ print, echo, printf และ sprintf มีรายละเอียด ดังต่อไปนี้

2.3.1 คำสั่ง print

คำสั่ง print เป็นคำสั่งที่ใช้แสดงผลลัพธ์ของข้อมูลบนเว็บบราวเซอร์ โดยมีรูปแบบ (คำสั่ง print ไม่ใช่พังก์ชันแต่เป็นคำสั่งโครงสร้างพื้นฐานของภาษา PHP) มีรูปแบบและตัวอย่างการใช้งานดังนี้

รูปแบบ

```
int print (string $arg)
```

เมื่อ \$arg หมายถึง ตัวแปรข้อมูลนำเข้า (The input data)

หากการทำงานของคำสั่ง print ถูกต้อง จะมีการส่งค่ากลับเป็น "1" เมื่อ หากการทำงานไม่ถูกต้องจะส่งค่ากลับเป็นค่าอื่นๆ

ตัวอย่างที่ 2.8 การใช้งานคำสั่ง print

```
1 <?php
2     print ("I love the summertime.");
3 ?>
```

จากตัวอย่างที่ 2.8 เป็นตัวอย่างการใช้งานคำสั่ง print แบบปกติ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 แสดงผลคำว่า I love the summertime.

บรรทัดที่ 3 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.9 การแสดงค่าของตัวแปรผ่านคำสั่ง print

```
1 <?php
2     $season = "summertime";
3     print "I love the $season.;"
```



4 ?>

จากตัวอย่างที่ 2.9 เป็นตัวอย่างการใช้คำสั่ง print โดยแสดงผลค่าของตัวแปร อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$season มีค่าเท่ากับ summertime

บรรทัดที่ 3 แสดงผลคำว่า I love the summertime. คำว่า summertime มาจากค่าที่เก็บไว้ในตัวแปร \$season

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.10 การใช้ประโยชน์พารามิเตอร์ (Parameter) ที่อยู่คุณลักษณะบรรทัดผ่านคำสั่ง print

```
1 <?php
2     print "I love the
3     summertime.";
4 ?>
```

จากตัวอย่างที่ 2.10 ตัวอย่างการใช้ประโยชน์พารามิเตอร์ ที่อยู่คุณลักษณะบรรทัดผ่านคำสั่ง print ก็สามารถแสดงผลลัพธ์ได้เหมือนกัน เพราะคำสั่ง print ออกแบบมาให้ง่ายและยืดหยุ่น ดังนั้นจะใส่วงเล็บหรือไม่ก็ได้ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 แสดงผลคำว่า I love the

บรรทัดที่ 3 แสดงผลคำว่า summertime. ต่อกับบรรทัดก่อนหน้านี้

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

2.3.2 คำสั่ง echo

คำสั่ง echo เป็นคำสั่งที่ใช้แสดงผลลัพธ์ของข้อมูลบนเว็บбраузอร์ เมื่อกับคำสั่ง print ความแตกต่าง คือ คำสั่ง print จะมีการตรวจสอบข้อผิดพลาดมากกว่าคำสั่ง echo โดยรูปแบบของคำสั่ง echo (คำสั่ง echo ไม่ใช่ฟังก์ชันแต่เป็นคำสั่งโครงสร้างพื้นฐานของภาษาเขียนเดียวกับคำสั่ง print) มีรูปแบบและตัวอย่างการใช้งานดังนี้

รูปแบบ

```
void echo ( string $arg1 [, string $... ] )
```

เมื่อ \$arg1 หมายถึง ตัวแปรข้อมูลนำเข้า (สามารถกำหนดได้หลายตัวแปร)

ตัวอย่างที่ 2.11 การใช้คำสั่ง echo

```
1 <?php
2     $short = "SRU";
3     $long = "Suratthani Rajabhat University.";
4     echo $short, " is ", $long, " We are a great university.;"
```

```
5    ?>
```

ผลลัพธ์

SRU is Suratthani Rajabhat University. We are a great university.

จากตัวอย่างที่ 2.11 ตัวอย่างการใช้คำสั่ง echo อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$short มีค่าเท่ากับ SRU

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$long มีค่าเท่ากับ Suratthani Rajabhat

University.

บรรทัดที่ 4 แสดงผลคำว่า SRU is Suratthani Rajabhat University. We are a great university. เป็นการแสดงผลร่วมกันระหว่างข้อมูลแบบคงที่และข้อมูลแบบไดนามิกผ่านตัวแปร

บรรทัดที่ 5 สิ้นสุดสคริปต์ PHP

คำแนะนำเพิ่มเติม หากต้องการแสดงผลลัพธ์ที่มีการผสมผสานระหว่างข้อมูลแบบคงที่และข้อมูลแบบไดนามิกผ่านตัวแปร ขอให้พิจารณาใช้ printf แทน ซึ่งจะแนะนำในลำดับต่อไป หากเป็นเพียงข้อความแบบคงที่รرمดาทั่วๆ ไป ขอแนะนำให้ใช้คำสั่ง echo หรือ print เนื่องจากง่ายต่อการใช้งาน

2.3.3 คำสั่ง printf

คำสั่ง printf ใช้สำหรับแสดงผลลัพธ์ในรูปแบบข้อความ (Formatted String) เหมาะสำหรับการแสดงผลลัพธ์แบบผสมผสานระหว่างข้อความแบบคงที่และข้อมูลแบบไดนามิกที่เก็บไว้ในตัวแปรหรือหลายๆ ตัวแปร และยังสามารถควบคุมการแสดงผลข้อมูลแบบไดนามิก มีรูปแบบและตัวอย่างการใช้งานดังนี้

รูปแบบ

```
int printf ( string $format [, mixed $args [, mixed $... ] ] )
```

เมื่อ \$format หมายถึง ตัวกำหนดชนิดการแสดงผล (Type Specifies)

\$args หมายถึง ตัวแปรข้อมูลนำเข้า

ตัวอย่างที่ 2.12 การใช้คำสั่ง printf สำหรับแสดงผลลัพธ์ในรูปแบบข้อความ

```
1 <?php
```

```
2     printf ("Bar inventory: %d bottles of tonic water.", 100);
```

```
3 ?>
```

ผลลัพธ์

Bar inventory: 100 bottles of tonic water.

จากตัวอย่างที่ 2.12 แสดงตัวอย่างการใช้คำสั่ง printf สำหรับแสดงผลลัพธ์ในรูปแบบข้อความ อธิบายดังนี้



บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 แสดงผลลัพธ์ในรูปแบบข้อความด้วยคำสั่ง printf ในตัวอย่างมีการกำหนดรูปแบบการแสดงผลด้วย %d คือ ตัวกำหนดชนิดการแสดงผลลัพธ์เป็นค่าของเลขจำนวนเต็ม เมื่อใช้คำสั่ง printf มีค่าพารามิเตอร์ ที่ส่งเข้ามา คือ 100 ค่า 100 จะถูกแทนที่ใน %d ระหว่างข้อความ กรนีที่ ค่าของตัวเลขที่ส่งเข้ามาเป็นค่าทศนิยม ค่าดังกล่าวจะถูกปัดเศษให้เป็นจำนวนเต็มที่ใกล้เคียงที่สุด เช่น ถ้าส่งค่าเป็น 100.2 ผลลัพธ์จะแสดงเป็น 100 สำหรับตัวกำหนดชนิดการแสดงผล มีรายละเอียดดังตารางที่ 2.1

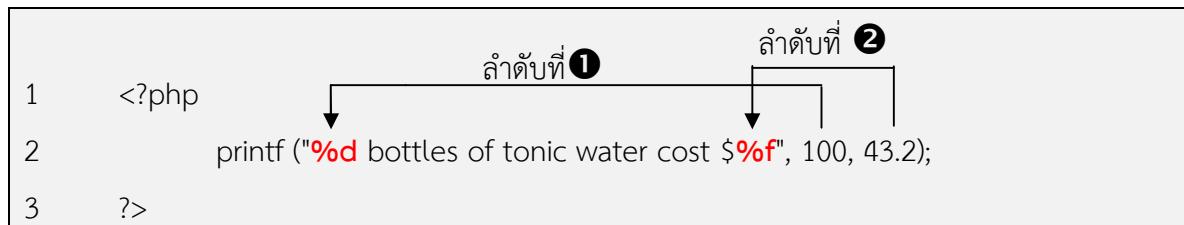
บรรทัดที่ 3 สิ้นสุดสคริปต์ PHP

ตารางที่ 2.1 แสดงตัวกำหนดชนิดการแสดงผลที่ใช้ร่วมกับคำสั่ง printf

ตัวกำหนดชนิด	ความหมาย
%b	เป็นชนิดตัวเลข แสดงผลเป็นเลขไบนาเรี่ย (a binary number)
%c	เป็นชนิดตัวเลข แสดงผลเป็นตัวอักษร ASCII
%d	เป็นชนิดเลขจำนวนเต็มบวกและลบ แสดงผลเป็นตัวเลขฐานสิบ
%f	เป็นชนิดเลขจำนวนจริง แสดงผลเป็นตัวเลขฐานสิบและทศนิยม
%o	เป็นชนิดเลขจำนวนเต็ม แสดงผลเป็นตัวเลขฐานแปด
%s	เป็นชนิดข้อความ แสดงผลเป็นข้อความ
%u	เป็นชนิดเลขจำนวนเต็มบวก แสดงผลเป็นตัวเลขฐานสิบ
%x	เป็นชนิดตัวเลข แสดงผลเป็นเลขฐานสิบหก (อักษรพิมพ์เล็ก)
%X	เป็นชนิดตัวเลข แสดงผลเป็นเลขฐานสิบหก (อักษรพิมพ์ใหญ่)

ดังนั้นหากต้องการที่จะส่งผ่านข้อมูล 2 ค่า สามารถทำได้โดยระบุตัวกำหนดชนิด 2 ตำแหน่ง โดยค่าที่ถูกส่งจะเรียงตามลำดับก่อนหลัง ดังตัวอย่างที่ 2.13

ตัวอย่างที่ 2.13 ตัวอย่างการใช้คำสั่ง printf โดยระบุตัวกำหนดชนิดมากกว่า 1 ชนิด



ผลลัพธ์

100 bottles of tonic water cost \$43.200000

หากต้องการกำหนดจำนวนของการแสดงผลจำนวนเลขทศนิยม สามารถกำหนดได้โดยระบุตัวเลขไว้ก่อนหน้าตัวกำหนดชนิด เช่น หากต้องการกำหนดรูปแบบให้แสดงจำนวนเลขทศนิยมจำนวน 2 หลัก จำเป็นต้องใช้ตัวกำหนดชนิด %f เนื่องจากเป็นเลขจำนวนจริงพร้อมระบุจำนวนเลขที่ต้องการ ดังนี้

ตัวอย่างที่ 2.14 ตัวอย่างการใช้คำสั่ง printf เพื่อการแสดงผลจำนวนเลขทศนิยม

```

1 <?php
2   printf ("%d bottles of tonic water cost $%.2f", 100, 43.2);
3 ?>

```

ผลลัพธ์

```
100 bottles of tonic water cost $43.20
```

ลำดับที่ ①

ลำดับที่ ②

จากตัวอย่างที่ 2.13 และ 2.14 แสดงตัวอย่างการใช้คำสั่ง printf อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 แสดงผลลัพธ์ในรูปแบบข้อความด้วยคำสั่ง printf ในตัวอย่างมีการกำหนดรูปแบบการแสดงผลตำแหน่งที่ 1 ด้วย %d คำสั่งจะส่งค่าข้อมูลมาให้ตำแหน่ง %d ด้วยค่า 100 และตามด้วยข้อความ bottles of tonic water cost \$ และตำแหน่งที่ 2 กำหนดรูปแบบด้วย %f (ทั้ง 2 ตัวอย่างต่างกันที่ ตัวอย่างที่ 2.14 ไม่ได้ระบุจำนวนหลักทศนิยม) คำสั่งจะส่งค่าข้อมูลมาให้ตำแหน่ง %f หรือ %.2f ด้วยค่า 43.2

บรรทัดที่ 3 สิ้นสุดสคริปต์ PHP

2.3.4 คำสั่ง sprintf

คำสั่ง sprintf เป็นคำสั่งที่ทำหน้าที่เหมือนคำสั่ง printf ข้อแตกต่าง คือ sprintf ใช้สำหรับส่งค่ากลับเป็นข้อความหรือเป็นการนำข้อความมาต่อให้เป็นประโยคเดียวกันแล้วกำหนดค่าให้กับตัวแปร เพื่อประโยชน์การทำงานตามเหมาะสม ไม่มีการแสดงผลออกไปยังเว็บбраузอร์ ลักษณะการใช้งานเหมือน printf ดังนั้นจะใช้ ตัวกำหนดชนิด เมมอนกัน ตามตารางที่ 2.1 มีรูปแบบและตัวอย่างการใช้งานดังนี้

รูปแบบ

```
string sprintf ( string $format [, mixed $args [, mixed $... ] ] )
```

ตัวอย่างที่ 2.15 การใช้คำสั่ง sprintf สำหรับส่งค่ากลับเป็นข้อความให้เป็นประโยคเดียวกัน

```

1 <?php
2   $text = sprintf ("%d bottles of tonic water cost $%.2f", 100, 43.2);
3   echo $text;
4 ?>

```

ผลลัพธ์

```
100 bottles of tonic water cost $43.20
```

จากตัวอย่างที่ 2.15 แสดงตัวอย่างการใช้คำสั่ง sprintf อธิบายดังนี้



บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$text มีค่าเท่ากับ 100 bottles of tonic water cost \$ 43.20 โดย 100 ถูกส่งมาให้กับตำแหน่ง %d และ 43.20 ถูกส่งมาให้กับตำแหน่ง %.2f

บรรทัดที่ 3 แสดงผลค่าข้อมูลตัวแปร \$text

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

2.4 ชนิดข้อมูลที่รองรับในภาษา PHP

ในภาษา PHP มีข้อมูลอยู่หลายชนิด ไม่ว่าจะเป็นข้อมูลชนิดบูลิน (Boolean) เลขจำนวนเต็ม (Integer) เลขจำนวนจริง (Float) ข้อความ (String) และอาร์เรย์ (Array) เพื่อทำความเข้าใจและสามารถนำไปใช้ร่วมกับการกำหนดตัวแปรในหัวข้อต่อๆ ไป ดังรายละเอียดต่อไปนี้

2.4.1 ชนิดข้อมูลพื้นฐาน (Scalar Data Types)

ชนิดข้อมูลพื้นฐาน เป็นชนิดข้อมูลที่กำหนดให้ตัวแพรenhนึงๆ เก็บข้อมูลได้เพียงค่าเดียว ชนิดของข้อมูลที่จัดอยู่ในกลุ่มนี้ประกอบด้วย บูลิน เลขจำนวนเต็ม เลขจำนวนจริง และข้อความ มีรายละเอียด ดังนี้

1) ข้อมูลชนิดบูลิน

ข้อมูลชนิดบูลินเป็นชื่อหลังของ George Boole (1815-1864) นักคณิตศาสตร์ที่ได้รับการยกย่องให้เป็นหนึ่งในผู้ก่อตั้งทฤษฎีข้อมูลตัวแปรบูลิน เป็นตัวแทนของความจริงที่สนับสนุนเพียงสองค่าเท่านั้น คือ true (จริง) และ false (เท็จ) หรือสามารถใช้เลขศูนย์แทน false และค่าใดๆ ที่ไม่ใช่ศูนย์แทนค่าของ true ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.16 การกำหนดค่าตัวแปรชนิดบูลิน

```
1 <?php
2     $alive = false;
3     $alive = true;
4     $alive = 1;
5     $alive = 0;
6 ?>
```

จากตัวอย่างที่ 2.16 แสดงตัวอย่างการกำหนดค่าตัวแปรชนิดบูลิน อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$alive มีค่าเป็น false (มีค่าเป็นเท็จ)

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$alive มีค่าเป็น true (มีค่าเป็นจริง)

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$alive มีค่าเป็น 1 (มีค่าเป็นจริง)

บรรทัดที่ 5 กำหนดค่าให้กับตัวแปร \$alive มีค่าเป็น 0 (มีค่าเป็นเท็จ)

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

นอกจากการกำหนดตัวแปรเป็นชนิดบูลีนแล้ว การให้มาของข้อมูลชนิดบูลีนยังได้มาจากการเปรียบเทียบทางตรรกะ เพื่อตัดสินใจดำเนินการร่วมกับโครงสร้างควบคุมอื่นๆ ที่เกี่ยวข้อง เช่น หากผลลัพธ์ออกมาเป็น true ก็จะทำงานต่อ แต่ถ้าออกมาเป็น false ก็จะหยุดทำงาน เป็นต้น

2) ข้อมูลชนิดเลขจำนวนเต็ม

ข้อมูลชนิดเลขจำนวนเต็มเป็นตัวแทนของจำนวนเต็มใดๆ ที่ไม่ประกอบด้วยส่วนที่เป็นเศษส่วน ภาษา PHP สนับสนุนค่าจำนวนเต็มแสดงในเลขฐาน 10 (Decimal) เลขฐาน 8 (Octal) และเลขฐาน 16 (Hexadecimal) ตัวอย่าง ดังนี้

ตัวอย่างที่ 2.17 ตัวอย่างข้อมูลชนิดเลขจำนวนเต็ม

```
42    // เลขฐาน 10
0755 // เลขฐาน 8
0xC4E // เลขฐาน 16
```

ขนาดสูงสุดของข้อมูลชนิดเลขจำนวนเต็มสูงสุดที่รองรับ ขึ้นอยู่กับระบบปฏิบัติการที่ใช้ สำหรับ PHP 5 และเวอร์ชันก่อนหน้านี้ จะเป็น 32 bit จะมีขนาดสูงสุดของเลขเต็มกว่าและเต็มลบเท่ากับ 2^{31} สำหรับ PHP 6 จะสนับสนุนค่าสูงสุดของเลขเต็มกว่าและเต็มลบเท่ากับ 2^{63}

3) ข้อมูลชนิดเลขจำนวนจริง

ข้อมูลชนิดเลขจำนวนจริงหรือเรียกว่าเลขที่มีศนนิยม ช่วยให้สามารถระบุตัวเลขที่มีรายละเอียดที่แท้จริงของเลข เช่น ใช้แทนค่าทางการเงิน น้ำหนัก ระยะทาง และพื้นที่ทั้งหมด เป็นต้น มีตัวอย่าง ดังนี้

ตัวอย่างที่ 2.18 แสดงตัวอย่างข้อมูลชนิดเลขจำนวนจริง

```
4.5678
4.0
8.7e4
1.23E+11
```

4) ข้อมูลชนิดข้อความ

ข้อมูลชนิดข้อความ คือ ข้อมูลที่เป็นอักษร อาจจะเป็นภาษาอังกฤษ ไทย อื่นๆ ก็ได้ แต่เนื่องจากข้อความมีความยาวเท่าไหร่ก็ได้ และไม่จำเป็นต้องเขียนอักษรทุกดัวติดกันไปจนจบข้อความ เหมือนกับการเขียนตัวเลข จึงทำให้โปรแกรมไม่สามารถตัดสินใจได้ว่า ข้อความนั้นเริ่มต้นและสิ้นสุดที่ใด ดังนั้นการกำหนดข้อมูลที่เป็นข้อความ ต้องกำหนดจุดเริ่มต้นและสิ้นสุดเสมอ ด้วยเครื่องหมาย Double Quotes ("...") หรือ Single Quotes ('... ') เช่น "Hello, world" หรือ 'Welcome to Thailand' เป็นต้น แต่อย่างไรก็ตาม การใช้เครื่องหมาย Quotes ทั้งสองแบบจะมีข้อแตกต่างกันในบางกรณี ซึ่งจะได้กล่าวถึงในลำดับต่อไป



ตัวอย่างที่ 2.19 แสดงตัวอย่างข้อมูลชนิดข้อความ

```
"PHP is a great language"  
"Suratthani-Rajabhat-University"  
'*9subway\n'
```

นอกจากนี้ยังสามารถเลือกอักษรบางตัวจากข้อความทั้งหมดได้ โดยใช้อาร์เรย์ ดังนี้

ตัวอย่างที่ 2.20 ตัวอย่างการเลือกอักษรบางตัวจากข้อความทั้งหมด

```
1 <?php  
2     $name = "PARINYA";  
3     $parser = $name[2];  
4     echo $parser;  
5 ?>
```

ผลลัพธ์

R

จากตัวอย่างที่ 2.20 ตัวอย่างการเลือกอักษรบางตัวจากข้อความทั้งหมด อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$name มีค่าเท่ากับ PARINYA

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$parser มีค่าเท่ากับอักษรจากข้อความที่อยู่ในตัวแปร \$name ตำแหน่งอักษรที่ 2 เมื่อ \$name มีรูปแบบการเก็บสายอักษร คือ P[0] A[1] R[2] I[3] N[4] Y[5] และ A[6] ดังนั้น เมื่อกำหนดให้ \$parser=\$name[2] จึงหมายความว่า กำหนดค่าให้ตัวแปร \$parser มีค่าเท่ากับ R นั่นเอง (วิธีการเรียงลำดับเริ่มจากซ้ายมือ เริ่มต้นด้วยลำดับเลข 0)

บรรทัดที่ 4 แสดงค่าตัวแปร \$parser คือ R

บรรทัดที่ 5 สิ้นสุดสคริปต์ PHP

2.4.2 ชนิดข้อมูลแบบชุด (Compound Data Types)

ชนิดข้อมูลแบบชุด เป็นการรวมชุดของรายการที่คล้ายกันไว้ด้วยกัน อาจเป็นตัวแปรชื่อเดียวกันเป็นข้อมูลชนิดเดียวกันเพื่อใช้ในการอ้างอิงถึงกลุ่มข้อมูลชนิดเดียวกัน มีรายละเอียด ดังนี้

1) ชนิดข้อมูลอาร์เรย์ (Array Data Types)

อาร์เรย์จะมีประโยชน์ในการรวมชุดของรายการที่คล้ายกันรวมกันไว้ด้วยกัน เพื่อใช้สำหรับการอ้างอิงถึงข้อมูลของระบบอย่างเฉพาะเจาะจง โครงสร้างข้อมูลที่เรียกว่าอาร์เรย์ ที่ถูกกำหนดโดยอ้างมีรูปแบบเป็นกลุ่มเดียวกัน มีดังนี้เพื่อใช้เข้าถึงหรืออ้างถึงสมาชิกข้อมูลในลำดับต่างๆ การอ้างถึงสมาชิกหนึ่งๆ ในอาร์เรย์จะต้องใช้ดัชนีหรือคีย์ของอาร์เรย์ในการชี้ตำแหน่ง อ้างอิงค่าที่สอดคล้องกัน และแต่ละข้อมูลในสมาชิกสามารถที่จะอ้างอิงด้วยตัวเลขที่เรียบง่ายไปยังตำแหน่งของค่าในชุดหรืออาจมีบางส่วนสัมพันธ์โดยตรงกับค่า

ตัวอย่างที่ 2.21 ตัวอย่างการกำหนดค่าสมาชิกให้กับตัวแปรนิดอาร์เรย์โดยใช้ตัวชี้อาร์เรย์

```
<?php
$countries [0] = "Thailand";
$countries [1] = "Myanmar";
...
$countries [50] = "Cambodia";
?>
```

นอกจากนี้ยังสามารถสร้างตัวชี้ความสัมพันธ์เพื่อเชื่อมโยงข้อมูล เช่น สร้างความสัมพันธ์ของอาร์เรย์ไปยังเมืองหลวงของประเทศต่างๆ แทนที่จะใช้ตัวชี้ที่เป็นตัวเลข โดยใช้ชื่อของประเทศแทนตัวชี้ลักษณะ เช่นนี้จะเรียกว่าคีย์อาร์เรย์สำหรับสร้างความเชื่อมโยงแทน ดังนี้

ตัวอย่างที่ 2.22 แสดงตัวอย่างการกำหนดค่าสมาชิกให้กับตัวแปรนิดอาร์เรย์โดยใช้คีย์อาร์เรย์

```
<?php
$capital ["Thailand"] = "Bangkok";
$capital ["Myanmar"] = "Naypyidaw";
...
$capital ["Cambodia"] = "Phnom Penh";
?>
```

2) ชนิดข้อมูลวัตถุ (Object Data Types)

ชนิดข้อมูลวัตถุ ที่สนับสนุนในภาษา PHP มีแนวคิดมาจากกระบวนการทัศน์ในการเขียนโปรแกรมเชิงวัตถุ แตกต่างจากชนิดข้อมูลอื่นๆ ในภาษา PHP วัตถุจะต้องประกาศอย่างชัดเจน การประกาศลักษณะของวัตถุ และลักษณะการทำงานนี้จะเกิดขึ้นภายในสิ่งที่เรียกว่า "คลาส" มีตัวอย่าง ดังนี้

ตัวอย่างที่ 2.23 แสดงตัวอย่างการเขียนโปรแกรมเชิงวัตถุด้วยภาษา PHP

```
1 <?php
2 class Appliance {
3     private $_power;
4     function setPower ($status) {
5         $this->_power = $status;      // การเรียกใช้เมธอดภายในคลาสเดียวกัน
6     }
7 }
8 $blender = new Appliance;          // การกำหนดให้ $blender ให้สามารถ
9                                // เรียกใช้เมธอดของคลาส Appliance
10 ?>
```



2.5 การแปลงชนิดข้อมูล

การแปลงชนิดข้อมูลหนึ่งไปเป็นชนิดข้อมูลอีก ได้นั้น จำเป็นต้องรู้ถึงชนิดของข้อมูลที่จะแปลงไป การแปลงค่าชนิดข้อมูลสามารถทำได้โดยการวางแผนที่ต้องการแปลงไว้ในด้านหน้าตัวแปรที่จะทำการแปลง สำหรับรายละเอียดชนิดของตัวแปลง แสดงในตารางที่ 2.2 มีรายละเอียด ดังต่อไปนี้

ตารางที่ 2.2 ตัวดำเนินการแปลงชนิดของข้อมูล

Cast Operators	Conversion	ความหมาย
(array)	Array	กำหนดให้เป็นข้อมูลชนิดอาร์เรย์
(bool) or (boolean)	Boolean	กำหนดให้เป็นข้อมูลชนิดบูลีน
(int) or (integer)	Integer	กำหนดให้เป็นข้อมูลชนิดเลขจำนวนเต็ม
(int64) 64-bit	integer	กำหนดให้เป็นข้อมูลชนิดเลขจำนวนเต็ม 64 บิต
(object)	Object	กำหนดให้เป็นข้อมูลชนิดวัตถุ
(real) or (double) or (float)	Float	กำหนดให้เป็นข้อมูลชนิดเลขจำนวนจริง
(string)	String	กำหนดให้เป็นข้อมูลชนิดข้อความ

ตัวอย่างที่ 2.24 การแปลงเลขจำนวนเต็มเป็นเลขจำนวนจริง

```

1 <?php
2     $score=(double)13;
3     printf("Data Type is: %s = %.2f", gettype($score), $score);
4 ?>

```

ผลลัพธ์

Data Type is: double = 13.00

จากตัวอย่างที่ 2.24 การแปลงเลขจำนวนเต็มเป็นเลขจำนวนจริง อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดชนิดให้กับตัวแปร \$score เป็นชนิดเลขจำนวนจริง และกำหนดค่าให้มีค่าเท่ากับค่าที่ได้มาจากการแปลงเลขจำนวนเต็มเป็นเลขจำนวนจริง ด้วย (double)13

บรรทัดที่ 3 ใช้คำสั่ง printf เพื่อแสดงผลข้อมูลจึงได้ผลลัพธ์ตามตัวอย่าง (เมื่อฟังก์ชัน gettype ใช้สำหรับดูชนิดของข้อมูลว่าเป็นชนิดอะไร ผลของฟังก์ชันจะส่งผลกลับเป็นข้อมูลชนิดข้อความ ดังนั้น ภายในคำสั่ง printf จึงต้องกำหนดรูปแบบการแสดงผลด้วย %s)

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

ตัวอย่างการแปลงค่าข้อมูลชนิดเลขจำนวนจริง เป็นเลขจำนวนเต็ม ในกรณีที่ค่าของข้อมูลมีเลขทศนิยม เลขทศนิยมที่อยู่ด้านหลังจะถูกปัดทิ้ง ตัวอย่างดังนี้

ตัวอย่างที่ 2.25 การแปลงค่าข้อมูลนิดเลขจำนวนจริงเป็นเลขจำนวนเต็ม

```

1 <?php
2     $score=(int)13.99;
3     printf("Data Type is: %s = %d", gettype($score), $score);
4 ?>
```

ผลลัพธ์

Data Type is: integer = 13

จากตัวอย่างที่ 2.25 การแปลงค่าข้อมูลนิดเลขจำนวนจริงเป็นเลขจำนวนเต็ม อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดชนิดให้กับตัวแปร \$score เป็นชนิดเลขจำนวนเต็ม และกำหนดค่าให้เท่ากับค่าที่ได้มาจากการแปลงเลขจำนวนจริงเป็นเลขจำนวนเต็ม ด้วย (int)13.99

บรรทัดที่ 3 ใช้คำสั่ง printf เพื่อแสดงผลข้อมูลจึงได้ผลลัพธ์ตามตัวอย่าง โดยเลขทศนิยมที่อยู่ด้านหลังจะถูกปัดทิ้งทั้งหมด

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.26 การแปลงค่าข้อมูลนิดข้อความไปเป็นชนิดเลขจำนวนเต็ม

```

1 <?php
2     $score = (int)"This is a text of 2015";
3     printf("Data Type is: %s = %d", gettype($score), $score);
4 ?>
```

ผลลัพธ์

Data Type is: integer = 0

จากตัวอย่างที่ 2.26 การแปลงค่าข้อมูลนิดข้อความไปเป็นชนิดเลขจำนวนเต็ม อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดชนิดให้กับตัวแปร \$score เป็นชนิดเลขจำนวนเต็ม และกำหนดค่าให้มีเท่ากับค่าที่ได้มาจากการแปลงข้อความเป็นเลขจำนวนเต็ม ด้วย (int)"This is a text of 2015"

บรรทัดที่ 3 ใช้คำสั่ง printf เพื่อแสดงผลข้อมูลจึงได้ผลลัพธ์ตามตัวอย่าง จะเห็นได้ว่าข้อมูลนิดข้อความเมื่อถูกแปลงข้อมูลเป็นชนิดตัวเลข ไม่ว่าจะเป็นเลขจำนวนเต็มหรือเลขจำนวนจริง ค่าของตัวแปรจะมีค่าเป็น 0 ดังนั้นควรระวัง เมื่อมีการแปลงข้อมูลลักษณะเช่นนี้

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP



ตัวอย่างที่ 2.27 การแปลงค่าข้อมูลปกติเป็นข้อมูลชนิดอาร์เรย์

```
1 <?php  
2     $name = "Suratthani Rajabhat University";  
3     $arrayname = (array)$name;  
4     printf("Data Type is: %s = %s", gettype($arrayname), $arrayname[0]);  
5 >
```

ผลลัพธ์

```
Data Type is: array = Suratthani Rajabhat University
```

จากตัวอย่างที่ 2.27 การแปลงค่าข้อมูลปกติเป็นข้อมูลชนิดอาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$name มีค่าเป็นข้อความคำว่า "Suratthani Rajabhat University"

บรรทัดที่ 3 กำหนดชนิดให้กับตัวแปร \$arrayname เป็นชนิดอาร์เรย์ และกำหนดค่าให้กับตัวนี้ อาร์เรย์ลำดับที่ 0 มีค่าเท่ากับค่าที่อยู่ในตัวแปร \$name

บรรทัดที่ 4 ใช้คำสั่ง printf เพื่อแสดงผลข้อมูลจึงได้ผลลัพธ์ตามตัวอย่าง

บรรทัดที่ 5 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.28 การแปลงค่าข้อมูลปกติเป็นข้อมูลชนิดวัตถุ

```
1 <?php  
2     $name = "Suratthani Rajabhat University";  
3     $obj = (object) $name;  
4     printf("Data Type is: %s = %s", gettype($obj), $obj -> scalar);  
5 >
```

ผลลัพธ์

```
Data Type is: object = Suratthani Rajabhat University
```

จากตัวอย่างที่ 2.28 การแปลงค่าข้อมูลปกติเป็นข้อมูลชนิดวัตถุ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$name มีค่าเป็นข้อความคำว่า "Suratthani Rajabhat University"

บรรทัดที่ 3 กำหนดชนิดให้กับตัวแปร \$obj เป็นชนิดวัตถุ และกำหนดค่าให้เท่ากับค่าที่อยู่ในตัว แปร \$name

บรรทัดที่ 4 ใช้คำสั่ง printf แสดงค่าของตัวแปรวัตถุด้วยรูปแบบ \$obj -> scalar เพื่อแสดงผล ค่าข้อมูลจึงได้ผลลัพธ์ตามตัวอย่าง

บรรทัดที่ 5 สิ้นสุดสคริปต์ PHP

2.6 การปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ

ภาษา PHP ลูกออกแบบมาให้มีความยืดหยุ่นสูง สามารถปรับเปลี่ยนค่าตัวแปรได้หลากหลายและทันท่วงทีแบบอัตโนมัติ หรือสามารถปรับเปลี่ยนชนิดของตัวแปรแบบอัตโนมัติ โดยระบบจะทำการพิจารณาความเหมาะสมกับสถานการณ์ที่มีการอ้างอิงตัวแปร มีตัวอย่างดังนี้

ตัวอย่างที่ 2.29 การปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ แบบที่ 1

```

1 <?php
2     $total=5;
3     $count="15";
4     printf("Data Type of total variable is : %s<br>", gettype($total));
5     printf("Data Type of count variable is : %s<br>", gettype($count));
6     printf("%d + %d ", $total, $count);
7     $total=$total+$count;
8     printf(" = %d", $total);
9 ?>
```

ผลลัพธ์

Data Type of total variable is : integer

Data Type of count variable is : string

5 + 15 = 20

จากตัวอย่างที่ 2.29 การปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ แบบที่ 1 อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$total มีค่าเท่ากับ 5 (เลขจำนวนเต็ม)

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$count มีค่าเท่ากับ 15 (ข้อความ)

บรรทัดที่ 4 และ 5 แสดงผลชนิดของตัวแปรแต่ละตัว โดยใช้ฟังก์ชัน gettype()

บรรทัดที่ 6 แสดงค่าตัวแปร \$total + \$count ผลลัพธ์ที่ได้ คือ 5 + 15

บรรทัดที่ 7 กำหนดค่าใหม่ให้กับตัวแปร \$total มีค่าเท่ากับผลรวมของตัวแปร \$total+\$count คือ มีค่าเท่ากับ 20 (ลักษณะดังกล่าวจึงถือได้ว่าเป็นการปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ ภาษาอื่นๆ จะไม่สามารถนำเอาเลขจำนวนเต็มมาบวกกับข้อความได้)

บรรทัดที่ 8 แสดงเครื่องหมายเท่ากับและค่าตัวแปร \$total คือ =20

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.30 การปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ แบบที่ 2

```

1 <?php
2     $total = "45 fire engines";
```



```
3     $incoming = 10;  
4     $total = $incoming + $total;  
5     printf("%d ", $total);  
6 ?>
```

ผลลัพธ์

55

จากตัวอย่างที่ 2.30 การปรับเปลี่ยนชนิดข้อมูลแบบอัตโนมัติ แบบที่ 2 อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$total มีค่าเป็นชนิดข้อความ และกำหนดให้มีค่าเท่ากับ "45 fire engines"

บรรทัดที่ 3 กำหนดให้ตัวแปร \$incoming เป็นชนิดเลขจำนวนเต็ม และกำหนดให้มีค่าเท่ากับ 10

บรรทัดที่ 4 กำหนดให้ตัวแปร \$total เดิมเป็นชนิดข้อความ เมื่อกำหนดให้มีค่าเท่ากับผลบวกของ ตัวมันเองบวกด้วยค่าในตัวแปร \$incoming โดยระบบจะตรวจสอบว่าข้อความที่อยู่ในตัวแปร \$total จึงต้น ด้วยตัวเลขหรือไม่ หากมีระบบจะเลือกค่าตัวเลขในช่วงต้นแล้วตัดข้อมูลที่เป็นข้อความออกทั้งหมดแล้วนำไป บวกกับค่าในตัวแปร \$incoming โดยสรุปกล่าวคือ จากสมการ \$total = \$incoming + \$total จะเป็นการ กำหนดให้ตัวแปร \$total มีค่าเท่ากับ 45+10 ผลลัพธ์ที่ได้จึงมีค่าเท่ากับ 55

บรรทัดที่ 5 แสดงผลค่าตัวแปร \$total

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

2.7 พิมพ์ข้อความที่เกี่ยวข้องกับชนิดข้อมูล

พิมพ์ข้อความที่เกี่ยวข้องกับชนิดข้อมูล (Type-Related Functions) มีรายละเอียด ดังนี้

2.7.1 พิมพ์ข้อความที่เกี่ยวกับชนิดของข้อมูล (Retrieving Types Functions)

พิมพ์ข้อความ `gettype()` เป็นพิมพ์ข้อความที่ใช้สำหรับเรียกคืนชนิดของข้อมูล โดยการส่งผลลัพธ์เป็น ชนิดของข้อมูลในตัวแปร ค่าที่ส่งกลับมีทั้งหมด 9 ค่า ประกอบด้วย boolean, integer, double หรือ float, string, array, object, resource, NULL และ unknown type รูปแบบการใช้งาน ดังนี้

รูปแบบ

```
string gettype ( mixed $var )
```

เมื่อ \$var หมายถึง ตัวแปรที่ต้องการตรวจสอบชนิด

ตัวอย่างที่ 2.31 การใช้พิมพ์ข้อความ `gettype()` เรียกคืนชนิดของข้อมูล

```
1 <?php  
2     $name="Suratthani Rajabhat University";  
3     printf("Data Type is: %s ", gettype($name));  
4 ?>
```

ผลลัพธ์

Data Type is: string

จากตัวอย่างที่ 2.31 การใช้ฟังก์ชัน `gettype()` เรียกดูชนิดของข้อมูล อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร `$name` มีค่าเท่ากับ Suratthani Rajabhat University

บรรทัดที่ 3 แสดงผลชนิดของตัวแปร `$name` ด้วยใช้ฟังก์ชัน `gettype()` สำหรับเรียกดูชนิดของข้อมูลที่อยู่ในตัวแปร

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

2.7.2 ฟังก์ชันแปลงชนิดข้อมูล (Converting Types Functions)

ฟังก์ชัน `settype()` เป็นฟังก์ชันสำหรับแปลงตัวแปรตามที่ระบุให้เป็นชนิดอื่นๆ ตามต้องการ ชนิดของข้อมูลที่สามารถแปลงได้นั้น มี 7 ชนิด ประกอบด้วย array, boolean, float, integer, null, object และ string นอกจากนี้ถ้าหากการแปลงชนิดข้อมูลประสบความสำเร็จ ฟังก์ชัน `settype()` จะมีการส่งค่ากลับผลลัพธ์จะมีค่าเป็น TRUE หากไม่สำเร็จจะส่งค่ากลับเป็น FALSE รูปแบบการใช้งาน ดังนี้

รูปแบบ

`bool settype (mixed &$var , string $type)`

เมื่อ `$var` หมายถึง ตัวแปรที่ต้องการแปลงชนิด

`$type` หมายถึง ชนิดที่ต้องการแปลง มีรายละเอียดดังนี้

"boolean" คือ บูลีน

"integer" คือ เลขจำนวนเต็ม

"float" คือ เลขจำนวนจริง

"string" คือ ข้อความ

"array" คือ อาร์เรย์

"object" คือ օբเจ็กต์

"null" คือ ค่าว่างหรือไม่กำหนดชนิดใดๆ

ตัวอย่างที่ 2.32 การใช้ฟังก์ชัน `settype()` เพื่อแปลงหรือกำหนดชนิดข้อมูล

```

1 <?php
2     $foo = "5bar";           // เป็นชนิด string
3     $bar = true;            // เป็นชนิด boolean
4     settype ($foo, "integer"); // ผลลัพธ์ $foo จะมีค่าเท่ากับ 5 และเป็นชนิด integer
5     settype ($bar, "string"); // ผลลัพธ์ $bar จะมีค่าเท่ากับ "1" และเป็นชนิด string
6 ?>

```

จากตัวอย่างที่ 2.32 การใช้ฟังก์ชัน `settype()` เพื่อแปลงหรือกำหนดชนิดข้อมูล อธิบายดังนี้



บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$foo มีค่าเท่ากับ "5bar" โดยปริยายข้อมูลลักษณะเช่นนี้คือข้อมูลชนิดข้อความ

บรรทัดที่ 3 กำหนดให้ตัวแปร \$bar มีค่าเท่ากับ true โดยปริยายข้อมูลลักษณะเช่นนี้คือข้อมูลชนิดตรรกะหรือบูลีน

บรรทัดที่ 4 ใช้ฟังก์ชัน settype เพื่อกำหนดให้ตัวแปร \$foo แปลงเป็นข้อมูลชนิดเลขจำนวนเต็ม ฟังก์ชันจะพิจารณาว่าค่าภายในขึ้นต้นด้วยตัวเลขหรือไม่ เช่น จากตัวอย่างค่าภายในประกอบด้วยเลข 5 แล้วตามด้วยข้อความ ฟังก์ชันจะตัดเลือกเฉพาะตัวเลขแล้วลบ ข้อความทั้งไป ดังนั้นค่าใหม่ที่กำหนดให้ตัวแปร \$foo จะเป็นชนิดเลขจำนวนเต็มและมีค่าเท่ากับ 5

บรรทัดที่ 5 ใช้ฟังก์ชัน settype เพื่อกำหนดให้ตัวแปร \$bar เป็นชนิดข้อความ จากเดิมเป็นชนิดตรรกะหรือบูลีน ดังนั้นเมื่อเปลี่ยนแปลงชนิดใหม่ค่าภายในตัวแปรจะเปลี่ยนเป็น 1 หากก่อนหน้าค่าภายในตัวแปรเป็นชนิด false เมื่อปรับเปลี่ยนเป็นชนิดข้อความจะกลับเป็น 0

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

2.7.3 ฟังก์ชันที่ใช้สำหรับตรวจสอบชนิดของข้อมูล (Type Identifier Functions)

ฟังก์ชันที่ใช้สำหรับตรวจสอบชนิดของข้อมูล ประกอบด้วย

ตารางที่ 2.3 ฟังก์ชันที่ใช้สำหรับตรวจสอบชนิดของข้อมูล

ชื่อฟังก์ชัน	หน้าที่
is_array ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดอาร์เรย์จริงหรือไม่
is_bool ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดตรรกะหรือบูลีนจริงหรือไม่
is_float () หรือ is_real	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดเลขจำนวนจริงจริงหรือไม่
is_integer () หรือ is_int ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดเลขจำนวนเต็มจริงหรือไม่
is_null ()	ตรวจสอบชนิดของข้อมูลว่าเป็นค่าว่างจริงหรือไม่
is_numeric ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดตัวเลขจริงหรือไม่
is_object ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดวัตถุจริงหรือไม่
is_resource ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดติดต่อใช้ทรัพยากรของระบบจริงหรือไม่
is_scalar ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดข้อมูลพื้นฐานที่ภาษา PHP รองรับจริงหรือไม่ ชนิดข้อมูลพื้นฐานประกอบด้วย Boolean, integer, float และ string สำหรับ array, object, null และ resource ไม่ถือว่าเป็นชนิดข้อมูลพื้นฐาน
is_string ()	ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดข้อความจริงหรือไม่

จากตารางที่ 2.3 ฟังก์ชันที่ใช้สำหรับตรวจสอบชนิดของข้อมูล ทุกฟังก์ชันจะมีรูปแบบการใช้งานเหมือนกัน ประกอบด้วย ชื่อฟังก์ชันตรวจสอบ และในพารามิเตอร์ให้ใส่ตัวแปรที่ต้องการตรวจสอบ หลังจากที่ฟังก์ชันทำการตรวจสอบแล้ว จะจะส่งค่ากลับเป็น true หากค่าของตัวแปรทั้งหมดเป็น true หากไม่ตรงกับชนิดข้อมูลที่ตรวจสอบจะส่งค่ากลับเป็น false การใช้ฟังก์ชันตรวจสอบชนิดข้อมูล นิยมใช้ร่วมกับโครงสร้างควบคุมเพื่อตรวจสอบเงื่อนไข เช่น โครงสร้างเงื่อนไข if เป็นต้น ดังนี้

รูปแบบ

```
boolean is_name (mixed var) // เมื่อกำหนดให้ is_name แทนชื่อฟังก์ชันตรวจสอบ
```

เมื่อ \$var หมายถึง ตัวแปรที่ต้องการตรวจสอบชนิด

ตัวอย่างที่ 2.33 การใช้ฟังก์ชัน is_bool () ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดตรรกะหรือบูลีน

```
1 <?php
2     $a = false;
3     if (is_bool($a) === true) {
4         echo "Yes, this is a boolean";
5     } else {
6         echo "No, this is a ",gettype($a);
7     }
8 ?>
```

ผลลัพธ์

```
Yes, this is a boolean
```

จากตัวอย่างที่ 2.33 การใช้ฟังก์ชัน is_bool () อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$a มีค่าเท่ากับ false

บรรทัดที่ 3 ใช้ฟังก์ชัน is_bool () ร่วมกับโครงสร้างเงื่อนไข if โดยฟังก์ชัน is_bool () ใช้สำหรับตรวจสอบว่าตัวแปรหรือค่าของข้อมูลที่ระบุเป็นชนิดตรรกะหรือบูลีนจริงหรือไม่ หากผลของการตรวจสอบเป็นจริงตามเงื่อนไข จะไปทำงานต่อบรรทัดที่ 4 เมื่อทำงานคำสั่งโครงสร้างเงื่อนไข if เสร็จเรียบร้อย ทำงานต่อหลังบรรทัดที่ 7

บรรทัดที่ 4 แสดงคำว่า Yes, this is a boolean เมื่อทำงานตามคำสั่งเสร็จจะออกจากโครงสร้างเงื่อนไข if ทำงานต่อหลังบรรทัดที่ 7

บรรทัดที่ 5 ในกรณีที่เป็นเท็จจะเข้าสู่เงื่อนไขของโครงสร้างเงื่อนไข else

บรรทัดที่ 6 แสดงคำว่า No, this is a แล้วตามด้วยฟังก์ชัน gettype (\$a) เพื่อแสดงชนิดของตัวแปร เมื่อทำงานเสร็จจะไปบรรทัดที่ 7 เพื่อออกจากโครงสร้างเงื่อนไข if...else



บรรทัดที่ 7 สื้นสุดโครงสร้างเงื่อนไข if...else

บรรทัดที่ 8 สื้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.34 การใช้ฟังก์ชัน is_int() ตรวจสอบชนิดของข้อมูลว่าเป็นชนิดเลขจำนวนเต็ม

```
1 <?php
2     if (is_int(23)) {
3         echo "is an integer";
4     } else {
5         echo "is not an integer";
6     }
7 ?>
```

ผลลัพธ์

```
is an integer
```

จากตัวอย่างที่ 2.34 การใช้ฟังก์ชัน is_int() อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 ใช้ฟังก์ชัน is_int() ร่วมกับโครงสร้างเงื่อนไข if โดยฟังก์ชัน is_int ใช้สำหรับตรวจสอบว่าตัวแปรหรือค่าของข้อมูลที่ระบุเป็นเลขจำนวนเต็มจริงหรือไม่ หากผลของการตรวจสอบเป็นจริงตามเงื่อนไข จะไปทำงานต่อบรรทัดที่ 3 หากเป็นเท็จจะไปทำงานต่อบรรทัดที่ 4

บรรทัดที่ 3 แสดงคำว่า is an integer (เป็นเลขจำนวนเต็ม) เมื่อทำงานตามคำสั่งเสร็จจะออกจากโครงสร้างเงื่อนไข if...else ทำงานต่อหลังบรรทัดที่ 6

บรรทัดที่ 4 ในกรณีที่เป็นเท็จจะเข้าสู่โครงสร้างเงื่อนไข else

บรรทัดที่ 5 แสดงผลคำว่า is not an integer (ไม่ใช่เลขจำนวนเต็ม)

บรรทัดที่ 6 สื้นสุดโครงสร้างเงื่อนไข if...else

บรรทัดที่ 7 สื้นสุดสคริปต์ PHP

ตัวอย่างที่ 2.35 ตัวอย่างการใช้ฟังก์ชัน is_resource()

```
1 <?php
2     $db_link = mysql_connect("localhost","mysql_user","mysql_pass");
3     if (!is_resource($db_link)) {
4         die("Can't connect" . mysql_error());
5     }
6 ?>
```

จากตัวอย่างที่ 2.35 การใช้ฟังก์ชัน `is_resource()` เพื่อตรวจสอบการเชื่อมต่อกับฐานข้อมูล MySQL ว่าสามารถเชื่อมต่อได้หรือไม่ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร `$db_link` เป็นชนิด `resource` เพื่อเชื่อมต่อ กับฐานข้อมูล MySQL

บรรทัดที่ 3 ตรวจสอบเงื่อนไข โดยใช้โครงสร้างเงื่อนไข `if` ร่วมกับฟังก์ชัน `is_resource($db_link)` หากไม่สามารถเชื่อมต่อได้ แสดงว่า ตัวแปร `$db_link` ปิดการเชื่อมต่อ กับฐานข้อมูล อาจมีสาเหตุมาจากการไม่สามารถเชื่อมกับฐานข้อมูล MySQL อันเนื่องมาจากชื่อโฮส หรือชื่อผู้ใช้งาน หรือรหัสผ่าน สำหรับการเชื่อมต่อฐานข้อมูล MySQL ไม่ถูกต้อง ดังนั้นเมื่อใช้ฟังก์ชัน `is_resource` เพื่อตรวจสอบตัวแปร `$db_link` หากส่งผลค่ากลับเป็น `false` แสดงว่า เชื่อมต่อไม่ได้ หากเป็น `true` แสดงว่า เชื่อมต่อได้ แต่ เมื่อใช้งานร่วมกับโครงสร้างเงื่อนไข `if` เงื่อนไขจะต้องเป็นจริงเท่านั้นถึงจะเข้าไปทำคำสั่งภายใต้เงื่อนไข จึง เป็นเหตุผลที่ต้องการใช้เครื่องหมาย `!` (`not`) เพื่อทำเป็นเงื่อนไขปฏิเสธหรือตรงกันข้าม เช่น `!true` คือ `false` และ `!false` คือ `true` โดยสรุป หากเงื่อนไขของการตรวจสอบด้วยโครงสร้างเงื่อนไข `if` ได้ผลเป็น `!false` จะทำให้เงื่อนไขเป็นจริง จะไปทำงานต่อบรรทัดที่ 4 หากเป็นเท็จจากเงื่อนไข คือ สามารถเชื่อมต่อ ฐานข้อมูลได้จะออกจากโครงสร้างเงื่อนไข `if` คือ ทำงานต่อหลังจากบรรทัดที่ 5 (หากมีคำสั่งอื่นๆ เพิ่มเติม)

บรรทัดที่ 4 สั่งให้โปรแกรมหยุดทำงาน โดยใช้ฟังก์ชัน `die()` และแจ้งข้อความสาเหตุที่ไม่ สามารถเชื่อมต่อฐานข้อมูลได้

บรรทัดที่ 5 สิ้นสุดโครงสร้างเงื่อนไข `if`

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

2.8 คำส่วนของภาษา PHP

คำส่วนของภาษา PHP (Reserved Words in PHP) หมายถึง คำที่จะใช้เป็นคำสั่งเฉพาะของ ภาษา PHP ดังนั้นต้องไม่นำคำเหล่านี้ไปตั้งหรือกำหนดเป็นชื่อให้กับตัวแปร หรือฟังก์ชัน มิฉะนั้นจะทำให้ เกิดข้อผิดพลาดขึ้นได้มีรายละเอียด ดังตารางที่ 2.4

ตารางที่ 2.4 คำส่วนของภาษา PHP

<code>and</code>	<code>or</code>	<code>xor</code>	<code>_FILE_</code>	<code>exception (5)</code>
<code>_LINE_</code>	<code>array()</code>	<code>as</code>	<code>break</code>	<code>case</code>
<code>class</code>	<code>const</code>	<code>continue</code>	<code>declare</code>	<code>default</code>
<code>die()</code>	<code>do</code>	<code>echo</code>	<code>else</code>	<code>elseif</code>
<code>empty()</code>	<code>enddeclare</code>	<code>endfor</code>	<code>endforeach</code>	<code>endif</code>
<code>endswitch</code>	<code>endwhile</code>	<code>eval()</code>	<code>exit()</code>	<code>extends</code>

ตารางที่ 2.4 (ต่อ)

For	foreach	function	global	if
include ()	include_once ()	isset ()	list ()	new
print	require ()	require_once ()	return ()	static
switch	unset ()	use	var	while
FUNCTION	_CLASS_	_METHOD_	nal ()	php_user_lter (5)
interface (5)	implements (5)	extends	public (5)	private (5)
protected (5)	abstract (5)	clone (5)	try (5)	catch (5)
throw (5)	cfunction (4 only)	old_function (4 only)		

สรุป

เครื่องมือที่ใช้เขียนสคริปต์ PHP สามารถเลือกใช้ได้หลายโปรแกรม เช่น Notepad, Notepad++, EditPlus, vi และ Adobe Dreamweaver เป็นต้น รูปแบบการเขียนสคริปต์ PHP นั้น จะใช้เทคนิควิธีการแทรกประโยคคำสั่งร่วมกับ HTML tags จุดที่สำคัญ คือ จุดเริ่มต้นและจุดสิ้นสุดของ สคริปต์ PHP ซึ่งหลักๆ การบอกจุดเริ่มต้นและจุดสิ้นสุดของสคริปต์ PHP มี 4 รูปแบบ ประกอบด้วย

1) XML Style หรือ Default Syntax 2) SGML Style หรือ Short Tags 3) Script Style และ 4) ASP Style และประโยคคำสั่งที่เขียนในแต่ละบรรทัดเรียกว่าสคริปต์คำสั่ง แต่ละสคริปต์คำสั่งจะต้องใส่ เครื่องหมาย ; (Semicolon) เพื่อบอกจุดสิ้นสุดในแต่ละสคริปต์คำสั่ง ชนิดของข้อมูลที่รองรับในภาษา PHP มีหลายชนิด ดังนั้นก่อนการเขียนสคริปต์ PHP ควรศึกษา ทำความเข้าใจ และเลือกใช้ชนิดของข้อมูล ให้เหมาะสมตามลักษณะงาน เพื่อความถูกต้องในการประมวลผลและแสดงผลลัพธ์ตามต้องการ

คำถามท้ายบท

1. จงอธิบายเหตุผล และรูปแบบการเขียนสคริปต์ PHP แทรกใน HTML tags มีรูปแบบ อะไรบ้าง พร้อมยกตัวอย่างประกอบในแต่ละรูปแบบ
2. จงอธิบายคำสั่งที่ใช้แสดงผลข้อมูลบนเว็บบราวเซอร์ที่สำคัญของภาษา PHP รูปแบบการใช้งาน และยกตัวอย่างประกอบ
3. จงอธิบายถึงตัวกำหนดชนิดการแสดงผลที่ใช้งานร่วมกับคำสั่ง printf พร้อมยกตัวอย่าง ประกอบ
4. จงอธิบายถึงชนิดข้อมูลที่รองรับในภาษา PHP รายละเอียดของแต่ละชนิดพoSังเขป พร้อม ยกตัวอย่างประกอบ
5. จงอธิบายถึงฟังก์ชันเรียกดูชนิดของข้อมูล ฟังก์ชันแปลงชนิดข้อมูล และฟังก์ชันตรวจสอบ ชนิดของข้อมูล พร้อมยกตัวอย่างประกอบ

บทที่ 3

ตัวแปร ตัวดำเนินการและนิพจน์

ในการศึกษาการเขียนโปรแกรมทุกภาษา สิ่งที่ควรเรียนรู้เป็นลำดับแรกๆ คือ การกำหนดตัวแปร สำหรับเก็บข้อมูลบางอย่างเอาไว้ก่อนจะนำไปใช้งานอื่นๆ ต่อไป และขั้นตอนถัดไปก็คือ ทำความรู้จักการใช้ตัวดำเนินการแบบต่างๆ เพื่อใช้ในการประมวลผลข้อมูลให้ได้ผลลัพธ์ตามต้องการ

3.1 ตัวแปร

ตัวแปร (Variable) ใช้ในการเก็บพักข้อมูลบางอย่างก่อนที่จะนำข้อมูลนั้นไปใช้งานอื่นๆ ต่อไป สำหรับตัวแปรในภาษา PHP ไม่จำเป็นต้องได้รับการประกาศอย่างชัดเจนเหมือนภาษาอื่นๆ แต่ก็ไม่เป็นสิ่งที่ดีนักหากขาดทักษะการเขียนโปรแกรมที่ดี ก็อาจจะทำให้เกิดความสับสนในการใช้งานได้ ในภาษา PHP มีข้อกำหนดสำหรับประกาศใช้ตัวแปรที่สำคัญ ดังต่อไปนี้

3.1.1 ตัวแปรในภาษา PHP ไม่จำเป็นต้องระบุชนิดของข้อมูล เนื่องจากตัวแปรแต่ละตัวสามารถเก็บข้อมูลชนิดใดก็ได้

3.1.2 ตัวแปรในภาษา PHP จะต้องขึ้นด้วยสัญลักษณ์เครื่องหมาย \$ (Dollar Sign) และตามด้วยชื่อของตัวแปรที่ต้องการใช้งาน เช่น \$name, \$value, \$a, \$x เป็นต้น

3.1.3 การตั้งชื่อตัวแปรต้องขึ้นต้นด้วยอักษร a-z หรือ A-Z หรือเครื่องหมาย _ (Underscore) เท่านั้น ห้ามขึ้นต้นด้วยตัวเลข 0-9 หรืออักษรระเอื่ី นอกเหนือจากนี้ ตัวอย่างการกำหนดตัวแปรที่ถูกต้อง เช่น \$name, \$_price, \$value1, \$num2string เป็นต้น

3.1.4 การเขียนตัวแปรด้วยลักษณะ ตัวพิมพ์ที่แตกต่างกัน ถือว่าเป็นตัวแปรคละตัว เช่น \$abc, \$ABC เป็นต้น จะถือว่าไม่ใช่ตัวแปรเดียวกัน

3.1.5 สามารถนำอักษรภาษาอื่นๆ มาตั้งเป็นชื่อตัวแปรได้ เช่น \$ชื่อ, \$จำนวน1, \$ผลลัพธ์ เป็นต้น แต่โดยทั่วไปแล้ว นิยมตั้งชื่อตัวแปรเป็นภาษาอังกฤษมากกว่า

3.2 การกำหนดค่าตัวแปร

เมื่อได้ประกาศตัวแปรก็สามารถเริ่มต้นใช้งานได้ทันที แต่ก่อนที่จะนำตัวแปรไปใช้งานได้ต้อง ตัวแปรจะต้องเก็บข้อมูลบางอย่างเอาไว้ หรือเรียกว่าการกำหนดค่าตัวแปร การกำหนดค่าตัวแปรแบบง่ายๆ คือ การคัดลอกค่าหรือโอนถ่ายค่าของนิพจน์หรือสมการต่างๆ โดยมีสัญลักษณ์ที่ใช้กำหนดค่า คือ เครื่องหมาย = (เท่ากับ) การกำหนดค่าตัวแปรหลักๆ จะมีอยู่ด้วยกัน 4 วิธี คือ 1) กำหนดค่าตัวแปรปกติ (Value assignment) 2) กำหนดค่าตัวแปรโดยการอ้างอิง (Reference assignment) 3) กำหนดค่าตัว

แปรผัณฑ์ตัวแปร (Variable variables assignment) และ 4) กำหนดตัวแปรค่าคงที่ (Constants variables) ดังนี้

3.2.1 การกำหนดค่าตัวแปรปกติ

การกำหนดค่าตัวแปรรูปแบบนี้ เป็นการกำหนดค่าตัวแปรแบบง่ายที่สุดและนิยมใช้งานโดยทั่วไป สามารถกำหนดได้หลายลักษณะขึ้นอยู่กับชนิดข้อมูล ดังนี้

- 1) การกำหนดค่าตัวแปรชนิดตัวเลข ตัวอย่างดังนี้

ตัวอย่างที่ 3.1 การกำหนดค่าตัวแปรชนิดตัวเลข

```
1      <?php  
2          $x = 123;  
3          $y = 4.56;  
4          $z = -789;  
5      ?>
```

จากตัวอย่างที่ 3.1 เมื่อการกำหนดค่าตัวแปรนิดตัวเลข อธิบายดังนี้

บริหัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร Rx มีค่าเท่ากับ 123

บริบทที่ 3 กำหนดค่าให้กับตัวแปร R_V มีค่าเท่ากับ 4.56

ฯ នរោត្តិទី ៤ ការណែនគោលការណ៍ និងការពិភាក្សា នៃ មីគារ់ខោក៉ា -789

บรรทัดที่ 5 สิ้นสุดสคริปต์ PHP

- 2) การกำหนดค่าตัวแปรชนิดข้อความ สามารถกำหนดได้ตามรูปแบบการเขียนข้อความนั่นคือกำหนดข้อความไว้ภายในเครื่องหมาย Double Quotes ("...") หรือ Single Quotes ('...') เท่านั้น
ตัวอย่างดังนี้

ตัวอย่างที่ 3.2 การกำหนดค่าตัวแปรชนิดข้อความ

```
1 <?php  
2 $name = "Parinya";  
3 $country = 'Thailand';  
4 $phone = '0123456789';  
5 ?>
```

จากตัวอย่างที่ 3.2 เมื่อการกำหนดค่าตัวแปรนิดี้ความ อริบายดั้งนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$name มีค่าเท่ากับ "Parinya"

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$country มีค่าเท่ากับ 'Thailand'

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$phone มีค่าเท่ากับ '0123456789' เลขที่ใช้ในอยู่ภายในเครื่องหมาย Double Quotes หรือ Single Quotes จะถือว่าเป็นชนิดข้อความ แต่สามารถนำไปใช้คำนวณได้

บรรทัดที่ 5 สิ้นสุดสคริปต์ PHP

3) การกำหนดค่าตัวแปรชนิดบูลีน สามารถกำหนดค่าเป็น true หรือ false อย่างใดอย่างหนึ่ง ดังนี้

ตัวอย่างที่ 3.3 การกำหนดค่าตัวแปรชนิดบูลีน

```

1 <?php
2     $first_time = true ;
3     $is_valid = false ;
4 ?>

```

จากตัวอย่างที่ 3.3 เป็นการกำหนดค่าตัวแปรชนิดบูลีน อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าตัวแปร \$first_time มีค่าเท่ากับ true

บรรทัดที่ 3 กำหนดค่าตัวแปร \$is_valid มีค่าเท่ากับ false

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

การใช้งานตัวแปรใน PHP มีความยืดหยุ่น ไม่จำกัดชนิดของตัวแปรเป็นชนิดใดชนิดหนึ่ง และยังสามารถสลับปรับเปลี่ยนได้ในทันที มีตัวอย่าง ดังนี้

ตัวอย่างที่ 3.4 การกำหนดค่าตัวแปรแบบไม่จำกัดชนิดของตัวแปร

```

1 <?php
2     $color = "red";
3     $number = 12;
4     $age = '12';
5     $sum = 12 + "15";           // ผลลัพธ์ $sum = 27
6 ?>

```

จากตัวอย่างที่ 3.4 เป็นการกำหนดค่าตัวแปรแบบไม่จำกัดชนิดของตัวแปร อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าตัวแปร \$color มีค่าเท่ากับ "red" (เป็นชนิดข้อความ)

บรรทัดที่ 3 กำหนดค่าตัวแปร \$number มีค่าเท่ากับ 12 (เป็นชนิดเลขจำนวนเต็ม)

บรรทัดที่ 4 กำหนดค่าตัวแปร \$age มีค่าเท่ากับ '12' (เป็นชนิดข้อความ)



บรรทัดที่ 5 กำหนดค่าตัวแปร \$sum มีค่าเท่ากับ $12 + '15'$ (เมื่อ 12 คือ ข้อมูลชนิดเลขจำนวนเต็ม นำมาบวกกับ '15' ข้อความ) ลักษณะเช่นนี้ในภาษาอื่นๆ จะไม่สามารถคำนวณได้ แต่สำหรับในภาษา PHP จะปรับเปลี่ยนชนิดของตัวแปรโดยอัตโนมัติ ทำให้สามารถคำนวณค่าได้ ดังนั้นในบรรทัดที่ 5 จึงหมายถึง กำหนดค่าตัวแปร \$sum มีค่าเท่ากับ 27 นั่นเอง

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

3.2.2 การกำหนดค่าตัวแปรโดยการอ้างอิง

การกำหนดค่าตัวแปรโดยการอ้างอิง ช่วยให้สามารถสร้างตัวแปรใหม่ให้มีค่าเข่นเดียวกันกับตัวแปรใดๆ ก็ได้ สามารถสร้างความสัมพันธ์ระหว่างตัวแปรหลักและตัวแปรอ้างอิง เมื่อมีการเปลี่ยนแปลงค่าในตัวแปรใดๆ ค่าของตัวแปรทั้งหมดที่มีการอ้างอิงกันจะมีค่าหรือข้อมูลเดียวกัน การกำหนดตัวแปรโดยอ้างอิงสามารถทำได้โดยการเพิ่มเครื่องหมายแอนด์ (&) ด้านขวาของเครื่องหมายเท่ากับ หรือด้านหน้าของตัวแปร มีตัวอย่างดังนี้

ตัวอย่างที่ 3.5 การกำหนดค่าตัวแปรโดยการอ้างอิง

```

1 <?php
2     $value1 = "Hello";
3     $value2 =& $value1; // $value1 และ $value2 มีค่าเท่ากับ "Hello"
4     $value2 = "Goodbye"; // $value1 และ $value2 มีค่าเท่ากับ "Goodbye"
5     printf("Value 1 is %s and Value 2 is %s.", $value1, $value2);
6 ?>

```

ผลลัพธ์

Value 1 is Goodbye and Value 2 is Goodbye.

จากตัวอย่างที่ 3.5 เป็นการกำหนดค่าตัวแปรโดยการอ้างอิง อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าตัวแปร \$value1 มีค่าเท่ากับ "Hello"

บรรทัดที่ 3 กำหนดค่าตัวแปร \$value2 อ้างอิงค่ากับตัวแปร \$value1 (สมือนเป็นตัวแปรตัวเดียวกัน)

บรรทัดที่ 4 กำหนดค่าตัวแปร \$value2 มีค่าเท่ากับ "Goodbye" ยังผลให้ตัวแปร \$value1 เปลี่ยนแปลงค่าตามไปด้วย คือ ทั้งตัวแปร \$value1 และ \$value2 มีค่าเท่ากับ "Goodbye"

บรรทัดที่ 5 แสดงผลค่าที่เก็บในตัวแปร \$value1 และ \$value2

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

3.2.3 การกำหนดค่าตัวแปรซ้อนตัวแปร

การกำหนดค่าตัวแปรช้อนตัวแปร เป็นการกำหนดค่าให้กับตัวแปร โดยกำหนดผ่านอีกตัวแปรหนึ่ง หรือการกำหนดชื่อตัวแปรตามค่าของตัวแปร จะสามารถกำหนดค่าลักษณะแบบนี้ได้นั้น ค่าของตัวแปรหลักจะต้องเป็นชนิดข้อความเท่านั้น ตัวอย่างดังนี้

ตัวอย่างที่ 3.6 การกำหนดค่าตัวแปรช้อนตัวแปร

```

1 <?php
2     $test = "x";
3     $$test = 5;
4     printf("Test value = %s <br>", $test);
5     printf("X value = %d ", $x);
6 ?>
```

ผลลัพธ์

```

Test value = x
X value = 5
```

จากตัวอย่างที่ 3.6 เป็นการกำหนดค่าตัวแปรช้อนตัวแปร อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าตัวแปร \$test มีค่าเท่ากับ "x" (ข้อความเท่านั้น)

บรรทัดที่ 3 กำหนดค่าตัวแปร \$\$test มีค่าเท่ากับ 5 (ใช้สัญลักษณ์ \$ ซ้อนอีก 1 เครื่องหมาย หมายความว่า กำหนดค่าให้กับค่าของตัวแปร \$test เมื่อค่าของตัวแปร \$test มีค่าเท่ากับ "x" คือ มีตัวแปรเพิ่มขึ้นมาอีก 1 ตัวแปร โดยปริยาย คือ \$x)

บรรทัดที่ 4 แสดงผลค่าที่เก็บในตัวแปร \$test (เป็นชนิดข้อความ)

บรรทัดที่ 5 แสดงผลค่าที่เก็บในตัวแปร \$x (เป็นชนิดเลขจำนวนเต็ม)

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

3.2.4 การกำหนดตัวแปรค่าคงที่

การกำหนดตัวแปรค่าคงที่ เป็นการกำหนดค่าให้กับตัวแปรมีค่าคงที่หนึ่งตลอดการใช้งานในโปรแกรม โดยไม่มีการเปลี่ยนแปลงค่า เมื่อตัวแปรค่าคงที่ถูกกำหนดแล้ว ตัวแปรจะไม่สามารถเปลี่ยนหรือคืนค่าได้อีกต่อไป จะแตกต่างจากการกำหนดค่าตัวแปรปกติ นั่นก็คือ ตัวแปรค่าคงที่ไม่ต้องใช้เครื่องหมาย \$ นำหน้าตัวแปร การสร้างและกำหนดค่าตัวแปรจำเป็นต้องใช้ฟังก์ชัน define() สำหรับการสร้างและกำหนดค่า มีรูปแบบดังนี้

รูปแบบ

```
bool define ( string $name , mixed $value [, bool $case_insensitive = false ] )
```

เมื่อ \$name หมายถึง ชื่อตัวแปร



\$value	หมายถึง	ค่าที่ต้องการกำหนด
\$case_insensitive	หมายถึง	false คือ ชื่อตัวแปรที่จะเรียกใช้ต้องเหมือนกับอักษรที่กำหนด เช่น กำหนดเป็นอักษรพิมพ์ใหญ่ เวลาเรียกใช้ก็ต้องใช้อักษรพิมพ์ใหญ่ด้วย เป็นต้น (ค่าโดยปริยาย) True คือ ไม่สนใจชื่อตัวแปรที่จะเรียกใช้ว่าเป็นอักษรพิมพ์เล็กหรือพิมพ์ใหญ่

ตัวอย่างที่ 3.7 การกำหนดตัวแปรค่าคงที่

```
1 <?php
2     define("CONSTANT", "Hello Suratthani.");
3     echo CONSTANT . "<br>";
4     echo Constant . "<br>";
5     define("GREETING", "Hello Suratthani.", true);
6     echo GREETING . "<br>";
7     echo Greeting . "<br>";
8 ?>
```

ผลลัพธ์

```
Hello Suratthani.
Constant
Hello Suratthani.
Hello Suratthani.
```

จากตัวอย่างที่ 3.7 เป็นการกำหนดตัวแปรค่าคงที่ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 ใช้ฟังก์ชัน define("CONSTANT", "Hello Suratthani.") หมายความว่า กำหนดตัวแปรค่าคงที่ CONSTANT มีค่าเท่ากับ "Hello Suratthani." (กำหนดให้เรียกใช้ตัวแปรด้วย อักษรพิมพ์ใหญ่เท่านั้น)

บรรทัดที่ 3 แสดงผลค่าที่เก็บในตัวแปรค่าคงที่ CONSTANT ผลลัพธ์ที่แสดงผล คือ คำว่า Hello Suratthani.

บรรทัดที่ 4 แสดงผลข้อความ Constant (ข้อความ ไม่ใช่ตัวแปร)

บรรทัดที่ 5 ใช้ฟังก์ชัน define("GREETING", "Hello Suratthani.", true) หมายความว่า กำหนดตัวแปรค่าคงที่ GREETING มีค่าเท่ากับ "Hello Suratthani." (โดยกำหนดพารามิเตอร์เพิ่มเป็น true คือ ไม่สนใจชื่อตัวแปรที่จะเรียกใช้ว่าเป็นอักษรพิมพ์เล็กหรือพิมพ์ใหญ่)

บรรทัดที่ 6 แสดงผลค่าที่เก็บในตัวแปรค่าคงที่ GREETING ผลลัพธ์ที่แสดงผล คือ คำว่า Hello Suratthani.

บรรทัดที่ 7 แสดงผลค่าที่เก็บในตัวแปรค่าคงที่ Greeting ผลลัพธ์ที่แสดงผล คือ คำว่า Hello Suratthani.

บรรทัดที่ 8 สิ้นสุดสคริปต์ PHP

3.3 ขอบเขตของตัวแปร

การประกาศและกำหนดค่าตัวแปรทั้งวิธีกำหนดค่าปกติ หรือกำหนดค่าแบบอ้างอิง หรือกำหนดค่าตัวแปรช้อนตัวแปร หรือกำหนดค่าตัวแปรคงที่ ในภาษา PHP สามารถประกาศได้ทุกที่ในสคริปต์ของ PHP ตำแหน่งของการประกาศตัวแปรมีผลต่อการใช้งานหรือเรียกว่าขอบเขตของตัวแปร (Variable Scope) จะมีผลต่อการเข้าถึงและใช้งาน ขอบเขตของตัวแปรมี 6 ชนิด ประกอบด้วย 1) ตัวแปรเฉพาะที่ (Local variables) 2) ตัวแปรฟังก์ชันพารามิเตอร์ (Function parameters) 3) ตัวแปรสาธารณะ (Global variables) 4) ตัวแปรคงที่ (Static variables) 5) ตัวแปรพิเศษ (Super Global variables) และ 6) ตัวแปรที่มีค่า NULL มีรายละเอียด ดังนี้

3.3.1 ตัวแปรเฉพาะที่

ตัวแปรเฉพาะที่ คือ ตัวแปรที่ถูกประกาศไว้ภายในฟังก์ชัน จะมีขอบเขตเฉพาะภายในฟังก์ชันที่ประกาศใช้ตัวแปรเท่านั้น และฟังก์ชันนั้นๆ ก็ไม่สามารถเรียกใช้ตัวแปรที่ประกาศไว้ภายนอก ฟังก์ชันได้ ถึงแม้ว่าชื่อตัวแปรจะเหมือนกันก็ตาม ถือว่าเป็นตัวแปรคละตัว ลักษณะเช่นนี้จึงเรียกว่า ขอบเขตตัวแปรเฉพาะที่ ตัวอย่างดังนี้

ตัวอย่างที่ 3.8 แสดงตัวอย่างการอ้างอิงตัวแปรเฉพาะที่

```

1 <?php
2     function update_counter () {
3         $counter=$counter*2;
4     }
5     $counter = 10;
6     update_counter ();
7     echo $counter;
8 ?>
```

ผลลัพธ์

10

จากตัวอย่างที่ 3.8 แสดงตัวอย่างการอ้างอิงตัวแปรเฉพาะที่ อธิบายดังนี้



บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 เริ่มต้นฟังก์ชัน update_count () (ฟังก์ชันจะไม่ทำงานหากไม่มีการเรียกใช้ งานฟังก์ชัน แต่ถ้าฟังก์ชันถูกเรียกใช้งานจะทำคำสั่งที่อยู่ภายใต้เครื่องหมาย { } ของฟังก์ชัน เมื่อฟังก์ชันทำงานตามลำดับคำสั่งแล้วเสร็จ จะไปทำงานต่อบรรทัดที่ 7)

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$counter มีค่าเท่ากับค่าของตัวแปรก่อนหน้า คุณด้วย 2 (หากไม่มีการกำหนดค่าเริ่มต้นให้กับตัวแปร โดยปริยายตัวแปรจะมีค่าเป็น NULL คือ ไม่มีค่า เมื่อนำมาคุณด้วย 2 จะมีค่าเท่ากับ 0)

บรรทัดที่ 4 จบการทำงานของฟังก์ชัน

บรรทัดที่ 5 กำหนดให้ตัวแปร \$counter มีค่าเท่ากับ 10

บรรทัดที่ 6 เรียกใช้งานฟังก์ชัน update_count ()

บรรทัดที่ 7 แสดงผลค่าที่เก็บในตัวแปร \$counter ผลลัพธ์ที่แสดง คือ 10 เป็นผลมาจากการกำหนดค่าตัวแปร ในบรรทัดที่ 5 โดยตัวแปรที่อยู่ในบรรทัดที่ 3 มีขอบเขตของตัวแปรเฉพาะที่ นั่นเอง ตามที่ได้กล่าวถึงไว้ข้างต้น

บรรทัดที่ 8 สิ้นสุดสคริปต์ PHP

3.3.2 ตัวแปรฟังก์ชันพารามิเตอร์

ตัวแปรฟังก์ชันพารามิเตอร์ คือ ตัวแปรที่ถูกส่งผ่านไปยังฟังก์ชันหนึ่งๆ การส่งค่าพารามิเตอร์นั้นสามารถส่งได้ครั้งละหลายๆ ตัวแปร โดยทำการคั่นแต่ละพารามิเตอร์ด้วยเครื่องหมาย comma (,) สำหรับรูปแบบการส่งค่าจะกล่าวถึงในบทถัดไป ตัวอย่างการส่งค่าตัวแปรฟังก์ชันพารามิเตอร์ ตัวอย่างดังนี้

ตัวอย่างที่ 3.9 แสดงตัวอย่างการอ้างอิงตัวแปรฟังก์ชันพารามิเตอร์

```

1 <?php
2     function update_counter ($counter) {
3         $counter=$counter*2;
4         return $counter;
5     }
6     $counter = 10;
7     $counter = update_counter ($counter);
8     echo $counter;
9 ?>

```

ผลลัพธ์

จากตัวอย่างที่ 3.9 แสดงตัวอย่างการอ้างอิงตัวแปรฟังก์ชันพารามิเตอร์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 เริ่มต้นฟังก์ชัน update_count (\$counter) (ฟังก์ชันจะไม่ทำงานหากไม่มีการเรียกใช้งานฟังก์ชัน แต่ถ้าฟังก์ชันถูกเรียกใช้งานจะทำการคำนวณที่อยู่ภายใต้ {} ของฟังก์ชัน เมื่อทำงานตามลำดับแล้วเสร็จจะ return ผลลัพธ์ไปให้กับตัวแปร \$counter ในบรรทัดที่ 7 ลักษณะการทำงานจะคล้ายกับตัวอย่างที่ 3.8 แตกต่างที่ฟังก์ชันมีการกำหนดตัวแปรฟังก์ชันพารามิเตอร์สำหรับการรับและส่งค่า)

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$counter มีค่าเท่ากับค่าของตัวแปร \$counter ที่มาจากการเรียกใช้งานฟังก์ชันพารามิเตอร์ ที่ส่งเข้าไปในฟังก์ชัน จากตัวอย่าง คือ เลข 10 คูณด้วย 2 (มีค่าเท่ากับ 20)

บรรทัดที่ 4 ฟังก์ชันส่งค่ากลับด้วย return \$counter (มีค่าเท่ากับ 20) กลับมาให้ตัวแปร \$counter ในบรรทัดที่ 7

บรรทัดที่ 5 จบการทำงานของฟังก์ชัน

บรรทัดที่ 6 กำหนดให้ตัวแปร \$counter มีค่าเท่ากับ 10

บรรทัดที่ 7 กำหนดค่าให้กับตัวแปร \$counter มีค่าเท่ากับ ผลของการเรียกใช้งานฟังก์ชัน update_count () คือ ค่าของผลลัพธ์ที่ได้จากการ return ของฟังก์ชัน

บรรทัดที่ 8 แสดงผลค่าที่เก็บในตัวแปร \$counter ผลลัพธ์ที่แสดง คือ 20

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

3.3.3 ตัวแปรสาธารณะ

ตัวแปรสาธารณะ คือ ตัวแปรที่ประกาศไว้ภายนอกฟังก์ชัน สามารถเข้าถึงจากส่วนใดๆ ของโปรแกรม แต่จะไม่สามารถใช้งานได้ภายในฟังก์ชัน แต่ถ้าต้องการเรียกใช้งานตัวแปรสาธารณะที่อยู่ภายนอกฟังก์ชัน ภายในฟังก์ชันจะต้องใช้คำสั่ง GLOBAL และตามด้วยชื่อของตัวแปรที่ต้องการใช้งาน หรืออ้างอิงถึงตัวแปรที่อยู่ภายนอกฟังก์ชัน ตัวอย่างดังนี้

ตัวอย่างที่ 3.10 แสดงตัวอย่างการอ้างอิงตัวแปรสาธารณะโดยใช้คำสั่ง GLOBAL

```

1 <?php
2     function update_counter () {
3         GLOBAL $counter;
4         $counter=$counter*2;
5     }
6     $counter = 10;
7     update_counter ();
8     echo $counter;           // ผลลัพธ์ คือ 20
9 ?>
```



จากตัวอย่างที่ 3.10 แสดงตัวอย่างการอ้างอิงตัวแปรสาธารณะ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 เริ่มต้นฟังก์ชัน update_count () (ฟังก์ชันจะไม่ทำงานหากไม่มีการเรียกใช้งานฟังก์ชัน แต่ถ้าฟังก์ชันถูกเรียกใช้งานจะทำการสั่งที่อยู่ภายใต้เครื่องหมาย { } ของฟังก์ชัน เมื่อทำงานแล้ว เสิร์จตามลำดับคำสั่ง จะไปบรรทัดที่ 8)

บรรทัดที่ 3 กำหนดให้ฟังก์ชันสามารถเรียกใช้ตัวแปร \$counter (ตัวแปรสาธารณะ) ที่กำหนดไว้ภายนอกฟังก์ชัน ด้วยการเพิ่ม GLOBAL นำหน้าชื่อตัวแปร \$counter

บรรทัดที่ 4 กำหนดให้ตัวแปร \$counter มีค่าเท่ากับ ค่าของตัวแปรก่อนหน้าคูณด้วย 2 (มีค่าเท่ากับ 20)

บรรทัดที่ 5 จบการทำงานของฟังก์ชัน

บรรทัดที่ 6 กำหนดให้ตัวแปร \$counter มีค่าเท่ากับ 10

บรรทัดที่ 7 เรียกใช้งานฟังก์ชัน update_count ()

บรรทัดที่ 8 แสดงผลค่าที่เก็บในตัวแปร \$counter ผลลัพธ์ที่แสดง คือ 20

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

นอกจากการอ้างอิงโดยใช้คำสั่ง GLOBAL แล้ว ยังมีอีกวิธีหนึ่งในการอ้างอิงตัวแปรสาธารณะได้ คือ การใช้ตัวแปรพิเศษ มีลักษณะรูปแบบการใช้งานแบบอาร์เรย์ ตัวแปรพิเศษที่ใช้สำหรับ อ้างอิงตัวแปรสาธารณะ มีชื่อเรียกเฉพาะ คือ \$GLOBALS ตัวอย่างดังนี้

ตัวอย่างที่ 3.11 การอ้างอิงตัวแปรสาธารณะผ่านตัวแปรพิเศษ \$GLOBALS

```
1 <?php
2     function update_counter () {
3         $GLOBALS["counter"] = $GLOBALS["counter"]*2;
4     }
5     $counter = 10;
6     update_counter ();
7     echo $counter;           // ผลลัพธ์ คือ 20
8 ?>
```

จากตัวอย่างที่ 3.11 แสดงตัวอย่างการอ้างอิงตัวแปรสาธารณะ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 เริ่มต้นฟังก์ชัน update_count () (ฟังก์ชันจะไม่ทำงานหากไม่มีการเรียกใช้งานฟังก์ชัน แต่ถ้าฟังก์ชันถูกเรียกใช้งานจะทำการสั่งที่อยู่ภายใต้เครื่องหมาย { } ของฟังก์ชัน เมื่อทำงานแล้ว เสิร์จตามลำดับคำสั่ง จะไปบรรทัดที่ 7)

บรรทัดที่ 3 เรียกใช้และกำหนดค่าตัวแปร Sacharat ขณะผ่านตัวแปรพิเศษ \$GLOBALS["counter"] มีค่าเท่ากับ \$GLOBALS["counter"] คูณด้วย 2 (มีค่าเท่ากับ 20)

บรรทัดที่ 4 จบการทำงานของฟังก์ชัน

บรรทัดที่ 5 กำหนดให้ตัวแปร \$counter มีค่าเท่ากับ 10

บรรทัดที่ 6 เรียกใช้งานฟังก์ชัน update_count ()

บรรทัดที่ 7 แสดงผลค่าที่เก็บในตัวแปร \$counter ผลลัพธ์ที่แสดง คือ 20

บรรทัดที่ 8 สิ้นสุดสคริปต์ PHP

3.3.4 ตัวแปรคงที่

ตัวแปรคงที่ จะทำการเก็บค่าของตัวแปรนั้นๆ ไว้ทุกครั้งที่เรียกใช้ฟังก์ชันกี่ครั้งค่าของตัวแปรที่ถูกประกาศเป็นแบบคงที่จะถูกปรับปรุงค่าไว้เสมอ สามารถใช้ได้กับตัวแปรเฉพาะที่เท่านั้น มีตัวอย่าง ตัวอย่างดังนี้

ตัวอย่างที่ 3.12 การใช้งานตัวแปรคงที่

```

1 <?php
2
3     function show_number ( ) {
4         STATIC $count = 0;
5         $count=$count+10;
6         echo $count;
7         echo "<br>";
8     }
9     show_number ( );
10    show_number ( );
11    show_number ( );
12 ?>
```

ผลลัพธ์ของการทำงานถ้าหากไม่มีการกำหนด STATIC ไว้ด้านหน้าของตัวแปร \$count ดังนี้

10

10

10

แต่ผลลัพธ์ของการทำงานเมื่อมีการกำหนด STATIC ไว้ด้านหน้าตัวแปร \$count จะได้ผลลัพธ์ ดังนี้

10

20

30



3.3.5 ตัวแปรพิเศษ

ตัวแปรพิเศษ คือ ตัวแปรที่กำหนดไว้ล่วงหน้าภายในระบบของภาษา PHP สามารถเข้าถึง และเรียกใช้ตัวแปรพิเศษได้จากทุกที่ภายในสคริปต์ที่ทำงานอยู่ เช่น เรียกดูรายละเอียดการใช้งานของผู้ใช้ ณ ปัจจุบัน รายละเอียดของเว็บเซิร์ฟเวอร์ ระบบปฏิบัติการ และอื่นๆ ตัวแปรพิเศษที่สำคัญ ประกอบด้วย 1) `$_SERVER` 2) `$_GET` 3) `$_POST` 4) `$_ENV` 5) `$_COOKIE` และ 6) `$_SESSION` รายละเอียด ดังนี้

1) `$_SERVER` เป็นตัวแปรพิเศษที่ใช้สำหรับเก็บค่าสารสนเทศต่างๆ ของเว็บเซิร์ฟเวอร์ที่กำลังทำงานอยู่ ตัวแปรพิเศษ `$_SERVER` จะเก็บอยู่ในรูปแบบของอาร์เรย์ ประกอบไปด้วยอิลิเม้นท์ (Element) ที่สามารถเรียกดูข้อมูลได้จำนวนมาก โดยอิลิเม้นท์แต่ละตัวของตัวแปรพิเศษ `$_SERVER` จะทำหน้าที่บอกสารสนเทศแตกต่างกัน โดยมีรูปแบบการเรียกดูข้อมูลตัวแปรพิเศษ `$_SERVER` ดังนี้

รูปแบบ

```
$_SERVER["ชื่ออิลิเม้นท์"]
```

ตารางที่ 3.1 แสดงชื่ออิลิเม้นท์ของตัวแปรพิเศษ `$_SERVER` ที่สำคัญ และใช้งานบ่อย

ชื่ออิลิเม้นท์	คำอธิบาย
<code>PHP_SELF</code>	เก็บชื่อไฟล์ที่กำลังทำงานอยู่ อิลิเม้นท์นี้มักใช้เมื่อต้องการอ้างถึงหรือเรียกใช้การประมวลผลที่ไฟล์สคริปต์เดิมในขณะนั้น
<code>HTTP_REFERER</code>	เก็บค่าที่อยู่ URL ของหน้าเว็บเพจที่อ้างถึงในหน้าเว็บเพจปัจจุบัน
<code>REMOTE_ADDR</code>	เก็บค่า IP Address ของเครื่องคอมพิวเตอร์ที่กำลังเข้ามาเว็บเพจนั้นอยู่
<code>QUERY_STRING</code>	เก็บค่า Query String ทั้งหมดที่ต่อท้ายจาก URL ของหน้าเว็บเพจปัจจุบัน

การเรียกใช้ตัวแปรพิเศษ `$_SERVER` เพื่อแสดง IP Address ที่กำลังใช้งานอยู่ ตัวอย่างดังนี้

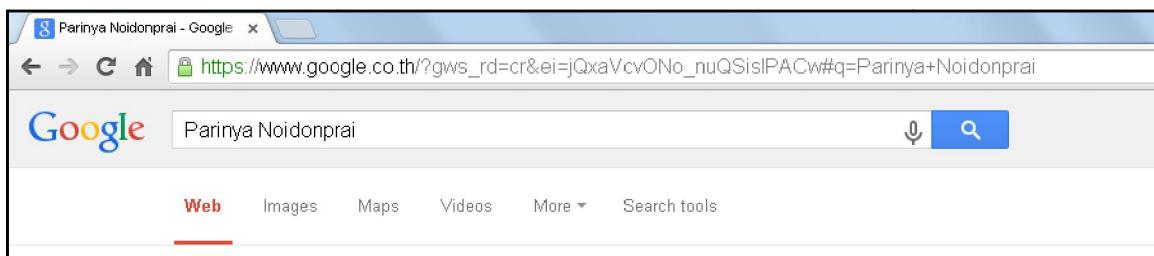
ตัวอย่างที่ 3.13 การใช้งานตัวแปรพิเศษ `$_SERVER` เพื่อเรียกดู IP Address

```
1 <?php
2     printf ("Your IP address is: %s", $_SERVER["REMOTE_ADDR"]);
3 ?>
```

ผลลัพธ์

```
Your IP address is: 127.0.0.1
```

2) `$_GET` เป็นตัวแปรพิเศษใช้เก็บค่าของตัวแปรที่ถูกส่งมากับสคริปต์ในรูปแบบของ Query String ต่อท้าย URL ดังภาพที่ 3.1



ภาพที่ 3.1 แสดงตัวอย่างการส่งค่าแบบ Query String

จากภาพที่ 3.1 เป็นการเชื่อมโยงไปยังหน้าค้นหาโดยมีการส่งค่าตัวแปรไป โดยตัวแปรที่ส่งไปจะอยู่หลังเครื่องหมาย "?" และสามารถส่งได้ครั้งละหลายๆ ตัวแปร โดยใช้สัญลักษณ์ "&" เชื่อมแต่ละตัวแปร

3) `$_POST` เป็นตัวแปรพิเศษใช้เก็บค่าของตัวแปรที่ถูกส่งมาจากการเข้าชมเว็บเพจที่มีการส่งค่าของมูลแบบ POST ส่วนใหญ่จะอยู่ในรูปแบบของฟอร์มที่กำหนด "method" เป็น POST เช่นเดียวกับ `$_GET` แต่ไม่แสดงผลตัวแปรและค่าตัวแปรบน URL

การกำหนด "method" โดยส่วนใหญ่จะกำหนดเป็น POST เพื่อให้ส่งข้อมูลจาก field ต่างๆ ไป ในส่วนหัวของโปรโตคอล HTTP request สามารถส่งข้อมูลจำนวนมากได้ ส่วนการส่งด้วย method GET จะใช้ในกรณีที่มี field และความยาวข้อมูลน้อย และไม่เป็นความลับ เนื่องจากการส่งแบบ GET เป็นการสร้าง Query String จาก field ต่างๆ ต่อท้าย URL โดยอัตโนมัติ และการส่งแบบ GET ไม่สามารถใช้ได้กับฟอร์มที่มีการส่งไฟล์

4) `$_ENV` เป็นตัวแปรพิเศษใช้สำหรับเก็บตัวแปรที่จัดเก็บสภาพแวดล้อมทั่วไป และค่าต่างๆ ของระบบที่ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ ผลลัพธ์ที่ได้จะแตกต่างกันตามรูปแบบของตัวแปรที่ใช้ เช่น `$COMPUTERNAME` เป็นตัวแปร Environment ชนิดหนึ่งที่ใช้แสดงชื่อของเครื่องคอมพิวเตอร์

5) `$_COOKIE` เป็นตัวแปรพิเศษที่ใช้สำหรับเก็บค่าของข้อมูลไว้ระยะเวลานึง เพื่อใช้งานกับเอกสารเว็บเพจทั้งเว็บไซต์ (การเก็บข้อมูลไว้ในตัวแปรพิเศษ `$_COOKIE` ระบบจะทำการสร้างไฟล์เก็บตัวแปรพิเศษ `$_COOKIE` ไว้ที่เครื่องคลอเอนต์) การที่จะใช้ตัวแปรพิเศษ `$_COOKIE` ได้นั้น จะต้องมีการสร้างตัวแปรพิเศษ `$_COOKIE` ก่อน โดยใช้ฟังก์ชัน `setcookie()` รายละเอียดการใช้งาน จะกล่าวถึงในบทต่อไป

6) `$_SESSION` เป็นตัวแปรพิเศษที่ใช้สำหรับเก็บค่าของข้อมูลไว้ระยะเวลานึง เพื่อใช้งานกับเอกสารเว็บเพจทั้งเว็บไซต์เหมือนกับตัวแปรพิเศษ `$_COOKIE` (การเก็บข้อมูลไว้ในตัวแปรพิเศษ `$_SESSION` นั้น ระบบจะทำการสร้างไฟล์เก็บตัวแปรพิเศษ `$_SESSION` ไว้ในผู้ของเว็บเซิร์ฟเวอร์) การที่จะใช้ตัวแปรพิเศษ `$_SESSION` ได้นั้น จะต้องมีการสั่งเริ่มใช้เซสชัน โดยใช้คำสั่ง `session_start()` รายละเอียดการใช้งาน จะกล่าวถึงในบทต่อไป

ตัวแปรพิเศษ `$_COOKIE` และ ตัวแปรพิเศษ `$_SESSION` เมื่อกัน ก็คือ ใช้สำหรับเก็บค่าตัวแปรไว้เพื่อเรียกใช้ในหน้าเว็บเพจต่างๆ ภายใต้เว็บไซต์ และทั้งคู่มีความแตกต่างกัน ก็คือ อายุของตัวแปร โดยอายุของตัวแปรพิเศษ `$_COOKIE` ถูกกำหนดด้วยเวลา ส่วนอายุของตัวแปรพิเศษ

`$_SESSION` ถูกกำหนดด้วยการทำางานของบราวเซอร์ เมื่อเคลื่อนต์ทำการปิดโปรแกรมเว็บบราวเซอร์ ตัวแปรพิเศษ `$_SESSION` ก็จะถูกทำลาย

3.3.6 ตัวแปรที่มีค่า NULL

ค่า NULL คือ ตัวแปรว่างเปล่าหรือไม่มีค่าใดๆ ทั้งนี้หากสร้างตัวแปรขึ้นมาโดยไม่มีการทำหนนค่าใดๆใหกับตัวแปร จะทำให้ตัวแปรนั้นมีค่าเป็น NULL หรืออาจเกิดจากการสร้างตัวแปรแล้วกำหนดค่าให้เท่ากับ NULL เช่น ในกรณีต่อไปนี้ ตัวแปร `$a` และ `$b` จะมีค่าเป็น NULL

ตัวอย่างที่ 3.14 ตัวแปรที่มีค่า NULL

```
1 <?php
2     $a;
3     $b = NULL;
4     printf ("Value A is: %s <br> Value B is: %s", $a, $b);
5 ?>
```

สำหรับในภาษา PHP ตัวแปรที่มีค่า NULL เมื่อนำไปใช้งาน จะมีผลดังนี้

- 1) หากนำไปคำนวณจะมีค่า เทียบเท่ากับ 0
- 2) หากนำไปใช้ในแบบข้อความ จะมีค่า เทียบเท่ากับ "" หรือค่าว่าง
- 3) หากนำไปเปรียบเทียบทาทางตรรกะ จะเทียบเท่า กับค่า false

3.4 การตรวจสอบและยกเลิกตัวแปร

การเขียนโปรแกรมในระดับสูงนี้ไปจะเป็นการทำางานแบบไดนามิก ในบางกรณีไม่อาจทราบล่วงหน้าได้ว่า ข้อมูล (ตัวแปร) ที่ต้องการใช้งานนั้นมีอยู่จริง หรือเก็บค่าไดๆ เอาไว้หรือไม่ จึงต้องมีคำสั่งในการตรวจสอบข้อมูลเหล่านี้ก่อนนำไปใช้เพื่อป้องกันข้อผิดพลาด นอกจากนี้ตัวแปรที่สร้างขึ้นอาจจะใช้งานเพียงชั่วระยะเวลาหนึ่ง และหลังจากนั้นหากมีตัวแปรน้อยต่อไปอาจจะส่งผลกระทบการทำงานของระบบ ดังนั้นจึงต้องมีคำสั่งสำหรับการตรวจสอบและยกเลิกตัวแปร พังก์ชันที่สำคัญประกอบด้วย 1) `isset ()` 2) `empty ()` และ 3) `unset ()` มีรายละเอียดดังนี้

3.4.1 `isset ()`

เป็นพังก์ชันที่ใช้สำหรับตรวจสอบว่ามีตัวแปรตามที่ระบุอยู่จริงหรือไม่ หากไม่มีตัวแปร (เนื่องจากยังไม่ได้ประกาศใช้ตัวแปร) หรือตัวแปรมีค่าเป็น NULL พังก์ชัน `isset ()` จะส่งค่า false แต่หากมีการประกาศใช้ตัวแปรอยู่จริง และไม่มีค่าเป็น NULL พังก์ชัน `isset ()` จะส่งค่า true มีรูปแบบดังนี้

```
bool isset ( mixed $var [, mixed $... ] )
```

เมื่อ `$var` หมายถึง ตัวแปรที่ต้องการตรวจสอบ

ตัวอย่างที่ 3.15 การใช้ฟังก์ชัน iset () สำหรับตรวจสอบการกำหนดตัวแปร

```

1 <?php
2     $value1 = NULL;
3     $value2 = 2015;
4     var_dump(isset($value1));
5     echo "<br>";
6     var_dump(isset($value2));
7 ?>
```

ผลลัพธ์

bool (false)

bool (true)

จากตัวอย่างที่ 3.15 ตัวอย่างการใช้ฟังก์ชัน iset () สำหรับตรวจสอบตัวแปร อธิบายดังนี้

- บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP
- บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$value1 มีค่าเท่ากับ NULL
- บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$value2 มีค่าเท่ากับ 2015
- บรรทัดที่ 4 แสดงผลการตรวจสอบตัวแปร \$value1 ว่าได้ประกาศใช้ และ ไม่เป็นค่า NULL จริงหรือเท็จ หากเป็นจริงจะแสดงคำว่า bool(true) หากเป็นเท็จจะแสดงคำว่า bool(false)
- บรรทัดที่ 5 จบบรรทัดใหม่
- บรรทัดที่ 6 แสดงผลการตรวจสอบตัวแปร \$value2 ว่าได้ประกาศใช้ และ ไม่เป็นค่า NULL จริงหรือเท็จ หากเป็นจริงจะแสดงคำว่า bool(true) หากเป็นเท็จจะแสดงคำว่า bool(false)
- บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

3.4.2 empty ()

เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบตัวแปร มีค่าว่างจริงหรือไม่ ในภาษา PHP จะถือว่าตัวแปรมีค่าว่างในลักษณะต่อไปนี้

- 1) มีค่าเป็น NULL
- 2) มีค่าเป็นข้อความว่าง หรือ ""
- 3) มีค่า 0 หรือ "0"

ถ้าตัวแปรมีค่าว่างจะได้ผลลัพธ์เป็น true หากตัวแปรมีการกำหนดค่า จะได้ผลลัพธ์เป็น false มีรูปแบบดังนี้

รูปแบบ

bool empty (mixed \$var)

เมื่อ \$var หมายถึง ตัวแปรที่ต้องการตรวจสอบ



ตัวอย่างที่ 3.16 การใช้ฟังก์ชัน empty () สำหรับตรวจสอบค่าว่างของตัวแปร

```
1 <?php  
2     $value1 = NULL;  
3     $value2 = 2015;  
4     var_dump (empty ($value1) );  
5     echo "<br>";  
6     var_dump (empty ($value2) );  
7 ?>
```

ผลลัพธ์

bool (true)

bool (false)

จากตัวอย่างที่ 3.16 ตัวอย่างการใช้ฟังก์ชัน empty () สำหรับตรวจสอบค่าว่างของตัวแปร อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$value1 มีค่าเท่ากับ NULL

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$value2 มีค่าเท่ากับ 2015

บรรทัดที่ 4 แสดงผลการตรวจสอบตัวแปร \$value1 เป็นค่าว่าง หรือ มีค่าเป็น NULL

จริงหรือเท็จ หากเป็นจริงจะแสดงคำว่า bool(true) หากเป็นเท็จจะแสดงคำว่า bool(false)

บรรทัดที่ 5 ขึ้นบรรทัดใหม่

บรรทัดที่ 6 แสดงผลการตรวจสอบตัวแปร \$value2 เป็นค่าว่าง หรือ มีค่าเป็น NULL

จริงหรือเท็จ หากเป็นจริงจะแสดงคำว่า bool(true) หากเป็นเท็จจะแสดงคำว่า bool(false)

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

3.4.3 unset ()

เป็นฟังก์ชันที่ใช้สำหรับยกเลิกการใช้ตัวแปรที่ระบุ เมื่อไม่ต้องการใช้ตัวแปรตัวนั้นอีกต่อไป เพื่อคืนทรัพยากรให้แก่ระบบ มีรูปแบบดังนี้

```
void unset ( mixed $var [, mixed $... ] )
```

เมื่อ \$var หมายถึง ตัวแปรที่ต้องการยกเลิก

ตัวอย่างที่ 3.17 การใช้ฟังก์ชัน unset () สำหรับยกเลิกการใช้ตัวแปร

```
1 <?php  
2     $value = "Parinya";  
3     var_dump (isset ($value) );  
4     echo "<br>";
```

```

5      unset ($value);
6      var_dump (isset ($value1) );
7  ?>

```

ผลลัพธ์

```

bool (true)
bool (false)

```

จากตัวอย่างที่ 3.17 ตัวอย่างการใช้ฟังก์ชัน `unset()` สำหรับยกเลิกตัวแปร อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร `$value` มีค่าเท่ากับ "Parinya"

บรรทัดที่ 3 แสดงผลการตรวจสอบตัวแปร `$value` ว่าได้ประกาศใช้ และ ไม่เป็นค่า `NULL` จริงหรือเท็จ หากเป็นจริงจะแสดงคำว่า `bool (true)` หากเป็นเท็จจะแสดงคำว่า `bool (false)` จากตัวอย่างจะเห็นว่าบรรทัดแรกที่แสดงผล คือ `bool (true)` อันเนื่องมาจากตัวแปร `$value` มีการประกาศใช้ และ ไม่เป็นค่า `NULL`

บรรทัดที่ 4 ขึ้นบรรทัดใหม่

บรรทัดที่ 5 ยกเลิกการใช้งานตัวแปร `$value`

บรรทัดที่ 6 แสดงผลการตรวจสอบตัวแปร `$value` ว่าได้ประกาศใช้ และ ไม่เป็นค่า `NULL` จริงหรือเท็จ หากเป็นจริงจะแสดงคำว่า `bool (true)` หากเป็นเท็จจะแสดงคำว่า `bool (false)` จากตัวอย่างจะเห็นว่าบรรทัดที่สองผลลัพธ์ คือ `bool (false)` อันเนื่องมาจากตัวแปร `$value` ถูกยกเลิก ด้วยฟังก์ชัน `unset()`

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

3.5 ตัวดำเนินการ และนิพจน์

ตัวดำเนินการ (Operators) คือ สัญลักษณ์ที่ใช้กำหนดรูปแบบการประมวลผลข้อมูล

นิพจน์ (Expressions) คือ การกระทำเพื่อให้ได้ผลลัพธ์ค่านึงค่า ประกอบไปด้วยตัวถูกกระทำ (Operands) และตัวดำเนินการ เช่นเรียงกันไป เช่น $3 * 2 - 1 + 7$ หรือ $a * 5$ เป็นต้น

ลำดับความสำคัญของตัวดำเนินการ มีความสำคัญมากในการเขียนโปรแกรม ถ้าจัดลำดับผิด ก็อาจจะทำให้โปรแกรมของเกิดข้อผิดพลาด (Bug) ได้ และบางที่จะทำให้หาจุดผิดพลาดนั้นๆ ยากด้วย ลำดับความสำคัญของตัวดำเนินการแต่ละตัวจะมีความสำคัญแตกต่างกัน โดยลำดับความสำคัญสูงสุดจะ ถูกดำเนินการก่อนลำดับความสำคัญต่ำ ตัวอย่าง ดังนี้

ตัวอย่างที่ 3.18 แสดงนิพจน์และตัวดำเนินการ

```
2 + 4 * 3
```



จากตัวอย่างที่ 3.18 แสดงนิพจน์และตัวดำเนินการ ถ้าลองมาคำนวณกันเอง ก็คงจะได้ผลลัพธ์ค่าแตกต่างกันไปเนื่องจากว่าไม่รู้ว่าจะนำตัวไหนมาคำนวณกันก่อน จากตัวอย่างเครื่องหมายที่มีความสำคัญสูงคือเครื่องหมายคูณ รองลงมา คือ บวก หมายความว่าจะดำเนินการคำนวณ $4 * 3$ ก่อน แล้วค่อยนำผลลัพธ์นี้ไปบวกกับ 2 ผลลัพธ์จึงมีค่าเท่ากับ 14

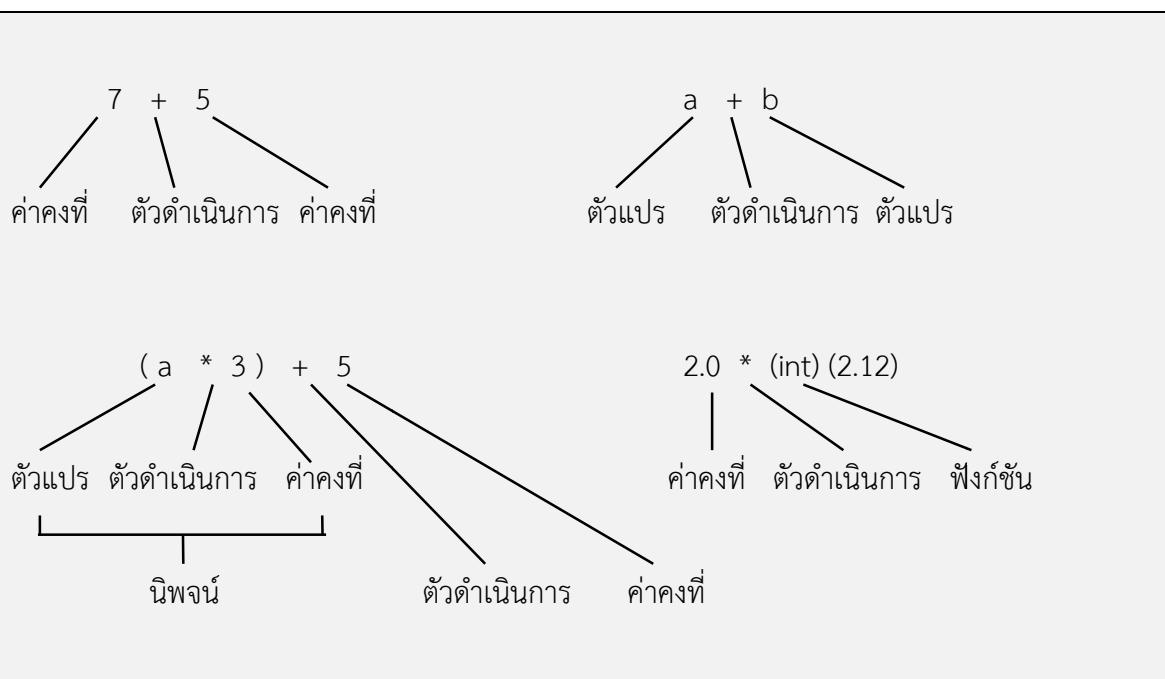
แต่ถ้าไม่ต้องการให้เกิดความยุ่งยากในการจัดลำดับความสำคัญ ขอแนะนำให้ใช้เครื่องหมายวงเล็บ () ในการแบ่งการคำนวณออกจากกัน ตัวอย่างดังนี้

ตัวอย่างที่ 3.19 แสดงนิพจน์และลำดับความสำคัญของตัวดำเนินการ

$$(2 + 4) * 3$$

หรือ ขึ้นอยู่กับว่าจะให้นิพจน์ใดคำนวณก่อน-หลัง การคำนวณจะเกิดขึ้นในวงเล็บก่อนเสมอ

$$2 + (4 * 3)$$



ภาพที่ 3.2 แสดงองค์ประกอบของการคำนวณทางคณิตศาสตร์

3.5.1 ประเภทของตัวดำเนินการ

ประเภทของตัวดำเนินการ สามารถแบ่งได้ 7 ประเภท ประกอบด้วย 1) ตัวดำเนินการทางคณิตศาสตร์ 2) ตัวดำเนินการเชื่อมต่อข้อความ 3) ตัวดำเนินการกำหนดค่า 4) ตัวดำเนินการเพิ่ม และลดค่า 5) ตัวดำเนินการเปรียบเทียบ 6) ตัวดำเนินการเปรียบเทียบทางตระกูล และ 7) ตัวดำเนินการอื่นๆ มีรายละเอียด ดังต่อไปนี้

- 1) ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators) ประกอบด้วย

ตารางที่ 3.2 ตัวดำเนินการทางคณิตศาสตร์

ลำดับความสำคัญ	สัญลักษณ์	ความหมาย
1	*	คูณ (Multiplication)
	/	หาร (Division)
	%	เศษจากการหาร (Modulus)
2	+	บวก (Addition)
	-	ลบ (Subtraction)

จากตารางที่ 3.2 ตัวดำเนินการทางคณิตศาสตร์ จะมีลำดับความสำคัญอยู่ 2 ระดับ คือ ระดับที่ 1 ที่มีความสำคัญมากสุด ประกอบด้วย *, / และ % และระดับที่ 2 ประกอบด้วย + และ - ในกรณีที่นิพจน์ใดๆ มีตัวดำเนินการระดับเดียวกันปรากฏอยู่ในนิพจน์ นิพจน์จะเริ่มคำนวณจากซ้ายไปขวา เสมอ รูปแบบการ

ตัวอย่างที่ 3.20 แสดงตัวอย่างการคำนวณทางคณิตศาสตร์และกำหนดค่าให้กับตัวแปร

```

1 <?php
2     $a = 2*100/10;
3     $b = 100/10%2;
4     $c = 100%80*2;
5     $d = 100/10+30-20;
6     ?>

```

จากตัวอย่างที่ 3.20 ตัวอย่างการคำนวณทางคณิตศาสตร์และลำดับความสำคัญ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$a มีค่าเท่ากับ 20 (เพราะลำดับความสำคัญ ของตัวดำเนินการคูณและหารมีลำดับความสำคัญเท่ากัน การคำนวณจะเริ่มจากซ้ายไปขวา คือ 2 คูณกับ 100 มีค่าเท่ากับ 200 และนำไปหารด้วย 10 จึงมีค่าเท่ากับ 20)

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$b มีค่าเท่ากับ 0 (เพราะลำดับความสำคัญ ของตัวดำเนินการหารและเศษจากการหาร มีลำดับความสำคัญเท่ากัน การคำนวณจะเริ่มจากซ้ายไปขวา คือ 100 หารกับ 10 มีค่าเท่ากับ 10 และนำไปหารผลเศษจากการหารด้วย 10 จึงมีค่าเท่ากับ 0)

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$c มีค่าเท่ากับ 40 (เพราะลำดับความสำคัญ ของตัวดำเนินการเศษจากการหารและคูณ มีลำดับความสำคัญเท่ากัน การคำนวณจะเริ่มจากซ้ายไปขวา คือ 100 หารผลเศษจากการหารกับ 80 มีค่าเท่ากับ 20 และนำไปคูณด้วย 2 จึงมีค่าเท่ากับ 40)



บรรทัดที่ 5 กำหนดค่าให้กับตัวแปร \$d มีค่าเท่ากับ 20 (เพรากำหนดจำนวนจะเริ่มต้นจากตัวดำเนินการหารก่อนเนื่องจากตัวดำเนินการหารมีความสำคัญมากกว่าบวกและลบ ดังนั้นจะเริ่มต้นจาก 100 หารด้วย 10 มีค่าเท่ากับ 10 ลำดับถัดไปบวกและลบมีลำดับความสำคัญเท่ากัน จะเริ่มจากซ้ายไปขวา คือ นำผลลัพธ์ที่ได้จากการหาร คือ 10 ไปบวกด้วย 30 มีค่าเท่ากับ 40 แล้วจึงลบด้วย 20 ผลลัพธ์ที่ได้จะมีค่าเท่ากับ 20)

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

ในทางปฏิบัติเพื่อหลีกเลี่ยงข้อผิดพลาดจากลำดับความสำคัญของตัวดำเนินการ ควรใช้วงเล็บในการจัดแบ่งกลุ่มให้ชัดเจน

2) ตัวดำเนินการเชื่อมต่อข้อความ (String Concatenation)

สำหรับในภาษา PHP จะใช้เครื่องหมายจุด (.) ในการเชื่อมต่อข้อความ ตัวอย่างตัวดำเนินการสำหรับเชื่อมต่อข้อความ ดังตัวอย่างที่ 3.21

ตัวอย่างที่ 3.21 การใช้เครื่องหมายจุด (.) ในการเชื่อมต่อข้อความ

```
1 <?php
2     $str = "My" . "SQL";           // $str = "MySQL"
3     $fname = "Parinya";
4     $lname = "Noidonprai";
5     $fullname = $fname . " " . $lname
6     echo $fullname;
7 ?>
```

ผลลัพธ์

Parinya Noidonprai

จากตัวอย่างที่ 3.21 ตัวอย่างการเชื่อมข้อความ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$str มีค่าเท่ากับข้อความ "MySQL"

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$fname มีค่าเท่ากับข้อความ "Parinya"

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$lname มีค่าเท่ากับข้อความ "Noidonprai"

บรรทัดที่ 5 กำหนดค่าให้กับตัวแปร \$fullname มีค่าเท่ากับตัวแปร \$fname เชื่อมด้วยช่องว่างและค่าของตัวแปร \$lname คือ มีค่าเท่ากับข้อความ "Parinya Noidonprai"

บรรทัดที่ 6 แสดงผลค่าที่เก็บไว้ในตัวแปร \$fname

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

3) ตัวดำเนินการกำหนดค่า (Assignment)

ตัวดำเนินการกำหนดค่า คือ การกำหนดค่าให้กับตัวแปรที่อยู่ทางด้านซ้ายของตัวดำเนินการกำหนดค่า ด้วยค่าที่อยู่ทางขวาโดยประกอบด้วยตัวดำเนินการดังต่อไปนี้

ตารางที่ 3.3 ตัวดำเนินการกำหนดค่า

สัญลักษณ์	ความหมาย	ตัวอย่าง
=	กำหนดค่าปกติ	<code>\$x = 10;</code>
+=	นำค่าที่กำหนดไปบวกเพิ่มจากค่าเดิม ตัวแปร แล้วนำผลลัพธ์ที่ได้จัดเก็บไว้ ในตัวแปรเดิม	<code>\$x = 10; \$x += 8; // \$x = 18</code>
-=	ลดค่าตัวแปรลงเท่ากับค่าที่ระบุ	<code>\$x = 10; \$x -= 8; // \$x = 2</code>
*=	คูณค่าเดิมของตัวแปรด้วยค่าที่ระบุ	<code>\$x = 10; \$x *= 8; // \$x = 80</code>
/=	หารค่าเดิมของตัวแปรด้วยค่าที่ระบุ	<code>\$x = 16; \$x /= 8; // \$x = 2</code>
%=	นำค่าที่ระบุไปหารค่าเดิมของตัวแปร แต่จะเอาเฉพาะเศษจากการหาร เท่านั้น หรือเรียกว่าการหารแบบ Modulus	<code>\$x = 10; \$x %= 3; // \$x = 1</code>
.=	ใช้ในการเชื่อมต่อข้อความโดยนำ ข้อความด้านขวาเมื่อไปต่อท้าย ข้อความด้านซ้ายเมื่อ	<code>\$x = "PHP"; \$x .= "/MySQL"; // \$x = "PHP/MySQL"</code>

4) ตัวดำเนินการเพิ่ม และลดค่า (Increment & Decrement)

ประกอบด้วยตัวดำเนินการดังต่อไปนี้

ตารางที่ 3.4 ตัวดำเนินการสำหรับเพิ่ม และลดค่า

สัญลักษณ์	ความหมาย	ตัวอย่าง
++	เพิ่มค่าในตัวแปรอีก 1 ค่า	<code>\$x = 10; \$x++; // \$x = 11</code>
--	ลดค่าในตัวแปรลง 1 ค่า	<code>\$x = 10; \$x--; // \$x = 9</code>



ข้อควรระวังเกี่ยวกับการใช้ตัวดำเนินการ ++ และ --

การวางแผนตัวดำเนินการ ++ หรือ -- ไว้ด้านหน้า หรือหลังตัวแปร หากตัวแปรนั้นอยู่เดียวๆ ค่าที่ได้จะไม่ต่างกัน เช่น กรณีตัวอย่างต่อไปนี้

ตัวอย่างที่ 3.22 ข้อควรระวังการวางแผนตัวดำเนินการ ++ หรือ -- ไว้ด้านหน้า หรือหลังตัวแปร

```
<?php
$x = 10;
$y = 10;
$x++; // ผลลัพธ์ คือ $x = 11
++$y; // ผลลัพธ์ คือ $y = 11
?>
```

แต่หากนำไปใช้ในรูปแบบของนิพจน์ หรือจะทำกับค่าอื่นๆ ด้วย ค่าที่ได้อาจแตกต่างกันไป เช่น สองกรณีต่อไปนี้

ตัวอย่างที่ 3.23 การเปรียบเทียบรูปแบบของการใช้นิพจน์สองกรณี

$\$x = 10;$ $\$y = 20;$ $\$y += ++$x; // \$x = 11, \$y = 31$ หากกำหนดแบบนี้ y จะมีค่าเท่ากับ $(20+1)+10=31$ นั่นคือ จะเพิ่มค่า x ขึ้นไปอีก 1 ก่อนแล้วค่อยนำไปบวกกับ y	$\$x = 10;$ $\$y = 20;$ $\$y += \$x++; // \$x = 11, \$y = 30$ หากกำหนดแบบนี้ y จะมีค่าเท่ากับ $20+10=30$ นั่นคือ จะนำค่า x เดิมไปบวกกับ y ก่อน แล้วค่อยเพิ่มค่า x ขึ้นไปอีก 1
--	--

5) ตัวดำเนินการเปรียบเทียบ (Comparison)

ตัวดำเนินการเปรียบเทียบใช้ในการเปรียบเทียบหาค่าความจริงระหว่าง 2 นิพจน์ โดยผลลัพธ์ที่ได้จะเป็นได้เพียง true หรือ false อีกอย่างหนึ่งเท่านั้น ตัวดำเนินการในกลุ่มนี้ มีดังนี้

ตารางที่ 3.5 ตัวดำเนินการเปรียบเทียบ

สัญลักษณ์	ความหมาย	สัญลักษณ์	ความหมาย
<	น้อยกว่า	==	เท่ากัน
<=	น้อยกว่า หรือเท่ากับ	====	เท่ากันทั้งหมดทั้งค่า และชนิดข้อมูล
>	มากกว่า	!=	ไม่เท่ากัน
=>	มากกว่า หรือเท่ากับ		

ตัวอย่างที่ 3.24 ตัวอย่างการใช้ตัวดำเนินการเปรียบเทียบ

```

1 <?php
2     $a = (10 <= 9);
3     $b = (10 == 10);
4     $c = (10 == "10");
5     $d = (10 === "10");
6     $e = ("php" == "PHP");
7 ?>

```

จากตัวอย่างที่ 3.24 ตัวอย่างการใช้ตัวดำเนินการเปรียบเทียบ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$a มีค่าเท่ากับ false (เพราะผลของการเปรียบเทียบค่า คือ 10 มีค่าน้อยกว่าหรือเท่ากับ 9 ไม่เป็นความจริง)

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$b มีค่าเท่ากับ true (เพราะผลของการเปรียบเทียบค่า คือ 10 มีค่าเท่ากับ 10 เป็นความจริง)

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$c มีค่าเท่ากับ true (เพราะผลของการเปรียบเทียบเฉพาะค่า คือ 10 มีค่าเท่ากับ "10" เป็นความจริง)

บรรทัดที่ 5 กำหนดค่าให้กับตัวแปร \$d มีค่าเท่ากับ false (เพราะผลของการเปรียบเทียบค่าและชนิด คือ 10 มีค่าเท่ากับ "10" ไม่เป็นความจริง คือ 10 ตัวแรกเป็นชนิดเลขจำนวนเต็ม ส่วน "10" เป็นข้อความ)

บรรทัดที่ 6 กำหนดค่าให้กับตัวแปร \$e มีค่าเท่ากับ false (เพราะผลของการเปรียบเทียบชนิดของตัวพิมพ์ไม่เหมือนกัน จึงทำให้ไม่เป็นความจริง)

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

6) ตัวดำเนินการเปรียบเทียบทางตรรกะ (Bitwise Operators)

ตัวดำเนินการเปรียบเทียบทางตรรกะ เป็นการเปรียบเพื่อหาค่าความจริงระหว่าง 2 นิพจน์ เช่น หากพิสูจน์นิพจน์แรกเป็นจริง และนิพจน์ที่สองเป็นเท็จ แล้วผลลัพธ์จะออกมาเป็นอย่างไร เป็นต้น ตัวดำเนินการในการเปรียบเทียบทางตรรกะมีดังต่อไปนี้

ตารางที่ 3.6 ตัวดำเนินการเปรียบเทียบทางตรรกะ

ตัวดำเนินการ	การเปรียบเทียบ	ผลลัพธ์
! หรือ not	!(true)	true
	!(false)	false



ตารางที่ 3.6 (ต่อ)

ตัวดำเนินการ	การเปรียบเทียบ	ผลลัพธ์
&& หรือ and	true && true	true
	true && false	false
	false && false	false
หรือ or	true true	true
	true false	true
	false false	false
^ หรือ xor	true ^ true	false
	true ^ false	true
	false ^ false	false

ตัวอย่างที่ 3.25 ตัวอย่างผลลัพธ์ของการเปรียบเทียบระหว่าง 2 นิพจน์

```

1 <?php
2 $a = !(1 == 2);
3 $b = (1 != 2) && (1>0);
4 $c = (1 == 2) && (1>0);
5 $d = (1 == 2) || (1>0);
6 ?>

```

จากตัวอย่างที่ 3.25 ตัวอย่างการใช้ตัวดำเนินการเปรียบเทียบ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$a มีค่าเท่ากับ true เมื่อ \$a = !(false)

บรรทัดที่ 3 กำหนดให้ตัวแปร \$b มีค่าเท่ากับ true เมื่อ \$b = (true) && (true)

บรรทัดที่ 4 กำหนดให้ตัวแปร \$c มีค่าเท่ากับ false เมื่อ \$c = (false) && (true)

บรรทัดที่ 5 กำหนดให้กับตัวแปร \$d มีค่าเท่ากับ true เมื่อ \$d = (false) || (true)

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

7) ตัวดำเนินการอื่นๆ

ในภาษา PHP ยังมีตัวดำเนินการอื่นๆ ที่ไม่ถูกจัดให้อยู่ในประเภทใดเลย ประกอบด้วย

- 1) ตัวดำเนินการใช้เงื่อนไขในการกำหนดค่า (Ternary Operator) 2) ตัวดำเนินการรับการเกิดข้อผิดพลาด (Error suppression Operator) และ 3) ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ (Execute Command Operator) และตัวดำเนินการในการเลือกค่า มีรายละเอียด ดังนี้

7.1) ตัวดำเนินการใช้เงื่อนไขในการกำหนดค่า

ตัวดำเนินการใช้เงื่อนไขในการกำหนดค่า คือ ตัวดำเนินการในการเปรียบเทียบเงื่อนไข ถ้าตรงกับเงื่อนไขผลลัพธ์จะเป็นอย่างไร และกรณีที่ไม่ตรงตามเงื่อนไขผลลัพธ์จะเป็นอย่างไร รูปแบบการใช้งาน ดังนี้

รูปแบบ

เงื่อนไข? $x : y.$

เมื่อ	เงื่อนไข	หมายถึง	เงื่อนไขสำหรับการกำหนดค่า
	x	หมายถึง	หากเงื่อนไขเป็นจริงจะมีค่าเท่ากับค่าที่กำหนดใน x
	y	หมายถึง	หากเงื่อนไขเป็นเท็จจะมีค่าเท่ากับค่าที่กำหนดใน y

ตัวอย่างที่ 3.26 ตัวดำเนินการใช้เงื่อนไขในการกำหนดค่า

```

1 <?php
2     $a = (1 > 0) ? true : false;
3     $b = (10 % 2 == 0) ? "เลขคู่" : "เลขคี่";
4
    ?>
```

จากตัวอย่างที่ 3.26 ตัวอย่างการใช้ตัวดำเนินการใช้เงื่อนไขในการกำหนดค่า อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร $\$a$ มีค่าเท่ากับ `true` เพราะ $(1 > 0)$ เป็นจริง

บรรทัดที่ 3 กำหนดให้ตัวแปร $\$b$ มีค่าเท่ากับ "เลขคู่" เมื่อ นำ 2 ไปหารเศษของผลหารเลข 10 มีค่าเท่ากับ 0

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

7.2) ตัวดำเนินการระงับการเกิดข้อผิดพลาด

ตัวดำเนินการระงับการเกิดข้อผิดพลาด แทนด้วยสัญลักษณ์ @ นำหน้าประโยค คำสั่ง หรือส่วนอื่นๆ ที่ต้องการเพื่อให้โปรแกรมมองข้ามข้อผิดพลาด (error) ที่อาจจะเกิดขึ้น มีตัวอย่าง การใช้งานตัวดำเนินการ ดังนี้

ตัวอย่างที่ 3.27 ตัวอย่างการใช้ตัวดำเนินการระงับการเกิดข้อผิดพลาด

```

1 <?PHP
2     @include ("file.php");
3     echo "Suratthani Rajabhat University";
4
    ?>
```

จากตัวอย่างที่ 3.27 ตัวอย่างการใช้ตัวดำเนินการระงับการเกิดข้อผิดพลาด อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP



บรรทัดที่ 2 นำเข้าไฟล์ file.php โดยใช้ร่วมกับตัวดำเนินการระงับการเกิดข้อผิดพลาด ในกรณีที่ไม่มีไฟล์ file.php ที่จะนำเข้าจะแสดงข้อความดังนี้

```
Warning: include(file.php) [function.include]: failed to open stream: No such file or directory in C:\AppServ\www\temp\test2.php on line 2
```

```
Warning: include() [function.include]: Failed opening 'file.php' for inclusion (include_path='.;C:\php5\pear') in C:\AppServ\www\temp\test2.php on line 2
```

Suratthani Rajabhat University

จากข้อความด้านบนระบบจะแจ้งข้อความเตือนเมื่อเพิ่งไฟล์ file.php เพื่อนำเข้ามาประมวลผลร่วม ในกรณีใช้ร่วมกับตัวดำเนินการระงับการเกิดข้อผิดพลาด จะแสดงผลตามปกติในบรรทัดที่ 3 โดยไม่ปรากฏข้อผิดพลาด

บรรทัดที่ 3 แสดงข้อความ Suratthani Rajabhat University

บรรทัดที่ 4 สิ้นสุดscriptr PHP

7.3) ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ

ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ โดยใช้สัญลักษณ์ ` (Grave Accent) แล้วตามด้วยคำสั่งภาษาของระบบปฏิบัติการที่ต้องการใช้งาน (ทั้งนี้ขึ้นอยู่กับระบบปฏิบัติการที่ประมวลผลภาษา PHP) ตัวอย่างดังนี้

ตัวอย่างที่ 3.28 ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ MS-Windows

```
1 <?php  
2     $listing = `dir /a c:';  
3     echo $listing;  
4 ?>
```

จากตัวอย่างที่ 3.28 ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ MS-Windows เพื่อเรียกดูไฟล์ข้อมูล ผลลัพธ์ที่ได้อาจแตกต่างกันไปขึ้นอยู่กับโครงสร้างไฟล์และไดเรกทอรีภายในเครื่องนั้นๆ

ตัวอย่างที่ 3.29 ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ Unix เรียกดูไฟล์ข้อมูล

```
1 <?php  
2     $listing = `ls -al`;  
3     echo $listing;  
4 ?>
```

จากตัวอย่างที่ 3.29 ตัวดำเนินการเรียกใช้คำสั่งของระบบปฏิบัติการ Unix เพื่อเรียกดูไฟล์ข้อมูล ซึ่งผลลัพธ์จะคล้ายๆ กับตัวอย่างที่ 3.28 ทั้งนี้สามารถประยุกต์ใช้งานได้กับทุกระบบปฏิบัติการ

สรุป

ตัวแปรในภาษา PHP ไม่จำเป็นต้องกำหนดชนิดให้กับตัวแปรเนื่องจากชนิดของตัวแปรสามารถปรับเปลี่ยนได้โดยอัตโนมัติขึ้นอยู่กับการกำหนดค่าให้กับตัวแปร การกำหนดค่าให้กับตัวแปรสามารถกำหนดค่าได้หลายรูปแบบ เช่น กำหนดค่าเริ่มต้น กำหนดค่าคงที่ กำหนดจากการคำนวณ กำหนดจากผลการทำงานของฟังก์ชัน เป็นต้น ซึ่งการเขียนสคริปต์ PHP ส่วนใหญ่จะนิยมให้ระบบพัฒนามีความยืดหยุ่น ดังนั้นการกำหนดค่าให้กับตัวแปรมักจะมาจากการคำนวณ เพื่อให้ได้ผลลัพธ์ที่หลากหลาย ใช้กำหนดทางเลือกในการแสดงผลที่แตกต่างกันไปตามค่าที่ได้จากการคำนวณ สำหรับการคำนวณนั้น จำเป็นต้องทราบถึงสัญลักษณ์ของเครื่องหมายดำเนินการ ลำดับความสำคัญ และหน้าที่ แล้วนำมาดำเนินการระหว่างตัวแปรกับตัวแปร หรือ ตัวแปรกับค่าใดๆ แล้วนำผลลัพธ์ที่ได้จากการดำเนินการกำหนดค่าให้กับตัวแปร เพื่อนำตัวแปรนั้นๆ ไปประมวลผลในส่วนอื่นๆ ของระบบต่อไป

คำถามท้ายบท

1. จงอธิบายข้อกำหนดสำหรับประกาศใช้ตัวแปร พิรุณยกตัวอย่างประกอบ
2. จงอธิบายวิธีการกำหนดให้กับตัวแปรที่สำคัญ มีกี่วิธี ประกอบด้วยอะไรบ้าง พิรุณยกตัวอย่างประกอบในแต่ละวิธี
3. จงอธิบายขอบเขตของตัวแปร มีกี่ชนิด ประกอบด้วยชนิดอะไรบ้าง พิรุณยกตัวอย่างประกอบในแต่ละชนิด
4. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการสคริปต์

```

1 <?php
2     function number ($value=10) {
3         $value=$value*2;
4         return $value;
5     }
6     $num = 5;
7     $num = number ($num);
8     echo $num;
9 ?>
```



5. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากสคริปต์

```
1 <?php  
2     $a = !(1 == "1");  
3     $b = (1 != 2) && (1>=0);  
4     $c = (1 <= 2) && (1>=0);  
5     $d = (1 === "1") || (1>0);  
6 ?>
```

บทที่ 4

โครงสร้างควบคุม

โครงสร้างควบคุม (Control Structures) คือ โครงสร้างที่ใช้สำหรับควบคุมการให้ผลของสคริปต์ คำสั่ง หรือกำหนดทางเลือกให้กระทำหรือไม่กระทำ เช่น การตรวจสอบเงื่อนไข การทำคำสั่งซ้ำๆ ตามเงื่อนไขกำหนด เป็นต้น ลักษณะการทำงานดังกล่าวจะถือเป็นพื้นฐานที่สำคัญที่ผู้เขียนโปรแกรมจำเป็นต้องเรียนรู้ ดังนั้นเนื้อหาในบทนี้จะได้เรียนรู้เกี่ยวกับโครงสร้างการควบคุมดังกล่าวที่สามารถใช้ได้ในภาษา PHP

โครงสร้างควบคุม สามารถแบ่งออกเป็น 3 กลุ่ม ประกอบด้วย 1) โครงสร้างเงื่อนไข (Conditional Structures) 2) โครงสร้างเงื่อนไขทำซ้ำ (Loop Structures) และ 3) คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ มีรายละเอียด ดังนี้

หมายเหตุ

รูปแบบโครงสร้างควบคุมของ if, while, for, foreach และ switch สามารถเปลี่ยนจากเครื่องหมายปีกกาเปิด (bracket) แทนด้วยเครื่องหมาย colon (:) และแทนปีกกาปิดด้วย endif;, endwhile;, endfor; endforeach; และ endswitch; ตามโครงสร้างควบคุมที่ใช้งาน

4.1 โครงสร้างเงื่อนไข

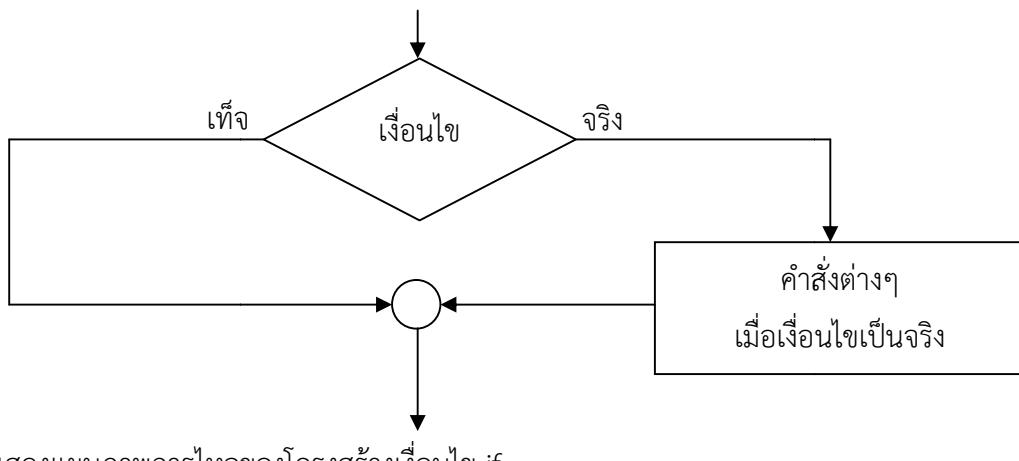
โครงสร้างเงื่อนไขเป็นหนึ่งในโครงสร้างควบคุมขั้นพื้นฐาน เพื่อกำหนดทางเลือกต่างๆ เพื่อให้โปรแกรมที่พัฒนาสามารถตอบสนองได้หลากหลายตามค่าที่รับเข้ามาหรือเงื่อนไขกำหนด โดยใช้รูปแบบการเปรียบเทียบทางตรรกะ โครงสร้างพื้นฐานของกลุ่มนี้ ประกอบด้วย 1) if 2) if...else 3) if ... elseif และ 4) switch มีรายละเอียด ดังนี้

4.1.1 โครงสร้างเงื่อนไข if (The if Statement)

โครงสร้างเงื่อนไข if ใช้ในการเปรียบเทียบเงื่อนไขที่ใช้สำหรับตัดสินใจ ถ้าเป็นจริงจะทำตามคำสั่งต่างๆ ที่กำหนดไว้ภายในได้เงื่อนไข ถ้าเป็นเท็จก็จะทำคำสั่งอื่นๆ หลังจากประโภคปิดโครงสร้างเงื่อนไข if โดยโครงสร้างเงื่อนไข if มีรูปแบบ ดังนี้

รูปแบบ

```
if (เงื่อนไข) {  
    คำสั่งเมื่อเงื่อนไขเป็นจริง  
}
```



ภาพที่ 4.1 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if

ตัวอย่างที่ 4.1 การใช้โครงสร้างเงื่อนไข if ในการเปรียบเทียบ

```

1 <?php
2     $a = 10;
3     $b = 5;
4     if ($a > $b) {
5         echo "a is bigger than b";
6     }
7 ?>
  
```

จากตัวอย่างที่ 4.1 การใช้โครงสร้างเงื่อนไข if ในการเปรียบเทียบ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$a มีค่าเท่ากับ 10

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$b มีค่าเท่ากับ 5

บรรทัดที่ 4 โครงสร้างเงื่อนไข if เปรียบเทียบค่าตัวแปร \$a มากกว่า ตัวแปร \$b จริงหรือไม่ ถ้าเป็นจริงเข้าสู่การทำงานภายในโครงสร้างเงื่อนไข if คือ ทำคำสั่งในบรรทัดที่ 5 ถ้าเป็นเท็จออกจากโครงสร้างเงื่อนไข if ไปทำการคำสั่งบรรทัดที่ 7

บรรทัดที่ 5 แสดงข้อความ a is bigger than b

บรรทัดที่ 6 ประযุคปิดโครงสร้างเงื่อนไข if

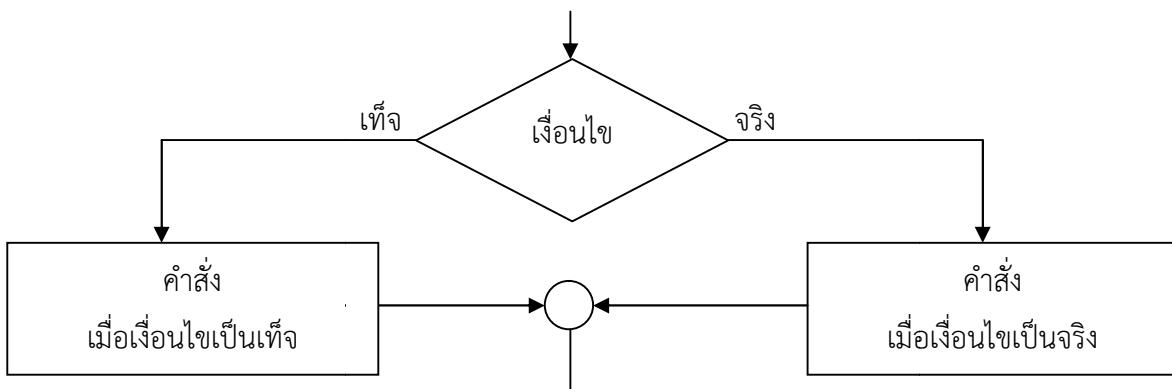
บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

4.1.2 โครงสร้างเงื่อนไข if ... else (The if ... else Statement)

โครงสร้างเงื่อนไข if...else เป็นโครงสร้างเงื่อนไขที่ใช้เปรียบเทียบเช่นเดียวกับ โครงสร้างเงื่อนไข if แต่มากกว่าตรงที่ โครงสร้างเงื่อนไข if .. else ถ้าเปรียบเทียบแล้วเป็นจริงก็จะให้ทำการเงื่อนไขภายในโครงสร้างเงื่อนไข if ถ้าเป็นเท็จก็จะทำการคำสั่งภายในโครงสร้างเงื่อนไข else มีรูปแบบ ดังนี้

รูปแบบ

```
if (เงื่อนไข) {
    คำสั่งเมื่อเงื่อนไขเป็นจริง
} else {
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ
}
```

**ภาพที่ 4.2** แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if ... else

จากโครงสร้างเงื่อนไข if ก่อนหน้านี้จะเห็นว่า เมื่อเปรียบเทียบค่าตัวแปร \$a มากกว่าค่าตัวแปร \$b จริงก็จะทำงานคำสั่งที่อยู่ภายใต้โครงสร้างเงื่อนไข if แต่หากค่าตัวแปร \$a น้อยกว่า หรือน้อยกว่าหรือเท่ากับ โปรแกรมจะไม่แสดงข้อความใดๆ หากประยุกต์ใช้โครงสร้างเงื่อนไข if ... else เพื่อแสดงข้อความเพิ่มเติม เช่น กรณีที่เป็นเท็จจากเงื่อนไข ในโครงสร้างเงื่อนไข if ให้แสดงข้อความอื่นๆ ตัวอย่างดังนี้

ตัวอย่างที่ 4.2 การใช้โครงสร้างเงื่อนไข if...else ใน การเปรียบเทียบ

```
1 <?php
2     $a = 10;
3     $b = 5;
4     if ($a > $b) {
5         echo "a is bigger than b";
6     } else {
7         echo "a is less than or equal b";
8     }
9 ?>
```

จากตัวอย่างที่ 4.2 การใช้โครงสร้างเงื่อนไข if....else ใน การเปรียบเทียบ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP



ปริญญา น้อยดอนไพร || การเขียนโปรแกรมบนเว็บด้วย PHP ร่วมกับฐานข้อมูล MySQL

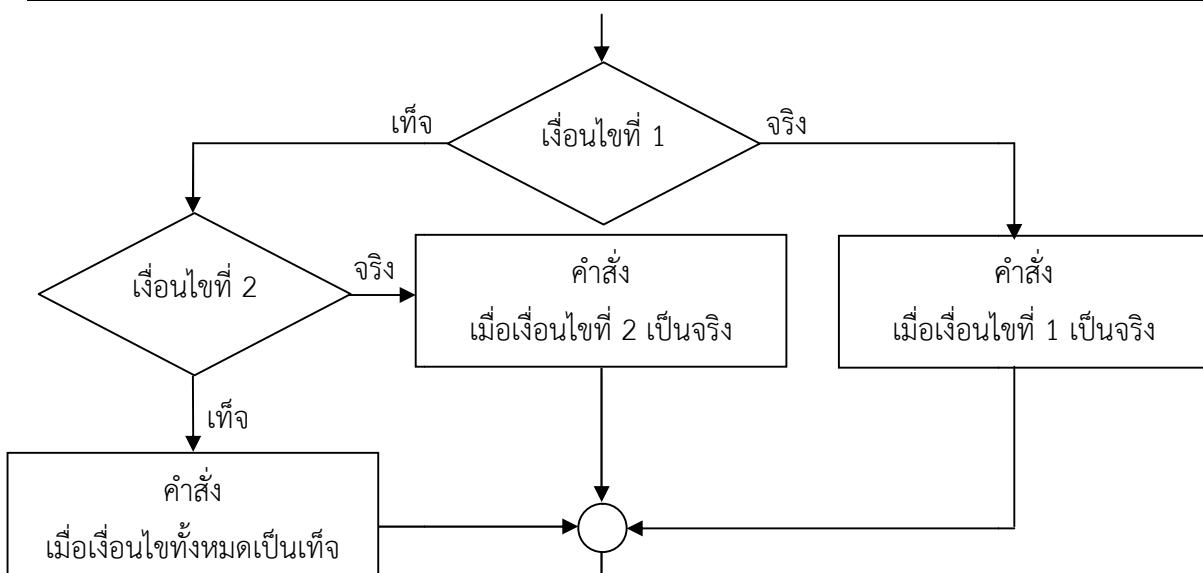
บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$a มีค่าเท่ากับ 10
 บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$b มีค่าเท่ากับ 5
 บรรทัดที่ 4 โครงสร้างเงื่อนไข if เปรียบเทียบค่าตัวแปร \$a มากกว่า ตัวแปร \$b จริง หรือไม่ ถ้าเป็นจริงเข้าสู่การทำงานภายใต้โครงสร้างเงื่อนไข if ไปทำการสั่งบรรทัดที่ 5 ถ้าเป็นเท็จจะออกจากประโยคปิดโครงสร้างเงื่อนไข if ในบรรทัดที่ 6 และเข้าสู่โครงสร้างเงื่อนไข else

- บรรทัดที่ 5 แสดงข้อความ a is bigger than b
- บรรทัดที่ 6 ประโยคปิดโครงสร้างเงื่อนไข if และเริ่มต้นโครงสร้างเงื่อนไข else
- บรรทัดที่ 7 แสดงข้อความ a is less than or equal b
- บรรทัดที่ 8 ประโยคปิดโครงสร้างเงื่อนไข else
- บรรทัดที่ 9 สิ้นสุด scrip ของ PHP

4.1.3 โครงสร้างเงื่อนไข if ... elseif (The if ... elseif Statement)

โครงสร้างเงื่อนไข if ... elseif หรือ if ... else if เป็นโครงสร้างเงื่อนไขที่ใช้เปรียบเทียบเงื่อนไขมากกว่า 1 เงื่อนไข เป็นการรวมกันของโครงสร้างเงื่อนไข if และ else ซ้อนกันเรียงตามลำดับ มีรูปแบบ ดังนี้

```
if (เงื่อนไขที่ 1) {
    คำสั่งเมื่อเงื่อนไขเป็นจริง
} elseif (เงื่อนไขที่ 2) {           // หรือเว้นวรรคเป็น else if (เงื่อนไขที่ 2)
    คำสั่งเมื่อเงื่อนไขที่ 2 เป็นจริง
} else {
    คำสั่งเมื่อเงื่อนไขที่ 2 เป็นเท็จ
}
```



ภาพที่ 4.3 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไข if ... elseif

ตัวอย่างที่ 4.3 การใช้โครงสร้างเงื่อนไข if...elseif ในการเปรียบเทียบหลายเงื่อนไข

```

1 <?php
2     $score = 78;
3     if ($score>=80) {
4         $grade="A";
5     } else if ($score>=70) {
6         $grade="B";
7     } else if ($score>=60) {
8         $grade="C";
9     } else if ($score>=50) {
10        $grade="D";
11    } else {
12        $grade="E";
13    }
14    printf ("Score is %d: Grade evaluation is %s.", $score, $grade);
15 ?>

```

ผลลัพธ์

Score is 78: Grade evaluation is B.

จากตัวอย่างที่ 4.3 การใช้โครงสร้างเงื่อนไข if...elseif ในการเปรียบเทียบหลายเงื่อนไข อธิบายดังนี้

- บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP
- บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$score มีค่าเท่ากับ 78
- บรรทัดที่ 3 โครงสร้างเงื่อนไข if (เงื่อนไขที่ 1) เปรียบเทียบค่าตัวแปร \$score มากกว่า หรือเท่ากับ 80 จริงหรือไม่ ถ้าเป็นจริงทำการสั่งในบรรทัดที่ 4 ถ้าเป็นเท็จไปบรรทัดที่ 5
- บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "A" และไปบรรทัดที่ 14
- บรรทัดที่ 5 ปิดประโยคโครงสร้างเงื่อนไข if (เงื่อนไขที่ 1) และเริ่มโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 2) เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 70 จริงหรือไม่ ถ้าเป็นจริงทำการสั่งบรรทัดที่ 6 ถ้าเป็นเท็จบรรทัดที่ 7
- บรรทัดที่ 6 กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "B" และไปทำการสั่งบรรทัดที่ 14
- บรรทัดที่ 7 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 2) และเริ่มโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 3) เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 60 จริงหรือไม่ ถ้าเป็นจริงไปทำการสั่งบรรทัดที่ 8 ถ้าเป็นเท็จไปทำการสั่งบรรทัดที่ 9



บรรทัดที่ 8 กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "C" และไปบรรทัดที่ 14

บรรทัดที่ 9 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 3) และเริ่มโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 4) เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 50 จริงหรือไม่ ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 10 ถ้าเป็นเท็จไปทำคำสั่งบรรทัดที่ 11

บรรทัดที่ 10 กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "D" และไปบรรทัดที่ 14

บรรทัดที่ 11 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 4) และเริ่มโครงสร้างเงื่อนไข else หมายความว่าเป็นเท็จจากทุกกรณีก่อนหน้าจะไปทำคำสั่งบรรทัดที่ 12

บรรทัดที่ 12 กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "E" และไปบรรทัดที่ 14

บรรทัดที่ 13 ปิดประโยคโครงสร้างเงื่อนไข else if (เงื่อนไขที่ 4)

บรรทัดที่ 14 แสดงข้อความ Score is 78: Grade evaluation is B.

บรรทัดที่ 15 สิ้นสุดสคริปต์ PHP

4.1.4 โครงสร้างเงื่อนไข switch (The switch Statement)

โครงสร้างเงื่อนไข switch เป็นโครงสร้างเงื่อนไขหนึ่งในโครงสร้างควบคุมขั้นพื้นฐาน เป็นการทดสอบของ if ... else โดยโครงสร้างเงื่อนไข switch นิยมใช้สำหรับการเปรียบเทียบตัวแปรตัวเดียวแต่มีค่าของตัวแปรจำนวนมาก เพื่อกำหนดทางเลือกตามค่าของตัวแปร อาจจะถูกกำหนดเป็นค่าใดค่าหนึ่งจากตัวเลือกที่กำหนด

โครงสร้างเงื่อนไข switch เป็นโครงสร้างเงื่อนไขในการเปรียบเทียบค่าตัวแปรในกรณีต่างๆ ภายในโครงสร้างเงื่อนไข switch ถ้าตัวแปรนั้นๆ ตรงกับ case ในที่จะทำงานภายในโครงสร้างของกรณีนั้นๆ และจะหยุดการทำงานคำสั่งในแต่ละกรณี ด้วยคำสั่ง break (หากไม่มีคำสั่ง break ในแต่ละกรณี โปรแกรมจะทำการเปรียบเทียบเงื่อนไขลำดับถัดๆ ไป ทั้งหมด) แต่ถ้าค่าตัวแปรไม่ตรงกับ case ใดๆ เลย จะเข้าสู่การทำงานในส่วนของ case "default" มีรูปแบบ ดังนี้

รูปแบบ

```
switch (ตัวแปร) {  
    case ค่าที่ 1 : ประโยคคำสั่งที่ 1; break;  
    case ค่าที่ 2 : ประโยคคำสั่งที่ 2; break;  
    ...  
    case ค่าที่ N : ประโยคคำสั่งที่ N; break;  
    default: ประโยคคำสั่ง //ในกรณีที่ไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้า  
}
```

ตัวอย่างที่ 4.4 การใช้โครงสร้างเงื่อนไข switch สำหรับการเปรียบเทียบค่าตัวแปรจำนวนมาก

1 <?php

2 \$i=2;

```

3     switch ($i) {
4         case 0: echo "i equals 0"; break;
5         case 1: echo "i equals 1"; break;
6         case 2: echo "i equals 2"; break;
7         default: echo "i is not equal to 0, 1 or 2";
8     }
9 ?>

```

จากตัวอย่างที่ 4.4 การใช้โครงสร้างเงื่อนไข switch สำหรับการเปรียบเทียบค่าตัวแปรจำนวนมาก อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$i มีค่าเท่ากับ 2

บรรทัดที่ 3 ตรวจสอบค่าตัวแปรโดยใช้โครงสร้างเงื่อนไข switch เปรียบเทียบค่าตัวแปร \$i เพื่อกำหนดทางเลือกตามค่าของตัวแปร

บรรทัดที่ 4 เปรียบเทียบค่าของตัวแปร \$i มีค่าเท่ากับ 0 จริงหรือไม่ ถ้าจริงแสดงคำว่า "i equals 0" แล้วออกจากโครงสร้างเงื่อนไข switch บรรทัดที่ 9 ถ้าเป็นเท็จไปบรรทัดที่ 5

บรรทัดที่ 5 เปรียบเทียบค่าของตัวแปร \$i มีค่าเท่ากับ 1 จริงหรือไม่ ถ้าเป็นจริงแสดงคำว่า "i equals 1" แล้วออกจากโครงสร้างเงื่อนไข switch บรรทัดที่ 9 ถ้าเป็นเท็จไปบรรทัดที่ 6

บรรทัดที่ 6 เปรียบเทียบค่าของตัวแปร \$i มีค่าเท่ากับ 2 จริงหรือไม่ ถ้าเป็นจริงแสดงคำว่า "i equals 2" แล้วออกจากโครงสร้างเงื่อนไข switch บรรทัดที่ 9 ถ้าเป็นเท็จไปบรรทัดที่ 7

บรรทัดที่ 7 ในกรณีที่เป็นเท็จจากทุกรูปนี้ข้างต้น ให้แสดงคำว่า "i is not equal to 0, 1 or 2"

บรรทัดที่ 8 ประโภคปิดโครงสร้างเงื่อนไข switch

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.5 การใช้โครงสร้างเงื่อนไข switch ประยุกต์จากตัวอย่างที่ 4.3

```

1 <?php
2     $score=78;
3     switch ($score) {
4         case ($score>=80): $grade="A"; break;
5         case ($score>=70): $grade="B"; break;
6         case ($score>=60): $grade="C"; break;
7         case ($score>=50): $grade="D"; break;
8         default: $grade="E";

```



```

9          }
10         printf ("Score is %d: Grade evaluation is %s.", $score, $grade);
11     ?>

```

ผลลัพธ์

Score is 78: Grade evaluation is B.

จากตัวอย่างที่ 4.5 การใช้โครงสร้างเงื่อนไข switch ในการเปรียบเทียบหลายเงื่อนไข อธิบายดังนี้

- บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP
- บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$score มีค่าเท่ากับ 78
- บรรทัดที่ 3 โครงสร้างเงื่อนไข switch เปรียบเทียบค่าตัวแปร \$score
- บรรทัดที่ 4 เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 80 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "A" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10
- บรรทัดที่ 5 เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 70 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "B" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10
- บรรทัดที่ 6 เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 60 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "C" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10
- บรรทัดที่ 7 เปรียบเทียบค่าตัวแปร \$score มากกว่าหรือเท่ากับ 50 ถ้าเป็นจริง กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "D" แล้วออกจากโครงสร้างเงื่อนไข if ไปทำคำสั่งบรรทัดที่ 10
- บรรทัดที่ 8 ในกรณีเป็นเท็จจากทุกรณีข้างต้น กำหนดค่าให้กับตัวแปร \$grade มีค่าเท่ากับ "E"
- บรรทัดที่ 9 ประโภคปิดโครงสร้างเงื่อนไข switch
- บรรทัดที่ 10 แสดงข้อความ Score is 78: Grade evaluation is B.
- บรรทัดที่ 11 สิ้นสุดสคริปต์ PHP

4.2 โครงสร้างเงื่อนไขทำซ้ำ

โครงสร้างเงื่อนไขทำซ้ำ เป็นหนึ่งในโครงสร้างควบคุม เพื่อควบคุมคำสั่งส่วนใดส่วนหนึ่งให้ทำงาน ซ้ำตามเงื่อนไขที่กำหนด เหตุผลสำคัญเพื่อลดจำนวนบรรทัดในการเขียนสคริปต์ และเพิ่มความคล่องตัวในการแก้ไข เมื่อต้องการปรับเปลี่ยนรูปแบบการแสดงผล เป็นต้น ตัวอย่างเช่น ต้องการแสดงผลเลข 1 ถึง 1000 หากเขียนสคริปต์โดยใช้คำสั่งแสดงผลปกติ อาจต้องใช้จำนวนบรรทัดมากกว่า 1000 บรรทัด เพื่อให้แสดงผลลัพธ์ตามต้องการ ทำให้เสียเวลาและขาดความคล่องตัว ดังนั้นจึงเป็นเหตุผลให้ใช้โครงสร้างเงื่อนไขทำซ้ำ เพื่อลดจำนวนบรรทัด เพิ่มความคล่องตัวในการเขียน และแก้ไขสคริปต์

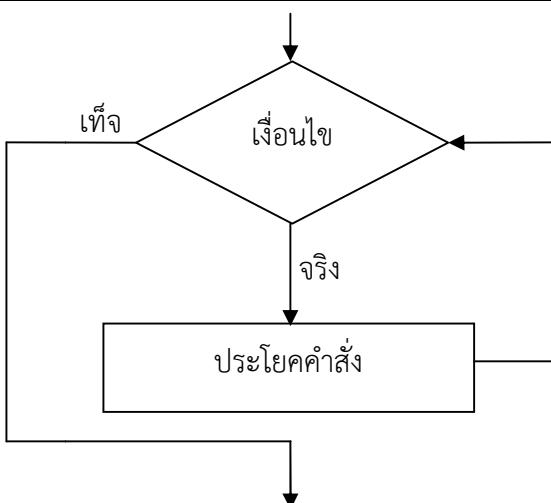
สำหรับในภาษา PHP มีโครงสร้างเงื่อนไขทำซ้ำที่สำคัญ ประกอบด้วย 1) while 2) do ... while 3) for และ 4) foreach มีรายละเอียดดังนี้

4.2.1 โครงสร้างเงื่อนไขทำซ้ำ while

โครงสร้างเงื่อนไขทำซ้ำ while เป็นหนึ่งในโครงสร้างเงื่อนไขทำซ้ำชนิดหนึ่ง หลักการทำงาน คือ ทำการตรวจสอบเงื่อนไขก่อนจะทำ ถ้าเงื่อนไขเป็นจริงจะเริ่มทำการสั่งภายในโครงสร้างเงื่อนไขทำซ้ำ และกลับมาตรวจสอบเงื่อนไขใหม่อีกรอบ หากยังคงเป็นจริงอยู่ก็จะทำซ้ำเช่นนี้ไปเรื่อยๆ จนกระทั่งเงื่อนไขเป็นเท็จจะออกจากโครงสร้างเงื่อนไขทำซ้ำได้ รูปแบบของโครงสร้างเงื่อนไขทำซ้ำ while มีดังนี้

รูปแบบ

```
while (เงื่อนไข) {
    ประযุคคำสั่ง
}
```



ภาพที่ 4.4 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ while

ตัวอย่างที่ 4.6 การใช้โครงสร้างเงื่อนไขทำซ้ำ while สำหรับแสดงตัวเลข 1 ถึง 5

```

1 <?php
2     $number = 1;
3     while ($number <= 5) {
4         echo $number
5         $number++;
6     }
7 ?>
```

ผลลัพธ์ คือ

```
1 2 3 4 5
```



จากตัวอย่างที่ 4.6 การใช้โครงสร้างเงื่อนไขทำซ้ำ while สำหรับแสดงตัวเลข 1 ถึง 5 อธิบายดังนี้

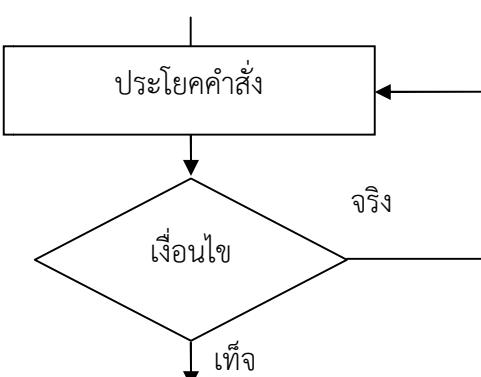
- บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP
- บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$number มีค่าเท่ากับ 1 (ตัวแปรควบคุม)
- บรรทัดที่ 3 เปรียบเทียบค่าตัวแปร \$number มีค่าน้อยกว่าหรือเท่ากับ 5 จริงหรือไม่
ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 4 – 5 ถ้าเป็นเท็จไปบรรทัดที่ 7
- บรรทัดที่ 4 แสดงค่าที่เก็บไว้ในตัวแปร \$number
- บรรทัดที่ 5 กำหนดให้ตัวแปร \$number เพิ่มค่าอีก 1 ค่า (เพิ่มค่าตัวแปรควบคุม)
- แล้วกลับไปตรวจสอบเงื่อนไขในบรรทัดที่ 3
- บรรทัดที่ 6 ประโภคปิดโครงสร้างเงื่อนไขทำซ้ำ while และกลับไปบรรทัดที่ 3
- บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

4.2.2 โครงสร้างเงื่อนไขทำซ้ำ do ... while

โครงสร้างเงื่อนไขทำซ้ำ do ... while เป็นหนึ่งในโครงสร้างเงื่อนไขทำซ้ำชนิดหนึ่ง หลักการทำงาน คือ จะเริ่มทำคำสั่งภายในโครงสร้างเงื่อนไขทำซ้ำ do ... while ก่อน 1 ครั้ง และวิ่ง ตรวจสอบเงื่อนไขภายในหลัง ถ้าเงื่อนไขเป็นจริง โปรแกรมจะกลับมาทำซ้ำเขียนนี้ไปเรื่อยๆ จนกระทั่งเงื่อนไข เป็นเท็จจึงจะสามารถจบการทำงานภายในโครงสร้างเงื่อนไขทำซ้ำ do ... while ได้ โครงสร้างเงื่อนไขทำซ้ำ do ... while มีรูปแบบ ดังนี้

รูปแบบ

```
do{
    ประโภคคำสั่ง
} while (เงื่อนไข);
```



ภาพที่ 4.5 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ do ... while

ตัวอย่างที่ 4.7 การใช้โครงสร้างเงื่อนไขทำซ้ำ do ... while สำหรับแสดงตัวเลข 1 ถึง 5

```

1 <?php
2     $number = 0;
3     do {
4         $number++;
5         echo $number;
6     } while ($number < 5);
7 ?>

```

ผลลัพธ์ คือ

1 2 3 4 5

จากตัวอย่างที่ 4.7 การใช้โครงสร้างเงื่อนไขทำซ้ำ do ... while สำหรับแสดงตัวเลข 1 ถึง 5 อธิบายดังนี้

- บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP
- บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$number มีค่าเท่ากับ 0 (ตัวแปรควบคุม)
- บรรทัดที่ 3 เริ่มต้นโครงสร้างเงื่อนไขทำซ้ำ do ... while
- บรรทัดที่ 4 กำหนดให้ตัวแปร \$number เพิ่มค่าอีก 1 ค่า (เพิ่มค่าตัวแปรควบคุม)
- บรรทัดที่ 5 แสดงค่าที่เก็บไว้ในตัวแปร \$number
- บรรทัดที่ 6 เปรียบเทียบค่าตัวแปร \$number มีค่าน้อยกว่า 5 จริงหรือไม่ ถ้าเป็นจริง กลับไปทำคำสั่งบรรทัดที่ 4 ถ้าเป็นเท็จไปบรรทัดที่ 7

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

4.2.3 โครงสร้างเงื่อนไขทำซ้ำ for

โครงสร้างเงื่อนไขทำซ้ำ for จะเป็นโครงสร้างที่ใช้ในการควบคุมการทำงานของโปรแกรม ประเภทโครงสร้างเงื่อนไขทำซ้ำ มีลักษณะคล้ายๆ กับโครงสร้างเงื่อนไขทำซ้ำ while ต่างที่โครงสร้างเงื่อนไขทำซ้ำ for สามารถกำหนดค่าเริ่มต้นตัวแปรเพื่อใช้เป็นตัวแปรควบคุมหรืออื่นๆ (กำหนดได้มากกว่า 1 ตัวแปร) ลักษณะรูปแบบที่สำคัญ แบ่งออกเป็น 3 ส่วน คือ 1) ส่วนกำหนดค่าเริ่มต้นตัวแปร 2) ส่วนของเงื่อนไข และ 3) ส่วนของการปรับเปลี่ยนค่าตัวแปร โดยทั้ง 3 ส่วน ถูกบรรจุอยู่ภายในโครงสร้างเงื่อนไขทำซ้ำ for เพียงบรรทัดเดียว มีรูปแบบดังนี้

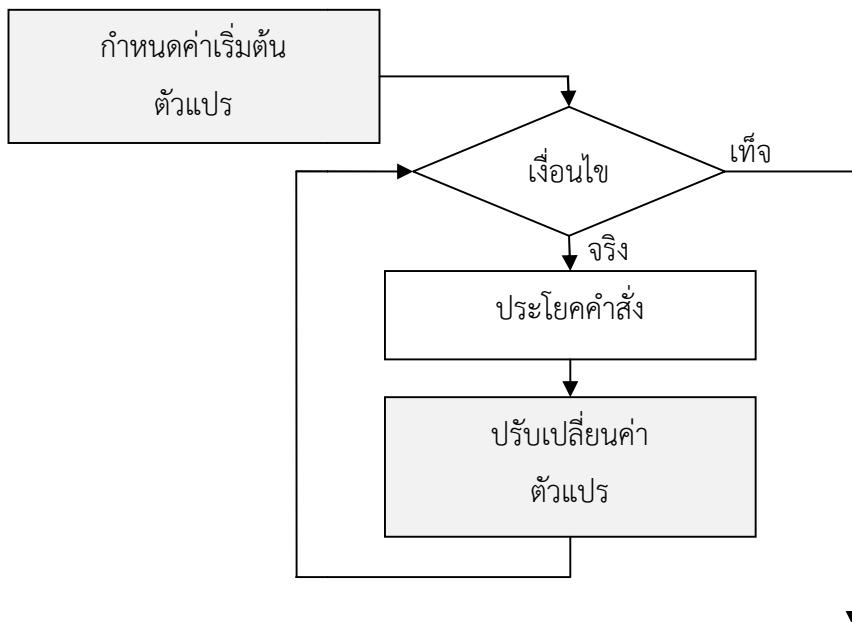
รูปแบบ

```
for (ค่าเริ่มต้นตัวแปร; เงื่อนไข; ปรับเปลี่ยนค่าตัวแปร) {
```

 ประโยคคำสั่ง;

}





ภาพที่ 4.6 แสดงแผนภาพการไหลของโครงสร้างเงื่อนไขทำซ้ำ for

หมายเหตุ

จากรูปแบบ for (ค่าเริ่มต้นตัวแปร; เงื่อนไข; ปรับเปลี่ยนค่าตัวแปร) ค่าในแต่ละตำแหน่งสามารถเว้นเป็นค่าว่างได้ และใช้ตรรศศาสตร์อื่นๆ ในประยุคคำสั่ง เพื่อควบคุมการทำงานของโครงสร้างเงื่อนไขทำซ้ำ for

ตัวอย่างที่ 4.8 ตัวอย่างการใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ

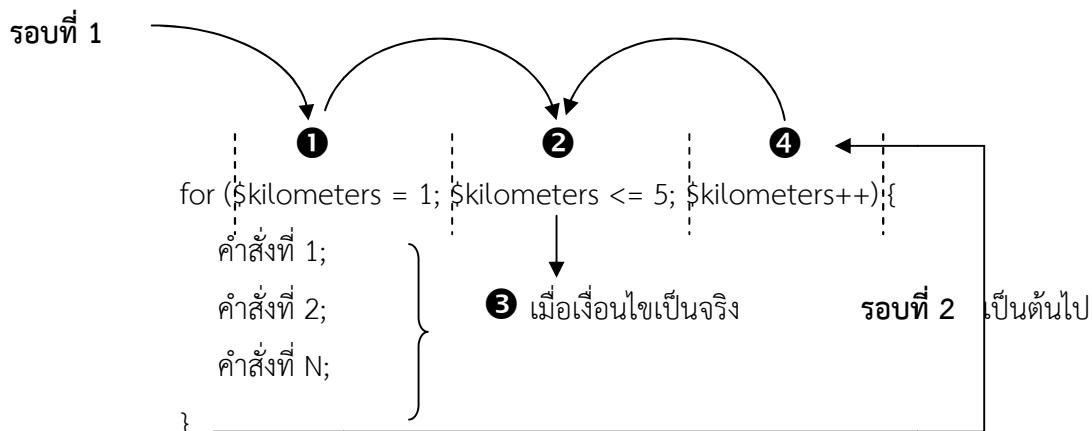
```

1 <?php
2     for ($kilometers = 1; $kilometers <= 5; $kilometers++) {
3         printf ("%d Kilometers = ", $kilometers);
4         printf("% .5f Miles <br>", $kilometers * 0.62140);
5     }
6 ?>
  
```

จากตัวอย่างที่ 4.8 การใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 อธิบายเพิ่มเติม ดังนี้



ภาพที่ 4.7 แสดงลักษณะการวนรอบของโครงสร้างเงื่อนไขทำซ้ำ for แบบปกติ

- ① หมายถึง ส่วนกำหนดค่าเริ่มต้นให้กับตัวแปร (กำหนดให้เพียงครั้งแรกเท่านั้น)
- ② หมายถึง ส่วนตรวจสอบเงื่อนไข ถ้าเป็นจริงไปทำคำสั่งบรรทัดที่ 3 ถ้าเป็นเท็จไป
 คำสั่งบรรทัดที่ 6
- ③ หมายถึง เมื่อเงื่อนไขเป็นจริงให้ทำคำสั่งอะไรบ้าง เรียงตามลำดับ
- ④ หมายถึง ส่วนปรับเปลี่ยนค่าให้กับตัวแปร

รอบที่ 1 จะเริ่มต้นที่ ① (กำหนดให้เพียงครั้งแรกเท่านั้น) โดยในรอบที่ 1 กำหนดให้ตัว
 แปร \$kilometers มีค่าเท่ากับ 1 แล้วตรวจสอบเงื่อนไข ใน ② ตัวแปร \$kilometers มีค่าน้อยกว่าหรือ
 เท่ากับ 5 จริงหรือไม่ ถ้าเป็นจริง ให้ไปทำคำสั่งในบรรทัดที่ 3 และอื่นๆ เรียงตามลำดับคำสั่ง ถ้าเป็นเท็จ
 ไปบรรทัดที่ 6 (ออกจากโครงสร้างเงื่อนไขทำซ้ำ for)

รอบที่ 2 เป็นต้นไป จะกลับมาที่ ④ เพื่อเพิ่มค่าให้กับตัวแปร \$kilometers ครั้งละ 1 ค่า
 แล้วตรวจสอบเงื่อนไข ใน ② ตัวแปร \$kilometers ยังคงมีค่าน้อยกว่าหรือเท่ากับ 5 จริงหรือไม่
 ถ้าเป็นจริง ให้ไปทำคำสั่งบรรทัดที่ 3 – 5 ถ้าเป็นเท็จ ไปบรรทัดที่ 6 (ออกจากโครงสร้างเงื่อนไขทำซ้ำ for)

บรรทัดที่ 3 แสดงผลค่าตัวแปร \$kilometers ในแต่ละรอบการทำซ้ำ และข้อความ
 คำว่า Kilometers =

บรรทัดที่ 4 แสดงผลการคำนวนค่าตัวแปร \$kilometers คูณด้วย 0.62140 ในแต่ละ
 รอบการทำซ้ำ ข้อความคำว่า Miles และขึ้นบรรทัดใหม่

บรรทัดที่ 5 ประยุคปิดโครงสร้างเงื่อนไขทำซ้ำ for และกลับไปส่วนที่ ④ ในบรรทัดที่ 2

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.9 ตัวอย่างการใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข

```

1 <?php
2     for ($kilometers = 1; ; $kilometers++) {
3         if ($kilometers > 5) break;
4         printf ("%d Kilometers = ", $kilometers);

```



```

5         printf("%.5f Miles <br>", $kilometers * 0.62140);
6     }
7 ?>

```

จากตัวอย่างที่ 4.9 การใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 อธิบายเพิ่มเติม ดังนี้

รอบที่ 1

1 for (\$kilometers = 1 ; ; \$kilometers++) {
2 ใช้โครงสร้างเงื่อนไข if กรณีเป็นจริงให้สิ้นสุดการทำซ้ำโดยใช้ break;

คำสั่งที่ 1;
 คำสั่งที่ 2;
 คำสั่งที่ N;

3 เมื่อเงื่อนไขเป็นเท็จ

รอบที่ 2 เป็นต้นไป

}

ภาพที่ 4.8 แสดงลักษณะการวนรอบของโครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดเงื่อนไข

1 หมายถึง ส่วนกำหนดค่าเริ่มต้นให้กับตัวแปร (กำหนดให้เพียงครั้งแรกเท่านั้น)

2 หมายถึง ส่วนตรวจสอบเงื่อนไข ถ้าเป็นจริงไปทำการคำสั่งบรรทัดที่ 4 ถ้าเป็นเท็จไปทำการคำสั่งบรรทัดที่ 7

3 หมายถึง ทำการคำสั่งเรียงตามลำดับ (อยู่ภายใต้โครงสร้างเงื่อนไขทำซ้ำ for)

4 หมายถึง ส่วนปรับเปลี่ยนค่าให้กับตัวแปร

รอบที่ 1 จะเริ่มต้นที่ **1** (กำหนดให้เพียงครั้งแรกเท่านั้น) โดยในรอบที่ 1 กำหนดให้ตัวแปร \$kilometers มีค่าเท่ากับ 1 และไปทำการคำสั่งในบรรทัดที่ 3

รอบที่ 2 เป็นต้นไป จะกลับมาที่ **4** เพื่อเพิ่มค่าให้กับตัวแปร \$kilometers ครั้งละ 1 ค่า และไปทำการคำสั่งในบรรทัดที่ 3

บรรทัดที่ 3 ตรวจสอบเงื่อนไขตัวแปร \$kilometers มีค่ามากกว่า 5 จริงหรือไม่ เมื่อตรวจสอบเงื่อนไขแล้ว ถ้าเป็นจริงให้หยุดและออกจากโครงสร้างเงื่อนไขทำซ้ำ for ไปบรรทัดที่ 7 ถ้าเป็นเท็จไปทำการคำสั่งบรรทัดที่ 4

บรรทัดที่ 4 แสดงผลค่าตัวแปร \$kilometers ในแต่ละรอบการทำซ้ำ และข้อความคำว่า Kilometers =

บรรทัดที่ 5 แสดงผลการคำนวณค่าตัวแปร \$kilometers คูณด้วย 0.62140 ในแต่ละรอบการทำซ้ำ ข้อความคำว่า Miles และขึ้นบรรทัดใหม่

บรรทัดที่ 6 ประยุกต์ปิดโครงสร้างเงื่อนไขทำซ้ำ for และกลับไปส่วนที่ ④ ในบรรทัดที่ 2

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.10 ตัวอย่างการใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดค่าใดๆ

```

1 <?php
2     $kilometers = 1;
3     for (; ;) {
4         if ($kilometers > 5) break;
5         printf ("%d Kilometers = ", $kilometers);
6         printf("% .5f Miles <br>", $kilometers * 0.62140);
7         $kilometers++
8     }
9 ?>
```

จากตัวอย่างที่ 4.10 การใช้โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดค่าใดๆ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$kilometers มีค่าเท่ากับ 1

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ for แบบไม่กำหนดค่าใดๆ (วนรอบโดยไม่มีเงื่อนไขการสิ้นสุด ลักษณะการทำงานดังกล่าวผู้เขียนโปรแกรมจะต้องระมัดระวัง เนื่องจากอาจทำให้โปรแกรมวนรอบไปเรื่อยๆ)

บรรทัดที่ 4 ตรวจสอบเงื่อนไขว่า ตัวแปร \$kilometers มีค่ามากกว่า 5 จริงหรือไม่ ถ้าเป็นจริง ให้หยุดและออกจากโครงสร้างเงื่อนไขทำซ้ำ for ไปบรรทัดที่ 9 ถ้าเป็นเท็จไปทำการคำสั่งบรรทัดที่ 5

บรรทัดที่ 5 แสดงผลค่าตัวแปร \$kilometers ในแต่ละรอบการทำซ้ำ และข้อความคำว่า Kilometers =

บรรทัดที่ 6 แสดงผลการคำนวณค่าตัวแปร \$kilometers คูณด้วย 0.62140 ในแต่ละรอบการทำซ้ำ ข้อความคำว่า Miles และขึ้นบรรทัดใหม่

บรรทัดที่ 7 เพิ่มค่าให้กับตัวแปร \$kilometers ครั้งละ 1 ค่า

บรรทัดที่ 8 ประยุกต์ปิดโครงสร้างเงื่อนไขทำซ้ำ for และกลับไปทำการคำสั่งบรรทัดที่ 4

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

ผลลัพธ์ทั้ง 3 ตัวอย่าง คือ

1 kilometers = 0.6214 miles

2 kilometers = 1.2428 miles



3 kilometers = 1.8642 miles

4 kilometers = 2.4856 miles

5 kilometers = 3.107 miles

4.2.4 โครงสร้างทำซ้ำ foreach

โครงสร้างทำซ้ำ foreach ใช้สำหรับการทำงานร่วมกับข้อมูลชนิดอาร์เรย์เท่านั้น (การจัดการกับข้อมูลชนิดอาร์เรย์ จะกล่าวถึงในบทต่อไป) ช่วยในการทำซ้ำหรือวนรอบดึงค่าข้อมูลอาร์เรย์ออกจากตัวแปรชนิดอาร์เรย์ตามลำดับ รูปแบบของโครงสร้างทำซ้ำ foreach มีรูปแบบการใช้งาน 2 รูปแบบ ดังนี้

รูปแบบที่ 1

```
foreach (array_expression as $value) {
    ประโยคคำสั่ง
}
```

เมื่อ array_expression หมายถึง ตัวแปรชนิดอาร์เรย์ที่ต้องการอ่านค่า

\$value หมายถึง ตัวแปรที่ใช้รับผลจากการอ่านค่าข้อมูลชนิดอาร์เรย์

รูปแบบที่ 2

```
foreach (array_expression as $key => $value)
    ประโยคคำสั่ง
?>
```

เมื่อ array_expression หมายถึง ตัวแปรชนิดอาร์เรย์ที่ต้องการอ่านค่า

\$key หมายถึง ลำดับคีย์ที่ต้องการอ่านค่า

\$value หมายถึง ตัวแปรที่ใช้รับผลจากการอ่านค่าข้อมูลชนิดอาร์เรย์

รูปแบบที่ 1 โครงสร้างทำซ้ำ foreach จะทำการวนรอบเพื่ออ่านค่าข้อมูลของสมาชิกในอาร์เรย์ โดยแต่ละรอบนั้น ค่าของสมาชิกปัจจุบันจะถูกกำหนดไปยังตัวแปร \$value และตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ (array pointer) จะเลื่อนไปยังสมาชิกลำดับถัดไปเพื่อรอการวนรอบใหม่ (ดังจะเห็นได้จากเมื่อเริ่มการวนรอบใหม่ ค่าของสมาชิกลำดับถัดไปจะถูกกำหนดไปยัง \$value แทน) โดยจะทำการวนรอบอย่างนี้ไปเรื่อยๆ จนกระทั่งครบจำนวนสมาชิกทั้งหมดในอาร์เรย์

รูปแบบที่ 2 โครงสร้างทำซ้ำ foreach จะทำงานคล้ายกับรูปแบบที่ 1 แต่ค่าที่ส่งนั้นจะประกอบด้วย คีย์ และค่า ของสมาชิกแต่ละตัวในอาร์เรย์ ซึ่งรูปแบบที่ 1 นั้นจะทำการส่งเฉพาะค่าของสมาชิกเพียงอย่างเดียว

เมื่อเริ่มการวนรอบในครั้งแรกโครงสร้างทำซ้ำ foreach จะกำหนดให้ตัวชี้ตำแหน่งสมาชิกในอาร์เรย์เริ่มต้นที่ไปยังตำแหน่งของสมาชิกลำดับแรกโดยอัตโนมัติ หมายความว่าไม่จำเป็นต้องใช้ฟังก์ชัน reset () ก่อนเข้าการวนรอบในโครงสร้างทำซ้ำ foreach และยังสามารถเปลี่ยนแปลงค่าสมาชิกในอาร์เรย์ได้

ตัวอย่างที่ 4.11 การใช้โครงสร้างทำซ้ำ foreach

```

1 <?php
2     $arr = array (1, 2, 3, 4);
3     foreach ($arr as $value) {
4         printf ("%d ", $value);
5     }
6 ?>

```

ผลลัพธ์

1 2 3 4

จากตัวอย่างที่ 4.11 ตัวอย่างการใช้โครงสร้างทำซ้ำ foreach นิยมใช้เพื่อแสดงผลข้อมูลในอาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$arr เป็นชนิดอาร์เรย์ ประกอบด้วยค่า 1, 2, 3

และ 4 ตามลำดับ

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ foreach

รอบที่ 1 จะกำหนดให้ตัวชี้ตำแหน่งอาร์เรย์ ชี้ไปยังตำแหน่งสมาชิกอาร์เรย์ ลำดับที่ 1 ในตัวแปร \$arr อ่านค่าข้อมูลจากสมาชิกอาร์เรย์ ลำดับที่ 1 และกำหนดค่าที่อ่านได้ไปยังตัวแปร \$value หลังจากนั้นไปทำการสั่งในบรรทัดที่ 4

รอบถัดไป ทำการเลื่อนตัวชี้ตำแหน่งอาร์เรย์ไปยังลำดับสมาชิกอาร์เรย์ ลำดับถัดไปเป็นตัวแปร \$arr อ่านค่าข้อมูลสมาชิกอาร์เรย์ และกำหนดค่าที่อ่านได้ไปยังตัวแปร \$value หลังจากนั้นไปทำการสั่งในบรรทัดที่ 4 หากไม่สามารถอ่านค่าข้อมูลได้ แสดงว่าข้อมูลของสมาชิกอาร์เรย์ในตัวแปร \$arr ได้ถูกอ่านครบแล้ว จะออกจากโครงสร้างทำซ้ำ foreach ไปบรรทัดที่ 6

บรรทัดที่ 4 แสดงค่าที่เก็บไว้ในตัวแปร \$value

บรรทัดที่ 5 ประโภคปิดโครงสร้างทำซ้ำ foreach และกลับไปทำการสั่งบรรทัดที่ 3

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

4.3 คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ

คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ คือ คำสั่งควบคุมที่ส่งผลต่อการทำงานของโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ เช่น ในบางครั้งของการเขียนโปรแกรมอาจจำเป็นต้องหยุดการทำงานของโครงสร้างเงื่อนไขทำซ้ำนั้นๆ อันเนื่องมาจากการตัวแปรบางอย่างอาจก่อให้เกิดข้อผิดพลาดจำเป็นต้องหยุดการทำงานก่อนปัญหาดังกล่าวจะเกิดขึ้น เป็นต้น คำสั่ง



ควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ ประกอบด้วย 1) คำสั่ง break และ 2) คำสั่ง continue มีรายละเอียด ดังนี้

4.3.1 คำสั่ง break (The break Statements)

คำสั่ง break เป็นคำสั่งที่สั่งให้หยุดการทำงานโครงสร้างเงื่อนไข และโครงสร้างเงื่อนไขทำซ้ำ ไม่ว่าจะเป็น for, foreach, while, do ... while หรือ switch ก็ตาม เมื่อโปรแกรมมาพบคำสั่ง break โปรแกรมจะหยุดประมวลผลคำสั่งทั้งหมดในโครงสร้างเงื่อนไขหรือโครงสร้างเงื่อนไขทำซ้ำ แล้วจะกระโดดออกจากโครงสร้างเงื่อนไขเดิมกล่าวทันที และทำการต่อในลำดับบรรทัดถัดไป (หากมี)

คำสั่ง break สามารถกำหนดพารามิเตอร์ โดยการระบุหมายเลขต่อท้ายคำสั่ง ประกอบด้วย 1 หมายถึง การสั่งให้ออกจากการทำงานโครงสร้างเงื่อนไข ชั้นที่ 1 (ค่าโดยปริยาย หากไม่ระบุจะมีค่าเท่ากับ 1) และ 2 หมายถึง การสั่งให้ออกจากโครงสร้างเงื่อนไข ชั้นที่ 1 และ 2 ตัวอย่างดังนี้

ตัวอย่างที่ 4.12 การใช้คำสั่ง break สำหรับออกจากการทำงานของโครงสร้างเงื่อนไขทำซ้ำ

```

1 <?php
2     $arr = array ("one", "two", "three", "four", "stop", "five");
3     foreach ($arr as $value) {
4         if ($value == "stop") break;
5         echo "$value ";
6     }
7 ?>
```

ผลลัพธ์

one two three four

จากตัวอย่างที่ 4.12 การใช้คำสั่ง break สำหรับออกจากการทำงานของโครงสร้างเงื่อนไขทำซ้ำ foreach เมื่อค่าของอาร์เรย์มีค่าเท่ากับ "stop" อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$arr เป็นชนิดอาร์เรย์ ประกอบด้วยค่า "one", "two", "three", "four", "stop" และ "five" ตามลำดับ

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ foreach

รอบที่ 1 จะกำหนดให้ตัวชี้ตำแหน่งอาร์เรย์ ชี้ไปยังตำแหน่งสมาชิกอาร์เรย์ ลำดับที่ 1 และกำหนดค่าที่อ่านได้ไปยังตัวแปร \$value หลังจากนั้นไปทำการคำสั่งในบรรทัดที่ 4

รอบถัดไป ทำการเลื่อนตัวชี้ตำแหน่งอาร์เรย์ไปยังลำดับสมาชิกอาร์เรย์ ลำดับถัดไปในตัวแปร \$arr อ่านค่าข้อมูลสมาชิกอาร์เรย์ และกำหนดค่าที่อ่านได้ไปยังตัวแปร \$value

หลังจากนั้นไปทำการสั่งในบรรทัดที่ 4 หากไม่สามารถอ่านค่าข้อมูลได้ แสดงว่าข้อมูลของสมาชิกการเรียนในตัวแปร \$arr ได้ถูกอ่านครบแล้ว จะออกจากโครงสร้างทำซ้ำ foreach ไปบรรทัดที่ 7

บรรทัดที่ 4 โครงสร้างเงื่อนไขตรวจสอบค่าที่เก็บในตัวแปร \$value หากมีค่าเท่ากับ "stop" ใช้คำสั่ง break เพื่ออกจากโครงสร้างเงื่อนไขทำซ้ำ foreach ไปยังบรรทัดที่ 7

บรรทัดที่ 5 แสดงค่าที่เก็บไว้ในตัวแปร \$value

บรรทัดที่ 6 ประযุคปิดโครงสร้างทำซ้ำ foreach และกลับไปทำการสั่งบรรทัดที่ 3

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 4.13 การใช้คำสั่ง break กำหนดพารามิเตอร์เพื่ออกจากโครงสร้างเงื่อนไขทำซ้ำ

```

1 <?php
2
3     $number = 1;
4
5     while ($number<=20) {
6
7         switch ($number) {
8
9             case 5: echo "Number = 5 <br>"; break 1;
10            case 10: echo "Number = 10 Quitting"; break 2;
11            default: break;
12
13        }
14
15        $number++;
16
17    }
18
19 ?>
```

ผลลัพธ์

```

Number = 5
Number = 10 Quitting
```

จากตัวอย่างที่ 4.13 การใช้คำสั่ง break กำหนดพารามิเตอร์เพื่ออกจากโครงสร้างเงื่อนไขทำซ้ำ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$number มีค่าเท่ากับ 1

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ while ตรวจสอบค่าตัวแปร \$number มีค่าน้อยกว่าหรือเท่ากับ 20 จริงหรือไม่ หากจริงไปทำการสั่งในบรรทัดที่ 4 หากเป็นเท็จไปบรรทัด 11

บรรทัดที่ 4 โครงสร้างเงื่อนไข switch ตรวจสอบค่าตัวแปร \$number

บรรทัดที่ 5 กรณีค่าของตัวแปร \$number มีค่าเท่ากับ 5 ให้แสดงข้อความ "Number = 5" หลังจากนั้นขึ้นบรรทัดใหม่
 และออกจากโครงสร้างเงื่อนไข switch ไปทำการสั่งบรรทัดที่ 9 แสดงค่าที่เก็บไว้ในตัวแปร \$value



บรรทัดที่ 6 กรณีค่าของตัวแปร \$number มีค่าเท่ากับ 10 ให้แสดงข้อความ "Number = 10 Quitting" แล้วออกจากโครงสร้างเงื่อนไขทำซ้ำ while ไปบรรทัดที่ 11

บรรทัดที่ 7 กรณีอื่นๆ ให้ออกจากโครงสร้างเงื่อนไข switch

บรรทัดที่ 8 ปิดประโยคโครงสร้างเงื่อนไข switch

บรรทัดที่ 9 กำหนดให้ตัวแปร \$number เพิ่มค่า 1 ค่า

บรรทัดที่ 10 ปิดประโยคโครงสร้างเงื่อนไขทำซ้ำ while

บรรทัดที่ 11 สิ้นสุดสคริปต์ PHP

4.3.2 คำสั่ง continue (The continue Statement)

คำสั่ง continue นิยมใช้ภายในโครงสร้างเงื่อนไขทำซ้ำ เพื่อให้ข้ามการวนรอบของรอบการทำงานโครงสร้างเงื่อนไขทำซ้ำในปัจจุบัน ไปทำงานในวนรอบถัดไป และคำสั่ง continue ยังสามารถกำหนดพารามิเตอร์ เพื่อข้ามการทำงานในวนรอบระดับ (level) ชั้นเดียวกันได้ ตัวอย่างดังนี้

ตัวอย่างที่ 4.14 การใช้คำสั่ง continue ข้ามการทำงานวนรอบของโครงสร้างเงื่อนไขทำซ้ำปัจจุบัน ไปทำงานในวนรอบถัดไป

```
1 <?php
2     $number = 0;
3     while ($number <= 20) {
4         $number++;
5         if ($number%2) continue;
6         echo "$number ";
7     }
8 ?>
```

ผลลัพธ์

```
2 4 6 8 10 12 14 16 18 20
```

จากตัวอย่างที่ 4.14 การใช้คำสั่ง continue ข้ามการทำงานวนรอบของโครงสร้างเงื่อนไขทำซ้ำปัจจุบัน ไปทำงานในวนรอบถัดไป อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$number มีค่าเท่ากับ 0

บรรทัดที่ 3 โครงสร้างเงื่อนไขทำซ้ำ while ตรวจสอบค่าตัวแปร \$number มีค่าน้อยกว่าหรือเท่ากับ 20 จริงหรือไม่ หากจริงไปทำคำสั่งในบรรทัดที่ 4 หากเป็นเท็จไปบรรทัด 8

บรรทัดที่ 4 เพิ่มค่าให้กับตัวแปร \$number 1 ค่า

บรรทัดที่ 5 โครงสร้างเงื่อนไข if ตรวจสอบผลการคำนวณของนิพจน์เพื่อหารหาผลเศษของตัวแปร \$number ด้วย 2 ถ้าผลหารมีค่าเป็น 1 (จริง) คือ หารไม่ลงตัว ให้ทำการสั่ง continue คือขั้นตอนการทำงานวนรอบปัจจุบัน กลับไปตรวจสอบเงื่อนไขบรรทัดที่ 3

บรรทัดที่ 6 แสดงค่าตัวแปร \$number (ผลลัพธ์ที่ได้ คือ แสดงผลเลขคู่)

บรรทัดที่ 7 ปิดประโยคโครงสร้างเงื่อนไขทำซ้ำ while

บรรทัดที่ 8 สิ้นสุดสคริปต์ PHP

สรุป

โครงสร้างควบคุมเป็นส่วนที่สำคัญมากสำหรับการควบคุมการให้ผลของสคริปต์คำสั่ง การกำหนดทางเลือก ตรวจสอบเงื่อนไข การทำคำสั่งซ้ำๆ ตามเงื่อนไขกำหนด ในภาษา PHP มีโครงสร้างควบคุมที่สำคัญแบ่งออกเป็น 3 กลุ่ม ประกอบด้วย 1) โครงสร้างเงื่อนไข 2) โครงสร้างเงื่อนไขทำซ้ำ และ 3) คำสั่งควบคุมอื่นๆ ที่เกี่ยวข้องกับโครงสร้างเงื่อนไข โดยก่อนเริ่มต้นการใช้โครงสร้างควบคุมนั้น จำเป็นต้องออกแบบและวิเคราะห์ทำความต้องการก่อน หรืออาจเขียนเป็นแผนภาพการให้ผลของสคริปต์ คำสั่งเพื่อจำลองลำดับเหตุการณ์ที่ต้องการให้ภาษา PHP ประมวลผล แล้วหลังจากนั้นพิจารณาเลือกใช้โครงสร้างควบคุมที่เหมาะสม จุดสำคัญของโครงสร้างควบคุมก็คือการเปรียบเทียบเงื่อนไขโดยใช้งานร่วมกับตัวดำเนินการเปรียบเทียบทางตระกูล

คำถามท้ายบท

1. โครงสร้างควบคุมแบ่งได้เป็นกี่กลุ่ม และประกอบด้วยกลุ่มอะไรบ้าง
2. จะอธิบายถึงรูปแบบโครงสร้างเงื่อนไข if และวัดแผนภาพการให้ผลของโครงสร้างเงื่อนไข if
3. จะอธิบายถึงรูปแบบโครงสร้างเงื่อนไขทำซ้ำ while และวัดแผนภาพการให้ผลของโครงสร้างเงื่อนไขทำซ้ำ while
4. จะอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย พิจารณาผลลัพธ์ที่ได้จากสคริปต์

```

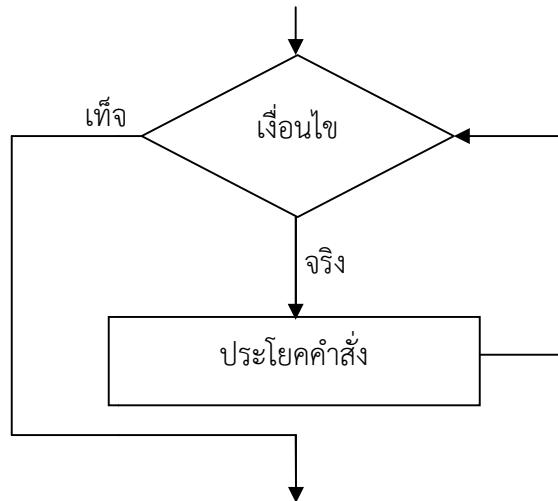
1 <?php
2     $row = 1;
3     $number = 12;
4     while ($row <= 20) {
5         printf ("%d %s %d %s %d <br>", $number, "x", $row, "=", $number*$row);
6         $row++;
7     }
8 ?>

```

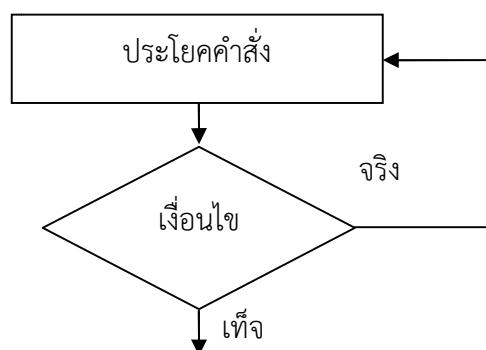


5. จงอธิบายหลักการทำงานของโครงสร้างควบคุม ดังนี้

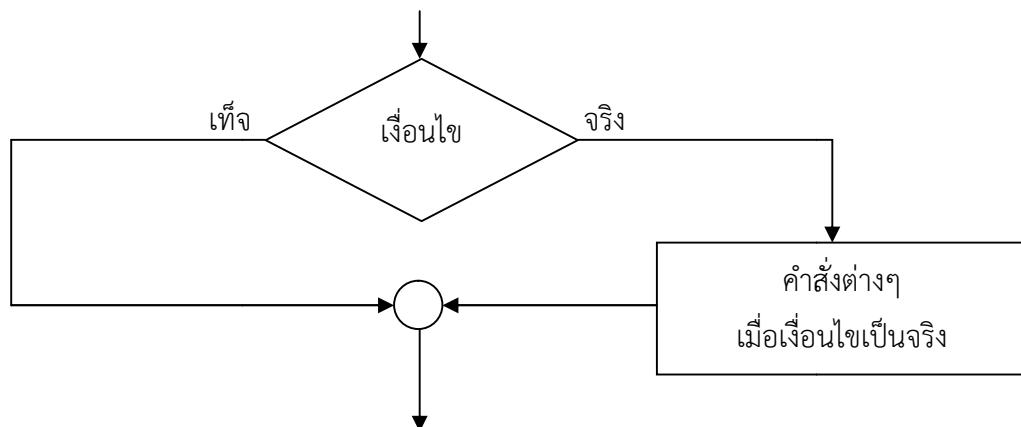
5.1 จากแผนภาพแสดงโครงสร้างควบคุมอะไร และมีหลักการทำงานอย่างไร



5.2 จากแผนภาพแสดงโครงสร้างควบคุมอะไร และมีหลักการทำงานอย่างไร



5.3 จากแผนภาพแสดงโครงสร้างควบคุมอะไร และมีหลักการทำงานอย่างไร



บทที่ 5

ฟังก์ชัน

ในบทนี้จะกล่าวถึง PHP ฟังก์ชัน รวมถึงวิธีการสร้าง ข้อกำหนด คุณสมบัติต่างๆ และการเรียกใช้งานฟังก์ชัน การส่งค่าตัวแปรเพื่อประมวลผลในฟังก์ชัน การส่งค่ากลับเมื่อฟังก์ชันทำงานเสร็จ เป็นต้น หน้าที่หลักๆ ของฟังก์ชันนั้น จะช่วยลดขั้นตอนการเขียนโปรแกรมที่ต้องทำซ้ำๆ หรือใช้งานบ่อยครั้ง ทำให้การเขียนโปรแกรมทำได้ง่ายและรวดเร็ว สามารถสร้างเป็นไลบรารีฟังก์ชันสำหรับการใช้งานในครั้งต่อๆ ไป

ฟังก์ชัน (Functions) คือ กลุ่มหรือชุดของคำสั่งที่สร้างขึ้นเพื่อทำหน้าที่เด่นๆ หนึ่ง เมื่อต้องการใช้งานก็เพียงเรียกชื่อฟังก์ชันนั้นก็สามารถใช้งานได้ทันที ฟังก์ชันสามารถแบ่งได้เป็น 2 กลุ่ม ประกอบด้วย 1) ฟังก์ชันมาตรฐาน (Built-in Functions) และ 2) ฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง (User Defined Functions)

ฟังก์ชันทั้ง 2 กลุ่ม สามารถจำแนกย่อยเป็น 2 ประเภทอย่เมื่อ กันทั้ง 2 กลุ่ม คือ 1) ฟังก์ชันไม่มีการส่งผ่านค่าพารามิเตอร์ และ 2) ฟังก์ชันที่มีการส่งผ่านค่าพารามิเตอร์ (พารามิเตอร์ หรือเรียกว่า อาคิวเมนต์ argument) หมายถึง ตัวแปรหรือข้อมูลที่ส่งจากภายนอกฟังก์ชันเข้ามาประมวลผลภายในฟังก์ชัน ในหนังสือเล่มนี้จะใช้คำว่า พารามิเตอร์ เพียงอย่างเดียวเมื่อกล่าวถึงตัวแปรหรือข้อมูลที่ส่งจากภายนอกฟังก์ชันเข้ามาประมวลผลภายในฟังก์ชัน

5.1 ฟังก์ชันมาตรฐาน

ฟังก์ชันมาตรฐาน คือ ฟังก์ชันที่มาพร้อมกับภาษา PHP สามารถเรียกใช้งานได้ทันที ฟังก์ชันมาตรฐานมีหลายกลุ่มการทำงาน (ในส่วนของฟังก์ชันมาตรฐานจะกล่าวถึงรูปแบบ และตัวอย่างการใช้งานในบทต่อไป) สามารถจำแนกตามหน้าที่ ดังนี้ 1) ฟังก์ชันที่เกี่ยวกับวันที่และเวลา 2) ฟังก์ชันที่เกี่ยวกับการคำนวนทางคณิตศาสตร์ 3) ฟังก์ชันที่เกี่ยวกับการติดต่อกับฐานข้อมูล และ 4) ฟังก์ชันที่เกี่ยวกับการจัดการข้อมูล

การเรียกใช้งานฟังก์ชันมาตรฐานภาษา PHP จะต้องตรวจสอบก่อนว่าฟังก์ชันนั้นๆ เป็นฟังก์ชันเพื่อทำหน้าที่อะไร มีการส่งผ่านค่าพารามิเตอร์หรือไม่ ถ้าไม่มีการส่งผ่านค่าพารามิเตอร์ ก็สามารถเรียกใช้งานได้เลย แต่ถ้ามีการส่งผ่านค่าพารามิเตอร์ ก็จำเป็นต้องระบุค่าพารามิเตอร์ ให้ถูกต้องตามรูปแบบที่ฟังก์ชันกำหนด

ตัวอย่างที่ 5.1 รูปแบบพัฟ์ชันที่ไม่มีการส่งผ่านค่าพารามิเตอร์

ชื่อพัฟ์ชัน ();

ตัวอย่างที่ 5.2 รูปแบบพัฟ์ชันที่มีการส่งผ่านค่าพารามิเตอร์

ชื่อพัฟ์ชัน (พารามิเตอร์ 1, พารามิเตอร์ 2, ...);

ตัวอย่างที่ 5.3 การเรียกใช้งานพัฟ์ชันมาตรฐาน พัฟ์ชัน date ()

```

1 <?php
2     $today = date("d/m/Y");
3     echo $today;
4 ?>
```

ผลลัพธ์

6/7/2012

จากตัวอย่างที่ 5.3 การเรียกใช้งานพัฟ์ชันมาตรฐาน พัฟ์ชัน date () อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$today มีค่าเท่ากับ ผลของการเรียกใช้พัฟ์ชัน date () โดยกำหนดพารามิเตอร์ ประกอบด้วย d/m/Y คือ วัน เดือน และปี ค.ศ.

บรรทัดที่ 3 แสดงค่าที่เก็บในตัวแปร \$today

บรรทัดที่ 4 สิ้นสุดสคริปต์ PHP

5.2 พัฟ์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง

พัฟ์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง คือ กลุ่มของคำสั่งที่ผู้ใช้งานเป็นผู้เขียนหรือพัฒนาขึ้นมาเอง เพื่อทำงานหรือทำหน้าที่อย่างโดยย่างหနึงตามต้องการ

5.2.1 การสร้างพัฟ์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง

ลักษณะของงานที่จะนำมาสร้างเป็นพัฟ์ชันนั้น ควรเป็นงานหรือการกระทำอย่างโดยย่างหนึง ที่จำเป็นต้องทำซ้ำๆ และบ่อยครั้ง เพื่อลดเวลาในการเขียนคำสั่งหรือชุดคำสั่งเมื่อต้องการทำงานแบบเดิม โดยสามารถแยกคำสั่งบางส่วนออกจากโครงสร้างเป็นพัฟ์ชันไว้เฉพาะ และเรียกใช้ตามลักษณะงานที่ต้องการ จะช่วยให้จำนวนบรรทัดคำสั่งน้อยลง (ได้ผลลัพธ์เหมือนเดิม) ช่วยลดการใช้ทรัพยากร และง่ายในการปรับปรุงแก้ไข เพราะสามารถแก้ไขเพียงครั้งเดียว ก็จะมีผลทุกจุดภายในโปรแกรมที่เรียกใช้พัฟ์ชัน

รูปแบบ

<?php

function ชื่อพัฟ์ชัน (พารามิเตอร์) {

```

    คำสั่ง 1;
    คำสั่ง 2;
    ...
}

?>
```

5.2.2 ข้อกำหนดหลักเกณฑ์การตั้งชื่อฟังก์ชัน มีหลักเกณฑ์คล้ายกับการตั้งชื่อตัวแปร ดังนี้

- 1) ต้องเป็นต้นตี่อตัวย า-z หรือ _ เท่านั้น
- 2) ต้องประกอบด้วย a-z, 0-9 หรือ _ เท่านั้น
- 3) ต้องไม่ซ้ำกับชื่อฟังก์ชันที่มีอยู่แล้วหรือฟังก์ชันมาตรฐานของภาษา PHP

5.2.3 การเรียกใช้ฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง สามารถทำได้เช่นเดียวกับการใช้ฟังก์ชันมาตรฐานของภาษา PHP คือ ต้องระบุชื่อฟังก์ชันที่ต้องการใช้งาน ระบุค่าพารามิเตอร์ (ถ้ามี) ตัวอย่างดังนี้

ตัวอย่างที่ 5.4 การสร้างฟังก์ชัน และเรียกใช้งานฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง

```

1 <?php
2     function multiplication ($number, $start, $end) {
3         line_(40, "*");
4         echo "Multiplication of $number : Starting at $start until $end <br>";
5         line_(40, "*");
6         for ($start=1; $start <= $end; $start++){
7             printf ("%d x %d = %d <br>", $number, $start, $number*$start);
8         }
9         line_(40, "*");
10    }
11    function line_ ($how_much, $symbol) {
12        for ($begin=1; $begin <= $how_much; $begin++) {
13            echo $symbol;
14        }
15        echo "<br>";
16    }
17    multiplication (12, 1, 5);
18 ?>
```



ผลลัพธ์

```
*****
Multiplication of 12 : Starting at 1 until 5
*****
```

$12 \times 1 = 12$

$12 \times 2 = 24$

$12 \times 3 = 36$

$12 \times 4 = 48$

$12 \times 5 = 60$

```
*****
```

จากตัวอย่างที่ 5.4 การสร้างฟังก์ชัน และเรียกใช้งานฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง ในสคริปต์ประกอบด้วย 2 ฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง อธิบายดังนี้

ฟังก์ชัน multiplication () ใช้สำหรับหาค่าผลคูณของเลขตามที่ระบุ (อยู่ในช่วงบรรทัดที่ 2 ถึงบรรทัดที่ 10) พารามิเตอร์ ประกอบด้วย \$number หมายถึง แม่เลขที่ต้องการแสดง \$start หมายถึง ลำดับเลขเริ่มต้นของการคูณ และ \$end หมายถึง ลำดับเลขสุดท้ายของการคูณ

ฟังก์ชัน line_ () ใช้สำหรับแสดงสัญลักษณ์และจำนวนสัญลักษณ์ตามที่ระบุ (อยู่ในช่วงบรรทัดที่ 11 ถึงบรรทัดที่ 16) พารามิเตอร์ ประกอบด้วย \$how_much หมายถึง จำนวนของ สัญลักษณ์ที่ต้องการแสดงต่อແລ厝 และ \$symbol หมายถึง สัญลักษณ์ที่ต้องการแสดง

จุดเริ่มต้นของคำสั่ง

บรรทัดที่ 17 ในช่วงเริ่มต้นนั้น ฟังก์ชันจะยังไม่ทำงานหากไม่มีการเรียกใช้งานฟังก์ชันดังนั้น เมื่อระบุ multiplication (12, 1, 5) จึงหมายความว่า แม่เลขที่ต้องการแสดงผล คือ 12 ลำดับเลขเริ่มต้นของการคูณ คือ เลข 1 และลำดับเลขสุดท้ายของการคูณ คือ เลข 5 ลำดับถัดไปที่จะเริ่มทำงาน คือ บรรทัดที่ 2

บรรทัดที่ 2 อธิบายเพิ่มเติมรูปแบบการส่งผ่านค่าพารามิเตอร์ไปยังฟังก์ชัน ดังภาพที่ 5.1

บรรทัดที่ 2 function multiplication (\$number, \$start, \$end) {

.....

}

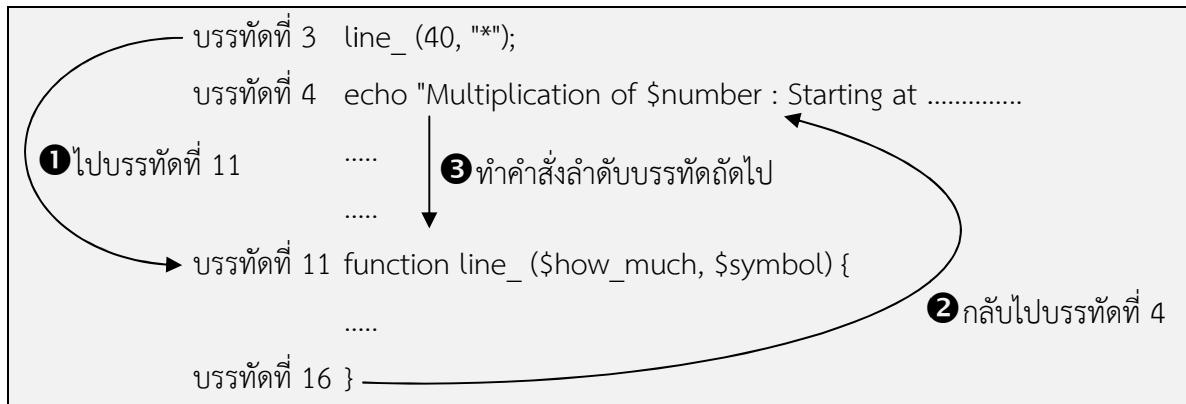
ส่งค่าพารามิเตอร์

บรรทัดที่ 17 multiplication (12, 1, 5);

ภาพที่ 5.1 แสดงการส่งผ่านค่าพารามิเตอร์เข้าไปประมวลผลภายใต้ฟังก์ชัน

จากภาพที่ 5.1 การส่งผ่านค่าพารามิเตอร์ ไปยังฟังก์ชันนั้น จะเรียงตามลำดับตัวแปรพารามิเตอร์ หมายความว่า ค่าเลข 12 จะถูกส่งไปยังตัวแปร \$number ค่าเลข 1 ถูกส่งไปยังตัวแปร \$start และ ค่าเลข 5 ถูกส่งไปยังตัวแปร \$end หลังจากนั้นฟังก์ชันจะนำตัวแปรทั้ง 3 เข้าไปประมวลผลภายในฟังก์ชันตามลำดับคำสั่งร่วมกับโครงสร้างเงื่อนไขทำซ้ำ for ภายในฟังก์ชัน (โครงสร้างเงื่อนไขทำซ้ำได้ก่อร่างและอธิบายแล้วในบทที่ 4)

ภายในฟังก์ชัน multiplication () มีการเรียกฟังก์ชัน line_() บรรทัดที่ 3 และบรรทัดอื่นๆ อธิบายลำดับการทำงาน ด้วยภาพที่ 5.2 ดังนี้



ภาพที่ 5.2 แสดงการเรียกใช้ฟังก์ชันภายนอกฟังก์ชันที่กำลังทำงานในปัจจุบัน

จากภาพที่ 5.2 บรรทัดที่ 3 line_() (เรียกใช้ฟังก์ชัน line_()) ภายในฟังก์ชัน multiplication () ดังนั้นลำดับบรรทัดถัดไปที่จะทำงาน คือ บรรทัดที่ 11 ถึงบรรทัดที่ 16 โดยค่าเลข 40 เป็นพารามิเตอร์ส่งไปให้ตัวแปร \$how_much และสัญลักษณ์ "*" ถูกส่งไปให้กับตัวแปร \$symbol ตามลำดับ และประมวลผลตามลำดับคำสั่งร่วมกับโครงสร้างเงื่อนไขทำซ้ำ for และเมื่อทำงานตามคำสั่งถึงบรรทัดที่ 16 จะกลับไปทำคำสั่งในบรรทัดที่ 4

5.3 ฟังก์ชันพารามิเตอร์

ฟังก์ชันพารามิเตอร์ คือ ค่าที่จะนำไปใช้หรือประมวลผลภายในฟังก์ชัน ซึ่งจะทำให้ฟังก์ชันมีความยืดหยุ่นต่อการใช้งาน ดังตัวอย่างที่ 5.4 เพราะผลลัพธ์จะแปรเปลี่ยนไปตามค่าพารามิเตอร์ที่ได้รับ รูปแบบการกำหนดและส่งผ่านค่าพารามิเตอร์ มีรายละเอียดดังนี้

5.3.1 พารามิเตอร์แบบกำหนดค่าเริ่มต้น (Default Parameter)

พารามิเตอร์แบบกำหนดค่าเริ่มต้น ในบางฟังก์ชันอาจใช้ค่าพารามิเตอร์ค่าเดิมที่ไม่เป็นค่าเริ่มต้น อาจจะมีการเปลี่ยนไปใช้ค่าอื่นบ้างในบางครั้ง ดังนั้นเพื่อความสะดวกจึงมีการกำหนดค่าพารามิเตอร์แบบกำหนดค่าเริ่มต้นขึ้น โดยจะกำหนดค่าพารามิเตอร์ที่ต้องใช้บ่อยๆ ไว้ล่วงหน้า หรือป้องกันปัญหาในกรณีที่ไม่ได้กำหนดค่าพารามิเตอร์ให้กับฟังก์ชัน เมื่อมีการเรียกใช้ฟังก์ชัน ซึ่งในกรณีที่ไม่ส่งค่าพารามิเตอร์มาให้ฟังก์ชัน ฟังก์ชันจะเรียกใช้ค่าเริ่มต้นที่กำหนดไว้ให้แทน รูปแบบการกำหนดพารามิเตอร์แบบกำหนดค่าเริ่มต้น มีรายละเอียด ดังนี้



รูปแบบ

```
<?php
    function ชื่อฟังก์ชัน (ชื่อพารามิเตอร์ = ค่าเริ่มต้น) {
        คำสั่ง;
    }
?>
```

ตัวอย่างที่ 5.5 พังก์ชันปรามิตตัวเลข

```
<?php
    function pyramidNumber ($number=7) {
        for ($loop1=$number; $loop1>=1; $loop1--) {
            for ($loop2=1; $loop2<=$loop1; $loop2++) {
                printf (" %d ",$loop2);
            }
            echo "<br>";
        }
    }
?>
```

ตัวอย่างที่ 5.6 การเรียกใช้ฟังก์ชันปรามิตตัวเลข

แบบไม่ส่งผ่านค่าพารามิเตอร์	แบบส่งผ่านค่าพารามิเตอร์
<?php pyramidNumber (); ?>	<?php pyramidNumber (4); ?>

ผลลัพธ์

1 2 3 4 5 6 7	1 2 3 4
1 2 3 4 5 6	1 2 3
1 2 3 4 5	1 2
1 2 3 4	1
1 2 3	
1 2	
1	

จากตัวอย่างที่ 5.6 การเรียกใช้ฟังก์ชันปรามิตตัวเลข คือ ฟังก์ชัน pyramidNumber () จากตัวอย่างที่ 5.5 จะเห็นได้ว่าหากไม่ส่งค่าพารามิเตอร์ไปยังฟังก์ชัน pyramidNumber () ฟังก์ชันจะ

กำหนดให้ตัวแปร \$number มีค่าเท่ากับ 7 และเริ่มทำงานภายในฟังก์ชัน ในกรณีเรียกใช้ฟังก์ชันโดยกำหนดค่าพารามิเตอร์เท่ากับ 4 ฟังก์ชัน pyramidNumber() ก็จะกำหนดให้ตัวแปร \$number มีค่าเท่ากับ 4 ตามค่าพารามิเตอร์ส่งค่าให้ฟังก์ชัน เป็นต้น

5.3.2 พารามิเตอร์แบบส่งผ่านค่าโดยการอ้างอิง (Passing Parameter by Value and Reference)

โดยปกติแล้วค่าที่ถูกส่งไปยังฟังก์ชันจะเป็นแบบส่งค่า (by Value) คือ เมื่อมีการเปลี่ยนแปลงค่าของตัวแปรภายในฟังก์ชันจะไม่ส่งต่อค่าตัวแปรตัวเดียวกันที่อยู่นอกฟังก์ชัน ทุกตัวอย่างก่อนหน้านี้ในเรื่องฟังก์ชันนี้ใช้วิธีการผ่านค่าพารามิเตอร์แบบส่งค่าทั้งหมด

การส่งผ่านค่าแบบอ้างอิง (by Reference) นั้น หากในฟังก์ชันมีการเปลี่ยนแปลงค่าของตัวแปรภายในฟังก์ชันจะส่งผลให้ค่าของตัวแปรที่อ้างอิงกันนอกฟังก์ชันมีค่าเปลี่ยนตามไปด้วย การส่งผ่านค่าแบบอ้างอิงสามารถทำได้โดยการใส่เครื่องหมาย & ไว้หน้าพารามิเตอร์ตัวที่ต้องการอ้างอิง

ตัวอย่างที่ 5.7 การเปรียบเทียบการส่งผ่านค่าพารามิเตอร์แบบปกติและแบบอ้างอิง

ก. การส่งผ่านค่าพารามิเตอร์แบบปกติ	ข. การส่งผ่านค่าพารามิเตอร์แบบอ้างอิง
1 <?php	1 <?php
2 function foo (\$var) {	2 function foo (&\$var) {
3 \$var++;	3 \$var++;
4 }	4 }
5 \$a=5;	5 \$a=5;
6 foo (\$a);	6 foo (\$a);
7 echo \$a;	7 echo \$a;
8 ?>	8 ?>
ผลลัพธ์ตัวแปร \$a มีค่าเท่ากับ 5	ผลลัพธ์ตัวแปร \$a มีค่าเท่ากับ 6

จากตัวอย่างที่ 5.7 การเปรียบเทียบการส่งผ่านค่าพารามิเตอร์แบบปกติและแบบอ้างอิงนั้น จริงๆ แล้วคือ ขอบเขตของตัวแปรนั้นเอง (ศึกษาเพิ่มเติมในบทที่ 3) อธิบายดังนี้

ตัวอย่าง 5.7 ก. การส่งผ่านค่าพารามิเตอร์แบบส่งผ่านค่าแบบปกติ ค่าพารามิเตอร์ที่ส่งไปยังฟังก์ชัน foo() ในบรรทัดที่ 6 มีค่าเท่ากับ 5 (ตัวแปร \$a มีค่าเท่ากับ 5) ภายในฟังก์ชัน foo() มีการนำค่าพารามิเตอร์ที่รับเข้ามาผ่านตัวแปร \$var บรรทัดที่ 3 มีการเพิ่มค่าให้กับตัวแปร \$var แล้วกลับมาบรรทัดที่ 7 เพื่อแสดงผลค่าตัวแปร \$a ค่าของตัวแปร \$a ที่แสดง คือ 5 เป็นผลมาจากบรรทัดที่ 5 ซึ่งถือว่าบรรทัดที่ 3 ไม่สามารถเปลี่ยนแปลงค่าตัวแปร \$a ในบรรทัดที่ 5 ได้

ตัวอย่าง 5.7 ข. การส่งผ่านค่าพารามิเตอร์แบบอ้างอิง ค่าพารามิเตอร์ที่ส่งไปยังฟังก์ชัน foo() ในบรรทัดที่ 6 มีค่าเท่ากับ 5 (ตัวแปร \$a มีค่าเท่ากับ 5) และฟังก์ชันมีการกำหนดแบบอ้างอิง คือ &\$var หมายความว่า กำหนดให้ตัวแปร \$var อ้างอิงไปยัง \$a (กำหนดให้เป็นตัวแปรอ้างอิง ดังนั้นหากตัวแปรตัวหนึ่งตัวใดมีการเปลี่ยนแปลงค่า จะยังผลให้ตัวแปรที่อ้างอิงกันมีค่าเปลี่ยนแปลงตามไปด้วย)



บรรทัดที่ 3 มีการเพิ่มค่าให้กับตัวแปร \$var (เพิ่มค่าให้กับตัวแปร \$a ด้วย) และกลับมาบรรทัดที่ 7 เพื่อแสดงผลค่าตัวแปร \$a ดังนั้นค่าของตัวแปร \$a ที่แสดง จึงมีค่าเท่ากับ 6

5.4 การส่งค่ากลับจากฟังก์ชันด้วยคำสั่ง return

เนื่องจากฟังก์ชันจะใช้ในการประมวลผลอย่างโดยย่างหนึ่ง โดยฟังก์ชันมักจะถูกเรียกใช้โดยส่วนต่างๆ ของโปรแกรม เพื่อประมวลผลตามหน้าที่ต่างๆ ของฟังก์ชัน ในบางครั้งฟังก์ชันอาจจำเป็นต้องส่งค่าผลลัพธ์ของการทำงานกลับไปยังส่วนที่เรียกใช้ฟังก์ชันนั้นๆ หรือสามารถประยุกต์ใช้สำหรับการตรวจสอบการทำงานของฟังก์ชัน เช่น ทำงานปกติอาจส่งค่ากลับเป็นเลข 1 ทำงานไม่ถูกต้องส่งค่ากลับเป็นเลข 2 หรืออื่นๆ ตามต้องการ เป็นต้น สำหรับวิธีการส่งค่ากลับออกไปจะใช้คำสั่ง return และตามด้วยค่าที่ต้องการส่งออกไป มีรูปแบบ ดังนี้

```
return ค่าที่จะส่งกลับ;
```

5.4.1 การส่งค่ากลับหนึ่งค่า (Returning Single Value)

กำหนดให้มีฟังก์ชันทำงานเสร็จมีการส่งผลลัพธ์ที่ได้จากการประมวลผล มากำหนดค่าให้กับตัวแปร หรือแสดงผล หรืออื่นๆ ค่าที่ส่งกลับจะมีเพียง 1 ค่า ตัวอย่างดังนี้

ตัวอย่างที่ 5.8 ฟังก์ชัน calcSalesTax () มีการส่งค่ากลับเป็นค่าของตัวแปร \$total

```
1 <?php
2     function calcSalesTax ($price, $tax=.0675) {
3         $total = $price + ($price * $tax);
4         return $total;
5     }
6 ?>
```

จากตัวอย่างที่ 5.8 แสดงตัวอย่างการส่งค่ากลับจากฟังก์ชัน calcSalesTax () ผ่านตัวแปร \$total โดยค่าของตัวแปร \$total มาจากนิพจน์ในบรรทัดที่ 3 และฟังก์ชันส่งค่ากลับเป็นค่าของตัวแปร \$total ไปยังตำแหน่งที่เรียกใช้ฟังก์ชัน

ตัวอย่างที่ 5.9 ฟังก์ชัน calcSalesTax () มีการส่งค่ากลับเป็นผลลัพธ์ของการคำนวณ

```
1 <?php
2     function calcSalesTax ($price, $tax=.0675) {
3         return $price + ($price * $tax);
4     }
5 ?>
```

จากตัวอย่างที่ 5.9 แสดงตัวอย่างการส่งค่ากลับจากฟังก์ชัน calcSalesTax () โดยใช้ผลลัพธ์จากนิพจน์ส่งค่ากลับไปยังตำแหน่งที่เรียกใช้ฟังก์ชัน

ตัวอย่างที่ 5.10 ตัวอย่างการเรียกใช้ฟังก์ชัน calcSalesTax ()

```

1 <?php
2     echo calcSalesTax (500, $tax=.0675);
3         // หรือใช้ตัวแปรรับผลการ return ของฟังก์ชัน ดังนี้
4         // $value = calcSalesTax (500, $tax=.0675); และนำตัวแปร $value ไปแสดงผล
5     ?>

```

ผลลัพธ์

533.75

จากตัวอย่างที่ 5.10 แสดงตัวอย่างการเรียกใช้ฟังก์ชัน calcSalesTax () จากตำแหน่งใดๆ ภายในเว็บเพจเดียวกัน ในบรรทัดที่ 2 เป็นการแสดงผลลัพธ์โดยตรงจากค่าที่ส่งกลับจากฟังก์ชัน หรือสามารถใช้ตัวแปรรับผลค่าจากฟังก์ชันก่อน แล้วจึงนำตัวแปรที่รับผลค่าไปแสดงผล ดังนี้

ตัวอย่างที่ 5.11 ตัวอย่างการเรียกใช้ตัวแปรรับผลค่าจากฟังก์ชัน calcSalesTax ()

```

1 <?php
2     $value = calcSalesTax (500, $tax=.0675);
3     echo $value;
4 ?>

```

5.4.2 การส่งค่ากลับแบบหลายค่า (Returning Multiple Values)

ในบางครั้งของการเรียกใช้ฟังก์ชัน อาจจำเป็นต้องต้องส่งค่ากลับมากกว่า 1 ค่า เช่น เขียนฟังก์ชันเพื่อติดต่อกับฐานข้อมูล และต้องการส่งค่ากลับประกอบด้วย ชื่อ-สกุล อีเมล์ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น ซึ่งในตัวอย่างก่อนหน้านี้จะกล่าวถึงเฉพาะส่งค่ากลับเพียง 1 ค่าเท่านั้น ในการนี้มีฟังก์ชัน มาตรฐานของภาษา PHP ที่สามารถช่วยได้ คือ ฟังก์ชัน list () ใช้สำหรับส่งค่ากลับมากกว่า 1 ค่าในรูปแบบของอาร์เรย์ ตัวอย่างดังนี้

ตัวอย่างที่ 5.12 การส่งค่ากลับแบบหลายค่า

```

1 <?php
2     function retrieveUserProfile ( ) {
3         $user [ ] = "Parinya";
4         $user [ ] = "parinya@example.com";
5         $user [ ] = "Thai";
6         return $user;

```



```

7      }
8      list ($name, $email, $language) = retrieveUserProfile ( );
9      echo "Name: $name, email: $email, language: $language";
10     ?>

```

ผลลัพธ์

Name: Parinya, email: parinya@example.com, language: Thai

จากตัวอย่างที่ 5.11 การส่งค่ากลับแบบหลายค่า เป็นการประยุกต์ใช้ข้อมูลอาร์เรย์สำหรับส่งค่ากลับจากฟังก์ชัน (โดยปกติฟังก์ชันจะส่งค่ากลับได้เพียง 1 ค่าเท่านั้น) ในบรรทัดที่ 8 ผลของฟังก์ชัน retrieveUserProfile () จะส่งค่ากลับเป็นข้อมูลชนิดอาร์เรย์ ให้กับฟังก์ชัน list () เพื่อแยกข้อมูลอาร์เรย์ออกเป็นตัวแปรเดี่ยวแล้วกำหนดค่าให้กับตัวแปรตามลำดับ ประกอบด้วยตัวแปร \$name, \$email และ \$language แล้วนำตัวแปรทั้งหมดไปแสดงผลในบรรทัดที่ 9

5.5 พัฟก์ชันแบบเรียกตัวเอง

ฟังก์ชันแบบเรียกตัวเอง (Recursive Functions) คือ พัฟก์ชันที่มีการเรียกใช้ฟังก์ชันตัวเอง ใช้สำหรับทำงานซ้ำๆ กันโดยมีจุดสิ้นสุดการทำงานอยู่ในฟังก์ชัน เพื่อให้ฟังก์ชันหยุดการทำงาน การเขียนฟังก์ชันแบบเรียกตัวเอง จะช่วยลดระยะเวลาในการเขียนสคริปต์ องค์ประกอบที่สำคัญของฟังก์ชันแบบเรียกตัวเอง มีรายละเอียดดังนี้

5.5.1 มีการเรียกใช้ฟังก์ชันซ้อนฟังก์ชันตัวเองภายในฟังก์ชัน อาจจะอยู่ในส่วนคำสั่งหรือเป็นองค์ประกอบในนิพจน์

5.5.2 มีการรับค่าผ่านพารามิเตอร์ของฟังก์ชัน โดยค่าดังกล่าวจะถูกนำไปใช้ภายในฟังก์ชันเรียกตัวเอง ผลของการคำนวณบางส่วนจะถูกนำกลับไปใช้เมื่อมีการเรียกใช้ฟังก์ชันแบบเรียกตัวเอง

5.5.3 มีการส่งผลการคำนวณกลับ โดยผลดังกล่าวจะนำไปใช้ในการคำนวณในฟังก์ชันแบบเรียกตัวเองหรืออาจจะส่งผลกลับไปให้กับส่วนที่เรียกใช้ฟังก์ชันในครั้งแรก

5.5.4 มีจุดสิ้นสุดของการเรียกใช้ฟังก์ชันตัวเอง ฟังก์ชันแบบเรียกตัวเองจะทำงานไม่ได้หากไม่มีช่องทางให้สคริปต์หยุดการเรียกตัวเองในขั้นใดขั้นหนึ่ง เพราะจะเกิดการเรียกฟังก์ชันแบบเรียกตัวเองไม่มีสิ้นสุด

การเขียนฟังก์ชันแบบเรียกตัวเองช่วยแก้ปัญหาที่ซับซ้อนได้โดยการเขียนฟังก์ชันในรูปแบบที่ง่ายและสั้น ตัวอย่างดังนี้

ตัวอย่างที่ 5.13 การสร้างและการเรียกใช้ฟังก์ชันแบบเรียกตัวเอง สำหรับหาค่าแฟกทอเรียล n!

```

1 <?php
2     function factorial ($n) {

```

```

3         if ($n < 1)
4             return 1;
5         else
6             return $n*factorial($n-1);
7     }
8     echo factorial (4);
9 ?>

```

ผลลัพธ์

24

จากตัวอย่างที่ 5.13 การสร้างและการเรียกใช้ฟังก์ชันแบบเรียกตัวเอง สำหรับหาค่าแฟกทอเรียล $n!$ โดยหลักแนวคิด และการทำงานของฟังก์ชัน มีดังนี้

$$\begin{aligned}
 \text{factorial } (\$n) &= \$n * (\$n-1) * (\$n-2) * (\$n-3) \\
 \text{ดังนั้น} \quad 4! &= 4 * 3 * 2 * 1 \\
 &= 24
 \end{aligned}$$

จะเห็นได้ว่าตัวแปร $\$n$ จะต้องเป็นตัวแปรที่เก็บค่าที่เป็นเลขจำนวนเต็ม และไม่เป็นลบ ถ้าต้องการจะเขียนฟังก์ชันให้มีความปลอดภัยในการใช้งาน ก็อาจจะเพิ่มเงื่อนไข เพื่อตรวจสอบก่อนว่า ผู้ใช้ ส่งค่าของตัวแปรตรงตามต้องการเงื่อนไขหรือไม่ เช่น ไม่ส่งค่าที่เป็นข้อความ หรือเป็นเลข SCN หรือ ค่าที่เป็นลบ เป็นต้น จากตัวอย่างจะเป็นการทำซ้ำโดยใช้รูปแบบฟังก์ชันเรียกใช้ตัวเอง จะหยุดกีต่อเมื่อ ตัวแปร $\$n$ มีค่าน้อยกว่าหรือเท่ากับ 1 เมื่อเป็นจริงตามเงื่อนไขฟังก์ชันจะหยุดทำงานโดยใช้คำสั่ง `return` 1 หรือ `return true`

ตัวอย่างที่ 5.14 การสร้างและการเรียกใช้ฟังก์ชันแบบเรียกตัวเอง สำหรับหาค่าผลรวม n จำนวน

```

1 <?php
2     function totalnum ($n) {
3         if ($n <=1 ) {
4             return 1;
5         }
6         $total=totalnum($n-1)+$n;
7         return $total;
8     }
9     echo totalnum (20);
10 ?>

```



ผลลัพธ์

210

จากตัวอย่างที่ 5.14 พัฟ์ชัน totalnum () ใช้สำหรับหาค่าผลรวมของ n จำนวน โดยหลักแนวคิด และการทำงานของพัฟ์ชัน มีดังนี้

$$\begin{aligned} \text{totalnum} (\$n) &= (\$n-19)+(\$n-18)+(\$n-17)+(\$n-16)+(\$n-15)+(\$n-14)+(\$n-13)+ \\ &\quad (\$n-12)+(\$n-11)+(\$n-10)+(\$n-9)+(\$n-8)+(\$n-7)+(\$n-6)+(\$n-5)+ \\ &\quad (\$n-4)+(\$n-3)+(\$n-2)+(\$n-1)+20 \end{aligned}$$

$$\begin{aligned} \text{ดังนั้น} &= 1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20 \\ &= 210 \end{aligned}$$

จากตัวอย่างจะเป็นการทำงานซ้ำโดยใช้รูปแบบพัฟ์ชันเรียกใช้ตัวเอง จะหยุดกีต่อเมื่อตัวแปร $\$n$ มีค่าน้อยกว่าหรือเท่ากับ 1 เมื่อเป็นจริงตามเงื่อนไขพัฟ์ชันจะหยุดทำงานโดยใช้คำสั่ง return 1 หรือ return true

5.6 พัฟ์ชันไลบรารี

พัฟ์ชันไลบรารี (Function Libraries) เป็นวิธีที่จะช่วยลดการเขียนคำสั่งหรือพัฟ์ชันซ้ำซ้อน คือ การเขียนคำสั่งการทำงานแยกเก็บไว้อีกไฟล์หนึ่ง ซึ่งอาจเป็นชุดคำสั่ง พัฟ์ชัน คลาส หรืออื่นๆ ที่เกี่ยวข้อง พัฟ์ชันไลบรารีหรืออาจจะเรียกว่าคลังเก็บพัฟ์ชันก็ว่าได้ และเมื่อต้องการนำไฟล์ดังกล่าวมาใช้งานที่เว็บเพจได ก็ให้นำเข้าไฟล์ที่เก็บพัฟ์ชันนั้นๆ หลังจากนั้นจะสามารถใช้งานชุดคำสั่ง พัฟ์ชัน คลาส หรือ อื่นๆ ได เสมือนว่าพัฟ์ชันที่จะเรียกใช้นั้นอยู่ในเว็บเพจเดียวกัน พัฟ์ชันสำหรับการเรียกใช้หรือนำเข้าไฟล์ จากภายนอก ประกอบด้วย 1) include () 2) include_once () 3) require () และ 4) require_once () มีรายละเอียดการใช้งานในแต่ละพัฟ์ชัน ดังนี้

5.6.1 พัฟ์ชัน include ()

เป็นพัฟ์ชันที่ใช้สำหรับนำเข้าไฟล์จากภายนอก เพื่อนำเข้ามาใช้งานร่วมกับคำสั่งอื่นๆ ภายในหน้าเว็บเพจ ในกรณีพัฟ์ชัน include () เรียกหาไฟล์ที่ระบุไม่พบ ระบบจะแจ้งเตือน (Warning) ข้อผิดพลาดแล้วข้ามไปทำส่วนอื่นๆ ของคำสั่งในลำดับถัดไป

5.6.2 พัฟ์ชัน include_once ()

เป็นพัฟ์ชันที่ใช้สำหรับนำเข้าไฟล์จากภายนอก เมื่อมองกับพัฟ์ชัน include () แต่ต่างกัน ที่พัฟ์ชัน include_once () จะนำเข้าไฟล์จากภายนอกนั้นได้เพียงครั้งเดียวเท่านั้น

ตัวอย่างที่ 5.15 ตัวอย่างไฟล์ที่เก็บชุดคำสั่งกำหนดชื่อเป็น lib.php

```
<?php
echo "parinya<br>";
?>
```

ตัวอย่างที่ 5.16 การเปรียบเทียบความแตกต่างระหว่างฟังก์ชัน include () และ include_once ()

1 <?php	1 <?php
2 include ("lib.php");	2 include_once ("lib.php");
3 include ("lib.php");	3 include_once ("lib.php");
4 include ("lib.php");	4 include_once ("lib.php");
5 ?>	5 ?>

ผลลัพธ์

ผลลัพธ์ของการใช้ฟังก์ชัน include (); parinya parinya parinya	ผลลัพธ์ของการใช้ฟังก์ชัน include_once (); parinya
---	--

ในกรณีที่เรียกใช้ฟังก์ชัน include () หรือ include_once () และไม่พบไฟล์ที่จะนำเข้า จะปรากฏข้อความตามตัวอย่าง แต่ยังสามารถทำงานได้ในลำดับบรรทัดถัดไป ตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 5.17 แสดงตัวอย่างข้อผิดพลาดเมื่อไม่พบไฟล์ที่จะนำเข้า

<?php include ("file.php"); echo "ยังสามารถแสดงผลบรรทัดนี้ได้ "; ?>

ผลลัพธ์

Warning: include(file.php) [function.include]: failed to open stream: No such file or directory in C:\AppServ\www\ test.php on line 2 Warning: include() [function.include]: Failed opening 'file.php' for inclusion (include_path='.;C:\php5\pear') in C:\AppServ\www\ test.php on line 2 ยังสามารถแสดงผลบรรทัดนี้ได้
--

5.6.3 ฟังก์ชัน require ()

เป็นฟังก์ชันที่ใช้สำหรับนำเข้าไฟล์จากภายนอกเหมือนกับฟังก์ชัน include () แต่ต่างกันที่ ฟังก์ชัน require () เมื่อไม่พบไฟล์ที่จะนำเข้าจะแจ้งเตือน และแสดงข้อผิดพลาด (Fatal Error) จากนั้น จะหยุดการทำงานไม่ทำคำสั่งในลำดับบรรทัดถัดไป นิยมใช้นำเข้าไฟล์จากภายนอกที่จัดเก็บฟังก์ชันหลัก หรือคลาสที่มีความจำเป็นต่อการทำงานของโปรแกรม

5.6.4 ฟังก์ชัน require_once ()

เป็นฟังก์ชันที่มีการทำงานเหมือนกับฟังก์ชัน require () แต่ต่างกันที่ฟังก์ชัน require_once () จะนำเข้าไฟล์จากภายนอกนั้นได้เพียงครั้งเดียว

ในกรณีที่เรียกใช้ฟังก์ชัน require () หรือ require_once () และไม่ระบุไฟล์ที่จะนำเข้า จะปรากฏข้อความตามตัวอย่าง และจะไม่ทำงานในลำดับบรรทัดต่อไป ตัวอย่างดังนี้

ตัวอย่างที่ 5.18 การใช้ฟังก์ชัน require_once ()

```

1 <?php
2     require ("file.php");
3     echo "ยังสามารถแสดงผลบรรทัดนี้ได้<br>";
4 ?>

```

ผลลัพธ์

Warning: require(file.php) [function.require]: failed to open stream: No such file or directory in C:\AppServ\www\ test.php on line 2

Fatal error: require() [function.require]: Failed opening required 'file.php' (include_path='.;C:\php5\pear') in C:\AppServ\www\test.php on line 2

จากตัวอย่างดังกล่าว จะเห็นได้ว่าไม่ปรากฏคำว่า "ยังสามารถแสดงผลบรรทัดนี้ได้" เนื่องจากลักษณะ ดังกล่าวว่า Fatal Error นิยมใช้นำเข้าไฟล์จากภายนอกที่จัดเก็บฟังก์ชันหลัก หรือคลาสที่มีความจำเป็น ต่อการทำงานของโปรแกรม

สรุป

การเขียนสคริปต์ PHP นอกจากจะมีฟังก์ชันมาตรฐานที่มากับโปรแกรมภาษา PHP เพื่ออำนวย ความสะดวกในการเขียนสคริปต์แล้ว ผู้ใช้งานสามารถสร้างฟังก์ชันขึ้นมาใช้งานได้ด้วยตนเอง ลักษณะของ งานที่จะนำมาสร้างเป็นฟังก์ชันนั้น ควรเป็นงานหรือการกระทำอย่างใดอย่างหนึ่ง ที่มีการทำงานแบบช้าๆ และบ่อยครั้ง สามารถนำมาสร้างเป็นฟังก์ชันเพื่อลดเวลาในการเขียนสคริปต์ ฟังก์ชันที่ผู้ใช้สร้างอาจจะมี การส่งผ่านค่าตัวแปรเข้าไปประมวลผลภายในฟังก์ชัน แล้วส่งค่ากลับเพื่อแสดงผลหรืออื่นๆ บางฟังก์ชัน อาจเป็นเพียงรูปแบบการแสดงผลง่ายๆ แต่ทำงานแบบช้าๆ อาจจะไม่จำเป็นต้องส่งผ่านค่าตัวแปรก็ได้ อีกทั้งยังสามารถสร้างแยกออกจากเว็บเพจ เป็นไฟล์ที่เก็บฟังก์ชันโดยเฉพาะ หรือเรียกว่าฟังก์ชันไลบรารี และเมื่อต้องการเรียกใช้งานฟังก์ชันใดๆ ก็เพียงนำเข้าไฟล์ที่เก็บฟังก์ชันนั้น ก็จะสามารถเรียกใช้งานได้ ทันที ทำให้การเขียนสคริปต์ทำได้ง่ายและรวดเร็ว

คำถามท้ายบท

1. จงอธิบายข้อกำหนด และหลักเกณฑ์การตั้งชื่อฟังก์ชัน พร้อมยกตัวอย่างรูปแบบการสร้าง ฟังก์ชันที่ผู้ใช้งานเป็นผู้สร้างเอง
2. จงอธิบายฟังก์ชันพารามิเตอร์แบบส่งผ่านค่าโดยการอ้างอิง พร้อมยกตัวอย่างประกอบ

3. จงอธิบายถึงหลักการทำงานของฟังก์ชันแบบเรียกตัวเอง และอธิบายถึงคำสั่งที่จะให้ฟังก์ชันแบบเรียกตัวเองสืบสุ่มการทำงาน

4. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการรันสคริปต์

```

1 <?php
2     function pyramid ($number=9) {
3         for ($loop1 = $number; $loop1 >= 1; $loop1--) {
4             for ($loop2=1; $loop2<=$loop1; $loop2++) {
5                 printf (" %d ",$loop1);
6             }
7             echo "<br>";
8         }
9     }
10    pyramid (5);
11 ?>

```

5. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการรันสคริปต์

```

1 <?php
2     function pyramid ($number=9) {
3         for ($loop = $number; $loop >=1; $loop--) {
4             printf (" %d ",$loop);
5         }
6     }
7     for($row=5; $row>=1; $row--) {
8         pyramid ($row);
9         echo "<br>";
10    }
11 ?>

```





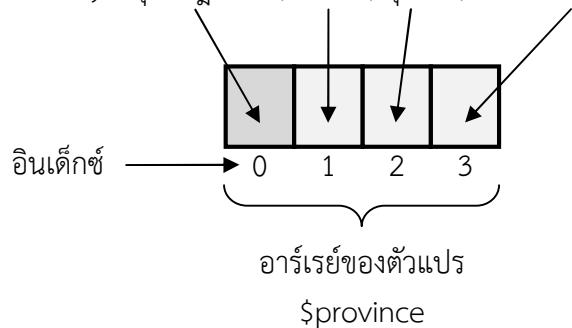
บทที่ 6

อาร์เรย์

อาร์เรย์ คือ ชุดของข้อมูล (Data Sets) หรือชุดของตัวแปร ใช้สำหรับเก็บค่าของข้อมูลที่อยู่ในกลุ่มเดียวกัน มีการเรียงลำดับที่แน่นอน ลำดับของอาร์เรย์โดยปกติจะเริ่มจากศูนย์ และเป็นลำดับต่อเนื่องไปจนถึงตัวสุดท้าย อาร์เรย์แตกต่างจากตัวแปรทั่วไป คือ ตัวแปรโดยทั่วไปจะถูกจัดเก็บในหน่วยความจำแบบไม่ต่อเนื่องกัน แต่ตัวแปรประเภทอาร์เรย์จะถูกจัดเก็บในหน่วยความจำในตำแหน่งที่ต่อเนื่องกัน

ข้อมูลแต่ละตัวในอาร์เรย์เรียกว่า "สมาชิก (Member)" หรืออลิเมนต์ (Element)" โดยสมาชิกเหล่านี้มักจะมีความสัมพันธ์กันในลักษณะใดลักษณะหนึ่ง สำหรับใน PHP ข้อมูลที่เก็บในอาร์เรย์ไม่จำเป็นต้องเป็นข้อมูลชนิดเดียวกัน สมาชิกแต่ละตัวจะประกอบด้วยค่าข้อมูล (Value) และอินเด็กซ์ (Index) หรือคีย์ (Key) ของอาร์เรย์ที่ใช้อ้างอิงตำแหน่งของสมาชิกแต่ละตัวในอาร์เรย์ ตัวอย่างรูปแบบการจัดเก็บตัวแปรอาร์เรย์ในหน่วยความจำ ดังภาพที่ 6.1

```
$province = array ("สุราษฎร์ธานี","กระบี่","ชุมพร","นครศรีธรรมราช");
```



ภาพที่ 6.1 ตัวอย่างการจัดเก็บตัวแปรอาร์เรย์ \$province ในหน่วยความจำ

จากภาพที่ 6.1 แสดงตัวอย่างการจัดเก็บตัวแปรอาร์เรย์ \$province ในหน่วยความจำ เมื่อกำหนดด้วยคำสั่ง `$province = array ("สุราษฎร์ธานี","กระบี่","ชุมพร","นครศรีธรรมราช");` ค่าของอินเด็กซ์จะเริ่มจาก 0 ถึง 3 และจะมีค่าเท่ากับข้อมูลที่กำหนดให้เรียงตามลำดับ ดังตารางที่ 6.1 แสดงความสัมพันธ์ระหว่างอินเด็กซ์กับค่าของข้อมูลที่กำหนดให้ตัวแปร \$province

ตารางที่ 6.1 แสดงความสัมพันธ์ระหว่างอินเด็กซ์กับค่าของข้อมูลที่กำหนดให้ตัวแปร \$province

ลำดับที่ของสมาชิก	อินเด็กซ์	ค่าของข้อมูล
1	\$province [0]	สุราษฎร์ธานี
2	\$province [1]	กระบี่
3	\$province [2]	ชุมพร
4	\$province [3]	นครศรีธรรมราช

6.1 การสร้างอาร์เรย์

การสร้างอาร์เรย์หลักๆ มี 4 วิธี คือ 1) การใช้ฟังก์ชัน array () 2) การใช้ฟังก์ชัน range () 3) การสร้างโดยใช้เครื่องหมายวงเล็บก้ามปู ([]) และ 4) การสร้างโดยนำค่าจากไฟล์ มีตัวอย่าง ดังนี้

6.1.1 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน array ()

1) การสร้างอาร์เรย์โดยใช้ฟังก์ชัน array () แบบไม่กำหนดอินเด็กซ์ให้กับสมาชิก ดังนี้

รูปแบบ

$\text{ชื่ออาร์เรย์} = \text{array} (\text{สมาชิกลำดับที่ 1}, \text{สมาชิกลำดับที่ 2}, \dots, \text{สมาชิกลำดับที่ N});$

ตัวอย่างที่ 6.1 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน array () ไม่กำหนดอินเด็กซ์ให้กับสมาชิก

```

1 <?php
2     $province = array ("สุราษฎร์ธานี", "กระบี่", "ชุมพร", "นครศรีธรรมราช");
3     print_r ($province); // แสดงข้อมูลในอาร์เรย์
4 ?>

```

ตัวอย่างที่ 6.2 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน array () แบบไม่กำหนดอินเด็กซ์ร่วมกับวงเล็บก้ามปูเพื่อกำหนดค่าให้กับสมาชิกในอาร์เรย์

```

1 <?php
2     $province = array ( );
3     $province [ ] = "สุราษฎร์ธานี";
4     $province [ ] = "กระบี่";
5     $province [ ] = "ชุมพร";
6     $province [ ] = "นครศรีธรรมราช";
7     print_r ($province);
8 ?>

```

ผลลัพธ์ของตัวอย่างที่ 6.1 และ 6.2 เมื่อนอกัน ดังนี้

Array ([0] => สุราษฎร์ธานี [1] => กระบี่ [2] => ชุมพร [3] => นครศรีธรรมราช)

การอ้างถึงสมาชิกแต่ละตัวต้องระบุหมายเลขอินเด็กซ์ เริ่มต้นจากเลข 0 และ 1, 2, 3 ตามลำดับ การระบุเลขอินเด็กซ์จะต้องใส่ภายในเครื่องหมายวงเล็บก้ามปู เช่น echo \$province [0] หมายถึง แสดงอาร์เรย์อินเด็กซ์ที่ 0 คือ สุราษฎร์ธานี เป็นต้น

หมายเหตุ

ฟังก์ชันที่ใช้แสดงผลข้อมูลชนิดอาร์เรย์ มีให้เลือกใช้ดังนี้

- ฟังก์ชัน `print_r()` ใช้แสดงข้อมูลจากตัวแปรทุกชนิด ไม่ว่าจะเป็น string, integer, array, object มีรูปแบบคำสั่ง ดังนี้

```
mixed print_r ( mixed $expression [, bool $return = false ] )
```

เมื่อ `$expression` หมายถึง ชื่อตัวแปรที่ต้องการนำข้อมูลมาแสดง

`$return` หมายถึง กำหนดให้มีการส่งค่าคืนหรือไม่ โดยปกติจะเป็น `false` คือ
ไม่มีการส่งค่าคืน และแสดงข้อมูลออกทางหน้าจอ หากกำหนด
เป็น `true` จะส่งคืนค่าและต้องกำหนดตัวแปรมารับค่า และ
ไม่แสดงข้อมูลออกทางหน้าจอ

- ฟังก์ชัน `var_dump()` ใช้แสดงข้อมูลเช่นเดียวกับ `print_r()` แต่สามารถใช้กับอาร์เรย์หลาย
มิติได้ และไม่มีการส่งค่าคืน รูปแบบดังนี้

```
void var_dump ( mixed $expression [, mixed $... ] )
```

2) การสร้างอาร์เรย์โดยใช้ฟังก์ชัน `array()` แบบกำหนดคีย์ให้กับสมาชิก มีรูปแบบดังนี้

```
$ชื่ออาร์เรย์ = array (คีย์ 1=>สมาชิกลำดับที่ 1, คีย์ 2=>สมาชิกลำดับที่ 2, ... );
```

ตัวอย่างที่ 6.3 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน `array()` แบบกำหนดคีย์ให้กับสมาชิก

```
1 <?php
2     $province = array ("Surat" => "สุราษฎร์ธานี", "Krabi" => "กระบี่", "Chumphon" =>
3     "ชุมพร", "Nakhon" => "นครศรีธรรมราช");
4     print_r ($province);
5 ?>
```

ผลลัพธ์

```
Array ( [Surat] => สุราษฎร์ธานี [Krabi] => กระบี่ [Chumphon] => ชุมพร [Nakhon] =>
    นครศรีธรรมราช )
```

การอ้างถึงสมาชิกแต่ละตัวต้องระบุคีย์ การระบุคีย์จะต้องใส่ภายในเครื่องหมายวงเล็บ
ก้ามปู เช่น `echo $province [Surat]` หมายถึง แสดงข้อมูลสมาชิกคีย์ Surat คือ สุราษฎร์ธานี เป็นต้น

ตัวอย่างที่ 6.4 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน `array()` กำหนดอินเด็กซ์ให้กับสมาชิก

```
1 <?php
2     $province = array ( );
3     $province [0] = "สุราษฎร์ธานี"; // สร้างอาร์เรย์ว่าง
4     // กำหนดสมาชิก
```



```

4 ...
5 $province [0] = "ยะลา";           // แก้ไขปรับปรุงค่าข้อมูลสมาชิกเดิม
6 $province [7] = "เชียงใหม่";       // เพิ่มข้อมูลสมาชิกใหม่
7 print_r ($province);
8 ?>

```

จากตัวอย่างที่ 6.4 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน array () กำหนดอินเด็กซ์ให้กับสมาชิก จะเห็นได้ว่าตำแหน่งบรรทัดที่ 3 มีการกำหนดให้ตัวแปร \$province [0] = "สุราษฎร์ธานี" และกำหนดค่าใหม่ให้กับตัวแปร \$province [0] = "ยะลา" ซึ่งประโยชน์ของการกำหนดอินเด็กซ์ให้กับสมาชิกจะทำให้ทราบที่อยู่ที่แน่นอนของสมาชิกแต่ละตัว ทำให้สะดวกในการเปลี่ยนแปลงข้อมูลหรืออื่นๆ

6.1.2 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน range ()

ฟังก์ชัน range () เป็นฟังก์ชันที่ใช้สำหรับสร้างอาร์เรย์ โดยกำหนดค่าข้อมูลเป็นช่วงของตัวเลขหรือตัวอักษร เรียงลำดับจากน้อยไปมากหรือมากไปหาน้อย (เป็นลักษณะข้อมูลที่เกี่ยวเนื่องกันเป็นชุด) มีรูปแบบ ดังนี้

รูปแบบ

```
array range (mixed $start, mixed $limit [, number $step = 1])
```

เมื่อ \$start หมายถึง ค่าเริ่มต้นของลำดับ

\$limit หมายถึง ค่าสุดท้ายของลำดับ

\$step หมายถึง ค่าความต่างของข้อมูล จะกำหนดหรือไม่ก็ได้ ถ้าไม่กำหนดค่าจะเพิ่มครั้งละ 1 ค่า

ตัวอย่างที่ 6.5 การสร้างอาร์เรย์โดยใช้ฟังก์ชัน range ()

```

1 <?php
2     foreach (range (0, 12) as $number) echo $number . " ";
3     foreach (range (0, 100, 10) as $number) echo $number . " ";
4     foreach (range ('a', 'i') as $character) echo $character . " ";
5     foreach (range ('i', 'a') as $number) echo $number . " ";
6 ?>

```

จากตัวอย่างที่ 6.5 การใช้ฟังก์ชัน range () อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$number เป็นช่วงของเลข โดยใช้ฟังก์ชัน range () กำหนดพารามิเตอร์ค่าเริ่มต้นของลำดับ คือ 0 และค่าสุดท้าย คือ 12 ร่วมกับโครงสร้างเงื่อนไขทำซ้ำ foreach ผลลัพธ์ที่แสดงผล คือ 0 1 2 3 4 5 6 7 8 9 10 11 12

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$number เป็นช่วงของเลข โดยใช้ฟังก์ชัน range () กำหนดพารามิเตอร์ค่าเริ่มต้นของลำดับ คือ 0 และค่าสุดท้าย คือ 100 ค่าความต่างของข้อมูล คือ 10 ร่วมกับโครงสร้างเงื่อนไขทำซ้ำ foreach ผลลัพธ์ที่แสดงผล คือ 0 10 20 30 40 50 60 70 80 90 100

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$character เป็นช่วงของตัวอักษร โดยใช้ฟังก์ชัน range () กำหนดพารามิเตอร์ค่าเริ่มต้นของลำดับ คือ 'a' และค่าสุดท้าย คือ 'i' ร่วมกับโครงสร้างเงื่อนไขทำซ้ำ foreach ผลลัพธ์ที่แสดงผล คือ a b c d e f g h i

บรรทัดที่ 5 กำหนดค่าให้กับตัวแปร \$character เป็นช่วงของตัวอักษร โดยใช้ฟังก์ชัน range () กำหนดพารามิเตอร์ค่าเริ่มต้นของลำดับ คือ 'i' และค่าสุดท้าย คือ 'a' ร่วมกับโครงสร้างเงื่อนไขทำซ้ำ foreach ผลลัพธ์ที่แสดงผล คือ i h g f e d c b a

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

6.1.3 การสร้างอาร์เรย์โดยใช้เครื่องหมายวงเล็บกัมปู

การสร้างอาร์เรย์ด้วยวิธีนี้ เป็นการกำหนดค่าให้กับอาร์เรย์โดยตรง โดยสร้างครั้งละ 1 สมาชิก นอกจักนี้ยังสามารถนำไปใช้ในการเพิ่มสมาชิกให้กับอาร์เรย์ที่มีอยู่แล้วก็ได้

รูปแบบ

```
array array_name [Key] = value;
```

เมื่อ array_name หมายถึง ชื่อตัวแปรอาร์เรย์

Key หมายถึง อินเด็กซ์ของตัวแปรอาร์เรย์ (จะกำหนดหรือไม่ก็ได้)

value หมายถึง ค่าของข้อมูลที่จะกำหนดให้ตัวแปรอาร์เรย์

ตัวอย่างที่ 6.6 การสร้างอาร์เรย์โดยใช้เครื่องหมายวงเล็บกัมปู []

```
1 <?php
2     $country [1] = "Thailand";
3     $country [A] = "Japan";
4     $country [ ] = "Cambodia";
5     print_r ($country);
6 ?>
```

ผลลัพธ์

```
Array ( [1] => Thailand [A] => Japan [2] => Cambodia )
```

จากตัวอย่างที่ 6.6 การใช้ฟังก์ชัน range () อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$country อินเด็กซ์ 1 มีค่าเท่ากับ "Thailand"

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$country คีย์ A มีค่าเท่ากับ "Japan"



บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$country แบบไม่กำหนดอินเด็กซ์หรือคีย์ มีค่าเท่ากับ "Cambodia" (การเพิ่มสมาชิกในอาร์เรย์แบบไม่กำหนดอินเด็กซ์หรือคีย์ ต่อจากสมาชิกเดิมที่มีอยู่ ก่อนหน้านั้น จะตรวจสอบหมายเลขอินเด็กซ์ลำดับก่อนหน้ามีการระบุหมายเลขอะไร แล้วเพิ่มค่าอีก 1 มากำหนดเป็นอินเด็กซ์ให้กับสมาชิกใหม่)

บรรทัดที่ 5 แสดงผลข้อมูลอาร์เรย์ ที่จัดเก็บในตัวแปร \$country

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

6.1.4 การสร้างอาร์เรย์โดยนำค่ามาจากไฟล์

การสร้างอาร์เรย์โดยนำค่ามาจากไฟล์ เป็นวิธีการดึงข้อมูลจากไฟล์มาเก็บไว้ในตัวแปรชนิดอาร์เรย์ ซึ่งข้อมูลแต่ละบรรทัดในไฟล์ ก็คือ ค่าข้อมูลของสมาชิกแต่ละตัวในอาร์เรย์

ตัวอย่าง ข้อมูลในไฟล์ text.txt กำหนดชื่อ text.txt

Suratthani

Chumphon

Ranong

ตัวอย่างที่ 6.7 การสร้างอาร์เรย์โดยนำค่ามาจากไฟล์

```

1 <?php
2     $txt_file = file ("text.txt");
3     $count_arr = count ($txt_file);
4     if ($count_arr == 0) {
5         echo "ไม่มีข้อมูลในไฟล์ <br>";
6     } else {
7         print_r($txt_file);
8     }
9 ?>

```

ผลลัพธ์

Array ([0] => Suratthani [1] => Chumphon [2] => Ranong)

จากตัวอย่างที่ 6.7 การสร้างอาร์เรย์โดยนำค่ามาจากไฟล์ หรือバイตังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 เรียกใช้ฟังก์ชัน file () เพื่ออ่านค่าข้อมูลทั้งหมดจากไฟล์ที่ระบุ แล้วนำมากำหนดค่าให้กับตัวแปร \$txt_file (เป็นข้อมูลชนิดอาร์เรย์โดยปริยาย)

บรรทัดที่ 3 เรียกใช้ฟังก์ชัน count () เพื่อนับจำนวนสมาชิกของอาร์เรย์ (ข้อมูลเท็กซ์ไฟล์ 1 บรรทัด จะมีค่าเท่ากับ 1 สมาชิกอาร์เรย์) และนำผลของการนับจำนวนสมาชิกมากำหนดให้กับตัวแปร \$count_txt

บรรทัดที่ 4 ใช้โครงสร้างเงื่อนไข if ตรวจสอบค่าของตัวแปร \$count_arr มีค่าเท่ากับ 0 จริงหรือไม่ (หากมีค่าเท่ากับ 0 แสดงว่าไม่มีข้อมูล) ถ้าเป็นจริงไปบรรทัดที่ 5 ถ้าเป็นเท็จไปบรรทัดที่ 6

บรรทัดที่ 5 แสดงข้อความ ไม่มีข้อมูลในไฟล์

บรรทัดที่ 6 ปิดโครงสร้างเงื่อนไข if เข้าสู่โครงสร้างเงื่อนไข else (กรณีจำนวนอาร์เรย์มากกว่า 0 หมายถึง มีข้อมูลที่สามารถอ่านได้จากเท็กซ์ไฟล์)

บรรทัดที่ 7 ใช้ฟังก์ชัน print_f เพื่อแสดงผลข้อมูลชนิดอาร์เรย์ \$txt_file

บรรทัดที่ 8 ปิดโครงสร้างเงื่อนไข else

บรรทัดที่ 9 สิ้นสุดสคริปต์ PHP

6.2 การเข้าถึงข้อมูลภายในอาร์เรย์

การเข้าถึงข้อมูลภายในอาร์เรย์มี 4 วิธี คือ 1) การอ้างตำแหน่งของอินเด็กซ์หรือคีย์ 2) การใช้โครงสร้างเงื่อนไขทำซ้ำ 3) การใช้ฟังก์ชัน each () และ 4) การใช้ฟังก์ชัน list () มีรายละเอียดดังนี้

6.2.1 การอ้างตำแหน่งของอินเด็กซ์หรือคีย์

ตัวแปรที่ไว้จะใช้เฉพาะชื่อของตัวแปรเพื่ออ้างถึงค่าในตัวแปร แต่สำหรับอาร์เรย์จะใช้ชื่อของอาร์เรย์และอินเด็กซ์หรือคีย์ในการอ้างถึงค่าในอาร์เรย์ที่ต้องการนำไปประมวลผล ค่าของอินเด็กซ์หรือคีย์ที่ต้องการอ้างถึงจะกำหนดในเครื่องหมายวงเล็บกัมมปู ชี้อยู่หัวลงชื่อตัวแปรอาร์เรย์ ตัวอย่างเช่น \$province [0] เป็นการอ้างถึงสมาชิกของอาร์เรย์อินเด็กซ์ลำดับที่ 0 เป็นต้น นอกจากนี้ค่าที่อยู่ในอาร์เรย์ยังสามารถเปลี่ยนแปลงได้โดยใช้รูปแบบกำหนดค่า (เครื่องหมาย =) เมื่ອนการกำหนดค่าให้กับตัวแปรทั่วไป

6.2.2 การใช้โครงสร้างเงื่อนไขทำซ้ำ

การเข้าถึงข้อมูลในอาร์เรย์ด้วยการใช้โครงสร้างเงื่อนไขทำซ้ำ เช่น for, while และ foreach เป็นต้น จะใช้ได้เฉพาะอาร์เรย์ที่มีอินเด็กซ์ที่เป็นตัวเลขที่มีการเรียงลำดับแล้ว ตัวอย่างการใช้งาน ดังนี้

ตัวอย่างที่ 6.8 การใช้โครงสร้างเงื่อนไขทำซ้ำเพื่อควบคุมการแสดงผลของอาร์เรย์

```

1 <?php
2     $province = array ("สุราษฎร์ธานี", "กระบี่", "ชุมพร", "นครศรีธรรมราช");
3     $count_province = count ($province);
4     for ($loop = 0; $loop < $count_province; $loop++) {

```



```

5         printf ("ลำดับที่ %d คือ %s <br>", $loop+1, $province [$loop]);
6     }
7 ?>

```

ผลลัพธ์

```

ลำดับที่ 1 คือ สุราษฎร์ธานี
ลำดับที่ 2 คือ กระบี่
ลำดับที่ 3 คือ ชุมพร
ลำดับที่ 4 คือ นครศรีธรรมราช

```

จากตัวอย่างที่ 6.8 การใช้โครงสร้างเงื่อนไขทำซ้ำ for เพื่อควบคุมการแสดงผลของอาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$province เป็นตัวแปรชนิดอาร์เรย์ ประกอบด้วย "สุราษฎร์ธานี", "กระบี่", "ชุมพร", "นครศรีธรรมราช" ตามลำดับ

บรรทัดที่ 3 ใช้ฟังก์ชัน count () นับจำนวนสมาชิกอาร์เรย์ในตัวแปร \$province และกำหนดค่าผลจำนวนสมาชิกให้กับตัวแปร \$count_province

บรรทัดที่ 4 ใช้โครงสร้างเงื่อนไขทำซ้ำ for วนรอบเพื่ออ่านและแสดงค่าข้อมูลสมาชิก อาร์เรย์แต่ละตัวตามลำดับ

บรรทัดที่ 5 แสดงค่าข้อมูลสมาชิกอาร์เรย์แต่ละตัวตามลำดับ

บรรทัดที่ 6 ปิดโครงสร้างเงื่อนไขทำซ้ำ for

บรรทัดที่ 7 สิ้นสุดสคริปต์ PHP

6.2.3 การใช้ฟังก์ชัน each ()

ฟังก์ชัน each () เป็นฟังก์ชันที่ใช้อ่านค่าอาร์เรย์ โดยจะส่งคืนค่าเป็นคีย์ และค่าข้อมูล หากอ่านค่าในอาร์เรย์จนหมดแล้ว จะส่งคืนค่าเป็น false

รูปแบบ

```
array each ( array &$array )
```

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่ต้องการอ่านค่า

ตัวอย่างที่ 6.9 การใช้ฟังก์ชัน each () สำหรับอ่านค่าอาร์เรย์

```

1 <?php
2 $province = array ("Surat" => "สุราษฎร์ธานี", "Krabi" => "กระบี่", "Chumphon" =>
"ชุมพร", "Nakhon" => "นครศรีธรรมราช");
3 while ($element = each ($province)) {

```

```

4         printf ("คีย์ที่ %s มีค่าเท่ากับ %s <br>",$element ["key"], $element ["value"]);
5     }
6 ?>

```

ผลลัพธ์

คีย์ Surat มีค่าเท่ากับ สุราษฎร์ธานี

คีย์ Krabi มีค่าเท่ากับ กระบี่

คีย์ Chumphon มีค่าเท่ากับ ชุมพร

คีย์ Nakhon มีค่าเท่ากับ นครศรีธรรมราช

6.2.4 การใช้ฟังก์ชัน list ()

ฟังก์ชัน list () ทำงานในลักษณะเดียวกับฟังก์ชัน each () แต่ต่างกันที่ฟังก์ชัน list ()

ไม่มีการส่งคืนค่า และจะแยกอาร์เรย์ออกจากกัน แล้วเก็บค่าของสมาชิกแต่ละตัวไว้ในตัวแปรที่กำหนดไว้

รูปแบบ

```
array array_values ( array $input )
```

เมื่อ \$input หมายถึง ตัวแปรอาร์เรย์ที่ต้องการอ่านค่า

ตัวอย่างที่ 6.10 การใช้ฟังก์ชัน list () สำหรับอ่านค่าอาร์เรย์

```

1 <?php
2     $data_set = array ("A1001", "ปริญญา", "น้อยดอนไพร", "อาจารย์", "42000");
3     list ($code, $name, $lastname, $occupation, $salary) = $data_set;
4     echo "รหัสพนักงาน: $code ";
5     echo "ชื่อ-สกุล: $name $lastname ";
6     echo "อาชีพ: $occupation เงินเดือน: $salary";
7 ?>

```

ผลลัพธ์

รหัสพนักงาน: A1001

ชื่อ-สกุล: ปริญญา น้อยดอนไพร

อาชีพ: อาจารย์ เงินเดือน: 42000

6.3 ฟังก์ชันอื่นๆ ที่เกี่ยวกับอาร์เรย์

สำหรับใน PHP จะมีฟังก์ชันที่เกี่ยวข้องกับอาร์เรย์อยู่จำนวนมาก แต่บางฟังก์ชันก็อาจไม่ค่อยได้ใช้งานบ่อยนัก หรืออาจใช้ฟังก์ชันอื่นทดแทนกันได้ ดังนั้นในส่วนนี้จะเน้นเฉพาะฟังก์ชันหลักๆ ที่สำคัญมากจะ



ใช้บ่อยครั้ง ทั้งนี้เพื่อความสะดวกต่อการทำความเข้าใจ จะแยกอธิบายตามความเกี่ยวข้องของฟังก์ชัน ต่างๆ ดังนี้

6.3.1 ฟังก์ชันในการอ่านค่าและคีย์ข้อมูลชนิดอาร์เรย์

1) การใช้ฟังก์ชัน array_keys ()

ฟังก์ชัน array_keys () ใช้สำหรับอ่านค่าคีย์ทั้งหมดของตัวแปรอาร์เรย์ โดยผลลัพธ์จะเป็นอาร์เรย์ของคีย์ที่อ่านได้ทั้งหมด

รูปแบบ

```
array array_keys ( array $input [, mixed $search_value = NULL [, bool $strict = false ]] )
```

เมื่อ \$input หมายถึง ตัวแปรอาร์เรย์ที่ต้องการอ่านค่าคีย์

\$search_value หมายถึง อ่านค่าคีย์เฉพาะค่าของอาร์เรย์ที่ตรงกับค่าที่กำหนด (จะกำหนดหรือไม่ก็ได้)

\$strict หมายถึง ใช้สำหรับการเปรียบเทียบ (==) ใช้ร่วมกับ \$search_value เพื่อเปรียบเทียบค่าของอาร์เรย์กับค่าที่เปรียบเทียบ

ตัวอย่างที่ 6.11 การใช้ฟังก์ชัน array_keys () สำหรับอ่านค่าคีย์ทั้งหมดของตัวแปรอาร์เรย์

```
1 <?php
2     $array = array (0 => 100, "color" => "red");
3     print_r (array_keys ($array)); echo "<br>";
4     $array = array ("blue", "red", "green", "blue", "blue");
5     print_r (array_keys ($array, "blue")); echo "<br>";
6 ?>
```

ผลลัพธ์

```
Array ( [0] => 0 [1] => color )
Array ( [0] => 0 [1] => 3 [2] => 4 )
```

จากตัวอย่างที่ 6.11 การใช้ฟังก์ชัน array_keys () สำหรับอ่านค่าคีย์ทั้งหมดของตัวแปรอาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$array เป็นตัวแปรชนิดอาร์เรย์ มีค่าตามลำดับดังนี้ \$array [0] มีค่าเท่ากับ 100 และ \$array ["color"] มีค่าเท่ากับ "red"

บรรทัดที่ 3 แสดงผลลัพธ์เป็นอาร์เรย์ของคีย์ที่อ่านได้ทั้งหมด ของตัวแปร \$array คือ Array ([0] => 0 [1] => color) หมายความว่า คีย์ที่อ่านได้ ประกอบด้วย 0 และ color

บรรทัดที่ 4 กำหนดให้ตัวแปร \$array เป็นตัวแปรชนิดอาร์เรย์ มีค่าตามลำดับดังนี้

\$array [0] มีค่าเท่ากับ "blue"

\$array [1] มีค่าเท่ากับ "red"

\$array [2] มีค่าเท่ากับ "green"

\$array [3] มีค่าเท่ากับ "blue"

\$array [4] มีค่าเท่ากับ "blue"

บรรทัดที่ 5 แสดงผลลัพธ์เป็นอาร์เรย์ของคีย์ที่ตรงกับค่าว่า "blue" ของตัวแปร อาร์เรย์ \$array ผลลัพธ์ที่แสดง คือ Array ([0] => 0 [1] => 3 [2] => 4) หมายความว่า คีย์ที่ตรงกับค่า ว่า "blue" อยู่ในลำดับที่ 0, 3 และ 4

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

2) การใช้ฟังก์ชัน array_values ()

ฟังก์ชัน array_values () ใช้สำหรับอ่านค่าของสมาชิกในอาร์เรย์ โดยผลลัพธ์จะเป็น อาร์เรย์ของค่าที่อ่านได้ทั้งหมด

รูปแบบ

```
array array_values ( array $input )
```

เมื่อ \$input หมายถึง ตัวแปรอาร์เรย์ที่ต้องการอ่านค่า

ตัวอย่างที่ 6.12 การใช้ฟังก์ชัน array_values () สำหรับอ่านค่าของสมาชิกในอาร์เรย์

```
1 <?php
2     $array = array ("size" => "XL", "color" => "gold");
3     print_r (array_values ($array));
4 ?>
```

ผลลัพธ์

```
Array ( [0] => XL [1] => gold )
```

จากตัวอย่างที่ 6.12 การใช้ฟังก์ชัน array_values () สำหรับอ่านค่าของสมาชิกใน อาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$array เป็นตัวแปรชนิดอาร์เรย์ มีค่าตามลำดับ ดังนี้

\$array ["size"] มีค่าเท่ากับ "XL"

\$array ["color"] มีค่าเท่ากับ "gold"

บรรทัดที่ 3 แสดงผลลัพธ์เป็นอาร์เรย์ของค่าที่อ่านได้จากสมาชิกอาร์เรย์ทั้งหมด ของ ตัวแปร \$array คือ Array ([0] => XL [1] => gold) หมายความว่า ไม่อ่านค่าคีย์ของอาร์เรย์ อ่านเฉพาะค่าของสมาชิกแต่ละตัวในอาร์เรย์



บรรทัดที่ 4 สีนสูดสคริปต์ PHP



3) การใช้ฟังก์ชัน array_unique()

ฟังก์ชัน array_unique() ใช้สำหรับอ่านค่าของสมาชิกในอาร์เรย์ โดยเลือกเฉพาะข้อมูลที่ไม่ซ้ำกัน (ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่สำหรับภาษาอังกฤษถือว่าไม่เหมือนกัน)

รูปแบบ

```
array array_unique ( array $array [, int $sort_flags = SORT_STRING ] )
```

เมื่อ \$array	หมายถึง ตัวแปรอาร์เรย์ที่ต้องการอ่านค่า
\$sort_flags	หมายถึง พารามิเตอร์ที่ใช้ร่วมกับการจัดเรียงตามชนิด โดยจะต้องกำหนดร่วมกับค่า SORT_STRING ของการจัดเรียง (จะกำหนดหรือไม่ก็ได้) ดังนี้
SORT_REGULAR	หมายถึง จัดเรียงโดยการเปรียบเทียบตามปกติ (ไม่มีการเปลี่ยนชนิด)
SORT_NUMERIC	หมายถึง จัดเรียงโดยการเปรียบเทียบระหว่างค่าที่เป็นชนิดตัวเลข
SORT_STRING	หมายถึง จัดเรียงโดยการเปรียบเทียบระหว่างค่าที่เป็นชนิดตัวอักษร หรือข้อความ

ตัวอย่างที่ 6.13 การใช้ฟังก์ชัน array_unique() สำหรับอ่านค่าของสมาชิกในอาร์เรย์ที่ไม่ซ้ำกัน

```
1 <?php
2     $input = array ("a" => "green", "red", "b" => "green", "blue", "red");
3     $result = array_unique ($input);
4     print_r ($result);
5 ?>
```

ผลลัพธ์

```
Array ( [a] => green [0] => red [1] => blue )
```

6.3.2 ฟังก์ชันในการเพิ่มสมาชิกในอาร์เรย์

1) การใช้ฟังก์ชัน array_push()

ฟังก์ชัน array_push() ใช้สำหรับเพิ่มสมาชิกลงในอาร์เรย์ โดยข้อมูลที่จะเพิ่มลงไปจะมีจำนวนเท่าไหร่ก็ได้ หรือจะนำมาจากอาร์เรย์อื่นก็ได้

รูปแบบ

```
int array_push ( array &$array , mixed $var [, mixed $... ] )
```

เมื่อ \$array	หมายถึง ตัวแปรอาร์เรย์ที่จะนำข้อมูลเพิ่ม
\$var	หมายถึง ข้อมูลที่ต้องการเพิ่ม



ตัวอย่างที่ 6.14 พิมพ์ฟังก์ชัน array_push() สำหรับเพิ่มสมาชิกลงในอาร์เรย์

```

1 <?php
2     $stack = array ("orange", "banana");
3     array_push ($stack, "apple", "raspberry");
4     print_r ($stack);
5 ?>

```

ผลลัพธ์

```
Array ( [0] => orange [1] => banana [2] => apple [3] => raspberry )
```

2) การใช้ฟังก์ชัน array_pad()

ฟังก์ชัน array_pad() ใช้สำหรับขยายสมาชิกเพิ่มให้กับอาร์เรย์

รูปแบบ

```
array array_pad ( array $input , int $pad_size , mixed $pad_value )
```

เมื่อ \$input หมายถึง ตัวแปรอาร์เรย์ที่จะขยายเพิ่มสมาชิก

\$pad_size หมายถึง ขนาดใหม่ของตัวแปรอาร์เรย์

\$pad_value หมายถึง ข้อมูลที่จะเพิ่ม

ตัวอย่างที่ 6.15 พิมพ์ฟังก์ชัน array_pad() สำหรับขยายสมาชิกเพิ่มให้กับอาร์เรย์

```

1 <?php
2     $input = array (12, 10, 9);
3     $result = array_pad ($input, 5, 0);
4     $result = array_pad ($input, -7, -1);
5     $result = array_pad ($input, 2, "noop");
6 ?>

```

จากตัวอย่างที่ 6.15 ฟังก์ชัน array_pad() สำหรับขยายสมาชิกเพิ่มให้กับอาร์เรย์
อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$input เป็นตัวแปรชนิดอาร์เรย์ มีค่า 12, 10 และ 9

ตามลำดับ

บรรทัดที่ 3 กำหนดให้ตัวแปร \$result เป็นตัวแปรชนิดอาร์เรย์ มีค่าเท่ากับ array (12, 10, 9, 0, 0) เพราะผลของการเรียกใช้ฟังก์ชัน array_pad (\$input, 5, 0) โดยพารามิเตอร์ 5 หมายความว่า ขยายขนาดใหม่ของตัวแปรอาร์เรย์ทางด้านขวา (ขยายเพิ่มหลังจากค่าในตัวแปร \$input) และ 0 หมายความว่า ข้อมูลที่จะกำหนดให้สมาชิกใหม่มีค่าเท่ากับ 0

บรรทัดที่ 4 กำหนดให้ตัวแปร \$result เป็นตัวแปรชนิดอาร์เรย์ มีค่าเท่ากับ array (-1, -1, -1, -1, 12, 10, 9) เพราะผลของการเรียกใช้ฟังก์ชัน array_pad (\$input, -7, -1) โดยพารามิเตอร์ -7 หมายความว่า ขยายขนาดใหม่ของตัวแปรอาร์เรย์ทางด้านซ้าย (ขยายเพิ่มก่อนค่าตัวแปร \$input) และ -1 หมายความว่า ข้อมูลที่จะกำหนดให้สมาชิกใหม่มีค่าเท่ากับ -1

บรรทัดที่ 5 ไม่สามารถกำหนดค่าให้กับตัวแปร \$result ได้ เนื่องจากไม่สามารถขยาย (not padded) เพาะกำหนดพารามิเตอร์ \$pad_size มีขนาดน้อยกว่า \$input

บรรทัดที่ 6 สินสุดสคริปต์ PHP

3) การใช้ฟังก์ชัน array_unshift ()

ฟังก์ชัน array_unshift () ใช้สำหรับเพิ่มสมาชิกในตำแหน่งแรกของตัวแปรอาร์เรย์

รูปแบบ

```
int array_unshift ( array &$array , mixed $var [, mixed $... ] )
```

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่จะเพิ่มสมาชิก

\$var หมายถึง ข้อมูลหรือค่าที่จะเพิ่ม

ตัวอย่างที่ 6.16 ฟังก์ชัน array_unshift () สำหรับเพิ่มสมาชิกตำแหน่งแรกของอาร์เรย์

```
1 <?php
2     $queue = array ("orange", "banana");
3     array_unshift ($queue, "apple", "raspberry");
4     print_r ($queue);
5 ?>
```

ผลลัพธ์

```
Array ( [0] => apple [1] => raspberry [2] => orange [3] => banana )
```

4) การใช้ฟังก์ชัน array_slice ()

ฟังก์ชัน array_slice () ใช้สำหรับลบ แทนที่ และเพิ่ม สมาชิกในตำแหน่งใดๆ ของอาร์เรย์

รูปแบบ

```
array array_splice ( array &$input , int $offset [, int $length = 0 [, mixed $replacement ]] )
```

เมื่อ \$input หมายถึง ตัวแปรอาร์เรย์ที่จะนำข้อมูลเพิ่ม

\$length หมายถึง จำนวนสมาชิกที่จะลบ (จะกำหนดหรือไม่ก็ได้) แบ่งเป็น 3 กรณี

คือ 1) หากกำหนด จะลบสมาชิกตามจำนวนที่กำหนด โดยเริ่ม

จากตำแหน่ง offset ที่ระบุ 2) หากไม่กำหนด จะลบสมาชิก

ตั้งแต่ตำแหน่ง offset ที่ระบุจนถึงสมาชิกสุดท้ายของอาร์เรย์



3) หากกำหนดเป็น 0 คือ ไม่ต้องการลบสมาชิกออก แต่ต้องการเพิ่มสมาชิกลงในตำแหน่ง offset ที่ระบุ แต่ต้องกำหนด replacement ด้วย

\$replacement หมายถึง ข้อมูลที่จะเพิ่มลงไปยังตำแหน่ง offset ที่ระบุ

ตัวอย่างที่ 6.17 พัฒนา array_slice() สำหรับลบ แทนที่ และเพิ่ม สมาชิกในตำแหน่งใดๆ ของอาร์เรย์

```

1 <?php
2     $input = array ("red", "green", "blue", "yellow");
3     array_splice ($input, 2);
4     print_r ($input);
5 ?>

```

ผลลัพธ์

```
Array ( [0] => red [1] => green )
```

6.3.3 พัฒนาในการลบสมาชิกในอาร์เรย์

1) การใช้พัฒนา unset()

พัฒนา unset() ใช้สำหรับลบหรือยกเลิกตัวแปรที่ไปหรือตัวแปรชนิดอาร์เรย์

รูปแบบ

```
void unset ( mixed $var [, mixed $... ] )
```

เมื่อ \$var หมายถึง ตัวแปรอาร์เรย์ที่ต้องการลบ

ตัวอย่างที่ 6.18 พัฒนา unset() สำหรับลบหรือยกเลิกตัวแปรที่ไปหรือตัวแปรชนิดอาร์เรย์

```

1 <?php
2     $stack = array ("orange", "banana");
3     unset ($stack [1]);
4     print_r ($stack);
5 ?>

```

ผลลัพธ์

```
Array ( [0] => orange )
```

2) การใช้พัฒนา array_pop()

พัฒนา array_pop() ใช้สำหรับลบสมาชิกตำแหน่งสุดท้ายของตัวแปรชนิดอาร์เรย์

หากตัวแปรเป็นค่าว่างหรือตัวแปรที่ส่งเข้าไปไม่ใช่ตัวแปรชนิดอาร์เรย์ พัฒนาจะส่งค่ากลับเป็น Null หากสามารถลบได้จะส่งค่ากลับเป็นค่าลำดับสุดท้ายของสมาชิกอาร์เรย์

รูปแบบ

mixed array_pop (array &\$array)

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่ต้องการลบ

ตัวอย่างที่ 6.19 พังก์ชัน array_pop () สำหรับลบสมาชิกตำแหน่งสุดท้ายของตัวแปรอาร์เรย์

```

1  <?php
2      $stack= array ("orange", "banana", "apple");
3      array_pop ($stack);
4      print_r ($stack);
5  ?>

```

ผลลัพธ์

Array ([0] => orange [1] => banana)

3) การใช้ฟังก์ชัน array_shift ()

ฟังก์ชัน array_shift () ใช้สำหรับลบสมาชิกตำแหน่งแรกของตัวแปรชนิดอาร์เรย์ และ
ย้ายลำดับของสมาชิกใหม่โดยเริ่มต้นจาก 0

รูปแบบ

mixed array_shift (array &\$array)

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่ต้องการลบ

ตัวอย่างที่ 6.20 ฟังก์ชัน array_shift () สำหรับลบสมาชิกตำแหน่งแรกของตัวแปรชนิดอาร์เรย์

```

1  <?php
2      $stack = array ("orange", "banana", "apple");
3      array_shift ($stack);
4      print_r ($stack);
5  ?>

```

ผลลัพธ์

Array ([0] => banana [1] => apple)

6.3.4 ฟังก์ชันในการนับจำนวนสมาชิกในอาร์เรย์

1) การใช้ฟังก์ชัน count ()

ฟังก์ชัน count () ใช้สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์

รูปแบบ

int count (mixed \$var [, int \$mode = COUNT_NORMAL])



เมื่อ \$var หมายถึง ตัวแปรอาร์เรย์ที่ต้องการนับจำนวนสมาชิก
 \$mode หมายถึง พารามิเตอร์เสริมสำหรับนับอาร์เรย์แบบ COUNT_RECURSIVE
 หมายเหตุ

ฟังก์ชัน count () จะมีการส่งค่ากลับเป็นเลขจำนวนสมาชิกทั้งหมดในอาร์เรย์ ในกรณีที่ตัวแปรที่ส่งเข้ามาไม่ใช่ตัวแปรชนิดอาร์เรย์ ฟังก์ชัน count () จะส่งค่ากลับเป็น 1 และในกรณีที่ตัวแปรอาร์เรย์เป็นค่าว่างจะส่งค่ากลับเป็น 0

ตัวอย่างที่ 6.21 การใช้ฟังก์ชัน count () สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์

```

1 <?php
2     $a = array ("1","3","6");
3     printf("%d<br>", count($a));
4     printf("%d<br>", count(null));
5     printf("%d<br>", count(false));
6
  ?>
```

จากตัวอย่างที่ 6.21 การใช้ฟังก์ชัน count () สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$a เป็นข้อมูลชนิดอาร์เรย์ ประกอบด้วย 1, 3 และ 6

บรรทัดที่ 3 แสดงค่าผลการนับจำนวนสมาชิกอาร์เรย์ของตัวแปร \$a ผลลัพธ์ที่ได้ คือ 3

บรรทัดที่ 4 แสดงค่าผลการนับจำนวนสมาชิกอาร์เรย์ ที่มีค่า null ผลลัพธ์ที่ได้ คือ 0

บรรทัดที่ 5 แสดงค่าผลการนับจำนวนสมาชิกอาร์เรย์ ที่มีค่า false ผลลัพธ์ที่ได้ คือ 1

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

2) การใช้ฟังก์ชัน sizeof()

ฟังก์ชัน sizeof () ใช้สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์ เช่นเดียวกับฟังก์ชัน count ()

รูปแบบ

```
int sizeof ( mixed $var )
```

เมื่อ \$var หมายถึง ตัวแปรอาร์เรย์ที่ต้องการนับจำนวนสมาชิก

ตัวอย่างที่ 6.22 การใช้ฟังก์ชัน sizeof () สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์

```

1 <?php
2     $a = array ("1","3","6");
3     printf("%d<br>", sizeof ($a));
4
  ?>
```

ผลลัพธ์

4

3) การใช้ฟังก์ชัน array_count_values()

ฟังก์ชัน array_count_values() ใช้สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์ แต่ผลลัพธ์ที่ได้จะเป็นอาร์เรย์ที่แสดงว่า สมาชิกแต่ละตัวมีอยู่จำนวนเท่าไหร่ โดยค่าคีย์จะเป็นสมาชิกแต่ละตัว และค่าจะเป็นจำนวนที่นับได้

รูปแบบ

```
array array_count_values ( array $input )
```

เมื่อ \$input หมายถึง ตัวแปรอาร์เรย์ที่ต้องการนับ

ตัวอย่างที่ 6.23 การใช้ฟังก์ชัน array_count_values() สำหรับนับจำนวนสมาชิกทั้งหมดในอาร์เรย์

```
1 <?php
2     $array = array (1, "hello", 1, "world", "hello");
3     print_r (array_count_values ($array));
4 ?>
```

ผลลัพธ์

```
Array ( [1] => 2 [hello] => 2 [world] => 1 )
```

6.3.5 ฟังก์ชันในการเรียงลำดับข้อมูลในอาร์เรย์โดยใช้อินเด็กซ์

ฟังก์ชันในการเรียงลำดับข้อมูลในอาร์เรย์โดยใช้อินเด็กซ์ จะมี 2 ฟังก์ชันให้เลือกใช้งาน คือ ฟังก์ชัน sort() และ rsort() มีรายละเอียด ดังนี้

1) การใช้ฟังก์ชัน sort()

ฟังก์ชัน sort() ใช้สำหรับเรียงลำดับข้อมูลในอาร์เรย์โดยใช้อินเด็กซ์ เรียงลำดับข้อมูล จากน้อยไปมาก นิรูปแบบ ดังนี้

รูปแบบ

```
bool sort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่ต้องการจัดเรียงและมีอินเด็กซ์เป็นตัวเลข

\$sort_flags หมายถึง พารามิเตอร์เสริมสำหรับการจัดเรียงเพิ่มเติม โดยการระบุสตริง สำหรับการจัดเรียง มีรายละเอียด (จะกำหนดหรือไม่ก็ได้) ดังนี้

- SORT_REGULAR จัดเรียงโดยการเปรียบเทียบระหว่างค่าตามปกติ (ไม่ต้องเปลี่ยนค่าคีย์)
- SORT_NUMERIC จัดเรียงโดยการเปรียบเทียบระหว่างค่าที่เป็นชนิดตัวเลข



- SORT_STRING จัดเรียงโดยการเปรียบเทียบระหว่างค่าที่เป็นตัวอักษรหรือข้อความ
- SORT_NATURAL จัดเรียงโดยการเปรียบเทียบระหว่างค่าที่เป็นตัวอักษรหรือข้อความเหมือนกับฟังก์ชัน natsort() ซึ่งใช้สำหรับการจัดเรียงข้อมูลในอาร์เรย์ที่มีค่าอินเด็กซ์เป็นชนิดตัวอักษรหรือข้อความ
- SORT_FLAG_CASE รูปแบบการจัดเรียงที่สามารถใช้ร่วมกับการจัดเรียงแบบ SORT_STRING หรือ SORT_NATURAL โดยการเปรียบเทียบในลักษณะตัวอักษรพิมพ์เล็กพิมพ์ใหญ่ถือว่าเป็นตัวเดียวกัน (case-insensitively)

ตัวอย่างที่ 6.24 การใช้ฟังก์ชัน sort()

```

1 <?php
2     $fruits = array ("lemon", "banana", "apple");
3     sort ($fruits);
4     foreach ($fruits as $key => $val) echo "fruits[" . $key . "] = " . $val . "<br>";
5 ?>

```

ผลลัพธ์

```

fruits [0] = apple
fruits [1] = banana
fruits [2] = lemon

```

ตัวอย่างที่ 6.25 การใช้ฟังก์ชัน sort() โดยกำหนดการจัดเรียงแบบ case-insensitive natural ordering

```

1 <?php
2     $fruits = array ( "Orange1", "orange2", "Orange3", "orange20" );
3     @sort($fruits, SORT_NATURAL | SORT_FLAG_CASE);
4     foreach ($fruits as $key => $val) echo "fruits[" . $key . "] = " . $val . "<br>";
5 ?>

```

ผลลัพธ์

```

fruits [0] = Orange1
fruits [1] = orange2
fruits [2] = Orange3
fruits [3] = orange20

```

2) การใช้ฟังก์ชัน rsort ()

ฟังก์ชัน rsort () ใช้สำหรับเรียงลำดับข้อมูลในอาร์เรย์โดยใช้อินเด็กซ์ เรียงลำดับข้อมูลจากมากไปหาน้อย มีรูปแบบเหมือนกับฟังก์ชัน sort () มีตัวอย่างดังนี้

ตัวอย่างที่ 6.26 การใช้ฟังก์ชัน rsort ()

```

1 <?php
2     $fruits = array ("lemon", "orange", "banana", "apple");
3     rsort ($fruits);
4     foreach ($fruits as $key => $val) echo "$key = $val";
5

```

ผลลัพธ์

```
0 = orange 1 = lemon 2 = banana 3 = apple
```

6.3.6 ฟังก์ชันที่ใช้ในการเรียงลำดับข้อมูลในอาร์เรย์โดยใช้คีย์อาร์เรย์

ฟังก์ชันในการเรียงลำดับข้อมูลในอาร์เรย์โดยใช้คีย์อาร์เรย์ มี 4 ฟังก์ชันให้เลือกใช้งาน ประกอบด้วย 1) asort () 2) arsort () 3) ksort () และ 4) krsort () มีรายละเอียด ดังนี้

1) การใช้ฟังก์ชัน asort ()

ฟังก์ชัน asort () เป็นฟังก์ชันที่ใช้สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์โดยพิจารณาจากค่าหรือข้อมูลในตัวแปรอาร์เรย์ จากน้อยไปมาก มีรูปแบบการใช้งาน ดังนี้

รูปแบบ

```
bool asort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่ต้องการจัดเรียง

\$sort_flags หมายถึง พารามิเตอร์เสริมสำหรับการจัดเรียงเพิ่มเติม โดยการระบุสตริง สำหรับการจัดเรียง จะมีรูปแบบเหมือนฟังก์ชันจัดเรียงก่อนหน้า

ตัวอย่างที่ 6.27 การใช้ฟังก์ชัน asort () สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์

```

1 <?php
2     $fruits = array("d" => "lemon", "b" => "banana", "c" => "apple");
3     asort ($fruits);
4     foreach ($fruits as $key => $val) echo "$key = $val <br>";
5

```



ผลลัพธ์

```
c = apple
b = banana
d = lemon
```

2) การใช้ฟังก์ชัน arsort ()

ฟังก์ชัน arsort () เป็นฟังก์ชันที่ใช้สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์โดยพิจารณาจากค่าหรือข้อมูลในตัวแปรอาร์เรย์ จำนวนมากไปหน้าอย่าง มีรูปแบบการใช้งานเหมือนกับฟังก์ชัน asort ()

ตัวอย่างที่ 6.28 การใช้ฟังก์ชัน arsort () เป็นฟังก์ชันที่ใช้สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์

```
1 <?php
2     $fruits = array("d" => "lemon", "b" => "banana", "c" => "apple");
3     arsort ($fruits);
4     foreach ($fruits as $key => $val) echo "$key = $val <br>";
5 ?>
```

ผลลัพธ์

```
d = lemon
b = banana
c = apple
```

3) การใช้ฟังก์ชัน ksort ()

ฟังก์ชัน ksort () เป็นฟังก์ชันที่ใช้สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์โดยพิจารณาจากคีย์ของตัวแปรอาร์เรย์ จำนวนมากอย่างมาก มีรูปแบบการใช้งาน ดังนี้

รูปแบบ

```
bool ksort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

เมื่อ \$array หมายถึง ตัวแปรอาร์เรย์ที่ต้องการจัดเรียง

\$sort_flags หมายถึง พารามิเตอร์เสริมสำหรับการจัดเรียงเพิ่มเติม โดยการระบุสตริงสำหรับการจัดเรียง จะมีรูปแบบเหมือนฟังก์ชันจัดเรียงก่อนหน้า

ตัวอย่างที่ 6.29 การใช้ฟังก์ชัน ksort () สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์

```
1 <?php
2     $fruits = array ("d" => "lemon", "b" => "banana", "c" => "apple");
3     ksort ($fruits);
4     foreach ($fruits as $key => $val) echo "$key = $val <br>";
5 ?>
```

ผลลัพธ์

```
b = banana
c = apple
d = lemon
```

4) การใช้ฟังก์ชัน krsort ()

ฟังก์ชัน krsort () เป็นฟังก์ชันที่ใช้สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์โดยพิจารณาจากคีย์ของตัวแปรอาร์เรย์ จำนวนมากไปหน้าย มีรูปแบบเหมือนกับฟังก์ชัน ksort () มีตัวอย่างดังนี้

ตัวอย่างที่ 6.30 การใช้ฟังก์ชัน krsort () สำหรับจัดเรียงลำดับข้อมูลในอาร์เรย์

```
1 <?php
2     $fruits = array ("d" => "lemon", "b" => "banana", "c" => "apple");
3     krsort ($fruits);
4     foreach ($fruits as $key => $val) echo "$key = $val <br>";
5 ?>
```

ผลลัพธ์

```
d = lemon
c = apple
b = banana
```

โดยสรุป ทั้ง 4 ฟังก์ชัน เป็นฟังก์ชันที่ใช้สำหรับเรียงลำดับข้อมูลในอาร์เรย์โดยใช้คีย์อาร์เรย์สามารถจำแนกได้เป็นกลุ่มใหญ่ ดังนี้

ตารางที่ 6.2 แสดงผลสรุปฟังก์ชันที่ใช้สำหรับเรียงลำดับข้อมูลในอาร์เรย์โดยใช้คีย์อาร์เรย์

เรียงจาก	ฟังก์ชัน	ลำดับการเรียง
ข้อมูล (Value)	asort ()	น้อยไปมาก
	arsort ()	มากไปหน้าย
อินเด็กซ์หรือคีย์ (Index or Key)	ksort ()	น้อยไปมาก
	krsort ()	มากไปหน้าย

6.3.7 ฟังก์ชันที่ใช้สำหรับการจัดการตัวชี้ตำแหน่งสมาชิกในอาร์เรย์

เมื่อเพิ่มหรือกำหนดค่าข้อมูลในอาร์เรย์ ตำแหน่งของตัวชี้ตำแหน่งสมาชิกในอาร์เรย์จะซึ้งอยู่ที่ตำแหน่งของสมาชิกปัจจุบัน หากต้องการกำหนดให้ตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ซึ้งไปที่ตำแหน่งต่างๆ ของอาร์เรย์ สามารถทำได้โดยใช้ฟังก์ชันต่างๆ ดังตารางที่ 6.3



ตารางที่ 6.3 สรุปฟังก์ชันที่ใช้สำหรับการจัดการตัวชี้ตำแหน่งสมาชิกในอาร์เรย์

ฟังก์ชัน	การทำงาน
each ()	ฟังก์ชันสำหรับเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ไปข้างหน้าครั้งละ 1 ตำแหน่ง หากเรียกใช้ฟังก์ชัน each () จะคืนค่าสมาชิกก่อนหน้าที่จะเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์
current ()	ฟังก์ชันที่ใช้สำหรับคืนค่าสมาชิกปัจจุบันที่ตัวชี้ตำแหน่งสมาชิกในอาร์เรย์กำลังชี้อยู่ หากก่อนหน้านี้ยังไม่มีการเรียกใช้ฟังก์ชันสำหรับการเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ หลังจากอาร์เรย์ถูกสร้าง ตำแหน่งของตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ปัจจุบัน คือ ตำแหน่งแรกของสมาชิกอาร์เรย์
reset ()	ฟังก์ชันที่ใช้สำหรับเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ไปยังสมาชิกตำแหน่งแรกของอาร์เรย์
end ()	ฟังก์ชันที่ใช้สำหรับเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ไปยังสมาชิกตำแหน่งสุดท้ายของอาร์เรย์
next ()	ฟังก์ชันสำหรับเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ไปข้างหน้าครั้งละ 1 ตำแหน่ง หากเรียกใช้ฟังก์ชัน next () จะเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ก่อนแล้วจึงคืนค่าสมาชิกก่อนหน้า
prev ()	ฟังก์ชันสำหรับเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ตรงข้ามกับฟังก์ชัน next () คือ ถอยกลับครั้งละ 1 ตำแหน่ง หากเรียกใช้ฟังก์ชัน prev () จะเลื่อนตัวชี้ตำแหน่งสมาชิกในอาร์เรย์ก่อนแล้วจึงคืนค่าสมาชิกก่อนหน้า

6.3.8 ฟังก์ชันสำหรับการรวมอาร์เรย์

ฟังก์ชันสำหรับการรวมอาร์เรย์ มีให้เลือกใช้งาน คือ 1) ฟังก์ชัน array_merge ()

2) array_merge_recursive () และ 3) array_combine ()

ทั้ง 2 ฟังก์ชันนี้ใช้สำหรับรวมอาร์เรย์เหมือนกัน หลักการรวมอาร์เรย์ คือ จะนำค่าข้อมูลมาเก็บในลักษณะของสมาชิกต่อ กันไปเรื่อยๆ ในรูปแบบของอาร์เรย์หลายมิติ และสร้างอินเด็กซ์ชนิดตัวเลขให้ใหม่เรียงตามลำดับ ข้อแตกต่างระหว่าง 2 ฟังก์ชันนี้ คือ

1) ฟังก์ชัน array_merge () หากอาร์เรย์ที่นำมารวมมีอินเด็กซ์หรือคีย์ที่เป็นสตริงซ้ำกัน จะนำค่าข้อมูลของอาร์เรย์ชุดหลังทับค่าข้อมูลของอาร์เรย์ชุดแรก

2) ฟังก์ชัน array_merge_recursive () จะนำข้อมูลของอาร์เรย์ชุดหลังมาต่อท้ายกับค่าข้อมูลตัวแรก ไม่ว่าค่าข้อมูลหรืออินเด็กซ์หรือคีย์จะซ้ำกันหรือไม่

รูปแบบ ฟังก์ชัน array_merge ()

```
array array_merge ( array $array1 [, array $... ] )
```

รูปแบบ พังก์ชัน array_merge_recursive ()

```
array array_merge_recursive ( array $array1 [, array $... ] )
```

เมื่อ \$array1 หมายถึง ตัวแปรอาร์เรย์ที่ต้องการนำมารวมลำดับที่ 1

\$... หมายถึง ตัวแปรอาร์เรย์ที่ต้องการนำมารวมลำดับต่อๆ ไป

ตัวอย่างที่ 6.31 การใช้พังก์ชัน array_merge () สำหรับรวมอาร์เรย์

```
1 <?php
2     $array1 = array ("color" => "red", 2, 4);
3     $array2 = array ("a", "b", "color" => "green", "shape" => "trapezoid", 4);
4     $result = array_merge ($array1, $array2);
5     print_r ($result);
6 ?>
```

ผลลัพธ์

หมายเหตุ หากค่าอินเด็กซ์หรือคีย์ซ้ำกันข้อมูลของอาร์เรย์ตัวแรกจะถูกทับด้วยข้อมูลอาร์เรย์ล่าสุด

```
Array ( [color] => green [0] => 2 [1] => 4 [2] => a [3] => b [shape] => trapezoid [4] => 4 )
```

ตัวอย่างที่ 6.32 การใช้พังก์ชัน array_merge_recursive ()

```
1 <?php
2     $array1 = array ("color" => "red", 2, 4);
3     $array2 = array ("a", "b", "color" => "green", "shape" => "trapezoid", 4);
4     $result = array_merge_recursive ($array1, $array2);
5     print_r ($result);
6 ?>
```

ผลลัพธ์

หมายเหตุ หากอินเด็กซ์หรือคีย์ซ้ำกันจะสร้างเป็นอาร์เรย์ช้อนอาร์เรย์ หรือเรียกว่าอาร์เรย์หลายมิติ

```
Array ( [color] => Array ( [0] => red [1] => green ) [0] => 2 [1] => 4 [2] => a [3] => b
[shape] => trapezoid [4] => 4 )
```

3) พังก์ชัน array_combine ()

พังก์ชัน array_combine () เป็นพังก์ชันที่ใช้สำหรับรวมอาร์เรย์ 2 อาร์เรย์เข้าด้วยกัน โดยใช้ค่าข้อมูลของอาร์เรย์ตัวแรกเป็นอินเด็กซ์หรือคีย์ ส่วนค่าข้อมูลของอาร์เรย์ตัวที่สองเป็นค่าข้อมูลของ อาร์เรย์ใหม่ที่เกิดจากการรวมกันของอาร์เรย์ทั้งสอง ถ้าจำนวนของสมาชิกของทั้งสองอาร์เรย์ไม่เท่ากัน หรืออาร์เรย์ตัวใดตัวหนึ่งเป็นค่าว่าง พังก์ชันจะส่งคืนค่าเป็นเท็จ (False)



รูปแบบ

```
array array_combine ( array $keys , array $values )
```

เมื่อ \$key คือ ตัวแปรอาร์เรย์ที่ใช้สำหรับเป็นอินเดกซ์หรือคีย์

\$values คือ ตัวแปรอาร์เรย์ที่ใช้สำหรับกำหนดค่า

ตัวอย่างที่ 6.33 การใช้ฟังก์ชัน array_combine () สำหรับรวมอาร์เรย์ 2 อาร์เรย์เข้าด้วยกัน

```
1 <?php
2     $a = array ('green', 'red', 'yellow');
3     $b = array ('avocado', 'apple', 'banana');
4     $c = array_combine ($a, $b);
5     print_r ($c);
6 ?>
```

ผลลัพธ์

```
Array ( [green] => avocado [red] => apple [yellow] => banana )
```

6.3.9 ฟังก์ชันสำหรับการสลับค่าระหว่างอินเดกซ์หรือคีย์กับค่าข้อมูลในอาร์เรย์

ฟังก์ชันสำหรับการสลับค่าระหว่างอินเดกซ์หรือคีย์กับค่าข้อมูลในอาร์เรย์ คือ ฟังก์ชัน array_flip () มีรูปแบบดังนี้

รูปแบบ

```
array array_flip ( array $trans )
```

เมื่อ \$trans หมายถึง ตัวแปรอาร์เรย์ที่ต้องการสลับค่า

ตัวอย่างที่ 6.34 การใช้ฟังก์ชัน array_flip () สำหรับสลับค่าระหว่างอินเดกซ์หรือคีย์กับค่าข้อมูล

```
1 <?php
2     $trans = array ("a" => 1, "b" => 1, "c" => 2);
3     $trans = array_flip ($trans);
4     print_r ($trans);
5 ?>
```

ผลลัพธ์

```
Array ( [1] => b [2] => c )
```

6.3.10 พังก์ชันสำหรับการค้นหาอินเด็กซ์หรือคีย์และค่าข้อมูลในอาร์เรย์

1) การใช้ฟังก์ชัน array_key_exists ()

ฟังก์ชัน array_key_exists () เป็นฟังก์ชันที่ใช้สำหรับหาอินเด็กซ์หรือคีย์ที่ระบุมีอยู่ในอาร์เรย์หรือไม่ หากพบอินเด็กซ์หรือคีย์ที่ค้นหา ฟังก์ชันจะคืนค่าเป็นจริง (True) หากไม่พบจะคืนค่าเป็นเท็จ (False)

รูปแบบ

```
bool array_key_exists ( mixed $key , array $search )
```

เมื่อ \$key หมายถึง อินเด็กซ์หรือคีย์ที่ค้นหา

\$search หมายถึง อาร์เรย์ที่นำมาใช้ค้นหา

ตัวอย่างที่ 6.35 การใช้ฟังก์ชัน array_key_exists () สำหรับหาอินเด็กซ์หรือคีย์ที่ระบุมีอยู่ในอาร์เรย์หรือไม่

```
1 <?php
2     $search_array = array ('first' => 1, 'second' => 4);
3     if (array_key_exists ('first', $search_array)) echo "พบข้อมูลในอาร์เรย์";
4     else echo "ไม่พบข้อมูลในอาร์เรย์";
5 ?>
```

ผลลัพธ์

```
พบข้อมูลในอาร์เรย์
```

2) การใช้ฟังก์ชัน in_array () และ array_search ()

ฟังก์ชัน in_array () และ array_search () เป็นฟังก์ชันที่ใช้สำหรับหาค่าข้อมูลที่ระบุมีอยู่ในอาร์เรย์หรือไม่ หากฟังก์ชัน in_array () และ array_search () พบรหัสค่าข้อมูลที่ค้นหา ฟังก์ชันจะคืนค่าเป็นจริง (True) หากไม่พบจะคืนค่าเป็นเท็จ (False) ซึ่งทั้ง 2 ฟังก์ชันมีรูปแบบเหมือนกัน ดังนี้

รูปแบบ

```
bool in_array ( mixed $needle , array $haystack [, bool $strict = FALSE ] )
```

เมื่อ \$needle หมายถึง ค่าข้อมูลที่ต้องการค้นหา

\$haystack หมายถึง ตัวแปรอาร์เรย์ที่นำมาใช้ในการค้นหา

\$strict หมายถึง ใช้สำหรับการเปรียบเทียบ (==) ใช้ร่วมกับ \$needle เพื่อเปรียบเทียบค่าของอาร์เรย์กับค่าที่ค้นหา



ตัวอย่างที่ 6.36 การใช้ฟังก์ชัน `in_array()` และ `array_search()` สำหรับหาค่าข้อมูลที่ระบุมีอยู่ในอาร์เรย์หรือไม่

```

1 <?php
2     $os = array ("Mac", "NT", "Irix", "Linux");
3     if (in_array ("Irix", $os)) echo "Got Irix";
4     if (in_array ("mac", $os)) echo "Got mac";
5 ?>

```

ผลลัพธ์

Got Irix

จากตัวอย่างในเงื่อนไขที่ 2 จะพบว่าเงื่อนไขไม่ถูกต้อง เพราะในฟังก์ชัน `in_array()` จะมีการตรวจสอบอักษรพิมพ์เล็กและพิมพ์ใหญ่ (case-sensitive)

6.3.11 ฟังก์ชันสำหรับการหาค่าข้อมูลสมาชิกที่เหมือนและแตกต่างกันในอาร์เรย์

1) การใช้ฟังก์ชัน `array_intersect()`

ฟังก์ชัน `array_intersect()` เป็นฟังก์ชันสำหรับหาค่าข้อมูลสมาชิกในอาร์เรย์ ตั้งแต่ 2 อาร์เรย์ขึ้นไป มาทำการ Intersection หาสมาชิกที่มีช้ากัน เพื่อกำหนดค่าข้อมูลให้กับอาร์เรย์ใหม่

รูปแบบ

```
array array_intersect ( array $array1 , array $array2 [, array $ ... ] )
```

เมื่อ `$array1` หมายถึง ตัวแปรอาร์เรย์ตัวที่ 1

`$array2` หมายถึง ตัวแปรอาร์เรย์ตัวที่ 2

`$...` หมายถึง ตัวแปรอาร์เรย์ลำดับต่อๆ ไป

ตัวอย่างที่ 6.37 การใช้ฟังก์ชัน `array_intersect()`

```

1 <?php
2     $array1 = array ("a" => "green", "red", "blue");
3     $array2 = array ("b" => "green", "yellow", "red");
4     $result = array_intersect ($array1, $array2);
5     print_r ($result);
6 ?>

```

ผลลัพธ์

Array ([a] => green [0] => red)

2) ฟังก์ชัน array_diff()

เป็นฟังก์ชันสำหรับหาค่าข้อมูล sama ซิกในอาร์เรย์ ตั้งแต่ 2 อาร์เรย์ขึ้นไป มาทำการ Differential หาสมาชิกที่มีเฉพาะในอาร์เรย์ตัวแรก แต่ไม่มีในอาร์เรย์อื่นๆ เพื่อกำหนดค่าข้อมูลให้กับ อาร์เรย์ใหม่

รูปแบบ

```
array array_diff ( array $array1 , array $array2 [, array $... ] )
```

เมื่อ \$array1 หมายถึง ตัวแปรอาร์เรย์ตัวที่ 1

\$array2 หมายถึง ตัวแปรอาร์เรย์ตัวที่ 2

\$... หมายถึง ตัวแปรอาร์เรย์ลำดับต่อๆ ไป

ตัวอย่างที่ 6.38 การใช้ฟังก์ชัน array_diff()

```
1 <?php
2     $array1 = array ("a" => "green", "red", "blue", "red");
3     $array2 = array ("b" => "green", "yellow", "red");
4     $result = array_diff ($array1, $array2);
5     print_r ($result);
6 ?>
```

ผลลัพธ์

```
Array ( [1] => blue )
```

6.4 การใช้ข้อมูลจากอาร์เรย์ \$_FILE

ตัวแปร \$_FILE เป็นตัวแปรชนิดอาร์เรย์เพื่อใช้อ้างถึงข้อมูลเกี่ยวกับการอัปโหลดไฟล์ (File Upload) ระหว่างโคลอนต์กับเว็บเชิร์ฟเวอร์ มีรายละเอียด ดังตารางที่ 6.4

ตารางที่ 6.4 แสดงค่าคีย์ของอาร์เรย์ \$_FILE

คีย์	ค่าข้อมูลและความหมาย
\$_FILE ['file'] ['type']	ชนิดของไฟล์ที่อัปโหลด เช่น .JPG, .PNG, .doc เป็นต้น
\$_FILE ['file'] ['size']	ขนาดของไฟล์
\$_FILE ['file'] ['tmp_name']	ตำแหน่ง临时 ไดเรกทอรีที่เก็บไฟล์ไว้ชั่วคราว
\$_FILE ['file'] ['name']	ชื่อไฟล์ที่อัปโหลด



ตารางที่ 6.4 (ต่อ)

คีย์	ค่าข้อมูลและความหมาย
\$_FILE ['file'] ['error']	ข้อมูลที่ผิดพลาดจากการอัปโหลด โดยมีการคืนค่า ดังนี้ 0 คือ แสดงว่าไม่มีข้อผิดพลาด 1 คือ ไฟล์ที่อัปโหลดมีขนาดเกินกว่าค่าที่กำหนดใน php.ini (ปกติ 2 MB) 2 คือ ไฟล์มีขนาดเกินค่าที่กำหนดใน MAX_FILE_SIZE ของฟอร์ม 3 คือ ข้อผิดพลาดในการสื่อสารทำให้อัปโหลดไฟล์ไม่ได้ 4 คือ ไม่มีไฟล์

หลังจากนั้นจะใช้ฟังก์ชัน move_uploaded_file () สำหรับในการเคลื่อนย้ายไฟล์จากไดเรกทอรีชั่วคราวไปยังตำแหน่งใหม่ โดยจะใช้ \$_FILE['file'] ['tmp_name'] ในการอ้างถึงไดเรกทอรีชั่วคราว ส่วนตำแหน่งไดเรกทอรีปลายทาง ถ้าต้องการใช้ชื่อไฟล์เดิมจะใช้ \$_FILE ['file'] ['name'] ถ้าต้องการย้ายไปไดเรกทอรีอื่นก็สามารถระบุได้ ตัวอย่างที่นิยมใช้งานเพื่อไม่ให้เกิดปัญหาชื่อไฟล์ซ้ำกัน คือ ใช้วันที่และเวลาสำหรับกำหนดชื่อไฟล์ มีตัวอย่าง ดังนี้

ตัวอย่างที่ 6.39 การกำหนดชื่อไฟล์โดยใช้วันที่และเวลา

```
$fileName = mktime (date('H'), date('i'), date('s'), date('m'), date('d'), date('Y')).'.jpg';
```

การกำหนดชื่อไฟล์นี้นั้นใช้สำหรับจัดเก็บลงในฐานข้อมูล หรืออ้างอิงอื่นๆ และใช้ฟังก์ชัน

```
move_uploaded_file ($_FILE ["file"] ["tmp_name"] , "../image/news/".$fileName);
```

หรือใช้ฟังก์ชัน

```
copy ($_FILE ["file"] ["tmp_name"] , "../image/news/".$fileName);
```

เพื่อเริ่มต้นอัปโหลดไฟล์

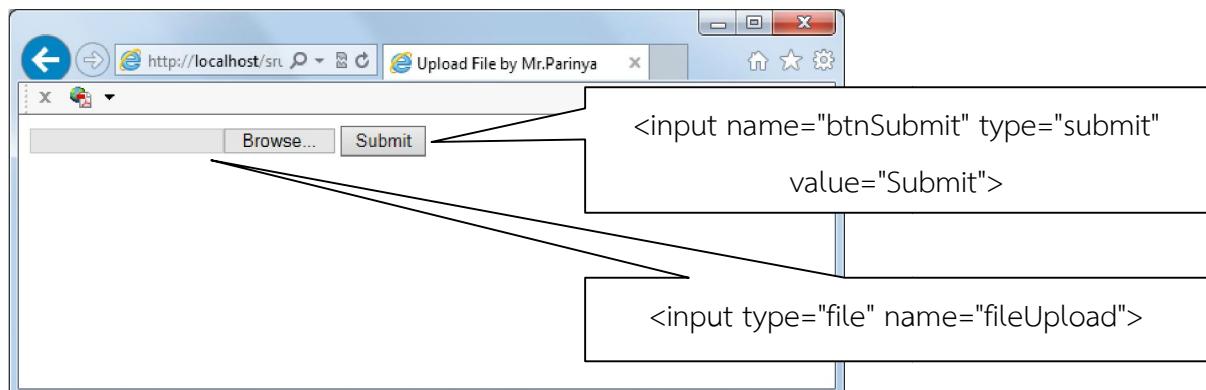
ตัวอย่างที่ 6.40 การประยุกต์พัฒนาเว็บเพจสำหรับอัปโหลดไฟล์

```
<?php /* กำหนดชื่อไฟล์เป็น upload.php*/ ?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Upload File by Mr.Parinya</title>
</head>
<body>
<form name="form1" method="post" action="chk.php" enctype="multipart/form-data">
```

```

<input type="file" name="fileUpload">
<input name="btnSubmit" type="submit" value="Submit">
</form>
</body>
</html>

```



ภาพที่ 6.2 แสดงหน้าเว็บเพจสำหรับเริ่มต้นอัปโหลดไฟล์

```

<?php /*กำหนดชื่อไฟล์ chk.php สำหรับเริ่มต้นอัปโหลด ไฟล์ไปเก็บไว้ยังเซิร์ฟเวอร์*/ ?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title> Upload File by Mr.Parinya</title>
</head>
<body>
<?php
    if(move_uploaded_file($_FILES["fileUpload"]["tmp_name"], "pictures/".$_FILES["fileUpload"]["name"])) { // กำหนดให้ pictures/ คือ โฟลเดอร์ปลายทางสำหรับเก็บไฟล์
        echo "อัปโหลดไฟล์เสร็จเรียบร้อยแล้ว";
    } else {
        echo "ไม่สามารถอัปโหลดไฟล์ได้ กรุณาระบุไฟล์ใหม่";
    }
?>
</body>
</html>

```



หมายเหตุ

ฟังก์ชันอัพโหลดไฟล์ move_uploaded_file () และ copy () มีรูปแบบการใช้งานเหมือนกัน ดังนี้ ซึ่งอัพโหลดชื่อฟังก์ชัน (ตำแหน่งไฟล์ต้นทาง, ตำแหน่งไฟล์ปลายทาง)

เมื่อ ตำแหน่งไฟล์ต้นทาง คือ \$_FILE ["file"] ["tmp_name"]

ตำแหน่งไฟล์ปลายทาง คือ ตำแหน่งใหม่ที่จะนำไฟล์ไปเก็บไว้ที่ใดของเครื่องเซิร์ฟเวอร์

ข้อระวังสำหรับการใช้งานทั้งตำแหน่งไฟล์ต้นทางและปลายทาง จะเป็นต้องมีที่ตั้งของไฟล์และซื้อไฟล์เสมอ สำหรับที่ตั้งของไฟล์จะเรียกว่า พาร์ไดเรกทอรี (Path Directory)

สรุป

ตัวแปรอาร์เรย์ใช้สำหรับเก็บค่าของข้อมูลที่อยู่ในกลุ่มเดียวกัน หรืออาจเรียกว่าชุดของตัวแปรที่มีการเรียงลำดับที่แน่นอนต่อเนื่องกันไปในหน่วยความจำ ตัวแปรแต่ละตัวจะเรียกว่าสมาชิกของอาร์เรย์ หรืออวิลิเม้นต์ของอาร์เรย์ สมาชิกแต่ละตัวของอาร์เรย์จะประกอบไปด้วย ค่าข้อมูล และอินเด็กซ์หรือคีย์ สามารถเข้าถึงค่าข้อมูลได้โดยใช้ชื่อตัวแปรเดียวกันตามด้วยอินเด็กซ์หรือคีย์

คำถามท้ายบท

1. จงอธิบายถึงลักษณะของตัวแปรอาร์เรย์ และความแตกต่างระหว่างตัวแปรทั่วไปกับตัวแปรชนิดอาร์เรย์
2. จงอธิบายวิธีและรูปแบบการสร้างอาร์เรย์โดยใช้ฟังก์ชัน array () พร้อมยกตัวอย่างประกอบ
3. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากสคริปต์

```

1 <?php
2     $province = array ("สุราษฎร์ธานี", "กระบี่", "ชุมพร", "นครศรีธรรมราช");
3     for ($loop = count ($province)-1; $loop >= 0; $loop--) {
4         printf ("%s <br>", $province [$loop]);
5     }
6 ?>

```

4. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากสคริปต์

```

1 <?php
2     $array = array ("number"=>"1", "text"=>"test", "1", 2, 3);
3     printf("%d<br>", sizeof ($array));
4 ?>

```

5. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการสคริปต์

```
1 <?php  
2     $array = array (1, "hello", 1, "world", "hello", "number"=> 1, "text"=>"hello");  
3     print_r (array_count_values ($array));  
4 ?>
```



บทที่ 7

ข้อความ ตัวเลข วันและเวลา

ในบทนี้จะกล่าวถึงการใช้ฟังก์ชันของ PHP เพื่อจัดการข้อมูลใน 3 รูปแบบ คือ ข้อความ ตัวเลข วันและเวลา โดยในบทที่ผ่านๆ มาหนึ่งได้ใช้งานข้อมูลแบบข้อความ และตัวเลข กันมาบ้าง แต่ก็เป็นเพียงการใช้ในแบบพื้นฐานทั่วไป ข้อมูลทั้งสองอย่างนี้ยังมีรายละเอียดปลีกย่อยที่จำเป็นต้องรู้อีกมาก many ส่วนข้อมูลประเภทวันและเวลาหนึ่ง ก็จัดว่าเป็นข้อมูลพิเศษอีกอย่างหนึ่ง ที่ไม่สามารถใช้ตัวดำเนินการได้ฯ ไปจัดการโดยตรงได้ ต้องอาศัยฟังก์ชันเฉพาะสำหรับจัดการ

7.1 ฟังก์ชันที่ใช้จัดการข้อความ

ข้อความ (String) คือ ชุดของตัวอักษร หรือการนำอักษรแต่ละตัวมาวางเรียงต่อกันเพื่อสื่อความหมาย เช่นคำว่า “String” เกิดจากอักษร S + t + r + i + n + g เรียงต่อกัน เป็นต้น ทั้งนี้อักษรที่วางเรียงต่อกันเป็นข้อความ จะมีลำดับของการจัดเก็บเหมือนกับอาร์เรย์ของข้อความ ดังนั้นการอ้างถึงอักษรย่อยๆ แต่ละตัวภายในข้อความ จึงใช้เลขลำดับเป็นตัวกำหนดเหมือนกับการอ้างอิงสมาชิกของอาร์เรย์ เช่น

ตัวอย่างที่ 7.1 แสดงตัวอย่างข้อความ

```
$str = "Thailand";  
echo $str[0] . $str[1] . $str[2] . $str[3]; //Thai
```

สำหรับฟังก์ชันที่ใช้จัดการข้อความมีจำนวนมาก แต่ที่ใช้งานกันบ่อยๆ มีอยู่ไม่นานนัก ดังนี้ในที่นี้จะกล่าวถึงเฉพาะฟังก์ชันที่น่าสนใจ และเพื่อให้ศึกษาได้ง่ายขึ้น จะแบ่งออกเป็นกลุ่มหัวข้อดังต่อไปนี้

7.1.1 ฟังก์ชันที่ใช้จัดการรหัสแอสกี

ฟังก์ชันที่ใช้จัดการรหัสแอสกี (ASCII) ที่น่าสนใจดังนี้

ตารางที่ 7.1 ฟังก์ชันที่ใช้จัดการรหัสแอสกี

ชื่อฟังก์ชัน	หน้าที่
ord (อักขระ)	หาค่ารหัสแอสกีของอักขระที่ระบุ
chr (ค่าแอสกี)	แปลงค่ารหัสแอสกีที่ระบุไปเป็นตัวอักขระ

ตัวอย่างที่ 7.2 การใช้ฟังก์ชัน ord () และ chr ()

```

<body>
<table border = "1" width = "100%" cellspacing = "0">
<caption>ตารางค่า ASCII ของ A – Z</caption>
<?php
    $a = ord ('A') ;
    $z = ord ('Z') ;
    for ($i = $a; $i <= $z; $i += 5) {
        echo "<tr align = center>";
        for ($j = $i ; $j < ($i+5) ; $j++) {
            $ch = chr ($j) ;
            echo "<td bgcolor = #ddd> $ch </td><td> $j </td>";
        }
        echo "</tr>" ;
    }
    ?>
</table>
</body>

```

// ผลลัพธ์ ดังนี้									
ตารางค่า ASCII ของ A – Z									
A	65	B	66	C	67	D	68	E	69
F	70	G	71	H	72	I	73	J	74
K	75	L	76	M	77	N	78	O	79
P	80	Q	81	R	82	S	83	T	84
U	85	V	86	W	87	X	88	Y	89
Z	90	[91	\	92]	93	^	94

7.1.2 ฟังก์ชันที่ใช้สำหรับหาขนาดของข้อความ

ฟังก์ชันที่ใช้สำหรับหาขนาดข้อความที่นำเสนอในมีรายละเอียด ดังนี้

- 1) ฟังก์ชัน strlen () เป็นฟังก์ชันที่ใช้สำหรับหาความยาวของข้อความ หรือนับจำนวนข้อความ โดยที่ซึ่งว่าง 1 ช่อง ก็จะนับเป็นอักษร 1 ตัวด้วย และในกรณีภาษาไทย และวรรณยุกต์ต่างๆ จะนับเป็นอักษร 1 ตัวเช่นกัน

รูปแบบ

```
int strlen ( string $string )
```

เมื่อ \$string หมายถึง ข้อความที่ต้องการนับจำนวน

ตัวอย่างที่ 7.3 การใช้ฟังก์ชัน strlen () สำหรับหาความยาวของข้อความ

```

1  <?php
2      $str = "abcdef";
3      echo strlen ($str); // ผลลัพธ์ คือ 6
4      $str = "ab cd";

```

```

5     echo strlen ($str); // ผลลัพธ์ คือ 7
6     ?>

```

จากตัวอย่างที่ 7.3 การใช้ฟังก์ชัน `strlen()` สำหรับหาความยาวของข้อความ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร `$str` มีค่าเท่ากับ "abcdef"

บรรทัดที่ 3 แสดงผลการเรียกใช้ฟังก์ชัน `strlen()` เพื่อนับตัวอักษรในตัวแปร `$str`

ผลลัพธ์ที่แสดง คือ 6

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร `$str` มีค่าเท่ากับ "abc def" (เพิ่มเว้นวรรค)

บรรทัดที่ 5 แสดงผลการเรียกใช้ฟังก์ชัน `strlen()` เพื่อนับตัวอักษรในตัวแปร `$str`

ผลลัพธ์ที่แสดง คือ 7

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

2) ฟังก์ชัน `str_word_count()` เป็นฟังก์ชันที่ใช้สำหรับนับจำนวนคำ โดยใช้อักษรที่ไม่ใช่ตัวอักษร a-z เป็นตัวคัตแยก (รวมถึงตัวเลขด้วย) ยกเว้นเครื่องหมาย ' และ -

รูปแบบ

```
mixed str_word_count ( string $string [, int $format = 0 [, string $charlist ]] )
```

เมื่อ `$string` หมายถึง ข้อความที่ต้องการนับจำนวน

`$format` หมายถึง กำหนดรูปแบบการนับ มีรายละเอียดดังนี้

0 คือ นับจำนวนคำ (โดยปริยายจะเป็น 0)

1 คือ ส่งค่ากลับเป็นอาร์เรย์ของคำ (ค่าอินเด็กซ์เรียงตามลำดับ)

2 คือ ส่งค่ากลับเป็นอาร์เรย์ของคำ (ค่าอินเด็กซ์มาจากการนับจำนวนตัวอักษรต่อกันไปเรื่อยๆ)

`$charlist` หมายถึง กำหนดอักษรพิเศษอื่นๆ ที่ต้องการนับเป็นคำ

ตัวอย่างที่ 7.4 การใช้ฟังก์ชัน `str_word_count()` สำหรับนับจำนวนคำ

```

1 <?php
2     $str = "Hello fri3nd, today! is good";
3     echo (str_word_count ($str));
4     print_r (str_word_count ($str, 1));
5     print_r (str_word_count ($str, 2));
6     ?>

```

จากตัวอย่างที่ 7.4 การใช้ฟังก์ชัน `str_word_count()` สำหรับนับจำนวนคำ อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP



บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$str มีค่าเท่ากับ "Hello fri3nd, today! is good"

บรรทัดที่ 3 แสดงผลการเรียกใช้ฟังก์ชัน str_word_count () เพื่อนับจำนวนคำ ในตัวแปร \$str ผลลัพธ์ที่แสดง คือ 6

บรรทัดที่ 4 แสดงผลการเรียกใช้ฟังก์ชัน str_word_count () โดยระบุ พารามิเตอร์ 1 เพื่อส่งค่ากลับเป็นอาร์เรย์ของคำ ผลลัพธ์ที่แสดง คือ Array ([0] => Hello [1] => fri [2] => nd [3] => today [4] => is [5] => good)

บรรทัดที่ 5 แสดงผลการเรียกใช้ฟังก์ชัน str_word_count () โดยระบุ พารามิเตอร์ 2 เพื่อส่งค่ากลับเป็นอาร์เรย์ของคำ ผลลัพธ์ที่แสดง คือ Array ([0] => Hello [6] => fri [10] => nd [14] => today [21] => is [24] => good)

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

7.1.3 ฟังก์ชันที่ใช้สำหรับเปลี่ยนรูปแบบของตัวพิมพ์

ฟังก์ชันที่ใช้สำหรับเปลี่ยนรูปแบบของตัวพิมพ์มีรายละเอียด ดังนี้

1) ฟังก์ชัน strtolower () เป็นฟังก์ชันที่ใช้สำหรับเปลี่ยนข้อความที่ระบุให้เป็นตัวพิมพ์ เล็กทุกตัว

รูปแบบ

```
string strtolower ( string $str )
```

เมื่อ \$str หมายถึง ข้อความที่ต้องการเปลี่ยนรูปแบบ

ตัวอย่างที่ 7.5 ฟังก์ชัน strtolower () สำหรับเปลี่ยนข้อความที่ระบุให้เป็นตัวพิมพ์เล็กทุกตัว

```
1 <?php  
2     $str = "Mary Had A Little Lamb and She LOVED It So";  
3     $str = strtolower($str);  
4     echo $str;  
5 ?>
```

ผลลัพธ์

```
mary had a little lamb and she loved it so
```

2) ฟังก์ชัน strtoupper () เป็นฟังก์ชันที่ใช้สำหรับเปลี่ยนข้อความที่ระบุให้เป็นตัวพิมพ์ ใหญ่ทุกตัว

รูปแบบ

`string strtoupper (string $str)`

เมื่อ \$str หมายถึง ข้อความที่ต้องการเปลี่ยนรูปแบบ

ตัวอย่างที่ 7.6 ฟังก์ชัน strtoupper () ใช้สำหรับเปลี่ยนข้อความที่ระบุให้เป็นตัวพิมพ์ใหญ่ทุกตัว

```
1 <?php
2     $str = "Mary Had A Little Lamb and She LOVED It So";
3     $str = strtoupper ($str);
4     echo $str;
5 ?>
```

ผลลัพธ์

MARY HAD A LITTLE LAMB AND SHE LOVED IT SO

3) ฟังก์ชัน ucfirst () เป็นฟังก์ชันที่ใช้สำหรับเปลี่ยนข้อความที่ระบุ เฉพาะตัวอักษรตัวแรกเท่านั้นเป็นตัวพิมพ์ใหญ่ ส่วนข้อความที่เหลือเหมือนเดิม

รูปแบบ

`string ucfirst (string $str)`

เมื่อ \$str หมายถึง ข้อความที่ต้องการเปลี่ยนรูปแบบ

ตัวอย่างที่ 7.7 การใช้ฟังก์ชัน ucfirst ()

```
1 <?php
2     $foo = 'welcome to thailand';
3     echo ucfirst ($foo);           // ผลลัพธ์ คือ Welcome to thailand
4     $bar = 'WELCOME TO THAILAND';
5     echo ucfirst ($bar);           // ผลลัพธ์ เมื่อเดิม คือ WELCOME TO THAILAND
6     echo ucfirst (strtolower ($bar)); // ผลลัพธ์ คือ Welcome to thailand
7 ?>
```

4) ฟังก์ชัน ucwords () เป็นฟังก์ชันที่ใช้สำหรับเปลี่ยนข้อความที่ระบุ เฉพาะตัวอักษรตัวแรกของทุกคำเป็นตัวพิมพ์ใหญ่ ส่วนข้อความที่เหลือเหมือนเดิม

รูปแบบ

`string ucwords (string $str)`

เมื่อ \$str หมายถึง ข้อความที่ต้องการเปลี่ยนรูปแบบ



ตัวอย่างที่ 7.8 การใช้ฟังก์ชัน ucwords ()

```
1 <?php
2     $foo = 'welcome to thailand';
3     echo ucwords ($foo);    // ผลลัพธ์ คือ Welcome To Thailand
4     $bar = 'WELCOME TO THAILAND';
5     echo ucwords ($bar);    // ผลลัพธ์ เหมือนเดิม คือ WELCOME TO THAILAND
6     echo ucwords (strtolower ($bar));    // ผลลัพธ์ คือ Welcome To Thailand
7 ?>
```

7.1.4 ฟังก์ชันที่ใช้สำหรับแยกและรวมข้อความ

ฟังก์ชันที่ใช้สำหรับแยกและรวมข้อความที่น่าสนใจมีรายละเอียด ดังนี้

1) ฟังก์ชัน explode () ใช้สำหรับคัดแยกข้อความออกเป็นข้อความย่อยๆ ด้วยสัญลักษณ์ที่กำหนดโดยผลลัพธ์ที่ได้จะอยู่ในรูปแบบของอาร์เรย์ของข้อความย่อยที่ถูกแยกออกมา

รูปแบบ

```
array explode ( string $delimiter , string $string )
```

เมื่อ \$delimiter หมายถึง สัญลักษณ์ที่ใช้แยก

\$string หมายถึง ข้อความที่ต้องการแยก

ตัวอย่างที่ 7.9 การใช้ฟังก์ชัน explode () คัดแยกข้อความออกเป็นข้อความย่อยๆ

```
1 <?php
2     // Example 1
3     $pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
4     $pieces = explode (" ", $pizza);
5     echo $pieces [0];    // ผลลัพธ์ คือ piece1
6     echo $pieces [1];    // ผลลัพธ์ คือ piece2
7     // Example 2
8     $data = "foo:*:1023:1000::/home/foo:/bin/sh";
9     list ($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode (":", $data);
10    echo $user;        // ผลลัพธ์ คือ foo
11    echo $pass;        // ผลลัพธ์ คือ *
12 ?>
```

2) ฟังก์ชัน `implode()` และ `join()` ทั้ง 2 ฟังก์ชันจะทำงานตรงข้ามกับฟังก์ชัน `explode()` คือ เป็นฟังก์ชันรวมอาร์เรย์เข้าด้วยกันเป็นข้อความ โดยค้นแต่ละข้อความด้วยสัญลักษณ์ที่ระบุ รูปแบบ ทั้ง 2 ฟังก์ชันเหมือนกัน ดังนี้จะแนะนำเฉพาะฟังก์ชัน `implode()`

รูปแบบ

```
string implode ( string $glue , array $pieces )
```

เมื่อ `$glue` หมายถึง สัญลักษณ์ระบุเพิ่มเข้าไประหว่างข้อความ (หากไม่ต้องการคั่นด้วย สัญลักษณ์ก็สามารถระบุเป็นค่าว่างหรือเว้นวรรคได้ แต่หากจะแยกใหม่ ในภายหลังจะทำได้ยังยากหากไม่มีสัญลักษณ์คั่นในแต่ละข้อความ)

`$pieces` หมายถึง อาร์เรย์ที่ต้องการนำมารวมกันเป็นข้อความ

ตัวอย่างที่ 7.10 การใช้ฟังก์ชัน `implode()` และ `join()`

```
1 <?php
2     $array = array ("lastname", "email", "phone");
3     $comma_separated = implode (", ", $array);
4     echo $comma_separated;
5 ?>
```

ผลลัพธ์

```
lastname, email, phone
```

7.1.5 ฟังก์ชันที่ใช้จัดการข้อความย่อ

ฟังก์ชันที่ใช้จัดการข้อความย่อที่น่าสนใจรายละเอียด ดังนี้

1) ฟังก์ชัน `strstr()` เป็นฟังก์ชันที่ใช้สำหรับค้นหาและเลือกเอาเฉพาะข้อความที่ค้นพบ

เป็นต้นไป

รูปแบบ

```
string strstr ( string $haystack , mixed $needle [, bool $before_needle = false ] )
```

เมื่อ `$haystack` หมายถึง ข้อความต้นฉบับ

`$needle` หมายถึง ระบุข้อความที่ต้องการ

`$before_needle` หมายถึง โดยปริยายจะมีค่าเป็น `false` คือ เอาเฉพาะข้อความที่ ระบุตั้งแต่คำที่พบรหัส `$needle` เป็นต้นไป แต่ถ้าระบุเป็น `true` จะเอาเฉพาะข้อความทั้งหมดในด้านซ้ายก่อน ข้อความที่ระบุ



ตัวอย่างที่ 7.11 การใช้ฟังก์ชัน strstr ()

```

1 <?php
2     $email = 'name@example.com';
3     $domain = strstr ($email, '@');
4     echo $domain;           // ผลลัพธ์ คือ @example.com
5     $user = strstr ($email, '@', true);
6     echo $user;            // ผลลัพธ์ คือ name
7 ?>

```

2) ฟังก์ชัน substr () เป็นฟังก์ชันที่ใช้สำหรับคัดเอาข้อความย่ออย โดยกำหนดเอาตำแหน่งเริ่มต้น และความยาวที่ต้องการ

รูปแบบ

```
string substr ( string $string , int $start [, int $length ] )
```

เมื่อ \$string หมายถึง ข้อความต้นฉบับ

\$start หมายถึง ตำแหน่งเริ่มต้น เช่น ข้อความ 'abcdef' ตำแหน่งของตัวอักษรจะเริ่มต้นที่ 0 คือ 'a' ตำแหน่งของตัวอักษรตำแหน่งที่ 2 คือ 'c' และตำแหน่งอื่นๆ ตามลำดับ

\$length หมายถึง ความยาว

ตัวอย่างที่ 7.12 การใช้ฟังก์ชัน substr ()

```

1 <?php
2     $rest = substr ("abcdef", 0, -1); // ผลลัพธ์ คือ "abcde"
3     $rest = substr ("abcdef", 2, -1); // ผลลัพธ์ คือ "cde"
4     $rest = substr ("abcdef", 4, -4); // ผลลัพธ์ คือ false ไม่สามารถคัดเอาข้อความได้
5     $rest = substr ("abcdef", -3, -1); // ผลลัพธ์ คือ "de"
6 ?>

```

3) ฟังก์ชัน substr_count () เป็นฟังก์ชันที่ใช้สำหรับนับจำนวนคำที่มีอยู่ในข้อความ โดยรูปแบบของการค้นหาคำที่ตรงกัน (case-sensitive)

รูปแบบ

```
int substr_count (string $haystack , string $needle [, int $offset = 0 [, int $length ]])
```

เมื่อ \$haystack หมายถึง ข้อความที่ต้องการค้นหาเพื่อนับจำนวน

\$needle หมายถึง คำที่ใช้สำหรับการค้นหา

`$offset` หมายถึง ตำแหน่งของตัวอักษรที่จะเริ่มนับ

`$length` หมายถึง ความยาว

ตัวอย่างที่ 7.13 การใช้ฟังก์ชัน substr_count()

```

1 <?php
2     $text = 'This is a test';
3     echo strlen ($text); // ผลลัพธ์ ของจำนวนตัวอักษร คือ 14
4     echo substr_count ($text, 'is'); // ผลลัพธ์ ของ 'is' คือ 2
5     echo substr_count ($text, 'is', 3); // การนับจะเริ่มต้นที่ 'is a test' ผลลัพธ์ คือ 1
6     // ไม่พบคำที่ต้องการนับ เพราะตำแหน่งนับ คือ 's i' ผลลัพธ์ คือ 0
7     echo substr_count ($text, 'is', 3, 3);
8     // ไม่สามารถนับได้และแสดงข้อความเตือน เพราะ 5+10 > 14 (ย่างกว่าข้อความ)
9     echo substr_count($text, 'is', 5, 10);
10    ?>

```

7.1.6 ฟังก์ชันค้นหาข้อความ

ฟังก์ชันค้นหาข้อความ มีดังนี้

1) ฟังก์ชัน strpos () เป็นฟังก์ชันที่ใช้สำหรับค้นหาตำแหน่งของข้อความย่อย ค่าที่คืนกลับมาจะเป็นตำแหน่งที่ฟังก์ชันค้นพบข้อความย่อยเป็นครั้งแรก แต่หากไม่พบจะคืนค่า null กลับมา (ตัวพิมพ์เล็กและพิมพ์ใหญ่ถือว่าเป็นคนละตัวกัน: case-sensitive)

รูปแบบ

```
int strpos ( string $haystack , mixed $needle [, int $offset = 0 ] )
```

เมื่อ `$haystack` หมายถึง ข้อความที่ต้องการค้นหา

`$needle` หมายถึง คำที่ใช้สำหรับการค้นหา

`$offset` หมายถึง ตำแหน่งของตัวอักษรที่จะเริ่มค้นหา

ตัวอย่างที่ 7.14 การใช้ฟังก์ชัน strpos()

```

1 <?php
2     $findme = "ya";
3     $mystring = "parinya";
4     $pos = strpos ($mystring, $findme);
5     echo "Finding is $findme in $mystring ";
6     echo "at $pos";
7     ?>

```



2) พังก์ชัน `stripos()` เป็นพังก์ชันที่ใช้สำหรับค้นหาตำแหน่งของข้อความย่อๆ เมื่อพังก์ชัน `strpos()` ค่าที่คืนกลับมาจะเป็นตำแหน่งที่พังก์ชันค้นพบย่อๆ เป็นครั้งแรก แต่หากไม่พบจะคืนค่า `null` กลับมา (ตัวพิมพ์เล็กและพิมพ์ใหญ่ถือว่าเป็นตัวเดียวกัน)

รูปแบบ

```
int stripos ( string $haystack , string $needle [, int $offset = 0 ] )
```

เมื่อ `$haystack` หมายถึง ข้อความที่ต้องการค้นหา

`$needle` หมายถึง คำที่ใช้สำหรับการค้นหา

`$offset` หมายถึง ตำแหน่งของตัวอักษรที่จะเริ่มค้นหา

ตัวอย่างที่ 7.15 การใช้ฟังก์ชัน `stripos()`

```
1 <?php
2     $findme = "ya";
3     $mystring = "parinya";
4     $pos = strpos ($mystring, $findme);
5     echo "Finding is $findme in $mystring ";
6     echo "at $pos";
7 ?>
```

ผลลัพธ์ ของตัวอย่างที่ 7.14 และ 7.15 ดังนี้

```
Finding is ya in parinya at 5
```

7.1.7 พังก์ชันแทนที่ข้อความ

ฟังก์ชันแทนที่ข้อความ มีรายละเอียด ดังนี้

1) พังก์ชัน `str_replace()` เป็นพังก์ชันที่ใช้สำหรับค้นหา และแทนที่ข้อความย่อๆด้วยข้อความใหม่ที่ต้องการ หากข้อความที่ต้องการค้นหามีมากกว่า 1 ครั้ง ก็จะถูกแทนที่ทั้งหมด (ตัวพิมพ์เล็กและพิมพ์ใหญ่ถือว่าเป็นตัวเดียวกัน)

รูปแบบ

```
mixed str_replace ( mixed $search , mixed $replace , mixed $subject [, int &$count ] )
```

เมื่อ `$search` หมายถึง ข้อความที่ต้องการค้นหา

`$replace` หมายถึง ข้อความที่ต้องการแทนที่

`$subject` หมายถึง ข้อความ หรืออาร์เรย์ ที่ใช้สำหรับการค้นหาและแทนที่ (ไม่ต้องระบุก็ได้)

`$count` หมายถึง จำนวนครั้งที่ต้องการให้แทนที่

ตัวอย่างที่ 7.16 การใช้ฟังก์ชัน str_replace()

```

1 <?php
2
3     $vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
4     echo str_replace ($vowels, "", "Hello World of PHP");
5
6     $phrase = "You should eat fruits, vegetables, and fiber every day.";
7     $healthy = array ("fruits", "vegetables", "fiber");
8     $yummy = array ("pizza", "beer", "ice cream");
9     echo str_replace ($healthy, $yummy, $phrase);
10
11 ?>

```

ผลลัพธ์

You should eat pizza, beer, and ice cream every day

จากตัวอย่างที่ 7.16 การใช้ฟังก์ชัน str_replace() สำหรับค้นหาและแทนที่ข้อความย่อyledeim ด้วยข้อความย่อใหม่ อธิบายเพิ่มเติม ดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$vowels มีค่าเท่ากับ array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U")

บรรทัดที่ 3 แสดงผลการเรียกใช้ฟังก์ชัน str_replace() ค้นหาและแทนที่ข้อความย่อyledeim ด้วยข้อความย่อใหม่ จากตัวอย่างกำหนดเป็นค่าว่าง ดังนั้นผลลัพธ์ที่จะแสดงผล คือ Hll Wrld f PHP

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$phrase มีค่าเท่ากับข้อความ "You should eat fruits, vegetables, and fiber every day."

บรรทัดที่ 5 กำหนดค่าให้กับตัวแปร \$healthy มีค่าเท่ากับ array ("fruits", "vegetables", "fiber")

บรรทัดที่ 6 กำหนดค่าให้กับตัวแปร \$yummy มีค่าเท่ากับ array ("pizza", "beer", "ice cream")

บรรทัดที่ 7 แสดงผลการเรียกใช้ฟังก์ชัน str_replace() ค้นหาและแทนที่ข้อความย่อyledeim ด้วยข้อความย่อใหม่ ผลลัพธ์ที่จะแสดงผล คือ You should eat pizza, beer, and ice cream every day.

บรรทัดที่ 8 สิ้นสุดสคริปต์ PHP

2) ฟังก์ชัน str_ireplace() เป็นฟังก์ชันที่ใช้สำหรับค้นหาและแทนที่ข้อความย่อyledeim ด้วยข้อความใหม่ที่ต้องการ หากข้อความย่อที่ต้องการค้นหามีมากกว่า 1 ครั้ง เมื่อกับฟังก์ชัน str_replace() เพียงแต่จะไม่สนใจความแตกต่างของตัวพิมพ์ (ignore case)



รูปแบบ

```
mixed str_ireplace ( mixed $search , mixed $replace , mixed $subject [, int &$count ] )
```

เมื่อ \$search หมายถึง ข้อความที่ต้องการค้นหา
 \$replace หมายถึง ข้อความที่ต้องการแทนที่
 \$subject หมายถึง ข้อความ หรืออาร์เรย์ ที่ใช้สำหรับการค้นหาและแทนที่ (ไม่ต้องระบุก็ได้)
 \$count หมายถึง จำนวนครั้งที่ต้องการให้แทนที่

3) พังก์ชัน strstr () เป็นพังก์ชันที่ใช้สำหรับการแปลงตัวอักษรหรือแทนที่ข้อความ และสามารถกำหนดข้อความย่ออย่างในการแทนที่ได้มากกว่า 1 คำ โดยอาร์เรย์ของข้อความที่จะใช้แทนที่จะต้องกำหนดในรูปแบบของคีย์และค่าอาร์เรย์ โดยใช้คีย์อาร์เรย์เป็นสิ่งที่ต้องการค้นหา และค่าอาร์เรย์เป็นสิ่งที่จะนำไปแทนที่ รูปแบบดังนี้

รูปแบบ

```
string strstr ( string $str , array $replace_pairs )
```

เมื่อ \$str หมายถึง ข้อความที่ต้องการแปลง
 \$from หมายถึง ข้อความที่ต้องการค้นหา
 \$to หมายถึง ข้อความที่ต้องการแทนที่
 \$replace_pairs หมายถึง พารามิเตอร์ ใช้ระบุข้อความที่ต้องการค้นหาและแทนที่ในกรณีของอาร์เรย์ array ('from' => 'to', ...) ในในรูปแบบของคีย์และค่าอาร์เรย์

ตัวอย่างที่ 7.17 การใช้งานพังก์ชัน strstr ()

```
1 <?php
2     $trans = array ("hello" => "hi", "hi" => "hello");
3     echo strstr ("hi all, I said hello", $trans);
4 ?>
```

ผลลัพธ์

```
hello all, I said hi
```

7.1.8 พังก์ชันที่ใช้สำหรับตัดช่องว่างและเติมข้อความ

1) พังก์ชันที่ใช้สำหรับตัดช่องว่าง ประกอบไปด้วย 1) ltrim () 2) rtrim () และ 3) trim () โดยทั้ง 3 พังก์ชันมีรูปแบบการใช้งานเหมือนกัน มีรายละเอียดดังนี้

ตารางที่ 7.2 ฟังก์ชันในการตัดซองว่าง

ชื่อฟังก์ชัน	หน้าที่
ltrim ()	ตัดซองว่างที่อยู่ด้านซ้ายของข้อความออกทั้งหมด
rtrim ()	ตัดซองว่างที่อยู่ด้านขวาของข้อความออกทั้งหมด
trim ()	ตัดซองว่างทั้งด้านซ้าย และด้านขวาของข้อความออกทั้งหมด

ตัวอย่างที่ 7.18 การประยุกต์ใช้ฟังก์ชันที่ใช้สำหรับตัดซองว่าง

```

1 <?php
2     $str = "      PHP      ";
3     echo strlen ($str).<br>;           // ผลลัพธ์ของการนับ คือ 19
4     echo strlen (ltrim ($str)).<br>;    // ผลลัพธ์ของการนับ คือ 11
5     echo strlen (rtrim ($str)).<br>;    // ผลลัพธ์ของการนับ คือ 11
6     echo strlen (trim ($str));         // ผลลัพธ์ของการนับ คือ 3
7 ?>

```

จากตัวอย่างจะใช้วิธีการทดสอบโดยการนับจำนวนเนื้อจากซองว่าง (whitespace) จะไม่สามารถมองเห็นได้ แต่จะส่งผลเมื่อใช้สำหรับการค้นหาหรือเปรียบเทียบข้อความ เพราะซองว่างเทียบได้กับอักขระ 1 ตัว

2) ฟังก์ชัน str_pad () เป็นฟังก์ชันที่ใช้สำหรับขยายความยาวของข้อความหรือเติมอักขระหรือข้อความลงในข้อความเดิม

รูปแบบ

```

string str_pad ( string $input , int $pad_length [, string $pad_string = " " [, int
$pad_type = STR_PAD_RIGHT ]]] )

```

เมื่อ \$input หมายถึง ข้อความนำเข้า

\$pad_length หมายถึง ความยาวที่ต้องการขยายเพิ่มเติม (ถ้าค่าตัวเลขน้อยกว่าหรือเท่ากับจำนวนความกว้างของข้อความใน \$input จะไม่มีการขยาย)

\$pad_string หมายถึง ตัวอักษรหรือข้อความอื่นๆ ที่จะเติมลงไป (มีระบุได้)

\$pad_type หมายถึง ด้านที่ต้องการขยายเพิ่มเติม โดยปริยายจะเติมด้านขวา มีรายละเอียด ดังนี้ STR_PAD_RIGHT (ขยายเพิ่มด้านซ้าย), STR_PAD_LEFT (ขยายเพิ่มด้านขวา) หรือ STR_PAD_BOTH (ขยายทั้งซ้ายและขวาท่าๆ กัน)



ตัวอย่างที่ 7.19 การใช้ฟังก์ชัน str_pad()

```

1 <?php
2     $input = "Alien";
3     echo str_pad($input, 10);           // ผลลัพธ์ คือ "Alien      "
4     echo str_pad($input, 10, "-=", STR_PAD_LEFT); // ผลลัพธ์ คือ "-==Alien"
5     echo str_pad($input, 10, "_", STR_PAD_BOTH); // ผลลัพธ์ คือ "__Alien__"
6     echo str_pad($input, 6, "__");        // ผลลัพธ์ คือ "Alien_"
7 ?>

```

7.1.9 ฟังก์ชันที่ใช้จัดการอักษรพิเศษของ HTML

ฟังก์ชันที่ใช้จัดการอักษรพิเศษของ HTML มีรายละเอียด ดังนี้

1) ฟังก์ชัน nl2br() เป็นฟังก์ชันสำหรับเปลี่ยนอักษรพิเศษ เช่น \n, \t, \r และ \r เป็นคำสั่งที่ใช้สำหรับขึ้นบรรทัดใหม่ ให้เปลี่ยนเป็นแท็ก
 ของภาษา HTML มีความจำเป็นในกรณีที่ต้องการให้ผู้ใช้ป้อนข้อมูลผ่านทางจอภาร เช่น ป้อนข้อมูลผ่านกล่องข้อความ (Text Field หรือ Text Box) หลายๆ บรรทัด หรือโดยปกติแล้วเมื่อกดปุ่ม Enter เพื่อขึ้นบรรทัดใหม่ ตำแหน่งที่กดปุ่ม Enter ก็จะถูกบันทึกด้วยสัญลักษณ์ \n หรือ new line เพื่อเป็นสัญลักษณ์บ่งชี้ว่าเป็นตำแหน่งที่จะต้องขึ้นบรรทัดใหม่ แม้จะมองไม่เห็นสัญลักษณ์นี้ก็ตาม แต่ในเอกสาร HTML สัญลักษณ์ \n จะไม่ทำให้ขึ้นบรรทัดใหม่ เพราะในเอกสาร HTML จะขึ้นบรรทัดใหม่ด้วยสัญลักษณ์
 เท่านั้น ดังนั้นฟังก์ชัน nl2br() จึงใช้ในการเปลี่ยนจากสัญลักษณ์ \n ให้เป็น

รูปแบบ

```
string nl2br ( string $string [, bool $is_xhtml = true ] )
```

เมื่อ \$string หมายถึง ข้อความ

\$is_xhtml หมายถึง จะใช้รูปแบบ XHTML หรือไม่ (ค่าโดยปริยายจะกำหนดเป็น true)

ตัวอย่างที่ 7.20 การใช้ฟังก์ชัน nl2br()

```

1 <?php
2     echo nl2br("สวัสดีครับ\nwww.sru.ac.th");
3 ?>

```

ผลลัพธ์

สวัสดีครับ

www.sru.ac.th

2) พังก์ชัน wordwrap () เป็นพังก์ชันสำหรับแบ่งข้อความตามจำนวนตัวอักษรที่กำหนด โดยจะไม่ตัดคำหรือฉีกข้อความ ใช้แยกคำจากการเว้นวรรค ภายหลังจากการแบ่งข้อความแล้วจะดำเนินการอย่างไรก็แล้วแต่กำหนด

รูปแบบ

```
string wordwrap (string $str [, int $width = 75 [, string $break = "\n" [, bool $cut = false ]]])
```

เมื่อ \$str หมายถึง ตัวแปรที่ใช้เก็บค่าข้อความ

\$width หมายถึง จำนวนตัวอักษรที่ต้องการ (โดยปริยาย คือ 75)

\$break หมายถึง คำสั่งที่ต้องการให้ดำเนินการ (โดยปริยาย คือ \n เพื่อขึ้นบรรทัดใหม่)

\$cut หมายถึง ต้องการตัดคำหรือไม่ หากกำหนดเป็น true จะตัดคำที่มีความยาว
เกินกว่าที่ \$width กำหนด

ตัวอย่างที่ 7.21 การใช้พังก์ชัน wordwrap ()

```
1 <?php
2     $text = "The quick brown fox jumped over the lazy dog.";
3     $newtext = wordwrap ($text, 20, "<br>\n");
4     echo $newtext;
5 ?>
```

ผลลัพธ์

```
The quick brown fox
jumped over the lazy
dog.
```

7.2 พังก์ชันที่ใช้จัดการจำนวนและตัวเลข

พังก์ชันที่ใช้จัดการจำนวนและตัวเลข ในภาษา PHP มีพังก์ชันที่ใช้จัดการข้อมูลชนิดนี้อยู่หลายพังก์ชัน แต่สำหรับในบทที่ จะแนะนำเฉพาะพังก์ชันที่นิยมใช้งาน มีรายละเอียด ดังนี้

7.2.1 พังก์ชันประมาณค่าจำนวนและตัวเลข

พังก์ชันประมาณค่าจำนวนและตัวเลข มีดังนี้

ตารางที่ 7.3 พังก์ชันประมาณค่า

ceil (จำนวน)	ใช้ในการปัดเศษขึ้นไปเป็นเลขจำนวนเต็มตัวถัดไปถ้าเศษมีค่ามากกว่า 0 เช่น
	echo ceil (10.01); //ผลลัพธ์ คือ 11
	echo ceil (10.50); //ผลลัพธ์ คือ 11



ตารางที่ 7.3 (ต่อ)

floor (จำนวน)	ใช้ในการตัดเศษทิ้ง ไม่ว่าเศษนั้นจะมีค่าเท่าใดก็ตาม เช่น
	echo floor (10.01); //ผลลัพธ์ คือ 10
	echo floor (10.50); //ผลลัพธ์ คือ 10
round (จำนวน, [, ทศนิยม])	ใช้ในการประมาณค่าเป็นจำนวนเต็มที่ใกล้เคียง หากเศษน้อยกว่า 0.5 จะตัดเศษทิ้ง แต่หากเศษมีค่าตั้งแต่ 0.5 จะปัดขึ้นไปเป็นจำนวนเต็มถัดไป เช่น
	echo round (10.49); //ผลลัพธ์ คือ 10
	echo round (10.50); //ผลลัพธ์ คือ 11
	ส่วนทศนิยมเป็นการกำหนดว่าจะให้มีทศนิยมกี่ตำแหน่ง
intval (จำนวน)	เลือกเอาเฉพาะจำนวนเต็มของจำนวนที่ระบุ หากมีทศนิยมจะตัดทิ้ง เช่น
	echo intval (10.01); //ผลลัพธ์ คือ 10
	echo intval (10.99); //ผลลัพธ์ คือ 10
	echo intval (-1.23); //ผลลัพธ์ คือ -1
floatval (จำนวน)	เลือกเอาจำนวนที่ระบุในแบบ float ฟังก์ชันนี้จะมีประโยชน์ในกรณีที่จำนวนนั้นวางอยู่ข้างหน้าข้อความ เช่น
	echo floatval ("12.34 MB"); //ผลลัพธ์ คือ 12.34

7.2.2 ฟังก์ชันเปรียบเทียบจำนวน

ฟังก์ชันเปรียบเทียบจำนวน มีรายละเอียด ดังนี้

ตารางที่ 7.4 ฟังก์ชันเปรียบเทียบจำนวน

min (n1, n2, ...n)	หรือ หาค่าที่น้อยที่สุดของช่วงที่กำหนด
min (อาร์เรย์)	\$m1 = min (6, 7, 3, 8, 9); //ผลลัพธ์ คือ \$m1 = 3
	\$m2 = min (array (10, 10.5, 3.4, 5.2, 20)); //ผลลัพธ์ คือ \$m2 = 3.4
max (n1, n2, ...n)	หรือ หาค่าที่มากที่สุดของช่วงที่กำหนด
max (อาร์เรย์)	\$m1 = max (6, 7, 3, 8, 9); //ผลลัพธ์ คือ \$m1 = 9
	\$m2 = max (array (10, 10.5, 3.4, 5.2, 20)); //ผลลัพธ์ คือ \$m2 = 20

7.2.3 ฟังก์ชันการตรวจสอบ และจัดรูปแบบตัวเลข

ฟังก์ชันที่น่าสนใจเกี่ยวกับการตรวจสอบ และจัดรูปแบบตัวเลข มีดังนี้

ตารางที่ 7.5 ฟังก์ชันการตรวจสอบ และจัดรูปแบบตัวเลข

number_format (จำนวน)	ใช้ในการจัดรูปแบบตัวเลข เช่น จาก 1234 เป็น 1,234 เป็นต้น เช่น
	echo number_format (1234567); // 1,234,567

ตารางที่ 7.5 (ต่อ)

is_numeric (ข้อมูล)	ตรวจสอบว่าข้อมูลที่ระบุนั้นเป็นตัวเลขหรือไม่ โดยจะคืนค่า true ถ้าเป็นข้อมูลตัวเลขที่ไม่มีอักษรอื่นๆ ปะปนอยู่ด้วย ทั้งนี้ตัวเลขที่เขียนในแบบข้อความ เช่น “123” ก็ถือว่าเป็นตัวเลขด้วย เพราะสามารถนำไปใช้ คำนวนได้ เช่น \$a = is_numeric(123); //ผลลัพธ์ คือ \$a = true \$b = is_numeric("1.23"); //ผลลัพธ์ คือ \$b = true \$c = is_numeric("123abc"); //ผลลัพธ์ คือ \$c = false
is_int (ข้อมูล)	ตรวจสอบว่าเป็นจำนวนเต็มหรือไม่ หากใช่จะคืนค่า true
is_float (ข้อมูล)	ตรวจสอบว่าเป็นจำนวนจริงหรือไม่ หากใช่จะคืนค่า true

7.2.4 พังก์ชันอื่นๆ ที่เกี่ยวข้องกับจำนวนและตัวเลข

พังก์ชันอื่นๆ ที่เกี่ยวข้องกับจำนวนและตัวเลข มีรายละเอียด ดังนี้

ตารางที่ 7.6 พังก์ชันอื่นๆ ที่เกี่ยวข้องกับตัวเลข

abs (จำนวน)	ใช้สำหรับหาค่า absolute เช่น \$x = abs(10); //ผลลัพธ์ คือ \$x = 10 \$y = abs (-10); //ผลลัพธ์ คือ \$y = 10
rand () หรือ rand (ค่าน้อยที่สุด, ค่ามากที่สุด)	ใช้ในการสร้างเลขสุ่ม โดยค่าที่ได้จะไม่แน่นอนในแต่ละครั้ง โดยปกติแล้ว เลขสุ่มที่ได้จะเป็นเลขจำนวนเต็มที่มีค่าระหว่าง 0-32,768 แต่อย่างไรก็ตาม PHP อนุญาตให้สามารถกำหนดช่วงตัวเลขผลลัพธ์ที่ต้องการได้ว่าให้อยู่ระหว่างค่าใดถึงค่าใด เช่น \$r1 = rand (); \$r2 = rand (10, 20); //เลขสุ่มที่ได้จะมีค่าระหว่าง 10-20
pow (เลขฐาน, เลขชี้กำลัง)	ใช้ในการค่าเลขยกกำลัง เช่น echo pow (10, 2); //100
sqrt (ตัวเลข)	หาค่ารากที่สองของจำนวนที่ระบุ แต่ต้องไม่ใช่จำนวนที่ติดลบ เช่น \$a = sqrt (100); // \$a = 10

7.3 พังก์ชันที่ใช้จัดการวันและเวลา

ข้อมูลเกี่ยวกับวันและเวลา นับว่าเป็นข้อมูลที่สำคัญอีกอย่างหนึ่งของการเขียนโปรแกรมแบบสคริปต์ การใช้งานส่วนใหญ่จะทำผ่านพังก์ชันต่างๆ ที่เกี่ยวข้อง รูปแบบของวันที่และเวลาสามารถแบ่งเป็น 2 ประเภท คือ 1) Local Mean Time (LMT) คือ เวลาเฉลี่ยของแต่ละท้องถิ่น และ 2) Greenwich Mean Time (GMT) คือ เวลามาตรฐานสากล เป็นเวลาที่เทียบกับเมืองกรีนิช ประเทศอังกฤษ โดยหน่วยเวลาอย่างเป็นทางการ เรียกว่า UTC พังก์ชันที่ใช้จัดการวันและเวลา มีดังนี้



7.3.1 พังก์ชัน date_default_timezone_set()

พังก์ชัน date_default_timezone_set() ใช้สำหรับกำหนดรูปแบบ Timezone ที่ต้องการใช้ และจะแสดงวันและเวลาตาม Timezone ที่กำหนด รูปแบบการใช้พังก์ชัน มีดังนี้

รูปแบบ

```
bool date_default_timezone_set( string $timezone_identifier )
```

เมื่อ \$timezone_identifier หมายถึง เขตเวลา (Timezone) สำหรับเวลาในประเทศไทย ต้องระบุค่าเป็น "Asia/Bangkok" เป็นเวลา LMT สำหรับเวลา GMT สามารถระบุค่าเป็น GMT หรือ UTC

7.3.2 พังก์ชัน time() เป็นพังก์ชันที่ใช้แสดงวันที่และเวลาปัจจุบันในรูปแบบของ Unix Timestamp จะแสดงเวลาเป็นหน่วยวินาที โดยเริ่มนับค่าตั้งแต่วันที่ 1 มกราคม ค.ศ.1970 เวลา 00:00:00 น. ตามเวลา GMT จนถึงวันที่และเวลาปัจจุบัน ยกตัวอย่างเช่น วันที่ 26 พฤษภาคม 2012 เวลา 13:47:00 มีค่า Unix Timestamp คือ "1353912424"

รูปแบบ

```
int time( void )
```

ตัวอย่างที่ 7.22 การใช้พังก์ชัน time()

```
1 <?php
2     $nextWeek = time() + (7 * 24 * 60 * 60);
3     echo "Now: ". date ("Y-m-d") .<br>;
4     echo "Next Week: ". date ("Y-m-d", $nextWeek) .<br>;
5     echo "Next Week: ". date ("Y-m-d", strtotime("+1 week")) .<br>;
6 ?>
```

ผลลัพธ์

```
Now: 2012-11-26
Next Week: 2012-12-03
Next Week: 2012-12-03
```

จากตัวอย่างที่ 7.22 การใช้พังก์ชัน การใช้พังก์ชัน time() สำหรับคำนวณหัววัน อธิบายดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$nextWeek มีค่าเท่ากับ time() บวกด้วยเวลาใน 1 สัปดาห์ คือ 7 วัน คูณด้วย 24 ชั่วโมง คูณด้วย 60 นาที คูณด้วย 60 วินาที

บรรทัดที่ 3 แสดงผลวันที่ปัจจุบัน โดยใช้พังก์ชัน date()

บรรทัดที่ 4 แสดงผลวันที่ในสัปดาห์ถัดไป โดยใช้ฟังก์ชัน date()

บรรทัดที่ 5 แสดงผลวันที่ในสัปดาห์ถัดไป โดยใช้ฟังก์ชัน date() ร่วมกับตัวแปร
ฟังก์ชันแปลงข้อความเป็นเวลา (มีค่าเท่ากัน เมื่อตอนบรรทัดที่ 4)

บรรทัดที่ 6 สิ้นสุดสคริปต์ PHP

7.3.3 ฟังก์ชัน strtotime() เป็นฟังก์ชันที่ใช้สำหรับแปลงข้อความให้เป็นข้อมูลวันเวลา
ในรูปแบบ timestamp แต่ข้อความนั้นต้องเขียนในรูปแบบที่สามารถแปลงได้ ถ้าแปลงไม่ได้จะคืนค่า false
รูปแบบ

```
int strtotime ( string $time [, int $now = time() ] )
```

เมื่อ \$time หมายถึง วันและเวลาในรูปแบบข้อความ

\$now หมายถึง วันและเวลาปัจจุบันที่เรียกใช้ฟังก์ชัน

ตัวอย่างที่ 7.23 การใช้ฟังก์ชัน strtotime() สำหรับแปลงข้อความให้เป็นข้อมูลวันและเวลา

```
1 <?php
2     echo strtotime ("now"), "<br>";
3     echo strtotime ("10 September 2012"), "<br>";
4     echo strtotime ("+1 day"), "<br>";
5     echo strtotime ("+1 week"), "<br>";
6 ?>
```

ผลลัพธ์

```
1432783796
1347206400
1432870196
1433388596
```

จากตัวอย่างที่ 7.23 การใช้ฟังก์ชัน strtotime() สำหรับแปลงข้อความให้เป็นข้อมูลวัน
และเวลา ในรูปแบบ timestamp นั้น ค่าผลลัพธ์ที่ได้ จะยกเว้นการทำการคำนวณ เช่น
เปลี่ยนรูปแบบ timestamp เป็นวันที่จะต้องใช้ร่วมกับฟังก์ชัน date() ตัวอย่างดังนี้

ตัวอย่างที่ 7.24 การใช้ฟังก์ชัน strtotime() ร่วมกับฟังก์ชัน date()

```
1 <?php
2     echo date("Y-m-d", strtotime ("now")), "<br>";
3     echo date("Y-m-d", strtotime ("10 September 2012")), "<br>";
4     echo date("Y-m-d", strtotime ("+1 day")), "<br>";
5     echo date("Y-m-d", strtotime ("+1 week")), "<br>";
6 ?>
```



ผลลัพธ์

2012-05-28
2012-09-10
2012-05-29
2012-06-04

จากตัวอย่างที่ 7.24 การใช้ฟังก์ชัน strtotime() ร่วมกับฟังก์ชัน date() เพื่อแสดงผลในรูปแบบวันที่ที่ง่ายแก่การทำความเข้าใจ

7.3.4 ฟังก์ชัน date() เป็นฟังก์ชันที่ใช้สำหรับแสดงวันที่และเวลา โดยสามารถกำหนดรูปแบบของผลลัพธ์ของฟังก์ชันได้ เช่น ต้องการเฉพาะวันที่ หรือต้องการเฉพาะเวลา รูปแบบคำสั่งมีดังนี้

รูปแบบ

```
string date ( string $format [, int $timestamp = time() ] )
```

เมื่อ \$format หมายถึง รูปแบบวันและเวลา

\$timestamp หมายถึง พารามิเตอร์เสริมชนิดตัวเลขจำนวนเต็ม ของเวลาปัจจุบันบนระบบ Unix ถ้าไม่มีการกำหนดค่าใดๆ ค่าโดยบริยาย คือ ค่าของฟังก์ชัน time() หรือวันและเวลาปัจจุบัน

อักษรต่างๆ ที่ใช้กำหนดรูปแบบวันและเวลาสำหรับฟังก์ชัน date() ส่งผ่านพารามิเตอร์ \$format สามารถแยกออกเป็นกลุ่มได้ดังนี้

1) อักษรที่ใช้แทน วันและสัปดาห์

ตารางที่ 7.7 อักษรที่ใช้แทนความหมายของวันและสัปดาห์ สำหรับฟังก์ชัน date()

อักษร	ความหมาย
d	ใช้แทนวันที่ของเดือน โดยเป็นเลข 2 หลัก เช่น 01, 02, 31
j	ใช้แทนวันที่ของเดือน แบบไม่มี 0 นำหน้าเลขหลักเดียว เช่น 1, 2, 30
D	ใช้แทนวันในรอบสัปดาห์แบบย่อ เช่น Sun, Mon
l	ตัวแอลพิมพ์เล็กใช้แทนวันในรอบสัปดาห์แบบชื่อเต็ม เช่น Sunday, Monday
S	ตัวS(พิมพ์ใหญ่) เป็นตัวอักษรย่อที่ใช้บ่งบอกลำดับที่ของวันในรอบเดือน เช่น st, nd, rd, th โดยปกติจะใช้ต่อท้ายตัว j หรือ d (ใช้แทนวันที่) เช่น JS ลักษณะผลลัพธ์คือ 1st
w	ใช้แทนลำดับของวันในรอบสัปดาห์จาก 0-6
z	แทนลำดับของวันในรอบปีจาก 0-365
W	ลำดับของสัปดาห์ในรอบปี

2) อักษรที่ใช้แทนความหมายเดือน สำหรับฟังก์ชัน date ()

ตารางที่ 7.8 อักษรที่ใช้แทนความหมายของเดือน สำหรับฟังก์ชัน date ()

อักษร	ความหมาย
F	ชื่อเดือนแบบเต็ม เช่น January, March
M	ลำดับที่ของเดือนในรอบปีโดยมี 0 นำหน้าเลขหลักเดียว เช่น 01, 02, 12
M	ชื่อของเดือนแบบย่อ เช่น 3 ตัวแรก เช่น Jan, Feb
n	ลำดับที่ของเดือนแบบไม่มี 0 นำหน้าเลขหลักเดียว เช่น 1, 2, 12
t	จำนวนวันของเดือนนั้นๆ โดยมีค่าระหว่าง 28- 31

3) อักษรที่ใช้แทนความหมายของปี สำหรับฟังก์ชัน date ()

ตารางที่ 7.9 อักษรที่ใช้แทนความหมายของปี สำหรับฟังก์ชัน date ()

อักษร	ความหมาย
L	ตรวจสอบว่าเป็นปี Leap Year หรือไม่(เดือนกุมภาพันธ์มี 29 วัน) ถ้าใช่จะคืนค่า 1 ถ้าไม่ใช่จะคืนค่า 0
Y	ใช้แทนปี ค.ศ. แบบเลข 4 หลัก เช่น 1975
y	ใช้แทนปี ค.ศ. แบบเลข 2 หลัก เช่น 75

4) อักษรที่ใช้แทนความหมายของเวลา สำหรับฟังก์ชัน date ()

ตารางที่ 7.10 อักษรที่ใช้แทนความหมายของเวลา สำหรับฟังก์ชัน date ()

อักษร	ความหมาย
a	ใช้แทนค่า Ante Meridiem และ Post Meridiem แบบพิมพ์เล็ก ได้แก่ am หรือ pm
A	ใช้แทนค่า Ante Meridiem และ Post Meridiem แบบพิมพ์ใหญ่ ได้แก่ AM หรือ PM
g	ใช้แทนค่าชั่วโมงของวันแบบ 12 ชั่วโมง โดยไม่มี 0 นำหน้าเลขหลักเดียวยาว 0-12
G	ใช้แทนค่าชั่วโมงของวันแบบ 24 ชั่วโมง โดยไม่มี 0 นำหน้าเลขหลักเดียวยาว 0-23
h	ใช้แทนค่าชั่วโมงของวันแบบ 12 ชั่วโมง โดยไม่มี 0 นำหน้าเลขหลักเดียวยาว 01-12
H	ใช้แทนค่าชั่วโมงของวันแบบ 24 ชั่วโมง โดยไม่มี 0 นำหน้าเลขหลักเดียวยาว 00-23
i	ค่านาทีแบบมี 0 นำหน้าเลขหลักเดียวจาก 00-59
s	ค่าวินาทีแบบมี 0 นำหน้าเลขหลักเดียวจาก 00-59
O	ค่าความแตกต่างเมื่อเทียบกับเวลา Greenwich(GMT) เช่น +0700
r	เวลาตามมาตรฐานของ RFC 2822 เช่น Sun, 1 Mar 2009 15:03:57 +0700

ตัวอย่างที่ 7.25 การแสดงวันที่และเวลาในรูปแบบต่างๆ ด้วยฟังก์ชัน date () แบบที่ 1

```
1 <?php
2     date_default_timezone_set('Asia/Bangkok');
```



```

3 echo "การแสดงวันที่และเวลาในรูปแบบต่างๆ ด้วยฟังก์ชัน date () <br>";
4 echo date ("r") . "<br>";
5 echo date ('l jS \of F Y h:i:s A') . "<br>";
6 echo "July 1, 2012 is on a " . date ("l", mktime(0, 0, 0, 7, 1, 2012)) . "<br>";
7 ?>

```

ผลลัพธ์แบบที่ 1

การแสดงวันที่และเวลาในรูปแบบต่างๆ ด้วยฟังก์ชัน date ()

Mon, 26 Nov 2012 16:36:34 +0700

Monday 26th of November 2012 04:36:34 PM

July 1, 2012 is on a Sunday

ตัวอย่างที่ 7.26 การแสดงวันและเวลาในรูปแบบต่างๆ ด้วยฟังก์ชัน date () แบบที่ 2

```

1 <?php
2     date_default_timezone_set('Asia/Bangkok');
3     $birth = strtotime ("12/10/1978");
4     echo date ("ข้าพเจ้าเกิดเมื่อ j-m-Y <br>", $birth);
5     $day = array("อาทิตย์", "จันทร์", "อังคาร", "พุธ", "พฤหัสบดี", "ศุกร์", "เสาร์");
6     $month =array("มกราคม", "กุมภาพันธ์", "มีนาคม", "เมษายน", "พฤษภาคม", "มิถุนายน",
"กรกฎาคม", "สิงหาคม" , "กันยายน" , "ตุลาคม" , "พฤศจิกายน" , "ธันวาคม");
7     $d = date ("w");
8     $day = $day [$d];
9     $date = date ("j");
10    $m = date ("m")-1;           // เพราะลำดับเดือนจะมีค่าระหว่าง 1-12
11    $month = $months [$m];
12    $year = date ('Y') + 543;
13    echo "วันนี้ตรงกับวัน $day วันที่ $date เดือน $month ปี $year ";
14    echo date("ขณะนี้เวลา H:i:s");
15 ?>

```

ผลลัพธ์แบบที่ 2

ข้าพเจ้าเกิดเมื่อ 10-12-1978

วันนี้ตรงกับวัน จันทร์ วันที่ 26 เดือน ปี 2555 ขณะนี้เวลา 16:43:48

7.3.5 ฟังก์ชัน getdate () เป็นฟังก์ชันที่ใช้สำหรับแสดงวันที่และเวลาเช่นเดียวกับฟังก์ชัน date () แต่ไม่สามารถกำหนดรูปแบบเองได้ และฟังก์ชันจะคืนค่าเป็นอาร์เรย์แบบคิล์และค่าอาร์เรย์ รูปแบบมีดังนี้

รูปแบบ

```
array getdate ([ int $timestamp = time() ] )
```

เมื่อ \$timestamp หมายถึง พารามิเตอร์ชนิดตัวเลขจำนวนเต็ม ของเวลาปัจจุบันบนระบบ Unix ถ้าไม่มีการกำหนดค่าใดๆ ค่าโดยปริยาย คือ ค่าของฟังก์ชัน time () หรือวันและเวลาปัจจุบัน

อาร์เรย์ที่ได้จากฟังก์ชัน getdate () ส่งคืนค่า ประกอบด้วยสมาชิกของอาร์เรย์ ดังต่อไปนี้

ตารางที่ 7.11 คีย์และค่าของอาร์เรย์ที่ได้จากการใช้ฟังก์ชัน getdate ()

คีย์	คำอธิบาย	ลักษณะค่าที่ได้
seconds	อ่านค่า “วินาที”	0-59
minutes	อ่านค่า “นาที”	0 to 59
hours	อ่านค่า “ชั่วโมง”	0 to 23
mday	อ่านค่า “วันที่”	1 to 31
wday	อ่านค่าลำดับของวันในรอบสัปดาห์	0 (อาทิตย์)-6 (เสาร์)
mon	อ่านค่าลำดับของเดือนในรอบปี	1-12
Year	อ่านค่า “ปี” แบบเลข 4 หลัก	เช่น 2010
yday	อ่านค่าลำดับของวันในรอบปี	0-365
weekday	อ่านค่า “ชื่อวัน” ของสัปดาห์	Sunday-Saturday
month	อ่านค่า “ชื่อเดือน” ของปี	January-December

ตัวอย่างที่ 7.27 การใช้ฟังก์ชัน getdate ()

```
<?php
    $today = getdate ( );
    print_r ($today);
?>
```

ผลลัพธ์

```
Array ( [seconds] => 30 [minutes] => 55 [hours] => 16 [mday] => 26 [wday] => 1 [mon] => 11 [year] => 2012 [yday] => 330 [weekday] => Monday [month] => November [0] => 1353923730 )
```

7.3.6 ฟังก์ชัน mktime () เป็นฟังก์ชันที่ใช้สำหรับหาค่า Unix Timestamp ของวันที่และเวลา เช่นเดียวกับฟังก์ชัน time () แต่ฟังก์ชัน time () หาค่า Unix Timestamp ของวันที่และเวลาปัจจุบัน เท่านั้น ส่วนฟังก์ชัน mktime () สามารถกำหนดวันและเวลาที่ต้องการหาค่า Unix Timestamp ได้ รูปแบบคำสั่งมีดังนี้



รูปแบบ

```
int mktime ([ int $hour = date("H") [, int $minute = date("i") [, int $second = date("s")
[, int $month = date("n") [, int $day = date("j") [, int $year = date("Y") [, int $is_dst = -1 ]]]]]]) )
```

เมื่อ \$hour หมายถึง ชั่วโมง

\$minute หมายถึง นาที

\$second หมายถึง วินาที

\$month หมายถึง เดือน

\$day หมายถึง วัน

\$year หมายถึง ปี

\$is_dst หมายถึง การกำหนดรูปแบบวันและเวลาตามฤดูกาล (กำหนดพารามิเตอร์เป็น 1) ไม่กำหนดเวลา (กำหนดพารามิเตอร์เป็น 0) หรือหากไม่ทราบ สามารถกำหนดเป็น -1 หรือไม่กำหนด จะมีค่าปริยายเท่ากับ -1

ตัวอย่างที่ 7.28 การใช้ฟังก์ชัน mktime ()

```
1 <?php
2     date_default_timezone_set ('Asia/Bangkok');
3     echo date("M-d-Y", mktime(0, 0, 0, 12, 32, 2012)) . "<br>";
4     echo date("M-d-Y", mktime(0, 0, 0, 13, 1, 2012)). "<br>";
5     echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 2013)). "<br>";
6     echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 13));
7 ?>
```

ผลลัพธ์

Jan-01-2013

Jan-01-2013

Jan-01-2013

Jan-01-2013

7.3.7 ฟังก์ชัน checkdate () เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบความถูกต้องของวันที่ เช่น วันที่ที่ถูกต้องจะต้องไม่เกิน 31 วัน หรือเดือนจะต้องเป็นตัวเลขที่อยู่ระหว่าง 1 ถึง 12 หากฟังก์ชันตรวจสอบแล้วพบว่าวันที่ไม่ถูกต้อง จะคืนค่าเป็นเท็จ (False) จากถูกต้องจะคืนค่าเป็นจริง (True) รูปแบบคำสั่งมีดังนี้

รูปแบบ

```
bool checkdate ( int $month , int $day , int $year )
```

เมื่อ \$month หมายถึง จำนวนเดือนใน 1 ปี มีค่าอยู่ระหว่าง 1 - 12 เดือน

\$day หมายถึง จำนวนวันของแต่ละเดือน

\$year หมายถึง ปี มีค่าอยู่ระหว่าง 1 - 32767

ตัวอย่างที่ 7.29 การใช้ฟังก์ชัน checkdate () สำหรับตรวจสอบความถูกต้องของวันที่

```

1 <?php
2     var_dump (checkdate (12, 31, 2012)); // เป็นจริง เพราะเดือนธันวาคมมี 31 วัน ในปี 2012
3     var_dump (checkdate (2, 29, 2013)); // เป็นเท็จ เพราะเดือนกุมภาพันธ์ในปี 2013 มี 28 วัน
4 ?>

```

จากตัวอย่างที่ 7.29 การใช้ฟังก์ชัน checkdate () สำหรับตรวจสอบความถูกต้องของวันที่ในบรรทัดที่ 2 จะได้ผลเป็นจริง เพราะเดือนธันวาคม ในปี ค.ศ.2012 มี 31 วัน สำหรับผลของบรรทัดที่ 3 จะได้ผลเป็นเท็จ เพราะเดือนกุมภาพันธ์ ในปี ค.ศ.2012 มี 28 วัน

7.3.8 การจัดการกับวันที่และเวลาแบบ Greenwich Mean Time

1) ฟังก์ชัน gmdate () เป็นฟังก์ชันที่ใช้แสดงวันที่และเวลาเหมือนกับฟังก์ชัน date () แต่จะคืนค่าเป็นเวลา GMT โดยสามารถกำหนดรูปแบบวันที่และเวลาได้เหมือนกับฟังก์ชัน date () มีรูปแบบการใช้งานฟังก์ชัน ดังนี้

รูปแบบ

```
string gmdate ( string $format [, int $timestamp = time() ] )
```

เมื่อ \$format หมายถึง รูปแบบวันและเวลา

\$timestamp หมายถึง พารามิเตอร์เสริมชนิดตัวเลขจำนวนเต็ม ของเวลาปัจจุบันระบบ Unix ถ้าไม่มีการกำหนดค่าใดๆ ค่าโดยเบริယาย คือ ค่าของฟังก์ชัน time () หรือวันและเวลาปัจจุบัน

ตัวอย่างที่ 7.30 การใช้ฟังก์ชัน gmdate ()

```

1 <?php
2     echo date("M d Y H:i:s", mktime(0, 0, 0, 1, 1, 2012)) . "<br>";
3     echo gmdate("M d Y H:i:s", mktime(0, 0, 0, 1, 1, 2012));
4 ?>

```

ผลลัพธ์

Jan 01 2012 00:00:00

Dec 31 2011 17:00:00

2) ฟังก์ชัน gmstrftime () เป็นฟังก์ชันที่ใช้แสดงวันที่และเวลาเหมือนกับฟังก์ชัน strftime () แต่จะคืนค่าเป็นเวลา GMT มีรูปแบบ ดังนี้



รูปแบบ

```
int gmmktime ([ int $hour = gmdate("H") [, int $minute = gmdate("i") [, int $second  
= gmdate("s") [, int $month = gmdate("n") [, int $day = gmdate("j") [, int $year =  
gmdate("Y") [, int $is_dst = -1 ]]]]]]) )
```

ตัวอย่างที่ 7.31 การใช้งานฟังก์ชัน gmmktime ()

```
<?php  
echo "July 1, 2012 is on a " . date ("l", gmmktime (0, 0, 0, 7, 1, 2012));  
?> // ผลลัพธ์ คือ July 1, 2012 is on a Sunday
```

7.3.9 การประยุกต์ใช้งานฟังก์ชันที่เกี่ยวข้องกับวันที่และเวลา

1) การประยุกต์ใช้ฟังก์ชันวันที่และเวลา เพื่อสร้างปฏิทินออนไลน์ เป็นการประยุกต์ใช้ฟังก์ชันวันที่และเวลาที่ได้กล่าวถึงไว้ ตัวอย่างดังนี้

ตัวอย่างที่ 7.32 การประยุกต์ใช้ฟังก์ชันเพื่อทำปฏิทินออนไลน์ส่วนของโครงสร้าง CSS tags เพื่อควบคุมการแสดงผลข้อมูลในรูปแบบตาราง

```
<style type="text/css">  
div.holder{  
    position:relative;  
    font-family:tahoma, "Microsoft Sans Serif", Vanessa;  
    border:2px solid #999;  
    float:left;  
    font-size:12px;  
    width:175px;  
    padding:5px;  
}  
div.month {  
    position:relative;  
    display:block;  
    height:18px;  
    width:100%;  
    float:left;  
    left:0;  
    top:0;
```

```

background:#567;
color:#fff;
border-bottom:2px solid #89a;
text-align:center;
}

div.wkday {
    position:relative;
    clear:both;
    float:left;
    height:18px;
    display:block;
    width:100%;
    background:#567;
    color:#fff;
    border-bottom:3px solid #234;
}

div.wkday span {
    display:block;
    float:left;
    width:25px;
    text-align:center;
}

div.box_day {
    position:relative;
    width:100%;
    clear:both;
    float:left;
    background:#EEEEEE;
    color:#000;
    word-spacing:8px;
    text-indent:3px;
}

```



</style>

ตัวอย่างที่ 7.33 การประยุกต์ใช้ฟังก์ชันเพื่อทำปฏิทินออนไลน์ส่วนของ PHP สคริปต์ เพื่อ
ประมวลผลและแสดงผลร่วมกับ CSS tags

```
<body>
<?php
$thai_month_arr=array ("0"=>"", "1"=>"มกราคม", "2"=>"กุมภาพันธ์", "3"=>"มีนาคม",
"4"=>"เมษายน", "5"=>"พฤษภาคม", "6"=>"มิถุนายน",
"7"=>"กรกฎาคม", "8"=>"สิงหาคม", "9"=>"กันยายน",
"10"=>"ตุลาคม", "11"=>"พฤศจิกายน", "12"=>"ธันวาคม");

$now_month = date ("Y-m-01");
$mk_time = strtotime ($now_month);
$day_no = date ("t",$mk_time);
$wan_no = date ("w",$mk_time);
$box_day = $day_no+$wan_no;
$rows_week = ceil ($box_day/7);
$total_box = $rows_week * 7;

function get_day ($no_day,$wan_no,$day_no) {
    $wan_tee = $no_day-$wan_no;
    if($wan_tee<=0) {
        $wan_tee="__";
        return $wan_tee;
    } else {
        if ($wan_tee <= $day_no) {
            return str_pad ($wan_tee,2,"0",STR_PAD_LEFT);
        } else {
            return "__";
        }
    }
}

?>
<div class="holder">
```

```

<div class="month">
<?=$thai_month_arr [intval (date ("m"))] ?> <?=date ("Y")+543 ?>
</div>

<div class="wkday">
    <span>ອາ</span>
    <span>ຈັ</span>
    <span>ອຸ</span>
    <span>ພິ</span>
    <span>ພຸ</span>
    <span>ສີ</span>
    <span>ສົ</span>
</div>

<?php
    for ($i = 1; $i <= $total_box; $i++) {
?
<?php
    if ($i%7 == 1) echo "<div class='box_day'>";
?
<?=get_day ($i,$wan_no,$day_no) ?>
<?php
    if ($i%7 == 0 || $i == $total_box ) {
        echo "</div>n";
    }
?
<?php } ?>
</div>
<br style="clear:both;" />
</body>

```



พฤษจิกายน 2555						
อา	จ	อ	พ	พุ	ศ	ส
			01	02	03	
๙๙๙๙๙						
04	05	06	07	08	09	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

ภาพที่ 7.1 แสดงตัวอย่างผลลัพธ์การสร้างปฏิทินออนไลน์

2) การประยุกต์ใช้ฟังก์ชันวันที่และเวลา เพื่อคำนวณอายุ มีตัวอย่าง ดังนี้

ตัวอย่างที่ 7.34 การประยุกต์โปรแกรมคำนวณอายุ กรณีศึกษาแบบที่ 1

```

1 <?php
2     function AgeCalculator ($year,$month,$day) {
3         $birthday_convert = mktime (1, 1, 1, $month, $day, $year);
4         return (floor ((time ( ) - $birthday_convert) / 31556926));
5     }
6     echo AgeCalculator (1978, 1, 2);           // ผลลัพธ์จะได้ 35
7 ?>

```

จากตัวอย่างที่ 7.34 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชันวันที่และเวลาเพื่อคำนวณหาอายุ อธิบายดังนี้

บรรทัดที่ 6 แสดงผลค่าการเรียกใช้ฟังก์ชัน AgeCalculator () โดยพารามิเตอร์ที่ส่งประกอบด้วย ปี ค.ศ.เกิด เดือนเกิด และวันที่เกิด ตัวอย่างค่าที่ส่งเข้าไปประกอบด้วย ปี ค.ศ.1978 เดือนมกราคม วันที่ 2 เรียกใช้ฟังก์ชันด้วย AgeCalculator (1978, 1, 2)

บรรทัดที่ 2 เมื่อฟังก์ชัน AgeCalculator () ถูกเรียกใช้ ฟังก์ชัน AgeCalculator () กำหนดรับพารามิเตอร์ 3 ค่า ประกอบด้วย \$year, \$month และ \$day โดยค่าถูกส่งมาจากการบรรทัดที่ 6 ดังนั้นหมายความว่า กำหนดค่า \$year มีค่าเท่ากับ 1978 กำหนดค่า \$month มีค่าเท่ากับ 1 และ \$day มีค่าเท่ากับ 2 ตามลำดับ

บรรทัดที่ 3 กำหนดให้ตัวแปร \$birthday_convert มีค่าเท่ากับผลของฟังก์ชัน mktime () เพื่อแปลงวัน เดือน ปี ค.ศ. เกิด เป็นรูปแบบของ timestamp จะมีค่าเท่ากับ 252525661

บรรทัดที่ 4 ส่งผลการประมวลผลกลับไปยังส่วนที่เรียกใช้ คือ บรรทัดที่ 6 การประมวลผลนิพจน์ ประกอบด้วยค่าดังนี้ time () คือ (วันและเวลาปัจจุบัน – วัน เดือน ปี เกิด) ทั้งหมดหารด้วย 31556926 (เมื่อ 1 ปี มีค่าเท่ากับ 31556926 วินาที) ผลหารจะออกมาอยู่ในรูปเลขทศนิยม ดังนั้นจึงใช้ฟังก์ชัน floor () เพื่อตัดทศนิยม

หมายเหตุ ถ้าในกรณีจะเขียนฟังก์ชันรับค่าเป็นปี พ.ศ. จะต้องลบออกด้วย 543 เพื่อแปลงเป็น ปี ค.ศ. ก่อนเข้าสู่นิพจน์

7.3.10 การเทียบเวลาวินาที

การเทียบเวลาวินาทีเพื่อความสะดวกในการพัฒนาโปรแกรมที่เกี่ยวข้องกับเวลา มีรายละเอียด ดังนี้ (วิกิพีเดีย สารานุกรมเสรี. 2556)

1 มิลลิวินาที เท่ากับ 0.001 วินาที

1 วินาที เท่ากับ 1 วินาที

1 นาที เท่ากับ 60 วินาที

1 ชั่วโมง เท่ากับ 3600 วินาที

1 วัน เท่ากับ 86400 วินาที

1 สัปดาห์ เท่ากับ 604800 วินาที

1 เดือน เท่ากับ 2629743.83 วินาที

1 ปี เท่ากับ 31556926 วินาที

1 ปีปกติสุรทิน เท่ากับ 31536000 วินาที (ปีปกติที่มี 365 วัน)

1 ปีอธิกสุรทิน เท่ากับ 31622400 วินาที (เป็นปีที่มีการเพิ่มหนึ่งวัน ผลมาจากการเดือนกุมภาพันธ์ ในปีอธิกสุรทินมี 29 วัน แทนที่จะมี 28 วันตามปกติ ดังนั้น ปีดังกล่าวจึงมี 366 วัน แทนที่จะมี 365 วัน ตามปกติ)

สรุป

กระบวนการทำงานของภาษา PHP ตั้งแต่นำข้อมูลเข้า ประมวล และแสดงผลลัพธ์ ชนิดของข้อมูลที่จำเป็นต้องข้องเกี่ยวอยู่บ่อยครั้ง นั่นก็คือ ข้อมูลชนิดข้อความ ตัวเลข วันและเวลา ลักษณะข้อมูล ดังกล่าวลือเป็นพื้นฐานสำคัญ ดังนั้นภาษา PHP จึงมีฟังก์ชันการทำงานที่สนับสนุนการประมวลผล ข้อมูล ดังกล่าวอยู่เป็นจำนวนมาก ผู้พัฒนาสามารถประยุกต์ใช้ฟังก์ชันที่มีอยู่ในรูปแบบต่างๆ เพื่อประมวลผลให้ได้ผลลัพธ์ตามต้องการ เช่น การนับจำนวนตัวอักษร การนับคำ การค้นหาข้อความ การเปลี่ยนรูปแบบตัวพิมพ์ การแยกและเชื่อมต่อข้อความ การจัดการรูปแบบการแสดงผลตัวเลข การแสดงผลวันที่และเวลา การคำนวณหารวันเกิด และปฏิทินออนไลน์ เป็นต้น

คำถามท้ายบท

1. จงอธิบายฟังก์ชันที่ใช้สำหรับขนาดของข้อความ พร้อมยกตัวอย่างประกอบ
2. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากสคริปต์



```
1 <?php  
2     $variable = "Computer Science of Suratthani Rajabhat University";  
3     $pieces = explode (" ", $variable);  
4     echo $pieces [0];  
5     echo $pieces [3];  
6 ?>
```

3. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการสคริปต์

```
1 <?php  
2     $array = array("Color"=>"yellow", "red", "pink", "black", "blue");  
3     $array = ucwords( implode (" ", $array));  
4     echo $array;  
5 ?>
```

4. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการสคริปต์

```
1 <?php  
2     function digit ($bit) {  
3         for ($loop = 0, $number = 0; $loop <= $bit; $loop++){  
4             $number = $number + pow (2, $loop);  
5         }  
6         return $number;  
7     }  
8     echo digit(7);  
9 ?>
```

5. จงอธิบาย หน้าที่ของฟังก์ชัน strtotime () รูปแบบการใช้งาน และยกตัวอย่างการประยุกต์ใช้งาน ไม่น้อยกว่า 1 ตัวอย่าง

บทที่ 8

การจัดการไฟล์และไดเรกทอรี

การจัดเก็บข้อมูลของคอมพิวเตอร์จะทำในรูปแบบของไฟล์ ทั้งนี้ในกระบวนการสร้างเว็บในบางครั้งมีความจำเป็นต้องใช้ความรู้เรื่องการจัดการไฟล์ (File) และไดเรกทอรี (Directory) มาเกี่ยวข้องดังนั้นในบทนี้จึงจะกล่าวถึงเรื่องราวพื้นฐานที่จำเป็นที่ควรรู้เกี่ยวกับไฟล์และไดเรกทอรี เช่น การอ้างอิงตำแหน่ง การสร้าง การลบ การเปลี่ยนชื่อ การเคลื่อนย้าย การเปิด และการอ่าน เป็นต้น

8.1 การอ้างอิงตำแหน่งไฟล์และไดเรกทอรี

ในการสร้างเว็บเพจนั้น บางครั้งจำเป็นต้องนำข้อมูลจากไฟล์ต่างๆ เข้ามาใช้ร่วมด้วย เช่น รูปภาพ หรือการนำไฟล์เข้ามาแทรก ทั้งนี้ไฟล์แต่ละชนิด มักแยกเก็บให้เป็นสัดส่วนเฉพาะ เช่น ไฟล์ภาพ อาจเก็บไว้ในไดเรกทอรี images หรือ pictures หรือไฟล์สำหรับนำไปแทรกจะเก็บไว้ในไดเรกทอรี inc เป็นต้น ด้วยเหตุนี้จึงมักเกิดปัญหาในการอ้างอิง ตำแหน่งไฟล์ที่อยู่ต่างไดเรกทอรี ดังนั้นก่อนที่จะศึกษาเรื่องอื่นๆ จะต้องทำความเข้าใจเกี่ยวกับการอ้างอิงตำแหน่งของไฟล์และไดเรกทอรีก่อน มีรายละเอียด ดังนี้

8.1.1 การอ้างถึง Document Root

Document Root คือ ไดเรกทอรีเริ่มต้นของการจัดเก็บข้อมูลของเว็บไซต์ ตำแหน่งนี้ปกติแล้วจะขึ้นอยู่กับการตั้งค่าและปรับแต่งระบบโดยผู้ดูแลระบบ ทั้งนี้หากเป็นผู้ติดตั้งระบบเองด้วย AppServ ตามค่าปกติจะกำหนดที่อยู่ไว้ ดังนี้

C:/AppServ/www

สำหรับตำแหน่งนี้ถ้าเป็นรูปแบบ URL จะเท่ากับ <http://localhost> หรือ <http://127.0.0.1> นั้นเอง ตามรูปแบบของระบบเว็บจะแทนตำแหน่งนี้ด้วยเครื่องหมาย / และตามด้วยตำแหน่งอ้างอิงตำแหน่งไฟล์ และไดเรกทอรี เป็นต้นไป เช่น “/myphp/example/index.php”

8.1.2 การอ้างอิงพาราแบบ Absolute

พาร คือ เส้นทางไปยังไฟล์หรือไดเรกทอรี การอ้างอิงแบบ Absolute จะเป็นการระบุตำแหน่งที่แท้จริงของไฟล์ หรือไดเรกทอรี โดยเริ่มจาก Document Root เป็นต้นไป

ลักษณะการอ้างอิงไฟล์ภาพต่างๆ จะเป็นดังนี้

```
<img src = “/myphp/img/logo.png” />
<a href = “/myphp/example/login.php”>เข้าสู่ระบบ </a>
```

จะเห็นว่าการอ้างอิงพาราแบบ Absolute นี้ต้องไปเริ่มจาก Document Root เสมอ (เริ่มพารด้วยเครื่องหมาย /) และตามด้วยไดเรกทอรีตามลำดับจนถึงไฟล์ที่ต้องการ ไม่ว่าจะนำไฟล์ไปใช้ที่เว็บเพจ

โดยทั่วไป เผราระบบจะอ้างอิงจากตำแหน่ง Document Root เสมอ อย่างไรก็ตามการระบุพาร์เซนต์ Absolute นี้ จะใช้ได้เฉพาะกับ HTML tags เท่านั้น หากใช้ร่วมกับฟังก์ชันต่างๆ ของระบบไฟล์ด้วยภาษา PHP จะเกิดปัญหา เว้นแต่จะระบุตำแหน่ง Document Root ก่อนการเรียกใช้งาน ตัวอย่างดังนี้

ตัวอย่างที่ 8.1 การระบุตำแหน่ง Document Root ก่อนการใช้งาน

```
$root = $_SERVER['DOCUMENT_ROOT'];
$path = "myphp/example/inc/header.inc.php";
include ("$root . $path");
//ลักษณะผลลัพธ์ C:/AppServ/www/myphp/example/inc/header.inc.php
```

8.1.3 การอ้างอิงพาร์เซนต์ Relative

การอ้างอิงพาร์เซนต์ Relative จะอ้างอิงโดยการเทียบกับตำแหน่งไฟล์ที่เป็นผู้เรียกไฟล์อื่นมาใช้เป็นหลัก โดยมีสัญลักษณ์การเปรียบเทียบตำแหน่งเพิ่มเติมเข้ามาอีก 2 รูปแบบ ดังนี้

- 1) การใช้สัญลักษณ์ .. (จุดหนึ่งจุด) ใช้แทนไดเรกทอรีปัจจุบัน
- 2) การใช้สัญลักษณ์ .. (จุดสองจุด) ใช้แทนไดเรกทอรีก่อนหน้านี้ 1 ระดับ เช่น หากปัจจุบันอยู่ที่ไดเรกทอรี /myphp/example ถ้าใช้ .. จะหมายถึงการอ้างอิงไปยังตำแหน่งไดเรกทอรี /myphp เป็นต้น

8.2 การจัดการกับไฟล์

การจัดเก็บข้อมูลลงในไฟล์ทำได้ 2 วิธี คือ 1) จัดเก็บในรูปของแฟลตไฟล์ (Flat File) หรือเทกซ์ไฟล์ (Text File) และ 2) จัดเก็บในรูปแบบของฐานข้อมูล (Database) แต่ไม่ว่าจะมีการจัดเก็บแบบใดก็ตาม จำเป็นต้องมีการเขียน และอ่านข้อมูลในไฟล์เสมอ โดยมีขั้นตอนดังนี้

ตารางที่ 8.1 ขั้นตอนการเขียนและอ่านข้อมูลในไฟล์

การทำงาน	ขั้นตอน	การทำงาน	ขั้นตอน
การเขียน (Writing)	<ol style="list-style-type: none"> 1. เปิดไฟล์ที่ต้องการหรือสร้างไฟล์ใหม่ 2. เขียนข้อมูลลงไฟล์ 3. ปิดไฟล์ 	การอ่าน (Reading)	<ol style="list-style-type: none"> 1. เปิดไฟล์ 2. อ่านข้อมูลจากไฟล์ 3. ปิดไฟล์

ในหัวข้อนี้จะกล่าวถึงการจัดการกับไฟล์ประเภทเทกซ์ไฟล์ มีฟังก์ชันในการจัดการดังต่อไปนี้

8.2.1 การเปิดไฟล์

การอ่านข้อมูลจากไฟล์หรือบันทึกข้อมูลลงไฟล์ จะต้องเริ่มด้วยการเปิดไฟล์ขึ้นมาก่อนเสมอ โดยใช้ฟังก์ชัน fopen () สำหรับการเปิดไฟล์เพื่อการทำงานใดๆ นั้น สามารถระบุได้ในส่วนของโหมดไฟล์ (File Mode) ถ้าหากสามารถเปิดไฟล์ได้จะคืนค่าเป็นตำแหน่งของตัวชี้ตำแหน่งที่อยู่ใน

ไฟล์ข้อมูล (Resource) หากเปิดไม่ได้ หรือไม่พบไฟล์ที่ระบุ จะคืนค่าเป็นเท็จ รูปแบบการใช้งานฟังก์ชัน fopen () ดังนี้

รูปแบบ

```
resource fopen ( string $filename, string $mode [, bool $use_include_path = false] )
```

เมื่อ \$filename หมายถึง ชื่อไฟล์ที่ต้องการเปิด

\$mode หมายถึง โหมดไฟล์ ใช้ระบุว่าเปิดไฟล์เพื่อวัตถุประสงค์ใด (ดูค่าของโหมดไฟล์ได้จากตารางที่ 8.2)

\$use_include_path หมายถึง หากกำหนดพารามิเตอร์เป็น 1 หรือ True ฟังก์ชันจะค้นหาไฟล์จากไดเรกทอรีที่กำหนดใน "include_path" (กำหนดค่าในไฟล์ php.ini)

ตารางที่ 8.2 โหมดไฟล์

โหมดไฟล์	ความหมาย
r	โหมดอ่าน – เปิดไฟล์สำหรับการอ่าน เริ่มทำงานจากตอนต้นไฟล์
r+	โหมดอ่าน – เปิดไฟล์สำหรับการอ่านและเขียน เริ่มทำงานจากตอนต้นไฟล์
w	โหมดเขียน – เปิดไฟล์สำหรับการเขียน เริ่มทำงานจากตอนต้นไฟล์ ถ้าไฟล์มีอยู่แล้วจะลบข้อมูลที่มีอยู่ ถ้าไม่มีไฟล์จะสร้างขึ้นใหม่
w+	โหมดเขียน – เปิดไฟล์สำหรับการเขียนและอ่าน เริ่มทำงานจากตอนต้นไฟล์ ถ้าไฟล์มีอยู่แล้วจะลบข้อมูลที่มีอยู่ ถ้าไม่มีไฟล์จะสร้างขึ้นใหม่
a	โหมดเพิ่ม – เปิดไฟล์สำหรับการเพิ่ม (เขียน) เริ่มทำงานจากจุดสิ้นสุดของข้อมูลที่มีอยู่ ถ้าไม่มีไฟล์จะได้รับการสร้างขึ้นใหม่
a+	โหมดเพิ่ม – เปิดไฟล์สำหรับการเพิ่ม (เขียน) และอ่าน เริ่มทำงานจากจุดสิ้นสุดของข้อมูลที่มีอยู่ ถ้าไม่มีไฟล์จะได้รับการสร้างขึ้นใหม่
b	โหมดไบนารี – ใช้ร่วมกับโหมดอื่น โดยอาจจะจำเป็น ถ้าระบบไฟล์แยกระหว่างไฟล์ไบนารีและเทกซ์ไฟล์ ระบบ Windows มีการแยก แต่ระบบ Unix ไม่มีการแยก
t	ใช้ร่วมกับโหมดอื่นๆ ที่กล่าวมาข้างต้น เพื่อเป็นการเปิดไฟล์ในโหมดเทกซ์ไฟล์รวมๆ

หมายเหตุ

ตัวชี้ตำแหน่งข้อมูลในไฟล์ (File Pointer) หมายถึง ตัวแปรที่กำหนดขึ้นเพื่อใช้สำหรับการซื้อตำแหน่งข้อมูลในไฟล์ เมื่อเปิดไฟล์ ตัวชี้ตำแหน่งไฟล์ข้อมูลจะถูกกำหนดให้ชี้ไปยังข้อมูลแรกในไฟล์ โดยทุกครั้งหลังจากการอ่านหรือเขียนข้อมูลในไฟล์ ตัวชี้จะเลื่อนไปยังข้อมูลที่อยู่ตำแหน่งถัดไปโดยอัตโนมัติ



ตัวอย่างที่ 8.2 การเปิดไฟล์เพื่อเขียน

```

1 <?php
2 $fp = fopen ("test.txt", "w");
3 ?>

```

จากตัวอย่างที่ 8.2 การเปิดไฟล์เพื่อเขียนมีการกำหนดโหมดเป็น “w” เป็นโหมดสำหรับเปิดไฟล์เพื่อเขียน โดยถ้าไฟล์มีอยู่แล้วจะลบข้อมูลทั้งหมด ถ้าไม่มีไฟล์จะสร้างไฟล์ขึ้นใหม่ แต่หากต้องการเพิ่มข้อมูลต่อท้ายเรื่อยๆ จะต้องเปลี่ยนไฟล์โหมดตามตัวอย่างที่ 8.3

ตัวอย่างที่ 8.3 การเปิดไฟล์เพื่อเขียนต่อท้าย

```

1 <?php
2 $fp = fopen ("test.txt", "a");
3 ?>

```

จากตัวอย่างที่ 8.2 และ 8.3 กำหนดให้ตัวแปร \$fp ทำหน้าที่เป็นตัวชี้ตำแหน่งข้อมูลในไฟล์ ซึ่งเป็นผลมาจากการใช้ฟังก์ชัน fopen () หากไฟล์ที่ต้องการเปิดอยู่คุณจะไดเรกทอรี จะต้องระบุไดเรกทอรีของไฟล์ที่ต้องการเปิดด้วย เช่น "C:/AppServ/www/file/test.txt" โดย "test.txt" คือ ชื่อไฟล์ ส่วน "C:/AppServ/www/file/" คือ ไดเรกทอรีที่เก็บไฟล์ "test.txt" นอกจากนี้ยังอ้างถึงไฟล์ใน Internet/Intranet ได้ โดยระบุเป็น URL เช่น "http://www.freebsd.sru.ac.th" หรือเรียกไฟล์ผ่านโปรโตคอล FTP เช่น ftp://user:password@www.freebsd.sru.ac.th/file/test.txt เป็นต้น

ปัญหาที่อาจเกิดขึ้นจากการเปิดไฟล์หรือเขียนไฟล์ด้วย fopen () คือ ผู้ใช้อาจไม่มีสิทธิ์ในการอ่านหรือเขียนข้อมูลในไฟล์ที่เซิร์ฟเวอร์ ส่วนใหญ่จะพบในระบบปฏิบัติการ Unix หรือ Linux เนื่องจากมีการกำหนดสิทธิการใช้งานของไฟล์แต่ละไฟล์ไว้ ดังนั้นถ้าใช้ฟังก์ชัน fopen () และเกิด error ในลักษณะของ “Permission denied” ให้ตรวจสอบสิทธิ์ในการใช้งานไฟล์อีกครั้งว่าสามารถใช้งานได้หรือไม่

ในกรณีที่ไม่สามารถใช้ฟังก์ชัน fopen () เพื่อเปิดไฟล์ข้อมูลได้ สามารถเขียนคำสั่งเพื่อแจ้งข้อผิดพลาด โดยใช้โครงสร้างเงื่อนไข if ในการตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นได้ ดังนี้

ตัวอย่างที่ 8.4 การใช้โครงสร้างเงื่อนไข if ในการตรวจสอบข้อผิดพลาดเมื่อเปิดไฟล์ข้อมูลไม่ได้

```

1 <?php
2 $fp = @open (" /usr/local/apache/data/doc/customer.txt ". " w ")
3 If (!$fp) {
4     echo " ไม่สามารถเปิดไฟล์ได้ ";
5     exit;
6 }
7 ?>

```

จากตัวอย่างที่ 8.4 การใช้โครงสร้างเงื่อนไข if ในการตรวจสอบข้อผิดพลาดเมื่อเปิดไฟล์ข้อมูลไม่ได้ มีการระบุ @ ไว้หน้าฟังก์ชัน fopen() เพื่อไม่ให้ฟังก์ชันแสดงข้อผิดพลาดของระบบขึ้นมาในกรณีที่ไม่สามารถเปิดไฟล์ได้ แต่จะใช้โครงสร้างเงื่อนไข if ในการตรวจสอบผลลัพธ์ที่ได้จากการเปิดไฟล์ด้วยผู้พัฒนาเอง เป็นต้น

8.2.2 การปิดไฟล์

ทุกรั้งที่มีการเปิดไฟล์ เมื่อใช้ทำงานตามวัตถุประสงค์เสร็จ หลังจากนั้นควรจะปิดไฟล์ทุกครั้ง โดยใช้ฟังก์ชัน fclose() เพื่อปิดการเขียนต่อและคืนค่าหน่วยความจำกลับสู่ระบบ โดยเมื่อเรียกใช้ฟังก์ชัน fclose() หากสามารถปิดไฟล์ได้ ฟังก์ชันจะคืนค่าเป็น true และหากปิดไฟล์ไม่สำเร็จ จะคืนค่าเป็น false รูปแบบการใช้งานฟังก์ชัน มีดังนี้

รูปแบบ

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

เมื่อ \$handle หมายถึง Resource หรือตำแหน่งของตัวชี้ที่อยู่ในไฟล์ข้อมูล เป็นค่าที่ส่งคืนมาจากฟังก์ชัน fopen()

ตัวอย่างที่ 8.5 การใช้ฟังก์ชัน fclose()

```
1 <?php
2     $handle = fopen ("somefile.txt", "r");
3     fclose ($handle);
4 ?>
```

8.2.3 การเขียนข้อมูลลงในไฟล์

การเขียนข้อมูลลงในไฟล์ สามารถประยุกต์ใช้ได้ 3 ฟังก์ชันหลักๆ ประกอบด้วย 1) fwrite() 2) fputs() และ 3) file_put_content() โดยก่อนจะเขียนข้อมูลลงในไฟล์จะต้องเปิดไฟล์ด้วยฟังก์ชัน fopen() ก่อนเสมอ รายละเอียดดังนี้

1) ฟังก์ชัน fwrite() เป็นฟังก์ชันสำหรับเขียนข้อมูลลงในไฟล์ มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

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

เมื่อ \$handle หมายถึง Resource หรือตำแหน่งของตัวชี้ที่อยู่ในไฟล์ข้อมูล
 \$string หมายถึง ข้อมูลหรือข้อความที่ต้องการจัดเก็บลงไฟล์
 \$length หมายถึง จำนวนตัวอักษรที่ต้องการบันทึกลงไฟล์ (มีหน่วยเป็นไบต์) หากไม่กำหนดจะบันทึกกลุ่มตัวอักษรทั้งหมดจากตัวแปร \$string ลงในไฟล์



ตัวอย่างที่ 8.6 การใช้ฟังก์ชัน fwrite()

```

1 <?php
2     $path = "C:/AppServ/www/file";
3     $filename = "testfwrite.txt";
4     $file = fopen ( $path . "/" . $filename, "w");
5     $data = "ทดสอบการเขียนข้อมูลลงไฟล์ด้วยฟังก์ชัน fwrite()";
6     if (fwrite ($file, $data)==True) {
7         echo "เขียนข้อมูลลงไฟล์เรียบร้อยแล้ว";
8     }
9     fclose ($file);
10    ?>

```

จากตัวอย่างที่ 8.6 การใช้ฟังก์ชัน fwrite() เพื่อเขียนข้อมูลลงในไฟล์ อธิบายดังนี้

บรรทัดที่ 2 กำหนดให้ตัวแปร \$path มีค่าเท่ากับ "C:/AppServ/www/file" โดยความหมายแล้วหมายถึงการกำหนดตำแหน่งของ Document Root เพื่อใช้อ้างอิงตำแหน่งสำหรับเก็บไฟล์ข้อมูล

บรรทัดที่ 3 กำหนดให้ตัวแปร \$filename มีค่าเท่ากับ "testfwrite.txt" (ชื่อของไฟล์ที่ต้องการเขียนข้อมูล)

บรรทัดที่ 4 กำหนดให้ตัวแปร \$file มีค่าเท่ากับ เรียกใช้ฟังก์ชัน fopen() โดยระบุหรืออ้างอิงตำแหน่งของ Document Root ระบุไฟล์ที่ต้องการใช้งานและmodeไฟล์ที่ต้องการใช้งานคือ w (กำหนดให้ตัวแปร \$file ทำหน้าที่เป็นตัวชี้ตำแหน่งข้อมูลในไฟล์)

บรรทัดที่ 5 กำหนดให้ตัวแปร \$data มีค่าเท่ากับข้อความ "ทดสอบการเขียนข้อมูลลงไฟล์ด้วยฟังก์ชัน fwrite()"

บรรทัดที่ 6 ใช้โครงสร้างเงื่อนไข if ตรวจสอบผลการทำงานของฟังก์ชัน fwrite() โดยระบุพารามิเตอร์ประกอบด้วย \$file (ทำหน้าที่ตัวชี้ตำแหน่งข้อมูลในไฟล์) และ \$data (คือ ข้อมูลที่จะใช้เขียนลงในไฟล์) กรณีผลการตรวจสอบมีค่าเป็น true ไปบรรทัดที่ 7

บรรทัดที่ 7 แสดงข้อความ "เขียนข้อมูลลงไฟล์เรียบร้อยแล้ว"

บรรทัดที่ 8 ปิดการเชื่อมต่อกับไฟล์

2) ฟังก์ชัน fputs() เป็นฟังก์ชันที่ใช้สำหรับเขียนข้อมูลลงไฟล์ โดยการเรียกใช้ และการทำงานจะเหมือนกับฟังก์ชัน fwrite() มีรูปแบบดังนี้

รูปแบบ

```
int fputs ( resource $handle , string $string [, int $length ] )
```

เมื่อ \$handle หมายถึง Resource หรือตำแหน่งของตัวชี้ที่อยู่ในไฟล์ข้อมูล

\$string หมายถึง ข้อมูลหรือข้อความที่ต้องการจัดเก็บลงไฟล์
 \$length หมายถึง จำนวนตัวอักษรที่ต้องการบันทึกลงไฟล์ (มีหน่วยเป็นไบต์) หากไม่
 กำหนดจะบันทึกกลุ่มตัวอักษรทั้งหมดจากตัวแปร \$string ลงในไฟล์

ตัวอย่างที่ 8.7 การใช้ฟังก์ชัน fputs ()

```

1 <?php
2     $filename = "sitevisitors.txt";
3     if (file_exists ($filename)) {
4         $count = file($filename);
5         $count [0]++;
6         $fp = fopen ($filename, "w");
7         fputs ($fp, "$count[0]");
8         fclose ($fp);
9         echo $count[0];
10    } else {
11        $fh = fopen ($filename, "w");
12        if ($fh==false) die ("ไม่สามารถสร้างไฟล์ได้");
13        fputs ($fh, 1);
14        fclose ($fh);
15        $count = file ($filename);
16        echo $count[0];
17    }
18 ?>

```

จากตัวอย่างที่ 8.7 การใช้ฟังก์ชัน fputs () เพื่อพัฒนาระบบตัวนับจำนวนผู้เข้าชมเว็บไซต์ อธิบายดังนี้

บรรทัดที่ 2 กำหนดให้ตัวแปร \$filename มีค่าเท่ากับ "sitevisitors.txt" (ชื่อของไฟล์)

บรรทัดที่ 3 ใช้โครงสร้างเงื่อนไข if ตรวจสอบ ผลการทำงานของฟังก์ชัน file_exit () ใช้สำหรับตรวจสอบว่ามีไฟล์ที่ระบุหรือไม่ ถ้ามีไฟล์ที่ระบุจะคืนค่า true หากไม่มีไฟล์ที่ระบุจะคืนค่า false ในกรณีคืนค่า กรณีคืนค่า true ไปบรรทัดที่ 4 หากคืนค่า false ไปบรรทัด 10

บรรทัดที่ 4 กำหนดให้ตัวแปร \$count อ่านไฟล์ข้อมูลจากไฟล์ "sitevisitors.txt" ด้วยฟังก์ชัน file () รูปแบบการอ่านจะอ่านข้อมูลทั้งหมดออกมายังรูปแบบอาร์เรย์



บรรทัดที่ 5 กำหนดให้ตัวแปร \$count [0] เพิ่มค่า 1 ค่า (การที่ต้องเรียกใช้ \$count [0] เป็นเพราะผลมาจากการฟังก์ชัน file () เมื่ออ่านข้อมูลทั้งหมดตัวแปรจะเปลี่ยนเป็นชนิดอาร์เรย์ ดังนั้น การอ้างอิงค่าต้องระบุในรูปแบบของอาร์เรย์)

บรรทัดที่ 6 กำหนดให้ตัวแปร \$fp ทำหน้าที่เป็นตัวชี้ตำแหน่งข้อมูลในไฟล์ เปิดไฟล์ "sitevisitors.txt" ในโหมด "w" ด้วยฟังก์ชัน fopen ()

บรรทัดที่ 7 เขียนข้อมูลลงในไฟล์ (\$fp คือ ตัวแทนของไฟล์ข้อมูล และ \$count [0] คือ ข้อมูลที่จะใช้เขียนลงในไฟล์)

บรรทัดที่ 8 ปิดการเขียนต่อไฟล์

บรรทัดที่ 9 แสดงค่าผลในตัวแปร \$count [0]

บรรทัดที่ 10 ปิดโครงสร้างเงื่อนไข if เริ่มต้นโครงสร้างเงื่อนไข else (กรณีเป็น false จากโครงสร้างเงื่อนไข if)

บรรทัดที่ 11 กำหนดให้ตัวแปร \$fh ทำหน้าที่เป็นตัวชี้ตำแหน่งข้อมูลในไฟล์ เปิดไฟล์ "sitevisitors.txt" ในโหมด "w" ด้วยฟังก์ชัน fopen ()

บรรทัดที่ 12 ตรวจสอบค่าตัวแปร \$fh ด้วยโครงสร้างเงื่อนไข if ว่าสามารถเปิดไฟล์ได้ หรือไม่ หากเป็น true ไปบรรทัดที่ 13 หากเป็น false ให้หยุดทำงานแล้วแจ้งข้อความ "ไม่สามารถสร้างไฟล์ได้"

บรรทัดที่ 13 เขียนค่าเลข 1 ลงใน \$fh (\$fh คือ ตัวแทนของไฟล์ข้อมูล)

บรรทัดที่ 14 ปิดการเขียนต่อไฟล์

บรรทัดที่ 15 กำหนดให้ตัวแปร \$count อ่านไฟล์ข้อมูลจากไฟล์ "sitevisitors.txt" ด้วยฟังก์ชัน file () รูปแบบการอ่านจะอ่านข้อมูลทั้งหมดออกมายังรูปแบบอาร์เรย์

บรรทัดที่ 16 แสดงค่าตัวแปร \$count [0] (จะแสดงหมายเลข 1 ในรอบถัดๆ ไปจะทำงานเฉพาะช่วงบรรทัดที่ 3 ถึงบรรทัดที่ 9 เท่านั้น)

3) ฟังก์ชัน file_put_content () เป็นฟังก์ชันสำหรับเขียนข้อมูลลงไฟล์ เช่นเดียวกับ ฟังก์ชัน fwrite () และ fputs () แต่ฟังก์ชันนี้จะดำเนินการเองตั้งแต่เปิดไฟล์ เขียนข้อมูลลงไฟล์ และปิดไฟล์ ต่างจากวิธีเดิมที่ต้องใช้ฟังก์ชัน fopen () เปิดไฟล์ก่อน แล้วจึงใช้ฟังก์ชัน fwrite () หรือฟังก์ชัน fputs () เขียนข้อมูลลงไฟล์ เมื่อการทำงานเสร็จสิ้นจึงใช้ฟังก์ชัน fclose () ปิดไฟล์ รูปแบบการใช้งาน ฟังก์ชัน file_put_content () มีดังนี้

รูปแบบ

```
int file_put_contents ( string $filename , mixed $data )
```

เมื่อ \$filename หมายถึง ชื่อไฟล์ที่จะเขียนข้อมูล

\$data หมายถึง ข้อมูลที่ต้องการบันทึกลงไฟล์

ตัวอย่างที่ 8.8 การใช้ฟังก์ชัน file_put_content ()

```

1 <?php
2     $file = 'people.txt';
3     $current = file_get_contents ($file);           // เปิดไฟล์รับข้อมูลเดิม
4     $current .= "John Smith \n";                  // เพิ่มข้อมูลบุคลากร
5     file_put_contents ($file, $current);          // เขียนข้อมูลลงไฟล์
6 ?>

```

8.2.4 การอ่านข้อมูลจากไฟล์

การอ่านข้อมูลจากไฟล์ มีขั้นตอนการทำงานเหมือนกับการเขียนไฟล์ คือ เปิดไฟล์ด้วยฟังก์ชัน fopen () ก่อนจะสามารถอ่านไฟล์ข้อมูลได้ เมื่อเลิกใช้ไฟล์แล้วจะต้องปิดไฟล์ด้วยฟังก์ชัน fclose () สำหรับฟังก์ชันที่ใช้อ่านไฟล์ข้อมูลมีดังนี้

1) ฟังก์ชันfeof () เป็นฟังก์ชันที่ใช้ตรวจสอบและอ่านข้อมูลในไฟล์ เมื่อข้อมูลทั้งหมดถูกอ่านฟังก์ชันfeof จะส่งคืนค่าเป็น true แต่ถ้ายังอ่านไม่หมด จะส่งคืนค่าเป็น false มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool feof ( resource $handle )
```

เมื่อ \$handle หมายถึง ตัวแปรสำหรับชี้ข้อมูลในไฟล์

ตัวอย่างที่ 8.9 การใช้ฟังก์ชันfeof ()

```

1 <?php
2     $file = fopen ("data.txt", "r");
3     while (!feof ($file)) {
4         $char = fgetc ($file);
5         echo $char . "<br>";
6     }
7     fclose($file);
8 ?>

```

จากตัวอย่างที่ 8.9 การใช้ฟังก์ชันfeof () เพื่ออ่านค่าข้อมูลในไฟล์ อธิบายบรรทัดที่สำคัญ ดังนี้

บรรทัดที่ 2 กำหนดให้ตัวแปร \$file เท่ากับผลการเรียบใช้ฟังก์ชัน fopen () เปิดไฟล์ "data.txt" ด้วยไฟล์โหมด "r"



บรรทัดที่ 3 ใช้โครงสร้างเงื่อนไขทำซ้ำ while เพื่ออ่านค่าข้อมูลตรวจสอบข้อมูลสุดท้ายด้วย feof (เพื่อข้อมูลไม่สามารถอ่านได้ฟังก์ชันจะคืนค่า false) โครงสร้างเงื่อนไขทำซ้ำ while จะวนรอบเพื่ออ่านข้อมูลไปเรื่อยๆ จนกว่าจะไม่มีข้อมูลให้อ่าน

2) ฟังก์ชัน fgetc () ใช้อ่านข้อมูลจากไฟล์ครั้งละ 1 อักษร รูปแบบการใช้งานฟังก์ชันมีดังนี้

รูปแบบ

```
string fgetc ( resource $handle )
```

เมื่อ \$handle หมายถึง ตัวแปรสำหรับชี้ข้อมูลในไฟล์

ตัวอย่างที่ 8.10 การใช้ฟังก์ชัน fgetc ()

```
1 <?php
2     $fp = fopen ("somefile.txt", "r");
3     if (!$fp) {
4         echo "ไม่สามารถเปิดไฟล์ somefile.txt ได้";
5     } else {
6         while (false !== ($char = fgetc ($fp))) {
7             echo "$char<br>";
8         }
9     }
10    ?>
```

จากตัวอย่างที่ 8.10 แสดงตัวอย่างการใช้ฟังก์ชัน fgetc () สำหรับอ่านข้อมูลจากไฟล์ครั้งละ 1 อักษร

3) ฟังก์ชัน fread () ใช้อ่านข้อมูลจากไฟล์ตามความยาวของอักขระที่ระบุ รูปแบบการใช้งานฟังก์ชัน มีดังนี้

รูปแบบ

```
string fread ( resource $handle , int $length )
```

เมื่อ \$handle หมายถึง ตัวแปรสำหรับชี้ตำแหน่งข้อมูลในไฟล์

\$length หมายถึง ความยาวของอักขระที่ต้องการอ่าน

ตัวอย่างที่ 8.11 การใช้ฟังก์ชัน fread ()

```
1 <?php
2     $filename = "/usr/local/something.txt";
3     $handle = fopen ($filename, "r");
4     $contents = fread ($handle, filesize ($filename));
```

```

5      fclose ($handle);
6      ?>

```

จากตัวอย่างที่ 8.11 แสดงตัวอย่างการใช้ฟังก์ชัน fread () ใช้อ่านข้อมูลจากไฟล์ตามความยาวของอักษรที่ระบุ

4) ฟังก์ชัน fgets () เป็นฟังก์ชันที่ใช้อ่านข้อมูลจากไฟล์ครั้งละ 1 บรรทัด และจะอ่านข้อมูลจนกระทั่งพบอักษรสำหรับขึ้นบรรทัดใหม่ (\n) หรือสิ้นสุดไฟล์ จึงหยุดอ่าน มีรูปแบบการใช้งานฟังก์ชันมีดังนี้

รูปแบบ

```
string fgets ( resource $handle [, int $length ] )
```

เมื่อ \$handle หมายถึง ตัวแปรสำหรับชี้ข้อมูลในไฟล์

\$length หมายถึง ความยาวของอักษรที่ต้องการอ่าน

ตัวอย่างที่ 8.12 การใช้ฟังก์ชัน fgets ()

```

1 <?php
2     $handle = @fopen ("/tmp/inputfile.txt", "r");
3     if ($handle) {
4         while (($buffer = fgets ($handle, 4096)) !== false) echo $buffer;
5         if (!feof ($handle)) {
6             echo "Error: unexpected fgets () fail\n";
7         }
8         fclose ($handle);
9     }
10    ?>

```

5) ฟังก์ชัน fgetss () เป็นฟังก์ชันที่ให้ผลลัพธ์เหมือนกับฟังก์ชัน fgets () แต่ต่างกันที่ฟังก์ชัน fgetss () สามารถกำหนดให้แสดงหรือไม่แสดง HTML tags หรือ PHP ได้ โดยฟังก์ชัน fgets () จะข้าม HTML tags หรือ PHP ไป แต่จะสนใจเฉพาะข้อความตัวอื่น ฟังก์ชัน fgetss () มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

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

เมื่อ \$handle หมายถึง ตัวแปรสำหรับชี้ข้อมูลในไฟล์

\$length หมายถึง ความยาวของอักษรที่ต้องการอ่าน

\$allowable_tags หมายถึง อนุญาตให้แท็กทำงานหรือไม่ทำงาน



ตัวอย่างที่ 8.13 การใช้ฟังก์ชัน fgetss ()

```

1 <?php
2     $str = <<<EOD
3     <html><body>
4     <p>Welcome! Today is the <?php echo (date ("jS")); ?> of <?= date ("F");
?>.</p>
5     </body></html>
6     Text outside of the HTML block.
7     EOD;
8     file_put_contents ("sample.php", $str);
9     $handle = @fopen ("sample.php", "r");
10    if ($handle) {
11        while (!feof ($handle)) {
12            $buffer = fgetss ($handle, 4096);
13            echo $buffer;
14        }
15        fclose ($handle);
16    }
17    ?>

```

6) ฟังก์ชัน `readfile()` เป็นฟังก์ชันที่ใช้อ่านข้อมูลในไฟล์ทั้งหมด สามารถเปิดไฟล์ได้เอง โดยไม่ต้องใช้ฟังก์ชัน `fopen()` ฟังก์ชันจะแสดงข้อมูลที่อ่านได้ทางหน้าจอ พร้อมคืนค่าเป็นขนาดของข้อมูล (หน่วยเป็นไบต์) หากเกิดข้อผิดพลาด จะคืนค่าเป็น `false` รูปแบบการใช้งานฟังก์ชัน มีดังนี้

รูปแบบ

```
int readfile ( string $filename [, bool $use_include_path = false] )
```

เมื่อ `$filename` หมายถึง ชื่อไฟล์ที่จะอ่านข้อมูล

`$use_include_path` หมายถึง กำหนดเป็น `True` หากต้องการให้ค้นหาไฟล์ใน `include_path`

ตัวอย่างที่ 8.14 การใช้ฟังก์ชัน readfile ()

```

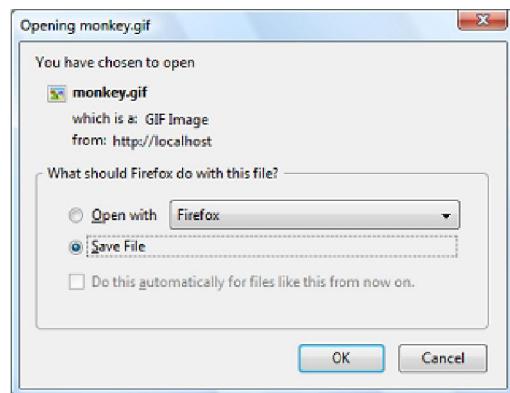
1 <?php
2     $file = "monkey.gif";
3     if (file_exists ($file)) {
4         header ("Content-Description: File Transfer");

```

```

5      header ("Content-Type: application/octet-stream");
6      header ("Content-Disposition: attachment; filename=".basename ($file));
7      header ("Content-Transfer-Encoding: binary");
8      header ("Expires: 0");
9      header ("Cache-Control: must-revalidate");
10     header ("Pragma: public");
11     header ("Content-Length: " . filesize ($file));
12     ob_clean (); flush ();
13     readfile ($file); exit;
14 }
15 ?>

```



ภาพที่ 8.1 แสดงการเปิดไฟล์รูปภาพ

7) ฟังก์ชัน fpassthru () เป็นฟังก์ชันที่ใช้อ่านข้อมูลในไฟล์เหมือนกับฟังก์ชัน readfile () แต่จะต้องเปิดไฟล์ด้วยฟังก์ชัน fopen () ก่อน จึงจะใช้ฟังก์ชัน fpassthru () อ่านข้อมูลได้ มีรูปแบบการใช้งานฟังก์ชัน ดังนี้

รูปแบบ

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

เมื่อ \$handle หมายถึง ตัวแปรสำหรับชี้ข้อมูลในไฟล์

ตัวอย่างที่ 8.15 การใช้ฟังก์ชัน fpassthru() สำหรับไฟล์ชนิดใบหน้า

```

1 <?php
2     $name = "./img/ok.png";
3     $fp = fopen ($name, "rb");           // เปิดไฟล์ด้วยโหมดใบหน้า
4     header ("Content-Type: image/png");
5     header ("Content-Length: " . filesize ($name));

```

```

6      fpassthru ($fp);
7      exit;
8  ?>
```

8) พังก์ชัน file () เป็นพังก์ชันสำหรับอ่านข้อมูลในไฟล์ทั้งหมดเหมือนกับพังก์ชัน readfile () แต่ต่างกันที่พังก์ชัน file () จะนำข้อมูลเก็บลงอาร์เรย์ มีรูปแบบการใช้งานพังก์ชันดังนี้
รูปแบบ

```
array file ( string $filename )
```

เมื่อ \$filename หมายถึง ชื่อไฟล์ที่ต้องการอ่านข้อมูล

ตัวอย่างที่ 8.16 การใช้ฟังก์ชัน file ()

```

1 <?php
2     $lines = file ("http://www.example.com/");
3     foreach ($lines as $line_num => $line) {
4         echo "Line #<b>{$line_num}</b> : " . htmlspecialchars($line) . "<br>";
5     }
6     $html = implode ("", file ("http://www.example.com/"));
7     $trimmed = file("somefile.txt", FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
8 ?>
```

9) พังก์ชัน file_get_content () เป็นพังก์ชันสำหรับอ่านข้อมูลในไฟล์ทั้งหมดเหมือนกับพังก์ชัน file () แต่ต่างกันที่พังก์ชัน file_get_content () จะคืนค่าเป็นข้อความ มีรูปแบบการใช้งานพังก์ชันดังนี้

รูปแบบ

```
string file_get_contents ( string $filename )
```

เมื่อ \$filename หมายถึง ชื่อไฟล์ที่ต้องการอ่านข้อมูล

ตัวอย่างที่ 8.17 การใช้ฟังก์ชัน file_get_content () หน้าโฮมเพจของเว็บไซต์

```

1 <?php
2     $homepage = file_get_contents ("http://www.example.com/");
3     echo $homepage;
4 ?>
```

ตัวอย่างที่ 8.18 การใช้ฟังก์ชัน file_get_content () ค้นหาในตำแหน่ง include_path

```

1 <?php
2     $file = file_get_contents ("./people.txt", FILE_USE_INCLUDE_PATH);
3 ?>
```

8.2.5 การลบไฟล์

สามารถใช้ฟังก์ชัน unlink() ได้ มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool unlink ( string $filename )
```

เมื่อ \$filename หมายถึง ชื่อไฟล์ที่จะลบ

ตัวอย่างที่ 8.19 การใช้ฟังก์ชัน unlink()

```
1 <?php
2     $fh = fopen ("test.html", "a");
3     fwrite ($fh, "<h1>Hello world!</h1>");
4     fclose ($fh);
5     unlink ("test.html");
6 ?>
```

8.2.6 ฟังก์ชันอื่นๆ ที่ใช้จัดการกับไฟล์

1) ฟังก์ชันสำหรับการตรวจสอบไฟล์

จากตัวอย่างก่อนหน้านี้ หากไฟล์ที่ต้องการเปิดไม่มีอยู่จริง หรือไฟล์นั้นมีอนุญาตให้มีการเขียน/อ่านไฟล์ จะทำให้ฟังก์ชันคืนค่าเป็นความผิดพลาด หากไม่ตักจับหรือตรวจสอบความผิดพลาดเหล่านั้น อาจทำให้มีผลกระทบต่อการทำงานของส่วนอื่นๆ สำหรับฟังก์ชันที่ใช้ตรวจสอบไฟล์ มีดังต่อไปนี้

ตารางที่ 8.3 ฟังก์ชันที่ใช้ตรวจสอบเกี่ยวกับไฟล์

ฟังก์ชัน	การทำงาน	รูปแบบ
is_file()	ตรวจสอบว่าเป็นไฟล์ทั่วไปหรือไม่ หากใช่ จะคืนค่าเป็น True หากไม่ใช่ จะคืนค่าเป็น False	bool is_file (string \$filename)
is_writable()	ตรวจสอบว่าสามารถเขียนข้อมูลลงในไฟล์ได้หรือไม่ หากได้จะคืนค่าเป็น True หากไม่ได้ จะคืนค่าเป็น False	bool is_writable (string \$filename)
is_readable()	ตรวจสอบว่าสามารถอ่านข้อมูลจากไฟล์นั้นได้หรือไม่ การคืนค่าจะเหมือนกับฟังก์ชัน is_writable()	bool is_readable (string \$filename)



2) การตรวจสอบเวลาที่เกี่ยวข้องกับไฟล์

ตารางที่ 8.4 ฟังก์ชันที่ใช้ตรวจสอบเวลาที่เกี่ยวข้องกับไฟล์

ฟังก์ชัน	การทำงาน	รูปแบบ
fileatime ()	ใช้เวลา (ค่า Unix Timestamp) ที่มีการเข้าถึงไฟล์ครั้งล่าสุด	int fileatime (string \$filename)
filectime ()	ใช้เวลา (ค่า Unix Timestamp) ที่มีการเปลี่ยนแปลงข้อมูลของไฟล์ครั้งล่าสุด ข้อมูลของไฟล์ที่จะบันทึกได้แก่ การกำหนดกลุ่มผู้ใช้งาน มี 3 กลุ่ม ประกอบด้วย user, group และ ownership การกำหนดสิทธิ์การเข้าถึงไฟล์ โดยเวลานี้จะถูกบันทึกขณะสร้างไฟล์ และทุกครั้งที่มีการเปลี่ยนแปลงข้อมูล	int filectime (string \$filename)
filemtime ()	ใช้เวลา (ค่า Unix Timestamp) ที่มีการแก้ไขข้อมูลในไฟล์ครั้งล่าสุด โดยเวลานี้จะถูกบันทึกตอนสร้างไฟล์ และทุกครั้งที่มีการแก้ไขข้อมูล	int filemtime (string \$filename)

8.3 การจัดการกับไดเรกทอรี

ภาษา PHP มีฟังก์ชันที่ใช้จัดการเกี่ยวกับไดเรกทอรี ดังนี้

8.3.1 การอ่านไดเรกทอรี

1) ฟังก์ชัน opendir () เป็นฟังก์ชันสำหรับเปิดไดเรกทอรีเพื่ออ่านรายชื่อไฟล์ภายใน โดยจะส่งคืนค่าเป็นตัวชี้ตำแหน่งไดเรกทอรี (Directory Handler) หากเปิดไดเรกทอรีไม่ได้ จะคืนค่าเป็น False มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
resource opendir ( string $path )
```

เมื่อ \$path หมายถึง ไดเรกทอรีที่ต้องการเปิด

2) ฟังก์ชัน closedir () เป็นฟังก์ชันสำหรับปิดไดเรกทอรี มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
void closedir ([ resource $dir_handle ])
```

เมื่อ \$dir_handle หมายถึง ตัวแปรที่ใช้เก็บค่าของตัวชี้ที่ใช้มาไดเรกทอรี

3) ฟังก์ชัน readdir () เป็นฟังก์ชันสำหรับอ่านไดเรกทอรี มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
string readdir ([ resource $dir_handle ] )
```

เมื่อ \$dir_handle หมายถึง ตัวแปรที่ใช้เก็บค่าของตัวชี้ที่ใช้สำหรับการอ่าน

ตัวอย่างที่ 8.20 การใช้ฟังก์ชัน opendir (), closedir () และ readdir ()

```

1 <?php
2     $dir = "/etc/php5/";
3     if (is_dir ($dir)) {
4         if ($dh = opendir ($dir)) {
5             while (($file = readdir ($dh)) != false) {
6                 echo "filename: $file : filetype: " . filetype ($dir . $file) . "<br>";
7             }
8             closedir ($dh);
9         }
10    }
11 ?>
```

4) ฟังก์ชัน rewindddir () เป็นฟังก์ชันสำหรับกลับไปที่ส่วนต้นของไดเรกทอรี มีรูปแบบ
การใช้งานฟังก์ชันดังนี้

รูปแบบ

```
void rewindddir ([ resource $dir_handle ] )
```

เมื่อ \$dir_handle หมายถึง ตัวแปรที่ใช้เก็บค่าของตัวชี้ที่ใช้สำหรับการอ่าน

8.3.2 การดูรายละเอียดของไดเรกทอรี

1) ฟังก์ชัน dirname () เป็นฟังก์ชันที่ใช้สำหรับแสดงพารามิเตอร์ของไดเรกทอรี มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
string dirname ( string $path )
```

เมื่อ \$path หมายถึง พารามิเตอร์ของไดเรกทอรี

2) ฟังก์ชัน basename () เป็นฟังก์ชันที่ใช้สำหรับแสดงชื่อของไฟล์ มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
string basename ( string $path [, string $suffix ] )
```

เมื่อ \$path หมายถึง พารามิเตอร์ของไดเรกทอรี



\$suffix หมายถึง ส่วนขยายไฟล์นั้น เช่น .php, .html, .txt หากไม่กำหนดค่าที่ให้จะมีทั้งชื่อไฟล์ และส่วนขยาย

ตัวอย่างที่ 8.21 การใช้ฟังก์ชัน basename ()

```

1 <?php
2 $p = "c:/windows/php.ini";
3 $base1 = basename ($p);           // ผลลัพธ์ คือ $base1 = " php.ini"
4 $base2 = basename ($p, ".ini");   // ผลลัพธ์ คือ $base2 = "php"
5 ?>

```

2) ฟังก์ชัน realpath () เป็นฟังก์ชันที่ใช้สำหรับแสดงพารามิเตอร์ทั้งหมด มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

string realpath (string \$path)

เมื่อ \$path หมายถึง พารามิเตอร์

ตัวอย่างที่ 8.22 การใช้ฟังก์ชัน realpath ()

```

<?php
    echo realpath('/windows/system32');           // ผลลัพธ์ คือ C:\WINDOWS\System32
?>

```

3) ฟังก์ชัน pathinfo () เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบพารามิเตอร์โดยการส่งคืนค่ากลับมาเป็นอาร์เรย์ที่สมาชิกแต่ละตัวประกอบไปด้วย ชื่อไดเรกทอรี, basename และส่วนขยายของไฟล์ อาร์เรย์ผลลัพธ์ที่จะได้อยู่ในรูปแบบคีย์และค่าอาร์เรย์ โดยค่าคีย์อาร์เรย์ จะประกอบไปด้วย dirname , basename และ extension มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```

mixed pathinfo ( string $path [, int $options = PATHINFO_DIRNAME | |
PATHINFO_BASENAME | PATHINFO_EXTENSION | PATHINFO_FILENAME ] )

```

เมื่อ \$path หมายถึง พารามิเตอร์

\$options หมายถึง กำหนดเพื่อดึงเฉพาะข้อมูลที่ต้องการ มี 4 ค่า ได้แก่

PATHINFO_DIRNAME เฉพาะข้อมูลไดเรกทอรี

PATHINFO_BASENAME เฉพาะชื่อไฟล์และนามสกุลไฟล์

PATHINFO_EXTENSION เฉพาะชื่อนามสกุลของไฟล์

PATHINFO_FILENAME เฉพาะชื่อไฟล์

หมายเหตุ

หากไม่ระบุพารามิเตอร์ฟังก์ชัน pathinfo () จะคืนค่าเป็นอาร์เรย์ประกอบด้วยอินเด็กซ์ dirname, basename, extention, และ filename ใช้สำหรับเก็บค่าไดเรกทอรี ชื่อไฟล์ (รวมนามสกุลไฟล์) และชื่อไฟล์ ตามลำดับ

ตัวอย่างที่ 8.23 การใช้ฟังก์ชัน pathinfo ()

```

1 <?php
2     $path_parts = pathinfo ("/www/htdocs/inc/lib.inc.php");
3     echo $path_parts ["dirname "]. "<br>";
4     echo $path_parts ["basename "] . "<br>";
5     echo $path_parts ["extension "] . "<br>"";;
6     echo $path_parts ["filename "] . "<br>";
7 ?>

```

ผลลัพธ์

```

/www/htdocs/inc
lib.inc.php
php
lib.inc

```

8.3.3 การสร้างและลบไดเรกทอรี

1) ฟังก์ชัน mkdir () เป็นฟังก์ชันที่ใช้สำหรับสร้างไดเรกทอรี หากสร้างได้ ฟังก์ชันจะส่งคืนค่าเป็น true หากสร้างไม่ได้ เพราะมีไดเรกทอรีนั้นอยู่แล้ว หรือไม่ได้รับสิทธิ์ให้สร้างไดเรกทอรี (ขึ้นอยู่กับระบบความปลอดภัยของแต่ละเครื่อง และการกำหนดค่าโหมดของไฟล์) ฟังก์ชันจะส่งคืนค่าเป็น false มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool mkdir ( string $pathname [, int $mode = 0777 ] )
```

เมื่อ	\$pathname	คือ	พาราเมตอร์ที่ใช้ในการระบุชื่อไดเรกทอรี
	\$mode	คือ	โหมดของไดเรกทอรีที่ใช้ใน Unix เท่านั้น ในระบบปฏิบัติการ MS-Windows พารามิเตอร์ตัวนี้จะไม่ถูกนำมาใช้ โดยปริยาย กำหนดให้มีค่าโหมดเป็น 0777 สามารถศึกษารายละเอียดเพิ่มเติมโหมดในฟังก์ชัน chmod ()



ตัวอย่างที่ 8.24 การใช้ฟังก์ชัน mkdir()

```

1 <?php
2     mkdir ("/path/to/my/dir", 0700); // สร้างไดเรกทอรีพร้อมกำหนดโหมดเป็น 0700
3 ?>

```

2) ฟังก์ชัน rmdir() เป็นฟังก์ชันที่ใช้สำหรับลบไดเรกทอรี มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool rmdir ( string $dirname )
```

เมื่อ \$dirname หมายถึง พาธของไดเรกทอรีที่ต้องการลบ

ตัวอย่างที่ 8.25 การใช้ฟังก์ชัน rmdir()

```

1 <?php
2     if (!is_dir ("examples ")) {
3         mkdir ("examples ");
4     }
5     rmdir ("examples ");
6 ?>

```

8.3.4 การตรวจสอบการมีอยู่ของไฟล์และไดเรกทอรี

ฟังก์ชัน file_exists() เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบว่ามีไฟล์ หรือไดเรกทอรีตามที่ระบุอยู่หรือไม่ หากมีจะคืนค่ากลับมาเป็นค่า true

รูปแบบ

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

เมื่อ \$filename หมายถึง ชื่อของไฟล์หรือไดเรกทอรีที่ต้องการตรวจสอบ

ตัวอย่างที่ 8.26 การใช้ฟังก์ชัน file_exists()

```

1 <?php
2     $filename = "/path/to/foo.txt";
3     if (file_exists ($filename)) {
4         echo "ตรวจพบไฟล์เอกสาร $filename";
5     } else {
6         echo "ไม่พบไฟล์เอกสาร $filename";
7     }
8 ?>

```

8.3.5 การคัดลอกไฟล์ เปลี่ยนแปลงชื่อไฟล์ และย้ายไดเรกทอรี

ฟังก์ชัน `copy()` เป็นฟังก์ชันที่ใช้สำหรับคัดลอกไฟล์ โดยกำหนดชื่อไฟล์ต้นฉบับ และชื่อไฟล์ปลายทางที่นำไฟล์คัดลอกไปไว้ หากกำหนดไดเรกทอรีร่วมกับชื่อไฟล์ จะหมายความว่า ให้คัดลอกไฟล์ไปยังไดเรกทอรีที่ระบุ หากไดเรกทอรีของปลายทางมีชื่อไฟล์ซ้ำกัน ฟังก์ชันจะนำไฟล์ที่คัดลอกหับไฟล์เดิมที่มีอยู่ มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool copy ( string $source , string $dest )
```

เมื่อ \$source หมายถึง ชื่อไฟล์ต้นฉบับ

\$dest หมายถึง ชื่อไฟล์หรือตำแหน่งปลายทางที่จะคัดลอกไฟล์

ตัวอย่างที่ 8.27 การใช้ฟังก์ชัน `copy()`

```
<?php
    $file = 'example.txt';
    $newfile = 'example.txt.bak';
    if (!copy ($file, $newfile)) {
        echo "ไม่สามารถคัดลอกไฟล์ $file... ได้ <br>";
    }
?>
```

8.3.6 การเปลี่ยนชื่อไฟล์และย้ายไดเรกทอรี

ฟังก์ชัน `rename()` เป็นฟังก์ชันที่ใช้สำหรับเปลี่ยนชื่อไฟล์ หากมีการกำหนดไดเรกทอรีร่วมกับชื่อไฟล์ใหม่ จะหมายถึงให้ย้ายไฟล์ไปยังไดเรกทอรีที่กำหนดด้วย และหากไดเรกทอรีนั้นมีชื่อไฟล์ซ้ำกัน ฟังก์ชันจะเขียนไฟล์ใหม่ทับไฟล์เดิมที่มีอยู่ มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool rename ( string $oldname , string $newname )
```

เมื่อ \$oldname หมายถึง ชื่อไฟล์เดิม

\$newname หมายถึง ชื่อไฟล์ใหม่

ตัวอย่างที่ 8.28 การใช้ฟังก์ชัน `rename()`

```
1 <?php
2     rename ("/tmp/tmp_file.txt", "/home/user/login/docs/my_file.txt");
3 ?>
```



8.4 การอัปโหลดไฟล์

ตัวแปร `$_FILES` เป็นตัวแปรประเภทตัวแปรพิเศษ ที่โปรแกรมภาษา PHP กำหนดไว้เพื่อเข้าถึงการทำงานเดี๋ยวก่อนของสคริปต์ โดยตัวแปร `$_FILES` จะเก็บข้อมูลที่เกี่ยวกับไฟล์อัปโหลด (จะต้องใช้รูปแบบการส่งฟอร์มแบบ HTTP POST method เท่านั้น) มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

`$_FILES [userfile] [key]`

เมื่อ `userfile` หมายถึง ชื่อที่กำหนดในแอทริบิวต์ `name` ที่ใช้รับชื่อไฟล์ที่ต้องการอัปโหลดจากเว็บฟอร์ม

`key` หมายถึง คีย์ในตัวแปร `$_FILES` ที่ใช้เก็บค่าต่างๆ ของไฟล์ โดยที่ค่าคีย์ของตัวแปร `$_FILES` มีดังนี้

ตารางที่ 8.5 ค่าคีย์ของตัวแปร `$_FILES`

คีย์	คำอธิบาย
<code>name</code>	ชื่อของไฟล์อัปโหลด
<code>type</code>	ชนิดของไฟล์อัปโหลด
<code>tmp_name</code>	ไดเรกทอรีผู้ใช้ร่วมสำหรับเก็บไฟล์อัปโหลดชั่วคราว ซึ่งพารก็คือค่าของ "upload_tmp_dir" ส่วนชื่อไฟล์ PHP จะสร้างให้โดยอัตโนมัติ
<code>error</code>	ข้อผิดพลาดที่เกิดจากการอัปโหลด จะส่งคืนค่าแสดงเป็นตัวเลข มี 4 ค่า ได้แก่ <ul style="list-style-type: none"> ■ 0 หมายถึง การอัปโหลดสมบูรณ์ ■ 1 หมายถึง ขนาดไฟล์ใหญ่กว่าที่กำหนดใน <code>upload_max_filesize</code> (กำหนดใน <code>php.ini</code>) ■ 2 หมายถึง ขนาดไฟล์ใหญ่กว่าที่กำหนดใน <code>MAX_FILE_SIZE</code> ซึ่งกำหนดในฟอร์ม ■ 3 หมายถึง มีเพียงเฉพาะบางส่วนของไฟล์เท่านั้นที่ถูกอัปโหลด ■ 4 หมายถึง ไม่มีไฟล์ที่อัปโหลด
<code>size</code>	ขนาดของไฟล์อัปโหลด (หน่วยเป็นไบต์)

8.4.1 ฟังก์ชันเกี่ยวกับการอัปโหลด

1) ฟังก์ชัน `is_upload_file ()` เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบว่ามีการส่งไฟล์ที่ต้องการอัปโหลดหรือไม่ และไฟล์ที่ส่งมาเป็นแบบ POST หรือไม่ การอัปโหลดไฟล์จากเว็บฟอร์มจะต้องส่งแบบ POST เท่านั้น เพราะสามารถส่งข้อมูลที่มีความยาวไม่จำกัด (การส่งแบบ GET จะจำกัดความยาวในการส่งได้ไม่เกิน 255 ตัวอักษร) มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

`bool is_uploaded_file (string $filename)`

เมื่อ `$filename` หมายถึง ชื่อไฟล์ที่ต้องการตรวจสอบ

ตัวอย่างที่ 8.29 การใช้ฟังก์ชัน is_upload_file()

```

1 <?php
2     if (is_uploaded_file($_FILES['userfile']['tmp_name'])) {
3         echo "ไฟล์ ". $_FILES['userfile']['name'] . " อัปโหลดสำเร็จแล้ว<br>";
4         echo "มีรายละเอียด ดังต่อไปนี้ <br>";
5         readfile($_FILES['userfile']['tmp_name']);
6     } else {
7         echo "การอัปโหลดไม่สำเร็จ กรุณาตรวจสอบไฟล์ใหม่ ::";
8         echo "ชื่อไฟล์ คือ ". $_FILES['userfile']['tmp_name'] . ":" ;
9     }
10    ?>

```

2) ฟังก์ชัน move_upload_file() เนื่องจากการอัปโหลดไฟล์ของ PHP จะเก็บในไดเรกทอรีชั่วคราวก่อน และเมื่อใช้ฟังก์ชัน move_upload_file() ย้ายไฟล์ที่ได้ไปเก็บในไดเรกทอรีปลายทางที่กำหนด หากย้ายไฟล์ได้สำเร็จ จะคืนค่าเป็น true หากไม่สำเร็จ จะคืนค่าเป็น false มีรูปแบบการใช้งานฟังก์ชันดังนี้

รูปแบบ

```
bool move_uploaded_file( string $filename , string $destination )
```

เมื่อ \$filename หมายถึง ชื่อไฟล์ที่ต้องการอัปโหลด

\$destination หมายถึง ตำแหน่งไดเรกทอรีปลายทางที่จะเก็บไฟล์อัปโหลด

ตัวอย่างที่ 8.30 ฟอร์มอัปโหลดไฟล์ (แบบครั้งละ 1 ไฟล์)

```

//ก่อนอื่นจะส่งไฟล์เข้าเครื่องเซิร์ฟเวอร์ต้องแก้ฟอร์มโดยเพิ่ม enctype="multipart/form-data"
//และส่งไฟล์โดยใช้อนุพัทธนิดไฟล์
<form action="uploading.php" method="post" enctype="multipart/form-data">
<br /><input type="file" name="picture" />
</form>

<?php // การใช้ฟังก์ชัน move_upload_file() เพื่ออัปโหลดไฟล์
if (isset($_FILES['picture'])) { // ตรวจสอบก่อนว่าอัปโหลดไฟล์เข้ามาจริงๆ
    move_uploaded_file($_FILES['picture']['name']
    , 'uploads/' . $_FILES['picture']['tmp_name']);
}
?>

```



ตัวอย่างที่ 8.31 ฟอร์มอัพโหลดไฟล์ (แบบหลายไฟล์)

```
//ก่อนอื่นจะส่งไฟล์เข้าเครื่องเซิร์ฟเวอร์ต้องแก้ฟอร์มโดยเพิ่ม enctype="multipart/form-data"
//และส่งไฟล์โดยใช้อินพุทชนิดไฟล์

<form action="uploading.php" method="post" enctype="multipart/form-data">
<br /><input type="file" name="pictures []" />
<br /><input type="submit" />
</form>

<?php // move_upload_file () เพื่ออัพโหลดไฟล์
if ( isset($_FILES ['pictures'])) {
    foreach ($_FILES ['pictures'] ['error'] as $index => $value) {
        if ($value == UPLOAD_ERR_OK) {
            move_uploaded_file($_FILES['pictures']['tmp_name'][$index]
,'uploads/'.$_FILES['pictures']['name'][$index]);
        }
    }
}
?>
```

8.4.2 พักรชันการหาขนาด และชนิดของไฟล์

1) พักรชัน filesize () เป็นพักรชันที่ใช้สำหรับหาขนาดของไฟล์ ค่าที่ได้จะมีหน่วยเป็น byte หากต้องการเปลี่ยนหน่วยเป็น KB ต้องหารด้วย 1,024 หากต้องการเปลี่ยนเป็น MB ต้องหารด้วย 1,048,576

รูปแบบ

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

เมื่อ \$filename หมายถึง ชื่อของไฟล์ที่ต้องการหาขนาดของไฟล์

ตัวอย่างที่ 8.32 การใช้พักรชัน filesize ()

```
1 <?php
2     $filename = 'somefile.txt';
```

```

3      echo $filename . ':' . filesize($filename) . ' bytes';
4      ?>

```

2) พังก์ชัน filetype () เป็นพังก์ชันที่ใช้สำหรับตรวจสอบว่าไฟล์หรือไดร์กทอรีที่ระบุ

- หากเป็นไฟล์ (File) จะคืนค่ากลับมาเป็นคำว่า file
- หากเป็นไดร์กทอรี (Directory) จะคืนค่าเป็นคำว่า dir
- นอกจากนี้แล้วอาจเป็นไปได้อีก เช่น char, block, link, unknown

รูปแบบ

```
string filetype ( string $filename )
```

เมื่อ \$filename หมายถึง ชื่อของไฟล์ที่ต้องการหาขนาดของไฟล์

ตัวอย่างที่ 8.33 การใช้พังก์ชัน filetype ()

```

1 <?php
2     echo filetype ("/etc/passwd");           // file
3     echo filetype ("/etc/");                  // dir
4 ?>

```

สรุป

ในบทนี้ได้กล่าวถึงการจัดการกับไฟล์และไดร์กทอรี การประยุกต์ใช้พังก์ชันสำหรับจัดการกับระบบไฟล์และไดร์กทอรี ตัวอย่างเช่น การอ้างอิงตำแหน่งไฟล์และไดร์กทอรี การเปิดและปิดไฟล์ การเขียนและอ่านไฟล์ การจัดการตัวชี้ตำแหน่งเพื่ออ่านและเขียนค่าลงในไฟล์ การลบไฟล์ การสร้างไดร์กทอรี การลบไดร์กทอรี การเปลี่ยนชื่อดเรอทอรี การย้ายตำแหน่ง และการอัพโหลดไฟล์ เป็นต้น กิจกรรมทั้งหมดที่กล่าวถึงจะต้องอยู่ภายใต้กรอบและสิทธิในการเข้าถึงไฟล์และไดร์กทอรี ตามแนวโน้มโดยที่ผู้ดูแลระบบเครือข่ายหรือผู้ดูแลเว็บเซิร์ฟเวอร์เป็นผู้กำหนด เพื่อความปลอดภัยของไฟล์ข้อมูลในระบบ ดังนั้นบางพังก์ชันอาจจะใช้งานไม่ได้บนเว็บเซิร์ฟเวอร์หากไม่มีสิทธิในการเข้าถึง

คำถามท้ายบท

1. จงอธิบายความแตกต่างระหว่างการอ้างอิงตำแหน่งของไฟล์และไดร์กทอรีแบบ Absolute Path และการอ้างอิงแบบ Relative Path
2. จงอธิบายรายละเอียดของพังก์ชัน fopen () โดยมุณฑ์ที่เกี่ยวข้อง และยกตัวอย่างการเขียนสคริปต์เพื่อเปิดไฟล์สำหรับการเขียนและอ่าน โดยเริ่มทำงานจากจุดสิ้นสุดของข้อมูลที่มีอยู่ ถ้าไม่มีไฟล์จะทำการสร้างไฟล์ขึ้นมาใหม่
3. จงอธิบายความหมายและหน้าที่ของตัวชี้ตำแหน่งข้อมูลในไฟล์



4. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกรายละเอียดที่ได้จากสคริปต์

```
1     <?php
2         if (file_exists "file.txt")) {
3             $reading = file ("file.txt");
4             $reading [0]++;
5             $writing = fopen ("file.txt", "w");
6             fputs ($file_pointer, $reading [0]);
7             fclose ($writing);
8             echo $reading [0];
9         } else {
10             $writing = fopen ("file.txt", "w");
11             if ($writing==false) die ("ไม่สามารถสร้างไฟล์ได้");
12             fputs ($writing, 1);
13             fclose ($writing);
14             $reading = file ("file.txt");
15             echo $reading [0];
16         }
17     ?>
```

5. จงบอกชื่อและอธิบายหน้าที่ของฟังก์ชันในภาษา PHP ที่ใช้สำหรับบริหารจัดการไดเรกทอรี

บทที่ 9

การจัดการข้อมูลจากฟอร์ม

กระบวนการที่สำคัญของการสร้างเว็บแอพพลิเคชัน คือ การนำข้อมูลที่ถูกส่งจากเว็บбраузர์ ไปประมวลผลในฝั่งของเริบเซิร์ฟเวอร์ จึงทำให้ต้องมีส่วนในการกำหนดข้อมูลที่จะส่งไปประมวลผล ดังนั้น ในบทนี้จะศึกษาเกี่ยวกับการสร้างส่วนสำหรับรับข้อมูลจากผู้ใช้งาน เรียกว่า "ฟอร์ม (Form)" โดยภายใน ฟอร์มนี้จะประกอบด้วยการรับข้อมูลได้หลายชนิด การรับข้อมูลในรูปแบบที่แตกต่างกัน และขั้นตอน ทางด้านเซิร์ฟเวอร์ก็จะกำหนดรูปแบบวิธีการรับข้อมูลจากฟอร์ม จะแตกต่างตามชนิดของข้อมูลที่รับเข้า ไปประมวลผล ในบทนี้จะอธิบายการสร้างฟอร์มและกำหนดส่วนที่ใช้รับข้อมูลต่างๆ รวมถึงวิธีการส่งข้อมูล เพื่อนำข้อมูลที่รับเข้ามาไปประมวลผลหรือจัดเก็บข้อมูลลงในไฟล์หรือฐานข้อมูลต่อไป ซึ่งถือเป็นสิ่งสำคัญ อย่างยิ่งสำหรับการสร้างเว็บแอพพลิเคชัน

9.1 กำหนดขอบเขตของฟอร์ม

ก่อนจะเริ่มสร้างรับข้อมูลต่างๆ จะต้องกำหนดขอบเขตให้กับตัวฟอร์ม โดยการกำหนดขอบเขต ฟอร์ม ด้วยการแทรกแท็ก `<form> ... </form>` เข้าไปในเว็บเพจ การใช้งานแท็ก `<form>` มีรูปแบบการ ใช้งานและคุณสมบัติที่จำเป็น ดังนี้

รูปแบบ

```
<form name = "ชื่อฟอร์ม" method = "วิธีส่งข้อมูล" action = "ชื่อไฟล์สคริปต์ที่จะประมวลผล">  
...  
</form>
```

โดยวิธีการส่งข้อมูลมีอยู่ด้วยกัน 2 รูปแบบ ประกอบด้วย POST และ GET ทั้ง 2 รูปแบบมี ลักษณะการใช้งานที่แตกต่างกันทั้งนี้การเลือกวิธีการส่งข้อมูลจะขึ้นอยู่กับความเหมาะสมของงาน ดังนี้

ตารางที่ 9.1 แสดงลักษณะการใช้งานวิธีการส่งข้อมูลแบบ POST และ GET

วิธีส่งข้อมูล	ลักษณะการใช้งาน
POST	เป็นการส่งข้อมูลจากฟอร์มไปยังสคริปต์โดยตรง ซึ่งข้อมูลที่ส่งผ่านไม่สามารถมองเห็น ได้ด้วยวิธีปกติ วิธีนี้จึงเหมาะสมสำหรับข้อมูลที่ต้องการปิดเป็นความลับ
GET	เป็นการส่งข้อมูลจากฟอร์มไปยังสคริปต์โดยผ่าน URL ของเว็บเพจ ซึ่งวิธีนี้สามารถ มองเห็นข้อมูลที่ส่งผ่านในช่อง Address Bar ของเว็บбраузอร์ ดังนั้นจึงไม่เหมาะสมกับ ข้อมูลที่เป็นความลับ

9.2 การใช้งานส่วนนำเข้าข้อมูล

การสร้างฟอร์มขึ้นมาหนึ่งฟอร์ม ไม่ว่าจะเป็นฟอร์มสมัครสมาชิก ฟอร์มสั่งซื้อสินค้า หรือฟอร์มสำหรับล็อกอินเข้าสู่ระบบสมาชิก จะประกอบไปด้วยส่วนนำเข้าข้อมูลหลากหลายรูปแบบ เพื่อให้ง่ายแก่การทำความเข้าใจ ดังนั้นจะแบ่งส่วนนำเข้าข้อมูล ออกเป็น 4 กลุ่ม ประกอบด้วย 1) กลุ่มซ่องรับข้อความ 2) กลุ่มตัวเลือก 3) กลุ่มปุ่มส่งข้อมูล และ 4) กลุ่มอื่นๆ ทั้งนี้การประยุกต์ใช้งานในแต่ละกลุ่มนั้นอยู่กับความเหมาะสมของข้อมูลในฟอร์มและความต้องการ โดยแต่ละกลุ่มมีรายละเอียด ดังนี้

9.2.1 กลุ่มซ่องรับข้อความ

กลุ่มซ่องรับข้อความจะเป็นกลุ่มที่นิยมใช้งานมากที่สุด สามารถกำหนดซ่องรับข้อความได้ 3 รูปแบบ คือ 1) ซ่องรับข้อความบรรทัดเดียว (Single Line) 2) ซ่องรับข้อความหลายบรรทัด (Multi Line) และ 3) ข้อความรหัสผ่าน (Password) รายละเอียดดังนี้

1) ซ่องรับข้อความบรรทัดเดียว เหมาะสำหรับการป้อนข้อมูลสั้นๆ เช่น ชื่อ-สกุล อีเมล เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติให้กับซ่องรับข้อความแบบบรรทัดเดียว โดยกำหนดผ่านแท็ก `<input>` มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
<input type = "text" name = "ชื่อส่วนนำเข้าข้อมูล" size = "ความกว้างของซ่องรับข้อความ"
maxlength = "จำนวนตัวอักษรสูงสุดที่สามารถพิมพ์ลงในช่องได้">
```

สำหรับคุณสมบัติ `size` และ `maxlength` จะกำหนดหรือไม่ก็ได้

การตั้งชื่อซ่องรับข้อความควรสื่อความหมายของข้อมูลในช่องนั้นๆ ด้วยชื่อสั้นๆ ใจจดใจจำ ง่าย และที่สำคัญที่สุดจะต้องตั้งชื่อให้กับส่วนนำเข้าข้อมูลทุกตัว และชื่อที่ตั้งจะต้องไม่ซ้ำกัน เนื่องจากชื่อของส่วนนำเข้าข้อมูลเหล่านี้จะใช้ในการแทนค่าซ่องรับข้อมูลในฟอร์มเพื่อนำไปอ้างอิง เรียกใช้ และประมวลผลต่อไป

2) ซ่องรับข้อความหลายบรรทัด เหมาะสำหรับข้อมูลที่มีเนื้อหายาวๆ หลายบรรทัด เช่น ข้อมูลความคิดเห็น ข้อมูลที่อธิบาย เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติให้กับซ่องรับข้อความแบบหลายบรรทัด โดยกำหนดผ่านแท็ก `<textarea> ... </textarea>` มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
<textarea name = "ชื่อส่วนนำเข้าข้อมูล" cols = "จำนวนตัวอักษรที่สามารถพิมพ์ได้ใน 1
บรรทัด" rows = "จำนวนบรรทัดที่แสดงผล">
```

3) ซ่องรับข้อความรหัสผ่าน ส่วนนำเข้าข้อมูลชนิดนี้จะมีความพิเศษกว่าซ่องรับข้อความที่ผ่านมา ซึ่งเมื่อผู้ใช้ป้อนข้อมูลลงในซ่องรับข้อความรหัสผ่าน ข้อมูลจะถูก加密เป็นเครื่องหมายวงกลมดำเนินตัวอักษร เพื่อป้องกันไม่ให้ผู้อื่นเห็นข้อมูลรหัสผ่าน สามารถเขียนสคริปต์และคุณสมบัติให้กับซ่องรับข้อความรหัสผ่าน โดยกำหนดผ่านแท็ก `<input>` มีรูปแบบการใช้งานดังนี้

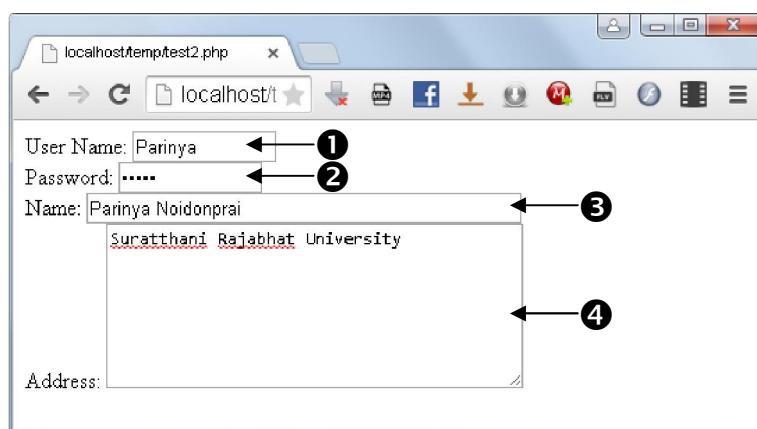
รูปแบบ

```
<input type = "password" name = "ชื่อส่วนนำเข้าข้อมูล" size = "ความกว้างของช่องรับ  
ข้อมูล" maxlength = "จำนวนตัวอักษรสูงสุดที่สามารถพิมพ์ลงในช่องได้">
```

ตัวอย่างที่ 9.1 สรุปตัวอย่างการสร้างฟอร์มรับข้อมูลแบบช่องรับข้อมูล

```
<form name= "frmMember" method = "POST" action = "member.php">  
  
User Name: <input name = "txtUser" type = "text" size = "10"><br>  
Password: <input name = "txtPass" type = "password" size = "10"><br>  
Name: <input name = "txtName" type = "text" size = "40"><br>  
Address: <textarea name = "txtAddress" cols = "40" row = "5"></textarea>  
</form>
```

ผลลัพธ์ แสดงดังภาพที่ 9.1



ภาพที่ 9.1 แสดงตัวอย่างการสร้างฟอร์มเพื่อนำเข้าข้อมูล

จากภาพที่ 9.1 แสดงตัวอย่างการสร้างฟอร์มรับข้อมูล ประกอบด้วยช่องรับข้อมูล ดังนี้

- ① ช่องรับข้อมูลบรรทัดเดียว ชื่อส่วนนำเข้าข้อมูล คือ txtUser
- ② ช่องรับข้อมูลรหัสผ่าน ชื่อส่วนนำเข้าข้อมูล คือ txtPass
- ③ ช่องรับข้อมูลบรรทัดเดียว ชื่อส่วนนำเข้าข้อมูล คือ txtName
- ④ ช่องรับข้อมูลหลายบรรทัด ชื่อส่วนนำเข้าข้อมูล คือ txtAddress

9.2.2 กลุ่มตัวเลือก

กลุ่มตัวเลือก คือ กลุ่มที่ใช้กำหนดตัวเลือกให้กับผู้ใช้งานระบบสามารถคลิกเพื่อเลือก คำตอบที่ต้องการ สามารถสรุปได้ 2 ประเภท คือ 1) ตัวเลือกที่เลือกตอบได้เพียงชื่อเดียว และ 2) ตัวเลือกที่เลือกตอบได้หลายชื่อ ซึ่งแต่ละประเภทจะประกอบไปด้วยส่วนนำเข้าข้อมูลต่างๆ ดังนี้

- 1) ตัวเลือกที่เลือกตอบได้เพียงชื่อเดียว มี 2 ชนิดอยู่ คือ 1) ปุ่มตัวเลือกแบบปุ่มกลม (Radio Button) และ 2) รายการตัวเลือกแบบ Menu รายละเอียดดังนี้

1.1) ปุ่มตัวเลือกแบบปุ่มกลม ส่วนนำเข้าข้อมูลลักษณะปุ่มกลม ใช้สำหรับเลือกตัวเลือกได้เพียง 1 ตัวเลือกเท่านั้น ลักษณะการประยุกต์เลือกตอบ เช่น เพศ ศาสนา หรือสถานภาพสมรส เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติให้กับปุ่มตัวเลือกแบบปุ่มกลม โดยกำหนดผ่านแท็ก `<input>` มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
<input type = "radio" name = "ชื่อส่วนนำเข้าข้อมูล" value = "ค่าของตัวเลือก" checked = "checked">ข้อความแสดงเพื่อเลือกตอบ</input>
```

สำหรับคุณสมบัติ checked จะกำหนดหรือไม่ก็ได้ โดยจะใช้ในกรณีที่ต้องการกำหนดให้ตัวเลือกนี้ถูกเลือกไว้เป็นค่าเริ่มต้น ตัวอย่างดังนี้

ตัวอย่างที่ 9.2 สคริปต์การสร้างฟอร์มรับข้อมูลตัวเลือกแบบปุ่มกลม

```
<form name= "frmGender" method = "POST" action = "gender.php">
```

Gender:

```
<input type ="radio" name = "rdoGender" value = "M" checked = "checked">Male  
<input type ="radio" name = "rdoGender" value = "F">Female
```

```
</form>
```

ผลลัพธ์



ภาพที่ 9.2 แสดงตัวอย่างการใช้งานปุ่มตัวเลือกแบบปุ่มกลม

จากภาพที่ 9.2 แสดงตัวอย่างการใช้งานปุ่มตัวเลือกแบบปุ่มกลม สำหรับเลือกเพศ โดยกำหนดให้ชื่อกลุ่มตัวเลือกมีชื่อว่า rdoGender และกำหนดให้เลือกข้อมูล Male (เพศชาย) เป็นค่าเริ่มต้น

1.2) รายการตัวเลือกแบบ Menu จะมีลักษณะเป็นกล่องรายการแสดงขึ้นมาให้เห็นเพียงตัวเลือกเดียว เพื่อให้คลิกเลือกรายการ สามารถเขียนสคริปต์และคุณสมบัติให้กับรายการตัวเลือกแบบ Menu โดยกำหนดผ่านแท็ก `<select>` มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
<select name = "ชื่อส่วนนำเข้าข้อมูล">  
    <option value = "ค่าของตัวเลือกที่ 1" selected = "selected">ตัวเลือกที่ 1 </option>  
    <option value = "ค่าของตัวเลือกที่ 2">ตัวเลือกที่ 2 </option>
```

```
<option value = "ค่าของตัวเลือกที่ 2">ตัวเลือกที่ N </option>
</select>
```

ตัวอย่างที่ 9.3 สรุปตัวอย่างการสร้างฟอร์มรับข้อมูลรายการตัวเลือกแบบ Menu

```
<form name= "frmProvince" method = "POST" action = "province.php">
    Province:
    <select name = "sltProvince">
        <option value = "1" selected = "selected">Suratthani</option>
        <option value = "2">Chumphon</option>
        <option value = "3">Ranong</option>
    </select>
</form>
```

ผลลัพธ์



ภาพที่ 9.3 แสดงตัวอย่างการใช้งานรายการตัวเลือกแบบ Menu

จากการที่ 9.3 แสดงตัวอย่างการใช้งานรายการตัวเลือกแบบ Menu สำหรับเลือกจังหวัด โดยกำหนดให้มีรายการตัวเลือกแบบ Menu มีชื่อว่า sltProvince และกำหนดให้เลือกข้อมูล Suratthani (จังหวัดสุราษฎร์ธานี) เป็นค่าเริ่มต้น

2) ตัวเลือกที่เลือกตอบได้หลายข้อ มี 1 ชนิด คือ ปุ่มตัวเลือกแบบปุ่มเหลี่ยม (Checkbox) โดยส่วนนำเข้าข้อมูล จะต้องมีชื่อกลุ่มตัวเลือกเดียวกันจึงสามารถเลือกตัวเลือกได้หลายตัวเลือก ข้อมูลที่ใช้ส่วนใหญ่จะเป็นข้อมูลเกี่ยวกับความชอบ สามารถเขียนสคริปต์และคุณสมบัติให้กับปุ่มตัวเลือกแบบปุ่มเหลี่ยม โดยกำหนดผ่านแท็ก <input> มีรูปแบบดังนี้

รูปแบบ

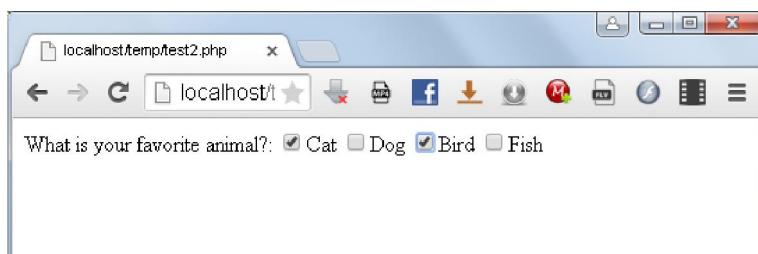
```
<input type = "checkbox" name = "ชื่อส่วนนำเข้าข้อมูล" value = "ค่าของตัวเลือก"
checked = "checked">ข้อความให้เลือก</input>
```



ตัวอย่างที่ 9.4 สรุปการสร้างฟอร์มรับข้อมูลตัวเลือกแบบปุ่มเหลี่ยม (เลือกตอบได้มากกว่า 1)

```
<form name= "frmAnimal" method = "POST" action = "animal.php">
    What is your favorite animal?:
    <input type = "checkbox" name = "chkAni" value = "cat" checked = "checked">Cat
    <input type = "checkbox" name = "chkAni" value = "dog">Dog
    <input type = "checkbox" name = "chkAni" value = "bird">Bird
    <input type = "checkbox" name = "chkAni" value = "fish">Fish
</form>
```

ผลลัพธ์



ภาพที่ 9.4 แสดงตัวอย่างการใช้งานตัวเลือกแบบปุ่มเหลี่ยม

จากการสั่นร่องที่ชื่นชอบหรือประทับใจ สามารถเลือกตอบได้มากกว่า 1 รายการ โดยกำหนดให้กลุ่มตัวเลือกแบบปุ่มเหลี่ยมมีชื่อส่วนนำเข้าข้อมูล คือ chkAni และกำหนดให้เลือกข้อมูล Cat (แมว) เป็นค่าเริ่มต้น

9.2.3 กลุ่มปุ่มสั่งงาน

สำหรับส่วนนำเข้าข้อมูลในกลุ่มนี้มีความสำคัญมาก และพบเห็นมากที่สุด เนื่องจากเมื่อสร้างฟอร์มพร้อมส่วนนำเข้าข้อมูลที่เกี่ยวข้องทั้งหมดแล้ว หากขาดกลุ่มปุ่มสั่งงานจะทำให้ไม่สามารถส่งข้อมูลที่ผู้ใช้ป้อนไปยังสคริปต์ที่กำหนดไว้ในฟอร์มได้ กลุ่มปุ่มสั่งงานแบ่งออกเป็น 3 รูปแบบ ประกอบด้วย 1) ปุ่มส่งข้อมูล (Submit Button) 2) ปุ่มล้างข้อมูล (Reset Button) และ 3) ปุ่มส่งข้อมูลแบบรูปภาพ (Image Button) มีรายละเอียดดังนี้

1) ปุ่มส่งข้อมูล จะทำหน้าที่ส่งข้อมูลที่ผู้ใช้ป้อนไว้ในฟอร์มไปประมวลผลยังสคริปต์ที่ระบุไว้ในฟอร์ม สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก `<input>` มีรูปแบบดังนี้

รูปแบบ

```
<input type = "submit" name = "ชื่อปุ่ม" value = "ข้อความที่แสดงบนปุ่ม">
```

2) ปุ่มล้างข้อมูล จะทำหน้าที่ลบค่าข้อมูลที่ผู้ใช้ป้อนไว้ในฟอร์มขณะนั้นออกໄไปโดยไม่มีการส่งข้อมูลไปประมวลผล สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก `<input>` มีรูปแบบดังนี้

```
<input type = "reset" name = "ซื้อปุ่ม" value = "ข้อความที่แสดงบนปุ่ม">
```

ตัวอย่างที่ 9.5 แสดงตัวอย่างการใช้ปุ่มส่งข้อมูล และล้างข้อมูล

```
<form name= "frmButton" method = "POST" action = "test-button.php">
    <input type ="submit" name = "btnSubmit" value = "Send"><br>
    <input type ="reset" name = "btnReset" value = "Clear">
</form>
```

ผลลัพธ์



ภาพที่ 9.5 แสดงตัวอย่างการใช้งานปุ่มส่งค่าข้อมูล และล้างข้อมูล

3) ปุ่มส่งข้อมูลแบบรูปภาพ คือ การนำรูปภาพมาใช้แทนข้อความที่ปรากฏบนปุ่ม ทำให้เกิดความสวยงามหรือสร้างความโดดเด่นให้กับปุ่ม สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก `<input>` มีรูปแบบดังนี้

รูปแบบ

```
<input type = "image" name = "ชื่อส่วนนำเข้าข้อมูล" src = "ไดเรกทอรี/ชื่อไฟล์รูปภาพ">
```

ตัวอย่างที่ 9.6 แสดงตัวอย่างการใช้ปุ่มส่งข้อมูลแบบรูปภาพ

```
<form name= "frmPhoto" method = "POST" action = "upload.php">
    <input type = "image" name = "btnUpload" src = "parinya.jpg">
</form>
```

ผลลัพธ์



ภาพที่ 9.6 แสดงตัวอย่างการใช้งานปุ่มส่งค่าข้อมูลแบบรูปภาพ

9.2.4 กลุ่มอื่นๆ นอกจากส่วนนำเข้าข้อมูลทั้ง 3 กลุ่มข้างต้นที่ได้กล่าวไว้แล้วนั้น ในภาษา PHP ยังมีส่วนนำเข้าข้อมูลที่นิยมใช้งานบ่อยๆ อีก ได้แก่ 1) ส่วนนำเข้าข้อมูลแบบซ่อนข้อมูล (Hidden Field) และ 2) ส่วนนำเข้าข้อมูลจากไฟล์ (File Field) โดยมีรายละเอียดดังนี้

1) ส่วนนำเข้าข้อมูลแบบซ่อนข้อมูล นิยมใช้ในการซ่อนข้อมูลสำหรับส่งข้อมูลในฟอร์มไปยังไฟล์สคริปต์ที่กำหนด เพื่อนำไปประมวลผลโดยไม่ต้องการให้ผู้ใช้เห็นข้อมูล ส่วนนำเข้าข้อมูลแบบซ่อนข้อมูลนิยมใช้กันมากในระบบการทำเว็บอร์ดโดยซ่อนหมายเลขค่าตามไว้ในฟอร์มตอบคำถาม เป็นต้น สามารถเขียนสคริปต์และคุณสมบัติ โดยกำหนดผ่านแท็ก <input> มีรูปแบบ ดังนี้

รูปแบบ

```
<input type = "hidden" name = "ชื่อส่วนนำเข้าข้อมูล" value = "ค่าข้อมูล">
```

ตัวอย่างที่ 9.7 แสดงตัวอย่างการเขียนสคริปต์ส่วนนำเข้าข้อมูลแบบซ่อนข้อมูล

```
<form name= "frmHiddenField" method = "POST" action = "test-hidden.php">  
    <input type ="hidden" name = "id" value = "001">  
</form>
```

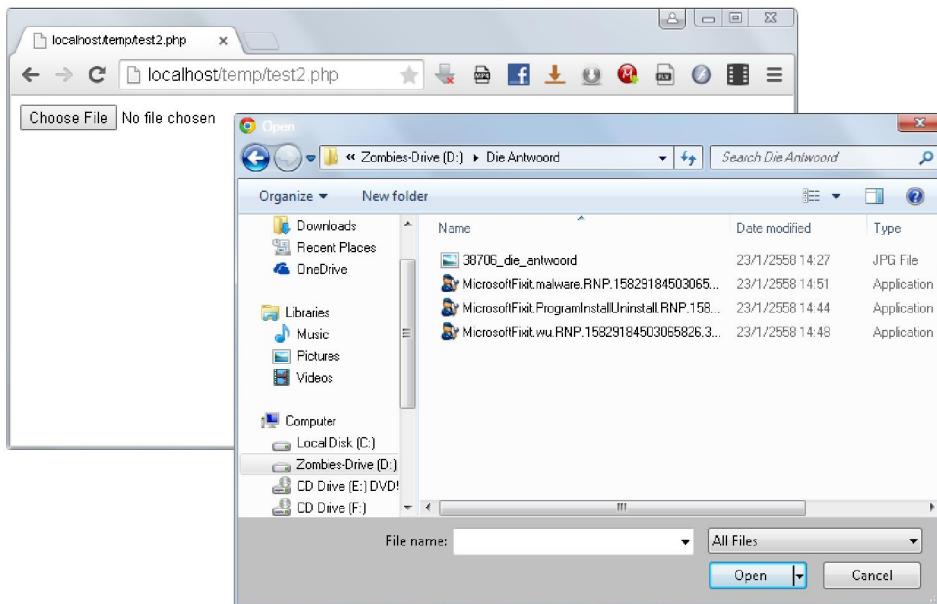
2) ส่วนนำเข้าข้อมูลจากไฟล์ ใช้สำหรับเปิดไฟล์ที่ต้องการได้ทุกนามสกุล เช่น .txt, .jpg, .gif เป็นต้น วิธีการที่สำคัญ คือ เพิ่มสคริปต์ในส่วนของ <form> คือ enctype = "multipart/form-data" เพื่อให้สามารถส่งข้อมูลรูปภาพจากส่วนนำเข้าข้อมูล มีรูปแบบดังนี้

รูปแบบ

```
<input type = "file" name = "ชื่อส่วนนำเข้าข้อมูล" size = "ความกว้างของช่องรับ">
```

ตัวอย่างที่ 9.8 แสดงตัวอย่างการเขียนสคริปต์เพื่อให้สามารถรับข้อมูลจากไฟล์

```
<form name= "frmMember" method = "POST" action = "member.php" enctype =  
"multipart/form-data">  
    <input type ="file" name = "photo">  
</form>
```



ภาพที่ 9.7 แสดงตัวอย่างการใช้งานส่วนนำเข้าข้อมูลจากไฟล์

9.3 การทดสอบการแสดงผลฟอร์ม

กรณีตัวอย่างที่จะอธิบายในลำดับถัดไป จะประกอบไปด้วยไฟล์สคริปต์ PHP จำนวน 2 ไฟล์ ดังนี้
 1) regist.php ใช้สำหรับสมัครสมาชิกของระบบจากผู้ใช้งาน และ 2) member.php ใช้สำหรับแสดงผล
 การสมัครสมาชิก รูปแบบการเรียกใช้ดูข้อมูลจากฟอร์มจะขึ้นอยู่กับวิธีการส่งข้อมูล มีรูปแบบดังนี้

ตารางที่ 9.2 ความสัมพันธ์ระหว่างวิธีการส่งและรูปแบบการเรียกใช้ข้อมูล

วิธีส่งข้อมูล	รูปแบบการเรียกใช้ข้อมูล
POST	<code>\$_POST ["ชื่อส่วนนำเข้าข้อมูล"]</code>
GET	<code>\$_GET ["ชื่อส่วนนำเข้าข้อมูล"]</code>

จากตารางที่ 9.2 แสดงความสัมพันธ์ระหว่างวิธีการส่งและรูปแบบการเรียกใช้ข้อมูล จะเห็นได้ว่า
 วิธีการส่งข้อมูลจากฟอร์มจะมีผลต่อรูปแบบการเรียกใช้ข้อมูล ดังนั้นจะต้องตรวจสอบวิธีการส่งให้ชัดเจน
 ไม่เช่นนั้นอาจทำให้การเรียกใช้งานไม่ถูกต้อง นอกจากรูปแบบการเรียนใช้ข้อมูล `$_POST` และ `$_GET`
 แล้วยังสามารถใช้ `$_REQUEST` แทนวิธีการส่งข้อมูลทั้ง 2 วิธีได้

9.3.1 regist.php มีสคริปต์ดังนี้

ตัวอย่างที่ 9.9 แสดงสคริปต์ของไฟล์ regist.php

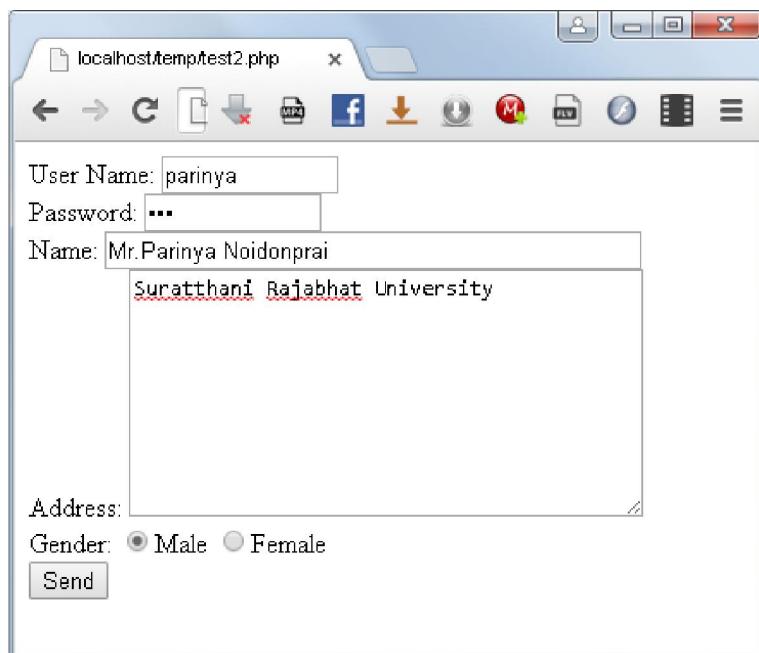
```
<form name= "frmMember" method = "POST" action = "member.php">
    User Name: <input name = "txtUser" type = "text" size = "10"><br>
    Password: <input name = "txtPass" type = "password" size = "10"><br>
    Name: <input name = "txtName" type = "text" size = "40"><br>
    Address: <textarea name = "txtAddress" cols = "40" row = "5"></textarea><br>
```



Gender:

```
<input type = "radio" name = "rdoGender" value = "M" checked = "checked">Male
<input type = "radio" name = "rdoGender" value = "F">Female
<input type = "submit" name = "btnSubmit" value = "Send">
</form>
```

ผลลัพธ์



ภาพที่ 9.8 แสดงตัวอย่างหน้าเว็บเพจเมื่อเรียกคุ้นไฟล์ regist.php บนเว็บบราวเซอร์

จากภาพที่ 9.8 เมื่อป้อนข้อมูลต่างๆ ตามลำดับส่วนนำเข้าข้อมูลเรียบร้อยแล้ว เมื่อกดปุ่ม Send จะส่งข้อมูลทั้งหมดไปประมวลผลที่ไฟล์ member.php โดยสคริปต์ของ member.php มีรายละเอียดดังนี้

9.3.2 member.php มีสคริปต์ดังนี้

ตัวอย่างที่ 9.10 แสดงสคริปต์ของไฟล์ member.php

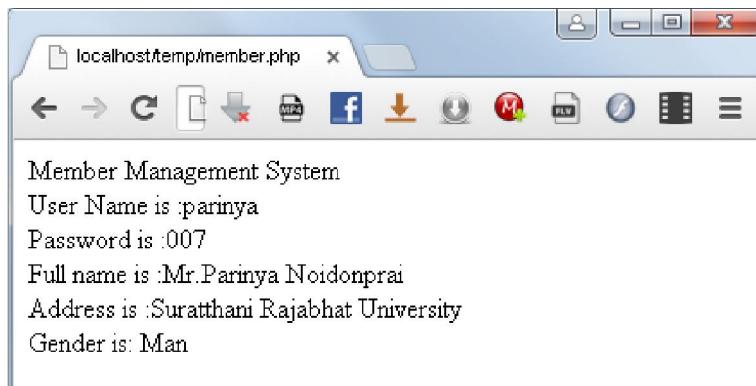
```
1 <?php
2     printf ("Member Management System<br>");
3     printf ("User Name is :%s <br>", $_POST["txtUser"]);
4     printf ("Password is :%s <br>", $_POST["txtPass"]);
5     printf ("Full name is :%s <br>", $_POST["txtName"]);
6     printf ("Address is :%s <br>", $_POST["txtAddress"]);
7     if ($_POST["rdoGender"]=="M") {
8         printf ("Gender is: Man");
```

```

9         } else {
10            printf ("Gender is: Female");
11        }
13    ?>

```

ผลลัพธ์



ภาพที่ 9.9 แสดงตัวอย่างหน้าเว็บเพจเมื่อคลิก Send เพื่อส่งข้อมูลไปประมวลผลที่ member.php

จากตัวอย่างที่ 9.10 แสดงสคริปต์ของไฟล์ member.php เพื่อใช้เรียกดูข้อมูลที่ถูกส่งมา จากไฟล์ regist.php ด้วยวิธีการส่งรูปแบบ POST ดังนั้นส่วนนำเข้าข้อมูลทั้งหมดที่ส่งมาจากไฟล์ regist.php เมื่อประมวลผลในไฟล์ member.php จะถูกใส่ลงในตัวแปรพิเศษ \$_POST โดยบรรทัดที่ 2-8 จะเป็นการแสดงผลค่าตัวแปรปกติ ส่วนในบรรทัดที่ 7 มีการใช้โครงสร้างเงื่อนไขเพื่อตรวจสอบค่าตัวแปร (\$_POST["rdoGender"]) เป็นเพราะค่าที่ถูกส่งจาก regist.php มีค่าเที่ยง 2 ค่า คือ M แทนค่า Man และ F แทนค่า Female ดังนั้นการแสดงผลจำนวนเป็นแปลงข้อมูลกลับเพื่อให้ผู้ใช้สามารถอ่านและแปลความหมายได้ง่าย

9.4 การจัดเก็บข้อมูลลงไฟล์

สำหรับในหัวข้อนี้จะประยุกต์เพิ่มเติมจากตัวอย่างก่อนหน้า เพื่อจัดเก็บข้อมูลสมาชิกลงในไฟล์ จะเพิ่มเติมในสคริปต์ member.php ดังนี้

ตัวอย่างที่ 9.11 แสดงสคริปต์เพื่อจัดเก็บข้อมูลลงไฟล์ เพิ่มเติมจาก member.php

```

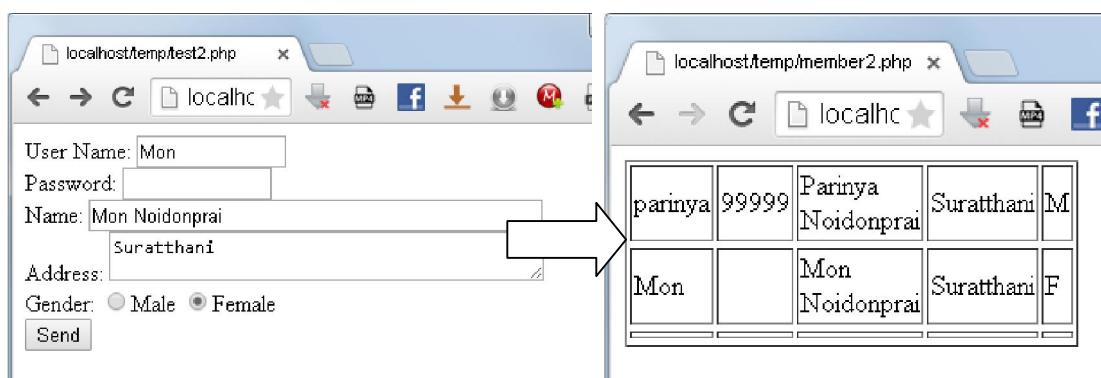
1<?php
2   function show_member ($filemember) {
3     $fp = file ($filemember);
4     $row = explode ("-", $fp[0]);
5     echo "<table border=1 width=50%>";
6     for($var=0; $var <= count($row)-1; $var++){
7       echo "<tr>";

```



```
8      $field = explode ("*", $row [$var]);
9      for($col = 0; $col <= 4; $col++) {
10         echo "<td>"; echo $field[$col]; echo "</td>";
11     }
12     echo "<tr>";
13 }
14 echo "</table>";
15 }
16 $filemember = "member.txt";
17 if (file_exists ($filemember)) {
18     $member = fopen ($filemember, "a+");
19     if ($member==false) die ("Can not create member.txt");
20     $table=sprintf("%s">%s">%s">%s%"$-", $_POST["txtUser"],$_POST["txtPass"],
$_POST["txtName"], $_POST["txtAddress"], $_POST["rdoGender"]);
21     fwrite ($member, $table);
22     fclose ($member);
23     show_member ($filemember);
24 } else {
25     echo "<br>Creating member.txt in first time<br>";
26     $member = fopen ($filemember, "w");
27     if ($member==false) die ("Can not create member.txt");
28     $table=sprintf("%s">%s">%s">%s%"$-", $_POST["txtUser"],$_POST["txtPass"],
$_POST["txtName"], $_POST["txtAddress"], $_POST["rdoGender"]);
29     fwrite ($member, $table);
30     fclose ($member);
31     show_member($filemember);
32 }
33?>
```

จากตัวอย่างที่ 9.11 แสดงสคริปต์เพื่อจัดเก็บข้อมูลลงเท็กซ์ไฟล์ โดยสามารถเพิ่มข้อมูลสมาชิกได้เรียบๆ พร้อมหน้ารายงานผล ดังภาพที่ 9.10



ภาพที่ 9.10 แสดงหน้าเพิ่มข้อมูลสมาชิก และหน้ารายงานผลข้อมูลสมาชิก

สรุป

การพัฒนาเว็บแอปพลิเคชันที่สมบูรณ์จะต้องมีการรับค่าข้อมูลหรือนำเข้าข้อมูลเพื่อส่งไปประมวลผลให้ได้ผลลัพธ์ตามที่ได้ออกแบบไว้ การนำเข้าข้อมูลสามารถนำเข้าผ่านทางฟอร์มของภาษา HTML โดยจะต้องระบุวิธีการส่งข้อมูลให้ถูกต้อง และกำหนดชื่อส่วนนำเข้าข้อมูลให้ชัดเจน เพราะจะมีผลต่อการเรียกใช้ในไฟล์สคริปต์ที่ระบุในฟอร์ม ซึ่งในแท็กของฟอร์มนั้นจะเป็นต้องระบุส่วนนำเข้าข้อมูลคุณสมบัติต่างๆ ที่เกี่ยวข้อง เช่น ความกว้าง จำนวนบรรทัด ลักษณะการแสดงผลตัวเลือกทั้งแบบเลือกตอบได้เพียงครั้งเดียว และแบบเลือกซ้ำได้ เป็นต้น ดังนั้นผู้พัฒนาระบบจะต้องศึกษาและพิจารณาเลือกใช้ตามความเหมาะสมของงาน

คำถามท้ายบท

1. จงอธิบายถึงรูปแบบการใช้งานแท็ก <form> และอธิบายถึงการส่งข้อมูลมีวิธีใด มีความเหมือนและแตกต่างกันอย่างไรของแต่ละวิธี
2. จงอธิบายหลักการทำงานของสคริปต์ที่กำหนดให้ต่อไปนี้

```
<form name= "frmMember" method = "POST" action = "member.php">
    User Name: <input name = "txtUser" type = "text" size = "10"><br>
    Password: <input name = "txtPass" type = "password" size = "10"><br>
    Name: <input name = "txtName" type = "text" size = "40"><br>
    Address: <textarea name = "txtAddress" cols = "40" row = "5"></textarea>
</form>
```

3. จงอธิบายหลักการทำงานของสคริปต์ที่กำหนดให้ต่อไปนี้

```
<form name= "frmtest" method = "POST" action = "#>
    Gender:
    <input type ="radio" name = "rdoTest" value = "1">Apple
```

```
<input type = "radio" name = "rdoTest" value = "2">Coconut  
<input type = "radio" name = "rdoTest" value = "3" checked = "checked">Pineapple  
<input type = "radio" name = "rdoTest" value = "4">Durian  
</form>
```

4. จงอธิบายหลักการทำงานของสคริปต์ที่กำหนดให้ต่อไปนี้

```
<form name= "frmTest" method = "POST" action = "#>  
Provincce:  
<select name = "sltProvince">  
    <option value = "1" selected = "selected">Suratthani</option>  
    <option value = "2">Chumphon</option>  
    <option value = "3">Ranong</option>  
</select>  
</form>
```

5. จงอธิบายและยกตัวอย่าง ส่วนนำเข้าข้อมูลกลุ่มปุ่มคำสั่ง

บทที่ 10

การเข้มโยงระหว่างเว็บเพจ คุกเก้ และเซสชัน

การเข้มต่อระหว่างบราวเซอร์และเว็บเซิร์ฟเวอร์นั้น จะเกิดขึ้นเมื่อบราวเซอร์ส่งการร้องขอ (Request) ออกไป และเมื่อเว็บเซิร์ฟเวอร์ส่งข้อมูลผลลัพธ์กลับมาครบถ้วนแล้ว การเข้มต่อจะสิ้นสุดลง จะส่งผลให้ข้อมูลต่างๆ ที่ถูกสร้างขึ้นภายในเว็บเพจที่เรียกใช้ในขณะนั้นถูกทำลายลงไปด้วย แต่ถ้าข้อมูลที่ถูกสร้างขึ้นในเว็บเพจนี้ และมีความจำเป็นต้องนำไปใช้ในเว็บเพจอื่นๆ ด้วย ทำอย่างไรจึงจะเก็บรักษาข้อมูลนั้นเอาไว้ได้ ดังนั้นในบทนี้จะมาศึกษาถึงวิธีการที่จะจัดเก็บข้อมูลเพื่อให้สามารถนำไปใช้งานระหว่างเว็บเพจ ที่จะกล่าวถึงในบทนี้ คือ การเข้มโยงระหว่างเว็บเพจ คุกเก้และเซสชัน

10.1 เฮดเดอร์

เฮดเดอร์ (Header) คือ ข้อมูลบางอย่างที่ใช้ในการสื่อสารกันระหว่างเว็บเซิร์ฟเวอร์ และบราวเซอร์ เฮดเดอร์อาจจะเป็นข้อมูลที่ส่งมาจากเว็บเซิร์ฟเวอร์ไปยังบราวเซอร์ หรือส่งจากบราวเซอร์ไปยังเว็บเซิร์ฟเวอร์ก็ได้ โดยวัตถุประสงค์ของการส่งヘดเดอร์นั้นมีหลายลักษณะ การกำหนดヘดเดอร์จะใช้ฟังก์ชัน header () มีรูปแบบดังนี้

รูปแบบ

```
void header ( string $string [, bool $replace = true [, int $http_response_code ]] )
```

เมื่อ \$string หมายถึง ชื่อヘดเดอร์ มีหลายลักษณะขึ้นอยู่กับว่าจะส่งข้อมูลอะไรไปยังบราวเซอร์

\$replace หมายถึง การแทนที่ ค่าโดยปริยายคือ true

\$http_response_code หมายถึง ลักษณะการตอบสนอง จะต้องกำหนดให้อยู่ในสตริงเดียวกันกับชื่อヘดเดอร์

ตัวอย่างที่ 10.1 การย้ายหน้าเว็บเพจ (Webpage Redirect) ไปยัง URL อื่น ๆ

```
1 <?php  
2     header ('Location: http://www.freebsd.sru.ac.th/');  
3 ?>
```

จากตัวอย่างที่ 10.1 การย้ายหน้าเว็บเพจ (Webpage Redirect) ไปยัง URL อื่น ๆ โดยใช้ฟังก์ชัน header () โดยพารามิเตอร์ที่ระบุภายในฟังก์ชัน คือ URL ที่ต้องการย้ายหรือไปทำงานต่อ

ตัวอย่างที่ 10.2 การย้ายหน้าเว็บเพจแบบ refresh (หน่วงเวลา ก่อนเปลี่ยนหน้าเว็บเพจ มีหน่วยเป็นวินาที)

```

1 <?php
2     header ('Refresh: 10; url=http://www.freebsd.sru.ac.th/');
3 ?>

```

จากตัวอย่างที่ 10.2 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header () สำหรับย้ายหน้าเว็บเพจ แบบกำหนดพารามิเตอร์ refresh (หน่วงเวลา ก่อนเปลี่ยนหน้าเว็บเพจ มีหน่วยเป็นวินาที) และกำหนดเวลา 10 นาที ก่อนทำการย้าย

ตัวอย่างที่ 10.3 การกำหนดภาษาที่ใช้ในหน้าเว็บ

```

1 <?php
2     header ('Content-language: en');
3 ?>

```

จากตัวอย่างที่ 10.3 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header () สำหรับกำหนดภาษาที่ใช้แสดงภายในหน้าเว็บเพจ จากตัวอย่างเป็นการกำหนดให้แสดงผลเป็นภาษาอังกฤษ

ตัวอย่างที่ 10.4 การใช้ฟังก์ชัน header () สำหรับสร้างไฟล์สำหรับการดาวน์โหลด

```

1 <?php
2     header ('Content-Type: application/octet-stream');
3     header ('Content-Disposition: attachment; filename="example.zip"');
4     header ('Content-Transfer-Encoding: binary');
5 ?>

```

จากตัวอย่างที่ 10.4 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header () สำหรับสร้างไฟล์สำหรับการดาวน์โหลด

ตัวอย่างที่ 10.5 การสร้างเว็บเพจโดยไม่มีการเขียนเว็บแคช (web cache)

```

1 <?php
2     header ('Cache-Control: no-cache, no-store, max-age=0, must-revalidate');
3     header ('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
4 ?>

```

จากตัวอย่างที่ 10.5 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน header () สำหรับการสร้างเว็บเพจโดยไม่มีการเขียนเว็บแคช เว็บแคชทำให้เข้าถึงเว็บไซต์ได้เร็วขึ้น เพราะจะไม่มีการประมวลผลใดๆ ของหน้าเว็บไซต์ที่ผู้ใช้งานเรียก แต่อย่างไรก็ตามเว็บแคช ก็ยังมีข้อเสียในการทำงาน นั่นคือข้อมูลที่ได้ อาจจะไม่ อัปเดต เพราะฉะนั้นระบบเว็บแคชจึงจำเป็นที่จะต้องมีการตั้งเวลาให้หมดอายุของแคช และทำการอัปเดตแคชใหม่ ถ้าหากว่าเวลาของแคชหมดเกินไป ก็จะทำให้ข้อมูลอัปเดตช้าไปด้วย

ตัวอย่างที่ 10.6 การกำหนดประเภทของการแสดงเนื้อหาของเว็บ (web content)

```

1 <?php
2     header ('Content-Type: text/html; charset=iso-8859-1');
3     header ('Content-Type: text/html; charset=utf-8');
4     header ('Content-Type: text/plain');           // plain text file
5     header ('Content-Type: image/jpeg');          // JPG picture
6     header ('Content-Type: application/zip');      // ZIP file
7     header ('Content-Type: application/pdf');      // PDF file
8     header ('Content-Type: audio/mpeg');          // Audio MPEG (MP3,...) file
9 ?>

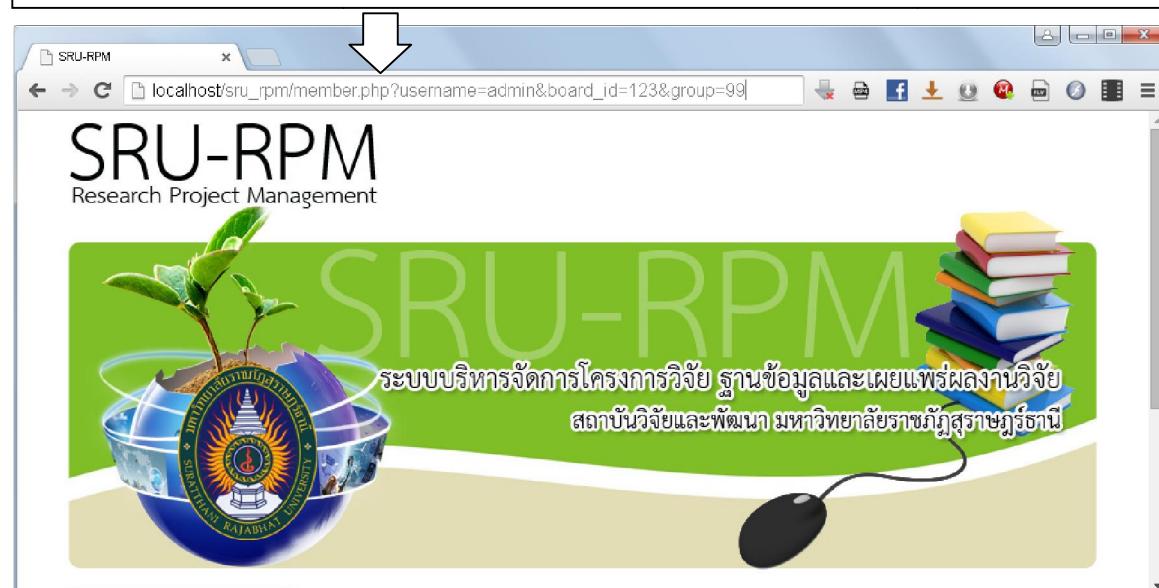
```

จากตัวอย่างที่ 10.6 เป็นตัวอย่างการประยุกต์ใช้ฟังก์ชัน `header()` สำหรับการกำหนดประเภทของการแสดงเนื้อหาของเว็บ

10.2 การส่งข้อมูลระหว่างเว็บเพจแบบ Query String

Query String คือ รูปแบบการส่งข้อมูลจากเว็บเพจนึงไปยังอีกเว็บเพจนึง โดยแนบข้อมูลต่อท้าย URL มีตัวอย่างดังต่อไปนี้

```
http://localhost/sru_rpm/member.php?username=admin&board_id=123&group=99
```



ภาพที่ 10.1 แสดงตัวอย่างรูปแบบของ Query String

จากตัวอย่างที่ 10.1 แสดงตัวอย่างรูปแบบ Query String จะเห็นได้ว่าตัวแปรและค่าของตัวแปรแต่ละชุดจะใช้สัญลักษณ์คั่นเป็นช่วงๆ ตามลำดับ โดยมีรายละเอียดของสัญลักษณ์ ดังตารางที่ 10.1

ตารางที่ 10.1 สัญลักษณ์ที่ใช้ร่วมกับ Query String เพื่อการส่งค่าระหว่างเว็บเพจ

สัญลักษณ์	ความหมาย
?	ใช้ค้น URL กับข้อมูลที่จะแนบไป
&	ใช้คั่นตัวแปรแต่ละตัว
=	ใช้คั่นระหว่างตัวแปรและค่าของตัวแปร เช่น password=123
+	ใช้แทนช่องว่าง 1 ช่องของชุด URL เช่น keyword=parinya+noidonprai
%	ใช้แทนอักษรพิเศษที่ไม่ใช่ตัวอักษรหรือตัวเลข โดยจะใช้เครื่องหมาย % นำหน้า เลขฐาน 16 ที่จะใช้แทนอักษรตัวหนอนๆ เช่น %28 ใช้แทนเครื่องหมายวงเล็บเปิด เป็นต้น

สำหรับการอ่านข้อมูลที่เว็บเพจป้ายทางที่ได้รับข้อมูล Query String สามารถเรียกใช้ได้หลายลักษณะ ดังนี้

10.2.1 ใช้ตัวแปร `$_GET` เมื่อกับการอ่านข้อมูลที่ส่งจากฟอร์มด้วยวิธี GET ตามปกติ เช่น จากภาพที่ 10.1 `$member = $_GET ["username"];` หมายความว่า กำหนดให้ตัวแปร `$member` มีค่าเท่ากับ `admin` เพราะ `$_GET ["username"]` มีค่าเท่ากับ `admin`

10.2.2 ถ้าต้องการอ่าน Query String ทั้งหมดออกจาก URL เพื่อส่งต่อไปยังเว็บเพจอื่นๆ ก็สามารถอ่านจากตัวแปร `$_SERVER ['Query_STRING']` ดังตัวอย่างที่ 10.7

ตัวอย่างที่ 10.7 การอ่าน Query String ด้วย `$_SERVER ['Query_STRING']`

```

1 <?php
2     $qrystr = $_SERVER ['Query_STRING'];
3     $goto = "nextpage.php" . "?" . $qrystr;
4     header ("Location: $goto");
5 ?>

```

จากตัวอย่างที่ 10.7 การอ่าน Query String ทั้งหมดออกจาก URL ด้วย `$_SERVER ['Query_STRING']` อธิบายร่วมกับภาพที่ 10.1 ดังนี้

บรรทัดที่ 2 จะหมายความว่า กำหนดให้ตัวแปร `$qrystr` มีค่าเท่ากับ `"username=admin &board_id=123&group=99"`

บรรทัดที่ 3 กำหนดให้ตัวแปร `$goto` มีค่าเท่ากับ `"nextpage.php?username=admin &board_id=123&group=99"` (คือ การต่อข้อความให้เป็นประโยชน์เดียวกัน)

บรรทัดที่ 4 ใช้ฟังก์ชัน `header ()` เพื่อย้ายหน้าเว็บปัจจุบันไปยังเว็บเพจ `nextpage.php` พร้อมด้วยค่า Query String เพื่อนำค่าทั้งหมดไปประมวลผลต่อ

10.2.3 ใช้ฟังก์ชัน `parse_str()` จะได้ผลลัพธ์เป็นตัวแปรพารามิเตอร์ และค่าของตัวแปรนั้นๆ มีตัวอย่างการใช้งาน ดังนี้

ตัวอย่างที่ 10.8 การอ่าน Query String ทั้งหมดจากมา URL ด้วยฟังก์ชัน `parse_str()`

```

1 <?php
2     $qrystr = $_SERVER ['Query_STRING'];
3     parse_str ($qrystr);
4     echo "$username <br>";
5     echo "$board_id <br>";
6     echo "$group <br>";
7 ?>

```

ผลลัพธ์

```

admin
123
99

```

จากตัวอย่างที่ 10.8 การอ่าน Query String ทั้งหมดจากมา URL ด้วยฟังก์ชัน `parse_str()` อธิบายร่วมกับภาพที่ 10.1 ดังนี้

บรรทัดที่ 2 จะหมายความว่า กำหนดให้ตัวแปร `$qrystr` มีค่าเท่ากับ "username=admin &board_id=123&group=99"

บรรทัดที่ 3 ใช้ฟังก์ชัน `parse_str ($qrystr)` ผลของฟังก์ชันจะได้ตัวแปรพร้อมใช้งาน ในหน้าเว็บเพจที่ใช้ฟังก์ชัน ประกอบด้วยตัวแปร ดังนี้

บรรทัดที่ 4 แสดงผลค่าตัวแปร `$username`

บรรทัดที่ 5 แสดงผลค่าตัวแปร `$board_id`

บรรทัดที่ 6 แสดงผลค่าตัวแปร `$group`

10.3 การจัดเก็บข้อมูลแบบคุกเก้

คุกเก้ (Cookie) เป็นรูปแบบของการเก็บรักษาข้อมูลบางอย่างไว้บนเครื่องของผู้ใช้ เพื่อจะนำข้อมูลนี้กลับมาใช้ใหม่ภายหลังได้ โดยข้อมูลที่เก็บในแบบคุกเก้ไม่ควรเป็นข้อมูลที่ต้องเก็บเป็นความลับ เช่น รหัสผ่าน เป็นต้น เนื่องจากข้อมูลคุกเก้จะเก็บไว้ในรูปแบบของไฟล์ข้อความธรรมดา ระดับความปลอดภัยจึงค่อนข้างต่ำ แต่อย่างไรก็ตาม เนื่องจากผู้ใช้งานกลุ่มกลับเห็นว่า คุกเก้นน์เป็นการล่วงละเมิดสิทธิของผู้ใช้ ดังนั้นโปรแกรมเว็บบราวเซอร์จึงมีตัวเลือกสำหรับให้ผู้ใช้ สามารถปฏิเสธการบันทึกคุกเก้จากเว็บไซต์ที่ไม่ต้องการได้ ด้วยเหตุนี้การใช้คุกเก้จึงอาจไม่บรรลุผลเสมอไป



นอกจากการปฏิเสธคุกกี้จากผู้ใช้แล้ว ยังมีสาเหตุอื่นๆ ที่อาจทำให้ไม่สามารถใช้คุกกี้ได้ เช่น การหมดอายุของคุกกี้ หรือไฟล์ที่จัดเก็บคุกกี้อาจเสียหาย ใช้การไม้ได้ หรือถูกทำลาย เช่น การติดตั้งระบบใหม่ เป็นต้น ดังนั้นต้องจัดเตรียมทางเลือกไว้เพื่อรับมือด้วย

10.3.1 การสร้างคุกกี้

ฟังก์ชันหลักในการใช้งานเกี่ยวกับคุกกี้ คือ ฟังก์ชัน setcookie () มีรูปแบบดังนี้

รูปแบบ

```
bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path [, string $domain]]]] )
```

เมื่อ	\$name	หมายถึง ชื่อคุกกี้ที่ต้องการสร้าง
	\$value	หมายถึง ค่าที่จะกำหนดให้กับคุกกี้
	\$expire	หมายถึง อายุของคุกกี้
	\$path	หมายถึง พารามิเตอร์ที่คุกกี้สามารถใช้งานได้
	\$domain	หมายถึง กำหนดโดเมนที่สามารถอ่านคุกกี้ได้

ตัวอย่างที่ 10.9 การใช้ฟังก์ชัน setcookie ()

```
1 <?php
2     $value = "something from somewhere";
3     setcookie ("TestCookie", $value);
4     setcookie ("TestCookie", $value, time() + 3600); /* หมดอายุใน 1 ชั่วโมง */
5     setcookie ("TestCookie", $value, time() + 3600, "/~rasmus/", "example.com", 1);
6 ?>
```

จากตัวอย่างที่ 10.9 ตัวอย่างการใช้ฟังก์ชัน setcookie () กรณีกำหนดชื่อคุกกี้ซ้ำกับชื่อที่มีอยู่แล้ว จะเป็นการแทนที่ข้อมูลเดิม หรือเท่ากับเป็นการแก้ไขค่าของคุกกี้นั้นเอง สำหรับรูปแบบของฟังก์ชัน setcookie () นี้เป็นแบบอย่างง่ายเท่านั้น ในความเป็นจริงสามารถกำหนดพารามิเตอร์เพิ่มเติมได้อีกหลายรูปแบบ

10.3.2 การอ่านข้อมูลจากคุกกี้

การอ่านคุกกี้จะใช้ตัวแปร \$_COOKIE เป็นตัวแปรแบบอาร์เรย์ โดยระบุชื่อของคุกกี้เป็นค่าคีย์ ก็จะได้ค่าของคุกกี้ ที่กำหนดไว้ตอนสร้างคุกกี้ แต่อย่างไรก็ตาม หากคุกกินั้นไม่มีอยู่จริง หรือหมดอายุไปแล้ว การอ่านคุกกินั้นโดยไม่ตรวจสอบก่อนจะเกิดข้อผิดพลาดขึ้น ดังนั้นเพื่อให้แน่ใจว่ามีคุกกี้นั้นอยู่จริงหรือไม่ ควรตรวจสอบด้วยฟังก์ชัน isset () ก่อนอ่านค่าเสมอ ตัวอย่างดังนี้

ตัวอย่างที่ 10.10 ตัวอย่างการตรวจสอบตัวแปรคุกเก้ก่อนการอ่านค่าจากคุกเก้

```

1 <?php
2     if (isset ($_COOKIE ["book"])) {
3         $b = $_COOKIE ["book"];
4     }
5 ?>

```

จากตัวอย่างที่ 10.10 ตัวอย่างการตรวจสอบตัวแปรคุกเก้ก่อนการอ่านค่าจากคุกเก้ เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้

10.3.3 การกำหนดอายุ และลบคุกเก้

ปกติแล้วหากสร้างคุกเก้ด้วยวิธีการตามที่กล่าวมาแล้ว คือ ไม่ได้กำหนดอายุ จะทำให้คุกเก้หมดอายุทันทีที่ปิดบราวเซอร์ แต่โดยทั่วไปมักจะเก็บข้อมูลคุกเก้นี้ไว้ใช้ในคราวต่อๆ ไปด้วย

การกำหนดอายุของคุกเก้จะกำหนดเป็นเวลาในหน่วยวินาทีโดยเทียบกับเวลาปัจจุบัน โดยทั่วไปแล้วมักอ้างอิงเวลาปัจจุบันด้วยฟังก์ชัน time () ฟังก์ชันนี้จะคืนค่า timestamp ของเวลาปัจจุบัน ดังนั้นสามารถนำอายุของคุกเก้มาบวกเพิ่มเข้าไปได้เลย เช่น time () + 30 หมายความว่ากำหนดให้คุกเก้หมดอายุภายใน 30 วินาที หลังจากนี้ หรือหากต้องการกำหนดเวลาในหน่วยที่ยาวนานกว่านี้ต้องนำมาแปลงเป็นวินาที เช่น หากต้องการให้คุกเก้มีอายุ 1 วัน ต้องเขียนเป็น time () + (24 * 60 * 60) เป็นต้น มีตัวอย่างดังนี้

ตัวอย่างที่ 10.11 การกำหนดอายุของคุกเก้

```

1 <?php
2     $expire = time ( ) + (24 * 60 * 60);
3     setcookie ("mycookie", "Parinya Noidonprai", $expire);
4 ?>

```

จากตัวอย่างที่ 10.11 การกำหนดอายุของคุกเก้ให้มีอายุ 1 วัน แต่หากต้องการลบคุกเก้ที่ชื่อ "mycookie" สามารถประยุกต์ตามตัวอย่างที่ 10.12 ดังนี้

ตัวอย่างที่ 10.12 การลบคุกเก้

```

1 <?php
2     $expire = time ( ) - 3600; // ลบด้วยเลขอะไรก็ได้เพื่อให้เป็นเวลาที่ผ่านมาแล้ว
3     setcookie ("mycookie", "", $expire);
4 ?>

```

จากตัวอย่างที่ 10.12 การลบคุกเก้ โดยใช้เทคนิควิธีการย้อนเวลาเพื่อให้คุกเก้หมดเวลา หรือความหมายคือการลบคุกเก้ออกนั่นเอง



10.4 การจัดเก็บข้อมูลแบบเซสชัน

เซสชัน (Session) เป็นการจัดเก็บข้อมูลบางอย่างไว้ชั่วคราวในฝั่งของเว็บเซิร์ฟเวอร์ เพื่อให้สามารถนำข้อมูลกลับมาใช้งานใหม่หลังจากที่เว็บเพจนั้นสิ้นสุดการทำงานไปแล้ว และยังสามารถนำไปใช้งานที่เว็บเพจอื่นๆ ได้

10.4.1 ลักษณะและการทำงานของเซสชัน มีดังนี้

1) เมื่อผู้ใช้เข้ามายังเว็บไซต์แต่ละแห่ง เว็บเซิร์ฟเวอร์จะสร้างรหัสสำหรับการอ้างอิงถึงผู้ใช้แต่ละคน โดยรหัสนี้เรียกว่า Session ID หรือ SID ประกอบด้วยตัวเลข 0 - 9 และตัวอักษร a - z จำนวน 32 ตัว เช่น d77a4f704b938b240e4228a7e0847895 ทั้งนี้ค่า SID ของผู้ใช้แต่ละคนที่เข้ามายังเว็บไซต์ในช่วงเวลาเดียวกันจะต้องมี SID ไม่ซ้ำกัน

2) ค่า SID นี้จะถูกนำมาใช้ในการอ้างอิงถึงผู้ใช้คนนั้นตลอดการเข้ามายังต่อ และเมื่อใดก็ตามที่ผู้ใช้เลิกการเข้ามายังต่อ เช่น การปิดบราวเซอร์ จะส่งผลให้ค่า SID ของผู้ใช้คนนั้นถูกยกเลิกไป จนกว่าจะมีการเข้ามายังเว็บไซต์แห่งนั้น ครั้งต่อไปค่า SID จึงจะถูกสร้างขึ้นมาใหม่เป็นเช่นนี้ไปตลอด

3) เมื่อสร้างข้อมูลแบบเซสชันสำหรับผู้ใช้คนใด ข้อมูลนั้นก็จะมีผลหรือใช้งานได้เฉพาะกับผู้ใช้คนนั้นส่วนคนอื่นๆ ไม่สามารถเรียกใช้งานได้ เนื่องจากข้อมูลเซสชันของผู้ใช้แต่ละคนจะถูกแยกออกจากกัน

4) ข้อมูลแบบเซสชันที่สร้างขึ้นจากเว็บเพจนั้นจะสามารถนำไปใช้งานที่เว็บเพจอื่นๆ ที่ถูกเรียกใช้โดยผู้ใช้ที่เป็นผู้สร้างเซสชันนั้นได้

5) ข้อมูลเซสชันที่ถูกสร้างขึ้นจะสามารถใช้งานได้ชั่วระยะเวลาที่เปิดใช้บราวเซอร์อยู่ แต่ไม่เกิน 180 นาที ถ้าปิดบราวเซอร์หรือเวลาเกินกว่า 180 นาที ข้อมูลก็จะถูกยกเลิกไป หรือนอกจากนี้ ยังสามารถสั่งลบข้อมูลเซสชันเมื่อไม่ต้องการใช้งานต่อไปได้

6) เซสชันจะมีความเกี่ยวข้องกับคุกเก้ โดยเซสชันจะใช้การเก็บข้อมูลรหัสแบบคุกเก้ไว้ที่เครื่องของผู้ใช้แต่ละคน และเมื่อเข้ามายังเว็บเซิร์ฟเวอร์ ข้อมูลนี้จะถูกนำไปใช้ในการตรวจสอบว่าเป็นผู้ใช้คนใด ดังนั้นหากเครื่องของผู้ใช้คนใดปฏิเสธการใช้คุกเก้จะส่งผลถึงเซสชันด้วย แต่ปัจจุบันนี้มีการใช้เทคนิค URL Rewriting หรือนำรหัสเซสชันไปต่อท้าย URL ในแบบ Query String ในชื่อตัวแปร PHPSESSID

7) แม้เซสชันจะเป็นการพักเก็บข้อมูลไว้ชั่วคราวคล้ายกับคุกเก้ แต่ทั้งคุกเก้และเซสชันก็มีแนวทางการนำไปใช้งานที่แตกต่างกัน

10.4.2 การเริ่มการทำงานของเซสชัน

ก่อนที่จะเริ่มใช้งานเซสชัน ต้องเริ่มด้วยการสั่งให้เซสชันเริ่มทำงานด้วยฟังก์ชัน session_start() โดยต้องกำหนดไว้ที่ส่วนบนสุดของเว็บเพจ ตัวอย่างการใช้งานดังนี้

ตัวอย่างที่ 10.13 การเริ่มการทำงานของเซสชัน

```

1 <?php
2     session_start ( );
3 ?>

```

จากตัวอย่างที่ 10.13 แสดงตัวอย่างการเริ่มการทำงานของเซสชัน โดยต้องกำหนดไว้ที่ส่วนบนสุดของเว็บเพจ

หมายเหตุ

ข้อควรระวัง คือ เนื่องจากเซสชันต้องอาศัยคุกเก้เป็นกลไกร่วมด้วย และคุกเก้จะอาศัยการส่งข้อมูลแบบเซตเดอร์ ทำให้เซสชันมีความเกี่ยวพันกับเซตเดอร์ด้วย ดังนั้นการใช้ฟังก์ชัน session_start () ต้องทำก่อนการส่งข้อมูลใดๆ ไปที่บรัวเซอร์ เช่นเดียวกับเซตเดอร์และคุกเก้

1) การเก็บข้อมูลด้วยตัวแปร \$_SESSION

ตัวแปร \$_SESSION เป็นตัวแปรอาร์เรย์สำหรับจัดเก็บข้อมูลแบบเซสชัน ตัวแปรนี้จะมีลักษณะเป็นคีย์และค่าอาร์เรย์ โดยที่คีย์ คือ ชื่อของเซสชัน (คล้ายกับชื่อของคุกเก้) และค่าอาร์เรย์ คือค่าของเซสชัน ทั้งนี้จะสร้างตัวแปร \$_SESSION จำนวนเท่าไหร่ก็ได้ ตัวอย่างดังนี้

ตัวอย่างที่ 10.14 การเก็บข้อมูลด้วยตัวแปร \$_SESSION

```

1 <?php
2     session_start ( );
3     $_SESSION ["user"] = "Parinya";
4     $_SESSION ["pass"] = $_POST ["pass"];
5 ?>

```

จากตัวอย่างที่ 10.14 แสดงการเก็บข้อมูลด้วยตัวแปร \$_SESSION โดยเมื่อต้องการนำค่าเซสชันได้ก็สามารถระบุผ่านตัวแปร \$_SESSION พร้อมระบุชื่อเซสชันที่ต้องการ แต่อย่างไรก็ตามเนื่องจากเซสชันสามารถถูกลบ หรือหมดอายุได้ ก่อนอ่านค่าตัวแปรควรตรวจสอบการตัวแปรเซสชันก่อนการเรียกใช้งาน โดยใช้ฟังก์ชัน iset () ตรวจสอบก่อนเรียกใช้งานตัวแปร ตัวอย่างดังนี้

ตัวอย่างที่ 10.15 การใช้งานตัวแปร \$_SESSION

```

1 <?php
2     session_start ( );
3     if (isset ($_SESSION ["login"])) $login = $_SESSION ["login"];
4 ?>

```



10.4.3 การลบข้อมูลเซสชัน

ปกติแล้วเมื่อ bravacheor ที่เปิดเอาไว้ถูกปิดทั้งหมด หรือระยะเวลาที่สร้างเซสชันเอาไว้นานเกินกว่า 180 นาที (3 ชั่วโมง) ค่าตัวแปรเซสชันที่ถูกสร้างขึ้นก็จะยกเลิกไปเองโดยอัตโนมัติ แต่บางครั้งก็ต้องการยกเลิกข้อมูลเซสชันที่เวลาได้เวลาหนึ่งขณะที่ยังใช้งานเว็บไซต์อยู่ เช่น กรณีการออกจากระบบ (Logout) เป็นต้น การลบข้อมูลเซสชันสามารถใช้ฟังก์ชัน ดังนี้

1) ฟังก์ชัน `unset()` เป็นฟังก์ชันที่ใช้ในการยกเลิกตัวแปรทั่วๆ ไปของ PHP สามารถนำมายังกับตัวแปรเซสชันได้เช่นกัน มีรูปแบบดังนี้

รูปแบบ

```
void unset ( mixed $var [, mixed $... ] )
```

เมื่อ `$var` หมายถึง ชื่อของตัวแปรที่จะลบหรือยกเลิก

ตัวอย่างที่ 10.16 การใช้ฟังก์ชัน `unset()`

```
1 <?php
2 session_start ();
3 $_SESSION ["name"] = "Mr.Parinya";
4 echo $_SESSION ["name"] . "<br>";
5 unset ($_SESSION ["name"]);
6 echo $_SESSION ["name"] . "<br>"; // จะไม่พบร่องรอยใดๆ เนื่องจากได้ลบแล้ว
7 ?>
```

2) ฟังก์ชัน `session_destroy()` เป็นฟังก์ชันที่ใช้สำหรับลบข้อมูลหรือยกเลิกตัวแปรเซสชันทั้งหมดในระบบที่ได้สร้างขึ้นให้โดยผู้ใช้คนนั้นๆ มีรูปแบบดังนี้

รูปแบบ

```
bool session_destroy ( void )
```

หมายเหตุ

ฟังก์ชัน `session_destroy()` ใช้สำหรับลบหรือยกเลิกตัวแปรเซสชันทั้งหมด หากต้องการลบตัวแปรเซสชันตัวใดตัวหนึ่ง ขอแนะนำให้เลือกใช้ฟังก์ชัน `unset()`

10.4.4 ฟังก์ชันอื่นๆ ที่เกี่ยวข้องกับเซสชัน

1) ฟังก์ชัน `session_id()` เป็นฟังก์ชันที่ใช้สำหรับคืนค่า SID มีรูปแบบดังนี้

รูปแบบ

```
string session_id ([ string $id ])
```

ตัวอย่างที่ 10.17 การใช้ฟังก์ชัน session_id ()

```

1 <?php
2     session_start();
3     echo session_id();
4 ?>

```

2) ฟังก์ชัน session_regenerate_id () เป็นฟังก์ชันที่ใช้สำหรับการสร้าง SID ใหม่
รูปแบบ

```
bool session_regenerate_id ([ bool $delete_old_session = false ] )
```

ตัวอย่างที่ 10.18 การใช้ฟังก์ชัน session_regenerate_id ()

```

1 <?php
2     session_start();
3     echo session_id();
4     session_regenerate_id();
5     echo session_id();
6 ?>

```

3) ฟังก์ชัน session_encode () เป็นฟังก์ชันที่ใช้สำหรับเข้ารหัสข้อมูลเซสชันทั้งหมด
โดยจะคืนค่าข้อมูลที่เข้ารหัสไว้แล้ว

รูปแบบ

```
string session_encode ( void )
```

ตัวอย่างที่ 10.19 การใช้ฟังก์ชัน session_encode ()

```

1 <?php
2     session_start();
3     $_SESSION ["user"] = "parinya";
4     $_SESSION ["pass"] = "1234";
5     echo session_encode();
6 ?>

```

4) ฟังก์ชัน session_decode () เป็นฟังก์ชันที่ใช้สำหรับถอดรหัสข้อมูล ที่ถูกเข้ารหัส^{ทั้งหมด}
โดย session_encode (); มีรูปแบบดังนี้

รูปแบบ

```
bool session_decode ( string $data )
```



ตัวอย่างที่ 10.20 การใช้ฟังก์ชัน session_decode()

```
1 <?php  
2     session_start();  
3     $_SESSION["user"] = "parinya";  
4     $_SESSION["pass"] = "1234";  
5     $secret = session_encode();  
6     session_decode($secret);  
7 ?>
```

5) ฟังก์ชัน session_register() เป็นฟังก์ชันที่ใช้สำหรับใช้ลงทะเบียน ตัวแปร ให้เป็นตัวแปรเซสชันมีรูปแบบดังนี้

รูปแบบ

```
bool session_register(mixed $name [, mixed $... ])
```

ตัวอย่างที่ 10.21 การใช้ฟังก์ชัน session_register()

```
1 <?php  
2     session_start();  
3     $the_best = "Computer Science of SRU";  
4     session_register("the_best");  
5     echo $_SESSION["the_best"];  
6 ?>
```

จากตัวอย่างที่ 10.21 แสดงการใช้ฟังก์ชัน session_register() เพื่อลงทะเบียนตัวแปรเซสชัน อธิบายดังนี้

บรรทัดที่ 2 เริ่มต้นใช้งานเซสชัน

บรรทัดที่ 3 กำหนดให้ตัวแปร \$the_best มีค่าเท่ากับ "Computer Science of SRU"

บรรทัดที่ 4 ลงทะเบียนตัวแปร \$the_best เป็นตัวแปรชนิดเซสชัน

บรรทัดที่ 5 แสดงผลตัวแปรเซสชัน\$_SESSION["the_best"] ที่ได้ลงทะเบียนไว้ในบรรทัดที่ 4

6) ฟังก์ชัน session_unregister() เป็นฟังก์ชันที่ใช้สำหรับยกเลิกการลงทะเบียนตัวแปรเซสชันมีรูปแบบดังนี้

รูปแบบ

```
bool session_unregister(mixed $name [, mixed $... ])
```

ตัวอย่างที่ 10.22 การใช้ฟังก์ชัน session_unregister()

```

1 <?php
2     $the_best = "Computer Science of SRU";
3     session_register ("the_best");
4     echo $_SESSION ["the_best"];
5     session_unregister ("the_best");
6     echo $_SESSION ["the_best"];
7 ?>

```

จากตัวอย่างที่ 10.22 การใช้ฟังก์ชัน session_unregister() สำหรับยกเลิกการลงทะเบียนตัวแปรเซสชัน อธิบายดังนี้

บรรทัดที่ 2 กำหนดค่าให้กับตัวแปร \$the_best มีค่าเท่ากับ "Computer Science of SRU"

บรรทัดที่ 3 ลงทะเบียนตัวแปรเซสชัน session_register ("the_best")

บรรทัดที่ 4 แสดงผลค่าตัวแปรเซสชัน จะได้ผลลัพธ์ คือ "Computer Science of SRU"

บรรทัดที่ 5 ยกเลิกตัวแปรเซสชัน

บรรทัดที่ 6 แสดงผลค่าตัวแปรเซสชัน (ไม่แสดงข้อมูลใดๆ เนื่องจากตัวแปรเซสชันได้ถูกยกเลิกแล้ว)

7) ฟังก์ชัน session_unset() เป็นฟังก์ชันที่ใช้สำหรับใช้ยกเลิกตัวแปรเซสชันทั้งหมด คือ ยกเลิกตัวแปร ที่ได้ลงทะเบียนไว้มีรูปแบบดังนี้

รูปแบบ

```
void session_unset ( void )
```

ตัวอย่างที่ 10.23 การใช้ฟังก์ชัน session_unset()

```

1 <?php
2     session_start ( );
3     $the_best = "Computer Science of SRU";
4     session_register ( "the_best" );
5     session_unset ( );
6 ?>

```

8) ฟังก์ชัน session_is_registered() เป็นฟังก์ชันที่ใช้สำหรับใช้ตรวจสอบว่าตัวแปรนั้นได้ลงทะเบียนเซสชันแล้วหรือยังมีรูปแบบดังนี้

รูปแบบ

```
bool session_is_registered ( string $name )
```



216 | บทที่ 10 การเชื่อมโยงระหว่างเว็บเพจ คุกเก้ และเซสชัน

ตัวอย่างที่ 10.24 การใช้งานฟังก์ชัน session_is_registered()

```
1 <?php  
2     session_start();  
3     $the_best = "Computer Science of SRU";  
4     if ( ! session_is_registered ( $the_best ) ) {  
5         session_register( $the_best );  
6     }  
7 ?>
```

จากตัวอย่างที่ 10.24 ตัวอย่างการใช้งานฟังก์ชัน session_is_registered() อธิบายดังนี้

บรรทัดที่ 2 เริ่มต้นใช้งานเซสชัน

บรรทัดที่ 3 กำหนดให้ตัวแปร \$the_best มีค่าเท่ากับ "Computer Science of SRU"

บรรทัดที่ 4 ใช้โครงสร้างเงื่อนไข if ร่วมกับฟังก์ชัน session_is_registered() เพื่อตรวจสอบตัวแปรที่ระบุได้ลงทะเบียนเซสชันแล้วหรือไม่ หากยังไม่ได้ลงทะเบียนไปบรรทัด 5

บรรทัดที่ 5 ลงทะเบียนตัวแปร \$the_best เป็นตัวแปรชนิดเซสชัน

10.5 ขั้นตอนการประยุกต์ใช้เซสชัน เพื่อเป็นแนวทางในการพัฒนาระบบ Login

10.5.1 สร้างฟอร์มสำหรับการ Login ประกอบด้วย User Name และ Password ดังตัวอย่างที่ 10.25

ตัวอย่างที่ 10.25 ฟอร์มสำหรับ Login (index.php) มีสคริปต์ ดังนี้

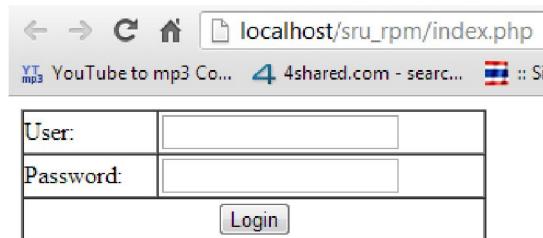
```
<?php  
    session_start();  
    session_destroy();  
?  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>ตัวอย่างหน้า Login</title>  
</head>  
<body>
```

```

<form id="form1" name="frmLogin" method="POST" action="check_login.php">
    <table width="300" border="1" cellspacing="0" cellpadding="0">
        <tr>
            <td width="117">User:</td>
            <td width="283"><label>
                <input type="text" name="user" id="user" />
            </label></td>
        </tr>
        <tr>
            <td>Password:</td>
            <td><label>
                <input type="password" name="pass" id="pass" />
            </label></td>
        </tr>
        <tr>
            <td colspan="2" align="center"><label>
                <input type="submit" name="login" id="login" value="Login" />
            </label></td>
        </tr>
    </table>
</form>
</body>
</html>

```

จากตัวอย่างที่ 10.25 ฟอร์มสำหรับ Login เพื่อเข้าสู่ระบบ กำหนดส่วนนำเข้าข้อมูลที่สำคัญ 3 ส่วน คือ 1) ส่วนนำเข้าข้อมูล ประกอบด้วย user และ pass 2) รูปแบบวิธีส่งฟอร์มใช้รูปแบบวิธีแบบ POST และ 3) скриปต์ที่จะประมวลผลต่อ คือ check_login.php ผลลัพธ์ที่ได้ ดังภาพที่ 10.2



ภาพที่ 10.2 สร้างฟอร์มสำหรับการ Login กรณีศึกษาและประยุกต์ใช้งานเซสชัน

10.5.2 สร้างไฟล์เพื่อตรวจสอบข้อมูลที่ผู้ใช้ Login ผ่านส่วนนำเข้าข้อมูล user และ pass กำหนดให้ใช้ชื่อ check_login.php มี скриปต์คำสั่ง ดังนี้

ตัวอย่างที่ 10.26 สคริปต์สำหรับตรวจสอบข้อมูลที่ผู้ใช้ Login เข้ามา (check_login.php)

```
1 <?php
2 session_start();
3 $user_check="testuser";
4 $pass_check="testpass";
5 if (($_POST['user']==$user_check) && ($_POST['pass']==$pass_check)) {
6     $user = $_POST['user'];
7     $pass = $_POST['pass'];
8     $fullname = "Mr.Parinya Noidonprai";
9     session_register ("fullname");
10    session_register ("user");
11    session_register ("pass");
12    header ("location: member.php");
13 } else header ("location: index.php");
14 ?>
```

จากตัวอย่างที่ 10.26 สคริปต์สำหรับตรวจสอบข้อมูลที่ผู้ใช้ Login เข้ามา อธิบายดังนี้

บรรทัดที่ 2 เริ่มใช้งานเชสชัน

บรรทัดที่ 3 กำหนดค่าให้กับตัวแปร \$user_check มีค่าเท่ากับ testuser (ใช้
เปรียบเทียบชื่อผู้ใช้กับส่วนนำเข้าข้อมูลที่ผู้ใช้งานป้อนผ่าน user)

บรรทัดที่ 4 กำหนดค่าให้กับตัวแปร \$pass_check มีค่าเท่ากับ testpass (ใช้
เปรียบเทียบรหัสผ่านกับส่วนนำเข้าข้อมูลที่ผู้ใช้งานป้อนผ่าน pass)

บรรทัดที่ 5 ใช้โครงสร้างเงื่อนไข if ตรวจสอบส่วนนำเข้าที่รับเข้ามากับค่าที่กำหนดไว้
ตรงกันหรือไม่ ถ้าชื่อผู้ใช้และรหัสผ่านเหมือนกับค่าที่กำหนดไว้ในตัวแกร บรรทัดที่ 3 และ 4 ไปบรรทัดที่
6 ถ้าไม่ใช่ไปทำคำสั่งในบรรทัดที่ 13

บรรทัดที่ 6 กำหนดให้ตัวแปร \$user มีค่าเท่ากับส่วนนำเข้าข้อมูล \$_POST['user']

บรรทัดที่ 7 กำหนดให้ตัวแปร \$pass มีค่าเท่ากับส่วนนำเข้าข้อมูล \$_POST['pass']

บรรทัดที่ 8 กำหนดตัวแปรอื่นๆ ตามผู้ใช้จะใช้งานเชสชัน จากตัวอย่างกำหนดให้ตัว
แปร \$fullname มีค่าเท่ากับ "Mr.Parinya Noidonprai"

บรรทัดที่ 9 ลงทะเบียนตัวแปร fullname ให้เป็นตัวแปรเชสชัน

บรรทัดที่ 10 ลงทะเบียนตัวแปร user ให้เป็นตัวแปรเชสชัน

บรรทัดที่ 11 ลงทะเบียนตัวแปร pass ให้เป็นตัวแปรเชสชัน

บรรทัดที่ 12 ใช้ฟังก์ชัน header () เพื่อย้ายหน้าเว็บเพจไปยัง member.php

บรรทัดที่ 13 โครงสร้างเงื่อนไข else (กรณีที่ชื่อผู้ใช้บรรหัสผ่านที่ป้อนผ่านส่วนนำเข้าข้อมูลไม่ตรงกับค่าตัวแปรที่กำหนดไว้) ใช้ฟังก์ชัน header () เพื่อกลับไปยังเว็บเพจ index.php (กลับไป Login ใหม่)

10.5.3 กำหนดหน้าเว็บเพจที่จะประมวลผลต่อ เมื่อตรวจสอบชื่อผู้ใช้และรหัสผ่านถูกต้องให้ทำงานต่อที่เว็บเพจ member.php มีสคริปต์คำสั่ง ดังนี้

ตัวอย่างที่ 10.27 สคริปต์ตรวจสอบการลงทะเบียนตัวแพรเซสชัน (member.php)

```

1 <?php
2 session_start ();
3 if (isset ($_SESSION["user"]) && isset ($_SESSION["pass"]))
4     echo "Congratulations, You are login with: ". $_SESSION["fullname"];
5 else header ("location: index.php");
6 ?>

```

จากภาพที่ 10.27 สคริปต์ตรวจสอบการลงทะเบียนตัวแพรเซสชัน อธิบายดังนี้

บรรทัดที่ 2 เริ่มต้นใช้งานเซสชัน

บรรทัดที่ 3 ตรวจสอบการลงทะเบียนตัวแพรเซสชัน \$_SESSION["user"] และ \$_SESSION["pass"] หากไม่มีการลงทะเบียนตัวแพรดังกล่าวแสดงว่าเข้าสู่สคริปต์นี้อย่างไม่ถูกต้อง ไปทำงานต่อบรรทัดที่ 5 กรณีตรวจสอบแล้วพบการลงทะเบียนตัวแพรเซสชันที่ระบุ ไปบรรทัดที่ 4

บรรทัดที่ 4 แสดงความยินดีต้อนรับ หรือผู้พัฒนาระบบสามารถประยุกต์ใช้ฟังก์ชัน header () เพื่อย้ายหน้าเว็บเพจไปยังหน้าเว็บเพจของระบบสมาชิก หรือเว็บเพจอื่นๆ ตามต้องการ

บรรทัดที่ 5 กรณีที่พบตัวแพรเซสชันที่ลงทะเบียนกลับไป Login ใหม่ที่ index.php

สรุป

ลักษณะของเว็บแอพพลิเคชันที่พัฒนาเป็นระบบงานนั้น จำเป็นต้องแบ่งเว็บเพจออกเป็นส่วนๆ แยกตามบทบาทหน้าที่การทำงาน เช่น ตรวจสอบสิทธิผู้ใช้งานระบบ จัดการข้อมูลสมาชิก สืบค้นข้อมูลรายงาน แก้ไข และลบข้อมูล เป็นต้น และเมื่อเริ่มทำงานติดต่อกับผู้ใช้งานระบบ ทุกเว็บเพจภายในเว็บไซต์จำเป็นต้องทำงานประสานกันเป็นระบบ ดังนั้นจึงจำเป็นต้องเรียนรู้วิธีการควบคุมหน้าเว็บเพจ เช่น ย้ายไปยังเว็บเพจที่ทำหน้าที่ในเรื่องนั้นๆ ตามหน้าที่ที่ผู้พัฒนาได้ออกแบบไว้ เป็นต้น ปัญหาที่สำคัญเมื่อมีการย้ายหรือปรับเปลี่ยนเว็บเพจ คือ ตัวแปรที่ทำงานอยู่ภายในเว็บเพจนั้นๆ จะถูกยกเลิกการใช้งานโดยปริยายไม่สามารถนำไปใช้ในเว็บเพจอื่นๆ ได้ นี่คือสาเหตุที่ต้องเรียนรู้วิธีการส่งค่าตัวแปรในรูปแบบที่ได้กล่าวถึงในบทนี้



คำถามท้ายบท

1. จงอธิบายหน้าที่ของฟังก์ชัน header () พร้อมยกตัวอย่างประกอบ
2. จงอธิบายหลักการส่งข้อมูลระหว่างเว็บเพจแบบ Query String และอธิบายถึงหลักการอ่านข้อมูลที่เว็บเพจปลายทางที่ได้รับข้อมูล Query String
3. จงอธิบายถึงประโยชน์ของคุกเก้และเซสชัน และความแตกต่างระหว่างคุกเก้และเซสชัน
4. จงอธิบายหน้าที่ของฟังก์ชัน session_start () และฟังก์ชัน session_destroy () พร้อมยกตัวอย่างสคริปต์ประกอบการอธิบาย
5. จงอธิบายหลักการทำงานของสคริปต์ด้านล่างตั้งแต่บรรทัดที่ 1 ถึงบรรทัดสุดท้าย และบอกผลลัพธ์ที่ได้จากการสคริปต์

```
1      <?php  
2          session_start ( );  
3          $the_best = "Computer Science of SRU";  
4          session_register ("the_best");  
5          echo $_SESSION["the_best"];  
6      ?>
```

บทที่ 11

การเขียนโปรแกรมเชิงวัตถุ

ในบทนี้จะกล่าวถึงการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ซึ่งแม้ว่าในความเป็นจริงแล้ว ไม่จำเป็นต้องมีความรู้เกี่ยวกับเชิงวัตถุเลย ก็สามารถสร้างเว็บด้วยภาษา PHP ได้ แต่ในปัจจุบันมักจะมีผู้สร้างส่วนที่มายा�ຍขึ้นความสามารถของภาษา PHP (หรือ Extension) ใหม่ๆ เพิ่มขึ้นมาเรื่อยๆ โดย Extension เหล่านี้ มักถูกสร้างในรูปแบบของคลาส ดังนั้นจึงควรมีพื้นฐานความรู้เกี่ยวกับการเขียนโปรแกรมเชิงวัตถุเพื่อจะนำไปใช้กับการเขียนโปรแกรมในบางเรื่องได้

11.1 คลาส

โดยปกติแล้วภาษา PHP จะเป็นภาษาแบบ Procedural โดยไม่ได้อยู่ภายใต้ข้อกำหนด หรือโครงสร้างแบบเชิงวัตถุ จะเห็นได้ว่าภาษา PHP นั้นประกอบด้วยฟังก์ชันต่างๆ เป็นจำนวนมาก และก็สามารถใช้ฟังก์ชันเหล่านี้ได้โดยตรง และตั้งแต่ PHP4 เป็นต้นมา ได้พัฒนาให้ภาษา PHP รองรับการเขียนโปรแกรมทางด้านเชิงวัตถุเพิ่มเข้ามา โดยรูปแบบของการเขียนโปรแกรมเชิงวัตถุนั้นจะช่วยให้สามารถรวมการกระทำที่เกี่ยวข้องกับเรื่องใดเรื่องหนึ่งให้มารวมเป็นหน่วยเดียวกัน หรือที่เรียกว่า "ออบเจกต์" นั่นเอง โดยภายในตัวออบเจกต์นั้นก็จะมีองค์ประกอบหลายอย่าง เช่น เมธอด (Method) ฟิลด์ (Field) พร้อมเพอร์ตี้ (Property) เป็นต้น มักจะเรียกสิ่งที่อยู่ในออบเจกต์ว่า "สมาชิกของออบเจกต์" และการเรียกใช้งานสมาชิกเหล่านี้ก็จะต้องทำผ่านออบเจกต์เท่านั้น ดังนั้นการเขียนโปรแกรมเชิงวัตถุเป็นวิธีการเรียกใช้การทำงานของออบเจกต์ จะเรียกว่า "คลาส (Class)" ดังนั้นจากที่ได้กล่าวมาทั้งหมดก็พอสรุปได้ว่า คลาส คือ โครงสร้างที่ใช้ในการกำหนดองค์ประกอบ และวิธีการทำงานของออบเจกต์

ก่อนที่จะเรียนรู้เกี่ยวกับออบเจกต์ และองค์ประกอบภายในออบเจกต์ ต้องเริ่มศึกษาจากรูปแบบของคลาสก่อน โดยรูปแบบทั่วไปของคลาส มีดังนี้

รูปแบบ

```
Class ชื่อคลาส {  
    ....  
    ....  
}
```

โดยทั่วไปนิยมตั้งชื่อคลาสเป็นชื่อคลาสที่ต้นด้วยอักษรตัวพิมพ์ใหญ่ (ไว้เป็นเพียงจุดสังเกตเท่านั้น เพราะจริงๆแล้วไม่จำเป็น) ส่วนข้อกำหนดอื่นๆ ในการตั้งชื่อจะคล้ายกับการตั้งชื่อตัวแปร เช่น

```
Class MyClass {  
    ....  
    ....  
}
```

11.2 การนำคลาสมาใช้ในสคริปต์ PHP

เนื่องจากภาษา PHP ไม่ได้มีโครงสร้างแบบเชิงวัตถุอย่างแท้จริง การนำคลาสที่สร้างขึ้นมาใช้ก็จะใช้รูปแบบเดียวกับการนำเข้าไฟล์โดยทั่วๆ ไป มีแนวทางดังนี้

11.2.1 สร้างเอกสารเว็บเพจ ชนิด PHP

11.2.2 เขียนคลาสที่จะสร้างลงไปเมื่อมีการเขียนสคริปต์ PHP ปกติ เช่น

```
<?php  
Class PHPMySQL {  
    ....  
    ....  
}  
?>
```

11.2.3 การบันทึกให้กำหนดชื่อเหมือนกับไฟล์เว็บเพจทั่วไป แต่ให้มีส่วนขยายเป็น .php เช่น MyClass.php หรือบางคนอาจต้องการให้ต่างจากไฟล์เว็บเพจทั่วไป อาจกำหนดชื่อในรูปแบบ ชื่อไฟล์.class.php เช่น สมมติคลาสชื่อ "PHPMySQL" ก็บันทึกไฟล์นี้ด้วยชื่อ "PHPMySQL.class.php" เป็นต้น

เมื่อจะนำคลาสนี้ไปใช้ในเว็บเพจได้ ก็ใช้ฟังก์ชัน include () หรือ require () เพื่อนำคลาสเข้าไปใช้งานร่วมกับสคริปต์อื่นๆ

```
<?php  
include ("PHPMySQL.class.php");  
....  
....  
?>
```

11.3 ออกแบบ และอินสแตนซ์

คลาสที่สร้างขึ้นเป็นเพียงข้อกำหนดหรือวิธีในการประมวลผลข้อมูลเท่านั้น แต่ยังไม่สามารถนำคลาสไปใช้งานได้ จนกว่าคลาสนั้นจะถูกสร้างเป็นօบเจ็คต์เสียก่อน ดังนั้นคลาสกับօบเจ็คต์ก็คือ สิ่งเดียวกันเพียงแต่อยู่ในสถานภาพ ด้วยเหตุนี้จึงเรียกօบเจ็คต์ว่าเป็นอินสแตนซ์ (Instance) ของคลาส หรือสิ่งที่ใช้แทนคลาสนั้นเอง การสร้างօบเจ็คต์มีรูปแบบดังนี้

รูปแบบ

```
$ชื่อของอินสแตนซ์ = new ชื่อคลาส ( );
```

ตัวอย่างที่ 11.1 การสร้างออบเจกต์

```
$mycls = new MyClass ( );
```

ขั้นตอนในการสร้างออบเจกต์ของคลาสนี้เรียกว่า Instantiate หลังจากออบเจกต์ถูกสร้างขึ้นแล้ว ก็จะมีความสามารถตามที่ระบุไว้ในคลาสทุกประการโดยคลาสแต่ละคลาสมีความสามารถนำไปสร้างเป็น ออบเจกต์ได้มากกว่า 1 ออบเจกต์ มีตัวอย่างดังนี้

ตัวอย่างที่ 11.2 การสร้างเป็นออบเจกต์ได้มากกว่า 1 ออบเจกต์

```
<?php
    $myclass1 = new MyClass ( );
    $myclass2 = new MyClass ( );
?>
```

11.4 เมธอด

เมธอด (Method) เป็นองค์ประกอบของคลาஸสำหรับการกระทำอย่างใดอย่างหนึ่ง โดยทั่วไปแล้ว คลาสจะต้องประกอบด้วยเมธอดอย่างน้อย 1 เมธอดเสมอ รายละเอียดเกี่ยวเมธอดมีดังนี้

11.4.1 การสร้างเมธอด

ความจริงแล้วเมธอดก็คือ พังก์ชันของคลาสนั่นเอง ดังนั้นรูปแบบ และวิธีการสร้างเมธอด ก็เหมือนกันกับการสร้างพังก์ชันธรรมดาทั่วไป แต่เมธอดมีลักษณะ ของการสร้างเมธอด ตัวอย่างดังนี้

ตัวอย่างที่ 11.3 ลักษณะของการสร้างเมธอด

```
1 <?php
2     class Circle {
3         function area ($radius) return pi ( ) * $radius * $radius;
4         function perimeter ($radius) return 2 * pi ( ) * $radius;
5     }
6 ?>
```

11.4.2 การเรียกใช้เมธอด

การเรียกใช้เมธอดจะใช้โ坡เรเตอร์ -> ในกรอบเมธอดที่ต้องการเรียกใช้งาน โดย หากเป็นการเรียกใช้ภายในคลาสเดียวกัน (เรียกใช้เมธอดหนึ่งจากอีกเมธอดหนึ่ง) ก็ใช้รูปแบบดังนี้



`$this -> เมธอด (อาร์กิวเม้นต์)`

เช่น สมมติว่าต้องการเรียกใช้เมธอด `area ()` ภายในคลาส `Circle` ก็ทำดังนี้

ตัวอย่างที่ 11.4 ตัวอย่างการเรียกใช้เมธอด

`$a = $this -> area (10);`

หากเป็นการเรียกจากนอกคลาส ก็ต้องเรียกผ่านอินสแตนซ์ในรูปแบบดังนี้

`$อินสแตนซ์ -> เมธอด (อาร์กิวเม้นต์)`

เช่น สมมติว่าต้องการเรียกใช้เมธอด `area ()` ของคลาส `Circle`

ตัวอย่างที่ 11.5 ตัวอย่างการเรียกใช้เมธอด `area ()` ของคลาส `Circle`

```
1 <?php  
2     $cir = new Circle ();  
3     echo $cir -> area(10);  
4 >
```

จากตัวอย่างที่ 11.5 ตัวอย่างการเรียกใช้เมธอด `area ()` ของคลาส `Circle` เป็นการทดสอบการสร้าง และการใช้คลาส โดยเป็นคลาสง่ายๆ เกี่ยวกับการคำนวณเลข ทั้งนี้ต้องสร้างขึ้นมา 2 เว็บเพจ โดยเว็บเพจแรกเป็นคลาส ส่วนอีกเว็บเพจสำหรับทดสอบคลาส ดังนี้

ตัวอย่างที่ 11.6 ตัวอย่าง เป็นการทดสอบการสร้าง และการใช้คลาส โดยเป็นคลาสง่ายๆ ชื่อไฟล์ `calculator.class.php`

```
1<?php  
2     class Calculator {  
3         function showResult ($num1, $num2, $op) {  
4             if (!is_numeric ($num1) || !is_numeric ($num2)) {  
5                 echo "ตัวแปรที่รับมาไม่ใช่ตัวเลขไม่สามารถคำนวณได้ <br>";  
6             } else if ( !in_array ($op, array ("+", "-", "*", "/")) {  
7                 echo "เครื่องหมายดำเนินการไม่ถูกต้อง <br>";  
8             } else {  
9                 echo "$num1 $op $num2 = ";  
10                eval ("echo $num1 $op $num2;");  
11                // พังก์ชัน eval () ใช้ประมวลผล คำสั่ง PHP ที่เขียนในแบบสตริง  
12            }  
13        }
```

```
14    }
15?>
```

ข้อไฟล์ที่จะเรียกใช้งานคลาส index.php

```
1 <?php
2     include ("calculator.class.php");
3     $cal = new Calculator ();
4     $cal -> showResult (10, 20, "*");
5 ?>
```

ผลลัพธ์

```
10 * 20 = 200
```

11.4.3 การกำหนดค่าพารามิเตอร์เริ่มต้นโดยปริยาย (Default Parameter) ให้กับเมธอด

เนื่องจากเมธอดนั้นก็คือฟังก์ชัน และฟังก์ชันใน PHP สามารถกำหนดค่าเริ่มต้นโดยปริยายให้ได้ สามารถนำรูปแบบตรงนี้มาใช้เพื่อเป็นทางเลือกให้ผู้ใช้สามารถที่จะกำหนดพารามิเตอร์บางตัวหรือไม่ก็ได้ ทั้งนี้วิธีการต่างๆ ก็เหมือนกับฟังก์ชันที่ได้ศึกษามาแล้ว มีตัวอย่างดังนี้

ตัวอย่างที่ 11.7 การกำหนดค่าพารามิเตอร์เริ่มต้นโดยปริยายให้กับเมธอด

```
1 <?php
2     class Payment {
3         function total ($quantity, $price, $vat=7) {
4             return $quantity * $price * (1+$vat/100);
5         }
6     }
7 ?>
```

จากคลาส Payment เมื่อเรียกใช้เมธอด total () จะกำหนด %VAT หรือไม่ก็ได้ โดยหากไม่กำหนดก็จะใช้ค่าเริ่มต้นโดยปริยาย คือ 7% แต่หากต้องการใช้ค่าอื่นก็กำหนดลงไปโดยตรง มีตัวอย่างดังนี้

ตัวอย่างที่ 11.8 การเรียกใช้คลาส Payment

```
1 <?php
2     $p = new Payment();
3     $t1 = $p -> total (10, 20); //VAT 7%
```



```

4      $t2 = $p -> total (30, 40, 10);           //VAT 10%
5      ?>

```

11.5 โมดิฟายเออร์ แบบ public และ private

โมดิฟายเออร์ (Modifier) เป็นการควบคุมการเข้าใช้งานสมาชิกของคลาส เช่น เมื่อตัด และ พร้อมเพร็งต์ทั้งนี้เนื่องจากตามข้อกำหนดของการเขียนโปรแกรมเชิงวัตถุนั้น สามารถที่จะเรียกใช้สมาชิกของคลาสนั่งจากภายนอกคลาสได้ แต่ภายในคลาสอาจมีข้อมูลบางอย่างที่มีความสำคัญ จึงจำเป็นต้องมีวิธีการที่จะจำกัดขอบเขตในการเข้าใช้งานข้อมูลเหล่านี้ โดยใช้คีย์เวิร์ดคลุมหนึ่งเรียกว่า โมดิฟายเออร์ สำหรับภาษา PHP มีโมดิฟายเออร์หลายตัว ในบทนี้จะกล่าวถึงเฉพาะโมดิฟายเออร์บางชนิดที่ควรรู้จักในเบื้องต้น ดังนี้

public ใช้ public เมื่อไม่ต้องการปกป้องสิทธิการเข้าใช้สมาชิกนั้น

private ใช้ private เมื่อไม่ต้องการให้เข้าถึงสมาชิกนั้นจากภายนอกคลาสได้ จึงมักใช้ private กับสมาชิกที่มีความสำคัญ และต้องการสงวนไว้ใช้เฉพาะภายในคลาสเองเท่านั้น

ลักษณะการนำไปใช้กับเมธอด มีด้วยกันดังนี้

ตัวอย่างที่ 11.9 ตัวอย่างการกำหนดใช้โมดิฟายเออร์ให้กับเมธอด

```

1 <?php
2     class Cylinder {
3         private function CircleArea ($radius) {
4             return pi () * $radius * $radius;
5         }
6         public function volume ($radius, $height) {
7             $base_area = $this -> circleArea ($radius);
8             $volume = $base_area * $height;
9             return $volume;
10        }
11    }
12 ?>

```

จากตัวอย่างที่ 11.9 ตัวอย่างการกำหนดใช้โมดิฟายเออร์ให้กับเมธอด อธิบายดังนี้

บรรทัดที่ 3 กำหนดโมดิฟายเออร์ ชนิด private คือ ไม่ต้องการให้เข้าถึงสมาชิกนั้นจากภายนอกคลาส และต้องการสงวนไว้ใช้เฉพาะภายในคลาสเท่านั้น

บรรทัดที่ 6 กำหนดโดยพิฟายเออร์ ชนิด public คือ อนุญาตให้เข้าถึงสมาชิกจากทุกๆ ส่วนของ скриปต์

ตัวอย่างที่ 11.10 ลักษณะผลการเรียกใช้เมธอดแบบ private และ public

```

1 <?php
2 include ("Cylinder.class.php");
3 $cyl = new Cylinder ( );
4 echo $cyl -> circleArea (10);
5 echo $cyl -> volume (10, 20);
6 ?>

```

จากตัวอย่างที่ 11.10 แสดงลักษณะผลการเรียกใช้เมธอดแบบ private และ public อธิบายเพิ่มเติม ดังนี้

บรรทัดที่ 4 ไม่สามารถเรียกใช้งานได้ เนื่องจาก circleArea() เป็น private method

บรรทัดที่ 5 สามารถใช้งานได้ เพราะ volume () เป็น public method

11.6 พร็อปเพอร์ตี้ และฟิลด์

พร็อปเพอร์ตี้ (Property) คือ ข้อมูลประจำตัวของคลาส ส่วนใหญ่แล้วมักเป็นข้อมูลที่ต้องใช้ร่วมกันในหลายๆ เมธอด หรือไม่ก็เป็นข้อมูลเบื้องต้นที่จำเป็นต่อการทำงานของคลาส เช่น หากเป็นวงกลมข้อมูลที่จำเป็นต้องใช้เสมอ คือ รัศมี เป็นต้น ถึงแม้ว่าข้อมูลนี้จะรับผ่านพารามิเตอร์ของเมธอดได้แต่ก็จะต้องระบุข้อมูลนี้ทุกๆ ครั้ง ทั้งนี้การสร้างพร็อปเพอร์ตี้มีรูปแบบดังนี้

```
public $ชื่อพร็อปเพอร์ตี้
```

สำหรับโมเดลไฟล์เอกสาร โดยที่จะกำหนดเป็น public เนื่องจากในกรณีของพร็อปเพอร์ตี้นั้น จะต้องสามารถเข้าถึงจากภายนอกคลาสได้ มีตัวอย่างดังนี้

```

class Cylinder {
    public $radius;
    public $height;
    ....
    ....
}

```



หมายเหตุ

การเขียนโปรแกรมเชิงวัตถุสำหรับภาษา PHP ในยุคแรกการกำหนดพรีอปเพอร์ตี้จะใช้คีย์เวิร์ด "var" ตามด้วยชื่อพรีอปเพอร์ตี้ เช่น var \$radius; แต่ในปัจจุบันนิยมกำหนดโดยโมดิฟายเออร์ "public" แทน และหากกำหนดโดยโมดิฟายเออร์เป็น "private" มักเรียกว่า "ฟิลเตอร์"

เนื่องจากพรีอปเพอร์ตี้ คือ ข้อมูลของคลาส ดังนั้นจึงอาจถูกเรียกใช้จากทั้งภายใน และภายนอก คลาส หากเป็นการเรียกภายในคลาสจะต้องระบุผ่านคีย์เวิร์ดพิเศษ \$this และหากเป็นการอ้างอิง ภายนอกคลาสจะต้องระบุผ่านอินสแตนซ์ ตามรูปแบบดังนี้

เรียกใช้ภายในคลาส: \$this -> ชื่อพรีอปเพอร์ตี้

เรียกใช้ภายนอกคลาส: \$อินสแตนซ์ -> ชื่อพรีอปเพอร์ตี้

ตัวอย่างที่ 11.11 การเรียกใช้ภายในคลาสผ่านคีย์เวิร์ดพิเศษ \$this

```

1 <?php
2     class Cylinder {
3         public $radius = 0;
4         public $height = 0;
5         private function CircleArea () {
6             return pi () * $this -> circleArea ($this -> radius);
7         }
8         public function volume () {
9             $base_area = $this -> circleArea($this -> radius);
10            $volume = $base_area * $this -> height;
11            return $volume;
12        }
13    }
14 ?>
```

ตัวอย่างที่ 11.12 การใช้พรีอปเพอร์ตี้จากภายนอกคลาสผ่านอินสแตนซ์

```

1 <?php
2     include ("Cylinder.class.php");
3     $cy1 = new Cylinder ();
4     $cy1 -> redius = 10;
5     $cy2 -> redius = 20;
```

```

6      echo $cy1 -> volume();
7      ?>

```

ฟิลด์ (Field) ตามรูปแบบการเขียนโปรแกรมเชิงวัตถุของภาษา PHP จะมีลักษณะคล้ายกับพาร์สเปอร์ตี้ คือ เป็นข้อมูลของคลาสแต่ละข้อมูลแบบฟิลด์นี้มักจะใช้เฉพาะภายในคลาสเท่านั้น โดยไม่อนุญาตให้เข้าถึงจากนอกคลาสได้ ดังนั้นจึงต้องกำหนดโมดิฟายเออร์เป็น "private" ขณะที่พาร์สเปอร์ตี้นั้นสามารถเข้าถึงจากนอกคลาสได้ โดยที่นำไปแล้วมักใช้ฟิลด์ร่วมกับคอนสตรัคเตอร์เป็นหลัก เพื่อรับข้อมูลที่จำเป็นเบื้องต้นจากนอกคลาสเข้ามา ตัวอย่างลักษณะการกำหนดฟิลด์ดังนี้

```

class Cylinder {
    private $radius = 0;
    private $height = 0;
    ....
}

```

11.7 ค่าคงที่

ค่าคงที่ (Constant) คือ ค่าที่กำหนดไว้อย่างตายตัว เหมือนกับค่าคงที่ที่ใช้ในการเขียนสคริปต์ปกตินั่นเอง โดยรูปแบบการกำหนดดังนี้

```
const ชื่อค่าคงที่ = ค่าที่กำหนด;
```

ค่าคงที่นี้ ไม่ต้องกำหนดโมดิฟายเออร์ใดๆ และชื่อค่าคงที่ก็ไม่มีเครื่องหมาย \$ นำหน้าและวิธีการเรียกใช้ค่าคงที่ก็จะต่างไปจากการเรียกใช้พาร์สเปอร์ตี้ โดยสามารถเรียกผ่านชื่อคลาสได้โดยตรงโดยไม่ต้องเรียกผ่านอินสแตนซ์ของคลาสมีรูปแบบดังนี้

```
ชื่อคลาส :: ชื่อค่าคงที่
```

ตัวอย่างที่ 11.13 การกำหนดค่าคงที่ภายในคลาส

```

1 <?php
2     class Circle {
3         const PI = 3.14;
4         public function area($radius) {
5             return Circle :: PI * $radius * $radius;
6         }
7     }
8 ?>

```



11.8 คอนสตรัคเตอร์

คลาสบางคลาสก่อนที่จะทำการประมวลผลใดๆ ได้ จะเป็นต้องอาศัยข้อมูลพื้นฐานบางอย่าง เสียก่อน หากขาดข้อมูลเหล่านี้ไปแล้ว คลาสจะทำงานไม่ได้เลย เช่น คลาสว่างกลม หากขาดข้อมูลที่เป็นค่ารัศมี ก็ไม่อาจคำนวนหาค่าพื้นที่ และเส้นรอบวงได้ และผู้ใช้คลาสก็อาจไม่ทราบว่าต้องกำหนดข้อมูลพื้นฐานอะไรให้แก่คลาสนั้นบ้าง ด้วยเหตุนี้คลาสจึงต้องมีเมธอดพิเศษที่เรียกว่าคอนสตรัคเตอร์ (Constructor) เพื่อใช้ในการกำหนดข้อมูลที่จำเป็นเริ่มแรกให้แก่คลาส โดยลักษณะที่สำคัญของคอนสตรัคเตอร์ ดังนี้

11.8.1 คอนสตรัคเตอร์เป็นเมธอดพิเศษที่ใช้ในการรับข้อมูลเริ่มต้นเข้ามาบังคลาสในรูปแบบพารามิเตอร์ แต่คอนสตรัคเตอร์ไม่จำเป็นต้องมีพารามิเตอร์เสมอไป

11.8.2 เมธอดที่เป็นคอนสตรัคเตอร์จะถูกเรียกขึ้นมาทำงานโดยอัตโนมัติในขั้นตอนการสร้างออบเจกต์

11.8.3 สำหรับ PHP คอนสตรัคเตอร์จะถูกกำหนดตามตัวไปโดยว่าต้องใช้ชื่อ `__construct()` เท่านั้น (หน้า `construct` มีเครื่องหมาย underscore 2 จุด) ส่วนโมเดลไฟล์เอกสารไม่จำเป็นต้องกำหนด หรือกำหนดเป็น `public` เท่านั้น

11.8.4 พารามิเตอร์ที่คอนสตรัคเตอร์รับเข้ามาจะต้องนำไปเก็บไว้ในตัวแปรฟิลด์ เพื่อรอการนำไปใช้งานเมธอดอื่นๆ ต่อไป

ตัวอย่างที่ 11.14 การสร้างคอนสตรัคเตอร์ของคลาส

```

1 <?php
2     class Circle{
3         private $radius = 0;
4         function __construct($radius) {
5             $this -> radius = $radius;      //นำค่าพารามิเตอร์ไปเก็บไว้ในตัวแปรฟิลด์
6         }
7     }
8 ?>
```

11.8.5 การเรียกใช้คอนสตรัคเตอร์ เมธอดที่เป็นคอนสตรัคเตอร์จะถูกเรียกขึ้นมาทำงานโดยอัตโนมัติในขั้นตอนการสร้างออบเจกต์ ทั้งนี้หากคอนสตรัคเตอร์นั้นมีพารามิเตอร์ ต้องระบุลงไปด้วย เช่น

```
$cir = new Circle(10);           //กำหนดรัศมีเท่ากับ 10
```

11.9 โมดิฟายเออร์ Static

โมดิฟายเออร์แบบ static ใช้เมื่อต้องการให้เกิดการอ้างอิงข้อมูลเดียวกันของทุกออบเจกต์ที่สร้างมาจากการคลาสเดียวกัน ทำให้เมื่อเปลี่ยนแปลงข้อมูลที่ออบเจกต์ตัวใด จะส่งผลให้ข้อมูลอันเดียวกันในออบเจกต์อื่นๆ เปลี่ยนตามไปด้วย เพราะเป็นการอ้างอิงค่าเดียวกัน ทั้งนี้ static ใช้ได้กับทั้งพร็อปเพอร์ตี้ และเมธอด โดยแต่ละคลาสมีสมาชิกที่เป็นแบบ static มีดังต่อไปนี้

```
โมดิฟายเออร์ static $พร็อปเพอร์ตี้;
โมดิฟายเออร์ static function เมธอด () {
    .
    .
}

}
```

ตัวอย่างที่ 11.15 โมดิฟายเออร์ Static

```
private static $field;
public static $property;

public static function method () {
    .
    .
}
```

11.9.1 การเรียกใช้งานสมาชิกแบบ static ภายในคลาส

ภายในคลาสมีเมื่อต้องการเรียกใช้สมาชิกแบบ static จะต้องระบุด้วยคำว่า self ตามด้วย โอเปอเรเตอร์ :: และชื่อพร็อปเพอร์ตี้หรือเมธอด ตามรูปแบบคือ

self :: \$พร็อปเพอร์ตี้
หรือ
self :: เมธอด ()

ตัวอย่างที่ 11.16 การเรียกใช้งานสมาชิกแบบ static ภายในคลาส

self :: \$x	//พร็อปเพอร์ตี้
self :: f1 ();	//เมธอด

11.9.2 การเรียกใช้งานสมาชิกแบบ static จากนอกคลาส

ส่วนภายนอกคลาสการระบุถึงสมาชิกที่เป็น static จะระบุผ่านชื่อคลาสโดยตรง หรือระบุผ่านอินสแตนซ์ก็ได้ ในรูปแบบดังนี้



คลาส :: เมธอด

คลาส :: พร็อปเพอร์ตี้

หรือเรียกใช้ผ่านอินสแตนซ์

อินสแตนซ์ -> เมธอด

อินสแตนซ์ -> พร็อปเพอร์ตี้

นอกจากนี้ในการเรียกใช้สมาชิกแบบ static มีข้อกำหนดที่สำคัญดังนี้

1) สามารถเรียกใช้สมาชิกแบบ static ได้โดยตรง โดยไม่จำเป็นต้องเรียกผ่านอินสแตนซ์

2) เฉพาะเมธอดแบบ static เท่านั้นที่สามารถเรียกพร็อปเพอร์ตี้แบบ static ได้

ตัวอย่างการทดสอบการกำหนดสมาชิกแบบ static โดยประกอบด้วย 2 เว็บเพจ สำหรับคลาส และทดสอบการใช้คลาส

ตัวอย่างที่ 11.17 การกำหนดคลาสภายนอกในไฟล์ Visitor.class.php

```
1 <?php
2     class Visitor {
3         private static $num_visitor = 0;
4         function __construct () {
5             self::$num_visitor++;
6         }
7         public static function getVisitor () {
8             return self::$num_visitor;
9         }
10    }
11 ?>
```

ตัวอย่างที่ 11.18 การเรียกใช้คลาสในไฟล์ index.php

```
1 <?php
2     include ("Visitor.class.php");
3     $v1 = new Visitor();
4     echo "Visitors :" . $v1 -> getVisitor () . "<br>";
5     $v2 = new Visitor ();
6     echo "Visitors :" . $v2 -> getVisitor () . "<br>";
7     $v3 = new Visitor ();
8     echo "Visitors :" . $v3 -> getVisitor () . "<br>";
```

9 ?>

ตัวอย่างนี้สร้างออบเจกต์ของคลาส Visitor จำนวน 3 ครั้ง โดยที่คอนสตรักเตอร์ได้เพิ่มค่าให้แก่พรีอปเพอร์ตี้ \$num_visitor เป็นแบบ static ไปอีก 1 และเมื่อเรียกเมธอด getVisitor() เมธอดนี้จะไปอ่านข้อมูลจาก \$num_visitor จะพบว่าค่าจะเพิ่มขึ้นเรื่อยๆ แสดงว่าพรีอปเพอร์ตี้ \$num_visitor มีการอ้างอิงข้อมูลเดียวกัน จึงทำให้ค่าเพิ่มขึ้นจากค่าเดิมได้ไม่ เช่นนั้นแล้วค่าจะต้องเป็น 1 ทั้งหมด

11.10 การสืบทอด

การสืบทอด (Inheritance) หมายถึง การที่คลาสหนึ่งถูกสร้างขึ้นโดยการสืบทอดมาจากอีกคลาสหนึ่ง ผลจากการสืบทอดจะทำให้คลาสที่เป็นผู้สืบทอดมีองค์ประกอบของตัวเองเพิ่มมากับคลาสต้นแบบ เรียกคลาสที่เป็นคลาสต้นแบบว่า Parent Class ส่วนคลาสที่เป็นผู้ทำการสืบทอดมาเรียกว่า Sub Class ทั้งนี้การสืบทอดจะใช้คีย์เวิร์ด extends เป็นตัวกำหนด เช่น

```
class Shape {
    ...
    ...
}

} class Rectangle extends Shape {           //Rectangle สืบทอดมาจาก Shape
    ...
    ...
}
```

การสืบทอดมักทำเพื่อความสามารถบางอย่างเข้าไปใน Sub Class โดยนำความสามารถเดิมบางอย่างของ Parent Class มาใช้ เนื่องจากภายหลังการสืบทอดจะทำให้สามารถเรียกสมาชิกต่างๆ ของ Parent Class จาก Sub Class ได้เหมือนกับคลาสเดียวกัน ยกเว้นสมาชิกที่มีโมดิฟายเออร์เป็น private มีตัวอย่างดังนี้

ตัวอย่างที่ 11.19 การสืบทอดคลาส

```
1 <?php
2     class circle {
3         function circle_area ($radius) {
4             return pi() * pow($radius, 2);
5         }
6     }
7     class Cylinder extends Circle {
8         function volume ($radius, $height) {
```



```

9          $base_area = $this -> circle_area ($radius); //ปริมาตร=พื้นที่ฐาน*สูง
10         return $base_area * $height;
11     }
12 }
13 $cyl = new Cylinder ();
14 echo $cyl -> volume (10, 20);
15 echo $cyl -> circle_area (30);
16 ?>

```

จากโค้ดตัวอย่างนี้จะพบว่าคลาส Cylinder นั้นสืบทอดมาจาก Circle และได้เพิ่มเมธอด volume () เข้ามาในคลาส Cylinder จึงสามารถเข้าถึงเมธอด circle_area () ของคลาส Circle จากคลาส Cylinder ได้โดยใช้ \$this -> เมื่อกับอยู่ในคลาสเดียวกัน รวมถึงการเข้าถึงผ่านทางอินสแตนซ์ได้เช่นกัน

แต่ที่สำคัญคือคลาสมักต้องมีคอนสตรัคเตอร์ และกรณีการสืบทอด Sub Class สามารถมีคอนสตรัคเตอร์เป็นของตนเองที่ต่างไปจาก Parent Class ได้ และสามารถเข้าถึงคอนสตรัคเตอร์ของ Parent Class ด้วยการอ้างอิงในรูปแบบดังนี้

```
parent :: __construct (อาร์กิวเมนต์)
```

จากการสืบทอดระหว่างคลาส Circle/Cylinder ที่ผ่านมา สามารถแก้ไขให้ถูกต้องตามแนวทาง การเขียนโปรแกรมเชิงวัตถุมากยิ่งขึ้นได้ดังนี้

ตัวอย่างที่ 11.20 การสืบทอดระหว่างคลาส

```

<?php
class Circle {
    private $radius = 0;
    function __construct($radius) {
        $this -> radius = $radius;
    }
    function area () {
        return pi () * pow ($this -> radius, 2);
    }
}
class Cylinder extends Circle {
    private $height = 0;
}

```

```

function __construct ($radius, $height) {
    $this -> height = $height;
    private :: __construct($radius); //เรียกค่าอนสตรัคเตอร์ของคลาส Circle
}

function volume () {
    $base_area = $this -> area ();
    return $base_area * $this -> height;
}

}

//-----
$cylinder = new Cylinder (10, 20);
echo $cylinder -> volume();
?>

```

สรุป

หลักการของการเขียนโปรแกรมเชิงวัตถุ เป็นการสร้างความสัมพันธ์ระหว่างฟังก์ชันการทำงานที่อยู่ภายในโปรแกรมให้อยู่ในรูปของวัตถุ โดยที่แต่ละวัตถุ จะประกอบไปด้วยการกระทำที่เรียกว่าเมธอด และคุณสมบัติของวัตถุเรียกว่าพาร์มาเพอร์ต์ ที่มีลักษณะคล้ายกันจะถูกจัดเป็นกลุ่มที่เรียกว่าคลาส การเขียนโปรแกรมเชิงวัตถุ อาจไม่เหมาะสมหรับผู้ที่เริ่มศึกษาเนื่องจากมีความรายละเอียดของแต่ละส่วนค่อนข้างซับซ้อน แต่หากมีความเข้าใจแล้วจะเห็นได้ว่าการเขียนโปรแกรมเชิงวัตถุจะทำให้การพัฒนาระบบทำได้รวดเร็ว ปรับแก้ไขศรีปต์ได้ง่าย เพราะใช้เทคนิคบริการสืบทอดคุณสมบัติ

คำถามท้ายบท

1. จงอธิบายองค์ประกอบของคลาส พื้นที่ตัวอย่างประกอบ
2. จงอธิบายถึงขอบเขต และอินสแตนซ์ รูปแบบการสร้าง พื้นที่ตัวอย่างประกอบ
3. จงอธิบายถึงเมธอด และความเกี่ยวข้องกับคลาส
4. จงอธิบายถึงโมเดลไฟล์เอกสาร แบบ public และ private มีความเหมือนหรือแตกต่างกันอย่างไร
5. จงอธิบายถึงลักษณะที่สำคัญของค่าอนสตรัคเตอร์





บทที่ 12

การใช้เทคนิค AJAX ร่วมกับ PHP

ปัจจุบันนี้ ลักษณะการทำงานแบบ โคลอนต์ - เชิร์ฟเวอร์ เริ่มถูกนำมาใช้งานอย่างแพร่หลายใน ลักษณะการติดต่อสื่อสารผ่านทางเว็บบราวเซอร์ การทำงานแบบนี้ จะมีการทำงานโดย โคลอนต์ จะร้อง ขอและต้องการข้อมูลบางอย่างจาก เชิร์ฟเวอร์ ดังนั้นการโหลดและการรีเฟรชหน้าจอ เป็นสิ่งที่หลีกเลี่ยง ไม่ได้ จึงเป็นผลให้การทำงานของผู้ โคลอนต์ นี้ทำให้ผู้ใช้ต้องหยุดรอการโหลดและการรีเฟรชหน้าจอ ถือว่าเป็นการทำงานที่ไม่มีประสิทธิภาพ

12.1 AJAX

AJAX (Asynchronous JavaScript and XML) ไม่ใช่ชื่อของการเขียนโปรแกรมหรือเป็นชื่อของ ภาษาที่ใช้ในการโปรแกรม แต่เป็นชุดของเทคโนโลยีต่างๆ AJAX ย่อมาจาก Asynchronous JavaScript And XML หมายถึง การทำงานร่วมกันของ JavaScript และ XML แบบ Asynchronous มีหลักการ ทำงาน 2 ประดิ่น คือ การอัพเดท (update) หน้าจอแบบบางส่วน และการติดต่อสื่อสารกับ เชิร์ฟเวอร์ โดยใช้หลักการ Asynchronous ทำให้ผู้ใช้ไม่ต้องหยุดการทำงาน เพื่อรอการประมวลผลจาก เชิร์ฟเวอร์ รวมถึงการโหลดและการรีเฟรชหน้าจอ ของบราวเซอร์ในผู้ โคลอนต์ มีการใช้ AJAX โดยการเพิ่มเลเยอร์ ระหว่างเว็บบราวเซอร์ของผู้ใช้กับ เชิร์ฟเวอร์ ทำให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องรอให้ โคลอนต์ติด ต่อไปยังเชิร์ฟเวอร์ รวมถึงการโหลดและการรีเฟรชหน้าจอทั้งหมดด้วย ดังนั้นผู้ใช้สามารถใช้งาน แอพ พลิกะชันได้อย่างมีประสิทธิภาพมากขึ้น

AJAX จึงไม่ใช้เทคโนโลยีในตัวของมันเอง แต่ว่าเป็นการนำเทคโนโลยีหลายๆ ตัวมารวมกันเข้าด้วยกัน เช่น JavaScript, DHTML, XML, CSS, DOM และ XMLHttpRequest

AJAX Engine ทำหน้าที่เป็นตัวกลางระหว่าง โคลอนต์และเชิร์ฟเวอร์ ฉะนั้นเมื่อ โคลอนต์ มีการ ร้องขอแทนที่จะส่ง HTTP request ไปยังเชิร์ฟเวอร์โดยตรง โคลอนต์จะส่ง JavaScript call ไปยัง AJAX Engine เพื่อโหลดข้อมูลที่ผู้ใช้ต้องการ และหาก AJAX Engine ต้องการข้อมูลเพิ่มเติมในการตอบสนอง ต่อ AJAX Engine ของผู้ใช้จะส่งการร้องขอไปยังเชิร์ฟเวอร์ โดยใช้ XML เทคโนโลยีต่างๆ ที่เป็น ส่วนประกอบของ AJAX ดังนี้

- 12.1.1 HTML/XHTML เป็นภาษาในการจัดแสดงข้อมูล
- 12.1.2 CSS เป็นรูปแบบการจัดแต่ง XHTML
- 12.1.3 Document Object Model (DOM) สำหรับ dynamic display and interaction
- 12.1.4 XML เป็นรูปแบบการแลกเปลี่ยนข้อมูล

- 12.1.5 XSLT สำหรับ แปลง XML เป็น XHTML
- 12.1.6 XMLHttpRequest สำหรับ asynchronous data retrieval
- 12.1.7 JavaScript เป็นภาษาในการใช้งาน AJAX Engine

12.2 ประวัติความเป็นมา

ในช่วงแรกๆ ของการพัฒนา คือ ประมาณปี ค.ศ. 1997 นั้น องค์ประกอบแรกๆ ที่เกิดขึ้นทางฝั่งไคลเอนต์ ถูกเขียนขึ้นโดยทีมพัฒนา Outlook Web Access ต่อมาถูกนำมาใช้เป็นส่วนหนึ่งของ Internet Explorer 5.0 นั่นก็คือจุดเริ่มต้นที่เริ่มรู้จักการทำงานแบบ AJAX และในปี 2005 Google ได้ใช้การติดต่อสื่อสารแบบ Asynchronous เพื่อเป็นรากฐานที่ทำให้รู้จักกับ AJAX กันอย่างแพร่หลาย การทำงานแบบ ไคลเอนต์ - เซิร์ฟเวอร์ ถูกนำมาใช้งานเป็นจำนวนมาก เช่น การติดต่อกับฐานข้อมูลที่เซิร์ฟเวอร์ หรือ การให้บริการทางอินเทอร์เน็ต โดย Google เป็นผู้ลงทุนลงแรงอย่างหนัก ในพัฒนาและการทดสอบ AJAX จึงสังเกตได้ว่า ผลผลิตใหญ่ของ Google คริส อัลแมน และลูซินดา ดิเคส (Chris Ullman and Lucinda Dykes, 2007, p.3) ได้กล่าวไว้ว่า ในช่วงต้นปี 2005 ได้มีการนำ AJAX มาประยุกต์ใช้งานกับเว็บไซต์ต่างๆ ทั่วโลก เช่น Flickr (www.flickr.com), Basecamp (basecamp.com), Amazon CloudSearch (www.A9.com), Google Suggest และ Google Maps และอีกหลายเว็บไซต์ทั่วโลก

12.3 ที่มาของปัญหา

เนื่องจากแอปพลิเคชันที่ใช้งานในปัจจุบันนี้ มีหลักการทำงานแล้วเกิดการสูญเสียเวลาและทรัพยากรของผู้ใช้ในการรอคุยกิจกรรมการทำงานต่างๆ ทำให้ผู้ใช้ต้องหยุดค่อย ดังนั้นการทำงานของผู้ใช้จึงเป็นไปอย่างไม่ต่อเนื่อง หลักการดังกล่าวคือ

12.3.1 Click, wait, and refresh" user interaction paradigm

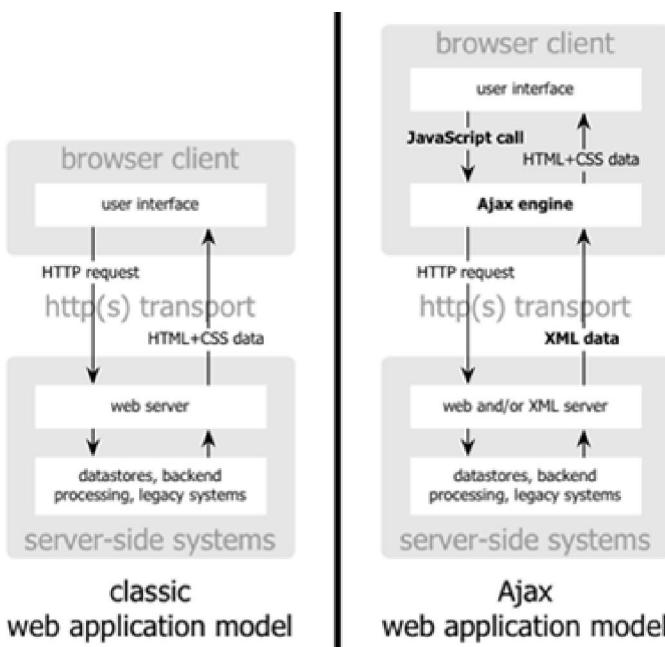
การที่บราวเซอร์ตอบสนองต่อการทำงานของผู้ใช้ โดยจะทิ้งหน้าเว็บที่แสดงอยู่ในขณะนั้น แล้วไปทำการส่ง HTTP request กลับไปยังเซิร์ฟเวอร์แทน ทำให้ผู้ใช้ไม่สามารถทำอะไรได้เลย ในขณะนั้น นอกจากการรอคุยกิจกรรมการทำงานต่างๆ เมื่อเซิร์ฟเวอร์ทำการประมวลผลเสร็จก็จะส่งหน้า HTML กลับมายัง บราวเซอร์ ต่อจากนั้นบราวเซอร์ก็จะรีเฟรชและแสดงหน้า HTML ใหม่ และนี้เองที่ทำให้ผู้ใช้สามารถใช้งานต่อไปได้

จะเห็นว่าผู้ใช้มีช่วงเวลาของการหยุดรอคุยกิจกรรมเป็นเวลานานสำหรับการประมวลผลของเซิร์ฟเวอร์ และการรีเฟรชหน้า HTML ใหม่ทั้งหน้า ซึ่งเป็นสิ่งที่ไม่มีประสิทธิภาพในเชิงดำเนินมิภกิจของการทำงานบนเว็บแอปพลิเคชัน

12.3.2 Synchronous "request/response" communication mode

การที่บราวเซอร์เริ่มทำการร้องขอข้อมูลและเชิร์ฟเวอร์ ก็ต้องสนองเฉพาะการร้องขอที่บราวเซอร์ร้องขอมาเชิร์ฟเวอร์ จะไม่สามารถส่งข้อมูลได้ถ้าบราวเซอร์ไม่ได้ร้องขอข้อมูลในขณะนั้น ถือได้ว่าเป็นการติดต่อสื่อสารเป็นแบบทิศทางเดียว

วัจกรการ Request/Response แบบ Synchronous คือ การทำงานแบบประสานจังหวะระหว่างบราวเซอร์กับเชิร์ฟเวอร์ ทำให้เกิดความล่าช้าในการทำงานทำให้ผู้ใช้ทำอะไรไม่ได้อีกนอกจากการรอการตอบสนองกลับมายังเชิร์ฟเวอร์ เมื่อเชิร์ฟเวอร์ประมวลผลเสร็จ



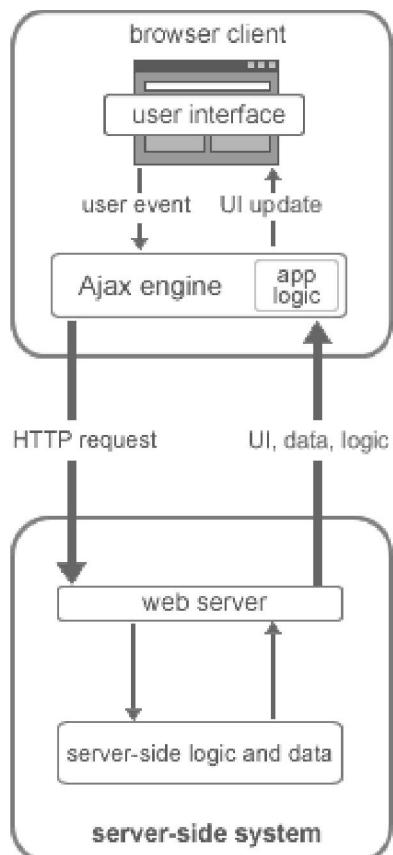
ภาพที่ 12.1 เปรียบเทียบการทำงานแบบเดิม กับ AJAX

ที่มา: เนคเทค. (2556).

12.4 โครงสร้างของ AJAX

มุ่งมองของโครงสร้างของ AJAX ต่างจากเว็บแอปพลิเคชันในทุกวันนี้ เนื่องจากมีการเพิ่ม Engine ทางฝั่งไคลเอนต์





ภาพที่ 12.2 โครงสร้างของ AJAX ในปัจจุบัน

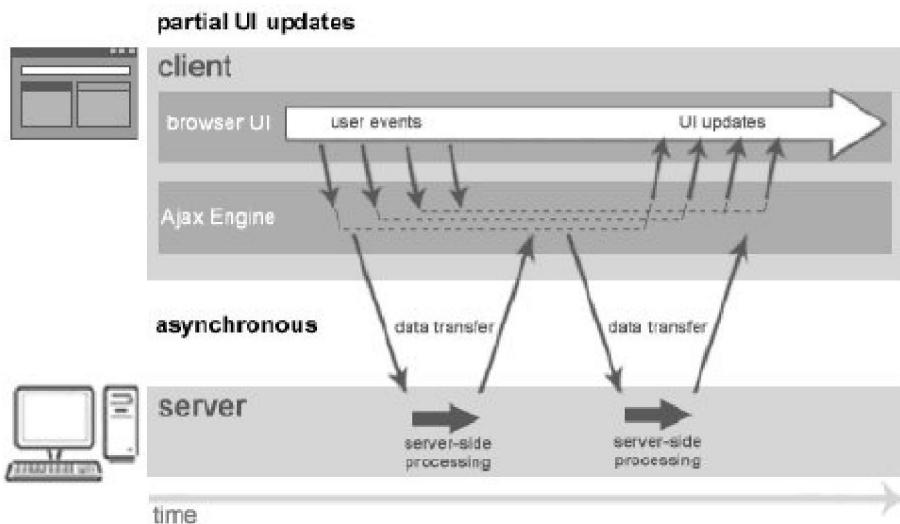
ที่มา: เนคเทค. (2556).

จากภาพ AJAX Engine นี้ อยู่ระหว่างผู้ใช้งานกับเซิร์ฟเวอร์ จะมองว่าเป็นการทำงานที่ “คลาอนต์” การทำงานต่างๆ ของผู้ใช้ โปรแกรมจะไปเรียก AJAX Engine ตัวนี้ขึ้นมา แทนที่การร้องขอหน้าเว็บจากเซิร์ฟเวอร์โดยตรง และจะใช้โครงสร้างข้อมูลแบบ XML ในการխยा�ยข้อมูลระหว่างเซิร์ฟเวอร์ กับ AJAX Engine เมื่อ布拉เวอร์ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์

นอกจากนี้ AJAX Engine ไม่ต้องทำการติดตั้งส่วนเสริมการทำงานต่างๆ และไม่สามารถถูดាដวน์ให้หลดได้ เพราะ AJAX เป็นแนวคิดในการแก้ปัญหาการหยุดชะงักการทำงานของผู้ใช้

12.5 การทำงานของ AJAX

AJAX จะช่วยลดการติดต่อระหว่างคลาอนต์กับเซิร์ฟเวอร์ โดยในการโหลดหน้าเว็บเพจนั้น บราวเซอร์จะโหลดข้อมูลจาก AJAX Engine แทนการร้องขอข้อมูลจากเซิร์ฟเวอร์ โดยตรง ดังนั้น AJAX จะทำหน้าที่ทั้งการแสดงผลรวมทั้งติดต่อกับผู้ใช้ และติดต่อไปยัง เซิร์ฟเวอร์ แล้ว AJAX Engine อนุญาตให้การกระทำต่างๆ ในเว็บแอพพลิเคชันเป็นแบบ Asynchronous คือ ความเป็นอิสระในการติดต่อไปยัง เซิร์ฟเวอร์นั่นเอง ดังนั้นผู้ใช้จะไม่พบกับบริวาร์หน้าขาวๆ อีกต่อไป และไม่ต้องรอการโหลดข้อมูลต่างๆ จากเซิร์ฟเวอร์



ภาพที่ 12.3 หลักการทำงานของ AJAX

ที่มา: เนคเทค. (2556).

12.5.1 Asynchronous

ลักษณะของ Asynchronous ในที่นี้คือ ไม่ต้องหยุดรอข้อมูล หมายความว่า หลังจากส่งข้อมูลออกไปแล้วก็ยังสามารถใช้งานส่วนอื่นๆ ในเว็บเพจนั้นได้ตามปกติ เพราะกระบวนการทั้งหมดเป็นแบบ behind the scene

12.5.2 JavaScript

เนื่องจาก AJAX คือ การเอาเทคโนโลยีอื่นๆ มาใช้ร่วมกัน การที่จะใช้เทคโนโลยีต่างๆ เหล่านี้ได้ จำเป็นต้องใช้ภาษาอย่าง JavaScript มาเป็นตัวจัดการให้ และสำหรับ AJAX ต้องใช้ JavaScript เป็นตัวควบคุมกระบวนการทั้งหมด นับตั้งแต่เริ่มต้นจนสิ้นสุดการติดต่อสื่อสาร ดังนั้นผู้จะศึกษาเทคนิค AJAX นี้ได้จะจำเป็นต้องมีพื้นฐานด้านภาษา JavaScript และ DHTML ที่ดีพอสมควร ยิ่งรู้เกี่ยวกับ DHTML/ JavaScript 多大得多มากเท่าใด ก็ยิ่งมีแนวทางในการประยุกต์ใช้ AJAX มากเท่านั้น

12.5.3 XML

XML (Extensible Markup Language) เป็นภาษาที่ใช้สำหรับการเขียนเอกสาร markup (markup document) โดยที่เอกสาร markup นั้นมีการใช้ metadata หรือ tags เพื่อบอกหน้าที่และประเภทของข้อมูลของส่วนต่างๆ ของเอกสารนั้นได้โดยชัดเจน การเพิ่ม metadata หรือ tags เข้าไปในเอกสารสามารถทำให้โครงสร้างของเอกสารซัดเจนขึ้น และทำให้การประมวลผลเอกสารเป็นไปโดยง่ายและไม่จำเป็นที่จะต้องอาศัยมนุษย์เพื่อตีความเอกสาร

การใช้เทคโนโลยี XML 在 XML การพัฒนามาตรฐานเพื่อการกระจายข่าวเนื่องจาก XML เป็นภาษาที่เหมาะสมกับการแลกเปลี่ยนข้อมูลผ่านเครือข่ายคอมพิวเตอร์ เนื่องจาก XML ไม่ได้ขึ้นอยู่กับโปรแกรมประยุกต์หรือระบบปฏิบัติการใด นอกจากนี้ XML ยังเป็นภาษาที่มีความยืดหยุ่น เนื่องจากผู้ใช้สามารถที่จะกำหนดและตั้งค่า metadata หรือ tags ให้เหมาะสมกับเอกสารเฉพาะที่ต้องการได้อย่างอิสระ

และยังสามารถเพิ่มเติม metadata หรือ tags ได้ในภายหลังโดยไม่มีผลกระทบต่อโปรแกรมที่มีอยู่ก่อนหน้า

XML สำหรับการพัฒนาร่วมกับ AJAX Application แล้วนั้น อาจไม่จำเป็นต้องรู้ XML เลยก็ได้ เพราะ AJAX มีเมธอดที่ใช้แปลงข้อมูลให้อยู่ในรูปแบบสตริงธรรมชาติแบบง่ายๆ ได้

12.5.4 Server Side Script กับ AJAX

ตามหลักการของ AJAX จะเป็นต้องส่งข้อมูลบางอย่างไปยังเซิร์ฟเวอร์ ดังนั้นในฝั่งเซิร์ฟเวอร์ก็จะเป็นต้องเขียนสคริปต์ไว้สำหรับรับข้อมูลเข้ามาประมวลผล และส่งผลลัพธ์กลับไปให้สามารถเลือกใช้ Engine ตัวใดก็ได้ที่สนับสนุน เช่น PHP, ASP, NET, JSP เป็นต้น ทั้งนี้การเขียนสคริปต์ทางฝั่งเซิร์ฟเวอร์นั้น ก็เหมือนกับการเขียนสำหรับเว็บเพจทั่วไปตามปกติ ไม่มีวิธีการหรือองค์ประกอบใดๆ เป็นพิเศษสำหรับ AJAX

12.6 การใช้ AJAX Framework

ตามขั้นตอนที่แท้จริงของเทคนิค AJAX นั้น ต้องเริ่มจากการใช้ JavaScript สร้างแบบเจ็กต์ XMLHTTP (สำหรับ IE 5-6) หรือไม่ก็ XMLHttpRequest (สำหรับ IE 7 ขึ้นไป และбраузอร์อื่นๆ) ขึ้นมาก่อน แล้วขั้นตอนต่อๆ ของ AJAX ก็ทำผ่านเมธอด และพร้อมเพร็งต์ของอบเจกต์นี้ แต่เนื่องจาก ลำดับขั้นตอนมีค่อนข้างมาก เพื่อลดความยุ่งยากตั้งกล่าวนี้ ในที่นี้จะนำ AJAX Framework มาใช้แทน ลักษณะของเฟรมเวิร์กนี้คือจะแยกส่วนที่เป็น JavaScript ที่ต้องทำชำๆ เหมือนกันทุกครั้งในรับส่งข้อมูล ไปไว้ในไฟล์ต่างหาก และสามารถนำเข้ามาใช้งานได้ในทุกเว็บเพจที่ต้องการ (คล้ายกับการ include file ใน PHP นั่นเอง) ช่วยลดขั้นตอนการเขียน JavaScript ลงໄປได้มากที่เดียว ข้อกำหนดเบื้องต้นที่ต้องรู้ที่เอาไว้ในเบื้องต้นของการใช้ AJAX Framework มีดังนี้

12.6.1 เมธอดที่ใช้ในการส่งข้อมูล

ใน AJAX ต้องกำหนดเมธอดในการส่งข้อมูล เช่นเดียวกับฟอร์ม โดยเมธอดที่กำหนดได้ก็คือ POST หรือ GET หากเป็นข้อมูลสั้นๆ ไม่เน้นความปลอดภัยนักก็ใช้เมธอด GET หากเป็นข้อมูลยาว และต้องการความปลอดภัยก็ใช้แบบ POST

12.6.2 เว็บเพจปลายทางด้านเซิร์ฟเวอร์

เว็บเพจปลายทาง คือ เว็บเพจที่จะเป็นผู้รับข้อมูลมาประมวลผล หากเป็นข้อมูลยาว และต้องการความปลอดภัยก็ใช้แบบ POST

12.6.3 ข้อมูลที่จะส่งออกໄไป

ถ้าหากเขียนโปรแกรมตามขั้นตอนปกติ จะต้องนำข้อมูลจากฟอร์มมาเรียงต่อกันในลักษณะของ Query String เช่น

Keyword=AJAX&page=2&rowperpage=20

แต่เพื่อลดความยุ่งยากตรงจุดนี้ ผู้เขียนจึงได้สร้างฟังก์ชันไว้ใน AJAX Framework สำหรับการอ่านข้อมูลจากฟอร์มเพื่อมาเรียงต่อ กันแบบ Query String ให้เรียกใช้งานแบบง่ายๆ แต่หากข้อมูลนั้นไม่ได้อ่านจากฟอร์มต้องกำหนดวิธีการจัดเรียงเอง แต่ก็ไม่มีอะไรยุ่งยาก สามารถนำความรู้เรื่องการสร้าง Query String ใน PHP มีลักษณะเช่นเดียวกันมาใช้ได้ทันที

12.6.4 การแสดงผลลัพธ์

หลังส่งข้อมูลขึ้นไปประมวลผลที่เซิร์ฟเวอร์แล้ว เมื่อผลลัพธ์ถูกส่งกลับมา หากเป็นข้อมูลstructuredata (หรืออาจมีแท็ก HTML รวมอยู่ด้วย) ต้องกำหนดว่าจะนำไปแสดงที่อิลิเม้นต์ (Element หรือส่วนนำเข้าข้อมูลจากฟอร์ม คือ ตัวเดียวกัน) ได้ภายในเว็บเพจนั้น นั่นแสดงว่าต้องสร้างอิลิเม้นต์ที่ใช้แสดงผลเอาไว้ล่วงหน้าแล้ว ก่อนที่จะเริ่มการทำงานของ AJAX โดยทั่วไปมักใช้ div หรือไม้กี span พร้อมกำหนดค่า id ให้กับมันเพื่อนำมาใช้อ้างอิงได้

12.6.5 การสร้างออบเจกต์ XMLHTTP/XMLHttpRequest

บัญชา ประศีลักษะสัง (2551, หน้า 174) ได้กล่าวไว้ว่า คลาส XMLHttpRequest และ XMLHttpRequest คือ คลาสที่ใช้จัดการข้อมูลของ AJAX ทั้งหมด ตั้งแต่การส่งข้อมูลไปจนถึงการรับข้อมูลผลลัพธ์กลับมา แต่อย่างไรก็ตาม เนื่องบรรวางเชอร์แต่ละตัวนั้นรองรับการใช้คลาสนี้แตกต่างกันออกไป ถ้าเป็น IE จะใช้คลาสที่มีชื่อว่า XMLHttpRequest แต่ถ้าเป็น Netscape/Firefox/Mozilla จะใช้คลาสที่ชื่อว่า XMLHttpRequest แต่สามารถใช้ในคลาส คือ เมธอด และพารามิเตอร์ตัวคัญที่จะใช้งานหลักๆ จะคล้ายกัน แต่เนื่องจากทั้ง XMLHttpRequest/XMLHttpRequest อยู่ในรูปแบบของคลาสไม่ใช้ออบเจกต์ จึงยังไม่สามารถนำมาใช้งานโดยตรงได้ ต้องสร้างเป็นออบเจกต์เสียก่อน โดยแยกพิจารณาตามชนิดของบรรวางเชอร์ดังนี้

ตัวอย่างที่ 12.1 การออบเจกต์ XMLHttpRequest สำหรับ IE

```
<script>
    var ajax = null;
    if (window.ActiveXObject) {
        ajax = new ActiveXObject ("Microsoft.XMLHTTP");
    }
    .
    // เขียนสคริปต์คำสั่งตามความต้องการของผู้ใช้
    .
</script>
```

จากตัวอย่างที่ 12.1 แสดงการออบเจกต์ XMLHttpRequest สำหรับ IE อธิบายได้ดังนี้

- 1) สร้างตัวแปรในสแตนช์สำหรับออบเจกต์ โดยกำหนดตัวแปรชื่อ ajax



- 2) ตรวจสอบว่าบราวเซอร์รองรับการใช้ ActiveX หรือไม่ ถ้ารองรับแสดงว่าเป็น IE เพราะเทคโนโลยี ActiveX นั้นจะมีเฉพาะในไมโครซอฟต์ แต่ถ้าไม่รองรับแสดงว่าเป็นบราวเซอร์ชนิดอื่นๆ
- 3) สร้างออบเจกต์สำหรับ AJAX เพื่อนำไปใช้งานต่อไป

ตัวอย่างที่ 12.2 การออบเจกต์ XMLHttpRequest สำหรับบราวเซอร์ที่ไม่ใช่ IE

```
<script>
    var ajax = null;
    if (window.XMLHttpRequest) {
        ajax = new XMLHttpRequest;
    }
    .
    // เขียนสคริปต์คำสั่งตามความต้องการของผู้ใช้
    .
</script>
```

จากตัวอย่างที่ 12.2 การออบเจกต์ XMLHttpRequest สำหรับบราวเซอร์ที่ไม่ใช่ IE หรือบราวเซอร์ใดดังนี้

- 1) สร้างตัวแปรอินสแตนซ์สำหรับออบเจกต์ โดยใช้ตัวแปรชื่อ ajax แบบ XMLHttpRequest
- 2) ตรวจสอบว่าบราวเซอร์รองรับการใช้ XMLHttpRequest หรือไม่ ถ้ารองรับจึงจะสร้างออบเจกต์นี้ได้
- 3) สร้างออบเจกต์สำหรับ AJAX เพื่อนำไปใช้งานต่อไป

แม้ว่าผู้ใช้โดยส่วนใหญ่จะใช้ IE เป็นหลัก และมีเพียงส่วนน้อยที่ใช้บราวเซอร์อื่นๆ แต่การสร้างเว็บไซต์ที่ดีควรสามารถใช้ได้ในหลายๆ บราวเซอร์ จะได้ไม่เกิดปัญหา จากการสร้างออบเจกต์สำหรับ AJAX ที่ผ่านมาจะเห็นว่าสามารถตรวจสอบบราวเซอร์ของผู้ใช้งานได้ ดังนั้นการประยุกต์ใช้ในขั้นตอนนี้ คือ การตรวจสอบบราวเซอร์ก่อนการใช้งานดังนี้

ตัวอย่างที่ 12.3 การออบเจกต์ของ AJAX ให้รองรับในทุกบราวเซอร์

```
<script>
    var ajax = null;
    if (window.ActiveXObject) {
        ajax = new ActiveXObject ("Microsoft.XMLHTTP");
    } else if (window.XMLHttpRequest) {
        ajax = new XMLHttpRequest;
```

```

} else {
    alert ("บรรดาเซอร์ของคุณไม่รองรับการใช้งาน AJAX");
}
.
// เขียนสคริปต์คำสั่งตามความต้องการของผู้ใช้
.

</script>

```

12.7 พังก์ชันที่ใช้จัดการข้อมูล

ภายใน AJAX Framework ผู้เขียนได้สร้างฟังก์ชันเพิ่มเติมขึ้นมา 2 พังก์ชัน เพื่อใช้ในการรับส่งข้อมูล และอ่านข้อมูลจากฟอร์มดังนี้

12.7.1 พังก์ชัน AJAXLoad ()

ใช้ในการส่งข้อมูลจากบรรดาเซอร์ไปยังเซิร์ฟเวอร์ และรับผลลัพธ์กลับมาแสดงผลที่อัลิเมนต์ที่กำหนด มีรูปแบบดังนี้

รูปแบบ

AJAXLoad (เมธอดเว็บเพจปลายทาง, ข้อมูลที่จะส่งออกไป, อัลิเมนต์แสดงผล)

ตัวอย่างที่ 12.4 พังก์ชัน AJAXLoad ()

```
AJAXLoad ("post,test.php","kw=AJAX&page=10","mydiv");
```

12.7.2 พังก์ชัน getFormData()

ใช้ในการอ่านข้อมูลจากฟอร์มมาต่อ กันแบบ Query String โดยต้องระบุ id หรือชื่อแอ็ตทริบิวต์ของฟอร์มที่ต้องการอ่านข้อมูล

รูปแบบ

getFormData (form_id หรือ form_name)

ตัวอย่างที่ 12.5 พังก์ชัน getFormData()

```
var data = getFromData ("from1");
```

12.8 แนวทางการพัฒนา AJAX Application

กระบวนการของ AJAX สามารถแยกกระบวนการทำงานเป็น 2 ส่วน คือ 1) กระบวนการฝังบรรดาเซอร์ และ 2) กระบวนการฝังเซิร์ฟเวอร์ มีรายละเอียดดังนี้

12.8.1 การบวนการฝังบรรดาเซอร์

- ถ้าจะส่งข้อมูลจากฟอร์มออกไป ก็ให้แนบฟอร์มและส่วนนำเข้าข้อมูลมาทางตามปกติทั้งนี้ต้องกำหนดค่า id หรือ name ให้กับฟอร์มด้วย เพื่อใช้อ้างถึงในการอ่านข้อมูล



2) จัดวางอิลิเม้นต์ที่จะใช้แสดงผลลงใน ตัวหนังที่ต้องการ อาจใช้ Div หรือ Span และจะต้องกำหนดค่า id ให้กับอิลิเม้นต์ด้วย ตัวอย่างดังนี้

```
<div id="diplayAJAX"> </div>
```

3) สร้างฟังก์ชัน JavaScript สำหรับจัดการ AJAX โดยภายในฟังก์ชันเป็นการจัดเตรียมข้อมูลสำหรับฟังก์ชัน AJAXLoad() ตามที่ได้กล่าวถึงมาแล้ว ตัวอย่างดังนี้

ตัวอย่างที่ 12.6 การสร้างฟังก์ชัน JavaScript สำหรับจัดการ AJAX

```
<head>
    <script src="../AJAX/framework.js"> </script>
    <script>
        function AJAXCall () {
            var data = getFromdataa ('form1');
            var URL = 'AJAX_test.php'; // นำข้อมูลไปกำหนดให้แก่ฟังก์ชัน AJAXLoad()
                // เพื่อส่งออกไป ฟังก์ชันนี้อยู่ใน framework.js
            AJAXLoad ('post', URL,data, 'DisplayAJAX');
        }
    </script>
</head>
```

4) จากฟังก์ชัน AJAXCall() ที่สร้างขึ้นในข้อที่ 3 ต้องพิจารณาต่อไปว่า จะเรียกฟังก์ชันนี้ขึ้นมาทำงานเมื่อใด เช่นอาจเพิ่มปุ่ม Button เข้าไป แล้วเพื่อเรียกใช้ฟังก์ชัน AJAXCall ดังนี้

ตัวอย่างที่ 12.7 การสร้างปุ่มเพื่อเรียกใช้ฟังก์ชัน AJAXCall

```
<input type="button" value="OK" onclick = "AJAXCall ()"/>
```

12.8.2 กระบวนการฝึกเชิร์ฟเวอร์

ทางด้านเชิร์ฟเวอร์นั้น ก็เชิร์ฟที่เหมือนกับการเขียนเว็บเพจปกติ โดยไม่มีอะไรเป็นพิเศษ สำหรับ AJAX แต่ก็พอกสรุปเป็นหลักการได้ดังนี้

- 1) การรับข้อมูลที่ถูกส่งเข้ามาผ่านตัวแปร \$_GET หรือ \$_POST ขึ้นอยู่กับเมธอดที่ถูกส่งเข้ามา
- 2) หลังการประมวลผล ก็ส่งข้อมูลกลับออกไปด้วยเพื่อแสดงผลด้วยฟังก์ชัน echo() หรือ print() ตามปกติ ถ้าข้อมูลที่ส่งนั้นยาวมากก็ควรนำรวมเป็นสตริงเดียวกันก่อนส่ง

12.9 ปัญหาภาษาไทยใน AJAX

ในการใช้ AJAX ร่วมกับภาษา PHP จะมีปัญหาสำคัญอย่างหนึ่ง คือ การแสดงภาษาไทยจะกลายเป็นอักขระที่อ่านไม่รู้เรื่อง เช่น จากตัวอย่างที่แล้ว หากไม่ใช้ฟังก์ชัน header () และถ้าใส่ข้อมูลที่ไม่ใช่ตัวเลข จะได้ผลลัพธ์เป็นคำว่า "ค่าที่ใส่ไม่ใช่ตัวเลข"

สำหรับปัญหาอักขระภาษาไทยที่เกิดขึ้นกับ AJAX/PHP โดยส่วนใหญ่จะมี 2 ลักษณะ คือ

12.9.1 ถ้าผลลัพธ์ที่ส่งกลับมาแสดงผลภาษาไทยไม่ถูกต้อง แก้ปัญหาโดยใช้ฟังก์ชัน header () แล้วกำหนด Content-Type เป็น tis-620 หรือ utf-8 เช่น

ตัวอย่างที่ 12.8 การใช้ฟังก์ชัน header () เพื่อควบคุมการแสดงผลภาษาไทย

```
header ("Content-Type:text/plain; charset=tis-620");
```

12.9.2 ถ้ามีการส่งข้อมูลที่มีภาษาไทยจากฝั่งбраузอร์เข้าไป ก่อนส่ง JavaScript จะเปลี่ยนเป็น Unicode แบบ utf-8 ถึงแม้ว่า utf-8 จะรองรับภาษาไทยได้ แต่ก็อาจเกิดปัญหานางกรณี ดังนั้นเพื่อเป็นการป้องกันปัญหาเกี่ยวกับอักขระ ทางฝั่งเซิร์ฟเวอร์ที่รับข้อมูลควรเข้ารหัสใหม่ให้เป็น tis-620 ด้วย ฟังก์ชัน iconv () ฟังก์ชันนี้ก็คุ้นเคยกันดีอยู่แล้ว เพราะในหลาย ๆ บทที่ผ่านมาก็ใช้งานมาตลอด เช่น

ตัวอย่างที่ 12.9 การส่งข้อมูลที่มีภาษาไทยจากฝั่งbrauzerเข้าไป ก่อนส่ง JavaScript

```
$input = iconv ("utf-8","tis-620",$_POST["input"]);
```

ในทางปฏิบัติ การรับข้อมูลของทางฝั่งเซิร์ฟเวอร์ที่ส่งผ่าน AJAX เข้ามา ควรเข้ารหัสก่อนเสมอ ถึงแม้ว่าข้อมูลที่ผ่านส่งไปจะไม่มีภาษาไทยอยู่ด้วยก็ไม่มีปัญหา หลังการใช้ฟังก์ชัน iconv () ข้อมูลที่ได้จะตรงกับข้อมูลที่ส่งไปเสมอ

12.10 การส่งผลลัพธ์กลับมาเป็น JavaScript

การส่งผลลัพธ์กลับมาในรูปแบบสตริงธรรมดา อาจทำให้ไม่สามารถแก้ไขอะไรได้มากนัก โดยเฉพาะอย่างยิ่งในกรณีที่ต้องควบคุมการทำงานจากเซิร์ฟเวอร์ วิธีที่จะทำได้ก็คือ ส่งผลลัพธ์กลับคืนมา ในรูปแบบคำสั่งของ JavaScript โดยแยกพิจารณาระหว่างฝั่งเซิร์ฟเวอร์และbrauzerดังนี้

12.10.1 การส่งผลลัพธ์ที่เป็น JavaScript ในฝั่งเซิร์ฟเวอร์

จะต้องสร้างคำสั่ง JavaScript ในรูปแบบของสตริงธรรมดา โดยจะมีจำนวนคำสั่งเท่าไหร่ ก็ได้ แต่ต้องไม่มีแท็ก <script> และ </script> คร่อมระหว่างคำสั่งเหล่านั้น นอกจากนี้เพื่อให้ทางฝั่ง brauzer สามารถตรวจสอบได้ว่า ผลลัพธ์ที่ได้รับมานั้นเป็น JavaScript ควรกำหนดเดอร์กำกับไว้ด้วยฟังก์ชันheader () ในลักษณะดังนี้

```
header ("Content-type:text/javascript; charset=tis-620");
```

หรือบางครั้งการจะส่งผลลัพธ์กลับไปแบบใด อาจขึ้นกับเงื่อนไข ดังนั้นอาจตรวจสอบเงื่อนไขก่อนกำหนดเดอร์ เช่น



ตัวอย่างที่ 12.10 การเขียนคำสั่งเพื่อให้รองรับภาษาไทยผ่านฟังก์ชัน header ()

```
<?php  
$x = ...;  
if (empty($x)) {  
    header ("Content-Type: text/plain; charset=tis-620");  
    echo "...";  
} else {  
    header ("Content-Type: text/javascript; charset=tis-620");  
    //สร้างคำสั่ง JavaScript ในแบบสตริง กี่คำสั่งก็ได้ แล้วส่งออกไป  
    $js = "  
var el = document.getElementById('x');  
el.style.display='none'  
alert ('...');  
";  
echo $js;  
}  
?>
```

12.10.2 การจัดการผลลัพธ์ที่เป็น JavaScript ผ่านบรรทัดเดียว

หากเขียนโปรแกรม Javascript เอง ปกติก็ต้องมีการตรวจสอบเชดเดอร์ของผลลัพธ์ก่อนแล้วใช้ฟังก์ชัน eval () ของใน JavaScript ในการประมวลผล แต่เนื่องจากใน AJAX Framework นั้นผู้เขียนได้ทำให้รองรับผลลัพธ์ที่เป็นคำสั่ง JavaScript โดยอัตโนมัติอยู่แล้ว ดังนั้นจึงไม่ต้องทำอะไรเพียงแต่ผ่านเซิร์ฟเวอร์ต้องกำหนด เชดเดอร์ ก่อนไม่เช่นนั้นจะไม่สามารถจำแนกได้ว่า ผลลัพธ์เป็นสตริง หรือเป็น JavaScript และสำหรับอิลิเมนต์ที่จะใช้แสดงผล หากจะกำหนดวิธีการผ่าน JavaScript อาจไม่มีอิลิเมนต์นี้ก็ได้ และกำหนดค่าให้กับฟังก์ชัน AJAXLoad () เป็น null เช่น

```
AJAXLoad ('get', URL, data, null);
```

เพื่อให้เข้าในแนวทางการส่งผลลัพธ์เป็น JavaScript มากยิ่งขึ้น ตัวอย่าง การส่งผลลัพธ์เป็น JavaScript โดยการจะเป็นการรับข้อมูลตัวเลข 2 ตัวและเครื่องหมายแล้วส่งด้วยเทคนิค AJAX ไปคำนวณที่เซิร์ฟเวอร์ หลังจากนั้นส่งผลลัพธ์กลับมาในแบบคำสั่ง JavaScript

ตัวอย่างที่ 12.11 ตัวอย่างการเรียกใช้ฟังก์ชัน AJAXcall ในไฟล์ index.html

```
<head>
```

```
.
```

```
.
```

```

<script src="../AJAX/framework.js"></script>

<script>
function AJAXcall () {
    var data = getFormData ('from1');
    var URL = 'calculator.php';
    AJAXLoad ('post', URL, data, null);
    //ให้อิเลเมนต์ที่ใช้แสดงผลเป็น null เพราะส่งผลลัพธ์เป็น JavaScript
}
</script>
</head>

```

เพิ่มตัวจัดการอีเวนต์ (Event) ให้กับปุ่ม Button โดยการเรียกใช้ฟังก์ชัน AJAXCall ()

```
<input type="button" value="คำนวน" onclick = "AJAXCall ()"/>
```

ตัวอย่างที่ 12.12 ไฟล์ calculator.php เป็นเว็บเพจฝึกเชิร์ฟเวอร์สำหรับรับข้อมูลเพื่อประมวลผล

```

<?php
$num1 = $_POST ['num1'];
$num2 = $_POST ['num2'];
$op = $_POST ['op'];
$result = 0 ;
switch ($op) {
    case "+": $result = num1 + $num2; break;
    case "-": $result = num1 - $num2; break;
    case "*": $result = num1 * $num2; break;
    case "/": $result = num1 / $num2; break;
}
// สร้างคำสั่ง JavaScript ในแบบสตริงของ PHP มาแทรกลงໄປได้
$js =
    var msg = '$num1 $op $num2 = $result';
";
header ("Content-Type:Text/javascript;Charset=tis-620");
echo $js;
?>

```



การใช้ AJAX กับการตรวจสอบข้อมูล เทคนิคนิยมที่ใช้กันมากในปัจจุบัน สำหรับตัวอย่างนี้จะเป็นการตรวจสอบล็อกอินว่าใช้งานได้หรือไม่ โดยส่งล็อกอินผ่าน AJAX ไปยังเซิร์ฟเวอร์ และส่งผลลัพธ์เป็น JavaScript

ตัวอย่างที่ 12.13 การใช้ AJAX กับการตรวจสอบข้อมูล ภายในไฟล์ index.php

```
<head>
    .
    .
<script src="../AJAX/framework.js"></script>
<script>
function AJAXCall () {
    var date = getFormData ('form1');
    var url = 'check_login.php';
    AJAXLoad ('post', url, date, 'displayAJAX');
}
// พังก์ชันนี้จะถูกเรียกขึ้นมาทำงานจากฝั่งเซิร์ฟเวอร์
// เพื่อแสดงผลลัพธ์ว่าล็อกอินนั้นสามารถใช้ได้หรือไม่
function displayResult (valid) {
    var el = document.getElementById('displayAJAX');
    if (!valid) {
        el.style.color = 'red';
        el.innerHTML = 'ชื่อผู้ใช้ไม่ได้ หรือมีผู้อื่นใช้แล้ว กรุณาระบุชื่อใหม่';
        Document.getElementById('login').select();
    } else {
        el.style.color = 'black';
        el.innerHTML = 'ชื่อผู้ใช้สามารถใช้งานได้';
    }
}
</script>
</head>
```

กำหนดอีเวนต์ onblur เพื่อว่าหลังป้อนข้อมูลไปแล้ว และย้ายไปยังส่วนนำเข้าข้อมูลอื่นๆ จะใช้อีเวนต์นี้ในการส่งล็อกอินผ่าน AJAX ไปตรวจสอบ โดยในมุมมองอัพเดทให้เพิ่มอีเวนต์ onblur และเพิ่มแท็ก ดังนี้

```
<input name="login" type="text" id="login" onblur="AJAXCall ( )"/>
<span id="displayAJAX"></span>
```

ตัวอย่างไฟล์ check_login.php เป็นเว็บเพจฝั่งเซิร์ฟเวอร์ที่รับข้อมูลล็อกอินไปตรวจสอบ แล้วส่งผลลัพธ์เป็น JavaScript กลับมาโดยในที่นี่จะเป็นการเรียกใช้ฟังก์ชัน displayResult ที่เว็บเพจ index.php ขึ้นมาทำงาน

ตัวอย่างที่ 12.14 รับข้อมูลล็อกอินไปตรวจสอบ แล้วส่งผลลัพธ์เป็น JavaScript

```
<?php
$login = $_POST['LOGIN'];
$patter = "^[a-zA-Z0-9]{3,10}$";
$users = array ("abc","123","xyz"); //สมมติซื่อผู้ใช้ที่มีอยู่แล้ว
// ตรวจสอบ login และเรียกฟังก์ชัน Javascript displayResult ()
// ที่เว็บเพจ index.php ขึ้นมาทำงาน
If (!eregi (pattern,$login) || in_array ($login,$users)) { //ถ้าล็อกอินไม่สำเร็จ
    $js = "displayResult (false);";
} else { //ถ้าล็อกอินสำเร็จ
    $js = "displayResult (ture);";
}
header ("Content-Type:text/javascript;Charset=tis-620");
echo $js;
?>
```

12.11 การอัปเดตส่วนนำเข้าข้อมูล Select ด้วยเทคนิค AJAX

โดยปกติแล้วตัวเลือกของส่วนนำเข้าข้อมูล Select จะสร้างไว้ล่วงหน้าด้วยแท็ก <option> เช่น อัพเดตส่วนนำเข้าข้อมูล Select ด้วยเทคนิค AJAX

การเปลี่ยนแปลงตัวเลือกของส่วนนำเข้าข้อมูล Select จะต้องอาศัยคำสั่ง JavaScript เป็นหลัก ดังนั้นก่อนที่จะกล่าวถึงเทคนิคการใช้ร่วมกับ AJAX จะกล่าวถึงการใช้ JavaScript ในการแปลงรายการใน Select ก่อน

12.11.1 การสร้างตัวเลือกด้วย JavaScript

หากต้องการจะเพิ่มตัวเลือกนั้นต้องเริ่มที่การสร้างตัวเลือกใหม่ขึ้นมาก่อน การสร้างตัวเลือกใหม่นั้นวิธีที่ง่ายที่สุด คือ การสร้างด้วยออบเจกต์ Option โดยมีรูปแบบดังนี้

```
option ('text', 'value')
```



เมื่อ text หมายถึง ข้อความที่ปรากฏบนตัวเลือกนั้น ต้องกำหนดในรูปแบบสตริง
value หมายถึง ค่าของตัวเลือกนั้น ต้องกำหนดในรูปแบบสตริง

ตัวอย่างที่ 12.15 การสร้างตัวเลือกด้วย JavaScript

```
var opt = new option ('Thailand', 'th');
```

เทียบได้กับในรูปแบบ HTML ต่อไปนี้

```
<option value="th">Thailand</option>
```

12.11.2 การเพิ่มตัวเลือกด้วย JavaScript

การเพิ่มตัวเลือกเข้าไปในส่วนนำเข้าข้อมูล Select นั้นปกติทำได้หลายวิธี แต่บางวิธีจะได้ผลเฉพาะกับбраузอร์ IE เท่านั้น ดังนั้นจะกล่าวถึงเฉพาะวิธีที่ใช้ได้ทั้ง IE และ Firefox โดยใช้วิธีการกำหนดค่าให้แก่ คุณสมบัติตัวเลือก เช่น ถ้าต้องการเพิ่มตัวเลือกเข้าไปให้เป็นรายการลำดับที่สาม ก็กำหนดสคริปต์คำสั่งดังนี้

ตัวอย่างที่ 12.16 การเพิ่มตัวเลือกด้วย JavaScript

```
Document.getElementById ('id').option [2] = opt; // เมื่อรายการแรกลำดับเป็น0
```

12.11.3 การลบตัวเลือกด้วย JavaScript

การลบนั้นเพื่อให้เป็นแนวทางเดียวกับการเพิ่ม ก็อาจใช้วิธีกำหนดคุณสมบัติตัวเลือกของรายการลำดับนั้นให้เป็น Null เช่น

ตัวอย่างที่ 12.17 การลบตัวเลือกด้วย JavaScript

```
Document.getElementById ('id').option [2] = Null; // ลบรายการที่ 3
```

12.11.4 การใช้ AJAX กับการอัพเดทส่วนนำเข้าข้อมูล Select

หลังจากที่ทราบเทคนิคการเปลี่ยนแปลงตัวเลือกด้วย JavaScript แล้ว ต่อไป คือ การนำมาประยุกต์ใช้ร่วมกับ AJAX การเปลี่ยนแปลงตัวเลือกที่ได้กล่าวมาแล้ว ข้อมูลที่จะทำตัวเลือกถูกกำหนดตามตัวมาแล้ว แต่เป็นกรณีที่ข้อมูลเก็บอยู่ในฐานข้อมูลทางฝั่งเซิร์ฟเวอร์ อาจไม่สามารถใช้วิธีตามที่กล่าวมาแล้วได้ แต่จากการที่สร้างคำสั่งที่เซิร์ฟเวอร์ แล้วส่งผ่านทาง AJAX จึงแสดงผลบนбраузอร์ได้

เว็บเพจที่จะแสดงตัวเลือกทางด้านบริษัทของตัวเอง ต้องประกอบด้วยสิ่งที่จะใช้กำหนดเงื่อนไขในการเปลี่ยนตัวเลือก โดยทั่วไปนิยมใช้ส่วนนำเข้าข้อมูล Select คู่กัน เช่น Select อันแรกแสดงชื่อฐานข้อมูลที่มีใน MySQL และอีกอันแสดงชื่อตารางที่มีอยู่ในฐานข้อมูลที่เลือกใน Select อันแรก เป็นต้น

เมื่อส่งข้อมูลจาก Select อันแรกผ่าน AJAX ขึ้นไป ทางฝั่งเซิร์ฟเวอร์จะนำข้อมูลนั้นไปใช้เป็นเงื่อนไขในการอ่านข้อมูลจากตาราง เพื่อนำมาสร้างตัวเลือกใน Select อันที่สอง

หลังจากได้ข้อมูลต่อไป คือ การส่งผลลัพธ์กลับไปเพื่ออัพเดทส่วนนำเข้าข้อมูล Select โดยก่อนที่จะอัพเดท ต้องลบตัวเลือกเดิมทั้งหมดออกไปก่อน แต่การลบตัวเลือกนั้นไม่เขียนกับข้อมูล ดังนั้นในที่นี้การลบตัวเลือกจะสร้างเป็นฟังก์ชันเอาไว้ที่เว็บเพจด้านบริษัท แล้วใช้วิธีสั่งคำสั่ง JavaScript จากด้านเซิร์ฟเวอร์มาเรียกฟังก์ชันนี้ขึ้นมา

ต่อไปคือ การนำข้อมูลที่อ่านได้จากฐานข้อมูลจากขั้นตอนก่อนนี้ มาสร้างเป็นตัวเลือกในรูปแบบคำสั่ง JavaScript เพื่อส่งกลับไปประมวลผลที่ด้านบริเวณเซอร์ ทั้งนี้ตัวเลือกอาจมีมากกว่า 1 รายการดังนั้นอาจจะต้องใช้การวนลูปเพื่อสร้างคำสั่ง JavaScript ตามจำนวนตัวเลือกที่มี

12.11.5 ขั้นตอนการออกแบบ และเขียนสคริปต์

สร้างเว็บเพจจำนวน 2 เว็บเพจ โดยเว็บเพจแรกสำหรับวางแผนนำเข้าข้อมูล Select จำนวน 2 รายการ เอาไว้ทดสอบ ส่วนอีกเว็บเพจนึงผ่านเชิร์ฟเวอร์สำหรับอ่านชื่อตารางฐานข้อมูล และส่งผลลัพธ์กลับมาในแบบ JavaScript

ตัวอย่างที่ 12.18 ตัวอย่างข้อมูลภายในไฟล์ index.php

```
<head>
.
.
<script>
function AJAXCall () {
    var data = getFormData ('form1');
    var URL = 'update_tables.php';
    AJAXLoad ('get', URL, data, null);
}
function removeOption () {
    var el = document.getElementById ('table');
    while (el.length > 0) {
        el.option [0] = null;
    }
}
</script>
</head>
```

ตำแหน่งส่วนนำเข้าข้อมูล Select อันแรก (ที่ใช้เก็บชื่อฐานข้อมูล) สิ่งที่ต้องทำก็คือ

- 1) เพิ่มอีเวนต์ change เพื่อเรียกใช้ AJAX ในการอัพเดท Select
- 2) เนื่องจากเป็นการแสดงชื่อฐานข้อมูล ควรอ่านจาก MySQL โดยตรงมากกว่าที่จะสร้างเป็นรายการไว้ล่วงหน้าดังนั้นในส่วนรายตัวเลือกใช้ สคริปต์ PHP ในการอ่านชื่อฐานข้อมูลแล้วนำมาสร้างเป็นรายการตัวเลือก



ตัวอย่างที่ 12.19 แสดงส่วนของการติดต่อกับฐานข้อมูล

```
<select id = "database" name = "database" onchange = "AJAXCall ( )>
<?php
    @mysql_connect ("localhost", "root", "leaf") or die (mysql_error ( ))
    $ dbs = mysql_list_dbs();
    while (list ($db) = mysql_fetch_row ($dbs)) {
        echo "<option value = $db></option>";
    }
?>
</select>
```

เพื่อให้เกิดการอัพเดททันทีที่เปิดเว็บเพจ ให้เพิ่มการเรียกใช้ฟังก์ชัน AJAXCall () ไว้ที่ส่วนท้ายของเว็บเพจดังนี้

```
</form>
<script> AJAXCall ( ); </script>
</body>
```

update_tables.php เว็บเพจนี้จะนำข้อมูลที่รับเข้ามาไปใช้การอ่านชื่อตารางที่มีอยู่ในฐานข้อมูลนั้น แล้วนำมาสร้างเป็นคำสั่ง JavaScript เพื่อส่งกลับไปอัพเดทที่ส่วนนำเข้าข้อมูล Select ที่แสดงชื่อตาราง ตามหลักการที่ได้กล่าวมาแล้ว

ตัวอย่างที่ 12.20 ส่วนของการ Update ข้อมูลภายในไฟล์ update_tables.php

```
<?php
    @mysql_connect ("localhost", "root", "leaf") or die (mysql_error ( ));
    $db = $_GET ['database'];
    $tbs = mysql_list_tables($db);
    // คำสั่ง JavaScript สำหรับลบ Option เดิม โดยเรียกฟังก์ชัน removeOption ( )
    // สร้างไว้ที่เว็บเพจ index.php
    $js = "removeOption ( );";
    if (mysql_num_rows($tbs) == 0 {      // ถ้าฐานข้อมูลนั้นไม่มีตารางอยู่เลย
        $js .= "
            var opt = new Option (", ");
            Document.getElementById ('table') .options [0] = opt;
    } else {
```

```

$js = 0;

while (list ($tb) = mysql_fetch_row ($tbs)) {
    //คำสั่ง javascript สำหรับสร้าง และเพิ่ม Option ใหม่ๆในครบทุกตัว
    $js .= "
        var opt = new Option ('$tb', '$tb');
        Document.getElementById ('table') .options [$i] = opt;
    ";
    $i++;
}

header ("Content-Type:text/Javascript; charset=tis-620");
echo $js;
?>

```

12.12 อีเวนต์

บุญชา ประสีลະเตสัง. (2551: 103) อธิบายความหมายของอีเวนต์ (Event) คือ เหตุการณ์หรือการกระทำการอย่างที่เกิดขึ้น เช่น การคลิกหรือเคลื่อนย้ายเมาส์ การกดปุ่มคีย์บอร์ด เป็นต้น ทั้งนี้มักใช้อีเวนต์ที่เกิดขึ้นเป็นตัวกำหนดให้เกิดการตอบสนอง หรือกระทำการอย่างต่อไป เช่น เมื่อคลิกปุ่มก็ให้ส่งข้อมูลจากฟอร์มไปยังเซิร์ฟเวอร์ หรือ เมื่อมีการพิมพ์คีย์บอร์ด ก็อาจตรวจสอบว่าข้อมูลที่ใส่เข้ามาตรงตามเงื่อนไขที่กำหนดหรือไม่ เป็นต้น

ชนิดของการเกิดอีเวนต์ เนื่องจากอีเวนต์เกิดได้จากหลายสาเหตุ เช่น การคลิกเมาส์ การเคลื่อนย้ายเมาส์ การพิมพ์ข้อความ เป็นต้น นอกจากนี้แต่ละอิเลิมเม้นท์สามารถเกิดอีเวนต์ได้เพียงบางลักษณะเท่านั้น เช่น div สามารถเกิดอีเวนต์เกี่ยวกับเมส์ได้ แต่ไม่สามารถเกิดอีเวนต์เกี่ยวกับคีย์บอร์ด เป็นต้น ดังนั้นจึงควรทราบก่อนว่าใน JavaScript มีอีเวนต์ชนิดใดบ้าง เพื่อจะได้นำไปใช้ร่วมกันได้อย่างถูกต้อง มีรายละเอียดดังนี้

ตารางที่ 12.1 แสดงอีเวนต์ของ JavaScript สำหรับกำหนดให้กับอิเลิมเม้นต์

Event Handler	การเกิดอิเลิมเม้นต์
onclick	เกิดเมื่อคลิกที่อิเลิมเม้นต์ (กดแล้วปล่อย)
ondblclick	เกิดเมื่อตัวเบิลคลิก
onmousedown	เกิดเมื่อกดเมาส์ลงไป (เกิดทันทีที่กด)
onmouseup	เกิดเมื่อปล่อยมือจากการกดเมาส์



ตารางที่ 12.1 (ต่อ)

Event Handler	การเกิดอิลิเมนต์
onmouseover	เกิดเมื่อเคลื่อนย้ายเมาส์เข้าไปในอิลิเมนต์
onmouseout	เกิดเมื่อนำเมาส์ออกจากอิลิเมนต์
onmousemove	เกิดเมื่อเคลื่อนย้ายเมาส์ไปบนอิลิเมนต์
onkeypress	เป็นลักษณะการพิมพ์ทั่วไป คือ กดแล้วปล่อย
onkeydown	เกิดเมื่อกดปุ่มคีย์บอร์ด (เกิดทันทีที่กด)
onkeyup	เกิดเมื่อปล่อยมือจากปุ่มคีย์บอร์ด
onabort	เกิดเมื่อขัดจังหวะขณะที่กำลังโหลดรูปภาพ เช่น คลิกปุ่ม Stop ของбраузர ขณะที่ยังโหลดภาพไม่สมบูรณ์
onblur	เกิดเมื่ออิลิเมนต์นั้นเสียไฟกัส หรือย้ายไฟกัสไปยังอิลิเมนต์
onchange	เกิดเมื่อเปลี่ยนข้อความใช้ช่องรับข้อมูล เช่น text, textarea หรือเกิดเมื่อเปลี่ยนรายการที่เลือกจากเดิมไปเป็นรายการอื่นใน <select>
onerror	เกิดเมื่อมีข้อผิดพลาดเกิดขึ้นขณะที่กำลังโหลดภาพ
onfocus	เกิดเมื่ออิลิเมนต์นั้นได้รับไฟกัส
onload	เกิดเมื่อเริ่มทำการโหลดเอกสารหรือรูปภาพ
onreset	เกิดเมื่อคลิกปุ่ม Reset ของฟอร์ม
onresize	เกิดเมื่อเปลี่ยนขนาดของวินโดว์
onselect	เกิดเมื่อทำแบบสีเพื่อเลือกข้อความ
onsubmit	เกิดเมื่อส่งข้อมูลจากฟอร์มไปยังเซิร์ฟเวอร์
onunload	เกิดเมื่อยกเลิกการโหลดเอกสาร เช่น คลิกปุ่ม Stop

หมายเหตุ

Event Handler คือ ตัวจัดการอีเวนต์ หรือหน้าที่การเรียกสคริปต์ส่วนที่ใช้ตอบสนองอีเวนต์ขึ้นมาทำงาน เมื่อเกิดอีเวนต์ขึ้น ถ้าตัดคำว่า "on" ข้างหน้า Event Handler ออกก็จะกลายเป็นชนิดของอีเวนต์นั้นๆ

12.13 การกำหนดอีเวนต์

ก่อนการกำหนดอีเวนต์จะต้องตรวจสอบและพิจารณา ก่อนว่าต้องการกระทำเมื่อเกิดอีเวนต์ใด และเกิดกับอิลิเมนต์ใด โดยคุ่าว่าอิลิเมนต์นั้นรองรับอีเวนต์ที่ต้องการหรือไม่ เช่น หากต้องการจะกระทำบางอย่างเมื่อผู้ใช้คลิกที่ <div> ก็ต้องคุ่าว่า <div> รองรับอีเวนต์ที่เกี่ยวกับ "click" หรือไม่เป็นต้น

12.13.1 การกำหนดอีเวนต์แบบแอททริบิวต์

วิธีนี้จะเขียนสคริปต์แทรกร่างไว้ในแท็กของอัลิเมนต์ที่เกิดอีเวนต์ โดยนำ Event Handler ของอีเวนต์นั้นมาเป็นแอททริบิวต์ตัวหนึ่ง เช่น

ตัวอย่างที่ 12.21 การแทรกรสคริปต์ในแท็กของอัลิเมนต์ที่เกิดอีเวนต์

```
<body onclick = "alert ('Hello')">
```

หมายความว่า เมื่อคลิกที่เอกสารจะเกิด alert ที่แสดงข้อความ Hello วิธีนี้เหมาะสมสำหรับสคริปต์ที่ไม่ยาวมากนัก เพราะหากเป็นสคริปต์ยาวๆ อาจจะเขียนไม่สะดวก เช่น

ตัวอย่างที่ 12.22 การแทรกรสคริปต์ในแท็กของอัลิเมนต์ที่เกิดอีเวนต์ แบบมีเงื่อนไข

```
<div id="dv" onmousedown="if (confirm ('Are you 18+')) {  
    location.href='index.html'; } else { window.close () }" > 18+ ? </div>
```

จากสคริปต์ หมายถึง เมื่อเกิดอีเวนต์ mousedown ที่ <div> จะปรากฏกล่องโต๊ะตอบ (Dialog box) ยืนยันอายุ โดยหากยืนยันว่าคุณอายุมากกว่า 18 จะโหลดหน้า index.html หากไม่จะปิดбраузอร์ที่กำลังเรียกใช้งานอยู่

12.13.2 การกำหนดอีเวนต์แบบพร้อมเพอร์ตี

วิธีนี้จะไม่เขียน Event Handler ไว้ในแท็กของอัลิเมนต์ที่ต้องการดักอีเวนต์ แต่จะนำมากำหนดในรูปแบบพร้อมเพอร์ตีของอัลิเมนต์นั้นแทน เช่น

ตัวอย่างที่ 12.23 การกำหนดอีเวนต์แบบพร้อมเพอร์ตี

```
<script>  
    function fa () {  
        .  
        .  
    }  
</script>  
<div id="dv"> Click Me </div>  
<script>  
    document.getElementById ("dv").onclick = f1;  
</script>
```

จะพบว่าชื่อฟังก์ชันที่นำมากำหนดให้กับ Event Handler นั้นจะต้องไม่มีวงเล็บต่อท้าย ต้องมีเฉพาะชื่อฟังก์ชันอย่างเดียวเท่านั้น ทั้งนี้อัลิเมนต์เดียวกันสามารถมีได้มากกว่า 1 Event Handler เช่น

ตัวอย่างที่ 12.24 การกำหนดให้อัลิเมนต์เดียวมีมากกว่า 1 Event Handler

```
var el = document.getElementById ("dv");  
el.onclick = f1;  
el.onmouseover = f2;  
el.onmouseout = f3;
```



หรือจะเขียนแบบ Anonymous Function ก็ได้ เช่น

ตัวอย่างที่ 12.25 การกำหนดอัล"in" เดียว แบบ Anonymous Function

```
var el = document.getElementById ("dv");
el.onclick = function () {
    .
    .
}
```

12.14 ขอบเจ็กต์อีเวนต์

เมื่อกิจกรรมที่เกิดขึ้นอาจจะมีข้อมูลบางอย่างเกี่ยวกับอีเวนต์นั้นที่อาจต้องการทราบ เช่น ปุ่มมาส์ หรือคีย์บอร์ดที่ถูกกด หรือตำแหน่งที่เกิดอีเวนต์ เป็นต้น เมื่อมีอีเวนต์เกิดขึ้น สำหรับ IE และข้อมูลต่างๆ เกี่ยวกับอีเวนต์จะถูกเก็บไว้ในขอบเจ็กต์อีเวนต์ (Object Event) ทั้งนี้จะต้องอ้างถึงข้อมูลเหล่านั้นในแบบพร้อมเพอร์ตี้ พร้อมเพอร์ตี้ที่นำเสนอในมีรายละเอียดดังนี้

ตารางที่ 12.2 แสดงอีเวนต์ของ JavaScript แบบพร้อมเพอร์ตี้

พร้อมเพอร์ตี้	ความหมาย
allKey	ตรวจสอบว่าคีย์บอร์ดปุ่ม "Alt" ถูกกดขณะเกิดอีเวนต์หรือไม่
ctrlKey	ตรวจสอบว่าคีย์บอร์ดปุ่ม "Ctrl" ถูกกดขณะเกิดอีเวนต์หรือไม่
shiftKey	ตรวจสอบว่าคีย์บอร์ดปุ่ม "Shift" ถูกกดขณะเกิดอีเวนต์หรือไม่
keyCode	รหัสของปุ่มคีย์บอร์ดที่ถูกกด
button	ตรวจสอบว่าปุ่มของเมาส์ปุ่มใดที่ถูกคลิก โดย 1=ซ้าย, 2=ขวา, 4=กลาง และ 3=ซ้าย+ขวา
fromElement	อิลิเมนต์ที่เมาส์ชี้อยู่ก่อนจะมาที่อิลิเมนต์ปัจจุบัน เช่น ตอนแรกเมาส์ชี้อยู่ที่ <div id="div1"> ต่อมาเลื่อนเมาส์มาอยัง <div id="div2"> ถ้าอ่านค่า fromElement ที่ div2 จะได้ค่าเป็น div1
toElement	อิลิเมนต์ที่เมาส์ชี้อยู่หลังจากเลื่อนไปจากอิลิเมนต์เดิม เช่น ตอนแรกเมาส์ชี้อยู่ที่ <div id="div1"> ต่อมาเลื่อนเมาส์มาอยัง <div id="div2"> ถ้าอ่านค่า fromElement ที่ div1 จะได้ค่าเป็น div2
srcElement	อิลิเมนต์ที่ทำให้เกิดอีเวนต์ เช่น ถ้าคลิกที่ <div id="div1"> จะได้ค่า srcElement เป็น div1
type	ชนิดของอีเวนต์ที่เกิดขึ้น คือ การนำชื่อ Event Handler มาตัดคำว่า "on" ออกไป เช่น เป็นอีเวนต์ "onclick" จะได้ type เป็น "click" เป็นต้น

ตารางที่ 12.2 (ต่อ)

พร็อปเพอร์ตี้	ความหมาย
clientX, clientY	เป็นการวัดระยะโดยเทียบกับพื้นที่ในการแสดงผลของбраузอร์
offsetX, offsetY	เป็นการวัดค่าจะวัดจากขอบด้านซ้าย และด้านบนของอิลิเม้นต์ที่ทำให้เกิดอีเวนต์
screenX, screenY	เป็นการวัดระยะโดยเทียบกับหน้าจอด้านซ้าย และด้านบนmanyจุดที่เกิดอีเวนต์
cancelButton	ใช้ในการหยุดการเกิด Event Bubbling จาก child ไปยังพาราenter
returnValue	กำหนดค่าที่ส่งคืนให้แก่ระบบ มักใช้เพื่อยกเลิกพฤติกรรมปกติของอีเวนต์ เช่น ไม่ต้องการแสดงเมนูเมื่อคลิกขวาถ้ากำหนดค่า returnValue เป็น false เป็นต้น

เนื่องจากในแต่ละพร็อปเพอร์ตี้ยังมีรายละเอียดเพิ่มเติมอีกมาก จะนำไปกล่าวถึงในหัวข้อต่อๆ ไป แยกตามความเกี่ยวข้องของอีเวนต์ สำหรับแนวทางการใช้พร็อปเพอร์ตี้ของอปเจ็กต์อีเวนต์ มีตัวอย่างดังนี้

ตัวอย่างที่ 12.26 การใช้พร็อปเพอร์ตี้ของอปเจ็กต์อีเวนต์

```
<script>
    function eventType () {
        var ev_type = event.type;
        alert ("อีเวนต์ที่เกิดขึ้น คือ " + ev_type);
    }
</script>
<body onclick = "eventType ()" onmousedown = "eventType ()">
```

12.15 อีเวนต์เกี่ยวกับเมาส์

จากเรื่องของอปเจ็กต์อีเวนต์ จะเห็นว่า เมื่อเกิดอีเวนต์เกี่ยวกับเมาส์ขึ้น มีข้อมูลหลายอย่างที่สามารถตรวจสอบ หรือนำมาใช้งานต่อไปได้ เช่น ปุ่มหรือตำแหน่งที่ถูกคลิก เป็นต้น

12.15.1 อีเวนต์เกี่ยวกับการเคลื่อนไหวของเมาส์

อีเวนต์ที่เกี่ยวกับการเคลื่อนไหวของเมาส์ได้แก่ mousemove, mouseover และ mouseout โดยมีรายละเอียดดังนี้

- 1) เมื่อเคลื่อนเมาส์เข้ามาและของของอิลิเม้นต์จะเกิดอีเวนต์ mouseover
- 2) ถ้าเคลื่อนเมาส์เข้าไปอยู่เหนืออิลิเม้นต์จะเกิดอีเวนต์ mousemove แม้จะหยุดการเคลื่อนไหวยังเป็นอีเวนต์ mousemove
- 3) หากนำเมาส์ออกไปจากอิลิเม้นต์จะเกิดอีเวนต์ mouseout
- 4) หากไม่กำหนดอีเวนต์ mousemove แต่กำหนดเฉพาะ mouseover เมื่อเคลื่อนที่เมาส์มาวางเหนืออิลิเม้นต์ก็จะเกิดอีเวนต์ mousemove



12.15.2 อีเวนต์เกี่ยวกับการกดเมาส์

อีเวนต์ที่เกี่ยวกับการกดเมาส์ได้แก่ click, dblclick, mousedown และ mouseup มีตัวอย่างดังนี้

ตัวอย่างที่ 12.27 อีเวนต์ที่เกี่ยวกับการกดเมาส์

```
<body onmousedown = "window.status=event.type"  
      onmouseup = "alert (event.type)"  
>
```

12.15.3 การตรวจสอบปุ่มเมาส์ที่ถูกกด

ปุ่มเมาส์ที่ถูกกดสามารถตรวจสอบได้จากพารามิเตอร์ button ของออบเจกต์อีเวนต์ค่าที่ได้สำหรับเบราว์เซอร์ IE มีตัวอย่างดังนี้

ตัวอย่างที่ 12.28 การตรวจสอบปุ่มเมาส์ที่ถูกกด

```
<script>  
function mouseButton () {  
    var bt = event.button;  
    switch (bt) {  
        case 1 : window.status = "ซ้าย"; break;  
        case 2 : window.status = "ขวา"; break;  
        case 4 : window.status = "กลาง"; break;  
        case default : window.status = "ไม่สามารถระบุได้"; break;  
    }  
}</script>
```

12.16 อีเวนต์เกี่ยวกับคีย์บอร์ด

อีเวนต์ที่เกี่ยวกับคีย์บอร์ด เช่น การตรวจสอบว่าปุ่มถูกกดหรือมีการกดปุ่ม Alt/Ctrl/Shift ด้วยหรือไม่ เป็นต้น ทั้งนี้การตรวจสอบปุ่มนั้น สามารถนำไปใช้ในการป้องกันไม่ให้ผู้ใช้ใส่ข้อมูลที่ไม่ตรงตามเงื่อนไข เช่น ต้องการระบุให้ป้อนข้อมูลชนิดตัวเลข 0-9 เท่านั้น หากผู้ใช้กดปุ่มคีย์ที่ไม่ใช่ 0-9 ก็จะไม่ยอมรับข้อมูลนั้น เป็นต้น

อีเวนต์ที่เกี่ยวกับคีย์บอร์ดมี 3 อีเวนต์ คือ keypress, keydown และ keyup มีตัวอย่างดังนี้

ตัวอย่างที่ 12.29 วีเ溫ต์ที่เกี่ยวกับคีย์บอร์ด

```
<body onkeydown = "window.status = event.type" onkeypress = "alert (event.type)"  
onkeyup = "alert (event.type)">
```

12.16.1 การตรวจสอบปุ่มคีย์บอร์ดที่ถูกกด

เมื่อคีย์บอร์ดปุ่มใดๆ ถูกกดสามารถตรวจสอบได้ว่าปุ่มนั้น คือ ปุ่มอะไร โดยใช้ฟังก์ชัน keyCode แต่ค่า keyCode ที่ได้จะเป็นตัวเลขรหัสที่ใช้แทนปุ่มนั้น มีรายละเอียดของรหัสดังตารางต่อไปนี้

ตารางที่ 12.3 แสดงตัวอย่างรหัสที่ใช้แทนปุ่มคีย์บอร์ด

ปุ่มคีย์บอร์ด	keyCode	ปุ่มคีย์บอร์ด	keyCode
backspace	8	right window key	92
tab	9	select key	93
enter	13	numpad 0-9	96-105
shift	16	multiply	106
ctrl	17	add	107
alt	18	subtract	109
pause/break	19	decimal point	110
caps lock	20	divide	111
escape	27	F1-F12	112-123
page up	33	num lock	144
page down	34	scroll lock	145
end	35	semi-colon	186
home	36	equal sign	187
left arrow	37	comma	188
up arrow	38	dash	189
right arrow	39	period	190
down arrow	40	forward slash	191
insert	45	grave accent	192
delete	46	open bracket	219
เลข 0-9	48-57	back slash	220
อักษร a-z	65-90	close bracket	221



ตารางที่ 12.3 (ต่อ)

ปุ่มคีย์บอร์ด	keyCode	ปุ่มคีย์บอร์ด	keyCode
left window key	91	single quote	222
ก-ซ	3585-3630	๐-๙	3664-3673

ในทางตรงข้าม ถ้าทราบค่า keyCode และต้องการทราบว่าตั้งกับอักษรใด ให้ใช้ เมธอด fromCharCode () ของคลาส String ในการแปลง เช่น

```
var char = String.fromCharCode (65);
```

ตัวอย่างที่ 12.30 การเขียนคำสั่งเพื่อควบคุมการป้อนเลข 0-9 เท่านั้น

```
<html>
<head>
<script>
    function isValidKey () {
        var keyCode = event.keyCode;
        if (keyCode < 48 || event.keyCode > 57) {
            window.status = "key invalid";
            alert ("กรุณาป้อนตัวเลข 0-9 เท่านั้น");
        } else {
            window.status = String.fromCharCode (keyCode);
        }
    }
</script>
</head>
<body onkeydown = "isValidKey ()">
</body>
</html>
```

12.17 ตัวอย่างการสร้าง AJAX dictionary จาก longdo ด้วย jQuery

ตัวอย่างการใช้งานต่อไปนี้ เป็นการนำระบบค้นหาคำศัพท์ จากเว็บไซต์ dict.longdo.com เหมาะสำหรับใครที่ต้องการนำระบบค้นหาคำศัพท์ dictionary รูปแบบ AJAX ไปตกแต่งเว็บไซต์ โดยเฉพาะเว็บไซต์เกี่ยวกับการศึกษา มีขั้นตอนดังต่อไปนี้

12.17.1 สร้างไฟล์ php ชื่อ ajax_translate.php มีคำสั่งดังต่อไปนี้

ตัวอย่างที่ 12.31 การสร้าง AJAX dictionary ในไฟล์ ajax_translate.php

```

1<?php
2     header("Content-type:text/html; charset=UTF-8");
3     header("Cache-Control: no-store, no-cache, must-revalidate");
4     header("Cache-Control: post-check=0, pre-check=0", false);
5     if(isset($_GET['keyword']) && trim($_GET['keyword'])!=""){
6         $keyword=trim($_GET['keyword']);
7         $data=file_get_contents("http://dict.longdo.com/mobile.php?search=".$keyword);
8         echo strip_tags($data,"<a><table><td><tr><font><style><meta><br>");
9     } else { // กรณีไม่มีการส่งคำค้นมา
10        echo "โปรดระบุคำที่ต้องการแปล"; // แสดงข้อความแจ้งเตือน
11    }
12?>

```

12.17.2 สร้างไฟล์สำหรับทดสอบ demo_ajax_translate.php โดยจะมีอยู่ 3 ส่วน หลักๆ ดังนี้

- 1) ส่วนของ css สำหรับกำหนดรูปแบบ สีข้อความ พื้นหลัง หรืออื่นๆ ตามต้องการ
- 2) ส่วนของ element องค์ประกอบในการใช้งาน
- 3) ส่วนของ javascript โดยใช้งานผ่าน jQuery

ตัวอย่างที่ 12.32 ไฟล์สำหรับทดสอบ demo_ajax_translate.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=tis-620" />
<title>longdo ajax dictionary</title>
<!--1.ส่วนของ css สำหรับกำหนดรูปแบบ สีข้อความ พื้นหลัง หรืออื่นๆ ตามต้องการ-->
<style type="text/css">
div#myblock_dict{
    font-family:tahoma, "Microsoft Sans Serif", sans-serif, Verdana;
    font-size:12px;
    margin:auto;
    width:350px;
}

```



```
input#translate_it{
    background-color:#F6C;
    color:#FFF;
    border:1px groove #F9C;
    cursor:pointer;
}

div#input_search{
    background-color:#000;
    color:#FFF;
    text-align:center;
}

div#context_search{
    border:1px solid #F9C;
    height:300px;
    overflow:auto;
}

div#context_search{
    font-family:tahoma, "Microsoft Sans Serif", sans-serif, Verdana;
    font-size:12px;
    padding:5px;
    background-color:#FDDEDF;
}

div#context_search a{
    margin-left:3px !important;
    color:#F09;
}

div#context_search td{
    padding:5px !important;
}

</style>
</head>
<body>
```



```

<!--2.ส่วนของ element องค์ประกอบในการใช้งาน-->
<div id="myblock_dict">
<div id="input_search">คำค้น
    <input type="text" name="keyword_q" id="keyword_q" />
    <input type="button" name="translate_it" id="translate_it" value="แปล" />
</div>
<div id="context_search">
</div>
</div>

<!--3.ส่วนของ javascript โดยใช้งานผ่าน jQuery-->
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script>
<script type="text/javascript">
$(function(){
    var loading_img='<center>';
    loading_img+='';
    loading_img+='</center>';
    $("#keyword_q").click(function(){ // เมื่อคลิกที่ช่องคำค้น
        $(this).select(); // ถ้ามีข้อความอยู่ ให้ทำการเลือกข้อความนั้น
        $("#context_search").html(""); // ล้างค่าข้อความผลลัพธ์เดิม ถ้ามี
    });
    $("#translate_it").click(function(){ // เมื่อคลิกที่ปุ่มคำว่า แปล
        $("#context_search").html(loading_img);
        $.get("ajax_translate.php",{keyword:$._trim($("#keyword_q").val())
},function(data) {
            $("#context_search").html(data); // แสดงผลลัพธ์จากการค้นหา
        });
    });
    $("#keyword_q").keyup(function(event){
        if(event.keyCode==13) { // เมื่อกดปุ่ม Enter ให้เริ่มการค้นหา
            $("#context_search").html(loading_img);
            $.get("ajax_translate.php",{keyword:$._trim($("#keyword_q").val())},function(data){}
        }
    });
});

```



```

        $("#context_search").html(data);

    });

}

});

$("div#context_search a").live("click",function(){

    var text_search=$.trim($(this).text());

    $("#context_search").html(loading_img);

    $("#keyword_q").val(text_search);

    $.get("ajax_translate.php",{keyword:text_search},function(data){

        $("#context_search").html(data); // แสดงผลลัพธ์จากการค้นหา

    });

    return false;

});

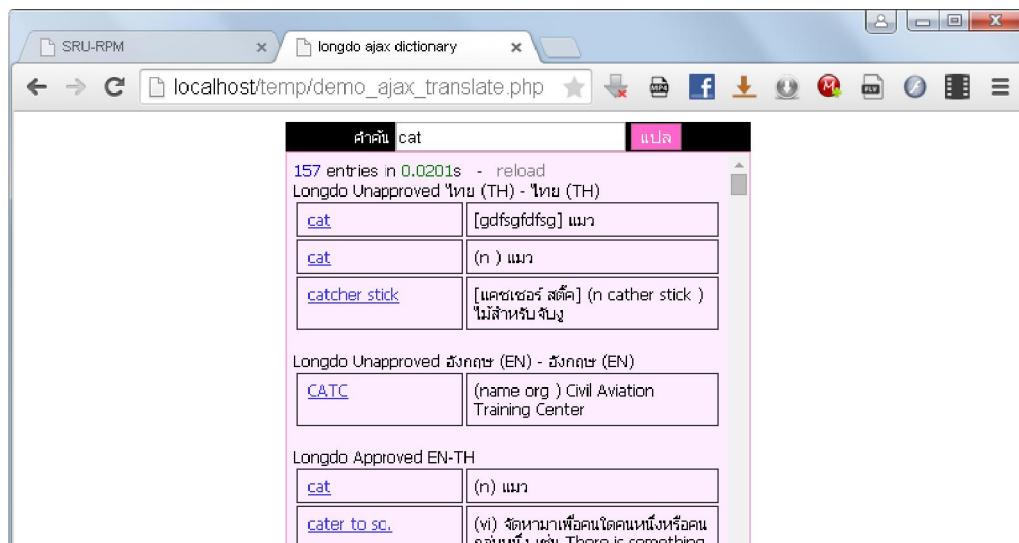
});

</script>

</body>

</html>

```



ภาพที่ 12.4 ตัวอย่างการสร้าง AJAX dictionary จาก longdo ด้วย jQuery

สรุป

Ajax เป็นเทคนิคใหม่ที่คนส่วนใหญ่กำลังเริ่มศึกษา กัน โดยการนำ Ajax เข้ามาช่วยนั้นทำให้ ขึ้นงานเว็บเพจดูมีความน่าสนใจมากยิ่งขึ้น สามารถทำงานร่วมกับภาษา PHP และ HTML ได้อย่างลงตัว อีกทั้งช่วยเพิ่มประสิทธิภาพในการแสดงผลหน้าเว็บเพจได้รวดเร็วขึ้น ไม่ต้องเสียเวลาหยุดรอการโหลดหรือ การรีเฟรชหน้าจอ แต่ในการนำเทคนิคนี้มาใช้งานได้นั้นควรศึกษาและมีพื้นฐานภาษา JavaScript หลังจากนั้นจึงค่อยเริ่มศึกษา Ajax เนื้อหาในบทนี้เป็นเพียงแนะนำในเบื้องต้นเท่านั้น ยังไม่ครอบคลุมถึง คำสั่งทั้งหมด

คำถามท้ายบท

1. จงอธิบายหลักการทำงานของ AJAX
2. จงยกตัวอย่างอีเวนต์ที่เกี่ยวกับเม้าส์ พร้อมอธิบาย
3. จงยกตัวอย่างอีเวนต์ที่เกี่ยวกับคีย์บอร์ด พร้อมอธิบาย
4. จงอธิบายสคริปต์คำสั่งต่อไปนี้ จะเกิดอีเวนต์เมื่อไหร่ และจะได้ผลลัพธ์อย่างไร

```
<div id="dv" onmousedown="if (confirm ('Are you 18+')) {
location.href='index.html'; } else { window.close ( ) }" > 18+ ? </div>
```

5. จงอธิบายความเหมือนและความแตกต่างระหว่าง JavaScript และ AJAX





บทที่ 13

ฐานข้อมูล MySQL และการใช้งาน phpMyAdmin

MySQL จัดอยู่ในกลุ่มของระบบบริหารจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) ชนิดหนึ่ง เป็นที่นิยมใช้กันมากในปัจจุบัน เหมาะกับองค์กรขนาดกลางที่มีข้อมูลไม่มากนัก ในบทนี้จะกล่าวถึงคำสอบถามหรือคำสั่ง (Query) พื้นฐานของ MySQL ที่ใช้งานบ่อยครั้ง เช่น 1) การจัดการฐานข้อมูล 2) การจัดการตาราง และ 3) การจัดการกับขอบเขตข้อมูล เป็นต้น ซึ่งทั้ง 3 ส่วนนี้เป็นองค์ประกอบที่สำคัญของฐานข้อมูล ในส่วนของขอบเขตข้อมูลก็จำเป็นต้องกำหนดชนิดของขอบเขตข้อมูลให้ถูกต้องกับลักษณะการใช้งานรวมถึงการกำหนดคุณสมบัติอื่นๆ จะเห็นได้ว่ามีรายละเอียดปลีกย่อยจำนวนมากที่จำเป็นต้องกำหนดให้ถูกต้องในแต่ละส่วน ทำให้ผู้ใช้งานอาจจะไม่สะดวกในการใช้งาน ดังนั้นในปัจจุบันก็มีการพัฒนาโปรแกรมเพื่ออำนวยความสะดวกสำหรับการบริหารจัดการฐานข้อมูล MySQL ก็คือ โปรแกรม phpMyAdmin

13.1 คำสอบถามหรือคำสั่งพื้นฐานของ MySQL

การศึกษา MySQL ควรเริ่มต้นจากการใช้คำสอบถามหรือคำสั่งพื้นฐานให้เกิดความคุ้นเคยเสียก่อน แม้ในการใช้งานจริงจะส่งคำสอบถามหรือคำสั่งผ่านฟังก์ชันต่างๆ ของภาษา PHP ก็ตาม แต่หากไม่มีพื้นฐานในการใช้งานกับฐานข้อมูลโดยตรงจะทำให้ไม่เข้าใจขั้นตอน และหลักการทำงานที่แท้จริง ลักษณะคำสอบถามหรือคำสั่งของ MySQL จะใช้รูปแบบคำสอบถามหรือคำสั่งตามมาตรฐานของภาษา SQL (Structure Query Language)

13.1.1 SHOW DATABASES;

SHOW DATABASES เป็นคำสอบถามเพื่อเรียกดูรายชื่อฐานข้อมูลทั้งหมดที่มีอยู่ภายในระบบฐานข้อมูล MySQL มีตัวอย่างดังนี้

ตัวอย่างที่ 13.1 การใช้คำสอบถามเพื่อเรียกดูรายชื่อฐานข้อมูลทั้งหมด

```
mysql> SHOW DATABASES;
```

Database
mysql
test
parinya_db

13.1.2 USE;

USE เป็นคำสั่งที่ใช้สำหรับเลือกใช้งานฐานข้อมูล มีตัวอย่างดังนี้

รูปแบบ

USE ชื่อฐานข้อมูล;

ตัวอย่างที่ 13.2 ตัวอย่างการเลือกใช้งานฐานข้อมูลชื่อ parinya_db

```
mysql>USE parinya_db;
```

13.1.3 SHOW TABLES;

SHOW TABLES เป็นคำสອบถາມเพื่อเรียกดูรายชื่อตารางทั้งหมดที่มีอยู่ในฐานข้อมูล ที่ถูกเลือกไว้ก่อนหน้านี้ มีตัวอย่างดังนี้

ตัวอย่างที่ 13.3 การใช้คำสອบถາມ SHOW TABLES;

```
mysql> USE parinya_db;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_parinya_db |
+-----+
| question_suit1
| question_user
+-----+
```

1 เลือกใช้งานฐานข้อมูล parinya_db
2 คำสອบถາມ SHOW TABLES;
3 แสดงรายชื่อตารางในฐานข้อมูล parinya_db

13.1.4 DESCRIBE หรือ DESC

DESCRIBE หรือ DESC เป็นคำสອบถາມที่ใช้สำหรับดูโครงสร้างของขอบเขตข้อมูล (Field) และข้อมูล ที่มีอยู่ในตารางที่ระบุ มีรูปแบบการใช้งานดังนี้

รูปแบบ

DESCRIBE ชื่อตาราง;

หรือ

DESC ชื่อตาราง;

ตัวอย่างที่ 13.4 การใช้คำสوبถาม DESCRIBE หรือ DESC

```
mysql> DESCRIBE question_user;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
user_id	varchar(30)	NO		NULL	
password	varchar(30)	NO		NULL	
before_name	varchar(20)	YES		NULL	
name	varchar(60)	YES		NULL	
lastname	varchar(60)	YES		NULL	
privilege	varchar(1)	NO		3	

13.1.5 DROP

DROP เป็นคำสั่งที่ใช้สำหรับลบฐานข้อมูล หรือตาราง มีรูปแบบดังนี้

รูปแบบ

DROP ชื่อตาราง; หรือ ชื่อฐานข้อมูล;

หากลบฐานข้อมูลออกจะทำให้ทุกๆ ตารางที่อยู่ในฐานข้อมูลนั้นๆ ถูกลบออกไปด้วย ตัวอย่างดังนี้

ตัวอย่างที่ 13.5 การใช้คำสั่ง DROP เพื่อลบฐานข้อมูล parinya_db

```
mysql>DROP parinya_db;
```

13.1.6 QUIT หรือ EXIT

QUIT หรือ EXIT เป็นคำสั่งที่ใช้สำหรับออกจากโปรแกรม MySQL ตัวอย่างดังนี้

ตัวอย่างที่ 13.6 การใช้คำสั่ง QUIT หรือ EXIT เพื่ออกจากระบบฐานข้อมูล MySQL

```
mysql>EXIT; หรือ QUIT
```

13.2 องค์ประกอบของฐานข้อมูล

โดยทั่วไปแล้ว องค์ประกอบหลักของระบบฐานข้อมูล MySQL จะมีลักษณะดังต่อไปนี้

13.2.1 ฐานข้อมูล (Database) ในการที่จะจัดเก็บข้อมูลใน MySQL ได้ จะต้องเริ่มต้นที่การสร้าง "ฐานข้อมูล" ก่อน ฐานข้อมูลนี้อาจเปรียบได้กับโฟลเดอร์ที่อยู่ในไดเรกทอรี



13.2.2 ตาราง (Table) คือ ตาราง มีลักษณะการแยกจัดเก็บข้อมูล ในแต่ละเรื่องออกจากกัน เช่น ตารางพนักงาน ตารางสินค้า ตารางลูกค้า เป็นต้น ทั้งนี้ตารางอาจเปรียบได้กับไฟล์ต่างๆ ที่อยู่ในโฟลเดอร์ นั่นเอง และฐานข้อมูลหนึ่งๆ จะมีตารางจำนวนเท่าไหร่ก็ได้ (ขึ้นอยู่กับการวิเคราะห์ความสมั้นธ์ของข้อมูล ภายในระบบที่จะพัฒนา) เช่นเดียวกับโฟลเดอร์หนึ่งๆ จะมีไฟล์อยู่จำนวนเท่าไหร่ก็ได้

13.2.3 ขอบเขตข้อมูล ภายในตารางจะประกอบไปด้วยขอบเขตข้อมูลต่างๆ เพื่อกำหนดว่า ตารางนั้นจะเก็บข้อมูลอะไรบ้าง โดยปกติแล้วข้อมูลที่จัดเก็บในตารางเดียวกัน จะต้องมีความเกี่ยวข้อง อย่างโดยอย่างหนึ่งต่อกัน เช่น หากเป็นตารางที่จัดเก็บข้อมูลของพนักงาน ก็อาจประกอบไปด้วยขอบเขต ข้อมูลรหัส ชื่อ ที่อยู่ ตำแหน่ง และเงินเดือน เป็นต้น นอกจากนี้แล้ว แต่ละขอบเขตข้อมูลจะต้องระบุชนิด ข้อมูลให้สมพันธ์กับข้อมูลที่จะจัดเก็บในขอบเขตข้อมูลนั้นด้วย

13.3 ชนิดข้อมูลสำหรับกำหนดให้กับขอบเขตข้อมูล

เนื่องจากขอบเขตข้อมูลใช้สำหรับเก็บข้อมูลในแต่ละเรื่อง โดยข้อมูลเหล่านี้อาจจะมีหลายชนิด แตกต่างกันออกไป เช่น ชื่อ หรือ ที่อยู่ เป็นข้อมูลชนิดข้อความ และเงินเดือนมักจะเป็นชนิดตัวเลข และถ้า เป็นวันเกิดมักจะเป็นข้อมูลชนิดวันเวลา เป็นต้น ทั้งนี้เนื่องจากวิธีในการจัดการกับข้อมูลเหล่านี้ จะ ต่างกันออกไปตามลักษณะของข้อมูล ดังนั้นทุกขอบเขตข้อมูลของตาราง จะต้องกำหนดชนิดข้อมูลให้กับ ขอบเขตข้อมูลเสมอ สำหรับ MySQL ได้แบ่งชนิดข้อมูลของขอบเขตข้อมูล ดังนี้

13.3.1 ข้อมูลชนิดตัวเลข

ข้อมูลชนิดตัวเลขนั้นมีอยู่หลายชนิดตามขนาดของตัวเลข และในแต่ละชนิดสามารถ กำหนดเพิ่มเติมเป็นจำนวนบวกหรือลบ (SIGNED) หรือเป็นได้เฉพาะจำนวนบวกอย่างเดียว (UNSIGNED) ดังตารางต่อไปนี้

ตารางที่ 13.1 แสดงชนิดข้อมูลของ MySQL ชนิดตัวเลข

ชนิด	ไบต์	รองรับ	ค่าต่ำสุด	ค่าสูงสุด
TINYINT	1	SIGNED	-128	127
		UNSIGNED	0	255
SMALLINT	2	SIGNED	-32768	32767
		UNSIGNED	0	65535
MEDIUMINT	3	SIGNED	-8388608	8388067
		UNSIGNED	0	16777215
INT	4	SIGNED	-2147483648	2147483647
		UNSIGNED	0	4294967295
BIGINT	8	SIGNED	-9223372036854775808	9223372036854775807
		UNSIGNED	0	18446744073709551615

ตารางที่ 13.1 (ต่อ)

ชนิด	ไบต์	รองรับ	ค่าต่ำสุด	ค่าสูงสุด
FLOAT	4	SIGNED	-3.402823466E+38	-1.1754966E-38
		UNSIGNED	1.7976931348632157E+38	3.402823466+38
DOUBLE	8	SIGNED	-1.7976931348623157E+308	-2.2250738585072014E-308
		UNSIGNED	2.225073858072014E-308	1.7976931348623157E+308

13.3.2 ข้อมูลชนิดข้อความ

ข้อมูลชนิดข้อความ ที่กำหนดได้ ดังตารางที่ 13.2 ทั้งนี้ข้อมูลชนิด CHAR และ VARCHAR จะต้องระบุความยาวสูงสุดของข้อความลงไปด้วย กำหนดได้ไม่เกิน 255 เช่น CHAR(50) หรือ VARCHAR(250) เป็นต้น ส่วนชนิดอื่นๆ นอกจากนี้ ไม่จำเป็นต้องระบุค่าความยาวของข้อความ

ตารางที่ 13.2 แสดงชนิดข้อมูลของ MySQL ชนิดข้อความ

ชนิด	ค่าสูงสุด
CHAR (length)	255
VARCHAR (length)	255
YINYTEXT	255
TEXT	65535
MEDIUMTEXT	16777215
LONGTEXT	4294967295

13.3.3 ข้อมูลชนิด BLOB (Binary Large Object)

ข้อมูลชนิด BLOB มักใช้กับข้อมูลประเภทรูปภาพ หรือมัลติมีเดีย มีลักษณะเป็นข้อมูลแบบไบナรี มีรายละเอียดดังตารางที่ 13.3

ตารางที่ 13.3 แสดงชนิดข้อมูลของ MySQL ชนิด BLOB

ชนิด	ค่าสูงสุด
TINYBLOB	255
BLOB	65535
MEDIUMBLOB	16777215
LONGBLOB	4294967295

13.3.4 ข้อมูลชนิด SET และ ENUM

ข้อมูลชนิด SET และ ENUM คือ กลุ่มของข้อมูลที่ trabค่าที่แน่นอนแล้วจำนวนหนึ่ง ข้อมูลที่จะกำหนดให้กับขอบเขตข้อมูลชนิดนี้ได้ต้องเป็นข้อมูลที่เป็นสมาชิกของ SET หรือ ENUM เท่านั้น



โดยข้อมูลชนิด SET จะมีสมาชิกได้ไม่เกิน 64 ตัว และ ENUM มีสมาชิกได้ไม่เกิน 65,535 ตัว รูปแบบของขอบเขตข้อมูลประเภทนี้ ดังตารางที่ 13.4

ตารางที่ 13.4 แสดงชนิดข้อมูลของ MySQL ชนิด SET และ ENUM

ชนิด	รูปแบบ	ค่าสูงสุด
SET	SET ("member1", "member2",..., "member64")	64
ENUM	ENUM ("member1", "member2",..., "member65535")	65535

13.3.5 ข้อมูลชนิดวันเวลา

โดยปกติข้อมูลชนิดวันเวลาของ MySQL จะอยู่ในรูปแบบต่อไปนี้

YYYY-MM-DD หรือ Year-Month-Date

เช่น วันที่ 20 เดือนตุลาคม ปี ค.ศ. 2010 จะต้องกำหนดเป็น 2010-10-20

ตารางที่ 13.5 แสดงชนิดข้อมูลของ MySQL ชนิดวันเวลา

ชนิด	รายละเอียด	ค่าสูงสุด
DATE	เฉพาะข้อมูลวันเดือนปี	1000 01 01 ถึง 9999 12 31
TIME	เฉพาะข้อมูลเวลา	-838:59:59 ถึง 838:59:59
DATETIME	ทั้งวันเดือนปีและเวลา	1000-01-01 00:00:00 ถึง 9999-12-31 23-59-59

13.4 แออททริบิวต์สำหรับกำหนดให้กับขอบเขตข้อมูล

แออททริบิวต์ (Attribute) หรือแฟล็ก (Flag) คือ ข้อกำหนดเพิ่มเติมนอกเหนือจากชนิดข้อมูลที่กำหนดให้กับขอบเขตข้อมูล แออททริบิวต์จะช่วยให้จัดการข้อมูลได้สะดวกขึ้น และบางแออททริบิวต์ก็ยังช่วยตรวจสอบความถูกต้องของข้อมูลได้ด้วย แต่ต้องกำหนดให้สัมพันธ์กับชนิดข้อมูลของขอบเขตข้อมูล แออททริบิวต์ของขอบเขตข้อมูล ดังตารางที่ 13.6

ตารางที่ 13.6 แสดงแออททริบิวต์สำหรับกำหนดให้กับขอบเขตข้อมูลใน MySQL

แออททริบิวต์	รายละเอียด
NOT NULL	ห้ามขอบเขตข้อมูลเป็นค่าว่าง (NULL) คือ จะต้องใส่ข้อมูลให้กับขอบเขตข้อมูลเสมอ นิยมใช้กำหนดเพื่อควบคุมความถูกต้องของตาราง
BINARY	ใช้กับข้อมูลชนิด CHAR หรือ VARCHAR โดยปกติแล้วการจัดเรียงข้อมูลชนิด CHAR หรือ VARCHAR จะเป็นแบบ case-sensitive (ตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่มีความหมายแตกต่างกัน) แต่หากระบุ "แออททริบิวต์" เป็น BINARY การสืบค้นจะไม่คำนึงว่าจะเป็นตัวอักษรพิมพ์เล็กหรือพิมพ์ใหญ่

ตารางที่ 13.6 (ต่อ)

แอ็ตทริบิวต์	รายละเอียด
AUTO_INCREMENT	กำหนดให้ตัวเลขลงไปในขอบเขตข้อมูลแบบอัตโนมัติ ตัวเลขที่ใส่เข้าไปโดยปกติจะเริ่มจาก 1 แล้วเพิ่มค่าครั้งละ 1 ไปเรื่อยๆ ค่าที่โปรแกรมใส่ให้จะไม่สามารถเปลี่ยนแปลงแก้ไขได้ ทั้งนี้ขอบเขตข้อมูลที่จะกำหนดให้เป็นแบบ AUTO_INCREMENT ได้ ต้องกำหนดชนิดข้อมูลให้เป็นแบบเลขจำนวนเต็ม เช่น INT เป็นต้น และห้ามใช้ร่วมกับแอ็ตทริบิวต์ NULL แต่ส่วนมากนิยมใช้ร่วมกับ NOT NULL
DEFAULT	เป็นการกำหนดค่าให้กับขอบเขตข้อมูล ด้วยค่าใดค่าหนึ่งเอาไว้ล่วงหน้า มักเป็นค่าที่ใช้กันบ่อยๆ เช่น พนักงานส่วนใหญ่เป็นเพศหญิง ดังนั้นจึงอาจกำหนดค่า DEFAULT ให้แก่ขอบเขตข้อมูลเพศ เป็น 'หญิง' เป็นต้น
INDEX	ใช้ในการจัดเรียงลำดับข้อมูลจากน้อยไปมาก หรือมากไปหนาแน่น ข้อมูลที่เรียงลำดับจะช่วยให้ค้นหารวดเร็วขึ้น สามารถจัดเรียงลำดับข้อมูลได้มากกว่า 1 ขอบเขตข้อมูล เมื่อสร้าง INDEX โปรแกรมฐานข้อมูลจะสร้างไฟล์ INDEX แยกออกไปต่างหาก เนื่องจากเรื่อง INDEX ถือว่าเป็นเรื่องสำคัญอย่างหนึ่งของฐานข้อมูล
UNIQUE	เป็น INDEX ในอีกลักษณะหนึ่ง โดยมีข้อกำหนดที่สำคัญ คือ ห้ามมีข้อมูลซ้ำกันในขอบเขตข้อมูลเดียวกัน ใช้กับข้อมูลที่คาดว่าข้อมูลในแต่ละแถวจะไม่มีทางซ้ำกันได้เลย เช่น รหัสพนักงาน อีเมล เป็นต้น และถ้าใส่ข้อมูลซ้ำกับที่มีอยู่แล้ว โปรแกรมจะไม่ยอมรับข้อมูลนั้น ดังนั้นแอ็ตทริบิวต์ UNIQUE สามารถนำมาช่วยในการตรวจสอบความถูกต้องได้อีกด้วยหนึ่ง
PRIMARY KEY	เป็น INDEX ในอีกลักษณะหนึ่ง โดย PRIMARY KEY คือ ขอบเขตข้อมูลที่จะใช้เป็นขอบเขตข้อมูลหลักของตาราง แต่ PRIMARY KEY มีข้อกำหนดที่สำคัญคือ ขอบเขตข้อมูลที่จะใช้เป็น PRIMARY KEY ข้อมูลในขอบเขตข้อมูลนั้นจะต้องไม่ซ้ำกัน โดยทั่วไปทุกๆ ตารางจะต้องมีขอบเขตข้อมูลอย่างน้อย 1 ขอบเขตข้อมูล ที่กำหนดเป็น PRIMARY KEY

13.5 การสร้างฐานข้อมูล

ขั้นตอนการสร้างฐานข้อมูลเป็นขั้นตอนแรกสุดก่อนเริ่มขั้นตอนต่อๆ ไป การสร้างฐานข้อมูลมีรูปแบบดังนี้

รูปแบบ

CREATE DATABASE (IF NOT EXISTS) ชื่อฐานข้อมูล;



ปริญญา น้อยดอนไพร || การเขียนโปรแกรมบนเว็บด้วย PHP ร่วมกับฐานข้อมูล MySQL

หมายเหตุ

IF NOT EXISTS ใช้ป้องกันการสร้างชื่อฐานข้อมูลซ้ำกัน ความหมายคือ ระบบจะไม่สร้างฐานข้อมูลใหม่ให้ หากชื่อฐานข้อมูลใหม่ไปตรงกับชื่อฐานข้อมูลเดิมที่มีอยู่ (จะระบุหรือไม่ก็ได้)

ตัวอย่างที่ 13.7 การใช้ CREATE DATABASE เพื่อสร้างฐานข้อมูลใหม่ดังนี้

```
mysql> CREATE DATABASE guestbook;
```

จากตัวอย่างที่ 13.7 เป็นการสร้างฐานข้อมูลชื่อ guestbook โดยฐานข้อมูลที่สร้างขึ้นนี้ใช้เพื่อเก็บตารางต่างๆ ที่เกี่ยวข้อง สามารถตรวจสอบได้ว่าฐานข้อมูลได้ถูกสร้างแล้วหรือไม่ โดยใช้คำสบสอบตาม SHOW DATABASES ก่อนหน้าที่ได้อธิบายไปแล้ว หรือใช้ IF NOT EXISTS ร่วมด้วย

13.6 การสร้างตาราง

การสร้างตารางใหม่จะใช้คำสั่งเพื่อกำหนดโครงสร้างของตาราง ตารางใหม่ที่จะสร้างนี้ จะถูกสร้างไว้ในฐานข้อมูลที่ได้เลือกใช้ไว้ก่อนหน้านี้ โดยมีกฎการตั้งชื่อตารางที่สำคัญ คือ อนุญาตให้ตั้งชื่อที่มีความยาวได้ไม่เกิน 64 ตัวอักษร และห้ามใช้อักษร "/" หรือ "." ในชื่อตาราง เพราะอาจเกิดข้อผิดพลาดในการอ้างถึงตาราง ตารางจะมีโครงสร้างประกอบด้วย ชื่อตาราง ชื่อขอบเขตข้อมูล ชนิดข้อมูล ขนาด และแอททริบิวต์เพิ่มเติมอื่นๆ มีรายละเอียดดังนี้

ตารางที่ 13.7 ตัวอย่างโครงสร้างตาราง user

ขอบเขต ข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า บริยาย	คีย์	หมายเหตุ
name	VARCHAR(40)	-	-	-	ชื่อผู้ใช้งาน
address	VARCHAR(40)	-	-	-	ที่อยู่
email	VARCHAR(40)	-	-	-	อีเมล
url	VARCHAR(40)	-	-	-	ที่อยู่ URL
comment	TEXT	-	-	-	หมายเหตุ

ตัวอย่างที่ 13.7 ตัวอย่างการสร้างตาราง user ในฐานข้อมูล guestbook

```
mysql> USE guestbook;
Database changed
mysql> CREATE TABLE user (
    -> name VARCHAR (40),
    -> address VARCHAR (40),
    -> email VARCHAR (40),
    -> url VARCHAR (40),
    -> comment TEXT);
```

จากตัวอย่างที่ 13.7 แสดงตัวอย่างการสร้างตาราง user ในฐานข้อมูล guestbook ประกอบด้วย 5 ขอบเขตข้อมูล ประกอบด้วย name, address, email, url และ comment นอกจากนี้ตารางยังสามารถสร้างได้หลายวิธีและสามารถกำหนดแอ็ตทริบิวต์ให้กับขอบเขตข้อมูลได้มากกว่า 1 แอ็ตทริบิวต์ มีตัวอย่างดังนี้

ตารางที่ 13.8 ตัวอย่างโครงสร้างตาราง employees

ขอบเขต ข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
emp_id	INT	NOT NULL	-	PK	รหัสพนักงาน
firstname	VARCHAR(30)	NOT NULL	-	-	ชื่อพนักงาน
lastname	VARCHAR(30)	NOT NULL	-	-	นามสกุลพนักงาน
position	VARCHAR(20)	-	-	-	ตำแหน่งงาน
salary	MEDIUMINT	-	-	-	เงินเดือน (UNSIGNED)
address	TEXT	-	-	-	ที่อยู่
email	VARCHAR(40)	NOT NULL	-	-	อีเมล์ (UNIQUE)
phone	VARCHAR(30)	NOT NULL	-	-	เบอร์โทรศัพท์ (UNIQUE)

ตัวอย่างที่ 13.8 การสร้างตารางและกำหนดแอ็ตทริบิวต์ขอบเขตข้อมูลมากกว่า 1 แอ็ตทริบิวต์

```
mysql> CREATE TABLE employees (
    -> emp_id      INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    -> firstname   VARCHAR (30) NOT NULL,
    -> lastname    VARCHAR (30) NOT NULL,
    -> position    VARCHAR (20),
    -> salary      MEDIUMINT UNSIGNED,
    -> address     TEXT,
    -> email       VARCHAR (40) NOT NULL UNIQUE,
    -> phone       VARCHAR (30) NOT NULL UNIQUE);
```

13.6.1 การกำหนด PRIMARY KEY

PRIMARY KEY คือ ขอบเขตข้อมูลที่ใช้เป็นขอบเขตข้อมูลหลักของตาราง โดยมีข้อกำหนดที่สำคัญ คือ ขอบเขตข้อมูลที่จะใช้เป็น PRIMARY KEY ต้องไม่มีข้อมูลที่ซ้ำกันเลย แต่กรณีที่ขอบเขตข้อมูลที่ต้องการใช้เป็น PRIMARY KEY มีโอกาสที่ข้อมูลจะซ้ำกัน ก็สามารถเลือกมากกว่า 1 ขอบเขตข้อมูล มาเป็น PRIMARY KEY ร่วมกันได้ เช่น สมมติตารางพนักงานประกอบด้วยขอบเขตข้อมูล "firstname" และ "lastname" หากคาดว่าชื่อของพนักงานอาจซ้ำกันได้ ดังนั้นจึงไม่สามารถใช้ขอบเขตข้อมูล "firstname" เป็น PRIMARY KEY เพียงขอบเขตข้อมูลเดียวได้ จึงอาจเลือกใช้ทั้ง firstname และ lastname เป็น PRIMARY KEY ร่วมกัน



กรณีที่ต้องใช้หลายขอบเขตข้อมูลเป็น PRIMARY KEY ร่วมกัน จึงไม่สามารถระบุ
แอ็คทริบิวต์ PRIMARY KEY แบบแยกขอบเขตข้อมูลได้ จะต้องกำหนดในรูปแบบของฟังก์ชันรูปแบบดังนี้
รูปแบบ

PRIMARY KEY (column1, column2,...)

โดยการระบุต่อจากขอบเขตข้อมูลสุดท้ายของตารางตามรูปแบบที่ได้กล่าวมาแล้ว เช่น

ตัวอย่างที่ 13.9 การระบุ PRIMARY KEY ให้กับขอบเขตข้อมูลของตาราง employees

```
mysql>CREATE TABLE employees (
    -> firstname  VARCHAR (30) NOT NULL,
    -> lastname   VARCHAR (30) NOT NULL,
    -> position   VARCHAR (20),
    -> salary     MEDIUMINT UNSIGNED,
    -> address    TEXT,
    -> email      VARCHAR (40) NOT NULL UNIQUE,
    -> phone      VARCHAR (30) NOT NULL UNIQUE
    -> PRIMARY KEY (firstname, lastname));
```

จากตัวอย่างที่ 13.9 แสดงตัวอย่างการสร้างตาราง โดยระบุ PRIMARY KEY ให้กับ
ขอบเขตข้อมูลของตาราง employees มากกว่า 1 ขอบเขตข้อมูล จากตัวอย่างกำหนดให้ขอบเขตข้อมูล
firstname และ lastname เป็น PRIMARY KEY

13.6.2 ขอบเขตข้อมูลแบบ UNIQUE

จากข้อกำหนดของ UNIQUE คือ ข้อมูลในขอบเขตข้อมูลนั้นจะต้องไม่ซ้ำกันเลย สามารถ
สร้างขอบเขตข้อมูลแบบ UNIQUE ได้หลายขอบเขตข้อมูล และแต่ละขอบเขตข้อมูลจะเป็นอิสระจากกัน
(ต่างจาก PRIMARY KEY ที่จะนำขอบเขตข้อมูลที่เป็น PRIMARY KEY มาพิจารณาร่วมกัน) นั่นคือ ไม่ว่า
จะสร้างขอบเขตข้อมูลแบบ UNIQUE ขึ้นมากี่ขอบเขตข้อมูลก็ตาม ขอบเขตข้อมูลเหล่านั้น ในขอบเขต
ข้อมูลเดียวกันต้องไม่มีข้อมูลที่ซ้ำกันเลย ดังนั้นการสร้าง UNIQUE แบบระบุแยกขอบเขตข้อมูล หรือจะ
สร้างรวมด้วยฟังก์ชัน UNIQUE () ก็ให้ผลที่ไม่แตกต่างกัน (เปลี่ยนแปลงคำสั่งจากตัวอย่างที่ 13.9 โดยใช้
ร่วมกับฟังก์ชัน UNIQUE ()) ตัวอย่างดังนี้

ตัวอย่างที่ 13.10 การระบุ UNIQUE ให้กับขอบเขตข้อมูลของตาราง employees

```
mysql>CREATE TABLE employees (
    -> firstname  VARCHAR (30) NOT NULL,
    -> lastname   VARCHAR (30) NOT NULL,
    -> position   VARCHAR (20),
    -> salary     MEDIUMINT UNSIGNED,
```

```

-> address    TEXT,
-> email      VARCHAR (40) NOT NULL,
-> phone      VARCHAR (30) NOT NULL,
-> UNIQUE (email, phone));

```

13.6.3 การสร้าง INDEX

INDEX เป็นการจัดเรียงลำดับข้อมูลเพื่อให้สามารถค้นหาข้อมูลได้รวดเร็วยิ่งขึ้นโดย MySQL จะสร้างไฟล์ INDEX แยกจากข้อมูลไว้ต่างหาก การเรียงลำดับข้อมูลสามารถเรียงได้ทั้งแบบจากน้อยไปมาก หรือถ้าเป็นตัวอักษรคือ เรียงจาก A ไป Z โดยเรียนแนนด้วย ASC และเรียงจากมากไปหาน้อย หรือจาก Z ไป A เรียนแนนด้วย DESC สำหรับค่าโดยปริยาย คือ ASC ทั้งนี้สามารถสร้าง INDEX ให้กับขอบเขตข้อมูลใดขอบเขตข้อมูลหนึ่ง หรือกำหนดให้กับหลายๆ ขอบเขตข้อมูลได้ เช่นเดียวกับ UNIQUE หรือ PRIMARY KEY ด้วยการกำหนดแบบฟังก์ชัน ดังตัวอย่างที่ 13.11

ตัวอย่างที่ 13.11 การระบุ INDEX ให้กับขอบเขตข้อมูลของตาราง employees

```

mysql>CREATE TABLE employees (
-> firstname  VARCHAR (30) NOT NULL,
-> lastname   VARCHAR (30) NOT NULL,
-> position   VARCHAR (20),
-> salary     MEDIUMINT UNSIGNED,
-> address    TEXT,
-> email      VARCHAR (40) NOT NULL,
-> phone      VARCHAR (30) NOT NULL,
-> INDEX (email, phone));

```

13.6.4 การกำหนดหลายๆ แอ็ตทริบิวต์แบบฟังก์ชัน

หากต้องการกำหนดหลายๆ แอ็ตทริบิวต์แบบฟังก์ชัน ก็ให้วางต่อท้ายขอบเขตข้อมูล สุดท้าย โดยเรียงลำดับอย่างไรก็ได้ ตัวอย่างดังนี้

ตัวอย่างที่ 13.12 การกำหนดหลายๆ แอ็ตทริบิวต์แบบฟังก์ชัน

```

mysql>CREATE TABLE employees (
-> emp_id     INT NOT NULL ,
-> firstname   VARCHAR (30) ,
-> lastname    VARCHAR (30) ,
-> position    VARCHAR (20),
-> salary      MEDIUMINT UNSIGNED,

```



```
-> address      TEXT,
-> email        VARCHAR (40) NOT NULL,
-> phone        VARCHAR (30) NOT NULL,
-> PRIMARY KEY (emp_id) , INDEX (firstname, lastname) , UNIQUE (email, phone);
```

13.6.5 การสร้างตารางใหม่จากตารางเดิมที่มีอยู่

การสร้างตาราง อาจสร้างโดยการคัดลอกมาจากตารางที่มีอยู่ก่อนหน้า ถ้าคัดลอกมาทั้งหมดก็จะทำให้ตารางใหม่มีลักษณะเหมือนกับตารางต้นแบบทุกประการ หรืออาจเลือกเพียงบางขอบเขตข้อมูลก็ได้ รูปแบบการคัดลอกตารางดังนี้

รูปแบบ

```
CREATE TABLE ชื่อตารางใหม่
```

```
SELECT ขอบเขตข้อมูล_1, ขอบเขตข้อมูล_2, ..., ขอบเขตข้อมูล_N
```

```
FROM ชื่อตารางเดิม
```

```
[WHERE เงื่อนไข]
```

SELECT...FROM เป็นคำสอบถามสำหรับการคัดเลือกขอบเขตข้อมูลที่ต้องการ หากต้องการคัดลอกทุกขอบเขตข้อมูล อาจใช้เครื่องหมาย * แทนการระบุชื่อขอบเขตข้อมูล ตัวอย่างดังนี้

ตัวอย่างที่ 13.13 การใช้คำสั่ง CREATE TABLE ร่วมกับคำสอบถาม SELECT...FROM

```
mysql>CREATE TABLE employee2 SELECT * FROM employees;
```

ตาราง employees

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

```
CREATE TABLE employee2 SELECT * FROM employees;
```

ตาราง employees2

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

ภาพที่ 13.1 แสดงลักษณะการทำงานและผลลัพธ์เมื่อใช้ CREATE TABLE ร่วมกับ SELECT...FROM

จากตัวอย่างที่ 13.13 เป็นการใช้คำสั่ง CREATE TABLE ร่วมกับคำสอบถาม SELECT...FROM เพื่อสร้างตารางใหม่ ชื่อ employee2 โดยคัดเลือกขอบเขตข้อมูลและข้อมูลทั้งหมดจากตาราง employees แสดงผลลัพธ์ที่ได้ ดังภาพที่ 13.1

หากต้องการคัดเลือกเพียงบางขอบเขตข้อมูล ก็ให้ระบุชื่อขอบเขตข้อมูลลงไป ดังนี้

```
mysql>CREATE TABLE employee2 SELECT firstname, lastname FROM employees;
```

ปกติหากไม่ระบุเงื่อนไข ถ้าตารางต้นแบบมีข้อมูลอยู่แล้ว ข้อมูลนั้นก็จะถูกคัดลอกมาทั้งหมดทุกແล้า แต่สามารถกำหนดเงื่อนไขเพื่อคัดลอกเฉพาะข้อมูลที่ต้องการได้ เช่น

ตัวอย่างที่ 13.14 การใช้ CREATE TABLE ร่วมกับ SELECT...FROM แบบมีเงื่อนไข

```
mysql>CREATE TABLE emp
```

```
-> SELECT firstname, lastname, position FROM employees  
-> WHERE position = "Programmer";
```

ตาราง employees

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

```
CREATE TABLE emp SELECT firstname, lastname, position FROM employees WHERE  
position = "Programmer";
```

ตาราง emp

firstname	lastname	position
Jaidee	Ruktum	Programmer
Dennis	Robert	Programmer

ภาพที่ 13.2 แสดงลักษณะการทำงานและผลลัพธ์เมื่อใช้การใช้ CREATE TABLE ร่วมกับ SELECT ...FROM แบบมีเงื่อนไข

13.7 การแทรก ปรับปรุง ลบ และการเรียกดูข้อมูลในตาราง

13.7.1 การแทรกข้อมูลในตารางด้วย INSERT

การแทรกข้อมูลลงในตารางด้วย INSERT มีรูปแบบดังนี้

รูปแบบ

```
INSERT INTO ชื่อตาราง [ขอบเขตข้อมูล1, ขอบเขตข้อมูล2,... ]) VALUES (ค่า1, ค่า2, ...);
```



ชื่อขอบเขตข้อมูล จะกำหนดหรือไม่ก็ได้ โดยหากกำหนด จะต้องระบุค่าเรียงตามลำดับการจัดเรียงของขอบเขตข้อมูลหากไม่กำหนด จะต้องระบุค่าให้ครบถ้วนของขอบเขตข้อมูล และเรียงตามลำดับการจัดเรียงของขอบเขตข้อมูลในตาราง

ตัวอย่างที่ 13.15 การแทรกข้อมูลในตารางด้วย INSERT

```
mysql>INSERT INTO employees VALUES (123, "John", "john@example.com");
```

จากรูปแบบของ INSERT ที่กล่าวมานี้ สามารถใส่ข้อมูลได้เพียงครั้งละ 1 แถว หากจะแทรกข้อมูลมากกว่า 1 แถว ก็ต้องใช้ INSERT ซ้ำๆ ในทุกแถว อาจจะดูข้อมูลไม่สะดวกนัก แต่ใน MySQL ยังมีวิธีที่สามารถแทรกข้อมูลพร้อมกันครั้งละหลายๆ แถว โดย INSERT เพียงครั้งเดียว มีรูปแบบดังนี้

INSERT INTO ชื่อตาราง VALUES

(ค่าขอบเขตข้อมูล_1_ ของแถวที่_1, ค่าของขอบเขตข้อมูล_2_ ของแถวที่_1,...) ,

(ค่าขอบเขตข้อมูล_1_ ของแถวที่_2, ค่าของขอบเขตข้อมูล_2_ ของแถวที่_2,...) ,

....

....

....

(ค่าขอบเขตข้อมูล_N_ ของแถวที่_N, ค่าของขอบเขตข้อมูล_N_ ของแถวที่_N,...);

ตัวอย่างที่ 13.16 การแทรกข้อมูลในตารางด้วย INSERT แบบหลายระเบียนในครั้งเดียว

```
mysql>INSERT INTO employees VALUES
```

```
-> (123, "John", "Lee", "Finance", "20000", "Suratthani", "jo@sru.ac.th" , "077355890") ,
```

```
-> (456, "Jim", "Murray", "Staff", "17000", "Suratthani", "jim@sru.ac.th" , "077355000") ,
```

```
-> (789, "Jane", "Nuch", "Staff", "19000", "Chumphon", "jan@sru.ac.th" , "077355339");
```

13.7.2 การปรับปรุงข้อมูลในตารางด้วย UPDATE

การปรับปรุงข้อมูลในตารางด้วย UPDATE สามารถทำการปรับปรุงข้อมูลได้หลายขอบเขตข้อมูล และหลายระเบียนภายในคำสั่ง 1 คำสั่ง ทั้งนี้ขึ้นอยู่กับเงื่อนไข ซึ่งกำหนดไว้ใน WHERE การปรับปรุงข้อมูลลงในตารางจะต้อง โดยใช้ UPDATE มีรูปแบบโดยทั่วไปดังนี้

รูปแบบ

UPDATE ชื่อตาราง SET ขอบเขตข้อมูล1=การคำนวนหรือค่าใหม่ที่ต้องการกำหนด1,

ขอบเขตข้อมูล2=การคำนวนหรือค่าใหม่ที่ต้องการกำหนด2,

... ,

... ,

ขอบเขตข้อมูล N = การคำนวนหรือค่าใหม่ที่ต้องการกำหนด N
[WHERE เงื่อนไข] [LIMIT กำหนดแถวเริ่มต้นและจำนวนแถวที่ต้องการปรับปรุง];

ตัวอย่างที่ 13.17 การปรับปรุงข้อมูลในตาราง employees ด้วย UPDATE

```
mysql> UPDATE employees SET address='Chumphon',
->     position='President'
-> WHERE emp_id=001;
```

จากตัวอย่างเป็นการปรับปรุงข้อมูลในตาราง employees โดยกำหนดให้ขอบเขตข้อมูล address มีค่าเท่ากับ 'Chumphon' และ position มีค่าเท่ากับ 'President' โดยมีเงื่อนไขว่าจะปรับปรุงข้อมูลก็ต่อเมื่อ ขอบเขตข้อมูล emp_id มีค่าเท่ากับ 001 เท่านั้น

13.7.3 การลบข้อมูลในตารางด้วย DELETE

การลบข้อมูลในตารางด้วย DELETE สามารถทำการลบได้หลายระเบียนภายใน หรือ ระเบียนเดียว ทั้งนี้ขึ้นอยู่กับการกำหนดเงื่อนไข WHERE การลบข้อมูลในตาราง โดยใช้ DELETE มีรูปแบบโดยทั่วไปดังนี้

รูปแบบ

```
DELETE FROM ชื่อตาราง [WHERE เงื่อนไข] [LIMIT แถวเริ่มต้นและจำนวนแถวที่จะลบ];
```

ตัวอย่างที่ 13.18 การลบข้อมูลในตารางด้วย DELETE

```
mysql> DELETE FROM employees WHERE emp_id = 004;
```

จากตัวอย่างเป็นการลบข้อมูลในตารางด้วย DELETE ในตาราง "employees" โดยมีเงื่อนไขการลบ คือ ขอบเขตข้อมูล "emp_id" จะต้องมีค่าเท่ากับ 004 จึงจะลบข้อมูล

13.7.4 การใช้คำสوبatham เพื่อเรียกดูข้อมูลในตารางด้วย SELECT

การใช้คำสوبatham เพื่อเรียกดูข้อมูลในตารางด้วย SELECT สามารถระบุเงื่อนไขต่างๆ ได้ การใช้คำสوبatham SELECT สามารถแบ่งตามลักษณะการใช้งาน มีรูปแบบดังนี้

- 1) การใช้คำสوبatham เพื่อเรียกดูข้อมูลในตารางด้วย SELECT อย่างง่าย

รูปแบบ

```
SELECT ขอบเขตข้อมูล FROM ชื่อตาราง;
```

ขอบเขตข้อมูล คือ ขอบเขตข้อมูลที่ต้องการสوبatham หากมีหลายขอบเขตข้อมูลจะคั่นด้วยเครื่องหมาย Comma หากต้องการสوبatham ข้อมูลทุกขอบเขตข้อมูล สามารถใช้เครื่องหมาย * แทน

ตัวอย่างที่ 13.19 การใช้คำสوبatham เพื่อเรียกดูข้อมูลทั้งหมด จากตาราง employees

```
mysql>SELECT * FROM employees;
```



ผลลัพธ์

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

4 rows in set (0.00 sec)

จากตัวอย่างที่ 13.19 เป็นการใช้คำสบถามเพื่อเรียกดูข้อมูลทั้งหมดจากตาราง employees ผลลัพธ์ที่ได้ มี 4 แถว

ตัวอย่างที่ 13.20 การใช้คำสบถามเพื่อเรียกดูข้อมูลบางรายการ

```
mysql>SELECT emp_id, firstname, position, salary FROM employees;
```

ผลลัพธ์

emp_id	firstname	position	salary
100	Parinya	Manager	45000
200	Jaidee	Programmer	17000
300	Dennis	Programmer	19000
400	John	Finance	18000

4 rows in set (0.00 sec)

2) การใช้คำสบถามเพื่อเรียกดูข้อมูลในตารางด้วย SELECT แบบมีเงื่อนไข

รูปแบบ

```
SELECT ขอบเขตข้อมูล FROM ชื่อตาราง
[WHERE เงื่อนไข]
[GROUP BY รายชื่อขอบเขตข้อมูลที่ใช้ในการกำหนดกลุ่ม]
[HAVING เงื่อนไขที่ใช้ร่วมกับ GROUP BY]
[ORDER BY ชนิดของการเรียงลำดับ]
[LIMIT และเริ่มต้นและจำนวนแถวที่จะดูข้อมูล] ;
```

หมายเหตุ

WHERE ใช้สอบถามข้อมูลแบบมีเงื่อนไข เป็นการระบุค่าเฉพาะข้อมูลที่ต้องการสอบถาม อาจใช้เงื่อนไขเพื่อดึงข้อมูลบางແລ魷จากตารางก็ได้ โดยภายใน WHERE ประกอบด้วยส่วนสำคัญ ได้แก่ ชื่อขอบเขตข้อมูล ได้แก่ ชื่อขอบเขตข้อมูลโอลีเปอร์เรเตอร์ และข้อมูลเฉพาะที่ต้องการแสดงเป็นเงื่อนไขของขอบเขตข้อมูลที่ระบุ

ตารางที่ 13.9 แสดงโอลีเปอร์เรเตอร์ที่ใช้งานร่วมกับ WHERE

โอลีเปอร์เรเตอร์	ชื่อ	ตัวอย่าง
=	เท่ากับ	column = 8000
>	มากกว่า	column > 66.00
<	น้อยกว่า	column < 66.00
>=	มากกว่าหรือเท่ากับ	column >= 66.00
<=	น้อยกว่าหรือเท่ากับ	column <= 66.00
!= หรือ <>	ไม่เท่ากับ	column !=0
IS NULL	ค่าว่าง	column IS NULL
IS NOT NULL	ไม่ใช่ค่าว่าง	column IS NOT NULL
BETWEEN	ค่าที่อยู่ระหว่าง	column BETWEEN 0 AND 66.00
IN	ค่าที่อยู่ในขอบเขตข้อมูลที่กำหนด	column IN ("mai", "mon")
NOT IN	ค่าที่ไม่ได้อยู่ในขอบเขตข้อมูลที่กำหนด	column NOT IN ("mai", "mon")
LIKE	เหมือนกับรูปแบบที่กำหนด	column LIKE ("mai%")
NOT LIKE	ไม่เหมือนกับรูปแบบที่กำหนด	column NOT LIKE ("mai%")
REGEXP	Regular Expression	column REGEXP

ตัวอย่างที่ 13.21 การใช้คำสอบถามเพื่อเรียกดึงข้อมูลในตารางด้วย SELECT แบบมีเงื่อนไข

```
mysql>SELECT * FROM employees WHERE address = "Suratthani";
```

ผลลัพธ์

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443



ตัวอย่างที่ 13.22 การใช้คำสอปถามเพื่อเรียกดูข้อมูลในตารางด้วย SELECT แบบมีเงื่อนไข เป็นช่วงของค่า

```
mysql>SELECT * FROM employees WHERE salary BETWEEN 15000 AND 20000;
```

ผลลัพธ์

emp_id	firstname	lastname	position	salary	address	email	phone
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

3 rows in set (0.00 sec)

ตัวอย่างที่ 13.23 การใช้คำสอปถามเพื่อเรียกดูข้อมูลในตารางด้วย SELECT แบบกำหนดเงื่อนไขด้วย LIKE

```
mysql>SELECT * FROM employees WHERE lastname LIKE 'R%';
```

ผลลัพธ์

emp_id	firstname	lastname	position	salary	address	email	phone
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443

2 rows in set (0.00 sec)

2.1) การใช้ GROUP BY เพื่อค้นหาข้อมูลแล้วแยกเป็นกลุ่มตามเงื่อนไขที่กำหนด สามารถใช้ร่วมกับฟังก์ชัน Aggregate เพื่อหาค่าทางสถิติของกลุ่มข้อมูล มีรายละเอียดดังนี้

ตารางที่ 13.10 แสดงฟังก์ชัน Aggregate สำหรับใช้ร่วมกับ GROUP BY

ฟังก์ชัน	คำอธิบาย
AVG (column)	หาค่าเฉลี่ยของขอบเขตข้อมูล
COUNT (item)	นับจำนวนรายการ
MIN (column) / MAX (column)	หาค่าต่ำสุด / ค่าสูงสุด ในขอบเขตข้อมูล
STD (column) / STDDEV (column)	หาค่า Standard Deviation (ส่วนเบี่ยงเบนมาตรฐาน) ในขอบเขตข้อมูล

ตารางที่ 13.10 (ต่อ)

ฟังก์ชัน	คำอธิบาย
SUM (column)	หาค่าผลรวมของขอบเขตข้อมูล

ตัวอย่างที่ 13.24 การใช้ GROUP BY และฟังก์ชัน COUNT ()

```
mysql>SELECT address, COUNT(address) FROM employees GROUP BY address;
```

จากตัวอย่างที่ 13.24 เป็นการสอบถามข้อมูล โดยใช้ SELECT เพื่อแสดง ขอบเขตข้อมูล "address" และใช้ฟังก์ชัน COUNT () นับขอบเขตข้อมูล "address" จากนั้นใช้ GROUP BY เพื่อหาค่าของข้อมูลตามกลุ่มของขอบเขตข้อมูล "address"

ผลลัพธ์

address	COUNT(address)
Chumphon	1
Suratthani	3

2 rows in set (0.01 sec)

2.2) การใช้ ORDER BY ร่วมกับ SELECT เพื่อเลือกข้อมูลโดยการเรียงลำดับตาม เงื่อนไขที่กำหนดจากน้อยไปมาก (ASC: Ascending) หรือมากไปน้อย (DESC: Descending) มีตัวอย่าง ดังนี้

ตัวอย่างที่ 13.25 การใช้ ORDER BY ร่วมกับ SELECT

```
mysql>SELECT * FROM employees ORDER BY salary DESC;
```

จากตัวอย่างเป็นการใช้ SELECT โดยเรียงลำดับค่าในขอบเขตข้อมูล "salary" จากมากไปน้อย (DESC)

ผลลัพธ์

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123



2.3) การใช้ LIMIT ร่วมกับ SELECT ใช้เพื่อระบุแนวเริ่มต้นและจำนวนแถวที่จะแสดง โดยจะต้องส่งค่าพารามิเตอร์ 2 ค่า คือ ตำแหน่งของแถวเริ่มต้นและจำนวนแถว มีตัวอย่างดังนี้

ตัวอย่างที่ 13.26 การใช้ LIMIT ร่วมกับ SELECT

```
mysql>SELECT * FROM employees LIMIT 2, 2;
```

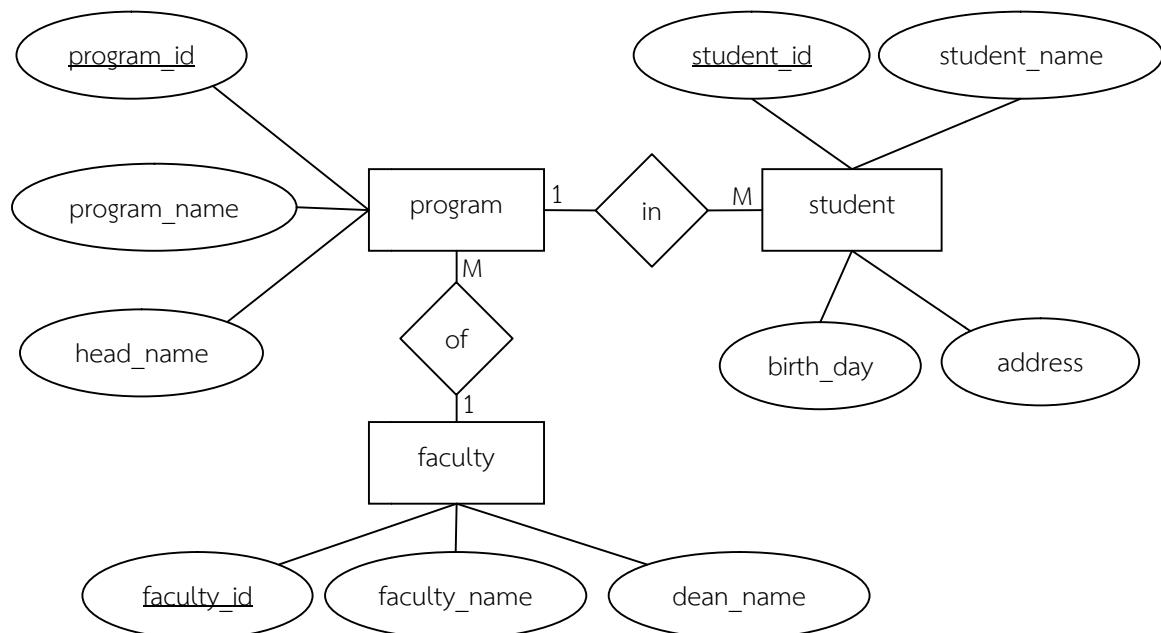
จากตัวอย่างเป็นการใช้ SELECT ร่วมกับ LIMIT โดยเริ่มต้นจากแถวที่ 3 (เริ่มนับแถวแรก คือ 0) และแสดงทั้งหมด 2 แถว

ผลลัพธ์

emp_id	firstname	lastname	position	salary	address	email	phone
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

4 rows in set (0.00 sec)

3) การใช้คำสอบถามเพื่อเรียกดูข้อมูลที่มาจากการทางมากกว่า 1 ตาราง (ตารางที่มีความสัมพันธ์กัน โดยใช้ขอบเขตข้อมูลสำหรับการอ้างอิง) ด้วย SELECT แบบมีเงื่อนไข ดังนี้



ภาพที่ 13.3 แสดงความสัมพันธ์ของข้อมูล ในรูปแบบ ER-Diagram สำหรับเก็บข้อมูลนักศึกษา

จากภาพที่ 13.3 แสดงความสัมพันธ์ของข้อมูล ในรูปแบบ ER-Diagram สำหรับเก็บข้อมูลนักศึกษา หลังจากขั้นตอนนี้จะต้องถอดความสัมพันธ์เพื่อหาเอฟเฟกต์ของความสัมพันธ์ (ไม่อธิบายส่วนของการถอดความสัมพันธ์) และนำไปเขียนเป็นพจนานุกรมข้อมูล (Data Dictionary) เพื่อ

ความสะดวกสำหรับการพัฒนาฐานข้อมูล จากภาพที่ 13.3 สามารถ สร้างตารางความสัมพันธ์ได้ 3 ตาราง ประกอบด้วย 1) ตารางข้อมูลนักศึกษา (student) 2) ตารางข้อมูลสาขาวิชา (program) และ 3) ตาราง คณะ (faculty) มีรายละเอียดของตารางความสัมพันธ์ดังนี้

ตารางที่ 13.11 ตัวอย่างตารางข้อมูลนักศึกษา

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>student_id</u>	VARCHAR(5)	Not	-	PK	รหัสนักศึกษา
student_name	VARCHAR(40)	Not	-	-	ชื่อ-สกุล
birth_day	DATE	-	-	-	วันเดือนปีเกิด
address	VARCHAR(40)	-	-	-	ที่อยู่
program_id	VARCHAR(3)	Not	-	FK	รหัสสาขาวิชา

ตารางที่ 13.12 ตัวอย่างตารางข้อมูลสาขาวิชา

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>program_id</u>	VARCHAR(3)	Not	-	PK	รหัสสาขาวิชา
program_name	VARCHAR(40)	Not	-	-	ชื่อสาขาวิชา
head_name	VARCHAR(40)	-	-	-	ชื่อหัวหน้าสาขาวิชา
faculty_id	VARCHAR(3)	Not	-	FK	รหัสคณะ

ตารางที่ 13.13 ตัวอย่างตารางข้อมูลคณะ

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>faculty_id</u>	VARCHAR(3)	Not	-	PK	รหัสคณะ
faculty_name	VARCHAR(40)	Not	-	-	ชื่อคณะ
dean_name	VARCHAR(40)	-	-	-	ชื่อคณบดี

ตารางที่ 13.14 แสดงตัวอย่างข้อมูลในตารางข้อมูลนักศึกษา

<u>student_id</u>	student_name	birth_day	address	program_id
57001	Southern Thailand	1990/01/02	Suratthani	101
57002	Siriwatthan Jaidee	1990/05/18	Suratthani	101
57003	Rakthum Surat	1991/08/20	Chumphon	102
57004	Arun Gnamthong	1989/09/11	Suratthani	103
57005	Thongchai Swee	1990/08/12	Chumphon	104



ตารางที่ 13.15 แสดงตัวอย่างข้อมูลในตารางข้อมูลสาขาวิชา

program_id	program_name	head_name	faculty_id
101	Computer Science	Parinya Noidonprai	A01
102	Business	Aurai Somwang	A02
103	Biology	Siwat Hengchaiyo	A01
104	Human Develop	Siriwan Suwanmani	A03

ตารางที่ 13.16 แสดงตัวอย่างข้อมูลในตารางข้อมูลคณะ

faculty_id	faculty_name	dean_name
A01	Science and Techlogy	Somporn Thojai
A02	Management Science	Anirut Gnamsuwan
A03	Humanity Science	Sawai Suwanpradit

การใช้คำสอบถามเพื่อเรียกดูข้อมูลที่มาจากการมากกว่า 1 ตาราง (ตารางที่มีความสัมพันธ์กัน โดยใช้ขอบเขตข้อมูลสำหรับการอ้างอิง) ด้วย SELECT แบบมีเงื่อนไข อธิบายวิธีการดังนี้

กรณีศึกษาที่ 1 ต้องการดูข้อมูลนักศึกษาทั้งหมดในสาขาวิชา Computer Science สามารถใช้คำสอบถาม SELECT แบบมีเงื่อนไข ได้ดังนี้

ตัวอย่างที่ 13.27 การใช้คำสอบถามเพื่อเรียกดูข้อมูลที่มาจากการหลายตารางด้วย SELECT กรณีศึกษาที่ 1

```
mysql> SELECT `student`.`student_id`,
       `student`.`student_name`,
       `student`.`program_id`,
       `program`.`program_id`,
       `program`.`program_name`
  FROM student, program
 WHERE `student`.`program_id` = `program`.`program_id` AND
 `program`.`program_id` = '101'
 ORDER BY `student`.`program_id` ASC;
```

ผลลัพธ์

student_id	student_name	program_id	program_id	program_name
57001	Southern Thailand	101	101	Computer Science

57002 Siriwatthan Jaidee 101 101 Computer Science
57005 SThongchai Swee 101 101 Computer Science

กรณีศึกษาที่ 2 ต้องการดูข้อมูลนักศึกษาทั้งหมดในคณะ Science and Techlogy สามารถใช้คำสอบถาม SELECT แบบมีเงื่อนไข ได้ดังนี้

ตัวอย่างที่ 13.28 การใช้คำสอบถามเพื่อเรียกดูข้อมูลที่มาจากการถ่ายตารางด้วย SELECT กรณีศึกษาที่ 2

```
mysql> SELECT `student`.`student_id`,
       `student`.`student_name`,
       `student`.`program_id`,
       `program`.`program_id`,
       `program`.`program_name`,
       `program`.`faculty_id`,
       `faculty`.`faculty_id`,
       `faculty`.`faculty_name`
  FROM student, program, faculty
 WHERE `student`.`program_id`='`program`.`program_id`
 AND `program`.`faculty_id`='`faculty`.`faculty_id`
 AND `faculty`.`faculty_id`='A01'
 ORDER BY `student`.`program_id` ASC;
```

ผลลัพธ์

student_id	student_name	program_id	program_id	program_name	faculty_id	faculty_id	faculty_name
57001	Southern Thailand	101	101	Computer Science	A01	A01	Science and Techlogy
57002	Siriwatthan Jaidee	101	101	Computer Science	A01	A01	Science and Techlogy
57005	SThongchai Swee	101	101	Computer Science	A01	A01	Science and Techlogy
57004	Arun Gnamthong	103	103	Biology	A01	A01	Science and Techlogy

4 rows in set (0.00 sec)

13.8 จัดการฐานข้อมูลด้วย phpMyAdmin

เนื่องจากฐานข้อมูล MySQL ติดต่อกับผู้ใช้ผ่านทางคอมมานต์ไลน์ (Command Line) ทำให้ผู้ใช้ต้องพิมพ์คำสอบถามหรือคำสั่งต่างๆ ด้วยตนเอง จึงทำให้มีสะดวกต่อการใช้งาน จากปัญหาดังกล่าวทำให้มีผู้ที่คิดสร้างเครื่องมือขึ้นมาช่วยให้บรรดาโปรแกรมเมอร์ PHP สามารถจัดการกับฐานข้อมูล MySQL ได้



ง่ายขึ้น โดยเครื่องมือที่ว่ามีชื่อว่า phpMyAdmin เป็นเว็บแอปพลิเคชันที่เขียนด้วย PHP แต่มีความสามารถในการจัดการกับ MySQL ได้เกือบทุกอย่าง โดยจะติดต่อกับผู้ใช้ในแบบ GUI ทำให้ใช้งานง่าย ประหยัดเวลาในการพิมพ์คำสบคานหรือคำสั่ง และช่วยลดข้อผิดพลาดลงได้ค่อนข้างมาก นอกจากนี้ บรรดาผู้ให้บริการ Web Hosting ส่วนใหญ่จะมี phpMyAdmin ติดตั้งไว้ให้ใช้งานด้วย

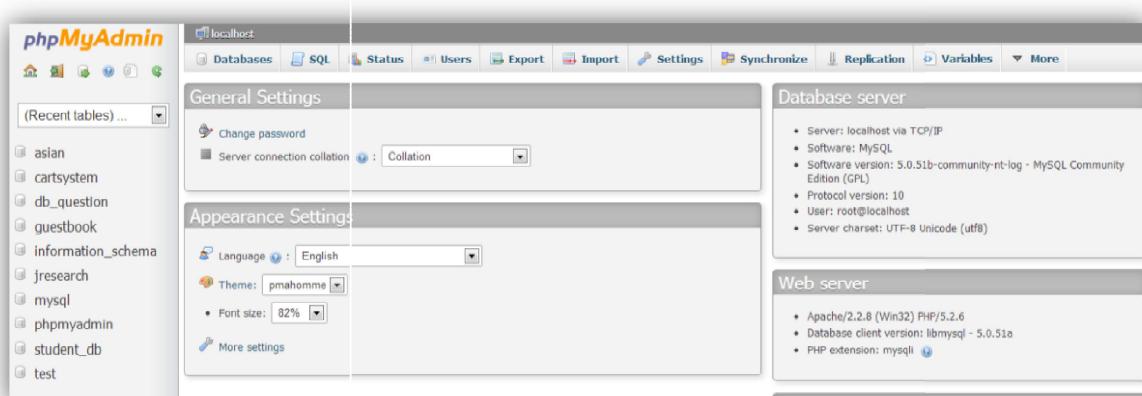
สำหรับผู้ที่ติดตั้ง AppServ จะมี phpMyAdmin เป็นองค์ประกอบอย่างหนึ่งอยู่แล้ว และหากในตอนติดตั้ง เลือกให้ติดตั้ง phpMyAdmin ลงไปด้วย ก็พร้อมใช้งานได้ทันที หรือสามารถติดตั้งเวอร์ชันใหม่ๆ สามารถดาวน์โหลดได้ที่ http://www.phpmyadmin.net/home_page/downloads.php เนื่องจาก phpMyAdmin มีหลายเวอร์ชัน สำหรับหนังสือเล่มนี้ผู้แต่งได้เลือกใช้เวอร์ชัน 3.5.4

การใช้งาน phpMyAdmin จะต้องทำผ่านเบราว์เซอร์เท่านั้น หน้าโฮมเพจ (Homepage) ปกติจะอยู่ที่ <http://localhost/phpmyadmin/> INDEX.php เมื่อเปิดเพจนี้ จะพบกับหน้าจอที่ให้ใส่ล็อกอิน และพัสเวิร์ด สำหรับพัสดุ์ที่เป็นของ MySQL ตามที่กำหนดไว้ตอนติดตั้ง



ภาพที่ 13.4 หน้าล็อกอินเพื่อเข้าสู่ระบบของ phpMyAdmin เวอร์ชัน 3.5.4

หลังการล็อกอินเข้าสู่ phpMyAdmin จะปรากฏหน้าเพจ ดังรูป

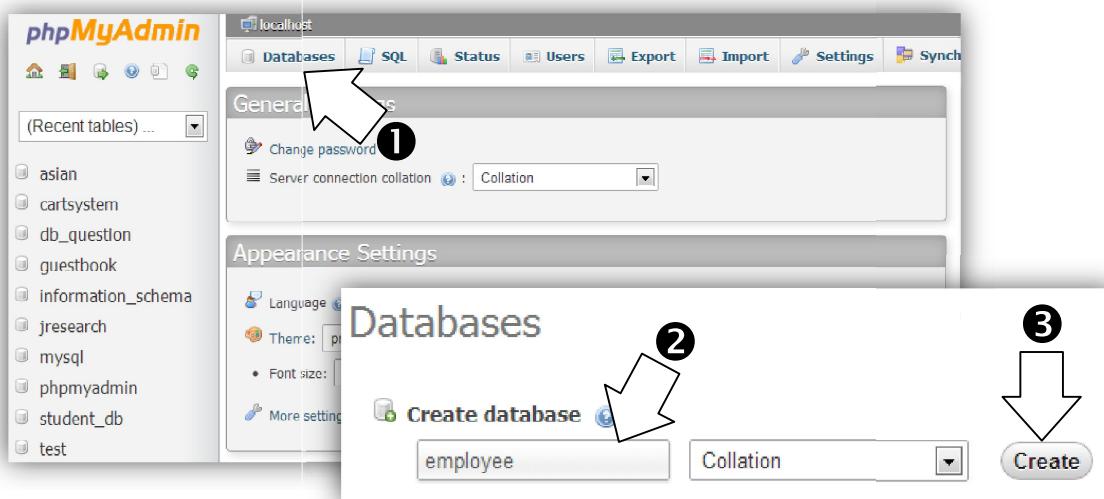


ภาพที่ 13.5 หน้าเพจหลังจากล็อกอินเข้าสู่ระบบ phpMyAdmin เวอร์ชัน 3.5.4

13.8.1 การสร้างฐานข้อมูล

การสร้างฐานข้อมูลนี้ จะเหมือนกับการใช้ `CREATE DATABASE` ที่ใช้ผ่านคอมมานด์ไลน์ของ MySQL โดยเริ่มต้นจากการสร้างฐานข้อมูลให้เริ่มจากเพจ มีขั้นตอนดังนี้

- ❶ คลิก "Databases"
- ❷ พิมพ์ชื่อฐานข้อมูลที่ต้องการสร้างใหม่ ส่วนการเรียงลำดับ (Collation) หากมีข้อมูลเป็นภาษาไทย และต้องการใช้เป็นเกณฑ์ในการเรียงลำดับ ก็ให้เลือก `utf8_general_ci`
- ❸ คลิกที่ปุ่ม "Create" เพื่อเริ่มต้นการสร้างฐานข้อมูลใหม่ หากต้องการสร้างฐานข้อมูลเพิ่ม ก็ให้ย้อนกลับไปขั้นตอนเริ่มแรกใหม่

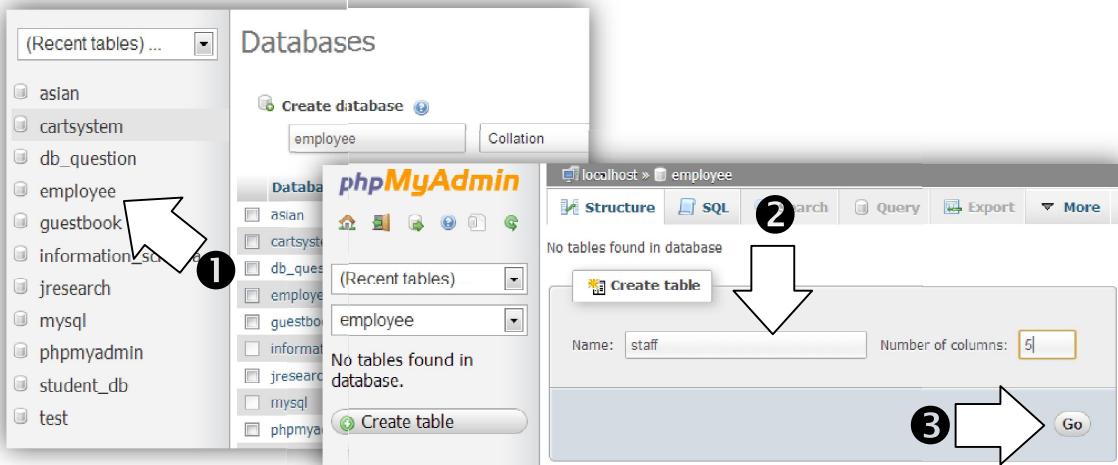


ภาพที่ 13.6 แสดงขั้นตอนการสร้างฐานข้อมูลใหม่

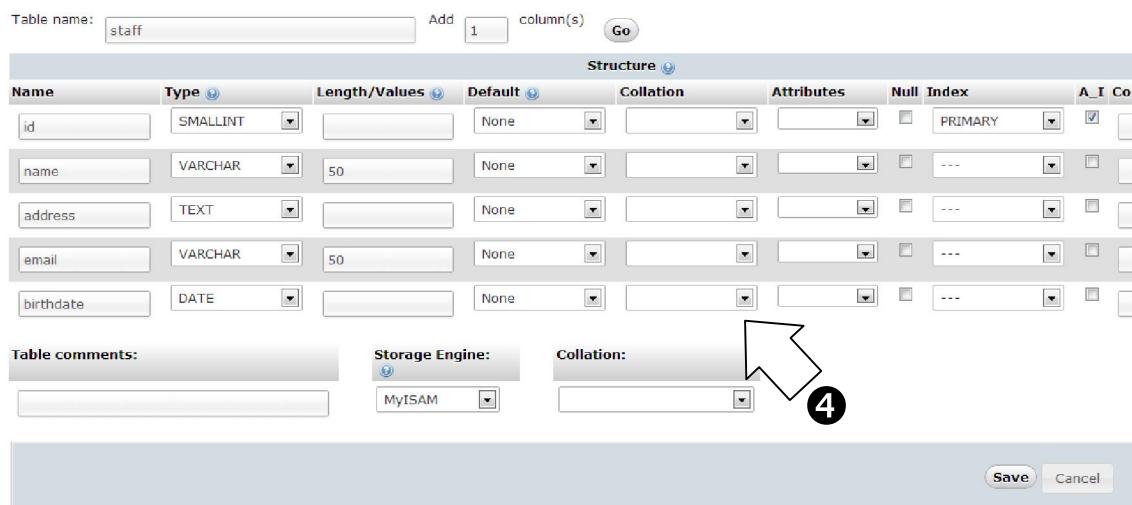
13.8.2 การสร้างตาราง

ก่อนการสร้างตารางต้องสร้างฐานข้อมูลไว้ก่อนแล้ว หลังจากนั้นก็ทำการสร้างฐานข้อมูลได้

- ❶ เลือกชื่อฐานข้อมูลจากเฟรมด้านซ้ายก่อน ว่าจะสร้างตารางนี้ไว้ในฐานข้อมูลใด
- ❷ หลังการเลือกชื่อฐานข้อมูลในข้อที่ 1 ให้ไปที่กรอบ "Create Table" ที่เฟรมด้านขวา แล้วใส่ชื่อตาราง และจำนวนขอบเขตข้อมูลของตารางนั้นลงไป
- ❸ คลิกปุ่ม "Go" หรือกดปุ่ม Enter เพื่อเริ่มสร้างตาราง
- ❹ ในขั้นตอนต่อไป จะเป็นการกำหนดโครงสร้างของตาราง หลังการสร้างตาราง จะปรากฏรายชื่อตารางที่เพิ่มด้านซ้าย (ถ้าไม่ปรากฏ ให้เลือกชื่อฐานข้อมูลจากอินพุต Select ก่อน) หากต้องการจัดการไดๆ เกี่ยวกับตาราง เช่น เพิ่มข้อมูล แก้ไขข้อมูล ก็ให้มาเริ่มเลือกที่นี่ก่อน
- ❺ ถ้าต้องการสร้างตารางเพิ่ม ให้ระบุชื่อตารางและจำนวนขอบเขตข้อมูล



ภาพที่ 13.7 แสดงขั้นตอนการสร้างตารางใหม่



ภาพที่ 13.8 แสดงขั้นตอนการสร้างขอบเขตข้อมูลในตาราง

เมื่อ Name	คือ ชื่อขอบเขตข้อมูล
Type	คือ ชนิดของขอบเขตข้อมูล
Length/Value	ความกว้างของข้อมูล
Default	กำหนดค่าปริยาย เมื่อผู้ใช้ไม่ป้อนข้อมูล
Collation	กำหนดรูปแบบการจัดเรียง
Attributes	กำหนดรูปแบบการจัดเก็บข้อมูล
Null	อนุญาตให้เป็นค่าว่างหรือไม่ หากเลือกจะหมายถึงอนุญาตให้เป็นค่าว่าง
INDEX	การกำหนดให้เป็นคีย์ประเภทได
A_I	คือ Auto Increment หากเลือกจะหมายถึงกำหนดให้เพิ่มข้อมูลแบบอัตโนมัติ (จะใช้รูปแบบนี้ได้จะต้องกำหนดชนิดข้อมูล เป็นชนิดตัวเลขจำนวนเต็มเท่านั้น)

Comments คือ หมายเหตุเพิ่มเติม

The screenshot shows the phpMyAdmin interface with the 'employee' database selected. In the center, there's a 'Create table' dialog box. At the top of this dialog, there's a 'Comments' input field. A large black arrow with the number '5' on it points from the explanatory text above to this 'Comments' field.

ภาพที่ 13.9 แสดงขั้นตอนการสร้างตารางและกำหนดขอบเขตข้อมูลในตาราง

13.8.3 การแทรกข้อมูลลงในตาราง

การแทรกข้อมูลลงในตารางต้องเลือกที่ตารางนั้นก่อน โดยไปที่เฟรมด้านซ้าย เลือกฐานข้อมูลแล้วเลือกชื่อตารางที่ต้องการ จากนั้นไปที่เฟรมด้านขวาเมื่อคลิก "Insert" ดังภาพ

The screenshot shows the phpMyAdmin interface with the 'employee' database selected. In the top navigation bar, the 'Insert' button is highlighted with a large black arrow pointing towards it.

ภาพที่ 13.10 แสดงขั้นตอนการแทรกข้อมูลลงในตาราง

หลังจากคลิก "Insert" จะมีฟอร์มมาให้ป้อนข้อมูลลงในแต่ละขอบเขตข้อมูลโดยมีหลักการสำคัญดังนี้

The screenshot shows the 'Insert Table' dialog for the 'staff' table. It contains fields for 'id' (smallint(6)), 'name' (varchar(50)), 'address' (text), 'email' (varchar(50)), and 'birthdate' (date). At the bottom is a 'Go' button.

Column	Type	Function	Null	Value
id	smallint(6)			
name	varchar(50)			
address	text			
email	varchar(50)			
birthdate	date			

ภาพที่ 13.11 แสดงตัวอย่างรายการสำหรับการแทรกข้อมูลลงในตาราง

1) ช่อง "Function" นั้นใช้ในกรณีที่ต้องการกำหนดค่าด้วยฟังก์ชันบางอย่าง โดยข้อมูลที่ใส่ลงในช่อง Value นั้น จะถูกนำไปร่วมกับฟังก์ชัน แล้วผลลัพธ์ที่ได้จากฟังก์ชัน จะถูกใส่เป็นข้อมูลให้แก่ขอบเขตข้อมูลนั้น

2) ถ้าต้องการใส่ข้อมูลไปโดยตรง ก็ให้วันช่องฟังก์ชันเอาไว้

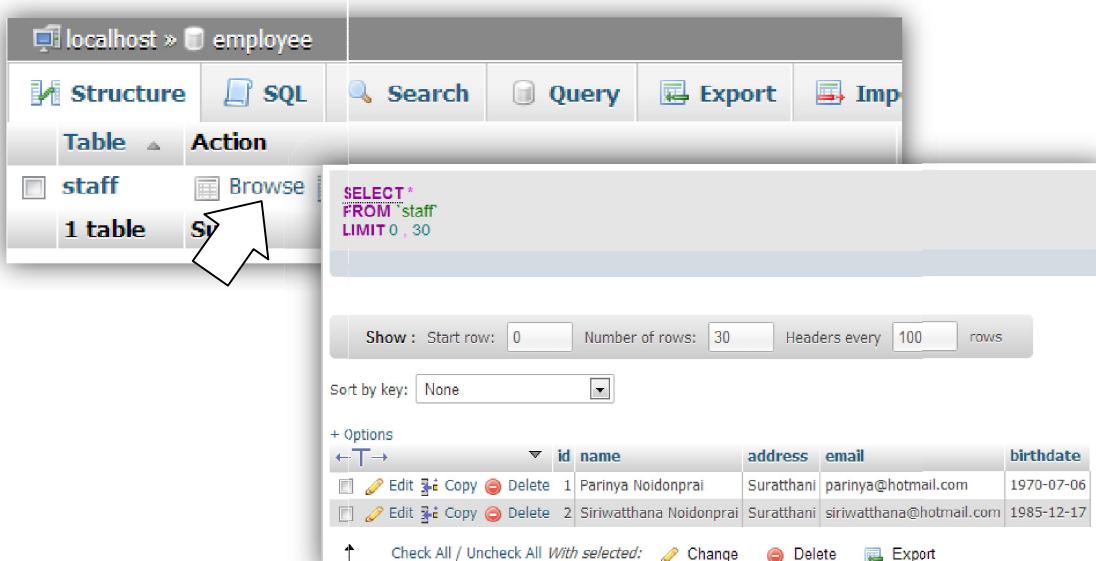
3) หากคุณต้องการเปลี่ยนเครื่องจักรให้เสื่อมูล 2 ฟอร์ม (ในรูปที่แสดงนี้ได้ตัดมาเฉพาะฟอร์มอันบน) โดยแต่ละฟอร์มใช้สำหรับข้อมูล 1 แค่ (ฟอร์มด้านบนจะมีการตรวจสอบรูปแบบของขอบเขตข้อมูลตามโครงสร้างที่ระบุ เช่น ไม่อนุญาตให้เป็นค่าว่าง เป็นต้น ส่วนฟอร์มด้านล่างจะข้ามการตรวจสอบลักษณะดังกล่าว)

4) หากต้องการใส่ข้อมูลพร้อมกัน 2 แถว ต้องคลิกເເຊື້ອກບົກ໌ "Ignore" ອອກ ແຕ່ຄ້າ
ຕ້ອງການໃສ່ເພີ່ມແລວເດືອຍ ກີ່ໄໜ້ຄຸລິກທີ່ເຂົ້າບົກ໌ຕ້າວເລືອກນີ້ ມີຈະນັ້ນຈະເກີດການແຫ່ງແລກວ່າງເຂົ້າໄປ 1 ແລວ

5) หลังจากที่ใส่ข้อมูลลงไปจนครบทุกขอบเขตข้อมูลแล้ว คลิกที่ปุ่ม "Go" เพื่อบันทึกลงฐานข้อมูล และถ้าต้องการใส่ข้อมูลแถวต่อไป ก็ให้เริ่มที่คุลิก "Insert" เช่นเดิม

13.8.4 การอ่านข้อมูลจากตาราง

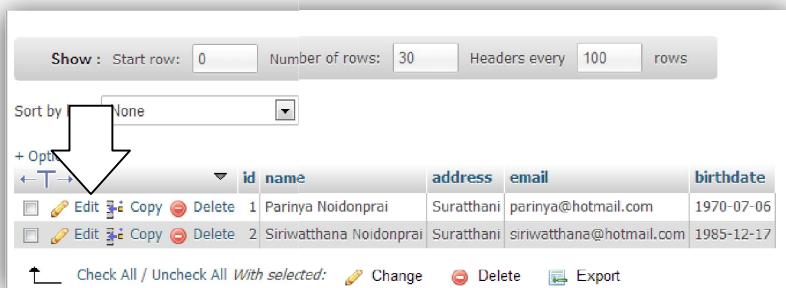
การอ่านข้อมูลทั้งหมดให้เริ่มจากการเลือกฐานข้อมูล และตารางจากเฟรมด้านซ้าย แล้วคลิกที่ "Browse" ดังขั้นตอนในรูปต่อไปนี้



ภาพที่ 13.12 แสดงขั้นตอนการอ่านข้อมูลในตาราง

13.8.5 การปรับปรุงข้อมูล

การที่จะปรับปรุงข้อมูลในตารางใดนั้น ต้องเปิดข้อมูลในตารางนั้นขึ้นมาแสดงก่อน โดยอาจเลือกจาก "Browse" ตามที่ได้กล่าวมาแล้ว จะสังเกตเห็นว่า ข้อมูลในแต่ละแควนั้นจะมีไอคอนรูปดินสอหรือข้อความ "Edit" ไอคอนดังกล่าวใช้สำหรับการเลือกข้อมูลเพื่อปรับปรุงข้อมูลແກาที่ถูกเลือก



	<input type="checkbox"/>				id	name	address	email	birthdate
	<input type="checkbox"/>				1	Parinya Noidonprai	Suratthani	parinya@hotmail.com	1970-07-06
	<input type="checkbox"/>				2	Siriwatthana Noidonprai	Suratthani	siriwatthana@hotmail.com	1985-12-17

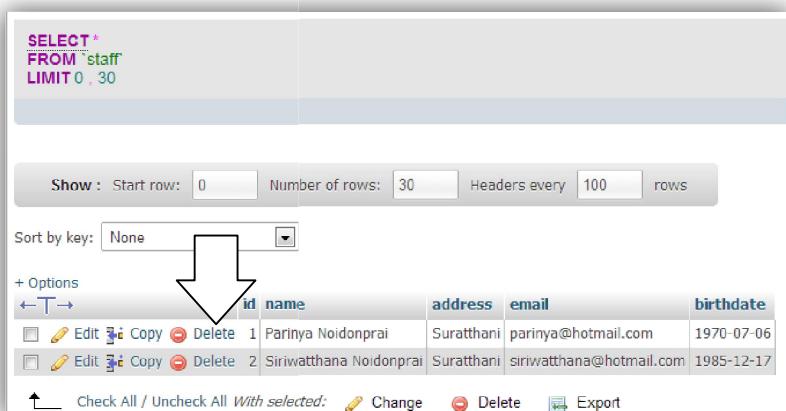
Check All / Uncheck All With selected: Change Export

ภาพที่ 13.13 แสดงขั้นตอนการเลือกข้อมูลเพื่อแก้ไขข้อมูล

ให้คลิกที่รูปดินสอตรงแถบที่ต้องการปรับปรุงข้อมูล "Edit" จากนั้นจะมีฟอร์มแสดงข้อมูลเดิมขึ้นมาให้เลือกปรับปรุงข้อมูล สำหรับขั้นตอนนี้ก็เหมือนกับการเพิ่มข้อมูลแล้วใหม่นั่นเอง

13.8.6 การลบข้อมูล

การลบก็ต้องเกิดข้อมูลตารางนั้นขึ้นมาก่อน เช่นเดียวกับการแก้ไข จากนั้นคลิกที่ไอคอนวงกลมสีแดงหรือที่ข้อความ "Delete" หน้าແຄວที่ต้องการลบ ดังภาพ



	<input type="checkbox"/>				id	name	address	email	birthdate
	<input type="checkbox"/>				1	Parinya Noidonprai	Suratthani	parinya@hotmail.com	1970-07-06
	<input type="checkbox"/>				2	Siriwatthana Noidonprai	Suratthani	siriwatthana@hotmail.com	1985-12-17

Check All / Uncheck All With selected: Change Export

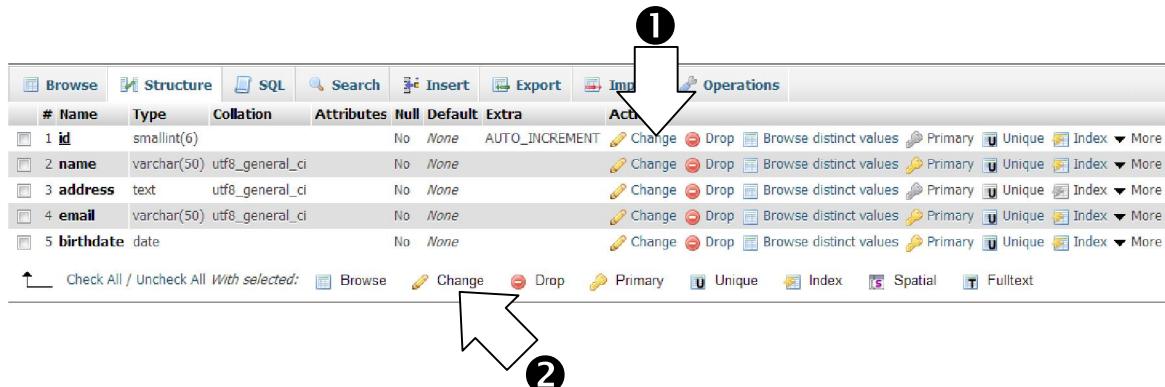
ภาพที่ 13.14 แสดงขั้นตอนการเลือกข้อมูลเพื่อลบข้อมูล

จากนั้นโปรแกรมจะถามยืนยันการลบอีกรอบ หากยืนยันที่จะลบ ก็ให้เลือกคลิกที่ปุ่ม OK แต่หากต้องการยกเลิกให้คลิกเลือกที่ปุ่ม Cancel

13.8.7 การเปลี่ยนแปลงโครงสร้างของตาราง

การแก้ไขโครงสร้างของตารางที่ต้องการจากเฟรมด้านซ้ายก่อน แล้วคลิกที่ "Structure" จากนั้นจะปรากฏโครงสร้างของตารางให้เลือกแก้ไขดังนี้

- ถ้าต้องการเปลี่ยนแปลงเพียงขอบเขตข้อมูลหนึ่ง ให้เลือกคลิก "Change" ตามແກ່ที่ต้องการเปลี่ยนแปลง
- ถ้าต้องการเปลี่ยนแปลงมากกว่า 1 ขอบเขตข้อมูล ให้คลิกเข้าบ็อกซ์ที่หน้าขอบเขตข้อมูลนั้นๆ แล้วเลือกที่ "Change" ด้านล่างของข้อมูล



ภาพที่ 13.15 แสดงตัวอย่างการเลือกข้อมูลเพื่อแก้ไขโครงสร้างข้อมูล

13.9 การลบฐานข้อมูล และตาราง

การลบฐานข้อมูล หรือตารางใดนั้น โดยแยกพิจารณาได้ดังนี้

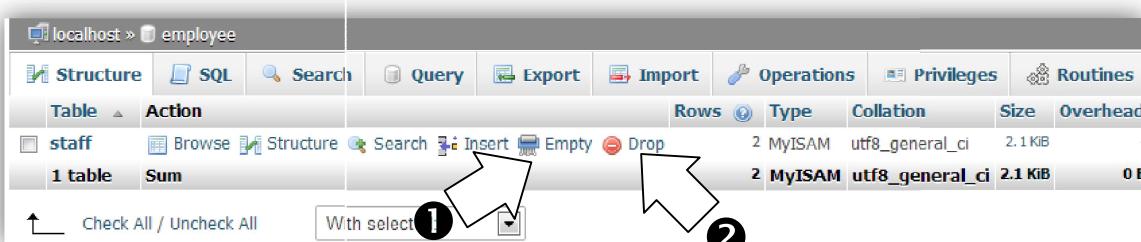
13.9.1 การลบฐานข้อมูล

ให้เลือกชื่อฐานข้อมูลจากเฟรมด้านซ้ายก่อน แล้วเลือกที่ "Drop" จากนั้นจะปรากฏข้อความเพื่อถามยืนยันการลบ การลบฐานข้อมูลจะทำให้ทุกตารางที่อยู่ในฐานข้อมูลนั้นถูกลบทิ้งไปด้วย

13.9.2 การลบตาราง

ต้องเริ่มจากการเลือกตารางที่ต้องการลบจากเฟรมด้านซ้ายก่อน ทั้งนี้การลบตารางจะแบ่งเป็น 2 ตัวเลือก ประกอบด้วย 1) ลบเฉพาะข้อมูลในตาราง และ 2) ลบทั้งตาราง ทั้ง 2 ตัวเลือก มีรายละเอียดดังนี้

- 1) "Empty" ใช้สำหรับการลบข้อมูลทั้งหมดในตาราง เป็นการลบเฉพาะข้อมูล แต่โครงสร้างของตารางยังคงอยู่ เมื่อคลิกเลือกรายการนี้จะมีข้อความเพื่อให้ยืนยัน
- 2) "Drop" สำหรับการลบข้อมูลทั้งหมด รวมถึงโครงสร้างตารางด้วย เมื่อคลิกรายการนี้จะมีข้อความเพื่อให้ยืนยัน



ภาพที่ 13.16 แสดงตัวอย่างการลบข้อมูลในตารางและลบตาราง

13.10 ส่วนของการใช้คำสອบแบบกำหนดเอง

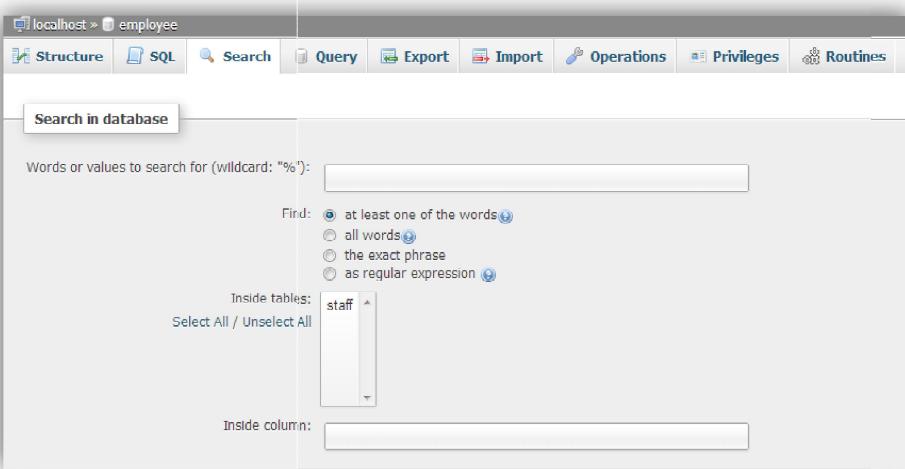
เมื่อคลิก SQL จะปรากฏหน้าจอสำหรับพิมพ์คำสອบตามหรือคำสั่งด้วยตนเองตามรูปแบบมาตรฐาน SQL และเมื่อพิมพ์คำสອบตามหรือคำสั่งเสร็จ ให้คลิกปุ่ม "Go" มีตัวอย่างดังภาพ



ภาพที่ 13.17 แสดงตัวอย่างการใช้คำสອบถາມหรือคำສັ່ງແບບກຳຫນດເອງຜ່ານ "Run SQL Query"

13.11 ส່ວນຂອງກຳຫນາ

ໃຊ້ກຳຫນາຂໍ້ອມູລໃນຕາຮາງຕາມຄຳແລະຕາຮາງທີ່ຮັບ ເມື່ອຄົລິກສ່ວນຂອງກຳຫນາ "Search" ມີຕົວຢ່າງດັ່ງການ



ภาพที่ 13.18 แสดงຕົວຢ່າງກຳຫນາຂໍ້ອມູລເພື່ອກຳຫນາໃນ "Run SQL Query" ເພື່ອກຳຫນາຂໍ້ອມູລ

- | | |
|---------------------------------|--------------------------------------|
| ເມື່ອ at least one of the words | ກືອ ແມ່ນອຍ່າງນ້ອຍໜຶ່ງຄຳຂອງຄຳທີ່ກຳຫນາ |
| all words | ກືອ ແມ່ນທຸກຄຳທີ່ກຳຫນາ |
| the exact phrase | ກືອ ແມ່ນທຸກລື້ຖານທີ່ຕ້ອງກຳຫນາ |
| as regular expression | ກືອ ກຳຫນາຕາມໜັກ Regular Expression |

13.12 ສ່ວນຂອງກຳຫນດສີທີ່

ໃຊ້ກຳຫນດສີທີ່ກຳຫນາຂໍ້ອມູລໄຟກັບຜູ້ໃໝ່ ເມື່ອຄົລິກ "Privileges" ຈະປາກງູ້ນໍາແສດງຮາຍຊື່ອຸ່ປ້າໃໝ່ ແລະສີທີ່ໃນການເຂົ້າໃຈ້ານฐานຂໍ້ອມູລ ມີຕົວຢ່າງດັ່ງການ

User	Host	Type	Privileges	Grant	Action
root	127.0.0.1	global	ALL PRIVILEGES	yes	Edit Privileges
root	localhost	global	ALL PRIVILEGES	yes	Edit Privileges
root	production.mysql.com	global	ALL PRIVILEGES	yes	Edit Privileges

ภาพที่ 13.19 แสดงตัวอย่างการใช้คำสั่งผ่าน "Run SQL Query" เพื่อกำหนดสิทธิ์

13.13 เทคนิคการนำเข้าข้อมูลจากไฟล์ Excel

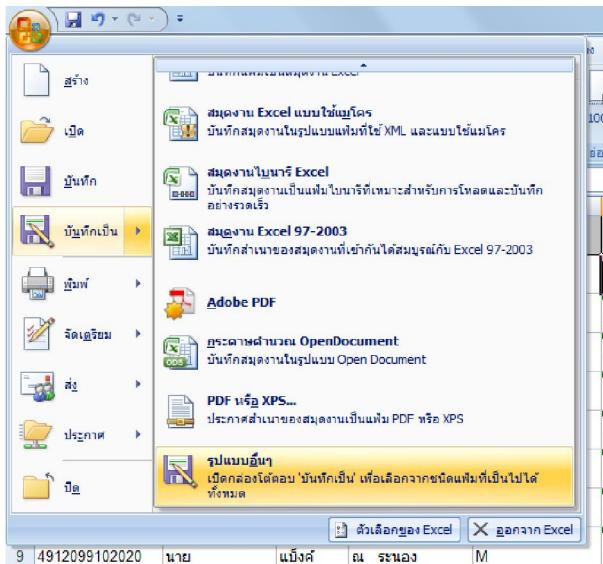
ผู้พัฒนาเว็บแอปพลิเคชันหลายคนคงเจอปัญหาเหมือนๆ กัน คือ มีข้อมูลอยู่ในไฟล์ MS-Excel แล้วต้องการที่จะนำข้อมูลไปใช้กับฐานข้อมูล MySQL ปัญหาที่พบ คือ เมื่อมีการนำเข้าไฟล์ (Import File) แล้วจะพบข้อผิดพลาดหลายประการและที่สำคัญ คือ ไม่สามารถแสดงเป็นภาษาไทยได้

สำหรับเนื้อหาในส่วนนี้จะอธิบายวิธีการนำเข้าข้อมูลจาก Excel เข้าสู่ฐานข้อมูล MySQL และสามารถแสดงออกมาเป็นภาษาไทยในรูปแบบ UTF-8 แบบไม่มีปัญหาใดๆ โดยมีขั้นตอน ดังต่อไปนี้

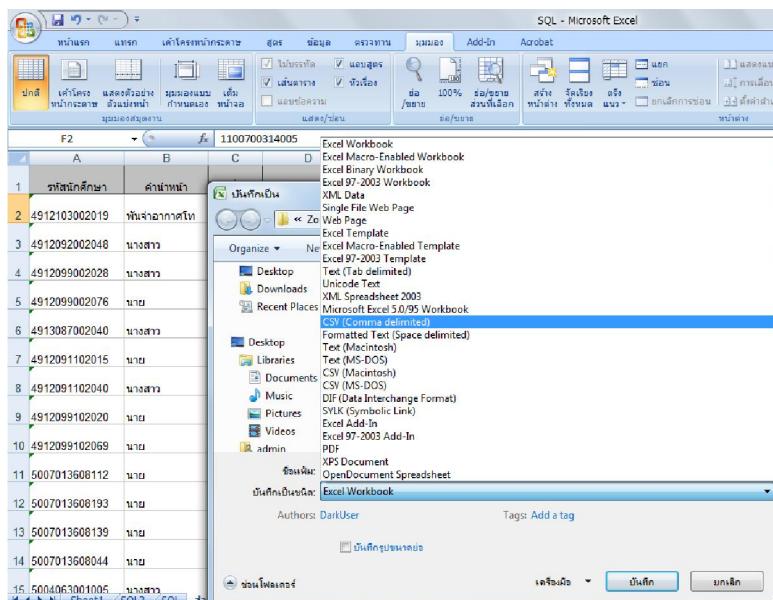
	A	B	C	D	E
1	รหัสพนักงาน	คำนำหน้า	ชื่อ	นามสกุล	เพศ
2	4912103002019	พันจ้าวภาคโน	บัญฑิต	เอียนสะวัด	M
3	4912092002048	นางสาว	กานกรัตน์	บัวเพชร	F
4	4912099002028	นางสาว	เกศรินทร์	ขอตศชัย	F
5	4912099002076	นาย	กันก	ท่วมจร	M
6	4913087002040	นางสาว	อธิกาญจน์	อภิรัตน์	F
7	4912091102015	นาย	กมล	จารุรงค์	M
8	4912091102040	นางสาว	สมพร	น่วงทอง	F

ภาพที่ 13.20 แสดงตัวอย่างข้อมูลในไฟล์ Excel

13.13.1 บันทึกไฟล์ให้เป็น CSV โดยไปที่รายการ บันทึกเป็น -> รูปแบบอื่นๆ (File -> Save As -> OtherFormats) และเลือกชนิดเป็น CSV (Comma delimited) (*.csv)

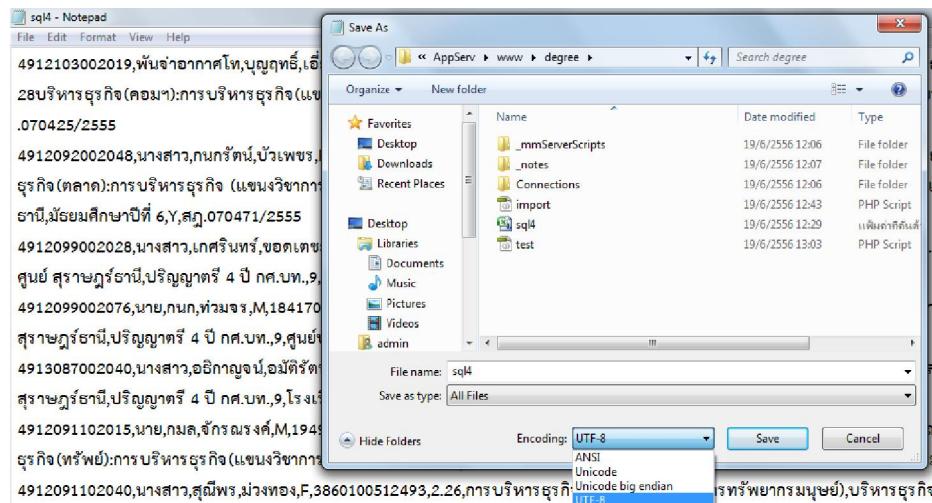


ภาพที่ 13.21 แสดงตัวอย่างการบันทึกไฟล์ให้เป็นรูปแบบอื่นๆ จากรายการ



ภาพที่ 13.22 แสดงตัวอย่างการบันทึกไฟล์ให้เป็นรูปแบบ CSV (Comma Delimited)

13.13.2 จักนั้นทำการแปลง Encode ของไฟล์เป็น UTF-8 โดยเปิดไฟล์ CVS ด้วย Notepad แล้วบันทึกไฟล์ชื่ออีกครั้ง (เลือก Save as Type เป็นชนิด All File และระบุนามสกุลไฟล์เป็น .csv) ในส่วนของ Encoding เลือก UTF-8 ดังภาพที่ 13.19



ภาพที่ 13.23 แสดงตัวอย่างการบันทึกไฟล์ให้เป็นรูปแบบ CSV และกำหนด UTF-8 ใน Notepad

13.13.3 สร้างฐานข้อมูลและตารางใน MySQL ตามข้อมูลที่ต้องการ ในตัวอย่างจะมีจำนวนขอบเขตข้อมูลทั้งหมด 18 ขอบเขตข้อมูล (ในตัวอย่างจะใช้ตัวแปร \$objArr[ลำดับ] เมื่อ ลำดับ คือ ตำแหน่งของอาร์เรย์ที่ใช้เก็บข้อมูลช่วงระหว่างก่อนแทรกลงตาราง โดยเริ่มต้นที่ 0-17)

13.13.4 เขียนสคริปต์ PHP เพื่ออ่านข้อมูลจากไฟล์ CVS ลงฐานข้อมูล MySQL มีตัวอย่าง สคริปต์ดังนี้

ตัวอย่างที่ 13.29 ตัวอย่างการใช้งานข้อมูลจากไฟล์ CVS บันทึกลงฐานข้อมูล MySQL

```
<?php
//ส่วนของการเชื่อมต่อฐานข้อมูล MySQL
$objConnect = mysql_connect("localhost","parinya","password") or die("Error
Connect to Database"); // Conect to MySQL

$objDB = mysql_select_db ("degree");
//ทำการเปิดไฟล์ CSV เพื่อนำข้อมูลไปใส่ใน MySQL
$objCSV = fopen ("sql.csv", "r");
while (($objArr = fgetcsv($objCSV, 3555, ",")) != FALSE) {
    //นำข้อมูลใส่ในตาราง (ให้ระบุชื่อตารางตามต้องการ ในตัวอย่างชื่อ student)
    $strSQL = "INSERT INTO student ";
    //ข้อมูลใส่ใน field ข้อมูลดังนี้
    $strSQL .="(`student_id` , `before_name` , `name` , `last_name` , `gender` ,
`people_id` , `gpa` , `program` , `degree` , `student_type` , `faculty` , `name_group` ,
`level` , `top` , `school` , `name_school` , `confirm` , `degree_number` ) ";
    $strSQL .="VALUES ";
}

```

```

// ข้อมูลตามที่อ่านได้จากไฟล์ลงฐานข้อมูล (ระบุอาร์เรย์เท่ากับจำนวน Field ที่ต้องการ)
// โดยตัวแปร $strSQL คือ ใช้ต่อไปยังคำสั่งให้เป็นประโยชน์เดียวกัน
// สามารถเขียนบรรทัดเดียวกันได้ตามที่ต้องการ

$strSQL .="(" . $objArr[0]. " ' ". $objArr[1]. " ' ". $objArr[2]. " ' ";
$strSQL .=" ". $objArr[3]. " ' ". $objArr[4]. " ' ". $objArr[5]. " ' ". $objArr[6]. " ' ";
$strSQL .=" ". $objArr[7]. " ' ". $objArr[8]. " ' ". $objArr[9]. " ' ". $objArr[10]. " ' ";
$strSQL .=" ". $objArr[11]. " ' ". $objArr[12]. " ' ". $objArr[13]. " ' ". $objArr[14]. " ' ";
$strSQL .=". $objArr[15]. " ' ";
$strSQL .=" ". $objArr[16]. " ' ". $objArr[17]. " ') ";
mysql_query ("SET NAMES UTF8");
//เพิ่มข้อมูลลงฐานข้อมูล
$objQuery = mysql_query ($strSQL);
}

fclose ($objCSV);
echo "Import Done.";

?>

```

หากการนำเข้าไฟล์ข้อมูลเข้าไม่มีปัญหาจะขึ้นข้อความว่า Import Done. แสดงว่าการนำเข้าเสร็จสมบูรณ์ และเมื่อลองดูในฐานข้อมูลก็จะพบว่ามีข้อมูลที่นำเข้าอยู่ในฐานข้อมูล ตัวอย่างดังภาพที่ 13.24

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	student_id	before_name	name	last_name	gender
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	491210300201	หนึ่งอาทิตย์	นฤกษา	เบี้ยนทร์	M
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4912092002048	นางสาว	กานต์	น้ำเพชร	F
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4912099002028	นางสาว	เกศรินทร์	น้อยเหลา	F
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4912099002076	นาย	กานต์	ท่วงจร	M
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4913087002040	นางสาว	อริกาญจน์	อภิสิริณ์	F
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4912091102015	นาย	กมล	จักรธรรมศ	M
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4912091102040	นาย	สมพร	ม่วงกาวงศ์	F
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4912099102020	นาย	แม่ใจ	ณ ธนาวงศ์	M

ภาพที่ 13.24 แสดงตัวอย่างข้อมูลในตาราง student เป็นผลมาจากการนำเข้ามาจากไฟล์ CSV



สรุป

ในบทนี้ได้อธิบายวิธีการใช้คำสอปถามและคำสั่งพื้นฐาน รวมถึงเครื่องมือสำหรับบริหารจัดการฐานข้อมูล MySQL จะเห็นได้ว่าองค์ประกอบที่สำคัญของระบบฐานข้อมูลนี้ จะประกอบไปด้วยฐานข้อมูล ตารางฐานข้อมูล ขอบเขตข้อมูล ชนิดของขอบเขตข้อมูล และแอฟทริบิวต์ของขอบเขตข้อมูล ซึ่งภายในฐานข้อมูลจะประกอบไปด้วยตารางฐานข้อมูลที่มีความสัมพันธ์กัน โดยจะมีจำนวนเท่าไหร่นั้นก็ขึ้นอยู่กับระบบงานที่จะพัฒนา และในแต่ละตารางฐานข้อมูลก็จะประกอบไปด้วยขอบเขตข้อมูลที่เกี่ยวข้อง กันเป็นตัวแทนของสิ่งที่สนใจ แต่ละขอบเขตข้อมูลก็จะเป็นต้องกำหนดชนิดให้สอดคล้องกับข้อมูลที่จะจัดเก็บด้วย โดยสรุปแล้วสิ่งที่สำคัญกว่าการใช้เครื่องมือบริหารจัดการฐานข้อมูลก็คือ การวิเคราะห์และออกแบบระบบฐานข้อมูล เพราะหากวิเคราะห์และออกแบบระบบฐานข้อมูลไม่ดี ระบบงานที่จะเกิดขึ้นย่อมที่จะมีปัญหาตามมา และยากที่จะแก้ไข เช่น เกิดความช้าช้อนในการจัดเก็บข้อมูล ไม่สามารถคำนวณหาค่าผลสรุปได้ เนื่องจากกำหนดชนิดของขอบเขตข้อมูลไม่ถูกต้อง เป็นต้น ดังนั้นก่อนจะเริ่มสร้างฐานข้อมูลและองค์ประกอบต่างๆ ที่เกี่ยวข้อง ควรเริ่มต้นจากการวิเคราะห์และออกแบบระบบฐานข้อมูลที่ดี

คำถามท้ายบท

1. จงอธิบายถึงระบบบริหารจัดการฐานข้อมูลเชิงสัมพันธ์ และยกตัวอย่างระบบบริหารจัดการฐานข้อมูลเชิงสัมพันธ์ที่นิยมใช้ในปัจจุบัน
2. จงอธิบายคำสั่ง และรูปแบบของคำสั่ง พร้อมยกตัวอย่างประกอบการใช้งาน ดังนี้
 - 2.1 คำสั่งที่ใช้สำหรับลบฐานข้อมูลหรือตารางฐานข้อมูล
 - 2.2 คำสั่งที่ใช้สำหรับสร้างฐานข้อมูล
 - 2.3 คำสั่งที่ใช้สำหรับสร้างตารางจากตารางอื่นที่มีอยู่แล้ว
3. จงเขียนโปรแกรมหาข้อมูลตามโจทย์ และตารางข้อมูลที่กำหนด ดังนี้

ชื่อตาราง employees (พนักงาน)

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889

- 3.1 สອบถามข้อมูลพนักงานที่อยู่ในจังหวัดสุราษฎร์ธานี
- 3.2 สອบถามข้อมูลพนักงานที่มีเงินเดือนอยู่ในช่วง 18000 – 30000 บาท
- 3.3 สອบถามข้อมูลพนักงานที่มีตำแหน่งเป็นโปรแกรมเมอร์
4. จากตาราง employees ในข้อที่ 3 จงอธิบายคำสั่งที่ใช้สำหรับคัดลอกขอบเขตข้อมูล และข้อมูล ไปสร้างเป็นตารางใหม่ ชื่อ ProgrammerStaff
5. จากตาราง employees ในข้อที่ 3 จงอธิบายคำสั่งที่ใช้สำหรับแทรกข้อมูลใหม่ลงในตาราง

บทที่ 14

การใช้ PHP ร่วมกับ MySQL

ในบทนี้จะกล่าวถึงฟังก์ชันของภาษา PHP สำหรับการเชื่อมต่อและประยุกต์ใช้งานร่วมกับ MySQL สำหรับฟังก์ชันที่เกี่ยวข้องนั้นมีจำนวนมาก แต่มีข้อแนะนำ คือ ชื่อฟังก์ชันที่เกี่ยวข้องเหล่านี้จะขึ้นต้นด้วยคำว่า mysql เป็นส่วนใหญ่ ทำให้จำได้ไม่ยาก เน้นประยุกต์การใช้งานเป็นสำคัญเพื่อเป็นพื้นฐานสำหรับการนำไปประยุกต์ใช้สำหรับการพัฒนาระบบที่มีการเชื่อมต่อกับฐานข้อมูล MySQL ต่อไป

14.1 ฟังก์ชันเชื่อมต่อ/ยกเลิกการเชื่อมต่อระหว่างภาษา PHP และระบบฐานข้อมูล MySQL

ก่อนที่จะใช้งานฐานข้อมูลได้ จะต้องเปิดการเชื่อมต่อกับฐานข้อมูลให้ได้ก่อน โดยฟังก์ชันที่ใช้ในการเปิดการเชื่อมต่อกับฐานข้อมูล MySQL ประกอบด้วย 1) mysql_connect () และ 2) mysql_pconnect () แต่เนื่องจากระบบฐานข้อมูล MySQL เป็นระบบฐานข้อมูลเชิงสัมพันธ์ที่มีการตรวจสอบสิทธิ์ในการเข้าใช้งานก่อนการเข้าใช้งานจัดการกับระบบฐานข้อมูล ดังนั้นฟังก์ชันที่ใช้จะต้องระบุข้อมูลที่สำคัญประกอบด้วย ชื่อหรือหมายเลข IP ของเซิร์ฟเวอร์ที่ติดตั้งฐานข้อมูล MySQL หากเป็นเครื่องเซิร์ฟเวอร์เดียวกันกับเว็บเซิร์ฟเวอร์ที่กำลังใช้งานอยู่ในปัจจุบัน สามารถระบุเป็น localhost แทนชื่อเซิร์ฟเวอร์ได้ ตามด้วยชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) เพื่อใช้ตรวจสอบสิทธิ์ก่อนการเข้าใช้งานฐานข้อมูล MySQL หากข้อมูลที่ระบุถูกต้อง ฟังก์ชันจะคืนค่าเป็น MySQL link identifier (Resource) แต่หากไม่ถูกต้องจะคืนค่าเป็นเท็จ (False) สำหรับความแตกต่างของฟังก์ชัน mysql_connect () และ mysql_pconnect () มี 2 ประเด็น ดังนี้

14.1.1 การใช้งานด้วยฟังก์ชัน mysql_pconnect () จะคืนหัวว่ามีการเชื่อมต่อกับระบบฐานข้อมูล MySQL เซิร์ฟเวอร์ ด้วยชื่อผู้ใช้ และรหัสผ่าน อยู่ก่อนหน้านี้หรือไม่ หากพบ จะใช้ MySQL link identifier เดิม (เรียกวิธีการเชื่อมต่อแบบนี้ว่า "Persistent") แต่หากไม่พบการเชื่อมต่อเดิม จะทำการสร้างการเชื่อมต่อใหม่ แต่สำหรับฟังก์ชัน mysql_connect () จะทำการเชื่อมต่อใหม่ทุกครั้งเมื่อมีการเรียกใช้ฟังก์ชัน

14.1.2 การยกเลิกการเชื่อมต่อกับฐานข้อมูล MySQL ฟังก์ชัน mysql_pconnect () ไม่ต้องใช้คำสั่งยกเลิกการเชื่อมต่อกับ MySQL link identifier หลังจากที่ใช้งานระบบฐานข้อมูลเสร็จแล้ว เพราะจะเก็บไว้ใช้งานในครั้งต่อไป ส่วนฟังก์ชัน mysql_connect () จะต้องใช้ฟังก์ชัน mysql_close () สำหรับยกเลิกการเชื่อมต่อกับ MySQL link identifier เพื่อคืนหน่วยความจำให้กับระบบ

รูปแบบการใช้งานฟังก์ชัน mysql_connect() และ mysql_pconnect() ใช้สำหรับการเชื่อมต่อกับระบบฐานข้อมูล MySQL มีรูปแบบดังนี้

รูปแบบ

```
resource mysql_connect | mysql_pconnect (string $server, string $username,
string $password)
```

เมื่อ \$server หมายถึง ชื่อเครื่องเซิร์ฟเวอร์ที่ติดตั้งระบบฐานข้อมูล MySQL หรือหมายเลข IP หรือสามารถใช้ localhost (กรณีเป็นเครื่องเซิร์ฟเวอร์เดียวกันกับเว็บเซิร์ฟเวอร์ปัจจุบันที่กำลังใช้งานอยู่)
\$username หมายถึง ชื่อผู้ใช้ที่มีสิทธิเชื่อมต่อและเรียกใช้งานระบบฐานข้อมูล MySQL
\$password หมายถึง รหัสผ่านของผู้ใช้ที่มีสิทธิเชื่อมต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

รูปแบบการใช้งานฟังก์ชัน mysql_close() ใช้สำหรับการยกเลิกการเชื่อมต่อกับระบบฐานข้อมูล MySQL มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
bool mysql_close ([ resource $link_identifier = NULL ] )
```

เมื่อ \$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.1 การเชื่อมต่อและยกเลิกการเชื่อมต่อระบบฐานข้อมูล MySQL ด้วยฟังก์ชัน mysql_connect() และ mysql_close()

```
1 <?php
2     $link = mysql_connect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) {
4         echo "<br>ไม่สามารถเชื่อมต่อกับระบบฐานข้อมูล MySQL<br>";
5         exit;
6     } else {
7         echo "<br>การเชื่อมต่อเสร็จสมบูรณ์ ด้วยฟังก์ชัน mysql_connect<br>";
8         mysql_close ($link);
9     }
10 ?>
```

จากตัวอย่างที่ 14.1 การเชื่อมต่อและยกเลิกการเชื่อมต่อด้วยฟังก์ชัน mysql_connect() อธิบายรายละเอียด ดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP



บรรทัดที่ 2 กำหนดตัวแปร \$link (ทำหน้าที่เป็น MySQL link identifier) รับผลจากการเรียกใช้ฟังก์ชัน mysql_connect() โดยกำหนดพารามิเตอร์ ประกอบด้วย

"localhost" คือ ชื่อเครื่องเซิร์ฟเวอร์ที่ติดตั้งระบบฐานข้อมูล MySQL โดยการที่จะกำหนดแบบนี้ได้นั้นแสดงว่าเครื่องเซิร์ฟเวอร์ที่เรียกใช้งานเป็นเครื่องเดียวกันกับเว็บเซิร์ฟเวอร์ปัจจุบันที่กำลังใช้งานอยู่

"mysql_user" คือ ชื่อผู้ใช้ที่มีสิทธิเข้มต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

"mysql_password" คือ รหัสผ่านของผู้ใช้ที่มีสิทธิเข้มต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

บรรทัดที่ 3 ใช้โครงสร้างเงื่อนไข if ตรวจสอบค่าของตัวแปร \$link ว่าสามารถเชื่อมต่อระบบฐานข้อมูล MySQL ได้หรือไม่ หากเขื่อมต่อไปได้ ไปบรรทัดที่ 4 หากสามารถเชื่อมต่อได้ไปบรรทัดที่ 6

บรรทัดที่ 4 แสดงข้อความ "ไม่สามารถเชื่อมต่อกับระบบฐานข้อมูล MySQL"

บรรทัดที่ 5 หยุดการทำงานคำสั่งที่เหลือทั้งหมด

บรรทัดที่ 6 สิ้นสุดสคริปต์โครงสร้างเงื่อนไข if และเริ่มโครงสร้างเงื่อนไข else (กรณีเป็นเท็จจากบรรทัดที่ 3)

บรรทัดที่ 7 แสดงข้อความ "การเชื่อมต่อเสร็จสมบูรณ์ ด้วยฟังก์ชัน mysql_pconnect" หรือโดยความหมาย คือ ใช้คำสั่งอื่นๆ ตามที่ผู้พัฒนาต้องการจะกำหนดให้ระบบทำอะไรต่อจากนี้เป็นต้นไป

บรรทัดที่ 8 ใช้ฟังก์ชัน mysql_close() เพื่อปิดการเชื่อมต่อกับระบบฐานข้อมูล MySQL

บรรทัดที่ 9 สิ้นสุดสคริปต์โครงสร้างเงื่อนไข else

บรรทัดที่ 10 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 14.2 การเชื่อมต่อระบบฐานข้อมูล MySQL ด้วยฟังก์ชัน mysql_pconnect()

```

1 <?php
2     $link = mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) {
4         echo "<br>ไม่สามารถเชื่อมต่อกับระบบฐานข้อมูล MySQL<br>";
5         exit;
6     } else {
7         echo "<br>การเชื่อมต่อเสร็จสมบูรณ์ ด้วยฟังก์ชัน mysql_pconnect<br>";
8     }
9 ?>
```



จากตัวอย่างที่ 14.1 และ 14.2 จะเห็นได้ว่าการใช้งานฟังก์ชันอื่นๆ หลังจากเชื่อมต่อกับระบบฐานข้อมูล MySQL เมื่อนักพัฒนาต้องการตัดขาดการเชื่อมต่อ ก็จะต้องใช้ฟังก์ชัน mysql_connect() และ mysql_close() ร่วมด้วยเสมอเมื่อต้องการยกเลิกการเชื่อมต่อ

นอกจากเรียกใช้ฟังก์ชันสำหรับเชื่อมต่อ/ยกเลิกการเชื่อมต่อระหว่างภาษา PHP และระบบฐานข้อมูล MySQL แล้ว ยังมีการตรวจสอบผลลัพธ์ของฟังก์ชันก่อนจะดำเนินการใดๆ ต่อไป หรือหากไม่ใช้การตรวจสอบตามตัวอย่างข้างต้น อาจใช้เทคนิค Error Suppression ร่วมกับฟังก์ชัน die() หรือ exit() มีตัวอย่างดังนี้

ตัวอย่างที่ 14.3 เทคนิคการประยุกต์ใช้ Error Suppression ร่วมกับฟังก์ชัน die() หรือ exit()

```

1 <?php
2     $link = @mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้: " . mysql_error ());
4     echo "การเชื่อมต่อเสร็จสมบูรณ์";
5 
```

แต่ตัวอย่างไรก็ตาม การใช้ฟังก์ชัน mysql_close() ไม่ใช่สิ่งที่จำเป็น เนื่องจากภาษา PHP จะปิดการเชื่อมต่อเองโดยอัตโนมัติอยู่แล้วเมื่อสิ้นสุดสคริปต์ของเว็บเพจนั้นๆ จึงทำให้การเขียนสคริปต์คำสั่งทำได้สะดวกขึ้น เพราะในบางครั้งการเปิดการเชื่อมต่อ กับระบบฐานข้อมูล MySQL จะขึ้นอยู่กับเงื่อนไขต่างๆ ที่กำหนด ดังนั้นจึงไม่ต้องมาค่อยสั่งปิดการเชื่อมต่อตามเงื่อนไขนั้นๆ

14.2 ฟังก์ชันสำหรับการเลือกใช้งานฐานข้อมูล

การที่จะเรียกใช้งานตารางฐานข้อมูลใดๆ ได้นั้น จะเป็นต้องเลือกฐานข้อมูลที่มีตารางฐานข้อมูลนั้นบรรจุอยู่ก่อน หากใช้คำสั่งโดยตรง ก็คือ การใช้คำสั่ง USE แล้วตามด้วยฐานข้อมูลที่ต้องการเลือกใช้ สำหรับภาษา PHP มีฟังก์ชันที่ใช้สำหรับเลือกใช้งานฐานข้อมูล คือ mysql_select_db() มีรูปแบบดังนี้

รูปแบบ

```
bool mysql_select_db ( string $database_name [, resource $link_identifier ] )
```

เมื่อ \$database_name หมายถึง ชื่อฐานข้อมูลที่ต้องการเลือกใช้งาน

\$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.4 แสดงตัวอย่างการใช้งานฟังก์ชัน mysql_select_db()

```

1 <?php
2     $link = mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) {
4         die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้ : " . mysql_error ());

```

```

5      }
6
7      $db_selected = mysql_select_db ("employees", $link);
8
9      if (!$db_selected) {
10         die ("ไม่สามารถเลือกใช้งานฐานข้อมูล employees ได้ : " . mysql_error ());
11     }
12
13   ?>

```

หากไม่ใช้ฟังก์ชัน mysql_select_db() สามารถประยุกต์ใช้ฟังก์ชัน mysql_query() แทนได้ ระบุชื่อฐานข้อมูลที่ต้องการเลือกใช้งานโดยตรงไปยังระบบฐานข้อมูล MySQL โดยใช้คำสบคาม USE เล้าตามด้วยชื่อฐานข้อมูลที่ต้องการเลือกใช้งาน มีตัวอย่างดังนี้

ตัวอย่างที่ 14.5 การใช้ฟังก์ชัน mysql_query() แทน ฟังก์ชัน mysql_select_db()

```

1 <?php
2
3     $link = mysql_pconnect ("localhost", "mysql_user", "mysql_password");
4
5     if (!$link) {
6
7         die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้ : " . mysql_error ());
8
9     }
10
11   $db_selected = mysql_query ("USE employees;" , $link);
12
13   if (!$db_selected) {
14
15       die ("ไม่สามารถเลือกฐานข้อมูล employees ได้ : " . mysql_error ());
16
17   }
18
19   ?>

```

การใช้ฟังก์ชัน mysql_query() จะมีความสามารถการทำงานได้เร็วกว่าการใช้ฟังก์ชัน mysql_select_db() เพราะเป็นการใช้คำนิยมข้อมูลโดยตรงไปยังระบบฐานข้อมูล MySQL สำหรับรายละเอียดของฟังก์ชัน mysql_query() จะกล่าวถึงในลำดับต่อไป

14.3 การส่งคำสบคามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL

การส่งคำสบคามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL การที่จะเลือกใช้คำสบคามหรือคำสั่งอะไรนั้น ขึ้นอยู่กับว่าต้องการให้ระบบฐานข้อมูล MySQL ทำงานอะไร เช่น การสบคามเพื่อเรียกดูข้อมูล การใช้คำสั่งเพื่อเพิ่มข้อมูล เปลี่ยนแปลงค่าข้อมูล เป็นต้น (ศึกษาเพิ่มเติมได้ในบทที่ 13) ทั้งนี้ภาษา PHP ไม่ได้เป็นผู้ประมวลผลคำสบคามหรือคำสั่ง SQL เอง แต่เป็นเพียงผู้ส่งคำสบคามหรือคำสั่ง มาตรฐาน SQL ที่กำหนดขึ้นในรูปแบบข้อความไปยังระบบฐานข้อมูล MySQL เท่านั้น การที่จะสามารถใช้คำสบคามหรือคำสั่งมาตรฐาน SQL ได้นั้น จำเป็นต้องใช้ฟังก์ชัน mysql_query() สำหรับการส่งคำสบคามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL มีรูปแบบดังนี้



รูปแบบ

```
resource mysql_query ( string $query [, resource $link_identifier = NULL ] )
```

เมื่อ \$query หมายถึง คำสอ逼ถามหรือคำสั่งที่ต้องการใช้งาน

\$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.6 การส่งคำสอ逼ถามเพื่อเรียกคุ้ข้อมูลไปยังระบบฐานข้อมูล MySQL

```
1 <?php
2     $host = "localhost";
3     $user = "parinya";
4     $password = "12345";
5     $link = mysql_pconnect ($host, $user, $password);
6     $db_selected = mysql_select_db ("company", $link);
7     $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8     $result = mysql_query ($query, $link);
9     if (!$result) die ("Can not Query because: " . mysql_error ());
10    echo "<table border=1 width=50%>";
11    while ($row = mysql_fetch_assoc ($result)) {
12        echo "<tr>";
13        echo "<td> " . $row ["firstname"] . "</td>";
14        echo "<td> " . $row ["lastname"] . "</td>";
15        echo "<td> ". $row ["address"] . "</td>";
16        echo "</tr>";
17    }
18    echo "</table>";
19    mysql_free_result ($result);
20 ?>
```

ผลลัพธ์

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

จากตัวอย่างที่ 14.6 การส่งคำสอบถามเพื่อเรียกดูข้อมูลไปยังระบบฐานข้อมูล MySQL อธิบายดังนี้ (ตัวอย่างที่ 14.6 และผลลัพธ์ที่ได้ จะใช้ประกอบการบรรยายการใช้งานฟังก์ชันสำหรับอ่านข้อมูล ผลลัพธ์ ในหัวข้อถัดไป)

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$host มีค่าเท่ากับ "localhost"

บรรทัดที่ 3 กำหนดให้ตัวแปร \$user มีค่าเท่ากับ "parinya" (ชื่อผู้ใช้ที่มีสิทธิใช้งานฐานข้อมูล)

บรรทัดที่ 4 กำหนดให้ตัวแปร \$password มีค่าเท่ากับ "12345"

บรรทัดที่ 5 กำหนดให้ตัวแปร \$link ทำหน้าที่เป็น MySQL link identifier รับค่าที่ส่งมาจากการฟังก์ชัน mysql_pconnect () (ใช้สำหรับเชื่อมต่อระบบฐานข้อมูล MySQL) ระบุพารามิเตอร์ 3 ค่า ประกอบด้วย 1) ชื่อเซิร์ฟเวอร์ 2) ชื่อผู้ใช้ที่มีสิทธิใช้งานฐานข้อมูล และ 3) รหัสผ่าน

บรรทัดที่ 6 กำหนดให้ตัวแปร \$db_selected รับค่าที่ส่งมาจากการฟังก์ชัน mysql_select_db () ใช้สำหรับเลือกฐานข้อมูล ระบุพารามิเตอร์ 2 ค่า ประกอบด้วย 1) ชื่อฐานข้อมูล และ 2) ตัวแปรที่ทำหน้าที่ MySQL link identifier

บรรทัดที่ 7 กำหนดให้ตัวแปร \$query รับค่าที่ส่งมาจากการฟังก์ชัน sprintf () ใช้สำหรับต่อประโยคข้อความให้เป็นประโยคเดียวกัน

บรรทัดที่ 8 กำหนดให้ตัวแปร \$result รับค่าที่ส่งมาจากการฟังก์ชัน mysql_query () ใช้สำหรับส่งคำสอบถามไปยังระบบฐานข้อมูล MySQL โดยระบุค่าพารามิเตอร์ 2 ค่า ประกอบด้วย 1) คำสอบถาม และ 2) MySQL link identifier

บรรทัดที่ 9 ตรวจสอบผลของการสอบถามข้อมูลจากตัวแปร \$result โดยใช้โครงสร้างเงื่อนไข if ในการตรวจสอบ หากไม่สามารถสอบถามข้อมูลได้ สั่งให้หยุดการทำงานบรรทัดที่เหลือทั้งหมดด้วยฟังก์ชัน die () และแสดงเหตุผลที่ไม่สามารถสอบถามข้อมูลได้

บรรทัดที่ 10 เริ่มต้นแท็ก <table> (ใช้สำหรับแสดงข้อมูลให้เป็นระเบียบ)

บรรทัดที่ 11 ใช้โครงสร้างเงื่อนไขทำซ้ำ while โดยจะวนรอบอ่านข้อมูลไปเรื่อยๆ มีเงื่อนไขการอ่านค่าดังนี้ กำหนดให้ตัวแปร \$row รับค่าที่ส่งมาจากการฟังก์ชัน mysql_fetch_assoc (\$result) ซึ่งฟังก์ชัน mysql_fetch_assoc () ใช้สำหรับอ่านค่าชุดของข้อมูลครั้งละ 1 แถว หรือ 1 ระเบียบ โครงสร้างเงื่อนไขทำซ้ำ while จะหยุดก็ต่อเมื่อไม่มีชุดข้อมูลให้อ่านและไปทำการสั่งในบรรทัดที่ 18 หากยังสามารถอ่านชุดข้อมูลได้ ไปทำการสั่งบรรทัดที่ 12 ถึงบรรทัดที่ 17

บรรทัดที่ 12 เริ่มต้นแท็ก <tr>

บรรทัดที่ 13 เริ่มต้นแท็ก <td> และแสดงค่าของตัวแปร \$row [คีย์ firstname] ปิดแท็ก </td>

บรรทัดที่ 14 เริ่มต้นแท็ก <td> และแสดงค่าของตัวแปร \$row [คีย์ lastname] ปิดแท็ก </td>

บรรทัดที่ 15 เริ่มต้นแท็ก <td> และแสดงค่าของตัวแปร \$row [คีย์ address] ปิดแท็ก </td>

บรรทัดที่ 16 ปิดแท็ก </tr>



บรรทัดที่ 17 สินสุดสคริปต์โครงสร้างเงื่อนไขทำซ้ำ while กลับไปตรวจสอบเงื่อนไขบรรทัดที่ 11

บรรทัดที่ 18 ปิดแท็ก </table>

บรรทัดที่ 19 ฟังก์ชัน mysql_free_result() ใช้สำหรับคืนหน่วยความจำให้กับระบบ

บรรทัดที่ 20 สิ้นสุดสคริปต์ PHP

ฟังก์ชัน mysql_query() จะใช้คำสอบถามหรือคำสั่งในรูปแบบข้อความ ดังนั้นจึงสามารถใช้ร่วมกับฟังก์ชัน sprintf() เพื่อต่อข้อความให้เป็นประโยคเดียวกัน ก่อนส่งเข้าไปประมวลผลในฟังก์ชัน mysql_query() และผลลัพธ์ที่ได้จากฟังก์ชันนี้จะเป็นอะไรนั้น ขึ้นอยู่กับคำสอบถามหรือคำสั่งที่กำหนด

สำหรับผลลัพธ์ที่คืนกลับมาจากการใช้ฟังก์ชัน mysql_query() นั้นจะขึ้นอยู่กับคำสอบถามหรือคำสั่งที่ส่งเข้าไปในฟังก์ชัน สามารถสรุปได้ดังนี้

14.3.1 หากเป็นคำสอบถามสำหรับการอ่านข้อมูล เช่น SELECT, SHOW, DESCRIBE หากทำงานสำเร็จจะคืนค่ากลับมาเป็นชุดข้อมูล ผลลัพธ์ที่ได้หรือเรียกว่า result set

14.3.2 หากเป็นคำสั่งในกรณีอื่นๆ เช่น การเปลี่ยนแปลงข้อมูล (INSERT, UPDATE, DELETE) หากการทำงานสำเร็จจะคืนค่า true

14.3.3 หากเกิดข้อผิดพลาดจากการใช้ฟังก์ชัน mysql_query() เช่น เขียนคำสอบถามหรือคำสั่งผิด จะคืนค่า false

14.4 การอ่านข้อมูลผลลัพธ์

หลังจากการใช้ฟังก์ชัน mysql_query() หากคำสอบถามเกี่ยวข้องกับการค้นหาข้อมูล ข้อมูลที่ได้จะเป็นผลลัพธ์มาจากขอบเขตข้อมูลใดๆ เรียกว่า result set โดยข้อมูลนี้จะถูกเก็บไว้ในตัวแปรที่ใช้รับค่าที่ส่งมาจากการใช้ฟังก์ชัน mysql_query() สำหรับการอ่านข้อมูลนั้นสามารถประยุกต์ใช้ได้หลายฟังก์ชัน ตัวอย่างฟังก์ชันที่ใช้สำหรับอ่านข้อมูลผลลัพธ์ มีดังนี้

14.4.1 ฟังก์ชัน mysql_num_rows()

เป็นฟังก์ชันที่ใช้สำหรับนับจำนวนแถวผลลัพธ์ มีรูปแบบดังนี้

รูปแบบ

```
int mysql_num_rows( resource $result )
```

เมื่อ \$result หมายถึง ตัวแปรที่เก็บผลลัพธ์ที่ได้จากการใช้ฟังก์ชัน mysql_query()

ตัวอย่างที่ 14.7 การใช้ฟังก์ชัน mysql_num_rows()

```
1 <?php  
2     $host = "localhost";  
3     $user = "parinya";  
4     $password = "12345";
```

```

5      $link = mysql_pconnect ($host, $user, $password);
6      $db_selected = mysql_select_db ("company", $link);
7      $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8      $result = mysql_query ($query, $link);
9      if (!$result) die ("Can not Query because: " . mysql_error ());
10     echo "ข้อมูลที่อ่านได้มีทั้งหมด " . mysql_num_rows ($result) . " ระเบียน";
11     mysql_free_result ($result)
12 ?>

```

ผลลัพธ์ (อ้างอิงจากตัวอย่างที่ 14.6 และผลลัพธ์)

ข้อมูลที่อ่านได้มีทั้งหมด 3 ระเบียน

14.4.2 พังก์ชัน mysql_result ()

พังก์ชันสำหรับการอ่านผลลัพธ์แบบระบบแ奎และคอลัมน์ วิธีนี้เป็นการอ่านข้อมูลโดยใช้พังก์ชันในการดึงข้อมูล result set ในคอลัมน์ และลำดับแ奎ที่ระบุโดยมีรูปแบบดังนี้

```
string mysql_result ( resource $result , int $row [, mixed $field = 0 ] )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากการพังก์ชัน mysql_query ()

\$row หมายถึง แ夸ของข้อมูล

\$field หมายถึง ขอบเขตข้อมูลหรือคอลัมน์ของข้อมูล (เป็นตัวเลขหรือข้อความก็ได้)

ตัวอย่างที่ 14.8 การใช้พังก์ชัน mysql_result ()

```

1 <?php
2
3     $host = "localhost";
4     $user = "parinya";
5     $password = "12345";
6
7     $link = mysql_pconnect ($host, $user, $password);
8     $db_selected = mysql_select_db ("company", $link);
9
10    $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
11
12    $result = mysql_query ($query, $link);
13    if (!$result) die ("Can not Query because: " . mysql_error ());
14    echo mysql_result ($result, 1);
15    echo "<br>";

```



```
12     echo mysql_result ($result, 2);  
13     mysql_free_result ($result)  
14     ?>
```

ผลลัพธ์ (อ้างอิงจากตัวอย่างที่ 14.6 และผลลัพธ์)

Jaidee

Dennis

14.4.3 พังก์ชัน mysql_fetch_row ()

การอ่านผลลัพธ์แบบตามลำดับແກ່ เป็นการอ่านแบบເລືອນພອຍນ໌ເຕອຮ (Pointer) ໄປຢັງ
ຕາມແນ່ນທີ່ຕ້ອງການອ່ານ ແລ້ວອ່ານຂໍ້ມູນທີ່ພອຍນ໌ເຕອຮຂໍ້ອ່າງ ດ້ວຍພັກໆສັນ mysql_fetch_row () ຂໍ້ມູນທີ່
ອ່ານໄດ້ຈະອ່າງໃນຮູບແບບຂອງອາຣເຣຍເຮັດວຽກລຳດັບຕາມອິນເດີກ໌ ຮູບແບບຂອງພັກໆສັນມີດັ່ງນີ້

ຮູບແບບ

```
array mysql_fetch_row ( resource $result )
```

ເນື້ອ \$result ໝາຍຖື່ງ ຕັ້ງແປຣທີ່ໃຊ້ເກີບຜຸລັພົກທີ່ໄດ້ຈາກພັກໆສັນ mysql_query ()

ຕັ້ງອ່າງໃໝ່ 14.9 ການໃຊ້ພັກໆສັນ mysql_fetch_row ()

```
1 <?php  
2     $host = "localhost";  
3     $user = "parinya";  
4     $password = "12345";  
5     $link = mysql_pconnect ($host, $user, $password);  
6     $db_selected = mysql_select_db ("company", $link);  
7     $query = sprintf ("SELECT firstname, lastname, address FROM employees  
WHERE address='%s';", "Suratthani");  
8     $result = mysql_query ($query, $link);  
9     if (!$result) die ("Can not Query because: " . mysql_error ());;  
10    echo "<table border=1 width=50%>";  
11    while ($row = mysql_fetch_row ($result)) {  
12        echo "<tr>";  
13        echo "<td> " . $row [0] . "</td>";  
14        echo "<td> " . $row [1] . "</td>";  
15        echo "<td> ". $row [2] . "</td>";  
16        echo "</tr>";
```

```

17 }
18 echo "<table>";
19 mysql_free_result ($result)
20 ?>

```

ผลลัพธ์ (ผลลัพธ์ที่ได้จะเหมือนกับตัวอย่างที่ 14.6)

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

14.4.4 พิมพ์ชั้น mysql_fetch_array ()

การอ่านผลลัพธ์แบบอาร์เรย์เป็นการอ่านข้อมูลจาก result_set ในรูปแบบคีย์และค่า อาร์เรย์ โดยคีย์จะเป็นชื่อของเขตข้อมูล และค่าจะเป็นข้อมูลที่ส่งมาจากฟังก์ชัน mysql_fetch_array () มีรูปแบบดังนี้

```
array mysql_fetch_array ( resource $result [, int $result_type ] )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากการฟังก์ชัน mysql_query () \$result_type หมายถึง กำหนดค่าอินเด็กซ์ของอาร์เรย์ (ผลลัพธ์ที่เก็บอยู่ในหน่วยความจำ) สามารถกำหนดได้ 3 ค่า คือ MYSQL_ASSOC (ใช้ชื่อพิลเด็อนในเด็กซ์), MYSQL_NUM (ใช้ตัวเลขลำดับเป็นอินเด็กซ์) และ MYSQL_BOTH (ใช้ชื่อพิลเด็อนและตัวเลขลำดับเป็นอินเด็กซ์ร่วมกัน) หากไม่กำหนด ตัวแปร \$result_type จะมีอินเด็กซ์เป็นแบบ MYSQL_BOTH เป็นค่าปริยาย

ตัวอย่างที่ 14.10 การใช้ฟังก์ชัน mysql_fetch_array ()

```

1 <?php
2     $host = "localhost";
3     $user = "parinya";
4     $password = "12345";
5     $link = mysql_pconnect ($host, $user, $password);
6     $db_selected = mysql_select_db ("company", $link);
7     $query = sprintf ("SELECT firstname, lastname, address FROM employees
8 WHERE address='%s';", "Suratthani");
9     $result = mysql_query ($query, $link);
10    if (!$result) die ("Can not Query because: " . mysql_error ());

```



```

10 echo "<table border=1 width=50%>";
11 while ($row = mysql_fetch_array ($result, MYSQL_NUM)) {
12     echo "<tr>";
13     echo "<td> " . $row [0] . "</td>";
14     echo "<td> " . $row [1] . "</td>";
15     echo "<td> ". $row [2] . "</td>";
16     echo "</tr>";
17 }
18 echo "<table>";
19 mysql_free_result ($result)
20 ?>

```

ผลลัพธ์ (ผลลัพธ์ที่ได้จะเหมือนกับตัวอย่างที่ 14.6)

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

14.4.5 พังก์ชัน mysql_fetch_object ()

เป็นพังก์ชันที่ใช้สำหรับอ่านข้อมูลแบบօบเจ็คต์ การอ่านแบบนี้หมายความว่าสำหรับผู้ที่สนใจในแนว OOP โดยใช้พังก์ชัน mysql_fetch_object () ผลลัพธ์ที่ได้จะอยู่ในรูปแบบօบเจ็คต์ ทั้งนี้การอ้างถึงขอบเขตข้อมูลต่างๆ จะใช้รูปแบบการอ้างถึงพร้อมเพอร์ตีของօบเจ็คต์ จะใช้หลักการเลื่อนพอยน์เตอร์ไปยังแเรกที่ต้องการอ่านข้อมูล คล้ายรับฟังก์ชันที่ได้ศึกษาผ่านมาแล้ว มีรูปแบบดังนี้

รูปแบบ

```
object mysql_fetch_object ( resource $result )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากพังก์ชัน mysql_query ()

ตัวอย่างที่ 14.11 การใช้พังก์ชัน mysql_fetch_object ()

```

1 <?php
2     $host = "localhost";
3     $user = "parinya";
4     $password = "12345";
5     $link = mysql_pconnect ($host, $user, $password);
6     $db_selected = mysql_select_db ("company", $link);

```

```

7      $query = sprintf ("SELECT firstname, lastname, address FROM employees
8          WHERE address='%s';", "Suratthani");
9
10         $result = mysql_query ($query, $link);
11
12         if (!$result) die ("Can not Query because: " . mysql_error ());
13
14         echo "<table border=1 width=50%>";
15
16         while ($row = mysql_fetch_object ($result)) {
17
18             echo "<tr>";
19
20             echo "<td> " . $row->firstname . "</td>";
21             echo "<td> " . $row->lastname . "</td>";
22             echo "<td> " . $row->address . "</td>";
23
24             echo "</tr>";
25
26         }
27
28         echo "<table>";
29
30         mysql_free_result ($result)
31
32     ?>

```

ผลลัพธ์ (ผลลัพธ์ที่ได้จะเหมือนกับตัวอย่างที่ 14.6)

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

14.5 การเปลี่ยนแปลงข้อมูล

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ประกอบด้วย 1) การแทรกข้อมูลใหม่ (INSERT) 2) การแก้ไขข้อมูล (UPDATE) และ 3) การลบข้อมูล (DELETE) มีแนวทางประยุกต์ใช้ดังนี้ (ใช้ตารางที่ 14.1 เป็นตารางอ้างอิงของเขตข้อมูลเมื่อมีการเปลี่ยนแปลงข้อมูลในรูปแบบต่าง)

ตารางที่ 14.1 ตัวอย่างโครงสร้างตาราง customer

ขอบเขต ข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
cus_id	VARCHAR(5)	NOT NULL		PK	รหัสลูกค้า
name	VARCHAR(40)	-	-	-	ชื่อ-สกุลลูกค้า
address	VARCHAR(40)	-	-	-	ที่อยู่ลูกค้า
email	VARCHAR(40)	-	-	-	อีเมล
tel	VARCHAR(20)	-	-	-	เบอร์โทรศัพท์



สร้างตารางข้อมูล customer มีขั้นตอนดังนี้

ตัวอย่างที่ 14.12 ตัวอย่างการสร้างตาราง customer ในฐานข้อมูล company

```
mysql> USE company;
Database changed
mysql> CREATE TABLE customer (
->   cus_id  VARCHAR (5) NOT NULL PRIMARY KEY,
->   name    VARCHAR (40),
->   address VARCHAR (40),
->   email   VARCHAR (40),
->   tel     VARCHAR (20));
```

เพื่อเพิ่มความเข้าใจให้สามารถประยุกต์ใช้งานได้ ดังนั้นจะอธิบายแต่ละส่วนแยกเป็นเว็บเพจ ตาม
หน้าที่การใช้งาน มีรายละเอียดดังนี้

14.5.1 การแทรกข้อมูลใหม่

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ด้วยรูปแบบการแทรกข้อมูลใหม่นั้น
ก่อนอื่น จะเริ่มด้วยการสร้างหน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูล ดังนี้

- 1) ออกแบบแบบเว็บเพจส่วนนำเข้าข้อมูล (customer_form.php)

ตัวอย่างที่ 14.13 скриปต์เว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูลลูกค้า (customer_form.php)

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>เพิ่มข้อมูลลูกค้าใหม่</title>
</head>
<body>
<p>เพิ่มข้อมูลลูกค้าใหม่</p>
<form id="form1" name="form1" method="POST" action="customer_insert.php">
<table width="550" border="1">
<tr>
<td width="135">รหัสลูกค้า</td>
<td width="249"><label>
<input name="cus_id" type="text" id="cus_id" size="20" maxlength="5" />
</label></td>
```

❶ ไฟล์สคริปต์ที่จะทำงานต่อ

❷ เมธอดที่ใช้เป็น POST

❸ ส่วนนำเข้ารหัสลูกค้า cus_id

```

</tr>
<tr>
    <td>ชื่อ-สกุลลูกค้า</td> ④ ส่วนนำเข้าชื่อลูกค้า name
    <td><label>
        <input name="name" type="text" id="name" size="40" maxlength="40" />
    </label></td>
</tr>
<tr>
    <td>ที่อยู่</td> ⑤ ส่วนนำเข้าที่อยู่ลูกค้า address
    <td><label>
        <input name="address" type="text" id="address" size="40" maxlength="40" />
    </label></td>
</tr>
<tr>
    <td>อีเมล์</td> ⑥ ส่วนนำเข้าอีเมลลูกค้า email
    <td><label>
        <input name="email" type="text" id="email" size="40" maxlength="40" />
    </label></td>
</tr>
<tr>
    <td>เบอร์โทรศัพท์</td>
    <td><label>
        <input name="tel" type="text" id="tel" size="20" maxlength="20" />
    </label></td> ⑦ ส่วนนำเข้าเบอร์โทรศัพท์ tel
</tr>
<tr>
    <td>&nbsp;</td>
    <td><label>
        <input type="submit" name="btnSend" id="btnSend" value="Save" />
    </label></td>
</tr>
</table> ⑧ ปุ่มน้ำส่งข้อมูล

```



```
</form>
</body>
</html>
```

จากตัวอย่างที่ 14.13 หน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูลลูกค้า มีส่วนที่สำคัญ สำหรับใช้เป็นข้อมูลนำส่งไปยัง ск립ต์เพื่อการบันทึกข้อมูลต่อไป อธิบายดังนี้

- ❶ ไฟล์ скриптที่จะทำงานต่อเมื่อกดปุ่มบันทึกข้อมูล
- ❷ ฟอร์มกำหนดรูปแบบวิธีการส่งเมื่อต้องแบบ POST
- ❸ ส่วนนำเข้ารหัสลูกค้า ใช้ชื่อ cus_id
- ❹ ส่วนนำเข้าชื่อลูกค้า ใช้ชื่อ name
- ❺ ส่วนนำเข้าที่อยู่ลูกค้า ใช้ชื่อ address
- ❻ ส่วนนำเข้าอีเมลลูกค้า ใช้ชื่อ email
- ❼ ส่วนนำเข้าเบอร์โทรศัพท์ ใช้ชื่อ tel
- ❽ ปุ่มนำส่งข้อมูล

รหัสลูกค้า	<input type="text"/>	❸
ชื่อ-สกุลลูกค้า	<input type="text"/>	❹
ที่อยู่	<input type="text"/>	❺
อีเมล	<input type="text"/>	❻
เบอร์โทรศัพท์	<input type="text"/>	❼
<input type="button" value="Save"/> ❽		❽

ภาพที่ 14.1 หน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูลลูกค้า

2) หน้าเว็บเพจสำหรับแทรกข้อมูลลูกค้าใหม่ลงในระบบฐานข้อมูล MySQL (customer_insert.php) เมื่อหน้าเว็บเพจนี้ทำงานจะทำการแทรกข้อมูลลูกค้าใหม่ลงในฐานข้อมูล MySQL และใช้ฟังก์ชัน header() ย้ายหน้าเว็บเพจไปยังหน้ารายงานข้อมูลลูกค้า (report.php)

ตัวอย่างที่ 14.14 скриптสำหรับแทรกข้อมูลลูกค้าใหม่ (customer_insert.php)

```
<?php
ob_start();
$host = "localhost";
$user = "parinya";
$password = "12345";
```

```

$link = mysql_pconnect ($host, $user, $password);
$db_selected = mysql_select_db ("company", $link);
$query = sprintf ("INSERT INTO customer VALUES ('%s', '%s', '%s', '%s', '%s')",
$_POST["cus_id"], $_POST["name"], $_POST["address"], $_POST["email"], $_POST["tel"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
header('Location: report.php');
?>

```

3) SCRIPT รายงานผลข้อมูลลูกค้าทั้งหมด (report.php) รายละเอียด ดังนี้

ตัวอย่างที่ 14.15 SCRIPT สำหรับรายงานผลข้อมูลลูกค้าทั้งหมด

```

<?php
$host = "localhost";
$user = "parinya";
$password = "12345";
$link = mysql_pconnect ($host, $user, $password);
$db_selected = mysql_select_db ("company", $link);
$query = sprintf ("SELECT * FROM customer");
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>รายงานข้อมูลลูกค้า</title>
</head>
<body>
<p>รายงานข้อมูลลูกค้า</p>
<table width="70%" border="1">
<tr>
<td width="10%">รหัสลูกค้า</td>
<td width="20%">ชื่อ-สกุลลูกค้า</td>
<td width="20%">ที่อยู่</td>

```



```

<td width="20%">>อีเมล</td>
<td width="15%">>เบอร์โทรศัพท์</td>
<td width="15%">>&ampnbsp</td>
</tr>

<?php
    while ($row = mysql_fetch_assoc ($result)) {
        echo "<tr>";
        echo "<td>" . $row ["cus_id"] . "</td>";
        echo "<td>" . $row ["name"] . "</td>";
        echo "<td>" . $row ["address"] . "</td>";
        echo "<td>" . $row ["email"] . "</td>";
        echo "<td>" . $row ["tel"] . "</td>";
        printf("<td>[ <a href = 'customer_edit.php?cus_id=%s'=>แก้ไข</a> ]", $row
        ["cus_id"]);
        printf("[ <a href = 'customer_del.php?cus_id=%s'=>ลบ</a> ]</td>", $row
        ["cus_id"]);
        echo "</tr>";
    }
?>
</table>
<p>&ampnbsp</p>
</body>
</html>

<?php
    mysql_free_result ($result);
?>

```

เขื่อมโยงไปยังหน้าแก้ไข
customer_edit.php
และส่งรหัสลูกค้าไปด้วย

เขื่อมโยงไปยังหน้าลบ
customer_del.php
และส่งรหัสลูกค้าไปด้วย

รหัสลูกค้า	ชื่อ-สกุลลูกค้า	ที่อยู่	อีเมล	เบอร์โทรศัพท์
00001	นวลพรรณ โพธิ์ศรี	บุษบรา	teacher@sru.ac.th	077233466
00002	สิริวัฒนา น้อยดอนไพร	บุษบรา	mon@sru.ac.th	077433009
00003	กิตติ ใจดี	สาระนุรชานี	kit@sru.ac.th	

ภาพที่ 14.2 แสดงหน้าเว็บเพจรายงานข้อมูลลูกค้าทั้งหมด

14.5.2 การแก้ไขข้อมูล

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ด้วยรูปแบบการแก้ไขข้อมูล หน้าแก้ไขข้อมูลลูกค้า (customer_edit.php) รายละเอียด ดังนี้

ตัวอย่างที่ 14.16 скриптสำหรับแสดงข้อมูลลูกค้าก่อนการแก้ไข

```
<?php
    $host = "localhost";
    $user = "parinya";
    $password = "12345";
    $link = mysql_pconnect ($host, $user, $password);
    $db_selected = mysql_select_db ("company", $link);
    $query = sprintf ("SELECT * FROM customer WHERE cus_id='%s';", $_GET["cus_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
    $row = mysql_fetch_assoc ($result);

?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>แก้ไขข้อมูลลูกค้า</title>
</head>
<body>
<p>แก้ไขข้อมูลลูกค้าใหม่</p>
<form id="form1" name="form1" method="POST" action="edit.php">
    <table width="550" border="1">
        <tr>
            <td width="135">รหัสลูกค้า</td>
            <td width="249"><label>
                <input name="cus_id" type="text" id="cus_id" value="<?php echo $row["cus_id"];
?>" size="20" maxlength="5" />
            </label></td>
        </tr>
        <tr>
```

\$_GET["cus_id"] เป็นตัว
แปรผ่าน URL จะต้องใช้



สคริปต์สำหรับบันทึก
ข้อมูลที่แก้ไข





```
<td>ชื่อ-สกุลลูกค้า</td>
<td><label>
    <input name="name" type="text" id="name" value=<?php echo $row["name"]; ?>" size="40" maxlength="40" />
</label></td>
</tr>
<tr>
    <td>ที่อยู่</td>
    <td><label>
        <input name="address" type="text" id="address" value=<?php echo $row["address"]; ?>" size="40" maxlength="40" />
    </label></td>
</tr>
<tr>
    <td>อีเมล์</td>
    <td><label>
        <input name="email" type="text" id="email" value=<?php echo $row["email"]; ?>" size="40" maxlength="40" />
    </label></td>
</tr>
<tr>
    <td>เบอร์โทรศัพท์</td>
    <td><label>
        <input name="tel" type="text" id="tel" value=<?php echo $row["tel"]; ?>" size="20" maxlength="20" />
    </label></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td><label>
        <input type="submit" name="btnSend" id="btnSend" value="Save" />
    </label></td>
</tr>
```

```

</tr>
</table>
</form>
<p>&nbsp;</p>
</body>
</html>
<?php
    mysql_free_result ($result);
?>

```

รหัสลูกค้า	00001
ชื่อ-สกุลลูกค้า	ນາລພຣະນ ໂພຍົກສີ
ที่อยู่	ອຸປະກອດ
อีเมล	teachen@sru.ac.th
เบอร์โทรศัพท์	077544123
	<input type="button" value="Save"/>

ภาพที่ 14.3 หน้าเว็บเพจแสดงข้อมูลลูกค้าที่ต้องการแก้ไข

5) หน้าเว็บเพจสำหรับ UPDATE ข้อมูลลงฐานข้อมูล MySQL (edit.php) รายละเอียดดังนี้

ตัวอย่างที่ 14.17 スคริปต์สำหรับแก้ไขข้อมูลลูกค้า

```

<?php
    ob_start();
    $host = "localhost";
    $user = "parinya";
    $password = "12345";
    $link = mysql_pconnect ($host, $user, $password);
    $db_selected = mysql_select_db ("company", $link);
    $query = sprintf ("UPDATE customer SET name='%s', address='%s', email='%s',
tel='%s' WHERE cus_id='%s';", $_POST["name"], $_POST["address"], $_POST["email"],
$_POST["tel"], $_POST["cus_id"]);

```



```

$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
header('Location: report.php');

?>

```

14.5.3 การลบข้อมูล

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ด้วยรูปแบบการลบข้อมูล

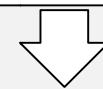
ตัวอย่างที่ 14.18 ศิริป์สำหรับลบข้อมูลลูกค้า

```

<?php
ob_start();
$host = "localhost";
$user = "parinya";
$password = "12345";
$link = mysql_pconnect ($host, $user, $password);
$db_selected = mysql_select_db ("company", $link);
$query = sprintf ("DELETE FROM customer WHERE cus_id='%s';", $_GET["cus_id"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
header('Location: report.php');

?>

```

ส่งรหัสลูกค้ามาจาก
 หน้ารายงานผ่าน URL
 

14.6 การแก้ไขข้อมูลภาษาไทยกลายเป็น "?????...."

สำหรับผู้ที่เกิดปัญหาการแสดงข้อมูลที่เป็นภาษาไทย แล้วกลายเป็น "???...." ให้ลองแก้ปัญหาโดยการกำหนดเซ็ตอักษรใหม่ให้รองรับภาษาไทย ด้วยการส่งคำสั่งต่อไปนี้ก่อนที่จะอ่านข้อมูล

ตัวอย่างที่ 14.19 การแก้ไขข้อมูลภาษาไทยกลายเป็น "?????...."

```

mysql_query ("SET CHARACTER SET tis620");
// หรือใช้ mysql_query ("SET NAMES tis620");

```

14.7 การตรวจสอบการเปลี่ยนแปลงข้อมูล

การตรวจสอบการเปลี่ยนแปลงข้อมูล เป็นสิ่งที่ทำให้พัฒนาเว็บแอปพลิเคชันทราบถึงการเปลี่ยนแปลงต่างๆ ที่เกิดขึ้นครั้งล่าสุดกับระบบฐานข้อมูล MySQL เช่นการ INSERT, DELETE และ UPDATE เป็นต้น การตรวจสอบการเปลี่ยนแปลงที่เกิดขึ้นในกรณีต่างๆ ดังต่อไปนี้

14.7.1 พังก์ชัน mysql_affected_rows ()

เป็นฟังก์ชันที่ใช้สำหรับการนับจำนวนแถวที่เปลี่ยนแปลงข้อมูล โดยถ้าต้องการตรวจสอบว่า มีข้อมูลจำนวนกี่แถว ที่เกิดการเปลี่ยนแปลงจากการใช้คำสั่งข้อมูลครั้งล่าสุด ก็สามารถตรวจสอบได้โดยใช้ฟังก์ชัน mysql_affected_rows () มีรูปแบบดังนี้

รูปแบบ

```
int mysql_affected_rows ([ resource $link_identifier = NULL ] )
```

เมื่อ \$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.20 การใช้ฟังก์ชัน mysql_affected_rows ()

```
1 <?php
2     $link = mysql_connect ('localhost', 'mysql_user', 'mysql_password');
3     if (!$link) {
4         die ('ไม่สามารถเชื่อมตอกับฐานข้อมูลได้ เนื่องจาก: ' . mysql_error ( ));
5     }
6     mysql_select_db ('mydb');
7     mysql_query ('DELETE FROM mytable WHERE id < 10');
8     printf ("Records deleted: %d<br>", mysql_affected_rows ( ));
9 ?>
```

14.7.2 พังก์ชัน mysql_insert_id ()

เป็นฟังก์ชันที่ใช้สำหรับการตรวจสอบ ID ของข้อมูลที่เพิ่มใหม่ จากการที่ขอบเขตข้อมูลแบบ "AUTO_INCREMENT" นั้น ข้อมูลจะถูกเพิ่มต่อจากข้อมูลล่าสุดโดยอัตโนมัติ โดยส่วนใหญ่จะใช้ AUTO_INCREMENT กับข้อมูลที่เป็นค่า ID เนื่องจากจะมีลักษณะการเรียงลำดับที่ต่อเนื่อง และไม่ซ้ำกัน และหลังจากที่เพิ่มข้อมูลลงในตารางแล้ว อาจจะเป็นต้องนำค่า ID นั้นไปใช้งานอีกต่อไป เช่น นำไปใช้กับการประมวลผลอื่นๆ ของ PHP หรือการนำไปเติมกับตารางอื่นๆ ที่เกี่ยวข้องกัน เป็นต้น นี่คือเหตุผลที่จำเป็นต้องทราบว่า แต่ละสุดท้ายเพิ่มลงในตารางมีค่า ID เป็นเท่าไร โดย PHP มีฟังก์ชันในการตรวจสอบค่า ID นี้ได้คือ พังก์ชัน mysql_insert_id () โดยมีรูปแบบดังนี้

รูปแบบ

```
int mysql_insert_id ([ resource $link_identifier = NULL ] )
```

เมื่อ \$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ฟังก์ชันนี้จะคืนค่ากลับมาเป็นข้อมูลในขอบเขตข้อมูลที่เป็น AUTO_INCREMENT ของแถวล่าสุดที่ถูกเพิ่มเข้าไปด้วยคำสั่ง INSERT หรือ REPLACE เช่น สมมติจากเดิมข้อมูลในตารางแรก และขอบเขตข้อมูล id เป็นแบบ AUTO_INCREMENT



ตัวอย่างที่ 14.21 การใช้ฟังก์ชัน mysql_insert_id()

```

1 <?php
2     $link = mysql_connect ('localhost', 'mysql_user', 'mysql_password');
3     if (!$link) {
4         die ('ไม่สามารถเชื่อมต่อ กับฐานข้อมูล เนื่องจาก: ' . mysql_error ());
5     }
6     mysql_select_db ('mydb');
7     mysql_query ("INSERT INTO mytable (product) values ('kossu')");
8     printf ("Last inserted record has id %d<br>", mysql_insert_id ());
9 ?>

```

แต่อย่างไรก็ตาม ต้องเรียกใช้ฟังก์ชัน mysql_insert_id() ทันทีหลังจากเพิ่มข้อมูล ด้วยคำสั่ง INSERT หรือคำสั่ง REPLACE ทั้งนี้เพื่อให้ได้ค่า ID ของแถวล่าสุดที่ถูกเพิ่มเข้าใหม่อย่างแท้จริง

14.7.3 ฟังก์ชัน mysql_data_seek()

เป็นฟังก์ชันที่ใช้สำหรับระบุตำแหน่งของแถวข้อมูลที่ต้องการให้เริ่มต้น นับแถวแรกเป็น 0 โดยมีรูปแบบดังนี้

รูปแบบ

```
bool mysql_data_seek ( resource $result , int $row_number )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากการใช้ฟังก์ชัน mysql_query()

\$row_number หมายถึง ค่าของแถวที่ต้องการเริ่มต้น

ตัวอย่างที่ 14.22 การใช้ฟังก์ชัน mysql_data_seek()

```

1 <?php
2     $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
3     if (!$link) {
4         die ('ไม่สามารถเชื่อมต่อ กับฐานข้อมูลได้ เนื่องจาก: ' . mysql_error ());
5     }
6     $db_selected = mysql_select_db ('sample_db');
7     if (!$db_selected) {
8         die ('ไม่สามารถเลือกฐานข้อมูลได้ เนื่องจาก: ' . mysql_error ());
9     }
10    $query = 'SELECT last_name, first_name FROM friends';
11    $result = mysql_query ($query);
12    if (!$result) {

```

```

13         die('การสอบถามข้อมูลเกิดข้อผิดพลาด เนื่องจาก: ' . mysql_error());
14     }
15     /* เรียกคุณข้อมูลในตำแหน่งลำดับแบบย้อนกลับ */
16     for ($i = mysql_num_rows($result) - 1; $i >= 0; $i--) {
17         if (!mysql_data_seek($result, $i)) {
18             echo "ไม่สามารถระบุตำแหน่งแถวข้อมูลแรกที่ $i: " . mysql_error() . "<br/>";
19             continue;
20         }
21         if (!$row = mysql_fetch_assoc($result)) {
22             continue;
23         }
24         echo $row ['last_name'] . ' ' . $row ['first_name'] . "<br />\n";
25     }
26     mysql_free_result($result);
27 ?>

```

14.8 การตรวจสอบข้อผิดพลาดระหว่างการทำงานร่วมกับ MySQL

เมื่อเกิดข้อผิดพลาดในการทำงาน MySQL ขึ้น หากต้องการทราบว่าข้อผิดพลาดนั้นมาจากสาเหตุอะไร ก็สามารถตรวจสอบได้โดยใช้ฟังก์ชัน `mysql_error()` มีรูปแบบดังนี้

รูปแบบ

```
string mysql_error ([ resource $link_identifier = NULL ] )
```

เมื่อ `$link_identifier` หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier
ในทางปฏิบัติแล้ว มักจะใช้ฟังก์ชันนี้ร่วมกับขั้นตอนที่คาดว่า อาจเกิดข้อผิดพลาดขึ้น และอ่าน
ข้อผิดพลาดนั้นมาแสดง

14.9 การตรวจสอบฐานข้อมูลและตาราง

บางครั้งต้องการทราบว่า ภายในฐานข้อมูล MySQL นั้นประกอบด้วยฐานข้อมูล หรือตารางซึ่ง
อะไรบ้าง ก็จะมีการตรวจสอบดังนี้

14.9.1 การตรวจสอบฐานข้อมูล

ชื่อฐานข้อมูลที่ใช้ตรวจสอบได้ต้องเป็นชื่อของฐานข้อมูลที่เป็นผู้สร้างขึ้นมา หรือมีสิทธิ์ใน
การใช้งาน โดยการตรวจสอบจะใช้ฟังก์ชัน `mysql_list dbs()` ฟังก์ชันนี้จะคืนค่ากลับคืนมาเป็น result
set โดยที่ข้อมูลในแต่ละแถวคือ ชื่อฐานข้อมูลที่อ่านได้ มีรูปแบบดังนี้



รูปแบบ

```
resource mysql_list_dbs ([ resource $link_identifier = NULL ] )
```

เมื่อ \$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

จากการที่ฟังก์ชันนี้คืนมาเป็น result set หากต้องการจะอ่านรายชื่อรากฐานข้อมูลเหล่านี้ จะต้องใช้ฟังก์ชันที่ใช้ในการอ่านข้อมูลจาก result set เช่น mysql_fetch_row () หรือ mysql_fetch_array () หรือ mysql_fetch_object () หรือ mysql_fetch_assoc () เป็นต้น มีตัวอย่างดังนี้

ตัวอย่างที่ 14.23 การตรวจสอบชื่อรากฐานข้อมูล

```
1 <?php
2 // ตัวอย่างการใช้งานแบบทั่วไปถ้าไม่ใช้ฟังก์ชัน mysql_list_dbs ()
3 $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
4 $res = mysql_query ("SHOW DATABASES");
5 while ($row = mysql_fetch_assoc ($res)) {
6     echo $row ['Database'] . "<br/>";
7 }
8 // ตัวอย่างการใช้งานแบบทั่วไปถ้าไม่ใช้ฟังก์ชัน mysql_list_dbs () สำหรับ PHP 5.4.0
9 $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
10 $db_list = mysql_list_dbs ($link);
11 while ($row = mysql_fetch_object ($db_list)) {
12     echo $row->Database . "<br/>";
13 }
14 ?>
```

14.9.2 การตรวจสอบชื่อตารางที่มีอยู่ในฐานข้อมูล

ถ้าต้องการทราบว่า ภายในฐานข้อมูลนั้นมีตารางชื่ออะไรบ้าง ก็สามารถตรวจสอบด้วยฟังก์ชัน mysql_list_tables () มีรูปแบบดังนี้

รูปแบบ

```
resource mysql_list_tables ( string $database [, resource $link_identifier = NULL ] )
```

เมื่อ \$database หมายถึง ชื่อของฐานข้อมูล ต้องกำหนดในแบบข้อความผลลัพธ์ที่ได้จากฟังก์ชันนี้จะอยู่ในรูปแบบของ result set

\$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.24 การตรวจสอบชื่อตารางที่มีอยู่ในฐานข้อมูล

```

1 <?php
2     $dbname = 'mysql_dbname';
3
4     if (!mysql_pconnect ('mysql_host', 'mysql_user', 'mysql_password')) {
5         echo "ไม่สามารถเชื่อมต่อกับฐานข้อมูล MySQL ได้ เนื่องจาก: " . mysql_error ();
6         exit;
7     }
8
9     $sql = "SHOW TABLES FROM $dbname";
10    $result = mysql_query ($sql);
11
12    if (!$result) {
13        echo "เกิดข้อผิดพลาด ไม่พบข้อมูลตารางในฐานข้อมูล $dbname ";
14        exit;
15    }
16
17    while ($row = mysql_fetch_row ($result)) {
18        echo "Table: {$row [0]} <br/>";
19    }
20
21    mysql_free_result ($result);
22
23    ?>
```

14.10 การตรวจสอบโครงสร้างของขอบเขตข้อมูล

การตรวจสอบโครงสร้างของขอบเขตข้อมูลเป็นการอ่านรายละเอียดทั้งหมดเกี่ยวกับขอบเขตข้อมูล ชนิดข้อมูล และแอ็ฟทริบิวต์ต่างๆ สำหรับใน PHP มีฟังก์ชันที่ใช้ในการตรวจสอบองค์ประกอบของขอบเขตข้อมูล ดังต่อไปนี้

14.10.1 การนับจำนวนขอบเขตข้อมูล

การนับจำนวนพิวเดอร์จะนับจำนวนจากขอบเขตข้อมูลที่มีอยู่ใน result set ที่กำหนด เนื่องจากเป็นการนับจำนวนขอบเขตข้อมูลที่มีอยู่ใน \$result ที่ได้จากการใช้ฟังก์ชัน mysql_query () กำหนดให้อ่านข้อมูลจากขอบเขตข้อมูลใดบ้าง ดังนั้นจำนวนขอบเขตข้อมูลที่นับได้ อาจแตกต่างจำนวนจริงที่มีอยู่ในตารางมีรูปแบบดังนี้

รูปแบบ

```
int mysql_num_fields ( resource $result )
```

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากการฟังก์ชัน mysql_query ()



ตัวอย่างที่ 14.25 การนับจำนวนขอบเขตข้อมูล

```
1 <?php
2     $result = mysql_query ("SELECT id, email FROM people WHERE id = '42'");
3     if (!$result) {
4         echo "ไม่สามารถสอบถามข้อมูลได้ เนื่องจาก: ' . mysql_error ( );
5         exit;
6     }
7     echo mysql_num_fields ($result);
8 ?>
```

14.10.2 การตรวจสอบชื่อขอบเขตข้อมูล

การตรวจสอบชื่อขอบเขตข้อมูลในตาราง จะมีฟังก์ชันดังนี้

1) ฟังก์ชัน mysql_list_fields ()

เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบชื่อขอบเขตข้อมูลในตารางของฐานข้อมูลที่กำหนด
มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
resource mysql_list_fields ( string $database_name , string $table_name
[, resource $link_identifier = NULL ] )
```

เมื่อ \$database_name หมายถึง ชื่อฐานข้อมูล

\$table_name หมายถึง ชื่อตาราง

\$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.26 การใช้ฟังก์ชัน mysql_list_fields ()

```
1 <?php
2     $result = mysql_query ("SHOW COLUMNS FROM sometable");
3     if (!$result) {
4         echo "ไม่สามารถสอบถามข้อมูลได้ เนื่องจาก: ' . mysql_error();
5         exit;
6     }
7     if (mysql_list_fields ($result) > 0) {
8         while ($row = mysql_fetch_assoc ($result)) {
9             print_r ($row);
10        }
11    }
```

12 ?>

2) พังก์ชัน mysql_field_name ()

เป็นพังก์ชันที่ใช้สำหรับตรวจสอบชื่อของเขตข้อมูลในตารางของฐานข้อมูลที่กำหนด เช่นเดียวกับพังก์ชัน mysql_list_fields () มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
string mysql_field_name ( resource $result )
```

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากพังก์ชัน mysql_query ()

ตัวอย่างที่ 14.27 การใช้พังก์ชัน mysql_field_name ()

```

1 <?php /* ตัวอย่าง ตารางประกอบด้วย 3 ขอบเขตข้อมูล ดังนี้:
2      * user_id , username และ password */
3
4      $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
5
6      if (!$link) {
7          die ("ไม่สามารถเชื่อมต่อกับ MySQL server เนื่องจาก: " . mysql_error ());
8
9      }
10
11     $dbname = 'mydb';
12
13     $db_selected = mysql_select_db ($dbname, $link);
14
15     if (!$db_selected) {
16         die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูล $dbname ได้เนื่องจาก: " . mysql_error ());
17
18     }
19
20     $res = mysql_query ('select * from users', $link);
21
22     echo mysql_field_name ($res, 0) . "<br/>";
23
24     echo mysql_field_name ($res, 2);
25
26 ?>

```

ผลลัพธ์

```
user_id
```

```
password
```

14.10.3 การตรวจสอบความยาวของขอบเขตข้อมูล

การตรวจสอบความยาวของขอบเขตข้อมูลจะใช้พังก์ชัน mysql_fields_len () มีรูปแบบดังนี้

รูปแบบ

```
int mysql_field_len ( resource $result )
```

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากพังก์ชัน mysql_query ()



ตัวอย่างที่ 14.28 การตรวจสอบความยาวของขอบเขตข้อมูล

```

1 <?php
2     $result = mysql_query ("SELECT id,email FROM people WHERE id = '42'");
3     if (!$result) {
4         echo "ไม่สามารถสอบถามข้อมูลได้ เนื่องจาก: ' . mysql_error ( ); exit;
5     }
6     $length = mysql_field_len ($result, 0); // ตรวจสอบความยาวของขอบเขตข้อมูล "id"
7     echo $length;
8 ?>

```

14.10.4 การตรวจสอบชนิดข้อมูลของขอบเขตข้อมูล

การตรวจสอบชนิดข้อมูลของขอบเขตข้อมูลด้วยฟังก์ชัน mysql_field_type () มีรูปแบบดังนี้

รูปแบบ

```
string mysql_field_type ( resource $result )
```

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากการฟังก์ชัน mysql_query ()

ตัวอย่างที่ 14.29 การตรวจสอบชนิดข้อมูลของขอบเขตข้อมูล

```

<?php
    mysql_pconnect ("localhost", "mysql_username", "mysql_password");
    mysql_select_db ("mysql");
    $result = mysql_query ("SELECT * FROM func");
    $fields = mysql_num_fields ($result);
    $rows = mysql_num_rows ($result);
    $table = mysql_field_table ($result, 0);
    echo "ตาราง " . $table . " มี " . $fields . " ขอบเขตข้อมูล จำนวน " . $rows . " ระเบียน<br/>";
    echo "โดยตารางประกอบด้วยขอบเขตข้อมูลดังนี้:<br/>";
    for ($i=0; $i < $fields; $i++) {
        $type = mysql_field_type ($result, $i);
        $name = mysql_field_name ($result, $i);
        $len = mysql_field_len ($result, $i);
        $flags = mysql_field_flags ($result, $i);
        echo $type . " " . $name . " " . $len . " " . $flags . "<br>";
    }
}

```

```
mysql_free_result ($result);
?>
```

ผลลัพธ์

ตาราง 'func' มี 4 ขอบเขตข้อมูล จำนวน 1 ระเบียน

โดยตารางประกอบด้วยขอบเขตข้อมูลดังนี้:

```
string name 64 not_null primary_key binary
int ret 1 not_null
string dl 128 not_null
string type 9 not_null enum
```

สรุป

ในการเขียนสคริปต์ PHP สำหรับการติดต่อกับฐานข้อมูล MySQL นั้น จะสังเกตุได้ว่า การเริ่มต้น การเขียนสคริปต์นั้น มีขั้นตอนสำคัญ คือ 1) เริ่มจากการติดต่อฐานข้อมูล 2) เลือกฐานข้อมูลที่จะใช้งาน และ 3) การส่งคำสอบถามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL เพื่อประมวลผลตามคำสอบถามหรือคำสั่งที่ส่งเข้าไป ซึ่งอาจเป็นการสอบถามเพื่อเรียกดูข้อมูล หรือคำสั่งเพื่อแทรกข้อมูล แก้ไขข้อมูล และลบ ข้อมูล เป็นต้น นี่คือลักษณะพื้นฐานของการพัฒนาเว็บแอปพลิเคชันที่มีการติดต่อกับฐานข้อมูล จาก ตัวอย่างทั้งหมดที่อธิบายทั้งหมดจะสังเกตุได้ว่าคำสั่งช่วงบรรทัดบนสุดของแต่ละสคริปต์จะมีการเปิดการ เชื่อมต่อ เลือกใช้ฐานข้อมูล มีรูปแบบช้าๆ ในทุกหน้าเว็บเพจ ดังนั้น ผู้พัฒนาเว็บแอปพลิเคชันสามารถ เขียนสคริปต์แยกเป็นไฟล์เฉพาะเก็บไว้ และเมื่อต้องการใช้งานในหน้าเว็บเพจได้สามารถใช้วิธีการนำเข้าไฟล์เพื่อประมวลผล จะทำให้ช่วยลดบรรทัดสำหรับการเขียนสคริปต์ และสะดวกในการใช้งาน

คำถามท้ายบท

1. จงอธิบายหน้าที่ รูปแบบ และความแตกต่าง ระหว่างฟังก์ชัน mysql_connect () และ mysql_pconnect ()
2. จงอธิบายหน้าที่ของฟังก์ชัน mysql_query () พิจารณาด้วย mysql_select_db ()

ชื่อตาราง employees (พนักงาน) ใช้ประกอบการตอบคำถามข้อที่ 3 -5

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889



3. จงอธิบายหลักการทำงานของสคริปต์ และผลลัพธ์ที่ได้ จากสคริปต์่อไปนี้

```
1 <?php
2     $link = mysql_pconnect ("localhost", "username", "password");
3     $db_selected = mysql_select_db ("company", $link);
4     $query = "SELECT firstname, position, address FROM employees WHERE
position= " . "Programmer";
5     $result = mysql_query ($query, $link);
6     if (!$result) die ("Can not Query because: " . mysql_error ());
7         echo "<table border=1 width=50%>";
8     while ($row = mysql_fetch_assoc ($result)) {
9         echo "<tr>";
10        echo "<td> " . $row ["firstname"] . "</td>";
11        echo "<td> " . $row ["position"] . "</td>";
12        echo "<td> " . $row ["address"] . "</td>";
13        echo "</tr>";
14    }
15    echo "<table>";
16    mysql_free_result ($result);
17 ?>
```

4. สคริปต์จากข้อที่ 3. หากลบบรรทัดที่ 7 - 15 ออก แล้วเพิ่มฟังก์ชัน echo mysql_affected_rows (\$link) จะได้ผลลัพธ์อย่างไร และอธิบายเหตุผลประกอบ

5. จงเขียนสคริปต์ PHP เพื่อสำหรับเรียกดูขอบเขตข้อมูลพนักงานทั้งหมด โดยมีเงื่อนไขว่าดูข้อมูลเฉพาะพนักงานที่มีช่วงเงินเดือนระหว่าง 18000 – 30000 บาทเท่านั้น

បរណានៃក្រម

บรรณานุกรม

- กิตติ ภักดีวัฒนาภูล และจันทร์ขอ แซ่ อุ่น. (2552). **PHP**. พิมพ์ครั้งที่ 12. กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลต์.
- กิตติศักดิ์ เจริญโภคานนท์. (2550). **คู่มือเรียนเขียนเว็บอีคอมเมิร์ชด้วย PHP 5**. พิมพ์ครั้งที่ 2. กรุงเทพฯ: ชั้นเชิง มีเดีย.
- เจริญศักดิ์ รัตนาราห แลและสันต์ พิพย์ศุภรณนท์. (2554). **PHP & MySQL**. กรุงเทพฯ: เน็ตดีไซน์ พับลิชชิ่ง.
- ธันยพัฒน์ วงศ์รัตน์. (2554). **เรียนลัดสร้างเว็บแอปพลิเคชันด้วย PHP & MySQL ฉบับ Workshops**. กรุงเทพฯ: สวัสดี ไอที.
- เนคเทค. (2556). **Ajax**. [Online]. เข้าถึงได้จาก : <http://wiki.nectec.or.th/gitiwiki/pub/Knowledge/Ajax>. [2556, พฤษภาคม 5].
- บัญชา ประสีลະเตสัง. (2553). **พัฒนาเว็บแอปพลิเคชันด้วย PHP ร่วมกับ MySQL และ Dreamweaver**. กรุงเทพฯ: ชีเอ็ดดี้เคชั่น.
- _____ . (2551). **พัฒนาเว็บด้วยเทคนิค Ajax และ PHP**. กรุงเทพฯ: ชีเอ็ดดี้เคชั่น.
- ภาควิชาบริการคอมพิวเตอร์. (2553). **Introduction to MySQL Database Server**. เชียงใหม่: คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่.
- ภานุพงศ์ ปัญญาดี. (2549). **AppServ คืออะไร**. [Online]. เข้าถึงได้จาก : <http://www.appservnetwork.com/modules.php?name=Content&pa=showpage&pid=3>. [2558, เมษายน 1].
- มาตรฐานกุณภานุพงศ์ ปัญญาดี ผู้พัฒนา AppServ. (2552). [Online]. เข้าถึงได้จาก : <http://www.sutenm.com/tag/ภานุพงศ์-ปัญญาดี/>. [2556, พฤษภาคม 5].
- วิกิพีเดีย สารานุกรมเสรี. (2556). **วินาที**. [Online]. เข้าถึงได้จาก : <http://th.wikipedia.org/wiki/วินาที>. [2556, พฤษภาคม 5].
- สาธิ ชัยวัฒน์ตระกูล. (2550). **เก่ง PHP ให้ครบสูตร**. กรุงเทพฯ: วิตตี้กรุ๊ป.
- สุธีร นาวกุล. (2555). **ออกแบบและสร้างเว็บไซต์ด้วย Dreamweaver CS6**. กรุงเทพฯ: ชีเอ็ดดี้เคชั่น.
- อนรรชนนค์ คุณณี. (2555). **พัฒนา Web App แบบมืออาชีพด้วย PHP + AJAX และ jQuery**. นนทบุรี: ไอเดีย พรีเมียร์.
- Chris Ullman and Lucinda Dykes. (2007). **Beginning Ajax**. Canada: Wiley Publishing.
- Declan McCullagh. (2007). **David Axmark, co-founder of MySQL, at open source government conference**. [Online]. Available : <http://www.mccullagh.org/image/10d-1/david-axmark-mysql.html>. [2012, October 1].

បរណ្ណានុករម (ពេទ្យ)

- Hege Refsnes, Stale Refsnes, Kai Jim Refsnes, and Jan Egil Refsnes with Kelly Dobbs Henthorne. (2010). **Learn JavaScript and AJAX with w3schools**. Canada: Wiley Publishing.
- Helsinki, Finland. (2009). **Michael Widenius, founder of MySQL, invests in WOT, Web of Trust**. [Online]. Available : http://www.prweb.com/releases/Michael_Widenius/Web_of_Trust/prweb2009984.htm. [2012, October 1].
- History of MySQL. (2011). **MySQL 5.1 Reference Manual**. MySQL AB. [Online]. Available : <http://dev.mysql.com/doc/refman/5.1/en/history.html>. [2015, April 1].
- Jay Greenspan and Brad Bulger. (2001). **MySQL/PHP Database Applications**. Chicago, New York: IDG Books Worldwide.
- Kevin Yank. (2004). **Build Your Own Database Driven Website Using PHP and MySQL**. 3rd Edition, United States of America: SitePoint Pty.
- Luke Welling and Laura Thomson. (2001). **PHP and MySQL Web Development**. United States of America: SAMS.
- Markbuchner. (2010). **Zeev Suraski, co-founder of Zend is TEC 2010 Keynote!**. [Online]. Available : <http://tec20ten.wordpress.com/page/2/>. [2012, October 1].
- Oswald. (2011). **About Me**. [Online]. Available : <http://oswaldatwork.thetaoofamp.com/>. [2012, October 1].
- Peter Judge. (2010). **PHP Forever – Andi Gutmans Of Zend**. [Online]. Available : <http://www.techweekeurope.co.uk/interview/php-forever-andi-gutmans-of-zend-120288>. [2012, October 1].
- Rasmus. (2011). **Rasmus' Toys Page**. [Online]. Available : <http://toys.lerdorf.com>. [2012, October 1].
- Shelley Powers. (2007). **Adding Ajax**. United States of America: O'Reilly Media.
- The PHP Group. (2012). **PHP Manual**. [Online]. Available : <http://www.php.net>. [2012, October 1].
- The phpMyAdmin devel team. (2012). **Documentation**. [Online]. Available : http://www.phpmyadmin.net/home_page/docs.php. [2012, December 4].
- Tim Converse and Joyce Park with Clark Morgan. (2004). **PHP5 and MySQL Bible**. Canada: Wiley Publishing.

បរណ្ណានុករម (ពេទ)

W. Jason Gilmore. (2008). **Beginning PHP and MySQL: From Novice to Professional**,

Third Edition. United States of America: Apress.

Week of Innovative Regions in Europe. (2007). **Lecturers**. [Online]. Available : <http://www.wire2011.eu/lecturer/list?fl=l>. [2012, October 1].

ภาคผนวก



ภาคผนวก ก
เครื่องมือที่ใช้สำหรับเขียนโปรแกรมบนเว็บ



ภาคผนวก ก

เครื่องมือที่ใช้สำหรับเขียนโปรแกรมบนเว็บ

การเขียนโปรแกรมบนเว็บ หรือเรียกว่าเว็บแอพพลิเคชัน คือ การเขียนโปรแกรมประยุกต์ที่สามารถเข้าถึงด้วยโปรแกรมเว็บบราวเซอร์ผ่านระบบเครือข่ายคอมพิวเตอร์ เช่น อินเทอร์เน็ต หรือ อินทราเน็ต เป็นต้น เว็บแอพพลิเคชันเป็นที่นิยมในปัจจุบันเนื่องจากความสามารถในการปรับปรุงระบบให้ทันสมัยอยู่ตลอดเวลา การดูแลรักษาทำได้ง่าย เสียค่าใช้จ่ายน้อย และไม่จำเป็นต้องติดตั้งซอฟต์แวร์บนเครื่องผู้ใช้งาน ตัวอย่างเว็บแอพพลิเคชัน เช่น เว็บเมล์ เว็บพาณิชย์อิเล็กทรอนิกส์ เว็บประมูลออนไลน์ กระดานสนทนากลุ่ม วิกิ และเฟสบุ๊ก เป็นต้น

องค์ประกอบของการพัฒนาเว็บแอพพลิเคชันด้วยภาษา PHP ร่วมกับฐานข้อมูล MySQL ประกอบด้วย 1) โปรแกรมจำลองเว็บเซิร์ฟเวอร์ 2) เครื่องมือสำหรับการเขียนสคริปต์ 3) เว็บบราวเซอร์ และ 4) โปรแกรมโอนย้ายไฟล์

1.1 โปรแกรมจำลองเว็บเซิร์ฟเวอร์

โปรแกรมจำลองเว็บเซิร์ฟเวอร์ คือ โปรแกรมที่ใช้จำลองเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) ให้เป็นเว็บเซิร์ฟเวอร์ สำหรับทดสอบสคริปต์ภาษา PHP ร่วมกับฐานข้อมูล MySQL ภายใต้เครื่องของผู้ใช้งาน แล้วหลังจากที่ผู้พัฒนาเว็บแอพพลิเคชันเสร็จเรียบร้อยแล้ว ก็จะนำขึ้นไปไว้บนเว็บเซิร์ฟเวอร์ที่ใช้งานจริงต่อไป โปรแกรมจำลองเว็บเซิร์ฟเวอร์เป็นการรวมโปรแกรมที่มีความจำเป็นสำหรับการพัฒนาเว็บแอพพลิเคชันเข้าไว้ด้วยกัน ประกอบด้วย 1) Apache HTTP Server 2) PHP 3) MySQL และ 4) PHPMyAdmin ซึ่งปกติแล้วโปรแกรมเหล่านี้จะต้องติดตั้งร่วมกัน ทำให้เสียเวลาและการตั้งค่าคอนฟิก (Configuration) ค่อนข้างมาก ในปัจจุบันมีโปรแกรมจำลองเว็บเซิร์ฟเวอร์ให้เลือกใช้หลายโปรแกรม เช่น AppServ, XAMPP, PHPThai Localhost, AMPPS และ Alchemy Eye เป็นต้น สำหรับในหนังสือเล่มนี้จะแนะนำรายละเอียดโปรแกรมจำลองเว็บเซิร์ฟเวอร์ 2 โปรแกรม เนื่องจากได้รับความนิยมใช้ในประเทศไทย คือ AppServ และ XAMPP มีรายละเอียดดังนี้

1.1.1 AppServ

โปรแกรม AppServ พัฒนาโดย ภาณุพงศ์ ปัญญาดี พัฒนาครั้งแรกเมื่อปี พ.ศ. 2543 เป็นการรวมโปรแกรมจำนวน 4 โปรแกรม สำหรับจำลองเครื่องคอมพิวเตอร์ให้เป็นเว็บเซิร์ฟเวอร์ ประกอบด้วย Apache HTTP Server, PHP, MySQL, และ phpMyAdmin เวอร์ชันปัจจุบันได้แก่ 2.5.10 (สำหรับ PHP 5) และ 2.6.0 (สำหรับ PHP 6) เหตุผลที่พัฒนาโปรแกรม AppServ เพราะผู้พัฒนาต้องพยายามให้กับเพื่อนถึงวิธีการติดตั้ง Apache, PHP, และ MySQL อยู่บ่อยครั้ง จึงเป็นจุดเริ่มต้นในการพัฒนาชุดติดตั้งดังกล่าว และตั้งชื่อว่า AppServ จุดเด่น คือ ติดตั้งและใช้งานได้ทันที ปัจจุบันโปรแกรม AppServ ได้หยุดพัฒนานานมากแล้ว (เวอร์ชันสุดท้ายเมื่อปี พ.ศ. 2551)

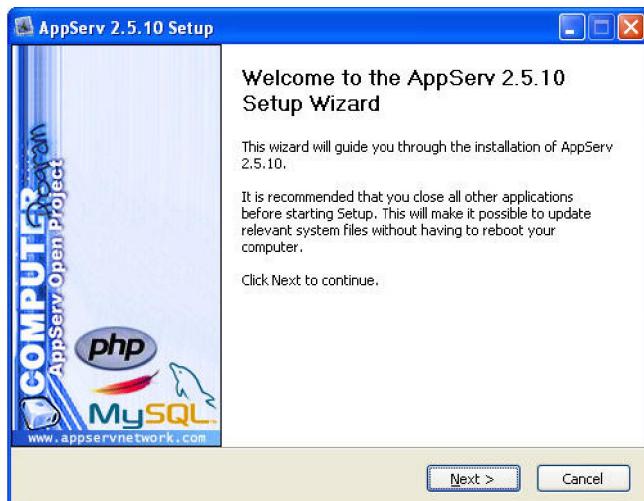


ภาพที่ ก-1 ภาณุพงศ์ ปัญญาดี ผู้พัฒนา AppServ

ที่มา: มารู้จักคุณภาณุพงศ์ ปัญญาดี ผู้พัฒนา AppServ. (2552).

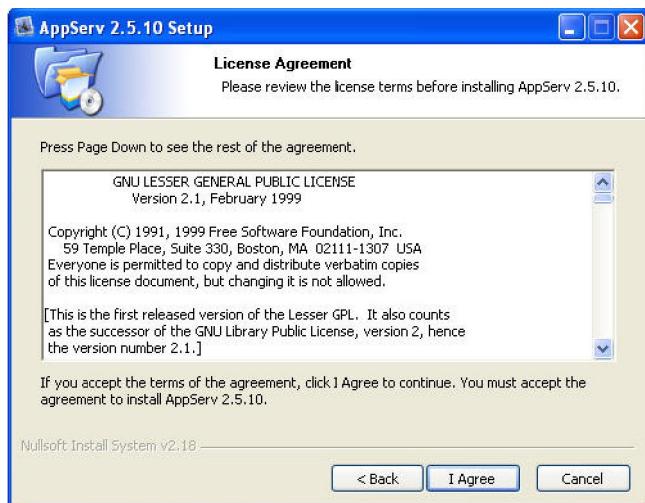
สามารถดาวน์โหลดโปรแกรม AppServ จากเว็บไซต์ <http://www.appservnetwork.com> หรือเว็บไซต์ <http://www.freebsd.sru.ac.th/web> เลือกเวอร์ชันที่ต้องการติดตั้ง (แนะนำเวอร์ชัน 2.5.10) มีขั้นตอนการติดตั้งโปรแกรม AppServ ดังนี้

1) ดับเบิลคลิกไฟล์ appserv-win32-x.x.x.exe เพื่อเริ่มติดตั้งโปรแกรม จะปรากฏหน้าจอดังภาพที่ ก-1



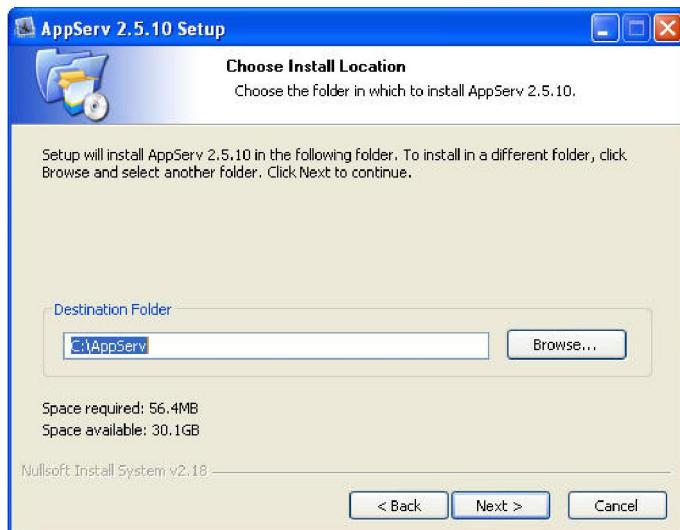
ภาพที่ ก-2 เริ่มการติดตั้งโปรแกรม AppServ

2) เข้าสู่ขั้นตอนเงื่อนไขการใช้งานโปรแกรม โดยโปรแกรม AppServ ได้แจกว่าในรูปแบบ GNU License หากผู้ติดตั้งอ่านเงื่อนไขต่างๆ เสร็จสิ้นแล้ว หากยอมรับเงื่อนไขให้คลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป แต่หากไม่ยอมรับเงื่อนไข ให้คลิก Cancel เพื่อออกจาก การติดตั้งโปรแกรม AppServ ดังภาพที่ ก-3



ภาพที่ ก-3 แสดงรายละเอียดเงื่อนไขของ GNU License

3) เลือกตำแหน่งปลายทางที่เก็บไฟล์ติดตั้งโปรแกรม AppServ โดยค่าเริ่มต้นจะถูกกำหนดไว้ที่ C:\AppServ หากต้องการเปลี่ยนตำแหน่งปลายทาง คลิกปุ่ม Browse เพื่อเลือกตำแหน่งใหม่ เสร็จแล้วคลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนถัดไป



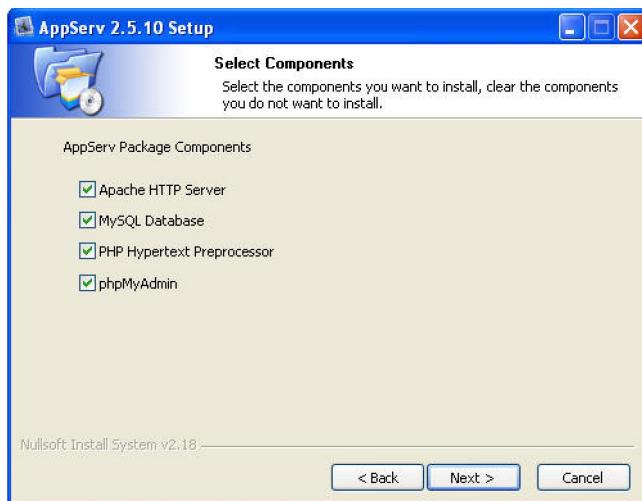
ภาพที่ ก-4 เลือกตำแหน่งปลายทางที่เก็บไฟล์ติดตั้งโปรแกรม AppServ

4) เลือกชุดติดตั้งที่ต้องการใช้งาน โดยค่าเริ่มต้นนั้นจะให้เลือกทุกรายการ แต่หากผู้ใช้งานต้องการเลือกลงเฉพาะบางรายการก็สามารถเลือกเฉพาะชุดตามต้องการ แต่ละรายการมีความหมาย ดังนี้

- 4.1) Apache HTTP Server คือ โปรแกรมเว็บเซิร์ฟเวอร์
- 4.2) MySQL Database คือ โปรแกรม MySQL
- 4.3) PHP Hypertext Preprocessor คือ โปรแกรมภาษา PHP
- 4.4) phpMyAdmin คือ โปรแกรมบริหารจัดการฐานข้อมูล MySQL บนบราวเซอร์

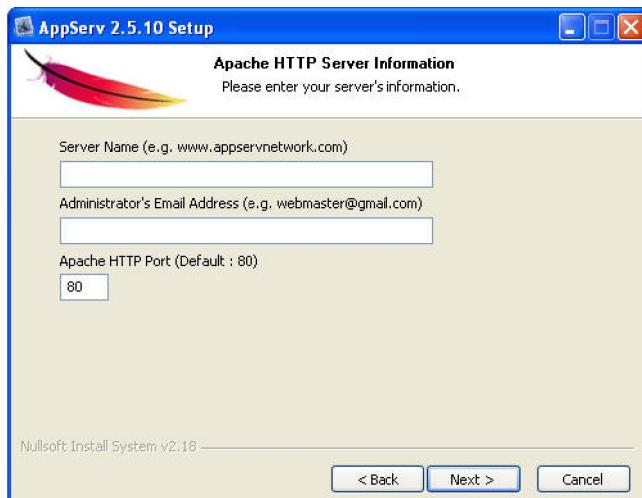
เมื่อเลือกชุดติดตั้งเรียบร้อยแล้ว ให้คลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป





ภาพที่ ก-5 เลือกชุดติดตั้งที่ต้องการใช้งาน

5) กำหนดค่าคอนฟิกสำหรับ Apache ดังภาพที่ ก-6



ภาพที่ ก-6 แสดงการกำหนดค่าคอนฟิกสำหรับ Apache

จากภาพที่ ก-6 แสดงการกำหนดค่าคอนฟิกสำหรับ Apache มีรายละเอียด ดังนี้

5.1) Server Name คือ ชื่อเว็บเซิร์ฟเวอร์

5.2) Administrator's Email Address คือ อีเมล์ของผู้ดูแลระบบ

5.3) Apache HTTP Port คือ กำหนดพอร์ตที่จะเรียกใช้งานเว็บเซิร์ฟเวอร์ โดยปกติจะใช้พอร์ต 80

6) กำหนดค่าคอนฟิกสำหรับ MySQL Database ดังภาพที่ ก-7 มีรายละเอียด ดังนี้

6.1) Enter root password คือ กำหนดรหัสผ่านการเข้าใช้งานฐานข้อมูลของ root หรือผู้ดูแลระบบฐานข้อมูล

6.2) Re-enter root password คือ ป้อนรหัสผ่านที่ได้ป้อนไว้ใน Enter root password ซ้ำอีกครั้ง (เพื่อยืนยันการกำหนดรหัสผ่านสำหรับผู้ดูแลระบบฐานข้อมูล)

6.3) Character Sets and Collations ใช้สำหรับกำหนดค่าภาษาที่ใช้ในการจัดเก็บฐานข้อมูล

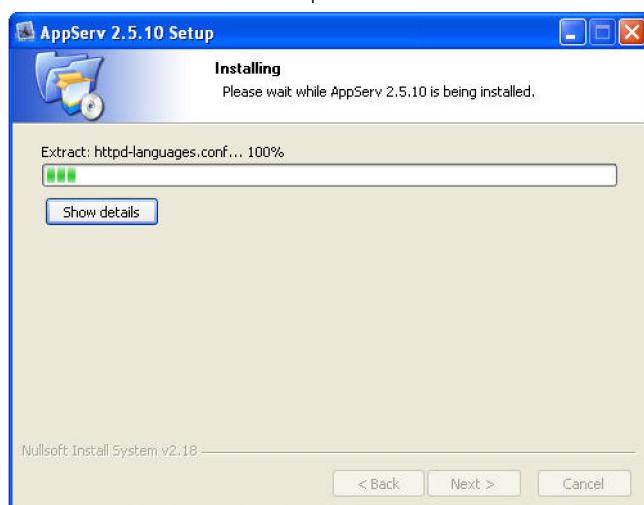
6.4) Old Password Support ใช้สำหรับกำหนดให้ MySQL รองรับมาตรฐานการเข้ารหัสผ่านแบบเก่า

6.5) Enable InnoDB หากต้องการใช้งานฐานข้อมูลในรูปแบบ InnoDB (ใช้กับฐานข้อมูลระดับกลาง โดยปกติจะใช้ในรูปแบบ MyISAM ต่างกันที่ความเร็วในการสอบถามข้อมูลและเรียกใช้งาน หากมีจำนวนข้อมูลไม่มากจะทำงานเหมือนกัน)



ภาพที่ ก-7 แสดงการกำหนดค่าคอนฟิกสำหรับฐานข้อมูล MySQL

7) หลังกำหนดค่าต่างๆ เรียบร้อยแล้ว ระบบจะดำเนินการติดตั้งชุดโปรแกรมและกำหนดค่าคอนฟิกตามที่ระบุ



ภาพที่ ก-8 แสดงการติดตั้งโปรแกรม AppServ และชุดโปรแกรมต่างๆ ตามที่ระบุ



8) สิ้นสุดขั้นตอนการติดตั้งโปรแกรม AppServ สำหรับขั้นตอนสุดท้ายนี้จะมีตัวเลือกสอบถาม ต้องการสั่งให้โปรแกรม Apache และ MySQL ทำงานหลังจากนี้เลยหรือไม่ หากต้องการให้คลิกเลือก หลังจากนั้นคลิกปุ่ม Finish เพื่อเสร็จสิ้นการติดตั้งโปรแกรม



ภาพที่ ก-9 แสดงขั้นตอนสุดท้ายของการติดตั้งโปรแกรมและตัวเลือกการทำงาน

9) โหลดโปรแกรม Apache และ MySQL เข้าสู่หน่วยความจำ



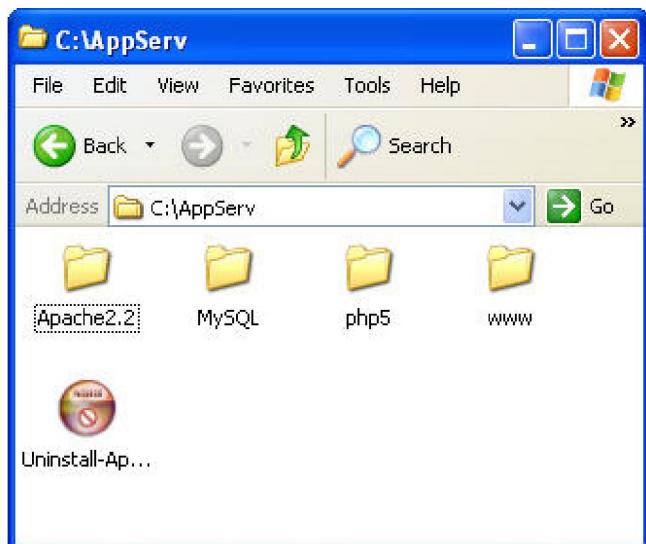
ภาพที่ ก-10 แสดงการโหลดโปรแกรม Apache และ MySQL เข้าสู่หน่วยความจำ

10) ทดสอบผลการติดตั้ง AppServ บนเว็บбраузอร์ โดยระบุ ยูอาร์แอล (URL) เป็น <http://localhost> หรือ <http://127.0.0.1> หากโปรแกรม Apache ทำงานได้เป็นปกติจะปรากฏข้อความดังภาพที่ ก-11



ภาพที่ ก-11 แสดงผลการทำงานของ AppServ บนหน้าเว็บбраузอร์

11) องค์ประกอบต่างๆ ของ AppServ จะถูกติดตั้งไว้ที่โฟลเดอร์ C:\AppServ



ภาพที่ ก-12 แสดงไฟล์ และโฟลเดอร์ต่างๆ ของโปรแกรม AppServ

จากภาพที่ ก-12 แสดงไฟล์ และโฟลเดอร์ต่างๆ ของโปรแกรม AppServ มีรายละเอียดเพิ่มเติม ดังนี้

- 11.1) Apache2.2 ใช้สำหรับเก็บไฟล์ระบบของ Apache HTTP Server
- 11.2) MySQL ใช้สำหรับเก็บไฟล์ระบบของโปรแกรม MySQL และฐานข้อมูล

ทั้งหมด

- 11.3) php5 ใช้สำหรับเก็บโปรแกรมระบบของภาษา PHP
- 11.4) www ใช้สำหรับเก็บเอกสารเว็บทั้งหมด หรืออาจเรียกว่า Document Root ใช้สำหรับอ้างอิงกับตำแหน่งการเรียกดูบนบราวเซอร์ http://localhost ถ้าภายในโฟลเดอร์ www มีโฟลเดอร์ย่อย test อุปกรณ์ภายใน การเรียกดู คือ http://localhost/test/ แล้วตามด้วยเอกสารที่อยู่ภายใน เป็นต้น ดังนั้น กระบวนการพัฒนาเว็บแอฟพลิเคชันที่จะเกิดขึ้นหลังจากนี้ ผู้พัฒนาจะต้องนำเอกสารเว็บ เพจทั้งหมดมาไว้ในพื้นที่นี้

1.1.2 XAMPP

โปรแกรม XAMPP พัฒนาโดยโครงการ Apache Friends ที่เป็นโครงการไม่แสวงหาผลกำไร ที่จัดตั้งในปี ค.ศ. 2002 โดย Kai ‘Oswald’ Seidler (Oswald, 2011) โปรแกรม XAMPP ประกอบด้วยโปรแกรมย่อย ได้แก่ Apache, MySQL, PHP และ Perl โปรแกรม XAMPP สามารถดาวน์โหลดได้ที่ <http://xampp.en.softonic.com/download>

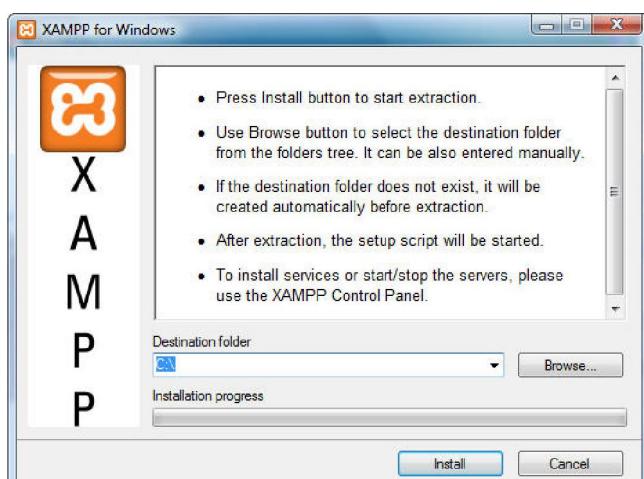


ภาพที่ ก-13 Kai 'Oswald' Seidler ผู้พัฒนา XAMPP

ที่มา: Oswald. (2011).

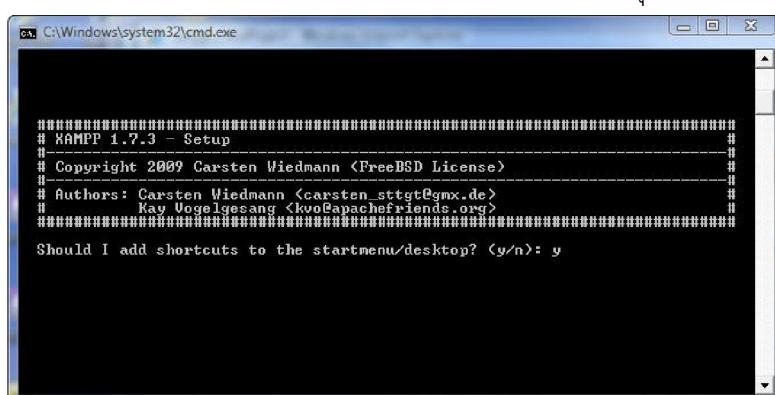
วิธีการติดตั้ง XAMPP บนระบบปฏิบัติ Windows มีขั้นตอนการติดตั้งดังนี้

- 1) ดับเบิลคลิกไฟล์ xampp-win32-1.x.x.exe เพื่อเริ่มติดตั้งโปรแกรม จะปรากฏหน้าจอตามภาพที่ ก-14



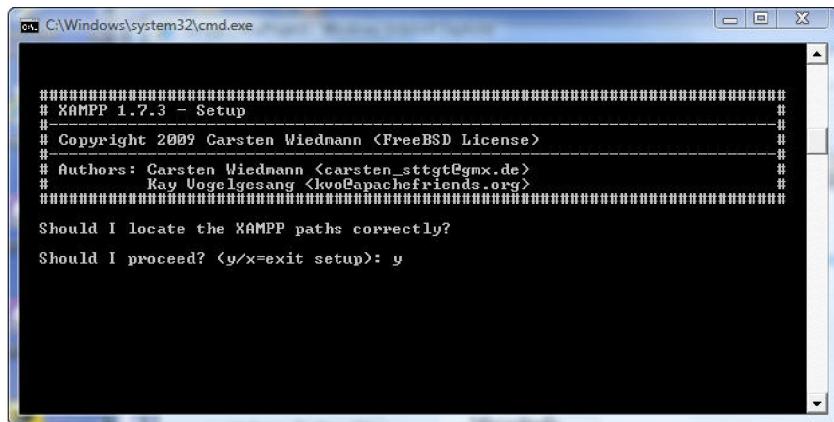
ภาพที่ ก-14 เริ่มการติดตั้ง XAMPP

- 2) จากภาพที่ ก-14 เข้าสู่ขั้นตอนการติดตั้ง XAMPP และเลือกตำแหน่งปลายทางที่เก็บไฟล์ติดตั้งโปรแกรม หากต้องการเปลี่ยนตำแหน่งปลายทาง คลิกปุ่ม Browse เพื่อเลือกตำแหน่งใหม่ เมื่อปรับเปลี่ยนตำแหน่งปลายทางเรียบร้อยแล้ว คลิกปุ่ม Install เพื่อเริ่มติดตั้งโปรแกรมและเข้าสู่การติดตั้งในลำดับขั้นตอนต่อไป แต่หากไม่ต้องการติดตั้งตั้ง ให้คลิกปุ่ม Cancel เพื่อออกจากกระบวนการติดตั้งโปรแกรม



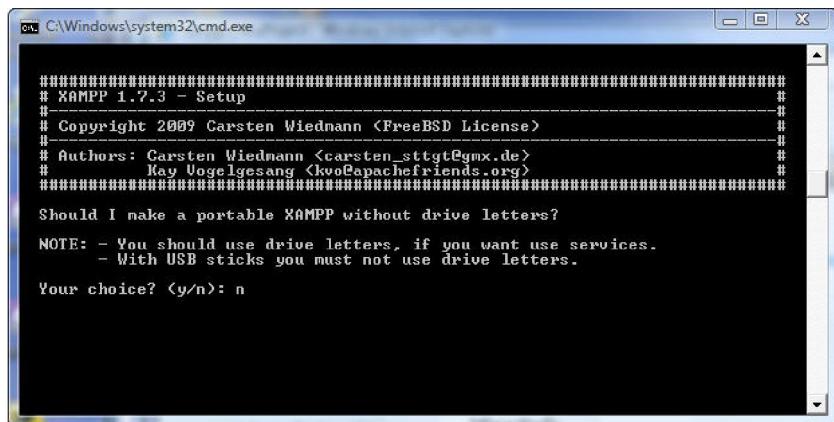
ภาพที่ ก-15 ขั้นตอนการติดตั้ง XAMPP สอบถามต้องการสร้าง Shortcuts หรือไม่

3) จากภาพที่ ก-15 แสดงขั้นตอนการติดตั้ง XAMPP สอบความต้องการสร้าง shortcuts ใน Start Menu และ Desktop หรือไม่ ในลักษณะของคำ答 yes หรือ no ถ้าต้องการให้พิมพ์ y หากไม่ต้องการให้พิมพ์ n เมื่อใส่คำตอบแล้วก็กดปุ่ม Enter เพื่อเข้าสู่การติดตั้งในลำดับขั้นตอนต่อไป



ภาพที่ ก-16 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการทวนถามความถูกต้องของตำแหน่งติดตั้ง

4) จากภาพที่ ก-16 เป็นส่วนของการทวนถามความถูกต้องของตำแหน่งปลายทางที่จะใช้ติดตั้ง XAMPP ถูกต้องและต้องการทำงานต่อหรือไม่ ในลักษณะของคำ答 yes หรือ exit ถ้าถูกต้องให้พิมพ์ y หากต้องการออกจากกระบวนการติดตั้งให้พิมพ์ x เมื่อใส่คำตอบแล้วก็กดปุ่ม Enter



ภาพที่ ก-17 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการติดตั้ง portable แบบ without drive letters

5) จากภาพที่ ก-17 ตามว่าจะสร้างที่อยู่สำหรับ XAMPP แบบพกพาโดยไม่ต้องสร้างไดร์ฟหรือไม่สำหรับขั้นตอนนี้ขอแนะนำให้กด y แล้วกด Enter เพื่อวันข้างหน้าจะได้มียุ่งยากเวลาจ้ายไฟล์ไปเครื่องอื่นด้วยการคัดลอกใส่ USB drive หรืออุปกรณ์เก็บข้อมูลอื่นๆ



```

cmd C:\Windows\system32\cmd.exe
Should I make a portable XAMPP without drive letters?
NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? (y/n): y

relocating XAMPP...
relocate XAMPP base package
relocate Apache
relocate FileZilla FTP Server
relocate Mercury
relocate MySQL
relocate OpenSSL
relocate Perl
relocate PHP
relocate phpMyAdmin
relocate Sendmail
relocate Webalizer
relocate XAMPP Demopage
relocating XAMPP successful.

XAMPP is ready to use.

Press <Return> to continue:
  
```

ภาพที่ ก-18 ขั้นตอนการติดตั้ง XAMPP แสดงผลการดำเนินการติดตั้ง XAMPP แบบพกพา

6) จากภาพที่ ก-18 แสดงผลการดำเนินการติดตั้ง XAMPP แบบพกพา และแจ้งให้ทราบว่า XAMPP พร้อมใช้งานแล้ว ให้กดปุ่ม Enter เพื่อเข้าสู่ขั้นตอนลำดับถัดไป

```

cmd C:\Windows\system32\cmd.exe
#####
## XAMPP 1.7.3 - Setup
## Copyright 2009 Carsten Wiedmann <FreeBSD License>
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Ugelgesang <kvo@apachefriends.org>
#####

I have set the timezone in 'php.ini' and 'my.ini' to "Asia/Krasnoyarsk".
You should correct these values if my guess was wrong.

Press <Return> to continue:
  
```

ภาพที่ ก-19 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการปรับโซนเวลาในไฟล์ php.ini และ my.ini

7) จากภาพที่ ก-19 ระบบแจ้งรายละเอียดการปรับเปลี่ยนโซนเวลาในไฟล์ในไฟล์ php.ini และ my.ini ซึ่งสามารถปรับเปลี่ยนในภายหลังได้ ให้กดปุ่ม Enter เพื่อเข้าสู่ขั้นตอนลำดับถัดไป

```

cmd C:\Windows\system32\cmd.exe
#####
## XAMPP 1.7.3 - Setup
## Copyright 2009 Carsten Wiedmann <FreeBSD License>
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Ugelgesang <kvo@apachefriends.org>
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: \xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 <auto>)
6. disable mod_perl
7. disable Apache::ASP
x Exit

Please choose <1-7/x>
  
```

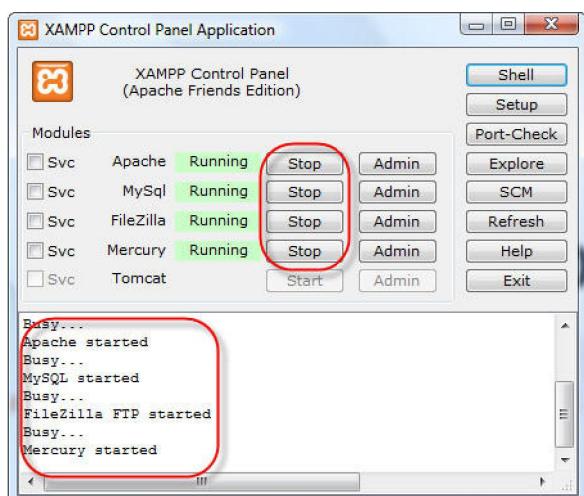
ภาพที่ ก-20 ขั้นตอนการติดตั้ง XAMPP ในส่วนของการปรับเปลี่ยนการใช้งานเพิ่มเติม

8) จากราฟที่ ก-20 ใช้สำหรับการปรับเปลี่ยนคุณลักษณะอื่นๆ เพิ่มเติม ขอแนะนำให้พิมพ์ x เพื่อออกจากการติดตั้งและเริ่มใช้งานในลำดับ ถัดไป หรือพิมพ์ 1 เพื่อเปิด XAMPP Control Panel เพื่อควบคุมการทำงานของโปรแกรมต่างๆ ที่เกี่ยวข้อง แล้วกดปุ่ม Enter เพื่อเริ่มต้นใช้งาน XAMPP

9) เปิด XAMPP Control Panel แล้วตรวจสอบโปรแกรมหลักๆ ที่จะใช้งานว่า Started หรือไม่ หลักๆ แล้วให้ดูที่ Apache และ MySQL จะต้องมีสถานะเป็น Started ดังภาพที่ ก-21



ภาพที่ ก-21 แสดงขั้นตอนการเปิด XAMPP Control Panel



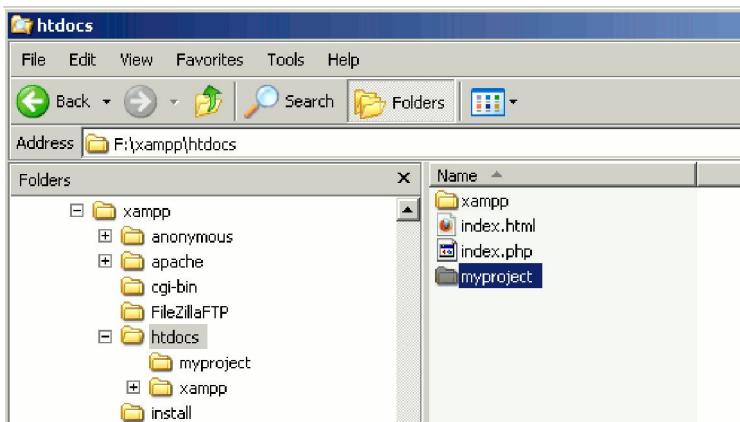
ภาพที่ ก-22 แสดงหน้าจอ XAMPP Control Panel เพื่อควบคุมโปรแกรมต่างๆ ที่เกี่ยวข้อง

10) เปิดโปรแกรมเว็บбраузर และพิมพ์คำว่า <http://localhost> หรือ <http://127.0.0.1> ลงไปในช่องรับยูอาร์แอล หากระบบสามารถทำงานได้เป็นปกติจะปรากฏข้อความ ดังภาพที่ ก-23



ภาพที่ ก-23 แสดงผลการทำงานของ XAMPP บนหน้าเว็บбраузอร์

11) ที่เก็บเอกสารสำหรับเริ่มต้นพัฒนาเว็บไซต์ จะอยู่ในโฟลเดอร์ htdocs ในตำแหน่งพารที่ได้ดำเนินการติดตั้ง XAMPP และควรสร้างโฟลเดอร์ใหม่ให้ป็นกับเอกสารไดๆ เพื่อความสะดวกและป้องกันข้อผิดพลาดเนื่องจากตั้งชื่อซ้ำกับเอกสารหลัก ตัวอย่างเช่น สร้างโฟลเดอร์ชื่อ project ไว้ใน c:\xampp\htdocs ดังในภาพ ก-24



ภาพที่ ก-24 แสดงการสร้างโฟลเดอร์เก็บเอกสารเว็บเพจของ XAMPP

1.2 เครื่องมือสำหรับการเขียนสคริปต์

การเริ่มต้นเขียนสคริปต์ภาษา PHP มีเครื่องมือสำหรับการสร้างและแก้ไขสคริปต์หลายโปรแกรม เช่น Notepad++, Editplus, vi, และ Adobe Dreamweaver CS3 เป็นต้น มีหลายเหตุผลในการเลือกเพื่อให้การพัฒนาเป็นไปอย่างราบรื่น แต่ละโปรแกรมจะมีจุดด้อยจุดเด่นแตกต่างกันไป ตัวอย่างเครื่องมือที่ใช้สำหรับการเขียนสคริปต์ มีดังนี้

1.2.1 Notepad++

Notepad++ คือ โปรแกรมแก้ไขเอกสารข้อความ รองรับการเขียนภาษาต่างๆ ของโปรแกรมภาษาหลายชนิด เป็นโปรแกรมที่โหลดให้ใช้ฟรี ภายใต้สัญญา GPL สามารถดาวน์โหลดได้ที่ <http://notepad-plus.sourceforge.net/>

คุณสมบัติหลัก

- มีตัวเน้นประโยคคำสั่ง (syntax highlight) ช่วยให้การเขียนสคริปต์โปรแกรมได้สะดวก รองรับภาษาคอมพิวเตอร์หลายภาษา เช่น C, C++, Java, C#, XML, HTML, PHP, CSS, makefile ASCII art (.nfo), doxygen, ini file, batch file, Javascript, ASP, VB/VBS, SQL, Objective-C, RC resource file, Pascal, Perl, Python, Lua, TeX, TCL, Assembler และ Ruby เป็นต้น

2) มีระบบแสดงผลในตัวในลักษณะ WYSIWYG (What You See Is What You Get)

3) สามารถกำหนดและปรับแต่งตัวเน้นประโยคคำสั่งได้เอง และมีระบบเติมคำอัตโนมัติ

4) สามารถเปิดเอกสารได้หลายหน้าในเวลาเดียวกัน

5) เปลี่ยนมุมมองในการใช้โปรแกรมได้หลายมุมมอง (Multi View)

6) ระบบกรองคำ เพื่อการค้นหาและแทนที่คำ

- 7) รองรับการใช้แม่ส์ลากวาง
- 8) ขยายและย่อตัวอักษรได้ด้วยมาส์
- 9) รองรับหลายภาษา รวมทั้งภาษาไทยด้วย
- 10) ทำ Bookmark ตำแหน่งบรรทัดที่แก้ไขบ่อยๆ ได้
- 11) มีเส้นแสดงตำแหน่งของวงเล็บเปิด/ปิด ทำให้การใส่ скриปต์ทำได้ง่ายและสะดวก
- 12) บันทึกการทำงานแบบมาโคร ช่วยให้การทำงานแบบซ้ำๆ ทำได้เร็วขึ้น
- 13) มีชุดรูปแบบ (Theme) ที่หลากหลายให้เลือกใช้ตามความต้องการ

1.2.2 EditPlus

Editplus คือ โปรแกรมแก้ไขข้อความชนิดหนึ่ง มีคุณสมบัติเป็นทั้งโปรแกรมสำเร็จรูปสำหรับการสร้างและพัฒนาเว็บเพจ สามารถประยุกต์ใช้สำหรับการเขียนสคริปต์โปรแกรมต่างๆ ไม่ว่าจะเป็น เขียนและแก้ไขสคริปต์ ด้วยภาษา HTML, PHP และ Java เป็นต้น เป็นโปรแกรมที่ทำงานบนระบบปฏิบัติ MS-Windows เท่านั้น

คุณสมบัติหลัก

- 1) สามารถใช้ภาษาไทยได้
- 2) แยกคำสั่งต่างๆ ด้วยการแสดงสีที่ไม่เหมือนกัน ทำให้สามารถตรวจสอบได้ง่ายเมื่อพิมพ์คำสั่งผิด
- 3) สามารถใช้เป็นเครื่องตรวจเชอร์ฟิล์ด
- 5) สามารถเปิดไฟล์ได้ครั้งละหลายๆ ไฟล์พร้อมกัน
- 6) สามารถค้นหาและแทนที่ (Find & Replace) ข้อความเดียวกันได้ครั้งละหลายๆ ไฟล์พร้อมกัน
- 7) สามารถค้นหาข้อความที่ต้องการ ว่าปรากฏอยู่ในไฟล์ไหนบ้าง (พร้อมแสดงหมายเลขบรรทัด) ในไดเรกทอรีเดียวกัน

1.2.3 Adobe Dreamweaver

Adobe Dreamweaver ถือได้ว่าเป็นเครื่องมือสำหรับสร้างเว็บเพจ และดูแลเว็บไซต์ ที่มีประสิทธิภาพสูง เป็นที่นิยมใช้ของผู้พัฒนาเว็บอย่างกว้างขวาง สามารถใช้เขียนภาษาสำหรับการพัฒนาเว็บเพจที่หลากหลาย เช่น HTML, PHP, ASP และ JSP เป็นต้น ช่วยประหยัดเวลา และทำงานได้สะดวกยิ่งขึ้น

ความสามารถของ Adobe Dreamweaver จะมีลักษณะคล้ายกับการพิมพ์งานในโปรแกรมแก้ไขข้อความ ทั่วไป สรุปได้ดังนี้

- 1) มีระบบแสดงผลในตัวในลักษณะ WYSIWYG หมายความว่า เว็บที่เขียนในหน้าของ Dreamweaver เขียนไว้อย่างไร ก็จะแสดงแบบเดียวกับในเว็บเพจจริงๆ ในเว็บбраузอร์ ช่วยให้เขียนเว็บเพจได้ง่าย โดยไม่ต้องเขียนสคริปต์ HTML เองทั้งหมด



- 2) มีเครื่องมือในการช่วยสร้างเว็บเพจ ที่มีความยืดหยุ่นสูง
- 3) สนับสนุนภาษาสคริปต์ต่างๆ ทั้งฝั่ง โคลอเน็นท์และเซิร์ฟเวอร์ เช่น Java, ASP, PHP, CGI และ VBScript เป็นต้น
- 4) มีเครื่องมือในการอัปโหลด หน้าเว็บเพจไปยังเซิร์ฟเวอร์เพื่อเผยแพร่วेबเพจที่สร้างในอินเทอร์เน็ต โดยส่งผ่าน FTP
- 5) รองรับมัลติมีเดีย เช่น การแทรกไฟล์เสียง ไฟล์วิดีโอ แฟลช (Flash) เป็นต้น

1.3 เว็บбраузอร์

เว็บбраузอร์ คือ โปรแกรมคอมพิวเตอร์ ที่ใช้สำหรับเรียกดูหรือค้นดูข้อมูล และโต้ตอบกับข้อมูลสารสนเทศที่จัดเก็บในเว็บเพจที่สร้างและพัฒนาด้วยภาษาเฉพาะ เช่น ภาษา HTML ที่จัดเก็บไว้บนระบบบริการเว็บหรือเว็บเซิร์ฟเวอร์ หรือระบบคลังข้อมูลอื่นๆ

ประโยชน์ของbrauzer คือ สามารถเรียกดูเอกสารภายในเว็บเซิร์ฟเวอร์ได้ สามารถแสดงผลข้อมูลในรูปของข้อความ ภาพ และระบบมัลติมีเดียต่างๆ ทำให้การเรียกดูเว็บเพจมีความน่าสนใจมากขึ้น ส่งผลให้อินเตอร์เน็ตได้รับความนิยมเป็นอย่างมากเช่นในปัจจุบัน ตัวอย่างโปรแกรมเว็บเบราว์เซอร์ที่นิยมใช้งาน มีดังนี้

- 1.3.1 Internet Explorer
- 1.3.2 Mozilla Firefox
- 1.3.3 Google Chrome
- 1.3.4 Safari

1.4 โปรแกรมโอนย้ายไฟล์

โปรแกรมโอนย้ายไฟล์ เป็นโปรแกรมที่ใช้ร่วมกับโปรโตคอล FTP (File Transfer Protocol) เพื่อใช้สำหรับโอนย้ายไฟล์ระหว่างเครื่องคอมพิวเตอร์โคลอเนนต์และเซิร์ฟเวอร์ ระหว่างการสร้างและพัฒนาเว็บแอปพลิเคชันนั้น จะใช้เครื่องคอมพิวเตอร์ส่วนบุคคล担当เป็นเว็บเซิร์ฟเวอร์ เพื่อทดสอบการประมวลผลต่างๆ หลังจากการพัฒนาและทดสอบเสร็จสิ้น ขั้นตอนสุดท้ายก็คือ การนำขึ้นบนระบบเว็บเซิร์ฟเวอร์จริง โดยจำนวนต้องอาศัยโปรแกรมโอนย้ายไฟล์เป็นตัวช่วยในการนำไฟล์ออกสารรูปภาพ วิดีโอ สื่อมัลติมีเดีย หรืออื่นๆ ขึ้นสู่เว็บเซิร์ฟเวอร์เพื่อเผยแพร่ประชาสัมพันธ์หรืออื่นๆ ตามวัตถุประสงค์ โปรแกรมที่ได้รับความนิยมใช้ในปัจจุบันมีดังนี้

- 1.4.1 FileZilla Client
- 1.4.2 WS FTP Pro
- 1.4.3 CuteFTP

สรุป

เครื่องมือที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันนั้นมีหลากหลายโปรแกรมให้เลือกใช้ ผู้พัฒนา จะต้องศึกษาหาโปรแกรมที่เหมาะสมกับความถนัดของตนเอง ส่วนสำคัญที่จะประสบความสำเร็จในการ พัฒนาเว็บแอปพลิเคชัน ก็คือ ผู้พัฒนาจะต้องศึกษา วางแผน วิเคราะห์ ออกแบบ พัฒนา และทดสอบ หลังจากที่ได้ต้นแบบเรียบร้อยแล้ว ก็เริ่มขั้นตอนการพัฒนาเว็บแอปพลิเคชัน 3 ขั้นตอนสำคัญ

- 1) ออกแบบเว็บเพจในลักษณะของแม่แบบ เพื่อให้ทุกเว็บเพจภายในเว็บไซต์เป็นรูปแบบเดียวกันทั้งหมด
- 2) สร้างเว็บเพจตามหน้าที่การทำงานในแต่ละส่วนที่ได้ออกแบบไว้และทดสอบการทำงานในเบื้องต้น
- 3) เมื่อพัฒนาเสร็จสมบูรณ์ก็จะโอนย้ายไฟล์ไปยังเว็บเซิร์ฟเวอร์สาธารณะเพื่อเผยแพร่ และเรียกใช้งาน จาก 3 ขั้นตอนสำคัญที่กล่าวมาแล้วนั้น ในแต่ละส่วนผู้พัฒนาจะต้องเลือกใช้เครื่องมือให้เหมาะสมและ ณ นั้น เพื่อพัฒนาระบบที่มีประสิทธิภาพต่อไป





ภาคผนวก ข

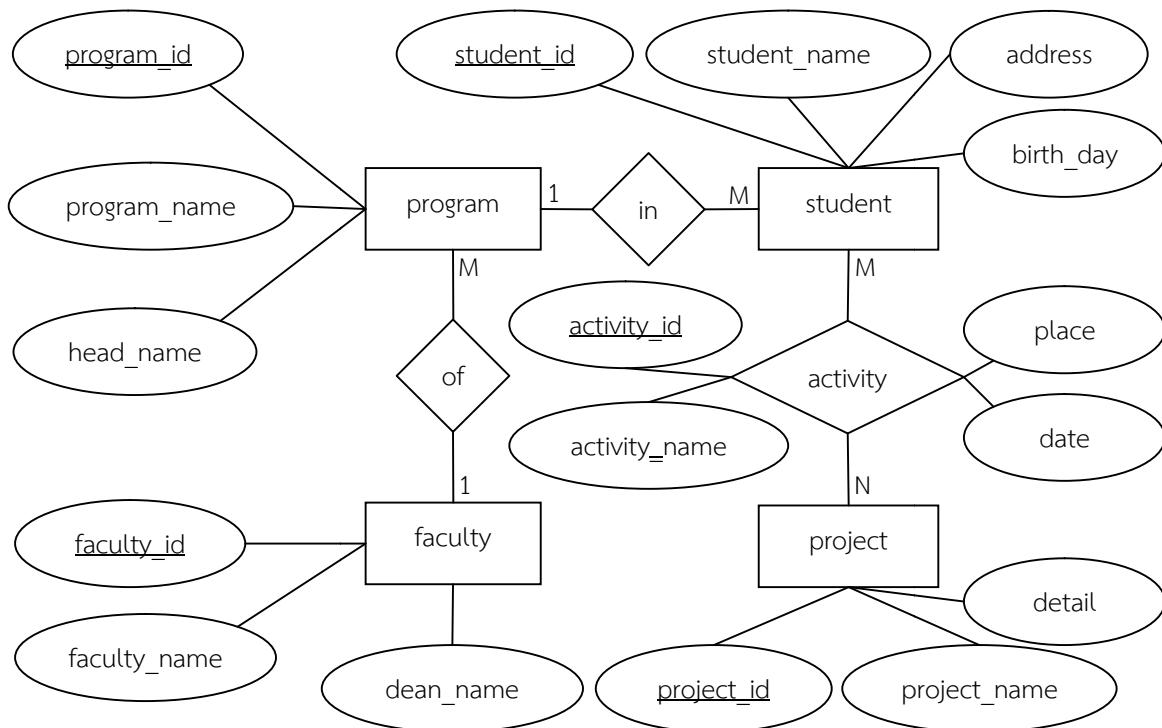
กรณีศึกษา: ระบบบันทึกการทำกิจกรรมของนักศึกษา



ภาคผนวก ข

กรณีศึกษา: ระบบบันทึกการทำกิจกรรมของนักศึกษา

กรณีศึกษา: ระบบบันทึกการทำกิจกรรมของนักศึกษา เป็นระบบที่ไม่ยุ่งยากและไม่ซ้ำซ้อนมากนัก เพื่อเป็นกรณีศึกษาและง่ายสำหรับผู้เริ่มต้นพัฒนาระบบ ในที่นี้จะกล่าวถึง วิธีการ เพิ่ม แก้ไข ลบ และรายงานข้อมูลที่เกี่ยวข้อง มีรายละเอียดดังนี้



ภาพที่ ข-1 แสดงความสัมพันธ์ของข้อมูล ในรูปแบบ ER-Diagram ระบบบันทึกการทำกิจกรรมของนักศึกษา

1.1 กำหนดแອททริบิวต์และความสัมพันธ์

แອททริบิวต์และความสัมพันธ์ อธิบายได้ดังนี้

student (student_id, student_name, address, birth_day, program_id)

program (program_id, program_name, head_name, faculty_id)

faculty (faculty_id, faculty_name, dean_name)

project (project_id, project_name, detail)

activity (activity_id, activity_name, date, place, student_id, project_id)

ภาพที่ ข-2 แสดงความสัมพันธ์ของข้อมูลระบบบันทึกการทำกิจกรรมของนักศึกษา

จากรูปภาพที่ ข-2 แสดงความสัมพันธ์ของข้อมูล ระบบบันทึกการทำกิจกรรมของนักศึกษา สามารถสร้างตารางความสัมพันธ์ได้ 5 ตาราง ประกอบด้วย 1) ตารางข้อมูลนักศึกษา 2) ตารางข้อมูลสาขาวิชา 3) ตารางข้อมูลคณะ 4) ตารางข้อมูลกิจกรรม และ 5) ตารางข้อมูลการเข้าร่วมกิจกรรม และสามารถนำมาเขียนเป็นพจนานุกรมข้อมูล มีรายละเอียดดังนี้

ตารางที่ ข-1 ตารางข้อมูลนักศึกษา (student)

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>student_id</u>	VARCHAR(5)	Not	-	PK	รหัสนักศึกษา
<u>student_name</u>	VARCHAR(40)	Not	-	-	ชื่อ-สกุล
<u>birth_day</u>	DATE	-	-	-	วันเดือนปีเกิด
<u>address</u>	VARCHAR(40)	-	-	-	ที่อยู่
<u>program_id</u>	VARCHAR(3)	Not	-	FK	รหัสสาขาวิชา

ตารางที่ ข-2 ตารางข้อมูลสาขาวิชา (program)

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>program_id</u>	VARCHAR(3)	Not	-	PK	รหัสสาขาวิชา
<u>program_name</u>	VARCHAR(40)	Not	-	-	ชื่อสาขาวิชา
<u>head_name</u>	VARCHAR(40)	-	-	-	ชื่อหัวหน้าสาขาวิชา
<u>faculty_id</u>	VARCHAR(3)	Not	-	FK	รหัสคณะ

ตารางที่ ข-3 ตารางข้อมูลคณะ (faculty)

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>faculty_id</u>	VARCHAR(3)	Not	-	PK	รหัสคณะ
<u>faculty_name</u>	VARCHAR(40)	Not	-	-	ชื่อคณะ
<u>dean_name</u>	VARCHAR(40)	-	-	-	ชื่อคณบดี

ตารางที่ ข-4 ตารางข้อมูลโครงการ (project)

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
<u>project_id</u>	VARCHAR(5)	Not	-	PK	รหัสโครงการ
<u>project_name</u>	VARCHAR(40)	Not	-	-	ชื่อโครงการ
<u>detail</u>	TEXT	-	-	-	รายละเอียดโครงการ

ตารางที่ ข-5 ตารางข้อมูลการทำกิจกรรม (activity)

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่า ปริยาย	คีย์	หมายเหตุ
activity_id	INT	NOT	-	PK	รหัสกิจกรรม (AUTO_INCREMENT)
activity_name	VARCHAR(40)	NOT	-	-	ชื่อกิจกรรม
activity_date	DATE	-	-	-	วันที่ทำกิจกรรม
activity_place	VARCHAR(40)	-	-	-	สถานที่ทำกิจกรรม
student_id	VARCHAR(5)	Not	-	FK	รหัสนักศึกษา
project_id	VARCHAR(5)	Not	-	FK	รหัสโครงการ

1.2 สร้างฐานข้อมูลและตาราง

กำหนดให้สร้างฐานข้อมูลชื่อ university และสร้างตาราง 5 ตาราง มีรายละเอียดดังนี้

1.2.1 สร้างฐานข้อมูล university ใช้คำสั่งดังนี้**ตัวอย่างที่ ข-1 การสร้างฐานข้อมูล university**

```
CREATE DATABASE university;
```

1.2.2 สร้างตารางข้อมูล ดังนี้**1) ตารางข้อมูลนักศึกษา ใช้คำสั่งดังนี้****ตัวอย่างที่ ข-2 การสร้างตารางข้อมูลนักศึกษา**

```
USE university;
CREATE TABLE student (
    student_id VARCHAR(5) NOT NULL PRIMARY KEY,
    student_name VARCHAR(40) NOT NULL,
    birth_day DATE,
    address VARCHAR(40),
    program_id VARCHAR(3) NOT NULL);
```

ก่อนสร้างตารางใดๆ เป็นตารางแรก จะต้องเลือกฐานข้อมูลก่อนการใช้คำสั่งอื่นๆ คำสั่งที่ใช้สำหรับการเลือกฐานข้อมูล คือ USE และตามด้วยชื่อฐานข้อมูล

2) ตารางข้อมูลสาขาวิชา ใช้คำสั่งดังนี้**ตัวอย่างที่ ข-3 การสร้างตารางข้อมูลสาขาวิชา**

```
CREATE TABLE program (
    program_id VARCHAR(3) NOT NULL PRIMARY KEY,
    program_name VARCHAR(40) NOT NULL,
    head_name VARCHAR(40),
    faculty_id VARCHAR(3) NOT NULL);
```



3) ตารางข้อมูลคน ใช้คำสั่งดังนี้

ตัวอย่างที่ ข-4 การสร้างตารางข้อมูลคน

```
CREATE TABLE faculty (
    faculty_id VARCHAR(3) NOT NULL PRIMARY KEY,
    faculty_name VARCHAR(40) NOT NULL,
    dean_name VARCHAR(40));
```

4) ตารางข้อมูลโครงการ ใช้คำสั่งดังนี้

ตัวอย่างที่ ข-5 การสร้างตารางข้อมูลโครงการ

```
CREATE TABLE project (
    project_id VARCHAR(5) NOT NULL PRIMARY KEY,
    project_name VARCHAR(40) NOT NULL,
    detail TEXT);
```

5) ตารางข้อมูลการทำกิจกรรม ใช้คำสั่งดังนี้

ตัวอย่างที่ ข-6 การสร้างตารางข้อมูลการทำกิจกรรม

```
CREATE TABLE activity (
    activity_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    activity_name VARCHAR(40) NOT NULL,
    activity_date DATE,
    activity_place VARCHAR(40),
    student_id VARCHAR(5) NOT NULL,
    project_id VARCHAR(5) NOT NULL);
```

1.3 รายละเอียดของไฟล์ที่เกี่ยวข้อง

รายละเอียดของไฟล์ที่เกี่ยวข้องกับระบบบันทึกการทำกิจกรรมของนักศึกษา มีทั้งหมด 38 ไฟล์ มีรายละเอียดดังนี้

ตารางที่ ข-6 แสดงรายละเอียดของไฟล์ที่เกี่ยวข้องในระบบบันทึกการทำกิจกรรมของนักศึกษา

ชื่อไฟล์	คำอธิบาย
activity_delete.php	ลบข้อมูลกิจกรรม
activity_edit.php	สคริปต์คำสั่งแก้ไขข้อมูลข้อมูลการทำกิจกรรม
activity_form_edit.php	ฟอร์มสำหรับแก้ไขข้อมูลการทำกิจกรรม
activity_form_insert.php	ฟอร์มสำหรับแทรกข้อมูลการทำกิจกรรมใหม่
activity_insert.php	สคริปต์คำสั่งที่ใช้แทรกข้อมูลการทำกิจกรรม
activity_show.php	แสดงการทำกิจกรรมของนักศึกษารายบุคคล
admin_faculty.php	เว็บเพจหลักสำหรับจัดการข้อมูลคน

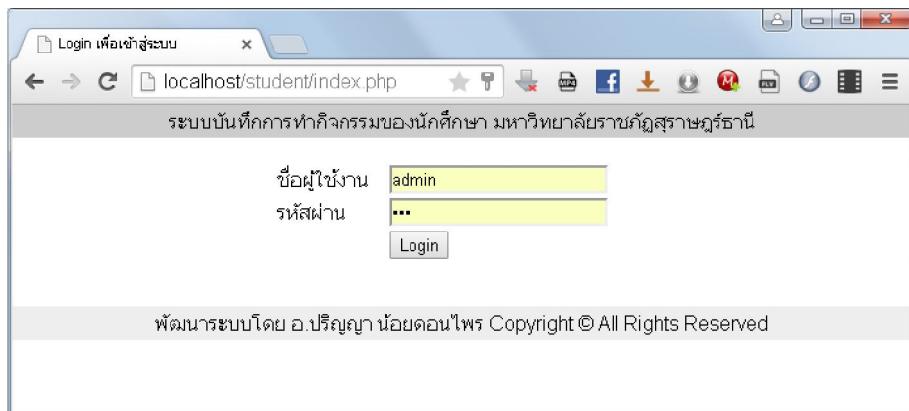
ตารางที่ ข-6 (ต่อ)

ชื่อไฟล์	คำอธิบาย
admin_header.php	ส่วนหัวของหน้าเว็บเพจ ร่วมกับ header.php
admin_menu.php	ส่วนของเมนูสำหรับเลือกจัดการข้อมูลในระบบ
admin_panel.php	เว็บเพจหลักสำหรับเชื่อมโยงหน้าจัดการระบบทั้งหมด
admin_program.php	เว็บเพจหลักจัดการข้อมูลสาขาวิชา
admin_project.php	เว็บเพจหลักจัดการข้อมูลโครงการ
admin_student.php	เว็บเพจหลักจัดการข้อมูลนักศึกษา
check_login.php	ตรวจสอบการเข้าสู่ระบบ
connect_database.php	ติดต่อฐานข้อมูล
faculty_edit.php	스크립ต์คำสั่งแก้ไขข้อมูลข้อมูลคนงาน
faculty_form_edit.php	ฟอร์มสำหรับแก้ไขข้อมูลคนงาน
faculty_form_insert.php	แก้ไขข้อมูลการทำกิจกรรม (ฟอร์มสำหรับแก้ไขข้อมูล)
faculty_insert.php	스크립ต์คำสั่งที่ใช้แทรกข้อมูลคนงาน
footer.php	ส่วนท้ายของเว็บเพจ
header.php	ส่วนหัวของเว็บเพจ
index.php	หน้าแรกของเว็บไซต์ ใช้ระบุชื่อผู้ใช้และรหัสผ่าน
logout.php	ออกจากระบบ
program_delete.php	스크립ต์คำสั่งที่ใช้ลบข้อมูลสาขาวิชา
program_edit.php	스크립ต์คำสั่งที่ใช้แก้ไขข้อมูลสาขาวิชา
program_form_edit.php	ฟอร์มสำหรับแก้ไขข้อมูลสาขาวิชา
program_form_insert.php	ฟอร์มสำหรับแทรกข้อมูลสาขาวิชาใหม่
program_insert.php	스크립ต์คำสั่งที่ใช้แทรกข้อมูลสาขาวิชาใหม่
project_delete.php	스크립ต์คำสั่งที่ใช้ลบข้อมูลโครงการ
project_edit.php	스크립ต์คำสั่งที่ใช้แก้ไขข้อมูลโครงการ
project_form_edit.php	ฟอร์มสำหรับแก้ไขข้อมูลโครงการ
project_form_insert.php	ฟอร์มสำหรับแทรกข้อมูลโครงการใหม่
project_insert.php	스크립ต์คำสั่งที่ใช้แทรกข้อมูลโครงการใหม่
student_delete.php	스크립ต์คำสั่งที่ใช้ลบข้อมูลนักศึกษา
student_edit.php	스크립ต์คำสั่งที่ใช้แก้ไขข้อมูลนักศึกษา
student_form_edit.php	ฟอร์มสำหรับแก้ไขข้อมูลนักศึกษา
student_form_insert.php	ฟอร์มสำหรับแทรกข้อมูลนักศึกษาใหม่
student_insert.php	스크립ต์คำสั่งที่ใช้แทรกข้อมูลนักศึกษาใหม่

จากตารางที่ ข-6 แสดงรายละเอียดของไฟล์ที่เกี่ยวข้องในระบบบันทึกการทำกิจกรรมของนักศึกษา ในลำดับถัดไปจะแสดงscriptภายในเว็บเพจ พร้อมภาพประกอบเพื่อให้สามารถในการศึกษา ลักษณะการทำงานจะแบ่งส่วนการทำงานตามลำดับ ดังต่อไปนี้



1.3.1 หน้าแรกของระบบ



ภาพที่ ข-3 แสดงหน้าแรกของเว็บไซต์ ใช้ระบุชื่อผู้ใช้และรหัสผ่านเพื่อเข้าสู่ระบบ

จากภาพที่ ข-3 แสดงหน้าแรกของเว็บไซต์ ชื่อไฟล์ index.php ใช้สำหรับตรวจสอบสิทธิเพื่อเข้าใช้งานระบบ หลังจากป้อนชื่อผู้ใช้และรหัสผ่าน คลิกปุ่ม Login ระบบจะส่งชื่อผู้ใช้และรหัสผ่านไปประมวลผล ที่ไฟล์ check_login.php รายละเอียดดังนี้

index.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Login เพื่อเข้าสู่ระบบ</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
-->
</style></head>
<body>
<?php
    include("header.php");
?>
<form action="check_login.php" method="post" name="login">
```

```

<br>
<table width="300" border="0" align="center">
<tr>
    <td>ชื่อผู้ใช้งาน</td>
    <td><input type="text" name="user" id="user"></td>
</tr>
<tr>
    <td>รหัสผ่าน</td>
    <td><input type="password" name="pass" id="pass"></td>
</tr>
<tr>
    <td>&ampnbsp</td>
    <td><input type="submit" name="btnLogin" id="btnLogin" value="Login"></td>
</tr>
</table>
<br>
</form>
<?php
    include("footer.php");
?>
</body>
</html>

```

ภายในเว็บเพจ index.php มีไฟล์นำเข้า มีรายละเอียดดังนี้

1) ส่วนหัวของเว็บเพจ ชื่อไฟล์ header.php

header.php

```

<table width="100%" border="0" bgcolor="#CCCCCC">
<tr>
    <td><?php echo "<center>ระบบบันทึกการทำกิจกรรมของนักศึกษา มหาวิทยาลัยราช
ก្ភสุราษฎร์ธานี</center>"; ?>
    </td>
</tr>
</table>

```

2) ส่วนท้ายของเว็บเพจ ชื่อไฟล์ footer.php

footer.php

```

<table width="100%" border="0" bgcolor="#EEEEEE">
<tr>

```



```
<td><?php echo "<center>พัฒนาระบบโดย อ.ปริญญา น้อยดอนไพร Copyright ©  
All Rights Reserved</center>"; ?></td>  
</tr>  
</table>
```

3) ตรวจสอบสิทธิเพื่อเข้าใช้งานระบบ ชื่อไฟล์ check_login.php

check_login.php

```
<?php  
    session_start();  
    $admin="admin";  
    $password="123";  
    if($_POST["user"]==$admin and $_POST["pass"]==$password) {  
        session_register("admin");  
        session_register("password");  
        header("Location: admin_panel.php");  
        exit;  
    } else {  
        header("Location: index.php");  
        session_destroy();  
        exit;  
    }  
?>
```

ถ้าชื่อผู้ใช้และรหัสผ่านถูกต้องจะย้ายหน้าเว็บเพจไปยัง admin_panel.php ถ้าไม่ตรง กับที่กำหนดไว้ ย้ายหน้าเว็บเพจไปยัง index.php เพื่อตรวจสอบสิทธิใหม่ หลังจากเว็บเพจ admin_panel.php เริ่มทำงานจะนำเข้าไฟล์ส่วนตัวและเมนู มีสคริปต์ดังนี้

4) หน้าเว็บเพจหลักสำหรับเจ้าหน้าที่ ชื่อไฟล์ admin_header.php มีสคริปต์ดังนี้

admin_header.php

```
<table width="100%" border="0" bgcolor="#CCCCCC">  
    <tr><td><center><h1>  
        <?php  
            echo "ยินดีต้อนรับคุณ "; echo $_SESSION["admin"]; echo " เข้าสู่ระบบ";  
        ?>  
        </h1></center></td></tr>  
    <tr><td><center>[ <a href="logout.php"> Logout </a> ]</center></td></tr>  
    <tr><td><?php include("header.php"); ?></td></tr>  
</table>  
<?php include("admin_menu.php"); ?>
```

5) เมนูการใช้งานสำหรับเจ้าหน้าที่ admin_menu.php มีสคริปต์ดังนี้

admin_menu.php

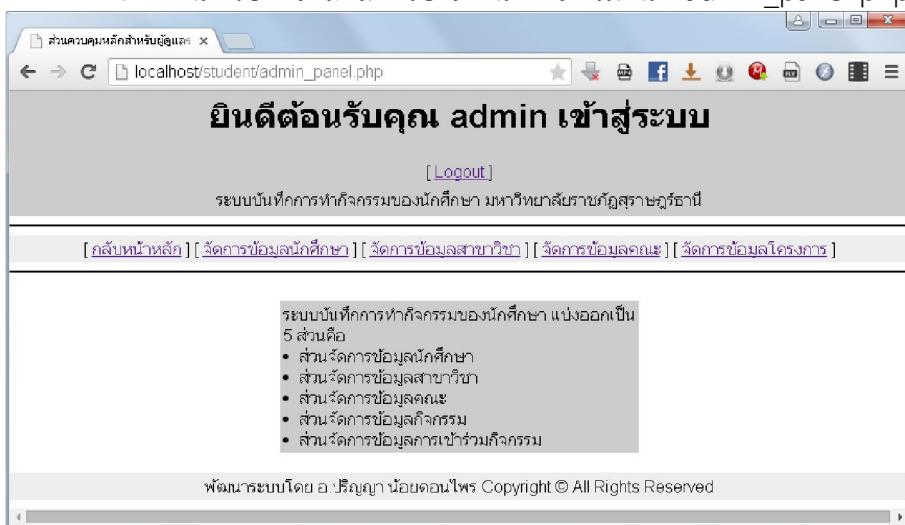
```
<hr color="black" width="100%">





```

1.3.2 หน้าเว็บเพจหลักสำหรับเจ้าหน้าที่ จะเริ่มต้นที่ admin_panel.php ตัวอย่าง ดังนี้



ภาพที่ ข-4 หน้าเว็บเพจหลักสำหรับเจ้าหน้าที่ใช้เข้ามายังหน้าจัดการระบบทั้งหมด

1) เว็บเพจหลักสำหรับเข้ามายังหน้าจัดการระบบทั้งหมด ชื่อไฟล์ admin_panel.php

admin_panel.php

```
<?php
session_start();
if (!isset ($_SESSION["admin"])) { header("Location: index.php"); exit; }
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
```



```

body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}

body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}

-->

</style></head>

<body>
<?php
    include("admin_header.php");
?>
<br> <table align="center" width="40%" border="0" bgcolor="#CCCCCC">
    <tr> <td>
        ระบบบันทึกการทำกิจกรรมของนักศึกษา แบ่งออกเป็น 5 ส่วนคือ
        <li>ส่วนจัดการข้อมูลนักศึกษา </li>
        <li>ส่วนจัดการข้อมูลสาขาวิชา </li>
        <li>ส่วนจัดการข้อมูลคณะ</li>
        <li>ส่วนจัดการข้อมูลกิจกรรม</li>
        <li>ส่วนจัดการข้อมูลการเข้าร่วมกิจกรรม</li>
    </td> </tr>
</table> <br>
<?php
    include ("footer.php");
?>
</body>
</html>

```

2) ออกจากระบบ ชื่อไฟล์ logout.php

logout.php

```

<?php session_start ( );
    if ($_SESSION["user"]) session_destroy ( );
    header ("Location: index.php"); exit;
?>

```

3) ส่วนติดต่อฐานข้อมูล ชื่อไฟล์ connect_database.php มีสคริปต์ดังนี้



connect_database.php

```
<?php
$link = mysql_pconnect ("localhost", "parinya", "12345");
if (!$link) die (" Can not connect to MySQL" . mysql_error ( ));
$db_selected = mysql_select_db ("university", $link);
if (!$db_selected) die ("Can not select DB" . mysql_error ( ));
?>
```

1.3.3 จัดการข้อมูลนักศึกษา มีรายละเอียดดังนี้

เว็บเพจหลักจัดการข้อมูลนักศึกษา ชื่อไฟล์ admin_student.php การทำงานภายในหน้าเว็บเพจนี้ใช้เทคนิควิธีการแสดงผลข้อมูลจากตารางนักศึกษา และสร้างความสัมพันธ์ไปยังตารางสาขาวิชา และคงจะเพื่อความสมบูรณ์ของการแสดงผล สำหรับหน้าเว็บเพจนี้จะมีการเชื่อมโยงสำหรับบริหารจัดการข้อมูลนักศึกษา ประกอบด้วย 4 ส่วนหลัก อธิบายรายละเอียดดังนี้

รหัสนักศึกษา	ชื่อสกุล	วันเกิด	ต่อไป	สาขาวิชา	คณะ
51047	ธีรวัฒนา น้อยดอนไฟฟ์	1978-02-01	ถ่ายรูปหน้า	วิทยาการคอมพิวเตอร์	วิทยาศาสตร์
57003	คงศักดิ์ วนิษฐ์	1991-08-20	บุพเพ	การจัดการห้องเรียน	วิทยาศาสตร์
57004	นราพร พานิช	1978-03-21	ถ่ายรูปหน้า	วิทยาการคอมพิวเตอร์	วิทยาศาสตร์และเทคโนโลยี
57005	กฤษณา นันทะยาด	1988-03-12	ถ่ายรูปหน้า	วิทยาการคอมพิวเตอร์	วิทยาศาสตร์และเทคโนโลยี

ภาพที่ ข-5 หน้าเว็บเพจสำหรับจัดการข้อมูลนักศึกษา

1) เพิ่มข้อมูลนักศึกษาใหม่ เชื่อมโยงไปยังไฟล์ student_form_insert.php ใช้สำหรับเพิ่มข้อมูลนักศึกษาใหม่ผ่านฟอร์มกำหนด หลังจากคลิกปุ่มบันทึกข้อมูล จะส่งค่าไปยังไฟล์ student_insert.php เพื่อบันทึกข้อมูล แล้วกลับมาที่ไฟล์ admin_student.php

ภาพที่ ข-6 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลนักศึกษาใหม่

2) แก้ไขข้อมูล เชื่อมโยงไปยังไฟล์ student_form_edit.php ค่าที่ส่งไปประมวลผล ดังนี้ student_form_edit.php?student_id=รหัสนักศึกษา เมื่อแก้ไขข้อมูลนักศึกษาเสร็จ จะส่งค่าไปยังไฟล์ student_edit.php เพื่อปรับปรุงข้อมูล แล้วกลับมาที่ไฟล์ admin_student.php

ภาพที่ ข-7 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลนักศึกษา

3) ลบข้อมูล เชื่อมโยงไปยังไฟล์ student_delete.php ค่าที่ส่งไปประมวลผล ดังนี้ student_delete.php?student_id=รหัสนักศึกษา เมื่อลบข้อมูลนักศึกษาเสร็จ จะกลับมาที่ไฟล์ admin_student.php

4) บันทึกกิจกรรม เชื่อมโยงไปยังไฟล์ activity_show.php ในหน้าเว็บเพจนี้จะแสดงข้อมูลการทำกิจกรรมของนักศึกษารายบุคคล การเชื่อมโยงจากหน้าจัดการข้อมูลนักศึกษาหลัก ค่าที่ส่งไปประมวลผล ดังนี้ activity_show.php?student_id=รหัสนักศึกษา (การทำกิจกรรมของนักศึกษาจะบันทึกเป็นประวัติการทำกิจกรรมรายบุคคล) จะมีส่วนย่ออยู่อีก 3 ส่วนย่อย ประกอบด้วย

รหัสกิจกรรม	ชื่อ-สกุlnักศึกษา:	วันที่กิจกรรม	สถานที่
00001	To be Number 1	เก็บบัช	วิทยาการคอมพิวเตอร์ สังกัดคณะฯ วิทยาศาสตร์และเทคโนโลยี
00002	หันสอนปามาเนียทีมายา	อัลลี่เบกุพรหารส์ยูจาร	สถานที่ท่องเที่ยว
00003	มหาวิทยาลัยฯ ปลดบุญ	ร่วมกิจกรรม	มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

ภาพที่ ข-8 หน้าเว็บเพจแสดงข้อมูลการทำกิจกรรมของนักศึกษา (รายบุคคล)

4.1) เพิ่มข้อมูลกิจกรรม เชื่อมโยงไปยังไฟล์ activity_form_insert.php ค่าที่ต้องส่งไปประมวลผล คือ รหัสนักศึกษา ดังนี้ activity_form_insert.php?student_id=รหัสนักศึกษา เมื่อคลิกปุ่มบันทึก จะส่งค่าไปประมวลผลต่อที่ไฟล์ activity_insert.php หลังจากนั้นกลับมาที่ไฟล์ activity_show.php พร้อมค่าของตัวแปร student_id ดังนี้ activity_show.php?student_id=รหัสนักศึกษา

ยินดีต้อนรับคุณ admin เข้าสู่ระบบ

[Logout]

ระบบบันทึกการทำกิจกรรมของนักศึกษา มหาวิทยาลัยราชภัฏอุบลราชธานี

[กลับหน้าหลัก] [จัดการข้อมูลนักศึกษา] [จัดการข้อมูลอาจารย์] [จัดการข้อมูลโครงการ]

รหัสนักศึกษา: 57004 ชื่อ-สกุlnักศึกษา: นวลพรรณ โพธิ์เรือง สาขาวิชา: วิทยาการคอมพิวเตอร์ สังกัดคณะ: วิทยาศาสตร์และเทคโนโลยี

ใช้งานครั้ง: มากที่สุด ปัจจุบัน

ชื่อผู้ใช้งาน: _____

รหัสผ่าน: _____

สถานที่ทํางาน: _____

จําแนกบัญชี ลําดับบัญชี

พัฒนาโดย อ.เบรินญา น้อมดอนไฟฟ์ Copyright © All Rights Reserved

ภาพที่ ข-9 หน้าเว็บเพจแสดงตัวอย่างการเพิ่มข้อมูลการทำกิจกรรม

4.2) แก้ไขข้อมูลการทำกิจกรรม เชื่อมโยงไปยังไฟล์ activity_form_edit.php ค่าที่ส่งไปประมวลผล มี 2 ค่า ประกอบ รหัสนักศึกษาและรหัสโครงการ ดังนี้ activity_form_edit.php ?student_id=รหัสนักศึกษา&project_id=รหัสโครงการ เมื่อคลิกปุ่มบันทึก จะส่งค่าไปประมวลผลต่อที่ไฟล์ activity_edit.php หลังจากนั้นกลับมาที่ไฟล์ activity_show.php พร้อมค่าของตัวแปร student_id ดังนี้ activity_show.php?student_id=รหัสนักศึกษา

4.3) ลบข้อมูลการทำกิจกรรม เชื่อมโยงไปยังไฟล์ activity_delete.php ค่าที่ส่งไปประมวลผล มี 2 ค่า ประกอบด้วย รหัสนักศึกษาและรหัสโครงการ ดังนี้ activity_delete.php ?student_id=รหัสนักศึกษา&project_id=รหัสโครงการ หลังจากนั้นกลับมาที่ไฟล์ activity_show.php พร้อมค่าของตัวแปร student_id ดังนี้ activity_show.php?student_id=รหัสนักศึกษา

ไฟล์ที่เกี่ยวข้องกับการจัดการข้อมูลนักศึกษา มีรายละเอียดแต่ละไฟล์ ดังต่อไปนี้
admin_student.php

```
<?php
    session_start();
    if (!isset ($_SESSION["admin"])) {
        header ("Location: index.php");
        exit;
    }
    include ("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
```



```
font-family: Arial, Helvetica, sans-serif;
}

body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}

-->

</style></head>

<body>
<?php
    include("admin_header.php");
?>
<br> <center>
<table align="center" width="80%" border="1" bgcolor="#EEEEEE">
<tr>
<td> <center>รหัสนักศึกษา</center> </td>
<td> <center>ชื่อสกุล</center> </td>
<td> <center>วันเกิด</center> </td>
<td> <center>ที่อยู่</center> </td>
<td> <center>สาขาวิชา</center> </td>
<td> <center>คณะ</center> </td>
</tr>
<?
    $query = sprintf("SELECT * FROM `student`, `program`, `faculty` WHERE
(`student`.`program_id`='program`.`program_id`) AND `program`.`faculty_id` =
`faculty`.`faculty_id` ORDER BY `student`.`student_id` ASC;");
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
    while ($row = mysql_fetch_assoc ($result)) {
        echo "<tr>";
        echo "<td> " . $row ["student_id"] . "</td>";
        echo "<td> " . $row ["student_name"] . "</td>";
        echo "<td> " . $row ["birth_day"] . "</td>";
        echo "<td> " . $row ["address"] . "</td>";
        echo "<td> " . $row ["program_name"] . "</td>";
```

```

        echo "<td> " . $row ["faculty_name"] . "</td>";
    ?>
        <td align="center">
            [ <a href="student_form_edit.php?student_id=<?php echo $row ["student_id"];?>">
                แก้ไข</a> ]
            [ <a href="student_delete.php?student_id=<?php echo $row ["student_id"];?>">ลบ
                </a> ]
            [ <a href="activity_show.php?student_id=<?php echo $row["student_id"];?>">บันทึก
                กิจกรรม</a> ]</td>
        <?php
            echo "</tr>";
        }
    ?>
</td> </tr> </table>
[ <a href="student_form_insert.php">เพิ่มข้อมูลนักศึกษาใหม่</a> ]
</center> <br>
<?php
    include("footer.php");
?>
</body>
</html>
<?php
    mysql_free_result ($result);
?>

```

student_form_insert.php

```

<?php
    session_start();
    if(!isset($_SESSION["admin"])) {
        header("Location: index.php");
        exit;
    }
    include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>

```



```
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>
<body>
<?php
    include("admin_header.php");
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr> <td>
<form action="student_insert.php" method="post" name="frmstudent">
    <table width="100%" border="0">
        <tr>
            <td width="34%"><div align="right">รหัสนักศึกษา::</div></td>
            <td width="66%"><input name="student_id" type="text" id="student_id" size="15" maxlength="5"></td>
        </tr> <tr> <td><div align="right">ชื่อสกุล::</div></td>
            <td><input name="student_name" type="text" id="student_name" size="40" maxlength="40"></td>
        </tr>
        <tr> <td><div align="right">วันเกิด::</div></td>
            <td><input name="birth_day" type="text" id="birth_day" size="20" maxlength="10"></td> </tr> <tr>
            <td><div align="right">ที่อยู่::</div></td>
            <td><input name="address" type="text" id="address" size="40" maxlength="40"></td>
        </tr> <tr> <td><div align="right">สาขาวิชา::</div></td>
    </table>
<?php
```

```

$query = "SELECT * FROM program;";
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
?>
<td>
<select name="program_id" id="program_id">
<?
while ($row = mysql_fetch_assoc ($result)) {
?>
<option value="
<?php echo $row ["program_id"];?>">
<?php echo $row ["program_name"]; ?>
</option>
<?php }?>
</select>
</td> </tr> <tr>
<td>&nbsp;</td>
<td><input type="submit" name="btnInsert" id="btnInsert" value="เพิ่มข้อมูล">
<input type="reset" name="btnCancel" id="btnCancel" value="ล้างข้อมูล"></td>
</tr>
</table>
</form>
</td>
</tr>
</table> <br>
<?php
    include ("footer.php");
    mysql_free_result ($result);
?>
</body>
</html>

```

student_insert.php

```

<?php
    include("connect_database.php");
    $query = sprintf ("INSERT INTO student VALUES ('%s', '%s', '%s', '%s', '%s');",
    $_POST["student_id"], $_POST["student_name"], $_POST["birth_day"], $_POST["address"],
    $_POST["program_id"]);

```



```
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
header ('Location: admin_student.php');
?>
```

student_form_edit.php

```
<?php
    session_start();
    if(!isset($_SESSION["admin"])) {
        header("Location: index.php");
        exit;
    }
    include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>

<body>
<?php
    include("admin_header.php");
?>
<?php
```

```

$query = sprintf("SELECT * FROM student WHERE student_id='%s';",
$_GET["student_id"]);
$show_result = mysql_query ($query, $link);
if (!$show_result) die ("Can not Query because: " . mysql_error ());
$student_show = mysql_fetch_assoc ($show_result);
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr> <td>
<form action="student_edit.php" method="post" name="frmstudent">
<table width="100%" border="0">
<tr>
<td width="34%"><div align="right">รหัสนักศึกษา::</div></td>
<td width="66%"><input name="student_id" type="text" id="student_id"
value=<?php echo $student_show["student_id"];?>" size="15" maxlength="5"></td>
</tr> <tr>
<td> <div align="right">ชื่อสกุล::</div></td>
<td> <input name="student_name" type="text" id="student_name" value=<?php
echo $student_show["student_name"];?>" size="40" maxlength="40"></td>
</tr> <tr>
<td><div align="right">วันเกิด::</div></td>
<td><input name="birth_day" type="text" id="birth_day" value=<?php echo
$student_show["birth_day"];?>" size="20" maxlength="10"></td>
</tr> <tr>
<td> <div align="right">ที่อยู่::</div></td>
<td> <input name="address" type="text" id="address" value=<?php echo
$student_show["address"];?>" size="40" maxlength="40"></td>
</tr> <tr>
<td><div align="right">สาขาวิชา::</div></td>
<?php
    $query = "SELECT * FROM program;";
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
?>
    <td><select name="program_id" id="program_id">
<?php
        while ($row = mysql_fetch_assoc ($result)) {
?>

```



```

<option value=<?php echo $row ["program_id"];?>> <?php if($row
["program_id"]===$student_show["program_id"]) echo "selected"; ?>><?php echo $row
["program_name"]; ?></option>
<?php ?>
</select>
</td> </tr> <tr> <td>&nbsp;</td>
<td><input type="submit" name="btnInsert" id="btnInsert" value="บันทึกข้อมูล"></td>
</tr> </table> </form> </td> </tr> </table>
<br>
<?php
    include("footer.php");
    mysql_free_result ($result);
?>
</body>
</html>

```

student_edit.php

```

<?
    include ("connect_database.php");
    $query = sprintf ("UPDATE student SET student_name='%s', birth_day='%s',
address='%s', program_id='%s' WHERE student_id='%s';", $_POST["student_name"],
$_POST["birth_day"], $_POST["address"], $_POST["program_id"], $_POST["student_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ( ));
    header('Location: admin_student.php');
?>

```

student_delete.php

```

<?
    include("connect_database.php");
    $query = sprintf ("DELETE FROM student WHERE student.student_id='%s'",
$_GET["student_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Delete in Student table because: " . mysql_error ( ));
    // Delete all data linking in Activity
    $query = sprintf ("DELETE FROM activity WHERE activity.student_id='%s'",
$_GET["student_id"]);
    $result = mysql_query ($query, $link);

```

```

if (!$result) die ("Can not Delete in Activity table because: " . mysql_error ());
header('Location: admin_student.php');
?>

```

activity_show.php

```

<?php
    session_start();
    if(!isset($_SESSION["admin"])){
        header("Location: index.php");
        exit;
    }
    include ("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>
<body>
<?php
    include("admin_header.php");
?>
<br> <center>
<?php
    $student_query = sprintf("SELECT * FROM `student`, `program`, `faculty` WHERE
`student`.`student_id`='%s' AND `student`.`program_id`='program`.`program_id` AND

```



```

`program`.`faculty_id`='faculty`.`faculty_id` ORDER BY `student`.`student_id` ASC",
$_GET["student_id"]);

$student_result = mysql_query ($student_query, $link);
if (!$student_result) die ("Can not Query because: " . mysql_error ( ));

$student_show = mysql_fetch_assoc ($student_result);

?>
<center>
<?php echo "รหัสนักศึกษา:: ". $student_show["student_id"]. " ชื่อ-สกุlnักศึกษา:: ".
$student_show["student_name"]; ?>
<?php echo " สาขาวิชา:: ". $student_show["program_name"]. " สังกัดคณะ:: ".
$student_show["faculty_name"]; ?>
<center>
<table align="center" width="80%" border="1" bgcolor="#EEEEEE">
<tr>
<td><center>รหัสโครงการ</center></td>
<td><center>ชื่อโครงการ</center></td>
<td><center>กิจกรรมที่ทำ</center></td>
<td><center>วันที่ทำกิจกรรม</center></td>
<td><center>สถานที่</center></td>
</tr>
<?
$query = sprintf("SELECT * FROM student, project, activity WHERE
`activity`.`student_id`='%s' AND `activity`.`project_id`='project`.`project_id` GROUP BY
`project`.`project_id`;", $_GET["student_id"]);

$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ( ));

while ($row = mysql_fetch_assoc ($result)) {
    echo "<tr>";
    echo "<td> " . $row ["project_id"] . "</td>";
    echo "<td> " . $row ["project_name"] . "</td>";
    echo "<td> " . $row ["activity_name"] . "</td>";
    echo "<td> " . $row ["activity_date"] . "</td>";
    echo "<td> " . $row ["activity_place"] . "</td>";
?>
    <td align="center">[ <a href="activity_form_edit.php?student_id=<?php echo $row
["student_id"];?>&project_id=<?php echo $row ["project_id"];?>">แก้ไข</a> ] [ <a
href="activity_delete.php?student_id=<?php echo $row
["student_id"];?>&activity_id=<?php echo $row ["activity_id"];?>">ลบ</a> ] </td>

```

```

<?php
    echo "</tr>";
}
?>
</td> </tr> </table>
[ <a href="activity_form_insert.php?student_id=<?php echo $_GET["student_id"];?>">เพิ่ม
ข้อมูลกิจกรรม</a> ]
</center> <br>
<?php
    include("footer.php");
?>
</body>
</html>
<?php
    mysql_free_result ($student_result);
?>

```

activity_form_insert.php

```

<?php
session_start();
if(!isset($_SESSION["admin"])) {
    header("Location: index.php");
    exit;
}
include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}

```



```
body {  
    margin-left: 0px;  
    margin-top: 0px;  
    margin-right: 0px;  
    margin-bottom: 0px;  
}  
-->  
</style></head>  
<body>  
<?php  
    include("admin_header.php");  
?>  
<br> <center>  
<?php  
    $student_query = sprintf("SELECT * FROM `student`, `program`, `faculty` WHERE  
    `student`.`student_id`='%s' AND `student`.`program_id`=`program`.`program_id` AND  
    `program`.`faculty_id`=`faculty`.`faculty_id` ORDER BY `student`.`student_id` ASC;",  
    $_GET["student_id"]);  
    $student_result = mysql_query ($student_query, $link);  
    if (!$student_result) die ("Can not Query because: " . mysql_error ( ));  
    $student_show = mysql_fetch_assoc ($student_result);  
?>  
<center>  
<?php echo "รหัสนักศึกษา:: ". $student_show["student_id"]. " ชื่อ-สกุlnักศึกษา:: ".  
$student_show["student_name"]; ?>  
<?php echo " สาขาวิชา:: ". $student_show["program_name"]. " สังกัดคณะ:: ".  
$student_show["faculty_name"]; ?>  
<center> <br>  
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">  
    <tr> <td>  
        <form action="activity_insert.php" method="post" name="frmActivity"  
id="frmActivity">
```

```

<table width="100%" border="0">
<tr>
<td width="34%><div align="right">โครงการ:</div></td>
<td width="66%">
<select name="project_id" id="project_id">
<?php
$project_query = "SELECT * FROM project ORDER BY project.project_id ASC;";
$project_result = mysql_query ($project_query, $link);
if (!$project_result) die ("Can not Query because: " . mysql_error ());
while ($row = mysql_fetch_assoc ($project_result)) {
?
<option value="<?php echo $row ["project_id"];?>"><?php echo $row
["project_name"]; ?></option>
<?php }?>
</select></td> </tr> <tr>
<td><div align="right">ชื่อกิจกรรม:</div></td>
<td><input name="activity_name" type="text" id="activity_name" size="40"
maxlength="40"></td> </tr> <tr>
<td><div align="right">วันที่ทำกิจกรรม:</div></td>
<td><input name="activity_date" type="text" id="activity_date" size="20"
maxlength="10"></td> </tr> <tr>
<td><div align="right">สถานที่ทำกิจกรรม:</div></td>
<td><input name="activity_place" type="text" id="activity_place" size="40"
maxlength="40"></td> </tr> <tr>
<td><input name="student_id" type="hidden" id="student_id" value="<?php echo
$_GET["student_id"];?>"></td>
<td><input type="submit" name="btnInsert" id="btnInsert" value="เพิ่มข้อมูล">
<input type="reset" name="btnCancel" id="btnCancel" value="ล้างข้อมูล"></td>
</tr> </table> </form>
</td> </tr>
</table> <br>
<?php

```



```
include("footer.php");
mysql_free_result ($project_result);
mysql_free_result ($student_result);
?>
</body>
</html>
```

activity_insert.php

```
<?
include("connect_database.php");
$query = sprintf ("INSERT INTO activity VALUES (Null,'%s', '%s', '%s', '%s', '%s', '%s');",
$_POST["activity_name"], $_POST["activity_date"], $_POST["activity_place"],
$_POST["student_id"], $_POST["project_id"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Insert data because: " . mysql_error ());
$url=sprintf("Location: activity_show.php?student_id=%s",$_POST["student_id"]);
header($url);
?>
```

activity_form_edit.php

```
<?php
session_start();
if(!isset($_SESSION["admin"])){
    header("Location: index.php");
    exit;
}
include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
```

```

body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>
<body>
<?php
    include("admin_header.php");
?>
<br>
<center>
<?php
    $student_query = sprintf("SELECT * FROM `student`, `program`, `faculty` WHERE
`student`.`student_id`='%s' AND `student`.`program_id`='program`.`program_id` AND
`program`.`faculty_id`='faculty`.`faculty_id` ORDER BY `student`.`student_id` ASC;",
$_GET["student_id"]);
    $student_result = mysql_query ($student_query, $link);
    if (!$student_result) die ("Can not Query because: " . mysql_error ());
    $student_show = mysql_fetch_assoc ($student_result);
?>
<center>
<?php
    echo "รหัสนักศึกษา:: ". $student_show["student_id"]. " ชื่อ-สกุลนักศึกษา:: ".
$student_show["student_name"];
?>
<?php
    echo " สาขาวิชา:: ". $student_show["program_name"]. " สังกัดคณะ:: ".
$student_show["faculty_name"];
?>
<center>
<br>
<?php
    $activity_query = sprintf("SELECT * FROM `student`, `project`, `activity` WHERE
`activity`.`student_id`='%s' AND `activity`.`project_id`='%s';", $_GET["student_id"],
$_GET["project_id"]);

```



```
$activity_result = mysql_query ($activity_query, $link);
if (!$activity_result) die ("Can not Query because: " . mysql_error ());
$activity_show = mysql_fetch_assoc ($activity_result);
?>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
    <tr><td>
        <form action="activity_edit.php" method="post" name="frmActivity"
id="frmActivity">
            <table width="100%" border="0">
                <tr>
                    <td width="34%"><div align="right">โครงการ:</div></td>
                    <td width="66%">
<select name="project_id" id="project_id">
<?php
    $project_query = "SELECT * FROM project ORDER BY project.project_id ASC;";
    $project_result = mysql_query ($project_query, $link);
    if (!$project_result) die ("Can not Query because: " . mysql_error ());
    while ($row = mysql_fetch_assoc ($project_result)) {
?>
    <option value=<?php echo $row ["project_id"];?> <?php if($activity_show
["project_id"]==$row["program_id"]) echo "selected"; ?>>
    <?php
        echo $row ["project_name"];
    ?>
    </option>
<?php }?
</select> </td> </tr> <tr>
    <td><div align="right">ชื่อกิจกรรม:</div></td>
    <td>
        <input name="activity_name" type="text" id="activity_name" value=<?php echo
$activity_show["activity_name"];?>" size="40" maxlength="40"> </td>
    </tr> <tr>
    <td> <div align="right">วันที่ทำกิจกรรม:</div> </td>
    <td>
        <input name="activity_date" type="text" id="activity_date" value=<?php echo
$activity_show["activity_date"];?>" size="20" maxlength="10"></td>
    </tr> <tr>
    <td> <div align="right">สถานที่ทำกิจกรรม:</div> </td>
```

```

<td>
    <input name="activity_place" type="text" id="activity_place" value="<?php echo
$activity_show["activity_place"];?>" size="40" maxlength="40"></td>
</tr> <tr>
    <td>
        <input name="student_id" type="hidden" id="student_id" value="<?php echo
$_GET["student_id"];?>">
        <input name="activity_id" type="hidden" id="activity_id" value="<?php echo
$activity_show["activity_id"];?>"></td>
    <td>
        <input type="submit" name="btnInsert" id="btnInsert" value="เพิ่มข้อมูล">
        <input type="reset" name="btnCancel" id="btnCancel" value="ล้างข้อมูล">
    </td> </tr> </table>
</form> </td> </tr> </table> <br>
<?php
    include("footer.php");
    mysql_free_result ($project_result);
    mysql_free_result ($student_result);
?>
</body>
</html>

```

activity_edit.php

```

<?php
    include ("connect_database.php");
    $query = sprintf ("UPDATE activity SET activity_name='%s', activity_date='%s',
activity_place='%s', student_id='%s', project_id='%s' WHERE activity_id=%d;",
    $_POST["activity_name"], $_POST["activity_date"], $_POST["activity_place"],
    $_POST["student_id"], $_POST["project_id"], $_POST["activity_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
    $url=sprintf("Location: activity_show.php?student_id=%s", $_POST["student_id"]);
    header($url);
?>

```

activity_delete.php

```

<?
    include ("connect_database.php");

```



```

$query = sprintf ("DELETE FROM activity WHERE activity.activity_id=%d;",
$_GET["activity_id"]);

$result = mysql_query ($query, $link);
if (!$result) die ("Can not Delete in Activity table because: " . mysql_error ());
$url=sprintf("Location: activity_show.php?student_id=%s", $_GET["student_id"]);
header($url);

?>

```

1.3.4 ส่วนจัดการข้อมูลสาขาวิชา มีรายละเอียดดังนี้

เว็บเพจหลักจัดการข้อมูลสาขาวิชา ชื่อไฟล์ admin_program.php การทำงานภายในหน้าเว็บเพจนี้ใช้เทคนิควิธีการแสดงผลข้อมูลจากตารางสาขาวิชา และสร้างความสัมพันธ์ไปยังตารางคนเพื่อความสมบูรณ์ของการแสดงผล สำหรับหน้าเว็บเพจนี้จะมีการเชื่อมโยงสำหรับบริหารจัดการข้อมูลสาขาวิชา ประกอบด้วย 3 ส่วนหลัก อธิบายรายละเอียดดังนี้

The screenshot shows a web browser window with the title 'ยินดีต้อนรับคุณ admin เข้าสู่ระบบ'. The URL is 'localhost/student/admin_program.php'. The page contains a navigation menu with links: [Logout], [กลับหน้าหลัก], [จัดการข้อมูลนักศึกษา], [จัดการข้อมูลสาขาวิชา] (highlighted), [จัดการข้อมูลคณาจารย์], and [จัดการข้อมูลโครงการ]. Below the menu is a table listing subjects:

รหัสสาขาวิชา	ชื่อสาขาวิชา	ชื่อหัวหน้าสาขาวิชา	ลេងគណន៍
101	วิทยาการคอมพิวเตอร์	อ.ปรีญญา น้อยดอนไพร	[แก้ไข] [ลบ]
102	ภาษาจีน	อ.เงสินมาศ ลี่เจือง	[แก้ไข] [ลบ]
103	ชีววิทยา	ดร.ศิริรัตน์ เผงชัย	[แก้ไข] [ลบ]
104	การจัดการทรัพยากรัฐมนตรี	ดร.สมพงษ์ ยังคงเมือง	[แก้ไข] [ลบ]
105	เทคโนโลยีสารสนเทศ	นรนงค์ศักดิ์ จาบังกร	[แก้ไข] [ลบ]
107	ภาษาอังกฤษ	พงษ์ภานุ	[แก้ไข] [ลบ]

At the bottom left is a button labeled '[เพิ่มข้อมูลสาขาวิชาใหม่]' (1). At the top right is a button labeled '[ลบ]' (2). A large arrow points from the bottom right towards the table area. At the very top right is a button labeled '3'.

ภาพที่ ข-10 หน้าเว็บเพจสำหรับจัดการข้อมูลสาขาวิชา

1) เพิ่มข้อมูลสาขาวิชาใหม่ เชื่อมโยงไปยังไฟล์ program_form_insert.php ใช้สำหรับเพิ่มข้อมูลสาขาวิชาใหม่ผ่านฟอร์มกำหนด หลังจากคลิกปุ่มบันทึกข้อมูล จะส่งค่าไปยังไฟล์ program_insert.php เพื่อบันทึกข้อมูล แล้วกลับมาที่ไฟล์ admin_program.php

The screenshot shows a web browser window with the title 'ยินดีต้อนรับคุณ admin เข้าสู่ระบบ'. The URL is 'localhost/student/program_form_insert.php'. The page contains a navigation menu with links: [Logout], [กลับหน้าหลัก], [จัดการข้อมูลนักศึกษา], [จัดการข้อมูลสาขาวิชา] (highlighted), [จัดการข้อมูลคณาจารย์], and [จัดการข้อมูลโครงการ]. Below the menu is a form for adding a new subject:

รหัสสาขาวิชา:	<input type="text"/>
ชื่อสาขาวิชา:	<input type="text"/>
ชื่อหัวหน้าสาขาวิชา:	<input type="text"/>
ลេងគណន៍:	<input type="radio"/> วิทยาการคอมพิวเตอร์และเทคโนโลยี <input type="radio"/> ชีววิทยา <input type="radio"/> การจัดการทรัพยากรัฐมนตรี <input type="radio"/> ภาษาจีน <input type="radio"/> ภาษาอังกฤษ
<input type="button" value="เพิ่มนักเรียน"/> <input type="button" value="ล้างข้อมูล"/>	

At the bottom is a copyright notice: 'พัฒนาระบบโดย อ.ปรีญญา น้อยดอนไพร Copyright © All Rights Reserved'.

ภาพที่ ข-11 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลสาขาวิชาใหม่

2) แก้ไขข้อมูลสาขาวิชา เชื่อมโยงไปยังไฟล์ program_form_edit.php ค่าที่ส่งไปประมวลผล ดังนี้ program_form_edit.php?program_id=รหัสสาขาวิชา เมื่อแก้ไขข้อมูลสาขาวิชาเสร็จจะส่งค่าไปยังไฟล์ program_edit.php เพื่อปรับปรุงข้อมูล แล้วกลับมาที่ไฟล์ admin_program.php

ล็อกอินผ่านหน้าเว็บไซต์ของมหาวิทยาลัยราชภัฏธัญรักษ์

[Logout](#)

ระบบบันทึกการทำกิจกรรมของนักศึกษา มหาวิทยาลัยราชภัฏธัญรักษ์นี้

[ล็อกอินผ่านเว็บไซต์] [ล็อกออกโดยไม่ต้องจำ] [ล็อกออกโดยคลิกขวา] [ล็อกออกโดยคลิกที่หน้าจอ] [ล็อกออกโดยคลิกที่หน้าจอ]

รหัสสาขาวิชา: 101
ชื่อสาขาวิชา: วิทยาการคอมพิวเตอร์
ชื่อหน่วยงานที่สอน: มหาวิทยาลัยราชภัฏธัญรักษ์
สถานะ: ใช้งานอยู่ ปีสอน: ปี 2558
ปีสอน: ปี 2558

พัฒนาโดย นพ. อรุณรัตน์ ธรรมรงค์ Copyright © All Rights Reserved

ภาพที่ ข-12 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลสาขาวิชา

3) ลบข้อมูลสาขาวิชา เชื่อมโยงไปยังไฟล์ program_delete.php ค่าที่ส่งไปประมวลผล ดังนี้ program_delete.php?program_id=รหัสสาขาวิชา ปัญหาของการลบข้อมูลสาขาวิชา คือ หากลบเฉพาะข้อมูลสาขาวิชา จะทำให้เกิดข้อผิดพลาดในตารางที่สร้างความสัมพันธ์มายังตารางสาขาวิชา นั่นก็คือ ตารางข้อมูลนักศึกษา ดังนั้นเมื่อมีการลบข้อมูลสาขาวิชาจะต้องไปลบข้อมูลในตารางที่เกี่ยวข้องด้วย หลังจากลบข้อมูลสาขาวิชาเสร็จ จะกลับมาที่ไฟล์ admin_program.php

ไฟล์ที่เกี่ยวข้องกับการจัดการข้อมูลสาขาวิชา มีรายละเอียดแต่ละไฟล์ ดังต่อไปนี้

admin_program.php

```
<?php
    session_start();
    if(!isset($_SESSION["admin"])){
        header("Location: index.php");
        exit;
    }
    include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;

```



```

}

body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}

-->

</style></head>

<body>

<?php
    include("admin_header.php");
?>

<br> <center>
<table align="center" width="60%" border="1" bgcolor="#EEEEEE">
<tr>
<td><center>รหัสสาขาวิชา</center></td>
<td><center>ชื่อสาขาวิชา</center></td>
<td><center>ชื่อหัวหน้าสาขาวิชา</center></td>
<td><center>สังกัดคณะ</center></td>
</tr>
<?php

    $query = sprintf("SELECT * FROM `program`, `faculty` WHERE
`program`.`faculty_id`='faculty`.`faculty_id` ORDER BY `program`.`program_id` ASC;");
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
    while ($row = mysql_fetch_assoc ($result)) {
        echo "<tr>";
        echo "<td><center> " . $row ["program_id"] . "</center></td>";
        echo "<td> " . $row ["program_name"] . "</td>";
        echo "<td> " . $row ["head_name"] . "</td>";
        echo "<td> " . $row ["faculty_name"] . "</td>";
    }
?>
<td align="center">[ <a href="program_form_edit.php?program_id=<?php echo
$row ["program_id"];?>">แก้ไข</a> ] [ <a href="program_delete.php?program_id=<?php
echo $row ["program_id"];?>">ลบ</a> ]</td>
<?php
    echo "</tr>";

```

```

        }
    ?>
</td> </tr>
</table>
[ <a href="program_form_insert.php">เพิ่มข้อมูลสาขาวิชาใหม่</a> ]
</center>
<br>
<?php
    include("footer.php");
?>
</body>
</html>
<?php
    mysql_free_result ($result);
?>

```

program_form_insert.php

```

<?php
    session_start();
    if(!isset($_SESSION["admin"])) {
        header("Location: index.php");
        exit;
    }
    include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;

```



```
margin-bottom: 0px;
}

-->
</style></head>
<body>
<?php
    include("admin_header.php");
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr> <td>
<form action="program_insert.php" method="post" name="frmprogram">
<table width="100%" border="0">
<tr>
    <td width="34%"><div align="right">รหัสสาขาวิชา:</div></td>
    <td width="66%"><input name="program_id" type="text" id="program_id" size="15" maxlength="3"></td>
</tr>
<tr>
    <td><div align="right">ชื่อสาขาวิชา:</div></td>
    <td><input name="program_name" type="text" id="program_name" size="40" maxlength="40"></td>
</tr>
<tr>
    <td><div align="right">ชื่อหัวหน้าสาขาวิชา:</div></td>
    <td><input name="head_name" type="text" id="head_name" size="40" maxlength="40"></td>
</tr>
<tr>
    <td><div align="right">สังกัดคณะ:</div></td>
<?php
    $query = "SELECT * FROM faculty;";
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query from Faculty table because: " . mysql_error ());
?>
    <td><select name="faculty_id" id="faculty_id">
<?php
    while ($row = mysql_fetch_assoc ($result)) {
```

```

?>
    <option value=<?php echo $row ["faculty_id"];?>><?php echo $row
["faculty_name"]; ?></option>
<?php }?>
    </select>
    </td> </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type="submit" name="btnInsert" id="btnInsert" value="เพิ่มข้อมูล">
        <input type="reset" name="btnCancel" id="btnCancel" value="ล้างข้อมูล"></td>
    </tr>
</table>
</form>
</td> </tr>
</table>
<br>
<?php
    include("footer.php");
    mysql_free_result ($result);
?>
</body>
</html>

```

program_insert.php

```

?>
    include ("connect_database.php");
    $query = sprintf ("INSERT INTO program VALUES ('%s', '%s', '%s', '%s');",
$_POST["program_id"], $_POST["program_name"], $_POST["head_name"],
$_POST["faculty_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Insert data because: " . mysql_error ( ));
    header ('Location: admin_program.php');
?>

```

program_form_edit.php

```

<?php
    session_start();
    if(!isset($_SESSION["admin"])) {

```



```
        header("Location: index.php"); exit;  
    }  
    include("connect_database.php");  
?>  
<html>  
<head>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
    <title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>  
    <style type="text/css">  
        <!--  
        body,td,th {  
            font-family: Arial, Helvetica, sans-serif;  
        }  
        body {  
            margin-left: 0px;  
            margin-top: 0px;  
            margin-right: 0px;  
            margin-bottom: 0px;  
        }  
        -->  
    </style></head>  
<body>  
<?php  
    include("admin_header.php");  
?>  
<?php  
    $query = sprintf("SELECT * FROM program WHERE program_id='%s';",  
    $_GET["program_id"]);  
    $show_result = mysql_query ($query, $link);  
    if (!$show_result) die ("Can not Query because: " . mysql_error ( ));  
    $program_show = mysql_fetch_assoc ($show_result);  
?>  
<br>  
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">  
<tr> <td>  
    <form action="program_edit.php" method="post" name="frmprogram">  
        <table width="100%" border="0">  
            <tr> <td width="34%><div align="right">รหัสสาขาวิชา:</div></td>
```

```

<td width="66%>
    <input name="program_id" type="text" id="program_id" value="<?php echo
$program_show["program_id"];?>" size="15" maxlength="5"></td>
</tr> <tr>
<td> <div align="right">ชื่อสาขาวิชา::</div></td>
<td>
    <input name="program_name" type="text" id="program_name" value="<?php echo
echo $program_show["program_name"];?>" size="40" maxlength="40"></td>
</tr> <tr>
<td><div align="right">ชื่อหัวหน้าสาขาวิชา::</div></td>
<td>
    <input name="head_name" type="text" id="head_name" value="<?php echo
$program_show["head_name"];?>" size="40" maxlength="40"></td>
</tr> <tr>
<td><div align="right">สังกัดคณะ::</div></td>
<?php
    $query = "SELECT * FROM faculty;";
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
?
<td><select name="faculty_id" id="faculty_id">
<?php
    while ($row = mysql_fetch_assoc ($result)) {
?
        <option value="<?php echo $row ["faculty_id"];?>" <?php if($row
["faculty_id"]===$program_show["faculty_id"]) echo "selected"; ?>><?php echo $row
["faculty_name"]; ?></option>
?
    }
</select> </td></tr>
<tr>
<td>&nbsp;</td>
<td><input type="submit" name="btnInsert" id="btnInsert" value="บันทึก
ข้อมูล"></td>
</tr>
</table>
</form>
</td> </tr>
</table>

```



```
<br>
<?php
    include("footer.php");
    mysql_free_result ($result);
?>
</body>
</html>
```

program_edit.php

```
<?
    include ("connect_database.php");
    $query = sprintf ("UPDATE program SET program_name='%s', head_name='%s',
faculty_id='%s' WHERE program_id='%s';", $_POST["program_name"],
$_POST["head_name"], $_POST["faculty_id"], $_POST["program_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ( ));
    header ('Location: admin_program.php');
?>
```

program_delete.php

```
<?
    include("connect_database.php");
    // Delete Data in Program
    $query = sprintf ("DELETE FROM program WHERE program.program_id='%s';",
$_GET["program_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Delete in Program table because: " . mysql_error ( ));
    // Delete all Data Linking in program
    $query = sprintf("SELECT * FROM student WHERE student.program_id='%s';",
$_GET["program_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ( ));
    if(mysql_num_rows($result)>0){
        $student = mysql_fetch_assoc ($result);
        $student_query = sprintf("DELETE FROM student WHERE
student.program_id='%s';", $_GET["program_id"]);
        $student_result = mysql_query ($student_query, $link);
    }
?>
```

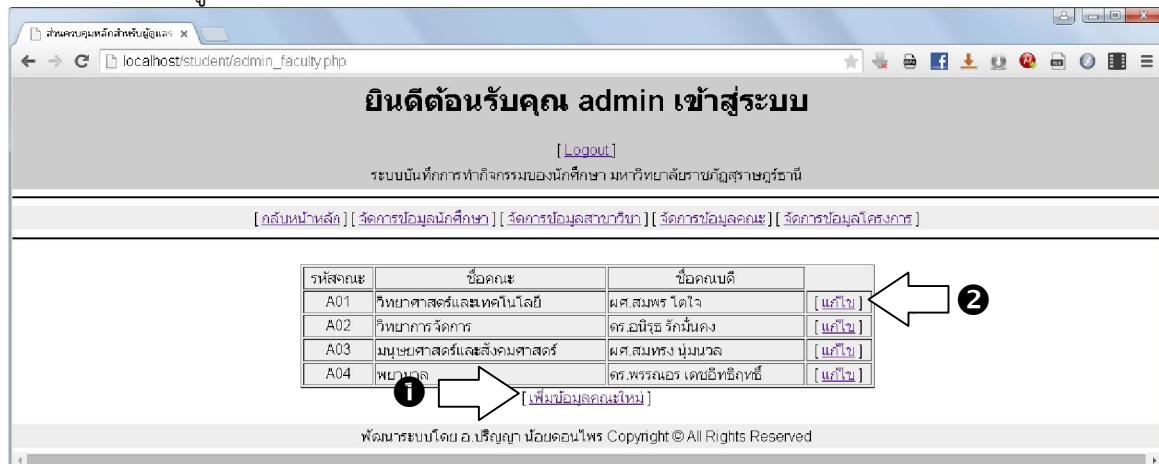
```

$activity_query = sprintf("SELECT * FROM activity WHERE
activity.student_id='%s';", $student["student_id"]);
$activity_result = mysql_query ($activity_query, $link);
if(mysql_num_rows($activity_result)>0){
    $activity_query = sprintf ("DELETE FROM activity WHERE
activity.student_id='%s';", $student["student_id"]);
    $activity_result = mysql_query ($activity_query, $link);
}
header('Location: admin_program.php');
mysql_free_result($result);
mysql_free_result($student_result);
mysql_free_result($activity_result);
?>

```

1.3.5 ส่วนจัดการข้อมูลคณะ มีรายละเอียดดังนี้

เว็บเพจหลักจัดการข้อมูลคณะ ชื่อไฟล์ admin_faculty.php การทำงานภายใต้หน้าเว็บ เพจนี้ใช้เทคนิควิธีการแสดงผลข้อมูลจากตารางคณะ สำหรับหน้าเว็บเพจนี้จะมีการเชื่อมโยงสำหรับ บริหารจัดการข้อมูลคณะ ประกอบด้วย 2 ส่วนหลัก อธิบายรายละเอียดดังนี้



ภาพที่ ข-13 หน้าเว็บเพจสำหรับจัดการข้อมูลคณะ

- เพิ่มข้อมูลคณะใหม่ เชื่อมโยงไปยังไฟล์ faculty_form_insert.php ใช้สำหรับเพิ่ม ข้อมูลคณะใหม่ผ่านฟอร์มกำหนด หลังจากคลิกปุ่มบันทึกข้อมูล จะส่งค่าไปยังไฟล์ faculty_insert.php เพื่อบันทึกข้อมูล และกลับมาที่ไฟล์ admin_faculty.php

ยินดีต้อนรับคุณ admin เข้าสู่ระบบ

[Logout]
ระบบบันทึกการทำกิจกรรมของนักศึกษา มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

[กดลบหน้าหลัก] [จัดการข้อมูลนักศึกษา] [จัดการข้อมูลสาขาวิชา] [จัดการข้อมูลคณะ] [จัดการข้อมูลโครงการ]

รหัสคุณะ: _____
ชื่อคุณะ: _____
ชื่อคุณเบติ: _____
เข้าสู่ระบบ | ลืมรหัสผ่าน

พัฒนาระบบโดย อ.ปรีญญา น้อยดอนไพร Copyright © All Rights Reserved

ภาพที่ ข-14 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลคณะใหม่

2) แก้ไขข้อมูลคณะ เชื่อมโยงไปยังไฟล์ faculty_form_edit.php ค่าที่ส่งไปประมวลผล ดังนี้ faculty_form_edit.php?faculty_id=รหัสคณะ เมื่อแก้ไขข้อมูลคณะเสร็จ จะส่งค่าไปยังไฟล์ faculty_edit.php เพื่อปรับปรุงข้อมูล แล้วกลับมาที่ไฟล์ admin_faculty.php

ยินดีต้อนรับคุณ admin เข้าสู่ระบบ

[Logout]
ระบบบันทึกการทำกิจกรรมของนักศึกษา มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

[กดลบหน้าหลัก] [จัดการข้อมูลนักศึกษา] [จัดการข้อมูลสาขาวิชา] [จัดการข้อมูลคณะ] [จัดการข้อมูลโครงการ]

รหัสคุณะ: A01
ชื่อคุณะ: วิทยาลัยศรีสะเกษในใจ
ชื่อคุณเบติ: นศ. สุภาร โนนไง
บันทึกข้อมูล

พัฒนาระบบโดย อ.ปรีญญา น้อยดอนไพร Copyright © All Rights Reserved

ภาพที่ ข-15 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลคณะ

ไฟล์ที่เกี่ยวข้องกับการจัดการข้อมูลคณะ มีรายละเอียดแต่ละไฟล์ ดังต่อไปนี้
admin_faculty.php

```
<?php
    session_start();
    if(!isset($_SESSION["admin"])){
        header("Location: index.php");
        exit;
    }
    include ("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```

<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>
<body>
<?php
    include ("admin_header.php");
?>
<br>
<center>
<table align="center" width="50%" border="1" bgcolor="#EEEEEE">
<tr>
<td><center>รหัสคณะ</center></td>
<td><center>ชื่อคณะ</center></td>
<td><center>ชื่อคณบดี</center></td>
</tr>
<?
    $query = sprintf("SELECT * FROM `faculty` ORDER BY `faculty`.`faculty_id` ASC;");
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ());
    while ($row = mysql_fetch_assoc ($result)) {
        echo "<tr>";
        echo "<td><center> " . $row ["faculty_id"] . "</center></td>";
        echo "<td> " . $row ["faculty_name"] . "</td>";
        echo "<td> " . $row ["dean_name"] . "</td>";
    }
    <td align="center">[ <a href="faculty_form_edit.php?faculty_id=<?php echo $row
["faculty_id"];?>">แก้ไข</a> ] </td>

```



```
<?php
    echo "</tr>";
}
?>
</td>
</tr>
</table>
[ <a href="faculty_form_insert.php">เพิ่มข้อมูลคนใหม่</a> ]
</center>
<br>
<?php
    include("footer.php");
?>
</body>
</html>
<?php
    mysql_free_result ($result);
?>
```

faculty_form_insert.php

```
<?php
    session_start();
    if(!isset($_SESSION["admin"])) {
        header("Location: index.php");
        exit;
    }
    include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
```

```

margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
}

-->
</style></head>
<body>
<?php
    include ("admin_header.php");
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr> <td>
<form action="faculty_insert.php" method="post" name="frmfaculty">
<table width="100%" border="0">
<tr>
<td width="34%"><div align="right">รหัสคณะ::</div></td>
<td width="66%">
    <input name="faculty_id" type="text" id="faculty_id" size="15" maxlength="3">
</td>
</tr> <tr>
<td> <div align="right">ชื่อคณะ:</div></td>
<td>
    <input name="faculty_name" type="text" id="faculty_name" size="40"
maxlength="40">
</td> </tr> <tr>
<td><div align="right">ชื่อคณบดี:</div></td>
<td>
    <input name="dean_name" type="text" id="dean_name" size="40"
maxlength="40">
</td> </tr> <tr>
<td>&nbsp;</td>
<td>
    <input type="submit" name="btnInsert" id="btnInsert" value="เพิ่มข้อมูล">
    <input type="reset" name="btnCancel" id="btnCancel" value="ล้างข้อมูล">
</td> </tr>
</table>

```



```
</form>
</td> </tr> </table>
<br>
<?php
    include("footer.php");
?>
</body>
</html>
```

faculty_insert.php

```
<?
    include ("connect_database.php");
    $query = sprintf ("INSERT INTO faculty VALUES ('%s', '%s', '%s');", $_POST
["faculty_id"], $_POST ["faculty_name"], $_POST ["dean_name"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Insert data because: " . mysql_error ());
    header ('Location: admin_faculty.php');
?>
```

faculty_form_edit.php

```
<?php
    session_start();
    if(!isset($_SESSION["admin"])){
        header("Location: index.php");
        exit;
    }
    include("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
```

```

margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
}

-->
</style></head>
<body>
<?php
    include ("admin_header.php");
?>
<?php
    $query = sprintf("SELECT * FROM faculty WHERE faculty_id='%s';",
$_GET["faculty_id"]);
    $show_result = mysql_query ($query, $link);
    if (!$show_result) die ("Can not Query because: " . mysql_error ());
    $faculty_show = mysql_fetch_assoc ($show_result);
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr> <td>
<form action="faculty_edit.php" method="post" name="frmfaculty">
<table width="100%" border="0">
<tr>
<td width="34%"><div align="right">รหัสคณะ::</div></td>
<td width="66%">
    <input name="faculty_id" type="text" id="faculty_id" value="<?php echo
$faculty_show["faculty_id"];?>" size="15" maxlength="3">
</td> </tr> <tr>
<td><div align="right">ชื่อคณะ::</div></td>
<td>
    <input name="faculty_name" type="text" id="faculty_name" value="<?php echo
$faculty_show["faculty_name"];?>" size="40" maxlength="40">
</td> </tr> <tr>
<td><div align="right">ชื่อคณบดี::</div></td>
<td>
    <input name="dean_name" type="text" id="dean_name" value="<?php echo
$faculty_show ["dean_name"];?>" size="40" maxlength="40">

```



```

</td>
</tr> <tr>
<td>&nbsp;</td>
<td>
    <input type="submit" name="btnInsert" id="btnInsert" value="บันทึกข้อมูล">
</td> </tr>
</table>
</form> </td> </tr> </table>
<br>
<?php
    include ("footer.php");
?>
</body>
</html>

```

faculty_edit.php

```

<?php
    include ("connect_database.php");
    $query = sprintf ("UPDATE faculty SET faculty_name='%s', dean_name='%s'
WHERE faculty_id='%s';", $_POST["faculty_name"], $_POST["dean_name"],
$_POST["faculty_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ( ));
    header('Location: admin_faculty.php');
?>

```

1.3.6 ส่วนจัดการข้อมูลโครงการ มีรายละเอียดดังนี้

เว็บเพจหลักจัดการข้อมูลโครงการ ชื่อไฟล์ admin_project.php การทำงานภายใต้หน้าเว็บเพจนี้ใช้เทคนิควิธีการแสดงผลข้อมูลจากตารางโครงการ สำหรับหน้าเว็บเพจนี้จะมีการเชื่อมโยง สำหรับบริหารจัดการข้อมูลโครงการ ประกอบด้วย 3 ส่วนหลัก อธิบายรายละเอียดดังนี้

The screenshot shows a Windows desktop environment with a browser window open to `localhost/student/admin_project.php`. The title bar says "ยินดีต้อนรับคุณ admin เข้าสู่ระบบ". The main content area displays a table with three rows of project data:

รหัสโครงการ	ชื่อโครงการ	รายละเอียดโครงการ	แก้ไข []	ลบ []
00001	To be Number 1	เบ้าชนิดที่สามารถนำไปใช้ในงานวิชาชีพได้	[แก้ไข]	[ลบ]
00002	รับสถาปัตย์ใหม่	14 ภูมิภาคที่มีอยู่ทั่วโลก	[แก้ไข]	[ลบ]
00003	นักวิทยาศาสตร์ไทย	กลุ่มของนักวิทยาศาสตร์ที่มีความสามารถด้านวิทยาศาสตร์	[แก้ไข]	[ลบ]

Below the table, there is a note: "[ลงทะเบียนโครงการใหม่]". At the bottom of the page, it says "พัฒนาโดย อ.ปรีญญา ป้อบดอนไพโร Copyright © All Rights Reserved". Three numbered arrows point to specific elements: arrow 1 points to the "注册新项目" button, arrow 2 points to the "Edit" button for the first row, and arrow 3 points to the "Delete" button for the first row.

ภาพที่ ข-16 หน้าเว็บเพจสำหรับจัดการข้อมูลโครงการ

1) เพิ่มข้อมูลโครงการใหม่ เชื่อมโยงไปยังไฟล์ project_form_insert.php ใช้สำหรับเพิ่มข้อมูลโครงการใหม่ผ่านฟอร์มกำหนด หลังจากคลิกปุ่มบันทึกข้อมูล จะส่งค่าไปยังไฟล์ project_insert.php เพื่อบันทึกข้อมูล แล้วกลับมาที่ไฟล์ admin_project.php

ภาพที่ ข-17 หน้าเว็บเพจแบบฟอร์มสำหรับเพิ่มข้อมูลโครงการใหม่

2) แก้ไขข้อมูลโครงการ เชื่อมโยงไปยังไฟล์ project_form_edit.php ค่าที่ส่งไปประมวลผล ดังนี้ project_form_edit.php?project_id=รหัสโครงการ เมื่อแก้ไขข้อมูลโครงการเสร็จ จะส่งค่าไปยังไฟล์ project_edit.php เพื่อปรับปรุงข้อมูล แล้วกลับมาที่ไฟล์ admin_project.php

ภาพที่ ข-18 หน้าเว็บเพจแบบฟอร์มสำหรับแก้ไขข้อมูลโครงการ

3) ลบข้อมูลโครงการ เชื่อมโยงไปยังไฟล์ project_delete.php ค่าที่ส่งไปประมวลผล ดังนี้ project_delete.php?project_id=รหัสโครงการ ปัญหาของการลบข้อมูลโครงการ คือ หากลบเฉพาะข้อมูลโครงการ จะทำให้เกิดข้อผิดพลาดในตารางที่สร้างความสัมพันธ์มายังตารางโครงการ นั่นก็คือ ตารางข้อมูลกิจกรรม ดังนั้นเมื่อการลบข้อมูลโครงการจะต้องลบข้อมูลในตารางที่เกี่ยวข้องด้วย หลังจาก ลบข้อมูลโครงการเสร็จ จะกลับมาที่ไฟล์ admin_project.php

ไฟล์ที่เกี่ยวข้องกับการจัดการข้อมูลสาขาวิชา มีรายละเอียดแต่ละไฟล์ ดังต่อไปนี้

project_form_insert.php

```
<?php
```



```
session_start();
if (!isset ($_SESSION["admin"])) {
    header ("Location: index.php"); exit;
}
include ("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>
<body>
<?php
    include("admin_header.php");
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr> <td>
<form action="project_insert.php" method="post" name="frmProject" id="frmProject">
<table width="100%" border="0">
<tr> <td width="34%"><div align="right">รหัสโครงการ:</div></td>
<td width="66%">
    <input name = "project_id" type = "text" id = "project_id" size = "15"
maxlength="5">
</td> </tr> <tr>
<td> <div align="right">ชื่อโครงการ:</div> </td>
```

```

<td>
    <input name = "project_name" type = "text" id = "project_name" size = "40"
maxlength="40">
</td> </tr> <tr>
<td valign="top"><div align="right">รายละเอียดโครงการ::</div></td>
<td>
    <textarea name="detail" cols="40" rows="5" id="detail"></textarea></td>
</tr> <tr>
<td>&nbsp;</td>
<td>
    <input type="submit" name="btnInsert" id="btnInsert" value="เพิ่มข้อมูล">
    <input type="reset" name="btnCancel" id="btnCancel" value="ล้างข้อมูล">
</td>
</tr> </table>
</form>
</td> </tr>
</table>
<br>
<?php
    include("footer.php");
?>
</body>
</html>

```

project_insert.php

```

<?
    include ("connect_database.php");
    $query = sprintf ("INSERT INTO project VALUES ('%s', '%s', '%s');", $_POST
["project_id"], $_POST ["project_name"], $_POST ["detail"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Insert data because: " . mysql_error ());
    header ('Location: admin_project.php');
?>

```

project_form_edit.php

```

<?php
    session_start();
    if(!isset($_SESSION["admin"])) {

```



```
        header("Location: index.php");
        exit;
    }
    include ("connect_database.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>ส่วนควบคุมหลักสำหรับผู้ดูแลระบบ</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>
<body>
<?php
    include("admin_header.php");
?>
<?php
    $query = sprintf("SELECT * FROM project WHERE project_id = '%s';",
$_GET["project_id"]);
    $show_result = mysql_query ($query, $link);
    if (!$show_result) die ("Can not Query because: " . mysql_error ());
    $project_show = mysql_fetch_assoc ($show_result);
?>
<br>
<table align="center" width="40%" border="0" bgcolor="#CCCCCC">
<tr>
<td>
<form action="project_edit.php" method="post" name="frmProject" id="frmProject">
```

```

<table width="100%" border="0">
<tr>
<td width="34%"><div align="right">รหัสโครงการ:</div></td>
<td width="66%">
    <input name="project_id" type="text" id="project_id" value="<?php echo
$project_show["project_id"];?>" size="15" maxlength="5">
</td> </tr>
<tr>
<td><div align="right">ชื่อโครงการ:</div></td>
<td>
    <input name="project_name" type="text" id="project_name" value="<?php echo
$project_show["project_name"];?>" size="40" maxlength="40"></td>
</tr>
<tr>
<td><div align="right">รายละเอียดโครงการ:</div></td>
<td>
    <input name="detail" type="text" id="detail" value="<?php echo
$project_show["detail"];?>" size="40" maxlength="40"></td>
</tr> <tr>
<td>&nbsp;</td>
<td>
    <input type="submit" name="btnInsert" id="btnInsert" value="บันทึกข้อมูล"></td>
</tr> </table> </form> </td> </tr> </table> <br>
<?php
    include("footer.php");
?>
</body>
</html>

```

project_edit.php

```

<?
include ("connect_database.php");
$query = sprintf ("UPDATE project SET project_name='%s', detail='%s' WHERE
project_id='%s';", $_POST["project_name"], $_POST["detail"], $_POST["project_id"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
header ('Location: admin_project.php');
?>

```



project_edit.php

```
<?
include ("connect_database.php");
// Delete Data in Program
$query = sprintf ("DELETE FROM project WHERE project.project_id='%s';",
$_GET["project_id"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Delete in Program table because: " . mysql_error ());
// Delete all Data Linking in program
$query = sprintf("SELECT * FROM activity WHERE activity.project_id='%s';",
$_GET["project_id"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ());
if(mysql_num_rows($result)>0){
    $activity = mysql_fetch_assoc ($result);
    $activity_query = sprintf("DELETE FROM activity WHERE
activity.project_id='%s';", $_GET["project_id"]);
    $activity_result = mysql_query ($activity_query, $link);
}
header ('Location: admin_program.php');
mysql_free_result ($result);
mysql_free_result ($activity_result);
?>
```

สรุป

กรณีศึกษา: ระบบบันทึกการทำกิจกรรมของนักศึกษา เป็นตัวอย่างการพัฒนาเว็บแอพคลิเคชัน เพื่อเก็บบันทึกข้อมูลการทำกิจกรรมของนักศึกษา มหาวิทยาลัยราชภัฏสุราษฎร์ธานี ซึ่งเป็นการประมวลความรู้ทักษะในการเขียนสคริปต์ด้วยภาษา PHP ร่วมกับฐานข้อมูล MySQL ลักษณะการเขียนสคริปต์อาจแตกต่างกันแต่ผลลัพธ์ที่ได้เหมือนกัน สำหรับการติดตอกับฐานข้อมูล MySQL หลักการทำงานของ PHP เมื่อต้องการติดต่อและเรียกใช้ฐานข้อมูล จะต้องใช้รูปแบบคำสั่งหรือคำสอบถามข้อมูลบรรทัดเดียว และส่งไปประมวลผลในโปรแกรมฐานข้อมูล MySQL ในกรณีที่คำสั่งหรือคำสอบถามข้อมูลมีความซับซ้อนจำเป็นต้องต่อประโยคคำสั่งหรือคำสอบถามข้อมูลให้เป็นประโยคบรรทัดเดียวกันก่อน และจึงค่อยส่งไปประมวลผล ประเด็นที่น่าสนใจจากตัวอย่างคือ เมื่อมีการลบข้อมูลในตารางใดๆ หากข้อมูลนั้นมีความสัมพันธ์กัน เช่น การใช้อ้างอิงในตารางอื่นๆ หากจำเป็นต้องลบข้อมูลนั้นทิ้ง จำเป็นต้องตามไปลบหรือเปลี่ยนแปลงข้อมูลในตารางต่างๆ ที่เกี่ยวข้องทั้งหมดด้วย เพื่อป้องกันความผิดพลาดในการเรียกสืบค้นข้อมูลในครั้งถัดไป และความน่าเชื่อถือของระบบสารสนเทศที่พัฒนา