



แบบเรียนวิชา

รหัสวิชา

3901-2107

การพัฒนาเว็บ ด้วยภาษา PHP

(Web Development with PHP Language)

อ. ชานูชัย ศุภอรรถกร

ตรงตามหลักสูตร
ของสำนักงานคณะกรรมการการอาชีวศึกษา
พุทธศักราช 2557

การพัฒนาเว็บด้วยภาษา PHP

(Web Development with PHP Language)

ตรงตามหลักสูตรของสำนักงานคณะกรรมการการอาชีวศึกษา พุทธศักราช 2557

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ 2537

จัดพิมพ์ และ จำหน่ายโดย



บริษัท ซัคเซส มีเดีย จำกัด

1000/19-20 อาคารลิเบอร์ตี้พลาซ่า ชั้น 12A
ซอยสุขุมวิท 55 (ทองหล่อ) แขวงคลองตันเหนือ
เขตวัฒนา กรุงเทพฯ 10110
โทร. 0-2762-9000 โทรสาร 0-2762-9001
E-mail : info@successmedia.com
www.successmedia.com

ข้อมูลบรรณานุกรมของสำนักงานหอสมุดแห่งชาติ

ชาญชัย ศุภอรธกร.
การพัฒนาเว็บด้วยภาษา PHP (3901-2107).-- กรุงเทพฯ :
ซัคเซส มีเดีย, 2015.
276 หน้า. -- (แบบเรียน ปวส.).

1. พีเอชพี (ภาษาคอมพิวเตอร์). 2. การพัฒนาเว็บไซต์.
I. ชื่อเรื่อง.

005.133

ISBN 978-616-205-347-4

เรียบเรียงโดย

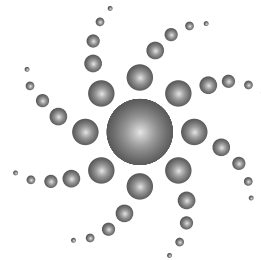
อ. ชาญชัย ศุภอรธกร

การศึกษา :

- ศึกษานิเทศก์ คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
- ศึกษานิเทศก์ คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหิดล

ประวัติการทำงาน :

- ผู้ช่วยนักวิจัย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)
- นักโปรแกรมระบบ บริษัทสยามสารสนเทศและประมวลผล ธนาकारไทยพาณิชย์ จำกัด (มหาชน)
- อาจารย์สาขาคอมพิวเตอร์ธุรกิจ คณะบริหารธุรกิจวิทยาลัยโปไลเทคนิคภาคตะวันออกเฉียงเหนือ อุบลราชธานี
- อาจารย์สาขาคอมพิวเตอร์ธุรกิจ คณะการบัญชีและการจัดการ มหาวิทยาลัยมหาสารคาม
- ปัจจุบันเป็นอาจารย์ภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี



คำนำ

หนังสือแบบเรียนวิชา การพัฒนาเว็บด้วยภาษา PHP รหัสวิชา 3901-2107 มี 3 หน่วยกิต 5 ชั่วโมงต่อสัปดาห์ ซึ่งเรียบเรียงเนื้อหาตรงตามจุดประสงค์รายวิชา มาตรฐานรายวิชา และคำอธิบายรายวิชา ในหลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) พุทธศักราช 2557 ของสำนักงานคณะกรรมการการอาชีวศึกษา

หนังสือแบบเรียนเล่มนี้ ประกอบด้วยเนื้อหา 13 บท ได้แก่ ความรู้เบื้องต้นการเขียนโปรแกรมบนอินเทอร์เน็ต การติดตั้งโปรแกรมเพื่อใช้งาน PHP PHP เบื้องต้น เรื่องของตัวแปร และค่าคงที่ ตัวดำเนินการและนิพจน์ PHP คำสั่งควบคุมสลับเงื่อนไข และคำสั่งทำซ้ำ การสร้างฟังก์ชันของ PHP ฟังก์ชันของ PHP ที่จัดเตรียมไว้ การสร้างช่องรับข้อมูล Session และ Cookie ฐานข้อมูล MySQL การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin การเขียน PHP จัดการ MySQL นอกจากนี้ยังมีแบบฝึกหัดและใบงานฝึกปฏิบัติในส่วนท้ายของแต่ละบท ซึ่งจะช่วยให้ผลการเรียนรู้ และฝึกการเรียนรู้ได้ด้วยตนเอง

คณะผู้เรียบเรียงหวังเป็นอย่างยิ่งว่าหนังสือแบบเรียนเล่มนี้ จะเป็นหนังสือประกอบการเรียนที่เป็นประโยชน์ต่อครูสอนของคณะอาจารย์ และการเรียนรู้ของนักศึกษา เพื่อให้การศึกษาทัดเทียมกับระดับสากล

บรรณาธิการ ชัคเชส มีเดีย

เมษายน 2558

จุดประสงค์ และคำอธิบายรายวิชา

การพัฒนาเว็บด้วยภาษา PHP รหัสวิชา 3901-2107

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง พุทธศักราช 2557

สำนักงานคณะกรรมการอาชีวศึกษา

จุดประสงค์รายวิชา เพื่อให้

1. มีความรู้ความเข้าใจเกี่ยวกับการโปรแกรมเว็บด้วยภาษา PHP
2. ปฏิบัติการโปรแกรมเว็บด้วยภาษา PHP
3. มีกิจนิสัยในการทำงานด้วยความประณีต รอบคอบและปลอดภัย ตระหนักถึงคุณภาพของงานและมีจริยธรรมในงานอาชีพ

มาตรฐานรายวิชา

1. ออกแบบโปรแกรมเว็บด้วยภาษา PHP ให้เหมาะสมกับงาน
2. ปฏิบัติการโปรแกรมเว็บด้วยภาษา PHP ตามที่ผู้สอนออกแบบ
3. ทดสอบโปรแกรมเว็บที่สร้างขึ้น

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการของการสื่อสารข้อมูลบนเครือข่ายอินเทอร์เน็ต รูปแบบ Client Server การทำงานของ Web Server การติดตั้งเครื่องมือและโปรแกรมที่ใช้ในการพัฒนาเว็บด้วยภาษา PHP ลำดับขั้นตอนการพัฒนาเว็บ ข้อมูล ตัวแปร ตัวดำเนินการ การควบคุมคำสั่ง ฟังก์ชัน การจัดการแฟ้มข้อมูล คุกกี้ เซสชัน การรับส่งข้อมูลผ่านฟอร์ม การอัปโหลดไฟล์ การเชื่อมต่อและใช้งานระบบฐานข้อมูล การสร้างเว็บไซต์เพื่อประยุกต์ใช้งานในหน่วยงานหรือภาคธุรกิจ

หน่วยการเรียนรู้วิชา

การพัฒนาเว็บด้วยภาษา PHP รหัสวิชา 3901-2107

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง พุทธศักราช 2557

จำนวน 3 หน่วยกิต 5 ชั่วโมง/สัปดาห์ รวม 85 ชั่วโมง

สัปดาห์ที่	หน่วยที่	หน่วยการสอน	รวม (ชม.)
1	1	ความรู้เบื้องต้นการเขียนโปรแกรมบนอินเทอร์เน็ต	2.5
1	2	การติดตั้งโปรแกรมเพื่อใช้งาน PHP	2.5
2	3	PHP เบื้องต้น	5
3	4	เรื่องของตัวแปรและค่าคงที่	5
4	5	ตัวดำเนินการและนิพจน์ใน PHP	5
5-6	6	คำสั่งตรวจสอบเงื่อนไข และคำสั่งทำซ้ำ	10
7-8	7	การสร้างฟังก์ชันของ PHP	10
9	8	ฟังก์ชันของ PHP ที่จัดเตรียมไว้	5
10	9	การสร้างชุดรับข้อมูล	5
11	10	Session และ Cookie	5
12-13	11	ฐานข้อมูล MySQL	10
14-15	12	การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin	10
16-17	13	การเขียน PHP จัดการ MySQL	10
18		ประเมินผล	

บทที่ 1 ความรู้เบื้องต้นการเขียนโปรแกรมบนอินเทอร์เน็ต	1
1.1 หลักการทำงานของ WWW	2
1.2 ประเภทของโปรแกรมบนเว็บ	4
1.3 รู้จักกับ PHP	6
1.4 การเตรียมการเพื่อเขียนโปรแกรมภาษา PHP	8
แบบฝึกหัดท้ายบทที่ 1	10
บทที่ 2 การติดตั้งโปรแกรมเพื่อใช้งาน PHP	11
2.1 การติดตั้ง PHP Engine	12
2.2 การติดตั้งโปรแกรม Apache Web Server	18
2.3 การปรับแต่ง Apache Web Server	23
2.4 การทดสอบการทำงานของ PHP	27
แบบฝึกหัดท้ายบทที่ 2	29
บทที่ 3 PHP เบื้องต้น	31
3.1 รูปแบบบิล็อกคำสั่งของ PHP	32
3.2 คำสั่งแสดงผลลัพธ์	35
3.3 คำสั่งหมายเหตุ (Comment)	38
3.4 แท็กคำสั่งที่สำคัญของ HTML	39
แบบฝึกหัดท้ายบทที่ 3	47
บทที่ 4 เรื่องของตัวแปรและค่าคงที่	49
4.1 ตัวแปร	50
4.2 การประกาศตัวแปร	51
4.3 ชนิดของข้อมูล	53
4.4 ตัวแปรอาร์เรย์	57
4.5 ฟังก์ชันที่เกี่ยวข้องกับตัวแปร	62
4.6 ค่าคงที่	65
แบบฝึกหัดท้ายบทที่ 4	67

บทที่ 5 ตัวดำเนินการและนิพจน์ใน PHP	69
5.1 ความหมายของตัวดำเนินการและประเภทของตัวดำเนินการ	70
5.2 ตัวดำเนินการทางคณิตศาสตร์	70
5.3 ตัวดำเนินการเปรียบเทียบ	71
5.4 ตัวดำเนินการทางตรรกะ	73
5.5 ตัวดำเนินการสตริง	75
5.6 ตัวดำเนินการระดับบิต	76
5.7 นิพจน์และลำดับการประมวลผลของนิพจน์	77
แบบฝึกหัดท้ายบทที่ 5	81
บทที่ 6 คำสั่งตรวจสอบเงื่อนไข และคำสั่งทำซ้ำ	83
6.1 คำสั่งตรวจสอบเงื่อนไข If	84
6.2 คำสั่งตรวจสอบเงื่อนไข Switch	93
6.3 คำสั่งทำซ้ำ While	95
6.4 คำสั่งทำซ้ำ For	97
6.5 คำสั่งทำซ้ำ Do...while	99
6.6 คำสั่งออกจากการทำงาน	100
แบบฝึกหัดท้ายบทที่ 6	103
บทที่ 7 การสร้างฟังก์ชันของ PHP	105
7.1 การประกาศฟังก์ชัน	106
7.2 การใช้งานฟังก์ชันโดยไม่มีค่าส่งผ่าน	106
7.3 การใช้งานฟังก์ชันโดยมีการส่งผ่านค่า	107
7.4 การคืนค่ากลับของฟังก์ชัน	110
7.5 ประเภทของตัวแปร และขอบเขตของตัวแปร	110
7.6 ประเภทของการส่งผ่านค่า	112
แบบฝึกหัดท้ายบทที่ 7	116
บทที่ 8 ฟังก์ชันของ PHP ที่จัดเตรียมไว้	117
8.1 ฟังก์ชันเกี่ยวกับสตริง	118
8.2 ฟังก์ชันทางคณิตศาสตร์	128
8.3 ฟังก์ชันเกี่ยวกับวันและเวลา	133
แบบฝึกหัดท้ายบทที่ 8	140

บทที่ 9 การสร้างช่องรับข้อมูล	141
9.1 ช่องรับข้อมูลต่างๆ ใน HTML	143
9.2 การส่งข้อมูล	154
9.3 การรวมไฟล์ HTML และ PHP	158
แบบฝึกหัดท้ายบทที่ 9	160
บทที่ 10 Session และ Cookie	161
10.1 Session คืออะไร	162
10.2 ฟังก์ชันที่เกี่ยวข้องกับ Session	163
10.3 ฟังก์ชันที่เกี่ยวข้องกับ Cookie	169
10.4 การประยุกต์ Session และ Cookie ในโปรแกรม Log in	173
แบบฝึกหัดท้ายบทที่ 10	183
บทที่ 11 ฐานข้อมูล MySQL	185
11.1 การติดตั้ง MySQL	
และการปรับแต่งค่าคอนฟิก	186
11.2 การใช้งาน MySQL	191
แบบฝึกหัดท้ายบทที่ 11	201
บทที่ 12 การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin	203
12.1 การติดตั้งโปรแกรม PhpMyAdmin	204
12.2 การใช้งานโปรแกรม PhpMyAdmin	207
แบบฝึกหัดท้ายบทที่ 12	220
บทที่ 13 การเขียน PHP จัดการ MySQL	221
13.1 การเขียน PHP เชื่อมต่อและเปิดการเชื่อมต่อ MySQL	222
13.2 การเขียน PHP เพื่อสร้างฐานข้อมูลและตาราง	224
13.3 การเขียน PHP เพื่อเข้าถึงข้อมูลในตาราง	226
แบบฝึกหัดท้ายบทที่ 13	236
ภาคผนวก กรณีศึกษา ระบบงานอพาร์ทเมนท์	237
บรรณานุกรม	267

ความรู้เบื้องต้นการเขียนโปรแกรมบนอินเทอร์เน็ต

ลักษณะของการเขียนโปรแกรมจะสามารถแบ่งได้เป็น 2 ประเภท คือ การเขียนโปรแกรมบนวินโดวส์ (Windows-Based Application) และการเขียนโปรแกรมบนเว็บ (Web-Based Application) ซึ่งในหนังสือเล่มนี้จะได้กล่าวถึงเฉพาะการเขียนโปรแกรมบนเว็บเท่านั้น โดยลักษณะที่สำคัญก็คือ การแสดงผลของโปรแกรมจะอยู่บนบราวเซอร์ (Browser) ผู้ใช้สามารถเข้าถึงได้ผ่านทางเครือข่ายอินเทอร์เน็ต ไม่ว่าจะอยู่ที่ไหน หรือจะเข้าใช้งานเมื่อไรก็สามารถใช้งานได้ตลอดเวลา ซึ่งเป็นจุดเด่นที่ทำให้โปรแกรมบนเว็บมีเหนือกว่าโปรแกรมบนวินโดวส์ ระบบงานทางธุรกิจก็เริ่มที่จะพัฒนาในลักษณะเขียนโปรแกรมบนเว็บมากขึ้น อันเนื่องมาจากข้อดีดังกล่าวนั่นเอง เนื้อหาใหม่ที่จะกล่าวถึงความรู้พื้นฐานในการเขียนโปรแกรมบนอินเทอร์เน็ต เพื่อให้ผู้อ่านได้เข้าใจลักษณะการทำงานของโปรแกรมก่อนที่จะเริ่มเข้าสู่เนื้อหาของการเขียนโปรแกรม

จุดประสงค์ทั่วไป

1. เพื่อศึกษาหลักการการทำงานของ WWW
 2. เพื่อให้ทราบถึงประโยชน์ของการทำงานแบบ Client-Server
 3. เพื่อให้ทราบถึงกาประยุกต์หลักการการทำงานแบบ Client-Server กับงานต่างๆ ในสำนักงานได้
 4. เพื่อให้ทราบและแยกความแตกต่างระหว่างประเภทโปรแกรมแบบ Static Programming และ Dynamic Programming
- เพื่อให้รู้จักโปรแกรมภาษา PHP
5. เพื่อให้ทราบสิ่งที่จะต้องเตรียมการก่อนที่จะเขียนโปรแกรมภาษา PHP

จุดประสงค์เชิงพฤติกรรม

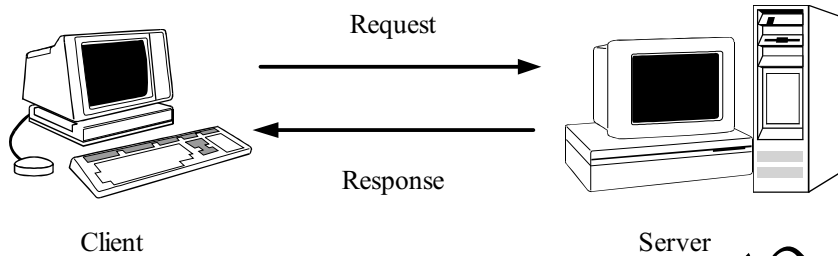
1. บอกหลักการการทำงานของ WWW
2. บอกความแตกต่างของประเภทของโปรแกรมบนเว็บ
3. เข้าใจหลักการของภาษา PHP

เนื้อหาสาระ

1. หลักการทำงานของ WWW
2. ประเภทของโปรแกรมบนเว็บ
3. รู้จักกับ PHP
4. การเตรียมการเพื่อเขียนโปรแกรมภาษา PHP

1.1 หลักการทำงานของ WWW

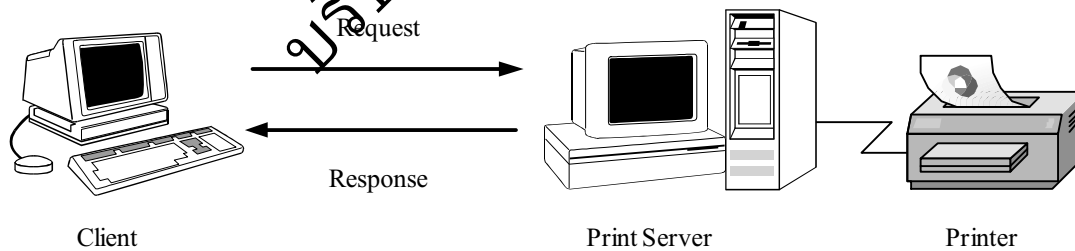
การทำงานของ World Wide Web หรือ WWW จะมีลักษณะเช่นเดียวกับการทำงานในลักษณะไคลเอ็นต์-เซิร์ฟเวอร์ (Client-Server) คือ มีลักษณะของการเชื่อมต่อของเครื่องผู้ให้บริการ (Server) และเครื่องผู้ใช้บริการ (Client) พิจารณารูปต่อไปนี้



รูปที่ 1.1 แสดงการทำงานของ Client-Server

จากรูปที่ 1.1 นี้การทำงานจะเริ่มจากเครื่องผู้ขอใช้บริการ (Client) ซึ่งอาจจะเรียกว่าเป็นเครื่องลูกที่ทำกรร้องขอ (Request) ใช้บริการจากเครื่องผู้ให้บริการ (Server) ซึ่งอาจจะเรียกว่าเป็นเครื่องแม่ หลังจากเครื่องผู้ให้บริการทำการจัดเตรียมข้อมูลหรือบริการตามที่เครื่องผู้ขอใช้บริการได้ร้องขอมาก็จะทำการตอบกลับ (Response) คืนไปยังเครื่องของผู้ขอใช้บริการ โดยปกติเครื่องแม่จะมีอยู่เพียงเครื่องเดียว ในขณะที่เครื่องลูกอาจจะมีได้หลายๆ เครื่อง และเครื่องลูกหลายๆ เครื่องนี้ก็อาจจะเข้ามาขอใช้บริการจากเครื่องแม่พร้อมกันก็ได้

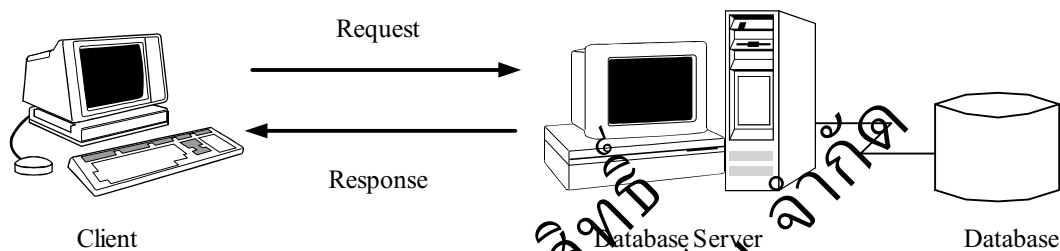
ในสำนักงานต่างๆ ได้นำประโยชน์ของการทำงานในลักษณะ Client-Server นี้ไปใช้จัดการงานต่างๆ ภายในสำนักงาน พิจารณารูปต่อไปนี้



รูปที่ 1.2 แสดงการทำงานของ Client-Server ในลักษณะของการให้บริการงานพิมพ์

จากรูปที่ 1.2 เครื่องแม่ได้ทำการเชื่อมต่อกับเครื่องพิมพ์ (Printer) ดังนั้นเครื่องแม่นี้เราจะเรียกว่า Print Server คือเป็นเครื่องที่ให้บริการงานทางด้านเครื่องพิมพ์ กระบวนการทำงานจะเริ่มต้นจากเครื่องลูกที่ต้องการพิมพ์เอกสารจึงได้ทำการสั่งพิมพ์จากเครื่องลูกดังกล่าว ข้อมูลที่ต้องการพิมพ์ก็จะถูกส่งมายังเครื่อง Print Server ซึ่งก็จะทำการพิมพ์ข้อมูลออกทางเครื่องพิมพ์ ข้อดีของการเชื่อมต่อในลักษณะนี้ก็คือน ในสำนักงานนั้นก็เพียงแต่จัดหาเครื่องพิมพ์ 1 เครื่องเชื่อมต่อกับเครื่องคอมพิวเตอร์ที่จะใช้เป็น Print Server เมื่อพนักงานภายในบริษัทต้องการสั่งพิมพ์เอกสารใดๆ ก็สามารส่งงานที่เครื่องลูกได้เลย ประหยัดงบประมาณที่จะต้องทำการจัดหาเครื่องพิมพ์ให้กับพนักงานทุกๆ คน

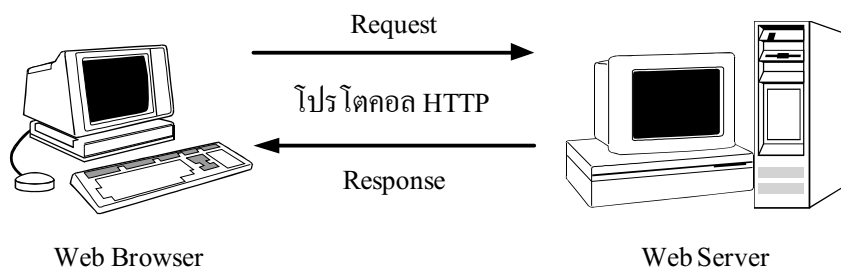
พิจารณาตัวอย่างของการทำงานแบบ Client-Server อีกตัวอย่าง ดังรูปต่อไปนี้



รูปที่ 1.3 แสดงการทำงานของ Client-Server ในลักษณะของการให้บริการฐานข้อมูล

จากรูปที่ 1.3 ก็เป็นอีกตัวอย่างของการทำงานในลักษณะ Client-Server โดยเครื่องแม่ทำหน้าที่จัดเก็บข้อมูลต่างๆ ทั้งหมดของบริษัทไว้ที่ฐานข้อมูล ดังนั้นเครื่องแม่นี้เราก็จะเรียกว่า Database Server การทำงานก็จะเริ่มจากเครื่องลูกเช่นเดียวกัน คือ เครื่องลูกอาจจะทำการเพิ่ม ลบ แก้ไข หรือดึงข้อมูลซึ่งจัดเก็บไว้ที่ฐานข้อมูล ข้อดีของการเชื่อมต่อแบบนี้ก็คือ การจัดการข้อมูลต่างๆ ซึ่งถูกจัดเก็บไว้เพียงที่เดียวทำให้ข้อมูลมีความเป็นหนึ่งเดียว (Uniqueness) คือข้อมูลไม่กระจัดกระจายและมีความถูกต้อง

จากทั้ง 2 ตัวอย่างของการทำงานของ Client-Server ลำดับถัดไปจะกล่าวถึงลักษณะของการทำงานของ WWW ซึ่งก็จะมีลักษณะการทำงานเหมือนกับ 2 ตัวอย่างที่ได้อธิบายไปแล้ว พิจารณารูปต่อไปนี้



รูปที่ 1.4 แสดงการทำงานของ WWW

จากรูปที่ 1.4 เครื่องแม่ในที่นี้จะให้บริการต่างๆ ที่เกี่ยวกับเว็บทั้งหมด ไฟล์เว็บเพจ รูปภาพ หรือ โปรแกรมบนเว็บต่างๆ จะถูกจัดเก็บไว้ในเครื่องนี้ ซึ่งเราจะเรียกเครื่องนี้ว่า Web Server ส่วนเครื่องลูกซึ่งเป็นเครื่องที่ขอใช้บริการเราจะเรียกว่า Client เนื่องจากการแสดงผลจะถูกแสดงบนบราวเซอร์ เช่น Internet Explorer หรือ Firefox เป็นต้น กระบวนการทำงานจะเริ่มจาก Web Browser จะทำการร้องขอหน้าเว็บใดๆ โดยการพิมพ์ URL (Universal Resource Locator) จากโปรแกรมบราวเซอร์ ซึ่งข้อมูลจะถูกกระทำผ่านโปรโตคอล HTTP (Hypertext Transfer Protocol) เครื่อง Web Server เมื่อได้รับคำร้องขอก็จะทำการจัดส่งหน้าเว็บเพจนั้นผ่านทาง Web Browser

1.2 ประเภทของโปรแกรมบนเว็บ

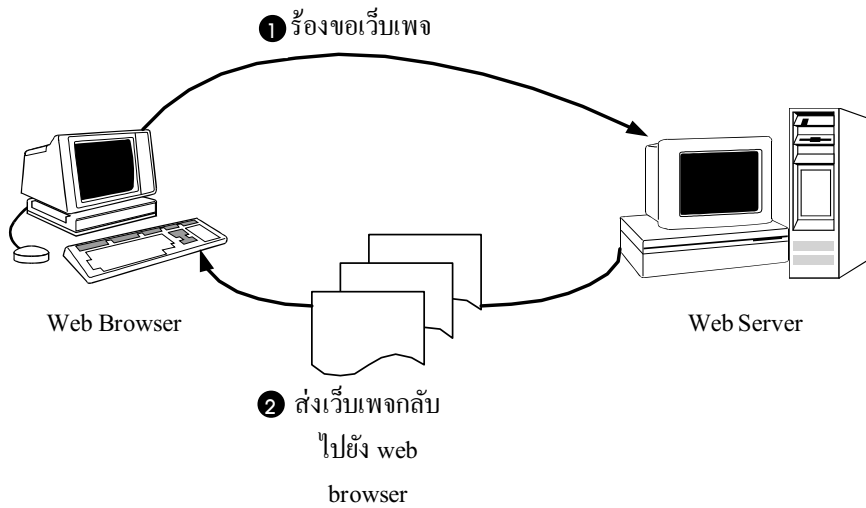
ในการเขียนโปรแกรมบนเว็บสามารถที่จะแบ่งลักษณะการทำงานของโปรแกรมได้เป็น 2 ประเภท คือ

1. Static Programming
2. Dynamic Programming

• **Static Programming** เป็นลักษณะของโปรแกรมบนเว็บที่ไม่ค่อยมีการเปลี่ยนแปลง เช่น เว็บประวัติส่วนตัว, เว็บนำเสนอประวัติและโครงสร้างขององค์กร เป็นต้น ลักษณะเว็บประเภทนี้เมื่อผู้พัฒนาเว็บสร้างเว็บขึ้นมาแล้วหากต้องการที่จะทำการแก้ไขข้อมูลบนเว็บนั้นก็ต้องใช้โปรแกรมในการสร้างเว็บเพจ เช่น Adobe Dreamweaver, Microsoft FrontPage เป็นต้น เปิดไฟล์ของหน้าเว็บนั้นแล้วจึงทำการแก้ไขข้อมูลที่ต้องการ เมื่อเสร็จแล้วก็ต้องทำการบันทึกไฟล์ดังกล่าว และทำการอัปโหลดไฟล์ขึ้นไปเก็บไว้ที่ Web Server จะเห็นว่าลักษณะของโปรแกรมบนเว็บประเภทนี้จะไม่มีความยืดหยุ่นในการจัดการ และสร้างความยุ่งยากให้แก่ผู้พัฒนาเว็บเพจด้วย

ในการพัฒนาโปรแกรมบนเว็บประเภทนี้ ผู้พัฒนาโปรแกรมเพียงแค่จัดหาเครื่องคอมพิวเตอร์และทำการติดตั้งซอฟต์แวร์ที่จะทำให้เครื่องคอมพิวเตอร์เครื่องนี้เป็นเครื่องแม่ (Web Server) เท่านั้น และในการพัฒนาโปรแกรมบนเว็บ ผู้พัฒนาที่ไม่ต้องมีความรู้ในการเขียนโปรแกรมก็ได้ เพียงแค่สามารถเขียนเว็บเพจโดยใช้โปรแกรมที่ใช้สร้างเว็บเพจก็เพียงพอแล้ว

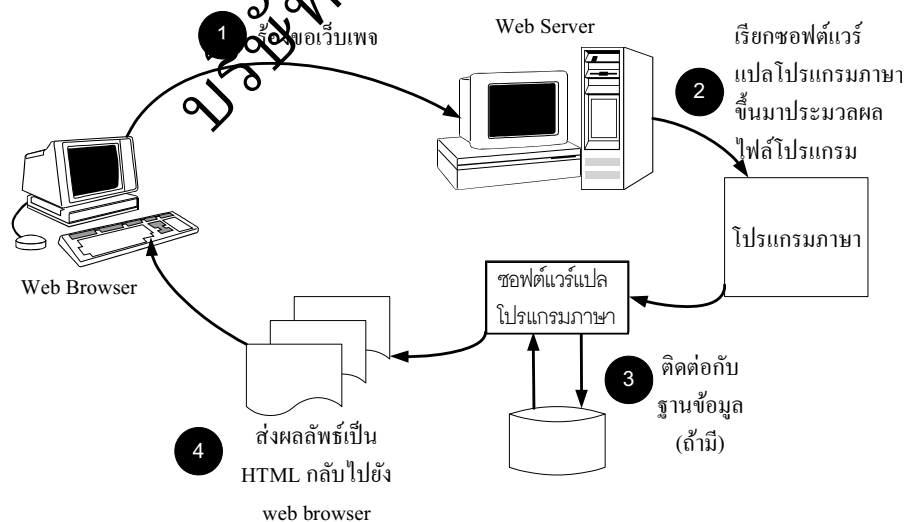
ลักษณะการทำงานของโปรแกรมบนเว็บประเภทนี้มีลักษณะดังรูปต่อไปนี้



รูปที่ 1.5 แสดงการทำงานของ Static Programming

• **Dynamic Programming** เป็นลักษณะของโปรแกรมบนเว็บที่เกิดขึ้นเพื่อแก้ปัญหาความไม่ยืดหยุ่นในการจัดการข้อมูลของ Static Programming โดยมุ่งมาสำหรับเว็บที่ต้องมีการเปลี่ยนแปลงข้อมูลอยู่บ่อยครั้งหรือเว็บที่มีการจัดเก็บข้อมูลจำนวนมากไว้ภายในเซิร์ฟเวอร์ข้อมูล เช่น เว็บหนังสือพิมพ์, เว็บแสดงรายละเอียดของสินค้า เป็นต้น

ในการพัฒนาโปรแกรมบนเว็บประเภทนี้จะต้องอาศัยผู้พัฒนาโปรแกรมที่มีความรู้ ความสามารถในการเขียนโปรแกรม นอกจากนี้ในส่วนของซอฟต์แวร์ที่ต้องติดตั้งประกอบไปด้วยซอฟต์แวร์ที่ใช้ในการทำให้เครื่องคอมพิวเตอร์มีความสามารถเป็นเครื่อง Web Server, ซอฟต์แวร์ที่ใช้ในการแปลโปรแกรมภาษาและฐานข้อมูล พิจารณาลักษณะการทำงานของโปรแกรมบนเว็บประเภทนี้ ดังรูปต่อไปนี้



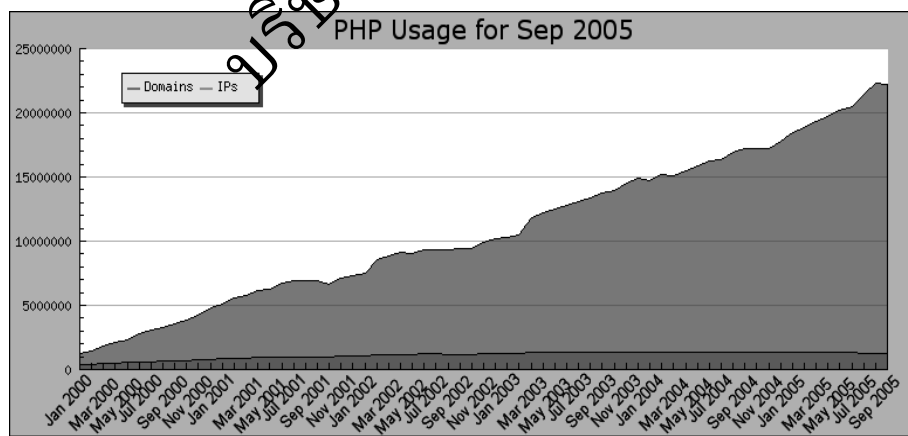
รูปที่ 1.6 แสดงการทำงานของ Dynamic Programming

จากรูปที่ 1.6 โปรแกรมก็จะเริ่มกระบวนการทำงานจากเครื่องลูกทำการร้องขอเว็บเพจ ซึ่งเว็บเพจที่ร้องขอได้มีการเขียนโปรแกรมบนเว็บในลักษณะ Dynamic Programming ได้ ดังนั้นที่เครื่องแม่ก็จะทำการเรียกซอฟต์แวร์แปลโปรแกรมภาษาขึ้นมาเพื่อแปลโปรแกรมภาษา (Programming Language) ให้เป็นภาษาเครื่อง (Machine Language) เนื่องจากเครื่องคอมพิวเตอร์จะเข้าใจเฉพาะภาษาเครื่องเท่านั้น ถ้าโปรแกรมที่เขียนขึ้นมีคำสั่งที่ทำการจัดการฐานข้อมูลก็จะทำการ เพิ่ม ลบ แก้ไข หรือดึงข้อมูลจากฐานข้อมูล ซึ่งโดยปกติตัวฐานข้อมูลนี้อาจจะอยู่ภายในเครื่องแม่ หรืออาจจะแยกออกมาเป็นเครื่องคอมพิวเตอร์อีกเครื่องก็ได้ หลังจากนั้นขั้นตอนสุดท้ายหลังจากเครื่องแม่ทำการประมวลผลเสร็จเรียบร้อยแล้วก็จะทำการส่งข้อมูลผลลัพธ์กลับในลักษณะ HTML กลับไปยังเครื่องลูก หรือเครื่องที่ทำการร้องขอนั่นเอง

1.3 รู้จักกับ PHP

ในปัจจุบันการเขียนโปรแกรมบนอินเทอร์เน็ตมีโปรแกรมภาษาให้เลือกจำนวนมาก อาทิเช่น PHP, ASP, JSP, Java เป็นต้น แต่สำหรับในที่นี้จะเลือกใช้โปรแกรมภาษา PHP ซึ่งย่อมาจากคำว่า Personal Home Page ข้อดีของภาษา PHP มีด้วยกันหลายข้อ คือ

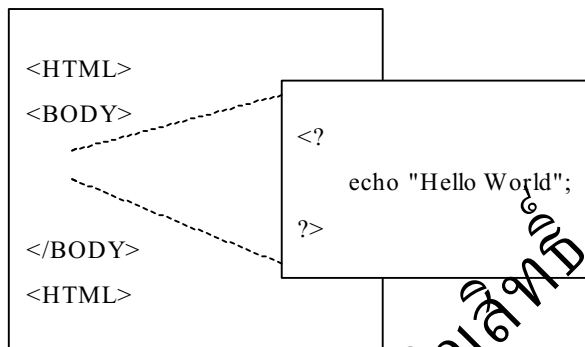
- PHP เป็นโปรแกรมภาษาที่สามารถดาวน์โหลดได้ฟรีไม่มีปัญหาเรื่องลิขสิทธิ์
- PHP มีการแปลภาษาและทำการประมวลผลได้อย่างรวดเร็ว
- PHP สามารถทำงานได้ทั้งบนระบบปฏิบัติการ Windows, Unix, Linux และ Macintosh
- สามารถกล่าวได้ว่า PHP เป็นโปรแกรมภาษาที่พัฒนาระบบงานบนเว็บที่ได้รับความนิยมมากที่สุดในปัจจุบัน
- ฯลฯ



รูปที่ 1.7 กราฟแสดงปริมาณการใช้ PHP ตั้งแต่ มกราคมปี 2000 ถึง กันยายน 2005
(<http://th2.php.net/usage.php>)

PHP เป็นโปรแกรมภาษาที่ทำงานในลักษณะ Server-side scripting language คือมีลักษณะของการประมวลผลที่ฝั่ง Server หลังจากนั้นก็จะทำการส่งผลลัพธ์ในรูปแบบ HTML กลับไปยัง Web browser ซึ่งลักษณะเช่นนี้จะทำให้เราสามารถใช้งานภาษา PHP พัฒนาระบบงานในลักษณะ Dynamic Programming ได้

ไฟล์ PHP จะเหมือนกับเอกสาร HTML ทั่วไป เนื่องจากเราสามารถเขียนแท็กซึ่งเป็นคำสั่งภาษา PHP ลงไปในแท็กของ HTML ได้ ไฟล์นามสกุลของ PHP คือ .php ตัวอย่างต่อไปนี้จะแสดงถึงลักษณะของการเขียนโปรแกรมภาษา PHP โดยทำการเขียนคำสั่ง PHP ให้ทำการพิมพ์ข้อความว่า Hello World และทำการแทรก PHP ไว้ในไฟล์ HTML



รูปที่ 1.8 แสดงลักษณะของการเขียนโปรแกรมภาษา PHP แทรกไว้ใน HTML

ในทางกลับกันเราก็สามารถเขียนโปรแกรมภาษา HTML แทรกไว้ในภาษา PHP ได้เช่นกัน ดังตัวอย่างต่อไปนี้จะทำการพิมพ์ข้อความว่า Hello World เป็นตัวหนังสือสีแดงโดยใช้แท็กของ HTML เพื่อแสดงสีของตัวอักษร

```
<?
echo "<HTML><BODY>";
echo "<Font Color=Red>";
echo "Hello World";
echo "</Font>";
echo "</BODY></HTML>";
?>
```

รูปที่ 1.9 แสดงลักษณะของการเขียนแท็กภาษา HTML แทรกไว้ในคำสั่ง PHP

1.4 การเตรียมการเพื่อเขียนโปรแกรมภาษา PHP

ในการเขียนโปรแกรมบนอินเทอร์เน็ตโดยใช้ภาษา PHP สิ่งที่คุณจำเป็นต้องมีประกอบไปด้วย

- เครื่องคอมพิวเตอร์อย่างน้อย 1 เครื่อง โดยปกติการพัฒนาระบบงานบนเว็บจะต้องมีเครื่องคอมพิวเตอร์ที่ทำเป็น Web Server และ Web Browser แต่ถ้ามีงบประมาณจำกัดในการจัดหาเครื่องคอมพิวเตอร์ก็สามารถใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียวก็ได้ โดยให้เครื่องคอมพิวเตอร์นี้ทำหน้าที่เป็นทั้ง Web Server และ Web Browser ในตัวเดียวกัน
- ระบบปฏิบัติการ Windows หรือ Unix หรือ Linux แต่ในที่นี้จะกำหนดให้ใช้ระบบปฏิบัติการ Windows XP
- โปรแกรม Web Server ซึ่งสามารถเลือกใช้โปรแกรมอะไรก็ได้ เช่น Apache, Internet Information Server (IIS), Personal Web Server (PWS), OmniHTTPd เป็นต้น แต่ในที่นี้จะเลือกใช้โปรแกรม Apache Web Server
- PHP Engine หรือตัวแปลภาษา PHP ซึ่งเป็นโปรแกรมที่จะทำการแปลภาษา PHP เป็นภาษาเครื่องเพื่อให้คอมพิวเตอร์สามารถทำงานและประมวลผลคำสั่งที่ต้องการ
- โปรแกรมฐานข้อมูล ซึ่งโปรแกรมภาษา PHP สามารถเลือกใช้ฐานข้อมูลได้หลายประเภท เช่น dBase, IMAP, MySQL, MS Access, Oracle เป็นต้น แต่ในที่นี้จะเลือก MySQL เป็นฐานข้อมูล
- PhpMyadmin ซึ่งเป็นโปรแกรมที่ใช้จัดการฐานข้อมูล MySQL โดยเฉพาะ
- โปรแกรม Web Authoring และ Editor คือ โปรแกรมที่ใช้ในการสร้างเว็บเพจ เช่น Adobe Dreamweaver, Microsoft Frontpage, Editplus เป็นต้น

สรุป

การทำงานของ WWW จะมีลักษณะเช่นเดียวกับการทำงานในลักษณะไคลเอ็นต์-เซิร์ฟเวอร์ (Client-Server) คือมีลักษณะของการเชื่อมต่อของเครื่องผู้ให้บริการ (Server) และเครื่องผู้ใช้บริการ (Client) ในสำนักงานต่างๆ ได้นำประโยชน์ของการทำงานในลักษณะ Client-Server นี้ไปใช้จัดการงานต่างๆ ภายในสำนักงาน เช่น Client-Server ในลักษณะของ File Server, Print Server และ Web Server

ในการเขียนโปรแกรมบนเว็บสามารถที่จะแบ่งลักษณะการทำงานของโปรแกรมได้เป็น 2 ประเภท คือ Static Programming, Dynamic Programming โดย Static Programming เป็นลักษณะของโปรแกรมบนเว็บที่ไม่ค่อยมีการเปลี่ยนแปลง เช่น เว็บประวัติส่วนตัว, เว็บนำเสนอประวัติและโครงสร้างขององค์กร เป็นต้น ส่วน Dynamic Programming เป็นลักษณะของโปรแกรมบนเว็บที่เกิดขึ้นมาเพื่อแก้ปัญหาความไม่ยืดหยุ่นในการจัดการข้อมูลของ Static Programming โดยเหมาะสำหรับเว็บที่ต้องมีการ



เปลี่ยนแปลงข้อมูลอยู่บ่อยครั้งหรือเว็บที่มีการจัดเก็บข้อมูลจำนวนมากไว้ภายในฐานข้อมูล เช่น เว็บหนังสือพิมพ์, เว็บแสดงรายละเอียดของสินค้า เป็นต้น

โปรแกรมภาษา PHP ซึ่งย่อมาจากคำว่า Personal Home Page ข้อดีของภาษา PHP มีด้วยกันหลายข้อ เช่น เป็นโปรแกรมภาษาที่สามารถดาวน์โหลดได้ฟรีไม่มีปัญหาเรื่องลิขสิทธิ์, มีการแปลภาษาและทำการประมวลผลได้อย่างรวดเร็ว, สามารถทำงานได้ทั้งบนระบบปฏิบัติการ Windows, Unix, Linux และ Macintosh เป็นต้น ก่อนที่จะเริ่มลงมือเขียนโปรแกรมภาษา PHP จะต้องเตรียมการต่างๆ คือ เครื่องคอมพิวเตอร์อย่างน้อย 1 เครื่อง, ระบบปฏิบัติการ, โปรแกรม Web Server, PHP Engine, โปรแกรมฐานข้อมูล, PhpMyAdmin, โปรแกรม Web Authoring และ Editor

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

แบบฝึกหัดบทที่ 1

1. จงอธิบายลักษณะการทำงานของ Client-Server รวมทั้งยกตัวอย่างงานภายในสำนักงานที่นำหลักการ Client-Server ไปประยุกต์ใช้ในการทำงาน

.....

.....

.....

.....

2. ให้อธิบายความแตกต่างระหว่าง Static Programming และ Dynamic Programming

.....

.....

.....

.....

3. ทำไมจึงกล่าวว่า "PHP เป็นโปรแกรมภาษาที่ทำงานในลักษณะ Server Side Scripting Language"

.....

.....

.....

.....

4. ข้อดีของโปรแกรมภาษา PHP คืออะไร

.....

.....

.....

.....

5. ก่อนที่จะลงมือเขียนโปรแกรมภาษา PHP จะต้องเตรียมการอะไรบ้าง จงอธิบาย

.....

.....

.....

.....

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

การติดตั้งโปรแกรมเพื่อใช้งาน PHP

ก่อนที่จะเราจะพัฒนาระบบงานบนอินเทอร์เน็ตด้วยภาษา PHP ได้ก็จะต้องทำการติดตั้งโปรแกรมบนเครื่องคอมพิวเตอร์ และทำการจัดเตรียมโปรแกรมเครื่องมือต่างๆ ที่จะช่วยในการพัฒนาระบบงาน นอกจากนี้ในการติดตั้งโปรแกรมต่างๆ อาจจะต้องมีการปรับแต่งค่าบางอย่างของโปรแกรมเพื่อให้การใช้งานมีความถูกต้อง และใช้งานได้สะดวกยิ่งขึ้น เนื้อหาในบทนี้จะทำการติดตั้งโปรแกรม 2 โปรแกรมด้วยกัน คือ PHP Engine และ Apache Web Server ส่วนโปรแกรมที่เหลือ คือ MySQL และ Phpmyadmin จะขอกล่าวถึงในส่วนถัดไป เนื่องจากเนื้อหาในช่วงต้นนี้จะช่วยอธิบายคำสั่งพื้นฐานที่สำคัญของ PHP

จุดประสงค์ทั่วไป

1. เพื่อเตรียมความพร้อมของโปรแกรมก่อนที่จะเริ่มเขียนโปรแกรมภาษา PHP
2. เพื่อแนะนำให้ผู้รู้กับโปรแกรม PHP Engine และ Apache Web Server
3. เพื่อให้ทราบขั้นตอนการติดตั้ง PHP Engine
4. เพื่อให้ทราบวิธีการปรับแต่ง PHP Engine
5. เพื่อให้ทราบขั้นตอนการติดตั้ง Apache Web Server
6. เพื่อให้ทราบวิธีการปรับแต่ง Apache Web Server
7. เพื่อให้ทราบวิธีการทดสอบการใช้งานของโปรแกรมภาษา PHP หลังจากทำการติดตั้ง PHP Engine และ Apache Web Server เสร็จเรียบร้อยแล้ว

จุดประสงค์เชิงปฏิบัติการ

1. สามารถติดตั้งโปรแกรม PHP Engine
2. สามารถปรับแต่งโปรแกรม PHP Engine
3. สามารถติดตั้งโปรแกรม Apache Web Server
4. สามารถปรับแต่งโปรแกรม Apache Web Server

เนื้อหาสาระ

1. การติดตั้ง PHP Engine
2. การติดตั้ง Apache Web Server
3. การปรับแต่ง Apache Web Server
4. การทดสอบการทำงานของ PHP

2.1 การติดตั้ง PHP Engine

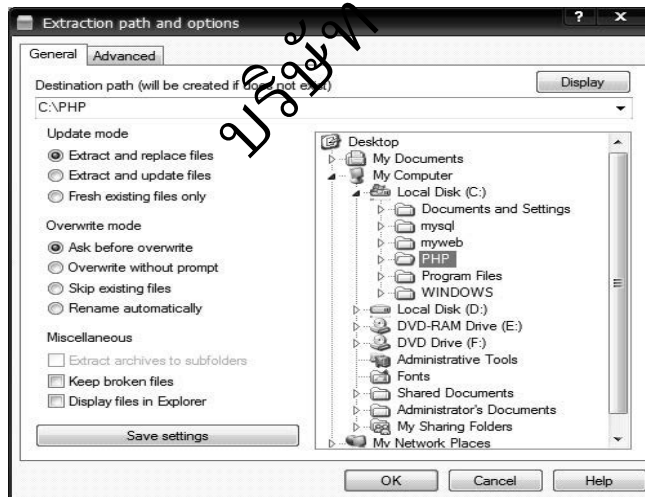
โปรแกรม PHP Engine เป็นโปรแกรมที่ใช้ในการแปลภาษา PHP ให้เป็นภาษาเครื่อง ขั้นตอนในการติดตั้ง PHP Engine ประกอบไปด้วยขั้นตอน ดังนี้

1. Download โปรแกรมติดตั้งที่เว็บ <http://www.php.net/downloads.php> โดยเลือกดาวน์โหลดไฟล์ที่เป็น Zip



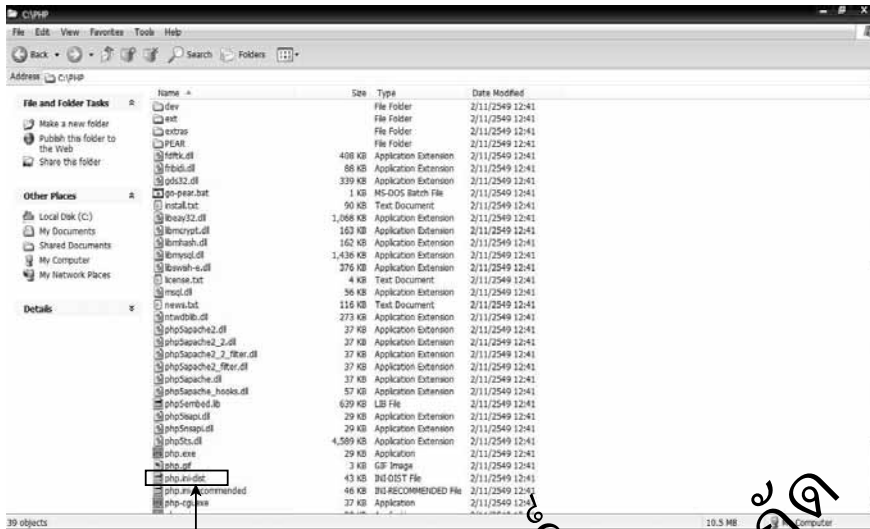
รูปที่ 2.1 เว็บไซต์ดาวน์โหลดโปรแกรม PHP Engine

2. สร้างโฟลเดอร์ PHP ในไดรฟ์ C: ซึ่งจากนั้นให้ทำการแตกไฟล์ Zip นี้ไปที่ C:\PHP



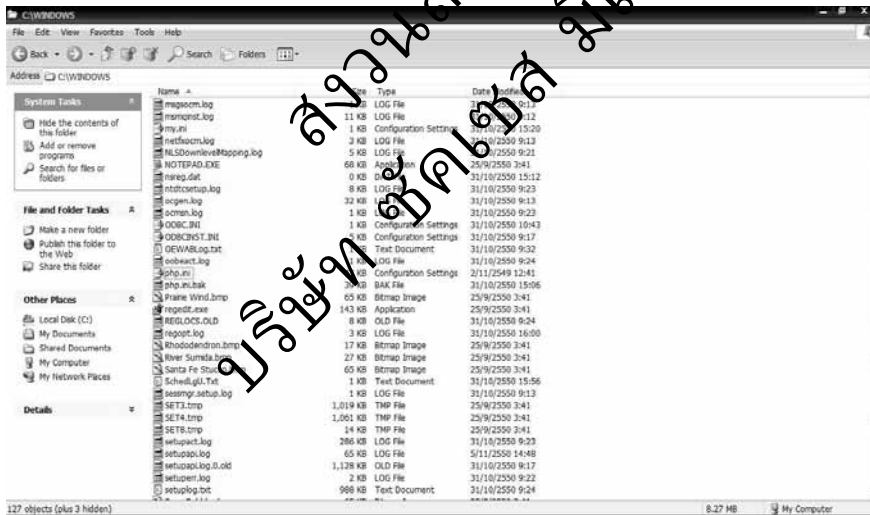
รูปที่ 2.2 หน้าจอแสดงการแตกไฟล์ Zip ไปไว้ที่ C:\PHP

3. หาไฟล์ชื่อ php.ini-dist ในโฟลเดอร์ PHP เปลี่ยนชื่อเป็น php.ini



รูปที่ 2.3 หน้าจอแสดงชื่อไฟล์ php.ini-dist ภายใน C:\PHP

4. Copy ไฟล์ php.ini ที่เปลี่ยนชื่อจากข้อที่ 3. มาใส่ที่ C:\WINDOWS



รูปที่ 2.4 หน้าจอแสดงการ Copy ไฟล์ php.ini มาไว้ที่ C:\Windows

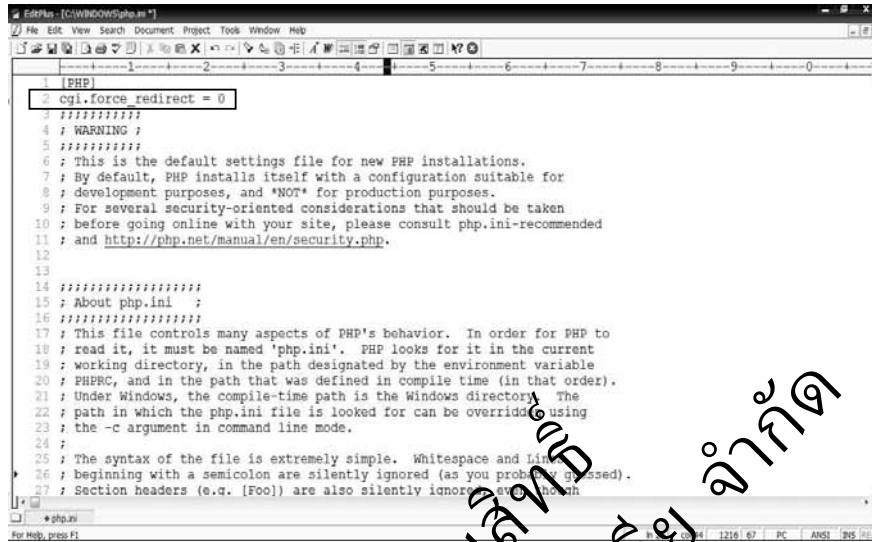
หลังจากทำการติดตั้งโปรแกรม PHP Engine เรียบร้อยแล้ว ขั้นตอนต่อไป คือ การปรับแต่งโปรแกรม เพื่อให้พร้อมใช้งาน

บทที่

2

5. เปิดไฟล์ php.ini ด้วย Text Editor เพื่อปรับแต่งค่าต่างๆ ดังนี้

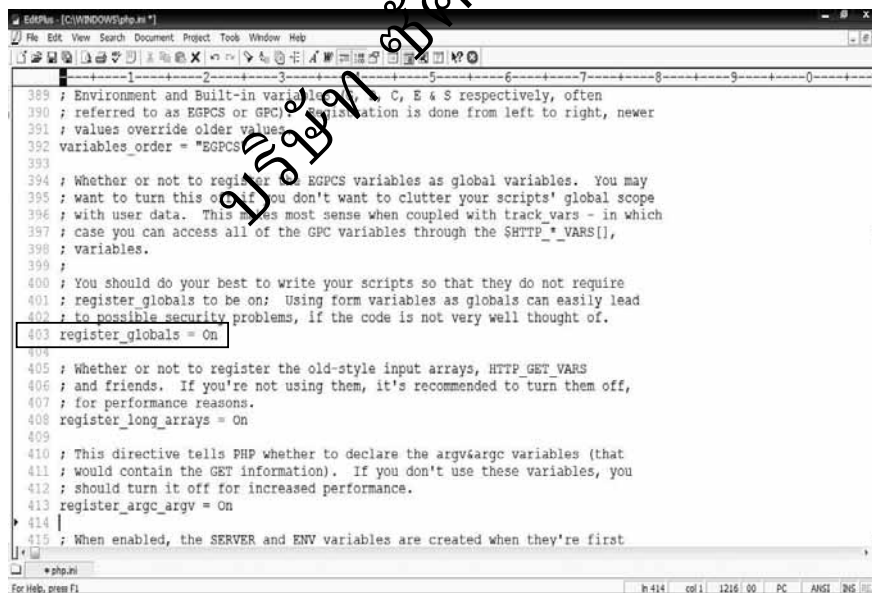
5.1) เพิ่มบรรทัด `cgi.force_redirect = 0` ไว้ต่อบรรทัด `[PHP]` ซึ่งเป็นการกำหนดให้ PHP สามารถทำการ redirect ไฟล์ได้



```
[PHP]
cgi.force_redirect = 0
;
; WARNING ;
;
; This is the default settings file for new PHP installations.
; By default, PHP installs itself with a configuration suitable for
; development purposes, and *NOT* for production purposes.
; For several security-oriented considerations that should be taken
; before going online with your site, please consult php.ini-recommended
; and http://php.net/manual/en/security.php.
;
; About php.ini ;
;
; This file controls many aspects of PHP's behavior. In order for PHP to
; read it, it must be named 'php.ini'. PHP looks for it in the current
; working directory, in the path designated by the environment variable
; PHP_INI_C, and in the path that was defined in compile time (in that order).
; Under Windows, the compile-time path is the Windows directory. The
; path in which the php.ini file is looked for can be overridden using
; the -c argument in command line mode.
;
; The syntax of the file is extremely simple. Whitespace and lines
; beginning with a semicolon are silently ignored (as you probably guessed).
; Section headers (e.g. [Foo]) are also silently ignored, even though
```

รูปที่ 2.5 หน้าจอแสดงการเพิ่มบรรทัด `cgi.force_redirect = 0`

5.2) ทาบรรทัดที่เขียนว่า `register_globals = Off` แล้วเปลี่ยนเป็น `register_globals = On` ซึ่งเป็นการกำหนดให้สามารถใช้ตัวแปรในระบอบ `global` ในการเรียกใช้ตัวแปรข้ามไฟล์



```
389 ; Environment and Built-in variables C, E & S respectively, often
390 ; referred to as EGPCS or GPC). Initialization is done from left to right, newer
391 ; values override older values
392 variables_order = "EGPCS"
393
394 ; Whether or not to register the EGPCS variables as global variables. You may
395 ; want to turn this off if you don't want to clutter your scripts' global scope
396 ; with user data. This makes most sense when coupled with track_vars - in which
397 ; case you can access all of the GPC variables through the $HTTP_*_VARS[]
398 ; variables.
399 ;
400 ; You should do your best to write your scripts so that they do not require
401 ; register_globals to be on: Using form variables as globals can easily lead
402 ; to possible security problems, if the code is not very well thought of.
403 register_globals = On
404
405 ; Whether or not to register the old-style input arrays, HTTP_GET_VARS
406 ; and friends. If you're not using them, it's recommended to turn them off,
407 ; for performance reasons.
408 register_long_arrays = On
409
410 ; This directive tells PHP whether to declare the argv&argc variables (that
411 ; would contain the GET information). If you don't use these variables, you
412 ; should turn it off for increased performance.
413 register_argv = On
414
415 ; When enabled, the SERVER and ENV variables are created when they're first
```

รูปที่ 2.6 หน้าจอแสดงการปรับแต่งค่า `register_globals` ให้เป็น `On`

5.3) ทหารรทัดที่เขียนว่า :error_reporting = E_ALL & ~E_NOTICE หลังจากนั้นให้ทำการลบเครื่องหมาย ; ข้างหน้าออก ซึ่งเป็นคำสั่งให้มีการพิมพ์ error ตอนที่แสดงผลด้วย

```

282 ; E_COMPILE_ERROR - fatal compile-time errors
283 ; E_COMPILE_WARNING - compile-time warnings (non-fatal errors)
284 ; E_USER_ERROR - user-generated error message
285 ; E_USER_WARNING - user-generated warning message
286 ; E_USER_NOTICE - user-generated notice message
287 ;
288 ; Examples:
289 ;
290 ; - Show all errors, except for notices and coding standards warnings
291 ;
292 error_reporting = E_ALL & ~E_NOTICE
293 ;
294 ; - Show all errors, except for notices
295 ;
296 ;error_reporting = E_ALL & ~E_NOTICE | E_STRICT
297 ;
298 ; - Show only errors
299 ;
300 ;error_reporting = E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR
301 ;
302 ; - Show all errors except for notices and coding standards warnings
303 ;
304 error_reporting = E_ALL & ~E_NOTICE
305 ;
306 ; Print out errors (as a part of the output). For production web sites,
307 ; you're strongly encouraged to turn this feature off, and use error logging
308 ; instead (see below). Keeping display errors enabled on a production web site

```

รูปที่ 2.7 หน้าจอแสดงการลบเครื่องหมาย ; ด้านหน้า error_reporting = E_ALL & ~E_NOTICE

5.4) ทหารรทัดที่เขียนว่า :default_charset = "iso-8859-1" แล้วเปลี่ยนจาก "iso-8859-1" เป็น "TIS-620" และเอาเครื่องหมาย ; ข้างหน้าออกด้วย ซึ่งเป็นนกที่กำหนดให้ PHP สามารถใช้งานภาษาไทยได้

```

435 magic_quotes_sybase = Off
436 ;
437 ; Automatically add files before or after any PHP document.
438 auto_prepend_file =
439 auto_append_file =
440 ;
441 ; As of 4.0b4, PHP always outputs character encoding by default in
442 ; the Content-type: header. Multiple sending of the charset, simply
443 ; set it to be empty.
444 ;
445 ; PHP's built-in default character encoding is 'UTF-8'.
446 ; default_mimetype = "text/html"
447 default_charset = "TIS-620"
448 ;
449 ; Always populate the $HTTP_RAW_POST_DATA variable.
450 ;always_populate_raw_post_data = On
451 ;
452 ;
453 ;;;;;;;;;;;;;;;;;;;;;;;;;
454 ; Paths and Directories ;
455 ;;;;;;;;;;;;;;;;;;;;;;;;;
456 ;
457 ; UNIX: "/path1:/path2"
458 ;include_path = "./php/includes"
459 ;
460 ; Windows: "\path1;\path2"
461 ;include_path = ".:c:\php\includes"

```

รูปที่ 2.8 หน้าจอแสดงการปรับแต่งการแสดงผลภาษาไทย

5.5) ทหารรทัดที่เขียนว่า extension_dir = "./" แล้วทำการแก้ไขเป็น extension_dir = "C:\PHP\ext" ซึ่งเป็นกรกำหนดโฟลเดอร์ที่เก็บไฟล์ส่วนขยายความสามารถของ PHP (extension)

```

464 ; if PHP was not compiled with FORCE_REDIRECT, you SHOULD set doc root
465 ; if you are running php as a CGI under any web server (other than IIS)
466 ; see documentation for security issues. The alternate is to use the
467 ; cgi.force_redirect configuration below
468 doc_root =
469
470 ; The directory under which PHP opens the script using /~username used only
471 ; if nonempty.
472 user_dir =
473
474 ; Directory in which the loadable extensions (modules) reside.
475 extension_dir = "C:\PHP\ext"
476
477 ; Whether or not to enable the dl() function. The dl() function does NOT work
478 ; properly in multithreaded servers, such as IIS or Zeus, and is automatically
479 ; disabled on them.
480 enable_dl = On
481
482 ; cgi.force_redirect is necessary to provide security running PHP as a CGI under
483 ; most web servers. Left undefined, PHP turns this on by default. You can
484 ; turn it off here AT YOUR OWN RISK
485 ; **You CAN safely turn this off for IIS, in fact, you MUST.**
486 cgi.force_redirect = 1
487
488 ; if cgi.nph is enabled it will force cgi to always sent Status: 200 with
489 ; every request.
490 cgi.nph = 1
  
```

รูปที่ 2.9 หน้าจอแสดงการปรับแต่งค่าการโหลด extension (modules)

5.6) ทหารรทัดที่เขียนว่า ;upload_tmp_dir = แล้วแก้ไขเป็น upload_tmp_dir = C:\php\uploadtemp โดยเอาเครื่องหมาย ; ด้านซ้ายออก และให้สร้างโฟลเดอร์ชื่อว่า uploadtemp ไว้ภายใน C:\PHP\ ด้วย ซึ่งเป็นกรสร้างโฟลเดอร์ชั่วคราวสำหรับการอัปโหลด

```

512 ; Default is zero.
513 ;cgi.rfc2616_headers = 0
514
515
516 ; File Uploads ;
517 ; File Uploads ;
518 ; File Uploads ;
519
520 ; Whether to allow HTTP file uploads.
521 file_uploads = On
522
523 ; Temporary directory for HTTP uploaded files (will use system default if not
524 ; specified).
525 upload_tmp_dir = c:\php\uploadtemp
526
527 ; Maximum allowed size for uploaded files.
528 upload_max_filesize = 2M
529
530
531 ; Fopen wrappers ;
532 ; Fopen wrappers ;
533 ; Fopen wrappers ;
534
535 ; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
536 allow_url_fopen = On
537
538 ; Whether to allow include/require to open URLs (like http:// or ftp://) as files.
  
```

รูปที่ 2.10 หน้าจอแสดงการปรับแต่งค่าของ Temporary directory

5.7) ทหารตัดที่เขียนว่า ;session.save_path = "/tmp" แล้วแก้ไขเป็น session.save_path = C:\php\sessiondata โดยเอาเครื่องหมาย ; ด้านหน้าออก และให้สร้างโฟลเดอร์ชื่อว่า sessiondata ไว้ภายใน C:\PHP\ ด้วย ซึ่งในที่นี้จะเป็นการสร้างโฟลเดอร์เก็บข้อมูลในกรณีที่มีการใช้ตัวแปรประเภท session

```

914 ; NOTE 2: See the section on garbage collection below if you choose to
915 ; use subdirectories for session storage
916 ;
917 ; The file storage module creates files using mode 600 by default.
918 ; You can change that by using
919 ;
920 ;     session.save_path = "N:MODE;/path"
921 ;
922 ; where MODE is the octal representation of the mode. Note that this
923 ; does not overwrite the process's umask
924 session.save_path = "C:\php\sessiondata"
925 ;
926 ; Whether to use cookies.
927 session.use_cookies = 1
928 ;
929 ;session.cookie_secure =
930 ;
931 ; This option enables administrators to make their users invulnerable to
932 ; attacks which involve passing session ids in URLs; defaults to 0.
933 ; session.use_only_cookies = 1
934 ;
935 ; Name of the session (used as cookie name).
936 session.name = PHPSESSID
937 ;
938 ; Initialize session on request startup.
939 session.auto_start = 0
940 ;

```

รูปที่ 2.11 หน้าจอแสดงการปรับแต่งค่าโฟลเดอร์ในไฟล์เก็บค่า session

5.8) ทหารตัดที่เขียนว่า ;extension=php_mysql.dll ให้ลบเครื่องหมาย ; ด้านหน้าออก ซึ่งเป็นการเพิ่ม extension เกี่ยวกับการทำงานของฐานข้อมูล MySQL

```

597 ;extension=php_imap.dll
598 ;extension=php_interbase.dll
599 ;extension=php_ldap.dll
600 ;extension=php_mcrypt.dll
601 ;extension=php_mhash.dll
602 ;extension=php_mime_magic.dll
603 ;extension=php_ming.dll
604 ;extension=php_mysql.dll
605 ;extension=php_mysql.dll
606 extension=php_mysql.dll
607 ;extension=php_oci8.dll
608 ;extension=php_openssl.dll
609 ;extension=php_oracle.dll
610 ;extension=php_pgsql.dll
611 ;extension=php_shmop.dll
612 ;extension=php_snmp.dll
613 ;extension=php_sockets.dll
614 ;extension=php_sqlite.dll
615 ;extension=php_sybase_ct.dll
616 ;extension=php_tidy.dll
617 ;extension=php_xmlrpc.dll
618 ;extension=php_xsl.dll
619 ;
620 ;
621 ;::::::::::::::::::|
622 ; Module Settings ;
623 ;::::::::::::::::::|

```

รูปที่ 2.12 หน้าจอแสดงการลบ ; ด้านหน้าคำสั่ง extension=php_mysql.dll

5.9) บันทึกไฟล์ php.ini นี้ และปิดโปรแกรม Editor

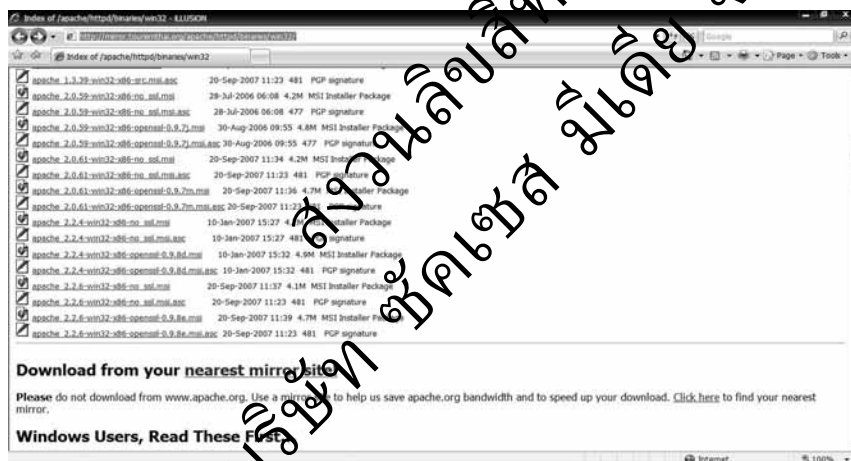
5.10) Copy ไฟล์ libmysql.dll และ phpsts.dll ซึ่งอยู่ในโฟลเดอร์ C:\PHP แล้วนำไปวางไว้ที่ C:\WINDOWS\system32 เป็นอันเสร็จสิ้นขั้นตอนการติดตั้งโปรแกรม PHP Engine

2.2 การติดตั้งโปรแกรม Apache Web Server

โปรแกรม Apache Web Server เป็นโปรแกรมที่ใช้ในการทำให้เครื่องคอมพิวเตอร์มีคุณสมบัติเป็น Web server คือเครื่องคอมพิวเตอร์เครื่องนั้นสามารถให้บริการต่างๆ ทางด้านเว็บเพจได้ ขั้นตอนในการติดตั้ง Apache Web Server ประกอบไปด้วยขั้นตอน ดังนี้

1. Download โปรแกรมติดตั้ง (โดยเลือกไฟล์ที่เป็น MSI Installer Package) ที่เว็บ <http://httpd.apache.org/download.cgi> หรือ

<http://mirror.tourismthai.org/apache/httpd/binaries/win32/>

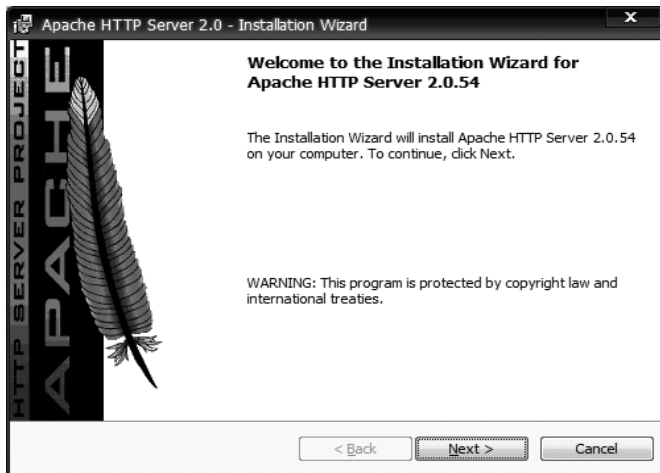


รูปที่ 2.13 เว็บไซต์ดาวน์โหลด Apache Web Server

2. ในที่นี้จะทำการติดตั้ง Apache Web Server ในเวอร์ชัน 2.0.54 ให้ทำการดับเบิลคลิกที่ไฟล์ติดตั้ง apache_2.0.54-win32-x86-no_ssl.msi ก็จะเริ่มเข้าสู่กระบวนการในการติดตั้ง โดยจะเข้าสู่หน้าจอของการต้อนรับการติดตั้งโปรแกรม ให้ทำการคลิกที่ปุ่ม Next

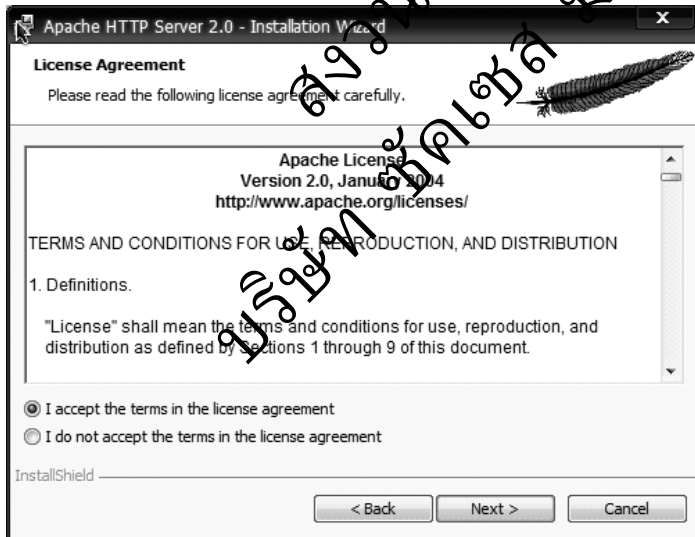


การติดตั้ง Apache Web Server เวอร์ชันอื่นๆ ที่ไม่ตรงกับตัวอย่างในหนังสือ อาจมีขั้นตอนที่ไม่เป็นไปตามตัวอย่าง ซึ่งเราสามารถอ่านขั้นตอนการติดตั้งโปรแกรมเวอร์ชันนั้นได้จากเว็บ www.apache.org



รูปที่ 2.14 หน้าจอเริ่มต้นรับการติดตั้ง

3. เข้าสู่หน้าจอของข้อมูลลิขสิทธิ์ของโปรแกรม ใช้ทำการเลือก I accept the terms in the license agreement หลังจากนั้นให้ทำการคลิกปุ่ม Next



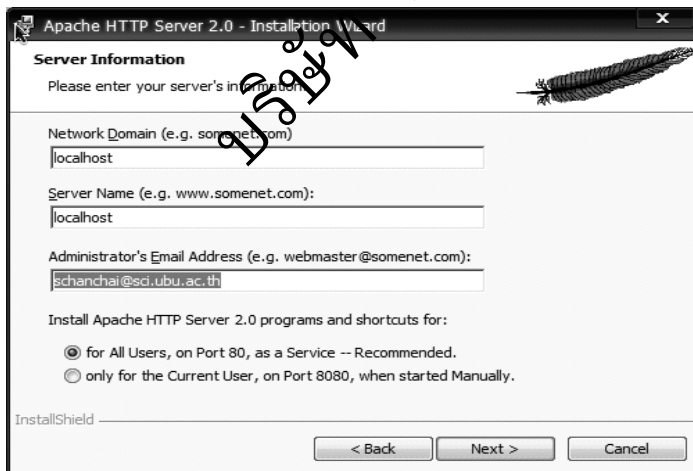
รูปที่ 2.15 หน้าจอแสดงลิขสิทธิ์ของโปรแกรม

4. เข้าสู่หน้าจอของรายละเอียดของโปรแกรม Apache Web Server ให้ทำการคลิกที่ปุ่ม Next



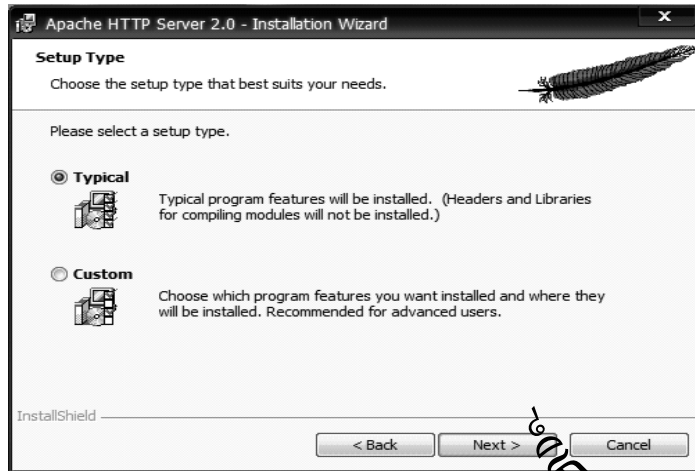
รูปที่ 2.16 หน้าจอแสดงรายละเอียดของโปรแกรม

5. เข้าสู่หน้าจอของการกำหนด Network Domain, Server Name และ Administrator's Email Address โดยใน 2 ช่องแรกใช้ทำการพิมพ์ localhost ส่วนช่องสุดท้ายให้ทำการพิมพ์ E-mail address ของตนเองลงไป หลังจากนั้นให้ทำการคลิกปุ่ม Next



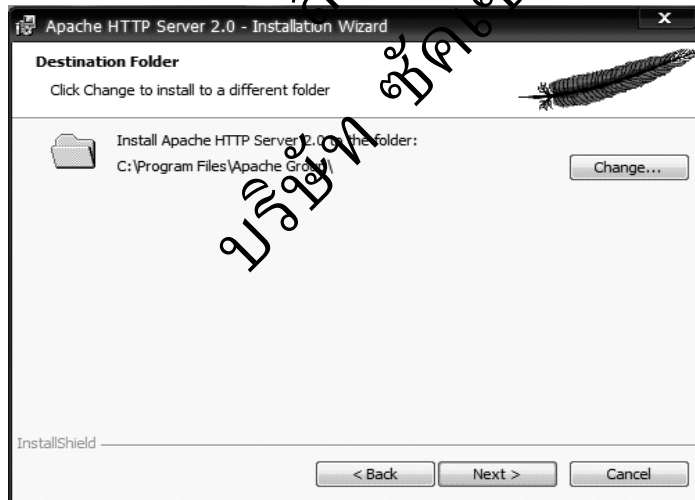
รูปที่ 2.17 หน้าจอการกำหนดข้อมูลของ Server

6. เข้าสู่หน้าจอของการเลือกประเภทของการติดตั้ง ให้เลือกประเภทของการติดตั้งเป็น Typical และทำการคลิกปุ่ม Next



รูปที่ 2.18 หน้าจอการเลือกประเภทของการติดตั้ง

7. เข้าสู่หน้าจอของการเลือกไดรฟ์และโฟลเดอร์ในการติดตั้งโปรแกรม ในที่นี้เราจะไม่ทำการเปลี่ยนแปลงใดๆ คือทำการติดตั้งที่ C:\Program Files\Apache Group\ หลังจากนั้นให้ทำการคลิกที่ปุ่ม Next



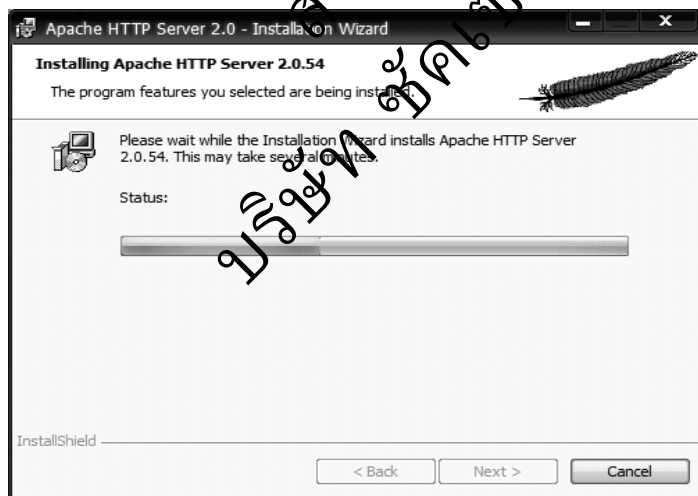
รูปที่ 2.19 หน้าจอการเลือกไดรฟ์ และโฟลเดอร์ในการติดตั้ง

8. เข้าสู่หน้าจอก่อนที่จะเริ่มต้นกระบวนการของการติดตั้ง ให้ทำการคลิกที่ปุ่ม Install



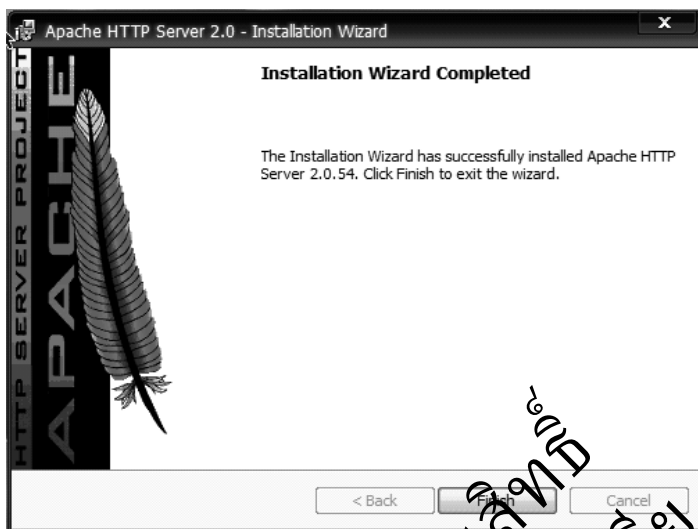
รูปที่ 2.20 หน้าจอก่อนเริ่มต้นกระบวนการติดตั้ง

9. เข้าสู่หน้าจอแสดงความคืบหน้าในการติดตั้ง ให้รอจนกระทั่งโปรแกรมทำการติดตั้งเสร็จ



รูปที่ 2.21 หน้าจอแสดงสถานะในการติดตั้ง

10. หลังจากติดตั้งเสร็จแล้วจะปรากฏหน้าจอแสดงข้อความว่า Installation Wizard Completed ซึ่งแสดงว่าการติดตั้งเสร็จสมบูรณ์แล้ว ให้ทำการคลิกที่ปุ่ม Finish เป็นอันเสร็จขั้นตอนของการติดตั้ง Apache Web Server



รูปที่ 2.22 หน้าจอสุดท้ายหลังจากติดตั้งโปรแกรมเสร็จสมบูรณ์

2.3 การปรับแต่ง Apache Web Server

หลังจากทำการติดตั้งโปรแกรม Apache Web Server เสร็จเรียบร้อยแล้ว เพื่อให้การทำงานของโปรแกรมมีความสมบูรณ์ให้ทำการปรับแต่งโปรแกรมโดยทำการเปิดไฟล์ httpd.conf ด้วย Text Editor โดยเข้าไปที่ไดเรกทอรีที่ทำการติดตั้งโปรแกรม คือ C:\Program Files\Apache Group\Apache2\Conf\ และให้ทำการแก้ไข ดังนี้

1. ทหารรทัดที่เขียนว่า

DocumentRoot "C:/Program Files/Apache Group/Apache2/htdocs" ซึ่งเป็นบรรทัดที่บอกตำแหน่งจัดเก็บหน้าเว็บ โดยถ้าทำการพิมพ์ URL เป็น http://localhost/ และตามด้วยชื่อไฟล์จะสามารถเข้าถึงข้อมูลของไฟล์นั้นที่อยู่ในตำแหน่ง DocumentRoot ได้ทันที ในที่นี้ให้ทำการแก้ไขที่จัดเก็บไฟล์เว็บเพจใหม่เป็นชื่อโฟลเดอร์ใหม่และลึกลงไปเพียงระดับเดียวก็พอ ซึ่งให้ทำการแก้ไขเป็น DocumentRoot "C:/myweb" และที่ไดเรก C: ให้เราทำการสร้างโฟลเดอร์ที่มีชื่อว่า myweb ด้วย

บทที่

2

```

218 # by the Client. When set "On", Apache will use the value of the
219 # ServerName directive.
220 #
221 UseCanonicalName Off
222 #
223 #
224 # DocumentRoot: The directory out of which you will serve your
225 # documents. By default, all requests are taken from this directory, but
226 # symbolic links and aliases may be used to point to other locations.
227 #
228 DocumentRoot "C:/myweb"
229 #
230 #
231 # Each directory to which Apache has access can be configured with respect
232 # to which services and features are allowed and/or disabled in that
233 # directory (and its subdirectories).
234 #
235 # First, we configure the "default" to be a very restrictive set of
236 # features.
237 #
238 <Directory />
239     Options FollowSymLinks
240     AllowOverride None
241 </Directory>
242 #
243 #
244 # Note that from this point forward you must specifically allow

```

รูปที่ 2.23 หน้าจอแสดงการปรับแต่ง DocumentRoot

2. ทาบรรทัดที่เขียนว่า

<Directory "C:/Program Files/Apache Group/Apache2/html/docs"> ซึ่งความหมายของบรรทัดนี้มีความหมายเหมือนกับบรรทัดในข้อที่ 1. โดยเ้าการแก้ไขเป็น <Directory "C:/myweb">

```

244 # Note that from this point forward you must specifically allow
245 # particular features to be enabled - so if something's not working as
246 # you might expect, make sure that you have specifically enabled it
247 # below.
248 #
249 #
250 # This should be changed to whatever you set DocumentRoot
251 #
252 <Directory "C:/myweb">
253 #
254 #
255 # Possible values for the Options directive are "None", "All",
256 # or any combination of:
257 #   Indexes Includes FollowSymLinks SymLinksToOuter Match ExecCGI MultiViews
258 # Note that "MultiViews" must be named "explicitly" --- "Options All"
259 # doesn't give it to you.
260 #
261 # The Options directive is both respected and important. Please see
262 # http://httpd.apache.org/docs/2.0/mod/core.html#options
263 # for more information.
264 #
265 Options Indexes FollowSymLinks
266 #
267 #
268 # AllowOverride controls what directives may be placed in .htaccess files.

```

รูปที่ 2.24 หน้าจอแสดงการปรับแต่ง Directory

3. ทาบรรทัดที่เขียนว่า DirectoryIndex index.html index.html.var ซึ่งเป็นบรรทัดที่ทำการกำหนดชื่อไฟล์เริ่มต้นในกรณีที่มีพื่ม URL ไม่ใส่ชื่อไฟล์ เช่น http://localhost/ หรือ http://127.0.0.1/ ซึ่งเรียกว่าเป็นไฟล์เว็บเพจดีฟอลต์ โดยสามารถทำการกำหนดชื่อไฟล์นี้ได้หลายๆ ชื่อ และในการเรียกชื่อไฟล์เหล่านี้จะทำการหาชื่อไฟล์ตามลำดับที่กำหนดไว้ ถ้าไฟล์แรกไม่เจอก็จะไปหาไฟล์ที่สอง และสามต่อไปจนถึงชื่อไฟล์สุดท้ายที่กำหนดไว้ ในที่นี้ได้ทำการเพิ่มไฟล์ index.php ดังนี้


```
308 # Order deny,allow
309 # Deny from all
310 # </LimitExcept>
311 #</Directory>
312 #
313 #
314 # DirectoryIndex: sets the file that Apache will serve if a directory
315 # is requested.
316 #
317 # The index.html.var file (a type-map) is used to deliver content-
318 # negotiated documents. The MultiViews option can be used for the
319 # same purpose, but it is much slower.
320 #
321 # DirectoryIndex index.html index.html.var index.php
322 #
323 #
324 # AccessFileName: The name of the file to look for in each directory
325 # for additional configuration directives. See also the AllowOverride
326 # directive.
327 #
328 # AccessFileName .htaccess
329 #
330 #
331 # The following lines prevent .htaccess and .htpasswd files from being
332 # viewed by Web clients.
333 #
334 #<Files ~ "\.ht*">
```

รูปที่ 2.25 หน้าจอแสดงการปรับแต่ง DirectoryIndex

4. เพิ่ม 4 บรรทัดนี้ต่อท้ายบรรทัดสุดท้ายของไฟล์ ดังที่

```
LoadModule php5_module "C:/php/php5apache2_3.dll"
AddType application/x-httpd-php .php
ScriptAlias /php/ "C:/php/"
Action application/x-httpd-php "/php/php5apache2.dll"
```

- LoadModule php5_module "C:/php/php5apache2.dll"

เป็นการกำหนดให้ Apache โหลดโมดูล php5apache2.dll ขึ้นมาทำงาน โดยไฟล์ php5apache2.dll นี้เป็น library ที่เกี่ยวกับการประมวลผลของไฟล์ PHP เวอร์ชัน 5

- AddType application/x-httpd-php .php

เป็นการกำหนดให้นามสกุลของไฟล์ PHP เป็น .php แต่ถ้าต้องการใช้นามสกุลเป็น .php3 ก็สามารรถแก้ไขได้ นอกจากนี้บรรทัดนี้ยังมีการกำหนด MIME (Multimedia Internet Mail Extensios) ให้มี type เป็นแบบ "application/x-httpd-php"

- ScriptAlias /php/ "C:/php/"

เป็นการกำหนดโฟลเดอร์ C:/php/ ให้เวลาที่ Apache จะทำการเข้าถึงไฟล์ php5apache2.dll (ตามที่ระบุในบรรทัด Action) จะต้องเข้าถึงผ่านทางโฟลเดอร์เสมือน C:/php/ นี้

- Action application/x-httpd-php "/php/php5apache2.dll"

เป็นการกำหนดให้ Apache เรียกไฟล์ php5apache2.dll ที่อยู่ในโฟลเดอร์เสมือน C:/php/ ขึ้นมาประมวลผลไฟล์ PHP

สงวนลิขสิทธิ์
ห้ามคัดลอก
โดยไม่ได้รับอนุญาต

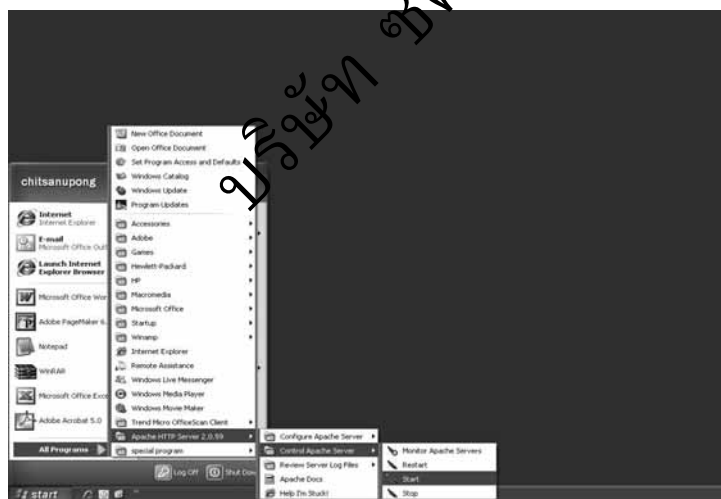
```

341 # VirtualHost example:
342 # Almost any Apache directive may go into a VirtualHost container.
343 # The first VirtualHost section is used for requests without a known
344 # server name.
345 #
346 #<VirtualHost *:80>
347 #   ServerAdmin webmaster@dummy-host.example.com
348 #   DocumentRoot /www/docs/dummy-host.example.com
349 #   ServerName dummy-host.example.com
350 #   ErrorLog logs/dummy-host.example.com-error_log
351 #   CustomLog logs/dummy-host.example.com-access_log common
352 #</VirtualHost>
353 LoadModule php5_module "C:/php/php5apache2.dll"
354 AddType application/x-httpd-php .php
355 ScriptAlias /php/ "C:/php/"
356 Action application/x-httpd-php "/php/php5apache2.dll"
357

```

รูปที่ 2.26 หน้าจอแสดงการปรับแต่งค่าใน 4 บรรทัดสุดท้าย

5. ทำการบันทึกไฟล์ httpd.conf และปิดโปรแกรม Editor นี้
6. ให้คลิกเลือก Start>Programs>Apache HTTP Server 2.0.54>Control Apache Server>Stop เพื่อปิดการทำงานของ Apache และทำการคลิกเลือก Start>Programs>Apache HTTP Server 2.0.54>Control Apache Server>Start เพื่อเรียกใช้โปรแกรม Apache ขึ้นมาทำงานใหม่ เหตุที่ต้องทำเช่นนี้ก็เพื่อให้การเปลี่ยนแปลงที่ได้จากการปรับแต่งโปรแกรมมีผลตามไปด้วย



รูปที่ 2.27 หน้าจอแสดงการ Stop และ Start โปรแกรม Apache WebServer

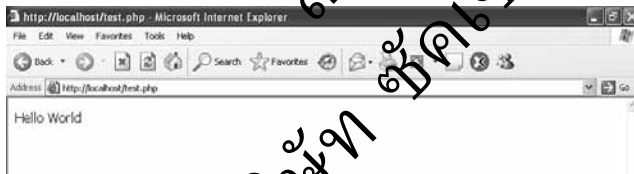
2.4 การทดสอบการทำงานของ PHP

หลังจากทำการติดตั้งโปรแกรม PHP Engine และ Apache Web Server เสร็จเรียบร้อยแล้ว ในขั้นตอนต่อไปเราจะมาทำการทดสอบการทำงานของโปรแกรมทั้งสองว่าสามารถรองรับโปรแกรมภาษา PHP ได้หรือไม่ โดยให้ทำการเขียนโปรแกรมสั้นๆ ต่อไปนี้

test.php

```
<?
    echo "Hello World";
?>
```

หลังจากนั้นให้ทำการบันทึกไฟล์นี้ไว้ที่ C:\myweb ตามที่ได้ระบุไดรฟ์และโฟลเดอร์ที่จัดเก็บไฟล์ต่างๆ ตามที่ได้อธิบายไว้ในหัวข้อที่ผ่านมา ในที่นี้บันทึกชื่อไฟล์ คือ test.php เสร็จแล้วให้ทำการเปิดโปรแกรมบราวเซอร์ และพิมพ์ URL คือ <http://localhost/test.php> หรือ <http://127.0.0.1/test.php> ซึ่งถ้าปรากฏหน้าจอต่อไปนี้ก็แสดงว่าการทำงานของโปรแกรม PHP Engine สามารถทำการแปลภาษา PHP ที่เราเขียนขึ้นได้ และโปรแกรม Apache Web Server สามารถทำงานประสานงานระหว่าง Web Server และส่งข้อมูลเว็บเพจไปแสดงผลที่ Web Browser ได้ ในที่นี้ทั้ง Web Server และ Web Browser อยู่ในเครื่องคอมพิวเตอร์เครื่องเดียวกัน)



รูปที่ 2.28 หน้าจอของการทดสอบการทำงานของ PHP

สรุป

โปรแกรม PHP Engine เป็นโปรแกรมที่ใช้ในการแปลภาษา PHP ให้เป็นภาษาเครื่อง เนื่องจากเครื่องคอมพิวเตอร์จะเข้าใจหรือรู้จักเฉพาะภาษาเครื่องเท่านั้น ส่วนขั้นตอนการติดตั้งโปรแกรม PHP Engine สามารถดาวน์โหลดตัวติดตั้งได้จากภายในเว็บไซต์ <http://www.php.net/downloads.php> และดำเนินการติดตั้งตามขั้นตอนที่ได้อธิบายไว้

โปรแกรม Apache Web Server เป็นโปรแกรมที่ใช้ในการทำให้เครื่องคอมพิวเตอร์มีคุณสมบัติเป็น Web Server คือเครื่องคอมพิวเตอร์เครื่องนั้นสามารถให้บริการต่างๆ ทางด้านเว็บเพจได้ ส่วนขั้น



บทที่

2

ตอนการติดตั้งโปรแกรม Apache Web Server สามารถดาวน์โหลดตัวติดตั้งได้จากภายในเว็บไซต์ <http://httpd.apache.org/download.cgi> หรือ <http://mirror.tourismthai.org/apache/httpd/binaries/win32/> และดำเนินการติดตั้งตามขั้นตอนที่ได้อธิบายไว้

หลังจากติดตั้งและปรับแต่งโปรแกรม PHP Engine และ Apache Web Server แล้วก็จะสามารถเขียนโปรแกรมภาษา PHP และให้ทำการประมวลผลบน Web Browser ได้ทันที

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

แบบฝึกหัดบทที่ 2

1. PHP Engine คืออะไร จงอธิบาย

.....

.....

.....

.....

.....

2. โปรแกรม Apache Web Server ทำหน้าที่อะไร

.....

.....

.....

.....

.....

3. ให้อธิบายขั้นตอนการติดตั้ง Apache Web Server

.....

.....

.....

.....

.....

4. ในการปรับแต่ง Apache Web Server บรรทัดต่อไปนี้หมายความว่าอย่างไร

DocumentRoot "C:/myweb/"

.....

.....

.....

.....

.....

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

5. ในการปรับแต่ง Apache Web Server บรรทัดต่อไปให้หมายความว่าอย่างไร

DirectoryIndex index.html

DirectoryIndex index.php

DirectoryIndex default.html

.....

.....

.....

.....

.....

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

หลังจากได้ทำการติดตั้งตัวแปลภาษา (PHP Engine) และโปรแกรม Web Server (Apache) แล้ว เนื้อหาในบทนี้จะเริ่มแนะนำการเขียนโปรแกรมภาษา PHP โดยจะเริ่มตั้งแต่บล็อกคำสั่งของ PHP, คำสั่งในการแสดงผลพีธีในรูปแบบต่างๆ, คำสั่ง Comment และส่วนสุดท้ายจะแนะนำแท็กคำสั่งของ HTML ซึ่งสามารถนำมาประยุกต์ในการเขียนโปรแกรมภาษา PHP ได้ ทำให้ผลลัพธ์ที่ได้มีการแสดงผลที่น่าสนใจและหลากหลายยิ่งขึ้น เนื้อหาโดยรวมของบทนี้ยังไม่ยาก และซับซ้อนมากนัก แต่ก็ถือว่าเป็นพื้นฐานที่สำคัญในการเขียนโปรแกรมภาษา PHP ก่อนที่จะได้ศึกษาคำสั่ง และโปรแกรมที่มีความซับซ้อนมากกว่านี้ในเนื้อหาบทถัดๆ ไป

จุดประสงค์ ทั่วไป

1. เพื่อให้ทราบวิธีการเขียนโปรแกรมภาษา PHP เบื้องต้น
2. เพื่อให้สามารถเขียนโปรแกรมภาษา PHP ง่ายๆ ในเบื้องต้นได้
3. เพื่อให้ทราบรูปแบบบล็อกคำสั่งในการเขียนโปรแกรมภาษา PHP
4. เพื่อให้ทราบคำสั่งในการแสดงผลพีธีในแต่ละคำสั่ง และใช้งานได้อย่างถูกต้อง
5. เพื่อให้ทราบประโยชน์ของการใช้คำสั่งหมายเหตุ (Comment)
6. เพื่อให้รู้จักและใช้งานคำสั่งหมายเหตุในแต่ละรูปแบบได้อย่างถูกต้อง
7. เพื่อให้ทราบการประยุกต์แท็กของ HTML มาใช้ร่วมกับโปรแกรมภาษา PHP

จุดประสงค์ เชิงพฤติกรรม

1. สามารถเขียนโปรแกรมภาษา PHP พื้นฐานได้
2. สามารถใช้คำสั่งในการแสดงผลพีธีได้
3. สามารถใช้คำสั่งหมายเหตุได้
4. สามารถประยุกต์แท็ก HTML มาใช้ร่วมกับโปรแกรมภาษา PHP ได้

เนื้อหาสาระ

1. รูปแบบบล็อกคำสั่งของ PHP
2. คำสั่งแสดงผลพีธี
3. คำสั่งหมายเหตุ (Comment)
4. แท็กคำสั่งที่สำคัญของ HTML

ในการเขียนโปรแกรมภาษา HTML จะประกอบขึ้นจากแท็กคำสั่งต่างๆ แบ่งออกเป็น 4 ส่วน ดังนี้

```
<HTML>
  <HEAD>
    <TITLE> หัวข้อเว็บเพจ </TITLE>
  </HEAD>
  <BODY>
    ส่วนที่มีเนื้อหาและข้อมูลเกี่ยวกับภาพ Link ที่ปรากฏในเว็บเพจ
  </BODY>
</HTML>
```

<HTML>...</HTML> เป็นคำสั่งแรกที่ต้องมีในภาษา HTML ซึ่งบ่งบอกว่านี่คือภาษา HTML โดยที่จะอยู่ที่จุดเริ่มต้นของเอกสารและท้ายเอกสารในแต่ละแฟ้ม

<HEAD>...</HEAD> เป็นส่วนกำหนดรายละเอียดหัวข้อเรื่อง ภายในจะมีคำสั่งย่อยอีกหนึ่งคำสั่ง คือ <TITLE>...</TITLE> โดยข้อความภายในคำสั่งนี้ให้เราใส่ชื่อหัวข้อเว็บเพจที่ต้องการ ซึ่งจะต้องมีความยาวไม่เกิน 64 ตัวอักษร หัวข้อนี้จะปรากฏที่ Title bar ของบราวเซอร์ที่ใช้งานอยู่ และปรากฏอยู่บน Bookmark ของโปรแกรมบราวเซอร์ด้วย

<BODY>...</BODY> ส่วนนี้เป็นส่วนเนื้อหาของเว็บเพจของเรา อาจจะเป็นข้อความ รูปภาพ ตารางเสียง และการเชื่อมต่อไปยังเอกสารอื่นๆ ที่จะแสดงผลที่เว็บเบราว์เซอร์

3.1 รูปแบบบล็อกคำสั่งของ PHP

สำหรับในโปรแกรมภาษา PHP ก็มีแท็กที่ใช้สำหรับการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของคำสั่ง PHP เช่นเดียวกัน ซึ่งในภาษา PHP มีรูปแบบของบล็อกคำสั่ง 3 รูปแบบ คือ

1. ใช้ <?...?>

```
<?
    print "Hello World";
?>
```

2. ใช้ <?php...?>

```
<?php
    print "Hello World";
?>
```


3. ใช้ `<script language='php'>...</script>`

```
<script language='php'>
    print "Hello World";
</script>
```

เราสามารถใช้น็ลลอกคำสั่งของ PHP รูปแบบใดก็ได้ ซึ่งก็จะให้ผลลัพธ์ไม่แตกต่างกัน โดยจากตัวอย่างข้างต้นโปรแกรมภาษา PHP จะทำการพิมพ์ข้อความว่า Hello World ออกมาแสดงผลยังบราวเซอร์ ต่อไปนี้คือข้อแนะนำบางประการสำหรับบ็ลลอกคำสั่งของ PHP

1. เราสามารถเขียนบ็ลลอกคำสั่งของ PHP แทรกไว้ในแท็ก HTML ได้ ดังตัวอย่างต่อไปนี้

```
<HTML>
<BODY>
  <?
    print "Hello World";
  ?>
</BODY>
</HTML>
```

2. เราสามารถเขียนบ็ลลอกคำสั่งของ PHP นี้ได้หลายๆ บ็ลลอกคำสั่งภายในโปรแกรมเดียวกัน ดังตัวอย่างต่อไปนี้ ซึ่งจะทำการพิมพ์ข้อความว่า Hello World Good Morning How are you? เรียงต่อกัน

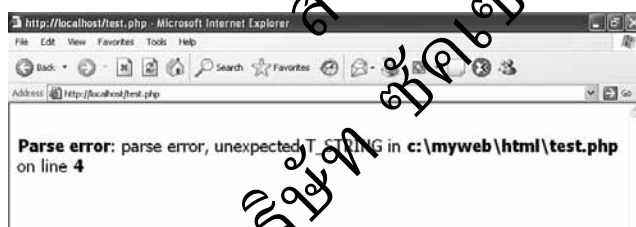
```
<HTML>
<BODY>
  <?
    print "Hello World ";
  ?>
  Good Morning
  <?
    print "How are you";
  ?>
</BODY>
</HTML>
```

3. เราสามารถเขียนบล็อกคำสั่งของ PHP ได้อย่างอิสระโดยไม่ต้องอยู่ภายในแท็กของคำสั่ง HTML ก็ได้ ดังตัวอย่างต่อไปนี้

```
<?
    print "Hello World";
?>
```

4. เมื่อเราเปิดทำการเขียนบล็อกคำสั่งของ PHP แล้ว จำไว้เสมอว่าภายในบล็อกคำสั่งนี้จะต้องเป็นคำสั่งภาษา PHP เท่านั้น มิฉะนั้นจะไม่สามารถแสดงผลลัพธ์ได้ และโปรแกรมจะแจ้ง error หรือความผิดพลาดที่เกิดขึ้นบนเบราว์เซอร์ ดังตัวอย่างต่อไปนี้

```
<HTML>
<BODY>
    <?
        Hello World
    ?>
</BODY>
</HTML>
```



รูปที่ 3.1 เบราวเซอร์แสดงความผิดพลาดเนื่องจากภายในบล็อกคำสั่ง PHP ไม่ใช่คำสั่งของ PHP

5. เมื่อเราเปิดบล็อกคำสั่ง PHP แล้วภายในบล็อกคำสั่งนี้จะบรรจุคำสั่งภาษา PHP ก็คำสั่งก็ได้ แต่ว่าในแต่ละคำสั่งจะต้องลงท้ายด้วยเครื่องหมาย ; (semi-colon) และเราสามารถเขียนคำสั่งของ PHP หลายๆ คำสั่งภายใน 1 บรรทัดก็ได้ ดังตัวอย่างต่อไปนี้

```
<?
    print "Hello "; print "World";
?>
```

3.2 คำสั่งแสดงผลลัพธ์

ในการแสดงผลลัพธ์ของโปรแกรมภาษา PHP ได้จัดเตรียมคำสั่งไว้ 3 รูปแบบ คือ

1. echo รูปแบบของคำสั่งนี้เป็น ดังนี้

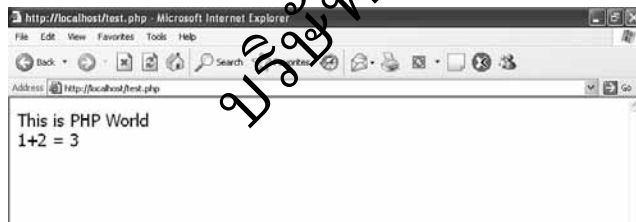
รูปแบบ

```
echo "... ข้อมูลที่ต้องการแสดงผลลัพธ์ ..."
```

พิจารณาตัวอย่างของการใช้คำสั่ง echo เพื่อแสดงผลลัพธ์ ต่อไปนี้

```
<HTML>
<BODY>
  <?
    echo "This is PHP World";
  ?>
  <br>
  <?
    echo "1+2 = ",1+2;
  ?>
</BODY>
</HTML>
```

จากโปรแกรมจะแสดงผลลัพธ์บนเบราว์เซอร์ ดังต่อไปนี้



รูปที่ 3.2 เบราวเซอร์แสดงผลลัพธ์ของคำสั่ง echo

จากโปรแกรมข้างต้นสามารถอธิบายการทำงานได้ ดังนี้

echo "This is PHP World";	พิมพ์ข้อความว่า This is PHP World บนเบราว์เซอร์
 	แท็กของ HTML ในการขึ้นบรรทัดใหม่
echo "1+2 = ",1+2;	พิมพ์ข้อความ 1+2 ตามด้วย ผลลัพธ์ของ 1+2 โดยใช้เครื่องหมาย , แยกข้อมูล 2 ส่วนนี้

2. print รูปแบบของคำสั่งนี้เป็น ดังนี้

รูปแบบ

```
print "... ข้อมูลที่ต้องการแสดงผลลัพธ์ ..."
```

รูปแบบคำสั่งการใช้งานของคำสั่ง print และ echo จะมีรูปแบบเหมือนกัน พิจารณาตัวอย่างของการใช้คำสั่ง print เพื่อแสดงผลลัพธ์ ต่อไปนี้

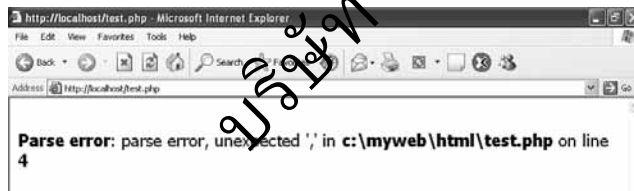
```
<?
    print "This is PHP World";
?>
```

จากโปรแกรมนี้จะทำการพิมพ์ข้อความว่า This is PHP World ปรากฏบนเบราว์เซอร์

ข้อแตกต่างของคำสั่ง echo และ print ก็คือ คำสั่ง print ไม่สามารถแยกข้อความ กับนิพจน์ หรือค่าของตัวแปรใดๆ ด้วยเครื่องหมาย , ได้ แต่คำสั่ง echo สามารถทำได้ สมมติเราเขียนโปรแกรม ดังต่อไปนี้

```
<?
    print "1+2 = ",1+2;
?>
```

เมื่อทำการรันโปรแกรมนั้นบนเบราว์เซอร์จะแสดงข้อผิดพลาด ดังนี้



รูปที่ 3.3 เบราวเซอร์แสดงความผิดพลาดจากการใช้คำสั่ง print

3. printf รูปแบบของคำสั่งนี้เป็น ดังนี้

รูปแบบ

```
printf ("... ข้อมูลที่ต้องการแสดงผลลัพธ์ ...")
```

รูปแบบคำสั่ง printf จะแตกต่างจากคำสั่ง echo และ print คือจะต้องมีเครื่องหมายวงเล็บเปิดและปิด เช่นตัวอย่างโปรแกรมต่อไปนี้

```
<?
    printf ("This is PHP World");
?>
```

นอกจากนั้นคำสั่ง printf ยังมีลักษณะพิเศษก็คือ คำสั่ง printf สามารถจัดรูปแบบการแสดงผลของข้อมูลที่เป็นตัวเลขได้ด้วย ดังตารางต่อไปนี้

สัญลักษณ์	คำอธิบาย
%d	แสดงค่าอาทิวเมตติในรูปแบบของเลขฐานสิบ
%b	แสดง Integer ในรูปแบบของเลขฐานสอง
%c	แสดง Integer ในรูปแบบของรหัส ASCII
%f	แสดง Integer ในรูปแบบของเลขทศนิยม
%o	แสดง Integer ในรูปแบบของเลขฐานแปด
%s	แสดงค่าอาทิวเมตติในรูปแบบของเลขสตริง
%x	แสดง Integer ในรูปแบบของเลขฐานสิบหก (ตัวพิมพ์เล็ก)
%X	แสดง Integer ในรูปแบบของเลขฐานสิบหก (ตัวพิมพ์ใหญ่)

พิจารณาตัวอย่างโปรแกรมต่อไปนี้

```
<?
    printf ("10/3 = %d + 1/3", 10/3);
    printf ("10/3 = %f", 10/3);
?>
```

เมื่อทำการรันโปรแกรมแล้ว ผลลัพธ์จะแสดงผลบนบราวเซอร์ เป็นดังนี้



รูปที่ 3.4 บราวเซอร์แสดงผลของการใช้คำสั่ง printf

ผลลัพธ์ในแต่ละบรรทัดสามารถอธิบายได้ ดังนี้

```
printf ("10/3 = %d <br>", 10/3);
```

ใช้รหัสรูปแบบ %d แสดงรูปแบบของเลข

จำนวนเต็ม ดังนั้น 10/3 จะแสดงผลคือ 3

```
printf ("10/3 = %.2f", 10/3);
```

ใช้รหัสรูปแบบ %.2f แสดงรูปแบบของเลข

ทศนิยม 2 ตำแหน่ง ดังนั้น 10/3 จะแสดงผลคือ 3.33

3.3 คำสั่งหมายเหตุ (Comment)

ในการเขียนโปรแกรมบางครั้งเราอาจต้องการเขียนข้อความอะไรบางอย่างเพื่ออธิบายว่าคำสั่งในแต่ละบรรทัดนั้น คืออะไร หรือมันทำงานอย่างไร เพื่อที่เราอาจจะย้อนกลับมาศึกษาโปรแกรมที่เขียนขึ้นอีกครั้ง และทำให้เข้าใจโปรแกรมเดิมได้อย่างรวดเร็ว โปรแกรมภาษา PHP มีเครื่องหมาย Comment ที่ใช้สำหรับการระบุข้อความใดลงในโปรแกรม โดยที่ส่วนของโปรแกรมที่ถูก Comment ไว้จะไม่ถูกแปลภาษา ดังนั้นไม่ว่าเราจะเขียนอะไรลงในส่วนที่ถูก Comment ไว้ก็จะมีผลต่อการทำงานของโปรแกรมไม่ได้ เครื่องหมาย Comment ของโปรแกรมภาษา PHP มีอยู่ 3 รูปแบบ คือ

1. เครื่องหมาย #

2. เครื่องหมาย //

ทั้ง 2 คำสั่งนี้ใช้ระบุ Comment ได้ทีละ 1 บรรทัด ดังตัวอย่างต่อไปนี้

```
<?
    echo "This is PHP World";
    # echo "Good Morning";
    // echo "How are you?";
?>
```

จากโปรแกรมนี้เมื่อรันโปรแกรมแล้วเบราว์เซอร์จะทำการแสดงข้อความว่า This is PHP World เพียงบรรทัดเดียวเท่านั้น

3. เครื่องหมาย /*...*/ เครื่องหมาย Comment นี้สามารถทำการ Comment ได้ครั้งละหลายบรรทัด ดังตัวอย่างต่อไปนี้

```
<?
    echo "This is PHP World";
    /* echo "Good Morning";
       echo "How are you?"; */
?>
```

เช่นเดียวกันเมื่อรันโปรแกรมนี้ บราวเซอร์จะแสดงข้อความว่า This is PHP World เพียงบรรทัดเดียวเท่านั้น แต่โปรแกรมนี้ใช้เครื่องหมาย /*...*/ ซึ่งจะทำให้ Comment ได้ทีละหลายๆ บรรทัด

3.4 แท็กคำสั่งที่สำคัญของ HTML

เนื้อหาที่สำคัญในหัวข้อนี้จะกล่าวถึงแท็กพื้นฐานที่สำคัญของ HTML ที่ควรทราบ โดยจะทำการแทรกแท็กของ HTML นี้เข้าไปยังโปรแกรมภาษา PHP ซึ่งจะทำให้โปรแกรมที่ได้สามารถแสดงผลพีชในลักษณะที่หลากหลายยิ่งขึ้น นอกจากนี้ถ้าพื้นฐาน HTML เรายดี การเขียนโปรแกรมก็จะสามารถทำได้รวดเร็วมากยิ่งขึ้นด้วย สรุปแท็กภาษา HTML ที่ควรทราบดังต่อไปนี้

แท็กภาษา HTML	ความหมาย
• <TITLE>...</TITLE>	แท็กคำสั่งในการระบุข้อความบริเวณ Title Bar ของบราวเซอร์
• <H1>...</H1>	แท็กคำสั่งในการกำหนดขนาดของตัวอักษรขนาด 1 (ใหญ่ที่สุด)
• <H2>...</H2>	แท็กคำสั่งในการกำหนดขนาดของตัวอักษรขนาด 2
• <H3>...</H3>	แท็กคำสั่งในการกำหนดขนาดของตัวอักษรขนาด 3
• <H4>...</H4>	แท็กคำสั่งในการกำหนดขนาดของตัวอักษรขนาด 4
• <H5>...</H5>	แท็กคำสั่งในการกำหนดขนาดของตัวอักษรขนาด 5
• <H6>...</H6>	แท็กคำสั่งในการกำหนดขนาดของตัวอักษรขนาด 6 (เล็กที่สุด)
• ...	แท็กคำสั่งในการกำหนดสีของตัวอักษร
• <P>	แท็กคำสั่งในการขึ้นบรรทัดใหม่และเว้น 1 บรรทัด
• 	แท็กคำสั่งในการขึ้นบรรทัดใหม่
• ข้อความที่ใช้เป็น link 	แท็กคำสั่งในการสร้าง link ของตัวหนังสือ
• 	แท็กคำสั่งในการสร้าง Bullet

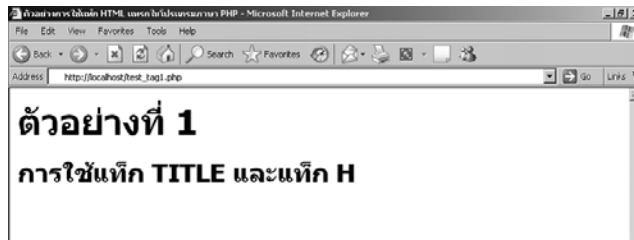
แท็กภาษา HTML	ความหมาย
<ul style="list-style-type: none"> • <ul style="list-style-type: none"> • 	แท็กคำสั่งในการสร้างหมายเลขลำดับของหัวข้อต่างๆ
<ul style="list-style-type: none"> • ... 	แท็กคำสั่งในการกำหนดให้ตัวหนังสือเป็นตัวหนา
<ul style="list-style-type: none"> • <I>...</I> 	แท็กคำสั่งในการกำหนดให้ตัวหนังสือเป็นตัวเอียง
<ul style="list-style-type: none"> • <U>...</U> 	แท็กคำสั่งในการกำหนดให้ตัวหนังสือมีเส้นใต้
<ul style="list-style-type: none"> • <CENTER>...</CENTER> 	แท็กคำสั่งในการกำหนดให้ผลลัพธ์อยู่กึ่งกลางหน้า
<ul style="list-style-type: none"> • <TABLE> <ul style="list-style-type: none"> <TR> <ul style="list-style-type: none"> <TD>...</TD> <TD>...</TD> ... </TR> ... • </TABLE> 	แท็กคำสั่งในการสร้างตาราง

ตัวอย่างต่อไปนี้ทั้งหมดจะแสดงให้เห็นถึงการนำแท็กต่างๆ ของ HTML ตามที่อธิบายไว้ข้างต้นแทรกเข้าไปในโปรแกรมภาษา PHP

test_tag1.php

```
<?
echo "<TITLE>ตัวอย่างการใช้แท็ก HTML แทรกในโปรแกรมภาษา PHP </TITLE>
<H1>ตัวอย่างที่ 1 </H1>;
<H2> การใช้แท็ก TITLE และแท็ก H </H2>";
?>
```


ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 3.5 ผลลัพธ์จากการใช้แท็ก TITLE และแท็ก H

ในโปรแกรมนี้แสดงการใช้แท็ก TITLE และแท็ก H โดยโปรแกรมจะใช้แท็ก TITLE เพื่อระบุข้อความว่า "ตัวอย่างการใช้แท็ก HTML แทรกในโปรแกรมภาษา PHP" บริเวณบน Title bar ของบราวเซอร์ และใช้แท็ก H ในการกำหนดขนาดของตัวอักษร ซึ่งในโปรแกรมนี้ทำการกำหนดขนาดของข้อความ "ตัวอย่างที่ 1" ให้มีขนาด 1 ซึ่งมีขนาดใหญ่ที่สุด และสุดท้ายในบรรทัดถัดมาก็ทำการกำหนดขนาดของข้อความ "การใช้แท็ก TITLE และแท็ก H" ให้มีขนาด 2 ซึ่งมีขนาดเล็กกว่าขนาด 1 เล็กน้อย

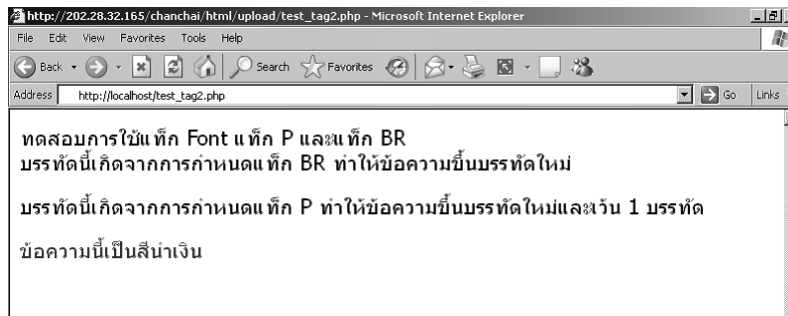
Tip

จากโปรแกรมนี้เราใช้คำสั่ง echo ในการระบุแท็กต่างๆ ของ HTML เนื่องจากภายในบล็อกคำสั่งของ PHP จะต้องเป็นคำสั่ง PHP เท่านั้น

test_tag2.php

```
<?
echo "ทดสอบการใช้แท็ก Font แท็ก P และแท็ก BR";
echo "<BR>บรรทัดนี้เป็นการกำหนดแท็ก BR ทำให้ข้อความขึ้นบรรทัดใหม่";
echo "<P> บรรทัดนี้เกิดจากการกำหนดแท็ก P ทำให้ข้อความขึ้นบรรทัดใหม่และเว้น 1
บรรทัด ";
echo "<FONT COLOR=Blue>";
echo "<P>ข้อความนี้เป็นสีน้ำเงิน";
echo "</FONT>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



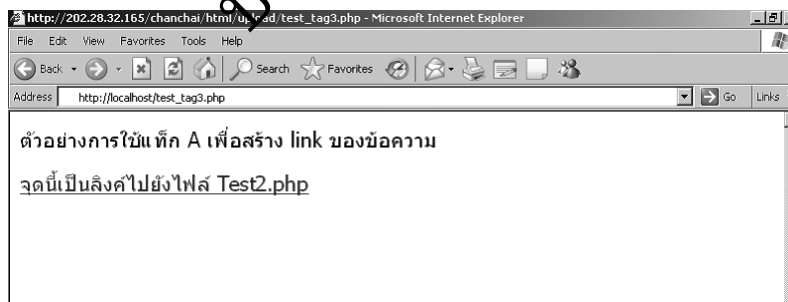
รูปที่ 3.6 ผลลัพธ์จากการใช้แท็ก FONT แท็ก P และแท็ก BR

ผลลัพธ์ของโปรแกรมนี้ในบรรทัดแรกเป็นการใช้คำสั่ง echo พิมพ์ข้อความว่า "ทดสอบการใช้แท็ก Font แท็ก P และแท็ก BR" หลังจากนั้นข้อความที่บอกว่า "บรรทัดนี้เกิดจากการกำหนดแท็ก BR ทำให้ข้อความขึ้นบรรทัดใหม่" จะแสดงผลโดยการขึ้นบรรทัดใหม่ ซึ่งก็เนื่องมาจากแท็ก
 หลังจากนั้นข้อความที่บอกว่า "บรรทัดนี้เกิดจากการกำหนดแท็ก P ทำให้ข้อความขึ้นบรรทัดใหม่และเว้น 1 บรรทัด" จะแสดงผลโดยการขึ้นบรรทัดใหม่และเว้น 1 บรรทัด ซึ่งก็เนื่องมาจากแท็ก <P> และสุดท้ายข้อความที่บอกว่า "ข้อความนี้เป็นสีน้ำเงิน" จะแสดงผลเป็นตัวหนังสือสีน้ำเงิน ซึ่งก็เนื่องมาจากการใช้แท็ก ...

test_tag3.php

```
<?
echo "ตัวอย่างการใช้แท็ก A เพื่อสร้าง link ของข้อความ <P>";
echo "<A HREF=Test2.php >จุดนี้เป็นลิงค์ไปยังไฟล์ Test2.php </A> ";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



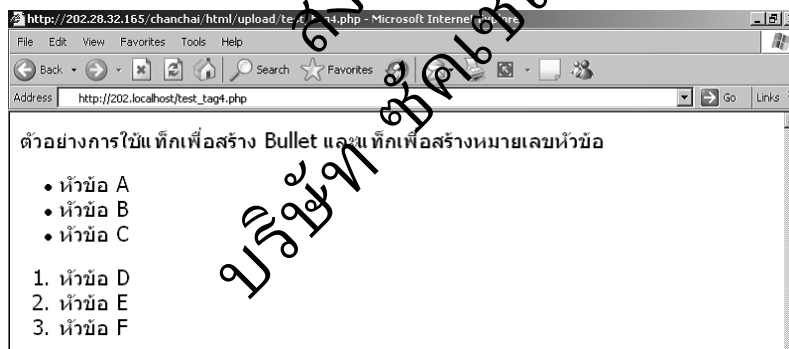
รูปที่ 3.7 ผลลัพธ์จากการใช้แท็ก A

ผลลัพธ์ของโปรแกรมนี้เป็นการใช้แท็ก ข้อความที่ต้องการลิงค์ โดยในที่นี้ ไฟล์ที่ต้องการลิงค์ คือ Test2.php และข้อความที่ต้องการลิงค์ คือ "จุดนี้เป็นลิงค์ไปยังไฟล์ Test2.php"

test_tag4.php

```
<?
echo "ตัวอย่างการใช้แท็กเพื่อสร้าง Bullet และแท็กเพื่อสร้างหมายเลขหัวข้อ <P>";
echo "<UL>";
    echo "<LI>หัวข้อ A </LI>";
    echo "<LI>หัวข้อ B </LI>";
    echo "<LI>หัวข้อ C </LI>";
echo "</UL>";
echo "<OL>";
    echo "<LI>หัวข้อ D </LI>";
    echo "<LI>หัวข้อ E </LI>";
    echo "<LI>หัวข้อ F </LI>";
echo "</OL>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



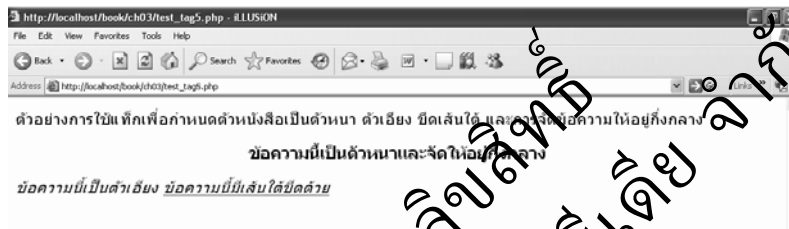
รูปที่ 3.8 ผลลัพธ์จากการใช้แท็กเพื่อสร้าง Bullet และสร้างหมายเลขหัวข้อ

ผลลัพธ์ของโปรแกรมนี้ ข้อความ "หัวข้อ A", "หัวข้อ B" และ "หัวข้อ C" จะมี Bullet ปากฎขึ้น ซึ่งเนื่องมาจากการใช้แท็ก ... และข้อความ "หัวข้อ D", "หัวข้อ E" และ "หัวข้อ F" จะมีหมายเลขหัวข้อปากฎขึ้น ซึ่งเนื่องมาจากการใช้แท็ก ...

test_tag5.php

```
<?
echo "ตัวอย่างการใช้แท็กเพื่อกำหนดตัวหนังสือเป็นตัวหนา ตัวเอียง ชิดเส้นใต้ และการจัดข้อความให้อยู่กึ่งกลาง <P>";
echo "<B><CENTER>";
echo "ข้อความนี้เป็นตัวหนาและจัดให้อยู่กึ่งกลาง";
echo "</B></CENTER><P>";
echo "<I> ข้อความนี้เป็นตัวเอียง </I>";
echo "<U> ข้อความนี้มีเส้นใต้ขีดด้วย </U>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 3.9 ผลลัพธ์จากการใช้แท็กเพื่อตกแต่งข้อความ

ผลลัพธ์ของโปรแกรมนี้จะเริ่มจากการเขียนข้อความว่า "ข้อความนี้เป็นตัวหนาและจัดให้อยู่กึ่งกลาง" โดยทำการกำหนดให้ตัวหนังสือเป็นตัวหนาและจัดให้อยู่กึ่งกลางหน้า ซึ่งก็เนื่องมาจากการใช้แท็ก `...` และแท็ก `<CENTER>...</CENTER>` ถัดมาข้อความว่า "ข้อความนี้เป็นตัวเอียง" โดยทำการกำหนดให้ตัวหนังสือเป็นตัวเอียง ซึ่งก็เนื่องมาจากการใช้แท็ก `<I>...</I>` และสุดท้ายข้อความว่า "ข้อความนี้มีเส้นใต้ขีดด้วย" โดยทำการกำหนดให้ตัวหนังสือมีเส้นใต้ ซึ่งก็เนื่องมาจากการใช้แท็ก `<U>...</U>`

test_tag6.php

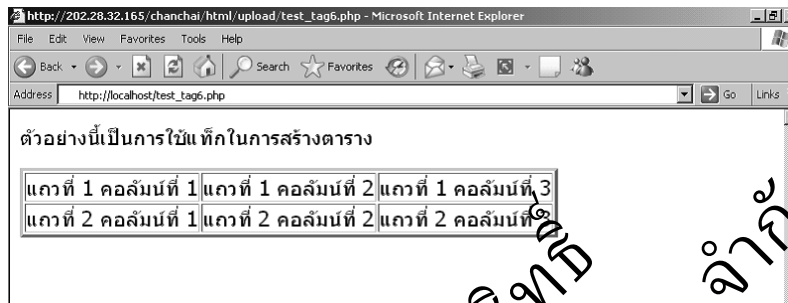
```
<?
echo "ตัวอย่างนี้เป็นการใช้แท็กในการสร้างตาราง <P>";
echo "<TABLE BORDER=3>";
echo "<TR>
    <TD>แถวที่ 1 คอลัมน์ที่ 1 </TD>
    <TD>แถวที่ 1 คอลัมน์ที่ 2 </TD>
    <TD>แถวที่ 1 คอลัมน์ที่ 3 </TD>
</TR>
<TR>
    <TD>แถวที่ 2 คอลัมน์ที่ 1 </TD>
```

```

<TD>แถวที่ 2 คอลัมน์ที่ 2 </TD>
<TD>แถวที่ 2 คอลัมน์ที่ 3 </TD>
</TR>";
echo "</TABLE>";
?>

```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 3.10 ผลลัพธ์จากการใช้แท็กเพื่อสร้างตาราง

ผลลัพธ์ของโปรแกรมนี้ เป็นการสร้างตารางที่มีขนาด 2 แถว 3 คอลัมน์ โดยเป็นการใช้แท็ก <TABLE>...</TABLE> เพื่อบอกจุดเริ่มต้นและสิ้นสุดของตาราง และใช้แท็ก <TR>...</TR> เพื่อสร้างแถวของตาราง สู้ท้ายใช้แท็ก <TD>...</TD> เพื่อสร้างคอลัมน์ของตาราง

สรุป

ในการเขียนโปรแกรมภาษา PHP จะต้องเขียนอยู่ภายในบล็อกคำสั่งของ PHP เสมอ โดยในภาษา PHP มีรูปแบบของบล็อกคำสั่ง 3 รูปแบบ คือ

```
<?...?>
```

```
<?php...?>
```

```
<script language='php'>...</script>
```

ในการแสดงผลรหัสภาษา PHP มีคำสั่งทั้งหมด 3 คำสั่ง คือ echo, print และ printf โดยคำสั่งทั้ง 3 คำสั่งจะมีรูปแบบ ดังนี้

```
echo "... ข้อมูลที่ต้องการแสดงผล ..."
```

```
print "... ข้อมูลที่ต้องการแสดงผล ..."
```

```
printf ("... ข้อมูลที่ต้องการแสดงผล ...")
```

ในการเขียนโปรแกรมเราสามารถใส่คำสั่งหมายเหตุ (Comment) เพื่อเขียนอธิบายโค้ดคำสั่งไว้สำหรับการทบทวนเมื่อกลับมาศึกษาอีกครั้ง โดยที่ข้อความใดๆ ที่ถูกหมายเหตุไว้ในกาแปลภาษาก็จะข้ามในส่วนนั้นไป รูปแบบคำสั่งหมายเหตุมี 3 รูปแบบ คือ // และ # ใช้หมายเหตุได้ครั้งละ 1 บรรทัด และ /*...*/ ใช้หมายเหตุได้ครั้งละหลายบรรทัด

ในการเขียนโปรแกรมภาษา PHP ให้มีประสิทธิภาพควรที่จะรู้จักแท็กพื้นฐานของ HTML ซึ่งจะช่วยให้การเขียนโปรแกรมได้รวดเร็วยิ่งขึ้น เพราะเวลาเราแสดงผลที่ได้จากโค้ดคำสั่ง PHP เราต้องแสดงออกมาในรูปแบบ HTML ถึงจะสามารถเปิดดูได้จากเบราว์เซอร์

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

แบบฝึกหัดบทที่ 3

1. ในการเขียนโปรแกรมภาษา PHP เราจะต้องเขียนอยู่ภายในบล็อกคำสั่งของ PHP ถ้ามว่า บล็อกคำสั่งของ PHP มีรูปแบบ อะไรบ้าง

.....

.....

.....

2. พิจารณาโปรแกรมภาษา PHP ต่อไปนี้

```
<?HTML>
<BODY>
  <?
    Good Morning
    My Name is David Beckham
  ?>
</BODY>
</HTML>
```

ถ้ามว่าโปรแกรมข้างต้นมีข้อผิดพลาดอะไรเกิดขึ้น และเพราะอะไร

.....

.....

.....

3. ให้เขียนโปรแกรมภาษา PHP โดยใช้คำสั่งแสดงผลเพื่อสร้างนามบัตรของตัวเองให้สวยงาม

.....

.....

.....

.....

4. การใช้คำสั่งหมายเหตุ (Comment) มีประโยชน์อย่างไร จงอธิบาย

.....

.....

.....

5. ให้เขียนโปรแกรมภาษา PHP โดยใช้แท็กคำสั่งของ HTML มาประกอบด้วย โดยให้มีลักษณะของผลลัพธ์ ดังนี้

คณะวิทยาศาสตร์	
สาขาวิชา	เว็บไซต์
• เคมี	http://chem.ubu.ac.th
• ฟิสิกส์	http://phy.ubu.ac.th
• วิทยาการคอมพิวเตอร์	http://com.ubu.ac.th
• เทคโนโลยีสารสนเทศ	http://inf.ubu.ac.th

หมายเหตุ: ตัวหนังสือในแต่ละบรรทัดต้องใส่สีที่ไม่ซ้ำกัน

สงวนลิขสิทธิ์
บริษัท ชัคเชส มีเดีย จำกัด

เรื่องของตัวแปรและค่าคงที่

ในการประมวลผลใดๆ ในคอมพิวเตอร์ต้องมีการจัดเก็บข้อมูลไว้ภายในหน่วยความจำของคอมพิวเตอร์ มิฉะนั้นแล้วการประมวลผลจะไม่สามารถเกิดขึ้นได้เลย เหตุผลง่ายๆ คือถ้ามีการประมวลผลเกิดขึ้นแล้วต้องการที่จะนำผลลัพธ์ (Output) จากการประมวลผลนี้จัดเก็บเอาไว้เพื่อที่จะนำไปเป็นข้อมูลนำเข้า (Input) ของการประมวลผลอื่นๆ ถัดไป การจัดเก็บข้อมูลดังกล่าวนี้ภายในคอมพิวเตอร์ต้องมีการจัดสรรพื้นที่ของหน่วยความจำเอาไว้เก็บข้อมูล ซึ่งการจัดสรรพื้นที่ในหน่วยความจำนี้เราสามารถกระทำได้โดยการใช้ตัวแปร (Variables) หรือ ค่าคงที่ (Constant) แต่ทั้งตัวแปร และค่าคงที่มีลักษณะและการทำงานที่แตกต่างกัน ในบทนี้เนื้อหาจะเน้นให้รู้จักการใช้งานของตัวแปร และค่าคงที่

จุดประสงค์
ทั่วไป

1. เพื่อให้รู้จักความหมายของตัวแปร และสามารถนำไปใช้งานได้
2. เพื่อให้ทราบวิธีการประกาศตัวแปร และกฎในการตั้งชื่อตัวแปรที่ถูกต้อง
3. เพื่อให้ทราบชนิดของข้อมูลต่างๆ แบ่งความแตกต่าง และนำไปใช้งานในแต่ละชนิดได้
4. เพื่อให้รู้จักตัวแปรประเภทอาร์เรย์
5. เพื่อให้ทราบถึงชื่อของตัวแปรประเภทอาร์เรย์
6. เพื่อให้ทราบถึงขั้นพื้นฐานของ PHP ที่เกี่ยวข้องกับตัวแปร และนำไปใช้งานได้
7. เพื่อให้รู้จักค่าคงที่ และเหตุผลในการใช้ค่าคงที่ในการเขียนโปรแกรม

จุดประสงค์
เชิงพฤติกรรม

1. บอกความหมายของตัวแปร
2. สามารถตั้งชื่อตัวแปรได้ถูกต้อง
3. บอกความหมายและประโยชน์ของตัวแปรอาร์เรย์
4. สามารถเขียนโปรแกรมโดยใช้ตัวแปรประเภทต่างๆ ได้
5. สามารถเขียนโปรแกรมโดยใช้ฟังก์ชันที่เกี่ยวข้องกับตัวแปรได้
6. สามารถเขียนโปรแกรมใช้ค่าคงที่ได้

เนื้อหาสาระ

1. ตัวแปร
2. การประกาศตัวแปร
3. ชนิดของข้อมูล
4. ตัวแปรอาร์เรย์
5. ฟังก์ชันที่เกี่ยวข้องกับตัวแปร
6. ค่าคงที่

4.1 ตัวแปร

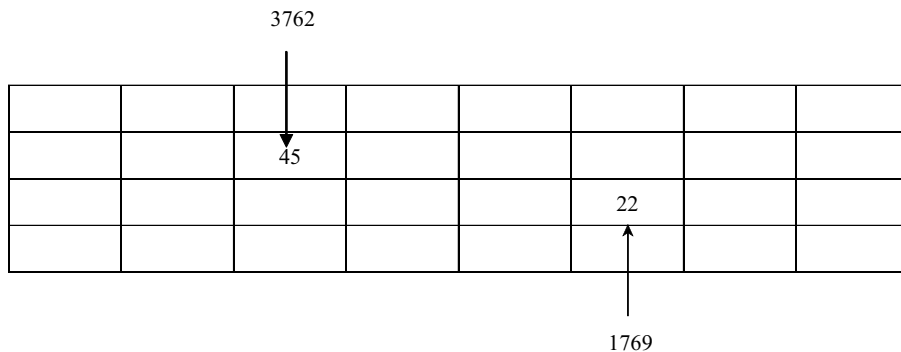
ตัวแปร (Variables) คือ ชื่อที่ตั้งขึ้นมาเพื่อใช้จัดเก็บข้อมูลภายในหน่วยความจำ (RAM) ของคอมพิวเตอร์ ตัวแปรสามารถจัดเก็บข้อมูลใดๆ ก็ได้ ไม่ว่าจะเป็นข้อมูลตัวเลข ตัวอักษร เครื่องหมาย เป็นต้น ลักษณะที่สำคัญของตัวแปร คือ

- ค่าที่จัดเก็บไว้ในตัวแปรใดๆ แล้ว เมื่อปิดโปรแกรม หรือเครื่องคอมพิวเตอร์ ข้อมูลนั้นจะถูกลบทิ้งไปจากหน่วยความจำ
- ตัวแปรสามารถที่จะเปลี่ยนแปลงค่าที่จัดเก็บไว้ได้ตลอดเวลา

เหตุผลที่บอกว่าทำไมต้องมีการใช้ตัวแปรภายในโปรแกรม พิจารณาโปรแกรม PHP ต่อไปนี้

```
<?
echo 40 + 5;
echo "<br>";
echo 30-8;
?>
```

จากโปรแกรมนี้คำสั่ง `echo 40 + 5;` จะมีค่าเท่ากับ 45 และคำสั่ง `echo 30-8;` จะมีค่าเท่ากับ 22 ถ้าต้องการนำค่าที่ได้จากทั้งสองคำสั่งนี้มาทำการลบกัน เราจะได้ได้อย่างไร ? วิธีการแก้ปัญหานี้ก็คือ เราต้องทำการจัดเก็บค่า 45 และ 22 ไว้ที่ใดที่หนึ่งเสียก่อนหลังจากนั้นก็ค่อยนำค่าทั้งสองมาทำการลบกัน ซึ่งที่ใดที่หนึ่งก็คือในหน่วยความจำนั่นเอง โดยปกติหน่วยความจำในคอมพิวเตอร์จะมีการแบ่งพื้นที่ในการจัดเก็บและมีการอ้างถึงพื้นที่ที่เก็บข้อมูลต่างๆ ด้วยหมายเลขที่อยู่ของหน่วยความจำ เช่น หมายเลข 3762 เป็นหมายเลขที่อยู่ที่จัดเก็บข้อมูลเลข 45 และหมายเลข 1769 เป็นหมายเลขที่อยู่ที่จัดเก็บข้อมูลเลข 22 ดังนั้นเราก็นำหมายเลขที่อยู่ 3762 และ 1769 มาทำการลบกัน



รูปที่ 4.1 ลักษณะของการจัดเก็บข้อมูลในหน่วยความจำของคอมพิวเตอร์

แต่ถ้าในทางปฏิบัติจริงๆ แล้วจะไม่สามารถทราบได้ว่าข้อมูลต่างๆ ถูกจัดเก็บไว้ที่หมายเลขที่อยู่อะไร และถ้าปรากฏว่าเราทราบว่าข้อมูลนั้นถูกจัดเก็บไว้ที่หมายเลขที่อยู่อะไรก็จะเป็นการลำบากในการจดจำหมายเลขที่อยู่ดังกล่าว ดังนั้น**ตัวแปร**จึงถูกนำมาใช้เพื่อจัดเก็บข้อมูลต่างๆ ในหน่วยความจำ โดยที่ตัวแปรนี้จะเป็นการตั้งชื่อขึ้นมาเพื่ออ้างอิงถึงข้อมูลต่างๆ ในหน่วยความจำนั่นเอง

4.2 การประกาศตัวแปร

จากหัวข้อที่ผ่านมาเราได้รู้จักความหมายของตัวแปรไปแล้ว ว่าเป็นชื่อที่ตั้งขึ้นมาเพื่ออ้างอิงถึงข้อมูลต่างๆ ในหน่วยความจำ ซึ่งชื่อของตัวแปรนี้ไม่ใช่เราจะตั้งชื่ออะไรก็ได้ แต่จะมีการกำหนดกฎเกณฑ์ในการตั้งชื่อตัวแปรดังนี้

- ชื่อตัวแปรต้องขึ้นต้นด้วยเครื่องหมาย \$ (dollar sign) เสมอ
- หลังเครื่องหมาย \$ จะตามด้วยชื่อตัวแปรโดยที่ตัวแรกของชื่อต้องเป็นตัวอักษร หรือเครื่องหมาย _ (underscore) เท่านั้น
- ตัวถัดมาของชื่อตัวแปรสามารถเป็นตัวอักษร หรือตัวเลข หรือเครื่องหมาย _ (underscore) เพื่อผสมเป็นชื่อได้
- ชื่อของตัวแปรมีคุณสมบัติเป็น Case-sensitive คือตัวอักษรตัวใหญ่และตัวเล็กถือว่าเป็นคนละตัวกัน

ถ้าเราเคยศึกษาโปรแกรมภาษาใดภาษาหนึ่งมาก่อน เช่น Basic, Pascal หรือ C ในการประกาศตัวแปรจะต้องทำการระบุชนิดของตัวแปรด้วย แต่ในภาษา PHP นี้ในการประกาศตัวแปรเราไม่ต้องระบุชนิดของตัวแปรเลย เราเพียงตั้งชื่อของตัวแปรตามกฎข้างต้นนั้น และนำชื่อตัวแปรนั้นไปกำหนดค่าต่างๆ ได้เลย พิจารณาตัวอย่างโปรแกรมต่อไปนี้

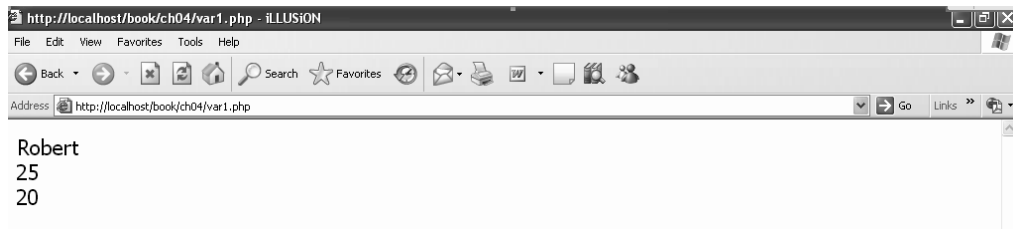
var1.php

```
<?
$name = "Robert";
$Age = 25;
$age = 20;
echo "$name <br>";
echo "$Age <br>";
echo "$age";
?>
```

บทที่

4

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 4.2 แสดงผลลัพธ์ของโปรแกรม var1.php

จากโปรแกรมนี้มีตัวแปรทั้งหมด 3 ตัว คือ \$name, \$Age และ \$age ซึ่งตัวแปรทั้งสามได้ทำการกำหนดค่าโดยใช้เครื่องหมายเท่ากับ และตัวแปรทั้งสามมีค่าเท่ากับ Robert, 25 และ 20 ตามลำดับ หลังจากนั้นก็ทำการใช้คำสั่ง echo เพื่อพิมพ์ค่าของตัวแปรทั้งหมดออกมา และจากผลลัพธ์ในตัวอย่างนี้มีจุดที่ควรสังเกต คือ

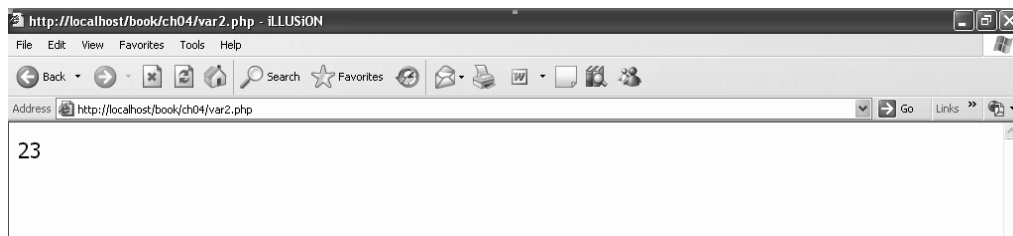
- ในการกำหนดค่าให้กับตัวแปรที่เป็นตัวอักษรจะต้องกำหนดอยู่ในเครื่องหมาย " (Double Quote)
- ชื่อตัวแปร \$Age และ \$age จะถือว่าเป็นตัวแปรคนละตัวกัน

และจากตัวอย่างที่กล่าวถึงในหัวข้อ 4.1 คือตัวอย่างการจัดเก็บค่าเช่น $40 + 5$ และ $30 - 8$ และให้นำค่าหลังจากการคำนวณได้ผลลัพธ์ทั้งสองมาทำการลบกัน ให้ทำการพิมพ์ผลลัพธ์ดังกล่าวออกมา จะสามารถเขียนโปรแกรมได้ ดังนี้

var2.php

```
<?
    $num1 = 40 + 5;
    $num2 = 30 - 8;
    $result = $num1 - $num2;
    echo "$result";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 4.3 แสดงผลลัพธ์ของโปรแกรม var2.php

จากโปรแกรมข้างต้นนี้ มีตัวแปรทั้งหมด 3 ตัวแปร คือ \$num1, \$num2 และ \$result โดยที่ตัวแปร \$num1 ทำการจับเก็บค่า 45 จากการบวกของ 40 และ 5 และตัวแปร \$num2 ทำการจับเก็บค่า 22 จากการลบของ 30 และ 8 หลังจากนั้นจะนำตัวแปรทั้งสองมาทำการลบกัน ผลลัพธ์ที่ได้จากการลบจะนำไปเก็บไว้ที่ตัวแปร \$result (45 - 22) ซึ่งตัวแปร \$result นี้จะจับเก็บตัวเลข 23 และสุดท้ายจะทำการพิมพ์ค่าที่ตัวแปร \$result ทำการจับเก็บไว้ออกมา ผลลัพธ์ตามรูปที่ 4.3

4.3 ชนิดของข้อมูล

ตามที่กล่าวในหัวข้อที่ผ่านมาเกี่ยวข้องกับเรื่องของตัวแปรว่าสามารถนำมาจัดเก็บข้อมูลใดๆ ก็ได้ในหน่วยความจำของคอมพิวเตอร์ ในที่นี้จะแบ่งชนิดของข้อมูลที่ตัวแปรสามารถจัดเก็บได้ให้ชัดเจน โดยสามารถแบ่งชนิดของข้อมูลได้เป็น 4 ชนิด คือ

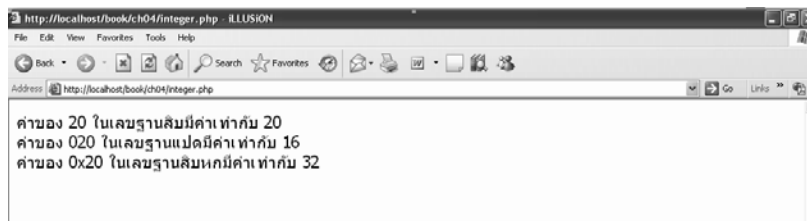
1. ข้อมูลชนิดจำนวนเต็ม (Integer)

ข้อมูลชนิดจำนวนเต็มเป็นตัวเลขจำนวนเต็มใดๆ ซึ่งสามารถแบ่งออกได้ 3 กลุ่ม คือ จำนวนเต็มบวก (1, 2, 3,...) จำนวนเต็มลบ (-1, -2, -3,...) และจำนวนเต็มศูนย์ นอกจากนี้เราสามารถนำตัวแปรมากำหนดค่าตัวเลขในเลขฐานแปด และฐานสิบหก และให้ทำการแสดงผลที่ในเลขฐานสิบได้ โดยการกำหนดค่าในรูปเลขฐานแปดให้เติม 0 (ศูนย์) นำหน้า และกรกำหนดค่าในรูปเลขฐานสิบหกให้เติม 0x (ศูนย์เอ็กซ์) นำหน้า พิจารณาตัวอย่างโปรแกรมต่อไปนี้

`integer.php`

```
<?
    $num1 = 20;
    $num2 = 020;
    $num3 = 0x20;
    echo "ค่าของ 20 ในเลขฐานสิบมีค่าเท่ากับ $num1 <br>";
    echo "ค่าของ 020 ในเลขฐานแปดมีค่าเท่ากับ $num2 <br>";
    echo "ค่าของ 0x20 ในเลขฐานสิบหกมีค่าเท่ากับ $num3 <br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.4 แสดงผลลัพธ์ของโปรแกรม `integer.php`

จากโปรแกรมข้างต้นมีตัวแปร 3 ตัวแปร คือ `$num1` เก็บค่าจำนวนเต็ม 20, `$num2` เก็บค่า 020 ซึ่งเป็นตัวเลขในรูปของเลขฐานแปด และ `$num3` เก็บค่า 0x20 ซึ่งเป็นตัวเลขในรูปของเลขฐานสิบหก หลังจากนั้นทำการใช้คำสั่ง `echo` เพื่อพิมพ์ผลลัพธ์ของตัวแปรทั้งสามโดยจะแสดงผลลัพธ์ในลักษณะของเลขจำนวนเต็มในรูปของเลขฐานสิบ คือ 20, 16 และ 32 ตามลำดับ

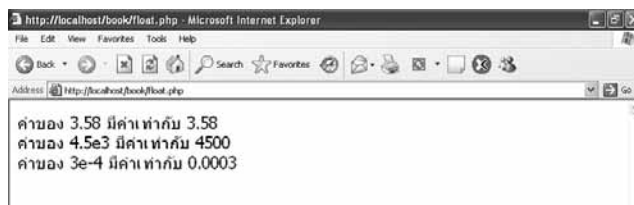
2. ข้อมูลชนิดจำนวนทศนิยม (Floating-Point Number)

ลักษณะของข้อมูลชนิดจำนวนทศนิยม คือตัวเลขที่มีจุดทศนิยมหรือเลขจำนวนเต็ม หรือเป็นตัวเลขแบบเศษส่วน หรือเป็นตัวเลขในลักษณะเลขยกกำลัง สามารถนำตัวอย่างโปรแกรมต่อไปนี้

`float.php`

```
<?
$num1 = 3.58;
$num2 = 4.5e3;
$num3 = 3e-4;
echo "ค่าของ 3.58 มีค่าเท่ากับ $num1 <br>";
echo "ค่าของ 4.5e3 มีค่าเท่ากับ $num2 <br>";
echo "ค่าของ 3e-4 มีค่าเท่ากับ $num3 <br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.5 แสดงผลลัพธ์ของโปรแกรม `float.php`

จากโปรแกรมข้างต้นมีตัวแปร 3 ตัวแปร คือ \$num1 เก็บค่าตัวเลขทศนิยม 3.58, \$num2 เก็บค่า 4.5e3 ซึ่งก็คือตัวเลข 4.5×10^3 และ \$num3 เก็บค่า $3e-4$ ซึ่งก็คือตัวเลข 3×10^{-4} หลังจากนั้นก็ทำการใช้คำสั่ง echo เพื่อพิมพ์ผลลัพธ์ของตัวแปรทั้งสามโดยจะแสดงผลคือ 3.58, 4500 และ 0.0003 ตามลำดับ

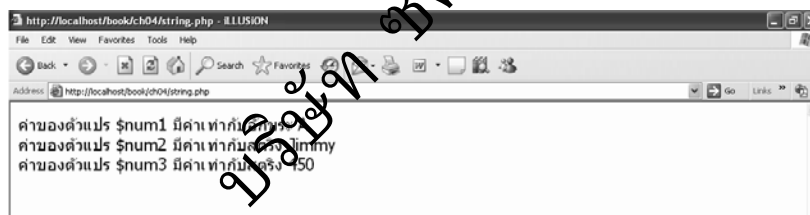
3. ข้อมูลชนิดสตริง (String)

ลักษณะข้อมูลชนิดสตริงคือข้อมูลที่นำตัวอักษรมาประกอบกันขึ้น โดยตัวอักษร 1 ตัวเราอาจจะเรียกว่าอักขระก็ได้ เมื่อนำหลายๆ อักขระมาประกอบกันแล้วเราก็จะเรียกว่าสตริงนั่นเอง ในการกำหนดค่าของอักขระและสตริงจะต้องใช้เครื่องหมาย " " (Double quote) ด้วยทุกครั้ง พิจารณาโปรแกรมต่อไปนี้

string.php

```
<?
$num1 = "A";
$num2 = "Jimmy";
$num3 = "450";
echo "ค่าของตัวแปร \ $num1 มีค่าเท่ากับอักขระ $num1 <br>";
echo "ค่าของตัวแปร \ $num2 มีค่าเท่ากับสตริง $num2 <br>";
echo "ค่าของตัวแปร \ $num3 มีค่าเท่ากับสตริง $num3 <br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.6 แสดงผลลัพธ์ของโปรแกรม string.php

จากโปรแกรมข้างต้นมีตัวแปร 3 ตัวแปร คือ \$num1 เก็บค่าอักขระ "A", \$num2 เก็บค่าสตริง "Jimmy" และ \$num3 เก็บค่าสตริง "450" หลังจากนั้นก็ทำการใช้คำสั่ง echo เพื่อพิมพ์ผลลัพธ์ของตัวแปรทั้งสามโดยจะแสดงผลคือ A, Jimmy และ 450 ตามลำดับ จากโปรแกรมนี้มีจุดที่น่าสังเกต คือ

- ตัวแปร \$num3 เก็บสตริง 450 ไม่ใช่ตัวเลข 450
- ในโปรแกรมมีการใช้รหัส Escape (\) นำหน้าชื่อตัวแปรทั้ง 3 ในบรรทัดคำสั่ง echo ซึ่งรหัส Escape นี้จะใช้เพื่อให้พิมพ์เครื่องหมาย \$ ออกมา และมีผลทำให้ num1, num2 และ num3 ที่ตามมามีค่าเป็นข้อความธรรมดาไม่ใช่ชื่อตัวแปร แต่ถ้าเราไม่ใช้ เมื่อคำสั่ง echo เจอเครื่องหมาย \$ จะเข้าใจว่าให้พิมพ์ค่าในตัวแปรนั้นออกมา

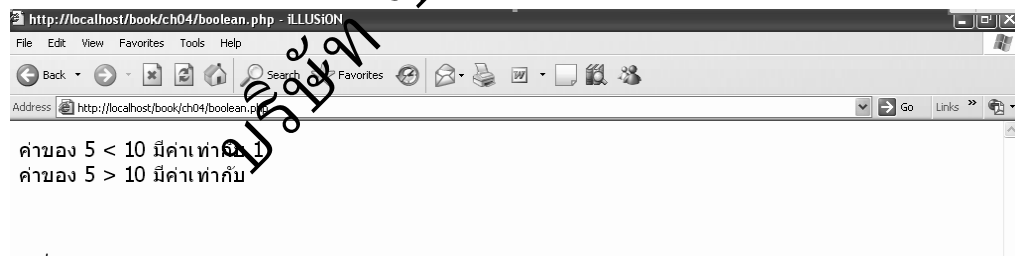
4. ข้อมูลชนิดตรรกะ (Boolean)

ลักษณะของข้อมูลชนิดตรรกะจะเก็บค่าเพื่อใช้สำหรับการตรวจสอบเงื่อนไข ซึ่งจะมีค่าที่เก็บเพียง 2 ค่าเท่านั้น คือเป็นค่าจริง (True) หรือเท็จ (False) พิจารณาโปรแกรมต่อไปนี้

boolean.php

```
<?
    $num1 = 5;
    $num2 = 10;
    $num3 = $num1 < $num2;
    $num4 = $num1 > $num2;
    echo "ค่าของ $num1 < $num2 มีค่าเท่ากับ $num3 <br>";
    echo "ค่าของ $num1 > $num2 มีค่าเท่ากับ $num4";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.7 แสดงผลลัพธ์ของโปรแกรม boolean.php

จากโปรแกรมข้างต้นนี้มีตัวแปรทั้งหมด 4 ตัวแปร คือ \$num1 เก็บตัวเลข 5, \$num2 เก็บตัวเลข 10 \$num3 เก็บค่าของการเปรียบเทียบว่า 5 น้อยกว่า 10 ใช่หรือไม่ซึ่งจะมีค่าเป็นจริง (true) และ \$num4 เก็บค่าของการเปรียบเทียบว่า 5 มากกว่า 10 ใช่หรือไม่ซึ่งจะมีค่าเป็นเท็จ (false) หลังจากนั้นก็ใช้คำสั่ง echo เพื่อพิมพ์ผลลัพธ์ของตัวแปร \$num3 และ \$num4 เราจะเห็นว่าตัวแปร \$num3 พิมพ์ผลลัพธ์คือ 1 ออกมา ซึ่งก็คือค่าจริงนั่นเอง และตัวแปร \$num4 พิมพ์ผลลัพธ์คือ ช่องว่างออกมาซึ่งก็คือค่าเท็จนั่นเอง

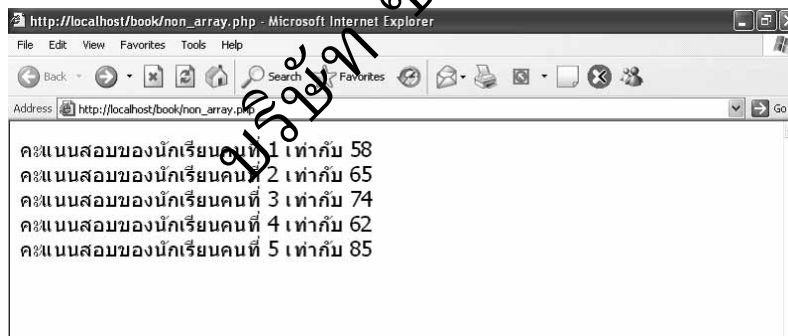
4.4 ตัวแปรอาร์เรย์

ก่อนที่จะกล่าวถึงความหมายและลักษณะของตัวแปรอาร์เรย์ พิจารณาโปรแกรม PHP ต่อไปนี้ โดยเป็นการเก็บข้อมูลคะแนนสอบของนักเรียนทั้งหมด 5 คน ไว้ในตัวแปรทั้งหมด 5 ตัวแปร คือ \$score1, \$score2, \$score3, \$score4 และ \$score5 หลังจากนั้นก็ใช้คำสั่ง echo เพื่อพิมพ์ผลลัพธ์ของคะแนนสอบทั้ง 5 จำนวนออกมา

non_array.php

```
<?
    $score1 = 58;
    $score2 = 65;
    $score3 = 74;
    $score4 = 62;
    $score5 = 85;
    echo "คะแนนสอบของนักเรียนคนที่ 1 เท่ากับ $score1 <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 2 เท่ากับ $score2 <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 3 เท่ากับ $score3 <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 4 เท่ากับ $score4 <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 5 เท่ากับ $score5 <br>";
?>
```

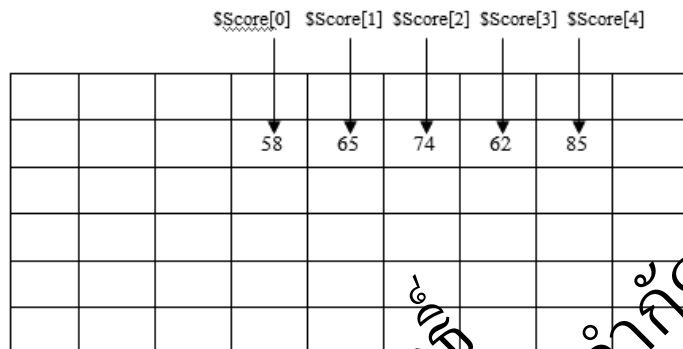
ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.8 แสดงผลลัพธ์ของโปรแกรม non_array.php

ลองคิดดูเล่นๆ ว่าถ้าเปลี่ยนจากจำนวนนักเรียน 5 คน เป็น 100 คน หรือมากกว่านี้ เราจะต้องแก้ไขโปรแกรมนี้อย่างไร คำตอบก็คือเราก็ต้องสร้างตัวแปรให้เท่ากับจำนวนนักเรียนเพื่อเก็บค่าคะแนนสอบ และเขียนคำสั่ง echo เท่ากับจำนวนนักเรียนนั่นเอง ลักษณะเช่นนี้คงเป็นการเขียนโปรแกรมที่ยาวมาก

วิธีการแก้ปัญหาดังกล่าวนี้ก็คือการจัดเก็บข้อมูลคะแนนสอบของนักเรียนด้วยตัวแปรอาร์เรย์แทนที่จะเป็นตัวแปรธรรมดาทั่วไป ดังนั้นความหมายของตัวแปรอาร์เรย์ ก็คือตัวแปรที่ใช้ชื่อตัวแปรเพียงชื่อเดียว แต่มีการจัดแบ่งพื้นที่การจัดเก็บข้อมูลในหน่วยความจำออกเป็นช่องๆ แต่ละช่องของอาร์เรย์จะมี index หรือหมายเลขที่ใช้ในการอ้างอิงถึงตำแหน่งที่อยู่ของช่องในการจัดเก็บนั้นเอง ลักษณะของการจัดเก็บข้อมูลของตัวแปรอาร์เรย์เป็นดังรูปต่อไปนี้



รูปที่ 4.9 ลักษณะของการจัดเก็บข้อมูลของตัวแปรอาร์เรย์

จากรูปที่ 4.9 ลักษณะของการจัดเก็บข้อมูลของตัวแปรอาร์เรย์อาจเปรียบเทียบกับกับการจัดเก็บข้อมูลในตาราง ที่มีการแบ่งพื้นที่ในการจัดเก็บข้อมูลแต่ละตัวไว้ในแต่ละช่องของตาราง โดยตัวแปรอาร์เรย์ที่ใช้ก็มีเพียงชื่อเดียวเท่านั้น คือ \$score แต่จะมีระบบหมายเลข index ไว้ในวงเล็บด้วย ซึ่งจากรูปนี้ความหมายก็คือ

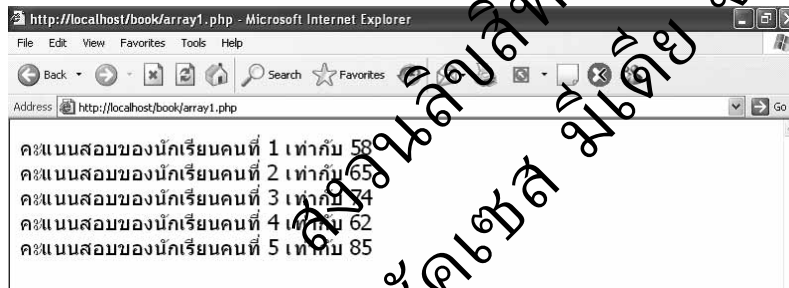
- เก็บค่าคะแนนสอบ 58 ไว้ในตัวแปรอาร์เรย์ \$score[0] (อาร์เรย์ช่องที่ 0)
- เก็บค่าคะแนนสอบ 65 ไว้ในตัวแปรอาร์เรย์ \$score[1] (อาร์เรย์ช่องที่ 1)
- เก็บค่าคะแนนสอบ 74 ไว้ในตัวแปรอาร์เรย์ \$score[2] (อาร์เรย์ช่องที่ 2)
- เก็บค่าคะแนนสอบ 62 ไว้ในตัวแปรอาร์เรย์ \$score[3] (อาร์เรย์ช่องที่ 3)
- เก็บค่าคะแนนสอบ 85 ไว้ในตัวแปรอาร์เรย์ \$score[4] (อาร์เรย์ช่องที่ 4)

จากโปรแกรม non_array.php เราจะสามารถทำการแก้ไขโดยใช้ตัวแปรอาร์เรย์เข้ามาจัดการเก็บข้อมูลคะแนนสอบของนักเรียนทั้ง 5 คน ได้ ดังนี้

array1.php

```
<?
    $score[0] = 58;
    $score[1] = 65;
    $score[2] = 74;
    $score[3] = 62;
    $score[4] = 85;
    echo "คะแนนสอบของนักเรียนคนที่ 1 เท่ากับ $score[0] <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 2 เท่ากับ $score[1] <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 3 เท่ากับ $score[2] <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 4 เท่ากับ $score[3] <br>";
    echo "คะแนนสอบของนักเรียนคนที่ 5 เท่ากับ $score[4] <br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.10 แสดงผลลัพธ์ของโปรแกรม array1.php

จากโปรแกรม array1.php และผลลัพธ์ที่ได้บางคนอาจจะคิดว่า การจัดเก็บข้อมูลไว้ในตัวแปรอาร์เรย์ไม่เห็นจะช่วยให้การเขียนโปรแกรมดีขึ้นกว่าการใช้ตัวแปรทั่วไปเลย ซึ่งตรงนี้ขอติตไว้ก่อน เนื่องจากการใช้ตัวแปรอาร์เรย์ให้มีประสิทธิภาพต้องใช้ร่วมกับคำสั่งทำซ้ำ และเมื่อได้กล่าวถึงเรื่องของคำสั่งทำซ้ำจะขอกลับมาอธิบายในเรื่องนี้อีกครั้ง อย่างไรก็ตามจากตัวอย่างโปรแกรมที่ใช้ตัวแปรอาร์เรย์นี้สิ่งที่เราเห็นได้อย่างชัดเจนถึงประโยชน์ของการใช้ตัวแปรอาร์เรย์ก็คือ เราทำการเก็บค่าคะแนนสอบ 5 จำนวน แต่เราใช้ตัวแปรเพียง 1 ตัวเท่านั้น

ในการกำหนดค่าให้กับตัวแปรอาร์เรย์นอกจากจะกำหนดตั้งตัวอย่างโปรแกรม array1.php แล้วยังสามารถทำการกำหนดในรูปแบบต่อไปนี้

รูปแบบ

```
ชื่อตัวแปรอาร์เรย์ = array("ค่าของอาร์เรย์ช่องที่ 0", "ค่าของอาร์เรย์ช่องที่ 1", ...)
```

บทที่

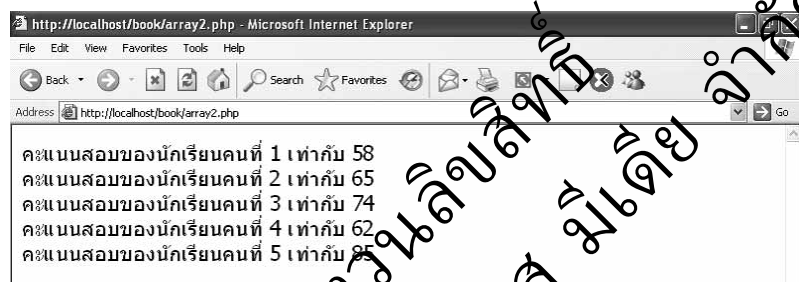
4

พิจารณาโปรแกรมต่อไปนี้

array2.php

```
<?
$score = array("58","65","74","62","85");
echo "คะแนนสอบของนักเรียนคนที่ 1 เท่ากับ $score[0] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 2 เท่ากับ $score[1] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 3 เท่ากับ $score[2] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 4 เท่ากับ $score[3] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 5 เท่ากับ $score[4] <br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.11 แสดงผลลัพธ์ของโปรแกรม array2.php

จากผลลัพธ์ของโปรแกรม array2.php ก็จะมีผลลัพธ์เช่นเดียวกับโปรแกรม array1.php แต่ในโปรแกรมนี้ใช้รูปแบบการกำหนดค่าให้กับตัวแปรอาร์เรย์ ทำให้โปรแกรมที่เขียนขึ้นสั้นลงกว่าเดิม

ข้อดีของการใช้ตัวแปรอาร์เรย์ในภาษา PHP ซึ่งเหนือกว่าโปรแกรมภาษาอื่นก็คือ การกำหนด index หรือหมายเลขอ้างอิงของช่องอาร์เรย์นั้นไม่จำเป็นต้องกำหนดเป็นหมายเลขก็ได้ เราสามารถกำหนด index เป็นข้อความตัวอักษรก็ได้ โดยยึดรูปแบบการกำหนดค่า ดังนี้

รูปแบบ

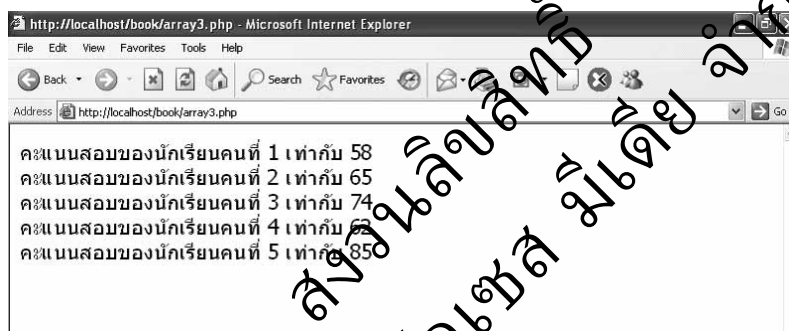
```
ชื่อตัวแปรอาร์เรย์ = array("index" => "ค่าของอาร์เรย์", ...)
```

พิจารณาโปรแกรมต่อไปนี้ โดยเป็นการกำหนดค่าให้กับตัวแปรอาร์เรย์ตามรูปแบบข้างต้น

array3.php

```
<?
$score= array("student1"=>"58","student2"=>"65",
              "student3"=>"74","student4"=>"62","students"=>"85");
echo "คะแนนสอบของนักเรียนคนที่ 1 เท่ากับ $score[student1] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 2 เท่ากับ $score[student2] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 3 เท่ากับ $score[student3] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 4 เท่ากับ $score[student4] <br>";
echo "คะแนนสอบของนักเรียนคนที่ 5 เท่ากับ $score[students] <br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.12 แสดงผลลัพธ์ของโปรแกรม array3.php

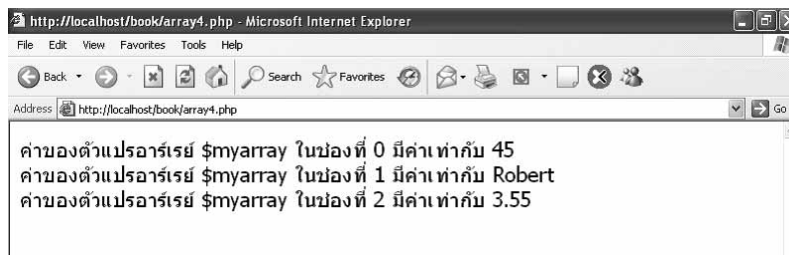
จากโปรแกรม array3.php ได้ทำการกำหนดค่า index ทั้ง 5 ช่องของอาร์เรย์เป็นข้อความว่า student1, student2, student3, student4 และ students ตามลำดับ ซึ่งผลลัพธ์ที่ประมวลผลออกมา ก็ยังได้เหมือนเดิม

นอกจากข้อดีในเรื่องของการกำหนด index เป็นตัวอักษรหรือข้อความได้แล้ว ตัวแปรอาร์เรย์ยังสามารถกำหนดค่าในแต่ละช่องด้วยข้อมูลต่างชนิดกันได้ พิจารณาโปรแกรมต่อไปนี้

array4.php

```
<?
$myarray = array("45","Robert", "3.55");
echo "ค่าของตัวแปรอาร์เรย์ \$myarray ในช่องที่ 0 มีค่าเท่ากับ $myarray[0] <br>";
echo "ค่าของตัวแปรอาร์เรย์ \$myarray ในช่องที่ 1 มีค่าเท่ากับ $myarray[1] <br>";
echo "ค่าของตัวแปรอาร์เรย์ \$myarray ในช่องที่ 2 มีค่าเท่ากับ $myarray[2]";
?>
```

ผลลัพธ์ของโปรแกรมเป็นดังนี้



รูปที่ 4.13 แสดงผลลัพธ์ของโปรแกรม array4.php

จากโปรแกรม array4.php นี้ได้สร้างตัวแปรอาร์เรย์มีชื่อว่า \$myarray และได้ทำการกำหนดค่าให้กับตัวแปรอาร์เรย์นี้ในแต่ละช่อง เริ่มจากช่องที่ 0 มีค่าเท่ากับจำนวนเต็ม 45, ช่องที่ 1 มีค่าเท่ากับสตริง Robert และช่องที่ 2 มีค่าเท่ากับเลขทศนิยม 3.55 จะเห็นว่าเป็นการใช้ชื่อตัวแปรอาร์เรย์ชื่อเดียวกัน แต่เก็บข้อมูลในแต่ละช่องต่างชนิดกัน ซึ่งโปรแกรมภาษาอื่นๆ ไม่สามารถทำได้ เพราะตัวแปรอาร์เรย์ชื่อใดชื่อหนึ่งค่าในแต่ละช่องต้องกำหนดข้อมูลเป็นชนิดเดียวกัน

4.5 ฟังก์ชันที่เกี่ยวข้องกับตัวแปร

โปรแกรมภาษา PHP ได้จัดเตรียมฟังก์ชันที่เกี่ยวข้องกับตัวแปรให้ใช้งานด้วยกันหลายฟังก์ชัน ซึ่งเราไม่ต้องทำการเขียนโปรแกรมใดๆ เลย เพียงแค่รู้จักและระมัดระวังการประมวลผลของฟังก์ชันนั้นๆ และเรียกใช้งานให้ถูกต้องตามรูปแบบเท่านั้น

• **ฟังก์ชัน gettype (ตัวแปร)** เป็นฟังก์ชันที่ใช้ตรวจสอบชนิดข้อมูลของตัวแปร ซึ่งอาจจะคืนค่าชนิดของตัวแปรต่างๆ ต่อไปนี้กลับมา

integer	ชนิดข้อมูลตัวเลขแบบจำนวนเต็ม
double	ชนิดข้อมูลตัวเลขแบบทศนิยม
string	ชนิดข้อมูลตัวอักษร ข้อความ
array	ชนิดข้อมูลแบบอาร์เรย์
object	ชนิดข้อมูลแบบออบเจกต์
class	ชนิดข้อมูลแบบคลาส
unknown type	ไม่สามารถระบุชนิดตัวแปรได้

พิจารณาตัวอย่างโปรแกรมต่อไปนี้

gettype.php

```
<?
    $var = 15;
    echo "$var เป็นตัวแปรชนิด ".gettype($var)."<br>";
    $var = 3.48;
    echo "$var เป็นตัวแปรชนิด ".gettype($var)."<br>";
    $var = "Hello";
    echo "$var เป็นตัวแปรชนิด ".gettype($var)."<br>";
    $var = array("10","20");
    echo "\$var = array(\"10\",\"20\")." เป็นตัวแปรชนิด ".gettype($var)."<br>";
?>
```

ผลลัพธ์ของโปรแกรมเป็น ดังนี้



รูปที่ 4.14 แสดงผลลัพธ์ของโปรแกรม gettype.php

• ฟังก์ชัน **empty** (ตัวแปร) เป็นฟังก์ชันที่ใช้ตรวจสอบตัวแปรว่าเก็บข้อมูลไว้หรือไม่ โดยถ้าตัวแปรเก็บข้อมูลไว้จะคืนค่าเท็จ (false) ซึ่งคืนค่าเท็จนี้จะป้อนช่องว่าง แต่ถ้าไม่ได้เก็บข้อมูลไว้จะคืนค่าจริง (true) ซึ่งค่าจริงนี้จะป้อนหมายเลข 1 ไว้ พิจารณาโปรแกรมต่อไปนี้

empty.php

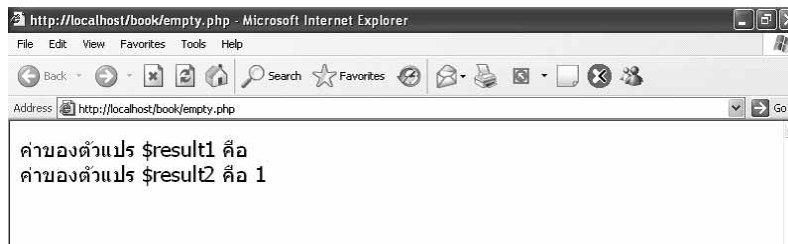
```
<?
    $val1 = 50;
    $result1 = empty($val1);
    echo "ค่าของตัวแปร \$result1 คือ $result1 <br>";

    $val2 = null;
    $result2 = empty($val2);
    echo "ค่าของตัวแปร \$result2 คือ $result2";
?>
```

บทที่

4

ผลลัพธ์ของโปรแกรมเป็น ดังนี้



รูปที่ 4.15 แสดงผลลัพธ์ของโปรแกรม `empty.php`

- ฟังก์ชัน **isset (ตัวแปร)** ฟังก์ชันนี้ใช้สำหรับตรวจสอบว่ามีตัวแปรตามชื่อที่ระบุไว้หรือไม่ โดยถ้ามีตัวแปรนั้นจะคืนค่าจริง (true) แต่ถ้าไม่มีจะคืนค่าเท็จ (false)

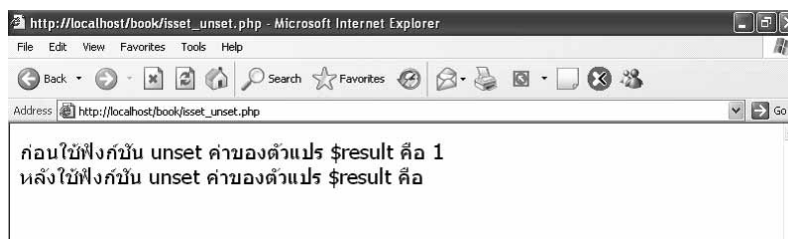
- ฟังก์ชัน **unset (ตัวแปร)** ฟังก์ชันนี้ใช้ทำลายตัวแปรและทำการคืนพื้นที่ในหน่วยความจำ ดังนั้นหลังจากการใช้ฟังก์ชันนี้กับตัวแปรใดๆ แล้วตัวแปรนั้นก็ถือว่าไม่มีตัวตนคือไม่มีตัวแปรนั้นอีกแล้ว

พิจารณาโปรแกรมต่อไปนี้ ซึ่งเป็นการใช้ฟังก์ชัน `isset (ตัวแปร)` และฟังก์ชัน `unset (ตัวแปร)`

`isset_unset.php`

```
<?
    $val = 50;
    $result = isset($val);
    echo "ก่อนใช้ฟังก์ชัน unset ค่าของตัวแปร \$result คือ $result <br>";
    unset($val);
    $result = isset($val);
    echo "หลังใช้ฟังก์ชัน unset ค่าของตัวแปร \$result คือ $result";
?>
```

ผลลัพธ์ของโปรแกรมเป็น ดังนี้



รูปที่ 4.16 แสดงผลลัพธ์ของโปรแกรม `isset_unset.php`

4.6 ค่าคงที่

ค่าคงที่ (Constant) คือชื่อที่กำหนดขึ้นมาเพื่อเก็บค่าใดๆ เอาไว้ โดยปกติเราจะนำค่าคงที่ที่เก็บค่าใดๆ ที่ไม่ค่อยมีการเปลี่ยนแปลง รูปแบบการใช้งานคำสั่งค่าคงที่ คือ

รูปแบบ

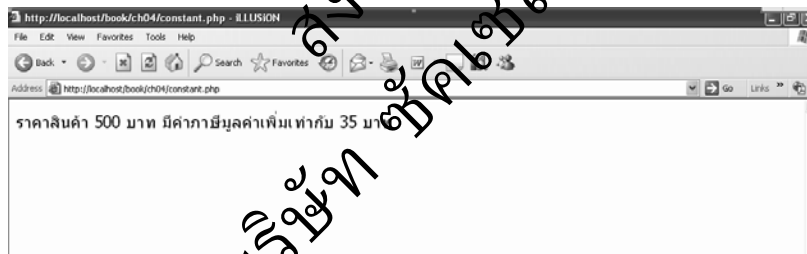
```
define(ชื่อค่าคงที่, ค่า)
```

พิจารณาโปรแกรมต่อไปนี้

constant.php

```
<?
    define("vat", 0.07);
    $price = 500;
    $tax = $price*vat;
    echo "ราคาสินค้า $price บาท มีค่าภาษีมูลค่าเพิ่มเท่ากับ $tax บาท";
?>
```

ผลลัพธ์ของโปรแกรมเป็น ดังนี้



รูปที่ 4.17 แสดงผลลัพธ์ของโปรแกรม constant.php

สรุป

ตัวแปร (Variables) คือ ชื่อที่ตั้งขึ้นมาเพื่อใช้จัดเก็บข้อมูลภายในหน่วยความจำ (RAM) ของคอมพิวเตอร์ ตัวแปรสามารถจัดเก็บข้อมูลใดๆ ก็ได้ ไม่ว่าจะเป็นข้อมูลตัวเลข ตัวอักษร เครื่องหมาย เป็นต้น และชื่อของตัวแปรจะต้องยึดกฎเกณฑ์ในการตั้งชื่อ คือ ชื่อตัวแปรต้องขึ้นต้นด้วยเครื่องหมาย \$ (dollar sign), หลังเครื่องหมาย \$ จะตามด้วยชื่อตัวแปรโดยที่ตัวแรกของชื่อต้องเป็นตัวอักษร หรือเครื่องหมาย _ (underscore) เท่านั้น, ตัวถัดมาของชื่อตัวแปรสามารถเป็นตัวอักษร หรือตัวเลข หรือเครื่องหมาย _ (underscore) เพื่อผสมเป็นชื่อได้ และชื่อของตัวแปรมีคุณสมบัติเป็น case-sensitive

บทที่

4

ชนิดของตัวแปรทั้งหมด 4 ชนิด คือ ข้อมูลชนิดจำนวนเต็ม (Integer), ข้อมูลชนิดจำนวนทศนิยม (Floating-Point Number), ข้อมูลชนิดสตริง (String) และข้อมูลชนิดตรรกะ (Boolean)

ชนิดของตัวแปรพิเศษอีกชนิดหนึ่งคือ ตัวแปรอาร์เรย์ โดยเป็นตัวแปรที่จัดเก็บข้อมูลได้หลายๆ จำนวน โดยใช้ชื่อตัวแปรเพียงชื่อเดียวเท่านั้น ทำให้ไม่ต้องประกาศตัวแปรจำนวนมาก และการเขียนโปรแกรมก็จะง่ายยิ่งขึ้น

ฟังก์ชันที่เกี่ยวข้องกับตัวแปรในนี้มี 4 ฟังก์ชัน คือ ฟังก์ชัน `gettype()`, `empty()`, `isset()`, `unset()`

ค่าคงที่ (Constant) คือชื่อที่กำหนดขึ้นมาเพื่อเก็บค่าใดๆ เอาไว้ โดยปกติเราจะนำค่าคงที่มาเก็บค่าใดๆ ที่ไม่ค่อยมีการเปลี่ยนแปลง

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

แบบฝึกหัดบทที่ 4

1. พิจารณาชื่อตัวแปรต่อไปนี้ ชื่อตัวแปรใดที่ตั้งได้ถูกต้องตามกฎเกณฑ์ในการตั้งชื่อ และชื่อใดที่ตั้งได้ ผิด ถ้าผิดให้บอกเหตุผลด้วย

1.1) \$yenta4

.....
.....

1.2) \$hotmail.com

.....
.....

1.3) \$4u

.....
.....

1.4) \$Good morning

.....
.....

1.5) \$MSU@Radio

.....
.....

2. ให้เขียนโปรแกรมภาษา PHP ในการคำนวณหาค่าคะแนนเฉลี่ยของการสอบครั้งหนึ่งซึ่งมีนักเรียน 5 คน โดยให้สร้างตัวแปรทั้งหมด 7 ตัว ดังนี้

- ตัวแปร 5 ตัวแปรแรกใช้เก็บค่าคะแนนสอบของนักเรียนทั้ง 5 คน
- ตัวแปรตัวที่ 6 ใช้เก็บค่าผลรวมของคะแนนสอบของนักเรียนทั้ง 5 คน
- ตัวแปรตัวที่ 7 ใช้เก็บค่าคะแนนเฉลี่ยของคะแนนสอบของนักเรียนทั้ง 5 คน

.....
.....
.....
.....
.....

สงวนลิขสิทธิ์
สำนักพิมพ์ ชัคเชส มีเดีย จำกัด

3. ให้เขียนโปรแกรมภาษา PHP โดยพิมพ์ข้อมูลประวัติส่วนตัวโดยให้กำหนดตัวแปรต่างๆ ดังนี้

- ชื่อ-สกุล ให้เก็บไว้ในตัวแปรชนิดสตริง
- เพศ ให้เก็บไว้ในตัวแปรสตริงในลักษณะเป็นอักขระ (M คือชาย และ F คือหญิง)
- น้ำหนัก ให้เก็บไว้ในตัวแปรชนิดทศนิยม
- ส่วนสูง ให้เก็บไว้ในตัวแปรชนิดทศนิยม
- อายุ ให้เก็บไว้ในตัวแปรชนิดเลขจำนวนเต็ม

.....

.....

.....

.....

4. ให้อธิบายลักษณะของตัวแปรอาร์เรย์ และประโยชน์ของการนำตัวแปรอาร์เรย์มาใช้งาน

.....

.....

.....

.....

5. ให้เขียนโปรแกรมภาษา PHP โดยสร้างตัวแปร 2 ตัว คือตัวแปรอาร์เรย์ตัวที่ 1 ให้เก็บชื่อนักเรียน 5 คน และตัวแปรอาร์เรย์ตัวที่ 2 ให้เก็บอายุของนักเรียนทั้ง 5 คน หลังจากนั้นให้พิมพ์ผลลัพธ์บอกว่านักเรียนแต่ละคนมีอายุเท่าไร

.....

.....

.....

.....

ตัวดำเนินการและนิพจน์ใน PHP

ในการเขียนโปรแกรมไม่ว่าโปรแกรมภาษาใดก็ตามเราหลีกเลี่ยงไม่ได้ที่จะมีการคำนวณหาผลลัพธ์ต่างๆ เช่นการคำนวณหาค่าส่วนลดจากการซื้อสินค้า, การคำนวณหาราคาสินค้าสุทธิ, การคำนวณหาค่าล่วงเวลาจากการทำงานของพนักงาน เป็นต้น ซึ่งการที่จะทำการคำนวณหาค่าของผลลัพธ์ต่างๆ ก็จะต้องมีความรู้ในเรื่องของตัวดำเนินการ (Operator) และนิพจน์ (Expression) ถ้าเปรียบเทียบให้เห็นภาพชัดเจนกับวิชาคณิตศาสตร์ คำว่านิพจน์ก็คือสมการ หรืออสมการ ส่วนคำว่า ตัวดำเนินการก็เปรียบเทียบกับเครื่องหมายต่างๆ ที่ใช้สร้างสมการ หรืออสมการขึ้นมานั่นเอง เนื้อหาในบทนี้จะเน้นให้รู้จักการใช้ตัวดำเนินการต่างๆ เพื่อสร้างนิพจน์ขึ้นมา ซึ่งจะทำให้เราสามารถเขียนโปรแกรมเพื่อทำการคำนวณหาค่าผลลัพธ์บางอย่างตามที่ต้องการได้ นอกจากนั้นแล้วยังเป็นพื้นฐานที่สำคัญในการสร้างนิพจน์แบบเงื่อนไขเพื่อใช้ในการคำสั่งการตรวจสอบเงื่อนไข และคำสั่งการทำซ้ำซึ่งจะได้กล่าวถึงในบทถัดไป

จุดประสงค์ทั่วไป

1. เพื่อให้ทราบความหมายของตัวดำเนินการและประเภทของตัวดำเนินการ
2. เพื่อศึกษาและรู้จักตัวดำเนินการทางคณิตศาสตร์ ตลอดจนนำไปใช้งานได้
3. เพื่อศึกษาและรู้จักตัวดำเนินการเปรียบเทียบ ตลอดจนนำไปใช้งานได้ถูกต้อง
4. เพื่อศึกษาและรู้จักตัวดำเนินการทางตรรกะ ตลอดจนนำไปใช้งานได้ถูกต้อง
5. เพื่อศึกษาและรู้จักตัวดำเนินการระดับบิต ตลอดจนนำไปใช้งานได้ถูกต้อง
6. เพื่อให้ทราบความหมายของนิพจน์ และสร้างนิพจน์ได้อย่างถูกต้อง
7. เพื่อให้ทราบลำดับในการประมวลผล และคำนวณผลลัพธ์ของนิพจน์ได้

จุดประสงค์เชิงพฤติกรรม

1. บอกความหมายของตัวดำเนินการ และประเภทของตัวดำเนินการ
2. บอกความหมายของนิพจน์
3. สามารถสร้างนิพจน์ได้ถูกต้อง
4. สามารถเขียนโปรแกรมโดยใช้ตัวดำเนินการประเภทต่างๆ ได้
5. สามารถเขียนโปรแกรมโดยสร้างนิพจน์ได้ถูกต้อง

เนื้อหาสาระ

1. ความหมายของตัวดำเนินการและประเภทของตัวดำเนินการ
2. ตัวดำเนินการทางคณิตศาสตร์
3. ตัวดำเนินการเปรียบเทียบ
4. ตัวดำเนินการทางตรรกะ
5. ตัวดำเนินการสตริง
6. ตัวดำเนินการระดับบิต
7. ความหมายของนิพจน์และลำดับการประมวลผลของนิพจน์

5.1 ความหมายของตัวดำเนินการและประเภทของตัวดำเนินการ

ตัวดำเนินการ (Operator) คือ เครื่องหมายต่างๆ ที่ใช้ในการคำนวณ หรือการทำฟังก์ชันทางคณิตศาสตร์ ตัวดำเนินการบางตัวให้ผลลัพธ์เป็นตัวเลขใดๆ ตัวดำเนินการบางตัวให้ผลลัพธ์เป็นค่าความจริงเป็นจริง (True) หรือเท็จ (False) เท่านั้น ตัวดำเนินการบางตัวก็ใช้จัดการเฉพาะข้อมูลที่เป็นสตริงหรือเป็นอักขระเท่านั้น และตัวดำเนินการบางตัวก็ให้ผลลัพธ์เป็นเลขฐาน 2 ซึ่งเราสามารถแบ่งประเภทของตัวดำเนินการในภาษา PHP ได้เป็น 5 ประเภท ดังนี้

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)
- ตัวดำเนินการเปรียบเทียบ (Comparison Operator)
- ตัวดำเนินการทางตรรกะ (Logical Operator)
- ตัวดำเนินการสตริง (String Operator)
- ตัวดำเนินการระดับบิต (Bitwise Operator)

ตัวดำเนินการประเภทที่ 1-4 เป็นตัวดำเนินการที่เราจะเห็นและใช้บ่อยที่สุด และบางครั้งก็จะมีการใช้ตัวดำเนินการทั้ง 4 ประเภทนี้รวมกัน ส่วนตัวดำเนินการประเภทที่ 5 อาจจะไม่ค่อยเห็นการใช้งานบ่อยนัก ในหัวข้อถัดไปจะได้อธิบายถึงตัวดำเนินการในแต่ละประเภทและการใช้งานต่างๆ

5.2 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์เป็นตัวดำเนินการในกลุ่มที่เกี่ยวข้องกับการคำนวณหาผลลัพธ์ต่างๆ ทางคณิตศาสตร์ ผลลัพธ์จากการใช้ตัวดำเนินการทางคณิตศาสตร์นี้จะได้ผลลัพธ์เป็นตัวเลขใดๆ พิจารณาตัวดำเนินการในกลุ่มนี้แต่ละตัวพร้อมทั้งตัวอย่างประกอบจากตารางสรุปต่อไปนี้ (กำหนดให้ตัวแปร $a = 10$ และ $b = 3$)

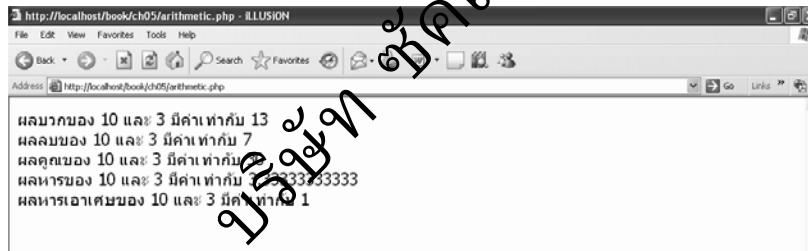
ตัวดำเนินการ	สัญลักษณ์	ตัวอย่าง
บวก	+	$\$a + \b (ความหมายก็คือ $10 + 3$ ซึ่งมีค่าเท่ากับ 13)
ลบ	-	$\$a - \b (ความหมายก็คือ $10 - 3$ ซึ่งมีค่าเท่ากับ 7)
คูณ	*	$\$a * \b (ความหมายก็คือ $10 * 3$ ซึ่งมีค่าเท่ากับ 30)
หาร	/	$\$a / \b (ความหมายก็คือ $10 / 3$ ซึ่งมีค่าเท่ากับ 3.33)
หารเอาเศษ	%	$\$a \% \b (ความหมายก็คือ $10 \% 3$ ซึ่งมีค่าเท่ากับ 1)
เพิ่มค่าขึ้น 1	++	$\$a++$ (ความหมายก็คือ เพิ่มค่าให้กับตัวแปร $\$a$ ขึ้น 1 ซึ่งมีค่าเท่ากับ 11)
ลดค่าลง 1	--	$\$b--$ (ความหมายก็คือ ลดค่าให้กับตัวแปร $\$b$ ลง 1 ซึ่งมีค่าเท่ากับ 2)

จากตารางสรุปข้างต้น เราจะมาลองเขียนโปรแกรมเพื่อทดสอบการใช้งานตัวดำเนินการทางคณิตศาสตร์ดังต่อไปนี้

arithmetic.php

```
<?
    $a = 10;
    $b = 3;
    $add = $a + $b;
    $minus = $a - $b;
    $multiply = $a * $b;
    $divide = $a / $b;
    $modulus = $a % $b;
    echo "ผลบวกของ $a และ $b มีค่าเท่ากับ $add <br>";
    echo "ผลลบของ $a และ $b มีค่าเท่ากับ $minus <br>";
    echo "ผลคูณของ $a และ $b มีค่าเท่ากับ $multiply <br>";
    echo "ผลหารของ $a และ $b มีค่าเท่ากับ $divide <br>";
    echo "ผลหารเอาเศษของ $a และ $b มีค่าเท่ากับ $modulus <br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็นดังนี้



รูปที่ 5.1 แสดงผลลัพธ์ของโปรแกรม arithmetic.php

5.3 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการเปรียบเทียบเป็นตัวดำเนินการที่ใช้สำหรับการเปรียบเทียบตัวเลข หรือตัวแปร โดยผลลัพธ์จากการใช้ตัวดำเนินการเปรียบเทียบนี้จะได้ผลลัพธ์ที่มีค่าความจริงเป็นจริง (True) หรือเท็จ (False)

พิจารณาตัวดำเนินการในกลุ่มนี้แต่ละตัวพร้อมทั้งตัวอย่างประกอบจากตารางสรุปต่อไปนี้ (กำหนดให้ตัวแปร \$a = 10, \$b = 3 และ \$c = 10)

ตัวดำเนินการ	สัญลักษณ์	ตัวอย่าง
เท่ากับ	=	<code>\$a == \$b</code> (ได้ผลลัพธ์เป็นเท็จ เพราะ 10 ไม่เท่ากับ 3)
ไม่เท่ากับ	!= หรือ <>	<code>\$a != \$b</code> (ได้ผลลัพธ์เป็นจริง เพราะ 10 ไม่เท่ากับ 3)
น้อยกว่า	<	<code>\$a < \$b</code> (ได้ผลลัพธ์เป็นเท็จ เพราะ 10 มากกว่า 3)
มากกว่า	>	<code>\$a > \$b</code> (ได้ผลลัพธ์เป็นจริง เพราะ 10 มากกว่า 3)
น้อยกว่าหรือเท่ากับ	<=	<code>\$a <= \$c</code> (ได้ผลลัพธ์เป็นจริง เพราะ 10 น้อยกว่าหรือเท่ากับ 10)
มากกว่าหรือเท่ากับ	>=	<code>\$a >= \$c</code> (ได้ผลลัพธ์เป็นจริง เพราะ 10 มากกว่าหรือเท่ากับ 10)

พิจารณาโปรแกรมต่อไปนี้เพื่อทดสอบการใช้งานตัวดำเนินการทางคณิตศาสตร์

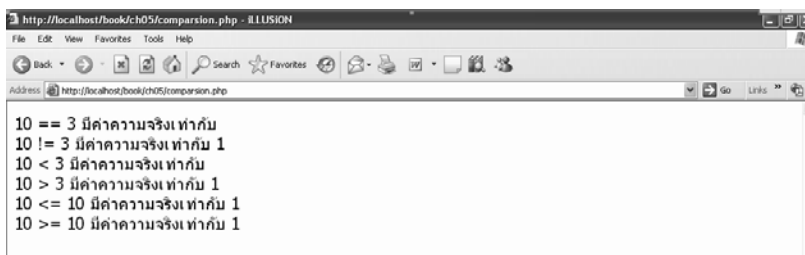
comparison.php

```

<?
    $a = 10;
    $b = 3;
    $c = 10;
    $equal = $a == $b;
    $not_equal = $a != $b;
    $less_than = $a < $b;
    $greater_than = $a > $b;
    $less_than_or_equal = $a <= $c;
    $greater_than_or_equal = $a >= $c;
    echo "$a == $b มีค่าความจริงเท่ากับ $equal <br>";
    echo "$a != $b มีค่าความจริงเท่ากับ $not_equal <br>";
    echo "$a < $b มีค่าความจริงเท่ากับ $less_than <br>";
    echo "$a > $b มีค่าความจริงเท่ากับ $greater_than <br>";
    echo "$a <= $c มีค่าความจริงเท่ากับ $less_than_or_equal <br>";
    echo "$a >= $c มีค่าความจริงเท่ากับ $greater_than_or_equal <br>";
?>

```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 5.2 แสดงผลลัพธ์ของโปรแกรม comparison.php

จากผลลัพธ์ข้างต้น ค่าความจริงเท่ากับ 1 คือค่าความจริงที่เป็นจริง (True) และค่าความจริงเป็นช่องว่างก็คือค่าความจริงที่เป็นเท็จ (False)

5.4 ตัวดำเนินการทางตรรกะ

ตัวดำเนินการทางตรรกะเป็นตัวดำเนินการที่ใช้ทำการเชื่อมค่าความจริงต่างๆ โดยเฉพาะการนำตัวดำเนินการทางตรรกะนี้ไปประยุกต์ใช้กับคำสั่งตรวจสอบเงื่อนไข ผลลัพธ์จากการใช้ตัวดำเนินการทางตรรกะจะได้ผลลัพธ์ที่มีค่าความจริงเป็นจริง (True) หรือเท็จ (False) ตัวดำเนินการทางตรรกะมีทั้งหมด 4 ตัว คือ ตัวดำเนินการ และ (And), ตัวดำเนินการหรือ (Or), ตัวดำเนินการนิเสธ (Not) และตัวดำเนินการ xor (Exclusive Or) ก่อนอื่นให้ทำการพิจารณาผลจากการใช้ตัวดำเนินการทั้ง 4 นี้ ดังนี้

ตัวกระทำที่ 1	ตัวกระทำที่ 2	And	Or	xor
True	True	True	True	False
True	False	False	True	True
False	True	False	True	True
False	False	False	False	False

ตัวกระทำ	Not
True	False
False	True

พิจารณาตัวดำเนินการในกลุ่มนี้แต่ละตัวพร้อมทั้งตัวอย่างประกอบจากตารางสรุปต่อไปนี้ (กำหนดให้ตัวแปร \$a = 10\$, \$b = 3\$, \$c = 5\$ และ \$d = 7\$)

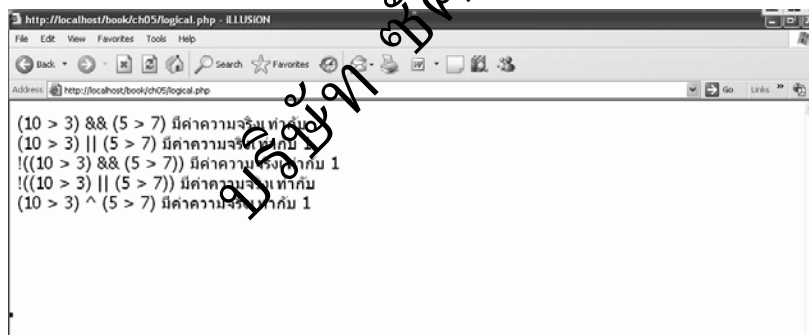
ตัวดำเนินการ	สัญลักษณ์	ตัวอย่าง
และ (And)	&& หรือ and	$(\$a > \$b) \ \&\& \ (\$c > \$d)$ (ได้ผลลัพธ์มีค่าความจริงเป็นเท็จ)
หรือ (Or)	หรือ Or	$(\$a > \$b) \ \ (\$c > \$d)$ (ได้ผลลัพธ์มีค่าความจริงเป็นจริง)
นิเสธ (Not)	!	$!(\$a > \$b) \ \&\& \ (\$c > \$d)$ (ได้ผลลัพธ์มีค่าความจริงเป็นจริง) $!(\$a > \$b) \ \ (\$c > \$d)$ (ได้ผลลัพธ์มีค่าความจริงเป็นเท็จ)
xor (Exclusive Or)	^ หรือ xor	$(\$a > \$b) \ \wedge \ (\$c > \$d)$ (ได้ผลลัพธ์มีค่าความจริงเป็นจริง)

พิจารณาโปรแกรมต่อไปนี้เพื่อทดสอบการใช้งานตัวดำเนินการทางคณิตศาสตร์

logical.php

```
<?
$a = 10;
$b = 3;
$c = 5;
$d = 7;
$logic_and = ($a > $b) && ($c > $d);
$logic_or = ($a > $b) || ($c > $d);
$logic_not1 = !(($a > $b) && ($c > $d));
$logic_not2 = !(($a > $b) || ($c > $d));
$logic_xor = ($a > $b) ^ ($c > $d);
echo "$a > $b) && ($c > $d) มีค่าความจริงเท่ากับ $logic_and <br>";
echo "$a > $b) || ($c > $d) มีค่าความจริงเท่ากับ $logic_or <br>";
echo "!(($a > $b) && ($c > $d)) มีค่าความจริงเท่ากับ $logic_not1 <br>";
echo "!(($a > $b) || ($c > $d)) มีค่าความจริงเท่ากับ $logic_not2 <br>";
echo "$a > $b) ^ ($c > $d) มีค่าความจริงเท่ากับ $logic_xor <br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะออกเป็น ดังนี้



```
http://localhost/book/ch05/logical.php - ILLUSION
File Edit View Favorites Tools Help
Back Forward Stop Home Search Favorites
Address http://localhost/book/ch05/logical.php
(10 > 3) && (5 > 7) มีค่าความจริงเท่ากับ 0
(10 > 3) || (5 > 7) มีค่าความจริงเท่ากับ 1
!((10 > 3) && (5 > 7)) มีค่าความจริงเท่ากับ 1
!((10 > 3) || (5 > 7)) มีค่าความจริงเท่ากับ 0
(10 > 3) ^ (5 > 7) มีค่าความจริงเท่ากับ 1
```

รูปที่ 5.3 แสดงผลลัพธ์ของโปรแกรม logical.php

5.5 ตัวดำเนินการสตริง

ตัวดำเนินการทางสตริงเป็นตัวดำเนินการที่ใช้จัดการข้อมูลประเภทอักขระ หรือข้อความ ตัวดำเนินการสตริงจะมีเพียง 1 ตัวดำเนินการคือ ตัวดำเนินการต่อสตริง (Concatenation Operator) สัญลักษณ์ที่ใช้เป็นเครื่องหมาย . (จุด) โดยตัวดำเนินการต่อสตริงนี้สามารถใช้เชื่อมต่อสตริง 2 สตริงเข้าด้วยกัน หรือใช้เชื่อมต่อสตริงกับข้อมูลในตัวแปรก็ได้ พิจารณาโปรแกรมต่อไปนี้

string.php

```
<?
    $a = "Mr.";
    $b = "Chanchai";
    $c = "Supaartagorn";
    echo $a.$b." ".$c;
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 5.4 แสดงผลลัพธ์ของโปรแกรม string.php

จากโปรแกรมได้มีการจัดเก็บข้อมูลไว้ในตัวแปร 3 ตัวด้วยกัน คือ ตัวแปร \$a เก็บสตริง "Mr." ตัวแปร \$b เก็บสตริง "Chanchai" และตัวแปร \$c เก็บสตริง "Supaartagorn" หลังจากนั้นก็ใช้คำสั่ง echo เพื่อทำการพิมพ์ผลลัพธ์โดยมีการใช้ตัวดำเนินการต่อสตริงทั้งหมด 3 จุดด้วยกัน คือ

- จุดที่ 1 เชื่อมระหว่างข้อมูลในตัวแปร \$a กับเครื่องหมายข้อมูลในตัวแปร \$b
- จุดที่ 2 เชื่อมระหว่างข้อมูลของตัวแปร \$b กับช่องว่าง
- จุดที่ 3 เชื่อมระหว่างช่องว่างกับข้อมูลในตัวแปร \$c

5.6 ตัวดำเนินการระดับบิต

ตัวดำเนินการระดับบิตเป็นตัวดำเนินการที่กระทำกับตัวเลขฐาน 2 ดังนั้นก่อนที่ตัวดำเนินการในกลุ่มนี้จะทำการประมวลผลก็จะมีการแปลงตัวเลขใดๆ ให้เป็นกลุ่มของตัวเลขฐาน 2 เสียก่อน การทำงานของตัวดำเนินการในกลุ่มนี้สามารถสรุปได้ดังตารางต่อไปนี้

(กำหนดให้ตัวแปร $a = 14$)

ตัวดำเนินการ	สัญลักษณ์	ตัวอย่าง
Shift left	<<	$a \ll 2$ (มีค่าเท่ากับ 56)
Shift right	>>	$a \gg 2$ (มีค่าเท่ากับ 3)

จากตารางข้างต้นตัวดำเนินการ shift left จะทำการแปลงตัวเลข 14 ให้เป็นเลขฐาน 2 ซึ่งมีค่าเท่ากับ 1110 หลังจากนั้นจะทำการเลื่อนบิตไปทางซ้าย 2 ตำแหน่ง และสุดท้ายก็จะทำการแปลงเลขฐาน 2 กลับไปเป็นเลขฐานสิบตามเดิม ดังนี้

14 แปลงเป็นเลขฐาน 2 \longrightarrow 0000 1110
 เลื่อนบิตไปทางซ้าย 2 ตำแหน่ง \longrightarrow 0011 1000
 แปลงกลับไปเป็นเลขฐาน 10 \longrightarrow 56

เช่นเดียวกันในการใช้ตัวดำเนินการ shift right ก็ทำการแปลงตัวเลข 14 ให้เป็นเลขฐาน 2 ซึ่งมีค่าเท่ากับ 1110 หลังจากนั้นจะทำการเลื่อนบิตไปทางขวา 2 ตำแหน่ง และสุดท้ายก็จะทำการแปลงตัวเลขฐาน 2 กลับไปเป็นเลขฐานสิบตามเดิม ดังนี้

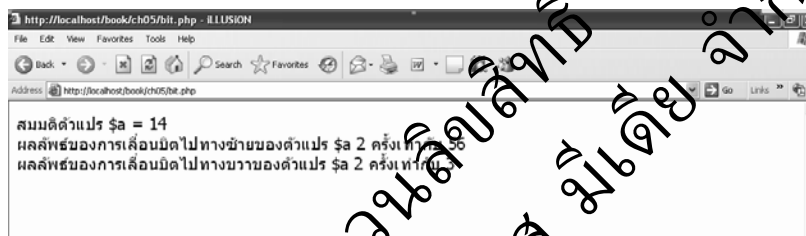
14 แปลงเป็นเลขฐาน 2 \longrightarrow 0000 1110
 เลื่อนบิตไปทางขวา 2 ตำแหน่ง \longrightarrow 0000 0011
 แปลงกลับไปเป็นเลขฐาน 10 \longrightarrow 3

พิจารณาการใช้ตัวดำเนินการระดับบิตจากโปรแกรมต่อไปนี้

bit.php

```
<?
$a = 14;
$bit_shift_left = $a << 2;
$bit_shift_right = $a >> 2;
echo "สมมติตัวแปร \$a = $a <br>";
echo "ผลลัพธ์ของการเลื่อนบิตไปทางซ้ายของตัวแปร \$a 2 ครั้งเท่ากับ $bit_shift_left
<br>";
echo "ผลลัพธ์ของการเลื่อนบิตไปทางขวาของตัวแปร \$a 2 ครั้งเท่ากับ $bit_shift_right
<br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 5.5 แสดงผลลัพธ์ของโปรแกรม bit.php

5.7 นิพจน์และลำดับการประมวลผลของนิพจน์

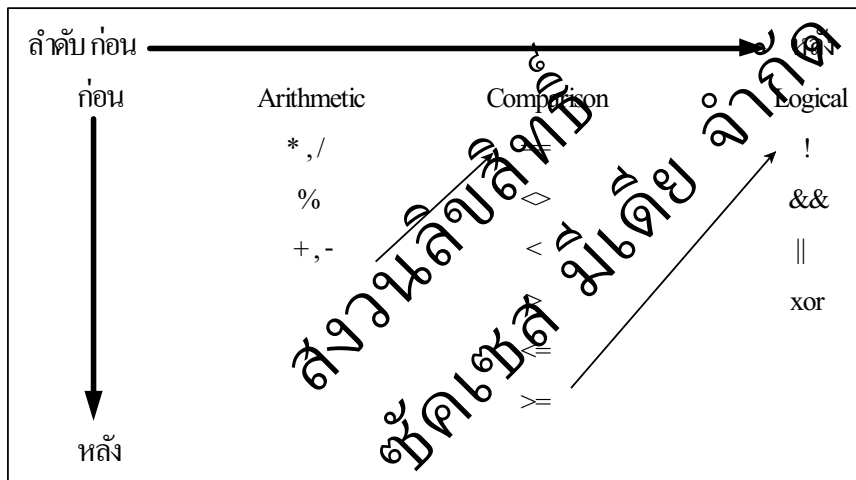
นิพจน์ (Expression) เกิดจาก ค่าคงที่ ตัวแปร หรือ ตัวเลข มากระทำกับตัวดำเนินการ (Operator) ในทางคณิตศาสตร์นิพจน์นี้อาจจะมองว่าเป็น สมการ หรือสมการก็ได้ พิจารณาตัวอย่างของนิพจน์จากตารางต่อไปนี้

ตัวดำเนินการ	ลักษณะของนิพจน์
$\$a = 5 + 6$	นิพจน์นี้เกิดจากตัวเลข 5 และ 6 ถูกนำมากระทำกับตัวดำเนินการทางคณิตศาสตร์ คือ เครื่องหมายบวก เมื่อบวกกันแล้วก็นำค่าที่ได้ไปเก็บไว้ในตัวแปร \$a
$\$a = \$b * \$c$	นิพจน์นี้เกิดจากตัวแปร \$b และ \$c ถูกนำมากระทำกับตัวดำเนินการทางคณิตศาสตร์ คือ เครื่องหมายคูณ เมื่อคูณกันแล้วก็นำค่าที่ได้ไปเก็บไว้ในตัวแปร \$a
$(3 > 5) \&\& (7 < 8)$	นิพจน์นี้เกิดจากนิพจน์ย่อย 2 ชุด ชุดแรก $(3 > 5)$ เป็นนิพจน์ที่เกิดจากตัวเลข 3 และ 5 กระทำกับตัวดำเนินการเปรียบเทียบ คือ เครื่องหมายมากกว่า นิพจน์ชุดที่สอง $(7 < 8)$ เป็นนิพจน์ที่เกิดจากตัวเลข 7 และ 8 กระทำกับตัวดำเนินการเปรียบเทียบ คือ เครื่องหมายไม่เท่ากับ หลังจากนั้นก็นำนิพจน์ย่อยทั้งสองมากระทำกับตัวดำเนินการทางตรรกะ คือ เครื่องหมาย And

จากตารางข้างต้นเราเห็นตัวอย่างของนิพจน์ในหลายๆ รูปแบบ แต่สิ่งหนึ่งที่มีความสำคัญก็คือเรื่องของลำดับการประมวลผล ลองพิจารณาตัวอย่างนิพจน์ต่อไปนี้เป็นตัวแปร \$a\$ มีผลลัพธ์เท่ากับเท่าไร

```
$a = 4 + 5 * 3
```

ถ้าเราไม่รู้จักเรื่องของลำดับการประมวลผล เราก็จะตอบว่าตัวแปร \$a\$ จะมีค่าเท่ากับ 27 เนื่องจากการคำนวณจะเริ่มจากการบวกของตัวเลข 4 กับ 5 ได้เป็น 9 หลังจากนั้นก็นำค่าตัวเลข 9 ไปคูณกับตัวเลข 3 ทำให้ได้ผลลัพธ์เท่ากับ 27 ซึ่งเป็นคำตอบที่ผิด เพราะว่าผลลัพธ์ของตัวแปร \$a\$ นี้มีค่าเท่ากับ 19 ทำไมผลลัพธ์ของตัวแปร \$a\$ นี้จึงมีค่าเท่ากับ 19 พิจารณาแผนภาพลำดับการประมวลผลต่อไปนี้



รูปที่ 5.6 ลำดับการประมวลผลของนิพจน์

จากรูปที่ 5.6 จะแสดงถึงลำดับการประมวลผลก่อน-หลัง ถ้าในนิพจน์มีการใช้ตัวดำเนินการหลายๆ ประเภทผสมกันให้ยึดแผนภาพลำดับนี้ โดยพิจารณาว่าในนิพจน์ตัวดำเนินการใดมีลำดับการประมวลผลก่อน เราก็ต้องทำการประมวลผลแก่ตัวดำเนินการนั้นก่อน ถึงแม้ว่าตัวดำเนินการนั้นจะอยู่ที่หลังก็ตาม ดังนั้นจากนิพจน์ตัวอย่างก่อนหน้านี้ คือ $a = 4 + 5 * 3$ ในนิพจน์นี้มีตัวดำเนินการคือ เครื่องหมาย + และ * ซึ่งเราจะต้องทำการคำนวณที่เครื่องหมาย * ก่อนเพราะลำดับการประมวลผลของ * มาก่อน + ผลลัพธ์ที่ได้ก็คือ นำตัวเลข 5 และ 3 คูณกันก่อนได้เท่ากับ 15 หลังจากนั้นค่อยนำตัวเลข 4 ไปบวกกับตัวเลข 15 เพราะฉะนั้นผลลัพธ์ของตัวแปร \$a\$ จะมีค่าเท่ากับ 19 นั่นเอง อย่างไรก็ตามจากตารางสรุปข้างต้นยังมีข้อยกเว้นอยู่ 2 ประการ คือ

1. ถ้าในนิพจน์มีการใช้วงเล็บให้ดำเนินการคำนวณในส่วนของนิพจน์ที่มีวงเล็บนั้นก่อนถึงแม้ว่านิพจน์นั้นจะมีลำดับในการประมวลผลที่หลังก็ตาม
2. ถ้าในนิพจน์มีการใช้ตัวดำเนินการซ้ำกัน หรือมีการใช้ตัวดำเนินการที่มีลำดับการประมวลผลเท่ากัน ให้ทำการประมวลผลนิพจน์นั้นจากซ้ายไปขวา

พิจารณานิพจน์เพิ่มเติมอีก 2 นิพจน์ต่อไปนี้ว่าตัวแปร \$a และ \$b มีค่าเท่ากับเท่าไร

- \$a = (4 + 5) * 3
- \$b = 4 + 5 - 3

ถ้าคิดไม่ออกอาจจะลองเขียนโปรแกรมเพื่อทดสอบค่าของตัวแปร \$a และ \$b ดังนี้

sequence.php

```
<?
    $a = (4 + 5) * 3;
    $b = 4 + 5 - 3;
    echo "ค่าของตัวแปร \$a มีค่าเท่ากับ $a <br>";
    echo "ค่าของตัวแปร \$b มีค่าเท่ากับ $b <br>";
?>
```

สรุป

ตัวดำเนินการ (Operator) คือ เครื่องหมายต่างๆ ที่ใช้ในการคำนวณ หรือการทำฟังก์ชันทางคณิตศาสตร์ โดยสามารถแบ่งประเภทของตัวดำเนินการได้เป็น 5 ประเภท

ตัวดำเนินการทางคณิตศาสตร์เป็นตัวดำเนินการในกลุ่มที่เกี่ยวข้องกับการคำนวณหาค่าผลลัพธ์ต่างๆ ทางคณิตศาสตร์ ผลลัพธ์จากการใช้ตัวดำเนินการทางคณิตศาสตร์นี้จะได้ผลลัพธ์เป็นตัวเลขใดๆ เช่น เครื่องหมายบวก, ลบ, คูณ และหาร

ตัวดำเนินการเปรียบเทียบเป็นตัวดำเนินการที่ใช้สำหรับการเปรียบเทียบตัวเลข หรือตัวแปร โดยผลลัพธ์จากการใช้ตัวดำเนินการเปรียบเทียบนี้จะได้ผลลัพธ์มีค่าความจริงเป็นจริง (True) หรือเท็จ (False) เช่น เครื่องหมายเท่ากับ, มากกว่า, น้อยกว่า, ไม่เท่ากับ

ตัวดำเนินการทางตรรกศาสตร์เป็นตัวดำเนินการที่ใช้ทำการเชื่อมค่าความจริงต่างๆ โดยเฉพาะการนำตัวดำเนินการทางตรรกศาสตร์นี้ไปประยุกต์ใช้กับคำสั่งตรวจสอบเงื่อนไข ผลลัพธ์จากการใช้ตัวดำเนินการทางตรรกศาสตร์นี้จะได้ผลลัพธ์ที่มีค่าความจริงเป็นจริง (True) หรือเท็จ (False) เช่น ตัวดำเนินการ And, Or, Not และ xor (Exclusive Or)

ตัวดำเนินการทางสตริงเป็นตัวดำเนินการที่ใช้จัดการข้อมูลประเภทอักขระ หรือข้อความ ตัวดำเนินการสตริงจะมีเพียง 1 ตัวดำเนินการคือ ตัวดำเนินการต่อสตริง (Concatenation Operator) สัญลักษณ์ที่ใช้เป็นเครื่องหมาย . (จุด) โดยตัวดำเนินการต่อสตริงนี้สามารถใช้เชื่อมต่อสตริง 2 สตริงเข้าด้วยกัน

ตัวดำเนินการระดับบิตเป็นตัวดำเนินการที่กระทำกับตัวเลขฐาน 2 ดังนั้นก่อนที่ตัวดำเนินการในกลุ่มนี้จะทำการประมวลผลก็จะมีค่าการแปลงตัวเลขใดๆ ให้เป็นกลุ่มของตัวเลขฐาน 2 เสียก่อน

นิพจน์ (Expression) เกิดจาก ค่าคงที่ ตัวแปร หรือ ตัวเลข มากระทำกับตัวดำเนินการ (Operator) ในทางคณิตศาสตร์นิพจน์เราอาจมองว่าเป็น สมการ หรือสมการก็ได้ ประการที่สำคัญในเรื่องของนิพจน์ คือเรื่องของลำดับในการประมวลผล เพราะจะมีผลต่อผลลัพธ์ที่ได้จากการประมวลผล

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

แบบฝึกหัดบทที่ 5

1. ให้อธิบายลักษณะการใช้งานของตัวดำเนินการในภาษา PHP ทั้ง 5 ประเภทพอสังเขป

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. กำหนดตัวแปรและข้อมูล ดังนี้

```
$date = 25
```

```
$month = "เมษายน"
```

```
$year = 2550
```

ให้เขียนโปรแกรมภาษา PHP โดยใช้ตัวแปรข้างต้นเพื่อพิมพ์ผลลัพธ์ให้มีลักษณะ ดังนี้

วันที่ 25 เดือน เมษายน ค.ศ. 2007

.....

.....

.....

.....

.....

.....

.....

.....

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

3. ลำดับในการประมวลผลของนิพจน์มีความสำคัญอย่างไรในการเขียนโปรแกรม

.....
.....
.....
.....

4. ให้หาค่าของนิพจน์ต่อไปนี้

4.1) $(7 + 3 * 5 / 7) / 2$

.....
.....
.....
.....

4.2) `("OTOP" == "otop") && (10 >=`

.....
.....
.....
.....

5. จากสมการทางคณิตศาสตร์ต่อไปนี้ ให้เขียนเป็นนิพจน์ของโปรแกรมภาษา PHP เมื่อกำหนดให้ (a, b, c และ d คือชื่อตัวแปร)

$$d = \frac{b^2 + 4ac}{2a}$$

.....
.....
.....
.....

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

คำสั่งตรวจสอบเงื่อนไข และคำสั่งทำซ้ำ

ในชีวิตประจำวันเราจะพบกับการตัดสินใจ และการทำซ้ำอยู่บ่อยครั้งมาก อาทิเช่น ก่อนจะออกจากบ้านเราจะต้องคิดก่อนว่าวันนี้จะแต่งตัวอย่างไรดี จะใส่เสื้อหรือกางเกงชุดไหนดีหรือชุดไหนที่เหมาะสมกับกิจกรรมที่จะทำในวันนี้ นิสิตนักศึกษาอ่านหนังสือเพื่อเตรียมตัวสอบบางคนอาจจะอ่านหนังสือหลายรอบก่อนเข้าสอบโดยมีข้อกำหนดว่าจะอ่านซ้ำไปเรื่อยๆ จนกระทั่งจะเข้าใจ จากสถานการณ์ที่ยกขึ้นมานี้เป็นตัวอย่างของการตรวจสอบเงื่อนไข และการทำซ้ำในชีวิตประจำวัน ในการเขียนโปรแกรมคำสั่งตรวจสอบเงื่อนไขและคำสั่งทำซ้ำนี้จะช่วยให้การเขียนโปรแกรมง่ายขึ้น และทำให้โปรแกรมที่เขียนขึ้นมีประสิทธิภาพ คอมพิวเตอร์จะทำการประมวลผลเพื่อตัดสินใจเงื่อนไขต่างๆ ที่ผู้เขียนโปรแกรมกำหนดขึ้น และคอมพิวเตอร์ยังสามารถทำการวนซ้ำทำคำสั่งที่ต้องการจำนวนมากได้ในเวลาอันรวดเร็ว ในบทนี้เราจะได้ศึกษารูปแบบของคำสั่งทั้ง 2 คำสั่งนี้

จุดประสงค์ ทั่วไป

1. เพื่อให้รู้จัก เข้าใจและสามารถใช้งานคำสั่ง If ได้
2. เพื่อให้รู้จัก เข้าใจและสามารถใช้งานคำสั่ง Switch ได้
3. เพื่อให้รู้จัก เข้าใจและสามารถใช้งานคำสั่ง While ได้
4. เพื่อให้รู้จัก เข้าใจและสามารถใช้งานคำสั่งทำซ้ำ For ได้
5. เพื่อให้รู้จัก เข้าใจและสามารถใช้งานคำสั่งทำซ้ำ Do...while ได้
6. เพื่อให้รู้จักคำสั่งออกจากการทำงาน และสามารถนำไปใช้ร่วมกับคำสั่งตรวจสอบเงื่อนไข และคำสั่งทำซ้ำได้

จุดประสงค์ เชิงพฤติกรรม

1. สามารถเขียนโปรแกรมโดยใช้คำสั่งตรวจสอบเงื่อนไข If ได้
2. สามารถเขียนโปรแกรมโดยใช้คำสั่งตรวจสอบเงื่อนไข Case ได้
3. สามารถเขียนโปรแกรมโดยใช้คำสั่งทำซ้ำ For ได้
4. สามารถเขียนโปรแกรมโดยใช้คำสั่งทำซ้ำ While ได้
5. สามารถเขียนโปรแกรมโดยใช้คำสั่งทำซ้ำ Do...while ได้
6. สามารถเขียนโปรแกรมโดยใช้คำสั่ง Break และ Exit ได้

เนื้อหาสาระ:

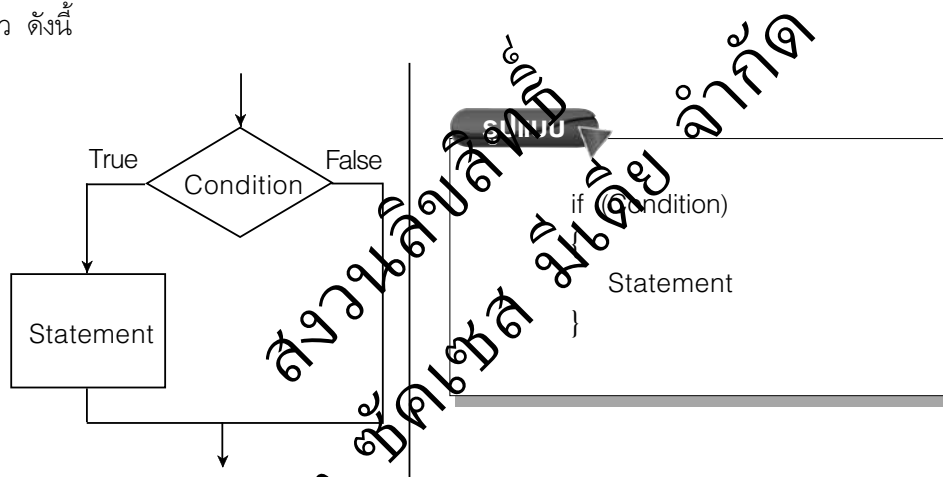
1. คำสั่งตรวจสอบเงื่อนไข If
2. คำสั่งตรวจสอบเงื่อนไข Switch
3. คำสั่งทำซ้ำ While
4. คำสั่งทำซ้ำ For
5. คำสั่งทำซ้ำ Do...while
6. คำสั่งออกจากการทำงาน

6.1 คำสั่งตรวจสอบเงื่อนไข If

คำสั่งเพื่อใช้ในการตรวจสอบเงื่อนไข หรืออาจจะเรียกว่าคำสั่งแบบทางเลือกในลักษณะของคำสั่ง if นี้ สามารถแบ่งออกได้เป็น 3 ลักษณะด้วยกัน คือ คำสั่งตรวจสอบเงื่อนไขแบบทางเลือกเดียว, คำสั่งตรวจสอบเงื่อนไขแบบ 2 ทางเลือก และคำสั่งตรวจสอบเงื่อนไขแบบหลายทางเลือก จะขออธิบายลักษณะของแต่ละคำสั่ง พร้อมทั้งยกตัวอย่างโปรแกรม PHP ของแต่ละคำสั่ง ดังนี้

คำสั่งตรวจสอบเงื่อนไขแบบทางเลือกเดียว (Single-alternative)

เป็นลักษณะของเงื่อนไขที่ให้ตรวจสอบ หรือทางเลือกที่ให้ตรวจสอบมีแค่ทางเลือกเดียว โดยเงื่อนไขที่ใช้ในการตรวจสอบปกติจะเป็นนิพจน์ ดังนั้นผลของนิพจน์ดังกล่าวก็จะมีค่าไม่จริงก็เท็จอย่างใดอย่างหนึ่ง พิจารณา Flowchart เพื่อแสดงการทำงานของคำสั่งตรวจสอบเงื่อนไขแบบทางเลือกเดียว รวมทั้งรูปแบบคำสั่ง If ทางเลือกเดียว ดังนี้



รูปที่ 6.1 Flowchart และรูปแบบคำสั่ง if แบบทางเลือกเดียว

จากรูปที่ 6.1 ลักษณะการทำงานของคำสั่งตรวจสอบเงื่อนไขแบบทางเลือกเดียวจะมีนิพจน์หลังคำสั่ง if ถ้าเงื่อนไขดังกล่าวมีค่าเป็นจริงก็จะลงมาทำคำสั่งที่อยู่ภายในวงเล็บปีกกา ซึ่งคำสั่งภายในวงเล็บปีกกานี้จะมีคำสั่งก็ได้ แต่ถ้านิพจน์หลังคำสั่ง if มีค่าเป็นเท็จโปรแกรมก็จะข้ามคำสั่ง if และคำสั่งในวงเล็บปีกกาลงไปทำคำสั่งที่อยู่ถัดลงไป

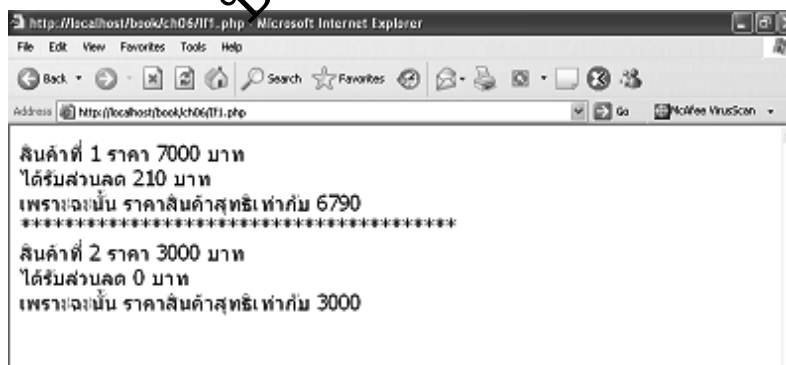
พิจารณาตัวอย่างโปรแกรมต่อไปนี้โดยต้องการเขียนโปรแกรมภาษา PHP เพื่อคำนวณหาส่วนลดที่ได้จากการซื้อสินค้า และคำนวณคืนค่าสุทธิ โดยกำหนดเงื่อนไขในการให้ส่วนลด คือ ถ้าลูกค้าซื้อสินค้าเป็นจำนวนเงินมากกว่าหรือเท่ากับ 5,000 บาทจะได้รับส่วนลด 3% ของราคาสินค้า และได้กำหนดให้ \$price1 และ \$price2 คือตัวแปรของราคาสินค้าก่อนหักส่วนลด, \$discount คือตัวแปรของส่วนลดที่คำนวณได้ และ \$netprice คือ ตัวแปรของของราคาสินค้าหลังจากหักส่วนลดแล้ว

if1.php

```
<?
    $price1 = 7000;
    $price2 = 3000;
    $discount = 0;
    if ($price1 >= 5000)
    {
        $discount = $price1*0.03;
    }
    $netprice = $price1 - $discount;
    echo "สินค้าที่ 1 ราคา $price1 บาท<br>";
    echo "ได้รับส่วนลด $discount บาท<br>";
    echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
    echo "***** <br>";

    $discount = 0;
    if ($price2 >= 5000)
    {
        $discount = $price2*0.03;
    }
    $netprice = $price2 - $discount;
    echo "สินค้าที่ 2 ราคา $price2 บาท<br>";
    echo "ได้รับส่วนลด $discount บาท<br>";
    echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



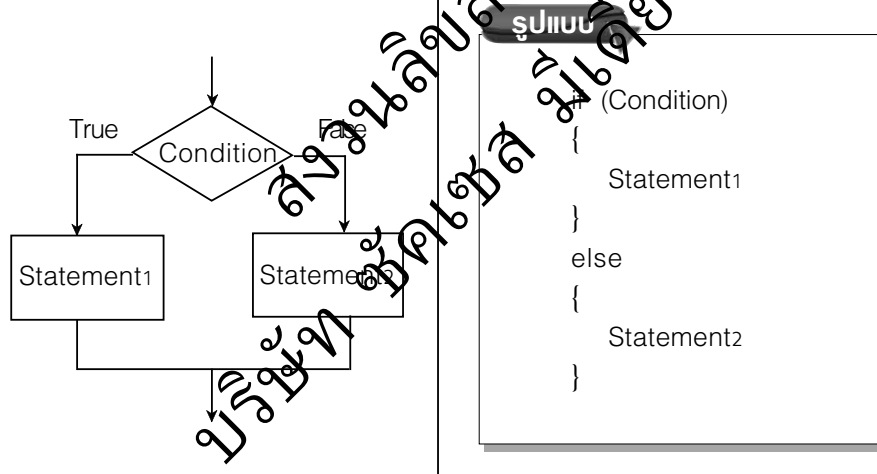
รูปที่ 6.2 แสดงผลลัพธ์ของโปรแกรม if1.php

จากโปรแกรมและผลลัพธ์ของโปรแกรมข้างต้นได้สมมติราคาสินค้า 2 ราคา คือ 7,000 บาท และ 3,000 บาท หลังจากนั้นก็นำราคาสินค้าแต่ละราคาเข้าไปทำการตรวจสอบเงื่อนไขว่าราคาสินค้าดังกล่าวมีค่ามากกว่าหรือเท่ากับ 5,000 บาทหรือไม่ ถ้าจริงก็จะทำการคำนวณส่วนลด 3% ให้ แต่ถ้าราคาสินค้าไม่ถึง 5,000 บาท ก็จะไม่มีการคำนวณส่วนลดใดๆ ให้เลย หลังจากนั้นก็จะนำค่าส่วนลดที่คำนวณได้ไปทำการคำนวณหาราคาสินค้าสุทธิ และก็พิมพ์ผลลัพธ์ทั้งหมดออกมา จะเห็นว่าราคาสินค้าที่ 1 มีราคา 7,000 บาท โปรแกรมจะมีการคำนวณส่วนลดให้ แต่ราคาสินค้าที่ 2 มีราคา 3,000 บาท ก็จะไม่มีการคำนวณส่วนลดให้ ดังนั้นสินค้าที่ 2 ก็จะมีส่วนลดเป็นศูนย์

คำสั่งตรวจสอบเงื่อนไขแบบ 2 ทางเลือก (Dual-alternative)

เป็นลักษณะของเงื่อนไขที่ให้ตรวจสอบ หรือทางเลือกที่ให้ตรวจสอบมี 2 ทางเลือก โดยเงื่อนไขที่ใช้ในการตรวจสอบปกติจะเป็นนิพจน์ ดังนั้นผลลัพธ์ของนิพจน์ดังกล่าวก็จะมีค่าไม่จริงก็เท็จอย่างใดอย่างหนึ่ง

พิจารณา Flowchart เพื่อแสดงการทำงานของคำสั่งตรวจสอบเงื่อนไขแบบ 2 ทางเลือก รวมทั้งรูปแบบคำสั่ง If แบบ 2 ทางเลือก ดังนี้



รูปที่ 6.3 Flowchart และรูปแบบคำสั่ง If แบบ 2 ทางเลือก

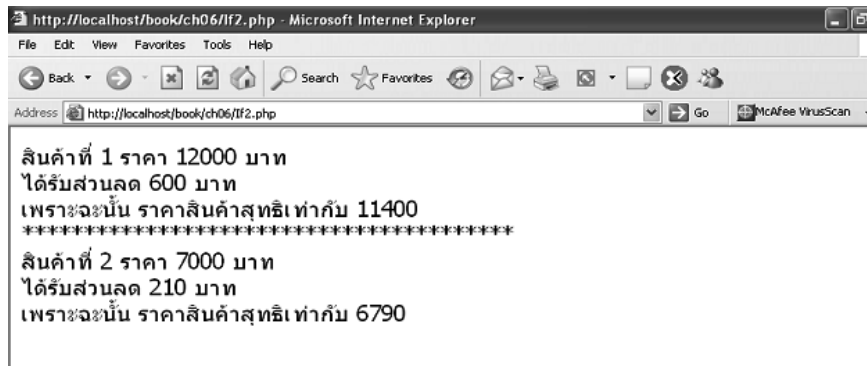
จากรูปที่ 6.3 ลักษณะการทำงานของคำสั่งตรวจสอบเงื่อนไขแบบ 2 ทางเลือกจะมีนิพจน์หลังคำสั่ง if ถ้าเงื่อนไขดังกล่าวมีค่าเป็นจริงก็จะลงมาทำคำสั่งที่อยู่ภายในวงเล็บปีกกาของคำว่า if ซึ่งคำสั่งภายในวงเล็บปีกกาจะมีคำสั่งก็ได้ แต่ถ้านิพจน์หลังคำสั่ง if มีค่าเป็นเท็จโปรแกรมก็จะลงมาทำคำสั่งในวงเล็บปีกกาที่อยู่หลังคำสั่ง else และเช่นเดียวกันคำสั่งภายในวงเล็บปีกกาหลังคำสั่ง else นี้จะมีคำสั่งก็ได้

พิจารณาตัวอย่างโปรแกรมต่อไปนี้โดยต้องการเขียนโปรแกรมภาษา PHP เพื่อคำนวณหาส่วนลดที่ได้จากการซื้อสินค้า และค่าราคาสินค้าสุทธิ โดยกำหนดเงื่อนไขในการให้ส่วนลด คือ ถ้าลูกค้าซื้อสินค้าเป็นจำนวนเงินมากกว่าหรือเท่ากับ 10,000 บาทจะได้รับส่วนลด 5% ของราคาสินค้า แต่ถ้าราคาสินค้าไม่ถึง 10,000 บาทก็จะได้รับส่วนลดเพียง 3% ของราคาสินค้า และได้กำหนดให้ \$price1 และ \$price2 คือตัวแปรของราคาสินค้าก่อนหักส่วนลด, \$discount คือตัวแปรของส่วนลดที่คำนวณได้ และ \$netprice คือ ตัวแปรของของราคาสินค้าหลังจากหักส่วนลดแล้ว

if2.php

```
<?
$price1 = 12000;
$price2 = 7000;
$discount = 0;
if ($price1 >= 10000)
{
    $discount = $price1*0.05;
}
else
{
    $discount = $price1*0.03;
}
$netprice = $price1 - $discount;
echo "สินค้าที่ 1 ราคา $price1 บาท<br>";
echo "ได้รับส่วนลด $discount บาท<br>";
echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
echo "*****<br>";
$discount = 0;
if ($price2 >= 10000)
{
    $discount = $price2*0.05;
}
else
{
    $discount = $price2*0.03;
}
$netprice = $price2 - $discount;
echo "สินค้าที่ 2 ราคา $price2 บาท<br>";
echo "ได้รับส่วนลด $discount บาท<br>";
echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



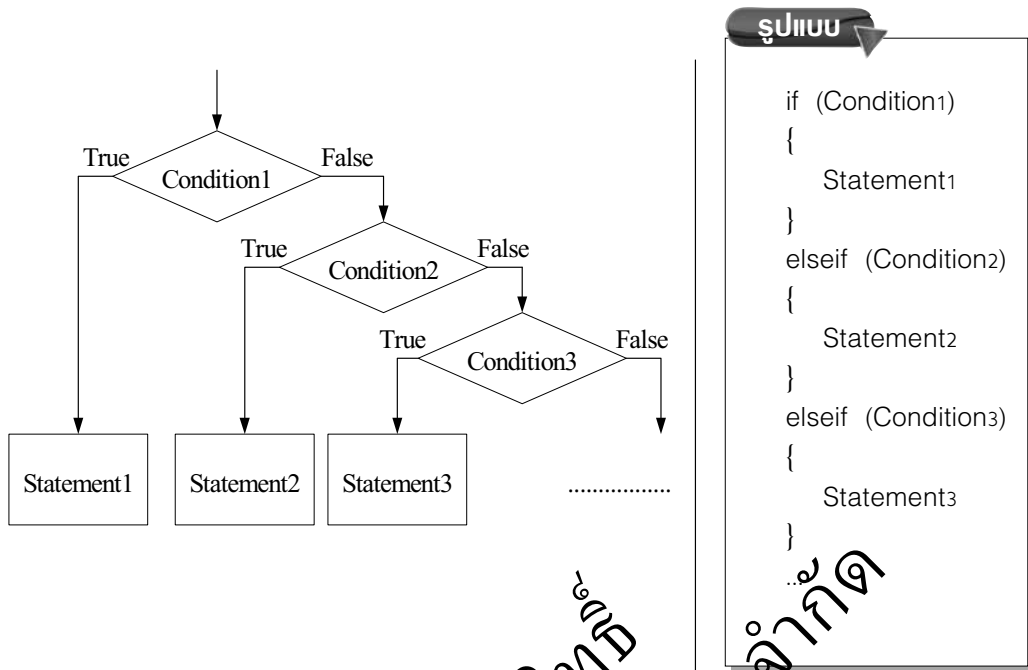
รูปที่ 6.4 แสดงผลลัพธ์ของโปรแกรม if2.php

จากโปรแกรมและผลลัพธ์ของโปรแกรมข้างต้นได้สมมติราคาสินค้า 2 ราคา คือ 12,000 บาท และ 7,000 บาท หลังจากนั้นก็นำราคาสินค้าแต่ละราคาเข้าไปทำการตรวจสอบเงื่อนไขว่าราคาสินค้าจึงกล่าวว่ามีค่ามากกว่าหรือเท่ากับ 10,000 บาทหรือไม่ ถ้าจริงก็จะทำการคำนวณส่วนลด 5% ให้ แต่ถ้าราคาสินค้าไม่ถึง 10,000 บาท ก็จะทำการคำนวณส่วนลดให้เพียง 3% หลังจากนั้นก็จะนำค่าส่วนลดที่คำนวณได้ไปทำการคำนวณหาราคาสินค้าสุทธิ และก็พิมพ์ผลลัพธ์ทั้งหมดออกมา จะเห็นว่าราคาสินค้าที่ 1 มีราคา 12,000 บาท ลูกค้านี้จะได้รับส่วนลด 5% คือ 600 บาท แต่ราคาสินค้าที่ 2 มีราคา 7,000 บาท ลูกค้าจะได้รับส่วนลด 3% คือ 210 บาท

คำสั่งตรวจสอบเงื่อนไขแบบหลายทางเลือก (หรือ) ทางเลือกแบบซ้อน (Nested-alternative)

เป็นลักษณะของเงื่อนไขที่ให้ตรวจสอบ หรือมีทางเลือกจำนวนหลายทาง ซึ่งแต่ละเงื่อนไขจะมีลักษณะวางซ้อนกัน ลักษณะในการประมวลผลของคำสั่ง if แบบนี้ก็ยังมีลักษณะการทำงานเหมือนเดิม คือหลังคำสั่ง if จะเป็นนิพจน์ให้ตรวจสอบว่าจริงหรือเท็จ ถ้านิพจน์มีค่าเป็นจริงก็จะทำคำสั่งในวงเล็บปีกกาที่อยู่ถัดลงไป แต่ถ้านิพจน์มีค่าเป็นเท็จก็จะลงไปยังคำสั่งในวงเล็บปีกกาหลังคำสั่ง else แต่ในกรณีนี้หลังคำสั่ง else จะมีคำสั่ง if อีกคำสั่งซ้อนอยู่ ซึ่งเราก็ต้องเริ่มพิจารณาเงื่อนไขหลัง if นี้ตามหลักการเดิม ลักษณะคำสั่งก็จะเป็นเช่นนี้ไปเรื่อยๆ จนกระทั่งเงื่อนไขที่ต้องตรวจสอบครบทุกเงื่อนไข

พิจารณา Flowchart เพื่อแสดงการทำงานของคำสั่งตรวจสอบเงื่อนไขแบบหลายทางเลือก รวมทั้งรูปแบบคำสั่ง if แบบหลายทางเลือก ดังนี้



รูปที่ 6.5 Flowchart และรูปแบบของ if แบบหลายทางเลือก

ตัวอย่างโปรแกรมต่อไปนี้จะคำนวณหาส่วนลดที่ได้จากการซื้อสินค้า และค่าราคาสินค้าสุทธิ โดยกำหนดเงื่อนไขในการให้ส่วนลด คือ ถ้าลูกค้าซื้อสินค้าเป็นจำนวนเงินตั้งแต่ 5,000 บาท ถึง 10,000 บาท จะได้รับส่วนลด 3% ของราคาสินค้า ถ้าลูกค้าซื้อสินค้าเป็นจำนวนเงินตั้งแต่ 10,001 บาท ถึง 20,000 บาท จะได้รับส่วนลด 5% ของราคาสินค้า ถ้าหากราคาสินค้าที่ลูกค้าซื้อหรือเท่ากับ 20,001 บาท จะได้รับส่วนลด 10% ของราคาสินค้า และถ้าหากจำนวนเงินที่ลูกค้าซื้อสินค้านอกเหนือจากนี้ลูกค้าจะไม่ได้รับส่วนลดใดๆ และได้กำหนดให้ \$price1, \$price2, \$price3 และ \$price4 คือตัวแปรของราคาสินค้าก่อนหักส่วนลด, \$discount คือตัวแปรของส่วนลดที่คำนวณได้ และ \$netprice คือ ตัวแปรของของราคาสินค้าหลังจากหักส่วนลดแล้ว

If3.php

```
<?
    $price1 = 7000;
    $price2 = 12000;
    $price3 = 25000;
    $price4 = 4500;

    if (($price1 >= 5000) && ($price1 <= 10000))
    {
```

```

    $discount = $price1*0.03;
}
elseif (($price1 >= 10001) && ($price1 <= 20000))
{
    $discount = $price1*0.05;
}
elseif ($price1 >= 20001)
{
    $discount = $price1*0.1;
}
else
{
    $discount = 0;
}
$netprice = $price1 - $discount;
echo "สินค้าที่ 1 ราคา $price1 บาท<br>";
echo "ได้รับส่วนลด $discount บาท<br>";
echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
echo "***** <br>";
if (($price2 >= 5000) && ($price2 <= 10000))
{
    $discount = $price2*0.03;
}
elseif (($price2 >= 10001) && ($price2 <= 20000))
{
    $discount = $price2*0.05;
}
elseif ($price2 >= 20001)
{
    $discount = $price2*0.1;
}
else
{
    $discount = 0;
}
$netprice = $price2 - $discount;
echo "สินค้าที่ 2 ราคา $price2 บาท<br>";
echo "ได้รับส่วนลด $discount บาท<br>";
echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
echo "***** <br>";

```

สงวนลิขสิทธิ์
 บริษัท ซัคเซส มีเดีย จำกัด

```

if (($price3 >= 5000) && ($price3 <= 10000))
{
    $discount = $price3*0.03;
}
elseif (($price3 >= 10001) && ($price3 <= 20000))
{
    $discount = $price3*0.05;
}
elseif ($price3 >= 20001)
{
    $discount = $price3*0.1;
}
else
{
    $discount = 0;
}
$netprice = $price3 - $discount;
echo "สินค้าที่ 3 ราคา $price3 บาท<br>";
echo "ได้รับส่วนลด $discount บาท<br>";
echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
echo "***** <br>";

if (($price4 >= 5000) && ($price4 <= 10000))
{
    $discount = $price4*0.03;
}
elseif (($price4 >= 10001) && ($price4 <= 20000))
{
    $discount = $price4*0.05;
}
elseif ($price4 >= 20001)
{
    $discount = $price4*0.1;
}
else
{
    $discount = 0;
}
$netprice = $price4 - $discount;

```

สงวนลิขสิทธิ์
บริษัท ชัคเซล จำกัด

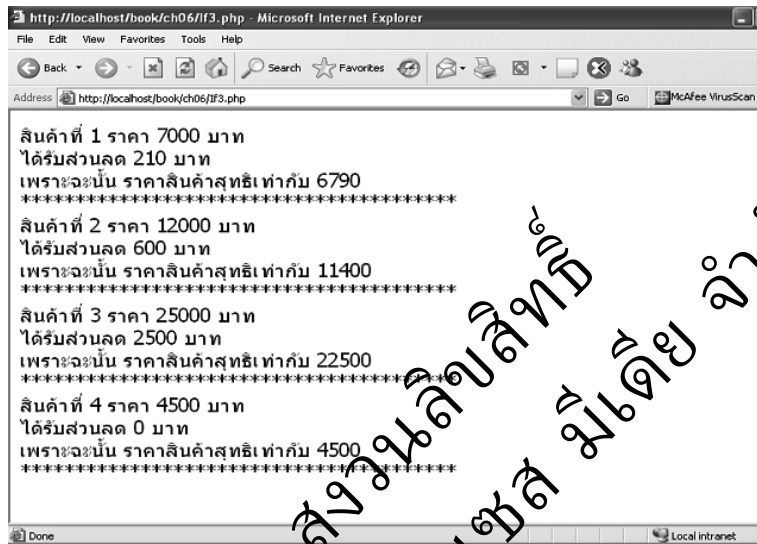


```

echo "สินค้าที่ 4 ราคา $price4 บาท<br>";
echo "ได้รับส่วนลด $discount บาท<br>";
echo "เพราะฉะนั้น ราคาสินค้าสุทธิเท่ากับ $netprice <br>";
echo "***** <br>";
?>

```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 6.6 แสดงผลลัพธ์ของโปรแกรม IF3.php

จากโปรแกรมและผลลัพธ์ที่ได้ของโปรแกรมในลักษณะหลายทางเลือก ได้กำหนดราคาสินค้าที่ลูกค้าซื้อทั้งหมด 4 ราคา คือ 7,000 บาท, 12,000 บาท, 25,000 บาท และ 4,500 บาท ซึ่งจากเงื่อนไขในการให้ส่วนลดนี้ทั้งหมด 4 ทางเลือก ดังนั้นในการเขียนโปรแกรมต้องใช้คำสั่งแบบหลายทางเลือก ซึ่งจะสามารถคำนวณส่วนลดที่ได้ และราคาสินค้าสุทธิได้ดังนี้

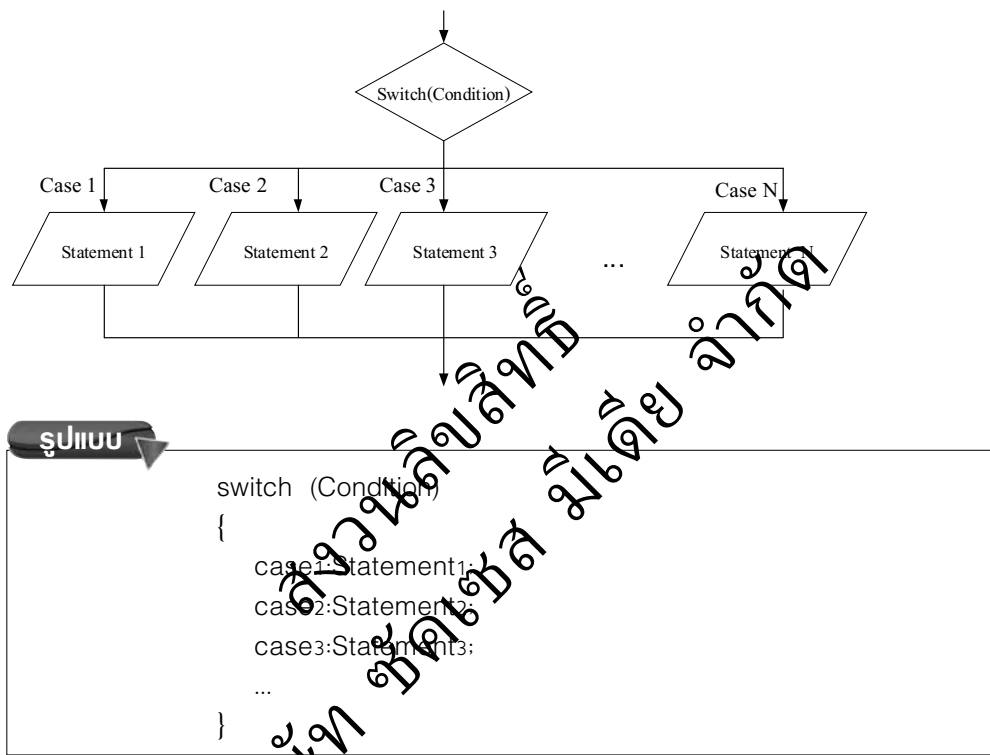
ถ้าราคาสินค้า 7,000 บาท ตรงกับทางเลือกที่ 1 คือได้ส่วนลด 3% ดังนั้นส่วนลดที่ได้ คือ 210 บาท
 ถ้าราคาสินค้า 12,000 บาท ตรงกับทางเลือกที่ 2 คือได้ส่วนลด 5% ดังนั้นส่วนลดที่ได้ คือ 600 บาท
 ถ้าราคาสินค้า 25,000 บาท ตรงกับทางเลือกที่ 3 คือได้ส่วนลด 10% ดังนั้นส่วนลดที่ได้ คือ 2,500 บาท
 ถ้าราคาสินค้า 4,500 บาท ตรงกับทางเลือกที่ 4 คือไม่ได้รับส่วนลด ดังนั้นส่วนลดที่ได้ คือ 0 บาท

หลังจากคำนวณส่วนลดได้แล้ว ก็ทำการคำนวณราคาสินค้าสุทธิ ซึ่งคำนวณได้จากราคาสินค้าก่อนหักส่วนลดลบด้วยส่วนลดที่ได้รับ

6.2 คำสั่งตรวจสอบเงื่อนไข Switch

คำสั่งตรวจสอบเงื่อนไข switch เป็นคำสั่งที่เหมาะสมสำหรับเงื่อนไขที่มีหลายๆ ทางเลือก การใช้คำสั่ง switch จะช่วยให้โปรแกรมที่เขียนขึ้นสั้นกระชับ

พิจารณา Flowchart และรูปแบบคำสั่ง switch ตามรูปต่อไปนี้



รูปที่ 6.7 Flowchart และรูปแบบคำสั่ง Switch

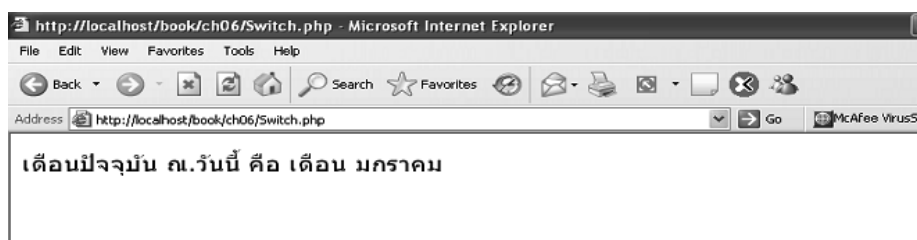
จากรูปที่ 6.7 คำสั่ง switch จะทำการตรวจสอบนิพจน์หรือเงื่อนไขที่อยู่หลังคำสั่ง switch ถ้าตรงกับกรณีไหนก็จะเลือกทำคำสั่งที่อยู่ในกรณีนั้นเลย ลักษณะนี้จะต่างจากคำสั่งแบบหลายทางเลือกที่คำสั่งในการตรวจสอบเงื่อนไขจะวางซ้อนๆ กัน ถ้าไม่ตรงกับเงื่อนไขแรกก็ลงไปตรวจสอบเงื่อนไขที่วางซ้อนถัดลงไป จะเห็นว่า การใช้คำสั่ง switch จะทำให้การเขียนโปรแกรมแบบหลายทางเลือกง่ายยิ่งขึ้น

สมมติตัวอย่างต่อไปนี้เป็นการใช้ฟังก์ชัน date("M") เพื่อดึงชื่อเดือนปัจจุบันของเครื่องเฉพาะตัวอักษร 3 ตัวแรก เช่น "Jan", "Feb" เป็นต้น หลังจากนั้นก็ใช้คำสั่ง switch เพื่อทำการเปรียบเทียบ และพิมพ์ชื่อเดือนเป็นภาษาไทย สังเกตว่าการเปรียบเทียบในแต่ละกรณีจะใช้คำสั่ง break เพื่อสั่งให้โปรแกรมออกจากการเปรียบเทียบในคำสั่ง switch เพราะถ้าไม่ใช้คำสั่ง break หลังจากโปรแกรมเข้าทำในกรณีไหนแล้วก็จะยังคงลงไปทำคำสั่งที่อยู่ในกรณีถัดลงไปอีกด้วย ดังโปรแกรมต่อไปนี้

Switch.php

```
<?
    $month = date("M");
    echo "เดือนปัจจุบัน ณ.วันนี้ คือ เดือน ";
    switch($month)
    {
        case "Jan": echo "มกราคม"; break;
        case "Feb": echo "กุมภาพันธ์"; break;
        case "Mar": echo "มีนาคม"; break;
        case "Apr": echo "เมษายน"; break;
        case "May": echo "พฤษภาคม"; break;
        case "Jun": echo "มิถุนายน"; break;
        case "Jul": echo "กรกฎาคม"; break;
        case "Aug": echo "สิงหาคม"; break;
        case "Sep": echo "กันยายน"; break;
        case "Oct": echo "ตุลาคม"; break;
        case "Nov": echo "พฤศจิกายน"; break;
        case "Dec": echo "ธันวาคม"; break;
    }
?>
```

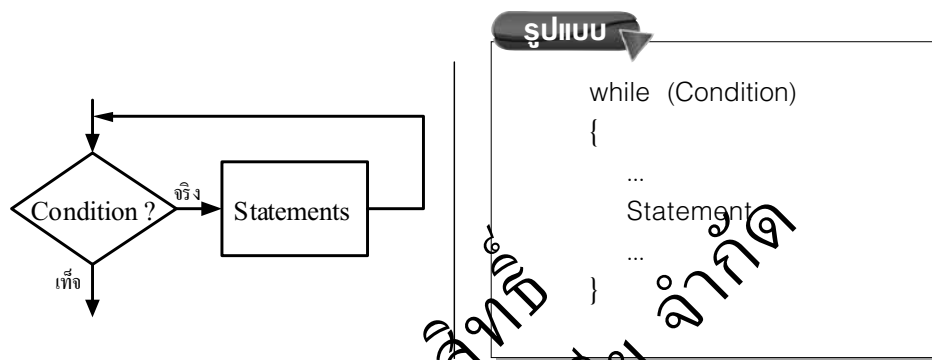
ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 6.8 แสดงผลลัพธ์ของโปรแกรม Switch.php

6.3 คำสั่งทำซ้ำ While

คำสั่งทำซ้ำ while มีลักษณะของการวนเพื่อทำคำสั่งที่อยู่ภายในช่วงของการทำซ้ำ เหมาะสำหรับการเขียนโปรแกรมที่ต้องการเขียนคำสั่งบางคำสั่งซ้ำกัน โดยไม่ต้องพิมพ์คำสั่งนั้นหลายๆ ครั้ง ใช้การเขียนคำสั่งนั้นๆ เพียงครั้งเดียวและให้มีการวนรอบตามจำนวนที่ต้องการ ลักษณะของการทำซ้ำ while เป็นการทำซ้ำแบบ Indefinite loop คือเป็นการทำซ้ำที่ไม่ทราบจำนวนรอบที่แน่นอน พิจารณา Flowchart และรูปแบบคำสั่ง while ตามรูปต่อไปนี้

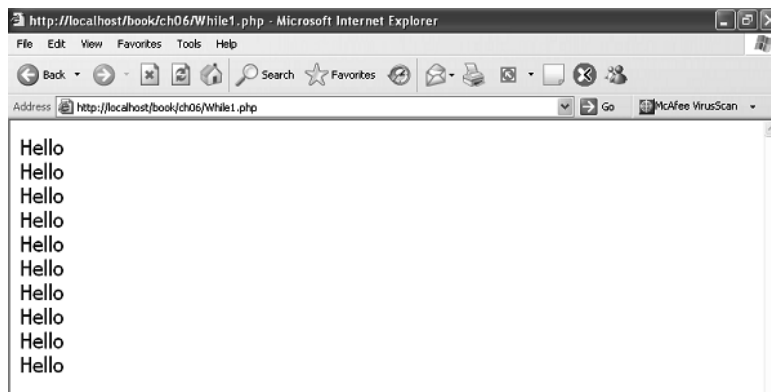


รูปที่ 6.9 Flowchart และรูปแบบคำสั่ง While

จากรูปที่ 6.9 คำสั่ง while จะมีการตรวจสอบเงื่อนไข (Condition) หลัง while ว่าเป็นจริงหรือเท็จ ถ้าเงื่อนไขมีค่าเป็นจริงก็จะทำคำสั่ง (Statement) ที่อยู่ภายในเครื่องหมายปีกกาเปิดและปิด แต่ถ้าเงื่อนไขมีค่าเป็นเท็จจะหยุดการทำซ้ำ คือลงไปทำคำสั่งข้อถัดไปของเครื่องหมายปีกกาปิด พิจารณาโปรแกรมต่อไปนี้โดยเป็นโปรแกรมที่ทำการพิมพ์ข้อความว่า "Hello" จำนวน 10 บรรทัด ดังนี้

```
While1.php
<?
    $count = 1;
    while ($count <= 10)
    {
        echo "Hello <br>";
        $count = $count + 1;
    }
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 6.10 ผลลัพธ์จากการแสดงผลของโปรแกรม While1.php

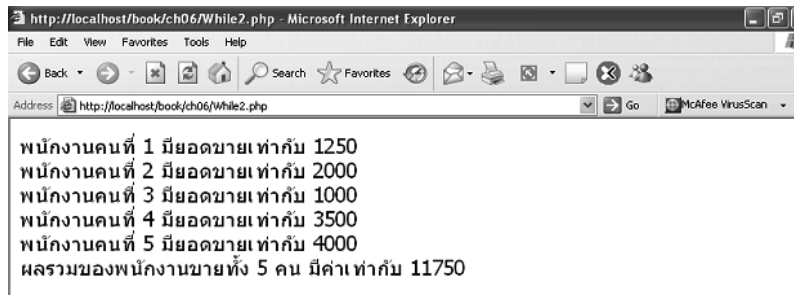
จากโปรแกรมข้างต้นจะกำหนดเงื่อนไขในการทำซ้ำว่าตัวแปร \$count มีค่าน้อยกว่าหรือเท่ากับ 10 โปรแกรมจะทำการพิมพ์ข้อความ "Hello" แต่ถ้าตัวแปร \$count มีค่าไม่น้อยกว่าหรือเท่ากับ 10 คือมีค่ามากกว่า 10 โปรแกรมจะหยุดการทำซ้ำทันที

ตัวอย่างถัดไปจะใช้คำสั่งทำซ้ำ while ในการหาผลรวมของยอดขายของพนักงานจำนวน 5 คน โดยยอดขายของพนักงานขายแต่ละคนจัดเก็บไว้ในตัวแปรอาร์เรย์ หลังจากนั้นจะใช้คำสั่งทำซ้ำ while เพื่อหาผลรวมของยอดขายของพนักงานทั้ง 5 คน ดังโปรแกรมต่อไปนี้

While2.php

```
<?
    $sell = array(1250, 2000, 1000, 3500, 4000);
    $count = 0;
    while ($count <= 4)
    {
        $sellno = $count + 1;
        echo "พนักงานคนที่ $sellno มียอดขายเท่ากับ $sell[$count] <br>";
        $sum = $sum + $sell[$count];
        $count = $count + 1;
    }
    echo "ผลรวมของพนักงานขายทั้ง 5 คน มีค่าเท่ากับ $sum";
?>
```


ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้

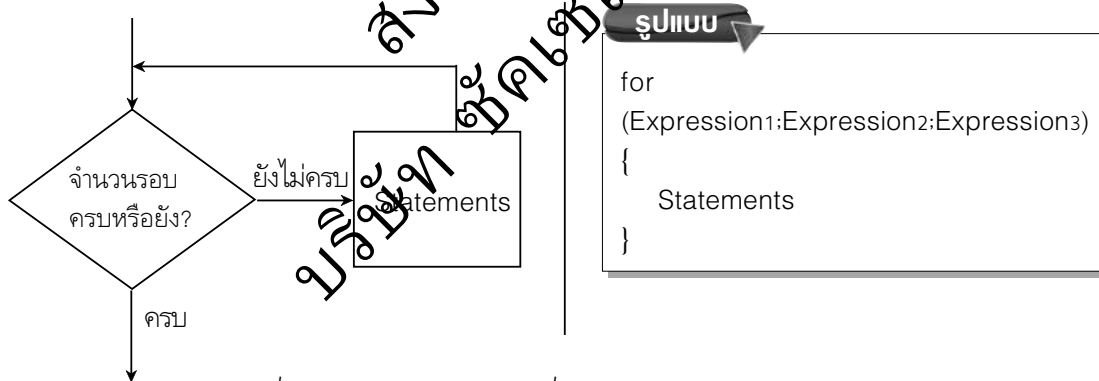


รูปที่ 6.11 ผลลัพธ์จากการแสดงผลของโปรแกรม While2.php

6.4 คำสั่งทำซ้ำ For

จากหัวข้อที่ผ่านมาเราได้รู้จักคำสั่งทำซ้ำ while ไปแล้ว ซึ่งเป็นคำสั่งทำซ้ำในลักษณะ Indefinite loop คือไม่ทราบจำนวนรอบที่แน่นอนของการทำซ้ำ ทราบแต่เพียงว่า จะทำการวนซ้ำก็ต่อเมื่อเงื่อนไขในการพิจารณาเป็นจริง แต่ถ้าเงื่อนไขเป็นเท็จก็จะหยุดการทำซ้ำ แต่สำหรับคำสั่ง for เป็นคำสั่งทำซ้ำในลักษณะ Definite loop คือทราบรอบที่แน่นอนว่าซ้ำกี่รอบ

ลักษณะการทำงานของคำสั่งทำซ้ำ for แสดงด้วย Flowchart พร้อมทั้งรูปแบบคำสั่ง ดังนี้



รูปที่ 6.12 Flowchart และรูปแบบคำสั่ง For

จากรูปที่ 6.12 รูป Flowchart จะมีการตรวจสอบการทำซ้ำว่าครบจำนวนรอบที่กำหนดหรือไม่ ถ้ายังไม่ครบก็จะเข้าไปทำคำสั่ง (Statements) แต่ถ้าครบจำนวนรอบที่กำหนดแล้วในรอบถัดไปก็จะไม่เข้าไปทำคำสั่ง แต่จะลงไปที่คำสั่งที่อยู่ถัดลงไปของวงเล็บปีกกาปิด

ในส่วนของรูปแบบคำสั่ง for จะมีการกำหนดนิพจน์ (Expression) ทั้งหมด 3 ชุด นิพจน์ชุดที่ 1 (Expression1) จะเป็นนิพจน์สำหรับการระบุค่าเริ่มต้นของการทำซ้ำ หรือค่าเริ่มต้นตัวนับรอบในการทำซ้ำ นิพจน์ชุดที่ 2 (Expression2) จะเป็นนิพจน์สำหรับการกำหนดเงื่อนไขในการทำซ้ำ โดยถ้านิพจน์นี้มีค่าเป็นจริงก็จะดำเนินการทำซ้ำต่อไปคือ ทำคำสั่ง (Statements) ที่อยู่ภายในวงเล็บปีกกาเปิดปิด และนิพจน์ชุดที่ 3 (Expression3) จะเป็นนิพจน์ที่ใช้สำหรับการเพิ่มหรือลดค่าตัวนับรอบในการทำซ้ำ พิจารณาโปรแกรมต่อไปนี้ โดยเป็นโปรแกรมที่ทำการพิมพ์ข้อความว่า "Hello" จำนวน 10 บรรทัด (เหมือนกับตัวอย่างที่เคยกล่าวถึงในหัวข้อคำสั่งทำซ้ำ while) ดังนี้

For1.php

```
<?
    for($count=1; $count<=10; $count++)
    {
        echo "Hello <br>";
    }
?>
```

จากโปรแกรมข้างต้นนี้ได้กำหนดนิพจน์ค่าเริ่มต้นของตัวนับไว้ที่ตัวแปร \$count ให้เริ่มต้นที่ 1 นิพจน์ในการเปรียบเทียบเงื่อนไข (นิพจน์ที่ 2) ได้กำหนดว่าจะทำคำสั่งในการทำซ้ำก็ต่อเมื่อค่าของตัวแปร \$count มีค่าน้อยกว่าหรือเท่ากับ 10 ถ้าค่าของตัวแปร \$count มีค่ามากกว่า 10 ก็จะลงไปทำคำสั่งที่อยู่ถัดจากคำสั่ง for ลงไป และในส่วนของนิพจน์สุดท้ายใช้ในการเพิ่มหรือลดค่าตัวนับ ในที่นี้ได้กำหนดให้เพิ่มค่าตัวแปร \$count ทีละ 1 ผลลัพธ์จากการประมวลผลโปรแกรมนี้จะมีผลลัพธ์เหมือนกับในรูปที่ 6.10

ตัวอย่างถัดไปจะใช้คำสั่ง for สำหรับการทำซ้ำเพื่อคำนวณหาผลบวกเลข 1-100 คือ 1+2+3+...+100 โปรแกรมจะสามารถเขียนได้ดังนี้

For2.php

```
<?
    $sum = 0;
    for($count=1; $count<=100; $count++)
    {
        $sum = $sum + $count;
    }
    echo "ผลบวกของเลข 1-100 คือ $sum";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้

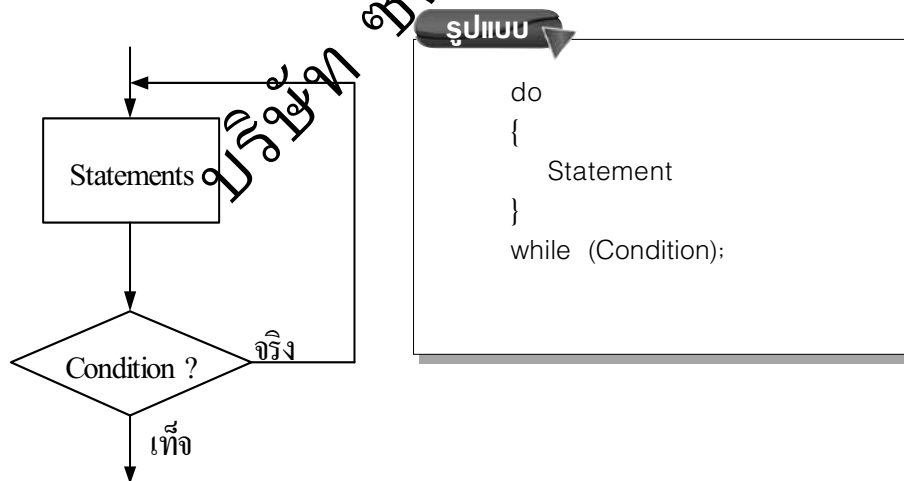


รูปที่ 6.13 แสดงผลลัพธ์ของโปรแกรม For2.php

จากโปรแกรมและผลลัพธ์ข้างต้น จะมีการทำซ้ำทั้งหมด 100 รอบ และแต่ละรอบจะมีการคำนวณหาผลรวมในลักษณะของการบวกแบบสะสม เก็บไว้ในตัวแปร \$sum คือ นำผลลัพธ์จากรอบที่ผ่านมาบวกกับค่าในรอบใหม่ และทำเช่นนี้ไปจนกระทั่งครบ 100 รอบ ผลลัพธ์ของตัวแปร \$sum หลังจากทำครบ 100 รอบแล้วจะเป็นค่าผลรวมของเลข 1-100

6.5 คำสั่งทำซ้ำ Do...while

คำสั่งทำซ้ำ do...while นี้มีลักษณะการทำงานแบบ Indefinite loop เหมือนกับคำสั่งทำซ้ำ while คือ เป็นการทำซ้ำที่ไม่ทราบจำนวนรอบที่แน่นอน ลักษณะการทำงานสามารถอธิบายได้ด้วย Flowchart รวมทั้งรูปแบบของคำสั่งเป็นดังรูปต่อไปนี้



รูปที่ 6.14 Flowchart และรูปแบบคำสั่ง Do...while

ตามรูปที่ 6.14 ลักษณะในการทำงานจะคล้ายกับคำสั่ง while คือ จะมีเงื่อนไข (Condition) ในการตัดสินใจว่าจะทำซ้ำต่อไปหรือไม่ ถ้าเงื่อนไขเป็นจริงก็จะวนกลับขึ้นไปทำคำสั่ง (Statements) ภายใต้อีก แต่ถ้าเงื่อนไขเป็นเท็จก็จะออกจากการทำซ้ำ

ส่วนรูปแบบของคำสั่ง do...while ก็คล้ายกับคำสั่ง while เพียงแต่ว่าตำแหน่งในการวางเงื่อนไขจะแตกต่างกัน คือ คำสั่ง while จะวางในส่วนก่อนของเงื่อนไขตอนต้นของการทำซ้ำ แต่คำสั่ง do...while จะวางส่วนก่อนของเงื่อนไขไว้ตอนท้ายของการทำซ้ำ ซึ่งข้อแตกต่างนี้จะมีผลทำให้การทำซ้ำของคำสั่ง do...while มีการทำคำสั่งภายในวงเล็บปีกกาเปิดปิดอย่างน้อย 1 ครั้งเสมอ แล้วจึงค่อยตรวจสอบเงื่อนไข

พิจารณาโปรแกรมในการพิมพ์ข้อความว่า "Hello" จำนวน 10 บรรทัด เพื่อเปรียบเทียบให้เห็นถึงข้อแตกต่างในการใช้คำสั่งทำซ้ำทั้ง 3 คำสั่ง คือ while, for และ do...while โปรแกรมจะมีลักษณะดังนี้

do_while1.php

```
<?
    $count = 1;
    do
    {
        echo "Hello <br>";
        $count = $count + 1;
    } while($count<=10);
?>
```

6.6 คำสั่งออกจากการทำงาน

คำสั่งทำซ้ำ while, for, do...while และคำสั่งตรวจสอบเงื่อนไข switch จะทำการประมวลผลตามเงื่อนไขที่ได้กำหนดไว้ แต่ในบางครั้งที่เราอาจต้องการออกจากการทำงานก่อนที่คำสั่งนั้นๆ จะประมวลผลเสร็จ โดยคำสั่งออกจากการทำงานแบ่งออกได้เป็น 2 ลักษณะ คือ

- คำสั่ง break จะออกจากการทำงานของคำสั่งนั้นและลงไปทำคำสั่งที่อยู่ถัดลงไป
- คำสั่ง exit หรือ die จะออกจากโปรแกรมนั้นไปเลย

พิจารณา 2 โปรแกรมต่อไปนี้เพื่อเปรียบเทียบการใช้คำสั่ง break และ exit หรือ die ดังต่อไปนี้

break.php

```
<?
    for($count=1; $count<=10; $count++)
    {
```

```

        echo "$count <br>";
        if ($count == 5)
        {
            break;
        }
    }
    echo "The End ";
?>

```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 6.15 แสดงผลลัพธ์ของโปรแกรม *break.php*

จากโปรแกรมและผลลัพธ์ข้างต้นได้ใช้คำสั่งทำซ้ำ `for` เพื่อวนรอบจำนวน 10 รอบ และแต่ละรอบก็ให้พิมพ์ตัวเลขที่ใช้เป็นตัวแปรในรอบ ตัวแปรพิมพ์ตัวแปร `$count` แต่ในช่วงของการทำซ้ำได้มีคำสั่ง `if` เพื่อเปรียบเทียบว่าถ้าตัวแปร `$count` มีค่าเท่ากับ 5 ให้ทำการออกจากคำสั่งทำซ้ำ `for` แต่ถ้าตัวแปร `$count` ไม่เท่ากับ 5 ก็ยังคงวนกลับไปทำซ้ำอีก ดังนั้นโปรแกรมจะทำการพิมพ์ตัวเลข 1-5 ในแต่ละบรรทัด หลังจากนั้นก็จะออกจากคำสั่ง `for` ไปเจอคำสั่งให้พิมพ์ข้อความว่า "The End" และสุดท้ายก็จบโปรแกรม

ตัวอย่างต่อไปเราจะแก้ไขโปรแกรมโดยเปลี่ยนจากคำสั่ง `break` เป็น `exit` แทน โปรแกรมที่ได้เป็นดังนี้

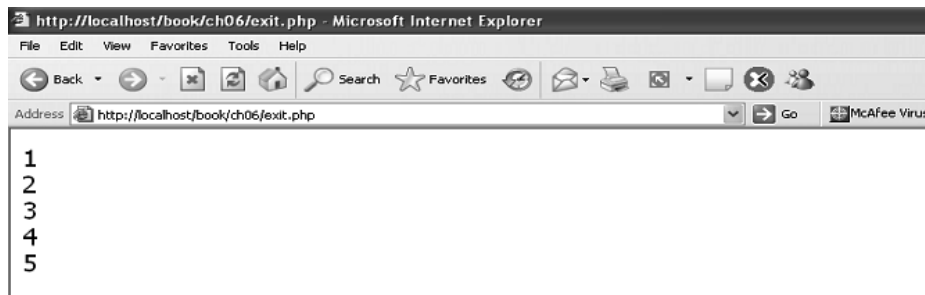
exit.php

```

<?
    for($count=1; $count<=10; $count++)
    {
        echo "$count <br>";
        if ($count == 5)
        {
            exit;
        }
    }
    echo "The End ";
?>

```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็น ดังนี้



รูปที่ 6.16 แสดงผลลัพธ์ของโปรแกรม exit.php

จากโปรแกรมเรากำหนดให้ทำซ้ำ 10 รอบ และแต่ละรอบให้พิมพ์ตัวเลขที่ใช้นับรอบ และมีการเปรียบเทียบว่าตัวแปร \$count ที่ใช้นับรอบถ้ามีค่าเท่ากับ 5 ให้ออกจากโปรแกรมไปเลย แต่ถ้าไม่เท่ากับ 5 ก็ทำการวนกลับขึ้นไปทำซ้ำอีก ดังนั้นผลลัพธ์จะทำการพิมพ์ตัวเลข 1-5 ในแต่ละบรรทัดแต่จะไม่พิมพ์ข้อความ "The End" เนื่องจากคำสั่ง exit จะสั่งให้ออกจากโปรแกรมไปเลยซึ่งจะเหลือคำสั่งอยู่หลังจากคำสั่ง for ก็ตาม

สรุป

คำสั่งเพื่อใช้ในการตรวจสอบเงื่อนไข if สามารถแบ่งออกได้เป็น 3 ลักษณะด้วยกัน คือ คำสั่งตรวจสอบเงื่อนไขแบบทางเลือกเดียว, คำสั่งตรวจสอบเงื่อนไขแบบ 2 ทางเลือก และคำสั่งตรวจสอบเงื่อนไขแบบหลายทางเลือก

คำสั่งตรวจสอบเงื่อนไข switch เป็นคำสั่งที่เหมาะสมสำหรับเงื่อนไขที่มีหลายๆ ทางเลือก การใช้คำสั่ง switch จะช่วยให้โปรแกรมที่เขียนขึ้นสั้นกะทัดรัด ทำให้ง่ายต่อการพิจารณาโปรแกรม

คำสั่งทำซ้ำ while มีลักษณะของการวนเพื่อทำคำสั่งที่อยู่ภายในช่วงของการทำซ้ำ เหมาะสำหรับการเขียนโปรแกรมที่ต้องการตรวจสอบบางคำสั่งซ้ำกันหลายหน โดยพิมพ์คำสั่งนั้นเพียงครั้งเดียวและให้มีการวนรอบตามจำนวนที่ต้องการ ลักษณะของการทำซ้ำ while เป็นการซ้ำแบบ Indefinite loop คือเป็นการทำซ้ำที่ไม่ทราบจำนวนรอบที่แน่นอน

คำสั่งทำซ้ำ for เป็นคำสั่งทำซ้ำในลักษณะ Definite loop คือ ทราบจำนวนรอบที่แน่นอนว่าต้องทำการวนซ้ำจำนวนกี่รอบ

คำสั่งทำซ้ำ do...while มีลักษณะการทำซ้ำแบบ Indefinite loop เหมือนกับคำสั่งทำซ้ำ while คือเป็นการทำซ้ำที่ไม่ทราบจำนวนรอบที่แน่นอน แต่ต่างจากคำสั่งทำซ้ำ while ที่คำสั่ง while อาจจะไม่เข้าไปทำคำสั่งในลูปเลยก็ได้ ส่วนคำสั่ง do...while จะต้องเข้าไปทำคำสั่งในลูปอย่างน้อย 1 ครั้งเสมอ

คำสั่งทำซ้ำ while, for, do...while และคำสั่งตรวจสอบเงื่อนไข switch จะทำการประมวลผลตามเงื่อนไขที่ได้กำหนดไว้ แต่ในบางครั้งเราอาจต้องการออกจากการทำงานก่อนที่คำสั่งนั้นๆ จะประมวลผลเสร็จ โดยคำสั่งออกจากการทำงานแบ่งออกได้เป็น 2 ลักษณะ คือ คำสั่ง break จะออกจากการทำงานของคำสั่งนั้นและลง去做คำสั่งที่อยู่ถัดลงไป และคำสั่ง exit หรือ die จะออกจากโปรแกรมนั้นไปเลย

แบบฝึกหัดบทที่ 6

1. ให้เขียนโปรแกรมภาษา PHP เก็บค่าคะแนนสอบของนักเรียนจำนวน 10 คน ไว้ในตัวแปรอาร์เรย์ หลังจากนั้นนำไปเปรียบเทียบว่าคะแนนของนักเรียนแต่ละคนได้เกรดอะไรตามข้อกำหนดดังนี้

80-100	ได้เกรด A
70-79	ได้เกรด B
60-69	ได้เกรด C
50-59	ได้เกรด D
0-49	ได้เกรด F

(ในการเปรียบเทียบสามารถใช้คำสั่ง If หรือ Switch ก็ได้ แต่ห้ามใช้คำสั่งทำซ้ำ)

2. จากโจทย์ข้อ 1 อนุญาตให้ใช้คำสั่งทำซ้ำในการเปรียบเทียบคะแนนสอบของนักเรียนแต่ละคน

3. ให้เขียนโปรแกรมภาษา PHP ในการหาค่าของเลขยกกำลังกำลัง 1 กำลัง 2 และ กำลัง 3 ของตัวเลข 1-10 ตามรูปแบบ ดังนี้ (ใช้คำสั่งทำซ้ำมาประยุกต์ใช้ช่วย)

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

4. ให้เขียนโปรแกรมภาษา PHP โดยใช้คำสั่งทำซ้ำ ในการหาค่าผลบวกรวมของเลขคู่ และค่าผลบวกรวมของเลขคี่ตั้งแต่ตัวเลข 1 -100

.....

5. ให้เขียนโปรแกรมภาษา PHP โดยใช้คำสั่งทำซ้ำและ Tag <H1> ถึง <H6> เพื่อพิมพ์ข้อความว่า "Hello" ในลักษณะดังนี้

Hello

Hello

Hello

Hello

Hello

Hello

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

การสร้างฟังก์ชันของ PHP

ในการเขียนโปรแกรมแบบโครงสร้าง (Structured programming) คุณสมบัติหนึ่งที่สำคัญก็คือการเขียนโปรแกรมแบบโปรแกรมย่อย ซึ่งจะช่วยให้การจัดการต่างๆ ทำได้ง่ายยิ่งขึ้น ไม่ว่าจะเป็นการเพิ่ม ลบ หรือแก้ไขโปรแกรม เนื่องจากแต่ละโปรแกรมย่อยจะแบ่งหน้าที่การทำงานอย่างชัดเจน นอกจากนี้การเขียนโปรแกรมแบบโปรแกรมย่อยจะช่วยประหยัดเวลาในการพัฒนาโปรแกรม เนื่องจากโปรแกรมในส่วนที่มีการทำงานซ้ำๆ กันสามารถเขียนในลักษณะโปรแกรมย่อย และเรียกชื่อโปรแกรมย่อยนั้นให้ทำงานได้ทันที และการเขียนเป็นโปรแกรมย่อยสามารถแจกจ่ายงานให้แก่ทีมผู้พัฒนาได้ ทั้งหมดนี้ทำให้ประหยัดเวลาในการพัฒนาไปได้มากทีเดียว

จุดประสงค์ทั่วไป

1. เพื่อให้ทราบวิธีการประกาศฟังก์ชัน และสร้างฟังก์ชันได้ถูกต้อง
2. เพื่อให้สามารถสร้างฟังก์ชันแบบไม่มีการส่งผ่านค่าได้
3. เพื่อให้สามารถสร้างฟังก์ชันแบบที่มีการส่งผ่านค่าได้
4. เพื่อให้สามารถสร้างฟังก์ชันแบบที่มีการคืนค่ากลับได้
5. เพื่อให้ทราบและแยกแยะความแตกต่างของตัวแปรประเภท Local และ Global
6. เพื่อให้ทราบและแยกแยะความแตกต่างของการส่งผ่านค่า ประเภทส่งผ่านด้วยค่า และส่งผ่านด้วยการอ้างอิงในหน่วยความจำ

จุดประสงค์เชิงพฤติกรรม

1. สามารถประกาศฟังก์ชันได้ถูกต้อง
2. สามารถเขียนโปรแกรมโดยสร้างฟังก์ชันแบบไม่มีการส่งผ่านค่าได้
3. สามารถเขียนโปรแกรมโดยสร้างฟังก์ชันแบบมีการส่งผ่านค่าได้
4. สามารถเขียนโปรแกรมโดยสร้างฟังก์ชันแบบมีการคืนค่ากลับได้
5. บอกความหมายของตัวแปร Local และ Global
6. บอกความแตกต่างระหว่างการส่งผ่านค่า ประเภทส่งผ่านด้วยค่า และส่งผ่านด้วยการอ้างอิง

เนื้อหาสาระ

1. การประกาศฟังก์ชัน
2. การใช้งานฟังก์ชันโดยไม่มีการส่งผ่านค่า
3. การใช้งานฟังก์ชันโดยมีการส่งผ่านค่า
4. การคืนค่ากลับของฟังก์ชัน
5. ประเภทของตัวแปร และขอบเขตของตัวแปร
6. ประเภทของการส่งผ่านค่า

7.1 การประกาศฟังก์ชัน

ก่อนที่เราจะใช้งานฟังก์ชันใดๆ เราจะต้องสร้างฟังก์ชันขึ้นก่อน หลังจากนั้นค่อยเรียกชื่อฟังก์ชันนั้นเมื่อต้องการใช้งาน ซึ่งการสร้างฟังก์ชันนี้เราอาจจะเรียกว่าเป็นการประกาศฟังก์ชัน โดยรูปแบบของการประกาศฟังก์ชันในภาษา PHP เป็นดังนี้

รูปแบบ

```
function ชื่อฟังก์ชัน([argument1, argument2, ...])
{
    Statement;
}
```

รายละเอียดของการประกาศฟังก์ชัน เป็นดังนี้

ชื่อฟังก์ชัน	เป็นชื่อที่ตั้งขึ้นเพื่อเรียกใช้งานฟังก์ชัน
argument1, argument2, ...	เป็นตัวแปรที่ตั้งขึ้นเพื่อรอรับค่าจากกรณีที่ฟังก์ชันส่งผ่านค่าเข้ามาประมวลผลภายในฟังก์ชัน โดยจะมีอาร์กิวเมนต์ก็ได้ แต่ถ้าในการใช้งานฟังก์ชันไม่มีการส่งผ่านค่าในส่วนนี้ก็ไม่ต้องการอาร์กิวเมนต์ เพียงวงเล็บเปิด-ปิดไว้เท่านั้น
Statement	เป็นคำสั่งที่ประมวลผลภายในฟังก์ชัน

ยกตัวอย่างการประกาศฟังก์ชันชื่อว่า asterisk โดยเป็นฟังก์ชันที่ใช้พิมพ์เครื่องหมายดอกจันจำนวน 30 ครั้ง รายละเอียดของฟังก์ชันนี้เป็น ดังนี้

```
function asterisk()
{
    echo "*****";
}
```

7.2 การใช้งานฟังก์ชันโดยไม่มีการส่งผ่านค่า

หลังจากประกาศฟังก์ชันแล้ว ในการใช้งานฟังก์ชันก็เพียงเรียกชื่อฟังก์ชันนั้น แต่ให้สังเกตว่าฟังก์ชันที่จะเรียกใช้งานนั้นมีการส่งผ่านค่าหรือไม่ ในกรณีที่ไม่มีค่าก็เพียงเรียกชื่อฟังก์ชันตามด้วยเครื่องหมายวงเล็บเปิด-ปิดเท่านั้น ตัวอย่างถัดไปจะเรียกใช้ฟังก์ชัน asterisk ที่ได้ประกาศไว้ในหัวข้อที่ผ่านมา ดังนี้

function_nopass1.php

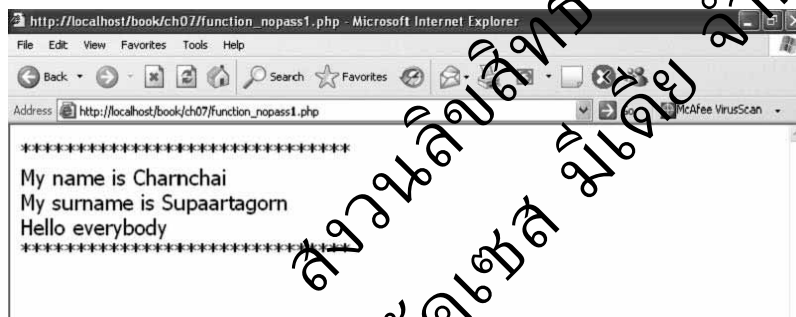
```
<?
function asterisk()
{
    echo "***** <br>";
}
asterisk();
echo "My name is Charnchai <br>" ;
echo "My surname is Supaartagorn <br>";
echo "Hello everybody <br>";
asterisk();
?>
```

นิยามฟังก์ชัน

เรียกใช้ฟังก์ชัน

เรียกใช้ฟังก์ชัน

ผลลัพธ์ของโปรแกรมข้างต้นเป็น ดังนี้



รูปที่ 7.1 แสดงผลลัพธ์ของโปรแกรม function_nopass1.php

จากโปรแกรมและผลลัพธ์ของโปรแกรมนี้มีการเรียกใช้ฟังก์ชัน asterisk 2 ครั้ง คือเรียกใช้งานในตอนเริ่มต้น และตอนท้ายของผลลัพธ์ โดยที่เราไม่ต้องเสียเวลาเขียนโปรแกรมเพื่อสั่งให้พิมพ์เครื่องหมายดอกจันจำนวน 30 ครั้งใหม่ เราเพียงเรียกชื่อฟังก์ชันให้ทำงานเท่านั้น

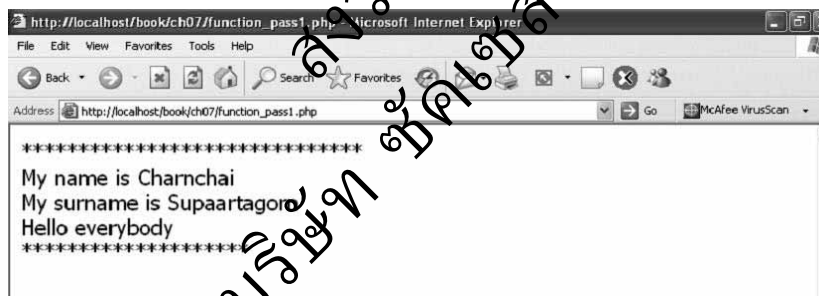
7.3 การใช้งานฟังก์ชันโดยมีการส่งผ่านค่า

เพื่อให้การใช้งานฟังก์ชันมีความยืดหยุ่นมากขึ้น ในบางครั้งเราอาจจะสร้างฟังก์ชันในลักษณะของการส่งผ่านค่า จากตัวอย่างโปรแกรมในหัวข้อที่ผ่านมาถ้าเราต้องการให้ฟังก์ชัน asterisk สามารถพิมพ์เครื่องหมายดอกจันจำนวนเท่าไรก็ได้ขึ้นอยู่กับผู้ใช้งานต้องการ เราจะสามารถปรับปรุงฟังก์ชัน และโปรแกรมในการเรียกใช้งานได้ดังนี้

function_pass1.php

```
<?
function asterisk($no)
{
    $sign = "";
    for($count=1; $count<=$no; $count++)
    {
        $sign = $sign."*";
    }
    echo "$sign <br>";
}
asterisk(30);
echo "My name is Charnchai <br>" ;
echo "My surname is Supaartagorn <br>";
echo "Hello everybody <br>";
asterisk(20);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็น ดังนี้



รูปที่ 7.2 แสดงผลลัพธ์ของโปรแกรม function_pass1.php

จากโปรแกรมและผลลัพธ์ข้างต้นนี้ได้ทำการปรับปรุงฟังก์ชัน asterisk โดยเขียนฟังก์ชันในลักษณะที่มีการส่งผ่านค่า ให้กับฟังก์ชันซึ่งมีอาร์กิวเมนต์ \$no ทำหน้าที่เป็นตัวรับค่าจำนวนที่ต้องการให้พิมพ์เครื่องหมายดอกจัน ภายในฟังก์ชันก็นำอาร์กิวเมนต์ \$no นี้ไปเข้าเงื่อนไขเพื่อทำซ้ำของคำสั่ง For กำหนดให้ตัวแปร \$sign เป็นตัวแปรที่ใช้ต่อสตริงเครื่องหมายดอกจันเท่ากับจำนวนค่าของอาร์กิวเมนต์

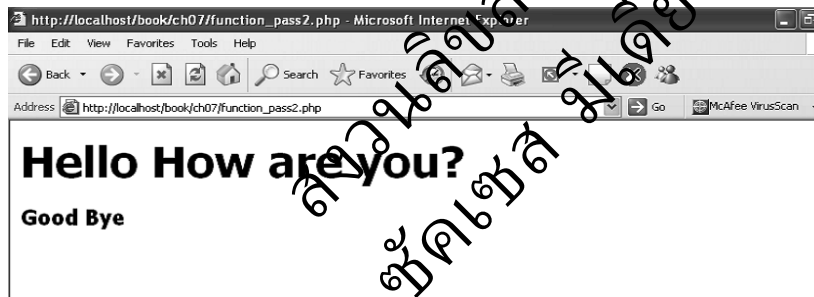
ในส่วนของการเรียกใช้งานฟังก์ชันก็มีการเรียกใช้ 2 ครั้ง ครั้งแรกส่งผ่านค่าตัวเลข 30 ดังนั้นผลลัพธ์ที่ได้ก็จะทำการพิมพ์เครื่องหมายดอกจันจำนวน 30 ครั้ง ครั้งที่ 2 ส่งผ่านค่าตัวเลข 20 ดังนั้นผลลัพธ์ที่ได้ก็จะทำการพิมพ์เครื่องหมายดอกจันจำนวน 20 ครั้ง

ตัวอย่างถัดไปจะเป็นการแสดงผลฟังก์ชันที่มีส่งผ่านค่า 2 จำนวน โดยค่าแรกเป็นสตริง และค่าที่ 2 เป็นตัวเลข รายละเอียดของโปรแกรม เป็นดังนี้

function_pass2.php

```
<?
function show($message, $size)
{
    echo "<H".$size.">";
    echo "$message <br>";
    echo "</H".$size.">";
}
show("Hello How are you?",1);
show("Good Bye",4);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็น ดังนี้



รูปที่ 7.3 แสดงผลลัพธ์ของโปรแกรม function_pass2.php

จากโปรแกรมข้างต้นได้ทำการประกาศฟังก์ชันชื่อว่า show ซึ่งฟังก์ชันนี้มีอาร์กิวเมนต์ 2 ตัว คือ \$message และ \$size โดยอาร์กิวเมนต์ \$message ใ้รับข้อความสตริงใดๆ และอาร์กิวเมนต์ \$size ใ้รับค่าตัวเลขใดๆ

ในการเรียกใช้งานฟังก์ชันของโปรแกรมนี้มีการเรียกใช้งาน 2 ครั้ง ครั้งแรกส่งผ่านข้อความ "Hello How are you?" และตัวเลข 1 ซึ่งภายในฟังก์ชันก็ทำการพิมพ์ข้อความนี้ และนำตัวเลขที่ผ่านค่าเข้ามาไปกำหนดขนาดของตัวหนังสือ จากผลลัพธ์จะเห็นว่าข้อความ "Hello How are you?" มีขนาดเท่ากับ 1 ซึ่งเป็นขนาดที่ใหญ่ที่สุด ส่วนการเรียกใช้ฟังก์ชันครั้งที่ 2 มีการส่งผ่านข้อความ "Good Bye" และตัวเลข 4 ดังนั้นผลลัพธ์ที่ได้ก็คือการพิมพ์ข้อความ "Good Bye" มีขนาดเท่ากับ 4 ซึ่งเล็กกว่าข้อความแรก

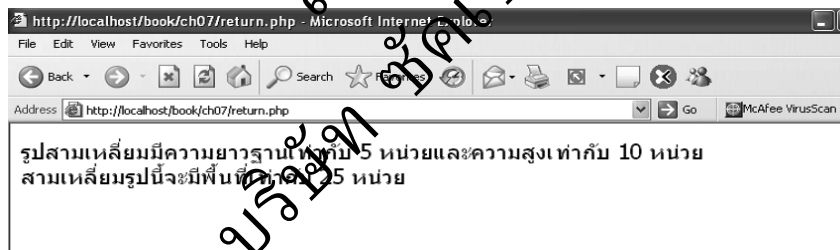
7.4 การคืนค่ากลับของฟังก์ชัน

ในการใช้งานฟังก์ชันนอกจากเราจะสามารถส่งผ่านค่าใดๆ ไปประมวลผลภายในฟังก์ชันได้แล้ว เรายังสามารถคืนค่าใดๆ กลับมายังส่วนที่เรียกใช้งานฟังก์ชันได้ โดยใช้คำสั่ง return ตามด้วยค่าที่ต้องการคืนกลับ และค่าที่คืนกลับจะถูกแทนที่ในตำแหน่งที่เรียกใช้ฟังก์ชัน

พิจารณาโปรแกรม และผลลัพธ์ต่อไปนี้

```
return.php
<?
function cal_triangle($base,$height)
{
    $area = 0.5*$base*$height;
    return $area;
}
echo "รูปสามเหลี่ยมมีความยาวฐานเท่ากับ 5 หน่วยและความสูงเท่ากับ 10 หน่วย
<br>";
echo "สามเหลี่ยมรูปนี้จะมีพื้นที่เท่ากับ .cal_triangle(5,10)." หน่วย";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นมีดังนี้



รูปที่ 7.4 แสดงผลลัพธ์ของโปรแกรม return.php

7.5 ประเภทของตัวแปร และขอบเขตของตัวแปร

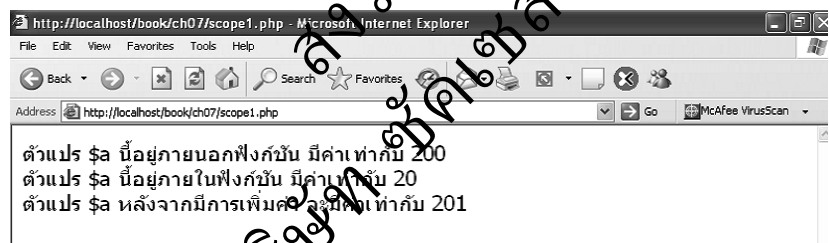
ในการใช้งานฟังก์ชันเราจะเห็นว่ามีการใช้งานตัวแปรภายในฟังก์ชัน และในบางครั้งชื่อตัวแปรที่ใช้ภายในฟังก์ชันมีชื่อเดียวกับชื่อตัวแปรที่ใช้อยู่ภายนอกฟังก์ชัน ในกรณีนี้ชื่อตัวแปรนี้เราจะถือว่าเป็นตัวแปรตัวเดียวกันหรือไม่ ในการตอบคำถามนี้เราจะต้องทำความรู้จักประเภทของตัวแปรซึ่งแบ่งออกเป็น 2 ประเภท คือตัวแปรประเภท Local และ Global ความหมายของประเภทตัวแปรทั้ง 2 คือ

- **ตัวแปร local** คือ ตัวแปรที่มีขอบเขตการใช้งานภายในบริเวณใดบริเวณหนึ่งเท่านั้น เช่น มีขอบเขตการใช้งานเฉพาะภายในฟังก์ชันเท่านั้น ถ้ามีตัวแปรที่ใช้งานภายในฟังก์ชัน และใช้งานภายนอกเป็นชื่อตัวแปรเดียวกัน เราจะถือว่าตัวแปร 2 ตัวนี้เป็นตัวแปรคนละตัวกัน พิจารณาโปรแกรมต่อไปนี้

scope1.php

```
<?
function testvar()
{
    $a = 20;
    echo "ตัวแปร \$a นี้อยู่ภายในฟังก์ชัน มีค่าเท่ากับ $a <br>";
}
$a = 200;
echo "ตัวแปร \$a นี้อยู่ภายนอกฟังก์ชัน มีค่าเท่ากับ $a <br>";
testvar();
$a++;
echo "ตัวแปร \$a หลังจากมีการเพิ่มค่า จะมีค่าเท่ากับ $a ";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นนี้เป็นดังนี้



รูปที่ 7.5 แสดงผลลัพธ์ของโปรแกรม scope1.php

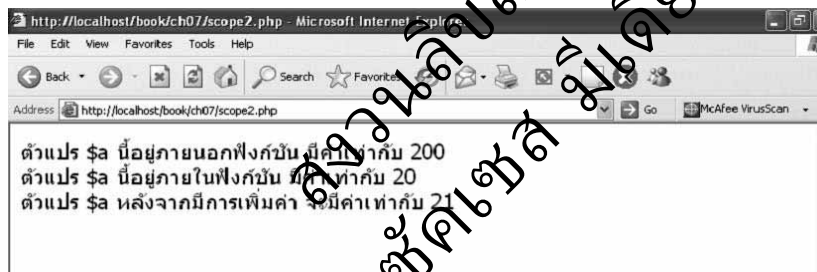
จากตัวอย่างตัวแปร \$a ในฟังก์ชันจะไม่ใช้ตัวเดียวกับตัวแปร \$a นอกฟังก์ชัน และขอบเขตของตัวแปร \$a ในฟังก์ชันจะใช้ถึงแค่จบฟังก์ชันเท่านั้น

- **ตัวแปร Global** คือ ตัวแปรที่มีขอบเขตการใช้งานทั่วทั้งโปรแกรม ไม่ว่าจะใช้ตัวแปรประเภทนี้บริเวณใดก็ตามทุกส่วนของโปรแกรมจะรู้จักตัวแปรตัวนี้ ในการประกาศให้ตัวแปรใดเป็นประเภท global เราจะใช้คำสั่ง Global วางไว้หน้าชื่อตัวแปรที่ต้องการ พิจารณาโปรแกรมต่อไปนี้

scope2.php

```
<?
function testvar()
{
    global $a;
    $a = 20;
    echo "ตัวแปร \$a นี้อยู่ภายในฟังก์ชัน มีค่าเท่ากับ $a <br>";
}
$a = 200;
echo "ตัวแปร \$a นี้อยู่ภายนอกฟังก์ชัน มีค่าเท่ากับ $a <br>";
testvar();
$a++;
echo "ตัวแปร \$a หลังจากมีการเพิ่มค่า จะมีค่าเท่ากับ $a ";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นนี้เป็น ดังนี้



รูปที่ 7.6 แสดงผลลัพธ์ของโปรแกรม scope2.php

7.6 ประเภทของการส่งผ่านค่า

ในการส่งผ่านค่าใดๆ ไปยังฟังก์ชัน หลังจากภายในฟังก์ชันประมวลผลเสร็จ และมีการเปลี่ยนค่าที่ผ่านเข้ามาเกิดขึ้น ถ้าหากกลับไปยังส่วนที่เรียกใช้งานฟังก์ชันแล้วค่าของตัวแปรที่ส่งผ่านค่าไปยังฟังก์ชันจะถูกเปลี่ยนตามไปด้วยหรือไม่ ค่าตอบก็คือขึ้นอยู่กับประเภทของการส่งผ่านค่า โดยการส่งผ่านค่ามี 2 ประเภท ดังนี้

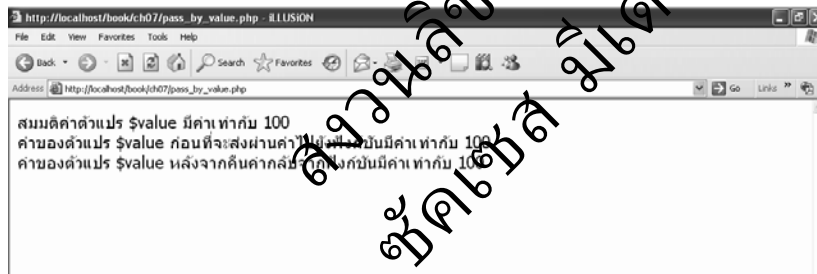
- **การส่งผ่านด้วยค่าของตัวแปร (Pass by Value)** การเปลี่ยนแปลงใดๆ ของค่าอาร์กิวเมนต์จะไม่มีผลต่อค่าของตัวแปรที่ส่งผ่านค่าไปยังฟังก์ชัน พิจารณาโปรแกรมต่อไปนี้

pass_by_value.php

```
<?
function pass($val)
{
    $val = $val + 10;
    return $val;
}

$value = 100;
echo "สมมติค่าตัวแปร \$value มีค่าเท่ากับ $value <br>";
echo "ค่าของตัวแปร \$value ก่อนที่จะส่งผ่านค่าไปยังฟังก์ชันมีค่าเท่ากับ $value <br>";
pass($value);
echo "ค่าของตัวแปร \$value หลังจากคืนค่ากลับจากฟังก์ชันมีค่าเท่ากับ $value <br>";
?>
```

ผลลัพธ์ของโปรแกรมนี้เป็นดังนี้



รูปที่ 7.7 แสดงผลลัพธ์ของโปรแกรม pass_by_value.php

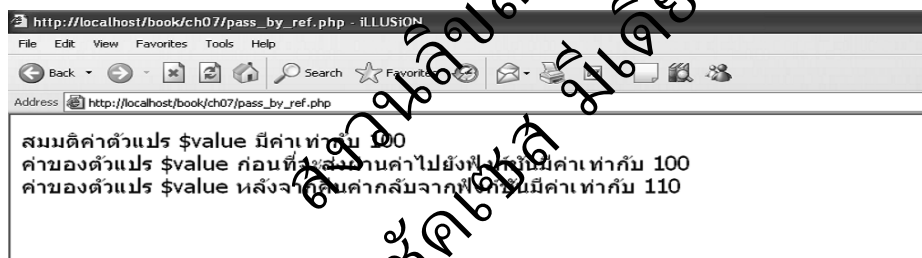
ผลลัพธ์ของโปรแกรมนี้ เนื่องจากการส่งผ่านค่าของตัวแปร \$value ก่อนส่งผ่านไปยังฟังก์ชันและหลังจากคืนค่ากลับจากฟังก์ชันจะมีค่าเท่ากัน

• **การส่งผ่านด้วยการอ้างอิงในหน่วยความจำ (Pass by Reference)** การเปลี่ยนแปลงใดๆ ของค่าอาร์กิวเมนต์ จะมีผลต่อค่าของตัวแปรที่ส่งผ่านค่าไปยังฟังก์ชัน โดยจะใช้สัญลักษณ์ & นำหน้าตัวแปรที่จะส่งผ่านไปยังฟังก์ชัน พิจารณาโปรแกรมต่อไปนี้

pass_by_ref.php

```
<?
function pass($val)
{
    $val = $val + 10;
    return $val;
}
$value = 100;
echo "สมมติค่าตัวแปร \$value มีค่าเท่ากับ $value <br>";
echo "ค่าของตัวแปร \$value ก่อนที่จะส่งผ่านค่าไปยังฟังก์ชันมีค่าเท่ากับ $value
<br>";
pass(&$value);
echo "ค่าของตัวแปร \$value หลังจากคืนค่ากลับจากฟังก์ชันมีค่าเท่ากับ $value
<br>";
?>
```

ผลลัพธ์ของโปรแกรมนี้เป็นดังนี้



รูปที่ 7.8 แสดงผลลัพธ์ของโปรแกรม pass_by_ref.php

ผลลัพธ์ของโปรแกรมนี้ เนื่องจากเป็นการส่งผ่านด้วยการอ้างอิงในหน่วยความจำ ดังนั้นค่าของตัวแปร \$value ก่อนส่งผ่านไปยังฟังก์ชันและหลังจากคืนค่ากลับจากฟังก์ชันจะมีค่าเปลี่ยนแปลงไป โดยค่าของอาร์กิวเมนต์ \$val ภายในฟังก์ชันจะทำการบวกค่าเพิ่มขึ้นอีก 10 ทำให้ค่าของอาร์กิวเมนต์เปลี่ยนจาก 100 เป็น 110 และหลังจากคืนค่ากลับมาที่จุดเรียกใช้ฟังก์ชันค่าของตัวแปร \$value ก็จะถูกเปลี่ยนแปลงตามไปด้วย

สรุป

ก่อนที่เราจะใช้งานฟังก์ชันใดๆ เราจะต้องสร้างฟังก์ชันขึ้นก่อน หลังจากนั้นค่อยเรียกชื่อฟังก์ชันนั้นเมื่อต้องการใช้งาน ซึ่งการสร้างฟังก์ชันนี้เราอาจจะเรียกว่าเป็นการประกาศฟังก์ชัน และหลังจากประกาศฟังก์ชันแล้ว ใ้การใช้งานฟังก์ชันก็เพียงเรียกชื่อฟังก์ชันนั้น โดยสามารถแบ่งประเภทของการเรียกใช้ฟังก์ชันได้เป็น แบบไม่มีการส่งผ่านค่า และแบบที่มีการส่งผ่านค่า และนอกจากนั้นหลังจากที่ทำการประมวลผลภายในฟังก์ชันแล้ว ยังแบ่งฟังก์ชันเป็นแบบที่ไม่มีการคืนค่ากลับ และมีการคืนค่ากลับ

ประเภทของตัวแปรแบ่งออกได้เป็น 2 ประเภท คือ Local และ Global โดยที่ตัวแปร Local คือตัวแปรที่มีขอบเขตการใช้งานภายในบริเวณใดบริเวณหนึ่งเท่านั้น ส่วนตัวแปร Global คือตัวแปรที่มีขอบเขตการใช้งานทั่วทั้งโปรแกรม ไม่ว่าจะใช้ตัวแปรประเภทนี้บริเวณใดก็ตามทุกส่วนของโปรแกรมจะรู้จักตัวแปรตัวนี้

ประเภทของการส่งผ่านค่าแบ่งออกเป็น 2 ประเภท คือ การส่งผ่านด้วยค่าของตัวแปร (Pass by Value) คือ การเปลี่ยนแปลงใดๆ ของค่าอาร์กิวเมนต์จะไม่ส่งผลต่อค่าของตัวแปรที่ส่งผ่านค่าไปยังฟังก์ชัน และการส่งผ่านด้วยการอ้างอิงในหน่วยความจำ (Pass by Reference) การเปลี่ยนแปลงใดๆ ของค่าอาร์กิวเมนต์จะมีผลต่อค่าของตัวแปรที่ส่งผ่านค่าไปยังฟังก์ชัน

สงวนลิขสิทธิ์
บริษัท ชัคเชส มีเดีย จำกัด

แบบฝึกหัดบทที่ 7

1. ให้สร้างฟังก์ชันแบบที่มีการรับอาร์กิวเมนต์ 1 จำนวน โดยรับค่าราคาสินค้าเพื่อไปเปรียบเทียบเงื่อนไขในการให้ส่วนลดภายในฟังก์ชัน ดังนี้

ราคาสินค้าน้อยกว่า 5,000 บาท ได้ส่วนลด 3%

ราคาสินค้ามากกว่าหรือเท่ากับ 5,000 บาท ได้ส่วนลด 5%

ให้พิมพ์ค่าส่วนลดที่ได้รับ และราคาสินค้าสุทธิหลังหักส่วนลดแล้วภายในฟังก์ชัน หลังจากนั้นกำหนดตัวแปร $\$price1 = 1000$ และ $\$price2 = 5000$ แล้วเรียกใช้ฟังก์ชัน 2 ครั้ง และแต่ละครั้งให้ส่งผ่านค่าตัวแปรที่ละตัวตามลำดับ)

2. จากโจทย์ข้อ 1 ให้สร้างฟังก์ชันโดยให้มีการคืนค่าส่วนลดที่คำนวณได้กลับคืนมายังโปรแกรมหลักแล้วทำการหารราคาสินค้าสุทธิ หลังจากนั้นค่อยพิมพ์ผลลัพธ์ของส่วนลด และราคาสินค้าสุทธิออกมา (พิมพ์ค่าส่วนลดและราคาสินค้าสุทธินอกฟังก์ชัน)

3. ให้อธิบายความแตกต่างระหว่างประเภทของตัวแปร Local และ Global

4. ให้อธิบายความแตกต่างระหว่างการส่งผ่านด้วยค่าของตัวแปร และการส่งผ่านด้วยการอ้างอิงในหน่วยความจำ

5. กำหนดสูตรในการคำนวณหาค่าฟังก์ชันกำลังสอง ดังนี้

$$d = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

ให้เขียนโปรแกรมภาษา PHP โดยที่ส่วนของโปรแกรมหลักให้กำหนดค่าตัวเลขใดๆให้กับตัวแปร a, b และ c หลังจากนั้นให้ส่งค่าของตัวแปรทั้ง 3 ตัวไปยังฟังก์ชันเพื่อคำนวณหาค่า d และให้ส่งผลลัพธ์ค่า d กลับมาแสดงผลที่โปรแกรมหลัก

หมายเหตุ: ในการหาค่ารากที่ 2 สามารถใช้ฟังก์ชัน `sqrt()` ได้โดยมีรูปแบบการใช้งาน คือ `sqrt(ตัวเลขที่ต้องการหาค่ารากที่ 2)`

ฟังก์ชันของ PHP ที่จัดเตรียมไว้

ในบทที่ผ่านมาเราได้เรียนรู้การสร้างฟังก์ชันไว้ใช้งาน ซึ่งผู้พัฒนาจะต้องเขียนโปรแกรมขึ้นเองทั้งหมด และต้องอาศัยความรู้ในการประกาศฟังก์ชัน การเรียกฟังก์ชันใช้งาน แต่เพื่อความสะดวกโปรแกรมภาษา PHP ได้จัดเตรียมฟังก์ชันสำเร็จรูปต่างๆ ไว้ใช้งานได้โดยที่เราไม่ต้องเขียนโปรแกรมเลย เพียงแต่รู้จักชื่อฟังก์ชัน และการเรียกใช้งานฟังก์ชันให้ถูกต้อง ฟังก์ชันในลักษณะนี้มีชื่อว่า Pre-Defined Function

จุดประสงค์
ทั่วไป

1. เพื่อศึกษาการใช้งานฟังก์ชันสำเร็จรูปต่างๆ ที่จัดเตรียมไว้ให้ (Pre-Defined Function)
2. เพื่อให้ทราบและสามารถนำฟังก์ชันเกี่ยวกับสตริงไปจัดการข้อมูลชนิดสตริงได้
3. เพื่อให้ทราบและสามารถนำฟังก์ชันทางคณิตศาสตร์ไปจัดการข้อมูลทั้งชนิดเลขจำนวนเต็ม และเลขทศนิยมได้
4. เพื่อให้ทราบและสามารถนำฟังก์ชันที่เกี่ยวกับวันและเวลาไปจัดการข้อมูลที่เกี่ยวข้องกับวัน และเวลา เช่น การตั้งวัน เวลาปัจจุบันได้

จุดประสงค์
เชิงปฏิบัติ

1. สามารถเขียนโปรแกรมโดยใช้ฟังก์ชันในกลุ่มสตริงได้
2. สามารถเขียนโปรแกรมโดยใช้ฟังก์ชันในกลุ่มคณิตศาสตร์ได้
3. สามารถเขียนโปรแกรมโดยใช้ฟังก์ชันในกลุ่มวัน-เวลาได้

เนื้อหาสาระ

1. ฟังก์ชันเกี่ยวกับสตริง
2. ฟังก์ชันทางคณิตศาสตร์
3. ฟังก์ชันเกี่ยวกับวันและเวลา

8.1 ฟังก์ชันเกี่ยวกับสตริง

สตริงเป็นประเภทข้อมูลชนิดหนึ่งที่ใช้เก็บข้อมูลที่มีลักษณะเป็นอักขระ หรือตัวอักษร โปรแกรมภาษา PHP ได้จัดเตรียมฟังก์ชันในกลุ่มนี้ด้วยกันหลายฟังก์ชัน แต่จะขอกล่าวถึงเฉพาะฟังก์ชันที่จำเป็นเท่านั้น ดังนี้

- **addslashes()** ใช้เพิ่มเครื่องหมาย \ ไว้ข้างหน้าให้กับตัวอักษรพิเศษ
- **chr()** และ **ord()** ใช้จัดการรหัสแอสกีของตัวอักษร
- **explode()** และ **implode()** ใช้แบ่งสตริงออกเป็นสตริงย่อยๆ และรวมมสตริง
- **htmlspecialchars()** ใช้แปลงตัวอักษรพิเศษ
- **nl2br()** ใช้แทรกแท็ก
 หรือแท็ก

- **strcmp()** ใช้เปรียบเทียบสตริง
- **strlen()** ใช้หาความยาวสตริง
- **strrev()** ใช้กลับลำดับของตัวอักษร
- **substr()** ใช้ดึงบางส่วนของสตริงออกจากสตริงหลัก
- **trim(), ltrim** และ **rtrim()** ใช้ตัดช่องว่าง แท็บ ตัวอักษรขึ้นบรรทัดใหม่

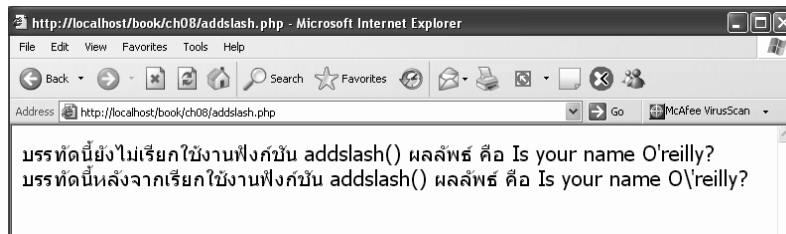
ฟังก์ชัน addslashes()

ฟังก์ชันนี้เป็นการเพิ่มเครื่องหมาย \ (Backslash) \ ให้กับตัวอักษร 3 ตัวคือ " (Double Quote), ' (Single Quote) และ \ (Backslash) ตัวอย่างโปรแกรมการใช้งานสำหรับฟังก์ชัน addslashes() เป็นดังนี้

addslash.php

```
<?
$str = "Is your name O'reilly?";
echo "บรรทัดนี้ยังไม่เรียกใช้งานฟังก์ชัน addslashes() ผลลัพธ์ คือ $str <br>";
echo "บรรทัดนี้หลังจากเรียกใช้งานฟังก์ชัน addslashes() ผลลัพธ์ คือ
".addslashes($str);
?>
```

ผลลัพธ์ของโปรแกรมนี้เป็นดังนี้



รูปที่ 8.1 ผลลัพธ์จากการแสดงผลของโปรแกรม addslash.php



ในการเก็บตัวอักษรพิเศษ คือ `'` และ `\` ลงฐานข้อมูล บางฐานข้อมูลไม่อนุญาตให้ใส่ตัวอักษรเหล่านี้ได้โดยตรง ต้องนำหน้าด้วยเครื่องหมาย `\` ก่อนจึงจะสามารถเก็บตัวอักษรดังกล่าวลงฐานข้อมูลได้

ฟังก์ชัน `chr()` และ `ord()`

ฟังก์ชันทั้งสองนี้ใช้สำหรับจัดการรหัสแอสกีของตัวอักษร โดยที่ฟังก์ชัน `chr()` จะทำการแปลงจากรหัสแอสกีให้เป็นตัวอักษร ส่วนฟังก์ชัน `ord()` จะทำการแปลงกลับจากตัวอักษรให้เป็นรหัสแอสกี ตัวอย่างโปรแกรมการใช้งานฟังก์ชัน `chr()` และ `ord()` เป็นดังนี้

`chr_ord.php`

```
<?
    for($i=65; $i<=90; $i++)
    {
        $str = chr($i);
        echo "$str ";
    }
    echo "<p>";
    for($j=a; $j<z; $j++)
    {
        $str = ord($j);
        echo "$str ";
    }
?>
```

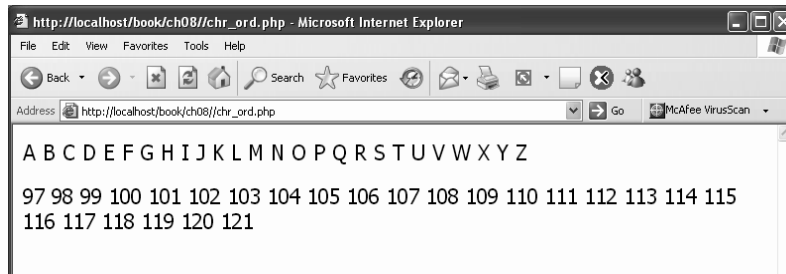
แปลงรหัสแอสกีเป็นตัวอักษร

แปลงรหัสแอสกีเป็นตัวอักษร

บทที่

8

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.2 ผลลัพธ์จากการแสดงผลของโปรแกรม chr_ord.php

ฟังก์ชัน `explode()` และ `implode()`

ฟังก์ชันทั้งสองนี้ทำงานตรงกันข้ามกัน คือฟังก์ชัน `explode()` ใช้แบ่งสตริงออกเป็นสตริงย่อยๆ ด้วยสัญลักษณ์ที่ใช้เป็นตัวแบ่ง ซึ่งสตริงย่อยที่ถูกแบ่งจะถูกจัดเก็บไว้ในตัวแปรอาร์เรย์ได้แก่ `อาร์เรย์ช่องที่ศูนย์` โดยมีรูปแบบการใช้งานของฟังก์ชัน ดังนี้

รูปแบบ

```
explode(สัญลักษณ์ที่ใช้แบ่ง, สตริงของข้อความ, [จำนวนสตริงย่อยที่ต้องการแบ่งออกมา])
```

ตัวอย่างการใช้งานของฟังก์ชัน `implode()` เป็นดังนี้

Explode.php

```
<?
    $str = "one | two | three | four";
    $a = explode('|', $str, 2);
    $b = explode('|', $str);
    for($i=0; $i<count($a); $i++)
    {
        echo "$a[$i] ";
    }
    echo "<p>";
    for($j=0; $j<count($b); $j++)
    {
        echo "$b[$j] ";
    }
?>
```


ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.3 แสดงผลลัพธ์ของโปรแกรม `explode.php`

มีการเรียกใช้ฟังก์ชัน `explode` 2 ครั้ง ครั้งแรกเราระบุให้แยกแค่ 2 ส่วนเก็บในอาร์เรย์ `$a` อีกครั้งเราไม่กำหนดคี่ส่วน คือ แยกทั้งหมดที่มีเก็บในอาร์เรย์ `$b`

ส่วนฟังก์ชัน `implode()` ใช้ข้อมูลอาร์เรย์มาประกอบรวมกันเป็นสตริงเพียงชุดเดียว โดยมีรูปแบบการใช้งานของฟังก์ชัน ดังนี้

รูปแบบ

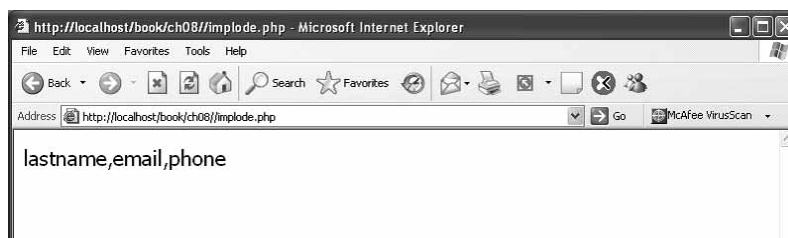
`implode(สัญลักษณ์ที่ใช้แบ่ง, อาร์เรย์)`

ตัวอย่างการใช้งานของฟังก์ชัน `implode()` เป็นดังนี้

`implode.php`

```
<?
$array = array('lastname', 'email', 'phone');
$comma_separated = implode(",", $array);
echo $comma_separated;
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นข้อมูลในอาร์เรย์มารวมกันดังนี้



รูปที่ 8.4 แสดงผลลัพธ์ของโปรแกรม `implode.php`

ฟังก์ชัน `htmlspecialchars()`

ฟังก์ชัน `htmlspecialchars()` ใช้สำหรับการแปลงตัวอักษรพิเศษ คือ < (น้อยกว่า), > (มากกว่า), & (ampersand), ' (single quote) และ " (double quote) ให้เป็นรหัสเพื่อให้สามารถแสดงตัวอักษรพิเศษเหล่านี้บนบราวเซอร์ได้อย่างถูกต้อง โดยรูปแบบของการแปลงตัวอักษรพิเศษเป็นดังนี้

<	แปลงได้เป็น	<
>	แปลงได้เป็น	>
"	แปลงได้เป็น	"
'	แปลงได้เป็น	'
&	แปลงได้เป็น	&

ยกตัวอย่างเช่น เครื่องหมาย <, > เป็นเครื่องหมายที่เข้ากับแท็ก HTML ดังนั้น เซอร์จะตีความข้อความในเครื่องหมายดังกล่าวเป็นแท็กไป วิธีแก้คือต้องแปลงเป็นเครื่องหมายดังกล่าวอยู่ในรูปรหัส > และ < แทน

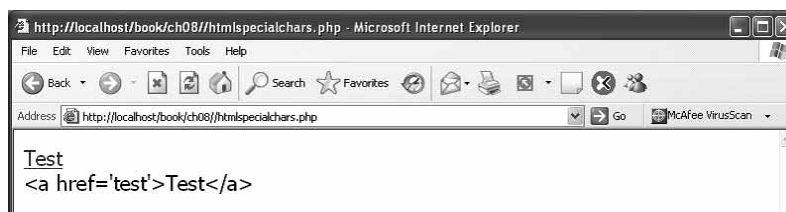
พิจารณาโปรแกรมในการใช้ฟังก์ชันนี้ ดังนี้

`htmlspecialchars.php`

```
<?
    $old = "<a href='test'>Test</a>";
    $new = htmlspecialchars("<a href='test'>Test</a>");
    echo "$old <br>";
    echo $new;
?>
```

กำหนดให้แสดงตัวอักษรพิเศษ

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.5 แสดงผลลัพธ์ของโปรแกรม `htmlspecialchars.php`

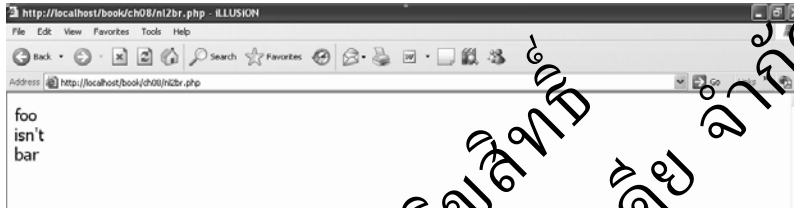
ฟังก์ชัน nl2br()

ฟังก์ชันนี้ใช้สำหรับการแทรกแท็ก
 หรือแท็ก
 ในกรณี PHP เวอร์ชัน 4.0.5 ขึ้นไป โดยจะทำการแทรกในตำแหน่งที่ข้อความขึ้นบรรทัดใหม่ โดยเราใช้แท็กขึ้นบรรทัด \n และจะแทรกแท็ก
 ไว้ก่อนหน้า \n พิจารณาโปรแกรมในการใช้งานฟังก์ชัน nl2br() ดังนี้

nl2br.php

```
<?
    echo nl2br("foo \n isn't\n bar");
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.6 แสดงผลลัพธ์ของโปรแกรม nl2br.php

ฟังก์ชัน strcmp()

ฟังก์ชันนี้ใช้สำหรับการเปรียบเทียบว่า string1 มีค่ามากกว่า string2 หรือไม่เท่ากัน โดยรูปแบบการใช้งานของฟังก์ชันนี้เป็นดังนี้

รูปแบบ

```
strcmp(string1, string2)
```

ฟังก์ชันนี้จะคืนค่าน้อยกว่า 0 ถ้าค่า string1 น้อยกว่า string2

ฟังก์ชันนี้จะคืนค่ามากกว่า 0 ถ้าค่า string1 มากกว่า string2

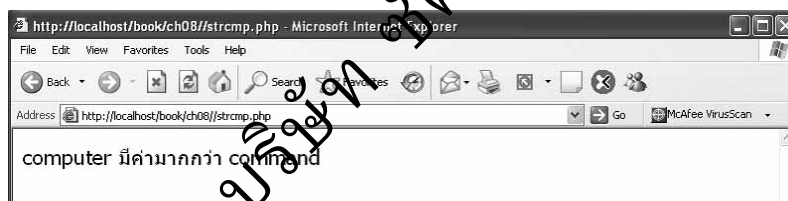
ฟังก์ชันนี้จะคืนค่าเท่ากับ 0 ถ้าค่า string1 เท่ากับ string2

พิจารณาโปรแกรมต่อไปนี้

strcmp.php

```
<?
    $str1 = "computer";
    $str2 = "command";
    $result = strcmp($str1,$str2);
    if ($result < 0)
    {
        echo "computer มีค่าน้อยกว่า command";
    }
    elseif ($result > 0)
    {
        echo "computer มีค่ามากกว่า command";
    }
    else
    {
        echo "computer มีค่าเท่ากับ command";
    }
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.7 แสดงผลลัพธ์ของโปรแกรม strcmp.php

โดยการเปรียบเทียบจะจับชนตัวอักษรที่ละหลัก ถ้าหลักซ้ายสุดเท่ากัน ก็จะดูหลักต่อไป ซึ่งจากตัวอย่างจะเป็นการเปรียบเทียบหลักที่ 4 คือ p กับ m โดย p มีค่ามากกว่า และจะถือว่าสตริงแรกมีค่ามากกว่าสตริงที่ 2 โดยไม่ดูหลักต่อไปอีกเลย

ในกรณีที่มีการจับชนที่ละหลักแล้วเท่ากันไปเรื่อยๆ จนอีกฝั่งไม่มีตัวอักษรเหลือแล้ว ให้ถือว่า สตริงฝั่งที่ยาวกว่ามีค่ามากกว่า แต่ถ้าเจอหลักที่ไม่เท่ากันก่อนและฝั่งที่หลักมีค่ามากกว่า มีความยาวสตริงสั้นกว่าอีกฝั่ง ก็จะถือว่าสตริงฝั่งสั้นมีค่ามากกว่าไป ซึ่งจากตัวอย่าง ถ้าเราตัด "uter" ออกจากสตริงตัวที่ 1 ก็ยังได้ผลลัพธ์เหมือนเดิม

ฟังก์ชัน strlen()

ฟังก์ชันนี้ใช้สำหรับหาค่าความยาวของสตริง รูปแบบการใช้งานของฟังก์ชันนี้เป็นดังนี้

รูปแบบ

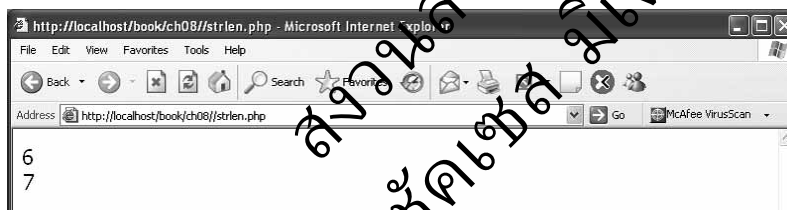
strlen(สตริงที่ต้องการหาค่าความยาว)

พิจารณาการใช้งานของฟังก์ชัน strlen() จากโปรแกรมต่อไปนี้

strlen.php

```
<?
    $str = 'abcdef';
    echo strlen($str)."<br>";
    $str = ' ab cd ';
    echo strlen($str);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.8 แสดงผลลัพธ์ของโปรแกรม strlen.php

ฟังก์ชัน strrev()

ฟังก์ชันนี้ใช้สำหรับกลับลำดับของตัวอักษรของสตริงจากด้านหลังมาด้านหน้า รูปแบบการใช้งานของฟังก์ชันนี้เป็น ดังนี้

รูปแบบ

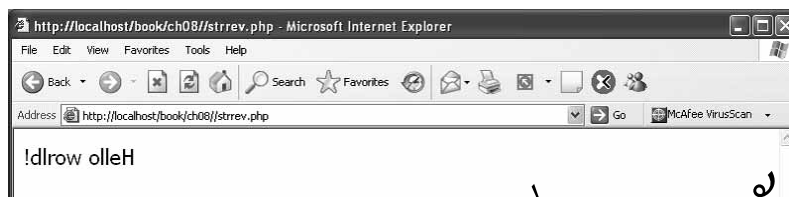
strrev(สตริงที่ต้องการกลับลำดับ)

พิจารณารูปร่างการใช้งานของฟังก์ชัน `strrev()` จากโปรแกรมต่อไปนี้

`strrev.php`

```
<?
    echo strrev("Hello world!");
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.9 ผลลัพธ์จากการแสดงผลของโปรแกรม `strrev.php`

ฟังก์ชัน **`substr()`**

ฟังก์ชันนี้ใช้สำหรับดึงบางส่วนของสตริงออกมาจากสตริงหนึ่ง โดยใช้วิธีการกำหนดว่าจะเริ่มดึงสตริงออกมาตั้งแต่ส่วนไหนและดึงออกมากี่ตัว รูปแบบการใช้งานของฟังก์ชันนี้เป็นดังนี้

รูปแบบ

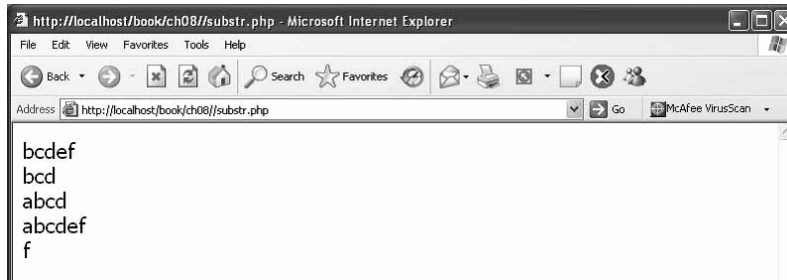
```
substr(สตริงหลัก, จุดเริ่มต้นที่ต้องการดึง, [ความยาวของสตริงที่ต้องการดึงออกมา])
```

พิจารณารูปร่างการใช้งานของฟังก์ชัน `substr()` ตามโปรแกรมต่อไปนี้

`substr.php`

```
<?
    echo substr('abcdef', 1)."<br>";
    echo substr('abcdef', 1, 3)."<br>";
    echo substr('abcdef', 0, 4)."<br>";
    echo substr('abcdef', 0, 8)."<br>";
    echo substr('abcdef', -1, 1);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.10 แสดงผลลัพธ์ของโปรแกรม `substr.php`

สำหรับค่าในการดึงบางส่วนของสตริงนั้น ถ้าเป็นค่าบวกก็เริ่มนับจากฝั่งซ้ายไปตามจำนวนที่ระบุ แล้วดึงส่วนของสตริงไปฝั่งซ้าย ถ้าเป็นค่าลบก็เริ่มนับจากฝั่งขวาตามจำนวนที่ระบุ แล้วดึงส่วนของสตริงไปทางฝั่งซ้ายเหมือนเดิม

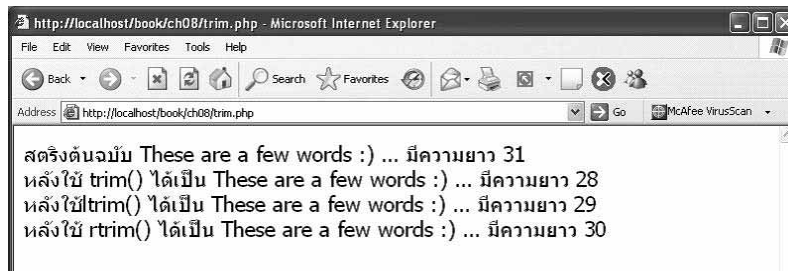
ฟังก์ชัน `trim()`, `ltrim()` และ `rtrim()`

ทั้ง 3 ฟังก์ชันนี้ใช้ตัด whitespace (ช่องว่าง ว่างๆ ตัวอักษรที่เพิ่มล่าสุดใหม่) ออกจากสตริงหลัก โดยขึ้นกับตำแหน่งที่ต้องการตัด คือ ฟังก์ชัน `trim()` ใช้ตัด whitespace ทั้งด้านหน้าและหลังออกจากสตริง, ฟังก์ชัน `ltrim()` ใช้ตัด whitespace เฉพาะที่อยู่ด้านหน้าของสตริง และฟังก์ชัน `rtrim()` ใช้ตัด whitespace เฉพาะที่อยู่ด้านหลังของสตริง พิจารณาการใช้งานของทั้ง 3 ฟังก์ชันจากโปรแกรมต่อไปนี้

`trim.php`

```
<?
    $text = "\t\tThere are a few words :) ... ";
    $text1 = trim($text);
    $text2 = ltrim($text);
    $text3 = rtrim($text);
    echo "สตริงต้นฉบับ $text มีความยาว ".strlen($text)."<br>";
    echo "หลังใช้ trim() ได้เป็น $text1 มีความยาว ".strlen($text1)."<br>";
    echo "หลังใช้ ltrim() ได้เป็น $text2 มีความยาว ".strlen($text2)."<br>";
    echo "หลังใช้ rtrim() ได้เป็น $text3 มีความยาว ".strlen($text3)."<br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.11 แสดงผลลัพธ์ของโปรแกรม trim.php

8.2 ฟังก์ชันทางคณิตศาสตร์

ฟังก์ชันในกลุ่มนี้ใช้สำหรับการจัดการข้อมูลประเภทตัวเลข มีฟังก์ชันในกลุ่มนี้ดังต่อไปนี้

- **abs()** ใช้หาค่าสัมบูรณ์ของตัวเลข
- **max()** **min()** ใช้หาค่ากลุ่มของตัวเลขที่กำหนด ค่าใดมีค่ามากที่สุด ค่าใดมีค่าน้อยสุด
- **pow()** ใช้หาค่าเลขยกกำลัง
- **round()** ใช้ปัดเศษทศนิยม
- **sqrt()** ใช้หาค่าที่สองของตัวเลขใดๆ

ฟังก์ชัน abs()

ฟังก์ชันนี้ใช้หาค่าสัมบูรณ์ของตัวเลข (Absolute Number) คือทำการเปลี่ยนตัวเลขให้เป็นค่าบวก ไม่ว่าตัวเลขที่นำมาเข้าฟังก์ชันนี้จะติดลบหรือไม่ก็ตาม และนอกจากนั้นฟังก์ชันนี้ยังสามารถใช้กับตัวเลขจำนวนเต็ม หรือเลขทศนิยมก็ได้ พิจารณาการใช้งานของฟังก์ชันจากโปรแกรมต่อไปนี้

abs.php

```
<?
$num1 = -4.2;
$num2 = 5;
$num3 = -5;
$abs1 = abs($num1);
$abs2 = abs($num2);
$abs3 = abs($num3);
echo "ค่าสัมบูรณ์ของ $num1 คือ $abs1 <br>";
```



```

echo "ค่าสัมบูรณ์ของ $num2 คือ $abs2 <br>";
echo "ค่าสัมบูรณ์ของ $num3 คือ $abs3 <br>";

?>

```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.12 แสดงผลลัพธ์ของโปรแกรม abs.php

ฟังก์ชัน `max()` และ `min()`

ฟังก์ชันทั้งสองนี้ใช้ในการหากลุ่มของตัวเลขที่ก่อกำเนิดขึ้นมาตั้งแต่หนึ่งค่ามากที่สุด และตัวเลขใดมีค่าน้อยที่สุดตามลำดับ โดยสามารถนำไปประยุกต์ใช้เพื่อหาค่าราคาสินค้าที่มีราคาแพงที่สุดคือเท่าไร และราคาสินค้าที่ถูกที่สุดคือเท่าไร รูปแบบการใช้งานของทั้งสองฟังก์ชันนี้เป็นดังนี้

รูปแบบ

```
max(ตัวเลขที่1, ตัวเลขที่2,...)
```

```
min(ตัวเลขที่1, ตัวเลขที่2,...)
```

พิจารณารูปแบบการใช้งานทั้งสองฟังก์ชันจากโปรแกรมต่อไปนี้

`max_min.php`

```

<?
    $price1 = 5000;
    $price2 = 4500;
    $price3 = 6500;
    echo max($price1, $price2, $price3)."<br>";
    echo min($price1, $price2, $price3);

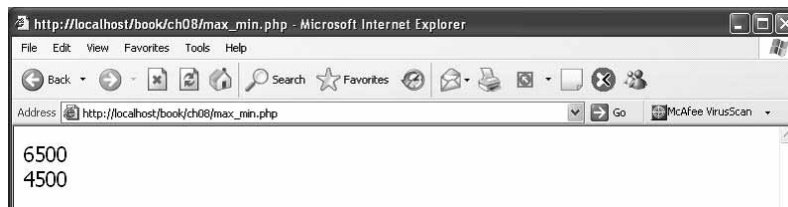
?>

```

บทที่

8

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.13 แสดงผลลัพธ์ของโปรแกรม `max_min.php`

ฟังก์ชัน `pow()`

ฟังก์ชันนี้ใช้หาค่าของเลขยกกำลัง โดยรูปแบบการใช้งานของฟังก์ชันนี้เป็นดังนี้

รูปแบบ

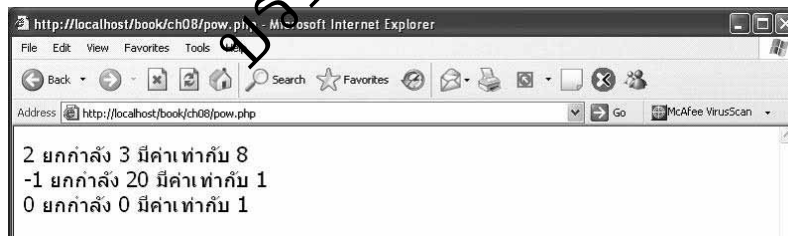
`Pow(ตัวเลขฐาน, ตัวเลขชี้กำลัง)`

พิจารณาการใช้งานจากโปรแกรมต่อไปนี้

`pow.php`

```
<?
    echo "2 ยกกำลัง 3 มีค่าเท่ากับ ".pow(2, 3)."<br>";
    echo "-1 ยกกำลัง 20 มีค่าเท่ากับ ".pow(-1, 20)."<br>";
    echo "0 ยกกำลัง 0 มีค่าเท่ากับ ".pow(0, 0)."<br>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.14 แสดงผลลัพธ์ของโปรแกรม `pow.php`

ฟังก์ชัน rand()

ฟังก์ชันนี้ใช้หาค่าของตัวเลขสุ่ม โดยการใช้งานของฟังก์ชันนี้จะสุ่มตัวเลขขึ้นมาโดยอัตโนมัติทำให้เราจะได้ค่าตัวเลขที่ไม่เหมือนกันแล้วแต่ว่ามันจะสุ่มอะไรให้ รูปแบบการใช้งานเป็นดังนี้

รูปแบบ

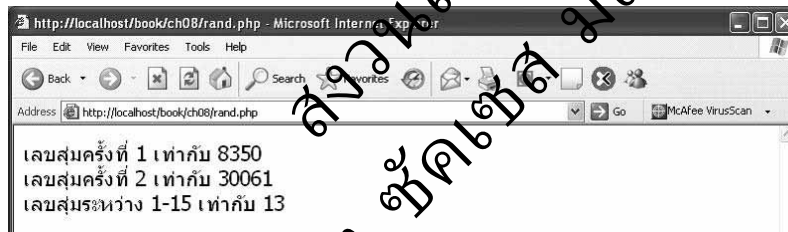
```
rand([ค่าต่ำสุดที่ต้องการให้สุ่ม , ค่าสูงสุดที่ต้องการให้สุ่ม])
```

พิจารณาการใช้งานจากโปรแกรมต่อไปนี้

rand.php

```
<?
echo "เลขสุ่มครั้งที่ 1 เท่ากับ ".rand() . "<br>";
echo "เลขสุ่มครั้งที่ 2 เท่ากับ ".rand() . "<br>";
echo "เลขสุ่มระหว่าง 1-15 เท่ากับ ".rand(1, 15);
?>
```

ผลลัพธ์จากโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.15 แสดงผลลัพธ์ของโปรแกรม rand.php

ฟังก์ชัน round()

ฟังก์ชันนี้ใช้ปัดตัวเลขทศนิยม ถ้าตำแหน่งที่ต้องการปัดมีค่ามากกว่าหรือเท่ากับ 5 ตัวเลขด้านซ้ายที่ติดกันจะปัดขึ้น 1 แต่ถ้าตำแหน่งที่ต้องการปัดมีค่าน้อยกว่า 5 ตัวเลขด้านซ้ายที่ติดกันจะเท่าเดิมคือไม่มีการปัด

รูปแบบการใช้งานของฟังก์ชันนี้เป็นดังนี้

รูปแบบ

```
round(ตัวเลขทศนิยม, [จำนวนตัวเลขหลังจุดทศนิยมที่ต้องการ])
```

พิจารณาตัวอย่างการใช้งานของฟังก์ชันนี้เป็นดังนี้

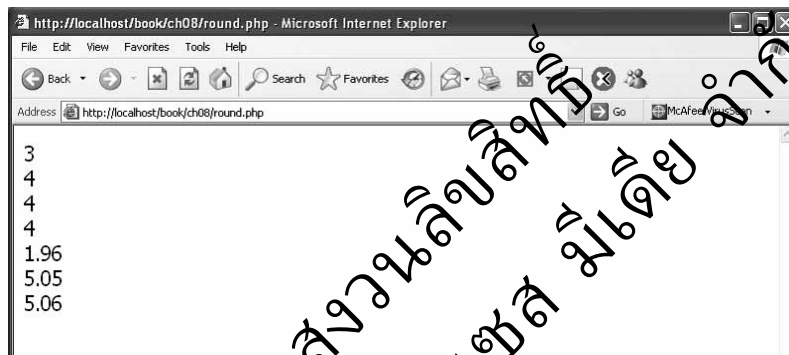
บทที่

8

round.php

```
<?
    echo round(3.4)."<br>";
    echo round(3.5)."<br>";
    echo round(3.6)."<br>";
    echo round(3.6, 0)."<br>";
    echo round(1.95583, 2)."<br>";
    echo round(5.045, 2)."<br>";
    echo round(5.055, 2);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.16 แสดงผลลัพธ์ของโปรแกรม round.php

ฟังก์ชัน sqrt()

ฟังก์ชันนี้ใช้หาค่ารากที่ 2 ของตัวเลขใดๆ (Square Root) พิจารณาการใช้งานของฟังก์ชันนี้จากโปรแกรมดังนี้

sqrt.php

```
<?
    echo "ค่ารากที่ 2 ของ 9 มีค่าเท่ากับ ".sqrt(9)."<br>";
    echo "ค่ารากที่ 2 ของ 10 มีค่าเท่ากับ ".sqrt(10);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.17 แสดงผลลัพธ์ของโปรแกรม `sqrt.php`

8.3 ฟังก์ชันเกี่ยวกับวันและเวลา

ฟังก์ชันในกลุ่มนี้ใช้จัดการข้อมูลของวัน และเวลาโดยเฉพาะ เช่น ใช้ตั้งวันเดือนปีปัจจุบันในขณะที่ทำการประมวลผล ใช้ตรวจสอบวันเดือนปีว่ามีอยู่จริงหรือไม่ เป็นต้น ฟังก์ชันที่สำคัญ คือ

- `checkdate()` ใช้ตรวจสอบวันเดือนปี มีอยู่จริงหรือไม่
- `date()` ใช้ตั้งวันเดือนปี และเวลาที่ปัจจุบันของเครื่อง
- `getdate()` ให้ผลลัพธ์เป็นวันเวลาปัจจุบัน
- `time()` จะคืนค่าของวันที่ปัจจุบันของเครื่อง

ฟังก์ชัน `checkdate()`

ฟังก์ชันนี้ใช้สำหรับตรวจสอบว่าวันเดือนปีมีอยู่จริงหรือไม่ รูปแบบการใช้งานเป็นดังนี้

รูปแบบ

```
checkdate(เดือน, วันปี, ปี)
```

โดยข้อกำหนดในการตรวจสอบ คือ

ปี จะต้องอยู่ระหว่าง 1 ถึง 32767

เดือน จะต้องอยู่ระหว่าง 1 ถึง 12

วันที่ ตรวจสอบว่าวันทีนั้นในเดือน และปีนั้นมีอยู่จริงหรือไม่

ผลลัพธ์ของฟังก์ชันนี้จะคืนค่าจริง ถ้าวันเดือนปีนั้นมีอยู่จริง แต่จะคืนค่าเท็จถ้าวันเดือนปีนั้นไม่มีอยู่จริง พิจารณาตัวอย่างการใช้งานของฟังก์ชันนี้ ดังนี้

บทที่

8

checkdate.php

```
<?
function testdate($day, $month, $year)
{
    if (checkdate($month, $day, $year))
    {
        echo "วันที่ $day เดือน $month ปี $year มีอยู่จริง <br>";
    }
    else
    {
        echo "วันที่ $day เดือน $month ปี $year ไม่มีอยู่จริง <br>";
    }
}

$day1 = 31;
$month1 = 12;
$year1 = 2000;
testdate($day1, $month1, $year1);

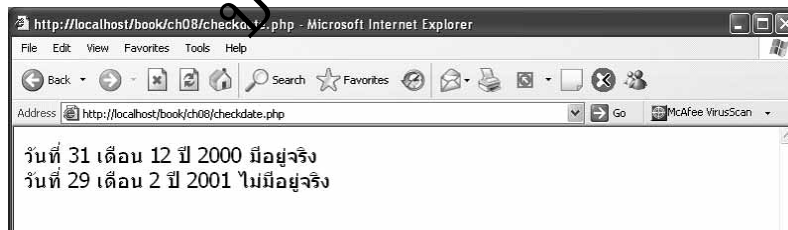
$day2 = 29;
$month2 = 2;
$year2 = 2001;
testdate($day2, $month2, $year2);
?>
```

กำหนดฟังก์ชัน
ที่เรียกใช้
checkdate ()

เรียกใช้ฟังก์ชัน

เรียกใช้ฟังก์ชัน

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.18 แสดงผลลัพธ์ของโปรแกรม checkdate.php

ฟังก์ชัน date()

ฟังก์ชันนี้ใช้สำหรับดึงวันเดือนปี และเวลาปัจจุบันของเครื่องตามรูปแบบที่ต้องการ รูปแบบคำสั่งในการใช้งานเป็นดังนี้

รูปแบบ

date(ตัวอักษรที่ระบุรูปแบบในการแสดงผล, [เวลาที่ระบุ])

ตารางสรุปตัวอักษรที่ใช้ระบุรูปแบบในการแสดงผลลัพท์ เป็นดังนี้

รูปแบบตัวอักษร	คำอธิบาย	ตัวอย่างค่าที่แสดงผลกลับ
Day	—	—
d	แสดงวันที่เป็นตัวเลข 2 หลัก มี 0 นำหน้า	01 ถึง 31
j	แสดงวันที่เป็นตัวเลข	1 ถึง 31
S	แสดงตัวยกต่อท้ายวันที่ ไข้ร่วมกับ j	st, nd, rd หรือ th
D	แสดงวันเป็นตัวอักษร 3 หลัก	Mon ถึง Sun
l (แอลเล็ก)	แสดงวันเป็นชื่อเต็ม	Sunday ถึง Saturday
N	แสดงวันเป็นตัวเลข มาตรฐาน ISO-8601	1 (Monday) ถึง 7 (Sunday)
w	แสดงวันเป็นตัวเลข	0 (Sunday) ถึง 6 (Saturday)
z	แสดงลำดับวันใน 1 ปี	0 ถึง 365
Week	—	—
W	แสดงลำดับสัปดาห์ใน 1 ปี	1 ถึง 52
Month	—	—
m	แสดงเดือนเป็นตัวเลข 2 หลัก มี 0 นำหน้า	01 ถึง 12
n	แสดงเดือนเป็นตัวเลข	1 ถึง 12
M	แสดงเดือนเป็นตัวอักษร 3 หลัก	Jan ถึง Dec
F	แสดงเดือนเป็นชื่อเต็ม	January ถึง December
t	แสดงจำนวนวันในเดือนทั้งหมด	28 ถึง 31
Year	—	—
L	ตรวจสอบว่าเป็นปีอธิกสุรทินหรือไม่	1 เป็น, 0 ไม่เป็น
Y	แสดงปีเป็นตัวเลข 4 หลัก	1999 หรือ 2008
y	แสดงปีเป็นตัวเลข 2 หลัก	99 หรือ 08
Time	—	—
a	แสดงตัวอักษรเล็ก บอกก่อนเที่ยง (Ante meridiem) และหลังเที่ยง (Post meridiem)	am หรือ pm
A	แสดงตัวอักษรใหญ่ บอกก่อนเที่ยง (Ante meridiem) และหลังเที่ยง (Post meridiem)	AM หรือ PM
g	แสดง 12 ชั่วโมงเป็นตัวเลข	1 ถึง 12
G	แสดง 24 ชั่วโมงเป็นตัวเลข	0 ถึง 23
h	แสดง 12 ชั่วโมงเป็นตัวเลข มี 0 นำหน้า	01 ถึง 12

รูปแบบตัวอักษร	คำอธิบาย	ตัวอย่างค่าที่แสดงผลกลับ
H	แสดง 24 ชั่วโมงเป็นตัวเลข มี 0 นำหน้า	00 ถึง 23
i	แสดงนาทีเป็นตัวเลข มี 0 นำหน้า	00 ถึง 59
s	แสดงวินาทีเป็นตัวเลข มี 0 นำหน้า	00 ถึง 59
u	แสดงมิลลิวินาทีเป็นตัวเลข	เช่น 54321
Timezone	—	—
e	แสดง Timezone ที่ใช้	เช่น UTC, GMT, Atlantic/Azores
O	แสดงจำนวนชั่วโมงที่ห่างจากเวลาของ Greenwich (GMT)	เช่น +0200
Full Date/Time	—	—
c	แสดงวันเวลาสมบูรณ์ มาตรฐาน ISO-8601	เช่น 2008-01-15T23:55:49+07:00
r	แสดงวันเวลาสมบูรณ์ มาตรฐาน RFC 2822	เช่น Tue, 15 Jan 2008 23:55:49 +0700
U	แสดงวินาทีปัจจุบันในระบบนับเวลาของ Unix	เช่น 1200416149

Note

ระบบนับเวลาของ Unix จะเริ่ม 0 ตั้งแต่ January 1, 1970 00:00:00 GMT และนับต่อมาเรื่อยๆ เป็นวินาที จนถึงปัจจุบัน เช่น 1200416149 คือ วันที่ Tue, 15 Jan 2008 23:55:49 +0700 เป็นต้น

พิจารณาตัวอย่างการใช้ฟังก์ชันในหลายๆ รูปแบบ ตัวอย่าง ดังนี้

date.php

```
<?
// สมมติวันนี้ คือ January 31, 2006, 23:34:40 pm

$today = date("j, Y, g:i a");
echo "$today <br>";
$today = date("m.d.y");
echo "$today <br>";
$today = date("j, n, Y");
echo "$today <br>";
$today = date("Ymd");
echo "$today <br>";
$today = date('h-i-s, j-m-y, it is w Day z ');
echo "$today <br>";
$today = date('\i\t \i\s \t\h\e jS \d\aly.);
echo "$today <br>";
$today = date("D M j G:i:s T Y");
```



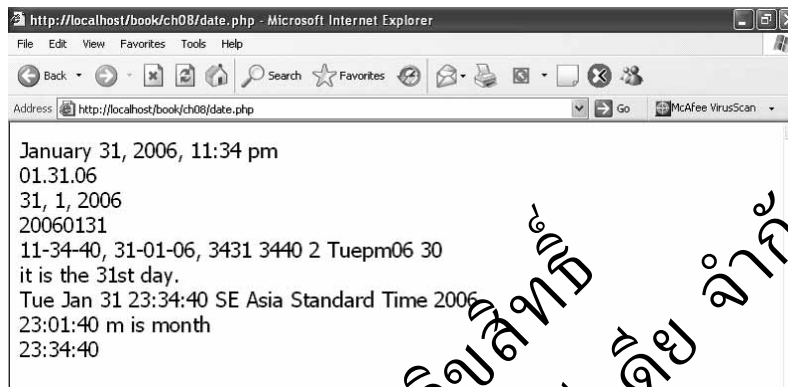
```

echo "$today <br>";
$today = date('H:m:s \m \i\s\ \m\o\n\t\h');
echo "$today <br>";
$today = date("H:i:s");
echo "$today <br>";

?>

```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.19 แสดงผลลัพธ์ของโปรแกรม date.php

ฟังก์ชัน getdate()

ฟังก์ชัน getdate() นี้จะให้ผลลัพธ์เหมือนฟังก์ชัน date() คือให้ผลลัพธ์เป็นวันและเวลาปัจจุบัน แต่ต่างกันที่ผลลัพธ์จะแสดงเป็นข้อมูลชนิดอาร์เรย์ โดยกำหนดอินดิเคอร์ของตัวแปรอาร์เรย์เพื่อแสดงผลลัพธ์ ดังนี้

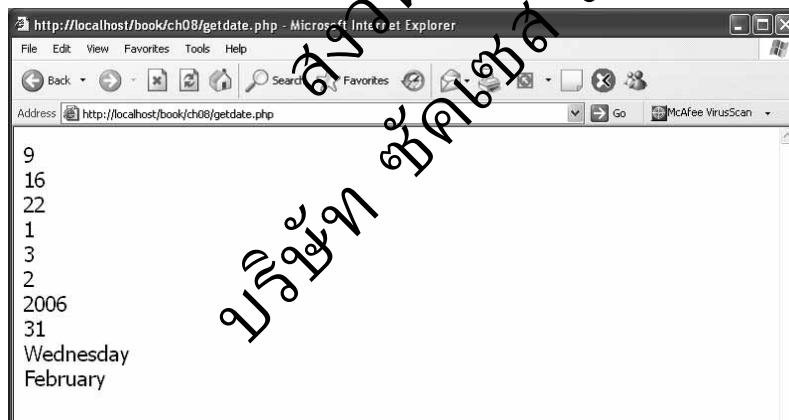
อินดิเคอร์	คำอธิบาย	ตัวอย่างผลลัพธ์
"seconds"	แสดงวินาทีเป็นตัวเลข	0 ถึง 59
"minutes"	แสดงนาทีเป็นตัวเลข	0 ถึง 59
"hours"	แสดงชั่วโมงเป็นตัวเลข	0 ถึง 23
"mday"	แสดงวันที่ใน 1 เดือนเป็นตัวเลข	1 ถึง 31
"wday"	แสดงวันในสัปดาห์เป็นตัวเลข	0 (คือวันอาทิตย์) ถึง 6 (คือวันเสาร์)
"mon"	แสดงเดือนใน 1 ปีเป็นตัวเลข	1 ถึง 12
"year"	แสดงปีเป็นหมายเลข 4 หลัก	ตัวอย่าง: 1999 หรือ 2008
"yday"	แสดงจำนวนวันในรอบ 1 ปี	0 ถึง 365
"weekday"	แสดงวันในสัปดาห์เป็นตัวหนังสือภาษาอังกฤษ	Sunday ถึง Saturday
"month"	แสดงเดือนใน 1 ปีเป็นตัวหนังสือภาษาอังกฤษ	January ถึง December

พิจารณาผลลัพธ์จากการใช้ฟังก์ชัน `getdate()` ดังนี้

Getdate.php

```
<?
// สมมติวันนี้ คือ February 1, 2006, 22:16:9
$today = getdate();
echo "$today[seconds] <br>";
echo "$today[minutes] <br>";
echo "$today[hours] <br>";
echo "$today[mday] <br>";
echo "$today[wday] <br>";
echo "$today[mon] <br>";
echo "$today[year] <br>";
echo "$today[yday] <br>";
echo "$today[weekday] <br>";
echo "$today[month] ";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.20 แสดงผลลัพธ์ของโปรแกรม `getdate.php`

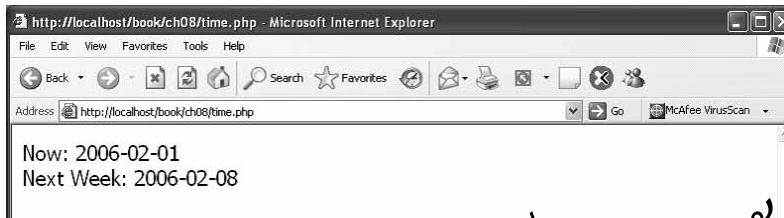
ฟังก์ชัน `time()`

ฟังก์ชัน `time()` จะคืนค่าของวัน/เวลาปัจจุบันของเครื่องกลับมาให้ โดยผลลัพธ์เป็นจำนวนวินาที นับตั้งแต่วันที่ 1 มกราคม 1970 เช่น ฟังก์ชัน `time()` คืนค่าวินาทีปัจจุบันเป็น 1200416149 คือ วันที่ 15 มกราคม 2008 เป็นต้น พิจารณาการใช้งานฟังก์ชัน `time()` ตามตัวอย่างต่อไปนี้

Time.php

```
<?
    $nextWeek = time() + (7 * 24 * 60 * 60);
    echo 'Now:      '. date('Y-m-d') ."<br>";
    echo 'Next Week: '. date('Y-m-d', $nextWeek);
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 8.21 แสดงผลลัพธ์ของโปรแกรม time.php

จากตัวอย่าง \$nextWeek จะเก็บค่าวันที่ปัจจุบันบวกกับจำนวนวันที่ใน 1 สัปดาห์ จากนั้นใช้ฟังก์ชัน date() เปลี่ยนวันที่ในตัวแปร \$nextWeek ให้อยู่ในรูปแบบ 'Y-m-d'

สรุป

ฟังก์ชันเกี่ยวกับสตริงใช้จัดการข้อมูลที่มีลักษณะเป็นอักขระ หรือตัวอักษร ฟังก์ชันที่ PHP จัดเตรียมไว้ให้ในกลุ่มนี้ เช่น addslashes(), chr(), ord(), explode(), implode(), htmlspecialchars(), nl2br(), strcmp(), strlen(), strtolower(), substr(), trim(), ltrim(), rtrim() เป็นต้น

ฟังก์ชันทางคณิตศาสตร์ ฟังก์ชันในกลุ่มนี้ใช้สำหรับจัดการข้อมูลประเภทตัวเลข ทั้งเลขจำนวนเต็ม และทศนิยม ฟังก์ชันที่ PHP จัดเตรียมไว้ให้ในกลุ่มนี้ เช่น abs(), max(), min(), pow(), rand(), round(), sqrt(), เป็นต้น

ฟังก์ชันเกี่ยวกับวันและเวลา ฟังก์ชันในกลุ่มนี้ใช้จัดการข้อมูลของวัน และเวลาโดยเฉพาะ เช่น ใช้ตั้งวันเดือนปีปัจจุบันในขณะทำการประมวลผล ใช้ตรวจสอบวันเดือนปีว่ามีอยู่จริงหรือไม่ เป็นต้น ฟังก์ชันที่ PHP จัดเตรียมไว้ให้ในกลุ่มนี้ เช่น checkdate(), date(), getdate(), time() เป็นต้น

แบบฝึกหัดบทที่ 8

1. ให้เขียนโปรแกรมภาษา PHP โดยกำหนดค่าของตัวแปรเก็บข้อความใดๆ หลังจากนั้นให้ทำการแปลงตัวอักษรแต่ละตัวในตัวแปรจากตัวอักษรตัวเล็กให้เป็นตัวอักษรตัวใหญ่ และจากตัวอักษรตัวใหญ่ให้เป็นตัวอักษรตัวเล็ก เช่น Hello แปลงได้เป็น hELLO

Hint: ใช้ฟังก์ชัน `char()`, `ord()` และ `substr()`

2. ให้เขียนโปรแกรมภาษา PHP โดยกำหนดค่าของตัวแปร วัน เดือน และปีในรูปแบบ ดังนี้

ตัวเลขวันที่/ ตัวเลขเดือน / ตัวเลขปี

เช่น 1/4/2549 หลังจากนั้นให้ใช้ฟังก์ชัน `explode()` เพื่อดึงค่าวัน เดือน และปีออกมาเก็บไว้ในตัวแปรอาร์เรย์และใส่รูปแบบเพื่อแสดงผลลัพธ์ในลักษณะดังนี้

วันที่ 1 เดือน เมษายน ค.ศ. 2006

3. ให้เขียนโปรแกรมภาษา PHP เพื่อสร้างเกมสุ่มตัวเลข โดยทำการสุ่มผลรวมของลูกเต๋า 3 ลูก จำนวน 10 ครั้ง และกำหนดเงื่อนไขของผลรวมของลูกเต๋านั้นๆ ดังนี้

3-9	ต่ำ
10-11	กลาง
12-18	สูง

หลังจากนั้นให้นับว่ามีค่า ต่ำ, กลาง และสูง อย่างละกี่ครั้งจากการสุ่มทั้งหมด 10 ครั้งนี้

4. ให้เขียนโปรแกรมเพื่อแสดงวัน เดือน ปี ปัจจุบันในรูปแบบ ดังนี้

วันนี้เป็นวัน วันในสัปดาห์เป็นภาษาไทย ที่ ตัวเลขวันที่ เดือน เดือนเป็นภาษาไทย พ.ศ. ตัวเลขพ.ศ. 4 หลัก เช่น วันเสาร์ที่ 22 เดือน เมษายน พ.ศ. 2549

5. ให้สร้างฟังก์ชันแบบที่มีการส่งผ่านค่าโดยรับค่าจำนวนสินค้าเพื่อตรวจสอบว่า การกรอกตัวเลขจำนวนสินค้าไม่ได้กรอกเป็นค่าติดลบ โดยส่งข้อความบอกว่าตัวเลขที่กรอกถูก หรือผิดกลับมายังโปรแกรมหลักด้วย

การสร้างช่องรับข้อมูล

เราคงเคยเห็นและเคยใช้งานระบบงานบนเว็บมาบ้างแล้ว ยกตัวอย่างเช่น ระบบการรับสมัครสมาชิก ซึ่งจะมีแบบฟอร์มให้กรอกรายละเอียดต่างๆ ซึ่งจะถูกนำไปประมวลผลต่อไป, ระบบตะกร้า หรือรถเข็น สำหรับใช้เลือกซื้อสินค้าที่ต้องการโดยหยิบสินค้าลงตะกร้า ระบบจะทำการคำนวณปริมาณสินค้า และราคารวมให้โดยอัตโนมัติ เป็นต้น เราจะเห็นว่าในการทำงานดังกล่าวนี้จะต้องมีส่วนของการรับข้อมูลนำเข้า (input) เพื่อนำไปประมวลผล เนื้อหาในบทนี้จึงจะกล่าวถึงการสร้างแบบช่องรับข้อมูลด้วยภาษา HTML ซึ่งมี ออบเจกต์รับข้อมูลอยู่หลายประเภท

จุดประสงค์ ทั่วไป

1. เพื่อเรียนรู้ช่องรับข้อมูลต่างๆ ใน HTML
2. เพื่อให้เข้าใจโค้ดของช่องรับข้อมูลต่างๆ และสามารถสร้างช่องรับข้อมูลแบบต่างๆ ได้
3. เพื่อเรียนรู้และสามารถเขียนโปรแกรมเพื่อส่งและรับข้อมูลได้
4. เพื่อให้ทราบถึงวิธีการรับข้อมูลประเภท Post และ Get
5. เพื่อให้ทราบความแตกต่างระหว่างการส่งข้อมูลประเภท Post และ Get
6. เพื่อให้เรียนรู้และสามารถเขียนโปรแกรมการส่ง และรับข้อมูลในลักษณะการรวมไฟล์ HTML และ PHP

จุดประสงค์ เชิงพฤติกรรม

1. สามารถสร้างช่องรับข้อมูลได้
2. สามารถเขียนโปรแกรมช่องรับข้อมูลต่างๆ ได้
3. บอกความแตกต่างระหว่างการส่งข้อมูลประเภท Post และ Get
4. สามารถเขียนโปรแกรมการส่ง และรับข้อมูลได้

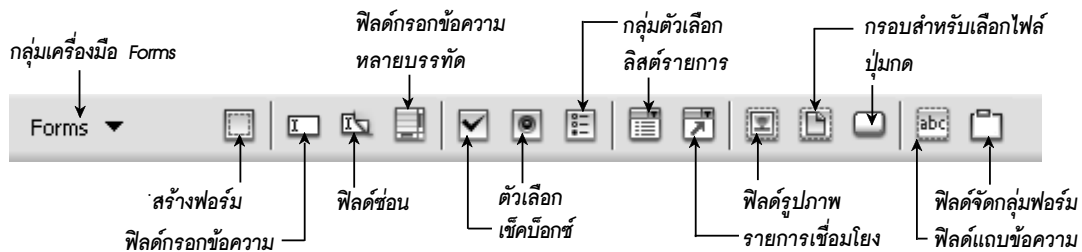
เนื้อหาสาระ:

1. ช่องรับข้อมูลต่างๆ ใน HTML
2. การส่งข้อมูล
3. การรวมไฟล์ HTML และ PHP

ปัจจุบันมีการใช้แบบฟอร์มในเว็บเพจเพื่อเก็บข้อมูล ตั้งแต่รายละเอียดเกี่ยวกับผู้ที่มาเยี่ยมชมเว็บเพจ เพื่อตั้งแต่รายละเอียดเกี่ยวกับผู้ที่มาเยี่ยมชมเว็บเพจ การสมัครสมาชิก ไปจนถึงการรวบรวมความคิดเห็นเกี่ยวกับเรื่องต่างๆ ที่กำลังอยู่ในความสนใจ เช่น แสดงความคิดเห็น เป็นต้น

แบบสอบถามขอรายละเอียดเกี่ยวกับผู้ใช้

ใน Dreamweaver ได้จัดเตรียมกลุ่มเครื่องมือ Forms ที่ใช้ในการสร้างฟอร์ม โดยเราสามารถนำช่องรับข้อมูล (Input Object) เช่น Text Areas, Checkbox และอื่นๆ ไปวางบนเอกสารฟอร์ม แต่จะต้องเขียนสคริปต์ PHP ทำงานควบคู่กับฟอร์มด้วย เพื่อใช้ในการประมวลผลข้อมูลที่ได้รับจากแบบสอบถาม



9.1 ช่องรับข้อมูลต่างๆ ใน HTML

ฟอร์ม (Form)

โดยปกติก่อนที่จะสร้างช่องรับข้อมูลต่างๆ เราจะต้องทำการสร้างฟอร์มก่อน ซึ่งฟอร์มจะทำหน้าที่เป็นที่ยรวมของช่องรับข้อมูลต่างๆ และที่สำคัญฟอร์มยังทำหน้าที่กำหนดวิธีการส่งข้อมูล การระบุไฟล์ที่จะส่งข้อมูลไปประมวลผล

รูปแบบโค้ด HTML ของฟอร์ม เป็นดังนี้

รูปแบบ

```
<form name=ชื่อฟอร์ม method=วิธีส่งข้อมูล action=ไฟล์ที่รับข้อมูลจากฟอร์ม>
...
ช่องรับข้อมูลต่างๆ
...
</form>
```

โดยที่

name=ชื่อฟอร์ม

ระบุชื่อของฟอร์ม ในส่วนนี้ถ้าจะไม่มีระบุก็ได้

method=วิธีส่งข้อมูล

กำหนดวิธีการส่งข้อมูลไปยังเซิร์ฟเวอร์ ซึ่งมีอยู่ 2 ประเภท คือ

- Get เป็นวิธีการส่งข้อมูลที่ค่าของช่องรับข้อมูลรวมไปกับ URL
- Post เป็นวิธีการส่งข้อมูลที่ต้องการให้ปลอดภัย คือจะไม่เห็นค่าของช่องรับข้อมูล

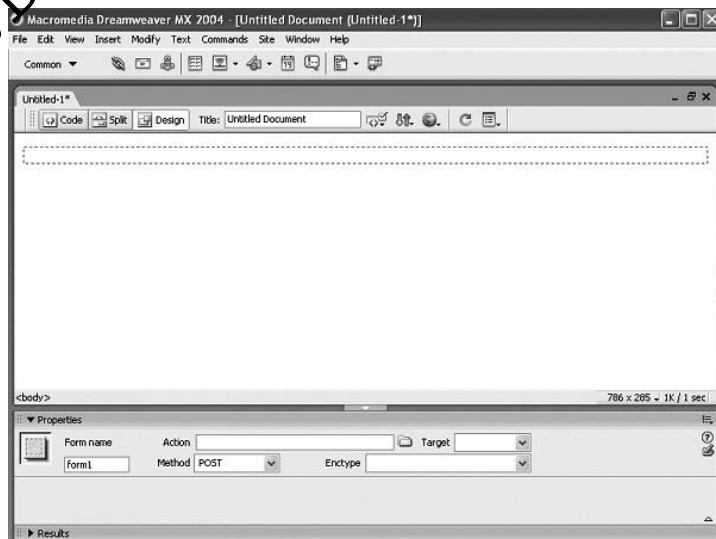
action=ไฟล์ที่รับข้อมูลจากฟอร์ม

เป็นไฟล์ที่ระบุชื่อไฟล์ที่จะส่งไปเพื่อทำการประมวลผล โดยนำค่าจากช่องรับข้อมูลไปประมวลผลต่อที่ไฟล์นี้

ถ้าใช้ Dreamweaver

ก็ทำการคลิกที่ปุ่ม  (Form)

จะปรากฏเส้นประแสดงขอบเขตของฟอร์ม ดังนี้



รูปที่ 9.1 การสร้างฟอร์มด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของฟอร์มได้ ดังนี้

- Action** ระบุชื่อไฟล์ที่ฟอร์มนี้จะส่งข้อมูลไปประมวลผล
- Method** ระบุวิธีการส่งข้อมูล (POST หรือ GET)
- Enctype** กำหนดประเภทของข้อมูล เช่น ถ้าเป็นการอัปโหลดไฟล์ก็จะกำหนด Enctype เป็น multipart/form-data

ช่องรับข้อความ (Text Field)

เป็นช่องเพื่อให้ผู้ใช้กรอกข้อมูล หรือรายละเอียด เช่น เป็นช่องรับชื่อ-นามสกุล, ที่อยู่ของสมาชิก เป็นต้น รูปแบบโค้ด HTML ของช่องรับข้อความเป็นดังนี้

รูปแบบ

```
<input type="text" name="ชื่อของ Text Field">
```

โดยที่

type="text"


name=ชื่อของ Text Field

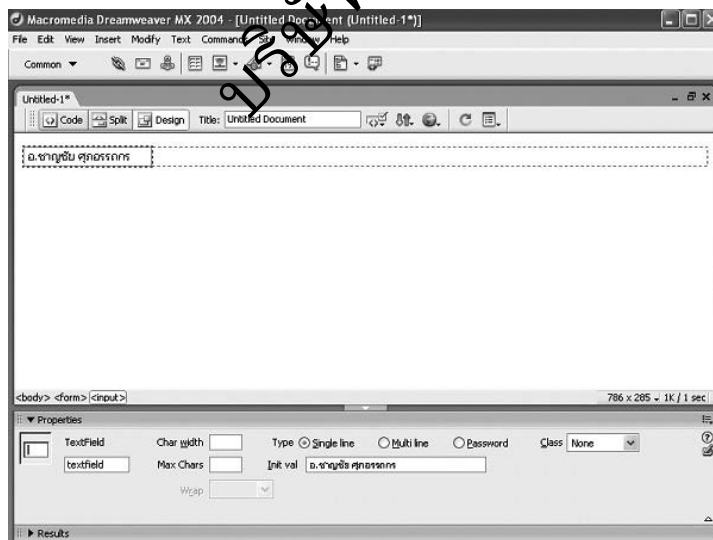
ระบุว่าเป็น ช่องรับข้อความ

ระบุชื่อของช่องรับข้อความ ให้ตั้งชื่อให้สื่อความหมายกับข้อมูล

ที่จะจัดเก็บในช่องนี้ เช่น Name คือช่องรับข้อมูล ชื่อ และ

ห้ามตั้งชื่อซ้ำกัน ถ้ามีช่องรับข้อความหลายตัว

ถ้าใช้ Dreamweaver ก็ทำการคลิกปุ่ม  (Text Field) จะปรากฏช่องรับข้อความขึ้นมา ซึ่งจะต้องสร้างช่องรับข้อความนี้ให้อยู่ภายในฟอร์ม ดังนี้



รูปที่ 9.2 การสร้างช่องรับข้อความด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับข้อความได้ ดังนี้

- Char width** ระบุขนาดความกว้างของช่องรับข้อความ
- Max chars** ระบุจำนวนตัวอักษรสูงสุดที่ช่องรับข้อความสามารถรับได้
- Init val** ระบุค่าเริ่มต้นในช่องรับข้อความที่ต้องการให้แสดง
- Type**
- Single line ลักษณะของช่องรับข้อความจะเป็นแบบรับข้อความได้เพียงบรรทัดเดียว
 - Multi line ลักษณะของช่องรับข้อความจะเป็นแบบรับข้อความได้หลายบรรทัดเดียว
 - Password ลักษณะของช่องรับข้อความจะใช้สำหรับการรับค่ารหัสผ่าน โดยสิ่งที่กรอกลงในช่องรับข้อความนี้จะเบลอจนอ่านข้อความดังกล่าวไม่ได้


ช่องรับข้อความแบบหลายบรรทัด (Text Area)

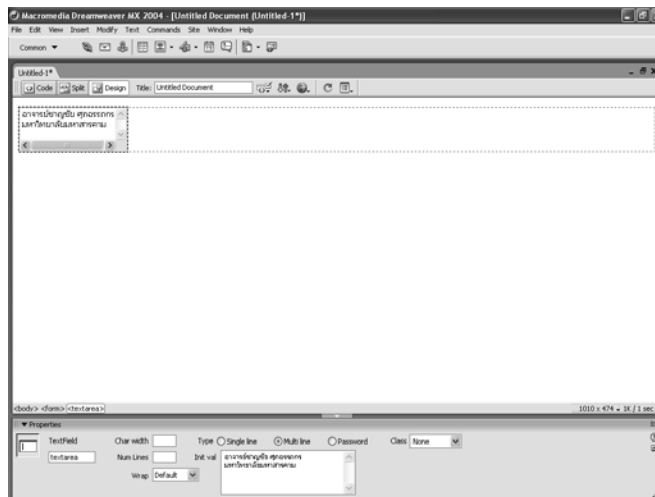
ลักษณะเหมือนกับช่องรับข้อความ (Text Field) แต่สามารถรับข้อความได้หลายบรรทัด โดยถ้าข้อความที่กรอกลงไปมีจำนวนหลายบรรทัดมากกว่าความสูงของช่องรับข้อความ หรือข้อความมีความยาวมากกว่าความกว้างของช่องรับข้อความก็จะปรากฏ Scroll Bar ขึ้นมาเพื่อใช้ในการเลื่อนดูข้อมูล

รูปแบบโค้ด HTML ของ Text Area เป็นดังนี้

รูปแบบ

```
<textarea name= ชื่อของ text area ...
    ค่าเริ่มต้น
    ...
</textarea>
```

ถ้าใช้ Dreamweaver ก็ทำการคลิกที่ปุ่ม  (Textarea) จะปรากฏช่องรับข้อความแบบหลายบรรทัดขึ้นมา ซึ่งจะต้องสร้างช่องรับข้อความแบบหลายบรรทัดนี้ให้อยู่ภายในฟอร์ม ดังนี้



รูปที่ 9.3 การสร้างช่องรับข้อความแบบหลายบรรทัดด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver ที่สามารถปรับแต่งลักษณะของช่องรับข้อความแบบหลายบรรทัดได้ ดังนี้

Char width ระบุขนาดความกว้างของช่องรับข้อความแบบหลายบรรทัด

Num Lines ระบุจำนวนบรรทัดของข้อความโดยไม่ปรากฏ scroll bar

กล่องตัวเลือก (Check box)

กล่องตัวเลือกนี้มีลักษณะเป็นช่องสี่เหลี่ยมเล็ก ๆ ใช้สำหรับการคลิกเลือกรายการที่ต้องการ โดยเราสามารถสร้างกล่องตัวเลือกได้หลายกล่อง และในการเลือกรายการต่างๆ สามารถทำการเลือกได้หลายรายการพร้อมกัน ถ้ามีการเลือกที่รายการใดภายในกล่องตัวเลือกนั้นก็จะมีเครื่องหมายถูก

รูปแบบโค้ด HTML ของกล่องตัวเลือกเป็นดังนี้

รูปแบบ

```
<input type="checkbox" name=ชื่อของกล่องตัวเลือก value=ค่าของกล่องตัวเลือก>
```

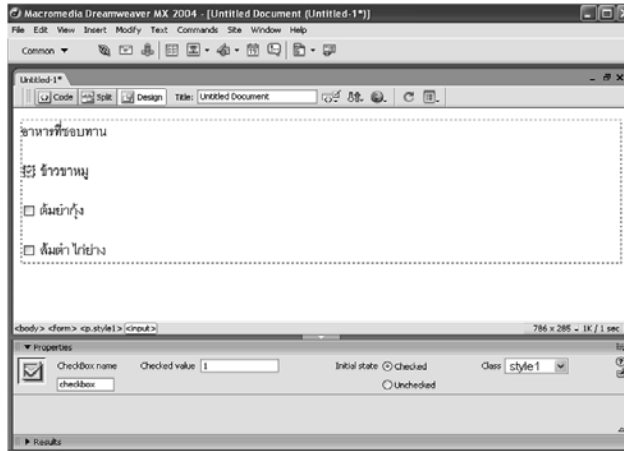
โดยที่

type="checkbox" การระบุว่า Input Object นี้คือ กล่องตัวเลือก

name=ชื่อของกล่องตัวเลือก ระบุชื่อของกล่องตัวเลือก

value=ค่าของกล่องตัวเลือก ระบุค่าของกล่องตัวเลือกถ้ามีการเลือกกล่องตัวเลือกนี้

ถ้าใช้ Dreamweaver ก็ทำการคลิกที่ปุ่ม (checkbox) จะปรากฏกล่องตัวเลือกขึ้นมา ซึ่งจะต้องสร้างกล่องตัวเลือกนี้ให้อยู่ภายในฟอร์ม ดังนี้



รูปที่ 9.4 การสร้างกล่องตัวเลือกด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของกล่องตัวเลือกได้ ดังนี้

Checked value

ระบุค่าของกล่องที่ถูกเลือกแต่ละตัว

Initial state

- Checked ระบุให้กล่องตัวเลือกนี้ มีวงเลือกก็จะปรากฏเครื่องหมายถูกขึ้น
- Unchecked ระบุให้กล่องตัวเลือกนี้ ไม่มีการเลือก กล่องตัวเลือกก็จะว่าง

จากรูปที่ 9.4 ได้ทำการสร้างกล่องตัวเลือกของรายการอาหารที่ขอรับประทาน โดยมีอาหารอยู่ 3 รายการ เราสามารถเลือกรายการอาหารรายการเดียว หรือ 2 รายการ หรือ ทั้ง 3 รายการก็ได้ โค้ด HTML ของกล่องตัวเลือกในรูปแบบนี้จะเป็นดังนี้

```

อาหารที่ขอทาน <p>
<input type="checkbox" name="ka_moo" value="1" checked> ข้าวขาหมู<p>
<input type="checkbox" name="tumyum_kung" value="2"> ต้มยำกุ้ง<p>
<input type="checkbox" name="samtam" value="3"> ส้มตำ ไก่ย่าง
  
```

จากโค้ดข้างต้นนี้ได้ตั้งชื่อกล่องตัวเลือกเป็นชื่ออาหารเพื่อให้การเขียนโปรแกรมจะได้สะดวกต่อการเรียกใช้ ที่ค่าของกล่องตัวเลือกก็ได้กำหนดค่าเป็น 1, 2 และ 3 ตามลำดับ โดยถ้ามีการเลือก "ข้าวขาหมู" ค่าของ "ka_moo" ก็จะมีค่าเท่ากับ 1, ถ้ามีการเลือก "ต้มยำกุ้ง" ค่าของ "tumyum_kung" ก็จะมีค่าเท่ากับ 2 และ ถ้ามีการเลือก "ส้มตำ ไก่ย่าง" ค่าของ "samtam" ก็จะมีค่าเท่ากับ 3

ปุ่มตัวเลือก (Radio Button)

ปุ่มตัวเลือกจะมีลักษณะเป็นวงกลมเล็กๆ ใช้สำหรับการคลิกเลือกรายการที่ต้องการ โดยเราสามารถสร้างปุ่มตัวเลือกได้หลายปุ่ม แต่ในการเลือกรายการต่างๆ สามารถทำการเลือกได้เพียงรายการใดรายการหนึ่งเท่านั้น ถ้ามีการเลือกรายการที่ปุ่มตัวเลือกใดภายในวงกลมนั้นก็จะมีเครื่องหมายจุดสีปรากฏขึ้น

รูปแบบโค้ด HTML ของปุ่มตัวเลือกเป็นดังนี้

รูปแบบ

```
<input type="radio" name=ชื่อของปุ่มตัวเลือก value=ค่าของปุ่มตัวเลือก>
```

โดยที่

type="radio"

ระบุว่าเป็นปุ่มตัวเลือก

name=ชื่อของปุ่มตัวเลือก

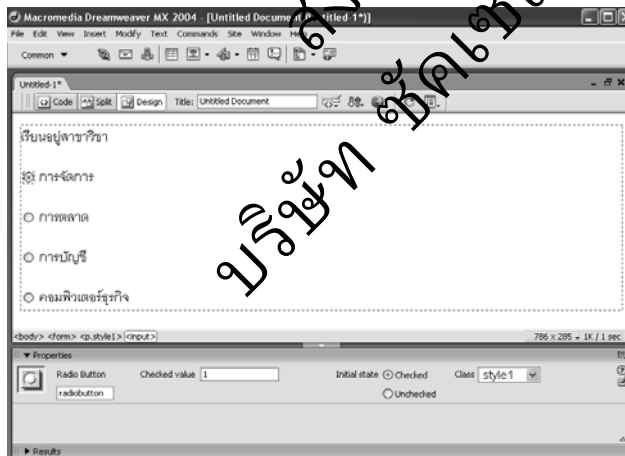
ระบุชื่อของปุ่มตัวเลือก

value=ค่าของปุ่มตัวเลือก

ระบุค่าของปุ่มตัวเลือกที่มีการเลือก

ปุ่มตัวเลือกนี้

ถ้าใช้ Dreamweaver ก็ทำการคลิกที่ปุ่ม  (Radio Button) จะปรากฏปุ่มตัวเลือกขึ้นมา ซึ่งจะต้องสร้างปุ่มตัวเลือกนี้ให้อยู่ภายในฟอร์มดังนี้



รูปที่ 9.5 การสร้างปุ่มตัวเลือกด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของปุ่มตัวเลือกได้
ดังนี้

Checked value

ระบุค่าของปุ่มตัวเลือกแต่ละตัว

Initial state

- Checked ระบุให้ปุ่มตัวเลือกนี้มีการเลือกก็จะปรากฏจุดสีปรากฏขึ้น
- Unchecked ระบุให้ปุ่มตัวเลือกนี้ไม่มีการเลือก ปุ่มตัวเลือกก็จะว่าง

จากรูปที่ 11.5 ได้ทำการสร้างปุ่มตัวเลือกของสาขาวิชาที่สังกัด โดยมีสาขาวิชาอยู่ 4 สาขาวิชา เราสามารถเลือกสาขาวิชาที่สังกัดได้เพียงสาขาเดียวเท่านั้น โค้ด HTML ของปุ่มตัวเลือกในรูปนี้จะเป็นดังนี้

```
เรียนอยู่สาขาวิชา <p>
<input type="radio" name="major" value="1" checked> การจัดการ<p>
<input type="radio" name=" major " value="2"> การตลาด<p>
<input type="radio" name=" major " value="3"> การบัญชี<p>
<input type="radio" name=" major " value="4"> คอมพิวเตอร์ธุรกิจ
```

จากโค้ดข้างต้นนี้ได้ตั้งชื่อปุ่มตัวเลือกของทั้ง 4 สาขาวิชาคือ "major" ซึ่งโดยปกติในปุ่มตัวเลือกของข้อมูลชุดเดียวกันเราจะตั้งชื่อให้แต่ละตัวเป็นชื่อเดียวกันทั้งหมด นอกจากนี้เป็นที่ค่าของปุ่มตัวเลือกก็ได้กำหนดค่าเป็น 1, 2, 3 และ 4 ตามลำดับ โดยถ้ามีการเลือก "การจัดการ" ค่าของ "major" ก็จะมีค่าเท่ากับ 1, ถ้ามีการเลือก "การตลาด" ค่าของ "major" ก็จะมีค่าเท่ากับ 2, ถ้ามีการเลือก "การบัญชี" ค่าของ "major" ก็จะมีค่าเท่ากับ 3 และ ถ้ามีการเลือก "คอมพิวเตอร์ธุรกิจ" ค่าของ "major" ก็จะมีค่าเท่ากับ 4


กล่องรายการ (List Box)

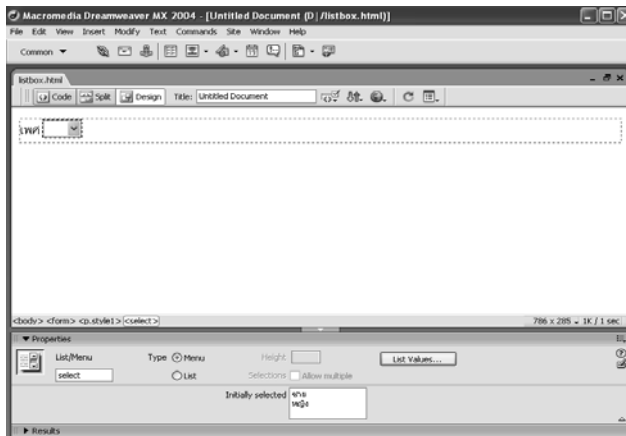
กล่องรายการจะมีลักษณะเป็นกล่องสี่เหลี่ยมใช้บรรจุรายการตัวเลือกต่างๆ ถ้าจะเลือกรายการใดก็ทำการคลิกให้กล่องนี้เลื่อนข้อมูลออกมาจากกล่อง

รูปแบบของโค้ด HTML ของกล่องรายการเป็นดังนี้

รูปแบบ

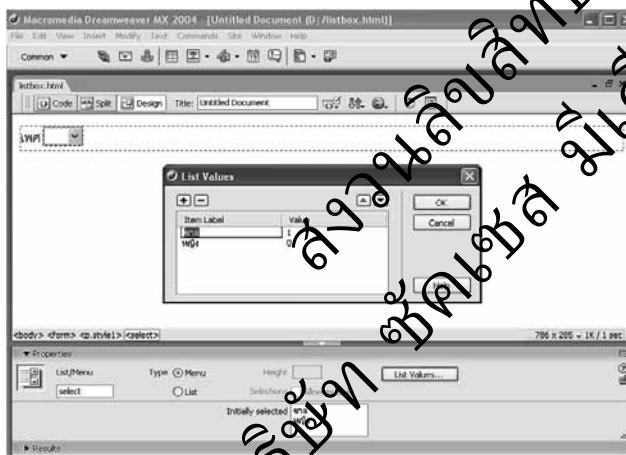
```
<select name=ชื่อของกล่องรายการ>
...
<option value=ค่าของกล่องรายการ>ข้อความ</option>
...
</select>
```

ถ้าใช้ Dreamweaver ก็ทำการคลิกที่ปุ่ม  (List/Menu) จะปรากฏกล่องรายการขึ้นมา ซึ่งจะต้องสร้างกล่องรายการนี้ให้อยู่ภายในฟอร์ม ดังนี้



รูปที่ 9.6 การสร้างกล่องรายการด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถทำการเพิ่มรายการลงในกล่องรายการได้โดยคลิกที่ปุ่ม List Values... ซึ่งเมื่อคลิกที่ปุ่มนี้ก็จะปรากฏหน้าต่างให้ทำการกรอกรายการต่างๆ ดังนี้



รูปที่ 9.7 การใส่รายการลงในกล่องรายการด้วย Dreamweaver

ทดสอบการใช้งานของกล่องรายการนี้ โดยเปิดดูที่โปรแกรมบราวเซอร์จะได้ผลลัพธ์ ดังนี้



รูปที่ 9.8 ลักษณะของ List Box

จากกล่องรายการข้างต้นนี้เป็นกล่องรายการของการเลือกเพศว่าเป็นชายหรือหญิง โค้ด HTML ของกล่องรายการเพศจะเป็นดังนี้

```
<select name="Sex">
  <option value="1">ชาย</option>
  <option value="0">หญิง</option>
</select>
```

จากโค้ดข้างต้นนี้ได้ตั้งชื่อกล่องรายการว่า "Sex" ส่วนที่ค่าของกล่องรายการก็ได้กำหนดค่าเป็น 1 และ 2 โดยถ้ามีการเลือกเพศเป็น "ชาย" ค่าของ "Sex" ก็จะมีค่าเท่ากับ 1 แต่ถ้ามีการเลือกเพศเป็น "หญิง" ค่าของ "Sex" ก็จะมีค่าเท่ากับ 0

ช่องรับชื่อไฟล์ (File Field)

ช่องรับชื่อไฟล์ใช้สำหรับการเลือกไฟล์ โดยมักใช้ในการเลือกไฟล์เพื่ออัปโหลด (Upload) ไฟล์ไปยังเซิร์ฟเวอร์ (Server) ลักษณะจะเป็นเหมือนช่องรับข้อความ แต่จะมีปุ่ม Browse อยู่ข้างๆ ด้วยเพื่อใช้ในการเปิดเข้าไปเลือกไฟล์

รูปแบบโค้ด HTML ของช่องรับชื่อไฟล์เป็นดังนี้

รูปแบบ

```
<input type="file" name="ชื่อของช่องรับชื่อไฟล์">
```

โดยที่

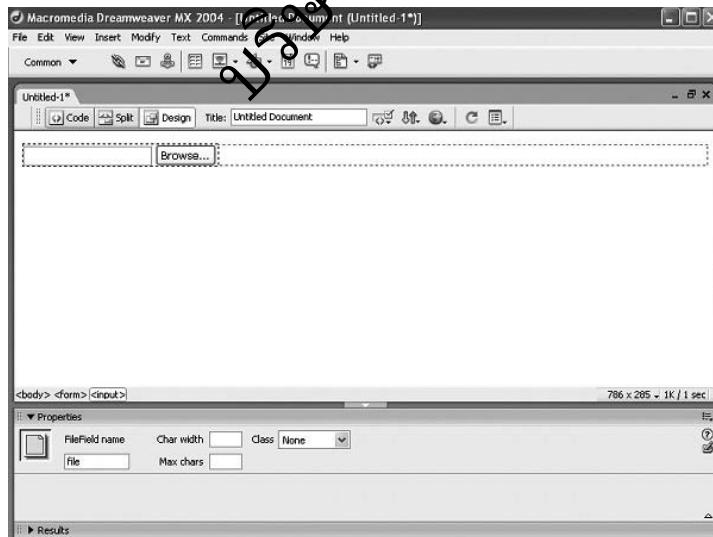
type="file"

ระบุว่าเป็นช่องรับชื่อไฟล์

name=ชื่อของช่องรับชื่อไฟล์

ระบุชื่อของช่องรับชื่อไฟล์

ถ้าใช้ Dreamweaver ก็ทำการคลิกที่  (File Field) จะปรากฏช่องรับชื่อไฟล์ขึ้นมา ซึ่งจะต้องสร้างช่องรับชื่อไฟล์นี้ให้อยู่ภายในฟอร์มดังนี้



รูปที่ 9.9 การสร้างช่องรับชื่อไฟล์ด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับชื่อไฟล์ได้ ดังนี้

- Char width** ระบุขนาดความกว้างของช่องรับชื่อไฟล์
- Max chars** ระบุจำนวนตัวอักษรที่ช่องรับชื่อไฟล์สามารถรับได้

ปุ่มคำสั่ง (Button)

การรับข้อมูลที่ใช้ได้ทำการกรอกหรือเลือก และนำไปประมวลต่อไปจะต้องใช้ปุ่มคำสั่งเพื่อทำการส่งข้อมูลนำเข้าต่างๆ เหล่านั้น (Submit) นอกจากการส่งข้อมูลปุ่มคำสั่งยังสามารถใช้ทำการยกเลิกข้อมูลนำเข้าต่างๆ ได้อีกด้วย

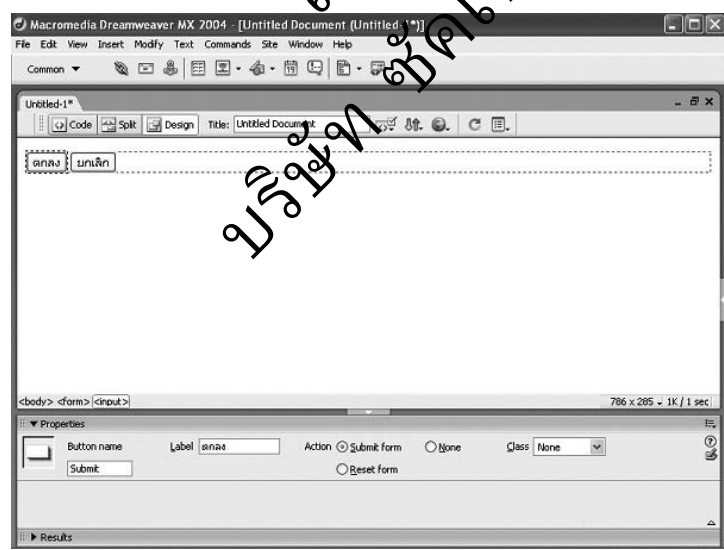
รูปแบบโค้ด HTML ของปุ่มคำสั่งเป็นดังนี้

รูปแบบ

```
<input type="submit" name="ชื่อปุ่ม" value="ข้อความบนปุ่ม">
<input type="reset" name="ชื่อปุ่ม" value="ข้อความบนปุ่ม">
<input type="button" name="ชื่อปุ่ม" value="ข้อความบนปุ่ม">
```

ปุ่มประเภทส่งข้อมูล
ปุ่มประเภทยกเลิกข้อมูล
ปุ่มประเภทธรรมดา

ถ้าใช้ Dreamweaver ก็ทำการคลิกที่ปุ่ม  (Button) จะปรากฏปุ่มคำสั่งขึ้นมา ซึ่งจะต้องสร้างปุ่มคำสั่งนี้ให้อยู่ภายในฟอร์มดังนี้



รูปที่ 9.10 การสร้างปุ่มคำสั่งด้วย Dreamweaver

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของปุ่มคำสั่งได้ ดังนี้

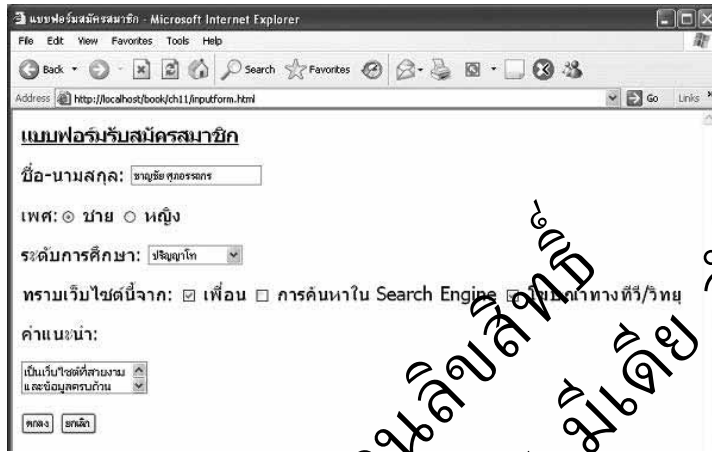
Label

ใช้พิมพ์ข้อความที่ต้องการให้ปรากฏบนปุ่มคำสั่ง

Action

- Submit form ใช้สร้างปุ่มคำสั่งประเภทส่งข้อมูล
- Reset form ใช้สร้างปุ่มคำสั่งประเภทยกเลิกข้อมูล
- None ใช้สร้างปุ่มคำสั่งประเภทธรรมดา

ก่อนที่เราจะศึกษาเรื่องการส่งข้อมูล ให้เราสร้างฟอร์มรับข้อมูลสมัครสมาชิกดังรูปต่อไปนี้



รูปที่ 9.11 ตัวอย่างฟอร์มรับสมัครสมาชิก (inputform.html)

โค้ด HTML ของฟอร์มข้างต้นเป็นดังนี้

inputform.html

```

<html>
<head>
<title>แบบฟอร์มสมัครสมาชิก </title>
</head>
<body>
<form action="memberdata.php" method="post">
  <strong><u>แบบฟอร์มรับสมัครสมาชิก</u></strong><p>
  ชื่อ-นามสกุล: <input type="text" name="name"> <p>
  เพศ:<input name="sex" type="radio" value="1"> ชาย
      <input name="sex" type="radio" value="0"> หญิง<p>
  ระดับการศึกษา:
      <select name="education">
        <option value="1">ต่ำกว่าปริญญาตรี</option>
        <option value="2">ปริญญาตรี</option>

```

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

```

<option value="3">ปริญญาโท</option>
<option value="4">ปริญญาเอก</option>
</select> <p>
ทราบเว็บไซต์นี้จาก:
<input type="checkbox" name="source1" value="1"> เพื่อน
<input type="checkbox" name="source2" value="2"> การค้นหาใน Search Engine
<input type="checkbox" name="source3" value="3"> โฆษณาทางทีวี/วิทยุ <p>
คำแนะนำ:<p>
<textarea name="comment"></textarea> <p>
<input type="submit" name="Submit" value="ตกลง">
<input type="reset" name="Submitz" value="ยกเลิก">
</form>
</body>
</html>

```

9.2 การส่งข้อมูล

หลังจากเราได้ทำการสร้างฟอร์มซึ่งจะออกไปด้วยฟอร์มรับข้อมูลต่างๆ แล้วลำดับถัดไปเราจะทำการเขียนโปรแกรม เพื่อทำการรับข้อมูลนำเข้าสู่ฟอร์ม และนำข้อมูลประมวลผลต่อไป

จากฟอร์มรับสมัครสมาชิกข้างต้น เมื่อผู้ใช้ทำการกรอกข้อมูลเรียบร้อยแล้ว ต้องทำการคลิกที่ปุ่มส่งข้อมูล (ตกลง) ข้อมูลของสมาชิกก็จะถูกส่งไปยังไฟล์ที่ชื่อว่า memberdata.php (ที่อยู่ในเซิร์ฟเวอร์) ทำการประมวลผล ซึ่งระบุไว้ที่คำสั่ง action ในบรรทัดต่อไปนี้

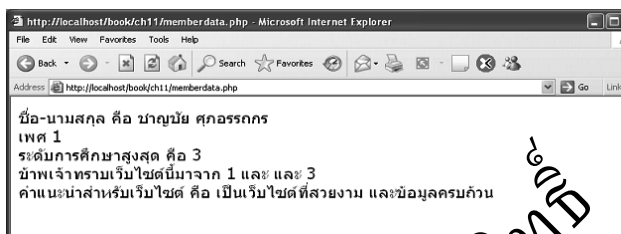
```
<form action="memberdata.php" method="post">
```

ในส่วนของโปรแกรม memberdata.php เราจะทำการเขียนโปรแกรมภาษา PHP เพื่อรับค่าที่ผู้ใช้ นำเข้า โดยมีหลักการง่ายๆ คือ ให้จำชื่อของช่องรับข้อมูลแต่ละตัวจากไฟล์ HTML โดยค่าของฟอร์มรับข้อมูล แต่ละตัวจะเก็บไว้ที่ตัวแปรชื่อเดียวกับชื่อของช่องรับข้อมูล ดังนั้นในการแสดงผลว่าเรามีข้อมูลนำเข้าคืออะไร เราอาจใช้คำสั่ง echo เพื่อพิมพ์ค่าของตัวแปรต่างๆ ซึ่งตัวแปรต่างๆ เราก็เพียงเติมเครื่องหมาย \$ หน้าชื่อช่องรับข้อมูลเท่านั้น ดังนั้นโปรแกรม memberdata.php สามารถเขียนได้ ดังนี้

memberdata.php

```
<?
echo "ชื่อ-นามสกุล คือ $name <br>";
echo "เพศ $sex <br>";
echo "ระดับการศึกษาสูงสุด คือ $education <br>";
echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก $source1 และ $source2 และ $source3 <br>";
echo "คำแนะนำสำหรับเว็บไซต์ คือ $comment";
?>
```

ผลลัพธ์จากการส่งข้อมูลเป็นดังรูปต่อไปนี้

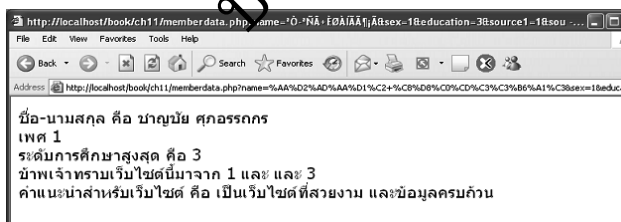


รูปที่ 9.12 ผลลัพธ์ของโปรแกรม memberdata.php โดยใช้วิธี Post ส่งข้อมูล

จากตัวอย่างข้างต้นนี้ ในส่วนของไฟล์ inputform.html เป็นการส่งข้อมูลแบบ Post ซึ่งเราจะไม่เห็น ว่าผู้ใช้ทำการนำข้อมูลอะไร (สังเกตในช่อง Address) ถ้าสมมติลองเปลี่ยนวิธีการส่งข้อมูลเป็นแบบ Get ในบรรทัดคำสั่ง ดังนี้

```
<form action="memberdata.php" method="Get">
```

หลังจากนั้นลองกรอกข้อมูลใหม่ และคลิกปุ่มส่งข้อมูลจะได้ผลลัพธ์เหมือนเดิม แต่ให้สังเกตที่ช่อง Address จะเห็นว่าค่าของข้อมูลนำเข้าถูกรวมเข้ากับ URL ดังนี้



ค่าของข้อมูลนำเข้าจะถูกรวม
เข้ากับ URL

รูปที่ 9.13 ผลลัพธ์ของโปรแกรม memberdata.php โดยใช้วิธี Get ส่งข้อมูล

จากการแสดงผลของโปรแกรม memberdata.php เราจะสังเกตเห็นว่า ข้อมูลบางอย่างแสดงค่าไม่สื่อว่าหมายถึงอะไร เช่น ระดับการศึกษา สูงสุด คือ 3 ทำให้ไม่ทราบได้ว่าจบการศึกษาระดับไหนกันแน่ ดังนั้นเราสามารถทำการปรับปรุงโปรแกรม memberdata.php และทำการ save as เพื่อเปลี่ยนชื่อไฟล์เป็น memberdata2.php และในส่วนของไฟล์ inputform.html ก็ให้ทำการเปลี่ยนบรรทัดคำสั่ง action เป็นดังนี้

```
<form action="memberdata2.php" method="Post">
```

หลังจากนั้นก็ทำการ save as เพื่อเปลี่ยนชื่อไฟล์เป็น inputform2.html

โปรแกรม memberdata2.php หลังการปรับปรุงแล้วเป็นดังนี้

memberdata2.php

```
<?
    echo "ชื่อ-นามสกุล คือ $name <br>";
    if ($sex == "1")
    {
        echo "เพศชาย <br>";
    }
    else
    {
        echo "เพศหญิง <br>";
    }

    switch ($education)
    {
    case "1": echo "ระดับการศึกษาสูงสุด คือ ต่ำกว่าปริญญาตรี <br>"; break;
    case "2": echo "ระดับการศึกษาสูงสุด คือ ปริญญาตรี <br>"; break;
    case "3": echo "ระดับการศึกษาสูงสุด คือ ปริญญาโท <br>"; break;
    case "4": echo "ระดับการศึกษาสูงสุด คือ ปริญญาเอก <br>"; break;
    }

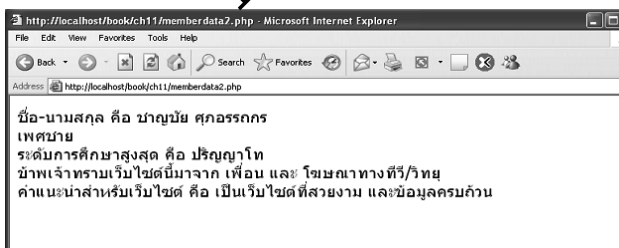
    if (($source1=="1") && ($source2=="2") && ($source3=="3"))
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก เพื่อน และ การค้นหาจาก Search Engine
        และ โฆษณาทางทีวี/วิทยุ <br>";
    }
    elseif (($source1=="1") && ($source2=="2"))
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก เพื่อน และ การค้นหาจาก Search Engine
        <br>";
    }
    elseif (($source1=="1") && ($source3=="3"))
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก เพื่อน และ โฆษณาทางทีวี/วิทยุ <br>";
    }
}
```

```

    }
    elseif ($source1=="1")
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก เพื่อน <br>";
    }
    elseif (($source2=="2") && ($source3=="3"))
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก การค้นหาจาก Search Engine และ
        โฆษณา ทางทีวี/วิทยุ <br>";
    }
    elseif ($source2=="2")
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก การค้นหาจาก Search Engine <br>";
    }
    elseif ($source3=="3")
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก โฆษณาทางทีวี/วิทยุ <br>";
    }
    else
    {
        echo "ข้าพเจ้าทราบเว็บไซต์นี้มาจาก ท่านไม่ได้เลือกตัวเลือกนี้ <br>";
    }
    echo "คำแนะนำสำหรับเว็บไซต์ คือ <comment>";
?>

```

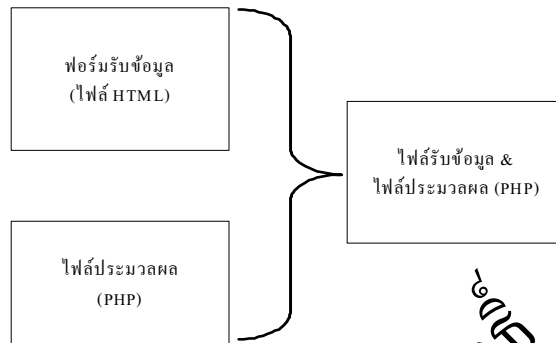
ผลลัพธ์หลังจากการกรอกข้อมูลนำเข้า และคลิกที่ปุ่มคำสั่งเพื่อส่งข้อมูลของไฟล์ inputform2.html ไปยังโปรแกรม memberdata2.php จะเป็นดังนี้



รูปที่ 9.14 ผลลัพธ์ของโปรแกรม memberdata2.php

9.3 การรวมไฟล์ HTML และ PHP

จากตัวอย่างในหัวข้อที่ผ่านมาเราจะสังเกตเห็นว่าในการประมวลผลฟอร์มรับข้อมูล เราจะต้องสร้างไฟล์ที่เป็นฟอร์มรับข้อมูล (HTML) และไฟล์ที่ทำการรับข้อมูลจากไฟล์ฟอร์มตัวแรก (PHP) เพื่อนำค่าที่ส่งผ่านเข้าไปประมวลผลต่อไป แต่ในการเขียนโปรแกรมเราอาจจะทำการรวมไฟล์ทั้งสองเป็นไฟล์เพียงไฟล์เดียวก็ได้ พิจารณาลักษณะการทำงานของกรรวมไฟล์ HTML และ PHP ดังนี้



รูปที่ 9.15 ลักษณะการรวมไฟล์ HTML และ PHP

ตัวอย่างต่อไปนี้จะแสดงฟอร์มรับข้อมูล Username และ Password ของผู้ใช้ หลังจากนั้นจะทำการแสดงผลลัพธ์ของทั้ง 2 ค่าที่ผู้ใช้กรอกลงในฟอร์ม โดยเขียนโปรแกรมในลักษณะรวมไฟล์ HTML และ PHP ไว้ในไฟล์เดียวกัน ทำให้ลดจำนวนไฟล์ข้อมูลของระบบงานลง ดังนี้

login.php

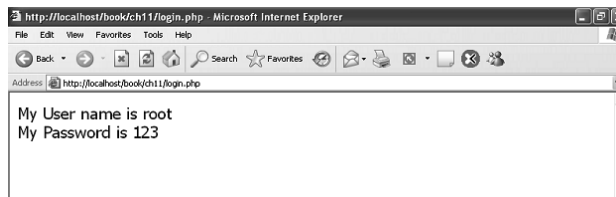
```
<?
    if(!isset) {
?>
    <form method="post" action="<? $PHP_SELF; ?>">

    <b>Please enter Username and Password</b><br>
    User name: <input type="text" name="user"><br>

    Password: <input type="password" name="pass"><br>
    <input type="submit" value="OK" name="send">
    </form>
<?
    }
    else
    { echo "My Username is $user <br>";
      echo "My Password is $pass ";
    }
?>
```



รูปที่ 9.16 φόρμรับข้อมูลชื่อผู้ใช้ และรหัสผ่าน (login.php)



รูปที่ 9.17 ไฟล์แสดงผลรหัสข้อมูลที่ผู้ใช้กรอก (login.php)

อธิบายการทำงานของโปรแกรมนี้ได้ ดังนี้

- if(!\$send) ตรวจสอบว่าปุ่ม submit (OK) ซึ่งตั้งชื่อว่า send ถูกคลิกหรือไม่
- ถ้าปุ่ม Submit ไม่ได้ถูกคลิก ตัวแปร \$send จะเป็นเท็จ ดังนั้น if(!\$send) จะกลายเป็นจริง

ก็จะทำบรรทัดคำสั่ง

```
<form method="post" action="<? $PHP_SELF; ?>">
  <b>Please enter Username and Password</b><br>
  User name: <input type="text" name="user"><br>

  Password: <input type="password" name="pass"><br>
  <input type="submit" value="OK" name="send">
</form>
```

ซึ่งเป็นการสร้างฟอร์มรับข้อมูลนั่นเอง สังเกตที่คำสั่ง action="<? \$PHP_SELF; ?>" เป็นการบอกให้ฟอร์มรับข้อมูลนี้ส่งข้อมูลที่ใช้กรอกไปยังไฟล์ของตัวเอง

• แต่ถ้าปุ่ม Submit ถูกคลิก ตัวแปร \$send จะเป็นจริง ดังนั้น if(!\$send) จะกลายเป็นเท็จ ก็จะทำบรรทัดคำสั่ง

```
else
{ echo "My Username is $user <br>";
  echo "My Password is $pass ";
}
```

โดยเป็นการใช้คำสั่ง echo เพื่อพิมพ์ค่าของ Username และ Password ออกมานั่นเอง

สรุป

ในการสร้างฟอร์มรับข้อมูลจะประกอบขึ้นจากช่องรับข้อมูลต่างๆ คือ ฟอร์ม (Form), ช่องรับข้อความ (Text Field), ช่องรับข้อความแบบหลายบรรทัด (Text Area), กล่องตัวเลือก (Check box), ปุ่มตัวเลือก (Radio Button), กล่องรายการ (List box), ช่องรับชื่อไฟล์ (File Field), ปุ่มคำสั่ง (Button)

ในการส่งข้อมูลแบ่งออกได้เป็น 2 วิธี คือ วิธี Post และ Get โดยที่วิธี Post ผู้ใช้จะไม่เห็นว่ามีกรนำเข้าข้อมูลอะไร ส่วนวิธี Get ผู้ใช้จะสามารถมองเห็นว่าค่าของข้อมูลนำเข้าคืออะไรโดยค่าของข้อมูลนำเข้าจะถูกรวมเข้ากับ URL

โดยปกติการเขียนโปรแกรมส่งและรับไฟล์จะประกอบไปด้วยไฟล์ 2 ไฟล์ คือ ฟอร์มรับข้อมูล (ปกติจะเขียนเป็น HTML) และไฟล์ที่รับข้อมูลจากฟอร์มเพื่อนำไปประมวลผล (ปกติจะเขียนด้วย PHP) แต่เราสามารถเขียนโปรแกรมโดยรวมไฟล์ทั้งสองเข้าด้วยกันเพียง 1 ไฟล์ ซึ่งจะทำให้การจัดการไฟล์ในระบบงานไม่ยุ่งยาก เนื่องจากจำนวนไฟล์มีไม่มากนัก

แบบฝึกหัดบทที่ 9

1. ให้ออกแบบฟอร์มรับสมัครงาน โดยประกอบไปด้วยข้อมูล ชื่อ-สกุล, วันเดือนปีเกิด, เพศ, ระดับการศึกษา และประสบการณ์ทำงาน
2. ให้ออกแบบฟอร์มการขอเข้าชมภาพยนตร์ออนไลน์ โดยประกอบไปด้วยข้อมูล ชื่อ-สกุลผู้จอง, ชื่อภาพยนตร์, วันเดือนปีที่ต้องการจอง, รอบฉาย, ที่นั่ง
3. ให้เขียนโปรแกรมภาษา PHP แสดงผลลัพธ์ของฟอร์มรับข้อมูลในข้อที่ 1.
4. ให้เขียนโปรแกรมภาษา PHP แสดงผลลัพธ์ของฟอร์มรับข้อมูลในข้อที่ 2.
5. ให้อธิบายความแตกต่างระหว่างการส่งข้อมูลของฟอร์มแบบ Post และ Get

Session และ Cookie เป็นคุณสมบัติหนึ่งที่สำคัญของโปรแกรมภาษา PHP สำหรับติดตามและตรวจสอบผู้ใช้ เช่น การสร้างแบบฟอร์มเพื่อให้สมาชิกหรือผู้ดูแลระบบทำการล็อกอิน (Log in) โดยจะต้องกรอกชื่อผู้ใช้ (User name) และรหัสผ่าน (Password) ในการเขียนโปรแกรมนี้เราต้องตรวจสอบว่าชื่อผู้ใช้และรหัสผ่านถูกต้องหรือไม่ ถ้าถูกต้องผู้ใช้งานจะสามารถเข้าไปใช้งานต่างๆ ภายในระบบได้ ถ้ามองเพียงผิวเผินในการเขียนโปรแกรมเราจะใช้คำสั่งตรวจสอบเงื่อนไขเพื่อเปรียบเทียบว่าชื่อผู้ใช้ และรหัสผ่านที่กรอกเหมือนกับชื่อผู้ใช้ และรหัสผ่านที่ถูกต้องหรือไม่ แต่ถ้าเราทำระบบล็อกอินจะต้องหาวิธีที่สามารถนุกรุกเข้าไปใช้งานภายในระบบได้ เนื่องจากผู้ใช้อาจจะทำการจด IP ของหน้าที่เข้าไปใช้งานภายในระบบ และก็เข้าถึงหน้านั้นโดยไม่ต้องผ่านหน้าล็อกอิน แต่ถ้าใช้คำสั่ง Session และ Cookie จะสามารถแก้ปัญหาเหล่านี้ได้

จุดประสงค์ทั่วไป

1. เพื่อศึกษาการใช้ Session และ Cookie มาประยุกต์ใช้ในการเขียนโปรแกรม
2. เพื่อเรียนรู้ฟังก์ชันที่เกี่ยวกับ Session
3. เพื่อเรียนรู้ฟังก์ชันที่เกี่ยวกับ Cookie
4. เพื่อสามารถใช้เขียนโปรแกรมโดยใช้เทคนิคของ Session และ Cookie

จุดประสงค์เชิงพฤติกรรม

1. บอกความหมายของ session และ cookie
2. สามารถเขียนโปรแกรมโดยใช้ฟังก์ชัน session และ cookie ได้
3. สามารถประยุกต์ฟังก์ชัน session และ cookie กับโปรแกรม Log in ได้

เนื้อหาสาระ

1. ฟังก์ชันที่เกี่ยวกับ Session
2. ฟังก์ชันที่เกี่ยวกับ Cookie
3. การประยุกต์ Session และ Cookie ในโปรแกรม Log in

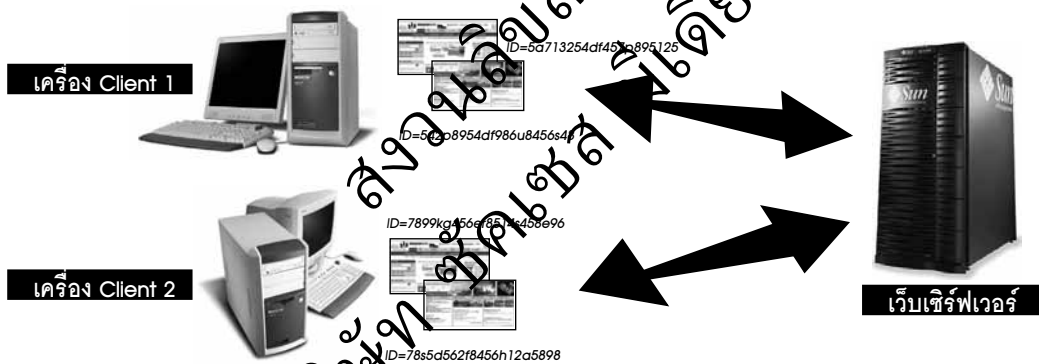
10.1 Session คืออะไร

การที่เราเปิดโปรแกรมบราวเซอร์ขึ้นมา แล้วทำการร้องขอเพิ่มเอกสารไปยังเว็บเซิร์ฟเวอร์ หากเพิ่มเอกสารดังกล่าวเป็นเพิ่มเอกสารธรรมดาทั่วไปแล้ว การทำงานก็ถือว่าเป็นธรรมดาปกติทั่วไป

แต่หากว่า ถ้าเพิ่มเอกสารดังกล่าวที่เราร้องขอไปยังเว็บเซิร์ฟเวอร์ มีการระบุโค้ดในการสั่ง Start Session ขึ้นมาแล้ว โปรแกรมบราวเซอร์ของเราจะได้รับหมายเลขประจำตัวที่ส่งมาจากเว็บเซิร์ฟเวอร์ขึ้นมาทันที โดยที่เราเรียกมันว่า **Session ID** ซึ่งถูกเข้ารหัสอยู่ในรูปแบบของ MD5 (อักขระผสมกับตัวเลข) ทำให้โคลเอนต์ที่ร้องขอเข้ามาจะได้รับ Session ID ที่ไม่มีทางซ้ำกันได้เลย

เมื่อโคลเอนต์แต่ละตัวได้รับ Session ID แล้ว เราก็สามารถอ้างอิงค่าของ Session ID ไปได้ยังทุกเว็บเพจ จนกว่าโคลเอนต์นั้นๆ จะทำการปิดหน้าต่างของโปรแกรมบราวเซอร์ลง เราก็ไม่สามารถอ้างอิงค่า Session ID ของโคลเอนต์ดังกล่าวได้อีกต่อไป

นอกจากนี้ โคลเอนต์จะทำการเปิดโปรแกรมบราวเซอร์ขึ้นมาใหม่ ก็จะส่งผลให้หน้าต่างของโปรแกรมบราวเซอร์ที่ถูกเปิดขึ้นมาใหม่นั้นได้รับ Session ID ใหม่หนึ่งอัน



จากรูป เราจะเห็นได้ว่ามีโคลเอนต์อยู่ 2 ตัว โดยแต่ละตัวมีการเปิดโปรแกรมบราวเซอร์ขึ้นมา 2 หน้าต่าง แล้วทำการร้องขอไปยังเว็บเซิร์ฟเวอร์ ส่งผลให้แต่ละหน้าต่างของโปรแกรมบราวเซอร์แต่ละตัวได้รับ Session ID ที่ไม่เหมือนกัน ทำให้เว็บเซิร์ฟเวอร์สามารถติดตามข้อมูลทางฝั่งของโคลเอนต์ได้ตลอดที่โปรแกรมบราวเซอร์นั้นเปิดใช้งานอยู่นั่นเอง

ด้วยความสามารถของ Session เราสามารถนำไปใช้ในการพัฒนาเว็บแอปพลิเคชันต่างๆ เช่น ร้านค้าออนไลน์ หรือการตรวจสอบผู้ใช้ที่ออนไลน์อยู่ขณะนั้น เป็นต้น

10.2 ฟังก์ชันที่เกี่ยวข้องกับ Session

Session มีคุณสมบัติใช้ส่งค่าของตัวแปรที่สร้างขึ้นจากไฟล์หนึ่งไปแสดงผลยังอีกไฟล์ได้ ทำให้เราสามารถทำการติดตามและตรวจสอบผู้ใช้ได้ ในการใช้งาน Session ภาษา PHP ได้จัดเตรียมฟังก์ชันให้ใช้งานอยู่หลายฟังก์ชัน ในที่นี้จะขอกล่าวถึงเฉพาะบางฟังก์ชันที่จำเป็นเท่านั้น ดังนี้

- **session_start()** ใช้กำหนดให้ข้อมูลของ session เริ่มต้นทำงาน
- **session_register()** ใช้สำหรับลงทะเบียนให้กับตัวแปร ให้มีคุณสมบัติเป็นตัวแปร session
- **session_unregister()** ใช้สำหรับยกเลิกการลงทะเบียนของตัวแปร
- **session_destroy()** ใช้สำหรับทำลายข้อมูลทั้งหมดของ session

ฟังก์ชัน session_start()

ฟังก์ชันนี้ใช้กำหนดให้ข้อมูลของ session เริ่มต้นทำงาน ตัวอย่างต่อไปนี้จะใช้ฟังก์ชัน session_start() เพื่อกำหนดให้โปรแกรมเริ่มต้นเก็บค่าของตัวแปร session ที่ไฟล์ page1.php หลังจากนั้นก็จะทำการเรียกให้ไฟล์ page2.php ทำการแสดงค่าของตัวแปร session ดังกล่าว

โปรแกรมของทั้ง 2 ไฟล์เป็น ดังนี้

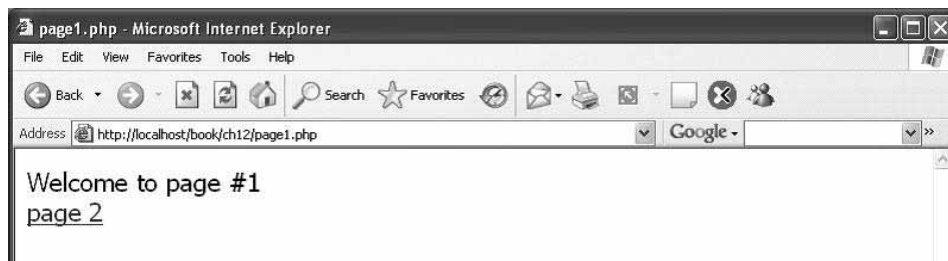
page1.php

```
<?
    session_start();
    echo 'Welcome to page #1';
    $_SESSION['color'] = 'green';
    $_SESSION['animal'] = 'cat';
    $_SESSION['time'] = time();
    echo '<br /><a href="page2.php">page 2</a>';
?>
```

page2.php

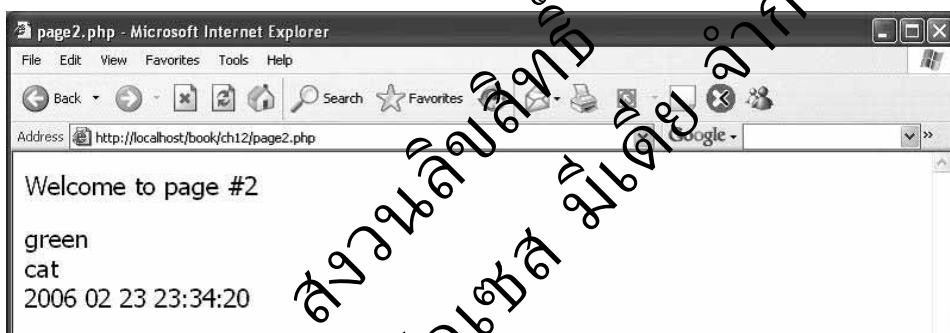
```
<?
    session_start();
    echo 'Welcome to page #2 <p>';
    echo $_SESSION['color'].'<br>';
    echo $_SESSION['animal'].'<br>';
    echo date('Y m d H:i:s', $_SESSION['time']);
?>
```

ผลลัพธ์ของ page1.php เป็น ดังนี้



รูปที่ 10.1 ผลลัพธ์ของโปรแกรม page1.php

เมื่อเรียกโปรแกรม page1.php ทำงานแล้ว จะมีการกำหนดค่าของตัวแปร `$_SESSION` ซึ่งเป็นตัวแปรเฉพาะของ session โดยเป็นตัวแปรประเภทอาร์เรย์ ในที่นี้กำหนดให้เก็บค่าของสี, ประเภทสัตว์และวันเวลาปัจจุบันของเครื่อง หลังจากนั้นเมื่อทำการคลิกที่ลิงค์ page 2 จะได้ผลลัพธ์เป็นดังนี้



รูปที่ 10.2 ผลลัพธ์ของโปรแกรม page2.php

ฟังก์ชัน `session_register()`

ฟังก์ชันนี้ใช้สำหรับลงทะเบียนให้กับตัวแปรใดๆ ทำให้มีคุณสมบัติเป็นตัวแปร session ซึ่งมีลักษณะเป็นตัวแปรประเภท global หรือตัวแปรสาธารณะ คือสามารถเรียกใช้งานตัวแปรตัวนี้ได้ทุกที่

รูปแบบการใช้งานของฟังก์ชันนี้เป็นดังนี้

รูปแบบ

```
session_register("ชื่อตัวแปร1", "ชื่อตัวแปร2", ... )
```

พิจารณาตัวอย่างการใช้งานของฟังก์ชันนี้โดยไฟล์โปรแกรมที่ 1 จะทำการลงทะเบียนให้กับตัวแปร (เรียกใช้ฟังก์ชัน `session_register()`) ที่ได้ทำการกำหนดค่าเอาไว้ หลังจากนั้นก็ทำการคลิกลิงค์ไปยังไฟล์โปรแกรมที่ 2 ซึ่งมีการสั่งให้เริ่มต้นการทำงานของตัวแปร session (เรียกใช้ฟังก์ชัน `session_start()`) เพื่อให้สามารถแสดงค่าของตัวแปรที่ได้กำหนดค่าไว้ที่ไฟล์ที่ 1

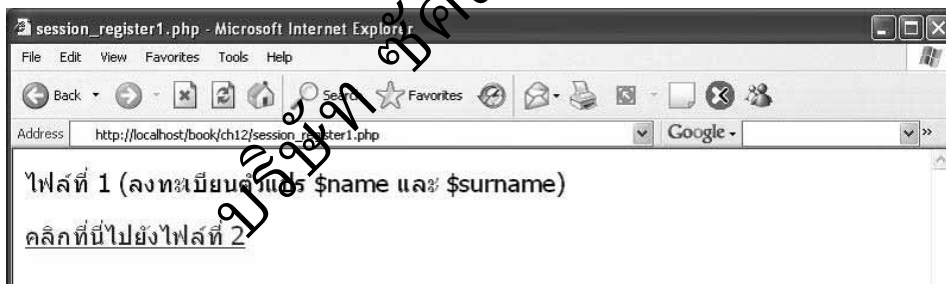
session_register1.php

```
<?
echo "ไฟล์ที่ 1 (ลงทะเบียนตัวแปร \$name และ \$surname) <p>";
$name = "ชาญชัย";
$surname = "ศุภอรรณกร";
session_register("name","surname");
echo "<a href=session_register2.php>คลิกที่นี่ไปยังไฟล์ที่ 2</a>";
ob_end_flush();
?>
```

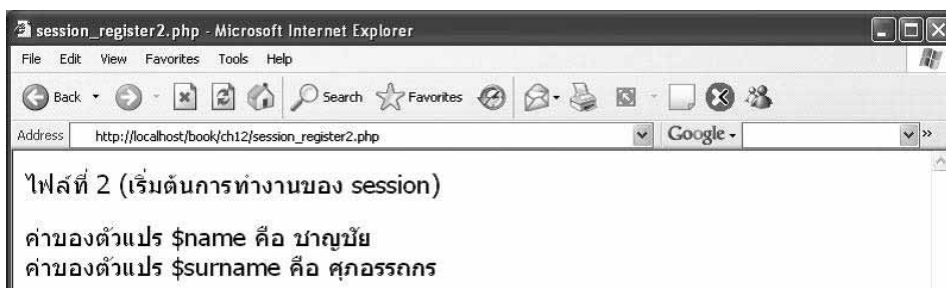
session_register2.php

```
<?
echo "ไฟล์ที่ 2 (เริ่มต้นการทำงานของ session) <p>";
session_start();
echo "ค่าของตัวแปร \$name คือ $name";
echo "ค่าของตัวแปร \$surname คือ $surname";
ob_end_flush();
?>
```

ผลลัพธ์ของโปรแกรม session_register1.php และหลังจากคลิกลิงค์ไปยังโปรแกรม session_register2.php จะมีผลลัพธ์ ดังนี้



รูปที่ 10.3 ผลลัพธ์ของโปรแกรม session_register1.php



รูปที่ 10.4 ผลลัพธ์ของโปรแกรม session_register2.php

ฟังก์ชัน `session_unregister()`

ฟังก์ชันนี้ใช้สำหรับยกเลิกการลงทะเบียนของตัวแปร `session` ที่เคยลงทะเบียนเอาไว้ ทำให้ตัวแปรที่ถูกยกเลิกนั้นไม่เก็บค่าข้อมูลใดๆ ซึ่งการยกเลิกการลงทะเบียนเราจะกระทำได้ฟังก์ชันละ 1 ตัวแปรเท่านั้น รูปแบบการใช้งานของฟังก์ชันนี้เป็น ดังนี้

รูปแบบ

```
session_unregister("ชื่อตัวแปร")
```

ต่อเนื่องจากฟังก์ชันที่ผ่านมา ที่ได้ลงทะเบียนตัวแปร `$name` และ `$surname` ไว้ โปรแกรมต่อไปนี้จะทำการยกเลิกการลงทะเบียนเฉพาะตัวแปร `$name` เท่านั้น แต่ตัวแปร `$surname` ยังคงทำงานตามปกติ ดังนั้นค่าของตัวแปร `$name` จะไม่ได้เก็บค่าใดๆ ในขณะที่ตัวแปร `$surname` ยังคงเก็บค่า "ศุภอรธกร" ตามเดิม

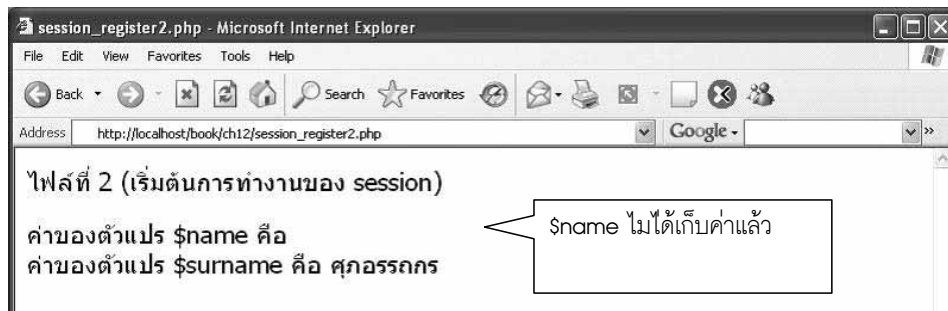
`session_unregister.php`

```
<?
    session_start();
    echo "ยกเลิกการลงทะเบียนตัวแปร $name";
    session_unregister("name");
    echo "<a href=session_register2.php>คลิกที่นี่ไปยังไฟล์ที่ 2</a>";
?>
```

ทดสอบการทำงานโดยเรียกโปรแกรม `session_register1.php` และ `session_register2.php` ซึ่งจะได้ผลลัพธ์เหมือนเดิมตามรูปที่ 10.3 และ 10.4 หลังจากนั้นให้ทำการเรียกโปรแกรม `session_unregister.php` และคลิกลิงค์ไปยังโปรแกรม `session_register2.php` อีกครั้ง จะสังเกตเห็นว่าเฉพาะตัวแปร `$surname` เท่านั้นที่มีค่าเก็บไว้เหมือนเดิม ผลลัพธ์ของโปรแกรม `session_unregister.php` และ `session_register2.php` เป็นดังนี้



รูปที่ 10.5 ผลลัพธ์ของโปรแกรม `session_unregister.php`



รูปที่ 10.6 ผลลัพธ์ของโปรแกรม session_register2.php

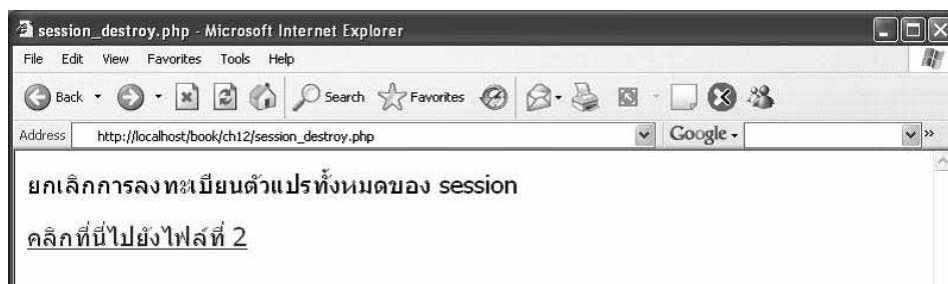
ฟังก์ชัน session_destroy()

ฟังก์ชันนี้ใช้สำหรับทำลายข้อมูลทั้งหมดของ session ต่างจากฟังก์ชัน session_unregister() ที่ต้องระบุชื่อตัวแปรที่ต้องการยกเลิกการลงทะเบียน ถ้าตัวแปรตัวไหนไม่ถูกยกเลิกการลงทะเบียนก็ยังคงเก็บข้อมูลไว้อยู่ ทดสอบการทำงานของฟังก์ชันนี้โดยสร้างโปรแกรมให้ทำลายตัวแปร \$name และ \$surname ที่สร้างจากโปรแกรม session_register1.php ดังนี้

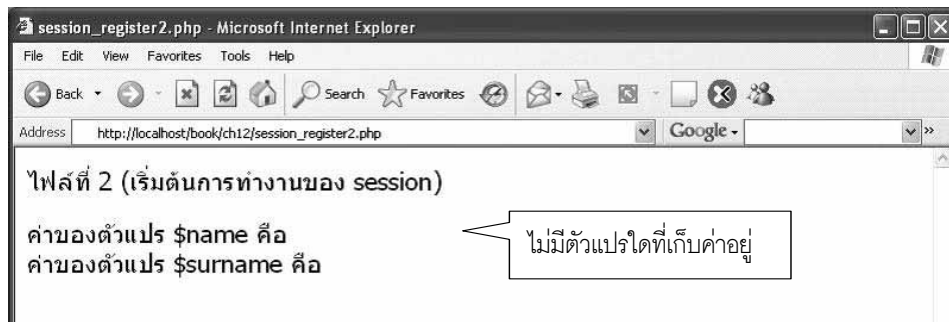
session_destroy.php

```
<?
    session_start();
    echo "ยกเลิกการลงทะเบียนตัวแปรทั้งหมดของ session <p>";
    session_destroy();
    echo "<a href=session_register2.php>คลิกที่นี่ไปยังไฟล์ที่ 2</a>";
?>
```

ทดสอบการทำงานโดยเริ่มจากโปรแกรม session_register1.php และ session_register2.php ซึ่งจะได้ผลลัพธ์เหมือนเดิมตามรูปที่ 10.3 และ 10.4 หลังจากนั้นให้ทำการเรียกโปรแกรม session_destroy.php และคลิกลิงค์ไปยังโปรแกรม session_register2.php อีกครั้ง จะสังเกตเห็นว่าตัวแปร \$name และ \$surname ถูกทำลายค่าที่เก็บไว้แล้ว ผลลัพธ์ของโปรแกรม session_destroy.php และ session_register2.php เป็นดังนี้



รูปที่ 10.7 ผลลัพธ์ของโปรแกรม session_destroy.php



รูปที่ 10.8 ผลลัพธ์ของโปรแกรม session_register2.php

คุกกี้ (Cookies)

เนื่องจากเครื่องบริการ หรือเว็บเซิร์ฟเวอร์ได้เปิดบริการในหลายการเชื่อมต่อ (Connection) ด้วยกัน โดยแต่ละเว็บเซิร์ฟเวอร์อาจมีการจำกัดจำนวนการเชื่อมต่อ (Limited Connection) หรือไม่มีการจำกัดจำนวนการเชื่อมต่อก็ได้ (Unlimited Connection) ดังนั้น เครื่องผู้ใช้ (ไคลเอนต์หรือเครื่องของบราวเซอร์) ที่ร้องขอเข้ามาใช้บริการก็จะเกิดการเชื่อมต่อในหลายๆ Connection ด้วยกัน โดยแต่ละ Connection ที่ทำการเชื่อมต่อร้องขอไฟล์ข้อมูลเข้ามานั้น เมื่อเว็บเซิร์ฟเวอร์ทำการส่งไฟล์ข้อมูลดังกล่าวให้กับ Connection นั้นจนครบแล้ว ก็จะมีการปิดการเชื่อมต่อไปทันที (Close Connection) ทั้งนี้เพื่อต้องการเปิดรับการเชื่อมต่อของ Connection ใหม่เข้ามาแทน

ส่งผลให้เกิดเป็นวงจรของการเชื่อมต่อแบบนี้เรื่อยๆ เพื่อที่เว็บเซิร์ฟเวอร์จะสามารถจัดการกับผู้ใช้จำนวนมากๆ ในแต่ละ Connection ได้ ทำให้เกิดประสิทธิภาพและความรวดเร็วในการติดต่อร้องขอบริการกับเว็บเซิร์ฟเวอร์

แต่ประเด็นอยู่ตรงที่ว่า เมื่อเกิดเว็บเซิร์ฟเวอร์ต้องการติดต่อกับฝั่งผู้ใช้ขึ้นมา จะทำอย่างไร เราอาจคิดว่าน่าจะมีการสร้างตัวแปรขึ้นมาเก็บค่าในการทำงานจริงๆ เราไม่สามารถทำได้ เพราะเมื่อเกิดการปิดการเชื่อมต่อ ตัวแปรที่สร้างไว้ก็จะหายไป

ด้วยเหตุผลนี้เอง จึงถือกำเนิดเจ้าตัวคุกกี้ขึ้นมา ต่อไปเราจะมาดูกันว่า คุกกี้คืออะไร

คุกกี้ (Cookies) คืออะไร

คุกกี้ คือข้อมูลที่เราส่งไปเก็บไว้ในเครื่องของผู้ใช้ เพื่อประโยชน์ในการตรวจสอบ และติดตามการทำงานของผู้ใช้ โดยบราวเซอร์จะทำการส่งข้อมูลของผู้ใช้ที่จัดเก็บไว้ในเครื่องมายังเว็บเซิร์ฟเวอร์ และจะนำ Cookie ดังกล่าวมาเก็บไว้ในตัวแปรทำให้สามารถตรวจสอบค่าต่างๆ ของผู้ใช้ได้

10.3 ฟังก์ชันที่เกี่ยวข้องกับ Cookie

ฟังก์ชันที่สำคัญในการจัดการ Cookie มีดังนี้

- **setcookie()** ใช้สำหรับสร้างและทำลายตัวแปร cookie
- **ob_start()** และ **ob_end_flush()** เพื่อเปิดและปิดการใช้งานหน่วยความจำชั่วคราว

ฟังก์ชัน setcookie()

ฟังก์ชันนี้ใช้สำหรับการสร้าง และทำลายตัวแปร cookie โดยรูปแบบของฟังก์ชันเป็น ดังนี้

setcookie(ชื่อของ cookie, ค่าของ cookie, เวลาหมดอายุ)	สร้างตัวแปร cookie โดยมีอายุของตัวแปรเท่ากับที่ระบุไว้ ถ้าพ้นของเวลาดังกล่าวตัวแปร cookie นี้จะไม่เก็บค่าใดๆ ไว้
setcookie(ชื่อของ cookie, ค่าของ cookie)	สร้างตัวแปร cookie โดยไม่มีระยะเวลาหมดอายุ
setcookie(ชื่อของ cookie)	ทำลายตัวแปร cookie ทำให้ตัวแปร cookie นี้จะไม่เก็บค่าใดๆ ไว้

ทดสอบการทำงานของฟังก์ชัน setcookie() โดยเขียนโปรแกรมทั้งหมด 3 ไฟล์ ดังนี้

- โปรแกรม cookiemake.php สร้างตัวแปร cookie ขึ้นมา 2 ตัวแปร คือ \$username และ \$password ให้มีอายุ 60 วินาที และมีลิงค์ไปยังโปรแกรม showcookie.php
- โปรแกรม showcookie.php แสดงผลลัพธ์ของตัวแปร cookie ทั้ง 2 ตัวแปร
- โปรแกรม delcookie.php ทำลายตัวแปร cookie ทั้ง 2 ตัวแปร

cookiemake.php

```
<?
    setcookie("username", "charnchai", time()+60);
    setcookie("password", "12345", time()+60);
    echo "<a href=showcookie.php>คลิกที่นี่ เพื่อตรวจสอบค่า cookie</a>";
?>
```

showcookie.php

```
<?
echo "<u><b>ตรวจสอบค่าของ cookie </b></u><p>";
echo "ชื่อผู้ใช้ $username <br>";
echo "รหัสผ่าน $password";
?>
```

delcookie.php

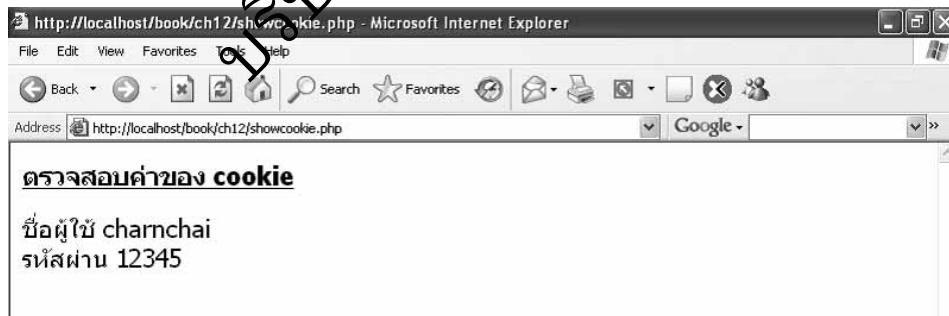
```
<?
setcookie("username");
setcookie("password");
echo "<a href=showcookie.php>คลิกที่นี่ เพื่อตรวจสอบค่า cookie</a>";
?>
```

เริ่มทดสอบโดยเรียกโปรแกรม cookiemake.php จากระบบหน้าจอดังต่อไปนี้



รูปที่ 10.9 ผลลัพธ์ของโปรแกรม cookiemake.php

หลังจากนั้นทำการคลิกที่ลิงค์ "คลิกที่นี่เพื่อตรวจสอบค่า cookie" ซึ่งจะแสดงผลพบว่าตัวแปร cookie ทั้ง 2 ตัวแปรเก็บค่าอะไรไว้ ดังนี้



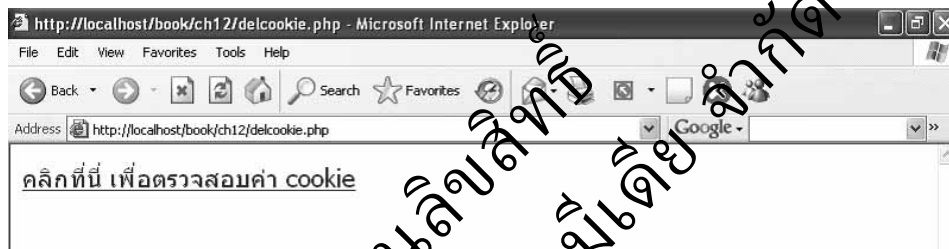
รูปที่ 10.10 ผลลัพธ์ของโปรแกรม showcookie.php

จากรูปที่ 10.10 จะเห็นว่าตัวแปร \$username และ \$password เก็บค่า "charnchai" และ "12345" ไว้ตามลำดับ โดยมีอายุของตัวแปรทั้งสองเป็น 60 วินาที ทดสอบอายุของตัวแปรโดยคลิกที่ปุ่ม Refresh บนเบราว์เซอร์หลังจากผ่านไป 60 วินาที ตัวแปรทั้งสองจะไม่เก็บค่าใดๆ ไว้แล้ว ดังนี้



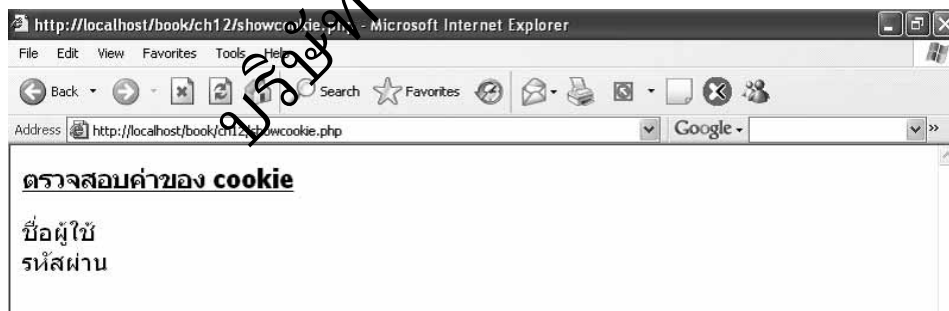
รูปที่ 10.11 ผลลัพธ์ของโปรแกรม showcookie.php เมื่อคลิกปุ่ม Refresh หลังจากผ่านไป 60 วินาที

หากต้องการให้ทำลาย Cookie ก่อนที่จะถึงเวลาที่หมดอายุ ให้ทดสอบเรียกโปรแกรม cookiemake.php ใหม่ หลังจากนั้นทำการเรียกโปรแกรม delcookie.php และ showcookie.php ตามลำดับ ผลลัพธ์ของโปรแกรมจะเป็นดังนี้ พอเราเรียกโปรแกรม delcookie.php



รูปที่ 10.12 ผลลัพธ์ของโปรแกรม delcookie.php

หลังจากนั้นให้คลิกที่ลิงค์ "คลิกที่นี่ เพื่อตรวจสอบค่า cookie" เพื่อแสดงผลว่าตัวแปร cookie ทั้ง 2 ตัวแปรไม่ได้เก็บค่าอะไรไว้แล้ว ดังนี้



รูปที่ 10.13 ผลลัพธ์ของโปรแกรม showcookie.php

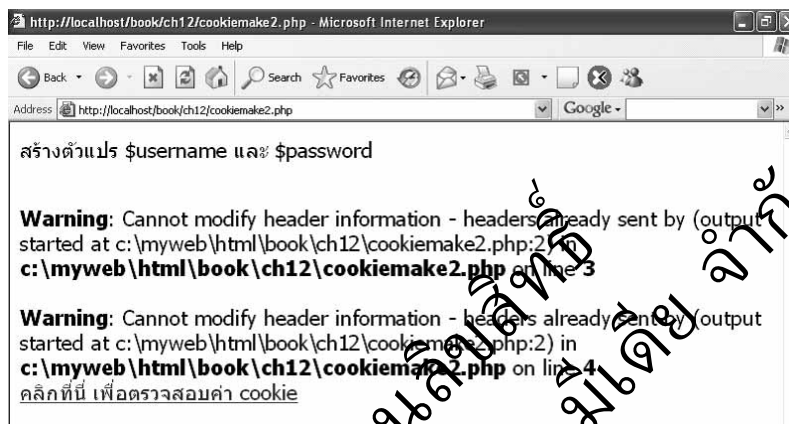
ฟังก์ชัน `ob_start()` และ `ob_end_flush()`

ข้อจำกัดที่สำคัญของฟังก์ชัน `setcookie()` คือ ห้ามใช้ฟังก์ชัน `setcookie()` นี้หลังฟังก์ชันหรือคำสั่งต่างๆ ของ PHP มิฉะนั้นจะเกิดข้อผิดพลาดขึ้น ทดสอบข้อผิดพลาดดังกล่าวโดยแก้ไขโปรแกรม `cookiemake.php` และบันทึกเปลี่ยนชื่อไฟล์เป็น `cookiemake2.php` ดังนี้

cookiemake2.php

```
<?
echo "สร้างตัวแปร \$username และ \$password <p>";
setcookie("username", "charnchai", time()+60);
setcookie("password", "12345", time()+60);
echo "<a href=showcookie.php>คลิกที่นี่ เพื่อตรวจสอบค่า cookie</a>";
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็น ดังนี้



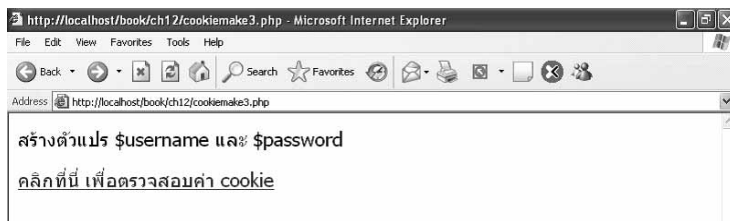
รูปที่ 10.14 ผลลัพธ์ของโปรแกรม cookiemake2.php

เหตุที่เกิดข้อผิดพลาดเช่นนี้ เนื่องมาจากที่คอมพิวเตอร์-ส่งระหว่างเว็บเซิร์ฟเวอร์ และเว็บเบราว์เซอร์เป็นข้อมูลในส่วนของเฮดเดอร์ของโปรโตคอล HTTP เกิดความผิดพลาดที่เกิดขึ้น (Cannot modify header information) ไม่อนุญาตให้ทำการเปลี่ยนแปลงแก้ไขในส่วนของเฮดเดอร์ ดังนั้นวิธีการแก้ปัญหาเราจึงต้องทำการเก็บข้อมูลพักไว้ในหน่วยความจำชั่วคราว (buffer) ก่อน แล้วค่อยส่งไปที่เดียวในตอนท้าย โปรแกรมภาษา PHP ได้จัดเตรียมฟังก์ชัน ob_start() เพื่อใช้เปิดการใช้งานหน่วยความจำชั่วคราว และฟังก์ชัน ob_end_flush() เพื่อใช้ปิดการใช้งานของหน่วยความจำชั่วคราว ดังนั้นจากโปรแกรม cookiemake2.php ในหัวข้อที่ผ่านมาเราจะทำการปรับปรุงแก้ไข และบันทึกเปลี่ยนชื่อไฟล์เป็น cookiemake3.php ดังนี้

cookiemake3.php

```
<?
ob_start();
echo "สร้างตัวแปร \$username และ \$password <p>";
setcookie("username", "charnchai", time()+60);
setcookie("password", "12345", time()+60);
echo "<a href=showcookie.php>คลิกที่นี่ เพื่อตรวจสอบค่า cookie</a>";
ob_end_flush();
?>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเห็นว่าไม่มีข้อผิดพลาดเกิดขึ้น ดังนี้



รูปที่ 10.15 ผลลัพธ์ของโปรแกรม cookie3.php

10.4 การประยุกต์ Session และ Cookie ในโปรแกรม Log in

ในหัวข้อนี้จะยกตัวอย่างการใช้ฟังก์ชัน Session และ Cookie ในโปรแกรม Log in ก่อนอื่นพิจารณาโปรแกรม Log in ง่ายๆ ต่อไปนี้

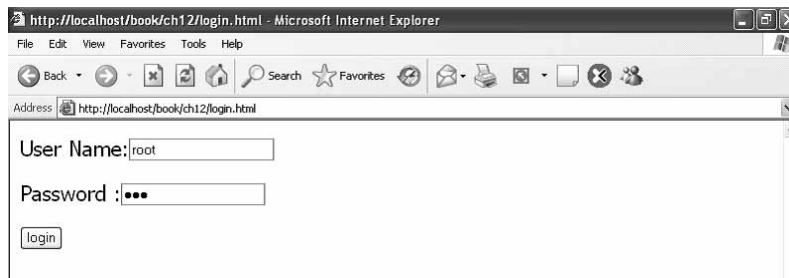
login.html

login.php

login.html

```
<html>
<body>
  <form method=post action=login.php>
    User Name:<input type=text name=username><p>
    Password :<input type=password name=password><p>
    <input type=submit value=login>
  </form>
</body>
</html>
```

ผลลัพธ์ของโปรแกรมข้างต้นจะเป็นฟอร์มรับข้อมูลของ User name และ Password เพื่อเข้าสู่การใช้งานภายในระบบ ดังนี้



รูปที่ 10.16 ผลลัพธ์ของโปรแกรม login.html

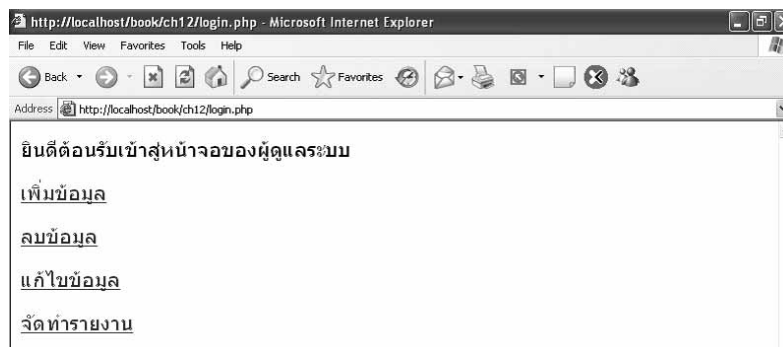
หลังจากกรอก User name และ Password ที่ถูกต้อง (ในที่นี้กำหนดเป็น root และ 123 ตามลำดับ) และคลิกที่ปุ่ม login โปรแกรมจะนำส่งค่าของ username และ password ไปยังโปรแกรม login.php ซึ่งมีไฟล์โปรแกรม ดังนี้

```

login.php
<?
    if ($username=="root" && $password=="123")
    {
        echo "include(adminform.html)";
    }
    else
    {
        echo "Error Login";
    }
?>

```

ผลลัพธ์ของโปรแกรมข้างต้น เมื่อกรอก username และ password ถูกต้องจะเป็นดังรูปที่ 11.17 ดังนี้ และโปรแกรมจะเรียกไฟล์ adminform.html ซึ่งเปรียบเสมือนหน้าจอของผู้ดูแลระบบ (ในส่วนนี้ใช้คำสั่ง include ซึ่งเป็นการเรียกไฟล์อื่นเข้ามาประมวลผล)



รูปที่ 10.17 ผลลัพธ์ของโปรแกรม login.php

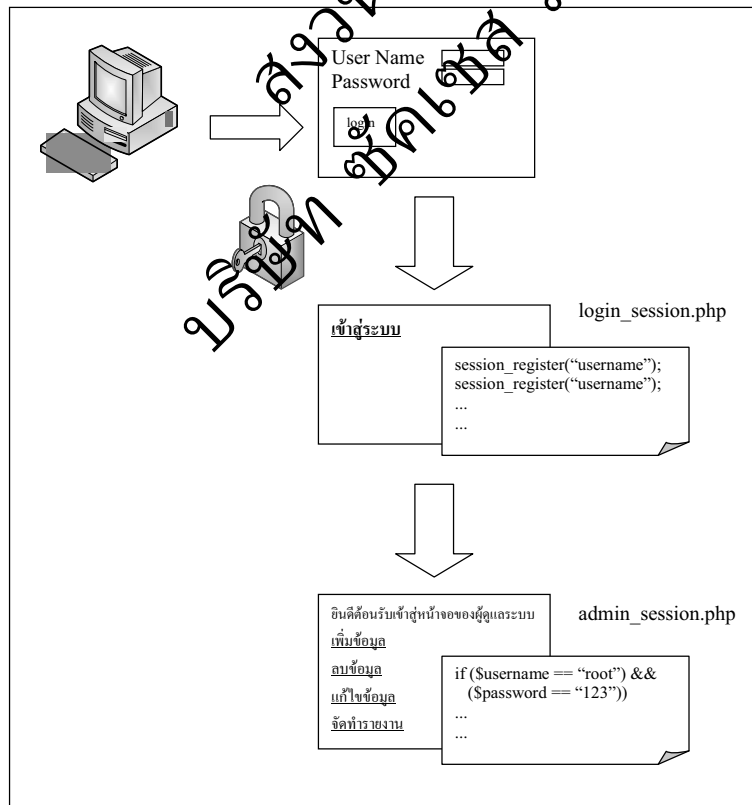
adminform.html

```
<html>
<body>
  ยินดีต้อนรับเข้าสู่หน้าจอของผู้ดูแลระบบ<p>
  <a href="insert.php">เพิ่มข้อมูล</a><p>
  <a href="delete.php">ลบข้อมูล</a><p>
  <a href="updateload.php">แก้ไขข้อมูล</a><p>
  <a href="select.php">จัดทำรายงาน</a><p>
</body>
</html>
```

จากโปรแกรมข้างต้นนี้มีส่วนเสีย คือ ถ้าเราทำการปิดบราวเซอร์นี้ และเปิดบราวเซอร์ใหม่ขึ้นมา หลังจากนั้น ก็พิมพ์ URL ในช่อง Address ของเว็บบราวเซอร์เป็น http://localhost/adminform.html เราจะสามารถ เข้าถึงหน้าจอของผู้ดูแลระบบได้ โดยที่ไม่ต้องผ่านระบบ Log in เลย ซึ่งถือว่าเป็นช่องโหว่ที่สามารถเจาะเข้าไปใช้ งานภายในระบบได้ วิธีแก้ปัญหานี้ เราสามารถใช้ session และ cookie มาช่วยได้ ดังนี้

ปรับโปรแกรม Log in โดยใช้ Session

พิจารณารูปต่อไปนี้



รูปแสดงโปรแกรม Login โดยใช้ Session

จากรูปข้างต้น มีหลักการทำงานดังนี้

1. สร้างฟอร์มรับ username และ password (login_session.html)
2. กรอก username และ password ให้ถูกต้องหลังจากนั้นทำการคลิกที่ปุ่ม login
3. หลังจากคลิกที่ปุ่ม login โปรแกรมจะส่ง username และ password ไปที่โปรแกรมเพื่อตรวจสอบความถูกต้องของ username และ password (login_session.php)
4. ที่โปรแกรม login_session.php มีการลงทะเบียนตัวแปร session 2 ตัวแปร คือ \$username และ \$password และยังมีการสร้างลิงค์เพื่อคลิกเข้าสู่หน้าจอของผู้ดูแลระบบ (admin_session.php)
5. ที่โปรแกรม admin_session.php มีการตรวจสอบค่าของตัวแปร session อีกครั้ง ดังนั้นถ้าผู้ใช้ทำการ copy URL ของไฟล์นี้ไว้ และเปิดบราวเซอร์ใหม่ขึ้นมาทำการ paste URL ก็จะไม่สามารถเข้าถึงหน้าจอของผู้ดูแลระบบได้

login_session.html

```
<html>
<body>
  <form method=post action=login_session.php>
    User Name:<input type=text name=username><p>
    Password :<input type=password name=password><p>
    <input type=submit value=login>
  </form>
</body>
</html>
```

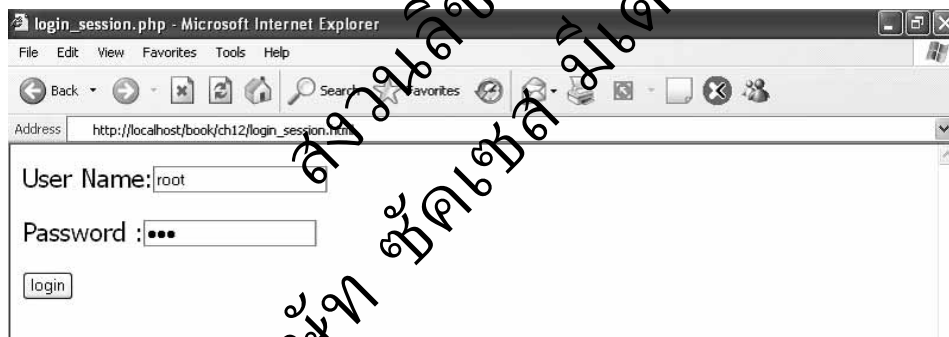
login_session.php

```
<?
  if (($username == "root") && ($password == "123"))
  {
    session_register("username");
    session_register("password");
    echo "<a href=admin_session.php>เข้าสู่ระบบ </a>";
  }
  else
  {
    echo "Error Login";
  }
?>
```


admin_session.php

```
<?
if (($username=="root") && ($password=="123"))
{
    echo "ยินดีต้อนรับเข้าสู่หน้าจอของผู้ดูแลระบบ<p>";
    echo "<a href=insert.php>เพิ่มข้อมูล</a><p>";
    echo "<a href=delete.php>ลบข้อมูล</a><p>";
    echo "<a href=updateload.php>แก้ไขข้อมูล</a><p>";
    echo "<a href=select.php>จัดทำรายงาน</a><p>";
}
else
{
    echo 'Invalid to entry the Admin page';
}
?>
```

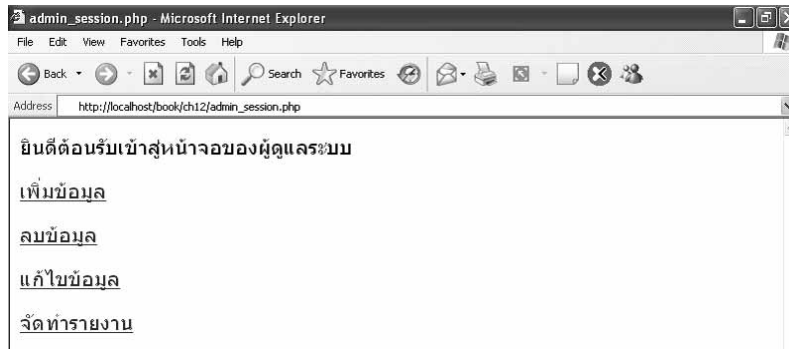
ผลลัพธ์ของโปรแกรมทั้งหมดเป็น ดังนี้



รูปที่ 10.18 ผลลัพธ์ของโปรแกรม login_session.html



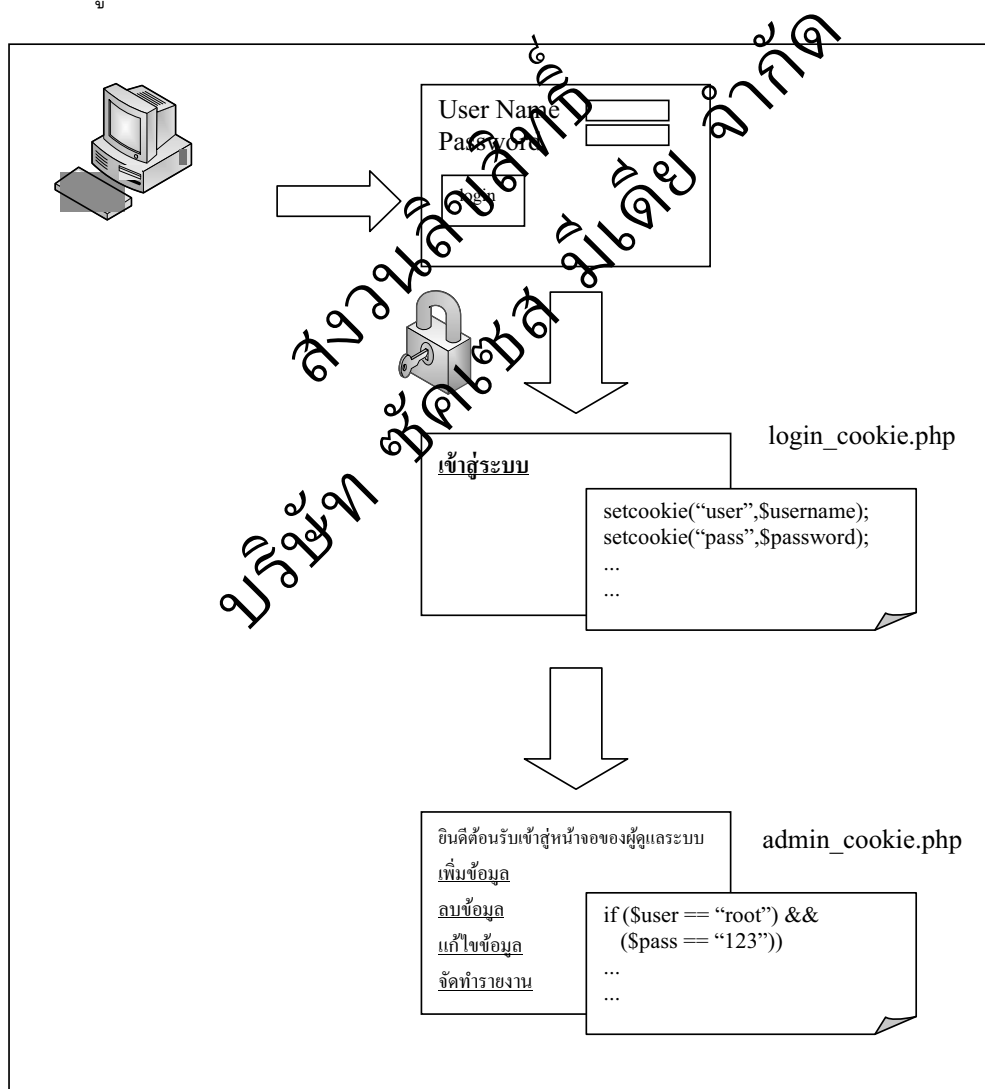
รูปที่ 10.19 ผลลัพธ์ของโปรแกรม login_session.php



รูปที่ 10.20 ผลลัพธ์ของโปรแกรม admin_session.php

ปรับโปรแกรม Log in โดยใช้ Cookie

พิจารณารูปต่อไปนี้



รูปแสดงโปรแกรม Login โดยใช้ cookie

จากรูปข้างต้น มีหลักการทำงานดังนี้

1. สร้างฟอร์มรับ username และ password (login_cookie.html)
2. กรอก username และ password ให้ถูกต้องหลังจากนั้นทำการคลิกที่ปุ่ม login
3. หลังจากคลิกที่ปุ่ม login โปรแกรมจะส่ง username และ password ไปที่โปรแกรมเพื่อตรวจสอบความถูกต้องของ username และ password (login_cookie.php)
4. ที่โปรแกรม login_cookie.php มีการสร้างตัวแปร cookie คือ \$user และ \$pass มีอายุ 1 ชั่วโมงและมีการสร้างลิงค์เพื่อคลิกเข้าสู่หน้าจอของผู้ดูแลระบบ (admin_cookie.php)
5. ที่โปรแกรม admin_cookie.php มีการตรวจสอบค่าของตัวแปร cookie อีกครั้ง ดังนั้นถ้าผู้ใช้ทำการคลิกที่ลิงค์ "ออกจากระบบ" (logout.php) และทดลองเข้าถึงไฟล์ admin_cookie.php ใหม่โดยไม่ต้องผ่านระบบ Log in ก็จะไม่สามารถเข้าถึงหน้าจอของผู้ดูแลระบบได้
6. ที่โปรแกรม logout.php มีการทำลายตัวแปร cookie ทั้ง 2 ตัวแปร

login_cookie.html

```
<html>
<body>
  <form method=post action=login_cookie.php>
    User Name:<input type=text name=username><p>
    Password :<input type=password name=password><p>
    <input type=submit value=login>
  </form>
</body>
</html>
```

login_cookie.php

```
<?
  if (($username == "root") && ($password == "123"))
  {
    setcookie("user",$username,time()+3600);
    setcookie("pass", "$password",time()+3600);
    echo "<a href=admin_cookie.php>เข้าสู่ระบบ</a>";
  }
  else
  {
    echo "Error Login";
  }
?>
```

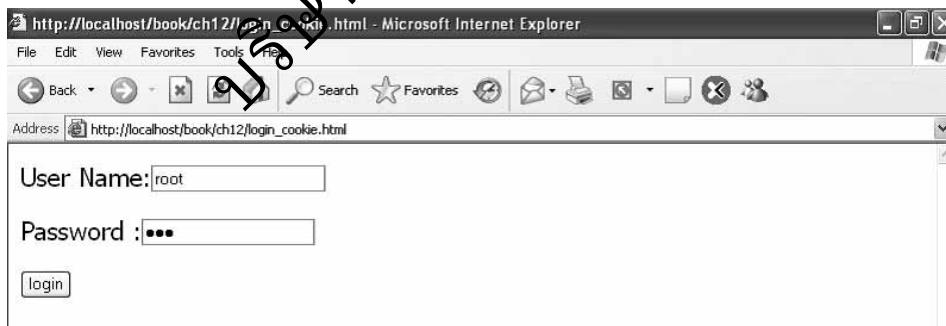
admin_cookie.php

```
<?
if (($user == "root") && ($pass == "123"))
{
    echo "ยินดีต้อนรับเข้าสู่หน้าจอของผู้ดูแลระบบ<p>";
    echo "<a href=insert.php>เพิ่มข้อมูล</a><p>";
    echo "<a href=delete.php>ลบข้อมูล</a><p>";
    echo "<a href=updateload.php>แก้ไขข้อมูล</a><p>";
    echo "<a href=select.php>จัดทำรายงาน</a><p>";
    echo "<a href=logout.php>ออกจากระบบ</a><p>";

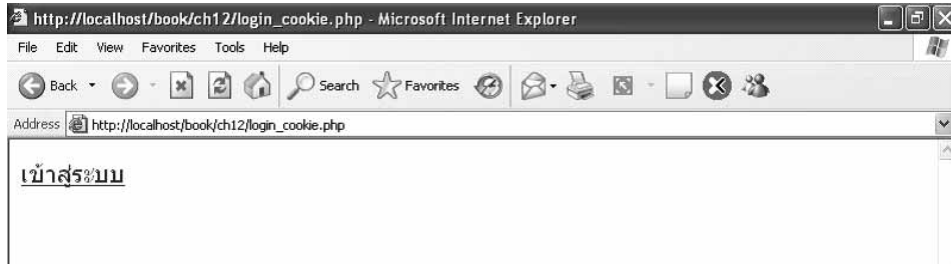
}
else
{
    echo 'Invalid to entry the Admin page';
}
?>
```

logout.php

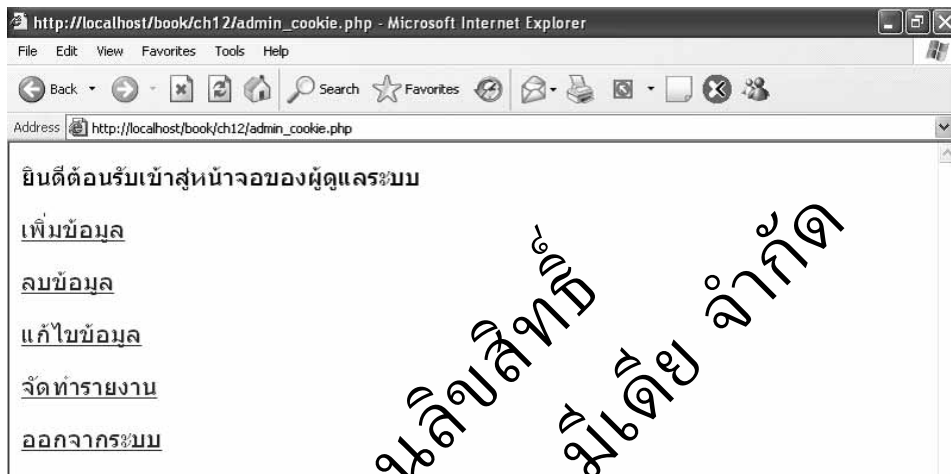
```
<?
setcookie("user");
setcookie("pass");
echo "ออกจากระบบเรียบร้อยแล้ว";
?>
```



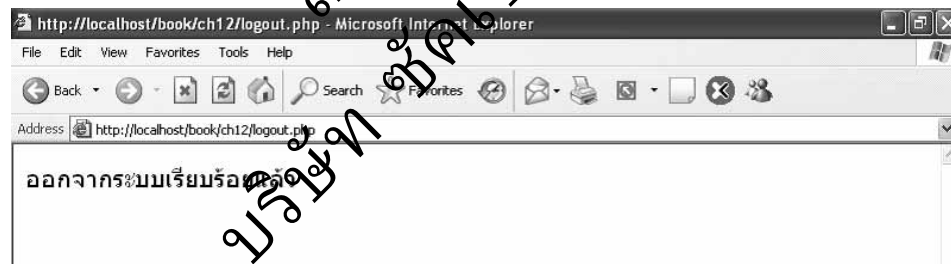
รูปที่ 10.21 ผลลัพธ์ของโปรแกรม login_cookie.html



รูปที่ 10.22 ผลลัพธ์ของโปรแกรม login_cookie.php



รูปที่ 10.23 ผลลัพธ์ของโปรแกรม admin_cookie.php



รูปที่ 10.24 ผลลัพธ์ของโปรแกรม logout.php

สงวนลิขสิทธิ์
บริษัท ชัยเชษฐ์ มีเดีย จำกัด

สรุป

Session มีคุณสมบัติใช้ส่งค่าของตัวแปรที่สร้างขึ้นจากไฟล์หนึ่งไปแสดงผลยังอีกไฟล์ได้ ทำให้เราสามารถทำการติดตามและตรวจสอบผู้ใช้ได้ ในการใช้งาน Session โปรแกรมภาษา PHP ได้จัดเตรียมฟังก์ชันให้ใช้งานอยู่หลายฟังก์ชัน เช่น `session_start()`, `session_register()`, `session_unregister()`, `session_destroy()`

Cookie คือข้อมูลที่เราส่งไปเก็บไว้ในเครื่องของผู้ใช้ เพื่อประโยชน์ในการตรวจสอบ และติดตามการทำงานของผู้ใช้ โดยการทำงานบราวเซอร์จะทำการส่งข้อมูลของผู้ใช้ที่จัดเก็บไว้ในเครื่องมายังเว็บเซิร์ฟเวอร์ และจะนำ Cookie ดังกล่าวมาเก็บไว้ในตัวแปรทำให้สามารถตรวจสอบค่าต่างๆ ของผู้ใช้ได้ ฟังก์ชันที่สำคัญในการจัดการ cookie คือ `setcookie()`, `ob_start()`, `ob_end_flush()`

ในบทนี้เราได้นำ Session และ Cookie มาประยุกต์ทำงานร่วมกับโปรแกรม Log in เพื่อเพิ่มความปลอดภัยในการเข้าถึงข้อมูล เนื่องจากโปรแกรม Log in ที่ไม่ได้ใช้ Session หรือ Cookie ผู้ใช้งานจะสามารถเข้าถึงหน้าเว็บที่เป็นความลับ หรือหน้าเว็บเฉพาะของผู้ดูแลระบบเท่านั้น โดยการ copy URL และวาง URL นั้นยังช่อง Address ของเว็บเบราว์เซอร์ก็จะสามารถเข้าถึงหน้าเว็บดังกล่าวได้เลย

สงวนลิขสิทธิ์
บริษัท ชัคเซส มีเดีย จำกัด

แบบฝึกหัดบทที่ 10

1. จงอธิบายลักษณะการทำงาน และประโยชน์ของ Session

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. จงอธิบายลักษณะการทำงาน และประโยชน์ของ Cookie

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. จากแบบฝึกหัดบทที่ 9 ในการสร้างแบบฟอร์มรับสมัครงาน ให้เขียนโปรแกรมสร้างตัวแปร session เก็บข้อมูลทั้งหมด, โปรแกรมแสดงข้อมูลที่เก็บในตัวแปร session ทั้งหมด และโปรแกรมทำลายตัวแปร session ทั้งหมด

4. จากแบบฝึกหัดบทที่ 9 ในการสร้างแบบฟอร์มการจองตั๋วออนไลน์ ให้เขียนโปรแกรมสร้างตัวแปร cookie เก็บข้อมูลทั้งหมด, โปรแกรมแสดงข้อมูลที่เก็บในตัวแปร cookie ทั้งหมด และโปรแกรมทำลายตัวแปร cookie ทั้งหมด

สงวนลิขสิทธิ์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

5. เราสามารถนำหลักการของ session และ cookie ไปประยุกต์ใช้ในโปรแกรมอะไรได้บ้าง ให้ยกตัวอย่างมา 3 โปรแกรม พร้อมทั้งอธิบายการทำงานด้วย

.....

.....

.....

.....

.....

.....

.....

.....

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

ฐานข้อมูล MySQL

ฐานข้อมูลเป็นองค์ประกอบหนึ่งที่สำคัญในการพัฒนาแอปพลิเคชัน (Application) เนื่องจากระบบงานต่างๆ จะมีการจัดเก็บข้อมูลไว้ในฐานข้อมูล เพื่อให้สามารถเข้าถึงข้อมูลที่ฐานข้อมูลได้ คำว่า "เข้าถึงข้อมูล" ในที่นี้ก็คือ การที่ระบบงานสามารถเพิ่ม, ลบ, แก้ไขข้อมูล และจัดทำรายงานได้ ซึ่งถือว่าเป็นฟังก์ชันพื้นฐานของระบบงานต่างๆ เช่น ระบบงานทางธุรกิจ เราจัดเก็บข้อมูลของสินค้าคงคลังเพื่อเก็บรายละเอียดสินค้าและสต็อกสินค้าของบริษัท ถ้ามีลูกค้ามาสั่งซื้อสินค้าเราก็สามารถทำการตรวจสอบว่ามีสินค้าที่ลูกค้าต้องการหรือไม่ หลังจากนั้นเมื่อลูกค้าซื้อสินค้าแล้วเราก็สามารถทำการจัดส่งหรือปรับปรุงจำนวนสินค้าที่ขายไปได้ เป็นต้น ในบทนี้จะเริ่มตั้งแต่ขั้นตอนการติดตั้ง MySQL และวิธีการใช้งาน

จุดประสงค์
ทั่วไป

1. เพื่อให้เรียนรู้วิธีการติดตั้งโปรแกรมฐานข้อมูล MySQL
2. เพื่อศึกษาการใช้งาน MySQL
3. เพื่อศึกษาคำสั่งในการแสดงรายชื่อฐานข้อมูล เลือกใช้ฐานข้อมูล แสดงรายชื่อตาราง และแสดงรายละเอียดของฟิลด์
4. เพื่อศึกษาคำสั่ง SQL
5. เพื่อศึกษาคำสั่งในการออกจากฐานข้อมูล

จุดประสงค์
เชิงพฤติกรรม

1. สามารถติดตั้งโปรแกรมฐานข้อมูล MySQL
2. สามารถใช้งาน MySQL ได้
3. สามารถใช้คำสั่ง SQL เพื่อจัดการฐานข้อมูลได้

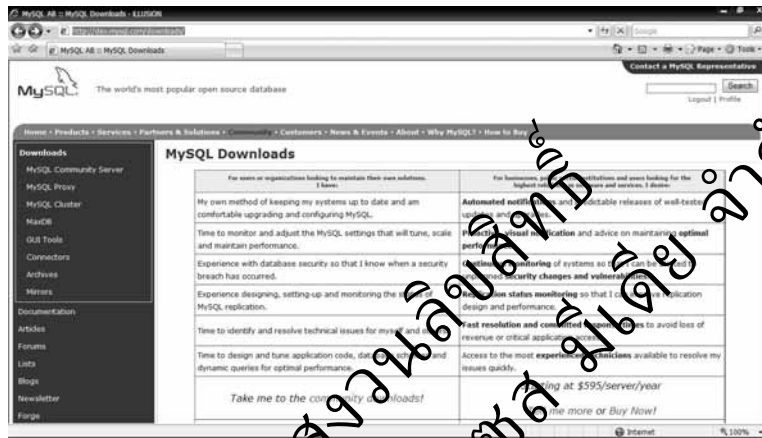
เนื้อหาสาระ

1. การติดตั้ง MySQL และการปรับแต่งค่าคอนฟิก
2. การใช้งาน MySQL

MySQL ถูกพัฒนามาจากโปรแกรม mSQL ซึ่งมีจุดด้อยและข้อจำกัดอยู่มาก โดยทางผู้พัฒนาโปรแกรม MySQL ได้ทำการแก้ไขข้อบกพร่อง ข้อจำกัดต่างๆ ที่มีอยู่มากมายนี้ ให้มีประสิทธิภาพในการทำงานมากยิ่งขึ้นจนกลายเป็นตัวโปรแกรม MySQL และสร้างให้ตัวโปรแกรมสามารถสนับสนุนการทำงานระบบปฏิบัติการทั้งระบบปฏิบัติการ Windows หรือ Linux และที่เด่นไปกว่านั้น ตัวโปรแกรม MySQL ยังเป็นของแจกฟรี เพราะเป็นโปรแกรมประเภท Open Source ซึ่งมีลิขสิทธิ์ในแบบ GPL

11.1 การติดตั้ง MySQL และการปรับแต่งค่าคอนฟิก

1. ดาวน์โหลด MySQL ได้จากเว็บไซต์ <http://dev.mysql.com/downloads/>



รูปที่ 11.1 หน้าจอเว็บไซต์ดาวน์โหลดโปรแกรม MySQL

2. ไฟล์ที่ดาวน์โหลดได้เป็นไฟล์ Zip ดังนั้นให้ทำการแตกไฟล์ด้วย Winzip หรือ WinRAR จะได้ไฟล์ Setup.exe ให้ทำการดับเบิลคลิกที่ไฟล์นี้ก็จะเริ่มเข้าสู่ขั้นตอนของการติดตั้งโดยเริ่มจากหน้าจอยินดีต้อนรับของ MySQL ให้คลิกปุ่ม **Next**



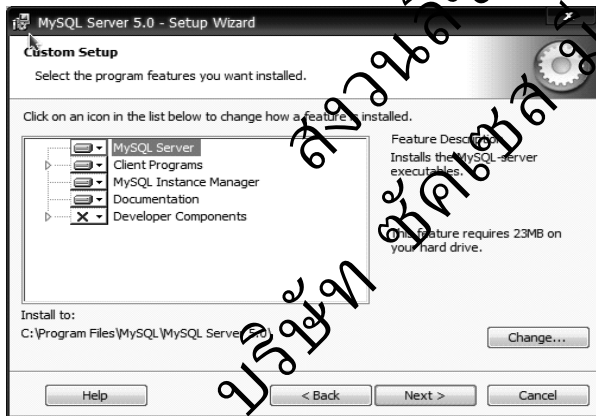
รูปที่ 11.2 หน้าจอเริ่มต้นขั้นตอนรับโปรแกรม MySQL

3. เข้าสู่หน้าจอของการเลือกประเภทของการติดตั้ง ให้เลือกแบบ Custom แล้วคลิกที่ปุ่ม Next



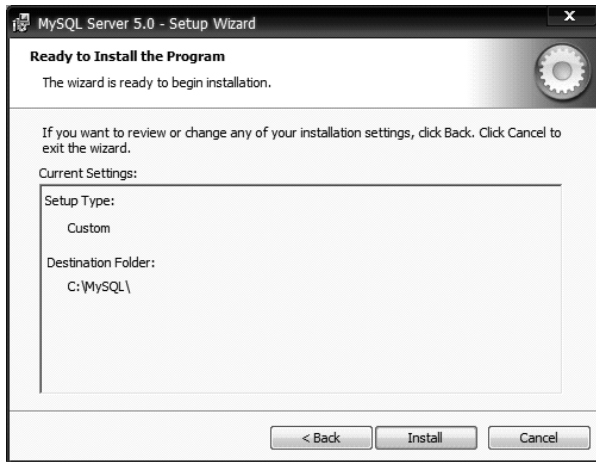
รูปที่ 11.3 หน้าจอแสดงการเลือกประเภทของการติดตั้ง

4. เข้าสู่หน้าจอของการเลือกไดรฟ์และโฟลเดอร์ที่ต้องการติดตั้ง ให้ทำการเปลี่ยนจาก C:\Program Files\MySQL Server 5.0\ เป็น C:\MySQL\ โดยคลิกเปลี่ยนที่ปุ่ม Change... หลังจากนั้นให้คลิกที่ปุ่ม Next



รูปที่ 11.4 หน้าจอแสดงการเลือกไดรฟ์และโฟลเดอร์ในการติดตั้ง

5. เข้าสู่หน้าจอเตรียมความพร้อมก่อนเริ่มต้นทำการติดตั้ง โดยถ้าต้องการเปลี่ยนแปลงใดๆ ให้คลิกที่ปุ่ม Back แต่ถ้าพร้อมที่จะติดตั้งก็ให้คลิกปุ่ม Install



รูปที่ 11.5 หน้าจอเตรียมความพร้อมก่อนเริ่มการติดตั้ง

6. โปรแกรมจะแสดงความคืบหน้าในการติดตั้ง



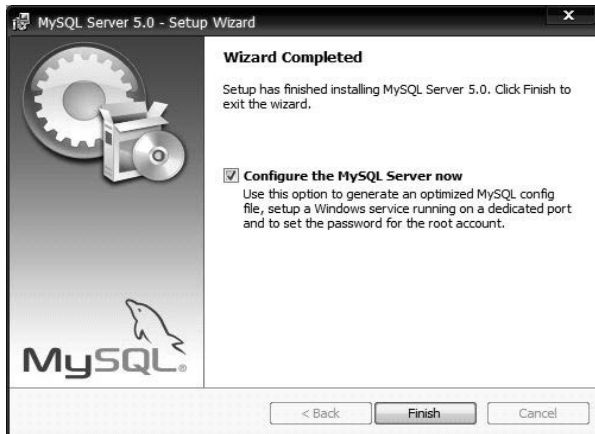
รูปที่ 11.6 หน้าจอแสดงความคืบหน้าในการติดตั้ง

7. เข้าสู่หน้าจอการ Sign-Up แต่ในที่นี้ให้คลิกเลือก Skip Sign-up เพื่อข้ามขั้นตอนนี้ไปก่อนเสร็จแล้วคลิกปุ่ม Next



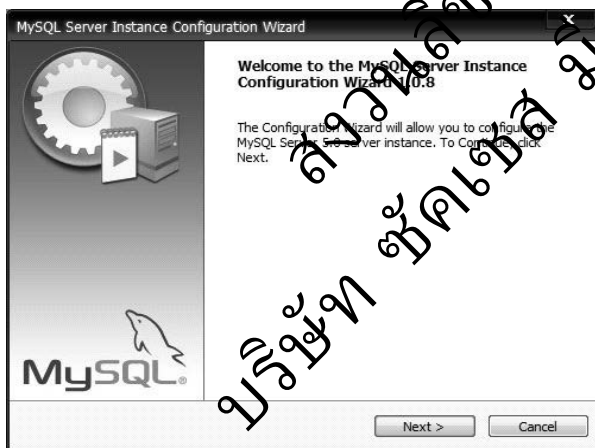
รูปที่ 11.7 หน้าจอแสดงการ Sign-Up

8. คลิกให้มีเครื่องหมายถูกที่ **Configure the MySQL Server now** เพื่อเข้าสู่การปรับแต่งคอนฟิกเบื้องต้น เสร็จแล้วคลิกที่ปุ่ม **Finish**



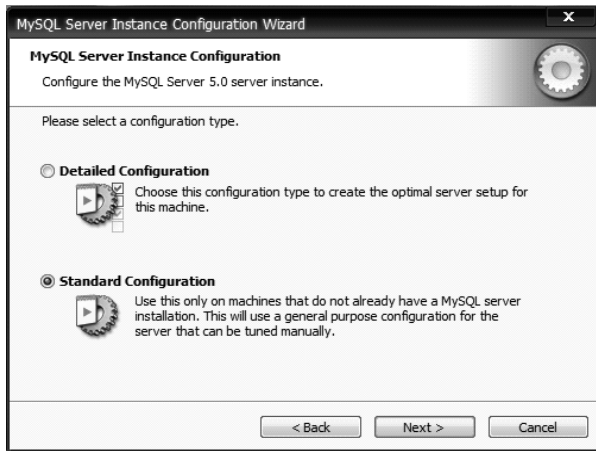
รูปที่ 11.8 หน้าจอแสดง Wizard ของการปรับแต่งคอนฟิก

9. ระบบจะนำเข้าสู่หน้าจอของการปรับแต่งค่าคอนฟิก MySQL ให้คลิกที่ปุ่ม **Next>**



รูปที่ 11.9 หน้าจอเริ่มต้นยินดีต้อนรับสู่การปรับแต่งค่าคอนฟิกของ MySQL

10. เข้าสู่หน้าจอการเลือกประเภทของการปรับแต่งค่าคอนฟิก โดยให้คลิกเลือกประเภทของการปรับแต่งค่าคอนฟิกเป็น **Standard Configuration** เสร็จแล้วคลิกปุ่ม **Next**



รูปที่ 11.10 หน้าจอแสดงการเลือกประเภทของการปรับแต่งค่าคอนฟิก

11. ที่ Service Name: ให้เลือกเวอร์ชันของ MySQL ที่ใช้งานเป็น MySQL5 และคลิกเลือกที่ Include Bin Directory in Windows PATH เพื่อให้สามารถเรียกใช้งานผ่าน Command Line เสร็จแล้วคลิกปุ่ม Next



รูปที่ 11.11 หน้าจอแสดงการเลือกเวอร์ชันของ MySQL

12. กำหนดรหัสผ่าน root ที่ต้องการโดยพิมพ์รหัสที่ต้องการลงในช่อง New root password: และช่อง Confirm: เพื่อยืนยันรหัสผ่านดังกล่าว โดยให้ทำการคลิกเลือกที่ Enable root access from remote machines (กรณีที่ต้องการใช้งาน MySQL จากเครื่องภายนอก) หลังจากนั้นให้ทำการคลิกปุ่ม Next



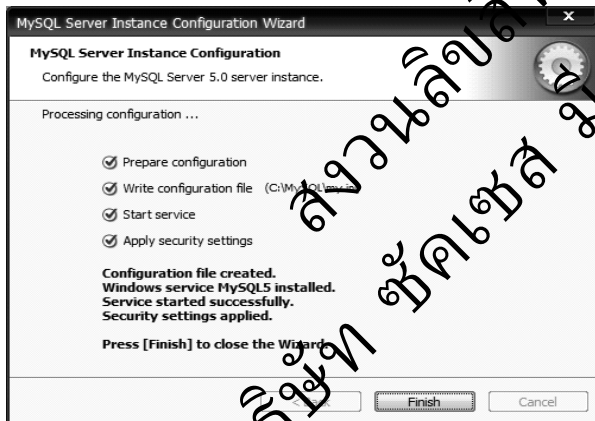
รูปที่ 11.12 หน้าจอแสดงการกำหนดรหัสผ่านในการเข้าใช้งาน MySQL

13. เข้าสู่หน้าจอก่อนที่จะเริ่มทำการปรับแต่งค่าคอนฟิก โดยถ้ามีการแก้ไขใดๆ ให้ทำการคลิกที่ปุ่ม Back แต่ถ้าไม่มีการแก้ไขใดๆ ก็ให้ทำการคลิกที่ปุ่ม Execute เพื่อเริ่มต้นการปรับแต่งค่าคอนฟิก



รูปที่ 11.13 หน้าจอก่อนเริ่มต้นการปรับแต่งค่าคอนฟิก

14. เมื่อโปรแกรมทำการปรับแต่งค่าคอนฟิกเสร็จแล้ว ให้ทำการคลิกที่ปุ่ม Finish เป็นอันเสร็จขั้นตอนของการติดตั้ง และปรับแต่งค่าคอนฟิกของ MySQL

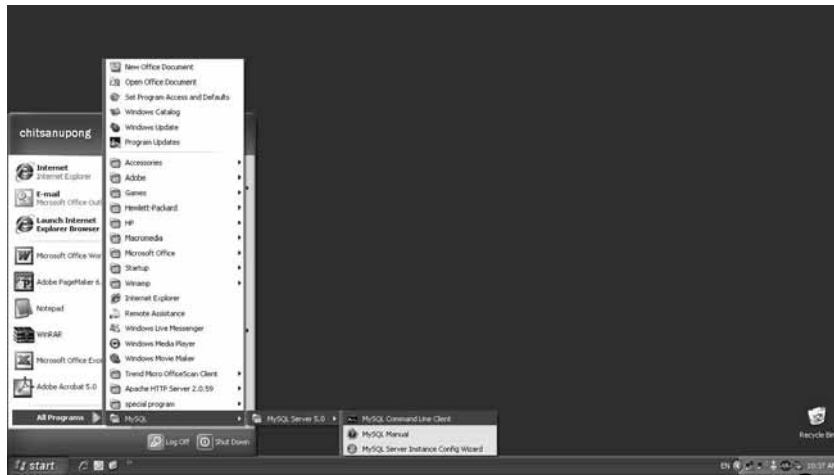


รูปที่ 11.14 หน้าจอแสดงหลังการปรับแต่งค่าคอนฟิกของ MySQL เสร็จสมบูรณ์

11.2 การใช้งาน MySQL

หลังจากทำการติดตั้ง MySQL เสร็จสมบูรณ์แล้วตามหัวข้อ 11.1 เราสามารถเข้าไปใช้งาน MySQL ได้ ดังนี้

1. ให้คลิกเลือก Start>MySQL>MySQL Server 5.0>MySQL Command Line Client



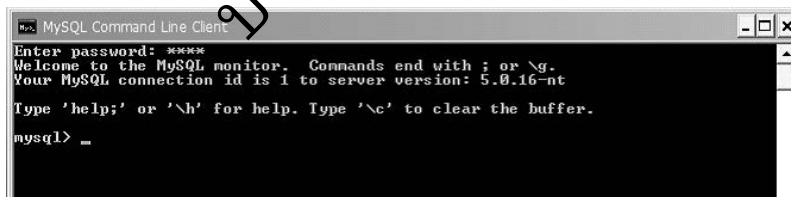
รูปที่ 11.15 หน้าจอแสดงการเข้าสู่การใช้งาน MySQL

2. จะปรากฏหน้าจอของ MySQL Command Line Client โดยให้ทำการพิมพ์รหัสผ่านตามที่เคยกำหนดในขั้นตอนการติดตั้ง MySQL เสร็จแล้วกด <Enter>



รูปที่ 11.16 หน้าจอการป้อนรหัสผ่านเข้าใช้งาน MySQL

3. จะเข้าสู่หน้าจอการใช้งาน MySQL โดยสังเกตที่ Mysql>



รูปที่ 11.17 หน้าจอเข้าสู่โปรแกรม MySQL

4. ในกรณีที่ต้องการกำหนดรหัสผ่านใหม่ (ในที่นี้กำหนดรหัสผ่านคือ 1234) ให้ทำการพิมพ์คำสั่งต่อไปนี้

```
mysql>update mysql.user set password=password('1234') where user ='root'; (กด<Enter>)  
mysql>flush privileges; (กด<Enter>)
```


หลังจากนั้นถ้าจะเข้าไปใช้งาน MySQL ครั้งต่อไปก็ให้ใช้รหัสผ่านที่เปลี่ยนแปลงใหม่นี้

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p

mysql> update mysql.user set password=password('1234') where user='root';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 2 Changed: 0 Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> _
    
```

รูปที่ 11.18 หน้าจอแสดงคำสั่งการเปลี่ยนแปลงรหัสผ่านของ MySQL

ในการใช้งานฐานข้อมูล MySQL จำเป็นที่จะต้องรู้จักคำสั่งพื้นฐานในการเข้าไปดูรายละเอียดของฐานข้อมูล และคำสั่งเพื่อดำเนินการกับฐานข้อมูล ดังตารางสรุปต่อไปนี้

คำสั่ง	คำอธิบาย
Show databases;	แสดงรายชื่อฐานข้อมูลทั้งหมด
Use ชื่อฐานข้อมูล;	เลือกฐานข้อมูลที่จะทำงานด้วย
Show tables;	แสดงชื่อของตารางทั้งหมดภายในฐานข้อมูล
Show column from ชื่อตาราง;	แสดงชื่อฟิลด์และรายละเอียดของตาราง
Create Database ชื่อฐานข้อมูล;	สร้างฐานข้อมูลใหม่
Create Table (ชื่อฟิลด์ และรายละเอียดตาราง);	สร้างตารางข้อมูลใหม่
Insert into ชื่อตาราง(ชื่อฟิลด์) values (ค่าของฟิลด์);	เพิ่มข้อมูลเรคคอร์ดใหม่ลงในตาราง
Select ชื่อฟิลด์ From ชื่อตาราง (Where เงื่อนไข);	ดึงข้อมูลจากตารางที่กำหนด และตามเงื่อนไขที่ต้องการ
Update ชื่อตาราง Set ชื่อฟิลด์ = ค่าที่แก้ไข (Where เงื่อนไข);	แก้ไขข้อมูลในตารางตามเงื่อนไขที่ต้องการ
Delete From ชื่อตาราง Where เงื่อนไข;	ลบเรคคอร์ดจากตารางที่ระบุ และตามเงื่อนไขที่ต้องการ
exit	ออกจากการทำงาน MySQL

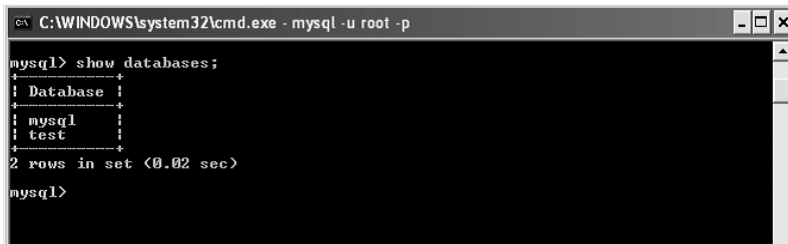
จะแสดงตัวอย่างการใช้งานของคำสั่งต่างๆ ตามลำดับในตารางข้างต้น โดยใช้ไฟล์ฐานข้อมูลที่ MySQL เตรียมไว้ให้อยู่แล้ว ดังนี้

คำสั่งแสดงรายชื่อฐานข้อมูล

รูปแบบ

```
Show Databases;
```

ทดสอบโดยพิมพ์คำสั่งนี้ที่ MySQL Prompt และกด <Enter> จะได้ผลลัพธ์ที่แสดงดังนี้



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.02 sec)
mysql>
```

รูปที่ 11.19 ตัวอย่างการใช้คำสั่ง Show Databases

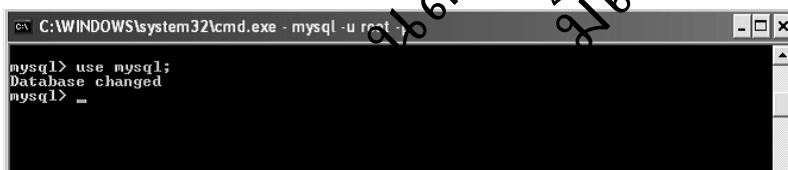
จากตัวอย่างข้างต้นผลลัพธ์จะแสดงรายชื่อฐานข้อมูลจำนวน 2 ฐานข้อมูลคือ mysql และ test

คำสั่งเลือกใช้ฐานข้อมูล

รูปแบบ

Use ชื่อฐานข้อมูล;

ทดสอบโดยพิมพ์คำสั่ง Use mysql; ที่ MySQL Prompt และกด <Enter> จะได้ผลลัพธ์ที่แสดงดังนี้



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> use mysql;
Database changed
mysql> _
```

รูปที่ 11.20 ตัวอย่างการใช้คำสั่ง Use

จากตัวอย่างข้างต้นเราได้เลือกฐานข้อมูลชื่อ mysql เพื่อที่จะเข้าไปใช้งาน ผลลัพธ์จะแสดงข้อความว่า Database changed แสดงว่าได้เข้าไปยังฐานข้อมูลนั้นแล้ว

คำสั่งแสดงรายชื่อตาราง

รูปแบบ

Show tables;

สมมติขณะนี้อยู่ในฐานข้อมูลที่ชื่อว่า mysql ให้ทดสอบโดยพิมพ์คำสั่ง show tables; ที่ MySQL Prompt และกด <Enter> จะได้ผลลัพธ์ที่แสดงดังนี้

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> use mysql;
Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| host            |
| tables_priv     |
| user            |
+-----+
6 rows in set (0.02 sec)

mysql>

```

รูปที่ 11.21 ตัวอย่างการใช้งานคำสั่ง show tables

จากตัวอย่างข้างต้นจะแสดงรายชื่อตารางต่างๆ ในฐานข้อมูล mysql ซึ่งประกอบไปด้วยตารางทั้งหมด 6 ตาราง คือ columns_priv, db, func, host, tables_priv และ user

คำสั่งแสดงรายละเอียดของฟิลด์

รูปแบบ

Show columns from ชื่อตาราง;

สมมติขณะนี้อยู่ในฐานข้อมูลที่ชื่อว่า mysql ให้ทดสอบโดยพิมพ์คำสั่ง show columns from func; ที่ MySQL Prompt และกด <Enter> จะได้ผลลัพธ์ที่แสดงดังนี้

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> show columns from func;
+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name  | char(64) binary |      | PRI |          |       |
| ret   | tinyint(1)      |      |     | 0        |       |
| dl    | char(128)       |      |     |          |       |
| type  | enum('function','aggregate') |      |     |          |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _

```

รูปที่ 11.22 ตัวอย่างการใช้งานคำสั่ง show columns

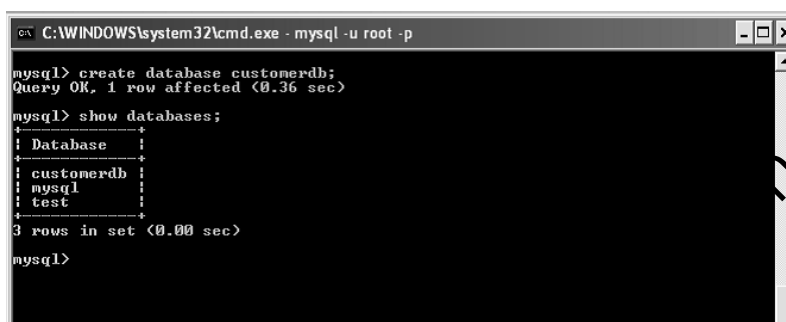
จากตัวอย่างข้างต้นเป็นการแสดงรายชื่อฟิลด์ของตาราง func โดยจะเห็นว่าประกอบไปด้วยฟิลด์ทั้งหมด 4 ฟิลด์ และรายละเอียดในแต่ละฟิลด์ก็จะแสดงชื่อฟิลด์ (Field), ชนิดของข้อมูล (Type), ค่าว่างเปล่า (Null), คีย์ (Key), ค่าปริยาย (Default) และค่าเพิ่มเติม (Extra)

คำสั่งสร้างฐานข้อมูล

รูปแบบ

Create database ชื่อฐานข้อมูล;

ทดสอบโดยพิมพ์คำสั่ง Create database customerdb; ที่ MySQL Prompt และกด <Enter> จะได้ผลลัพธ์เป็นการสร้างฐานข้อมูลชื่อ customerdb เพื่อทดสอบว่า ณ. ขณะนี้ฐานข้อมูล customerdb ได้ถูกสร้างแล้วจริงให้ทดสอบโดยการพิมพ์คำสั่ง show databases เพื่อแสดงรายชื่อฐานข้อมูล ผลลัพธ์ทั้งหมดเป็นดังนี้



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> create database customerdb;
Query OK, 1 row affected (0.36 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| customerdb |
| mysql |
| test |
+-----+
3 rows in set (0.00 sec)

mysql>
```

รูปที่ 11.23 ตัวอย่างการใช้คำสั่ง create database

คำสั่งสร้างตาราง

รูปแบบ

Create table ชื่อตาราง(รายละเอียดของ) ของตาราง);

สมมติต้องการสร้างตารางชื่อ customer โดยมีรายละเอียดของตาราง ดังนี้

Table: Customer

ชื่อฟิลด์	ชนิดข้อมูล	ความยาว	ค่าปริยาย	คีย์
Id	Int	4	0	Primary key
Name	Varchar	20	ไม่กำหนด	-
Address	Varchar	50	Bangkok	-

ดังนั้นเราสามารถเขียนคำสั่งเพื่อสร้างตารางข้อมูลได้ ดังนี้

Create table customer(id int(4), name varchar(20) not null, address varchar(50) default 'Bangkok' not null, primary key(id));

หลังจากนั้นลองทดสอบว่าตาราง customer ได้ถูกสร้างตามที่ต้องการหรือไม่ โดยพิมพ์คำสั่ง

show columns from customer;

ผลลัพธ์ของการทำงานทั้งหมดเป็นดังนี้

```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
Database changed
mysql> create table customer(id int(4), name varchar(20) not null, address varchar(50) default 'Bangkok' not null, primary key(id));
Query OK, 0 rows affected (0.08 sec)

mysql> show columns from customer;
+----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id    | int(4)       |      | PRI |          |       |
| name  | varchar(20)  |      |     |          |       |
| address | varchar(50)  |      |     | Bangkok |       |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

รูปที่ 11.24 ตัวอย่างการใช้คำสั่ง create table

คำสั่งเพิ่มข้อมูลเรคคอร์ดใหม่

รูปแบบ

Insert into ชื่อตาราง(ชื่อฟิลด์1, ชื่อฟิลด์2, ...) values(ค่าของฟิลด์1, ค่าของฟิลด์2, ...);

สมมติต้องการเพิ่มเรคคอร์ดใหม่ลงในตาราง customer ซึ่งมีรายละเอียด ดังนี้

id: 1111
name: Mr.Charnchai
address: U of Ubonratchatani

จะสามารถใช้คำสั่งเพิ่มข้อมูลเรคคอร์ดใหม่ได้ ดังนี้

Insert into customer values(1111, 'Mr.Charnchai', 'U of Ubonratchatani');

ผลลัพธ์ของการใช้คำสั่งนี้ ถ้าสามารถเพิ่มเรคคอร์ดใหม่ได้จะขึ้นข้อความว่า "Query OK" ดังตัวอย่างต่อไปนี้

```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> Insert into customer values(1111, 'Mr.Charnchai', 'U of Ubonratchatani');
Query OK, 1 row affected (0.00 sec)

mysql> _
```

รูปที่ 11.25 ตัวอย่างการใช้คำสั่ง Insert

สงวนลิขสิทธิ์
บัญชี ชัคเชษฐ์ มุณีชัย จำกัด

คำสั่งดึงข้อมูลจากตาราง

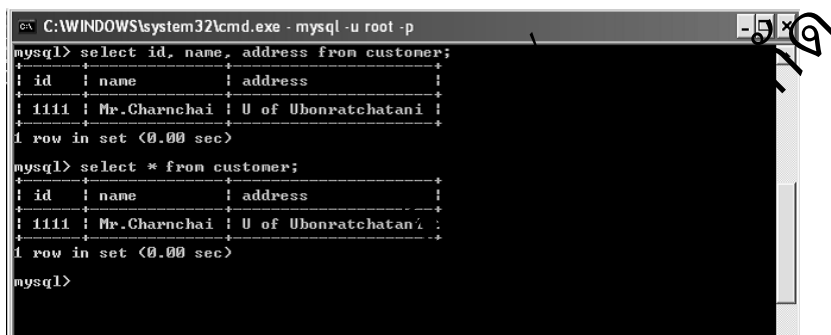
รูปแบบ

```
Select ชื่อฟิลด์1, ชื่อฟิลด์2,... From ชื่อตาราง [Where เงื่อนไข];
```

จากคำสั่งเพิ่มข้อมูลเรคคอร์ดใหม่ที่ผ่านมาเราได้ทำการเพิ่มข้อมูลเรคคอร์ดเข้าไป 1 เรคคอร์ด ดังนั้นเราจะตรวจสอบว่าภายในตาราง customer มีข้อมูลดังกล่าวจริงหรือไม่ โดยใช้คำสั่งดึงข้อมูล ดังนี้

```
Select id, name, address from customer; หรือ  
Select * form customer;
```

ผลลัพธ์จากการใช้คำสั่งข้างต้นเป็น ดังนี้



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p  
mysql> select id, name, address from customer;  
+----+-----+-----+  
| id | name | address |  
+----+-----+-----+  
| 1111 | Mr.Charnchai | U of Ubonratchatani |  
+----+-----+-----+  
1 row in set (0.00 sec)  
mysql> select * from customer;  
+----+-----+-----+  
| id | name | address |  
+----+-----+-----+  
| 1111 | Mr.Charnchai | U of Ubonratchatani |  
+----+-----+-----+  
1 row in set (0.00 sec)  
mysql>
```

รูปที่ 11.26 ตัวอย่างการใช้คำสั่ง select

จากตัวอย่างข้างต้นเป็นการใช้คำสั่งดึงข้อมูลจากตารางแบบไม่มีเงื่อนไข คือจะดึงข้อมูลหมดทุกฟิลด์ที่มีอยู่ในตาราง แต่ในบางครั้งเราอาจต้องการข้อมูลเพียงบางเรคคอร์ดเท่านั้น เช่นต้องการข้อมูลของลูกค้าที่มีรหัสลูกค้า คือ 2222 เท่านั้น เราจะสามารถเขียนคำสั่งได้ ดังนี้

```
Select * from customer where id = 2222;
```

คำสั่งแก้ไขข้อมูลในตาราง

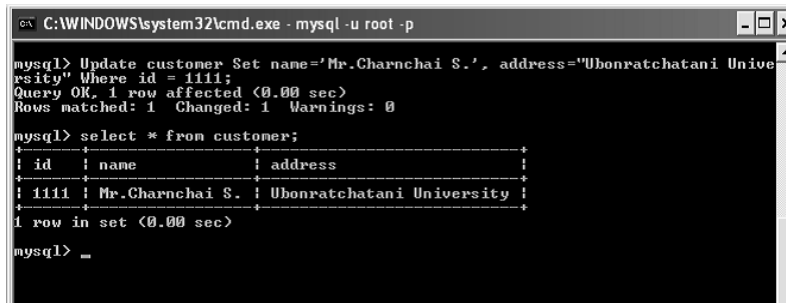
รูปแบบ

```
Update ชื่อตาราง Set ชื่อฟิลด์1 = ค่าที่แก้ไข1, ชื่อฟิลด์2 = ค่าที่แก้ไข2, ... Where เงื่อนไข;
```

สมมติต้องการแก้ไขข้อมูล (1111, "Mr.Charnchai", "U of Ubonratchatani") เป็น (1111, "Mr.Charnchai S.", "Ubonratchatani University") จะสามารถเขียนคำสั่งได้ ดังนี้

Update customer Set name="Mr.Charnchai S.", address="Ubonratchatani University" Where id = 1111;

หลังจากนั้นให้ทดสอบว่าการแก้ไขนี้ได้แก้ไขตามที่ต้องการหรือไม่ โดยใช้คำสั่ง Select * from customer; เช่นเดิมเพื่อตรวจสอบเรคคอร์ดที่อยู่ในตาราง ผลลัพธ์ของทั้งหมดเป็น ดังนี้



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> Update customer Set name='Mr.Charnchai S.', address='Ubonratchatani University' Where id = 1111;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from customer;
+----+-----+-----+
| id | name          | address                |
+----+-----+-----+
| 1111 | Mr.Charnchai S. | Ubonratchatani University |
+----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

รูปที่ 11.27 ตัวอย่างการใช้คำสั่ง Update

คำสั่งลบเรคคอร์ดในตาราง

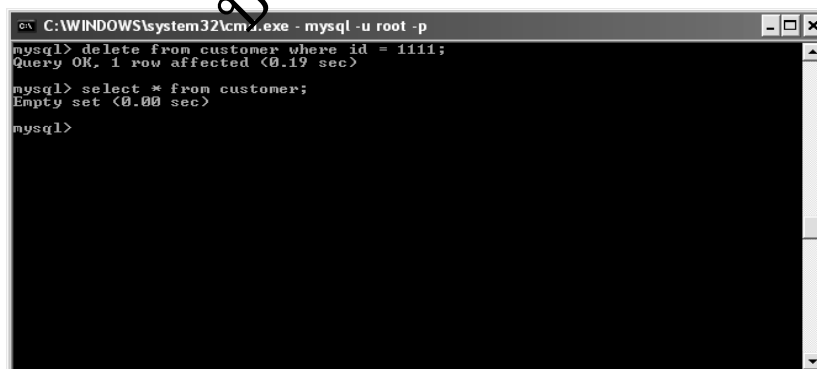
รูปแบบ

Delete From ชื่อตาราง Where ชื่อฟิลด์;

สมมติต้องการลบเรคคอร์ดของข้อมูลลูกค้าคนหนึ่งที่รหัสลูกค้า คือ 1111 จะสามารถเขียนคำสั่งได้ ดังนี้

Delete From customer Where id = 1111;

หลังจากนั้นให้ทดสอบว่าการใช้คำสั่งนี้จะลบเรคคอร์ดที่ต้องการให้หรือไม่ โดยพิมพ์คำสั่ง select * from customer; ผลลัพธ์ทั้งหมดเป็น ดังนี้



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql> delete from customer where id = 1111;
Query OK, 1 row affected (0.19 sec)

mysql> select * from customer;
Empty set (0.00 sec)

mysql>
```

รูปที่ 11.28 ตัวอย่างการใช้คำสั่ง Delete

สงวนลิขสิทธิ์
ผู้ดูแลสิทธิ์
มีเดีย จำกัด

จากตัวอย่างข้างต้นจะเห็นว่า หลังจากใช้คำสั่ง Delete แล้วทดสอบว่าข้อมูลเรคคอร์ดที่ลบไปนั้นหายไปหรือไม่ ผลลัพธ์ที่ได้ก็คือ Empty set ซึ่งหมายความว่าขณะนี้ในตารางนี้ว่างเปล่า

Warning

สำหรับคำสั่งลบเรคคอร์ดในตารางนี้ ในกรณีที่ใช้คำสั่งโดยไม่ระบุเงื่อนไขใดๆ ในการลบ ผลลัพธ์ก็คือ จะทำการลบเรคคอร์ดทุกเรคคอร์ดที่อยู่ในตาราง ซึ่งจะไม่สามารถเรียกให้ข้อมูลกลับคืนมาได้ด้วย ดังนั้นในการใช้คำสั่งนี้จะต้องตรวจสอบให้แน่ใจก่อนทุกครั้ง

คำสั่งออกจาก MySQL

Exit:

คำสั่งนี้ใช้ในกรณีที่ต้องการออกจากฐานข้อมูล MySQL ดังนั้นหลังจากเรียกใช้คำสั่งนี้แล้ว จะกลับออกไปอยู่ที่ระบบปฏิบัติการปกติเหมือนก่อนที่จะเข้ามาใช้ MySQL

สรุป

โปรแกรมฐานข้อมูล MySQL เป็นฐานข้อมูลที่ใช้จัดเก็บข้อมูลของระบบงานที่ถูกพัฒนาด้วยโปรแกรมภาษา PHP โดยขั้นตอนของการติดตั้งจะเรียกจากการดาวน์โหลด MySQL ได้จากเว็บไซต์ <http://www.mysql.com/download/> หลังจากนั้นก็เริ่มเข้าสู่กระบวนการติดตั้ง ซึ่งได้อธิบายไว้ในบทนี้

เนื้อหาในส่วนของการใช้งาน MySQL ในบทนี้จะเป็นการใช้งานในโหมด Command Line คือผู้ใช้จะต้องทำการพิมพ์คำสั่งต่างๆ เอง ซึ่งจะต้องศึกษาและทำความเข้าใจคำสั่งต่างๆ โดยเฉพาะอย่างยิ่งคำสั่ง SQL ในการจัดการฐานข้อมูล เช่น การสร้างฐานข้อมูล, การสร้างตาราง, การเพิ่มเรคคอร์ด, การแก้ไขเรคคอร์ด, การลบเรคคอร์ด เป็นต้น

แบบฝึกหัดบทที่ 11

1. ให้เขียนคำสั่ง SQL เพื่อสร้างฐานข้อมูล และตาราง ซึ่งมีรายละเอียด ดังนี้

Database Name: Employee

Table: Emp

ชื่อฟิลด์	ชนิด(ความยาว)	คำอธิบาย	หมายเหตุ
Empno	Int(4)	รหัสพนักงาน	Primary Key
Ename	VarChar(25)	ชื่อพนักงาน	
Job	VarChar(15)	ตำแหน่งงาน	
Mgr	Int(4)	รหัสพนักงานที่เป็นหัวหน้า	
Hiredate	VarChar(20)	วันที่เริ่มงาน	
Sal	Int(5)	เงินเดือน	
Comm	Int(4)	ค่าคอมมิชชั่น	
Deptno	Int(2)	รหัสแผนก	

2. จากตารางฐานข้อมูลที่สร้างในข้อ 1. ให้ทำการเขียนคำสั่ง SQL เพื่อเพิ่มเรคคอร์ดลงไปในตาราง Emp ดังนี้

Table: Emp

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SOMPONG	CLERK	7902	17 DEC 1980	800	NULL	20
7499	AEK	SALESMAN	7902	20 FEB 1981	1600	300	30
7521	WILAI	SALESMAN	7902	22 FEB 1981	1250	500	30
7902	KING	PRESIDENT	NULL	17 NOV 1981	5000	NULL	10

3. จากเรคคอร์ดในข้อ 2. ให้ทำการเขียนคำสั่ง SQL เพื่อแก้ไขเรคคอร์ดของพนักงานที่มีชื่อว่า KING โดยแก้ไขจากเงินเดือน 5000 เป็น 6500

4. จากตารางในข้อ 2. ให้ทำการเขียนคำสั่ง SQL เพื่อดึงเรคคอร์ดตามเงื่อนไขต่อไปนี้

4.1) ดึงทุกเรคคอร์ดจากตาราง Emp

4.2) แสดงเฉพาะชื่อพนักงาน และเงินเดือนของพนักงานที่มีเงินเดือนมากกว่า 1500

4.3) ดึงทุกเรคคอร์ด โดยแสดงเฉพาะชื่อพนักงาน และตำแหน่งงาน

4.4) ดึงเรคคอร์ดของพนักงานคนที่ไม่มีความคอมมิชชั่น

4.5) ดึงเรคคอร์ดของพนักงานที่มีตัวอักษร "I" อยู่ภายในชื่อ

5. จากเรคคอร์ดในข้อ 2. ให้ทำการเขียนคำสั่ง SQL เพื่อลบเรคคอร์ดของพนักงานที่มีเงินเดือนต่ำกว่า 1,000 บาท

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด

การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin

จากเนื้อหาในบทที่ผ่านมาเราได้เรียนรู้การใช้งานฐานข้อมูล MySQL รวมทั้งเรียนรู้คำสั่งที่ใช้ในการจัดการฐานข้อมูลด้วย เราจะเห็นข้อเสียของการใช้งานดังกล่าวก็คือ ความยากลำบากในการจัดการฐานข้อมูล เช่น การที่ต้องเข้าใจและเรียกใช้คำสั่งได้ถูกต้อง มิฉะนั้นแล้วเราก็ไม่สามารถใช้งาน MySQL ได้เลย ในปัจจุบันเราสามารถแก้ปัญหาความยากลำบากดังกล่าวได้โดยการใช้โปรแกรมสำเร็จรูปที่มีคุณสมบัติของการทำงานในลักษณะ Graphic User Interface (GUI) โดยที่ไม่ต้องเรียนรู้คำสั่งในการจัดการฐานข้อมูลเลย ผู้ใช้เพียงคลิกเลือกคำสั่งต่างๆ ที่จัดเตรียมมาให้แล้ว เราก็จะสามารถใช้จัดการฐานข้อมูล MySQL ได้อย่างง่ายดาย โปรแกรมที่ใช้จัดการฐานข้อมูลดังกล่าวมีอยู่หลายโปรแกรม แต่ที่นิยมใช้มากที่สุดก็คือ โปรแกรม PhpMyAdmin

จุดประสงค์ ทั่วไป

1. เพื่อศึกษาการติดตั้งโปรแกรม PhpMyAdmin
2. เพื่อเรียนรู้การใช้งานโปรแกรม PhpMyAdmin
3. เพื่อศึกษาการใช้งาน PhpMyAdmin

จุดประสงค์ เชิงพฤติกรรม

1. สามารถติดตั้งโปรแกรม PhpMyAdmin ได้
2. สามารถใช้โปรแกรม PhpMyAdmin จัดการฐานข้อมูลได้

เนื้อหาสาระ

1. การติดตั้งโปรแกรม PhpMyAdmin
2. การใช้งานโปรแกรม PhpMyAdmin

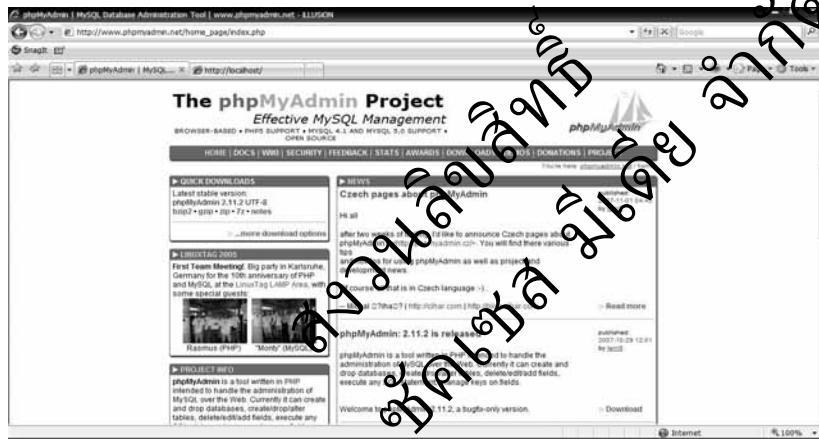
การทำงานกับโปรแกรมฐานข้อมูล MySQL หากเป็นการใช้งานสำหรับมือใหม่ ก่อนข้างที่จะใช้งานยากอยู่พอสมควร เพราะต้องจดจำคำสั่งต่างๆ มากมาย และมีการทำงานอยู่ในรูปแบบของเท็กซ์โหมด (Text Mode) อีกด้วย ซึ่งผู้ใช้งานใหม่ในปัจจุบันคงไม่ค่อยคุ้นเคยนัก

ในปัจจุบันมีเครื่องมือตัวหนึ่งที่มีผู้ใช้งานมากที่สุดในขณะนี้ ชื่อว่า PhpMyAdmin ซึ่งพัฒนามาจาก PHP โดยทีมงาน Tobias Ratschiller หากต้องการรายละเอียดเพิ่มเติม หรือดาวน์โหลดโปรแกรมสามารถเข้าไปที่เว็บไซต์ <http://www.phpmyadmin.net>

12.1 การติดตั้งโปรแกรม PhpMyAdmin

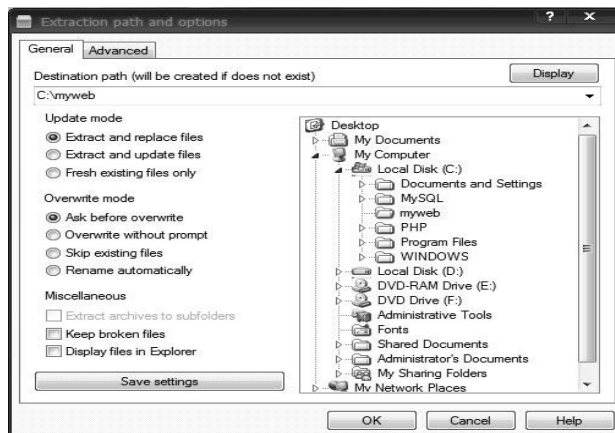
การติดตั้ง PhpMyAdmin มีขั้นตอน ดังต่อไปนี้

1. ทำการ download โปรแกรม PhpMyAdmin ได้ที่เว็บไซต์ <http://www.phpmyadmin.net/>



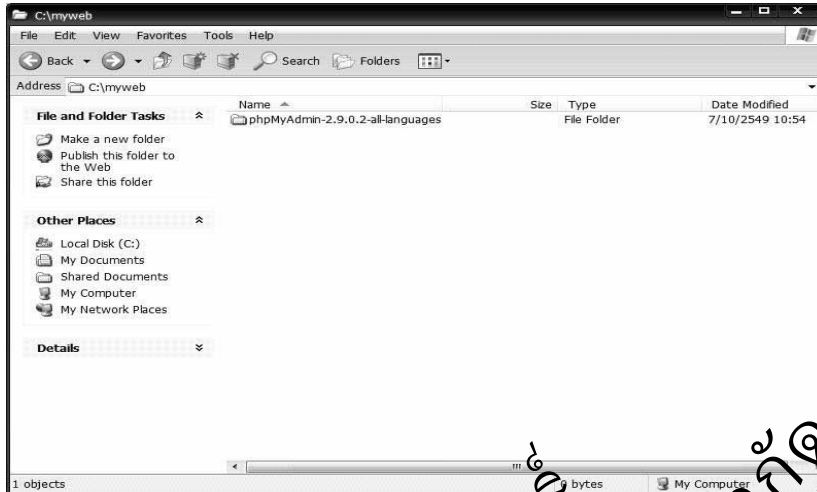
รูปที่ 12.1 หน้าจอเว็บไซต์ในการดาวน์โหลดโปรแกรม PhpMyAdmin

2. ไฟล์ที่ download มาได้จะเป็นไฟล์ Zip ให้ทำการแตกไฟล์ดังกล่าวไว้ในโฟลเดอร์ที่เป็น Document Root (โฟลเดอร์ที่เก็บไฟล์ php ตามที่ปรับแต่งค่าไว้ในโปรแกรม Apache) โดยในที่นี้จะทำการแตกไฟล์ไว้ที่ C:\myweb



รูปที่ 12.2 หน้าจอแสดงการแตกไฟล์ไว้ที่ C:\myweb

3. เข้าไปที่โฟลเดอร์ C:\myweb จะเห็นโฟลเดอร์ชื่อว่า phpMyAdmin-2.9.0.2-all-languages ให้ทำการแก้ไขชื่อเป็น phpmyadmin

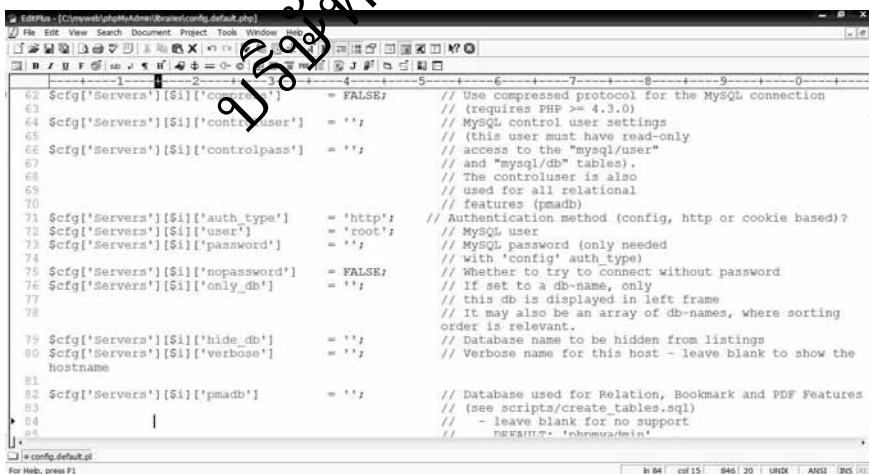


รูปที่ 12.3 หน้าจอ Windows Explorer แสดงข้อมูลในโฟลเดอร์ C:\myweb

4. หลังจากนั้นให้เข้าไปในโฟลเดอร์ C:\phpmyadmin\libraries แล้วเปิดไฟล์ที่ชื่อว่า config.default.php ขึ้นมา เพื่อทำการปรับแต่งจากคอนฟิก

4.1) ทหารรหัสที่เขียนว่า \$cfg['Servers'][\$i]['auth_type'] = 'config'; ให้ทำการแก้ไขเป็น \$cfg['Servers'][\$i]['auth_type'] = 'http';

โดยในบรรทัดนี้เป็นการ authentication คือการตรวจสอบการเข้าใช้งาน PhpMyAdmin ให้ผ่านโปรโตคอล http

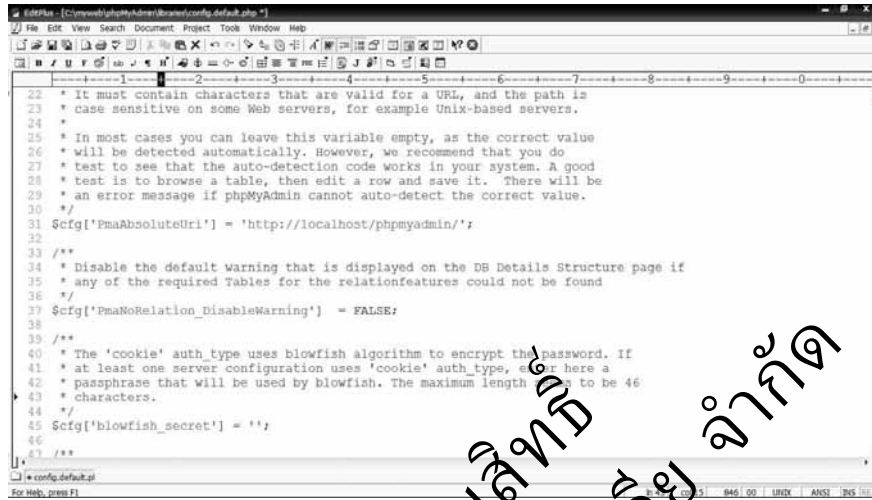


รูปที่ 12.4 หน้าจอแสดงการปรับแต่งค่า Authentication method

4.2) ทหารที่ที่เขียนว่า \$cfg['PmaAbsoluteUri'] = ''; ให้ทำการแก้ไขเป็น

\$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin/';

ซึ่งเป็นการกำหนด URL ในการเข้าถึง PhpMyAdmin โดยจะต้องกำหนดชื่อพาธให้ถูกต้อง และจะมองเรื่องของ Case-sensitive ด้วย



```
22 * It must contain characters that are valid for a URL, and the path is
23 * case sensitive on some Web servers, for example Unix-based servers.
24 *
25 * In most cases you can leave this variable empty, as the correct value
26 * will be detected automatically. However, we recommend that you do
27 * test to see that the auto-detection code works in your system. A good
28 * test is to browse a table, then edit a row and save it. There will be
29 * an error message if phpMyAdmin cannot auto-detect the correct value.
30 */
31 $cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin/';
32
33 /**
34 * Disable the default warning that is displayed on the DB Details Structure page if
35 * any of the required Tables for the relationfeatures could not be found
36 */
37 $cfg['PmaNoRelation_DisableWarning'] = FALSE;
38
39 /**
40 * The 'cookie' auth_type uses blowfish algorithm to encrypt the password. If
41 * at least one server configuration uses 'cookies' auth type, here a
42 * passphrase that will be used by blowfish. The maximum length should be 46
43 * characters.
44 */
45 $cfg['blowfish_secret'] = '';
46
47 /**
```

รูปที่ 12.5 หน้าจอแสดงการปรับแต่งค่า \$cfg('PmaAbsoluteUri')

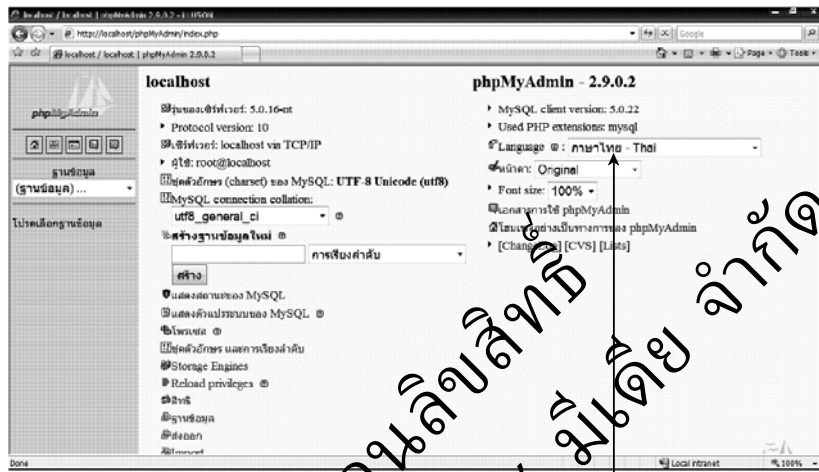
4.3) บันทึกไฟล์ config.default.php เป็นอันเสร็จขั้นตอนการติดตั้งโปรแกรม Phpmyadmin ทดสอบการทำงานของโปรแกรมโดยเปิดโปรแกรมบราวเซอร์ขึ้นมาแล้วพิมพ์ URL เป็น http://localhost/phpmyadmin/ จะปรากฏหน้าจอของการ Login เข้าสู่โปรแกรม ให้ทำการพิมพ์ User name และ Password ตามที่เคยกำหนดไว้ตอนที่ติดตั้งโปรแกรม MySQL เสร็จแล้วทำการคลิกที่ปุ่ม OK จะเข้าสู่หน้าจอโปรแกรม PhpMyAdmin



รูปที่ 12.6 หน้าจอแสดงการ Login เข้าใช้งานโปรแกรม PhpMyAdmin เพื่อเข้าสู่หน้าจอ PhpMyAdmin

12.2 การใช้งานโปรแกรม PhpMyAdmin

จากรูปที่ 12.6 คือหน้าหลักของ PhpMyAdmin โดยแบ่งพื้นที่ของหน้าเว็บออกเป็น 2 ส่วน คือพื้นที่ฝั่งซ้าย จะมี Drop Down เพื่อเลือกฐานข้อมูลที่ต้องการจัดการ นอกจากนั้นรายชื่อตาราง (Table) ก็จะถูกกำหนดไว้ในส่วนนี้ด้วย พื้นที่ฝั่งขวาจะแสดงรายละเอียดต่างๆ ของฐานข้อมูลและตารางที่เรากำลังจัดการอยู่ ก่อนเริ่มต้นใช้งานให้คลิกเลือกภาษาไทยในการใช้งานที่ช่อง Drop Down ของ Language บริเวณฝั่งด้านขวาของโปรแกรม

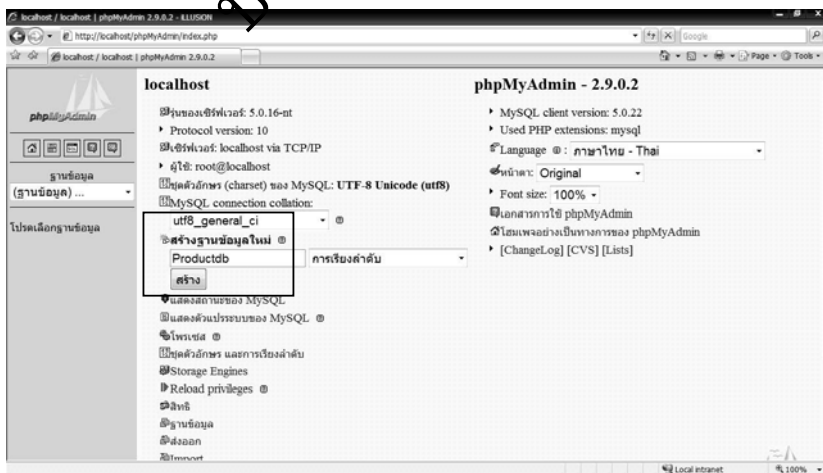


รูปที่ 12.7 คลิกเลือกเพื่อใช้งานภาษาไทย

การสร้างและลบฐานข้อมูล

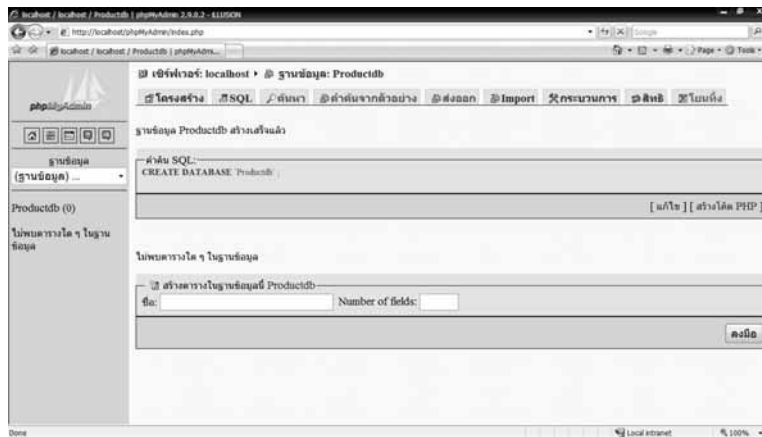
ทดลองสร้างฐานข้อมูลชื่อว่า Productdb ดังนี้

1. จากหน้าหลักของ PhpMyAdmin ให้ทำการพิมพ์ชื่อ Productdb ลงไปในช่อง text box "สร้างฐานข้อมูลใหม่" หลังจากนั้นให้คลิกที่ปุ่ม "สร้าง"



รูปที่ 12.8 พิมพ์ชื่อฐานข้อมูลที่ต้องการสร้าง

2. จะปรากฏหน้าจอ ดังต่อไปนี้



รูปที่ 12.9 หน้าจอหลังจากสร้างฐานข้อมูล

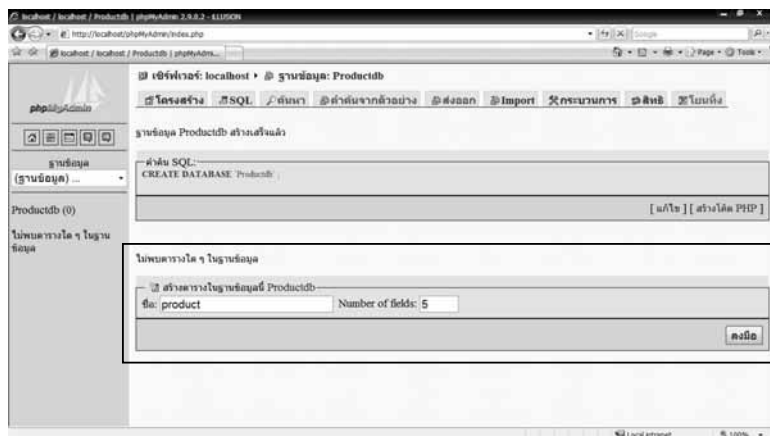
จากรูปที่ 12.9 จะมีรายละเอียด ดังนี้

- **พื้นที่ฝั่งซ้าย** จะแสดงชื่อฐานข้อมูลใหม่บนปุ่ม Drop Down ลังเกตที่ข้อความใต้ Drop Down บอกว่า "ไม่พบตารางใดๆ ในฐานข้อมูล" ซึ่งหมายความว่าขณะนี้ในฐานข้อมูลนี้ไม่มีตารางเลย โดยวิธีการสร้างตารางภายในฐานข้อมูลจะได้กล่าวถึงในส่วนถัดไป
- **พื้นที่ฝั่งขวา** จะแสดง Text box ให้พิมพ์ชื่อตารางที่ต้องการสร้างในฐานข้อมูลนี้ และระบุจำนวนฟิลด์ที่ต้องการภายในช่อง Number of field

การสร้างตาราง

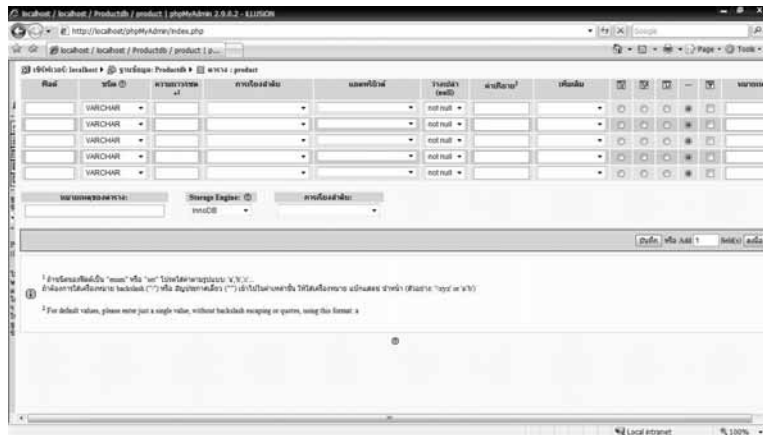
ทดลองสร้างตารางชื่อว่า product โดยมีฟิลด์ทั้งหมด 5 ฟิลด์ ดังนี้

1. พิมพ์ชื่อตาราง product ในช่องชื่อ และพิมพ์จำนวนฟิลด์เท่ากับ 5 ในช่อง Number of field หลังจากนั้นให้คลิกที่ปุ่ม "ลงมือ" ดังนี้




รูปที่ 12.10 การตั้งชื่อตารางและกำหนดจำนวนฟิลด์

2. จะปรากฏหน้าจอในการกำหนดรายละเอียดของตาราง ดังนี้



รูปที่ 12.11 การกำหนดรายละเอียดของตาราง

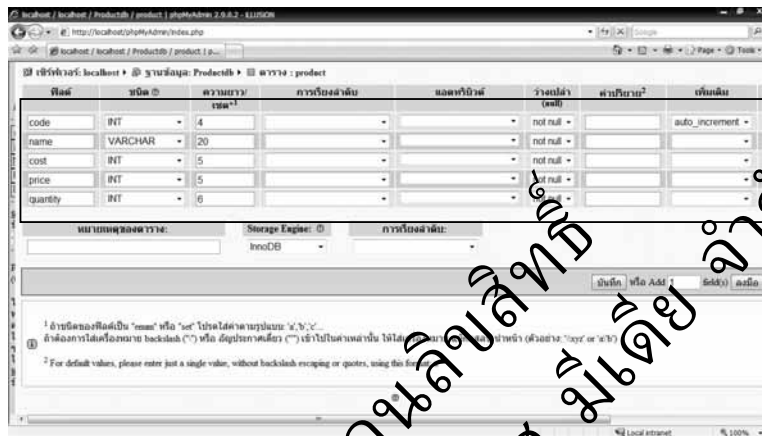
จากรูปที่ 12.11 จะปรากฏหน้าจอให้กรอกรายละเอียดต่างๆ ของตาราง **product** (มีทั้งหมด 5 แถว โดย 1 แถว เท่ากับ 1 ฟิลด์) โดยประกอบไปด้วยรายการข้อมูลดังนี้

- ฟิลด์ ใช้สำหรับพิมพ์ชื่อฟิลด์
- ชนิด ใช้สำหรับกำหนดชนิดของข้อมูลที่อยู่ในฟิลด์นั้นๆ
- ความยาว/เซต * ใช้สำหรับกำหนดความยาวของฟิลด์นั้นๆ
- การเรียงลำดับ ใช้สำหรับกำหนดการเรียงลำดับของข้อมูลภายในตาราง
- แอตทริบิวต์ ใช้สำหรับกำหนดลักษณะของฟิลด์นั้นๆ
- ว่างเปล่า (null) ใช้สำหรับกำหนดว่าฟิลด์นั้นๆ จะให้มีค่าว่างเปล่าได้หรือไม่
- ค่าปริยาย ใช้สำหรับกำหนดค่าของฟิลด์ในกรณีที่ไม่ได้มีการกรอกข้อมูลใดๆ
- เพิ่มเติม ใช้สำหรับกำหนดให้ฟิลด์นั้นมีลักษณะที่เป็นแบบ auto_increment
-  ใช้สำหรับกำหนดให้ฟิลด์นั้นเป็นคีย์หลัก (Primary Key)

จากรายการข้อมูลเราไม่จำเป็นต้องกรอกข้อมูลให้ครบทุกรายการก็ได้ ในที่นี้สมมติว่าต้องการสร้างตาราง **product** ที่มีรายละเอียดต่างๆ ดังนี้

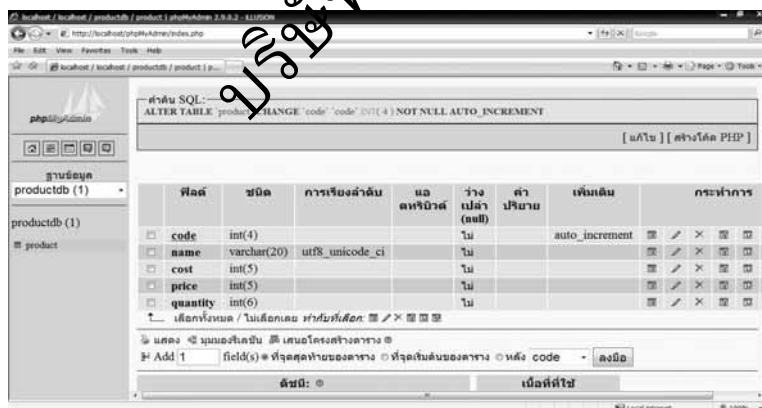
ชื่อฟิลด์	ชนิดข้อมูล	ความยาว	อื่นๆ
code	INT	4	ให้เป็นคีย์หลัก (ไพรมารี) และมีลักษณะเป็น auto_increment
name	VARCHAR	20	
cost	INT	5	
price	INT	5	
quantity	INT	6	

ดังนั้น เราจะสามารถกำหนดรายละเอียดของตาราง product นี้ด้วยโปรแกรม PhpMyAdmin ได้
ดังนี้





รูปที่ 12.12 การกำหนดรายละเอียดของตาราง


หลังจากกำหนดรายละเอียดต่างๆ ของตารางเรียบร้อยแล้ว ให้ทำการคลิกที่ปุ่ม "บันทึก" โปรแกรมก็จะสร้างตาราง และรายละเอียดตามที่ต้องการ โดยจะปรากฏหน้าจอสกรูปรายละเอียดต่างๆ ของตาราง ดังนี้

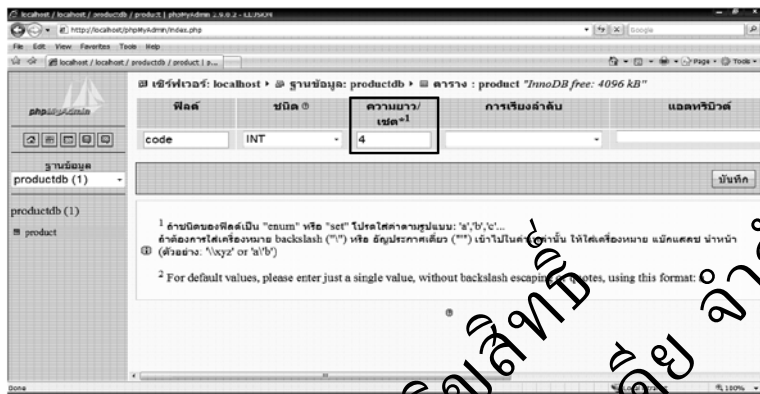


รูปที่ 12.13 หน้าจอสกรูปรายละเอียดของตาราง

การแก้ไขรายละเอียดของตาราง

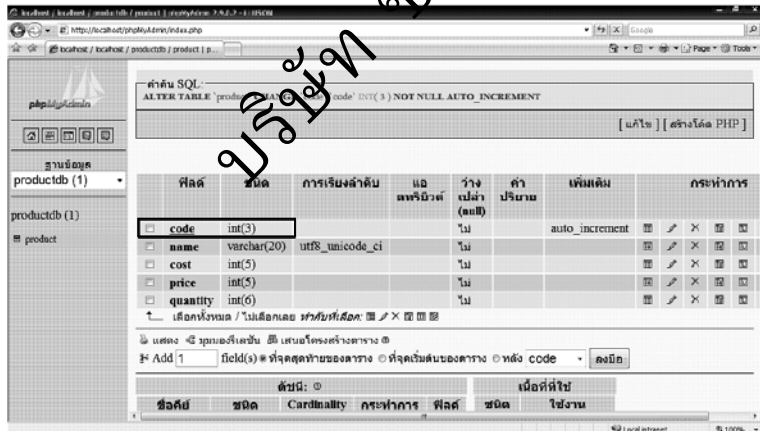
จากรูปที่ 12.13 นี้เราสามารถทำการแก้ไขรายละเอียดของตารางได้โดยคลิกที่ Check box หน้าชื่อฟิลด์ที่ต้องการแก้ไข หรือลบออก โดยให้เลือกฟิลด์ที่ต้องการ และคลิกที่ไอคอน  ถ้าต้องการแก้ไข หรือคลิกที่ไอคอน  ถ้าต้องการลบฟิลด์นั้นออกจากตาราง

ยกตัวอย่างโดยทำการคลิกที่ไอคอน  ด้านท้ายฟิลด์ที่ชื่อ code เพื่อแก้ไขรายละเอียดของฟิลด์จะปรากฏหน้าจอ ดังนี้



รูปที่ 12.14 ตัวอย่างการแก้ไขรายละเอียดของฟิลด์ code

สมมติเราต้องการแก้ไขความยาวของฟิลด์ code จากความยาว 4 เป็น 3 ก็ทำการแก้ไขค่าความยาวในช่อง Text box หลังจากนั้นก็ให้ทำการคลิกที่ปุ่ม  ก็จะมีหน้าจอ ดังนี้

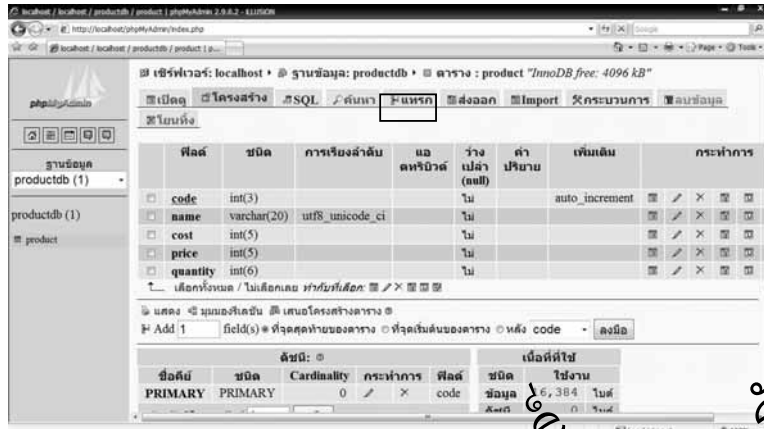


รูปที่ 12.15 ผลลัพธ์หลังจากทำการแก้ไขความยาวของฟิลด์ code

การเพิ่มเรคคอร์ดใหม่ในตาราง

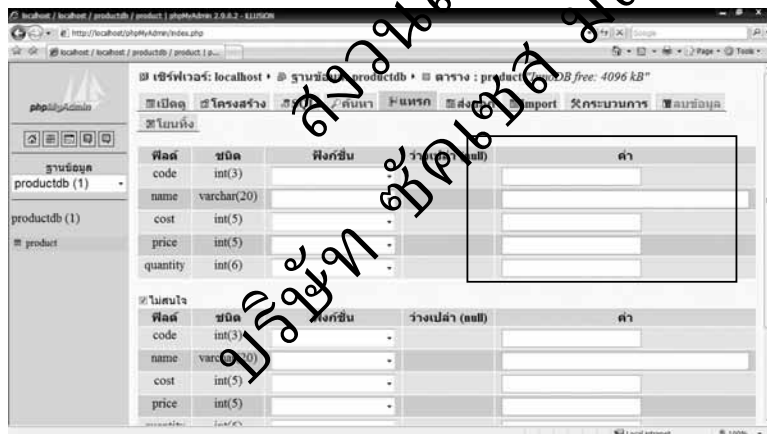
การเพิ่มเรคคอร์ดใหม่ในตาราง มีขั้นตอนดังนี้

1. คลิกที่ลิงค์ "แทรก" บนเนวิเกชันบาร์



รูปที่ 12.16 ลิงค์ "แทรก" สำหรับการเพิ่มเรคคอร์ดใหม่ในตาราง

2. จะปรากฏหน้าจอการเพิ่มเรคคอร์ดใหม่ในตาราง ดังนี้



รูปที่ 12.17 หน้าจอการเพิ่มเรคคอร์ดใหม่

3. ที่คอลัมน์ "ค่า" ให้กรอกรายละเอียดลงไปในแต่ละช่องของ Text box ในที่นี้ทดลองใส่ค่าดังนี้
code : 111, name : RAM 512 MB, cost : 500, price : 750, quantity : 50

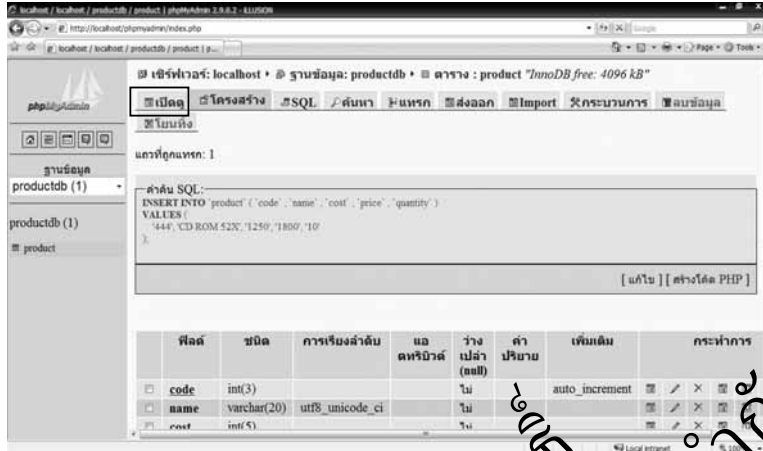
หลังจากนั้นคลิกที่ปุ่ม "ลงมือ" บริเวณด้านล่าง

สำหรับกรณีที่ต้องการเพิ่มเรคคอร์ดใหม่ในตาราง ก็สามารถทำได้โดยด้วยวิธีการเดิม คือ คลิกที่ลิงค์ "แทรก" บนเนวิเกชันบาร์ ซึ่งจะปรากฏช่องว่างให้ทำการเพิ่มเรคคอร์ดใหม่ลงในแต่ละฟิลด์ หลังจากนั้นก็ให้ทำการคลิกที่ปุ่ม "ลงมือ"

การดูข้อมูลภายในตาราง

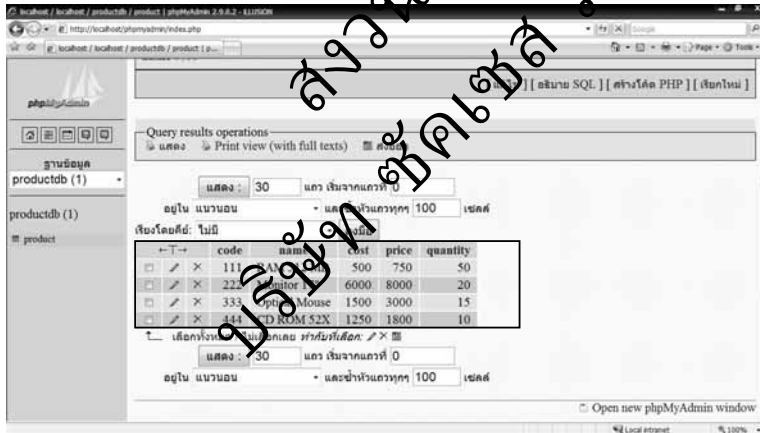
การดูข้อมูลที่ได้อีกรายละเอียดลงไปแล้ว มีขั้นตอน ดังนี้

1. คลิกที่ลิงค์ "เปิดดู" บนเนวิเกชันบาร์



รูปที่ 12.18 ลิงค์ "เปิดดู" สำหรับการดูข้อมูลในตาราง


2. จะปรากฏหน้าจอแสดงเรคคอร์ดทั้งหมดที่อยู่ในตารางนี้ ดังนี้

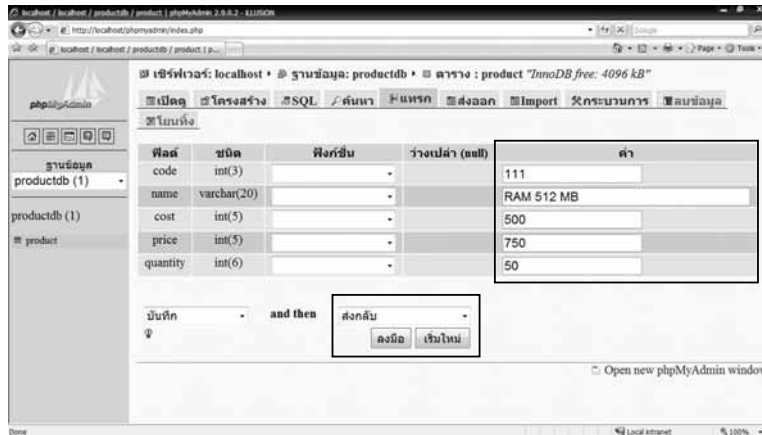


รูปที่ 12.19 หน้าจอแสดงเรคคอร์ดที่อยู่ในตาราง

การแก้ไขข้อมูลของเรคคอร์ด

สมมติเราต้องการแก้ไขเรคคอร์ดที่ 1 (111, RAM 128 MB, 500, 750, 50) ให้ทำตามขั้นตอน ดังนี้

1. จากรูปที่ 12.19 บริเวณด้านหน้าของแต่ละฟิลด์ให้คลิกที่ไอคอน  หน้าเรคคอร์ดที่ต้องการ จะปรากฏหน้าจอ ดังนี้



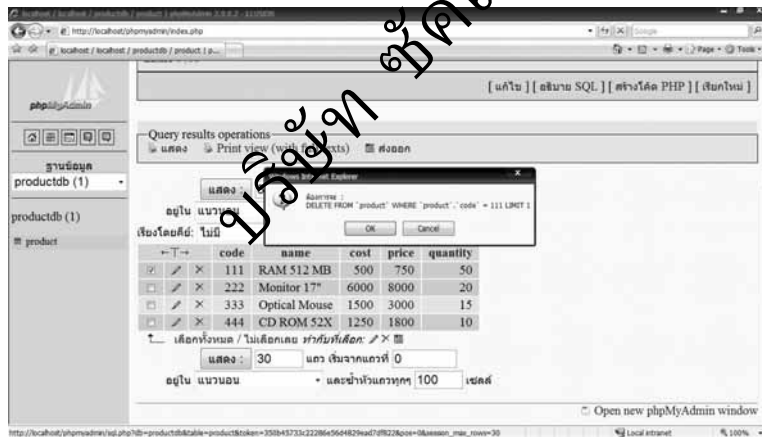
รูปที่ 12.20 หน้าจอการแก้ไขเรคคอร์ด

2. ทำการแก้ไขข้อมูลของเรคคอร์ดใหม่ (ที่คอลัมน์ "ค่า") ตามที่ต้องการ หลังจากนั้นก็ทำการคลิกที่ปุ่ม "ลงมือ"

การลบเรคคอร์ด

หากต้องการลบเรคคอร์ดที่ไม่ต้องการทิ้ง ไม่ให้ทำตามขั้นตอนดังนี้

1. จากรูปที่ 12.18 บริเวณด้านหน้าของแต่ละฟิลด์ ให้คลิกที่ไอคอน ด้านหน้าเรคคอร์ดที่ต้องการลบ ก็จะปรากฏหน้าจอ ดังนี้



รูปที่ 12.21 หน้าจอแสดง Dialogue Box ยืนยันการลบ

2. จะปรากฏหน้าจอเพื่อให้ยืนยันการลบเรคคอร์ดนี้ โดยถ้าต้องการลบก็ให้ทำการคลิกที่ปุ่ม "OK" แต่ถ้าไม่ต้องการลบก็ให้คลิกที่ปุ่ม "Cancel"

การลบตาราง

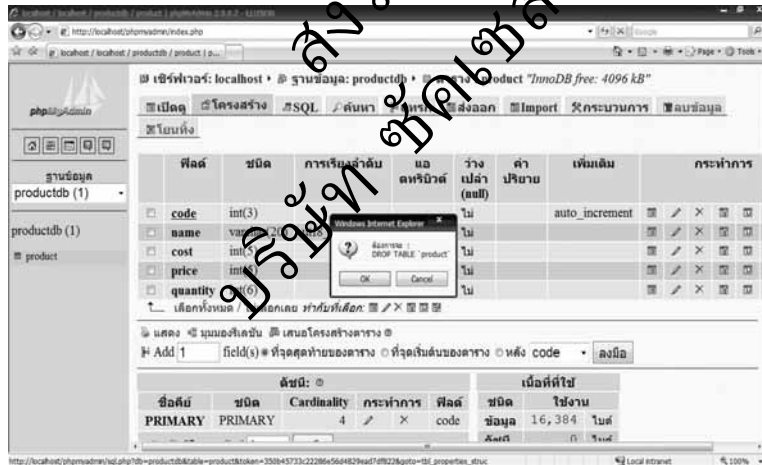
การลบตารางที่ มีขั้นตอน ดังนี้

1. คลิกที่ลิงค์ "โยนทิ้ง" บนเนวิเกชันบาร์



รูปที่ 12.22 ลิงค์ "โยนทิ้ง" สำหรับการลบตาราง

2. จะปรากฏหน้าต่างเพื่อให้ยืนยันการลบตารางนี้ ดังนี้



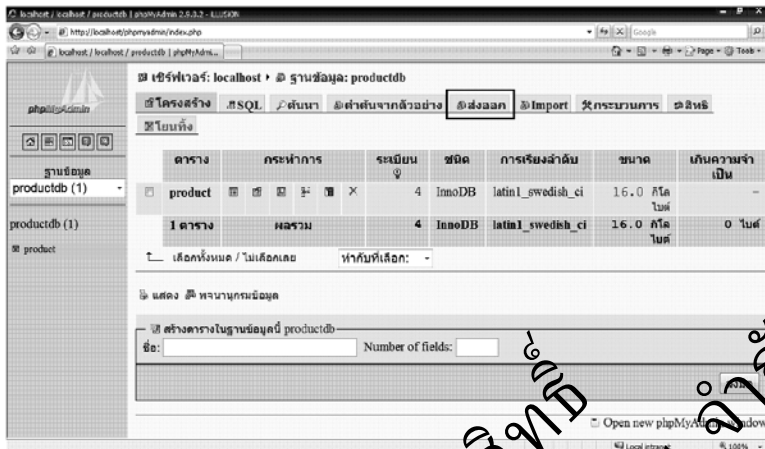
รูปที่ 12.23 หน้าจอยืนยันการลบตาราง

3. ถ้าต้องการลบตารางก็ให้คลิกที่ปุ่ม "OK" แต่ถ้าไม่ต้องการลบก็ให้คลิกที่ปุ่ม "Cancel"

การส่งออกฐานข้อมูล

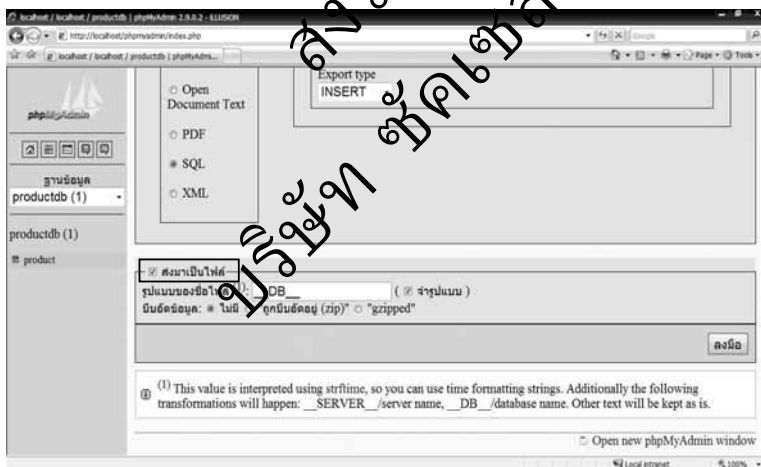
เมื่อเราจัดสร้างฐานข้อมูลโดยใช้โปรแกรม PhpMyAdmin แล้ว และต้องการนำฐานข้อมูลดังกล่าวนี้ไปจัดเก็บที่เครื่องคอมพิวเตอร์เครื่องอื่น เราสามารถทำได้ตามขั้นตอน ต่อไปนี้

1. คลิกที่ลิงค์ "ส่งออก" บนเมนูเช็คนบาร์



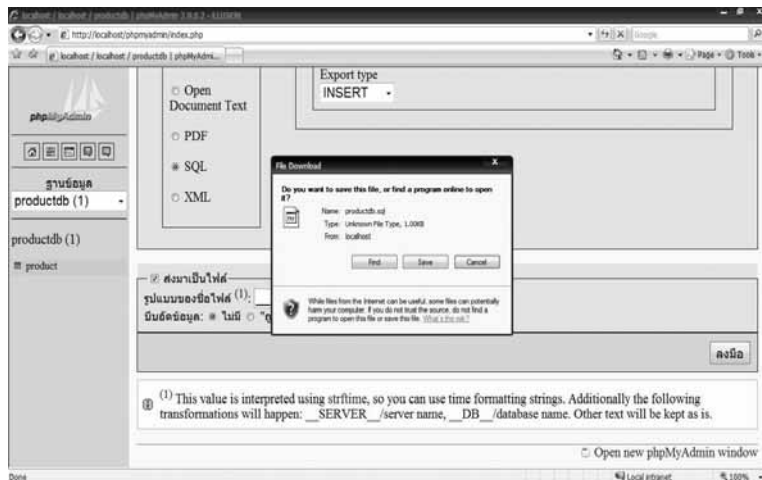
รูปที่ 12.24 ลิงค์ "ส่งออก" สำหรับการส่งออกฐานข้อมูล

2. จะปรากฏหน้าจอเพื่อใช้ส่งออกฐานข้อมูล เลื่อนสกรอลิ่งไปด้านล่างจะเห็นรายละเอียด ดังนี้



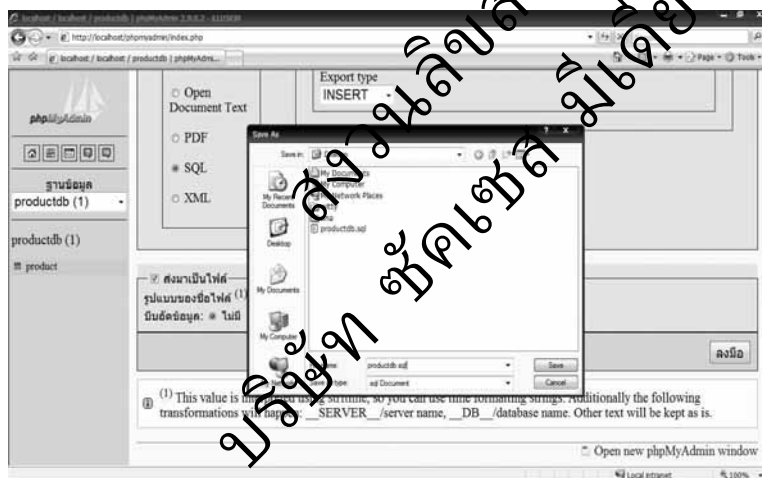
รูปที่ 12.25 หน้าจอการส่งออกฐานข้อมูล

3. ให้ทำการคลิกเลือกที่ Check Box "ส่งมาเป็นไฟล์" หลังจากนั้นก็ให้ทำการคลิกที่ปุ่ม "ลงมือ"
4. จะปรากฏหน้าจอให้ทำการบันทึกฐานข้อมูลนี้เป็นไฟล์ ดังนี้



รูปที่ 12.26 หน้าจอให้บันทึกเป็นไฟล์

5. คลิกที่ปุ่ม "Save" ก็จะมีปรากฏหน้าต่างให้ทำการเลือกที่จัดเก็บไฟล์นี้ ซึ่งไฟล์ที่ได้จะมีนามสกุลเป็น .sql ที่สามารถนำไปติดตั้งยังเครื่องคอมพิวเตอร์เครื่องอื่น ๆ ได้




รูปที่ 12.27 หน้าจอแสดงการบันทึกไฟล์ที่ส่งออก

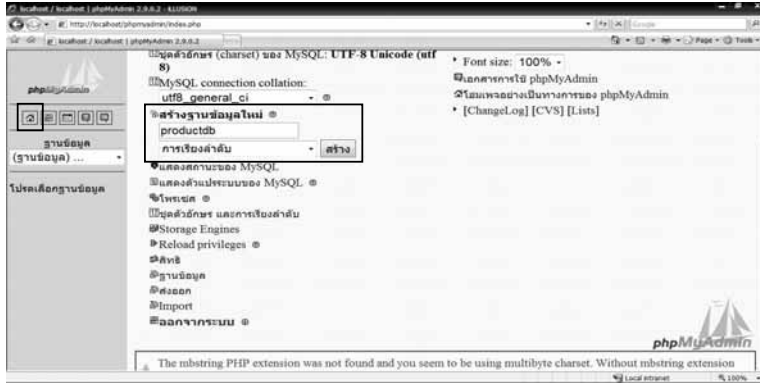
การนำเข้าฐานข้อมูล

หลังจากเราได้ทำการส่งออกฐานข้อมูลในหัวข้อที่ผ่านมาแล้ว เราก็สามารถนำไปติดตั้งที่เครื่องคอมพิวเตอร์เครื่องอื่น ๆ ได้ (ถ้าไม่ทำเช่นนี้และอยากจะได้ฐานข้อมูลที่เหมือนกันบนเครื่องคอมพิวเตอร์เครื่องอื่น เราก็ต้องเริ่มสร้างฐานข้อมูลใหม่ตั้งแต่ต้น) ขั้นตอนการนำเข้าฐานข้อมูลมีลำดับขั้นตอน ดังนี้

บทที่

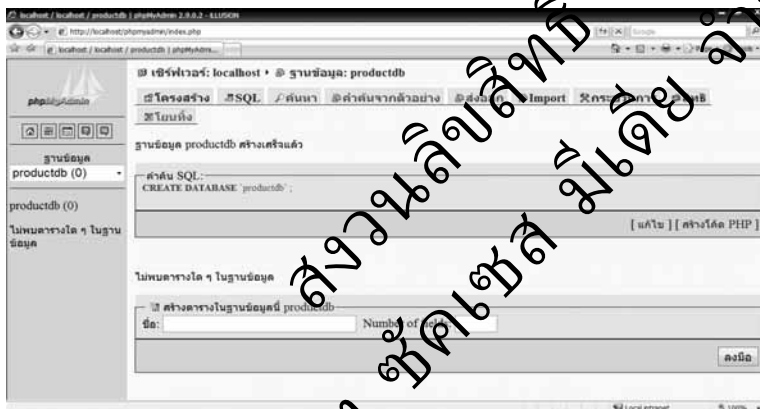
12

1. กลับไปที่หน้าแรกของ PhpMyAdmin โดยคลิกที่ไอคอนรูป  ที่บริเวณพื้นที่ฝั่งซ้าย ซึ่งจะปรากฏหน้าจอ ดังนี้



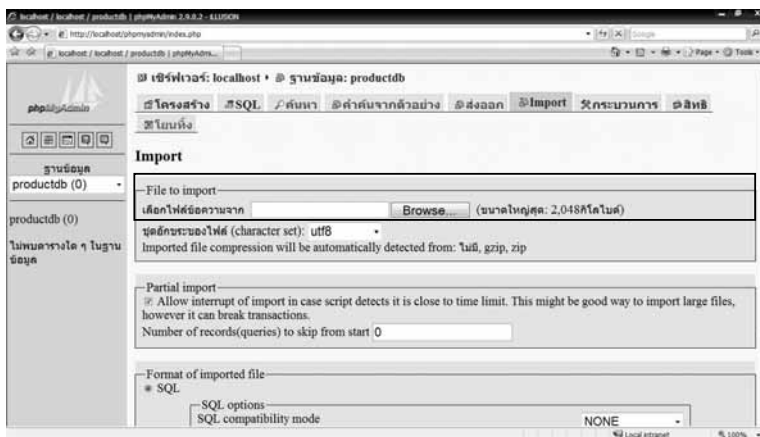
รูปที่ 12.28 หน้าแรกของโปรแกรม PhpMyAdmin

2. ให้ทำการสร้างฐานข้อมูลใหม่ก่อน โดยกรอกชื่อฐานข้อมูล และคลิกที่ปุ่ม "สร้าง" จะปรากฏหน้าจอ ดังนี้



รูปที่ 12.29 หน้าจอหลังจากการสร้างฐานข้อมูล

3. เราไม่ต้องทำการสร้างดัชนีข้อมูล เนื่องจากเราจะทำการนำเข้าฐานข้อมูล ให้ทำการคลิกที่ลิงค์ Import จะปรากฏหน้าจอ ดังนี้



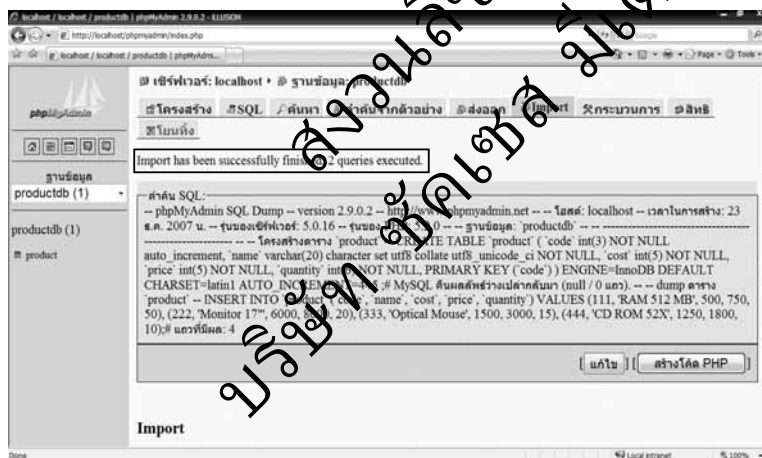
รูปที่ 12.30 หน้าจอแสดงการเลือกไฟล์ฐานข้อมูลที่จะนำเข้า

4. ให้ทำการคลิกปุ่ม Browse ที่ช่อง File to import เพื่อเลือกไฟล์ฐานข้อมูลที่ต้องการนำเข้า หลังจากนั้นจะปรากฏหน้าต่างให้เลือกไฟล์ .sql เพื่อใช้ในการนำเข้าฐานข้อมูล และเมื่อเลือกไฟล์นำเข้าได้แล้ว ก็ให้ทำการคลิกที่ปุ่ม "ลงมือ" บริเวณด้านล่าง



รูปที่ 12.31 หน้าจอแสดงหน้าต่างเพื่อเลือกไฟล์นำเข้า

5. เพียงเท่านี้เราก็จะได้ฐานข้อมูลที่ประกอบด้วยตารางและข้อมูลเรคคอร์ดทั้งหมด โดยให้สังเกตที่ข้อความ Import has been successfully finished



รูปที่ 12.32 หน้าจอแสดงการนำเข้าฐานข้อมูลเสร็จสมบูรณ์

สรุป

โปรแกรม PhpMyAdmin เป็นโปรแกรมที่ช่วยอำนวยความสะดวกในการจัดการฐานข้อมูล MySQL โดยเป็นโปรแกรมที่ทำงานในลักษณะ Graphic User Interface (GUI) ทำให้การใช้งานได้ง่าย ผู้ที่ไม่มีความรู้คำสั่ง SQL ก็สามารถใช้งานได้ โดยสามารถจัดการเข้าถึงฐานข้อมูลได้ทั้งหมด คือ สร้างและลบฐานข้อมูล, สร้างตาราง, แก้ไขรายละเอียดของตาราง, เพิ่มเรคคอร์ดใหม่ในตาราง, ดูข้อมูลภายในตาราง, แก้ไขข้อมูลของเรคคอร์ด, ลบตาราง, ส่งออกฐานข้อมูล และนำเข้าฐานข้อมูล

บทที่

12

แบบฝึกหัดบทที่ 12

1. จากฐานข้อมูล Employee ในแบบฝึกหัดบทที่ 11 ข้อที่ 1. ให้ใช้โปรแกรม PhpMyAdmin สร้างตารางเพิ่มขึ้นอีก 1 ตาราง ดังรายละเอียด ดังนี้

Database: Employee

Table: Dept

ชื่อฟิลด์	ชนิด(ความยาว)	คำอธิบาย	หมายเหตุ
Deptno	Int(2)	รหัสแผนก	Primary Key
Dname	VarChar(25)	ชื่อแผนก	
Loc	VarChar(20)	สถานที่ทำงาน	

2. จากตารางที่สร้างในข้อ 1. ให้ใช้โปรแกรม PhpMyAdmin ทำการเพิ่มเรคคอร์ดลงไปตาราง Dept ดังนี้

Table: Dept

Deptno	DName	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

3. ให้ใช้โปรแกรม PhpMyAdmin แก้ตั้งเรคคอร์ดจากทั้ง 2 ตาราง

4. ให้ใช้โปรแกรม PhpMyAdmin เปลี่ยนชื่อพนักงานจาก "KING" เป็น "CHANCHAI" ในตาราง Emp

5. จากฐานข้อมูล Employee ให้ทำการส่งออกฐานข้อมูลโดยสร้างเป็นไฟล์ที่แทนฐานข้อมูล

เราได้เรียนรู้การใช้งานฐานข้อมูล MySQL และการใช้งานโปรแกรม PhpMyAdmin เพื่อจัดการฐานข้อมูล MySQL ไปแล้ว และถึงแม้ว่าโปรแกรม PhpMyAdmin จะสามารถใช้จัดการฐานข้อมูล MySQL ได้ง่าย แต่ในทางปฏิบัติแล้วเราคงไม่อนุญาตให้ผู้ใช้สามารถเข้ามาใช้โปรแกรม PhpMyAdmin เพื่อจัดการฐานข้อมูล MySQL เพราะมีฉะนั้นแล้วผู้ใช้อาจจะทำการเพิ่ม ลบ หรือแก้ไขข้อมูลต่างๆ ในฐานข้อมูลโดยปกติในการจัดการฐานข้อมูลจะเป็นหน้าที่ของผู้ดูแลระบบ (Administrator) เท่านั้น วิธีแก้ปัญหาดังกล่าวก็คือ เราจะต้องเขียนโปรแกรมภาษา PHP เพื่อให้สามารถติดต่อและเข้าถึงฐานข้อมูล MySQL

จุดประสงค์ทั่วไป

1. เพื่อศึกษาการเขียนโปรแกรมภาษา PHP เพื่อติดต่อฐานข้อมูล MySQL
2. เพื่อเรียนรู้คำสั่ง และสามารถเขียนโปรแกรม PHP ในการเชื่อมต่อ และเปิดการเชื่อมต่อ MySQL
3. เพื่อเรียนรู้คำสั่ง และสามารถเขียนโปรแกรม PHP ในการสร้างฐานข้อมูลในตาราง
4. เพื่อเรียนรู้คำสั่ง และสามารถเขียนโปรแกรม PHP ในการเพิ่มเรคคอร์ดใหม่ลงในตาราง
5. เพื่อเรียนรู้คำสั่ง และสามารถเขียนโปรแกรม PHP ในการแก้ไขเรคคอร์ดในตาราง
6. เพื่อเรียนรู้คำสั่ง และสามารถเขียนโปรแกรม PHP ในการลบเรคคอร์ดในตาราง

จุดประสงค์เชิงพฤติกรรม

1. สามารถเขียนโปรแกรม PHP เพื่อจัดการฐานข้อมูล MySQL ได้

เนื้อหาสาระ

1. การเขียน PHP เชื่อมต่อและเปิดการเชื่อมต่อ MySQL
2. การเขียน PHP เพื่อสร้างฐานข้อมูลและตาราง
3. การเขียน PHP เพื่อเข้าถึงมูลในตาราง

13.1 การเขียน PHP เชื่อมต่อและปิดการเชื่อมต่อ MySQL

เราทราบแล้วว่าก่อนที่จะเข้าใช้งานฐานข้อมูล MySQL ได้ก็จะต้องรู้ชื่อผู้ใช้ (User name) และรหัสผ่าน (Password) เสียก่อน ดังนั้นการที่เราจะเข้าถึงฐานข้อมูล MySQL ได้โดยเขียนโปรแกรมภาษา PHP ก็จะต้องเรียนรู้คำสั่งที่ใช้เชื่อมต่อ MySQL และหลังจากเข้าถึงถึง MySQL เสร็จแล้วเราก็ต้องทำการปิดการเชื่อมต่อ MySQL

คำสั่งเชื่อมต่อ MySQL

รูปแบบ

```
mysql_connect(host name, user name, password)
```

host name	เป็นการระบุชื่อโฮสต์ที่ใช้
user name	เป็นการระบุชื่อผู้ใช้เพื่อเข้าไปใช้งาน MySQL
password	เป็นการระบุรหัสผ่านเพื่อเข้าไปใช้งาน MySQL

ผลลัพธ์ของคำสั่งนี้ ถ้าสามารถเชื่อมต่อ MySQL ได้จะส่งผลลัพธ์เป็นค่าหมายเลขของการเชื่อมต่อ ซึ่งเราจะนำหมายเลขการเชื่อมต่อนี้ไประบุในคำสั่งอื่นต่อไป แต่หาไม่สามารถเชื่อมต่อ MySQL ได้จะส่งผลลัพธ์เป็นค่าเท็จ (False) กลับมาให้

คำสั่งปิดการเชื่อมต่อ MySQL

ถ้าเราต้องการจบการใช้งาน MySQL ก็ให้ใช้คำสั่งปิดการเชื่อมต่อ MySQL ดังนี้

รูปแบบ

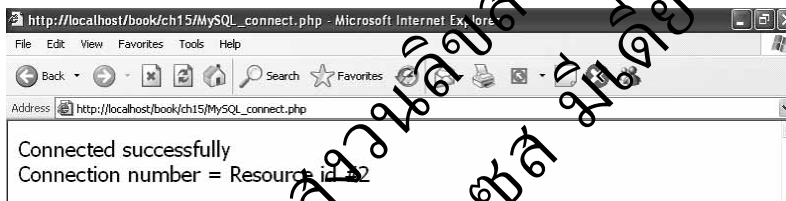
```
mysql_close(หมายเลขการเชื่อมต่อที่ได้จากการเชื่อมต่อ MySQL)
```

นอกจากนั้น ในที่นี้จะแนะนำคำสั่งเพื่อใช้ในการแจ้งความผิดพลาดที่เกิดขึ้นจากการเชื่อมต่อ MySQL คือ คำสั่ง `mysql_error()` โดยจะบอกเหตุผลว่ามีความผิดพลาดอะไรเกิดขึ้นจากการเชื่อมต่อ ดังนั้นเราสามารถนำคำสั่ง `mysql_error()` มาประยุกต์ใช้กับคำสั่ง `mysql_connect()` พิจารณาการใช้งานคำสั่งทั้งหมดจากโปรแกรมต่อไปนี้

MYSQL_connect.php

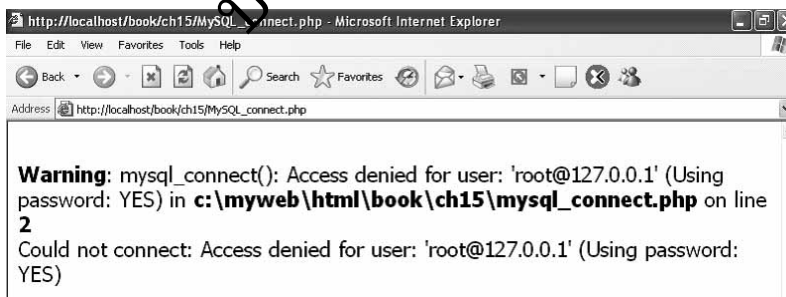
```
<?
$link = mysql_connect("localhost", "root", "1234");
if ($link)
{
    echo "Connected successfully <br>";
    echo "Connection number = $link";
    mysql_close($link);
}
else
{
    die("Could not connect: ' . mysql_error());
}
?>
```

ผลลัพธ์ของโปรแกรมข้างต้น เป็นดังนี้



รูปที่ 13.1 ผลลัพธ์ของโปรแกรม `mysql_connect.php` ในกรณีที่สามารถเชื่อมต่อ MySQL ได้

ในกรณีที่ไม่สามารถเชื่อมต่อ MySQL ได้ เช่นพิมพ์ชื่อผู้ใช้ หรือ รหัสผ่านผิด ผลลัพธ์ของโปรแกรมข้างต้นจะมีลักษณะ ดังนี้



รูปที่ 13.2 ผลลัพธ์ของโปรแกรม `mysql_connect.php` ในกรณีที่ไม่สามารถเชื่อมต่อ MySQL ได้

บทที่

13

13.2 การเขียน PHP เพื่อสร้างฐานข้อมูลและตาราง

โดยปกติเราจะสร้างฐานข้อมูลและตารางจากโปรแกรม PhpMyAdmin (ตามวิธีการที่ได้อธิบายไว้ในบทที่ 12) แต่เราก็สามารถเขียนโปรแกรมภาษา PHP เพื่อสร้างฐานข้อมูลและตารางได้เช่นเดียวกัน คำสั่งที่สำคัญที่ใช้ในการจัดการคือคำสั่ง `mysql_query()` โดยคำสั่งนี้จะรับคำสั่ง SQL แล้วนำไปประมวลผลต่อไปในรูปแบบในการใช้งานเป็นดังนี้

รูปแบบ

`mysql_query` (คำสั่ง SQL)

ผลลัพธ์ของคำสั่งนี้จะส่งค่าจริง (True) เมื่อสามารถทำคำสั่ง SQL ได้ แต่จะส่งค่าเป็นเท็จ (False) เมื่อไม่สามารถทำคำสั่ง SQL ได้ เช่นอาจจะเนื่องมาจากการพิมพ์คำสั่งที่ผิด เป็นต้น พิจารณาโปรแกรมต่อไปนี้ โดยเริ่มจากการสร้างฐานข้อมูลชื่อว่า `testdb` และสร้างตารางชื่อว่า `testtable` ซึ่งรายละเอียดดังนี้

Database Name: testdb

Table Name: testtable

ชื่อฟิลด์	ชนิดข้อมูล	ความยาว	อื่นๆ
id	VARCHAR	3	ให้เป็นคีย์หลัก
name	VARCHAR	30	
age	INT	2	

โปรแกรมในการสร้างฐานข้อมูลและตารางข้างต้นเป็น ดังนี้

MYSQL_create.php

```
<?
$link = mysql_connect("localhost", "root", "1234");
$sql = "Create Database testdb:";
$result = mysql_query($sql);
if ($result)
{
    echo "การสร้างฐานข้อมูลสำเร็จ <br>";
}
else
{
    echo "ไม่สามารถสร้างฐานข้อมูลได้ <br>";
}
```



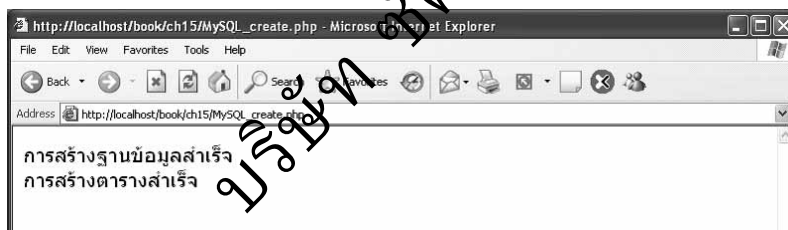
```

$sql = "Use testdb;";
$result = mysql_query($sql);
$sql = "Create Table testtable(id Varchar(3), name Varchar(25), age Int(2),
Primary Key (id));";
$result = mysql_query($sql);
if ($result)
{
    echo "การสร้างตารางสำเร็จ";
}
else
{
    echo "ไม่สามารถตารางได้";
}
mysql_close($link);
?>

```

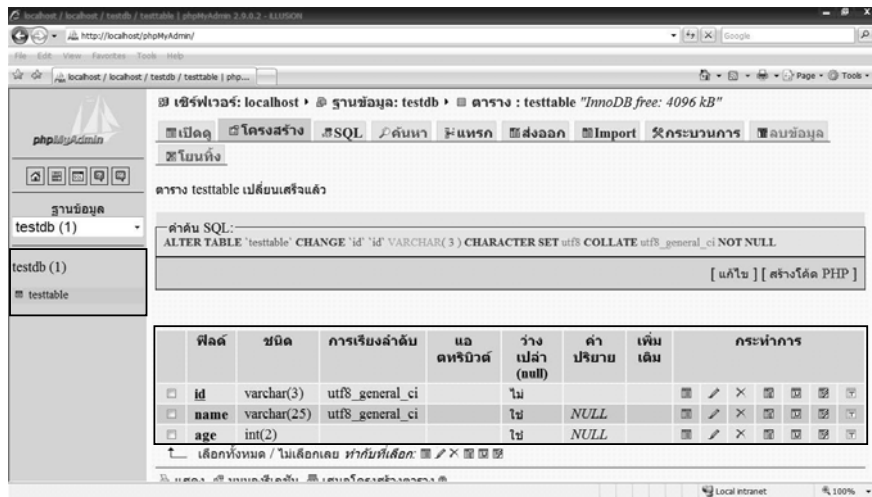
โปรแกรมนี้เป็นการสร้างฐานข้อมูลชื่อว่า testdb โดยใช้คำสั่ง SQL คือ "Create Database testdb;" หลังจากนั้นก็จะทำการสร้างตารางชื่อว่า testtable โดยใช้คำสั่ง "Create Table testtable(id Varchar(3), name Varchar(25), age Int(2), Primary Key (id));" ในที่นี้จะเก็บไฟล์ฐานข้อมูลไว้ที่ C:\MySQL\data

ในกรณีที่ไม่มีปัญหาใดๆ เกิดขึ้นก็จะสร้างฐานข้อมูลและตารางให้ ซึ่งผลลัพธ์ของโปรแกรมนีจะเป็นดังนี้



รูปที่ 13.3 ผลลัพธ์ของโปรแกรม MySQL_create.php

ลองทดสอบว่าโปรแกรมได้สร้างฐานข้อมูลและตารางให้จริงหรือไม่ โดยการเปิดเข้าไปดูด้วยโปรแกรม PhpMyAdmin ซึ่งก็จะได้ผลลัพธ์ ดังนี้



รูปที่ 13.4 ฐานข้อมูล testdb และตาราง testtable

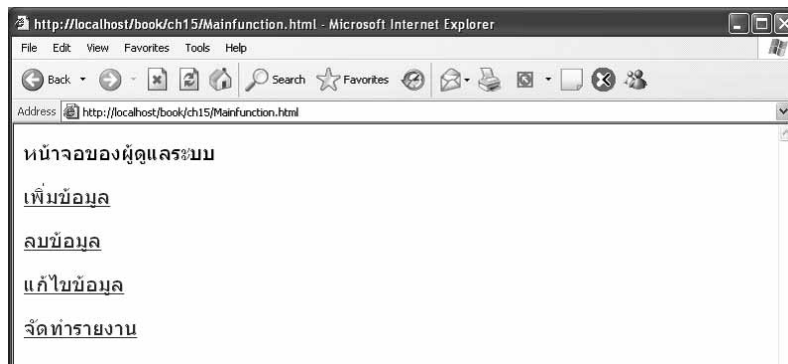
13.3 การเขียน PHP เพื่อเข้าถึงข้อมูลในตาราง

ฟังก์ชันที่สำคัญในการเข้าถึงข้อมูลในตาราง คือ การเพิ่ม ลบ แก้ไข และการแสดงข้อมูล หรือการจัดทำรายงาน ในที่นี้จะใช้ฐานข้อมูล testdb และตาราง testtable ที่สร้างขึ้นในหัวข้อที่ผ่านมา โปรแกรมจะเริ่มจากเว็บหน้าหลักที่รวมลิงค์ฟังก์ชันในการเข้าถึงดังนี้

Mainfunction.html

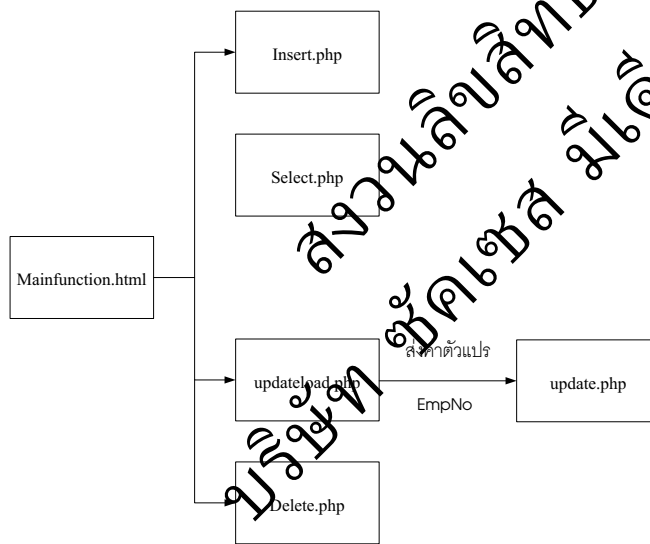
```
<html>
<body>
หน้าจอของผู้ดูแลระบบ<br>
<a href="insert.php">เพิ่มข้อมูล</a><p>
<a href="delete.php">ลบข้อมูล</a><p>
<a href="update/load.php">แก้ไขข้อมูล</a><p>
<a href="select.php">จัดทำรายงาน</a><p>
</body>
</html>
```

ผลลัพธ์ของโปรแกรมข้างต้นเป็น ดังนี้



รูปที่ 13.5 ผลลัพธ์ของโปรแกรม Mainfunction.html

จากรูปที่ 13.5 เป็นหน้าเว็บหลักของการเข้าถึงข้อมูลโดยประกอบไปด้วยฟังก์ชัน การเพิ่มข้อมูล (insert.php), การลบข้อมูล (delete.php), การแก้ไขข้อมูล (updateload.php) และการจัดทำรายงาน (select.php) สามารถสรุปความสัมพันธ์ของโปรแกรมต่างๆ ทั้งหมดได้ ดังนี้



รูปที่ 13.6 แสดงความสัมพันธ์ของโปรแกรมในการเข้าถึงข้อมูลในตาราง

การเพิ่มเรคคอร์ดใหม่ลงในตาราง

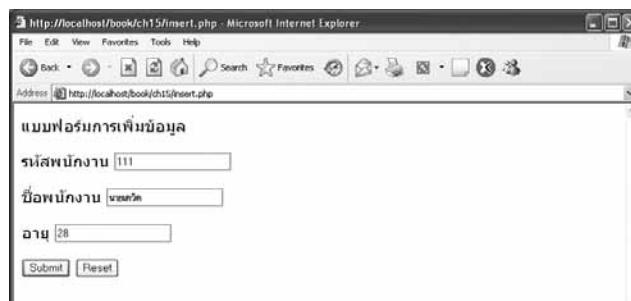
จากรูปที่ 13.5 ถ้าเราคลิกที่ลิงค์ "เพิ่มข้อมูล" จะทำการเรียกโปรแกรม insert.php ซึ่งเป็นแบบฟอร์มสำหรับให้กรอกข้อมูลลงไป โค้ดโปรแกรมเป็นดังนี้

insert.php

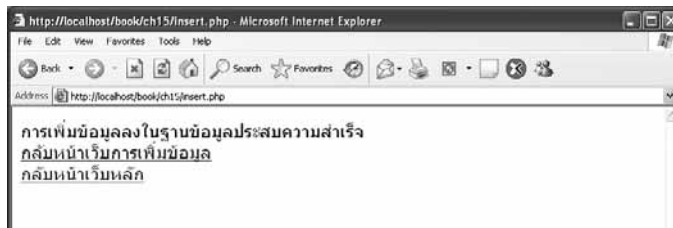
```
<?
if ($send == null) {
?> <form method="post" action="<? echo $PHP_SELF ?>">
แบบฟอร์มการเพิ่มข้อมูล<p>
รหัสพนักงาน <input type="text" name="id"><p>
ชื่อพนักงาน <input type="text" name="name"><p>
อายุ <input type="text" name="age"><p>
<input type="submit" name="send" value="Submit">
<input type="reset" name="cancel" value="Reset">
</form>
<? }
else {
$link = mysql_connect("localhost", "root", "1234");
$sql = "use testdb";
$result = mysql_query($sql);
$sql = "Insert into testtable values($id, '$name', $age)";
$result = mysql_query($sql);
if ($result)
{
echo "การเพิ่มข้อมูลลงในฐานข้อมูลประสบความสำเร็จ<br>";
mysql_close($link);
}
else
{
echo "ไม่สามารถเพิ่มข้อมูลใหม่ลงในฐานข้อมูลได้<br>";
}
echo "<a href=insert.php>กลับหน้าเว็บการเพิ่มข้อมูล</a><br>";
echo "<a href=Mainfunction.html>กลับหน้าเว็บหลัก</a><br>";
}
?>
```

จากโค้ดโปรแกรม insert.php
จะเริ่มจากแบบฟอร์มในการเพิ่มข้อมูล ดังนี้

รูปที่ 13.7 หน้าเว็บแบบฟอร์มในการเพิ่มข้อมูล

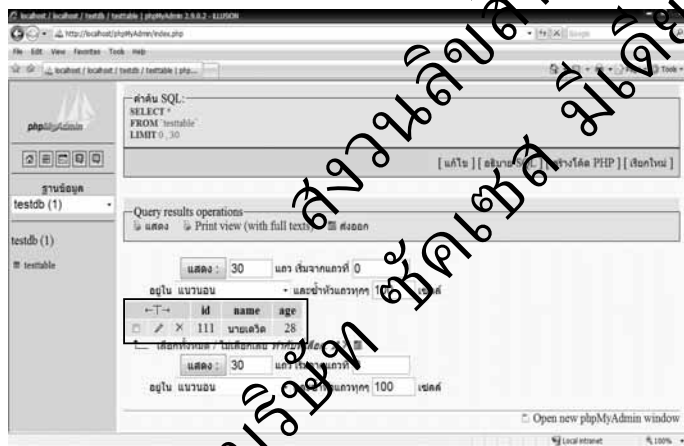


ให้กรอกข้อมูลลงไปในช่วง Text box หลังจากนั้นก็ทำการคลิกที่ปุ่ม "Submit" โปรแกรมจะทำการเพิ่มข้อมูลดังกล่าวลงในตารางฐานข้อมูลให้ โดยเป็นการเพิ่มเรคคอร์ดใหม่ลงไปเ็นตารางฐานข้อมูล ใช้คำสั่ง SQL คือ "Insert into testtable values('sid', 'sname', sage)"; ซึ่งเป็นการเพิ่มเรคคอร์ดลงในตาราง testable โดยนำข้อมูลที่เก็บไว้ในตัวแปร sid, sname และ sage เพิ่มลงไปเ็นฟิลด์ id, name และ age ตามลำดับ ผลลัพธ์ของโปรแกรม เ็นดังนี้



รูปที่ 13.8 ผลลัพธ์หลังจากกรอกข้อมูลและคลิกปุ่ม "Submit"

เราสามารถทดสอบว่าโปรแกรมได้ทำการเพิ่มข้อมูลลงไปเ็นตารางฐานข้อมูลหรือไม่โดยการเปิดดูด้วยโปรแกรม PhpMyAdmin ดังนี้



รูปที่ 13.9 ตาราง testtable มีข้อมูลตามที่กรอกลงไป

การแสดงผลเรคคอร์ดในตาราง

เราสามารถจะเขียนโปรแกรมภาษา PHP เพื่อแสดงว่ามีข้อมูลอะไรบ้างอยู่ภายในตาราง โดยไม่ต้องใช้โปรแกรม PhpMyAdmin จากรูปที่ 13.5 ถ้าเรคลิกที่ลิงค์ "จัดทำรายงาน" จะทำการเรียกโปรแกรมที่ใช้แสดงข้อมูลเรคคอร์ดที่อยู่ภายในตาราง โดยเ็นการเขียนโปรแกรมเราจะใช้คำสั่ง mysql_fetch_array() รูปแบบของคำสั่งนี้เ็น ดังนี้

รูปแบบ

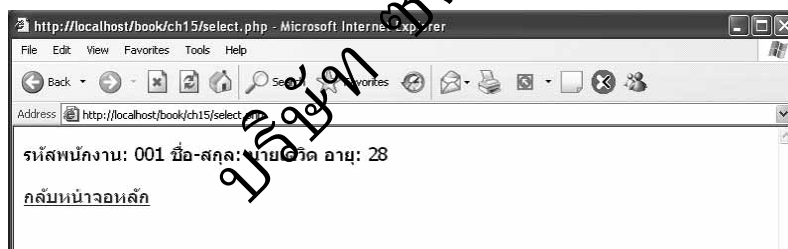
```
mysql_fetch_aray(ผลลัพธ์ของคำสั่ง SQL)
```

ซึ่งคำสั่งนี้จะทำการเรียกดูข้อมูล โดยส่งคืนค่าเป็นข้อมูลชนิดอาร์เรย์ซึ่งมี index เป็นฟิลด์ต่างๆ ของเรคคอร์ด และในการดึงข้อมูลเราจะใช้วิธีการวนรอบดึงข้อมูลออกมาทีละเรคคอร์ด ในที่นี่จะใช้คำสั่ง while ดังตัวอย่างเราจะแสดงเรคคอร์ดทั้งหมดในตาราง โค้ดโปรแกรมทั้งหมดเป็นดังนี้

select.php

```
<?
$link = mysql_connect("localhost", "root", "1234");
$sql = "use testdb";
$result = mysql_query($sql);
$sql = "select * from testtable:";
$result = mysql_query($sql);
while ($dbarr = mysql_fetch_array($result))
{
    echo "รหัสพนักงาน: ".$dbarr['id']." ชื่อ-สกุล: ".$dbarr['namea']
    ".$dbarr['age']."<p>";
}
echo "<a href=Mainfunction.html>กลับหน้าจอหลัก</a>";
mysql_close($link);
?>
```

โปรแกรมนี้จะเรียกใช้คำสั่ง SQL คือ "select * from testtable;" ซึ่งเป็นการดึงทุกเรคคอร์ดจากตาราง testable ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้



รูปที่ 13.10 ผลลัพธ์ของโปรแกรม select.php

การแก้ไขเรคคอร์ดในตาราง

ในการแก้ไขเรคคอร์ดในตารางจะมีขั้นตอนยุ่งยากกว่าการเพิ่มเรคคอร์ด และการแสดงเรคคอร์ดดังที่เคยกล่าวมา ขั้นตอนเริ่มแรกจะต้องเลือกเรคคอร์ดที่ต้องการแก้ไขก่อน โดยแสดงรายละเอียดในแต่ละฟิลด์ในช่อง Text box เพื่อให้สามารถทำการแก้ไขได้ และหลังจากแก้ไขแล้วก็ต้องทำการคลิกที่ปุ่ม "Submit" เพื่อส่งข้อมูลไปแก้ไขภายในตารางฐานข้อมูล จากรูปที่ 13.5 ถ้าทำการคลิกที่ลิงค์ "แก้ไขข้อมูล" จะทำการเรียกโปรแกรมต่อไปนี้

updateload.php

```
<?
    if ($send == null) {
?>

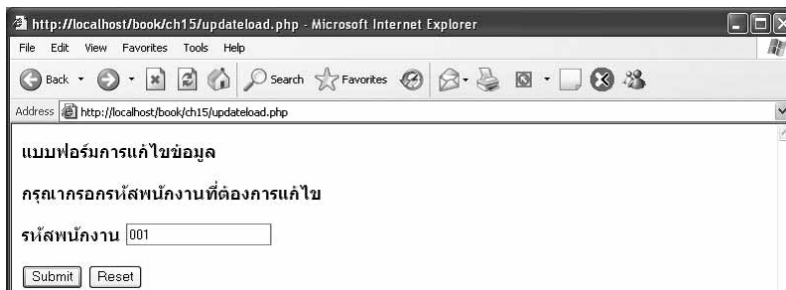
<form method="post" action="<? echo $PHP_SELF ?>">
    แบบฟอร์มการแก้ไขข้อมูล<p>
    กรุณากรอกรหัสพนักงานที่ต้องการแก้ไข<p>
    รหัสพนักงาน <input type="text" name="id_val"><p>
    <input type="submit" name="send" value="Submit">
    <input type="reset" name="cancel" value="Reset">
</form>

<?
}
else
{

$link = mysql_connect("localhost", "root", "1234");
$sql = "use testdb";
$result = mysql_query($sql);
$sql = "select * from testtable where id = '$id_val'";
$result = mysql_query($sql);
echo "<form action=update.php?Ename=$id_val method=post>";
$dbarr = mysql_fetch_array($result);
echo "รหัสพนักงาน: '$id_val.'<br>";
echo "ชื่อพนักงาน: ";
echo "<input type=text name=Ename value=$dbarr[name]> <br>";
echo "อายุ: ";
echo "<input type=text name=EAge value=$dbarr[age]> <p>";
echo "<input type=submit name=Submit value=Submit>";
echo "<input type=reset name=reset value=Cancel>";
echo "</form>";
mysql_close($link);

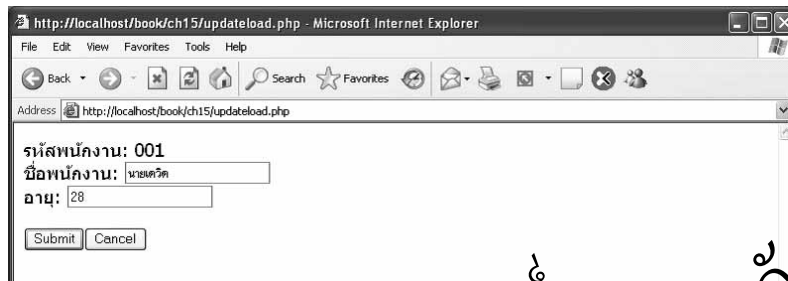
}
?>
```

เมื่อเรียกโปรแกรมนี้ทำงานโปรแกรมจะเริ่มจากการให้ผู้ใช้กรอกรหัสพนักงานที่ต้องการแก้ไข ดังนี้



รูปที่ 13.11 แบบฟอร์มกรอกรหัสพนักงานที่ต้องการแก้ไข

หลังจากกรอกรหัสพนักงาน และคลิกที่ปุ่ม "Submit" แล้วจะแสดงรายละเอียดของพนักงานที่ต้องการแก้ไขขึ้นมาทุกฟิลด์บนช่อง Text box ยกเว้นฟิลด์รหัสพนักงานซึ่งไม่อนุญาตให้แก้ไข โดยการทำงานในส่วนที่ดึงข้อมูลของพนักงานขึ้นมาแสดงนี้จะใช้คำสั่ง SQL คือ "select * from testtable where id = 'sid_val:'" ซึ่งเป็นการดึงทุกเรคคอร์ดจากตาราง testable โดยมีเงื่อนไขว่าให้ดึงเฉพาะเรคคอร์ดที่มีค่าของฟิลด์ id เท่ากับค่าที่เก็บไว้ในตัวแปร \$sid_val เท่านั้น ในตัวอย่างที่จะแสดงให้ดูนี้ตัวแปร \$sid_val ได้เก็บค่า 001 ผลลัพธ์ของตัวอย่างนี้จะเป็น ดังนี้



รูปที่ 13.12 หน้าเว็บแสดงข้อมูลของพนักงานที่ต้องการแก้ไข

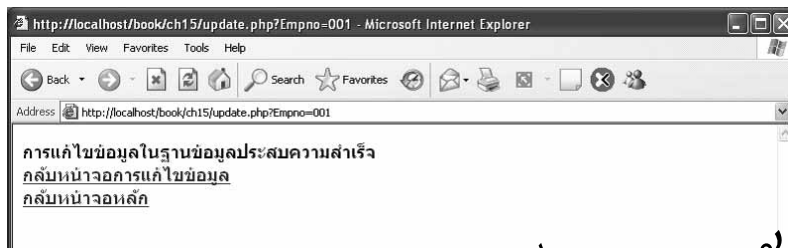
จากรูปที่ 13.12 เราสามารถทำการแก้ไขที่ช่อง text box ของชื่อพนักงานและอายุได้ (ในที่นี้สมมติได้ทำการแก้ไขชื่อพนักงานจาก "นายเดวิด" เป็น "นายไจ้" และแก้ไขอายุจาก 28 เป็น 30) หลังจากนั้นก็ทำการคลิกที่ปุ่ม "Submit" โปรแกรมจะทำการเรียกไฟล์ update.php ซึ่งมีรายละเอียดของโค้ดโปรแกรม ดังนี้

update.php

```
<?
    $link = mysql_connect("localhost", "root", "1234");
    $sql = "use testdb";
    $result = mysql_query($sql);
    $sql = "update testtable set name='$Ename', age='$EAge' where
    id='$Empno' ";
    $result = mysql_query($sql);
    if ($result)
    {
        echo "การแก้ไขข้อมูลในฐานข้อมูลประสบความสำเร็จ<br>";
        mysql_close($link);
    }
    else
    {
        echo "ไม่สามารถแก้ไขข้อมูลในฐานข้อมูลได้<br>";
    }
    echo "<a href=updateload.php>กลับหน้าจอการแก้ไขข้อมูล</a><br>";
    echo "<a href=Mainfunction.html>กลับหน้าจอหลัก</a><br>";
?>
```


โปรแกรมนี้ จะมีการเรียกใช้คำสั่ง SQL คือ "update testtable set name='\\$Ename', age='\\$EAge' where id='\\$Empno'"; โดยจะทำการแก้ไขข้อมูลภายในฟิลด์ต่างๆ ของตาราง testtable ซึ่งในที่นี้จะทำการแก้ไขค่าของฟิลด์ name ให้มีค่าเท่ากับค่าในตัวแปร \\$Ename และ age ให้มีค่าเท่ากับค่าในตัวแปร \\$EAge โดยกำหนดเงื่อนไขในการแก้ไขว่าจะทำการแก้ไขเฉพาะเรคคอร์ดที่มีค่าของฟิลด์ id เท่ากับค่าในตัวแปร \\$Empno เท่านั้น ซึ่งในตัวอย่างที่จะแสดงให้เห็นตัวแปร \\$Empno มีค่า คือ 001

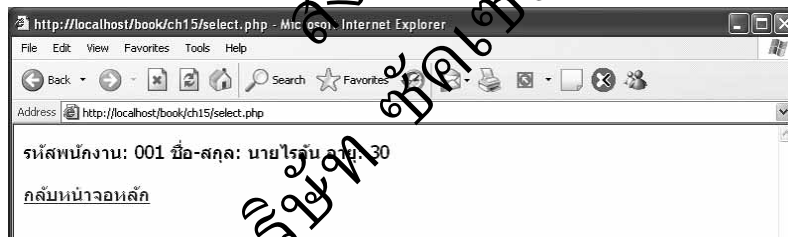
ผลลัพธ์ของโปรแกรมข้างต้นจะทำการแก้ไขข้อมูลตามที่ต้องการ และจะแสดงผลดังต่อไปนี้



รูปที่ 13.13 ผลลัพธ์ของโปรแกรม update.php

ข้อควรสังเกตของรูปที่ 13.13 คือโปรแกรมจะมีคำสั่งให้ส่งค่ารหัสพนักงานที่ต้องการแก้ไขผ่านทาง URL ด้วย เนื่องจากการแก้ไขต้องมีการระบุเงื่อนไขว่าจะต้องแก้ไขเรคคอร์ดใดด้วย

เราอาจจะทดสอบว่าโปรแกรมได้ทำการแก้ไขเรคคอร์ดจริงหรือไม่ โดยการคลิกที่ลิงค์ "จัดทำรายงาน" ตามรูปที่ 13.5 ซึ่งผลลัพธ์จะเป็น ดังนี้



รูปที่ 13.14 แสดงเรคคอร์ดหลังจากทำการแก้ไขแล้ว

การลบเรคคอร์ดในตาราง

ขั้นตอนในการลบเรคคอร์ดในตารางจะเริ่มจากการสร้างแบบฟอร์มให้ผู้ใช้งานกรอกรหัสพนักงานที่ต้องการลบออกไปจากตารางและทำการคลิกที่ปุ่ม "Submit" โปรแกรมก็จะทำการลบเรคคอร์ดที่ต้องการให้ ดังนั้นจากรูปที่ 13.5 ถ้าทำการคลิกที่ลิงค์ "ลบข้อมูล" จะทำการเรียกโปรแกรมต่อไปที่ทำงาน

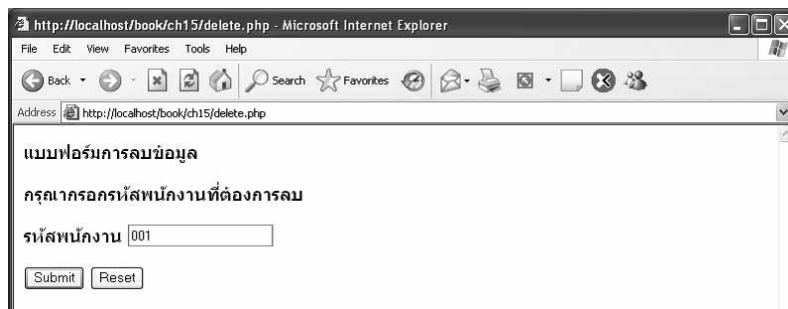
delete.php

```
<?
    if ($send == null) {
?>
    <form method="post" action="<? echo $PHP_SELF ?>">
    แบบฟอร์มการลบข้อมูล<p>
    กรุณากรอกรหัสพนักงานที่ต้องการลบ<p>
    รหัสพนักงาน <input type="text" name="id_val"><p>
    <input type="submit" name="send" value="Submit">
    <input type="reset" name="cancel" value="Reset">
    </form>

<?
}
else
{
    $link = mysql_connect("localhost", "root", "1234");
    $sql = "use testdb";
    $result = mysql_query($sql);
    $sql = "Delete From testtable where id = 'id_val' ";
    $result = mysql_query($sql);
    if ($result)
    {
        echo "การลบข้อมูลในฐานข้อมูลประสบความสำเร็จ<br>";
        mysql_close($link);
    }
    else
    {
        echo "ไม่สามารถลบข้อมูลในฐานข้อมูลได้<br>";
    }
    echo "<a href=delete.php>กลับหน้าจอการลบข้อมูล</a><br>";
    echo "<a href=Mainfunction.html>กลับหน้าจอหลัก</a><br>";
}
?>
```

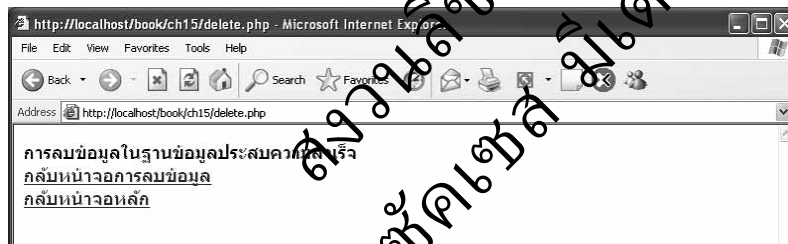
สงวนลิขสิทธิ์
บริษัท ซัคเซส มิดีเอ จำกัด

หน้าเว็บของโปรแกรมจะเริ่มต้น ดังนี้



รูปที่ 13.15 แบบฟอร์มรับรหัสพนักงานที่ต้องการลบ

จากรูปที่ 13.15 ให้ทำการกรอกรหัสพนักงานที่ต้องการลบ หลังจากนั้นทำการคลิกที่ปุ่ม "Submit" โปรแกรมก็จะทำการลบเรคคอร์ดที่ตรงกับรหัสพนักงานที่กรอกลงไป โดยภายในโปรแกรม delete.php นี้มีการใช้คำสั่ง SQL คือ "Delete From testtable where id = \$id_val" ซึ่งเป็นการลบเรคคอร์ดที่อยู่ในตาราง testable โดยมีเงื่อนไขว่าลบเฉพาะเรคคอร์ดที่มีค่าของฟิลด์ id เท่ากับค่าของตัวแปร \$id_val ในตัวอย่างนี้ตัวแปร \$id_val มีค่าเท่ากับ 001 ผลลัพธ์ของโปรแกรมดังนี้



รูปที่ 13.16 หน้าเว็บแสดงผลหลังจากทำการลบเรคคอร์ด

สรุป

ในทางปฏิบัติการเข้าถึงฐานข้อมูลใดๆ จะไม่อนุญาตให้ผู้ใช้งานทั่วไปมีสิทธิเข้าถึงได้ เช่น อนุญาตให้ใช้งานโปรแกรม PhpMyAdmin เนื่องจากผู้ใช้งานทั่วไปอาจจะเข้าไปทำการลบ แก้ไข หรือทำการใดๆ ที่อาจจะส่งผลเสียหายต่อข้อมูล โดยเฉพาะอย่างยิ่งข้อมูลที่เป็นความลับขององค์กร ดังนั้นเราจึงจะต้องทำการเขียนโปรแกรมขึ้นมาเพื่อให้สามารถเข้าถึงข้อมูลในฐานข้อมูลได้ ซึ่งอาจจะเรียกโปรแกรมในลักษณะเช่นนี้ว่า Front-End

โปรแกรมภาษา PHP มีคำสั่งที่ได้จัดเตรียมให้สามารถเข้าถึงข้อมูลในฐานข้อมูลไว้แล้ว เช่น คำสั่งในการเชื่อมต่อ และปิดการเชื่อมต่อฐานข้อมูล, คำสั่งในการจัดการฐานข้อมูล MySQL ซึ่งทำให้สามารถเข้าถึงข้อมูล ทั้ง เพิ่ม ลบ และแก้ไขเรคคอร์ดต่างๆ ภายในตารางฐานข้อมูลได้

บทที่

13

แบบฝึกหัดบทที่ 13

ให้ใช้ฐานข้อมูล Employee ที่สร้างในบทที่ 11 และ 12 สำหรับแบบฝึกหัดในบทที่ 13 นี้

1. ให้สร้างเว็บหน้าหลักที่ประกอบไปด้วยลิงค์ในลักษณะ ดังนี้

หน้าผู้ดูแลระบบ

[เพิ่มข้อมูล](#)

[ลบข้อมูล](#)

[แก้ไขข้อมูล](#)

[จัดทำรายงาน](#)

2. จากข้อที่ 1. เมื่อคลิกที่ลิงค์เพิ่มข้อมูลแล้วให้แสดงหน้าฟอร์มที่ประกอบด้วย Text Field เท่ากับจำนวนของฟิลด์ในตาราง Emp และปุ่มส่งข้อมูล โดยหลังจากกรอกข้อมูลลงใน Text Field ทั้งหมดแล้วให้ทำการคลิกที่ปุ่มส่งข้อมูล ก็จะไปเรียกโปรแกรมภาษา PHP ที่ทำการเพิ่มเรคคอร์ดจากฟอร์มดังกล่าวลงไปในตาราง Emp

3. จากข้อที่ 1. เมื่อคลิกที่ลิงค์ลบข้อมูลแล้วให้แสดงหน้าฟอร์มที่ประกอบไปด้วย Text Field ของรหัสพนักงาน และปุ่มส่งข้อมูล โดยหลังจากกรอกรหัสพนักงานที่ต้องการลบลงไปใน Text Field แล้วให้ทำการคลิกที่ปุ่มส่งข้อมูล ก็จะไปเรียกโปรแกรมภาษา PHP ที่ทำการลบเรคคอร์ดจากตาราง Emp ที่ตรงกับรหัสพนักงานที่กรอกลงไปในฟอร์ม

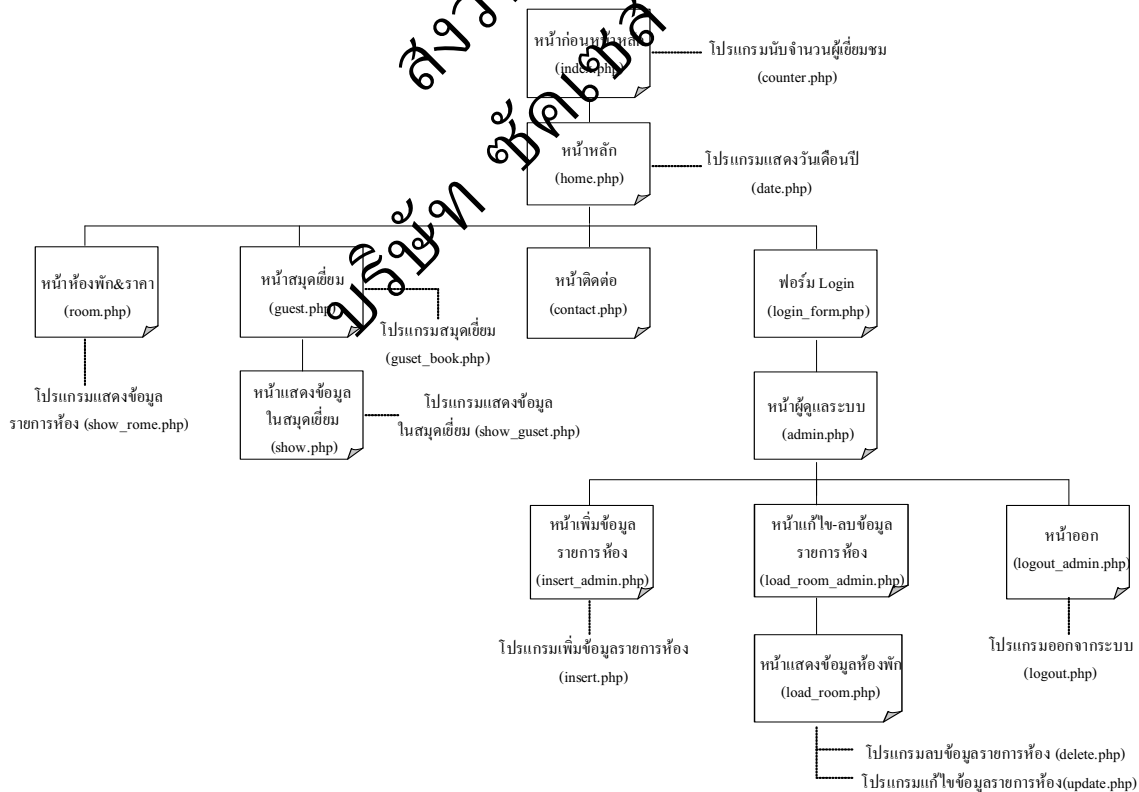
4. จากข้อที่ 1. เมื่อคลิกที่ลิงค์แก้ไขข้อมูลแล้วให้แสดงหน้าฟอร์มที่ประกอบด้วย Text Field ของรหัสพนักงานที่ต้องการแก้ไข และปุ่มส่งข้อมูล หลังจากกรอกรหัสพนักงาน และคลิกปุ่มส่งข้อมูลแล้วก็จะไปเรียกโปรแกรมภาษา PHP ที่ทำการดึงข้อมูลในแต่ละฟิลด์แสดงลงไปใน Text Field โดยดึงเรคคอร์ดเฉพาะรหัสพนักงานที่ต้องการแก้ไข หลังจากนั้นให้ทำการแก้ไขฟิลด์ที่ต้องการใน Text Field แล้วให้ทำการคลิกที่ปุ่มส่งข้อมูล ก็จะไปเรียกโปรแกรมภาษา PHP ที่ทำการแก้ไขเรคคอร์ดตามที่ต้องการ

5. จากข้อที่ 1. เมื่อคลิกที่ลิงค์จัดทำรายงานแล้วให้แสดงรายงานข้อมูลพนักงานทั้งหมด และทุกฟิลด์ (ออกแบบหน้ารายงานให้สวยงามด้วย)

ภาคผนวก ▶ กรณีศึกษา ระบบงานอพาร์ทเมนท์

เนื้อหาในส่วนของภาคผนวกนี้จะเป็นการนำเสนอระบบงานอพาร์ทเมนท์ โดยใช้ความรู้จากที่ศึกษาการเขียนโปรแกรมภาษา PHP มา ซึ่งในระบบงานอพาร์ทเมนท์นี้จะประกอบไปด้วยโปรแกรมย่อยทั้งหมด 9 โปรแกรม คือ

1. โปรแกรมนับจำนวนผู้เยี่ยมชม (Counter) ไฟล์ counter.php
2. โปรแกรมแสดงวัน เดือน ปี ไฟล์ date.php
3. โปรแกรมสมุดเยี่ยม (Guest Book) ไฟล์ guest_book.php และ show_guest.php
4. โปรแกรม Login-Logout ไฟล์ login_form.php และ admin.php
5. โปรแกรมแสดงข้อมูลรายการห้อง ไฟล์ show_room.php
6. โปรแกรมเพิ่มข้อมูลรายการห้อง ไฟล์ insert.php
7. โปรแกรมลบข้อมูลรายการห้อง ไฟล์ delete.php
8. โปรแกรมแก้ไขข้อมูลรายการห้อง ไฟล์ update.php
9. โปรแกรมออกจากระบบ ไฟล์ logout.php



กรณีศึกษา ระบบงานอพาร์ทเมนต์

จากโปรแกรมย่อยทั้ง 8 โปรแกรมเราก็นำมาประยุกต์ใช้กับระบบงานอพาร์ทเมนต์ที่ได้ ดังนี้



รูปที่ 1 เว็บไซต์แรกของระบบงานอพาร์ทเมนต์

หน้าก่อนหน้าหลัก (index.php)

```
<? //1
echo "<body bgcolor=#0099FF>" //2
echo "<center><h1><i>พรชัยอพาร์ทเมนต์</i></h1></center>"; //3
echo "<center><img src=http://localhost/apartment/images/1.JPG width=300 //4
height=230>"
<img src=http://localhost/apartment/images/2.JPG width=300 height=230>
</center><p>"; //5
echo "<center><a href=home.php>เข้าสู่หน้าหลัก</a></center>"; //6
echo "<hr>"; //7
echo "<p><center><img src=http://localhost/apartment/images/digits.gif><b>//8
You are visitor number "; //9
include "counter.php"; //10
?> //11
```

จากโปรแกรม index.php เป็นเว็บไซต์แรกของระบบงานอพาร์ทเมนต์ สังเกตที่บรรทัดที่ 10 เราได้ทำการ include "counter.php": ซึ่งเป็นการเรียกโปรแกรมนับจำนวนผู้เยี่ยมชมให้มาแสดงผลพร้อมต่อจากคำว่า "You are visitor number"

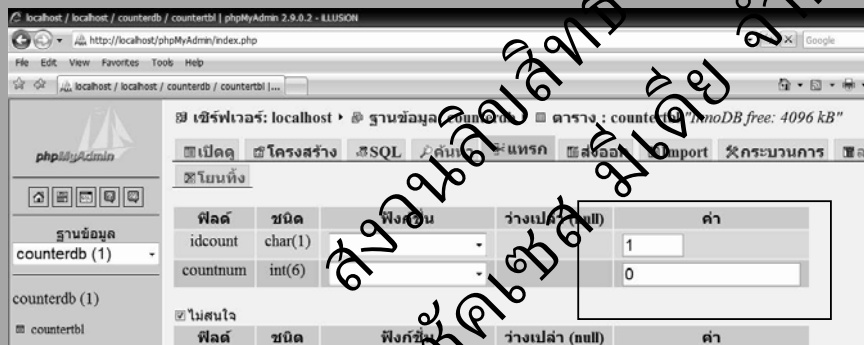
โปรแกรมนับจำนวนผู้เยี่ยมชม

เป็นโปรแกรมที่ใช้เก็บสถิติผู้เยี่ยมชม ทำให้ทราบว่า มีผู้สนใจเข้ามาเยี่ยมชมเว็บไซต์ของเรามากน้อยเพียงใด โดยปกติเราจะนำโปรแกรมนี้ไว้ที่หน้าแรกของเว็บ หลักการทำงาน คือ เริ่มจากการดึงจำนวนผู้เข้ามาภายในเว็บไซต์ ณ.ปัจจุบันจากฐานข้อมูล counterdB ซึ่งมีตาราง countertbl ไว้สำหรับเก็บจำนวนผู้เข้าชม หลังจากนั้นก็ทำการเพิ่มจำนวนผู้เข้ามาภายในเว็บไซต์ขึ้น 1 และสุดท้ายก็ทำการแก้ไขกลับไปยังตาราง countertbl ตามเดิม รายละเอียดของตารางฐานข้อมูล เป็นดังนี้

Database Name: counterdB

Table Name: countertbl		
เป็นตารางเก็บจำนวนผู้เข้ามาภายในเว็บไซต์ ประกอบด้วยฟิลด์และรายละเอียดต่างๆ ดังนี้		
ชื่อฟิลด์	ประเภท(ความยาว)	ความหมาย
idcount	Char(1)	รหัสของจำนวนผู้เข้ามาภายในเว็บไซต์
countnum	INT(6)	จำนวนผู้เข้ามาภายในเว็บไซต์

ในที่นี่ให้ทำการเพิ่มข้อมูลในสถานะตอนเริ่มต้น ดังนี้



โปรแกรม counter.php

```
<?
// กำหนดชื่อโฮสต์, user name และ รหัสผ่าน
$link = mysql_connect("localhost","root","1234");
mysql_query("Use Counterdb:"); // เรียกใช้ฐานข้อมูล Counterdb
$sql = "select * from Countertbl: // เรียกข้อมูลทุกเรคคอร์ดจากตาราง Countertbl
$result = mysql_query($sql);
while ($dbarr = mysql_fetch_array($result))
{
    $pgcount = $dbarr["countnum"] . "<br>";
    // กำหนดจำนวนผู้เยี่ยมชมไว้ในตัวแปร $pgcount
}
$pgcount = $pgcount+1; //เพิ่มค่าตัวแปร $pgcount ขึ้น 1
```

ภาคผนวก

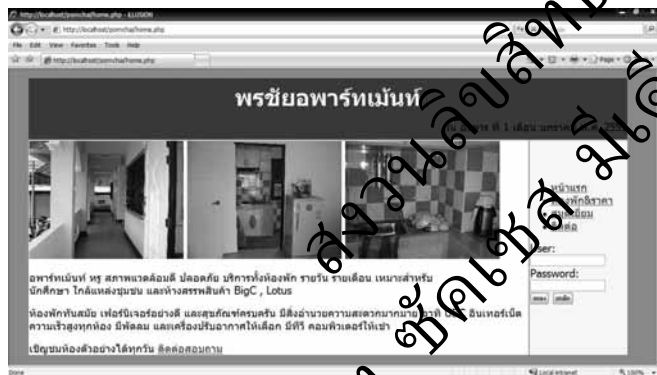


```

$pgcount = "00000" . $pgcount; //เดิม 00000 หน้า
//ตัดสตริงให้เหลือ 6 ตัวอักษร โดยเอาทางขวาของสตริงเป็นหลัก
pgcount = substr($pgcount, -6);
echo "$pgcount ";
// แก้ไขค่าจำนวนผู้เยี่ยมชมที่เป็นปัจจุบันกลับไปในตาราง counterdbl
$sql = "Update countertbl Set countnum = '$pgcount' Where idcount = '1:'";
$result = mysql_query($sql);
?>

```

และจากเว็บหน้า (index.php) ถ้าทำการคลิกที่ลิงค์ "เข้าสู่หน้าหลัก" ก็จะเปิดเว็บหน้าหลักของระบบงานอพาร์ทเมนท์ ดังนี้



รูปที่ 2 เว็บหน้าหลักของระบบงานอพาร์ทเมนท์

หน้าเว็บหลัก (home.php)

```

<? //1
echo "<body bgcolor=#CC9900><table width=1238 height=530 border=0> //2
<tr><td width=23 rowspan=2>&nbsp;&nbsp;&nbsp;</td> //3
<td height=119 colspan=2 bgcolor=#006600><h1><center>
<font color=#FFFFFF>
พรชัยอพาร์ทเมนท์</font></center></h1>"; //4
include "date.php"; //5
echo "</td> //6
<td width=43 rowspan=2>&nbsp;&nbsp;&nbsp;</td> //7
</tr> //8
<tr><td width=1007 bgcolor=#FFFFFF> //9

```



```

<img src=http://localhost/apartment/images/3.JPG width=300 height=230>
<img src=http://localhost/apartment/images/5.JPG width=300 height=230>
<img src=http://localhost/apartment/images/7.JPG width=300height=230><p>//10
อพาร์ทเมนท์ หูรุ สภาพแวดล้อมดี ปลอดภัย บริการทั้งห้องพัก รายวัน รายเดือน เหมาะสำหรับ
<br> นักศึกษา ใกล้แหล่งชุมชน และห้างสรรพสินค้า BigC , Lotus <p> ห้องพักทันสมัย
เฟอร์นิเจอร์อย่างดีและสุขภัณฑ์ครบครัน มีสิ่งอำนวยความสะดวกมากมาย อาทิ UBC อินเทอร์เน็ตความเร็วสูงทุกห้อง มีพัดลมและเครื่องปรับอากาศให้เลือก มีทีวี คอมพิวเตอร์ให้เช่า <p> //11
เชิญชมห้องตัวอย่างได้ทุกวัน <a href=contact.php>ติดต่อสอบถาม</a></td> //12
<td width=190 bgcolor=#FFFF66><ul> //13
<li><a href=home.php>หน้าแรก</a></li> //14
<li><a href=room.php>ห้องพัก&ราคา</a></li> //15
<li><a href=guest.php>สมุดเยี่ยม</a></li> //16
<li><a href=contact.php>ติดต่อ</a></li></ul><p>; //17
include "login_form.php"; //18
echo "</td>
</tr>
</table>" //19
?> //20

```

จากโปรแกรม home.php เป็นเว็บหน้าหลัก โดยหน้าเว็บได้ออกแบบพื้นที่การแสดงผลเป็น 3 ส่วน คือ ส่วนด้านบนใช้สำหรับแสดงชื่ออพาร์ทเมนท์, ส่วนเนื้อหาใช้สำหรับแสดงข้อมูล และส่วนพื้นที่ฝั่งขวาเป็นลิงค์รายการต่างๆ คือ หน้าแรก, ห้องพัก&ราคา, สมุดเยี่ยม และติดต่อ จากเว็บหน้านี้มีจุดที่น่าสนใจ 2 จุด คือ บรรทัดที่ 5 มีการ include "date.php"; ซึ่งเป็นการเรียกโปรแกรมแสดงวัน เดือน ปี มาแสดงบริเวณพื้นที่ด้านบน จุดที่สองคือ บรรทัดที่ 18 มีการ include "login_form.php"; ซึ่งเป็นการเรียกโปรแกรมสร้างฟอร์ม Login

โปรแกรมแสดงวันเดือนปี

เป็นโปรแกรมใช้แสดงวัน เดือน ปีปัจจุบัน โดยสามารถนำไปโชว์ไว้บนหน้าเว็บเพื่อให้ผู้เข้าชมทราบว่า ณ.ขณะนี้เป็นวัน เดือน ปีอะไร หลักการทำงาน คือ ใชฟังก์ชัน Date() ดึง วัน วันที่ เดือน และ ปีปัจจุบันออกมา โดยค่าของวันในสัปดาห์จะนำไปเปรียบเทียบเงือนไขเพื่อเปลี่ยนเป็นชื่อวันภาษาไทย, ค่าของเดือนจะนำไปเปรียบเทียบเงือนไขเพื่อเปลี่ยนเป็นชื่อเดือนภาษาไทย และค่าของปีจะนำไปบวก 543 เพื่อเปลี่ยนเป็นปี พ.ศ.



ภาคผนวก

...

โปรแกรม Date.php

```
<?
$date = date("j"); // ดึงวันที่โดยไม่มีศูนย์นำหน้า
$day = date("D"); // ดึงชื่อวันในสัปดาห์ภาษาอังกฤษเป็นตัวอักษร 3 ตัวแรก
$month = date("M"); // ดึงชื่อเดือนภาษาอังกฤษเป็นตัวอักษร 3 ตัวแรก
$year = date("Y"); // ดึงปี ค.ศ. 4 ตัว
switch($day) // เปรียบเทียบเงื่อนไขของวันในสัปดาห์โดยเปลี่ยนเป็นภาษาไทย
{
    case Sun: $day = "อาทิตย์"; break;
    case Mon: $day = "จันทร์"; break;
    case Tue: $day = "อังคาร"; break;
    case Wed: $day = "พุธ"; break;
    case Thu: $day = "พฤหัสบดี"; break;
    case Fri: $day = "ศุกร์"; break;
    case Sat: $day = "เสาร์"; break;
}
switch($month) // เปรียบเทียบเงื่อนไขของเดือนโดยเปลี่ยนเป็นภาษาไทย
{
    case Jan: $month = "มกราคม"; break;
    case Feb: $month = "กุมภาพันธ์"; break;
    case Mar: $month = "มีนาคม"; break;
    case Apr: $month = "เมษายน"; break;
    case May: $month = "พฤษภาคม"; break;
    case Jun: $month = "มิถุนายน"; break;
    case Jul: $month = "กรกฎาคม"; break;
    case Aug: $month = "สิงหาคม"; break;
    case Sep: $month = "กันยายน"; break;
    case Oct: $month = "ตุลาคม"; break;
    case Nov: $month = "พฤศจิกายน"; break;
    case Dec: $month = "ธันวาคม"; break;
}
$year = $year + 543; // บวก 543 เพื่อเปลี่ยน ค.ศ. เป็น พ.ศ.
// พิมพ์ค่าของวัน วันที่ เดือน ปี ชิดขวา
echo "<p><div align=right> วัน $day ที่ $date เดือน $month พ.ศ. $year </div>";
?>
```

ฟอร์ม Login

เป็นโปรแกรมที่ใช้รักษาความปลอดภัยของระบบ โดยกำหนดชื่อ และรหัสผ่านก่อนที่จะสามารถเข้าไปใช้งานได้ เช่นการนำโปรแกรม Login-Logout นี้ไปใช้กับส่วนของผู้ดูแลระบบ โดยผู้ที่ทราบชื่อ และรหัสผ่านเท่านั้นถึงจะมีสิทธิ์เข้าไปจัดการข้อมูลของผู้ดูแลระบบได้ หลักการทำงาน คือ สร้างฟอร์มรับข้อมูลชื่อ และรหัสผ่าน ซึ่งจะมีการส่งผ่านค่าทั้งสองไปยังอีกโปรแกรมเพื่อทำการตรวจสอบว่าชื่อ และรหัสผ่านถูกต้องหรือไม่ นอกจากนั้นยังทำการลงทะเบียนค่าของชื่อและรหัสผ่านยังตัวแปร session

โปรแกรม login_form.php

```
<? // สร้างฟอร์มรับข้อมูล โดยส่งข้อมูลไปยังไฟล์ admin.php
echo "<form method=post action=admin.php target=_blank>
User: <input type=text name=user><br> // สร้างช่องรับชื่อผู้ใช้
Password: <input type=text name=password><br> // สร้างช่องรับรหัสผ่าน
<input type=submit value=ตกลง name=send> // สร้างปุ่ม submit
<input type=reset value=ยกเลิก name=cancel> // สร้างปุ่ม reset
</form>";

?>
<?
ob_start(); // คำสั่งของ buffer
echo "<body bgcolor=#CC9900 >
<table width=700 height=480 border=1>
<tr>
<td height=119 colspan=2 bgcolor=#006600><h1><center>
<font color=#FFFFFF>ผู้ดูแลระบบ</font></center></h1>
</td>
</tr>";
if (($user == "root" && $password == "123")) // ตรวจสอบชื่อผู้ใช้ และรหัสผ่าน
{
session_register("user"); // ลงทะเบียนตัวแปร session ของชื่อผู้ใช้
session_register("password"); // ลงทะเบียนตัวแปร session ของรหัสผ่าน
echo "<td width=1007 bgcolor=#FFFFFF>
<h3>ยินดีต้อนรับเข้าสู่ส่วนของผู้ดูแลระบบ </h3>
</td>";
}
else // ถ้าชื่อผู้ใช้ และรหัสผ่านไม่ถูกต้อง
{
echo "<td width=1007 bgcolor=#FFFFFF>
คุณกรอก user และ password ผิดกรุณากลับไปกรอกใหม่
</td>";
}
```

ภาพรวม

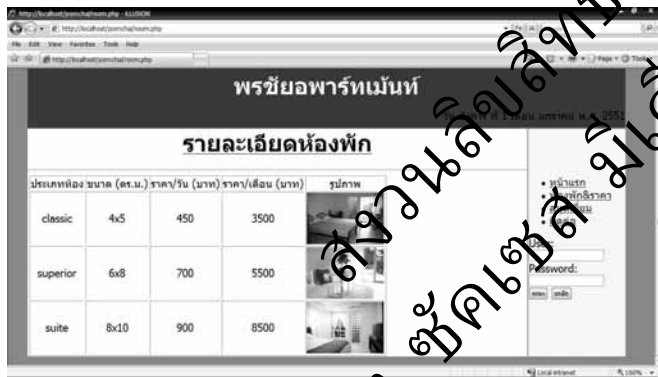
...

```

}
echo "<td width=190 bgcolor=#FFFF66> // ส่วนของแถบรายการลิงค์ภายใน admin
    <ul><li><a href=insert_admin.php>เพิ่มข้อมูลรายการห้อง </a></li>
    <li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลรายการห้อง </a></li>
    <li><a href=logout_admin.php>ออก</a></li>
    </ul>
    </td></table>";
ob_end_flush(); // คำสั่งข้อมูลจาก buffer ไปยัง browser
?>

```

จากเว็บหน้าหลักนี้ (index.php) ถ้าเราทำการคลิกที่ลิงค์ "ห้องพัก&ราคา" ก็จะปรากฏหน้าเว็บ (room.php) ขึ้นมา ดังนี้



รูปที่ 3 เว็บแสดงข้อมูลห้องพัก&ราคา

หน้าห้องพัก&ราคา (room.php)

```

<? //1
echo "<body bgcolor=#CC9900>"; //2
echo "<table width=1238 height=530 border=0> //3
<tr> //4
    <td width=23 rowspan=2>&nbsp;&nbsp;&nbsp;</td> //5
    <td height=119 colspan=2 bgcolor=#006600><h1><center>
    <font color=#FFFFFF>พรชัยอพาร์ทเมนท์</font></center></h1>"; //6
    include "date.php"; //7
echo "</td> //8
<td width=43 rowspan=2>&nbsp;&nbsp;&nbsp;</td> //9
</tr> //10

```

```

<tr> //11
  <td width=1007 bgcolor=#FFFFFF>; //12
  include "show_room.php"; //13
  echo "</td> //14
  <td width=190 bgcolor=#FFFF66><ul> //15
    <li><a href=home.php>หน้าแรก</a></li> //16
    <li><a href=room.php>ห้องพัก&ราคา</a></li> //17
    <li><a href=guest.php>สมุดเยี่ยม</a></li> //18
    <li><a href=contact.php>ติดต่อ</a></li> //19
  </ul><p>; //20
  include "login_form.php"; //21
  echo "</td> //22
</tr> //23
</table>"; //24
?>

```

หน้าเว็บห้องพัก&ราคา นี้ได้ออกแบบให้มีลักษณะเหมือนกับเว็บหน้าหลัก จุดที่สำคัญคือ บรรทัดที่ 13 include "show_room.php"; ซึ่งได้เรียกโปรแกรมแสดงข้อมูลรายการห้อง (show_room.php) เราจะเห็นรายละเอียดของรายการห้องพัก คือ ประเภทห้องพัก, ขนาด, ราคาต่อวัน, ราคาต่อเดือน และรูปภาพ แสดงบริเวณพื้นที่ส่วนเนื้อหา

โปรแกรมแสดงข้อมูลรายการห้อง

เป็นส่วนที่ไปแสดงข้อมูลรายการห้อง โดยประกอบไปด้วยการแสดงผลข้อมูล ประเภทห้องพัก, ขนาดห้อง, ราคาต่อวัน, ราคาต่อเดือน และรูปภาพห้อง ซึ่งโปรแกรมนี้เราสามารถนำไปประยุกต์ดัดแปลงไปแสดงผลข้อมูลอื่นๆ ได้ เช่น แสดงข้อมูลรายการสินค้า, แสดงข้อมูลลูกค้า เป็นต้น หลักการทำงาน คือ ดึงข้อมูลทุกเรคคอร์ดจากฐานข้อมูล room ซึ่งมีตาราง room_data มาแสดงผลโดยใช้แท็ก Table มาจัดข้อมูลให้สวยงาม รายละเอียดของตารางฐานข้อมูล เป็นดังนี้

Database Name: room

Table Name: room_data		
เป็นตารางเก็บรายละเอียดต่างๆ ของห้องพัก ดังนี้		
ชื่อฟิลด์	ประเภท(ความยาว)	ความหมาย
room_id	Int(2), Auto_increment	รหัสประเภทห้องพัก
room_type	VarChar(50)	ประเภทห้องพัก
room_size	VarChar(10)	ขนาดห้องพัก
priceperday	Int(11)	ราคาต่อวัน
pricepermonth	Int(11)	ราคาต่อเดือน
picture	VarChar(20)	รูปภาพห้องพัก

ภาคผนวก

โปรแกรม show_room.php

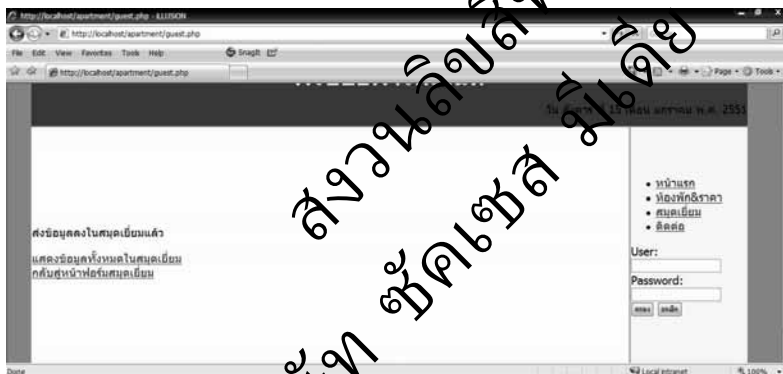
```
<?
$path = "http://localhost/apartment/images/"; //กำหนดพาทที่ใช้เก็บรูปภาพ
echo "<center><h1><u>รายละเอียดห้องพัก</u></h1></center>";
echo "<hr>";
echo "<table border=1>
    <tr>
        <td><center>ประเภทห้อง</center></td>
        <td><center>ขนาด (ตร.ม.)</center></td>
        <td><center>ราคา/วัน (บาท)</center></td>
        <td><center>ราคา/เดือน (บาท)</center></td>
        <td><center>รูปภาพ</center></td>
    </tr>";
$link = mysql_connect("localhost", "root", "1234");
// กำหนดชื่อโฮสต์, user name และ รหัสผ่าน
mysql_query("Use room;"); // เรียกใช้ฐานข้อมูล room
mysql_query("set NAMES tis620;"); // กำหนดการแสดงผลภาษาไทย
$sql = "Select * From room_data;"; // ดึงเรคคอร์ดทั้งหมดจากตาราง room_data
$result = mysql_query($sql);
while ($dbarr = mysql_fetch_array($result)) //วนลูปดึงเรคคอร์ดในตาราง
{
    echo "<tr>
        <td><center>".$dbarr[room_type]."</center></td>
        <td><center>".$dbarr[room_size]."</center></td>
        <td><center>".$dbarr[priceperday]."</center></td>
        <td><center>".$dbarr[pricepermonth]."</center></td>
        <td><center></center></td>
    </tr>";
}
echo "</table>";
mysql_close($link);
?>
```

จากเว็บหน้านี้ (room.php) ถ้าเราคลิกที่ลิงค์ "สมุดเยี่ยม" ก็จะปรากฏหน้าเว็บสมุดเยี่ยม (guest.php) ต่อไปนี้



รูปที่ 4 เว็บแสดงฟอร์มสมุดเยี่ยม

และเมื่อเรากรอกข้อมูลลงในสมุดเยี่ยมแล้วทำการคลิกปุ่ม ตกลง ก็จะปรากฏหน้าจอแจ้งว่าได้ส่งข้อมูลลงในสมุดเยี่ยมแล้ว



รูปที่ 5 เว็บแสดงรายการในสมุดเยี่ยม

หน้าสมุดเยี่ยม (guest.php)

```
<? //1
echo "<body bgcolor=#CC9900><table width=1238 height=530 border=0> //2
<tr> //3
    <td width=23 rowspan=2>&nbsp;</td> //4
    <td height=119 colspan=2 bgcolor=#006600><h1><center>
<font color=#FFFFFF>พรชัยอพาร์ทเมนท์</font></center></h1>"; //5
include "date.php"; //6
echo "</td> //7
<td width=43 rowspan=2>&nbsp;</td> //8
```

ภาพรวม



```

</tr> //9
<tr> //10
    <td width=1007 bgcolor=#FFFFFF>; //11
    include "guest_book.php"; //12
    echo "</td> //13
    <td width=190 bgcolor=#FFFF66><ul> //14
    <li><a href=home.php>หน้าแรก</a></li> //15
    <li><a href=room.php>ห้องพัก&ราคา</a></li> //16
    <li><a href=guest.php>สมุดเยี่ยม</a></li> //17
    <li><a href=contact.php>ติดต่อ</a></li> //18
    </ul><p>; //19
    include "login_form.php"; //20
    echo "</td> //21
</tr> //22
</table>; //23
?> //24

```

จากหน้า guest.php จุดที่สำคัญก็คือโค้ดในบรรทัดที่ 12 include "guset_book.php" ซึ่งเป็นการเรียกโปรแกรมสมุดเยี่ยม (guest_book.php) โดยให้ทำการแสดงผลในบริเวณพื้นที่ส่วนของเนื้อหา

โปรแกรมสมุดเยี่ยม (Guest Book)

เป็นโปรแกรมที่ให้ผู้ใช้เข้ามาชมเว็บไซต์ของเรา โดยจะเข้ามาเขียนคำ แนะนำ ตีชม หรือเขียนคำทักทาย ทำให้ระบบงานของเรามีจุดดึงดูดแก่ผู้เยี่ยมชม หลักการทำงานของมัน คือ สร้างตารางฐานข้อมูลเพื่อเก็บรายการในสมุดเยี่ยม ดังนี้

Table Name: guestdata		
เป็นตารางเก็บรายละเอียดต่างๆ ของคำแนะนำตีชม ประกอบไปด้วยฟิลด์และรายละเอียดต่างๆ ดังนี้		
ชื่อฟิลด์	ประเภท(ความยาว)	ความหมาย
name	VarChar(50)	ชื่อผู้มาเยี่ยม
email	VarChar(50)	อีเมลล์ของผู้มาเยี่ยม
comment	LongText	คำแนะนำ ตีชม
date	datetime	วัน และเวลาที่เข้ามาเยี่ยม
ip	VarChar(40)	หมายเลขไอพีแอดเดรสของเครื่องที่เข้ามาเยี่ยม

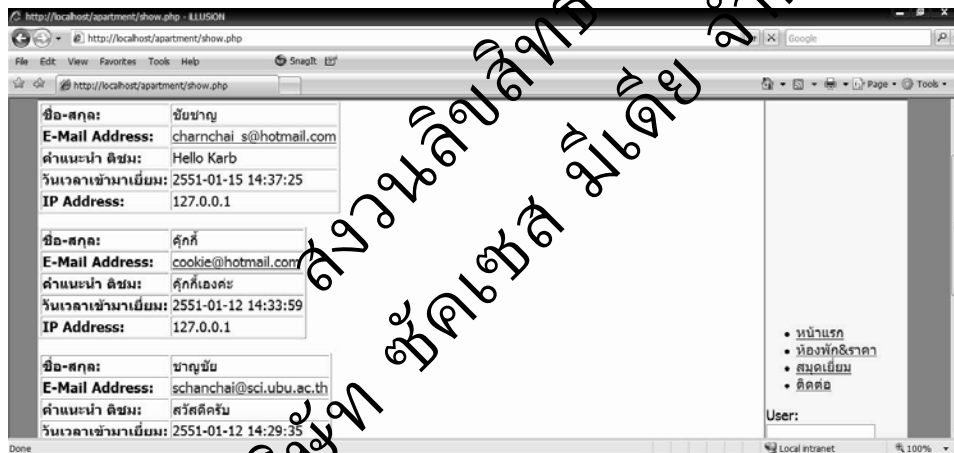
โปรแกรม guest_book.php จะเป็นโปรแกรมที่ทำหน้าที่จัดเก็บรายการที่มีผู้เข้ามากรอกข้อมูลในแบบฟอร์มสมุดเยี่ยม โดยการทำงานจะรับข้อมูล คือ ชื่อ-สกุล, E-Mail และคำตีชมในแบบฟอร์ม รวมทั้งมีการเก็บข้อมูลวันเดือนปี และหมายเลข IP Address ของผู้ใช้ หลังจากนั้นจะทำการเพิ่มลงไปเ็นตารางฐานข้อมูล guestdata


```

    } else
    {
        echo "ไม่สามารถส่งข้อมูลลงในสมุดเยี่ยมได้<p>";
    }
    echo "<a href=show.php>แสดงข้อมูลทั้งหมดในสมุดเยี่ยม</a><br>";
    echo "<a href=guest.php>กลับสู่หน้าฟอร์มสมุดเยี่ยม</a><br>";
}
?>

```

ถ้าเราทำการคลิกที่ลิงค์แสดงข้อมูลทั้งหมดในสมุดเยี่ยม ก็จะปรากฏหน้าเว็บแสดงข้อมูลในสมุดเยี่ยม (show.php) แสดงข้อมูลทั้งหมดที่มีผู้ที่เคยมากรอกรายละเอียดในสมุดเยี่ยม โดยเป็นการแสดงเรคคอร์ดทั้งหมดเรียงตามลำดับวันที่จากล่าสุดเป็นต้นไป ดังนี้



รูปที่ 6 เว็บแสดงรายการในสมุดเยี่ยม (show.php)

หน้าแสดงข้อมูลในสมุดเยี่ยม (show.php)

```

<? //1
echo "<body bgcolor=#CC9900><table width=1238 height=530 border=0> //2
<tr> //3
    <td width=23 rowspan=2>&nbsp;&nbsp;&nbsp;</td> //4
    <td height=119 colspan=2 bgcolor=#006600><h1><center>
    <font color=#FFFFFF>
    พรชัยอพาร์ทเมนท์</font></center></h1>"; //5
include "date.php"; //6

```

```

        echo "</td>" //7
        <td width=43 rowspan=2>&nbsp;</td> //8
    </tr> //9
    <tr> //10
        <td width=1007 bgcolor=#FFFFFF>; //11
        include "show_guest.php"; //12
        echo "</td>" //13
        <td width=190 bgcolor=#FFFF66><ul> //14
            <li><a href=home.php>หน้าแรก</a></li> //15
            <li><a href=room.php>ห้องพัก&ราคา</a></li> //16
            <li><a href=guest.php>สมุดเยี่ยม</a></li> //17
            <li><a href=contact.php>ติดต่อ</a></li> //18
        </ul><p>; //19
        include "login_form.php"; //20
        echo "</td>" //21
    </tr> //22
</table>; //23
?> //24

```

จากหน้าแสดงข้อมูลในสมุดเยี่ยม (show.php) นี้ ให้สังเกตที่บรรทัดที่ 12 จะมีการ include "show_guest.php" ซึ่งเป็นการเรียกโปรแกรม show_guest.php ให้ทำงาน โดยโปรแกรม show_guest.php จะทำการดึงเรคคอร์ด ทั้งหมดในตารางที่เก็บข้อมูลในสมุดเยี่ยมมาแสดงผล

โปรแกรมแสดงข้อมูลในหน้าสมุดเยี่ยม (show_guest.php)

เป็นโปรแกรมที่ทำหน้าที่ดึงเรคคอร์ดทั้งหมดจากตารางฐานข้อมูล guestdata มาแสดงผลในลักษณะเรียงตามลำดับวันที่จากมากไปน้อย ค่ะจากวันที่ปัจจุบันเป็นต้นไป และในการแสดงผลยังได้ใช้แท็ก Table มาช่วยเพื่อจัดข้อมูลให้สวยงาม

```

<?
$link = mysql_connect("localhost", "root", "1234");
// กำหนดชื่อโฮสต์, user name และ รหัสผ่าน
mysql_query("Use guest:"); // เรียกใช้ฐานข้อมูล guest
mysql_query("set NAMES tis620"); // กำหนดการใช้ภาษาไทยในการแสดงผล
$sql = "Select * From guestdata Order By date Desc: ";
// ดึงเรคคอร์ดทั้งหมดจากตาราง guestdata แบบเรียงตามวันที่จากมากไปน้อย
$result = mysql_query($sql);
while ($dbarr = mysql_fetch_array($result)) //วนลูปดึงเรคคอร์ดในตาราง

```

ภาพรวม

```

echo "<Table Border=1>";
echo "<Tr>";
echo "<Td ><b>ชื่อ-สกุล:</b></Td>";
echo "<Td>${dbarr[name]}</Td>";           // แสดงชื่อ-สกุลของผู้เยี่ยม
echo "</Tr>";
echo "<Tr>";
echo "<Td><b> E-Mail Address: </b></Td>";
echo "<Td><a href=mailto:${dbarr[email]}>${dbarr[email]}</Td>";
//แสดงอีเมลของผู้เยี่ยม และสร้างลิงค์
echo "</Tr> ";
echo "<Tr>";
echo "<Td ><b> คำแนะนำ ทิชิม:</b></Td>";
echo "<Td>${dbarr[comment]}</Td>";       // แสดงคำแนะนำทิชมของผู้เยี่ยม
echo "</Tr>";
echo "<Tr>";
echo "<Td><b> วันเวลาเข้ามาเยี่ยม: </b></Td>";
echo "<Td>${dbarr[date]} </Td>";         // แสดงวันและเวลาที่เข้ามาเยี่ยม
echo "</Tr>";
echo "<Tr>";
echo "<Td><b>IP Address: </b></Td>";
echo "<Td>${dbarr[ip]}</Td>";
// แสดงหมายเลขของ address ของเครื่องที่ใช้
echo "</Tr> ";
echo "</Table><P>";
}
mysql_close($link);
?>

```

บริษัท ซัคเซสมีเดีย จำกัด

จากเว็บหน้านี้ ถ้าเราคลิกที่ลิงค์ ติดต่อ ก็จะปรากฏหน้าเว็บติดต่อ (contact.php) ดังนี้

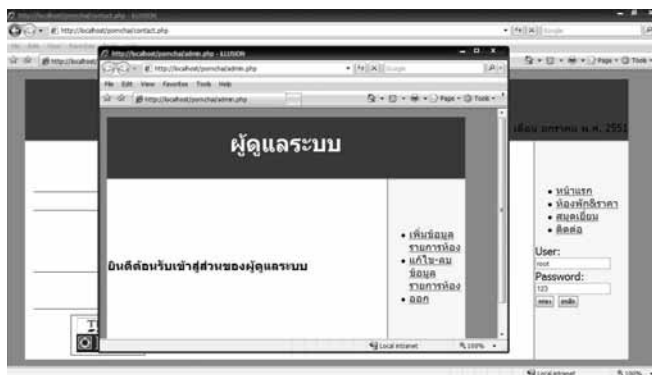


รูปที่ 7 เว็บแสดงข้อมูลการติดต่อ

หน้าติดต่อ (contact.php)

```
<?
echo "<body bgcolor=#CC9900><table width=1238 height=530 border=0>
<tr><td width=23 rowspan=2>&nbsp;</td>
    <td height=119 colspan=2 bgcolor=#006600><h1><center>
    <font color=#FFFFFF>
    พรชัยอพาร์ทเมนท์</font></center></h1>";
include "date.php";
echo "</td>
<td width=43 rowspan=2>&nbsp;</td>
</tr>
<tr><td width=1007 bgcolor=#FFFFFF><img src=http://localhost/apartment/
images/map.gif>
</td>
<td width=190 bgcolor=#FFFF66><ul>
<li><a href=home.php>หน้าแรก</a>
</li>
<li><a href=room.php>ห้องพัก</a></li>
<li><a href=guest.php>สมุดเยี่ยม</a></li>
<li><a href=contact.php>ติดต่อ</a></li>
</ul><p>";
include "login_form.php";
echo "</td>
</tr>
</table>";
?>
```

ลำดับถัดไป ถ้าเราต้องการเข้าไปงานในส่วนของผู้ดูแลระบบก็ให้ทำการกรอกชื่อผู้ใช้ และรหัสผ่าน และคลิกปุ่มตกลง ซึ่งถ้ากรอกชื่อผู้ใช้ และรหัสผ่านถูกต้องก็จะเปิดหน้าต่างบราวเซอร์ใหม่ขึ้นมา ดังต่อไปนี้



รูปที่ 7 เว็บแสดงส่วนของผู้ดูแลระบบ

ภาคผนวก

จากหน้าเว็บในส่วนของผู้ดูแลระบบนี้ (admin.php) ยังคงมีลักษณะของหน้าเว็บเหมือนกับหน้าเว็บที่กล่าวมา คือประกอบไปด้วยพื้นที่ 3 ส่วน คือ ส่วนที่ 1 คือ พื้นที่ด้านบน ในที่นี้ให้แสดงข้อความว่า "ผู้ดูแลระบบ" ส่วนที่ 2 คือส่วนของเนื้อหา ซึ่งใช้สำหรับแสดงผลลัพธ์ และส่วนที่ 3 คือ พื้นที่ฝั่งขวา ในที่นี้กำหนดให้แสดงลิงค์ "เพิ่มข้อมูลรายการห้อง", "แก้ไข-ลบข้อมูลรายการห้อง" และ "ออก" โดยในเว็บหน้านี้ได้ทำการลงทะเบียนตัวแปร session ของชื่อผู้ใช้ และรหัสผ่านแล้ว (ตามที่อธิบายในหัวข้อ โปรแกรม Login-Logout) ดังนั้นจึงเป็นการป้องกันผู้ใช้ที่ทำการจด URL จากเว็บหน้านี้ แล้วภายหลังเข้ามาใช้งานโดยไม่ผ่านการ Login

หน้าผู้ดูแลระบบ (admin.php)

```
<?
ob_start(); // คำสั่งจอง buffer
echo "<body bgcolor=#CC9900 >
<table width=700 height=480 border=0>
<tr>
  <td height=119 colspan=2 bgcolor=#006699><h1><center>
    <font color=#FFFFFF>ผู้ดูแลระบบ</font></center></h1>
  </td>
</tr>";
if (($user == "root") && ($password == "123")) // ตรวจสอบชื่อผู้ใช้ และรหัสผ่าน
{
  session_register("user"); // ลงทะเบียนตัวแปร session ของชื่อผู้ใช้
  session_register("password"); // ลงทะเบียนตัวแปร session ของรหัสผ่าน
  echo "<td width=1007 bgcolor=#FFFFFF>
  <h3>ยินดีต้อนรับเข้าสู่ส่วนของผู้ดูแลระบบ </h3>
  </td>";
}
else // ถ้าชื่อผู้ใช้ และรหัสผ่านไม่ถูกต้อง
{
  echo "<td width=1007 bgcolor=#FFFFFF>
  คุณกรอก user และ password ผิดกรุณากลับไปกรอกใหม่
  </td>";
}

echo "<td width=190 bgcolor=#FFFF66> // ส่วนของแถบรายการลิงค์ภายใน admin
<ul>
<li><a href=insert_admin.php>เพิ่มข้อมูลรายการห้อง </a></li>
<li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลรายการห้อง </a></li>
<li><a href=logout_admin.php>ออก</a></li>
</ul>
</td>
```

บริษัท ซอซเซ็ท มีเดีย จำกัด

```

</table>;
ob_end_flush(); // คำสั่งข้อมูลจาก buffer ไปยัง browser
?>

```

ลำดับถัดไปถ้าเราต้องการเพิ่มรายการห้องพักใหม่ ก็ให้คลิกที่ลิงค์ "เพิ่มข้อมูลรายการห้อง" จะปรากฏหน้าเว็บเพิ่มข้อมูลรายการห้อง (insert_admin.php) ดังนี้



รูปที่ 8 หน้าเว็บการเพิ่มข้อมูลรายการห้อง

หน้าเว็บเพิ่มข้อมูลรายการห้อง (insert_admin.php)

```

<? //1
ob_start(); //2
echo "<body bgcolor=#CC9900 //3
<table width=700 height=480 border=0 //4
<tr><td height=119 colspan=2 bgcolor=#006600><h1><center>
<font color=#FF5555>ผู้ดูแลระบบ</font></center></h1><br></td> //5
</tr> //6
<td width=1007 bgcolor=#FFFFFF>; //7
include "insert.php"; //8
echo "</td> //9
<td width=190 bgcolor=#FFFF66> //10
<ul> //10
<li><a href=insert_admin.php>เพิ่มข้อมูลรายการห้อง </a></li> //11
<li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลรายการห้อง </a></li> //12
<li><a href=logout_admin.php>ออก</a></li></ul> //13
</td>
</table>; //14
ob_end_flush(); //15
?> //16

```

ภาพรวม



ในเว็บการเพิ่มข้อมูลรายการห้อง ผู้ใช้ก็จะต้องกรอกข้อมูลห้องที่ต้องการเพิ่มลงไป ในฟอร์มที่สร้างเตรียมไว้ให้ โดยประกอบไปด้วยช่องรับข้อมูล ประเภทห้อง, ขนาด, ราคา/วัน, ราคา/เดือน ส่วนช่องรับข้อมูล รูปภาพ จะต้องทำการคลิกปุ่ม "Browse" เพื่อเลือกไฟล์รูปภาพห้องพัก หลังจากนั้นให้ทำการคลิกที่ปุ่ม "ตกลง" สำหรับในส่วนของโปรแกรม insert_admin.php จุดที่สำคัญก็คือในบรรทัดที่ 8 include "insert.php" ซึ่งจะทำให้การเรียกโปรแกรมเพิ่มข้อมูลรายการห้องให้ทำการประมวลผลในส่วนนี้

โปรแกรมเพิ่มข้อมูลรายการห้อง

เป็นโปรแกรมใช้เพิ่มห้องพัก ในกรณีที่มีห้องพักประเภทใหม่ขึ้น โดยเราสามารถประยุกต์ไปใช้กับระบบงานอื่น เช่น เพิ่มประเภทสินค้าใหม่ เป็นต้น หลักการทำงาน คือ เริ่มจากการตรวจสอบตัวแปร session ของชื่อและรหัสผ่านว่ามีค่าตรงกันหรือไม่ ถ้าตรงก็สามารถเข้าไปทำการเพิ่มข้อมูลรายการห้องได้ ซึ่งจะมีการสร้างฟอร์มรับรายการห้องพัก และจากฟอร์มรับข้อมูลนี้เมื่อผู้ใช้กรอกรายละเอียดครบแล้ว ทำการส่งข้อมูล รายละเอียดทั้งหมด ก็จะถูกเพิ่มเป็นเรคคอร์ดใหม่ลงในฐานข้อมูล room ภายในตาราง room_data ในส่วนของไฟล์รูปภาพจะมีการ upload และ copy ลงสู่พาดตามที่กำหนดไว้ด้วย โดยเปลี่ยนชื่อไฟล์รูปตามชื่อประเภทของห้อง

โปรแกรม insert.php

```
<?
session_start();
if (($user=="root") && ($password=="123")) // ตรวจสอบชื่อผู้ใช้และรหัสผ่าน
{
if(!$send) {
?>
<form method="post" action="1?PHP_SELF; ?>">
<h3><b>เพิ่มรายการห้องพัก</b></h3><br>
ประเภทห้อง: <input type="text" name="type"><br>
ขนาด: <input type="text" name="size"><br>
ราคา/วัน: <input type="text" name="price_day"><br>
ราคา/เดือน: <input type="text" name="price_month"><br>
รูปภาพ: <input type="file" name="picture"><br>
<input type="submit" value="ตกลง" name="send">
<input type="reset" value="ยกเลิก" name="cancel">
</form>
<?
} else
{
$path_pic="c:/myweb/apartment/images/"; // กำหนดพาทที่เก็บรูปภาพ
if ($picture != none) // ตรวจสอบว่ามีรูปภาพที่ต้องการ upload หรือไม่
{ if (copy ($picture, "$path_pic/$type.jpg"))
```



```

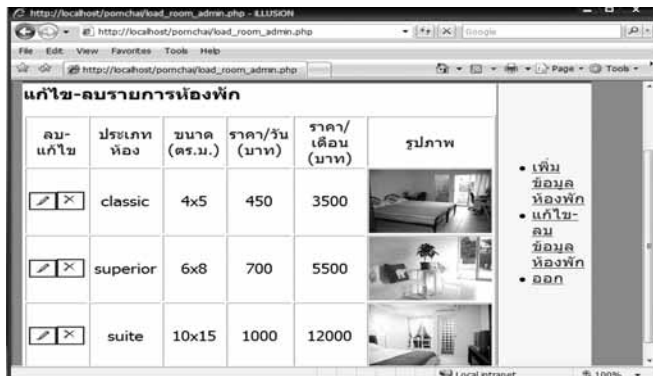
    { $image = $type.".jpg";          // เชื่อมสตริงระหว่างประเภทห้องกับ .jpg
$link = mysql_connect("localhost", "root", "1234");
// กำหนดชื่อโฮสต์, user name และ รหัสผ่าน
mysql_query("Use room:");          // เรียกใช้ฐานข้อมูล room
mysql_query("set NAMES tis620"); // กำหนดการเพิ่มข้อมูลภาษาไทย
// เรียกใช้คำสั่งเพิ่มเรคคอร์ดใหม่ลงในตาราง room_data
$sql = "Insert Into room_data(room_type, room_size, priceperday,
pricepermonth, picture) Values ('$type', '$size', '$price_day', '
$price_month', '$image)";
// คำสั่ง SQL ในการเพิ่มเรคคอร์ดใหม่ตาราง
$result = mysql_query($sql);
if ($result)
{
    echo "เพิ่มรายการห้องพักใหม่แล้ว<p>";
    mysql_close($link);
}
else
{
    echo "ไม่สามารถเพิ่มรายการห้องพักใหม่ได้<p>";
}
}
else
    print ("Error copying file.");
}
else
    echo "No file to be uploaded!";
}
}
else
{
    echo "Invalid to entry the Admin Page";
}
?>

```

ลำดับถัดไปถ้าเราต้องการ แก้ไข หรือลบข้อมูลรายการห้องก็ให้ทำการคลิกที่ลิงค์ "แก้ไข-ลบข้อมูลรายการห้อง" ก็จะปรากฏหน้าเว็บแก้ไข-ลบข้อมูลรายการห้อง (load_room_admin.php) ดังนี้

ภาพรวม





รูปที่ 9 หน้าเว็บการแก้ไข-ลบข้อมูลรายการห้อง

หน้าเว็บแก้ไข-ลบข้อมูลรายการห้อง (load_room_admin.php)

```

<? //1
ob_start(); //2
echo "<body bgcolor=#CC9900 > //3
<table width=700 height=480 border=0> //4
<tr> //5
    <td height=119 colspan=2 bgcolor=#006600><center> //6
        <font color=#FFFFFF>ผู้ดูแลระบบ</font></center></h1><br> //7
    </td>
</tr>
<td width=1007 bgcolor=#FFFFFF> //9
include "load_room.php"; //10
echo "</td> //11
<td width=190 bgcolor=#FFFF66> //12
<ul> //12
    <li><a href=insert_admin.php>เพิ่มข้อมูลห้องพัก </a></li> //13
    <li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลห้องพัก </a></li> //14
    <li><a href=logout_admin.php>ออก</a></li> //15
</ul> //16
</td></table>"; //17
ob_end_flush(); //18
?>

```

พิจารณาโปรแกรมในบรรทัดที่ 10 include "load_room.php" ซึ่งเป็นการเรียกหน้าเว็บแสดงข้อมูลห้องพัก (load_room.php) ให้ทำงานบริเวณส่วนของการแสดงเนื้อหา ตามรูปที่ 9.

หน้าเว็บแสดงข้อมูลห้องพัก (load_room.php)

```
<? //1
session_start(); //2
if (($user=="root") && ($password=="123")) //3
{ //4
$path = "http://localhost/apartment/images/"; //5
echo "<h3><b>แก้ไข-ลบรายการห้องพัก</b></h3>"; //6
echo "<table border=1> //7
<tr> //8
<td><center>ลบ-แก้ไข</center></td> //9
<td><center>ประเภทห้อง</center></td> //10
<td><center>ขนาด (ตร.ม.)</center></td> //11
<td><center>ราคา/วัน (บาท)</center></td> //12
<td><center>ราคา/เดือน (บาท)</center></td> //13
<td><center>รูปภาพ</center></td> //14
</tr>;
$link = mysql_connect("localhost", "root", "1234"); //16
// กำหนดชื่อโฮสต์, user name และ รหัสผ่าน //17
mysql_query("Use room;"); // เรียกใช้ฐานข้อมูล room //18
mysql_query("set NAMES 'tisao'"); // กำหนดการแสดงผลภาษาไทย //19
$sql = "Select * From room_data;"; // ดึงเรคคอร์ดทั้งหมดจากตาราง room_data //20
$result = mysql_query($sql); //21
while ($dbarr = mysql_fetch_array($result)) //วนลูปเพื่อดึงเรคคอร์ดออกมาจากตาราง //22
{ //23
echo "<tr> //24
<td><center><a href=update.php?id=$dbarr[room_id]&room_type=$dbarr[room_type]
&room_size=$dbarr[room_size]&price_day=$dbarr[priceperday]&price_month=$dbarr
[pricepermonth]></a><a href=delete.php?id=$dbarr
[room_id]&room_type=$dbarr[room_type]>
</a></center>
</td> //25
<td><center>$dbarr[room_type]</center></td> //26
<td><center>$dbarr[room_size]</center></td> //27
<td><center>$dbarr[priceperday]</center></td> //28
<td><center>$dbarr[pricepermonth]</center></td> //29
<td><center>
</center></td> //30
</tr>;
```

ภาพรวม



```

    } //31
    echo "</table>";
  } //32
else //33
{ //34
    echo "Invalid to entry the Admin Page"; //35
} //36
?>

```

พิจารณาโปรแกรมในบรรทัดที่ 25 เป็นการเรียกโปรแกรม update.php (โปรแกรมแก้ไขข้อมูลรายการห้อง) และ delete.php (โปรแกรมลบข้อมูลรายการห้อง) โดยโปรแกรม update.php จะมีการส่งค่ารหัสประเภทห้องพัก, ประเภทห้องพัก, ขนาดห้อง, ราคาต่อวัน, ราคาต่อเดือน ผ่านตัวแปร id, room_type, room_size, price_day, price_month ตามลำดับ โดยใช้เครื่องหมาย & เป็นตัวแยกตัวแปรแต่ละตัว และในส่วนของโปรแกรม delete.php จะมีการส่งค่า รหัสประเภทห้องพัก และ รหัสห้องพักผ่านตัวแปร id, room_type ตามลำดับ

โปรแกรมลบข้อมูลรายการห้อง

เป็นโปรแกรมใช้ลบรายการห้องพัก ในกรณีที่ต้องการลบรายการของห้องพักนั้นออกไป เราสามารถประยุกต์โปรแกรมนี้กับระบบอื่นๆ เช่น ลบข้อมูลประเภทของสินค้า, ลบข้อมูลลูกค้า เป็นต้น หลักการทำงาน คือ เริ่มจากการตรวจสอบค่าตัวแปร \$id ของชื่อและรหัสผ่าน ถ้าตรงกันถึงจะสามารถเข้าไปใช้งานการลบข้อมูลได้ ซึ่งการลบข้อมูลรายการห้องพักจะเป็นการลบเรคคอร์ดที่ต้องการออกจากฐานข้อมูล room ภายในตาราง room_data ซึ่งเงื่อนไขในการลบจะใช้ฟิลด์รหัสประเภทห้องพักเป็นตัวที่กำหนดว่าต้องการลบห้องพักไหนออกจากตารางฐานข้อมูล

โปรแกรม delete.php

```

<?
ob_start();
echo "<body bgcolor=#CC9900 >
<table width=700 height=480 border=0>
<tr>
<td height=119 colspan=2 bgcolor=#006600><h1><center>
<font color=#FFFFFF>
ผู้ดูแลระบบ</font></center></h1><br>

```

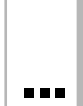
```

</td>
</tr>
<td width=1007 bgcolor=#FFFFFF>;
session_start();
if (($user=="root") && ($password=="123"))
{
    $link = mysql_connect("localhost","root","1234");
    // กำหนดชื่อโฮสต์, user name และ รหัสผ่าน
    mysql_query("Use room:"); // เรียกใช้ฐานข้อมูล room
    mysql_query("set NAMES tis620");
    $sql = "Delete From room_data where room_id = 'sid'";
    $result = mysql_query($sql);
    if ($result)
    {
        echo "ลบรายการห้องพักออกจากฐานข้อมูลแล้ว<p>";
        mysql_close($link);
    }
    else
    {
        echo "ไม่สามารถลบรายการห้องพักออกจากฐานข้อมูลได้<p>";
    }
}
else
{
    echo "Invalid to entry the Admin Page";
}
include "load_room.php";
echo "</td>";
<td width=190 bgcolor=#FFFF66><ul>
    <li><a href=insert_admin.php>เพิ่มข้อมูลรายการห้อง </a></li>
    <li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลรายการห้อง </a></li>
    <li><a href=logout_admin.php>ออก</a></li>
</ul>
</td></table>;
ob_end_flush();
?>

```

สงวนลิขสิทธิ์
บริษัท ชัคเซลมเดีย จำกัด

ภาพรวม



โปรแกรมแก้ไขข้อมูลรายการห้อง

เป็นโปรแกรมแก้ไขเปลี่ยนแปลงรายการห้องพัก ในกรณีที่พาร์ทเมนท์มีการเปลี่ยนแปลงลักษณะของห้องพัก เช่นเปลี่ยนแปลงราคาเช่า หรือแม้กระทั่งเปลี่ยนแปลงรูปภาพที่ต้องการแสดงให้ผู้ใช้เว็บดู เราสามารถประยุกต์โปรแกรมนี้กับระบบงานอื่นๆ เช่น แก้ไขข้อมูลสินค้า, แก้ไขข้อมูลลูกค้า เป็นต้น หลักการทำงาน คือ เริ่มจากการตรวจสอบค่าตัวแปร session ของชื่อและรหัสผ่าน ถ้าตรงกันถึงจะสามารถเข้าไปใช้งานการแก้ไขข้อมูลได้ ซึ่งการแก้ไขข้อมูลรายการห้องพักจะทำการใช้คำสั่ง SQL Update เพื่อแก้ไขข้อมูลในเรคคอร์ดภายในฐานข้อมูล room และตาราง room_data ที่ต้องการ โดยใช้ฟิลด์รหัสประเภทห้องพักเป็นตัวกำหนดประเภทห้องพักที่ต้องการแก้ไข ในส่วนของไฟล์รูปภาพจะมีการ upload และ copy ลงสู่พาทตามที่กำหนดไว้ด้วย โดยเปลี่ยนชื่อไฟล์รูปตามชื่อประเภทของห้อง

โปรแกรม update.php

```
<?
ob_start();
echo "<body bgcolor=#CC9900 >
<table width=700 height=480 border=0>
<tr>
<td height=119 colspan=2 bgcolor=#006600><center>
<font color=#FFFFFF>
ผู้ดูแลระบบ</font></center><hr><br>
</td>
</tr>
<td width=1007 bgcolor=#FFFFFF>";
session_start();
if (($user=="root") && ($password=="123"))
{
if (!$send)
{
echo "<form method=post action=$PHP_SELF>
<input type=hidden name=code value=$sid>
ประเภทห้อง: <input type=text name=type value=$room_type> <br>
ขนาดห้อง: <input type=text name=size value=$room_size> <br>
ราคา/วัน: <input type=text name=priceday value=$price_day> <br>
ราคา/เดือน: <input type=text name=pricemonth value=$price_month> <br>
รูปภาพ: <input type=file name=picture> <br>
<input type=submit value=ตกลง name=send>
<input type=reset value=ยกเลิก name=cancel>
</form>";
}
}
```

```




else
{
    $path_pic="c:/myweb/apartment/images/";
    copy ($picture, "$path_pic/$type.jpg");
    $image = $type.".jpg";
    $link = mysql_connect("localhost","root","1234");
    // กำหนดชื่อโฮสต์, user name และ รหัสผ่าน
    mysql_query("Use room:"); // เรียกใช้ฐานข้อมูล room
    mysql_query("set NAMES tis620");
    $sql = "Update room_data Set room_type = '$type', room_size =
    '$size', priceperday = '$priceday', pricepermonth = '$pricemonth',
    picture = '$image' Where room_id = '$code'";
    $result = mysql_query($sql);
    if ($result)
    {
        echo "แก้ไขรายการห้องพักแล้ว<p>";
        mysql_close($link);
    }
    else
    {
        echo "ไม่สามารถแก้ไขรายการห้องพักได้<p>";
    }
}
}
else
{
    echo "Invalid to enter the Admin Page";
}
include "load_room.php";
echo "</td>";
<td width=190 bgcolor=#FFFF66><ul>
    <li><a href=insert_admin.php>เพิ่มข้อมูลรายการห้อง </a></li>
    <li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลรายการห้อง </a></li>
    <li><a href=logout_admin.php>ออก</a></li>
</ul>
</td></table>";
ob_end_flush();
?>

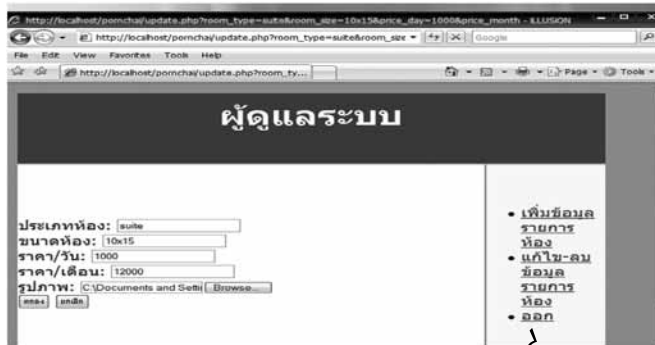
```

สงวนลิขสิทธิ์
 บริษัท ชัคเชล มีเดีย จำกัด

ภาคผนวก

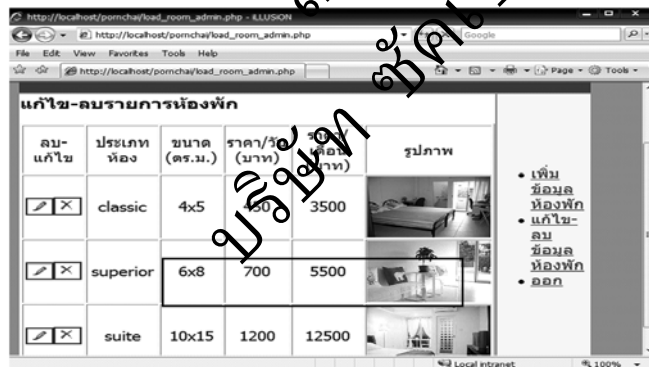


จากหน้าเว็บข้างต้น (รูปที่ 9) ให้สังเกตที่ข้อมูลห้องรายการสุดท้าย ซึ่งเป็นข้อมูลห้องพักที่เราได้ทำการเพิ่มประเภทห้องพักใหม่ไปก่อนหน้านี้ และในการแก้ไข-ลบรายการห้องพักก็เพียงคลิกที่รูป  หน้ารายการห้องที่ต้องการแก้ไข และคลิกที่รูป  หน้ารายการห้องที่ต้องการลบ ยกตัวอย่างเช่น ถ้าต้องการแก้ไขห้องประเภท suite ก็ให้คลิกที่รูป  ด้านหน้ารายการห้อง suite ก็จะปรากฏหน้าเว็บ ดังต่อไปนี้




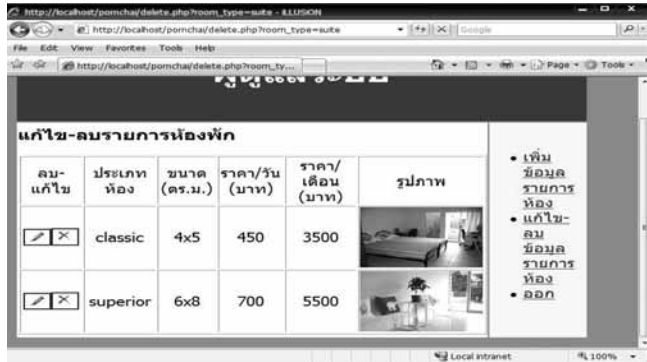
รูปที่ 10 หน้าเว็บการแก้ไขรายการห้องพัก

สมมติต้องการแก้ไขราคาห้องพัก/วันจาก 1000 บาท เป็น 1200 บาท และ ราคาห้องพัก/เดือนจาก 12000 บาท เป็น 12500 บาท ก็ให้ทำการแก้ไขตัวเลขดังกล่าวในช่องรับข้อมูลได้เลย หลังจากนั้นก็ทำการคลิกที่ปุ่ม "ตกลง" โปรแกรมก็จะทำการแก้ไขข้อมูลที่ต้องการ ลองคลิกที่ลิงค์ "แก้ไข-ลบข้อมูลรายการห้อง" อีกครั้งก็จะเห็นการเปลี่ยนแปลงตามที่ต้องการ ดังนี้



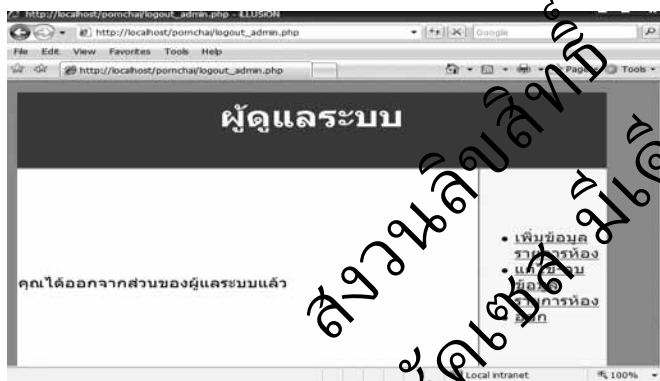
รูปที่ 11 หน้าเว็บดูผลการแก้ไขรายการห้องพัก

ในกรณีที่ต้องการลบรายการห้องพักก็ทำในลักษณะเดียวกัน ยกตัวอย่างต้องการลบรายการห้องประเภท suite ก็ให้คลิกที่รูป  ด้านหน้ารายการห้องประเภท suite สังเกตผลลัพธ์ที่ปรากฏขึ้นก็คือ รายการห้องพักประเภท suite จะหายไป ดังนี้



รูปที่ 12 หน้าเว็บดูผลการลบรายการห้องพัก

และสุดท้ายเมื่อต้องการออกจากในส่วนของผู้ดูแลระบบก็ให้ทำการคลิกที่ลิงค์ "ออก" ก็จะปรากฏหน้าเว็บ ดังนี้



รูปที่ 13 หน้าเว็บเมื่อออกจากส่วนของผู้ดูแลระบบ

โปรแกรมการออกจากส่วนของผู้ดูแลระบบ (logout_admin.php) เป็นดังนี้
โปรแกรมหน้าออก (logout_admin.php)

```
<? //1
ob_start(); //2
echo "<body bgcolor=#CC9900 > //3
<table width=700 height=480 border=0> //4
<tr> //5
    <td height=119 colspan=2 bgcolor=#006600><h1><center>
    <font color=#FFFFFF>ผู้ดูแลระบบ</font></center></h1><br> //6
    </td> //7
</tr> //8
<td width=1007 bgcolor=#FFFFFF>; //9
```

ภาพรวม



```

include "logout.php"; //10
echo "</td>" //11
<td width=190 bgcolor=#FFFF66><ul> //12
    <li><a href=insert_admin.php>เพิ่มข้อมูลรายการห้อง </a></li> //13
    <li><a href=load_room_admin.php>แก้ไข-ลบข้อมูลรายการห้อง </a></li> //14
    <li><a href=logout_admin.php>ออก</a></li> //15
</ul> //16
</td></table>"; //17
ob_end_flush(); //18
?> //19

```

โปรแกรมออกจากระบบ (logout.php)

โปรแกรมนี้จะทำงานโดยเริ่มจากการใช้ฟังก์ชัน session_start() เพื่อกำหนดให้เริ่มต้นการทำงานของ session หลังจากนั้นจะใช้ฟังก์ชัน session_destroy() เพื่อทำลายตัวแปร session ทั้งหมดที่ลงทะเบียนไว้

โปรแกรม logout.php

```

<?
    session_start(); // เริ่มต้นการทำงานของ session
    session_destroy(); // ทำลายตัวแปร session ทั้งหมด
    echo "คุณได้ออกจากส่วนของผู้นี้และระบบแล้ว";
?>

```

บรรณานุกรม

Paul DuBois. **MySQL. 4th Edition.** : Addison-Wesley. 2008

Luke Welling, Laura Thomson. **MySQL Tutorial.** : Sams Publishing. 2004

Christopher Jones, Alison Holloway. **The Underground PHP and Oracle Manual, Release 1.4.** : Oracle. 2007

แหล่งข้อมูล **Appendix I: PHP API Reference.** [ออนไลน์]. เข้าถึงได้จาก : <http://www.kitebird.com/mysql-book/>

แหล่งข้อมูล **mysqli overview.** [ออนไลน์]. เข้าถึงได้จาก : <http://th.php.net/manual/en/mysqli.overview.php>

แหล่งข้อมูล รูปแบบการติดต่อแบบ mysqli [ออนไลน์]. เข้าถึงได้จาก : <http://iambase.allblogthai.com/42>

แหล่งข้อมูล **Implement MySQL-based transactions with a new set of PHP extensions.** [ออนไลน์]. เข้าถึงได้จาก : <http://www.techrepublic.com/article/implement-mysql-based-transactions-with-a-new-set-of-php-extensions/6085922>

<http://www.techrepublic.com/article/implement-mysql-based-transactions-with-a-new-set-of-php-extensions/6085922>

สงวนลิขสิทธิ์
บริษัท ซัคเซส มีเดีย จำกัด