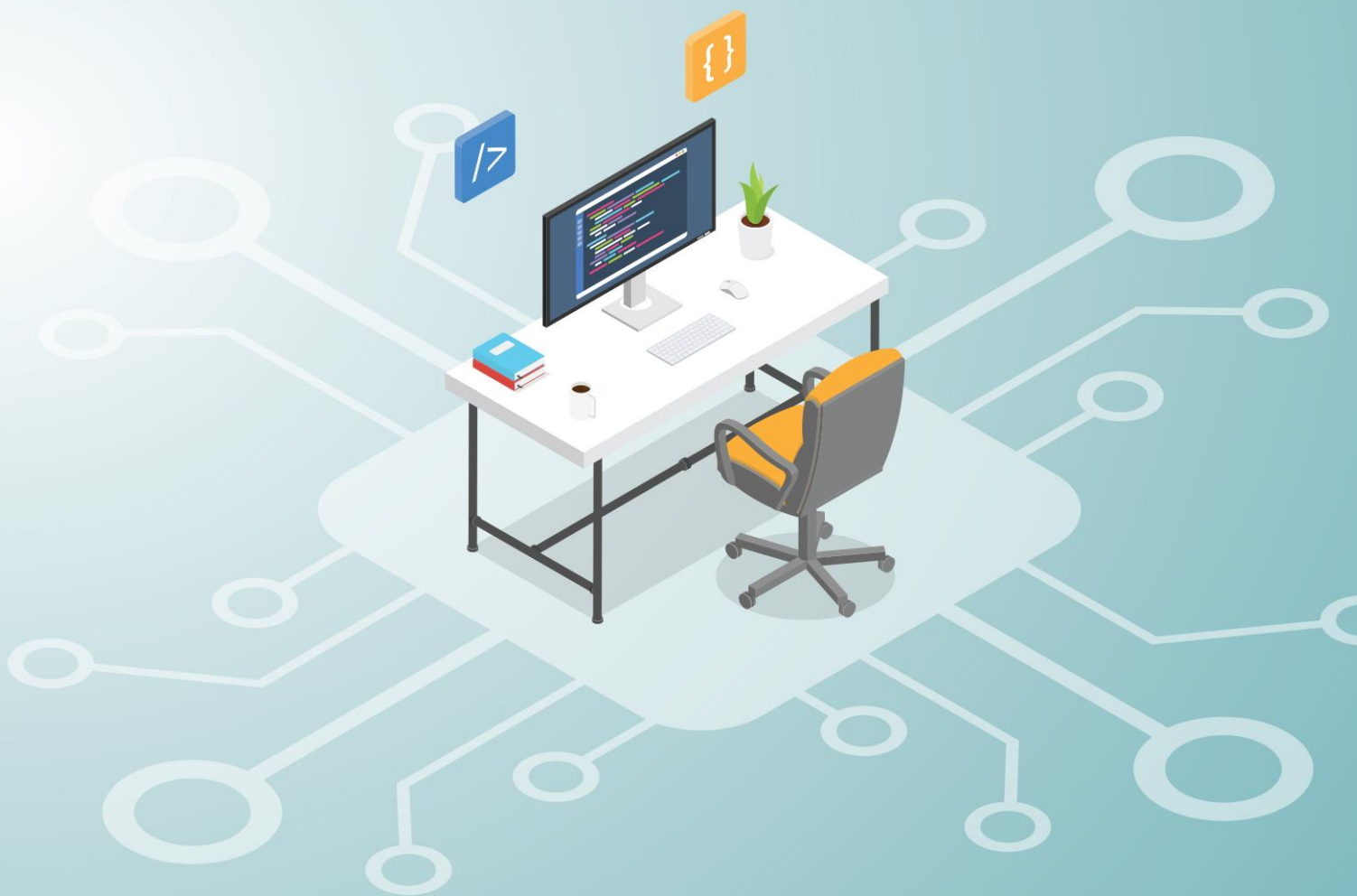


หนังสือคู่มือ
การพัฒนาโปรแกรมบนเว็บ
Web Applications
[สำหรับนักเรียนชั้นมัธยมศึกษาตอนปลาย]



จัดทำโดย
นายเจนรบ โกรธา ตำแหน่ง ครู คศ.1
สำนักงานเขตพื้นที่การศึกษามัธยมศึกษาขอนแก่น
jenropkrotha@kkumail.com | 0908529136



ความรู้เบื้องต้นเกี่ยวกับการเขียน โปรแกรมบนเว็บ

1. ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรม

ลักษณะของการเขียนโปรแกรมจะสามารถแบ่งได้เป็น 2 ประเภท คือ การเขียนโปรแกรมบนวินโดวส์ (Windows-Based Application) และการเขียนโปรแกรมเว็บ (Web-Based Application) ซึ่งในรายวิชานี้จะได้อธิบายถึงเฉพาะการเขียนโปรแกรมบนเว็บเท่านั้น โดยลักษณะที่สำคัญก็คือ การแสดงผลของโปรแกรมจะอยู่บนบราวเซอร์ (Browser) ผู้ใช้สามารถเข้าถึงได้ผ่านทางเครือข่ายอินเทอร์เน็ต ไม่ว่าจะอยู่ที่ไหน หรือจะเข้าใช้งานเมื่อไรก็สามารถใช้งานได้ตลอดเวลา ซึ่งเป็นจุดเด่นที่ทำให้โปรแกรมบนเว็บมีเหนือกว่าโปรแกรมบนวินโดวส์ ระบบงานทางธุรกิจก็เริ่มหันมาพัฒนาในลักษณะโปรแกรมเว็บมากขึ้นอันเนื่องมาจากข้อดีดังกล่าวนั่นเอง เนื้อหาในบทนี้จะกล่าวถึงความรู้พื้นฐานในการเขียนโปรแกรมบนอินเทอร์เน็ตเพื่อทำให้ผู้อ่านได้เข้าใจลักษณะทางงานของโปรแกรมก่อนที่จะเริ่มเข้าสู่เนื้อหาของการเขียนโปรแกรม

1.1 หลักการทำงานของ WWW

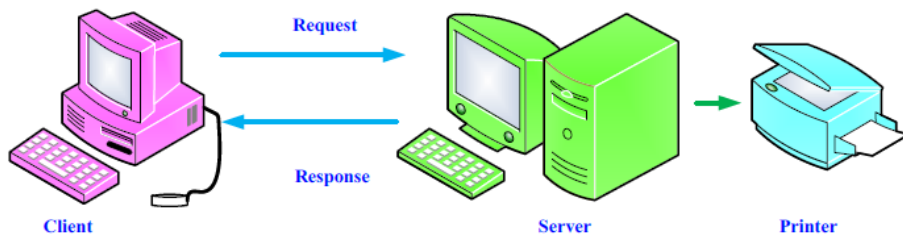
การทำงาน World Wide Web หรือ WWW จะมีลักษณะเช่นเดียวกับการทำงานในลักษณะไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) คือ มีลักษณะของการเชื่อมต่อของเครื่องผู้ให้บริการ (Server) และ เครื่องผู้ให้บริการ (client) พิจารณารูปต่อไปนี้



รูปที่ 1.1 แสดงการทำงานของ Client-Server

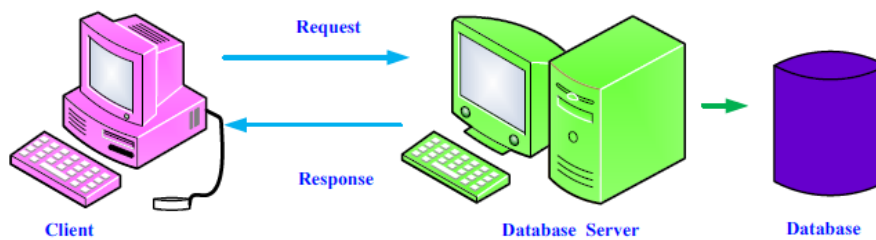
การทำงานเริ่มจากเครื่องผู้ขอใช้บริการ (Client) ซึ่งอาจจะเรียกว่าเป็นเครื่องลูกที่ทำงานร้องขอ (Request) ใช้บริการจากเครื่องผู้ให้บริการ (Server) ซึ่งอาจจะเรียกว่าเป็นเครื่องแม่ หลังจากเครื่องผู้ให้บริการทำการจัดเตรียมข้อมูลหรือบริการตามที่เครื่องผู้ขอใช้บริการได้ร้องขอมาก็จะทำการตอบกลับ (Response) คืนไปยังเครื่องของผู้ขอใช้บริการ โดยปกติเครื่องแม่จะมีอยู่เพียงเครื่องเดียว ในขณะที่เครื่องลูกอาจจะมีได้หลายๆ เครื่องและเครื่องลูกหลายๆ เครื่องนี้ก็อาจจะเข้ามาขอใช้บริการจากเครื่องแม่พร้อมกันก็ได้

ในสำนักงานต่าง ๆ ใต้นา ประโยชน์ของการทำงานในลักษณะ Client-Server นี้ไปใช้จัดการงานต่าง ๆ ภายในสำนักงานพิจารณารูปต่อไปนี้



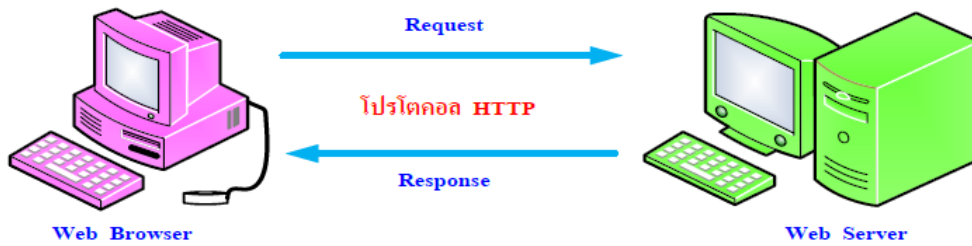
เครื่องแม่ได้ทำการเชื่อมต่อกับเครื่องพิมพ์ (Printer) ดังนั้นเครื่องแม่นี้เราจะเรียกว่า Print Server คือ เป็นเครื่องที่ให้บริการงานทางด้านการพิมพ์ กระบวนการทำงานจะเริ่มต้น จากเครื่องลูกที่ต้องการพิมพ์ เอกสารจึงได้ทำการสั่งพิมพ์จากเครื่องลูกดังกล่าว ข้อมูลที่ต้องการพิมพ์ก็จะถูกส่งมายังเครื่อง Print Server ซึ่งก็จะทำการพิมพ์ข้อมูลออกทางเครื่องพิมพ์ ข้อดีของการเชื่อมต่อในลักษณะนี้ก็คือ ในสำนักงานนั้นก็เพียงแค่ จัดหาเครื่องพิมพ์ 1 เครื่องเชื่อมต่อกับเครื่องคอมพิวเตอร์ที่จะใช้เป็น Print Server เมื่อพนักงานภายในบริษัท ต้องการสั่งพิมพ์เอกสารใด ๆ ก็สามารถสั่งงานที่เครื่องลูกได้เลย ประหยัดงบประมาณที่จะต้องทำการจัดหา เครื่องพิมพ์ให้กับพนักงานทุกคน

พิจารณาตัวอย่างของการทำงานแบบ Client-Server อีกตัวอย่าง ดังรูปต่อไปนี้



ก็เป็นอีกตัวอย่างของการทำงานในลักษณะ Client-Server โดยเครื่องแม่ทำหน้าที่จัดเก็บข้อมูลต่าง ๆ ทั้งหมดของบริษัทไว้ในฐานข้อมูล ดังนั้นเครื่องแม่นี้เราก็จะเรียกว่า Database Server การทำงานก็จะเริ่มจาก เครื่องลูกเช่นเดียวกัน คือ เครื่องลูกอาจจะทำการเพิ่ม ลบ แก้ไข หรือดึงข้อมูลซึ่งจัดเก็บไว้ในฐานข้อมูล ข้อดี ของการเชื่อมต่อแบบนี้ก็คือ การจัดการข้อมูลต่างๆซึ่งถูกจัดเก็บไว้เพียงที่เดียวทำให้ข้อมูลมีความเป็นหนึ่งเดียว (Uniqueness) คือข้อมูลไม่กระจัด กระจายและมีความถูกต้อง

จากทั้ง 2 ตัวอย่างของการทำงานของ Client-Server ลำดับถัดไปจะกล่าวถึงลักษณะของการทำงาน ของ WWW ซึ่งก็จะมีลักษณะการทำงานเหมือนกับ 2 ตัวอย่างที่ได้อธิบายไปแล้ว พิจารณารูปต่อไปนี้



เครื่องแม่ในที่นี้จะให้บริการต่างๆ ที่เกี่ยวกับเว็บทั้งหมด ไฟล์เว็บเพจ รูปภาพ หรือ โปรแกรมเว็บต่างๆ จะถูกจัดเก็บไว้ในเครื่องนี้ ซึ่งเราจะเรียกเครื่องนี้ว่า Web Server ส่วนเครื่องลูกซึ่งเป็นเครื่องที่ใช้บริการเรา จะเรียกว่า Client เนื่องจากการแสดงผลจะถูกแสดงบนบราวเซอร์ เช่น Internet Explore หรือ URL (Universal Resource Locator) จากโปรแกรมบราวเซอร์ ซึ่งข้อมูลจะถูกกระทำผ่านโปรโตคอล HTTP (Hypertext Transfer Protocol) เครื่อง Web Server เมื่อได้รับคำร้องขอก็จะทำการจัดส่งหน้าเว็บเพจนั้น ผ่านไปทาง Web Browser

1.2 ประเภทของโปรแกรมบนเว็บ

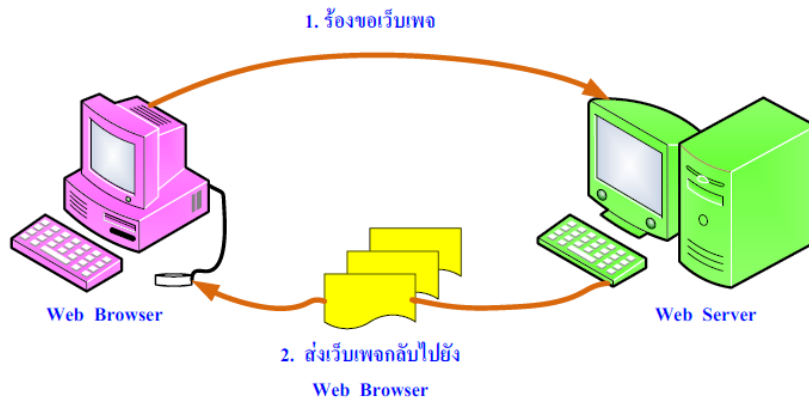
ในการเขียนโปรแกรมบนเว็บสามารถที่จะแบ่งลักษณะการทำงานของโปรแกรมได้เป็น 2 ประเภท คือ

1. Static Programming
2. Dynamic Programming

Static Programming เป็นลักษณะของโปรแกรมบนเว็บที่ไม่ค่อยมีการเปลี่ยนแปลง เช่น เว็บประวัติส่วนตัว, เว็บนำเสนอประวัติและโครงสร้างขององค์กร เป็นต้น ลักษณะเว็บประเภทนี้ เมื่อผู้พัฒนาเว็บสร้างเว็บขึ้นมาแล้วหากต้องการที่จะทำการแก้ไขข้อมูลบางอย่างนั้นก็ต้องใช้ โปรแกรมในการสร้างเว็บเพจ เช่น Adobe Dreamweaver Microsoft FrontPage เป็นต้น เปิดไฟล์ของหน้าเว็บนั้นแล้วจึงทำการแก้ไขข้อมูลที่ต้องการ เมื่อเสร็จแล้วก็ต้องทำการบันทึกไฟล์ดังกล่าวและทำการอัปโหลดไฟล์ขึ้นไปเก็บไว้ที่ Web Server จะเห็นว่าลักษณะของโปรแกรมบนเว็บประเภทนี้จะไม่มีความยืดหยุ่นในการจัดการ และสร้างความยุ่งยากให้แก่ผู้พัฒนาเว็บเพจด้วย

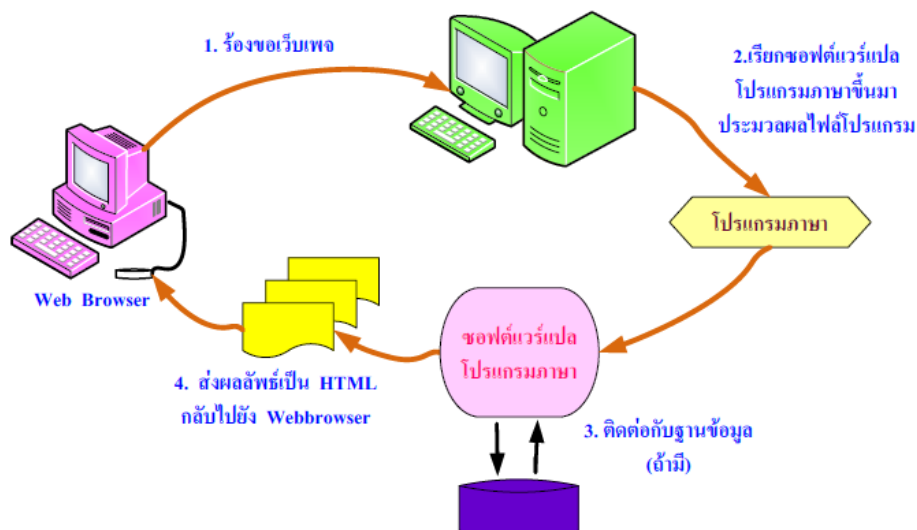
ในการพัฒนาโปรแกรมบนเว็บประเภทนี้ ผู้พัฒนาโปรแกรมเพียงแค่จัดหาเครื่องคอมพิวเตอร์และทำการติดตั้งซอฟต์แวร์ที่จะทำให้เครื่องคอมพิวเตอร์เครื่องนี้เป็นเครื่องแม่ (Web Server) เท่านั้น และในการพัฒนาโปรแกรมบนเว็บ ผู้พัฒนา也不需要มีความรู้ในการเขียนโปรแกรมก็ได้ เพียงแค่สามารถเขียนเว็บเพจโดยใช้โปรแกรมที่ใช้สร้างเว็บเพจก็เพียงพอแล้ว

ลักษณะการทำงานของโปรแกรมบนเว็บประเภทนี้มีลักษณะดังรูปต่อไปนี้



Dynamic Programming เป็นลักษณะของโปรแกรมบนเว็บที่เกิดขึ้นมาเพื่อแก้ปัญหาคความไม่ยืดหยุ่นในการจัดการข้อมูลของ Static Programming โดยเหมาะสมสำหรับเว็บที่ต้องมีการเปลี่ยนแปลงข้อมูลอยู่บ่อยครั้งหรือเว็บที่มีการจัดเก็บข้อมูล จำนวนมากได้ ภายในฐานข้อมูล เช่นเว็บหนังสือพิมพ์, เว็บแสดงรายละเอียดของสินค้า เป็นต้นในการพัฒนาโปรแกรมบนเว็บประเภทนี้จะต้องอาศัยผู้พัฒนาโปรแกรมที่มีความรู้ความสามารถในการเขียนโปรแกรม นอกจากนั้นในส่วนของซอฟต์แวร์ที่ต้องติดตั้งก็ประกอบไปด้วยซอฟต์แวร์ที่ใช้ในการทำ ให้เครื่องคอมพิวเตอร์มีความสามารถเป็นเครื่อง Web Server, ซอฟต์แวร์ที่ใช้ในการแปลโปรแกรมภาษาและฐานข้อมูล

พิจารณาลักษณะการทำงานของโปรแกรมเว็บประเภทนี้ ดังรูปต่อไปนี้



โปรแกรมก็จะเริ่มกระบวนการทำงานจากเครื่องลูกทำการร้องขอเว็บเพจ ซึ่งเว็บเพจที่ร้องขอได้มีการเขียนโปรแกรมบนเว็บในลักษณะ Dynamic Programming ได้ ดังนั้นที่เครื่องแม่ก็จะทำการเรียกซอฟต์แวร์

และโปรแกรมภาษาขึ้นมาเพื่อแปลโปรแกรมภาษา (Programming Language) ให้เป็นภาษาเครื่อง (Machine Language) เนื่องจากเครื่องคอมพิวเตอร์จะเข้าใจเฉพาะภาษาเครื่องเท่านั้น ถ้าโปรแกรมที่เขียนขึ้นมีคำสั่งที่ทำการจัดการฐานข้อมูลก็จะทำการ เพิ่ม ลบ แก้ไข หรือดึงข้อมูลจากฐานข้อมูล ซึ่งโดยปกติตัวฐานข้อมูลนี้จะอยู่ภายในเครื่องแม่หรืออาจจะแยกออกมาเป็นเครื่องคอมพิวเตอร์อีกเครื่องก็ได้หลังจากนั้นขั้นตอนสุดท้ายหลังจากเครื่องแม่ทำการประมวลผลเสร็จเรียบร้อยแล้วก็จะทำการส่งข้อมูลผลลัพธ์กลับในลักษณะ HTML กลับไปยังเครื่องลูก หรือเครื่องที่ทำการร้องขอของมันเอง

1.3 รู้จักกับ PHP

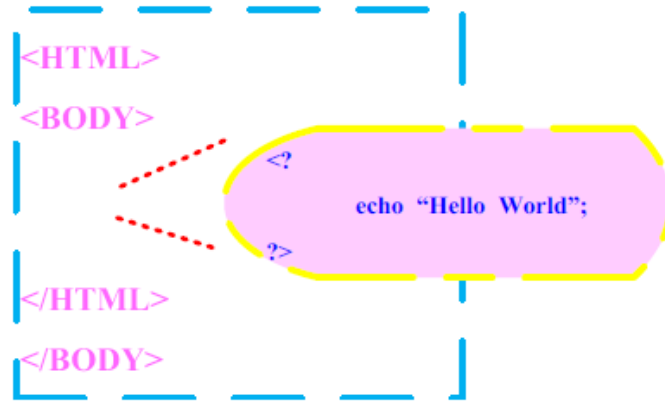
ในปัจจุบันการเขียนโปรแกรมบนอินเทอร์เน็ตมีโปรแกรมภาษาให้เลือกจำนวนมาก อาทิเช่น PHP, ASP, JSP, Java เป็นต้น แต่สำหรับในที่นี่จะเลือกใช้โปรแกรมภาษา PHP ซึ่งย่อมาจากคำว่า Personal Home Page ข้อดีของภาษา PHP มีด้วยกันหลายข้อ คือ

- PHP เป็นโปรแกรมภาษาที่สามารถดาวน์โหลดได้ฟรีไม่มีปัญหาเรื่องลิขสิทธิ์
- PHP มีการเปลี่ยนแปลงภาษาและทำการประมวลผลได้อย่างรวดเร็ว
- PHP สามารถทำงานได้ทั้งบนระบบปฏิบัติการ Windows, Unix, Linux และ Macintosh
- สามารถกล่าวได้ว่า PHP เป็นโปรแกรมภาษาที่พัฒนาระบบงานบนเว็บที่ได้รับความนิยมมากที่สุดใน

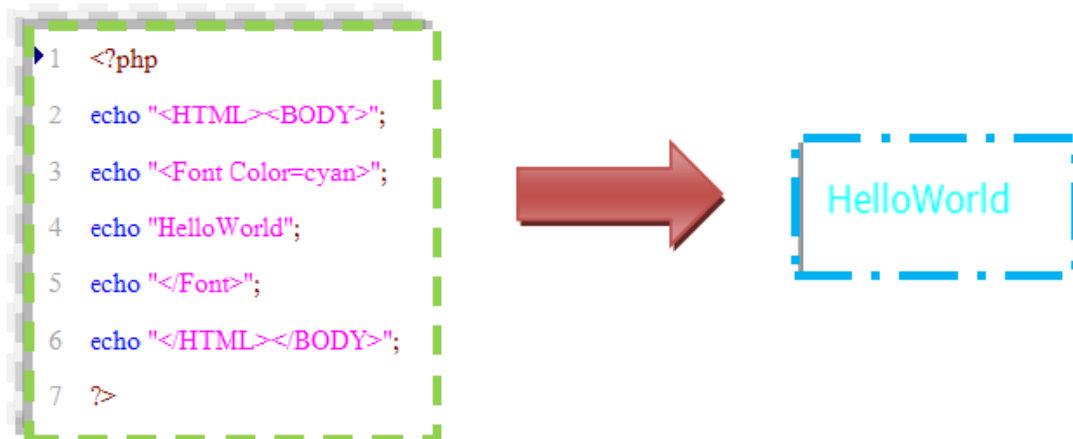
ปัจจุบัน

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 ตัวอย่างไปนี้จะแสดงถึงลักษณะของการเขียนโปรแกรมภาษา PHP โดยทำการเขียนคำสั่ง PHP ในทำการพิมพ์ข้อความว่า Hello World และทำการแทรก PHP ไว้ในไฟล์ HTML



ในทางกลับกันเราก็สามารถเขียนโปรแกรมภาษา HTML แทรกไว้ภายในภาษา PHP ได้เช่นกัน ดังตัวอย่างต่อไปนี้จะทำการพิมพ์ข้อความว่า Hello World เป็นตัวหนังสือสีฟ้าโดยใช้แท็กของ HTML เพื่อแสดงสีของตัวอักษร



1.4 การเตรียมการเพื่อเขียนโปรแกรมภาษา PHP

ในการเขียนโปรแกรมบนอินเทอร์เน็ตโดยใช้ภาษา PHP สิ่งที่คุณพัฒนาต้องมีประกอบไปด้วย

- เครื่องคอมพิวเตอร์อย่างน้อย 1 เครื่อง โดยปกติการพัฒนาระบบงานบนเว็บจะต้องมีเครื่องคอมพิวเตอร์ที่ทำเป็น Web Server และ Web Browser แต่ถ้ามีงบประมาณจำกัดในการจัดหาเครื่องคอมพิวเตอร์ก็สามารถใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียวก็ได้ โดยให้เครื่องคอมพิวเตอร์นี้ทำหน้าที่เป็นทั้ง Web Server และ Web Browser ในตัวเดียวกัน

- ระบบปฏิบัติการ Windows หรือ Unix หรือ Linux แต่ในที่นี้จะกำหนดระบบปฏิบัติการ Windows
- โปรแกรม Web Server ซึ่งสามารถเลือกใช้โปรแกรมอะไรก็ได้ เช่น Apache, Internet Information Server (IIS), Personal Web Server (PWS), OminHTTPd เป็นต้น
- PHP Engine หรือตัวแปลภาษา PHP ซึ่งเป็นโปรแกรมที่จะทำหน้าที่แปลภาษา PHP เป็นภาษาเครื่องเพื่อให้คอมพิวเตอร์สามารถทำงานและประมวลผลได้ตามที่ต้องการ
- 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 ในรายวิชานี้จะเลือกใช้โปรแกรม Adobe Dreamweaver

สรุป

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

ในการเขียนโปรแกรมบนเว็บสามารถที่จะแบ่งลักษณะการทำงานโปรแกรมได้เป็น 2 ประเภท คือ Static Programming, Dynamic Programming โดย Static Programming เป็นลักษณะของโปรแกรมเว็บที่ไม่ค่อยมีการเปลี่ยนแปลง เช่น เว็บประวัติส่วนตัว, เว็บนำเสนอประวัติและโครงสร้างขององค์กร เป็นต้น ส่วน Dynamic Programming เป็นลักษณะของโปรแกรมบนเว็บที่เกิดขึ้นมาเพื่อแก้ปัญหาความไม่ยืดหยุ่นในการจัดการข้อมูลของ Static Programming โดยเหมาะสำหรับเว็บที่ต้องการมีการเปลี่ยนแปลงข้อมูลอยู่บ่อยครั้งหรือเว็บที่มีการจัดเก็บข้อมูลจำนวนมากไว้ในฐานข้อมูล เช่น เว็บหนังสือพิมพ์, เว็บแสดงรายละเอียดของสินค้า เป็นต้น

โปรแกรมภาษา PHP ซึ่งย่อมาจากคำว่า Personal Home Page ข้อดีของภาษา PHP มีด้วยกันหลายข้อ เช่น เป็นโปรแกรมภาษาที่สามารถดาวน์โหลดได้ฟรีไม่มีปัญหาเรื่องลิขสิทธิ์, มี

การแปลภาษาและทำการประมวลผลได้อย่างรวดเร็ว, สามารถทำงานได้ทั้งบนระบบปฏิบัติการ Windows, Unix, Linux และ Macintosh เป็นต้น ก่อนที่จะเริ่มลงมือเขียนโปรแกรมภาษา PHP จะต้องเตรียมการต่าง ๆ คือ เครื่องคอมพิวเตอร์อย่างน้อย 1 เครื่อง, ระบบปฏิบัติการ, โปรแกรม Web Server, PHP Engine, โปรแกรมฐานข้อมูล, PhpMyAdmin, โปรแกรม Web Authoring และ Editor

2. PHP เบื้องต้น

รูปแบบของภาษา HTML

```
<html>
```

```
<head>
```

```
<title> ข้อความแถบเรื่อง </title>
```

```
</head>
```

```
<body>
```

```
ข้อความเอกสาร HTML
```

```
</body>
```

```
</html>
```

<html>...</html> เป็นคำสั่งที่แสดงให้เห็นว่าเอกสารที่เขียนนี้เป็นเอกสาร HTML

<head>...</head> เป็นคำสั่งที่แสดงให้รู้ว่าส่วนนี้คือส่วนของ title bar

<title>...</title> เป็นคำสั่งที่ใช้แสดงข้อความใน title bar

<body>...</body> เป็นคำสั่งที่ใช้แสดงให้รู้ว่าส่วนนี้คือส่วนของ เอกสารที่เราต้องการให้ปรากฏบนหน้าจอ

สำหรับในโปรแกรมภาษา PHP ก็มีแท็กที่ใช้สำหรับการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของคำสั่ง PHP

เช่นเดียวกัน ซึ่งภาษา PHP มีรูปแบบของบล็อกคำสั่ง 3 รูปแบบ คือ

รูปแบบคำสั่งของภาษา PHP

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>

เราสามารถใช้อีกคำสั่ง PHP รูปแบบใดก็ได้ ซึ่งก็จะให้ ผลลัพธ์ไม่แตกต่างกัน โดยจากตัวอย่างข้างต้น โปรแกรมภาษา PHP จะทำการพิมพ์ข้อความว่า Hello World ออกมาแสดงผลยังเบราว์เซอร์ ต่อไปนี้คือ ข้อเสนอแนะบางประการสำหรับบล็อกรหัสคำสั่ง PHP

1. เราสามารถเขียนบล็อกรหัสคำสั่งของ PHP แทรกไว้ในแท็ก HTML ได้ เช่น

<HTML>

<BODY>

<?

print "Hello World"

?>

```
</BODY>
```

```
</HTML>
```

2. เราสามารถเขียนบล็อกคำสั่งของ PHP นี้ได้หลายๆบล็อกคำสั่งภายในโปรแกรมเดียวกัน ดังตัวอย่างต่อไปนี้จะทำการพิมพ์ข้อความว่า Hello World Good Morning How are you? เรียงต่อกัน

```
<HTML>
```

```
<BODY>
```

```
<?
```

```
    print "Hello Eorld ";
```

```
?>
```

```
    Good Morning
```

```
<?
```

```
    print "How are you";
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

3. เราสามารถเขียนบล็อกคำสั่งของ PHP ได้อย่างอิสระโดยไม่ต้องอยู่ภายในแท็กของคำสั่ง HTML ก็ได้ เช่น

```
<?
```

```
    print "How are you";
```

```
?>
```

4. เมื่อทำการเขียนบล็อกคำสั่งของ PHP แล้ว จำไว้เสมอว่าภายในบล็อกคำสั่งต้องเป็นภาษา PHP เท่านั้น มิเช่นนั้นจะแสดงผล Error หรือแจ้งความผิดพลาด ดังตัวอย่าง

```
<HTML>

<BODY>

    <?

        Hello World

    ?>

</BODY>

</HTML>
```

5. เมื่อเราเปิดบล็อกคำสั่ง PHP แล้ว ภายในบล็อกคำสั่งนี้จำบรรจุคำสั่งภาษา PHP ก็คำสั่งก็ได้ แต่ว่าในแต่ละคำสั่งจะต้องลงท้ายด้วยเครื่องหมาย; (Semi-colon) และเราสามารถเขียนคำสั่งของ PHP หลายๆ คำสั่งภายใน 1 บรรทัดได้ เช่น

```
<?

    print "How are you";      print "Hello World";

?>
```

คำสั่งแสดงผลลัพธ์

ในการแสดงผลลัพธ์ของโปรแกรมภาษา PHP ได้จัดเตรียมคำสั่งไว้ 3 รูปแบบ คือ

1. **echo** รูปแบบของคำสั่งนี้เป็น ดังนี้

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

พิจารณาตัวอย่างของการใช้คำสั่งนี้เป็น echo เพื่อแสดงผลลัพธ์ ต่อไปนี้

```
<HTML>

<BODY>
```

```

<?
    echo "This is PHP World";

?>

<br>

<?
    echo "1+2 = ",1+2;

?>

</BODY>

</HTML>

```

จากผลลัพธ์ ที่เขียน โปรแกรมจะแสดงออกมาดังนี้

This is PHP World

1+2=3

จากตัวอย่างข้างต้นสามารถอธิบายการทำงานได้

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

2. print รูปแบบของคำสั่งนี้เป็น ดังนี้

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

รูปแบบคำสั่งการใช้งานของคำสั่ง print และ echo จะมีรูปแบบเหมือนกันพิจารณาตัวอย่างของการใช้คำสั่ง Print เพื่อแสดงผล

```
<?
```

```
print "This is PHP World";
```

```
?>
```

จากโปรแกรมนี้จะทำการพิจารณาข้อความว่า This is PHP World ปรากฏบนบราวเซอร์

ข้อแตกต่างของคำสั่ง `echo` และ `print` คือ `print` ไม่สามารถแยกข้อความ กับ นิพจน์ ค่าตัวแปรใดๆ ด้วยเครื่องหมาย , ได้ แต่คำสั่ง `echo` สามารถทำได้

3. printf รูปแบบคำสั่งนี้เป็น ดังนี้

รูปแบบคำสั่ง `printf` จะแตกต่างจากคำสั่ง `echo` และ `print` คือจะต้องมีเครื่องหมายวงเล็บเปิดและปิด เช่น

```
<?
```

```
printf ("This is PHP World");
```

```
?>
```

นอกจากคำสั่ง `Printf` ยังมีลักษณะพิเศษก็คือ คำสั่ง `printf` สามารถจัดรูปแบบการแสดงผลของข้อมูลที่เป็นตัวเลขได้ด้วย ดังตารางต่อไปนี้

สัญลักษณ์	คำอธิบาย
%d	แสดงค่าอาทิกวเมนตีในรูปแบบเลขฐานสิบสอง
%b	แสดง Integer ในรูปของเลขฐานสอง
%c	แสดง Integer ในรูปของรหัส ASCII
%f	แสดง Integer ในรูปของทศนิยม
%o	แสดง Integer ในรูปของฐานแปด
%s	แสดงค่าอาทิกวเมนตีในรูปแบบของเลขสตริง
%x	แสดง Integer ในรูปของเลขฐานสิบหก (ตัวเล็ก)
%X	แสดง Integer ในรูปของเลขฐานสิบหก (ตัวใหญ่)

พิจารณาตัวอย่างโปรแกรมต่อไปนี้

```
<?
```

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

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

```
?>
```

เมื่อทำการรัน แล้ว ผลลัพธ์จะออกมาเป็นแบบนี้

10/3 = 3

10/2 = 3.33

ผลลัพธ์ ในแต่ละบรรทัดสามารถอธิบายได้ดังนี้

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

ใช้รหัสรูปแบบ %d แสดงรูปแบบของเลข

จำนวนเต็ม ดังนั้น 10/3 จะแสดงผลคือ 3

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

ใช้รหัสรูปแบบ %.2f แสดงรูปแบบของเลข

ทศนิยม 2 ตำแหน่ง ดังนั้น 10/3 จะแสดงผล คือ 3.33

คำสัง่หมายเหตุ

ในการเขียนโปรแกรมบางครั้งเราอาจต้องการเขียนข้อความอะไรบางอย่างเพื่ออธิบายว่าคำสั่งในแต่ละบรรทัดนั้น คือ อะไรหรือมันทำงานอย่างไร เพื่อที่เราอาจจะย้อนกลับมาศึกษาโปรแกรมที่เขียนขึ้นอีกครั้ง และ ทำให้เข้าใจโปรแกรมเดิมได้อย่างรวดเร็ว โปรแกรมภาษา PHP มีเครื่องหมาย Comment ที่ใช้สำหรับการระบุข้อความใดลงไปในโปรแกรม โดยที่ส่วนของโปรแกรมถูก Comment ไว้จะไม่ถูกแปลภาษา ดังนั้นไม่ว่าเราจะเขียนอะไรลงในโปรแกรม โดยที่ส่วนของโปรแกรมที่ถูก Comment ไว้จะไม่ถูกแปลภาษา ดังนั้นไม่ว่าเราจะเขียนอะไรลงในส่วนที่ถูก Comment ไว้ก็จักจะไม่มีความผิดพลาดเกิดขึ้น เครื่องหมาย Comment ของโปรแกรมภาษา PHP มีอยู่ 3 รูปแบบ คือ

1. เครื่องหมาย #

2. เครื่องหมาย //

ทั้ง 2 คำสัง่นี้จะระบุ Comment ได้ทีละ 1 บรรทัด ดังตัวอย่าง ต่อไปนี้

<?

echo "This is PHP World";

echo "Good Morning";

// echo "How are you?";

?>

จากโปรแกรมนี้เมื่อรันโปรแกรมแล้วบราวเซอร์จะทำการแสดงข้อความว่า This is PHP World เพียงบรรทัดเดียว

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 /*....*/ ซึ่งจะทำการ Comment ได้ทีละหลายๆ บรรทัด

แท็กคำสั่งที่สำคัญของ HTML

แท็กภาษา HTML	ความหมาย
<TITEL>...</TITEL>	ระบุข้อความบริเวณ Title bar

<H ₁ >...</ H ₁ >	กำหนดขนาดอักษรขนาด 1
< H ₂ >...</ H ₂ >	กำหนดขนาดอักษรขนาด 2
< H ₃ >...</ H ₃ >	กำหนดขนาดอักษรขนาด 3
...</ FONT >	กำหนดสีตัวอักษร
<P>	ขึ้นบรรทัดใหม่และเว้นบรรทัด
 	ขึ้นบรรทัดใหม่
...	กำหนดตัวหนังสือเป็นตัวหนา



ความรู้พื้นฐานเกี่ยวกับภาษา PHP

ในการประมวลผลใดๆ ในคอมพิวเตอร์ต้องมีการจัดเก็บข้อมูลไว้ในหน่วยความจำของคอมพิวเตอร์ มิฉะนั้นแล้วการประมวลผลจะไม่สามารถเกิดขึ้นได้เลย เหตุผลง่ายๆ คือถ้ามีการประมวลผลเกิดขึ้นแล้วต้องการที่จะนำผลลัพธ์ (Output) จากการประมวลผลนี้จัดเก็บเอาไว้เพื่อที่จะนำไปเป็นข้อมูลนำเข้า (Input) ของการประมวลผลอื่นๆ ถัดไป การจัดเก็บข้อมูลดังกล่าวนี้ภายในคอมพิวเตอร์ต้องมีการจัดสรรพื้นที่ของหน่วยความจำเอาไว้เก็บข้อมูล ซึ่งการจัดสรรพื้นที่ในหน่วยความจำนี้เราสามารถกระทำได้โดยการใช้ตัวแปร (Variables) หรือ ค่าคงที่ (Constant) แต่ทั้งตัวแปรและค่าคงที่มีลักษณะและการใช้งานที่แตกต่างกัน ในบทนี้เนื้อหาจะเน้นให้นักเรียนรู้จักการใช้งานของตัวแปรและค่าคงที่

การเปิดปิดแท็ก PHP (PHP Code Syntax)

รูปแบบแท็ก	เปิดแท็ก PHP	ปิดแท็ก PHP
แบบมาตรฐาน	<?php	?>
แบบสั้น	<?	?>
แบบ ASP	<%	%>
แบบ Script	<script language="PHP">	</script>

รูปแบบคำสั่ง (PHP Statement)

```
<BODY>
```

```
<?php
```

```
echo "Hello, World!!";
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

ตัวแปร (Variables)

ตัวแปร คือ ชื่อที่ตั้งขึ้นมาเพื่อใช้จัดเก็บข้อมูล การเขียนคำสั่งให้โปรแกรม (PHP) จองพื้นที่ในหน่วยความจำ (Memory) ไว้ส่วนหนึ่ง เพื่อเก็บข้อมูลใดๆ ก็ได้ ไม่ว่าจะเป็นข้อมูลตัวเลข ตัวอักษร เครื่องหมาย เป็นต้น ตามที่ผู้เขียนโปรแกรมกำหนด

ลักษณะที่สำคัญของตัวแปร คือ

1. ค่าที่จัดเก็บในตัวแปรใดๆ เมื่อปิดโปรแกรมหรือเครื่องคอมพิวเตอร์ข้อมูลนั้นจะถูกลบทิ้งจาก RAM
2. ตัวแปรสามารถที่จะเปลี่ยนแปลงค่าที่จัดเก็บไว้ได้ตลอดเวลา ลองพิจารณาโปรแกรม PHP ต่อไปนี้

<?

echo 40+5;

echo "
";

echo 30-8;

?>

จากโปรแกรมนี้คำสั่ง echo 40 + 5; จะมีค่าเท่ากับ 45 และ คำสั่ง echo 30 + 8; จะมีค่าเท่ากับ 22 ถ้าต้องการนำข้อมูลหรือค่าที่ได้จากสองคำสั่งนี้มาทำการลบกัน หรือดำเนินการกัน เราจะทำอย่างไร

ปัญหานี้ก็คือ เราต้องทำการจัดเก็บค่า 45 และ 22 ไว้ที่ใดที่หนึ่งเสียก่อน หลังจากนั้นก็ค่อยนำค่าทั้งสองออกมาลบกัน ซึ่งที่ใดที่หนึ่งในนี้ก็คือหน่วยความจำนั่นเอง โดยปกติแล้วหน่วยความจำคอมพิวเตอร์จะมีการแบ่งพื้นที่ในการจัดเก็บและมีการอ้างอิงถึงพื้นที่ที่เก็บข้อมูลต่างๆ ด้วยหมายเลขที่อยู่ของหน่วยความจำ เช่น หมายเลข 3762 เป็นหมายเลขที่อยู่จัดเก็บข้อมูลเลข 45 หมายเลข 1769 เป็นหมายเลขที่อยู่จัดเก็บข้อมูลเลข 22 ดังนั้นเราก็นำหมายเลขที่อยู่ 3762 และ 1769 มาลบกันได้

			45																	
										22										

แต่ว่าในทางปฏิบัติจริงๆ แล้วจะไม่สามารถทราบได้ว่าข้อมูลต่างๆ ถูกจัดเก็บไว้ที่หมายเลขที่อยู่อะไร ถึงแม้เราจะทราบว่าหมายเลขที่อยู่อะไร ก็ถือว่าเป็นความลำบากในการจดจำหมายเลขที่อยู่ดังกล่าว

ดังนั้น **ตัวแปร** จึงถูกนำมาใช้เพื่อจัดเก็บข้อมูลต่างๆ ในหน่วยความจำ โดยที่ตัวแปรนี้จะเป็นการตั้งชื่อขึ้นมาเพื่ออ้างอิงถึงข้อมูลต่างๆ ในหน่วยความจำนั่นเอง

การประกาศตัวแปร

- การประกาศตัวแปรเริ่มต้นด้วยเครื่องหมาย \$ (Dollar sign) เสมอ
- หลังเครื่องหมาย \$ ชื่อตัวแปรจะต้องเริ่มต้นด้วยตัวอักษรภาษาอังกฤษหรือเครื่องหมาย

underscore ("_")

- ตัวถัดมาของชื่อตัวแปรสามารถเป็นตัวอักษร หรือตัวเลข หรือเครื่องหมาย _ (Under-score) เพื่อ

ผสมชื่อได้เสมอ

- ชื่อของตัวแปรมีความสมบัติเป็น case-sensitive คือตัวอักษรตัวใหญ่และตัวเล็กถือว่าเป็นคนละตัว

ตัวอย่าง test1.php

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

- ตัวแปรที่เป็นอักษรต้องอยู่ในเครื่องหมาย "....."
- ตัวแปร \$Age และ \$age ถือว่าเป็นคนละตัวกัน

Test2.php

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

ชนิดของข้อมูล

ในการเขียนโปรแกรมด้วยภาษา PHP เช่นการประกาศตัวแปร หรือกำหนดค่าข้อมูลต่างๆ ต้องระบุชนิดข้อมูลอย่างชัดเจน โดยชนิดข้อมูลในภาษา PHP มี 2 ประเภท คือ

1. ชนิดข้อมูลพื้นฐาน (Primitive Data Type) คือชนิดข้อมูลที่ใช้ในการเขียนโปรแกรมสำหรับเก็บข้อมูลชนิดต่างๆ ได้แก่ จำนวนเต็ม(Integer) จำนวนทศนิยม(Floating Point) ข้อมูลอักขระ(Character) และข้อมูลตรรกะ (Logical Data)

2. ชนิดข้อมูลแบบอ้างอิง (Reference Data Type) คือชนิดข้อมูลที่มีการอ้างอิงตำแหน่งในหน่วยความจำ ได้แก่ข้อมูลของคลาส เช่น String และข้อมูลแบบอาเรย์

ชนิดข้อมูลพื้นฐานในภาษา PHP มีทั้งหมด 8 ชนิดข้อมูลโดยแบ่งออกเป็น 4 ประเภท ประกอบด้วย

1. ชนิดข้อมูลจำนวนเต็ม(Integer) ได้แก่ long, int, short และ byte
2. ชนิดข้อมูลจำนวนทศนิยม(Floating Point) ได้แก่ double และ float
3. ชนิดข้อมูลอักขระ(Character) ได้แก่ char
4. ชนิดข้อมูลตรรกะ (Logical Data) ได้แก่ Boolean

ชนิดข้อมูลแต่ละประเภทจะมีขนาดและช่วงค่าของข้อมูลแตกต่างกัน และสามารถสรุปประเภทชนิดข้อมูล ขนาดและช่วงค่าของข้อมูล และค่าเริ่มต้น

ชนิดข้อมูลจำนวนเต็ม(Integer)

ชนิดข้อมูลจำนวนเต็มประกอบด้วย 4 ชนิด ได้แก่ long, int, short และ byte เป็นชนิดข้อมูลที่ใช้เก็บข้อมูลตัวเลขจำนวนเต็มในทางคณิตศาสตร์ ช่วงของข้อมูลขึ้นอยู่กับขนาดของตัวแปรที่ประกาศ ตัวอย่างเช่น ตัวแปร int จะมีขนาด 32 บิต เก็บข้อมูลได้ -2,147,483,648 ถึง 2,147,483,647 การกำหนดค่าข้อมูลให้กับตัวแปรชนิดข้อมูลจำนวนเต็ม มีดังนี้

1. เลขฐานสิบคือการเขียนค่าของข้อมูลจำนวนเต็มทั่วไป เช่น 1200 150 -250 เป็นต้น โดยการเขียนจะไม่มีเครื่องหมาย , (Comma) เช่น 25,000 จะต้องเขียนเป็น 25000
2. เลขฐานแปดคือการเขียนค่าของข้อมูลจำนวนเต็มโดยใช้เลขฐานแปด โดยการขึ้นต้นข้อมูลด้วยเลข 0 แล้วตามด้วยเลข 0 ถึง 7 ตัวอย่างเช่น 036 เป็นเลขฐานแปด มีค่าเท่ากับ 30 ในเลขฐานสิบ
3. เลขฐานสิบหกคือการเขียนค่าของข้อมูลจำนวนเต็มโดยใช้เลขฐานสิบหก โดยการขึ้นต้นข้อมูลด้วยเลข 0x หรือ 0X แล้วตามด้วยเลข 0 ถึง 9 และ ตัวอักษร A ถึง F ตัวอย่างเช่น 0XB2 เป็นเลขฐานแปด มีค่าเท่ากับ 178 ในเลขฐานสิบ

การประกาศตัวแปรใดๆ ให้เป็นข้อมูลจำนวนเต็มต้องคำนึงถึงขอบเขตของช่วงข้อมูลที่เก็บได้ และต้องกำหนดค่าชนิดข้อมูลในเหมาะสมตรงกับชนิดข้อมูลที่ประกาศตัวแปร

Integer.php

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

ชนิดข้อมูลจำนวนทศนิยม(Floating Point)

ชนิดข้อมูลจำนวนทศนิยม ประกอบด้วยข้อมูลที่เป็นตัวเลขทศนิยม เช่น 1200.578 การเขียนโปรแกรมด้วยภาษา PHP จะประกาศตัวแปรด้วยคำว่า float และ double โดยชนิดข้อมูล float จะเก็บข้อมูล 32 บิตตามมาตรฐาน single precision คือมีส่วนละเอียดยของตัวเลขจำนวน 24 บิต และส่วนเลขยกกำลัง 8 บิต และชนิดข้อมูล double จะเก็บข้อมูล 64 บิตตามมาตรฐาน double precision คือมีส่วนละเอียดยของตัวเลขจำนวน 53 บิต และส่วนเลขยกกำลัง 11 บิต รูปแบบการตัวเลขทศนิยมมี 2 แบบ ดังนี้

1. ตัวเลขทศนิยม เป็นการเขียนตัวเลขทศนิยมที่มีเครื่องหมายจุดทศนิยม ตัวอย่างเช่น 3.14 เป็นต้น
2. ตัวเลขยกกำลังสิบ เป็นการเขียนตัวเลขทศนิยมอยู่ในรูปเลขยกกำลังสิบ (Exponential Form) โดยใช้ตัวอักษร E หรือ e ระบุจำนวนที่เป็นเลขยกกำลัง เช่น 6.12E12 หรือ 125.03E-5 เป็นต้น

Float.php

```
<?
    $num1 = 3.58;
    $num2 = 3.5e3;
    $num3 = 3e-4;
    echo “ค่าของ 3.58 มีค่าเท่ากับ $num1 <br>”;
    echo “ค่าของ 3.5e3 มีค่าเท่ากับ $num2 <br>”;
```


echo “ค่าของ 3e-4 มีค่าเท่ากับ \$num3
”;

?>

ชนิดข้อมูลอักขระ (Character)

ชนิดข้อมูลอักขระจะต้องประกาศตัวแปรด้วยคำว่า char คือการเก็บข้อมูลเป็นตัวอักษร หรืออักขระบนแป้นพิมพ์ได้เพียง 1 ตัวอักษร ซึ่งในภาษา PHP จะเก็บข้อมูลอักขระอยู่ในรูปแบบของรหัสแอสกี (ASCII Code) หรือรูปแบบมาตรฐาน Unicode ขนาด 16 บิต การกำหนดค่าข้อมูลอักขระจะอยู่ในเครื่องหมาย ‘ ’ โดยจะขึ้นต้นด้วยสัญลักษณ์ \u และตามด้วยเลขฐานสิบหก (Hexadecimal Number) โดยมีค่าตั้งแต่ ‘\u0000’ ถึง ‘\uFFFF’ หรือกำหนดค่าข้อมูลอักขระด้วยตัวอักษรบนแป้นพิมพ์ ตัวอย่างเช่น ‘A’, ‘x’, ‘\$’ หรือ ‘1’ เป็นต้น

ตัวอย่างการประกาศชนิดข้อมูลอักขระ เช่น

```
char grade= 'A';
```

```
char grade= '\u0041';
```

เป็นการประกาศตัวแปรที่มีชื่อว่า grade เป็นชนิดข้อมูลอักขระ มีการกำหนดค่าเริ่มต้นเป็นตัวอักษร A ซึ่งมีค่าเป็นเลขฐานสิบหก คือ 0041

ตัวอักขระพิเศษ หรือ Escape Sequence เป็นการแสดงข้อมูลอักขระบนแป้นพิมพ์ที่ใช้แทนการขึ้นบรรทัดใหม่ หรือ ตัวอักขระพิเศษในการเขียนโปรแกรม

ตัวอักขระพิเศษ	รหัส Unicode	ความหมาย
\b	\u000B	Backspace ลบอักขระตำแหน่งเคอร์เซอร์
\t	\u0009	Tab การเว้นช่องว่างตามระยะ
\n	\u000A	New Line การขึ้นบรรทัดใหม่
\r	\u000D	Return การขึ้นบรรทัดใหม่
\\	\u005C	Backslash แสดงตัวอักขระ \
\'	\u0027	Single Quote แสดงตัวอักขระ ‘
\“	\u0022	Double Quote แสดงตัวอักขระ “

Float.php

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

- \$num3 จะเก็บสตริง 450 ไม่ใช่ตัวเลข 450
- ในโปรแกรมการใช้รหัส (\) Escape นำหน้าชื่อเพื่อให้เห็นเครื่องหมาย \$ ออกมา ซึ่งทำให้ num1 num2 num3 เป็นข้อความธรรมดา ไม่ได้เป็นตัวแปร

ข้อมูลชนิดตรรกะ (Boolean)

ลักษณะข้อมูลจะเก็บค่าเพื่อใช้สำหรับตรวจสอบเงื่อนไข ซึ่งจะมีค่าที่เก็บเพียง 2 ค่าเท่านั้น คือ ค่าจริง (True) และเท็จ (False)

boolean.php

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

ตัวแปรอาร์เรย์

Non_array.php

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

Array1.php

```
<?
    $score[1] = 58;
    $score[2] = 65;
    $score[3] = 74;
    $score[4] = 62;
    $score[5] = 85;
    echo “คะแนนสอบนักเรียนคนที่ 1 เท่ากับ $score[1] <br>”;
    echo “คะแนนสอบนักเรียนคนที่ 2 เท่ากับ $score[2] <br>”;
    echo “คะแนนสอบนักเรียนคนที่ 3 เท่ากับ $score[3] <br>”;
    echo “คะแนนสอบนักเรียนคนที่ 4 เท่ากับ $score[4] <br>”;
    echo “คะแนนสอบนักเรียนคนที่ 5 เท่ากับ $score[5] <br>”;
?>
```

Array หมายถึง ตัวแปรชุดที่ใช้เก็บตัวแปรชนิดเดียวกันไว้ด้วยกัน เช่น เก็บ ข้อมูล char ไว้กับ char เก็บ int ไว้กับ int ไม่สามารถเก็บข้อมูลต่างชนิดกันได้ เช่น char กับ int เรียก array อีกอย่างว่า หน่วยความจำแบ่งเป็นช่อง การกำหนดสมาชิกของ array จะเขียนภายในเครื่องหมาย []

Array2.php

<?

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

?>

ฟังก์ชันที่เกี่ยวข้องกับตัวแปร

1. ฟังก์ชัน `gettype()` จะ return ชนิดของตัวแปรกลับมา

Boolean	T-F
integer	ตัวเลขจำนวนเต็ม
double	ตัวเลขจำนวนทศนิยม
string	ตัวอักษร
array	อาร์เรย์
object	แบบอบเจ็กต์
NULL	ว่าง
unknown type	ไม่ระบุชนิดตัวแปร

นิพจน์ (Expression) คือ การนำค่าคงที่หรือตัวแปรมาเชื่อมต่อกัน ด้วยเครื่องหมายทางคณิตศาสตร์ เช่น ตัวดำเนินการคณิตศาสตร์ เช่น บวก ลบ คูณ หาร เปอร์เซ็นต์ ดังตัวอย่างในตารางข้างล่างนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
+	บวก (Addition)	$X + y$
-	ลบ (Subtraction)	$X - y$
*	คูณ (Multiplication)	$X * y$
/	หาร (Division)	X / y
%	หารเอาผลลัพธ์เฉพาะเศษ (Modulus)	$X \% y$
++	เพิ่มค่าให้กับตัวแปร \$a\$ ขึ้นหนึ่งค่า	\$a++
--	ลดค่าให้กับตัวแปร \$a\$ ลงหนึ่งค่า	\$b--

ตัวดำเนินการเปรียบเทียบ คือ เครื่องหมายที่ใช้ในการเปรียบเทียบในทางคณิตศาสตร์

ผลลัพธ์จะมี 2 กรณีคือ ถ้าผลลัพธ์ถูกต้องหรือเป็นจริง (True) จะมีค่าเป็น 1 ถ้าผลลัพธ์ผิดหรือเป็นเท็จ (False) จะมีค่าเป็น 0 ผลลัพธ์ของการเปรียบเทียบ เรียกว่า ค่าคงที่บูลีน (Boolean Constant)

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
>	มากกว่า (Greater Than)	$X > y$
<	น้อยกว่า (Less Than)	$X < y$
>=	มากกว่าหรือเท่ากับ (Greater Than or Equal)	$X >= y$
<=	น้อยกว่าหรือเท่ากับ (Less Than or Equal)	$X <= y$
==	เท่ากับ (Equal)	$X == y$
!=	ไม่เท่ากับ (Not Equal)	$X != y$

ตัวดำเนินการตรรกะ คือเครื่องหมายที่ใช้เชื่อมเงื่อนไข 2 เงื่อนไข หรือมากกว่า 2 เงื่อนไข เพื่อให้การเปรียบเทียบมีความละเอียดมากขึ้น ใช้สัญลักษณ์แทนในแต่ละเครื่องหมาย ดังนี้

ตัวดำเนินการ	อธิบาย	ความหมาย
&&	และ (and)	จะให้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขทั้งสองเป็นจริง แต่ถ้าเงื่อนไขใดเป็นเท็จ หรือทั้งสองเงื่อนไขเป็นเท็จจะทำให้ผลลัพธ์เป็นเท็จ
	หรือ (or)	จะให้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง หรือเป็นจริงทั้งสองเงื่อนไข แต่ถ้าเป็นเท็จทั้งสองเงื่อนไขจะทำให้ผลลัพธ์เป็นเท็จ
!	ไม่ใช่ (not)	จะให้ผลลัพธ์เป็นจริงเมื่อเงื่อนไขหลัง not เป็นเท็จ แต่ถ้าเงื่อนไขหลัง not เป็นจริงจะทำให้ผลลัพธ์เป็นเท็จ ตัวดำเนินการ ความหมาย



คำสั่ง และเงื่อนไข

เงื่อนไขในการเขียนโปรแกรม เป็นการกำหนดเส้นทางการทำงานของโปรแกรมตามที่เราต้องการ เราสามารถกำหนดการทำงานได้จาก ผลลัพธ์ของการเปรียบเทียบ หรือ การคืนค่าของฟังก์ชัน

คำสั่งเงื่อนไขในภาษา PHP มี 2 คำสั่งคือ

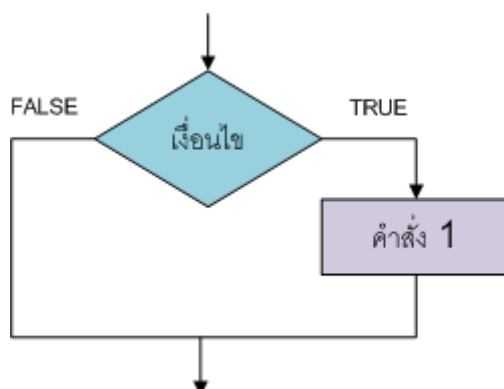
if... elseif... else... switch... case...

คำสั่งตรวจสอบเงื่อนไข If

เงื่อนไข if... elseif... else... จะถูกแบ่งเป็น 3 แบบใหญ่ๆ คือ

1. มีเงื่อนไขเดียว
2. มีสองเงื่อนไข
3. มีหลายเงื่อนไข

1. แบบเงื่อนไขเดียว (Single-alternative) จะทำงานก็ต่อเมื่อเงื่อนไขเป็นจริงเท่านั้น รูปแบบการเขียนคือ



```

if (เงื่อนไข)
{
    คำสั่ง 1;
}
  
```

If1.php

```
<?>
```

```

$price1 = 7000; // ราคาสินค้า 1
$price2 = 3000; // ราคาสินค้า 2
$discount = 0; // ตัวแปรส่วนลดที่คำนวณได้
if ($price1 >= 5000)
{
    $discount = $price1*0.03;
}
$netprice = $price1 - $discount;
echo "สินค้าที่ 1 ราคา $price1 บาท <br>";
  
```

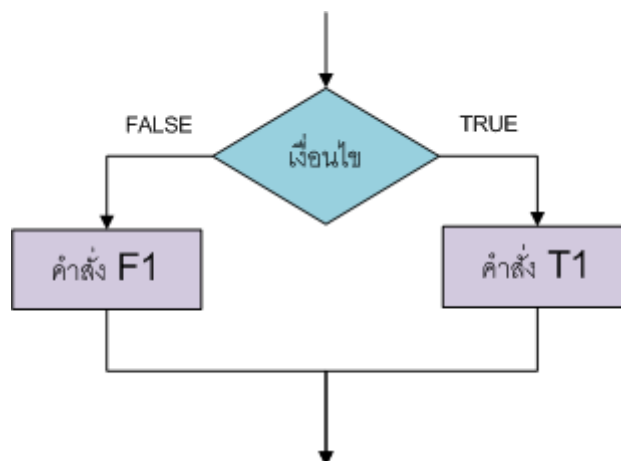
```

echo "ได้รับส่วนลด $discount บาท <br>";
echo "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <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 "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <br>";
?>

```

จะเห็นได้ว่า เราสามารถทำการคำนวณในเงื่อนไขได้ โปรแกรมจะทำการหาผลลัพธ์จากการคำนวณก่อน จากนั้นจึงทำการเปรียบเทียบ ถ้าหากผลการเปรียบเทียบเป็นจริง จึงจะทำงานในส่วนของเงื่อนไข แต่ถ้าหากเงื่อนไขเป็นเท็จ จะข้ามส่วนเงื่อนไขนี้ไป

2. แบบสองเงื่อนไข (Dual-alternative) เมื่อเงื่อนไขเป็นจริง จะทำงานในส่วน if แต่ถ้าเงื่อนไขเป็นเท็จจะทำงานในส่วน else



```

if (เงื่อนไข)
{
    คำสั่ง T1;
}
else
{
    คำสั่ง F1;
}

```

If2.php

<?

```
$price1 = 12000; // ราคาสินค้า 1
```

```
$price2 = 7000; // ราคาสินค้า 2
```



```

$discount = 0; // ตัวแปรส่วนลดที่คำนวณได้
if ($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 "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <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 "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <br>";
?>

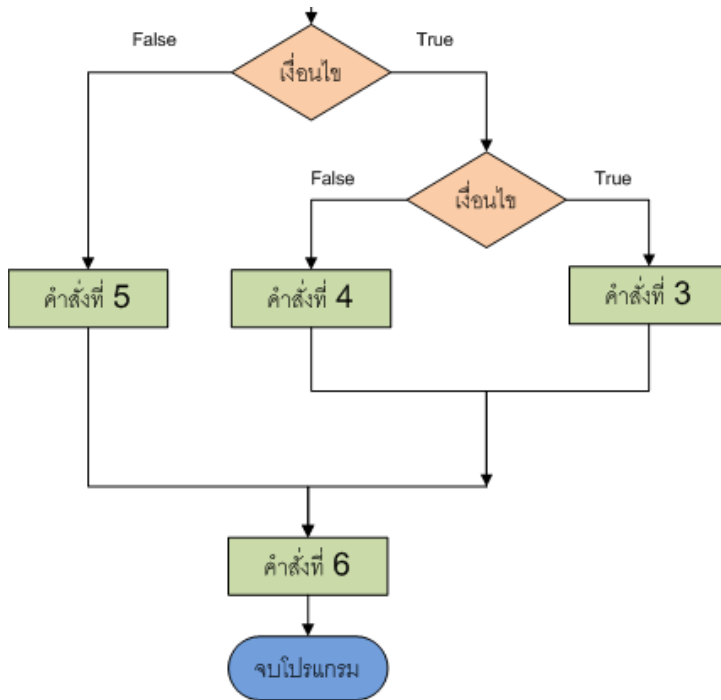
```

3. แบบหลายเงื่อนไข (Nested-alternative) การเอา If ไปไว้ใน If ทางฝั่ง True แต่ละครั้ง โดยโครงสร้าง If ซ้อน If นั้นเอาไว้ใช้กับ กรณีตรวจสอบเงื่อนไขที่มากกว่า 2 กรณี เพราะเนื่องจาก If else นั้น ตรวจสอบเงื่อนไขได้เพียง true หรือ false เท่านั้น

```

คำสั่ง 1;
คำสั่ง 2;
if (เงื่อนไข)
{
    if (เงื่อนไข)
    {
        คำสั่ง 3;
    }
}

```



If3.php

```

$price1 = 7000;
$price2 = 12000;
$price3 = 25000;
$price4 = 4500;
if (($price1 >= 5000) && ($price1 <= 10000))
{
    $discount = $price1*0.05;
}
Elseif (($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 "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <br>";
    echo "----- <br>";
    if (($price2 >= 5000) && ($price2 <= 10000))
    {
        $discount = $price2*0.05;
    }
    Elseif (($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 "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <br>";
    echo "----- <br>";
    if (($price3 >= 5000) && ($price3 <= 10000))
    {
        $discount = $price3*0.05;
    }
}
```

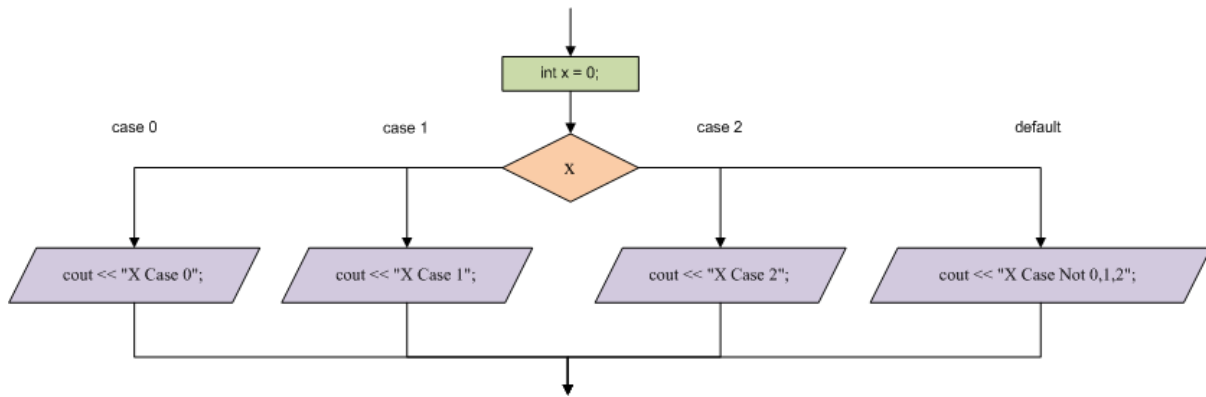
```
Elseif (($price3 >= 10001) && ($price3 <= 20000))
{
    $discount = $price3*0.05;
}
Elseif ($price3 >= 20001)
{
    $discount = $price3*0.1;
}
else
{
    $discount = 0;
}
$netprice = $price3 - $discount;
echo "สินค้าที่ 3 ราคา $price3 บาท <br>";
echo "ได้รับส่วนลด $discount บาท <br>";
echo "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <br>";
echo "----- <br>";
if (($price4 >= 5000) && ($price4 <= 10000))
{
    $discount = $price4*0.05;
}
Elseif (($price4 >= 10001) && ($price4 <= 20000))
{
    $discount = $price4*0.05;
}
Elseif ($price4 >= 20001)
{
    $discount = $price4*0.1;
}
else
{
```

```
        $discount = 0;
    }
    $netprice = $price4 - $discount;
    echo "สินค้าที่ 4 ราคา $price4 บาท <br>";
    echo "ได้รับส่วนลด $discount บาท <br>";
    echo "ราคาสินค้าสุทธิเท่ากับ $netprice บาท <br>";
    echo "----- <br>";

?>
```

คำสั่งตรวจสอบเงื่อนไข Switch

คำสั่งนี้เป็นคำสั่งที่มีโครงสร้างเป็นแบบเลือกทำ (Selection) ซึ่งคำสั่ง switch ใช้กับกรณีที่มีการเลือกการทำงานหลายเส้นทาง (Multi-Way Selection)



รูปแบบโครงสร้างคำสั่ง switch

```

switch ( นิพจน์ที่ใช้ทดสอบ )
{
    case ค่าคงที่ 1 : คำสั่ง 1;
                    break;
    case ค่าคงที่ 2 : คำสั่ง 2;
                    break;
    ...
    case ค่าคงที่ N : คำสั่ง N;
                    break;
    default       : คำสั่ง;
}
  
```

คำอธิบายรูปแบบโครงสร้างคำสั่ง switch

นิพจน์ที่ใช้ทดสอบ คือ นิพจน์ (Expression) ที่จะใช้ในการตรวจสอบว่าค่านั้นตรงกับ ค่าคงที่ ไหน โดยนิพจน์นั้นเป็นได้ทั้ง ตัวแปร หรือ Function และต้องมีค่าเป็นตัวเลขจำนวนเต็มและตัวอักษรเท่านั้น เช่น `int x` , `x` จะนำไปใช้เป็น นิพจน์ที่ใช้ทดสอบ ได้เพราะเป็นตัวแปรตัวเลขจำนวนเต็ม เป็นต้น

ค่าคงที่ คือ ค่าที่จะถูกนำมาเทียบกับ นิพจน์ที่ใช้ทดสอบ โดยถ้าตรวจสอบแล้วตรงกัน ก็จะไปทำคำสั่งที่อยู่หลัง : และถ้าตรวจสอบทุกๆ ค่าคงที่ แล้วไม่มี case ไหนตรงเลยก็จะเข้าไปยัง default และทำ คำสั่งหลัง : ของ default

คำสั่ง 1; คือ คำสั่งที่จะถูกทำเมื่อ นิพจน์ที่ใช้ทดสอบ เทียบกับ ค่าคงที่ แล้วตรงกันซึ่งคำสั่งนั้นมีคำสั่งเดียวหรือหลายๆ คำสั่งก็ได้ และเมื่อจบทุกคำสั่งแล้วจะต้องเขียน break; เพื่อเป็นตัวบอกกว่าหมด case นี้แล้ว

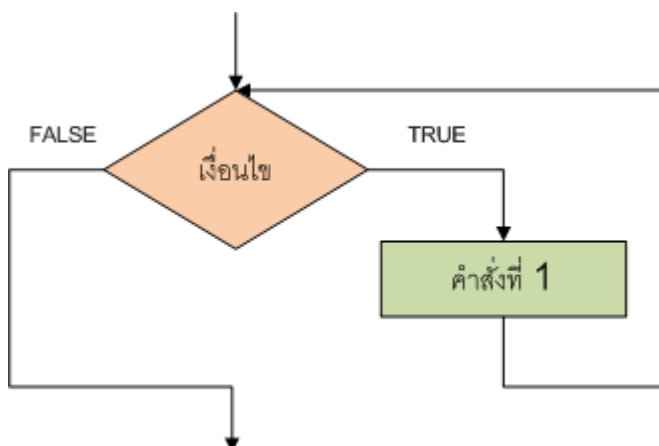
Switch.php

```
<?php
    $grade = 'C';
    echo 'เกรดที่คุณต้องการคือ เกรด ' . $grade;
    switch( $grade ){
        case 'A' : echo "คุณจะต้องได้คะแนนตั้งแต่ 80 คะแนนขึ้นไป"; break;
        case 'B' : echo "คุณจะต้องได้คะแนน 70 - 79 คะแนน"; break;
        case 'C' : echo "คุณจะต้องได้คะแนน 60 - 69 คะแนน"; break;
        case 'D' : echo "คุณจะต้องได้คะแนน 50 - 59 คะแนน"; break;
        default : echo "คุณแค่ไม่ต้องมาสอบ ไม่ต้องส่งงาน ก็จะติด 0 ไปเอง ^^"; break;
    }
?>
```

Switch Case นั้นกับการตรวจสอบเงื่อนไขที่ เขียนโค้ดใน case นั้นไม่ก็บรรทัดเท่านั้นครับ ถ้าเขียนยาวกว่านั้น จะค่อนข้างสับสน ว่าจบเคสที่ยัง (break ;) ถ้าหากมีการตรวจสอบแล้วต้องเขียนโค้ดยาวๆก็ใช้ if else ดีกว่าครับ เข้าใจง่ายกว่าว่าจบเคสที่ไหน เพราะมีวงเล็บปีกกา () เป็นตัวเปิดปิดคำสั่งให้

คำสั่งทำซ้ำ While

เป็นคำสั่งที่ใช้วนซ้ำ โดยจะตรวจสอบเงื่อนไขก่อน ถ้าพบว่าเงื่อนไขเป็นจริง จึงจะทำคำสั่งที่อยู่ภายใน ลูป (loop) หลังจากนั้นจึงกลับมาตรวจสอบเงื่อนไขอีกครั้ง วนซ้ำเช่นนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็นเท็จ



while (เงื่อนไข)

```
{
    คำสั่ง;
}
```

While.php

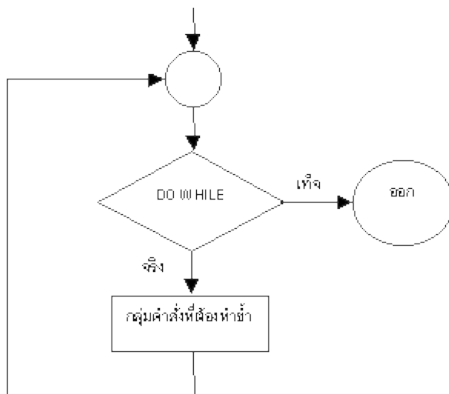
```
<?
    $count = 1;
    while ($count <= 10)
    {
        echo "Hello <br>";
        $count = $count+1;
    }
?>
```

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 เท่ากับ $sum";
?>
```


คำสั่ง do..while

คำสั่ง do..while จะมีการทำงานคล้ายกับ while แต่ต่างกันตรงที่ คำสั่ง do..while จะตรวจสอบเงื่อนไขทีหลัง นั่นคือจะมีการทำคำสั่งในลูปอย่างน้อย 1 ครั้งเสมอ เมื่อทำคำสั่งในลูปแล้วจึงตรวจสอบเงื่อนไขทีหลัง



```

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

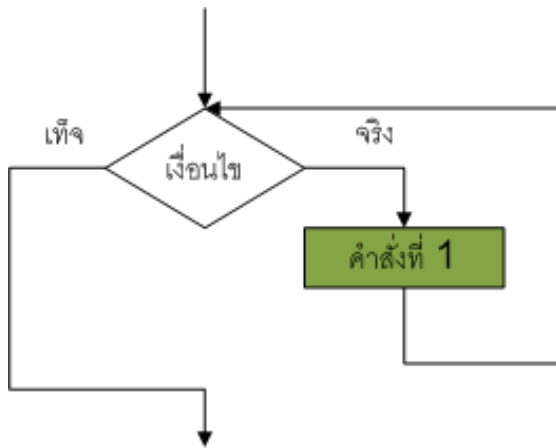
dowhile.php

```

<?
    $a = 1;
    do
    {
        echo $a, " ";
        $a++;
    } while($a<=20)
?>
  
```

คำสั่ง for

สำหรับคำสั่ง for นั้น จะมีการทำงานเหมือนกับ while คือตรวจสอบเงื่อนไขหรือการเปรียบเทียบก่อน แล้วจึงทำคำสั่งภายในลูป แต่มักใช้ในกรณีที่เราทราบจำนวนครั้งในการวนซ้ำที่แน่นอน



```
for (กำหนดค่าเริ่มต้น ; เงื่อนไข ; การเพิ่ม/ลดค่าตัวแปร)
{
    คำสั่ง;
}
```

For1.php

```
<?
    for($a = 1 ; $a < 13 ; $a++)
    {
        echo "<font color=green>";
        echo "2 x $a =",2*$a,"<br>";
        echo "</font>";
    }
?>
```

จากตัวอย่าง เป็นโปรแกรมสูตรคูณแม่ 2 โดย \$a จะเริ่มเก็บค่าจาก 1 ในขณะที่เงื่อนไขยังเป็นจริง (นั่นคือ \$a < 13) ก็จะทำคำสั่งในลูปทั้ง 3 คำสั่ง หลังจากนั้นตัวแปร \$a จะถูกเพิ่มค่าทีละหนึ่ง และทำงานไปเรื่อยๆ จนกระทั่งเงื่อนไข \$a < 13 เป็นเท็จ

For2.php

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

4

ฟังก์ชัน PHP และการจัดการไฟล์

ฟังก์ชันใน PHP

ฟังก์ชันในโปรแกรมส่วนใหญ่ได้รับการเรียกคำสั่งเพื่อทำงานอย่างเดียว สิ่งนี้ทำให้คำสั่งอ่านได้ง่ายและยอมให้ใช้คำสั่งใหม่แต่ละครั้งเมื่อต้องการทำงานเดียวกัน

ฟังก์ชันเป็นโมดูลเก็บคำสั่งที่กำหนดการเรียกอินเทอร์เฟซ ทำงานเดียวกัน และตัวเลือกส่งออกค่าจากการเรียกฟังก์ชัน คำสั่งต่อไปเป็นการเรียกฟังก์ชันอย่างง่าย

`my_function ()`; คำสั่งเรียกฟังก์ชันชื่อ `my_function` ที่ไม่ต้องการพารามิเตอร์ และไม่สนใจค่าที่อาจจะส่งออกโดยฟังก์ชันนี้

ฟังก์ชันจำนวนมากได้รับการเรียกด้วยวิธีนี้ เช่น ฟังก์ชัน `phpinfo ()` สำหรับแสดงเวอร์ชันติดตั้งของ PHP สารสนเทศเกี่ยวกับ PHP การตั้งค่าแม่ข่ายเว็บ ค่าต่างๆ ของ PHP และตัวแปร ฟังก์ชันนี้ไม่ใช่พารามิเตอร์และโดยทั่วไปไม่สนใจค่าส่งออก ดังนั้นการเรียก `phpinfo ()` จะประกอบขึ้นดังนี้ `phpinfo ();`

การกำหนดฟังก์ชันและการเรียกฟังก์ชัน

การประกาศฟังก์ชันเริ่มต้นด้วยคีย์เวิร์ด `function` กำหนดชื่อฟังก์ชัน พารามิเตอร์ที่ต้องการ และเก็บคำสั่งที่จะประมวลผลแต่ละครั้งเมื่อเรียกฟังก์ชันนี้

```
<?php
    function function_name(parameter1,...)
    {
        ชุดคำสั่ง ...
    }
```

```
?>
```

ชุดคำสั่งต้องเริ่มต้นและสิ้นสุดในวงเล็บปีกกา (`{ }`) ตัวอย่างฟังก์ชัน `my_function`

การใช้งานฟังก์ชันโดยไม่มีการส่งผ่านค่า

`Function_nopass.php`

```
<?
```

```
function asterisk($no)
{
    $sign = "";
    for($count=1; $count<=$no; $count++)
    {
        $sign = $sign . "*";
    }
    echo "$sign <br>";
}
```

```

        asterisk(30);
        echo "My name is TUM <br>";
        echo "My name is 555<br>";
        echo "Hello everybody <br>";
        asterisk(20);

?>
การใช้งานฟังก์ชันโดยมีการส่งผ่านค่า
Function_pass.php
<?

function asterisk($no)
{
    $sign = "";
    for($count=1; $count<=$no; $count++)
    {
        $sign = $sign "*";
    }
    echo "$sign <br>";
}
asterisk(30);
echo "My name is TUM <br>";
echo "My name is 555<br>";
echo "Hello everybody <br>";
asterisk(20);

?>

```

จากโปรแกรมได้ทำการปรับปรุงฟังก์ชัน asterisk โดยเขียนฟังก์ชันในลักษณะที่มีการส่งผ่านค่าฟังก์ชันอาร์กิวเมนต์ \$no ทำหน้าที่รับค่าจำนวนที่ต้องการพิมพ์เครื่องหมายดอกจัน ภายในฟังก์ชันก็นำอาร์กิวเมนต์ \$no นี้ไปเข้าเงื่อนไขเพื่อทำซ้ำของคำสั่ง for กำหนดตัวแปร \$sign เป็นตัวแปรที่ใช้ต่อสตริงเครื่องหมายดอกจันเท่ากับจำนวนค่าของอาร์กิวเมนต์

ประเภทของตัวแปร และขอบเขตของตัวแปร

การประกาศและกำหนดค่าตัวแปรทั้งวิธีกำหนดค่าปกติหรือกำหนดค่าแบบอ้างอิง ใน PHP สามารถประกาศได้ทุกที่ในสคริปต์ของ PHP ซึ่งตำแหน่งของการประกาศตัวแปรมีผลต่อการใช้งานหรือเรียกว่าขอบเขตของตัวแปร (Variable Scope) จะมีผลต่อการเข้าถึงและใช้งาน ขอบเขตของตัวแปรมี 4 ชนิด คือ ตัวแปรท้องถิ่น (Local variables), ตัวแปรฟังก์ชันพารามิเตอร์ (Function parameters), ตัวแปรสาธารณะ

(Global variables) และตัวแปรคงที่ (Static variables) มีรายละเอียด ดังนี้

ตัวแปรท้องถิ่น

ตัวแปรท้องถิ่น คือ ตัวแปรที่ถูกประกาศไว้ภายในฟังก์ชันนั้น จะมีขอบเขตเฉพาะภายในฟังก์ชันนั้นๆ เท่านั้น หรือถ้าจะพูดให้ง่ายก็คือจะสามารถใช้งานตัวแปรนั้นๆ ได้ภายในฟังก์ชันนั้น ในทางกลับกันถ้าประกาศตัวแปรภายนอกฟังก์ชัน (เป็นขอบเขตตัวแปรสาธารณะ: Global Variables) ก็จะไม่สามารถใช้งานตัวแปรนั้นๆ ภายในฟังก์ชันได้ ตัวอย่างดังนี้

```
<?php
function update_counter ( )
{
    $counter++;
}
$counter = 10; update_counter ( );
echo $counter;
?>
```

ตัวแปรฟังก์ชันพารามิเตอร์

ตัวแปรฟังก์ชันพารามิเตอร์ คือ ตัวแปรต่างๆ ที่ถูกส่งผ่านไปยังฟังก์ชันหนึ่งๆ ซึ่งการส่งค่าพารามิเตอร์นั้นสามารถส่งได้ครั้งละหลายๆ ตัวแปร โดยทำการคั่นแต่ละพารามิเตอร์ด้วยเครื่องหมาย comma (,) สำหรับรูปแบบการส่งค่าจะกล่าวถึงในบทถัดไป ตัวอย่างการการส่งค่าตัวแปรฟังก์ชันพารามิเตอร์ ตัวอย่างดังนี้

```
<?php
function x10 ($value) { // ตัวอย่าง ตัวแปรฟังก์ชันพารามิเตอร์
    $value $value = $value * 10;
    return $value; // มีการส่งค่ากลับตัวแปรฟังก์ชันพารามิเตอร์
}
?>
```

ตัวแปรสาธารณะ

ตัวแปรสาธารณะ คือ ตัวแปรที่ประกาศไว้ภายนอกฟังก์ชัน สามารถเข้าถึงจากส่วนใดๆ ของโปรแกรม แต่จะไม่สามารถใช้งานได้ภายในฟังก์ชัน แต่ถ้าต้องการเรียกใช้งานตัวแปรสาธารณะภายในฟังก์ชันนั้นจริงๆ สามารถใช้คำสั่ง GLOBAL ในการอ้างอิงได้ ตัวอย่างดังนี้

```
<?php
$somevar = 15;
function addit ( )
{
    GLOBAL $somevar;
```

```

    $somevar++;
}
addit ( );
echo "Somevar is $somevar"; // ผลลัพธ์ คือ Somevar is 16
?>

```

ฟังก์ชันของ php ที่จัดเตรียมไว้

ในส่วนนี้จะกล่าวถึงการใช้ฟังก์ชันของ PHP เพื่อจัดการข้อมูลใน 3 รูปแบบคือ สตริง ตัวเลข และวัน เวลา แม้ว่าในบทที่ผ่านมาได้ใช้งานข้อมูลแบบสตริง และตัวเลขกันมาบ้างแล้ว แต่ก็ยังเป็นเพียงการใช้ในแบบพื้นฐานทั่วไป ซึ่งข้อมูลทั้งสองอย่างนี้ยังมีรายละเอียดปลีกย่อยที่จำเป็นต้องรู้อีกมากมาย ส่วนข้อมูลประเภทวันเวลานั้น ก็จัดว่าเป็นข้อมูลพิเศษอีกอย่างหนึ่ง ที่ไม่สามารถใช้โอเปอเรเตอร์ไหนไปจัดการโดยตรงได้ ต้องอาศัยฟังก์ชันเฉพาะทางด้านนี้ของ PHP เท่านั้น

ฟังก์ชันเกี่ยวกับสตริง

สตริง (String) คือ ชุดของตัวอักษร หรือเป็นการนำอักษรแต่ละตัวมาวางเรียงต่อกันเพื่อสื่อความหมาย เช่นคำว่า “String” เกิดจากอักษร S + t + r + i + n + g เป็นต้น ทั้งนี้อักขระที่วางเรียงต่อกันเป็นสตริง ก็จะมีลำดับของมันเองเหมือนกับอาร์เรย์ของสตริงหรือค่านั้นเอง ดังนั้นการอ้างถึงอักขระย่อยๆ แต่ละตัวภายในสตริง จึงใช้เลขลำดับเป็นตัวกำหนดเหมือนกับการอ้างอิงสมาชิกของอาร์เรย์ เช่น

1. addslashes() จะทำการเพิ่มเครื่องหมาย \ (Back Slashes) ให้กับ String ที่มีสัญลักษณ์ ('), ("), (\n)

Syntax

```

<?
    addslashes($string)

```

```

?>

```

Sample

```

<?php

```

```
$str = "Is your name O'reilly?"; // Outputs: Is your name O\`reilly?  
echo addslashes($str);
```

```
?>
```

2. chr() แปลงรหัส ASCII เป็นตัวอักษร

Syntax

```
<?
```

```
    chr($string)
```

```
?>
```

Sample

```
<?php  
echo chr(27)."<br>";  
echo chr(97)."<br>";  
echo chr(98)."<br>";  
echo chr(99)."<br>";  
?>
```

3. ord() แปลงตัวอักษรเป็นรหัส ASCII

Syntax

```
<?
```

```
    ord($string)
```

```
?>
```

Sample

```
<?php  
    echo ord("a")."<br>";  
    echo ord("b")."<br>";  
    echo ord("c")."<br>";  
    echo ord("d")."<br>";  
    echo ord("e")."<br>";  
?>
```

4. explode() แยกข้อความให้อยู่ในรูปแบบของ array โดยใช้เครื่องหมายแยก

Syntax

```
<?
```

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



```
?>
```

Sample

```
<?php
```

```
// Example 1
```

```
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";  
$pieces = explode(" ", $pizza);  
echo $pieces[0]."<br>"; // piece1  
echo $pieces[1]."<br>"; // piece2
```

```
// Example 2
```

```
$data = "foo*:1023:1000::/home/foo:/bin/sh";  
list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $data);  
echo $user."<br>"; // foo  
echo $pass."<br>"; // *
```

```
?>
```

5. implode() รวม Array เป็นข้อความ

Syntax

```
<?
```

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

```
?>
```

Sample

```
<?php
```

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

```
?>
```

6. htmlspecialchars() เปลี่ยน String เป็นแท็ก Html

Syntax

```
<?
```

```
htmlspecialchars($string)
```

```
?>
```

Sample

```
<?php
    $new = htmlspecialchars("<a href='test'>Test</a>", ENT_QUOTES);
    echo $new; // <a href='test'>Test</a>

?>
```

7. **nl2br()** แทนที่ Enter ด้วย `
` ขึ้นบรรทัดใหม่

Syntax

```
<?
    nl2br($string)

?>
```

Sample

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

?>
```

8. **stristr()** ตัดข้อความบางส่วนตั้งแต่ตัวแรกที่พบจนถึงตัวสุดท้าย ทั้งตัวพิมพ์เล็กและพิมพ์ใหญ่

Syntax

```
<?
    stristr($string,$needle)

?>
```

Sample

```
<?php
    $email = 'USER@EXAMPLE.com';
    echo stristr($email, 'e')."<br>"; // outputs ER@EXAMPLE.com
    echo stristr($email, 'e', true)."<br>"; // As of PHP 5.3.0, outputs US

?>
```

9. **strlen()** หาความยาวของข้อความ

Syntax

```
<?
    strlen($value)

?>
```

Sample

```
<?php
    $str = 'abcdef';
```

```
    echo strlen($str)."<br>"; // 6
    $str = ' ab cd ';
    echo strlen($str)."<br>"; // 7
?>
```

10. strrev() เรียงสลับข้อความจากหลังไปหน้า

Syntax

```
<?
    strrev($string)
?>
```

Sample

```
<?php
    echo strrev("Hello world!"); // outputs "!dlrow olleH"
?>
```

11. trim() ใช้ในการตัดช่องว่าง (Trim Space) ทางด้านหน้าและด้านหลังของข้อความ

Syntax

```
<?
    trim($string)
?>
```

Sample

```
<?php
    echo trim(" These are a few words ");
?>
```

12. ltrim() ตัดช่องว่างด้านหน้าข้อความ

Syntax

```
<?
    ltrim($string)
?>
```

Sample

```
<?php
    echo ltrim("\t\t These are a few words :) ... ")."<br>";
    echo ltrim(" These are a few words :) ... ")."<br>";
?>
```

13. rtrim() ตัดช่องว่างด้านหลังของข้อความ

Syntax

```
<?
    rtrim($string)
?>
```

Sample

```
<?php
    echo rtrim("\t\t These are a few words :) ... ")."<br>";
    echo rtrim(" These are a few words :) ... ")."<br>";
?>
```

ฟังก์ชันทางคณิตศาสตร์

1. abs() หาค่าสัมบูรณ์ของจำนวนจริง

Syntax

```
<?
    abs($value)
?>
```

Sample

```
<?php
    echo abs(-4.2)."<br>";
    echo abs(5)."<br>";
    echo abs(-5)."<br>";
?>
```

2. max() ฟังก์ชันใช้สำหรับหาค่าสูงสุดใน Array

Syntax

```
<?
    max($array)
?>
```

Sample

```
<?php
    echo max(1, 3, 5, 6, 7)."<br>"; // 7
    echo max(array(2, 4, 5))."<br>"; // 5
    echo max(0, 'hello')."<br>"; // 0
    echo max('hello', 0)."<br>"; // hello
    echo max(-1, 'hello')."<br>"; // hello
    $val = max(array(2, 4, 8), array(2, 5, 7))."<br>"; // array(2, 5, 7)
    $val = max('string', array(2, 5, 7), 42)."<br>"; // array(2, 5, 7)
?>
```

3. min() ฟังก์ชันใช้สำหรับหาค่าต่ำสุดใน Array

Syntax

```
<?
    min($array)
?>
```

Sample

```
<?php
    echo min(2, 3, 1, 6, 7)."<br>"; // 1
    echo min(array(2, 4, 5))."<br>"; // 2
    echo min(0, 'hello')."<br>"; // 0
    echo min('hello', 0)."<br>"; // hello
    echo min('hello', -1)."<br>"; // -1
    $val = min(array(2, 4, 8), array(2, 5, 1))."<br>"; // array(2, 4, 8)
    $val = min('string', array(2, 5, 7), 42)."<br>"; // string
?>
```

4. pow() หาค่าของเลขยกกำลัง

Syntax

```
<?
    pow(ค่า,ตัวเลขยกกำลัง)
?>
```

Sample

```
<?php
    echo pow(2, 8)."<br>"; // int(256)
```

```

echo pow(-1, 20)."<br>"; // 1
echo pow(0, 0)."<br>"; // 1
echo pow(-1, 5.5)."<br>"; // PHP >4.0.6 NAN
echo pow(-1, 5.5)."<br>"; // PHP <=4.0.6 1.#IND

?>

```

5. round() ฟังก์ชันใช้สำหรับปัดเศษทศนิยม

Syntax

```

<?
    round(ข้อมูลตัวเลข, จำนวนหน่วยทศนิยม)

?>

```

Sample

```

<?php
    echo round(3.4)."<br>"; // 3
    echo round(3.5)."<br>"; // 4
    echo round(3.6)."<br>"; // 4
    echo round(3.6, 0)."<br>"; // 4
    echo round(1.95583, 2)."<br>"; // 1.96
    echo round(1241757, -3)."<br>"; // 1242000
    echo round(5.045, 2)."<br>"; // 5.05
    echo round(5.055, 2)."<br>"; // 5.06

?>

```

ฟังก์ชันเกี่ยวกับวันและเวลา

1. **getdate()** เป็น Function อ่านค่าเวลาและวันที่ปัจจุบันของเครื่อง Server

Syntax

```

<?
    getdate($format);

?>

```

Sample

```

<?php
    $today = getdate();
    echo "Day = ".$today["mday"]."<br>";

```

```

echo "Month = ".$today["mon"]."<br>";
echo "Year = ".$today["year"]."<br>";
echo "<hr>";
foreach($today as $key => $value) {
echo "Key: $key; Value: $value<br>";
}

```

?>

2. **getdate()** เป็น Function อ่านค่าเวลาและวันที่ปัจจุบันของเครื่อง Server

Syntax

<?

```
date($format);
```

?>

Sample

<?php

```
echo date("Y-m-d H:i:s");
```

?>

พารามิเตอร์บางตัวที่สำคัญของฟังก์ชัน date()

พารามิเตอร์บางตัวที่สำคัญที่ใช้ส่งให้ฟังก์ชัน date()

วัน

พารามิเตอร์	คำอธิบาย	ตัวอย่าง
d	วันที่ ตัวเลข 2 ตัว	01 - 31
D	วัน ตัวอักษร 3 ตัว	Mon - Sun
j	วันที่ ตัวไม่มี 0 นำหน้า	1 - 31
l (แอลตัวเล็ก)	วัน เต็มข้อความ	Sunday - Saturday
N	วัน ตัวเลข 1แทนวันจันทร์ -7 แทนวันอาทิตย์	1 - 7
S	ลำดับต่อท้ายของภาษาอังกฤษ	st, nd, rd or th
w	วัน ตัวเลข 0 แทนวันอาทิตย์ -6 แทนวันเสาร์	0 - 6
z	วันของปี เริ่มจาก 0 -365	0 -365

สัปดาห์

พารามิเตอร์	คำอธิบาย	ตัวอย่าง
W	ตัวเลขสัปดาห์ที่เท่าไรของปี เริ่มต้นที่วันจันทร์	42 (สัปดาห์ที่ 42 ของปี)

เดือน

พารามิเตอร์	คำอธิบาย	ตัวอย่าง
F	ข้อความเต็ม ชื่อเดือน	January
m	ตัวเลขบอกเดือน มี 0 นำ	01 - 12
M	ตัวย่อเดือน	Jan
n	ตัวเลขเดือนไม่มี 0 นำ	1 - 12
t	จำนวนของวันในเดือน	28 - 31

ปี

พารามิเตอร์	คำอธิบาย	ตัวอย่าง
L	บอกว่าเป็น leap year หรือไม่	1 เป็น 0 ไม่เป็น
o	ตัวเลข 4 ตัวแทนปี	2003
Y	ตัวเลข 4 ตัวแทนปี	2003
y	ตัวเลข 2 ตัวแทนปี	03 (แทน 2003)

เวลา

พารามิเตอร์	คำอธิบาย	ตัวอย่าง
a	am หรือ pm ตัวเล็ก	am
A	AM หรือ PM ตัวเล็ก	AM
B	Swatch Internet time	000 through 999
g	รูปแบบชั่วโมง 12 ชั่วโมง ไม่มี 0 นำ	1 - 12
G	รูปแบบชั่วโมง 24 ชั่วโมง ไม่มี 0 นำ	0 - 23

h	รูปแบบชั่วโมง 12 ชั่วโมง มี 0 นำ	01 - 12
H	รูปแบบชั่วโมง 24 ชั่วโมง มี 0 นำ	00 - 23
i	นาที มี 0 นำ	00 - 59
s	วินาที มี 0 นำ	00 - 59
u	Microseconds (added in PHP 5.2.2)	Example: 654321

การจัดการไฟล์และไดเรกทอรี

ไดเรกทอรี (Directory) คือ พื้นที่บนคอมพิวเตอร์ที่เราใช้เก็บไฟล์ (หรือไดเรกทอรีอื่นๆ) ไว้ภายใน หากจะเปรียบคอมพิวเตอร์เป็นประเทศ ไดเรกทอรีก็เหมือนกับจังหวัด ซึ่งภายในสามารถแบ่งย่อยเป็นอำเภอ หรือเล็กกลงกว่านั้นได้ ตัวไฟล์ที่เก็บก็อาจจะเหมือนกับคนที่อยู่ในจังหวัดหรืออำเภอใดๆ ในประเทศ ซึ่งเราต้อง ระบุตำแหน่งที่อยู่นั้นให้ได้ (สำหรับใน Windows ไดเรกทอรีก็เทียบได้กับโพลเดอร์ต่างๆ ในคอมพิวเตอร์ของ เรานั้นเอง)

ในการจัดการกับไดเรกทอรี จะมีขอบเขตตั้งแต่การเปิดไดเรกทอรี, การอ่านไดเรกทอรีว่าภายในไดเรก ทอรีนั้นๆ มีไฟล์หรือไดเรกทอรีใดๆ อยู่บ้าง, การสร้างไดเรกทอรีเพื่อบรรจุไฟล์ที่สร้างขึ้นมา, การปิดไดเรกทอรี และการลบไดเรกทอรี

การเปิดและปิดไฟล์

ใช้ฟังก์ชัน `fopen()` ใช้เปิดไฟล์ข้อมูล มีรูปแบบการใช้งานคือ `fopen(ตำแหน่งและชื่อไฟล์,โหมดการ เปิด)`ค่าที่ได้จาก `fopen()` คือหมายเลขอ้างอิงไฟล์ ซึ่งเราต้องนำไประบุ เมื่อต้องการใช้ฟังก์ชันเกี่ยวกับไฟล์อื่น ๆ ดำเนินการกับไฟล์นั้น มี 6 ชนิดคือ

r เปิดเพื่ออ่านอย่างเดียว

r+ เปิดเพื่ออ่านและเขียนไฟล์

w เปิดเพื่อเขียนไฟล์ โดยเนื้อหาทั้งหมดของไฟล์จะถูกลบทิ้งไป

w+ เปิดเพื่ออ่านและเขียนไฟล์ โดยเนื้อหาทั้งหมดจะถูกลบทิ้ง

a เปิดเพื่อเขียนต่อท้ายไฟล์

a+ เปิดเพื่ออ่านและเขียนต่อท้ายไฟล์

ฟังก์ชัน `fclose()` ใช้ปิดไฟล์ รูปแบบการทำงานคือ `fclose(หมายเลขอ้างอิงของไฟล์)`การอ่านข้อมูล จากไฟล์ การอ่านข้อมูลจากไฟล์ ฟังก์ชัน `fpasssthru()` เป็นฟังก์ชันที่ใช้อ่านข้อมูลทั้งหมดจากไฟล์ แล้วส่งไปยัง เว็บเบราว์เซอร์ทันที ซึ่งการใช้คำสั่งนี้ไม่จำเป็นต้องจบด้วย `fclose`

ตัวอย่าง

```
<?php
    $filenum=fopen("c:/autoexec.bat","r");
    $mydata=fread($filenum,20);
    echo "$mydata
    \n";
    fclose($filenum)
?>
```

การอ่านข้อมูลจากไฟล์

file() อ่านข้อมูลออกมาเป็นแอสไค ตามบรรทัด (เข้ากับข้อมูลด้านบนได้ดี)

fread() อ่านข้อมูลจากไฟล์ที่เราได้ทำการเปิด

fgets() อ่านไฟล์ ตรงกันข้ามกับ fwrite() ที่ผมใช้ในตัวอย่างละครับ

fpasssthru() อ่านข้อมูลที่เก็บไว้ในไฟล์มาแสดง

ฟังก์ชัน fread()

รูปแบบ fread(หมายเลขอ้างอิงไฟล์, จำนวนไบต์ที่ต้องการ)

fread.php

```
<?
    $handle = fopen('c:/windows/wiin.ini','r');
    $content = fread($handle, 50);
    fclose($handle);
?>
```

ฟังก์ชัน fgets()

รูปแบบ fread(หมายเลขอ้างอิงไฟล์, จำนวนไบต์ที่ต้องการ)

info.txt

Hello

My name in Jenrop

I am teacher at Demonstration of Khon Kaen University

Faculty of Education, Khon Kaen University

info.txt

```
<?
```

```
$handle = fopen("info.txt", "r");
```

```
if ($handle)
{
    while (!feof($handle))
    {
        $buffer = fgets($handle, 10);
        echo "$buffer <br>";
    }
    fclose($handle);
}
?>
```

5

Regular Expression และการจัดการภาพกราฟิก

Regular Expression

Meta – Character ใน Regular Expressions คือ การนำเอาตัวอักษรพิเศษ มาใช้งานในการกำหนด รูปแบบ pattern ที่มีความซับซ้อน โดยจะแบ่ง Meta-characters ออกเป็น 2 ส่วน คือ

1. ส่วนที่ใช้งานในการกำหนดรูปแบบ pattern ในทุกที่ ยกเว้นในเครื่องหมายปีกกาสี่เหลี่ยม []
2. ส่วนที่ใช้งานในการกำหนดรูปแบบ pattern ได้ทั้งข้างในและนอก เครื่องหมายปีกกาสี่เหลี่ยม []

Meta – Character ได้แก่

\ ใช้เป็น escape character

^ ใช้สำหรับกำหนด ตำแหน่งเริ่มต้นของข้อความ เช่น "^PHP" จะตรงกับข้อความที่ต้องขึ้นต้นด้วย PHP เท่านั้น

\$ ใช้สำหรับกำหนด ตำแหน่งสิ้นสุด เช่น "PHP\$" จะตรงกับข้อความที่ต้องลงท้ายด้วย PHP เท่านั้น

. ใช้สำหรับแทนตัวอักษรใด ๆ 1 ตัวอักษร เช่น "P.P" จะตรงกับ "PHP" แต่จะไม่ตรงกับ "PHHP"

เนื่องจากต้องการตัวอักษรตัวใดก็ได้ที่อยู่ตรงกลาง

[เริ่มต้นการกำหนดระดับอักษรที่เป็นไปได้ในเงื่อนไข

] สิ้นสุดการกำหนดระดับอักษร

| เป็นสัญลักษณ์สำหรับเป็นทางเลือก

(ใช้สำหรับเริ่มจัดกลุ่มตัวอักษร

) ใช้สำหรับสิ้นสุดการจัดกลุ่มตัวอักษร

? แทนตัวอักษร แต่จะสามารถปรากฏ หรือไม่ปรากฏก็ได้ 1 ครั้ง

* แทนตัวอักษร แต่จะสามารถปรากฏ หรือไม่ปรากฏ ได้หลาย ๆ ครั้ง เช่น "wh*" จะตรงกับ "w"

และ "where"

+ แทนตัวอักษร ที่จะต้องปรากฏ 1 ครั้ง หรือปรากฏได้หลายครั้ง เช่น "wh+" จะตรงกับ "where"

แต่จะตรงกับ "w"

{ เริ่มตรวจสอบจำนวนตัวอักษร

} สิ้นสุดการตรวจสอบตัวอักษร

- ใช้สำหรับระบุช่วงตัวอักษร ต้องใช้ภายใต้ [] เท่านั้น

การใช้งาน [และ]

การใช้งาน [และ] เป็นการตรวจสอบระดับตัวอักษรที่เป็นไปได้ในเงื่อนไข จะสามารถตรวจสอบได้เพียง 1 ตัวอักษร ตัวอย่าง "[abcde]" จะตรงกับ "a" แต่จะไม่ตรงกับ "f" และ "g"

การใช้งาน { และ }

เราสามารถระบุจำนวนของตัวอักษร การใช้งานของ {} สามารถใช้ได้ 3 แบบ คือ

{n} n แทนตัวเลขที่ปรากฏได้ หมายถึง ต้องมีตัวอักษรปรากฏเท่ากับจำนวนตัวเลขที่ระบุอยู่ เช่น "{3}" จะต้องมี ตัวอักษรหรือตัวเลขปรากฏ 3 ตัวอักษร ซึ่งจะตรงกับ "are" แต่จะไม่ตรงกับ "were"

{n,} n แทนตัวเลขที่ปรากฏได้ หมายถึง ต้องมีตัวอักษรปรากฏอยู่ มากกว่าหรือเท่ากับ จำนวนตัวเลขที่ระบุอยู่ เช่น "{3,}" จะต้องมี อักษรปรากฏอย่างน้อย 3 ตัวอักษร ซึ่งจะตรงกับ "are" และ "were" แต่จะไม่ตรงกับ "do"

{n,m} n แทนตัวเลขที่ปรากฏได้ m แทนตัวเลขที่ปรากฏได้สูงสุด หมายถึง ต้องมีตัวอักษรปรากฏอยู่ อย่างน้อย n ตัว แต่ต้องไม่เกิน m ตัว เช่น "{3,5}" ซึ่งจะตรงกับ "are" และ "where" แต่จะไม่ตรงกับ "do"

การใช้งาน [] ร่วมกับ {}

การนำเอา "[]" มาใช้งานร่วมกับ "{}" ก็จะทำให้ "[]" ที่ตรวจสอบตัวอักษรได้แต่ 1 ตัวอักษร ก็ทำให้สามารถตรวจสอบ ได้หลาย ตัวอักษร เราสามารถใช้ "-" เช่น "[a-z]{3}" จะตรงกับ "who" และ "are" ซึ่งตรงตามเงื่อนไขก็คือ สามารถมี a - z แต่ต้องมีขนาด 3 ตัวอักษร แต่จะไม่ตรงกับ "2bb" เนื่องจากมี ตัวเลขปรากฏอยู่

การใช้ () และ |

เราสามารถใช้ () จัดกลุ่มข้อความที่มีอยู่ และ | ใช้สำหรับเลือกทางเลือก เช่น "a(b|c)" จะตรงกับ "ab" และ "ac" แต่จะไม่ตรงกับ "bc" ลองดูตัวอย่างที่มีการทำงานซับซ้อนมากขึ้นอีกนะครับ " $^{\wedge}[a-z]+[a-z]^{\$}$ " ความหมายคือ ต้องขึ้นต้นด้วยตัวอักษร และ ลงท้ายด้วยตัวอักษร อย่างน้อย 1 ตัว " $^{\wedge}[a-z-]" ความหมายคือ ต้องขึ้นต้นด้วยตัวอักษร หรือ "-" ตัวอย่างของ Regex ที่มีใช้บ่อย ๆ " $^{\wedge}[a-z0-9_\\-]+(\\.[a-z0-9\\-]+)*@[a-$$

z0-9\-\.\.)([a-

z]{2}|aero|arpa|biz|com|coop|edu|gov|info|int|jobs|mil|museum|name|nato|net|org|pro|travel)">\$

ใช้สำหรับตรวจสอบ email "`^([0-9]{1-9}[0-9]{1-9}|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.)\{3\}([0-9]{1-9}[0-9]{1-9}|1[0-9]{2}|2[0-4][0-9]|25[0-5])$"` ใช้สำหรับตรวจสอบ IP Address

ฟังก์ชันที่เกี่ยวข้องกับ Regular Expression

ฟังก์ชัน `ereg()` และ `eregi()`

ฟังก์ชันทั้งสองนี้ใช้สำหรับตรวจสอบข้อความที่ต้องการตรวจสอบมีรูปแบบตามที่กำหนดไว้หรือไม่
 ต่างกันที่ฟังก์ชัน `ereg()` จะถือว่าข้อความที่ตรวจสอบนั้น ตัวอักษรภาษาอังกฤษทั้งตัวพิมพ์ใหญ่และตัวพิมพ์
 เล็กนั้นแตกต่างกัน (Case-Sensitive) ส่วนฟังก์ชัน `eregi()` นั้นจะถือว่าข้อความที่ตรวจสอบ ตัวอักษร
 ภาษาอังกฤษทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็กไม่แตกต่างกัน (Case-Insensitive) รูปแบบการใช้งานฟังก์ชันทั้ง
 สองเป็น ดังนี้

รูปแบบ

`ereg(รูปแบบข้อความ, ข้อความที่ต้องการตรวจสอบ)`

`eregi(รูปแบบข้อความ, ข้อความที่ต้องการตรวจสอบ)`

`ereg.php`

<?

```
function datecheck($date)
```

```
{
```

```
    if (ereg ("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date))
```

```
    {
```

```
        echo "Valid date format: $date <br>";
```

```

    }
    else
    {
        echo "Invalid date format: $date <br>";
    }
}

$d1 = "2009-01-10";
$d2 = "01-10-2009";
$d3 = "2009:01:10";
datecheck($d1);
datecheck($d2);
datecheck($d3);

?>

```

ผลลัพธ์ของโปรแกรมข้างต้นต้องการตรวจสอบรูปแบบของวันเดือนปี โดยรูปแบบที่ถูกต้องคือ YYYY-DD-MM (ตัวเลข 0-9 จำนวน 4 ตัว-ตัวเลข 0-9 จำนวน 2 ตัว-ตัวเลข 0-9 จำนวน 2 ตัว) ทดสอบด้วยข้อมูล 3 ชุด คือ

| | |
|------------|---|
| 2009-01-10 | เป็นรูปแบบที่ถูกต้อง |
| 01-10-2009 | ไม่ถูก เพราะชุดแรกต้องการตัวเลข 4 ตัว |
| 2009:01:10 | ไม่ถูก เพราะเครื่องหมายคั่นตัวเลขที่ต้องการคือ - ไม่ใช่ : |

การตรวจสอบ E-mail

รูปแบบ

username@hostname.domainname

เช่น username@mydomain.com หรือ username@mydomain.co.th ดังนั้นการกำหนด

รูปแบบของ E-mail เพื่อตรวจสอบสามารถกำหนดได้ ดังนี้

```
^[^\.\\$\_\\`"<>].+[^\.\\$\_\\`"][:space:]<>]@[^\.\\$\_\\`"][:space:]<>".+\\..+[^\.\\$\_\\`"<>]`$
```

เมื่อแยกออกเป็นส่วนๆ ได้ดังนี้

^[^\.\\\$_\\`"<>] กำหนดว่าต้องขึ้นต้นไม่เป็นเครื่องหมาย . \$ _ ‘ “ < >

| | |
|---------------------|---|
| + | ต้องประกอบไปด้วยตัวอักษร 1 ตัวขึ้นไป ไม่มีเลยไม่ได้ |
| [^\\.\\$_\'\">]@ | ก่อนหน้าเครื่องหมาย @ ห้ามเป็นเครื่องหมาย . \$ _ ' " < > ช่องว่าง |
| [^\\.\\$_\'\">"] | ต่อจากเครื่องหมาย @ ห้ามเป็นเครื่องหมาย . \$ _ ' " < > ช่องว่าง |
| + | ต้องประกอบไปด้วยตัวอักษร 1 ตัวขึ้นไป ไม่มีเลยไม่ได้ |
| \\. | เครื่องหมาย . |
| + | ต้องประกอบไปด้วยตัวอักษร 1 ตัวขึ้นไป ไม่มีเลยไม่ได้ |
| [^\\.\\$_\'\">]{\$ | กำหนดว่าต้องไม่ลงท้ายด้วยเครื่องหมาย . \$ _ ' " < > |

ให้นักเรียนลองพิจารณาโปรแกรมตรวจสอบ E-mail ต่อไปนี้ ว่าเป็นรูปแบบที่ถูกตั้งหรือไม่

ereg2.php

```
<?
function Emailcheck($Email)
{
    if (ereg ("^[^\\.\$\_\'\">].+[^\\.\$\_\'\">"][:space:]<>]@[^\\.\$\_\'\">"][:space:]<>"].+\\.+[^\\.\$\_\'\">]{$", $Email))
    {
        echo "Valid E-mail format: $Email <br>";
    }
    else
    {
        echo "Invalid E-mail format: $Email <br>";
    }
}

$mail1 = "mymail@localhost.com";
$mail2 = "mymail$@localhost.com";
$mail3 = "mymail@ localhost.com";
$mail4 = "mymail@localhost";
$mail5 = "mymail@localhost._";
```

```
Emailcheck($mail1);  
Emailcheck($mail2);  
Emailcheck($mail3);  
Emailcheck($mail4);  
Emailcheck($mail5);
```

```
?>
```

ฟังก์ชัน `ereg_replace` และ `eregi_replace()`

ฟังก์ชันทั้งสองนี้ใช้สำหรับแทนที่คำหรือข้อความในสตริงตามรูปแบบที่กำหนดไว้ ต่างกันที่ฟังก์ชัน `ereg_replace` จะถือว่าข้อความที่ตรวจสอบนั้น ตัวอักษรภาษาอังกฤษทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็กนั้นแตกต่างกัน (Case-Sensitive) ส่วนฟังก์ชัน `eregi_replace()` นั้นจะถือว่าข้อความที่ตรวจสอบ ตัวอักษรภาษาอังกฤษทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็กไม่แตกต่างกัน (Case-Insensitive) รูปแบบการใช้งานฟังก์ชันทั้งสองเป็น ดังนี้

รูปแบบ

`ereg_replace`(รูปแบบที่ใช้ค้นหา, คำหรือข้อความที่ใช้แทนที่, ตัวแปรสตริงที่ใช้เก็บข้อมูลหลังจากแทนที่)

`eregi_replace`(รูปแบบที่ใช้ค้นหา, คำหรือข้อความที่ใช้แทนที่, ตัวแปรสตริงที่ใช้เก็บข้อมูลหลังจากแทนที่)

นักเรียนลองพิจารณาโปรแกรมต่อไปนี้

`ereg_replace.php`

```
<?
```

```
$num = 4;  
$string = "This string has four words.";  
$string = ereg_replace('four', $num, $string);
```

```
echo $string. "<br>";

$num = '4';
$string = "This string has four words.";
$string = ereg_replace('four', $num, $string);
echo $string;

?>
```

ฟังก์ชัน split()

ฟังก์ชัน split() นี้ใช้สำหรับแบ่งข้อความที่ต้องการออกเป็นส่วนๆ โดยทำการกำหนดว่าจะใช้รูปแบบหรือข้อความอะไรเป็นตัวแบ่ง และผลลัพธ์จากการใช้ฟังก์ชันนี้จะเก็บที่ตัวแปรอาร์เรย์ในแต่ละช่องเริ่มจากอาร์เรย์ช่องที่ 0

รูปแบบ

split(รูปแบบหรือข้อความที่ใช้เป็นตัวแบ่ง, ข้อความต้นฉบับ, จำนวนครั้งที่ต้องการแบ่ง)
นักเรียนลองพิจารณาโปรแกรมต่อไปนี้

split.php

```
<?

$string = "aBBBaCCCADDDaEEeGGGA";
$chunks = split ("a", $string, 5);
for($i=0; $i<count($chunks); $i++)
{
    echo "$chunks[$i] <br>";
}

?>
```

PHP กับการจัดการภาพกราฟิก

ฟังก์ชัน Header()

รูปแบบ

header(สตริงที่กำหนดเฮดเดอร์ บอกประเภทข้อมูล)

ตัวอย่าง header("Content-type: image/png")

ฟังก์ชัน ImageCreate() สร้างรูปภาพบนบราวเซอร์

รูปแบบ

ImageCreate(int \$width . int \$height)

ฟังก์ชัน ImageColorAllocate() กำหนดค่าสี โดยระบุค่าแม่สี RGB

รูปแบบ

ImageColorAllocate(resource \$image, int \$red, int \$green, int \$blue)

ฟังก์ชัน ImagePNG() ImageJPEG() ImageGIF() ส่งข้อมูลรูปภาพไปแสดงบนบราวเซอร์

ฟังก์ชันขึ้นอยู่กับตั้งแต่เริ่มต้น

รูปแบบ

ImagePNG(\$image)

ImageJPEG(\$image)

ImageGIF(\$image)

ฟังก์ชัน ImageDestroy() คืนค่าหน่วยความจำในการสร้างภาพแก่ระบบ

รูปแบบ

ImageDestroy(resource \$image)

ฟังก์ชัน ImageString() เขียนข้อความลงบนรูปภาพ

รูปแบบ

ImageString(resource \$image, int \$font, int \$x, int \$y, string \$string, int \$color)

ฟังก์ชัน ImageChar() เขียนข้อความลงบนรูปภาพ แต่เขียนได้เพียง 1 ตัวเท่านั้น

รูปแบบ

ImageChar(resource \$image, int \$font, int \$x, int \$y, string \$string, int \$color)

6

การเขียนโปรแกรมเชิงวัตถุ
และการสร้างช่องรับข้อมูล

การเขียนโปรแกรมเชิงวัตถุ

การเขียนโปรแกรมเชิงวัตถุ (Object-oriented programming, OOP) คือหนึ่งในรูปแบบการเขียนโปรแกรมคอมพิวเตอร์ ที่ให้ความสำคัญกับ วัตถุ ซึ่งสามารถนำมาประกอบกันและนำมาทำงานรวมกันได้ โดยการแลกเปลี่ยนข่าวสารเพื่อนำมาประมวลผลและส่งข่าวสารที่ได้ไปให้ วัตถุ อื่นๆที่เกี่ยวข้องเพื่อให้ทำงานต่อไป

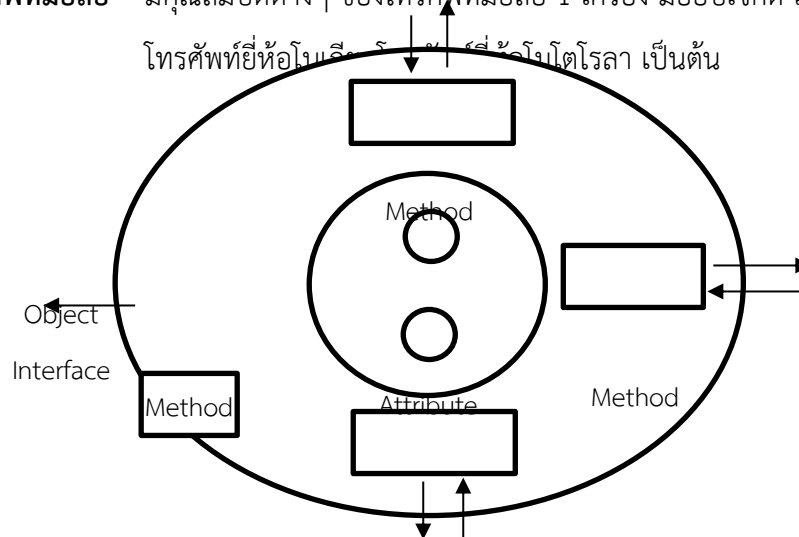
Object Oriented Programming หรือ OOP หรือ การเขียนโปรแกรมเชิงวัตถุ หมายถึง การเขียนโปรแกรมโดยการมองว่าโปรแกรมใดๆเป็นวัตถุชนิดหนึ่ง ที่ประกอบไปด้วย คุณสมบัติ (property) และ วิธีการ (method) เช่น คลาสรถ ประกอบด้วย คุณสมบัติ สี จำนวนล้อ และ วิธีการ เช่น วิธีขับ วิธีจอด เป็นต้น นอกจากนี้ OOP ที่ดี ยังต้องมีคุณสมบัติการสืบทอด (inherited) เช่น คลาสรถยนต์ สืบทอดมาจากคลาสรถ และ คลาสรถมอเตอร์ไซด์ ก็สืบทอดมาจากคลาสรถเช่นเดียวกัน แต่ทั้ง 2 คลาส(ซึ่งเรียกว่าคลาสลูก หรือ sub class) ก็อาจมีคุณสมบัติบางอย่างที่แตกต่างจาก คลาสรถ(คลาสแม่ หรือ base class) เช่น คลาสรถยนต์ มี 4 ล้อ และ คลาสรถมอเตอร์ไซด์ มี 2 ล้อ เป็นต้นโปรแกรมเชิงวัตถุ OOP ช่วยให้เขียนโปรแกรมได้อย่างสง่างาม ตรวจสอบได้ง่าย สามารถนำโค้ดที่เขียนไว้ กลับมาใช้ใหม่ได้

คลาสและออบเจกต์ (Class and Object)

คลาส (Class) คือกลุ่มของสิ่งของ หรือกลุ่มของออบเจกต์ ส่วนออบเจกต์ (Object) คือวัตถุที่เป็นเสมือนตัวแทนของคลาสนั้นๆ โดยสามารถแบ่งได้เป็นทั้งวัตถุที่เป็นรูปธรรม (จับต้องได้) เช่น จักรยาน รถ สุนัข เป็นต้น และนามธรรม (จับต้องไม่ได้) เช่น เพลง วิชาเรียน เป็นต้น ที่มีอยู่ในโลกความจริง (Real World)

ความสัมพันธ์ระหว่างคลาสและออบเจกต์ คือ การสร้างออบเจกต์ใดๆ ได้จะต้องสร้างมาจากคลาสซึ่งเป็นเสมือนกลุ่มของประเภทและคุณสมบัติของออบเจกต์นั้น เช่น

- คลาสคน มีคุณสมบัติต่างๆ ของคน 1 คน มีออบเจกต์ เช่น นายแดง, นายสมชาย
- คลาสรถยนต์ มีคุณสมบัติต่างๆ ของรถยนต์ 1 คัน มีออบเจกต์ เช่น รถฮอนด้า, รถโตโยต้า, รถนิสสัน เป็นต้น
- คลาสโทรศัพท์มือถือ มีคุณสมบัติต่างๆ ของโทรศัพท์มือถือ 1 เครื่อง มีออบเจกต์ เช่น โทรศัพท์ยี่ห้อโนเกีย, โทรศัพท์ยี่ห้อไอโฟน เป็นต้น





Method

ออบเจ็กต์ทุกออบเจ็กต์จะมีส่วนประกอบ 2 ส่วน คือ คุณสมบัติ (Attribute/Property) และการดำเนินการหรือพฤติกรรม (Method/Behavior) พิจารณาจากภาพด้านบน

จากรูป ออบเจ็กต์นี้ประกอบไปด้วย 2 แอตทริบิวต์ และ 4 เมธอด โดยจะถือว่าแอตทริบิวต์ในออบเจ็กต์จะเป็นตัวแทนของตัวแปรของข้อมูล ส่วนเมธอดในออบเจ็กต์จะเป็นตัวดำเนินการโดยอาจจะมีการรับข้อมูลเข้าเพื่อทำกระบวนการต่างๆ โดยใช้การผ่านข้อมูล (Message Passing) โดยปกติโปรแกรมเมอร์จะใช้แผนภาพของคลาส (Class Diagram) ในการอธิบายลักษณะของคลาส โดยจะใช้รูปสี่เหลี่ยมแบ่งเป็น 3 ส่วน คือ ส่วนบนใช้แสดงชื่อคลาส ส่วนกลางใช้แสดงแอตทริบิวต์ต่างๆ ของคลาส

| |
|------------|
| Class name |
| Attribute |
| Method |

ยกตัวอย่างเช่นคลาส Employee ซึ่งเป็นคลาสของพนักงาน ภายในคลาสนี้จะประกอบไปด้วยแอตทริบิวต์ที่ใช้แสดงถึงคุณสมบัติต่างๆ ที่พนักงานทุกคนพึงมี เช่น ชื่อ-สกุล (name), เงินเดือน (Salary), และค่าล่วงเวลา (ot) ในส่วนของเมธอดใช้แสดงถึงกิจกรรมหรือการดำเนินการของคลาสนี้ เช่น เมธอดการกำหนดข้อมูลของพนักงาน (Employee()), เมธอดการแสดงผลข้อมูลของพนักงาน (Print_EmployeeData()) คลาสของ

| |
|------------------------------------|
| Employee |
| name
salary
ot |
| Employee()
Print_EmployeeData() |

จากรูปแสดงถึงคลาส Class Diagram ของคลาส Employee ซึ่งจะทำให้เราเห็นส่วนประกอบของคลาสนี้ โดยประกอบไปด้วย 3 แอตทริบิวต์ คือ name, salary และ ot มีเมธอด คือ Employee() และ Print_EmployeeData() ตามที่ระบุไว้ใน Class Diagram

ตัวอย่างต่อไปนี้เป็นสร้างคลาส Employee ด้วยภาษา PHP

```
<?>
```

```
class Employee {
```



```
var $name;
var $salary;
var $ot;

function Employee($name,$salary,$ot) {
    $this->name = $name;
    $this->salary = $salary;
    $this->ot = $pt;
}

function Print_EmployeeData(){
    echo "ชื่อ-สกุลพนักงาน : $this->name <br>";
    echo "เงินเดือน : $this->salary <br>";
    echo "ค่าล่วงเวลา : $this->ot <br>";
    echo "เงินประจำตำแหน่ง : $this->position_salaty <br>";
}
}
?>
```

การสร้างออบเจ็กต์และการใช้งานเมธอด

จากที่กล่าวไว้ในหัวข้อที่ผ่านมาว่าออบเจ็กต์ต่างๆ จะสร้างมาจากคลาส โดยเมื่อออบเจ็กต์ถูกสร้างขึ้นแล้ว ออบเจ็กต์นั้นจะมีคุณสมบัติของคลาสนั้นด้วย คือออบเจ็กต์นั้นจะมีแอตทริบิวต์และสามารถใช้เมธอดต่างๆ ที่อยู่ภายในคลาสนั้นได้ รูปแบบการสร้างออบเจ็กต์ การเรียกใช้แอตทริบิวต์และการใช้งานเมธอดของออบเจ็กต์จะเป็นดังนี้

| | |
|------------------------|---|
| การสร้างออบเจ็กต์ | <code>[\$ชื่อออบเจ็กต์ หรือตัวแปร Instance] = new ชื่อคลาส</code> |
| การเรียกใช้แอตทริบิวต์ | <code>[\$ชื่อออบเจ็กต์ หรือตัวแปร Instance] ->Attribute</code> |
| การใช้งานเมธอด | <code>[\$ชื่อออบเจ็กต์ หรือตัวแปร Instance] ->Method()</code> |

จากตัวอย่างคลาสในหัวข้อที่ผ่านมา เราได้ทำการสร้างคลาส Employee ขึ้นมา แต่เรายังไม่สามารถใช้งานคลาสนี้ได้ เราจะต้องทำการสร้างออบเจกต์ขึ้นก่อน ดังนั้นจากโปรแกรมที่ผ่านมาเราจะทำการเพิ่มเติมในส่วนของการสร้างออบเจกต์ การเรียกใช้แอตทริบิวต์และการใช้งานเมธอดดังนี้

```
$somchai = new Employee(สมชัย,15000,3500);
$somchai->Print_EmployeeData();
echo "*****<p>";
$precha = new Employee(ปรีชา,25000,5000);
$precha->Print_EmployeeData();
```

โดยเริ่มจากการสร้างออบเจกต์ชื่อ \$somchai โดยมีการผ่านค่าไป 3 ตัว คือ สมชัย,15000 และ 3500 ซึ่งจะมีการเรียกใช้ constructor คือเมธอด Employee() โดยอัตโนมัติค่าที่ผ่านไปทั้ง 3 ตัวก็จะนำไปกำหนดค่าให้กับ ชื่อ-สกุลพนักงาน, เงินเดือน และค่าล่วงเวลา ตามลำดับ หลังจากนั้นออบเจกต์ \$somchai นี้ก็ได้มีการเรียกใช้เมธอด Print_EmployeeData() เพื่อพิมพ์รายละเอียดของพนักงานที่ชื่อสมชัยออกมา ต่อจากนั้นเราได้ทำการสร้างออบเจกต์ที่สองขึ้นมาคือ \$precha

ดังนั้นโปรแกรมโดยสมบูรณ์ทั้งหมดจะเป็นดังนี้

employee.php

```
<?
class Employee {
    var $name;
    var $salary;
    var $ot;

    function Employee($name,$salary,$ot) {
        $this->name = $name;
        $this->salary = $salary;
        $this->ot = $ot;
    }

    function Print_EmployeeData(){
        echo "ชื่อ-สกุลพนักงาน : $this->name <br>";
```

```

        echo "เงินเดือน : $this->salary <br>";
        echo "ค่าล่วงเวลา : $this->ot <br>";
        echo "เงินประจำตำแหน่ง : $this->position_salaty <br>";
    }
}

$somchai = new Employee(สมชัย,15000,3500);
$somchai->Print_EmployeeData();
echo "*****<p>";
$precha = new Employee(ปรีชา,25000,5000);
$precha->Print_EmployeeData();

?>

```

ผลลัพธ์ของโปรแกรมข้างต้นเป็นดังนี้

```

<-- --> localhost/Test/method.php
ชื่อ-สกุลพนักงาน : สมชัย
เงินเดือน : 15000
ค่าล่วงเวลา : 3500
เงินประจำตำแหน่ง :
*****

ชื่อ-สกุลพนักงาน : ปรีชา
เงินเดือน : 25000
ค่าล่วงเวลา : 5000
เงินประจำตำแหน่ง :

```

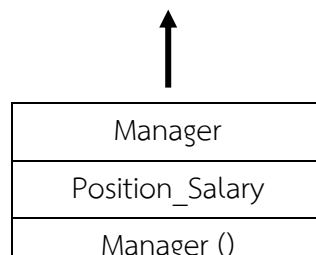
คุณสมบัติการสืบทอด (Inheritance)

การสืบทอด (Inheritance) หมายถึง การที่คลาสหนึ่งถูกสร้างขึ้นโดยการสืบทอดมาจากอีกคลาสหนึ่ง ผลจากการสืบทอดจะทำให้คลาสที่เป็นผู้สืบทอดมีองค์ประกอบทั้งหมดเหมือนกับคลาสต้นแบบ เรียกคลาสที่เป็นคลาสต้นแบบว่า Parent Class ส่วนคลาสที่เป็นผู้ทำการสืบทอดมาเรียกว่า Sub Class ทั้งนี้การสืบทอดจะใช้คีย์เวิร์ด extends เป็นตัวกำหนด

จากตัวอย่างที่ผ่านมาเรามีคลาสพนักงาน (Class Employee) ซึ่งประกอบด้วย 3 แอตทริบิวต์ และ 2 เมธอด ต่อมาได้ทำการสร้างคลาสขึ้นมาอีกหนึ่งคลาสชื่อว่า คลาสผู้จัดการ (Class Manager) โดยทำการสืบทอดมาจากคลาสพนักงาน ดังนั้นคลาสผู้จัดการนี้จะมีคุณสมบัติทุกอย่างที่คลาสพนักงานมี คือ 3 แอตทริบิวต์ และ 2 เมธอด เช่นเดียวกัน นอกจากนั้นคลาสผู้จัดการนี้ยังมีคุณสมบัติที่คลาสพนักงานไม่มีก็คือ การที่พนักงานที่มีตำแหน่งเป็นผู้จัดการจะมีเงินเดือนประจำตำแหน่งเพิ่มขึ้นอีกด้วย โดยที่พนักงานทั่วไปจะไม่มีเงิน

| |
|----------|
| Employee |
|----------|

ในส่วนนี้ ดังนั้นคลาสผู้จัดการจะมีการสร้างแอตทริบิวต์เพิ่มอีก 1 ตัว คือ position_salary ใช้เก็บค่าเงินประจำตำแหน่ง และในส่วนของ Constructor ของคลาส Manager นี้จะเป็นการผ่านค่าของเงินประจำตำแหน่ง เพื่อนำไปกำหนดค่าให้กับวัตถุที่สร้างจากคลาส Manager



ตัวอย่างโปรแกรมต่อไปนี้ เราจะแสดงให้เห็นวิธีการที่คลาส Manager สืบทอดคุณสมบัติของคลาส Employee() รวมทั้งการสร้างออบเจกต์จากคลาส Manager และการใช้งานแอตทริบิวต์และเมธอดต่างๆ จากออบเจกต์ของคลาส Manager

manager.php

<?

```
require('employee.php');
class Manager extends Employee {
    var $position_salary;

    function Manager($position_salary){
        $this->position_salary = $position_salary;
    }
}
echo "*****<p>";
$chanchai = new Manager(15000);
```

```
$chanchai->Employee(ชาญชัย,35000,5000);
```

```
$chanchai->Print_EmployeeData();
```

```
?>
```

ตัวอย่างผลลัพธ์จากโปรแกรมข้างต้น

```

<-- --> C localhost/Test/manager.php
ชื่อ-สกุลพนักงาน : สมชัย
เงินเดือน : 15000
ค่าล่วงเวลา : 3500
เงินประจำตำแหน่ง :
*****

ชื่อ-สกุลพนักงาน : ปรีชา
เงินเดือน : 25000
ค่าล่วงเวลา : 5000
เงินประจำตำแหน่ง :
*****

ชื่อ-สกุลพนักงาน : ชาญชัย
เงินเดือน : 35000
ค่าล่วงเวลา : 5000
เงินประจำตำแหน่ง : 15000

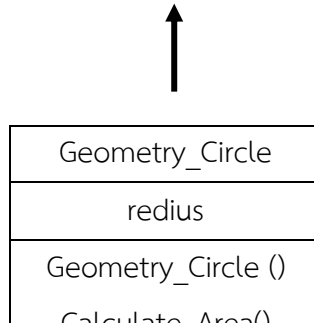
```

การพ้องรูป (Polymorphism)

คุณสมบัติการพ้องรูป (Polymorphism) เป็นคุณสมบัติที่สำคัญของการเขียนโปรแกรมเชิงวัตถุโดยที่ การเขียนโปรแกรมในลักษณะอื่นไม่มี โดยเป็นคุณสมบัติที่สอดคล้องตามข้อดีในเรื่องของการนำกลับมาใช้ใหม่ คลาสใดๆ ที่สร้างขึ้นมาจะสามารถกำหนดให้ใช้ชื่อเมธอดเดียวกันได้ แต่มีการปฏิบัติการภายในเมธอดที่ แตกต่างกัน

ตัวอย่างคลาสรูปทรงเรขาคณิต ภายในคลาสมีการคำนวณหาพื้นที่ของรูปสี่เหลี่ยม และพื้นที่วงกลมไว้ ภายในเมธอด โดยกำหนดชื่อของเมธอดให้ชื่อเดียวกัน แต่การปฏิบัติการภายในเมธอดทั้งสองจะแตกต่างกัน เนื่องจากมีสูตรการคำนวณที่ต่างกัน

| |
|----------------------|
| Geometry_Rectangle |
| width |
| length |
| Area |
| Geometry_Rectangle() |
| Calculate_Area() |
| Print_Area() |



จากรูปประกอบไปด้วย 2 คลาส คือ คลาส Geometry_Rectangle และคลาส Geometry_Circle รายละเอียดของทั้ง 2 คลาส เป็นดังนี้

| Class | Attribute | Method |
|--------------------|-------------------------|--|
| Geometry_Rectangle | width
length
Area | Geometry_Rectangle()
Calculate_Area()
Print_Area() |
| Geometry_Circle | radius | Geometry_Circle ()
Calculate_Area() |

ลองศึกษาตัวอย่างโปรแกรมต่อไปนี้

geometry.php

?>

```

class Geometry_Rectangle {
    var $width;
    var $length;
    var $area;
    function Geometry_Rectangle($width,$length) {
        $this->width = $width;
        $this->length = $length;
    }
}
  
```

```
    }
    function Calculate_Area() {
        $this->area = $this->width * $this->length;
    }
    function Print_Area() {
        echo "รูปทรงเรขาคณิตนี้มีพื้นที่ = $this->area <p>";
    }
}

class Geometry_Circle extends Geometry_Rectangle {
    var $radius;
    function Geometry_Circle($radius) {
        $this->radius = $radius;
    }
    function Calculate_Area() {
        $this->area = 3.14 * $this->radius * $this->radius;
    }
}

$rectangle = new Geometry_Rectangle(5,3);
echo "รูปสี่เหลี่ยมผืนผ้า กว้าง 5 ยาว 3 <br>";
$rectangle->Calculate_Area();
$rectangle->Print_Area();
echo "*****<br>";
$circle = new Geometry_Circle(10);
echo "รูปทรงกลมมีรัศมี 10<br>";
$circle->Calculate_Area();
$circle->Print_Area();
```

?>

ผลลัพธ์ที่ได้จากโปรแกรม

```

localhost/Test/geometry.php
รูปสี่เหลี่ยมผืนผ้า กว้าง 5 ยาว 3
รูปทรงเรขาคณิตนี้มีพื้นที่ = 15

*****
รูปทรงกลมมีรัศมี 10
รูปทรงเรขาคณิตนี้มีพื้นที่ = 314

```

การสร้างช่องรับข้อมูล

Microsoft Corporation [US] https://signup.live.com/signup.aspx?wa=wsignin1.0&ct=1404572574&rver=6.1.6206.0&sa=1&ntprob=-1&wp=MBL_SSL_

Microsoft

สร้างบัญชี

หากคุณลงชื่อเข้าใช้ที่ชี้ที่ใช้ Windows แท็บเล็ต หรือโทรศัพท์ Xbox Live, Outlook.com หรือ OneDrive อยู่แล้ว ให้ใช้อีเมลแอดเดรสนั้นเพื่อ [ลงชื่อเข้าใช้](#) หรือสร้างอีเมลแอดเดรส Outlook.com ใหม่

ชื่อ

ชื่อผู้ใช้

@ outlook.co.th

สร้างรหัสผ่าน

อย่างน้อย 8 อักขระและต้องมีใหญ่-เล็กต้องตรงกัน

ป้อนรหัสผ่านอีกครั้ง

ประเทศ/ภูมิภาค

ไทย

รหัสไปรษณีย์

วันเกิด

วันที่ เดือน ปี

ในโปรแกรม Adobe Dreamweaver CS6 ได้จัดกลุ่มเครื่องมือ Forms ที่ใช้ในการสร้างฟอร์ม โดยเราสามารถนำช่องรับข้อมูล (Input Object) เช่น Text Areas, Checkbox และอื่นๆ ไปวางบนเอกสารฟอร์ม แต่ต้องเขียนสคริปต์ PHP ทำงานควบคู่กับฟอร์มด้วย เพื่อใช้ในการประมวลผลข้อมูลที่ได้รับจากแบบสอบถาม

1. ฟอร์ม (Form)



จะเป็นตัวเริ่มต้นก่อนสร้างฟอร์มอื่นๆ ที่ทำหน้าที่เป็นช่องรับข้อมูล รวมทั้งยังทำหน้าที่กำหนดวิธีการส่งข้อมูล การระบุไฟล์ที่จะส่งไปประมวลผล

รูปแบบโค้ด HTML

```
<form id="form1" name="form1" method="post" action=""> </form>
```

โดยที่ **name=** คือชื่อ

method= วิธีส่งข้อมูล จะมี 2 ประเภท คือ Get ส่งข้อมูลโดยนำค่าข้อมูลไปรวมกับ URL และ Post ส่งข้อมูลที่ต้องการเป็นความลับ จะไม่เห็นค่าช่องรับข้อมูล

action=ไฟล์ที่รับข้อมูลจากฟอร์ม

2. ช่องรับข้อความ (Text Field) เพื่อให้ผู้ใช้ได้กรอกข้อมูล



รูปแบบโค้ด HTML

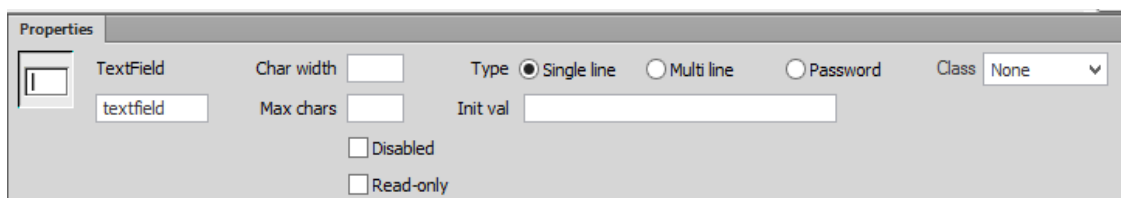
```
<label for="textfield"></label>
```

```
<input type="text" name="textfield" id="textfield" />
```

โดยที่ **type="text"** คือ ช่องรับข้อความ

name="textfield" คือ ชื่อของ Text Field

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



Char width ระบุขนาดความกว้างของกล่องรับข้อความ

Max chars ระบุจำนวนสูงสุดของตัวอักษรที่รับได้ในกล่องนี้

Init val ระบุค่าเริ่มต้นที่ต้องการให้แสดง

Type Single line กล่องข้อความแบบบรรทัดเดียว

Multi line กล่องข้อความแบบหลายบรรทัด

Password กล่องข้อความแบบรหัส ที่แสดงเป็นลักษณะจุดสีดำ

3. **ช่องรับข้อความแบบหลายบรรทัด (Text Area)** มีลักษณะเหมือนกันกับ Text Field แต่รับข้อความได้หลายบรรทัดมากกว่า หากความยาวข้อความเกินขนาดกล่องจะเพิ่ม Scroll bar เพื่อเลือกดูข้อความที่เกินมาได้

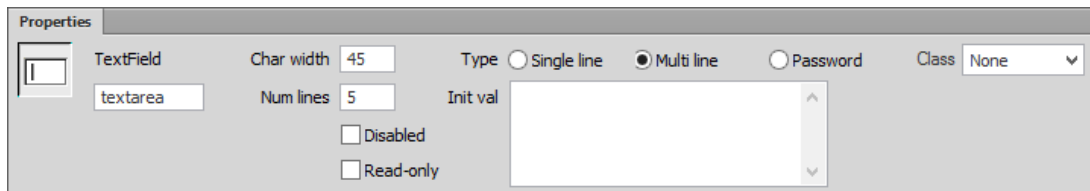


รูปแบบโค้ด HTML

```
<label for="textarea"></label>
```

```
<textarea name="textarea" id="textarea" cols="45" rows="5"></textarea>
```

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



Char width ระบุขนาดความกว้างของกล่องรับข้อความแบบหลายบรรทัด

Num Lines ระบุจำนวนบรรทัดของข้อความโดยไม่ปรากฏ scroll bar

4. **กล่องตัวเลือก (Check box)** มีลักษณะเป็นกล่องสี่เหลี่ยมเล็กๆ ใช้สำหรับคลิกรายการที่ต้องการเลือก เราสามารถสร้างกล่องได้หลายกล่อง ในแต่ละรายการเลือก รวมทั้งสามารถเลือกได้หลายรายการพร้อมกัน โดยจะมีเครื่องหมายถูกที่กล่องที่เลือก



รูปแบบโค้ด HTML

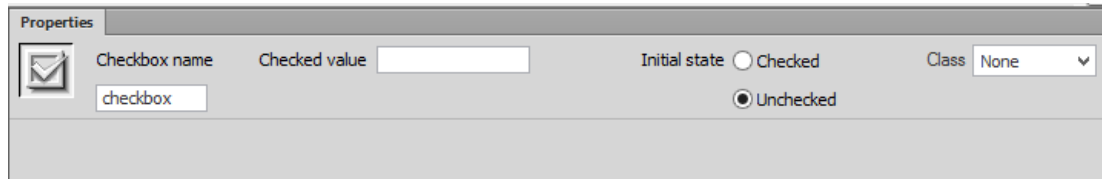
```
<input type="checkbox" name="checkbox" id="checkbox" />
```

```
<label for="checkbox"></label>
```

โดยที่ `type="checkbox"` คือ ระบุว่า input object นี้คือ ตัวเลือก

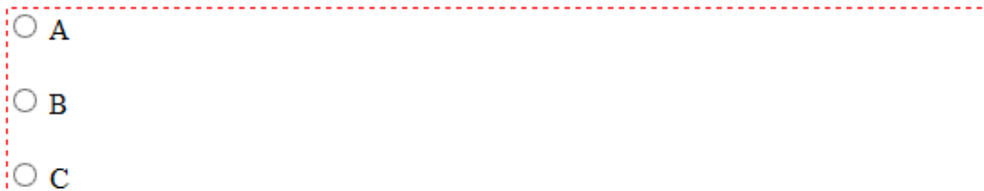
`name="checkbox"` คือ ชื่อของ checkbox

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



Checked value	ระบุค่าของกล่องตัวเลือกแต่ละตัว
Initial state	Checked บังคับเลือก
	Unchecked ไม่บังคับว่ามีการเลือก

5. ปุ่มตัวเลือก (Radio Button) มีลักษณะเป็นกล่องกลมเล็กๆ ใช้สำหรับคลิกรายการที่ต้องการเลือก เราสามารถสร้างกล่องได้หลายกล่อง ในแต่ละรายการเลือก รวมทั้งสามารถเลือกได้หลายรายการพร้อมกัน โดยจะมีเครื่องหมายถูกที่กล่องที่เลือก



รูปแบบโค้ด HTML

```
<input type="radio" name="radio" id="radio" value="radio" />
```

```
<label for="radio"></label>
```

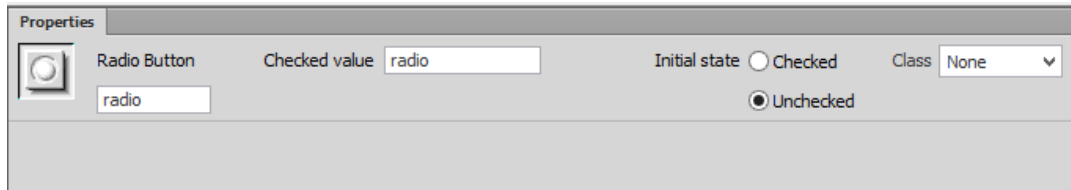
```
<label for="checkbox"></label>
```

โดยที่ `type="radio"` คือ ระบุว่าเป็นปุ่มตัวเลือก

`name="radio"` คือ ชื่อของปุ่มตัวเลือก

`value="radio"` คือ ระบุค่าปุ่มตัวเลือกถ้ามีการเลือกปุ่มนี้

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



Checked value	ระบุค่าของกล่องตัวเลือกแต่ละตัว
Initial state	Checked บังคับเลือก
	Unchecked ไม่บังคับว่ามีการเลือก

5. กล่องรายการ (List Box) มีลักษณะเป็นกล่องสี่เหลี่ยม บรรจุรายการตัวเลือกต่างๆ ถ้าจะเลือกรายการก็ทำการคลิกให้กล่องนี้เลื่อนข้อมูลออกมาจากกล่อง

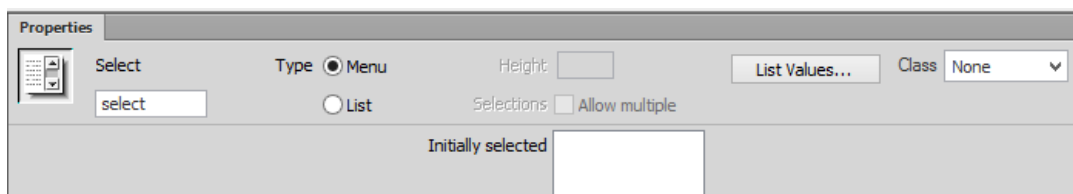


รูปแบบโค้ด HTML

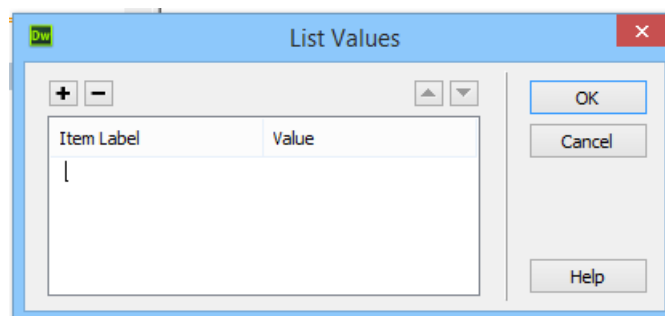
```
<label for="select"></label>
```

```
<select name="select" id="select"></select>
```

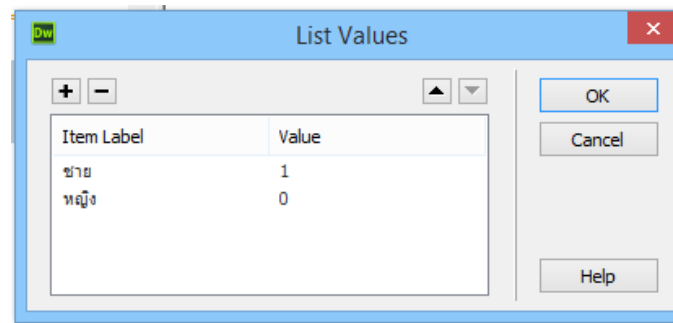
สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



เราสามารถทำการเพิ่มรายการลงกล่องรายการได้ โดยการคลิกที่ปุ่ม List Values... ซึ่งเมื่อเราคลิกแล้ว ก็จะปรากฏหน้าต่างเพื่อให้เพิ่มรายการต่างๆ ดังนี้



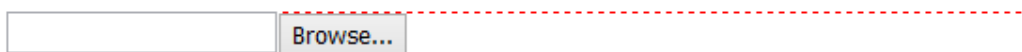
ให้เราทำการเพิ่มตัวเลือกที่เราต้องการเข้าไปใน Item Label ในส่วนของ Value ให้กำหนดค่าของตัวเลือกว่ามีค่าเท่ากับเท่าไร เพื่อนำไปใช้ในการประมวลผลต่อไป ดังตัวอย่าง



ซึ่งจะมีลักษณะรูปแบบโค้ด HTML ของกล่องเป็นดังนี้

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

6. **กล่องรับชื่อไฟล์ (File Field)** มีลักษณะเป็นกล่องสี่เหลี่ยม ใช้สำหรับเลือกไฟล์ เพื่อใช้ในการอัปโหลดไฟล์ไปยังเซิร์ฟเวอร์ ลักษณะคล้ายกล่องรับข้อความ



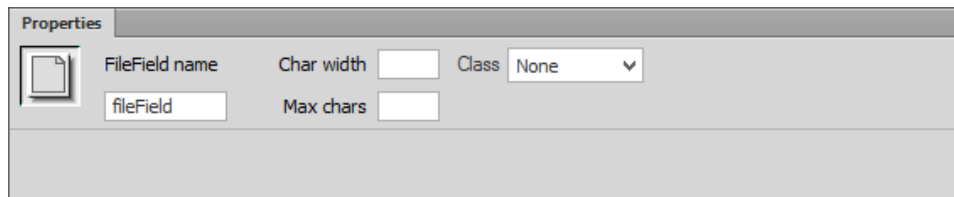
รูปแบบโค้ด HTML

```
label for="fileField"></label>
<input type="file" name="fileField" id="fileField" />
```

โดยที่ **type="file"** คือ ระบุว่า เป็นช่องรับชื่อไฟล์

name="fileField" คือ ชื่อของช่องรับไฟล์

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



Char width ระบุขนาดความกว้างของกล่องรับชื่อไฟล์

Max chars ระบุจำนวนตัวอักษรที่ช่องรับไฟล์สามารถรับได้

7. ปุ่มคำสั่ง (Button) การรับข้อมูลที่ใช้ได้ทำการกรอกหรือเลือก และนำไปประมวลผลต่อไป จะต้องใช้ปุ่มคำสั่งทำการส่งข้อมูลนำเข้าต่างๆ เหล่านี้ (Submit)

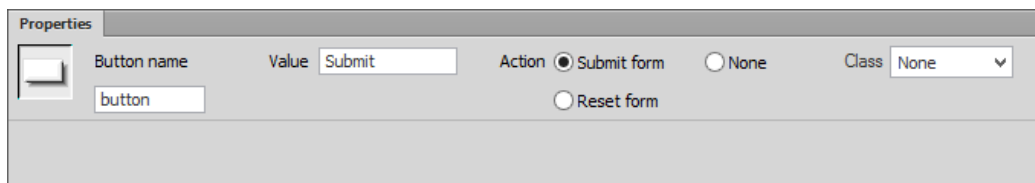


รูปแบบโค้ด HTML

Submit `<input type="submit" name="button" id="button" value="Submit" />`

Reset `<input type="reset" name="button2" id="button2" value="Reset" />`

สังเกตที่บริเวณ Properties ของ Dreamweaver เราสามารถปรับแต่งลักษณะของช่องรับได้ดังนี้



Value ข้อความที่ต้องการให้ปรากฏบนปุ่มคำสั่ง

Action Submit ส่งข้อมูล Reset ยกเลิกหรือล้างข้อมูล None ปุ่มธรรมดา



การเขียน ER – diagram
และการสร้างฟอร์มรับข้อมูล

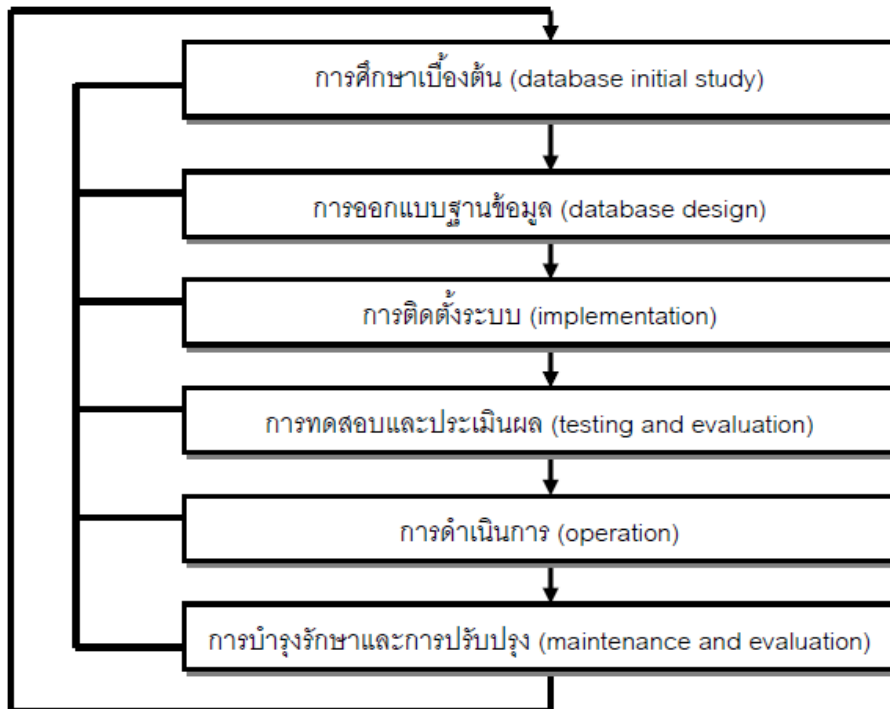
การสร้าง ER Diagram

การออกแบบฐานข้อมูล

ฐานข้อมูลที่มีประสิทธิภาพและตรงกับความต้องการของผู้ใช้นั้น ต้องอาศัยการพัฒนาฐานข้อมูลที่มีการวางแผนอย่างเป็นระบบและมีขั้นตอนที่ถูกต้อง โดยในบทนี้จะศึกษาถึงขั้นตอนการพัฒนาฐานข้อมูล ซึ่งจะเน้นขั้นตอนการออกแบบฐานข้อมูลเป็นหลัก ตั้งแต่การพัฒนาแบบจำลองอี-อาร์ และการทำนอร์มัลไลเซชัน เพื่อให้ได้ฐานข้อมูลที่สมบูรณ์ที่สุด

ขั้นตอนการพัฒนาฐานข้อมูล

วัฏจักรฐานข้อมูล (The Database Life Cycle: DBLC) เป็นขั้นตอนในการพัฒนาหรือจัดทำระบบฐานข้อมูล ซึ่งประกอบด้วย 6 ขั้นตอน



1. การศึกษาเบื้องต้น

การศึกษาเบื้องต้นมีวัตถุประสงค์เพื่อวิเคราะห์สถานการณ์ขององค์กร กำหนดปัญหาและข้อจำกัด กำหนดวัตถุประสงค์และขอบเขตของระบบ ดังรายละเอียดต่อไปนี้

1.1 การวิเคราะห์สถานการณ์ขององค์กร เพื่อศึกษาสภาพแวดล้อมในการทำงานขององค์กร ความต้องการในการปฏิบัติงาน โดยควรรู้ว่าโครงสร้างขององค์กรเป็นอย่างไร ใครเป็นผู้ควบคุมอะไร และใครทารายงานให้ใคร เป็นต้น

1.2 กำหนดปัญหาและข้อจำกัด โดยการศึกษาว่า ระบบที่มีอยู่มีการทำงานอย่างไร ข้อมูลที่ป้อนเข้าสู่ระบบ มีอะไรบ้าง และระบบสร้างรายงานอะไร มีการใช้รายงานเหล่านั้นอย่างไรและใครเป็นผู้ใช้ เพื่อให้ทราบถึงปัญหาและข้อจำกัดในการป้อนข้อมูล หรือการค้นหาข้อมูลเพื่อการทำรายงาน

1.3 กำหนดวัตถุประสงค์และขอบเขตของระบบ ในการกำหนดวัตถุประสงค์ของระบบ ฐานข้อมูลควรสอดคล้องกับความต้องการของผู้ใช้ จากคำถามเหล่านี้

- วัตถุประสงค์แรกเริ่มของระบบที่นำเสนอคืออะไร
- ระบบนี้ต้องเชื่อมต่อกับระบบอื่นๆ ที่มีอยู่ในองค์กรหรือไม่
- ระบบนี้จะมีการใช้ข้อมูลร่วมกับระบบหรือผู้ใช้อื่นหรือไม่

เมื่อทราบวัตถุประสงค์แล้วก็ทำการกำหนดขอบเขตของระบบโดยการออกแบบ ตามความต้องการในการปฏิบัติงาน เพื่อใช้ในการออกแบบฐานข้อมูลต่อไป

2. การออกแบบฐานข้อมูล

เมื่อผู้ออกแบบฐานข้อมูลมีความเข้าใจลักษณะขององค์กร ปัญหาและข้อจำกัด รวมทั้งวัตถุประสงค์และขอบเขตของระบบแล้ว ก็ทำการออกแบบฐานข้อมูลดังต่อไปนี้

2.1 การออกแบบเชิงแนวคิด โดยการพัฒนาแบบจำลองอี-อาร์ (E-R Model) ที่ใช้อธิบายถึงความสัมพันธ์ระหว่างสิ่งที่เราสนใจจะจัดเก็บ ที่เรียกว่า เอนทิตี (entity) และรายละเอียดหรือคุณสมบัติ (attribute) ของสิ่งที่จัดเก็บ แล้วทำการแปลงแบบจำลองอี-อาร์ เป็นโครงสร้างตารางฐานข้อมูล จากนั้นก็ทำการนอร์มัลไลเซชัน (normalization) เพื่อให้ได้โครงสร้างของตารางที่ดี สามารถควบคุมความซ้ำซ้อนของข้อมูล หลีกเลี่ยงความผิดปกติของข้อมูล

2.2 การเลือกโปรแกรมจัดการฐานข้อมูล ในการตัดสินใจเลือกชื่อโปรแกรมจัดการฐานข้อมูลขององค์กรใด ควรพิจารณาถึงสิ่งต่อไปนี้

2.2.1 ค่าใช้จ่ายต่างๆ เช่น ราคาการซ่อมบำรุง การปฏิบัติงาน ลิขสิทธิ์ การติดตั้ง การฝึกอบรม และค่าใช้จ่ายในการเปลี่ยนไปใช้ระบบใหม่

2.2.2 คุณลักษณะและเครื่องมือของระบบจัดการฐานข้อมูล โปรแกรมฐานข้อมูลบางตัวจะรวมเอาเครื่องมือต่างๆ ที่ให้ความสะดวกในงานการพัฒนาโปรแกรมประยุกต์ ตัวอย่างเช่น การออกแบบหน้าจอ การสร้างรายงาน การสร้างโปรแกรมประยุกต์ และพจนานุกรมข้อมูล เป็นต้น ทำให้สะดวกในการบริหารฐานข้อมูล ใช้ง่าย มีความสามารถในการรักษาความปลอดภัย และการควบคุมการใช้งานพร้อมกัน เป็นต้น

2.2.3 ความสามารถในการใช้ข้าม platforms ข้ามระบบและภาษา

2.2.4 ความต้องการด้านฮาร์ดแวร์ หน่วยความจำ และเนื้อที่ที่ใช้ในการจัดเก็บ

2.3 การออกแบบทางตรรกะ จะเกี่ยวข้องกับการตัดสินใจใช้รูปแบบเฉพาะของฐานข้อมูล (แบบลำดับชั้น แบบเครือข่าย และแบบเชิงสัมพันธ์ เป็นต้น) การกำหนดรูปแบบของฐานข้อมูล ซึ่งการออกแบบเชิงตรรกะจะเป็นการแปลงการออกแบบระดับเชิงแนวคิด ให้เป็นแบบจำลองของฐานข้อมูล ในระดับภายใน (internal model) ตามระบบการจัดการฐานข้อมูล (DBMS) เช่น MS-Access และ Oracle โดยการสร้างตาราง ฟอรั่ม คิวรี และรายงาน เป็นต้น

2.4 การออกแบบทางกายภาพ การออกแบบทางกายภาพ คือ กระบวนการในการเลือกหน่วยจัดเก็บข้อมูล และลักษณะการเข้าถึงข้อมูลของฐานข้อมูล การสร้างดัชนี (index) การจัดทำคลัสเตอร์ (clustering) ซึ่งเป็นการจัดเก็บข้อมูลที่มีการใช้งานบ่อยๆ ไว้ในหน่วยเก็บข้อมูลเดียวกัน หรือการใช้เทคนิคแฮชซิง (hashing technique) ในการจัดตำแหน่งที่อยู่ของข้อมูลภายในหน่วยเก็บข้อมูล เป็นต้น

3. การติดตั้งระบบ

ขึ้นอยู่กับระบบจัดการฐานข้อมูลที่ใช้ โดยเริ่มต้นจากการสร้างฐานข้อมูล กำหนดผู้จัดการฐานข้อมูล กำหนดพื้นที่ๆ ที่ต้องการใช้ และการสร้างตารางต่างๆ ในระบบ

4. การทดสอบและประเมินผล

เพื่อการตรวจสอบดูว่าระบบที่พัฒนามาสามารถทำงานได้ตามที่ต้องการหรือไม่ ซึ่งควรมีการเตรียมข้อมูลทดสอบไว้ล่วงหน้า

5. การดำเนินการ

เมื่อฐานข้อมูลผ่านขั้นตอนการทดสอบและประเมินผล ต่อไปก็เป็นขั้นตอน การดำเนินการ หรือ การติดตั้งระบบ ซึ่งต้องเป็นระบบที่สมบูรณ์พร้อมให้ผู้ใช้ได้ใช้งานนั่นเอง ซึ่งอาจรวมไปถึงการฝึกอบรมให้แก่ผู้ใช้ ที่เป็นพนักงานที่ต้องใช้งานจริงด้วย

6. การบำรุงรักษาและการปรับปรุง

หลังจากระบบได้เริ่มดำเนินการ ผู้จัดการฐานข้อมูลจะต้องเตรียมการบำรุงรักษาฐานข้อมูลโดยการสำรองข้อมูลไว้ เพื่อสะดวกในการกู้คืนข้อมูล เมื่อระบบมีปัญหา และหากมีการใช้งานไปนานๆ อาจต้องทำการปรับปรุงแก้ไขโปรแกรมให้เหมาะสมกับเหตุการณ์ และ ความต้องการของผู้ใช้ที่เปลี่ยนแปลงไป

เมื่อเราทราบขั้นตอนการพัฒนาฐานข้อมูลทั้งหมดแล้ว ในบทนี้จะเน้นถึงรายละเอียดของขั้นตอนการออกแบบฐานข้อมูลเชิงแนวคิด ตั้งแต่การพัฒนาแบบจำลองอี-อาร์ (E-R Model) และการทำออร์มัลไลเซชัน (normalization) ดังรายละเอียดที่จะกล่าวต่อไป

แบบจำลองอี-อาร์

แบบจำลองอี-อาร์ (Entity-Relationship Model: E-R Model) เป็นแบบจำลองข้อมูล ที่ประยุกต์มาจากแนวคิดเรื่อง Semantic Model และมีการพัฒนามาเป็น E-R Model โดย Peter Pin Shan Chen จาก Massachusetts Institute of Technology ในปี ค.ศ. 1976 และได้รับความนิยมาจนถึงปัจจุบัน

1. ความหมายและความสำคัญของแบบจำลองอี-อาร์

แบบจำลองอี-อาร์ เป็นเครื่องมือที่ใช้ในการออกแบบฐานข้อมูล ที่แสดงความสัมพันธ์ระหว่าง เอนทิตีหรือสิ่งที่เราต้องการจะจัดเก็บไว้ในฐานข้อมูล โดยนำเสนอในรูปของของแผนภาพ ที่เรียกว่า อี-อาร์ ไดอะแกรม (E-R Diagram) ด้วยการใช้อยู่ลักษณะต่างๆ

แบบจำลองอี-อาร์ มีความสำคัญในการเป็นสื่อกลางเพื่อสื่อสารกับบุคลากรต่างๆ ที่เกี่ยวข้องกับระบบฐานข้อมูล ไม่ว่าจะเป็นในระดับผู้บริหาร นักเขียนโปรแกรม และผู้ใช้ในระดับปฏิบัติการ เป็นต้น ทำให้เข้าใจระบบได้อย่างถูกต้องตรงกัน เนื่องจากการแสดงภาพรวม ของระบบในลักษณะของรูปภาพหรือแผนภาพ ทำให้เข้าใจง่าย ดังนั้นระบบที่ออกแบบมาจึงมีความถูกต้องและเป็นไปตามวัตถุประสงค์ขององค์กร

2. องค์ประกอบของแบบจำลองอี-อาร์

แบบจำลองอี-อาร์ ประกอบด้วย เอนทิตี แอตทริบิวต์ คีย์ และความสัมพันธ์ ดังรายละเอียดต่อไปนี้

2.1 เอนทิตี เอนทิตี (entity) คือ สิ่งต่างๆ ที่ผู้ใช้งานฐานข้อมูลต้องการจะจัดเก็บ ซึ่งมีลักษณะเป็นคำนาม ทั้งรูปธรรมและนามธรรม เช่น บุคคล สถานที่ วัตถุสิ่งของ และเหตุการณ์ต่างๆ เป็นต้น ตัวอย่างของเอนทิตีใน “ระบบการลงทะเบียนเรียนของนักศึกษา” ประกอบด้วย รายวิชา นักศึกษา การลงทะเบียน ผลการเรียน วิชาเรียน สาขาวิชา คณะ และหลักสูตร เป็นต้น

เอนทิตีที่รวบรวมได้จากระบบสามารถแยกแยะและจัดเป็นหมวดหมู่ได้ตามชนิดของเอนทิตี ได้ดังต่อไปนี้

- หมวดบุคคล ได้แก่ เอนทิตี - นักศึกษา พนักงาน ประชาชน ผู้ป่วย และลูกค้า เป็นต้น
- หมวดสถานที่ ได้แก่ เอนทิตี - รัฐ ประเทศ จังหวัด ภาค สาขา และวิทยาเขต เป็นต้น
- หมวดวัตถุ ได้แก่ เอนทิตี - อาคาร เครื่องจักร ผลผลิต หนังสือ วัตถุดิบ และรถยนต์ข
- หมวดเหตุการณ์ ได้แก่ เอนทิตี - การขาย การลงทะเบียน การเดินทาง การสั่งซื้อของ การออกใบเสร็จรับเงิน และการให้รางวัล เป็นต้น

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้า แทนหนึ่งเอนทิตี โดยใช้ชื่อของเอนทิตีนั้นๆ กำกับอยู่ภายใน เช่น



2.2 แอตทริบิวต์ แอตทริบิวต์ (attribute) คือ คุณสมบัติต่างๆ ของเอนทิตีที่เราต้องการจัดเก็บในฐานข้อมูล ตัวอย่างเช่น

- เอนทิตีบัตรประชาชน ประกอบด้วยแอตทริบิวต์ หรือสิ่งที่บ่งบอกคุณสมบัติของประชาชนแต่ละคน ได้แก่ หมายเลขบัตรประชาชน ชื่อ นามสกุล วันเดือนปีเกิด ภูมิลำเนา วันที่ออกบัตร วันที่บัตรหมดอายุ ส่วนสูง น้ำหนัก และกรุ๊ปเลือด เป็นต้น

- เอนทิตีพนักงาน ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสพนักงาน ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ สถานภาพสมรส และเงินเดือน เป็นต้น

- เอนทิตีสินค้า ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสสินค้า ชื่อสินค้า ราคา และจำนวน เป็นต้น

- เอนทิตีนักศึกษา ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสนักศึกษา ชื่อ นามสกุล เพศ วันเดือนปีเกิด ที่อยู่ และเบอร์โทรศัพท์ เป็นต้น

- เอนทิตีวิชา ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสวิชา ชื่อวิชา และจำนวนหน่วยกิต เป็นต้น

ค่าของข้อมูลในแต่ละแอตทริบิวต์ประกอบกัน เรียกว่า ทูเพิล (tuple) ซึ่งเป็นแถวของข้อมูลในตาราง โดยแต่ละแถวหรือแต่ละทูเพิลจะประกอบด้วยหลายแอตทริบิวต์หรือ หลายคอลัมน์ของข้อมูล จำนวนแถวของข้อมูลในตารางเรียกว่า Cardinality และจำนวน แอตทริบิวต์ทั้งหมดในตารางเรียกว่า Degree

แอตทริบิวต์ (attribute)					
	รหัสนักศึกษา	ชื่อ	นามสกุล	หลักสูตร	คณะ
ทูเพิล (tuple)	5600111	สาธิต	กิตติพงศ์	คอมพิวเตอร์	วิทยาศาสตร์
	5600222	ชานนท์	สกุลวงศ์	เทคโนโลยีสารสนเทศ	มนุษยศาสตร์
	5600333	ธัญญา	โชติช่วง	เทคโนโลยีสารสนเทศ	มนุษยศาสตร์
	5600444	โสภณ	ปัญญาเลิศ	บริหารธุรกิจ	วิทยาการจัดการ

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปวงรี แทนหนึ่งแอตทริบิวต์ โดยใช้ชื่อของ แอตทริบิวต์นั้นๆ กำกับอยู่ภายใน เช่น

ชื่อ

แทนแอตทริบิวต์ของชื่อ

2.3 คีย์ คีย์ (key) คือ แอตทริบิวต์ที่สามารถใช้บ่งบอกความแตกต่างของแต่ละทูเพิลได้ อาจเป็นแอตทริบิวต์เดี่ยวๆ หรือ กลุ่มของแอตทริบิวต์ก็ได้ ประเภทของคีย์ประกอบด้วย

2.3.1 ซุปเปอร์คีย์ (super key) คือ แอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ ที่สามารถบ่งบอกความแตกต่างของแต่ละทูเพิลได้

รหัสนักศึกษา	ชื่อ	นามสกุล	เลขที่บัตรประชาชน
56001	สามารถ	ประเสริฐกุล	3120100475991
56002	วิชา	ปัญญาเลิศ	3120100475992
56003	น้ำทิพย์	วิเศษศิริ	3120100475993

2.3.2 คีย์คู่แข่ง (candidate key) คือ ซุปเปอร์คีย์ที่น้อยที่สุด ที่สามารถบ่งบอกความแตกต่างของแต่ละทูเพิลได้

2.3.3 คีย์หลัก (primary key) คือ คีย์คู่แข่งที่ถูกเลือก เพื่อใช้บ่งบอกความแตกต่างของแต่ละทูเพิล

คุณสมบัติของคีย์หลัก

- 1) คีย์หลักซ้ำกันไม่ได้
- 2) คีย์หลักอาจเป็นแค่หนึ่งแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ก็ได้ อย่างในตาราง มีแอตทริบิวต์เดียวที่เป็นคีย์หลัก ซึ่งอาจจะเป็น “รหัสนักศึกษา” หรือ “เลขที่บัตรประชาชน” ก็ได้ แต่ข้อมูลบางตารางอาจต้องอาศัยแอตทริบิวต์ตั้งแต่ 2 ตัวขึ้นไปมาประกอบกันเป็นคีย์หลัก เพื่อให้เกิดความแตกต่างระหว่างทูเพิล

รหัสนักศึกษา	ชื่อ	รหัสนวิชา	ชื่อวิชา	ปีการศึกษา
5620249001	ปรีชา	111	คอมพิวเตอร์	1/56
5620249001	ปรีชา	222	ภาษาไทย	1/56
5620249002	เกรียงไกร	111	คอมพิวเตอร์	1/56
5620249003	ฉัตรชัย	333	ภาษาอังกฤษ	1/56
5620249003	ฉัตรชัย	222	ภาษาไทย	1/56
5620249003	ฉัตรชัย	444	สิ่งแวดล้อม	1/56

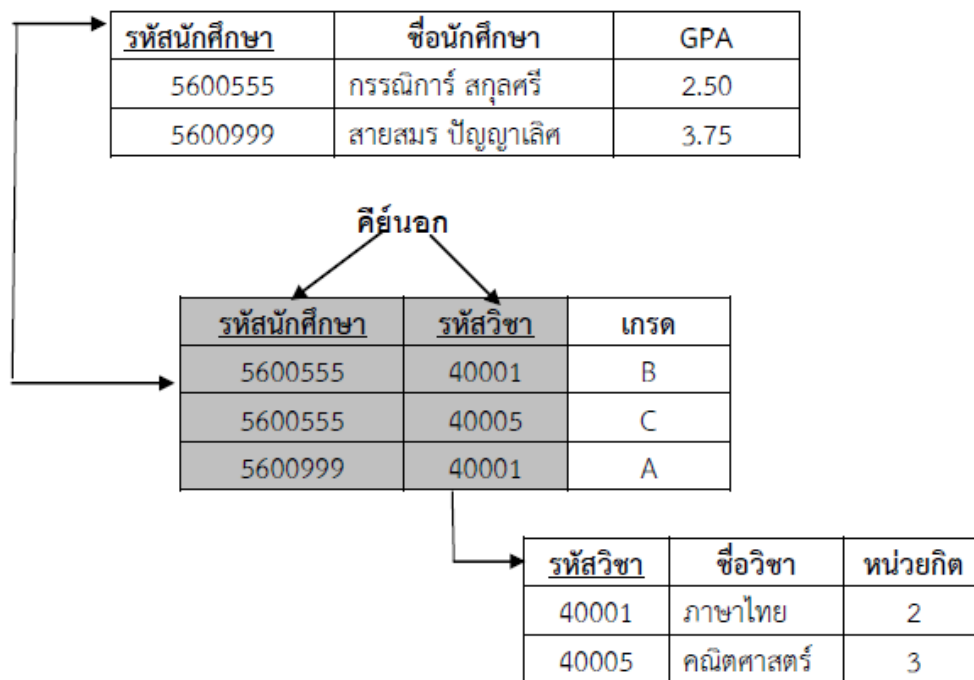
จากตาราง ไม่สามารถให้แอตทริบิวต์รหัสนักศึกษา เป็นคีย์หลักเพียงแอตทริบิวต์เดียวได้ เพราะจะเห็นว่า รหัสนักศึกษา 5620249001 ของทูเพิลหรือแถวที่ 1 จะไปซ้ำกับแถวที่ 2 แต่ถ้าให้แอตทริบิวต์ “รหัสนักศึกษา” และ “รหัสนวิชา” เป็นคีย์หลัก แล้วพิจารณาข้อมูลของ 2 แอตทริบิวต์นี้ จะเห็นว่าข้อมูลไม่ซ้ำ

กันแล้ว ดังนั้นตาราง จึงมีคีย์หลักซึ่งประกอบด้วยแอตทริบิวต์ 2 ตัวประกอบกัน คือ “รหัสนักศึกษา” และ “รหัสวิชา”

3) คีย์หลักจะเป็นค่าว่าง (null) ไม่ได้ เพราะฉะนั้นในการกรอกข้อมูลต่างๆ ลงในตาราง แอตทริบิวต์ใดที่เรากำหนดให้เป็นคีย์หลักต้องกรอกข้อมูลให้ครบ คือ จะไม่มีค่าไม่ได้ แต่แอตทริบิวต์อื่นอาจจะปล่อยเว้นว่างไว้ก็ได้ถ้าไม่ทราบค่า

2.3.4 คีย์นอก (foreign key) คือ แอตทริบิวต์ที่ใช้ในการเชื่อมต่อกับเอนทิตี อื่นๆ เพื่อแสดงความสัมพันธ์ คุณสมบัติของคีย์นอก คือ

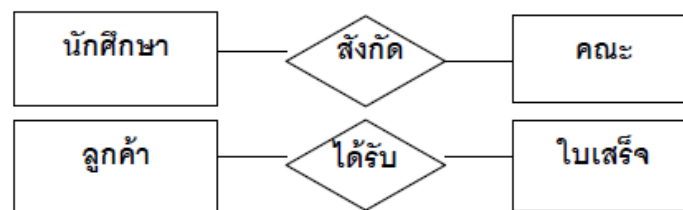
- คีย์นอกสามารถมีค่าซ้ำกันได้
- คีย์นอกสามารถเป็นค่าว่างได้
- คีย์นอกที่ไม่เป็นค่าว่างจะเป็นค่าที่ชี้ไปยังคีย์หลักของเอนทิตีที่สัมพันธ์กัน



2.3.4 คีย์รอง (secondary key) คือ แอตทริบิวต์ที่ไม่เป็น key หลัก แต่สามารถใช้ในการค้นหาข้อมูลนั้นๆ ได้ โดยคีย์รองจะมีค่าซ้ำกันได้ ตัวอย่างเช่น ในตาราง มีรหัสนักศึกษาเป็นคีย์หลัก แต่หากต้องการค้นหาข้อมูลจากชื่อนักศึกษา แอตทริบิวต์ชื่อก็คือ คีย์รอง หรือถ้าต้องการค้นหาข้อมูลจากนามสกุล นักศึกษา แอตทริบิวต์นามสกุลก็จะเป็น คีย์รอง เป็นต้น

รหัสนักศึกษา	ชื่อ	นามสกุล	เบอร์โทรศัพท์
56111	สามารถ	ประเสริฐกุล	0-1111-1111
56112	วิชา	ปัญญาเลิศ	0-2222-2222
56113	น้ำทิพย์	วิเศษศิริ	0-5555-5555
56114	สมจิตร	สมสกุลวงศ์	0-6666-6666
56115	วิชา	รักศักดิ์ศรี	0-9999-9999

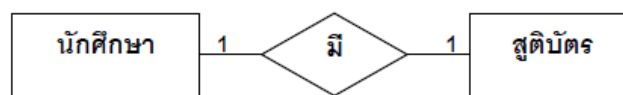
2.4 ความสัมพันธ์ ความสัมพันธ์ (relationship) เป็นการอธิบายความสัมพันธ์ระหว่างเอนทิตีที่มีความความสัมพันธ์กัน ว่ามีความสัมพันธ์กันอย่างไร โดยในอี-อาร์ไดอะแกรมใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัด ที่มีชื่อของความสัมพันธ์นั้นกำกับอยู่ภายใน และเชื่อมต่อกับเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์ด้วยเส้นตรง ดังตัวอย่างด้านล่าง



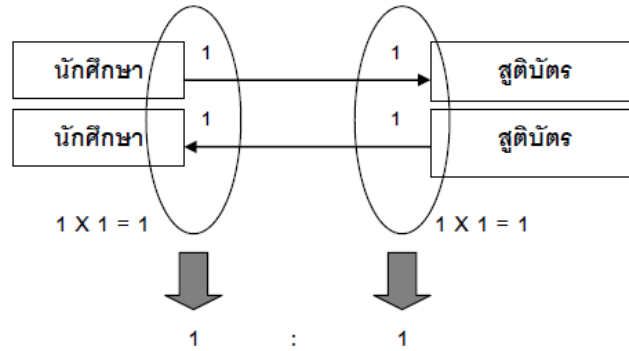
ความสัมพันธ์ระหว่างเอนทิตีแบ่งเป็น 3 ประเภท คือ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง ความสัมพันธ์แบบหนึ่งต่อกลุ่ม และความสัมพันธ์แบบกลุ่มต่อกลุ่ม ดังรายละเอียดต่อไปนี้

2.4.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one relationship หรือ 1:1)

หมายถึง ข้อมูลในเอนทิตีหนึ่ง มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตีเพียงข้อมูลเดียว ตัวอย่างเช่น นักศึกษาแต่ละคนจะมีสูติบัตรได้เพียงใบเดียวเท่านั้น และสูติบัตรหนึ่งใบก็เป็นของนักศึกษาได้เพียงคนเดียวเท่านั้นเช่นกัน

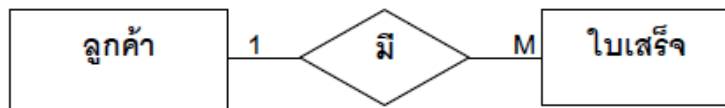


ในการพิจารณาความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง ต้องมองสองทิศ คือ มองจากซ้ายไปขวา และก็ต้องมองจากขวาไปซ้าย แล้วจึงนำความสัมพันธ์ทั้งสองทิศ มาพิจารณารวมกัน ดังภาพ

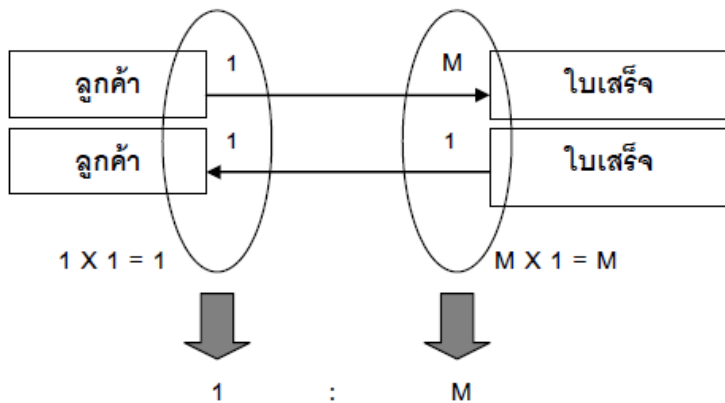


2.4.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many relationship หรือ 1:M)

หมายถึง ข้อมูลในเอนทิตีหนึ่ง มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตีมากกว่าหนึ่งข้อมูล ตัวอย่างเช่น ลูกค้านึงคนมีใบเสร็จได้หลายใบ เนื่องจากลูกค้าหนึ่งคนอาจมาซื้อสินค้าหลายครั้ง แต่ใบเสร็จหนึ่งใบต้องเป็นของลูกค้าเพียงคนเดียวเท่านั้น

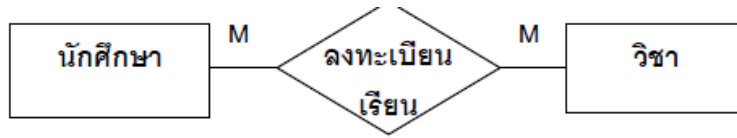


ในการพิจารณาความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่ม ต้องมองสองทิศ คือ มองจากซ้ายไปขวา และก็ต้องมองจากขวาไปซ้าย แล้วจึงนำความสัมพันธ์ทั้งสองทิศ มาพิจารณารวมกัน ดังภาพ

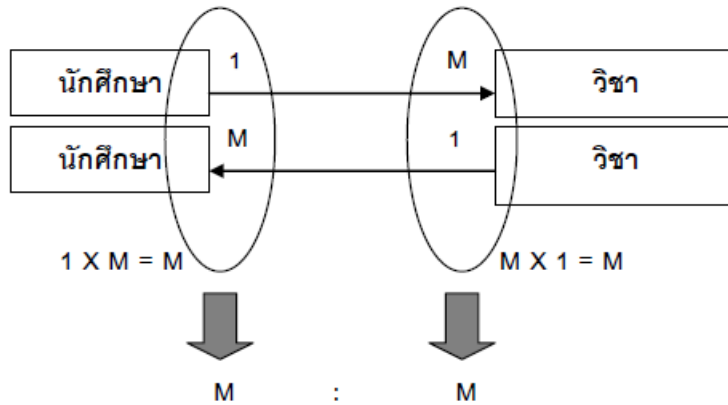


2.4.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many relationship หรือ M:M)

หมายถึง ข้อมูลมากกว่าหนึ่งข้อมูลในเอนทิตีหนึ่ง มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตีมากกว่าหนึ่งข้อมูล ตัวอย่างเช่น นักศึกษาหนึ่งคนสามารถลงทะเบียนเรียนได้หลายวิชา และวิชาแต่ละวิชามีนักศึกษาลงทะเบียนเรียนได้หลายคน



ในการพิจารณาความสัมพันธ์ระหว่างเอนทิตีแบบกลุ่มต่อกลุ่ม ต้องมองสองทิศ คือ มองจากซ้ายไปขวา และก็ต้องมองจากขวาไปซ้าย แล้วจึงนำความสัมพันธ์ทั้งสองทิศ มาพิจารณารวมกัน ดังภาพ



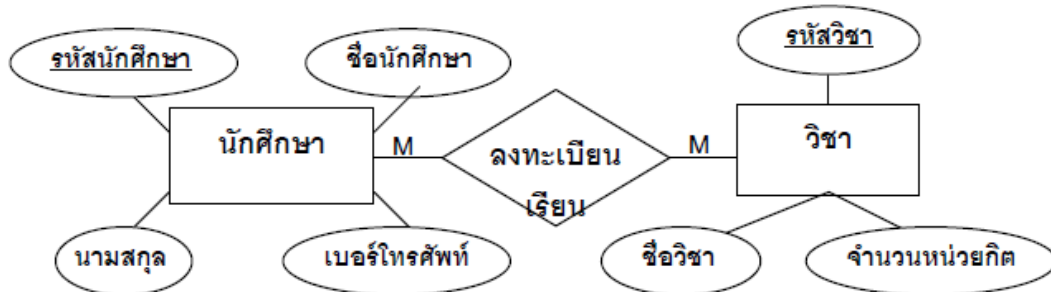
3. สัญลักษณ์ในแบบจำลองอี-อาร์

สัญลักษณ์	ความหมาย
	เอนทิตี
	ความสัมพันธ์
	แอตทริบิวต์
	คีย์หลัก

ตัวอย่าง

นักศึกษา และ วิชา เป็นเอนทิตีที่เราสนใจจะจัดเก็บ ซึ่งเอนทิตีนักศึกษาก็ประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสนักศึกษา ชื่อนักศึกษา นามสกุล และเบอร์โทรศัพท์ เป็นต้น โดยมีรหัสนักศึกษาเป็นคีย์หลัก ส่วนเอนทิตีวิชาจะประกอบด้วยแอตทริบิวต์ ได้แก่ รหัสวิชา ชื่อวิชา และจำนวนหน่วยกิต เป็นต้น โดยมีรหัสวิชาเป็นคีย์หลัก ซึ่งความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีวิชา เป็นแบบกลุ่มต่อกลุ่ม คือ นักศึกษา

หนึ่งคนสามารถลงทะเบียนเรียนได้หลายวิชา และวิชาแต่ละวิชามีนักศึกษาลงทะเบียนเรียนได้หลายคน ดังนั้นเราสามารถนำเสนอในรูปของของแผนภาพ ที่เรียกว่า อี-อาร์ไดอะแกรม (E-R Diagram) ด้วยการใช้สัญลักษณ์ต่างๆ ดังนี้



4. การแปลงแบบจำลองอี-อาร์เป็นโครงสร้างตารางฐานข้อมูล

ขั้นตอนในการแปลงแบบจำลองอี-อาร์เป็นโครงสร้างของตารางในฐานข้อมูล มีขั้นตอนดังต่อไปนี้

4.1 แปลงเอนทิตีที่ปกติในแบบจำลองอี-อาร์เป็น 1 ตาราง ซึ่งประกอบด้วยแอตทริบิวต์ของเอนทิตีนั้นๆ โดยชื่อของตารางก็คือชื่อของเอนทิตี และแอตทริบิวต์ของเอนทิตี ก็คือ แอตทริบิวต์ของตาราง สำหรับแอตทริบิวต์ที่เป็นคีย์หลักของตาราง ให้ขีดเส้นใต้ที่แอตทริบิวต์นั้น เช่นเดียวกับในแบบจำลองอี-อาร์ ซึ่งจากภาพ นำมาแปลงเป็นตารางได้ 2 ตาราง คือ

นักศึกษา			
รหัสนักศึกษา	ชื่อ	นามสกุล	เบอร์โทรศัพท์
56111	สามารถ	ประเสริฐกุล	0-1111-1111
56112	วิชา	ปัญญาเลิศ	0-2222-2222

วิชา		
รหัสวิชา	ชื่อวิชา	หน่วยกิต
40001	ภาษาไทย	2
40005	คณิตศาสตร์	3

4.2 แปลงความสัมพันธ์เป็นตาราง

4.2.1 แปลงความสัมพันธ์แบบ 1 : M นั้นไม่ต้องสร้างตารางใหม่ แต่ให้นำแอตทริบิวต์ ที่เป็นคีย์หลักของเอนทิตีที่อยู่ด้านความสัมพันธ์ที่เป็น 1 ไปเพิ่มเป็นแอตทริบิวต์ของตารางด้านที่มีความสัมพันธ์เป็น M

4.2.2 แปลงความสัมพันธ์แบบ $M : M$ จะได้ตารางใหม่ 1 ตาราง ซึ่งประกอบด้วยแอตทริบิวต์ของความสัมพันธ์นั้นรวมกับแอตทริบิวต์ที่เป็นคีย์หลักของ 2 เอนทิตีที่มีความสัมพันธ์ แบบ $M : M$

จากแบบจำลองอี-อาร์ในภาพที่ 6.11 สามารถสร้างตารางตามขั้นตอนนี้ได้อีก 1 ตาราง คือ ตารางการลงทะเบียน ซึ่งประกอบด้วยแอตทริบิวต์ รหัสนักศึกษา (คีย์หลักของ เอนทิตีนักศึกษา) และ รหัสวิชา (คีย์หลักของเอนทิตีวิชา) ฉะนั้นตารางใหม่ที่เกิดขึ้น ซึ่งก็คือ ตารางการลงทะเบียน มี รหัสนักศึกษาและ รหัสวิชา เป็นคีย์หลัก ดังนี้

การลงทะเบียน	
รหัสนักศึกษา	รหัสวิชา
56111	40001
56111	40005
56112	40001

จากการแปลงแบบจำลองอี-อาร์ตามขั้นตอนข้างต้น สรุปตารางที่ได้ทั้งหมด 3 ตาราง ดังต่อไปนี้

นักศึกษา			
รหัสนักศึกษา	ชื่อ	นามสกุล	เบอร์โทรศัพท์
56111	สามารถ	ประเสริฐกุล	0-1111-1111
56112	วิชา	ปัญญาเลิศ	0-2222-2222

วิชา		
รหัสวิชา	ชื่อวิชา	หน่วยกิต
40001	ภาษาไทย	2
40005	คณิตศาสตร์	3

การลงทะเบียน	
รหัสนักศึกษา	รหัสวิชา
56111	40001
56111	40005
56112	40001

ทั้งนี้โครงสร้างฐานข้อมูลที่ได้จากการแปลงแบบจำลองอี-อาร์นั้นจะอยู่ใน 1NF ดังนั้นจึงจำเป็นต้องนำมาทำนอร์มัลไลเซชันต่อ เพื่อให้ได้ฐานข้อมูลที่ปราศจากความซ้ำซ้อนหรือซ้ำซ้อนน้อยที่สุด แต่

ถ้าได้ทำการออกแบบฐานข้อมูลโดยใช้แบบจำลองอี-อาร์ มาอย่างถูกต้องแล้ว เมื่อแปลงเป็นโครงสร้างฐานข้อมูลแบบสัมพันธ์ จะได้โครงสร้างความสัมพันธ์ที่จัดกลุ่มของแอตทริบิวต์มาเป็นอย่างดี และบางทีโครงสร้างของความสัมพันธ์ที่ได้นั้นอาจอยู่ในนอร์มัลฟอร์มที่สูงกว่านอร์มัลฟอร์มที่ 1 แล้ว อย่างไรก็ตามขั้นตอนถัดมาจำเป็นต้องวิเคราะห์ความสัมพันธ์ระหว่างแอตทริบิวต์ ซึ่งก็คือ วิธีนอร์มัลไลเซชันที่จะกล่าวต่อไป

นอร์มัลไลเซชัน (Normalization)

การออกแบบฐานข้อมูลที่ดี ต้องมีความซ้ำซ้อนในการจัดเก็บข้อมูลน้อยที่สุด หรือไม่มี ความซ้ำซ้อนเลย ซึ่งต้องอาศัยหลักการในการทานอร์มัลไลเซชันดังรายละเอียดต่อไปนี้

1. แนวคิดเกี่ยวกับนอร์มัลไลเซชัน

นอร์มัลไลเซชันเป็นวิธีการที่ใช้ในการตรวจสอบและแก้ไขปัญหาเกี่ยวกับความซ้ำซ้อนของข้อมูล โดยดำเนินการให้ข้อมูล ในแต่ละรีเลชัน (relation) อยู่ในรูปที่เป็นหน่วยที่เล็กที่สุดที่ไม่สามารถแตกออกเป็นหน่วยย่อยๆ ได้อีก โดยยังคงความสัมพันธ์ระหว่างข้อมูลในรีเลชันต่างๆ ไว้ตามหลักการที่กำหนดไว้ใน relational model การทานอร์มัลไลเซชันนี้ เป็นการดำเนินการอย่าง เป็นลำดับ ที่กำหนดไว้ด้วยกันเป็นขั้นตอน ตามปัญหาที่เกิดขึ้นในขั้นต่อนั้นๆ ซึ่งแต่ละขั้นตอน จะมีชื่อตามโครงสร้างข้อมูลที่กำหนดไว้ดังนี้

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. Boyce-Codd Normal Form (BCNF)
5. Fourth Normal Form (4NF) และ
6. Fifth Normal Form (5NF)

ในการออกแบบฐานข้อมูลเพื่อลดความซ้ำซ้อนในการจัดเก็บข้อมูลอย่างน้อยต้องมีคุณสมบัติ เป็น 3 NF เพราะจริงๆ แล้ว ในการทำงานทุกๆ ไป แค่ 3 NF ก็สามารถใช้งานได้แล้ว แต่สำหรับ BCNF ไปจนถึง 5NF เป็นฐานข้อมูลชนิดพิเศษจริงๆ ที่แทบจะไม่มีในชีวิตประจำวัน โอกาสพบประมาณ 0.01 % ดังนั้นในที่นี้จะศึกษาเพียงแค่ 1NF 2NF และ 3NF เท่านั้น เพื่อใช้เป็นแนวทางในการศึกษาระดับอื่นต่อไป

2. รูปแบบของนอร์มัลฟอร์ม (Normal Form : NF)

2.1 First Normal Form (1NF)

ตารางที่ผ่านการทำนอร์มัลไลเซชันระดับที่ 1 หรือ First Normal Form ต้องมีคุณสมบัติดังนี้

**ไม่มีคอลัมน์ใดในตารางที่มีค่ามากกว่า 1 ค่า คือ ค่าในแต่ละคอลัมน์ต้องเป็น atomic
หรือไม่อยู่ในรูปของ repeating group**

หมายความว่าข้อมูลที่เก็บในแต่ละคอลัมน์จะต้องมีลักษณะเป็นค่าเดียว (single valued) ไม่สามารถแบ่งย่อยได้อีก

ในการทำนอร์มัลไลเซชันจะต้องดูข้อมูลในตารางเป็นหลัก ตัวอย่างเช่น ข้อมูลในตาราง แสดงการเก็บข้อมูลเกี่ยวกับนักศึกษา ซึ่งแต่ละคนสามารถอยู่ชมรมและมีงานอดิเรกได้มากกว่า 1 อย่าง

รหัสนักศึกษา	ชื่อ	ที่อยู่	ชมรม	งานอดิเรก
56001	นารี ศิริพร	กรุงเทพ	ดนตรี อาสาพัฒนาชนบท อนุรักษ์สิ่งแวดล้อม	เล่นกีตาร์
56002	ศรีสมร อมรชัย	นนทบุรี	พระพุทธศาสนา	สะสมพระเครื่อง ร้องเพลง
56003	อรอนงค์ สมประสงค์	กรุงเทพ	ดนตรี	ตีกลอง

จากตาราง จะเห็นว่าข้อมูลในคอลัมน์ชมรมและงานอดิเรกมีค่ามากกว่า 1 ค่า แสดงว่าไม่เป็น atomic หรืออยู่ในรูปของ repeating group ดังนั้นตารางที่ 6.5 จึงไม่เป็น 1NF โดยเราจะเรียกตารางที่ยังไม่ผ่านแม้แต่ 1NF ว่า Unnormalized Form (UNF) ซึ่งมีวิธีการ ที่จะทำให้เป็น 1NF คือ

- 1) แยกคอลัมน์ที่มีค่ามากกว่า 1 ค่าออกเป็นแถวใหม่
- 2) เพิ่มข้อมูลที่เหมาะสมเข้าไปในคอลัมน์ที่ว่างอยู่ของแถวที่เกิดขึ้นใหม่

จากตาราง ที่ไม่มีคุณสมบัติเป็น 1NF สามารถทำให้มีคุณสมบัติเป็น 1NF ได้ดังตารางด้านล่าง ซึ่งมีรหัสนักศึกษา ชมรม และงานอดิเรก เป็นคีย์หลัก

รหัสนักศึกษา	ชื่อ	ที่อยู่	ชมรม	งานอดิเรก
56001	นารี ศิริพร	กรุงเทพ	ดนตรี	เล่นกีตาร์
56001	นารี ศิริพร	กรุงเทพ	อาสาพัฒนาชนบท	เล่นกีตาร์
56001	นารี ศิริพร	กรุงเทพ	อนุรักษ์สิ่งแวดล้อม	เล่นกีตาร์
56002	ศรีสมร อมรชัย	นนทบุรี	พระพุทธศาสนา	สะสมพระเครื่อง
56002	ศรีสมร อมรชัย	นนทบุรี	พระพุทธศาสนา	ร้องเพลง
56003	อรอนงค์ สมประสงค์	กรุงเทพ	ดนตรี	ตีกลอง

สาเหตุที่แยกคอลัมน์ที่มีค่ามากกว่า 1 ค่าออกเป็นแถวใหม่ เนื่องจากไม่รู้จำนวนที่แน่นอนของค่าที่มีอยู่ในคอลัมน์นั้น เช่น ไม่ทราบว่านักศึกษาแต่ละคนจะมีงานอดิเรกกันคนละไม่เกินกี่อย่าง แต่ถ้าเราทราบจำนวนที่แน่นอนของคอลัมน์ที่มีหลายค่า นั้น เราอาจแยกเป็นคอลัมน์ใหม่ได้เลย ตัวอย่างเช่น การเก็บชื่อผู้แต่งของหนังสือในห้องสมุด ซึ่งหนังสือเล่มหนึ่งอาจจะมีผู้แต่งหลายคน แต่ในการเก็บชื่อผู้แต่งจะเก็บเพียง 3 คนเท่านั้น ในกรณีนี้ควรแบ่งคอลัมน์ซึ่งเก็บชื่อผู้แต่งออกเป็นหลายคอลัมน์โดยขึ้นกับจำนวนผู้แต่งที่มากที่สุดที่มีอยู่หรือเราต้องการเก็บข้อมูลเอาไว้ ซึ่งจะทำให้ค่าของแต่ละคอลัมน์ เป็น Atomic ดังตารางต่อไปนี้

ISBN	ชื่อหนังสือ	ผู้แต่ง 1	ผู้แต่ง 2	ผู้แต่ง 3
9749151001	การจัดการฐานข้อมูล	วาสนา ทรัพย์แก้ว		
9749151002	ระบบสารสนเทศ	วิเชียร เรียรชัย	นฤมล สมสกุล	
9749151003	เทคโนโลยีสารสนเทศ	สุขุม เฉลยทรัพย์	ปรีศนา มัชฌิมา	กาญจนา เมื่อกอง

ถึงแม้ว่าตารางด้านบน จะได้รับการออกแบบให้อยู่ในรูปแบบ 1NF แล้ว แต่ลักษณะของข้อมูลภายในอาจก่อให้เกิดปัญหาขึ้นได้อีก เช่น ข้อมูลที่เกี่ยวกับนักศึกษารหัส 56001 ถูกจัดเก็บไว้ในแถวที่ 1, 2 และ 3 ได้แก่ ชื่อ และที่อยู่ โดยจะเห็นว่าเป็นการเก็บข้อมูลที่ซ้ำซ้อนกัน ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ และก่อให้เกิดปัญหาในการเปลี่ยนแปลงข้อมูลด้วย เช่น ถ้านักศึกษารหัส 56001 มีการเปลี่ยนชื่อ หรือที่อยู่ ก็ต้องทำการแก้ไขข้อมูลหลายแถว ซึ่งถ้ามีการแก้ไขข้อมูลไม่ครบ ก็อาจทำให้ข้อมูลภายในตารางเกิดความขัดแย้งกันได้ ดังนั้นจึงต้องมีการนอร์มัลไลเซชันระดับที่ 2 ต่อไป

การสร้างฟอร์มรับข้อมูล

นอกจากการใช้เว็บเพจเพื่อนำเสนอเนื้อหาต่างๆแล้ว คุณก็สามารถใช้เว็บเพจเพื่อรับข้อมูลจากผู้ชมเข้ามาเก็บหรือประมวลได้เช่นกัน ตัวอย่างเช่น การใช้แบบสอบถามหรือแบบสำหรับ (poll, vote), สมุดแสดงความคิดเห็น-พักทาย-ติชมเว็บไซต์ (guestbook), การลงทะเบียนสมัครสมาชิก (registration form), กระดานสนทนา (web board) และบันทึกส่วนบุคคล (blog) เป็นต้น ซึ่งในเว็บเพจเหล่านี้องค์ประกอบที่ใช้รับข้อมูลจากผู้ชมก็คือ “ฟอร์ม” (form) นั่นเอง

รู้จักกับฟอร์ม (Form)

ฟอร์ม คือองค์ประกอบบนเว็บเพจที่ทำหน้าที่รับข้อมูลจากผู้ชม แล้วส่งไปยังเว็บเซิร์ฟเวอร์เพื่อประมวลผลอย่างใดอย่างหนึ่ง ภายในฟอร์มจะประกอบด้วยออบเจ็ค (Form Object) ต่างๆ ที่เป็นตัวรับข้อมูลซึ่งมีหลายชนิด เช่น Text Field ,Checkbox และ Radio Button เป็นต้น

นอกจากนี้ภายในฟอร์มยังอาจจะมีองค์ประกอบสำหรับตกแต่งหรือให้คำอธิบาย ซึ่งส่วนนี้ก็คือเนื้อหาปกติของเว็บเพจนั่นเอง ไม่ว่าจะเป็นข้อความ,ภาพกราฟิก,ตาราง ฯลฯ

หลักการทำงานของเว็บในการรับข้อมูลจากผู้ชม

เมื่อผู้ชมกรอกข้อมูลลงในฟอร์มบนเว็บเพจแล้วคลิกปุ่ม Submit ข้อมูลทั้งหมดจะถูกส่งไปยังเว็บเซิร์ฟเวอร์ ซึ่งโปรแกรมสคริปต์หรือแอปพลิเคชันที่กำหนดไว้จะนำข้อมูลดังกล่าวไปประมวลผลดังนั้นการทำงานของฟอร์มจะต้องมีองค์ประกอบเพิ่มขึ้นจากเว็บเพจปกติ 2 อย่างคือ

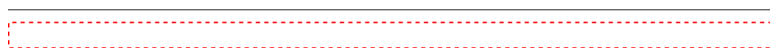
- ฟอร์มบนเว็บเพจ เพื่อใช้รับข้อมูล
- โปรแกรมบนเว็บเซิร์ฟเวอร์ เพื่อประมวลผลข้อมูลและส่งผลลัพธ์กลับไปยังผู้ชม

โปรแกรมสำหรับประมวลผลข้อมูลจากฟอร์ม

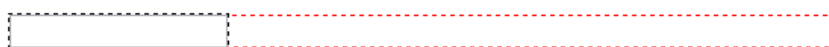
การประมวลผลข้อมูลจากฟอร์มจะขึ้นกับความต้องการของงานแต่ละอย่าง ดังนั้นจึงต้องใช้โปรแกรมที่ถูกเขียนมาโดยเฉพาะ โดยเว็บเซิร์ฟเวอร์จะเป็นเพียงตัวกลางที่คอยรับส่งข้อมูลให้เท่านั้นโปรแกรมดังกล่าวจะใช้ทำงานอะไรและเขียนขึ้นด้วยภาษาใดก็ได้ แต่ส่วนใหญ่แล้วมักนิยมใช้ภาษาประเภทที่เรียกว่า “สคริปต์” (script) เช่น PHP,ASP เป็นต้น แต่ถ้าคุณไม่ใช่ นักเขียนโปรแกรมก็สามารถใช้วิธีอื่นๆต่อไปนี้แทน

องค์ประกอบต่างๆของฟอร์มรับข้อมูล

1. ฟอร์ม (Form)



2. ช่องรับข้อความ (Text Field) เพื่อให้ผู้ใช้ได้กรอกข้อมูล



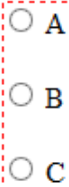
3. ช่องรับข้อความแบบหลายบรรทัด (Text Area) มีลักษณะเหมือนกันกับ Text Field แต่รับข้อความได้หลายบรรทัดมากกว่า หากความยาวข้อความเกินขนาดกล่องจะเพิ่ม Scroll bar เพื่อเลือกดูข้อความที่เกินมาได้



4. กล่องตัวเลือก (Check box) มีลักษณะเป็นกล่องสี่เหลี่ยมเล็กๆ ใช้สำหรับคลิกรายการที่ต้องการเลือก เราสามารถสร้างกล่องได้หลายกล่อง ในแต่ละรายการเลือก รวมทั้งสามารถเลือกได้หลายรายการพร้อมกัน โดยจะมีเครื่องหมายถูกที่กล่องที่เลือก



5. ปุ่มตัวเลือก (Radio Button) มีลักษณะเป็นกล่องกลมเล็กๆ ใช้สำหรับคลิกรายการที่ต้องการเลือก เราสามารถสร้างกล่องได้หลายกล่อง ในแต่ละรายการเลือก รวมทั้งสามารถเลือกได้หลายรายการพร้อมกัน โดยจะมีเครื่องหมายถูกที่กล่องที่เลือก



5. กล่องรายการ (List Box) มีลักษณะเป็นกล่องสี่เหลี่ยม บรรจุรายการตัวเลือกต่างๆ ถ้าจะเลือกรายการก็ทำการคลิกให้กล่องนี้เลื่อนข้อมูลออกมาจากกล่อง



6. กล่องรับชื่อไฟล์ (File Field) มีลักษณะเป็นกล่องสี่เหลี่ยม ใช้สำหรับเลือกไฟล์ เพื่อใช้ในการอัปโหลดไฟล์ไปยังเซิร์ฟเวอร์ ลักษณะคล้ายกล่องรับข้อความ



7. ปุ่มคำสั่ง (Button) การรับข้อมูลที่ใช้ได้ทำการกรอกหรือเลือก และนำไปประมวลผลต่อไป จะต้องใช้ปุ่มคำสั่งทำการส่งข้อมูลนำเข้าต่างๆ เหล่านั้น (Submit)

ตัวอย่างการสร้างฟอร์มรับข้อมูล

ดูตัวอย่างไฟล์ >> http://std.kku.ac.th/5330503366/Form/Form_register.php

รายละเอียด Code จากไฟล์ด้านบน

```
<form action="register.php" method="post" enctype="multipart/form-data"
name="form1" id="form1">
```

```
<label for="username"></label>
```


```
<input type="text" name="username" id="username" />
```

```
<label for="pass"></label>
```

```
<input type="password" name="pass" id="pass" />
```


```
<label for="conpass"></label>
```

```
<input type="password" name="conpass" id="conpass" />
```



```
<label for="name"></label>
```

```
<input type="text" name="name" id="name" />
```




```
<label for="mail"></label>
```

```
<input type="text" name="mail" id="mail" />
```



```
<label for="address"></label>
```

```
<textarea name="address" id="address" cols="45" rows="7"></textarea>
```

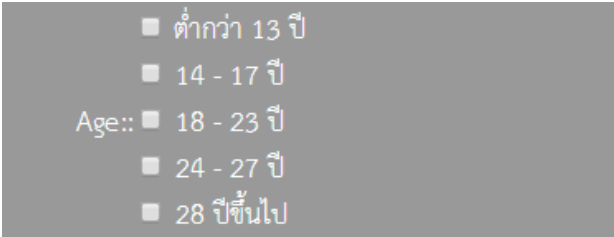


```
<input type="radio" name="gender" id="radio" value="Male" />
```

```
<label for="radio" class="thw">Male
```

```
<input type="radio" name="gender" id="radio2" value="Female" />
```

```
Female</label>
```



```
<label for="select">
```

```
<input name="age1" type="checkbox" id="checkbox" value="ต่ำกว่า 13 ปี" />
```

```
<span class="thw">ต่ำกว่า 13 ปี <br />
```

```
<input name="age2" type="checkbox" id="checkbox2" value="14 - 17 ปี" />
```

```
14 - 17 ปี<br />
```

```
<input name="age3" type="checkbox" id="checkbox3" value="18 - 23 ปี" />
```

```
18 - 23 ปี<br />
```

```
<input name="age4" type="checkbox" id="checkbox4" value="24 - 27 ปี" />
```

```
24 - 27 ปี<br />
```

```
<input name="age5" type="checkbox" id="checkbox5" value="28 ปีขึ้นไป" />
```

```
28 ปีขึ้นไป</span></label>
```



ในส่วนตัวเลือกของ Day, Month, Year จะมีลักษณะของ Code m ใกล้เคียงกัน

```
<label for="select"></label>
```

```
<select name="day" id="select">
```

```
<option value="Sunday">Sunday</option>
```

```
<option value="Monday">Monday</option>
```

```
<option value="Tuesday">Tuesday</option>
```

```
<option value="Wednesday">Wednesday</option>
```

```
<option value="Thursday">Thursday</option>
```

```
<option value="Friday">Friday</option>
```

```
<option value="Saturday">Saturday</option>
```

```
</select>
```

ตัวอย่างการสร้างฟอร์มแสดงข้อมูล

ข้อมูลสมาชิก

:KunKroo TUm
Username::

Password:::1234
Confirm-Password:::1234

Name-Last Name:::Jenrop Krotha
E-mail:::jenropkrota_99@hotmail.com

Address:::KKU
Sex:::Male
Age:: , ,18 - 23 ปี , ,
Birthday::Day:: Sunday Month:: April Year:: 2534

Choose File for upload::

[<< Back](#)

ดูตัวอย่างไฟล์ >> <http://std.kku.ac.th/5330503366/Form/register.php>

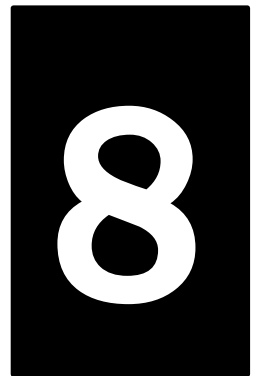
<?php

```
//-----ประกาศตัวรับค่าที่ส่งมาจาก-form-1-----
$username = $_POST['username'];
$pass = $_POST['pass'];
$conpass = $_POST['conpass'];
$name = $_POST['name'];
$email = $_POST['mail'];
$address = $_POST['address'];
$gender = $_POST['gender'];
$age1 = $_POST['age1'];
$age2 = $_POST['age2'];
$age3 = $_POST['age3'];
$age4 = $_POST['age4'];
$age5 = $_POST['age5'];
$day = $_POST['day']; $month = $_POST['month']; $year = $_POST['year'];
```

?>

ส่วนของ Code ที่ใช้แสดงข้อมูล

Username	<code><? echo ":" . \$username. "
"; ?></code>
Password	<code><? echo ":" . \$pass. "
"; ?></code>
Confirm-Password	<code><? echo ":" . \$pass. "
"; ?></code>
Name-Last Name	<code><? echo ":" . \$name. "
"; ?></code>
E-mail	<code><? echo ":" . \$email. "
"; ?></code>
Address	<code><? echo ":" . \$address. "
"; ?></code>
Sex	<code><? echo ":" . \$gender. "
"; ?></code>
Age	<code><? echo ":" . \$age1; echo " , " . \$age2; echo " , " . \$age3; echo " , " . \$age4; echo " , " . \$age5; ?></code>
Birthday	<code>Day <? echo " : " . \$day."
"; ?> Month <? echo " : " . \$month."
"; ?> Year <? echo " : " . \$year."
"; ?></code>



การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin

การจัดการฐานข้อมูล MySQL ด้วย PhpMyAdmin

PhpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการฐานข้อมูล MySQL ผ่านบราวเซอร์ได้โดยตรง โปรแกรมนี้เขียนขึ้นโดยนาย Tobias Retschler และเป็นโปรแกรมแบบ Open Source ด้วย เพราะว่าเขียนด้วย PHP ทั้งหมด

PhpMyAdmin ตัวนี้จะทำงานบน Web Server เป็น PHP Application ที่ใช้ควบคุมจัดการกับ MySQL Server ความสามารถของ PhpMyAdmin คือ

1. สร้างและลบ Database
 2. สร้างและจัดการ Table เช่น แทรก record, ลบ record, แก้ไข record หรือลบ Table, เพิ่มหรือแก้ไข field ในตาราง
 3. โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
 4. ทหาผลสรุป (Query) ด้วยคำสั่ง SQL
- และอีกหลายๆ ความสามารถที่ PhpMyAdmin ทำได้ ซึ่งเราจะมาเรียนกันภายในบทนี้

การติดตั้ง PhpMyAdmin

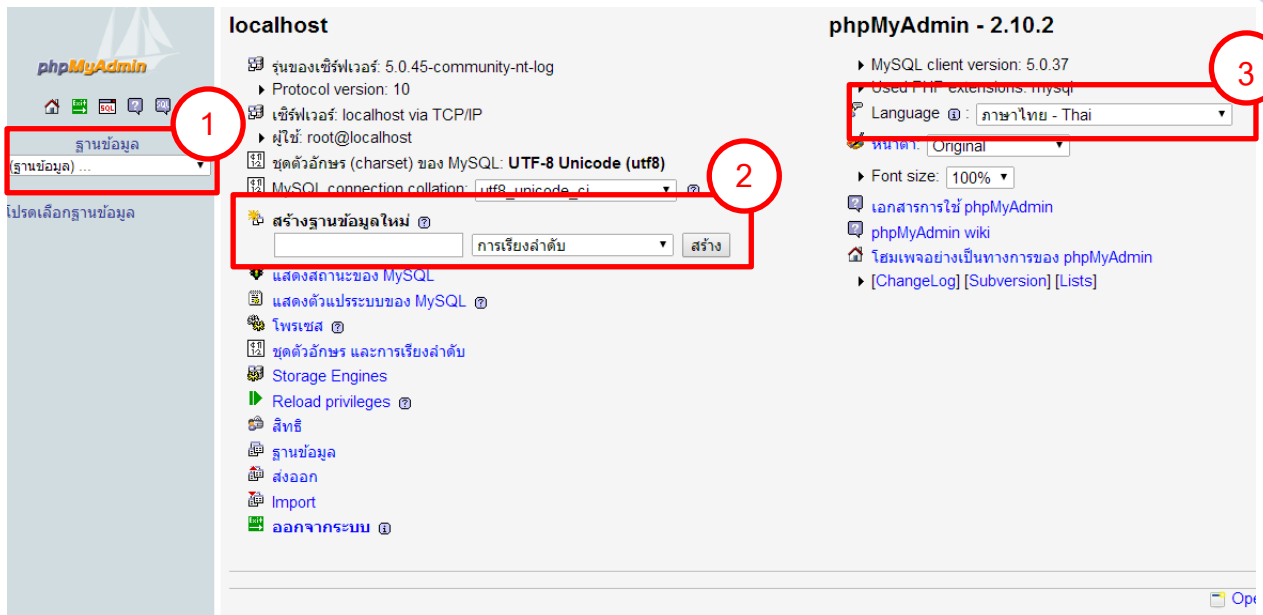
ก่อนอื่น ให้ดาวน์โหลดโปรแกรม PhpMyAdmin จากเว็บไซต์

<http://www.appservnetwork.com/>

การติดตั้งสำหรับระบบ Windows สำหรับการติดตั้งบนระบบปฏิบัติการ Windows ให้ทำการ unzip ไฟล์ PhpMyAdmin มีนามสกุลไฟล์เป็น .zip (ชื่อไฟล์อาจจะไม่เหมือนกัน เนื่องจากเปลี่ยนแปลงตามเวอร์ชัน) ที่ดาวน์โหลดมา ไปไว้ยัง root ไดเรกทอรีของเว็บเซิร์ฟเวอร์ ซึ่งในที่นี้คือ C:\AppServ\www ในลำดับถัดไปให้เข้าไปยังไดเรกทอรี C:\AppServ\www จะพบกับ ไดเรกทอรี phpMyAdmin-x.x.x (x แทนเวอร์ชัน) จากนั้นเปลี่ยนชื่อจาก phpMyAdmin-x.x.x ให้เป็น PhpMyAdmin เพื่อให้สะดวกต่อการเรียกใช้งาน

การเข้าใช้งาน

การเข้าใช้งานโปรแกรม PhpMyAdmin จะใช้งานผ่านโปรแกรม Internet Explorer และพิมพ์ URL ไปยังไดเรกทอรีของ PhpMyAdmin เช่น <http://localhost/phpMyAdmin> เป็นต้น (ชื่อไดเรกทอรีอาจจะเป็นอย่างอื่น ขึ้นอยู่กับขั้นตอนการเปลี่ยนชื่อไดเรกทอรี) และท่านต้องแน่ใจว่าได้ติดตั้งโปรแกรม webserver เช่น Apache เป็นต้น ไว้ภายในเครื่องเป็นที่เรียบร้อยแล้ว ซึ่งหน้าจอแรกในการใช้งาน แสดงดังรูป



จากรูป จะเห็นได้ว่าโปรแกรมได้แบ่งส่วนของการแสดงผลออกเป็น 2 ส่วน คือ ส่วนซ้าย(เลข 1) และ ส่วนขวา (เลข 2, 3) ซึ่งจะอธิบายได้ดังนี้

1. ใช้สำหรับ เลือก/เปลี่ยน จัดการกับฐานข้อมูลเดิมที่มีอยู่แล้ว
2. สำหรับใส่ชื่อฐานข้อมูลใหม่ที่ต้องการสร้าง
3. สำหรับเลือกภาษาที่ต้องการให้แสดงภายในโปรแกรม PhpMyAdmin

การสร้างฐานข้อมูลใหม่

ก่อนที่จะเก็บข้อมูลลงใน MySQL จะต้องเลือกเสียก่อนว่า จะสร้างฐานข้อมูลชื่ออะไร จากนั้นจึงทำการสร้างตาราง และ ฟิลด์ เพื่อบันทึกข้อมูล ในอันดับแรกจะเป็นการทดลองสร้าง ฐานข้อมูลขึ้นมาใหม่ โดยจะยกตัวอย่างฐานข้อมูลเกี่ยวกับสินค้า ซึ่งมีชื่อฐานข้อมูลว่า “store” ดังรูป



จากรูป อธิบายขั้นตอนการสร้างฐานข้อมูลใหม่ตามลำดับ ได้ดังนี้

1. ใส่ชื่อของฐานข้อมูลที่ต้องการสร้าง ในที่นี้ให้ชื่อว่า store
2. คลิกที่ปุ่ม สร้าง เพื่อสร้างฐานข้อมูล

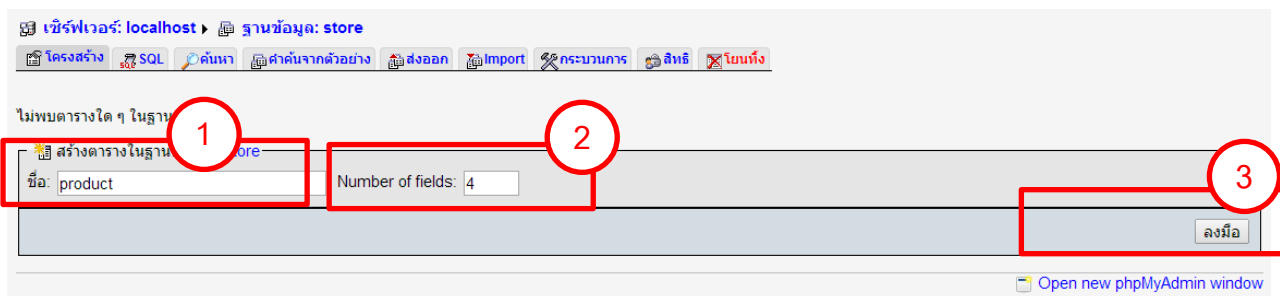
การสร้างตารางใหม่

ภายหลังจากที่สร้างฐานข้อมูลเรียบร้อยแล้ว PhpMyAdmin จะแจ้งให้เราทราบว่า ได้ทำการสร้างฐานข้อมูล store เรียบร้อยแล้ว และท่านใช้งานฐานข้อมูล store อยู่ ยังไม่มีตารางใดๆ อยู่เลย ในที่นี่จะสร้างตารางสำหรับเก็บข้อมูลสินค้า โดยรายละเอียดในตารางจะมี 3 ฟิลด์ ดังนี้

- product_id รหัสสินค้า
- product_name ชื่อสินค้า
- product_price ราคาสินค้า
- product_num จำนวนสินค้า (ย่อมาจาก number แต่เขียนให้ง่ายและสั้นขึ้น)

ทำการสร้างตารางใหม่ โดยมีขั้นตอนดังนี้

1. ใส่ชื่อตารางที่ต้องการสร้าง
2. ใส่จำนวนฟิลด์ของตาราง สำหรับตาราง product จะมีจำนวน 4 ฟิลด์
3. คลิกปุ่ม ลงมือ เพื่อดำเนินการขั้นต่อไป



ขั้นตอนต่อไป ภายหลังจากคลิกปุ่ม ลงมือในขั้นตอนก่อนหน้า จะเป็นการกำหนด ชื่อฟิลด์ ชนิดของข้อมูลที่เก็บ และรายละเอียดอื่นๆ ซึ่งจะอธิบายในส่วนต่างๆ ได้ดังนี้

- ฟิลด์ สำหรับใส่ชื่อฟิลด์
- ชนิด สำหรับเลือกชนิดของข้อมูลที่ต้องการเก็บในฟิลด์นั้น
- ความยาว/เซต สำหรับกำหนดขนาดของข้อมูล
- แอตทริบิวต์ สำหรับเลือกลักษณะเฉพาะของข้อมูลที่จะเก็บ เช่น ตัวเลขแบบคิดเครื่องหมาย บวก หรือลบ เป็นต้น

หรือลบ เป็นต้น

- ค่าว่างเปล่า (null) สำหรับเลือกว่า ฟิลด์นั้นสามารถใส่ค่าว่างได้หรือไม่
- ค่าปริยาย สำหรับกำหนดค่าเริ่มต้นของฟิลด์ (ค่า Default)

- เพิ่มเติม สำหรับกำหนดค่าเพิ่มเติม เช่น กรณีที่ฟิลด์เก็บข้อมูลตัวเลขจำนวนเต็ม(Integer) จะสามารถเลือกให้มีการเพิ่มค่าอัตโนมัติ (auto_increment) ได้ เป็นต้น
- ไพรมารี เลือกเมื่อต้องการกำหนดให้ฟิลด์นั้นๆ เป็นไพรมารีคีย์ (Primary Key)
- ดัชนี เลือกเมื่อต้องการสร้างฟิลด์นั้นๆ เป็นดัชนี (Index)
- เอกลักษณ์ เลือกเมื่อต้องการให้ฟิลด์นั้นเป็น Unique

ต่อไปนี้เป็นคำอธิบายของชนิดข้อมูลแต่ละชนิดที่สามารถเลือกได้

VARCHAR : สำหรับเก็บข้อมูลประเภทตัวอักษร ทุกครั้งที่เลือกชนิดของฟิลด์เป็นประเภทนี้ จะต้องมีการกำหนดความยาวของข้อมูลลงไปด้วย ซึ่งสามารถกำหนดค่าได้ตั้งแต่ 1 - 255 ฟิลด์ชนิดนี้ เหมาะสำหรับการเก็บข้อมูลสั้นๆ เช่น ชื่อ นามสกุล หรือหัวข้อต่างๆ เป็นต้น... ในส่วนฟิลด์ประเภทนี้ จะสามารถเลือก "แอสกีไบต์" เป็น BINARY ได้โดยปกติแล้วการจัดเรียงข้อมูลเวลาสืบค้น (query) สำหรับ VARCHAR จะเป็นแบบ casesensitive (ตัวอักษรใหญ่ และเล็กมีความหมายแตกต่างกัน) แต่หากรระบุ "แอสกีไบต์" เป็น BINARY บั๊บ การสืบค้นจะไม่คำนึงตัวอักษรว่าจะเป็นตัวใหญ่ หรือตัวเล็ก

CHAR: สำหรับเก็บข้อมูลประเภทตัวอักษร แบบที่ถูกจำกัดความกว้างเอาไว้คือ 255 ตัวอักษร ไม่สามารถปรับเปลี่ยนได้เหมือนกับ VARCHAR หากทำการสืบค้นโดยเรียงตามลำดับ ก็จะได้เรียงข้อมูลแบบ case-sensitive เว้นแต่จะกำหนดแอสกีไบต์เป็น BINARY ที่จะทำให้การเรียงข้อมูลเป็นแบบ non case-sensitive เช่นเดียวกับ VARCHAR

TINYTEXT: ในกรณีที่ข้อความยาวๆ หรือต้องการที่จะค้นหาข้อความ โดยอาศัยฟูลเจอร์ FULL TEXT SEARCH ของ MySQL เราอาจจะเลือกที่จะไม่เก็บข้อมูลลงในฟิลด์ประเภท VARCHAR ที่มีข้อจำกัดแค่ 256 ตัวอักษร แต่เราจะเก็บลงฟิลด์ประเภท TEXT แทน โดย TINYTEXT นี้ จะสามารถเก็บข้อมูลได้ 256 ตัวอักษร ซึ่งมองเผินๆ ก็ไม่ต่างกับเก็บลงฟิลด์ ประเภท CHAR หรือ VARCHAR (255) เลย แต่จริงๆ มันต่างกันตรงที่มันทำ FULL TEXT SEARCH ได้

TEXT: สำหรับเก็บข้อมูลประเภทตัวอักษร เช่นเดียวกับ TINYTEXT แต่สามารถเก็บได้มากขึ้น โดยสูงสุดคือ 65,535 ตัวอักษร หรือ 64KB เหมาะสำหรับการเก็บข้อมูลพวกเนื้อหาต่างๆ ที่ยาวๆ

MEDIUMTEXT: เก็บข้อมูลประเภทตัวอักษร เช่นเดียวกับ TINYTEXT แต่เก็บข้อมูลได้ 16,777,215 ตัวอักษร

LONGTEXT: เก็บข้อมูลประเภทตัวอักษร เช่นเดียวกับ TINYTEXT แต่เก็บข้อมูลได้ 4,294,967,295 ตัวอักษร

TINYINT: สำหรับเก็บข้อมูลชนิดตัวเลขที่มีขนาด 8 บิต ข้อมูลประเภทนี้เราสามารถกำหนดเพิ่มเติมในส่วนของ "แอตทริบิวต์" ได้ว่าจะเลือกเป็น UNSIGNED หรือ UNSIGNED ZEROFILL โดยจะมีความแตกต่างกันดังนี้

-**UNSIGNED :** จะหมายถึงเก็บค่าตัวเลขแบบไม่มีเครื่องหมาย แบบนี้จะทำให้สามารถเก็บค่าได้ตั้งแต่ 0 – 255

-**UNSIGNED ZEROFILL :** เหมือนข้างต้น แต่ว่าหากข้อมูลที่กรอกเข้ามาไม่ครบตามจำนวนหลักที่เรากำหนด ตัว MySQL จะทำการเติม 0 ให้ครบหลักเอง เช่น ถ้ากำหนดให้ใส่ได้ 3 หลัก แล้วทำการเก็บข้อมูล 25 เข้าไป เวลาที่สืบค้นดู เราจะได้ค่าออกมาเป็น 025

หากไม่เลือก "แอตทริบิวต์" สิ่งที่เราจะได้ก็คือ SIGNED นั่นก็คือต้องเสียบิตหนึ่งไปเก็บเครื่องหมายบวก/ลบ ทำให้สามารถเก็บข้อมูลได้อยู่ในช่วง -128 ถึง 127 เท่านั้น

SMALLINT: สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 16 บิต จึงสามารถเก็บค่าได้ตั้งแต่ -32768 ถึง 32767 (ในกรณีแบบคิดเครื่องหมาย) หรือ 0 ถึง 65535 (ในกรณี UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

MEDIUMINT: สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 24 บิต นั่นก็หมายความว่าสามารถเก็บข้อมูลตัวเลขได้ตั้งแต่ -8388608 ไปจนถึง 8388607 (ในกรณีแบบคิดเครื่องหมาย) หรือ 0 ถึง 16777215 ในกรณีที่เลือกเป็น UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

INT: สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 32 บิต หรือสามารถเก็บข้อมูลได้ตั้งแต่ -2147483648 ไปจนถึง 2147483647 ครึ่ง (ในกรณีแบบคิดเครื่องหมาย) หรือ 0 ถึง 4294967295 (ในกรณีที่เลือกเป็น UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

BIGINT: สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 64 บิต สามารถเก็บข้อมูลได้ตั้งแต่ -9223372036854775808 ไปจนถึง 9223372036854775807 เลขที่เดียว (แบบคิดเครื่องหมาย) หรือ 0 ถึง 18446744073709551615 (ในกรณีที่เลือกเป็น UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

FLOAT [(M, D)]: ที่กล่าวถึงไปทั้งหมด ในตระกูล INT นั้นจะเป็นเลขจำนวนเต็ม หากเราบันทึกข้อมูลที่มีเศษทศนิยม มันจะถูกปัดทันทันที่ ดังนั้นหากต้องการจะเก็บค่าที่เป็นเลขทศนิยม ต้องเลือกชนิดของฟิลด์เป็น FLOAT โดยจะเก็บข้อมูลแบบ 32 บิต คือมีค่าตั้งแต่ $-3.402823466E+38$ ไปจนถึง $-1.175494351E-38$, 0 และ $1.175494351E-38$ ถึง $3.402823466E+38$

DOUBLE [(M, D)]: สำหรับเก็บข้อมูลประเภทตัวเลขทศนิยม เช่นเดียวกับ FLOAT แต่มีขนาดเป็น 64 บิต สามารถเก็บได้ตั้งแต่ $-1.7976931348623157E+308$ ถึง $-2.2250738585072014E-308$, 0 และ $2.2250738585072014E-308$ ถึง $1.7976931348623157E+308$

DECIMAL [(M, D)]: สำหรับเก็บข้อมูลประเภทตัวเลขทศนิยม เช่นเดียวกับ FLOAT แต่ใช้กับข้อมูลที่ต้องการความละเอียดและถูกต้องของข้อมูลสูง

ข้อสังเกต เกี่ยวกับข้อมูลประเภท *FLOAT*, *DOUBLE* และ *DECIMAL* ก็คือ เวลากำหนดความยาวของข้อมูลในฟิลด์ จะถูกกำหนดอยู่ในรูปแบบ (M,D) ซึ่งหมายความว่า ต้องมีการระบุว่าจะให้มีตัวเลขส่วนที่เป็นจำนวนเต็มกี่หลัก และมีเลขทศนิยมกี่หลัก เช่น ถ้าเรากำหนดว่า *FLOAT(5,2)* จะหมายความว่า เราจะเก็บข้อมูลเป็นตัวเลขจำนวนเต็ม 5 หลักและทศนิยม 2 หลัก ดังนั้นหากทำการใส่ข้อมูล 12345.6789 เข้าไป สิ่งที่จะเข้าไปอยู่ในข้อมูลจริงๆ ก็คือ 12345.68 (ขีดเศษให้มีจำนวนหลักตามที่กำหนดไว้)

DATE: สำหรับเก็บข้อมูลประเภทวันที่ โดยเก็บได้จาก 1 มกราคม ค.ศ. 1000 ถึง 31 ธันวาคม ค.ศ. 9999 โดยจะแสดงผลในรูปแบบ YYYY-MM-DD

DATETIME: สำหรับเก็บข้อมูลประเภทวันที่ และเวลา โดยจะเก็บได้ตั้งแต่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 ไปจนถึง 31 ธันวาคม ค.ศ. 9999 เวลา 23:59:59 โดยรูปแบบการแสดงผล เวลาที่ทำการสืบค้น (query) ออกมา จะเป็น YYYY-MM-DD HH:MM:SS

TIMESTAMP [(M)]: สำหรับเก็บข้อมูลประเภทวันที่ และเวลาเช่นกัน แต่จะเก็บในรูปแบบของ YYYYMMDDHHMMSS หรือ YYMMDDHHMMSS หรือ YYYYMMDD หรือ YYMMDD แล้วแต่ว่าจะระบุค่า M เป็น 14, 12, 8 หรือ 6 ตามลำดับ สามารถเก็บได้ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 ไปจนถึงประมาณปี ค.ศ. 2037

TIME: สำหรับเก็บข้อมูลประเภทเวลา มีค่าได้ตั้งแต่ -838:59:59 ไปจนถึง 838:59:59 โดยจะแสดงผลออกมาในรูปแบบ HH:MM:SS

YEAR [(2/4)]: สำหรับเก็บข้อมูลประเภทปี ในรูปแบบ YYYY หรือ YY แล้วแต่ว่าจะเลือก 2 หรือ 4 (หากไม่ระบุ จะถือว่าเป็น 4 หลัก) โดยหากเลือกเป็น 4 หลัก จะเก็บค่าได้ตั้งแต่ ค.ศ. 1901 ถึง 2155 แต่หากเป็น 2 หลัก จะเก็บตั้งแต่ ค.ศ. 1970 ถึง 2069

ข้อสังเกต ค่าที่เก็บในข้อมูลประเภท **TIMESTAMP** และ **YEAR** นั้นจะมีความสามารถพอๆ กับ การเก็บข้อมูลวันเดือนปี และเวลา ด้วยฟิลด์ชนิด **VARCHAR** แต่ต่างกันตรงที่ จะใช้เนื้อที่เก็บข้อมูลน้อยกว่า... ทว่า ฟิลด์ประเภท **TIMESTAMP** นั้นจะมีข้อจำกัดในเรื่องของเวลาที่สามารถเก็บได้ คือจะต้องอยู่ในระหว่าง 1 มกราคม ค.ศ. 1000 ไปจนถึงแถวๆ ค.ศ. 2037 อย่างที่บอกแต่หากเก็บเป็น **VARCHAR** นั้นจะไม่ติดข้อจำกัดนี้

ฟิลด์ชนิด YEAR ก็เช่นกันครับ... ใช้เนื้อที่แค่ 1 ไบต์เท่านั้นในการเก็บข้อมูล แต่ข้อจำกัดจะอยู่ที่ ปี ค.ศ. 1901 ถึง 2155 เท่านั้น (หรือ ค.ศ. 1970 ถึง 2069 ในกรณี 2 หลัก) แต่หากเก็บเป็น **VARCHAR** จะได้ตั้งแต่ 0000 ถึง 9999 เลย อันนี้เลยอยู่ที่ความจำเป็นมากกว่าครับ (แต่ด้วยความที่ว่า ปัจจุบันฮาร์ดดิสก์ราคาถูกลงๆ ผมเลยไม่ติดใจอะไรที่จะใช้ **VARCHAR** แทน เพื่อความสบายใจ อ้อ เพราะสมมติว่ากินเนื้อที่ต่างกัน 3 ไบต์ ต่อ 1 ระเบียบน มีข้อมูล 4 ล้านระเบียบน ก็เพ็งต่างกัน 12 ล้านไบต์ หรือ 12เมกะไบต์เท่านั้นเอง ซึ่งหากเทียบกับปริมาณข้อมูลทั้งหมดของข้อมูล 4 ล้านระเบียบนผมว่ามันต้องมือน้อยเป็นกิกะไบต์ ดังนั้นความแตกต่างที่ไม่กี่เมกะไบต์จึงไม่มากมายอะไรครับ)

TINYBLOB: สำหรับเก็บข้อมูลประเภทไบนารี ได้แก่ ไฟล์ข้อมูลต่างๆ, ไฟล์รูปภาพ, ไฟล์มัลติมีเดีย เป็นต้น คือไฟล์อะไรก็ตามที่อัปโหลดผ่านฟอร์มอัปโหลดไฟล์ในภาษา HTML โดย **TINYBLOB** นั้นจะมีเนื้อที่ให้เก็บข้อมูลได้ 256 ไบต์

BLOB: สำหรับเก็บข้อมูลประเภทไบนารี เช่นเดียวกับ **TINYBLOB** แต่สามารถเก็บข้อมูลได้ 64KB

MEDIUMBLOB: สำหรับเก็บข้อมูลประเภทไบนารี เช่นเดียวกับ **TINYBLOB** แต่เก็บข้อมูลได้ 16MB

LOB: สำหรับเก็บข้อมูลประเภทไบนารี เช่นเดียวกับ **TINYBLOB** แต่เก็บข้อมูลได้ 4GB

ข้อสังเกต ข้อมูลประเภท **BLOB** นั้น แม้จะมีประโยชน์ในเรื่องของการเก็บข้อมูลประเภท **BINARY** ให้อยู่กับตัวฐานข้อมูล ทำให้สะดวกเวลาสืบค้นก็ตาม แต่มันก็ทำให้ฐานข้อมูลมีขนาดใหญ่เกินความจำเป็นด้วย

ทำให้เกิดความไม่สะดวกในการสำรองฐานข้อมูลในกรณีที่ มีข้อมูลอัปเดตไปเก็บมากๆ โดยปกติแล้ว จะใช้วิธีการอัปเดตไปเก็บไว้ในไฟล์เดอร์แล้วเก็บลิงค์ไปยังไฟล์เหล่านั้น เป็นฟิลด์ชนิด VARCHAR มากกว่า

SET: สำหรับเก็บข้อมูลที่เป็นกลุ่มของข้อมูลที่ยอมให้เลือกได้ 1 ค่าหรือหลายๆ ค่า ซึ่งสามารถกำหนดได้ถึง 64 ค่า

ให้กรอกรายละเอียดต่างๆ ให้ครบ เช่น สร้างฟิลด์ 4 ฟิลด์ เราต้องระบุชื่อฟิลด์ ชนิดของข้อมูลที่จัดเก็บ หรือรายละเอียดอื่นๆ ให้ครบทั้ง 4 ฟิลด์ก่อน เมื่อเรียบร้อยแล้วคลิกที่ปุ่ม บันทึก เพื่อสร้างตารางใหม่ตามรายละเอียดที่ได้กำหนดไว้ ดังรูป

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: store ▶ ตาราง : product

ฟิลด์	ชนิด	ความยาว/เขต*	การเรียงลำดับ	แอตทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย ²	เพิ่มเติม				
product_id	INT	3			not null		auto_increment	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
product_name	VARCHAR	50			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
product_price	VARCHAR	4			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
product_num	VARCHAR	4			not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

หมายเหตุของตาราง:

Storage Engine: MyISAM

การเรียงลำดับ:

บันทึก หรือ Add 1 field(s) ลงมือ

หลังจากคลิกที่ปุ่ม บันทึก แล้ว โปรแกรมจะแจ้งให้ทราบว่า ได้สร้างตาราง product เรียบร้อยแล้ว และจะเข้าสู่หน้าจอหน้าจอสําหรับการจัดการตาราง ภายในหน้าจอจะมีเมนูต่างๆ หลายเมนู ซึ่งจะอธิบายในลำดับถัดไป และ มีการแจ้งให้ผู้ใช้ทราบรายละเอียดอื่นๆ ที่เกี่ยวข้องกับตารางนั้นๆ เช่น ดัชนี เนื้อที่ที่ใช้งาน ค่าสถิติต่างๆ เป็นต้น ดังรูป

หน้าจอภายหลังสร้างตารางใหม่

เบราว์เซอร์: localhost ▶ ฐานข้อมูล: store ▶ ตาราง: product

เปิด ▶ โครงสร้าง ▶ SQL ▶ ค้นหา ▶ แทรก ▶ ส่งออก ▶ Import ▶ กระบวนการ ▶ ลบข้อมูล ▶ โหม่ง

ตาราง 'store'. 'product' สร้างเสร็จแล้ว

คำสั่ง SQL:

```
CREATE TABLE `product` (
  `product_id` INT(3) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `product_name` VARCHAR(50) NOT NULL,
  `product_price` VARCHAR(4) NOT NULL,
  `product_num` VARCHAR(4) NOT NULL
) ENGINE = MYISAM;
```

[แก้ไข] [สร้างโค้ด PHP]

ฟิลด์	ชนิด	การเรียงลำดับ	แอตทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย	เพิ่มเติม	กระทำการ
<input type="checkbox"/> product_id	int(3)			ไม่		auto_increment	
<input type="checkbox"/> product_name	varchar(50)	utf8_general_ci		ไม่			
<input type="checkbox"/> product_price	varchar(4)	utf8_general_ci		ไม่			
<input type="checkbox"/> product_num	varchar(4)	utf8_general_ci		ไม่			

เลือกทั้งหมด / ไม่เลือกเลย ทำกับที่เลือก:

แสดง ▶ เสนอโครงสร้างตาราง ▶

Add 1 field(s) ที่จุดสุดท้ายของตาราง ที่จุดเริ่มต้นของตาราง หลัง product_id ▶ ลงมือ

ดัชนี	ชื่อ	ชนิด	Cardinality	กระทำการ	ฟิลด์	ชนิด	ใช้งาน	สเกิลของ แถว
PRIMARY	PRIMARY		0		product_id	ข้อมูล	0 ไบต์	คำสั่ง ค่า
สร้างดัชนีโดยคอลัมน์	1			ลงมือ		ดัชนี	0 ไบต์	
						รวม	0 ไบต์	

การจัดการกับฟิลด์และข้อมูลของตาราง

1. การเรียกดูโครงสร้างของตาราง การเรียกดูโครงสร้างของตาราง เราจะต้องอยู่ในหน้าจอการจัดการตารางก่อน ซึ่งการเข้ามายังหน้าจอนี้ได้ มี 2 วิธี คือ คลิกจากชื่อตารางซึ่งแสดงอยู่ส่วนซ้ายของโปรแกรมแสดงดังรูป หรือ ภายหลังจากสร้างตารางใหม่ก็จะเข้ามายังหน้าจอจัดการตารางเช่นเดียวกัน

ภายในหน้าจอจัดการตาราง โปรแกรมจะแสดงโครงสร้างของตารางเป็นหน้าจอแรก หรือ เราสามารถคลิกที่ เมนู โครงสร้าง เพื่อแสดงโครงสร้างของข้อมูลได้เช่นเดียวกันดังรูป โปรแกรมจะแสดงโครงสร้างของตารางนั้นๆ อันประกอบไปด้วย ชื่อฟิลด์ ชนิดของข้อมูล แอตทริบิวต์ การใส่ค่าว่าง และไอคอนสำหรับจัดการดังนี้



สำหรับแก้ไข ชื่อฟิลด์ ชนิดของข้อมูลที่เก็บ ไพรมารีคีย์ หรือค่าเริ่มต้นอื่นๆ



สำหรับลบฟิลด์นั้นๆ ออกจากตาราง



สำหรับกำหนดให้ฟิลด์นั้นๆ เป็นไพรมารีคีย์ (Primary Key)



สำหรับกำหนดให้ฟิลด์นั้นๆ เป็นดัชนี (Index)



สำหรับกำหนดให้ฟิลด์นั้นๆ เป็นเอกลักษณ์ (Unique)

Screenshot of phpMyAdmin interface showing the 'โครงสร้าง' (Structure) tab for a 'product' table. The 'product_id' field is highlighted with a red box. Below the table structure, there are options to 'แสดง' (Show) or 'เสนอโครงสร้างตาราง' (Propose table structure), and a 'Add' button with a dropdown menu set to 'product_id'.

ฟิลด์	ชนิด	การเรียงลำดับ	แอดทริบิวต์	ว่างเปล่า (null)	ค่าปริยาย	เพิ่มเดิม	กระทำการ
product_id	int(3)			ไม่		auto_increment	[ดู] [แก้ไข] [ลบ] [เพิ่ม] [รีเซ็ต]
product_name	varchar(50)	utf8_general_ci		ไม่			[ดู] [แก้ไข] [ลบ] [เพิ่ม] [รีเซ็ต]
product_price	varchar(4)	utf8_general_ci		ไม่			[ดู] [แก้ไข] [ลบ] [เพิ่ม] [รีเซ็ต]
product_num	varchar(4)	utf8_general_ci		ไม่			[ดู] [แก้ไข] [ลบ] [เพิ่ม] [รีเซ็ต]

การเปิดดูข้อมูลภายในตาราง

การเปิดดูข้อมูลภายในตาราง ทำได้โดยคลิกที่เมนู เปิดดู ซึ่งจะคลิกได้ก็ต่อเมื่อมีข้อมูลภายในตารางแล้วเท่านั้น สำหรับเมนู เปิดดู

MySQL คืนผลลัพธ์ว่างเปล่ากลับมา (null / 0 แถว). (คำค้นใช้เวลา 0.0047 วินาที)

```
SELECT *
FROM `product`
LIMIT 0, 30
```

แสดงระเบียนที่ 0 - 0 (1 ทั้งหมด, คำค้นใช้เวลา 0.0005 วินาที)

```
SELECT *
FROM `product`
LIMIT 0, 30
```

product_id	product_name	product_price	product_num
1	น้ำชาดื่ม	30	50

เราจะพบข้อมูลที่มีอยู่ในตารางของเรา ในที่นี้มีข้อมูลเพียง 1 ตัว

การเปิดดูข้อมูลภายในตารางด้วยคำสั่ง SQL

ในหัวข้อที่ผ่านมา (2) เป็นการเปิดดูข้อมูลโดยดูข้อมูลทั้งหมด ในการใช้งานจริงจะมีการเลือกดูข้อมูลแบบมีเงื่อนไข ซึ่งสามารถเลือกดูได้ในโปรแกรม โดยคลิกที่เมนู SQL จากนั้นจะปรากฏ textarea สำหรับพิมพ์คำสั่ง SQL ลงไป **1** เสร็จเรียบร้อยแล้วคลิกที่ปุ่ม ลงมือเพื่อรันคำสั่ง SQL ดังแสดงในรูป



1. กดที่ปุ่ม SQL เพื่อเข้าสู่หน้าเปิดดูข้อมูลภายในตารางด้วยคำสั่ง SQL
2. พิมพ์คำสั่งเข้าไป `SELECT * FROM `product``
3. กด ลงมือ

การค้นหา

การค้นหาภายในเมนูค้นหา มีความหมายเดียวกับการใช้คำสั่ง SQL สำหรับเปิดดูข้อมูล แต่ในเมนูการค้นหานี้ จะเป็น user interface ให้ผู้ใช้เลือก ทำให้ง่ายต่อผู้ใช้ที่ต้องการความสะดวก และ ง่ายต่อการใช้งาน สำหรับหน้าจอของเมนู การค้นหา แสดงดังรูป

เชอร์โฟเวอร์: localhost ▶ ฐานข้อมูล: store ▶ ตาราง : product

เปิด โครงสร้าง SQL ค้นหา แทรก ส่งออก Import กระบวนการ ลบข้อมูล โยนทิ้ง

เลือกฟิลด์ (อย่างน้อยหนึ่งฟิลด์):

- product_id
- product_name
- product_price
- product_num

 DISTINCT

ระบุเงื่อนไข: 30

ลำดับการแสดงผล:
 น้อยไปมาก
 มากไปน้อย

เพิ่มเงื่อนไขในการค้นหา: @

ลงมือ

หรือ ทำ "คำค้นจากตัวอย่าง" (wildcard: "%")

ฟิลด์	ชนิด	การเรียงลำดับ	Operator	ค่า
product_id	int(3)		=	
product_name	varchar(50)	utf8_general_ci	LIKE	
product_price	varchar(4)	utf8_general_ci	LIKE	
product_num	varchar(4)	utf8_general_ci	LIKE	

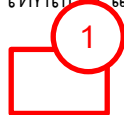
ลงมือ

จากรูป นำมาอธิบายการใช้งานการค้นหา ได้ดังนี้

1. เลือกฟิลด์ที่ต้องการแสดงข้อมูล (เลือกอย่างน้อยหนึ่งฟิลด์)
2. กำหนดจำนวนระเบียนที่ต้องการแสดงใน 1 หน้า
3. กรอกเงื่อนไขในการค้นหา (ถ้ามี)
4. ระบุเงื่อนไขของฟิลด์ต่างๆ

การเพิ่มข้อมูลลงยังตาราง

การเพิ่มข้อมูลลงยังตาราง สามารถทำได้โดยคลิกที่เมนู แทรก โปรแกรมจะแสดงหน้าจอสำหรับเพิ่มข้อมูลดังรูป จากนั้นให้กรอกข้อมูลที่ต้องการลงในคอลัมน์ ค่า (Value) เมื่อกรอกข้อมูลเรียบร้อยแล้ว หากต้องการกลับมาเพิ่มข้อมูลในแถวต่อไปเลย ให้คลิกที่ แทรกระเบียนใหม่ จากนั้นคลิกที่ ลงมือ



เบราว์เซอร์: localhost ▶ ฐานข้อมูล: store ▶ ตาราง: product

เปิด ▶ โครงสร้าง SQL ▶ ค้นหา ▶ แทรก ▶ ส่งออก ▶ Import ▶ กระบวนการ ▶ ลบข้อมูล ▶ โยนทิ้ง

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
product_id	int(3)			<input type="text"/>
product_name	varchar(50)			<input type="text"/>
product_price	varchar(4)			<input type="text"/>
product_num	varchar(4)			<input type="text"/>

ไม่สนใจ

ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
product_id	int(3)			<input type="text"/>
product_name	varchar(50)			<input type="text"/>
product_price	varchar(4)			<input type="text"/>
product_num	varchar(4)			<input type="text"/>

แทรกเป็นแถวใหม่ and then ลงมือ เริ่มใหม่

จากภาพข้างต้น สามารถอธิบายเพิ่มเติมได้ดังนี้

1. กดที่ “แทรก” เพื่อเข้าสู่หน้าเพิ่มข้อมูลลงในตาราง
2. ป้อนค่าข้อมูลต่างๆลงใน ตามฟิลด์ที่เราได้กำหนดไว้ ในกรณีนี้กำหนด product_id เป็นค่า auto_increment ดังนั้นเราไม่ต้องกรอกข้อมูลใดๆ ลงในช่องนี้
3. เราสามารถกรอกข้อมูลในตารางได้ครั้งละ 2 ชุด ดังนั้นหากต้องการกรอกและบันทึกทั้ง 2 ชุด ให้นักเรียนเอาเครื่องหมาย ✓ ออก หากต้องการบันทึกเพียงแค่ชุดด้านบนชุดเดียว ก็คงเครื่องหมาย ✓ ไว้
4. หากกรอกข้อมูลครบและถูกต้องแล้วให้นักเรียนกดที่ “ลงมือ” เพื่อบันทึกข้อมูล

การลบข้อมูลในตาราง

การลบข้อมูลภายในตาราง คลิกที่เมนู เปิดดู โปรแกรมจะแสดงรายการข้อมูลจากนั้นคลิกที่รูป เพื่อลบข้อมูล ดังรูป (สามารถให้คำสั่ง SQL ในการลบข้อมูลได้อีกวิธีหนึ่ง)

เบราว์เซอร์: localhost ▶ ฐานข้อมูล: store ▶ ตาราง: product

เปิด ▶ โครงสร้าง SQL ▶ ค้นหา ▶ แทรก ▶ ส่งออก ▶ Import ▶ กระบวนการ ▶ ลบข้อมูล ▶ โยนทิ้ง

2

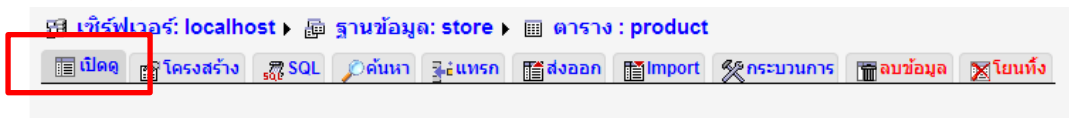
product_id	product_name	product_price	product_num
1	น้ำอัดลม	30	50

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน แนวนอน และข้าหัวแถวทุกๆ 100 เซลล์

การแก้ไขข้อมูลในตาราง

การแก้ไขข้อมูลในตาราง คลิกที่เมนูเปิดดู โปรแกรมจะแสดงรายการข้อมูลจากนั้นคลิกที่รูป เพื่แก้ไขข้อมูล ดังรูป (สามารถให้คำสั่ง SQL ในการลบข้อมูลได้อีกวิธีหนึ่ง) เมื่อคลิกที่รูปแก้ไขแล้ว จะปรากฏหน้าจอสำหรับแก้ไขข้อมูลดังรูป เป็นช่องสำหรับแก้ไข โดยจะแสดงทุกๆ ฟیلด์ภายในตาราง และหน้าจอแก้ไขนี้สามารถ **1** ฝึก หรือแทรกข้อมูลเป็นแถวใหม่ได้ ด้วย



product_id	product_name	product_price	product_num
1	น้ำอัดลม	30	50

แสดง: 30 แถว เริ่มจากแถวที่ 0

อยู่ใน แนวนอน และข้าหัวแถวทุกๆ 100 เซลล์

เมื่อกดที่ปุ่ม แก้ไข แล้วก็จะได้จอดังนี้

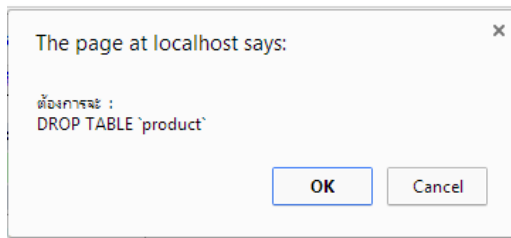
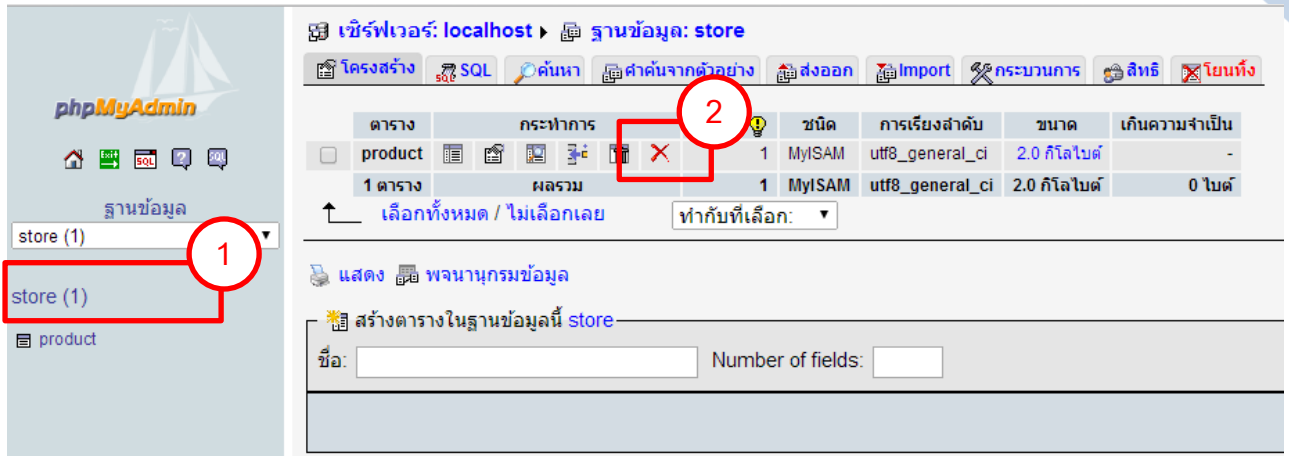
ฟิลด์	ชนิด	ฟังก์ชัน	ว่างเปล่า (null)	ค่า
product_id	int(3)			1
product_name	varchar(50)			น้ำอัดลม
product_price	varchar(4)			30
product_num	varchar(4)			50

บันทึก and then ส่งกลับ

ลงมือ เริ่มใหม่

การลบตาราง

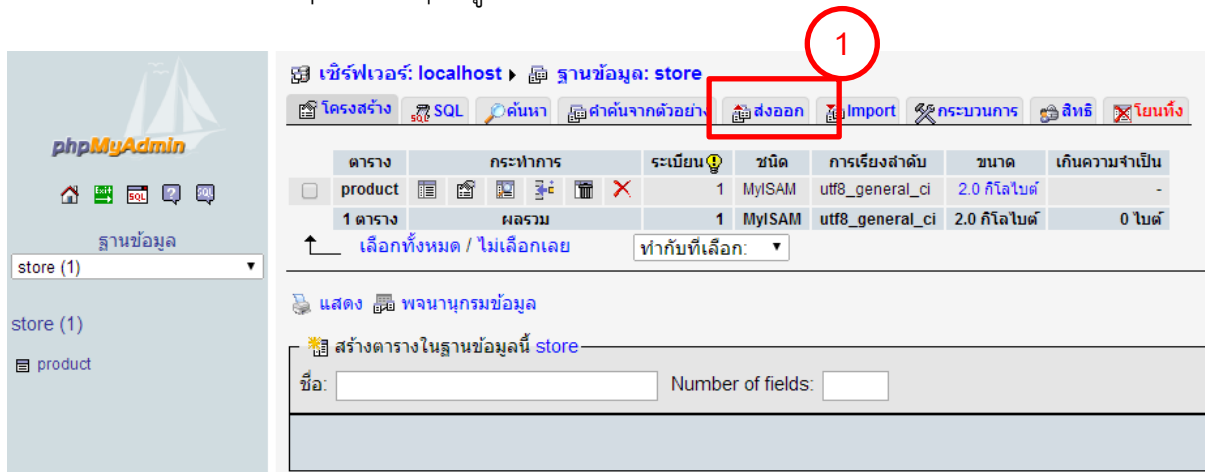
การลบตารางออกจากฐานข้อมูล ทำได้โดยคลิกที่ชื่อฐานข้อมูลทางด้านซ้ายของโปรแกรม เพื่อแสดงตารางทั้งหมดขึ้นมา จากนั้นเลือกคลิกที่รูป เพื่อลบตารางที่ต้องการ แสดงการลบตาราง ดังรูป



จากรูป เป็นหน้าจอยืนยันการลบตาราง ถ้าต้องการลบตารางให้คลิกที่ OK ถ้าไม่ต้องการลบตารางให้คลิกที่ Cancel

การ Export ข้อมูล (ส่งออก)

การ Export ข้อมูล หรือ ฐานข้อมูล ทำได้โดย คลิกที่เมนู ส่งออก จากรูป จากนั้นโปรแกรมจะแสดงหน้าจอสำหรับตั้งค่าการ Export ต่างๆ ดังรูป



The screenshot shows a web-based database export tool. On the left, there's a 'ส่งออก' (Export) section with a dropdown menu set to 'product' and several radio button options for file formats like CSV, Excel, Word, etc., with 'SQL' selected. The main area is 'ตัวเลือก SQL' (SQL Options) with checkboxes for 'Add custom comment into header', 'Enclose export in a transaction', 'Disable foreign key checks', and 'SQL compatibility mode' set to 'NONE'. There are sections for 'โครงสร้าง' (Structure) and 'ข้อมูล' (Data) with various options like 'Add DROP TABLE / DROP VIEW', 'Add IF NOT EXISTS', 'เพิ่มค่า AUTO_INCREMENT', 'ใส่ 'backquote' ให้กับชื่อตารางและฟิลด์', 'Add CREATE PROCEDURE / FUNCTION', 'Maximal length of created query' set to 50000, and 'Export type' set to 'INSERT'. At the bottom, there's a 'ส่งมาเป็นไฟล์' checkbox (annotated with 1), a 'รูปแบบของชื่อไฟล์' field with 'DB' (annotated with 2), and a 'ลงมือ' button (annotated with 3).

จากภาพสามารถอธิบายได้ดังนี้

1. ให้ทำเครื่องหมาย ✓ ส่งมาเป็นไฟล์
2. ตั้งชื่อไฟล์ข้อมูล
3. เมื่อทำทุกขั้นตอนเสร็จแล้ว กด ลงมือ เพื่อ Export ฐานข้อมูลออกมา

จากรูป เป็นหน้าจอสำหรับการตั้งค่าต่างๆ ของการ Export ซึ่งประกอบไปด้วย ส่วนต่างๆ ดังนี้

- ส่งออก ใช้สำหรับเลือกชนิดของข้อมูลที่จะส่งออก เช่น SQL, XML เป็นต้น
- โครงสร้าง

- o เพิ่มคำสั่ง drop table ใช้ในกรณีที่ต้องการให้ลบตารางเก่าก่อนสร้างตารางใหม่ ในขณะที่

นำไฟล์มา Import

- o Add AUTO_INCREMENT value ใช้เพิ่มข้อมูลประเภท auto_increment

ในขณะที่สร้างตาราง ในกรณีที่นำไฟล์มา Import

- o ใส่ backquote ให้กับชื่อตารางและฟิลด์

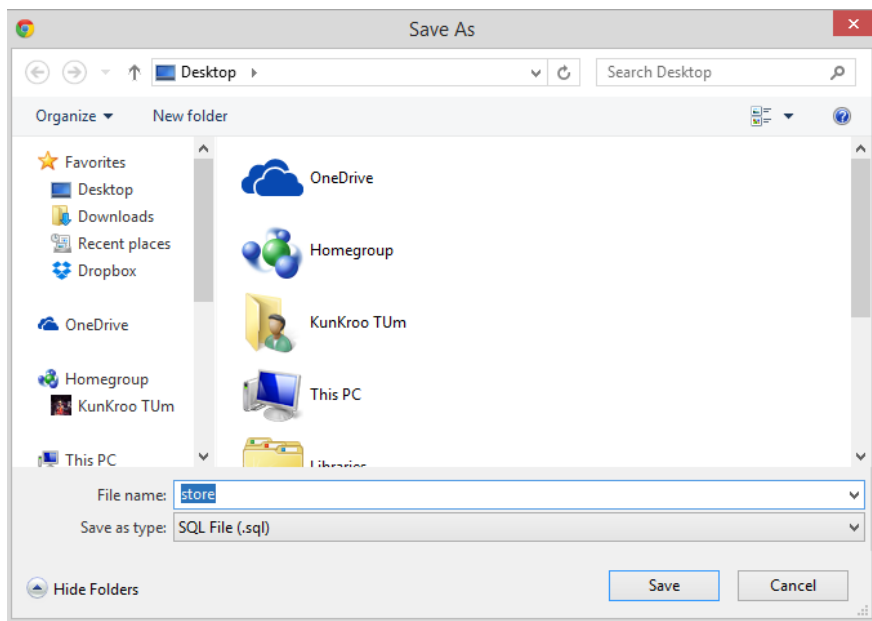
- ข้อมูล

- o คำสั่ง INSERT เต็มรูปแบบ
- o แทรกหลายระเบียนในคราวเดียว

o Export Type เป็นการเลือกเฉพาะชนิดของคำสั่งที่ต้องการ ได้แก่ INSERT UPDATE และ REPLACE

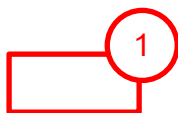
- ส่งมาเป็นไฟล์ ให้คลิกที่ checkbox ด้านหน้าด้วย
- ใส่รูปแบบของชื่อไฟล์ เป็นการใส่ชื่อไฟล์ที่ต้องการ

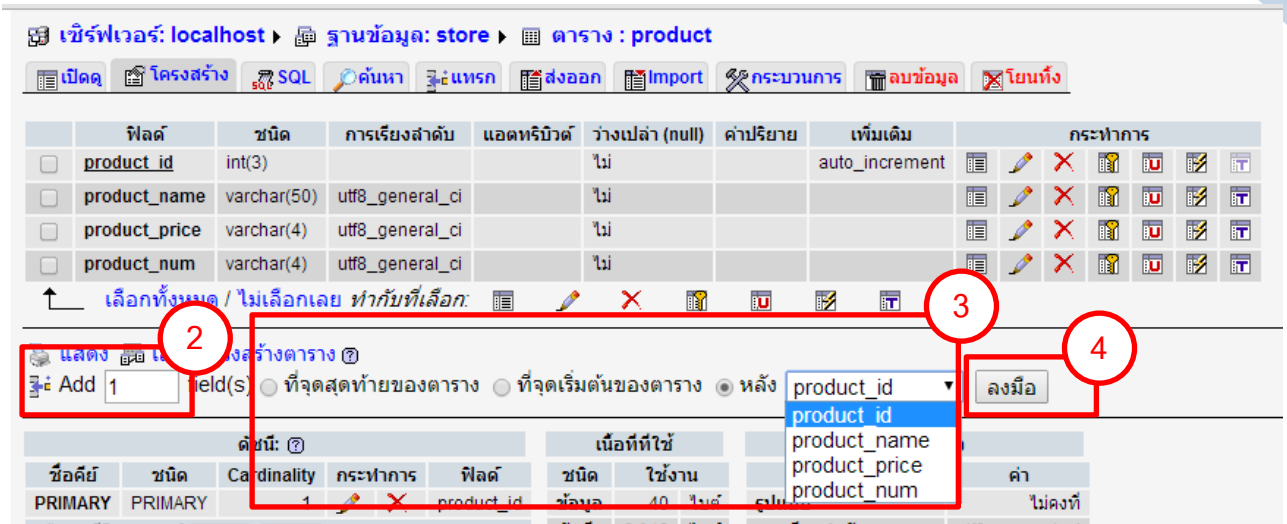
เมื่อกำหนดค่าการ Export ต่างๆ เรียบร้อยแล้ว ให้คลิกที่ ลงมือ จะปรากฏหน้าต่างสำหรับเลือกว่าจะบันทึกไฟล์ หรือเปิดไฟล์ ให้คลิกที่ SAVE ดังรูปที่ 20 หลังจากนั้นเลือก location ภายในเครื่องเพื่อแจ้งให้โปรแกรมทราบว่า จะเซฟไฟล์ไว้ที่ใด เมื่อเรียบร้อยแล้วคลิกที่ SAVE เพื่อบันทึกไฟล์ข้อมูลลงเครื่องดังรูป



การเพิ่มฟิลด์ใหม่ลงไปในตารางที่มีอยู่แล้ว

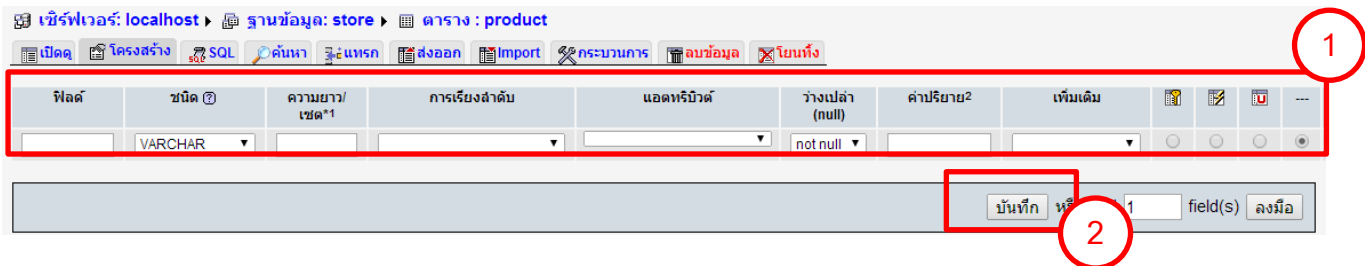
การเพิ่มฟิลด์ใหม่ลงในตารางที่มีอยู่แล้ว จะดำเนินการผ่านเมนู โครงสร้าง ขึ้นแรกให้คลิกที่เมนู โครงสร้าง จะพบส่วนของการเพิ่มฟิลด์ใหม่ ซึ่งต้องระบุจำนวนฟิลด์ที่ต้องการเพิ่ม และ ระบุว่า จะเพิ่มฟิลด์ใหม่ไว้หน้าหรือหลังฟิลด์ใดในตาราง การเพิ่มฟิลด์ใหม่แสดงดังรูป





1. คลิกที่โครงสร้าง
2. แล้วกรอกจำนวนฟิลด์ที่ต้องการ
3. เลือกตำแหน่งที่ต้องการแทรก แล้ว
4. กดลงมือ

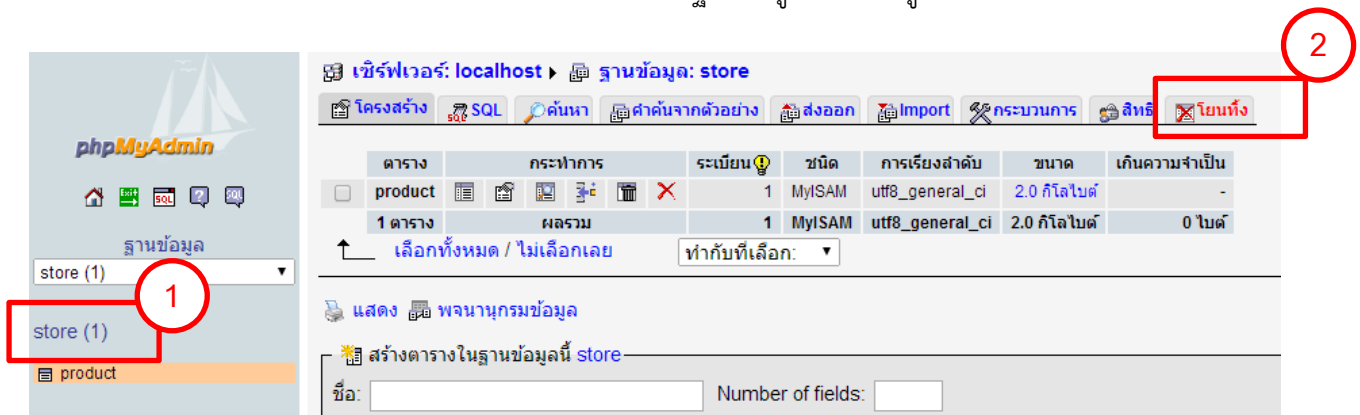
จะพบกับหน้าต่างดังภาพ



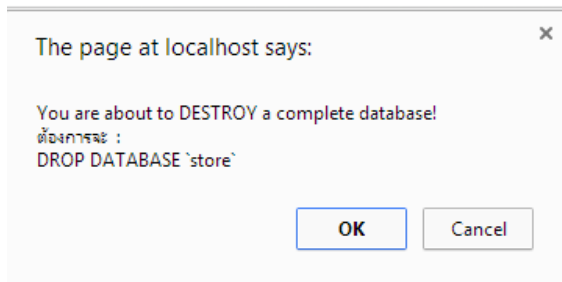
จากรูป เป็นหน้าจอสำหรับใส่รายละเอียดของฟิลด์ใหม่ โดยผู้ใช้ต้องกรอกชื่อฟิลด์ชนิดของข้อมูลที่จะเก็บ ความยาว และค่าอื่นๆ เมื่อกรอกข้อมูลเรียบร้อยแล้วคลิกที่ บันทึก เพื่อเพิ่ม ฟิลด์ใหม่ลงยังตาราง

การลบฐานข้อมูล (Drop Database)

การลบฐานข้อมูลทำได้โดยคลิกเลือกที่ชื่อฐานข้อมูลจากเมนูทางด้านซ้ายของหน้าจอ ก่อน จากนั้นคลิกเลือกที่ โยนทิ้ง โปรแกรมจะแสดงข้อความยืนยันก่อนลบฐานข้อมูล แสดงดังรูป



1. ให้นักเรียนไปคลิกที่ชื่อฐานข้อมูล ในที่นี้คือ store
2. แล้วให้นักเรียนกดที่ โยนทิ้ง จะพบกับหน้าต่างดังรูปด้านล่าง



จากรูป เป็นหน้าจอยืนยันการลบฐานข้อมูล ถ้าต้องการลบฐานข้อมูลให้คลิกที่ OK ถ้าไม่ต้องการลบฐานข้อมูลให้คลิกที่ Cancel



หนังสือคู่มือการพัฒนาโปรแกรมบนเว็บ
Web Application