

2.2.2 ภาษา PHP

1. ภาษา PHP คืออะไร

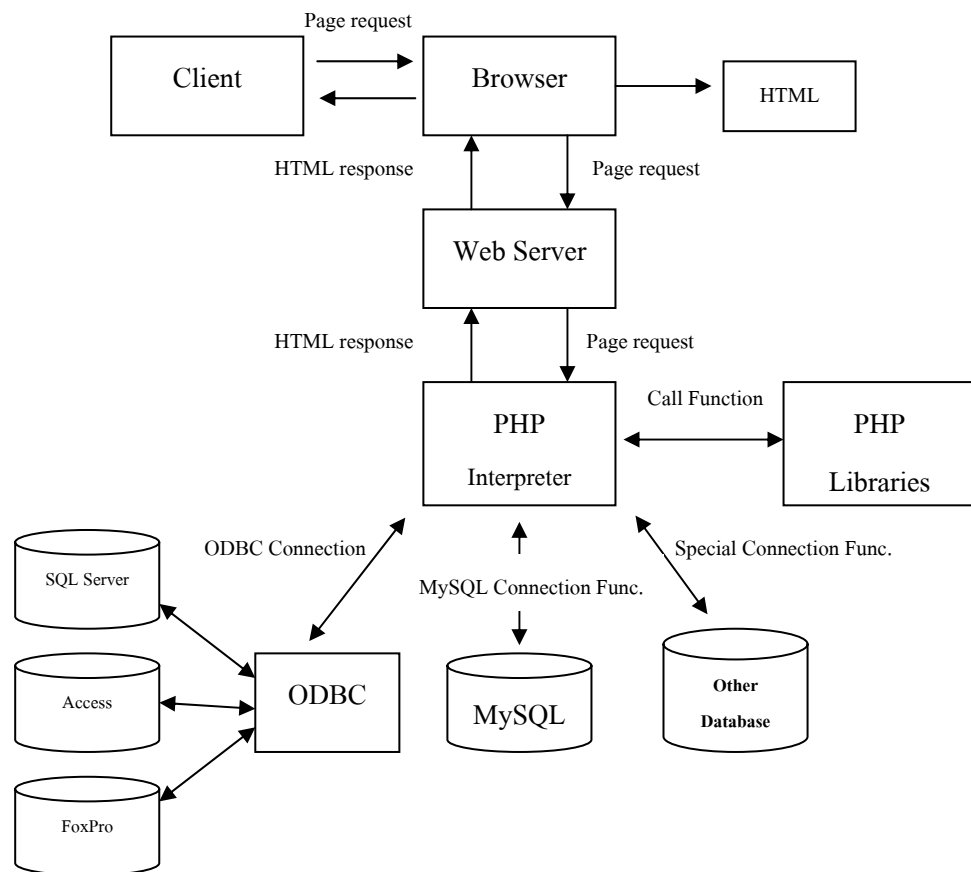
เดิมทีนั้น PHP เป็นชื่อย่อของภาษาโปรแกรมมิ่งชนิดหนึ่งที่มีชื่อว่า “Professional Home Pages” แต่ในปัจจุบันภาษาชนิดนี้ถูกพัฒนาต่อมาจนกลายเป็นภาษาโปรแกรมมิ่งชนิดใหม่ซึ่งมีชื่อว่า “Personal Hypertext Processor; PHP” ภาษาชนิดใหม่นี้เป็นที่นิยมในการนำมาใช้เขียนสคริปต์ (ชุดคำสั่งควบคุมการทำงานของโปรแกรม ซึ่งมีความยาวไม่มากนักและสามารถทำงานได้ดีกับเว็บไซต์เป็นอย่างมาก) PHP เป็นภาษาสคริปต์ที่เป็น Server Side Script และเป็น Open Source ที่ผู้ใช้ทั่วไปสามารถ Download Source Code ได้ฟรี จุดประสงค์ที่สำคัญของภาษา PHP คือการช่วยให้นักพัฒนาเว็บเพจสามารถเขียนเว็บเพจที่เป็นแบบไดนามิกได้อย่างรวดเร็ว ภาษา PHP จะทำงานร่วมกันกับเอกสาร HTML โดยการสร้างโค้ดแทรกระหว่าง Tag HTML และสร้างเป็นไฟล์ที่มีนามสกุล .php .php3 หรือ php4 ไวยากรณ์ที่ใช้ใน PHP เป็นการนำรูปแบบของภาษาต่าง ๆ มารวมกัน ได้แก่ C Perl และ Java ทำให้ผู้ใช้ที่มีพื้นฐานของภาษาเหล่านี้สามารถใช้งาน PHP ได้ไม่ยาก

เนื่องจาก PHP จะทำงานโดยมีตัวแปลและเอ็กซิกิวต์ที่ฝั่งเซิร์ฟเวอร์ อาจจะเรียกการทำงานว่าเป็นเซิร์ฟเวอร์ไซด์ (Server Side) ส่วนการทำงานของบราวเซอร์ของผู้ใช้เรียกว่าไคลเอนต์ไซด์ (Client Side) โดยการทำงานจะเริ่มตัวที่ผู้ใช้ส่งความต้องการผ่านเว็บเบราว์เซอร์ทาง HTTP (HTTP Request) ซึ่งอาจเป็นการกรอกแบบฟอร์ม หรือใส่ข้อมูลที่ต้องการ หรือแสดงคู่มือ โดยเรียกเอกสาร PHP (เอกสารนี้จะมีส่วนขยายเป็น php) เช่น test.php เมื่อเอกสาร PHP เข้ามาถึงเว็บเซิร์ฟเวอร์ก็จะถูกส่งต่อไปให้ PHP Interpreter เพื่อทำหน้าที่แปลคำสั่งแล้วเอ็กซิกิวต์คำสั่งตามบรรทัดที่ระบุคำสั่งนั้น ๆ จากนั้น PHP จะสร้างผลลัพธ์ในรูปแบบเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปให้บราวเซอร์แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP Response) ดังรูป 2.4 ตามกระบวนการดังนี้

1. จากไคลเอนต์ จะเรียกไฟล์ php script ผ่านทางโปรแกรมบราวเซอร์ (Internet Explore)
2. บราวเซอร์จะส่งคำร้อง (Request) ไปยังเว็บเซิร์ฟเวอร์ ผ่านทางเครือข่ายอินเทอร์เน็ต
3. เมื่อเว็บเซิร์ฟเวอร์รับคำร้องขอจากบราวเซอร์แล้ว ก็จะนำสคริปต์ php ที่เก็บอยู่ในเซิร์ฟเวอร์มาประมวลผลด้วยโปรแกรมแปลภาษา PHP ที่เป็น อินเตอร์พรีเตอร์
4. กรณีที่ php script มีการเรียกใช้ข้อมูลก็จะติดต่อกับฐานข้อมูลต่าง ๆ ผ่านทาง ODBC Connection ถ้าเป็น ฐานข้อมูลกลุ่ม Microsoft SQL Server, Microsoft Access, FoxPro หรือใช้ Function Connection ที่มีอยู่ใน PHP Library ในการเชื่อมต่อฐานข้อมูลเพื่อ

ดึงข้อมูลออกมา หลังจากแปลสคริปต์ PHP เสร็จแล้วจะได้รับไฟล์ HTML ใหม่ที่มีแต่แท็ก HTML ไปยัง Web Service

5. Web Service ส่งไฟล์ HTML ที่ได้ผ่านการแปลแล้วกลับไปยังบราวเซอร์ที่ร้องขอผ่านทางเครือข่ายอินเทอร์เน็ต
6. บราวเซอร์รับไฟล์ HTML ที่เว็บเซิร์ฟเวอร์ส่งมาให้แปล HTML แสดงผลออกมาทางจอภาพเป็นเว็บเพจ โดยใช้ตัวแปลภาษา HTML ที่อยู่ในบราวเซอร์ ซึ่งเป็นอินเทอร์เน็ตพริเตอร์เช่นเดียวกัน



รูปที่ 2.4 แสดงขั้นตอนการทำงาน PHP Script Request/Response

ตัวอย่างโค้ดที่ใช้ใน PHP

```

1. <html>
2. <head>
3. <title> ตัวอย่าง </title>
4. </head>
5. <body>
6. <?php
7. echo “ยินดีต้อนรับสู่ PHP Script”;
8. ?>
9. </body>
10. </html>

```

รูปที่ 2.5 แสดง html tag และ PHP Script

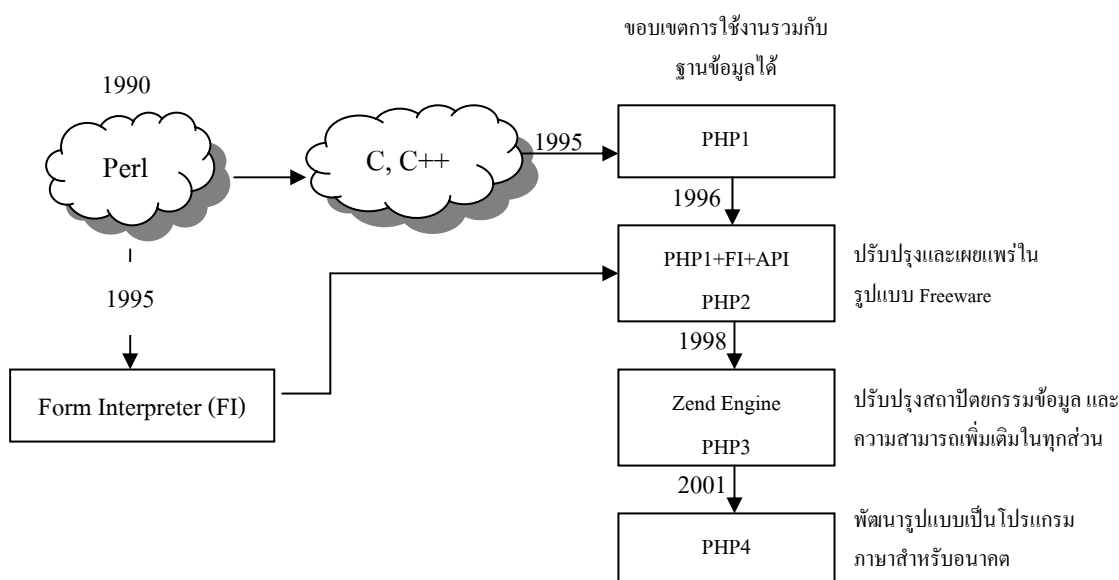
จากรูปที่ 2.5 บรรทัดที่ 6 ถึง 9 เป็นส่วนของ Script PHP ซึ่งเริ่มต้นด้วย <?php คำสั่งเรียกใช้ฟังก์ชัน หรือข้อความ ?> สำหรับ Script นี้เป็นการแสดงข้อความว่า “ยินดีต้อนรับสู่ PHP Script” โดยใช้คำสั่ง echo “ยินดีต้อนรับสู่ PHP Script”; ซึ่งจะแสดงผลดังรูปที่ 2.6



รูปที่ 2.6 แสดงผลการทำงานของโค้ดผ่านบราวเซอร์

2. วิวัฒนาการของภาษา PHP

พัฒนาการของภาษาโปรแกรมมิ่งสำหรับสร้างเว็บไซต์ เริ่มต้นขึ้นในช่วงต้นของทศวรรษที่ 1980 ในสมัยก่อนข้อมูลที่ส่งผ่านระบบเครือข่าย และอินเทอร์เน็ต เป็นข้อมูลตัวอักษรเพียงอย่างเดียวจนกระทั่งมีการคิดค้น และพัฒนารูปแบบภาษาโปรแกรมมิ่งสำหรับเว็บไซต์ในช่วงปี ค.ศ. 1981-1987 โดยบริษัท Microsoft ภาษาโปรแกรมมิ่งสำหรับสร้างเว็บไซต์จึงเกิดการพัฒนาอย่างรวดเร็ว ในปี ค.ศ. 1990 นับได้ว่าเป็นช่วงเวลาในการเปิดยุคสมัยแห่งการสร้างเว็บไซต์ ภาษาโปรแกรมมิ่งสำหรับสร้างเว็บไซต์ถูกพัฒนาขึ้นเป็นจำนวนมากนับจาก HTML, CGI, Perl, ASP จนถึง PHP สามารถแสดงพัฒนาการของภาษา PHP เป็นดังรูปที่ 2.7



รูปที่ 2.7 แสดงความเป็นมาของ PHP

จากรูป เป็นลำดับการพัฒนาการของภาษา PHP ซึ่งรายละเอียดของการพัฒนามีดังนี้

- ปี ค.ศ. 1990 ภาษาโปรแกรมมิ่ง ซึ่งมีความสามารถในการทำงานร่วมกันกับเว็บไซต์ที่มีชื่อว่า Perl ถูกพัฒนาขึ้น โดยมีจุดมุ่งหมายเพื่อใช้พัฒนาเว็บเพจให้สามารถทำงานได้ในสถานะที่แตกต่างกัน โดยสามารถนำไปใช้งานได้กับเว็บเซิร์ฟเวอร์ บนระบบปฏิบัติการที่ต่างกัน รวมทั้งสามารถทำความเข้าใจและเรียนรู้ได้ง่ายอีกด้วย
- ปี ค.ศ. 1995 นาย Rasmus Lerdorf พัฒนาภาษา PHP Script ของ Perl โดยมีวัตถุประสงค์เพื่อใช้สำหรับผู้ที่ต้องการเรียกดูประวัติการทำงานของเขา ซึ่งต่อมา

- Lerdorf นำภาษาที่พัฒนาขึ้นมาเขียนในรูปแบบของภาษา C และพัฒนาขึ้นจนเกิดเป็น PHP 1 ขึ้นมา ในช่วงเวลาเดียวกัน Lerdorf ยังพัฒนาเครื่องมือที่ใช้แปลภาษาสคริปต์นี้เรียกว่า “FI (Form Interpreter)” จากนั้นในกลางปี ค.ศ. 1996 Lerdorf ได้รวมการทำงานของ PHP และ FI เข้าด้วยกันและพัฒนาขึ้นจนกลายเป็น PHP 2
- ปี ค.ศ. 1997 ได้มีกลุ่มของนักพัฒนาเข้าร่วมโครงการและปรับปรุง PHP 2 โดยเพิ่มความสามารถในการถ่ายทอดข้อมูลแบบ API (Application Programming Interface) รวมทั้งปรับปรุงเครื่องมือต่าง ๆ และสร้างเป็น Zend engine ซึ่งเรียกว่า “PHP 3” ขึ้นมาและเนื่องจาก Zend engine ได้รับความนิยมเป็นอย่างมากนักพัฒนาจึงได้ปรับปรุงฟังก์ชันการทำงานมากขึ้นจนกลายเป็น PHP 4 ซึ่งเป็นรุ่นล่าสุดของ PHP
 - ในรุ่นล่าสุดของ PHP คือ PHP 5 ซึ่งยังได้มีการพัฒนาฐานข้อมูลที่สามารถนำมาใช้กับ PHP 5 ได้ฟรีอีกด้วย นั่นคือ MySQL 5 ซึ่งในปัจจุบันนี้มีผู้เลือกใช้งาน PHP 5 เพิ่มขึ้นอย่างมากมา โดยผู้ใช้เหล่านี้มีจุดประสงค์ในการพัฒนาเว็บเพจแบบไดนามิก และการบริการ Web Hosting ในปัจจุบันก็สนับสนุนการทำงานด้วย PHP อีกด้วย

3. ความสามารถของภาษา PHP

ภาษา PHP เป็นภาษาที่พัฒนาขึ้นจากพื้นฐานของภาษาโปรแกรมมิ่งชนิดอื่น ๆ เช่น C, C++ และ Perl ทำให้มีลักษณะเด่นของภาษาดั้งเดิมของแต่ละชนิดรวมกันอยู่ ความสามารถของภาษา PHP ที่เห็นได้อย่างเด่นชัด สามารถจำแนกออกได้ดังนี้

- เป็นภาษาที่ทำความเข้าใจและใช้งานง่ายไม่เหมือนกับ JAVA หรือ C++ และมีส่วนที่สนับสนุนการทำงานได้กับทุกเว็บไซต์
- เป็น Open Source ผู้ใช้สามารถดาวน์โหลด และนำ source code ของ PHP ไปใช้ได้โดยไม่เสียค่าใช้จ่าย
- เป็นสคริปต์แบบเซิร์ฟเวอร์ไซด์ ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ ไม่ส่งผลกระทบต่อการทำงานของเครื่องไคลเอนต์ โดย PHP จะอ่านโค้ด และทำงานที่เซิร์ฟเวอร์จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของเอกสาร HTML ซึ่งอ่านโค้ดของ PHP ผู้ใช้ไม่สามารถมองเห็นได้
- PHP สามารถทำงานได้ในระบบปฏิบัติการที่ต่างชนิดกัน เช่น Unix, Windows, Mac, OS หรือ Risc OS อย่างดี ประสิทธิภาพเนื่องมาจาก PHP เป็นสคริปต์ที่ต้องทำงานบนเซิร์ฟเวอร์ ดังนั้นคอมพิวเตอร์ที่ใช้สำหรับเรียกคำสั่ง PHP จึงจำเป็นต้อง

ติดตั้งโปรแกรมประเภทเว็บเซิร์ฟเวอร์ไว้ด้วยเพื่อให้สามารถประมวลผล PHP ได้ ซึ่งเป็นเหตุผลที่ทำให้ PHP สามารถทำงานได้กับหลายระบบปฏิบัติการหลายชนิด

- PHP สามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Personal Web Server (PWS), Apache, OmniHttpd, Microsoft Internet Information Server (IIS) เป็นต้น
- สนับสนุนการเขียนสคริปต์ที่ใช้หลักของ Object Orientation
- PHP สามารถสร้างเว็บไซต์ที่บรรจุข้อมูลรูปแบบต่าง ๆ ลงในเว็บ เช่น รูปภาพ ไฟล์ PDF หรือ Flash Movie เป็นต้น
- คุณสมบัติที่สำคัญอีกประการหนึ่งของ PHP คือความสามารถในการทำงานร่วมกับระบบจัดการฐานข้อมูลที่หลากหลายซึ่งระบบการจัดการฐานข้อมูลที่สนับสนุนการทำงานของ PHP มีตัวอย่างดังนี้
 - 1) ชนิด ORACLE เช่น Oracle (OC17 and OC18), AdabasD, Ingres, FilePro (read-only) และ Solid เป็นต้น
 - 2) ชนิด Access เช่น dBase, InterBase, Ovrimos Empress และ FrontBase เป็นต้น
 - 3) ชนิด SQL เช่น MS SQL, PostgreSQL, mSQL และ MySQL เป็นต้น
- PHP อนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่านโปรโตคอล (Protocol) ชนิดต่าง ๆ ได้ เช่น LDAP, IMAP, SNMP, NNTP, POP3, HTTP และ COM (สำหรับ Windows) เป็นต้น
- ผู้ใช้สามารถเขียนโค้ด PHP และอ่านข้อมูลในรูปแบบของ Extensible Markup Language (XML) ได้

4. ไวยากรณ์ของภาษา PHP

เมื่อผู้ใช้ทราบถึงความสามารถของภาษา PHP และเลือกที่จะใช้ภาษา PHP เป็นเครื่องมือช่วยในการควบคุมการทำงานระหว่างเว็บเพจและฐานข้อมูล การทำความเข้าใจอย่างมาก เนื้อหาในส่วนนี้จะกล่าวถึงการใช้คำสั่ง และไวยากรณ์ของภาษา PHP ซึ่งมีรายละเอียดดังนี้

4.1 การ Comments

Comments คือข้อความที่เขียนไว้ในโปรแกรมเพื่ออธิบายโค้ด โดยใส่ไว้ในโปรแกรมที่ต้องการอธิบาย ซึ่งจะไม่มีผลกับการทำงานของโปรแกรมในระหว่างโปรแกรมทำงาน การใส่ Comments โดยทั่วไปจะมี 2 แบบ คือ การอธิบายแบบที่ละบรรทัดและการอธิบายแบบหลาย ๆ บรรทัด สำหรับการอธิบายที่ละบรรทัดนั้นใช้เครื่องหมาย // หรือ # โดยใส่ไว้ข้างหลังโค้ดที่ต้องการ

อธิบาย เมื่อขึ้นบรรทัดใหม่ Comments นั้นก็จะไม่มีผล ถ้าต้องการอธิบายก็ได้ Comment ใหม่ สำหรับการใส่ Comments อีกรูปแบบนั้นเป็นการใส่คอมม่อข้อความทั้งหมดที่ต้องการอธิบายในครั้งเดียว ซึ่งจะใช้เครื่องหมาย /* ข้อความที่ต้องการอธิบาย /*

4.2 ตัวแปรใน Script PHP

การตั้งชื่อตัวแปรของ PHP นั้นสามารถใช้ตัวเล็กหรือตัวใหญ่ก็ได้ นอกจากนี้ยังสามารถใช้เครื่องหมาย Underscore (_) ได้เช่นกัน แต่ตัวอักษรตัวแรกของชื่อห้ามเป็นตัวเลข โดยหน้าตัวแปรที่กำหนดจะต้องมีเครื่องหมาย \$ นำหน้า เช่น

```
<?
$ Name= “นิโคล”;
Echo “($Name);
?>
```

จากตัวอย่างเป็นการกำหนดตัวแปร Name ขึ้นมาโดยใส่ค่า String เข้าไปคือ “นิโคล” และทำการแสดงผลโดยใช้คำสั่ง echo

ชนิดข้อมูล (Data Type) ใน PHP นั้นมีอยู่ด้วยกัน 5 แบบ คือ

- Integer เป็นการกำหนดตัวแปรแบบตัวเลข ซึ่งเป็นเลขจำนวนเต็ม เช่น 100 เป็นต้น
- Floating เป็นการกำหนดเลขซึ่งเป็นจำนวนทศนิยม เช่น 100.25 เป็นต้น
- String เป็นการกำหนดตัวแปรชนิดอักขระ เช่น “บริษัท KTP Comp and consult” เป็นต้น
- Array เป็นการกำหนดตัวแปรชนิดอาเรย์ ซึ่งเป็นลักษณะของชุดข้อมูล เช่น

```
$province= array (“Bangkok” , Thonburi”, “Trad”);
```

\$province(0) \$province (1) \$province (2)

- Object และ Class เป็นการกำหนดตัวแปรให้เป็น Object ไว้เก็บคุณสมบัติ และหน้าที่ต่าง ๆ ซึ่งถูกกำหนดโดย Class

4.3 การใช้อักขระพิเศษ

อักขระพิเศษใช้สำหรับช่วยให้การแสดงผลเป็นไปอย่างสมบูรณ์ตามที่ผู้ต้องการ เช่น เมื่อต้องการขึ้นบรรทัดใหม่จะใช้ คำสั่ง \n ดังตัวอย่างดังนี้

```
echo "Hello my friend, \n John";
```

สำหรับตัวอย่างนี้การแสดงผลจะเป็นดังนี้

```
Hello my friend, John
```

ตารางแสดงความหมายของอักขระพิเศษ ต่าง ๆ ซึ่ง PHP Interpreter แปลคำสั่งจะพิมพ์อักขระพิเศษนี้ไปยังเว็บเบราว์เซอร์

ตารางที่ 2.1 แสดงความหมายของอักขระ

อักขระพิเศษ	ความหมาย
\r	Carriage ขึ้นบรรทัดใหม่
\t	Horizontal tab เลื่อนไปทางขวา 1 ช่องตัวอักษร
\n	New line ขึ้นบรรทัดใหม่
\"	Double quote แสดงเครื่องหมาย "
\\$	Dollar Sign แสดงเครื่องหมาย \$
\\	Backslash แสดงเครื่องหมาย \

4.4 Class และ Object

Class ถูกสร้างขึ้นแบบ Abstract (กำหนดหน้าที่การทำงาน และคุณสมบัติต่าง ๆ ไว้ แต่ไม่สามารถนำไปใช้งานได้ โดยตรงต้องสร้างตัวแทนขึ้นมาก่อน) เป็นที่เก็บของคุณสมบัติ (Attribute) และหน้าที่ (Property) โดยการเรียกใช้คุณสมบัติของ Class นี้จะต้องสร้าง Object ของ Class ขึ้นมาก่อน ซึ่ง Object ก็คือตัวแทนของ Class นั้นเอง โดยใน Class เพียงตัวเดียวก็สามารถสร้าง Object ขึ้นมาใช้ก็ได้ ซึ่งวิธีการนี้เป็นการทำงานในแบบของ Object Orientation นั้นเอง ดังตัวอย่างดังต่อไปนี้


```

1.      Class CreateFunction {
2.      function a() {
3.      echo “สร้างฟังก์ชัน a”;
4.      }
5.      function b() {
6.      echo “สร้างฟังก์ชัน b”;
7.      }
8.      }\\Class CreateFunction

```

Class นี้ทำการสร้างฟังก์ชันขึ้นมา 2 ฟังก์ชัน ชื่อว่า a และ b โดยแต่ละ class จะมีหน้าที่แสดงผล โดยการพิมพ์ข้อความออกทางหน้าจอ วิธีการเรียกใช้ Class เหล่านี้อาจใช้คำสั่งได้ดังนี้

```
$instance = new CreateFunction();
```

คำสั่งข้างต้นเป็นการเรียกใช้ Class ที่สร้างขึ้นผ่าน Object ชื่อ instance ซึ่ง Object นี้จะมีค่าเท่ากับ Class ที่สร้างไว้ ตัวแปร \$instance ถูกสร้างขึ้นมาเพื่อใช้เป็น Object ให้อ้างอิงถึง Class CreateFunction โดยถ้าต้องการอ้างอิง เช่น Function a() สามารถทำได้โดยใช้ “->” ดังนี้

```
$instance -> a();
```

4.5 Strength และ Casting

Type Strength คือการกำหนดตัวแปรของ PHP โดยไม่ระบุชนิดของตัวแปรนั้นว่า เป็นตัวแปรชนิดใด ใช้กรณีที่ตัวแปรมีชนิดของข้อมูลไม่แน่นอน โดยปกติ Strength จะใช้กับตัวแปรซึ่งรับค่าจากการคำนวณตัวเลข เช่น

```
$ Total = 100*50; // PHP
```

โดยที่จะกำหนดให้ตัวแปร Total เป็นตัวแปรชนิด Integer ให้โดยอัตโนมัติ

ส่วนกรณีที่ต้องการกำหนดชนิดให้กับตัวแปรหลังจากที่ใช้ตัวแปรไปแล้ว หรือเมื่อมีการกระทำบางอย่าง แล้วต้องการเปลี่ยนชนิดของตัวแปรเพื่อนำไปใช้ในกรณีที่ต้องการ โดยเรียกตัวแปรชนิดนี้ว่า “Type Casting” ตัวอย่างเช่น ต้องการใช้ตัวแปรเดิมซึ่งมีชนิดเป็น Float นั้นไปใช้คำนวณร่วมกับตัวแปรชนิดอื่น เช่น Integer วิธีการมีดัง ตัวอย่างนี้

```

1.    <?
2.    $x =100.46;// Float Type } Strength
3.    $y =500; // Integer Type }
4.    $z =$x+$y; //ผลลัพธ์จะได้ตัวแปร $z เป็นตัวแปร $z เป็นตัวแปรชนิด
      Float
5.    $a = (Integer) $z; } Casting
6.    $b = $a/20;
7.    echo ($b)
8.    ?>

```

จากตัวอย่างตัวแปร z เป็นผลลัพธ์ที่ได้จากการบวกกันของตัวแปร x (มีชนิดเป็น Float) และตัวแปร y (มีชนิดเป็น Integer) จากนั้นกำหนดชนิดของตัวแปร z เป็น integer โดยให้ตัวแปร a มีค่าเท่ากับ z และให้ตัวแปร a หารด้วย 20 ผลลัพธ์ที่ได้จากการเรียกใช้โค้ดเหล่านี้ผ่านเว็บเบราว์เซอร์ จะมีลักษณะดังนี้

30

การกำหนดค่าคงที่ การกำหนดค่าคงที่ใน Script ของ PHP นั้นสามารถกำหนดได้ 2 แบบ คือ

- การกำหนดแบบค่าคงที่มาตรฐาน (**Built-in Constants**) เป็นการกำหนดค่าคงที่ที่มีมาอยู่แล้วของ PHP ดังแสดงในตาราง ต่อไปนี้

ตารางที่ 2.2 แสดงค่าคงที่มาตรฐาน

ค่าคงที่มาตรฐาน	หน้าที่
1. <code>_File_</code>	ใช้สำหรับเก็บชื่อไฟล์ของสคริปต์
2. <code>_Line_</code>	เก็บข้อมูลของเลขบรรทัดขณะที่อ้างถึง
3. <code>False</code>	เก็บค่า Boolean เท็จ
4. <code>True</code>	เก็บค่า Boolean จริง
5. <code>PHP_VERSION_</code>	เก็บค่าที่เป็นรุ่นของ PHP
6. <code>PHP_OS</code>	เก็บค่าชื่อของระบบปฏิบัติการที่ใช้

- การกำหนดค่าคงที่แบบกำหนดเอง (**Defining Constants**) เป็นการกำหนดตัวแปรขึ้นมาเองของผู้เขียนโปรแกรม ซึ่งการกำหนดตัวแปรนี้มีประโยชน์ คือ ทำให้สามารถนำค่ามาใช้ได้ตลอดทั้งโปรแกรม ซึ่งเมื่อต้องการเปลี่ยนแปลงค่าก็แก้ไขเพียงจุดที่ต้องการเปลี่ยนค่าโดยไม่ต้องตามไปแก้ไขหลาย ๆ จุด การกำหนดค่าคงที่แบบนี้จะใช้คำสั่ง Define โดยมีรูปแบบดังนี้

Define (ตัวแปร, ค่าคงที่)

ตัวอย่างเช่น Define (Fix, 2154); เป็นการกำหนดตัวแปร Fix มีค่าคงที่เป็น Integer ซึ่งมีค่าเท่ากับ 2154 เป็นต้น

4.6 Operators

Operator เป็นเครื่องหมายในการคำนวณ หรือเปรียบเทียบสามารถจำแนกประเภทของ Operator ได้ดังนี้

- **โอเปอเรเตอร์เชิงคณิตศาสตร์ (Arithmetic Operators)** เป็นเครื่องหมายที่ใช้คำนวณทางคณิตศาสตร์ประกอบด้วยเครื่องหมาย ดังตาราง

ตารางที่ 2.3 โอเปอเรเตอร์เชิงคณิตศาสตร์ (Arithmetic Operators)

เครื่องหมาย	ความหมาย	ตัวอย่าง
+	Addition แทนการบวก	$\$x+\y
-	Subtraction แทนการลบ	$\$x-\y
*	Multiplication แทนการคูณ	$\$x*\y
/	Division แทนการหาร	$\$x/\y
%	Modules แทนการหารเอาเศษ	$\$x\%\y

การระบุจำนวนลบสามารถนำเครื่องหมาย - ใส่หน้าจำนวนที่ต้องการได้ทันที เช่น -1 เป็นต้น

- **โอเปอเรเตอร์เชิงเปรียบเทียบ (Comparison Operators)** การเปรียบเทียบค่า 2 ค่าของค่าใด ๆ ผลที่ได้จะเป็นค่าจริง (True) หรือเท็จ (False) สำหรับ PHP ในเวอร์ชัน 4 มี Operator ที่เพิ่มขึ้นมาใหม่ คือ Identical Operator (==) ซึ่งการเปรียบเทียบของ 2 ค่าใด ๆ นอกจากที่กำหนดไว้โดย == ว่าถ้า 2 ค่ามีค่าเท่ากันจะได้ผลเป็นจริง (True) ถ้าต่างกันค่าใดค่าหนึ่งเป็นเท็จ (False) แต่

สำหรับ `==` จะรวมถึงการเปรียบเทียบชนิดของข้อมูลด้วย รูปแบบของ Comparison Operator ที่กำหนดไว้ใน PHP มีดังนี้

ตารางที่ 2.4 โอเปอเรเตอร์เชิงเปรียบเทียบ (Comparison Operators)

เครื่องหมาย	ความหมาย	ค่าที่ทำให้เป็นจริง
<code>a == b</code>	เท่ากับ	ค่าของ a เท่ากับ b
<code>a === b</code>	เหมือนกันทั้งค่า และชนิด	ค่าและชนิดของ a เท่ากับ b
<code>a != b</code>	ไม่เท่ากับ	ค่าของ a ไม่เท่ากับ b
<code>a <> b</code>	ไม่เท่ากับ	ค่าของ a ไม่เท่ากับ b
<code>a < b</code>	น้อยกว่า	ค่าของ a น้อยกว่า b
<code>a > b</code>	มากกว่า	ค่าของ a มากกว่า b
<code>a <= b</code>	น้อยกว่าหรือเท่ากับ	ค่าของ a น้อยกว่าหรือเท่ากับ b
<code>a >= b</code>	มากกว่าหรือเท่ากับ	ค่าของ a มากกว่าหรือเท่ากับ b

- โอเปอเรเตอร์เชิงตรรกะ (Boolean (logical) Operators) การเปรียบเทียบเชิงตรรกะมักจะนำมาใช้ในประโยคที่ใช้การตัดสินใจก่อนเข้าไปทำหรือการตัดสินใจเมื่อทำงานจบ ในแต่ละรอบ เช่น การเปรียบเทียบในประโยค `if....else` เป็นต้น ซึ่งการเปรียบเทียบในรูปแบบต่าง ๆ มีดังตารางต่อไปนี้

ตารางที่ 2.5 โอเปอเรเตอร์เชิงตรรกะ (Boolean (logical) Operators)

เครื่องหมาย	ผลลัพธ์
<code>!a</code>	เป็นจริงเมื่อค่า a เป็นเท็จ
<code>a && b</code>	เป็นจริงเมื่อค่าของ a และ b เป็นจริงหรือเป็นเท็จเหมือนกัน
<code>a b</code>	เป็นจริงเมื่อค่าของ a และ b ไม่เป็นเท็จทั้งคู่
<code>a and b</code>	เหมือนกับ <code>&&</code> แต่จะมีความสำคัญน้อยกว่า
<code>a or b</code>	เหมือนกับ <code> </code> แต่จะมีความสำคัญน้อยกว่า

4.7 การใช้ไฟล์ด้วยคำสั่ง include()

PHP อนุญาตให้มีการเรียกใช้ไฟล์จากไฟล์อื่นได้ เพื่อนำไฟล์มาเข้าร่วมทำงาน หรืออ้างถึงตัวแปรการทำงานในอีกไฟล์หนึ่ง รูปแบบการเขียนมีดังนี้

```
<? include("ชื่อไฟล์ที่เรียกใช้");?>
```

ภายในเครื่องหมาย (".....") จะกำหนดแหล่งข้อมูล และชื่อไฟล์ที่มีนามสกุลเป็น .inc หรือ .php เช่น

```
<? include("include.inc");?>
```

จากคำสั่งข้างต้นเป็นการเรียกใช้ไฟล์ include.inc เข้าร่วมใช้งานซึ่งประกาศเป็น parameter ของฟังก์ชัน include

4.8 Combination Assignment Operators

การแสดงผลการกระทำทางคณิตศาสตร์ในรูปแบบของสมการ PHP กำหนดให้สามารถเขียนแบบย่อได้ ดังตัวอย่างในตาราง

ตารางที่ 2.6 Combination Assignment Operators

เครื่องหมาย	การนำไปใช้	เทียบเท่ากับ
+=	$\$x+=\y	$\$x=\$x+\$y$
-=	$\$x-=\y	$\$x=\$x-\$y$
=	$\$x=\y	$\$x=\$x*\$y$
/=	$\$x/=\y	$\$x=\$x/\$y$
%=	$\$x\%=\y	$\$x=\$x\%+\$y$
.=	$\$x.=\y	$\$x=\$x.\$y$

4.9 การเพิ่มและลดค่าตัวแปรโดยใช้ Pre- และ Post-

การเพิ่มค่า (++) และการลดค่า (--) ของ pre- และ post- เป็นการเขียนสมการทางคณิตศาสตร์ในรูปแบบย่อ เพื่อความสะดวกในการนำไปใช้งาน ดังนี้

ตารางที่ 2.7 การเพิ่มและลดค่าตัวแปร โดยใช้ Pre- และ Post-

รูปแบบ	เท่ากับ	ความหมาย
$\$x++$	$\$x=\$x+1$	นำค่า x ไปคำนวณก่อนแล้วจึงเพิ่มค่า x อีก 1
$++\$x$	$\$x=\$x+1$	เพิ่มค่า x อีก 1 ก่อนแล้วจึงนำค่า x ไปคำนวณ
$\$x--$	$\$x=\$x-1$	นำค่า x ไปคำนวณก่อนแล้วจึงลดค่า x ลง 1
$--\$x$	$\$x=\$x-1$	ลดค่า x ลง 1 ก่อนแล้วจึงนำค่า x ไปคำนวณ

4.10 ฟังก์ชัน (Function)

ฟังก์ชันเป็นส่วนของโค้ดที่ระบุการทำงานของโปรแกรมโดยเมื่อต้องการเรียกใช้งานก็ให้ทำการเรียกใช้ ซึ่งในการเรียกอาจจะส่งค่าบางอย่างไปพร้อมกันได้ การสร้างฟังก์ชันขึ้นมาใช้งานมีประโยชน์คือ ในกรณีที่ต้องการเรียกใช้การทำงานแบบใดแบบหนึ่งบ่อยครั้ง การเรียกใช้ฟังก์ชันที่มีการส่งค่ากลับอาจจะเก็บค่าจากฟังก์ชันนั้นไว้ในตัวแปรก่อน เพื่อเรียกใช้งานต่อไปได้ สำหรับฟังก์ชันที่ไม่มีการคืนค่าจะต้องระบุคำสั่ง return ในส่วนท้ายของฟังก์ชันโดยไม่มีส่วนใดต่อเพื่อบอกให้โปรแกรมกลับไปยังจุดที่เรียกใช้ฟังก์ชันนั้นด้วย ส่วนฟังก์ชันที่มีการคืนค่าก็จะต้องระบุคำสั่ง return ตัวแปรที่เก็บค่าที่ต้องการ ; ไว้ด้วย โดยถ้าการคืนค่านั้นได้ค่าใดค่าหนึ่งออกมา ก็จะนำค่านั้นส่งกลับไปยังจุดที่เรียกมา

4.11 การใช้งาน String

การทำงานเกี่ยวกับ String ซึ่งเป็นชุดของตัวอักษรนั้น PHP ได้เตรียมฟังก์ชันต่าง ๆ ไว้ให้มากมาย ดังต่อไปนี้

- การเชื่อม String

การนำ String หรือข้อความมาเชื่อมต่อกันหรือเรียกว่า “String Concatenation”

ภาษา PHP กำหนดให้ใช้เครื่องหมาย “.” วางไว้ระหว่างข้อความที่ต้องการ เช่น

```
$String = “การเขียน”;
echo $String. “PHP Script”;
```

เป็นการแสดงข้อความของ String ที่เก็บอยู่ในตัวแปร \$String1 ต่อกับคำว่า “PHP Script” ซึ่งจะได้ผลการทำงานเป็น “การเขียน PHP Script”

- การตัดช่องว่างภายใน String

การตัดช่องว่างภายใน String ส่วนมากจะนำมาใช้เมื่อต้องการเปรียบเทียบ String หรือต้องการเก็บ String ในฐานข้อมูล ซึ่งมี 3 ฟังก์ชัน ได้แก่

ฟังก์ชัน trim()

ใช้สำหรับตัดช่องว่างทั้งหมดที่อยู่ด้านหน้าและด้านหลังของ String ทิ้งไป เช่น

```
string trim( String str)
```

โดยที่ Str คือ ตัวแปรที่มีลักษณะเป็น String

ฟังก์ชัน ltrim()

ใช้สำหรับตัดช่องว่างที่อยู่ด้านหน้าของ String เช่น

```
string ltrim(string str [string, charlist])
```

ฟังก์ชัน chop()

ใช้สำหรับตัดช่องว่างที่อยู่ด้านหลังของ String

```
string chop(string str);
```

4.12 การจัดการเกี่ยวกับวันและเวลา (Date and Time)

การเขียนสคริปต์เพื่อจัดการเกี่ยวกับเวลาและวันที่ถือว่าเป็นงานที่พบบ่อยมากในการเขียนสคริปต์ทั่ว ๆ ไป แต่ถึงแม้ว่าภาษา PHP จะเตรียมฟังก์ชันสำหรับอำนวยความสะดวกไว้อย่างมากมาย การนำฟังก์ชันต่าง ๆ เหล่านั้นไปใช้ก็ยังคงมีความสับสน และข้อจำกัดในการใช้งานอยู่บ้าง ถ้าผู้ใช้ไม่เข้าใจการทำงานอย่างแท้จริง ซึ่งการจัดการเกี่ยวกับวันและเวลามีรายละเอียดดังต่อไปนี้

- **Timestamp**

Timestamp เป็นเวลาแบบ Timestamp เป็นเลขจำนวนเต็มที่แสดงเป็นวินาที โดยเริ่มนับตั้งแต่ช่วงเวลาที่กำหนดใน Unix คือเริ่มนับในเดือนมกราคม ปี ค.ศ. 1970 timestamp สามารถนำมาใช้เป็นเวลาสำหรับเทียบหาเวลาอื่น ๆ ที่ต้องการทราบ ตัวอย่าง เช่น ถ้าต้องการรู้เวลาใน 1 สัปดาห์ถัดไปเมื่อเทียบกับปัจจุบันก็สามารถทำได้โดยบวกค่า Timestamp 640800 เพิ่มจากค่า Timestamp เดิม โดยยังสามารถใช้ operation ต่าง ๆ ได้อย่างครบถ้วนอีกด้วย ถ้าต้องการเปลี่ยนค่า Timestamp เป็นรูปแบบทั่วไป ก็เพียงใช้ฟังก์ชันของ PHP ในการแปลงและยังสามารถแปลงกลับมาก็ได้เช่นกัน

- **การแปลงเวลา Timestamp**

การแปลงค่า Timestamp มีอยู่หลายวิธีโดยอาจจะแปลงจากเวลาปัจจุบันของเครื่องที่ทำงานอยู่หรือแปลงจากการกำหนดวันเวลาที่ต้องการได้

ฟังก์ชัน mktime() , gmmktime()

เป็นฟังก์ชันที่ใช้สำหรับสร้างเวลา timestamp ขึ้นมาจากลำดับตัวเลขที่แสดงวัน เดือน ปี ชั่วโมง นาที และวินาที ตามที่กำหนดไว้ โดยมีรูปแบบการใช้งาน ดังนี้

```
mktime(int hour, int minute, int second, int month, int day, int year, int is_dst)
gmmktime(int hour, int minute, int second, int month, int day, int year, int is_dst)
```

ทั้ง 2 ฟังก์ชันมีหน้าที่แปลง Argument ที่ส่งเข้าไปเป็นเวลาแบบ Timestamp เช่นเดียวกัน แต่สิ่งที่แตกต่างกัน คือ ฟังก์ชัน gmmktime() จะคาดเดาค่าที่รับมาเป็นแบบ GMT (Greenwich Mean Time หรือเรียกว่า UTC “Universal Time Constant”)

ตัวอย่างเช่น การใช้ฟังก์ชัน mktime() , gmmktime()

```
1. <?php
2. echo mktime(0,0,0,7,25,2003)."<br>";
3. echo mktime(0,0,0,5,30,2002);
4. ?>
```


แสดงผลลัพธ์ได้จากการใช้ฟังก์ชัน mktime() , gmmktime()

```
1059066000
1022691600
```

ฟังก์ชัน time()

ฟังก์ชัน time() เป็นฟังก์ชันที่ง่ายที่สุดในการสร้างเวลา timestamp เนื่องจากไม่ต้องรับค่า argument ใด ๆ สำหรับค่าที่ถูกส่งจะเป็นเวลา timestamp ของเวลาปัจจุบัน โดยมีรูปแบบการใช้งานดังนี้

รูปแบบ

```
int time()
```

ตัวอย่างเช่น การใช้ฟังก์ชัน time()

```
1. <?php
2.     echo time();
3. ?>
```

ฟังก์ชัน strtotime() เป็นฟังก์ชันที่มีประโยชน์มาก โดยสามารถรับค่าที่เป็น String แทนวัน เวลา และคืนค่าเป็นเวลา Timestamp ออกมา รูปแบบของฟังก์ชัน strtotime() มีรูปแบบดังนี้

รูปแบบ

```
1. <?php
2.     echo strtotime("last month")."<br>";
3.     echo strtotime("2003-2-14 13:30:55")."<br>";
4. ?>
```

คำอธิบาย

บรรทัดที่ 2 แสดงหมายเลขของเดือนที่ผ่านมา

บรรทัดที่ 3 แสดงวันเวลาตามที่ระบุ

แสดงผลลัพธ์ได้จากการใช้ฟังก์ชัน `time()`

```
1041824929
1045204255
```

- การแปลงค่า Timestamp และการจัดการเกี่ยวกับวันเวลา

หลังจากที่ได้รู้จักวิธีการสร้าง Timestamp แล้ว ในหัวข้อนี้จะเป็นการอธิบายเกี่ยวกับฟังก์ชันสำหรับแปลงค่าจากรูปแบบของ Timestamp ให้อยู่ในรูปแบบอื่นที่ต้องการได้ ซึ่งมีรายละเอียดดังนี้

ฟังก์ชัน `date()`

ฟังก์ชัน `date()` เป็นฟังก์ชันที่แสดงวันที่และเวลา ซึ่งมีรูปแบบในการแสดงหลากหลายรูปแบบ ขึ้นอยู่กับการกำหนดค่าพารามิเตอร์กับฟังก์ชัน รูปแบบของฟังก์ชัน `date()` มีรูปแบบดังนี้

```
date(string format);
```

โดยที่ `format` หมายถึงรูปแบบของวันที่และเวลา

ตัวอย่างเช่น การใช้ฟังก์ชัน `date()`

```
1.      echo "<center><font size= 14><b>";
2.      echo date("D j F Y");
3.      echo "</b></font></center>";
```

คำอธิบาย

บรรทัดที่ 2 เรียกใช้ฟังก์ชัน `date()` แสดงวันที่ปัจจุบันตามรูปแบบที่กำหนด โดยการส่งค่าพารามิเตอร์เป็น "D j F Y" โดยที่

- พารามิเตอร์ D แสดงว่าเป็นตัวย่อ (คือ Wed)
- พารามิเตอร์ j แสดงวันที่ หากวันที่อยู่ในช่วง 1-9 จะไม่แสดงตัวเลขนำหน้า (คือ 5)
- พารามิเตอร์ F แสดงชื่อแบบเต็ม (คือ February)
- พารามิเตอร์ Y แสดงปี ค.ศ. แบบ 4 ตัวอักษร (คือ 2003)

แสดงผลลัพธ์ได้จากการใช้ฟังก์ชัน date()

Wed 5 February 2003

ตารางแสดงพารามิเตอร์ของฟังก์ชัน date()

ตารางที่ 2.8 แสดงรูปแบบพารามิเตอร์ของฟังก์ชัน date()

Format	คำอธิบาย
a	แสดงเวลาช่วงเช้าหรือช่วงบ่าย เป็นตัวพิมพ์เล็ก คือ “am” หรือ “pm”
A	แสดงเวลาช่วงเช้าหรือบ่าย เป็นตัวพิมพ์ใหญ่ คือ “AM” หรือ “PM”
B	เวลาบนระบบอินเทอร์เน็ต ศึกษาข้อมูลเพิ่มเติมได้จาก http://swatch.com/internettime.php
d	แสดงวันด้วยเลข 2 หลัก คือ “01” ถึง “31”
D	แสดงวันด้วยตัวย่อ 3 ตัวอักษร คือ “Mon” ถึง “Sun”
F	แสดงชื่อเดือนแบบเต็ม คือ “January” ถึง “December”
g	แสดงชั่วโมง ในรูปแบบ 12 ชั่วโมง คือ “1” ถึง “12”
G	แสดงชั่วโมงในรูปแบบ 24 ชั่วโมง คือ “0” ถึง “24”
h	แสดงชั่วโมงในรูปแบบ 12 ชั่วโมง โดยแสดงตัวเลข 0 ด้วย คือ “01” ถึง “12”
H	แสดงชั่วโมงในรูปแบบ 24 ชั่วโมง โดยแสดงเลข 0 ด้วย คือ “00” ถึง “24”
i	แสดงนาทีโดยแสดงเลข 0 ด้วย คือ “00” ถึง “59”
I	แสดงช่วงเวลา Daylight saving จะคืนค่าเป็น 1 หากเป็นช่วง Daylight saving และคืนค่าเป็น 0 เมื่อไม่ใช่เวลา Daylight saving
J	แสดงวันที่โดยไม่แสดงเลข 0 คือ “1” ถึง “31”
l	แสดงชื่อวันแบบเต็ม คือ “Monday” ถึง “Sunday”
L	จะคืนค่าเป็น 1 หากเป็นปีอธิสุรทิน และคืนค่าเป็น 0 หากไม่ใช่ปี อธิสุรทิน
m	แสดงเดือนด้วยเลข 2 หลัก โดยแสดงเลข 0 คือ “01” ถึง “12”
M	แสดงชื่อเดือนด้วยตัวย่อ 3 ตัวอักษร คือ “Jan” ถึง “Dec”
N	แสดงเดือน ด้วยเลข 2 หลัก โดยไม่แสดงเลข 0 คือ “1” ถึง “12”
s	แสดงวินาทีโดยแสดงเลข 0 คือ “00” ถึง “59”
S	แสดงลำดับของวันที่ ตัวอย่างเช่น “st”, “nd”, “rd” หรือ “th”
t	จำนวนวันทั้งหมดในแต่ละเดือน คือ “28” ถึง “31”
T	แสดงเขตของ TimeZone ซึ่งแสดงด้วยเลข 3 ตัวอักษร เช่น “ETS”
U	ผลรวมของวินาทีตั้งแต่วันที่ 1 มกราคม ค.ศ. 1970 จนถึงปัจจุบัน

ตารางที่ 2.8 แสดงรูปแบบพารามิเตอร์ของฟังก์ชัน date() (ต่อ)

Format	คำอธิบาย
W	แสดงวันในแต่ละสัปดาห์ ด้วยเลข 1 หลัก คือ “0” (วันอาทิตย์) ถึง “6” (วันเสาร์)
y	แสดงปี ค.ศ. ด้วยตัวเลข 2 หลัก ตัวอย่างเช่น “02”
Y	แสดงปี ค.ศ. ด้วยตัวเลข 4 หลัก ตัวอย่าง เช่น “2002”
z	แสดงวันที่ในแต่ละปี คือ “0” ถึง “365”
Z	แสดงวินาทีใน Timezone ปัจจุบัน คือ “-43200” ถึง “43200”

ฟังก์ชัน checkdate()

ฟังก์ชัน checkdate() เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบความถูกต้องของวันที่ของผู้ใช้ ตัวอย่างเช่น การป้อนวันที่อย่างถูกต้องจะต้องไม่เกิน 31 วันเท่านั้น หรือเดือนจะต้องเป็นตัวเลขที่อยู่ระหว่าง 1 ถึง 12 เป็นต้น มีรูปแบบดังต่อไปนี้

รูปแบบ

```
int checkdate(int month , int day, int year) ;
```

โดยที่	month	หมายถึง	เดือน
	day	หมายถึง	วัน
	year	หมายถึง	ปี

ตัวอย่างเช่น การใช้ฟังก์ชัน checkdate()

```
1.     if (checkdate(15, 6, 2003) )
2.         echo “<center><font size = 14 >รูปแบบวันที่ถูกต้อง </font></center>”;
3.         else echo “<center><font size = 14>รูปแบบวันที่ไม่ถูกต้อง </font></center>”;
```

คำอธิบาย

ใช้ฟังก์ชัน checkdate() ตรวจสอบรูปแบบของวันที่ว่าแต่ละค่าถูกต้องหรือไม่ หากถูกต้องจะแสดงข้อความ “รูปแบบวันที่ถูกต้อง” จากตัวอย่างส่งค่า checkdate(15, 6, 2003) ซึ่งค่าของเดือนเท่ากับ 15 จะไม่ถูกต้อง เพราะเกิน 12 ฟังก์ชันจึงคืนค่าเป็นเท็จมาให้ แสดงผลลัพธ์จากการใช้ฟังก์ชัน checkdate()

แสดงผลลัพธ์ได้จากการใช้ฟังก์ชัน `checkdate()`

รูปแบบวันที่ไม่ถูกต้อง

5. Session

เนื่องจากการทำงานของเว็บเพจเป็นแบบ `stateless environment` ซึ่งเป็นการติดต่อแบบทางเดียว หมายความว่า ในขณะที่เว็บเซิร์ฟเวอร์กำลังให้บริการจะไม่สามารถระบุผู้ใช้ได้ว่าใครที่กำลังเข้าชมและใช้บริการอะไรบ้าง ดังนั้นเมื่อผู้ใช้องค์ที่ต้องการไปเว็บเพจจึงไม่สามารถส่งสิ่งที่ผู้ใช้องค์ต้องการได้อย่างถูกต้อง และเหมาะสม `session` จึงถูกนำมาใช้แก้ปัญหานี้ โดยตัวแปร `session` จะถูกเก็บเป็นไฟล์แบบชั่วคราวบนหน่วยความจำหลักของเครื่องเซิร์ฟเวอร์ ซึ่งไฟล์เหล่านี้จะประกอบด้วยข้อมูลต่าง ๆ ที่เกี่ยวกับการใช้งานของผู้ใช้ในขณะที่อายุของ `session` จะหมดเมื่อผู้ใช้ปิดบราวเซอร์หรือเรียกใช้ฟังก์ชันที่ไปเรียกใช้ฟังก์ชันสำหรับลบข้อมูล `session` เหล่านี้ ได้แก่ ฟังก์ชัน `session_unregister()` หรือ `unset() session` จึงถือเป็นเครื่องมือที่สำคัญสำหรับการใช้ในการสร้างแอปพลิเคชัน เช่น `forums` หรือ `shopping carts` เป็นต้น

5.1 แนะนำ session

การใช้ `session` เริ่มมีใน PHP 4 โดยใน PHP 3 ซึ่งเป็นรุ่นก่อนหน้านี้จะต้องเรียกใช้ Library เพิ่มเติม เรียกว่า “PHPLib” นั่นหมายความว่า จะต้องทำการ `include` ไฟล์นี้ทุกครั้งที่ต้องการใช้งานเกี่ยวกับ `session` แต่ในรุ่น PHP 4 จะมีมาให้พร้อมการติดตั้งจึงไม่จำเป็นต้อง `include` ก็สามารถเรียกใช้งานได้ทันทีนับว่าสะดวกมากทีเดียว `session` ใน PHP จะทำงานด้วย `session ID` ซึ่งได้มาจากการสุ่มตัวเลข `session ID` ถูกสร้างขึ้นโดย PHP และ เก็บไว้บนเครื่องของผู้ใช้ตามอายุของ `session` ซึ่งสามารถส่งไปกับ URL ได้เลย `session ID` จะทำหน้าที่คล้ายกับคีย์ที่อนุญาตให้ใช้ได้กับตัวแปรเฉพาะเท่านั้น ซึ่งตัวแปรดังกล่าว คือ ตัวแปร `session` ซึ่งรายละเอียดของตัวแปรเหล่านี้จะเก็บไว้ที่เซิร์ฟเวอร์ ส่วน `session ID` เป็นข้อมูลเดียวที่ปรากฏบนฝั่งไคลเอนต์ในเวลาที่ผู้ใช้ติดต่อไปยังเว็บไซต์ `session ID` จะปรากฏ ผ่านทาง URL ซึ่งจะช่วยให้สามารถเข้าถึงตัวแปร `session` ที่เก็บบนเซิร์ฟเวอร์สำหรับ `session` นั้น ๆ ได้ โดยทั่วไปแล้วตัวแปร `session` จะเก็บใน `flat file` บนเซิร์ฟเวอร์ แต่สามารถเขียนฟังก์ชันขึ้นมาจัดการเองได้ เพื่อเก็บลงฐานข้อมูลซึ่งจะกล่าวต่อไป สำหรับการตั้งค่า `session` เบื้องต้นนั้น PHP จะเป็นผู้กำหนดไว้ โดยสามารถเข้าไปดูได้ที่ไฟล์ `phpinfo.php` ถ้าติดตั้ง PHP เรียบร้อยแล้ว โดยการใช้ `session` ใน PHP นั้นจะต้องกำหนดค่าบางอย่างเพื่อให้เหมาะสมกับผู้ใช้แต่ละคน ในขั้นแรกควรที่จะกำหนดตำแหน่งการจัดเก็บไฟล์ `session` โดยให้เปิดไฟล์ `phpinfo.php` จากนั้นหาคำว่า `session.save_path` จะพบว่า PHP ได้

กำหนดค่าเริ่มต้นไว้เป็น /tmp ซึ่งใน Windows ไม่ได้สร้างโฟลเดอร์นี้ไว้ให้ เพื่อความสะดวกและเหมาะสมให้เปลี่ยน /tmp เป็น c:/phpsessions บันทึกไฟล์ และอย่าลืมสร้างโฟลเดอร์ phpsessions ไว้ในไดเรกทอรี c: ด้วย

5.2 การสร้างและจัดการ session

ขั้นตอนการสร้าง session และจัดการ session มี 4 ขั้นตอน ดังนี้

1. การเริ่มต้น session
2. การกำหนดตัวแปร session
3. การใช้ตัวแปร session
4. การยกเลิกตัวแปรและการทำลาย session

1. การเริ่มต้น session

ก่อนที่จะใช้ฟังก์ชันการทำงานของ session นั้น จะต้องเรียกใช้ฟังก์ชันสำหรับเริ่มต้น session ก่อน ซึ่งมีด้วยกัน 3 วิธี คือ

วิธีที่ 1 การเริ่มต้น session ด้วยฟังก์ชัน `session_start()`

ฟังก์ชัน `session_start()`

เป็นฟังก์ชันสำหรับเริ่มต้นการใช้ session โดยจะตรวจสอบว่ามี session ID อยู่แล้วหรือไม่ ถ้าไม่มีจะทำการสร้างขึ้นมาใหม่ ถ้ามีอยู่แล้วจะโหลดตัวแปร session ที่สามารถใช้งานได้

รูปแบบ

`bool session_start (void)`

วิธีที่ 2 session จะถูกเริ่มต้นเมื่อมีการสร้างตัวแปร session เกิดขึ้น

วิธีที่ 3 ทำการแก้ไขค่า `session.auto_start` ในไฟล์ `php.ini` จาก 0 เป็น 1 เพื่อให้สร้าง session อัตโนมัติเมื่อผู้ใช้เริ่มดาวน์โหลดโหลดเว็บ

2. การกำหนดตัวแปร session

การกำหนดตัวแปร session สามารถทำได้โดยใช้ฟังก์ชัน `session_register()` ซึ่งจะช่วยให้สคริปต์อื่น ๆ สามารถใช้ตัวแปร session ตัวเดียวกันนี้ได้

ฟังก์ชัน `session_register()`

เป็นฟังก์ชันสำหรับกำหนดตัวแปร session

รูปแบบ

```
boolean session_register(mixed name [,mixed...])
```

โดยที่ name หมายถึง ชื่อของตัวแปร session ที่จะกำหนด โดยสามารถกำหนดได้หลายตัว ตัวอย่างเช่น การกำหนดตัวแปร session

```
1. <?
2. session_register("count");
3. echo "คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน ".$count++." ครั้ง";
4. ?>
```

คำอธิบาย

จากตัวอย่างจะใช้ฟังก์ชัน `session_register` กำหนดตัวแปร session ชื่อ `count` สังเกตได้ว่าชื่อของตัวแปรไม่มีเครื่องหมาย "\$" นำหน้า จากนั้นแสดงจำนวนครั้งที่ผู้ใช้เข้ามาชมเว็บเพจหน้านี้ ผลลัพธ์ก็คือจะแสดงข้อความว่า "คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน 3 ครั้ง"

การกำหนดค่าให้กับตัวแปร session

หลังจากลงทะเบียนตัวแปร session โดยใช้ฟังก์ชัน `session_register()` สามารถกำหนดค่าตัวแปรนั้นได้ โดยมี รูปแบบการใช้งาน ดังนี้

รูปแบบ

```
$_session["session's name"] = session 's value
```

โดยที่ `session's name` หมายถึง ชื่อของตัวแปร session ที่ต้องการกำหนดค่า ซึ่งอาจจะยังไม่ต้องลงทะเบียนก็ได้

`session's value` หมายถึง ค่าของ session ที่ต้องการกำหนดให้กับตัวแปร

ตัวอย่างเช่น การกำหนดค่าให้กับตัวแปร session

```
1. <?
2. session_start();
3. $_SESSION['count'] = "50";
4. echo "คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน ".$count++."ครั้ง";
5. ?>
```

คำอธิบาย

กำหนดค่าให้กับตัวแปร session "count" ให้มีค่าเท่ากับ 50 จากนั้นจึงแสดงค่าของตัวแปร ซึ่งค่าของ session จะยังคงอยู่ จนกระทั่ง session ถูกปิดโดยการปิดจากบราวเซอร์ หรือสั่งปิดด้วยคำสั่ง session_unregister()

3. การใช้ตัวแปร session

การเรียกใช้ตัวแปร session สามารถทำได้ 2 วิธี วิธีแรก สามารถเรียกใช้ตัวแปรแบบวิธีเดียวกับการเข้าถึงตัวแปรธรรมดาทั่วไป (เช่น \$var) โดยวิธีนี้สามารถเข้าไปกำหนดในไฟล์ php.ini ที่ค่า register_globals ในเป็น on ส่วนอีกวิธีคือ การเรียกใช้ตัวแปรผ่านทาง associative \$HTTP_SESSION_VARS เช่น

```
$HTTP_SESSION_VARS["var"]
```

ตัวแปร session นี้ไม่สามารถถูกเขียนค่าทับด้วยข้อมูล GET หรือ POST ซึ่งนับเป็นคุณสมบัติที่ดีในการรักษาความปลอดภัย แต่สิ่งหนึ่งที่เราควรคำนึงถึงเป็นพิเศษคือ การตรวจสอบตัวแปร session เนื่องจากตัวแปรทั่วไปสามารถสร้าง โดยผู้ใช้ผ่านทางวิธี GET หรือ POST ได้ ดังนั้น จึงไม่สมควรที่จะตรวจสอบตัวแปร session ด้วยฟังก์ชัน isset() หรือ empty() แต่ควรใช้ฟังก์ชัน session_is_registered() ในการตรวจสอบตัวแปร session แทน

ฟังก์ชัน session_is_registered()

ฟังก์ชันสำหรับตรวจสอบตัวแปร session

รูปแบบ

```
bool session_is_registered(string name)
```

โดยที่ name หมายถึง ชื่อของตัวแปร session ที่ต้องการค้นหา

ตัวอย่างเช่น การกำหนดค่าให้กับตัวแปร session

```
1. <?
2. session_register("count");
3. if(session_is_registered("count")){
4. echo "ตัวแปร session count ถูกสร้างแล้ว<br>";
5. echo "คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน ".$count++." ครั้ง";
6. }
7. else
8. echo "ไม่พบตัวแปร session count";
9. ?>
```

คำอธิบาย

จากตัวอย่างจะเป็นการตรวจสอบตัวแปร session ที่มีชื่อ count ว่า ถูกสร้างขึ้น หรือยัง ถ้าถูกสร้างขึ้นแล้วจะแสดงค่าที่อยู่ภายในตัวแปร session count นั้น ซึ่งแสดงผลลัพธ์ด้วยข้อความดังนี้ “ตัวแปร session count ถูกสร้างแล้ว” และ “คุณเข้าชมเว็บเพจหน้านี้เป็นจำนวน 10 ครั้ง”

ฟังก์ชัน session_unregister()

เมื่อสร้างตัวแปร session แล้วหากไม่ต้องการตัวแปรนั้นอีก สามารถยกเลิกด้วยฟังก์ชัน

session_unregister()

รูปแบบ

```
bool session_unregister(string name)
```

โดยที่ name หมายถึง ชื่อของตัวแปร session
 เมื่อต้องการจะปิด session ควรจะทำการยกเลิกตัวแปร session ทุกตัวก่อน แล้วจึงปิด session ด้วยฟังก์ชัน session_destory()

ฟังก์ชัน session_destory()

ฟังก์ชันสำหรับปิดการเชื่อมต่อ session

รูปแบบ

```
bool session_destory(string name)
```

โดยที่ name หมายถึง ชื่อของตัวแปร session

6. PHP และระบบการจัดการฐานข้อมูล

PHP มีความสามารถในการติดต่อกับฐานข้อมูลต่าง ๆ มากมาย เช่น dBase, Informix, Interbase, Oracle, MySQL และอื่น ๆ รวมทั้งการติดต่อผ่าน ODBC (Open Database Connectivity) โดย MySQL นับเป็นระบบจัดการฐานข้อมูลที่ได้รับความนิยมสูงสุดในปัจจุบัน เนื่องจากสามารถทำงานได้ในระบบปฏิบัติการต่างชนิดกัน เช่น UNIX, Linux และ Windows เป็นต้น

ในการใช้ PHP ติดต่อกับระบบจัดการฐานข้อมูลนั้นจำเป็นต้องใช้คำสั่ง SQL ร่วมด้วย โดยรายละเอียดเกี่ยวกับ MySQL ได้พูดถึงไปในหัวข้อ MySQL ไปตอนต้นแล้ว

6.1 การสร้างส่วนเชื่อมโยงฐานข้อมูล

MySQL เป็นระบบการจัดการฐานข้อมูลซึ่งประกอบด้วย ระบบจัดการฐานข้อมูล และฐานข้อมูลเซิร์ฟเวอร์ ซึ่งเป็นโปรแกรมฟรีแวร์ มีขนาดเล็ก แต่มีความสามารถสูงซึ่งการติดต่อกับ MySQL โดยใช้ PHP นั้นมีฟังก์ชันมีการติดต่อดังนี้

```
MySQL_connect([hostname(port)[:path to socket]], [username], [password])
```

โดยที่ hostname หมายถึง ชื่อของ host ที่ MySQL กำลังทำงาน เช่นอาจจะเป็นชื่อเครื่อง หรือหมายเลข IP Address ที่กำหนดไว้สำหรับเครื่องนั้นก็ได้

username หมายถึง ชื่อผู้ที่กำหนดไว้ใน การติดต่อกับฐานข้อมูล MySQL

password หมายถึง รหัสผ่านที่ใช้ร่วมกับ username

หลังจากที่ผู้ใช้ติดต่อกับฐานข้อมูล และทำงานเรียบร้อยแล้ว จะต้องทำการปิดการติดต่อกับฐานข้อมูลทุกครั้ง โดยใช้ คำสั่ง `mysql_close()` เพื่อที่จะได้เป็นการลด *overhead* (การจองเซสชัน :*session connection* ในการติดต่อเซิร์ฟเวอร์) ในกรณีที่ได้เปิดการติดต่อไว้แล้ว จะได้ไม่ต้องเปิดใหม่

6.2 การเรียกใช้ฐานข้อมูลของ MySQL

คำสั่งใน MySQL ที่ใช้เลือกฐานข้อมูลมาใช้งาน มีดังนี้

```
“use ชื่อฐานข้อมูล;”
```

ส่วนใน PHP ก็สามารถเลือกได้เช่นกัน โดยใช้คำสั่ง

```
mysql_select_db(“ชื่อฐานข้อมูล”)
```

โดยที่ “ชื่อฐานข้อมูล” คือฐานข้อมูลที่ต้องการเลือกใช้ เช่น ต้องการเลือกใช้ฐานข้อมูลชื่อ *book* จะใช้คำสั่งดังนี้ `mysql_select_db(“book”);`

6.3 เก็บค่าการติดต่อไว้ในตัวแปรเพื่อเรียกใช้งาน

ในการติดต่อกับฐานข้อมูลจะใช้คำสั่ง `MySQL_query()` เพื่อเข้าถึงตาราง และข้อมูลภายในฐานข้อมูล แต่ก่อนการใช้ คำสั่งนี้ผู้ใช้ควรเก็บค่าการติดต่อไว้ในรูปของตัวแปรเพื่อความสะดวกในการนำมาใช้ภายหลัง

```
$query=”select * from books where “.$searchtype.” Like “%”$searchterm.”%”;
```

ในตัวอย่างนี้เป็นการสร้าง *query* สำหรับค้นหาข้อมูลที่ได้จากการใช้ข้อมูลของผู้ใช้เก็บไว้ในตัวแปร (*\$searchtype*) โดยค้นหาข้อมูลในตาราง *book* ซึ่งมีรูปแบบของการสืบค้นที่ถูกกำหนดโดยตัวแปร *\$searchtype* ผลของการใช้ตัวแปร *\$query* จะถูกส่งไปเก็บค่าในตัวแปร *\$result* โดยฟังก์ชัน `MySQL_query($query);` ดังนี้

```
$result = mysql_query($query);
```