

บทที่ 1 ความรู้เบื้องต้นเกี่ยวกับเว็บโปรแกรมมิ่ง

ความหมายของเว็บโปรแกรมมิ่ง

เว็บโปรแกรมมิ่ง (Web programming) คือ การเขียนโปรแกรมบนเว็บไซต์ โดยใช้การพัฒนาแอปพลิเคชันแบบ Web Based Application โดยใช้ภาษามากมายหลายภาษา เช่น Perl, PHP, JAVA, C#, XML หรือ ASP.NET เป็นต้น ซึ่งแต่ละภาษาล้วนต่างมีข้อดีข้อเสียแตกต่างกันไป โดยมุ่งให้รองรับความก้าวหน้าของอินเทอร์เน็ต ตัวอย่างเช่น การเขียนโปรแกรมเพื่อรับข้อมูลจากฟอร์ม หรือโปรแกรม ChatRoom

ประเภทของการเขียนโปรแกรมบนเว็บ

การเขียนโปรแกรมบนเว็บสามารถแบ่งได้เป็น 2 ประเภทคือ Client-Side และ Server-Side

1. Client-Side เป็นการเขียนโปรแกรมที่ใช้ Resource (Memory) จากเครื่องคอมพิวเตอร์ของผู้เรียกใช้โปรแกรม ตัวอย่าง เช่น Javascript เป็นต้น
2. CGI (Common Gateway Interface) เป็น Server-Side Language นั่นคือการประมวลผลทั้งหมดถูกจัดการบน web server

องค์ประกอบการพัฒนาเว็บโปรแกรมมิ่ง

1. เซิร์ฟเวอร์ (server) หรือ เครื่องแม่ข่าย คือ เครื่องหรือซอฟต์แวร์คอมพิวเตอร์ซึ่งทำงานให้บริการ ในระบบเครือข่ายแก่ลูกข่าย (ซึ่งให้บริการผู้ใช้อีกทีหนึ่ง) เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์นี้ควรมีประสิทธิภาพสูง มีความเสถียร สามารถให้บริการแก่ผู้ใช้ได้เป็นจำนวนมาก ภายในเซิร์ฟเวอร์ให้บริการได้ด้วยโปรแกรมบริการ ซึ่งทำงานอยู่บนระบบปฏิบัติการอีกชั้นหนึ่ง
2. ไคลเอนต์ (Client) คือ เครื่องที่ไปขอใช้บริการอย่างใดอย่างหนึ่งจากเซิร์ฟเวอร์ ซึ่งไคลเอนต์เป็นเครื่องคอมพิวเตอร์ที่ไม่สามารถเรียกใช้งานโปรแกรมคอมพิวเตอร์ในตัวเองได้ แต่สามารถใช้งานโปรแกรมนั้นผ่านทางระบบเครือข่าย
3. โปรแกรมเว็บเซิร์ฟเวอร์ (Web Server) คือ ซอฟต์แวร์คอมพิวเตอร์ ซึ่งทำหน้าที่ให้บริการข้อมูล แก่ Client หรือ เครื่องคอมพิวเตอร์ที่ขอรับบริการ ในรูปแบบ สื่อผสม ผ่านระบบเครือข่าย โดยสามารถแสดงผล ผ่านโปรแกรมเว็บเบราว์เซอร์ (Web Browser) หรืออาจกล่าวได้ว่า โปรแกรมเว็บเซิร์ฟเวอร์เป็นโปรแกรมที่คอยให้บริการแก่ไคลเอนต์ที่ร้องขอข้อมูลเข้ามาโดยผ่านเว็บเบราว์เซอร์ พร้อมรองรับการใช้งานจากไคลเอนต์หลาย ๆ เครื่องพร้อมกัน

สำหรับซอฟต์แวร์เว็บเซิร์ฟเวอร์ที่ได้รับความนิยมสูงสุด 4 อันดับแรก คือ

- Apache HTTP Server จาก Apache Software Foundation
- Internet Information Server (IIS) จากไมโครซอฟท์
- Sun Java System Web Server จากซัน ไมโครซิสเต็มส์ (เดิมชื่อ Sun ONE Web Server, iPlanet Web Server และ Netscape Enterprise Server)

- Zeus Web Server จาก Zeus Technology
4. โปรแกรมเท็กซ์เอดิเตอร์ (Text Editor) ซอฟต์แวร์คอมพิวเตอร์ที่ทำหน้าที่สร้างหรือปรับแก้ข้อความ รวมทั้งปรับรูปแบบข้อมูลต่าง ๆ ในการพัฒนาเว็บไซต์โดยเฉพาะ ซึ่งนักเขียนโปรแกรมจะใช้โปรแกรมนี้ช่วยแก้ไขปรับปรุงเว็บไซต์ที่ได้พัฒนาไว้
 5. ภาษาสคริปต์ (Script Language)

ภาษาสคริปต์ที่ใช้ในการสร้างเว็บเพจ แบ่งได้เป็น 2 ประเภท ได้แก่

 - Server-Side Script เช่น PHP, ASP, JSP, CGI เป็นภาษาสคริปต์ที่ประมวลผลที่ฝั่งเซิร์ฟเวอร์ แล้วส่งผลลัพธ์ไปแสดงผลที่ฝั่งไคลเอนต์ผ่านโปรแกรมเว็บเบราว์เซอร์ ซึ่งฝั่งไคลเอนต์ไม่สามารถเห็นโค้ดในส่วนนี้ได้
 - Client-Side Script เช่น JavaScript, VBScript, JScript เป็นภาษาสคริปต์ที่ประมวลผลบนเครื่องคอมพิวเตอร์ของผู้เยี่ยมชมเว็บไซต์หรือเครื่องไคลเอนต์โดยใช้โปรแกรมเว็บเบราว์เซอร์ ซึ่งจะช่วยแบ่งเบาการทำงานให้กับเครื่องเว็บเซิร์ฟเวอร์ได้
 6. โปรแกรมดาต้าเบสเซิร์ฟเวอร์ (Database Server) เป็นซอฟต์แวร์ที่ทำงานบนเซิร์ฟเวอร์ ทำให้เซิร์ฟเวอร์ให้บริการเกี่ยวกับฐานข้อมูลได้ เช่น เปิดให้ผู้ใช้เพิ่มข้อมูล ลบ หรือแก้ไข สำหรับโปรแกรม
 7. โปรแกรมดาต้าเบสเมเนเจอร์ (Database Manager) เป็นซอฟต์แวร์ที่ช่วยอำนวยความสะดวกในการจัดการระบบฐานข้อมูล ทั้งนี้โปรแกรม Database Server บางตัว เช่น MySQL ไม่ได้สร้างส่วนที่จัดการ สร้าง แก้ไข database ทำให้จำเป็นต้องมีผู้ช่วยที่คอยจัดการเกี่ยวกับฐานข้อมูล ซึ่งก็คือ phpMyAdmin

บทที่ 2 โปรแกรม Appserv

ความหมายของโปรแกรม AppServ

AppServ คือโปรแกรมที่รวบรวมเอา Open Source Software หลายๆ อย่างมารวมกัน โดยมี Package หลักดังนี้

- Apache
- PHP
- MySQL
- phpMyAdmin

โปรแกรมต่างๆ ที่นำมารวบรวมไว้ทั้งหมดนี้ ได้ทำการดาวน์โหลดจาก Official Release ทั้งสิ้น โดยตัว AppServ จึงให้ความสำคัญว่าทุกสิ่งทุกอย่างจะต้องให้เหมือนกับต้นฉบับ เราจึงไม่ได้ตัดทอนหรือเพิ่มเติมอะไรที่แปลกไปกว่า Official Release แต่อย่างใด เพียงแต่มีบางส่วนเท่านั้นที่เราได้เพิ่มประสิทธิภาพการติดตั้งให้สอดคล้อง กับการทำงานแต่ละคน โดยที่การเพิ่มประสิทธิภาพไม่ได้ไปยุ่ง ในส่วนของ Original Package เลยแม้แต่น้อยเพียงแค่เป็นการกำหนดค่า Config เท่านั้น เช่น Apache ก็จะเป็นในส่วนของ httpd.conf, PHP ก็จะเป็นในส่วนของ php.ini, MySQL ก็จะเป็นในส่วนของ my.ini ดังนั้นเราจึงรับประกันได้ว่าโปรแกรม AppServ สามารถทำงานและความเสถียรของระบบ ได้เหมือนกับ Official Release ทั้งหมด

จุดประสงค์หลักของการรวบรวม Open Source Software เหล่านี้เพื่อทำให้การติดตั้งโปรแกรมต่างๆ ที่ได้กล่าวมา ให้ง่ายขึ้น เพื่อลดขั้นตอนการติดตั้งที่แสนจะยุ่งยากและใช้เวลานาน โดยผู้ใช้งานเพียงดับเบิลคลิก setup ภายในเวลา 1 นาที ทุกอย่างก็ติดตั้งเสร็จสมบูรณ์ระบบต่างๆ ก็พร้อมที่จะทำงานได้ทันทีทั้ง Web Server, Database Server เหตุผลนี้จึงเป็น เหตุผลหลักที่หลายๆ คนทั่วโลก ได้เลือกใช้โปรแกรม AppServ แทนการที่จะต้องมาติดตั้งโปรแกรมต่างๆ ที่ละส่วน

เพราะไม่ว่าจะเป็นผู้ที่ความชำนาญในการติดตั้ง Apache, PHP, MySQL ก็ไม่ได้เป็นเรื่องง่ายเสมอไป เนื่องจากการ ติดตั้งโปรแกรมที่แยกส่วนเหล่านี้ให้มารวมเป็นชิ้นอันเดียวกัน ก็ใช้เวลาค่อนข้างมากพอสมควร แม้แต่ตัวผู้พัฒนา AppServ เอง ก่อนที่จะ Release แต่ละเวอร์ชันให้ดาวน์โหลด ต้องใช้ระยะเวลาในการติดตั้งไม่น้อยกว่า 2 ชั่วโมง เพื่อทดสอบความ ถูกต้องของระบบ ดังนั้นจึงจะเห็นว่าเราเองนั้นเป็นมือใหม่หรือมือเก่า ย่อมไม่ใช่เรื่องง่ายเลยที่จะติดตั้ง Apache, PHP, MySQL ในพริบตาเดียว

ข้อแตกต่างของ AppServ ในแต่ละเวอร์ชัน

AppServ ได้แบ่งเวอร์ชันออกเป็น 2 ส่วนด้วยกัน คือ

- 2.5.x คือเวอร์ชันที่นำ Package ใหม่ๆ นำมาใช้งานโดยเฉพาะ เหมาะสำหรับนักพัฒนาที่ต้องการระบบใหม่ๆ หรือต้องการทดสอบ ทดลองใช้งานฟังก์ชันใหม่ ซึ่งอาจจะไม่ได้ความเสถียรของระบบได้ 100% เนื่องจากว่า Package จากนักพัฒนานั้นยังอยู่ในช่วงของขั้นทดสอบ ทดลองเพื่อหาข้อผิดพลาดอยู่
- 2.4.x คือเวอร์ชันที่นำ Package ที่มีความเสถียรเป็นหลัก เหมาะสำหรับผู้ที่ต้องการความมั่นคงของระบบ โดยไม่ได้มุ่งเน้นที่จะใช้ฟังก์ชันใหม่

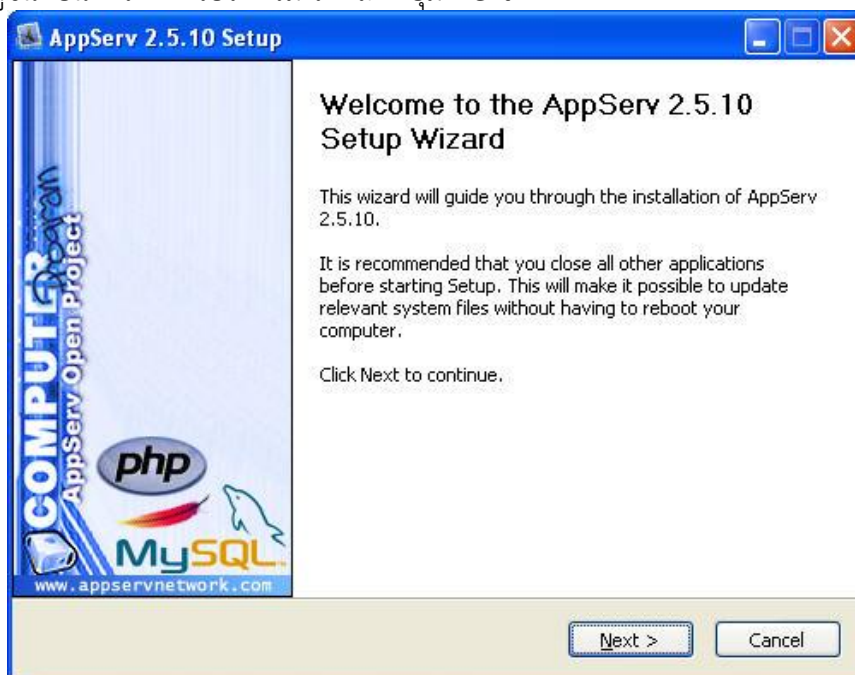
ขั้นตอนการติดตั้งโปรแกรม Appserv

ก่อนทำการติดตั้งโปรแกรม Appserv จะต้องทำการตรวจสอบความต้องการขั้นต่ำของระบบคอมพิวเตอร์ที่ใช้ติดตั้งโปรแกรม

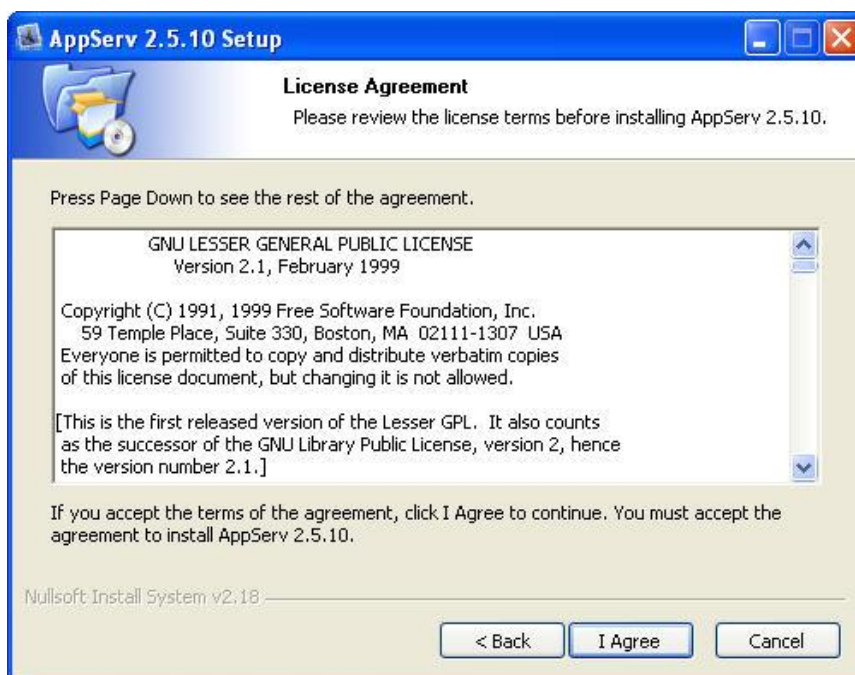
1. ดาวน์โหลดโปรแกรม AppServ จากเว็บไซต์ <http://www.appservnetwork.com> หลังจากที่ดาวน์โหลดโปรแกรมเสร็จเรียบร้อยแล้ว ดับเบิลคลิกที่ไอคอน Appserv เพื่อติดตั้งโปรแกรม



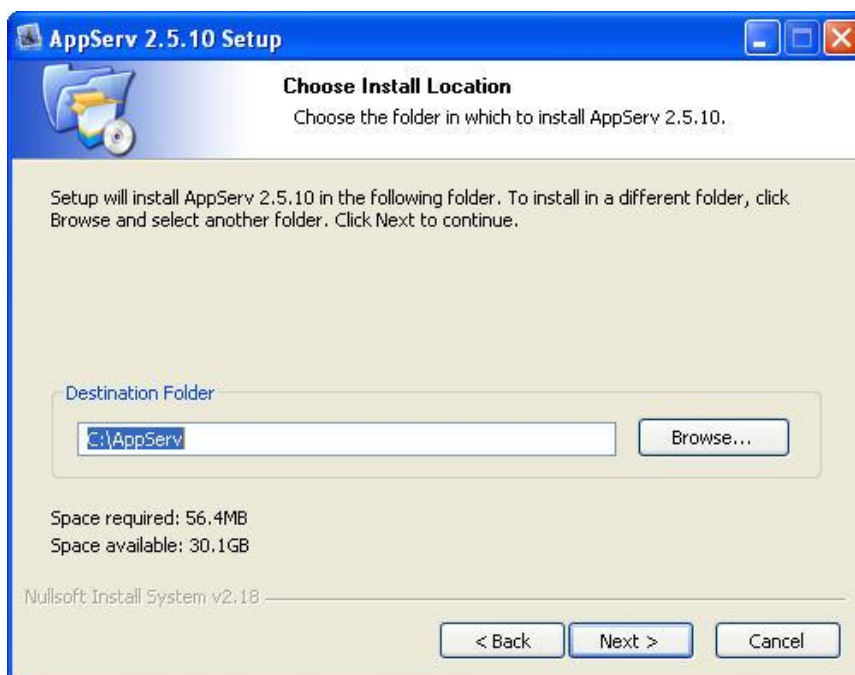
2. เข้าสู่ขั้นตอนการติดตั้งโปรแกรม ให้คลิกที่ปุ่ม Next



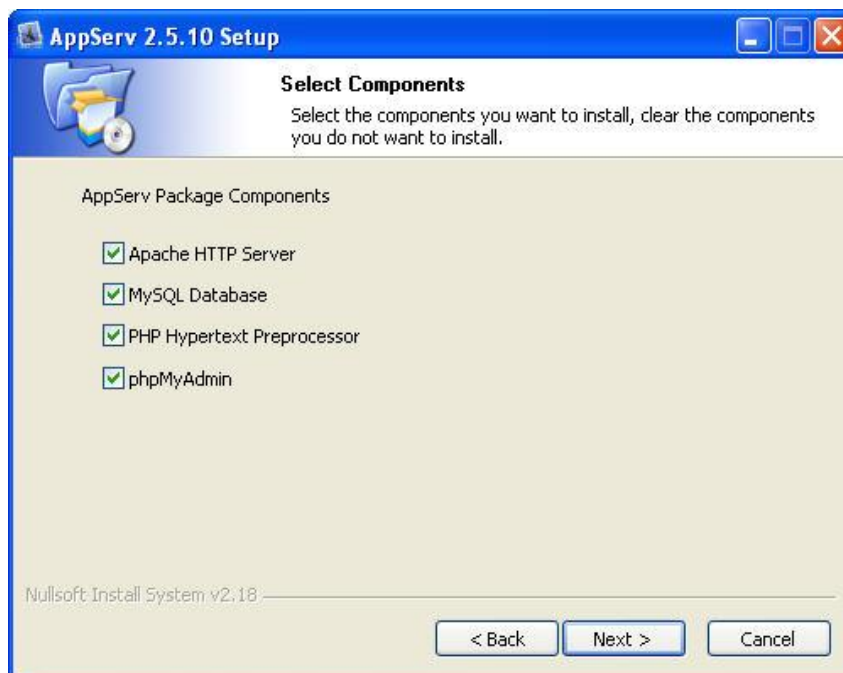
3. แสดงข้อตกลง License Agreement คลิกที่ปุ่ม I Agree เพื่อยอมรับเงื่อนไขการติดตั้ง



4. ชุดติดตั้งโปรแกรมแสดง Directory ที่จะติดตั้งโปรแกรม (ซึ่งผู้ลงโปรแกรมไม่จำเป็นต้องเปลี่ยน Directory) คลิกปุ่ม Next



5. แสดง Package ของโปรแกรมที่จะติดตั้ง ให้เลือกองค์ประกอบ (Components) สำหรับการติดตั้ง




• แสดง package ที่จะติดตั้ง

Apache	ทำหน้าที่เป็น web server
MySQL	ทำหน้าที่เป็นฐานข้อมูล
PHP	ทำหน้าที่เป็น script ติดต่อกับฐานข้อมูล
phpMyAdmin	ทำหน้าที่เป็นโปรแกรมช่วยจัดการฐานข้อมูล ให้ง่ายขึ้น

6. จะปรากฏหน้าจอสำหรับให้กรอกข้อมูลเซิร์ฟเวอร์ของ (Server Information) ซึ่งประกอบด้วย

- ชื่อเซิร์ฟเวอร์ หรือ ยูอาร์แอล (URL)
- อีเมลล์ของผู้ดูแลเซิร์ฟเวอร์
- พอร์ตสำหรับใช้งาน หรือติดต่อ



AppServ 2.5.10 Setup

Apache HTTP Server Information
Please enter your server's information.

Server Name (e.g. www.appservnetwork.com)

Administrator's Email Address (e.g. webmaster@gmail.com)

Apache HTTP Port (Default : 80)


Nullsoft Install System v2.18

< Back Next > Cancel

.7ขั้นตอนต่อไปคือการกำหนดค่าสำหรับ MySQL Server ซึ่งต้องระบุ

- รหัสผ่าน (Password) สำหรับ root
- ชุดภาษา (Character Sets and Collations) ที่ใช้

หลังจากกำหนดค่าดังกล่าวแล้ว ให้กดปุ่ม **Install**



AppServ 2.5.10 Setup

MySQL Server Configuration
Configure the MySQL Server instance.

Please enter Root password for MySQL Server.

Enter root password

Re-enter root password

MySQL Server Setting

Character Sets and Collations

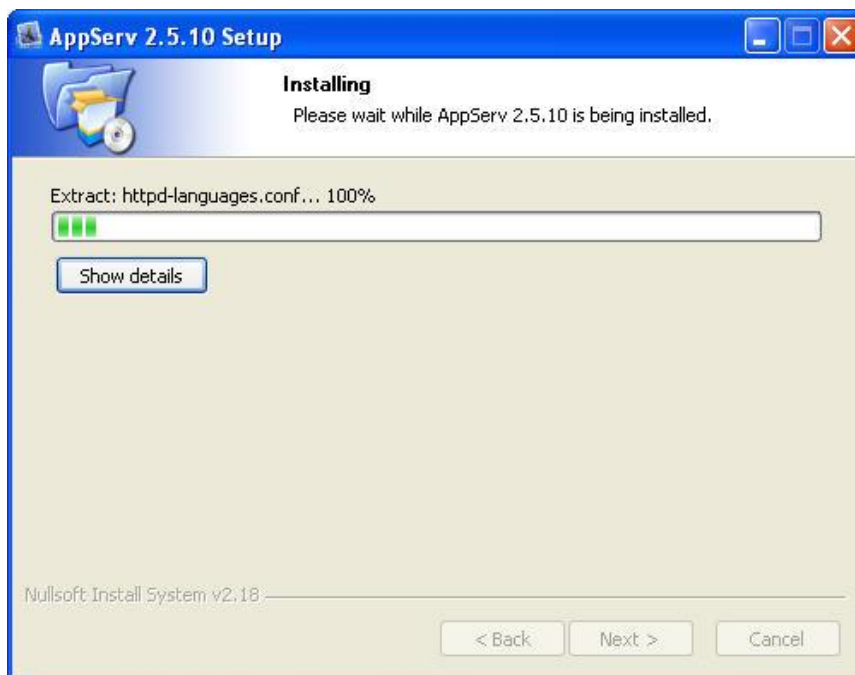
☐ Old Password Support (PHP MySQL API function.)

☐ Enable InnoDB

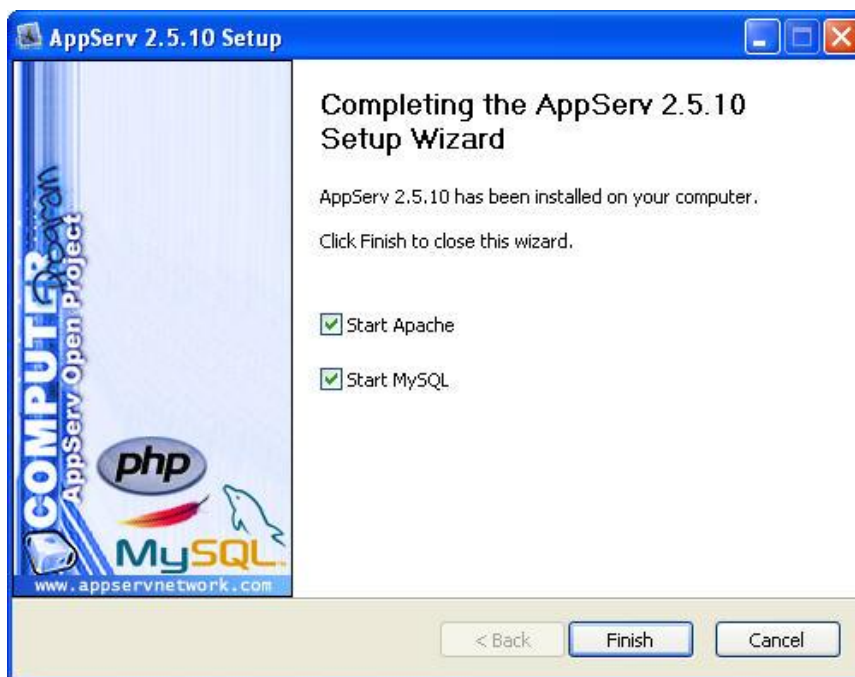
Nullsoft Install System v2.18

< Back Install Cancel

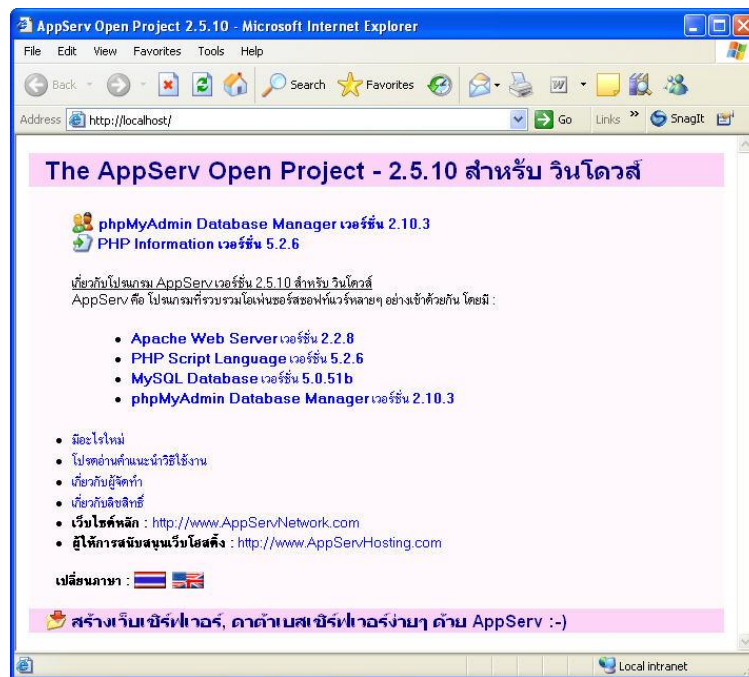
8. หลังกำหนดค่าสำหรับ MySQL Server แล้ว ตัวติดตั้งจะดำเนินการติดตั้งองค์ประกอบต่างๆ ลงในระบบ



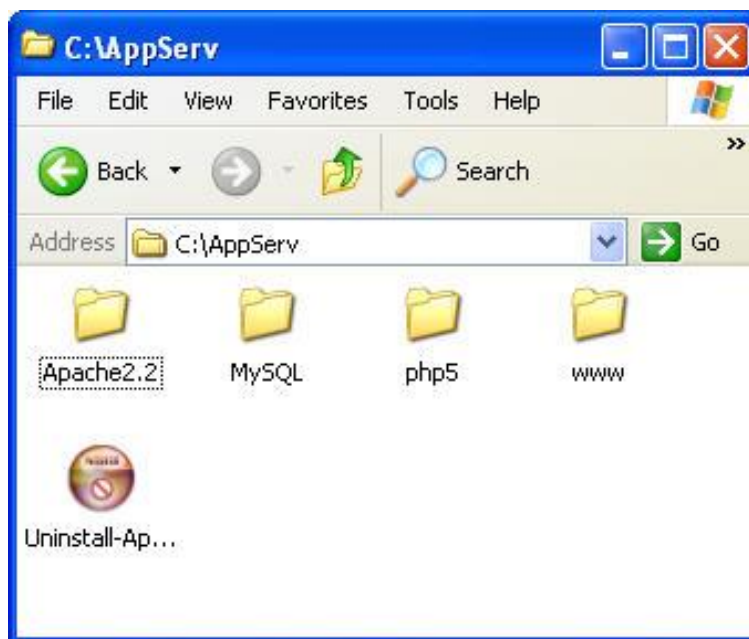
9. เมื่อการติดตั้งเสร็จสิ้น ให้กดปุ่ม Finish



10. ทดสอบการใช้งานโปรแกรม Appserv โดยการเปิดโปรแกรมบราวเซอร์ แล้วพิมพ์ URL ในช่อง Address <http://localhost> หรือ 127.0.0.1 หากโปรแกรมสามารถใช้งานได้จะแสดงดังรูป



11. สำหรับองค์ประกอบต่างๆ ของ AppServ จะถูกติดตั้งไว้ที่โฟลเดอร์ C:\AppServ สำหรับไฟล์ภาษา PHP ที่เขียนขึ้น ต้องถูกบันทึกในโฟลเดอร์ www เท่านั้น



บทที่ 3 ความรู้เบื้องต้นเกี่ยวกับภาษา PHP

ภาษาพีเอชพีเป็นภาษาสำหรับใช้ในการเขียนโปรแกรมบนเว็บไซต์ สามารถเขียนได้หลากหลายโปรแกรม เช่นเดียวกับภาษาทั่วไป ภาษาพีเอชพีมีความแตกต่างจากภาษาเอชทีเอ็มแอล (HTML) อย่างสิ้นเชิงเพราะว่าภาษาเอชทีเอ็มแอลนั้นเป็นภาษาที่ใช้ในการจัดรูปแบบของเว็บไซต์ จัดตำแหน่งรูป จัดรูปแบบตัวอักษร หรือใส่สีให้กับเว็บไซต์ของเราซึ่งมีการทำงานแบบสแตติกเว็บไซต์ (Static Website) แต่ภาษาพีเอชพีนั้นเป็นภาษาที่ใช้ในการคำนวณ ประมวลผล เก็บค่า และทำตามคำสั่งต่างๆ อย่างเช่น รับค่าจากแบบฟอร์มหน้าเว็บที่เราทำ รับค่าจากช่องคำตอบของเว็บบอร์ดและเก็บไว้เพื่อนำมาแสดงผลต่อไป ซึ่งเว็บไซต์ที่พัฒนาด้วยภาษาพีเอชพี เป็นเว็บไซต์ที่สามารถโต้ตอบกับผู้ใช้กันได้ หรือที่เรียกว่าไดนามิกเว็บไซต์ (Dynamic Website) อาจกล่าวได้ว่า เว็บไซต์ต้องมีภาษาพีเอชพีเพื่อเป็นส่วนในการประมวลผลคำสั่งและควบคุมการทำงานของเว็บไซต์ และมีส่วนของภาษาเอชทีเอ็มแอล หรือ จาวาสคริปต์ (JavaScript) ใช้เป็นเพียงแค่ตัวควบคุมการแสดงผลเท่านั้น

การกำเนิดภาษา PHP แต่ละเวอร์ชัน

PHP เกิดในปี 1994 โดย Rasmus Lerdorf โปรแกรมเมอร์อเมริกันได้คิดค้นสร้างเครื่องมือที่ใช้ในการพัฒนาเว็บไซต์ส่วนตัวของเขา โดยใช้ชื่อของภาษา C และ Perl เรียกว่า Personal Home Page และได้สร้างส่วนติดต่อกับฐานข้อมูลที่ชื่อว่า Form Interpreter (FI) รวมทั้งสองส่วน เรียกว่า PHP/FI ซึ่งก็เป็นจุดเริ่มต้นของ ภาษาพีเอชพี หลังจากได้พัฒนาเว็บไซต์ขึ้นมาก็มีผู้สนใจเข้ามาเยี่ยมชมเว็บไซต์ของ Rasmus Lerdorf และมีความรู้สึกชื่นชอบจึงติดต่อเพื่อนำโค้ดนั้นไปใช้และนำไปพัฒนาต่อ ในลักษณะของ Open Source ภายหลังมีความนิยมขึ้นเป็นอย่างมาก เพียงระยะเวลาภายใน 3 ปีมีเว็บไซต์ที่พัฒนาด้วย PHP/FI ในการติดต่อฐานข้อมูลและแสดงผลแบบไดนามิกก็เพิ่มมากขึ้นเรื่อยและมีจำนวนมากกว่า 50,000 เว็บไซต์

PHP2 (ในตอนนั้นใช้ชื่อว่า PHP/FI) ในช่วงระหว่าง 1995-1997 Rasmus Lerdorf ได้มีผู้ที่มาช่วยพัฒนาอีก 2 คน คือ Zeev Suraski และ Andi Gutmans ชาวอิสราเอล ซึ่งปรับปรุงโค้ดของ Lerdorf ใหม่โดยใช้ C++ ให้มีความสามารถจัดการเกี่ยวกับแบบฟอร์มข้อมูลที่ถูกสร้างมาจากภาษาเอชทีเอ็มแอลและสนับสนุนการติดต่อกับโปรแกรมจัดการฐานข้อมูล mSQL จึงทำให้ภาษาพีเอชพี เริ่มถูกใช้มากขึ้นอย่างรวดเร็ว และเริ่มมีผู้สนับสนุนการใช้งานของภาษาพีเอชพีมากขึ้น โดยในปลายปี 1996 ภาษาพีเอชพีถูกนำไปใช้ประมาณ 15,000 เว็บไซต์ทั่วโลก และเพิ่มจำนวนขึ้นเรื่อยๆ ต่อมาก็มีผู้เข้ามาช่วยพัฒนาอีก 3 คน คือ Stig Bakken รับผิดชอบความสามารถในการติดต่อ Oracle, Shane Caraveo รับผิดชอบดูแลภาษาพีเอชพีบน Window 9x/NT, และ Jim Winstead รับผิดชอบการตรวจความบกพร่องต่างๆ และได้เปลี่ยนชื่อเป็น Professional Home Page ในเวอร์ชันที่ 2

PHP3 ออกมาในช่วงระหว่างเดือน มิถุนายน 1997 ถึง 1999 ได้ออกสู่สายตาของนักโปรแกรมเมอร์ มีคุณสมบัติเด่นคือสนับสนุนระบบปฏิบัติการทั้ง Window 95/98/ME/NT, Linux และเว็บเซิร์ฟเวอร์ อย่าง IIS, PWS, Apache, OmniHTTPd สนับสนุน ฐานข้อมูลได้หลายรูปแบบเช่น SQL Server, MySQL, mSQL, Oracle, Informix, ODBC

PHP4 ตั้งแต่ 1999 - 2007 ซึ่งได้เพิ่ม Functions การทำงานในด้านต่างๆให้มากและง่ายขึ้นโดย บริษัท Zend ซึ่งมี Zeev และ Andi Gutmans ได้ร่วมก่อตั้งขึ้น (<http://www.zend.com>) ในเวอร์ชันนี้จะเป็น compile script ซึ่งในเวอร์ชันหน้าจะเป็น embed script interpreter ในปัจจุบันมีคนได้ใช้ PHP สูงกว่า 5,100,000 ไซต์ แล้วทั่วโลก และ ผู้พัฒนา

ได้ตั้งชื่อของ PHP ใหม่ว่า PHP: Hypertext Preprocessor ซึ่งหมายถึงมีประสิทธิภาพระดับโปรเฟสเซอร์สำหรับไฮเปอร์เท็กซ์

PHP5 ตั้งแต่ 2007-ปัจจุบัน มี ได้เพิ่ม Functions การทำงานในด้านต่าง ๆ เช่น

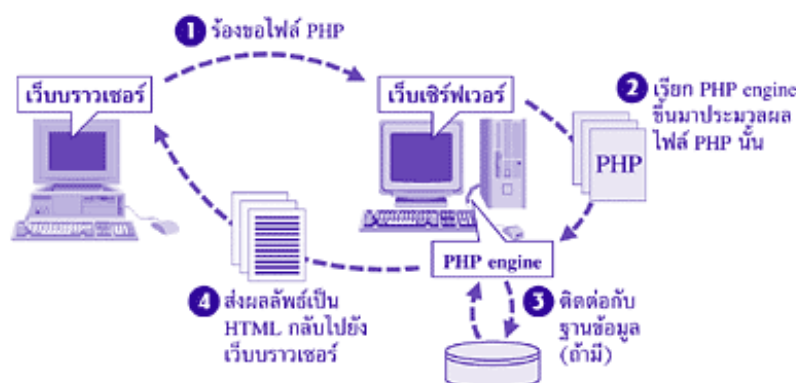
- * Object Oriented Model
- * การกำหนดสโคป public/private/protected
- * Exception handling
- * XML และ Web Service
- * MySQLi และ SQLite
- * Zend Engine 2.0

รายชื่อของนักพัฒนาภาษาพีเอชพี ที่เป็นแก่นสำคัญในปัจจุบันมีดังต่อไปนี้

- * Zeev Suraski, Israel
- * Andi Gutmans, Israel
- * Shane Caraveo, Florida USA
- * Stig Bakken, Norway
- * Andrey Zmievski, Nebraska USA
- * Sascha Schumann, Dortmund, Germany
- * Thies C. Arntzen, Hamburg, Germany
- * Jim Winstead, Los Angeles, USA
- * Rasmus Lerdorf, North Carolina, USA

การทำงานของภาษาพีเอชพี

การทำงานของภาษาพีเอชพี เริ่มจากฝั่งไคลเอ็นต์เปิดเว็บเบราว์เซอร์เพื่อร้องขอไฟล์พีเอชพี โปรแกรมเว็บเบราว์เซอร์จะร้องขอไฟล์พีเอชพีที่เก็บในเครื่องเซิร์ฟเวอร์ จากนั้นฝั่งเซิร์ฟเวอร์จะทำการค้นหาไฟล์พีเอชพีและเรียกใช้งาน PHP engine



เพื่อแปลงไฟล์พีเอชพีและนำข้อมูลในฐานข้อมูลมาใช้ร่วมกับประมวลผลไฟล์พีเอชพี และส่งผลลัพธ์จากการประมวลผลไปให้เครื่องไคลเอ็นต์ โดยการแปลงผลลัพธ์ที่ได้จากการประมวลผลเป็นภาษาเอชทีเอ็มแอลส่งกลับไปยังเว็บเบราว์เซอร์

โครงสร้างภาษาพีเอชพี

ภาษาพีเอชพี มีลักษณะเป็น embedded script หมายความว่าเราสามารถฝังคำสั่งพีเอชพีไว้ในเว็บเพจร่วมกับคำสั่ง (Tag) ของเอชทีเอ็มแอลได้ และสร้างไฟล์ที่มีนามสกุลเป็น .php, .php3 หรือ .php4 ซึ่งไวยากรณ์ที่ใช้ใน PHP เป็นการนำรูปแบบของภาษาต่างๆ มารวมกันได้แก่ C, Perl และ Java

คำสั่ง PHP ที่อยู่ภายในเอกสาร HTML จึงได้มีการกำหนดสัญลักษณ์ไว้ดังนี้ ซึ่งสามารถทำได้หลายรูปแบบ เช่น

แบบที่ 1

<? ... ?> (SGML style)

ตัวอย่างแบบที่ 1

```
1 <?
2 echo "PHP Hello World!!";
3 ?>
```

แบบที่ 2

<?php ... ?> (XML style)

ตัวอย่างแบบที่ 2

```
1 <?php
2 echo "PHP Hello World!!";
3 ?>
```

แบบที่ 3

<script language="php"> ... </script> (JavaScript style)

ตัวอย่างแบบที่ 3

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

การแทรกคำสั่งภาษา PHP ในเอกสาร HTML

ที่นิยมก็คือแบบแรก โดยเริ่มต้นด้วย `<? และจบด้วย ?>` และตรงกลางจะเป็นคำสั่งในภาษา PHP เราสามารถวางคำสั่ง PHP ไว้ภายในเอกสาร HTML ตามที่ต้องการได้ อาจจะสลับกับ Tag ของภาษา HTML ก็ได้ ตัวอย่างเช่น

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/h
5  <title>Untitled Document</title>
6  </head>
7  ข้อความทั่วไปส่วนของ Html<br>
8  <? echo"ส่วนของ php"; ?><br>
9  ส่วนของ Html<br>
10 <?php echo"ตรงนี้ก็สามารรถแทรก Script php "; ?>
11 </body>
12 </html>

```

คำสั่งแรกที่ย่างที่สุดสำหรับการเรียนรู้ ก็คือคำสั่ง echo แล้วตามด้วยข้อความหรือสตริงค์ (string) ข้อความในภาษา PHP จะเริ่มต้นและจบด้วย double quote (") เหมือนในภาษาซี และคำสั่งแต่ละคำสั่งในภาษา PHP จะจบท้ายคำสั่งด้วย semicolon (;) เหมือนในภาษาซี ซึ่งคำสั่งหรือฟังก์ชันในภาษา PHP นั้นจะเขียนด้วยตัวพิมพ์เล็กหรือใหญ่

คำอธิบาย(หมายเหตุ)ในภาษา PHP (Comment)

การใส่หมายเหตุ (Comment) ในสคริปต์ PHP สามารถทำได้สองวิธี คือ การใส่หมายเหตุเฉพาะบรรทัด และการใส่หมายเหตุ แบบหลาย ๆ บรรทัด ซึ่งสามารถแบ่งรูปแบบการคอมเม้นท์ได้สามรูปแบบด้วยกัน

การใส่หมายเหตุ วิธีที่ 1 ใช้เครื่องหมาย # หรือ //

ข้อความที่ตามหลังเครื่องหมายนี้ จนถึงตัวอักษรตัวสุดท้ายของบรรทัดนั้น ๆ จะไม่ถูกแสดงผลออกทางหน้าจอ

ตัวอย่างการเขียน

```

1. <?php
2.
3. $a = "worldphp.com"; # assign value ให้กับตัวแปร $a
4. echo $a; // แสดงผล ตัวแปร $a
5. ?>

```

การใส่หมายเหตุ วิธีที่ 2 ใช้เครื่องหมาย /* ร่วมกับ */

การเขียนโปรแกรมบนเว็บด้วยภาษา PHP (ฉบับพื้นฐาน)

อ.ชนิดา แก้วเพชร

การใส่หมายเหตุแบบนี้เป็นการใส่แบบเป็นช่วงของข้อมูล โดยสามารถควบคุมบริเวณที่ไม่ต้องการให้แสดงผลออกทางจอภาพได้มากกว่า 1 บรรทัด

ตัวอย่างการเขียน

```

1.  <?php
2.  $a = "worldphp.com";
3.  /*
4.  $b = "www.worldphp.com";
5.  $c = "worldphp";
6.  */
7.
8.  // แสดงผล ตัวแปร $a
9.  echo $a;
10. ?>

```

ทดลองเขียนภาษา PHP ครั้งแรก

การ Set ค่า Editpuls รัน PHP

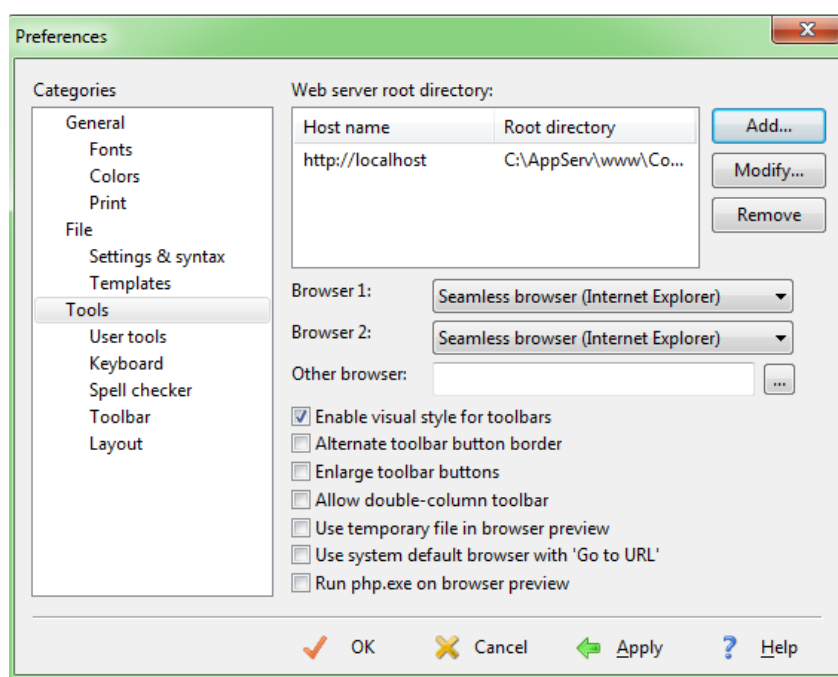
วิธีการ

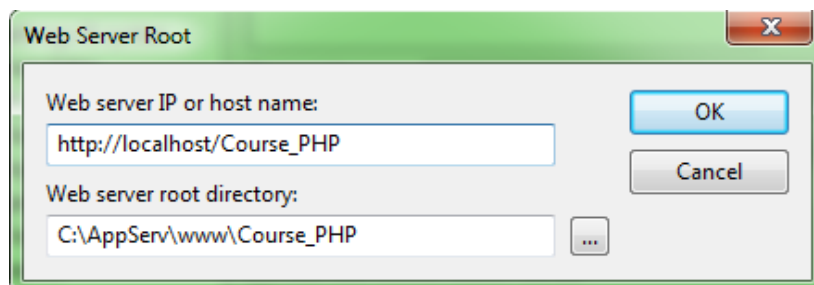
1. เลือก Tools ที่แถบเครื่องมือ เลือก Configure User Tools
2. เลือก คำว่า Tools แล้วเลือก Add

Web server IP or hostname ให้พิมพ์ <http://localhost/> ชื่อโพล์เตอร์สำหรับเก็บไฟล์ PHP หรือ <http://127.0.0.1/> ชื่อโพล์เตอร์สำหรับเก็บไฟล์ PHP

Web server root directory ให้เลือก C:\Appserv\www\ ชื่อโพล์เตอร์สำหรับเก็บไฟล์ PHP


3. คลิก OK

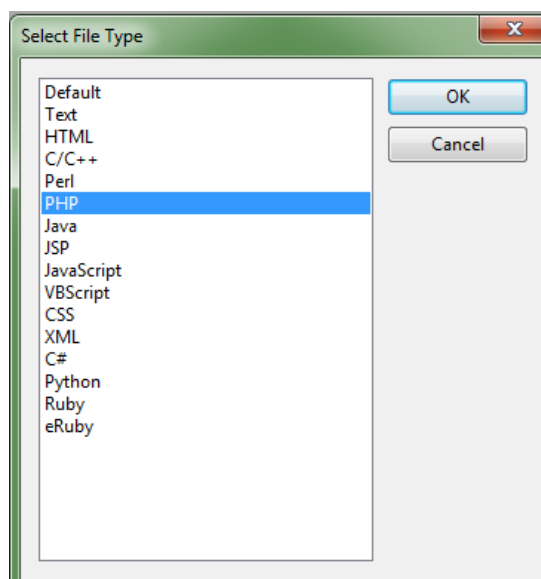
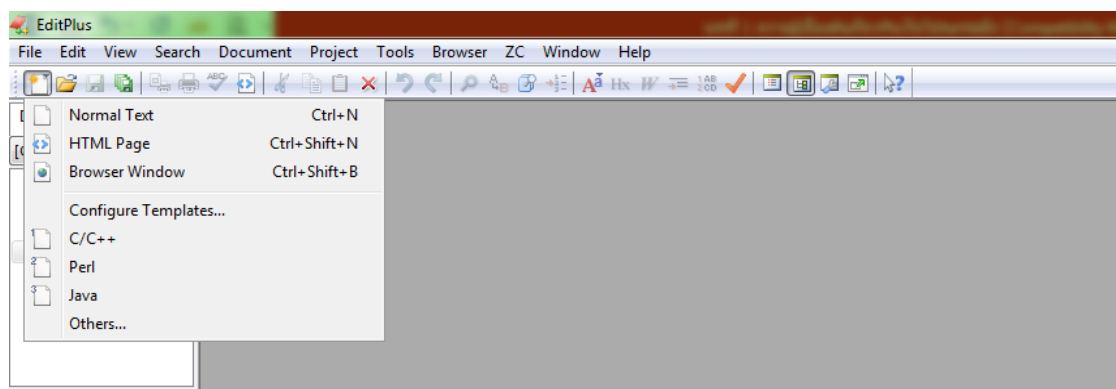




หมายเหตุ คีย์ร้อนสำหรับรัน php บน Editplus คือ Ctrl+B

ทดลองเขียนสคริปต์ภาษาพีเอชพี

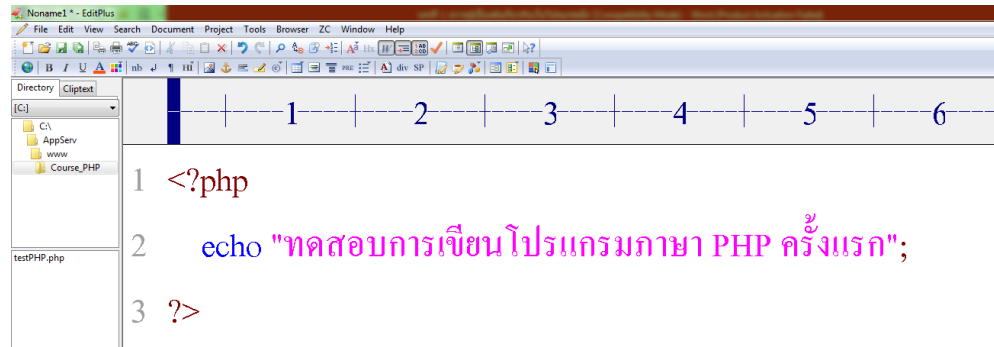
1. เปิดโปรแกรม EditPlus เลือก ไอคอน  คลิกเลือก Other... เลือก PHP



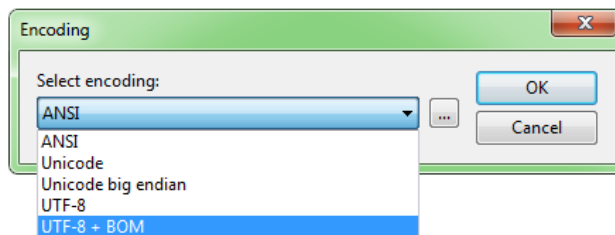
การเขียนโปรแกรมบนเว็บด้วยภาษา PHP (ฉบับพื้นฐาน)

อ.ชนิดา แก้วเพชร

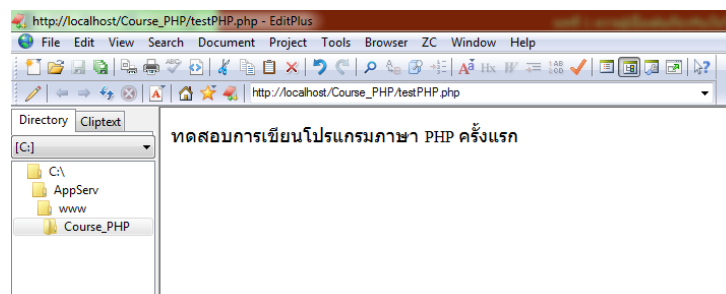
2. ทดลองเขียนสคริปต์ภาษาพีเอชพี และ บันทึกไฟล์ลง C:\Appserv\www\ชื่อไฟล์เดอร์สำหรับเก็บไฟล์ PHP



3. หลังจากนั้นเลือก UTF-8+BOM เพื่อรองรับภาษาไทย



4. ทดลองรันโปรแกรมโดยการกดปุ่ม Ctrl+B



บทที่ 4 ชนิดข้อมูล ตัวแปรและตัวดำเนินการในภาษาพีเอชพี

ชนิดข้อมูล (Data Type)

การสร้างตัวแปรในภาษา PHP ผู้เขียนโปรแกรมไม่จำเป็นต้องประกาศชนิดของตัวแปรนั้นๆ เหมือนการเขียนโปรแกรมในภาษาอื่นๆ เพียงแค่กำหนดค่าให้กับตัวแปร ตัวแปรนั้นๆ ก็จะถูกประกาศ (cast) ให้เป็นชนิดตามข้อมูลที่ได้กำหนดเข้าไป ภาษา PHP พัฒนามาจากภาษา C จึงมีชนิดข้อมูลที่คล้ายกัน แต่ได้ตัดชนิดข้อมูลที่ซ้ำซ้อนกันออกไป

• ชนิดข้อมูลกลุ่มแรก สามารถเก็บได้เพียงข้อมูลเดียว (Scalar Datatypes) เป็นชนิดข้อมูลที่เก็บค่าไว้ได้เพียงชนิดเดียว มีทั้งหมด 4 ชนิดคือ

Boolean เก็บข้อมูลเพียง 2 ค่าคือ true กับ false เท่านั้น ใช้ในคำสั่งเปรียบเทียบ เพื่อเลือกเส้นทางการทำงานของโปรแกรม และตัวแปรทุกๆ ตัว ถือว่าเป็นตัวแปรชนิด Boolean ด้วยเหมือนกัน ถ้าหากนำไปใช้ในคำสั่งเปรียบเทียบ จะได้ค่าที่เป็น true กลับมาทั้งหมด ยกเว้นตัวแปรที่เก็บค่า 0 เท่านั้น จะได้ค่าเป็น false

```

1  <?php
2      $foo = true;    // จริง
3      $foo = false;   // เท็จ
4      $foo = 1;       // จริง
5      $foo = -1;      // จริง
6      $foo = "hello"; // จริง
7      $foo = 0;       // เท็จ
8
9      // เพื่อนๆ ว่าโค้ดข้างล่างนี้ จะรันได้หรือไม่ และผลลัพธ์จะเป็นอย่างไรครับ?
10     $foo = "hello";
11     if ($foo) {
12         echo "TRUE";
13     }
14     else {
15         echo "FALSE";
16     }
17  ?>

```

Integer หรือ จำนวนเต็ม ถ้าหากกำหนดข้อมูลที่มีเลขทศนิยม ตัวแปรนั้นๆ จะถูกประกาศให้เป็น Float โดยอัตโนมัติ ตัวแปรชนิดจำนวนเต็มสามารถเก็บข้อมูลได้ทั้งบวกและลบตามเลขฐาน 3 ชนิดคือ

- ฐานสิบ (Decimal) กำหนดตัวเลขเข้าไปตรงๆ เลย เป็นเลขฐานที่คนคุ้นเคยมากที่สุด
- ฐานแปด (Octal) เติมเลข 0 เข้าไปข้างหน้าตัวเลขชุดนั้นก่อน
- ฐานสิบหก (Hexadecimal) เติม 0x เข้าไปข้างหน้าตัวเลขชุดนั้น

```

1  <?php
2      42 // decimal
3      -234034 // decimal
4      0436 // octal
5      0x7ABF // hexadecimal
6      123e10 // decimal
7  ?>

```

Float หรือ Double หรือจำนวนจริง ตัวเลขทุกตัวที่มีทศนิยมไม่ว่าจะเป็นบวกหรือลบจะถือว่าเป็นจำนวนจริงทั้งหมด

```
1 <?php
2 // ทุกตัวเป็น Float
3 1.23456
4 4.00
5 11.3e4
6 11.22e+20
7 ?>
```

String เป็น ข้อความ หรือ ตัวอักษร ในภาษา PHP ไม่มีตัวแปรแบบ char เพราะฉะนั้น ข้อความทุกข้อความจะถือว่าเป็น string ทั้งหมด ตัวแปร string จะถูกกำหนดอยู่ภายใน ' (single quote) หรือ " (double quote) ก็ได้

```
1 <?php
2 // ทุกตัวเป็น string หมด
3 "hello world"
4 'cmdevhub\n'
5 "123&%$abc^"
6 ?>
```

นอกจากนี้เรายังสามารถเข้าถึงตัวอักษรภายใน string ได้โดยมองตัวแปร string นั้นเป็น array ตัวหนึ่ง และเข้าถึงข้อมูลภายในโดยการอ้างตัวแปรแบบ array

```
1 <?php
2 $str = "www.cmdevhub.com";
3 echo $str[0]; // ได้ตัว w
4 echo $str[6]; // ได้ตัว d
5 ?>
```

• ชนิดข้อมูลกลุ่มที่สอง เก็บข้อมูลได้หลายๆ แบบในตัวเดียวกัน (Compound Datatypes) เป็นตัวแปรที่เก็บข้อมูลได้หลายๆ ค่าในตัวแปรตัวเดียวมี 2 ชนิดคือ

Array (อาเรย์) เป็นชุดข้อมูลที่มีค่าหลายๆ ค่า ในตัวเดียว โดยไม่จำเป็นต้องเป็นข้อมูลชนิดเดียวกัน หมายความว่าเราสามารถเก็บข้อมูล Integer, Float, String ไว้ในตัวแปรตัวเดียวกันได้ โดยใช้เครื่องหมาย [...] และมี key เป็นตัวแยกชุดข้อมูลนั้นๆ

ตัวแปรอาเรย์ใน PHP จะแบ่งเป็น 2 ชนิดใหญ่ๆ คือ อาเรย์ธรรมดา และ อาเรย์แบบจับคู่ (Associative Array)

- อาเรย์ธรรมดา ก็คืออาเรย์แบบทั่วๆ จะอ้างอิงโดยใช้ key เป็นหมายเลขตาม index ตัวแปรอาเรย์เริ่มจากเลข 0 ไปจนหมดข้อมูลอาเรย์ ตัวอย่างการสร้างอาเรย์

```

1  <?php
2      $a = array(); // ประกาศตัวแปรว่าเป็น อาร์เรย์
3      $a[0] = "PHP";
4      $a[1] = "ASP";
5      $a[2] = "JSP";
6      $a[3] = "Python";
7      $a[4] = "Ruby";
8
9      // เราสามารถเข้าถึงอาร์เรย์ได้โดยใช้หมายเลข index เป็นตัวอ้างอิง
10
11     echo $a[0]; // ผลลัพธ์ PHP
12     echo $a[4]; // ผลลัพธ์ Ruby
13 ?>

```

ใน PHP เราสามารถกำหนดค่าในอาร์เรย์โดยไม่ต้องกำหนด index ก็ได้ เช่น

```

1  <?php
2      $a = array(); // ประกาศตัวแปรอาร์เรย์
3      $a[] = "PHP";
4      $a[] = "ASP";
5      $a[] = "JSP";
6
7      // ตัวแปร $a จะเก็บค่าโดยเริ่มจากหมายเลข index 0 เป็นต้นไป และสร้างหมายเลข
8      index ขึ้นมาโดยอัตโนมัติ
9
10     echo $a[2]; // ผลลัพธ์ JSP
11 ?>

```

- อาร์เรย์แบบจับคู่ หรือ Associative Array ตามปกติ key ที่ใช้อ้างอิงอาร์เรย์จะเป็นตัวเลข 0 ไปจนถึงตัวสุดท้ายของอาร์เรย์ แต่เราสามารถใช้ข้อความมาเป็น key เพื่ออ้างอิงค่าในอาร์เรย์นั้นๆ ได้ ตัวอย่างการสร้างอาร์เรย์

```

1  <?php
2  echo "สร้างข้อมูลชนิด Associative Array เพื่อเก็บข้อมูลส่วนตัว<br>";
3  $MyRec["name"]="zealot13";
4  $MyRec["age"]=25;
5  $MyRec["gender"]="Male";
6  echo "ชื่อ " . $MyRec["name"] . "<br>";
7  echo "ขณะนี้อายุ " . $MyRec["age"] . " ปี, เพศ " . $MyRec["gender"];
8  ?>

```

ตัวแปร(Variable)

การกำหนดและใช้ตัวแปร (variable) ในภาษา PHP จะเหมือนกับในภาษา Perl คือเริ่มต้นด้วยเครื่องหมาย dollar (\$) และไม่จำเป็นต้องกำหนดชนิดของข้อมูล (data type) อย่างเจาะจงเหมือนในภาษาซี เพราะว่า ตัวแปรภาษาจะจำแนกเองโดยอัตโนมัติว่า ตัวแปรดังกล่าว ใช้ข้อมูลแบบใด ในช่วงเวลานั้นๆ เช่น ข้อความ จำนวนเต็ม จำนวนที่มีเลขจุด ทศนิยม ตรรก เป็นต้น

1. การประกาศตัวแปร

รูปแบบ \$var = value;

คำอธิบาย

\$var = ชื่อตัวแปร

value= ค่าของตัวแปร

ตัวอย่าง

```

1  <?php
2      $a = 1234; // Integer หรือ จำนวนเต็ม
3      $b = 1234.567; # Float หรือ Double
4      $c = "Weerachai Nukitram"; // String เป็น ข้อความ หรือ ตัวอักษร
5      $d[0] = "PHP"; // Array 1 มิติ Index[0]
6      $d[1] = "ASP"; // Array 1 มิติ Index[1]
7      $e[name];="Wanida"; //Associative Array
8      $f = true; // boolean
9  ?>
```

2. หลักการตั้งชื่อตัวแปร

- เริ่มต้นด้วยเครื่องหมาย \$ แล้วตามด้วยตัวอักษร A-Z,a-z หรือเครื่องหมาย_ (Underscore)
- มีความยาวไม่เกิน 255 ตัวอักษร
- ห้ามขึ้นต้นด้วยตัวเลข แต่สามารถใช้ตัวเลขประกอบในชื่อตัวแปรได้
- ห้ามมีจุดทศนิยม ช่องว่าง หรือเว้นวรรค ภายในชื่อตัวแปร
- ห้ามใช้สัญลักษณ์ทางคณิตศาสตร์หรือสัญลักษณ์พิเศษประกอบเป็นชื่อตัวแปร
- ห้ามตั้งชื่อตัวแปรตรงกับคำสั่ง เช่น \$if, \$else, \$do เป็นต้น
- การใช้ตัวอักษรพิมพ์เล็กกับตัวอักษรพิมพ์ใหญ่ แม้จะสื่อถึงคำๆ เดียวกัน แต่ถือว่าตัวแปรนั้น เป็นตัวแปรคนละตัว ที่เรียกว่า Case-Sensitive
- การตั้งชื่อตัวแปรหากตั้งชื่อเดียวกันทุกประการ ตัวแปรที่สร้างใหม่นั้นจะทับค่าตัวแปรเดิม

ตัวดำเนินการ (Operator)

ตัวดำเนินการ (Operators) เป็นสัญลักษณ์ที่ใช้กำหนดรูปแบบการประมวลผลข้อมูล ซึ่งแบ่งได้หลายประเภท เช่น ตัวดำเนินการเกี่ยวกับการคำนวณทางคณิตศาสตร์ การเปรียบเทียบ และการประมวลผลทางตรรกะ (Logic) เป็นต้น

นิพจน์ (Expressions) คือ การกระทำเพื่อให้ได้ผลลัพธ์ค่าหนึ่งค่า ประกอบไปด้วยตัวถูกกระทำ (Operands) และ ตัวดำเนินการ (Operators) เขียนเรียงกันไป เช่น $3 * 2 - 1 + 7$ หรือ $a * 5$ เป็นต้น

1. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator) เป็นตัวดำเนินการที่ใช้ดำเนินการกับตัวเลขมีดังนี้

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	ผลลัพธ์
+	บวก (Add)	$7+3$	10
-	ลบ (Minus)	$8-4$	4
*	คูณ (Multiply)	$6*2$	12
/	หาร (Divide)	$5/2$	2.5
%	หารเอาเศษ (Modulus)	$10\%3$	1

การประมวลผลทางคณิตศาสตร์เกี่ยวกับการบวก ลบ คูณ หาร และการหารเอาเศษ ซึ่งจะมีลำดับการคำนวณของเครื่องหมายทางคณิตศาสตร์ดังต่อไปนี้

1. วงเล็บ (...(...)) โดยจะทำการคำนวณจากข้างในไปข้างนอก
2. เครื่องหมายหาร (/) เครื่องหมายคูณ (*) และการหารเอาเศษ (%)
3. เครื่องหมายบวก (+) เครื่องหมายลบ (-)
4. ถ้าเป็นเครื่องหมายเดียวกัน การคำนวณจะทำจากด้านซ้ายไปด้านขวา
5. การคำนวณจะคำนวณสมการทางด้านขวาให้เสร็จก่อน แล้วจึงนำไปให้กับตัวแปรทางด้านซ้าย

2. ตัวดำเนินการทางด้านการเปรียบเทียบ (Comparison Operator) เป็นการเปรียบเทียบกันระหว่างข้อมูล หรือ ตัวถูกดำเนินการ อย่างน้อย 2 จำนวน โดยที่การเปรียบเทียบนั้นจะทำการเปรียบเทียบทีละ 2 จำนวน และผลลัพธ์ที่ได้จะอยู่ในรูปตรรกะ (Boolean) คือ True (จริง) และ False (เท็จ) ถ้ามีการเปรียบเทียบมากกว่า 2 จำนวน จะทำการเปรียบเทียบจากด้านซ้ายมือไปขวามือหรือภายในวงเล็บก่อน โดยมีการเชื่อมด้วยตรรกะ เช่น and (และ) or (หรือ)

ตัวดำเนินการ	ชื่อ	การเปรียบเทียบ	ผลลัพธ์
==	เท่ากับ	$\$a == \b	จริงถ้า \$a มีค่าเท่ากับ \$b
!=	ไม่เท่ากับ	$\$a != \b	จริงถ้า \$a มีค่าไม่เท่ากับ \$b
===	เท่ากับ	$\$a === \b	จริงถ้า \$a เท่ากับ \$b และชนิดข้อมูลเหมือนกัน
>	มากกว่า	$\$a > \b	จริงถ้า \$a มีค่ามากกว่า \$b
>=	มากกว่าหรือเท่ากับ	$\$a >= \b	จริงถ้า \$a มีค่ามากกว่าหรือเท่ากับ \$b
<	น้อยกว่า	$\$a < \b	จริงถ้า \$a มีค่าน้อยกว่า \$b
<=	น้อยกว่าหรือเท่ากับ	$\$a <= \b	จริงถ้า \$a มีค่าน้อยกว่าหรือเท่ากับ \$b

3. ตัวดำเนินการทางด้านการตรรกศาสตร์ (logical Operator) ตัวดำเนินการทางตรรกะ จะนำข้อมูลสองค่ามากระทำทางลอจิกต่อกันเมื่อกระทำค่าใดผลลัพธ์ที่ออกมาจะเป็นจริงหรือเท็จ

ตัวดำเนินการ	ชื่อ	การเปรียบเทียบ	ผลลัพธ์
&& หรือ and	และ	\$a && \$b หรือ \$a and \$b	จริงถ้า \$a และ \$b เป็นจริงทั้งคู่
หรือ or	หรือ	\$a \$b หรือ \$a or \$b	จริงถ้า \$a หรือ \$b เพียงตัวใดตัวหนึ่งเป็นจริง หรือเป็นจริงทั้งคู่
! หรือ not	ไม่หรือนิเสธ	!\$a	จริงถ้า \$a เป็นเท็จ, เท็จถ้า \$a เป็นจริง

4. ตัวดำเนินการกำหนดค่า (Assignment Operator) เป็นตัวดำเนินการที่ใช้ในการกำหนดค่าให้กับตัวแปรที่อยู่ทางด้านซ้ายของตัวดำเนินการ ด้วยค่าของตัวแปรที่อยู่ทางขวา แล้วนำผลลัพธ์ที่ได้เก็บไว้ในตัวแปรเดิม เป็นการเขียนคำสั่งในรูปแบบย่อลงมา

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	ผลลัพธ์
=	กำหนดค่า (กำหนดให้ตัวแปรทางด้านซ้ายมีค่าเท่ากับค่าที่ระบุ)	\$a=5 \$a = 1	\$a มีค่าเท่ากับ 1
+=	เพิ่มค่า (นำค่าที่กำหนดไปบวกเพิ่มจากค่าเดิมตัวแปร)	\$a=7 \$a += 2	\$a มีค่าเท่ากับ 9
-=	ลบค่า (ลดค่าตัวแปรลงเท่ากับค่าที่ระบุ)	\$a=20 \$a -= 5	\$a มีค่าเท่ากับ 15
*=	คูณค่า (คูณค่าเดิมของตัวแปรด้วยค่าที่ระบุ)	\$a=10 \$a *= 4	\$a มีค่าเท่ากับ 40
/=	หารค่า(หารค่าเดิมของตัวแปรด้วยค่าที่ระบุ)	\$a=7 \$a /= 2	\$a มีค่าเท่ากับ 3.5
%=	หารค่าแบบเก็บเศษ (นำค่าที่ระบุไปหารค่าเดิมของตัวแปร แต่จะเอาเฉพาะเศษจากการหารเท่านั้น)	\$a=10 \$a %= 3	\$a มีค่าเท่ากับ 1
.=	เชื่อมต่อข้อความ (ใช้ในการเชื่อมต่อข้อความโดยนำข้อความด้านขวามือไปต่อท้ายข้อความด้านซ้ายมือ)	\$a= "Hello" \$a.= "PHP"	\$a มีค่าเท่ากับ "Hello PHP"

5. ตัวดำเนินการเพิ่มและลดค่า เป็นตัวดำเนินการที่ใช้สำหรับการเพิ่มค่าตัวแปรครั้งละ 1 หรือลดค่าตัวแปรลงครั้งละ 1 แต่มีการเรียกใช้งานแตกต่างกันไปตามตำแหน่งของตัวดำเนินการว่าด้านหน้าหรือด้านหลัง ทำให้มีผลต่างกัน

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	ผลลัพธ์
++	เป็นการเพิ่มตัวแปรขึ้นไปอีก 1 เช่น	++\$a	\$a ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการมีค่าเพิ่มขึ้น 1 ค่า
		\$a++	\$a ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการมีค่าคงที่ แต่หลังเรียกใช้งานตัวดำเนินการมีค่าเพิ่มขึ้น 1 ค่า
--	เป็นการลดค่าตัวแปรลงอีก 1	--\$a	\$a ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการมีค่าลดลง 1 ค่า
		\$a--	\$a ทำให้ค่าของตัวแปรขณะเรียกใช้งานตัวดำเนินการมีค่าคงที่ แต่หลังเรียกใช้งานตัวดำเนินการมีค่าลดลง 1 ค่า

ตัวอย่าง

```

1  <?php
2      $a = 10;
3      echo "ค่าของตัวแปร \$a = \$a <br>";
4      echo "ค่าของตัวแปร \$a ขณะเรียกใช้งานตัวดำเนินการ \$a++ = ". $a++." <br>";
5      echo "ค่าของตัวแปร \$a หลังเรียกใช้งานตัวดำเนินการ = \$a <br><br>";
6      $b = 15;
7      echo "ค่าของตัวแปร \$b = \$b <br>";
8      echo "ค่าของตัวแปร \$b ขณะเรียกใช้งานตัวดำเนินการ ++\$b = ". ++$b." <br>";
9      echo "ค่าของตัวแปร \$b หลังเรียกใช้งานตัวดำเนินการ = \$b <br><br>";
10     $c = 20;
11     echo "ค่าของตัวแปร \$c = $c <br>";
12     echo "ค่าของตัวแปร \$c ขณะเรียกใช้งานตัวดำเนินการ \$c-- = ". $c--." <br>";
13     echo "ค่าของตัวแปร \$c หลังเรียกใช้งานตัวดำเนินการ = $c <br><br>";
14     $d = 25;
15     echo "ค่าของตัวแปร \$d = $d <br>";
16     echo "ค่าของตัวแปร \$d ขณะเรียกใช้งานตัวดำเนินการ --$d = ". --$d." <br>";
17     echo "ค่าของตัวแปร \$d หลังเรียกใช้งานตัวดำเนินการ = $d <br><br>";
18  ?>

```

ผลการรันของโปรแกรม

```

ค่าของตัวแปร $a = 10
ค่าของตัวแปร $a ขณะเรียกใช้งานตัวดำเนินการ $a++ = 10
ค่าของตัวแปร $a หลังเรียกใช้งานตัวดำเนินการ = 11

ค่าของตัวแปร $b = 15
ค่าของตัวแปร $b ขณะเรียกใช้งานตัวดำเนินการ ++$b = 16
ค่าของตัวแปร $b หลังเรียกใช้งานตัวดำเนินการ = 16

ค่าของตัวแปร $c = 20
ค่าของตัวแปร $c ขณะเรียกใช้งานตัวดำเนินการ $c-- = 20
ค่าของตัวแปร $c หลังเรียกใช้งานตัวดำเนินการ = 19

ค่าของตัวแปร $d = 25
ค่าของตัวแปร $d ขณะเรียกใช้งานตัวดำเนินการ --$d = 24
ค่าของตัวแปร $d หลังเรียกใช้งานตัวดำเนินการ = 24

```

บทที่ 5 การสร้างฟอร์ม HTML และการรับ ส่งค่าข้อมูลในภาษา PHP

ฟอร์ม (Form) เป็นเครื่องมือสำหรับใช้ในการรับข้อมูลจากผู้ใช้งานเว็บไซต์ผ่านเว็บเบราว์เซอร์ และส่งข้อมูลนั้นต่อไปยังเว็บเซิร์ฟเวอร์เพื่อประมวลผลต่อไป ตัวอย่างแบบฟอร์มที่ใช้กันทั่วไปเช่น การสมัครสมาชิกผ่านเว็บไซต์ การตอบแบบสอบถามออนไลน์ การแสดงความคิดเห็นผ่านเว็บไซต์ เป็นต้น

การสร้างฟอร์ม HTML

รูปแบบ < form name="ชื่อฟอร์ม" method="วิธีส่งข้อมูล" action="ชื่อไฟล์สคริปต์ที่จะเรียกขึ้นมาใช้งาน">

ตัวอย่าง <form name= “frmRegis” method= “post” action= “regis.php”>

สำหรับวิธีการส่งข้อมูลผ่านฟอร์มนั้นสามารถเลือกใช้งานได้จาก 2 รูปแบบ คือ POST หรือ GET ขึ้นอยู่กับความเหมาะสมของงาน ซึ่งความแตกต่างของ POST และ GET จะกล่าวในหัวข้อต่อไป

องค์ประกอบต่างๆ ของฟอร์มรับข้อมูล

1. กล่องรับข้อความบรรทัดเดียว (Textfield) กล่องรับข้อความบรรทัดเดียวจะใช้กรอกข้อความสั้นๆ เช่นชื่อ, นามสกุล, เบอร์โทรศัพท์, หรืออีเมล เป็นต้น

รูปแบบ

< input type="text" name="ชื่อกล่องข้อความ" size="ความกว้างกล่องรับข้อความ" maxlength="จำนวนตัวอักษรสูงสุด">

ตัวอย่าง

< input type="text" name="user" size="25" maxlength="25">

2. กล่องรับรหัสผ่าน (PasswordField) กล่องรับรหัสผ่านจะรับข้อมูลเป็นรหัสผ่านโดยเฉพาะ เนื่องจากผู้ใช้กรอกข้อมูลลงในกล่องรับรหัสผ่านจะปรากฏเป็นเครื่องหมายดอกจัน (*) แทนตัวอักษรแต่ละตัวในรหัสผ่าน

รูปแบบ

< input type="password" name="ชื่อกล่องข้อความ" size="ความกว้างกล่องรับข้อความ" maxlength="จำนวนตัวอักษรสูงสุด">

ตัวอย่าง

< input type="password" name="Pwd" size="8" maxlength="8">

3. กล่องรับข้อความหลายบรรทัด (TextArea) กล่องรับข้อความหลายบรรทัดเหมาะสำหรับเป็นกล่องรับข้อความยาวๆ เช่น กล่องรับที่อยู่ หรือกล่องรับความคิดเห็น เป็นต้น

รูปแบบ

< textarea name="ชื่อกล่องข้อความ" cols="จำนวนของตัวอักษรใน 1 บรรทัด" rows="จำนวนบรรทัด">ข้อความ
< /textarea>

4. การใช้ปุ่มตัวเลือก การสร้างข้อมูลเป็นแบบปุ่มตัวเลือกซึ่งมีอยู่ 2 ประเภทหลักด้วยกันคือ

1. ปุ่มตัวเลือกแบบเรดิโอ (Radio) ลักษณะของ ปุ่มตัวเลือกแบบ Radio คือ ใน 1 ชุดคำถามผู้ใช้สามารถเลือกคำตอบได้เพียงแค่ 1 คำตอบเท่านั้น เช่นข้อมูลประเภท เพศ, ศาสนา หรือข้อสอบที่เลือกตอบได้เพียงคำตอบเดียวเท่านั้น

รูปแบบ

<input type="radio" name="ชื่อของกลุ่มตัวเลือก" value="ค่าของตัวเลือก"/>ข้อความที่ต้องการให้แสดง

ตัวอย่าง

การเขียนโปรแกรมบนเว็บด้วยภาษา PHP (ฉบับพื้นฐาน)

อ.ชนิดา แก้วเพชร

```
<input type="radio" name="gender" value="male"/>ชาย
<input type="radio" name="gender" value="female"/>หญิง
```

2. ปุ่มตัวเลือกแบบเช็คบ็อกซ์ (CheckBox) ลักษณะของปุ่มแบบ Checkbox คือสามารถเลือกได้มากกว่า 1 ตัวเลือก เช่น เลือกเกี่ยวกับความชอบ หรือเลือกข้อสอบที่ให้เลือกตอบมากกว่า 1 ข้อ เป็นต้น

รูปแบบ

```
< input type="checkbox" name="ชื่อของตัวเลือก" value="ค่าของตัวเลือก" />ข้อความที่ต้องการให้แสดง
```

ตัวอย่าง

```
<input type="checkbox" name="Hobby" value="อ่านหนังสือ"> อ่านหนังสือ
```

```
<input type="checkbox" name=" Hobby " value="ดูทีวี"> ดูทีวี
```

5. กล่องรับข้อมูลแบบตัวเลือก (List/Menu) เป็นกล่องรับข้อมูลที่มีลักษณะเป็นเมนูตัวเลือกในลักษณะ Drop down Menu ผู้ใช้สามารถคลิกเมาส์เลือกรายการตัวเลือกได้ 1 รายการเท่านั้นจากรายการทั้งหมดที่แสดง

รูปแบบ

```
< select name="ชื่อกล่องตัวเลือก">
```

```
    < option value="ค่าของตัวเลือกที่ 1" selected="select">ชื่อตัวเลือกที่ 1< /option>
```

```
    < option value="ค่าของตัวเลือกที่ 2">ชื่อตัวเลือกที่ 2< /option>
```

```
    < option value="ค่าของตัวเลือกที่ n">ชื่อตัวเลือกที่ n< /option>
```

```
< /select>
```

ตัวอย่าง

```
< select name="beverage">
```

```
    <option value="IcedTea" >ICED TEA </option>
```

```
    <option value="LemonTea"> LEMON TEA </option>
```

```
    <option value="COFF" >COFFEE </option>
```

```
< /select>
```

6. ปุ่มแบบซั่มมิท (Submit) หรือปุ่มส่งข้อมูล ทำหน้าที่ส่งข้อมูลที่ผู้ใช้กรอกไปประมวลผลต่อยังสคริปต์ที่ต้องการ เมื่อผู้ใช้คลิกที่ปุ่ม ข้อมูลจะถูกส่งจากฟอร์มผ่านเบราว์เซอร์ไปยังเซิร์ฟเวอร์

รูปแบบ

```
<input type= "submit" value= "ข้อความที่ปรากฏบนปุ่ม">
```

ตัวอย่าง

```
<input type= "submit" value= "Save">
```

7. ปุ่มแบบรีเซต (Reset) หรือปุ่มล้างข้อมูล ทำหน้าที่ลบข้อมูลทั้งหมดที่กรอกเข้ามาในฟอร์มขณะนั้นออกไป เหมือนกับการเคลียร์ค่าทั้งหมดเพื่อให้ผู้ใช้กรอกข้อมูลใหม่

รูปแบบ

```
<input type= "reset" value= "ข้อความที่ปรากฏบนปุ่ม">
```

ตัวอย่าง

```
<input type= "reset" value= "Cancel">
```

การรับ ส่งค่าข้อมูลในภาษา PHP

ในการเขียนโปรแกรมบนเว็บการรับ ส่งค่าข้อมูลจากภาษาพีเอชพีเพื่อนำไปประมวลต่อไป หรือการส่งข้อมูลไปยังเว็บเซิร์ฟเวอร์ก็เป็นสิ่งสำคัญ อย่างหนึ่งเช่นกัน การรับ ส่งค่าในภาษาพีเอชพีนั้นมีทั้งรูปแบบ วิธีการและการนำไปใช้ที่แตกต่างกัน ผู้พัฒนาโปรแกรมต้องเลือกวิธีการส่งค่าให้เหมาะสมกับงานเพื่อให้เว็บไซต์ที่พัฒนามีประสิทธิภาพ

รูปแบบการรับ ส่งค่าของภาษาพีเอชพี

การรับ ส่งค่าข้อมูลในภาษาพีเอชพี มีรูปแบบ 2 รูปแบบคือ

1. แบบ POST

POST เป็นค่าส่งค่าข้อมูลผ่านทางฟอร์มรับข้อมูลที่ใช้กรอกข้อมูลผ่านทางเว็บไซต์ไปยังสคริปต์ ทำให้การรับส่ง-ค่าข้อมูลแบบนี้ ไม่สามารถเห็นข้อมูลที่ถูกส่งไปได้ เช่น การส่งข้อมูลการสมัครสมาชิก การล็อกอินเข้าสู่ระบบ เป็นต้น

การรับค่าข้อมูลในภาษาพีเอชพีหากมีการส่งค่าข้อมูลจากฟอร์มรับข้อมูลในลักษณะ POST การรับค่าข้อมนั้นก็ต้องเป็นการรับข้อมูลแบบเดียวกัน จึงสามารถเรียกใช้งานข้อมูลที่มีการส่งมาใช้งานได้

ตัวอย่าง

```
1 <form method="post" action="">
2   <p>Username : <input type="text" name="Username"></p>
3   <p>Password : <input type="password" name="Password"></p>
4   <p><input type="submit"><input type="reset"></p>
5 </form>
```

จากตัวอย่างข้างต้นเป็นการสร้างฟอร์มเพื่อรับข้อมูลจากเว็บไซต์ และเลือกวิธีการส่งค่าข้อมูลเป็นแบบ POST (ผู้พัฒนาเว็บไซต์อาจเลือกวิธีการส่งเป็นแบบ GET ก็ได้ ทั้งนี้ขึ้นอยู่กับความเหมาะสมของงาน) ดังนั้นหากเราต้องการนำข้อมูลที่ส่งผ่านฟอร์มไปใช้งานจึงต้องมีการรับค่าข้อมูลแบบ POST เช่นกัน

ตัวอย่าง

```

1 <?php
2 $username=$_POST[Username];
3 $password=$_POST[Password];
4 echo"$username <br> ";
5 echo"$password";
6 ?>

```

จากตัวอย่าง อธิบายได้ดังนี้ การรับค่าข้อมูลจากฟอร์มนั้นคือการอ้างถึงฟอร์มที่ได้มีการส่งค่าจากหน้าเว็บไซต์เพื่อนำไปประมวลผลต่อไป จากตัวอย่างก่อนหน้าได้มีการสร้างฟอร์มเพื่อส่งค่าข้อมูล 2 ข้อมูลคือ ข้อมูล Username และ Password ดังนั้นหากต้องการนำข้อมูล Username และ Password ที่ส่งค่าผ่านฟอร์มไปใช้งาน จึงต้องมีการรับค่าข้อมูลแบบ POST เช่นกัน ซึ่งมีลักษณะการใช้งานดังรูปแบบต่อไปนี้

รูปแบบ	\$ชื่อตัวแปรเพื่อเก็บค่าที่ส่งมา=\$_POST[ชื่อกล่องรับข้อมูลจากหน้าฟอร์มส่งข้อมูล];
ตัวอย่างเช่น	\$username=\$_POST[Username];

2. แบบ GET

GET เป็นการส่งค่าข้อมูลผ่านทาง URL ซึ่งจะมีตัวแปรและค่าสำหรับใส่ในตัวแปรนั้น การเริ่มสร้างตัวแปร \$_GET จะเริ่มต้นด้วยเครื่องหมาย “?” และแต่ละตัวแปรจะถูกเชื่อมด้วย “&” ซึ่งข้อมูลที่ถูกส่งไปนั้น จะปรากฏในช่อง Address Bar ของเว็บเบราว์เซอร์

ตัวอย่าง

<http://www.rmutsb.ac.th/2011/news/detail.php?id=content13031123172201012>

จากตัวอย่าง อธิบายได้ดังนี้ จาก URL: <http://www.rmutsb.ac.th/2011/news/detail.php?id=content13031123172201012> คือ URL เพื่อเข้าสู่เนื้อหาของเว็บไซตนั้นตามปกติ

สำหรับเครื่องหมาย ? หลัง URL ปกติ นั้นคือเครื่องหมายของการเริ่มต้นสร้างตัวแปรแบบ GET ในที่นี้คือ id ทำหน้าที่เป็นตัวแปรของการส่งค่าข้อมูลแบบ GET ซึ่งถูกกำหนดให้มีค่าเท่ากับ “content13031123172201012”

การส่งค่าข้อมูลแบบ GET นั้นสามารถส่งค่าตัวแปรได้มากกว่า 1 ตัวแปร แต่ละตัวแปรจะถูกเชื่อมด้วยเครื่องหมาย “&” ดังที่กล่าวไว้ในข้างต้น

ตัวอย่าง

<http://rmutsb.ac.th/stdList.php?year=2550&semesterId=2&subjectId=3117>

จากตัวอย่าง อธิบายได้ดังนี้ <http://rmutsb.ac.th/stdList.php> คือ URL เพื่อเข้าสู่เนื้อหาของเว็บไซต์นั้นตามปกติ

สำหรับเครื่องหมาย ? หลัง URL ปกติ นั้นคือเครื่องหมายของการเริ่มต้นสร้างตัวแปรแบบ GET ในที่นี้คือ year ทำหน้าที่เป็นตัวแปรของการส่งค่าข้อมูลแบบ GET ซึ่งถูกกำหนดให้มีค่าเท่ากับ “2550” , semesterId ทำหน้าที่เป็นตัวแปรของการส่งค่าข้อมูลแบบ GET เช่นกัน ซึ่งถูกกำหนดให้มีค่าเท่ากับ “2” และตัวแปรสุดท้ายคือ subjectId ทำหน้าที่เป็นตัวแปรของการส่งค่าข้อมูลแบบ GET และถูกกำหนดให้มีค่าเท่ากับ “3117” จากตัวอย่าง จะสังเกตได้ว่าตัวแปรแต่ละตัวถูกเชื่อมด้วยเครื่องหมาย “&”

สำหรับการรับค่าแบบ GET ก็มีหลักเกณฑ์ในการรับค่าเช่นเดียวกับกับ POST คือ หากส่งค่าข้อมูลในรูปแบบ GET ก็ต้องรับค่าข้อมูลด้วย GET เช่นเดียวกัน

ตัวอย่าง

```
1 <?php
2 $yearREF=$_GET[year];
3 $semesterREF=$_GET[semesterId];
4 $subjectREF=$_GET[subjectId];
5 echo "$yearREF<br>";
6 echo "$semesterREF<br>";
7 echo "$subjectREF";
8 ?>
```

จากตัวอย่าง อธิบายได้ดังนี้ การรับค่าข้อมูลจาก URL หรือการรับค่าจากการส่งข้อมูลแบบ GET นั้น จากตัวอย่างมีการส่งค่าข้อมูลผ่านตัวแปร 3 ตัวแปร คือ ข้อมูล year, semesterId และ subjectId ดังนั้นหากต้องการนำข้อมูลที่ส่งค่าผ่าน URL ไปใช้งาน จึงต้องมีการรับค่าข้อมูลแบบ GET เช่นกัน ซึ่งมีลักษณะการใช้งานดังรูปแบบต่อไปนี้

รูปแบบ	\$ชื่อตัวแปรเพื่อเก็บค่าที่ส่งมา=\$_GET[ชื่อตัวแปรที่ส่งแบบ GET];
ตัวอย่างเช่น	\$yearREF=\$_POST[year];