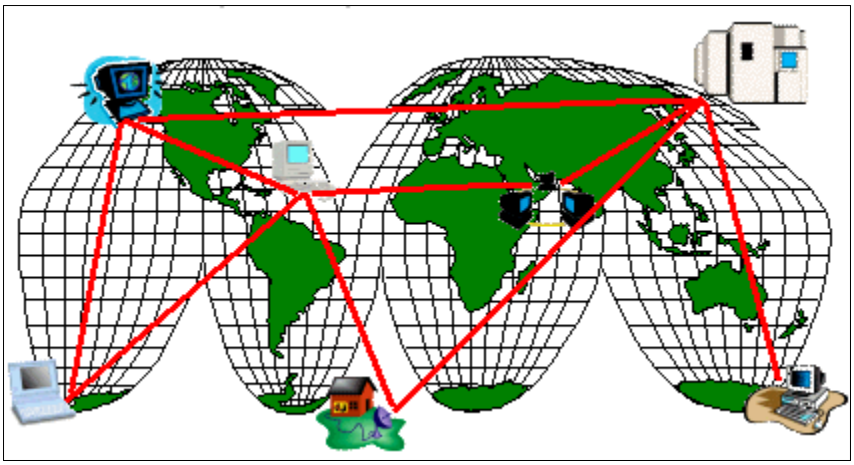
Chapter 1

แนะนำการสร้างเว็บไซต์

ก่อนที่เราจะสร้างเว็บไซต์ได้นั้น ควรทำความรู้จักคำว่า “เว็บไซต์” ให้พอเข้าใจถึงลักษณะการทำงานกันก่อน เว็บไซต์ (Web Site) แปลตามคำก็คือ Web แปลว่า เครือข่ายใยแมงมุม และ Site แปลว่า ที่ตั้ง,หรือที่อยู่ อธิบายให้เข้าใจได้ง่าย เว็บไซต์ก็คือ เว็บเพจหลายๆหน้า ซึ่งเชื่อมโยงกัน และจัดเก็บในรูปแบบของเวิลด์ไวด์เว็บ (World Wide Web) เชื่อมโยงกันทั่วโลก



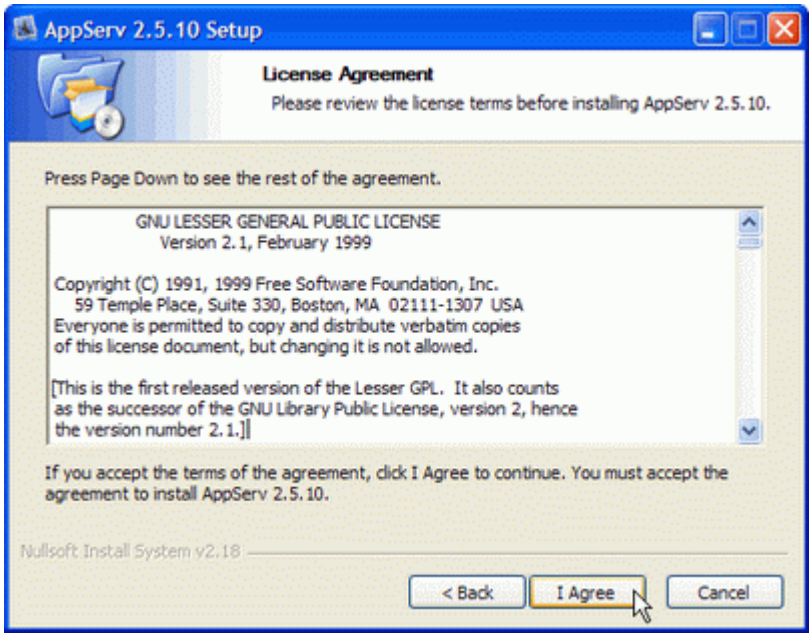
ภาพจาก: http://school.obec.go.th/borkruwitt/inter/internet01.gif

โดยทั่วไปการสร้างเว็บไซต์นั้น จะเขียนด้วยภาษา HTML เป็นหลัก แต่อาจจะแทรกด้วย ภาษา PHP,JavaScript,ASP เพื่อใช้ในการคำนวณ ประมวณผล หรือรันโปรแกรมต่างๆ ในที่นี้จะใช้ PHP ในการประมวณผลและติดต่อกับโปรแกรม SCILAB ซึ่งจะต้องมีการติดตั้งโปรแกรมต่างๆเพื่อเตรียมพร้อมก่อนการใช้งานเราจะใช้โปรแกรม AppServ ซึ่งเป็นชุดติดตั้งที่รวมโปรแกรมต่างๆที่เกี่ยวข้องกับการใช้งาน PHP มาใช้ สามารถดาวน์โหลดได้ที่ http://www.appservnetwork.com ขั้นตอนติดตั้งมีแนวทางดังนี้

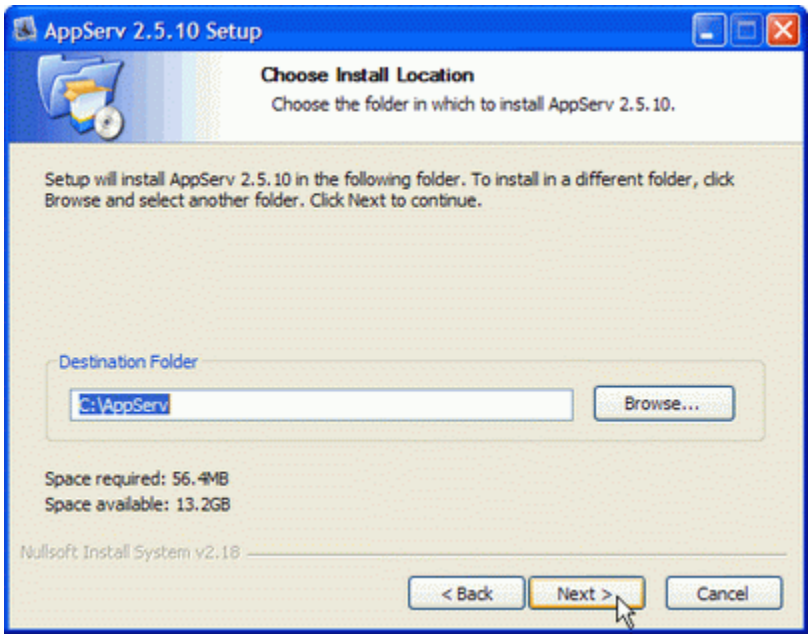
1. หลังจากดาวน์โหลดโปรแกรมมาแล้วให้ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมา  จะเข้าสู่หน้าจอการติดตั้งเหมือนกับโปรแกรมทั่วไป

2. จะปรากฏหน้าจอตัวช่วยการติดตั้ง คลิก Next

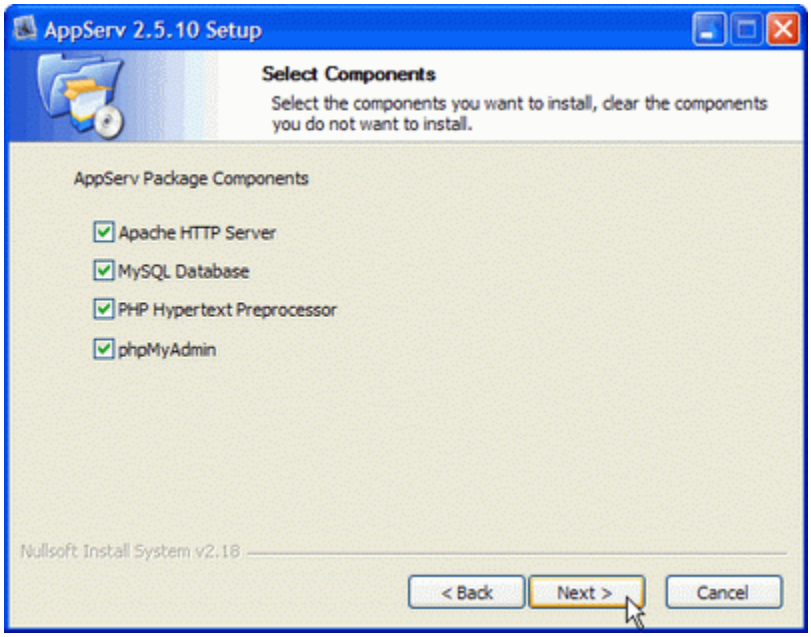
3. หน้าจอเงื่อนไขการใช้งาน ให้คลิก I Agree



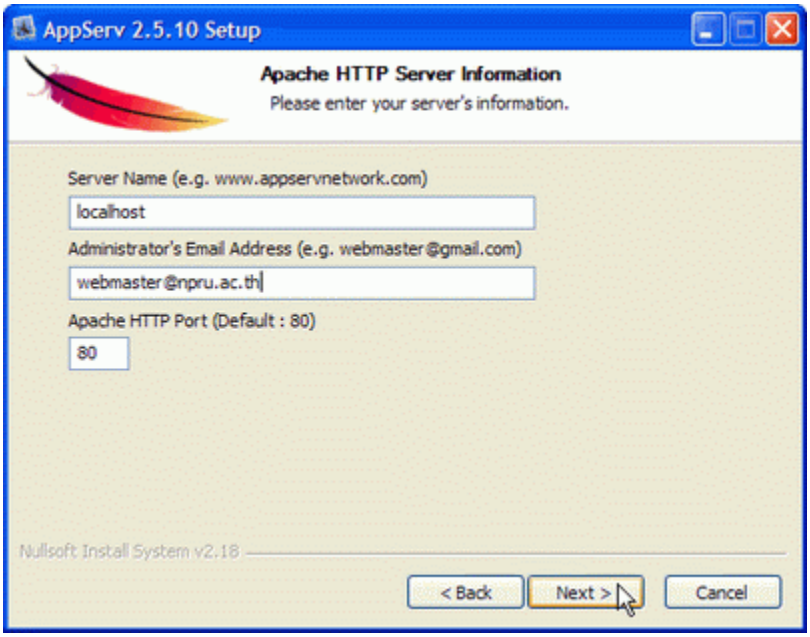
4. กำหนดตำแหน่งการติดตั้ง ให้ใช้ค่าดีฟอลต์คือ C:\AppServ



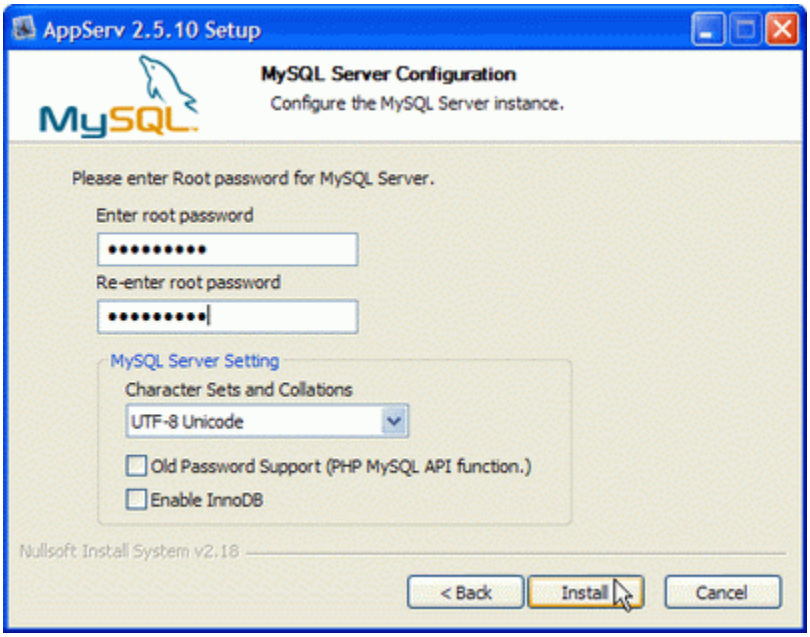
5. เลือกโปรแกรมที่จะติดตั้ง ให้เลือกทั้งหมด



6. กำหนดชื่อเซิร์ฟเวอร์ ให้กำหนดเป็น localhost และ อีเมล์ ให้ใส่เมล์อะไรก็ได้ ในตัวอย่างจะใช้ webmaster@npru.ac.th จากนั้น คลิก Next

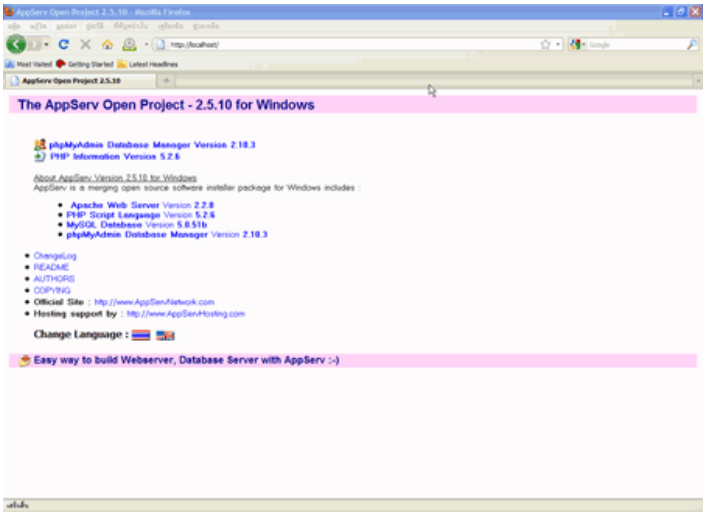


7. กำหนดพาสเวิร์ดของ MySQL ค่านี้สำคัญมากเราต้องจำให้ได้ มิฉะนั้นจะไม่สามารถใช้งานฐานข้อมูลได้ ช่อง Character Sets And Collations ให้กำหนดเป็น UTF-8 Unicode



8. จากนั้นจะเข้าสู่หน้าจอการติดตั้ง ให้รอจนกว่าจะติดตั้งเสร็จ

หลังจากติดตั้งโปรแกรม AppServ เสร็จเรียบร้อยแล้วให้ทดสอบการติดตั้งว่าโปรแกรมที่เราติดตั้งไปนั้นใช้งานได้หรือไม่ ให้เปิดโปรแกรม Browser ขึ้นมา พิมพ์ http://localhost จะปรกกฏหน้าจอดังรูป แสดงว่าการติดตั้งโปรแกรมนั้นสมบูรณ์พร้อมใช้งาน



Chapter 2

ภาษาที่ใช้ในการติดต่อกับ SCILAB

ภาษาที่ใช้ในเขียนเว็บไซต์เพื่อติดต่อกับ SCILAB มีดังนี้

HTML (Hypertext Markup Language)

ใช้ในการกำหนดรูปแบบของเอกสารเว็บเพจ ลักษณะของคำสั่งจะเรียกว่า แท็ก (Tag) ใช้ในการระบุจุดเริ่มต้นและจุดสิ้นสุดของคำสั่งที่ต้องการจัดรูปแบบเอกสาร ซึ่งแท็กใน HTML มีอยู่มากมายเช่น การกำหนดตัวอักษร ตาราง พื้นหลัง ฯลฯ

โครงสร้างของ HTML

จะประกอบไปด้วยแท็กหลักๆดังนี้

```
<html>
<head>
        <title>…</title>
</head>
<body>
 ……
 ……
</body>
</html>
```

จากโค้ดนี้ <html>…..</html>คือ เอกสาร HTML ทั้งหมด ซึ่งภายในก็จะมีองค์ประกอบที่สำคัญๆอยู่ 2 ส่วนคือ

1. ส่วนของ Header

จะอยู่ระหว่าง <head>…..</head>ในส่วนนี้เราจะเขียนแท็กที่ใช้เป็นข้อกำหนดดังตาราง

<head>…..</head>	ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของส่วนหัวเอกสาร
<title>…..</title>	กำหนดชื่อของเอกสาร ซึ่งจะปรากฏที่ Title Bar
<mata>	ใช้ในการกำหนดข้อมูลพิเศษบางอย่าง เช่น คีย์เวิร์ด (keyword) และคำอธิบายสำหรับเสิร์ชเอนจิ้น (Search Engine) หรือการรีเฟรชเพจ (Refresh Page) เป็นต้น
<style>…</style>	ใช้ในการกำหนดรูปแบบ CSS
<link>	ใช้สำหรับการเชื่อมโยงรูปแบบของ CSS จากไฟล์ภายนอก
<script>…</script>	ใช้ในการเรียกใช้งาน JavaScript

2. ส่วนของ Body อยู่ระหว่าง <body>….</body>

ในแต่ละแท็ก จะมีข้อกำหนดย่อยๆ ที่เรียกว่า แอตทริบิวต์ เช่น แท็กเกี่ยวกับตัวอักษร จะมีการกำหนดชนิด , ขนาด, สีตัวอักษร เป็นต้น โดยที่แต่ละแท็กก็จะมีแอตทริบิวต์ที่แตกต่างกันไป และเราจะเรียกทุกๆอย่างที่อยู่ระหว่างแท็กเปิดไปจะถึงแท็กปิดว่า อิลิเมนต์ (Element) รูปแบบการกำหนดแอตทริบิวต์คือ

```
<ชื่อแท็ก แอตทริบิวต์ที่1="ค่าที่กำหนด"แอตทริบิวต์ที่2="ค่าที่กำหนด"แอตทริบิวต์ที่3="ค่าที่กำหนด"…>ข้อความ</ชื่อแท็ก>
```

ตัวอย่าง

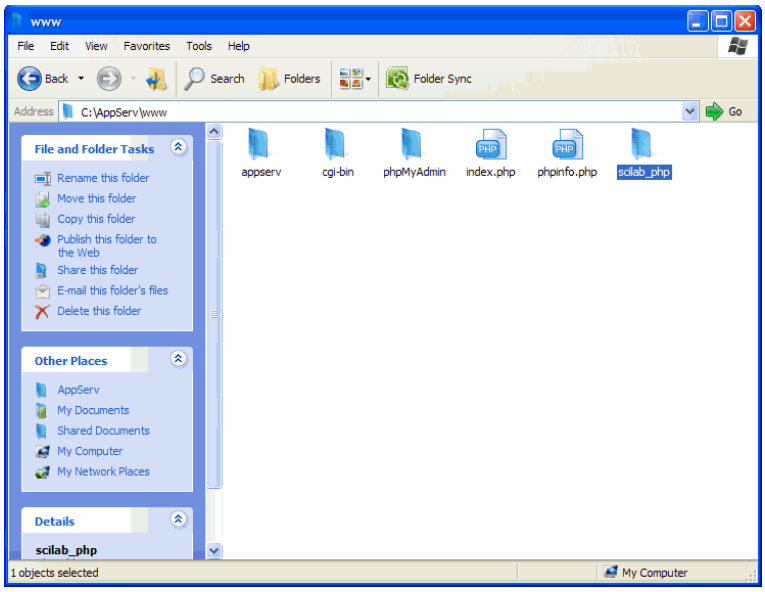
```
<font face="Tahoma" size="14" color="red">ทดสอบ</font>
```

เริ่มต้นกับ PHP

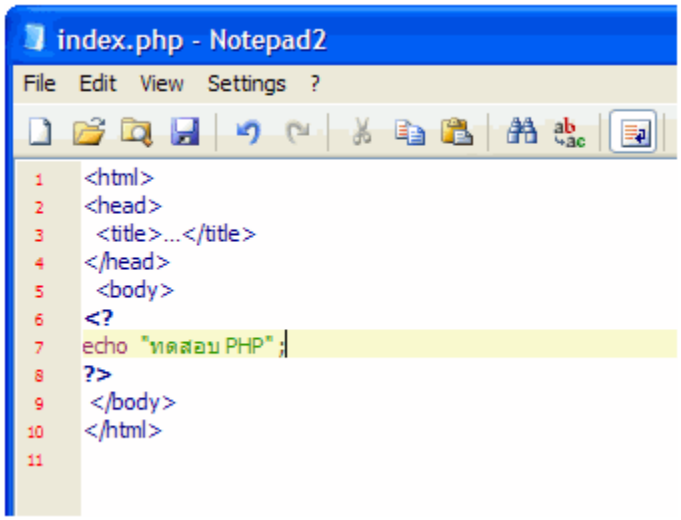
PHP แต่เดิมย่อมาจาก Personal Home Page แต่ต่อมาก็เปลี่ยนเป็นย่อมาจาก PHP Hypertext Preprocessor เป็นภาษาสคริปต์แบบเซิร์ฟเวอร์ไซด์ (server-side scripting language) หมายถึงการประมวลผลจะเกิดขึ้นบนเครื่องแม่ข่าย (Server) แล้วส่งผลลัพธ์กลับมายังเครื่องลูกข่าย (Client) การเขียน PHP จะแทรกลงบนเอกสาร HTML ด้วยการเปิดแท็ก <?php และปิดด้วยแท็ก ?>หรือ แท็ก <?...?> ก็ได้เช่นกัน PHP เป็นโอเพ่นซอร์ส (Open Source) สามารถดาวน์โหลด PHP มาใช้งานได้ฟรีจากเว็บไซต์ของ PHP (www.php.net/downloads.php)

การจัดเก็บและการทดสอบการทำงาน

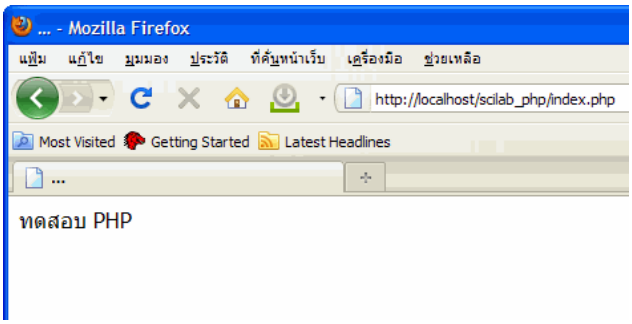
การจัดเก็บไฟล์ PHP นั้นต้องจัดเก็บไว้ใน Document Root ที่เรากำหนด สำหรับค่าดีฟอลต์ของ AppServ คือ C:\AppServ\www\ และสร้างโฟล์เดอร์ใหม่สำหรับเก็บไฟล์ชื่อ scilab_php



ทดสอบการทำงานด้วยคำสั่งง่ายๆ โดยการเปิดโปรแกรม Notepad แล้วพิมพ์คำสั่ง ดังนี้



ผลการรันโปรแกรม



การแสดงผล

การแสดงผลของ PHP มีอยู่หลายรูปแบบคำสั่ง แต่ในที่นี้จะใช้ 2 คำสั่งหลักๆ คือ print และ echo

ฟังก์ชัน print()

รูปแบบคำสั่ง

```
Print (ผลลัพธ์);  หรือ print ผลลัพธ์;
```

ตัวอย่าง

```
print("PHP &Scilab");

print "PHP world";
```

ฟังก์ชัน **echo()**

เป็นคำสั่งที่นิยมใช้กันมากที่สุด เพราะทำงานได้รวดเร็วกว่าคำสั่ง print เนื่องจากคำสั่งนี้ไม่มีการตรวจสอบข้อผิดพลาด

รูปแบบคำสั่ง

```
echo (ผลลัพธ์);  หรือ echo ผลลัพธ์;
```

ตัวอย่าง

```
echo("PHP & Scilab");

echo "PHP world";
```

คำอธิบาย (Comment)

การเขียนคำอธิบายคือการแทรกข้อความเข้าไปในโค้ดโปรแกรมแต่จะไม่นำไปประมวลผล ในภาษา PHP นั้นจะใช้รูปแบบเดียวกันกับ JavaScript คือ

Single-line comment จะใช้ Comment ที่ละบรรทัด เช่น //Comment PHP

Multiple-line comment จะใช้ Comment ครั้งละหลายๆบรรทัดติดต่อกัน เช่น

```
/*  This is the comment PHP

    This is Multiple-line Comment

*/
```

ตัวแปร (Variable)

การเขียนโปรแกรม PHP นั้นก็เหมือนกับการเขียนโปรแกรมด้วยภาษาอื่นๆ ซึ่งจะกำหนดตัวแปรเพื่อจองพื้นที่ในหน่วยความจำสำหรับเก็บพักข้อมูล ซึ่งมีวิธีการกำหนดตัวแปรดังนี้

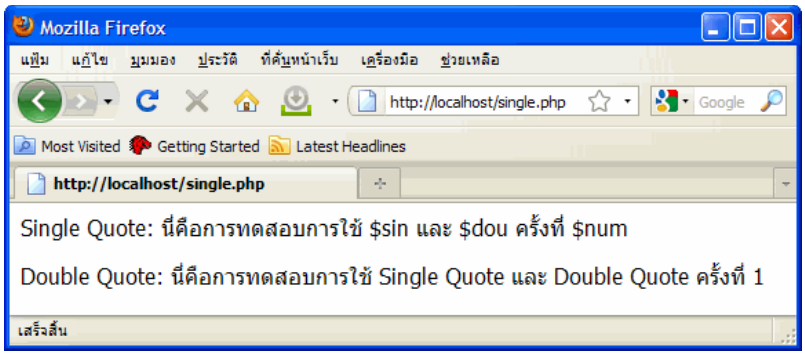
- ตัวแปรใน PHP จะต้องขึ้นต้นด้วยเครื่องหมาย “$” เช่น $name, $data, $_file เป็นต้น
- ต้องขึ้นต้นด้วยตัวอักษร a-z หรือ A-Z หรือเครื่องหมาย “-“ เท่านั้น ห้ามขึ้นต้นด้วยตัวเลขหรืออักขระอื่นๆ
- ตัวแปรใน PHP ไม่จำเป็นต้องระบุชนิดข้อมูล เนื่องจากตัวแปรแต่ละตัวสามารถเก็บข้อมูลชนิดใดก็ได้
- การเขียนตัวแปรด้วยลักษณะตัวพิมพ์ที่แตกต่างกัน ถือว่าเป็นตัวแปรคนละตัวไม่ใช่ตัวเดียวกัน เช่น $abc ,$ABC $aBC ถือเป็นคนละตัวกัน
- การกำหนดค่าให้กับตัวแปรชนิดสตริง(ตัวอักษร) จะต้องกำหนดให้อยู่ภายในเครื่องหมาย Single Quote (‘-’) หรือ Double Quote (“-”) เช่น $name = “วิโรจน์”
- การกำหนดค่าให้กับตัวแปรชนิดตัวเลข สามารถระบุค่าเข้าไปได้โดยไม่ต้องมีเครื่องหมาย Single Quote (‘-’) หรือ Double Quote (“-”) ถ้าใส่เครื่องหมายทั้งสองนี้ โปรแกรมจะถือว่าเป็นข้อมูลชนิดสตริงทันที เช่น $number = 12

เครื่องหมาย **Single Quote** และ **Double Quote**

ถ้าเราต้องการกำหนดค่าให้กับตัวแปรชนิดสตริงใน PHP สามารถใช้ได้ทั้ง Single Quote และ Double Quote แต่ทั้ง Single Quote และ Double Quote ก็มีข้อแตกต่างกันอยู่บ้างในเรื่องของการแสดงผลเมื่อเราต้องการกำหนดค่าให้กำตัวแปรเพื่อจะนำผลลัพธ์ไปแสดง ถ้าตัวแปรนั้นอยู่ภายใต้เครื่องหมาย **Single Quote** จะไม่สามารถแสดงค่าของตัวแปรได้เพราะสิ่งที่เราเขียนไปจะแสดงผลออกมาเช่นนั้นเปรียบเทียบจากตัวอย่างต่อไปนี้

```
1  <?
2  $sin = "Single Quote";
3  $dou = "Double Quote";
4  $num = 1;
5  echo 'Single Quote: นี่คือการทดสอบการใช้ $sin และ $dou ครั้งที่ $num';
6  echo "<p>";
7  echo "Double Quote: นี่คือการทดสอบการใช้ $sin และ $dou ครั้งที่ $num";
8  ?>
```

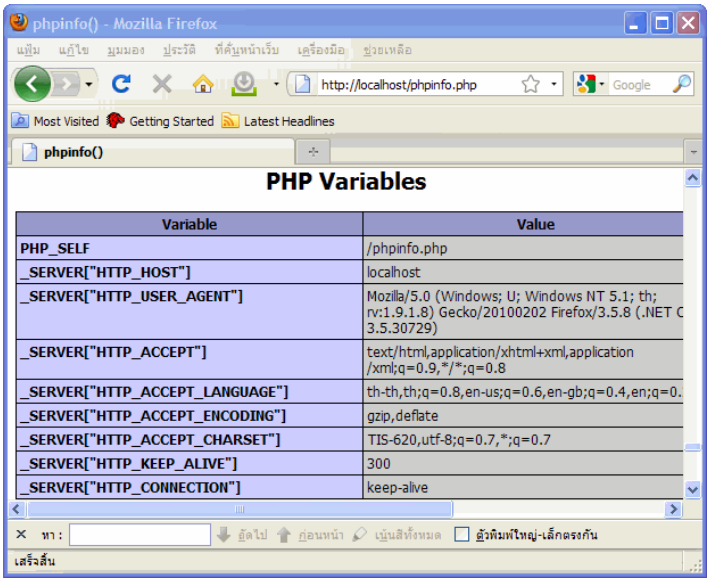
ผลลัพธ์



จากผลลัพธ์แสดงว่าเราไม่สามารถที่จะกำหนดค่าเพื่อแสดงภายใต้เครื่องหมาย Single Quote ได้

Predefined Variables

Predefined Variables คือตัวแปรที่ PHP สร้างไว้ล่วงหน้าแล้วเพื่อความสะดวกในการนำไปใช้และให้เป็นค่าพื้นฐานในการใช้งาน ส่วนใหญ่ตัวแปรเหล่านี้มักจะเก็บข้อมูลสำคัญบางอย่างเอาไว้ ตัวแปรชนิดนี้จะมีลักษณะเป็นอาร์เรย์มากกว่าแบบธรรมดา ซึ่งมีอยู่หลายตัวแต่ในที่นี้จะกล่าวถึงเฉพาะบางตัวที่สำคัญและจำเป็นต่อการใช้งาน เพื่อติดต่อกับ SCILAB เท่านั้น รายละเอียดเพิ่มเติมสามารถเรียกดูได้ที่ไฟล์ phpinfo.php ไปที่หัวข้อ PHP Variables



$_SERVER

ตัวแปร $_SERVER เป็นตัวแปรที่ใช้เก็บข้อมูลที่เกี่ยวข้องกับการเชื่อมต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์ เช่น ชื่อโฮสต์ พาธในการเก็บข้อมูล เมธอดในการส่งข้อมูล เป็นต้น ตัวแปร $_SERVER นี้เริ่มใช้ตั้งแต่ PHP 4.1.0 เป็นต้นมา โดยในเวอร์ชั่นก่อนหน้านี้จะใช้ชื่อตัวแปรนี้ว่า $HTTP_SERVER_VARS การอ้างถึงข้อมูลในตัวแปร $_SERVER จะใช้รูปแบบคือ

```
$_SERVER['variable_name'];
```

เมื่อ *variable_name* คือข้อมูลที่เราต้องการทราบ ซึ่งในการใช้งาน PHP เพื่อติดต่อกับ SCILAB นั้นมีการใช้งานดังนี้

$_SERVER['SystemRoot']

จะเก็บไดเรกเทอรีที่เก็บไฟล์ระบบปฏิบัติการ(System) เอาไว้

$_SERVER['DOCUMENT_ROOT']

ตัวแปรนี้จะเก็บ โฮมไดเรกเทอรี (Home Directory) คือไดเรกเทอรีที่เก็บไฟล์เว็บไซต์ที่เราสร้างขึ้นทั้งหมด

ชนิดข้อมูล **(Data Type)**

ชนิดข้อมูลพื้นฐานใน PHPมีดังนี้

ชนิดข้อมูล	คำอธิบาย
ตรรกศาสตร์	ค่าความจริงหรือที่เรียกว่าตรรกะ จะมีค่าเป็นจริง (True) หรือ เท็จ (False)
เลขจำนวนเต็ม	เป็นเลขจำนวนเต็มบวก จำนวนเต็มลบ หรือศูนย์
จำนวนทศนิยม(Float หรือ Double)	เป็นเลขทศนิยม
สตริง(String)	ค่าที่เป็นตัวอักษรทั้งหมด

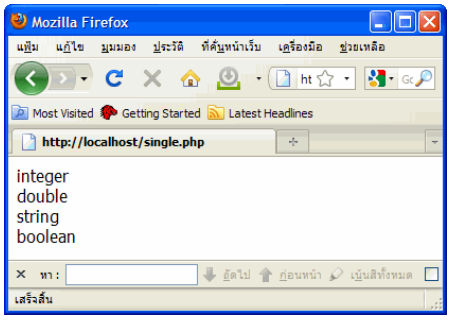
การตรวจสอบชนิดข้อมูล

ในจะใช้ฟังก์ชัน gettype() ดังตัวอย่าง

โค้ดโปรแกรม ผลลัพธ์

```php
<?
$a = -200;
$b = 0.25;
$c = "Test PHP";
$d = True;

echo gettype($a). "<br>";
echo gettype($b). "<br>";
echo gettype($c). "<br>";
echo gettype($d). "<br>";
?>
```



ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

ตัวดำเนินการ	ชื่อ	ตัวอย่าง (ให้ $a=20, $b=3)
+	บวก(Addition)	$a + $b = 23
-	ลบ(Subtraction)	$a - $b = 17
*	คูณ (Multiplication)	$a * $b = 60
/	หาร (Division)	$a / $b = 6.66
%	โมดูลัส (Modulus)การหาเศษจากการหาร	$a % $b = 2

ตัวดำเนินการเพิ่ม/ลดค่า **(Incrementing/Decrementing Operator)**

ได้แก่ตัวดำเนินการ ++ และ -- ซึ่งต้องการตัวถูกดำเนินการตัวเดียว (Unary Operator) และต้องเป็นตัวแปรเท่านั้น โดยจะเพิ่มค่าของตัวแปรขึ้น 1 และลดค่าของตัวแปรลง 1 ตามลำดับ

การใช้งานมีทั้งระบุไว้หน้าตัวแปรและหลังตัวแปร ซึ่งมีความหมายแตกต่างกัน ดังรายละเอียดในตาราง

ตัวอย่างการใช้งาน	ผลลัพธ์
++$a	เพิ่มค่าของ $a ขึ้น 1 ก่อน แล้วจึงกำหนดค่าให้กับ $a (++$a จะมีค่าเท่ากับ $a ที่เพิ่มค่าแล้ว)เช่น กำหนด $a = 200 ⇨++$a จะมีค่าเท่ากับ 201
$a++	ให้ค่าของ $a ก่อน แล้วจึงให้ค่าของ $a ($a++จะมีค่าเท่ากับ $a ก่อนเพิ่มค่า)เช่น กำหนด $a = 200 ⇨ $a++ จะมีค่าเท่ากับ 200
--$a	ลดค่าของ $a ลง 1 ก่อน แล้วจึงกำหนดค่าให้กับ $a (--$a จะมีค่าเท่ากับ $a ที่ลดค่าแล้ว) เช่น กำหนด $a = 200 ⇨ --$a จะมีค่าเท่ากับ 199
$a--	ให้ค่าของ $a ก่อน แล้วจึงลดค่าให้กับ $a ($a--จะมีค่าเท่ากับ $a ก่อนการลดค่า)เช่น กำหนด $a = 200 ⇨ $a--จะมีค่าเท่ากับ 200

ตัวดำเนินการเปรียบเทียบ (Comparison Operator)

ใช้เปรียบเทียบค่า 2 ค่าว่าเท่ากัน มากกว่า หรือ น้อยกว่า โดยผลลัพธ์ที่ได้จะเป็นค่าตรรกะ คือ จริง (True) และ เท็จ (False) อย่างใดอย่างหนึ่งเสมอ

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	คำอธิบาย
==	เท่ากับ (Equal)	$a == $b	ให้ค่า TRUE เมื่อ $a มีค่าเท่ากับ $b
===	เหมือนกับ (Identical)	$a === $b	ให้ค่า TRUE เมื่อ $a มีค่าเท่ากับ $b และข้อมูลต้องเป็นชนิดเดียวกัน
!= **หรือ** <>	ไม่เท่ากับ (Not equal)	$a != $b	ให้ค่า TRUE เมื่อ $a มีค่าไม่เท่ากับ $b
!==	ไม่เหมือนกับ (Not identical)	$a !=== $b	ให้ค่า TRUE เมื่อ $a มีค่าไม่เท่ากับ $b หรือเมื่อเป็นข้อมูลคนละชนิดกัน
<	น้อยกว่า (Less than)	$a < $b	ให้ค่า TRUE เมื่อ $a มีค่าน้อยกว่า $b
>	มากกว่า (Greater than)	$a > $b	ให้ค่า TRUE เมื่อ $a มีค่ามากกว่า $b
<=	น้อยกว่าหรือเท่ากับ (Less than or equal to)	$a <= $b	ให้ค่า TRUE เมื่อ $a มีค่ามากกว่าหรือเท่ากับ $b
>=	มากกว่าหรือเท่ากับ (Greater than or equal to)	$a >= $b	ให้ค่า TRUE เมื่อ $a มีค่าน้อยกว่าหรือเท่ากับ $b

ตัวดำเนินการทางตรรกศาสตร์

ตัวดำเนินการ	ชื่อ	ตัวอย่าง	คำอธิบาย
&&หรือ and	และ	$a&& $b	ให้ค่า TRUE ก็ต่อเมื่อ $a และ$bเป็นจริงทั้งคู่
|| หรือ or	หรือ	$a || $b	ให้ค่า TRUE เมื่อ $a หรือ$bตัวใดตัวหนึ่งเป็นจริง ถ้าตัวใดตัวหนึ่งเป็นเท็จ จะให้ค่า FALSE
xor	Exclusive or	$a xor $b	ให้ค่า TRUE เมื่อ $a หรือ$bตัวใดตัวหนึ่งเท่านั้น เป็น TRUE แต่ถ้าเป็น TRUE ทั้งคู่ หรือ เป็น FALSE ทั้งคู่ จะให้ค่าเป็น FALSE
!	นิเสธ	!$a	ให้ค่าตรงกันข้ามกับ $a ถ้า $a เป็น TRUE จะได้ FALSE แต่ถ้า $a เป็น FALSE จะได้ค่าเป็น TRUE

เงื่อนไขและการตัดสินใจ

เงื่อนไขและการตัดสินใจใน PHP ถือว่าสำคัญและจะใช้บ่อยมากในการเขียนโปรแกรม เพราะในการใช้งานจริงนั้นจะต้องมีการตรวจสอบเงื่อนไข เพื่อให้โปรแกรมที่เราเขียนขึ้นมานั้นทำงานได้ถูกต้องตามการออกแบบและความต้องการ ในที่นี้นั้นจะใช้หลักๆอยู่ดังนี้

- คำสั่ง if, else และ else if
- คำสั่ง switch

คำสั่ง if

คำสั่ง if จะตรวจสอบเงื่อนไขถ้าเงื่อนไขเป็นจริงก็จะทำงานในบรรทัดต่อไป ถ้าเป็นเท็จก็จะข้ามไปทำบรรทัดอื่น คำสั่ง if จะมีเครื่องหมายปีกา ({ }) หรือไม่มีก็ได้

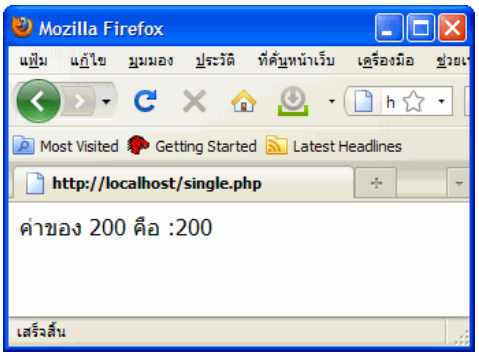
รูปแบบของคำสั่งคือ

```
if( เงื่อนไข){
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
}
```

ตัวอย่าง

```
<?
$a = 200;
if($a){
   echo "ค่าของ $a คือ :", $a;
}
?>
```

ผลลัพธ์



คำสั่ง if…else

คำสั่ง if…else นั้นมี 2 ทางเลือก จะทำการตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริงก็จะทำในคำสั่งเมื่อเงื่อนไขเป็นจริง ถ้าเงื่อนไขเป็นเท็จ ก็จะทำในงานในเงื่อนไขที่เป็นเท็จ

รูปแบบของคำสั่งคือ

```
if( เงื่อนไข){
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
}else{
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นเท็จ;
}
```

คำสั่ง if...elseif

คำสั่ง if…elseif นั้น มีมากว่า 2 ทางเลือก โดยจะทำการตรวจสอบเงื่อนไขไปเรื่อยๆ จนกว่าจะสิ้นสุดเงื่อนไขที่เรากำหนด

รูปแบบของคำสั่งคือ

```
If( เงื่อนไขที่ 1){
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
}elseif(เงื่อนไขที่ 2){
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
}elseif(เงื่อนไขที่ 3){
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
}else{
--- คำสั่งต่างๆเมื่อเงื่อนไขเป็นเท็จ;
}
```

คำสั่ง switch()

คำสั่ง switch() นั้นก็เป็นการติดสินใจแบบหลายทางเลือกโดยการตรวจสอบเงื่อนไขแล้วข้ามไปทำในคำสั่งที่เงื่อนไขเป็นจริง จากนั้นจะออกจาก คำสั่ง switch ไป

รูปแบบของคำสั่งคือ

```
switch(ค่าที่รับมา){
case เงื่อนไขที่ 1:
        --- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
        break;
case เงื่อนไขที่ 2:
        --- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
        break;
case เงื่อนไขที่ 3:
        --- คำสั่งต่างๆเมื่อเงื่อนไขเป็นจริง;
        break;
}
```

การทำซ้ำ (Repetition)

คือการประมวลผลคำสั่งที่ซ้ำๆกัน ตามเงื่อนไขที่เรากำหนด หรือเรียกกันว่าการวนลูป (Looping) การทำซ้ำหรือการวนลูปนี้มีประโยชน์มากในการเขียนโปรแกรม เพราะจะช่วยลดการเขียนโปรแกรมที่ซ้ำๆหลายบรรทัดรวมไว้ในลูป ทำให้การทำงานของโปรแกรมกระชับขึ้นการตรวจสอบหาข้อผิดพลาดในโค้ดของโปรแกรมก็กระทำได้สะดวกขึ้น

คำสั่ง while

คำสั่ง while จะทำการประมวลผลคำสั่งต่างๆไปเรื่อยๆจนกว่าเงื่อนไขจะเป็นเท็จ

รูปแบบของคำสั่งคือ

```
while ( เงื่อนไข){
--- คำสั่งต่างๆ;
}
```

ตัวอย่าง

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

คำสั่ง do…while

คำสั่ง do…while จะทำคำสั่งที่กำหนดก่อน แล้วทำการตรวจสอบเงื่อนไขทีหลัง ถ้าเงื่อนไขเป็นจริงจะกลับมาทำงานในคำสั่งอีกครั้ง ทำแบบนี้ไปเรื่อยๆจนกว่าจะเงื่อนไขจะเป็นเท็จ

รูปแบบ

```
do{
--- คำสั่งต่างๆ;
}while ( เงื่อนไข );
```

ตัวอย่าง

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

ฟังก์ชันเกี่ยวกับสตริง

ฟังก์ชันที่เกี่ยวของกับการจัดการสตริงใน PHP นั้นมีฟังก์ชันให้ใช้มากมาย ซึ่งในที่นี้นั้นเราจะนำมาแสดงเฉพาะที่น่าสนใจและเกี่ยวข้องกับการใช้งานร่วมกับ SCILAB เท่านั้น ซึ่งในการสร้างเว็บไซต์เพื่อติดต่อกับ SCILAB นั้น ฟังก์ชันที่เกี่ยวกับการจัดการสตริงนั้นจะมีการใช้งานบ่อยมาก เพราะ การแสดงผลลัพธ์ที่ส่งมาจากโปรแกรม SCILAB นั้น รูปแบบจะไม่คงที่และไม่สามารถนำมาแสดงผลบนเว็บเพจได้ทันที ต้องใช้ฟังก์ชันเหล่านี้ในการจัดการรูปแบบของผลลัพธ์เสียก่อน

ฟังก์ชัน strlen()

เป็นฟังก์ชันที่ใช้ในการหาความยาวของสติง หรือนับจำนวนสติง ซึ่งจะนับรวมช่องว่างเป็น 1 ตัวอักษรด้วย

รูปแบบ

```
strlen (สติง)
```

ตัวอย่าง

```
<?
echostrlen("Hello world!");
?>
```

ผลลัพธ์

```
Hello world!
```

ฟังก์ชัน explode()

เป็นฟังก์ชันในการแยกสตริงออกเป็นสตริงย่อยๆ ด้วยการกำหนดสตริงหรือสัญลักษณ์ที่จะใช้แยก ผลลัพธ์ที่ได้จะถูกเก็บอยู่ในรูปแบบของ Array

รูปแบบคือ

```
$result = explode (สัญลักษณ์ที่ใช้แยก, สตริง)
```

ตัวอย่าง

```
<?
$str = "Hello world.";
print (explode(" ",$str));
?>
```

ผลลัพธ์

```
Array ( [0] => Hello [1] => world. )
```

ฟังก์ชัน implode() หรือ join()

ฟังก์ชันนี้จะทำงานตรงกันข้ามกับ explode() คือเป็นการนำสตริงย่อยๆมารวมกัน ตามสัญลักษณ์ที่เราระบุ

รูปแบบคือ

```
$result = implode (สัญลักษณ์ที่ใช้รวม, อาร์เรย์ของสตริง)
```

ตัวอย่าง

```
<?
$arr = array('Hello','World!','Scilab','&','PHP!');
echo implode(" ",$arr);
?>
```

ผลลัพธ์

```
Hello World! Scilab& PHP!
```

ฟังก์ชัน strstr()

ฟังก์ชันนี้เป็นการตัดเอาเฉพาะสตริงที่เริ่มตั้งแต่สตริงย่อยที่เรากำหนดไปจนถึงจุดสิ้นสุดของสตริงนั้นๆ

รูปแบบคือ

```
strstr (สตริงย่อย,สตริงหลัก)
```

ตัวอย่าง

```
<?
echostrstr("Hello world!","w");
?>
```

ผลลัพธ์

```
world!
```

ฟังก์ชัน substr()

เป็นฟังก์ชันที่ใช้ตัดสตริงคล้ายๆกับ strstr() แต่สามารถกำหนดความยาวของสตริงที่ต้องการได้

รูปแบบคือ

```
substr (สตริงหลัก, ตำแหน่งเริ่มต้น, ความยาว)
```

ตัวอย่าง

```
<?
echosubstr("Hello world!",6,5);
?>
```

ผลลัพธ์

```
world
```

ฟังก์ชัน ltrim(), rtrim() และ trim()

ฟังก์ชันนี้ใช้สำหรับการตัดช่องว่างที่อยู่ด้านซ้าย,ด้วนขวา, และทั้งสองข้างของสตริงออกทั้งหมด

รูปแบบคือ

```
ltrim(สตริง)
rtrim(สตริง)
trim(สตริง)
```

ตัวอย่าง

```
<?
$str = "  Hello  ";
echo ltrim($str)."<br>";
echo rtrim($str)."<br>";
echo trim($str)."<br>";
?>
```

ผลลัพธ์

```
“Hello  “
“  Hello“
“ Hello  “
```

การจัดการไฟล์และไดเรกเทอรี่

การเปิดไฟล์, สร้างไฟล์

ในการอ่านข้อมูลจากไฟล์หรือบันทึกข้อมูลลงไฟล์ จะต้องเริ่มด้วยการเปิดไฟล์ขึ้นมาก่อนโดยใช้ฟังก์ชัน fopen() ซึ่งมีรูปแบบการใช้งานดังนี้

```
int fopen (string filename, string mode [,int use_include_path])
```

filename ชื่อไฟล์ที่ต้องการเปิดหรือสร้างขึ้นใหม่

mode วิธีการเปิดไฟล์ ซึ่งมีหลายกรณี ดังตาราง

'r'	**อ่านอย่างเดียว ตัวชี้จะเริ่มที่ตำแหน่งเริ่มต้นของไฟล์**
'r+'	อ่านและเขียนไฟล์ ตัวชี้จะเริ่มที่ตำแหน่งเริ่มต้นของไฟล์
'w'	เขียนได้อย่างเดียว ตัวชี้จะเริ่มที่ตำแหน่งเริ่มต้นของไฟล์ และถ้าเป็นไฟล์ที่มีอยู่แล้วจะตัดข้อมูลเดิมทิ้งทั้งหมด แต่ถ้าไม่มีไฟล์นี้อยู่จะสร้างไฟล์ขึ้นมาใหม่
'w+'	อ่านและเขียนไฟล์ ตัวชี้จะเริ่มที่ตำแหน่งเริ่มต้นของไฟล์ และถ้าเป็นไฟล์ที่มีอยู่แล้วจะตัดข้อมูลเดิมทิ้งทั้งหมด แต่ถ้าไม่มีไฟล์นี้อยู่จะสร้างไฟล์ขึ้นมาใหม่
'a'	เขียนได้อย่างเดียว ตัวชี้จะเริ่มที่ตำแหน่งสิ้นสุดไฟล์ และถ้าไม่มีไฟล์นี้อยู่จะสร้างไฟล์ขึ้นมาใหม่
'a+'	อ่านและเขียนไฟล์ ตัวชี้จะเริ่มที่ตำแหน่งสิ้นสุดไฟล์ และถ้าไม่มีไฟล์นี้อยู่จะสร้างไฟล์ขึ้นมาใหม่
'x'	เขียนได้อย่างเดียว ตัวชี้จะเริ่มที่ตำแหน่งเริ่มต้นของไฟล์ และถ้าไม่มีไฟล์นี้อยู่จะสร้างไฟล์ขึ้นมาให้ใหม่ แต่ถ้ามีไฟล์อยู่แล้วจะคืนค่า false และจะแจ้งข้อผิดพลาด
'x+'	อ่านและเขียนไฟล์ ตัวชี้จะเริ่มที่ตำแหน่งเริ่มต้นของไฟล์ และถ้าไม่มีไฟล์นี้อยู่จะสร้างไฟล์ขึ้นมาให้ใหม่ แต่ถ้ามีไฟล์อยู่แล้วจะคืนค่า false และจะแจ้งข้อผิดพลาด

use_include_path ระบุค่าเป็น เมื่อต้องการให้ค้นหาไฟล์จากที่ระบุไว้ใน 1include_pathในไฟล์คอนฟิกกูเรชัน (configuration file)

ตัวอย่างการใช้งานเช่น

```
$file = @fopen("test.txt", "r");
```

การอ่านไฟล์

การอ่านไฟล์สามารถทำได้หลายวิธี ได้แก่การอ่านไฟล์ด้วยฟังก์ชัน fgets(), fgetc(), fgetss(), fgetcsv() เพื่ออ่านข้อมูลมาเก็บไว้ในตัวแปร และเมื่อเลิกใช้ไฟล์แล้วควรปิดไฟล์ด้วยฟังก์ชัน fclose ซึ่งจะให้ค่าเป็น true การปิดไฟล์สำเร็จ และให้ค่าเป็น false หารการปิดไฟล์ล้มเหลว

ฟังก์ชั่นที่ใช้ในการอ่านไฟล์ทั้งสี่ มีหน้าที่และวิธีการใช้ดังนี้

ฟังก์ชั่น fgets()

รูปแบบ

```
string fgets (int fp [, int length ])
```

fp ไฟล์พอยเตอร์

length ความยาวของข้อความ (ในหน่วยไบต์) ที่ให้อ่าน โดยเริ่มจากตำแหน่งของตัวชี้เป็นจำนวนเท่ากับ length – 1 ไบต์ แต่หากไม่ระบุ ค่าโดยปริยายของพารามิเตอร์ length นี้คือ 1024

ฟังก์ชัน fgets จะอ่านข้อความทีละ 1 บรรทัด นับจากตัวชี้ปัจจุบันและไม่เกินจำนวน length-1 ไบต์

สัญลักษณ์ @ เมื่อนำมาใช้เป็น prefix ของ expressions จะทำหน้าที่เป็น Error Control Operator เพื่อไม่ให้มีการรายงานข้อผิดพลาดในกรณีที่การทำงานล้มเหลว

ตัวอย่าง การอ่านข้อความในไฟล์ fgets.txt ครั้งละไม่เกิน 1024 ไบต์

```php
<?php
    $file = fopen("fgets.text", "r");
    while ($data = fgets($file, 1024)){
        echo $data;
    }
?>
```

ฟังก์ชั่น fgets() จะอ่านข้อความจากไฟล์เข้ามาทีละบรรทัด (แต่บรรทัดละไม่เกิน 1023 ไบต์) แล้วส่งคืนข้อความนั้นกลับออกไปจากฟังก์ชัน ข้อความจะถูกนำไปกำหนดให้กับตัวแปร $data ซึ่งสำหรับข้อความใด ๆ ที่ไม่ใช่สตริงว่าง (“”) จะถูกประเมินค่าเป็น true เสมอ จึงเกิดการวนลูปต่อไปเรื่อย ๆ แต่เมื่อฟังก์ชั่น fgets() อ่านไปจนกระทั่งพลจุสิ้นสุดของไฟล์ (End Of File – EOF) จะส่งคืนค่าสตริงว่างออกไป ซึ่งจะถูกประเมินเป็นค่า false และทำให้ลูปจบการทำงาน

ฟังก์ชัน fgetc()

ฟังก์ชั่น fgetc() จะอ่านข้อความทีละตัวอักษร มีรูปแบบการใช้ดังนี้

รูปแบบ

```
string fgetc (int fp)
```

fp ไฟล์พอยเตอร์

หากพบจุดสิ้นสุด(End Of File – EOF) จะคืนค่าเป็น false

ฟังก์ชัน fgetss()

รูปแบบการใช้ฟังก์ชัน fgetss() PHP 4

รูปแบบ

```
stringfgetss (int fp, int length [, string allowable_tags])
```

รูปแบบการใช้ฟังก์ชัน fgetss() PHP 5

```
string fgetss (int fp[, int length [, string allowable_tags]])
```

fp ไฟล์พอยเตอร์

length ความยาวของข้อความ (ในหน่วยไบต์) ที่ให้อ่าน โดยเริ่มจากตำแหน่งของตัวชี้เป็นจำนวนเท่ากับ length – 1 ไบต์ (ใน PHP5 พารามิเตอร์ length จำเป็น optional parameter คือจะใส่หรือไม่ก็ได้)

allowable_tags ข้อความระบุแท็กในภาษา HTML ที่ต้องการให้แสดง

ฟังก์ชัน fgetcsv()

รูปแบบการใช้ฟังก์ชัน fgetcsv() PHP4

รูปแบบ

```
srrayfgetcsv (resource handle, intlength [, string delimiter[, string enclosure]]])
```

รูปแบบการใช้ฟังก์ชัน fgetcsv() PHP5

รูปแบบ

```
srrayfgetcsv (resource handle [, intlength[, string delimiter [, string enclosure]]])
```

fd ไฟล์พอยเตอร์

length ความยาวของข้อความ (ในหน่วยไบต์) ที่ให้อ่าน โดยเริ่มจากตำแหน่งของตัวชี้เป็นจำนวนเท่ากับ length – 1 ไบต์ ค่าของพารามิเตอร์ length นี้จะต้องมากกว่าความยาวของบรรทัดที่ยาวที่สุดในไฟล์ สำหรับ PHP5 นั้นพารามิเตอร์ length จำเป็น optional parameter คือจะระบุหรือไม่ก็ได้

delimiter ตัวแบ่งข้อความ (ขนาด 1 ตัวอักษร) หากไม่ระบุไว้ ค่าโดยปริยายคือ เครื่องหมาย comma (,)

enclosure เครื่องหมายสำหรับใช้แบ่งข้อความ หากไม่ระบุ ค่าโดยปริยายคือเครื่องหมาย double quote (“) (พารามิเตอร์ enclosure นี้เริ่มมีใช้ใน PHP4.3.0)

ฟังก์ชัน fgetcsv() จะอ่านข้อความทีละบรรทัดและไม่เกิน length – 1 ไบต์ เช่นเดียวกับฟังก์ชัน fgets() แต่ฟังก์ชัน fgetcsv() จะสามารถอ่านไฟล์ในฟอร์แมต CSV (Comma –Separated Values) ซึ่ง แต่ละข้อความจะถูกแบ่งด้วยตัวแบ่ง delimiter ได้ โดยฟังก์ชั่น fgetcsv() จะอ่านข้อความแต่ละบรรทัดแล้ว ให้ค่าเป็นอาร์เรย์ของแต่ละข้อความที่ถูกคั่นด้วย delimiter

การเขียนลงไฟล์

การเขียนข้อมูลลงเท็กซ์ไฟล์ด้วยฟังก์ชัน fputs() มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
intfputs (int fp, string str [, int length])
```

fp ไฟล์พอยเตอร์

str ข้อความที่ต้องการเขียนลงไฟล์

length ความยาวของข้อความ ที่ให้เขียน (ในหน่วยไบต์)

ตัวอย่าง การเขียนข้อมูลลงไฟล์ด้วยฟังก์ชัน fputs()

```
<? php
   $linklist[] = http://www.php.net;
   $linklist[] = http://www.zend.com;
   $linklist[] = http://www.apache.org;
   $linklist[] = http://www.linux.org;
   $fp = @fopen("link.htm", "w");
foreach ($linklist as $eachlink) {
fpust($fp, "<a href='$eachlink'>$eachlink</a><br>\n";
   }
@fclose($fp) or die("Cannot close file.");
?>
```

ผลลัพธ์ของโปรแกรม

โปรแกรมข้างบนนี้จะสร้างไฟล์ link.htm ขึ้นมา และเขียนข้อมูลที่เก็บอยู่ในตัวแปรอาร์เรย์ $linklist ลงไป ซึ่งสุดท้ายแล้วจะได้แท็กซ์ไฟล์ที่มีข้อความดังนี้

```
<a href='http://www.php.net'>http://www.php.net</a><br>
<a href='http://www.zend.com'>http://www.zend.com</a><br>
<a href='http://www.apache.org'>http://www.apache.org</a><br>
```

http://www.linux.org

ซึ่งสามารถนำมาแสดงผลบนบราวเซอร์ได้

การอ่านและเขียนไบนารีไฟล์

ในการอ่านและเขียนไบนารีไฟล์นั้น แทนที่จะใช้ฟังก์ชัน fgets() และ fputs() ควรจะใช้ฟังก์ชั่น fread() และ fwrite() และหากระบบปฏิบัติการที่ใช้มีความแตกต่างระหว่างไบนารีไฟล์กับแท็กซ์ไฟล์ก็จำเป็นต้องเพิ่มตัวอักษร b เข้าไปในอาร์กิวเมนต์ mode ของฟังก์ชั่น fopen() ด้วย

ฟังก์ชัน fread()

รูปแบบการใช้ฟังก์ชัน fread()

รูปแบบ

string fread (int *fp*, int *lengh*)

fp ไฟล์พอยเตอร์

length ความยาวของข้อความ ที่ให้อ่าน (ในหน่วยไบต์)

ฟังก์ชัน fread() จะอ่านไฟล์เป็นจำนวน length ไบต์นับจากตำแหน่งที่ไฟล์พอยเตอร์ชี้อยู่ และจะหยุดอ่านเมื่อพบจุดสิ้นสุดของไฟล์ (EOF)

ฟังก์ชัน fwrite()

รูปแบบการใช้ฟังก์ชัน fwrite()

รูปแบบ

int fwrite (int *fp*, string *string* [, int *length*])

fp ไฟล์พอยเตอร์

string ข้อความที่ต้องการเขียนลงไฟล์

length ความยาวของข้อความ ที่ให้เขียน (ในหน่วยไบต์)

ฟังก์ชัน fwrite() จะเขียนข้อมูลลงไฟล์เป็นจำนวน length ไบต์หรือจนสิ้นสุดข้อความใน อาร์กิวเมนต์ string และจะคืนค่าเป็นจำนวนไบต์ที่ถูกเขียนลงไฟล์ หรือหากมีข้อผิดพลาดจะคืนค่าเป็น -1

ตัวอย่างที่ 7.5 การทำสำเนาไฟล์ด้วยฟังก์ชัน fread() และ fwrite()

```
<? php
  $fp = @fopen(“sample.zip”, “rb”);
  $fpnew = @fopen(“new_sample.zip”,”wb”);
while ($ln = @fread($fp, 1024 )) {
fwrite ($fpnew,$ln);
  }
@fclsoe($fp) or die (“Cannot close old file.”);
@fclose($fpnew) or die (“Cannot cllse new file.”);
?>
```

ผลลัพธ์ของโปรแกรม

ในตัวอย่างนี้จะทำการคัดลอกข้อมูลจากไฟล์ sample.zip ครั้งละ 1024 ไบต์ นำมาเก็บไว้ในตัวแปร $ln แล้วจึงเขียนลงไฟล์ new_sample.zip ขึ้นมาใหม่ซึ่งมีข้อมูลเหมือนกับไฟล์ sample.zip ทุกประการ

ตรวจสอบการมีอยู่ของไฟล์

รูปแบบ

```
boolfile_exists( string filename)
```

filename ชื่อไฟล์ที่ต้องการตรวจสอบ

```php
<?php
    $file = "fgets.php";
    if(file_exists($file)){
        echo "พบไฟล์ชื่อ $file";
    }else{
        die("ไม่พบไฟล์ $file ในระบบ" );
    }
?>
```

ผลลัพธ์

พบไฟล์ชื่อ fgets.php

การเรียกใช้งานไฟล์ ด้วยคำสั่ง

```php
<?php
    $file = @fopen("file_load.sce", 'w');
    @fwrite($file, "log2(128)");
    fclose($file) ;
    shell_exec("file_load.sce");
?>
```

การลบไฟล์

การลบไฟล์ใน PHP จะใช้ฟังก์ชัน unlink()

รูปแบบ

```
int unlink(string filename)
```

ถ้าการลบไฟล์สำเร็จจะให้ค่าเป็น true และถ้าล้มเหลวจะให้ค่าเป็น false

ฟังก์ชัน exec()

เป็นฟังก์ชันที่ใช้รันโปรแกรมภายนอก หรือคำสั่งระบบ

รูปแบบคือ

```
String exec( string$command [, array &$output [, int&$return_var]] );
```

ฟังก์ชันที่เกี่ยวกับโปรแกรม SCILAB

ฟังก์ชัน file เป็นฟังก์ชันที่ใช้ในการเปิดหรือปิดแฟ้มข้อมูลหรือไฟล์

รูปแบบคือ

```
unit = file('open', filename, [status])
```

โดยที่พารามิเตอร์

- 'open' เป็นการบอกโปรแกรมให้เปิดไฟล์ filename ขึ้นมาใช้งาน
- filename เป็นชื่อไฟล์ข้อมูลที่ต้องการจะเรียกขึ้นมาใช้งาน
- status เป็นการกำหนดสถานะของไฟล์ที่เปิดขึ้นมาใช้งานซึ่งมีอยู่4รูปแบบคือ
  - "new" หมายถึงไฟล์ที่เปิดขึ้นมาจะต้องไม่เคยมีอยู่ในไดเร็กทอรีทำงาน
  - "old" หมายถึงไฟล์ที่เปิดขึ้นมาจะต้องมีอยู่แล้วในไดเร็กทอรีทำงาน
  - "unknown" หมายถึงไฟล์ที่เปิดขึ้นมาจะมีอยู่หรือไม่มีอยู่ในไดเร็กทอรีทำงานก็ได้
  - "scratch" หมายถึงไฟล์ที่เปิดขึ้นมาจะถูกลบทิ้งหลังจากเสร็จสิ้นการทำงาน
- Unit เป็นเลขจำนวนเต็มที่โปรแกรมใช้อ้างถึงชื่อไฟล์ filename นั้น

หมายเหตุหลังจากเสร็จสิ้นการใช้งานไฟล์ที่เปิดขึ้นมาแล้ว จะต้องทำการปิดการทำงานของไฟล์นั้นด้วยเสมอ โดย ใช้ฟังก์ชันดังนี้

```
file(‘close', unit)
```

ในทางปฏิบัติฟังก์ชันทั้งสองนี้มักจะใช้งานคู่กันเสมอ

ฟังก์ชัน fprintf

เป็นฟังก์ชันที่ทำหน้าที่พิมพ์ค่าของตัวแปรลงไปเก็บไว้ในไฟล์ แทนที่จะแสดงผลออกมาที่หน้าต่างฟังก์ชัน

รูปแบบคือ

```
fprintf(file, format, value_1,..., value_n)
```

โดยที่พารามิเตอร์

- *file* เป็นชื่อของไฟล์ที่ต้องการจะให้เก็บค่าของตัวแปร value_1 – n
- *format* รหัสควบคุมการพิมพ์
- *value_i - n* เป็นตัวกำหนดว่าจะให้ข้อมูลใดแสดงผลออกมาที่หน้าต่างคำสั่ง

ตารางที่ 1.1 รหัสรูปแบบในโปรแกรม SCILAB

รหัสรูปแบบ	คำอธิบาย
%d	แสดงผลเป็นเลขจำนวนเต็มฐานสิบแบบมีเครื่องหมาย (signed integer)
%u	แสดงผลเป็นเลขจำนวนเต็มฐานสิบแบบไม่มีเครื่องหมาย (unsigned integer)
%x หรือ %x	แสดงผลเป็นเลขจำนวนเต็มฐานสิบหกแบบไม่มีเครื่องหมาย
%f	แสดงผลเป็นเลขจำนวนจริง
%e	แสดงผลเป็นเลขจำนวนจริงในรูปของเลขยกกำลัง
%c	แสดงผลตัวอักขระ
%s	แสดงผลสายอักขระ

ตารางที่ 1.2 รหัสบังคับการพิมพ์ในโปรแกรม SCILAB

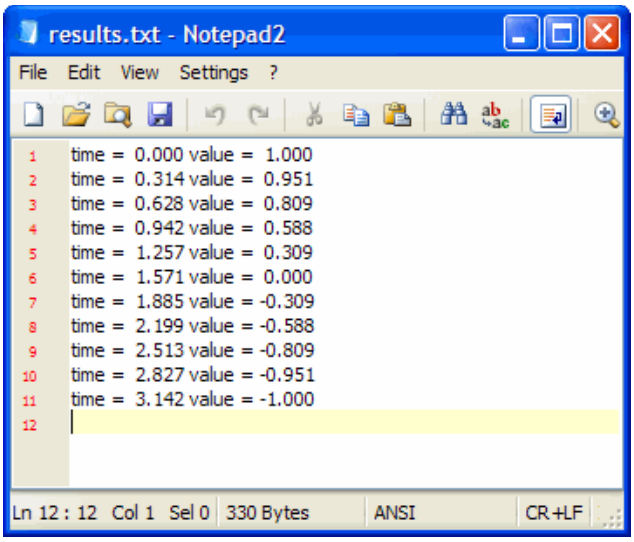
รหัสบังคับการพิมพ์	คำอธิบาย
\n	ขึ้นบรรทัดใหม่
\t	แท็ป (tab) ในแนวนอน
\v	แท็ปในแนวตั้ง
\b	เลื่อนเคอร์เซอร์ไปลบตัวอักขระทางซ้ายมือหนึ่งตัวอักขระ
\r	เครื่องหมาย return เหมือนกับการกดปม Enter
\f	ขึ้นหน้าใหม่
\a	ส่งเสียงดังออกลำโพงหนึ่งครั้ง
\\	เครื่องหมาย \ (backslash)
\’	เครื่องหมาย ' (single quote)
\”	เครื่องหมาย " (double quote)
\?	เครื่องหมาย ? (question mark)
\000	พิมพ์ตัวอักขระที่มีเลขฐานแปดตรงกับค่า 000
\xhh	พิมพ์ตัวอักขระที่มีเลขฐานสอบหกตรงกับค่า hh

ตัวอย่างการใช้งาน

เปิดโปรแกรม SCILAB แล้วพิมพ์คำสั่งดังนี้

```
-->u = file('open', 'results.txt', 'unknown');
-->for t = 0:%pi/10:%pi
-->fprintf(u, 'time = %6.3f value = %6.3f\n', t, cos(t));
-->end
-->file('close', u);
```

จากนั้นทดลองเปิดไฟล์ results.txt จะได้ผลลัพธ์ดังภาพ



คำสั่ง exec

เป็นคำสั่งที่ใช้ในการโหลดฟังก์ชันต่างๆ เข้ามาประมวลผลรวมกับโปรแกรม SCILAB โดยมีลักษณะดังนี้

```
exec(path)
```

โดยที่พารามิเตอร์

path เป็นชื่อและที่อยู่ของไฟล์

ตัวอย่างการใช้งาน

```
-->exec(' test.sci');
```

คำสั่ง savematfile

เป็นคำสั่งที่ใช้ในการบันทึกข้อมูลให้อยู่ในรูปแบบ binary หรือ ASCII ได้ โดยมีรูปแบบดังนี้

```
savematfile('filename','var1','var2')
```

โดยที่พารามิเตอร์

- *filename* ชื่อและทีอยู่ของไฟล์
- *var1*…*var2* เป็นตัวแปรที่จะเขียนลงในไฟล์

ตัวอย่างการใช้งาน

savematfile(‘test.txt’, ‘ans’)

คำสั่ง xs2gif

เป็นคำสั่งที่ใช้ในการบันทึกกราฟต่างๆ ที่วาดขึ้นเป็นไฟล์รูปภาพที่มีนามสกุล .gif

รูปแบบการใช้งาน

```
xs2gif(win_num, filen)
```

โดยที่พารามิเตอร์

- *win_num* คือ หน้าต่างกราฟฟิกตั้งแต่ 0.....N
- *filen* คือ ชื่อไฟล์ที่เราต้องการบันทึก นามสกุลเป็น .gif

ตัวอย่างการใช้งาน

```
xs2gif(0,'00 l.gif)
```

Chapter 3

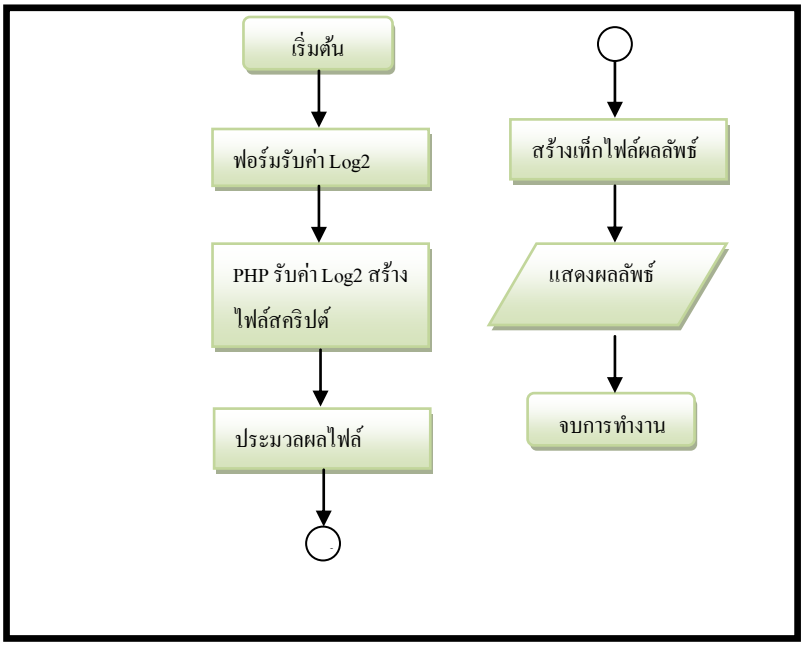
การเขียนเว็บ Applications ด้วย PHP ติดต่อกับ Scilab

ในบทนี้เราจะนำองค์ความรู้ที่กล่าวมาข้างต้นมาประยุกต์ใช้งานร่วมกับโปรแกรม SCILAB เพื่อประมวลผลคำสั่งและฟังก์ชันทางคณิตศาสตร์ โดยจะเริ่มจากหาค่า Log ฐาน 2 เพื่อให้เข้าใจได้ง่ายเพราะไม่ซับซ้อนมากนัก ลำดับขั้นตอนการเขียนโปรแกรมมีดังนี้

ขั้นตอนการเขียนโปรแกรม

1. สร้างไฟล์ HTML ขึ้นมาเพื่อใช้ในการจัดการข้อมูลที่จะส่งไปยังไฟล์ PHP เพื่อสร้างไฟล์สคลิปต์ของโปรแกรม SCILAB
2. เขียนโปรแกรมประมาลผลไฟล์สคลิปต์ด้วยคำสั่งของ PHP ในการเรียกใช้งานโปรแกรมภายนอก
3. เขียนโปรแกรม PHP ในการอ่านข้อมูลจากแท็กไฟล์นำมาแสดงผลทางหน้าเว็บเพจ

แผนผังระบบการทำงาน (System Flowchart)



1. การสร้างฟอร์ม เพื่อรับค่า Log 2 ส่งไปยังไฟล์ PHP

ในตัวอย่างการหาค่า Log ฐาน 2 เราจะทำการสร้างฟอร์ม ฟอร์มนี้จะอยู่ในไฟล์ HTML จากนั้นส่งค่าไปยังไฟล์ PHP ที่ทำหน้าที่ในการสร้างไฟล์สคริปต์ ที่เป็นคำสั่งของโปรแกรม SCILAB เพื่อหาค่า Log 2 ซึ่งในโปรแกรม SCILAB จะมีฟังก์ชัน Log2(x) ดังนั้นเราจะรับค่าเป็นตัวเลขมาจากฟอร์มแล้วทำการเขียนลงไปใน ตัวแปร x

นิยามลอการิทึม

ลอการิทึม เป็นการดำเนินการทางคณิตศาสตร์ ที่เป็นฟังก์ชันผกผันของ ฟังก์ชันเอ็กซ์โพเนนเชียล (การใช้ ค่าคงตัว หรือ "ฐาน" เป็นเลขยกกำลัง) ลอการิทึมของจำนวน **x** ที่มีฐาน **b** คือจำนวน **x** นั่นคือ **x = bn** เขียนได้เป็น

$$\log_b(x) = n$$

ตัวอย่างเช่น

$\text{Log}_2(128) = 7$

เพราะว่า 2 x 2 x 2 x 2 x 2 x 2 x 2 = 128

ขั้นตอนการเขียนโปรแกรม

1.1 เปิดโปรแกรม TextEditer จากนั้นพิมพ์โค้ดดังนี้

```
<html>
<head>
<title>Test Log 2</title>                          <!--แสดงข้อความที่ Title Bar -->
</head>
<body>
<form name="frm1" method="post" action="log2.php">  <!--สร้างฟอร์มชื่อ "frm1" กำหนดการส่งข้อมูลแบบ post และให้ส่งข้อมูลไปยังไฟล์ log2.php-->
กรุณากรอกตัวเลข Log <sub>2</sub>                    <!--แสดงข้อความ-->
<input type="text" id="log2" name="log2"></input>  <!--สร้างกล่องรับข้อมูบเป็นชนิด TextField และตั้งชื่อว่า TextField log2-->
<input type="submit" id="bntLog2" value="Submit"></input> <!--สร้างปุ่ม Submit -->
</form>                                            <!--สิ้นสุด tag form -->
</body>
</html>
```

<u>รูปที่ 3.1</u> แสดงโค้ดโปรแกรม

อธิบายโปรแกรม

เนื่องจากหน้าเว็บเพจนี้เป็นการสร้างฟอร์ม และเป็นคำสั่ง HTML ทั้งหมด

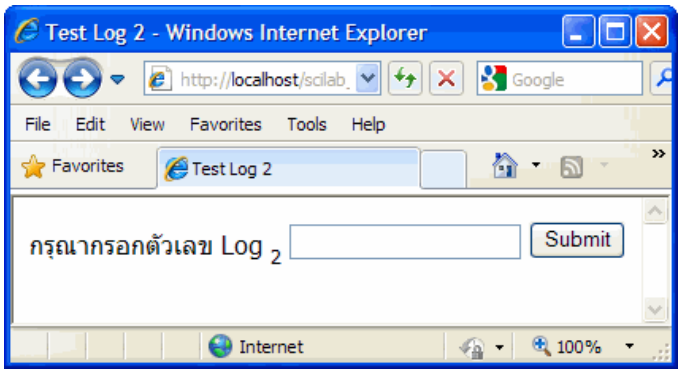
บรรทัดที่ 2-4 เป็นส่วนหัว (Header) ของหน้าเว็บเพจและจะแสดงข้อความ Test Log 2 ที่ Title Bar

บรรทัดที่ 5-11 เป็นส่วนของ Body ในส่วนของ Body นั้น ก็จะสร้างฟอร์มด้วย tag <form>…</form>

1.2 ทำการ Save ไฟล์ ตั้งชื่อไฟล์ เป็น log2.html ไปไว้ที่ไดเรกเทอรี่ C:\Appserv\www\scilab_php (ในที่นี้จะสร้าง *Folder* ชื่อ *scilab_php* สำหรับเก็บไฟล์ต่างๆที่เราสร้างขึ้นเพื่อให้เป็นระเบียบและเข้าใจได้ง่าย)

1.3 ทำการเรียกดูหน้าเว็บเพจที่สร้างขึ้น เปิด Browser พิมพ์ URL ดังนี้

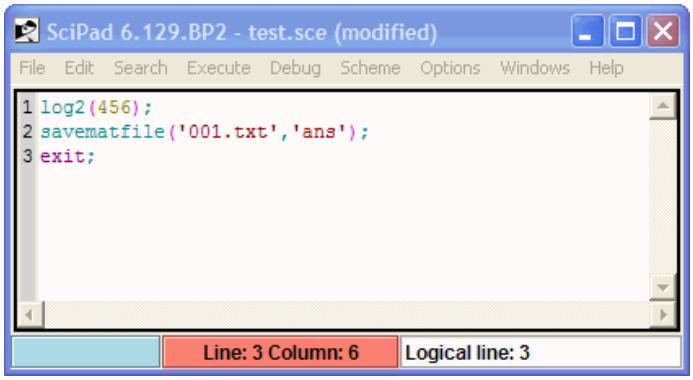
http://localhost/scilab_php/log2.php จะได้หน้าเว็บดังรูป



<u>รูปที่ 3.2</u> แสดงฟอร์มการกรอกข้อมูล

2. เขียนโปรแกรมประมวลผลไฟล์สคลิปต์ด้วยคำสั่งของ PHP

เมื่อเราสร้างฟอร์มในการรับข้อมูลแล้ว เราต้องส่งข้อมูลไปยังเครื่อง Server เพื่อประมวลผลไฟล์ ซึ่งฝั่ง Server นี้จะใช้ PHP เป็นตัวรับข้อมูลและสร้างไฟล์สคริปต์ นามสกุล .sce เพื่อประมวลผลด้วยโปรแกรม SCILAB แต่ก่อนที่จะสร้างไฟล์สคริปต์นั้น ต้องรู้ก่อนว่าคำสั่งมีอะไรบ้าง จากตัวอย่างจะได้ดังนี้



<u>รูปที่ 3.3</u> แสดงคำสั่งทั้งหมดของไฟล์สคริปต์

ขั้นตอนการเขียนโปรแกรม

2.1 เปิดโปรแกรม TextEditer จากนั้นพิมพ์โค้ดดังนี้

```
1  <?php
2  $log2 = $_POST['log2'];          //ประกาศตัวแปร $log 2 รับค่า log 2 ที่ส่งมาจากฟอร์ม
3  $data = "log2($log2)";           //สร้างตัวแปร $data เก็บข้อมูลที่ได้รับ
4
5  $file = @fopen("test.sce", "w"); //สร้างตัวแปร $file เก็บคำสั่งสร้าง และ เปิดไฟล์ไว้
6    @fwrite($file, $data);         //สร้างไฟล์ test.sce แล้วเขียนไฟล์
7  ?>
```

<u>รูปที่ 3.4</u> แสดงโค้ดการสร้างไฟล์สคริปต์

อธิบายโปรแกรม

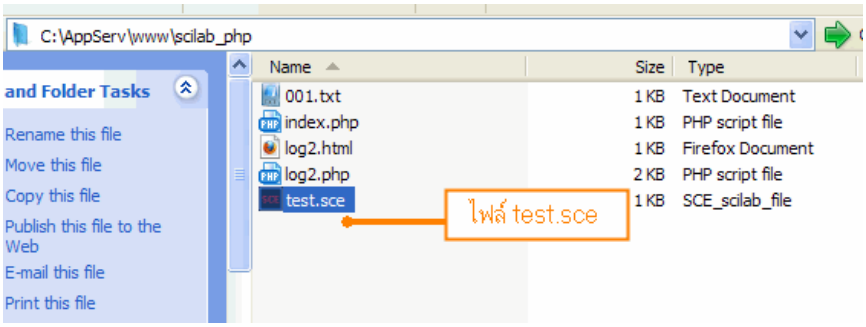
ในขั้นตอนนี้จะเขียนโค้ดเพื่อรับค่าจากเพจ log2.html ที่ส่งมา จากนั้นสร้างไฟล์เท็กขึ้นมาชื่อ test.sce นำค่าที่รับมาเขียนลงไฟล์

บรรทัดที่ 3-4 กำหนดตัวแปร $log2 เพื่อเก็บค่าที่รับมาจากเพจ log2.html แล้วนำค่าที่ได้ไปรวมกับคำสั่งที่จะเขียนลงในไฟล์สคริปต์

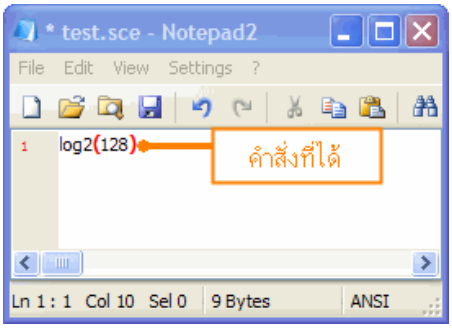
บรรทัดที่ 7-8 สร้างไฟล์สคริปต์ test.sce ด้วยคำสั่ง fopen() ระบุ mode ให้สามารถเขียนข้อมูลลงไฟล์อย่างเดียว

ทดสอบการทำงาน

เมื่อเปิดโปรแกรม Browser ขึ้นมาแล้วเรียก URL http://localhost/scilab_php/log2.html จากนั้นทดลองกรอกตัวเลข 128 เข้าไป เมื่อคลิกปุ่ม Submit แล้วข้อมูลจะถูกส่งไปยังไฟล์ log2.php จากนั้นก็จะสร้างไฟล์สคริปต์ชื่อ test.sce ดังรูปที่ 3.5 และเมื่อเปิดไฟล์ test.sce ด้วย NotePad จะได้ผลลัพธ์ดังรูปที่ 3.6



รูปที่ 3.5 แสดงไฟล์ test.sce



รูปที่ 3.6 แสดงการเปิดไฟล์ test.sce ด้วยโปรแกรม NotePad

2.2 เขียนโปรแกรมเพื่อรันไฟล์สคริปต์และบันทึกผลลัพธ์เป็นเท็กไฟล์

เมื่อเราทราบหลักการสร้างไฟล์สคริปต์แล้วขั้นตอนนี้จะเป็นการนำไฟล์สคริปต์ที่ได้ไปรันบนโปรแกรม SCILAB จากนั้นก็จะบันทึกผลลัพธ์เป็นเท็กไฟล์ ซึ่งในโปรแกรม SCILAB จะมีรูปแบบดังนี้

```
savematfile('filename', 'var1', 'var2', ...)
```

โดยที่

filename คือ ชื่อและนามสกุลของไฟล์

var1, *var2*… คือ ตัวแปรที่ต้องการบันทึก

ขั้นตอนการเขียนโปรแกรม

1. เปิดไฟล์ log2.php ขึ้นมาแก้ไข จากนั้นเพิ่มโค้ดดังนี้

```
<?php

$log2 = $_POST['log2'];                                  //ประกาศตัวแปร $log 2 รับค่า log 2 ที่ส่งมาจากฟอร์ม
$data = "log2($log2); \n savematfile('001.txt','ans'); \n exit;";   //สร้างตัวแปร $data เก็บข้อมูล log2($log2); \n savematfile('001.txt','ans'); \n exit;
                                                         //เพื่อทำการเขียนลงไฟล์ test.sce

$file = @fopen("test.sce", "w");                         //สร้างตัวแปร $file เก็บคำสั่งสร้าง และ เปิดไฟล์ไว้
  @fwrite($file, $data);                                 //สร้างไฟล์ test.sce แล้วเขียนไฟล์

$path=$_SERVER["SystemRoot"];                            //สร้างตัวแปร $path เก็บค่าไดเร็กเทอรี่ C:\WINDOWS
$com = "$path/scilab-4.1.2/bin/Scilex.exe -f test.sce";  //สร้างตัวแปร $com เก็บข้อความ คำสั่งรับโปรแกรมของ โปรแกรม Scilab เอาไว้

exec($com);                                              //สั่งประมวลผลไฟล์สคริปต์
?>
```

<u>รูปที่ 3.7</u> โค้ดการประมวลผลไฟล์สคริปต์

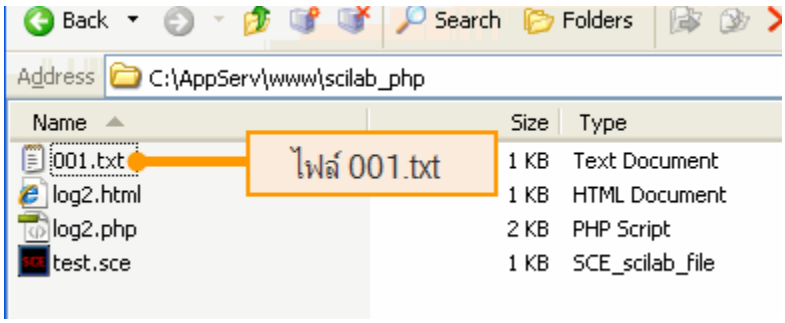
อธิบายโปรแกรม

หลังจากที่ได้สร้างไฟล์สคริปต์เรียนร้อยแล้วในหัวข้อที่ผ่านมาในหัวข้อนี้ก็จะเขียนโค้ดโปรแกรมต่อจากเดิมเพราะไฟล์ PHP ทั้งหมดจะอยู่ในไฟล์เดียวกัน

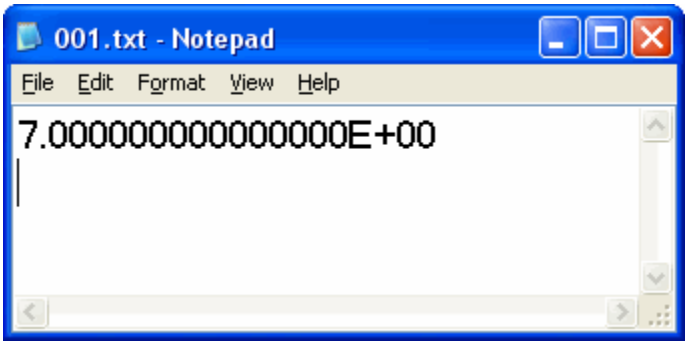
บรรทัดที่ 10 – 11 สร้างตัวแปร $path เก็บค่าที่อยู่ในตัวแปร PHP แบบ Predefinde Variables ซึ่งเป็นตัวแปรที่ PHP สร้างไว้ล่วงหน้าเพื่อให้ใช้งานได้สะดวก ซึ่งตัวแปร $_SERVER[“SystemRoot”] จะเก็บค่าไดเร็กเทอรีที่เก็บไฟล์ระบบของระบบปฏิบัติการ ในที่นี้คือ C:\Windows จากนั้นนำตัวแปร $path ไปกำหนดรวมกับคำสั่งที่ใช้ในการประมวลผลไฟล์สคริปต์เก็บไว้ในตัวแปร $com

ทดสอบโปรแกรม

ขั้นตอนการทดสอบให้ทำเหมือนการทดสอบก่อนหน้าคือ เปิดโปรแกรม Browser แล้วเรียกไฟล์ที่สร้างขึ้นผ่านทาง Address Bar ซึ่งในหัวข้อต่อๆไปก็ทำเช่นกัน หลังจากที่รันโปรแกรมจะก็จะสร้างไฟล์ 001.txt ขึ้นมา ดังรูปที่ 3.8 และเมื่อเปิดด้วยโปรแกรม NotePad จะได้ดังรูปที่ 3.9



รูปที่ 3.8 แสดงไฟล์ 001.txt



รูปที่ 3.9 แสดงการเปิดไฟล์ 001.txt

3. อ่านข้อมูลจากเท็กซ์ไฟล์นำมาแสดงผลทางหน้าเว็บเพจ

ขั้นตอนนี้จะอยู่ไฟล์เดียวกับขั้นตอนการสร้างไฟล์สคริปต์และประมวลผลไฟล์สคริปต์ เมื่อประมวลผลไฟล์สคริปต์แล้วผลลัพธ์ที่ได้นั้นจะอยู่ในรูปแบบของเท็กไฟล์ (นามสกุล .txt) ดังนั้นในขั้นตอนนี้เราจะเขียนคำสั่งเพื่อเปิดไฟล์ที่ได้แล้วทำการอ่านไฟล์ไปเรื่อยๆจนสิ้นสุดที่บรรทัดสุดท้าย จากนั้นนำผลลัพธ์มาแสดงทางหน้าจอ

ขั้นตอนการเขียนโปรแกรม

3.3 เปิดไฟล์ log2.php จากนั้นเพิ่มโค้ดต่อจากบรรทัดสุดท้ายของไฟล์เดิมดังนี้

```
14
15 $file = fopen("001.txt", "r");                    //สร้างตัวแปร $file เป็นคำสั่งในการอ่านไฟล์
16  while ($data = fgets($file, 1024)){              //อ่านข้อมูลแล้วเก็บไว้ในตัวแปร $data
17     echo "ผลลัพธ์ = ".number_format($data,7);     //แสดงผลลัพธ์ด้วยการแปลงข้อมูล เป็นทศนิยม 7 ไม่เกินตำแหน่ง
18  }
19  @fclose($file);                                  //คำสั่งปิดไฟล์
20  ?>
```

รูปที่ 3.10 แสดงโค้ดโปรแกรม

```php
<?php

$log2 = $_POST['log2'];                                  //ประกาศตัวแปร $log 2 รับค่า log 2 ที่ส่งมาจากฟอร์ม
$data = "log2($log2); \n savematfile('001.txt','ans'); \n exit;";   //สร้างตัวแปร $data เก็บข้อมูล log2($log2); \n savematfile('001.txt','ans'); \n exit;
                                                         //เพื่อทำการเขียนลงไฟล์ test.sce

$file = @fopen("test.sce", "w");                         //สร้างตัวแปร $file เก็บคำสั่งสร้าง และ เปิดไฟล์ไว้
 @fwrite($file, $data);                                  //สร้างไฟล์ test.sce แล้วเขียนไฟล์

$path=$_SERVER["SystemRoot"];                            //สร้างตัวแปร $path เก็บค่าไดเร็กเทอรี่ C:\WINDOWS
$com = "$path/scilab-4.1.2/bin/Scilex.exe -f test.sce";  //สร้างตัวแปร $com เก็บข้อความ คำสั่งรับโปรแกรมของ โปรแกรม Scilab เอาไว้

exec($com);                                              //สั่งประมวลผลไฟล์สคริปต์

$file = fopen("001.txt", "r");                           //สร้างตัวแปร $file เป็นคำสั่งในการอ่านไฟล์
 while ($data = fgets($file, 1024)){                     //อ่านข้อมูลแล้วเก็บไว้ในตัวแปร $data
    echo "ผลลัพธ์ = ".number_format($data,7);            //แสดงผลลัพธ์ด้วยการแปลงข้อมูล เป็นทศนิยม 7 ไม่เกินตำแหน่ง
 }
 @fclose($file);                                         //คำสั่งปิดไฟล์
?>
```

รูปที่ 3.11 แสดงไฟล์ log2.php ที่สมบูรณ์

อธิบายโปรแกรม

บรรทัดที่ 15 ทำการเปิดไฟล์ *001.txt* แบบอ่านอย่างเดียว จากนั้นเก็บไว้ในตัวแปร $file

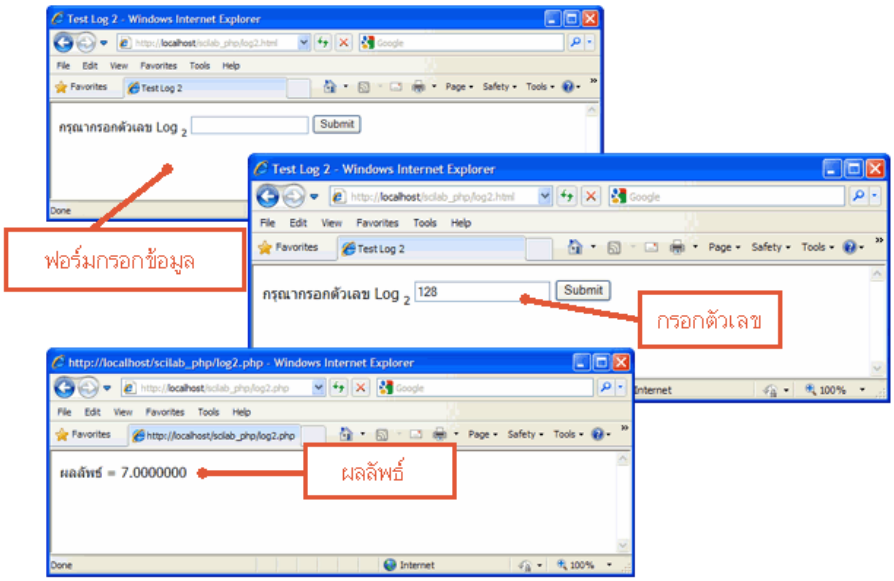
บรรทัดที่ 16–18 อ่านไฟล์ไปเรื่อยๆจนถึงจุดสิ้นสุดของไฟล์ ด้วยเงื่อนไข *while ($data = fgets($file,1024))* หมายความว่า ถ้าอ่านไฟล์สำเร็จด้วยฟังก์ชัน *fgets($file,1024)* จะได้ค่าเป็น *TRUE* ดังนั้นเงื่อนไขในคำสั่ง *while* ก็เป็นจริง จะทำแบบนี้ไปเรื่อยๆจนกว่าจะได้ค่าเป็น *FALSE* ก็จะออกจาก *Loop while*

บรรทัดที่ 19 คำสั่งที่ใช้ในการปิดไฟล์

การทดสอบการทำงาน

การทำงานของโปรแกรมจะรับค่าตัวเลขทางแป้นพิมพ์ผ่านฟอร์มส่งไปให้ไฟล์ PHP ทางฝั่งเซิร์ฟเวอร์สร้างไฟล์สริปต์แล้วอ่านไฟล์มาแสดงผล ดังนั้นในที่นี้ฟอร์มจะอยู่ที่ไฟล์ log2.html การเรียกใช้งานจึงเรียกไฟล์ log2.html ดังนี้

1. เปิดโปรแกรม Browser แล้วพิมพ์ URL ในช่อง Address ดังนี้ http://localhost/scilab_php/log2.html ดังรูปที่ 3.12



<u>รูปที่ 3.12</u> แสดงการทดสอบการทำงาน

Chapter 4

การประยุกต์ใช้งาน PHP เพื่อติดต่อกับ Scilab

ในบทนี้จะเป็นการประยุกต์องค์ความรู้ทั้งหมดในบทที่ผ่านมาเพื่อใช้ในการสร้างโปรแกรมเพื่อใช้ในการคำนวณทางคณิตศาสตร์ ฟิสิกส์ ไฟฟ้า และอื่นๆ ซึ่งฟังก์ชันต่างๆของโปรแกรม SCILAB สามารถศึกษาได้จากคู่มือภาษา SCILAB สำหรับผู้เริ่มต้น (ฉบับปรับปรุงใหม่) หรือสามารถดาวน์โหลดได้ที่เว็บไซต์ http://home.npru.ac.th/piya/webscilab ซึ่งจะมีตัวอย่างการใช้งาน การประยุกต์ใช้งานฟังก์ชัน การสร้างฟังก์ชันขึ้นมาใช้งาน รวมถึงเครื่องมือ (Tools) ที่ใช้ในการพัฒนา และสามารถดาวน์โหลดโปรแกรม SCILAB มาใช้งานได้ฟรีจากเว็บไซต์นี้ โดยผู้อ่านสามารถนำองค์ความรู้ที่ได้ไปต่อยอดพัฒนาสื่อการเรียนการสอนหรือนำไปใช้งานในทางคณิตศาสตร์ได้อีกมากมาย ซึ่งในบทนี้มีตัวอย่างการประยุกต์ใช้งานดังนี้

1. การหา det ของเมทริกซ์ขนาด 2x2
2. การหาคำตอบของสมการพหุนาม
3. การวาดกราฟของสมการพหุนาม

1. การหา det ของเมทริกซ์ขนาด 2x2

ขั้นตอนการเขียนโปรแกรม

1. สร้างฟอร์มสำหรับรับค่าเมทริกซ์ A ให้ออกแบบหน้าจอดังรูปที่ 4.1

<u>รูปที่ 4.1</u> หน้าจอการออกแบบ

โค้ดของฟอร์ม

```
<html>
<head>
<title> ทดสอบ Det(A)</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="">
  <table width="192" border="0" cellspacing="2" cellpadding="0">
    <tr>
      <td width="41" rowspan="2">A = </td>
      <td width="12" ><input name="d1" type="text" id="d1" size="1" /></td>
      <td width="77" valign="middle"><input name="d3" type="text" id="d3" size="1" /></td>
      <td width="58" rowspan="2" valign="middle"><input type="submit" name="button" id="button" value="Submit" /></td>
    </tr>
    <tr>
      <td><input name="d2" type="text" id="d2" size="1" /></td>
      <td><input name="d4" type="text" id="d4" size="1" /></td>
    </tr>
  </table>
</form>
</body>
</html>
```

อธิบายโปรแกรม

ในส่วนของฟอร์มนั้น จะเป็นภาษา HTML ซึ่งเป็นพื้นฐานของการเขียนเว็บไซต์ ในที่นี้จะไม่อธิบายอย่างละเอียดแต่จะอธิบายในส่วนที่สำคัญๆ เท่านั้น จากโค้ดจะได้หน้าเว็บเพจดังรูปที่ 4.1

2. เขียนโค้ดรับค่าจากฟอร์ม

   หลังจากเขียนหน้าฟอร์มไว้เรียบร้อยแล้วขั้นตอนต่อไปนี้จะเป็นการรับค่ามาเก็บไว้ในตัวแปรเพื่อเตียมพร้อมในการใช้งานดังนี้

```
<?php

$d1 = $_POST['d1'];
$d2 = $_POST['d2'];
$d3 = $_POST['d3'];
$d4 = $_POST['d4'];

```

สร้างตัวแปรเก็บค่าสมาชิกของเมทริกซ์ แต่ละตัว เป็น $d1 ,$d2 $d3, $d4

3. สร้างไฟล์สคริปต์

   ขั้นตอนนี้จะนำค่าในตัวแปรมาเขียนลงไฟล์สคริปต์ โดยจะสร้างไฟล์สคริปต์ชื่อ matrix.sce ดังนี้

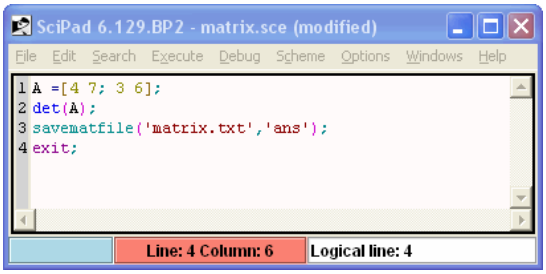
```
$data = "A =[$d1 $d2; $d3 $d4]; \n det(A); \n savematfile('matrix.txt','ans'); \n exit;";


$file = @fopen("matrix.sce", "w");
  @fwrite($file, $data);

```

บรรทัดที่ 8 สร้างตัวแปร $data เก็บข้อมูลที่จะเขียนลงไฟล์

บรรทัดที่ 11-12 สร้างและเปิดไฟล์ชื่อ matrix.sce เก็บไว้ในตัวแปรไฟล์ แล้วเขียนลงไฟล์ด้วยข้อมูลที่อยู่ในตัวแปร $data เมื่อรันโปรแกรมจะพบว่ามีโปรแกรมจะสร้างไฟล์ matrix.sce ขึ้นมา และเมื่อเปิดดูจะได้ดังรูปที่ 4.2



<u>รูปที่ 4.2</u> แสดงการเปิดไฟล์ matrix.sce

4. ประมวลผลไฟล์สคริปต์ด้วยโปรแกรม SCILAB

ขั้นตอนนี้เราจะเขียนโปรแกรม PHP เพื่อรันโปรแกรมภายนอก รูปแบบคือ exec(*ชื่อไฟล์หรือไดเร็กเทอรี*) เพิ่มโค้ดโปรแกรมดังนี้

```
14  $path=$_SERVER["SystemRoot"];
15  $com = "$path/scilab-4.1.2/bin/Scilex.exe -f matrix.sce";
16
17  exec($com);
```

บรรทัดที่ 14 สร้างตัวแปร $path เก็บข้อมูลไดเร็กเทอรี่ที่เก็บไฟล์ระบบปฏิบัติการ

บรรทัดที่ 15 สร้างตัวแปร $com เก็บคำสั่งที่ใช้ในการประมวลผลไฟล์สคริปต์ โดยอ้างถึงไฟล์ Scilex.exe ซึ่งเป็นไฟล์ที่ใช้ในการรันโปรแกรม SCILAB เพื่อประมวลผลไฟล์ matrix.sce

บรรทัดที่ 17 เป็นคำสั่งที่ใช้เรียกใช้งานโปรแกรมภายนอก ในที่นี้จะเรียกโปรแกรม SCILAB ขึ้นมา

5. นำผลลัพธ์ที่ได้จากการประมวลผลไฟล์สคริปต์ไปแสดงผลยังเว็บบราวเซอร์

หลังจากที่ประมวลผลไฟล์สคริปต์แล้วนั้นจะได้เท็กซ์ไฟล์ผลลัพธ์ขึ้นมา ขั้นตอนนี้เราจะใช้ PHP ในการเปิดไฟล์เพื่ออ่านผลลัพธ์และส่งไปแสดงยังเว็บบราวเซอร์ ให้เขียนโค้ดเพิ่มดังนี้

```
18
19  $file = fopen("matrix.txt", "r");
20   while ($data = fgets($file, 1024)){
21      echo "ผลลัพธ์ = ".number_format($data,7);
22   }
23   @fclose($file);
24  ?>
```

บรรทัดที่ 19 เปิดไฟล์ matrix.txt แบบอ่านอย่างเดียว แล้วเก็บไว้ในตัวแปรไฟล์

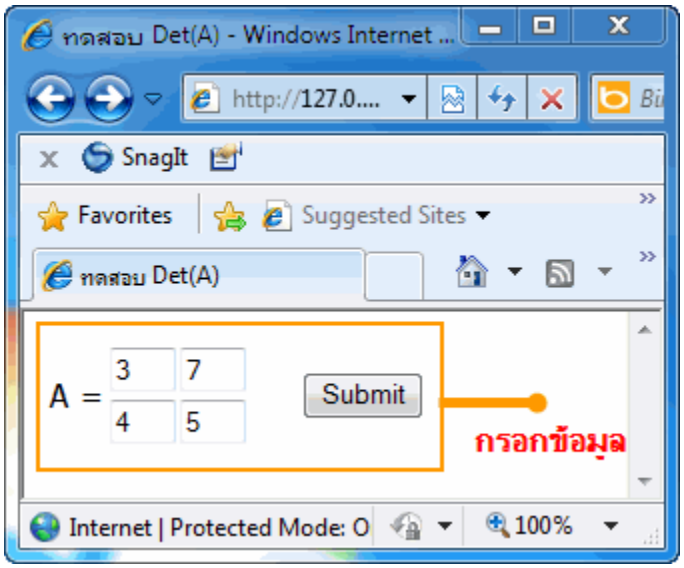
บรรทัดที่ 20 ทำการอ่านไฟล์ไปเรื่อยๆ จนกว่าเงื่อนไข while จะมีค่าเป็นเท็จ (Flase) จะได้ค่า Flase ก็ต่อเมื่ออ่านไฟล์ไปจนถึงจุดสิ้นสุดของไฟล์ (End Of File)

บรรทัดที่ 21 แสดงผลลัพธ์ที่ได้จากการอ่านไฟล์ โดยกำหนดรูปแบบการแสดงผลข้อมูลเป็นทศนิยมไม่เกิน 7 ตำแหน่ง

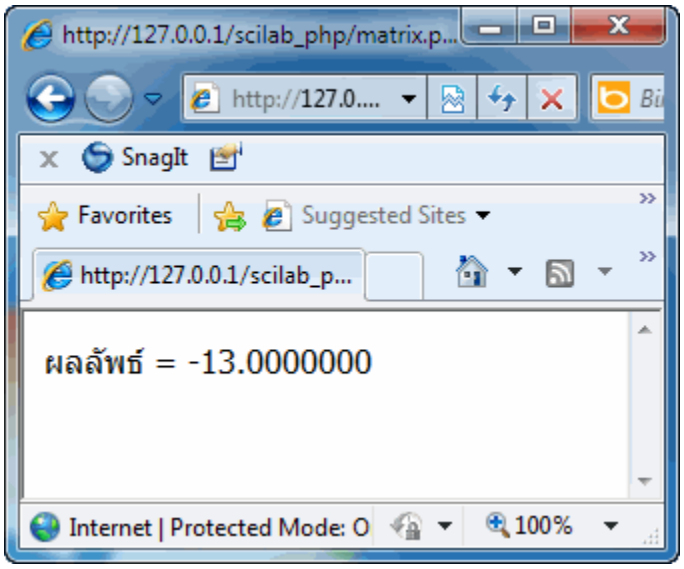
เสร็จจากขั้นตอนนี้แล้วจะได้ไฟล์ PHP ที่สมบูรณ์เพื่อทำงานในฝั่งเซิฟเวอร์ สรุปไฟล์ matrix.php ที่สมบูรณ์จะได้ดังนี้

```
1   <?php
2
3   $d1 = $_POST['d1'];
4   $d2 = $_POST['d2'];
5   $d3 = $_POST['d3'];
6   $d4 = $_POST['d4'];
7
8   $data = "A =[$d1 $d2; $d3 $d4]; \n det(A); \n savematfile('matrix.txt','ans'); \n exit;";
9
10
11  $file = @fopen("matrix.sce", "w");
12   @fwrite($file, $data);
13
14  $path=$_SERVER["SystemRoot"];
15  $com = "$path/scilab-4.1.2/bin/Scilex.exe -f matrix.sce";
16
17  exec($com);
18
19  $file = fopen("matrix.txt", "r");
20   while ($data = fgets($file, 1024)){
21      echo "ผลลัพธ์ = ".number_format($data,7);
22   }
23   @fclose($file);
24  ?>
```

เมื่อได้ไฟล์ที่สมบูรณ์แล้ว ให้ทดสอบการทำงานเมื่อรันโปรแกรมแล้วจะได้ดังรูปที่ 4.3⇨ กรอกตัวเลขสมาชิกของเมทริกซ์ A ลงไปในฟอร์ม ⇨ คลิกปุ่ม Submit จะได้ผลลัพธ์ดังรูปที่ 4.4



<u>รูปที่ 4.3</u> ฟอร์มกรอกข้อมูล



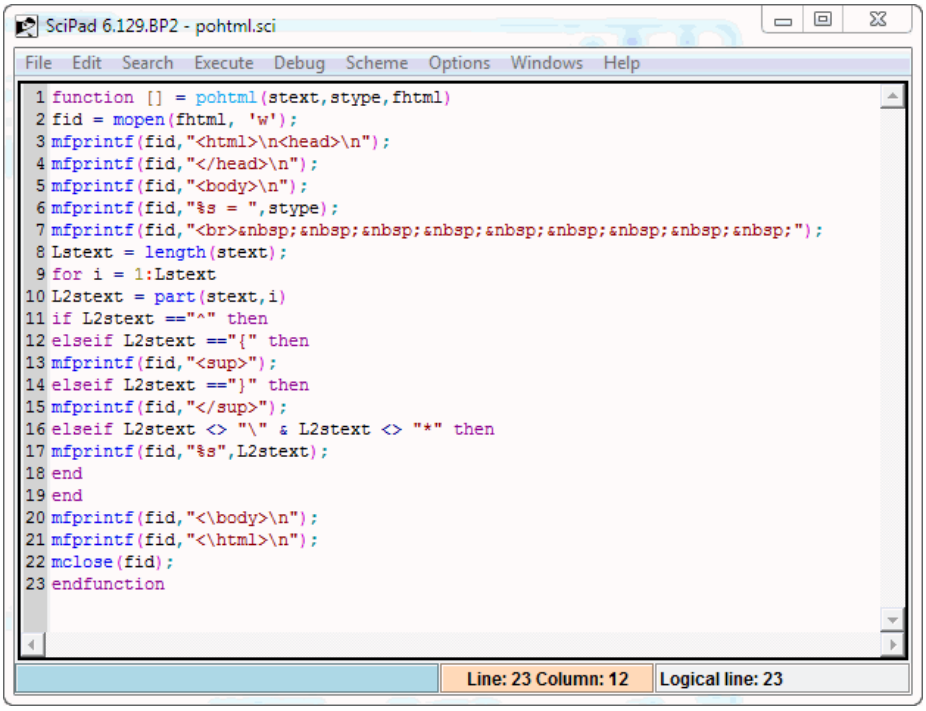
<u>รูปที่ 4.4</u> ผลลัพธ์ที่ได้จากการคำนวณ

2. การหาคำตอบของสมการพหุนาม

ในตัวอย่างนี้จะเป็นการหาคำตอบของสมาการพหุนาม เมื่อ y มีค่าเท่ากับ 0 เช่น $2x^3+5x^2-2=0$ สำหรับคำตอบที่ได้นั้นจะมีคำตอบเท่ากับเลขชี้กำลังตัวที่มากที่สุดของสมการพหุนาม ซึ่งในตัวอย่างนี้จะให้แสดงสมาการที่เรากรอกลงไปในรูปแบบที่สามารถเข้าใจได้ง่าย

ตัวอย่างเช่น $2x^3+5x^2-2$

จากสมาการตัวอย่าง ในโปรแกรม SCILAB จะไม่สามารถบันทึกข้อมูลให้อยู่ในรูปแบบนี้ได้ แต่จะบันทึกผลลัพธ์เป็นเท็กซ์ไฟล์ เช่น -2+5*x+2*x^{3} ดังนั้นเราจึงต้องสร้างฟังก์ชัน เพื่อที่จะแปลงค่าที่ได้เป็นรูปแบบที่เราอ่านได้ง่าย ดังนี้



<u>รูปที่ 4.5</u> แสดงฟังก์ชัน pohtml ที่เราสร้างขึ้น

ฟังก์ชันที่สร้างขึ้นมานั้นจะบันทึกไฟล์เป็นนามสกุล .sci และเรียกใช้งานฟังก์ชันด้วยคำสั่ง exec

ขั้นตอนการเขียนโปรแกรมมีดังนี้

1. ออกแบบหน้าจอและสร้างฟอร์มรับค่าสมการพหุนามดังรูปที่ 4.6

   กรุณากรอสมการ y = [ ] Submit

2. ให้ Save ไปไว้ที่ C:\AppServ\www\scilab_php ตั้งชื่อไฟล์เป็น poly.hmtl
3. เขียนโปรแกรม PHP เพื่อจัดการกับข้อมูลที่รับมาจากฟอร์มซึ่งจะแบ่ง โค้ดของโปรแกรมออกเป็นส่วนๆ คือ
   - ส่วนการรับค่าจากฟอร์ม
   - ส่วนของการสร้างไฟล์สคริปต์ และรันไฟล์สคริปต์
   - ส่วนการแสดงผลลัพธ์

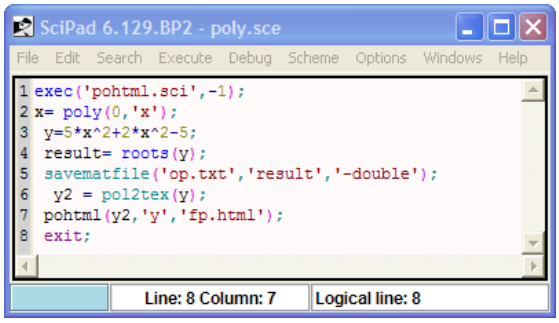
   จากการแบ่งโค้ดโปรแกรมเป็นส่วนๆสามารถเขียนโปรแกรมได้ดังนี้

เขียนโปรแกรมเพื่อรับค่าจากฟอร์ม

```
1  <?
2  $y = $_POST['poly'];
3
4  $data = "exec('pohtml.sci',-1); \nx= poly(0,'x'); \n y=$y;\n result= roots(y);\n savematfile('op.txt','result','-double');\n
   y2 = pol2tex(y);\n pohtml(y2,'y','fp.html');\n exit;";
```

บรรทัดที่ 2 รับค่าด้วยเมธอด POST จากฟอร์มชื่อ poly เก็บไว้ในตัวแปร $y

บรรทัดที่ 4 เขียนคำสั่งที่ต้องการสร้างไฟล์สคริปต์เก็บไว้ในตัวแปร $data จากคำสั่งดังกล่าวจะได้ไฟล์สริปต์ที่มีรูปแบบคำสั่งทั้งหมดดังรูปที่ 4.6



<u>รูปที่ 4.6</u> แสดงไฟล์สคิปต์ชื่อ poly.sce ที่จะใช้ในการคำนวณ

เขียนโปรแกรมสร้างไฟล์สคริปต์และรันไฟล์สคริปต์

```
5
6  $file = @fopen("poly.sce", "w");
7    @fwrite($file, $data);
8
9  $path=$_SERVER["SystemRoot"];
10 $com = "$path/scilab-4.1.2/bin/Scilex.exe -f poly.sce";
11
12 exec($com);
```

บรรทัดที่ 6 สร้างไฟล์สริปต์ชื่อ poly.sce แล้วเปิดไฟล์ จากนั้นเก็บไว้ในตัวแปร $file

บรรทัดที่ 7 นำค่าที่เก็บไว้ในตัวแปร $data มาเขียนลงไฟล์ poly.sce

บรรทัดที่ 9-10 สร้างตัวแปรเพื่อเก็บคำสั่งที่ใช้ในการรันไฟล์สคริปต์

บรรทัดที่ 12 เป็นคำสั่งที่ใช้ในการประมวลผลไฟล์ poly.sce

เขียนโปรแกรมเพื่อแสดงผลลัพธ์ทางหน้าจอ

```
13
14 $file = fopen("fp.html","r");
15   while ($data_y = fgets($file, 1024)){
16               list($exp1, $exp2)= explode("= ", trim($data_y), 2);
17              $result[]=$exp2;
18   }
19 echo "สมาการที่รับมา คือ y =  $result[4]<br>";
20 echo "**********<br>";
21
22 echo "<font color='red' size='3'>ค่าของ x </font><br>";
23 $file1 = fopen("op.txt", "r");
24 $i=1;
25   while ($data = fgets($file1, 1024)){
26
27      echo "ค่าที่ $i = ".number_format($data,7)."<br>";
28      $i++;
29   }
30   @fclose($file);
31 ?>
```

บรรทัดที่ 14 เปิดไฟล์ fp.html ซึ่งเป็นไฟล์ผลลัพธ์ที่ได้จากการรันไฟล์สริปต์เก็บไว้ในตัวแปร $file

บรรทัดที่ 15-18 อ่านไฟล์ด้วยฟังก์ชัน fgets() แล้วเก็บไว้ในตัวแปร $data_y จากนั้นใช้ฟังก์ชัน trim() ในการตัดช่องว่างออกไป เมื่อตัดช่องว่าออกแล้วก็ตัดข้อความอีกครั้งด้วยฟังก์ชัน explode() จากนั้นเก็บไว้ในตัวแปร $exp1 และ $exp2 จำค่าที่เก็บไว้ในตัวแปร $exp2 ไปเก็บไว้ในตัวแปร $result[] ซึ่งเป็นตัวแปรชนิด Array

บรรทัดที่ 19-20 ให้แสดงข้อความ “สมาการที่รับมา คือ y =” ตามด้วยข้อมูลที่เก็บอยู่ในตัวแปร $result[4] ลำดับที่ 4 ของตัวแปร Array แล้วขึ้นบรรทัดใหม่ด้วย แท็ก
 จากให้ให้แสดง “ ******”

บรรทัดที่ 22 แสดงข้อความ “ค่าของ x”

บรรทัดที่ 23 เปิดโปแกรมด้วยฟังก์ชัน fopen() เก็บไว้ในตัวแปร $file1

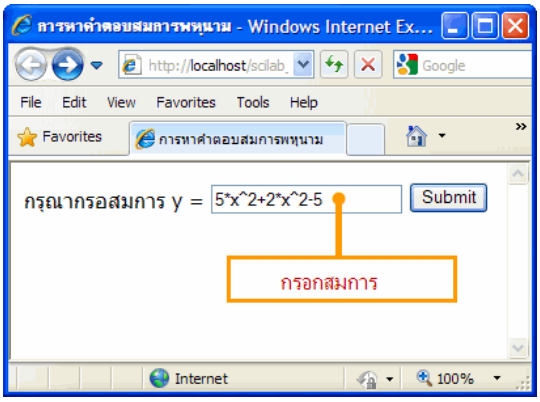
บรรทัดที่ 24-29 ให้ตัวแปร $i มีค่าเริ่มต้นเป็น 1 จากนั้นอ่านไฟล์ด้วยฟังก์ชัน fgets() จากนั้นนำค่าที่ได้ไปแสดงผลทางหน้าจอในบรรทัดที่ 27 โดยมีรูปแบบคือ “ค่าที่ 1=,ค่าที่ 2 =,…..ไปเรื่อยๆจนถึงจุดสิ้นสุดของไฟล์”

บรรทัดที่ 30 ปิดไฟล์ด้วยฟังก์ชัน fclose()

เมื่อสร้างไฟล์ poly.php เสร็จแล้วให้บันทึกไฟล์ไปที่ C:\AppServ\www\scilab_php จากนั้นทำการทดสอบโดยเรียกไฟล์ poly.html ขึ้นมาเพื่อกรอกสมาการ

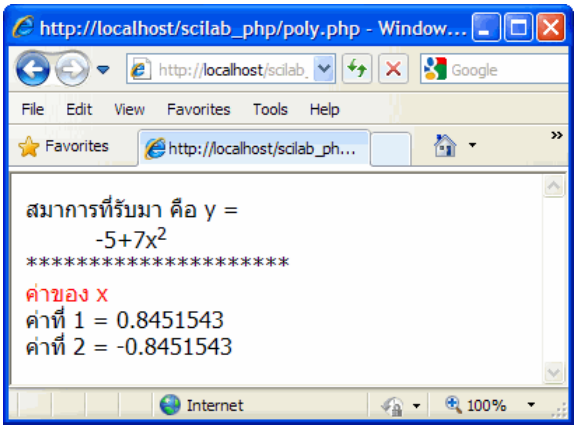
ทดสอบโปรแกรม

กรอกสมการ $5x^2 + 2x^2 - 5$ ซึ่งในโปรแกรม SCILAB จะมีรูปแบบการกรอกข้อมูลเป็น 5*x^2+2*x^2-5



<u>รูปที่ 4.7</u> แสดงการกรอกสมการ

เมื่อคลิกปุ่ม Submit จะได้ผลลัพธ์ดังรูปที่ 4.8



<u>รูปที่ 4.8</u> แสดงผลลัพธ์ที่ได้จากการอ่านไฟล์

3. การวาดกราฟของสมการพหุนาม

ในตัวอย่างนี้นั้นเป็นการวาดกราฟของสมการพหุนามดังนั้นผลลัพธ์ที่ได้นั้นจะแตกต่างจากตัวอย่างที่ผ่านมา เนื่องจากจะได้ผลลัพธ์เป็นไฟล์ภาพ (.gif) โดยเราจะวาดกราฟด้วยฟังก์ชัน plot(x,y) และบันทึกผลลัพธ์ด้วยฟังก์ชัน xs2gif()

ในการพัฒนาเว็บแอปพลิเคชันนั้นปัจจุบันนิยมใช้เทคนิค Ajax เข้ามาช่วยในการทำงานระหว่างเครื่องไคล์เอนกับเซริฟเวอร์ เพื่อให้การตรวจสอบข้อผิดพลาดนั้นทำได้ง่าย ดังนั้นในตัวอย่างนี้เราจะมาลองใช้ Ajax ในการทำงานร่วมกับ PHP และ SCILAB ซึ่งการพัฒนานั้น เราจะสร้างฟอร์มซึ่งจะอยู่ในเพจเดียวกันกับเพจที่ใช้ในการแสดงผล โดยจะใช้ Ajax เข้ามาช่วยในการส่งข้อมูลจากฟอร์มแล้วนำกลับมาแสดงผล ซึ่งมีขั้นตอนดังนี้

1. ออกแบบหน้าจอเพื่อรับและแสดงผลกราฟดังรูปที่ 4.9 จากนั้น Save ชื่อ gra.html

<u>รูปที่ 4.9</u> แสดงฟอร์มกรอกข้อมูลเพื่อวาดกราฟ

โค้ดของการออกแบบหน้าจอ

```
<html>
<head>
<title>ทดสอบ SCILAB BY Ajax & PHP</title>
<style type="text/css">
.txt20b {
		font-family: sans-serif;
		font-size: 24px;
		font-weight: bold;
		color: #FF0000;
}
</style>
</head>
<body ><center>
<table width="610" border="0" cellpadding="0" cellspacing="0" bgcolor="#FFFFCC">
 <tr>
 <td align="center" class="txt20b">Test SCILAB BY Ajax &amp;amp; PHP </td>
 </tr>
 <tr>
 <td><table width="610" border="0" cellpadding="0" cellspacing="2" bgcolor="#66FF99">
 <tr>
     <td align="center">
ค่าเริ่มต้น <input name="x1" type="text" id="x1" size="1">
ค่าสะสม <input name="x2" type="text" id="x2" size="1">
ค่าสิ้นสุด <input name="x3" type="text" id="x3" size="1">
</td>
  </tr>
    <tr>
     <td align="center">สมการ&amp;nbsp;&amp;nbsp;&amp;nbsp;
 <input type="text" name="poly" id="poly"/>&amp;nbsp;
       <button id="btnGrap">คำนวณ</button> &amp;nbsp;<button id="Clear">เคลียร์ค่า</button></td>
     </tr>
    <tr>
     <td align="center" class="txt20b">รูปกราฟ</td>
    </tr>
    <tr>
     <td><div id="result" align="center"></div>
       </td>
     </tr>
  </table></td>
```

```
 </tr>
 <tr>
  <td></td>
 </tr>
</table>
</center>
</body>
</html>
```

2. เขียนโปรแกรมด้วย Ajax แทรกลงในเพจ gra.html โดยแทรกโค้ดต่อไปนี้ในส่วนของแท็ก <head></head>

```
12  <script>
13  var ajax = null;
14  if(window.ActiveXObject){
15     ajax = new ActiveXObject("Microsoft.XMLHTTP");
16  }
17  else if(window.XMLHttpRequest){
18     ajax = new XMLHttpRequest();
19  }
20  function ajaxLoad(method, URL, data, displayId){
21     ajax.open(method, URL);
22     ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
23
24     ajax.onreadystatechange = function(){
25        if(ajax.readyState == 4 && ajax.status == 200){
26           ajaxCallback(displayId, ajax.responseText);
27        }else{
28           document.getElementById("result").innerHTML="<img src='indicator.gif'><font size='2' color='red'>  กรุณารอสักครู่...</font>";
29           }
30     }
31     ajax.send(data);
32  }
33
```

บรรทัดที่ 12 เปิดแท็กสคริปต์

บรรทัดที่ 13 ประกาศตัวแปร ajax ให้เป็นค่าว่าง

บรรทัดที่ 14-19 สร้าง Object HttpRequest แล้วเก็บไว้ในตัวแปร ajax

บรรทัดที่ 20 สร้างฟังก์ชัน ajaxLoad() โดยรับค่าพารามิเตอร์ 4 ตัว คือ

- method ใช้รับค่าเมธอดที่กำหนด โดยจะกำหนดเป็น POST หรือ GET
- URL ใช้รับค่าชื่อไฟล์ที่ต้องการส่งค่าไปให้
- data เก็บข้อมูลที่จะส่งไปยังไฟล์ PHP

- displayId เก็บชื่อ Oject หรือ Attribute ที่ต้องการให้แสดงผล

บรรทัดที่ 21 เรียกใช้งานเมธอด open และส่งพาราเมเตอร์ method และ url เข้าไป

บรรทัดที่ 22 เรียกใช้งานเมธอด setRequestHeader แล้วใส่ header เข้าไป

บรรทัดที่ 25-26 เรียกใช้งานเมธอด onreadystatechange เพื่อตรวจจับ Request ว่าสิ่งที่ส่งไปนั้นถูกตอบกลับมาหรือยัง โดยตรวจสอบด้วย if (ajax.readyState == 4 && ajax.status == 200) ถ้าเป็นจริงแสดงว่าข้อมูลถูกส่งกลับมาแล้วก็จะเรียกใช้งานฟังก์ชัน ajaxCallback() ซึ่งจะอธิบายในลำดับต่อไป

บรรทัดที่ 27-30 ถ้าเงื่อนไขเป็นเท็จหรือยังไม่มีการตอบกลับมาจะแสดงรูปนาฬิกาทรายและข้อความ "กรุณารอสักครู่..."

บรรทัดที่ 31 เรียกใช้งานเมธอด send แล้วส่งพารามิเตอร์ ด้วยตัวแปร data

```
33
34 function ajaxCallback(displayId, responseText){
35    var el = document.getElementById(displayId);
36    el.innerHTML = responseText;
37 }
38
39 function sendData(){
40    var URL = "gra.php";
41   URL += "?rand=" + Math.random();
42   var x1 =document.getElementById("x1").value
43   var x2 =document.getElementById("x2").value
44   var x3 =document.getElementById("x3").value
45    var poly= document.getElementById("poly").value;
46    var data = "&x1="+x1+"&x2="+x2+"&x3="+x3+"&poly=" + poly ;
47    ajaxLoad('post', URL, data, 'result');
```

บรรทัดที่ 34 สร้างฟังก์ชัน ajaxCallback() เพื่อกำหนดค่าให้กับอิลิเมนต์ ซึ่งชื่ออิลิเมนต์นั้นถูกเก็บไว้ในตัวแปร displayId จากนั้นกำหนดค่าให้กับอิลิเมนต์ ด้วยเมธอด responseText

บรรทัดที่ 39 สร้างฟังก์ชัน sendData()

บรรทัดที่ 40 ประกาศตัวแปร URL เก็บชื่อและที่อยู่ของไฟล์ gra.php

บรรทัดที่ 41 กำหนดค่าที่ได้จากฟังก์ชัน Math.random() ให้กับ URL

บรรทัดที่ 42 รับค่าจากอิลิเมนต์ชื่อ x1 จากนั้นกำหนดให้กับตัวแปร x1

บรรทัดที่ 43 รับค่าจากอิลิเมนต์ชื่อ x2 จากนั้นกำหนดให้กับตัวแปร x2

บรรทัดที่ 44 รับค่าจากอิลิเมนต์ชื่อ x3 จากนั้นกำหนดให้กับตัวแปร x3

บรรทัดที่ 45 รับค่าจากอิลิเมนต์ชื่อ poly จากนั้นกำหนดให้กับตัวแปร poly

บรรทัดที่ 46 นำค่าจากตัวแปรแต่ละตัวที่รับมาจากอิลิเมนต์เก็บไว้ในตัวแปร data

บรรทัดที่ 47 เรียกใช้ฟังก์ชัน ajaxLoad() แล้วกำหนดพารามิเตอร์

3. เพิ่มอีเวนต์ (Event) เพื่อกำหนดเหตุการณ์ต่างๆให้กับอิลิเมนต์โดยกำหนดเหตุการณ์เมื่อผู้ใช้คลิกปุ่ม “คำนวณ” กำหนดดังนี้

```
<button id="btnGrap" onClick="sendData()">คำนวณ</button>
```

4. สร้างไฟล์ PHP เพื่อประมวลผลข้อมูลที่ถูกส่งมาโดย Ajax และควบคุมการทำงานของโปรแกรม SCILAB ดังนี้

```
1  <?php
2  $x1 = $_POST["x1"];
3  $x2= $_POST["x2"];
4  $x3 = $_POST["x3"];
5  $plot = $_POST["poly"];
6
7  $plot1=str_replace(" ", "+",$plot);
8
9  $data = "x = $x1:$x2:$x3; \n y = $plot1; \n plot(x,y); \n xs2gif(0,'test.gif'); \n exit;";
10
11 $file = @fopen("gar.sce", "w");
12   @fwrite($file, $data);
13
14 $path=$_SERVER["SystemRoot"];
15 $com = "$path/scilab-4.1.2/bin/Scilex.exe -f gar.sce";
16
17 shell_exec($com);
18
19      echo "<img src='test.gif'>";
20   @fclose($file);
21 ?>
```

บรรทัดที่ 2-5 รับข้อมูลที่ส่งมาผ่าน URL เก็บไว้ในตัวแปร $x1,$x2,$x3,$plot ตามลำดับ

บรรทัดที่ 7 ตรวจสอบค่าที่เก็บไว้ในตัวแปร $plot ถ้ามีช่องว่าให้แทนด้วยเครื่องหมาย “+” (ค่าที่ส่งมานั้นเป็นสมการ อาจมีเครื่องหมายบวกซึ่งเมื่อส่งผ่าน URL แล้วนั้นเครื่องหมายบวกจะถือว่าเป็นช่องว่า 1 ช่อง)

บรรทัดที่ 9 สร้างคำสั่งที่ใช้ในการสร้างไฟล์สคริปต์เก็บไว้ในตัวแปร $data

บรรทัดที่ 11-12 สร้างไฟล์สคริปต์ gar.sce จากนั้นเปิดไฟล์เก็บไว้ในตัวแปร $file แล้วนำตัวแปร $file ไปเขียนด้วยข้อมูลที่อยู่ในตัวแปร $data ด้วยฟังก์ชัน fwrite()

บรรทัดที่ 14-15 สร้างคำสั่งเพื่อใช้ในการรันไฟล์สคริปต์ เก็บไว้ในตัวแปร $com

บรรทัดที่ 17 เป็นคำสั่งที่ใช้ในการรันไฟล์สคริปต์

บรรทัดที่ 19-20 แสดงผลลัพธ์ไฟล์รูปภาพ test.gif จากนั้นปิดไฟล์ด้วยฟังก์ชัน fcolse()

5. สร้างฟังก์ชัน Clear() เพื่อเคลียร์ข้อมูลในเพจ gra.html

ขั้นตอนนี้เป็นการสร้างฟังก์ชันเพื่อควบคุมการกรอกข้อมูลเมื่อกรอกข้อมูลไปแล้วนั้นข้อมูลจะค้างอยู่ที่ฟอร์มกรอกข้อมูล (Textfield) เพื่อความสะดวกในการกรอกข้อมูลครั้งต่อไป ให้คลิกปุ่ม Clear แล้วข้อมูลจะถูกล้างออกไป เราจะสร้างฟังก์ชันไว้ในส่วน head โค้ดมีดังนี้

```
50  function Clear(){
51     document.getElementById("x1").value="";
52     document.getElementById("x2").value="";
53     document.getElementById("x3").value="";
54     document.getElementById("poly").value="";
55     document.getElementById("result").innerHTML="<font size='2' color='red'>***ยังไม่มีข้อมูล***</font>";
56  }
```

บรรทัดที่ 50 สร้างฟังก์ชันชื่อ Clear

บรรทัดที่ 51-54 กำหนดให้อิลิเมนต์ x1,x2,x3 และ poly เป็นค่าว่าง

บรรทัดที่ 55 กำหนดให้อิลิเมนต์ result แสดงคำว่า “ยังไม่มีข้อมูล” เป็นตัวอักษรสีแดง ขนาด 2 px

6. เรียกใช้งานฟังก์ชัน Clear() ด้วยการเพิ่ม Even เมื่อมีการคลิกปุ่ม Clear ดังนี้

```
<button id="Clear" onClick="Clear()">เคลียร์ค่า</button>
```

7. สร้างฟังก์ชันตรวจสอบข้อผิดพลาดในการกรอกข้อมูล

ขั้นตอนนี้จะช่วยให้การทำงานของโปรแกรมทำได้โดยไม่เกิด ERROR เพราะเราจะตรวจสอบความถูกต้องของการกรอกข้อมูลก่อนส่งในทางฝั่งเซิร์ฟเวอร์โดยการกำหนดอีเวนต์ onkeyup ให้กับอิลิเมนต์เมื่อผู้ใช้กรอกข้อมูลลงไปในฟอร์มรับข้อมูล จะทำให้ฟังก์ชันที่เราสร้างขึ้นมาทำงาน โค้ดของฟังก์ชันมีดังนี้

```
58  function IsNumeric(sText,obj)
59  {
60     var ValidChars = "0123456789.";
61     var IsNumber=true;
62     var Char;
63     for (i = 0; i < sText.length && IsNumber == true; i++)
64       {
65       Char = sText.charAt(i);
66       if (ValidChars.indexOf(Char) == -1)
67         {
68         IsNumber = false;
69         }
70       }
71        if(IsNumber==false){
72           alert("กรอกเฉพาะตัวเลขเท่านั้น");
73           obj.value=sText.substr(0,sText.length-1);
74        }
75   }
```

อธิบายโปรแกรม

บรรทัดที่ 58 เป็นการสร้างฟังก์ชัน IsNumeric() โดยรับค่าพารามิเตอร์เข้ามา 2 ตัว คือ ตัวแปร sText และตัวแปร obj

- sText ใช้เก็บค่าของข้อมูลที่ส่งมาจากอิลิเมนต์
- obj ใช้เก็บชื่ออิลิเมนต์

บรรทัดที่ 60 ประกาศตัวแปร ValidChars เก็บข้อมูล 0-9 และ จุด(.)

บรรทัดที่ 61-62 ปรกาศตัวแปร IsNumber ให้เป็นค่า True และประกาศตัวแปร Char

บรรทัดที่ 63 ให้วนลูปตั้งแต่ตัวแปร i มีค่าเป็น 0 จนถึง ค่าที่ได้จากการนับตัวอักษรในอิลิเมนต์ และตัวแปร IsNumber มีค่าเป็นจริง (True)

บรรทัดที่ 65 ดึงตัวอักษรลำดับที่ i เก็บไว้ในตัวแปร Char

บรรทัดที่ 66-69 ตรวจสอบตัวแปร Char ว่าเท่ากับ -1 หรือไม่ ถ้าใช่ ถ้าใช้ก็กำหนดค่าตัวแปร IsNumber เท่ากับ False

บรรทัดที่ 70 จบลูป For

บรรทัดที่ 71 -74 ตรวจสอบตัวแปร IsNumber ว่าเท่ากับ False หรือไม่ ถ้าใช่ให้แสดงข้อความ “กรอกเฉพราะตัวเลขเท่านั้น” จากนั้นตัดตัวอักษรที่อยู่ในอิลิเมนต์นั้นทิ้งไปหนึ่งตำแหน่ง

บรรทัดที่ 75 จบฟังก์ชัน

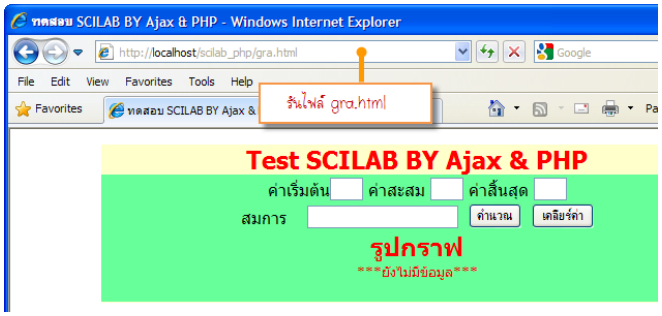
8. การเรียนใช้งานฟังก์ชัน IsNumeric() ทำได้โดยการกำหนด Even ให้กับอิลิเมนต์ TextFiled ที่ต้องการตรวจสอบในที่นี้เราจะตรวจสอบอิลิเมนต์ที่ชื่อ x1,x2,x3 ดังนี้

```
<input name="x1" type="text" id="x1" size="1" onKeyUp="IsNumeric(this.value,this)">
```

ทดสอบการทำงานของโปรแกรม

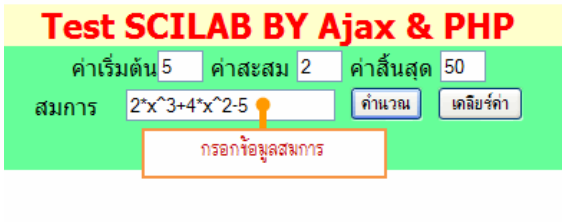
การทดสอบการทำงานของโปรแกรมทำดังนี้

1. เปิดโปรกแกรม Browser ในช่อง URL พิมพ์ http://localhost/scilab_php/gra.html



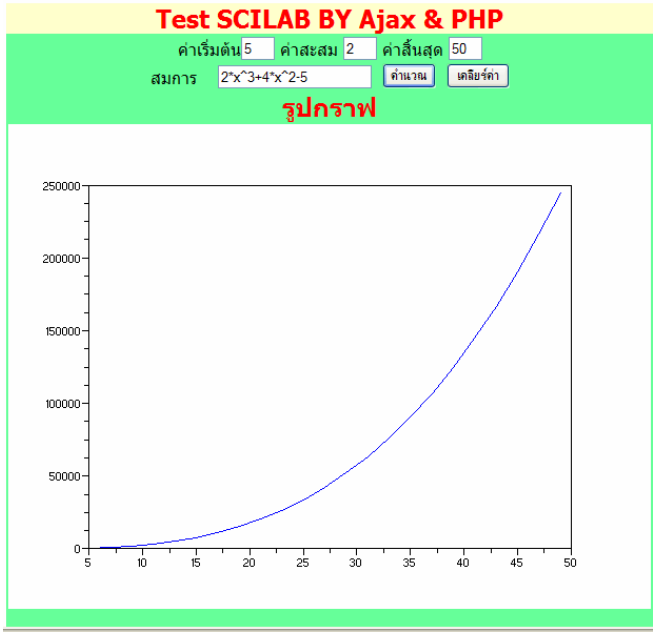
รูปที่ 4.10 แสดงการรันไฟล์ gra.html

2. กรอกข้อมูลค่าเริ่มต้น = 5 ค่าสะสม = 2 ค่าสิ้นสุด = 50 และสมการ $2x^3 + 4x^2 - 5$ ซึ่งในการกรอกสมการนั้นเราจะกรอกในตามรูปแบบของโปรแกรม SCILAB คือ $2x^3 + 4x^2 - 5$ จะได้ 2*x^3+4*x^2-5



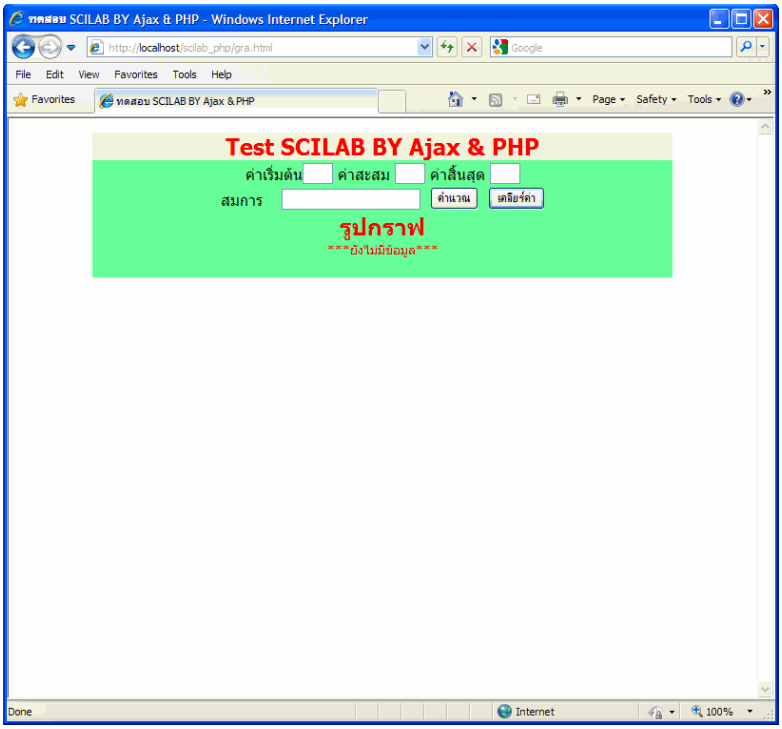
รูปที่ 4.11 แสดงการกรอกข้อมูลสมการ

3. เมื่อคลิกปุ่ม คำนวณ จะได้กราฟดังรูปที่ 4.12



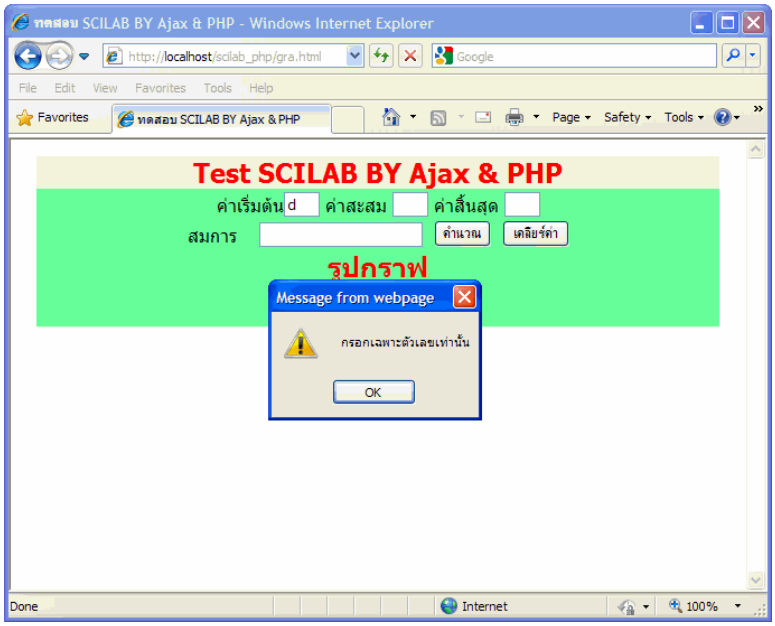
<u>รูปที่ 4.12</u> แสดงกราฟของสมการ $2x^3 + 4x^2 - 5$

4. คลิกปุ่ม เคลียร์ค่า ข้อมูลที่ปรากฏอยู่ก็จะหายไปดังรูปที่ 4.13



<u>รูปที่ 4.13</u> แสดงหน้าจอเมื่อมีการคลิกปุ่ม เคลียร์ค่า

5. ทดสอบฟังก์ชันการตรวจสอบข้อมูลโดยกรอกตัวอักษรลงในช่อง ค่าเริ่มต้น ค่าสะสม และค่าสิ้นสุด ดังรูปที่ 4.14



<u>รูปที่ 4.14</u> แสดงการตรวจสอบข้อมูล

เอกสารอ้างอิง

[1] สมศักดิ์ โชคชัยชุติกุล. 2550. Insight PHP ฉบับสมบูรณ์(พิมพ์ครั้งที่7). กรุงเทพฯ:โปรวิชั่น.

[2] บัญชา ปะสีละเตสัง. 2551. พัฒนาเว็บด้วยเทคนิค Ajax และ PHP. กรุงเทพฯ:ซีเอ็ดยูเคชั่น.

[3] บัญชา ปะสีละเตสัง. 2551. ออกแบบและพัฒนาเว็บไซต์ด้วย DHTML. กรุงเทพฯ:ซีเอ็ดยูเคชั่น

[4] ปิยะ โควินท์ทวีวัฒน์. 2549. คู่มือโปรแกรมภาษา SCILAB สำหรับผู้เริ่มต้น (พิมพ์ครั้งที่ 2). กรุงเทพฯ : ศูนย์ผลิตตำราเรียนสถาบันเทคโนโลยีพระจอมเกล้าเจ้าพระนครเหนือ.