

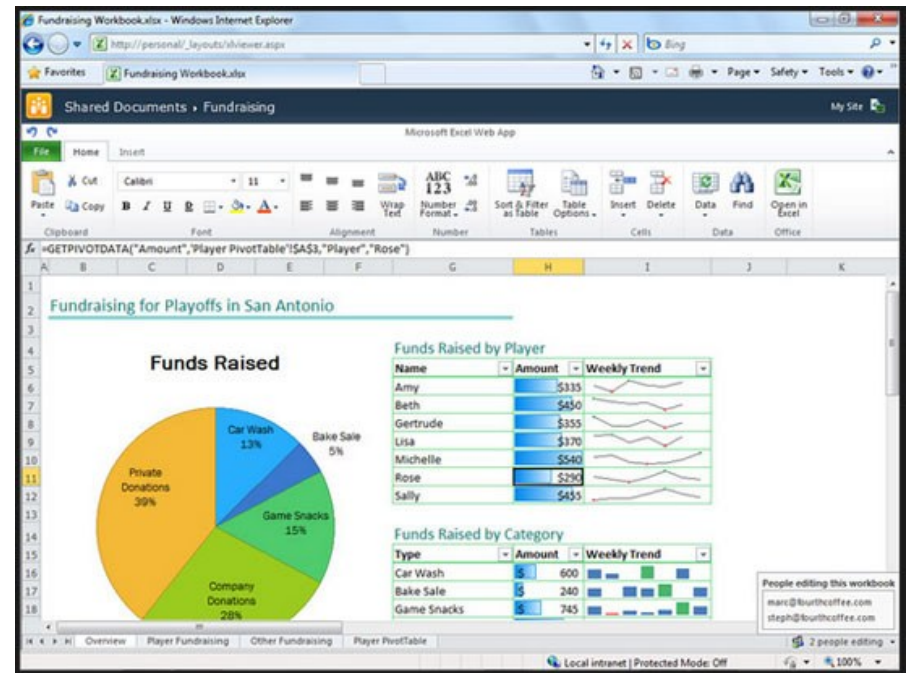
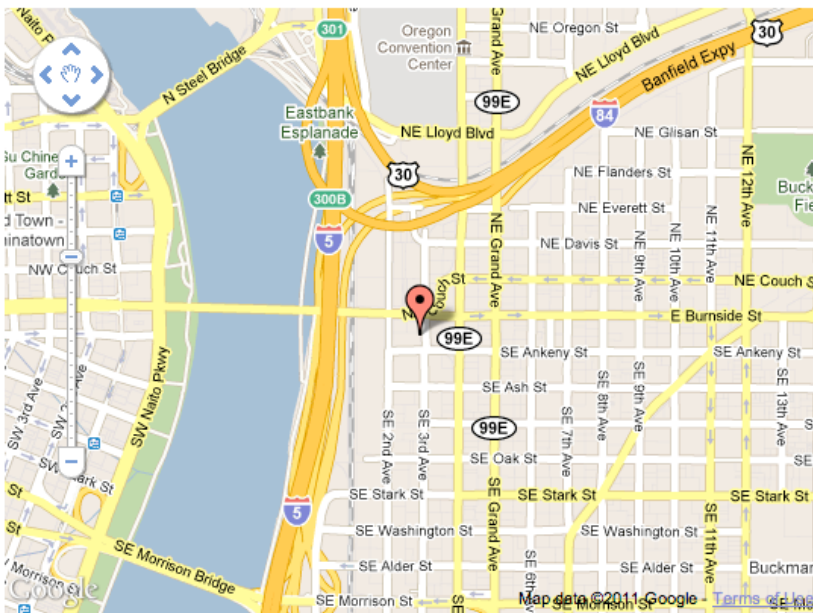
Web application development

PART I

Web Development

- Web Development หรือ Web Programming คือการออกแบบ Software Application ที่ทำงานบน Web Site (Web Applications: WebApps)

Our Location



Project (mini-twitter)

- สร้างเว็บเพจที่ทำงานเหมือน twitter
- มีการควบคุมสิทธิ์ในการทำงาน
 - แอดมินสามารถ สร้าง แก้ไขหรือ ลบข้อความได้
 - ผู้เข้าชมสามารถดูได้อย่างเดียว
- ตกแต่งหน้าตาเว็บให้สวยงาม
- เพิ่มเติมพิเศษ
 - ให้ผู้ชมสามารถเขียนคอมเมนต์ได้ด้วย

Network model for web service

Client-Server Model

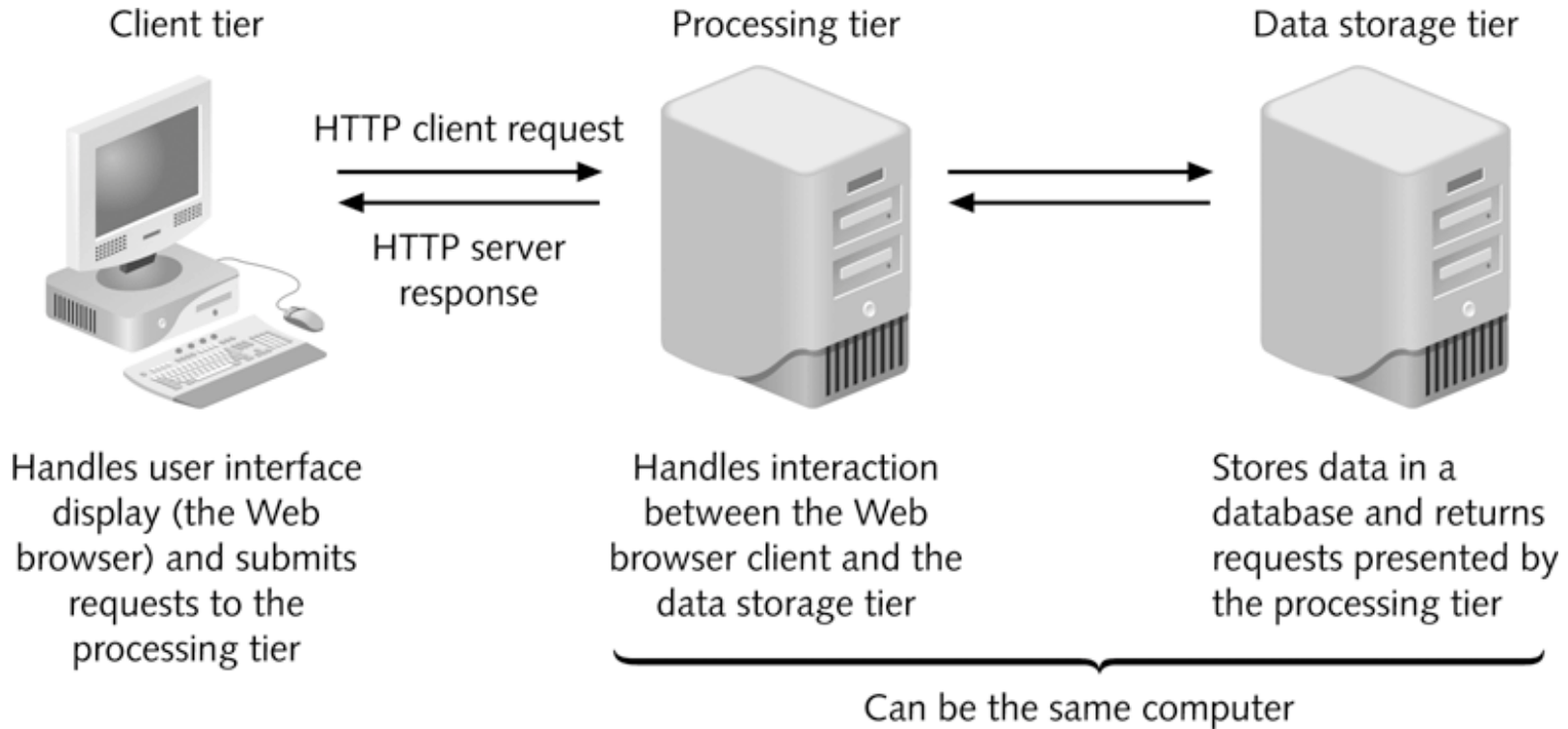
- Client-Server Model เป็นโครงสร้าง application ที่มีการแยกส่วนระหว่างฝั่งผู้ให้บริการ หรือทรัพยากร (Service Provider or Resource Provider) เรียกว่า ฝั่ง Server และ ผู้ขอใช้บริการ หรือทรัพยากร (Service Requester or Resource Requester) นั่นๆ หรือเรียกว่าฝั่ง Client
- โดยทั่วไปแล้วการสื่อสารระหว่าง Server และ Client จะทำผ่าน Computer Network โดยฝั่ง Client จะเป็นฝ่ายเริ่มต้นการสื่อสาร และ Server จะอยู่ในสถานะรอการเชื่อมต่อ (Connection)

CLIENT-SERVER ARCHITECTURE FOR WEB SERVICE

Three-Tier Architecture

- สถาปัตยกรรมแบบ Multi-tier (หรือ n-tier) คือสถาปัตยกรรมแบบ Client-Server ที่มีการแยกการแสดงผล (Presentation) การประมวลผล (Application Processing) และการจัดการข้อมูล (Data Management) ออกจากกัน
- หนึ่งในสถาปัตยกรรมแบบ Multi-tier ที่นิยมใช้กันอย่างแพร่หลายคือแบบ Three-Tier โดยประกอบด้วย 3 ส่วนคือ
 - Presentation Tier – User Interface (A PC)
 - Application Tier – Handles the interaction between the Web browser client and the data storage tier. (A server)
 - Data Storage Tier – Responsible for Data Storage (A database)

Three-Tier Architecture [2]

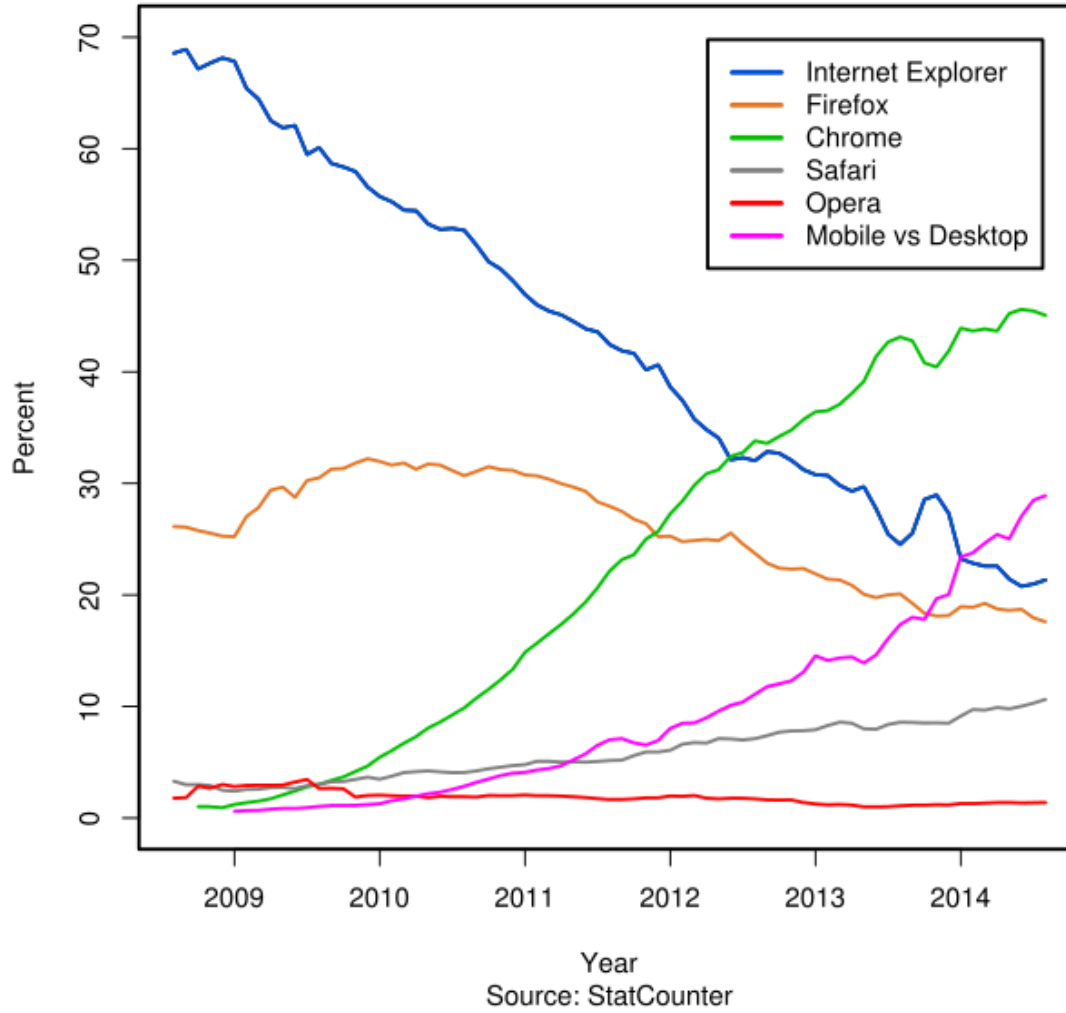


The design of a three-tier client/server system

1st Tier: Presentation

- Web browser
- คือซอฟต์แวร์ที่มีหน้าที่รับข้อมูลจาก server มาประมวลผลและแสดงผล ภาษาหลักในการสื่อสารระหว่าง server กับ web browser คือภาษา HTML

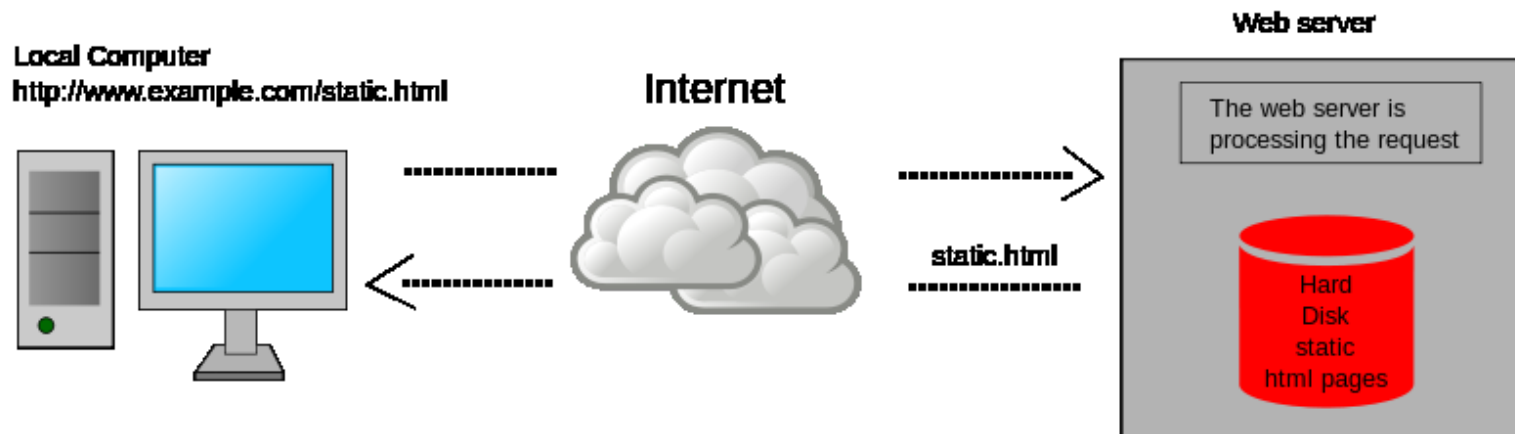
Usage share of web browsers



TWO TYPES OF WEBPAGES

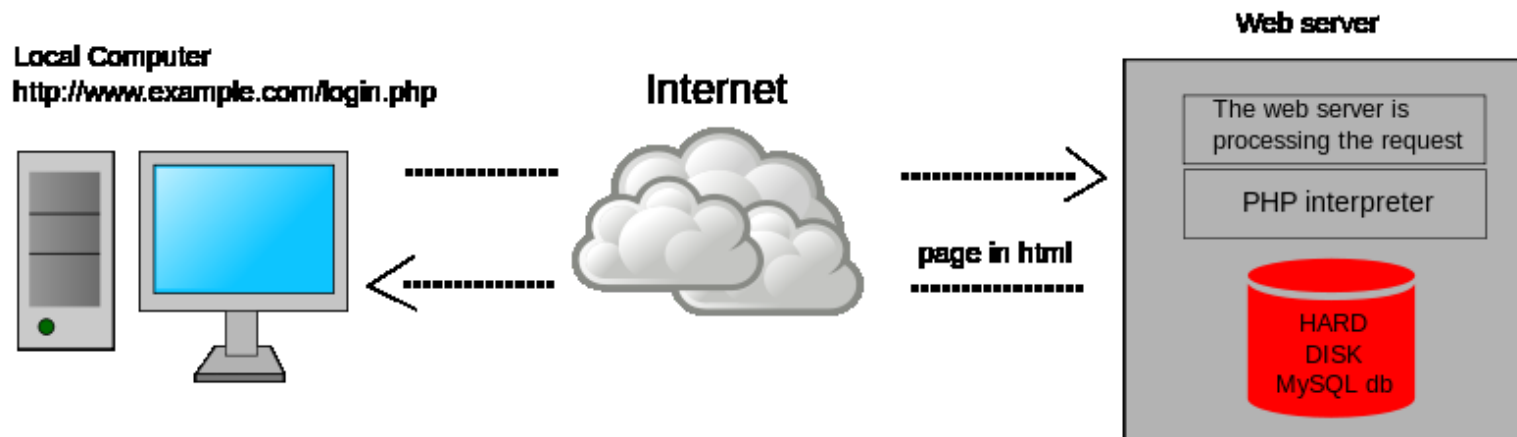
Static Webpage

- **Static Web Page** คือ เว็บไซต์ที่สร้างขึ้นมาจากภาษา HTML เพียงอย่างเดียวจะมีลักษณะเป็นเว็บเพจอย่างง่าย นำเสนอข้อมูลที่เป็นข้อความภาพ และเสียงแบบธรรมดา ไม่มีการโต้ตอบกับผู้ใช้ (Interactive)



Dynamic Webpage

- Dynamic Web Page คือ เว็บไซต์ที่มีลูกเล่นต่าง ๆ มีการโต้ตอบกับผู้ใช้ มีการประมวลผลต่าง ๆ มีการติดต่อกับฐานข้อมูล ได้แก่เว็บไซต์ที่พัฒนามาจากภาษา PHP, ASP, JSP, JavaScript, VbScript เป็นต้น



TWO PARADIGMS OF DYNAMIC WEB PROGRAMMING

Client-side programming

- Client-side programming เป็นการเขียนโปรแกรมที่ทำงานบน Browser ที่ฝั่ง Client
- อยู่ใน 1nd Tier ของ multi-tier architecture
- ตัวอย่างเช่น JavaScript
 - เป็น ภาษา Client-side scripting ที่ช่วยให้ Web Developer สร้าง dynamic web page ที่โต้ตอบกับผู้ใช้ได้ (interactive)
 - ใช้ในการคำนวณหรือตรวจสอบข้อมูลที่ไม่ซับซ้อน

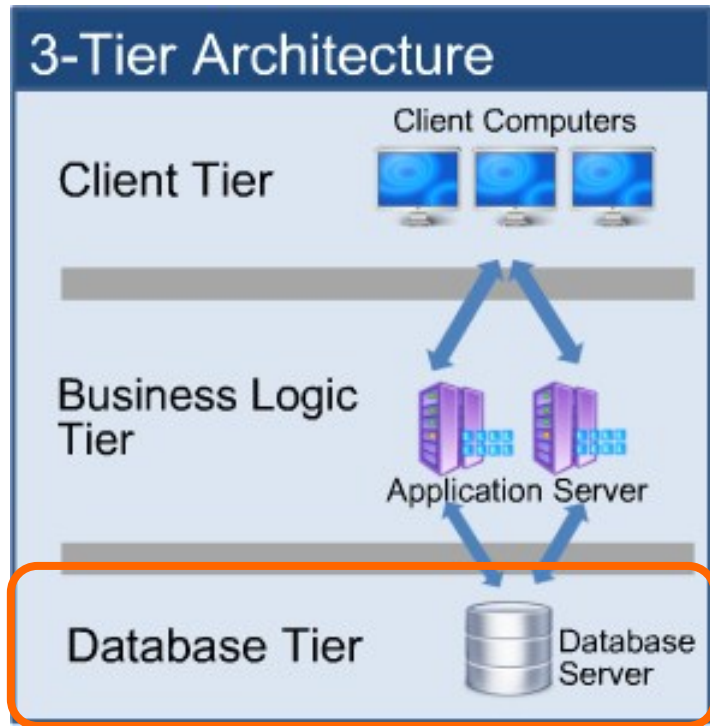
Server-Side Programming

- Server-side Programming เป็นการเขียนโปรแกรมที่ทำงานบนฝั่ง Web Server
- อยู่ใน 2nd Tier ของ multi-tier architecture
- สามารถใช้ภาษาเกือบทุกภาษาเขียนได้
 - ผ่านระบบที่เรียกว่า CGI
- แต่ที่นิยมมากก็จะเป็น scripting language
 - asp, php, ruby on rails

Server side programming using PHP

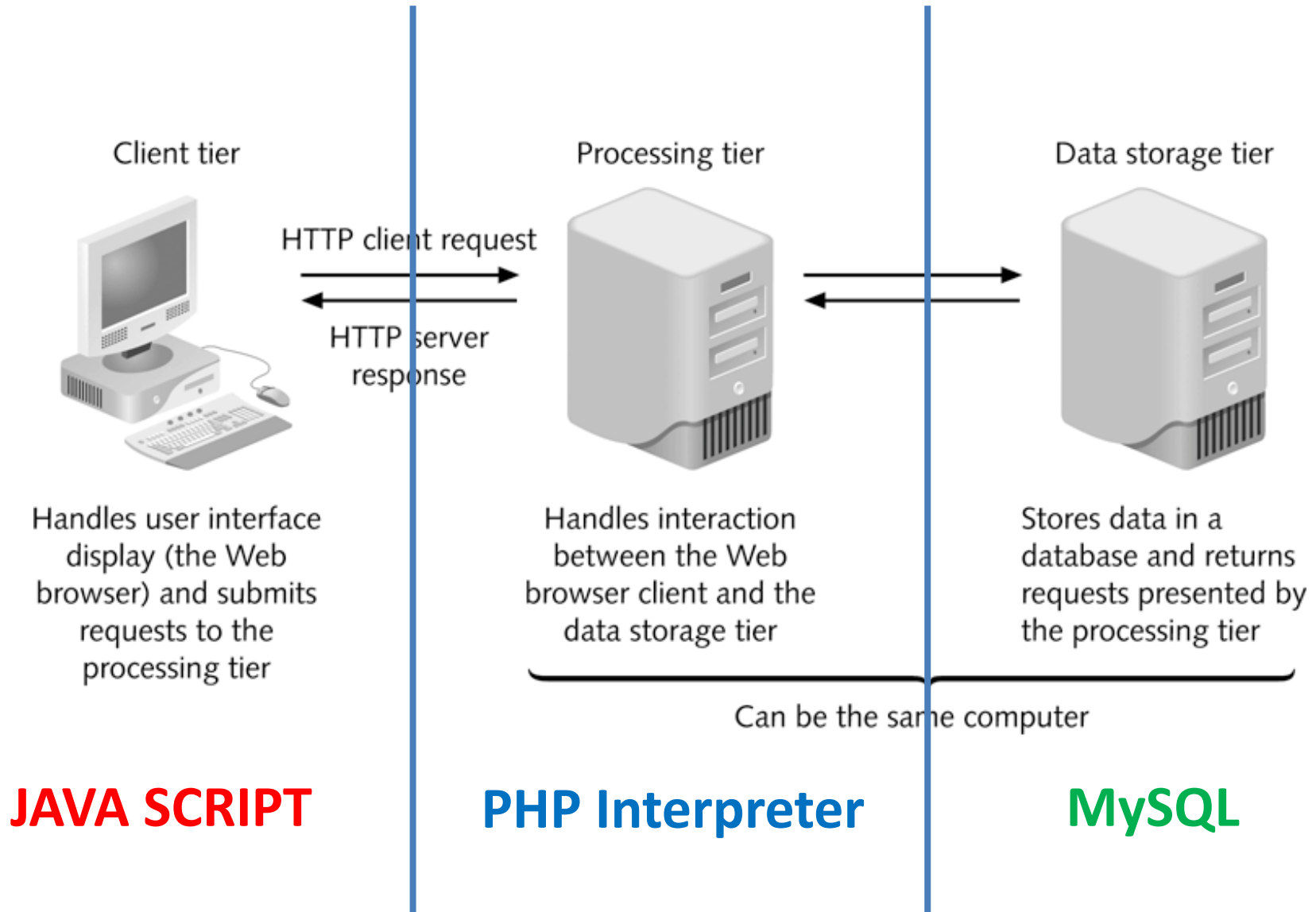
- PHP เป็น Open Source programming language
 - Open Source – software ที่ Code เปิดเผยและผู้ใช้สามารถเปลี่ยนแปลงได้
- ไม่สามารถเข้าถึงหรือเปลี่ยนแปลง browser ได้เหมือน Client Script
- ทำงานจากฝั่ง web Server เท่านั้น ไม่ว่าจะเป็นการประมวลผลหรือเชื่อมต่อกับฐานข้อมูล
- ทำงานร่วมกับ Web Server Application ได้หลากหลาย (Apache, IIS, etc.)
- Client-side script ใช้เพื่อควบคุม user interface และการคำนวณที่ไม่ซับซ้อน ในขณะที่ Server-side script ใช้เพื่อการประมวลผลที่ซับซ้อนและมีการเชื่อมต่อกับฐานข้อมูล

3rd Tier: Database: MySQL



- ในการประมวลผลข้อมูล มีความจำเป็นต้องจัดการข้อมูลจำนวนมากอย่างมีประสิทธิภาพ ซึ่งโดยทั่วไปจำเป็นต้องใช้ ฐานข้อมูล (Database)
- โดยฐานข้อมูลจะถูกจัดการโดยระบบจัดการฐานข้อมูล (Database Management System: DBMS)
- MySQL (My S-Q-L) เป็น DBMS ที่ได้รับความนิยมเป็นอันดับสองของโลก (July 2013: Wikipedia)
 - Open Source
 - Owned By Oracle
 - Free

Summarising



HTML Introduction

- HTML เป็นภาษาที่ใช้ในการเขียน Web Page ซึ่งสามารถสร้างให้แสดงผลให้อยู่ในรูปของตัวอักษร ภาพนิ่ง ภาพเคลื่อนไหว เสียง และยังสามารถเชื่อมโยงไปยังเว็บไซต์อื่นๆในระบบ Internet
- หน่วยงานหลักที่กำหนดมาตรฐาน HTML คือ World Wide Web Consortium ([W3C](#))

HTML Introduction [2]

- HTML ย่อมาจาก **HyperText Markup Language**
 - **HyperText** คือ ข้อความที่มีลิงค์ (Hyperlink or Link) เชื่อมโยงกับข้อความอื่น
 - **Markup Language** คือภาษาที่ประกอบด้วย Markup Tags ต่างๆ

HTML Tags

- เอกสาร HTML จะประกอบด้วย **Tag** และ ข้อความ

`<tagname>content</tagname>`

- ตัว Tag รวมทั้งข้อความระหว่าง Tag เปิดและปิด รวมแล้ว เรียกว่า **HTML Element**
- เอกสาร HTML มีนามสกุลเป็น `.htm` หรือ `.html`

HTML Versions

Version	Year
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2014 <- This class
HTML5.1	*stable release in 2016

เนื่องจากมีมาตรฐาน HTML อยู่หลาย Version ดังนั้นในแต่ละเอกสาร ควร
ต้องระบุด้วยว่าเป็นเอกสาร (HTML หรือ XHTML) Version ใด

The <!DOCTYPE> Declaration

- HTML5

```
<!DOCTYPE html>
```

- HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">
```

Tag <!DOCTYPE> ใช้เพื่อประกาศชนิด
ของเอกสาร HTML และ XHTML (รวมถึง XML)

การสร้างเอกสาร HTML [1/2]

- การเขียนเอกสาร HTML ประกอบไปด้วยโปรแกรม 2 ส่วน
 - Editor
 - เป็นโปรแกรมที่ใช้ในการเขียนภาษา HTML เพื่อสร้าง Web Page
 - ตัวอย่างเช่น
 - Notepad (Text Editor)
 - KompoZer (WYSIWYG: What You See is What You Get Editor)
 - Adobe Dreamweaver (WYSIWYG Editor)
 - Web browser
 - เอาไว้ดู web page ผลลัพธ์ที่เขียนไว้

การสร้างเอกสาร HTML [2/2]

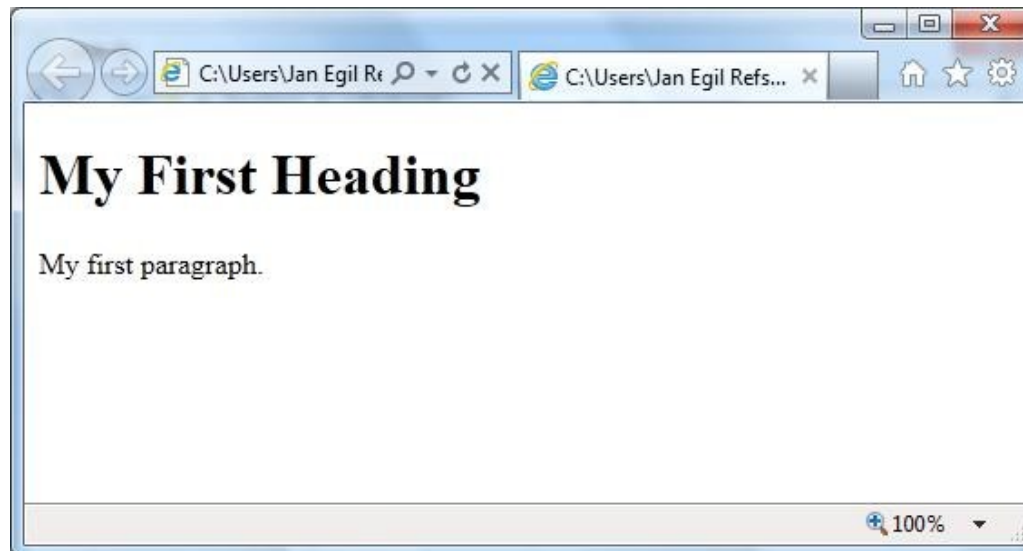


```
Untitled - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```



เอกสาร HTML

- เอกสาร HTML ประกอบด้วย 2 ส่วน คือ
 - ส่วนของคำสั่ง คือ tag เขียนอยู่ภายในเครื่องหมาย `< >` เช่น `
` โดยการแสดงผลใน Webpage จะไม่แสดง Tag เหล่านี้
 - ส่วนของข้อความที่ต้องการแสดงบน webpage
- เราสามารถเรียกดู HTML Tag บนแต่ละหน้า Webpage บน Browser โดยการเรียก Menu “View Source” (Chrome) หรือ “View Page Source” (Firefox) เป็นต้น

เอกสาร HTML [2]

- Tag มี 2 แบบ

- Tag เดี่ยว: Tag ที่มีคำสั่งเดียว สามารถใช้งานคำสั่งได้ ณ ตำแหน่งที่เราระบุ เช่น `
`
- Tag คู่: Tag ที่มี 2 ส่วน คือ Tag เปิด และ Tag ปิด
- โดย Tag ปิดมีรูปแบบเหมือน Tag เปิด แต่มีเครื่องหมาย / นำหน้า

`<ชื่อคำสั่ง> ข้อความที่ต้องการแสดง </ชื่อคำสั่ง>`

- เช่น

`<html> ข้อความที่ต้องการ </html>`

HTML Tags

- โดยปกติแล้วเราสามารถเขียน HTML Tag ด้วยตัวพิมพ์เล็ก (Lowercase) หรือตัวพิมพ์ใหญ่ (Uppercase) ก็ได้

`<HTML>` \equiv `<html>`

- W3C แนะนำให้ใช้ตัวพิมพ์เล็กใน HTML4 และ บังคับให้ใช้ตัวพิมพ์เล็กใน XHTML

โครงสร้างภาษา HTML

- ในการเขียนเอกสาร HTML 1 หน้า จะต้องประกอบไปด้วยคำสั่งหลักอยู่ 4 คำสั่ง
 - `<html>...</html>` เป็นคำสั่งที่กำหนดหน้าที่บอกจุดเริ่มต้น และจุดสิ้นสุดของเอกสาร HTML
 - `<head>...</head>` เป็นคำสั่งที่กำหนดส่วนหัวเรื่อง
 - `<title>...</title>` เป็นคำสั่งที่กำหนดข้อความที่ต้องการแสดงผลบนแถบชื่อเรื่อง
 - `<body>...</body>` เป็นคำสั่งที่กำหนดข้อความและรูปแบบคำสั่งใดๆ ที่ต้องการปรับแต่งเอกสารบนส่วนของจอภาพ และจะแสดงผลบนจอภาพเมื่อถูกเรียกใช้จาก Web Browser

โครงสร้างภาษา HTML [2]

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8"/>
```

```
    <title>
```

```
      ข้อความที่ต้องการแสดงผลบน Title Bar
```

```
    </title>
```

```
  </head>
```

```
  <body>
```

```
    ข้อความและคำสั่งใด ๆ ที่ต้องการปรับแต่งเอกสารบนส่วนของจอภาพ
```

```
  </body>
```

```
</html>
```

PHP - BASICS

PHP Basics: Syntax

- PHP standard ที่ใช้ในปัจจุบันเป็น PHP 5.x หรือ PHP 5
- เราสามารถใส่ PHP script ไว้ที่ส่วนไหนของไฟล์ PHP ก็ได้ โดยส่วนที่เป็น PHP script จะมีเครื่องหมาย `<?php ?>` ล้อมรอบโดยการประมวลผล จะเกิดขึ้นกับ code ที่อยู่ระหว่าง Tag เปิดและปิดนี้เท่านั้น
- ในกรณีที่ไฟล์นั้นไม่มี content อื่นๆ นอกจาก PHP code (pure PHP Code) ไม่ควรต้องใส่ tag ปิด `?>`

PHP Basics: Syntax [2]

- Content ของไฟล์ที่อยู่นอก Tag เปิดและปิดจะไม่ได้รับการประมวลผล
 - ทำให้เราสามารถแทรก content อื่นๆ เช่น HTML ลงในไฟล์ PHP ได้

```
<p>This is going to be ignored by PHP</p>
```

```
<?php echo 'While this is going to be parsed.'; ?>
```

```
<p>This will also be ignored by PHP</p>
```

- `echo` เป็นคำสั่ง PHP ที่ใช้แสดงข้อความที่อยู่ในเครื่องหมายคำพูด (e.g. "text" หรือ 'text')

PHP Basics: Syntax [3]

- แต่ละบรรทัดของคำสั่งในภาษา PHP ต้องจบด้วยเครื่องหมาย ;

```
<?php echo 'My name is John' ; ?>
```

- ชื่อคำสั่งหรือฟังก์ชัน (function) ในภาษา PHP มีลักษณะ case insensitive คือไม่มีความแตกต่างของตัวพิมพ์เล็กหรือพิมพ์ใหญ่ (echo = EcHo = ECHO...)
- แต่ตัวแปรในภาษา PHP เป็น แบบ case sensitive

- \$var

- \$VaR

- \$vAr

Three different variables

PHP Basics: Variables

- Variable หรือตัวแปรใช้เพื่อเก็บข้อมูล ในโปรแกรม เช่น

```
<?php
  $x = 5;
  $y = 6;
  $z = $x + $y;
  echo $z; // print out z
?>
```

```
x = 5
y = 6
z = x + y
z = ...
```

- ลักษณะการ แทนค่า หรือการคำนวณ เหมือนในวิชาคณิตศาสตร์
- ตัวแปรใน PHP ต้องขึ้นต้นด้วย \$ เสมอ
- ชื่อตัวแปรต้องประกอบด้วยตัวอักษร (aA-zZ) ตัวเลข (0-9) และ underscore (_)
- ขึ้นต้นด้วยตัวอักษร หรือ underscore เท่านั้น

PHP Basics: Comments

- Comments หมายถึงส่วนใดๆ ของ code ที่ไม่ถือเป็นคำสั่งแต่ใส่ไว้เพื่อเป็นการอธิบาย code เพื่อสะดวกในการทำความเข้าใจ Code
- ตัวอย่าง

```
<?php
    echo 'This is a test'; // This is a one line comment
    /* This is a multi line comment
       yet another line of comment */
    echo 'This is yet another test';
    echo 'One Final Test'; # This is a one-line comment
?>
```

PHP Basics: Operators

● Arithmetic

Example	Name	Result
$-\$a$	Negation (นิเสธ)	Opposite of $\$a$.
$\$a + \b	Addition (การบวก)	Sum of $\$a$ and $\$b$.
$\$a - \b	Subtraction (การลบ)	Difference of $\$a$ and $\$b$.
$\$a * \b	Multiplication (การคูณ)	Product of $\$a$ and $\$b$.
$\$a / \b	Division (การหาร)	Quotient of $\$a$ and $\$b$.
$\$a \% \b	Modulus (การหารเก็บเศษ)	Remainder of $\$a$ divided by $\$b$.

PHP Basics: Operators [2]

● Comparison

Example	Name	Result
$\$a == \b	Equal	TRUE if $\$a$ is equal to $\$b$ after type juggling.
$\$a != \b $\$a <> \b	Not equal	TRUE if $\$a$ is not equal to $\$b$ after type juggling.
$\$a < \b	Less than	TRUE if $\$a$ is strictly less than $\$b$.
$\$a > \b	Greater than	TRUE if $\$a$ is strictly greater than $\$b$.
$\$a <= \b	Less than or equal to	TRUE if $\$a$ is less than or equal to $\$b$.
$\$a >= \b	Greater than or equal to	TRUE if $\$a$ is greater than or equal to $\$b$.

PHP Basics: Operators [3]

- Assignment: การกำหนดค่าให้ variable ต่างๆ ทำได้โดยการ
ใช้เครื่องหมายเท่ากับ =
 - การกำหนดค่า จะอยู่ในรูปแบบของการนำค่าที่อยู่ด้านขวา ของ
operator ไปใส่ใน variable ที่อยู่ด้านซ้าย เช่น

```
$a = 3;  
$b = "Hello ";
```

\$a ← 3

Assignment	Same as
<code>\$a += \$b</code>	<code>\$a = \$a + \$b</code>
<code>\$a -= \$b</code>	<code>\$a = \$a - \$b</code>
<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>
<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>
<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>
...	...

PHP - CONTROL STRUCTURES

PHP Control Statement: Basics

- Control Statement มีเพื่อกำหนดเงื่อนไขในการทำตามคำสั่ง
 - เลือกทำงานหรือไม่ เพียงครั้งเดียว เช่น if, if..else, elseif, switch
 - เลือกทำงานแบบวนซ้ำ เช่น for, while, do..while

- ส่วนประกอบของ Control Statement

- คำสั่งควบคุม (if, while, for,...)
- เงื่อนไข (expr)
- คำสั่ง/กลุ่มของคำสั่งที่ต้องการ (statement)

```
if (expr)  
    statement
```

```
while (expr)  
    statement
```

```
for (expr1,expr2, expr3)  
    statement
```

PHP Control Statement:if, if..else, elseif

- Syntax

```
if (expr){  
    statement;  
}
```

```
if (expr){  
    statement_A;  
} else {  
    statement_B;  
}
```

```
if (expr_1){  
    statement_A;  
} elseif (expr_2){  
    statement_B;  
}else {  
    statement_C;  
}
```

- Alternative Syntax

```
if (expr_1):  
    statement_A;  
elseif (expr_2):  
    statement_B;  
else:  
    statement_C;  
endif;
```

***Note:** Mixing syntaxes in the same control block is not supported.

Conditional expression (expr)

- Boolean value = TRUE, FALSE
- สามารถสร้างเงื่อนไขได้ในรูปแบบต่างๆ เช่น
 - ค่าของตัวแปร ($\$a$), ($!\b)
 - Comparison operator ($\$a == 10$), ($++\$a > 1$)
 - ผลลัพธ์ของฟังก์ชัน
 $is_string(\$val)$, $isset(\$val)$, $empty(\$val)$, $is_null(\$val)$
 - Logical Operator ($\$a > 0$) $||$ ($\$b < 1$), ($0 > \a) $\&\&$ ($\$a < 5$)

Exercise 1

- มาลองเติมโค้ดที่กล่าวคำท้ายได้ถูกต้องตามช่วงเวลาต่อไปนี่ให้สมบูรณ์

```
<?php
```

```
date_default_timezone_set("Asia/Bangkok");  
$dd = date("H");
```

```
...
```

```
?>
```

+ Note +

- ฟังก์ชัน `date_default_timezone_set("Asia/Bangkok")`
 - เอาไว้ใช้บอกให้ php รู้ว่าเราอยู่ในเขตเวลาไหน
 - ในกรณีนี้ เราบอกว่า เราอยู่ทวีป Asia เมือง Bangkok
- ฟังก์ชัน `date("H")`
 - เอาไว้เรียกวันที่หรือเวลาปัจจุบัน
 - การระบุรูปแบบของผลลัพธ์ทำได้โดยกำหนดค่าของตัวแปรขาเข้า
 - H เป็นการระบุว่า ให้ฟังก์ชันบอก hour of the day ในรูปแบบ 24 ชั่วโมง
 - ถ้าใช้ h (อักษรตัวเล็ก) เราจะได้ hour of the day ในรูปแบบ 12 ชั่วโมง

PHP Control Statement: Switch/Case

- สามารถทำงานได้เหมือน if..elseif

```
if ($i == 0) {  
    echo "i equals 0";  
} elseif ($i == 1) {  
    echo "i equals 1";  
} elseif ($i == 2) {  
    echo "i equals 2";  
}
```

```
switch ($i) {  
    case 0:  
        echo "i = 0";  
        break;  
    case 1:  
        echo "i = 1";  
        break;  
    case 2:  
        echo "i = 2";  
        break;  
    default:  
        echo "no match any cases";  
}
```

Switch..Case: Example

```
<?php
switch ($i) {
    case "apple":
        echo "i is apple";
        break;
    case "bar":
        echo "i is bar";
        break;
    case "cake":
        echo "i is cake";
        break;
}
?>
```

```
<?php
switch ($i) {
    case 0:
    case 1:
    case 2:
        echo "i is less than 3 but
            not negative";
        break;
    case 3:
        echo "i is 3";
    }
?>
```


Exercise 2

- จงปรับปรุงโปรแกรมก่อนหน้าให้แสดงผลเวลาเป็นคำในภาษาไทย
- ดังตาราง
 - ถ้า \$dd เท่ากับ 1 ให้แสดงผลว่า ตี 1 ไล่ไปเรื่อยๆ ถึง ตี 5
 - ถ้า \$dd เท่ากับ 6 ให้แสดงว่า 6เช้า ไปเรื่อยๆถึง 11โมงเช้า
 - ถ้า \$dd เท่ากับ 12 ให้แสดงว่าเที่ยงวัน
 - ถ้า \$dd เท่ากับ 13 ให้แสดงว่า บ่าย 1 โมง ไปเรื่อยๆถึง บ่าย 5 โมง
 - ถ้า \$dd เท่ากับ 18 ให้แสดงว่า หกโมงเย็น
 - ถ้า \$dd เท่ากับ 19 ให้แสดงว่า 1ทุ่ม ไล่ไปเรื่อยๆ จนถึง 5 ทุ่ม
 - ถ้า \$dd เท่ากับ 0 ให้แสดงว่า เที่ยงคืน

PHP Control Statement: Loop

- Loop เป็นคำสั่งการวนทำงานตามเงื่อนไขที่กำหนด
- คำสั่งวนลูปใน PHP ประกอบด้วย
 - while เช็คเงื่อนไขก่อนทำงาน
 - do..while ทำงานก่อนเช็คเงื่อนไข
 - for กำหนดค่าและเช็คเงื่อนไขก่อนทำงาน
 - foreach ใช้ร่วมกับตัวแปรประเภท array
- คำสั่งเพื่อกระโดดออกจากลูปเป็นกรณีพิเศษ
 - Break; Continue;

PHP Control Statement: Loop[2]

- ส่วนประกอบที่สำคัญ
 - กำหนดค่าเริ่มต้นของเงื่อนไข
 - ตรวจสอบเงื่อนไข
 - ชุดคำสั่งที่ต้องการทำงานแบบลูป
 - ปรับปรุงค่าเงื่อนไข

PHP Control Statement: while

- Syntax

```
while (expr){  
    statement;  
    ...  
}
```

OR

```
while (expr):  
    statement;  
    ...  
endwhile;
```

- Example

```
<?php  
$i = 1;
```

1. กำหนดค่าเริ่มต้น

```
while ($i <= 10){
```

2. ตรวจสอบเงื่อนไข

```
    echo $i;  
    $i++;
```

3. คำสั่งที่ต้องการวนลูป

```
}
```

```
?>
```

4. ปรับปรุงค่าเงื่อนไข

PHP Control Statement: `while`[2]

- Sourcecode

```
<?php
    $x=1;
    while($x<=5){
    echo "The number is: $x <br>";
    $x++;
    }
?>
```

- Result

The number is 1
The number is 2
The number is 3
The number is 4
The number is 5

PHP Control Statement: **do while**

- Syntax

```
do{  
    statement;  
    ...  
} while (expr);
```

- Example

```
<?php  
$i = 1;
```

กำหนดค่าเริ่มต้น

```
do{
```

```
    echo $i;  
    $i++;
```

คำสั่งที่ต้องการวนลูป

ปรับปรุงค่าเงื่อนไข

```
} while ($i <= 10);
```

```
?>
```

ตรวจสอบเงื่อนไข

PHP Control Statement: `do while`[2]

- Sourcecode

```
<?php
    $x=6;
    do{
    echo "The number is: $x <br>";
    $x++;
    } while($x<=5);
?>
```

- Result

The number is 6

PHP Control Statement: **for**

- Syntax

```
for (init;expr;update){  
    statement;  
    ...  
}
```

OR

```
for (init;expr;update):  
    statement;  
    ...  
endfor;
```

init - กำหนดค่าเริ่มต้นของลูป
expr - เงื่อนไขที่ตรวจสอบ
update - การปรับปรุงค่า

- Example

```
for ($i = 1; $i <= 10; $i++){  
    echo $i;  
}
```


PHP Control Statement: for[2]

- Sourcecode

```
<?php
```

```
    for($x=0;$x<5;$x++){  
        echo "The number is: $x";  
    }
```

```
?>
```

- Result

The number is 0

The number is 1

The number is 2

The number is 3

The number is 4

Exercise 3

- จงเขียนโปรแกรมเพื่อแสดงคำว่า cuckoo เท่ากับค่าของชั่วโมงที่ได้จาก function date()
- แสดง cuckoo แต่ละครั้งให้เว้นบรรทัดด้วย

PHP Control Statement: **foreach**

- ใช้สำหรับทำงานกับข้อมูลที่เก็บไว้ในตัวแปร Array
- Syntax

```
foreach (array_expression as $value){  
    statement  
}
```

สำหรับ Array ที่มีการกำหนด value เพียงอย่างเดียว

```
foreach (array_expression as $key => $value){  
    statement  
}
```

สำหรับ Array ที่มีการกำหนด key และ Value

PHP Control Statement: `foreach`[2]

- Sourcecode

```
<?php
$colors =array("red","green","blue","yellow");
foreach ($colors as $value)
{
    echo "$value <br>";
}
?>
```

- Result

```
red
green
blue
yellow
```

PHP Control Statement: `foreach`[3]

- Sourcecode

```
<?php
```

```
    $age = array(  
        "Peter"=>"35",  
        "Ben"=>"37",  
        "Joe"=>"43");
```

```
foreach($age as $x=>$x_value){  
    echo "Key=".$x.", Value=".$x_value;  
    echo "<br>";  
}
```

```
?>
```

- Result

Key=Peter, Value=35

Key=Ben, Value=37

Key=Joe, Value=43

PHP Control Statement: **break;** and **continue;**

- เป็นคำสั่งในการออกจากกลุ่ม หรือโครงสร้าง if/switch

```
switch ($a) {  
    case 0: continue;  
    case 1: break;  
}
```

----- both jumps here -----
|<-----

```
while ($a) {  
    continue;  
    break;  
}
```

<-----
--- goes back here ---
----- jumps here -----
|<-----

PHP Control Statement: `break` and `continue`[2]

- Sourcecode

```
<?php
$i = 10;
while (--$i){
    if ($i == 8){
        continue;
    }if ($i == 5){
        break;
    }
    echo $i . "\n"
}
?>
```

- Result

9
7
6

HTML - FORM

- เขียนโปรแกรม PHP เพื่อสร้างเอกสาร HTML
- รับข้อมูลจาก Form แล้วประมวลผล แล้วสร้างเอกสาร HTML ขึ้นมาโต้ตอบ

HTML Forms and Input

- ฟอรั่มในเอกสาร HTML คือส่วนหนึ่งของเอกสาร ที่มีหน้าที่ส่งข้อมูลไปยัง Server โดยนอกจาก Element ปรกติ แล้วยังประกอบด้วย Element พิเศษ ที่เรียกว่า Control
- Control คือ Element ที่ใช้โต้ตอบ (Interact) กับ User ซึ่งมีหลายชนิดได้แก่

- ▣ text input
- ▣ checkbox
- ▣ radio
- ▣ file
- ▣ reset
- ▣ submit
- ▣ textarea
- ▣ select

Name	Value
Name	<input type="text"/>
Sex	<input type="radio"/> Male <input checked="" type="radio"/> Female
Eye color	<input type="text" value="green"/>
Check all that apply	<input type="checkbox"/> Over 6 feet tall <input type="checkbox"/> Over 200 pounds
Describe your athletic ability:	
<input type="text"/>	
<input type="button" value="Enter my information"/>	

HTML Forms and Inputs

- Tag `<form>` ใช้เพื่อการสร้างฟอร์ม (โดยที่ตัว Tag จะไม่แสดงผลใน browser) การสร้างฟอร์มสามารถทำได้ในรูปแบบ `<form action="http://somesite.com/action.php">`
 - input elements
 - </form>
 - Attribute `action` ใช้กำหนด URI ของ script ที่จะดำเนินการเมื่อมีการ submit ฟอร์ม
- Tag `<input>` ใช้ระบุ Control Elements ชนิดต่างๆ เพื่อรับข้อมูลจากฝั่ง user

HTML Control Elements [1]

- **text** ใช้รับ Input แบบบรรทัดเดียวจาก User หากต้องการรับ input หลายๆ บรรทัด ให้ใช้ **textarea** แทน (**text** default size = 20 ตัวอักษร)

```
<form>
```

```
  First name: <input type="text" name="firstname"><br>
```

```
  Last name: <input type="text" name="lastname">
```

```
</form>
```

Output

- Attribute **name** (control name) ใช้เพื่อตั้งชื่อ element เพื่อการอ้างอิง ผ่าน script หรือ CSS

First name:

Last name:

*กรณีต้องการรับ input แบบซ่อนข้อความ เช่น password ให้ใช้ type เป็น password
<input type="password"...

HTML Control Elements [2]

- **textarea** ใ้รับ Input แบบหลายบรรทัดจาก User โดยมี attribute เช่น

- rows กำหนดจำนวนบรรทัด
- cols กำหนดความกว้างเป็น character

```
<form action="demo_form.asp" id="usrform">  
  Name: <input type="text" name="username">  
  <input type="submit">  
</form>  
<textarea name="comment" form="usrform">  
  Enter text here...  
</textarea>
```

Output

Name:

Enter text here...

***textarea อยู่นอก Tag <form>**
แต่ถือเป็นส่วนหนึ่งของ form
และอ้างถึง form id โดยใช้
attribute form

HTML Control Elements [3]

- **checkbox** – มีลักษณะเป็นสวิตช์เปิด/ปิด (on/off) ที่ User กำหนดค่า
 - ค่า "on" เกิดขึ้นเมื่อ checkbox นั้นๆ มีค่า Attribute "checked"
 - user สามารถเลือกได้มากกว่า 1 checkbox
 - checkbox หลายๆ อันสามารถใช้ control name ร่วมกันได้

```
<form>
  <input type="checkbox" name="vehicle[]" value="Bike">I have a
bike<br>
  <input type="checkbox" name="vehicle[]" value="Car">I have a
car
</form>
```

Output

- I have a bike
- I have a car

- ▣ Attribute **value** คือค่าที่จะถูกส่งไปยัง script เมื่อมีการ submit ฟอร์ม (ในกรณีนี้ จะเป็นค่าของตัวแปรชื่อ vehicle)

HTML Control Elements [4]

- **radio** ใช้รับ Input ลักษณะเดียวกับ checkbox

- User สามารถเลือกได้เพียง 1 ตัวเลือกเท่านั้นจาก control name 1 ชื่อ

```
<form>  
  <input type="radio" name="sex" value="male"  
checked>Male<br>  
  <input type="radio" name="sex" value="female">Female  
</form>
```

Output

- Male
- Female

- Attribute **checked** กำหนดค่า default ของตัวเลือก (ใช้ได้เฉพาะ radio และ checkbox)

HTML Control Elements [5]

- **file** ใช้รับ Input ที่มีลักษณะเป็นการเลือกไฟล์ผ่าน User

Interface ของ browser

```
<form>
```

```
  Select File: <input type="file" name="filename"><br>  
</form>
```

Output Select File: No file chosen

- **hidden** ใช้ส่งข้อมูลที่ไม่แสดงผลให้ user เห็นไปกับข้อมูลอื่นๆ ของฟอร์ม

```
<form>
```

```
  <input type="hidden" name="form_page" value="3">  
</form>
```


HTML Control Elements [6]

- buttons – ปุ่มที่อยู่ใน form มี 3 ประเภทได้แก่
 - submit button – ส่ง form ไปยัง Server (type="submit")
 - reset button – ล้างข้อมูลใน form กลับไปเป็นค่า default (type="reset")
 - push button – ไม่มีหน้าที่ตายตัว Developer สามารถผูก button เข้ากับ Client-Side Script เพื่อให้ทำงานต่างๆ กันได้ (type="button")
 - เราสามารถสร้างปุ่ม โดยใช้ Tag <button> หรือ <input> ก็ได้

```
<form name="input" action="action.php" method="get">  
  Username: <input type="text" name="user">  
  <input type="submit" value="Submit">  
</form>
```

Output

Username: *Notice the **method** attribute

The **method** Attribute

- การส่งฟอร์ม (Form Submission) สามารถทำได้สองวิธี
 - ส่งผ่าน Method **get**
 - ข้อมูลที่ส่งไปจะส่งไปในลักษณะเป็นส่วนต่อของ URI ที่ระบุใน attribute **action** คั่นด้วยเครื่องหมาย **?** และ **&** เช่น
 - `http://www.somesite.com/login.php?user=adam&password=1234`
 - ส่งผ่าน Method **post**
 - ข้อมูลจะไม่แสดงบน URI ทำให้ปลอดภัยกว่าและรองรับปริมาณข้อมูลมากกว่า

HTML Control Elements [7]

- Tag `select` ใช้รับ Input ในลักษณะ dropdown menu มีลักษณะการทำงานเหมือน radio button โดยมี attribute ดังนี้
 - `size` กำหนดจำนวนตัวเลือกที่แสดง (แสดงผลเป็นลักษณะ scroll box)
 - `multiple` กำหนดให้สามารถเลือกได้มากกว่า 1 ตัวเลือก

```
<form>
```

```
  <select name="car">
```

```
    <option value="volvo">Volvo</option>
```

```
    <option value="saab">Saab</option>
```

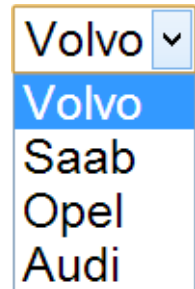
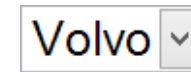
```
    <option value="mercedes">Mercedes</option>
```

```
    <option value="audi">Audi</option>
```

```
  </select>
```

```
</form>
```

Output



- Attribute `name` (control name) ใช้เพื่อตั้งชื่อ element เพื่อการอ้างอิง ผ่าน script หรือ CSS

HTML `<fieldset>` Tag

- Tag `fieldset` ใช้เพื่อจัดหมวดหมู่ Element ที่เกี่ยวข้องกันในฟอร์ม

```
<form>  
  <fieldset>  
    <legend>Personalia:</legend>  
    Name: <input type="text"><br>  
    Email: <input type="text"><br>  
    Date of birth: <input type="text">  
  </fieldset>  
</form>
```

Output

Example From: http://www.w3schools.com/tags/tag_fieldset.asp

Personalia: _____

Name:

Email:

Date of birth:

Exercise 4

- สร้างฟอร์มด้วย HTML ที่มีช่องไว้กรอกน้ำหนักและส่วนสูง
- สร้างปุ่มกดด้วย

PHP-FORM PROCESSING

PHP Basics: Variables [2]

- Variable จากแหล่งภายนอก

- เมื่อมีการส่ง Form ด้วยวิธี GET หรือ POST ไปยัง php script เราสามารถเข้าถึงค่าต่างๆ หรือ ข้อมูลที่ส่งมาได้โดยการเรียก variable `$_POST['variable_name']` หรือ `$_GET['variable_name']`

- ตัวอย่าง HTML form

```
<form action="greeting.php" method="post">  
  Name: <input type="text" name="username" /><br/>  
  Email: <input type="text" name="email" /><br/>  
  <input type="submit" name="submit" value="Submit me!" />  
</form>
```

PHP Basics: Variables [3]

● Variable จากแหล่งภายนอก [2]

```
<head>
  <meta charset="utf-8" />
  <title>Greetings!</title>
</head>
<body>
  Hello <?php echo $_POST["username"]; ?><br/>
  Your email address is: <?php echo $_POST["email"]; ?>
</body>
```

- เราสามารถใช้ " " หรือ ' ' ก็ได้ในการดึงค่าที่ถูกส่งมากับฟอร์มผ่าน variable `$_POST` (หรือ `$_GET`)

```
Hello, John
Your email address is: john@email.com
```


Exercise 5: HTML-Form & PHP basic

- จงสร้างโปรแกรมเพื่อคำนวณค่า ดัชนีมวลกาย (BMI) ของผู้ใช้ โดย
- โปรแกรมมีฟอร์มเพื่อ ให้ผู้ใช้ กรอกชื่อ น้ำหนักและส่วนสูง
- จากนั้น BMI สามารถคำนวณได้จากสูตร
 - $BMI = \frac{\text{น้ำหนัก}}{\text{ส่วนสูง}^2}$
- จากผลการคำนวณให้แสดงผลดังต่อไปนี้
 - ถ้า $BMI < 20$ ให้แสดงผลว่า Underweight
 - ถ้า BMI อยู่ระหว่าง 20 ถึง 24.9 ให้แสดงผลว่า Normal
 - ถ้า อยู่ระหว่าง 25 ถึง 29.9 ให้แสดงผลว่า Overweight
 - ถ้ามากกว่านี้ ให้แสดงผลว่า You are fat

PHP - FUNCTION

PHP Function: Function Type

- ฟังก์ชัน (Function) เป็นกลุ่มของคำสั่งสามารถเรียกใช้เป็นชุด
 - Predefined Function เป็นฟังก์ชันที่อยู่ใน libraries ของ PHP แล้วสามารถเรียกใช้งานได้ทันที เช่น
 - การเชื่อมต่อฐานข้อมูล
 - การทำงานเกี่ยวกับตัวแปรประเภทต่างๆ เช่น String, Array
 - รูปแบบการแสดงผล Date-Time
 - ตรวจสอบค่าของตัวแปร (Variable Handling- isset(), empty())
 - User defined Function เป็นฟังก์ชันที่ผู้ใช้งานเขียนขึ้นเองเพื่อสามารถเรียกใช้ซ้ำได้

PHP: User-defined Functions

- เป็นฟังก์ชันประเภทที่โปรแกรมเมอร์สามารถเขียนขึ้นเอง โดยมีการระบุรูปแบบตัวแปรรับเข้าและส่งออก (parameter and return type)
- ประโยชน์
 - ใช้เพื่อแตกปัญหาเป็นงานย่อยๆ ที่มีขอบเขตชัดเจน
 - สามารถเรียกใช้งานซ้ำๆ ได้โดยใช้โค้ดชุดเดิม
 - ง่ายต่อการแก้ไขโปรแกรม

The Need of Function

- ให้แสดงผลการคำนวณเส้นผ่าศูนย์กลาง, เส้นรอบวงและพื้นที่ของวงกลมที่มีรัศมี = 2 โดย $\pi = 3.14$

- Simple Calculation

```
<?php
//หาเส้นผ่าศูนย์กลาง
$radius=2;
$diameter = 2*$radius;
echo "Diameter = ". $diameter;

//หาเส้นรอบวง =2πr = πd
$perimeter= 2*3.14*$radius;
echo "Perimeter = ". $perimeter;

//หาพื้นที่วงกลม=πr2
$area= 3.14*$radius*$radius;
echo "Area = ". $area;
?>
```

The Need of Function

- ให้แสดงผลการคำนวณ เส้นผ่าศูนย์กลาง, เส้นรอบวงและพื้นที่ของ วงกลมที่มีรัศมีดังต่อไปนี้ โดยให้ $\pi = 3.14$
 - Radius = 2, 4, 5, 7, 11
- จากตัวอย่างโค้ดก่อนหน้า เราต้องเขียนซ้ำ 5 ครั้งโดยเปลี่ยนตัวแปรค่า radius ไปเรื่อยๆ
 - เสียเวลา
 - ถ้ามีการแก้ไขหรือเพิ่มเติม เราต้องแก้ทั้งหมด
 - ถ้าลืมแก้ไขค่า radius ผลลัพธ์ก็ไม่ถูกต้อง

PHP: User-defined Functions

- จากโจทย์ ให้คำนวณค่า เส้นผ่าศูนย์กลาง, เส้นรอบวงและพื้นที่วงกลม
- แแตกปัญหาออกเป็น
 - คำนวณเส้นผ่าศูนย์กลาง
 - คำนวณเส้นรอบวง
 - คำนวณพื้นที่วงกลม
 - แสดงผลการคำนวณ
 - การทำซ้ำสำหรับค่า R ที่เปลี่ยนไป

PHP: Definition of Function

- Syntax of Function

```
<?php
//define a function
function name_of_function(parameters) {
    statement_1;
    statement_2;
    return (); //อาจมีหรือไม่มีก็ได้ขึ้นกับรูปแบบการทำงาน
}

// call a function
name_of_function(parameters);
?>
```


Circle Function

```
<?php
```

```
define("PI",3.14);
```

```
//หาเส้นผ่าศูนย์กลาง
```

```
function diameter($r){  
return 2*$r;  
}
```

```
//หาเส้นรอบวง =  $2\pi r = \pi d$ 
```

```
function perimeter($r){  
return (2*PI*$r);  
}
```

```
//หาพื้นที่วงกลม =  $\pi r^2$ 
```

```
function area($r){  
return (PI*$r*$r);  
}
```

```
$circleList =array(2,4,5,7,11);
```

```
foreach($circleList as $r){
```

```
echo "R: ". $r."<br>";
```

```
echo "Diameter :".  
diameter($r)."<br>";
```

```
echo "Perimeter :".  
perimeter($r)."<br>";
```

```
echo "Area :".  
area($r)."<br>";
```

```
}  
?>
```

- ฟังก์ชันประกอบด้วย
 - ส่วนคำสั่งในฟังก์ชัน - ชุดคำสั่งที่ต้องการให้โปรแกรมทำงานในฟังก์ชันนั้นๆ
 - พารามิเตอร์ - ค่าตัวแปรที่รับเข้ามาคำนวณในฟังก์ชัน
 - การคืนค่า - ผลลัพธ์ที่ได้จากฟังก์ชัน (อาจมีหรือไม่มีก็ได้)

PHP Function: Parameter

- การรับค่าพารามิเตอร์ของ PHP มีได้สองแบบ
 - Pass by Value – ส่งค่าของตัวแปร
 - Pass by Reference – ส่งค่าที่¹อยู่ของตัวแปร

```
function swap(&$a, &$b) {
```

Pass by Reference

```
function add($a, $b) {
```

Pass by Value

Exercise 6

- จงทำการปรับปรุงโปรแกรม BMI โดยให้ย้ายขั้นตอนการคำนวณ BMI เข้าไปอยู่ในฟังก์ชันที่ชื่อ `compute_bmi()`
- ตัวอย่าง

```
<?php
```

```
...
```

```
$bmi = compute_bmi(x,y,z)
```

```
...
```

```
?>
```

PHP - STRING

PHP: String (Recap.)

- ในภาษา php ไม่จำเป็นต้องมีการประกาศตัวแปร (variable declarations) ก่อนนำมาใช้งาน
 - ชนิดของ variable (data type) ขึ้นอยู่กับข้อมูลที่เก็บใน variable
 - เราสามารถกำหนดค่าให้กับตัวแปรแบบ String ดังนี้
- ```
$stringVar = "There are 8 data types in PHP"; // String
```

Note: string can be as large as up to 2GB (2147483647 bytes maximum)

# PHP: Construct a String

- Double quoted “ ”
  - `$stringA = “Construct String using double quoted”;`
- Single quoted ‘ ’
  - `$stringB = ‘Construct String using single quoted’;`
- Heredoc Syntax <<<
  - `$stringC = <<< end`  
`Construct String using Heredoc Syntax`  
`end;`

# PHP: Single vs Double Quotes

- ในการใช้ single quote ในการสร้าง string นั้น, อักขระพิเศษหรือ variable ที่อยู่ระหว่างเครื่องหมาย ' ' จะไม่ได้รับการประเมินค่า แต่จะเป็นการ assign ค่าไปยัง string นั้นตามที่พิมพ์ แต่หากเป็น string ที่สร้างด้วย double quote " " จะมีการประเมินค่าก่อนแล้วจึง assign ค่า

```
$myName = 'John';
echo 'Hello, there. \tMy name is $myName'
```

```
Hello, there. \tMy name is $myName
```

```
echo "Hello, there. \tMy name is $myName"
```

```
Hello, there. My name is John
```

tab



# PHP: String Operator

- Concatenation operator (.)

```
<?php
$Destination = "Paris";
$Location = "France";
$Destination = $Destination . " is in " . $Location .
 ". ";
echo $Destination;
?>
```

- Result: Paris is in France.

# PHP String: Escape Character

- ในกรณีที่ต้องการใส่อักขระพิเศษในสตริงที่ต้องการ สามารถใช้ `\` ในการกำหนดตัวอักขระที่ต้องการได้ เช่น

```
echo "\“Apple\” is a healthy fruit.";
```

Result → “Apple” is a healthy fruit.

| Escape Sequence                  | Description                                                                     |
|----------------------------------|---------------------------------------------------------------------------------|
| <code>\\</code>                  | Inserts a backslash                                                             |
| <code>\\$</code>                 | Inserts a dollar sign                                                           |
| <code>\r</code>                  | Inserts a carriage return                                                       |
| <code>\"</code>                  | Inserts a double quotation mark                                                 |
| <code>\t</code>                  | Inserts a horizontal tab                                                        |
| <code>\n</code>                  | Inserts a new line                                                              |
| <code>\regular expression</code> | Inserts a character in hexadecimal notation that matches the regular expression |

# PHP: String Function

- เราสามารถจัดการกับ String ได้โดยใช้ฟังก์ชันของ PHP
- String Function ช่วยให้การแสดงผลในหน้า HTML ทำได้ง่ายขึ้น
- ตัวอย่างประเภทของ String Function

|                                         |                                                   |
|-----------------------------------------|---------------------------------------------------|
| <a href="#"><u>substr()</u></a>         | แสดง String ตามช่วงความยาวที่ต้องการ              |
| <a href="#"><u>substr_compare()</u></a> | เปรียบเทียบส่วนของ String                         |
| <a href="#"><u>substr_count()</u></a>   | นับจำนวน String ที่ปรากฏในเอกสาร                  |
| <a href="#"><u>substr_replace()</u></a> | แทนค่า String ที่เจอด้วยคำที่ต้องการ              |
| <a href="#"><u>trim()</u></a>           | ตัดส่วนที่เป็น Whitespace ทั้งด้านหน้าและด้านหลัง |

# PHP: String Function

- สำหรับฟังก์ชันที่มักใช้งานกับ String สามารถแยกเป็นกลุ่มๆ ได้ดังนี้
  - ฟังก์ชันการแสดงผล เช่น `echo()`; `print()`;
  - เปรียบเทียบหรือตรวจสอบค่า เช่น `substr_compare()`; `isset()`; `isstring()`;
  - ตัดหรือต่อสตริง เช่น `substr()`;
  - ค้นหาและแก้ไขสตริง เช่น `strpos()`; `substr_replace()`;
  - รูปแบบการแสดงผล เช่น `strtoupper()`; `strtolower()`;

# PHP Function: substr();

- substr();
  - Syntax : substr(\$mainstring, start, length);
  - Samplecode



```
<?php
$stringA = "There are 8 data types in PHP";
$substringA = substr($stringA,6,3);
Echo $substringA
?>
```

- Result: are

# PHP Function: strpos();

- ใช้ค้นหาตัวอักษร หรือสตริงที่ต้องการในสตริงหลัก
- Syntax : strpos(\$stringA, "a");
- Return Value: ค่าตำแหน่งของอักขระที่ต้องการ
- หากไม่เจอ จะ return เป็น False

```
$Email = "president@whitehouse.gov";
```

```
$NameEnd = strpos($Email, "@"); //หาตำแหน่ง @
```

```
echo "<p>The name portion of the e-mail address is ' " .
```

```
substr($Email, 0, $NameEnd) . "'.</p>"; //ใช้ตำแหน่ง @ มาใช้ในการ
ตัดสตริง
```

- Result: The name portion of the e-mail address is **president**

# Exercise 7: Function & String

- จงออกแบบโปรแกรมเพื่อรับ email address เป็นข้อมูลเข้าแล้วตรวจสอบว่า email นั้น เป็น email ของ google หรือไม่ แล้วแสดงผลว่า valid หรือ invalid ทางหน้าเว็บ
- กำหนดให้ email ที่ valid มีลักษณะดังต่อไปนี้
  - เป็น email ที่มีเครื่องหมาย @ ตัวเดียว
  - เครื่องหมาย @ ต้องไม่ใช่ตัวแรกในชื่อ email
  - หลังจากเครื่องหมาย @ จะต้องตามด้วย google.com
  - คำว่า google.com เป็นแบบ case insensitive นั่นคือ GOOGLE.COM หรือ gOOgLE.com หรือ Google.com ถือว่าเป็น email ที่ valid

# PHP-ARRAY



# PHP: Array (Recap.)

- Array คือ ชุดของข้อมูลที่มีการเรียงกันอย่างเป็นลำดับ
- ข้อมูลที่อยู่ใน Array จะเรียกว่า Element
- การอ้างอิงถึง Element จะมีการระบุ Key ไว้ในตัวแปร Array

```
<?php
```

```
 $arrayA = (“one”, “two”, “three”);
```

```
 echo $arrayA[‘2’];
```

```
?>
```

Result: two

# PHP: Construct an Array

- กำหนดค่าให้ตัวแปรแบบไม่มี Key

```
$name[] = "Andrew";
$name[] = "Steve";
$name[] = "Charlie";
```

- ใช้คำสั่ง array();

```
$name = array("Andrew", "Steve",
"Charlie");
```

| Key | Value   |
|-----|---------|
| 1   | Andrew  |
| 2   | Steve   |
| 3   | Charlie |

- กำหนดค่าให้ตัวแปรแบบมี Key

```
$name["A"] = "Andrew";
$name["B"] = "Steve";
$name["C"] = "Charlie";
```

- ใช้คำสั่ง array();

```
$name = array("A" => "Andrew",
"B" => "Steve", C => "Charlie");
```

| Key | Value   |
|-----|---------|
| A   | Andrew  |
| B   | Steve   |
| C   | Charlie |

# PHP: Array Function

- Array Function เป็นกลุ่มฟังก์ชันที่จะทำงานเพื่อจัดการข้อมูล (Element) ที่อยู่ในตัวแปร Array
  - วนลูปเพื่อจัดการค่าใน Array (foreach, each)
  - แปลงค่าในอาเรย์ให้เป็นสตริง (implode)
  - การรวมอาเรย์ (merge and combine)
  - เรียงข้อมูลในอาเรย์ (sort)

# PHP Array: foreach (Recap.)

- Sourcecode

```
<?php
$colors =
array("red", "green", "blue", "y
ellow");
foreach ($colors as $value)
{
 echo "$value
";
}
?>
```

- Result

```
red
green
blue
yellow
```

# PHP Array: foreach (Recap.)

- Sourcecode

```
<?php
```

```
$age = array(
 "Peter"=>"35",
 "Ben"=>"37",
 "Joe"=>"43");
```

```
foreach($age as $x=>$x_value){
 echo "Key=".$x.", Value=".$x_value;
 echo "
";
}
```

```
?>
```

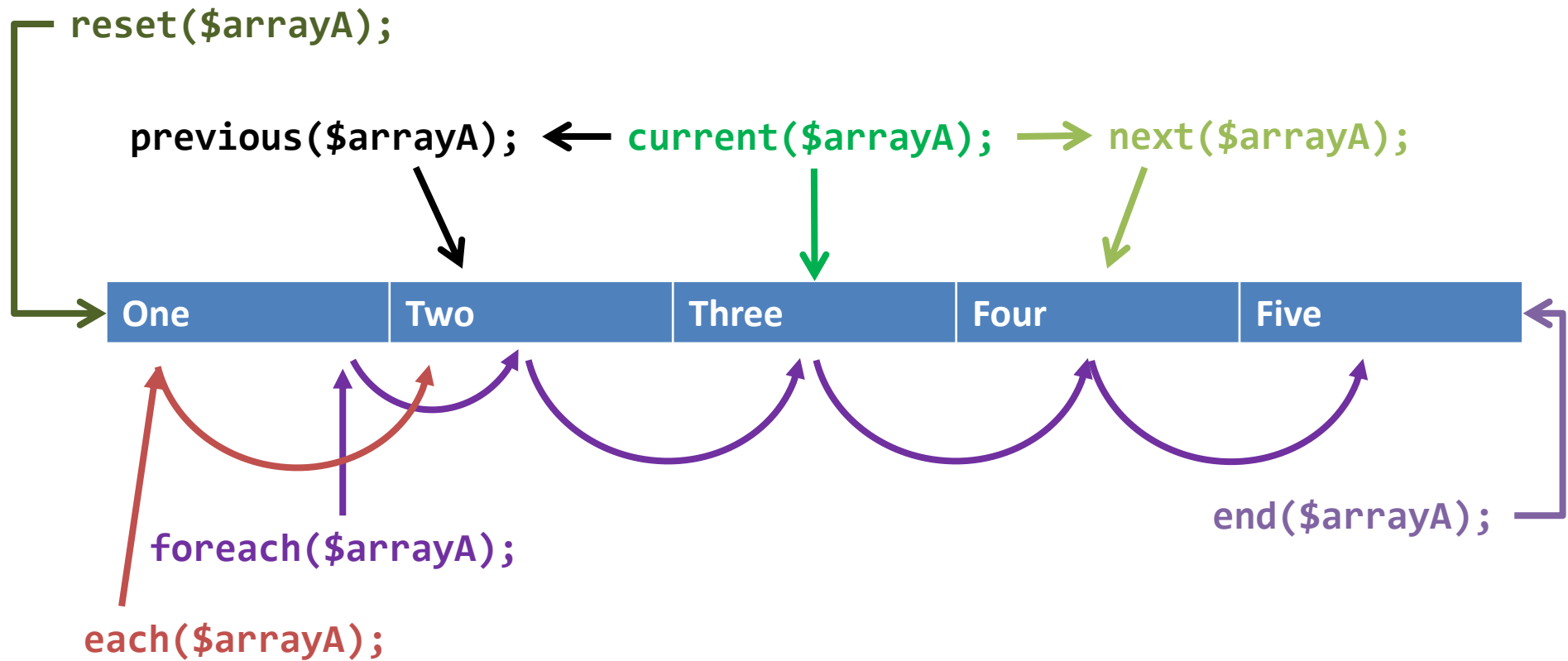
- Result

Key=Peter, Value=35

Key=Ben, Value=37

Key=Joe, Value=43

# Array Traversal



# PHP Array: loop with each();

```
<?php
$colors = array("red","green","blue","yellow");
echo " foreach" ;
foreach ($colors as $key =>$value){
 print $key . " : ".$value ."
\n";
}
```

foreach

0 : red

1 : green

2 : blue

3 : yellow

```
reset($colors);
echo " while..each" ;
while($val = each($colors)){
 print $val["key"]. " : ".$val["value"]."
\n";
}
?>
```

while..each

0 : red

1 : green

2 : blue

3 : yellow

# Array to String: implode();

- implode() เป็นการรวม Elements ใน Array ให้เป็น String
- Syntax : implode(string glue, array pieces);

```
<?php
```

```
 $arrayWord = ("This", "is", "a", "cat");
 $sentence=implode(" ", $arrayWord);
 echo $sentence;
```

```
?>
```

Result: This is a cat



# PHP Array: array\_unique();

- เป็นฟังก์ชันในการจัดการข้อมูลซ้ำที่อยู่ในอาเรย์
- Example

```
<?php
```

```
$colors =
array("red", "green", "blue", "red", "green", "yellow");
$COLOR = array_unique($colors);
```

```
?>
```

| \$colors | \$COLOR |
|----------|---------|
| red      | red     |
| green    | green   |
| blue     | blue    |
| red      | yellow  |
| green    |         |
| yellow   |         |

# PHP:Array merge

- เป็นฟังก์ชันการรวม element ในหลาย Array ให้เป็น Array เดียวกัน
- รูปแบบการใช้งาน
  - `$arrayA+ $arrayB***`
  - `array_merge($arrayA , $arrayB);***`
- \*\*\* สองกรณีนี้จะให้ผลต่างกันกรณีที่ Array ที่นำมารวมกันมีการเซตค่า key ซ้ำกัน

# PHP:Array merge

```
<?php
```

```
$arrayA = array("a" => "apple", "b" => "bee", "c" => "cat");
```

```
$arrayB = array("b" => "boy", "c" => "cat", "d" => "dog");
```

```
$arrayResult = $arrayA+ $arrayB;
```

```
$arrayMerge = array_merge($arrayA, $arrayB);
```

```
echo "+
";
```

```
foreach ($arrayResult as $key => $a){
```

```
print "[$key] : $a
\n";}
```

```
echo "merge
";
```

```
foreach ($arrayMerge as $key => $a){
```

```
print "[$key] : $a
\n";
```

```
}
```

```
?>
```

```
+
```

```
[a] : apple
```

```
[b] : bee
```

```
[c] : cat
```

```
[d] : dog
```

```
merge
```

```
[a] : apple
```

```
[b] : boy
```

```
[c] : cat
```

```
[d] : dog
```

# PHP Array: array\_combine();

- เป็นฟังก์ชันใช้เพื่อรวมอาเรย์ 2 ชุด ในลักษณะ key->value

- Syntax:

```
array array_combine (array $keys , array $values);
```

```
<$php
```

```
$country = array (“Thailand”, “Japan”, “Taiwan”);
```

```
$capital = array (“Bangkok”, “Tokyo”, “Taipei”);
```

```
$ListA = array_combine($country, $capital);
```

```
?>
```

| \$ListA  |         |
|----------|---------|
| Key      | Value   |
| Thailand | Bangkok |
| Japan    | Tokyo   |
| Taiwan   | Taipei  |

# PHP Array: Sorting Function

| Function | Meaning                                     |
|----------|---------------------------------------------|
| sort()   | จัดเรียง Value และ Key ในอาเรย์จากน้อยไปมาก |
| rsort()  | จัดเรียง Value และ Key ในอาเรย์จากมากไปน้อย |
| asort()  | จัดเรียง Value ในอาเรย์จากน้อยไปมาก         |
| arsort() | จัดเรียง Value ในอาเรย์จากมากไปน้อย         |
| ksort()  | จัดเรียง Key ในอาเรย์จากน้อยไปมาก           |
| krsort() | จัดเรียง Key ในอาเรย์จากมากไปน้อย           |

# Exercise 8: PHP Array and control statement

- สร้างฟอร์มเพื่อรับจำนวนตัวเลข  $n$  เข้ามา
- เมื่อ user ทำการ submit ให้ประมวลผลโดยการ สร้างฟอร์มอีก  
หนึ่งอันซึ่งมีจำนวนช่องให้ใส่ตัวเลข  $n$  ช่อง
- เมื่อ user ใส่ตัวเลขครบทั้ง  $n$  ตัวและทำการ submit แล้ว
- ให้ประมวลผลโดยการหา ค่า max ค่า min และค่าเฉลี่ยของ  
ตัวเลขเหล่านั้น

# PART II

# PHP-SESSION



# Information Flow

- การส่งข้อมูลระหว่าง webpage จะต้องอาศัย form เป็นตัวกลางในการส่งข้อมูล ซึ่งหากเราต้องการใช้ข้อมูลใน webpage หลายๆหน้า เราต้องสร้าง form ทุกครั้ง อีกทั้งไม่สามารถใช้ `<a></a>` ในการลิงค์ไปหน้าอื่น ๆ ได้ (เนื่องจากข้อมูลจะไม่ถูกส่งต่ออย่างต่อเนื่อง)
- สำหรับ web application เราสามารถจัดการกับปัญหานี้ได้โดยการให้ Web Browser จำข้อมูลเหล่านั้น ซึ่งสามารถทำได้สองวิธีคือ Cookies (Java script) และ Sessions (PHP)

# Memorization

- การจดจำข้อมูลในรูปแบบของตัวแปร เราจะใช้ระบบ Sessions ที่ PHP สร้างขึ้น
- ผู้เข้าชมเว็บจะได้รับ Session ID หรือมักเรียกว่า Unique ID (UID) ซึ่งจะถูกรับที่กึ่งกลางใน Cookie หรือส่งต่อผ่าน URL
- PHP Sessions อนุญาตให้เราบันทึกข้อมูลลงใน Superglobal Array ที่ชื่อว่า `$_SESSION`

# Sessions preprocessing

- ก่อนที่เราจะส่งให้ PHP Sessions จัดจำข้อมูลของเรา เราจะต้องทำการเรียกใช้ PHP Sessions ก่อน ด้วยคำสั่ง
  - `session_start();`
- คำสั่งข้างต้นเป็นการสั่งให้เว็บของเราเริ่มการทำงานของ PHP Sessions ซึ่งจะต้องประกาศไว้ในส่วนแรกสุดของทุกไฟล์ PHP ที่ต้องการใช้หรือบันทึกข้อมูลใน session

# Information Storing

- หลังจากเราเรียกใช้งาน PHP Sessions เราสามารถบันทึกข้อมูลลงใน Superglobal array `$_SESSION` ดังตัวอย่างด้านล่าง

```
<?php
```

```
session_start(); // เรียกใช้ PHP Session
```

```
$_SESSION['username'] = "admin"
```

```
$_SESSION['age'] = 20;
```

```
?>
```

# Information retrieval

- หลังจากที่เราบันทึกข้อมูลลงใน superglobal array `$_SESSION` แล้วนั้น เราสามารถดึงข้อมูลมาใช้ได้โดยไม่ต้องส่งข้อมูลผ่าน Form อีก

```
File1.php
<?php
session_start();

$_SESSION['username'] = "James";
$_SESSION['age'] = 25

// บันทึกข้อมูลใน Sessions

?>
```

```
File2.php
<?php
session_start();

echo $_SESSION['username'];
echo $_SESSION['age'];

// เอาข้อมูลที่บันทึกมาใช้

?>
```

# Existence Validation

- ข้อควรระวังในการเรียกใช้ Superglobal Array `$_SESSION` คือ การเรียกใช้ข้อมูลที่ยังไม่เคยมีการบันทึกมาก่อน
- หากเป็นเช่นนั้นจะเกิด **Error Undefined Index**
- เพื่อเลี่ยงปัญหานี้เราสามารถตรวจสอบว่าตัวแปรนั้นถูกบันทึกไว้หรือไม่ด้วยฟังก์ชัน `isset`

```
if(isset($_SESSION['username']))
 echo $_SESSION['username'];
```

# Terminating sessions

- หากเราต้องการทำลายข้อมูลที่บันทึกไว้ เราสามารถทำได้ด้วยคำสั่ง
  - `session_unset();` // ลบตัวแปรทั้งหมด
  - `session_destroy();` // จบการทำงานของ sessions

# Login system

- ระบบ Login เป็นระบบที่ทำการจดจำการลงชื่อเข้าใช้ โดยระบบจะต้องจดจำข้อมูลสำคัญสำหรับยืนยันตัวตนเช่น username, email และไม่ควรจดจำ password
- ระบบ Login จะประกอบไปด้วยสองส่วนหลัก คือ
  - Login.php ทำหน้าที่ตรวจสอบว่า username และ password ได้ผ่านการลงทะเบียนหรือไม่ หากใช่ ให้บันทึก username ลงใน session
  - Logout.php ทำหน้าที่ลบข้อมูลการ login ออกจากระบบ



# Login system [2]

## Homepage.php

ถ้ายังไม่ Login ให้แสดงฟอร์มเพื่อ Login

```
<form method=POST action=login.php>
```

...

```
</form>
```

ถ้า Login แล้ว ให้แสดงข้อความต้อนรับ และ  
ลิงค์เพื่อ Logout

```
echo "Hello, $_SESSION['username']";
```

```
echo " logout ";
```

## Login.php

- ❖ ตรวจสอบว่ามีการ Login แล้วหรือไม่
- ❖ ตรวจสอบว่ากรอก Username Password ถูกต้องหรือไม่ (ตรวจว่า Username นี้ได้ทำการลงทะเบียนหรือไม่)
- ❖ สร้าง Session เพื่อบันทึก Username

## Logout.php

- ❖ ทำลาย Session

# Exercise 9: Login system

- จงออกแบบระบบ login ซึ่งประกอบด้วยไฟล์ 3 ไฟล์
  - Homepage.php
  - Login.php
  - Logout.php
- Homepage คือหน้าหลัก ที่มีฟอร์มไว้สำหรับ login
- เมื่อ user ทำการ login ข้อมูลในฟอร์มจะถูกส่งไปจัดการที่หน้า login.php
- หน้า login.php ทำหน้าที่เช็คค่า username และ password ถูกต้องหรือไม่ (ในที่นี้ให้เก็บ username กับ password ในรูปของ string ก่อน) แล้วให้ redirect กลับมาหน้า homepage
- หาก user อยู่ในสถานะ login ให้แสดงคำพูดว่า Hello ... ตามด้วย username และ เปลี่ยนจากโชว์ฟอร์ม login เป็น ปุ่ม logout

# PHP - MYSQL

# การดึงข้อมูลจากฐานข้อมูลผ่าน PHP

- มีสามขั้นตอน
  - ขั้นตอนติดต่อกับฐานข้อมูล
  - ขั้นตอนดึงข้อมูล
  - ขั้นตอนตัดการเชื่อมต่อกับฐานข้อมูล

# ขั้นตอนแรก Connect

- การเริ่มการเชื่อมต่อกับฐานข้อมูล
- ทำได้โดยเรียกใช้ฟังก์ชัน `mysqli_connect()`
- มี `syntax` ในการเรียกใช้ดังนี้
- `mysqli_connect($host, $username, $password, $database_name)`
- ถ้าเชื่อมต่อสำเร็จ `mysqli_connect()` จะรีเทิร์น `object` ที่แสดงการเชื่อมต่อ หากไม่สำเร็จจะรีเทิร์น `FALSE`

# ตัวอย่าง

```
<?php
```

```
 $servername = "localhost";
```

```
 $username = "username";
```

```
 $password = "password";
```

```
 $dbname = "myDB";
```

```
 // Create connection
```

```
 $conn = mysqli_connect($servername, $username, $password,
$dbname);
```

```
 // Check connection
```

```
 if (!$conn) {
```

```
 die("Connection failed: " . mysqli_connect_error());
```

```
 } else {
```

```
 echo "Success";
```

```
 }
```

```
?>
```

# ขั้นตอนที่สอง Query

- สามารถทำได้โดยใช้ฟังก์ชัน `mysqli_query()`
- ซึ่งมี syntax ดังนี้
- `mysqli_query($con, $sql_statement)`
- `$con` คือสิ่งที่ได้จากการเรียก `mysqli_connect()` ที่สำเร็จ
- `$sql_statement` คือ string แสดงคำสั่งภาษา SQL ที่เราต้องการ
- `mysqli_query()` จะรีเทิร์นผลลัพธ์ในรูปแบบของ `mysqli_result` ซึ่งเรียกดูผลยาก หรือรีเทิร์น `FALSE` ถ้า query ไม่สำเร็จ
- แต่สามารถแปลง `mysqli_result` เป็น php array ได้ด้วยฟังก์ชัน `mysqli_fetch_assoc()`

# ตัวอย่าง

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
 // output data of each row
 while($row = mysqli_fetch_assoc($result)) {
 echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "
";
 }
} else {
 echo "0 results";
}
```



# ขั้นตอนสุดท้าย Close

- เป็นการปิดการเชื่อมต่อกับฐานข้อมูลเมื่อเราใช้งานเสร็จแล้ว
- สามารถทำได้โดยใช้ฟังก์ชัน `mysqli_close()`
- ตัวอย่าง
  - `mysqli_close($conn);`

# SQL Wildcard

- มีไว้เพื่อเติมอักขระอื่น ๆ ลงไปในข้อความ (string)
- % เติมอักขระอะไรก็ได้ตั้งแต่ 0 ตัวขึ้นไป
- \_ เติมอักขระอะไรก็ได้ตัวเดียว
- มีประโยชน์ในการเทียบหา substring ใน database
- ตัวอย่าง
  - %ber% จะเทียบตรงกับ number , berlin, uber
  - \_ber จะเทียบตรงกับ uber

# การใช้ Wildcard ในคำสั่ง Select

```
SELECT * FROM Customers
```

```
WHERE City LIKE 'ber%';
```

เลือกทุกคอลัมน์จากรายการ customers ที่ชื่อเมืองขึ้นต้นด้วย ber

# ORDER BY

- แสดงผลลัพธ์ที่ได้เรียงตามลำดับ
- `SELECT` column\_name(s)
- `FROM` table\_name
- `ORDER BY` column\_name(s) `ASC|DESC`;
  
- ตัวอย่าง
- `SELECT` student\_id, gpa
- `FROM` students
- `WHERE` gpa > 2.0 `AND` gpa < 3.0
- `ORDER BY` gpa `DESC`;

# Exercise 10 (mini boss)

- จงปรับปรุงโปรแกรมใน exercise 9 ให้ตรวจสอบ username และ password ที่ดึงมาจากฐานข้อมูลแทน

# PHP MYSQL II

# Database Modification

- คำสั่งปรับปรุงเปลี่ยนแปลงฐานข้อมูล ไม่ส่งผลลัพธ์กลับมา เหมือนคำสั่ง SELECT
- คำสั่งที่เกี่ยวข้องกับการปรับปรุงฐานข้อมูล
  1. INSERT
  2. DELETE
  3. UPDATE

# Insertion

- Syntax

```
INSERT INTO table
```

```
VALUES ('value1', 'value2', 'value3',...);
```

- ตัวอย่าง

```
INSERT INTO Likes
```

```
VALUES('Sally', 'Singha');
```



# DELETION

- Syntax

DELETE FROM table

WHERE condition;

- ตัวอย่าง

DELETE from Likes

WHERE drinker='Sally';

DELETE from Likes

WHERE drinker='Sally' AND beer='Singha';

- DELETE FROM table;
- ไม่ต้องมี WHERE clause
- ตัวอย่าง  
DELETE FROM Likes

# UPDATES

- ใช้ในการเปลี่ยน attribute บางตัวของ record ใน table

- Syntax

UPDATE table

SET <list of attribute assignments>

WHERE condition;

- ตัวอย่าง

UPDATE Drinkers

SET phone='0818889999'

WHERE name='Fred'

# Exercise 11

- จงปรับปรุงโปรแกรม login ข้างต้น ให้มีหน้าและฟังก์ชันต่อไปนี้
  - register.php เพื่อให้ผู้ใช้สามารถลงทะเบียนใหม่ได้
  - edit.php เพื่อให้ผู้ใช้สามารถเข้ามาแก้ไขข้อมูลส่วนตัวได้

ปรับปรุงโดย อ. กิตติพิชญ์ คุปตะวาณิช  
และ อ. สิทธิโชค ทรัพย์ไพบุลย์กิจ

## **PART III: HTML AND CSS**

# HTML

## คำสั่งในการจัดตัวอักษร

# HTML Text Formatting

<p>

<b>This text is bold</b>

**This text is bold**

<strong>This text is strong</strong>

**This text is strong**

<i>This text is italic</i>

*This text is italic*

<em>This text is emphasized</em>

*This text is emphasized*

<code>This is computer output</code>

This is computer output

This is <sub>subscript</sub>

This is subscript

This is <sup>superscript</sup>

This is superscript

</p>

# HTML Formatting

- More HTML Text Formatting Tags

Tag	Description
<a href="#"><u>&lt;small&gt;</u></a>	สำหรับ text ขนาดเล็ก (smaller)
<a href="#"><u>&lt;ins&gt;</u></a>	สำหรับ text ที่แทรกมา (inserted)
<a href="#"><u>&lt;del&gt;</u></a>	สำหรับ text ที่ถูกลบ (deleted)
<a href="#"><u>&lt;mark&gt;</u></a>	สำหรับ text ที่ถูก highlight (marked)



# HTML Formatting [2]

`<p>`

This is `<small>small</small>` text.  
`<br>`

This is small text.

This is `<ins>inserted</ins>` text.  
`<br>`

This is inserted text.

This is `<del>deleted</del>` text.  
`<br>`

This is ~~deleted~~ text.

This is `<mark>marked</mark>` text.

This is **marked** text.

`</p>`

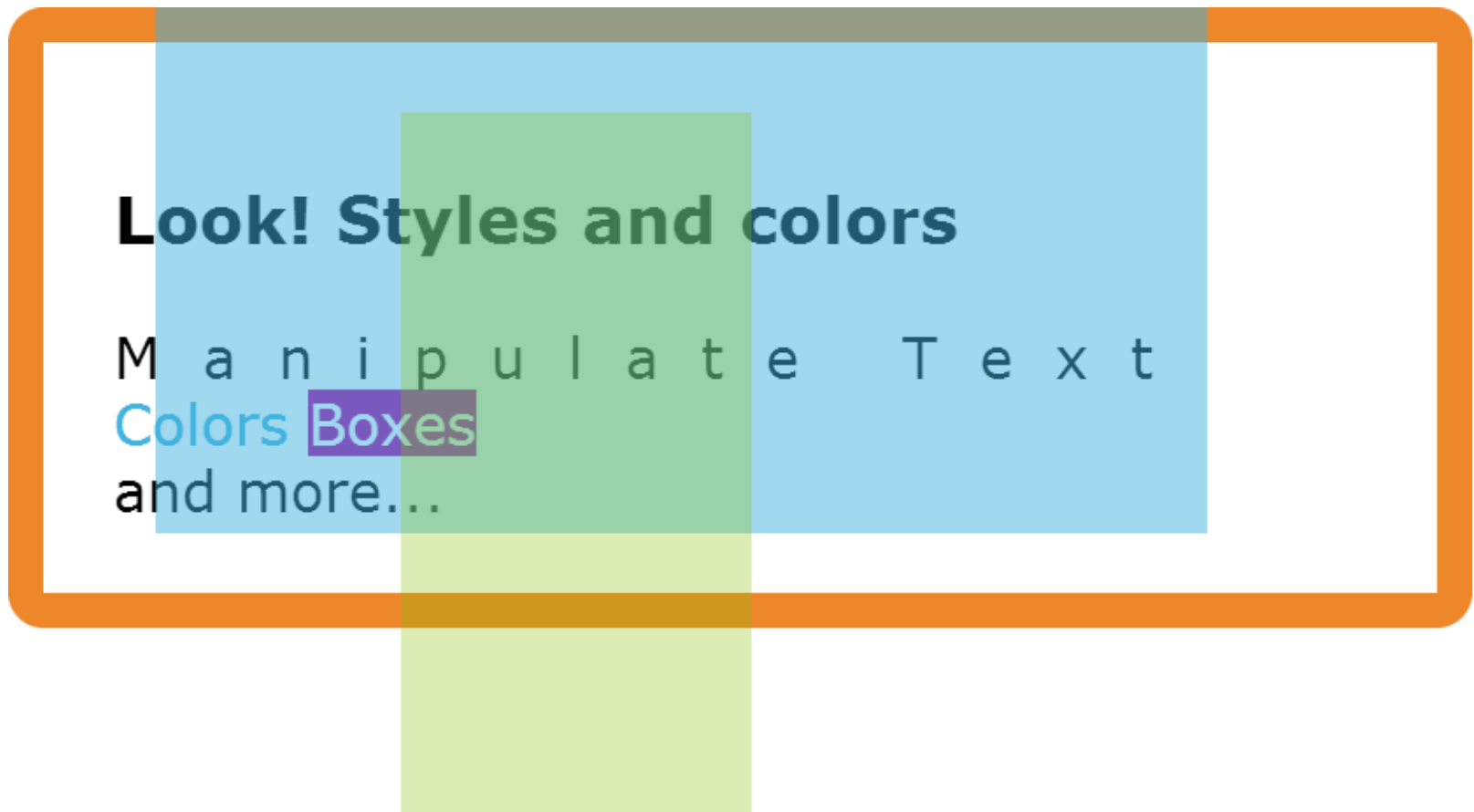
- What is CSS?
  - CSS ย่อมาจาก Cascading **Style** Sheet
  - **Style** ใช้เพื่อกำหนดรูปแบบการแสดงผล HTML element
  - CSS ถูกสร้างมาเพื่อแยกรูปแบบการแสดงผล (**Style**) ออกจากเนื้อหา (**Content**)

# HTML Styles – CSS [2]

- CSS Example

```
<head>
<style>
p {
 text-align: center;
 color: red;
}
</style>
</head>
```

## □ CSS Example - Result



# HTML - Adding CSS

- การกำหนด Style สามารถทำได้ 3 วิธีคือ
  1. กำหนดแบบ Inline โดยระบุเป็น Attribute "style" ในแต่ละ Element

```
<p style="color:blue;margin-left:20px;">Some text</p>
```

# HTML - Adding CSS

- กำหนดแบบ Internal Style Sheet โดยการใช้ Tag `<style>` ในส่วน `<head>` ของเอกสาร HTML

```
<head>
 <style type="text/css">
 body {background-color:yellow;}
 p {color:blue;}
 </style>
</head>
```

# HTML - Adding CSS

- กำหนดแบบ External Style Sheet ลงในไฟล์เอกสาร CSS แยกต่างหาก (CSS File) โดยการใช้ Tag `<link>` ซึ่งไฟล์ CSS หนึ่งไฟล์สามารถใช้ร่วมกันในเอกสาร HTML หลายๆ เอกสารได้

```
<head>
 <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

# HTML Style Example

## ● Example 1: Background Color

- background-color กำหนดสีของพื้นหลัง ซึ่งสามารถกำหนดสีโดยใช้
  - ชื่อสี (Color Names)
    - เช่น Black, Red, LightSkyBlue - ชื่อสีจะไม่มี space (วรรค)
    - สามารถดู list ชื่อสีเพิ่มเติมได้ที่
      - » [http://www.w3schools.com/html/html\\_colornames.asp](http://www.w3schools.com/html/html_colornames.asp)
  - ค่า RGB (Red Green Blue) โดยใช้รูปแบบเลขฐาน 16 #xxxxxx หรือ rgb(r,g,b)
    - เช่น #000000 = Black, rgb(255,0,0) = Red, #87CEFA = LightSkyBlue
    - สามารถดู list ค่า RGB เพิ่มเติมได้ที่
      - » [http://www.w3schools.com/html/html\\_colorvalues.asp](http://www.w3schools.com/html/html_colorvalues.asp)



# HTML Style Example [2]

- Example 1: Background Color [2]

```
<!DOCTYPE html>
<html>
 <body>
 <h1 style="background-color:green;">A heading</h1>
 <p style="background-color:chocolate">
 A paragraph.
 </p>
 </body>
</html>
```

# HTML Style Example [3]

- Example 2: Font, Color and Size

- font-family กำหนดชื่อ font
- color กำหนดสีของตัวอักษร
- size กำหนดขนาดของตัวอักษร

```
<!DOCTYPE html>
<html>
 <body>
 <h1 style="font-family:verdana;">A heading</h1>
 <p style="font-family:arial;color:red;font-size:20px;">
 A paragraph.
 </p>
 </body>
</html>
```

# HTML Style Example [4]

- Example 3: Border (เส้นขอบ)
  - border-style กำหนดแบบเส้นขอบ
    - none คือไม่มีขอบ

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders

groove: Defines a 3D grooved border

ridge: Defines a 3D ridged border

inset: Defines a 3D inset border

outset: Defines a 3D outset border

# HTML Style Example [5]

- Example 3: Border (เส้นขอบ) [2]
  - border-width กำหนดความหนาของเส้นขอบ
  - border-color กำหนดสีของเส้นขอบ

```
<!DOCTYPE html>
<html>
 <body>
 <p style="border-width:5px;border-style:solid;border-color:red;">
 A paragraph.
 </p>
 </body>
</html>
```

# HTML Style Example [6]

## ● Example 3: Border (เส้นขอบ) [3]

- เราสามารถกำหนดเส้นขอบอย่างย่อได้โดยใช้ลำดับ: ความหนา, ชนิดของขอบ และสี

```
<p style="border-width:5px;border-style:solid;border-color:red;">
```

```
<p style="border:5px solid red;">
```

- border-radius กำหนดเส้นขอบแบบมุมโค้ง

- ใช้ได้กับ IE9+, Firefox 4+, Chrome, Safari 5+, และ Opera.

```
<p style="border: 5px solid orange;border-radius:7px">
```

round border

# HTML การกำหนดเครื่องหมายหน้าหัวข้อ และเลขลำดับรายการ

# HTML Lists

- การทำ List (รายการ) ในเอกสาร HTML ทำให้ข้อมูลในเว็บเพจอ่านได้ง่าย และเป็นการจัดระเบียบเนื้อหาความ โดยใช้การแสดงเครื่องหมายหรือเลขลำดับรายการหน้าหัวข้อ
- List ในเอกสาร HTML ที่ใช้มากที่สุดคือ
  - List แบบไม่เรียงลำดับ <UL> (Unordered Lists)
  - List แบบเรียงลำดับ <OL> (Ordered Lists)
- มาตรฐาน HTML ไม่อนุญาตให้ใส่ List Tag ภายใน Paragraph Tag <p>

# HTML Lists

## ● HTML List Tags

Tag	Description
<code>&lt;ol&gt;</code>	กำหนด List แบบมีลำดับ ( <b>O</b> rdered <b>L</b> ist)
<code>&lt;ul&gt;</code>	กำหนด List แบบไม่มีลำดับ ( <b>U</b> nordered <b>L</b> ist)
<code>&lt;li&gt;</code>	กำหนดแต่ละรายการภายใน List ( <b>L</b> ist <b>I</b> tem)
<code>&lt;dl&gt;</code>	กำหนด List คำอธิบาย ( <b>D</b> escription <b>L</b> ist)
<code>&lt;dt&gt;</code>	กำหนดแต่ละรายการใน List คำอธิบาย ( <b>D</b> efinition <b>T</b> erm)
<code>&lt;dd&gt;</code>	กำหนดคำอธิบายของแต่ละรายการใน List ( <b>D</b> efinition <b>D</b> escription)



# HTML Unordered Lists

- เป็นการแสดงรายการกลุ่มที่เกี่ยวข้องกันแต่ไม่จำเป็นต้องเรียงลำดับ โดยแต่ละรายการจะมีหัวข้ออยู่ในรูปแบบวงกลมทึบ
- 

```
<p>Things to buy:</p>

 Coffee
 Milk

```



Things to buy:

- Coffee
- Milk

# HTML Ordered Lists

- เป็นการแสดงรายการที่มีลำดับก่อนหลัง โดยค่า Default จะเป็นตัวเลขเริ่มจากลำดับที่ 1.

```
<p>Things to buy:</p>

 Coffee
 Milk

```



Things to buy:

1. Coffee
2. Milk

# HTML Ordered Lists [2]

- Attribute ใน Ordered List

- `<ol>` Tag

- `start` ใช้กำหนดค่าเริ่มต้นของ list นั้นๆ
- `type` ใช้กำหนดชนิดของหัวข้อโดยค่าที่ใช้ได้คือ
  - 1, A, a, I, และ i

```
<p>Things to buy:</p>
<ol start="5" type="a">
 Coffee
 Tea
 Milk

```



Things to buy:

- e. Coffee
- f. Tea
- g. Milk

# HTML Description Lists

- Description List คือ List ที่มีคำอธิบายแต่ละรายการ

```
<dl>
 <dt>Coffee</dt>
 <dd>- black hot drink</dd>
 <dt>Milk</dt>
 <dd>- white cold drink</dd>
</dl>
```

Coffee

- black hot drink

Milk

- white cold drink

# HTML

## 4. การสร้างตาราง

รวบรวมโดย อ. กิตติพิชญ์ คุปตะวาณิช

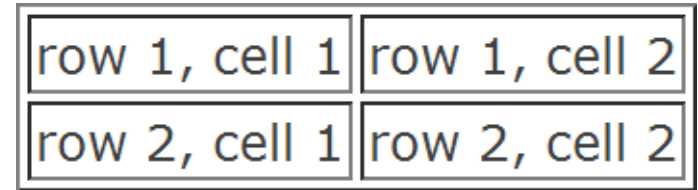
# HTML Tables

- การสร้างตารางในเอกสาร HTML ทำได้โดยการใช้ Tag `<table>`
- ตารางในเอกสาร HTML ประกอบด้วยแถวในแนวนอน (Row)
  - กำหนดด้วย Tag `<tr>` (tr: table row)
- ในแต่ละ Row จะประกอบด้วยช่องที่เรียกว่า Data Cell
  - กำหนดด้วย Tag `<td>` (td: table data)
  - ในแต่ละ Cell จะสามารถใส่ Content ได้หลายประเภท เช่น ข้อความ รูปภาพ ลิสต์ พารากราฟ ฟอรั่ม หรือตาราง

# HTML Table Example

```
<table border="1">
 <tr>
 <td>row 1, cell 1</td>
 <td>row 1, cell 2</td>
 </tr>
 <tr>
 <td>row 2, cell 1</td>
 <td>row 2, cell 2</td>
 </tr>
</table>
```

ใช้เครื่องหมาย "" และไม่มีหน่วย



row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

□ หากไม่มีการกำหนดค่าให้ Attribute "border" ตารางที่ได้จะไม่มีเส้นกรอบทั้งภายใน (ระหว่าง Cell) และภายนอกตาราง

□ ขนาดของ border ที่ระบุจะเป็นความกว้างของเส้นรอบนอกตารางเท่านั้น

# HTML Table Tags

- Tags ที่ใช้บ่อยในการสร้างตารางได้แก่

Tag	Description
<a href="#"><u>&lt;table&gt;</u></a>	กำหนดตาราง
<a href="#"><u>&lt;th&gt;</u></a>	กำหนดหัวตาราง ( <a href="#"><u>t</u>able <a href="#"><u>h</u>header</a>)</a>
<a href="#"><u>&lt;tr&gt;</u></a>	กำหนดแถว ( <a href="#"><u>t</u>able <a href="#"><u>r</u>ow</a>)</a>
<a href="#"><u>&lt;td&gt;</u></a>	กำหนดช่องในตาราง ( <a href="#"><u>t</u>able <a href="#"><u>d</u>ata</a>)</a>
<a href="#"><u>&lt;caption&gt;</u></a>	กำหนดชื่อเรื่องของตาราง



# HTML Table Tags [2]

- Tags อื่นๆ ในการสร้างตาราง

Tag	Description
<a href="#"><u>&lt;colgroup&gt;</u></a>	Specifies a group of one or more columns in a table for formatting
<a href="#"><u>&lt;col&gt;</u></a>	Specifies column properties for each column within a <a href="#"><u>&lt;colgroup&gt;</u></a> element
<a href="#"><u>&lt;thead&gt;</u></a>	Groups the <a href="#"><u>header</u></a> content in a table
<a href="#"><u>&lt;tbody&gt;</u></a>	Groups the <a href="#"><u>body</u></a> content in a table
<a href="#"><u>&lt;tfoot&gt;</u></a>	Groups the <a href="#"><u>footer</u></a> content in a table

# HTML Table Headers

- การกำหนดข้อมูลที่หัว Column (Header) ทำได้โดยการใช้ Tag <th>
- Browser ส่วนใหญ่จะแสดงผล Content ภายใต้ Tag <th> เป็นตัวหนาและอยู่ตรงกลาง

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

```
<table border="1">
 <tr>
 <th>Header 1</th>
 <th>Header 2</th>
 </tr>
 <tr>
 <td>row 1, cell 1</td>
 <td>row 1, cell 2</td>
 </tr>
 <tr>
 <td>row 2, cell 1</td>
 <td>row 2, cell 2</td>
 </tr>
</table>
```

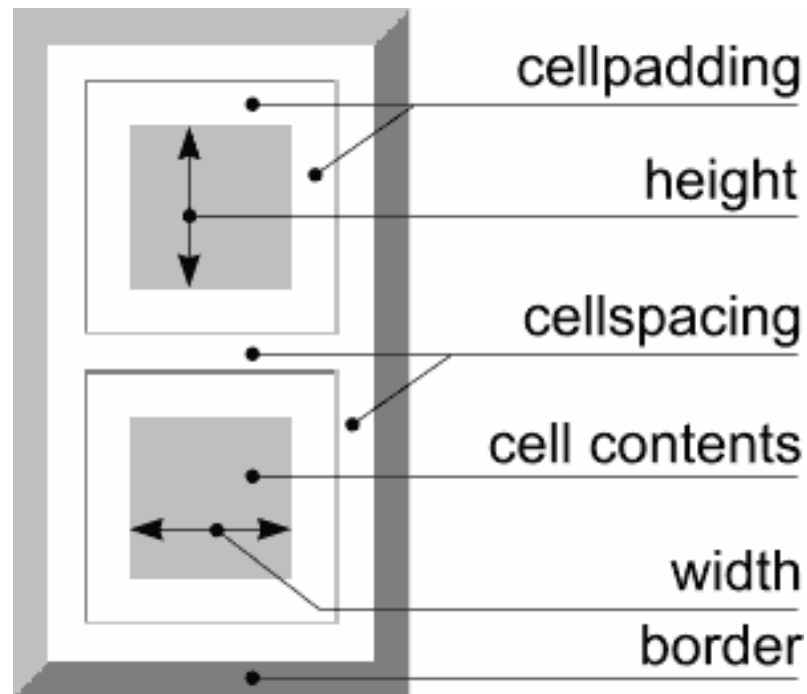
# HTML Table Attributes

- Attribute ต่างๆ ที่สามารถใช้กับ Tag <table> ได้แก่

Note: HTML5 จะ support แค่ border เท่านั้น

Attribute	Description
summary	สรุปเนื้อหาตาราง (ใช้สำหรับโปรแกรม Text-to-Speech กรณีผู้มีปัญหาทางการมองเห็น)
width	ความกว้างของตาราง
✓ border	ความกว้างของเส้นกรอบรอบนอกตาราง หากไม่ระบุจะไม่มีเส้นระหว่าง Cell
cellspacing	ระยะห่างระหว่าง Cell
cellpadding	ระยะห่างระหว่างตัวอักษรและขอบ Cell

# cellspacing and cellpadding



# HTML Table Formatting

- เช่นเดียวกับกับ Element อื่นๆ เราสามารถจัดรูปแบบ ตาราง หรือ cell ได้โดยใช้ Property ต่างๆ ใน style
- Example 1: alignment
  - margin-left กำหนดระยะห่างจากทางขอบซ้าย (default 0)
  - margin-right กำหนดระยะห่างจากทางขอบขวา (default 0)
    - auto - browser จะคำนวณระยะห่างเอง
    - inherit - ใช้ค่าเดียวกับ Element ที่เป็น Parent
    - ระยะห่าง (*length*)

- หน่วยที่สามารถใช้ได้ ใน CSS มีหลายแบบ
  - บางชนิดเป็นหน่วยที่ใช้ในงานด้านการออกแบบตัวอักษร (Typography) มาก่อน เช่น pc (pica) และ pt (point)
  - หรือหน่วยเป็นหน่วยความยาวพื้นฐานได้ เช่น cm, mm และ in (inch)
  - หน่วยที่สร้างขึ้นมาเพื่อใช้กับ CSS โดยเฉพาะคือ px (pixel)
  - หน่วย em และ ex เป็นหน่วยที่อ้างอิงจากความกว้างและความสูงของ Font ที่ใช้อยู่ ตามลำดับ
  - สามารถใช้หน่วยเป็น % อ้างอิงจาก Parent Element

# CSS Units [2]

- การเลือกใช้แต่ละหน่วยขึ้นอยู่กับวัตถุประสงค์ของเอกสารว่าเป็นเอกสารเพื่อการแสดงผลบนจอ (Screen) หรือการพิมพ์ (Print)

	<b>Recommended</b>	<b>Occasional use</b>	<b>Not recommended</b>
<b>Screen</b>	em, px, %	ex	pt, cm, mm, in, pc
<b>Print</b>	em, cm, mm, in, pt, pc, %	px, ex	

- ความสัมพันธ์ของหน่วยแบบ Absolute

$$1\text{in} = 2.54\text{cm} = 25.4\text{mm} = 72\text{pt} = 6\text{pc}$$

# HTML Table Formatting [2]

## ● Example 1: alignment [2]

- margin-left กำหนดระยะห่างจากทางขอบซ้าย (Default 0)
- margin-right กำหนดระยะห่างจากทางขอบขวา (Default 0)
  - หากต้องการให้ Table อยู่ตรงกลาง ทำได้โดยการกำหนด Property "margin-left" และ "margin-right" เป็น auto

```
<table border="1" style="margin-left:auto;margin-right:auto">
 <tr>
 <td>row 1, cell 1</td>
 <td>row 1, cell 2</td>
 </tr>
</table>
```



# HTML Table Formatting [2]

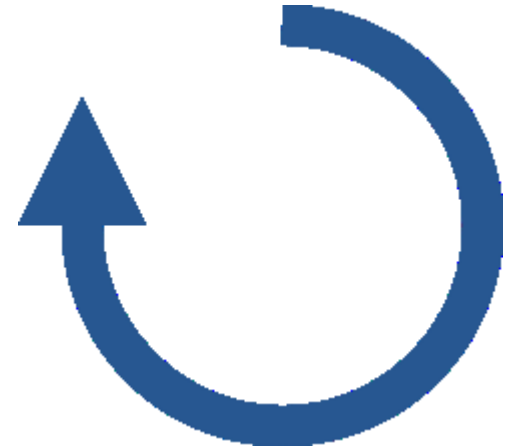
- The margin property

- ในทำนองเดียวกัน เราสามารถกำหนด margin-top และ margin-bottom ให้กับ element ได้
- โดยสามารถเขียนในรูปย่อได้ด้วยการใช้ Property "margin"

```
margin-left: 20px;
margin-right: 10px;
margin-top: 5px;
margin-bottom: 15px;
```



```
margin: 5px 10px 15px 20px;
```



- การระบุค่าจะเริ่มจากด้านบน (top) ก่อน แล้ววนตามเข็มนาฬิกา (top right bottom left)

# HTML Table Formatting [3]

- สามารถกำหนดค่า ให้ margin ได้ตั้งแต่ 1 – 4 ค่า

`margin:25px 50px 75px 100px;`

- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

`margin:25px 50px;`

- top & bottom margin is 25px
- right & left margin is 50px

กรณี น้อยกว่า 4 ค่า: เริ่มจาก top ไป right ไป bottom ไป left ถ้าค่าไหนหายไป ให้ใช้ค่าจากทิศฝั่งตรงข้าม

`margin:25px 50px 75px;`

- top margin is 25px
- right & left margin is 50px
- bottom margin is 75px

`margin:25px;`

- all margins are 25px

# HTML Table Formatting [4]

- ในการจัดรูปแบบตาราง การกำหนด style ให้แต่ละ element ที่มีรูปแบบเหมือนกันเป็นจำนวนมาก (เช่นการกำหนดขนาด Font ใน Cell <td> ให้เหมือนกันทั้งตาราง) ควรใช้วิธีกำหนดผ่าน **Internal** หรือ **External Style Sheet**

- ไม่ควรใช้วิธีกำหนดค่าแบบ Inline ทีละ Element จนครบ T \_ T

**Note:** Comments in HTML are denoted by <!-- text -->

```
<head><!--internal SS -->
 <style type="text/css">
 ...
 </style>
</head>
```

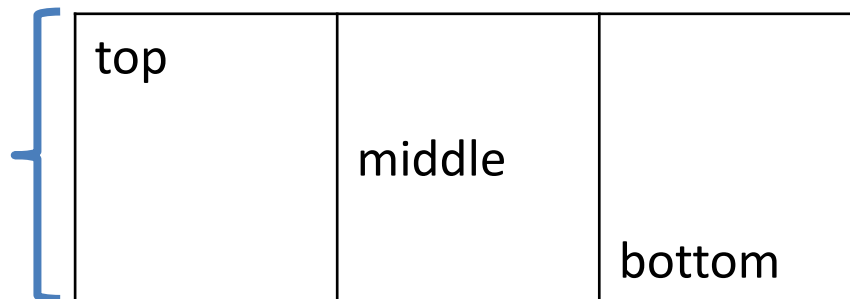
```
<head><!--external SS -->
 <link rel="stylesheet"
 type="text/css"
 href="mystyle.css">
</head>
```

# HTML Table Formatting [6]

- `vertical-align` กำหนดตำแหน่งการจัดวางตัวอักษรในแนวตั้ง (top, middle, bottom)

```
td
{
 height:50px;
 vertical-align:bottom;
}
```

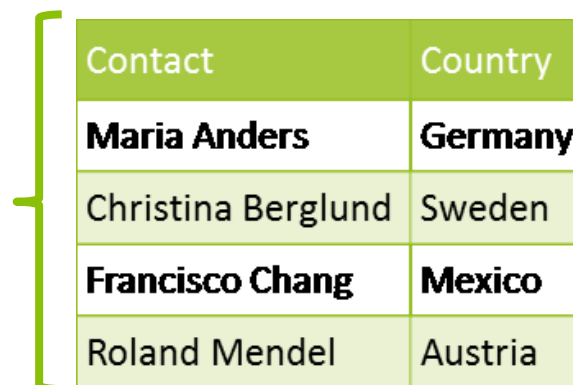
50px



top		
	middle	
		bottom

- `height` กำหนดความสูงของ cell

- Example 3: Fancy Table



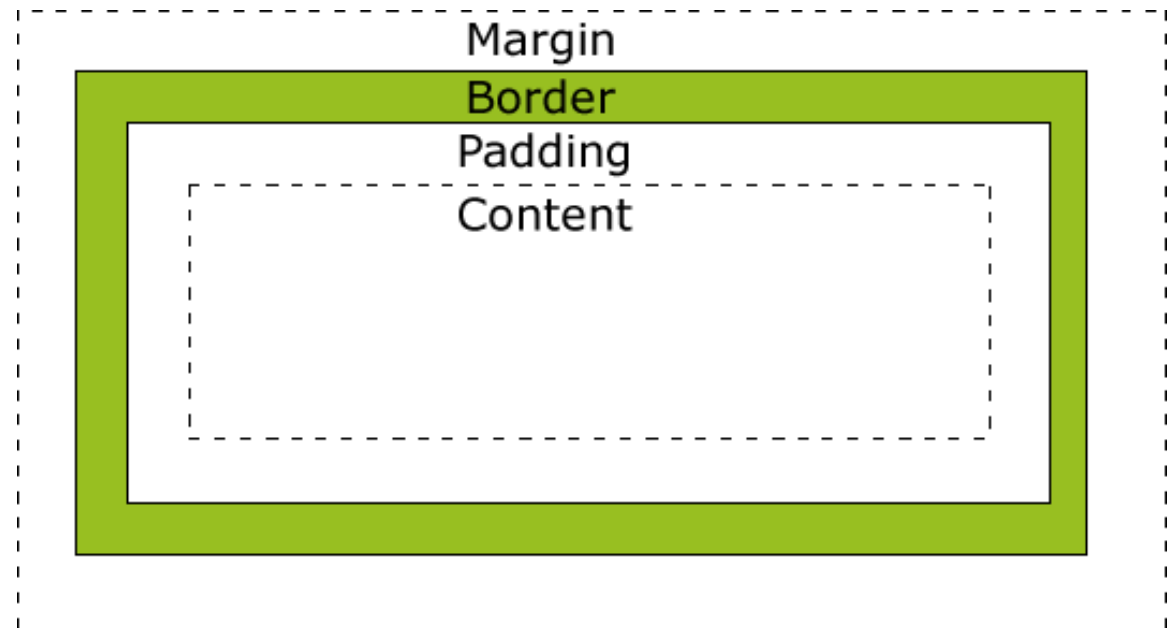
Contact	Country
<b>Maria Anders</b>	<b>Germany</b>
Christina Berglund	Sweden
<b>Francisco Chang</b>	<b>Mexico</b>
Roland Mendel	Austria

# HTML Table Formatting [7]

- padding, padding-top, padding-bottom, padding-left, และ padding-right ใช้กำหนดระยะห่างระหว่าง Content ของ Cell และขอบด้านในของ Cell โดยมี

Syntax

เหมือน "margin"



# HTML Cell Spanning

## ● Example 4: Row and Column Spanning

- เราสามารถกำหนดให้ Cell ในตารางกินพื้นที่ข้าม Column หรือ Row ได้ โดยการใช้ attribute colspan และ rowspan ใน Tag <td> และ <th> โดยจะทำให้ผลลัพธ์ที่ได้มีลักษณะเหมือนการ Merge Cell ใน Application ประเภท Spreadsheet

```
<td rowspan="3">rowspan</td>
...
<td colspan="2">colspan</td>
```

rowspan	2	3
	colspan	
	5	6

# HTML Cell Spanning [2]

- ในการสร้างตารางให้มีลักษณะ Cell Spanning ให้พิจารณาว่าตารางมีทั้งหมดกี่ Row และแต่ละ Row ประกอบด้วย Cell ใดบ้าง

- จากตัวอย่าง มี 3 แถว [r1 – r3]
- แถว r1 -> cell a, b, c
- แถว r2 -> cell d, e
- แถว r3 -> cell f, g

a	b	c
d	e	
f		G

- การพิจารณาว่า Cell ไหนอยู่ Row ใด ให้พิจารณาจากมุมซ้ายบนของ Cell นั้นๆ ว่าอยู่ที่ตำแหน่งใน Row ใด

# HTML Cell Spanning [3]

a [r1]	b [r1]	c [r1]
d [r2]	e [r2]	
f [r3]		g [r3]

```
<tr>
 <td>a</td>
 <td>b</td>
 <td>c</td>
</tr>
<tr>
 <td>d</td>
 <td>e</td>
</tr>
<tr>
 <td>f</td>
 <td>g</td>
</tr>
```

```
<tr>
 <td>a</td>
 <td>b</td>
 <td rowspan="2">c</td>
</tr>
<tr>
 <td>d</td>
 <td>e</td>
</tr>
<tr>
 <td colspan="2">f</td>
 <td>g</td>
</tr>
```

- แล้วจึงพิจารณาว่า Cell ใดมีลักษณะ Spanning ข้าม Row หรือ Column



# References

- W3C HTML Table Reference

<http://www.w3.org/TR/REC-html40/struct/tables.html>

- W3C CSS Units

<http://www.w3.org/Style/Examples/007/units.en.html>

# HTML

## 7. การตกแต่งเว็บเพจด้วยรูปภาพ

รวบรวมโดย อ. กิตติพิชญ์ คุปตะวาณิช

# HTML Images

- เราสามารถแทรกรูปภาพลงในเอกสาร HTML ได้โดยใช้ Tag `<img>`
  - โดยรูปที่นำมาแสดงผลสามารถอยู่ Website เดียวกับเอกสาร
  - หรือเป็นรูปที่เชื่อมโยงมาจาก Web Resource จาก Site อื่นๆ ก็ได้
- ตัวอย่าง Format ของ ไฟล์รูป
  - GIF: 256 colors, lossless compression, supports animation and transparency
  - JPEG: lossy compression, 24-bit color (8 bits each for red, green and blue), progressive display, smaller file size, great for finished product
  - PNG: successor to GIF, lossless compression, support transparency, progressive display, good for file under editing.
  - etc

# HTML `<img>` Tag

- Tag `<img>` เป็น Tag ว่าง (Empty Tag) ไม่จำเป็นต้องใส่ Tag ปิด
- Attribute ของ Tag `<img>`

- `src` กำหนด URI ของรูปที่จะใช้แสดงผล

```


```

- `width` กำหนดความกว้างของ Image
- `height` กำหนดความสูงของ Image

- จำนวนเต็มบวก มีหน่วยเป็น px แต่ไม่จำเป็นต้องใส่หน่วยเมื่อกำหนดค่า

```

```

- `alt` กำหนดคำอธิบายรูปภาพ

# The alt Attribute [2]

- ในกรณีที่รูปภาพไม่สามารถแสดงผลบนเอกสาร HTML ได้
  - มีปัญหาด้าน Internet (Speed, Connection, etc)
  - ไม่สามารถเข้าถึง Resource (รูปภาพ) ได้

ข้อความที่กำหนดไว้ที่ Attribute alt ก็ จะแสดงผลแทน

- ในบาง Browser เช่น IE (Internet Explorer) ข้อความที่กำหนดโดย alt จะแสดงผลในลักษณะ Tooltip

- ไม่ใช้การแสดงผล alt ตามมาตรฐานที่กำหนด หากต้องการแสดงผล Tooltip ควรใช้ Attribute **title** แทน



# HTML `<img>` Tag [2]

- Example 1: `<img>` Attribute

```
<body>
```

```

</body>
```

- สังเกตการใช้ Attribute `alt` ในการบรรยายภาพ

# HTML

## 6. การเชื่อมโยงเว็บเพจ (HYPERLINK)

รวบรวมโดย อ. กิตติพิชญ์ คุปตะวาณิช

# HTML Links

- Hyperlink หรือ Link คือ ตัวอักษรหรือกลุ่มตัวอักษร (คำ หรือกลุ่มคำ) หรือ รูปภาพที่เชื่อมต่อไปยัง Resource อื่นบน Web
- การสร้าง Link ใน เอกสาร HTML ทำได้โดยการใช้ Tag `<a>` โดย `a` คือ Anchor ซึ่งหมายถึงจุดเชื่อมโยง
- ใน 1 Link จะประกอบด้วย 2 Anchor คือ **Source** Anchor (จุดต้นทาง) และ **Destination** Anchor (จุดปลายทาง)
  - Tag `<a>` สามารถใช้ระบุ Source หรือ Destination Anchor ก็ได้
- จุดปลายทางสามารถเป็นจุดอ้างอิงภายในหน้า เอกสาร HTML เดียวกัน, เอกสาร HTML อื่น, ไฟล์ Video, ไฟล์เสียง หรือไฟล์ชนิดอื่น ๆ



# HTML `<a>` Tag

- Attribute ของ Tag `<a>`

- href (Hypertext Reference) กำหนด URI (Uniform Resource Identifier) ของจุดปลายทางซึ่งเป็น Attribute ที่สำคัญที่สุด ในการสร้าง Link กรณีนี้ Tag `<a>` ทำหน้าที่เป็นจุดต้นทาง (Source Anchor) โดยการระบุ URI จะอยู่ในรูปดังนี้

- กรณีเชื่อมโยงไปยังเอกสาร HTML ที่ Site อื่น

```
Click Here
```

» สังเกตการระบุ http: ภายในเครื่องหมาย ""

- กรณีเชื่อมโยงไปยังเอกสาร HTML ภายใน Site เดียวกัน

```
Click Here
```

# URI vs URL

- **URI หรือ Uniform Resource Identifier** เป็นกลุ่มอักขระ ที่ใช้ระบุ หรืออ้างถึง ทรัพยากร (Resource) ต่าง ๆ บน Web เช่น

- <http://www.google.com>
- <ftp://weblab.cs.science.cmu.ac.th>
- <mailto:zuck@facebook.com>
- <urn:issn:1535-3613>
- [.././resource.txt](#)
- [www.google.com](http://www.google.com)
- <facetime://+19995551234>
- <sms:+15105550101?body=hello%20there>

- ส่วนที่อยู่ซ้ายสุด ก่อนเครื่องหมาย : คือ **URI scheme**
- **URL** คือ **URI** ที่มีการระบุวิธีในการเข้าถึงข้อมูลนั้น ๆ เช่น **http: ftp: mailto:**

# The **mailto:** URL

- เราสามารถใช้ URI Scheme **mailto:** ในการสร้าง Link เพื่อส่ง email ได้โดยใช้รูปแบบดังนี้

```
Mail Zuck
```

```

```

- ระบุ subject

```

```

- ระบุ body (เนื้อหาจดหมาย) สังเกตการใช้ **%20** แทนอักขระ space (ช่องว่าง) และ **%0D%0A** แทน line break

```

```

- สังเกตการใช้เครื่องหมาย **&** เมื่อต้องการระบุมากกว่า 1 field (กรณีนี้คือ cc และ body)

# HTML <a> Tag [2]

\*เพื่อการอ้างอิงเท่านั้น ไม่แนะนำให้ใช้ Frame ในการวาง Layout

## ● Attribute ของ Tag <a> [2]

- target กำหนดว่าต้องการให้แสดงผลหน้าปลายทางในลักษณะใด  
`<a target="value" href = ".....">Click Here</a>`

Tag	Description
<code>_blank</code>	เปิด link และแสดงผลในหน้าใหม่ หรือ tab ใหม่
<code>_self</code>	เปิด link และแสดงผลใน frame ปัจจุบัน (default)
<code>_parent*</code>	เปิด link และแสดงผลใน frame แม่ (parent frame) ของ frame ปัจจุบัน
<code>_top*</code>	เปิด link และแสดงผลเต็มหน้า browser (ทับหน้าเดิมทั้งหมดของทุก frame)
<code>framename*</code>	เปิด link และแสดงผลใน frame ที่มีการระบุชื่อ

# HTML `<a>` Tag [3]

## ● Attribute ของ Tag `<a>` [3]

- name หรือ id กำหนดชื่อหรือ id ของ Anchor ซึ่งทำให้ใช้ Anchor นั้นๆ เป็นจุดปลายทาง (Destination Anchor) ได้

- การกำหนด id หรือ name ในเอกสารใดๆ ไม่สามารถใช้ซ้ำกันได้ เนื่องจาก id และ name ใช้ name space เดียวกัน

```
Section 2
```

- การกำหนด section2 เป็นจุดปลายทางทำได้จากจุด Anchor ภายในหน้าเอกสารอื่นหรือภายในหน้าเอกสารเดียวกันก็ได้

```
Click
```

OR

```
Click
```

# HTML Link Formatting

- เช่นเดียวกับกับ Element อื่นๆ เราสามารถจัดรูปแบบ Link ได้โดยใช้ Property ต่างๆ ใน style
- Example 1: Text Decoration
  - text-decoration กำหนดลักษณะการตกแต่งตัวอักษร
    - none - ตัวอักษรปกติไม่มีการตกแต่ง หากใช้กับ Link จะไม่มีการเส้นใต้
    - underline - มีการขีดเส้นใต้ตัวอักษร e.g. characters
    - overline - มีการขีดเส้นเหนือตัวอักษร e.g. characters
    - line-through - มีเส้นขีดทับกลางตัวอักษร e.g. ~~characters~~
    - blink - อักษรกระพริบ
    - inherit - ใช้ลักษณะเดียวกับ Parent Element

# HTML Link Formatting [2]

- นอกจากนี้เรายังสามารถกำหนด Style ให้กับสถานะต่างๆ ของ Link ได้โดยสถานะของ Link ที่สามารถใช้ CSS กำหนดรูปแบบได้คือ
  - Link ที่ยังไม่ได้คลิก (Unvisited Link) หรือ Link ในสถานะปกติ

```
a:link {color:#8AC007;}
```
  - Link ที่คลิกไปแล้ว (Visited Link)

```
a:visited {color:#8AC007;}
```
  - Link เมื่อนำ Mouse มาชี้ (Mouse Over Link)

```
a:hover {color:#8AC007;}
```
  - Link ขณะถูกคลิก (Selected Link)

```
a:active {color:#8AC007;}
```
- การกำหนดรูปแบบของ Link ต้องทำในลักษณะ Style Sheet เท่านั้น (Internal or External) ไม่สามารถกำหนดแบบ inline ได้

# HTML Link Formatting [3]

## ● Example 2

```
<head>
 <style>
 a:link {color:#8AC007;}
 a:visited {color:orange;}
 a:hover {color:red;}
 a:active {color:blue;}
 </style>
</head>
```

- การกำหนดรูปแบบ Link สถานะต่างๆ จำเป็นต้องกำหนดตามลำดับก่อนหลังที่ถูกต้อง กล่าวคือ
  - หากมีการกำหนด a:link และ a:visited ต้องกำหนดก่อน a:hover
  - หากมีการกำหนด a:hover ต้องกำหนดก่อน a:active



# References

- W3C HTML: Links <http://www.w3.org/TR/html401/struct/links.html>
- URI Scheme  
[http://en.wikipedia.org/wiki/URI\\_scheme](http://en.wikipedia.org/wiki/URI_scheme)
- mailto: URL  
<http://tools.ietf.org/html/rfc6068>
- W3C CSS Selector Pseudo Class  
<http://www.w3.org/wiki/CSS/Selectors#Pseudo-classes>
- W3Schools CSS Links  
[http://www.w3schools.com/css/css\\_link.asp](http://www.w3schools.com/css/css_link.asp)