

PHP

3. 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, อักขระพิเศษ หรือ 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                                                                     |
|---------------------|---------------------------------------------------------------------------------|
| \\                  | Inserts a backslash                                                             |
| \\$                 | Inserts a dollar sign                                                           |
| \r                  | Inserts a carriage return                                                       |
| \"                  | Inserts a double quotation mark                                                 |
| \t                  | Inserts a horizontal tab                                                        |
| \n                  | Inserts a new line                                                              |
| \regular expression | 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 <b>String</b> ตามช่วงความยาวที่ต้องการ              |
| <a href="#"><u>substr_compare()</u></a> | เปรียบเทียบส่วนของ <b>String</b>                         |
| <a href="#"><u>substr_count()</u></a>   | นับจำนวน <b>String</b> ที่ปรากฏในเอกสาร                  |
| <a href="#"><u>substr_replace()</u></a> | แทนค่า <b>String</b> ที่เจอด้วยค่าที่ต้องการ             |
| <a href="#"><u>trim()</u></a>           | ตัดส่วนที่เป็น <b>Whitespace</b> ทั้งด้านหน้าและด้านหลัง |

# PHP: String Function

- สำหรับฟังก์ชันที่มักใช้งานกับ String สามารถแยกเป็นกลุ่ม ๆ 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**

# References

- **PHP String and its Functions**

<http://www.php.net/manual/en/language.types.string.php>

<http://www.php.net/manual/en/ref.strings.php>

# PHP

## 4. 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
```

```
$colors = array("red", "green", "blue", "yellow");
foreach ($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 "
";
}
```

```
?>
```

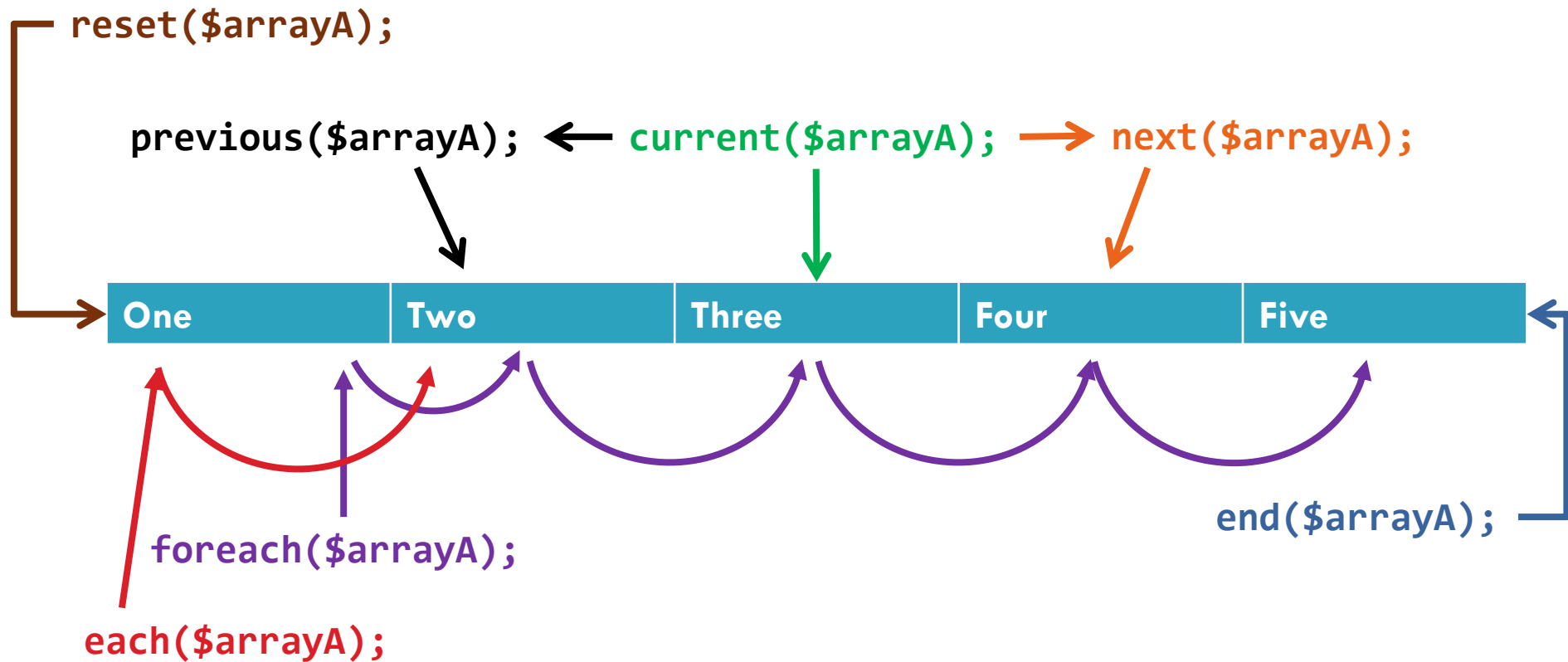
## □ Result

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           |

# References

- **PHP Array and Its Functions**

<http://www.php.net/manual/en/language.types.array.php>

<http://www.php.net/manual/en/ref.array.php>