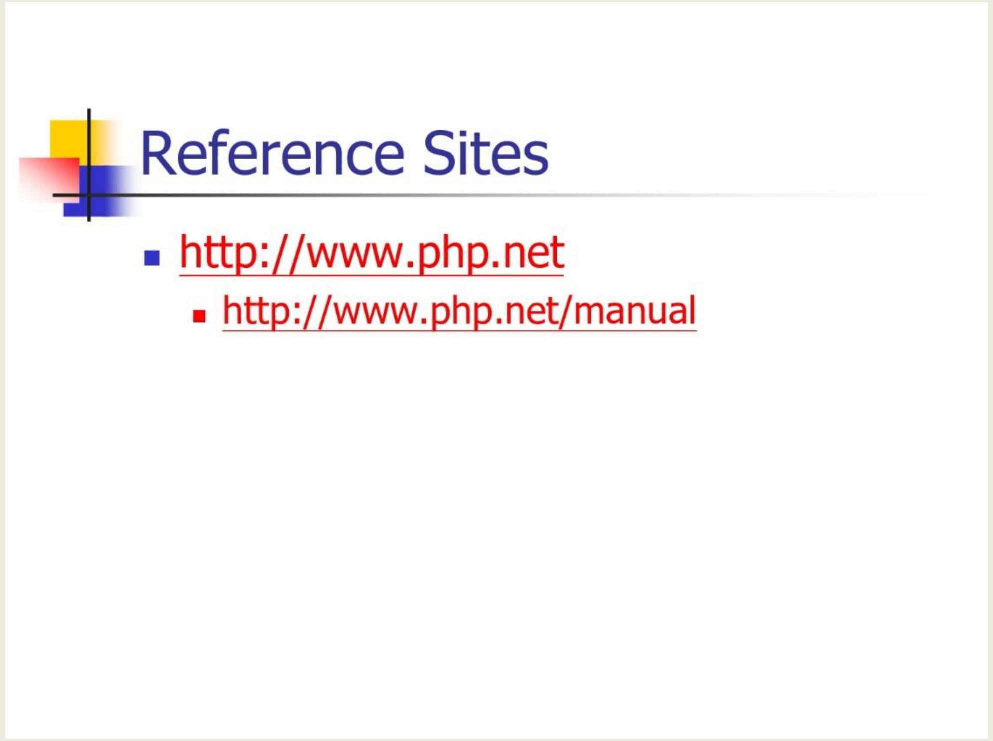
PHP

Final Homework

110816032 謝妤婕 OuO



Reference Sites

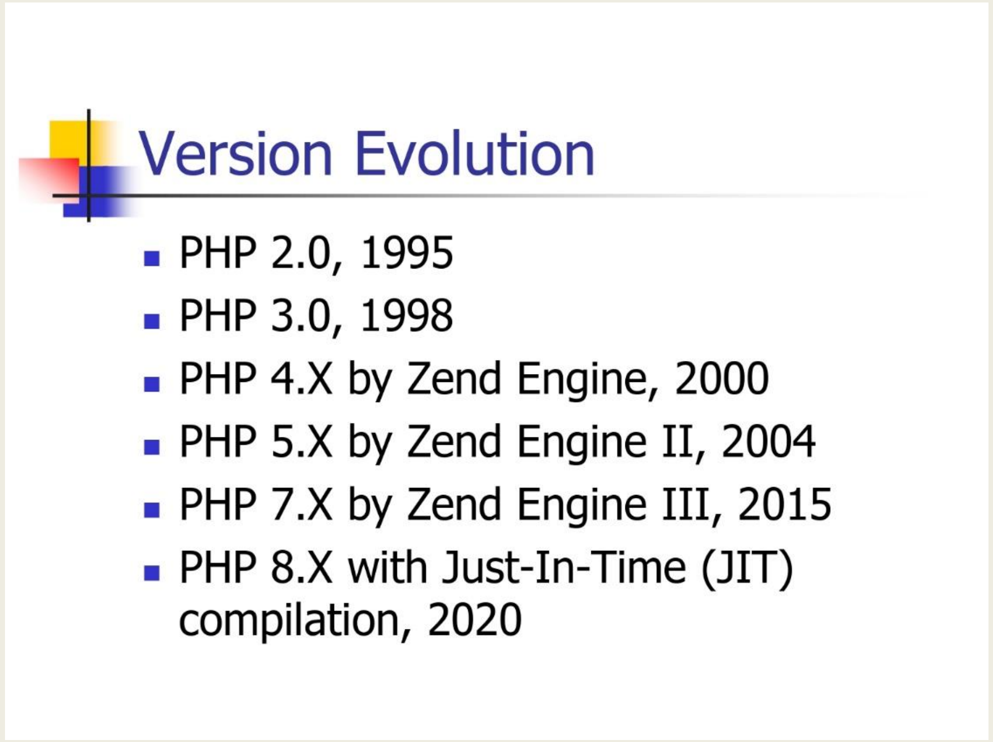
- http://www.php.net
  - http://www.php.net/manual

Reference

《PHP 學習手冊》

《PHP 設計模式學習手冊》

page-0006



Version Evolution

- PHP 2.0, 1995
- PHP 3.0, 1998
- PHP 4.X by Zend Engine, 2000
- PHP 5.X by Zend Engine II, 2004
- PHP 7.X by Zend Engine III, 2015
- PHP 8.X with Just-In-Time (JIT) compilation, 2020

版本差異

PHP 4

沒有套件管理器、自動加載機制，軟體要 include 所有檔案超級肥。

PHP 5.3

Namespace、anonymous functions、Closure、語法糖*?:* (三元運算子)

PHP 5.4

Composer 管理套件、自動加載機制、節省記憶體。

PHP 7.0

新增了 Error 類別

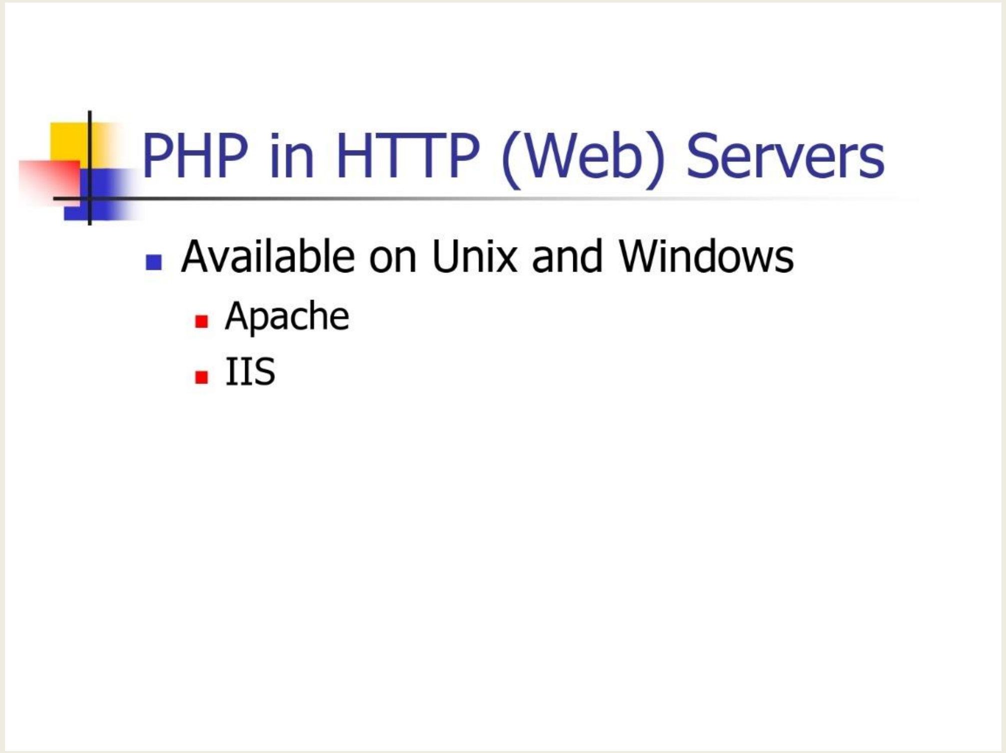
語法糖*??* *$action = $_POST['choice'] ?? 'no';* 等於

if (isset($_POST['choice'])) {$action = $_POST['choice'] ; }

else {$action = 'no' ; }

PHP 7.1

函式方法中新增了 null (?代表 null) 返回值新增了 void 這個類型。



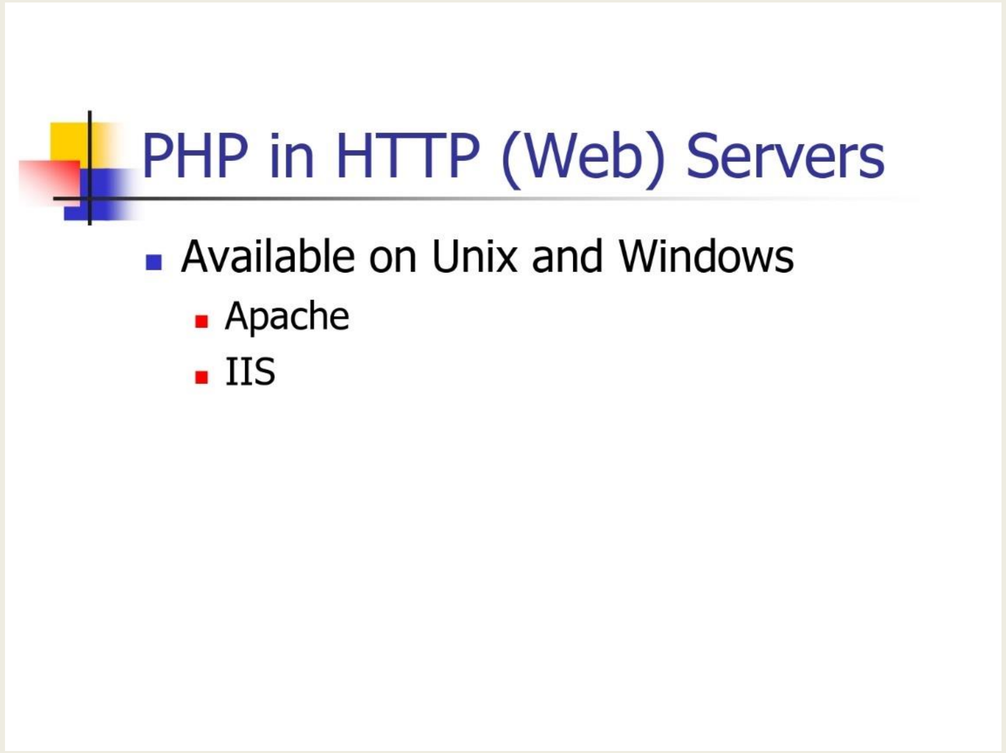
PHP in HTTP (Web) Servers

- Available on Unix and Windows
  - Apache
  - IIS

可內建 Web Server(PHP5.4 後)

不用 Apache 或 NGINX 提供 Web server 環境，可以用於本地開發測試。

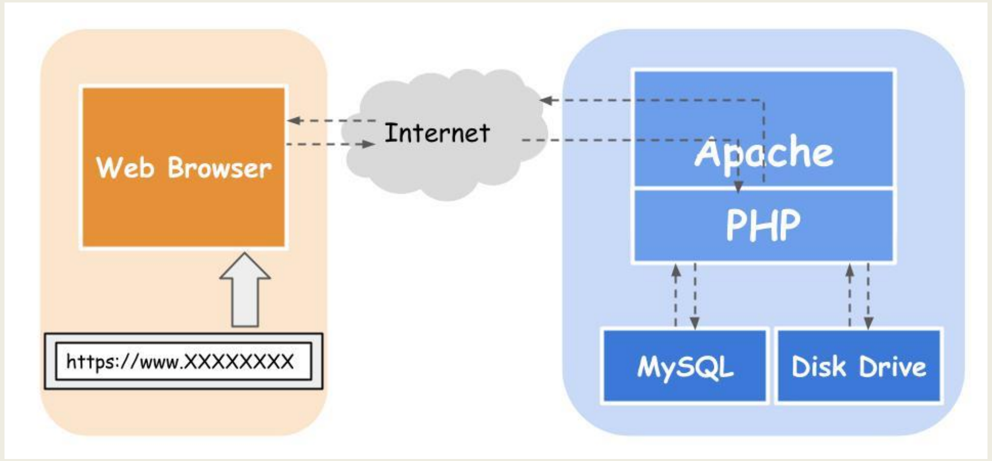
page-0008



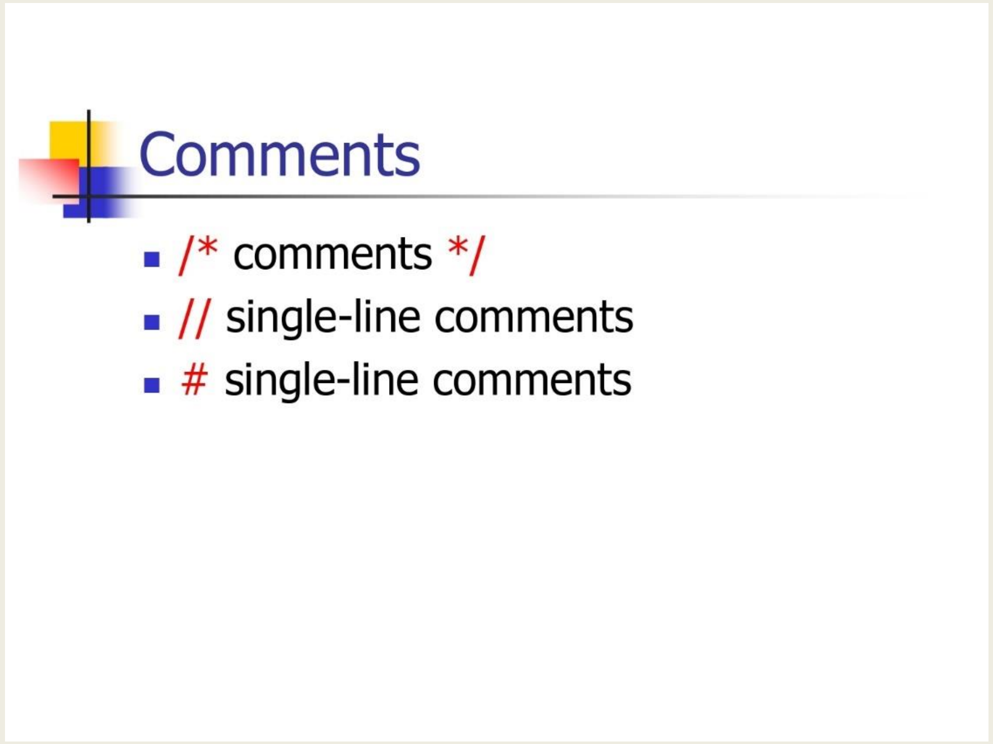
PHP in HTTP (Web) Servers

- Available on Unix and Windows
  - Apache
  - IIS

網頁架構圖



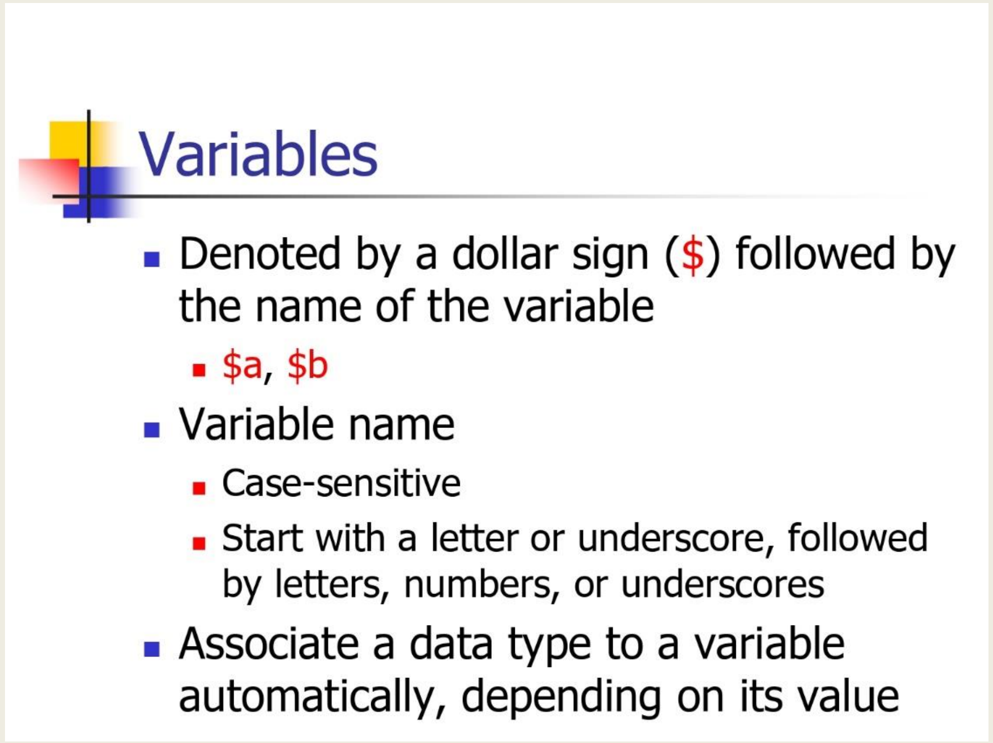
page-0015



Comments

- /* comments */
- // single-line comments
- # single-line comments

```
<!--HTML 註解-->
<?php
# 我也是單行註解
/* 多行註解
        echo "中間";
        echo "都被";
        echo "註解";
        echo "掉了";
        echo "啊啊啊！！！";   */
?>
<!--HTML 註解-->
```



Variables

- Denoted by a dollar sign ($) followed by the name of the variable
  - $a, $b
- Variable name
  - Case-sensitive
  - Start with a letter or underscore, followed by letters, numbers, or underscores
- Associate a data type to a variable automatically, depending on its value

$ABC 不等於$Abc 不等於$aBc

UTF-8

可以使用拉丁字母以外的符號命名變數

ex. $😊 (表情符號是合法的，真的)

page-00XX 數字運算

一般 + - * /

echo 2+3

➔ 5

echo 5-63

➔ -58

echo 5*6

➔ 30

echo 9/5

➔ 1.8 (沒有取整數)

指數 **

echo 5**2

➔ 25

php 5.6 前指數要用 ***pow()***

取餘數 %

echo 124%3

➔ 1

page-00XX 太空船運算子

$A <=> $B

PHP 7.0 後才可使用

類似 strcmp()但不限於字串

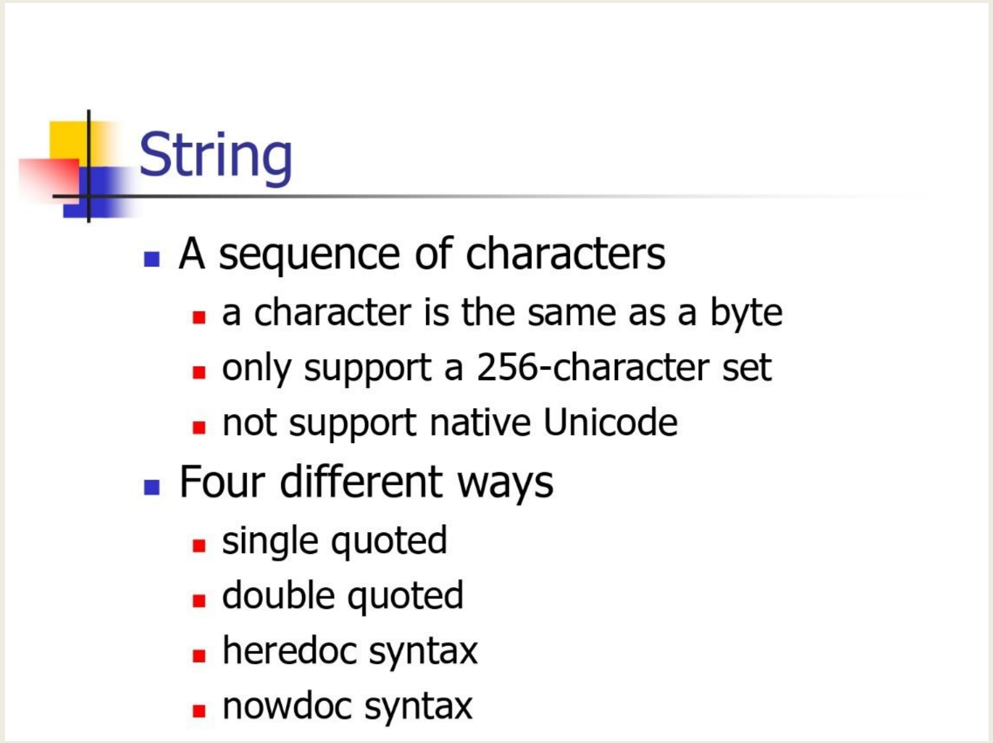
左邊大 return 1

相等 return 0

右邊大 return -1

英文字母按 abc 順序

```
echo "apple" <=> "bee"
➔ 1
echo 9<=>5;
➔ 1
echo 9<=>12.22;
➔ -1
echo 9<=>"123";
➔ -1
```



String

- A sequence of characters
  - a character is the same as a byte
  - only support a 256-character set
  - not support native Unicode
- Four different ways
  - single quoted
  - double quoted
  - heredoc syntax
  - nowdoc syntax

格式化輸出

printf()

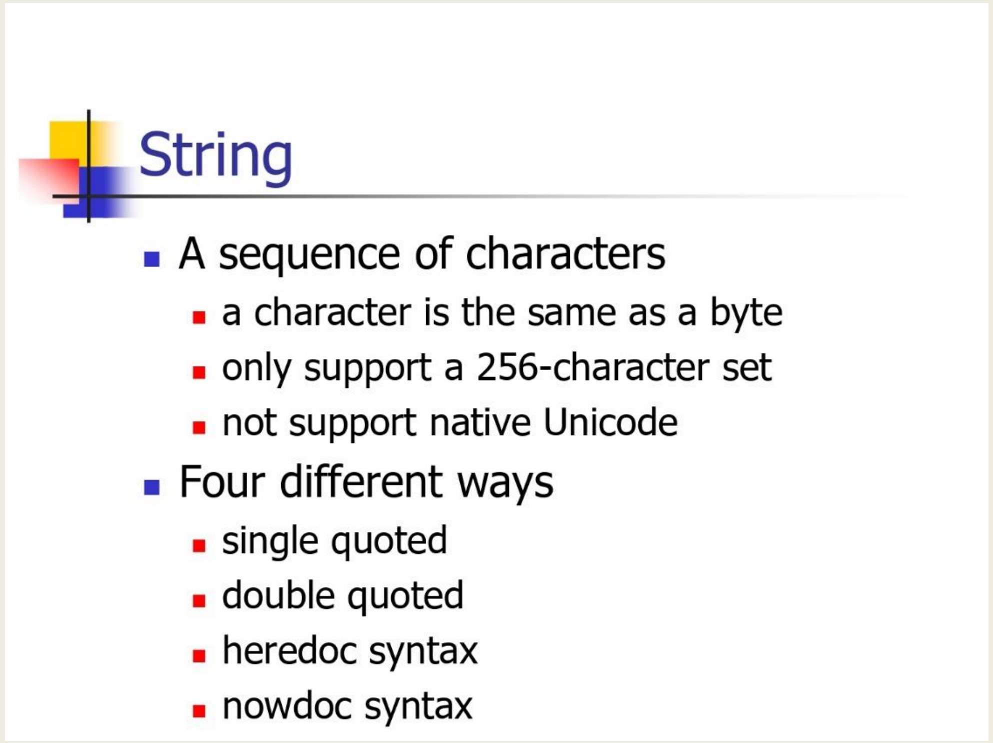
```
$AAA = 123.456
printf("number:%.2f"$AAA)
➔     number:123.45
```

%d (十進位 digital) %f (float)

%05d 以 0 補到最小五位的寬度

%-5d 右側對齊

%.3f 小數點位數



String

- A sequence of characters
  - a character is the same as a byte
  - only support a 256-character set
  - not support native Unicode
- Four different ways
  - single quoted
  - double quoted
  - heredoc syntax
  - nowdoc syntax

String

文字(字串)是 byte 型態**不是 char!!!**

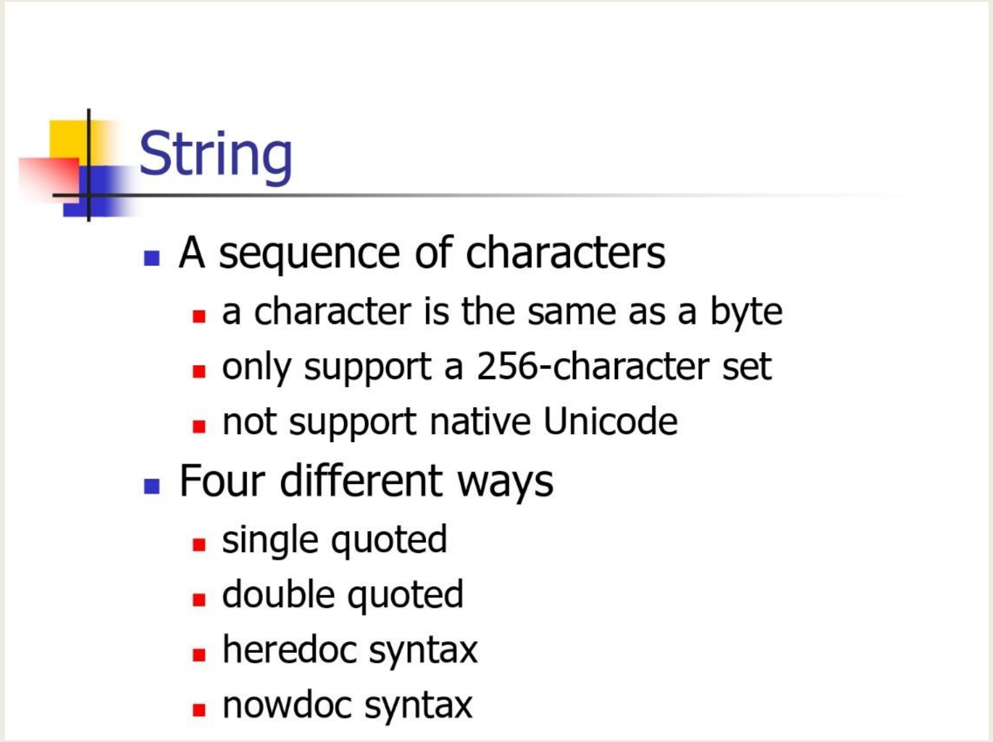
PHP 預設字集為 UTF-8

可以裝 binary file(影像或語音)

字串長度僅受限於電腦記憶體

字串(包含字串的變數)連接用.

$Name = "You" ; $fafa180 = "180cm" ;
echo "<br/>黑占黑占連結字串<br/>"."耶";
echo "聽說".$Name."身高".$fafa180;



String

- A sequence of characters
  - a character is the same as a byte
  - only support a 256-character set
  - not support native Unicode
- Four different ways
  - single quoted
  - double quoted
  - heredoc syntax
  - nowdoc syntax

String

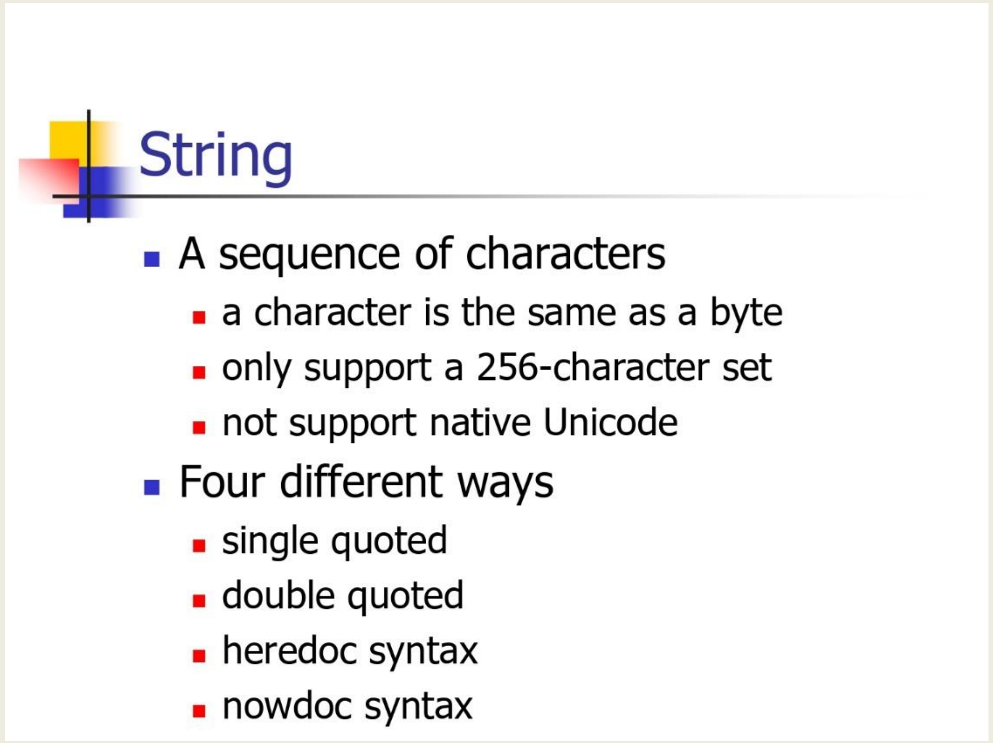
如+= -= *= /=

也有.=

```
$Name = $Name.$Domain;
```

等於

```
$Name .= $Domain;
```



String

- A sequence of characters
  - a character is the same as a byte
  - only support a 256-character set
  - not support native Unicode
- Four different ways
  - single quoted
  - double quoted
  - heredoc syntax
  - nowdoc syntax

strlen()

以 byte 數計算字數，當一個字元大於 1byte 時會回報錯誤。

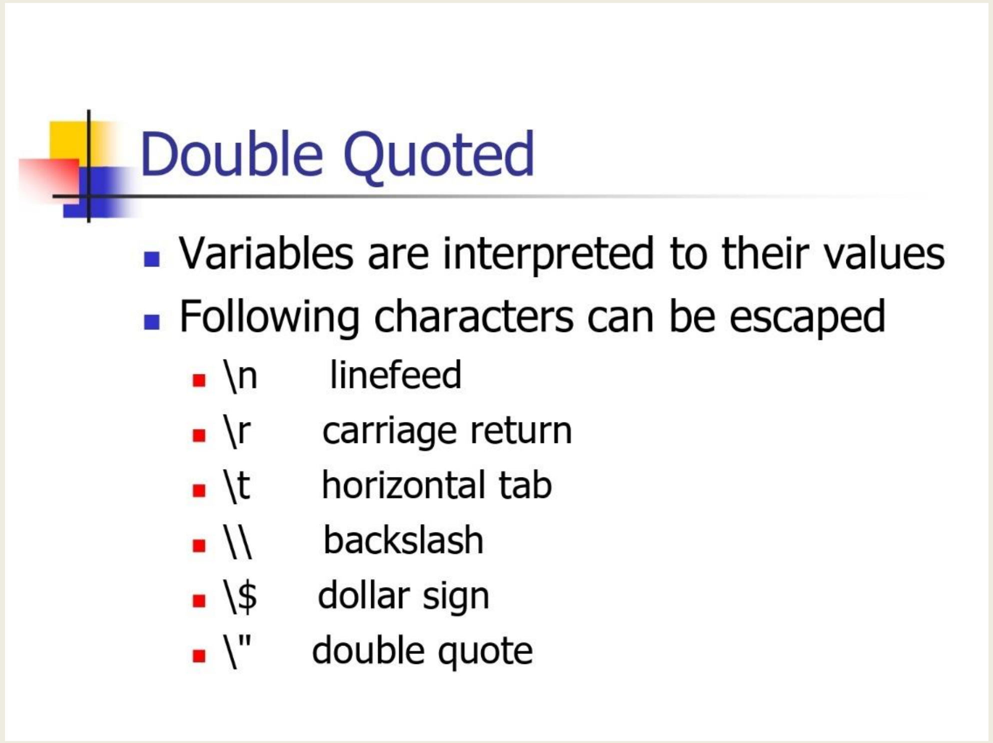
➔ ***mb_strlen()***

substr()

以 byte 為單位擷取子字串。

➔ ***mb_substr ()***

page-0038



Double Quoted

- Variables are interpreted to their values
- Following characters can be escaped

\n	linefeed
\r	carriage return
\t	horizontal tab
\\	backslash
\$	dollar sign
\"	double quote

單引號不處裡跳脫字元!!!

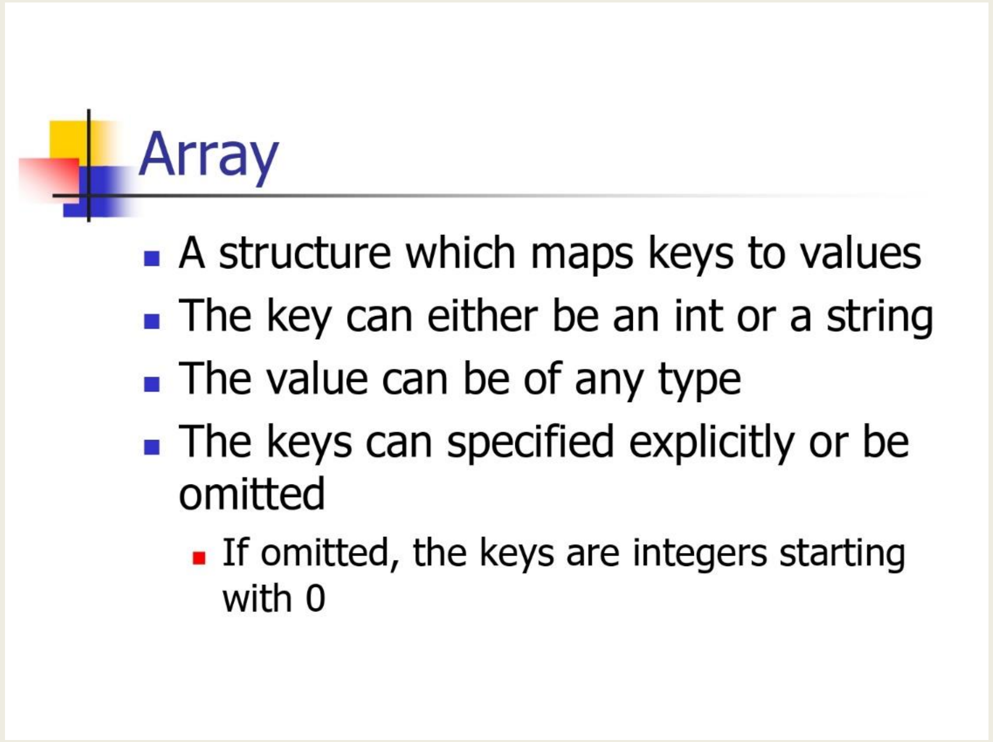
八進制、十六進制(跳脫字元)

\[0-7]{1,3}

the sequence of characters matching the regular expression is a character in octal notation, which silently overflows to fit in a byte (e.g. "\400" === "\000")

\x[0-9A-Fa-f]{1,2}

the sequence of characters matching the regular expression is a character in hexadecimal notation



Array

- A structure which maps keys to values
- The key can either be an int or a string
- The value can be of any type
- The keys can specified explicitly or be omitted
  - If omitted, the keys are integers starting with 0

把 Array 轉成 JSON 格式

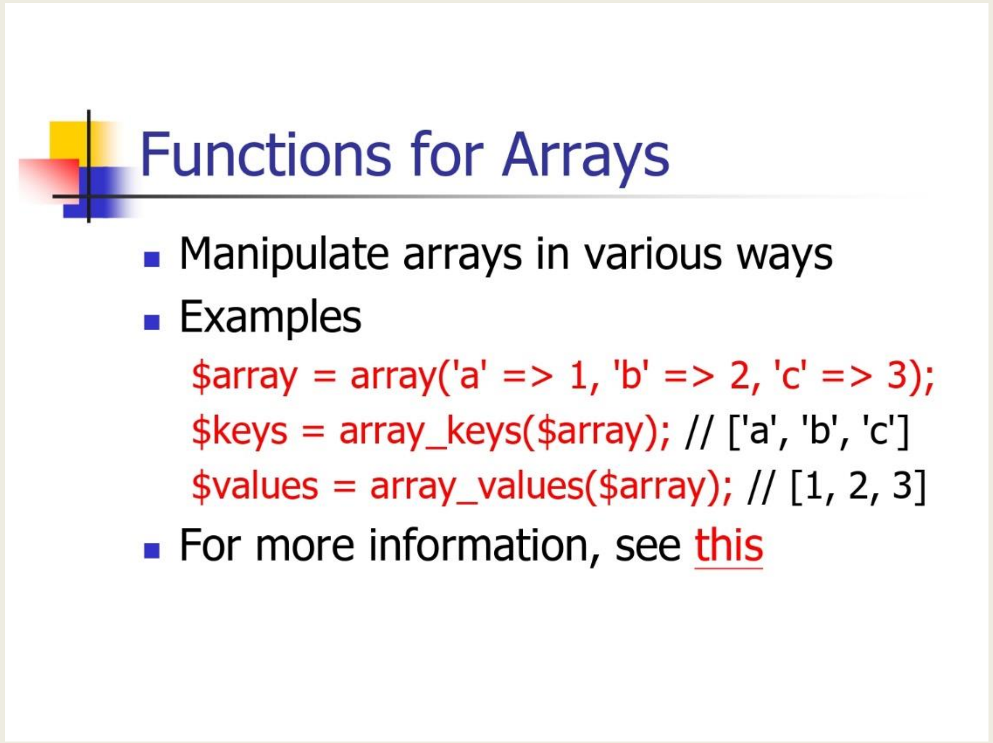
json_encode()

如果中文讓 json_encode()出現亂碼

可以先用 urlencode()

再轉成 JSON 格式

最後再用 urldecode()轉回中文



Functions for Arrays

- Manipulate arrays in various ways
- Examples

```
$array = array('a' => 1, 'b' => 2, 'c' => 3);
$keys = array_keys($array); // ['a', 'b', 'c']
$values = array_values($array); // [1, 2, 3]
```

- For more information, see this

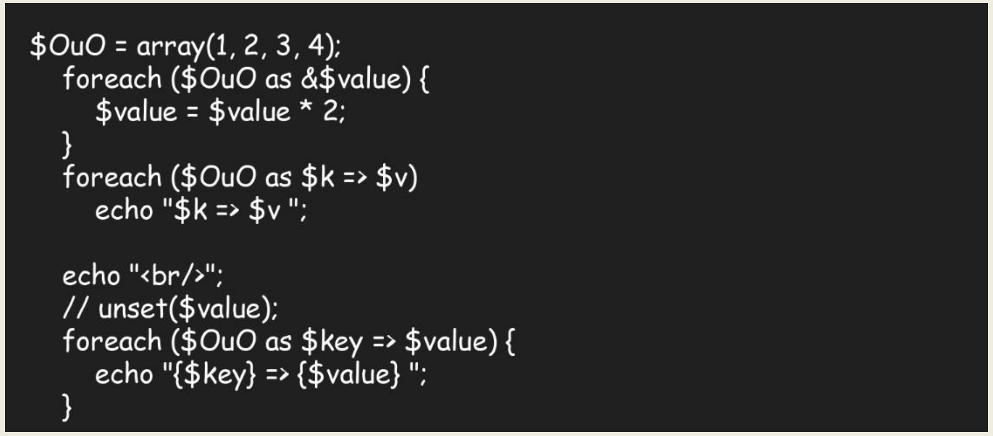
count()

取得陣列大小。

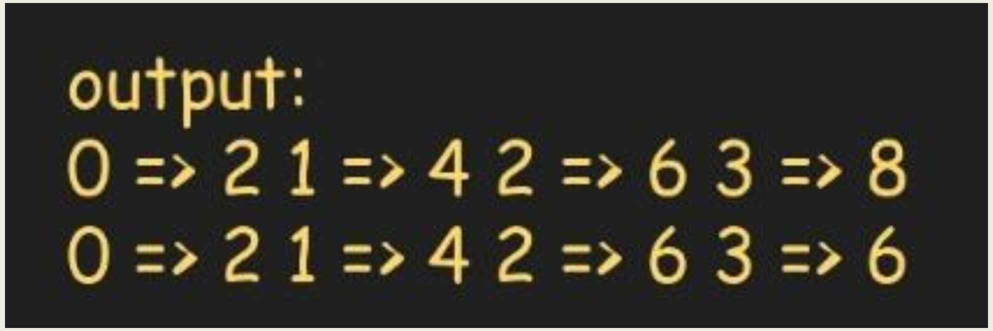
```
$OuO = array("A","b","cde");
count($OuO)
```

➔ 3

page-0088 foreach()& unset()



```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }
```

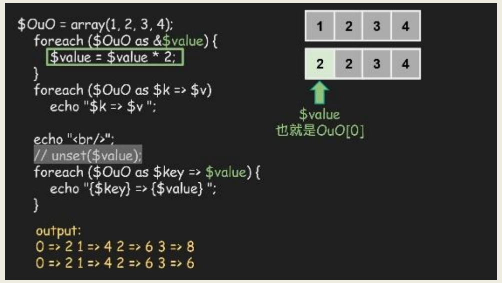


```
output:
0 => 2 1 => 4 2 => 6 3 => 8
0 => 2 1 => 4 2 => 6 3 => 6
```

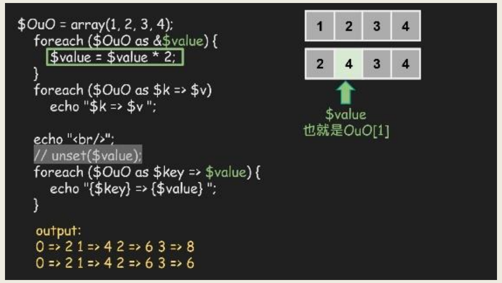
討論為何兩者輸出結果不同?

page-0088 foreach()& unset()

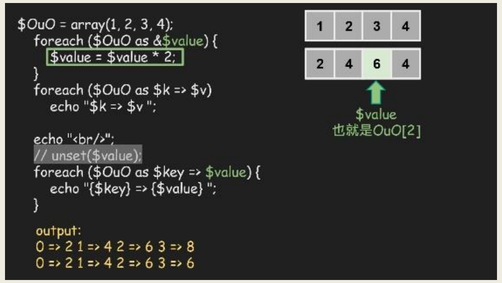
第一行 Output 執行狀況



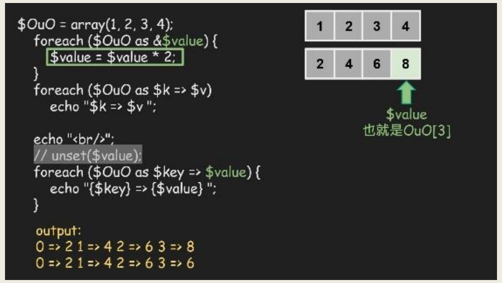
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



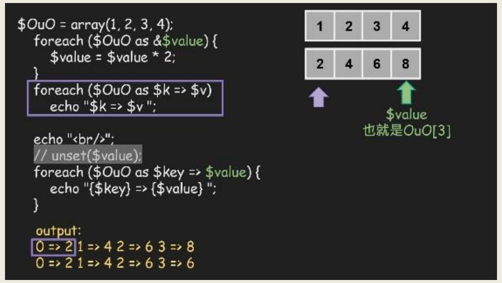
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



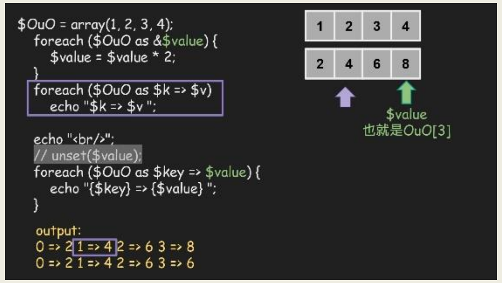
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



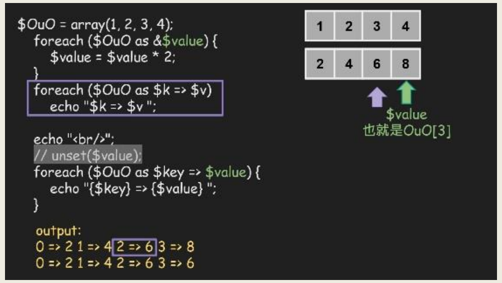
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



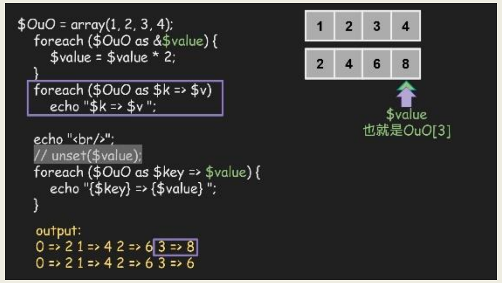
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



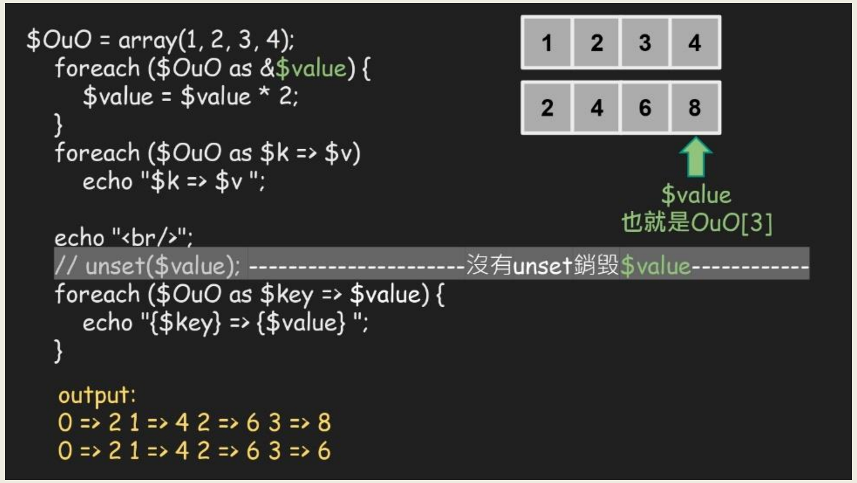
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



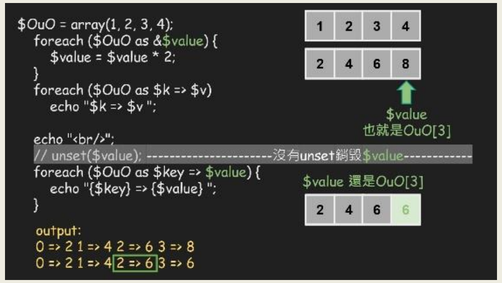
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value);
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```

page-0088 foreach()& unset()

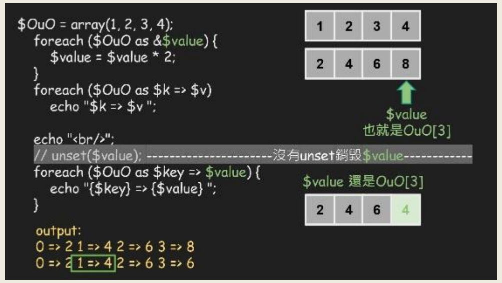
第二行 Output 執行狀況



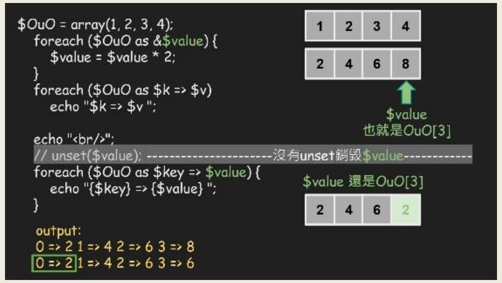
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value); ----------------------沒有unset銷毀$value-----------
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



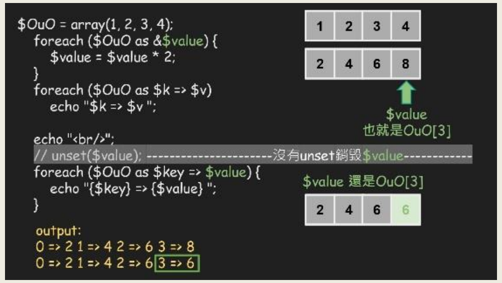
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value); --------------------沒有unset銷毀$value-----------
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value); --------------------沒有unset銷毀$value-----------
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



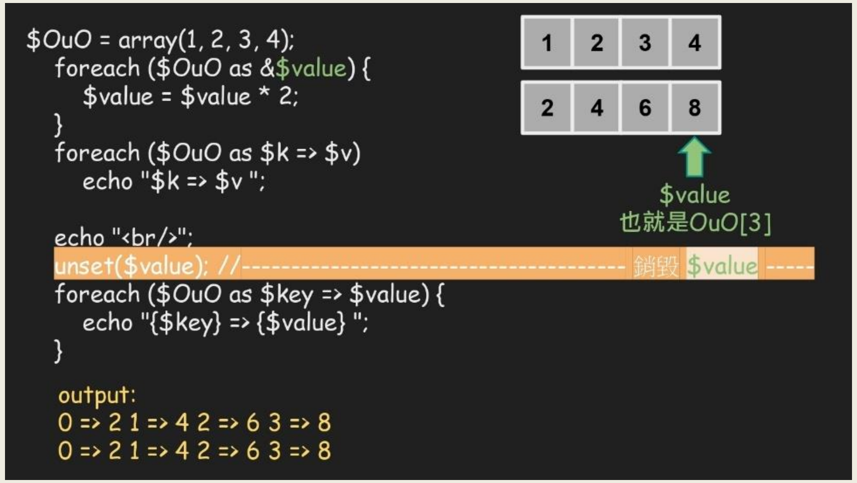
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value); --------------------沒有unset銷毀$value-----------
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```



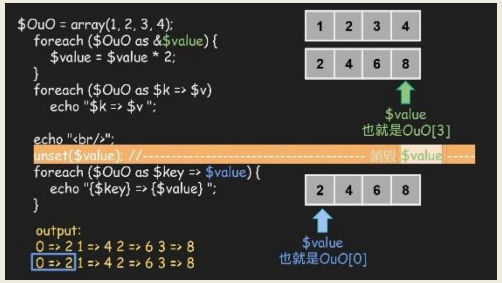
```
$OuO = array(1, 2, 3, 4);
   foreach ($OuO as &$value) {
      $value = $value * 2;
   }
   foreach ($OuO as $k => $v)
      echo "$k => $v ";

   echo "<br/>";
   // unset($value); --------------------沒有unset銷毀$value-----------
   foreach ($OuO as $key => $value) {
      echo "{$key} => {$value} ";
   }

   output:
   0 => 2 1 => 4 2 => 6 3 => 8
   0 => 2 1 => 4 2 => 6 3 => 6
```

page-0088 foreach()& unset()

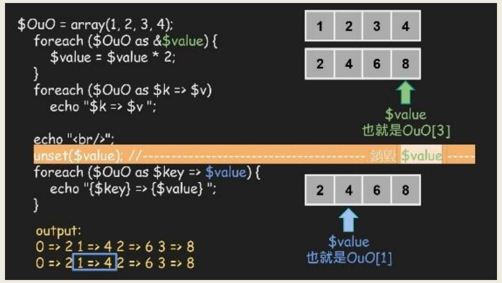
如果使用 unset() 第二行 Output 執行狀況



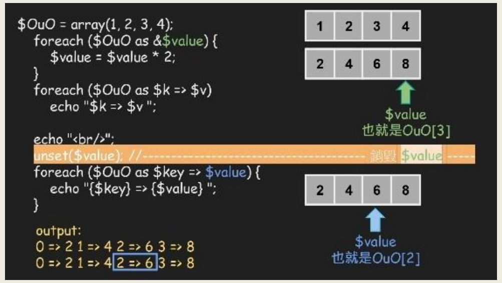
```
$OuO = array(1, 2, 3, 4);
    foreach ($OuO as &$value) {
        $value = $value * 2;
    }
    foreach ($OuO as $k => $v)
        echo "$k => $v ";

    echo "<br/>";
    unset($value); //------------------------------------ 銷毀 $value -----
    foreach ($OuO as $key => $value) {
        echo "{$key} => {$value} ";
    }

    output:
    0 => 2 1 => 4 2 => 6 3 => 8
    0 => 2 1 => 4 2 => 6 3 => 8
```



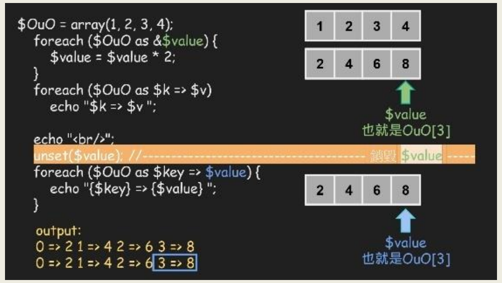
```
$OuO = array(1, 2, 3, 4);
    foreach ($OuO as &$value) {
        $value = $value * 2;
    }
    foreach ($OuO as $k => $v)
        echo "$k => $v ";

    echo "<br/>";
    unset($value); //------------------------------------ 銷毀 $value -----
    foreach ($OuO as $key => $value) {
        echo "{$key} => {$value} ";
    }

    output:
    0 => 2 1 => 4 2 => 6 3 => 8
    0 => 2 1 => 4 2 => 6 3 => 8
```



```
$OuO = array(1, 2, 3, 4);
    foreach ($OuO as &$value) {
        $value = $value * 2;
    }
    foreach ($OuO as $k => $v)
        echo "$k => $v ";

    echo "<br/>";
    unset($value); //------------------------------------ 銷毀 $value -----
    foreach ($OuO as $key => $value) {
        echo "{$key} => {$value} ";
    }

    output:
    0 => 2 1 => 4 2 => 6 3 => 8
    0 => 2 1 => 4 2 => 6 3 => 8
```



```
$OuO = array(1, 2, 3, 4);
    foreach ($OuO as &$value) {
        $value = $value * 2;
    }
    foreach ($OuO as $k => $v)
        echo "$k => $v ";

    echo "<br/>";
    unset($value); //------------------------------------ 銷毀 $value -----
    foreach ($OuO as $key => $value) {
        echo "{$key} => {$value} ";
    }

    output:
    0 => 2 1 => 4 2 => 6 3 => 8
    0 => 2 1 => 4 2 => 6 3 => 8
```



```
$OuO = array(1, 2, 3, 4);
    foreach ($OuO as &$value) {
        $value = $value * 2;
    }
    foreach ($OuO as $k => $v)
        echo "$k => $v ";

    echo "<br/>";
    unset($value); //------------------------------------ 銷毀 $value -----
    foreach ($OuO as $key => $value) {
        echo "{$key} => {$value} ";
    }

    output:
    0 => 2 1 => 4 2 => 6 3 => 8
    0 => 2 1 => 4 2 => 6 3 => 8
```

其他補充

1. HTML 表單製作、儲存狀態
2. 檔案讀寫

1. HTML 表單製作、儲存狀態 -- 一般表單

```
<form action="處理回應的檔案.php" method="post">
表單內容
</form>
```

其中，"處理回應的檔案.php" 可以是自己

1. HTML 表單製作、儲存狀態 -- 特殊表單(如:小算盤)

需要一直針對使用者活動改變狀態並記錄

```
<?php
if(isset($_POST["re_ var"]))
    $var = $_POST["re_ var"];
?>
```

<!--利用 isset()確認物件(上一輪的變數值)是否存在，存在則更新現在的值。-->

1. HTML 表單製作、儲存狀態 -- 特殊表單(如:小算盤)

需要一直針對使用者活動改變狀態並記錄

```
<form action="自己.php" method="post">
表單內容
<!--將這次的執行結果回傳給自己，不然每次都會清零-->
<input type="hidden" name="re_ var" value=" <?php echo $var ?> ">
<!-- "hidden"不影響畫面 name="隨意取一個新的變數名稱"-->
<!-- value = 要記錄的變數 -->
</form>
```

2. 檔案讀寫 -- 整個檔案

整個檔案讀取 ：

```
$fileContent = file_get_contents("檔案名稱.副檔名")
```

將變數寫入(新)檔案 ：

```
file_put_contents("檔案名稱.副檔名", $fileContent)
```

2. 檔案讀寫 -- 部分檔案

利用 file()先對檔案所有內容存取

file()的回傳值為 array，元素是每一行的內容(包含換行\n 在內)

再用 foreach()一行一行處理檔案內容

```
foreach(file("檔案名稱.txt") as $line ){
    $line = trim($line); //利用 trim()移除\n
    print($line)
}
//改自《PHP 學習手冊》p.188
```

2. 檔案讀寫 -- 部分檔案

不用 file()先對檔案所有內容存取

直接一次只讀一行檔案內容 ：fopen() / feof() / fgets() / fclose()

fopen()：建立與檔案的連結

feof() ：檢查是否已經讀到檔尾(End Of File)，回傳值為 Boolean

fgets()：讀取一行檔案內容(讀到\n)，讀完標記會移到下一行開頭，回傳值為 String

fclose() ：~~–斷開魂結–~~關閉與檔案的連結

2. 檔案讀寫 -- 部分檔案

```
$readFile = fopen("檔案名稱.txt", "rb")
while( (! feof($readFile)) && ($line = fgets($readFile)) ) {
    print($line)
}

//改自《PHP 學習手冊》p.189
```

2. 檔案讀寫

fopen("檔案名稱.txt", **"rb")**的**"rb"**是存取模式(file mode)

mode	action	標記初始位置	是否清除內容?	檔案不存在
rb	R	檔頭	否	return false 並警告
rb+	R、W	檔頭	否	return false 並警告
wb	W	檔頭	是	建立新檔
wb+	R、W	檔頭	是	建立新檔
ab	W	檔尾	否	建立新檔
ab+	R、W	檔尾	否	建立新檔

mode	action	標記初始位置	是否清除內容?	檔案不存在
xb	W	檔頭	否	建立新檔 但若檔案已存在 return false 並警告
xb+	R、W	檔頭	否	建立新檔 但若檔案已存在 return false 並警告
cb	W	檔頭	否	建立新檔
cb+	R、W	檔頭	否	建立新檔