網頁程式設計
PHP程式設計

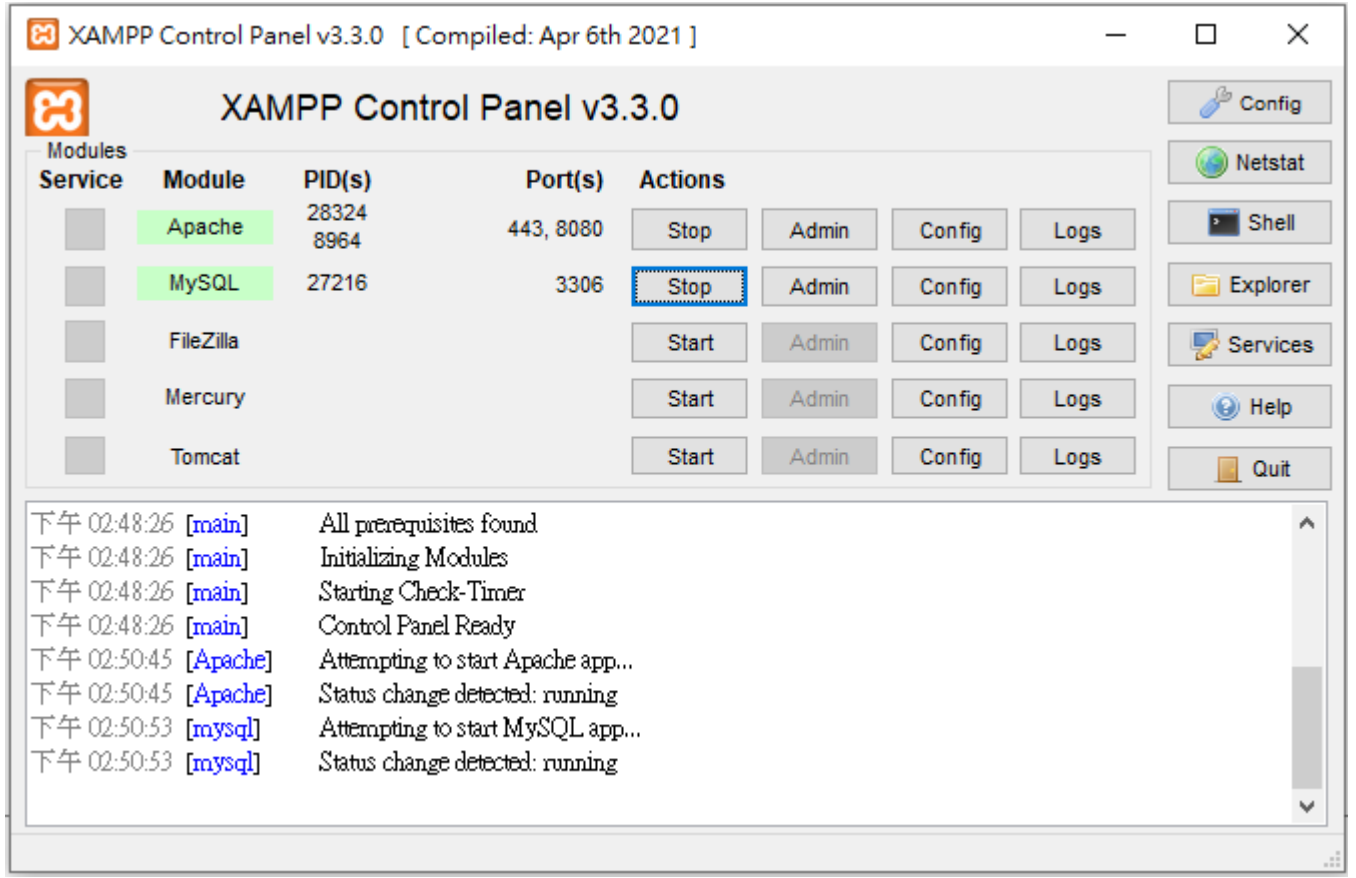
Instructor: 馬豪尚

XAMPP套件

› XAMPP是一套整合Apache網頁伺服器、MariaDB資料庫、PHP和Perl程式語言的架站工具整合包

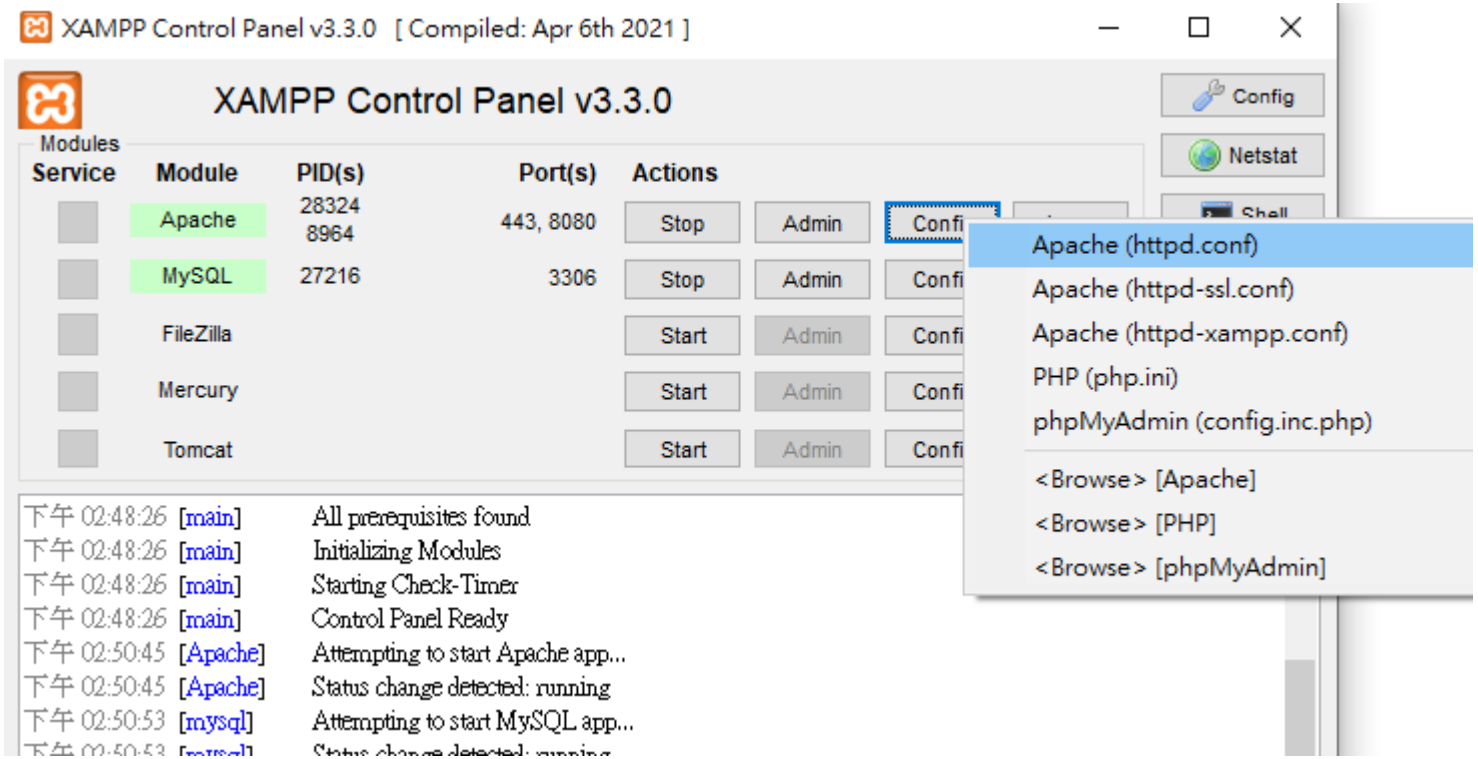
XAMPP套件控制面板

› 透過控制面板可以開啟Apache、MySQL或其他的服務功能



XAMPP套件控制面板

› 修改Apache服務使用的通訊port



XAMPP套件控制面板

- 修改Apache服務使用的通訊port

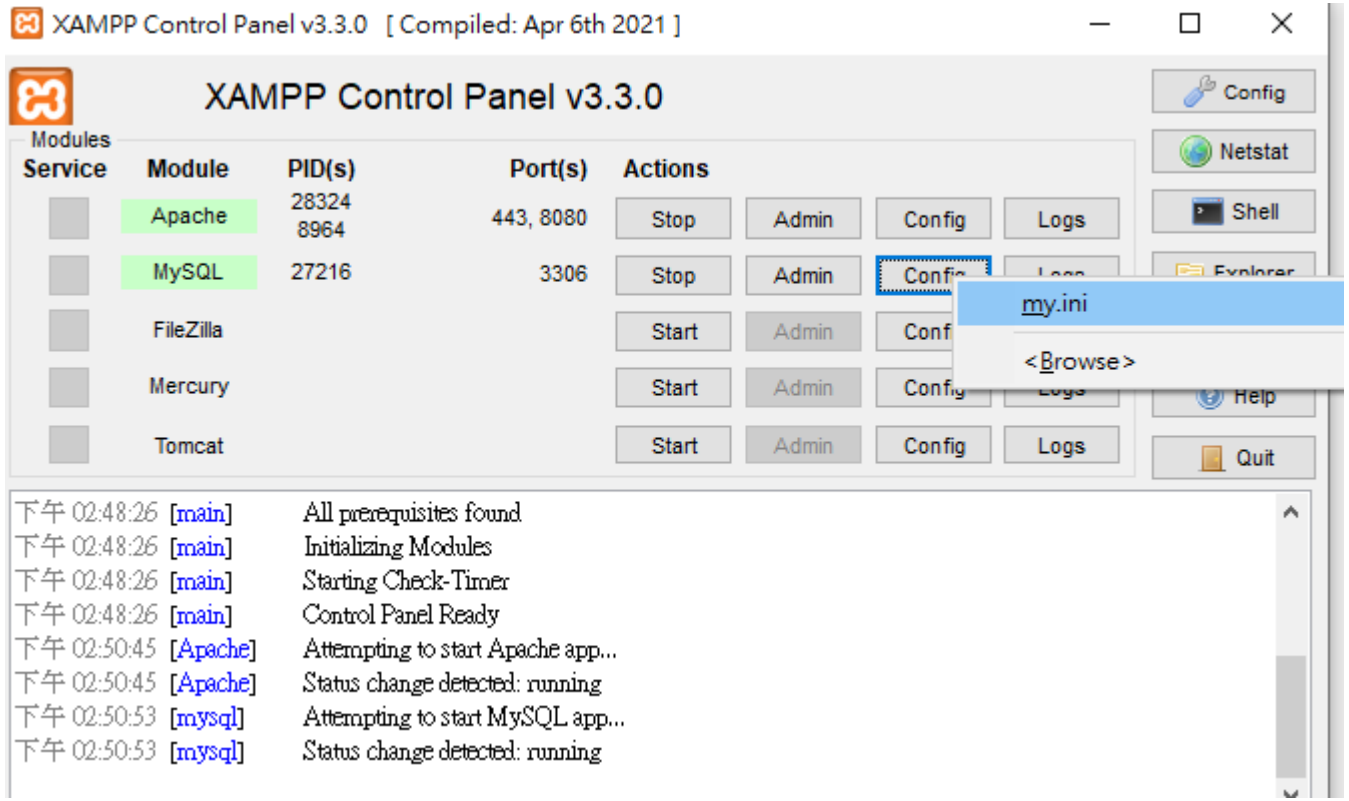
```
#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
#
# Uncommment and change the directory if mutexes are file-based and the default
# mutex file directory is not on a local disk or is not appropriate for some
# other reason.
#
# Mutex default:logs

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8080

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
```

XAMPP套件控制面板

› 修改MySQL服務使用的通訊port



XAMPP套件控制面板

› 修改MySQL服務使用的通訊port

```
#
# In this file, you can use all long options that a program supports.
# If you want to know which options a program supports, run the program
# with the "--help" option.

# The following options will be passed to all MySQL clients
[client]
# password       = your_password
port=3306
socket="C:/xampp/mysql/mysql.sock"


# Here follows entries for some specific programs

# The MySQL server
default-character-set=utf8mb4
[mysqld]
port=3306
socket="C:/xampp/mysql/mysql.sock"
basedir="C:/xampp/mysql"
tmpdir="C:/xampp/tmp"
datadir="C:/xampp/mysql/data"
pid_file="mysql.pid"
# enable-named-pipe
key_buffer=16M
```

為什麼要用PHP

- HTML、CSS、JavaScript已經可以完成一個網站
- PHP最大的目的
  - 接收使用者傳送資料
  - 傳送資料給使用者
  - PHP可以和許多資料庫系統結合去儲存資料

第一個PHP程式

- PHP程式語言基本寫法
  - PHP可以寫在HTML文件裡，但副檔名一定要存成 .php
  - 程式開始與結束
    - <?php...?>
- 無法透過瀏覽器直接開啟檔案，因為PHP是在伺服器端執行
  - 要將檔案放在伺服器的某個位置下
    - ./xampp/htdocs/
  - 用瀏覽器的網址來指定路徑開啟，Xampp套件的預設網址路徑
    - http://localhost:8080/xxx.php

第一個PHP程式

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>ch2-3.php</title>
    </head>
    <body>
        <h2>我的PHP程式</h2><br/>
        <?php
            echo “Hello World!”
        ?>
    </body>
</html>
```

PHP程式的基本結構

- 由於PHP是可以內嵌於HTML的語言，有時一個檔案會同時具有PHP和HTML，你必須告訴電腦說那些部分需要用PHP語言來解析
  - <?php...?>
  - 每一個指令完成後需要加分號來隔開
- 一個文件內的PHP程式是可以有很多個PHP語法區塊的，會照上到下順序執行
- 可以內嵌在HTML的標籤內來執行


國立臺中科技大學
NATIONAL TAICHUNG UNIVERSITY OF SCIENCE AND TECHNOLOGY

PHP基礎語法

PHP變數

- PHP的變數不需要事先宣告，也不需要指定型態
- 使用時直接給予變數名稱和其值即可
- 變數名稱前面必須要加上錢字符號

變數的命名規則

- 可以加底線
  - 可以命名$value和$_value做區隔
- 英文命名與語意化命名
  - 避免$a1, $a2 之類
- 大小寫不同視為不同變數
  - $Value與$value視為不同變數
- 不能以數字為開頭
  - $3value

PHP資料型態 – 數值

- 在PHP的數值資料型態有分成整數(Integers)和浮點數(float)
- Integers資料型態
  - 整數就是指 0, 1, 2, 3, -1, -2 ….
  - 整數範圍為-2147483648~2147483647
- Float資料型態
  - 浮點數就是 1.1, -1.2, -9.8 ….，包含小數點的數字
  - 精度可達到14位小數點

PHP資料型態 – 字串

- 字串（String）就是一連串的字元
- 可以包含文字、數字、符號等，在字串的前後必須加上單引號或雙引號來做為標示，兩者不能混用
- 字串串接運算子是 "." 不是"+"
- 跳脫字元

跳脫字元	意義	跳脫字元	意義
\'	單引號	\b	倒退鍵(backspace)
\"	雙引號	\f	換頁(formfeed)
\\	反斜線(\)	\r	歸位(carriage return)
\n	換行字元	\t	Tab鍵

PHP資料型態 – 布林值

- 布林型別，只有兩種值true/false
  - 通常用在判斷式或運算式是否成立
- PHP中不區分大小寫
  - 指true可寫成True、TRUE、 TRue 、TruE
  - 建議寫true。

PHP強迫型態轉換

型態迫換運算子	說明
(int)、(integer)	強迫轉換成整數
(real)、(double)、(float)	強迫轉換成浮點數
(string)	強迫轉換成字串
(array)	強迫轉換成陣列
(object)	強迫轉換成物件

PHP算術運算子

運算子	描述	範例
加法 (+)	運算元相加	$x+$y, $x+3
減法 (-)	運算元相減	$x-$y, $x-3
乘法(*)	運算元相乘	$x*$y, $x*3
除法(/)	運算元相除	$x/y, $x/3
增加 (++)	運算元之前 (++$x)，會回傳運算元增加 1 後的值; 在運算元之後($x++)， 會回傳運算元加 1 前的值。	假如 $x是 3，那 ++$x 將把 $x 設定為 4 並回傳 4，而 $x++ 會回傳 3 ， 接著才把 $x 設定為 4。
減少 (--)	將運算元減少 1。回傳值的情況與增加運算元 相同。	假如 $x是 3，那 --$x 將把 $x 設定為 2 並回傳 2，而 $x-- 會回傳 3 ， 接著才把 $x 設定為 2。
取餘數 (%)	回傳兩個運算元相除後的餘數。	12 % 5 回傳 2.
指數運算子 (**)	計算以 a 為底的 b 次方， 也就是, a^b	2 ** 3 回傳 8. 10 ** -1 回傳 0.1.

PHP賦值運算子

名稱	簡化的運算子	意義
賦值	$x = $y	$x = $y
加法賦值	$x += $y	$x = $x + $y
減法賦值	$x -= $y	$x = $x - $y
乘法賦值	$x *= $y	$x = $x * $y
除法賦值	$x /= $y	$x = $x / $y
餘數賦值	$x %= $y	$x = $x % $y
指數賦值	$x **= $y	$x = $x ** $y

PHP比較運算子

運算子	描述
等於 (==)	假如運算元等價就回傳 True。
嚴格等於 (===)	假如運算元具有相同型態且等價則回傳 True。
不等於 (!=)	假如運算元不等價就回傳 True。
不等於(<>)	假如運算元不等價就回傳 True。
大於 (>)	假如左方運算元大於右方運算元，回傳 True。
大於或等於 (>=)	假如左方運算元大於或等於右方運算元，回傳 True
小於 (<)	假如左方運算元小於右方運算元，回傳 True。
小於或等於 (<=)	假如左方運算元小於或等於右方運算元，回傳 True

PHP太空船運算子

- Combined Comparison Operator
- PHP7版以後才支援這個運算子

太空船運算子	描述	範例
<=>	可以比較左右兩個運算式的運算元，如果兩個運算元相等就傳回0；左邊大就傳回1；右邊大則傳回-1	$a <=> $b 若$a < $b 回傳-1 若$a == $b 回傳0 若$a > $b 回傳1

PHP邏輯運算子

運算子	使用方法	描述
邏輯 AND (and)	運算式1 and 運算式2	只有在兩個運算元都是 True 時才會回傳 True，否則回傳 false。
邏輯 OR (or)	運算式1 or 運算式2	在兩個運算元有任一個是 True 時就會回傳 True，否則回傳 false。
邏輯 NOT (!)	!運算式	假如單一個運算元能被轉換成 True 時，回傳false ， 不然回傳 true。

位元運算子

運算子	用法	描述
位元 AND	$a & $b	回傳兩個運算元對於每個 bit 做 AND 的結果。
位元 OR	$a | $b	回傳兩個運算元對於每個 bit 做 OR 的結果。
位元 XOR	$a ^ $b	回傳兩個運算元對於每個 bit 做 XOR 的結果。
位元 NOT	~ $a	將運算元中的每個 bit 反轉(1->0,0->1)
左移	$a << $b	將 a 的每個 bit 向左移動 b 個 bits，空餘的位數以 0 填滿。
右移	$a >> $b	將 a 的每個 bit 向右移動 b 個 bits，空餘位數以最高位補滿。

PHP流程控制

- 循序結構
  - 程式是從上到下循序一個任務接著一個任務執行
- 選擇結構
  - 用來檢查條件式，根據結果的值來執行不同的敘述
  - 例如: if...else、switch、match
- 迴圈結構
  - 用來重複執行指定的敘述
  - 例如: for、while、do...while
  - break和continue指令
    - break為跳出迴圈
    - continue為在迴圈內跳過後面的敘述，直接返回迴圈的開頭

PHP選擇結構 – if…elseif

- 條件式要用小括弧包起來
- 執行的動作用大括弧包起來

```
If ( 條件 )
{條件成立時執行的動作;
}else{條件不成立時執行的動作;
}
```

PHP選擇結構 – switch

› switch多選一條件敘述，依照符合條件來執行不同區塊的程式碼

```
switch( 變數 ){
    case ‘某個值’:
        當變數符合某個值時執行的動作;
        break;
    case ‘某個值2’:
        當變數符合某個值2時執行的動作;
        break;
}
```

PHP選擇結構 -match

- Match是PHP8才支援的選擇結構語法
- 可以取代switch和if/elseif多選一條件的功能來指定變數值
- 使用時機為當你有一個想多選一判斷的變數或條件時
- 條件式和條件成立之後的指定變數值用 "=>" 來連接
- Match運算式並不需要'case'和'break'的結構，而是在大括號中使用 ",” 逗點來分隔多個條件
- 預設default是例外條件，表示沒有符合的條件時指定該值

PHP選擇結構 -match

```
$指定變數 = match(判斷變數){
        ‘某個值’ => ‘給指定變數的結果值’,
        ‘某個值2’ => ‘給指定變數的結果值’,
        default => ‘預設給指定變數的值’
}

$指定變數 = match(true){
        ‘條件1’ => ‘給指定變數的結果值’,
        ‘條件2’ => ‘給指定變數的結果值’,
        default => ‘預設給指定變數的值’
}
```

PHP迴圈結構

- for迴圈 → 控制執行次數的迴圈結構
  - for ($i=0; $i < 10; $++){執行的動作;}
- while迴圈 → 用條件式判斷是否進入迴圈執行動作
  - while(條件) {執行的動作;}
- do...while迴圈 → 先進入回圈內執行一次之後再用條件式判斷是否再次進入迴圈
  - do{執行的動作;} while(條件)
- foreach()迴圈 → 通常用來遍歷陣列中的值
  - foreach($陣列變數 as $指陣列元素的變數)

PHP函數

- 函數是將一段具有某種功能或可以重複使用的敘述寫成獨立的程式區塊，供後續呼叫使用
- 使用者自訂函數
  - 使用function來宣告函數

Example

```
function 函數名稱(傳入參數){
        描述;
        return 傳回值;
}
```

PHP時間函數

- time()
  - 返回當前時間的Unix 時間戳
  - time() 函數返回自Unix 紀元（January 1 1970 00:00:00 GMT）起的當前時間的秒數
- mktime(hour, minute, second, month, day, year) 創建日期
  - 返回一個日期的Unix 時間戳

PHP時間函數

- date(format, timestamp)
  - 用來把時間戳格式化為更易讀的日期和時間
  - format是指定時間函數中的時間格式，必要參數
    - 可自訂顯示格式: "Y/m/d"、"Ymd"、"1"、"h:i:sa"
  - timestamp為指定時間點的參數，選擇性參數，若沒有指定則使用伺服器的當前日期/時間

時間單位	意義	時間單位	意義
d	日(01-31)	h	小時(幾點)
m	月份(01-12)	i	分鐘
Y	年(四位數)	s	秒
1	星期幾	a	am/pm

PHP一些常用的內建函數

- echo “string”
  - 在網頁中印出一個字串
- var_dump($var)
  - 返回$var變數的資料型態
- rand($min,$max)
  - 隨機產生一個範圍由$min~$max的亂數
- ceil($num)、floor($num)、round($num)
  - 對一個數值$num做無條件進位、無條件捨去、四捨五入
- count($array)
  - 計算陣列內的元素數量

PHP陣列

- Array 資料型態是一個用來儲存多數值的一個變數
- 可以分成一般索引陣列和使用者自訂索引陣列
- 一般索引陣列
  - $a=array("第一個值","第二個值","第三個值");
- 使用者自訂索引陣列(關聯陣列associative arrays)
  - $a=array(key1=>value1, key2=>value2, key3=>value3);
  - 索引鍵值式不能重複的

PHP索引陣列

- 陣列不需要事先宣告
- 兩種創建索引陣列方法
  - 沒有指定索引值(像python append)
    - $name[]="Allen";
    - $name[]="Kevin";
    - $name[]="Peggy";
  - 可指定索引值來將元素加入陣列
    - $name[3] = "Alex";
- 一維索引陣列取值
  - $array[索引值]

PHP索引陣列

- PHP有提供索引陣列的相關函式
  - 擴充、刪除、取代、反轉等
- 新增元素到陣列中
  - array_unshift($array, $ele) → 插入新元素到陣列最前端
  - array_push($array, $ele) →插入新元素到陣列最末端
  - array_pad($array, $length, $value) → 將陣列以$value填滿至陣列長度為$length

PHP索引陣列

- 刪除或取代陣列內的元素
  - array_unique($array) →刪除陣列中重複的元素
  - array_splice($array, $offset, $len, $arr_rp) → 刪除陣列中從$offset位置開始刪除參數$len個元素，如果有輸入$arr_rp變數，代表刪除的元素要以這個變數的內容來取代
- 排序和反轉陣列
  - sort($array) →排序陣列中的元素
  - array_reverse($array) → 反轉陣列中的元素順序

PHP字串函式

- 常用的字串處理函式
  - strlen($string) → 回傳字串的長度
  - trim($string) →刪除字串前後的空白字元
  - ltrim($string) → 刪除字串開頭的空白字元
  - rtrim($string) → 刪除字串結尾的空白字元
  - strrev($string) → 反轉字串
  - strpos($string, $query) → 搜尋$query在$string中第一次出現的索引位置
  - substr($string, $int) → 從字串$int位置開始取出剩下長度的字串
  - substr_replace($string, $replace, $int) → 從字串$int位置開始用$replace取代原本字串$string的內容

PHP預定變數

變數名稱	描述
$GLOBALS	包含目前執行PHP程式的所有全域變數
$_SERVER	Web伺服器的系統資訊變數
$_REQUEST	取得$_GET、$_POST、和$_COOKIE變數內容的陣列
$_POST	取得HTTP POST方法傳入的表單欄位值的關聯陣列，其鍵值為欄位名稱
$_GET	取得HTTP GET方法傳入的表單欄位值或URL參數的關聯陣列，其鍵值為欄位或參數名稱
$_FILES	取得HTTP POST方法上傳檔案相關資訊的關聯陣列
$_ENV	取得PHP執行時或CGI環境變數的關聯陣列
$_COOKIE	取得HTTP傳遞cookie的關聯陣列

PHP預定變數

- 伺服器的系統資訊變數$_SERVER內常用的資訊

變數名稱	描述
$_SERVER['PHP_SELF']	返回當前執行的PHP檔案名
$_SERVER['GATEWAY_INTERFACE']	返回伺服器使用的CGI的版本
$_SERVER['SERVER_NAME']	返回伺服器的網域名稱或IP位置
$_SERVER['SERVER_SOFTWARE']	返回伺服器的軟體和版本
$_SERVER['REMOTE_ADDR']	返回使用者端的IP位置
$_SERVER['REQUEST_METHOD']	返回使用者請求的方法(GET, PUT, POST)
$_SERVER['REQUEST_TIME']	返回使用者請求的時間
$_SERVER['QUERY_STRING']	返回URL參數的資料

練習

- 使用PHP完成一個找質數並控制印出的程式
- 隨機產生一個亂數(2-30)，找到小於該數的所有的整數質數
  - 例如:亂數產生到5, 會找到2, 3, 5三個質數
  - 判斷一個數是否能被自身小的正整數(除了1和自身)整除.如果能整除則不是質數,否則反之
- 使用PHP來控制在網頁上印出的內容
  - 印出找到的所有質數，但是印出的字體大小為該質數的數值
  - 例如: 假設找到一個質數2，就要在網頁上印出字體大小為2的數字"2"，找到質數3，就要在網頁上印出字體大小為3的數字"3"