



# 網頁程式設計

## PHP程式設計

Instructor: 馬豪尚

# PHP預定變數

變數名稱	描述
\$GLOBALS	包含目前執行 <b>PHP</b> 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 <b>HTTP POST</b> 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 <b>HTTP GET</b> 方法傳入的表單欄位值或 <b>URL</b> 參數的關聯陣列，其鍵值為欄位或參數名稱
\$_FILES	取得 <b>HTTP POST</b> 方法上傳檔案相關資訊的關聯陣列
\$_SESSION	取得 <b>PHP</b> 啟動交談之後儲存的 <b>SESSION</b> 資訊關聯陣列
\$_COOKIE	取得 <b>HTTP</b> 傳遞cookie的關聯陣列



# PHP 表單處理

# HTML 表單

- › 表單很常是用來讓使用者輸入資料，這些資料可以和伺服器互動，例如表單內容填完後可以傳回遠端伺服器 (web server)，像常見的購物車表單。
- › `<form>` 標籤是用來建立一個 HTML 表單，`<form>` 做為表單的容器，裡面還會有不同用途的其他標籤來建構出完整的表單內容。

# HTML 表單 <form> 標籤內的屬性

- › action: 用來指定一個位址 (URL)，這個位址是告訴瀏覽器當 user 按送出表單後，要將表格的內容送去的伺服器位址。
- › method: 用來指定資料傳輸時用的 HTTP 協議，
  - Get: 會將表單資料放在 form action 請求的位址 URL 網址參數列 (URL GET parameters) 上面送出。
  - Post: 會將表單資料放在 HTTP 傳輸封包 body 中送出。post 通常用在表單資料量比較大、有夾帶檔案上傳 (file upload) 或隱私性考慮的資料。

# HTML 表單 <form> 標籤內的屬性

- › target: 用來指定瀏覽器要在何處顯示表單送出後伺服器回應的結果。值有這些選項：
  - \_self: 顯示在表單所在的當前視窗
  - \_blank: 顯示在新頁籤/視窗
  - \_parent: 顯示在上一層的視窗 (用在例如如果表單是放在 <iframe> 中)
  - \_top: 顯示在最頂層的視窗
- › autocomplete: 指示這個表單中的欄位是否啟用瀏覽器自動完成機制。值有這些選項：
  - off: 否
  - on: 是，預設值

# HTML 表單元素

## › <input> 輸入欄位當中可用的屬性

屬性	說明
type	用來指明不同的用法功能
value	用來指定一個預設值
name	用來指定送出去的該筆資料要用什麼名稱，目的是讓遠端伺服器才能透過明確定義好的名稱去取出對應的欄位值
disabled	將元件設定為禁用狀態
readonly	將元件設為唯獨不可更改內容的狀態
autocomplete	是否啟用瀏覽器自動完成功能
autofocus	當頁面載入後，自動聚焦在此欄位上
required	指定為必填欄位

# HTML 表單

## input屬性-type類型

- › text: 文字輸入欄位
  - type 預設上就是 text，所以也可以省略不寫，沒 type 時其實就是 text
- › password: 密碼輸入欄位
  - 和 text 的差別是，使用者輸入的內容不會被明碼顯示在螢幕畫面中
- › placeholder: 提示使用者的訊息
- › submit: 表單的送出按鈕
  - 當使用者點了 submit button 就會送出該表單給遠端的伺服器，用 value 屬性可以設定按鈕名稱。



# PHP取得HTML表單內容

- › HTML網頁表單透過某一個method上傳表單內容
  - Get → 會存在php的 `$_GET` 預定變數中
  - Post → 會存在 php的 `$_POST` 預定變數中
- › `$_GET`和`$_POST`是關聯陣列型態
  - 索引值為包含在表單上傳內容中每一個輸入內容的name屬性的值
  - 其中的元素值就是使用者輸入的內容或是預設的value屬性的值

# GET vs. POST

- › \$\_GET 是通過URL 參數來傳遞所包含的參數到伺服器上
  - 所有參數名稱和值都會顯示在URL 中
  - 用GET 方法從表單傳遞的資訊對任何人都是可見的
  - 發送內容字符的數量也有限制，大約是2000 個字符
  - 絕不能使用GET 來發送密碼或其他敏感資訊
- › \$\_POST 是通過HTTP POST 來傳遞
  - 通過POST 方法從表單發送的内容對其他人是不可見的
  - 所有參數名稱和值會被嵌入HTTP 請求中
  - 發送內容字符的數量無限制

# PHP取得HTML表單內容

## HTML表單範例

```
<form action="welcome_get.php" method="get">  
  Name: <input type="text" name="name"><br>  
  E-mail: <input type="text" name="email"><br>  
  <input type="submit">  
</form>
```

- › \$\_GET=array(name=>使用者輸入名稱, email=>信箱)
- › PHP對關聯陣列取值
  - \$\_GET['name']
  - \$\_GET['email']

# PHP取得HTML表單內容

HTML表單範例

```
<form action="welcome_get.php" method="post">  
  Name: <input type="text" name="name"><br>  
  E-mail: <input type="text" name="email"><br>  
  <input type="submit">  
</form>
```

- › \$\_POST=array(name=>使用者輸入名稱, email=>信箱)
- › PHP對關聯陣列取值
  - \$\_POST['name']
  - \$\_POST['email']

# HTML 表單

## input屬性- type 選擇型輸入

### › checkbox: 核取方塊

- 用來讓使用者勾選某個選項是否成立，可以再搭配 **value** 屬性 (預設值是 "on") 來指定當使用者勾選此方塊時要傳送給遠端伺服器什麼值。

### › radio: 選項按鈕

- 用來處理表單中有多選一時的情況，搭配 **value** 屬性來指定當使用者選取此選項時要傳送給遠端伺服器什麼值。

### › reset: 重設按鈕

- 讓使用者點了可以重設表單內容回到初始狀態，而 **value** 屬性可以設定 **reset** 按鈕的名稱

# HTML 表單

## input屬性- type 選擇型輸入

### › range:數字範圍滑動選取欄位

- 讓使用者用滑動的方式在一個數字區間內選擇出一個值，可以應用在對數字精準度要求不高的場景，像是調整音量大小。
- max: 範圍中可選的最大值
- min: 範圍中可選的最小值
- step: 設定一個數字，用來控制元件數字一次跳動的幅度；或在送出表單之前，瀏覽器會對欄位做驗證，數字需要符合 step 設定的跳動區間

### › color:顏色選擇器

- 用來讓使用者挑選顏色，顏色的格式為 #rrggbb。

# PHP取得選擇型輸入的值

Checkbox輸入表單

```
<form action="form_get.php" method="post">
```

```
  <input type="checkbox" name="subscribe" value="sunshine"> 陽光文教基金會<br>
```

```
  <input type="checkbox" name="subscribe2" value="c-are-us"> 喜憨兒文教基金會<br>
```

```
  <input type="submit">
```

```
</form>
```

- › \$\_GET=array(subscribe=>"sunshine")
- › PHP對關聯陣列取值
  - \$\_GET['subscribe']

# PHP取得HTML的URL參數

Example

```
<body>
```

```
<a href="geturl.php?id=5612&Name=陳允傑">
```

```
登入學校網站</a>
```

```
</body>
```

- › \$\_GET=array(id=>5612, Name=>陳允傑)
- › PHP對關聯陣列取值
  - \$\_GET['id']
  - \$\_GET['Name']



# HTML表單上傳檔案

- › 在form表單裡
  - 加上enctype="multipart/form-data"
- › 在Input 輸入欄位裡
  - 指定type為file
  - accept 屬性可以限制上傳檔案的類型

## Example

```
<form action="file_get.php" method="post" enctype="multipart/form-data">  
  <input type="file" id="file-uploader" accept=".jpg .png">  
  <input type="submit">  
</form>
```

# PHP取得上傳檔案內容

- › 用表單送出的POST請求上傳檔案內容會存在\$\_FILES變數裡
  - \$\_FILES["file"] → file為上傳檔案的名稱
- › 確認檔案是否上傳成功
  - \$\_FILES['my\_file']['error'] → 成功的話會儲存UPLOAD\_ERR\_OK值
- › 取得上傳檔案的資訊
  - \$\_FILES["file"]["name"] → 檔案名稱
  - \$\_FILES["file"]["type"] → 檔案類型
  - \$\_FILES["file"]["size"] → 檔案大小
  - \$\_FILES["file"]["tmp\_name"] → 暫存名稱



# PHP COOKIE/SESSION處理

# PHP 取得 COOKIES

- › HTTP cookie ( web cookie 、 browser cookie ) 為伺服器傳送予使用者瀏覽器的一個小片段資料。
- › 瀏覽器可能儲存並於下一次請求回傳 cookie 至相同的伺服器
- › Cookies 主要應用於
  - Session 管理
    - › 帳號登入、購物車、遊戲分數，或任何其他伺服器應該記住的資訊
  - 個人化
    - › 使用者設定、佈景主題，以及其他設定
  - 追蹤
    - › 記錄並分析使用者行為

# PHP 對於COOKIES的處理

## › 新增cookie

### – setcookie(

string \$name,

string \$value = "",

int \$expires\_or\_options = 0,

string \$path = "",

string \$domain = "",

bool \$secure = false,

bool \$httponly = false

): bool

## › 可以設定很多個cookies

## › 主要使用前3個參數即可

– \$name為cookie的名稱

– \$value為cookie的值

– \$expire為一個數字代表多久  
之後這個cookie會失效，以  
秒為單位

## › Example

– setcookie("TestCookie",  
\$value, time()+3600);

# PHP 對於COOKIES的處理

- › Cookie的值會存在預定變數\$\_COOKIE裡面
- › \$\_COOKIE是一個關聯陣列，其索引值為cookie的名稱
- › 取得其中的值方法為
  - \$\_COOKIE["cookie名稱"];
- › 刪除cookie的方法為將cookie設定為過期就失效了
  - setcookie("cookie的名稱", "", time()-3600)
  - time()為取得現在時間戳記

# PHP 對於Session的處理

- › Session是一個交談期間，代表使用者從進入這個網站到離開的這個過程
- › PHP開啟session之後，PHP引擎會自動指定一個Session ID建立一個新的交談期，可以記錄在這期間該使用者的行為和留下的一些參數
- › 不同使用者進入網站會指定不同的Session ID來記錄，PHP會根據這個ID來判定是否為同一個使用者
- › 這個Session ID可以藉由Cookie儲存在使用者的瀏覽器端，當使用者向伺服器做HTTP的請求時會帶上這個參數

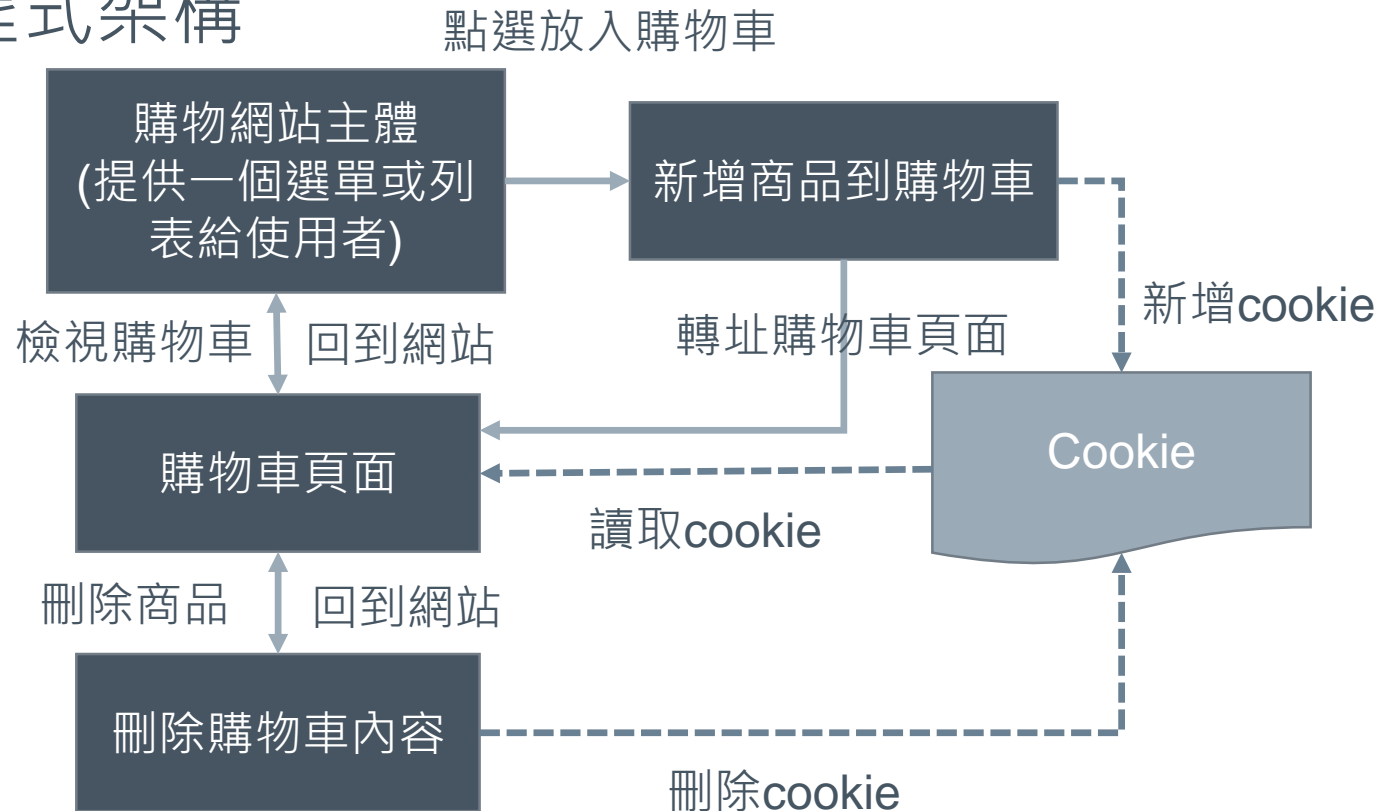
# PHP 對於Session的處理

- › Session\_start()
  - 啟用交談期
- › Session\_destroy
  - 關閉交談期和刪除所有\$\_SESSION變數
- › Session\_id
  - 回傳Session ID
- › Isset(\$\_SESSION[string])
  - 檢查某一個參數是否存在SESSION變數
- › Unset (\$\_SESSION[string])
  - 刪除在SESSION變數中的指定參數



# PHP實作一個購物車

## › 購物車程式架構



# 購物網站主體網頁

- › 這個網頁主要提供商品內容給使用者瀏覽
- › 放置一個選單或表單給使用者操作，新增商品到購物車
- › 送出表單之後給新增商品到購物車的程式處理

# 新增商品到購物車程式

- › 為了記錄使用者的行為開啟Session
- › 取得使用者送出的商品資料
- › 將使用者新增的商品資料增加到cookie裡
- › 轉址到購物車的頁面
  - header("Location: shoppingcart.php")

# 購物車頁面程式

- › 顯示使用者增加至購物車的商品資訊並顯示在這個頁面
  - 讀取cookie中使用者輸入的商品資訊
  - 顯示這些商品的資訊在網頁上
- › 提供刪除的選項給使用者
  - 傳送資料並連結到刪除購物車中商品的程式
- › 提供一個回到購物網站的連結
  - 轉址到購物網站本體

# 刪除購物車內容程式

- › 取得購物車頁面傳過來的商品資訊
- › 將這個商品從cookie移除
- › 轉址回到購物車頁面