Cookie的應用

Cookie 的功能是將資料存放在瀏覽器( 用戶/client ) 端，並且可以此資料來做為識別用戶的機制。網站利用cookie來保存使用者相關資訊，以加快與Web應用程式的連結。

Cookie的主要用途大略如下:

1. 保存個人資訊:
2. 記錄個人偏好設定:
3. 線上購物資訊:

一般瀏器中都有支援cookie物件，但 IE4.0或較早期版本，必須先設定【接受cookie物件】，否則無法使用。

* + Cookie物件資料可以被「修改、刪除或複製」，所以最好不要存放重要的資料。

新增cookie 【 setcookie() 】函數

Setcookie(name[,value[,expire[path[,domain[,secure]]]]]);

參數說明如下:

|  |  |
| --- | --- |
| 參數 | 說明 |
| Name | 設定cookie名稱 |
| Value | 指定cookie的值 |
| Expire | 指定cookie有效期限 |
| Path | 設定在伺服器端可以存取cookie的路徑 |
| Domain | 指定伺服器的網域名稱（此網域下的各子網域皆可存取此cookie） |
| secure | 設定是否只在安全的HTTP連線模式下傳輸(true or false ) |

因為建立cookie時會更動HTTP標頭資料，所以在PHP程式語言中建立cookie時，必須先利用ob\_start() 函數開啟 「輸出暫存器」。Cookie 建立完成後，可以用 ob\_flush() 清除 cookie內容 或 ob\_end\_flush() 送出 cookie資料並關閉 cookie 。

|  |  |
| --- | --- |
| 函數 | 說明 |
| ob\_start() | 開啟cookie暫存器 |
| ob\_flush() | 清除cookie暫存器內容 |
| ob\_end\_flush() | 送出cookie暫存器資料並關閉 cookie |

例.建立一個名稱為linda的cookie，對應的值是：努力存錢!，有效期限是一星期。

ch14-01.php

<?php

//開啟輸出暫存器

ob\_start();

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>建立Cookie</title>

</head>

<body>

<?php

$name="Linda";

$value="努力存錢!";

//設定有效期限為目前的時間加上一星期

$expire=strtotime("+1 week",time());

//建立Cookie

setcookie($name,$value,$expire);

//出清輸出暫存器

ob\_end\_flush();

//顯示目前日期

$now=getdate(time());

$shownow=$now["month"]." ".$now["mday"].", ".$now["year"];

//顯示有效日期

$expday=getdate($expire);

$showexp=$expday["month"]." ".$expday["mday"].", ".$expday["year"];

echo "Cookie的名稱是: ".$name."<p>";

echo "目前的時間: ".$shownow."<p>";

echo "Cookie有效期限: ".$showexp;

?>

</body>

</html>

註：strtotime("+1 week",time());

strtotime(“增加時間” , 目前時間函數 );

讀取cookie資料

有三種讀取cookie資料的指令

$變數 = $\_COOKIE[“cookie名稱”];

~~$變數 = $\_REQUEST[“cookie名稱”];~~

~~$變數 = $HTTP\_COOKIE\_VARS[“cookie名稱”];~~

例:建立三個cookie 資料，再分別以三種不同方法讀取這三種 Cookie 值。

ch14-02.php

<?php

//開啟輸出暫存器

ob\_start();

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>讀取Cookie資料</title>

</head>

<body>

<?php

//設定有效期限為目前的時間加上一星期

$expire=strtotime("+1 week",time());

//建立三個Cookie資料

setcookie("Charles","軟體工程師",$expire);

setcookie("Elvin","經銷業務",$expire);

setcookie("Peter","行政人員",$expire);

//以三種方法讀取Cookie資料

$cookie1=$\_COOKIE["Charles"];

$cookie2=$\_REQUEST["Elvin"];

$cookie3=$HTTP\_COOKIE\_VARS["Peter"];

//清除暫存器內容

ob\_flush();

//顯示有效日期

$expday=getdate($expire);

$showexp=$expday["month"]." ".$expday["mday"].", ".$expday["year"];

echo "Cookies的有效期限到: ".$showexp."<p>";

echo "三筆Cookie資料分別是:<p>";

echo $cookie1.", ";

echo $cookie2.", ";

echo $cookie3;

?>

</body>

</html>

刪除Cookie

有兩種做法

1. 利用 setcookie() 函數，將expire 參數設定己經過去的時間
2. 利用 setcookie() 函數建立相同名稱的 Cookie ，但不設定 Cookie值及參數。

例: 建立兩個 Cookie 資料，並利用 header() 函數切換到 2-3-3end.php程式，在2-3-3end.php程式中判定 Cookie 是否存在，存在就分別修改有效期限來刪除 Cookie ，再以 header() 函數來更新網頁

ch14-03.php

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>建立兩個Cookie</title>

</head>

<body>

<?php

//設定有效期限為目前的時間加上一天

$expire=strtotime("+1 day",time());

//建立Cookie

setcookie("Charles","軟體工程師",$expire);

echo "已經建立Cookie...<p>";

echo "3秒後會自動切換到7-1-3end.php...";

//以header()函數來切換網頁

header("Refresh:3; URL=7-1-3end.php");

//關閉輸出暫存器

ob\_end\_flush();

?>

</body>

</html>

ch14-03end.php

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>刪除Cookie</title>

</head>

<body>

<?php

//判定先前所建立的Cookie是否存在

if(isset($\_COOKIE["Charles"]))

{

echo "目前名稱為Charles的Cookie還存在!<p>";

//將時間設為目前時間減去30天

$newtime=strtotime("-30 days",time());

//設定有效時間已經過期

setcookie("Charles","",$newtime);

//更新畫面

echo "5秒後自動更新此網頁....";

header("Refresh:5");

ob\_end\_flush();

}

else

{

echo "用戶端已經不含名稱為Charles的cookie!";

}

?>

</body>

</html>

註：header()函數說明

格式：header(“參數：對應資料”[,True|False]);

|  |  |  |
| --- | --- | --- |
| 功能 | 參數 | 語法格式範例 |
| 網頁轉向 | Location | Header(“Location:http://tw.yahoo.com”); |
| 更新網頁 | Refresh | Header(“Refresh:5”); |
| 資料型態 | Content-type | Header(“Content-type:text/html”); |
| 保留期限 | Expires | Header(“Expires:Sun, 18 sep 2012 12:00:00”); |

Cookie 陣列（資料）

格式：

Setcookie(“陣列名稱[索引名稱]” , “cookie值”);

例: 設計兩個名稱分別為Charles和Elvin的Cookie陣列，並利用foreach重覆結構去讀取Cookie陣列中的資料。

ch14-04.php

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>Cookie陣列</title>

</head>

<body>

<?php

//設定名稱為Charles的Cookie陣列

setcookie("Charles[color]","blue");

setcookie("Charles[inttime]","PM10:00-12:00");

setcookie("Charles[fruit]","apple");

//設定名稱為Elvin的Cookie陣列

setcookie("Elvin[color]","red");

setcookie("Elvin[inttime]","AM1:00-3:00");

setcookie("Elvin[fruit]","orange");

//設定2秒後更新此網頁

header("Refresh:2");

//讀取Cookie陣列

$user\_Charles=$\_COOKIE["Charles"];

$user\_Elvin=$\_COOKIE["Elvin"];

//清空並關閉暫存器

ob\_end\_flush();

//顯示Cookie陣列中的資料

echo "Charles Cookie陣列中的資料有以下這些:<p>";

//以foreach重覆結構去讀取

foreach($user\_Charles as $item1=> $value1)

{

//顯示陣列的索引和對應的值

echo $item1." => ".$value1."<br>";

}

//加入水平線

echo "<p><hr>";

//顯示Cookie陣列中的資料

echo "Elvin Cookie陣列中的資料有以下這些:<p>";

//以foreach重覆結構去讀取

foreach($user\_Elvin as $item2=>$value2)

{

//顯示陣列的索引和對應的值

echo $item2." => ".$value2."<br>";

}

?>

</body>

</html>

其實，在強調大數據的今天，資料存放在server端已不構成困擾(速度更快，記憶容量更大)，如果將資料放置在client端，不僅有喪失資料的可能，在大數據的統計運作上，也會增加資料擷取的困擾。

Session的應用

網際網路上的網頁是開放給大眾使用，如果有多個使用者在某一個時段都在向server端傳輸資料，為正確區分出每一個使用者的資料，就必須使用Session的功能。

Session（交談）是指使用者從連線到離線的這一段期區。(它會判定瀏覽器(電腦)，在同一瀏覽器(電腦)中，同一Session id只能註冊一次。

Session相關函數指令格式

|  |  |
| --- | --- |
| 函數 | 說明 |
| Session\_id(“字串”); | 指定識別ID，，本指令要在Session\_start之前執行 |
| Session\_start(); | 啟動session |
| $\_SESSION[(“變數名稱”]; | 設定或讀取session變數的值 |
| if(isset($\_SESSION["變數名稱"])) | 判定對應的Session變數是否存在 |

註：Session\_start();函數會傳送session cookie，所以要先在程式的開頭加入開啟cookie暫存器的指令如下：

<?php

Ob\_start();

?>

Session 功能操作分析

會把資料丟到sever端內，他類似cookie差別只在於它是把資料放在sever內，這可有效的減少我們電腦內部的資料。! 在簡單來說就是，他可以是一種暫時存在的資料，他會幫忙把資料整理好，也可以把他想成一個暫時存放的倉庫，他雖然不是真正的資料庫軟體，但總體來說它可以做到類似的事情囉

下列我們以範例來說明 session 應用的各種狀況反應，以供使用者能得心應手的應用本指令

1. **不設定 id (server會自動註冊一個id)**

下面設計二個程式，分別為

Session-noid.php 不設定id

Session-out.php 不指定id(讀取最後一個id所設定的資料)

在不指定ID時，它會自動創造一個，但這樣的創造並不好用，因此在後面我們將會介紹如何創造。

**Session-noid.php**

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf8">

<title>建立session變數</title>

</head>

<body>

<?php

//不指定session\_ID

session\_id();

//啟動session

session\_start();

//echo "此次的session ID = ".session\_id()."<p>";(將//去掉看執行結果)

//分別指定兩個session變數(單一變數)的值

$\_SESSION["uname"]="linda";

$\_SESSION["upass"]="123456";

//設定陣列變數

$name=array("陳大同","張三丰","令狐沖");

$\_SESSION["aname"]=$name;

$score=array(0=>array("陳大同",91,92,93),

1=>array("張三丰",81,82,83),

2=>array("令狐沖",71,72,73));

$\_SESSION["ascore"]=$score;

echo "已經建立相關的session變數<p>";

echo "請切換到 session-out.php....";

//關閉輸出暫存器

ob\_end\_flush();

?>

</body>

</html>

上例中，分別陳述了「單一」及「陣列」變數的 設定 方法

**Session-out.php**

<?php

開啟之後session將會幫忙把內部的資料找出來，當然囉，如果沒有告訴對方我們要找的對象的話，那麼session就會把上一次我們開啟的資料召喚出來。

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf8">

<title>讀取session變數</title>

</head>

<body>

<?php

//不指定id

//啟動session

session\_start();

//讀取session ID(未指定id的情形下，會自動讀取最後一次設定的id)

echo "此次的session ID = ".session\_id()."<p>";

//判定session是否存在

echo "uname這個session變數存在,<br>";

echo "對應的變數值是: ".$\_SESSION["uname"]."<p>";

//判定另一個session是否存在

echo "upass這個session變數存在,<br>";

echo "對應的變數值是: ".$\_SESSION["upass"]."<p>";

//一維陣列變數讀取列印

$name=$\_SESSION["aname"];

for($a=0;$a<3;$a++)

{

echo $name[$a]."　　、";

}

echo "<p>二維陣列變數讀取列印<p>";

for($a=0;$a<3;$a++)

{

for($b=0;$b<4;$b++)

{

echo $\_SESSION["ascore"][$a][$b]."　　、";

}

echo "　<p>";

}

?>

</body>

</html>

本例中，則分別陳述了「單一」及「陣列」變數的讀取方法

**操作測試：**

請開啟任一瀏覽器(例:chromn)

先執行**Session-noid.php(http://localhost/109-2/** **Session-noid.php)**

**可以看到執行結果，畫面如下：**

**沒有指定就是最後一次開啟的喔!**

此次的session ID = nb0k0i3v0f48v6sf73udkrbkim

已經建立相關的session變數

請切換到 session-out.php....

再新增分頁執行**Session-out.php(http://localhost/109-2/** **Session-out.php)**

下圖標示為( server )自動設定的id

此次的session ID = nb0k0i3v0f48v6sf73udkrbkim

uname這個session變數存在,  
對應的變數值是: linda

upass這個session變數存在,  
對應的變數值是: 123456

陳大同　　、張三丰　　、令狐沖　　、

二維陣列變數讀取列印

陳大同　　、91　　、92　　、93　　、

張三丰　　、81　　、82　　、83　　、

令狐沖　　、71　　、72　　、73　　、

說明：在上面例子中，我們所建立的觀念如下：

1. 不管是 單一變數 或 陣列變數 ，透過 $\_SESSION["變數名稱"]=資料/變數/陣列變數; 可以建立session資料
2. session資料的讀取方法為：

　　$變數名稱=$\_SESSION["變數名稱"];

1. session\_id();指令，因為没有指定id名稱，電腦會自動產生一組id名稱，本指令必須放在session\_start();之前。
2. 在瀏覽器未關閉的情況下，若另外開一個標籤分頁執行其他程式，若未指定id( session\_id(); )，則執行 session\_ start (); 時，電腦會以最近一組的 id為操作標的。
3. **當瀏覽器關閉後(或關閉電腦)，id也會自動關閉?**

請關閉瀏覽器後再重啟，再直接執行

**Session-out.php(http://localhost/109-2/** **Session-out.php)**

下圖標示為( server )自動設定的id

此次的session ID = 51cf95cifeo6ie35ojhjra1rvg

uname這個session變數存在,  
  
**Notice**: Undefined index: uname in **C:\xampp\htdocs\109-2\session-out.php** on line **23**  
對應的變數值是:

upass這個session變數存在,  
  
**Notice**: Undefined index: upass in **C:\xampp\htdocs\109-2\session-out.php** on line **28**  
對應的變數值是:

**Notice**: Undefined index: aname in **C:\xampp\htdocs\109-2\session-out.php** on line **31**  
　　、　　、　　、

二維陣列變數讀取列印

**Notice**: Undefined index: ascore in **C:\xampp\htdocs\109-2\session-out.php** on line **41**

因為在**Session-out.php**程式中，我們並没有指定id，所以 ( server )碰到 session\_start(); 指令，又會自動產生一個id (與上一個 id 不同)，所以會找不到指定的id變數 ( $\_SESSION["uname"] )，因此，會讓我們誤以為 id自動關閉(消失)

**註:當然可以在程式中，設定儲存id名稱的指令**

1. **自行設定id**

修改**Session-noid.php中，**如下指令碼

//不指定session\_ID

session\_id();

為

//指定session\_ID

session\_id(“LD”); //設定session\_ID名稱為 “LD”

再另存新檔**Session-id.php**，如下：

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf8">

<title>建立session變數</title>

</head>

<body>

<?php

//指定session\_ID

session\_id("ld");

//啟動session

session\_start();

echo "此次的session ID = ".session\_id()."<p>";

//註冊二個session變數

// session\_register("uname");

// session\_register("upass");

//分別指定兩個session變數的值

$\_SESSION["uname"]="linda";

$\_SESSION["upass"]="123456";

//設定陣列變數

$name=array("陳大同","張三丰","令狐沖");

$\_SESSION["aname"]=$name;

$score=array(0=>array("陳大同",91,92,93),

1=>array("張三丰",81,82,83),

2=>array("令狐沖",71,72,73));

$\_SESSION["ascore"]=$score;

echo "已經建立相關的session變數<p>";

echo "請切換到 session-out.php....";

//3秒後自動切換到另一個網頁

// header("Refresh:3; URL=2-4-1end.php");

//關閉輸出暫存器

ob\_end\_flush();

?>

</body>

</html>

完成後，彷狀況一步驟執行如下：

請開啟任一瀏覽器(例:chromn)

先執行**Session-id.php(http://localhost/109-2/** **Session-id.php)**

下圖標示為( server )指定id的結果

此次的session ID = ld

已經建立相關的session變數

請切換到 session-out.php....

再新增分頁執行**Session-out.php(http://localhost/109-2/** **Session-out.php)**

此次的session ID = ld

uname這個session變數存在,  
對應的變數值是: linda

upass這個session變數存在,  
對應的變數值是: 123456

陳大同　　、張三丰　　、令狐沖　　、

二維陣列變數讀取列印

陳大同　　、91　　、92　　、93　　、

張三丰　　、81　　、82　　、83　　、

令狐沖　　、71　　、72　　、73　　、

現在，我們同樣先關閉瀏覽器視窗後再重啟，再直接執行一次

**Session-out.php(http://localhost/109-2/** **Session-out.php)**

下圖標示為( server )自動設定的id

此次的session ID = ifsmsgno7rvk2ge0m343gld0m6

uname這個session變數存在,  
**Notice**: Undefined index: uname in **C:\xampp\htdocs\109-2\session-out.php** on line **23**  
對應的變數值是:

upass這個session變數存在,  
**Notice**: Undefined index: upass in **C:\xampp\htdocs\109-2\session-out.php** on line **28**  
對應的變數值是:

**Notice**: Undefined index: aname in **C:\xampp\htdocs\109-2\session-out.php** on line **31**  
　　、　　、　　、

二維陣列變數讀取列印

**Notice**: Undefined index: ascore in **C:\xampp\htdocs\109-2\session-out.php** on line **41**

同樣

因為在**Session-out.php**程式中，我們並没有指定id，所以 ( server )碰到 session\_start(); 指令，又會自動產生一個id (與指定的id 不同)，所以會找不到指定的id變數 ( $\_SESSION["uname"] )，因此，會讓我們誤以為 id自動關閉(消失)

1. **將讀取session變數程式也設定為 指定ID**

接下來，我們再修改**Session-out.php**程式，在程式中增加「指定id」的程式碼如下:

<?php

//指定session\_ID

新增程式碼

session\_id("ld");

//啟動session

session\_start();

//讀取session ID

echo "此次的session ID = ".session\_id()."<p>";

再另存新檔**Session-id-out.php**，完成後，在瀏覽器中直接執行，可看到結果如下：

此次的session ID = ld

uname這個session變數存在,  
對應的變數值是: linda

upass這個session變數存在,  
對應的變數值是: 123456

陳大同　　、張三丰　　、令狐沖　　、

二維陣列變數讀取列印

陳大同　　、91　　、92　　、93　　、

張三丰　　、81　　、82　　、83　　、

令狐沖　　、71　　、72　　、73　　、

在上圖中可以看到，指定session\_id 的資料並没有因瀏覽器的關閉而消失。

1. **同一瀏覽器中，執行不同session\_id的程式**

在同一瀏覽器中，是否可以分別同時操作設定不同的session\_id的二支程式?

我們先設計另一程式session-id-x.php程式，程式中只有session\_id不同，其餘所使用的session變數皆相同，程式碼如下：

**session-id-x.php**

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf8">

<title>建立session變數</title>

</head>

<body>

<?php

//指定session\_ID

session\_id("X");

//啟動session

session\_start();

修改程式碼處

//分別指定兩個session變數的值

$\_SESSION["uname"]="文字";

$\_SESSION["upass"]="甲乙丙丁";

//設定陣列變數

$name=array("陳大同","張三丰","令狐沖");

$\_SESSION["aname"]=$name;

$score=array(0=>array("陳大同",91,92,93),

1=>array("張三丰",81,82,83),

2=>array("令狐沖",71,72,73));

$\_SESSION["ascore"]=$score;

echo "已經建立相關的session變數<p>";

echo "請切換到 session-out.php....";

//3秒後自動切換到另一個網頁

// header("Refresh:3; URL=2-4-1end.php");

//關閉輸出暫存器

ob\_end\_flush();

?>

</body>

</html>

現在我們分別依序分別執行

**Session-id.php(http://localhost/109-2/** **Session-id.php)**

**Session-id-x.php(http://localhost/109-2/** **Session-id-x.php)**

**Session-out.php(http://localhost/109-2/** **Session-out.php)**

因為**Session-out.php** 中没有指定 Session\_id 所以瀏覽器會以最後執行的程式(Session-id-x.php)中，所設定的 Session\_id 為 設定值，並取出該程式所設定的$\_SESSION變數資料，如下圖:

此次的session ID = X

uname這個session變數存在,  
對應的變數值是: 文字

upass這個session變數存在,  
對應的變數值是: 甲乙丙丁

陳大同　　、張三丰　　、令狐沖　　、

二維陣列變數讀取列印

陳大同　　、91　　、92　　、93　　、

張三丰　　、81　　、82　　、83　　、

令狐沖　　、71　　、72　　、73　　、

事實上，session\_id(“LD”) 的資料也仍然存在，只要在**Session-id-out.php程式中，指定** session\_id()之值即可，如下:

<?php

//指定session\_ID

修改id值( “LD” 或 “X”)各執行一次

session\_id("ld");

//啟動session

session\_start();

//讀取session ID

echo "此次的session ID = ".session\_id()."<p>";

例: 指定session\_ID 為 “ld”【 session\_id("ld"); 】



例: 指定session\_ID 為 “ld”（上左圖），【 session\_id("X"); 】(上右圖)

1. **在不同瀏覽器中，執行相同session\_id的程式(狀況A：session變數值不同)**

簡單來說:ID註冊是用電腦為主，但是瀏覽器內的資料呀，基本上不會互相影響，例如上一次用google開的id是xx， 接著用ie開的則是 yy下次，用google開的還是依樣是xx喔!

Session 功能是以電腦 (非以瀏覽器) 為主體，因此即使在不同的瀏覽器中，如果操作相同的 session\_id 也會相互干擾

修改**Session-id-x.php** 程式碼，如下:

<?php

//指定session\_ID

session\_id("X");

//啟動session

session\_start();

//分別指定兩個session變數的值

修改 $\_SESSION["uname"]

及 $\_SESSION["upass"]

的資料

$\_SESSION["uname"]="英文字母";

$\_SESSION["upass"]="ABCD";

//設定陣列變數

$name=array("陳大同","張三丰","令狐沖");

$\_SESSION["aname"]=$name;

另存新檔 **Session-id-x-2.php**

先開啟chromn瀏覽器，依序執行

**Session-id-x.php(http://localhost/109-2/** **Session-id-x.php)**

**Session-out.php(http://localhost/109-2/** **Session-out.php)**

結果如下圖:

此次的session ID = X

uname這個session變數存在,  
對應的變數值是: 文字

upass這個session變數存在,  
對應的變數值是: 甲乙丙丁

陳大同　　、張三丰　　、令狐沖　　、

二維陣列變數讀取列印

陳大同　　、91　　、92　　、93　　、

張三丰　　、81　　、82　　、83　　、

令狐沖　　、71　　、72　　、73　　、

接下來開啟另一瀏覽器 ( I.E. )

執行

**Session-id-x-2.php(http://localhost/109-2/** **Session-id-x-2.php)**

已經建立相關的session變數

請切換到 session-out.php....

雖然，我們是在 不同瀏覽器中，執行的程式也不同，但因 所設定的 session\_id 相同，所以 該 id 的變數值自然也會被改變。

現在，回到chromn瀏覽器中，重新執行

**Session-out.php(http://localhost/109-2/** **Session-out.php)**

可以看到 結果改變了，如下:

此次的session ID = X

uname這個session變數存在,  
對應的變數值是: 英文字母

upass這個session變數存在,  
對應的變數值是: ABCD

陳大同　　、張三丰　　、令狐沖　　、

二維陣列變數讀取列印

陳大同　　、91　　、92　　、93　　、

張三丰　　、81　　、82　　、83　　、

令狐沖　　、71　　、72　　、73　　、

1. **不同瀏覽器中，執行相同session\_id的程式**

(　狀況B：不會影响另一瀏覽器中的session\_id 執行順序 )

在前述第五項中，提到 [ 在同一個瀏覽器A中，可分別 執行 不同的 session\_id 程式，而在讀取時，若不指定 id，瀏覽器會自動設定最後執行的 id 為預設id ]

如第六項所談，當在瀏覽器B中，所變更的 指定 session\_id 的程式，只改變相同id的資料，而不會改變在瀏覽器A中的執行順序)

例:

1.在chromn瀏覽器中，依序執行

**Session-id-x.php(http://localhost/109-2/** **Session-id-x.php) 註:id為”X”**

**Session-id.php(http://localhost/109-2/** **Session-id.php) 註:id為”Ld”**

**Session-out.php(http://localhost/109-2/** **Session-out.php) 註:id為”LD”的資料**

2.再在I.E. 瀏覽器中，執行

**Session-id-x-2.php(http://localhost/109-2/** **Session-id-x-2.php)**

此時，再回到chromn瀏覽器中，重新執行

**Session-out.php(http://localhost/109-2/** **Session-out.php)**

畫面並没有改變(因為chromn瀏覽器中的執行順序没有被改變)

我們把**Session-id-out.php 程式碼中的 (id改為”Ld”)，再執行**

**Session-id-out.php(http://localhost/109-2/** **Session-id-out.php)**

才會看到變更的結果。

P.S.：本狀況請自行操作練習

因為session的資料會影响到server(伺服器)的效能，且session的資料通常為階段性(暫時性)的處置資料，所以，一旦session的資料確認並存入資料表後，就應該將session資料清除，以免繼續影响到server(伺服器)的效能。

**取消session的相關指令格式如下**

|  |  |
| --- | --- |
| 函數 | 說明 |
| if(isset($\_SESSION["變數名稱"])) | 判定對應的Session變數是否存在 |
| Session\_destory() | 關閉所有的session資料 |
| Session\_unset() | 取消所有己經註冊的session變數 |

例7-2. 設計一個網頁程式（2-4-2.php）建立三個session變數，並顯示資料，再以session\_unset()取消所有己註冊的變數

**Session-end.php**

<?php

ob\_start(); //開啟輸出暫存器

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>取消session</title>

</head>

<body>

<?php

//指定session ID

session\_id("x");

//啟動session

session\_start();

//分別指定兩個session變數的值

$\_SESSION["uname"]="Charles";

$\_SESSION["upass"]="566";

echo "已經建立相關的session變數<p>";

//取消所有的session註冊

session\_unset();

echo "已執行session\_unset()...<p>";

if(!(isset($\_SESSION ["uname"]) )&& !(isset($\_SESSION ["upass"])))

{

echo "對應的 session 變數已經被取消!<p>";

}else{

echo "有些 session 變數仍未被取消!<p>";

}

//關閉session

session\_destroy();

echo "已經關閉session <p>";

?>

</body>

</html>

下面我們應用session 功能來設計成績系統

程式名稱如下：以系科年班來設定id

**session-gradeA.php**

<html>

<html>

<head>

<title>

成績系統

</title>

</head>

<body>

<h1>成績系統</h1>

請輸入下列項目<br>

<form method="post" action="session-gradeB.php">

系科:<select name="dept">

<option value="資訊管理A" selected>資訊管理</option>

<option value="幼兒B" >幼兒</option>

<option value="化妝品C" >化妝品</option>

</select><br>

<select name="grade">

<option value="一A" selected>一</option>

<option value="二B" >二</option>

<option value="三c" >三</option>

</select>年:

<select name="class">

<option value="甲a" selected>甲</option>

<option value="乙b" >乙</option>

<option value="丙c" >丙</option>

</select>班<br>

科目數:<input type="text" name="counu"><br>

人　數:<input type="text" name="nu"><br>

<input type="submit" name="submit" value="送出">

<input type="reset" name="reset" value="重填">

</form>

</body>

</html>

**session-gradeB.php**

<html>

<head>

<title>

成績系統

</title>

</head>

<body>

<h1>成績系統</h1>

<?php

/\*

報表資料確認

系科變數:$dept , 年級變數:$grade , 班級變數:$class

科目數變數:$counu , 人數變數:$nu

科目名稱變數:couna.$counu. 科目學分變數: cousc.$counu

(陣列變數) (陣列變數)

\*/

//接收表單資料

//系科資料

$dept=$\_POST["dept"];

$dept1=substr($dept, -1); //取得系科代號

$dept=substr($dept,0,-1); //去除系科代號

//年級資料

$grade=$\_POST["grade"];

$grade1=substr($grade, -1); //取得年級代號

$grade=substr($grade,0,-1); //去除年級代號

//班級資料

$class=$\_POST["class"];

$class1=substr($class, -1); //取得班級代號

$class=substr($class,0,-1); //去除班級代號

//科目數資料

$counu=$\_POST["counu"];

//接收人數資料

$nu=$\_POST["nu"];

//列印報表

Echo "<center><h1>中職科技大學</h1>";

Echo "<h2>成績單</h2>";

Echo "科 系:".$dept."<br>";

Echo "年 班:".$grade."年".$class."班<br>";

Echo "科目數:".$counu."<p>";

////////// 設定 session 資料 //////////////////////////////

ob\_start(); //開啟輸出暫存器

//指定session\_ID

$idname=$dept1.$grade1.$class1;

session\_id($idname);

//啟動session

session\_start();

//分別設定「系科、年、班、科目數、人數」的 session變數 值

$\_SESSION["dept"]=$dept; //系科資料

$\_SESSION["grade"]=$grade; //年級資料

$\_SESSION["class"]=$class; //科目數資料

$\_SESSION["counu"]=$counu; //科目數資料

$\_SESSION["nu"]=$nu; //科目數資料

//////////////////////////////////////////////////////////////

//開始進行二次表單資料接收傳送工作

echo "請輸入下列資料<br>";

echo "<form method='post' action='session-gradeC.php'>";

echo "<input type='hidden' name='idname' value='".$idname."'>";

//傳送 session\_id名稱

//列印表格化輸入表單

echo "請輸入各科目名稱及學分:<br>";

Echo "<table border='1'>";

echo "<tr><td>科目編號</td><td>科目名稱</td><td>學分</td></tr>";

for($a=1;$a<=$counu;$a++)

{

Echo "<tr><td>".$a."</td>"; //編號

echo "<td><input type='text' name='couna".$a."' size='4'></td>"; //科目名稱表單

echo "<td><input type='text' name='cousc".$a."' size='4'></td></tr>"; //科目學分表單

}

?>

</table>

<input type="submit" name="submit" value="送出">

<input type="reset" name="reset" value="重填">

</form>

</body>

</html>

**session-gradeC.php**

<html>

<head>

<title>

成績系統

</title>

</head>

<body>

<h1>成績系統</h1>

<?php

/\*

報表資料確認

系科變數:$dept , 年級變數:$grade , 班級變數:$class

科目數變數:$counu , 人數變數:$nu

科目名稱變數:$couna[$couna] ,科目學分變數:$cousc[$couna]

資料形態採 2 維陣列設定，名稱設定為: scor.$x.$y.

$x 代表座號，

$y 表示項目，列示如后:1->姓名、2->科目１、3->科目２、4->科目３...以此類推

\*/

////////// 讀取 session 資料 //////////////////////////////

ob\_start(); //開啟輸出暫存器

//指定session\_ID

$idname=$\_POST["idname"]; //接收id(避免狀況五現象)

session\_id($idname);

//啟動session

session\_start();

//分別設定「系科、年、班、科目數、人數」的 session變數 值

$dept=$\_SESSION["dept"]; //系科資料

$grade=$\_SESSION["grade"]; //年級資料

$class=$\_SESSION["class"]; //班級資料

$counu=$\_SESSION["counu"]; //科目數資料

$nu=$\_SESSION["nu"]; //人數資料

//////////////////////////////////////////////////////////////

//依科目數設定接收科目名稱及學分資料廻圈

for($a=1;$a<=$counu;$a++)

{

$counb="couna".$a; //設定表單的科目名稱

$counna[$a]=$\_POST[$counb];

$counb="cousc".$a; //設定表單的學分名稱

$counsc[$a]=$\_POST[$counb];

}

//////////// 設定科目名稱及學分之 陣列 session 資料 ////////////////

$\_SESSION["counna"]=$counna; //科目名稱

$\_SESSION["counsc"]=$counsc; //學分名稱

//////////////////////////////////////////////////////////////////////

//列印報表

Echo "<center><h1>中職科技大學</h1>";

Echo "<h2>成績單</h2>";

Echo "科系:".$dept."<br>";

Echo "年班:".$grade."年".$class."班<br>";

echo "人數:".$nu."人<br>";

//開始進行二次表單資料接收傳送工作

echo "請輸入下列資料<br>";

echo "<form method='post' action='session-gradeD.php'>";

echo "<input type='hidden' name='idname' value='".$idname."'>"; //session\_id名稱

//列印表格化表單

Echo "<table border='1'>";

Echo "<tr><td>座號</td><td>姓名</td>";

//列印科目名稱欄位

for($a=1;$a<=$counu;$a++)

{

echo "<td>".$counna[$a]."(".$counsc[$a].")</td>";

}

echo "</tr>";

//列印分數資料接收表單

for($x=1;$x<=$nu;$x++)

{

echo "<tr><td>".$x."</td>";

for($a=1;$a<=($counu+1);$a++) //1為姓名 科目位置加1

{

echo "<td><input type='text' name='scor".$x.$a."' size='4'></td>";

}

echo "</tr>";

}

?>

</table>

<input type="submit" name="submit" value="送出">

<input type="reset" name="reset" value="重填">

</form>

</body>

</html>

**session-gradeD.php**

<?php

/\*

報表資料確認

系科變數:$dept , 年級變數:$grade , 班級變數:$class

科目數變數:$counu ， 人數變數:$nu

科目名稱變數:$couna[$couna] , 科目學分變數:$cousc[$couna]

班級資料設為二維陣列變數:$scor[$x][$y]

[$x]代表座號，

[$y]表示項目，列示如后:1->姓名、2->科目１、3->科目２、4->科目３...以此類推

資料獲得方式確認

透過表單接收:

系科變數:$dept , 年級變數:$grade , 班級變數:$class

科目變數:$course

班級各項資料設為二維陣列變數:$scor[$x][$y]，$y為1:姓名 ,2~$counu+1 為科名稱,$counu+2:總分 $counu+3:平均

$counu+4:名次 $counu+5:實得學分

\*/

////////// 讀取 session 資料 //////////////////////////////

ob\_start(); //開啟輸出暫存器

//指定session\_ID

$idname=$\_POST["idname"]; //避免狀況五現象

session\_id($idname);

//啟動session

session\_start();

//分別設定「系科、年、班、科目數、人數」的 session變數 值

$dept=$\_SESSION["dept"]; //系科資料

$grade=$\_SESSION["grade"]; //年級資料

$class=$\_SESSION["class"]; //班級資料

$counu=$\_SESSION["counu"]; //科目數資料

$nu=$\_SESSION["nu"]; //人數資料

$counna=$\_SESSION["counna"]; //科目名稱

$counsc=$\_SESSION["counsc"]; //學分名稱

//////////////////////////////////////////////////////////////

//依科目數設定接收 學生各科成績 資料廻圈 並 運用此廻圈計算 總學分 數

$ttsc=0; //預設 總學分數 之 變數初值

for($a=1;$a<=$counu;$a++)

{

$ttsc+=$counsc[$a]; //總學分數 之累加

}

// 接收班級各項分數資料

for($x=1;$x<=$nu;$x++) //$x代表座號，$nu為人數資料

{

$scor[$x][$counu+2]=0; //設定總分初值 $counu+2 為總分欄位

$scor[$x][$counu+5]=0; //設定總分初值 $counu+4 為實得學分欄位

for($a=1;$a<=($counu+1);$a++) //$a代表: 1 姓名,2~$counu+1 各科目分數

{

$midata="scor".$x.$a; //產生對應表單名稱 [scor11]~[scor(n1)(n2)]

$scor[$x][$a]=$\_POST[$midata]; //接收對應學生姓名及成績表單名稱的資料

//計算總分(累加) 及 實得學分

if($a<>1)

{

$scor[$x][$counu+2]+=$scor[$x][$a]\*$counsc[$a-1]; //總分累加(各科分數\*學分) 註:學分陣列變數索引值比分數索引值少1

if($scor[$x][$a]>=60)

{

$scor[$x][$counu+5]+=$counsc[$a-1]; //實得學分累加 註:學分陣列變數索引值比分數索引值少1

}

}

}

$scor[$x][$counu+3]=(int)($scor[$x][$counu+2]/$ttsc+0.5); //計算平均分數,$counu+3 為平均欄位

}

//設定名次(名次必須等所有學生的總分全部計算完成,才能比較)

for($x=1;$x<=$nu;$x++) //$x代表 比較者 座號，$nu為人數資料

{

$scor[$x][$counu+4]=1; //未比較前先設定名次初值為1(第 1 名)

for($y=1;$y<=$nu;$y++) //$x代表 被比較者 座號，$nu為人數資料

{

if($scor[$x][$counu+2]<$scor[$y][$counu+2])

{

$scor[$x][$counu+4]++; //當$y的分數比較高時，$x的名次就要加1(退步1名)

}

}

}

//列印報表

Echo "<center><h1>中職科技大學</h1>";

Echo "<h2>成績單</h2>";

Echo "科系:".$dept."<br>";

Echo "年班:".$grade."年".$class."班<br>";

echo "人數:".$nu."人<p>";

Echo "<table border='1'>";

//列印標題

Echo "<tr><td>座號</td><td>姓名</td>";

//列印科目名稱欄位

for($a=1;$a<=$counu;$a++)

{

echo "<td>".$counna[$a]."(".$counsc[$a].")</td>";

}

echo "<td>總分</td><td>平均</td><td>名次</td><td>實得學分</td></tr>";

//1~5號同學

for($x=1;$x<=$nu;$x++)

{

echo "<tr><td>".$x."</td>";

for($a=1;$a<=($counu+5);$a++)

{

if(($a<>1) and ($a<>$counu+4) and ($a<>$counu+5) and ($scor[$x][$a]<60)) //第1欄為姓名,$counu+4欄為名次,$counu+5欄為實得學分

{

echo "<td><font color='red'>".$scor[$x][$a]."</font></td>";

}

else

{

echo "<td>".$scor[$x][$a]."</td>";

}

}

echo "</tr>";

}

Echo "</table>";

?>

上面這套範例在操作時，必須一次完成；如果中途關視窗或有其他人換大手操作，則可能會幾個困擾：

1. 操作到相同「班級」，會產生相同的session\_id() 導致session資料改變
2. 如果是不同班級，則最新的 session\_id() 的名稱會被變更，原使用者必須重新操作
3. 如關閉視窗再重啟，也是要重新操作

**單元作業1**：輸入 教師帳號，並以教師帳號為SESSION\_ID()，再以 $\_session[“step”] ( 操作步驟 )變數設定應用來記憶教師上次操作階段，使教師可以中途離開工作，再進入時，只要輸入帳號即可進行後續作業：(se-grades-step-O.php、se-grades-step-A.php、se-grades-step-B.php、se-grades-step-C.php、se-grades-step-D.php)

**單元作業2**：應用session功能購物車功能畫面如下：

|  |  |  |
| --- | --- | --- |
| ＸＸ購物廣場 | | |
| 服飾  皮件  保養品  健康食品 | 健康食品示區（銷售畫面）   |  | | --- | |  | |  | |  | |  | |  |  |  | | --- | |  | |  | |  | |  | |  |  |  | | --- | |  | |  | |  | |  | |  |   送出 | 購物車  確認 |

說明：

1. 點選 menu選單，可切換銷售物品畫面
2. 點 送出 鈕，該畫面的購物資料會進入購物車清單中
3. 在未確認前，可隨時更換銷售畫面，進行「購物內容之更改」
4. 每次送出，購物車清單內容要重新核對(新增、修改、刪除)，顯示最新購物清單內容。
5. 點 確認 鈕，購物車清單資料傳送至server進行「新增購物資料至訂購資料表」(先做資料接收確認)。