

Cookie 與 Session

使用者在瀏覽網頁時並不是一直與伺服器保持在連線的狀態下，事實上當瀏覽者送出需求到伺服器端處理後將結果回傳顯示，就已經結束與伺服器的連線。所以當需要新資料或是更新顯示內容時，都必須重新載入頁面或是重新送出需求。

為什麼要使用 Cookie 與 Session？

Cookie 與 Session 的存在就是為了解決網站不能保存狀態的問題。以一般網站最常見的會員系統來說，當會員以帳號密碼登入系統的同時，程式可以有二個方式來記住登入會員的資料：一個方法是在登入者的電腦中放入一個小檔案來記憶，這個就是 Cookie；另一個方法是在伺服器的記憶體產生一個空間來記憶，這個就是 Session。

那什麼時候要使用 Cookie，什麼時候要使用 Session 呢？以下我們將說明他們之間的差異與比較，並提供您使用時機上的建議：

關於 Cookie

Cookie 是由伺服器傳送並儲存在至瀏覽者電腦中的小檔案，可以用來識別使用者的身份或是相關資訊。因為是放置在用戶端的電腦，瀏覽者不必與伺服器溝通即可取得其中的資訊，免除與伺服器之間多餘的連線。例如瀏覽者將未結帳的商品放置在購物車中，中途可能因故離開或是關閉瀏覽器，都能藉由 Cookie 的幫忙在下一次回到原網站操作時，調出未結帳的商品繼續購物。

cookie 的儲存每個瀏覽器不一樣，也就是說當你用 FireFox 儲存了一個網站的 cookie，在 IE 瀏覽器上是沒辦法讀取的，無法跨瀏覽器讀取。

關於 Session

Session 會將使用者資訊儲存在伺服器端暫存檔中，儲存的位置依照 <php.ini> 的設定，如：

```
session.save_path="C:\Windows\Temp"
```

當瀏覽者進入網站伺服器瀏覽時，在 Session 的開啟狀態下即會開始記錄使用者所賦予的資訊，一直到關閉瀏覽器才結束 Session 的使用。

Cookie 是將狀態資料記錄在用戶端電腦的技術，當瀏覽者開啟網站時即可在程式的設定下將指定的資料儲存在用戶端電腦中，並可以設定該資料的有效時間、存放路徑與有效網域。以下我們將介紹如何存取 Cookie 與設定 Cookie 的有效時間。

Cookie 的使用

由於相對於 session，cookie 由於是儲存在用戶端電腦，所以被竊取的風險比較高，只要有心人士到使用者的電腦，前往瀏覽器的跟目錄儲存 cookie 的資料夾，就可以把這些檔案取走，所以重要的資料應該盡量採用 session 來儲存。

- 目前每個瀏覽器最多只能儲存 300 個 cookie。
- 每個瀏覽器對單一網站只能儲存 20 的 cookie。
- 用戶端電腦的 cookie 只要關閉就沒辦法使用。

自從 PHP 推出 session 之後，cookie 大部分就用來搭配著 session 使用了，一方面比較安全，另一方便也可以避免網站某些功能不能使用的窘境，兩者搭配使用也可以降低一些伺服器的負荷，接著我們要一步一步的完成 cookie 的啟用、讀取與刪除。

1.儲存 Cookie 資料

您可以使用 setcookie() 函式將資料存入 Cookie 中，其格式如下：

```
setcookie(變數名稱, 變數值, [存活時間], [路徑], [網域], [安全性])
```

變數名稱可自訂，存活時間就是你希望這個 cookie 在用戶端電腦儲存多久，請使用時間參數 time()+秒

```
setcookie("user", "php");
```

```
setcookie("user", "php", time()+3600); // 有效時間為1個小時
```

```
setcookie("user", "php", time()+3600, "/tmp/", "example.com", 1);
```

2.讀取 Cookie 的資料

您可以利用 \$_COOKIE 變數的方式讀出 Cookie 中儲存的資料，其格式如下：

```
$_COOKIE[名稱]
```

```
echo $_COOKIE["user"];
```

3.Cookie 陣列

當多個 Cookie 同屬一個性質的資料時，可以利用陣列的方式儲存。設定的基礎格式為：

```
setcookie(陣列名稱[索引鍵], 值)
```

```
$_COOKIE[陣列名稱][索引鍵]
```

```
setcookie("school[no]", 1);  
setcookie("school[name]", " 高雄應用科大");
```

```
echo $_COOKIE["school"]["no"];  
echo $_COOKIE["school"]["name"];
```

Cookie 的有效時間

當您關閉瀏覽器再重新開啟瀏覽器執行該程式，會發現原來儲存的 Cookie 值已經消失，程式又必須重新加入。這是因為我們並沒有在加入 Cookie 時設定它的有效時間，原來的 Cookie 已經在所屬的瀏覽器關閉後自動刪除而消失了。如果想要保持 Cookie 的存在，就必須設定 Cookie 的有效時間。

存活時間就是你希望這個 cookie 在用戶端電腦儲存多久，請使用時間參數 time()+秒

```
setcookie("user", "php", time()+3600); // 有效時間為1個小時
```

可利用mktime()或strtotime()來指定Cookie的到期時間

```
setcoolie("user", "php", mktime(23,59,59,12,31,2016));
```

```
setcoolie("user", "php", strtotime("2016-12-31-23-59-59"));
```

輸出所有 cookie

```
print_r($_COOKIE);
```

刪除 Cookie

要如何刪除存在用戶端的 Cookie 呢？我們都知道 Cookie 在有效期間過後即會自動刪除消失，如果將 Cookie 的有效時間指定為立刻到期，是否就可以完成刪除 Cookie 的動作？沒錯，若將 Cookie 的有效時間指定在當下，最好是過去的時間，那麼該 Cookie 就會馬上失效刪除了。

```
setcookie("user", "", time()-3600);
```

```

<!DOCTYPE html>
<?php
    $cookie_name = "user";
    $cookie_value = "John";
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30),
"/"); // 86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>

```

```

if(isset($_COOKIE['countVisit'])){
    $countVisit = $_COOKIE['countVisit'];
    $countVisit ++;
} else {
    $countVisit = 1;
}
if(isset($_COOKIE['lastVisit'])){
    $lastVisit = $_COOKIE['lastVisit'];
}
// set cookie for countVisit
setcookie('countVisit', $countVisit, time()+3600);
// set cookie for last visit
setcookie('lastVisit', date("d-m-Y H:i:s"), time()+3600);

if($countVisit == 0){
    echo "Welcome";
} else {
    echo "This is your visit number ".$countVisit;
    echo '<br>';
    echo "Last time, you were here ".$lastVisit;
}

```

Session 的使用

Session 是瀏覽者與伺服器連線的工作期間所保持的狀態，它的使用時間是在開啟瀏覽器後進入啟動 Session 機制的網站開始，無論其中是否瀏覽同網站其他頁面，甚至離開該網站去瀏覽別的網站或頁面，只要瀏覽器沒有關閉，回到原網站時您會發現原來的 Session 仍然有效。

要注意的是使用一個瀏覽器開啟一個網站即代表一個 Session，即使同一個使用者開啟多個瀏覽器檢視同一個網站，都算是不同的 Session。但是要注意的，在新一代瀏覽器使用標籤的概念下，在同一個瀏覽器以不同標籤瀏覽同一個網站，就只能算是一個 Session。

Session 的運作原理

當使用者使用瀏覽器連線到伺服器網站時，網站伺服器會自動派發一個 SessionID 給這次的連線動作，網站程式即可依照這個 SessionID 分辨使用者來處理所儲存的狀態。事實上：在預設的狀態下，Session 會將伺服器所派發的 SessionID 以 Cookie 的方式儲存在用戶端來記錄狀態，同一個網站的不同網頁可以藉由這個 Cookie 的記錄來維持同一個 SessionID 的狀態。這個 Cookie 並沒有到期時間，所以預設會在瀏覽器關閉時同時消失。

存取 Session 資料

啟動 Session

在 PHP 中要使用 Session，必須在操作前以 `session_start()` 函式啟動頁面 Session 的功能，即可在以下的頁面中操作 Session 的功能。

儲存並取得 Session 的資料

啟動 Session 之後，我們就可以使用 `$_Session` 變數的方式去設定或取得 Session 的資料，其格式如下：

```
$_SESSION["名稱"]
```

```
session_start();  
$_SESSION["name"]="John";
```

```
echo $_SESSION["name"];
```

```
session_start();  
$_SESSION["people"]["id"]=1;  
$_SESSION["people"]["name"]="John";
```

以session製作計數器

```
<?php  
    header('Content-Type:text/html; charset=utf-8');  
    session_start();  
    if(empty($_SESSION['counter'])) {           //empty: 判斷變數是否為空值  
        // 為空值  
        //echo '沒有值';  
        $_SESSION['counter'] = 1;  
    } else {  
        // 不為空值  
        //echo '有值';  
        //$_SESSION['counter'] = $_SESSION['counter'] + 1;  
        //$_SESSION['counter']+=1;  
        $_SESSION['counter']++;  
    }  
    echo $_SESSION['counter'];  
  
?>
```

Session 的有效時間

Session預設的有效時間，基本上就是在瀏覽器開啟後宣告 Session 啟動開始，一直到關閉瀏覽器為止。但是實際上，其實 Session 還是有其使用期限。您一定常遇到登入了某個網站的會員系統，可能中途有事離開座位，等到回到工作崗位重新操作該網站時，畫面會顯示您已經被登出的訊息，需要重新執行登入的動作。

在php.ini中預設session的有效期限值為

```
session.gc_maxlifetime=1440
```

大約24分鐘($24 \times 60 = 1440$)

刪除 Session

若您想要讓 Session 失效，其實只要關閉瀏覽器即可。但若在瀏覽的過程中，想要刪除目前網站上的 Session，該如何做呢？最常見的狀況是當使用者登入了某個網站的會員系統，正在使用會員功能在編輯資料、收發信件或是購物消費時突然要離開座位去處理其他的事情，為了維護使用者的資訊安全，最好的方式是先登出系統，等回到座位時再重新登入繼續操作。登出會員系統的動作即是刪除登入時所產生的 Session。

```
unset($_SESSION[ 名稱])
```

若要刪除目前所有的Session

```
session_unset()
```

範例

login.php

```
<?php
if(isset($_POST["Submit"])){
    // 如果帳號和密碼正確的話，寫入Session變數，並視情況重導到相關的頁面
    if($_POST["account"] == "admin" && $_POST["password"] == "pass"){
        // 啟動 Session
        session_start();

        // 寫入 Session 變數值
        $_SESSION['authenticated'] = true;
        $_SESSION['fruits'] = array("orange", "banana", "apple");
        $_SESSION['login_time'] = date('Y-m-d h:i:s');
        // 檢查在 $_SESSION 全域變數中，是否有之前設定好的網址重導 Session 變數
        if(isset($_SESSION["UrlRedirect"])){
            $redir = $_SESSION["UrlRedirect"];
        }else{
            $redir = 'login_success.php';
        }

        // 重導到相關頁面
        header("Location: $redir");
        exit;
    }else{
        header('Location: access_denied.htm');
        exit;
    }
}

?>

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

<body>
<form name="form1" id="form1" method="post" action="">
<p align="center">帳號:
<input name="account" type="text" id="account" />
</p>
<p align="center">密碼:
<input name="password" type="password" id="password" />
</p>
<p align="center">
<input type="submit" name="Submit" value="登入" />
</p>
</form>
</body>
</html>
```


login_success.php

```
<?php
    session_start();
    if($_SESSION["authenticated"] == true){
        $result = 登入時間: '$_SESSION["login_time"]';
        $result.='<br>URL: '$_SESSION["UrlRedirect"];
        foreach($_SESSION["fruits"] as $fruit){
            $result .= "<br>$fruit";
        }
    }else{
        header('Location: login.php');
        exit;
    }

?>

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

<body>
<div align="center">
<p><font color="#0000FF"><strong>登入成功!</strong></font></p>
<p>Session 變數如下: </p>
<p><?php echo $result?></p>
<p><a href="logout.php">[ 登 出 ]</a></p>
</div>
</body>
</html>
```

protected.php

```
<?php
    session_start();

    $_SESSION["UrlRedirect"] = $_SERVER['SCRIPT_NAME'];
    if($_SESSION["authenticated"] == true){
        $result = 登入時間: '$_SESSION["login_time"]';
        foreach($_SESSION["fruits"] as $fruit){
            $result .= "<br>$fruit";
        }
    }else{
        header('Location: login.php');
        exit;
    }

?>

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

<body>
<div align="center">
<p><font color="#0000FF"><strong>權限控管頁面 !</strong></font></p>
<p>Session 變數如下 : </p>
<p><?php echo $result?></p>
<p><a href="logout.php">[ 登 出 ]</a></p>
</div>
</body>
</html>
```

logout.php

```
<?php
    session_start();
    session_unset();
    session_destroy();
    header('Location: index.htm');
    exit;

?>
```