Linguaggi di Programmazione per il Web – Parte 5

PHP – Hypertext Preprocessor

I Cookie e le Sessioni

Autore

Prof. **Rio Chierego**
riochierego@libero.it

Siti Utili

http://www.riochierego.it/mobile

http://www.html.it/

http://www.mrwebmaster.it

http://www.php.net/

http://php.html.it/

I COOKIE e le SESSIONI

- I **cookie** e le **sessioni** e rispondono alla necessità di **memorizzare in modo più o meno permanentemente alcuni dati e informazioni durante la navigazione dell'utente in un sito in modo da essere riutilizzati successivamente in altre pagine**.

Alcuni esempi classici:

- un utente dopo aver fatto correttamente il login risulterà essere un "utente loggato"; occorrerà "salvare" tale dato in modo da per poter essere accertato in tutte le altre pagine che desideriamo proteggere;
- un utente sceglie come lingua di visualizzazione del sito l'inglese e, da quel momento, tale scelta deve essere permanente anche nelle successive pagine.
- in un sito di e-commerce un utente navigando potrà aggiungere e rimuovere prodotti dal carrello; ognuna di queste operazioni devono essere registrate in modo da poter avere conoscenza di ciò che l'utente ha nel carrello in un determinato momento.

I COOKIE

Il **cookie** è il metodo per memorizzare, **sul computer dell'utente**, delle informazioni che vogliamo persistano anche nelle successive visite al nostro sito.

La **mole di tali dati deve essere rigorosamente limitata**;ad esempio un username, impostazioni di tipo booleano (vero/falso) o di preferenza dell'utente (ad esempio la lingua preferita), etc..

I dati vengono **salvati IN CHIARO in formato stringa**
(eventualmente serializzati/deserializzati con la funzione PHP **serialize() ed unserialize()**).

I **cookie** sono registrati dal browser in una specifica directory in formato di file di testo (secondo modalità diverse dei vari browser).

I **cookie** in molti applicativi possono tornare molto utili data la loro possibilità di salvare dati su pc dell'utente in modo durevole ma tuttavia soffrono di importanti implicazioni legati alla sicurezza.
Infatti, i cookie **non sono adatti per salvare dati sensibili** come password o dati personali.

I COOKIE

Teniamo presente che questi dati verranno **salvati in chiaro** (non criptati) in un file sul computer dell'utente e, quindi se letti, possono rivelare informazioni riservate.

Quindi singolo utente **potrà autoimpostarsi il valore** eventualmente diverso rispetto a ciò che noi desideriamo, nonchè eliminarlo: caso tipico è la rimozione di un **ban** basato sui cookie.

I **cookie** vengono cancellati automaticamente solo in due casi:
- alla loro scadenza, che decidiamo noi esplicitamente,
- oppure se l'utente li cancella manualmente attraverso le impostazioni del suo browser

I COOKIE

L'uso dei **cookie** in PHP è estremamente semplice anche perché avviene mediante un'unica funzione **setcookie()** che prevede **6** parametri (solo i primi 2 OBBLIGATORI):

- Il **primo** parametro è il nome della variabile di tipo cookie.
- Il **secondo** è il valore della variabile.
- Il **terzo** è specifica quando il cookie verrà cancellato dal client, cioè la data di scadenza; in mancanza si eliminerà alla chiusura del browser; In genere si usa la struttura *time()+n* dove n è il numero di secondi dopo cui scade il cookie
- Il **quarto** specifica la cartella del server per il quale il cookie sarà valido; se non verrà specificato il cookie sarà valido solo nell'ambito della directory in cui si trova la pagina che lo ha inviato; per essere valido in tutte le cartelle del sito si indicherà "/".
- Il **quinto** ci indica il dominio per il quale il cookie è valido e di default si intende il dominio completo del server.
- Il **sesto** può assumere valore 0 e 1 e assume di default assume valore pari a 0; se pari a 1 specifica che il cookie potrà essere inviato soltanto tramite https (ossia attraverso un protocollo di trasferimento sicuro).

I COOKIE – Impostazione

```
<?php
$nome_cookie  =  "test_cookie";
$valore_cookie  =  “Sono un cookie!";
$scadenza_cookie  =  time() + 86400;       //validità un giorno ossia 24x60x60 secondi
$dominio_cookie = “localhost";
setcookie($nome_cookie, $valore_cookie, $scadenza_cookie, "/", $dominio_cookie, 0);
                                                    oppure
setcookie($nome_cookie, $valore_cookie);  //scade quando si chiude il browser
?>                                                  //valido nella directory della pagina che lo ha inviato
                                                    //valido in tutto il dominio del server
                                                    // 0 ultimo parametro
```

Una volta impostato, i dati all'interno di un cookie questo sarà accessibile da PHP ricorrendo una variabile all'interno dell'array associativo denominato **$_COOKIE**

**IMPORTANTE: L'unica raccomandazione che occorre fare è che questa funzione deve essere impiegata prima che la pagina generi qualsiasi tipo di output (attraverso sia le funzioni PHP echo() o print() sia direttamente con i tag HTML anche eventualmente mettendo un rigo bianco tramite il tag
).**

I COOKIE – Lettura

Una volta impostato un cookie, questo può essere letto esattamente come si fa per una variabile. Usiamo la funzione PHP ***isset()*** per vedere se un cookie è stato impostato.

Leggiamo il cookie che abbiamo impostato prima nell'esempio.

```
<?php
if (isset($_COOKIE[$nome_cookie ]))     //sottinteso  == true
      {
      echo "Il cookie vale: " . $_COOKIE[$nome_cookie ] ;
      }
else
     {
     echo "Il cookie non è stato impostato";
     }
?>
```

I COOKIE – Modifica e eliminazione

La **modifica** di un cookie avviene semplicemente reimpostandolo.

Supponendo di voler modificare il precedente cookie fatto come esempio si avrà:

```
<?php
$nome_cookie  =  "test_cookie";
$valore_cookie  =  “Nuovo valore cookie!";
setcookie($nome_cookie, $valore_cookie);
?>
```

Per l'**eliminazione** si possono usano due escamotage:

- si reimposta il cookie con una data del passato (il suo valore resta)
- oppure si pone a **null** suo valore (il suo valore si perde)

```
<?php
 setcookie($nome_cookie, $valore_cookie, time()-3600);
Oppure                    //utilizzando, se necessario, i default per tutti gli altri parametri
setcookie($nome_cookie, null);
?>
```

LE SESSIONI

La logica sulla quale si basano le sessioni può essere così descritta in sintesi:

- tutte le volte che un utente, tramite browser, effettua una visita ad un sito quest'ultimo assegna a tale utente un **id di sessione** detto **SID** (ossia **S**ession **ID**) il quale verrà salvato sul pc dell'utente all'interno di un **cookie (PHPSESSID)**;

- durante tutta la navigazione, l'utente avrà sul suo pc tale **cookie** e **costituirà un identificativo univoco della sessione**

- una volta avviata la sessione potranno essere create sul server (in un **file di testo** che avrà come nome **l'id di sessione** generalmente contenuto nella *cartella tmp*) **variabili di sessione** utilizzando la variabile **$_SESSION** che è un' array associativo a cui possibile attribuire le chiavi che desideriamo

N.B. Quindi, mentre i cookie sono salvati sul pc dell'utente, le sessioni salvano i dati sul server servendosi comunque di un cookie, il SID. Ne consegue che se si hanno i cookie disattivati, i dati di sessione non saranno comunque accessibili a meno di non passare in tutte le pagine del sito il SID

LE SESSIONI

La prima cosa da fare se vogliamo lavorare con le sessioni è impostare nel file di configurazione del PHP ("php.ini") la direttiva **session.save_path**, indicando la directory nella quale verranno salvate le informazioni sulle sessioni dei nostri utenti.

La funzione da utilizzare poi all'interno delle nostre pagine .php per creare una sessione è **session_start()**. Questa funzione non prevede parametri.

N.B. La funzione **session_start() deve essere necessariamente utilizzata prima dell'invio di output**: nella parte precedente del nostro files .php non deve pertanto essere già stato scritto ed inviato del codice HTML (o altro tipo di output) - ad esempio attraverso l'uso delle funzioni PHP **echo()** o **print()** - il quale comprometterebbe il buon esito della nostra funzione.

N.B. Se i cookie non sono abilitati, allora l'identificativo univoco di sessione (SID) verrà inserito automaticamente negli URL, per trasmettere la sessione da una pagina all'altra.

LE SESSIONI – salvataggio di dati

```
<?php
//Apro la sessione e...
session_start();

//Recupero username e password dal form
$username = $_POST['user'];
$password = $_POST['pass'];

//Salvo i dati nella sessione
$_SESSION['username'] = $username;
$_SESSION['password'] = $password;

?>
```

A questo punto abbiamo salvato all'interno della nostra sessione (grazie alla variabile superglobale **$_SESSION**) due diversi valori: **username** e **password**.

LE SESSIONI – utilizzo di dati

```
<?php
//Apro la sessione e...
session_start();

//Recupero i dati dall'array $_SESSION
$username = $_SESSION['username'];
$password = $_SESSION['password'];

//Utilizzo i dati contenuti nella sessione
echo "Ciao " . $username . " la tua password è " . $password;
?>
```

E' possibile recuperare (grazie alla variabile superglobale **$_SESSION**) i due diversi valori: **username** e **password** salvati in precedenza

(vedi **PPT-5-esempio-1-sessioni.php** e **PPT-5-esempio-2-sessioni.php**)

LE SESSIONI – distruzione dei dati

Per eliminare una specifica variabile di sessione useremo:

```
unset ($_SESSION['username']);
```

oppure per eliminare tutte le variabili di una sessione useremo:

```
session_unset () ;
```

Per eliminare tutte le variabili di sessione useremo:

```
$_SESSION = array() ;
```

Per distruggere tutto il file delle sessioni

```
session_destroy () ;
```

LE SESSIONI – distruzione dei dati

La differenza tra le funzioni session_unset() e session_destroy()

-**Funzione sessione_destroy() :**

TUTTO IL CONTENUTO DELL'ARRAY $_SESSION VERRA' DISTRUTTO SOLO DOPO L'USCITA DELLO SCRIPT IN CUI VIENE ESEGUITA.

QUINDI IL CONTENUTO DELL'ARRAY $_SESSION (OSSIA LE VARIABILI DI SESSIONE) RESTA ANCORA DISPONIBILE FINCHE' LO SCRIPT E' IN ESECUZIONE.

- Funzione sessione_unset() :

TUTTO IL CONTENUTO DELL'ARRAY $_SESSION VIENE INIZIALIZZATO (DUNQUE AZZERATO) IMMEDIATAMENTE NEL CORSO DELLO SCRIPT.

QUINDI IL CONTENUTO DELL'ARRAY $_SESSION NON E' PIU' DISPONIBILE NELLO SCRIPT IN ESECUZIONE SIN DALLA PRIMA ISTRUZIONE DOPO LA SUA CHIAMATA.