

Corso di PHP

6.1 – PHP e il web 1

M. Malatesta 6.1-PHP e il web1-04

1
24/03/2017

Prerequisiti

- Conoscenza HTML
- Tecnica della programmazione
- Principi di programmazione web
- Array associativo

M. Malatesta 6.1-PHP e il web1-04

2
24/03/2017

Introduzione

In questa Unità illustriamo alcuni strumenti di programmazione di PHP che intervengono nella programmazione web.

In particolare, si esaminano alcune variabili di sistema, dette **variabili superglobali**, che consentono di agire sul server, sui cookie e sul client.

Variabili superglobali

L'interprete PHP, durante il processo di inizializzazione, crea delle variabili speciali (chiamate **variabili superglobali**) che contengono diversi tipi di informazione utili quando si è in ambiente web.

Queste variabili sono accessibili da qualunque script ed hanno nomi riservati che non andrebbero sovrascritti se non in casi particolari. Le variabili in questione sono:

Variabili superglobali	
\$GLOBALS	Accesso a tutte le variabili globali nello script
\$ _SERVER	Accesso alle variabili del web server
\$ _GET	Valori passati allo script tramite URL
\$ _POST	Valori passati allo script tramite post (HTTP)
\$ _COOKIE	Valori passati dal cookie dell'utente
\$ _FILES	Valori passati tramite HTTP Post File Uploads
\$ _ENV	Valori passati allo script tramite l'ambiente
\$ _REQUEST	Valori passati attraverso URL, HTTP Post, Cookies dell'utente
\$ _SESSION	Valori passati tramite la sessione utente

Variabile \$GLOBALS []

È un array contenente tutte le variabili che risultano globali nello scope corrente, ossia tutti i riferimenti a:

- \$_SERVER[]
- \$_GET[]
- \$_POST[]
- \$_COOKIE[]

Nella prossima Unità si esaminano le variabili:

- \$_FILES[]
- \$_ENV[]
- \$_REQUEST[]
- \$_SESSION[]

M. Malatesta 6.1-PHP e il web1-04

5
24/03/2017

Variabile \$GLOBALS []

È un array che contiene i valori di tutte le variabili globali ed è utilizzabile ovunque (entro e fuori dalle funzioni).

L'interprete memorizza le variabili globali nell'array associativo **\$GLOBAL []** usando come indice il nome delle variabili.

```
<!-- globals.php -->
```

```
<?php
```

```
$x = 75;
```

```
$y = 25;
```

```
function addition()
```

```
{ $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y']; }
```

```
addition();
```

```
echo $z;
```

```
?>
```

M. Malatesta 6.1-PHP e il web1-04

6
24/03/2017

Variabile \$_SERVER []

È un array contenente variabili relative al web server, o al browser dell'utente o al suo indirizzo IP. Vi sono molte variabili associate a questo array e pienamente descritte nella documentazione ufficiale di PHP. Le più utili sono:

- **REQUEST_URI**: il percorso completo dello script
- **PHP_SELF** : il nome del file dello script correntemente in esecuzione;
- **DOCUMENT_ROOT**: la root da cui viene eseguito lo script corrente;
- **REMOTE_ADDR**: l'indirizzo IP dell'utente che sta eseguendo lo script. Questo valore viene impostato in base ad un header dal browser, quindi non andrebbe utilizzato come discriminante in situazioni di sicurezza critiche;

M. Malatesta 6.1-PHP e il web1-04

7
24/03/2017

Variabile \$_SERVER []

```
<!--VariabiliServer.php-->
<?php
    echo $_SERVER['REQUEST_URI'], "<BR>";
    echo $_SERVER['PHP_SELF'], "<BR>";
    echo $_SERVER['DOCUMENT_ROOT'], "<BR>";
    echo $_SERVER['REMOTE_ADDR'], "<BR>";
?>
```

M. Malatesta 6.1-PHP e il web1-04

8
24/03/2017

Variabile \$_SERVER []

Alcuni esempi di variabili utilizzabili con \$_SERVER[]

Server	
\$ _SERVER['SERVER_SOFTWARE']	Il server
\$ _SERVER['SERVER_NAME']	Nome del server
\$ _SERVER['SCRIPT_FILENAME']	Percorso e nome dello script
\$ _SERVER['SCRIPT_NAME']	Nome dello script
Informazioni del client	
\$ _SERVER['REQUEST_METHOD']	Riporta se si era scelto GET o POST
\$ _SERVER['REMOTE_ADDR']	Indirizzo browser richiedente (è l'isp con si è connessi)
Variabili server_php	
HTTP_COOKIE_VAR['PHPSESSID']	Contiene il valore dell'ID di sessione
HTTP_SERVER_VAR['HTTP_COOKIE']	Contiene il valore di tutti i cookie
HTTP_SERVER_VAR['HTTP_HOST']	Contiene il nome dell'host su quale risiede il server web
HTTP_SERVER_VAR['REMOTE_ADDR']	Contiene l'indirizzo remoto del browser (sicuramente dell'isp)
HTTP_SERVER_VAR['SCRIPT_FILENAME']	Contiene il nome e il percorso completo dello script corrente
HTTP_SERVER_VAR['SERVER_ADMIN']	Dà l'indirizzo e-mail inserito nel file di configurazione
HTTP_SERVER_VAR['SERVER_NAME']	Contiene il nome del server web
HTTP_SERVER_VAR['SERVER_SOFTWARE']	Contiene il nome del server web

M. Malatesta 6.1-PHP e il web1-04

9
24/03/2017

Variabile \$_GET[]

È un array contenente tutti i parametri passati alla pagina tramite il metodo **GET**.

I parametri vengono scritti accodandoli all'URL, dopo un punto di domanda (?), e sono rappresentati mediante assegnazioni del tipo

nome = valore

Le singole assegnazioni sono separate dal simbolo &.

Esempio:

http://127.0.0.1/1-Introduzione/Somma.php?a=3&b=2

M. Malatesta 6.1-PHP e il web1-04

10
24/03/2017

Variabile \$_GET[]

```
<!-- form-get.html -->
<HTML>
<HEAD><TITLE>Prova form</TITLE>
<HEAD>
<BODY>
  <FORM name="prova" ACTION=form-get.php METHOD=GET>
    <INPUT NAME ="username" TYPE = "text" SIZE="25"
      MAXLENGTH ="25" >
    <INPUT NAME ="email" TYPE="text" SIZE="25" MAXLENGTH="25"
    <INPUT TYPE =submit NAME="invio">
  </FORM>
</BODY>
</HTML>
```

Un esempio di form che utilizza le due variabili *username* e *email*, passate con metodo **GET**

M. Malatesta 6.1-PHP e il web1-04

11
24/03/2017

Variabile \$_GET[]

```
<!-- form-get.php -->
<?php
  echo "<br>";
  echo "<br>";
  echo "<b>Username: </b>" .$_GET["username"]."<BR>";
  echo "<b>e-mail.....: </b>" .$_GET["email"]."<BR>";?>
?>
```

Lo script preleva mediante l'array **\$GET []** le variabili del form.

M. Malatesta 6.1-PHP e il web1-04

12
24/03/2017

Variabile \$_POST[]

L'array **\$POST[]** è un array contenente tutti i parametri passati alla pagina tramite il metodo **POST**.

L'utilizzo è analogo a quello dell'array **\$_GET[]**, con la differenza che i valori dei parametri non sono mostrati sulla barra degli indirizzi.

M. Malatesta 6.1-PHP e il web1-04

13
24/03/2017

Variabile \$_POST[]

```
<!-- form-post.html -->
<HTML>
<HEAD><TITLE>Prova form</TITLE>
<HEAD>
<BODY>
  <FORM name="prova" ACTION=form-post.php METHOD=POST>
    <INPUT NAME="username" TYPE="text" SIZE="25"
      MAXLENGTH="25" >
    <INPUT NAME="email" TYPE="text" SIZE="25"
      MAXLENGTH="25" >
    <INPUT TYPE = submit NAME="invio">
  </FORM>
</BODY>
</HTML>
```

Un esempio di form che utilizza le due variabili *username* e *email*, passate con metodo **POST**

M. Malatesta 6.1-PHP e il web1-04

14
24/03/2017

Variabile \$_POST[]

```
<!-- form-post.php -->
<?php
    echo "<br>";
    echo "<br>";
    echo "<b>Username: </b>".$_POST["username"]."<BR>";
    echo "<b>e-mail.....: </b>".$_POST["email"]."<BR>";?>
```

Lo script preleva mediante l'array **POST []** le variabili del form.

Si osservi che nella barra degli indirizzi non compaiono le variabili che il form passa allo script.

M. Malatesta 6.1-PHP e il web1-04

15
24/03/2017

I cookie

A volte conviene legare logicamente le pagine web, affinché possano condividere informazioni (cosa indispensabile, ad esempio, in un sito di vendita).

I **cookie** sono piccoli file che il server crea e conserva nel client per:

- identificare il client, ogni volta che invia una richiesta al server, abbinando i cookie alla richiesta (una specie di *tessera di riconoscimento*, con le credenziali dell'utente);
- salvare dati per un uso successivo (ad es. salvare su un cookie un valore indicante l'ultima pagina visualizzata, in modo da ridirigervi l'utente al successivo collegamento).

Se il browser è impostato in modo opportuno, il client può rifiutare i cookie.

M. Malatesta 6.1-PHP e il web1-04

16
24/03/2017

Variabile \$_COOKIE[]

A livello di programmazione, l'array associativo **\$_COOKIE[]** contiene tutte le coppie (*nome, valore*) dei cookie validi nella pagina corrente con il rispettivo valore.

Sui cookie sono possibili le seguenti operazioni:

- creazione
- lettura
- eliminazione

M. Malatesta 6.1-PHP e il web1-04

17
24/03/2017

Variabile \$_COOKIE[] - creazione cookie

I cookie vanno creati prima del tag <HTML>, mediante la funzione:

```
bool setcookie (name, value[, expire][, path][, domain]);
```

dove:

- *name* è il nome del cookie
- *value* è il valore (interi o stringhe) assegnato al cookie, memorizzato nel client
- *expire* indica la durata del cookie in secondi (0 indica validità fino a chiusura del browser)
- *path* è il percorso del cookie (in genere la directory corrente)
- *domain* indica il dominio in cui lo si vuole rendere disponibile

M. Malatesta 6.1-PHP e il web1-04

18
24/03/2017

Variabile \$_COOKIE[]

- creazione cookie

<!-- crea-cookie.php -->

<?php

setcookie ("user", "Mario Rossi", **time()**+3600); // crea cookie

echo \$_COOKIE["user"];

Questo cookie dura un'ora

// visualizza cookie

?>

<HTML>

codice HTML

</HTML>

Per versioni precedenti la 4.1.0 il recupero del cookie avveniva mediante la chiamata

`$HTTP_COOKIE_VARS['user']`

Il tempo si può impostare anche come

<?php

setcookie ("user", "Mario Rossi", **time()**+60*60*24*30);

?>

che equivale a un mese (60 sec * 60 min * 24 ore * 30 giorni)

M. Malatesta 6.1-PHP e il web1-04

19
24/03/2017

Variabile \$_COOKIE[]

- lettura cookie

<!-- contaVisite1.php -->

<?php

if (isset(\$_COOKIE['visita']))

{ \$visita = \$_COOKIE['visita'] + 1; }

else

{ \$visita = 1; }

setcookie ("visita", \$visita);

echo "Pagina visitata ".\$visita." volte";

?>

I cookie possono essere utilizzati per realizzare un contatore di visite alla pagina, come mostrato nell'esempio.

M. Malatesta 6.1-PHP e il web1-04

20
24/03/2017

Variabile \$_COOKIE[]

- eliminazione cookie

I cookie possono essere cancellati mediante la stessa funzione **setcookie()**,
impostando un tempo negativo.

Esempio:

<!--elimina-cookie.php -->

<?php

```
    setcookie("user", time()-50);    // elimina cookie
```

```
    echo $_COOKIE["user"];           // visualizza cookie?>
```

?>

M. Malatesta 6.1-PHP e il web1-04

21
24/03/2017

Variabile \$_COOKIE[]

- osservazioni

OSSERVAZIONI:

- È buona norma non creare troppi cookie, dato che i browser hanno un limite sia relativo ad uno specifico dominio, che ad una specifica cartella. In caso fosse necessario mantenere molti valori, è preferibile salvarli su database o file e salvare nel cookie una chiave che ne permetta l'accesso nelle pagine successive.
- I cookie sono facilmente recuperabili e quindi leggibili, per cui è importante non salvare mai informazioni private o riservate, salvo preventiva criptazione.

M. Malatesta 6.1-PHP e il web1-04

22
24/03/2017

Argomenti

- Variabili superglobali
- Variabile `$_GLOBALS` []
- Variabile `$_SERVER` []
- Variabile `$_GET` []
- Variabile `$_POST` []
- I cookie
- Variabile `$_COOKIE` []
 - creazione cookie
 - contatore visite
 - eliminazione cookie
 - osservazioni

M. Malatesta 6.1-PHP e il web1-04

23
24/03/2017