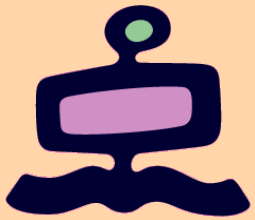


InformaticaUmanistica

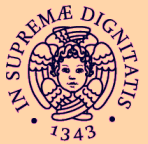
Mantenimento dello stato

Laboratorio Progettazione Web

AA 2009/2010

Chiara Renso

ISTI- CNR - c.renso@isti.cnr.it



UNIVERSITÀ DI PISA

Mantenere le informazioni

Abbiamo visto come il passaggio di informazioni (parametri) tra le pagine possa avvenire tramite le FORM e quindi con la opportune variabili predefinite.

Tramite l'uso della form passiamo i parametri da una pagina HTML alla pagine di gestione delle form, ma poi? Cosa succede per il resto della navigazione utente?

Il Web con il protocollo HTTP è *stateless*, mentre le applicazioni necessitano di mantenere i dati inseriti per tutta la navigazione dell'utente (sessione).

In alcuni casi è sufficiente il passaggio di pochi parametri tra due pagine, in altri casi il valore di alcune variabili deve essere accessibile da tutte le pagine che compongono l'applicazione (ad es. carrello della spesa). *Ricordiamo che l'ambito delle variabili in PHP è locale allo script.*

Mantenere lo stato

intendiamo con "stato" l'insieme dei valori risultanti dall'interazione dell'utente con l'applicazione.

Abbiamo quattro possibilità :

- **Campi Hidden delle form – passati da pagina a pagina**
- **Tramite querystring nella URL – passati da pagina a pagina**
- **Cookies – file di testo sul PC dell'utente, globale all'applicazione**
- **Variabili di Sessione – memorizzate sul server realizzate come cookies, globali dalla sessione utente**

Campi Hidden

Le form prevedono dei campi speciali invisibili all'utente ma che vengono trattati a tutti gli effetti come campi di una form e quindi sono in grado di passare come parametro il loro valore

<INPUT type=HIDDEN name="nascosto" value="miapass">

In PHP verrà trattato come un qualsiasi altro parametro e il valore passato in questo caso sarà "miapass".

Con questo metodo si riescono a passare in genere pochi parametri e per poche pagine.

Occorre creare una campo hidden (quindi una form) in ogni pagina in cui il parametro deve essere passato

Campi Hidden : Esempio

Una pagina php riceve da una form il valore del login e vuole passarlo ad una terza pagina

```
<?php  //questa pagina riceve un valore di
login da una form
$login = $_REQUEST["login"];  ?>
<FORM name="…",
action="ricevilogin.php", method=GET>
….
<INPUT type=hidden value="<? echo
$login ?>" name="log">
<INPUT type=submit>
</FORM>
```

ricevilogin.php

```
<?php
$loginricevuto = $_GET["log"]

echo $loginricevuto;

?>
```

QueryString

Il passaggio avviene tra due pagine, usando la tecnica della querystring e quindi il metodo GET, qui usato senza la FORM, in modo esplicito

```
<?php // passaggio del valore della variabile $my_id

//  tra le pagine current.php e next.php

$my_id="ciao"; ?>

<A HREF="next.php?my_id=<? echo $my_id; ?>">Next</A>
```

QueryString: esempio next.php

Il passaggio avviene tra due pagine, usando la tecnica della querystring e quindi il metodo GET, qui usato senza la FORM, in modo esplicito

```
<?

$my_id=$_GET[‘my_id’];

echo $my_id;

?>
```

QueryString: Esempio

Una pagina php riceve dalla queryString il valore di uno o piu' variabili, si usa il metodo GET

current.php

```
<?php
$my_id="ciao"; ?>

<A HREF="next.php?id=
    <?php echo $my_id; ?>">
Next
</A>
```

next.php?id="ciao"

next.php

```
<?php

$identifier=$_GET['id'];
echo $identifier;

?>
```

La funzione urlencode(URL) permette il passaggio di caratteri speciali nella url come spazi o accenti

Cookies

- Meccanismo che permette al server di memorizzare e poi reperire informazioni dal client.
- E' tipico della applicazioni web – ad esempio i siti che si "ricordano di noi" quando torniamo, hanno memorizzato un cookies sul nostro browser.
- La durata di un cookie può essere variabile e può durare anche per molte sessioni (molte visite al sito)
- In pratica l'applicazione web scrive un file nel file system del client che contiene le informazioni sulla visita dell'utente

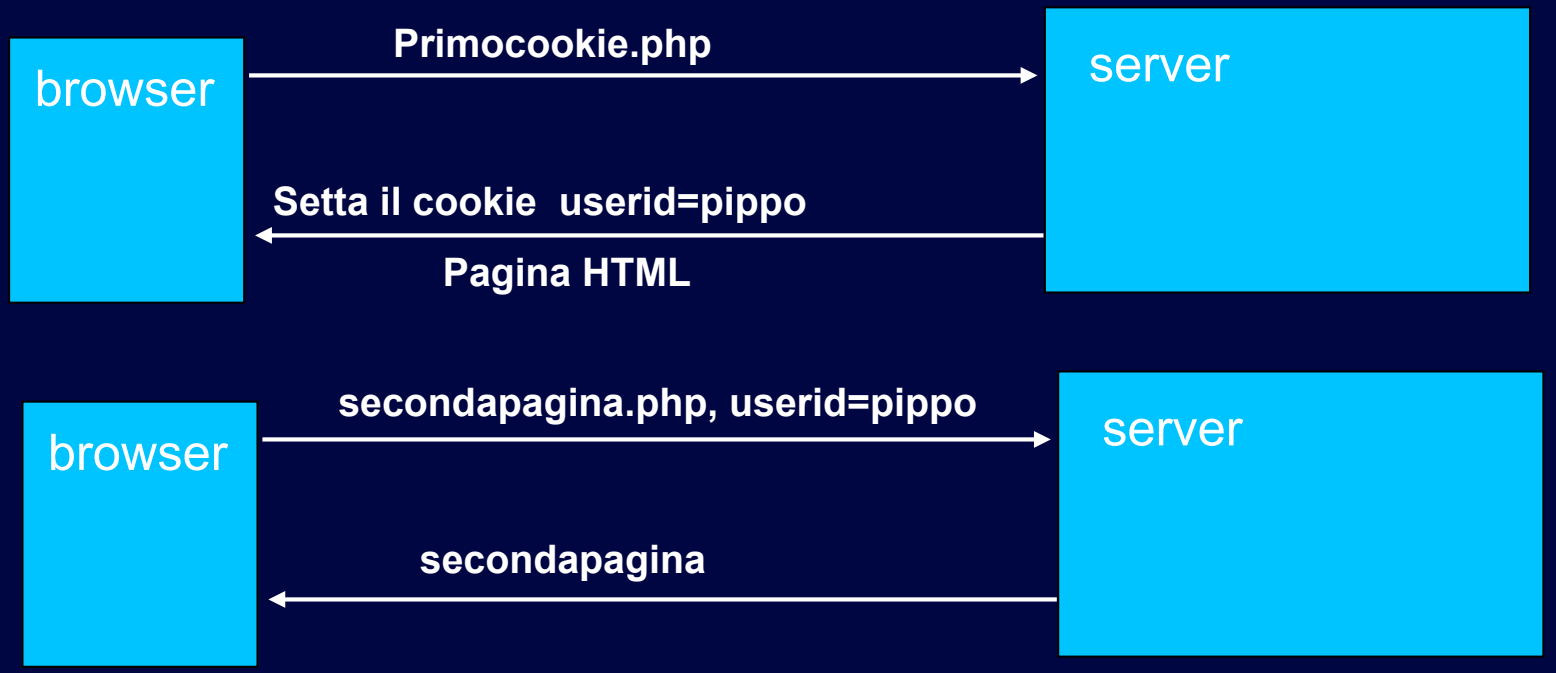
Cookies

- In PHP i cookies si possono settare, modificare e cancellare con l'istruzione setcookie()
- Importante: questa istruzione deve apparire nella pagina php prima di qualsiasi istruzione di stampa – quindi anche tag anche HTML. Un cookies settato in una pagina sarà visibile solo al caricamento successivo (i cookie vengono inviati nello header secondo il protocollo HTTP)
- Esempio:
  setcookie("utente","miosito");
  definisce un cookie di nome "utente" a cui assegna il valore "miosito".

Esempio: pagina primocookie.php

```
<? setcookie('userid','pippo'); ?>

<HTML> <HEAD>..... Questa è la pagina che setta il
    cookie .....  </HTML>
```



Cookies

Il valore di un cookie si può ottenere dall'array $_COOKIE ["nomecookie"]

```
<?

setcookie("utente","pippo");

echo $_COOKIE['utente'];

?>
```

Attenzione, notare che il valore del cookie sarà visibile solo al successivo caricamento della pagina!

Il valore dei cookies è sempre visibile tramite la funzione phpinfo();

Eliminazione di un cookie

Un cookie può essere eliminato dandogli una precisa scadenza:

setcookie(“user”,”pippo”,time()+3600)

Quindi non esisterà più dopo un’ora. Si può eliminare dando una scadenza anteriore oppure semplicemente con

setcookie(“user”);

Sessioni

- La sessione è un concetto astratto che in HTTP non esiste. Si definisce sessione una visita di un utente ad un sito web.
- Le sessioni vengono quindi simulate a programma e funzionano assegnando ad un utente un ID univoco ogni volta che accede al sito.
- Le sessioni vengono tipicamente realizzate con un cookie (PHPSESSID) e permettono di mantenere valori assegnati ad una sessione utente. Spariscono al terminare della sessione.
- Esempi: carrello della spesa nei siti di commercio elettronico, login e password dopo l'autenticazione, preferenze dell'utente...

Sessioni

- La sessione si attiva con l'istruzione session_start(), che deve essere sempre presente nella pagina PHP quando si usano le sessioni, quindi sia per definire una variabile che per accederne al valore.
- Session_start() deve essere nella parte iniziale della pagina php (prima della prima istruzione di stampa, come per i cookies)
- Le variabili di sessioni si memorizzano del global array $_SESSION[]
- Una variabile di sessione si cancella con l'istruzione unset (ad. es. unset($miavar);).
- Tutte le variabili di sessione posso essere cancellate contemporaneamente (logout) con la funzione session_destroy();

Sessioni: esempio

```
<? // definizione di una variabile di sessione e stampa del
    valore

session_start();

$_SESSION['login']="pluto";

echo $_SESSION[‘login’];

?>
```

Sessioni: Esempio

```
<?php  //conteggio degli accessi ad una pagina

session_start();

if (!isset($_SESSION[“count”]))

$_SESSION[“count”]=0;

   else

$_SESSION[“count”]++;

echo “hai visitato questa pagina $_SESSION[count] volte”;

?>
```

Architettura Server-Side