Programmazione lato server

Cookies e Sessioni

Cookies

" … a cookie is a bit of text, containing some unique information, that web servers send in the HTTP header. The client's browser keeps a list of cookies and web sites. When the user goes back to a web site, the browser will automatically return the cookie, provided it hasn't expired … "

Cookies

" … Lou Montulli, who wrote the cookies specification for Navigator 1.0, says there's nothing particularly amusing about the origin of the name: 'A cookie is a well-known computer science term that is used when describing an opaque piece of data held by an intermediary. The term fits the usage precisely; it's just not a well-known term outside of computer science circles.' … "

www.amazon.co.uk

Formato di un cookie

- Un cookie è una stringa di testo formata da diverse parti (separate da ;), alcune opzionali
  - ✓ name = <VALUE>; ← obbligatorio
  - ✓ expires = <DATE>;
  - ✓ path = <PATH>;
  - ✓ domain = <DOMAIN_NAME>;
  - ✓ secure

HTTP Response header: Set-Cookie

- Un cookie viene scritto sul client se il server include l'header Set-Cookie come parte di una risposta HTTP

```
Set-Cookie:
customer=vasco%20rossi;
domain=.spesaclick.it;
expires=Wednesday, 09-Jun-04
```

HTTP Request header: Cookie

- Quando un utente torna su un sito che ha già visitato e che gli ha "lasciato" un cookie, il suo browser invia automaticamente il cookie (la coppia name = <VALUE>) come parte della sua richiesta HTTP

```
Cookie: customer=vasco%20rossi;
```


File cookies.txt

```
# HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file!  Do not edit.
# To delete cookies, use the Cookie Manager.

.amazon.co.uk     TRUE    / FALSE  1052438410 session-id     026-2097198-8031661
.amazon.co.uk     TRUE    / FALSE  1052438410 session-id-time        1052438400
.amazon.co.uk     TRUE    / FALSE  2082758410 ubid-acbuk     432-3309289-1033068
.netscape.com     TRUE    / FALSE  1082621236 sampler        1051862839
.google.it        TRUE    / FALSE  2147368456 PREF   ID=3498e49f0b1d012e:LD=it:…
www.amazon.co.uk  FALSE   / FALSE  1052109350 seenpop        1
.amazon.co.uk     TRUE    / FALSE  2082758413 x-acbuk        qv28EGxbUrrEtGQ@…
.internet.com     TRUE    / FALSE  1293839599 RMID   82c0ef273eb23a30
.php.net          TRUE    / FALSE  1083403463 LAST_LANG      en
```


File cookies.txt

```
.amazon.co.uk  TRUE  /  FALSE  1052438410  session-id  026-2097198-8031661…
```

Leggendo da destra a sinistra:
domain – dominio che ha creato e che può leggere il cookie
flag – TRUE/FALSE indica se tutte le macchine all'interno di un dominio possono leggere il cookie
path – il camminio all'interno del dominio per cui il cookie è valido
secure - TRUE/FALSE
expiration - UNIX time, ovvero il numero di secondi trascorsi a partire Jan 1, 1970 00:00:00 GMT
name – il nome del cookie
value – il valore del cookie

Alcune limitazioni

- Max 300 cookie su ogni client
- Max 4 kilobyte per cookie
- Max 20 cookie dallo stesso server (o dominio)

Scrivere un cookie in PHP

```
setcookie (string name, string value,
           int expire, string path,
           string domain, int secure)

Esempio: setcookie("mycookie", "valore");
```


Leggere un cookie in PHP

Quando l'utente ritorna sul sito che ha scritto il cookie mycookie, il suo valore potrà essere letto

1) Nell'array $_COOKIE["mycookie"]
2) Nell'array $HTTP_COOKIE_VARS["mycookie"]
3) Nella variabile $mycookie

Cancellare un cookie in PHP

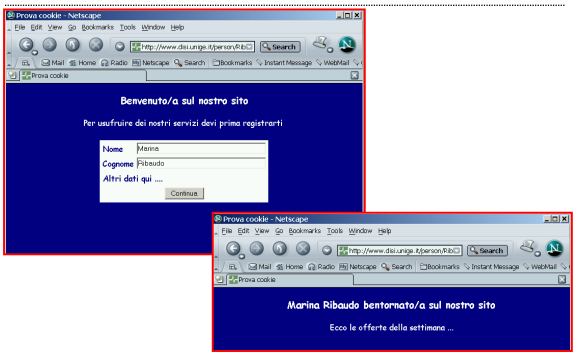
Per cancellare un cookie

1) si può portare la sua data di scadenza al passato

oppure

2) si può assegnare al cookie il valore nullo

Esempio

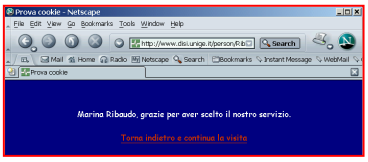


Esempio 1 in rete

Esempio (cookie1.php)

```
<?php
// verifico l'esistenza del cookie
if ($_COOKIE["mycookie"])
{
 <leggo il valore del cookie>
 <costruisco un messaggio personalizzato>
}
else
{
 <modulo per la registrazione>
}
?>
```


Esempio



Questo file PHP, oltre a restituire un messaggio di feedback per l'utente, crea il cookie

Esempio (cookie2.php)

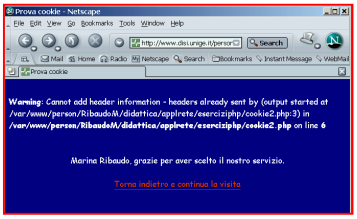
```
<?php

// scrivo il cookie usando la funzione urlencode()

$myval = "valore del cookie qui";
$expires = mktime(0,0,0,01,01,2004);

setcookie("mycookie",urlencode($myval),$expires);

?>
```

NB: il valore del cookie non può contenere i caratteri virgola, punto e virgola, spazio bianco e quindi è opportuno usare la funzione urlencode() che sostituisce questi caratteri con il codice %xx corrispondente.

Il cookie

domain	www.disi.unige.it
flag	FALSE
path	/person/RibaudoM/didattica/applrete/esercizip
secure	FALSE
expiration	1072911839
name	mycookie
value	Marina_Ribaudo

Un errore frequente …



"Cookie headers must be sent before any other headers or they will not work"

I cookie sono pericolosi?

- Sicuramente rivelano delle informazioni su di noi …
- Inoltre, come i campi nascosti, possono essere letti e modificati
- **Esempio:** supponete di avere un sito che ogni tanto fa degli sconti su alcuni prodotti. Se la codifica della percentuale di sconto è scritta nel cookie (o in un campo nascosto) in un formato semplice da leggere … potrebbe essere modificata e creare dei problemi … non si devono mai scrivere informazioni intelleggibili nei cookie (e nei campi nascosti)

Session control

HTTP è stateless, quindi non "mantiene informazioni" tra richieste successive, anche quando queste arrivano dallo stesso client. Per ovviare a questa "debolezza" è stato introdotto il session control, un meccanismo che permette di tener traccia dell'utente durante la sua interazione con un sito.

In PHP "… a visitor accessing your web site is assigned an unique id, the so-called session ID. This is either stored in a cookie on the user side or is propagated in the URL …"

Session control in PHP

- Session ID, numero casuale generato da PHP noto al client per la durata di una sessione
  - ✓ può essere memorizzato in un cookie (default)
  - ✓ passato mediante un URL
    (in automatico se PHP è stato compilato con l'opzione **--enable-trans- sid**)
  - ✓ scritto "manualmente"
    `<a href="link.php?<?=SID?>">`

Session control in PHP

- Questo ID permette di memorizzare (sul server) delle variabili particolari, dette variabili di sessione
- Le variabili di sessione sono memorizzate in un flat file
  (ma si possono anche scrivere delle funzioni per memorizzarle in un database)

Session control in PHP

session

Session Support	enabled

Directive	Local Value	Master Value
session.auto_start	Off	Off
session.cache_expire	180	180
session.cache_limiter	nocache	nocache
session.cookie_domain	*no value*	*no value*
session.cookie_lifetime	0	0
session.cookie_path	/	/
session.cookie_secure	Off	Off
session.entropy_file	*no value*	*no value*
session.entropy_length	0	0
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.name	PHPSESSID	PHPSESSID
session.referer_check	*no value*	*no value*
session.save_handler	files	files
session.save_path	/tmp	/tmp
session.serialize_handler	php	php
session.use_cookies	On	On

Session control in PHP

- Non tutti ne consigliano l'uso …

" … session variables are equivalent to global variables, except that each visitor to your site gets his or her own "set" of session variables. Session variables are variants, meaning that they can store anything, from strings to integers, to large objects. Session variables are useful at times because they make information passing very simple. They are bad because they really can hurt a site's performance, especially if you store large objects in session variables. "

Session control in PHP

- I passi da seguire sono i seguenti
  - ✓ Iniziare una sessione
  - ✓ Registrare le var. di sessione
  - ✓ Usare le var. di sessione
  - ✓ "Deregistrare" le var. di sessione e chiudere la sessione

Iniziare una sessione

```
session_start();
```

Questa funzione verifica se l'utente ha già un identificatore di sessione. Se non lo trova, ne crea uno, altrimenti rende "visibili" le variabili di sessione create per quell'utente

Quando si usano le sessioni è buona norma inziare tutti gli script con session_start()

Registrare le var. di sessione

```
$_SESSION["myvar"] = <valore>;

$HTTP_SESSION_VARS["myvar"] =
                        <valore>;
```

La variabile di sessione viene "tracciata" fino a quando non si termina la sessione

Registrare le var. di sessione *

```
$myvar = <valore>;
session_register("myvar");
```

* La direttiva register_global nel file di php.ini deve essere messa a On

Usare le var. di sessione

- Se si iniziano gli script con session_start() si possono usare le variabili di sessione usando gli array opportuni ($_SESSION, $HTTP_SESSION_VAR)o il nome della variabile
- Per controllare se una variabile di sessione è stata registrata …

Usare le var. di sessione

```
if (isset($_SESSION["myvar"]))

if (isset($HTTP_SESSION_VARS["myvar"]))
```

```
if (session_is_registered("myvar"))
```


"Deregistrare" le var. di sessione

```
unset($_SESSION["myvar"])

unset($HTTP_SESSION_VARS["myvar"])
```

```
session_unregister("myvar")
```

La sessione esiste ancora ma la variabile di sessione myvar non è più registrata come variabile di sessione

Chiudere la sessione

```
session_destroy()
```

cancella l'identificatore di sessione

Session ID: la costante SID

```
<?php
     session_start();
     echo "sid : " . SID;
?>
```

```
PHPSESSID=51afecb1a7fcd21072d689aa2904f77b
```

NB: per vedere il session ID generato da PHP (e memorizzato nella costante SID) si devono disabilitare i cookies!

Esempio: iniziare una sessione

```
<?php
  session_start();

  $_SESSION["msg"] = "Hello world";

  echo "Il valore della variabile di sessione è ";
  echo  $_SESSION["msg"] . "<br>";

?>

Vai alla <a href="page2.php">pagina 2</a>
```

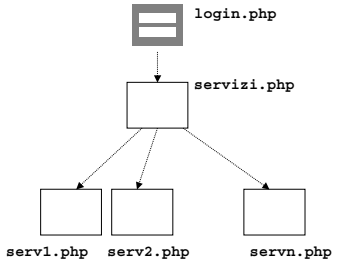

Esempio: accedere e "deregistrare"

```
<?php
  session_start();


  echo "Il valore della variabile di sessione è ";
  echo  $_SESSION["msg"] . "<br>";

  unset($_SESSION["msg"]);

?>

Vai alla <a href="page3.php">pagina 3</a>
```

Esempio: terminare una sessione

```
<?php
  session_start();

   echo "Il valore della variabile di sessione è ";
   echo  $_SESSION["msg"] . "<br>";

  session_destroy();

?>
```


Esempio: autenticazione con le sessioni



Esempio: autenticazione con le sessioni

- Se un utente cerca di accedere a pagine che dipendono dalla pagina login.php si dovrà fornire all'utente un messaggio del tipo "Per accedere a questi servizi devi prima autenticarti"
- Se l'utente digita username e password (corretti) gli verrà presentata la pagina con l'elenco dei servizi disponibili nell'area riservata

Esempio: autenticazione con le sessioni

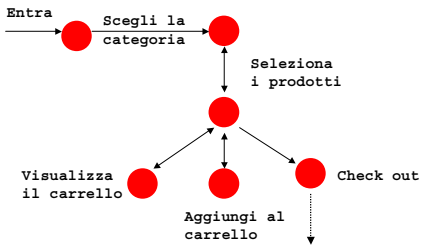
- Nella fase di autenticazione
  - ✓Si dovrà verificare la correttezza dei dati inseriti dall'utente andando ad interrogare il database degli utenti registrati
  - ✓Se l'utente è autorizzato, si dovrà creare una (o più) variabile di sessione e tutti i file successivi dovranno verificare l'esistenza della variabile di sessione

Contatore

```
<?php
  session_register("count");
  $count++;
?>

Hello visitor, you have seen this page
<? echo $count; ?> times.
<p>
To continue,
<A HREF="nextpage.php?<?=SID?>">click
here</A>
```

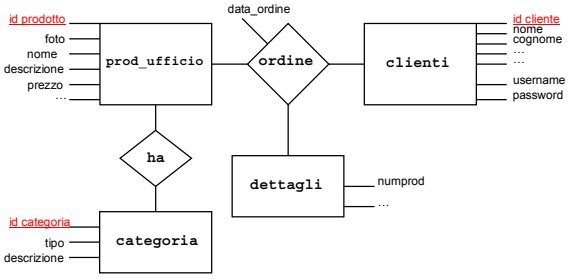
NB: se non si usa session_start(), la sessione viene iniziata la prima volta che si cerca di registrare una variabile di sessione

Esempio: gestione di un carrello

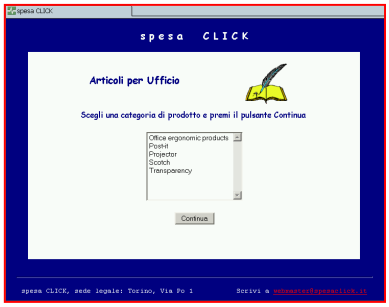


Esempio 2 sul web

Database dell'esempio



Esempio: gestione di un carrello

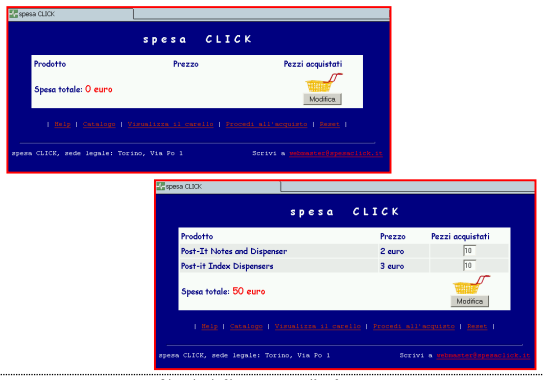


Creazione dinamica di un menu

```
<form name="frm" method="post" action="prodcat.php">
<select name="id_categoria" size="8">
<?
$query = "select * from categoria order by tipo";
$res = mysql_query($query);
$num_res = mysql_num_rows($res);

if ($num_res == 0)
      echo "Spiacenti, al momento non ci sono offerte";
else // formatto il risultato per il client
 while ($row = mysql_fetch_array($res))
   {
     echo "<option value='" . $row["id_categoria"] . "'>";
     echo " $row["tipo"] . "</option>\n";
   }
echo "</select>\n";
echo "…";
echo "</form>\n";
```


Esempio: gestione di un carrello

Esempio: gestione di un carrello



Esempio: gestione di un carrello

File condivisi

```
<?php
      include("comuni/sessione.php");
      include("comuni/header.html");
      include("comuni/connessionedb.php");

?>
```

```
<?php
      session_start();
?>
```


Carrello.php: aggiungi

```
// se il cliente ha selezionato il pulsante Aggiungi
// la prima volta creo le variabili di sessione

if ($azione=="aggiungi")
{
  if (!session_is_registered("carrello"))
 {
    // variabile per il carrello
    $carrello = array();
    session_register("carrello");

    // variabile per la spesa totale
    $spesa = 0;
    session_register("spesa");
  }

}
```


Carrello.php: aggiungi

```
// se il cliente ha selezionato il pulsante Aggiungi
// la prima volta creo le variabili di sessione
if ($azione=="aggiungi")
      {  … }

// aggiorno il  carrello
 foreach ($HTTP_POST_VARS as $key => $value)
{
    $value = intval($value);
    if ($value!="")
      {
       // aggiorno il carello
       if ($carrello[$key])
          $carrello[$key] = $carrello[$key] + $value;
       else
          $carrello[$key] = $value;
      }
}
```


Carrello.php: visualizza

```
// se il cliente ha selezionato il pulsante Visualizza

if ($azione=="visualizza")
{
 if (!session_is_registered("carrello"))
       echo "Non hai ancora scelto nessun prodotto\n";

 else

 foreach ($carrello as $key => $value)
  {
     <visualizza le info sui prodotti selezionati>
  }
}
```


Carrello.php: check out

```
// se il cliente ha selezionato il pulsante Check out
if ($azione=="checkout")
{
 if (!session_is_registered("carrello"))
      echo "Non hai scelto nessun prodotto.\n";
 else
  {
   if (!$spesa)
      echo "Non hai scelto nessun prodotto. \n";
   else
    {
      <rimando il cliente alla pagina di login>
    }
  }
 }
?>
```