PHP "avanzato": riepilogo preventivo

- adattamento al browser (stili alternativi)
- problema della "mancanza di stato"
  - hidden fields
  - cookie
  - session variables
- persistenza dei dati con session vars
  - autenticazione (sito con login)
  - carrello della spesa (base)
- PHP & MySQL (mariaDB)
  - carrello della spesa con memoria in una base di dati

Stili alternativi con PHP (1/2)

Lo stile adottato dipende dal risultato della preg_match

alternate/stili.alternativi.2.php

```
<?php
if (preg_match ("/Edg|Trident/",
$_SERVER['HTTP_USER_AGENT'])) {
   echo "<link rel=\"stylesheet\" type=\"text/css\"
href=\"mic.css\" />";
   $sty="mic.css";
}
elseif (preg_match ("/Firefox/",
...
elseif (preg_match ("/Safari/", $_SERVER['HTTP_USER_AGENT']))
{
   echo "<link rel=\"stylesheet\" type=\"text/css\"
href=\"goopple.css\" />";
   $sty="goopple.css";
}
else {
   echo "<link rel=\"stylesheet\" type=\"text/css\"
...
?>
```

PHP - problema della mancanza di stato

HTTP stateless. Soluzioni: 1) hidden field 2) cookie 3) session

hidden field = un campo di type hidden presente nella form, non viene visualizzato, ma è comunque portatore di una informazione del tipo "nome=valore": si fa in modo che questa informazione rappresenti dati provenienti dalle transazioni precedenti, che così possono essere usati per determinare la transazione attuale

1) Il flusso comincia con una form prodotta con anche uno o più campi hidden (contenenti info provenienti da scambi precedenti;
2) questa form arriva al client, che vede/compila i campi (tranne gli hidden, che non vede);
3) inviando, il client invia inconsapevolmente anche i campi hidden
4) i valori arrivati sono usati per produrre la risposta (che può essere conformata ai dati dei campi hidden)

```
<strong>Scegliere l'azione da eseguire</strong>,
…
<form action="http://www.dis.uniroma1.it/
~marte/cgi-bin/iscrizione.pl" method="post">
…
<input name="datappello"
       value="19 aprile 2013" type="hidden">
<input name="nomefiletesto" value="appelloF2apr13.txt" type="hidden">
<input name="insegnamento" value="Fondamenti di ..." type="hidden">...
```

PHP - hidden field

hidden/conteggio.hidden.1.php e 2

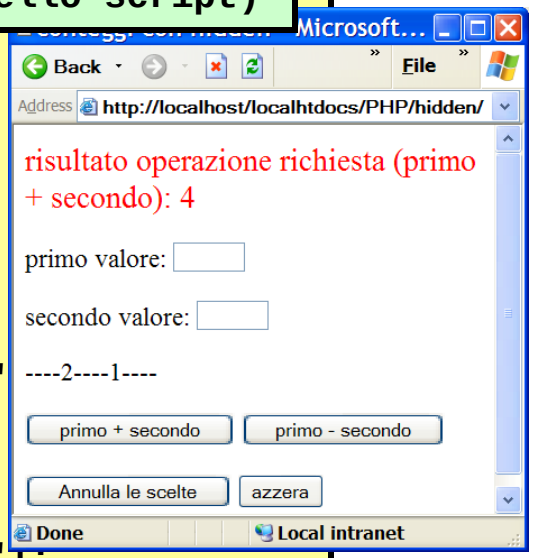
```
if (!isset($_POST["primoValore"]) || ($_POST["invio"]=="azzera")) {
   $som=0; $sot=0; echo "init";
}
else {                        // arriviamo dalla form
  $msg=""; $som=$_POST['sommè]; $sot=$_POST['sottrazionì];

  if ($_POST["primoValore"]=="")
    $msg="<br />ERRORE: primo non assegnato<br />";
...
  if ($msg=="") {
    $msg = "risultato operazione richiesta (".$_POST["invio"

    if ($_POST["invio"]=="primo + secondo") {
      $som++;
      $msg .= $_POST["primoValore"] + $_POST["secondoValore"];
    }
    else {
      ...}      }      }
?>
...
<body><p style="color: red;
<?php echo $msg;?>  </p>

<form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
...
<input type="hidden" name="somme" value="<?php echo $som ?>">
<input type="hidden" name="sottrazioni" value="<?php echo $sot; ?>">
...</form></body></html>
```

(è una chiamata diretta dello script)



- $sot è il numero di sottrazioni già fatte
- $som " " " " somme " "
- $msg è vuoto all'inizio e poi concatena i messaggi di errore o risultato prodotti via via
- $_POST['primoValorè] e $_POST['secondoValorè] sono gli addendi

PHP - hidden field

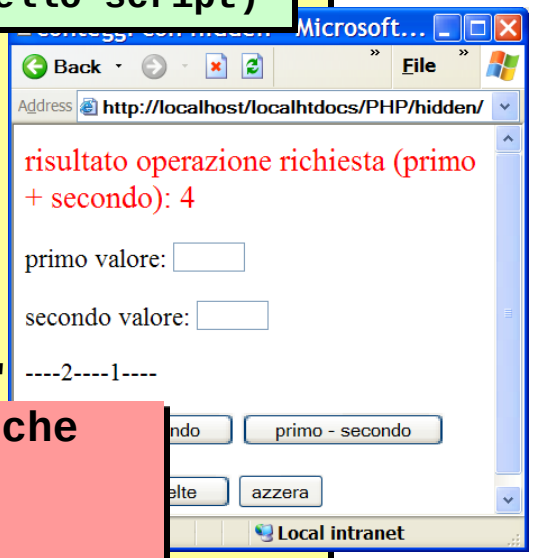
hidden/conteggio.hidden.1.php e 2

```
if (!isset($_POST["primoValore"]) || ($_POST["invio"]=="azzera")) {
   $som=0; $sot=0; echo "init";
}
else {                          // arriviamo dalla form
  $msg=""; $som=$_POST['sommè]; $sot=$_POST['sottrazionì];

  if ($_POST["primoValore"]=="")
    $msg="<br />ERRORE: primo non assegnato<br />";
...
  if ($msg=="") {
    $msg = "risultato operazione richiesta (".$_POST["invio"
...
?>
...
<bod
<?ph

<for
...
<input type="hidden" name="somme" value="<?php echo $som ?>">
<input type="hidden" name="sottrazioni" value="<?php echo $sot; ?>">
...</form></body></html>
```

(è una chiamata diretta dello script)



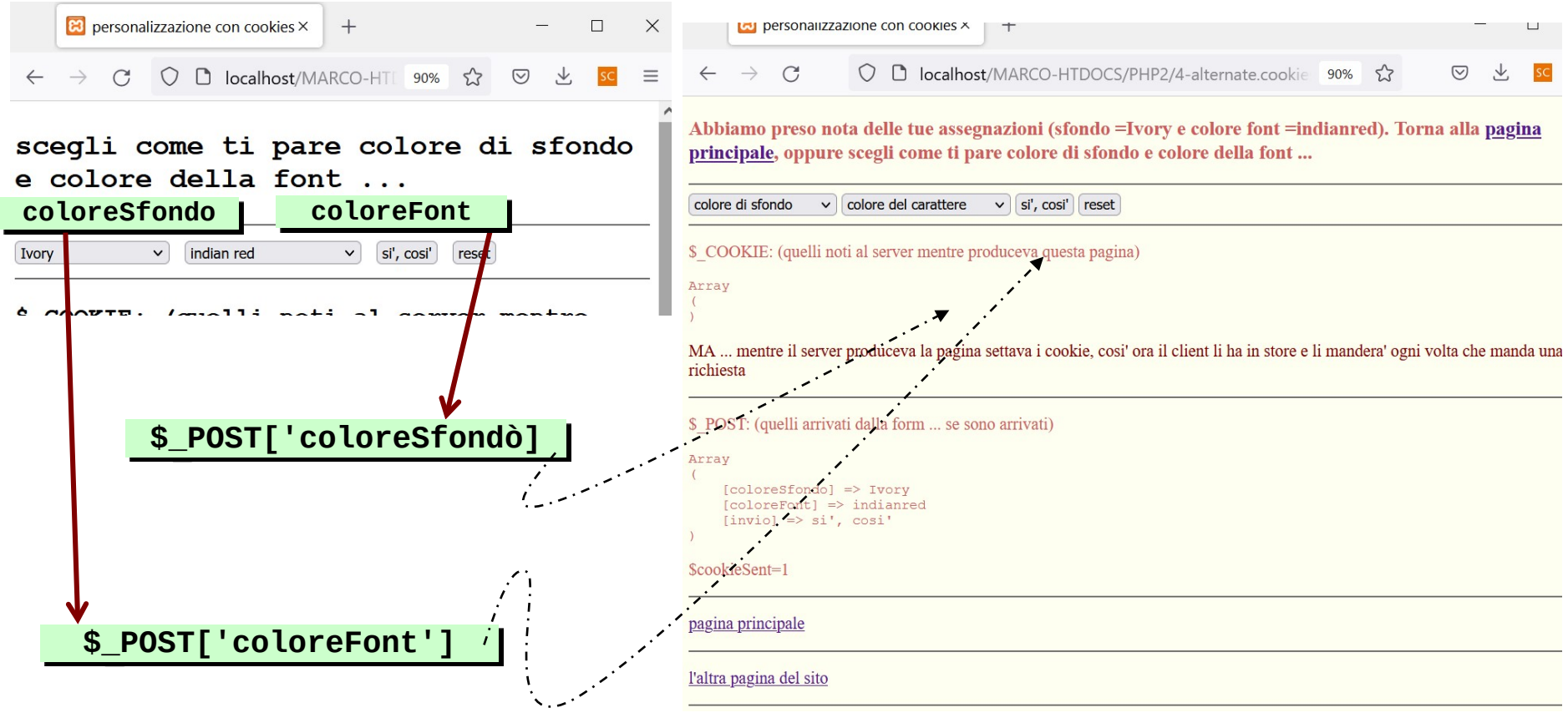
In conteggio.hidden.1.php c'è un piccolo errore che determina un warning … correggerlo

Vedi in conteggio.hidden.2 come evitare che per un errore nella form si perdano anche i dati che vi erano stati compilati correttamente.
Questo è anche un buon modo per sperimentare lo script numerose volte senza dover reimmettere i dati.

Poi chiama con il browser conteggio.hidden.3.php e risolvi l'esercizio descritto in fondo.

PHP - cookies

Un problema: i campi **coloreFont** e **coloreSfondo** della select sotto, arrivano attraverso i dati fluiti dalla form … permettendo di stabilire lo stile di visualizzazione per la pagina … ma questo solo se la pagina viene costruita a partire da quei dati … e in seguito?



Se visualizziamo una pagina del sito web senza provenire da questa form, dove sono queste informazioni sui colori?
Bisognerebbe mantenerle disponibili e valide anche dopo che abbiamo usato la form per definirle.
Come fare?? Una soluzione è nell'usare I campi hidden.

Ma una soluzione diversa
è metterle in cookies …

alternate.cookies/stili.alternativi.cookies.php

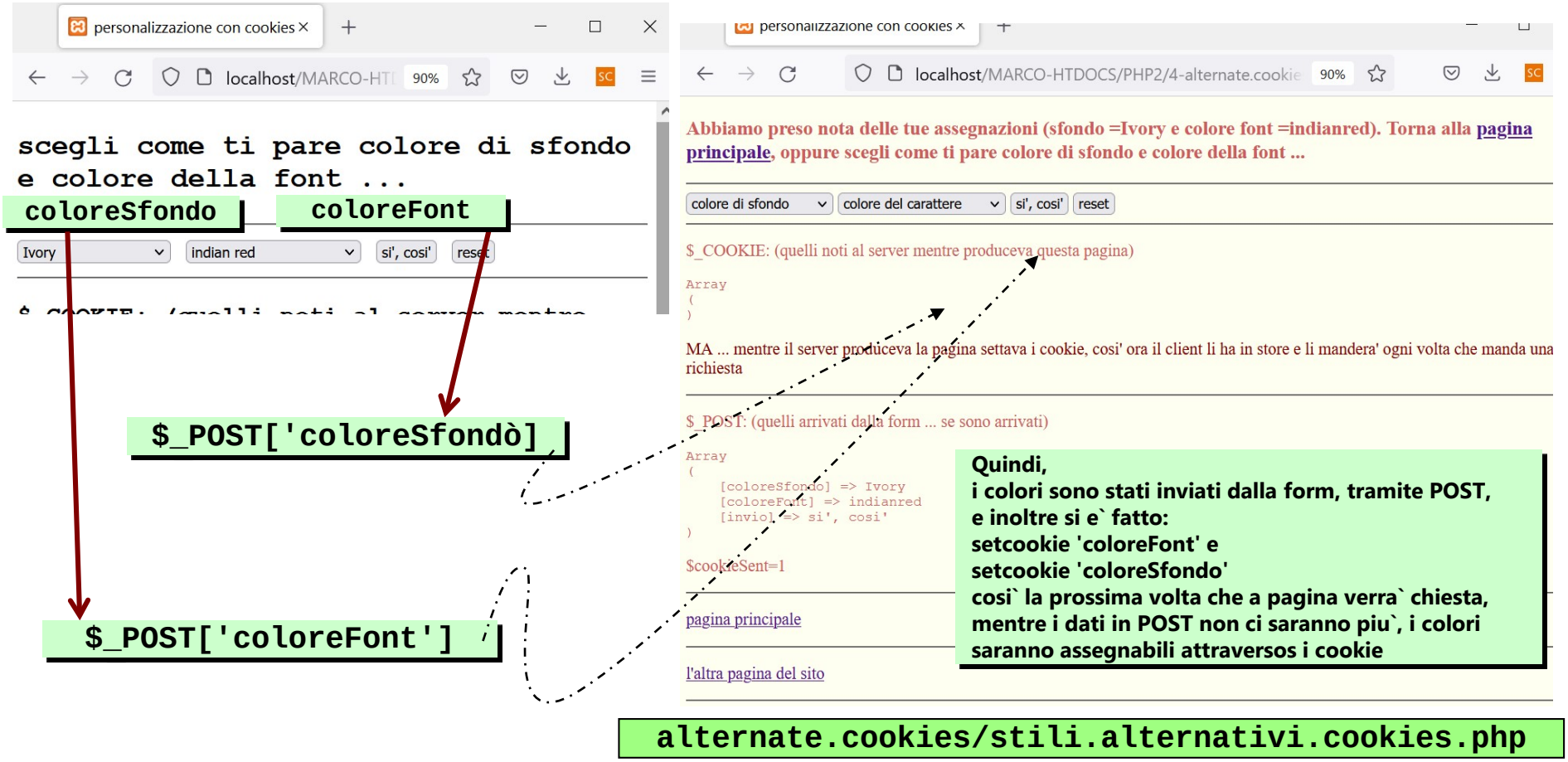
PHP - cookies

I cookies permettono al server di avere informazioni sul client, senza che queste informazioni arrivino come dati da una form.

Ad esempio i campi **`coloreFont`** e **`coloreSfondo`** della select sotto possono essere memorizzati in cookies **coloreFont** e **coloreSfondo**:

quando la pagina viene prodotta in risposta alla form di scelta colori, può essere colorata usando queste scelte; per ora i cookie sono stati solo settati sul cliente, ma non ancora inviati dal client al server …

invece …



`alternate.cookies/stili.alternativi.cookies.php`

PHP - cookies

I cookies permettono al server di avere informazioni sul client, senza che queste informazioni arrivino come dati da una form.
Ad esempio i campi **`coloreFont`** e **`coloreSfondo`** della select sotto possono essere memorizzati in cookies **coloreFont** e **coloreSfondo**:

… invece

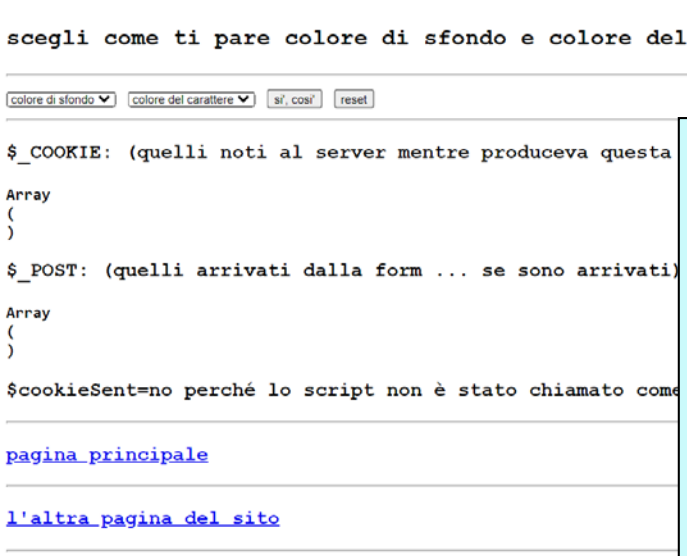
quando la pagina viene richiesta dal client al server, le specifiche di visualizzazione vengono inviate al server mediante i cookies (che ora sono disponibili sul client).

Qui non è una form ad inviare i dati di coloratura … ma la richiesta di una pagina al server, con contestuale invio dei cookie precedentemente settati. Il server riceve la richiesta, complete di cookie, così quando prepara la pagina di risposta può colorarla secondo i dati che ha ricevuto tramite cookies.

`alternate.cookies/stili.alternativi.cookies.php`

PHP - stili.alternativi.cookies.php - 1/3 -

alternate.cookies/



setcookie(nomecookie,valore)

produce la coppia nomecookie/valore, memorizzata in

$_COOKIE['nomecookiè]=>valore

si tratta di qualcosa che andrà nell'header del messaggio: non può esser fatto se già dell'HTML è stato inviato!

```
if (isset($_POST['invio']))          // abbiamo appena dato dati
  if ($_POST['coloreSfondò] && $_POST['coloreFont']) {
                    setcookie('coloreSfondò, $_POST['coloreSfondò]);
                    setcookie('coloreFont', $_POST['coloreFont']);
      $cookieSent=1;
  }
  else {
    $cookieSent= ……;  // non sono partiti nuovi cookie per errore dati
    echo "dati mancanti!<br />";
  }
else $cookieSent=……; // script chiamato da link (non come action)
```

PHP - stili.alternativi.cookies.php - 2/3 -

alternate.cookies/

$_COOKIE arriva al server automaticamente ad ogni richiesta;
il codice delle pagine del sito, ogni volta costruisce lo stile interno usando, se ci sono, i valori di $_COOKIE
(lo stile interno viene costruito nella stringa $stileInterno)
Se i cookies non sono disponibili, ma si viene dalla form di assegnazione colori, lo stileInterno viene costruito in base alle scelte date in $_POST.

estratto da stili.alternativi.cookies.php

```
if (isset($_COOKIE['coloreSfondò']))
  $stileInterno.="background: ".$_COOKIE['coloreSfondò'].";\n";
else
  $stileInterno.="background: blue;\n";

if (isset($_COOKIE['coloreFont']))
  $stileInterno.="color: "
.$_COOKIE['coloreFont'].";\n";
else
  $stileInterno.="color: yellow;\n";

$stileInterno.="}\n</style>\n";
```

Le altre pagine del "sito": principio.php, altra.pagina.php

```
…
<head>
<title>personalizzazione con cookies
(pagina main)</title>
<?php echo $stileInterno; ?>
</head>
```

PHP - stili.alternativi.cookies.php 3/3 -

alternate.cookies/cancella.cookies.php

In realtà la funzione ha più parametri: la chiamata completa è

`setcookie(nomecookie, valore, expiration, path, secure)`

expiration = "ora" in cui il cookie verrà rimosso (se non specificata, la rimozione è alla chiusura del browser)

path/domain = il cookie è accessibile solo se lo script agisce nel percorso/dominio indicato

secure = 1/0 se questa gestione è prevista solo per HTTPS

Il cookie viene cancellato
- usando setcookie senza valore
- e aggiungendo, se si vuole, *expiration time* nel passato

```
…
    setcookie('coloreSfondò, '');
    setcookie('coloreFont', '');
?>
```

```
…
    setcookie('coloreSfondò, '', time()-60);
    setcookie('coloreFont', '', time()-60);
?>
```

PHP - session

Così come i cookies, una sessione permette di mantenere informazioni sulle relazioni tra server e client.
A differenza del caso dei cookies, una sessione memorizza le informazioni sul server, in un'area di memoria associata ad un identificatore di sessione.

Quando, in conseguenza di una richiesta, una sessione viene aperta, il server manda al client (con la risposta) l'identificatore di sessione.
Il client, a sua volta, passa al server il suo id. di sessione ad ogni successiva richiesta, in modo che il server sappia a quale insieme di valori di sessione fare riferimento.

L'identificatore di sessione "viaggia" tra client e server in uno tra due possibili modi:
- con un cookie
- appendendolo, più o meno automaticamente, alla url della richiesta

- Con una sessione si può gestire **più memoria** che con i cookie.

- La gestione con una sessione rende necessario far **viaggiare solo l'identificatore** di sessione (e non tutti i dati significativi per l'interazione, come per i cookie).

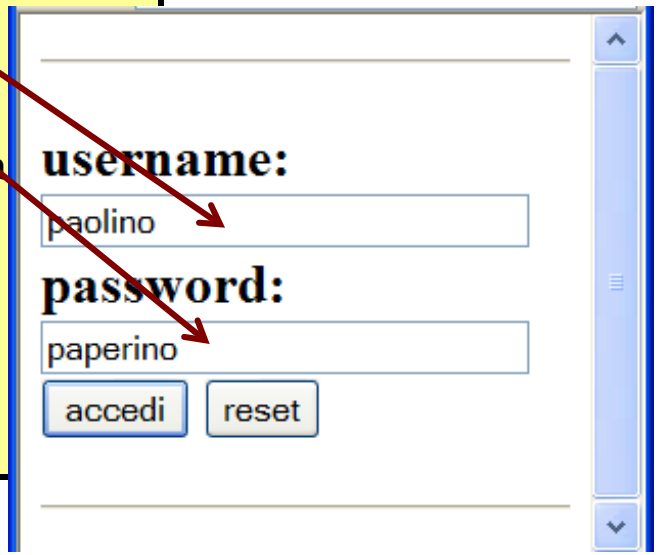
- **Se il browser non accetta cookie**, si può comunque instaurare una interazione usando l'id. di sessione, mentre se si usassero solo i cookie non ci sarebbe soluzione.

PHP - session creation

```php
<?php
ini_set('display_errors', 1);
error_reporting(E_ALL);

if (isset($_POST['invio']))    // abbiamo appena dato d
  if (empty($_POST['userName']) ||
      empty($_POST['password']))
    echo "<p>dati mancanti!!!</p>";
  else                         // controllo dati
    if ( ($_POST['userName']=="paolino") &&
         ($_POST['password']=="paperino") ) { //ok
      session_start();
      $_SESSION['userName']=$_POST['userName'];
      $_SESSION['dataLogin']=time();
      $_SESSION['accessoPermesso']=1000;
      header('Location: principio.php'); // pag inizia
      exit();
    }
    else {
    echo "<p>accesso negato!!!</p>";
  }
?>
```

session_start()
prima di qualsiasi HTML!
Sennò *headers already sent*
$_SESSION['nome']=valore;
registrazione e assegnazione di una variabile di sessione



In PHP le variabili di sessione sono rese disponibili nell'array $_SESSION (sono di qualunque tipo)

Ridireziona su principio.php

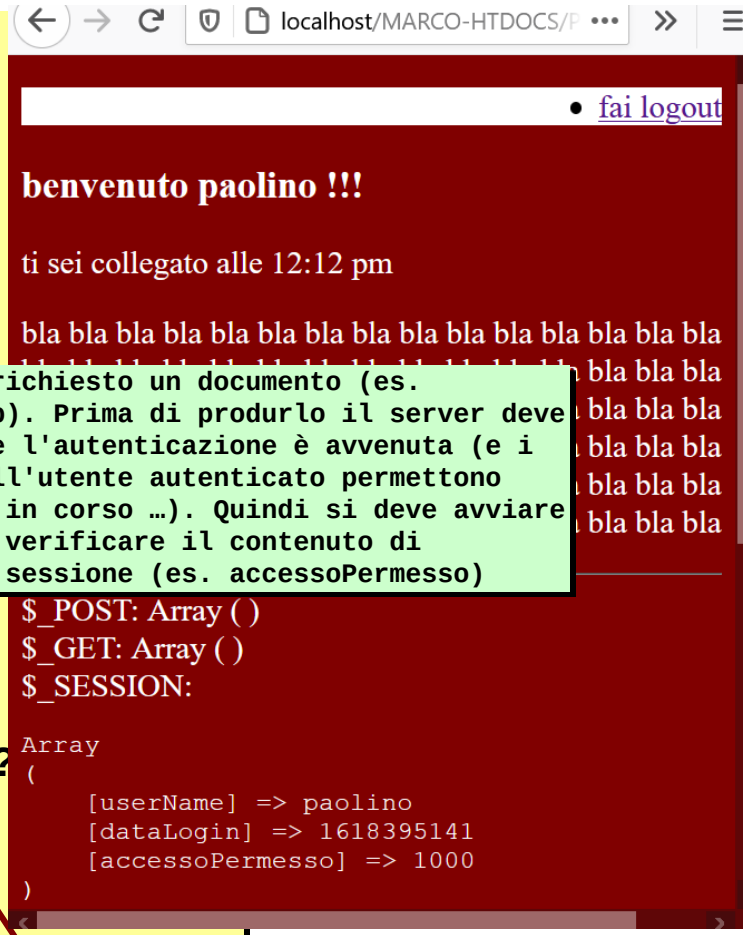
login.session/login0/login0.php

PHP - session variable check

```
<?php ...
$stileInterno=creaStileInterno();

session_start();
if (!isset($_SESSION['accessoPermesso']))
   header('Location: login.php');
?>
...
<?php echo $stileInterno; ?>
</head>

<body>
...
<h3>
benvenuto <?php echo $_SESSION['userName'];?>
</h3>
<p>ti sei collegato alle
<?php echo date('g:i a',
$_SESSION['dataLogin']);?></p>
<p>
bla bla bla bla bla bla bla bla bla bla bla b
```



Qui è stato richiesto un documento (es. principio.php). Prima di produrlo il server deve verificare se l'autenticazione è avvenuta (e i privilegi dell'utente autenticato permettono l'operazione in corso …). Quindi si deve avviare la session e verificare il contenuto di variabili di sessione (es. accessoPermesso)

Se `$_SESSION['accessoPermessò]` non e` assegnata, questo ridireziona su `login.php`

- `session_start()` ci vuole in ogni pagina
- `$_SESSION['<nome>']` è l'accesso ad una variabile di sessione esistente (una variabile di sessione esiste dal momento della sua definizione /assegnazione)

`login.session/login0/principio.php`

PHP - session deletion

login.session/login0

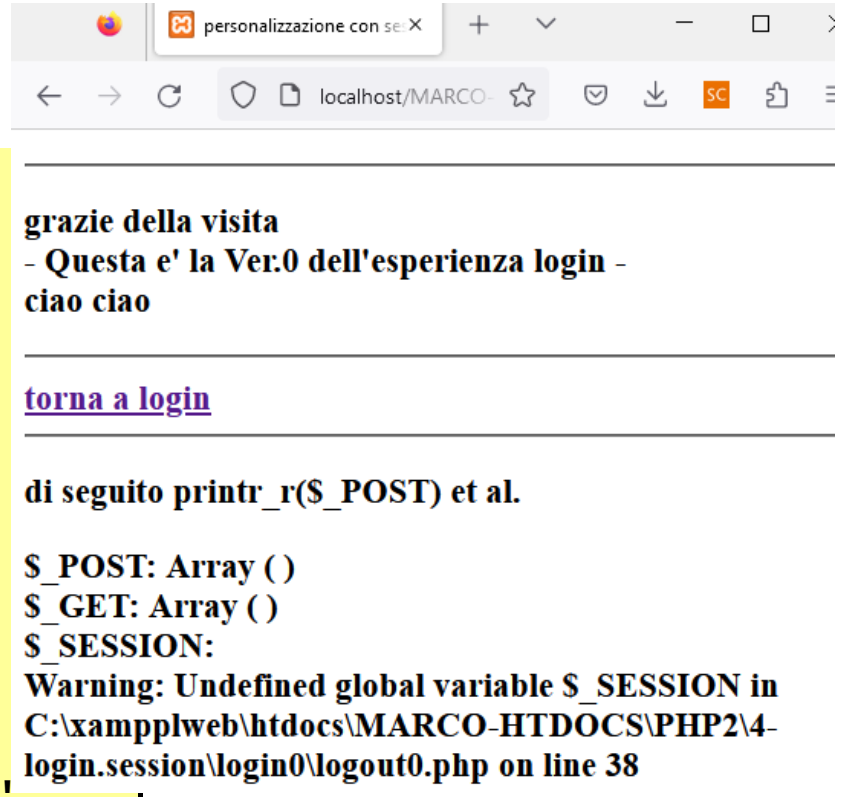
```
<?php
ini_set('display_errors', 1);
error_reporting(E_ALL);


session_start();
unset($_SESSION);
session_destroy();
?>

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
...
```

Cancellazione delle variabili in $_SESSION

Rimozione dal server



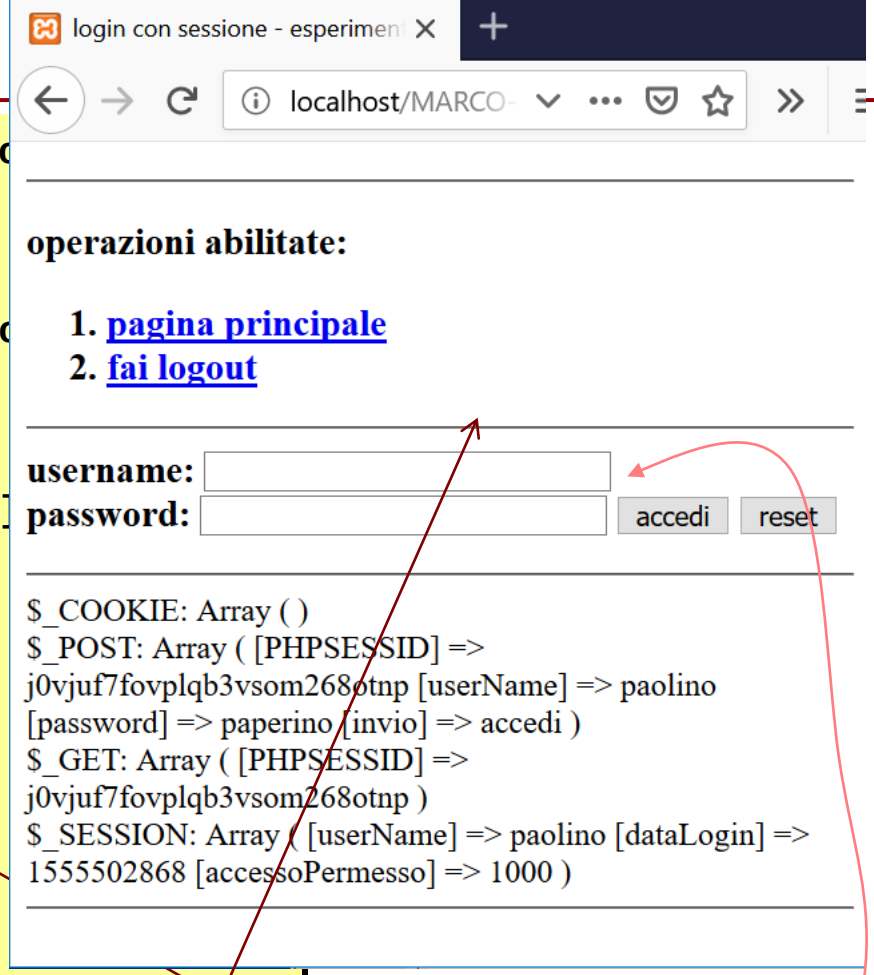
`session_start()` ci vuole in ogni pagina

login.session/login1/

```
check session_name()
check session_name("nomeDifferente")
```

PHP - session senza cookie (...)

```
if (isset($_POST['invio']))    // abbiamo dato
  if (empty($_POST['userName']) ||
      empty($_POST['password']))
    echo "<p>dati mancanti!!!</p>";
  else                              // controllo
    if ( ($_POST['userName']=="paolino") &&
($_POST['password']=="paperino") ) { //ok
      session_start();
      $_SESSION['userName']=$_POST['userName'];
      $_SESSION['dataLogin']=time();
      $_SESSION['accessoPermesso']=1000;
$link="<h3>operazioni abilitate:\n";
$link.="<ol><li><a href= ...
    }
    else {
    echo "<p>accesso negato!!!</p>";
  }
?>
…
<form action="login2.php?<?php
                    ...; ?>" method="post">
 username: <input type="text" name="userName" ...
 password: <input type="text" name="password" ...
...
</form>
```



$link è un esempio di *menù* ... (in caso di autenticazione viene costruita e stampata)

NB i campi di login non dovrebbero rimanere, per non generare equivoci ...]

login.session/login2/login2.php

PHP - session senza cookie (...)

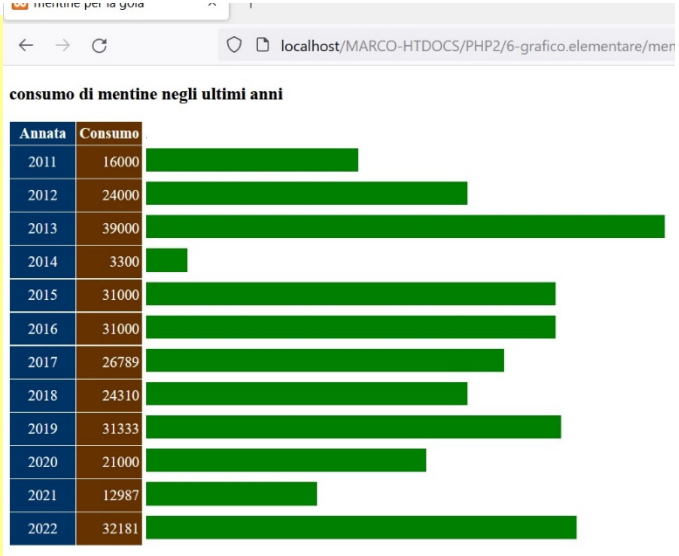
```
if (isset($_POST['invio']))    // abbiamo dat
  if (empty($_POST['userName']) ||
      empty($_POST['password']))
    echo "<p>dati mancanti!!!</p>";
  else                              // controll
    if ( ($_POST['userName']=="paolino") &&
($_POST['password']=="paperino") ) { //ok
      session_start();
      $_SESSION['userName']=$_POST['userName'
      $_SESSION['dataLogin']=time();
      $_SESSION['accessoPermessò]=1000;
$link="<h3>operazioni abilitate:\n";
$link.="<ol><li><a href= ...
    }
    else {
    echo "<p>accesso negato!!!</p>";
  }
?>
…
<form action="login2.php?<?php echo session_id();
?>" method="post"?>" method="post">
 username: <input type="text" name="userName" ...
 password: <input type="text" name="password" ...
...
```

visto che non possiamo usare i cookie, appendiamo SID (l'id. di sessione) ad ogni richiesta.
(Vedi lo script della paina principale in login2, per come forzare questo tipo di gestione)

login.session/login2/login2.php

PHP – un minimo di grafica e gestione file

```
<?php
   $valori = file("mentine.txt");
                          // massimo valore
   $max = $valori[1];
   for ($i = 1; $i<count($valori); $i+=2) {
      $valori[$i] = rtrim($valori[$i]);
      if ($valori[$i] > $max)
      { $max = $valori[$i]; }
   }
?>
<table border="0" cellspacing="1" cellpadding="3">
 ...
 <?php  // stampa 1 linea perognicoppia anno/consumo
   for ($i=0; $i<count($valori); $i+=2) {
 ?>
   <tr>
      <td class="anno" width="10%">
          <?php echo rtrim($valori[$i]) ?></td>
      <td class="consumo" width="10%">
          <?php echo ($valori[$i+1]) ?></td>
<td><img src="puntino.verde.png" height="25"
width="<?php echo round(($valori[$i+1]/$max)*100)."%";
?>" alt="<?php echo $valori[$i+1] ?>" /></td>
   </tr>
```



`file(nomefile)` produce un array con le linee del file (in mentine.txt ci sono coppie di linee anno/consumo)

`count(array)` =numero elementi array

riga verde = un'immagine di un pixel (ingrandito…)

dimensioni: il massimo valore occupa il 100% dello spazio disponibile; gli altri sono scalati.

grafico.elementare/mentine.php

Esercizio: mostrare colori diversi per le barre in base intervallic di valori.
Esercizio: mostrare diveramente le caselle e le barre se si tratta di previsioni.

PHP - carrello della spesa - 1/5 - ("architettura dell'applicazione")

ST.login.php
apertura sessione

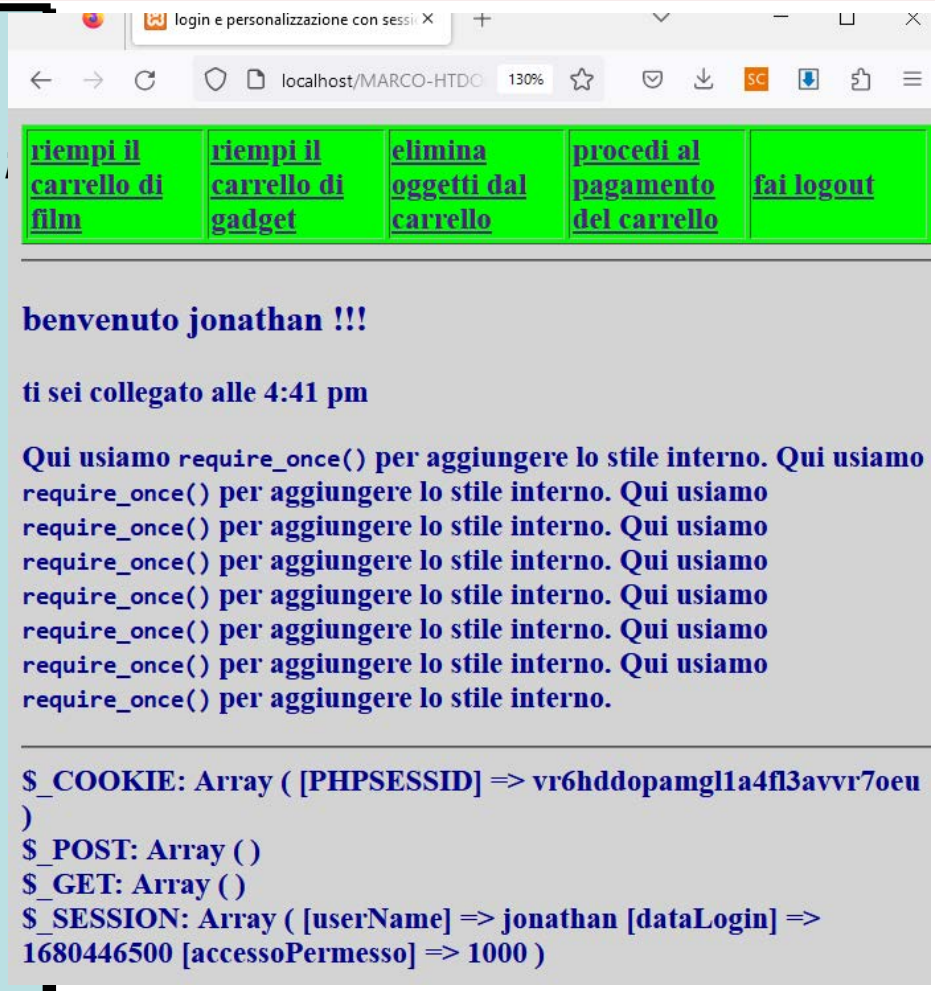
```
$_SESSION['userName']=$_POST['userName'];
$_SESSION['dataLogin']=time();
$_SESSION['accessoPermesso']=1000;
```

ST.inizio.php
ST.movie.php
ST.gadget.php
ST.elimina.php
ST.paga.php
ST.logout.php

- session identifier via cookie

- $_SESSION['carrello'] array acquisti;
  i suoi elementi vengono da
  - aggiunte fatte da ST.movie, ST.gadget
  - ed eliminazioni fatte da ST.elimina.
  - inoltre ST.paga accede a carrello per calcolare il costo complessivo.

- in ST.gadget ci sono dei problemi:
quali sono? Come risolverli?
Guardare nel file …



carrello.della.spesa/

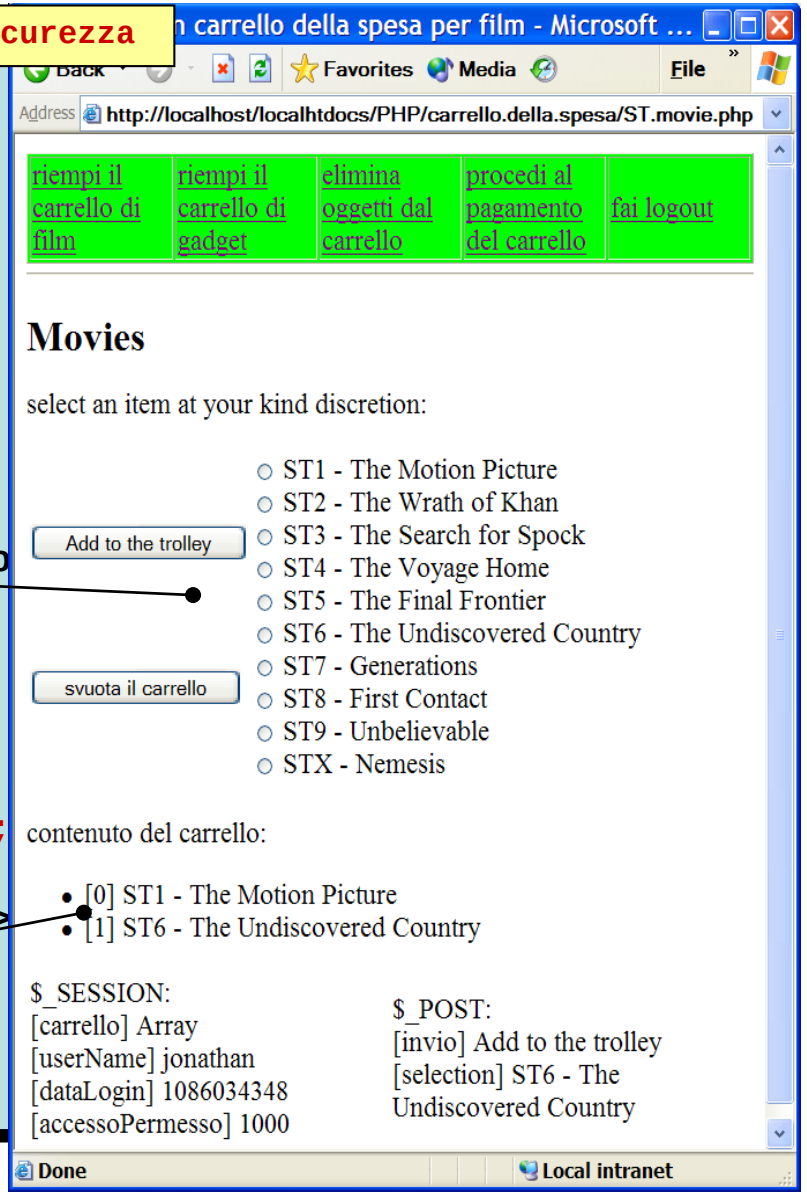
PHP - carrello della spesa - 2/5 -

Controllo di sicurezza

```
ST.login.php
apertura sessione
  $_SESSION['userName']=$_POST['userName'];
  $_SESSION['dataLogin']=time();
  $_SESSION['accessoPermessò]=1000;

ST.movie.php
if (!isset($_SESSION['accessoPermessò']))
header('Location: ST.login.php');


if ((!$_SESSION['carrello'] && !$_POST['selectio
    || $_POST['azzeraAcquistì]) {
 $_SESSION['carrello']=array();
 echo "<p> - carrello vuoto - </p>";
} else {
   if ( $_POST['selection']) {
   $_SESSION['carrello'][] =$_POST['selection'];
   }
echo "<p>contenuto del carrello:</p>";echo "<ul>
   foreach ($_SESSION['carrello'] as $k=>$v)
    echo "<li>[$k] $v</li>";
   echo "</ul>";
}?>
```



NB - $_SESSION['carrello'] array

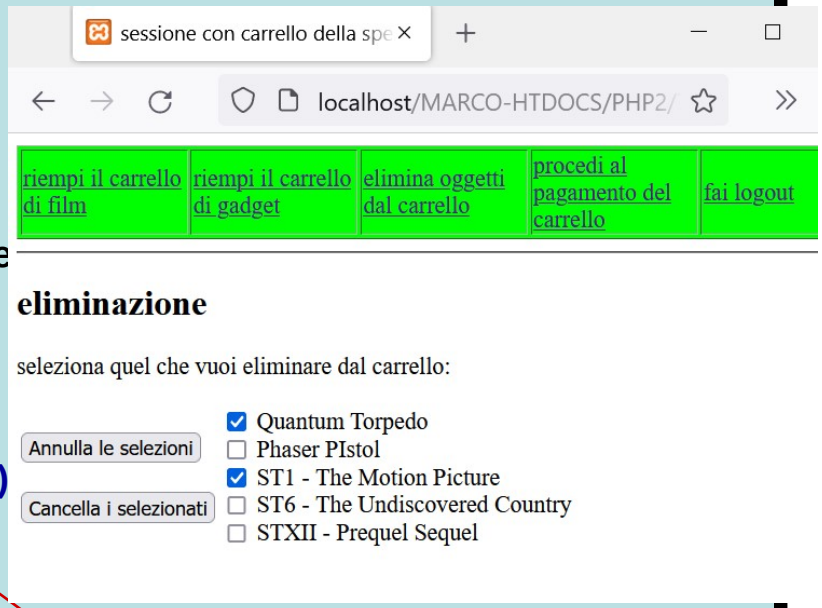
Aggiunta di un elemento nell'array $_SESSION

carrello.della.spesa/

PHP - carrello della spesa - 4/5 -

ST.elimina.php

```
<?php
if (!$_SESSION['carrello']) {
   echo "<p> - carrello vuoto - </p>";
} else {
   if ( !$_POST['eliminandi']) {
      echo "<p>seleziona quel che vuoi eliminare
   }
   else { // eliminazione da eliminandi[]
     foreach ($_POST['eliminandi']
              as $k=>$indiceDaEliminare)
unset($_SESSION['carrello'][$indiceDaEliminare])
   }
}
?>
…
<form action="<?php $_SERVER['PHP_SELF']?>"  method="post" >
...
 <td><?php
   foreach ($_SESSION['carrello'] as $k=>$v)
     echo "<input type=\"checkbox\"
           name=\"eliminandi[]\"
           value=\"$k\"> $v<br />";
  ?>
 </td></tr></table></form>
```



vedere

- **prima codice form**
- **e poi codice gestione dati form**

carrello.della.spesa/

PHP - carrello della spesa - 5/5 -

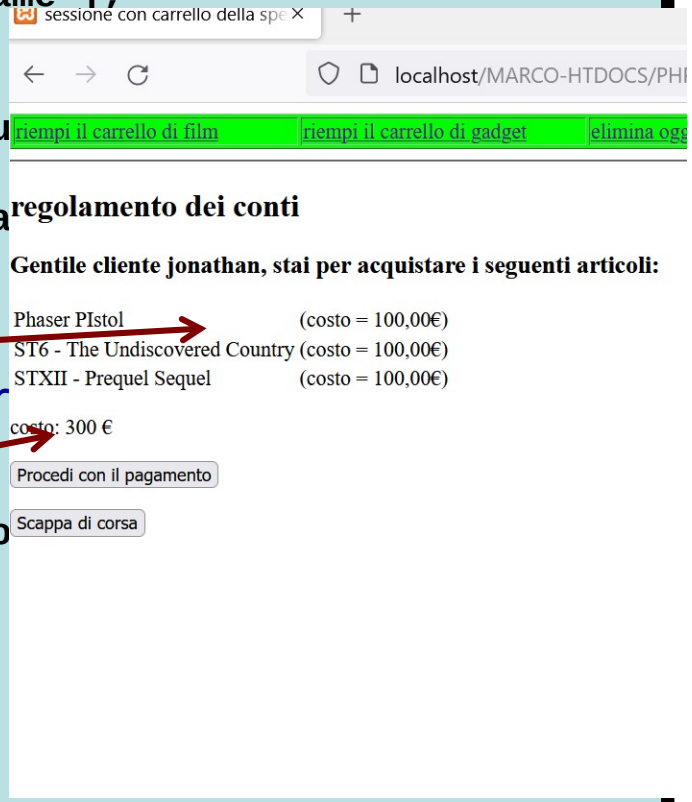
carrello.della.spesa/

ST.paga.php

```
<?php
session_start();
if (!isset($_SESSION['accessoPermessò']))
    header('Location: ST.login.php');

$costoFinale=0;
$outputTable="<h3>Gentile cliente ".$_SESSION['userName'];

if (!$_SESSION['carrello']) {
   $outputTable.= " che ci fai qui con un carrello vu
} else {
    $outputTable.=", stai per acquistare i seguenti a
    $outputTable.="<table>\n";
foreach ($_SESSION['carrello'] as $k=>$v) {
 $outputTable.="<tr>\n<td>$v</td>\n";
 $outputTable.="<td>(costo = 100,00&euro;)</td>\n</tr
 $costoFinale+=100.00;
}
$outputTable.="</table>\n<p>costo: $costoFinale &euro
  }
...
...
...
<?php echo $outputTable;?>
<form action="zona.pagamenti.php"  method="post" >
...
```



- ST.inizio.php usa require_once
- zona.pagamenti è fittizia; però unsetta il carrello
- ST.logout.php fa unset() e destroy() su $_SESSION

PHP - database connectivity

Connessioni a basi di dati sono rese possibili attraverso interfacce, tipicamente native nel linguaggio di scripting.

API native PHP

- gruppi di funzioni dedicate ciascun**o** a un db particolare, parte del linguaggio
- implementazioni meno generiche di quel che può essere con sistemi indipendenti da linguaggi e sistemi operativi (ODBC)
- piuttosto alte performance
- tanti gruppi per tanti db (oracle, db2, sybase/sap, postgresql, informix, ingres, interbase, MySQL)

A proposito di MySQL … DBMC (DataBase Management System) open source (... Oracle says), disponibile per molti so.

A proposito di Mariadb … continuazione free di mysql, disponibile per molti so

- non ha un'interfaccia utente grafica per definizione (ce ne sono alcune sviluppate separatamente, come PhPMyAdmin) ma tanto noi sappiamo usare l'interfaccia "a linea di comando"
- affidabile, scalabile e robusto (… wikipedia, twitter …)

MySQL (& MariaDB)

Tutto è gestito attraverso le tabelle di un database chiamato ***mysql***. Le tabelle di mysql servono a governare il controllo dell'accesso agli altri database (quelli che l'utente ha definito)

- **user** (controllo utenti, dove un utente ha uno userName e può avere una password - niente a che vedere con gli utenti del so … l'utente root di mysql può tutto, in MySQL, e inizialmente è senza password - mettetela!)

- **db** (controllo accesso ai database)

- **host**, **Tables_priv, Columns_priv** (controllo accesso in base a hostname e privilegi di accesso per tabelle e colonne)

PHP with MySQL

Varie funzioni PHP permettono di lavorare con le basi di dati in MySQL: la famiglia di funzioni che usiamo qui è *mysqli* (si tratta di una classe con numerosi metodi che permettono di gestire il dbms MySQL)

```
-$connection = new mysqli("host", “mysqluser", “pwd", dbname); //
stabilisce una connessione con il dbms ed una sua base di dati
-mysqli_errno ()...
-mysqli_query($sqlQuery); - send a SQL query to the active database:
$resultQ = mysqli_query($mysqliConnection, $sql)
-mysqli_fetch_array($resultQ) – … una delle righe risultanti da una query
-mysqli_close($connection);                oppure $connection->close();
```

PHP - carrello della spesa 2

mysql.ST1.php
passi significativi
- connessione a mysql
- Creazione della base di dati STdb, mediante mysqli_query (query passata tramite stringa …CREATE DATABASE
- creazione e popolamento iniziale delle tabelle STuser, STMovie, STGadget (query CREATE TABLE, INSERT INTO ...)

mysql.ST2.php form per inserire nuovi film in STmovie:
- Connessione
- processazione dati provenienti dalla form (action PHP_SELF)
- query INSERT INTO sulla tabella STmovie

mysql.ST.login.php
- Connessione e autenticazione (validazione username/password) mediante accesso a STuser

```
$_SESSION['userName']=$_POST['userName'];
$_SESSION['dataLogin']=time();
$_SESSION['accessoPermessò]=1000;
$_SESSION['spesaFinorà]=spese precedenti user
```

mysql.ST.movie.php def. $_SESSION['carrello'] e aggiunte film al carrello

mysql.ST.gadget.php aggiunte al carrello

ST.inizio.php ipotetica home-utente sito

ST.elimina.php eliminazioni dal carr.

ST.paga.php calcolo e attiv.pagam.

zona.pagamenti.php pagamento = modifica campo speseFatte nel record (riga) dell'utente nella tabella STuser

mysql.ST1.php e mysql.ST2.php sono privi di verifica di autorizzazione: dovrebbero essere protetti ...

carrello.della.spesa2/

carrello.della.spesa2 - 1/8 - mysql.ST.login.php

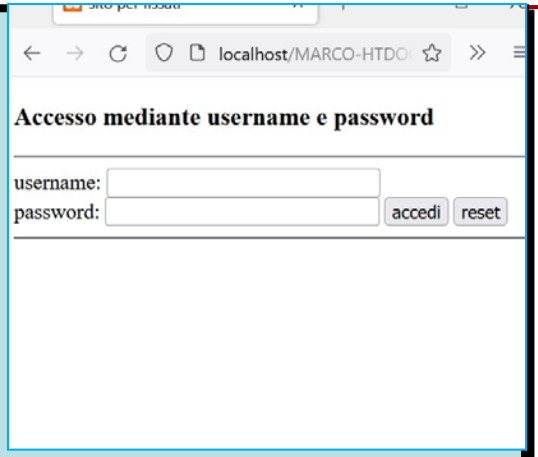
mysql.ST.login.php

```
$db_name = "STdb";
$STuser_table_name = "STuser";
$STmovie_table_name = "STmovie";
$STgadget_table_name = "STgadget";

$mysqliConnection =
      new mysqli("localhost", "archer", "archer", $db_name);

if (mysqli_connect_errno($mysqliConnection)) {
  printf("Oops, abbiamo problemi con la connessione al db:
%s\n",mysqli_connect_error($mysqliConnection));
  exit();
}
...
```

ma bisogna ripetere tutto in ogni file?



mysqli classe la cui istanziazione corrisponde alla effettuazne di una connessione;
il descrittore della connessione è memorizzato nella variabile $mysqliConnection;
localhost è l'host su cui gira il dbms MySQL;
archer, con password archer è un esempio di utente mysql (cioè registrato sul dbms con la possibilità di accedere ad alcune basi di dati).

mysqli_errno() and **mysqli_error()** restituiscono codici e messaggi relativi all'ultimo errore

carrello.della.spesa2/

carrello.della.spesa2 - 2/8 - ancora mysql.ST.login.php

```
...
else {                              // controllo dati
        $sql = "SELECT *
        FROM $STuser_table_name
        WHERE userName = \"{$_POST['userName']}\" AND
              password = \"{$_POST['password']}\"
        ";
if (!$resultQ = mysqli_query($mysqliConnection, $sql)) {
   printf("Oops, la query non ha risultato !!\n");
   exit();
}

$row = mysqli_fetch_array($resultQ);
if ($row) {   // utente esiste valido
      session_start();
      $_SESSION['userName']=$_POST['userName'];
      $_SESSION['dataLogin']=time();
      $_SESSION['numeroUtentè]=$row['userId'];
      $_SESSION['spesaFinorà]=$row['sommeSpesè];
      $_SESSION['accessoPermessò]=1000;
      header('Location: mysql.ST.inizio.php'); exit();
    }
    else echo "<p>accesso negato!!!</p>";
  }?>
```

`mysqli_query ()` esegue sul db la query passata; il risultato è un array php, contenente le righe della tabella che sono state selezionate

`mysqli_fetch_array ()` permette di scandire le righe del risultato della query (in questo caso una sola)

carrello.della.spesa2/

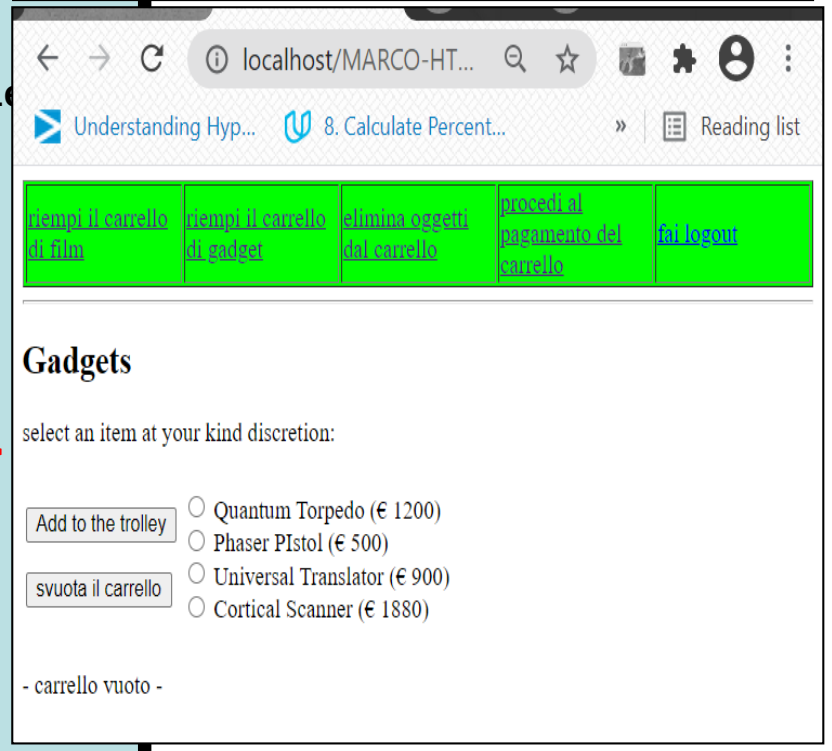
```
… if (!isset($_SESSION['accessoPermessò']))
header('Location: mysql.ST.login.php');
...  // selezione gadget disponibili da STgadge
$sql = "SELECT * FROM $STgadget_table_name ";

if (!$resultQ =
    mysqli_query($mysqliConnection, $sql)) {
       printf("Dammit! Can't execute movie sele
query.\n");
       exit();
}
$elenco="";
while ($row = mysqli_fetch_array($resultQ))
  $elenco .= "<input type=\"radio\"
name=\"selection\" value=\"{$row['nome']}\" />
    {$row['nome']} (&euro;
{$row['costoGadget']})<br />\n";

$mysqliConnection->close();
...
<form action="<?php $_SERVER['PHP_SELF']?>"
method="post" >
<table>
...
 <td> <?php echo $elenco; ?> </td><tr></table>
```

Tutte le righe della Stgadget vengono selezionate e stampate nella form

Il resto è come in ST.gadget.php: l'array $_SESSION['carrello'] viene riempito con le selezioni fatte

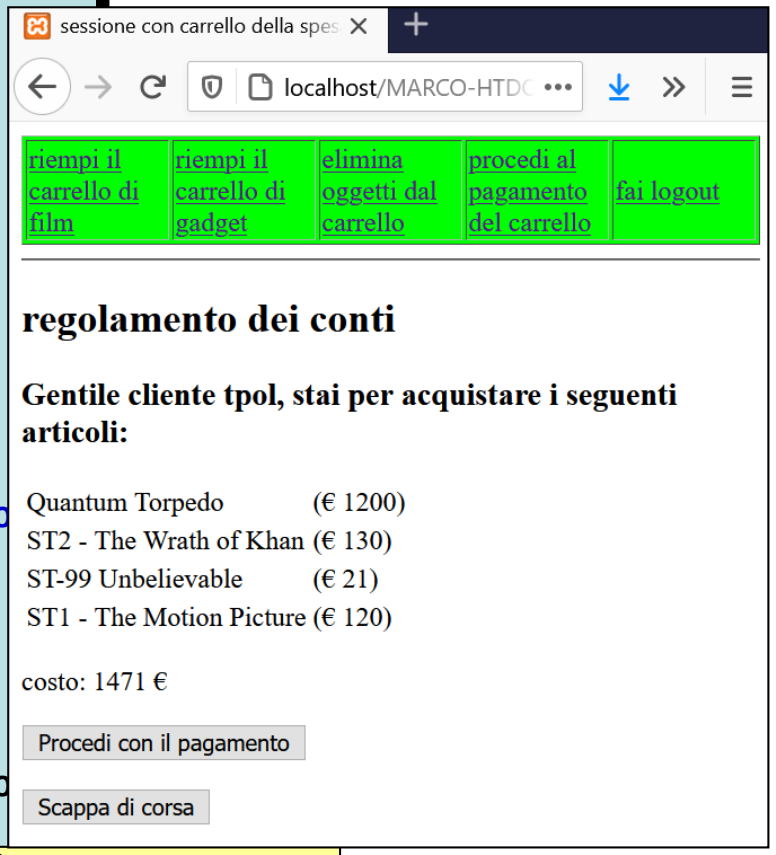


carrello.della.spesa2/

Carr.Spesa2 - 4/8 - mysql.ST.paga.php

```
$outputTable="<h3>Gentile ".$_SESSION['userName
if (!$_SESSION['carrello']) {
   $outputTable.= " che ci ... vuoto?\n";
} else { $outputTable.=", stai per ...
    $outputTable.="<table>\n";
    foreach ($_SESSION['carrello'] as $k=>$v) {
      $outputTable.="<tr>\n<td>$v</td>\n";
       // adesso serve il prezzo ...
 $sql1 = "SELECT * FROM $STmovie_table_name
          WHERE title = \"$v\"";
 $sql2 = "SELECT * FROM $STgadget_table_name
          WHERE nome = \"$v\" ";
 if (!$resultQ =
     mysqli_query($mysqliConnection, $sql1)) {
     …}
 $row1= mysqli_fetch_array($resultQ); // film/vuo
…idem …
 $outputTable.=
    $row2['costoGadget'] + $row1['costoMovie'];
…
$_SESSION['spesaFinorà]+=$row2['costoGadget']+$ro
'costoMoviè];
    }
$outputTable.="</table>\n<p>costo:
{$_SESSION['spesaFinorà]} &euro;</p>\n\n";}
```

è necessaria un'interazione con le tabelle Stgadget e Stmovie, per sapere i prezzi di quel che è nel carrello
Il codice costruisce $outputTable, che poi viene stampata nella parte XHTML



Copia non fedele

carrello.della.spesa2/

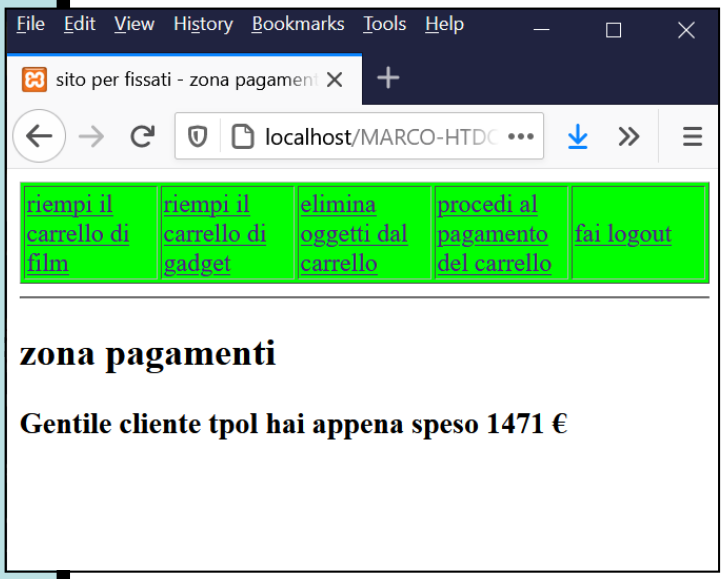
Carr.Spesa2 - 5/8 - zona.pagamenti.php

```
… if ($_POST['invioPagamentò]!="Procedi con il
pagamento") header('Location: …www.startrek.com
...
$outputTable="<h3>Gentile cliente ";
...
if (!$_SESSION['spesaFinora']) {
   $outputTable.= " che ci fai qui con una spes
nulla?\n";
} else {
    $sql1 = "UPDATE $STuser_table_name
    SET sommeSpese=\"{$_SESSION['spesaFinora']}\"
    WHERE userName = \"{$_SESSION['userName']}\"
    ";
if (!mysqli_query($mysqliConnection, $sql1)) {
   printf("Oops, errore nella gestione della query
%s\n", mysqli_error($mysqliConnection));
  exit();
}
// chiusura connessione (versione procedurale)
mysqli_close($mysqliConnection);

if(mysqli_affected_rows($mysqliConnection)==1)
      $outputTable.=" hai appena speso
      {$_SESSION['sommeDaPagarè]} &euro;</h3>\n";

$_SESSION['carrello']=array();  // svuota carr. …
```

è necessaria un'interazione con la tabella Stuser, per modificare il campo sommeSpese dello user
Come prima, viene costruita una stringa $output che poi viene stampata nella parte xhtml



_affected_rows() = quante righe sono state modificate

Controllare I risultati anche nella tabella utenti – capire cosa non va e correggerlo (vedi carrello_della_spesa3)

carrello.della.spesa2/

carrello.della.spesa2 - 6/8 - mysql.ST1.php

```
mysql.ST1.php
$db_name = "STdb";   $STuser_table_name = "STuser";
$STmovie_table_name = "STmovie";
$STgadget_table_name = "STgadget";

$connection = new mysqli("localhost", "archer", "arche
if (mysqli_connect_errno()) {
    printf(«…problemi con la … %s\n",mysqli_error());
    exit();

$queryCreazioneDatabase = "CREATE DATABASE $db_name";
if (mysqli_query($queryCreazioneDatabase, $connection)
{ echo "Database $db_name ok\n";} else { echo 'Error …

$db = mysql_select_db($db_name, $connection)
        or die("Couldn't select database.");

$sql = "CREATE TABLE if not exists $STuser_table_name
$sql.= "userId int NOT NULL auto_increment, primary ke
(userId), ";
$sql.= "userName varchar (50) NOT NULL, ";
$sql.= "password varchar (16) NOT NULL, ";
$sql.= "sommeSpese float";
$sql.= ");";

echo "<p>$sqlQuery</p>";
$resultSTuser = mysql_query($sql,$connection)
        or die("Couldn't create STuser table.");
```

mysqli_connect()
esegue la connessione; se ha successo rest. un intero che identifica la connessione;
parametri:
host sul quale è abilitato l'utente; utente (archer) e password per mysql

mysqli_error() restituisce l'errore occorso (per metterlo in evidenza eventualmemte)

mysqli_select_db() - il db va selezionato per poterlo usare; (parametro connessione, qui e altrove, opzionale)

mysqli_query () esegue sul db la query in argomento; produce un array php, contenente le righe della tabella che sono state selezionate

carrello.della.spesa2/mysql.ST1.php

```
mysql.ST1.php
…
// popolamento STuser (NB tre campi: userId auto
$sql = "INSERT INTO $STuser_table_name
        (username, password, sommeSpese)
        VALUES
        (\"tpol\", password(\"tpol\"), \"0\")
        ";
$result = mysql_query($sql,$connection)
        or die("Couldn't populate STuser table.");
...

// popolamento STmovie (NB tre campi: movieId auto
$sql = "INSERT INTO $STmovie_table_name
        (title, costoMovie)
        VALUES
        (\"ST1 - The Motion Picture\", \"120\")
        ";
echo $sql;
$result = mysqli_query($sql,$connection)
        or die("Couldn't populate STmovie 1table.");
…
mysql_close();
?>
</body></html>
```

`mysqli_close()` chiude la connessione; (succede comunque automaticamente alla fine dello script …)

Tabella Stuser
userId incremento auto
userName, password, sommeSpese

Tabella STmovie movieId incremento Auto, title, costoMovie

Tabella STgadget gadgetId incremento auto, nome, costoGadget

carrello.della.spesa2/mysql.ST1.php

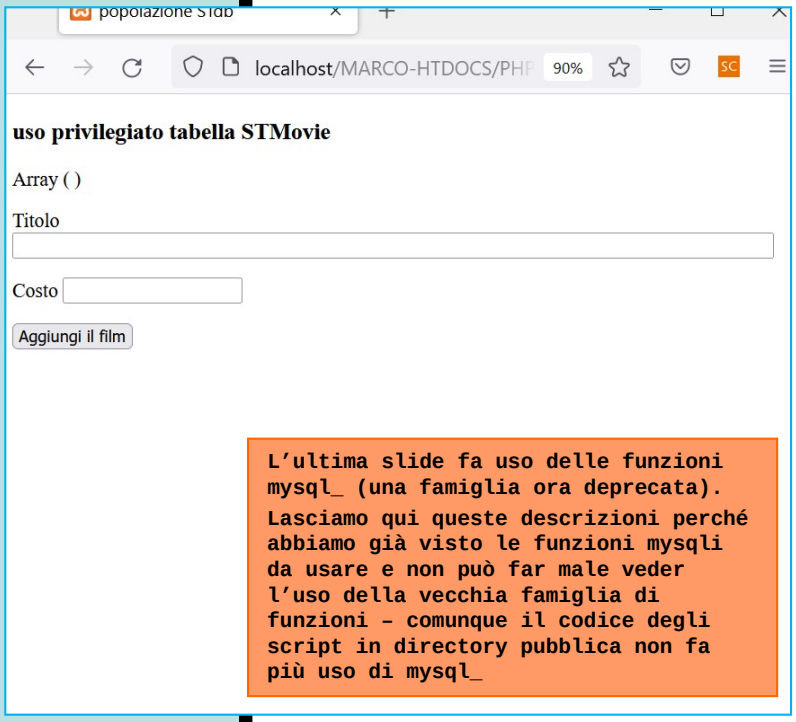
```
mysql.ST2.php
$db_name = "STdb";
$STmovie_table_name = "STmovie";
$connection = @mysql_connect("localhost", "archer", "")
        or die("Couldn't connect.");
$db = mysql_select_db($db_name, $connection)
        or die("Couldn't select database.");

print_r($_POST);
if ($_POST['invio']=="Aggiungi il film" &&
    $_POST['titolo'] && $_POST['costo']) {
  $sql = "INSERT INTO $STmovie_table_name
        (title, costoMovie)
        VALUES
        ('{$_POST['titolo']}',
          '{$_POST['costo']}')  ";

  $result = mysql_query($sql,$connection)
        or die("Couldn't populate STmovie table

  echo "<h3>fatto</h3>\n";
  $_POST['invio']="j";
}
mysql_close();
?>
```

se $_POST['invio'] non è uguale a "Aggiungi" si stampa solo la form; sennò si esegue l'aggiunta e si stampa la form.



L'ultima slide fa uso delle funzioni mysql_ (una famiglia ora deprecata).
Lasciamo qui queste descrizioni perché abbiamo già visto le funzioni mysqli da usare e non può far male veder l'uso della vecchia famiglia di funzioni – comunque il codice degli script in directory pubblica non fa più uso di mysql_

segue (nel file) la stampa della form

carrello.della.spesa2/mysql.ST2.php

mysql
*www.**mysql**.com*

***phpmyadmin**.sourceforge.net*

Attività in laboratorio / prodotti individuali

PHP-6

Una serie di problemi (in particolare warning su variabili non esistenti) proviene dall'uso non controllato delle variabili in uno script.

Controllato in che senso?

Soprattutto quando uno script viene usato sia per fornire una form (e quindi iniziare un'interazione) che per gestire I dati provenienti da quella form, succeed che certe variabili non esistano quando la form è stata viaualizzata ma non ancora inviata, e invece inizino ad esistere solo dopo che dati sono arrivati dalla form (cioè quando lo script riceve quei dati e magari li associa a variabili: ora queste variabili cominciano davvero ad esistere.

Le chiacchiere stanno a zero.

Esaminare il problema proposto da `conteggi.hidden.1.php` e risolverlo.

Riflettere sull'uso di `$_SERVER['PHP_SELF']`.

Riscrivere lo script, modificandolo per farne usi di proprio gusto, e includere nella pagina prodotta (dopo la stampa della form) una discussione della funzione isset() (source: documentazione) e sull'uso di PHP_SELF nella produzione di script che sono progettati per processare i dati che provengono da form che loro stessi hanno prodotto.

PHP-7

A partire da `conteggi.hidden.2.php` riprodurre `conteggi.hidden.3.php` (far girare questo terzo script, leggere le istruzioni e svolgere l'esercizio. Poi confrontare con `conteggi.hidden.3.php`.

PHP-8

Scrivere una pagina iniziale per un sito web, in cui venga contato e visualizzato il numeo delle volte che quel certo utente si è collegato. Usare un meccanismo di invio del proprio nome (o, meglio, username) che usi il meccanismo dei cookie per realizzare il conteggio. Costruire una directory `eser.php.08.01/` contenente la soluzione.

E poi rifare tutto usando il meccanismo di sessione. Costruire una directory `eser.php.08.02/` contenente la soluzione.

Attività in laboratorio / prodotti individuali - 1/2 -

PHP-9

Problema come in PHP-8, ma non vogliamo usare nè cookies nè sessioni.
Costruire una directory `eser.php.09/` contenente la soluzione.

PHP-10

Costruire un sito web che visualizza un rettangolo e del testo;
il rettangolo è dimensionato e colorato in base a scelte fatte dall'utente; almeno il colore della font e quello del background sono anche specificati dall'utente.

L'utente compila una form e vede il risultato.

In una prima versione del sito (costruire una cartella `eser.php.10.01/` per lei), si può evitare di rendere persistenti le scelte.

In una seconda (e ultima) versione (costruire una cartella `eser.php.10.01/` per lei), si aggiunge anche la persistenza delle scelte su più pagine come visto a lezione (usando il meccanismo preferito tra session e cookies).

PHP-11

Descrizione come per l'esercizio precedente (PHP-10) ma stavolta si usa l'altro meccanismo … quello meno preferito).
Costruire una directory `eser.php.11/` contenente la soluzione.

PHP-12

Nella slide sulle mentine … c'è già un esercizio … svolgerlo doppo di questo salvandolo in una directory `eser.php.12.02/`.

Un file `prodotto.txt` contiene, distribuiti come un dato per ogni linea, i dati sul consumo annuale di un certo prodotto, per gli ultimi 6 anni. Scrivere una pagina web che legga dati da quel file e visualizzi un grafico in cui, per ogni anno, sono mostrati il consumo relativo e la media del consumo negli anni precedenti. Il consumo sia visiulizzato numericamente e con una barra di un colore A; le medie siano visualizzate numericamente e con una barra di colore B.
Costruire una directory `eser.php.12.01/` contenente la soluzione.

Attività in laboratorio / prodotti individuali - 2/2 -

PHP-13

Sperimentare con il progettino di carrello della spesa; modificarlo in modo da trasformarlo in qualcosa che si adatti meglio al proprio gusto personale …

Costruire una directory `eser.php.13/` contenente la soluzione.

PHP-14

Sperimentare la creazione di un database minimale, con una tabella, attraverso l'uso di uno script php.

Qui si usa quel che si è visto nel carrello.della.spesa2, ma con ancora meno complessità.

Scrivere uno script che si connetta a mysql e selezioni il db e permetta di aggiungere dati nella tabella. Non c'è da preoccuparsi di validare l'utente. Si attiva lo script e si eseguono le modifiche. Lo script, dopo ogni modifica, deve mostrare il contenuto della tabella. Punto di partenza, forse ST1 …

Usare phpMyAdmin per verificare, dopo ogni immissione, il contenuto della tabella e per aggiungere a mano alcune altre righe alla tabella.

Nelle prime sperimentazioni di questo esercizio, bisognerà connettersi con username root e password vuota (che è lo standard iniziale per mysql). Verificare il contenuto della tabella user del db mysql.

Poi sarà bene aggiungere qualche altro utente (come nel caso di archer, password "archer" del carrello della spesa2.

Costruire una directory `eser.php.14/` contenente la soluzione.

PHP-15

Partire da Carrello della spesa 2.

Definire ed utilizzare uno script `connessione.php`, che gestisca la connessione al db, e sia importato in ogni script del sito web.

Definire ed utilizzare uno script `menu.php`, che si occupi di stampare il menu e venga importato in ciascuno script del sito web.

Queste due cose sono introdotte in carrello dela speasa 3 (vedi quella applicazione web e anche le slide …).

Costruire una directory `eser.php.15/` contenente la soluzione.

PHP-16

Definire una nuova base di dati STdb2, in cui la tabella utenti contenga anche un campo "tipologia", i cui valori possono essere "utente", "gestore", "admin" (potete usare 1001, 1002, 1003 in alternativa). Poi rielaborare l'applicazione web carrello della spesa in modo che ciascuna tipologia di utente abbia il proprio menu visualizzato dopo l'autenticazione.

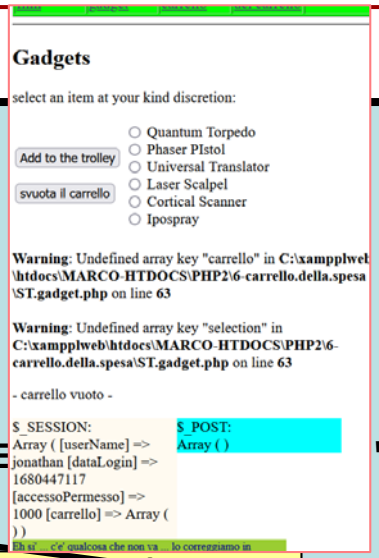
Potete usare menu con funzionalità fittizie. E poi magari sviluppare funzionalità come "aggiunta di nuovi prodotti" (solo gestore), "visualizzazione elenco utenti" (admin), "ban utente" (admin), "visualizzazione elenco acquisti di un utente" (gestore).

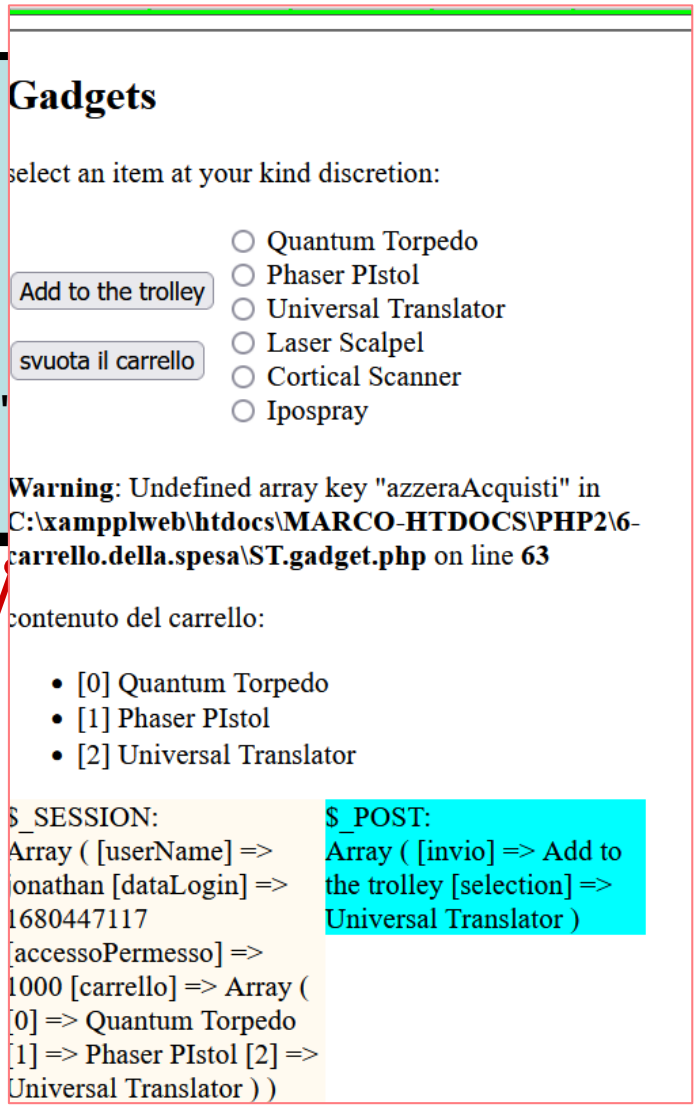
Costruire una directory `eser.php.16/` contenente la soluzione.

PHP - carrello della spesa - 3/5 - (problema posto a lezione)

ST.gadget.php

```
<?php
session_start();
…
if ((!$_SESSION['carrello'] && …
?>
...
<form action="<?php $_SERVER['PHP_SE
<table>
...
```





Come per ST.movie.php, la form action è questo stesso file (le azioni possibili sono distinte in base alle variabili POST e ai bottoni scelti)

alla prima chiamata, nella modalità "hacker del bar qui sotto" "carrello" non è registrata (non esiste). Tuttavia esisterà a partire dalla prima volta che una sua componente viene assegnata.

azzeraAcquisti è valutata in un if uguale a quello mostrato in ST.movie.php. Solo che qui gli errori non sono nascosti (vedi uso di error_reporting() in ST_Movie e ST_gadget ...

- alla prima entrata in ST_gadget.php nè azzeraAcquisti nè selection sono settate; quindi la loro valutazione produce una notice;
- poi, azzeraAquisti è settata solo e soltanto in corrispondenza degli svuotamenti del cestino; e selection è settata solo se abbiamo inviato una scelta

  (sennò è giusto che ci sia un errore, magari trattato dal codice e non mostrato splaid open ...)

carrello.della.spesa/