

Linguaggi

*Corso M-Z - Laurea in Ingegneria Informatica
A.A. 2009-2010*

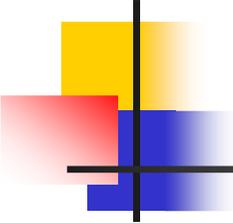
Alessandro Longheu

<http://www.diit.unict.it/users/alongheu>

alessandro.longheu@diit.unict.it

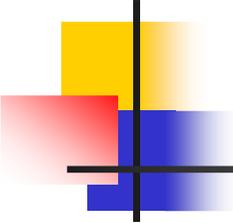
Esercitazione

Programmazione PHP



Form in PHP

- Una form è una pagina web, realizzata in HTML tramite il tag `<form>`, tag che permette alla pagina di diventare interattiva, ossia di contenere elementi (caselle di testo, di spunta o bottoni) il cui stato può essere comunicato ad un codice che lo elabora.
- Il codice, specificato come attributo del form, può essere di qualsiasi natura, nel nostro caso un file php (ma poteva essere un cgi, asp, ecc)
- lo stato degli elementi di interattività può essere comunicato in due modi: GET e POST. Nel GET i valori sono passati insieme nella stringa dell'URL, nel POST i valori sono passati separatamente e quindi non sono immediatamente visibili.
- Notare che il recupero di dati passati da una form è possibile richiamando direttamente il nome della variabile (con `$nome`) solamente se in php.ini la variabile `register_globals` è impostata a on cosa che normalmente non è. In caso `register_globals` sia impostato a off il recupero è possibile solamente nella forma `$_POST[nome_variabile]` o `$_GET[nome_variabile]`



Form in PHP

- Esempio di form con diversi elementi di interattività:

```
<html><body>
```

```
<form method=get action="visualizza.php">
```

```
<center>
```

```
Inserisci il tuo nome:<input type="text" name="nome" size="30"><br>
```

```
Inserisci cognome:<input type="text" name="cognome"
size="30"><br>
```

```
Inserisci la tua e-mail:<input type="text" name="email"
size="30"><br><br><br><br>
```

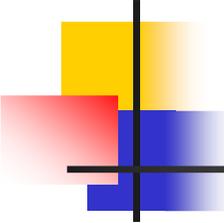
```
Scegli quale sezione vorresti che ci fosse nel sito:<br>
```

```
<input type="checkbox" name="sport" value="sport">sport
```

```
<input type="checkbox" name="cinema" value="cinema">cinema
```

```
<input type="checkbox" name="donne" value="donne">donne
```

```
<input type="checkbox" name="forum" value="forum"
checked>forum<br>
```



Form in PHP

- ...continua

```

<br>Scegli la connessione che usi<br>
<input type="radio" name="linea" value="56k"
checked>56k
<input type="radio" name="linea" value="isdn">isdn
<input type="radio" name="linea" value="adsl">adsl
<input type="radio" name="linea"
value="56k">fastweb<br>
<br>Inserisci ora i tuoi commenti<br>
<textarea name="commenti" rows="7"
cols="70"> </textarea>
<br><br><br><input type="submit" value="invia i dati">
<input type="reset" value="cancella">
</form><hr></body></html>

```

Form in PHP

- Il risultato è il seguente:

Inserisci il tuo nome:

Inserisci il tuo cognome:

Inserisci la tua e-mail:

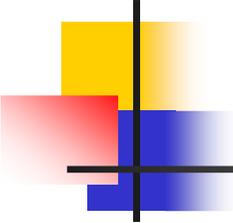
Scegli quale sezione vorresti che ci fosse nel sito:

sport cinema donne forum

Scegli la connessione che usi

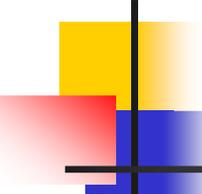
56k isdn adsl fastweb

Inserisci ora i tuoi commenti



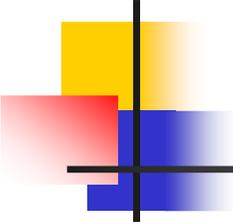
Form in PHP

- Il **metodo get** impostato nella form prevede che i nomi ed i valori da associare siano posti direttamente sul link che richiama la pagina.
La sintassi è: ``
- Se si vogliono inserire più nomi e valori, occorre `&` fra le coppie di parametri, ad esempio:
``
- eventuali spazi nei valori sono convertiti in `%20` o nel carattere `+`
- Dopo aver inserito questa riga (che per convenzione si chiama query string) abbiamo a disposizione un array di nome `$_GET` che contiene tutti i parametri che abbiamo inserito nella query string ed è visibile anche dalla nuova pagina.



Form in PHP

- Per chiarire, supponiamo per esempio di scrivere una riga come questa:
- ``
- Quando il codice arriva al server abbiamo a disposizione le variabili:
- `$ciao=2; $ciao2=4;`
- E le possiamo usare per generare la pagina `index.php` che vedrà l'utente sul suo browser.
- Se per motivi di sicurezza il server ha impostato `register_globals` su `OFF` non possiamo usare direttamente le variabili (con il loro nome) se non dopo averle ricavate dall'array `$_GET` nel seguente modo (nella pagina di destinazione, nel nostro esempio `index.php`):
`$ciao=$_GET['ciao']; $ciao2=$_GET['ciao2'];`
- Adesso possiamo usare le due variabili `$ciao=2` e `$ciao2=4`



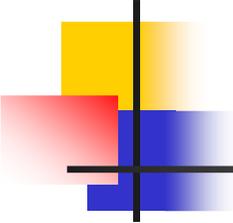
Form in PHP

- Tornando alla form, il codice di visualizza.php (lato "server") potrebbe essere:

```

<html>
<body>
<center>Riepilogo</center>
<?php
echo "Ciao <b>$nome $cognome</b><br>";
echo "Questa e la tua email <b>$email</b><br>";
echo "<br> Sezioni che vorresti(puoi anche non sceglierne):<br>";
echo "<b>$sport $cinema $donne $forum</b><br> ";
echo"<br> Tu navighi con la seguente connessione
   :<b>$linea</b><br>";
echo "<br>ecco i tuoi commenti:<br><b>$commenti</b>";
?>
</body>
</html>

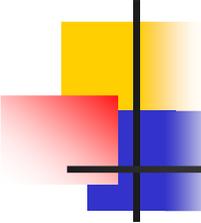
```



Form in PHP

- I form possono puntare anche a se stessi invece che puntare ad un'altra pagina:

```
<HTML>  
<BODY>  
<CENTER>  
<?php  
echo "<FORM METHOD=POST ACTION=|'$PHP_SELF'|">";  
?>  
<INPUT TYPE="SUBMIT" VALUE="Invia le informazioni!">  
<INPUT TYPE="RESET" VALUE="Cancella!">  
</FORM>  
</BODY>  
</HTML>
```

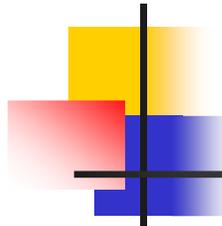


Form in PHP

- Come usare però al meglio il PHP_SELF?. Possiamo creare una pagina con un if. Cioè se il contenuto di una variabile è vuoto allora mostra il form altrimenti mostra un'altra cosa.

```
<HTML><BODY><?php
if ( (empty($nome) || (empty($cognome))) {
# esegue questa parte perché il form non è stato ancora inviato
echo "<FORM METHOD=POST ACTION=\"\$PHP_SELF\">";
echo "Nome? <BR> <INPUT TYPE=\"TEXT\" NAME=\"nome\">";
echo "Cognome? <BR><INPUT TYPE=\"TEXT\" NAME=\"cognome\">";
echo "<BR><BR><INPUT TYPE=\"SUBMIT\" NAME=\"submit\" VALUE=\"Invia
le informazioni!\">";
echo "<INPUT TYPE=\"RESET\" VALUE=\"Cancella!\">";
echo "</FORM>";
} else { # il form è stato inviato
echo "elaborazione...<BR>";
echo "Le informazioni sono... nome: \$nome , cognome: \$cognome";
}
?></BODY></HTML>
```

- Quindi se la variabile nome o cognome sono vuote mostra il form



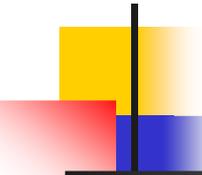
Form in PHP

- Per verificare che un form è stato inviato si può anche usare un campo nascosto, come indicato nell'esempio seguente:

```

<HTML><BODY>
<?php
# metto tutto il form in una variabile
$form="<FORM METHOD=|"GET|" ACTION=|"$_PHP_SELF|">
<CENTER><B><H3> Nuovo utente </H3></B></CENTER>
Nome utente<BR>
<INPUT TYPE=|"TEXT|" NAME=|"nome|" VALUE=|"$_nome|"><BR>
Indirizzo<BR>
<INPUT TYPE=|"TEXT|" NAME=|"indirizzo|" VALUE=|"$_indirizzo|"><BR>
<BR>
# ecco il campo nascosto
<INPUT TYPE=|"HIDDEN|" NAME=|"stato|" VALUE=|"inviato|">
<INPUT TYPE=|"SUBMIT|" NAME=|"submit|" VALUE=|"Invia |">
<INPUT TYPE=|"RESET|" VALUE=|"Cancella!|">
</FORM>";

```

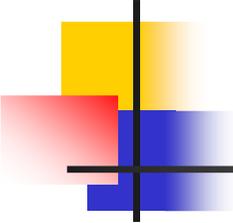


Form in PHP

```

# inizio a controllare
$errore_nome=0;
$errore_indirizzo=0;
# il form è stato inviato?
# se il campo nascosto non c'e mostra il form
if($stato!="inviato"){          echo "$form";
#ma se c'e controlli l'input
} else {
if (empty($nome)) {
echo "<B>Errore:</B> Manca nome <BR>"; $errore_nome=1; }
if (empty($indirizzo)) {
echo "<B>Errore:</B> Manca l'indirizzo <BR>";$errore_indirizzo=1; }
if (($errore_nome) || ($errore_indirizzo)) {          echo "$form";
} else {
echo "elaborazione... <BR>";
echo "<B>$nome</B> con indirizzo <B>$indirizzo</B> è stato aggiunto";
}}
?></BODY></HTML>

```



Form in PHP

- Si possono realizzare form a più pagine con passaggi di informazioni nei campi nascosti, ad esempio il primo form chiede nome e indirizzo, il secondo chiede altre info e nasconde nome e indirizzo e il terzo visualizza tutti i dati:

```
<HTML>
```

```
<BODY>
```

```
<FORM METHOD=GET ACTION="due.php">
```

```
<HR>
```

```
<B>Qual è il tuo nome completo?</B><BR><INPUT TYPE="text"
NAME="nome" SIZE=20>
```

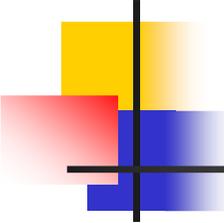
```
<BR><BR>
```

```
<B>Qual è il tuo indirizzo email?</B><BR><INPUT TYPE="text"
NAME="email" SIZE=20>
```

```
<BR><BR>
```

```
<B>Qual è il tuo numero di telefono?</B><BR><INPUT
TYPE="text" NAME="telefono" SIZE=20>
```

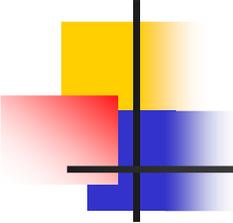
```
<BR><BR>
```



Form in PHP

- ...continua...

```
<B>Come desideri effettuare il pagamento?</B> <BR>
<SELECT NAME="pagamento" >
<OPTION SELECTED>Carta di Credito
<OPTION>Assegno
<OPTION>Bonifico Bancario
<OPTION>Contanti
</SELECT>
<HR>
<INPUT TYPE="submit" VALUE="Pagina seguente">
<INPUT TYPE="reset" VALUE="Cancella!">
</FORM>
</BODY>
</HTML>
```



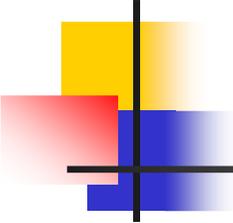
Form in PHP

- La seconda form riceve i campi della prima sull'URL con il metodo GET, e li propaga alla terza form tramite campi nascosti, così saranno insieme ai campi non nascosti di questa seconda form (quelli relativi alle informazioni aggiuntive), tutti propagati con GET :

```

<HTML><BODY>
<CENTER><H2>Seleziona le opzioni di consegna</H2></CENTER>
<FORM METHOD=GET ACTION="tre.php">
<B>Tipo di servizio:</B><BR>
<SELECT NAME="tipo_consegna" >
<OPTION SELECTED>Gold Star
<OPTION>Silver Star
<OPTION>Piccione viaggiatore
</SELECT><BR><BR>
<B>Giorno di ritiro del plico:</B><BR>
<SELECT NAME="giorno">
<OPTION SELECTED>Lunedì
<OPTION>Martedì<OPTION>Mercoledì<OPTION>Giovedì
</SELECT>

```



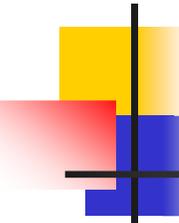
Form in PHP

- ...continua...

```

<?php
# innanzitutto codifica i tre valori nascosti
$nome=(urlencode($nome));
$email=(urlencode($email));
$telefono=(urlencode($telefono));
# ecco i campi nascosti all'interno del form
echo "<INPUT TYPE=|\"HIDDEN|\" NAME=|\"nome|\" value=$nome>\"";
echo "<INPUT TYPE=|\"HIDDEN|\" NAME=|\"email|\" value=$email>\"";
echo "<INPUT TYPE=|\"HIDDEN|\" NAME=|\"telefono|\" value=$telefono>\"";
echo "    <INPUT          TYPE=|\"HIDDEN|\"          NAME=|\"pagamento|\"
value=$pagamento>\"";
?>
<HR>
<INPUT TYPE="submit" VALUE="Pagina seguente">
<INPUT TYPE="reset" VALUE="Cancella!">
</FORM></BODY></HTML>

```



Form in PHP

- infine la terza form riceve e stampa tutto:

```

<HTML><BODY>
<H2>Consegne Spa ti ringrazia</B>
<TABLE BORDER=6 WIDTH=60% COLS=1>
<TR><TH>Ecco le informazioni da te indicate</TH></TR>
<TR><TD>
<?php
$name=(urldecode($nome));
$email=(urldecode($email));
$telefono=(urldecode($telefono));
echo "Il tuo nome è: <B>$name</B><BR>";
echo "Il tuo indirizzo email è: <B>$email</B><BR>";
echo "Il tuo numero di telefono è: <B>$telefono</B><BR>";
echo "Il metodo di pagamento è: <B>$pagamento</B><BR>";
echo "Il tipo di servizio selezionato è: <B>$tipo_consegna</B><BR>";
echo "Ritireremo il plico di: <B>$giorno</B><BR>";
echo "Il peso del plico è: <B>$peso</B><BR>";
?>
</TD></TR></TABLE></BODY></HTML>

```

PHP – collegamento con MySQL

- ▶ Spesso PHP viene utilizzato insieme ad un database, perché il primo permette la creazione di pagine dinamiche, mentre il secondo consente la memorizzazione dei dati, che spesso sono parecchi (tanto da richiedere un database, appunto)
- ▶ Prima di tutto, però, avremo bisogno di alcune informazioni relative all'accesso al database: l'host da cui si può raggiungere MySQL (generalmente è localhost); username e password per l'accesso al database; il nome del database. Questi quattro parametri vengono forniti dall'amministratore del nostro spazio web. Quindi creeremo una pagina con queste righe:

```
<?
```

```
// parametri del database  
$db_host = "localhost";  
$db_user = "";  
$db_password = "";  
$db_name = "";
```

```
?>
```

PHP – collegamento con MySQL

- ▶ La connessione al database si realizza attraverso la funzione `mysql_connect`. L'istruzione successiva ai parametri quindi sarà:

```
$db = mysql_connect($db_host, $db_user, $db_password);
```

- ▶ La funzione `mysql_connect` richiede diversi parametri, di cui generalmente si utilizzano i primi tre, che sono proprio quelli che abbiamo richiesto all'utente. Se la connessione ha buon fine ci restituisce un identificatore alla connessione che noi memorizziamo in `$db`.

PHP – collegamento con MySQL

- ▶ Se la connessione non dovesse andare a buon fine (per esempio se uno dei parametri fosse sbagliato) in `$db` verrebbe restituito `FALSE`. Quindi dovremo verificare il buon esito della connessione aggiungendo di seguito:

```
if ($db == FALSE)  
die ("Errore nella connessione. Verificare i parametri nel file  
config.inc.php");
```
- ▶ Grazie a queste righe se la connessione dovesse fallire, otterremmo il messaggio di errore e l'interruzione dell'esecuzione del programma.
- ▶ Fatto tutto questo dobbiamo specificare su quale database vogliamo lavorare e verificare nuovamente la riuscita dell'operazione:

```
mysql_select_db($db_name, $db)  
or die ("Errore nella selezione del database. Verificare i  
parametri nel file config.inc.php");
```
- ▶ In questo caso non abbiamo bisogno di memorizzare alcun valore, visto che la funzione restituisce solo `TRUE` o `FALSE`.

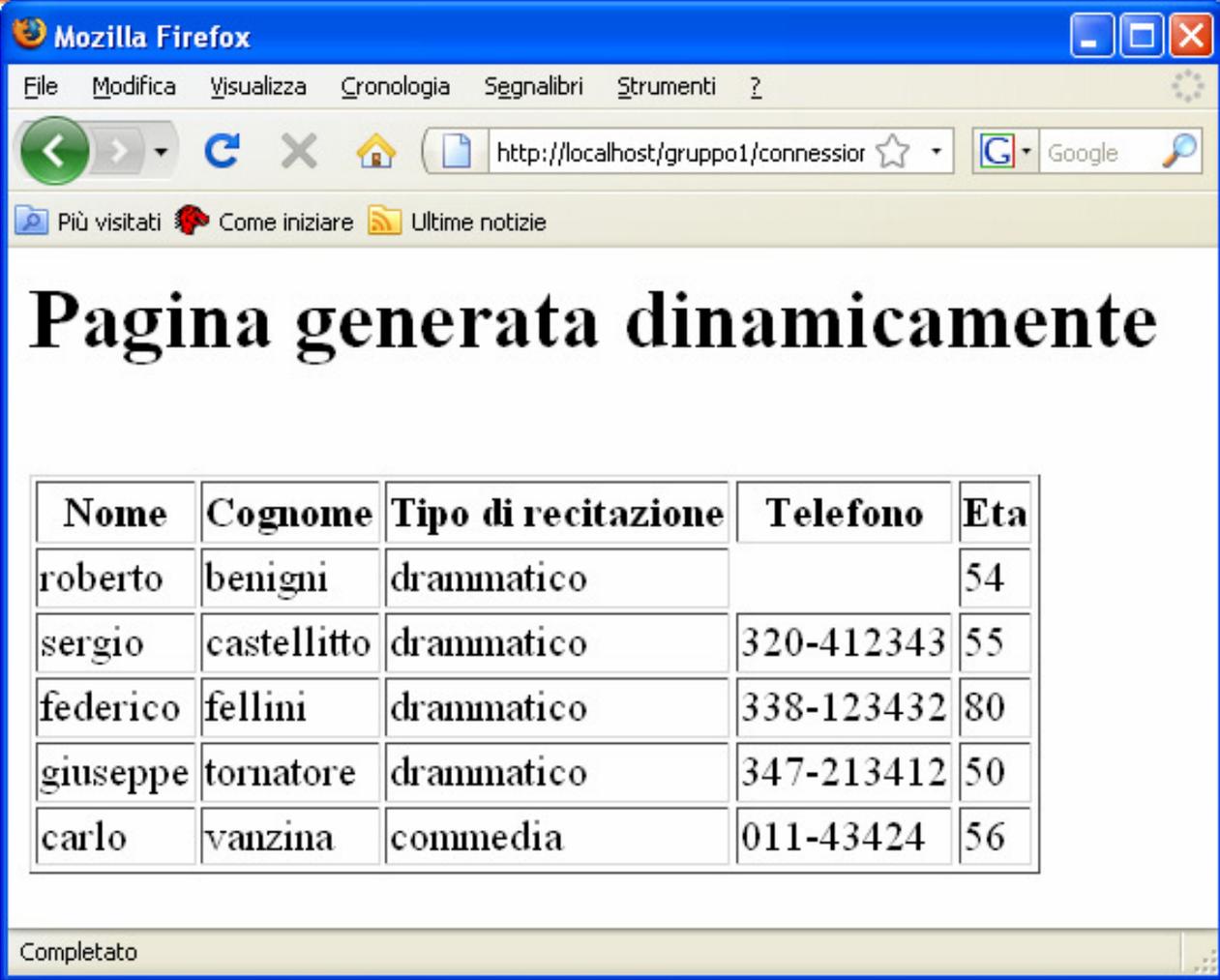
PHP – collegamento con MySQL

- ▶ Dopo aver creato la connessione possiamo finalmente agire sul database tramite invio di query. Prendendo ad esempio un database sui cinema:

```
$query = "SELECT * from regista order by cognome";
$result = mysql_query($query, $db);
echo "<html><h1>Pagina generata dinamicamente</h1><br>";
echo "<table border=1>";
echo "<th>Nome</th><th>Cognome</th><th>TipoR</th>
      <th>Tel</th><th>Eta</th>";
while ($row = mysql_fetch_array($result))
{
    echo "<tr>";
    echo "<td>$row[Nome]</td>";
    echo "<td>$row[Cognome]</td>";
    echo "<td>$row[tipoR]</td>";
    echo "<td>$row[tel]</td>";
    echo "<td>$row[eta]</td>";
    echo "</tr>";
}
echo "</table></html>";
mysql_close($db);
```

PHP – collegamento con MySQL

Il risultato è:



The screenshot shows a Mozilla Firefox browser window displaying a page titled "Pagina generata dinamicamente". The page contains a table with the following data:

Nome	Cognome	Tipo di recitazione	Telefono	Eta
roberto	benigni	drammatico		54
sergio	castellitto	drammatico	320-412343	55
federico	fellini	drammatico	338-123432	80
giuseppe	tornatore	drammatico	347-213412	50
carlo	vanzina	commedia	011-43424	56

Completato

PHP – collegamento con MySQL

- ▶ Con l'esempio non viene evidenziato l'aspetto dinamico, quindi integriamo una form con la connessione al db:

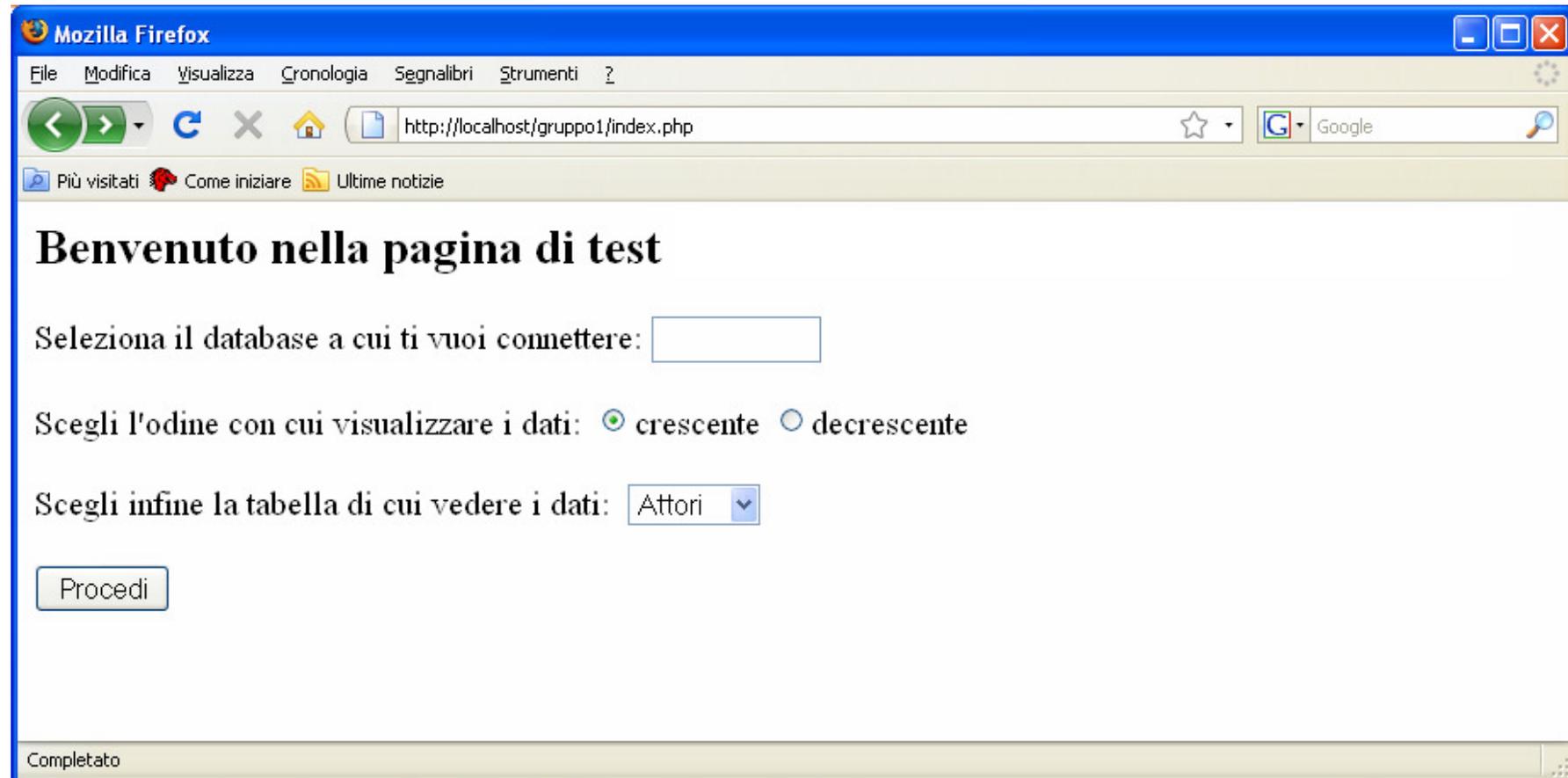
```

<html>
<h2>Benvenuto nella pagina di test</h2>
<form name="modulo" method="post" action="connessione_person.php">
  Seleziona il database:&nbsp;  <input type="text" name="db_name"><br><br>
  Scegli l'ordine con cui visualizzare i dati:
  <input type="radio" name="ordine" value="" checked>crescente
  <input type="radio" name="ordine" value="DESC">decrescente
  <br><br>
  Scegli infine la tabella di cui vedere i dati:&nbsp;  
  <SELECT NAME="nometabella" >
  <OPTION value="attore" SELECTED>Attori
  <OPTION value="regista">Registi
  </SELECT><br><br>
  <input type="submit" value="Procedi">
</form>

```

PHP – collegamento con MySQL

- ▶ La pagina in html (niente php, verra' richiamato dopo la pressione del tasto ed e' un'altra pagina):



PHP – collegamento con MySQL

- ▶ La form richiamerà una pagina in php che utilizza i valori scelti nella prima pagina per personalizzare la query e quindi creare una pagina "dinamica":

```
<?php
// parametri del database
$db_host = "localhost";
$db_user = "root";
$db_password = "";
$db_name = $_POST['db_name'];
// NOTA BENE: $_POST['db_name'] è scelto nella pagina precedente
$db = mysql_connect($db_host, $db_user, $db_password);
if ($db == FALSE)
    die ("Errore nella connessione. Verificare i parametri");
mysql_select_db($db_name, $db)
    or die ("Errore nella selezione del database. Verificare i
    parametri");...
```

PHP – collegamento con MySQL

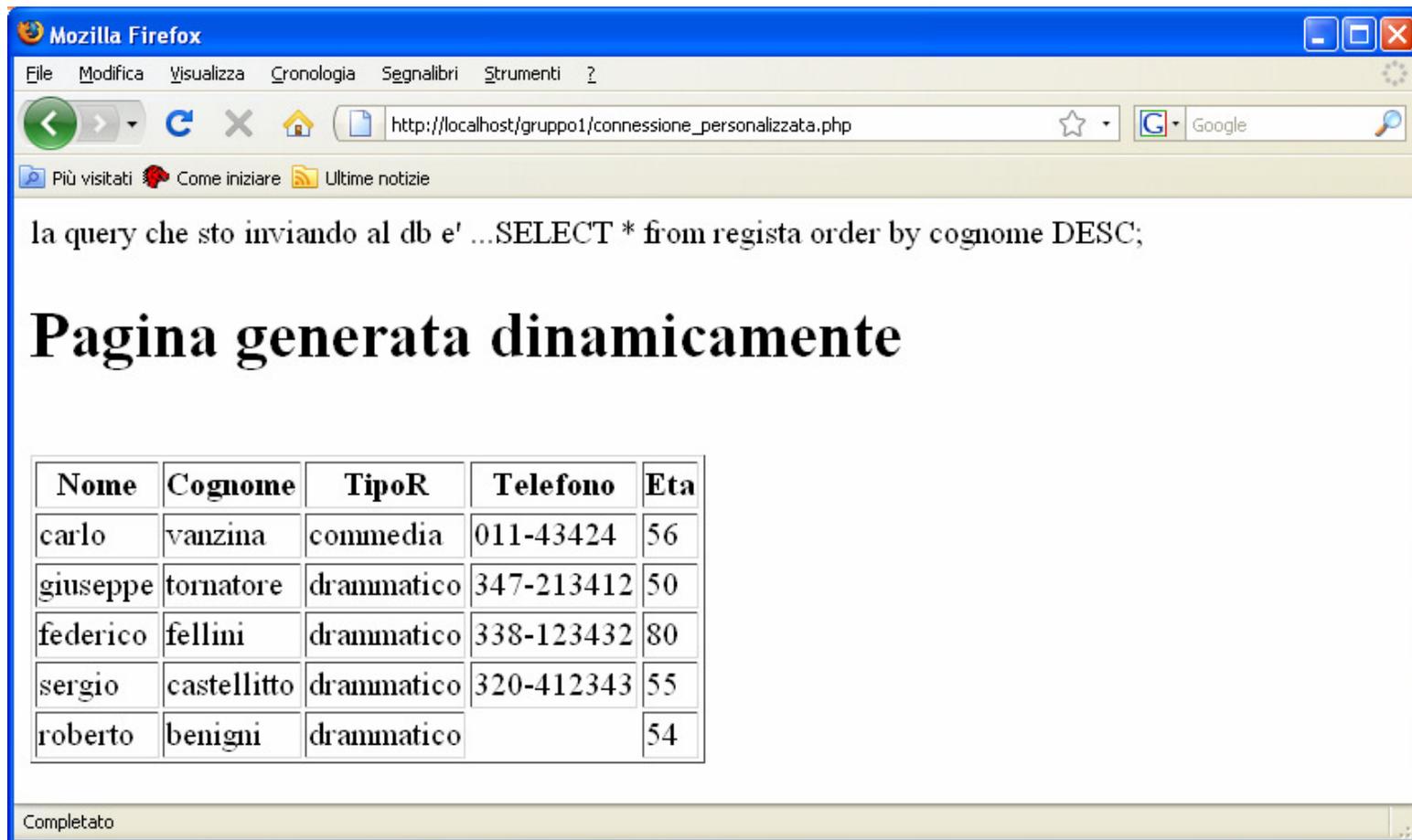
```

...
$query = "SELECT * from ".$_POST['nometabella']." order by cognome
        ".$_POST['ordine'].>";";
echo "la query che sto inviando al db e' ... ".$query;
$result = mysql_query($query, $db);
echo "<html><h1>Pagina generata dinamicamente</h1><br>";
echo "<table border=1>";
echo "<th>Nome</th><th>Cognome</th>
      <th>TipoR</th><th>Telefono</th><th>Eta</th>";
while ($row = mysql_fetch_array($result))
{
  echo "<tr>";
  echo "<td>$row[Nome]</td>";
  echo "<td>$row[Cognome]</td>";
  echo "<td>$row[tipoR]</td>";
  echo "<td>$row[tel]</td>";
  echo "<td>$row[eta]</td>";
  echo "</tr>"; }
  echo "</table></html>";
  mysql_close($db);
  ?>

```

PHP – collegamento con MySQL

- ▶ Il risultato potrebbe ad esempio essere:

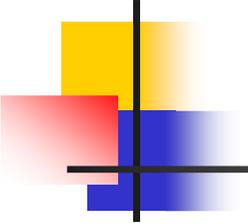


la query che sto inviando al db e' ...SELECT * from regista order by cognome DESC;

Pagina generata dinamicamente

Nome	Cognome	TipoR	Telefono	Eta
carlo	vanzina	commedia	011-43424	56
giuseppe	tornatore	drammatico	347-213412	50
federico	fellini	drammatico	338-123432	80
sergio	castellitto	drammatico	320-412343	55
roberto	benigni	drammatico		54

Completato



PHP – Esercizi

Scrivere uno script PHP che produca in output:

- La tabellina di un numero ricevuto in input tramite un modulo. Lo script deve verificare che l'utente abbia inserito realmente un numero, in caso contrario deve segnalare l'errore
- Il fattoriale di un numero ricevuto in input tramite un modulo. Lo script deve verificare che il numero si compreso tra 1 e 15, in caso contrario deve segnalare l'errore
- Data una stringa da una form, la frequenza con cui ogni carattere appare in una stringa inserita in una form

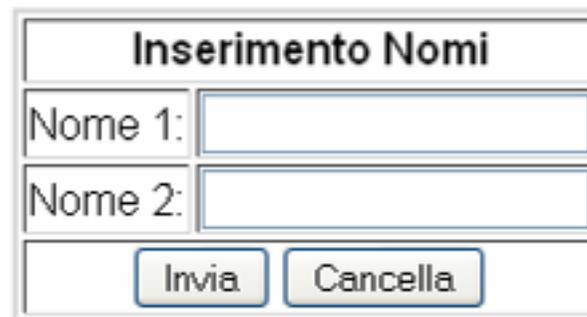
PHP - Esercizi

- Creare la seguente form



Numero di elementi da inserire:

- Quando si preme "Invia", viene creata una nuova form con numero di campi uguale al numero inserito nel campo di testo. Ad esempio, se abbiamo inserito 2, allora sarà creata la form:



Inserimento Nomi

Nome 1:

Nome 2:

- Inserire i nomi in un array, salvare i nomi in un file di testo e visualizzarli a video ordinati in base alla lunghezza.