

Capitolo 3 14/02/09

*IPSLA “Antonio Pacinotti” - Pistoia*

# Corso sul linguaggio PHP

*Gestione Web di un Magazzino*



III Area

**Classe V** - Gestione Web di un  
magazzino di materiale elettronico

Gualtiero Lapini

# Corso sul linguaggio PHP

Terminata la parte introduttiva sul linguaggio PHP, possiamo adesso alla connessione con pagine HTML per lo scambio di dati, facciamo cioè un primo passo verso la costruzione di pagine HTML “davvero” dinamiche ovvero modificabili tramite la scelta dell'utente.

Trattiamo in questo capitolo delle pagine di inserimento dati da parte dell'utente ovvero dei FORM.

## I Form di inserimento dati

Esiste un tag particolare nel linguaggio HTML che serve per l'inserimento dei dati da parte dell'utente, cioè di colui che sta utilizzando il browser, e la successiva “spedizione” degli stessi al server.

Vediamo un primo esempio molto semplice:

```
<form action="ese10.php" method="post">
  <b>PRENOTAZIONE LIBRO</b><br><br>
  Inserire i dati seguenti<br><br>
  Cognome: <input type="text" name="cognome" size="40"><br>
  Nome: <input type="text" name="nome" size="40"><br>
  Titolo Libro: <input type="text" name="titolo" size="40"><br>
  <br><br>
  <input type="submit" value="ACCETTA">
  <input type="reset" value="CANCELLA">
</form>
```

La pagina che stiamo guardando è la pagina **ese10.html**, è scritta quindi in puro linguaggio HTML, all'interno della sezione **<body>** proviamo ad inserire queste righe e vediamo a video il risultato ottenuto: come possiamo notare, oltre alle scritte compaiono tra campi di testo, all'interno dei quali possiamo scrivere quello che vogliamo, ed anche due pulsanti che possiamo premere.

Cerchiamo di capire che cosa esattamente fa il codice che abbiamo scritto.

Abbiamo il tag di apertura **<FORM>**, chiuso con **</FORM>** all'interno del quale possiamo inserire alcuni altri comandi.

Il tag **<FORM>** prevede una proprietà **ACTION** che indica quale programma PHP utilizzerà i dati inseriti in questo form ed una proprietà **METHOD** fissata a **POST** che indica con quale metodo la pagina HTML utilizzerà i dati. Useremo solo questo metodo anche se non è l'unico, con **POST** i dati vengono **inviati** alla pagina PHP che poi li tratterà.

I campi di inserimento dati possono essere molto diversi, in questo esempio vediamo solo un campo di tipo testo. Il tag **<INPUT>**, che non ha un tag di chiusura, segnala l'uso di un campo di input dati, la proprietà **TYPE="TEXT"** indica che il tipo usato è una riga di inserimento testo, la proprietà **SIZE** imposta le dimensioni grafiche del campo, un eventuale proprietà **VALUE="Testo iniziale"** imposta il testo che all'inizio sarà inserito nel riquadro, per default il riquadro è vuoto. Esiste infine una

### PRENOTAZIONE LIBRO

Inserire i dati seguenti

Cognome:

Nome:

Titolo Libro:

proprietà importantissima che è **NAME=**, questa proprietà permette di assegnare un nome di riferimento ai dati inseriti dall'utente, questo nome sarà sfruttato poi dal programma PHP per gestire correttamente i vari campi.

Si rimanda comunque alla documentazione HTML per tutte le spiegazioni necessarie per l'uso di molte altre proprietà che qui non tratteremo.

Al termine del codice HTML troviamo anche la definizione dei due pulsanti, anche in questo caso viene utilizzato il tag `<INPUT>` però come **TYPE** viene indicato un valore di **RESET** nel caso del pulsante che cancella il contenuto di tutti i campi di input testo ed un valore di **SUBMIT** per il pulsante che preleva tutti i contenuti inseriti dall'utente e li "posta" (cioè li invia) al programma PHP indicato in precedenza e che questi dati dovrà elaborare.

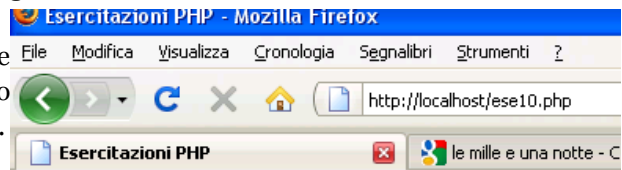
In questo caso la proprietà **VALUE** permette di impostare il testo contenuto all'interno del pulsante.

La pagina HTML è quindi pronta, vediamo adesso come andrà scritta la pagina PHP che dovrà gestire questi dati.

```
<?php
// Recupera i dati dal form
$nominativo = $_POST['nome']; // <==== Stesso nome utilizzato nel form di chiamata
$libro = $_POST['titolo']; // <==== Stesso nome utilizzato nel form di chiamata
// Utilizza il dato recuperato
echo "Benvenuto: <b>$nominativo</b><br>";
echo "Hai prenotato il libro: <b>$libro</b><br>";
?>
```

Come possiamo vedere esiste una apposita istruzione, **\$\_POST**, che provvede a ricevere i dati del campo indicato fra le parentesi quadre. Il nome del campo inserito fra apici e parentesi quadre deve essere il medesimo che abbiamo utilizzato nella pagina HTML.

Nell'esempio riportato prendiamo solo il nome (per ragioni di privacy ad esempio) e lo visualizziamo insieme al titolo del libro prenotato.



Come abbiamo visto, premendo il pulsante **ACCETTA**, viene in automatico aperta la pagina **ese10.php** la quale a sua volta, utilizzando i dati ricevuti, ha creato tramite istruzioni HTML la pagina che verrà infine visualizzata dal browser.

Esistono vari altri tag `<INPUT>` che permettono di accettare altri tipi di input da parte dell'utente.

Esiste cioè il controllo di tipo **RADIO button** che se, come il pulsante tipo selezione stazione nelle radio riceventi, viene impostata una scelta, automaticamente vengono deselezionate le altre. Graficamente viene visualizzato come un cerchietto che può essere selezionato.

Esiste il controllo di tipo **CHECKBOX** ovvero come una casella di spunta, si presenta graficamente come un quadratino al cui interno può o meno comparire il segno di spunta. In questo caso la scelta di una casella non esclude le altre.

Esiste infine il controllo di tipo **SELECT** che si presenta come una casella con selezione di una voce tra un elenco di voci che si apre a discesa premendo su di un piccolo pulsante con un triangolo (altrimenti detto *elenco a tendina*).

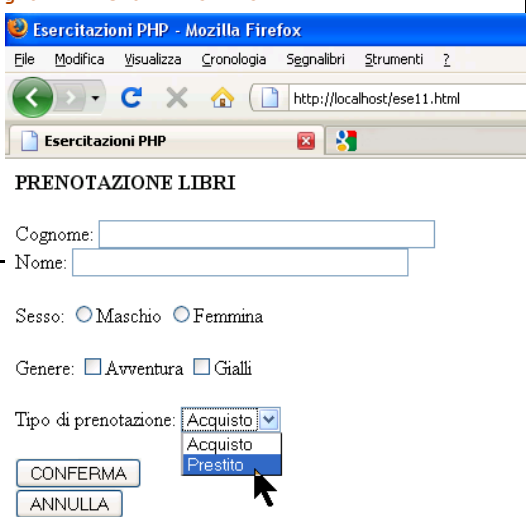
Esistono anche altri controlli che in questo corso, per ragioni di spazio e di tempo, non verranno trattati.

Proviamo un altro programma con tutti questi esempi appena indicati. Vediamo il listato del programma **ese11.html** che li raggruppa tutti quanti.

```

<form action="ese11.php" method="post">
  Cognome: <input type="text" name="cognome" size="40"><br>
  Nome: <input type="text" name="nome" size="40"><br><br>
  Sesso: <input type="radio" name="sesso" value="MASCHIO">Maschio
        <input type="radio" name="sesso" value="FEMMINA">Femmina<br><br>
  Genere: <input type="checkbox" name="c1" value="avventura">Avventura
        <input type="checkbox" name="c2" value="gialli">Gialli<br><br>
  Tipo di prenotazione:
  <select name="tipo">
    <option value="Acquisto">Acquisto</option>
    <option value="Prestito">Prestito</option>
  </select>
  <br><br>
  <input type="submit" value="CONFERMA"><br>
  <input type="reset" value="ANNULLA">
</form>

```



A video comparirà la schermata visibile qui a lato.

Guardiamo adesso come fare con il codice PHP ad utilizzare i dati inseriti dall'utente in questa pagina HTML di tipo FORM.

```

<?php
  $a = $_POST['cognome'];
  $b = $_POST['nome'];
  $c = $_POST['sesso'];
  if( $c == "FEMMINA" )
  {
    $titolo = "Sig.ra ";
  }
  else
  {
    $titolo = "Sig.";
  }
  echo "<br><br>Buongiorno<br>";
  echo "<i>$titolo</i>";
  echo "<b>$a</b>." " " "$b".<br><br>";
  echo "<i>Generi prenotati</i><br>";
  if ( $_POST['c1'] == "avventura" ) { echo "<b>Romanzi di Avventura</b><br>"; }
  if ( $_POST['c2'] == "gialli" ) { echo "<b>Romanzi Gialli/Polizieschi</b><br>"; }
  echo "<i>Tipo di prenotazione:</i><br>";
  if ( $_POST['tipo'] == "Acquisto" )
  { echo "<b>Acquisto libro</b><br>"; }
  else
  { echo "<b>Prestito libro</b><br>"; }
?>

```

Analizziamo meglio il codice: i dati ricevuti come cognome e nome sono inseriti rispettivamente nelle variabili **\$a** e **\$b**, il sesso indicato dell'utente viene inserito in una variabile **\$c** che viene successivamente utilizzata in un controllo di tipo **IF/ELSE** per determinare il titolo da anteporre al nominativo del cliente. Se viene scelto *femmina* nella variabile **\$titolo** viene memorizzata la parola *Signora*, altrimenti viene memorizzato *Signore*.

Queste variabili sono utilizzate poi per comporre una riga di saluto al cliente.

Vengono poi elencati i generi di libro che il cliente ha selezionato per la prenotazione. In sequenza vengono controllate le variabili di nome **c1** e **c2** proveniente dalla pagina HTML di chiamata e passati a questa pagina PHP con il metodo POST.

Viene infine controllato il contenuto della variabile tipo, anch'essa proveniente dal form di inserimento dati tramite il metodo POST.

Vediamo nelle due figure soprastanti il risultato visibile sul browser, sia nella pagina di introduzione dei dati, un attimo prima di premere il pulsante di CONFERMA, ed in seguito sulla pagina generata dal PHP.

Come si può vedere siamo finalmente riusciti a realizzare delle pagine “dinamiche”, che cambiano aspetto in base alle scelte dell'utente che naviga in esse con qualsiasi tipo browser.

Concludiamo questa terza lezione con alcuni spunti.

Con questo frammento di codice si vede che il controllo SELECT, utilizzando la proprietà **MULTIPLE**, può assumere anche la forma di una list-box, cioè una lista o elenco di voci inserite in un riquadro, anziché in una *tendina*.

```
<select name="siti" multiple="multiple">
  <option value="http://www.html.it">www.html.it</option>
  <option value="http://freephp.html.it">freephp.html.it</option>
</select>
```

Oppure, utilizzando l'attributo **DISABLED** è possibile visualizzare un pulsante ma renderlo inattivo.

```
<input disabled="disabled" type="submit" value="invia">
```

Oppure ancora come sia possibile, utilizzando il tipo **TEXTAREA**, inserire un testo più ampio, suddiviso in varie righe a video. Nell'esempio si possono inserire 5 righe di 40 colonne.

```
<textarea name="testo" rows="5" cols="40">
  Qui puoi scriver il tuo testo
</textarea>
```

Oppure campi testo con inserimento criptato (tipo le password), ecc. ecc.

Questo e molti altri esempi (in italiano) possono essere trovati nel sito di HTML.IT a questa pagina ad esempio

<http://xhtml.html.it/guide/leggi/51/guida-html/>