

HTML e PHP

<form>

Attraverso un form HTML il client è in grado di inviare dei dati al server.

Per definire un form si utilizza il tag:

```
<form method='post' action='elabora.php'>
```

L'attributo **action** indica la pagina che riceve i dati, l'attributo **method** specifica le modalità con cui li riceve, può assumere i seguenti valori:

- **post**: i dati vengono inviati attraverso una richiesta http dal browser al server, in modo che i dati non siano visibili per l'utente.
- **get**: i dati vengono accodati all'indirizzo della pagina richiesta, in chiaro, quindi visibili per l'utente

<input>

All'interno del form gli elementi deputati all'acquisizione dei dati sono definiti dal tag:

```
<input type='text' name='codice'>
```

Nella pagina PHP la variabile che recupera quel valore deve avere lo stesso nome preceduto dal simbolo \$ (\$codice per l'esempio precedente), oppure, in base al metodo utilizzato, a una delle variabili **\$_GET[codice]** o **\$_POST[codice]**.

L'attributo **type** definisce il tipo di elemento presente nel form e può assumere diversi valori, tra cui:

- **text**: casella di testo

```
<input type='text' name='NomeText'>
```

il dato è recuperate nel modulo PHP nella variabile \$NomeText secondo la modalità **\$_POST[NomeText]** o **\$_GET[NomeText]** in base al metodo scelto.

HTML	<pre><form method='post' action='elabora.php'> Cognome:<input type='text' name='cognome'></pre>
PHP	<pre><? \$cognome=\$_POST['cognome']; echo "Cognome=\$cognome"; ?></pre>

- **checkbox**: casella di controllo

```
<input type='checkbox' name='NomeCasella' value='valore'>
```

il dato è recuperate nel modulo PHP nella variabile \$NomeCasella secondo la modalità **\$_POST[NomeCasella]** o **\$_GET[NomeCasella]** in base al metodo scelto. In presenza di più caselle di controllo all'interno del form è consigliato attribuire ad ognuna un nome diverso.

HTML	<pre><form method='post' action='casellaControllo.php'> <input type='checkbox' name='calcio' value='1' checked>Calcio <input type='checkbox' name='nuoto' value='1'>Nuoto</pre>
PHP	<pre><? \$calcio=\$_POST['calcio']; \$nuoto=\$_POST['nuoto']; if(\$calcio==1) echo "Hai scelto Calcio"; if(\$nuoto==1) echo "hai scelto Nuoto"; ?></pre>

- **radio**: pulsante opzione

<input type='radio' name='NomePulsante' value='valore1'>etichetta1

<input type='radio' name='NomePulsante' value='valore2'>etichetta2

il valore del pulsante selezionato è recuperato nel modulo PHP nella variabile \$NomePulsante secondo la modalità \$_POST[NomePulsante] o \$_GET[NomePulsante] in base al metodo scelto.

HTML	<pre><form method='post' action='PulsantiRadio.php?> <input type='radio' name='sport' value='1'>Calcio <input type='radio' name='sport' value='2'>Nuoto</pre>
PHP	<pre><? \$sport=\$_POST['sport']; if(\$sport==1) echo "Hai scelto Calcio"; else echo "Hai scelto Nuoto"; ?></pre>

<select>: casella di riepilogo (menù a tendina)

il valore selezionato è recuperato nel modulo PHP nella variabile \$nomeList secondo la modalità \$_POST[nomeList] o \$_GET[nomeList] in base al metodo scelto.

HTML	<pre><form method='post' action='listBox.php'> <select name='nomeList'> <option value='valore1'>etichetta1 <option value='valore2'>etichetta2 </select></pre>
PHP	<pre><? \$nomeList=\$_POST['nomeList']; if(\$nomeList==valore1) echo "Hai scelto etichetta1"; elseif(\$nomeList==valore2) echo "Hai scelto etichetta2"; elseif(.....) ?></pre>

Variabili di sessione

Le sessioni consentono di mantenere informazioni tra le varie pagine navigate. In pratica viene creato un file di testo, sul server che ospita il sito, in cui sono contenute le informazioni dell'utente a cui la sessione è associata.

Per creare una sessione si utilizza la funzione:

session_start()

Per registrare variabili di sessione bisogna inserirle nell'array associativo globale

\$_SESSION:

\$_SESSION['nomeVar']='valore'

In questo modo l'array **\$_SESSION** conterrà nell'elemento indicizzato da **nomeVar** il valore associato **valore**.

Per cancellare una variabile di sessione useremo:

`unset($nome_var)`

Per chiudere completamente una sessione ed eliminare tutti i dati in essa contenuti:

`session_destroy()`

Tecnica postback

La tecnica postback utilizza una sola pagina, di tipo dinamico, per generare il form per la raccolta dei dati e per la loro elaborazione.

In pratica la pagina invia informazioni a se stessa; mediante il controllo sul contenuto di un campo possiamo verificare se la pagina viene richiesta per la prima volta o se è stata già richiamata in post da se stessa.

Un esempio ci permette di capire concretamente il funzionamento

```
<?PHP
echo "
    <html>
    <head></head>
    <body>";
//verifica se è il primo accesso attraverso la variabile d'ambiente $_POST
//se contiene true significa che il modulo è stato inviato
//pertanto non si tratta del primo accesso alla pagina
if($_POST)
{
    echo "Il tuo nome: ".$_POST['user'];
    echo "<br>";
    echo "La tua casella di posta: ".$_POST['mail'];
    echo "<br>";
}
else
{
    //utilizziamo la variabile d'ambiente $_SERVER[$PHP_SELF] per reindirizzare
    //il form alla stessa pagina (tecnica POST BACK)
    //equivalente a <form action='esempioPostback.php' method='post'>
    echo "
        <form action='".$_SERVER['$PHP_SELF']."' method='post'>
        <table>
            <tr><td>Nome Utente:
            <td><input type='text' name='user'><br>
            <tr><td>Email:
            <td><input type='text' name='mail'><br>
        </table>
        <input type='submit' value='INVIA'>
        </form>";
}
echo "</body>
    </html>";
?>
```

Vediamo un esempio commentato che unisce l'uso della tecnica postback e delle sessioni.

```
<?php
//avvio la sessione
session_start();
echo "
    <html>
    <head>
        <title>Indovina il numero</title>
    </head>
    <body align='center'>";
//se $_POST è vera allora la pagina ha richiamato se stessa in risposta all'invio dei dati dal
//form
if ($_POST){
    echo "Numero da indovinare: " . $_SESSION['num'] . "<br>";
    echo "Numero inserito: " . $_POST['n'] . "<br>";
    if($_POST['n']==$_SESSION['num']){
        echo "<h1>HAI INDOVINATO</h1>";
        //elimino la variabile di sessione per rigenerare, al prossimo giro, un nuovo
        //numero da indovinare
        session_unset($_SESSION['num']);
        echo "<a href='\"".$_SERVER['PHP_SELF']."'>Ricomincia</a>";
    }
    else{
        echo "<h1>MI DISPIACE NON HAI INDOVINATO, SE VUOI
RITENTA</h1>";
        echo "<a href='\"".$_SERVER['PHP_SELF']."'>Ritenta</a>";
    }
}
else{
    //se $_SESSION['num'] è falso (vuoto) allora siamo alla prima esecuzione o al caso
    //in cui l'utente ha indovinato il numero, quindi generiamo un nuovo numero da
    //indovinare e lo salviamo nella sessione
    if(!$_SESSION['num'])
        $_SESSION['num']=rand(1,9);
    echo "
    <h1>PROVA A INDOVINARE!</h1><br>
    <form action='\"".$_SERVER['PHP_SELF']."' method='POST'>
        Inserisci il numero (1 - 9): <input type='text' size='2' name='n'><br><br>
        <input type='submit' value='Tenta la fortuna'><br><br>
        <input type='reset' value='Reset'>
    </form>
    </body>
    </html>";
}
?>
```