



FORM

PROF. SANTANIELLO ASSUNTA

I DATI INVIATI DAI FORM

I moduli di HTML (form) possono inviare i propri campi a uno script php per la successiva elaborazione.

Per accedere al campo ricevuto da un modulo si utilizza l'array associativo `$_POST` oppure `$_GET`, a seconda del metodo di invio usato.

```
//se il form usa il metodo post  
$lettura_post=$_POST['nome campo']  
//se il form usa il metodo GET  
$lettura_get=$_GET['nome campo']
```

GET e POST specificano come le informazioni contenute nel form devono essere passate al server.

GET indica al browser di inviare i valori inseriti nel modulo dall'utente all'indirizzo della richiesta, separandoli dal nome della pagina di destinazione con un punto di domanda. Tali valori vengono perciò inseriti in una stringa di query lunga massimo 256 caratteri, visibile all'utente nella barra di indirizzo del browser e perciò poco sicura, soprattutto in caso di informazioni riservate, come per esempio password.

POST rende invece invisibile all'utente la stringa inviata al server in quanto viene inclusa nel corpo del messaggio.

ESEMPIO CHE ILLUSTRA IL FUNZIONAMENTO DEL METODO POST

SOMMA TRA DUE NUMERI

L'esempio si compone di due pagine. La prima che contiene solo codice HTML necessario per mostrare un modulo contenente due caselle di testo e il pulsante di invio, il cui nome è Somma.html.

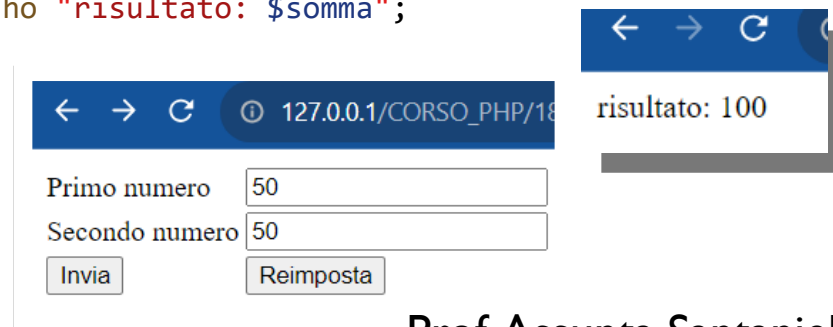
Mediante l'attributo ACTION=Somma.php viene indicato il percorso dello script che riceverà i campi modulo e provvederà alla loro elaborazione per fornire i risultati.

```
<!DOCTYPE html>
<head>
  <title>Pagina HTML contenente il FORM</title>
</head>
<body>
  <form action="18.Somma.php" method="post">
    <table>
      <tr><td> Primo numero
      <td><input type="text" name="numero1">
      <tr><td> Secondo numero
      <td><input type="text" name="numero2">
      <tr><td><input type="submit">
      <td><input type="reset">
    </table>
  </form>
</body>
</html>
```

18.Somma.html

```
<!DOCTYPE html>
<head>
</head>
<body>
  <?php
    $n1=$_POST["numero1"];
    $n2=$_POST["numero2"];
    $somma=$n1+$n2;
    echo "risultato: $somma";
  ?>
</body>
</html>
```

18.Somma.php



The screenshot shows a web browser window with the address bar displaying "127.0.0.1/CORSO_PHP/18". The page content shows the result of the form submission: "risultato: 100". Below this, there is a form with two input fields: "Primo numero" with the value "50" and "Secondo numero" with the value "50". There are two buttons: "Invia" and "Reimposta".

LA TECNICA POSTBACK

Nell'esempio precedente abbiamo visto come avviene il dialogo tra le pagine: la prima statica con il tag `<form>` che viene inviato alla pagina php, la quale elabora i valori ricevuti.

E' possibile leggere i dati provenienti da un form con una connessione di tipo POST e sfruttando la tecnica postback, che consente di inviare a se stessa il form.

La tecnica postback utilizza una sola pagina, di tipo dinamico, per effettuare entrambe le operazioni, cioè sia quella di raccolta dei dati dal form che quella di elaborazione dei risultati.

Mediante il controllo sul contenuto di un campo possiamo verificare se la pagina è stata richiesta per la prima volta o se è stata richiamata in post da se stessa

ESEMPIO POSTBACK

```
<!DOCTYPE html>
<head>
</head>
<body>
  <?php
    //verifica se primo accesso attraverso variabile d'ambiente $_POST
    //se contiene true significa che il modulo è stato inviato
    //pertranto non si tratta del primo accesso alla pagina
    if ($_POST)
    {
      echo "Il tuo nome:".$_POST['username'];
      echo "<br>";
      echo "la tua casella di posta:".$_POST['email'];
      echo "<br>";
    }
    else
    {
      //utilizzo della variabile d'ambiente per reindirizzare
      //il form alla stessa pagina (tecnica postback)
      echo "<FORM ACTION='" . $_SERVER['PHP_SELF'] . "' METHOD = 'POST'>";
      ?>
      <table border="0">
      <tr><td> Nome utente:
      <td><input type="text" name="username"><br>
      <td><td> Email:
      <td><input type="text" name="email"><br>
      </table><input type="submit" value="invio dei dati">
      </FORM>
    <?php
  }
?>
</body>
</html>
```

← → ↻ ⓘ 127.0.0.1/CORSO_PHP/19.postback.php

Nome utente: Email:

Il tuo nome:SuperMario
la tua casella di posta:mario.rossi@gmail.com

Lo script inizia con la verifica della variabile d'ambiente `$_POST`, che contiene true quando la pagina riceve almeno un campo con il metodo POST. Se il Form avesse usato il metodo GET, sarebbe stato sufficiente modificare lo script utilizzando la variabile d'ambiente `$_GET` al posto di `$_POST`.

Nello script possiamo notare come sia stata utilizzata anche la variabile d'ambiente `$_SERVER['PHP_SELF']`, che permette di conoscere l'indirizzo della pagina a cui reindirizzare il Form. Abbinata all'attributo ACTION, consente di reindirizzare la destinazione del Form alla pagina stessa