

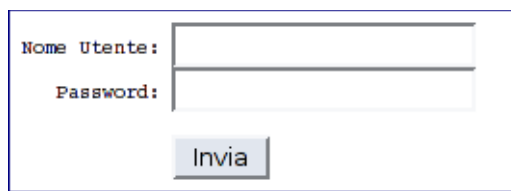
Uso Elementare di Form

Supponiamo di voler inviare delle informazioni, ad esempio **username** e **password**, affinché possano essere utilizzate ed elaborate. Utilizzeremo 2 pagine **form1.html** (contenente il formulario o form) e **risp1.php** con il software per gestire la risposta. Ecco come potrebbe essere strutturata la parte in HTML:

```
<HTML>
<HEAD>
<TITLE>Uso Elementare di Form</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>
<BODY>
<FORM NAME="formpsw" METHOD="post" ACTION="risp1.php">
<PRE>
Nome Utente: <INPUT NAME="username" TYPE="text" ID="username">
Password: <INPUT NAME="password" TYPE="password" ID="password">

                <INPUT NAME="send" TYPE="submit" ID="send" VALUE="Invia">
</PRE>
</FORM>
</BODY>
</HTML>
```

La sezione racchiusa tra i tag `<PRE>...</PRE>`, che servono a mantenere il testo così come digitato per ottenere un minimo di formattazione (PREformatted), visualizzerà il seguente formulario di richiesta per l'utente.

The image shows a rendered HTML form. It consists of two text input fields stacked vertically. The first field is labeled 'Nome Utente:' and the second is labeled 'Password:'. Below these fields is a submit button with the text 'Invia'. The entire form is enclosed in a blue border.

dopo il click sul pulsante **Invia** i dati contenuti nei vari campi **input** potranno essere raccolti dal Server tramite un programmino in **php** ed elaborati. Per semplicità didattica il nostro programma di esempio si limita soltanto a leggere e visualizzare i dati.

```
<?php
$usr = $_POST['username'];
$psw = $_POST['password'];
echo "Ricevute le seguenti informazioni:<br><br>";
echo "Utente: $usr<br>";
echo "Password: $psw<br><br>";
?>
```

Si noti l'uso dell'array predefinito `$_POST` che serve a recuperare il contenuto delle variabili (giunte tramite POST) con l'uso di indici associativi ('username').

Lo schema seguente illustra come, con qualche accorgimento, si possano accorpate i due file precedenti in un unico file **php**.

```
<HTML>
<HEAD>
<TITLE>Uso base di Form</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>
<BODY>
<?php
if(isset($_POST['send']))
{
    $usr = $_POST['username'];
    $psw = $_POST['password'];
    echo "Ricevute le seguenti informazioni:<br><br>";
    echo "Utente: $usr<br>";
    echo "Password: $psw<br><br>";
}
?>
<FORM NAME="form1" METHOD="post" ACTION="<?php echo $_SERVER['PHP_SELF'];?>">
<PRE>
Nome Utente: <INPUT NAME="username" TYPE="text" ID="username">
Password: <INPUT NAME="password" TYPE="password" ID="password">

        <INPUT NAME="send" TYPE="submit" ID="send" VALUE="Invia">
</PRE>
</FORM>
</BODY>
</HTML>
```

Ecco alcuni punti da sottolineare:

- Il file deve essere salvato con il suffisso **.php**
- La riga `if(isset($_POST['send']))` serve ad oltrepassare il contenuto del blocco tra `{ }` la prima volta che viene eseguito, consentendo la visualizzazione del formulario.
- Per ridirigere i contenuti del formulario allo stesso file si è fatto uso di un particolare accorgimento: `ACTION="<?php echo $_SERVER['PHP_SELF'];?>"`
- Così come scritto il programma è divenuto ciclico, dopo l'output dei dati ripresenta la maschera di input.

Quest'ultimo esempio, sicuramente migliorabile, dimostra come sia possibile una completa mescolanza di codice HTML e PHP. Si veda, ad esempio, come l'argomento stringa di ACTION venga generato da codice PHP.