

# PHP Lessons

By Lord\_Dex

## Capitolo 4

### Form HTML

In questa parte della guida parleremo dell'input utente ovvero i form html.

Un form html non e' altro che delle caselle di testo, dei pulsanti, delle aree di testo che comunicano nel nostro caso con uno script php per elaborare i dati.

Iniziamo prima con un semplicissimo esempio di form:

```
<form action="form.php" method="post">
  <input type="text" name="nome"><br>
  <input type="password" name="password"><br>
  <input type="submit" value="Entra">
</form>
```

Analizziamo questo form, non e' altro che html, per ora niente php, nel tag form abbiamo inserito le informazioni piu' importanti per la buona riuscita dello script; infatti in action va' messo il nome dello script a cui dovranno essere inviati i dati, e method deve essere impostato su get o su post.

La differenza tra i due e' notevole, infatti con get il contenuto delle caselle di testo verra' visualizzato nella barra dell'indirizzo nel browser, invece con post, verra' inviato alla pagina ma non verra' visualizzato nella barra del browser.

### Script per l'elaborazione

Salviamo la pagina con il nome di form.html e passiamo ad analizzare il codice dello script che elaborera' i dati ricevuti dal form html:

```
<?php
  $nome = $_POST['nome'];
  $password = $_POST['password'];
  echo "UserName: $nome <br>Password: $password";
?>
```

Prima cosa fate ATTENZIONE a salvare lo script con lo stesso nome della pagina inserita nel tag form nella proprieta' action.

Allora lo script e' molto semplice salviamo prima di tutto il contenuto delle input box in due variabili e poi le

visualizziamo.

Come avrete sicuramente capito per ottenere il contenuto di una input box si usano quelle variabili `$_POST` (o `$_GET` se nel form `method="get"`) più le parentesi quadre, un singolo apice e il nome dell'input inserito nella proprietà `name` del form.

### Un unico script

Ora andiamo a modificare un po' il tutto per renderlo più funzionale, infatti benché la realizzazione delle due pagine è molto semplice, il fatto di avere due file separati è un po' difficile da gestire specialmente nel caso di molti form sparsi per il sito, vediamo come fare per unificare il tutto in un'unica pagina.

Esiste infatti una funzione php che controlla l'esistenza di una determinata variabile e ritorna `true` se quest'ultima esiste.

```
<?php
    if (isset($_POST['nome'])) {
        $nome = $_POST['nome'];
        $password = $_POST['password'];
        echo "UserName: $nome <br>Password: $password";
    } else {
        ?>
        <form action="form.php" method="post">
            <input type="text" name="nome"><br>
            <input type="password" name="password"><br>
            <input type="submit" value="Entra">
        </form>
        <?php
    }
?>
```

Come possiamo vedere lo script controlla se la variabile `$_POST['nome']` (che dovreste sapere a cosa è associata) è stata creata e visualizza l'username e la password, se invece la variabile non è ancora stata creata, visualizza il form.

Non c'è niente di particolare a parte il fatto che nella visualizzazione del form chiudiamo il tag dello script php per poi riaprirlo a form concluso.

Come potete vedere la gestione di form non è affatto difficile, come al solito bisogna solamente stare attenti.

**Next Part:** Operazioni con i file

**Lord\_Dex**

[kapo.web@gmail.com](mailto:kapo.web@gmail.com)