

Input dei dati - I FORM

In questo capitolo tratteremo il principale metodo che ci permette un input di dati verso la nostra applicazione, ovvero i form. Vi capiterà certamente spesso di incontrare dei form; alla fine di questa lezione (come pure di tutte le altre), vi è la possibilità di inserire un commento o una domanda. Questa operazione avviene appunto tramite un form. Chiunque capisca qualcosa di html, sa benissimo come realizzarne uno. Un altro conto è come intercettare questi dati e come trattarli.

Iniziamo con creare un semplicissimo form che richiede il nome e un messaggio.

```
<form action="elaborazione.php" method="POST">
nome:<br>
<input type="text" name="nome" />
<br><br>
messaggio:<br>
<textarea name="messaggio"></textarea>
<br><br>
<input type="submit" value="invia" />
</form>
```

Salvatelo come form.php (o come form.html, in quanto contiene solo html).

Esaminiamo il nostro form:

- Con "action" indichiamo il nome della pagina che verrà caricata una volta premuto sul tasto invia.
- Method ci indica che metodo vogliamo utilizzare per passare i valori alla pagina definita in action.
- Esistono due metodi:
 - GET: Questo metodo permette di far passare i valori accordandoli all'url. Quando sulla barra dell'url vedete qualcosa tipo `www.nomesito.com/pagina.php?valore=1&opzione=cancella`, significa che si sta facendo passare a pagina.php tramite il metodo GET la variabile `valore` settata a 1 e la variabile `opzione` settata su `cancella`. Utilizzeremo il metodo GET quando per forza di cose dovremo far passare dei valori tramite un link. Nei form è preferibile utilizzare il metodo
 - POST che passa i valori in background e soprattutto non è limitato come GET a 256 caratteri.
- Infine con `name` definiamo il nome della variabile che passiamo.

Ora vediamo come recuperare i nostri valori. Creiamo la pagina `elaborazione.php` in questo modo:

```
<?php
```

```
$nome = $_POST['nome'];  
$messaggio = $_POST['messaggio'];  
  
echo "$nome ha scritto:<br><br>";  
echo $messaggio;
```

```
?>
```

Come vedete `$_POST` è un array associativo, dunque a `$_POST['nome']` corrisponderà quello che abbiamo inserito nel campo nome. Avremmo potuto utilizzare direttamente `$_POST`, ma per comodità spesso si assegna una variabile “normale” come ho fatto io.

Ovviamente quello che abbiamo fatto ha poco senso; ma vedremo nelle prossime lezioni come questi dati inseriti dall’utente possano essere salvati ad esempio in un file di testo o in una tabella compatibile con excel o finalmente in un database