



I moduli HTML

Interazione per l'invio di informazioni in Internet

Sviluppo di siti web – UD09

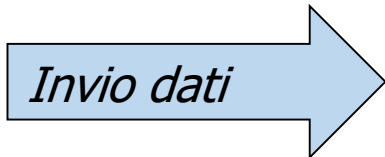
prof. Giovanni Borga

Interazione e uso dei moduli

Uno dei fattori che ha decretato il successo del Web è senz'altro la possibilità di interagire in modo bi-direzionale, ovvero di inviare informazioni e non solo di scaricarle.

Per realizzare questo tipo di servizi sono necessari strumenti co cui l'utente può inserire i dati da inviare ad un sistema di gestione; per farlo si utilizzano i moduli, detti anche **FORM**.

The image shows a web form titled "Seminar Form" with a "More Actions" dropdown menu. The form contains several input fields: "Name", "Date of Seminar" (with a calendar icon and "[dd-MMM-yyyy]" format), "Position", "Arrival Date" (with a calendar icon and "[dd-MMM-yyyy]" format), "Institution", "Departure Date" (with a calendar icon and "[dd-MMM-yyyy]" format), "Email", "Place of Stay", and "Phone". At the bottom left, there are radio buttons for "Format of Presentation" with options "Lecture", "Seminar", and "Colloquium". At the bottom right, the word "NAME" is displayed in large, bold, black letters. At the very bottom, there are "Submit" and "Reset" buttons.



ACTION

Pagina di programmazione lato server (cgi, asp, php, js...etc..)

Tag FORM e attributi NAME e ACTION

Il tag da utilizzare per creare un modulo è **<form>**.

Il tag form è un elemento di blocco perciò lascia uno spazio prima dell'apertura e dopo la chiusura.

I due principali attributi del tag <form> sono NAME, che contraddistingue il «pacchetto dei dati inviato» e ACTION, che altro non è se non la URL di una pagina sul server che si occupa di elaborare i dati inviati.

Ecco la sintassi di utilizzo di FORM:

<form name="datiUtenti" action="paginaRisposta.php">... campi da compilare ...</form>



NAME

Indica il nome del form



ACTION

URL della pagina

I due metodi di invio dei dati e l'attributo METHOD

Una volta creato il *form* dobbiamo scegliere tra i due metodi di invio: GET e POST utilizzando l'apposito attributo **METHOD**.

Metodo GET: la pagina di risposta viene contattata e contemporaneamente le vengono inviati i dati tramite la «query string» che è una parte della URL.

<http://www.sito.com/paginaRisposta.php?nome=Wolfgang&cognome=Cecchin&datiInviati=prova+invio>

Metodo POST: l'invio dei dati avviene in uno step successivo a quello della chiamata della pagina. Le due fasi sono:

1. viene contattata la pagina sul server che deve processare i dati,
2. vengono inviati i dati stessi

Differenze rispetto a GET: a) i dati inviati non compaiono nella URL; è quindi preferibile se non si desidera che l'utente li veda; b) la quantità di dati inviabili è superiore ed è possibile inviare immagini o altri files.

In questo esempio si utilizza il metodo GET:

```
<form name="datiUtenti" action="paginaRisposta.php" method="GET">....</form>
```


I campi dei moduli

Esistono dei **tag che identificano i «campi»** utili alla compilazione del modulo. Essi variano in funzione del tipo di ausilio che si vuole dare all'utente (ad esempio, le caselle combinate permettono la scelta tra una lista di voci precompilata).

La gran parte dei tipi di campo si realizza con l'impiego del **<input>**.

Si tratta di un tag vuoto, dotato di attributo **TYPE** che consente di specificare le varianti (es. testo, pulsante, lista ecc). Vediamo un primo esempio, la casella di testo semplice:

```
<input type="text" name="nome" value="Inserisci il tuo nome">
```

In questo esempio vediamo due degli attributi più diffusi, **NAME** e **VALUE**.

L'attributo VALUE è valido per la maggior parte degli elementi di un form: definisce il contenuto del campo.

Anche l'attributo NAME è valido per la maggior parte degli elementi: esso definisce l'etichetta associata al campo, appunto, il suo nome.

I pulsanti

Altri esempi: 3 tipi di pulsante:

Il tipo «button» è un pulsante generico che si utilizza per lo più con javascript.

<input type="button" value="Pulsante inattivo">

Pulsante inattivo

Il pulsante «submit» invece ha lo stesso aspetto ma è il pulsante più importante del form; salvo poche eccezioni, deve essere sempre presente in quanto è il controllo con cui l'utente effettua l'invio dei dati alla pagina indicata in ACTION:

<input type="submit" value="Invia i dati">

Invia i dati

Un terzo tipo di pulsante è il «reset» che riporta tutti gli elementi del form allo stato iniziale:

<input type="reset" value="Ricomincia...">

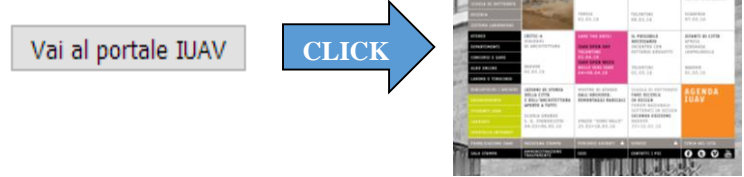
Ricomincia...

L'attributo **VALUE** in questo caso è il **testo** che compare sul pulsante stesso. Non è un dato che si invia.

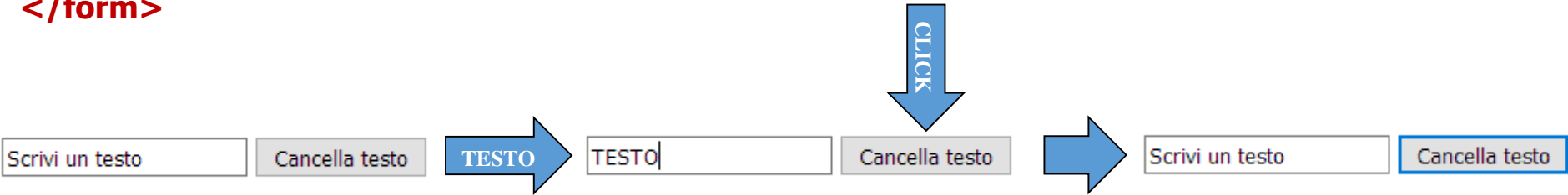
I pulsanti

Il comportamento di «submit» e «reset»:

```
<form action="http://www.iuav.it">  
  <input type="submit" value="Vai al portale IUAV">  
</form>
```



```
<form action="...">  
  <input type="text" value="Scrivi un testo">  
  <input type="reset" value="Cancella testo">  
</form>
```



Il pulsante grafico

Il tipo «image» ci consente di utilizzare come bottoni del form delle vere e proprie immagini e assegnare loro un valore grazie a JavaScript; in questo caso non si tratta propriamente di un bottone ma la funzionalità è la medesima.

```
<input name="invia" type="image" src="invia.gif" alt="invia dati" title="invia dati" width="78" height="38">
```



L'immagine «invia.gif» assume il comportamento di un «submit».
Gli **attributi** del campo immagine sono molto simili a quelli del tag .

Il pulsante grafico ha il grande vantaggio di rendere **illimitata la gamma di possibilità** nella realizzazione di pulsanti di invio ma ha uno svantaggio in termini di usabilità: a schermo **non dà l'effetto di «pressione»** di un tasto, tipico dei pulsanti classici.

Varianti del campo di testo

Per i campi di testo esistono due principali varianti : **Text** e **Textarea**.

Per l'INPUT TEXT, visto all'inizio, sono previsti i seguenti attributi:

- **MAXLENGTH** – indica il numero massimo di caratteri che l'utente può inserire,
- **SIZE** – indica le dimensioni del campo di testo (in numero di caratteri – NON in pixel).

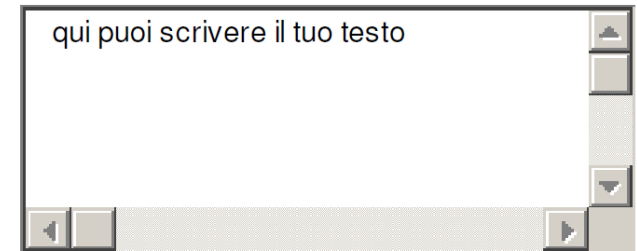
```
<input name="mioTesto" type="text" value="prova" size="40" maxlength="200" />
```

La **TEXTAREA** è invece un campo adatto all'inserimento di testi di una certa lunghezza e con più di una riga. Non è un tipo di <input> bensì un tag preciso: **<textarea>**

```
<textarea name="testo" rows="5" cols="40">  
  qui puoi scrivere il tuo testo  
</textarea>
```

Con gli attributi ROWS e COLS si attribuiscono le **dimensioni** in righe e colonne in unità carattere (non in pixel)

TEXTAREA non è un tag vuoto e non è prevista una lunghezza massima.

A screenshot of a web browser showing a text area input field. The text area contains the text "qui puoi scrivere il tuo testo". The text area has a light gray border and a vertical scrollbar on the right side. The text is centered within the field.

Ancora un campo di testo: «password»

L'INPUT type **PASSWORD** è identico al type TEXT con la sola eccezione di visualizzare asterischi o pallini (dipende dal browser) al posto dei caratteri:

```
<input type="password" maxlength="16" size="40" name="password" />
```

Scelta di opzioni

Per la scelta di opzioni abbiamo tre tipi di strumenti:

1. La scelta ad opzioni multiple (CHECKBOX)
2. La scelta ad opzione esclusiva (RADIO BUTTON)
3. La lista di opzioni - multipla o esclusiva – (CASELLA COMBINATA)

I primi due offrono visibilità completa a tutte le opzioni quindi sono più adatte a rose di pochi elementi mentre la lista di opzioni è «scrollabile» e si chiude automaticamente quindi consente di gestire numerose opzioni in poco spazio.

Vediamo la sintassi per i **checkbox**, adatti per **scelte multiple**, e relativo rendering:

```
<input type="checkbox" name="html" value="html"/> html
```

```
<br />
```

```
<input type="checkbox" name="css" value="css"/> css
```

```
<br />
```

```
<input type="checkbox" name="javascript" value="javascript"/> JavaScript
```

- html
- css
- JavaScript

NB: si noti come gli elementi definiti da questo codice HTML siano effettivamente 3 campi diversi.

Scelta di opzioni

Nell'esempio con i **radio button** optiamo per un layout orizzontale, quindi senza `
`:

HTML `<input type="radio" name="linguaggio" value="html" />`

CSS `<input type="radio" checked="checked" name="linguaggio" value="css" />`

JavaScript `<input type="radio" name="linguaggio" value="javascript" />`

HTML CSS JavaScript

Essendo il radio button lo strumento per le **scelte ad opzione esclusiva** (una sola scelta valida), il gruppo di elementi è **considerato un unico campo**.

Per questo motivo il NAME con lo stesso valore è ciò che raggruppa le diverse opzioni della scelta. L'attributo VALUE, come per i checkbox, è il contenuto del campo che viene inviato alla pagina di destinazione.

Vediamo inoltre anche l'attributo **CHECKED** che deve essere valorizzato con il testo «checked» che consente di pre-impostare una scelta iniziale. CHECKED è valido anche per i checkbox.

Scelta di opzioni

Per creare un **menu di opzioni** si utilizza un tag apposito: `<select>`. Con questo tag, ciascuna voce deve essere compresa all'interno del tag `<option>` (con chiusura opzionale).

Il valore viene sempre specificato con l'attributo VALUE. Analogo all'attributo CHECKED visto precedentemente, con l'attributo **SELECTED** si può pre-impostare una scelta iniziale. Inoltre con l'attributo **MULTIPLE** è possibile attivare la scelta multipla:

```
<select name="siti" multiple="multiple">  
  <option value="iuav" selected="selected">www.iuav.it</option>  
  <option value="google">www.google.it</option>  
  <option value="facebook">www.facebook.com</option>  
</select>
```



www.iuav.it	▼
www.iuav.it	
www.google.it	
www.facebook.com	

Esiste infine l'attributo **SIZE** con cui è possibile decidere di tenere la casella aperta con un numero fisso di righe visibili (es. `<select name="siti" size="3">` - casella aperta con tre righe visibili e scrollbar per lo scorrimento).

Campi speciali

Abbiamo infine due campi «speciali» che si utilizzano con i moduli più complessi: **HIDDEN** e **FILE**.

Sono entrambi tipi del tag <input>. Il primo campo ha la seguente sintassi:

```
<input type="hidden" name="sessionID" value="42533" />
```

E ha lo scopo di far transitare dei dati dal modulo alla pagina di destinazione senza che questi vengano visti dall'utente, o quando non c'è necessità che l'utente li modifichi. E' dunque un elemento che non viene renderizzato nella pagina.

Il tipo FILE è invece molto utilizzato perché consente l'upload di files dal computer dell'utente al server remoto. Ecco la sintassi:

```
<input type="file" name="fileUtente" size="20" />
```

L'attributo **SIZE** serve a specificare la larghezza dell'elemento e si esprime in caratteri (non in pixel).

Il rendering del tipo FILE è molto diverso da browser a browser ma in generale è composto da un pulsante con l'indicazione «browse file», «scegli file» o similare e uno spazio dove viene visualizzato il nome del file dopo averlo selezionato.