

# HTML 4.01

**Settima lezione**

---

**7 Aprile 2004**

**di Ivano Stranieri**

# Lezione precedente

- Cosa sono i FRAME
- A cosa servono i FRAME
- Sintassi
- Attributi
- Collegamenti e **NOFRAME**
- **IFRAME**
- Vantaggi e svantaggi dei FRAME

# Cosa vedremo oggi

- Cosa sono i FORM
- I moduli per interagire con l'utente
- Tag **INPUT**
- Bottoni
- Campi testuali
- Le scelte
- Altri tipi di campi particolari

# Cosa sono i FORM

I FORM sono delle strutture che permettono all'utente di interagire con il sito web (ad esempio iscrivendosi ad una mailing list) inviando alcuni dati tramite internet. Per interagire con l'utente è necessario raccogliergli i dati e poi utilizzarli in diverse maniere.



# Struttura dei FORM

L'invio dei dati è normalmente diviso in due fasi:

1. Una prima pagina in cui l'utente effettua alcune scelte, inserisce dei dati, scrive dei testi...
2. Una seconda pagina a cui vengono inviati questi dati e che si occupa di "lavorarli" (immagazzinarli in un database, processarli, controllarli, ecc...). Questa pagina non è fatta in HTML ma in linguaggi di tipo server-side (PHP, CGI, ASP, JSP...)

Le varie informazioni che vengono immesse dall'utente sono inserite all'interno di particolari aree che si chiamano CAMPI

# Come funziona un FORM



# Il tag FORM

Per creare un FORM bisogna inserire nella pagina i tag `<FORM> ... </FORM>`

Tutti gli elementi che inseriremo dentro al **FORM** dovranno essere inseriti dentro questi due tag (apertura e chiusura).

Attenzione perché il tag `<FORM>` è un elemento di blocco e lascia, quindi, una riga vuota prima della sua apertura e dopo la sua chiusura.

# Attributi di FORM

**NAME** = serve a dare un nome al form;  
utilizzando javascript risulterà fondamentale.

**ACTION** = serve a specificare cosa dovrà  
accadere quando si invia il form compilato  
(contiene quindi l'url della pagina di risposta).

Esempio:

```
<FORM NAME="pippo" ACTION="pippo.html">
```

# Attributi: METHOD

Quando si inviano i FORM, i dati vengono spediti alla seconda pagina (quella di risposta), secondi due diversi metodi:

- 1) GET: vengono inviati tutti i dati e questi compaiono nella barra del titolo...
- 2) POST: viene prima contattata la pagina di risposta e poi vengono inviati i dati => i dati non compaiono nella barra del titolo => metodo più sicuro.

Es: `<FORM METHOD="POST"> ... </FORM>`

# Tipo di codifica: ENCTYPE

Normalmente si spediscono solamente testi tramite i FORM. In questo caso la codifica utilizzata è quella testuale. Se invece spediamo anche le immagini o dei file non di testo tramite il FORM, bisogna impostare la codifica nel modo corretto.

Le due possibili impostazioni sono:

- 1) **ENCTYPE="text/plain"**
- 2) **ENCTYPE="multipart/form-data"**

Es: **<FORM ENCTYPE="text/plain"> ... </FORM>**

# TARGET

Come già visto nel caso dei LINK e nel caso dei FRAME, **TARGET** permette di specificare al form che la seconda pagina si dovrà aprire in un altro browser differente da quello ove risiede il form stesso.

Esempio:

```
<FORM ACTION="pippo.html" TARGET="_blank">
```

# Fare ordine... FIELDSET

Tramite questo nuovo tag è possibile ordinare i vari campi che compongono il form. Si possono quindi creare delle aree a cui si dare un nome (etichetta) tramite il tag **<LEGEND>**:

```
<FORM ACTION="pippo.html" METHOD="POST">  
  <FIELDSET>  
    <LEGEND>Dati personali</LEGEND>  
  </FIELDSET>  
</FORM>
```

# INPUT

Il tag **INPUT** è tag più utilizzato per creare i campi all'interno dei FORM.

Tramite diversi attributi si possono creare molti tipi di campi (testuali, bottoni, tasti di scelta esclusiva) ognuno con diversi valori e con associato sempre un nome (fondamentale per riconoscere il campo nella pagina successiva).

I diversi tipi di **INPUT** si differenziano per il valore assegnato all'attributo **TYPE**

# Campi testuali

Per creare un campo di semplice testo è sufficiente inserire il valore “**TEXT**” all’interno dell’attributo **TYPE** dell’**INPUT**.

Si può associare un valore al campo tramite l’attributo **VALUE** e tale valore verrà visualizzato al suo interno:

```
<INPUT TYPE=“TEXT” NAME=“testo” VALUE=“pippo”>
```

# Campi testuali

E' possibile stabilire la lunghezza del campo testuale con l'attributo **SIZE**, nonché delimitare il numero di caratteri che l'utente può inserire al suo interno tramite l'attributo **MAXLENGTH** (questo attributo risulta molto utile per evitare che si commettano errori nell'inserimento ad es. di codici fiscali, CAP altri dati di larghezza conosciuta):

Esempio:

```
<INPUT TYPE="TEXT" NAME="testo"  
VALUE="pippo" SIZE="10" MAXLENGTH="5">
```

# Bottoni generici

Può servire alle volte di utilizzare il tag **INPUT** per creare dei pulsanti di tipo generico. In questo caso il **TYPE** che si specifica è "**BUTTON**" e il valore che gli viene associato comparirà proprio come testo del bottone:

```
<INPUT TYPE="BUTTON" NAME="bottone"  
      VALUE="premi qui">
```

# PASSWORD

Quando all'interno del campo **INPUT** bisogna inserire alcuni dati "riservati" quali ad esempio una password al momento della registrazione di un utente, si inserisce il valore **PASSWORD** all'interno del **TYPE** dell'**INPUT**. In questo modo i dati inseriti all'interno del campo saranno asteriscati e non visibili all'utente (notate che non è neanche possibile copiarli e incollarli su altri testi):

Esempio:

```
<INPUT TYPE="PASSWORD" SIZE="10">
```

# Campi di sola lettura

Se si vuole impedire che l'utente modifichi il contenuto del campo è possibile inserire al suo interno l'attributo **READONLY**:

Esempio:

```
<INPUT TYPE="TEXT" VALUE="pippo e topolino"  
      SIZE="40" READONLY>
```

# Disabilitare i campi

Così come è possibile rendere di sola lettura i campi è possibile anche disabilitarli tramite l'attributo **DISABLED**.

Questo attributo è molto utilizzato insieme al linguaggio Javascript per impedire che un form venga inviato più di una volta dall'utente per errore (dopo l'invio del form il tasto di invio viene disabilitato).

```
<INPUT TYPE="TEXT" VALUE="pippo" DISABLED>
```

# Scelta multipla

Un altro valore di **TYPE** per il tag **INPUT** è **CHECKBOX** che permette di creare dei quadratini che l'utente può o meno spuntare con una croce effettuando quindi delle scelte multiple fra diverse alternative.

Attenzione: in questo caso per ogni **INPUT** i **NAME** sono gli stessi e cambiano solamente i **VALUE**.

```
<INPUT TYPE="CHECKBOX" NAME="citta" VALUE="roma">  
<INPUT TYPE="CHECKBOX" NAME="citta" VALUE="milano">  
<INPUT TYPE="CHECKBOX" NAME="citta" VALUE="parma">
```

# Scelta multipla

Per default un checkbox non è mai spuntato appena viene caricata la pagina: si può però decidere di impostarlo già selezionato attraverso l'attributo **CHECKED**.

```
<INPUT TYPE="CHECKBOX" NAME="citta" VALUE="roma">  
<INPUT TYPE="CHECKBOX" NAME="citta" VALUE="milano">  
<INPUT TYPE="CHECKBOX" NAME="citta" VALUE="parma"  
CHECKED>
```

# Scelta esclusiva

Se vogliamo che su diverse alternative l'utente ne scelga solamente una, dobbiamo impostare il **TYPE** dell'**INPUT** sul valore **RADIO**, in modo da creare non più dei quadratini ma dei pallini. Una scelta fatta dall'utente escluderà l'altra. Anche in questo caso per far sì che una scelta compaia come già selezionata si utilizza l'attributo **CHECKED**:

```
<INPUT TYPE="RADIO" NAME="citta" VALUE="roma" CHECKED>  
<INPUT TYPE="RADIO" NAME="citta" VALUE="milano">
```

# I campi nascosti

Se vogliamo inserire un campo nascosto (utile per spedire delle informazioni da una pagina all'altra senza interagire con l'utente) utilizziamo il campo **INPUT** con il valore del **TYPE** impostato a **HIDDEN**:

```
<INPUT TYPE="HIDDEN" NAME="nascosto"  
      VALUE="pippo">
```

# Il tag **BUTTON**

Esaurite le differenze fra i vari tipi di "INPUT TYPE", scopriamo che esiste anche il tag `<BUTTON>`.

Tale tag se utilizzato in maniera banale è esattamente identico al bottone generico visto in precedenza.

La sua particolarità consiste invece nella capacità di comprendere al suo interno tutto il codice HTML che si vuole inserire (e di assumere, quindi, un particolare aspetto tutt'altro che banale).

# Il tag **BUTTON**

Esempio:

```
<FORM METHOD="POST" ACTION="pippo.html">  
  <BUTTON NAME="pippo">  
    <B><FONT SIZE="4">Pippo</FONT></B>  
  </BUTTON>  
</FORM>
```

# TEXTAREA

Se si vuole dare la possibilità di inserire dentro un form dei testi molto lunghi, si può utilizzare il campo **TEXTAREA** che visualizza una finestrella indipendente in cui immettere il testo.

Gli attributi di questo tag (che si apre e si chiude), oltre al **NAME**, sono **COLS** e **ROWS** che specificano quanto debba essere largo e alto il campo in termini di colonne e di righe che stiamo inserendo.

Il testo che inseriamo tra il tag di apertura e il tag di chiusura lo ritroveremo all'interno del campo.

# TEXTAREA

Esempio:

```
<FORM METHOD="POST" ACTION="pippo.html">  
  <TEXTAREA COLS="30" ROWS="15" NAME="testo">  
    Questo testo lo troveremo nel campo  
  </TEXTAREA>  
</FORM>
```

# MENU' DI SCELTE

Grazie al tag **<SELECT>** si possono costruire dei menù di scelta. Ogni voce del menù sarà compresa all'interno del **SELECT** dentro il tag **<OPTION>** (la cui chiusura è opzionale) con il proprio **VALUE**.

La scelta preimpostata si può definire inserendo dentro il tag **OPTION** l'attributo **SELECTED** (come abbiamo visto con **CHECKED** per i tag già visti).

# MENU' DI SCELTE

Esempio:

```
<FORM METHOD="POST" ACTION="pippo.html">  
  <SELECT NAME="scelte">  
    <OPTION VALUE="uno">scelta uno</OPTION>  
    <OPTION VALUE="due">scelta due</OPTION>  
    <OPTION VALUE="tre">scelta tre</OPTION>  
  </SELECT>  
</FORM>
```

# SPEDIRE UN FORM

Per spedire il form è necessario che l'utente clicchi sul relativo tasto di invio.

Tale tasto si crea tramite il tag **INPUT** con l'attributo **TYPE** impostato su **"SUBMIT"**.

All'interno del **VALUE** si può inserire un testo che comparirà sul bottone di invio stesso.

```
<INPUT TYPE="SUBMIT" VALUE="Invia">
```

# Ripristinare il form

Utilizzando il tag **INPUT** con il **TYPE** impostato sul valore "**RESET**", si visualizzerà un bottone che, se premuto, riporterà il FORM ai valori iniziali dei suoi campi cancellando tutto ciò che è stato inserito dall'utente, senza dover ricaricare la pagina:

```
<INPUT TYPE="RESET" VALUE="Ricompila">
```

# Spedire con una immagine

Utilizzando il tag **INPUT** con il **TYPE** impostato sul valore "**IMAGE**" è possibile utilizzare una immagine come se questa fosse il tasto **SUBMIT**. Gli attributi sono quelli relativi al normale tag **IMG** con cui si inseriscono le immagini:

```
<INPUT TYPE="IMAGE" SRC="pippo.jpg"  
      WIDTH="170" HEIGHT="220">
```

# Spedire i dati via e-mail

Vediamo un esempio concreto e funzionante (in quanto non ha bisogno di una pagina di risposta creata con un altro linguaggio) per spedire i dati da una pagina con un FORM.

Si possono spedire tali dati ad un indirizzo e-mail specificando l'indirizzo dentro il tag **<FORM>**.

I dati partiranno dal client di posta installato sul PC dell'utente e si posizioneranno all'interno di una e-mail con l'elenco dei campi e i valori inseriti dall'utente stesso.

# Spedire i dati via e-mail

```
<FORM METHOD="POST" ENCTYPE="text/plain"  
  ACTION="mailto:pippo@pippo.it">
```

```
[...]
```

```
</FORM>
```