

Corso di HTML

Creare moduli html <FORM>

I moduli sono elementi importanti per ogni sito web che vuole raccogliere informazioni dai suoi utenti e quindi è l'unico sistema nel linguaggio Html che permette di interagire con l'utente (come abbiamo visto le pagine in HTML sono statiche e non permettono al visitatore di interagire con il sito) mentre con i moduli tutto questo è possibile, l'utente può mandare una propria opinione, chiedere informazioni, rispondere a sondaggi ecc. senza la necessità di scrivere via e-mail.

Da sottolineare che tutto ciò è possibile grazie a dei programmi che risiedono sul server che si chiamano CGI (Common Gateway Interface) quindi perchè un form Html funzioni è indispensabile un programma CGI residente sul proprio server.

Ma è possibile creare dei form anche senza CGI che tratteremo più avanti.

I tag <FORM></FORM> aprono e chiudono il modulo.

Adesso analizziamo questa sintassi:

```
<FORM method="get|post"  
action="http://www.tuosito.com/cgi-  
bin/nome_script.cgi">
```

Se method lo impostiamo come get cioè **method="get"** i dati vengono spediti al server e separati in due variabili, però con questo metodo il numero massimo di caratteri contenuti nel form è di 255.

Utilizzando **method="post"** i dati vengono ricevuti direttamente dallo script CGI senza un preventivo processo di decodifica. Questa caratteristica fa sì che lo script possa leggere una quantità illimitata di caratteri.

Iniziamo a creare dei campi per il nostro modulo cominciando da:

TEXT

Il valore **TEXT** crea i campi di testo, dove vengono richiesti dati quali il nome o l'indirizzo e-mail. È un valore usato soprattutto

per informazioni non predefinite che variano di volta in volta e con i seguenti attributi si crea un campo testo:

maxlength: definisce il numero massimo di caratteri inseribili nel campo, oltre il quale non e' possibile aggiungere

size: definisce la larghezza della stringa all'interno della pagina

value: visualizza un testo di default all'interno della stringa.

Esempio:

```
<INPUT type="TEXT" name="nome" maxlength="30" size="20" value="Inserisci nome">
```

ESEMPIO

PASSWORD

Questo campo ha una funziona simile a **TEXT** visto in precedenza, ma con la differenza che quando si digita all'interno della stringa bianca, non appaiono le lettere ma i classici asterischi delle password (anche se in realtà i dati non vengono codificati e quindi non c'è nessuna sicurezza).

```
<INPUT type="PASSWORD" name="nome" maxlength="30" size="20">
```

ESEMPIO

CHECKBOX

Questo attributo viene utilizzato per informazioni del tipo "si/no" oppure "vero/falso". Crea delle piccole caselle quadrate da spuntare o da lasciare in bianco. Se la casella e' spuntata input restituisce un valore al CGI, al contrario non restituisce alcun valore.

Value impostato su "**yes**" significa che di default la casella e' spuntata. **Checked** controlla lo stato iniziale della casella, all'atto del caricamento della pagina.

```
<INPUT type="CHECKBOX" name="età" value="yes" checked>
```

ESEMPIO

RADIO

Questo attributo ha funzioni simili a **CHECKBOX**, ma presenta più scelte possibili. Selezionando una voce tra quelle presenti, qualora abbiano tutte valore "name" identico, si deseleggono automaticamente le altre.

```
<INPUT type="RADIO" name="voto"
value="sufficiente">
<INPUT type="RADIO" name="voto" value="buono">
<INPUT type="RADIO" name="voto" value="ottimo">
```

SUFFICIENTE

BUONO

OTTIMO

SUBMIT

Questo è il classico bottone che invia il form con tutti i suoi contenuti. La larghezza del bottone dipende dalla lunghezza del testo.

```
<INPUT type="SUBMIT" value="Invia">
```

ESEMPIO

RESET

Bottone che cancella l'intero form eliminando i dati inseriti.

```
<INPUT type="RESET" value="Cancella">
```

ESEMPIO

TEXTAREA

Textarea viene utilizzato quando serve spazio per inserire molto testo. La larghezza è impostata da "**cols**" e l'altezza da "**rows**". **WRAP="physical"** stabilisce che qualora il testo inserito superi la larghezza della finestra, venga automaticamente riportato a capo.

```
<TEXTAREA cols=50 rows=4 WRAP="physical"
name="commento"></textarea>
```

ESEMPIO**SELECT**

Select permette la creazione di elenchi a discesa con varie possibilita' di scelta.

```
<SELECT size=1 cols=4 NAME="giudizio">
<OPTION selected Value=nessuna>
<OPTION value=buono> Buono
<OPTION value=sufficiente> Sufficiente
<OPTION Value=ottimo> Ottimo
</select>
```

ESEMPIO**Form senza programmi CGI**

All'inizio di questa lezione si è sottolineato come i form HTML diano risultati migliori quando vengono affiancati da programmi CGI residente sul server ma è possibile anche in assenza di questi programmi inserendo il seguente codice:

```
<form action="mailto:nome@provider.it"
method="post">
```

Con questo sistema cambiando "**nome@provider.it**" con l'indirizzo e-mail verso il quale devono essere indirizzati i moduli e possibile ricevere le informazioni tramite posta elettronica.

Di seguito un esempio di dati inviati con un form senza passare per un programma CGI

Consideriamo che nel modulo ci siano i seguenti campi Nome, Cognome e E-mail ecco il risultato:

Nome=Tizio&Cognome=Caio&E-mail=tizio@provider.it

Mentre con un programma CGI si avrebbe avuto questo risultato:

(Nome) **Tizio**
(Cognome) **Caio**
(Email) **tizio@provider.it**

Capito la differenza?

Non mi resta che augurarvi un buon lavoro, e dopo che avete imparato per bene l'html potete proseguire a studiare gli altri linguaggi tipo javascript, asp o php, obbligatori per rendere le pagine web dinamiche.

Fine del manuale
[Torna al menù principale](#)