

## **I MODULI ED IL SERVER WEB**

**I moduli permettono di effettuare un'interazione tra client e server.**

**Sono costituiti da una serie di campi che, una volta compilati, saranno inviati dal client al server.**

**In una pagina HTML è necessario inserire i tag del form per la raccolta delle informazioni.**

**Tramite il form i dati raccolti sono inviati al server.**

**Il server si occupa della loro elaborazione e della restituzione al client dei risultati.**

## FORM

La sintassi inizia sempre con la riga `<form>` e si conclude con `</form>`:

`<form>`

...

...

`</form>`

### Attributi accettati:

`action=" "`

Indica dove mandare il form. Può essere un URL, in default ha lo stesso url del sito.

`method=" "`

Indica come mandare le informazioni all'url. Può essere "post" oppure "get". L'unica differenza fra i due metodi è che post spedisce le variabili nascondendole al navigatore, mentre get inserisce le variabili nell'url del browser.

`enctype=" "`

Indica il tipo di contenuto del form quando `method="post"`.

- **application/x-www-form-urlencoded** Valore di default (se non specificato nel tag `form`) per inviare testo o valori comuni.
- **multipart/form-data** Per inviare grandi quantità di dati (file, immagini ecc.) oppure del testo che contenga alcuni caratteri non ascii.

`name=" "`

Indica il nome dell'elemento form per un eventuale richiamo di css o script. Il suo utilizzo è stato declassato a favore di `id=" "`.

`class=" "`

Specifica il nome della classe css che ne regola la grafica (è possibile utilizzare anche `style=" "`).

`title=" "`

Eventuale titolo del form

`target=" "`

Nel caso il sito utilizzi i `frame` è possibile far arrivare i dati del form in quella pagina.

## INPUT

Il tag *INPUT* associato al tipo (`type=""`) dà vita a numerosi formati per raccogliere informazioni dal navigatore (caselle di testo, pulsanti, password ecc.).

La sintassi è molto semplice, anche se ogni tipo ha la propria specifica:

`<input >`

### Attributi comuni:

`name=""` indica il nome del campo. Servirà per richiamarlo con script o altre procedure.

`type=""` button, checkbox, hidden, image, password, radio, reset, submit, text.

`id=""` indica il nome per un'eventuale richiamo css o script.

`class=""` indica il nome della classe css che ne regola la grafica (è possibile inserire anche `style=""`)

Analizziamo ogni tipo con relativi esempi. Da ricordare che non tutti i tipi hanno attributi uguali:

### `<input type="text">`

Il tipo *text* consente di inserire il testo dentro un campo composto da una sola riga (a differenza del tag *textarea*).

Codice:

`<form>`

`<input name="ciao" type="text" maxlength="20" size="10" value="inizio">`

`</form>`

Inserisci del testo.

### Attributi specifici `type="text"`:

`maxlength=""` numero max di caratteri che possiamo immettere.

`size=""` lunghezza della finestra misurata in numero di caratteri.

`value=""` la frase iniziale.

`readonly` se inseriamo questo attributo in un campo text precompilato (`input type="text" value="frase precompilata"`), il testo può solo essere letto e NON modificabile dal navigatore

**<input type="password">**

Il tipo *password* consente di inserire del testo coperto da asterischi.

Codice:

```
<form>
<input name="ciao" type="password" maxlength="4" size="7">
</form>
```

Inserisci la password.

**Attributi specifici *type="password"*:**

*maxlength="4"* numero max di caratteri da immettere.

*size="7"* lunghezza della finestra misurata in numero di caratteri.

**<input type="checkbox">**

Il *type="checkbox"* consente di inserire una casella di spunta all'interno del form. A differenza del *type="radio"* le caselle checkbox possono essere attivate senza un numero max di limite.

Codice:

```
<form>
<input name="ciao" type="checkbox" checked value="yes">
</form>
```

Premi per deselezionare.

**Attributi specifici *type="checkbox"*:**

*checked* questo attributo, inserito senza alcun valore (= " ") crea una casella già spuntata.

**<input type="radio">**

Il *type="radio"* inserisce una serie di caselle radio dove soltanto una può essere selezionata. Con l'attributo *checked* si preseleziona in automatico una casella. Di default (se non è presente *checked*) è selezionato il primo radio del form. Si suggerisce di inserire sempre l'attributo *checked* perchè non tutti i browser agiscono di default. Codice:

```
<form>
<input name="ciao" type="radio" value="1">
<input name="ciao" type="radio" value="2" checked>
<input name="ciao" type="radio" value="3">
</form>
```

Seleziona una casella

**Attributi specifici `type="radio"`:**

`name=" "` nel tipo `radio` l'attributo `name` è importante perchè raggruppa tutte le caselle `radio` che compongono il form.

`value=" "` indica il valore da associare ad un determinato `radio`.

`checked` indica quello preselezionato in automatico nel form.

**`<input type="file">`**

Il `type="file"` consente di selezionare un file (.txt immagine od altro) dal computer del navigatore per inviarlo tramite il form. In questo caso nella riga iniziale del form si deve utilizzare il metodo `post` ed `enctype="multipart/form-data"`. Codice:

```
<form action="pagina.php" method="post" enctype="multipart/form-data">  
<input name="ciao" type="file" >  
</form>
```

**Attributi specifici `type="file"`:**

`value=" "` indica il nome che in automatico associamo al file.

**`<input type="hidden">`**

Con il `type="hidden"` inseriamo un campo nascosto nel form per la trasmissione dei dati. La variabile che generiamo sarà utile per eventuali controlli o trasmissioni di dati che altrimenti andrebbero persi con l'invio del form. Codice:

```
<form>  
<input name="ciao" type="hidden" value="33">  
</form>
```

Non apparirà niente a video, ma è molto importante sapere che quando il navigatore spedisce il form sarà creata una variabile con il nome `ciao` con valore 33. Tale variabile sarà trasmessa con tutti gli altri dati del form, senza nessuna azione diretta dell'utente.

**Attributi specifici `type="file"`:**

`name=" "` indica il nome della variabile che trasmettiamo.

`value=" "` indica il valore che assume la variabile nascosta.

**`<input type="button">`**

Il tipo `button` non è da confondere con il tipo `submit` (che serve per inviare il form). Il tipo `button` si utilizza quando il form può contenere una funzione javascript, magari per aiutare nel compilamento oppure per fare un controllo.

Codice con esempio javascript:

```
<script type="text/javascript">
  function ciao3(form)
    { alert("Io sono la funzione ciao3"); }
</script>
.....
<form>
<input name="ciao" type="button" value="premi" onClick="ciao3(this.form)">
</form>
```

Premendo il tasto parte la funzione "ciao3" (un semplice alert javascript).

**Attributi specifici *type="button"*:**

*value=" "* il testo che leggiamo sopra il pulsante.

*onclick=" "* indica cosa fare se cliccato (javascript).

*ondblclick=" "* indica cosa fare se cliccato con un doppio clic (javascript).

***<input type="reset">***

Il pulsante *type="reset"* consente di resettare il form allo stato iniziale. In pratica cancella tutti i dati inseriti dal navigatore e riporta tutto il form allo stato di partenza. Codice:

```
<form>
<input type="reset" value="resettare">
</form>
```

Prova ad inserire del testo e dopo clicca resetta.

**Attributi specifici *type="reset"*:**

*value=" "* il testo che leggiamo sopra il pulsante.

***<input type="submit">***

Il tipo *type="submit"* consente di inviare il form alla pagina o programma prestabilito all'inizio. Se cliccato invieremo tutte le informazioni che il navigatore ha inserito nel form. Codice:

```
<form>
<input type="submit" value="spedisci">
</form>
```

Se clicchi ritorni su questa pagina.

**Attributi specifici `type="submit"`:**

`value=" "` il testo che leggiamo sopra il pulsante di invio.

**`<input type="image">`**

Il `type="image"` crea un pulsante di tipo `submit` (per l'invio del form) con l'inserimento di un'immagine. Si consiglia sempre di inserire l'attributo `alt=" "` per consentire leggere un testo alternativo nel caso l'immagine non sia raggiungibile o non visibile dal browser.

L'invio dei dati comprende anche l'invio delle coordinate del click (x ed y) consentendo di applicare un'azione diversa a seconda della posizione del clic.

In pratica otterremo l'invio del form con due variabili chiamate:

`nome.x`

con `nome` che prende il valore dell'attributo `name=" "`. Questa variabile trasmette la coordinata del clic a partire dalla sinistra dell'immagine (valore in pixel).

`nome.y`

con `nome` che prende il valore dell'attributo `name=" "`. Questa variabile trasmette la coordinata del clic a partire dall'alto dell'immagine (valore in pixel).

Codice:

```
<form>  
<input name="ciao" type="image" src="prova1.jpg" alt="immagine di prova">  
</form>
```

Per verificare l'invio delle coordinate (x ed y), in questo caso chiamate `ciao.x` e `ciao.y` provate a cliccare sull'immagine e leggere il risultato sulla barra dell'URL del browser. Se passate sull'immagine dopo averla già cliccata almeno una volta noterete che punta alle coordinate dell'ultimo clic. Tali valori varieranno in quelli correnti quando cliccherete di nuovo.

**Attributi specifici `type="image"`:**

`name=" "` il nome della variabile che sarà spedita con il clic e sarà associata al valore `.x` e `.y` delle coordinate.

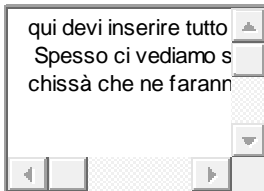
`src=" "` il percorso URL dell'immagine da visualizzare.

`alt=" "` il testo che apparirà se l'immagine non è visibile.

## TEXTAREA

Consente di inserire del testo da leggere all'interno di un form.

```
<form>
<textarea rows="5" cols="20" >
qui devi inserire tutto il testo che ti pare.
Spesso ci vediamo strane leggi e norme,
chissà che ne faranno dei nostri form...
</textarea>
</form>
```



### Attributi *textarea*:

*name*=" " Il nome associato alla textarea.

*title*=" " Il titolo della textarea.

*rows*=" " Il numero di righe che compone la textarea. Inserire un numero intero senza unità di misura, che corrisponde al numero di caratteri che vogliamo vedere.

*cols*=" " Il numero di colonne che compone la textarea. Inserire un numero intero senza unità di misura, che corrisponde al numero di caratteri che vogliamo vedere.

*id*=" " Il nome identificativo per un richiamo css o script.

*class*=" " Il nome della classe che regola la grafica della textarea, possiamo anche inserire direttamente *style*=" ".

*readonly* Se inseriamo questo attributo in una textarea precompilata, il testo può solo essere letto e NON modificabile dal navigatore

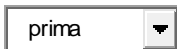
*wrap*=" " Imposta il tipo di tabulazione del testo. Può avere tre valori:

- *virtual* il testo va a capo ma viene spedito su una sola linea.
- *physical* il testo viene spedito come è scritto.
- *off* il testo non va a capo e viene spedito così.

## SELECT

Il tag *SELECT* consente al navigatore di scegliere fra più opzioni. Le opzioni possono essere singole o multiple. Il tag *SELECT* al suo interno contiene i tag *OPTION*, che in pratica sono le opzioni cliccabili disponibili.

```
<form>
<select name="ciao" size="1" >
<option selected value="1"> prima </option>
<option value="2"> seconda </option>
<option value="3"> terza </option>
<option value="4"> quarta </option>
</select>
</form>
```



### Attributi *select*:

*name*="" il nome che identifica le opzioni.

*title*="" il titolo del tag *select*.

*id*="" il nome identificativo per un eventuale richiamo css o script.

*class*="" il nome della classe css che regola l'aspetto grafico del tag *select* possiamo inserire anche direttamente *style*="".

*size*="" numero max di righe da vedere.

*multiple* se inseriamo questo valore si abilita scelte multiple.

### Attributi *option*:

*selected* l'opzione visualizzata in partenza. Se omesso non si vedrà niente oppure il primo option (dipende dal tipo di browser).

*value*="" il valore dell'opzione.

Se inseriamo il valore *multiple* possiamo inserire più *selected* contemporaneamente. Se abbiamo più *selected* ma non abbiamo inserito *multiple* nel tag *<select>* creeremo un errore.

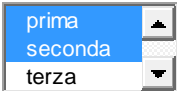
### Esempio **SELECT multiple**:

```
<form>
<select multiple name="ciao" size="3">
<option selected value="1"> prima </option>
```

```

<option value="2"> seconda </option>
<option value="3"> terza </option>
<option value="4"> quarta </option>
<option value="5"> quinta </option>
<option value="6"> sesta </option>
</select>
</form>

```



Puoi selezionare più opzioni contemporaneamente trascinando con il mouse.

## SELECT OPTGROUP

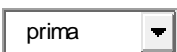
Questo tag si utilizza con *SELECT* solo se all'interno dobbiamo inserire dei gruppi di opzioni.

Il nome dell'opzione è inserito dall'attributo *LABEL*=" ".

```

<form>
<select name="ciao">
<optgroup label="Primo gruppo">
<option value="1"> prima </option>
<option value="2"> seconda </option>
</optgroup>
<optgroup label="Secondo gruppo">
<option value="3"> terza </option>
<option value="4"> quarta </option>
</optgroup>
</select>
</form>

```



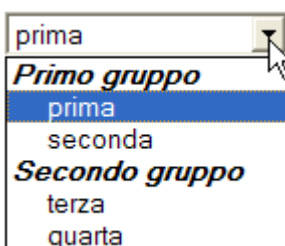
Attributi optgroup:

**title**=" " *il titolo del gruppo.*

**id**=" " *il nome identificativo per un eventuale richiamo css o script.*

**class**=" " *il nome della classe css che regola l'aspetto grafico del tag optgroup possiamo inserire anche direttamente style=" ".*

**label**=" " *Il nome del gruppo di opzioni.*



## FIELDSET - LEGEND

Nei form è possibile inserire una legenda, in pratica un semplice riquadro (con angoli arrotondati in alcuni browser) che ne attesta la composizione e dal quale possiamo ricavare un titolo.

Questa possibilità offre al webmaster una soluzione che se unita ai css può eliminare molti dei problemi grafici scaturiti dai form.

La delimitazione del form fornisce all'utente la sensazione di un modulo da compilare, e concentra la sua attenzione solo sui campi che lo compongono.

Il codice utilizza due tag:

- `<fieldset>` Che delimita tutti i campi del form.
- `<legend>` Dove possiamo inserire il titolo del form.

Codice:

```
<form>
<fieldset>
<legend>Legenda form</legend>
<input name="primo dato" type="text" tabindex="1"> Primo dato<br>
<input name="personali_nome" type="text" tabindex="2"> Secondo dato<br>
<input type="reset" value="reset">
<input type="submit" value="invia">
</fieldset>
</form>
```

In pratica:

Legenda form

<input type="text"/>	Primo dato
<input type="text"/>	Secondo dato
<input type="reset" value="reset"/>	<input type="submit" value="invia"/>