

# HTML

## Interazione con l'utente



# L'interazione con l'utente

Il web è interattivo e per questo motivo una pagina html deve prevedere la possibilità, per l'utente, di inviare informazioni o effettuare delle scelte.

Il linguaggio HTML, essendo un semplice linguaggio di markup, non è in grado di elaborare queste informazioni, ma solo di raccoglierle.

L'elaborazione delle informazioni è delegata ad appositi piccoli programmi, detti **script**, inseriti all'interno della pagina HTML.

# L'interazione con l'utente

Se l'elaborazione viene effettuata direttamente dal browser si sta programmando lato client.

Se l'elaborazione viene effettuata dal computer che ospita la pagina si sta programmando lato server.

# L'interazione con l'utente

Il più comune linguaggio di programmazione lato client è javascript.

I maggiori linguaggi di programmazione lato server sono PHP, Java (JSP), ASP.

Lo script costituisce il motore dell'interazione, mentre in HTML possiamo creare l'ambiente.

# L'interazione con l'utente

L'interazione con l'utente consiste nella raccolta e invio dei dati, nella loro elaborazione ed invio della risposta.

Solitamente ciò è organizzato creando:

- una pagina principale html che contiene le strutture che consentono all'utente di effettuare delle scelte, scrivere del testo, inserire un'immagine (inserire i dati) ed inviarli (bottoni di comando);
- una pagina secondaria che viene richiamata dalla principale e che effettua “il lavoro” vero e proprio di raccogliere e processare i dati e di inviare le risposte.

# L'interazione con l'utente

## I form

In HTML la raccolta dei dati è effettuata tramite i **form** (moduli), rappresentati dal tag `<form>` `</form>`.

All'interno del form possono essere collocati oggetti differenti (caselle di testo, pulsanti, checkbox, etc.) destinati a raccogliere le informazioni sulle quali lavoreranno gli script.

# L'interazione con l'utente

## I form

Il tag `<form>` `</form>` possiede i seguenti attributi:

- `name` → specifica il nome del form
- `action` → richiama la destinazione del form
- `method` → specifica la modalità di passaggio dei parametri (POST, GET)
- `target` → specifica se i dati del form saranno aperti nella stessa pagina o in un'altra (`_self`, `_blank`)
- `enctype` → specifica il tipo di codifica da utilizzare (`text/plain=default`, `multipart/form-data` per le immagini)

# L'interazione con l'utente

## I form

L'attributo action consente di :

- chiamare uno script locale → indicando nome ed estensione dello script
- chiamare uno script remoto → indicando l'URL
- mandare una mail → <mailto:indirizzo>



# L'interazione con l'utente

## I form

I metodi GET e POST consentono entrambi l'invio dei dati al server ma ...

GET ha limitazioni sulla lunghezza delle stringhe inviate ed il browser mette in chiaro i parametri nell'URL della pagina.

POST consente di inviare grandi quantità di dati in modo che non siano visibili all'utente.

# L'interazione con l'utente

## I form

La chiamata allo script funziona come la chiamata di funzione nei linguaggi di programmazione tradizionali.

I dati inseriti nel form sono i parametri sui quali lavora lo script.

I metodi GET e POST collocano i dati del form in un array associativo (chiamato `$_GET` o `$_POST`) in cui gli indici sono i nomi degli elementi del form che hanno raccolto i dati (valori dell'attributo name).

Dalla versione 5 di PHP i valori degli indici vanno messi tra apici singoli o doppi

# L'interazione con l'utente

## I form

Impostando `register_globals=On` su `php.ini`, oltre agli array `$_GET` e `$_POST` avremo anche delle variabili globali con identificatore pari al valore dell'attributo `Name`, che contengono direttamente i valori corrispondenti.

Dalla versione 4.2.0 di PHP, il valore di default di `register_globals` è `Off` e gli sviluppatori di PHP sconsigliano di rimetterlo ad `On`, perché, utilizzando gli array superglobali `$_GET` e `$_POST`, si rende il codice più chiaro e più sicuro.

# L'interazione con l'utente

Il tag `<input>` consente di inserire nel form oggetti per l'inserimento di un valore o per l'effettuazione di una scelta. Il tipo di oggetto viene impostato assegnando il valore al tag.

# L'interazione con l'utente

## Caselle di testo

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

Consente di inserire nel form una casella di testo.

L'attributo **name** è l'identificativo del parametro che contiene il contenuto della casella di testo quando esso viene inviato allo script.

L'attributo **size** consente di limitare la lunghezza del testo da inserire nella casella.

```
<input type="text" name="nomeecognome" size="25">
```

# L'interazione con l'utente

## Caselle di testo

E' bene associare un'etichetta alle caselle di testo che spieghi cosa l'utente deve inserire in quella casella.

L'attributo for del tag <label> deve essere uguale a quello dell'attributo name del tag <input>.

```
<label for="nomeecognome"> NOME E COGNOME <\label>  
<input type="text" name="nomeecognome" size="25">
```

# L'interazione con l'utente

## Caselle di testo - Password

Sono come le caselle di testo ma ciò che viene scritto non è riportato in chiaro ma sotto forma di puntini.

```
<input type="password" name="pwd" size="25">
```

# L'interazione con l'utente

## Pulsanti

Sono sempre tag di tipo input ma l'attributo type deve essere impostato al tipo di bottone prescelto.

**Reset** → cancella il contenuto di tutte le caselle di testo di quel form.

**Submit** → spedisce al server i dati contenuti nelle caselle di testo di quel form.

**Button** → bottone generico (solo programmazione lato client). Il programmatore deve gestire gli eventi e le azioni conseguenti.



# L'interazione con l'utente

## Pulsanti

L'evento tipo associato ad un bottone è la pressione (click).

L'evento è gestito tramite l' attributo

**onclick="nomefunzione()"**

da inserire tra gli attributi del tag <input>.

# L'interazione con l'utente

## Pulsanti

La funzione dovrà appartenere ad uno script che deve essere richiamato col percorso completo e l'estensione nell'header della pagina

Esempio:

```
<script language="javascript" src="nomescript.js">  
</script>
```

# L'interazione con l'utente

## Pulsanti

L'attributo **value** consente di inserire una scritta descrittiva nel bottone.

L'attributo **name** consente di attribuire un identificativo all'oggetto, in modo da poterlo richiamare in seguito.

# L'interazione con l'utente

## TextArea

Sono aree di testo più ampie delle caselle di testo, simili a piccoli editor. La dimensione dell'area si imposta indicando il numero di righe e colonne di cui deve essere composta.

```
<textarea name="nome_area" rows="numero_righe"  
  cols="numero_colonne"> testo che appare di default  
</textarea>
```

# L'interazione con l'utente

## **TextArea**

L'intero testo viene inviato al server che lo accoglie in una variabile di tipo stringa.

# L'interazione con l'utente

## Opzioni esclusive (Radio Button)

Permettono una scelta tra n alternative. Le scelte sono mutuamente esclusive.

Le opzioni si implementano ognuna con un suo tag ma devono tutte avere lo stesso attributo Name.

```
<input type=radio name="nome" value=valore> testo
```

L'attributo checked consente di indicare un valore di default.

# L'interazione con l'utente

## Opzioni multiple (checkbox)

Permettono più scelte tra n alternative.

Le opzioni si implementano ognuna con un suo tag ma devono tutte avere lo stesso attributo Name.

```
<input type="checkbox" name="nome" value=valore> testo
```

L'attributo checked consente di indicare un valore di default.

# L'interazione con l'utente

## Menù a tendina

Permettono una o più scelte tra n alternative.

Le opzioni si implementano con un tag simile a quello degli elenchi puntati o numerati.

Numero di opzioni visibili

Consente la scelta multipla

```
<select name="nome" size=valore multiple>
```

```
<option selected value=valore inviato> Opzione1 </option>
```

...

```
</select>
```

L'attributo selected consente di indicare una opzione di default.



# L'interazione con l'utente

## Passaggio di parametri tramite indirizzo URL

E' possibile passare dei parametri ad uno script php inserendoli direttamente nell'url della pagina Web che si sta richiamando.

E' una implementazione diversa della modalità GET.

<http://localhost/esempio.php?nomevariabile=valore>

Il php riceve valore in `$_GET["nomevariabile"]`.

**Posso fare una serie di assegnazioni separandole con &.**