

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
</body>
</html>
```

## Web Coding

*Prototipazione di ipertesti e siti web in HTML  
ed introduzione alla creazione di stili grafici con fogli di stile CSS.*

Unità Didattica UD02: Impostare URL, link, immagini, tabelle e moduli in HTML

*prof. Giovanni Borga*

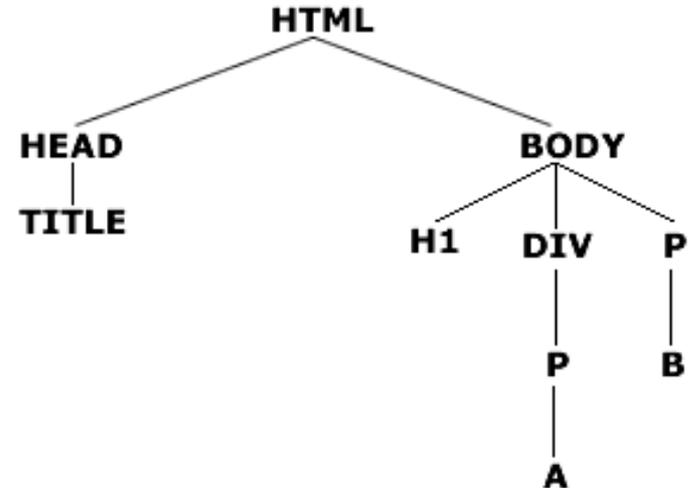
# **Struttura di un documento HTML ed ereditarietà**

*Gerarchia degli elementi e propagazione delle proprietà*

# Struttura di un documento HTML

Osserviamo il codice a sinistra; la sua rappresentazione gerarchica può essere l'immagine a destra:

```
<html>
  <head>
    <title>Struttura del documento</title>
  </head>
  <body>
    <h1>Titolo</h1>
    <div>
      <p>
        Primo <a href="pagina.htm">paragrafo</a>
      </p>
    </div>
    <p>Secondo<b>paragrafo</b></p>
  </body>
</html>
```



# Gerarchia degli elementi di un documento

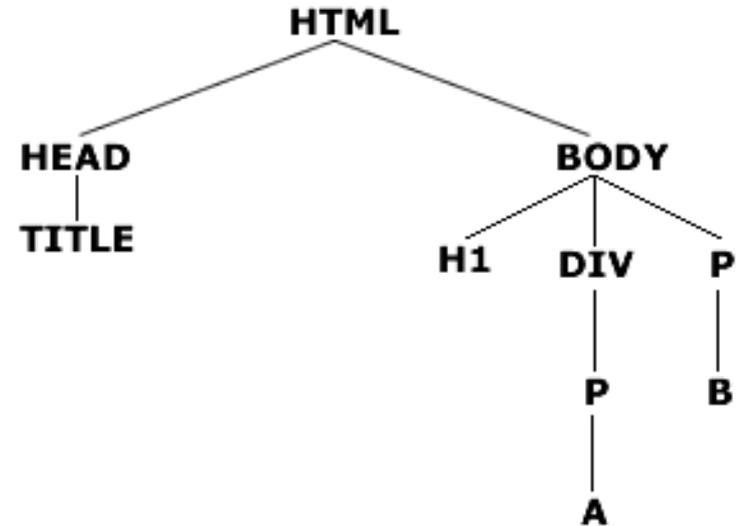
Il documento è in effetti una perfetta forma di gerarchia ordinata in cui tutti gli elementi hanno tra di loro una relazione del tipo **genitore-figlio**.

**PARENT-CHILD** in inglese; dicitura importante perchè utilizzata nei linguaggi come DOM o Javascript: **ogni elemento è genitore e/o figlio di un altro**.

- Un elemento si dice genitore (parent) quando contiene altri elementi.
- Si dice figlio (child) quando è racchiuso in un altro elemento.

In base a queste indicazioni il nostro documento ha le seguenti caratteristiche:

1. *BODY è figlio di HTML,*
2. *BODY è anche genitore di H1, DIV e P.*
3. *P è a sua volta genitore di un elemento B.*



## Gerarchia ed ereditarietà

Anche BODY è genitore di B? Non proprio.

Serve un'altra distinzione, quella tra **antenato** (*ancestor*) e **discendente** (*descendant*).

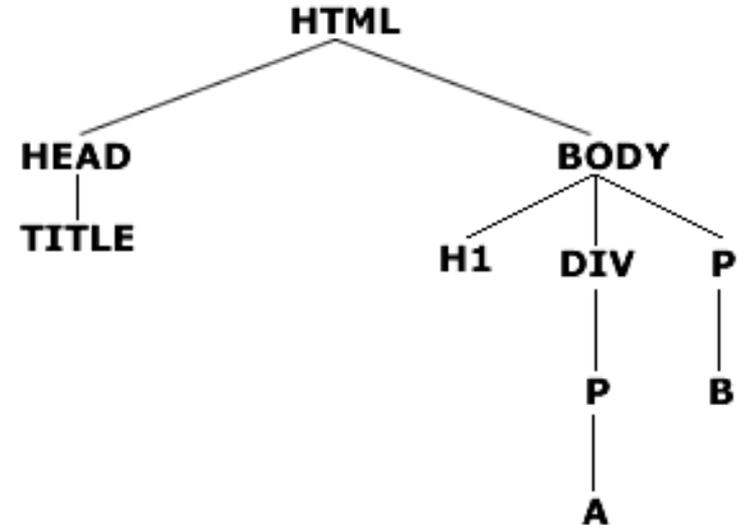
**La relazione parent-child** è valida solo se tra un elemento e l'altro si scende di **UN livello**. (Esattamente come in un albero familiare si indica la relazione tra padre e figlio).

- HEAD è figlio di HTML
- A è figlio di P
- ... ecc.

Tra **DIV e A**, invece si scende **di due livelli**: diciamo allora che **DIV è un antenato di A** e che questo è rispetto al primo un discendente.

C'è un solo elemento che racchiude tutti e non è racchiuso: HTML. Si dice infatti che HTML è l'elemento radice (root).

*NB: attenzione a non fraintendere: l'elemento radice di un documento (X)HTML non è BODY. HTML inoltre, non è una semplice dichiarazione ma a tutti gli effetti un TAG alla stregua di tutti gli altri.*



# Ereditarietà delle caratteristiche

In un documento HTML

**tutte le proprietà di un tag vengono applicate anche a tutti i tag discendenti.**

*Nel caso di **proprietà in conflitto**, ovvero quando sia per il tag antenato sia per il discendente vengano applicati valori diversi per una stessa caratteristica, **prevale il valore attribuito al discendente***

# **URL, nomi e percorsi**

*Indirizzi per il collegamento delle risorse web*

## Riferimenti alle risorse esterne alla pagina

Un documento come una pagina web è composta dalla parte di testo e da altre risorse di carattere non testuale come immagini, contenuti multimediali ma anche librerie di stili e di script.

**Queste risorse sono di fatto «esterne» alla pagina stessa. Non sono cioè contenute nel file HTML.**

Un esempio intuitivo di questo tipo di risorse sono le immagini; esse infatti sono semplicemente «**collegate**» alla pagina, non inserite nel codice.

Per queste risorse, il codice HTML prevede l'utilizzo di «**indirizzi**» da cui il browser può prelevare (scaricare) la risorsa stessa.

Ecco perché, anche una singola pagina web non è quasi mai un solo file ma un gruppetto di file: tipicamente troviamo:

- ***il file HTML***
- ***le immagini***
- ***i fogli di stile***
- ***gli script***

## Percorsi assoluti e percorsi relativi

I siti web composti di poche pagine possono essere gestiti utilizzando una sola cartella. La generalità dei siti però ha molte pagine per cui è necessario **mettere ordine alla numerosità dei file necessari**.

Nella maggioranza dei siti web abbiamo esigenze come ad esempio:

- *inserire le **immagini** in una cartella specifica o in più cartelle dedicate*
- *dividere il sito in varie **sezioni** utilizzando cartelle diverse*
- *gestire tutti i **documenti** dello stesso tipo in modo più efficiente organizzandoli in cartelle dedicate*

**I siti web sono dunque organizzati in strutture ordinate.** Si parla anche di **albero di un sito** a sottolineare la natura strettamente gerarchica di tale struttura.

L'organizzazione di un sito in cartelle e sottocartelle è da tempo la prassi. Per questo occorre imparare a muoverci tra i vari file che costituiscono il sito, in modo da essere in grado di **creare collegamenti** verso i documenti che risiedono ai diversi livelli di profondità della struttura.

**I collegamenti si effettuano sulla base di un «indirizzo», detto anche «percorso» che può essere di due tipi:**

- **ASSOLUTO**
- **RELATIVO**

# I percorsi assoluti e gli URL

Indipendentemente dalla cartelle in cui è localizzata la risorsa a cui vogliamo puntare, i percorsi assoluti contengono **per esteso la sequenza delle cartelle e sottocartelle che portano dalla radice alla risorsa stessa**.

Ad esempio, l'indirizzo <http://www.html.it/css/index.html> è un percorso assoluto ed è composto dei seguenti elementi:

**http://**

*Indica al browser di utilizzare il protocollo per navigare nel web (l'http)*

**www.html.it/**

*Indica che la risorsa si trova sul sito «www.html.it»*

**css/**

*Indica che la risorsa indicata si trova all'interno della cartella "css"*

**index.html**

*Indica che il file da collegare è quello chiamato «index.html»*

**Per tutte le risorse ospitate sui server web pubblici esiste dunque un percorso assoluto univoco da cui prelevarle; questo percorso è detto URL (Universal Resource Locator).**

**NB:** per collegare una risorsa ospitata su un server diverso da quello del nostro sito è necessario utilizzare un URL assoluta.

## Percorsi relativi

Raramente si sviluppa un sito web operando direttamente all'interno delle cartelle pubbliche del server. Spesso **si lavora «in locale»** ovvero sul nostro computer o su un computer della nostra rete interna, oppure si lavora su un server provvisorio e si effettua il «deploy» definitivo dei files una volta terminato lo sviluppo.

**I tutti questi casi l'utilizzo di percorsi assoluti è un serio problema.** Infatti lo spostamento dei files comporta spesso il cambiamento della «radice» dei percorsi, ovvero il nome o l'indirizzo IP del server provvisorio oppure anche la lettera che identifica l'unità disco in cui sono memorizzati i files.

**E' prassi consolidata utilizzare per tutti i riferimenti alle risorse dello stesso sito dei percorsi relativi**

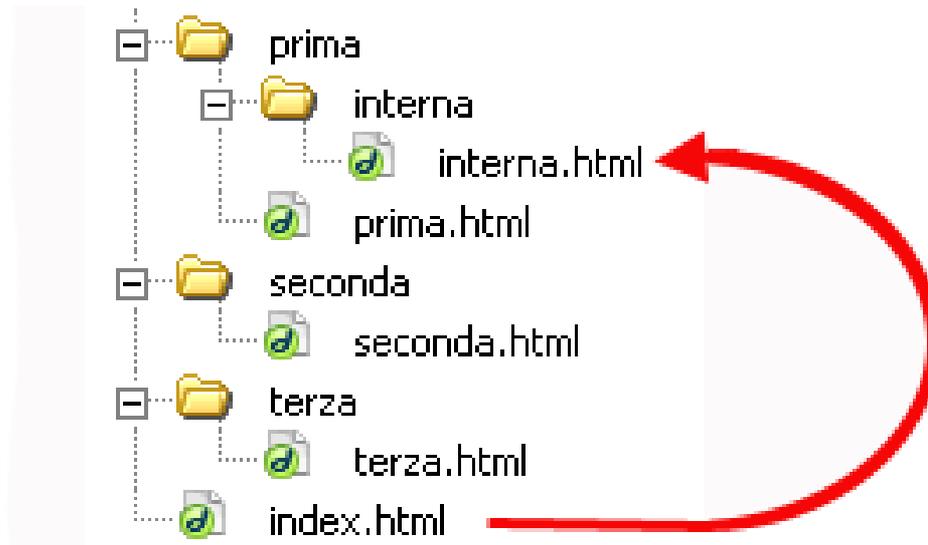
**I percorsi relativi fanno riferimento alla posizione delle risorse in base a quella in cui ci si trova il documento che le collega.**

Ad esempio la posizione delle immagini di una pagina web è relativa a quella della posizione del file HTML che le deve visualizzare.

*NB: il percorso relativo più semplice è quello in cui documento e risorsa collegata si trovano nella stessa cartella. In questo caso è sufficiente scrivere il nome del file senza i nomi di alcuna cartella.*

## Percorsi relativi

Vediamo ora un esempio in cui dalla pagina «index.html» si voglia far riferimento alla pagina «interna.html», che si trova all'interno della cartella «interna», che a sua volta si trova all'interno della cartella «prima».

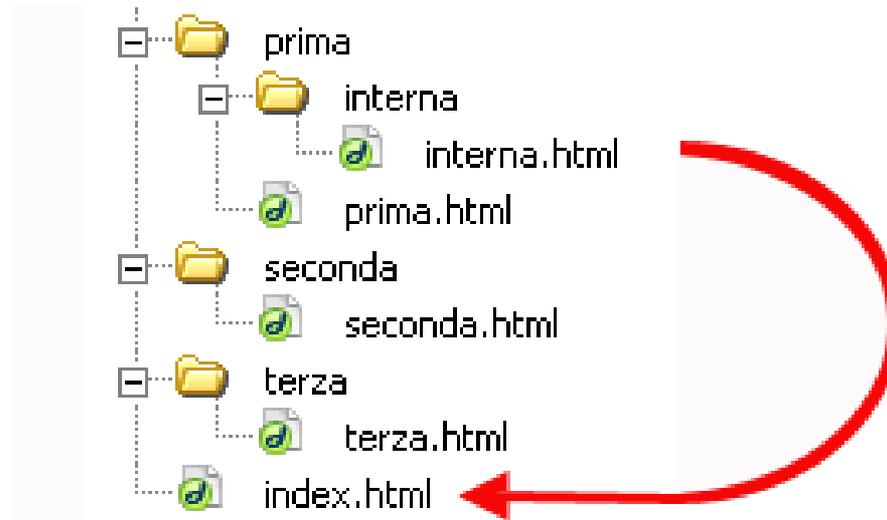


Il percorso da indicare sarà seguente: «**prima/interna/interna.html**»

*Si noti che il percorso non inizia da una «radice» («http://», «C:/» o simili) ma dalla prima sottocartella in cui si deve passare per raggiungere il file «interna.html».*

## Percorsi relativi

Ora un esempio opposto al precedente: dalla pagina «interna.html» vogliamo far riferimento a una pagina ("index.html") che si trova più in alto di due livelli:



In questo caso il percorso sarà il seguente: «../../**index.html**»

**La notazione «../»** (doppio punto + slash) è appunto la notazione che indica di risalire di un livello l'albero gerarchico delle cartelle. Nell'esempio, dovendo risalire due livelli occorre inserire la notazione due volte.

## Altre note

A volte si può trovare un percorso indicato in questo modo: «**/css/index.html**». Il primo elemento è uno slash invece che il nome di una radice o di una cartella.

Si tratta di una notazione che in origine era supportata dai sistemi Unix ma che oggi è supportata anche da quelli Windows:

il **carattere '/'** indica la directory principale del sito, altrimenti detta «root».

Si tratta dunque di un modo alternativo di esprimere percorsi assoluti.

Altra Un'altra cosa importante da sapere è che quando metterete il vostro sito all'interno dello spazio web, l'indicazione della index all'interno di una directory è facoltativa. Al posto di questo:

`http://www.html.it/css/index.html`

è sufficiente indicare la directory:

`http://www.html.it/css/`

Verificate solo con il vostro gestore dello spazio web (cioè "hosting"), se le pagine index della directory devono avere forma `index.html`, `index.htm`, `index.asp`, `index.php`, `home.asp`, o altro.

## Note sulla nominazione dei file

Una volta realizzato un sito web, per pubblicarlo occorre caricarlo su un server pubblico.

Esistono due famiglie di sistemi operativi orientati a svolgere la funzione server: **Windows e Unix**. Questi due sistemi operativi utilizzano differenti modi per gestire i file; vediamo alcuni accorgimenti:

- **Maiuscole e minuscole** possono fare la differenza. In ambiente Unix (e Linux) la stessa lettera scritta nei due modi viene trattata come due caratteri distinti mentre in ambiente Windows questo non accade. *Quindi il file **Index.html** può convivere con un file **index.html** SOLO su server Unix/Linux, e sono due file distinti.*
- E' consigliabile non lasciare spazi vuoti nei nomi dei file (gli spazi vuoti non sempre vengono interpretati correttamente), meglio ovviare a questa necessità con un "trattino basso" (cioè "\_").  
*Ad esempio: mio\_file.html*

### Nota sui collegamenti:

Quando si crea un collegamento occorre fare attenzione a non avere una notazione simile a questa:

«**file:///C|cartella/nomefile.html**»

Succede utilizzando alcuni editor lavorando in locale: significa che si sta facendo riferimento (assoluto) al proprio computer e quindi quando si caricano i file sullo spazio spazio web i collegamenti non funzioneranno più.

# **Elenchi, link, immagini e tabelle**

*Elementi per costruire liste, collegamenti, inserire immagini e tabelle*

# Gli elenchi dell'HTML

Se abbiamo la necessità di inserire un elenco di termini, possiamo utilizzare le «liste», che sono sostanzialmente di due tipi:

- Elenchi ordinati
- Elenchi non ordinati

Entrambi i tipi funzionano nel medesimo modo: si apre il tag, si elencano i vari elementi della lista (ciascuno con il proprio tag), si chiude il tag dell'elenco. La sintassi ha quindi questa forma:

```
<elenco>  
  <elemento>nome del primo elemento  
  <elemento>nome del secondo elemento  
</elenco>
```

come si può vedere, il tag che individua l'elemento della lista non ha bisogno di chiusura (la sua chiusura, in questo caso, è opzionale).

Le liste di definizioni hanno una struttura leggermente diversa che vedremo a breve.

# Elenchi ordinati

Detti anche «elenchi numerati», gli elenchi ordinati sono contraddistinti dall'**enumerazione degli elementi** che compongono la lista.

Avremo quindi una serie progressiva ordinata e individuata da lettere o numeri.

Il tag da utilizzare per aprire un elenco ordinato è **<ol>** («ordered list») e gli elementi sono individuati dal tag **<li>** («list item»)

Ecco un esempio di codice

*.... contenuto che precede la lista*

```
<ol>  
  <li>primo elemento</li>  
  <li>secondo elemento</li>  
  <li>terzo elemento</li>  
</ol>
```

*... cesto che segue la lista*

1. primo elemento
2. secondo elemento
3. terzo elemento

## Elenchi ordinati - tipi di numerazione

Da notare che il tag che individua l'**elenco** lascia una riga di **spazio prima e dopo il testo** che eventualmente lo circonda (come avviene per il <p>); fa eccezione però l'inclusione di un elenco all'interno di un elenco preesistente: in questo caso non viene lasciato spazio, né prima, né dopo.

Gli **elementi dell'elenco sono sempre rientranti** di uno spazio verso destra come accade utilizzando i programmi di videoscrittura.

Lo stile di **enumerazione visualizzata di default** dal browser è quello **numerica**.

E' però possibile indicare uno stile differente specificandolo per mezzo dell'attributo **type**. Ad esempio:

```
<ol type="a">
```

```
<li>primo elemento
```

```
<li>secondo elemento
```

```
<li>terzo elemento
```

```
</ol>
```

Valore dell'attributo type	Stile di enumerazione		
type="1" (è così di default)	<b>numeri arabi</b>	<pre>&lt;ol type="1"&gt; &lt;li&gt;primo&lt;/li&gt; &lt;li&gt;secondo&lt;/li&gt; &lt;li&gt;terzo&lt;/li&gt; &lt;/ol&gt;</pre>	1. primo 2. secondo 3. terzo
type="a"	<b>alfabeto minuscolo</b>	<pre>&lt;ol type="a"&gt; &lt;li&gt;primo&lt;/li&gt; &lt;li&gt;secondo&lt;/li&gt; &lt;li&gt;terzo&lt;/li&gt; &lt;/ol&gt;</pre>	a. primo b. secondo c. terzo
type="A"	<b>alfabeto maiuscolo</b>	<pre>&lt;ol type="A"&gt; &lt;li&gt;primo&lt;/li&gt; &lt;li&gt;secondo&lt;/li&gt; &lt;li&gt;terzo&lt;/li&gt; &lt;/ol&gt;</pre>	A. primo B. secondo C. terzo
type="i"	<b>numeri romani minuscoli</b>	<pre>&lt;ol type="i"&gt; &lt;li&gt;primo&lt;/li&gt; &lt;li&gt;secondo&lt;/li&gt; &lt;li&gt;terzo&lt;/li&gt; &lt;/ol&gt;</pre>	i. primo ii. secondo iii. terzo
type="I"	<b>numeri romani maiuscoli</b>	<pre>&lt;ol type="I"&gt; &lt;li&gt;primo&lt;/li&gt; &lt;li&gt;secondo&lt;/li&gt; &lt;li&gt;terzo&lt;/li&gt; &lt;/ol&gt;</pre>	I. primo II. secondo III. terzo

## Elenchi non ordinati

Anche detti «**elenchi puntati**», gli elenchi non ordinati sono individuati dal tag `<ul>` ("unordered list")

Gli elementi dell'elenco sono contraddistinti anch'essi dal tag `<li>`

```
<ul>
  <li>primo elemento
  <li>secondo elemento
  <li>terzo elemento
</ul>
```

Il tipo di segno grafico utilizzato per individuare gli elementi dell'elenco di default dipende dal browser, ma di solito è un «pallino pieno».

Anche per questo simbolo sono possibili alcune varianti sempre utilizzando l'attributo **TYPE** del tag `<ul>`.

Valore dell'attributo type	Stile di enumerazione		
type="disc" (è così di default)	visualizza un <b>"pallino" pieno</b> . È la visualizzazione di default	<pre>&lt;ul type="disc"&gt;   &lt;li&gt;primo&lt;/li&gt;   &lt;li&gt;secondo&lt;/li&gt;   &lt;li&gt;terzo&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>primo</li> <li>secondo</li> <li>terzo</li> </ul>
type="circle"	visualizza un <b>cerchio vuoto</b> al proprio interno	<pre>&lt;ul type="circle"&gt;   &lt;li&gt;primo&lt;/li&gt;   &lt;li&gt;secondo&lt;/li&gt;   &lt;li&gt;terzo&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>primo</li> <li>secondo</li> <li>terzo</li> </ul>
type="square"	Visualizza un <b>quadrato pieno</b> al proprio interno	<pre>&lt;ul type="square"&gt;   &lt;li&gt;primo   &lt;li&gt;secondo   &lt;li&gt;terzo &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>primo</li> <li>secondo</li> <li>terzo</li> </ul>

*NB: anche l'attributo type è **deprecato dal W3C**, perché si tratta di una caratteristica che riguarda la formattazione, e dunque andrebbe effettuata utilizzando i CSS. Con i fogli di stile c'è infatti anche la possibilità di scegliere un'immagine (ad esempio una GIF) come segno distintivo per l'elenco puntato.*

# Elenchi nidificati

Nel caso di annidamento delle liste, **indentazione e simboli** vengono adeguati automaticamente:

```
<ul>
  <li>primo della 1a lista</li>
  <li>secondo della 1a lista</li>
</ul>
<ul>
  <li>primo della 2a lista</li>
  <li>secondo della 2° lista</li>
</ul>
  <li>primo della 3a lista</li>
</ul>
</ul>
  <li>terzo della 1a lista</li>
</ul>
```

- primo della 1a lista
- secondo della 1a lista
  - primo della 2a lista
  - secondo della 2a lista
    - primo della 3a lista
  - terzo della 1a lista

# Link e ipertestualità

Una delle caratteristiche su cui si basa il web è il fatto di essere costituito non da testi ma da **ipertesti**.

**I collegamenti ipertestuali – o link – sono il «ponte»  
che consente di passare da un testo ad un altro**

I link sono formati da due componenti:

- Il **contenuto «attivo»**, ovvero quella parte della pagina che può essere «cliccata» per spostarsi ad un'altra pagina (tipicamente un testo o un'immagine). Essendo la parte fondamentale dell'interazione, l'utente deve essere sempre in grado di distinguerlo dal resto del contenuto della pagina.
- **La risorsa** verso cui il collegamento punta.  
Può essere o un'altra pagina (sullo stesso sito o su un sito diverso), oppure un collegamento «interno» a un punto della pagina in cui si trova.

## Link «esterni»

La sintassi per creare un link con riferimento a una pagina web diversa da quella di partenza è il seguente:

```
<a href="http://www.iuav.it/">Università Iuav di Venezia</a>
```

Il nome del tag <a> deriva da «anchor», ovvero àncora, l'elemento che tiene collegati origine e destinazione del flusso di navigazione.

Il rendering dell'esempio sarà il seguente:

[Università Iuav di Venezia](http://www.iuav.it/)

Facendo «click» sopra la scritta, l'utente visualizzerà la home page dell'Università Iuav raggiungibile dell'indirizzo inserito nell'**attributo HREF del tag <a>**.

Per ciò che riguarda la sintassi, il tipo di documento di destinazione è indifferente. Sia che si tratti di una pagina HTML di un sito, un'immagine, un file .pdf , un file .zip, o un file .exe la sintassi è la stessa; sarà il browser a comportarsi in modo diverso a seconda del tipo di risorsa.

Tipicamente:

- *Le immagini **.gif**, **.jpg**, **.png** vengono direttamente visualizzate nel browser*
- *I documenti **.html** vengono direttamente visualizzate nel browser*
- *Alcuni documenti come i **.pdf** o i **.doc** possono venire visualizzati dal browser tramite un «plugin» oppure scaricati come download con relativa richiesta della posizione in cui salvare il file.*
- *I file **.zip**, file **.exe** e molti altri vengono scaricati con la funzione di download.*

## Protocolli nell'HREF

La navigazione tra le pagine web e i download avvengono secondo un protocollo definito HTTP (Hypertext Transfer Protocol).

Esistono altri **protocolli associati ad altre funzioni** come ad esempio l'FTP definito per trasferire files tra cartelle locali e remote.

Un protocollo specifico è denominato **MAILTO**. Serve ad attivare un programma di posta elettronica. Questo protocollo è utilizzabile nei link con la seguente modalità:

**`<a href="mailto:pippo@gmail.com">Invia una e-mail a Pippo</a>`**

Il click sul collegamento aprirà il programma di posta elettronica predefinito dell'utente.

## Link «interni»

È possibile anche creare un **indice interno** al documento, utilizzando link «interni».

Per creare un link interno si utilizza l'attributo **NAME** anziché HREF. NAME definisce il nome del punto da contrassegnare all'interno di una pagina.

*Si noti che questa tecnica ha un senso in pagine relativamente lunghe che comportano un abbondante scroll.*

```
<a name="primo">Lorem ipsum factum ... eccetera...</a>
```

La mancanza dell'attributo HREF rende questi contrassegni «invisibili» all'utente. Tuttavia essi servono come riferimento per lo «**scroll**» **della pagina** che avviene cliccando un secondo elemento così definito:

```
<a href="#primo">Vai al primo paragrafo</a>
```

**Il cancelletto** indica che il collegamento deve cercare un contrassegno chiamato «primo» all'interno della pagina stessa.

*NB: se non si specifica il nome del contrassegno a cui si vuol puntare, viene comunque creato un link che punta ad inizio pagina.*

Per creare un indice interno alla pagina si **procede dunque in due fasi distinte:**

1. creazione del contrassegno a cui puntare (attributo NAME)
2. creazione del link con relativo riferimento interno (attributo HREF con cancelletto)

***NB: è fondamentale non confondere le due fasi.***

# Attributi dei link

Abbiamo due importanti attributi del tag `<a>`:

Il primo è **TARGET**. LA seguente sintassi:

```
<a href="http://www.iuav.it/" target = "_blank" >Università Iuav di Venezia</a>
```

Apri la nuova pagina in una **nuova finestra del browser** anziché sostituire quella di origine. E' consuetudine aprire i link su `_blank` quando si puntano pagine di altri siti mentre lasciare il default per le pagine interne.

*Altri valori per TARGET sono «\_parent», «\_self» e «\_top» ma hanno significato nell'impiego dei frames, cosa che è deprecata.*

---

Un secondo attributo è **TITLE** (da non confondere con il tag presente in intestazione):

```
<a href="http://www.iuav.it/" title="Università IUAV" >Università Iuav di Venezia</a>
```

TITLE attiva il cosiddetto «tooltip» per il link, ovvero l'etichetta che compare passando col mouse sopra il link senza cliccarlo. L'utilizzo di etichette con questo attributo è una pratica **fortemente consigliata per migliorare l'usabilità** delle pagine web.

# Immagini – i formati consentiti

Le immagini ovviamente possono essere inserite non solo come sfondo di una pagina web, ma anche come contenuto.

I formati ammessi nel Web sono sostanzialmente tre:

- **GIF** (Graphic Interchange Format):  
le GIF sono immagini con non più di **256 colori** (dunque con colori piatti e senza sfumature), come grafici o icone. Supporta la **trasparenza** anche se non con le sfumature possibili col il PNG.
- **JPG** (Joint Photographic experts Group)  
supporta i **16M di colori**, è l'acronimo del gruppo di ricerca che ha ideato questo formato, idoneo per le immagini di qualità fotografica. **Non supporta le trasparenze**. Permette livelli di **compressione** regolabili in funzione della qualità richiesta.
- **PNG** (Portable Network Graphic)  
un tipo di immagine introdotto più recentemente, elaborato dal W3C per risolvere i problemi di copyright del formato GIF (che è appunto proprietario); anch'esso supporta i **16M di colori** ed è ormai letto da tutti i browser e offre alcune caratteristiche che gli altri formati non hanno in primis il supporto al canale alfa che lo rende adatto a creare **trasparenze** molto morbide. Il livello di compressione è in genere intermedio tra GIF e JPG.

*NB: qualsiasi altro formato non è supportato (psd, tiff, ecc) e impedisce la visualizzazione dell'immagine nella pagina oppure attiva la funzione di download del file.*

# Il tag IMG e le immagini

Il tag per inserire una immagine è **<img>**; la sintassi è la seguente:

```

```

L'attributo essenziale è SRC, che significa «source», ovvero origine; tramite SRC si specifica l'indirizzo web da cui recuperare l'immagine da visualizzare. Per il riferimento al file immagine valgono tutte le regole già trattate riguardo ai percorsi e i nomi dei file.

Le immagini sono il primo elemento «esterno» alla pagina che trattiamo. Occorre comprendere come **l'accesso ai contenuti di un sito sia per lo più un processo ASINCRONO**, ovvero regolato da flussi di informazioni che hanno tempi correlati ma non sincronizzati. Un esempio può chiarire meglio il concetto:

Immaginiamo una pagina web con due immagini al suo interno. Il browser per prima cosa scaricherà il file HTML e lo interpreterà. Dalla lettura del codice capirà che deve scaricare altri 2 files (per esempio due JPG), inizierà il download, ma nel frattempo inizierà il rendering del testo. Il **rendering completo** (e l'eventuale riassetto del layout) avverrà nel momento in cui entrambi i JPG saranno nella cache.

*NB: normalmente le immagini e i multimedia sono gli elementi «pesanti» di una pagina; se le immagini sono delle grafiche necessarie al layout occorre fare attenzione a renderle il più «leggere» possibile per velocizzare l'impaginazione ed evitare fastidiosi effetti di riassetto/riallineamento delle varie parti della pagina.*

## Posizione del tag IMG e attributi principali

L'immagine **NON** è un elemento di blocco. Si allinea con il testo nel punto in cui viene inserito il tag e aumenta l'interlinea in base alla sua altezza in pixel.

Gli attributi principali del tag IMG sono i seguenti:

### **WIDTH / HEIGHT**

*Larghezza e altezza (in pixel). Non sono obbligatori ma è opportuno utilizzarli in modo che il browser disponga fin dall'inizio lo spazio sulla pagina evitando un riassetto durante il caricamento.*

### **ALT**

*E' una didascalia «non-scritta» dell'immagine; viene letta dai software per ipovedenti e, in alcuni browser attiva un tooltip.*

### **TITLE**

*E' l'attributo specifico per il tooltip, supportato da tutti i browser.*

### **BORDER**

*E' lo spessore del bordo in pixel. Se lo spessore è impostato a 0 non viene applicato alcun bordo all'immagine, mentre se viene omesso il bordo è di 1 pixel e, nel caso l'immagine sia in un tag <a> assume gli stili dei link.*

Ecco un esempio che li utilizza tutti:

```

```

## Immagini con collegamento ipertestuale

Per rendere un'immagine un collegamento ipertestuale è sufficiente nidificarla all'interno di un tag <a>:

```
<a href="http://www.iuav.it">  
    
</a>
```

Si noti l'attributo BORDER a zero per evitare un antiestetico bordo attorno all'immagine.

Oppure anche:

```
<a href="http://www.iuav.it">  
  Università Iuav di Venezia <br />  
    
</a>
```

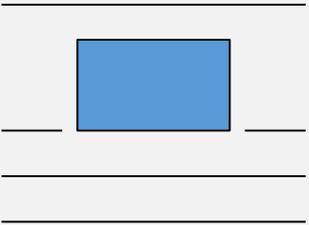
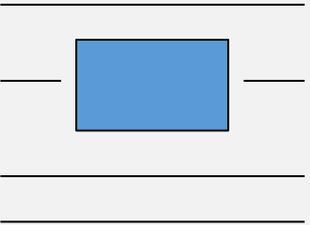
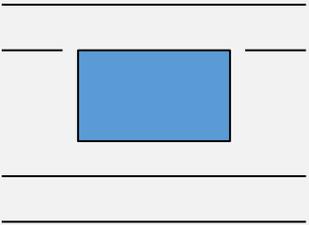
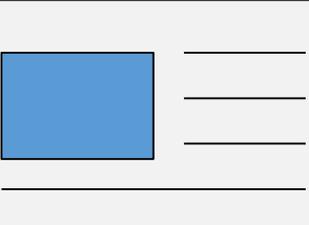
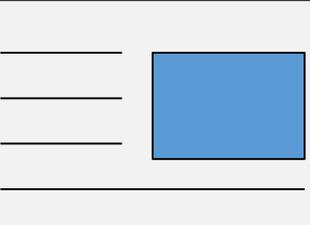
Dove abbiamo un link composito formato da testo (codice in rosso) e immagine a seguire (codice in blu), entrambi parte del collegamento ipertestuale.

# Allineamento delle immagini

Un attributo essenziale del tag IMG è **ALIGN**. ALIGN consente l'allineamento dell'immagine rispetto al flusso del testo. Di seguito sintassi e valori ammessi:

```

```

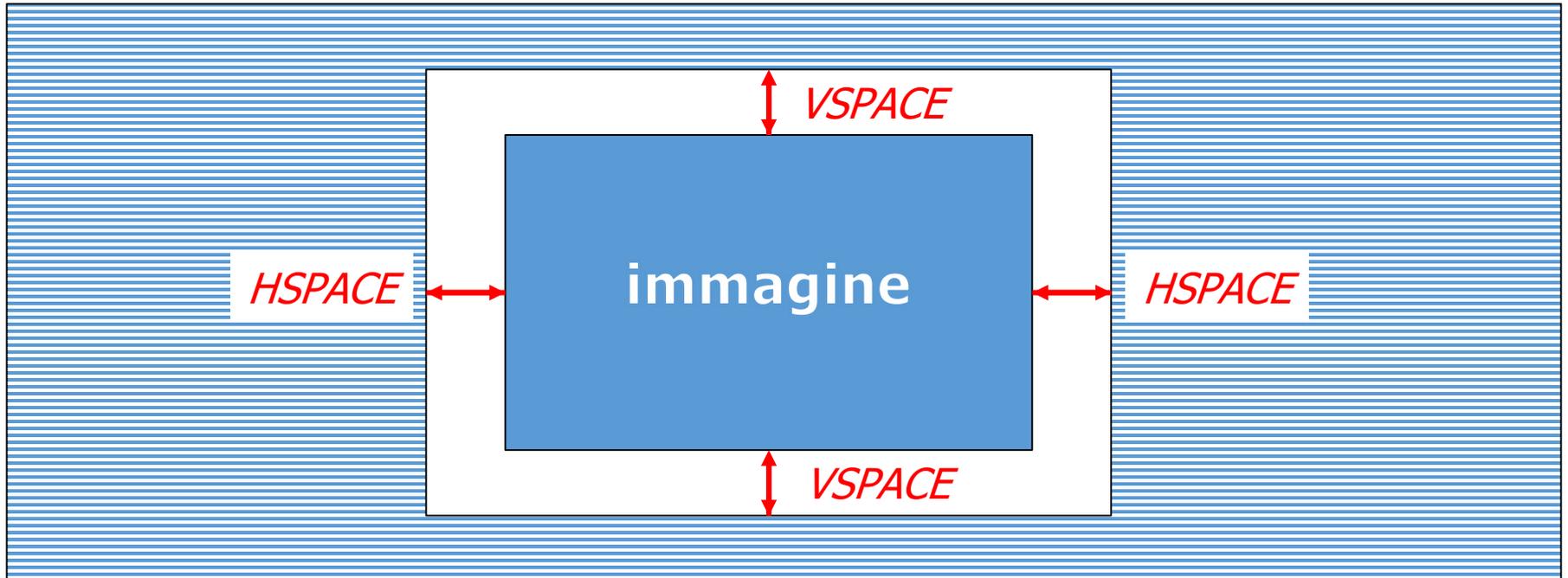
 <p><b><i>bottom</i></b> (default)</p>	 <p><b><i>middle</i></b></p>	 <p><b><i>top</i></b></p>
 <p><b><i>left</i></b></p>	 <p><b><i>right</i></b></p>	

## Spazi attorno alle immagini

Altri due pratici attributi del tag IMG sono **HSPACE** e **VSPACE** che governano gli spazi tra immagine e testo attorno ad essa nelle due direzioni.

```

```



# Le tabelle in HTML

Le tabelle HTML hanno avuto in passato un'importanza fin troppo elevata nella realizzazione dei siti web visto che sono state largamente **impiegate per realizzare i layout** delle pagine oltre che per organizzare i contenuti per righe e colonne.

Oggi **l'impaginazione mediante tabelle è fortemente sconsigliato e disapprovato dal W3C** che, prima con i fogli di stile, poi con le novità dell'HTML 5, ha definito modalità più efficaci e versatili per suddividere le pagine nelle diverse sezioni di contenuto.

I tag necessari per realizzare tabelle sono i seguenti:

**<table>** apre la tabella

**<tr>** «table row»: indica l'apertura di una riga

**<td>** «table data»: indica una cella all'interno di una riga

*(ovviamente i tre tag vanno chiusi con </table>, </tr> e </td>)*

La prima cosa che si nota è la mancanza di un tag per le colonne. Infatti, la natura sequenziale del codice HTML fa sì che la tabella venga definita come **serie di righe con  $n$  celle** e che le colonne vengano a generarsi in funzione di queste.

Una delle proprietà più interessanti delle tabelle HTML è quella di **auto-adattarsi sia al contenuto sia al contenitore** facilitando un utilizzo razionale dello spazio.

# Le tabelle in HTML

Ecco un esempio di sintassi per una tabella e relativo rendering:

```
<table border="1">  
  <tr>  
    <td>prima cella</td>  
    <td>seconda cella</td>  
  </tr>  
  <tr>  
    <td>terza cella</td>  
    <td>quarta cella</td>  
  </tr>  
</table>
```

prima cella	seconda cella
terza cella	quarta cella

Nell'esempio vediamo già il primo attributo di `<table>` ovvero **BORDER**, con cui è possibile specificare lo spessore dei bordi in pixel. La resa grafica di default è generalmente poco gradevole ma è possibile migliorarla sensibilmente con i fogli di stile.

## Dimensionamento delle celle di una tabella

Gli attributi **WIDTH** e **HEIGHT** consentono di specificare larghezza e altezza in pixel e possono essere utilizzati sia per `<table>`, sia per `<tr>` e `<td>`.

Oltre alle misure in pixel è possibile utilizzare dimensioni in percentuale con l'apposito carattere (es: 50%), nel qual caso le dimensioni si adattano in funzione del contenitore in cui è inserita la tabella.

```
<table width="300" border="1">
  <tr>
    <td>prima cella</td>
    <td>seconda cella</td>
  </tr>
  <tr>
    <td>terza cella</td>
    <td>quarta cella</td>
  </tr>
</table>
```

prima cella	seconda cella
terza cella	quarta cella

```
<table width="75%" border="1">
  <tr>
    <td width="25%">prima cella</td>
    <td width="75%">seconda cella</td>
  </tr>
  <tr>
    <td width="25%">terza cella</td>
    <td width="75%">quarta cella</td>
  </tr>
</table>
```

prima cella	seconda cella
terza cella	quarta cella

## Raggruppamento di celle in una tabella

Esistono due attributi per unire tra loro le celle di una tabella: si tratta di **COLSPAN** e **ROWSPAN**.

Si applicano entrambi solo al tag <td>. Il primo permette di specificare quante colonne occupa la cella, il secondo quante righe. L'utilizzo di questi attributi è tutt'altro che intuitivo per cui molto spesso ci si avvale di editor grafici. La sintassi negli esempi che seguono:

```
<table border="1">
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td colspan="2" align="center">
```

```
  cella unita
```

```
</td>
```

```
</tr>
```

```
</table>
```

	cella unita	

```
<table border="1">
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td rowspan="2" align="center">
```

```
  cella unita
```

```
</td>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
</table>
```

	cella unita	

*NB: le **celle vuote**, ovvero senza alcun carattere di contenuto vengono renderizzate senza bordo per cui è consuetudine inserire uno spazio con la sequenza **&nbsp;***

## Attributi del tag TABLE

Per quel che riguarda il tag `<table>`, vediamo i principali attributi, quelli che ci permettono di regolare le distanze tra i margini della tabella (o della cella) e il suo contenuto:

**border** (vedi sopra)

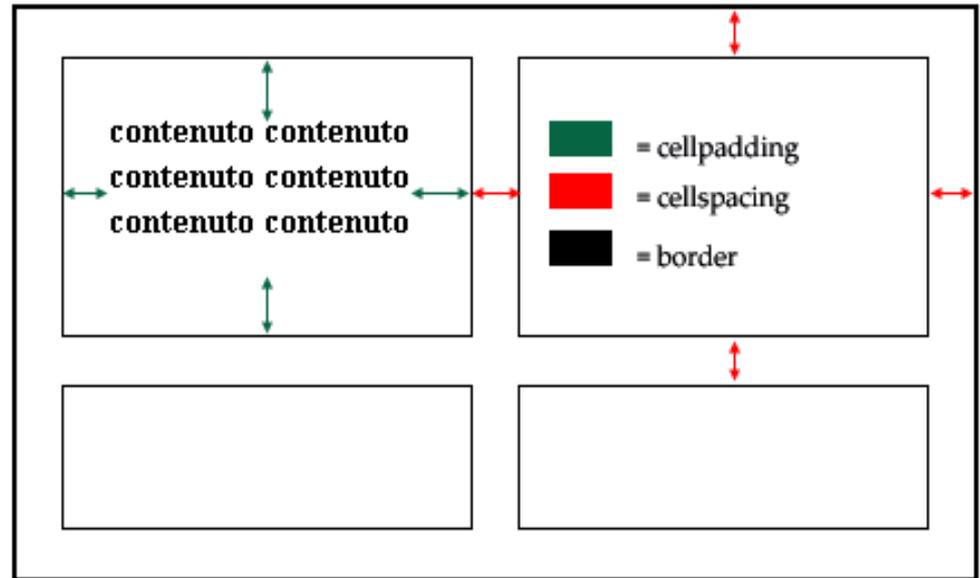
*specifica la larghezza dei bordi di una tabella (in pixel)*

**cellspacing**

*specifica la distanza (in pixel) tra una cella e l'altra, oppure tra una cella e il bordo. Di default è un pixel, dunque occorrerà sempre azzerarlo esplicitamente, quando non lo si desidera*

**cellpadding**

*indica la distanza tra il contenuto della cella e il bordo. Se il valore viene indicato con un numero intero, la distanza è espressa in pixel; il cellpadding tuttavia può anche essere espresso in percentuale. Di default la distanza è nulla.*

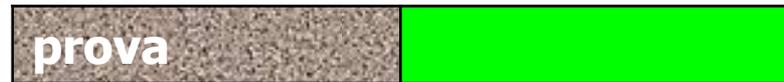


## Attributi validi per TABLE e per le celle

Oltre a WIDTH e HEIGHT di cui abbiamo già trattato, altri tre attributi importanti applicabili sia alla tabella sia alle celle sono **BGCOLOR**, **BACKGROUND** e **ALIGN**.

I primi due attributi hanno comportamento analogo a quanto visto per il tag body permettendo infatti di applicare un colore o una immagine allo sfondo della tabella o della singola cella.

```
<table width="300" border="1" align="center" bgcolor="#00FF00">
  <tr>
    <td width="50%" background="sfondo.gif">
      <font color="#FFFFFF">prova</font>
    </td>
    <td width="50%">
      &nbsp;
    </td>
  </tr>
</table>
```



Per quanto riguarda l'attributo **ALIGN** va specificato che il comportamento sulla tabella è diverso da quello sulle celle. Nel caso della tabella, l'allineamento riguarda la posizione della tabella rispetto alla pagina; nel caso delle celle riguarda invece la posizione del contenuto rispetto alla cella.

# **I moduli interattivi in HTML**

*Elementi per l'input di dati da inviare via Internet*

# 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**.

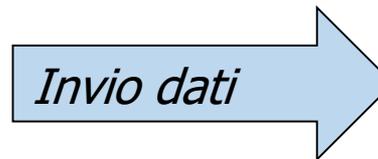
**Seminar Form** More Actions ▼

Name	<input type="text"/>	Date of Seminar	<input type="text"/> 31 [dd-MMM-yyyy]
Position	<input type="text"/>	Arrival Date	<input type="text"/> 31 [dd-MMM-yyyy]
Institution	<input type="text"/>	Departure Date	<input type="text"/> 31 [dd-MMM-yyyy]
Email	<input type="text"/>	Place of Stay	<input type="text"/>
Phone	<input type="text"/>		

Format of Presentation

Lecture  
 Seminar  
 Colloquium

**NAME**



## 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>
```

Vai al portale IUAV



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

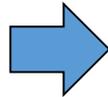
Scrivi un testo

Cancella testo



TESTO|

Cancella testo



Scrivi un testo

Cancella testo

## 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 <img>.

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" />
```

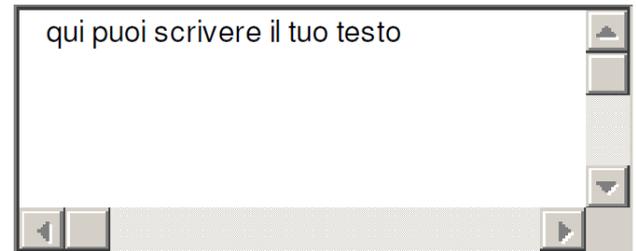
---

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.**



## 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
```

<input type="checkbox"/>	html
<input type="checkbox"/>	css
<input type="checkbox"/>	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 `<BR />`:

**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.

# **Esercizio n.2**

# Sviluppare una pagina di iscrizione ad esami universitari

La pagina deve essere composta da **due sezioni consecutive** secondo le indicazioni a seguire:

## Contenuti sezione 1:

- **Logo grafico** dell'ateneo con link all'home page dello stesso.
- **Immagine dell'ateneo cliccabile** con link verso una pagina web esistente.
- **Tabella dei corsi di laurea.** (layout a scelta)
- **Elenco puntato** con voci a scelta (es. istruzioni per la compilazione) e altro testo ritenuto necessario.
- Sfondi e colori a scelta.

## Contenuti sezione 2: un unico modulo che permetta:

- La selezione del docente e dell'esame da sostenere tramite **una casella combinata o altro** metodo.  
*(inserire alcune voci a scelta come esempio)*
- L'inserimento dei propri dati identificativi (nome cognome n. di matricola) in **apposite caselle di testo.**
- La conferma della prenotazione dell'esame con **un pulsante di invio.**
- L'azzeramento del modulo con **un pulsante di reset.**