

Michele  
Ferrara

# Manuale di HTML

fondamenti

impostazioni

multimedia

collegamenti  
ipertestuali

tabelle

elenchi

frame

form

tutti i TAG



con cd  
interattivo  
multimediale

  
**Canoro**

# **Guida di HTML**

**a cura di Michele Ferrara**

**edizioni La Casalinga**





## Premessa

Al crescere continuo degli utenti di internet sempre più persone si affacciano con interesse crescente al mondo del web, ossia a tutte quelle cose che ruotano attorno alla pubblicazione di proprie pagine in rete. Sempre più persone hanno la volontà, per lavoro o per hobby, di farsi conoscere mediante un proprio sito su Internet. La maggior parte delle persone impara a creare documenti Web solo quando è obbligata a farlo; magari è la richiesta del datore di lavoro di pubblicare un sito web dell'azienda, oppure la volontà di rendere disponibili alcune notizie o altro materiale alla comunità di internet e quindi si pone la questione ... "Forse è il caso che mi documenti". Nella vita di tutti i giorni questo è il modo più diffuso per ampliare per proprie conoscenze. Viene assegnato un lavoro che non si sa esattamente come eseguire, quindi è necessario imparare alla svelta.

Ecco il perché di questa guida, destinata a chi si avvicina per la prima volta a questo linguaggio di video-scrittura, ma che contiene diversi consigli e trucchetti per realizzare siti web degni di esperti.



# Indice generale

<b>Introduzione</b>	
Cosa è l'HTML . . . . .	7
<b>Nozioni generali</b>	
I TAG Fondamentali . . . . .	10
I META-TAG . . . . .	12
Impostazione della pagina . . . . .	14
<b>Inserimento del testo</b>	
Impostazione del testo . . . . .	18
Formattazione del testo . . . . .	21
<b>Elementi multimediali</b>	
Inserimento di immagini . . . . .	24
Suoni . . . . .	28
Animazioni . . . . .	32
<b>I collegamenti ipertestuali</b>	
Elementi fondamentali . . . . .	34
Approfondimenti . . . . .	37
Mappe immagine . . . . .	42
<b>Le tabelle</b>	
Creare una tabella . . . . .	46
Lavorare con le celle . . . . .	51
<b>Gli elenchi</b>	
Elenchi puntati . . . . .	55
Elenchi ordinati . . . . .	57
Elenchi di definizioni . . . . .	60
<b>I frame</b>	
Creare i frame . . . . .	62
Lavorare con i frame . . . . .	65
Gli Iframe . . . . .	67
<b>I FORM</b>	
I moduli e l'interfaccia CGI . . . . .	69
Raccolta delle informazioni . . . . .	71
Informazioni aggiuntive . . . . .	74
<b>Tutti i tag</b>	
Tutti i tag . . . . .	77
<b>Appendice</b>	
Tag, Attributi e Valori . . . . .	210
Le intestazioni . . . . .	212
Gli stili di testo . . . . .	213

Le immagini . . . . .	216
Le mappe immagine . . . . .	218
Le tabelle . . . . .	221
Istogrammi . . . . .	227
Informazioni generali sulle liste . . . . .	229
I frame . . . . .	230
I caratteri speciali . . . . .	232
I colori nel sistema esadecimale . . . . .	236
I colori più comuni . . . . .	238
Glossario . . . . .	241
Indice analitico . . . . .	245
Bibliografia. . . . .	247

## Introduzione

### Cosa è l'HTML

La storia del WWW comincia al Centro Europeo per le Ricerche Nucleari di Ginevra nel 1991. Nel Gennaio '92 era disponibile, su FTP anonimo, una prima versione di browser a linee. Poco dopo, Marc Andreessen, uno studente del National Center for Supercomputing Applications (NCSA) rilasciò la prima versione di un browser ipertestuale che gestiva, oltre ad http, anche ftp, gopher ed altri protocolli TCP/IP; stiamo parlando di Mosaic. Il protocollo HTTP introduce, di fatto, il nuovo concetto di hyperlink: diventa possibile, con un semplice click, attingere informazioni in maniera trasparente da nodi differenti, a partire dallo stesso documento.

Oltre a ciò, si sviluppano le tecniche per immagazzinare nel documento testi, immagini e suoni contemporaneamente. Un ulteriore passo avanti è l'introduzione degli helpers da parte di Netscape e, sempre da loro, l'introduzione recente dei plug-in.

Per la natura stessa di HTTP, cioè di essere indipendente dal tipo di dato che deve essere gestito, si è ritenuto di rendere uniforme (uniform) il modo di identificare i siti (locator) che sono la fonte delle risorse (resource) che HTTP deve poi gestire. La notazione è del tipo:

*protocollo://sito.internet/percorso\_dati*

per esempio, un URL (Uniform Resource Locator) HTTP può essere:

*http://www.canoro.org/guide/corso.html*

dove http è il protocollo, www.canoro.org il sito e guide/corso.html il percorso dati (in questo caso, un file).

Questo tipo di notazione ci consente di avere una sola struttura per tutti i tipi di protocollo gestibili dal World Wide Web. Come avremo modo di vedere più avanti, questa funzione, applicata ad un browser come Netscape Navigator o Internet Explorer, fa del WWW uno dei mezzi più potenti di comunicazione su rete di tipo Internet.

La sigla HTML è l'acronimo di HiperText Markup Language e tradotto letteralmente indica un "linguaggio a marcatori per l'ipertesto". In pratica è un linguaggio di scripting che contiene anche la formattazione del testo, ossia i codici che indicano come e dove deve essere scritta una parte di testo. L'HTML è un linguaggio sviluppato su piattaforme incrociate, e quindi funziona (e viene letto) su qualunque piattaforma, su qualunque tipo di computer e con qualunque sistema operativo.



L'HTML è più che sufficiente a realizzare un sito web, senza l'ausilio di altri linguaggi di scripting o di programmazione e altre tecniche di implementazione, come javascript, Java, CSS, PHP, ASP o il più recente DHTML, l'HTML dinamico; certamente mediante questi ulteriori strumenti si potrà ottenere una migliore disposizione degli oggetti all'interno della pagina e soprattutto una maggiore interattività tra la pagina e gli utenti, ma gli strumenti che l'HTML ci mette a disposizione costituiscono una discreta attrezzatura per realizzare anche un sito web coi fiocchi.

Ed è su questi strumenti basilari che si focalizzerà la nostra attenzione e su cui verteranno gli argomenti di questa guida. Cominceremo dalle nozioni fondamentali di una pagina HTML e dagli elementi essenziali che debbono comparire in ciascuna pagina, e continueremo cercando di inserire prima del testo poi delle immagini all'interno della pagina e poi di disporli nel modo in cui fa più piacere, in modo da curare anche la grafica del documento realizzato.

Un ipertesto è un documento che ha la possibilità di una lettura non lineare, ossia è possibile saltare arbitrariamente da un argomento all'altro selezionando alcuni collegamenti disposti all'interno delle pagine, e ha la possibilità di visualizzare immagini, suoni o animazioni, elementi multimediali insomma.

Questa possibilità ha favorito lo sviluppo del **WWW** (World Wide Web) ossia Internet a livello mondiale.

Prima di cominciare vediamo gli strumenti che ci servono e che sono assolutamente **necessari** per creare una pagina HTML. Ecco un elenco dettagliatissimo: • **Nessuno**.

Ebbene sì, per creare una pagina HTML non occorrono particolari strumenti chissà quanto sofisticati. Basta un normale editor di testo, va benissimo il Notepad di Windows o il Vi o Xedit di Unix/Linux.

Naturalmente esistono decine di editor specializzati e molti di essi sono anche WYSIWYG (What You See Is What You Get), cioè permettono di creare la pagina disponendo tutti gli elementi proprio come essi appariranno una volta avviato il browser, ma per adesso utilizzeremo un semplice editor di testo, perché meglio ci aiuterà a capire come funzionano le varie strutture dell'HTML. Col tempo può tornare utile usare un editor di testo più sofisticato come UltraEdit o EditPlus, che aiuti tramite la scelta di colori all'individuazione di eventuali errori di digitazione..

Infine i browser. Naturalmente occorre un programma che sia in grado di leggere il file HTML e di visualizzarlo sullo schermo come una pagina; questo per poter vedere il lavoro che abbiamo fatto. Anche di browser ne esistono decine; i più conosciuti sono **Internet Explorer** e **Netscape Navigator**, ed è a questi due che

faremo riferimento. Questi due browser però non sono uguali; sono stati messi sul mercato in tempi diversi e i progettisti hanno avuto indicazioni diverse su come implementare alcune funzionalità. Può capitare (raramente) che una funzionalità sia implementata su un browser ma non sull'altro oppure che per eseguire la stessa funzione sui due browser occorran due codici distinti; qualora dovesse capitare, metteremo entrambi i codici. Questa differenza è ancora maggiore per le ultime funzionalità dell'HTML, il javascript e il DHTML, ma di quelli tratteremo più avanti.

Nel prossimo capitolo inizieremo col vedere i TAG fondamentali, quelli che servono in tutte le pagine, e a fare in modo che un qualunque documento sia riconosciuto come un documento HTML.

# Nozioni generali

## I TAG fondamentali

I TAG sono i comandi che definiscono la struttura di una pagina HTML e che permettono una corretta visualizzazione da parte del browser; si riferiscono al titolo della pagina, al colore e alle impostazioni dei caratteri e dello sfondo, all'impostazione del testo, delle immagini e dei collegamenti ipertestuali e in pratica permettono di visualizzare e gestire tutto quello che si vuole far vedere o tenere nascosto nella pagina.

I tag sono racchiusi tra parentesi angolari ( "<" e ">" ) e possono essere digitati indifferentemente sia in minuscolo che in maiuscolo; la maggior parte dei tag ha anche un tag correlato che ne indica il termine e si scrive antepoendo al nome del tag il simbolo *slash* "/"; i due tag formano una coppia e si chiamano tag di apertura e chiusura. Ad esempio, i tag che indicano l'inizio e la fine di un paragrafo si indicano con **<P>** e **</P>**.

D'ora in avanti gli elementi di codice HTML saranno evidenziati in grassetto colorato, come **<P>** e **</P>**, ed in maiuscolo.

I tag fondamentali sono quelli che servono a far capire al browser che quello è un documento HTML, e che deve essere trattato nel modo appropriato. Un documento HTML è tutto ciò che è contenuto nei due tag **<HTML>** e **</HTML>**. Questi due tag sono generalmente il primo e l'ultimo di ogni documento HTML, e le prime versioni dei browser ignoravano qualunque cosa fosse all'esterno di questi tag, gli attuali browser, invece, sono molto più elastici, e provano ad interpretare qualunque documento come un documento HTML, per cui se si scrive qualcosa (del testo o anche dei comandi) all'esterno di questi tag, questo viene comunque riportato sullo schermo. Gli attuali browser sono così robusti che una pagina HTML viene visualizzata "correttamente" anche se i tag **<HTML>** e **</HTML>** mancano del tutto.

Un documento HTML si compone di due parti: una intestazione e un corpo, in inglese rispettivamente header e body. L'header è un contenitore di informazioni, e racchiude tutte le specifiche, i parametri e le funzioni che non sono utili direttamente all'utente finale, ossia il lettore, ma che vengono utilizzate dai browser, dai motori di ricerca e da tutti quei programmi in grado di leggere un file HTML, per poter meglio gestire il documento. Il body è la parte che effettivamente contiene tutte le cose che vengono visualizzate o che servono ad impostare la visualizzazione degli oggetti. Ancora una volta, i moderni browser (facciamo riferimento a Firefox, Netscape Navigator e Internet Explorer) si dimostrano abbastanza

robusti ed elastici, visualizzando la pagina anche se la parte di header manca del tutto e se mancano i tag che delimitano la parte del body. Noi scriveremo sempre correttamente un documento HTML, inserendo sempre tutti i tag al loro posto come grammatica insegna.

**L'intestazione** di un documento HTML è quella parte di codice compresa tra i due tag **<HEAD>** e **</HEAD>**, e che quindi contiene quei tag utili ai browser, ai server web e ai motori di ricerca.

Questa sezione contiene il titolo della pagina, e che è generalmente il primo tag dell'intestazione. Una pagina potrebbe avere per esempio per titolo "*Guida di HTML - I TAG fondamentali*" e ciò sarebbe visibile nella barra del titolo della finestra del browser (la riga azzurra in alto). I tag che specificano il titolo della pagina sono **<TITLE>** e **</TITLE>**. Nella sezione di header sono contenuti anche i META-TAG, che portano informazioni aggiuntive per il server web, e per i motori di ricerca che così possono classificare il documento senza caricarlo completamente in memoria. L'header è inoltre utilizzato per definire particolari funzioni di script che verranno utilizzate all'interno del documento per dare un tocco di interattività alla pagina HTML. Parleremo dei META-TAG nel prossimo capitolo.

**Il corpo** del documento contiene l'insieme del testo che deve venire visualizzato e i tag che fanno in modo da dare a quel testo una forma ben precisa, aggiungendo anche grafica, suono ed elementi multimediali. Il corpo del documento HTML si indica con i due tag **<BODY>** e **</BODY>** ed è soprattutto del contenuto di questa sezione, appunto quella più importante del documento, che si occuperà principalmente questa guida.

Prima di concludere il capitolo vediamo schematicamente i comandi e le nozioni introdotte. Un documento HTML è un insieme di righe che hanno l'aspetto mostrato qui sotto.

```
<HTML>
<HEAD>
<TITLE>Titolo del documento</TITLE>
...
</HEAD>

<BODY>
...
...
</BODY>
```

```
</HTML>
```

## Nozioni generali

### I META-TAG

I meta-tag sono pseudo-comandi che non hanno un'azione diretta nei confronti della pagina, ma hanno il fine di migliorare i rapporti con i programmi con cui la pagina si relaziona, come il browser, il server web, i motori di ricerca ...

I meta-tag sono identificati dal tag **<META ...>** e non hanno il tag correlato **</...>**; sono infatti dei veicoli di informazioni, e le informazioni sono portate da ciascun comando **<META ...>**; al posto dei puntini vanno inseriti gli attributi dei meta-tag, che possono essere **HTTP-EQUIV**, **NAME** e **CONTENT**. Non entreremo molto nel dettaglio per il momento, ma ci limiteremo ad introdurne alcuni e a spiegarne il significato.

**<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252">**

indica che il contenuto del documento è di tipo html e usa quel particolare set di caratteri.

**<META HTTP-EQUIV="Set-Cookie" CONTENT="ID=EiG4HtfyUIyf; path=/">**

specifica al server web in quale cartella disporre un eventuale cookie, quei file che vengono posti sul computer del visitatore e che servono a monitorare la navigazione e a memorizzare elementi.

**<META HTTP-EQUIV="Content-Language" CONTENT="It">**

indica che la pagina è in italiano (utile ai motori di ricerca).

**<META HTTP-EQUIV="Refresh" CONTENT="25; URL="http://www.canoro.org">**

è un comando molto utile per dare un certo dinamismo che permette di caricare automaticamente una ulteriore pagina dopo che questa pagina è stata caricata. Il numero 25 indica i secondi che devono passare prima di caricare la nuova pagina.

**URL="http://..."** specifica la nuova pagina da caricare. Utile per reindirizzare automaticamente il visitatore da una pagina ad un'altra.

**<META NAME="Generator" CONTENT="FrontPage">**

indica il programma utilizzato per creare il documento.

**<META NAME="Description" CONTENT="Questo sito parla di una guida di HTML">**

fornisce una descrizione del documento (utile ai motori di ricerca).

**<META NAME="Keywords" CONTENT="guida di html, html, imparare l'html ...">**

contiene un elenco di parole chiave utili ai motori di ricerca e ad eventuali utenti per cercare il documento su internet.

**<META NAME="Author" CONTENT="Michele Ferrara">**

specifica l'autore del documento.

**<META NAME="Reply-To" CONTENT="michefe@email.it">**

indica la email dell'autore, alla quale si vuole essere contattati per chiarimenti.

Naturalmente esistono moltissimi altri parametri da attribuire ai meta-tag, di più o meno vaga utilità. Per ora terminiamo qui l'elenco, riproponendoci di tornare sull'argomento qualora ne trovassimo la necessità.

Con le nozioni appena introdotte, un documento HTML può quindi apparire nella forma:

```
<HTML>
<HEAD>
<TITLE>Titolo del documento</TITLE>
<META HTTP-EQUIV="Content-
Type" CONTENT="text/html; charset=windows-1252">
<META NAME="Keywords" CONTENT="guida di html,
html, imparare l'html ...">
<META NAME="Author" CONTENT="Michele Ferrara">
...
</HEAD>

<BODY>
...
...
</BODY>
</HTML>
```

Con questo capitolo termina la trattazione riguardante l'intestazione di un documento HTML; nelle prossime sezioni ci addentreremo finalmente nel corpo di un documento HTML, cominciando col descrivere l'impostazione di una pagina.

## Nozioni generali

### Impostazione della pagina

Finalmente entriamo nel vivo del discorso; i comandi visti finora servivano a dare una impostazione al documento HTML e ad aiutare i vari programmi a trattare con lui ed i browser non riuscivano a mostrare nient'altro che una pagina bianca vuota, mentre gli argomenti che vedremo ora di seguito ci permetteranno invece di cominciare a visualizzare qualcosa sullo schermo.

Il tag che definisce la pagina, o meglio il corpo del documento HTML è **<BODY ... >** e **</BODY>**, dove al posto dei puntini ci possono essere dei parametri opzionali e che permettono di impostare la pagina; sarà di quei parametri, detti attributi, che si occuperà il presente capitolo.

Gli attributi presenti nella definizione del corpo della pagina sono quelli che definiscono la distanza del testo dai bordi della finestra, il colore e le proprietà dello sfondo della pagina, il tipo di testo utilizzato e il colore dei collegamenti ipertestuali; possono essere presenti dei comandi che servano ad attivare funzioni di script definite nell'intestazione del documento. Vediamo in dettaglio tutti questi parametri.

**<BODY BGCOLOR="#BBBBBB">** imposta lo il colore di fondo della pagina, a tinta unita, in questo caso grigio chiaro; il simbolo **#** indica che il colore è espresso in *esadecimale* come combinazione dei tre colori rosso, verde e blu, 2 numeri per colore. Avremmo anche potuto usare la dicitura **BGCOLOR="lightgray"** ma ogni browser ha le proprie impostazioni e specificando il colore direttamente col proprio valore numerico si ottiene il colore esatto per ciascun visualizzatore. Volendo usare uno sfondo blu si imposta **BGCOLOR="#0000FF"**.

Nel **GLOSSARIO** è data una definizione del sistema esadecimale e nell'**APPENDICE** è presente un'ampia trattazione sull'utilizzo dei colori mediante il sistema esadecimale.

**<BODY BACKGROUND="immagini/sfondo1.jpg">** imposta lo sfondo della pagina come una serie di immagini **"sfondo1.jpg"** messe una affianco all'altra. Come vediamo dalla sintassi del comando, l'immagine di sfondo è un file, residente eventualmente in un'altra cartella; essendo un file, che viene trasferito in rete insieme alla pagina, sarebbe bene scegliere un file piuttosto leggero, in modo che venga trasferito prima: una pagina caricata velocemente è apprezzata di più. La presenza del parametro **BACKGROUND** non annulla il parametro **BGCOLOR**, ma l'immagine viene posizionata sopra al fondo a tinta unita, quindi è

inutile che siano presenti entrambi nella definizione del body; in caso di rete particolarmente lenta o di immagini pesanti da caricarsi verrà mostrato lo sfondo colorato prima di visualizzare definitivamente l'immagine.

**<BODY BACKGROUND="immagini/sfondo1.jpg" BGPROPERTIES="fixed">**

imposta lo sfondo della pagina come una immagine particolare e lo fissa alla finestra e non lo fa scorrere insieme al testo su e giù per la pagina; con questo parametro sarà solo il testo a scorrere all'interno della finestra, sopra allo sfondo. Il parametro viene ignorato dal browser Netscape, e non esiste un comando equivalente per quel browser.

**<BODY TOPMARGIN="40">** indica che la pagina ha un margine di (ad esempio) 40 pixel dal bordo superiore della finestra a lei dedicata; è come se tutto il contenuto della pagina fosse spostato 40 punti più in basso. Usato da solo (senza il corrispettivo **BOTTOMMARGIN**) questo parametro pone una striscia vuota sia all'inizio che alla fine della pagina. Il parametro funziona solo con Internet Explorer e viene ignorato dal browser Netscape, su questo browser il parametro usato è **MARGINHEIGHT** (vedi più in basso).

**<BODY BOTTOMMARGIN="40">** indica che la pagina ha un margine di (ad esempio) 40 pixel dal bordo inferiore della finestra; è come se alla pagina seguisse una ulteriore striscia vuota alta 40 punti. Il parametro funziona solo con Internet Explorer e viene ignorato dal browser Netscape, su questo browser il parametro usato è **MARGINHEIGHT** (vedi più in basso).

**<BODY LEFTMARGIN="30">** indica che la pagina ha un margine di (ad esempio) 30 punti dal bordo sinistro della finestra; l'area effettivamente utilizzata per la pagina è minore, ma può tornare utile per dare una diversa impostazione grafica al proprio lavoro. Usato da solo (senza il corrispettivo **RIGHTMARGIN**) questo parametro pone una striscia vuota sia a sinistra che a destra della pagina. Il parametro funziona solo con Internet Explorer e viene ignorato dal browser Netscape, su questo browser il parametro usato è **MARGINWIDTH** (vedi più in basso).

**<BODY RIGHTMARGIN="30">** indica che la pagina ha un margine di (ad esempio) 30 punti dal bordo destro della finestra. Il parametro funziona solo con Internet Explorer e viene ignorato dal browser Netscape, su questo browser il parametro usato è **MARGINWIDTH** (vedi più in basso).

**<BODY MARGINHEIGHT="30" MARGINWIDTH="40">** si usano per impostare le semi-pagine dei frame (vedremo più in là cosa sono e come si usano) e indicano il margine della pagina dai bordi dei frame, rispettivamente bordo superiore/inferiore e bordo



sinistro/destro. Sul browser Netscape hanno lo stesso effetto su tutte le pagine, e non limitatamente ai frame, e pongono una striscia vuota rispettivamente sopra e sotto alla pagina e ad entrambi i lati della pagina.

**<BODY TEXT="#000080">** specifica il colore predefinito del testo all'interno della pagina (in questo esempio, blu scuro); se nella pagina sarà presente del testo privo di formattazione, ad esso sarà dato il colore definito.

**<BODY LINK="#0000FF">** specifica il colore predefinito dei collegamenti ipertestuali (vedremo in seguito cosa sono); generalmente sono di colore azzurro ("**#0000FF**") oppure personalizzati dall'utente mediante le impostazioni del browser.

**<BODY VLINK="#FF00FF">** specifica il colore predefinito dei collegamenti ipertestuali già visitati; generalmente sono di colore viola ("**#FF00FF**") oppure personalizzati dall'utente mediante le impostazioni del browser.

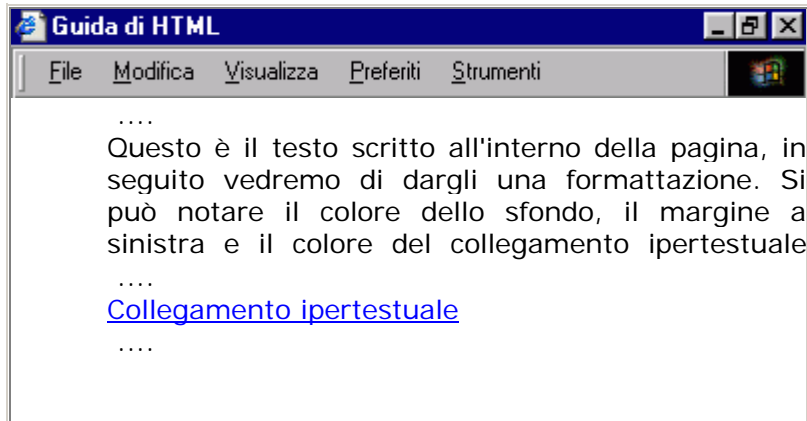
**<BODY ALINK="#FF0000">** specifica il colore predefinito dei collegamenti ipertestuali appena cliccati ma la cui pagina non è ancora stata caricata; generalmente sono di colore rosso ("**#FF0000**") oppure personalizzati dall'utente mediante le impostazioni del browser.

**<BODY ONLOAD=getimages();>** specifica (in questo caso) una funzione da implementare quando viene caricata la pagina. Di funzioni ne esistono tante quanto è vasta la fantasia umana e vengono chiamate mediante script o applicazioni CSS o javascript; vedremo in seguito cosa possono essere.

Con i comandi e i parametri ora introdotti non riusciremo a visualizzare un gran che sullo schermo, però già abbiamo dato alla pagina uno sfondo e abbiamo definito i principali parametri con i quali la pagina sarà visualizzata.

Con le nozioni appena introdotte, un documento HTML, e la pagina relativa, possono quindi apparire nella forma:

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#BBBBBB" VLINK="#0000FF" LEFTMARGIN="15">
....
Questo è il testo scritto all'interno della pagina, in seguito vedremo di dargli
una formattazione. Si può notare il colore dello sfondo, il margine a sinistra
e il collegamento ipertestuale
....
<formattazione> Collegamento ipertestuale
</BODY>
</HTML>
```



## Inserimento del testo

### Impostazione del testo

Creare una pagina web priva di significato e che non offre nulla di interessante è facile come imbrattare un muro con orrendi scarabocchi privi di forma. La situazione è la stessa in entrambi i casi: non c'è niente da fare se non sentirsi frustrati per essersi resi ridicoli. Il contenuto di un sito web è essenziale per una sua buona riuscita. Altrettanto essenziale è il modo in cui questo contenuto viene esposto e mostrato.

Un testo lungo e articolato, sebbene interessante, perde interesse se mostrato senza una degna struttura e impaginazione; l'uso del colore, di tabulazioni, di indentazioni, etc..., rende più leggibile anche il testo più complicato. È per questo che prima di definire quali caratteri usare, di che dimensione e colore descriviamo come impaginare e posizionare il testo all'interno di una pagina Web.

L'HTML dispone di TAG che consentono di costruire un layout (visualizzazione) abbastanza preciso del testo, permettendoci di disporlo come lo si vuole, utilizzando titoli, paragrafi, separatori e interlinee.

Il tag **<Hn>** definisce un titolo. Il simbolo n viene sostituito da un numero da **1** a **6** che indica la dimensione del titolo; **<H1>** è il più grande e **<H6>** il più piccolo. Il tag **<Hn>** ha l'attributo **ALIGN**, opzionale, che permette di dare l'allineamento al titolo; i valori possibili sono **"left"**, **"center"**, **"right"** e **"justify"**, ma quest'ultimo ha poco senso, parlando di un titolo. L'attributo predefinito è l'allineamento a sinistra, **"left"**, e può omettersi; la sintassi corretta, per un titolo di grandezza 2 allineato al centro, è **<H2 ALIGN="center">**.

Il tag **<DIV>** è utilizzato per raggruppare vari elementi isolati (testo, immagini e così via) in una singola unità, alla quale si vuole applicare la stessa formattazione (di solito l'allineamento); esso riconosce l'attributo **ALIGN**, con gli stessi valori del tag **<Hn>**; riconosce anche l'attributo **ID="nome"** utile nei fogli di stile CSS, insieme al tag **<STYLE>** che vedremo più avanti, per identificare l'oggetto all'interno della pagina.

Il tag **<P> ... <P>** indica un paragrafo (inizio e fine); ad un paragrafo viene sempre anteposta una interlinea di una riga e mezza e ad esso segue un'altra interlinea di una riga e mezza. Due paragrafi successivi appaiono separati da una sottile striscia vuota. Esattamente come per i tag precedenti riconosce l'attributo **ALIGN**, con gli stessi valori, il valore predefinito è **ALIGN="left"** e può omettersi. Il tag **<P>** può essere usato anche da solo

(senza il tag di chiusura `</P>`), purché si inizi il paragrafo successivo con un altro `<P>` altrimenti il browser suppone di stare ancora nel paragrafo corrente.

Il tag (o meglio il comando di markup) `<BR>` equivale esattamente a un ritorno a capo. Non contiene attributi degni di nota e non ha tag di chiusura. Esso manda a capo il testo interponendo una interlinea singola.

Il tag `<DIR>` inserisce una tabulazione. Le tabulazioni in HTML hanno un valore fisso (in genere una 80-ina di pixel a seconda del browser) e non possono essere modificate. Le tabulazioni possono essere inserite una dentro l'altra per ottenere una tabulazione doppia, tripla, etc... Vedere come esempio il titolo e l'argomento del presente capitolo, in alto nella pagina.

Il tag `<BLOCKQUOTE>` è identico al tag `<DIR>` e inserisce una tabulazione, dello stesso valore. Le tabulazioni possono essere inserite una dentro l'altra per ottenere una tabulazione doppia, tripla, etc...

Il tag `<HR>` inserisce una riga orizzontale (di solito con una ombreggiatura) di separazione tra due paragrafi consecutivi; non c'è il tag di chiusura e riconosce 4 attributi: **ALIGN** che definisce l'allineamento della riga con gli stessi valori dei tag precedenti (il valore predefinito è **ALIGN="center"**); **WIDTH** indica la lunghezza in pixel della linea, se questo numero è seguito dal simbolo **%** allora si tratta di una percentuale della larghezza della colonna; **SIZE** indica lo spessore in pixel della linea; **NOSHADE** (senza parametri) elimina l'ombreggiatura. Ad esempio, il comando `<HR SIZE="2" WIDTH="70%" NOSHADE>` visualizza la linea mostrata qui sotto.

Con i comandi e i parametri ora introdotti già possiamo visualizzare un testo; certo, sarà senza formattazione, però già gli possiamo dare una forma.

Vediamo di applicare con un piccolo esempio le nozioni appena introdotte.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
...
</HEAD>

<BODY>
<H2>Titolo della pagina</H2>
<P ALIGN="justify">Questo è un testo giustificato (notare l'allineamento
su entrambi i lati) scritto in un paragrafo. Quando la pagina termina la
frase viene troncata e la riga allineata. Notare che se anche si
```

va a capo nel testo HTML, nella pagina non si va a capo, perché il browser non trova alcun comando per farlo. Stesso discorso se si lasciano molti spazi tra una parola e l'altra, il browser li ignora e visualizza uno spazio solo.<BR>  
 Qui va a capo senza lasciare una interlinea. Qui sotto vediamo una linea allineata a destra, molto spessa.</P>  
 <HR ALIGN="right" SIZE="8" WIDTH="50%">  
 <DIR>Questo è invece un testo allineato a sinistra, a cui è stato posto una tabulazione; il testo risulta spostato più a destra.</DIR>  
 ....  
 </BODY>  
 </HTML>



## Inserimento del testo

### Formattazione del testo

Nelle prime versioni di HTML si faceva distinzione tra *formattazione logica* e *formattazione fisica*; gli stili logici descrivono le intenzioni dell'autore, mentre quelli fisici hanno più a che fare con un modo specifico di modificare l'aspetto del testo. Adesso questa sottigliezza ha perso notevolmente la sua importanza.

Per scrivere in **grassetto** si usa il tag contenitore **<B>** (oppure si può anche usare **<STRONG>**, è "lo stesso"); la lettera **B** è l'iniziale della parola bold, grassetto appunto.

Per scrivere in *corsivo* si usa il tag contenitore **<I>** (oppure si può anche usare **<EM>**); è l'iniziale della parola italic. Tag equivalenti, anche se nati con un significato logico diverso, sono **<CITE>** oppure **<ADDRESS>**.

Altro tag per scrivere in corsivo è **<DFN>** che però funziona solo con Internet Explorer e non con Netscape.

Per scrivere in sottolineato si usa il tag contenitore **<U>** e **</U>**; tutto il testo compreso tra i due tag verrà visualizzato come sottolineato. Alcuni editor di pagine web sono soliti utilizzare il tag **<U>** per evidenziare la sottolineatura nei collegamenti ipertestuali; tale sottolineatura è in genere già predisposta, ma può essere tolta con un piccolo artificio in CSS.

Nel capitolo precedente, **IMPOSTAZIONE DEL TESTO**, abbiamo visto come l'HTML ignora i tradizionali separatori, lo spazio e il ritorno a capo; un modo per restituire a questi separatori la loro solita funzione è mediante il tag **<PRE>**. Il testo viene scritto in carattere Courier, dimensione 10, e rappresentato con la stessa formattazione utilizzata durante la scrittura del codice HTML, compresi spazi, tabulazioni e ritorni a capo. A questo testo preformattato è possibile anche dare una formattazione diversa, utilizzando in combinazione anche gli altri comandi illustrati in questo capitolo.

Il tipo di carattere predefinito per il testo preformattato è il Courier, ma può dunque essere cambiato. Esistono altri modi per ottenere il carattere Courier, mediante i tag **<TT>**, **<CODE>**, **<KBD>** e **<SAMP>**. Questi tag sono tutti equivalenti e con l'avvento delle tabelle il loro uso è diventato raro.

È possibile aumentare o diminuire la dimensione dei caratteri da visualizzare; i tag **<BIG>** e l'altro **<SMALL>** rispettivamente consentono di aumentare e diminuire di un punto le dimensioni del carattere.

Infine il comando più completo e più pratico, ed in effetti quello che si utilizza di più per dare una formattazione al testo, il tag **<FONT>**. Scritto così non ha alcun effetto sul testo, ma questo tag accetta diversi attributi. L'attributo **FACE="..."** indica il tipo di carattere da utilizzare, se il computer su cui si sta guardando la pagina web non ha installati i tipi di carattere in questione prova ad utilizzarne uno simile o quelli che ha come predefiniti (in genere il **"Times New Roman"**). La dimensione del carattere può essere specificata con l'attributo **SIZE="n"**, in cui **n** rappresenta un valore assoluto (da **1** a **7**) oppure un valore relativo (da **-2** a **+4**) in relazione alla dimensione corrente del carattere. **SIZE="1"** indica il carattere più piccolo. Infine l'attributo **COLOR="..."** determina il colore del carattere; al posto dei puntini va il nome o il numero del colore (espresso come stringa esadecimale nelle combinazioni di rosso, verde e azzurro). Ad esempio il colore **rosa pallido** può essere indicato sia con **COLOR="RosyBrown"** che con **COLOR="#BC8F8F"**.

Vediamo con un esempio come si utilizzano i comandi ora introdotti, cercando di essere più chiari possibile; il testo all'interno dell'esempio aiuterà a capire le operazioni effettuate.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#FFEB CD">
<P ALIGN="center">
<FONT SIZE="5" COLOR="#006699"><B>Le fasi lunari</B>
</FONT></P>
<FONT FACE="Comic Sans MS">Mostriamo ora come con un testo
preformattato si possa ottenere una tabella. Lasciamo il <I>Courier</I>
come carattere, avremmo anche potuto cambiarlo.</FONT>
<PRE>
Mese      |   Giugno   |   Luglio   |   Agosto   |
Fase      |-----|-----|-----|
Nuova     |      5     |      4     |      3     |
Primo     |     13     |     12     |     11     |
Piena     |     20     |     20     |     18     |
Ultimo    |     27     |     26     |     25     |
</PRE>
<SAMP>Questo è invece un testo scritto con carattere monospazio
(<U>qui è sottolineato</U>), ma non è preformattato, infatti si può notare
come vada a capo in modo diverso nel documento e nella
pagina.</SAMP>
</BODY>
</HTML>
```

**Esempio**

**Guida di HTML**

File Modifica Visualizza Preferiti Strumenti

## Le fasi lunari

Mostriamo ora come con un testo preformattato si possa ottenere una tabella. Lasciamo il *Courier* come carattere, avremmo anche potuto cambiarlo.

Mese	Giugno	Luglio	Agosto
Nuova	5	4	3
Primo	13	12	11
Piena	20	20	18
Ultimo	27	26	25

Questo è invece un testo scritto con carattere monospazio (qui è sottolineato), ma non è preformattato, infatti si può notare come vada a capo in modo diverso nel documento e nella pagina.

....



## Elementi multimediali

### Inserimento di immagini

Molto probabilmente il WWW (World Wide Web, ossia il Web a livello mondiale) non avrebbe avuto lo sviluppo che lo ha visto protagonista in questi ultimi 15 anni se non ci fossero state le immagini, se non fosse stato possibile introdurre elementi di grafica nelle pagine che si presentavano. Le immagini, che possono essere fotografie, disegni o anche effetti colorati, contribuiscono a dare un tocco di vitalità alla pagina e, soprattutto, ad attirare maggiormente l'occhio del visitatore.

Esistono diversi formati di immagine, decine; l'HTML ne riconosce ufficialmente solo 2: GIF e JPEG. Un terzo formato, il PNG, potrebbe lentamente sostituire le GIF per ragioni che riguardano i diritti di autore, ma fino ad oggi è stato implementato solo in parte. Di solito è conveniente usare il formato JPEG per le fotografie, ricche di dettagli, e riservare il formato GIF per i disegni e i diagrammi a schema realizzati con appositi programmi di disegno.

I browser attuali sono molto robusti e permettono di visualizzare un numero elevato di formati grafici ma è bene non allontanarsi troppo dagli standard per questioni di compatibilità e di spazio occupato.

Oltre ai programmi di disegno e alla propria macchina fotografica digitale o il proprio scanner, le immagini possono essere reperite sul Web o nei siti FTP, prestando la dovuta attenzione all'utilizzo di materiale tutelato da copyright.

Il tag necessario per l'inserimento di una immagine è **<IMG>** che non ha tag di chiusura, una volta inserita l'immagine non ha senso dire al browser dove l'immagine finisce.

Il tag **<IMG>** scritto da solo non ha però alcun effetto sulla pagina; esso riconosce alcuni attributi, di cui uno è obbligatorio: occorre specificare qual'è l'immagine da inserire. L'attributo **SRC="..."** indica il percorso dell'immagine da caricare; può essere assoluto, e quindi essere del tipo

**SRC="http://www.sito.it/cartella/immagine.gif"**

oppure, se l'immagine si trova ospitata sullo stesso sito, può essere relativo, del tipo **SRC="cartella/immagine.gif"**; l'importante è specificare il percorso corretto.

Una precisazione importante riguardo ai nomi delle immagini e in generale di qualunque file. I server di tipo Unix/Linux fanno distinzione tra nomi MAIUSCOLI e minuscoli, è bene fare

attenzione sul tipo di carattere che si usa, e nel dubbio preferire l'uso del carattere minuscolo.

Quindi ci sono gli attributi opzionali, ma che servono se si vuole dare una impostazione diversa all'immagine introdotta. Se si carica la pagina web su un browser in cui lo scaricamento delle immagini è stato disattivato (per ragioni di velocità dei collegamenti, ma la pagina sembrerà più spoglia) è possibile non privare gli utenti di tutte le informazioni; l'attributo **ALT="..."** permette di inserire un testo alternativo al posto dell'immagine. Questo testo viene visualizzato quando si passa sopra l'immagine con il mouse, quando l'immagine manca oppure quando si utilizza un browser testuale.

Gli attributi **WIDTH="x"** e **HEIGHT="y"** determinano le dimensioni della immagine, in larghezza e in altezza. I numeri x e y possono essere sia numeri assoluti ed in tal caso indicano il numero di pixel che misura l'immagine, oppure una percentuale della dimensione della finestra. Se questi attributi sono assenti l'immagine viene mostrata alla sua grandezza naturale. Se è presente uno solo dei due attributi, l'altro si adegua in proporzione.

Su alcuni browser è implementato anche l'attributo **STYLE="..."** che racchiude questi due attributi in uno solo; la sua sintassi è: **STYLE="height:x; width:y"**.

L'attributo **BORDER="x"** indica lo spessore in pixel della cornice che si vuole mettere attorno all'immagine; il colore del bordo è nero; se è assente, l'immagine viene visualizzata senza bordo aggiuntivo.

L'attributo **ALIGN="..."** è qui più complesso che negli altri tag in cui compariva. Può avere i seguenti valori: **"left"**, **"right"**, **"bottom"**, **"top"** e **"middle"** (oppure **"absmiddle"**); ce ne sono altri, ma hanno lo stesso significato di questi ultimi tre. I valori **"left"** e **"right"** indicano che l'immagine è messa alla sinistra o alla destra di ciò che segue, se c'è un testo, e, se l'immagine fosse piuttosto grande, questo verrebbe disposto su più righe accanto all'immagine. I valori **"bottom"**, **"top"** e **"middle"** specificano che l'immagine viene visualizzata nella riga in cui è posizionata e il testo è allineato in una sola riga (anche se l'immagine è grande) al bordo inferiore o superiore o al centro dell'immagine. Se questo attributo è assente, il valore predefinito è **"bottom"**.

Infine gli attributi **HSPACE="x"** e **VSPACE="y"** indicano quanti pixel di spazio vuoto lasciare attorno alla immagine, rispettivamente a sinistra/destra e sopra/sotto, come una cornice vuota.

Con questo finiscono gli attributi del tag **<IMG>** e i modi in cui una immagine può essere posizionata con questo tag;

naturalmente esistono modi per visualizzare in maniera più complessa una immagine o gruppi di immagini, ma per fare questo si dovrà fare ricorso alle tabelle, oppure, per avere il massimo, ai fogli di stile CSS.

Vediamo di applicare con un piccolo esempio le nozioni appena introdotte.


```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>

<BODY BGCOLOR="#BBFFBB">
<P ALIGN="justify">
<IMG SRC="nonna.jpg" ALIGN="left" WIDTH="60">
Qui l'immagine è stata messa a sinistra e il testo alla sua destra (con
allineamento giustificato) su più righe.</P>
<P ALIGN="justify">Il testo continua al lato dell'immagine nonostante il
segno di paragrafo.</P>
Qui invece si è inserita <IMG SRC="haha.gif" ALIGN="absmiddle"> una
immagine dentro il testo, e siccome era piccola si è potuta inserirla
centrata con il testo, in modo da trovare una cosa accettabile
graficamente.</P>
<P ALIGN="justify">
<IMG SRC="manca.jpg" WIDTH="122" HEIGHT="50" ALT="Immagine
mancante, compare un testo"><BR>
Qui sopra l'immagine non viene trovata, oppure non viene mostrata, e al
suo posto compare il testo alternativo. Notare come il testo viene troncato
se non entra nello spazio. Con Netscape il testo compare solo quando si
passa col mouse sopra l'immagine.</P>
<P ALIGN="justify">
<IMG SRC="nonna.jpg" ALIGN="right" WIDTH="60" HEIGHT="60"
HSPACE="15" BORDER="3"> Qui l'immagine è stata messa a destra;
l'immagine appare distorta perché le abbiamo dato delle dimensioni non
proporzionate con le originali. Si noti lo spazio orizzontale lasciato ai lati
della immagine.</P>
....
</BODY>
</HTML>
```


**Esempio**

**Guida di HTML** [ \_ ] [ ] [ X ]

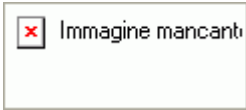
File Modifica Visualizza Preferiti Strumenti

 Qui l'immagine è stata messa a sinistra e il testo alla sua destra (con allineamento giustificato) su più righe.

Il testo continua al lato dell'immagine nonostante il segno di paragrafo.




Qui invece si è inserita una immagine dentro il testo, e siccome era piccola si è potuta inserirla centrata con il testo, in modo da trovare una cosa accettabile graficamente.



Qui sopra l'immagine non viene trovata, oppure non viene mostrata, e al suo posto compare il testo alternativo. Notare come il testo viene troncato se non entra nello spazio. Con Netscape il testo compare solo quando si passa col mouse sopra l'immagine.

Qui l'immagine è stata messa a destra; l'immagine appare schiacciata perché le abbiamo dato delle dimensioni non proporzionate con le originali. Si noti il bordo e lo spazio orizzontale lasciato ai lati della immagine.



....

Nei prossimi capitoli descriveremo come introdurre all'interno del documento suoni e animazioni.

## Elementi multimediali

### Inserimento del suono

Il suono e le animazioni multimediali non hanno riscontrato lo stesso sviluppo sul web delle immagini. Ciò è dovuto ad una causa semplicissima: la scarsa larghezza di banda dei collegamenti telefonici utilizzati per internet. Spesso infatti l'introduzione di questi elementi rallenta talmente la navigazione che il visitatore abbandona il sito prima ancora di averne appreso il contenuto.

La maggior parte dei file audio sono di grandi dimensioni e necessitano di molto tempo per essere caricati. Fanno eccezione i file **MIDI** che sono piccoli ma contengono solo comandi per sintetizzatori e non possono riprodurre la musica o i suoni naturali. Inoltre i browser sono raramente in grado di riprodurre i file audio che ricevono: hanno bisogno degli appositi programmi che lo fanno per loro, i plug-in; comunque le ultime versioni dei browser vengono distribuite già con tutti i plug-in più utilizzati.

Il modo più semplice per mettere una colonna sonora ad una pagina è mediante il tag **<BGSOUND>** che però non funziona con Netscape. La sua sintassi è molto semplice perché contiene solo 2 attributi: **SRC="..."** per indicare quale file musicale eseguire (allo stesso modo in cui **<IMG SRC="...">** indicava quale immagine visualizzare), e l'attributo **LOOP="n"** che indica quante volte eseguire il brano musicale. Il valore di **n** può essere **1**, **2**, ... oppure **0** o **infinite**; questi ultimi due valori indicano che il brano viene eseguito infinite volte.

La sintassi corretta del comando è dunque:

```
<BGSOUND SRC="prettywoman.mid" LOOP="2">
```

Per fare in modo che il suono si senta su tutti i browser si usa il tag **<EMBED>**. Gli attributi di questo comando sono: **SRC="..."**, **LOOP="..."**, **AUTOSTART="..."**, **VOLUME="..."**, **WIDTH="..."** e **HEIGHT="..."**.

In particolare:

**SRC="..."** e **LOOP="..."** hanno la stessa funzione che nel tag precedente;

**AUTOSTART="..."** indica se il brano deve partire automaticamente o è l'utente che deve farlo partire, e i valori ammessi sono **"true"** e **"false"**;

**VOLUME="..."** indica il volume iniziale del brano (l'utente può cambiarlo), e il valore ammesso può essere da **0** a **100**;

**WIDTH** e **HEIGHT** indicano le dimensioni che avrà la finestra di controllo, porre a 0 i valori di questi attributi significherà evitare che l'utente controlli il suono.

La sintassi corretta del comando è dunque:

```
<EMBED SRC="prettywoman.mid" AUTOSTART="true"
LOOP="2" VOLUME="70" WIDTH="144" HEIGHT="60">
```

Il tag **<EMBED>** visualizza sulla pagina una immagine composta con tutti i pulsanti per interagire con il file audio, quali i comandi di **PLAY**, **PAUSE**, **STOP** e il **VOLUME**.

Uno dei limiti all'inserimento dei suoni all'interno delle proprie pagine è che ogni volta che si cambia pagina il file musicale si interrompe bruscamente. E ciò è normale, visto che il file è legato a quella particolare pagina. Si può fare in modo che un suono rimanga anche durante la navigazione dividendo la pagina in frame. I frame sono appunto una divisione della pagina in più pagine web indipendenti e li studieremo più avanti in questa guida. Se si mette il suono in un frame fisso e si naviga in un altro frame il suono non si interromperà. E per i siti senza frame? C'è un piccolo espediente, basta fare un frame fisso invisibile (di dimensione zero) e mettere lì il file midi. Non entreremo molto nel dettaglio, in quanto i frame sono argomento dei prossimi capitoli, ma mostreremo l'effetto di questa partizionatura con qualche esempio.

```
-- finestra principale --
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>
<FRAMESET ROWS="30,*" BORDER=0>
  <FRAME NAME="alto" SRC="pagina_musica.html">
  <FRAME NAME="active" SRC="pagina_testo.html">
</FRAMESET>
</HTML>

-- pagina_musica.html --
<HTML>
<HEAD></HEAD>
<BODY bgcolor="#C0AAAA">
<EMBED SRC="prettywoman.mid" LOOP="2" AUTOSTART="true"
VOLUME="70">
Ciao Mondo !!
<BODY>
</HTML>
```

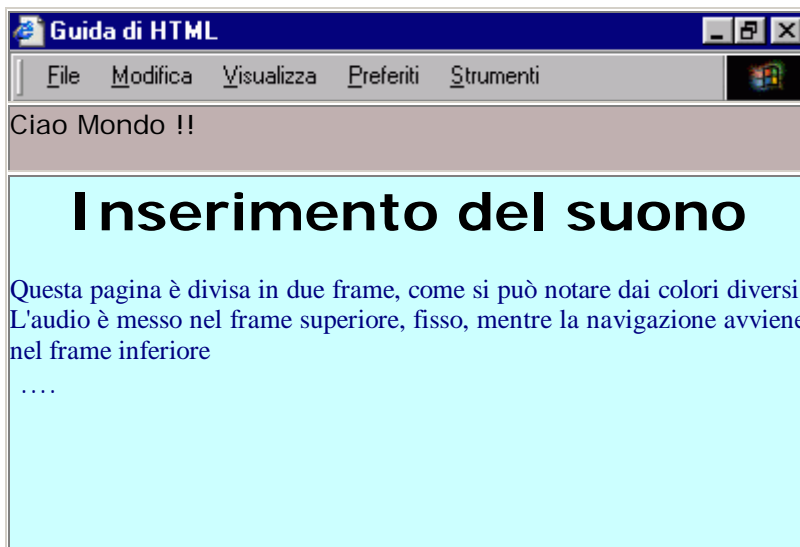
```

-- pagina_testo.html --
<HTML>
<HEAD></HEAD>
<BODY bgcolor="#CCFFFF">
<H2>Inserimento del suono</H2>
<P><FONT FACE="times new roman" COLOR="#000080">
Questa pagina è divisa in due frame, come si può notare dai colori
diversi. L'audio è messo nel frame superiore, fisso, mentre la navigazione
avviene nel frame inferiore.</FONT></P>
</BODY>
</HTML>

```

Notiamo che i frame interni non hanno intestazione, in quanto inutile, e che sono pagine HTML a tutti gli effetti. La pagina principale non ha il body, in quanto è divisa in due semi-pagine indipendenti.

Il disegno che segue mostra ciò che viene visualizzato sullo schermo:



Un ultimo modo per far ascoltare un file musicale è fare in modo che sia l'utente a richiederlo, e quindi ascoltare la musica in un'altra finestra. Per attuare ciò si ricorre ai collegamenti ipertestuali, che non abbiamo ancora introdotto ma che descriveremo a breve. Per ascoltare un file midi in un'altra finestra (**solo con Netscape**) si scrive il comando

**<A HREF="prettywoman.mid" TARGET="\_new">** e ciò che farà aprire un'altra finestra sarà proprio l'attributo **TARGET="..."**.

Ad esempio si può trovare nella pagina un testo come questo:

[Clicca qui](#) per ascoltare la colonna sonora di *Pretty Woman*.

Vedremo l'utilizzo del tag `<A HREF="..." ... >` più avanti in questa guida.

Con Internet Explorer la procedura funziona solo se si ha precedentemente associato il "Lettore Multimediale" di Windows ai file midi, in modo che il Lettore Multimediale parta automaticamente quando si è scaricato il file midi. Poiché non tutti avranno associato il Lettore Multimediale di Windows ai file midi, l'unico procedimento per fare in modo che il collegamento funzioni anche con Internet Explorer è associare il file midi ad un'altra pagina web mediante il comando `<EMBED ...>` in essa incluso.



## Elementi multimediali

### Video e animazioni

Se il suono non ha riscontrato il grandissimo successo ottenuto dalle immagini sul web, ancora meno sostanziale è la presenza di sequenze video su pagine web. E ciò è dovuto alla stessa identica causa, la scarsa larghezza di banda dei collegamenti telefonici utilizzati per internet. Un semplice filmato AVI di dieci secondi infatti occupa almeno 1 megabyte e per scaricare un file di tali dimensioni occorrono almeno 5 minuti, in condizioni normali (connessione a 56kbps e linea libera da sovraccarichi).

Solamente chi ha una connessione diretta e continuata può trasferire i file di animazione su internet; poiché non è possibile sapere se il visitatore ha questa connessione (in condizioni normali no) è consigliabile rendere lo scaricamento facoltativo anziché imporlo. Di buono c'è che i provider di servizi internet forniscono connessioni con larghezza di banda e velocità sempre maggiori da poter ormai permettere anche contenuti multimediali audio e video tramite internet e siti web.

I formati video principali sono tre: AVI, MPEG e QuickTime. Per ciascuno di questi il browser deve possedere un plug-in diverso, ma le ultime versioni dei browser vengono distribuite già con tutti i plug-in più utilizzati.

Esiste una soluzione più "economica" se ci si accontenta di pochi movimenti. La tecnica delle GIF ANIMATE deriva dai disegni animati e consiste nel riprodurre una serie di immagini (in genere non più di 20) una dopo l'altra, dando l'illusione del movimento. La GIF ANIMATA viene trattata come tutte le altre immagini, con la differenza che quando il file è stato caricato completamente il browser carica rapidamente le immagini una dopo l'altra.

Esistono molti programmi per creare GIF ANIMATE (come il Paint Shop Pro, il Gif Animator o il Gif Movie Gear) e sono piuttosto semplici da usare, è possibile impostare il numero di immagini da riprodurre e l'intervallo di tempo che deve trascorrere tra un fotogramma e l'altro; la maggior parte di questi programmi è shareware, ma è possibile trovarne anche di gratuiti.

Altro formato video è lo Shockwave Flash, realizzato da Macromedia. Shockwave è uno strumento per l'interpretazione di "scene" elaborate mediante l'editor Director di Macromedia appunto. Ovviamente anche questo formato necessita del suo plug-in, che si può scaricare gratis dal sito di Macromedia. Una volta che l'animazione è stata programmata, la si comprime con

una utility fornita anch'essa dal programma e si inserisce un collegamento dalla pagina web al file.

Dato l'alto prezzo dell'applicazione (circa 800 euro nel 2005) il suo uso è limitato soprattutto all'ambito commerciale. La Macromedia fornisce anche versioni *demo* del programma di animazione, limitando alcune funzioni ma lasciando all'utente la possibilità di scoprire tutte le potenzialità del programma.

Infine il VRML, che è un linguaggio per la modellazione della realtà virtuale. Il web designer definisce il suo mondo virtuale in modo simbolico utilizzando uno strumento chiamato Live3D; l'animazione diventa un file di testo dalle dimensioni ridotte che può essere interpretato nel browser dal plug-in WebFX. Il VRML è paragonabile ad una estensione tridimensionale di HTML e sarebbe necessario un libro grande 5 volte questa guida per imparare ad utilizzarlo.

# I collegamenti ipertestuali

## Elementi fondamentali

Abbiamo visto come inserire testo e immagini all'interno di un documento web e più avanti vedremo come dar loro una diversa impostazione grafica, come sistemarli meglio mediante le tabelle e gli elenchi.

Adesso finalmente esaminiamo la vera essenza del web, ciò che ne costituisce la struttura portante e che ha determinato il grande successo di Internet: il **collegamento ipertestuale**. È chiamato anche *link* e la sua funzione è di rimandare ad un'altra sezione della pagina stessa, oppure ad un'altra pagina presente sullo stesso sito o su un sito esterno. Esso permette inoltre di scaricare o eseguire programmi, immagini, file audio e video, presenti sul sito o su altri server, e tante altre cose.

L'uso del collegamento ipertestuale consente inoltre una lettura non sequenziale del documento, secondo la volontà del visitatore, che può saltare facilmente da un documento all'altro o da una sezione all'altra.

Il tag utilizzato per introdurre un collegamento ipertestuale è **<A>**, che però scritto così non ha alcun effetto e necessita di alcuni attributi. Gli attributi possibili sono 6 e servono a definire la destinazione del collegamento, e il modo in cui questo collegamento viene espletato. Gli attributi sono: **HREF**, **NAME**, **TITLE**, **TARGET**, **REL** e **REV**.

L'attributo che permette di creare un collegamento ipertestuale verso un'altra sezione o un altro documento è **HREF**, che definisce appunto il collegamento ipertestuale (HiperText REference), ma vediamo prima un altro attributo, in modo che poi potremo spiegare l'attributo **HREF** con maggior completezza.

L'attributo **NAME="..."** permette di inserire un segnalibro all'interno di una pagina, in modo da potercisi riferire tramite un collegamento situato in un'altra sezione della pagina o in un'altra pagina. È come mettere una bandieruola all'interno della pagina ed utilizzarla come riferimento per la navigazione. Usare un segnalibro all'interno di un collegamento ipertestuale vuol dire recarsi non all'inizio del documento da ad un punto preciso della pagina, esattamente lì dove è posizionato il segnalibro.

La sintassi corretta del tag in questo caso è: **<A NAME="nome">**[testo opzionale]**</A>**.

Siccome questo tag serve esclusivamente per porre un segnalibro all'interno del documento il testo può anche non essere scritto. Il testo **"nome"** invece è obbligatorio e deve essere univoco

all'interno del documento altrimenti il computer dovrebbe scegliere tra due elementi uguali. I segnalibri vengono molto utilizzati per creare glossari, FAQ (Frequently Asked Questions) ed elenchi in genere.

Veniamo adesso all'attributo che costituisce il collegamento ipertestuale vero e proprio, e che permette di collegare un documento ad un'altro.

L'attributo di cui stiamo parlando è **HREF** e la sua sintassi corretta è **<A HREF="indirizzo">[Testo HTML]</A>**.

La dicitura "testo html specifica un testo o anche un qualsiasi contenuto di tipo HTML che indica l'ipertesto all'interno della pagina; può essere un testo, una immagine o un qualunque altro elemento HTML. Nel caso sia un testo questo viene evidenziato con una sottolineatura e in colore azzurro (ma ciò si può anche modificare) mentre nel caso sia una immagine questa viene in genere evidenziata con un bordo colorato (che si può anche togliere).

La dicitura "indirizzo" indica l'indirizzo assoluto o relativo di un qualsiasi documento o file.

Si parla di indirizzo assoluto se esso è nella forma **"http://www.canoro.org/guide/html/corso.html"**, perché viene specificato l'indirizzo del server web su cui è posto il documento.

L'indirizzo è invece relativo se si considera a partire dalla posizione attuale del documento, ad esempio **"../sport/schede.html"**. Se il documento sta nella cartella corrente (quella in cui sta la pagina che stiamo visualizzando) allora si può scrivere solo il nome del file, ad esempio **"html-09.html"**.

Infine ci si può riferire ad un particolare punto di una pagina web, e non solo all'inizio del documento. Ciò si può fare mediante i segnalibri introdotti appena sopra. Il nome del segnalibro si indica alla fine dell'indirizzo, preceduto dal simbolo **#**, ad esempio, se avessimo messo un segnalibro alla fine di questa pagina e lo avessimo chiamato "fondo" si può fare un collegamento ad esso scrivendo **"#fondo"** come indirizzo.

Mediante i segnalibri vale la stessa distinzione tra indirizzi relativi e assoluti; ad esempio le tre scritte qui sotto sono equivalenti:

**<A HREF="http://www.canoro.org/guide/html-10.html#fondo">Clicca</A>** (indirizzo assoluto)

**<A HREF="html-10.html#fondo">Clicca</A>** (indirizzo relativo)

**<A HREF="#fondo">Clicca</A>** (indirizzo relativo)

e producono lo stesso effetto di indirizzare l'utente alla fine della pagina corrente, se questa si chiama "html-10.html".

Naturalmente in fondo alla pagina avremo messo il segnalibro: **<A NAME="fondo"></A>**.

Infine si può usare un collegamento ipertestuale per mostrare una immagine, oppure per ascoltare o scaricare una sequenza audio/video oppure per scaricare qualunque file.

Ecco alcuni esempi; nel primo descriviamo come legare una immagine ad un collegamento ipertestuale (si clicca e si vede l'immagine), nel secondo si consente all'utente di scaricare un piccolo file.

1° esempio:

**<A HREF="napoli.jpg"><IMG SRC="mini-napoli.jpg"></A>**



Questa riga produce l'effetto di visualizzare una immagine e di legare ad essa un collegamento ipertestuale per visualizzare la stessa immagine ingrandita; per far aprire l'immagine in un'altra finestra si utilizza l'attributo **TARGET** che sarà spiegato nel prossimo capitolo, senza quell'attributo l'immagine sarebbe visualizzata nella finestra corrente, sostituendola e facendo sparire il documento di partenza.

2° esempio:

**<A HREF="battle.exe">CLICCA QUI</A>** per scaricare il gioco....

Questa riga produce l'effetto di visualizzare alcune parole a cui è associato un link che porta ad un file eseguibile, un gioco di Battaglia Navale per PC. All'utente basterà fare click e, a meno che non ci siano impostazioni di protezione contro i file eseguibili da parte del suo browser, potrà scaricare immediatamente il file sul proprio pc o avviarlo direttamente. Il collegamento si adatta a qualunque tipo di file, eseguibile, compresso, immagine, testo, etc..

Quando si mettono collegamenti per scaricare file di grandi dimensioni è sempre una buona usanza segnalare all'utente la dimensione dei file.

## I collegamenti ipertestuali

### Approfondimenti

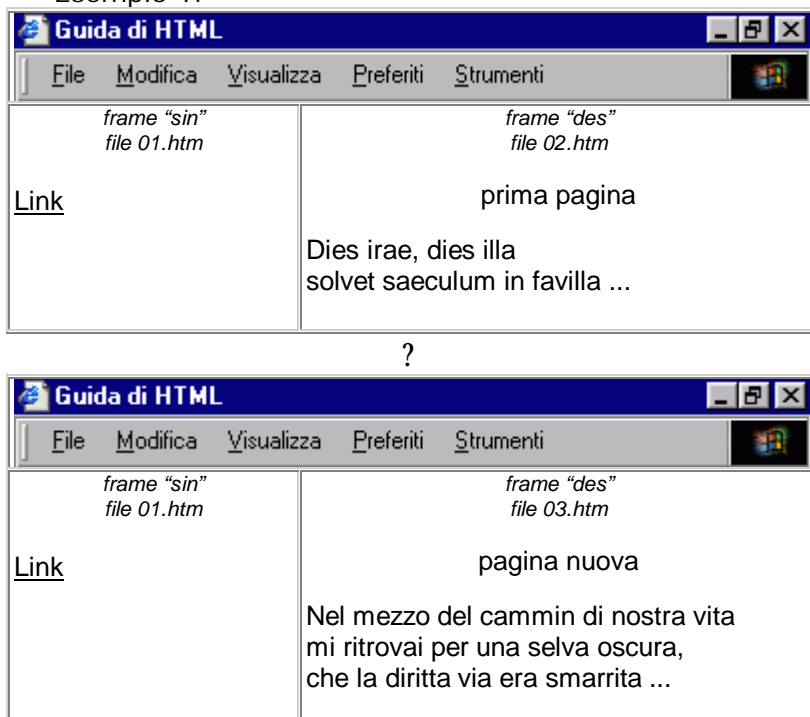
Gli elementi introdotti nel capitolo precedente permettono già di realizzare ed utilizzare un collegamento ipertestuale tra una pagina ed un qualunque altro file; adesso esaminiamo alcuni accorgimenti che possono aiutare ad utilizzare al meglio tutti gli attributi applicabili al tag **<A>**.

L'attributo **TARGET="..."** all'interno di un collegamento ipertestuale permette di aprire la nuova pagina in una finestra specifica che vogliamo noi, e non soltanto nella finestra attiva, ossia quella in cui è presente il collegamento ipertestuale. Questo attributo è utilizzato soprattutto con i **frame**, che sono delle divisioni della finestra del browser in più finestre e che permettono la visualizzazione contemporanea di più pagine web e soprattutto una navigazione più snella. Ebbene, dando un nome a ciascun frame è possibile indirizzare ciascun collegamento ipertestuale in un frame piuttosto che in un altro; è possibile, inoltre, far aprire la nuova pagina in una altra finestra, e questo è utile se si vuole che il visitatore navighi nella nuova pagina pur continuando a visualizzare la pagina di partenza. I frame, e il loro utilizzo, saranno descritti più dettagliatamente più avanti in questa guida.

I valori che può assumere l'attributo **TARGET** sono **"\_blank"**, per indicare che la nuova pagina verrà visualizzata in una nuova finestra; **"\_parent"**, che, in una divisione in più frame, indica la pagina che contiene il frame attivo (se una pagina è divisa in frame, questo attributo cancella la divisione e la nuova pagina sarà visualizzata nella pagina intera); **"\_top"**, che cancella la divisione in frame e apre la nuova pagina a finestra intera; inoltre questo attributo può assumere i valori **"nome"**, in cui evidentemente il testo **"nome"** indica il nome che verrà di volta in volta dato a ciascun frame della divisione.

L'utilizzo di questo attributo può risultare più chiaro mediante qualche esempio; come sarà spiegato più avanti, non tutti i browser supportano i frame, solo quelli più recenti (dal 1998 in poi) lo fanno, per cui sarebbe opportuno fornire 2 serie di esempi, uno mediante frame, uno senza, ma siccome l'attributo **TARGET** avrebbe poco senso senza la nozione di frame, riportiamo gli esempi solo per i browser che supportano i frame.

## Esempio 1:



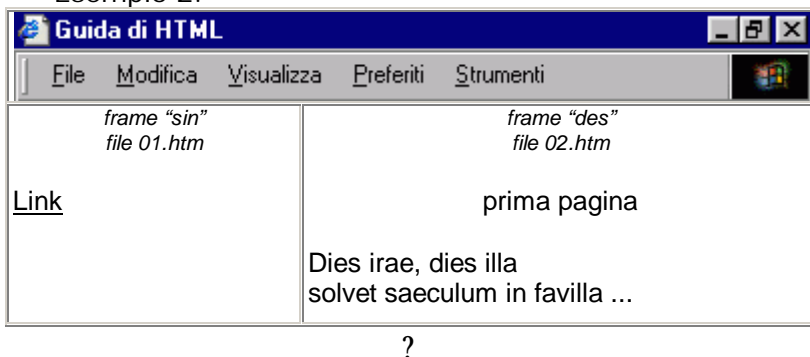
L'esempio qui sopra mostra una pagina divisa in due frame, chiamati "sin" e "des", con un link nel frame sinistro che apre una nuova pagina nel frame di destra.

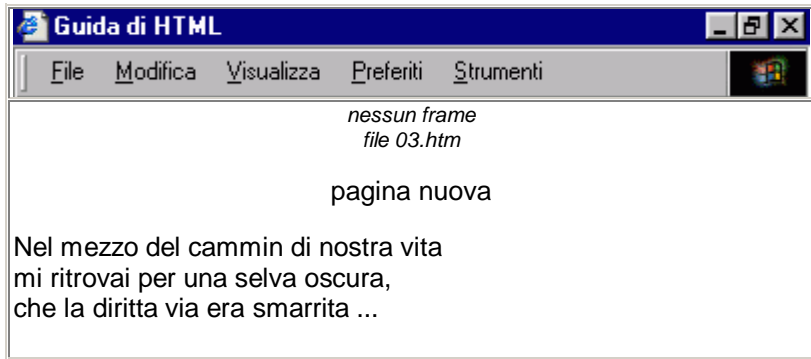
Al "Link" sarà applicato un collegamento ipertestuale del tipo:

**<A HREF="03.htm" TARGET="des">Link</A>** e ciò permette al browser di aprire la nuova pagina nel frame indicato.

Cliccando sulla parola Link l'utente vedrà caricarsi una nuova pagina nel frame di destra.

## Esempio 2:





L'esempio qui sopra mostra una pagina inizialmente divisa in due frame, chiamati "sin" e "des", con un link nel frame sinistro che apre una nuova pagina nella finestra intera.

Al "Link" sarà applicato un collegamento ipertestuale del tipo:

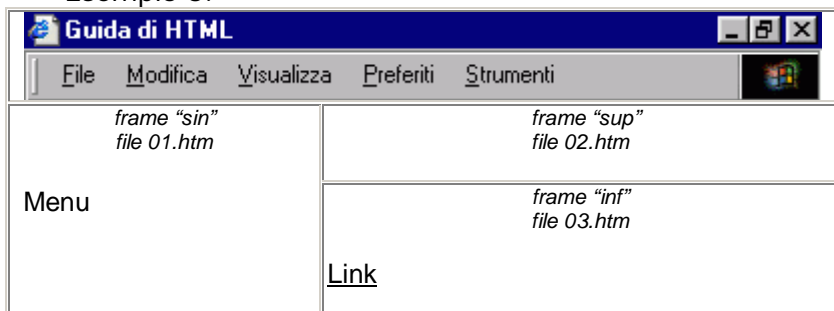
**<A HREF="03.htm" TARGET="\_parent">Link</A>** oppure  
**<A HREF="03.htm" TARGET="\_top">Link</A>** e ciò

permette al browser di aprire la nuova pagina nella finestra principale.

Normalmente **\_parent** e **\_top** non sono equivalenti, ma in questo caso il frame direttamente superiore al frame "sin" è proprio la finestra principale, per cui il comportamento dei due attributi coincide.

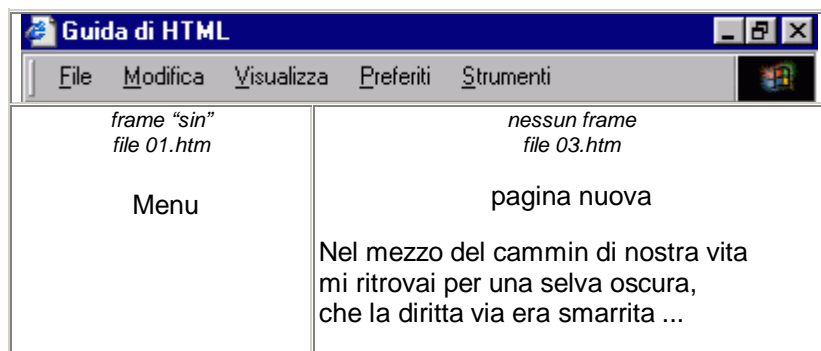
Cliccando sulla parola Link l'utente vedrà caricarsi una nuova pagina nel frame di destra.

Esempio 3:



?





L'esempio qui sopra mostra una pagina inizialmente divisa in due frame, chiamati "sin" e "des", ed il "des" suddiviso ulteriormente in un frame superiore "sup" ed uno inferiore "inf". È presente un collegamento nel frame inferiore che apre una nuova pagina nel frame che contiene la suddivisione; verrà mantenuta la divisione principale mentre la secondaria sarà annullata.

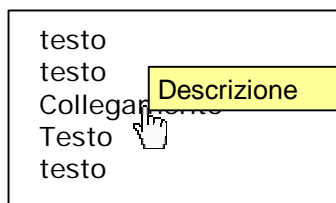
Al "Link" sarà applicato un collegamento ipertestuale del tipo:

**<A HREF="04.htm" TARGET="\_parent">Link</A>**.

Se fosse stato applicato un collegamento ipertestuale del tipo:

**<A HREF="04.htm" TARGET="\_top">Link</A>** la nuova pagina sarebbe stata aperta a finestra intera.

L'attributo **TITLE="..descrizione.."** non è di importanza fondamentale e serve solo per dare informazioni aggiuntive utili per il visitatore riguardo al documento che si va a visualizzare cliccando sul collegamento ipertestuale a cui è legato. Quando è presente questo attributo, passando con il mouse sopra al collegamento ipertestuale compare una strisciolina a sfondo giallo che riporta il testo scritto nella dicitura **"..descrizione.."**, un po' come l'attributo **ALT="..."** del tag **<IMG>**.



L'esempio a lato mostra il funzionamento di questo attributo, la cui sintassi corretta è:

**<A HREF="..url.." TITLE="Descrizione">**; al visitatore basta passare il mouse sopra il collegamento per vedere visualizzata la descrizione.

Infine gli attributi **REL** e **REV** indicano una relazione diretta o inversa tra i documenti sorgente e destinazione del collegamento ipertestuale; non sono molto utilizzati nella costruzione di pagine web e la maggior parte dei browser li ignora, per cui non ci addentreremo ulteriormente nella descrizione.

Ci basti sapere che la loro sintassi è la seguente:

```
<A HREF="indirizzo" REL="tipo1" REV="tipo2">TestoHTML  
</A>.
```

Allora l'attributo **REL** serve per specificare, tramite la stringa "tipo1", una relazione diretta tra il documento in cui risiede il tag e quello specificato nell'indirizzo; al contrario **REV** serve per specificare la stessa relazione, ma inversa tramite la stringa "tipo2".

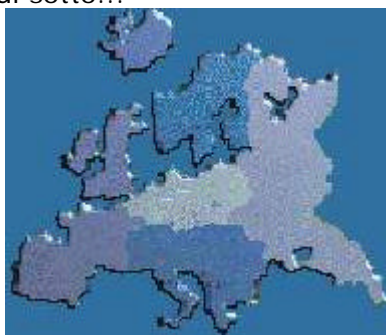
## I collegamenti ipertestuali

### Mappe immagine

Abbiamo visto come è possibile associare un collegamento ipertestuale ad una immagine; facendo click con il mouse sopra l'immagine si accede ad un'altra pagina web o un altro file. Mediante le mappe immagine è possibile utilizzare solo una porzione dell'immagine per il collegamento ipertestuale, destinando un'altra porzione dell'immagine ad un altro collegamento ipertestuale. Il principio è molto semplice, si suddivide una immagine in aree facilmente identificabili, e per ogni area si crea un collegamento. In tal modo, invece di fare click su un collegamento di testo o su una immagine si fa click su una porzione di immagine.

Non tutte le immagini sono adatte alla creazione di mappe immagine, devono essere immagini suddivisibili in zone di forme facilmente identificabili, cerchi, rettangoli o poligoni.

L'immagine di Nonna Papera riprodotta a lato non si presta bene ad essere utilizzata come mappa, perché non sapremo dividerla in parti distinte. Dovremmo cercare una immagine già divisa in pezzi distinti o comunque facilmente divisibile. Una immagine migliore può essere quella dell'Europa nel gioco del risiko, riprodotta qui sotto...



Per definire che una immagine è usata come mappa si utilizza l'attributo **USEMAP="..."** nel tag **<IMG>** che introduce l'immagine. Questo attributo assume come valore il nome che noi scegliamo di dare alla mappa, che è diverso dal nome dell'immagine. Supponiamo di aver scelto "maprisiko" come nome per la mappa dell'Europa disegnata sopra, la sintassi del tag diventa:

```
<IMG SRC="europe.jpg" BORDER="0"
USEMAP="#mapprisiko">
```

ove il simbolo del cancelletto "#" sta ad indicare che la mappa è specificata in questo file, esattamente come l'uso dei segnalibri.

Ma ciò non basta; occorre definire la mappa. Per definire una mappa si utilizza il tag **<MAP>** e **</MAP>** che ha un solo attributo, il nome della mappa, **NAME="nome"**, senza il simbolo del cancelletto "#". La sintassi completa è allora **<MAP NAME="mapprisiko">**.

Tra il tag di apertura e quello di chiusura è specificata la mappa, indicando una ad una le aree sensibili, specificandone la forma, la posizione e il collegamento associato.

Il tag da utilizzare è **<AREA>**, che non ha tag di chiusura. Il tag **<AREA>** accetta gli attributi **SHAPE**, **COORDS**, **HREF** e **TITLE**.

**SHAPE="..."** specifica la forma dell'area, e può essere **"circle"**, **"rect"** oppure **"poly"**, per indicare un cerchio, un rettangolo oppure in poligono.

**COORDS="..."** specifica le coordinate dell'area; per un cerchio si danno 3 numeri, separati da una virgola: la posizione del centro e il raggio; per un rettangolo si danno le coordinate delle estremità di una diagonale; per un poligono si danno le coordinate dei vertici in senso orario o antiorario.

**HREF="..."** specifica il collegamento ipertestuale a cui si riferisce quell'area, esattamente come un normale collegamento ipertestuale; l'indirizzo può essere assoluto o relativo e puntare a qualunque tipo di file.

**TITLE="..."** indica la descrizione che viene data all'area quando il puntatore del mouse ci passa sopra e serve a dare al visitatore informazioni riguardo al collegamento o all'area in questione.

Vediamo di applicare con un piccolo esempio le nozioni appena introdotte. Creiamo una mappa immagine in cui associamo a ciascuna area un collegamento ipertestuale di diverso tipo, in modo da far vedere come viene impiegato e i suoi effetti. Apriremo tutte le nuove pagine in una nuova finestra (mediante l'attributo **TARGET="\_blank"**) in modo che la finestra originaria del browser resti sempre visualizzata.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
...
</HEAD>

<BODY BGCOLOR="#EEDDCC">
<H2 ALIGN="center">Mappe immagine</H2>
```

```

<P ALIGN="justify">Clicca su una regione dell'immagine per aprire una
nuova pagina.</P>
<MAP NAME="europa">
<AREA SHAPE="circle" COORDS="66,15,19" HREF="corso.html"
TITLE="L'indice di questa guida" TARGET="_blank">
<AREA SHAPE="poly" COORDS="52,50,49,59,36,59..."
HREF="http://www.fiatpanda.org" TITLE="Il sito su Fabrizio De André"
TARGET="_blank">
<AREA SHAPE="rect" COORDS="4,101,57,147"
HREF="mailto:michefe@libero.it" TITLE="Mandami una email">
<AREA SHAPE="poly" COORDS="62,131,72,131,86,147..."
HREF="immagini/napoli.jpg" TITLE="Visualizza la foto di Napoli"
TARGET="_blank">
<AREA SHAPE="rect" COORDS="68,80,118,106" HREF="#top"
TITLE="Vai all'inizio di questo documento" TARGET="_blank">
<AREA SHAPE="poly" COORDS="73,40,77,74,91,66..."
HREF="battle.exe" TITLE="Scarica il gioco della battaglia navale"
TARGET="_blank">
<AREA SHAPE="poly" COORDS="129,25,120,40,123,72..."
HREF="http://www.canoro.org/download/winrisk.exe" TITLE="Scarica il
gioco del risiko" TARGET="_blank">
</MAP>
<CENTER>
<IMG SRC="europa.jpg" USEMAP="#europa" BORDER="0">
</CENTER>
<P ALIGN="justify">Per l'area dell'Islanda si può scegliere una forma
circolare; per l'Europa occidentale e per quella settentrionale un
rettangolo va più che bene. Per le altre aree sono più indicate forme
poligonali.</P>
<P ALIGN="justify">Quanto ai collegamenti ipertestuali, per l'Islanda
abbiamo messo un collegamento ad una pagina della cartella corrente e
quindi abbiamo utilizzato un <B>indirizzo relativo</B>, "corso.html"; per
la Gran Bretagna è stata indirizzata una pagina di un altro sito, e quindi
mediante un <B>indirizzo assoluto</B> per l'Europa occidentale si è
utilizzata un altro tipo di risorsa, e si è messo il collegamento di tipo
"mailto."; per l'Europa meridionale abbiamo collegato l'area ad una
immagine; per l'Europa Settentrionale e l'Ucraina abbiamo messo due
programmi da scaricare, uno residente sul server corrente e uno su un
altro sito, per cui abbiamo utilizzato un indirizzo relativo e uno
assoluto.</P>
....
</BODY>
</HTML>

```

Guida di HTML

File Modifica Visualizza Preferiti Strumenti

## Mappe immagine

Clicca su una regione dell'immagine per aprire una nuova pagina.



Per l'area dell'Islanda si può scegliere una forma circolare; per l'Europa occidentale e per quella settentrionale un rettangolo va più che bene. Per le altre aree sono più indicate forme poligonali.

Quanto ai collegamenti ipertestuali, per l'Islanda si è messo un collegamento ad una pagina della cartella corrente e quindi abbiamo utilizzato un **indirizzo relativo**, "corso.html"; per la Gran Bretagna è stata indirizzata una pagina di un altro sito, e quindi mediante un **indirizzo assoluto**; per l'Europa occidentale si è utilizzata un altro tipo di risorsa, e si è messo il collegamento di tipo "mailto:"; per l'Europa meridionale abbiamo collegato l'area ad una immagine; per l'Europa Settentrionale e l'Ucraina abbiamo messo due programmi da scaricare, uno residente sul server corrente e uno su un altro sito, per cui abbiamo utilizzato un indirizzo relativo e uno assoluto.

....

## Le tabelle

### Creare una tabella

Le prime versioni di HTML non prevedevano un layout elaborato, ne tantomeno la nozione di tabella. Ora che questo vuoto è stato colmato ci si può domandare come se ne sia potuto fare a meno. Le tabelle sono in assoluto lo strumento di formattazione più usato; esse infatti, non vengono utilizzate solo per visualizzare dati di fogli elettronici e database, ma anche per implementare testo su colonne multiple, o posizionare oggetti in punti particolari della pagina.

Una tabella è definita dal tag contenitore **<TABLE> ... </TABLE>** all'interno del quale si imposta la tabella riga per riga e poi cella per cella.

All'interno della tabella una riga si specifica mediante il tag **<TR> ... </TR>** (Table Row); tutto quello che è compreso tra questi tag appartiene alla stessa riga della tabella.

All'interno di una riga vengono poi specificate le celle della tabella, mediante il tag contenitore **<TD> ... </TD>** (Table Data).

Notiamo come all'interno di questi tag abbiamo volutamente lasciato dei puntini .... Questo per indicare che al loro posto va inserito qualcosa; ed infatti non avrebbe senso definire un elemento contenitore quale una tabella o una cella e poi non metterci nulla dentro, tranne in pochi casi, quando si vuole dare una certa impostazione grafica alla tabella o ad un gruppo di tabelle.

Vediamo subito l'esempio di una tabella semplice, ma prima definiamo un attributo del tag **<TABLE>**, l'attributo **BORDER="n"** che serve a dare un bordo alla tabella e alle sue celle; se è assente significa che la tabella non ha bordo e saranno visibili solo i contenuti delle celle, ma non la struttura. Il numero n sta ad indicare lo spessore del bordo in punti.

La sintassi corretta è **<TABLE BORDER="1">**.

I tag **<TABLE>**, **<TR>** e **<TD>** riconoscono molti attributi, ma li vedremo immediatamente dopo l'esempio, in modo da non mettere troppa carne al fuoco su un argomento che può portare una certa difficoltà all'inizio; ciò è dovuto non tanto ad una certa difficoltà intrinseca alla nozione di tabella, quanto alla complessità e alla grande quantità di elementi che ne contraddistinguono le proprietà.

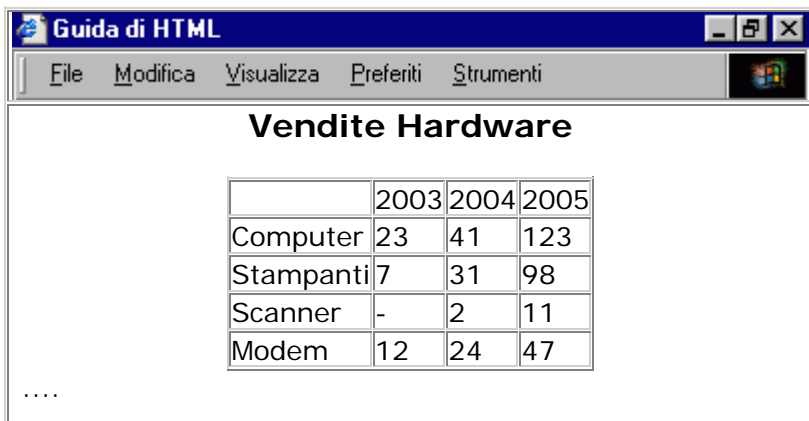
La tabella del nostro esempio è una tabella molto semplice di 5 righe per 4 colonne.

```

<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>

<BODY BGCOLOR="#ECAF AF">
<CENTER><H2>Vendite Hardware</H2>
<TABLE BORDER="1">
<TR>
<TD></TD><TD>1995</TD><TD>1996</TD><TD>
1997</TD>
</TR>
<TR>
<TD>Computer</TD><TD>23</TD><TD>41</TD>
<TD>123</TD>
</TR>
<TR>
<TD>Stampanti</TD><TD>7</TD><TD>31</TD><TD>98</TD>
</TR>
<TR>
<TD>Scanner</TD><TD>-</TD><TD>2</TD><TD>11</TD>
</TR>
<TR>
<TD>Modem</TD><TD>12</TD><TD>24</TD><TD>
47</TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>

```



Da questo esempio vediamo come non ci siano grosse difficoltà nell'uso di tabelle nell'impostazione del testo, e neanche ce ne saranno quando andremo a costruire tabelle più complesse con qualche elemento grafico in più. Quello che occorre sempre tenere



a mente è che HTML interpreta le tabelle leggendone prima una riga e poi l'altra, ed è così che dovremo cercare di costruire una tabella, curandone l'impostazione una riga dopo l'altra.

Il tag **<TABLE>** riconosce gli attributi **ALIGN**, **WIDTH**, **HEIGHT**, **BORDER**, **CELLSPACING** e **CELLPADDING**.

L'attributo **BORDER** lo abbiamo già visto. L'attributo **ALIGN** serve per allineare la tabella con il bordo sinistro o destro della finestra oppure mettere la tabella al centro della pagina; i valori che può assumere sono infatti **"left"**, **"right"** e **"center"**. Gli attributi **WIDTH** e **HEIGHT** servono per impostare le dimensioni della tabella, in punti o in percentuale della intera finestra. Porre **WIDTH="100%"** vuol dire far occupare alla tabella tutto lo spazio a sua disposizione in senso orizzontale. L'attributo **CELLSPACING="n"** consente di specificare la distanza tra i bordi di due celle adiacenti o tra una cella ed il bordo esterno; l'impostazione di default assegna a tale attributo il valore 2. L'attributo **CELLPADDING="n"** consente di impostare la distanza tra il testo ed il bordo della cella circostante; se non impostato assume il valore 1.

Il tag **<TR>** è il più semplice e riconosce gli attributi **ALIGN** e **VALIGN**. Il primo serve, all'interno di tutte le celle della riga, per allineare il testo a sinistra, destra o al centro (**"left"**, **"right"** e **"center"**); il valore predefinito è **"left"** e questo attributo può essere annullato reimpostando l'allineamento cella per cella. L'attributo **VALIGN** serve per allineare il testo in senso verticale, se c'è lo spazio; i valori possibili sono **"top"**, **"middle"** e **"bottom"** e allineano il testo in alto, al centro e in basso dentro ciascuna cella della riga; naturalmente anche questo attributo può essere annullato reimpostando l'allineamento cella per cella.

Il tag **<TD>** riconosce gli attributi **ALIGN**, **VALIGN**, **WIDTH**, **HEIGHT**, **ROWSPAN**, **COLSPAN**, **BGCOLOR** e **NOWRAP**; i primi due hanno lo stesso significato che in **<TR>** e servono per allineare il contenuto della cella in senso orizzontale e verticale; **WIDTH** e **HEIGHT** impostano le dimensioni della cella, rispettivamente la larghezza e l'altezza, entrambi assumono valori espressi sia in numero di punti che in percentuali dello spazio a loro disposizione. **ROWSPAN="x"** e **COLSPAN="y"** permettono di unire due o più celle, e specificano di quanto la cella deve essere espansa; indicare **COLSPAN="3"** vuol dire che la cella in questione deve occupare lo spazio di 3 colonne; descriveremo questi due attributi nel prossimo capitolo, **LAVORARE CON LE CELLE**. L'attributo **BGCOLOR**, come si può ben immaginare permette di impostare il colore di fondo della cella, espresso in notazione esadecimale **"#RRGGBB"** o mediante il nome del colore. Infine, molto interessante, è l'attributo **NOWRAP** che forza

il contenuto della cella a rimanere visualizzato sulla stessa riga, senza ritorno a capo; volendo andare a capo si utilizzano i tag **<BR>** e **<P>**.

Vediamo ora alcuni miglioramenti che possiamo apportare alla nostra tabella.

Il tag **<CAPTION>**, posto all'interno della dichiarazione della tabella, permette di inserire una intestazione o una didascalia alla tabella. Esso accetta l'attributo **ALIGN** con i valori **"top"** e **"bottom"**, con il quale è possibile mettere il titolo sopra o sotto alla tabella.

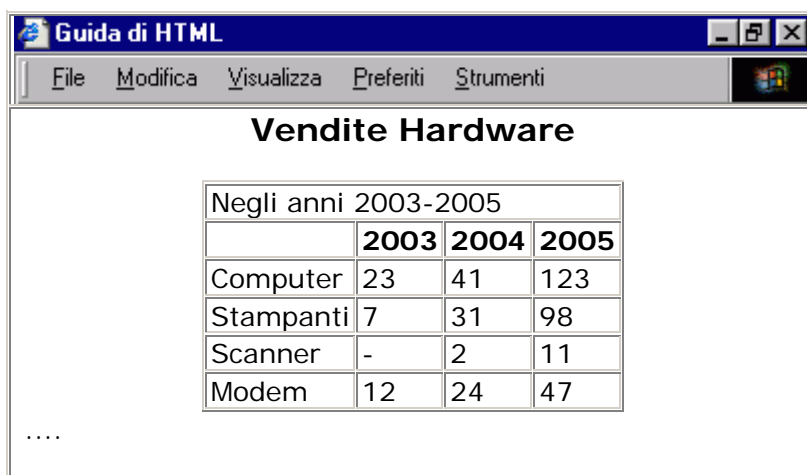
Infine nel visualizzare le colonne è possibile dar loro una intestazione, mediante il tag **<TH>** (Table Heading) al posto dei tag **<TD>**. Nell'esempio che segue aumenteremo le dimensioni della tabella mediante gli attributi **WIDTH** e **HEIGHT** posti dentro il tag **<TABLE>**; questi attributi ammettono come valore una percentuale della finestra a loro disposizione oppure il numero di punti che si vuole come dimensione per la tabella.

Applichiamo queste modifiche alla tabella dell'esempio precedente.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
...
</HEAD>

<BODY BGCOLOR="#BBFFBB">
<CENTER><H2>Vendite Hardware</H2>
<TABLE BORDER="1" WIDTH="90%" HEIGHT="160">
<CAPTION ALIGN="top">Negli anni 2003-2005 </CAPTION>
<TR>
<TH></TH><TH>1995</TH><TH>1996</TH><TH> 1997</TH>
</TR>
<TR>
<TD>Computer</TD><TD>23</TD><TD>41</TD> <TD>123</TD>
</TR>
<TR>
<TD>Stampanti</TD><TD>7</TD><TD>31</TD> <TD>98</TD>
</TR>
<TR>
<TD>Scanner</TD><TD>-</TD><TD>2</TD><TD>11</TD>
</TR>
<TR>
<TD>Modem</TD><TD>12</TD><TD>24</TD><TD>47</TD>
</TR>
</TABLE>
</CENTER>
```

....  
 <P ALIGN="justify">Notiamo come il testo inserito nel tag <CAPTION>  
 non subisce formattazione, e quindi sarebbe bene che gliene dessimo  
 una; mentre il tag <TH> centra il testo e lo visualizza in grassetto.  
 </BODY>  
 </HTML>



Notiamo come il testo inserito nel tag <CAPTION> non subisce formattazione, e quindi sarebbe bene che gliene dessimo una; mentre il tag <TH> centra il testo e lo visualizza in grassetto.

## Le tabelle

### Lavorare con le celle

Cosa può contenere una cella? In pratica tutto, qualunque testo HTML, un testo, delle immagini, e, con le ultime innovazioni di HTML 4.0, addirittura un frame, ossia una nuova pagina. La possibilità di porre qualunque cosa all'interno di una cella di una tabella permette di ottenere particolari layout grafici che non si potrebbero realizzare in altro modo. Togliendo infine i bordi della tabella e facendone sparire la struttura è possibile ottenere effetti grafici straordinari.

Mediante una tabella è possibile ad esempio impostare un menù di collegamenti, mettendo una immagine in ciascuna cella della tabella, e legando ad essa un collegamento ipertestuale. In questo modo si rende la visualizzazione indipendente da tutte le opzioni selezionate per la visualizzazione nei browser (ad esempio la dimensione dei caratteri). Un'altro modo utile per impostare un menù di collegamenti è mediante le mappe immagine, come abbiamo visto qualche capitolo indietro.

Disponendo di tante immagini ciascuna per un collegamento, l'impostazione mediante una tabella rende la visualizzazione dell'intero menù indipendente dalle dimensioni dello schermo, infatti mettendo semplicemente le immagini una dietro l'altra e andando a capo col comando **<BR>** risultava che con uno schermo troppo piccolo le immagini che non entrano nello schermo vanno comunque a capo.

L'esempio che segue mostra un menù di collegamenti realizzato tramite una tabella. Per semplicità nel testo dell'esempio sono state tagliate alcune righe della tabella, ed è mostrata la sintassi solo della prima ed ultima riga. Nella tabella abbiamo impostato **BORDER**, **CELLSPACING** e **CELLPADDING** tutti a zero per fare in modo che le immagini venissero tutte attaccate l'una all'altra. L'esempio fa uso anche dell'attributo **COLSPAN** che verrà descritto immediatamente di seguito.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>

<BODY BGCOLOR="#BBFFBB">
<CENTER><H2>Menù di collegamenti</H2>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0">
<TR>
```

```

<TD><A HREF="attivita.html"><IMG SRC="1.gif"></A></TD>
<TD><A HREF="notizie.html"><IMG SRC="2.gif"></A></TD>
<TD><A HREF="catalogo.html"><IMG SRC="3.gif"></A></TD>
<TD><A HREF="adesioni.html"><IMG SRC="4.gif"></A></TD>
</TR>
[... ]
<TR>
<TD>&nbsp;</TD>
<TD COLSPAN="2"><A HREF="iscrizione.html">
  <IMG SRC="13.gif"></A></TD>
<TD>&nbsp;</TD>
</TR>
</TABLE>
</CENTER>
....
</BODY>
</HTML>

```



Dulcis in fundo, veniamo adesso agli attributi più complessi del tag **<TD>**, complessi non perché siano di difficile comprensione o utilizzo, ma perché un loro utilizzo eccessivo all'interno della progettazione di una tabella può portare ad effetti indesiderati se si fa un po' di confusione.

Stiamo parlando degli attributi **ROWSPAN="x"** e **COLSPAN="y"** che permettono di espandere una cella in senso orizzontale e verticale, e di occupare il posto di un numero di celle specificato dal prodotto dei valori x e y. Se non sono specificati si intendono ovviamente **ROWSPAN="1"** e **COLSPAN="1"**, ossia una cella occupa il posto esattamente di 1 cella.

Specificare **COLSPAN="3"** vuol dire che la cella si espanderà di 2 posti in senso orizzontale, occupando lo spazio di 3 celle, e la riga avrà due celle in meno.

Occorre tenere in considerazione che l'espansione viene effettuata solamente con celle adiacenti che appartengono esclusivamente alla stessa riga e alla stessa colonna.

Poiché un esempio vale più di mille parole, presentiamo un esempio di utilizzo di questi due attributi, disegnando una tabella che non ha alcuna applicazione pratica ma che serve esclusivamente a visualizzare l'effetto di questi due attributi sulla impostazione della tabella.

```

<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
....
</HEAD>

<BODY BGCOLOR="#BBFFBB">
<CENTER><H2>Espansioni orizzontali e verticali</H2>
<TABLE BORDER="1" WIDTH="90%">
  <TR>
    <TD ALIGN="center">UNO</TD>
    <TD ALIGN="center" COLSPAN="2">DUE</TD>
    <TD ALIGN="center">TRE</TD>
    <TD ALIGN="center">QUATTRO</TD>
  </TR>
  <TR>
    <TD ALIGN="center">CINQUE</TD>
    <TD ALIGN="center">SEI</TD>
    <TD ALIGN="center">SETTE</TD>
    <TD ALIGN="center" ROWSPAN="3">OTTO</TD>
    <TD ALIGN="center">NOVE</TD>
  </TR>
  <TR>
    <TD ALIGN="center">DIECI</TD>
    <TD ALIGN="center">UNDICI</TD>
    <TD ALIGN="center">DODICI</TD>
    <TD ALIGN="center">TREDICI</TD>
  </TR>
  <TR>
    <TD ALIGN="center">QUATTORDICI</TD>
    <TD ALIGN="center">QUINDICI</TD>
    <TD ALIGN="center">SEDICI</TD>
    <TD ALIGN="center">DICIASSETTE</TD>
  </TR>
</TABLE>
</CENTER>
....

<P ALIGN="justify">Notiamo come la riga 1 sia formata da 4 elementi
perché l'elemento "DUE" occupa 2 posizioni; la riga 2 è formata da tutti e
5 gli elementi, di cui uno si espande in verticale; le righe 3 e 4 si
compongono di 4 elementi, perché l'elemento "OTTO" appartiene alla
seconda riga.
</BODY>
</HTML>

```

Guida di HTML

File Modifica Visualizza Preferiti Strumenti

## Espansioni orizzontali e verticali

UNO	DUE		TRE	QUATTRO
CINQUE	SEI	SETTE	OTTO	NOVE
DIECI	UNDICI	DODICI		TREDICI
QUATTORDICI	QUINDICI	SEDICI		DICIASSETTE

....

Notiamo come la riga 1 sia formata da 4 elementi perché l'elemento "DUE" occupa 2 posizioni; la riga 2 è formata da tutti e 5 gli elementi, di cui uno si espande in verticale; le righe 3 e 4 si compongono di 4 elementi, perché l'elemento "OTTO" appartiene alla seconda riga.

Avremmo potuto anche unire le celle DUE, SEI e SETTE in modo che la cella che ne risultava veniva parzialmente ad intersecarsi con la cella OTTO. In tal modo le celle SEI e SETTE sparivano, la cella DUE avrebbe avuto **ROWSPAN="2"** e **COLSPAN="2"**, per cui la prima riga avrebbe avuto quattro celle, la seconda ne avrebbe avute 3, la terza e la quarta rimangono con 4 elementi. L'esempio sarebbe però risultato troppo complesso per essere compreso ad un primo utilizzo dei tag relativi alle tabelle e sebbene fosse stato di chiaro effetto grafico non avrebbe portato molti giovamenti a questa guida.

In ogni caso l'esempio è sufficiente per mostrare tutte le potenzialità fornite nell'HTML dallo strumento tabella.

## Gli elenchi

### Elenchi puntati

Per visualizzare una serie di elementi o un elenco di oggetti ci si potrebbe anche accontentare dei normali paragrafi. Noi, nella vita di tutti i giorni, facciamo spesso uso degli elenchi, per fare la lista della spesa (tipico), per fare la lista dei nostri appuntamenti... Persino le cartelle le directory del nostro computer o le canzoni di un CD sono visualizzate tramite un elenco...

Per questo l'HTML ci mette a disposizione ben 5 tipi di elenchi, sebbene solo tre siano realmente utilizzati; li vedremo ciascuno in un capitolo, lasciando da parte gli elenchi di menù e di directory. In questo capitolo ci occuperemo di elenchi non numerati (o non ordinati). Nei capitoli successivi vedremo gli elenchi ordinati e gli elenchi di definizioni.

Gli elenchi non ordinati (o puntati, o non numerati) sono elenchi i cui elementi sono descritti senza una sequenza logica o temporale ma che sono in qualche modo correlati; sono oggetti che sono semplicemente raggruppati in una unica lista.

Tutti i tipi di elenco sono rappresentati da HTML allo stesso modo, con un tag che definisce l'elenco e un altro che indica ciascuna voce dell'elenco.

Il tag che indica un elenco non ordinato è **<UL>** (Unordered List), mentre il tag che indica una voce dell'elenco è **<LI>** (List Item), che può non avere il tag di chiusura **</LI>**. Ogni voce dell'elenco viene indentata in modo da separare l'elenco dal resto del documento.

Entrambi i tag possono avere l'attributo **TYPE** che permette di scegliere il tipo dei punti elenco e i valori che può assumere questo attributo sono **"circle"**, **"square"** o **"disc"**; il valore predefinito è **"disc"**, un cerchio pieno. Specificare l'attributo **TYPE** nel tag **<UL>** vuol dire assegnare a tutte le voci lo stesso simbolo, specificarlo nel tag **<LI>** vuol dire assegnare quel simbolo a quel particolare elemento soltanto.

Vediamo subito un esempio di utilizzo di questo tipo di elenco; mostreremo come scegliere un tipo di simbolo e come cambiarlo per una particolare voce dell'elenco.

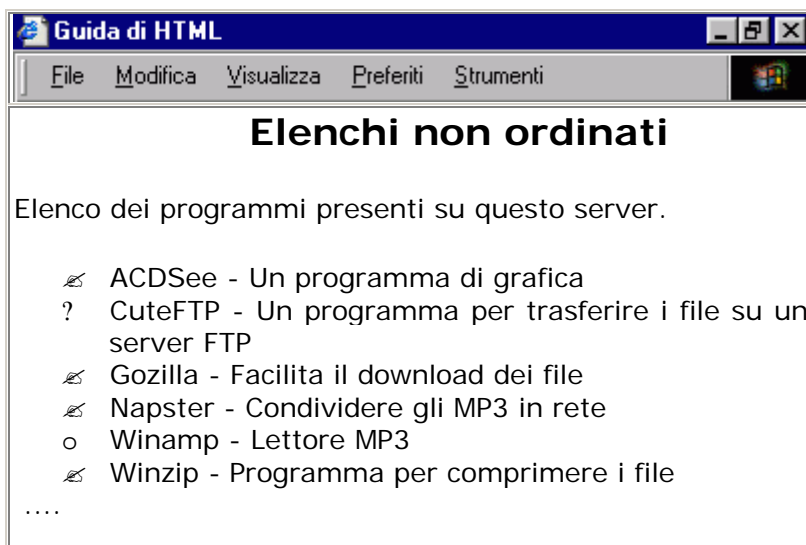
```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
....
</HEAD>
```



```

<BODY BGCOLOR="#FFEB3D">
<H2 ALIGN="center">Elenchi non ordinati</H2>
<P ALIGN="justify">Elenco dei programmi presenti su questo server.
<UL TYPE="square">
  <LI>ACDSee - Un programma di grafica
  <LI TYPE="disc">CuteFTP - Un programma per trasferire i file su un
server FTP
  <LI>Gozilla - Facilita il download dei file
  <LI>Napster - Condividere gli MP3 in rete
  <LI TYPE="circle">Winamp - Lettore MP3
  <LI>Winzip - Programma per comprimere i file
</UL>
....
<P ALIGN="justify">Notiamo come tutti gli elementi sono impostati a
square mediante la definizione nel tag </UL>, mentre per la seconda e
quinta voce sono stati scelti simboli diversi.
</BODY>
</HTML>

```



Notiamo come tutti gli elementi sono impostati a square mediante la definizione nel tag `</UL>`, mentre per la seconda e quinta voce sono stati scelti simboli diversi.

Nello scrivere il codice, abbiamo dato una certa indentatura alle voci interne degli elenchi in modo che siano meglio visibili in fase di analisi e correzione del testo HTML.

## Gli elenchi

### Elenchi ordinati

Gli elenchi ordinati sono utilizzati per descrivere una ben precisa sequenza di eventi o operazioni e a tale scopo ogni voce è contrassegnata da un simbolo di ordine come ad esempio un numero o un carattere.

A tale scopo HTML ci fornisce il tag **<OL>** (Ordered List), che serve appunto ad indicare un elenco numerato di elementi. Per indicare ciascuna voce dell'elenco si utilizza il tag **<LI>** (List Item), esattamente come per gli elenchi non ordinati.

Questi tag possono avere l'attributo **TYPE**, che permette di scegliere il simbolo da usare come numerazione e mentre il tag **<OL>** può avere anche l'attributo **START**, che permette di specificare il valore da cui iniziare il conteggio. Di conseguenza è possibile utilizzare per ciascuna voce un simbolo diverso e partire la numerazione dell'elenco da un numero diverso da **1**.

L'attributo **TYPE** può assumere i valori **"1"** (predefinito), **"A"**, **"a"**, **"I"** e **"i"**, che stanno ad indicare:

- 1** - Stile di enumerazione standard: 1, 2, 3 ...
- A** - mediante caratteri alfabetici maiuscoli: A, B, C ...
- a** - mediante caratteri alfabetici minuscoli: a, b, c ...
- I** - mediante caratteri romani maiuscoli: I, II, III, IV ...
- i** - mediante caratteri romani minuscoli: i, ii, iii, iv ...

L'attributo **START** può assumere valori da -2147483647 a 2147483647, e dare un numero che vada fuori da questo range fa sì che la numerazione riparta dall'estremo opposto. È da ricordare che solo lo stile di numerazione standard permette di utilizzare numeri negativi. Inoltre per specificare il punto di partenza per gli altri tipi si deve sempre utilizzare un numero (senza segno) e non quindi dei caratteri come sarebbe logico attendersi. Sarà quindi poi il browser a convertire il numero (inteso come indice di posizione) nel simbolo dello stile prescelto.

Vediamo subito un esempio di utilizzo di questo tipo di elenco; mostreremo come scegliere un tipo di simbolo e come cambiarlo per una particolare voce dell'elenco.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
....
</HEAD>

<BODY BGCOLOR="#FFEB3D">
```

```

<H2 ALIGN="center">Elenchi ordinati</H2>
<P ALIGN="justify">Istruzioni per installare un programma.
<OL START="2">
  <LI>Decomprimere il file ZIP
  <LI>Eeguire il programma SETUP.EXE
  <LI TYPE="a"> Scegliere la directory di installazione
  <LI TYPE="l">Proseguire con l'installazione
</OL>
....
</BODY>
</HTML>

```



Notiamo come il conteggio parte volontariamente da 2 anziché da 1, e come gli elementi "4" e "5" sono visualizzati con altri simboli.

La maggior parte dei tag contenitore di HTML può essere nidificata all'interno di altri tag; a questa regola non sfuggono gli elenchi. Unendo gli stili di elenco puntato e numerato possiamo ottenere un elenco più complesso.

Vediamo appunto un esempio di elenchi nidificati.

```

<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>

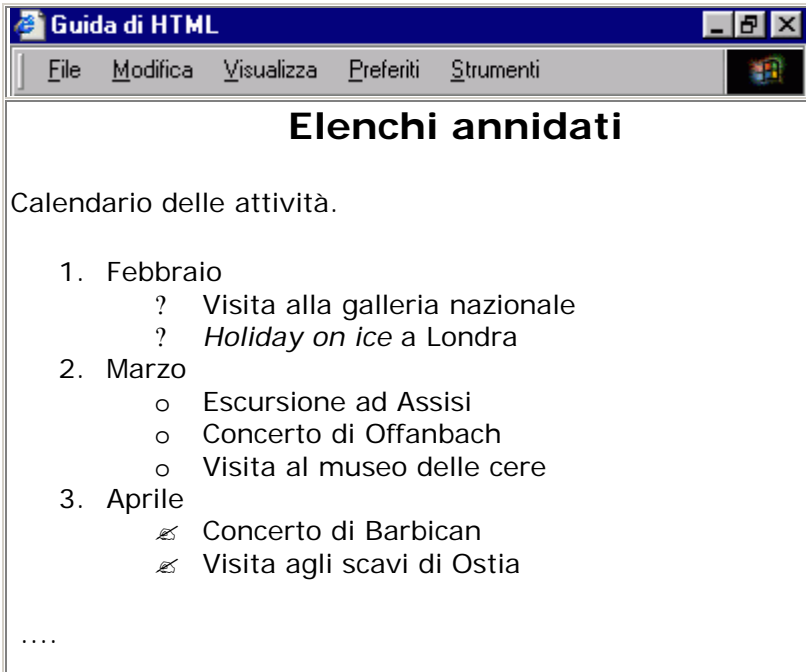
<BODY BGCOLOR="#BBFFBB">
<H2 ALIGN="center">Elenchi annidati</H2>
<P ALIGN="justify">Calendario delle attività.
<OL>
  <LI>Febbraio
  <UL type="disc">
    <LI>Visita alla galleria nazionale
    <LI><l>Holiday on ice</l> a Londra
  </UL>
</OL>

```

```

<LI>Marzo
<UL type="circle">
  <LI>Escursione ad Assisi
  <LI>Concerto di Offanbach
  <LI>Visita al museo delle cere
</UL>
<LI>Aprile
<UL type="square">
  <LI>Concerto di Barbican
  <LI>Visita agli scavi di Ostia
</UL>
</OL>
....
<P ALIGN="justify">Nello scrivere il codice, abbiamo dato una certa
indentatura alle voci interne degli elenchi in modo che siano meglio
visibili in fase di analisi e correzione del testo HTML.
</BODY>
</HTML>

```



The screenshot shows a browser window with the title "Guida di HTML". The menu bar includes "File", "Modifica", "Visualizza", "Preferiti", and "Strumenti". The main content area displays the following text:

## Elenchi annidati

Calendario delle attività.

1. Febbraio
  - ? Visita alla galleria nazionale
  - ? *Holiday on ice* a Londra
2. Marzo
  - o Escursione ad Assisi
  - o Concerto di Offanbach
  - o Visita al museo delle cere
3. Aprile
  - ✎ Concerto di Barbican
  - ✎ Visita agli scavi di Ostia

....

## Gli elenchi

### Elenchi di definizioni

Oltre agli elenchi puntati e numerati, esiste un altro tipo di elenco, molto vicino agli elenchi non numerati. È l'elenco di definizioni, ed è utilizzato quando si vuole visualizzare degli elementi a coppie, come un termine e la sua definizione, un capitolo e i suoi contenuti, o semplicemente si vuole realizzare un glossario.

L'implementazione di questo tipo di elenco è leggermente diversa da quella degli altri elenchi, perché occorre mostrare 2 elementi, che chiameremo per comodità "termine" e "definizione".

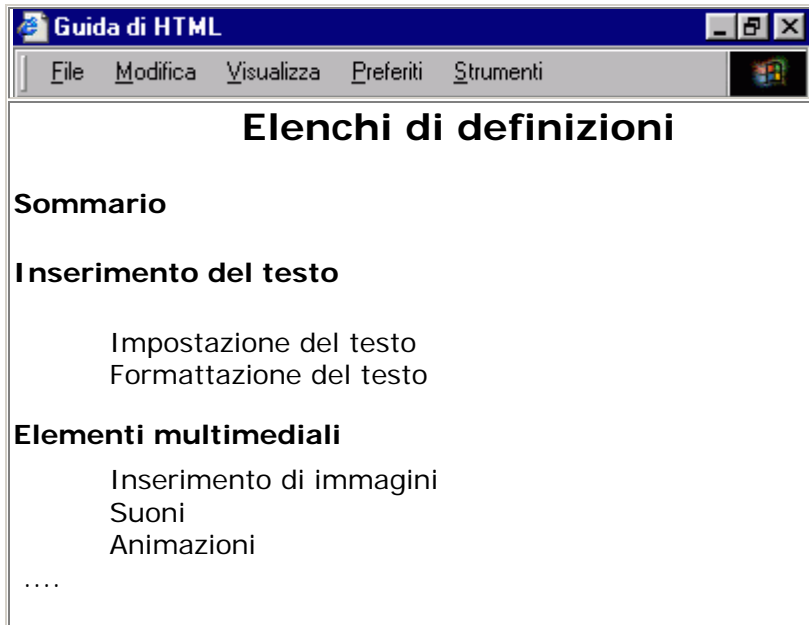
Per specificare un elenco di definizioni si usa il tag **<DL>**, (Definition List); ogni termine verrà indicato tramite il tag **<DT>**; (Definition Term) e ogni definizione mediante il tag **<DD>**; (Definition Data). È facile notare che **<DL>** viene utilizzato allo stesso modo di **<OL>** o **<UL>**, mentre al posto di **<LI>** adesso abbiamo i due tag **<DT>** e **<DD>**.

Nessuno di questi tag riconosce attributi, vanno utilizzati semplicemente così, come li abbiamo introdotti, per cui la loro spiegazione termina qui; l'esempio che segue, tratto dall'indice di questa guida, è più che sufficiente a mostrare l'utilizzo e il funzionamento di questo tag.

Un occhio di riguardo merita l'interlinea, ossia lo spazio tra una voce e l'altra dell'elenco. Non tutti i tipi di elenco sono uguali e parimenti i browser non si comportano tutti allo stesso modo. Quando si costruisce un elenco è bene controllare che ciò che viene visualizzato corrisponde o meno a quello che vorremmo rappresentare sullo schermo, e adottare una soluzione o l'altra a seconda dell'effetto grafico che si vuole ottenere.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#80DCDC">
<H2 ALIGN="center">Elenchi di definizioni</H2>
<P><B>Sommario</B>
<DL>
  <DT>Inserimento del testo
  <DD>Impostazione del testo
  <DD>Formattazione del testo
<DT>Elementi multimediali
  <DD>Inserimento di immagini
  <DD>Suoni
  <DD>Animazioni
```

```
</DL>  
....  
</BODY>  
</HTML>
```



Naturalmente il testo presente nelle definizioni può essere formattato; negli esempi abbiamo sempre usato lo stesso tipo di carattere, ma ovunque questo può essere formattato diversamente, un diverso font, diversa dimensione, colore, può essere messo in grassetto, sottolineato, corsivo... Possono essere presenti anche immagini ed altri elementi HTML.

## I frame

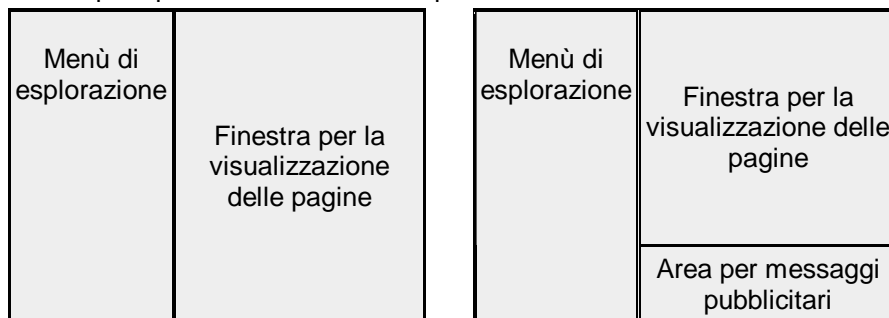
### Creare i frame

Grazie ai frame è possibile visualizzare diverse finestre contemporaneamente, alcune fisse e altre ricaricabili; questa funzione è particolarmente utile per mantenere sempre visualizzata la barra di esplorazione. L'unico problema è che alcuni utenti, che utilizzano vecchi browser, non sono in grado di visualizzarli.

La prima cosa che occorre fare è definire il layout che si vuole implementare per il proprio schermo. In quante aree suddividerlo? Come suddividerlo? Cosa mettere in ciascuna area? Le risposte a queste domande verranno soprattutto dal tipo di sito che si vuole realizzare, dal suo contenuto e dal modo in cui si vuole impostare la visita di un eventuale utente.

La descrizione geometrica della suddivisione in più aree sarà tradotta mediante il tag **<FRAMESET>** che definisce appunto una divisione in frame.

L'esempio qui sotto mostra due possibili suddivisioni in frame.



Sono 2 i tag relativi all'implementazione dei frame: **<FRAMESET>** e **<FRAME>**; il primo è un tag contenitore, ossia necessita anche di tag di chiusura **</FRAMESET>**, mentre il secondo è un comando di markup, e tutte le informazioni necessarie alla sua applicazione sono contenute nei suoi attributi. Il primo infatti imposta una reale partizione dello spazio, mentre il secondo definisce le proprietà generali delle singole aree impostate. In ciascuna area ci andrà quindi una certa pagina HTML. Un terzo tag, **<NOFRAMES>**, è utilizzato per quei browser che non supportano l'uso dei frame, per dare delle informazioni all'utente.

Il tag **<FRAMESET>** riconosce due attributi **COLS** e **ROWS** che servono a suddividere la finestra in colonne o righe. Generalmente si usa uno solo di questi attributi, utilizzando entrambi gli attributi

si può ottenere una suddivisione in entrambe le direzioni, che però non è molto estetica. Entrambi questi attributi assumono come valori le dimensioni delle varie parti in cui viene suddivisa la finestra; si aspettano come valori dei numeri interi che specificano la dimensione esatta in pixel, oppure delle percentuali della finestra complessiva, oppure un asterisco "\*", che indica che il frame occuperà tutto lo spazio rimasto a sua disposizione.

Ad esempio volendo una divisione in tre colonne, una larga 120 pixel, una larga la metà dello schermo e una che occupi il resto dello spazio, si scrive: **<FRAMESET COLS="120,50%,\*">**.

I frame possono essere nidificati uno dentro l'altro, ripetendo il tag **<FRAMESET>**.

Il tag **<FRAME>** riconosce molti attributi: **SRC**, **NAME**, **SCROLLING**, **NORESIZE**, **MARGINWIDTH** e **MARGINHEIGHT**.

Il primo, **SRC**, è indispensabile e indica il documento HTML da caricare in quell'area, ad esempio **SRC="pagina.html"**.

L'attributo **NAME="..."** è altresì molto importante, serve a dare un nome a ciascun frame ed è necessario per specificare in quale frame deve essere caricato un documento specificato in un collegamento ipertestuale.

L'attributo **SCROLLING="..."** consente di visualizzare o meno le barre di scorrimento e permette all'utente di scorrere i contenuti della finestra. I valori possibili sono **"yes"**, **"no"** oppure **"auto"**.

L'attributo **NORESIZE**, che non assume valori, serve per bloccare le dimensioni dei frame, in modo da impedire all'utente di modificarle.

**MARGINWIDTH** e **MARGINHEIGHT** consentono di specificare un margine intorno al frame, rispettivamente ai lati e sopra/sotto. I valori che assumono sono il numero di punti dello spessore del margine.

Infine il tag **<NOFRAMES>...</NOFRAMES>** si mette in aggiunta agli altri due per fornire delle informazioni nel caso l'utente utilizzi un browser che non permette i frame; al posto dei puntini ci andrà il testo HTML da visualizzare in tal caso. Il tag non aspetta attributi.

L'esempio visualizzato qui di seguito mostra un possibile utilizzo dei frame.

```
<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
....
</HEAD>

<FRAMESET COLS="30%,*">
  <FRAME NAME="menu" SRC="menu.htm" NORESIZE>
```

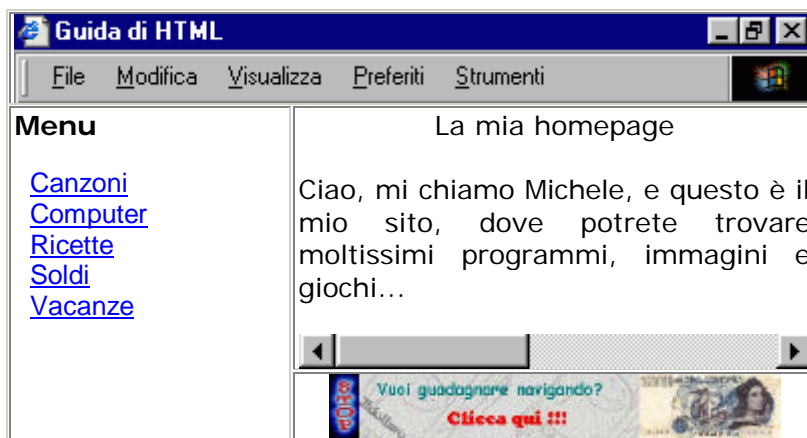


```

<FRAMESET ROWS="*,100">
  <FRAME NAME="main" SRC="pagina.htm">
  <FRAME NAME="publi" SRC="publi.htm" NORESIZE>
</FRAMESET>
</FRAMESET>

<NOFRAMES>
<P ALIGN="justify">Attenzione il tuo browser non supporta i frame.
</NOFRAMES>
</HTML>

```



Notiamo come il tag **<BODY>...</BODY>** sia assente. Questo è stato sostituito da **<FRAMESET>...</FRAMESET>**, che ha la stessa funzione e che contiene soltanto le impostazioni della partizione.

## I frame

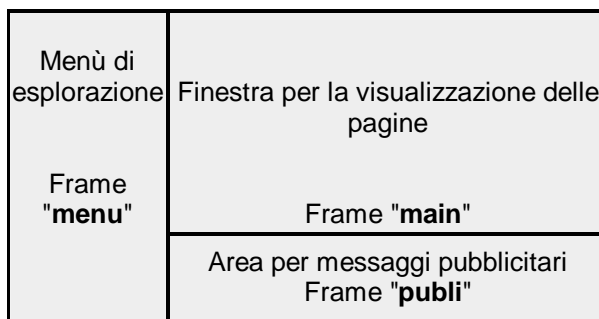
### Lavorare con i frame

Nel capitolo precedente abbiamo visto come dividere una pagina in frame, adesso vedremo come utilizzarli.

Abbiamo deciso di dividere una pagina in 2 o più frame? Bene, per prima cosa occorre creare le pagine interne da caricare; saranno normalissime pagine HTML, come quelle che abbiamo visto finora. Semplicemente non avranno intestazione, perché saranno documenti utilizzati solo dalle pagine che voi avrete creato come contenitore, quindi non avranno **<TITLE>** ne meta-tag, (potreste anche metterceli, ma la maggior parte non viene presa in considerazione)... Anche se i motori di ricerca riuscissero ad indicizzare la pagina, questa verrebbe visualizzata senza il contesto dei frame coi quali è stata creata, e potrebbe risultare non molto gradevole.

Una volta create le pagine da caricare, si può creare la pagina che farà da contenitore, in modo che avrà qualcosa da caricare, utilizzando i comandi visti nel capitolo precedente.

Supponiamo di aver deciso per una divisione a 3 frame, secondo lo schema mostrato qui sotto, e di aver chiamato i tre frame con i nomi "menu", "main" e "publi".



In seguito utilizzeremo sempre questo schema per mostrare esempi di utilizzo dei frame.

Abbiamo visto come far aprire ciascuna pagina in un determinato frame a partire da una pagina che fa da contenitore; ma se abbiamo scelto una impostazione a frame vogliamo poter aprire delle pagine in un determinato frame mediante un collegamento ipertestuale e non dover caricare ogni volta il contenitore. Non dobbiamo introdurre nessun altro comando, con gli strumenti che abbiamo a disposizione possiamo già fare tutto.

Riprendiamo il tag che permetteva di introdurre un collegamento ipertestuale, e lo utilizzeremo nella forma in cui serve a noi:

```
<A HREF="pagina.html" TARGET="main">Clicca qui</A>.
```

Questo comando permette di aprire il documento "pagina.html" nel frame nominato "main" se si clicca sul riferimento "Clicca qui".

Volendo far aprire il documento all'interno del frame "menu", avremmo semplicemente dovuto mettere **TARGET="menu"**.

L'attributo **TARGET** in un collegamento ipertestuale, quindi serve a far caricare il documento collegato al riferimento nel particolare frame deciso da noi. L'attributo assume anche dei valori speciali, in quanto esistono dei "frame" particolari che il browser riconosce sempre, e sono:

**"\_blank"** - per far aprire la pagina in una nuova finestra di browser creata a questo scopo.

**"\_self"** - apre la pagina nello stesso frame in cui è presente il collegamento ipertestuale; è il valore predefinito, quello che viene usato se non è presente l'attributo **TARGET**.

**"\_parent"** - apre la pagina nel frame genitore di quello corrente, ossia nella pagina che ha fatto da contenitore, oppure nella finestra in cui si è effettuata la richiesta, se non ci sono livelli superiori.

**"\_top"** - per far aprire la pagina in tutto lo spazio disponibile nella finestra di browser.

Nel prossimo capitolo descriveremo l'uso di un particolare tipo di frame, l'iframe, che ha la caratteristica di poter essere inserito all'interno di una pagina.

# I Frame

## Gli iframe

I frame sono delle divisioni della finestra in più pagine html indipendenti. Gli iframe (o Internal Frame) sono delle pagine che possono essere caricate all'interno di un'altra pagina, in un'area a loro dedicata, e possono contenere materiale completamente indipendente dalla pagina che li ospita.

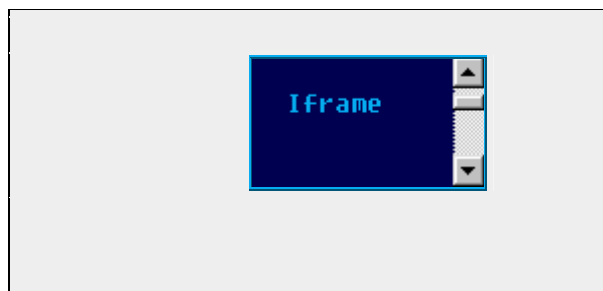
A differenza dei frame, che prevedono una divisione della finestra in parti ben distinte, gli iframe possono essere posizionati dove si vuole all'interno della pagina. All'atto pratico gli iframe sono delle finestrelle aperte all'interno della pagina e mostrano il contenuto di un'altra pagina. Il contenuto degli iframe è un altro documento html che può essere caricato a piacimento con un link nella pagina contenitrice o nella pagina contenuta nell'iframe stesso.

Gli iframe vengono posizionati allo stesso modo in cui si posiziona un qualunque altro oggetto nella pagina, come una immagine o una tabella.

La sintassi corretta per l'inserimento di un iframe all'interno di una pagina è:

**<IFRAME ...></IFRAME>**.

Essendo un oggetto che va inserito in una pagina non sarebbe necessario la chiusura del tag, ma non tutti i browser (soprattutto quelli non recenti) supportano gli iframe, e così tra il tag e la sua chiusura si può mettere un messaggio per avvertire chi non riesce a visualizzare l'iframe, allo stesso modo di come fa il tag **<NOFRAME>** per i frame. Le dimensioni dell'iframe non possono essere modificate dall'utente.



Gli attributi ammessi dal tag **<IFRAME>** sono:

**NAME="nome"**, che sta ad identificare l'iframe e viene utilizzato come **TARGET** per aprire le pagine all'interno degli iframe, mediante i collegamenti ipertestuali.

**SRC="file.html"** indica il documento che viene aperto all'interno dell'iframe.

**WIDTH="xx"** e **HEIGHT="yy"** stabiliscono, in pixel o in percentuale della pagina a disposizione, le dimensioni della finestra in cui si andrà a posizionare l'iframe; tale dimensione non potrà essere variata, se non con tecniche di scripting.

**MARGINWIDTH="tt"** **MARGINHEIGHT="vv"** indicano la distanza in punti dal bordo del testo e degli altri oggetti all'interno dell'iframe.

**SCROLLING="..."** imposta le barre di scorrimento della pagina all'interno dell'iframe. I valori possibili sono **"yes"**, **"no"** e **"auto"**; i primi servono per impostarle come visibili o non presenti, il valore **"auto"** imposterà le barre visibili/invisibili a seconda della lunghezza e delle dimensioni del contenuto della pagina aperta all'interno dell'iframe.

**FRAMEBORDER="nn"** imposta lo spessore in punti della cornice che farà da bordo all'iframe; un valore **"0"** non mostrerà la cornice.

**ALIGN="..."** come tutte le volte che l'abbiamo incontrato stabilisce l'allineamento dell'iframe all'interno della pagina o dell'elemento in cui è collocato, che può anche essere una cella di una tabella.

## I form

### I moduli e l'interfaccia cgi

Il WEB è utilizzato soprattutto per visualizzare informazioni, ma attraverso di esso è possibile anche raccogliere informazioni e dati. I moduli sono elementi che permettono un buon grado di interazione con l'utente e si rivelano particolarmente utili per la raccolta di informazioni.

Un form normalmente è composto da vari tipi di controlli che consentono di inserire testo, selezionare delle opzioni ed effettuare altre operazioni disposte dal WebMaster. Alla fine l'utente conferma le sue scelte effettuando l'invio dei dati immessi, premendo l'apposito bottone a cui normalmente si dà come etichetta Submit o Invia. In seguito a questa operazione viene eseguito un programma residente sul server e a questo vengono passati i dati inseriti; il programma elabora i dati immessi ed effettua alcune operazioni per le quali è stato realizzato (ad esempio la visualizzazione di una pagina, l'invio di una E-Mail o altro).

I programmi sono scritti solitamente in Perl o in C, ed utilizzano l'interfaccia CGI (Common Gateway Interface) per ricevere i dati dalla pagina web.

I metodi per passare i dati allo script sono sostanzialmente due: tramite una richiesta **POST** o tramite una richiesta **GET**; se non diversamente specificato è utilizzata la seconda possibilità.

Tramite il metodo GET i dati vengono passati allo script tramite **l'URL della richiesta** e risultano visualizzati nella barra degli indirizzi; in tale evenienza i parametri sono assegnati alla variabile speciale di ambiente dello script `QUERY_STRING`. Questa deve essere analizzata dallo script e solitamente non può contenere più di 255 caratteri.

La variabile `QUERY_STRING` è separata dall'URL dello script (contenuto nella variabile `SCRIPT_NAME`) da un segno di punto interrogativo '?', mentre i parametri in essa contenuti sono indicati nella forma "nome=valore" e separati dal segno di ampersand '&'.

Il metodo POST è invece molto più versatile perché non pone limiti alla dimensione dei dati passati allo script, in quanto questi vengono infatti inviati attraverso un canale di comunicazione separato (lo standard output). Questo metodo si sta diffondendo sempre di più ed è attualmente quello più utilizzato. Con tale metodo per segnalare la fine dell'input per lo script viene utilizzata la variabile `CONTENT_LENGTH` che indica la dimensione in byte del flusso di dati. Altra informazione passata tramite lo stream di

output è una descrizione del tipo di dati, questa viene copiata nella variabile di ambiente CONTENT\_TYPE.

Il tag che permette l'introduzione di un modulo è **<FORM>**, e tra esso ed il suo tag di chiusura può essere inserito qualunque contenuto HTML valido, tranne altri tag FORM. Esso riconosce due attributi: **ACTION="url"** e **METHOD="get | post"**. L'attributo **ACTION** è un po' come l'attributo **SRC** per le immagini; specifica l'indirizzo del programma che dovrà ricevere i dati immessi nel modulo. **METHOD** specifica invece il modo in cui questi parametri vanno spediti al programma cgi, mediante l'url oppure mediante lo standard output.

La sintassi corretta del tag è la seguente:

```
<FORM  
ACTION="www.server.com/directory/programma.cgi"  
METHOD="post">  
.... (modulo)  
</FORM>
```

I moduli vengono costruiti specificando i campi che devono contenere, ad esempio degli spazi in cui scrivere, dei pulsanti da cliccare, dei bottoni da selezionare etc... opportunamente disposti sullo schermo per dare una visualizzazione compatta e gradevole. Questo verrà realizzato mediante appositi tag, che sono:

**<INPUT>**, **<SELECT>** e **<TEXTAREA>**, che vedremo nel prossimo capitolo.

## I form

### Raccogliere le informazioni

Nel capitolo precedente abbiamo visto come introdurre un modulo in una pagina web; però un modulo così introdotto non serve a niente, non è neanche in grado di visualizzare qualcosa sullo schermo, occorre inserire degli elementi adatti a raccogliere le informazioni inserite dall'utente.

Questi elementi possono essere campi in cui l'utente possa scrivere, pulsanti da selezionare, bottoni da cliccare o altri elementi su cui effettuare una scelta.

Il tag principale e quello più utilizzato è **<INPUT>** che specifica un dato in ingresso per il modulo. Il tag **<INPUT>** riceve alcuni attributi che indicano la natura della informazione introdotta. L'attributo **NAME="testo"** specifica l'identificativo di quell'elemento introdotto, in pratica il nome della variabile a cui l'utente fornisce il valore.

Infine l'attributo **TYPE="tipo"** che indica il tipo di informazione da raccogliere. Può assumere i valori **"text"**, **"password"**, **"hidden"**, **"checkbox"**, **"radio"**, **"submit"**, **"reset"** e **"image"**.

Esaminiamo in dettaglio i valori di questo tipo di dato:

- ? **TYPE="text"** è utilizzato per ricevere del testo come input dall'utente, infatti esso visualizza una casella in cui scrivere; esso utilizza anche alcuni attributi supplementari:
  - MAXLENGTH="n"** indica il numero massimo di caratteri che è possibile inserire (se non specificato, non viene posto alcun limite), **SIZE="n"** indica in caratteri la dimensione della casella di testo, **VALUE="testo"** indica il valore predefinito iniziale a cui è impostata la variabile, se l'utente non scrive nulla sarà questo ad essere trasmesso.
- ? **TYPE="password"** si comporta allo stesso identico modo di **"text"** con la sola eccezione che i caratteri vengono visualizzati come asterischi '\*'; in ogni caso il valore è passato al programma senza alcuna criptazione e quindi nella forma originale inserita dall'utente.
- ? **TYPE="hidden"** si comporta alla stregua di "text", ma non visualizza alcunché sullo schermo. È utilizzato per passare dei valori allo script e/o per contenere valori temporanei acquisiti in moduli che si sviluppano su più pagine.
- ? **TYPE="checkbox"** visualizza sullo schermo delle caselle quadrate spuntabili per indicare delle situazioni di tipo vero/falso; un attributo che si può usare con questo tipo di dato è **VALUE="testo"** ed indica il valore da ritornare allo



script nel caso in cui la casella venga spuntata; se non specificato viene ritornato il valore 'on'; altro attributo è **CHECKED**, senza valori, che se utilizzato rende la casella già spuntata al momento della visualizzazione.

- ? **TYPE="radio"** definisce delle caselle di selezione a mutua esclusione, ossia è possibile selezionare solo una voce alla volta tra quelle disponibili; graficamente è rappresentato da un cerchio che se pieno contrassegna la voce come selezionata; per definire un gruppo di voci a mutua esclusione devono essere definiti tutti i pulsanti con **<INPUT TYPE="radio" NAME="nome">** e avere tutti lo stesso nome, ossia lo stesso valore assegnato all'attributo **NAME** di ogni pulsante; a questo tipo si applica anche l'attributo **VALUE="testo"** che è il valore ritornato al programma e può essere diverso da casella a casella, e l'attributo **CHECKED**, senza valori, per selezionare un valore iniziale (si consiglia, pena comportamenti imprevedibili da parte del browser, di settare come **CHECKED** una sola voce per ogni gruppo).
- ? **TYPE="submit"** crea il pulsante per l'invio dei dati raccolti; mediante l'attributo **VALUE="testo"** è possibile definire un'etichetta diversa da quella di default che è **"Submit"**.
- ? **TYPE="reset"** crea un pulsante per annullare tutti i valori inseriti dall'utente, in modo che possa ricominciare da capo; mediante l'attributo **VALUE="testo"** è possibile definire un'etichetta diversa da quella di default che è **"Reset"**.
- ? **TYPE="image"** si comporta esattamente come il pulsante **"Submit"** ma visualizza un'immagine al posto del pulsante di invio; l'immagine è specificata dall'attributo aggiuntivo **SRC="immagine.gif"**. Assegnando all'immagine anche l'attributo **NAME="nome"**, si fa sì che al programma vengano passate anche le informazioni sulla posizione del cursore del mouse al momento del clic; queste vengono passate nella forma **nome.x=CoordX&nome.y=CoordY**; nome indica il nome che si è assegnato all'immagine tramite l'attributo name, mentre CoordX e CoordY sono le coordinate x e y del cursore all'interno dell'immagine.

Vediamo adesso un esempio di modulo, in cui utilizziamo alcuni dei controlli del tag **<INPUT>**; sarà solo un esempio di visualizzazione delle caselle, non sarà un modulo vero e proprio, perché non possiamo ancora progettare il programma che riceverà ed elaborerà i dati, in quanto occorrono delle ulteriori conoscenze di programmazione per farlo; occorrerà conoscere linguaggi di programmazione come il C, oppure di scripting come il Perl, e di questi ce ne occuperemo nella prossima guida.

```

<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#BBFFBB">
<H2 ALIGN="center">I moduli</H2>
<P ALIGN="justify">Inserisci i dati richiesti.
<FORM ACTION="..url.." METHOD="post">
Nome : <INPUT TYPE="text" NAME="nome" SIZE="18"><BR>
Password : <INPUT TYPE="password"
NAME="pwd" SIZE="15"><BR>
Stato familiare<BR>
single <INPUT TYPE="radio" NAME="sfam"
VALUE="single" CHECKED>, coniugato <INPUT TYPE="radio"
NAME="sfam" VALUE="coniugato"><BR>
Hobbies<BR>
pesca <INPUT TYPE="checkbox"
NAME="pesca">, disegno <INPUT TYPE="checkbox"
NAME="disegno"><BR>
<INPUT TYPE="reset" VALUE="Annulla">
<INPUT TYPE="submit" VALUE="Invia">
<BR>
</FORM>
</BODY>
</HTML>

```

**Guida di HTML**

File Modifica Visualizza Preferiti Strumenti

## I moduli

Inserisci i dati richiesti.

Nome :

Password :

Stato familiare  
 single , coniugato

Hobbies:  
 pesca , disegno

Annulla Invia

## I form

### Informazioni aggiuntive

Il tag **<INPUT>** non è il solo utilizzato per reperire informazioni, ne esistono altri due, usati molto anch'essi. Sono i tag **<TEXTAREA>** e **<SELECT>**.

Il tag **<TEXTAREA>** si comporta come il tag **<INPUT TYPE="text">** solo che il contenuto della casella può essere inserito su più righe. Come **<INPUT>** esso infatti rileva l'attributo **NAME="nome"** con cui associare il nome della variabile, mentre ignora gli attributi **MAXLENGHT**, **SIZE** e **VALUE**; è possibile però definire le dimensioni della casella mediante gli attributi **cols="n"** e **rows="n"** che indicano la larghezza in numero di caratteri e l'altezza in numero di righe.

Il tag **<TEXTAREA>** necessita del tag di chiusura **</TEXTAREA>** ed in mezzo ai due tag ci va l'eventuale testo che vorremmo inserire di default all'interno della casella; sarà l'utente a cancellarlo se vorrà cambiarlo. Non è possibile dare una formattazione al testo da inserire nell'area, e quindi non è possibile inserire del codice HTML tra questi due tag; tutto quello che sarà scritto nel documento HTML apparirà all'interno della casella.

Il tag **<SELECT>** che permette di inserire una casella di selezione a discesa. Una casella di selezione a discesa è una casella contenente alcune voci, visualizzate una sopra l'altra e tra le quali è possibile selezionarne una. Tra il tag di apertura e di chiusura vanno messe le varie opzioni selezionabili, mediante il tag **<OPTION>**.

**<SELECT>** ammette tre attributi, **NAME="nome"**, **SIZE="n"** e **MULTIPLE**; l'attributo **NAME** serve per dare un nome alla variabile in questione, **SIZE** serve per indicare il numero di voci da visualizzare sullo schermo e infine **MULTIPLE** che se usato consente di selezionare più voci simultaneamente.

Il tag **<OPTION>**, come appena specificato, è usato per indicare una ad una le varie opzioni della casella; esso riconosce 2 attributi, **VALUE="testo"** e **SELECTED**; il primo serve per indicare il valore che sarà passato al programma legato alla variabile "nome", mentre **SELECTED**, dove utilizzato, indica che quella voce è inizialmente selezionata per default; per evitare comportamenti imprevedibili da parte del browser è bene che questo attributo sia associato ad una sola opzione. Il testo compreso tra tag di apertura e di chiusura sarà quello che verrà visualizzato nella pagina web all'interno della casella.

Vediamo quindi un esempio di un modulo che utilizzi questi due tag appena introdotti. Come per il capitolo precedente il modulo è solo fittizio, non essendoci alcun programma in grado di ricevere i dati.

```

<HTML>
<HEAD>
<TITLE>Guida di HTML</TITLE>
...
</HEAD>

<BODY BGCOLOR="#FFEBCD">
<H2 ALIGN="center">Modulo 2</H2>
<P>Seleziona un argomento e<BR> inserisci un breve commento.</P>
Argomento : <SELECT NAME="argom"
SIZE="3" ALIGN="middle">
<OPTION VALUE="sport">Sport</OPTION>
<OPTION VALUE="arte" SELECTED>Arte</OPTION>
<OPTION VALUE="econ">Economia</OPTION>
<OPTION VALUE="cultura">Cultura</OPTION>
<OPTION VALUE="storia">Storia</OPTION>
<OPTION VALUE="tecno">Tecnologia</OPTION>
</SELECT><BR>

Commento:<BR>
<TEXTAREA NAME="comment" rows="5"
cols="20"></TEXTAREA><BR>

Giudizio : <SELECT NAME="giudizio">
<OPTION VALUE="1">1</OPTION>
<OPTION VALUE="2">2</OPTION>
<OPTION VALUE="3">3</OPTION>
<OPTION VALUE="4" SELECTED>4</OPTION>
<OPTION VALUE="5">5</OPTION>
<OPTION VALUE="6">6</OPTION>
</SELECT><BR>
<INPUT TYPE="reset" VALUE="Annulla">
<INPUT TYPE="submit" VALUE="Invia">
<BR>
</FORM>

</BODY>
</HTML>

```

**Guida di HTML**

File Modifica Visualizza Preferiti Strumenti

## Modulo 2

Seleziona un argomento e inserisci un breve commento.

Argomento:

Commento:

Giudizio:

Con questo capitolo termina lo spazio dedicato alla costruzione dei moduli e all'HTML vero e proprio. Più avanti discuteremo dei linguaggi di programmazione e/o di scripting che ci permetteranno di creare i programmi che elaborino i dati raccolti mediante i moduli. Ricordiamo solamente che in internet numerosi siti mettono a disposizione migliaia di programmi già fatti e pronti per essere utilizzati all'interno di un sito web.

La prossima sezione descrive ciascun TAG nel dettaglio, permettendone di analizzare tutti gli aspetti, i modi di utilizzo e le differenze di implementazione e di visualizzazione tra i vari browser in circolazione.

## Tutti i tag

### Il tag **<A>**

L'elemento contenitore

**<A> ... </A>**

è usato per indicare o ancorare un particolare punto in un file HTML o per associare un collegamento ipertestuale alla porzione di documento racchiusa tra il tag di apertura e quello di chiusura, che può essere del semplice testo o una immagine, oppure una sequenza di elementi.

L'assenza di contenuto tra i tag **<A>** e **</A>** è visto come un errore da alcuni browser, ma la maggior parte dei browser ignora questo difetto.

Lo standard HTML 3.2 definisce cinque attributi per questo tag, tutti facoltativi, sebbene almeno uno di essi deve essere presente affinché il tag sia di qualche utilizzo. Gli attributi **HREF** e **NAME** sono quelli più comunemente usati.

#### 1. **HREF**

Il valore di questo attributo è un URL ovvero un indirizzo di un qualche elemento presente sul web. All'URL viene associato tutto ciò che è compreso tra i tag **<A> ... </A>**. Quando il contenuto viene selezionato mediante un click del mouse, il browser accederà alla risorsa indicata dall'URL (tipicamente un altro file). L'esempio qui sotto mostra la realizzazione di un collegamento ipertestuale:

Qui abbiamo un **<A HREF="a-esempio.html">Esempio</A>**, di collegamento ..

**NOTA.** Mediante l'uso di un diverso colore è possibile distinguere maggiormente il testo di un collegamento ipertestuale. Le immagini hanno solitamente un bordo che le distingue. Il colore è determinato dagli attributi **LINK**, **ALINK** e **VLINK** del tag **<BODY>**. L'uso dell'attributo **COLOR** del tag **<FONT>** non avrà effetto. Se gli attributi del tag **<BODY>** non sono esplicitamente dichiarati, allora la visualizzazione di un collegamento testuale (che comunemente include anche la sottolineatura) è interamente sotto il controllo del browser, a meno che non si dia una formattazione voluta al testo visualizzato.

## 2. NAME

Questo attributo permette di denominare la corrente posizione all'interno del file con il valore associato all'attributo. Un indirizzo può essere costruito puntando alla posizione all'interno del file aggiungendo *#stringa* alla fine dell'indirizzo che identifica il file, in cui *stringa* è proprio il valore dell'attributo.

Il seguente codice può essere inserito alla fine di questo capitolo

```
<A NAME=browser><H3>Browser</H3></A>
...
```

Questo implementa un'ancora in quella parte della pagina, e permette di raggiungerla con un semplice click. È possibile all'interno della stessa pagina raggiungere la posizione desiderata mediante **<A HREF..>**. I seguenti codici permettono tutti di raggiungere il fondo del capitolo.

```
<A HREF=#browser>Clicca qui</A>
<A HREF=a.html#browser>Clicca qui</A>
<A HREF=
http://www.canoro.altervista.org/guide/html/tag/a.html#browser>Clicca
qui</A>
```

L'uso di collegamenti interni ai documenti, se gestiti mediante un menù ad inizio pagina, rende più comoda la navigazione all'interno di pagine molto lunghe.

## 3. REL

Lo standard HTML 3.2 definisce questo attributo come una relazione diretta conosciuta anche come "tipo link". Esso può essere usato per per indicare il tipo di relazione esistente tra il documento corrente e quello collegato. **REL** indica la relazione diretta, che è di gran lunga la più utilizzata. Può assumere (così come **REV**) molti valori, tra i quali i più importanti sono: **alternate** che specifica una differente versione del documento, **author** che indica l'autore del documento, **copyright** per definire il copyright della pagina, **glossary** che collega ad un glossario per questo documento, **help** per la pagina di aiuto per il documento corrente, **made** per specificare l'autore della pagina, e forse il più utilizzato **stylesheet** che definisce il file con tutte le indicazioni per lo stile grafico del documento.

#### 4. **REV**

L'attributo **REV** definisce una relazione inversa. Ad esempio **REV=help** sta ad indicare che questo file è la pagina di aiuto per il documento collegato da quella relazione; oppure **REV=made** è utilizzato per specificare l'autore del documento, mediante l'uso di un indirizzo email con un URL di tipo *mailto*.

#### 5. **TITLE**

Questo attributo mostra semplicemente un messaggio di spiegazione che riguarda il file collegato.

#### 6. **TARGET**

Questo attributo non appartiene allo standard HTML 3.2 ed è usato in corrispondenza di frame per identificare la finestra o il frame all'interno del quale sarà aperto il file specificato nell'URL.

## **Browser**

Tutti i browser visualizzano questo tag allo stesso modo, sebbene ci siano alcune differenze nella visualizzazione quando il contenuto viene selezionato; Opera 2.22 disegna un rettangolo attorno al testo.



## Tutti i tag

### Il tag **<ADDRESS>**

L'elemento contenitore

**<ADDRESS> ... </ADDRESS>**

è inteso per informazioni come un indirizzo, un numero di telefono etc .., all'interno del documento.

Il testo incluso tra i tag viene comunemente visualizzato usando un tipo di carattere corsivo. Sebbene gli indirizzi possano consistere di più righe, il carattere di "a capo" tra i tag **<ADDRESS>** e **</ADDRESS>** non viene riconosciuto ed è necessario inserire il tag **<BR>** per causare l'interruzione di riga.

Ecco un esempio di utilizzo del tag:

*Università di Roma La Sapienza  
Piazzale Aldo Moro 5  
00185 Roma*

e questo è il codice che lo genera:

```
<ADDRESS>  
Università di Roma La Sapienza<BR>  
Piazzale Aldo Moro 5<BR>  
00185 Roma<BR>  
</ADDRESS>
```

## Browser

Ci sono grosse variazioni nel modo in cui questo tag viene visualizzato da parte dei vari browser. Lo standard 3.2 richiede che il browser inserisca una interruzione di paragrafo prima e dopo il testo contenuto nei tag. Solo Mosaic 3.0 e Ariadna 1.2 rispettano lo standard. Internet Explorer 3 addirittura dimentica di visualizzare il corsivo, un difetto corretto a partire di Internet Explorer 4, pur non gestendo l'interruzione di paragrafo. Opera 2.0 inserisce una interruzione di paragrafo all'inizio, ma non alla fine. Tutte le versioni successive si comportano allo stesso modo.

## Tutti i tag

### Il tag **<APPLET>**

L'elemento contenitore

**<APPLET> ... </APPLET>**

è usato per includere o avviare applicazioni Java nel codice HTML. Il testo incluso tra i tag viene ignorato dai browser che possono visualizzare i contenuti in Java, mentre viene mostrato dai browser che non riconoscono il tag e quindi lo ignorano.

Il tag ha nove attributi, che sono molto simili agli attributi del tag **<IMG>** e controllano la posizione e le caratteristiche dell'area in cui viene visualizzata l'applicazione.

#### 1. **CODEBASE**

Il valore di questo attributo, facoltativo, specifica la cartella dalla quale l'applicazione viene caricata; se è assente l'applicazione viene cercata nella stessa cartella della pagina.

#### 2. **CODE**

Questo attributo identifica il file della classe compilata contenente il codice dell'applicazione.

#### 3. **ALT**

Il valore di questo attributo è il testo che deve essere visualizzato dai browser che riconoscono il tag **APPLET** ma che non possono gestire applicazioni Java perché non abilitati, probabilmente per l'assenza di alcune librerie o perché l'utente ha non impostato il caricamento delle applicazioni Java.

#### 4. **NAME**

Questo attributo assegna un nome all'applet in modo che altre applicazioni (nella stessa pagina) possano comunicare con essa.

#### 5. **WIDTH**

Specifica la larghezza in pixel dell'area in cui verrà mostrata l'applicazione. Lo standard HTML 3.2 non consente l'uso di percentuali in questo contesto.

## 6. **HEIGHT**

Specifica l'altezza in pixel dell'area in cui verrà mostrata l'applicazione. Lo standard HTML 3.2 non consente l'uso di percentuali in questo contesto.

## 7. **ALIGN**

Il valore di questo attributo ed il suo significato sono gli stessi che per il tag IMG e specifica l'allineamento sul display dell'area relativa all'applet o al testo alternativo mostrato. I valori **texttop**, **absmiddle**, **baseline** e **absbottom** utilizzati con standard precedenti non sono menzionati nello standard HTML 3.2 e quindi alcuni browser potrebbero non interpretarli. Questi valori erano usati da vecchie versioni del browser Netscape per avere un più preciso controllo della posizione dell'immagine. Le versioni attuali di Internet Explorer sono in grado di interpretare anche questi valori, sebbene il loro utilizzo debba essere considerato un errore.

## 8. **VSPACE**

Specifica la distanza in pixel del testo dalla zona attribuita alla visualizzazione dell'applet, sopra e sotto l'area.

## 9. **HSPACE**

Specifica la distanza in pixel del testo dalla zona attribuita alla visualizzazione dell'applet, ai lati dell'area.

Gli attributi **CODE**, **WIDTH**, **HEIGHT** e **ALIGN** sono indicati come necessari dallo standard HTML 3.2

Vedi anche il tag **<PARAM>**.

## Tutti i tag

### Il tag **<AREA>**

L'elemento HTML

#### **<AREA>**

è usato per specificare ciascuna singola zona attiva dell'immagine. Un contenitore **<MAP> ... </MAP>** può contenere tanti tag **<AREA>** quanti sono necessari. L'effetto di includere qualcos'altro tra **<MAP>** e **</MAP>** non è completamente definito e dipende da browser e browser.

Il tag **<AREA>** ha cinque attributi:

#### 1. **SHAPE**

Definisce la forma dell'area attiva e può assumere quattro possibili valori: **rect**, **circle**, **poly** e **default**. Se l'attributo non è specificato, allora viene assunto **rect** come predefinito.

#### 2. **COORDS**

Definisce la zona attiva. Il valore assunto è una serie di numeri separati da una virgola, che indicano le coordinate in pixel (o in percentuale delle dimensioni dell'immagine) dei punti che definiscono l'area.

L'interpretazione dei valori è la seguente:

<b>SHAPE</b>	<b>numero di valori</b>	<b>valori di COORDS</b>
<b>rect</b>	4	x-sinistra, y-superiore, x-destra, y-inferiore
<b>circle</b>	3	x-centro, y-centro, raggio
<b>poly</b>	n >= 6	x1, y1, x2, y2, x3, y3 ...

Nel caso di un poligono, si assume la chiusura dell'area attiva. Il poligono si considera chiuso da una linea che parte dall'ultima coppia di coordinate alla prima.

Se le regioni si sovrappongono allora il browser prende in considerazione la prima regione che viene definita, come se le altre aree andassero sotto e non fossero raggiungibili dal click del mouse.

#### 3. **ALT**

Associa un testo di commento all'area selezionata, in maniera del tutto simile all'attributo **ALT** del tag **<IMG>**.

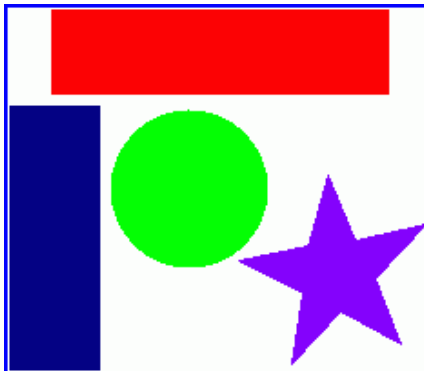
#### 4. **HREF**

Associa un URL, un indirizzo di una pagina HTML o un programma CGI alla regione in questione.

## 5. NOHREF

Questo attributo indica che non c'è alcun URL associato con l'area; può essere utilizzato per creare un buco in una regione.

Ecco un possibile esempio di visualizzazione di una mappa immagine. È possibile cliccare su ciascuna area colorata.



E questo sotto è il codice che lo genera

```
<IMG SRC=area.gif USEMAP="#map1">
<MAP NAME="map1">
<AREA NOHREF ALT="Buco nel rettangolo rosso" SHAPE=rect
COORDS="34, 8, 200, 40">
<AREA HREF=mappa1.html ALT="Rettangolo rosso" SHAPE=rect
COORDS="24, 1, 209, 47">
<AREA HREF=mappa2.html ALT="Rettangolo blu" SHAPE=rect
COORDS="1, 54, 50, 199">
<AREA HREF=mappa3.html ALT="Cercho verde" SHAPE=circle
COORDS="100, 100, 44">
<AREA HREF=mappa4.html ALT="Stella viola" SHAPE=poly
COORDS="175, 92, 191, 128, 229, 120, 202, 150, 215, 186, 182, 168,
155, 197, 161, 156, 126, 140, 164, 131">
<AREA HREF=mappa5.html ALT="Sfondo" SHAPE=default>
</MAP>
```

Non c'è bisogno di mettere il tag **<MAP>...</MAP>** immediatamente dopo il tag **<IMG>**. L'ancora "map1" serve per legarli insieme, e quindi il tag **<MAP>** può stare ovunque nel documento, anche prima del tag **<IMG>**.

Nell'esempio possiamo anche vedere come l'attributo **NOHREF** sia usato per impostare un buco non cliccabile all'interno del rettangolo rosso.

L'area selezionata non viene nemmeno convertita al valore associato all'attributo **SHAPE="default"**. Se il browser non visualizza il testo contenuto nell'attributo **ALT** quando si muove il

mouse sulle zone in cui non sono presenti collegamenti, il mouse cambierà forma quando si passa sul buco nel rettangolo rosso.

Nell'esempio sono utilizzate 10 coppie di coordinate per identificare la stella viola, una per ogni vertice del perimetro della stella. È possibile definire la stella anche usando solo 5 coppie di coordinate, come nella figura a destra, ma gli algoritmi utilizzati dalla maggior parte dei browser per determinare l'area cliccabile sono inadatti con questo ordine di punti. Solo Ariadna riesce a identificare correttamente l'area attiva.



Questo è il codice associato alla stella:

```
<IMG SRC=stellaper.gif ALIGN=right HSPACE=5 USEMAP="#map2">
<MAP NAME="map2">
<AREA HREF=mappa4.html ALT="Stella viola" SHAPE=poly
COORDS="3, 50, 106, 30, 32, 108, 52, 3, 93, 97">
</MAP>
```

## Browser

Ariadna, Internet Explorer, Netscape Navigator e Opera gestiscono le mappe area in modo corretto. Nessun browser riconosce correttamente l'attributo **ALT**. Internet Explorer non riconosce il valore **default** per l'attributo **SHAPE**.

Vedi anche i tag **<MAP>**, **<IMG>** e le note generali sulle mappe attive.

## Tutti i tag

### Il tag **<B>**

L'elemento contenitore

**<B> ... </B>**

ordina al browser di mostrare in grassetto il testo contenuto tra i tag di apertura e chiusura. Lo standard HTML definisce **<B>** uno stile fisico distinguendolo dal tag **<STRONG>** che è uno stile logico.

Ecco un esempio di utilizzo del tag:

Questo è uno stile **grassetto**.

e questo è il codice che lo genera:

```
Questo è uno stile <B>grassetto</B>
```

La differenza tra uno stile fisico e uno logico è che nel primo caso lo stile indica al browser di visualizzare il testo in un tipo di carattere in grassetto, mentre con uno stile logico al browser è detto di mostrare il testo con una enfasi maggiore, lasciando il browser libero di decidere come ciò verrà eseguito (possono essere usati colori diversi o sottolineatura), anche se ormai tutti i browser si stanno adattando a mostrare in grassetto un testo indicato come **STRONG**.

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

## Tutti i tag

### Il tag **<BASEFONT>**

L'elemento

**<BASEFONT>**

specifica il carattere base per quella pagina.

Questo sarà il font mediante il quale il testo sarà normalmente visualizzato; la dimensione corrente potrà essere modificata mediante l'attributo **SIZE** del tag **<FONT>** con un valore relativo o assoluto. Tutto il testo, fatta eccezione per i titoli, che non sia esplicitamente formattato sarà visualizzato nel modo indicato dal tag **<BASEFONT>** fino alla fine del documento oppure fin quando sarà incontrato un nuovo tag **<BASEFONT>**.

Il tag ha tre attributi, gli stessi che appartengono al tag **<FONT>**:

**1. SIZE**

Questo attributo ha un valore numerico da **1** a **7** che specifica la dimensione del carattere. In assenza di questo attributo la dimensione sarà stabilita al valore **3**.

**2. COLOR**

Questo attributo non standard HTML 3.2 stabilisce il colore del testo; può essere utilizzato un valore in esadecimale oppure il nome del colore (vedi la sezione apposita nell'appendice di questa guida).

**3. FACE**

Questo attributo non standard HTML 3.2 definisce il tipo di carattere utilizzato per visualizzare il testo.

**NOTA.** Quando si cambia il carattere predefinito per la visualizzazione del testo mediante il tag **<BASEFONT>** non c'è modo di tornare al vecchio tipo di carattere utilizzato o al tipo di carattere predefinito. Volendo cambiare il tipo di carattere occorre inserire un nuovo tag **<BASEFONT>**.

L'uso di fogli di stile è un modo migliore e più flessibile per ottenere i risultati associati ad un uso di tag **<BASEFONT>** multipli.



## Browser

Ariadna 1.2, Mosaic 3.0 e Opera 2.12 ignorano il tag **<BASEFONT>**. Netscape 3 elabora il tag, ma non gli attributi **COLOR** e **FACE**. Internet Explorer gestisce tutti gli attributi.

Vedi anche il tag **<FONT>**.

## Tutti i tag

### Il tag **<BGSOUND>**

Il tag

**<BGSOUND>**

non è un tag standard HTML ed è usato per impostare un suono di sottofondo per la pagina visualizzata.

Il tag prevede due attributi:

1. **SRC**

è l'indirizzo del file contenente il suono.

2. **LOOP**

Il valore di questo attributo è il numero delle volte che il suono viene ripetuto. I valori aspettati sono **"infinite"** oppure un qualunque intero positivo da **1** a **2147483648**.

Vedi anche il tag **<EMBED>**.

## Tutti i tag

### Il tag **<BIG>**

L'elemento contenitore

**<BIG> ... </BIG>**

indica al browser di visualizzare un testo con un carattere di dimensione maggiore rispetto al normale tipo di carattere.

Ecco un esempio di utilizzo del tag:

Questa è una parola scritta con carattere più grande.

e questo è il codice che lo genera:

```
<BIG>Questa</BIG> è una parola scritta con un carattere più grande.
```

L'effetto è più facile da osservare se il testo è inserito all'interno del normale testo; il tag è essenzialmente equivalente al tag **<FONT>** nel quale l'attributo **SIZE** è impostato al valore **"+1"**.

## Browser

Mosaic 3.0 non riconosce questo tag. Tutti gli altri browser lo riconoscono correttamente.

Vedi anche il tag **<SMALL>**.

## Tutti i tag

### Il tag **<BLINK>**

L'elemento contenitore

**<BLINK> ... </BLINK>**

è utilizzato per rendere lampeggiante il testo contenuto nei tag di apertura e chiusura.

Non è un tag standard HTML ed il suo utilizzo è disapprovato dalla maggior parte dei programmatori. Il tag si applica solo agli elementi di testo e non alle immagini; nel caso si voglia creare una immagine lampeggiante, essa dovrà essere realizzata mediante una GIF animata. Una tabella inclusa nei tag **<BLINK>** e **</BLINK>** non lampeggerà.

Non c'è modo di controllare la velocità del lampeggiamento, ed un testo che lampeggia continuamente può risultare molto fastidioso.

Ecco un esempio di utilizzo del tag:

```
<BLINK>Questo testo</BLINK> sta lampeggiando.
```

### Browser

Solo Netscape Navigator è in grado di gestire questo tag. Utenti di altri browser possono provocare brutti fastidi ai navigatori che utilizzano Netscape Navigator racchiudendo l'intera pagina all'interno dei tag **<BLINK> ... </BLINK>**.

## Tutti i tag

### Il tag **<BLOCKQUOTE>**

L'elemento contenitore di blocchi

**<BLOCKQUOTE>** ... **</BLOCKQUOTE>**

è usato per incorporare una porzione di testo che abbia una certa distinzione dal testo normale.

La distinzione viene generalmente ottenuta mediante una diversa indentatura del testo selezionato rispetto al resto della pagina. Il tag causa una interruzione di paragrafo, e di conseguenza una interlinea di una linea e mezza tra il testo della pagina e i paragrafi selezionati.

Ecco un esempio di utilizzo del tag:

La seguente citazione è presa dal V canto dell'*Inferno* della Divina Commedia di Dante Alighieri

Vien dietro a me, e lascia dir le genti:  
sta come torre ferma, che non crolla  
già mai la cima per soffiare di venti;

e questo è il codice che lo genera:

```
<BLOCKQUOTE>Vien dietro a me, e lascia dir le genti:  
sta come torre ferma, che non crolla  
già mai la cima per soffiare di venti; </BLOCKQUOTE>
```

Una ulteriore indentatura può essere applicata inserendo un secondo **<BLOCKQUOTE>** a cascata, dentro l'altro.

Questo che segue è lo stesso passo di Dante, con una indentatura doppia.

Vien dietro a me, e lascia dir le genti:  
sta come torre ferma, che non crolla  
già mai la cima per soffiare di venti;

e questo è il codice che lo genera:

```
<BLOCKQUOTE><BLOCKQUOTE>Vien dietro a me, e lascia dir le genti:  
sta come torre ferma, che non crolla  
già mai la cima per soffiare di venti; </BLOCKQUOTE></BLOCKQUOTE>
```

Lo standard HTML non specifica alcun attributo per il tag **<BLOCKQUOTE>**, tuttavia alcuni browser riconoscono a questo tag

gli stessi attributi del tag **<P>**, al quale facciamo riferimento per la descrizione di tali attributi.

## Browser

Internet Explorer inserisce una interlinea doppia dopo la chiusura del tag **</BLOCKQUOTE>**, mentre Netscape Navigator inserisce una interlinea di una riga e mezza. In entrambi i casi l'apertura di **<BLOCKQUOTE>** a cascata genera una sola riga tra i paragrafi selezionati ed i successivi.

Il solo Internet Explorer riconosce l'attributo **ALIGN**.

Vedi anche i tag **<ADDRESS>**, **<LISTING>**, **<PLAINTEXT>**, **<PRE>**, **<XMP>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag <BODY>

L'elemento contenitore

**<BODY> ... </BODY>**

è usato per definire il corpo principale di un documento HTML, distinguendolo dagli elementi che fanno parte della intestazione della pagina.

Il testo, le immagini e gli elementi grafici contenuti tra i tag **<BODY>** e **</BODY>** verranno visualizzati nella finestra del browser.

Molti browser supportano un gran numero di attributi per questo tag.

I più comuni sono i seguenti:

#### 1. **BGCOLOR**

Il valore di questo attributo è un colore che verrà usato come colore di fondo del testo visualizzato. Il suo utilizzo oculato può migliorare notevolmente l'aspetto e la lettura del testo, e va usato sempre in considerazione del colore o dei colori del testo.

#### 2. **BACKGROUND**

Il valore di questo attributo è l'URL di una immagine che può essere usata come sfondo per il testo visualizzato; l'immagine verrà replicata tante volte quanto basta per coprire tutta l'area visualizzabile. Va usato con attenzione, in quanto può accadere di utilizzare una immagine che distragga la lettura o addirittura renda il testo illeggibile. Normalmente il testo e lo sfondo scorrono insieme.

#### 3. **BGPROPERTIES**

Questo non è un attributo standard HTML 3.2 e viene riconosciuto solo da Internet Explorer. L'attributo controlla lo scorrimento della immagine di sfondo e ha due possibili valori, **FIXED** oppure **SCROLL**. Il primo permette al testo di scorrere sopra una immagine fissa, mentre col secondo, predefinito, il testo e l'immagine scorrono insieme.

#### 4. **TEXT**

Il valore di questo attributo è un colore che viene utilizzato per visualizzare il testo all'interno della pagina. Affinché sia

efficace è opportuno scegliere un colore in forte contrasto con il colore dello sfondo e dei collegamenti ipertestuali.

#### 5. **LINK**

Il valore di questo attributo è un colore che viene utilizzato per visualizzare il testo racchiuso all'interno del tag **<A>** e **</A>** con l'attributo **HREF**. Molti browser utilizzano il colore blue come colore predefinito per i collegamenti ipertestuali. Non c'è modo di controllare la sottolineatura dei collegamenti ipertestuali in HTML, ed alcuni browser non utilizzano la sottolineatura per i link, così, la scelta di un colore diverso è molto utile per distinguere un link dal testo generico. Mediante l'utilizzo dei fogli di stile, è possibile modificare la sottolineatura dei collegamenti ipertestuali.

#### 6. **VLINK**

Questo attributo stabilisce il colore dei collegamenti ipertestuali relativi a pagine già visitate. Il colore predefinito per un collegamento ipertestuale è generalmente il viola.

#### 7. **ALINK**

Il valore di questo attributo è un colore che viene utilizzato per visualizzare i collegamenti ipertestuali attivi, ossia il colore di un link quando gli si passa sopra con il puntatore del mouse. Il colore predefinito è il rosso e la sua visualizzazione è ovviamente momentanea.

#### 8. **TOPMARGIN**

Questo non è un attributo standard HTML 3.2 ed è gestito solo da Internet Explorer. Esso specifica un margine in numero di punti, all'inizio ed alla fine della pagina, tra il testo e la finestra del browser.

#### 9. **LEFTMARGIN**

Questo attributo non standard HTML 3.2 stabilisce un margine orizzontale, in pixel, tra il testo e il bordo della finestra del browser. L'attributo viene considerato solo da Internet Explorer.

## Scelta dei colori

C'è una grande scelta di colori possibili, sebbene l'utilizzo del blu per un testo che non sia un collegamento ipertestuale potrebbe creare confusione in persone abituate ad utilizzarlo per i link. Se



nessuno degli attributi per i colori è impostato, per questi verranno utilizzati i colori predefiniti dal browser. È importante definire in qualche modo il colore del testo generico e dei collegamenti, e non lasciare al caso o alle impostazioni dell'utente, che possono non essere prevedibili.

Vedi anche il tag **<HEAD>**.

## Tutti i tag

### Il tag **<BR>**

L'elemento HTML

**<BR>**

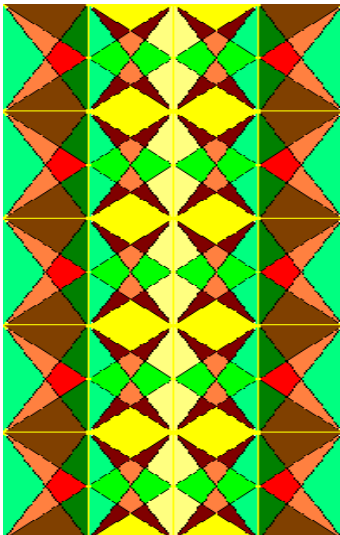
inserisce una interruzione di riga e forza il testo ad andare a capo, che sia richiesto o no dalla pagina. Non inserisce una riga vuota, a meno che non sia posto dopo la chiusura di un paragrafo, una tabella o una lista.

Vediamo un esempio di utilizzo del tag, mettendo QUI una interruzione di riga.

e questo è il codice che lo genera:

Vediamo un esempio di utilizzo del tag, mettendo QUI  
<BR> una  
interruzione di riga.

Il tag **<BR>** può avere l'attributo **CLEAR** con valore uno tra **left**, **both** e **right**. Questo attributo implica che la riga di testo successiva verrà visualizzata dopo l'appropriato margine per gli elementi allineati a sinistra o destra. Ciò è molto utile con le immagini.



L'immagine alla sinistra di questo testo è stata posizionata con l'allineamento a sinistra, mediante l'attributo **ALIGN** del tag **<IMG>**; il testo ovviamente sta a destra dell'immagine.

Qui siamo andati a capo con un nuovo paragrafo **<P>**, e il testo continua a stare a destra dell'immagine, e ciò accadrà finché l'immagine non sarà terminata. Alla fine di questa frase andremo a capo con una semplice interruzione di riga utilizzando l'attributo **CLEAR=left**.

Questo testo apparirà sotto l'immagine, sebbene appartenga allo stesso paragrafo.

## Browser

Tutti i browser gestiscono il tag nella sua forma base. Sia Mosaic che Ariadna hanno un comportamento non prevedibile in presenza dell'attributo **CLEAR=left**; Mosaic perde parte del testo, mentre Ariadna dispone il testo sul margine estremo destro della pagina, una parola per riga.

Vedi anche il tag **<P>**.

## Tutti i tag

### Il tag **<CAPTION>**

L'elemento contenitore HTML

**<CAPTION> ... </CAPTION>**

è usato all'interno di una tabella per realizzare una intestazione della tabella.

Il testo contenuto è usato per l'intestazione della tabella e risulta centrato rispetto al centro della tabella.

Il tag ha un solo attributo **ALIGN** che ammette i valori **top** e **bottom** e che consentono di visualizzare l'intestazione in alto o in basso rispetto alla tabella. Il valore predefinito è **top**.

Vediamo un esempio di utilizzo del tag:

Regione	Capoluogo
Sicilia	Palermo
Lazio	Roma
Campania	Napoli
Piemonte	Torino

Alcuni capoluoghi italiani

e questo è il codice che lo genera:

```
<TABLE BORDER=1>
<CAPTION ALIGN=bottom>Alcuni capoluoghi italiani</CAPTION>
<TR><TH width=50%>Regione</TH><TH
width=50%>Capoluogo</TH></TR>
<TR><TD>Sicilia</TD><TD>Palermo</TD></TR>
<TR><TD>Lazio</TD><TD>Roma</TD></TR>
<TR><TD>Campania</TD><TD>Napoli</TD></TR>
<TR><TD>Piemonte</TD><TD>Torino</TD></TR>
</TABLE>
```

## Browser

Tutti i browser visualizzano il tag allo stesso modo.

Vedi anche i tag **<TABLE>**, **<TD>**, **<TH>**, **<TR>** e le note generali sulle tabelle.

## Tutti i tag

### Il tag **<CENTER>**

L'elemento contenitore

**<CENTER> ... </CENTER>**

allinea al centro tutto ciò che viene incluso tra i due tag di apertura e chiusura, inclusi tabelle, paragrafi e immagini.

Ha lo stesso identico effetto del tag **<DIV ALIGN=center>** e non ha attributi.

### Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<DIV>** e **<P>**.

## Tutti i tag

### Il tag **<CITE>**

L'elemento contenitore

**<CITE> ... </CITE>**

è utilizzato per citazioni e riferimenti.

Il vero significato del tag non è molto chiaro dagli standard HTML, e così esso viene usato raramente. L'effetto di includere un blocco all'interno di un elemento **<CITE>** non è ben specificato e i browser visualizzano il testo contenuto con un carattere avente uno stile diverso, generalmente in corsivo. Il layout del testo non viene modificato e i caratteri speciali HTML andranno inseriti tramite i loro codici.

Ecco un esempio di utilizzo del tag:

`1==1 1<=2 1>=0`

e questo è il codice che lo genera:

```
<cite>
1==1
1<=2
1>=0
</CITE>
```

Notiamo che non è la stessa visualizzazione del testo preformattato ottenuto con **<PRE>**.

Può essere usato anche all'interno di un paragrafo, come ad esempio: *un esempio* di testo.

### Browser

Tutti i browser gestiscono questo tag visualizzando il testo con un carattere proporzionale corsivo.

Mosaic 3.0 e Ariadna 1.2 inoltre usano un carattere diverso all'interno di celle di tabella. Opera 2.12 non utilizza caratteri diversi se il tag è incluso in un paragrafo specificato con il tag **<P>**.

**NOTA.** In alcuni browser questo tag ha lo stesso comportamento degli altri appena menzionati, e l'effetto è identico. Le differenze sono puramente storiche.

Vedi anche i tag **<CODE>**, **<DFN>**, **<KBD>**, **<VAR>**, **<SAMP>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<CODE>**

L'elemento contenitore

**<CODE> ... </CODE>**

è utilizzato per porzioni di codici estratti da programmi e testi.

Il tag visualizza il testo, con carattere monospazio, esattamente come è stato inserito, ma non inserisce le interruzioni di riga se queste sono presenti nel testo. I caratteri speciali HTML devono essere scritti tramite i loro codici. L'effetto si applica anche a blocchi di testo. Il codice HTML presente tra i due tag viene ignorato e riportato esattamente come trovato, ad eccezione del comando di markup **<BR>**, che serve ad introdurre una interruzione di riga.

Ecco un esempio di utilizzo del tag:

Line 1 <A HREF=titolo.html>Titolo</A> Line 2 Line 3

e questo è il codice che lo genera:

```
<CODE>
Line 1
<A HREF=titolo.html>Titolo</A>
Line 2
Line 3
</CODE>
```

Notiamo che non è la stessa visualizzazione del testo preformattato ottenuto con **<PRE>**.

Può essere usato anche all'interno di un paragrafo, come ad esempio: un esempio di testo.

## Browser

Ci sono alcune significative differenze tra un browser e un altro. Tutti i browser utilizzano un carattere di tipo monospazio (non proporzionale). Mosaic utilizza un carattere in grassetto, Internet Explorer, Netscape Navigator e Ariadna utilizzano un carattere normale ma leggermente più piccolo, Opera utilizza un carattere delle stesse dimensioni del normale testo.

Anche l'inclusione di interi blocchi produce effetti diversi tra un browser e l'altro. Mosaic, Internet Explorer e Ariadna usano un

font diverso per gli elementi contenuti dentro tabelle, Opera smette di utilizzare un carattere diverso (monospazio) non appena trova il termine del primo blocco.

**NOTA.** In alcuni browser questo tag ha lo stesso comportamento degli altri appena menzionati, e l'effetto è identico. Le differenze sono puramente storiche.

Vedi anche i tag **<CITE>**, **<DFN>**, **<KBD>**, **<SAMP>**, **<VAR>** e le note generali sugli stili di testo.



## Tutti i tag

### Il tag **<DD>**

L'elemento contenitore

**<DD> ... </DD>**

è usato per inserire una definizione all'interno di una lista di definizioni (**<DL>**). Una definizione può contenere blocchi di testo nonché intestazioni e collegamenti ipertestuali.

Il tag di chiusura **</DD>** può non essere messo e viene automaticamente chiuso se si inserisce un nuovo tag **<DT>** oppure **</DL>**.

Vedi anche i tag **<DL>**, **<DT>**, **<OL>**, **<UL>** e le note generali sulle liste.

## Tutti i tag

### Il tag **<DFN>**

L'elemento contenitore

**<DFN> ... </DFN>**

è utilizzato per specificare definizioni all'interno di porzioni di programmi. Il significato non è completamente chiaro dallo standard HTML ed il tag è usato raramente.

Il tag visualizza il testo con un carattere diverso esattamente come è stato inserito, ma la formattazione non viene rispettata, ossia non inserisce le interruzioni di riga o altro se queste sono presenti nel testo. I caratteri speciali HTML devono essere scritti tramite i loro codici. L'effetto si applica anche a blocchi di testo.

Ecco un esempio di utilizzo del tag:

*1==1 1<=2 1>=0*

e questo è il codice che lo genera:

```
<DFN>
1==1
1<=2
1>=0
</DFN>
```

Notiamo che non è la stessa visualizzazione del testo preformattato ottenuto con **<PRE>**.

Può essere usato anche all'interno di un paragrafo, come ad esempio: *un esempio* di testo.

## Browser

Mosaic e Netscape Navigator non visualizzano questo tag. Ariadna, Internet Explorer e Opera presentano il testo con carattere corsivo.

**NOTA.** In alcuni browser questo tag ha lo stesso comportamento degli altri appena menzionati, e l'effetto è identico. Le differenze sono puramente storiche.

Vedi anche i tag **<CITE>**, **<CODE>**, **<KBD>**, **<SAMP>**, **<VAR>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<DIR>**

L'elemento contenitore

**<DIR> ... </DIR>**

è usato raramente per costruire delle liste ed ha lo stesso effetto del tag **<UL>**.

Microsoft Word utilizza **<DIR>** multipli annidati senza il tag **<LI>** per controllare l'indentatura del testo nel salvare un documento di Office in formato HTML.

Il tag produce una lista a più colonne ma nessun browser sembra assecondare questo comportamento, ed è un peccato. L'unico modo per ottenere una visualizzazione a più colonne è tramite una tabella che sono più difficili da gestire rispetto ad una semplice lista.

La lista qui sotto è facile da modificare, ma sarebbe sicuramente migliore in un formato a più colonne. Si potrebbe realizzare con una tabella ma ciò potrebbe risultare scomodo per un testo che si incolonna male.

Ecco un esempio:

- ? Abruzzo
- ? Basilicata
- ? Calabria
- ? Campania
- ? Emilia Romagna
- ? Friuli Venezia Giulia
- ? Lazio
- ? Liguria
- ? Lombardia
- ? Marche
- ? Molise
- ? Piemonte
- ? Puglia
- ? Sardegna
- ? Sicilia
- ? Toscana
- ? Trentino Alto Adige
- ? Umbria
- ? Valle D'Aosta
- ? Veneto

In generale il tag si comporta esattamente come il tag **<UL>**.

## Browser

Mosaic e Internet Explorer visualizzano la lista con gli stessi simboli utilizzati per il tag **<UL>** in tutti i livelli. Ariadna e Netscape Navigator utilizzano gli stessi simboli usati per una lista di primo livello. Opera visualizza la lista senza simboli distintivi. Tutti i browser visualizzano la lista in una sola colonna. L'attributo **COMPACT** non ha effetto.

Vedi anche i tag **<MENU>**, **<UL>** e le note generali sulle liste.

## Tutti i tag

### Il tag **<DIV>**

L'elemento contenitore

**<DIV> ... </DIV>**

estende gli attributi di allineamento del tag **<P>** ad un gruppo di blocchi come paragrafi, tabelle e immagini. Tutti i blocchi all'interno del contenitore **<DIV>** eredita gli attributi di allineamento del tag, a meno che, ovviamente, non abbia un proprio attributo **ALIGN**.

Il tag ha il solo attributo **ALIGN** che riconosce i quattro possibili valori **left** (predefinito), **center**, **right** e **justify**. Non è chiaro dallo standard HTML se il tag **<DIV>** condivida la proprietà di chiusura implicita propria dei tag **<P>** e **<LI>**. La presenza di un tag **<DIV>** implica un tag **</DIV>** alla fine del blocco cui si vuole dare l'allineamento, ma lo standard non specifica se l'apertura di un nuovo tag **<DIV>** implica la fine del precedente blocco.

I paragrafi seguenti illustrano l'effetto di questi allineamenti.

Notiamo come la presenza del tag **<P>** annulli il tag **<DIV>** inserendo una interlinea e forzando l'allineamento.

Questo breve testo è allineato al destra.

Questo è allineato al centro. L'esempio mostra il comportamento del browser con testi lunghi, che coprono più righe.

Questo invece è un nuovo paragrafo, giustificato, ed inserito senza chiudere il tag **<DIV>** del blocco precedente. Questo testo è allineato ad entrambi i lati della finestra del browser.

Il blocco successivo tiene l'allineamento al centro del testo immediatamente precedente al tag **<P>**.

Qui il testo riprende normalmente.

### Browser

Mosaic 3 e Ariadna non riconoscono il tag. Tutti gli altri browser visualizzano il tag alla stessa maniera.

Vedi anche i tag **<CENTER>** e **<P>**.

## Tutti i tag

### Il tag **<DL>**

L'elemento HTML

**<DL>**

è usato per inserire una lista di definizioni. I vari elementi della lista saranno rappresentati da due parti:

1. Un titolo indicato dal tag **<DT>**
2. Una definizione indicata dal tag **<DD>**

L'elemento di definizione viene indentato di circa 80 pixel rispetto al titolo.

Vediamo un esempio di utilizzo del tag:

Campania

Capoluogo: Napoli;  
Superficie: 13595 km<sup>2</sup>; Economia: Agricoltura, Pesca,  
Turismo, Industria alimentare.

Toscana

Capoluogo: Firenze;  
Superficie: 22993 km<sup>2</sup>; Economia: Industria siderurgica,  
Viticoltura, Turismo.

Possiamo notare che andando a capo col tag **<BR>** il testo mantiene l'allineamento e l'indentatura della riga superiore.

Le liste di definizioni possono anche essere annidate, ponendo tag **<DL>** uno dentro l'altro.

Campania

Capoluogo:  
Napoli;  
Superficie:  
13595 km<sup>2</sup>;  
Economia:  
Agricoltura,  
Pesca,  
Turismo,  
Industria alimentare.

Toscana

Capoluogo:  
Firenze;

Superficie:  
22993 km<sup>2</sup>;  
Economia:  
Industria siderurgica,  
Viticoltura,  
Turismo.

e questo è il codice utilizzato:

```
<DL>
<DT>Campania
<DD>Capoluogo: Napoli;<BR>
<DD>Superficie: 13595 km<SUP>2</SUP>; Economia: Agricoltura,
Pesca, Turismo, Industria alimentare.
<DT>Toscana
<DD>Capoluogo: Firenze;<BR>
<DD>Superficie: 22993 km<SUP>2</SUP>; Economia: Industria
siderurgica, Viticoltura, Turismo.
</DL>

<DL>
<DT>Campania
<DD>
<DL>
<DT>Capoluogo:
<DD>Napoli;
<DT>Superficie:
<DD>13595 km<SUP>2</SUP>;
<DT>Economia:
<DD>Agricoltura,
<DD>Pesca,
<DD>Turismo,
<DD>Industria alimentare.
</DL>
<DT>Toscana
<DD>
<DL>
<DT>Capoluogo:
<DD>Firenze;
<DT>Superficie:
<DD>22993 km<SUP>2</SUP>;
<DT>Economia:
<DD>Industria siderurgica,
<DD>Viticoltura,
<DD>Turismo.
</DL>
</DL>
```

Il tag **<DL>** prevede l'attributo **COMPACT** che, a detta dello standard HTML 3.2, dovrebbe indicare al browser di visualizzare la

lista in uno stile più compatto, ma ogni browser si comporta diversamente con questo attributo.

Lo standard HTML inoltre non include alcuna specifica di comportamento in presenza di paragrafi o nuove righe all'interno della lista e di modifiche dell'indentatura.

## Browser

Tutti i browser visualizzano una indentatura extra in corrispondenza degli elementi **<DD>**, ma l'entità dell'indentatura cambia da browser a browser. Ariadna, Netscape e Opera indentano di una quantità pari circa a 6 caratteri; Internet Explorer e Mosaic qualche carattere in meno. Solo Ariadna e Netscape riconoscono l'attributo **COMPACT**.

Tutti i browser inseriscono una interlinea singola prima e dopo la lista. Opera inoltre inserisce una interlinea anche prima di ciascun titolo **<DT>**.

Vedi anche i tag **<DD>**, **<DT>**, **<OL>**, **<UL>** e le note generali sulle liste.



## Tutti i tag

### Il tag **<DT>**

L'elemento HTML

#### **<DT>**

è usato per presentare il titolo o il termine oggetto di una lista di definizioni (**<DL>**). Blocchi di testo (come liste, immagini e paragrafi) non possono essere incluse in una lista di definizioni.

Vedi anche i tag **<DD>**, **<DL>**, **<OL>**, **<UL>** e le note generali sulle liste.

## Tutti i tag

### Il tag **<EM>**

L'elemento contenitore

**<EM> ... </EM>**

visualizza il testo enfattizzato in modo diverso.

Lo standard HTML definisce questo come un tag logico al contrario di **<I>testo</I>** che è un tag fisico.

*Questo è lo stile <EM>.*

La differenza tra i due tag consiste nel fatto che il tag fisico impone al browser di visualizzare il testo in corsivo, mentre il tag logico indica al browser di mostrare il testo con un carattere diverso ed enfattizzato lasciandogli la possibilità di scegliere il modo (possono anche essere usati la sottolineatura o colori differenti).

## Browser

Tutti i browser visualizzano il testo usando un carattere corsivo.

Vedi anche i tag **<I>**.

## Tutti i tag

### Il tag **<EMBED>**

L'elemento HTML

#### **<EMBED>**

permette di integrare sottofondi sonori ed è raccomandato in molti casi. Tuttavia non viene riconosciuto da molti browser. In tal caso si utilizza il tag **<NOEMBED> ... </NOEMBED>** per sostituirlo.

È bene utilizzare sempre il tag **<NOEMBED>** dopo il tag **<EMBED>** per supplire la mancanza da parte dei browser non aggiornati.

Il tag **<EMBED>** può venir messo in qualunque posto all'interno del corpo, BODY, del documento HTML e non ha tag di chiusura.

Lo standard HTML 4.0 definisce molti attributi per questo tag, per specificarne le modalità di visualizzazione e, non ultimo, il file da riprodurre.

#### 1. **SRC**

Il valore di questo attributo è l'URL del file musicale da eseguire.

#### 2. **ALIGN**

Questo attributo definisce l'allineamento orizzontale dell'elemento che viene visualizzato nella pagina HTML per mostrare i pulsanti di controllo. I valori possibili sono **"left"** (predefinito), **"center"** e **"right"**.

#### 3. **WIDTH** e **HEIGHT**

Impostano, se visualizzato, le dimensioni in pixel dell'immagine dell'elemento con i pulsanti di controllo.

#### 4. **BORDER**

Imposta per l'elemento di controllo un bordo delle dimensioni specificate. Il valore atteso è il numero di pixel dello spessore di questo bordo.

#### 5. **AUTOLOAD** e **AUTOSTART**

Richiedono il caricamento e la riproduzione automatica del file multimediale. Se attivi ricevono il valore **"true"**, che in ogni caso è predefinito. Al contrario, se il valore è **"false"**, il visitatore dovrà attivare manualmente un pulsante che avvierà la riproduzione del file.

## 6. LOOP

Questo attributo imposta il numero di riproduzioni che vogliamo far eseguire al file musicale. I valori **"-1"** o **"infinite"** impostano un ciclo continuo.

## 7. VOLUME

Questo attributo imposta il volume iniziale del suono riprodotto. I valori ammessi sono compresi tra 0 e 100 e l'utente può cambiarlo se ha accesso al pannello con i pulsanti di controllo.

## 8. HIDDEN

Offre la possibilità di nascondere il pannello di controllo. I valori possibili sono **"true"** o **"false"**.

Ecco un esempio di di utilizzo del tag:

```
<EMBED SRC="prettywoman.mid" WIDTH=150 HEIGHT=60  
AUTOSTART="false" LOOP="-1">  
<NOEMBED>Il tuo browser non supporta il sottofondo musicale per  
questa pagina.</NOEMBED>
```

## Browser

Tutti i browser recenti visualizzano questo tag allo stesso modo. A seconda delle impostazioni del browser a volte sarà necessario installare e configurare alcuni plug in multimediali, ad esempio di tipo "Windows Media" oppure "Real Audio" o "QuickTime".

## Tutti i tag

### Il tag **<FONT>**

L'elemento contenitore

**<FONT> ... </FONT>**

è un tag usato per cambiare il modo in cui è visualizzato il testo tra il tag di apertura e di chiusura.

L'effetto è controllato da numerosi attributi:

#### 1. **SIZE**

Il valore di questo attributo è un numero con segno che abitualmente va da **-6** a **+7**. La misura del tag corrente viene cambiata. Le unità di misura della dimensione del font sono arbitrarie, tutto quello che si può dire in questa sede è che **+5** è considerato minore di **+6** ecc. La dimensione del file può anche essere impostata ad un particolare valore omettendo il segno.

Ecco un esempio di utilizzo del tag:

Questa è dimensione **+2** e questa è **-2**

E qui di seguito è mostrato il codice che permette la realizzazione di questo testo:

```
<FONT SIZE=+2>Questa è dimensione +2</FONT>
e <FONT SIZE=-2>Questa è -2</FONT>
```

Il valore con segno cambia la dimensione del carattere bel numero di passi indicati rispetto alla dimensione corrente stabilita dal tag **<BASEFONT>** e dalle correnti impostazioni del browser. Se il valore corrente è già grande allora ulteriori ingrandimenti potrebbero non avere effetti visibili. Un valore senza segno imposta la dimensione del carattere ad un particolare valore, relativo solo alle impostazioni del browser.

#### 2. **COLOR**

Questo attributo è usato per cambiare il colore del testo visualizzato. Il valore dell'attributo è codificato nella solita maniera, ossia in esadecimale oppure indicando il nome del colore.

Ecco un **esempio** di utilizzo del tag.

E qui di seguito è mostrato il codice che permette la realizzazione di tale scritta colorata:

```
Ecco un <FONT COLOR=FF0000>es</FONT><FONT
COLOR=00FF00>em</FONT><FONT COLOR=0000FF>pio</FONT> di
utilizzo del tag.
```

### 3. FACE

Molti browser permettono di specificare un particolare tipo di carattere. Ciò funziona ovviamente solo sui browser che sono in grado di interpretare l'attributo **FACE** e che supportano il tipo di carattere indicato. L'attributo **FACE** non fa parte dello standard HTML 3.2, ma è stato inserito nello standard 4.0.

Questo testo è scritto in Courier, mentre questo è scritto nel tipo di carattere predefinito.

E qui di seguito è mostrato il codice che permette la visualizzazione di questo esempio:

```
<FONT FACE="courier">Questo testo è scritto in Courier</FONT>,
mentre questo è scritto nel tipo di carattere predefinito.
```

Notiamo che il nome del tipo di carattere non è case-sensitive e si può scrivere in minuscolo o in maiuscolo. Per comodità e per uniformarci alla maggior parte dei browser abbiamo utilizzato un carattere "Times New Roman" come carattere 'predefinito'.

La tabella seguente illustra alcuni esempi di utilizzo di tipi di carattere diversi e mostra le differenze tra un tipo e l'altro:

FACE	Esempio
Predefinito (Times)	abcdefghijklmnopqrstuvwxyz
Courier	abcdefghijklmnopqrstuvwxyz
Arial	abcdefghijklmnopqrstuvwxyz
Verdana	abcdefghijklmnopqrstuvwxyz
Impact	abcdefghijklmnopqrstuvwxyz
AsdfAsdf	abcdefghijklmnopqrstuvwxyz

Non esistendo alcun carattere chiamato AsdfAsdf, per l'ultima riga di tabella viene utilizzato il tipo di carattere predefinito.

Alcuni browser possono permettere di indicare più di un tipo di carattere, in modo che se il primo non viene trovato si prova a visualizzare il secondo tipo della lista, e così via.

Questo è utile in quei browser che non supportano alcuni tipi di carattere. Ad esempio Netscape conosce Helvetica ma non Arial, ed essendo i due tipi molto simili, scrivendo **FACE="Arial, Helvetica"** possiamo visualizzare entrambi.

## Annidamento

Il tag **FONT** può essere annidato, come nell'esempio che segue:

Ciao

E questo è il codice che lo genera:

```
<FONT SIZE=+2>C<FONT COLOR="FF0000">i<FONT
FACE="Times">a</FONT></FONT><FONT
COLOR="00FF80">o</FONT></FONT>
```

Il codice va ovviamente scritto tutto su una riga, altrimenti eventuali ritorni a capo vengono interpretati dal browser come spazi bianchi tra una lettera e l'altra.

Se i tag intermedi **</FONT>** vengono omessi l'effetto di ciascun tag aperto si ripercuote sui successivi con un annidamento progressivo ed il tag finale chiuderebbe un solo livello di annidamento.

## Browser

Tutti i browser visualizzano questo tag allo stesso modo, ad eccezione di Mosaic 3.0 che lo ignora completamente. Opera 2.0 non riconosce l'attributo **FACE**.

Vedi anche il tag **<BASEFONT>**.

## Tutti i tag

### Il tag **<FORM>**

L'elemento HTML

**<FORM> ... </FORM>**

definisce un modulo per un inserimento interattivo di informazioni attraverso un meccanismo di tipo CGI. Tra i tag **<FORM>** e **</FORM>** ci sarà un certo numero di caselle da riempire ed inviare come input. L'insieme di caselle costituisce il modulo.

Una volta che le varie caselle del modulo sono state riempite e il pulsante "Submit" ("Invia") è stato premuto, allora i valori inseriti vengono trasmessi al server, nel quale una interfaccia CGI gestisce le informazioni e genera una eventuale risposta.

Il tag **<FORM>** ha tre attributi significativi, **METHOD**, **ACTION** e **ENCTYPE**.

#### 1. **METHOD**

Per un uso semplice l'attributo **METHOD** può essere impostato a **GET** o **POST**, e controlla il modo in cui i dati inseriti vengono trasmessi al programma server. Mediante il metodo **GET** i dati vengono inviati tramite una variabile di ambiente chiamata `QUERY_STRING`. Mediante il metodo **POST** i dati vengono inviati al programma tramite il suo standard input.

Molti autori suggeriscono che il metodo **POST** sia preferibile per il fatto che alcuni host possono ridurre la dimensione delle variabili di ambiente e soprattutto per il fatto che i server WWW hanno un accesso pubblico e le richieste di tipo **GET** possono essere intercettate e/o costruite artificialmente, mentre le **POST** no.

#### 2. **ACTION**

Il valore di questo attributo indica il comando che sarà effettuato quando il modulo sarà stato riempito e sarà stato premuto il pulsante "Invia". Il valore è generalmente un indirizzo, se questo è il nome di un file eseguibile oppure un'altra pagina web.

Ecco un esempio di utilizzo del tag:

```
<FORM ACTION="ricerca.php" METHOD="POST">
```



### 3. ENCTYPE

Questo attributo determina il modo in cui le informazioni saranno codificate quando vengono inviate al programma CGI.

Il valore predefinito di questo attributo è "**application/x-www-form-urlencoded**" e non necessita di modifiche a meno che non ci sia un tag **<INPUT>** con l'attributo **TYPE** impostato su "**FILE**", per il quale **ENCTYPE** deve avere il valore di "**multipart/form-data**".

Vedi anche i tag **<INPUT>**, **<SELECT>**, **<TEXTAREA>** e le note sull'interfaccia CGI.

## Tutti i tag

### Il tag **<FRAME>**

Il tag

**<FRAME>**

definisce un frame in cui sarà caricato un documento HTML.

Questo tag non è previsto nello standard HTML 3.2 e può essere usato solo all'interno di un frame contenitore **<FRAMESET>**; la dimensione della finestra sarà definita dal tag **<FRAMESET>**.

Il tag ha 6 attributi.

#### 1. **SRC**

Il valore di questo attributo è l'indirizzo della pagina da caricare all'interno della finestra definita dal tag. La visualizzazione della pagina è gestita dal codice HTML presente all'interno del documento che si va ad aprire, e nessun attributo legato a testo o immagini del documento chiamante può essere esportato alla pagina caricata nel frame.

Se l'attributo viene omesso viene caricata una pagina vuota. Può successivamente venirci caricata una pagina se si clicca su un collegamento ipertestuale avente quel frame come target.

#### 2. **NAME**

Questo attributo definisce il nome del frame, e fornisce un meccanismo mediante il quale il browser è in grado di riconoscere la finestra in questione e così aprire una pagina all'interno di essa piuttosto che altrove.

Il nome non può iniziare col trattino in basso, e ci sono quattro nomi speciali che definiscono alcuni particolari frame:

- **\_blank** permette di aprire una pagina in una finestra separata del browser sul computer host.
- **\_self** è il frame da cui si parte; la pagina si aprirà nello stesso frame in cui è presente il link; equivale ad omettere l'attributo **TARGET** nel tag **<A>**.
- **\_parent** indica il frame che ha creato il frame in cui è presente il link; nella gerarchia di frame, indica il frame di livello direttamente superiore.
- **\_top** indica l'intera finestra del browser.

### 3. **MARGINWIDTH** e **MARGINHEIGHT**

Il valore di questi due attributi definisce lo spazio minimo in pixel tra il bordo del frame ed il testo contenuto nel frame stesso.

### 4. **SCROLLING**

Questo attributo può assumere i valori **"yes"**, **"no"** o **"auto"** e fa in modo che il testo contenuto all'interno del frame possa scorrersi mediante le barre di scorrimento poste a sinistra del frame e in basso. Se è omesso si intende **"auto"** e le barre di scorrimento sono presenti e attive solo se il testo supera la dimensione del frame che lo ospita.

### 5. **NORESIZE**

L'attributo **NORESIZE** non ha alcun valore associato e se presente blocca i bordi del frame in modo che non possano essere ridimensionati dall'utente mediante trascinamento col mouse.

Vedi anche il tag **<FRAMESET>** e le note generali sui frame.

## Tutti i tag

### Il tag **<FRAMESET>**

Il tag

#### **<FRAMESET>**

è usato per definire una divisione in frame della pagina corrente e non è previsto nello standard HTML 3.2.

Il tag ha due attributi mutuamente escludentesi che indicano come la pagina debba venire suddivisa, più due attributi che ne controllano i margini:

#### 1. **COLS**

Indica una divisione in colonne e specifica il numero di colonne e, in pixel oppure in percentuale dello spazio disponibile oppure in larghezze relative, l'ampiezza di ciascuna colonna. Il valore dell'attributo è una lista di numeri separati da virgola che indicano le dimensioni dei singoli frame. Il numero di questi valori rappresenterà il numero di frame creati. I primi due metodi sono familiari e sono gli stessi utilizzati per ridimensionare tabelle e immagini. L'ultimo metodo permette di indicare la dimensione come multiplo di una quantità  $n$ , nella forma di  $n^*$ . Ad esempio la dicitura **COLS="1\*, 2\*, 1\*"** ottiene 3 colonne in cui la colonna centrale è il doppio delle altre due. In questo caso l'"1" può essere omesso.

#### 2. **ROWS**

È uguale all'attributo precedente e controlla una suddivisione in righe orizzontali del frame corrente.

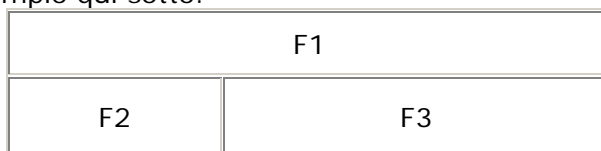
#### 3. **FRAMESPACING**

Imposta lo spazio in pixel tra un frame e l'altro.

#### 4. **FRAMEBORDER**

Imposta lo spessore del bordo dei frame.

Chiaramente **COLS** e **ROWS** sono mutuamente esclusive; se si vuole dividere una finestra in tre frame come mostrato nell'esempio qui sotto:



dovremo scrivere un codice HTML simile a questo che segue:

```
<FRAMESET ROWS="*, 2*">  
<FRAME NAME=F1 ...  
<FRAMESET COLS=20%,80% ....  
  <FRAME NAME=F2 ...  
  <FRAME NAME=F3 ...  
</FRAMESET>  
</FRAMESET>
```

Vedi anche il tag **<FRAME>** e le note generali sui frame.

## Tutti i tag

### Il tag **<H1>**

L'elemento HTML

**<H1>**

rappresenta un titolo e fa in modo che il testo venga visualizzato al livello 1 di intestazione. Questo è il livello più scuro e più grande tra i disponibili.

Il testo viene visualizzato con una interlinea doppia dopo il testo.

Il titolo **<H1>** equivale graficamente ad una formattazione con **<FONT SIZE=6><B>...</B></FONT><BR><BR>** ossia ad un carattere di dimensione 6, grassetto e con una riga vuota sotto. Nel caso si stampi il documento esso apparirà all'inizio della pagina.

Ecco un esempio di utilizzo del tag:

# Titolo di livello 1

e questo è il codice che lo genera:

```
<H1>Titolo di livello 1</H1>
```

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<H2>**, **<H3>**, **<H4>**, **<H5>**, **<H6>**, e le note generali sulle intestazioni.

## Tutti i tag

### Il tag **<H2>**

L'elemento HTML

**<H2>**

rappresenta un titolo e fa in modo che il testo venga visualizzato al livello 2 di intestazione. Questo è un livello piuttosto grande e scuro, ma più piccolo del corrispondente al livello 1.

Il testo viene visualizzato allineato a sinistra, senza indentazione, con una interlinea doppia dopo il testo.

Il titolo **<H2>** equivale graficamente ad una formattazione con **<FONT SIZE=5><B>...</B></FONT><BR><BR>** ossia ad un carattere di dimensione 5, grassetto e con una riga vuota sotto. Ecco un esempio di utilizzo del tag:

## Titolo di livello 2

e questo è il codice che lo genera:

```
<H2>Titolo di livello 2</H2>
```

### Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<H1>**, **<H3>**, **<H4>**, **<H5>**, **<H6>**, e le note generali sulle intestazioni.

## Tutti i tag

### Il tag **<H3>**

L'elemento HTML

#### **<H3>**

rappresenta un titolo e fa in modo che il testo venga visualizzato al livello 3 di intestazione. Il testo viene visualizzato in un carattere più piccolo del carattere di livello 2 e potrebbe, a seconda del browser, essere indentato o rappresentato in corsivo, con una interlinea doppia dopo il testo.

Ecco un esempio di utilizzo del tag:

### **Titolo di livello 3**

e questo è il codice che lo genera:

```
<H3>Titolo di livello 3</H3>
```

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<H1>**, **<H2>**, **<H4>**, **<H5>**, **<H6>**, e le note generali sulle intestazioni.



## Tutti i tag

### Il tag **<H4>**

L'elemento HTML

**<H4>**

rappresenta un titolo e fa in modo che il testo venga visualizzato al livello 4 di intestazione. Il testo può venir visualizzato in un carattere più piccolo del carattere di livello 3, possibilmente nel tipo di carattere predefinito e senza indentazione. Il titolo avrà una interlinea e mezza dopo il testo.

Ecco un esempio di utilizzo del tag:

#### **Titolo di livello 4**

e questo è il codice che lo genera:

```
<H4>Titolo di livello 4</H4>
```

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<H1>**, **<H2>**, **<H3>**, **<H5>**, **<H6>**, e le note generali sulle intestazioni.

## Tutti i tag

### Il tag **<H5>**

L'elemento HTML

**<H5>**

rappresenta un titolo e fa in modo che il testo venga visualizzato al livello 5 di intestazione. Il testo viene visualizzato in un carattere più piccolo del carattere di livello 4 e normalmente anche più piccolo del tipo di carattere predefinito per il testo generico, in grassetto e senza indentazione. Il titolo avrà una interlinea e mezza dopo il testo.

Ecco un esempio di utilizzo del tag:

**Titolo di livello 5**

e questo è il codice che lo genera:

```
<H5>Titolo di livello 5</H5>
```

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H6>**, e le note generali sulle intestazioni.

## Tutti i tag

### Il tag **<H6>**

L'elemento HTML

#### **<H6>**

rappresenta un titolo e fa in modo che il testo venga visualizzato al livello 6, che è il livello più basso delle intestazioni. Il testo viene visualizzato in un carattere più piccolo del carattere di livello 5 e generalmente anche più piccolo del tipo di carattere predefinito per il testo normale, in grassetto e senza indentazione. Il titolo avrà una interlinea e mezza dopo il testo.

Ecco un esempio di utilizzo del tag:

**Titolo di livello 6**

e questo è il codice che lo genera:

```
<H6>Titolo di livello 6</H6>
```

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera.

Vedi anche i tag **<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>**, e le note generali sulle intestazioni.

## Tutti i tag

### Il tag **<HEAD>**

Il tag contenitore

**<HEAD> ... </HEAD>**

è un elemento opzionale che serve a raccogliere un insieme di informazioni che non saranno visualizzate dal browser, ma che saranno utilizzate dal browser, e dai motori di ricerca per meglio visualizzare e gestire il documento.

Il testo racchiuso tra i due tag potrà contenere i tag **<TITLE>**, **<ISINDEX>**, **<BASE>**, **<META>**, **<LINK>**, **<SCRIPT>** e **<STYLE>**.

Il tag non ha attributi.

## Tutti i tag

### Il tag **<HR>**

Il tag

**<HR>**

è usato per visualizzare una riga orizzontale.

Il tag **<HR>** è utile per separare due parti di testo; un esempio base del suo utilizzo è il seguente:



e questo è semplicemente il codice che ne permette la visualizzazione:

```
<HR>
```

Il tag ha 5 attributi che controllano i vari aspetti della visualizzazione della riga.

#### 1. **ALIGN**

Questo attributo determina dove verrà allineata la riga, cioè a sinistra, al centro (impostazione predefinita) o a destra. I valori che può assumere sono **"left"**, **"center"** e **"right"**. Ovviamente l'effetto è visibile solo con righe che non prendano tutta l'intera larghezza della pagina. L'esempio qui sotto mostra i vari tipi di allineamento con una riga larga metà pagina.



Dopo la visualizzazione della riga, la visualizzazione del testo riprende dalla riga successiva.

#### 2. **NOSHADE**

La presenza di questo attributo, che non richiede valori, visualizzerà la riga senza effetti tridimensionali. Ecco qui sotto come verrà mostrata la riga, **senza** e **con** l'attributo impostato.



L'attributo è distinguibile solo con righe di spessore di almeno tre punti.

### 3. **SIZE**

Il valore di questo attributo stabilisce l'altezza o lo spessore della riga. Il valore predefinito è 2 e mostra una riga sottile.

La tabella sottostante mostra alcuni esempi di righe di diverse dimensioni:

SIZE	Esempio
Predefinito	<hr/>
1	<hr/>
2	<hr/>
8	<hr style="background-color: #808080; width: 100%;"/>
20	<hr style="background-color: #808080; width: 100%;"/>

### 4. **WIDTH**

L'attributo **WIDTH** specifica la lunghezza della riga in pixel o in percentuale dello spazio disponibile.

Il primo insieme di esempi mostra righe (allineate a sinistra) di 20, 40 ed 80 pixel rispettivamente:



mentre quest'altro esempio mostra righe, allineate al centro, di lunghezza rispettivamente il 25%, 50% e 75% della larghezza della pagina:



### 5. **COLOR**

Questo non è un attributo presente nello standard 3.2, ed è stato introdotto nello standard HTML 4. Come è facile aspettarsi, esso definisce il colore della linea e si utilizza allo stesso modo degli attributi **COLOR** degli altri tag.

L'esempio che segue mostra una riga rossa, se il vostro browser supporta l'attributo **COLOR**:

e questo è semplicemente il codice che ne permette la visualizzazione:

```
<HR COLOR="red">
```

---

## Browser

Tutti i browser visualizzano questo tag allo stesso modo, se nella forma base. Le differenze significative riguardano i seguenti aspetti:

- ? **Posizione di partenza:** Mosaic, Internet Explorer, Netscape Navigator e Ariadna iniziano la riga rispettando l'indentazione del paragrafo in cui sono inserite, Opera inizia la riga in una posizione intermedia tra il margine della pagina e l'indentazione del paragrafo.
- ? **Spazio verticale:** Mosaic, Netscape Navigator, Opera e Ariadna lasciano una riga intera tra due linee orizzontali successive. Internet Explorer lascia di sopra uno spazio pari all'altezza del tipo di carattere, e di sotto uno spazio pari all'altezza di una riga.

Solo Internet Explorer riconosce l'attributo **COLOR**.

## Tutti i tag

### Il tag **<HTML>**

Il tag contenitore

**<HTML> ... </HTML>**

indica al browser che il documento è di tipo HTML e dovrebbe essere visualizzato in accordo con le specifiche HTML. Il browser può capire il tipo di documento anche dal tipo di comandi che arrivano dal server che ha inviato il file, così anche se il documento non comincia col tag **<HTML>** verrà comunque riconosciuto dal browser.

Il tag non ha attributi.



## Tutti i tag

### Il tag **<I>**

L'elemento contenitore

**<I> ... </I>**

dice al browser di visualizzare il testo incluso mediante un formato corsivo. Lo standard HTML definisce questo come uno stile fisico, a differenza del tag **<EM>** che è uno stile logico.

Ecco un esempio di utilizzo del tag:

*Questo testo è in corsivo*

e questo è il codice che lo genera:

```
<I>Questo testo è in corsivo</I>
```

### Browser

Tutti i browser visualizzano il testo usando un carattere corsivo.

Vedi anche il tag **<EM>**.

## Tutti i tag

### Il tag **<IFRAME>**

Il tag

#### **<IFRAME>**

crea un frame interno alla pagina contenente un'altro documento. L'inserimento di un frame all'interno di una sezione di testo è quasi come inserire un oggetto tramite il tag **<OBJECT>**; entrambi permettono di inserire un documento HTML all'interno di un altro, ed entrambi possono essere trattati come un blocco di testo, allo stesso modo di una immagine o di una tabella. Il contenuto da visualizzare sarà definito all'interno di un altro file, che sarà specificato dall'attributo **SRC** nella dichiarazione dell'iframe. Siccome non tutti i browser sono in grado di visualizzare gli iframe sarà bene prevedere una procedura che mostrare un testo od un messaggio per i visitatori che utilizzano questi browser.

Il tag **<IFRAME>** è un "finto" contenitore, ovvero aspetta il tag di chiusura **</IFRAME>**, ma il testo contenuto tra di essi verrà visualizzato SOLO agli utenti il cui browser non consente la visualizzazione degli iframe. Nel testo da visualizzare per i browser non compatibili è consigliabile inserire il link alla pagina che si intende inserire nell'iframe, in modo da rendere le informazioni disponibili a tutti.

Il tag prevede 6 attributi.

#### 1. **SRC**

Il valore di questo attributo è l'indirizzo della pagina da caricare all'interno dell'iframe. La visualizzazione della pagina è gestita dal codice HTML presente all'interno del documento che si va ad aprire, e nessun attributo legato a testo o immagini del documento chiamante può essere esportato alla pagina caricata nell'iframe. Se l'attributo viene omesso viene caricata una pagina vuota. Può successivamente venirci caricata una pagina se si clicca su un collegamento ipertestuale avente quel frame come target.

#### 2. **NAME**

Questo attributo definisce il nome del frame, e fornisce un meccanismo mediante il quale il browser è in grado di riconoscere l'oggetto in questione e così aprire una pagina

all'interno di esso piuttosto che altrove. Il nome non può iniziare col trattino in basso.

### 3. **MARGINWIDT** e **MARGINHEIGHT**

Il valore di questi due attributi definisce lo spazio minimo in pixel tra il bordo dell'iframe ed il testo contenuto nell'iframe stesso.

### 4. **FRAMEBORDER**

Questo attributo attiva o disattiva il bordo dell'iframe. I valori possibili sono **"0"** o **"1"**.

### 5. **SCROLLING**

Questo attributo può assumere i valori **"yes"**, **"no"** o **"auto"** e fa in modo che il testo contenuto all'interno dell'iframe possa scorrersi mediante le barre di scorrimento poste a sinistra del frame e in basso. Se è omesso si intende "auto" e le barre di scorrimento sono presenti e attive solo se il testo supera la dimensione del frame che lo ospita.

### 6. **WIDTH** e **HEIGHT**

Questi attributi regolano la dimensione dell'iframe, in pixel o in percentuale della finestra.

### 7. **ALIGN**

Questo attributo controlla l'allineamento dell'iframe all'interno del documento che lo ospita. I valori possibili sono **"left"** (predefinito), **"center"**, **"right"**, **"top"**, **"middle"** o **"bottom"**.

Ecco un esempio di utilizzo di un iframe:

```
...
testo
...
<IFRAME SRC="ipagina.html" WIDTH="400"
HEIGHT="300"SCROLLING="no" FRAMEBORDER="1" NAME=ipagina>
Il tuo browser non supporta gli iframe o è configurato in modo da non
poterli visualizzare.<BR>
Puoi trovare il documento presso la pagina <A
HREF="ipagina.html">ipagina.html</A>.
</IFRAME>
...
```

## Browser

Pur facendo parte delle specifiche del W3C per l'HTML 4.01 e l'XHTML 1.0, attualmente il tag **<IFRAME>** è supportato solo da Internet Explorer 5 (e successivi) e da Amaya.

Vedi anche i tag **<FRAME>**, **<FRAMESET>** e le note generali sui frame.

## Tutti i tag

### Il tag **<IMG>**

L'elemento HTML

#### **<IMG>**

è usato per includere una immagine nella pagina visualizzata. Tutti i browser gestiscono le immagini usando gli standard conosciuti, come GIF (CompuServe Graphics Interchange), JPEG (Joint Photograoh Expert Group) e XBM (X-Windows Bit Map).

È importante capire che una immagine in linea fa parte della riga corrente del testo in cui è inserita. Per molte applicazioni è utile includere il tag **<IMG>** all'interno del tag **<P> ... </P>** in modo che essa si comporti come un blocco singolo e separato dal resto del testo.

Il tag **<IMG>**, che non ha il corrispondente tag di chiusura, ha un certo numero di attributi:

#### 1. **SRC**

Questo attributo è essenziale e indica l'indirizzo dell'immagine.

Per esempio l'immagine  è inclusa mediante il tag HTML:

```
<IMG SRC="busta.jpg" ALT="bustina">
```

#### 2. **ALIGN**

Il valore di questo attributo controlla il posizionamento dell'immagine relativamente al testo. Ci sono almeno 5 valori standard definiti nell'HTML 3.2, oltre che molti altri valori non standard:

##### ? **TOP**

La sommità dell'immagine è allineata con la sommità della riga di testo, che può anche essere la sommità di un'altra immagine. Notiamo che ciò non corrisponde necessariamente con la sommità del testo, in quanto una immagine vicina può essere più alta della riga di testo. Se l'immagine è più alta della riga di testo (come in questo caso), le successive righe di testo appariranno sotto l'immagine, e il testo non verrà visualizzato attorno all'immagine.

Ecco un esempio di una  immagine presentata con l'attributo **ALIGN**  impostato al valore **TOP**.

La sommità dell'immagine è allineata con la sommità del testo.



Quest'altro esempio presenta 2 immagini sulla stessa riga. La bandiera ha allineamento in basso (bottom), ma ha una altezza più grande della riga; successivamente la busta si allinea con la sommità dell'oggetto più alto della riga, ossia con la bandiera.

? **MIDDLE**

Il centro dell'immagine è allineato con la base della riga di testo.

Esempio   dell'allineamento di tipo **middle**.

Molti browser interpretano il valore non standard **CENTER** dandogli lo stesso comportamento di **MIDDLE**.

? **BOTTOM**

Questo è l'allineamento predefinito e può anche essere omesso.



Ecco un esempio di visualizzazione dell'attributo. Notiamo che la base dell'immagine corrisponde con la base del testo, e **non** con la base delle lettere il cui piede scende sotto la riga (lettere p, q, y ..).

? **LEFT**

Questo allineamento posiziona l'immagine sul margine sinistro della pagina, con il testo visualizzato sulla destra dell'immagine.



Ecco qui di fianco un esempio. Questo è il modo in cui HTML permette al testo di venir visualizzato attorno alle immagini e l'uso dell'allineamento a sinistra (o destra) è raccomandato per la maggior parte di opere illustrate con piccole immagini mostrate in riga col testo. Alcuni vecchi browser non consentono l'allineamento a sinistra o a destra. Normalmente viene disposto un piccolo spazio

tra l'immagine e il testo circostante, tale spazio può essere gestito mediante l'attributo **HSPACE** e **VSPACE**.

### ? **RIGHT**

Questo è simile all'allineamento a sinistra.



Ripetiamo anche l'esempio precedente. Notiamo che l'effetto stracciato dell'allineamento di destra del testo rende l'uso di piccole immagini allineate a destra meno efficace rispetto all'allineamento a sinistra.

Ecco un esempio con il testo giustificato. Questa è la visualizzazione più simile a quella che si ottiene facendo fluire un testo attorno ad una immagine, e l'uso di immagini allineate a destra o sinistra è raccomandato per la maggior parte di lavori illustrati con piccole immagini in linea. Le immagini allineate a sinistra o a destra interrompono molto meno lo scorrimento del testo rispetto ad immagini presentate all'interno del testo stesso.



I valori **TEXTTOP**, **BASELINE**, **ABSMIDDLE** e **ABSBOTTOM**, che non fanno parte dello standard HTML 3.2, sono riconosciuti dalla maggior parte dei browser ma sono interpretati in maniera più libera.

Teoricamente dovrebbe essere:

- ? **TEXTTOP** allinea la sommità dell'immagine con il punto più alto della riga di testo in cui è inserita.
- ? **BASELINE** allinea il fondo dell'immagine con la base della riga di testo, esclusi i caratteri il cui piede scende sotto la riga.
- ? **ABSMIDDLE** allinea la sommità dell'immagine con il punto più alto della riga di testo in cui è inserita.
- ? **ABSBOTTOM** allinea il fondo dell'immagine al punto più basso della riga.

Gli effetti dei vari allineamenti dipendono naturalmente dalle dimensioni relative di immagini e testo.

### 3. **BORDER**

Se una immagine è utilizzata per un collegamento ipertestuale, ad esempio è inclusa nei tag **<A> ... </A>**, viene visualizzata con un bordo azzurro, che può essere gestito mediante l'attributo **BORDER**.

Il valore assunto dall'attributo è lo spessore in pixel dalla cornice, che può essere un intero maggiore o uguale a zero. In caso di zero il bordo non sarà presente.



Ecco un esempio di una immagine con bordo pari a 5 inserita in un collegamento ipertestuale.

E questo è il codice che la genera:

```
<A HREF=#bordo><img src=busta.jpg border=5 align=left></a>
```

È utile impostare il bordo a zero in caso di immagini con lo sfondo trasparente, con le quali la presenza di un bordo azzurro sarebbe decisamente antiestetico.



L'effetto dell'attributo **BORDER** su una immagine non usata come link dipende dai browser; nella maggior parte dei casi viene messa attorno alla immagine una cornice nera il cui spessore varia da browser a browser.

#### 4. **WIDTH** e **HEIGHT**

Questi attributi possono essere utilizzati per allargare o comprimere una immagine ad una particolare dimensione in pixel o in percentuale della dimensione della finestra. Lo standard HTML semplicemente indica che questi attributi sono usati per indicare lo spazio da riservare all'immagine, l'espansione della immagine a riempire lo spazio disponibile è implicito. Questo è il motivo per il quale, in caso di immagini non trovate sul disco, viene visualizzata un'area vuota ridimensionata ai valori impostati (nell'esempio qui accanto sono 80x35).



Con la maggior parte dei browser, se è specificato un solo attributo allora l'altro si adatta automaticamente in proporzione.

In questo esempio la busta viene visualizzata con dimensioni di 120x40, contro le 28x21 originarie. Come si vede, l'immagine appare sgranata perché il browser non ha un programma di interpolazione e i singoli pixel vengono allargati a formare quadratini.



Nel caso le dimensioni sono specificate in termini di percentuale della finestra, ridimensionando la finestra le dimensioni dell'immagine si adattano al cambiamento. Ecco qui sotto una immagine la cui altezza è fissa (80 punti) e la larghezza è il 50% della pagina.





L'esempio mostra che è possibile modificare anche una sola dimensione, cambiando le proporzioni dell'immagine. Con immagini normali ciò non è realmente utile, ma con blocchetti di colore uniforme è possibile creare grafici ad istogrammi semplicemente allungando o accorciando una immaginina.

## 5. HSPACE e VSPACE

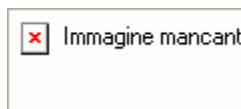
Questi attributi controllano lo spazio libero che viene impostato attorno all'immagine. I valori assunti indicano, in pixel, rispettivamente la larghezza e l'altezza di questo bordo.



L'esempio qui di lato mostra una immagine con HSPACE pari a 10 e VSPACE pari a 16.

## 6. ALT

L'attributo **ALT** è utilizzato per fornire un testo alternativo per i browser che non riescono a visualizzare l'immagine. Il testo inserito è utile anche per i motori di ricerca per indicizzare e archiviare la pagina. Il valore dell'attributo è semplicemente il testo che vogliamo venga visualizzato e può dare informazioni riguardo le dimensioni e la natura dell'immagine, oltre che un commento all'immagine stessa. Molti browser mostrano il testo in una casellina gialla quando passiamo il puntatore del mouse sopra l'immagine.



## 7. LOWSRC

Questo è un attributo abbastanza utile, che però non fa parte dello standard HTML 3.2. Lo scopo dell'attributo è di consentire ad una immagine a bassa risoluzione di venire visualizzata nel frattempo che viene caricata una immagine a risoluzione maggiore. Un esempio può essere una immagine in bianco e nero a bassa risoluzione, che pesa 35kB, mostrata nel frattempo che viene scaricata dal server l'immagine reale, del peso di 450kB. L'immagine a bassa risoluzione è derivata da quella a risoluzione maggiore tramite un qualunque programma di manipolazione di

immagini. Il risultato sarà una pagina che sembrerà caricata più velocemente.

Se il browser non riconosce l'attributo **LOWSRC** si può ottenere lo stesso effetto visivo utilizzando una GIF animata che non cicli, ma a differenza dell'attributo **LOWSRC** questa alternativa attende il caricamento dell'intera animazione prima di visualizzare l'immagine a bassa risoluzione.

## 8. USEMAP

È usato per associare l'immagine ad una lista di zone attive. Il valore di questo attributo è una url o un'ancora associato alla zona attiva definita dall'attributo **NAME** del tag **<MAP>**. È consigliato utilizzare il percorso completo per scrivere l'url alla zona attiva, sebbene ormai tutti i browser supportano la possibilità di indirizzo relativo.

## Immagini adiacenti

L'effetto di presentare due immagini adiacenti può variare da un browser all'altro; alcuni browser visualizzano le due immagini senza uno spazio tra di esse, altri invece mettono le immagini adiacenti solo se effettivamente non c'è spazio nel codice html tra i due tag **IMG**. L'esempio qui sotto mostra due immagini adiacenti, mentre quello sotto ancora mostra le stesse due immagini con tre spazi bianchi tra una immagine e l'altra.



## Browser

Le più grandi differenze riguardano l'allineamento quando vengono utilizzati valori non previsti nello standard HTML 3.2; le differenze di comportamento sembrano così complesse che il consiglio migliore sarebbe quello di evitare questi valori.

Opera ha alcuni problemi con le immagini allineate a sinistra, per le quali il bordo sinistro non viene allineato correttamente con la indentatura del testo.

La gestione dei bordi è piuttosto variabile; Ariadna e Internet Explorer producono bordi e colori dei link diversi a seconda che l'immagine sia inclusa o no in un tag **<A>**. Netscape Navigator cerca di ottenere lo stesso risultato, ma a volte sembra non disegnare tutto il bordo, Mosaic e vecchie versioni di Internet

Explorer visualizzano un bordo invisibile attorno all'immagine per immagini normali, un bordo colorato per immagini legate a collegamenti ipertestuali, di colore differente a seconda che il link sia visitato o attivo o no, mentre Mosaic visualizza un bordino attorno al bordo invisibile. Opera disegna una specie di bordo in 3D di un singolo pixel per i collegamenti ipertestuali.

Mosaic non riconosce gli attributi **HSPACE** e **VSPACE**. Ariadna non conserva le proporzioni originali se è specificata una sola delle due dimensioni **WIDTH** e **HEIGHT**. Mosaic non consente di indicare la dimensione in termini di percentuale della finestra, mentre alcune versioni di Internet Explorer sembrano avere uno strano bug per il quale il ridimensionamento funziona solo se l'immagine è inserita nelle prime 50 righe di testo.

Netscape Navigator, Internet Explorer e Opera separano due immagini adiacenti se c'è uno spazio tra i due tag **<IMG>**, mentre Mosaic e Ariadna visualizzano le immagini adiacenti e senza spazio intermedio.

Solo Netscape Navigator riconosce l'attributo **LOWSRC**.

Vedi anche i tag **<AREA>**, **<MAP>** e le note generali sulle mappe attive.

## Tutti i tag

### Il tag **<INPUT>**

L'elemento HTML

#### **<INPUT>**

è usato per ottenere ingressi da parte dell'utente in una gran varietà di modi all'interno di un blocco **<FORM>**. Il tipo particolare di input sarà indicato dall'attributo **TYPE**.

#### 1. Il bottone **RADIO**

Mediante questo tipo di ingresso il browser visualizza una serie di bottoni, dei quali solo uno può essere selezionato. È richiesto un singolo **<INPUT>** per ciascun bottone della serie, però tutti i bottoni avranno lo stesso valore per il campo **NAME** ed un valore diverso per l'attributo **VALUE**.

Ecco un esempio di utilizzo:

Selezionare un giorno della settimana:

- Lunedì
- Martedì
- Mercoledì
- Giovedì
- Venerdì

Invia

e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
Selezionare un giorno della settimana:<BR>
<INPUT TYPE="radio" NAME="giorni" VALUE="1" CHECKED>
Lunedì<BR>
<INPUT TYPE="radio" NAME="giorni" VALUE="2">Martedì<BR>
<INPUT TYPE="radio" NAME="giorni" VALUE="3"> Mercoledì<BR>
<INPUT TYPE="radio" NAME="giorni" VALUE="4"> Giovedì<BR>
<INPUT TYPE="radio" NAME="giorni" VALUE="5"> Venerdì<BR>
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

L'attributo **CHECKED** è usato per indicare un eventuale valore di partenza preselezionato.

## 2. Il bottone **CHECKBOX**

Questo bottone somiglia al **RADIO**, ma a ciascun bottone visualizzato è associato un nome diverso e, come visibile nell'esempio, può essere selezionato ogni elemento. I valori indicati dall'attributo **VALUE** sono trasmessi solo quando il bottone è selezionato.

Ecco un esempio di utilizzo:

Selezionare alcuni giorni della settimana:

- Lunedì
- Martedì
- Mercoledì
- Giovedì
- Venerdì



e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
Selezionare alcuni giorni della settimana:<BR>
<INPUT TYPE="checkbox" NAME="lunedì" VALUE="1" CHECKED>
Lunedì<BR>
<INPUT TYPE="checkbox" NAME="martedì" VALUE="1">Martedì<BR>
<INPUT TYPE="checkbox" NAME="mercoledì" VALUE="1" CHECKED>
Mercoledì<BR>
<INPUT TYPE="checkbox" NAME="giovedì" VALUE="1"> Giovedì<BR>
<INPUT TYPE="checkbox" NAME="venerdì" VALUE="1"> Venerdì<BR>
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

L'attributo **CHECKED** è usato allo stesso modo che per i bottoni **RADIO**, ma può essere applicato a ciascun bottone indifferentemente (anche più di uno contemporaneamente). Di fatto i singoli bottoni sono indipendenti l'uno dall'altro.

## 3. La casellina **TEXT**

Questo elemento fornisce una casella nella quale l'utente digita un certo testo che verrà trasmesso al server. Un eventuale testo iniziale può essere visualizzato mediante l'attributo **VALUE**.

Ecco un esempio di utilizzo:  
 Scrivere nella casella il giorno della settimana:

e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
Scrivere nella casella il giorno della settimana:
<INPUT TYPE="text" NAME="giorno" VALUE="Lunedì">
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

L'attributo **SIZE** può essere usato per indicare la dimensione della casella, e **MAXLENGTH** può essere utilizzato per specificare il numero massimo di caratteri che l'utente può inserire. I due valori non devono necessariamente essere gli stessi, e il comportamento del browser, se l'utente inserisce più caratteri di quanto specificati, è determinato dal browser stesso.

#### 4. La casellina **PASSWORD**

Questo elemento è del tutto simile al TEXT, però il browser non mostra i caratteri digitati, ma una serie di asterischi.

Ecco un esempio di utilizzo:

Inserisci una informazione segreta:  
 e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
Inserisci una informazione segreta:
<INPUT TYPE="password" NAME="pwd">
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

Naturalmente il testo inserito viene inviato "in chiaro" al programma che raccoglie le informazioni. Sebbene sia difficile, ma non impossibile, che qualcuno riesca a catturare le informazioni in transito è bene ricordare che molti server WWW tengono copia di richieste di dati in un file accessibile pubblicamente. Tutte le informazioni trasferite mediante il metodo GET saranno incluse in questo file di log, mentre quelle trasferite usando il POST non sono generalmente memorizzate.

## 5. Il pulsante **SUBMIT**

Questo elemento è stato già utilizzato in tutti gli esempi. Quando utilizzato, le informazioni associate ai tag **<INPUT>** compresi tra **<FORM>** e **</FORM>** vengono raccolte ed inviate al server.

Il valore assegnato all'attributo **VALUE** viene visualizzato come etichetta del pulsante, altrimenti in caso di assenza viene utilizzata l'etichetta predefinita **"Invia"** o **"Submit"**.

Ecco lo stesso esempio di prima con una etichetta diversa:

Inserisci una informazione segreta:

e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
Inserisci una informazione segreta:
<INPUT TYPE="password" NAME="pwd">
<INPUT TYPE="submit" VALUE="Spedisci">
</FORM>
```

Si possono avere tanti pulsanti **SUBMIT** quanti sono i tag **<FORM>** inseriti all'interno del testo.

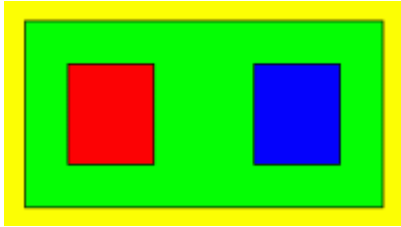
## 6. Il pulsante **RESET**

Questo pulsante è generalmente uguale a quello del **SUBMIT**, ma il suo effetto è quello di cancellare tutti i valori nei campi inseriti e di rimettere i valori iniziali.

## 7. Il metodo **IMAGE**

Questo tipo di ingresso ha un funzionamento simile al comportamento delle mappe immagine. La differenza principale è il fatto che le coordinate del punto in cui è posizionato il puntatore del mouse vengono trasferite al programma CGI associate ai campi di una variabile, come *name.x* e *name.y*, in cui *name* è la stringa associata all'attributo **VALUE**.

Ecco qui sotto un esempio di utilizzo, sullo schermo apparirà una immagine come questa, la cui area sarà tutta "sensibile" ed un eventuale click produrrà una coppia di coordinate che verranno trasmesse al file richiamato.



e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
<INPUT TYPE="image" NAME="coord" SRC="map2.gif">
</FORM>
```

Gli attributi **SRC** e **ALIGN** hanno lo stesso utilizzo che nel tag **<IMG>**.

Va notato che questo metodo implica l'invio al programma CGI dei dati raccolti nel FORM. Il click del mouse causa il trasferimento dei dati al programma e quindi non c'è necessità di inserire un pulsante di tipo **SUBMIT**.

Una immagine può essere utilizzata per inviare dati insieme con molti altri elementi, permettendo ad una mappa immagine di specificare molteplici aspetti dell'uso delle mappe prima di selezionare un punto della mappa. Questo non può essere realizzato con l'uso di **ISMAP** e **USEMAP**.

#### 8. Il metodo **FILE**

Questa è una recente aggiunta allo standard HTML. Come suggerito dal nome, esso permette ad un utente di selezionare un file sul proprio computer e di inviarlo al server. Generalmente viene visualizzato come una casella di testo.

Ecco un esempio di utilizzo:

Inserisci il file da trasferire:

Naturalmente gli attributi del tag **<FORM>** devono essere impostati correttamente affinché il trasferimento del file avvenga correttamente; in particolare:

- ? L'attributo **METHOD** deve essere specificato come **POST** ed il programma CGI deve essere opportunamente codificato.



? L'attributo **ENCTYPE** deve essere impostato a "multipart/form-data".

Nell'esempio qui sopra il programma semplicemente stampa il contenuto del file come se provenissero dallo standard input. Insieme al contenuto del file vengono stampate alcune informazioni come numeri di codice e una indicazione del contenuto Mime.

Questo metodo non va confuso con il RFC 959 del File Transfer Protocol. Se si utilizza questo metodo per consentire ai propri utenti di spedire file sul proprio server è opportuno considerare attentamente gli aspetti legati alla sicurezza e ai permessi di accesso.

## 9. Il campo **HIDDEN**

Ingressi di questo tipo non vengono mai visualizzati all'utente e possono essere utilizzati per trasmettere informazioni di stato o per capire quali pagine web sono state visitate.

Ecco un esempio di utilizzo:

Inserisci qui il messaggio:

e questo è il codice utilizzato:

```
<FORM ACTION="input2.html" METHOD="post">
<INPUT TYPE="hidden" NAME="cookie" value="12378XX57">
Inserisci qui il messaggio:
<INPUT TYPE="text" NAME="mesg">
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

Vedi anche i tag **<FORM>**, **<SELECT>**, **<TEXTAREA>** e le note generali sui CGI.

## Tutti i tag

### Il tag **<KBD>**

L'elemento contenitore

**<KBD> ... </KBD>**

è utilizzato per visualizzare campioni di testo come codici di programmi, testi inseriti da tastiera, etc.. Il significato del tag non è chiaro dallo standard HTML e il tag in effetti è usato raramente. La maggior parte dei browser visualizza il testo con un carattere di tipo monospazio, ma il layout di riga non viene rispettato; i caratteri speciali di html vengono rappresentati mediante i loro codici.

Ecco un esempio di testo:

```
1==1 1<=2 1>=0
```

e questo è il codice che abbiamo utilizzato:

```
<KBD>
1==1
1&lt;=2
1&gt;=0
</KBD>
```

Notiamo che non è la stessa cosa che usare un testo pre-formatato.

Il tag può essere usato anche con del testo in linea, eccone un esempio.

## Browser

Ariadna, Internet Explorer e Netscape Navigator visualizzano il testo con un carattere monospazio grassetto. Mosaic mostra il testo con un carattere proporzionale corsivo. Opera mostra il testo allo stesso modo del testo normale, probabilmente ignorando il tag.

Vedi anche i tag **<CITE>**, **<CODE>**, **<DFN>**, **<SAMP>**, **<VAR>** e le note generali sugli stili di testo.

**NOTA.** In alcuni browser questo tag ha lo stesso comportamento degli altri appena menzionati, e l'effetto è identico. Le differenze sono puramente storiche.

## Tutti i tag

### Il tag **<LI>**

Il tag

**<LI> ... </LI>**

è usato per mostrare un elemento di una lista ordinata o non. Il testo scritto dopo il tag sarà inteso come l'elemento della lista, finché non sarà incontrato un nuovo tag **<LI>** o la fine della lista oppure una nuova lista.

La presenza di un tag **<LI>** implica la fine di ogni precedente elemento di lista e quindi non c'è necessità del tag di chiusura **</LI>**, che quindi nella pratica è scarsamente utilizzato.

Il tag ha 2 attributi:

#### 1. **TYPE**

Questo attributo è utile solo con liste non ordinate, ossia costruite con **<UL>**, e permette di indicare l'elemento con un particolare simbolo grafico. I valori possibili sono **"disc"**, **"circle"** (predefinito) e **"square"**.

L'esempio che segue mostra l'utilizzo di questo attributo:

- ? disc
- o circle
- ▣ square
- ▣ default

e questo è il codice utilizzato:

```
<UL>
<LI TYPE=disc>disc
<LI TYPE=circle>circle
<LI TYPE=square>square
<LI>default
</UL>
```

#### 2. **VALUE**

Questo attributo è utile solo con liste ordinate, ossia costruite con il tag **<OL>**, e permette di impostare l'indicatore dell'elemento di lista ad un particolare valore; la numerazione continuerà poi dal nuovo valore. Eccone un esempio:

1. default
2. default
8. otto
9. default

e questo è il codice utilizzato:

```
<OL>  
<LI>default  
<LI>default  
<LI VALUE=8>otto  
<LI>default  
</OL>
```

Vedi anche i tag **<OL>**, **<UL>**, **<DL>**, e le note generali sulle liste.

## Tutti i tag

### Il tag **<LINK>**

L'elemento HTML

#### **<LINK>**

è un tag che offre numerose possibilità e definisce una relazione tra il documento ed un'altra risorsa. Diversamente dal tag **<A>**, il tag **<LINK>** non è un elemento contenitore e può comparire solo nell'intestazione del documento, tra i tag **<HEAD>** e **</HEAD>**.

Il tag **<LINK>** può essere utilizzato per:

- ? specifici menu di navigazione all'interno del documento
- ? controllare come deve essere stampata una serie di documenti HTML
- ? collegare risorse come script e fogli di stile
- ? pre-caricare la pagina successiva.

Lo standard HTML 4.0 definisce numerosi attributi per questo tag, alcuni dei quali sono effettivamente poco utilizzati.

#### 1. **HREF**

Il valore di questo attributo è l'indirizzo della risorsa collegata.

#### 2. **TYPE**

Questo attributo specifica il tipo di documento contenuto nella risorsa collegata; ciò permette di conoscere il tipo di dati attesi prima di caricare la pagina, evitando così il rischio di documenti non riconosciuti.

#### 3. **REL**

Questo attributo identifica il tipo di relazione tra il presente documento e la risorsa collegata. I tipi di relazione non sono ancora standardizzati nella versione dello standard HTML 4.01, sebbene sono in uso alcune convenzioni.

- ? **REL="top"** si riferisce alla cima di una gerarchia, come la prima pagina di una raccolta.
- ? **REL="contents"** indica una pagina contenente un sommario del documento.
- ? **REL="glossary"** si riferisce a un documento contenente il glossario dei termini utilizzati nel documento corrente.
- ? **REL="copyright"** contiene le informazioni riguardo al diritto d'autore sul documento.
- ? **REL="next"** e **REL="previous"** si riferiscono al successivo e precedente documento in una sequenza.

Sono utili per pre-caricare successive o precedenti pagine web.

- ? **REL="stylesheet"** si riferisce ad un foglio di stile esterno.
- ? **REL="start"** indica dove trovare la prima pagina di una raccolta di documenti.
- ? **REL="appendix"** definisce l'appendice per una raccolta di documenti.

#### 4. REV

Questo attributo è utilizzato per descrivere una relazione inversa tra il documento indicato e la pagina corrente. Può essere utilizzato per specificare l'autore del documento, il suo indirizzo email o un collegamento alla sua home page.

#### 5. HREFLANG

Questo attributo specifica dove trovare le versioni del documento nelle varie lingue indicate. Accetta come valori i nomi delle lingue nelle quali sono scritte le pagine e dovrebbe, di norma, essere utilizzato insieme con l'attributo **HREF**.

Ecco un esempio di utilizzo delle varie forme del tag **<LINK>**.

```
<HEAD>
<LINK REL="stylesheet" HREF="guide.css" TYPE="text/css">
<LINK REL="start" TITLE="La prima pagina del manuale"
  HREF="http://www.canoro.org/guide/manuale.html">
<LINK REL="alternate" TITLE="Il manuale in tedesco"
  HREFLANG="deutch" HREF="http://www.canoro.org/guide/deutch.html">
<LINK REL="next" HREF="listing.html">
<LINK REV="author" HREF="mailto:michefe@libero.it">
...
</HEAD>
...
```

Il tag **<LINK>** viene molto sfruttato con l'attributo **REL="stylesheet"** per inserire fogli di stile che permettono il controllo di tutti gli aspetti grafici della pagina e del sito. La versione elettronica di questa guida ed il sito **www.canoro.org** fanno abbondante uso dei fogli di stile e del tag **<LINK>**.

## Tutti i tag

### Il tag **<LISTING>**

Il tag

**<LISTING> ... </LISTING>**

è un tag obsoleto ed usato per presentare un testo con un tipo di carattere monospazio, come il Courier. Con lo standard HTML 3.2 è stato sostituito dal tag **<PRE>**.

Ecco un esempio del suo funzionamento:

Il tag **<LISTING>** è un tag obsoleto ed usato per presentare un testo con un tipo di carattere monospazio, come il Courier. Con lo standard HTML 3.2 è stato sostituito dal tag **<PRE>**.

Il testo contenuto nel tag **<LISTING>** può essere formattato, e non va a capo a meno che non si metta un **<BR>** o una interruzione manuale di riga (come il **<PRE>**).

## Browser

Ariadna ignora completamente il tag. Internet Explorer rende il testo con un carattere monospazio generalmente più piccolo di quello usato per il **<PRE>** e i caratteri speciali non sono interpretati. Mosaic si comporta come Internet Explorer ma il tipo di carattere è più grande. Netscape Navigator è simile a Internet Explorer ma riconosce i caratteri speciali, mentre non riconosce eventuale codice HTML all'interno del tag. Opera utilizza il tag allo stesso modo del tag **<PRE>**.

Vedi anche i tag **<ADDRESS>**, **<BLOCKQUOTE>**, **<PLAINTEXT>**, **<PRE>**, **<XMP>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<MAP>**

Il tag

**<MAP>**

è usato insieme all'attributo **USEMAP** del tag **<IMG>** e al tag **<AREA>**.

Questa combinazione è nota come tecnica della *mappa lato client*, che implementa una mappa in cui tutte le informazioni legate a zone attive sono incluse nel codice html in modo che il browser può trasmettere le coordinate al server.

La sintassi base è composta di tre parti:

1. L'attributo **USEMAP** del tag **<IMG>**, che è usato per associare l'immagine ad una lista di zone attive.

Il valore di questo attributo è una url o un'ancora associato alla zona attiva definita dall'attributo **NAME** del tag **<MAP>**.

È consigliato utilizzare il percorso completo per scrivere l'url alla zona attiva, sebbene ormai tutti i browser supportano la possibilità di indirizzo relativo.

2. Il tag **<MAP> ... </MAP>**

Questo elemento contenitore è usato per definire la lista delle zone attive; ha un singolo attributo **NAME** che serve ad indicare in modo univoco la mappa all'interno del codice html.

3. Il tag **<AREA>**

Questo tag è usato per specificare ciascuna singola zona attiva dell'immagine.

Un contenitore **<MAP> ... </MAP>** può contenere tanti tag **<AREA>** quanti sono necessari. L'effetto di includere qualcos'altro tra **<MAP>** e **</MAP>** non è completamente definito e dipende da browser e browser.

Vedi anche i tag **<AREA>**, **<IMG>** e le note generali sulle mappe attive.



## Tutti i tag

### Il tag **<MARQUEE>**

L'elemento contenitore

**<MARQUEE> ... </MARQUEE>**

è usato per visualizzare un testo che scorre orizzontalmente all'interno di una casella.

Non fa parte dello standard HTML 3.2 e originariamente era supportato solo da Internet Explorer. Tuttora alcuni browser non ne consentono la visualizzazione.

Nel realizzare una pagina web è bene non utilizzare troppi elementi animati, per evitare un effetto fastidioso e che possa distrarre l'utente dal reale contenuto del sito.

Un modo di utilizzo potrebbe essere il seguente:

```
<CENTER>
<MARQUEE WIDTH=75%><FONT SIZE=5 COLOR=90FF90>Ciao
mondo</FONT></MARQUEE>
</CENTER>
```

Alcuni browser non riconoscono altri tag inseriti tra **<MARQUEE>** e **</MARQUEE>**, sicché è impossibile animare testi formattati in modo diverso o immagini. La maggior parte dei browser ha superato questa difficoltà.

Il tag **<MARQUEE>** ha una decina di attributi:

#### 1. **ALIGN**

Questo attributo può assumere i valori **top**, **middle** e **bottom**, e controlla il posizionamento della scritta relativamente all'elemento di testo in cui è inserita, allo stesso modo dell'attributo **ALIGN** del tag **<IMG>**.

#### 2. **BEHAVIOR**

Questo tag controlla il comportamento dello scorrimento. Può assumere tre possibili valori e il testo comincia a scorrere non appena la pagina è caricata completamente, non quando l'utente visualizza il "marquee" sullo schermo.

VALORE	Effetto
Scroll	Il testo scorre da destra a sinistra e riappare quando è completamente scomparso alla fine della casella.

	Questo è il comportamento predefinito.
Slide	È simile al comportamento di scroll, ma quando il testo raggiunge la fine della casella, scompare e ricomincia da capo. Se la visualizzazione non è ciclica il testo rimane posizionato all'estremità della casella.
Alternate	Il testo rimbalza alternativamente tra i bordi della casella, senza uscirne fuori.

Internet Explorer non prende il valore **infinite** per l'attributo **LOOP** nel caso del comportamento di tipo **SLIDE**.

### 3. **LOOP**

Questo tag controlla il numero di iterazioni che compie il testo. Il valore predefinito è **"infinite"**, e i valori possibili sono un numero intero positivo da 1 fino a 2147483647.

### 4. **BGCOLOR**

Questo attributo controlla il colore di fondo della casella in cui scorre il testo. Nell'esempio superiore lo sfondo della casella era impostato a nero, mentre il fondo della cella di tabella rimaneva al colore originale.

### 5. **DIRECTION**

Questo attributo controlla la direzione dello scorrimento del testo. I valori permessi sono **"left"** e **"right"**, specificando la direzione della prima apparizione del testo. La scritta parte dal bordo opposto rispetto a quello indicato. Il valore di questo attributo non ha alcun effetto sul comportamento del testo se l'attributo **BEHAVIOR** è impostato su **alternate**.

### 6. **HEIGHT** e **WIDTH**

Controllano rispettivamente l'altezza e la lunghezza della casella in cui scorre il testo. Il comportamento è identico agli stessi attributi del tag **<IMG>**.

### 7. **HSPACE** e **VSPACE**

Controllano lo spazio orizzontale e verticale attorno alla casella di scorrimento. Il comportamento è identico agli stessi attributi del tag **<IMG>**.

### 8. **SCROLLAMOUNT**

Questo attributo controlla lo spostamento in pixel tra due successive visualizzazioni, dando l'impressione dell'animazione.

#### **9. SCROLLDELAY**

Questo attributo controlla il tempo intercorso tra una visualizzazione e la successiva dello scorrimento, dando l'impressione dell'animazione.

## Tutti i tag

### Il tag **<MENU>**

Il tag

**<MENU>**

è usato raramente e produce una lista non ordinata, con lo stesso effetto del tag **<UL>**. Il tag è usato per realizzare una lista di un'unica colonna.

Eccone un esempio:

- ? Barolo
- ? Fragolino
- ? Passito
- ? Sangiovese
- ? Brunello
- ? Chianti

Generalmente produce lo stesso effetto del tag **<UL>** e a partire dallo standard HTML 3.2 può avere anche l'attributo **COMPACT**.

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera, mostrando una semplice lista indicizzata con pallini.

Vedi anche i tag **<DIR>**, **<UL>** e le note generali sulle liste.

## Tutti i tag

### Il tag **<META>**

L'HTML permette di specificare informazioni riguardo il documento piuttosto che sul contenuto del documento.

L'elemento HTML

#### **<META>**

può essere usato per identificare le proprietà del documento ed assegnare valori a queste proprietà.

Ciascun elemento **<META>** specifica una coppia proprietà/valore.

L'attributo **NAME** identifica la proprietà e **CONTENT** ne specifica il valore.

È possibile specificare le proprietà di un documento anche tramite il tag **<LINK>**, collegando il file con le informazioni, oppure tramite l'attributo **PROFILE** del tag **<HEAD>**.

Il tag **<META>** accetta 4 attributi.

#### 1. **NAME**

Questo attributo identifica il nome della proprietà. Non esiste una normativa che regola l'insieme delle proprietà consentite, sebbene le proprietà più utilizzate sono più o meno standard.

#### 2. **CONTENT**

Questo attributo specifica il valore della proprietà. Anche per questi valori non esiste una normativa definita.

#### 3. **SCHEME**

Questo attributo specifica uno schema che permette al browser di interpretare correttamente il valore delle proprietà indicate dall'attributo **CONTENT**.

#### 4. **HTTP-EQUIV**

Questo attributo, utilizzato in luogo di **NAME** permette di costruire informazioni per le intestazioni dei messaggi nei protocolli HTTP. I server HTTP possono utilizzare il nome della proprietà specificata dall'attributo **HTTP-EQUIV** per creare una intestazione che rispetti lo standard RFC822 dei messaggi HTTP.

#### 5. **LANG**

Questo attributo specifica la lingua del documento e può essere utilizzato dal browser e dai motori di ricerca.

Alcuni browser supportano l'uso del tag **<META>** per ricaricare la pagina corrente dopo un determinato numero di secondi con la possibilità di sostituirla con una diversa.

```
<META HTTP-EQUIV="refresh"
CONTENT="3,http://www.canoro.org/index.html">
```

Il valore della proprietà è un numero che specifica i secondi di tempo da aspettare prima di ricaricare seguito dall'indirizzo della pagina da caricare.

Un utilizzo importante del tag **<META>** è quello di specificare parole chiavi in modo da aiutare i motori di ricerca a migliorare la qualità delle ricerche. Quando si indicano informazioni sul linguaggio del documento, i motori di ricerca possono filtrare in base all'attributo **LANG** e mostrare i risultati in base alle preferenze di lingua dell'utente.

```
<META NAME="keywords" LANG="it" CONTENT="vacanze, Grecia,
sole">
<META NAME="keywords" LANG="en" CONTENT="holiday, Greece,
sunshine">
<META NAME="keywords" LANG="fr" CONTENT="vacances, Grèce,
soleil">
```

L'efficienza dei motori di ricerca può essere ulteriormente migliorata utilizzando il tag **<LINK>** per indicare eventuali traduzioni del documento in altre lingue, oppure collegamenti a versioni di altro tipo come PDF oppure, se il documento fa parte di un insieme, un collegamento al punto di inizio o all'indice dell'insieme.

Il tag **<META>** può essere usato per specificare anche informazioni sul tipo di documento e sui documenti utilizzati per stabilirne il profilo, come la linguaggio predefinito di scripting, il linguaggio usato nei fogli di stile e il set di caratteri.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=ISO-8859-5">
```

Infine l'attributo **SCHEME** permette all'autore di suggerire al browser l'interpretazione corretta delle informazioni riportate dai meta-tag. Ad esempio il valore "Date-Month-Year" dell'attributo **SCHEME** potrebbe eliminare l'ambiguità che si presenta di fronte

ad una data nella forma "05-09-06", che può essere confusa con 5 Settembre 2006, con 6 Settembre 2005, con 9 Maggio 2006, etc..

Ecco un esempio di utilizzo del tag **<META>** con le proprietà più comuni.

```
<HEAD PROFILE="http://www.canoro.org/profili/guida">  
<META NAME="author" CONTENT="Michele Ferrara">  
<META NAME="copyright" CONTENT="© 1998 LaCasalinga">  
<META NAME="keywords" CONTENT="guida, HTML, siti web, motori di  
ricerca, tag">  
<META NAME="date" CONTENT="2006-04-18T08:49:37+00:00">  
</HEAD>
```

## Tutti i tag

### Il tag **<NOFRAMES>**

L'elemento

#### **<NOFRAMES>**

specifica un testo che deve venire visualizzato solo dai browser che non supportano i frame o che sono configurati in modo da non visualizzarli. Ad esempio il tag **<NOFRAMES>** può essere usato per spiegare cosa dovrebbe essere visualizzato nei documenti contenuti dai frame, in modo da fornire il maggior numero possibile di informazioni ad un eventuale lettore.

Il tag non ha attributi e può contenere un testo che può anche essere formattato.

Ecco un possibile esempio:

```
<FRAMESET ROWS="*, 2*">
  <FRAME NAME=F1 ...
  <FRAMESET COLS=20%,80% ....
    <FRAME NAME=F2 ...
    <FRAME NAME=F3 ...
  </FRAMESET>
  <NOFRAMES><P>Ecco la <A HREF="main-
noframes.html">pagina</A> che verrebbe visualizzata se il tuo browser
supportasse i frame.</NOFRAMES>
</FRAMESET>
```

Vedi anche i tag **<FRAME>**, **<FRAMESET>** e le note generali sui frame.



## Tutti i tag

### Il tag **<OL>**

L'elemento contenitore

**<OL> ... </OL>**

è usato per indicare una lista numerata. In una lista ordinata ciascun elemento è visualizzato preceduto da un numero di sequenza successivo a quello del precedente elemento.

Ecco un esempio di lista annidata:

1. Livello 1, primo elemento
2. Livello 1, secondo elemento
3. Livello 1, terzo elemento
  1. Livello 2, primo elemento
  2. Livello 2, secondo elemento
    1. Livello 3, primo elemento
    2. Livello 3, secondo elemento
  3. Livello 2, terzo elemento
    1. Livello 3, seconda lista, primo elemento
    2. Livello 3, seconda lista, secondo elemento
      1. Livello 4, primo elemento
4. Livello 2, quarto elemento
4. Livello 1, quarto elemento

Il tag **<OL>** ha tre attributi.

#### 1. **TYPE**

I valori di questo attributo controllano lo stile della numerazione e possono essere **a**, **i**, e **1**, indicando una numerazione mediante rispettivamente lettere, numeri romani o numeri decimali. I valori **A** e **I** visualizzeranno indici maiuscoli. Il valore predefinito è **1**.

#### 2. **START**

Questo attributo stabilisce l'indice di partenza della numerazione; dovrebbe essere sempre un decimale intero, indipendentemente dal tipo di numerazione utilizzato.

Ecco un esempio di lista che inizia dal valore 7:

- g. Primo
- h. Secondo
- i. Terzo

scritta mediante questo codice:

```
<OL TYPE=a START=7>
<LI>Primo
<LI>Secondo
```

```
<LI>Terzo
</OL>
```

### 3. COMPACT

Questo attributo poco conosciuto permette di visualizzare una lista in una maniera più compatta. Lo standard HTML 3.2 non fornisce altre indicazioni.

## Browser

Mosaic non riconosce gli attributi **TYPE** e **START** ed implementa una indentatura minore rispetto agli altri browser.

Una differenza interessante tra vecchie versioni di Netscape Navigator e Internet Explorer è la possibilità, col primo, di indentare gli elementi della lista di un valore tale che comincino tutti sotto la stessa verticale, anche con indici di lunghezza diversa.

- o Internet Explorer
  - 9. Lazio
  - 10. Campania
- o Netscape Navigator
  - 9. Lazio
  - 10. Campania

Non c'è dubbio che la visualizzazione offerta da Netscape sia più attraente.

L'inserimento da parte di Internet Explorer di spazi extra tra gli indici e il testo è graficamente molto antiestetico. A partire da Internet Explorer 4.0 il problema è stato risolto e la visualizzazione è simile allo stile adottato da Netscape Navigator.

Vedi anche i tag **<LI>**, **<UL>**, **<DL>** e le note generali sulle liste.

## Tutti i tag

### Il tag **<P>**

L'elemento HTML

**<P>**

indica l'inizio di un blocco di paragrafo. Questo significa che la riga corrente è terminata e il testo prosegue sulla riga successiva. Se il tag viene messo all'interno di un blocco di paragrafo si forza l'inizio di un nuovo paragrafo. Lo stesso effetto si ottiene con il tag di chiusura **</P>**.

Esempio:

C'è un tag **<P>** immediatamente dopo la parola **qui**

nel testo. Non c'è spazio attorno al tag, ma viene semplicemente inserito come:

```
... dopo la parola qui<P>nel testo...
```

L'effetto di una sequenza di tag **<P>** può variare da browser a browser. Sulla maggior parte dei browser l'effetto di una sequenza di tag **<P>** è lo stesso di un singolo tag **<P>**.

Il tag **<P>** può avere l'attributo **ALIGN** che riconosce i valori **left** (predefinito), **center**, **right** e **justify**. I paragrafi seguenti illustrano l'effetto di questi allineamenti; nel provare alcuni esempi col proprio pc può essere utile ridimensionare la finestra di browser per vedere meglio l'effetto dell'allineamento dei paragrafi.

Questo è un paragrafo allineato a **sinistra** mediante **<P ALIGN=left>** e risulta simile ad un qualunque altro paragrafo in cui non sia impostato alcun allineamento. L'effetto ottenuto è che il

bordo destro del testo appare disallineato, mentre il bordo sinistro è incolonnato.

Questo paragrafo invece è allineato al **centro**. Entrambi i bordi sono disallineati, della stessa quantità. L'effetto è simile al tag **<CENTER>**, con la differenza dell'interlinea lasciata prima e dopo il testo da parte del tag **<P>**. Inoltre **<CENTER> ... </CENTER>** può essere applicato ad una sequenza di paragrafi, mentre l'attributo **ALIGN** si applica solo al tag **<P>** in cui è inserito.

Questo è un paragrafo allineato a **destra**. Il bordo sinistro appare disallineato, mentre quello destro allineato correttamente al margine della pagina. Può essere utile se usato insieme ad una immagine allineata a destra, ma è comunque raramente usato.

Questo paragrafo, infine, è **giustificato**, ossia allineato sia al bordo sinistro che al bordo destro, mediante **<P ALIGN=justify>**. Questo è l'allineamento utilizzato in tutti i testi di questa guida di HTML e offre una visualizzazione più leggibile e gradevole. Visualizzando sullo schermo un esempio di questo tipo, nel ridimensionare la finestra di browser le parole scorrono da una riga all'altra, ma il bordo del testo resta sempre allineato col margine della finestra.

## Browser

Tutti i browser visualizzano il tag alla stessa maniera.

Vedi anche i tag **<BR>**, **<DIV>** e **<CENTER>**.

## Tutti i tag

### Il tag **<PARAM>**

Il tag

**<PARAM>**

è utilizzato solo all'interno di un contenitore **<APPLET>**. È utilizzato per passare un valore o un parametro ad una applet. Ha due attributi, **NAME** e **VALUE**.

Vedi anche il tag **<APPLET>**.

## Tutti i tag

### Il tag **<PLAINTEXT>**

Il tag

**<PLAINTEXT>**

è un elemento HTML obsoleto ed era usato per indicare la fine della parte HTML del documento. È stato rimpiazzato da **</HTML>**.

Lo standard HTML 3.2 suggerisce, probabilmente in maniera incorretta, che è equivalente al tag **<PRE>**.

### Browser

Internet Explorer e Mosaic lo interpretano come un tag contenitore che blocca l'interpretazione del codice HTML finché non si incontra un tag di chiusura **</PLAINTEXT>**. Ariadna, Opera e Netscape Navigator interpretano il tag come **</HTML>**, inteso come "*fine del documento*", e non riconoscono più alcun altro tag.

Vedi anche il tag **<ADDRESS>**, **<BLOCKQUOTE>**, **<LISTING>**, **<PRE>**, **<XMP>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<PRE>**

L'elemento contenitore

**<PRE> ... </PRE>**

è utilizzato per visualizzare testo pre-formatato. All'interno del tag tutti gli spazi e ritorni a capo sono visualizzati come tali, ma il codice HTML continua ad essere riconosciuto, quindi non è possibile copiare ed incollare brutalmente del testo tra i tag **<PRE>** e **</PRE>**, ma se si vuole inserire un simbolo di minore occorre scrivere `&lt;`. L'effetto di un tag **<FONT>** o di **<IMG>** non è completamente certo e definito, e spesso dipende dal browser utilizzato. Lo standard HTML 3.2 suggerisce che eventuali tag contenuti nel codice pre-formatato siano ignorati.

Il testo viene visualizzato con un carattere monospazio, con una interruzione di paragrafo prima e dopo.

Inoltre righe di testo molto lunghe non vengono spezzate quando termina la larghezza della pagina, sicché compare la barra di scorrimento orizzontale e la pagina può essere scorsa da un lato all'altro per visualizzare i caratteri che spariscono oltre il margine destro.

Sebbene lo standard HTML 3.2 fornisca l'attributo **WIDTH** per delimitare la zona preformattata, è bene non permettere l'utilizzo di righe troppo lunghe.

Ecco un semplice utilizzo del tag **<PRE>**:

Questo è un testo preformattato.

Ci sono quattro spazi all'inizio di questa riga.

Una riga corta.

Una riga vuota sopra.

.. e questo è il codice HTML:

```
<PRE>
Questo è un testo preformattato.
  Ci sono quattro spazi all'inizio di questa riga.
Una riga corta.

Una riga vuota sopra.
<PRE>
```

E adesso una riga lunga per mostrare il funzionamento dell'attributo **WIDTH** impostato al valore 40:

Questa è una riga molto lunga, con l'attributo **WIDTH** impostato ad

## Browser

Tutti i browser riconoscono il tag alla stessa maniera. Nessun browser testato sembra riconoscere l'attributo **WIDTH**. Tutti i browser permettono una formattazione del testo, riconoscendo i tag **<FONT>**, **<B>**, **<I>**, ... nonché l'uso del tag **<IMG>** per le immagini.

Vedi anche i tag **<ADDRESS>**, **<BLOCKQUOTE>**, **<LISTING>**, **<PLAINTEXT>**, **<XMP>** e le note generali sugli stili di testo.



## Tutti i tag

### Il tag `<SAMP>`

L'elemento contenitore

`<SAMP> ... </SAMP>`

è utilizzato per visualizzare porzioni di testo intese come output di programmi. Il significato non è completamente chiaro dallo standard HTML e il tag è usato raramente. La maggior parte dei browser visualizza il testo con un carattere monospazio così come è inserito ma il layout di riga non viene rispettato. I caratteri speciali di HTML vanno dichiarati mediante i loro codici.

Ecco un esempio di utilizzo:

```
1==1 1<=2 1>=0
```

e questo è il codice che lo genera:

```
<SAMP>
1==1
1<=2
1>=0
</SAMP>
```

Notiamo che non è la stessa visualizzazione del testo preformattato ottenuto con `<PRE>`.

Può essere usato anche all'interno di un paragrafo, come ad esempio: un esempio di testo.

## Browser

Ariadna, Netscape Navigator, Internet Explorer e Opera visualizzano il testo contenuto all'interno del tag `<SAMP>` con un carattere monospazio del tutto simile a quello del tag `<PRE>`; Mosaic si comporta allo stesso modo, ma utilizza un carattere di dimensioni maggiori.

Vedi anche i tag `<CITE>`, `<CODE>`, `<DFN>`, `<KBD>`, `<VAR>` e le note generali sugli stili di testo.

**NOTA.** In alcuni browser questo tag ha lo stesso comportamento degli altri appena menzionati, e l'effetto è identico. Le differenze sono puramente storiche.

## Tutti i tag

### Il tag **<SCRIPT>**

L'elemento contenitore

**<SCRIPT> ... </SCRIPT>**

permette di inserire un codice programmabile all'interno del documento. Questo elemento può comparire quante volte si vuole indifferentemente nell'intestazione o nel corpo del documento.

Il codice o lo script deve essere definito tra i tag **<SCRIPT>** e **</SCRIPT>** oppure in un file esterno. Se l'attributo **SRC** non è impostato il browser cerca i comandi di scripting nelle righe successive. Se l'attributo **SRC**, invece, contiene l'url di un file allora il contenuto tra i tag di apertura e chiusura viene ignorato.

Tutti gli script sono interpretati da un interprete di script che deve essere noto al browser ed abilitato.

Esistono numerosi linguaggi di scripting, dal JavaScript al VbScript, al Tcl, e la sintassi dello script dipende ovviamente dal linguaggio utilizzato.

Il tag **<SCRIPT>** prevede tre attributi, di cui uno "sempre" necessario.

#### 1. **TYPE**

Questo attributo dovrebbe essere sempre obbligatorio e specifica il tipo di linguaggio usato dallo script. Può essere omesso, ma in tal caso il browser usa l'interprete predefinito, che quasi sempre corrisponde all'interprete per JavaScript. In caso di mancato riconoscimento dello script il browser rileva l'errore, e visualizza la pagina senza eseguire i comandi di script. Il vecchio standard HTML prevedeva l'attributo **LANGUAGE** per questo scopo.

#### 2. **SRC**

Il valore di questo attributo indica il file esterno in cui è definito il codice da eseguire. Può essere un indirizzo locale o assoluto e risiedere anche su un server diverso.

#### 3. **DEFER**

Quando specificato, questo attributo booleano indica al browser che lo script non genera alcun contenuto nel documento (ad esempio con "document.write" in JavaScript) e il browser può visualizzare in maniera corretta il documento.

Nel caso il browser non preveda l'utilizzo di script è bene fare in modo che non visualizzi il contenuto del tag **<SCRIPT>**, inserendo il codice programmabile all'interno di un commento HTML.

```
<SCRIPT TYPE="text/javascript">
<!-- per nascondere il codice ai vecchi browser
function square(i)
{
  document.write("La chiamata ha passato il valore ", i, " alla
funzione.", "<BR>")
  return i * i
}
document.write("La funzione ha restituito ", square(5), ".")
// chiusura del commento per i vecchi browser -->
</SCRIPT>
...
```

Rimandiamo il lettore ad trattazione più approfondita sui linguaggi di scripting per pagine web.

## Tutti i tag

### I tag **<SELECT>** e **<OPTION>**

Gli elementi HTML

**<SELECT>** e **<OPTION>**

sono usati per selezionare una opzione in un menu a tendina o a finestra.

Selezionare un giorno della settimana:



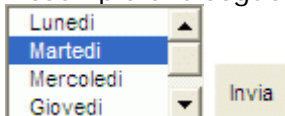
questo è il codice utilizzato per la casella qui sopra:

```
<FORM ACTION="input2.html" METHOD="post">
Selezionare un giorno della settimana:<BR>
<SELECT NAME="day">
  <OPTION>Lunedì
  <OPTION>Martedì
  <OPTION>Mercoledì
  <OPTION>Giovedì
  <OPTION>Venerdì
</SELECT>
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

Il tag **<SELECT>** ha due ulteriori attributi **SIZE** e **MULTIPLE** che controllano il numero di opzioni visualizzate e la possibilità di scelta multipla.

Il tag **<OPTION>** ha due attributi **SELECTED** e **VALUE** che permettono ad una opzione di apparire inizialmente come selezionata e per associare un valore che verrà associato alla particolare opzione al posto del nome dell'opzione (il testo descrittivo).

L'esempio che segue mostra tutti gli utilizzi descritti:



questo è il codice utilizzato nell'esempio:

```
<FORM ACTION="input2.html" METHOD="post">
<SELECT NAME="day" SIZE="4" MULTIPLE>
  <OPTION VALUE="1">Lunedì
  <OPTION VALUE="2" SELECTED>Martedì
  <OPTION VALUE="3">Mercoledì
  <OPTION VALUE="4">Giovedì
  <OPTION VALUE="5">Venerdì
</SELECT>
<INPUT TYPE="submit" VALUE=" Invia ">
</FORM>
```

Vedi anche i tag **<FORM>**, **<INPUT>** e **<TEXTAREA>**.

## Tutti i tag

### Il tag **<SMALL>**

L'elemento HTML

**<SMALL> ... </SMALL>**

dice al browser di visualizzare il testo mediante un tipo di carattere più piccolo.

Ecco un esempio di utilizzo del tag:

Questo testo è più piccolo

e questo è il codice che lo genera:

```
<SMALL>Questo testo è più piccolo</SMALL>
```

L'effetto si nota maggiormente se il testo piccolo è inserito nel testo "normale". Generalmente l'utilizzo di questo tag equivale ad impostare l'attributo **SIZE** del tag **<FONT>** al valore di **-1**.

## Browser

Tutti i browser riconoscono il tag allo stesso modo e visualizzano il testo più piccolo di una unità. Mosaic non riconosce il tag.

Vedi anche il tag **<BIG>**, **<FONT>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<SPACER>**

Il tag **<SPACER>** non appartiene allo standard HTML 3.2 ed introduce uno spazio del valore indicato.

Il tag ha due attributi:

#### 1. **TYPE**

Questo attributo permette di introdurre una spaziatura orizzontale o verticale. **horizontal** crea uno spazio della lunghezza indicata all'interno della riga corrente. Esempio: c'è uno spazio di 4 punti subito dopo questa parola (se il browser ne permette la visualizzazione). **vertical** crea uno spazio verticale (tra due righe) della dimensione indicata. Esempio: c'è uno spazio di 80 punti subito dopo questa

parola (se il browser ne permette la visualizzazione).

Notiamo che il tag **<SPACER>**, qualora venga riconosciuto, forza una nuova riga.

#### 2. **SIZE**

Specifica la dimensione dello spazio, in pixel.

## Browser

Solo Netscape Navigator riconosce questo tag.

## Tutti i tag

### Il tag **<SPAN>**

L'elemento contenitore

**<SPAN>** ... **</SPAN>**

in combinazione con gli attributi **ID** e **CLASS** offre un meccanismo per estendere gli attributi di allineamento del tag **<P>** ad un blocco di testo. Il tag viene generalmente utilizzato con i fogli di stile in modo da dare una struttura più accurata al testo.

Il tag **<SPAN>** ha 4 attributi:

#### 1. **ID**

Questo attributo assegna un nome all'elemento e tale nome deve essere unico all'interno del documento.

#### 2. **CLASS**

Questo attributo assegna il nome di una classe all'elemento. Al nome della classe può essere associato qualunque stringa alfanumerica. È possibile assegnare più di una classe, separando i nomi da spazi bianchi.

#### 3. **ALIGN**

Questo attributo permette di allineare orizzontalmente l'elemento. I valori possibili sono **"left"** (predefinito), **"center"**, **"right"** e **"justify"**.

#### 4. **STYLE**

Questo attributo è effettivamente il più utilizzato, e permette di definire uno stile grafico per il blocco di testo. All'attributo viene assegnato come valore una stringa, inclusa tra doppi apici, contenente le informazioni di stile per il determinato elemento. La stringa viene costruita sul modello dei fogli di stile CSS.

## Browser

Tutti i browser attuali visualizzano correttamente il tag.

Vedi anche i tag **<CENTER>**, **<DIV>** e **<P>**.



## Tutti i tag

### Il tag **<STRIKE>**

L'elemento HTML

**<STRIKE>** testo **</STRIKE>**

è utilizzato per visualizzare un testo barrato da una riga orizzontale.

Questo è un testo con lo stile ~~**<STRIKE>**~~.

Vedi anche le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<STRONG>**

L'elemento contenitore

**<STRONG> ... </STRONG>**

ordina al browser di mostrare in grassetto il testo contenuto tra i tag di apertura e chiusura. Lo standard HTML definisce **<STRONG>** uno stile logico distinguendolo dal tag **<B>** che è uno stile fisico.

Ecco un esempio di utilizzo del tag:  
Questo è uno stile **grassetto**.

e questo è il codice che lo genera:

```
Questo è uno stile <STRONG>grassetto</STRONG>
```

La differenza tra uno stile fisico e uno logico è che nel primo caso lo stile indica al browser di visualizzare il testo in un tipo di carattere in grassetto, mentre con uno stile logico al browser è detto di mostrare il testo con una enfasi maggiore, lasciando il browser libero di decidere come ciò verrà eseguito (possono essere usati colori diversi o sottolineatura), anche se ormai tutti i browser si stanno adattando a mostrare in grassetto un testo indicato come **STRONG**.

## Browser

Tutti i browser visualizzano questo tag alla stessa maniera, rendendo il testo più scuro e pesante.

Vedi anche il tag **<B>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<STYLE>**

L'elemento contenitore

**<STYLE> ... </STYLE>**

permette di inserire regole di stile nell'intestazione del documento. L'HTML permette di inserire un gran numero di elementi di stile nella sezione di intestazione del documento.

I browser che non supportano i fogli di stile o non riconoscono un particolare linguaggio per i fogli di stile semplicemente ignorano le istruzioni contenute tra i tag **<STYLE>** e **</STYLE>**; sarebbe un errore visualizzare il testo contenuto tra i due tag come parte del documento. Alcuni linguaggi di fogli di stile hanno comandi per nascondere le istruzioni ai browser non conformi.

Alcune implementazioni dei fogli di stile permettono, mediante il tag **<STYLE>** una personalizzazione della pagina in maniera decisamente più dettagliata rispetto all'attributo **STYLE** dei vari elementi che possono costituire il body di un documento.

Mediante i CSS il tag **<STYLE>** può descrivere proprietà per uno specifico elemento HTML, come **<P>**, **<H1>**, etc ... oppure per tutti gli elementi HTML appartenenti ad una specifica classe, come quelli a cui è assegnato l'attributo **CLASS** con lo stesso nome, oppure, infine, per un singolo tag, per il quale è stato definito l'attributo **ID**.

Le regole di precedenza e di eredità delle proprietà dipende dal linguaggio dello specifico foglio di stile.

Il tag **<STYLE>** prevede due attributi.

#### 1. **TYPE**

Questo attributo permette di definire il linguaggio del foglio di stile e sovrascrive il corrente foglio di stile utilizzato. Un valore comune per questo attributo è **"text/css"**.

#### 2. **MEDIA**

Questo attributo specifica la destinazione per le informazioni di stile introdotte. Normalmente è lo schermo, ma può essere anche la stampante o l'audio. Il valore predefinito è **"screen"** e può assumere anche i valori **"projection"**, **"print"** o **"aural"**.

Vediamo alcuni esempi di utilizzo del tag **<STYLE>** per definire le proprietà di altri elementi HTML.

L'esempio che segue mostra uno stile che mette un bordo di 1 pixel attorno ad ogni elemento **<H1>** e lo visualizza centrato.

```
<HEAD>
<STYLE type="text/css">
  H1 {border-width: "1"; border: "solid"; text-align: "center"}
</STYLE>
</HEAD>
...
```

In quest'altro esempio il bordo è applicato ai soli titoli appartenenti alla classe "miaclasse".

```
<HEAD>
<STYLE type="text/css">
  H1.miaclasse {border-width: "1"; border: "solid"; text-align: "center"}
</STYLE>
</HEAD>

<BODY>
<H1 class="miaclasse">Questo titolo sarà visualizzato col nuovo
stile</H1>
<H1>Questo titolo ignorerà il nuovo stile</H1>
</BODY>
...
```

Infine questo esempio limita l'applicazione dello stile al solo tag di nome "myid".

```
<HEAD>
<STYLE type="text/css">
  #myid {border-width: "1"; border: "solid"; text-align: "center"}
</STYLE>
</HEAD>

<BODY>
<H1 class="miaclasse">Questo titolo non sarà interessato dallo
stile</H1>
<H1 id="myid">Questo titolo sarà visualizzato col nuovo stile</H1>
<H1>Questo titolo ignorerà il nuovo stile</H1>
</BODY>
...
```

Gli esempi che seguono mostrano invece come indirizzare lo stile a particolari destinazioni, come il proiettore, la stampante o l'uscita audio.

L'esempio che segue applica al titolo **<H1>** uno stile diverso a seconda che la pagina venga proiettata o stampata. Il titolo apparirà di colore blu sul proiettore oppure centrato se stampato.

```
<HEAD>
<STYLE TYPE="text/css" MEDIA="projection">
  H1 {color="blue"}
</STYLE>
<STYLE TYPE="text/css" MEDIA="print">
  H1 {text-align:"center"}
</STYLE>
</HEAD>
...
```

Quest'ultimo invece applica un effetto sonoro ai collegamenti ipertestuali.

```
<HEAD>
<STYLE TYPE="text/css" MEDIA="aural">
  A {cue-before: "uri(bell.aiff)"; cue-after: "uri(dong.wav)"}
</STYLE>
</HEAD>
...
```

È buona norma mettere in un file esterno le descrizioni dei fogli di stile e collegare successivamente il file al documento mediante il tag **<LINK>**. Questa pratica, rispetto all'inserimento dei fogli di stile nell'intestazione del documento, permette una maggiore parametrizzazione di tutti i documenti di un sito e permette di cambiare le caratteristiche di molte pagine contemporaneamente semplicemente modificando il solo file contenente i fogli di stile. Se le informazioni di stile fossero contenute nell'intestazione di ciascun documento, le modifiche andrebbero cercate e applicate a ciascuna pagina singolarmente.

Vedi anche i tag **<DIV>**, **<LINK>**, **<SPAN>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<SUB>**

L'elemento HTML

**<SUB>** ... **</SUB>**

visualizza il testo come pedice, come mostrato negli esempi sottostanti.

Gli elementi di una matrice possono essere indicizzati con  $a_{ij} \dots a_{nn}$ , e questo è il codice utilizzato:

Gli elementi di una matrice possono essere indicizzati con  
`a<SUB>ii</SUB>` .. `a<SUB>nn</SUB>`.

Gli effetti dell'inserimenti di un tag **<SUB>** dentro l'altro non sono prevedibili, e dipendono dal browser utilizzato.

$a_i a_{i_i} a_{i_{i_i}} a_{i_{i_{i_i}}}$

### Browser

Netscape Navigator e Internet Explorer sembrano in grado di un numero indefinito di annidamenti. Mozilla sembra in grado di riprodurre indefiniti annidamenti, ma storpia i caratteri dopo il secondo livello di annidamento. Opera e Ariadna ignorano il tag.

Vedi anche il tag **<SUP>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<SUP>**

L'elemento HTML

**<SUP> ... </SUP>**

visualizza il testo ad apice, come mostrato negli esempi sottostanti.

Il Teorema di Pitagora stabilisce  $x^2 + y^2 = z^2$  in cui  $z$  è l'ipotenusa e  $x$ ,  $y$  sono i cateti di un triangolo rettangolo. e questo è il codice utilizzato per scrivere l'equazione:

```
x<SUP>2</SUP> + y<SUP>2</SUP> = z<SUP>2</SUP>
```

Gli effetti dell'inserimento di un tag **<SUP>** dentro l'altro non sono prevedibili, e dipendono dal browser utilizzato.

$$2^2 = 4$$

$$2^{2^2} = 16$$

$$2^{2^{2^2}} = 65536$$

$$2^{2^{2^{2^2}}} = \text{un numero enorme}$$

### Browser

Netscape Navigator, Mozilla e Internet Explorer sembrano in grado di un numero indefinito di annidamenti. Opera e Ariadna ignorano il tag.

Vedi anche il tag **<SUB>** e le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<TABLE>**

L'elemento HTML

**<TABLE> ... </TABLE>**

è un tag contenitore utilizzato per definire una tabella.

Il tag ha un gran numero di attributi, i più comuni dei quali sono:

**1. BORDER**

definisce lo spessore in pixel del bordo disegnato attorno alla tabella.

**2. CELLPADDING**

definisce lo spazio in pixel tra il contenuto della cella ed il bordo della stessa .

**3. CELLSPACING**

definisce lo spazio in pixel tra una cella e le celle vicine.

**4. ALIGN**

controlla l'allineamento della tabella; può assumere i valori **"left"**, **"center"** o **"right"**, secondo lo standard HTML 3.2.

**5. BORDERCOLOR**

questo utile attributo non standard HTML 3.2 controlla il colore del bordo della tabella, utilizzando le notazioni standard dei colori.

**6. BORDERCOLORLIGHT e BORDERCOLORDARK**

questi due attributi non standard HTML 3.2 consentono un controllo più preciso dell'aspetto della tabella. Un normale aspetto 3D di una tabella mostra i bordi sinistro e superiore più chiari dei bordi destro e inferiore, come se l'illuminazione provenisse dall'angolo in alto a sinistra. Questi due attributi possono impostare i colori utilizzati per ottenere questo effetto.

**7. RULES**

questo attributo non standard HTML 3.2 può essere utilizzato per cancellare alcune delle linee che separano le celle di una tabella.



I valori possibili sono 4, **"rows"**, **"cols"**, **"all"** e **"none"**, per indicare che vengono visualizzate le sole linee tra le righe, tra le colonne, tutte le linee o nessuna linea.

Roma	Napoli
Milano	Torino

La tabella qui a lato mostra un esempio in cui tutte le linee interne sono eliminate. Sebbene le linee sono state soppresse lo spazio tra le celle continua a persistere; in questo caso è stato impostato **RULES="NONE"** e **CELLSPACING=15**.

## 8. FRAME

questo attributo non standard HTML 3.2 permette la totale o parziale soppressione della cornice attorno alla tabella. I possibili valori sono **ABOVE**, **BELOW**, **LHS**, **RHS**, **SIDES**, **VSIDES**, **HSIDES** e **BOX** ed indicano la parte che verrà disegnata.

Roma	Napoli
Milano	Torino

L'esempio a lato mostra una tabella con **FRAME="HSIDES"**, ed infatti viene visualizzata una tabella con i bordi verticali nascosti.

## Browser

Solo Internet Explorer, Ariadna e Netscape Navigator riconoscono l'attributo **BORDERCOLOR**. Tutti i browser mostrano le tabelle con effetti 3d per bordi di spessore superiore a 5, e la possibilità di nascondere il bordo è fornita solo da Mosaic. Solo Internet Explorer riconosce gli attributi **RULES**, **FRAME**, **BORDERCOLORLIGHT** e **BORDERCOLORDARK**.

Vedi anche i tag **<TD>**, **<TH>**, **<TR>** e le note generali sulle tabelle.

## Tutti i tag

### Il tag **<TD>**

L'elemento contenitore

**<TD> ... </TD>**

è un tag usato per inserire un elemento della tabella.

Il tag può essere accompagnato da 5 attributi:

#### 1. **ALIGN**

Questo attributo può essere usato per controllare l'allineamento orizzontale di un elemento all'interno della cella.

Ecco un esempio di utilizzo del tag:

<b>Intestazione 1</b>	<b>Intestazione 2</b>	<b>Intestazione 3</b>
Elemento 1	Elemento 2	Elemento 3

E qui di seguito ne è mostrato il codice:

```
<TABLE>
<TR>
  <TH>Intestazione 1</TH><TH>Intestazione 2</TH><TH>Intestazione
3</TH>
</TR>
<TR>
  <TD ALIGN=LEFT>Elemento 1</TD><TD ALIGN=CENTER>Elemento
2</TD><TD ALIGN=RIGHT>Elemento 3</TD>
</TR>
</TABLE>
```

#### 2. **BGCOLOR**

Questo può essere usato per evidenziare il contenuto di una cella con un particolare colore di sfondo, da indicare in esadecimale o con il suo nome.

<b>Intestazione 1</b>	<b>Intestazione 2</b>	<b>Intestazione 3</b>
Elemento 1 (nero)	Elemento 2 (rosso)	Elemento 3 (verde)

Qui di seguito è mostrato il codice:

```
<TABLE>
<TR>
```

```

<TH>Intestazione 1</TH><TH>Intestazione 2</TH><TH>Intestazione
3</TH>
</TR>
<TR>
<TD BGCOLOR="black">Elemento 1 (nero)</TD><TD
BGCOLOR="700000">Elemento 2 (rosso)</TD><TD
BGCOLOR="007000">Elemento 3 (verde)</TD>
</TR>
</TABLE>

```

### 3. VALIGN

Questo attributo permette di controllare l'allineamento verticale di un elemento all'interno di una cella.

I valori possibili sono **TOP**, **MIDDLE** e **BOTTOM**, per allineare in alto, al centro e in basso; l'impostazione predefinita è in alto.

Ecco un esempio di utilizzo del tag:

Intestazione 1	Intestazione 2	Intestazione 3
Centro		Basso

E qui di seguito ne è mostrato il codice:

```

<TABLE>
<TR>
<TH>Intestazione 1</TH><TH>Intestazione 2</TH><TH>Intestazione
3</TH>
</TR>
<TR>
<TD VALIGN=MIDDLE>Centro</TD><TD><IMG
SRC=bandiera.jpg></TD><TD VALIGN=BOTTOM>Basso</TD>
</TR>
</TABLE>

```

### 4. NOWRAP

Normalmente il testo inserito in una cella va a capo a discrezione del browser. Tale comportamento può essere modificato mediante l'attributo **NOWRAP**. Se questo attributo, che non ha valori, è presente il testo non andrà a capo a meno che non sarà inserito il tag **<BR>**.

Colonna 1	Colonna 2	Colonna 3
Testo lungo che viene spezzato e mandato a capo.	Testo lungo che non viene spezzato.	Testo

Qui sotto la stessa tabella con la cella centrale senza l'attributo **NOWRAP**.

Colonna 1	Colonna 2	Colonna 3
Testo lungo che viene spezzato e mandato a capo.	Testo lungo che viene spezzato.	Testo

Vediamo come la larghezza delle celle si modifica a seconda della lunghezza delle frasi in esse contenute.

## 5. BACKGROUND

Questo attributo non standard HTML 3.2 permette di inserire una immagine come sfondo per una cella. Esso si aspetta un valore che è l'indirizzo relativo o assoluto (l'URL) della immagine.

Intestazione 1	Intestazione 2	Intestazione 3
Elemento 1	Elemento 2	Elemento 3

Qui di seguito ne è mostrato il codice:

```
<TABLE>
<TR>
  <TH>Intestazione 1</TH><TH>Intestazione 2</TH><TH>Intestazione
3</TH>
</TR>
<TR>
  <TD>Elemento 1</TD><TD BACKGROUND="bandiera.jpg">Elemento
2</TD><TD>Elemento 3</TD>
</TR>
</TABLE>
```

## 6. WIDTH

Questo attributo permette di specificare la larghezza della cella, in pixel o in percentuale della dimensione della tabella. Poiché le celle inferiori della tabella si adattano al contenuto della riga di dimensioni maggiori o alle celle le cui dimensioni sono fissate, è bene indicare la larghezza di tutti

gli elementi di una riga (ad esempio la prima) e lasciare che tutte le righe successive vi si adattino.

Colonna 1	Colonna 2	Colonna 3
width=30%	width=50%	width=20%
Testo	Testo	Testo

Qui sotto la stessa tabella con dimensioni diverse.

Colonna 1	Colonna 2	Colonna 3
width=15%	width=15%	width=70%
Testo	Testo	Testo

## Browser

Solo Internet Explorer e Netscape Navigator riconoscono l'attributo **BACKGROUND**. Ariadna non riconosce l'attributo **NOWRAP**.

Vedi anche i tag **<TABLE>**, **<TH>**, **<TR>**, **<CAPTION>** e le note generali sulle tabelle.

## Tutti i tag

### Il tag **<TEXTAREA>**

L'elemento HTML

**<TEXTAREA> ... </TEXTAREA>**

inserisce una finestra rettangolare nella quale l'utente può scrivere un testo su più righe.

Ecco di seguito un esempio:



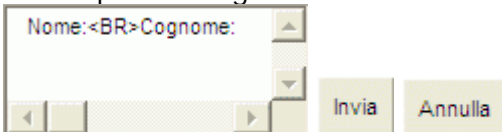
e questo è il codice utilizzato:

```
<FORM ACTION="" METHOD="POST">
<TEXTAREA NAME="testo" COLS="24" ROWS="3"></TEXTAREA>
<INPUT TYPE="submit" VALUE="Invia"><INPUT TYPE="reset"
VALUE="Annulla">
</FORM>
```

Gli attributi **COLS** e **ROWS** specificano le dimensioni della casella in termini di caratteri e sono gli unici attributi supportati.

Un qualunque testo (non formattato) inserito tra i tag di apertura e chiusura **<TEXTAREA>** e **</TEXTAREA>** apparirà come testo iniziale all'interno della casella.

L'esempio che segue mostra l'effetto dell'inserimento del testo:



Sfortunatamente l'utente può modificare ed eventualmente cancellare il testo pre-inserito nella **textarea**, così questa tecnica realizza solo un meccanismo ideale di input.

Per inserire un ritorno a capo occorre scrivere il codice macro del carattere speciale corrispondente al codice ascii del ritorno a capo, **&#13;**.

Vedi anche i tag **<FORM>**, **<INPUT>**, **<SELECT>** e le note sull'interfaccia CGI.

## Tutti i tag

### Il tag **<TH>**

L'elemento contenitore

**<TH>...</TH>**

è usato per inserire una intestazione alla tabella.

Come per il tag **<TD>** questo tag può essere accompagnato da 5 attributi:

#### 1. **ALIGN**

Questo attributo può essere usato per controllare l'allineamento orizzontale dell'intestazione all'interno della cella.

Ecco un esempio di utilizzo del tag:

<b>Intestazione 1</b>	<b>Intestazione 2</b>	<b>Intestazione 3</b>
Allineamento a sinistra	Allineamento al centro	Allineamento a destra

E qui di seguito ne è mostrato il codice:

```
<TABLE>
<TR>
  <TH ALIGN=LEFT>Intestazione 1</TH><TH
ALIGN=CENTER>Intestazione 2</TH><TH ALIGN=RIGHT>Intestazione
3</TH>
</TR>
<TR>
  <TD>Allineamento a sinistra</TD><TD>Allineamento al
centro</TD><TD>Allineamento a destra</TD>
</TR>
</TABLE>
```

#### 2. **BGCOLOR**

Questo può essere usato per evidenziare il contenuto di una cella con un particolare colore di sfondo, da indicare in esadecimale o con il suo nome.

<b>Intestazione 1</b> (nero)	<b>Intestazione 2</b> (rosso)	<b>Intestazione 3</b> (verde)
Elemento 1	Elemento 2	Elemento 3

Qui di seguito è mostrato il codice:

```
<TABLE>
<TR>
  <TH BGCOLOR=black>Intestazione 1 (nero)</TH><TH BGCOLOR
=700000>Intestazione 2 (rosso)</TH><TH bgcolor=007000>Intestazione
3 (verde)</TH>
</TR>
<TR>
  <TD>Elemento 1</TD><TD>Elemento 2</TD><TD>Elemento 3</TD>
</TR>
</TABLE>
```

### 3. VALIGN

Questo attributo permette di controllare l'allineamento verticale dell'intestazione all'interno di una cella. I valori possibili sono **TOP**, **MIDDLE** e **BOTTOM**, per allineare in alto, al centro e in basso; l'impostazione predefinita è in alto. Come per il tag **<TD>** l'effetto dell'applicazione di questo attributo si nota in presenza di immagini o di celle di dimensioni verticali diverse.

### 4. NOWRAP

Normalmente il testo inserito in una cella va a capo a discrezione del browser. Tale comportamento può essere modificato mediante l'attributo **NOWRAP**. Se questo attributo, che non ha valori, è presente il testo non andrà a capo a meno che non sarà inserito il tag **<BR>**.

### 5. BACKGROUND

Questo attributo non standard HTML 3.2 permette di inserire una immagine come sfondo per una cella. Esso si aspetta un valore che è l'indirizzo relativo o assoluto (l'URL) della immagine.

Intestazione 1	Intestazione 2	Intestazione 3
Elemento 1	Elemento 2	Elemento 3

Qui di seguito ne è mostrato il codice:

```
<TABLE>
<TR>
  <TH>Intestazione 1</TH><TH
BACKGROUND="bandiera.jpg">Intestazione 2</TH>
<TH>Intestazione 3</TH>
```



```

</TR>
<TR>
  <TD>Elemento 1</TD><TD>Elemento 2</TD><TD>Elemento 3</TD>
</TR>
</TABLE>

```

## 6. WIDTH

Questo attributo permette di specificare la larghezza della cella, in pixel o in percentuale della dimensione della tabella.

Poiché le celle inferiori della tabella si adattano al contenuto della riga di dimensioni maggiori o alle celle le cui dimensioni sono fissate, è bene indicare la larghezza di tutti gli elementi di una riga (ad esempio la prima) e lasciare che tutte le righe successive vi si adattino.

<b>width=30%</b>	<b>width=50%</b>	<b>width=20%</b>
Colonna 1	Colonna 2	Colonna 3
Testo	Testo	Testo

Qui sotto la stessa tabella con dimensioni diverse.

<b>width=15%</b>	<b>width=15%</b>	<b>width=70%</b>
Colonna 1	Colonna 2	Colonna 3
Testo	Testo	Testo

## Browser

Solo Internet Explorer e Netscape Navigator riconoscono l'attributo **BACKGROUND**. Ariadna non riconosce l'attributo **NOWRAP**.

Vedi anche i tag **<TABLE>**, **<TD>**, **<TR>**, **<CAPTION>** e le note generali sulle tabelle.

## Tutti i tag

### Il tag **<TITLE>**

L'elemento HTML

**<TITLE>** testo **</TITLE>**

è usato per specificare il titolo del documento. E' usato all'interno dei tag **<HEAD>** ... **</HEAD>** e quindi non influisce sulla visualizzazione della pagina del documento.

Il testo associato al tag è visualizzato nella barra del titolo della finestra del browser e comunemente può essere utilizzato dal browser nelle liste di navigazione e nei file di bookmark (Preferiti). Molti browser possono mostrare solo parte di un testo lungo, così è importante scegliere un titolo appropriato abbastanza corto per descrivere una pagina.

Vedi anche il tag **<HEAD>**.

## Tutti i tag

### Il tag `<TR>`

Il tag

`<TR> ... </TR>`

è usato per definire una riga di una tabella.

L'elemento è un tag contenitore e la coppia `<TR> ... </TR>` dovrebbe contenere una sequenza di contenitori `<TD> ... </TD>`.

Il tag `<TR>` ha senso solo se inserito all'interno di un contenitore `<TABLE> ... </TABLE>`.

Nella pratica la presenza di un tag `<TR>` implica la chiusura di una precedente riga di tabella e l'inizio di una nuova, per cui il tag di chiusura `</TR>` non è strettamente necessario.

Lo standard HTML 3.2 definisce due attributi, che controllano l'allineamento delle celle nella riga e il colore di fondo della riga. Questi possono anche essere sovrascritti dagli attributi delle celle nei tag `<TD>`.

#### 1. **ALIGN**

Questo attributo può essere usato per controllare l'allineamento orizzontale di tutte le celle all'interno della riga. Può assumere i valori **LEFT**, **CENTER** e **RIGHT** e l'impostazione predefinita è a sinistra.

#### 2. **VALIGN**

Questo attributo controlla l'allineamento verticale del contenuto di tutte le celle della riga. Può assumere i valori **TOP**, **MIDDLE** e **BOTTOM** e l'impostazione predefinita è in alto.

#### 3. **BGCOLOR**

Questo tag non è incluso nello standard HTML 3.2 e può essere usato per evidenziare il contenuto di tutte le celle della riga con un particolare colore di sfondo, da indicare in esadecimale o con il suo nome.

Ecco alcuni esempi di utilizzo degli attributi.

Prima abbiamo una tabella di tre righe che mostra i tre diversi stili di allineamento orizzontale, con l'allineamento verticale lasciato predefinito.

L'attributo **BGCOLOR** è utilizzato nell'ultima riga (con l'ultima cella che ne cambia nuovamente il colore).







Cella 11	Cella 12	Cella 13
Cella 21	Cella 22	Cella 23
Cella 31	Cella 32	Cella 33

Qui di seguito è mostrato il codice:

```
<TABLE WIDTH=400>
<TR ALIGN=LEFT>
  <TD WIDTH=33%>Cella 11</TD><TD WIDTH=33%>Cella 12</TD><TD
WIDTH=33%>Cella 13</TD>
</TR>
<TR ALIGN=CENTER>
  <TD>Cella 21</TD><TD>Cella 22</TD><TD>Cella 23</TD>
</TR>
<TR ALIGN=RIGHT BGCOLOR=700000>
  <TD>Cella 31</TD><TD>Cella 32</TD><TD BGCOLOR=007000>Cella
33</TD>
</TR>
</TABLE>
```

Notiamo l'uso dell'attributo **WIDTH** all'interno dei tag **<TABLE>** e **<TD>**.

Ecco un altro esempio in cui mostriamo l'allineamento verticale, visibile perché abbiamo utilizzato elementi che rendono la riga più alta.

 	Cella 12	Cella 13
Cella 21	 	Cella 23
Cella 31	Cella 32	 

## Browser

Internet Explorer, Netscape Navigator, Mozilla e Opera riconoscono l'attributo **BGCOLOR**. Gli altri browser non sono in grado di cambiare il colore di fondo.

Vedi anche i tag **<TABLE>**, **<TD>**, **<TH>**, **<CAPTION>** e le note generali sulle tabelle.

## Tutti i tag

### Il tag **<TT>**

L'elemento HTML

**<TT> ... </TT>**

dice al browser di visualizzare il testo che segue mediante un formato non proporzionato.

Questo è in stile **<TT>**.

Questo stile è chiamato solitamente teletype ed è usato raramente. Il tag **<CODE>** viene preferito quando è richiesto l'effetto non proporzionato, tipicamente per riportare parti di testo estratti da programmi.

Vedi anche il tag **<CODE>**.

## Tutti i tag

### Il tag **<U>**

L'elemento HTML

**<U>** ... **</U>**

dice al browser di visualizzare il testo con una sottolineatura.

Lo standard HTML 3.2 definisce questo come uno stile fisico, distinguendolo da uno stile logico. Al contrario degli altri tag fisici come **<B>**, **<I>**, etc, questo tag non ha il corrispondente tag logico equivalente.

Questo testo è sottolineato.

Uno stile fisico impone al browser di visualizzare il testo in un particolare tipo di carattere, mentre lo stile logico dice di mostrare il testo con una enfasi diversa, lasciando al browser la libertà di decidere come ottenere lo stile. (possono essere usate sottolineature o colori diversi).

**NOTA.** Molti utenti impostano il proprio browser in modo tale che i collegamenti ipertestuale vengano mostrati sottolineati oppure in un colore distintivo, l'utilizzo del tag **<U>** potrebbe causare confusione a questi utenti.

Vedi anche le note generali sugli stili di testo.

## Tutti i tag

### Il tag **<UL>**

L'elemento HTML

**<UL>**

...

**</UL>**

è usato per indicare una lista non numerata. Una lista non ordinata consiste in una sequenza di elementi identificati tramite il tag **<LI>**.

Ciascun elemento è visualizzato mediante un bottone o un simbolo grafico simile all'inizio della riga.

Le liste possono essere annidate una dentro l'altra, e le liste interne verranno visualizzate con una indentatura maggiore e possibilmente con un simbolo diverso.

Per ogni tag **<UL>** usato dovrà essere presente un corrispondente **</UL>**.

Ecco un esempio di lista annidata:

- Livello 1, Elemento 1
- Livello 1, Elemento 2
- Livello 1, Elemento 3
  - ? Livello 2, Elemento 1
    - ☒ Livello 3, Elemento 1
    - ☒ Livello 3, Elemento 2
  - ? Livello 2, Elemento 2
    - ☒ Livello 3, Elemento A
    - ☒ Livello 3, Elemento B
      - Livello 4, Elemento 1
  - ? Livello 2, Elemento 3
    - ☒ Livello 3, Elemento 1
    - ☒ Livello 3, Elemento 2
- ? Livello 2, Elemento 4

Alcuni browser usano diversi tipi di bottone per distinguere i vari livelli di liste annidate, sebbene poi viene usato lo stesso tipo di bottone per indicare ciascun elemento a partire dal terzo livello in poi. Ma liste di profondità superiore a 3 risultano di difficile comprensione e sono raramente usate.

Il tag **<UL>** ha il solo attributo **TYPE**, che indica il tipo di bottone utilizzato per evidenziare l'elemento della lista. Questo attributo riconosce 3 possibili valori.

**1. CIRCLE**

Il bottone è un cerchio vuoto come mostrato nell'esempio qui sotto. Mozilla non visualizza un cerchio perfetto, ma un pallino di forma ovale.

- 1° elemento
- 2° elemento

**2. SQUARE**

Il bottone è un quadratino pieno come mostrato nell'esempio qui sotto.

- ◻ 1° elemento
- ◻ 2° elemento

**3. DISC**

Il bottone è un cerchio pieno. Mozilla non visualizza un cerchio perfetto, ma un pallino di forma ovale.

- 1° elemento
- 2° elemento

Vedi anche i tag **<LI>**, **<OL>**, **<DL>** e le note generali sulle liste.



## Tutti i tag

### Il tag **<VAR>**

L'elemento contenitore

**<VAR> ... </VAR>**

è utilizzato per riferimenti a variabili all'interno di programmi. Il significato non è del tutto chiaro nello standard HTML e il tag è usato raramente.

Il tag visualizza il testo, con carattere monospazio, esattamente come è stato inserito, ma non inserisce le interruzioni di riga se queste sono presenti nel testo; per andare a capo occorre il tag **<BR>**. I caratteri speciali HTML devono essere scritti tramite i loro codici. L'effetto si applica anche a blocchi di testo.

Ecco un esempio di utilizzo del tag:

```
1==1 1<=2 1>=0
```

e questo è il codice che lo genera:

```
<VAR>  
1==1  
1&lt;=2  
1&gt;=0  
</VAR>
```

Notiamo che non è la stessa visualizzazione del testo preformattato ottenuto con **<PRE>**.

Può essere usato anche all'interno di un paragrafo, come ad esempio: un *esempio* di testo.

Vedi anche i tag **<CITE>**, **<CODE>**, **<DFN>**, **<KBD>**, **<SAMP>** e le note generali sugli stili di testo.

**NOTA.** In alcuni browser questo tag ha lo stesso comportamento degli altri appena menzionati, e l'effetto è identico. Le differenze sono puramente storiche.

## Tutti i tag

### Il tag **<XMP>**

L'elemento contenitore

**<XMP> ... </XMP>**

è un tag obsoleto ed è utilizzato per inserire del testo da visualizzare con un carattere di larghezza costante, come il Courier.

Il tag visualizza il testo, con carattere monospazio, esattamente come è stato inserito e inserisce le interruzioni di riga se queste sono presenti nel testo.

In alcuni browser questo tag ha l'interessante proprietà di disattivare l'interprete dei tag successivi finché non viene incontrato il tag di chiusura **</XMP>**. Internet Explorer, Netscape Navigator e Mozilla hanno questo comportamento.

Ecco un esempio di utilizzo del tag:

Ecco un tag `<br>` ed eccone `<xmp>` un altro.

Qui ci sono 4 spazi all'inizio della riga.

Qui & è visualizzata la "E commerciale"

e questo è il codice che lo genera:

```
<XMP>
Ecco un tag <BR> ed eccone un <XMP> altro.
  Qui ci sono 4 spazi all'inizio della riga.
Qui & è visualizzata la "E commerciale"
</XMP>
```

Vedi anche i tag **<ADDRESS>**, **<BLOCKQUOTE>**, **<LISTING>**, **<PLAINTEXT>**, **<PRE>** e le note generali sugli stili di testo.

## Appendice A

### Tag, Attributi e Valori

Ogni documento HTML è composto da un testo da visualizzare tramite il browser più informazioni di formattazione che controllano le modalità in cui il testo viene visualizzato. Queste informazioni si chiamano TAG. Un tag consiste di un tipo che ne specifica il nome e da alcuni attributi più o meno opzionali ai quali generalmente è associato un valore. Ecco un esempio del tag per visualizzare una immagine:

```
<IMG SRC="foto.jpg" align="center">
```

Nell'esempio mostrato il tag è **IMG** e ha due attributi **SRC** e **ALIGN**, con i loro valori, rispettivamente **"foto.jpg"** e **"center"**.

#### ? Tag

I tag si distinguono dalle parentesi angolari **<** e **>**. All'interno della dichiarazione del tag la prima parola deve essere il nome del tag, che può essere scritto indistintamente in maiuscolo o minuscolo. Molti programmi per fare pagine web sembrano preferire il maiuscolo, e questo forse per fare in modo che i tag si possano distinguere più facilmente dal resto del testo.

Se un browser non riconosce un tag, esso semplicemente lo ignora, con tutti i suoi attributi. Molti tag sono *"contenitori"*, nel senso che vanno chiusi dopo il loro utilizzo, formando una coppia **<TAG> ... </TAG>**. In alcuni casi la chiusura del tag può essere opzionale e viene chiuso dal successivo tag incontrato. Gli esempi più comuni sono costituiti dai tag **<LI>**, **<P>** e **<TD>**; per questi tag non è necessario scrivere il tag di chiusura, ma la presenza di un nuovo tag in apertura implica l'inizio di un nuovo blocco. Questo comportamento implica anche la impossibilità di annidamento di questi tag; ad esempio non si può avere un paragrafo dentro un altro paragrafo, ma solo una sequenza di paragrafi.

#### ? Attributi

Ciascun tag si aspetta determinati attributi per poter applicare le proprie funzionalità ai vari blocchi di testo. Se un browser non riconosce un attributo, questo viene

ignorato, insieme al suo valore. Il nome dell'attributo può essere scritto indifferentemente in maiuscolo o minuscolo.

### ? **Valori**

Non tutti gli attributi aspettano valori, sebbene la maggior parte li richieda. Il valore viene applicato all'attributo dal simbolo di assegnazione = e dovrebbe, di norma, essere contenuto tra virgolette ", soprattutto se contiene anche caratteri non alfanumerici come \$-\_.+!\*'().

La possibilità di accettare caratteri maiuscoli o minuscoli dipende dal particolare attributo. I valori normalmente "case sensitive" sono quelli che contengono annotazioni, URL o parti di esse o l'attributo **TYPE** del tag **<OL>**.

Non è completamente chiaro dallo standard HTML se la presenza dello spazio all'apertura del tag, tra la parentesi angolare di apertura < e il nome del tag, sia permessa, così come tra il nome degli attributi, il simbolo di assegnazione = e il valore.

## Appendice B

### Le intestazioni

Lo standard HTML prevede sei stili di intestazioni, noti come **H1**, **H2**, **H3**, **H4**, **H5** e **H6**. Sono utilizzati come titolo di una pagina, di un capitolo, come intestazione per una tabella o per una immagine, etc.

# Titolo di livello 1

## Titolo di livello 2

### Titolo di livello 3

#### Titolo di livello 4

##### Titolo di livello 5

###### Titolo di livello 6

Tutte le intestazioni vengono visualizzate con una interlinea doppia o di una riga e mezza sopra e sotto. Non tutti i browser visualizzano le intestazioni allo stesso modo. Nella pratica le intestazioni **H4**, **H5** e **H6** sono usate raramente, perché sono pressoché delle stesse dimensioni del testo normale, e si confondono con esso.

A partire dall'HTML 3.2 è stato previsto l'attributo **ALIGN** per ciascun tag di intestazione. I valori permessi sono "**left**", "**center**", "**right**" e "**justify**", anche se quest'ultimo non ha molto senso, trattandosi di titoli.

Ecco qui sotto un esempio di utilizzo dell'allineamento, nel caso del tag **<H3>**:

### Titolo di livello 3, a sinistra

### Titolo di livello 3, a centro

### Titolo di livello 3, a destra

## Appendice C

### Gli stili di testo

Lo standard HTML prevede non meno di 23 tag il cui scopo è controllare l'aspetto del testo visualizzato. Molti di questi, tuttavia, sono obsoleti ed usati raramente. Possiamo dividere questi tag in due gruppi a seconda che inseriscano o meno una interruzione di riga. Tutti i tag del primo gruppo non aspettano attributi.

### Tag per blocchi di testo

Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco <BR> di due righe." viene inserito nella spiegazione per mostrarne l'effetto.

<b>ADDRESS</b>	Contiene un indirizzo e viene normalmente visualizzato con un carattere in corsivo; non viene posta interlinea prima e dopo il blocco.	Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco di due righe." viene inserito nella spiegazione per mostrarne l'effetto.
<b>BLOCKQUOTE</b>	Contiene un blocco di testo che viene evidenziato rispetto al resto del testo e viene normalmente visualizzato con una indentatura rispetto al resto della pagina; viene posta interlinea doppia prima e dopo il blocco.	Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco di due righe." viene inserito nella spiegazione per mostrarne l'effetto.
<b>LISTING</b>	Racchiude un testo preformattato. È un tag obsoleto; viene posta una interlinea doppia prima e dopo il blocco.	Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco di due righe." viene inserito nella spiegazione per mostrarne l'effetto.
<b>PLAINTEXT</b>	È usato per indicare la fine della parte HTML del documento e da quel punto in poi non riconosce più alcun altro tag. Racchiude un testo preformattato. È un tag obsoleto e viene raramente utilizzato; viene posta una interlinea doppia prima del blocco (poiché il blocco finisce con la fine del file non ha senso parlare di fine del blocco). Moltissimi	Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco  di due righe." viene inserito nella spiegazione per mostrarne l'effetto. </TD> </TR> <TR> ...

	browser si comportano in maniera non prevedibile quando incontrano questo tag ed è meglio evitarlo.	
<b>PRE</b>	È usato per inserire un blocco di testo preformattato; viene posta una interlinea di una riga e mezza prima e dopo il blocco.	Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco di due righe."  viene inserito nella spiegazione per mostrarne l'effetto.
<b>XMP</b>	È usato per inserire un blocco di testo da visualizzare con carattere monospazio. I tag intermedi vengono ignorati finché non si chiude il blocco; viene posta una interlinea di una riga e mezza prima e dopo il blocco.	Questo gruppo di sei tag inserisce una interruzione di riga. Il loro effetto è illustrato nella tabella sottostante ed in ciascun caso la frase "Questo è un blocco   di due righe."  viene inserito nella spiegazione per mostrarne l'effetto.

Notiamo che solo i tag **<XMP>** e **<PLAINTEXT>** disabilitano l'interpretazione di tag intermedi, mentre con i browser precedenti al 1997 ciò avveniva anche con il tag **<LISTING>**. Quest'ultimo mostra in genere il testo con un carattere più piccolo.

## Tag in linea

Questo gruppo di 17 tag controlla l'aspetto del testo all'interno di ciascuna riga. Il tag **<FONT>** è il più completo e può essere utilizzato per ottenere la maggior parte degli effetti mostrati qui. Non sarà esaminato in questo capitolo.

Questi tag possono essere suddivisi in due grandi gruppi, noti come stile **logico** e stile **fisico** oppure come stile carattere e paragrafo. Lo scopo dei tag fisici è quello di controllare direttamente l'aspetto fisico del testo mentre i tag logici sono soggetti all'interpretazione da parte del browser. In particolare i tag logici sono usati da browser non convenzionali per visualizzare il testo in modalità speciali in modo che possano essere utilizzati da utenti con problemi fisici o per guidare l'interpretazione del testo da parte di programmi di indicizzazione.

Ecco la lista dei 17 tag con un esempio del loro effetto sul testo. Generalmente è possibile annidare questi tag in modo da produrre effetti più complessi, ma gli effetti possono essere imprevedibili.

Lasciamo al lettore la curiosità e il divertimento di sperimentare i vari comportamenti.

Nella tabella che segue, la frase *"mi fu quest'ermo"* sarà sottoposta al tag in questione, per mostrarne l'effetto e la differenza col testo normale.

<b>B</b>	Fisico	Grassetto	Sempre caro <b>mi fu quest'ermo</b> colle.
<b>BIG</b>	Fisico	Più grande di un'unità, equivalente a <b>&lt;FONT SIZE=+1&gt;</b>	Sempre caro <b>mi fu quest'ermo</b> colle.
<b>CITE</b>	Logico	Citazione	Sempre caro <i>mi fu quest'ermo</i> colle.
<b>CODE</b>	Logico	Codice, usato per riportare il codice di qualche programma	Sempre caro <code>mi fu quest'ermo</code> colle.
<b>DFN</b>	Logico	Definizione, usato per riportare la definizione di un termine	Sempre caro <i>mi fu quest'ermo</i> colle.
<b>EM</b>	Logico	Enfasi	Sempre caro <i>mi fu quest'ermo</i> colle.
<b>I</b>	Fisico	Corsivo	Sempre caro <i>mi fu quest'ermo</i> colle.
<b>KBD</b>	Logico	Tastiera, usato per riportare i caratteri digitati dalla tastiera	Sempre caro <code>mi fu quest'ermo</code> colle.
<b>SAMP</b>	Logico	Esempio, usato per riportare output di programmi	Sempre caro <code>mi fu quest'ermo</code> colle.
<b>SMALL</b>	Fisico	Più piccolo di un'unità, equivalente a <b>&lt;FONT SIZE=-1&gt;</b>	Sempre caro <small>mi fu quest'ermo</small> colle.
<b>STRIKE</b>	Fisico	Barrato, con una linea attraverso	Sempre caro <del>mi fu quest'ermo</del> colle.
<b>STRONG</b>	Logico	Più scuro	Sempre caro <b>mi fu quest'ermo</b> colle.
<b>SUB</b>	Fisico	Pedice	Sempre caro <sub>mi fu quest'ermo</sub> colle.
<b>SUP</b>	Fisico	Apice	Sempre caro <sup>mi fu quest'ermo</sup> colle.
<b>TT</b>	Fisico	Teletype, come da macchina da scrivere	Sempre caro <code>mi fu quest'ermo</code> colle.
<b>U</b>	Fisico	Sottolineato	Sempre caro <u>mi fu quest'ermo</u> colle.
<b>VAR</b>	Logico	Variabile, come le variabili nella documentazione di un software	Sempre caro <i>mi fu quest'ermo</i> colle.



## Appendice D

### Le immagini

La possibilità di inserire immagini è una delle caratteristiche che ha reso il World Wide Web così attraente per molti utenti. Però ci sono alcuni punti che devono essere valutati prima di utilizzare immagini in maniera intensiva.

#### ? **Dimensione delle immagini**

Se si inserisce una immagine in linea con il testo, allora tutto il file contenente l'immagine verrà trasferito dal server al browser. Anche una sola immagine piccola può pesare 5 o 10 KByte, che è la dimensione di una pagina anche molto grande di puro testo. L'inserimento di una immagine può anche raddoppiare il tempo di visualizzazione della pagina.

Le immagini ottenute da fotografie digitalizzate o da programmi di grafica sono molto più pesanti di semplici immagini. Molti utenti, soprattutto su linee vecchie e lente, disabilitano la visualizzazione delle immagini, anche se la cosa ormai avviene molto raramente. In ogni caso è bene ricordare anche questo fattore.

#### ? **Ripetizione di immagini**

Se si inserisce la stessa immagine diverse volte, l'immagine sarà trasferita al browser una volta sola, se ogni riferimento all'immagine usa lo stesso URL. La maggior parte dei browser infatti mantiene nella cache una copia locale delle immagini e queste non devono più essere caricate successivamente.

#### ? **Browser testuali**

Alcuni vecchi computer poco potenti utilizzano browser che visualizzano solo testo. Di questi il Lynx è il più conosciuto. Browser testuali sono utilizzati da persone con problemi fisici; è bene considerare anche essi quando si costruisce la propria pagina.

#### ? **Immagini offline**

Una immagine in linea è associata al tag **<IMG>**. Un semplice link ad una immagine può essere del tipo:

```
Ecco una <A HREF="immagine.jpg">immagine di  
Napoli</A>.
```

che farà in modo che l'immagine possa essere scaricata su richiesta.

Nel caso si voglia presentare le immagini in questo modo è buona norma di cortesia indicare anche la dimensione dell'immagine, ad esempio:

Ecco una `<A HREF="immagine.jpg">immagine di Napoli</A>` (247 KByte).

Il visitatore può decidere se scaricare o meno l'immagine, e consumare tempo e banda.

## Appendice E

### Le mappe immagine

Le mappe, conosciute anche come immagini cliccabili sono una caratteristica utile e molto adoperata nel creare pagine web. Una mappa è una immagine visualizzata in modo che il browser può determinare le coordinate del punto in cui l'utente clicca. Il punto selezionato può essere utilizzato per indirizzare ad un particolare URL associato a quel punto.

Un modo comune e naturale di utilizzo è quello di disegnare l'immagine come una mappa geografica e posizionare i link come le città e i paesi segnati sulla mappa. Le mappe possono essere anche usate per costruire un menu o una barra di navigazione.

Tutte le mappe necessitano il caricamento dell'immagine all'interno della pagina mostrata dal browser. Questa è strutturata in un sistema di coordinate che iniziano dall'angolo in alto a sinistra.

Moltissimi programmi di grafica possono mostrare queste coordinate. Le zone sensibili possono avere la forma circolare, rettangolare o poligonale.

Ci sono quattro modi per ottenere una mappa cliccabile.

#### 1. Lato client

Tutti gli URL e le aree sensibili sono definite nel codice HTML che viene inviato al browser e non c'è bisogno di alcun file speciale da definire sul server. È il modo migliore e più semplice per costruire una mappa immagine, finché il numero di aree rimane abbastanza piccolo. Per un numero di zone sensibili diventa molto alto servono particolari meccanismi di tipo CGI.

Le mappe lato client furono introdotte con lo standard HTML 3.2 ed alcuni vecchi browser non le supportano.

L'attributo **USEMAP** del tag **<IMG>** è utilizzato per indicare la dichiarazione di una mappa sensibile. La mappa sarà poi definita tra i tag **<MAP>** e **</MAP>** utilizzando il tag **<AREA>**.

Il valore associato all'attributo **USEMAP** è usato per identificare il particolare tag **<MAP>** che definisce la mappa, dal momento che più mappe immagine possono essere create nella stessa pagina. Il tag **<MAP>** usa l'attributo **NAME** per dichiarare il nome della mappa, allo

stesso modo in cui viene dichiarata un'ancora con il tag **<A>**.

## 2. Lato server

Le informazioni che definiscono le zone sensibili sono contenute in uno speciale file di mappatura che rimane sul server. Il client invia una richiesta indicando le coordinate del punto selezionato; il server quindi elabora il file di mappatura e restituisce un URL al client.

L'attributo **ISMAP** è associato al tag **<IMG>** all'interno di un contenitore **<A> ... </A>**. L'URL associato all'attributo **HREF** del tag **<A>** indica un file speciale sul server che riceve le coordinate e legge il file di mappatura. Il formato del file è specifico per ciascun programma ed il server deve essere configurato opportunamente per funzionare in modo corretto. Molti server richiedono che il file di mappatura abbia una estensione `.map` e che consista in una serie di coordinate che servano a delimitare figure all'interno di una immagine.

Ci sono 4 tipi possibili di figure lato server:

### 1. RECT

Questa figura definisce una zona rettangolare. Le coppie di coordinate richieste sono due e rappresentano i vertici in alto a sinistra e in basso a destra del rettangolo.

### 2. CIRCLE

Questa figura definisce un cerchio. Le coppie di coordinate richieste sono due ed indicano il centro del cerchio ed un qualunque punto della circonferenza. **Notiamo** come il cerchio **NON** sia definito utilizzando centro e raggio.

### 3. POLY

Questa definisce una zona a forma di poligono irregolare. È richiesta una coppia di coordinate per ciascuno dei vertici del poligono e la chiusura è implicita, per cui non c'è bisogno di ripetere l'ultima coppia di coordinate.

### 4. DEFAULT

Non sono richieste coordinate, ed indica una zona non sensibile. È previsto in ogni caso l'invio di un

URL al client, che sarà costruito in modo da indicare il fatto che si è cliccato su una zona non consentita.

Il formato del file cambia da serve a server. Questo appena descritto è quello utilizzato comunemente sui server di tipo Apache. I server del Cern utilizzano le parentesi per le coppie di coordinate e le virgolette per il nome delle zone, che può essere anche scritto per esteso. Altri server possono richiedere che il nome del file abbia estensione `.mapx`.

### 3. Meccanismi CGI

Questo metodo è identico al precedente. La differenza principale consiste nel fatto che il link è elaborato mediante il meccanismo CGI.

Il vantaggio di questo metodo è la possibilità di implementare zone più complesse di figure geometriche su una rappresentazione bitmap dell'immagine.

### 4. Immagini INPUT nei form

Il tag **<INPUT>** utilizzato all'interno di un form ha l'attributo **TYPE="image"** che permette di creare una immagine con la funzione di "submit" il cui valore associato viene riportato ad un file sul server alla stregua di un collegamento di tipo CGI.

Ciò ha il vantaggio del meccanismo CGI appena descritto con il vantaggio che l'utente può inserire anche altre informazioni nelle altre caselle di input. Ad esempio può essere possibile fornire un meccanismo per selezionare un punto dell'immagine piuttosto che un altro a seconda di particolari condizioni indicate in una casella di tipo **<SELECT>**. Occorrerà fare in modo che l'utente non clicchi la mappa prima di aver completato l'inserimento dei valori nelle altre caselle.

## Appendice F

### Le tabelle

L'HTML e tutti i browser non vecchissimi offrono la possibilità di visualizzare tabelle. Una tabella è definita dal tag contenitore **<TABLE>**. All'interno della coppia di tag la tabella è definita da una serie di righe mediante il tag **<TR>**. Ogni riga consiste in una sequenza di celle definite dai tag **<TD>** e **<TH>** per indicare rispettivamente i dati e le intestazioni.

All'interno di una cella può essere utilizzato qualunque contenuto HTML, come paragrafi, liste, immagini, iframe ed altre tabelle. La tabella avrà forma rettangolare e con un numero di colonne pari al numero massimo di celle in ciascuna riga. Una cella può espandersi andando ad occupare anche più righe o colonne.

Ecco un esempio di tabella semplice:

Colonna 1	Colonna 2
Elemento 11	Elemento 12
Elemento 21	Elemento 22

e questo è il codice che lo genera:

```
<TABLE BORDER=3>
<TR><TH>Colonna 1</TH><TH>Colonna 2</TH></TR>
<TR><TD>Elemento 11</TD><TD>Elemento 12</TD></TR>
<TR><TD>Elemento 21</TD><TD>Elemento 22</TD></TR>
</TABLE>
```

Alcuni programmatori web non chiudono i tag degli elementi interni alla tabella. La maggior parte dei browser è elastica nei confronti di questa pratica e consentono correttamente la visualizzazione della tabella, ma questo comportamento sarebbe da evitare perché può spesso indurre in errore.

Salvo diverse impostazioni le intestazioni appaiono in grassetto mentre i dati appaiono in testo normale.

### Espansione per colonne

Una cella può espandersi su più colonne mediante l'attributo **COLSPAN** associato ai tag **<TH>** e **<TD>**. Il valore di questo attributo è il numero di colonne che verranno occupate.

Eccone un esempio:

Rettangolo		Cerchio		
x	y	Centro	Raggio	
125	79	100	80	15
144	53	112	100	20

e questo è il codice che lo genera:

```
<TABLE BORDER=3>
  <TR><TH COLSPAN=2>Rettangolo</TH><TH COLSPAN=3>
<CENTER>Cerchio</CENTER></TH></TR>
  <TR><TH>x</TH><TH>y</TH><TH>Centro</TH><TH>Raggio</TH>
</TR>
  <TR><TD>125</TD><TD>79</TD><TD>100</TD><TD>80</TD>
<TD>15</TD></TR>
  <TR><TD>144</TD><TD>53</TD><TD>112</TD><TD>100</TD>
<TD>20</TD></TR>
</TABLE>
```

## Espansione per righe

Una cella può espandersi anche su più righe, mediante l'attributo **ROWSPAN** associato ai tag **<TH>** e **<TD>**. Il valore di questo attributo è il numero di righe che verranno occupate. È importante, mentre si compilano le celle delle righe successive, ricordarsi delle caselle già occupate dalle righe superiori, che nelle nuove righe non esisteranno.

Eccone un esempio:

Gruppo	Nazionalità	Numero
Studenti	Italiani	234
	Francesi	13
	Spagnoli	27
Personale	Italiani	14
	Spagnoli	7

e questo è il codice che lo genera:

```
<TABLE BORDER=3>
  <TR><TH>Gruppo</TH><TH>Nazionalità</TH><TH>Numero</TH></TR>
  <TR><TD
ROWSPAN=3>Studenti</TD><TD>Italiani</TD><TD>234</TD></TR>
  <TR><TD>Francesi</TD><TD>13</TD></TR>
```

```
<TR><TD>Spagnoli</TD><TD>27</TD></TR>
<TR><TD
ROWSPAN=2>Personale</TD><TD>Italiani</TD><TD>14</TD></TR>
<TR><TD>Spagnoli</TD><TD>7</TD></TR>
</TABLE>
```

Con un po' di fantasia e un minimo di attenzione è possibile combinare espansioni per righe e per colonne, ottenendo effetti grafici di maggiore effetto.

## Larghezza delle tabelle

Il calcolo della larghezza di una tabella è un conteggio piuttosto complesso da parte del browser che esamina la larghezza delle diverse colonne. Lo standard HTML permette di specificare la larghezza delle colonne, ma non tutti i browser reagiscono allo stesso modo.

La larghezza complessiva della tabella può essere impostata utilizzando l'attributo **WIDTH** del tag **<TABLE>**. Il valore inserito specifica la larghezza in pixel o in percentuale della finestra del browser.

La larghezza di una colonna è determinata dalla cella di dimensioni maggiori, la cui larghezza può essere influenzata dal contenuto della cella stessa, soprattutto se questo contenuto è una immagine o un'altra tabella. La larghezza di una cella può essere espresso esplicitamente mediante l'attributo **WIDTH** del tag **<TD>**.

Ecco qualche semplice esempio:

Regione	Capoluogo
Lazio	Roma
Piemonte	Torino
Toscana	Firenze
Campania	Napoli

e questo è il codice che lo genera:

```
<TABLE BORDER=3 WIDTH=50%>
<TR><TH WIDTH=50%>Regione</TH><TH
WIDTH=50%>Capoluogo</TH></TR>
<TR><TD>Lazio</TD><TD>Roma</TD></TR>
<TR><TD>Piemonte</TD><TD>Torino</TD></TR>
<TR><TD>Toscana</TD><TD>Firenze</TD></TR>
<TR><TD>Campania</TD><TD>Napoli</TD></TR>
</TABLE>
```

La dimensione delle colonne al 50% ciascuna è impostata nella



prima riga, nel tag **<TH>**, le righe successive ne ereditano la larghezza, adeguandosi.

Possiamo notare che allargando e restringendo la finestra del browser, la tabella continua ad occupare metà della larghezza.

Regione	Capoluogo
Lazio	Roma
Piemonte	Torino
Toscana	Firenze
Campania	Napoli

e questo è il codice che lo genera:

```
<TABLE BORDER=3 WIDTH=50%>
<TR><TH WIDTH=50%>Regione</TH><TH
WIDTH=50%>Capoluogo</TH></TR>
<TR><TD>Lazio</TD><TD>Roma</TD></TR>
<TR><TD>Piemonte</TD><TD>Torino</TD></TR>
<TR><TD>Toscana</TD><TD>Firenze</TD></TR>
<TR><TD>Campania</TD><TD>Napoli</TD></TR>
</TABLE>
```

Abbiamo fissato la larghezza della tabella a 250 punti e le colonne rimangono al 50% ciascuna. Anche ridimensionando la finestra del browser la tabella rimane delle stesse dimensioni.

## Celle vuote

Può capitare che una tabella venga costruita a partire da un file di dati e qualche cella può non contenere alcun dato. Lo standard HTML visualizza la cella con una superficie rialzata e senza bordi piuttosto che come una casella vuota. Ciò può sembrare sgradevole alla vista, come nell'esempio che segue.

Istruzione	Funzione	Note
strcmp	confronta due stringhe	
stricmp	confronta due stringhe trascurando i maiuscoli/minuscoli	Non ANSI
strcpy	copia una stringa in un'altra	
strlen	fornisce la lunghezza di una stringa	

Sarebbe bene mettere un trattino o un carattere simile in quelle posizioni, ma comunque è una operazione artificiosa. Lo standard HTML 3.2 suggerisce di mettere nella cella il carattere di "spazio

non interrompente" codificato con **&nbsp;**. Ecco di seguito lo stesso esempio con il simbolo dello spazio nelle caselle vuote.

Istruzione	Funzione	Note
strcmp	confronta due stringhe	
stricmp	confronta due stringhe trascurando i maiuscoli/minuscoli	Non ANSI
strcpy	copia una stringa in un'altra	
strlen	fornisce la lunghezza di una stringa	

### Proprietà di righe e colonne

Il tag **<TR>** ha diversi attributi che permettono un certo controllo sulle proprietà delle righe di una tabella. Sfortunatamente non c'è modo di specificare le proprietà di una colonna di tabella; questa utile caratteristica è stata introdotta con HTML 3.0, ma non è stata accettata da nessun browser. Il controllo delle proprietà di una colonna può essere ottenuto, pazientemente, impostando le proprietà richieste separatamente a tutte le celle della determinata colonna.

### Tipi di carattere

Potrebbe essere necessario utilizzare un carattere diverso per il testo inserito all'interno di una tabella. Non è possibile inserire la tabella nel tag **<FONT>** perché questo si riferisce solo ad elementi di testo. Lo stesso discorso vale per il tag **<BASEFONT>**. L'unico modo per cambiare il tipo di carattere in una tabella è quello di cambiare pazientemente il carattere in ciascuna casella della tabella. In alternativa è possibile impostare il tipo di carattere da utilizzare nelle celle tramite i fogli di stile, e per questo facciamo riferimento all'apposita guida sull'argomento.

### Annidamento di tabelle

All'interno di una cella è possibile inserire quasi qualunque contenuto di tipo HTML, inclusa un'altra tabella. L'annidamento di tabelle dentro altre celle è una pratica molto sfruttata per ottenere effetti grafici avanzati e presentazioni più complesse.

L'esempio che segue mostra un banale inserimento di una tabella all'interno di un'altra.

Casella 11	Casella 12				
Casella 21	<table border="1"> <tbody> <tr> <td>Casella 221</td> <td>Casella 222</td> </tr> <tr> <td>Casella 221</td> <td>Casella 222</td> </tr> </tbody> </table>	Casella 221	Casella 222	Casella 221	Casella 222
Casella 221	Casella 222				
Casella 221	Casella 222				

## Browser

Opera e Mosaic non richiedono l'uso di **&nbsp;**; all'interno di celle vuote. Mosaic sembra avere problemi con l'attributo **CELLSPACING=0**. Tutti i browser visualizzano il testo nelle tabelle con lo stesso tipo di caratteri del testo normale. Le vecchie versioni di Netscape non sono in grado di visualizzare correttamente e tabelle annidate.

## Appendice G

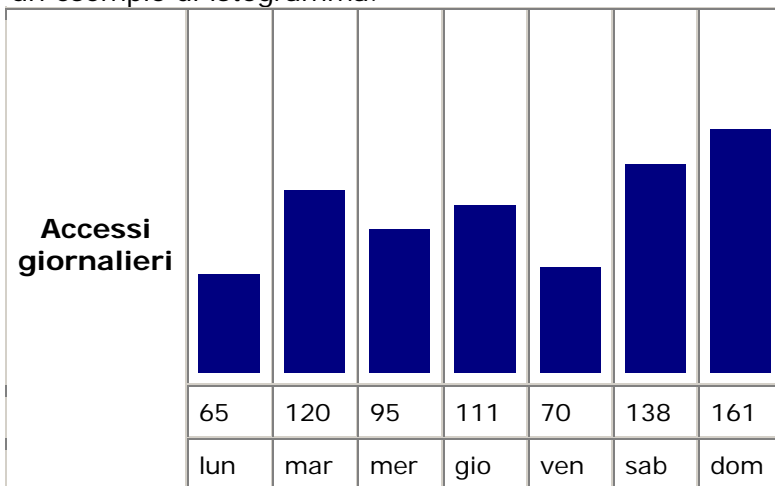
### Istogrammi

Qualche volta è utile presentare statistiche e risultati analitici tramite istogrammi a barre orizzontali o verticali la cui lunghezza sia proporzionale alla quantità indicata. Questi diagrammi possono essere generati in codice HTML da programmi di statistica.

Il tag **<IMG>** può visualizzare un semplice rettangolo colorato; gli attributi **HEIGHT** e **WIDTH** possono quindi allargare l'immagine in entrambe le direzioni.

Mediante il tag **<TABLE>** possiamo quindi costruire istogrammi di qualunque tipo.

Ecco un esempio di istogramma.



e questo è il codice HTML:

```
<TABLE BORDER=1 CELLPADDING=5 CELLSPACING=0>
<TR VALIGN="bottom">
<TH ROWSPAN=3 VALIGN="middle">Accessi<BR>giornalieri</TH>
<TD HEIGHT=200><IMG SRC="pixel.jpg" WIDTH=30
HEIGHT=65></TD>
<TD><IMG SRC="pixel.jpg" WIDTH=30 HEIGHT=120></TD>
<TD><IMG SRC="pixel.jpg" WIDTH=30 HEIGHT=95></TD>
<TD><IMG SRC="pixel.jpg" WIDTH=30 HEIGHT=111></TD>
<TD><IMG SRC="pixel.jpg" WIDTH=30 HEIGHT=70></TD>
<TD><IMG SRC="pixel.jpg" WIDTH=30 HEIGHT=138></TD>
<TD><IMG SRC="pixel.jpg" WIDTH=30 HEIGHT=161></TD>
</TR>
<TR>
<TD>65</TD><TD>120</TD><TD>95</TD><TD>111</TD>
<TD>70</TD><TD>138</TD><TD>161</TD>
</TR>
```

```
<TR>  
<TD>lun</TD><TD>mar</TD><TD>mer</TD><TD>gio</TD>  
<TD>ven</TD><TD>sab</TD><TD>dom</TD>  
</TR>  
</TABLE>
```

Notiamo che l'attributo **VALIGN** è impostato a **"bottom"** direttamente nel tag **<TR>** e vale per tutta la riga che conterrà le immagini. Quindi per la prima cella l'attributo viene corretto a **"middle"** (che sarebbe il predefinito) perché come elemento della riga, sarebbe stato a **"bottom"** perché era stato impostato così nel **<TR>**.

Inoltre l'altezza delle celle contenenti le barre viene forzata a 200, che è il massimo valore che ci potremmo aspettare per quelle immagini. Avremmo anche potuto non farlo, ma sarebbe stato brutto graficamente avere una delle barre che riempie tutta la cella a disposizione. Ovviamente basta applicarlo alla prima cella e l'altezza si estende anche alle altre caselle della riga.

## Appendice H

### Informazioni generali sulle liste

L'HTML supporta diversi tipi di liste:

- ? liste non ordinate **<UL>**
- ? liste ordinate (o numerate) **<OL>**
- ? elenchi di definizioni **<DL>**

All'interno di ciascuna lista, gli elementi sono identificati da un tag distintivo:

- ? **<LI>** per le liste ordinate e non ordinate
- ? **<DT>** e **<DD>** per gli elenchi di definizioni

Le liste possono essere annidate un numero arbitrario di volte e ciascun livello di annidamento è indicato con una ulteriore indentatura. Ovviamente possono essere annidati tipi diversi di liste, come nell'esempio che proponiamo:

- ? Primo elemento di livello 1, lista non numerata
- ? Secondo elemento di livello 1, lista non numerata
  1. Primo elemento di livello 2, lista numerata
  2. Secondo elemento di livello 2, lista numerata
    - Primo elemento di una lista di definizioni, livello 2  
qui una descrizione per questo elemento
    - Secondo elemento di una lista di definizioni, livello 2  
qui una descrizione per questo secondo elemento
      - ? Primo elemento in una lista non ordinata  
annidata, livello 3
      - ? un altro, livello 3
      - ? un altro ancora, livello 3
    - Terzo elemento di una lista di definizioni, livello 2  
e qui la sua descrizione
  3. Terzo elemento di livello 2, lista numerata
- ? Terzo elemento di livello 1, lista non numerata

Per presentare le liste possono anche essere usati i tag **<DIR>** e **<MENU>**, ma sono obsoleti e raramente utilizzati.

## Appendice I

### I frame

I frame sono comparsi a partire dallo standard HTML 4.0 e sono una "invenzione" piuttosto recente dell'HTML.

Il principio base dell'utilizzo dei frame è la divisione della finestra in due distinte sotto-finestre, ognuna delle quali può contenere una pagina web indipendente. Gli effetti di una azione in una certa finestra possono però ripercuotersi sul contenuto dell'altra finestra. L'organizzazione delle pagine avviene nella forma di frameset, che specifica il numero di sottofinestre in cui deve venir divisa la pagina corrente. Un frame può contenere una pagina web o, a sua volta, un'altra divisione in frame.

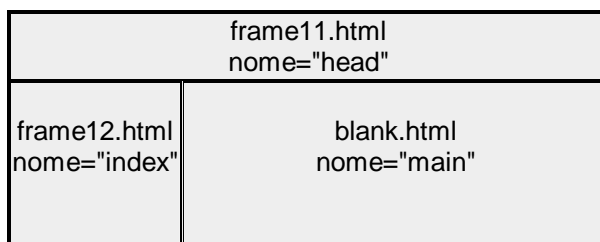
All'interno di un documento HTML, il tag **<FRAMESET>** è usato al posto del tag **<BODY>** e può contenere tag del tipo **<FRAME>**, **<FRAMESET>** e **<NOFRAME>**.

I frame normalmente hanno un nome specificato dall'attributo **NAME** all'interno del tag **<FRAME>**. I frame sono solitamente utilizzati insieme all'attributo **TARGET** del tag **<A>**; quando il visitatore seleziona il link collegato al tag **<A>** il valore del **TARGET** determina il frame nel quale si dovrà aprire l'URL selezionata.

### Esempi

Un uso comune dei frame è quello di utilizzare un frame sottile sulla sinistra dello schermo, che faccia da menu per la navigazione all'interno del sito ed il resto dello schermo per visualizzare i contenuti del sito. Ecco una pagina divisa in tre frame:

```
<frameset rows=20%,80%>
  <frame name=head src=framex11.html>
  <frameset cols=20%,80%>
    <frame name=index src=framex12.html>
    <frame name=main src=blank.html>
  </frameset>
</frameset>
```



Notiamo che ci sono 2 frameset. Il primo divide la finestra in due parti orizzontalmente, il secondo frameset divide il secondo frame in due parti verticalmente.

Volendo far aprire una pagina nel frame grande, chiamato "main" si utilizza un link di questo tipo:

```
<A HREF="marquee.html" TARGET="main">marquee</a>
```

Con Internet Explorer si può omettere l'attributo **SRC** se il frame dovrà visualizzare una pagina vuota. Netscape invece, in queste condizioni aprirebbe una nuova finestra di browser. Questo è probabilmente un errore nell'implementazione del browser.

Quando le pagine sono visualizzate, ciascuna ha le proprie barre di scorrimento, che si comportano in maniera indipendente l'una dall'altra.

Salvo diverse impostazioni nella costruzione del frameset mediante l'attributo **NORESIZE**, l'utente può ridimensionare a proprio piacimento la dimensione relativa di ciascun frame, agendo col mouse sulla riga che separa i frame.

È interessante scoprire come si comporta il proprio browser quando si prova a stampare una pagina o tutta la finestra oppure quando si preme il pulsante per tornare indietro all'ultima pagina visitata.

L'aspetto di un frame è determinato completamente dal codice HTML contenuto nella pagina visualizzata, non c'è alcun modo di controllare la formattazione di tutti i frame che costituiscono un frameset.

L'attributo **NORESIZE** fa in modo che le dimensioni di ciascun frame siano stabilite e che il visitatore non possa modificarle.

L'attributo **SCROLLING** controlla le barre di scorrimento di ciascun frame, permettendone la visualizzazione o la scomparsa. I valori possibili sono **"yes"**, **"no"** o **"auto"**. Forzando a **"no"** lo scrolling di una pagina molto lunga o larga si fa sì che una parte della pagina risulti inaccessibile al visitatore.

Infine, gli attributi **FRAMEBORDER** e **FRAMESPACING** del tag **<FRAMESET>** permettono di gestire i margini e i bordi di ciascun frame, impostandone la presenza e la dimensione.

## Browser

I browser di vecchio tipo, ormai pressoché scomparsi, non riconoscono i frame. Opera e Ariadna non sono in grado di rimuovere completamente i bordi dei frame.



## Appendice L

### I caratteri speciali

Sappiamo già che per inserire i caratteri `<`, `&` e `>` in un documento HTML è necessario scrivere `&lt;`, `&amp;` e `&gt;`; in quanto queste diciture hanno un significato speciale in seno agli interpreti HTML. Esistono numerosi caratteri speciali che possono essere introdotti utilizzando questa notazione. Si tratta di tutti i caratteri non compresi nel normale set di caratteri ASCII, il cui inserimento diretto può dare risultati indefinibili in quanto esistono differenti interpretazioni per i caratteri appartenenti al codice ASCII a più di 7 bit.

L'HTML ha due modi per inserire caratteri speciali.

1. Utilizzando la notazione `&#` seguito dal numero decimale associato col carattere, secondo la tabella qui sotto.
2. Utilizzando il carattere `&` seguito dal codice descrittivo per quella quantità, definito nella stessa tabella qui sotto.

Ad esempio la parola "Francese" nella lingua francese si scrive "*Français*", e possiamo notare il pedice sotto la lettera C, che la trasforma nel simbolo di cedilla. Il codice HTML per ottenere questa parola è quindi:

```
fran&ccedil;ais
oppure
fran&#231;ais
```

Descrizione	Entità	Decimale	Simbolo
spazio non interrompente	nbspc	160	
esclamativo capovolto	ixcl	161	¡
centesimo	cent	162	¢
sterlina	pound	163	£
valuta	curren	164	¤
yen	yen	165	¥
barra interrotta	brvbar	166	¦
sezione	sect	167	§
dieresi	uml	168	¨
copyright	copy	169	©
ordinale femminile	ordf	170	ª
freccie sinistre	laquo	171	«

not logico	not	172	¬
		173	
marchio registrato	reg	174	®
distanziatore	macr	175	˘
gradi	deg	176	°
più o meno	plusmn	177	±
2 ad apice	sup2	178	<sup>2</sup>
3 ad apice	sup3	179	<sup>3</sup>
accento acuto	acute	180	´
micro	micro	181	μ
paragrafo	para	182	¶
punto centrale	middot	183	·
cedilla	cedil	184	¸
1 ad apice	sup1	185	<sup>1</sup>
ordinale maschile	ordm	186	º
frecce destre	raquo	187	»
un quarto	frac14	188	¼
un mezzo	frac12	189	½
tre quarti	frac34	190	¾
interrogativo capovolto	iquest	191	¿
A grave	Agrave	192	À
A acuta	Aacute	193	Á
A circonflessa	Acirc	194	Â
A tilde	Atilde	195	Ã
A dieresi	Auml	196	Ä
A cerchiata	Aring	197	Å
AE latina	AElig	198	Æ
C cedilla	Ccedil	199	Ç
E grave	Egrave	200	È
E acuta	Eacute	201	É
E circonflessa	Ecirc	202	Ê
E dieresi	Euml	203	Ë
I grave	Igrave	204	Ì
I acuta	Iacute	205	Í
I circonflessa	Icirc	206	Î
I dieresi	Iuml	207	Ï
ETH	ETH	208	Ð
N tilde	Ntilde	209	Ñ

O grave	Ograve	210	Ò
O acuta	Oacute	211	Ó
O circonflessa	Ocirc	212	Ô
O tilde	Otilde	213	Õ
O dieresi	Ouml	214	Ö
prodotto	times	215	×
O barrata	Oslash	216	Ø
U grave	Ugrave	217	Û
U acuta	Uacute	218	Ú
U circonflessa	Ucirc	219	Û
U dieresi	Uuml	220	Ü
Y acuta	Yacute	221	Ý
THORN	THORN	222	þ
s sharp	szlig	223	ß
a grave	agrave	224	à
a acuta	aacute	225	á
a circonflessa	acirc	226	â
a tilde	atilde	227	ã
a dieresi	auml	228	ä
a cerchiata	aring	229	å
ae latina	aelig	230	æ
c cedilla	ccedil	231	ç
e grave	egrave	232	è
e acuta	eacute	233	é
e circonflessa	ecirc	234	ê
e dieresi	euml	235	ë
i grave	igrave	236	ì
i acuta	iacute	237	í
i circonflessa	icirc	238	î
i dieresi	iuml	239	ï
eth	eth	240	ð
n tilde	ntilde	241	ñ
o grave	ograve	242	ò
o acuta	oacute	243	ó
o circonflessa	ocirc	244	ô
o tilde	otilde	245	õ
o dieresi	ouml	246	ö
divisione	divide	247	÷

o barrata	oslash	248	ø
u grave	ugrave	249	ù
u acuta	uacute	250	ú
u circonflessa	ucirc	251	û
u dieresi	uuml	252	ü
y acuta	yacute	253	ý
thorn	thorn	254	þ
y dieresi	yuml	255	ÿ

I caratteri *eth* e *thorn* sono lettere islandesi. Altri caratteri provengono da alfabeti di altri paesi, e molti simboli non sono stati inseriti. Forse un giorno i browser WWW saranno compatibili con i caratteri Unicode a 16bit.

Sia Netscape Navigator che Internet Explorer visualizzano correttamente questi caratteri.

## Appendice M

### I colori e i codici esadecimali

L'HTML conosce due modi per esprimere i colori: o col loro nome oppure con un numero che ne identifica le tonalità.

Moltissimi colori hanno un nome, in inglese, sebbene non sia possibile dare un nome a tutti i 16.777.216 (256x256x256) colori che possono essere rappresentati. I colori primari sono 3, rosso, verde e blu, e mediante una combinazione di essi è possibile ricavare ogni colore utilizzabile. I principali colori sono [aqua](#), [black](#), [blue](#), [fuchsia](#), [gray](#), [green](#), [lime](#), [maroon](#), [navy](#), [olive](#), [purple](#), [red](#), [silver](#), [teal](#), [white](#), [yellow](#), ma poi ne esistono altri che sono sfumature dei colori primari e ai quali è stato dato un nome fantasioso per poterli ricordare meglio; è il caso di [RosyBrown](#), [Gold](#), [SandyBrown](#), [ForestGreen](#), [DarkTurquoise](#), e tanti altri che possiamo trovare nella pagina dei **COLORI PIÙ COMUNI**, nel prossimo capitolo.

Se vogliamo utilizzare un colore nella sua più piccola sfumatura occorre identificarlo tramite il suo numero. Chiarito che per ogni colore primario possiamo ricavare 256 tonalità ecco che esprimendo ogni colore col proprio numero possiamo ottenere  $256 \times 256 \times 256 = 16.777.216$  colori !!.

Per esprimere un colore in base al proprio numero si usa specificare di quante parti dei colori primari esso è composto, e si va da un minimo di 0 ad un massimo di 255: 0 vuol dire assenza di colore (nero), 255 vuol dire colore pieno; ecco che un rosso scuro può essere il numero 64 mentre un rosso acceso il 212 se non addirittura il 255. Allora il numero 127,0,255 (rosso, verde, blu) rappresenta un colore che contiene 127 parti di rosso (circa la metà di 255), nessuna parte di verde e un blu pieno, e quindi è il colore viola, mentre il numero 255,255,0 che contiene una parte intera di rosso e verde e nessuna parte di blu è il colore giallo; il colore 150,130,50 formato da 150 parti di rosso, 130 di verde e 50 di blu è una sfumatura di marrone.

L'HTML non usa esprimere i numeri dei colori nel sistema decimale, ma utilizza il codice esadecimale, ossia in base 16, che è rappresentato dai simboli **1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**. Con due simboli nel sistema decimale è possibile rappresentare fino a  $10 \times 10 = 100$  numeri, da 0 a 99, mentre con 2 simboli esadecimali è possibile ottenere  $16 \times 16 = 256$  numeri, ossia la totalità delle sfumature di un colore primario. Allora per esprimere un colore serviranno e basteranno 6 cifre esadecimali, 2 per il rosso, 2 per il verde e 2 per il blu.

### Conversione decimale-esadecimale.

Detto ciò non ci resta che vedere come esprimere un numero in base 10 nella numerazione esadecimale. Per quanto riguarda gli estremi è facile: lo 0 decimale è lo 00 esadecimale, mentre al 255 corrisponde il numero FF, ma i numeri intermedi? Per ricavare i numeri intermedi la procedura è molto semplice: si divide il numero decimale per 16 e con la parte intera si fa la prima cifra mentre col resto si ottiene la seconda. Questo procedimento può essere capito meglio con qualche esempio e un po' di pratica.

Trasformiamo il numero 128 in esadecimale: 128 diviso 16 fa 8 con resto di zero, allora il numero esadecimale corrispondente è 80; il numero 73 diventa  $73 \div 16 = 4$  col resto di 9, per cui è 49; il numero 93 è  $93 \div 16 = 5$  col resto di 13, ossia 5D; il numero 154 diventa  $154 \div 16 = 9$  col resto di 10 ossia 9A; ed infine 199 diventa  $199 \div 16 = 12$  col resto di 7, ossia C7.

Allora i colori visti prima sono:

$(127 \ 0 \ 255) = 7F00FF,$

$(255 \ 255 \ 0) = FFFF00,$  e infine

$(150 \ 130 \ 50) = 968232.$

La procedura inversa, da esadecimale a decimale è ancora più semplice: dato un numero esadecimale di 2 cifre, per portarlo nel sistema decimale basta moltiplicare la prima cifra per 16 ed aggiungere il risultato alla seconda cifra. In questo modo abbiamo:

$FF = 15*16+15 = 240+15 = 255;$

$80 = 8*16+0 = 128;$

$99 = 9*16+9 = 144+9 = 153;$

$7C = 7*16+12 = 112+12 = 124;$


$BE = 11*16+14 = 176+14 = 190.$

Nella prossima appendice vedremo i nomi dei principali colori ed i loro numeri.

## Appendice N

### I colori più comuni

Questi che seguono sono i 16 colori standard di windows.

	<b>black</b> ="#000000"		<b>green</b> ="#008000"
	<b>silver</b> ="#C0C0C0"		<b>lime</b> ="#00FF00"
	<b>gray</b> ="#808080"		<b>olive</b> ="#808000"
	<b>white</b> ="#FFFFFF"		<b>yellow</b> ="#FFFF00"
	<b>maroon</b> ="#800000"		<b>navy</b> ="#000080"
	<b>red</b> ="#FF0000"		<b>blue</b> ="#0000FF"
	<b>purple</b> ="#800080"		<b>teal</b> ="#008080"
	<b>fuchsia</b> ="#FF00FF"		<b>aqua</b> ="#00FFFF"

Questi sotto invece sono i nomi dei colori più comuni.

### Colori a prevalenza di ROSSO

<b>DarkRed</b> (#8B0000)	<b>SaddleBrown</b> (#8B4513)	<b>Brown</b> (#A52A2A)	<b>Sienna</b> (#A0522D)	<b>Firebrick</b> (#B22222)
<b>Crimson</b> (#DC143C)	<b>Tomato</b> (#FF6347)	<b>OrangeRed</b> (#FF4500)	<b>DarkOrange</b> (#FF8C00)	<b>Orange</b> (#FFA500)
<b>Coral</b> (#FF7F50)	<b>LightCoral</b> (#F08080)	<b>LightSalmon</b> (#FFA07A)	<b>Salmon</b> (#FA8072)	<b>DarkSalmon</b> (#E9967A)
<b>IndianRed</b> (#CD5C5C)	<b>Gold</b> (#FFD700)	<b>Pink</b> (#FFC0CB)	<b>LightPink</b> (#FFB6C1)	<b>MistyRose</b> (#FFE4E1)
<b>HotPink</b> (#FF69B4)	<b>DeepPink</b> (#FF1493)	<b>PaleVioletRed</b> (#DB7093)	<b>MediumVioletRed</b> (#C71585)	<b>RosyBrown</b> (#BC8F8F)
<b>Wheat</b> (#F5DEB3)	<b>Burlywood</b> (#DEB887)	<b>Tan</b> (#D2B48C)	<b>Goldenrod</b> (#DAA520)	<b>Chocolate</b> (#D2691E)
<b>DarkKhaki</b> (#BDB76B)	<b>Khaki</b> (#F0E68C)	<b>PaleGoldenrod</b> (#EEE8AA)	<b>DarkGoldenrod</b> (#B8860B)	<b>Moccasin</b> (#FFE4B5)
<b>NavajoWhite</b> (#FFDEAD)	<b>Bisque</b> (#FFE4C4)	<b>PapayaWhip</b> (#FFefd5)	<b>Oldlace</b> (#fdf5e6)	<b>Cornsilk</b> (#fff8dc)
<b>AntiqueWhite</b> (#FAEBD7)	<b>FloralWhite</b> (#FFFAF0)	<b>Seashell</b> (#FFF5EE)	<b>LavenderBlush</b> (#FFF0F5)	<b>SandyBrown</b> (#F4A460)

### Colori a prevalenza di VERDE

<b>SpringGreen</b> (#00FF7F)	<b>LawnGreen</b> (#7CFC00)	<b>Medium SpringGreen</b> (#00FA9A)	<b>GreenYellow</b> (#ADFF2F)	<b>Chartreuse</b> (#7FFF00)
<b>PaleGreen</b> (#98FB98)	<b>LightGreen</b> (#90EE90)	<b>YellowGreen</b> (#9ACD32)	<b>LimeGreen</b> (#32CD32)	<b>ForestGreen</b> (#228B22)
<b>DarkGreen</b> (#006400)	<b>OliveDrab</b> (#6B8E23)	<b>Dark OliveGreen</b> (#556B2F)	<b>SeaGreen</b> (#2E8B57)	<b>Medium SeaGreen</b> (#3CB371)
<b>Light SeaGreen</b> (#20B2AA)	<b>Dark SeaGreen</b> (#8FBC8B)	<b>Aquamarine</b> (#7FFFD4)	<b>Medium Aquamarine</b> (#66CDAA)	<b>Medium Turquoise</b> (#48D1CC)
<b>Turquoise</b> (#40E0D0)	<b>MintCream</b> (#F5FFFA)			

### Colori a prevalenza di BLU

<b>DeepSkyBlue</b> (#00BFFF)	<b>DodgerBlue</b> (#1E90FF)	<b>Cornflower Blue</b> (#6495ED)	<b>Medium SlateBlue</b> (#7B68EE)	<b>RoyalBlue</b> (#4169E1)
<b>SlateBlue</b> (#6A5ACD)	<b>Dark Turquoise</b> (#00CED1)	<b>MediumBlue</b> (#0000CD)	<b>Dark SlateBlue</b> (#483D8B)	<b>SteelBlue</b> (#4682B4)
<b>DarkBlue</b> (#00008B)	<b>Indigo</b> (#4B0082)	<b>MidnightBlue</b> (#191970)	<b>SkyBlue</b> (#87CEEB)	<b>LightSkyBlue</b> (#87CEFA)
<b>LightBlue</b> (#ADD8E6)	<b>PowderBlue</b> (#B0E0E6)	<b>LightSteelBlue</b> (#B0C4DE)	<b>CadetBlue</b> (#5F9EA0)	<b>MediumPurple</b> (#9370DB)
<b>BlueViolet</b> (#8A2BE2)	<b>DarkViolet</b> (#9400D3)	<b>DarkOrchid</b> (#9932CC)	<b>MediumOrchid</b> (#BA55D3)	<b>AliceBlue</b> (#F0F8FF)

### Colori con due o tre componenti uguali e prevalenti

<b>DarkGray</b> (#A9A9A9)	<b>DimGray</b> (#696969)	<b>SlateGray</b> (#708090)	<b>Light SlateGray</b> (#778899)	<b>LightGrey</b> (#D3D3D3)
<b>DarkCyan</b> (#008B8B)	<b>Gainsboro</b> (#DCDCDC)	<b>LightCyan</b> (#E0FFFF)	<b>Light Goldenrod Yellow</b> (#FAFAD2)	<b>Thistle</b> (#D8BFD8)
<b>Lavender</b> (#E6E6FA)	<b>Azure</b> (#F0FFFF)	<b>Dark SlateGray</b> (#2F4F4F)	<b>GhostWhite</b> (#F8F8FF)	<b>Beige</b> (#F5F5DC)
<b>Snow</b> (#FFFAFA)	<b>LightYellow</b> (#FFFFE0)	<b>Ivory</b> (#FFFFF0)	<b>Pale Turquoise</b> (#AFEEEE)	



*".. e se proprio non vi basta ...."*

esistono numerosi programmi che convertono decimali in esadecimali e viceversa e permettono di visualizzare i colori corrispondenti. È possibile trovare una comoda e funzionale applicazione java per ricavare il codice esadecimale, partendo dal colore che si vuole ottenere o dalle percentuali dei tre colori rosso, verde e blu, sul sito di riferimento di questa guida, [www.canoro.org](http://www.canoro.org), o sul cd allegato.

## Glossario

### ATTRIBUTO

Elemento che permette di specificare le proprietà di un tag. Ad esempio **HREF** è un attributo del tag **<A>**.

### BROWSER

Programma che ha la funzione di interpretare un documento HTML e rappresentarlo sullo schermo, inoltre permette una lettura e una "navigazione" ipertestuale.

### CGI

Acronimo di Common Gateway Interface: si tratta di uno standard di interfaccia tra i programmi dal lato server ed i browser dal lato client. I programmi sono scritti in vari linguaggi (Perl, C) che girano in aree apposite del server (cgi-bin) e permettono di creare pagine dinamiche basate su database, motori di ricerca, analizzatori di form, accesso con password ecc. Se ne trovano anche di gratuiti.

### COOKIE

File che viene memorizzato sul computer del visitatore e che serve a monitorare la navigazione del visitatore e a memorizzare elementi utili al browser e al sito web depositario delle pagine visitate.

### CSS

Acronimo di Cascading Style Sheet, indica una serie di tecniche utili per dare o cambiare uno stile di formattazione applicabile alle pagine di un sito web; è utile per disporre elementi sullo schermo nella posizione desiderata, senza dover ricorrere alle tabelle di HTML.

### DHTML

HTML Dinamico; è uno strumento molto potente, e permette di creare effetti che anche con HTML e javascript non si possono ottenere.

### ESADECIMALE

Tecnica di conteggio in base 16; per rappresentare i numeri in questo sistema vengono utilizzate le 10 cifre numeriche dallo 0 al 9 più le prime 6 lettere dell'alfabeto, dalla a alla f.

### FAQ

Acronimo di Frequently Asked Questions, ovvero "domande poste di frequente". Si tratta di raccolte di domande poste frequentemente con le relative risposte. Servono per dare informazioni aggiuntive agli utenti e per evitare loro di chiedere sempre le stesse cose a chi gestisce un sito o un newsgroup.

**FORMATTAZIONE**

È quell'insieme di operazioni atte a modificare le impostazioni di un elemento, in genere di testo, e a dargli la forma e l'aspetto voluti.

**FRAME**

Ciascuna parte in cui può venire divisa una pagina HTML, creando pagine nuove e indipendenti.

**GIF**

Graphic Interchange Format; è un formato di file grafico fatto soprattutto per disegni e immagini a tinta unita. Il formato GIF consente anche di creare animazioni e immagini trasparenti.

**HTML**

La sigla HTML è l'acronimo di HiperText Markup Language e tradotto letteralmente indica un "linguaggio a marcatori per l'ipertesto". In pratica è un linguaggio di scripting che contiene anche la formattazione del testo.

**IPERTESTO**

Documento con possibilità di lettura non lineare, con collegamenti per saltare arbitrariamente da un argomento ad un altro.

**JAVA**

Linguaggio di programmazione orientato agli oggetti realizzato dalla Sun Microsystems. È simile al linguaggio C++ ed è stato progettato specificatamente per diventare il linguaggio di sviluppo di applicazioni programmate per Internet. Ha alcune caratteristiche che ne denotano la versatilità: infatti le applicazioni sviluppate in Java possono girare su diverse piattaforme. Nelle pagine web si possono inserire "applet" (ovvero mini-programmi) Java, e si possono integrare applet Java e JavaScript per la costruzione di pagine web interattive.

**JAVASCRIPT**

Linguaggio di scripting simile al C, sviluppato dalla Netscape Communications. JavaScript è stato implementato come estensione del linguaggio HTML e consente agli sviluppatori di creare pagine Web contenenti elementi che rispondono ad eventi-utente ed eseguire calcoli e operazioni client-side. Microsoft ha sviluppato un linguaggio simile a JavaScript chiamato JScript, il quale però ha delle caratteristiche compatibili solo con il browser Internet Explorer.

**JPEG**

Joint Photographic Expert Group; un formato di file grafico usato per visualizzare immagini a colori ad alta risoluzione.

**LAYOUT**

È la visualizzazione di un documento; ossia quello che si riesce ad ottenere sullo schermo dopo avergli dato forma e impostazione (e contenuto...).

**LINK**

Collegamento ipertestuale, in grado di far saltare la visualizzazione da un punto in una pagina ad un altro nella stessa pagina o in una pagina diversa o in un sito diverso.

**MOTORE DI RICERCA**

Programma residente su un sito in grado di fornire un elenco di siti web (contenuti nel proprio database) corrispondenti a determinate parole chiave.

**PLUG-IN**

Programma che estende le capacità di un browser web in modo che possa elaborare file che normalmente non supporta; i file multimediali hanno bisogno di plug-in.

**ROLLOVER**

Effetto attivato al passaggio del mouse su un oggetto nella pagina, specialmente un'immagine. Si può avere l'attivazione di un'animazione o il cambio di colore dell'immagine. Lo si fa con JavaScript o applet Java.

**SCRIPTING**

Sono certi linguaggi di programmazione finalizzati alla realizzazione di funzioni per documenti HTML.

**SERVER**

Programma che gira su un computer a cui ci colleghiamo e che ci permette di effettuare alcune operazioni su quel computer. Senza il programma server è impossibile collegarsi ad un dato computer. Il termine indica impropriamente anche il computer a cui ci si collega e che ospita un sito web.

**URL**

Universal Resource Locator, ossia letteralmente "localizzatore di risorse universali"; è un indirizzo internet, come "http://canoro.altervista.org", oppure, nella forma numerica, "http://210.75.68.191".

**VIDEO SHOCKWAVE**

Un file multimediale creato con una delle applicazioni di Macromedia, tra cui Director e Flash.

**WYSIWYG**

Tecnica di videoscrittura in grado di permettere di creare una pagina disponendo gli elementi esattamente come essi appariranno quando si visualizza la pagina. Il Word è un programma di videoscrittura WYSIWYG, il Notepad non lo è.

**XML**

eXtensible Markup Language; è una tecnologia web che consente agli sviluppatori di pagine HTML di creare tag personalizzati, che aggiungono capacità e funzionalità nuove all'HTML.

**TAG**

Ciascuno dei comandi del linguaggio HTML. È l'elemento base del linguaggio HTML, racchiuso fra i caratteri `<` e `>`. Ogni tag ha determinati attributi, alcuni dei quali obbligatori. Alcuni tag richiedono anche un tag di chiusura (nell'XHTML tutti i tag devono essere chiusi).

## Indice analitico

Animazioni . . . . .	32	<DIV> . . . . .	18, 108
Browser . . . . .	7	<DL> . . . . .	60, 109
Caratteri . . . . .	21, 232	<DT> . . . . .	60, 112
Celle . . . . .	51	<EM> . . . . .	21, 113
Collegamenti ipertestuali . . . . .	34	<EMBED> . . . . .	28, 114
Colori . . . . .	236	<FONT> . . . . .	22, 116
CSS . . . . .	7	<FORM> . . . . .	70, 119
Elenchi . . . . .	55, 229	<FRAME> . . . . .	63, 121
Form . . . . .	69	<FRAMESET> . . . . .	62, 123
Formattazione . . . . .	21	<Hn> . . . . .	18, 125-130
Frame . . . . .	62	<HEAD> . . . . .	11, 131
Iframe . . . . .	67	<HR> . . . . .	19, 132
Immagini . . . . .	24, 216	<HTML> . . . . .	10, 135
Intestazioni . . . . .	212	<I> . . . . .	21, 136
Istogrammi . . . . .	227	<IFRAME> . . . . .	67, 137
Link . . . . .	34	<IMG> . . . . .	24, 140
Mappe . . . . .	42, 218	<INPUT> . . . . .	71, 147
Moduli . . . . .	69	<KBD> . . . . .	21, 153
Motori di ricerca . . . . .	12	<LI> . . . . .	55, 154
Paragrafi . . . . .	18	<LINK> . . . . .	156
Sfondo . . . . .	14	<LISTING> . . . . .	158
Stili . . . . .	213	<MAP> . . . . .	43, 159
Tabelle . . . . .	46, 221	<MARQUEE> . . . . .	160
TAG . . . . .	10, 210	<MENU> . . . . .	163
<A> . . . . .	34, 77	<META> . . . . .	12, 164
<ADDRESS> . . . . .	21, 80	<NOFRAMES> . . . . .	63, 167
<APPLET> . . . . .	81	<OL> . . . . .	57, 168
<AREA> . . . . .	43, 83	<OPTION> . . . . .	74, 179
<B> . . . . .	21, 86	<P> . . . . .	18, 170
<BASEFONT> . . . . .	87	<PARAM> . . . . .	172
<BG SOUND> . . . . .	28, 89	<PLAINTEXT> . . . . .	173
<BIG> . . . . .	21, 90	<PRE> . . . . .	21, 174
<BLINK> . . . . .	91	<SAMP> . . . . .	21, 176
<BLOCKQUOTE> . . . . .	19, 92	<SCRIPT> . . . . .	177
<BODY> . . . . .	14, 94	<SELECT> . . . . .	74, 179
  . . . . .	19, 97	<SMALL> . . . . .	21, 181
<CAPTION> . . . . .	49, 99	<SPACER> . . . . .	182
<CENTER> . . . . .	100	<SPAN> . . . . .	183
<CITE> . . . . .	21, 101	<STRIKE> . . . . .	184
<CODE> . . . . .	21, 102	<STRONG> . . . . .	21, 185
<DD> . . . . .	60, 104	<STYLE> . . . . .	18, 186
<DFN> . . . . .	21, 105	<SUB> . . . . .	189
<DIR> . . . . .	19, 106	<SUP> . . . . .	190

<TABLE> . . . . .	46, 191	<U> . . . . .	21, 205
<TD> . . . . .	52, 193	<UL> . . . . .	55, 206
<TEXTAREA> . . . . .	74, 197	<VAR> . . . . .	208
<TH> . . . . .	49, 198	<XMP> . . . . .	209
<TITLE> . . . . .	11, 201	Testo . . . . .	18
<TR> . . . . .	46, 202	Titoli . . . . .	212
<TT> . . . . .	21, 204		

## Bibliografia

Roberto Baldoni: *"Lezioni di HTML"*

Michel Dreyfus: *"Creer Page Web"*, 1999

Dan Livingston, Micah Brown: *"CSS & DHTML 4 for web professionals"*, 1999

[www.html.it](http://www.html.it) Guida online di HTML

[www.w3.org](http://www.w3.org) W3C - HTML 4.01 Specification





Destinata a chi vuole imparare velocemente a costruire un proprio sito web completo, questa guida vuole essere un riferimento anche per chi desidera una conoscenza approfondita del comportamento di ciascun tag. Non a caso il manuale contiene una ricca e dettagliata appendice che descrive tutti i campi di applicazione di ciascun tag riconosciuto dallo standard HTML 4.01.

Il manuale porta il lettore ad una conoscenza graduale del linguaggio, partendo dalla definizione più semplice di pagina fino alla introduzione di elementi sempre più complessi come tabelle e frame, passando per l'elemento che più di tutti ha reso bello e affascinante il mondo del web e che ne ha permesso la rapida ed enorme diffusione, il [collegamento ipertestuale](#).

Michele Ferrara conosce l'HTML fin dalle sue prime versioni nel 1996 e ha pubblicato diversi siti web. È laureato in ingegneria dal 2004 e lavora presso una importante società di telecomunicazioni. Ha realizzato la sua prima "Guida di HTML" nel 1999, sulla versione 3.2 dello Standard HTML, proseguendo con gli aggiornamenti ogni volta che lo standard veniva aggiornato con l'introduzione di nuovi tag e nuove caratteristiche.

Dello stesso autore

- ? Appunti di misure elettriche (2001)
- ? La nostra Roma (2003)
- ? I pannelli solari (2006)
- ? Manuale di CSS (2006)

[www.canoro.altervista.org](http://www.canoro.altervista.org)