

Introduzione all'HTML

tratto da www.html.it

a cura di **Massimiliano Valente**
adattamento e impaginazione a cura del **prof. Massimiliano Forgione**
per l'Istituto Bodoni Paravia

INTRODUZIONE

Benvenuto in questa sezione della guida che intende seguirti ed aiutarti nella realizzazione della tua home page. Questa premessa, apparentemente inutile, serve invece a sottolineare pochi ma fondamentali concetti che reggono l'HTML e che, se compresi tempestivamente, consentono di far proprio un sistema di contrassegno tutt'altro che complesso, ma pur sempre retto da regole e principi inderogabili.

Creare un sito Internet non è difficile, ne' può essere messo a confronto con veri e propri sistemi di programmazione. Su quest'ultimo punto, in particolare, è bene fin d'ora sottolineare come HTML non sia un linguaggio di programmazione ma un semplice sistema di contrassegno, i cui tag vengono riconosciuti ed interpretati dai browser Web (Netscape, MsIe, Opera ecc). Questa peculiarità rende HTML un sistema facile da comprendere perchè non presuppone alcuna conoscenza tecnica preesistente. Per questo motivo chiunque può avvicinarsi al Web Publishing senza prerequisiti di specializzazione e per questo in Internet esistono milioni di creatori di pagine Web.

Il mercato è gravido di decine di software specializzati nella creazione di pagine HTML: i cosiddetti editor HTML. La complessità di questi programmi ha raggiunto livelli molto alti ma ha seguito strade diverse. In buona sostanza il mercato propone due tipi di editor HTML:

Editor testuali

Si tratta di programmi che propongono modifiche dirette sul codice HTML puro con possibilità di preview del risultato. Per facilitare la scrittura del codice mettono a disposizione comandi preconfezionati attivabili con semplici click. L'utilità maggiore di questi strumenti è la padronanza del codice HTML puro che riescono a comunicare agli utilizzatori. Il difetto maggiore è nell'apprendimento più difficile rispetto agli editor WYSIWYG.

Editor WYSIWYG

L'acronimo sopraindicato sta per: What You See Is What You Get, ed italianizzando significa che ciò che si vede sullo schermo è ciò che si ottiene in un browser Web. In altre parole questi editor non visualizzano il codice HTML ma esclusivamente gli oggetti, le immagini ed il testo. In questo modo lo sviluppatore non si trova a lavorare su codice HTML ma sulla pagina così come verrà visualizzata dal browser: spostando oggetti con il semplice trascinarsi del mouse, sfruttando layout predefiniti ecc.

Il vantaggio di questo tipo di editor è evidente: lo sforzo di comprensione

delle specifiche HTML è ridotto al minimo e il tempo di apprendimento è brevissimo.

Secondo chi scrive, però, questo tipo di editor (un esempio concreto è Front Page) non dà la possibilità di comprendere l'HTML e personalizzarne l'uso. Si corre, infatti, il rischio di creare siti fotocopia con un layout identico agli altri e, in presenza di errori che l'editor non riesce a risolvere, di non riuscire ad andare oltre ciò che il programma prevede.

A prescindere da pareri soggettivi su programmi per l'HTML, la presente guida è concepita per far comprendere i rudimenti dell'HTML puro. Continuando questo corso non ti troverai di fronte un tutorial sui comandi degli editor più diffusi, ma davanti codice HTML puro ed universale. Solo in questo modo potrai comprendere le peculiarità dell'HTML e degli altri linguaggi che ne supportano e ampliano l'utilizzo.

Abbandona per tutto il corso di questo tutorial il tuo editor e concentrati sulle nozioni di codice che la guida ti presenta volta per volta. Comprese le basi dell'HTML potrai scegliere se usare editor WYSIWYG o editor testuali, in entrambi i casi avrai consapevolezza dei limiti e delle potenzialità del sistema, potendo usare meglio tutti gli strumenti che il mercato mette a tua disposizione.

Alcuni consigli per comprendere pienamente questa guida:

i siti Web, prima della pubblicazione, vanno creati in locali, cioè sul tuo hard disk. Solo successivamente vengono spediti attraverso programmi appositi.

Per i nomi dei file e per le estensioni degli stessi usa caratteri sempre in minuscolo.

Ricorda che la modifica del tuo sito Web avviene tramite programmi FTP (File Transfer Protocol) mediante password e UserId. Quindi nessun altro, oltre te, può modificare le pagine.

Per scrivere codice HTML puoi utilizzare qualsiasi editor HTML testuale (DreamWeaver ecc.), ma anche solo Blocco Notes di WinXP.

Prima di continuare è importante sottolineare una caratteristica dei tag HTML. Come più volte accennato, HTML è un sistema di contrassegno che produce i suoi effetti su testo e immagini. Perché tali effetti si producano è necessario che elementi specifici (standardizzati in HTML) vengano assegnati ai vari elementi testuali o d'immagine della pagina. Il termine marcatura deriva proprio da questa caratteristica. Per una migliore comprensione di questo fondamentale concetto di HTML prendiamo in considerazione il testo seguente:

HTML è un sistema di contrassegno indipendente dalla piattaforma.

Come vedi si tratta di un normalissimo testo per il quale non abbiamo specificato alcun tag HTML. In linea con i concetti che reggono il sistema di markup, senza altre indicazioni, il browser formatta il testo come quello che lo precede (infatti si tratta di un font arial di dimensioni e colore identici a questo stesso testo).

Consideriamo, ora, l'ipotesi di voler dare un colore rosso al testo di esempio. Per fare questo marchiamo il testo con lo specifico tag HTML che determina il colore del testo:

```
<font color="red">HTML è un sistema di contrassegno indipendente dalla  
piattaforma.</font>
```

In seguito vedremo il funzionamento del tag FONT. Quello che interessa sottolineare in questa sede è l'uso generale dei marcatori HTML: come si nota agevolmente dall'esempio, il tag `` è posto immediatamente prima e dopo la frase da formattare in rosso. L'unica differenza è che il tag prima della frase è ``, mentre quello che chiude è ``. La barra posta in verticale verso destra indica al browser che il tag `` precedentemente aperto, in questo caso va chiuso. Abbiamo così dato vita ad un piccola marcatura, comunque sintomatica del funzionamento del sistema HTML.

I tag HTML sono case insensitive, cioè assolutamente indipendenti dal maiuscolo e minuscolo. Per intenderci, il seguente codice:

```
<font color="red"></font>
```

ha gli stessi effetti del codice:

```
<FONT COLOR="RED"></font>
```

In HTML esistono TAG che non hanno bisogno di chiusura, perchè la loro funziona non è quella di marcare un elemento, ma di fornire informazioni di diverso tipo. Volta per volta indicheremo quali elementi hanno bisogno di chiusura e quali funzionano correttamente anche se ne sono privi.

<HEAD> Intestare il documento

Per essere visualizzati su Internet, i documenti vengono salvati in formato testo e contengono i tag necessari ad informare il browser (Netscape, MsIE o Explorer) sulla visualizzazione della stessa. In altri termini un documento contenente testo, salvato in formato .htm senza alcun tag HTML viene comunque visualizzato dal browser, ma privo di qualsiasi formattazione: senza ritorni a capo, paragrafi, testo centrato, grassetto, corsivo ecc.

Lo scopo dell'HTML è quello di fornire, attraverso comandi chiamati TAG , una formattazione al documento, oltre all'inserimento di immagini ed altri elementi multimediali (filmati, applet ecc.). Il lavoro che uno sviluppatore Web produce all'interno di un documento HTML è indirizzato a fornire tutte le informazioni necessarie al browser per interpretare correttamente la pagina.

Un documento HTML si divide in due parti fondamentali: l'intestazione e il corpo del documento.

E' facile comprendere che il corpo del documento contiene tutti gli elementi della pagina: il testo, le immagini, le applet Java, il codice Javascript e quant'altro viene materialmente visualizzato dal browser.

Al contrario, l'intestazione contiene una serie di informazioni necessarie al browser per una corretta interpretazione del documento, ma non visualizzate all'interno dello stesso. L'intestazione, quindi, ha un ruolo non apparente ma sicuramente fondamentale. Solo per citare alcuni elementi forniti dall'intestazione: il titolo della pagina, i termini chiave per i motori di ricerca, il tipo di HTML supportato ed i link base di riferimento.

In questa sede analizzeremo solo alcuni elementi, omettendo per il momento lo studio degli altri non immediatamente necessari alla comprensione di HTML:

- HTML
- DOCTYPE
- HEAD
- TITLE
- META

In seguito dettaglieremo per ognuno di questi elementi le peculiarità salienti.

<HTML>

Tutti gli elementi ed il contenuto di un documento HTML sono compresi all'interno dei marcatori <HTML></HTML> che, in altre parole, hanno il compito di aprire e chiudere il file. I tag <HTML></HTML> indicano al browser che il documento è marcato in HTML, anche se i browser più recenti (Netscape 3 e 4, MsIe 3,4 e 5) riescono ugualmente ad interpretare i tag successivi. Detto questo esistono comunque due ragioni per inserire sempre il tag <HTML></HTML> all'interno del documento:

HTML non è l'unico linguaggio di contrassegno presente sul WWW (si pensi solo a XML) e il browser rischia di mal interpretare i tag, confondendoli con altri linguaggi di markup

Gli utenti che usano vecchi browser rischiamo di visualizzare un documento pessimamente formattato.

DOCTYPE

Seguendo le indicazioni del W3C (consorzio internazionale che si occupa di armonizzare l'utilizzo dell'HTML) Doctype deve essere il primo elemento ad aprire il documento. Questo vuol dire che andrebbe posto prima di <HTML>. Si tratta di un tag che non ha bisogno di chiusura e che ha il compito di fornire informazioni al server Web che ospita la pagina. Le informazioni fornite da DOCTYPE riguardano il tipo di documento visualizzato oltre ad essere necessaria alla comunicazione tra browser e server. DOCTYPE deve essere scritto in una forma standard:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
```

Questa riga fornisce alcune informazioni sul documento:

HTML PUBLIC: il documento è pubblico

IETF: il tipo di HTML pubblico è gestito dalla Internet Engineering Task Force

DTD HTML 4.0: la versione di HTML supportata è la 4.0

EN: la lingua del documento è l'inglese

L'uso di DOCTYPE non è obbligatorio e può essere omesso. Certamente un suo utilizzo aiuta il server Web ad interpretare correttamente il documento, ma la sua assenza non è condizionamente ai fini della corretta visualizzazione.

Come si nota agevolmente DOCTYPE è un tag che non prevede un elemento di chiusura.

<HEAD>

Gli elementi <HEAD></HEAD> sono posti immediatamente dopo l'apertura del tag <HTML> e racchiudono l'intestazione vera e propria del documento; cioè tutte le informazioni necessarie al browser, al Webserver ed ai motori di ricerca. Si tratta del primo elemento letto dal browser e per questo è il luogo migliore per inserire sintassi script . All'intero di <HEAD></HEAD> va inserito il titolo del documento e altre informazioni. Ecco la sintassi HTML di un documento con i comandi finora esaminati:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD></HEAD>
</HTML>
```

<TITLE>

L'elemento <TITLE></TITLE> è il più utilizzato all'interno del tag <HEAD>, in quanto fornisce il titolo alla pagina. Il titolo viene solitamente visualizzato dai browser nell'intestazione di pagina. Quella che segue è l'immagine del TITLE di HTML.it (il testo è: HTML.it - il sito italiano sul Web publishing):



In caso di salvataggio dell'URL con "Aggiungi a preferiti" (per MsIe) e "Aggiungi Segnalibro" (per Netscape) il tag TITLE dà il nome al collegamento. In altre parole, quando si salva l'indirizzo, il browser assegna allo stesso quanto presente all'interno di <TITLE></TITLE>. L'immagine che segue mostra l'effetto in MsIe (del tutto simile per Netscape):



Il contenuto riportato tra i tag <TITLE></TITLE> è anche utilizzato da alcuni motori di ricerca per indicizzare la pagina e trovare parole chiave. Altavista è probabilmente l'esempio più eclatante. Per questo è bene fornire nel TITLE una descrizione dettagliata ma sintetica della pagina, con tutte le parole chiave che i motori possono indicizzare.

Un consiglio da tenere a mente è comunque quello di non esagerare con la

lunghezza del testo, ma di contemperare le esigenze di chi aggiunge il sito al bookmark e dei motori di ricerca.

La corretta sintassi per il tag TITLE è la seguente:

```
<TITLE>La mia prima home page con la guida di HTML.it</TITLE>
```

META

I motori di ricerca rappresentano una risorsa indispensabile per chi cerca informazioni sulla rete, e siti quali Altavista, Yahoo, Lycos e Excite sono tra i più visitati su internet.

Figurare all'interno di tali motori di ricerca è fondamentale per chi crea pagine web e vuole maggior visibilità. E' necessario innanzitutto segnalare le proprie pagine a questi motori di ricerca, e ciò viene fatto attraverso piccoli form presenti sulle pagine web sotto la scritta "Add your site". Periodicamente questi "spider" monitoreranno le milioni di pagine contenute nei loro database verificando le eventuali modifiche.

Oltre che essere inseriti nei motori di ricerca e' importante figurare tra i primi siti che vengono visualizzati nella ricerca (spesso, soprattutto quando i termini da ricercare sono di largo uso e di carattere generale, vengono presentate delle liste con migliaia di siti), e per fare questo è possibile adottare qualche semplice accorgimento.

Fondamentale a questo scopo risultano i META tag, stringhe di codice che figurano in testa al documento, tra i comandi <HEAD></HEAD> e che vengono, per prime, lette dai motori di ricerca.

Le "keywords" (parole chiave) sono i termini che, sinteticamente, descrivono il contenuto di una pagina web. Se, ad esempio, il sito si occupa della realizzazione di pagine Web, le keywords saranno:

HTML

realizzazione pagine Web

home page

motori di ricerca

prezzi modici

.....

Il META tag dovrà essere così impostato:

```
<META name="keywords" Content="HTML, realizzazione pagine Web, home page, motori di ricerca, prezzi modici">
```

La virgola divide i termini gli uni dagli altri. Le frasi non limitate a singoli vocaboli vanno scritte senza virgola perché il motore di ricerca le trovi in quella stessa forma.

Naturalmente, se il vostro sito e' multilingue sarà bene inserire keywords in italiano e nella lingua straniera presente nel sito, facendo sempre molta attenzione a non usare termini troppo generici. Nel caso in cui non si tratti di nomi propri e' buona regola inserire la forma singolare e plurale.

Il motore di ricerca considera il numero di volte in cui un termine e' presente all'interno della pagina e delle keywords, ma ripetere esageratamente una stessa parola nel META tag ha l'effetto opposto a quello voluto, in quanto la gran parte degli spider cancellano dal proprio database questi siti. E' bene quindi non esagerare col numero di termini all'interno delle keywords.

Altri META tag riguardano l'autore della pagina web:

```
<META name="author" content="Nome Cognome">
```

il titolo che apparira' alla fine della ricerca:

```
<META name="description" content="La mia prima home page con la guida di HTML.it">
```

e il nome dell'editor con cui il documento HTML e' stato generato:

```
<meta name="GENERATOR" content="Blocco note di WinXP">
```

Se per qualche ragione desideri che una delle tue pagine NON sia indicizzata nei motori di ricerca, devi inserire il seguente META tag:

```
<META NAME="ROBOTS" CONTENT="NOINDEX">
```

Un altro META Tag e' quello che permette il "refresh" della pagina. Cio' vuol dire che la stessa pagina o altra differente può essere automaticamente lanciata dopo un certo numero di secondi che noi stessi impostiamo.

Ecco il codice:

```
<META HTTP-EQUIV="Refresh" CONTENT="5; url=pippo.htm">
```

Dove CONTENT="5" e' il numero di secondi entro il quale la nuova pagina sara' caricata; mentre url=pippo.htm" e' il file che verra' caricato.

RIEPILOGO

Di seguito riportiamo una pagina HTML impostata con il codice visto in questa lezione. Questa pagina costituirà un punto di riferimento anche per le lezioni successive:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>

<META name="keywords" Content="HTML, realizzazione pagine Web, home
page, motori di ricerca, prezzi modici">
<META name="description" content="La mia prima home page con la guida
di HTML.it">
<meta name="GENERATOR" content="Blocco note di Win95">
<META name="author" content="Nome Cognome">

<TITLE>La mia prima home page con la guida di HTML.it</TITLE>

</HEAD>

</HTML>
```

Scrivere correttamente i meta tag

Lo studio dei dati forniti dai più frequentati Search Engine ha dimostrato come il termine più ricercato sia "sex", e come l'80% degli utenti non consulta oltre la seconda pagina di ogni ricerca effettuata. Ciò significa che se una ricerca per parola chiave risponde con 30 pagine di indirizzi Web, soltanto 2 utenti su 10 consulteranno oltre la seconda pagina. In ultima analisi, i siti contenuti nella trentesima pagina hanno una visibilità nulla. Questa considerazione dovrebbe chiarire la vitale importanza di risultare tra i primi posti in queste ricerche. E' dimostrato che i Search Engine e le directory sono i mezzi che più di altri (riviste, link su altri siti, televisione ecc.) portano accessi e visitatori in un sito.

Alla luce di queste considerazioni, l'inserimento di un sito all'interno di Search Engine o Directory è un'operazione fondamentale per la riuscita del sito stesso. Accorgimenti e piccoli trucchi possono aiutare in questo, ma non c'è un modo univoco per risultare in testa ad ogni motore di ricerca, per la banale constatazione che ogni motore ha peculiarità proprie. Prima di continuare è utile precisare alcuni concetti:

- non è importante risultare in tutti i motori di ricerca ma al massimo nei 10 più diffusi a livello mondiale e nei 2 a livello nazionale;
- diffidare dei numerosi servizi che per pochi dollari promettono l'inserimento in 500 motori di ricerca, perchè spesso non danno risultati e perchè, appunto, non ha alcuna rilevanza risultare nei motori minori;
- diffidare dai programmi automatici di inserimento;
- non sperare che le "keywords" dei Meta Tag siano la soluzione unica a questi problemi;
- spesso l'inserimento non è immediato, e in alcuni casi (Lycos e HotBot) è successo addirittura che il database non venisse aggiornato per mesi.

Il numero di persone che negli USA si occupano esclusivamente di inserimento siti nei motori di ricerca è stimato intorno alle 100 unità. Ciò significa che nessuna Directory può materialmente controllare tutti i nuovi siti inseriti, per cui molti vengono indicizzati senza essere previamente verificati. Questo stato di cose genera una certa confusione, oltre ad uno scadimento qualitativo degli inserimenti. Infoseek ha reso noto che giornalmente riceve 30.000 nuove richieste di inserimento, poco meno della metà di Yahoo.com.

L'inserimento nelle Directory è, per loro stessa natura, più umanizzato e difficilmente oggetto di trucchi più o meno leciti. Anzi, tentare di risultare

furbescamente ai primi posti delle Directory può essere controproducente, visto che gli operatori, resisi conto del trucco, inseriscono il sito nel "limbo dei furbi" in posizione quasi invisibile. Diverso, invece, il caso dei Search Engine dove non è un essere umano a ricevere le richieste di inserimento ma uno spider. Nel corso degli anni gli spider, non troppo perspicaci agli esordi, hanno affinato i propri strumenti e in molti casi radicalizzato l'intolleranza verso trucchi e trucchetti. Spesso è sufficiente una keyword ripetuta due volte per causare la cancellazione, tout court, del sito dal database. Prima di tutto, quindi, è buona regola evitare le penalizzazioni. Trucchi che fino ad un anno fa permettevano di risultare primi nei Search Engine oggi sono riconosciuti dagli spider. Per esempio non è una buona idea inserire testo nascosto nella pagina HTML di colore identico al background, come non è una buona idea creare pagine con parole chiave che non hanno nulla a che vedere col contenuto del sito, ma inserite solo perchè spesso ricercate dagli utenti (per es il termine "sex" in un sito che si occupa di informatica). Evitare in modo assoluto di inserire pagine con un meta tag "refresh" che rimanda ad un'altra pagina dopo un certo numero di secondi. Premesso che il delitto non paga, ecco una serie di suggerimenti per ottenere un buon posizionamento nei Search Engine:

- evitare, per quanto possibile, l'uso di frame HTML;
- non iniziare il documento HTML con un'immagine o una tabella;
- porre particolare attenzione alla home page del sito;
- evitare, per quanto possibile, l'uso di pagine ASP (Active Server Pages) o nominate con un nome di file dove compare un punto interrogativo "?";
- inserire le parole chiave del sito tra i tag <TITLE></TITLE>, perchè molti spider (Altavista in primis) danno molta importanza a questo;
- usare i tag <H1><H2> ecc.
- creare "doorpage", cioè delle pagine di supporto a quelle principali, contenenti testo e keywords attinenti al sito;
- registrare non più di una pagina al giorno;
- registrare domini di primo livello. Per esempio www.nome.com piuttosto che www.nome.com/directory;
- il meta tag "keywords" non deve contenere termini duplicati e non più di mille caratteri. Il 90% delle ricerche avviene per termini scritti in minuscolo, per cui è preferibile evitare quelli maiuscoli;
- il meta tag "description" è fondamentale perchè a questo il Search Engine associa il link al sito e non deve superare i 300 caratteri.

<BODY> Impostare lo sfondo

Nella precedente lezione abbiamo visto come creare da zero un documento HTML e come impostare il titolo ed i meta tag.

L'operazione successiva all'impostazione del documento è la definizione del colore o dell'immagine di sfondo, oltre ai colori dei link attivi e visitati.

<BODY>

L'elemento <BODY> è posto in posizione immediatamente successiva alla chiusura del tag </HEAD> e comunque all'interno degli elementi <HTML></HTML>; ha un tag di apertura e uno di chiusura, ed all'interno di esso si sviluppa il corpo del documento.

Se l'elemento <HEAD> conteneva metadati (cioè non materialmente visualizzati dal browser) la funzione del tag <BODY> è quella di mostrare gli oggetti (testo, immagini, suoni, applet ecc) della pagina.

La sintassi corretta per l'elemento <BODY> è la seguente:

```
<BODY>
Contenuto del documento
</BODY>
```

Il tag <BODY> è utilizzato, oltre che per fornire al browser indicazioni sulla posizione degli oggetti nel documento, anche per impostare vari attributi di visualizzazione per il documento. Di seguito vediamo quali.

BGCOLOR

L'attributo BGCOLOR imposta un colore unitario di sfondo. La sintassi corretta è la seguente:

```
<BODY BGCOLOR="red">
```

E' possibile sostituire al nome in inglese, valori esadecimali. Per esempio, il colore rosso (red) si sostituisce in questo modo:

```
<BODY BGCOLOR="#ff0000">
```

L'utilità dei colori esadecimali si ha laddove non si vuole un colore standard ma sfumato o con diversa tonalità. I più diffusi editor HTML prevedono palette per la definizione di colori esadecimali, mentre Paint Shop Pro fornisce, oltre al colore, anche il corrispondente valore esadecimale da copiare/incollare.

BACKGROUND

BACKGROUND ha una funzione simile a BGCOLOR, ma mentre il secondo mostra un tinta unica del colore, il primo visualizza sullo sfondo un'immagine in formato grafico .gif o .jpg.

Consideriamo, per esempio, di voler costruire uno sfondo con l'immagine seguente:



L'immagine si chiama sfondo.gif ed è depositata nella stessa directory del documento. La sintassi corretta per impostare l'immagine come sfondo è:

```
<BODY BACKGROUND="sfondo.gif">
```

Il browser visualizza l'immagine sfondo.gif e la ripete su ogni punto dello schermo disponibile. In altre parole non si limita a visualizzare l'immagine una sola volta, magari al centro della pagina, ma riempie ogni spazio disponibile.

Per questa ragione è assolutamente necessario creare uno sfondo che, se ripetuto, non presenti soluzione di continuità, ma un aspetto il più possibile uniforme.

E' bene scegliere un'immagine di sfondo che non infastidisca la lettura e che sia il più possibile coerente con il colore del testo. Per esempio, inserire un testo arancione su uno sfondo rosso non renderebbe leggibile il testo. Sempre meglio usare il colore nero per il testo e tinte leggere per lo sfondo.

TEXT

Se non stabilito diversamente il colore del testo del documento è nero, in quando i browser assegnano quel colore di default. Grazie all'attributo TEXT è possibile assegnare al testo un colore diverso dal nero. Questa la giusta sintassi:

```
<BODY BACKGROUND="sfondo.gif" TEXT="red">
```

Anche in questo caso i colori possono esprimersi in nomi o valori esadecimali. All'interno del documento è possibile marcare parte del testo in colori differenti da quello impostato su TEXT.

LINK

Se non stabilito diversamente il colore dei link (non ancora visitati) è blue. Grazie all'attributo LINK è possibile definire colori differenti:

```
<BODY BACKGROUND="sfondo.gif" LINK="red">
```

Tutti i link della pagina non saranno più blue ma rossi (red). Tale colore può essere espresso in valori esadecimali.

ALINK

Quando cliccati i link assumono un colore diverso da quello impostato su LINK (o dal blue di default). Grazie a ALINK (la A sta per Active) è possibile modificare questo colore:

```
<BODY BACKGROUND="sfondo.gif" ALINK="yellow">
```

VLINK

Quando un URL associato ad un link viene visitato, quest'ultimo assume un colore diverso da quello di LINK (o dal blue di default). Grazie a VLINK (la V sta per Visited) è possibile agire su questo colore:

```
<BODY BACKGROUND="sfondo.gif" VLINK="green">
```

BGPROPERTIES

Trattando dell'attributo BACKGROUND abbiamo sottolineato come le immagini richiamate siano disposte sullo schermo disponibile. Se, comunque, la pagina è tanto lunga da attivare lo scroller laterale, lo sfondo (e l'immagine associata) scorre insieme alla pagina. Per comprendere meglio di cosa stiamo parlando clicca qui per un esempio, e poi scrolla la pagina verso il basso.

Grazie alla proprietà BGPROPERTIES è possibile rendere lo sfondo immobile rispetto allo scroller di pagina. Questa la sintassi corretta:

```
<BODY BACKGROUND="sfondo.gif" BGPROPERTIES="fixed">
```

Questo espediente funziona solo con MsIe e non con Netscape che invece continua a scrollare la pagina.

Suoni o musiche di sottofondo

La presenza di file audio sotto forma di MIDI o WAV puo' essere una piacevole colonna sonora alla navigazione delle pagine web, oppure un fastidioso fardello capace solo di rallentare il caricamento ed innervosire il visitatore. Come spesso viene ripetuto in questa guida e' necessario considerare che si', e' importante la piacevolezza grafica e musicale delle pagine web, ma e' fondamentale che la navigazione non risulti lenta e, in ultima analisi, impossibile per chi abbia a cuore le sorti della propria bolletta telefonica. Il mio consiglio e' quindi di evitare grossi file MIDI (60Kb di files MIDI ad esempio sono esagerati) ed enormi WAVE (che possono essere sostituiti da altri files piu' leggeri, come vedremo in seguito).

Consideriamo un file MIDI; consideriamo che si chiami "imagine.mid"; e consideriamo che si voglia inserirlo come sottofondo ad pagina web non appena questa si apra, automaticamente quindi.

Dobbiamo innanzitutto considerare che Navigator e MSIE si servono di comandi diversi per richiamare automaticamente un file MIDI di sottofondo.

Per esempio il comando <BGSOUND> funziona solo con MSIE ma non con Navigator (LOOP indica il numero delle volte che il midi deve essere ripetuto):

```
<BGSOUND SRC="imagine.mid" LOOP=INFINITE>
```

Clicca qui sia con MSIE che Navigator per provare.

Oltre ai files MIDI, con il comando <BGSOUND> e' possibile inserire files .WAV e .AU

Per rendere possibile l'apertura automatica di un files audio anche con Netscape, si deve usare il comando <EMBED>:

```
<EMBED SRC="imagine.mid" AUTOSTART=true LOOP=true VOLUME="80" WIDTH="0" HEIGHT="0">
```

Anche in questo comando oltre ai MIDI e' possibile inserire files .WAV e .AU. Una delle remore all'inserimento di files midi all'interno delle proprie pagine, e' dovuta al fatto che, ogni volta che si cambia pagina il file musicale si interrompa bruscamente. Il problema non persiste nei siti divisi in frame, in quanto il file musicale viene caricato nel frame fisso e quindi non muta al cambiamento delle pagine negli altri frame. E per i siti senza frame? Un

piccolo espediente puo' essere quello mediante il quale, proprio prendendo spunto dai frames, si costruisce una pagina con un frame "invisibile". Per fare questo si crea una pagina con il seguente codice:

```
<FRAMESET rows="0,*" border=0 frameborder="0" framespacing="0">  
<frame me name="alto" src="top.htm" noresize>  
<frame me name="centrale" src="central.htm" noresize>  
</FRAMESET>
```

Dove nel frame "alto" inseriremo il codice che richiama il file audio:

```
<EMBED SRC="image.mid" AUTOSTART=true LOOP=true VOLUME="80"  
WIDTH="0" HEIGHT="0">
```

E nel frame "centrale" il contenuto delle pagine.

Altro "stratagemma" per far ascoltare un file audio (SOLO CON NETSCAPE) in modo continuativo anche quando si cambia pagina, e' quello di caricare tale file in una finestra del browser alternativa.

Per fare questo e' necessario servirsi di un piccolo ma importantissimo TARGET.

I TARGET vengono usati soprattutto nei frame ma possono esserlo anche in altri casi. Se per esempio si desidera che una pagina collegata ad un link venga caricata in una finestra diversa del browser, si dovra' inserire il TARGET="_new".

Facciamo conto che il midi da ascoltare si chiami "image.mid"; ecco il codice:

```
<A HREF="image.mid" TARGET="_new">Ascolta il midi image.mid</A>
```

Cliccando si aprira' una finestra di default la quale si affianchera' a quella principale (da cui si e' lanciato il link).

Questa procedura e' valida solo per Netscape.

In questo modo il visitatore potra' leggere il testo presente sulla finestra da cui ha lanciato il file "image.mid" e nel frattempo l'altra finestra produrra' il midi. Nel momento in cui il visitatore clicchera' su un altro midi da ascoltare, automaticamente tale midi verra' caricato nella finestra precedente aperta per "image.mid" (non verranno aperte, cioe', decine di finestre).

Purtroppo questo espediente funziona solo con Netscape e non con MSIE. Per venire incontro anche agli utenti del browser di Redmond e' necessario architettare una procedura un po' piu' complessa, la quale consiste nell'associare al link non il semplice file midi "image.mid" ma un file HTML che contenga un comando EMBED che faccia automaticamente partire il midi.

 formattare il testo

Prima di approfondire le tematiche legate alla formattazione del testo è bene precisare alcuni concetti che reggono l'uso di questi strumenti.

Tag di testo logici e fisici

HTML è un sistema di contrassegno con il compito di definire la struttura e l'aspetto di un documento. Questa definizione che senza altri commenti passerebbe inosservata, va invece approfondita per capire meglio i concetti di tag fisici e logici.

Un tag fisico ha il compito di formattare visivamente un documento , cioè di renderlo grassetto, corsivo, sottolineato ecc. La funzione di questi tag non è quella di dare un aspetto al documento, ma di marcare determinati punti per definirne una struttura.

Un tag logico, al contrario ha la funzione di definire una struttura indipendentemente dal suo aspetto . Ovvero, indipendentemente dal modo in cui il browser interpreterà visivamente il contrassegno.

Per comprendere meglio questa differenza prendiamo in esame due elementi volti a formattare testo:

<ADDRESS></ADDRESS>

Si tratta di un tag logico che contrassegna l'informazione di indirizzo nel documento. In altre parole quando si è in presenza di un indirizzo e-mail, di un numero di telefono o di un indirizzo secondo le regole di HTML va inserito questo tag logico.

<I></I>

Si tratta di un tag fisico che rende corsivo (italic) il testo compreso nell'elemento. E' quindi, un tag volto a dare un aspetto al documento e non a marcare un elemento di struttura.

Vediamo insieme qual è il risultato visivo di un loro utilizzo all'interno di un documento Web:

webmaster@html.it

webmaster@html.it

Senza guardare il source del documento ti sfidiamo a capire quale testo sia stato formattato con ADDRESS e quale con I. Visivamente entrambi i tag (il

primo è ADDRESS e il secondo I) danno un medesimo risultato (il testo corsivo), ma da un punto di vista di struttura del documento, solo il primo (ADDRESS) indica che al proprio interno è presente un indirizzo, mentre il secondo non da' alcuna informazione di questo tipo.

Tag FONT e nuovo standard HTML 4

HTML 4 è lo standard che da quasi due anni regge le sorti del linguaggio di markup più famoso al mondo. Senza entrare nel merito delle novità del nuovo standard vogliamo sottolineare come il W3C abbia "deprecato" l'uso del tag FONT nella formattazione del testo in HTML. Il termine "deprecare" (letteralmente tradotto dall'inglese) indica che i browser leggono ancora questo tag, ma il suo uso è decisamente vietato dal nuovo standard. Secondo HTML 4 la formattazione del testo è lasciata unicamente ai fogli di stile.

In questa lezione non terremo conto delle indicazioni del W3C e spiegheremo comunque l'uso dei tag classici di HTML 4. Rimandiamo comunque alla sezione sui Fogli di stile di questo sito per un approfondimento dell'argomento CSS.

Formattare titoli da <H1> ad <H6>

Il tag <Hn> (dove n è il numero da 1 a 6 che è possibile assegnare all'elemento) ha la funzione di fornire stili ai titoli di pagina, dando maggiore o minore enfasi a seconda del numero inserito. Come accennato i numeri vanno da 1 a 6, con maggior importanza dei numeri più bassi rispetto agli alti. Visivamente tale diversa importanza si traduce in differente grandezza del testo.

Oltre a definire gli stile dei titoli, il tag <Hn> inserisce spaziature di paragrafo rispetto agli elementi che seguono.

Il tag fisico FONT è uno dei più usati e frequenti nell'attuale Web publishing. Nella sezione: Impostare lo sfondo del documento (elemento BODY) abbiamo visto come l'attributo TEXT dia un colore univoco a tutto il testo del documento. Il tag FONT ha in parte una funzione simile, ma molto più ampia e concettualmente differente.

Il tag FONT ha la funzione di formattare tipo e grandezza del testo limitatamente ad alcuni punti del documento. In realtà se il tag font aprisse e chiudesse il documento, la totalità del testo compreso al suo interno sarebbe formattato di conseguenza. Ma concettualmente il tag FONT è stato pensato per definire parti limitate di testo. Mentre, poi, TEXT determina solo il colore del testo, il tag FONT può definire: tipo di font utilizzato, grandezza e colore.

Segue la sintassi corretta del tag:

```
<FONT FACE="arial" SIZE=5 COLOR=red>Carattere da formattare</FONT>
```

L'attributo FACE="arial" indica il tipo di carattere da richiamare per la visualizzazione. SIZE=5 la grandezza del carattere che può andare da 1 a 7 (con maggiore grandezza dei numeri vicini al 7); COLOR=red indica il colore del testo (esprimibile anche in valori esadecimali).

, <I>, <U>

, <I>, <U> sono tre dei più usati tag di formattazione fisica dell'HTML. Tutti vanno aperti e successivamente richiusi:

** Testo grassetto **

Il testo compreso tra questi tag è trasformato in grassetto (la B sta per BOLD).

<I> Testo corsivo </I>

Il testo compreso tra questi tag è formattato in corsivo (la I sta per ITALIC)

<U> Testo sottolineato </U>

Il testo compreso tra questi tag è sottolineato pur non essendo un link (la U sta per UNDERLINE).

Esiste anche il tag STRIKE per il testo barrato:

```
<STRIKE> Testo sottolineato </STRIKE>
```

<SUP> e <SUB>

Questi due tag di formattazione fisica creano rispettivamente: apici posti leggermente più in alto del normale testo (SUP) e pedici posti leggermente più in basso (SUB). E' possibile annidare diversi tag per ottenere effetti di sovrapposizione successiva. Questi elementi vengono usati soprattutto per indicare note o formule matematiche.

Questa la corretta sintassi:

```
HTML.it <SUP>©</SUP> è un marchio registrato
```

```
HTML.it <SUB>©</SUB> è un marchio registrato
```

Testo preformattato

HTML legge il testo ovviamente da sinistra verso destra e, senza alcun tipo di formattazione, manda a capo alla fine di ogni riga senza soluzione di continuità. La formattazione successiva esaminata in questa lezione ha il compito di indicare al browser dove andare a capo, quale font utilizzare, quali parole enfatizzare e così via. Spesso, però, un documento creato con un normale editor di testo deve essere importato senza formattazione, così

come è stato creato. A questo scopo HTML prevede i tag di testo preformattato (i quali non leggono ne' interpretano alcun tag di HTML, neanche i caratteri speciali):

```
<XMP> Testo da formattare </XMP>
```

```
<PRE> Testo da formattare </PRE>
```

Entrambi i tag hanno lo stessa funzione, anche se lo standard di HTML 4 prevede l'uso di XMP e non di PRE.

Breve descrizione degli stili logici

Abbiamo descritto in apertura di lezione la differenza tra tag fisici e logici. Ora elenchiamo i tag logici più interessanti e il corrispondente utilizzo in HTML:

<ADDRESS>

Contrassegna informazioni di indirizzo (mail, telefono ecc.)

<BLOCKQUOTE>

Usato per citazioni più lunghe di due o tre righe

<CITE>

Usato per indicare la fonte della citazione

<CODE>

Viene utilizzato per formattare righe di codice di programmazione

<DFN>

Indica che il testo compreso è una definizione.

Enfatizza il testo compreso all'interno del tag

<KBD> e <SAMP>

Poco utilizzati. Indicano cose che il computer dovrebbe dire all'utente e viceversa, secondo un tipico concetto di Unix

Enfatizza il testo

<VAR>

Legato a CODE, indica le variabili di programmazione.

Paragrafi e giustificazione

Nella lezione precedente abbiamo visto come formattare del semplice testo in HTML. A questo punto dobbiamo affrontare il modo in cui sistemare, dividere, posizionare il testo e gli altri oggetti di una pagina HTML all'interno del documento.

Creare paragrafi con **<P>**

Il tag **<P>** definisce un nuovo paragrafo del testo indicando al browser che lo stesso deve rimanere su una nuova riga ed essere posizionato a destra, a sinistra o al centro. Se non specificato oltre, il tag **<P>** allinea il testo di default sulla sinistra. Per indicare altri tipi di posizionamento esistono attributi specifici:

<P ALIGN=left>

Definisce un paragrafo e allinea sulla sinistra (left).

<P ALIGN=right>

Definisce un paragrafo e allinea sulla destra (right).

<P ALIGN=center>

Definisce un paragrafo ed allinea al centro (center).

Andare a capo con **
**

**
** è un tag di interruzione di riga. Ha un funzionamento simile al paragrafo visto in precedenza, ma se ne discosta perchè non inizia un nuovo paragrafo.

In altri termini la sua funzione è simile alla pressione del tasto "invio" della tastiera.

Va usato singolarmente senza tag di chiusura.

Posizionare il testo con **<DIV ALIGN>** e **<CENTER>**

L'elemento **<DIV>** viene utilizzato per allineare il testo in posizione orizzontale a sinistra, destra e centro della pagina. L'attributo **ALIGN** è fondamentale a questo scopo.

```
<DIV ALIGN=left>Testo e immagini a sinistra</DIV>
```

Sposta gli elementi contenuti tra i suoi tag sulla sinistra.

```
<DIV ALIGN=right>Testo e immagini a destra</DIV>
```

Sposta gli elementi sulla destra.

```
<DIV ALIGN=center>Testo e immagini centrate</DIV>
```

Sposta gli elementi in posizione centrale.

Il tag `<CENTER>` ha un funzionamento del tutto simile a `<DIV ALIGN=center>` ma il suo uso è stato "deprecato" dallo standard 4 di HTML. L'uso di `<CENTER>` è stato introdotto da Netscape, proprio quando l'HTML prevedeva l'uscita di DIV. L'enorme diffusione iniziale di Netscape ha reso questo tag altrettanto diffuso e per questa ragione ancora viene usato e riconosciuto dai browser (anche da MsIE). L'uso di `<CENTER>` è molto semplice:

```
<CENTER>Testo da centrare</CENTER>
```

Linee orizzontali con `<HR>`

Le linee orizzontali sono un ottimo strumento per dividere parti del documento e rendere il testo più leggibile. La sintassi necessaria al loro inserimento in un documento HTML è la seguente:

```
<HR align="CENTER" size="2" width="400" color="Red" noshade>
```

Il tag HR (acronimo che sta per Horizontal Rule) non ha bisogno di chiusure successive. Si compone di diversi attributi:

aling="CENTER": definisce la posizione della riga (center, right, left).

size="2": definisce l'altezza, in pixel, della riga.

width="400": definisce lunghezza orizzontale, in pixel, della linea. Può anche esprimersi in percentuale di spazio disponibile: width=80%.

color="RED": definisce il colore della linea.

noshade: se presente questo attributo elimina l'effetto 3D della linea. Se omissso produce tale effetto.

Creare elenchi puntati e numerati

Gli elenchi rappresentano uno dei modi più diffusi per organizzare informazioni all'interno di siti Web. Una delle loro caratteristiche principali è sicuramente quella di fornire un quadro chiaro e sintetico dell'argomento trattato.

HTML mette a disposizione diversi tipi di elenchi che di seguito andiamo a trattare singolarmente.

Elenchi ordinati (numerati)

Gli elenchi ordinati sono costituiti da un singolo tag di apertura e chiusura `` e tanti tag di apertura per quante sono le voci di menu ``. Questa è la corretta sintassi per creare elenchi ordinati:

```
<OL>
<LI> Prima voce di menu
<LI> Seconda voce di menu
<LI> Terza voce di menu
</OL>
```

Dall'esempio si nota come è possibile omettere il tag `
` per il ritorno a capo, visto che è automaticamente inserito da ``. Se non stabilito diversamente (come vedremo in seguito) il tipo di elenco ordinato che il browser visualizzato è numerato (cioè l'elenco è costituita da una serie di numeri che da 1 crescono progressivamente). Le ultime versioni di HTML prevedono la possibilità di elenchi ordinati che contengono un sistema di indicizzazione diverso dai numeri visti in precedenza:

Indicizzazione alfabetica maiuscola:

```
<OL TYPE=A>
<LI> Prima voce di menu
<LI> Seconda voce di menu
<LI> Terza voce di menu
</OL>
```

Indicizzazione alfabetica minuscola:

```
<OL TYPE=a>
<LI> Prima voce di menu
<LI> Seconda voce di menu
<LI> Terza voce di menu
</OL>
```

Indicizzazione con numeri romani maiuscoli:

```
<OL TYPE=I>  
<LI> Prima voce di menu  
<LI> Seconda voce di menu  
<LI> Terza voce di menu  
</OL>
```

Indicizzazione con numeri romani minuscoli:

```
<OL TYPE=i>  
<LI> Prima voce di menu  
<LI> Seconda voce di menu  
<LI> Terza voce di menu  
</OL>
```

Elenchi NON ordinati (puntati)

Gli elenchi non ordinati funzionano in modo simile a quelli ordinati. La differenza di fondo è che le risorse indicizzate non sono legate da stretti rapporti di successione gerarchica, per cui non sono previsti elenchi progressivi quali numeri o lettere.

Gli elenchi non ordinati (o puntati) si compongono di un tag unico di apertura e chiusura `` e tanti tag di elenco per quante sono le voci da indicizzare ``. La corretta sintassi per definire un elenco puntato è la seguente:

```
<UL>  
<LI> Prima voce di menu  
<LI> Seconda voce di menu  
<LI> Terza voce di menu  
</UL>
```

Se non definito diversamente l'elenco è costituito da una serie di pallini. Come per gli elenchi numerati anche in questo caso è possibile indicare diversi tipi di elenco:

I pallini pieni sono ottenibili con **disc**:

```
<UL TYPE=disc>  
<LI> Prima voce di menu  
<LI> Seconda voce di menu  
<LI> Terza voce di menu  
</UL>
```

L'attributo **circle** imposta pallini vuoti all'interno:

```
<UL TYPE=circle>  
<LI> Prima voce di menu  
<LI> Seconda voce di menu  
<LI> Terza voce di menu  
</UL>
```

L'attributo **square** imposta elenchi definiti da quadratini pieni:

```
<UL TYPE=square>  
<LI> Prima voce di menu  
<LI> Seconda voce di menu  
<LI> Terza voce di menu  
</UL>
```

 inserire immagini

Internet, solo negli ultimi 10 anni affermata come media di massa, è stato per lunghi anni un sistema di comunicazione ad esclusivo utilizzo di apparati militari prima, e universitari dopo. E' agevole immaginare quanto poco inclini alla grafica ed all'estetica fossero i vecchi navigatori della Rete. Alla fine degli anni Ottanta Internet ha subito una radicale trasformazione che ha portato sulle "autostrade telematiche" un'utenza di massa, con esigenze diverse da quelle militari o accademiche. Con la trasformazione della Rete si è imposta l'esigenza di mettere a disposizione strumenti di navigazione più "user friendly" per esigenze non più meramente di studio e ricerca. Così, nel 1989 nasce il WWW che grazie al primo browser della storia, Mosaic, trasforma il testo bianco su sfondo nero nell'attuale Web, fatto di colori e interattività.

La rivoluzione dell'immagine (criticata da alcuni puristi accademici che videro nell'Internet di massa la fine del sistema di comunicazione) fu anche merito di un tag oggi molto comune , cioè l'elemento necessario a richiamare immagini all'interno della pagina.

Prima di trattare nello specifico il tag principale per l'inserimento di immagini negli ipertesti è bene precisare alcuni concetti di HTML. A differenza di molti editor di testo (Ms Word, per esempio) gli ipertesti non vengono "fusi" con le immagini a corredo grafico, ma si limitano a richiamarle da un percorso specifico in locale o su Web. In altre parole esiste una soluzione di continuità molto chiara tra file .htm e immagini (ma anche suoni, applet ecc.). I documenti HTML si limitano a prevedere uno spazio al proprio interno entro il quale vanno inserite le immagini richiamate.

Il tag non ha bisogno di chiusura e la sua corretta sintassi è la seguente:

```
<IMG SRC="immagine.gif">
```

Dove SRC sta per Search ed è il percorso dal quale il browser ricava l'immagine (in questo caso "immagine.gif"). Come detto si tratta di un tag singolo, nel senso che NON va chiuso con un corrispondente .

I browser (Netscape, MsIe, Opera ecc.) riconoscono molti formati grafici, anche se sono due quelli più utilizzati: GIF (Graphics Interchange Format) e

JPEG (Joint Photographics Experts Group). Negli ultimi mesi sta diffondendosi su scala mondiale un altro formato: PNG (Portable Network Graphics).

L'elemento è corredato da diversi attributi molto utili ai fini del suo utilizzo. Vediamo insieme quali.

ALT

L'utilizzo di commenti testuali alle immagini è una fondamentale regola da osservare nella creazione di siti Web per alcune buone ragioni:

- taluni browser potrebbero essere impostati per non richiamare le immagini;
- i browser testuali per non vedenti non riuscirebbero ad interpretare le immagini senza un commento.
- è possibile evitare didascalie inserendo commenti all'interno dell'immagine stessa.

In tutti i casi considerati l'uso di commenti risolve il problema e consente di ottimizzare l'utilizzo di una pagina Web. La corretta sintassi per i commenti è la seguente:

```
<IMG SRC="immagine.gif" ALT="Opera di K. Haring">
```

Per verificare gli effetti passa con il mouse sull'immagine.

WIDTH e HEIGHT

Negli esempi finora indicati non abbiamo specificato le misure dell'immagine che il browser ha provveduto a cercare automaticamente. **E' possibile definire altezza e larghezza dell'immagine con gli attributi width ed height:**

```
<IMG SRC="immagine.gif" WIDTH=178 HEIGHT=180 ALT="Opera di K. Haring">
```

Dove **WIDTH=178** è la **dimensione orizzontale (larghezza)** dell'immagine espressa in pixel, e **HEIGHT=180** la **dimensione verticale (altezza)**.

Con questi attributi è possibile definire dimensioni differenti da quelle effettive dell'immagine. E', comunque, consigliabile inserire immagini di dimensioni naturali soprattutto se in formato GIF, visto che ridimensionando questo formato la qualità decade enormemente.

BORDER

Con l'attributo **BORDER** è possibile assegnare un bordo all'immagine. I valori sono numerici ed espressi in pixel. Il valore **BORDER** impostato su 0 ha l'effetto di non visualizzare alcun bordo. Se questo attributo viene omesso il browser non assegna nessun bordo, ma se l'immagine è anche un link viene automaticamente assegnato un **BORDER=1**. Questa che segue è la giusta sintassi:

```
<IMG SRC="immagine.gif" WIDTH=178 HEIGHT=180 BORDER=2  
ALT="Opera di K. Haring">
```

HSPACE e VSPACE

Con questi due attributi è possibile stabilire la distanza in pixel dell'immagine dagli oggetti che si trovano ai quattro lati della stessa.

HSPACE definisce la distanza dai lati destro e sinistro dell'immagine degli oggetti più vicini (testo, immagini, applet ecc.).

VSPACE definisce la distanza dai lati superiore e inferiore dell'immagine degli oggetti più vicini (testo, immagini, applet ecc.). Questa la corretta sintassi:

```
<IMG SRC="immagine.gif" WIDTH=178 HEIGHT=180 BORDER=2  
VSPACE=3 HSPACE=3 ALT="Opera di K. Haring">
```

Questi attributi risultano utili quando si vuole distanziare l'immagine dagli oggetti più vicini.

ALIGN

L'attributo **ALIGN** definisce la posizione dell'immagine rispetto al testo posto immediatamente prima o dopo la stessa. Esistono varie opzioni per l'attributo **ALIGN**:

- **ALIGN=top**: allinea la prima riga di testo sulla sinistra al top dell'immagine.
- **ALIGN=middle**: allinea la prima riga di testo sulla sinistra al centro dell'immagine.
- **ALIGN=bottom**: allinea la prima riga di testo sulla sinistra nella parte più bassa dell'immagine.
- **ALIGN=left**: allinea il testo sulla destra dell'immagine partendo dal top.
- **ALIGN=right**: allinea il testo sulla sinistra dell'immagine partendo dal top.

<HREF> creare link

Molte volte avrai sentito parlare di ipertesti e di come questi reggano il Web. Un ipertesto è un modo di formattare documenti in forma non sequenziale. Per comprendere questa caratteristica si pensi ad un libro cartaceo che si struttura in: indice, capitoli e glossario. Ha, dunque, una struttura sequenziale nel senso che si legge l'indice per passare ai capitoli (dal primo all'ultimo) e terminando con il glossario dei termini. Il lettore, in questo caso, ha un percorso di lettura che lo porta necessariamente a leggere il primo capitolo per giungere all'ultimo. Con gli ipertesti questa struttura sequenziale cessa di esistere grazie ai collegamenti ipertestuali (link) che permettono di leggere il documento senza seguire necessariamente un ordine sequenziale. La peculiarità di saltare da un punto all'altro del documento è tipica del WWW, dove spesso seguendo un documento si passa da un sito all'altro senza soluzione di continuità.

I documenti HTML sono degli ipertesti il cui funzionamento è in massima parte dovuto agli HyperLink o ancoraggi, il cui tag specifico è <A>.

<A HREF>

L'elemento <A> (la A sta per Anchor) ha bisogno di un tag di apertura e chiusura e al suo interno è possibile inserire testo, immagini o altri elementi multimediali.

Perchè funzioni, l'elemento <A> deve essere associato ad altri attributi. Il più importante di questi è HREF (abbreviazione di Hypertext Reference) contenente l'URL o la pagina da raggiungere. Questa la sintassi:

```
<A HREF="http://www.html.it">Visita HTML.it</A>
```

Nel codice sopracitato cliccando sul testo "Visita HTML.it" (compreso tra i tag A in apertura e chiusura) si raggiunge l'URL <http://www.html.it> indicato dall'attributo HREF.

E' possibile sostituire al testo un'immagine con effetto identico.

HREF può contenere sia link a risorse esterne (come in questo caso) ovvero link ad altri documenti dello stesso sito. Per esempio, se ci troviamo nella pagina 1.htm e vogliamo inserire un link alla pagina 2.htm presente nella stessa cartella, il codice corretto è:

```
<A HREF="2.htm">Clicca qui per raggiungere la nuova pagina</A>
```

TARGET

TARGET è un attributo implementato per esigenze legate alla gestione dei frame. In una pagina divisa in frame, infatti, questo attributo indica in quale di questi debba essere visualizzato il documento. Non entriamo nel merito dei frame.

In questa sede possiamo, però, trattare di un uso ulteriore di questo attributo. **Grazie a TARGET è possibile caricare un pagina associata a HREF in un'altra finestra del browser**, attraverso la seguente sintassi:

```
<A HREF="http://www.html.it" TARGET="_new">Visita HTML.it</A>
```

L'attributo TARGET="_new" indica al browser di caricare il link a <http://www.html.it> in una nuova (_new) finestra.

Il lancio di nuove finestre da link può essere molto utile nel caso in cui si rimandi a risorse esterne e non si voglia perdere il visitatore, che in questo modo terrà aperte due finestre.

TITLE

Come per l'elemento IMG anche per i collegamenti ipertestuali è possibile definire un testo di commento, attivato quando il mouse passa sull'area di link. Questa la sintassi:

```
<A HREF="http://www.html.it" TITLE="Il sito italiano sul Web publishing">Visita HTML.it</A>
```

MAILTO (link a e-mail)

E' possibile inserire collegamenti anche verso indirizzi e-mail, attraverso una sintassi leggermente diversa da quella indicata per le URL. Questo il codice:

```
<A HREF="mailto:webmaster@html.it">Scrivi a HTML.it</A>
```

Cliccando su questo link **viene automaticamente aperto il tuo programma di posta predefinito con il campo TO già impostato** su webmaster@html.it.

Link interni al documento

Finora abbiamo analizzato link a risorse esterne o altre pagine di uno stesso sito. E' possibile creare collegamenti con punti specifici di uno stesso documento, grazie al codice ``. In altri termini se con i collegamenti visti finora viene linkato un documento, con `<A NAME>` viene linkato un punto specifico del documento.

Seguono le procedure per creare collegamenti a punti interni del documento:

- Inserisci il tag `` nel punto del documento da raggiungere. `ancora` è il nome identificativo del punto in cui il browser dovrà visualizzare la pagina. Ovviamente puoi sostituirlo con altri termini.
- Nel collegamento dal quale vuoi raggiungere il punto del nuovo documento inserisci la seguente sintassi: `Visita HTML.it`

Il **cancelletto (#)** indica che si tratta di un link interno. L'esempio considerato rimanda ad un punto della stessa pagina, perchè se a destra del cancelletto c'è il nome del link interno, a sinistra non è scritto nulla. Se volessimo raggiungere un punto preciso di un documento esterno la sintassi corretta dovrebbe essere: `Visita HTML.it`. Dove `"nome_file.htm"` è il nome del documento da raggiungere, e `"ancora"` il punto preciso all'interno dello stesso.

<TABLE> gestire Tabelle

La larghezza e l'altezza complessiva di una tabella viene indicata all'interno dell'attributo <TABLE>, il quale appunto e' quello che deve aprire e chiudere una tabella. Le dimensioni vengono definite con WIDTH (larghezza) e HEIGHT (altezza) e possono essere espresse in pixel o in percentuale della pagina:

```
<TABLE WIDTH=500>  
</TABLE>
```

In questo esempio di codice la larghezza della tabella viene espressa in pixel, in questo caso 500. Se si sceglie questa opzione, a qualsiasi risoluzione venga vista la pagina, la misura della tabella rimarra' invariata, cioe' di 500 pixel.

Cosa diversa accade nel momento in cui si sceglie una grandezza in percentuali della pagina:

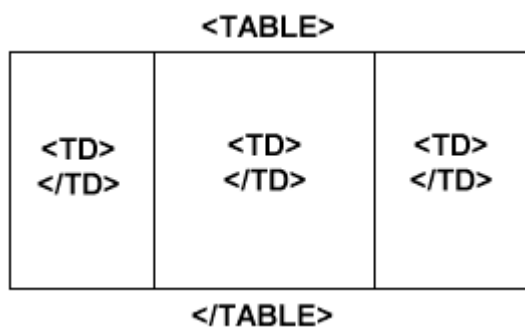
```
<TABLE WIDTH=50%>  
</TABLE>
```

In questo caso la larghezza della tabella sara' diversa a seconda della risoluzione video di colui, o colei, che ne esamini il contenuto.

Per esempio chi ha una risoluzione video di 640x480 vedra' un tabella di circa 320 pixels (perche' abbiamo impostato la tabella di una grandezza pari al 50% della pagina), mentre chi ha una risoluzione di 800x600 la vedra' di 400 pixel.

A mio avviso e' meglio mantenere il controllo assoluto sulla grandezza delle tabelle adottando una misurazione in pixel piuttosto che in percentuali.

Ora diamo uno sguardo al modo in cui deve strutturarsi una tabella. Per fare questo ci serviremo di alcune immagini:



Come si vede da questa immagine, l'attributo <TABLE> genera la tabella mentre con <TD> si definiscono i campi presenti all'interno di tale tabella. Ecco come il disegno puo' essere trasformato in codice e quindi in una vera e propria tabella:

```
<TABLE BORDER=1 WIDTH=300>  
<TD WIDTH=100>prova  
</TD>  
<TD WIDTH=100>prova  
</TD>  
<TD WIDTH=100>prova  
</TD>  
</TABLE>
```

Il risultato sarà il seguente:

prova	prova	prova
--------------	--------------	--------------

Consideriamo nel prossimo esempio la necessita' di generare una tabella di questo tipo:



Come si nota abbiamo introdotto un nuovo comando <TR> che e' una sorta di "a capo" all'interno della tabella. Ecco come il disegno puo' essere trasformato in codice e quindi in una vera e propria tabella:

```
<TABLE BORDER=1 WIDTH=300>
```

```
<TR>
```

```
<TD WIDTH=100>prova
```

```
</TD>
```

```
<TD WIDTH=100>prova
```

```
</TD>
```

```
<TD WIDTH=100>prova
```

```
</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD WIDTH=100>prova
```

```
</TD>
```

```
<TD WIDTH=100>prova
```

```
</TD>
```

```
<TD WIDTH=100>prova
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

prova	prova	prova
prova	prova	prova

Lo spazio tra i vari campi di una tabella e' definito all'interno del comando </TABLE> con CELLSPACING=X e CELLPADDING=X (dove X e' la distanza in pixel):

```
<TABLE CELLPADDING=10 CELLSPACING=10>
```

```
</TABLE>
```

La posizione del testo o delle immagini all'interno dei vari campi <TD> della tabella puo' essere modificata in diversi modi:

```

<TABLE WIDTH=300 HEIGHT=200>
<TD width=100 VALIGN=TOP>prova
prova </TD>
<TD WIDTH=100 VALIGN=BOTTOM>prova
prova </TD>
<TD WIDTH=100 VALIGN=MIDDLE>prova
prova </TD>
</TABLE>

```

La tabella ottenuta sarà la seguente :

prova		prova
	prova	

```

<TABLE WIDTH=300 HEIGHT=200>
<TD width=100 ALIGN=RIGHT>
prova </TD>
<TD WIDTH=100 ALIGN=CENTER>
Prova </TD>
<TD WIDTH=100 ALIGN=LEFT>
Prova </TD>
</TABLE>

```

In questo modo il testo viene allineato all'interno delle celle

prova	prova	prova
--------------	--------------	--------------

Ogni campo puo' avere un colore di background diverso dagli altri ed addirittura uno sfondo come delle normali pagine web (gli sfondi possono essere sostituiti da GIF animate):

```

<TABLE WIDTH=300 HEIGHT=200>
<TD width=100 BGCOLOR="red">
prova </TD>
<TD WIDTH=100 BGCOLOR="yellow">
prova </TD>
<TD WIDTH=100 BGCOLOR="gray">
prova </TD>
</TABLE>

```

L'effetto ottenuto sarà il seguente:

prova	prova	prova
--------------	--------------	--------------

Fogli di stile

HTML soffre di limiti propri di un sistema di contrassegno ideato per scopi molto lontani da quelli attualmente richiesti dal Web design. Se questi limiti appaiono marginali agli occhi dei neofiti, risultano fastidiosi, e molto spesso immobilizzanti, per i professionisti formati nella grafica tradizionale. Posizionare un'immagine, creare una banda laterale, giustificare del testo in HTML diventa un problema risolvibile esclusivamente con strumenti nati per tutt'altro scopo (le tabelle, per esempio, nel 90% dei casi vengono utilizzate per posizionare elementi nella pagina, invece che per ordinare dati).

Il problema, in termini minimamente più tecnici, riguarda la classica separazione SGML tra stile, contenuti e struttura. HTML "tradizionale" non soddisfa questa condizione, ma anzi, ne esaspera la confusione.

Ma cosa significa separare lo stile dai contenuti? Consideriamo un sito Web mediamente complesso, con un numero di pagine HTML pari a 100. Poniamo che il testo adottato per l'esposizione degli argomenti sia un "arial" corsivo. Con gli strumenti classici di HTML il codice per ottenere questo risultato è:

```
<FONT FACE="ARIAL"><I>Testo della pagina</I></FONT>
```

ripetuto in tutte le 100 pagine del sito, a chiusura e apertura del testo da formattare. Semplice, ovvio e per molti versi banale. **Ma cosa succede nel momento in cui scegliamo di modificare il tipo di carattere a tutte le pagine?** Non c'è altra soluzione che aprire le 100 pagine e procedere ad altrettante modifiche, che sostituiscano il nome "arial" con il nuovo font scelto. Un webmaster di medie capacità può impiegare solo qualche decina di minuti, che diventano ore per siti di grandi dimensioni. Bene, questa perdita di tempo è diretta conseguenza della promiscuità tra stile e contenuto, laddove il primo (il tag FONT) non è separato dal secondo (il testo della pagina).

Viene da sé che **la soluzione a questo problema** è nella separazione tra i due elementi sopracitati, che nella pratica **si risolve adottando i Cascading Style Sheets**. D'ora in poi ci serviremo dell'acronimo **CSS** per richiamare questi "fogli di stile a cascata" che da tempo sono stati introdotti nel Web publishing, ma che solo negli ultimi mesi hanno conosciuto una grande diffusione.

Il termine "a cascata" (cascading) richiama una delle caratteristiche principali di questa tecnologia, per cui è possibile incorporare nel documento differenti fogli di stile, ognuno dei quali, in base a regole gerarchiche, prevale sull'altro.

I CSS sono stati introdotti da Microsoft dalla terza versione di Internet Explorer, e parzialmente supportati da Netscape soltanto dalla quarta versione dell'omonimo browser. Chi accede con un browser obsoleto ad un documento formattato con fogli CSS, si troverà di fronte una pagina gestita dalle opzioni di default del browser (lo sfondo, per esempio, sarà grigio; il tipo di testo "times new romans" ecc).

I CSS sono stati ufficialmente riconosciuti e standardizzati dal W3C (consorzio internazionale per lo sviluppo del Web) nelle raccomandazioni "CSS1" prima, e "CSS2" poi. Si rimanda a questi documenti ufficiali per ulteriori approfondimenti.

L'argomento dei fogli di stile è abbastanza complesso da richiedere una trattazione molto più ampia di quella che si potrebbe fornire in questa sede.

<FRAME> Gestire Frame

Amati e odiati, utili o inutili, benefatti o pessimi **i frames sono strumenti entrati ormai nella consuetudine del web, e perfettamente gestiti da tutti i browser**. I detrattori dei frames affermano l'inutilita' di suddividere ulteriormente le pagine web; pagine che in ultima analisi possono risultare poco leggibili. Altri, come me, pensano che i frames possano rendersi utilissimi evitando di caricare le stesse immagini e mantenendo ordinata la struttura e i contenuti del sito. Naturalmente abusare dei frames puo' portare a traguardi di pessima impostazione grafica, giungendo ad un risultato opposto rispetto a quello preventivato.

Un punto certamente a sfavore dei frames e' la loro incompatibilita' con i programmi di navigazione dei non vedenti che davanti a tali strumenti si bloccano, non permettendo la lettura delle pagine.

Una buona soluzione e' quella di creare una versione frame ed una versione senza frame.

Per creare una pagina divisa in frame e' necessario creare piu' files HTML richiamati da un file principale che e' poi quello che ne permette la gestione. Prima di tutto quindi bisogna impostare questo file "sorgente", e poi successivamente gli altri files che compongono il frame.



Consideriamo di dover creare una finestra divisa in frame come da figura, con un frame in alto fisso (nel quale caricheremo il file "top.htm" da creare a parte) e un frame centrale (nel quale caricheremo il file "central.htm" da creare a parte) il quale cambi a seconda della pagina che si dovra' visualizzare. Come scritto sopra questi due frames devono essere gestiti da un terzo files il quale dovra' richiamarli assegnando loro una parte della pagina. Ecco il codice di questa pagina:

```
<FRAMESET rows="80,*">  
<frame name="alto" src="top.htm">  
<frame name="centrale" src="central.htm">  
</FRAMESET>
```

Come si vede il codice del frame e' racchiuso tra i comandi `<FRAMESET></FRAMESET>` che si comportano come gli usuali comandi `<HTML></HTML>`.

La grandezza dei frame (o meglio lo spazio che ognuno di essi deve occupare della pagina) e' stabilita dal comando `rows="80,*"`, che sta a significare che il frame alto (che in questo caso e' una riga, "rows") deve essere di 80 pixels, mentre quel "*" sta a significare che tutto il resto deve essere assegnato al frame centrale. Avremmo potuto esprimere anche in percentuali le grandezze dei frames, in questo modo:

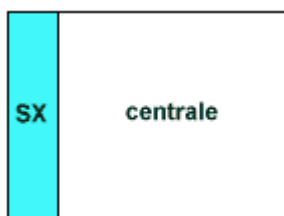
```
<FRAMESET rows="20%,*">
```

Impostati i due parametri `<FRAMESET></FRAMESET>` all'interno di essi si definiscono i nomi e i files da richiamare nei due frames creati. E' necessario dare un nome ai frames (`name="alto"`) e indicare il file HTML che dentro al frame dovra' essere caricato (`SRC="top.htm"`). Si devono quindi previamente creare due files di nome "top.htm" e "central.htm".

Nota come sia importante la posizione all'interno del codice per la corretta interpretazione da parte del browser. Se infatti si invertono gli ordini in questo modo:

```
<FRAMESET rows="80,*">  
<frame name="centrale" src="central.htm">  
<frame name="alto" src="top.htm">  
</FRAMESET>
```

Il browser invertira' l'ordine di assegnazione e carichera' il file "central.htm" nel frame superiore e il file "top.htm" nel frame centrale.



Per creare due frame verticali e' sufficiente sostituire a "rows" il termine "cols":

```
<FRAMESET cols="100,*">  
<frame name="sx" src="sx.htm">  
<frame name="centrale" src="central.htm">  
</FRAMESET>
```

I vecchi browsers non supportavano i frames per cui, nella prospettiva che le pagine vengano visualizzate da uno di questi "vecchi" software, e' utile inserire del codice che avverta della presenza di frames e della impossibilita' da parte del browser di visualizzarle. Ecco il codice da inserire:

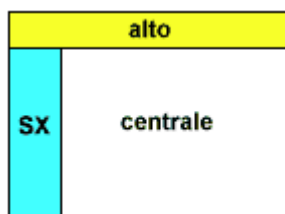
```
<noframe>
<HTML>
<body>
```

Attenzione. Il tuo browser non supporta l'opzione dei frame. Per visualizzare queste pagine e' necessario scaricare un browser che supporti tali opzioni. Ti consiglio Netscape 3.0 o superiore.

```
</body>
</html>
</noframe>
```

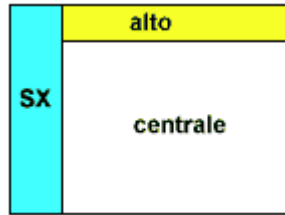
E' possibile adottare contemporaneamente una divisione sia in colonne (cols) che in righe (rows), in modo tale da creare una finestra divisa in piu' frames. Vediamo come si deve operare sul codice HTML del documento a seconda del numero e della posizione dei frames che si intendono creare.

Esempio 1



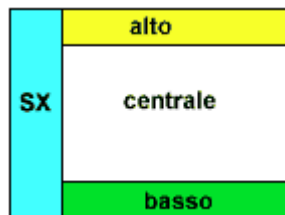
```
<frameset rows="100,*">
  <frame name="alto" src="top.htm">
    <frameset cols="150,*">
      <frame name="sx" src="sx.htm.htm">
      <frame name="centrale" src="central.htm">
    </frameset>
  </frameset>
```

Esempio 2



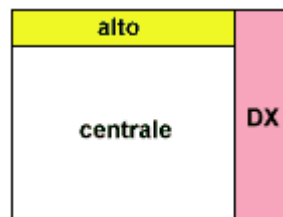
```
<frameset cols="120,*">
  <frame name="sx" src="sx.htm">
    <frameset rows="100,*">
      <frame name="alto" src="top.htm">
      <frame name="centrale" src="central.htm">
    </frameset>
  </frameset>
```

Esempio 3



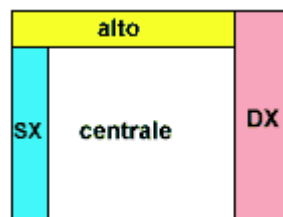
```
<frameset cols="120,*">
  <frame name="sx" src="sx.htm">
    <frameset rows="20%,60%,20%,*">
      <frame name="alto" src="top.htm">
      <frame name="centrale" src="central.htm">
      <frame name="basso" src="basso.htm">
    </frameset>
  </frameset>
```

Esempio 4



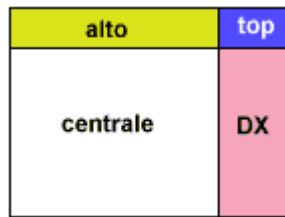
```
<frameset cols="75%,25%">
  <frameset rows="20%,80%*">
    <frame name="alto" src="top.htm">
    <frame name="centrale" src="central.htm">
  </frameset>
  <frame name="dx" src="dx.htm">
</frameset>
```

Esempio 5



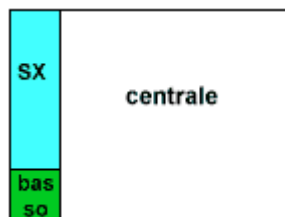
```
<frameset cols="75%,25%">
  <frameset rows="20%,80%*">
    <frame name="alto" src="top.htm">
    <frameset cols="20%,80%*">
      <frame name="sx" src="sx.htm">
      <frame name="centrale" src="central.htm">
    </frameset>
  </frameset>
  <frame name="dx" src="dx.htm">
</frameset>
```

Esempio 6



```
<frameset cols="75%,25%">
  <frameset rows="20%,80%*">
    <frame name="alto" src="top.htm">
    <frame name="centrale" src="central.htm">
  </frameset>
  <frameset rows="24%,76%">
    <frame name="top" src="top2.htm">
    <frame name="dx" src="dx.htm">
  </frameset>
</frameset>
```

Esempio 7



```
<frameset cols="25%,75%">
  <frameset rows="80%,20%">
    <frame name="alto" src="top.htm">
    <frame name="basso" src="basso.htm">
  </frameset>
  <frame name="centrale" src="central.htm">
</frameset>
```

Esempio 8



```
<frameset rows="20%,60%,20%">  
  <frame name="alto" src="top.htm">  
  <frame name="centrale" src="central.htm">  
  <frame name="basso" src="basso.htm">  
</frameset>
```

Esempio 9



```
<frameset cols="20%,60%,20%">  
  <frame name="sx" src="sx.htm">  
  <frame name="centrale" src="central.htm">  
  <frame name="dx" src="dx.htm">  
</frameset>
```

Per eliminare il bordo grigio dei frames si deve inserire il seguente codice:

```
<frameset cols="20%,60%,20%" border=0>
```

Per impedire che i frames vengano ridimensionati dal visitatore:

```
<frame name="alto" src="top.htm" noresize>
```

Per eliminare sempre le barre di scorrimento (scrollbars):

```
<frame name="alto" src="top.htm" scrolling="no">
```

Per renderle sempre visibili:

```
<frame name="alto" src="top.htm" scrolling="yes">
```

Per renderle visibili solo quando servono:

```
<frame name="alto" src="top.htm" scrolling="auto">
```

Per regolare la distanza dal contenuto del frame dal margine alto (marginheight) e sinistro e destro (marginwidth):

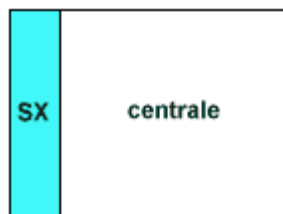
```
<frame name="alto" src="top.htm" marginheight=2 marginwidth=5>
```

Per quanto riguarda **i links all'interno dei frames** (cioe' come caricare una pagina in una frame diverso da quello in cui si trova il link) si deve far riferimento al nome che in fase di realizzazione abbiamo assegnato ai vari frames. Nome che non si riferisce al file ma a quanto scritto dopo "name=". In questo caso, per esempio:

```
<frame name="alto" src="top.htm">
```

il nome assegnato e' "alto".

Consideriamo la seguente pagina divisa in frames:



```
<frameset cols="20%,60%,20%">  
  <frame name="sx" src="sx.htm">  
  <frame name="centrale" src="central.htm">  
</frameset>
```

Consideriamo che da un link presente su "SX", dobbiamo caricare un'altra pagina sul frame "Centrale".

Se il link, presente sul frame "SX", fosse semplicemente:

```
<A HREF="nuova.htm">Clicca</A>
```

la pagina verrebbe caricata all'interno dello stesso frame (cioe' "SX"), perche' in assenza di comandi adatti il browser capisce di dover caricare la nuova pagina nello stesso frame in cui e' presente il link.

L'esatto codice e':

```
<A HREF="nuova.htm" TARGET="centrale">Clicca</A>
```

Altro uso fondamentale del comando <TARGET> e' quello di richiamare un link ad un'altra pagina la quale verra' visualizzata a pieno schermo, eliminando tutti i frame preesistenti.

Ecco il codice:

```
<A HREF="nuova.htm" TARGET="_parent">Clicca</A>
```

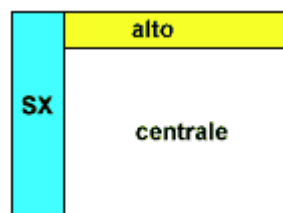
Se inserisci il codice:

```
<base target="_top">
```

in testa al documento HTML tutti i link presenti nella pagine elimineranno i frame esistenti, senza necessita' di inserirli in ogni link.

Un'altra necessita' puo' essere quella di caricare, con un solo click, due o piu' frames (naturalmente la finestra deve essere divisa in almeno tre frames).

Consideriamo una pagina suddivisa come nella figura:



```
<frameset cols="120,*">  
  <frame name="sx" src="sx.htm">  
<frameset rows="100,*">  
  <frame name="alto2" src="top.htm">  
  <frame name="centrale3" src="central.htm">  
</frameset>  
</frameset>
```

La nostra necessita' e' di inserire un solo link nel frame di "SX" che carichi contemporaneamente due pagine diverse nei due frames di destra: "alto" e "centrale". Per fare questo e' necessario inserire qualche riga di codice javascript. La prima parte del codice va inserita tra <HEAD></HEAD>:

```
<HEAD>
<script language="JavaScript">
<!-- Hiding
function loadtwo(page2, page3) {

parent.alto2.location.href=page2;
parent.centrale3.location.href=page3;
}
// -->
</script>
</HEAD>
```

Mentre la seconda parte va inserita tra <BODY></BODY> dove si vuole inserire il link:

```
<BODY>
<FORM NAME="buttons">

<INPUT TYPE="button" VALUE="Clicca"
onClick="loadtwo('nuovo1.htm','nuovo2.htm')">
</FORM>
</BODY>
```