

HTML

Ing. Marius Spinu
2000

1. INTRODUZIONE ALL'HTML	3
1.1. HTML	3
1.2. BROWSER	3
1.3. URL	3
1.4. TERMINOLOGIA	4
2. LA STRUTTURA DI UNA PAGINA	6
2.1. ELEMENTI DELLA PAGINA: I "TAG" <HTML>, <HEAD> E <BODY>	6
2.2. COMPONENTI ELEMENTARI DELLA PAGINA HTML: TESTO, IMMAGINI E LINK	9
2.3. LA GESTIONE DELLO SPAZIO, E &NBSP	9
3. PRIMO ELENCO DI TAG	11
3.1. COLLEGAMENTI DINAMICI (LINK)	11
3.2. LA FORMATTAZIONE DEL TESTO	14
3.3. L'INSERIMENTO DELLE IMMAGINI	17
3.4. LA GESTIONE DELLE LISTE	18
3.5. ELEMENTI DINAMICI	20
4. FORMATTAZIONE AVANZATA ED USO DI TABELLE	24
4.1. MARGINI, RIENTRI E ALLINEAMENTO	24
4.2. LA GESTIONE DELLE TABELLE	24
5. MAPPE SENSIBILI	28
6. I FRAME	31
6.1. UTILIZZO, NAVIGAZIONE E FORMATTAZIONE DI FRAME	31
7. FORM ED INTERAZIONI CON IL SERVER	37
7.1. CONTROLLI, PULSANTI E CAMPI NASCOSTI	37
7.2. LETTURA DELLE FORM.	38
7.3. CENNI A CGI E ASP	39
7.4. CENNI A JAVASCRIPT	40
8. APPENDIX	41

1. Introduzione all'HTML

1.1. HTML

HTML è un tipo di linguaggio SGML (Standard Generalized Markup Language) che può essere visualizzato da tutti i tipi di computer.

Il significato del nome HTML è HyperText Markup Language.

Con il termine HyperText (in italiano ipertesto) si intende un particolare documento che contiene elementi che permettono il passaggio ad altri documenti o parti di essi che possono essere interni o esterni al ipertesto originale.

Gli elementi che permettono il passaggio da un documento ad un altro sono evidenziati nel documento originale e sono individuati con il termine *link*.

Il Markup Language deriva da applicazioni editoriali dove il termine *mark* è utilizzato per caratteristiche particolari del testo.

I file HTML hanno estensione **htm** (per DOS) o **html** per gli altri sistemi operativi.

Come nel caso di altri file di codice sono file in formato testo e si possono editare con comuni editor testuali (senza formattazione). Si trovano in Internet o nei negozi dei programmi chiamati editor html che permettono di creare pagine html in modo facilitato. Inoltre si possono utilizzare anche i convertitori html ed i validatori html.

Tra gli editor html si distinguono due categorie:

- Editor WYSIWYG che durante la realizzazione della pagina la presentano con l'aspetto finale. Hanno il vantaggio della velocità permettendo di valutare il risultato finale mentre si crea il documento.
- Editor senza l'interazione diretta sulla parte grafica che però presentano la pagina html utilizzando colori diversi per i tag e per gli attributi ed eseguono delle correzioni o offrono dei suggerimenti. Spesso permettono di visualizzare un'anteprima del documento. Hanno il vantaggio di poter utilizzare esattamente il tag o attributo voluto e rende la pagina più facile da modificare.

1.2. Browser

I browser sono applicazioni (dipendenti dal tipo di computer che si sta utilizzando) che convertono i file HTML in forma testuale o grafica e gli rappresentano sullo schermo.

I Browser più comuni al giorno d'oggi sono Internet Explorer della Microsoft e Netscape Navigator della Netscape Communication Corporation. Browser meno utilizzati sono Mosaic della NCSA (National Center for Supercomputing Applications dell'Università di Illinois) e Lynx (browser di testo sviluppato dall'Università di Kansas).

1.3. URL

Tutti i testi (ipertesti) che si trovano nella WWW (World Wide Web) trattano le cosiddette URL (Uniform Resource Locator) che indica l'indirizzo Internet dove si trova una determinata risorsa utilizzata nel documento. Un URL può ricondurre ad un documento che si trova in una zona diversa della pagina corrente o ad una pagina WEB che si trova da qualsiasi altra parte del WWW.

Esempi di URL sono *http://www.dsi.unifi.it* o *http://www.ibm.com*. La parte iniziale del URL precisa che si sta effettuando un trasferimento di ipertesto (http = hypertext transfer protocol).

Il protocollo indica al browser il tipo di risorsa che sta utilizzando e può essere: http, file, ftp, gopher, mailto, news o telnet.

Il formato di un URL è:

[protocollo] / nome del dominio / [path] / [nome del file]

Gli elementi tra [] sono opzionali in certe condizioni. Il protocollo è considerato come http se non indicato, il path può mancare se la pagina si trova direttamente nella directory virtuale del server ed il nome file può mancare se ci sono delle impostazioni sul server che cercano automaticamente nomi di file predefiniti (index.html o home.html).

Il dominio specifica la posizione fisica delle risorse dove si sta accedendo e consiste nel nome del computer che esegue il software del server che gestisce il protocollo precedentemente specificato. Come dominio si può utilizzare anche l'indirizzo IP (Internet Protocol) del computer. Ci sono dei programmi speciali chiamati DNS (Domain Name Server) che si occupano della conversione tra indirizzo numerico ed il nome del server.

Il path specifica la posizione del file da visualizzare rispetto alla directory virtuale del server.

Nel esempio: *http://www.dsi.unifi.it/icsm2001/icsm_index.html* si ha:

protocollo = http,
dominio = www.dsi.unifi.it,
path = icsm2001
file = icsm_index.html

1.4. Terminologia

Come normale per linguaggi specializzati anche HTML richiede la conoscenza di alcuni termini specifici.

TAG E' uno dei termini più utilizzati e consiste in codice HTML scritto tra parentesi triangolari (< e >); è utilizzato per la formattazione del contenuto della pagina. HTML usa due tipi di TAG: *vuoti* e *contenitori*. I TAG contenitori hanno un inizio e una fine e si riferiscono a elementi che si trovano al loro interno. Un esempio può essere il seguente testo:

In questa riga si ha una *parola* in corsivo

che scritta con un TAG per il testo si potrebbe presentare come:

In questa riga si ha una <inizio corsivo> parola <fine corsivo> in corsivo

Nel linguaggio html l'inizio del tag consiste semplicemente nel suo nome tra parentesi triangolari. La fine del tag è uguale all'inizio con in più il simbolo "/" dopo la parentesi aperta. Un esempio chiarisce meglio il concetto:

inizio del tag per testo corsivo: <i>
fine del tag per testo corsivo: </i>

oppure

<center> con il tag di chiusura: </center>

Durante l'evoluzione del HTML (siamo alla versione 4.0) i TAG hanno subito un continuo processo di rinnovamento, alcuni sono diventati obsoleti, alcuni nuovi sono stati introdotti e qualche volta è stato cambiato il loro significato.

Attributi Sono altre parti di codice HTML che si possono aggiungere al TAG (vuoto o contenitore) e permettono di modificare e/o personalizzare l'effetto del TAG stesso.

NOTA: Anche se normalmente i tag sono scritti con lettere maiuscole questo non è un obbligo per il creatore della pagina. Certi tools che facilitano la creazione delle pagine HTML (editor HTML) impostano automaticamente le maiuscole quando il tag viene riconosciuto.

2. La struttura di una pagina

2.1. Elementi della pagina: I "tag" <html>, <head> e <body>

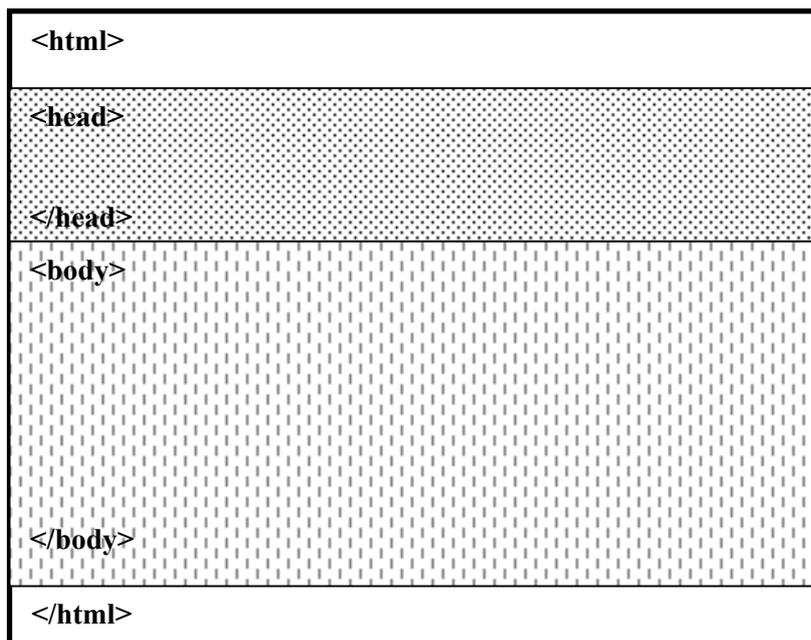
Elementi di base:

Tipo <HTML></HTML> (inizio e fine del file)

Testata <HEAD></HEAD> (informazioni descrittive; come il titolo)

Titolo <TITLE></TITLE> (titolo del documento; deve essere nella testata)

Corpo <BODY></BODY> (contenuto della pagina)



La pagina html ha normalmente due componenti: la testata ed il corpo

I tag <HTML></HTML> sono obbligatori e servono per specificare al browser il tipo di pagina che sta visualizzando.

Si consideri che i browser possono visualizzare anche file immagine di vario formato e in certi casi anche file diversi tramite apposite applicazioni chiamate plug-in.

I tag <HTML></HTML> devono essere utilizzati all'inizio e alla fine della pagina html (quindi i tag html sono tag contenitori che contengono l'intera pagina html).

NOTA: Il tag HTML normalmente identifica la prima e l'ultima riga di un documento html. Il tag di inizio può essere preceduto solo dal tag <!DOCTYPE> che è uno dei tag del SGML (sono stati adottati dai ISO per la descrizione dei documenti).

Gli attributi del tag HTML sono DIR e lang. DIR è utilizzato per specificare se il testo è scritto da sinistra a destra (LTR) o da destra a sinistra (RTL) mentre LANG è utilizzato per impostare una lingua. LANG non impedisce al browser di visualizzare caratteri o elementi non appartenenti alla lingua ma aiuta nell'organizzazione della pagina con le regole della lingua specificata.

Esempio: <HTML DIR="LTR" LANG="it">

oppure <HTML DIR="LTR" LANG="en-US">

Il tag **<head>** è un altro tag contenitore che definisce una parte di pagina che non è visualizzata ma che definisce delle caratteristiche della pagina come titolo **<title>**, indirizzo base del documento **<BASE>**, informazioni aggiuntive **<META>**, fogli di stile **<style>** ecc.

Il tag più utilizzato è il tag **<TITLE>** che permette di specificare il titolo della pagina. Tale titolo sarà visualizzato sulla barra superiore della finestra del browser. Il tag **<title>** ha come attributi **DIR** e **LANG** ed è obbligatorio nella pagina html ed è utilizzato anche per aggiunte di default nel bookmark (preferiti). Eventuali altri TAG inseriti nel tag **<Title>** saranno ignorati. Ovviamente la scelta del titolo è alla disposizione del webmaster ma sono da evitare nomi generici (“home page”) o titoli troppo lunghi (“questa home page è stata fatta per far vedere un sacco di cose belle”).

Esempio di utilizzo del tag **TITLE**:

```
<TITLE> La home page di Marius </TITLE>  
<TITLE> Foto dalla Groenlandia </TITLE>
```

Il tag **<BODY>** è un tag contenitore che delimita l’inizio della parte visualizzata del documento. Può contenere un numero qualsiasi di altri tag all’interno e dalla versione 3.0 di html ha una serie di attributi come: **alink**, **background**, **bgcolor**, **class**, **id**, **lang**, **link**, **style**, **text**, **title** e **vlink** oltre ad attributi come **onMouseOver**, **onMouseUp**, **onKeyPress** ecc.

MODIFICA DEI FILE HTML

Non basta scrivere del testo tra i tag **<BODY>** per rendere “leggibile” una pagina HTML. Per fare questo è necessario utilizzare tanti altri numerosi tag che permettano di formattare, allineare, modificare colori, inserire immagini o suoni ecc.

Come accennato in precedenza il documento deve essere salvato come file con estensione html o htm e successivamente aperto con un browser.

Le eventuali modifiche si possono fare da un qualsiasi editor di testo (notepad, wordpad o simili) e dopo averle salvate si deve utilizzare il pulsante “Aggiorna” del browser.

Il modo più veloce per lavorare è avere entrambe le applicazioni aperte sul desktop.

TAG META

I motori di ricerca rappresentano una risorsa indispensabile per chi cerca informazioni nella rete e figurare all’interno di tali motori di ricerca è fondamentale per chi crea pagine web e vuole maggiore visibilità.

Il primo passo da fare consiste nel segnalare le proprie pagine e questo si fa utilizzando le pagine dei motori di ricerca dove normalmente sono presenti dei link con la scritta “Add your site” o simile.

Il passo successivo è cercare di conquistare un posto in cima alle liste di siti presentati come risultato della ricerca (spesso, soprattutto quando i termini da ricercare sono di largo uso e di carattere generale, vengono presentate delle liste con migliaia di siti).

Periodicamente i siti visitano le milioni di pagine contenute nei loro database verificando le eventuali modifiche.

Molti motori di ricerca considerano prioritaria la presenza dei termini all’interno dei tag

```
<TITLE></TITLE>
```

, per cui se viene effettuata una ricerca inserendo la parola “viaggio”, e la pagina si chiama **<TITLE>viaggio in Internet</TITLE>** è probabile che risulti ai primi posti dei link presentati (anche se nel caso dell’esempio si tratta di una parola di largo uso e si potrebbero trovare un numero elevatissimo di siti di agenzie di viaggio). Probabilmente una ricerca su “viaggio in Internet” il sito potrebbe effettivamente risultare al primo posto.

Fondamentali nel intento di migliorare la posizione nelle liste elencate dai motori di ricerca 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 (in certi casi anche prima di leggere il contenuto dei tag TITLE).

Le "keywords" (parole chiave) sono i termini che, sinteticamente, descrivono il contenuto di una pagina web. Se, ad esempio, il sito si occupa della città di Firenze, le keywords potrebbero essere:

Uffizi
Ponte Vecchio
Palazzo Pitti
Piazzale Michelangelo
ecc.

Il META tag dovrà essere così impostato:

```
<META name="keywords" Content="Uffizi ponte vecchio piazzale  
michelangelo">
```

Un piccolo trucco consiste nell'inserire due o più volte le keywords usando sia le maiuscole che le minuscole, in più lingue (se il vostro sito è multilingua) facendo attenzione a non inserire termini troppo generici.

Il motore di ricerca considera il numero di volte in cui un termine è presente all'interno della pagina e delle keywords. È bene quindi non esagerare col numero di termini all'interno delle keywords.

Ci sono motori di ricerca che

Altri META tag riguardano l'autore della pagina web:

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

il titolo che apparirà alla fine della ricerca:

```
<META NAME="description" content="Il mio sito">
```

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

```
<META NAME="GENERATOR" content="Notepad">
```

Se per qualche ragione desiderate che una delle vostre pagine NON sia indicizzata nei motori di ricerca, dovete inserire il seguente META tag:

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

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

```
<META HTTP-EQUIV="Refresh" CONTENT="5; url=http://www.unifi.it">
```

Dove CONTENT="5" imposta il numero di secondi entro il quale la nuova pagina sarà caricata; mentre url=http://www.unifi.it è il file che verrà caricato.

2.2. Componenti elementari della pagina html: testo, immagini e link

Gli componenti principali della pagina HTML sono i testi le immagini ed i link. I testi sono necessari per rendere l'idea del sito e per comunicare con il visitatore. Ovviamente il testo della pagina è scritto nella lingua del autore anche se sono tantissime le pagine in lingua inglese da tutte le parti del mondo. Oltre al testo spesso sono presenti delle immagini inserite per necessità o per motivi estetici.

Sia il testo che le immagini possono essere utilizzati come collegamenti (link). Come già accennato i link non sono altro che porte verso altri documenti html.

Per inserire il testo nella pagina html basta scriverlo nel contenuto del tag <BODY>. Ogni formattazione inserita tramite l'editor di testo non verrà assolutamente considerata nella fase di visualizzazione del documento nel browser. Per poter fare delle modifiche nella presentazione del testo nella pagina si dovranno utilizzare opportuni tag e attributi.

A differenza del testo per l'inserimento delle immagini sarà necessario utilizzare da subito il tag. Si suppone che l'immagine sia già presente in formato gif o jpg nella stessa directory del documento. Si parla dei formati gif e jpeg perché sono supportati da tutti i browser; ci sono browser che supportano anche immagini PNG e alcuni che tramite plug-in visualizzano anche altri tipi di immagini (PDF).

Se il file che contiene l'immagine si chiamasse disegno.gif allora il codice che permette di inserire l'immagine nel documento sarebbe:

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

Oltre a immagini e testo nel file html si possono inserire componenti audio e video (tramite i così detti plug-in dei browser).

2.3. La gestione dello spazio,
 e

Un aspetto che spesso confonde gli utenti alle prime armi è lo spazio e il "a capo". Il browser sembra che non consideri assolutamente gli spazi e l'invio a capo. In fatti per il testo normale il browser visualizza solo uno spazio tra le parole mentre il "a capo" è inserito tramite il tag
.

Per esempio il testo:

```
I browser      più
      comuni sono      Netscape
Communicator      e Internet      Explorer.
```

Sarà visualizzato sulla stessa riga (almeno che non sia necessario separarlo per la larghezza della finestra) e con uno solo spazio tra le parole:

```
I browser più comuni sono Netscape Communicator e Internet Explorer.
```

L'utente può inserire spazi con il carattere speciale e righe nuove con il tag

Così il testo:

I browser più comuni sono Netscape
Communicator e Internet Explorer.

Sara visualizzato come:

I browser più comuni sono
Netscape Communicator
e Internet Explorer.

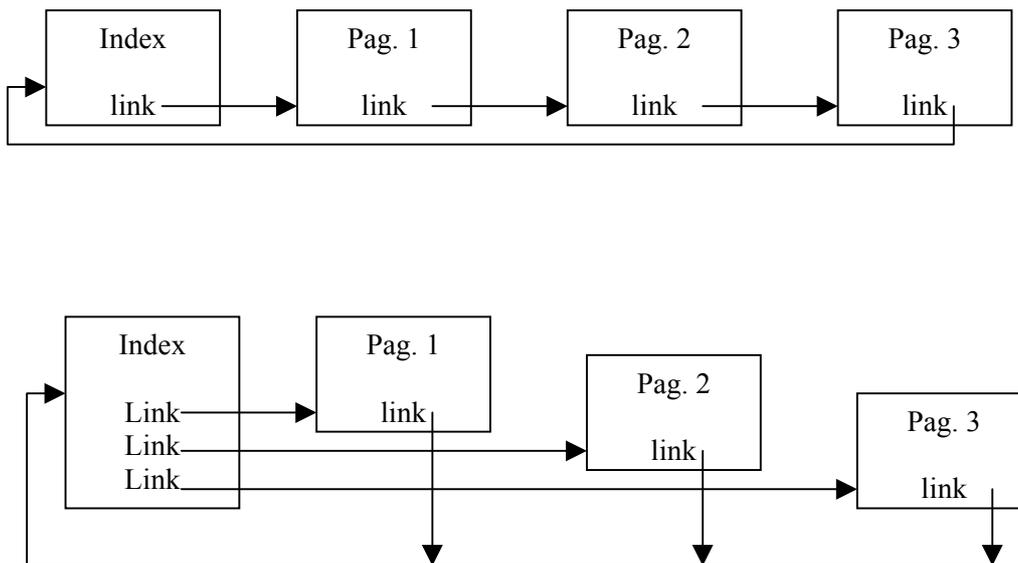
3. Primo elenco di TAG

3.1. Collegamenti dinamici (link)

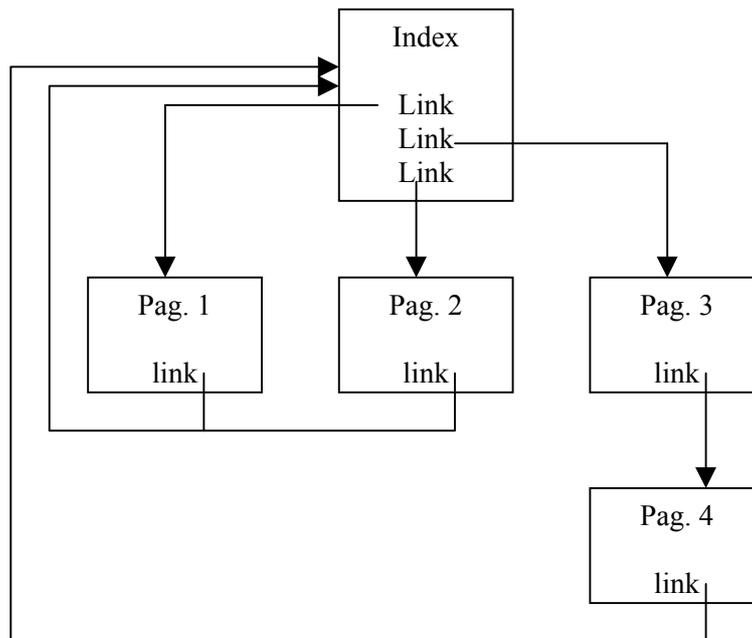
Come accennato durante l'introduzione al HTML un ipertesto è un documento particolare dove esistono varie porte verso altri documenti o parti di essi. Tali porte sono chiamate collegamenti (in inglese link) e si attivano tramite un semplice click del mouse. Le possibili azioni sono:

- Raggiungere un'altra parte del sito corrente
- Aprire un documento dello stesso sito web
- Aprire un documento di qualche altro sito.

Quindi HTML è un linguaggio che permette di realizzare pagine collegate in modo dinamico tramite i link. L'uso dei link è un fattore molto importante nella progettazione delle pagine web. Un sito povero di link è spesso anche non molto interessante per il visitatore ma un sito dove si esagera nell'utilizzo dei link è molto difficile da visitare e da valutare. Non è male progettare un diagramma (schema di link o storyboard) con i link che si vogliono utilizzare nel sito e valutare successivamente la sua complessità. Lo schema dei link si mostra particolarmente utile quando a lavorare per la realizzazione del sito ci sono più persone. Gli schemi di link fra pagine web possono essere sequenziali (fra pagine o indicizzato in una pagina), gerarchico o personalizzato.



Schemi di link sequenziali: fra pagine (sopra) e indicizzato in una pagina (sotto)



Schema di link gerarchico

Ovviamente il progettista può decidere di utilizzare qualsiasi tipo di schema e può decidere di crearne uno personalizzato (maggioranza dei casi). Uno schema personalizzato va gestito con più attenzione in quanto si deve verificare che tutte le pagine siano raggiungibili e che non si vada a finire in un loop senza uscita.

Il normale link HTML ha il seguente formato:

` testo evidenziato `

Il tag `<a>` è chiamato ancora, è un tag contenitore e crea un collegamento dal testo contenuto verso la destinazione specificata. Parte dei suoi possibili attributi sono: href, id, name, style. Inoltre ci sono attributi specifici come onMouseDown, onMouseOver, ecc. che permettono l'utilizzo nei Javascript.

L'**origine del link** è quella parte del documento dove il click del mouse genera l'inizio del collegamento. Normalmente nei browser l'origine del link è evidenziata (il colore di default è l'azzurro) e al passaggio del mouse il cursore cambia aspetto e diventa una mano.

La **destinazione del link** è la pagina (o parte di essa) che verrà visualizzata dopo il click del mouse. Nei browser la destinazione del link è visualizzata sulla barra di stato quando il cursore del mouse si trova sull'origine.

Il seguente link ha come origine il testo "Firenze" e come destinazione il sito del Comune di Firenze:

` Firenze `

Tramite il tag `<a>` si possono assegnare nomi a punti precisi del documento HTML. Supponendo di avere un documento contenente brevi descrizioni (10-20 righe) di 100 città italiane diventa difficile per il visitatore trovare la descrizione che lo interessa. Assegnando invece un nome (quello della città) al punto che si trova all'inizio del paragrafo relativo diventa possibile creare dei link che portano direttamente alla parte di interesse.

Il link nel esempio assegna al punto in qui viene inserito il nome "FI":

` Firenze `

Lo stesso risultato si ha utilizzando l'attributo ID al posto del NAME:

```
<a ID="FI"> Firenze </a>
```

L'attributo ID è stato introdotto dalla W3C nella versione 4.0 ed è diverso dal NAME perché può essere applicato in un numero maggiore di casi (anche per riferimento ad uno script o come identificativo di oggetto).

Il nome da assegnare può essere scelto dal webmaster con una qualche limitazione:

- Deve essere unico (non si possono utilizzare neanche le stesse parole con lettere maiuscole/minuscole – quindi se utilizzato FI non si può utilizzare ne fi ne Fi ne fI).
- Deve iniziare con una lettera dell'alfabeto
- Non deve contenere spazi

A differenza del ID l'attributo NAME può contenere anche caratteri speciali (β, À, Ê, Ð ecc.).

Se nel documento si ha una parte con un nome allora si può utilizzare il tag <a> per creare un link verso la questa zona:

```
<a href="#FI"> Città di Firenze </a>
```

Oppure si può creare un link che porti in una precisa zona di una pagina esterna:

```
<a href="pagina1.html#FI"> Città di Firenze </a>
```

E' conveniente utilizzare il tag <a> senza altri tag contenuti (nonostante sia un tag contenitore) perché su qualche browser si possono avere dei risultati imprevisti.

Gli esempi precedenti hanno fatto uso del attributo HREF (Hypertext REFerence) che provvede ad identificare il puntatore o percorso da seguire per la destinazione del link. Se la destinazione è una zona della pagina corrente allora href sarà seguito dal simbolo # (cancellato). Se la destinazione è una pagina sullo stesso sito allora href sarà seguito dal nome della pagina (con il relativo path) e se la destinazione fosse una pagina esterna al href segue il URL destinazione. In tutti i casi il testo che individua la destinazione deve essere tra virgolette doppie.

HREF = "#nome"	Area della stessa pagina
HREF = "[path] nome pagina"	Pagina sullo stesso sito
HREF = "URL"	Pagina esterna

Un caso particolare nell'utilizzo del href per destinare il link alla posta elettronica. Tramite l'utilizzo del protocollo **mailto** il tag <a> apre il software predefinito per la gestione della posta elettronica dove sarà creato un nuovo messaggio che ha come destinatario l'attributo del href.

Il codice

```
<a href="mailto:webmaster@questo.sito.com">webmaster</a>
```

ha come effetto l'apertura del software di gestione posta con il messaggio per webmaster@questo.sito.com.

3.2. La formattazione del testo

I paragrafi

I browser WWW, come noto, non tengono conto della formattazione del testo in HTML della pagina, ovvero non badano a spazi bianchi ridondanti, andate accapo, tabulazioni, ecc. Questo proprio perché la pagina deve poter essere vista con qualsiasi programma di comunicazione e in finestre di qualsiasi dimensione. Sarà dunque il browser stesso a riformattare la pagina.

Inoltre, il trasferimento del codice HTML avviene con 7 bit per carattere: questo significa che vengono trasmessi solo i primi 128 caratteri del codice ASCII. Per visualizzare gli altri caratteri, come lettere accentate, con diresi, ecc., si usano dei codici particolari, riportati in appendice.

Innanzitutto per andare accapo, come anticipato, l'istruzione è: **
**

Questa tag **impone** al browser di andare accapo: il testo proseguirà sulla riga successiva.

Per imporre ad una porzione di testo di non andare mai accapo, la si deve racchiudere fra **<NOBR>** e **</NOBR>**. Tutto il testo compreso fra tali tag, verrà posto su una sola linea. All'interno di un testo su una sola linea, può essere utilizzato sia il **
, che impone l'andata accapo, sia il nuovo comando **<WBR>; questo tag indica al browser che in quel punto, se necessario, *si può* andare accapo, ma non è una imposizione categorica.

Per cambiare paragrafo, invece, si usa il tag **<P>** con il tag di chiusura **</P>**.

A differenza del precedente, questo comando oltre a mandare accapo, lascia anche una riga di spazio.

Un altro tag interessante è **<HR>** che dà la possibilità di tracciare una riga orizzontale:

Il risultato è comodo, in quanto il browser dimensionerà automaticamente la riga sulla larghezza della finestra che si sta utilizzando. Questo tag ha qualche attributo, facente parte delle estensioni di Netscape, attributi che sono entrati anche nell'HTML 4. Essi sono:

SIZE=valore: specifica lo spessore della riga in pixel.

WIDTH=valore: specifica la larghezza in pixel o in percentuale (%) rispetto alla larghezza della finestra utilizzata.

NOSHADE : impone che la riga sia solida e non un solo effetto di ombra.

ALIGN="tipo": tipo di allineamento, che può essere LEFT, RIGHT, o CENTER.

Ad esempio l'istruzione:

```
<HR ALIGN="RIGHT" WIDTH="50%" SIZE=5>
```

crea una riga orizzontale che avrà sempre come lunghezza la metà della larghezza della finestra del browser anche se quest'ultima è ridimensionata.

Altro comando di formattazione dei paragrafi è **<BLOCKQUOTE>** ... **</BLOCKQUOTE>**. Tutto il testo compreso fra tali tag, verrà visualizzato leggermente rientrato ai lati; può essere utilizzato, ad esempio, per citazioni di parti di testi, per evidenziare delle parti, ecc.

Il testo

I comandi per la gestione degli attributi del testo hanno tutti la stessa forma: devono essere aperti prima della porzione di testo cui l'attributo si applica, e vanno chiusi alla fine della porzione stessa. Gli attributi da utilizzare riguardano la dimensione del testo, **il grassetto**, *il corsivo* ed alcune modalità speciali in grado di simulare, ad esempio, la resa di una macchina da scrivere.

Per esempio per rendere un testo in *corsivo* (od anche "italico" nell'accezione anglosassone) bisogna inserire il comando `<I>` prima del testo "corsivo" e poi il comando `</I>` subito dopo. Analogamente per tutti gli altri stili.

Quella che segue è una lista d'esempio di alcuni degli attributi disponibili:

`` Per un testo in **GRASSETTO** (o anche "bold") ``

`<I>` Per un testo in *CORSIVO* `</I>`

`` Per utilizzare uno stile più **MARCATO**``

`<PRE>`

Testo PREFORMATTO

(considera tutti gli spazi ed i ritorni a capo inseriti)

`</PRE>`

`<TT>` Stile da **MACCHINA DA SCRIVERE** `</TT>`

`<H6>` **INTESTAZIONE** (o "heading") di livello 6 `</H6>`

`<H5>` **INTESTAZIONE** di livello 5 `</H5>`

`<H4>` **INTESTAZIONE** di livello 4 `</H4>`

`<H3>` **INTESTAZIONE** di livello 3 `</H3>`

`<H2>` **INTESTAZIONE** di livello 2 `</H2>`

`<H1>` **INTESTAZIONE** di livello 1 `</H1>`

Si nota che alcuni comandi del tipo `<PRE>`, `<H1..6>` ed altri introducano necessariamente un ritorno a capo alla apertura e alla chiusura del comando stesso.

Se si vuole utilizzare una dimensione di testo variabile, senza mandare accapo il testo modificato si può inserire il seguente comando al posto del `<H...>`:

`` Dimensione 7 ``

`` Dimensione 6 ``

`` Dimensione 5 ``

`` Dimensione 4 ``

`` Dimensione 3 (Normale) ``

Dalla versione 3 dell'HTML è presente un nuovo tag del comando FONT che è "**FACE**": questo tag permette di ridefinire la fonte dei caratteri utilizzata. Fate attenzione perché il carattere specificato viene prelevato dal computer dell'utente, e se il carattere non è installato viene utilizzato il carattere di default.

Per chiarire l'esempio basti considerare questa scritta che dovrebbe essere visualizzata utilizzando il carattere "Arial" (che è normalmente presente in ogni Personal Computer dotato di Sistema Operativo Windows). Il comando utilizzato è il seguente:

``

Tramite il comando FONT è inoltre possibile modificare il **COLORE** delle scritte. Il tag da utilizzare è naturalmente "COLOR", che deve essere seguito dal numero esadecimale del colore da usare.

Ad esempio:

``Questa scritta è ROSSA``

Altri effetti sui caratteri sono ottenuti con i comandi:

<BLINK> (lampeggiante)
<TT> (telescrivente)
<BIG> (grande)
<SMALL> (piccolo)
<ADDRESS> (indirizzo)
<CODE> (codice)
<VAR> (variabile)
<KBD> (tastiera)
<SUP> (apice)
<SUB> (pedice)
<U> (sottolineato)
 (enfattizzato)
<CITE> (citazione)
<STRIKE> (barrato)
 (testo eliminato)
<INS> (testo inserito)
ecc.

Ultimamente la gestione del testo si sta orientando verso l'utilizzo di metodi diversi come gli stili. In questo caso la dimensione del font si gestisce con l'attributo font-size.

Analogamente gli attributi font-stretch, font-weight, font-variant, font-family e font permettono la gestione completa del font utilizzando la tecnica a stili.

FONT-SIZE

Dimensioni assolute:

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large

Dimensioni relative

- smaller – riduce di 20%
- larger – aumenta di 20%
- n% - rende la nuova dimensione n% della vecchia

FONT-FAMILY

Nome standard (W3C)

- serif
- sans-serif
- cursive
- fantasy
- monospace

Nome di font (funziona solo se il font esiste su tutte le macchine della rete)

- Times New Roman
- Arial
- ...

3.3. *L'inserimento delle immagini*

Qual è il miglior formato di immagine tra “gif” e “jpg”? E' difficile rispondere a questa domanda anche perché non vi è una risposta univoca che può dire la parola definitiva sulla bontà dell'una o dell'altra. Il problema principale di chi crea pagine web è quello di conciliare una decente velocità di caricamento e una grafica interessante.

In linea di massima si può affermare che per immagini di grandi dimensioni e con molti colori (una foto per esempio) sia opportuno usare il formato jpg; mentre per immagini di piccole dimensioni è meglio servirsi del formato gif.

Se consideriamo due piccole immagini di 70x70 pixel è possibile che in formato jpg l'immagine abbia 5Kb ed in formato gif (16 colori) occupi solamente 1,6Kb.

Le immagini rivestono altrettanta importanza del testo nel trasmettere un messaggio nell'interesse del suo significato. E' dunque possibile inserirne a volontà nei propri documenti per abbellirli esteticamente o per aumentarne la forza comunicativa. Naturalmente ciò deve essere fatto con discrezione, mantenendo chiaro in testa l'obiettivo da raggiungere: la pagina deve trasmettere informazioni, non solo attirare l'attenzione dell'osservatore.

Un'immagine inserita al punto giusto può essere molto più ricca di significato di tante immagini disordinate. Un ricco testo con una bella immagine può rappresentare una pagina accattivante capace di trasmettere interamente il messaggio del suo autore. Le immagini dunque rivestono un ruolo importante nella creazione di un documento HTML e bisogna imparare ad usarle con maestria.

L'HTML prevede un comando apposito per inserire un'immagine: **IMG**. Questo è un comando aperto, quindi non è necessario farlo seguire da nessun tag di chiusura.

IMG prevede vari parametri:

SRC per indicare l'immagine da inserire. E' chiaramente di fondamentale importanza e prevede di essere seguito dal nome del file dell'immagine.

LOWSRC per indicare un'altra immagine da caricare, prima della definitiva.

BORDER per indicare la dimensione dell'eventuale bordo che si vuole visualizzare intorno alla finestra.

ALIGN per indicare l'allineamento dell'immagine rispetto al testo: si potrebbe volere un allineamento a sinistra (LEFT) oppure a destra (RIGHT); oppure desiderare il testo centrato verticalmente rispetto all'immagine (CENTER).

ISMAP indica che l'immagine visualizzata è una mappa di coordinate: al click sopra il computer rileva le coordinate del mouse ed è in grado di inviarle al server che può elaborarle per individuare l'URL da attivare.

USEMAP per utilizzare, sull'immagine, una mappa locale.

WIDTH permette di specificare una larghezza predefinita per l'immagine, con un valore dato in punti.

HEIGHT analogamente al precedente imposta un'altezza predefinita.

VSPACE per indicare uno spazio da lasciare libero sopra e sotto l'immagine.

HSPACE per indicare uno spazio da lasciare libero a destra e a sinistra dell'immagine.

ALT permette di visualizzare un testo al posto dell'immagine se questa non è visualizzabile o quando il mouse passa sopra.

Se non viene specificato altrimenti tramite il desiderato allineamento, l'immagine viene inserita nel testo esattamente nel punto in cui appare il comando IMG, disponendo il testo ai suoi lati. Nel caso in cui il file dell'immagine non sia disponibile o ci sia un errore nel listato HTML, il browser può cercare di avvertire che c'è un'immagine non riconosciuta visualizzando un'immagine di default diversa a seconda del browser che si sta utilizzando.

Le immagini possono naturalmente essere inserite all'interno di comandi per utilizzarle come pulsanti di collegamento od icone. Il browser, se non altrimenti specificato, mette automaticamente un bordo intorno all'immagine per indicare che è un link. Se non lo si vuole si può usare il parametro BORDER=0 dentro IMG come:

```
<A HREF="link.htm">  
<IMG SRC="image1.gif" BORDER=0>  
</A>
```

3.4. *La gestione delle liste*

L'HTML prevede delle istruzioni apposite per inserire una lista di dati all'interno di un documento. Una lista è composta da una serie di paragrafi, ciascuno dei quali viene identificato dal comando e che possono essere elencati ordinati, non ordinati o visualizzati su colonne multiple.

Per specificare il tipo di lista che si vuole usare è necessario includere una delle seguenti istruzioni:

 ... per un elenco ordinato e numerato, tipicamente rientrato e con una spaziatura aggiuntiva tra i paragrafi.

 ... per un elenco non ordinato. Invece dei numeri le voci prevedono dei punti elenco.

<MENU> ... </MENU> per un elenco di brevi voci. In genere occupa una riga e viene visualizzato in modo più compatto dell'UL.

<DIR> ... </DIR> per visualizzare un tipico elenco di nomi di file. In genere vengono visualizzati su colonne multiple.

Per inserire una lista è allora sufficiente specificare il tipo di lista richiesta (OL, UL, MENU o DIR) e poi inserire ogni elemento tramite il comando .

Consideriamo il seguente script HTML e quanto viene visualizzato da un browser:

```
<HTML>  
<HEAD> <TITLE>Prova di una lista ordinata </TITLE></HEAD>  
<BODY>  
<H1>Indice di un libro HTML</H1>  
<H3>  
<OL>  
<LI> Introduzione all'HTML  
<LI> Terminologia  
<UL>  
<LI> Elenco dei tag  
<LI> Formattazione  
</UL>  
<LI> Esempio di un documento
```

```
</OL>
</BODY>
</HTML>
```

Indice di un libro HTML

1. Introduzione all'HTML
2. Terminologia
 - Elenco dei tag
 - Formattazione
3. Esempio di un documento

Come si nota dal listato qui sono state usate due liste diverse nidificate: una lista OL esterna in cui si inserisce una lista UL più interna. Il browser indenta automaticamente la lista più interna. Questa non è numerata e dunque i suoi elementi sono evidenziati da un quadratino. La lista esterna è invece una OL ordinata e quindi i suoi elementi sono numerati progressivamente. Si può provare a vedere cosa succede se si aggiungono elementi alla lista OL o a quella UL.

Si noti che, anche se esiste un identificatore di fine elemento (``) esso è diventato obsoleto e non importa utilizzarlo. Invece devono essere chiusi i formati OL, UL, MENU e DIR.

E' anche possibile cambiare le "intestazioni" degli elementi delle liste. Per le liste ordinate, esiste l'attributo "**TYPE**" che serve per indicare il tipo di indicizzazione che si vuole avere sulla lista.

Ad esempio, il comando `<OL TYPE="a">`, creerà una lista ordinata, utilizzando le lettere minuscole come indice. Gli altri valori possibili sono: "A" per usare lettere maiuscole; "I" per usare la numerazione romana maiuscola; "i" per i numeri romani minuscoli; "1" (default) per utilizzare i numeri.

Per le liste non ordinate, invece, i valori possibili sono "**DISC**" per dei cerchi pieni; "**CIRCLE**" per dei cerchi vuoti; "**SQUARE**" per dei quadratini (anche se, ripetiamo, non è detto che la resa sia esattamente quella voluta).

Per le liste ordinate, infine, esiste un ulteriore attributo, "**START**", per indicare il punto da cui far partire la numerazione: ad esempio `<OL TYPE="A" START="5">` farà partire la numerazione dalla lettera "E". Naturalmente tale attributo non ha alcuna funzionalità nell'ambito delle liste non ordinate.

Esiste un secondo tipo di lista chiamato **elenco di definizione**, o glossario. Si tratta di un elenco di paragrafi, ognuno dei quali è dotato di una breve intestazione e da un contenuto. E' utile per includere un insieme di nomi, ognuno accompagnato dalla sua definizione.

Un elenco di definizione è chiuso tra le istruzioni `<DL>` e `</DL>`. Ogni voce di definizione è composta di due parti: `<DT>` per l'intestazione, e `<DD>` per la spiegazione.

Per verificarne subito la resa si può utilizzare il seguente codice:

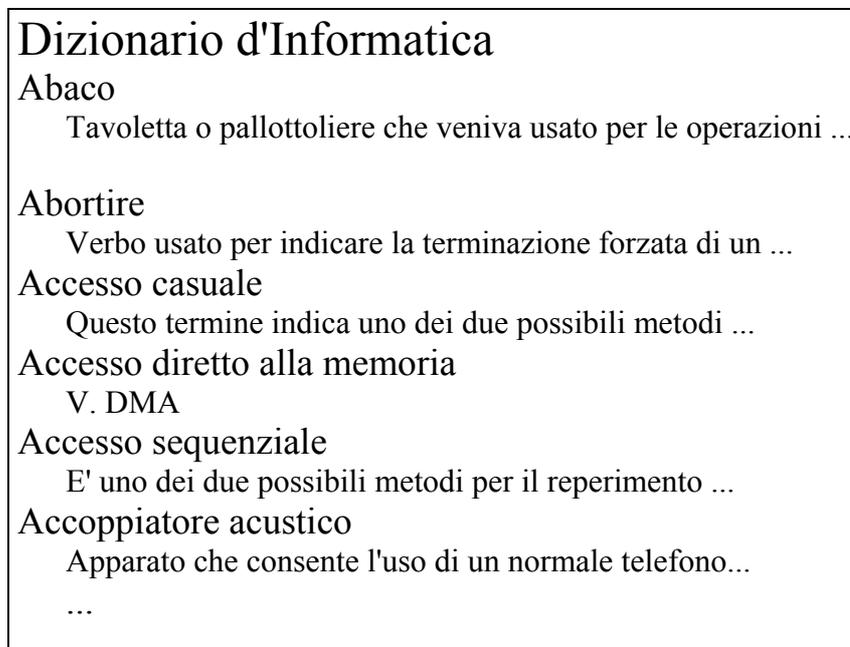
```
<HTML>
<HEAD> <TITLE> Dizionario d'Informatica </TITLE></HEAD>
<BODY>
<FONT SIZE=5> Dizionario d'Informatica</FONT><BR>
<HR>
<DL>
<DT><FONT SIZE=4>Abaco </FONT><DD>Tavoletta o pallottoliere che veniva
usato per le operazioni ...
```

```

<DT><FONT SIZE=4>Abortire </FONT><DD>Verbo usato per indicare la
terminazione forzata di un ...
<DT><FONT SIZE=4>Accesso casuale </FONT><DD>Questo termine indica uno dei
due possibili metodi ...
<DT><FONT SIZE=4>Accesso diretto alla memoria </FONT><DD>V. DMA
<DT><FONT SIZE=4>Accesso sequenziale </FONT><DD>E' uno dei due possibili
metodi per il reperimento ...
<DT><FONT SIZE=4>Accoppiatore acustico </FONT><DD>Apparato che consente
l'uso di un normale telefono...
<DT> <DD><FONT SIZE=4>...</FONT>
</DL>
</BODY>
</HTML>

```

Con il seguente risultato nella finestra del browser:



Come ultima nota si riporta un attributo del comando `<DL>` che permette di ridurre lo spazio bianco tra le righe e le indentazioni negli elenchi di definizioni: `<DL COMPACT>`.

3.5. Elementi dinamici

STILI

Uno stile è un insieme di regole di formattazione che indicano al browser come si vuole rappresentare la pagina. Per esempio può definire il carattere, l'allineamento, il colore del testo, il colore dei link ecc. Lo stile si può utilizzare direttamente nella pagina html tramite il tag `<STYLE>` o si possono utilizzare dei file esterni (fogli stile CSS – *cascading style sheet*).

Il tag `STYLE` si utilizza con il seguente formato:

```
<TAG STYLE=" Proprietà: valore; [Proprietà: valore;] " >
```

i due punti (:) separano la proprietà dal valore ed il punto e virgola (;) serve a separare una coppia proprietà/valore dalla (eventuale) seguente.

Esempio di utilizzo inline:

```
<h1 style="color:#AA0000"> titolo 1 </h1>  
<p style=" font-size: 30pt">
```

Come descritto in precedenza il tag STYLE può essere contenuto anche nella testata del documento (quindi contenuto nel tag <HEAD>). In questo caso la sintassi cambia:

Nel esempio seguente l'autore del sito decide di modificare l'aspetto del testo inserito come titolo. Tale modifica può essere fatta tutte le volte che si va a scrivere il testo oppure utilizzando il tag STYLE si può decidere di modificare dei tag classici html per la formattazione del testo. In particolare in questo caso si è deciso di modificare il font utilizzato per i titoli a livello 1 e al livello 2 cambiandolo in sans-serif e successivamente si sono assegnate altezze diverse ai due titoli.

```
<HEAD>  
<TITLE> titolo </title>  
<STYLE>  
<!--  
H1, H2  
{  
    font-family: sans-serif ;  
}  
  
H1  
{  
    font-size: 32pt;  
}  
H2  
{  
    font-size: 24pt;  
}  
-- >  
</STYLE>  
</HEAD>
```

Nel caso in cui si devono scrivere più pagine html (sito aziendale) e si vuole utilizzare per tutte lo stesso stile diventa necessario copiare nelle testate di tutte le pagine il pezzo di codice descritto in precedenza. Un'alternativa a questa soluzione consiste nell'utilizzo di file esterni per la definizione dei stili. Questi sono file ASCII salvati con estensione CSS.

Per il codice descritto il file CSS si presenterebbe così:

```
/* Foglio di stile per il sito aziendale */  
  
H1, H2  
{  
    font-family: sans-serif ;  
}
```

```
H1
{
    font-size: 32pt;
}
H2
{
    font-size: 24pt;
}
```

Il testo tra /* e */ è un commento come risulta familiare per i programmatori in C.

Per richiamare il file CSS nel documento si utilizza il tag LINK:

```
<HEAD>
<TITLE>TITOLO</TITLE>
<LINK rel="stylesheet" href="file.css" type="text/css">
</HEAD>
```

MULTIMEDIA

Per inserire un oggetto multimediale all'interno stesso della pagina, esiste un comando molto versatile:

"EMBED". Con questo comando si può inserire qualsiasi cosa, alla stregua di una semplice immagine. Gli attributi del comando sono:

SRC: indica cosa deve essere caricato (può essere un .mid, .wav, .avi, ecc.).

ALIGN: solito allineamento rispetto al testo.

BORDER: se si vuole mettere un bordo.

WIDTH: ne impone la larghezza.

HEIGHT: impone l'altezza.

AUTOSTART: specifica se l'oggetto multimediale deve partire automaticamente non appena caricato (se si ="TRUE", altrimenti ="FALSE").

LOOP: specifica in numero di volte che l'oggetto deve girare.

VOLUME: specifica il volume audio (valori da 0 a 100).

HSPACE: spazio libero ai lati dell'oggetto.

VSPACE: spazio libero sopra e sotto l'oggetto.

HIDDEN: usato solo per i file audio, indica se deve essere visualizzato o meno il pannellino di controllo con il volume e i tasti del "play", "pause" e "stop"; se si ="TRUE", altrimenti ="FALSE".

l'attributo HIDDEN è molto comodo nel caso in cui si volesse mettere una musica di sottofondo ad una pagina. Non appena caricata, la musica parte. E' ovvio, però, che non sarà possibile all'utente di variare il volume, fermarla, ecc.

Internet Explorer v.3 ancora non gestisce correttamente il comando EMBED (forse lo farà la versione 4); per ora è comunque disponibile, solo per lui, il comando <BGSOUND>, per specificare, appunto, una musica di

sottofondo alla pagina.

Per quanto riguarda i suoni, si deve sapere che i file audio con estensione .wav sono ascoltabili solo con sistemi windows. Per dare a tutti la possibilità di ascoltare i files audio che vorrete inserire nelle vostre pagine, li dovete convertire, ad esempio con "GoldWave", in formato "Sun Audio" (estensione .au); i file in questo formato vengono riprodotti direttamente dal browser, senza la necessità di usare applicazioni di supporto esterne.

Inoltre i file audio hanno il problema di essere in genere molto grandi, cosa che ne rende lento il caricamento: i file di tipo .au hanno di positivo il fatto di avere una maggiore compressione (fino ad un rapporto 1/20 rispetto ai file .wav) e quindi vengono letti molto più rapidamente.

Per quanto riguarda l'inserimento di animazioni, la procedura è la stessa. Anche in questo caso esisto molti formati: MPEG Video (.mpg), QuickTime (.mov), Audio/Video Interleaved (quello di Windows: .avi), AutoDesk Animator (.fli, .flc), ecc. Alcuni browser riconoscono alcuni di questi formati (quelli più diffusi sono .mpg e .mov); in caso contrario, come per i suoni, il browser chiederà con quale applicazione esterna lanciare l'animazione. Anche per le animazioni esistono programmi, come Adobe Premiere, che permettono la conversione fra i diversi formati.

Normalmente si utilizzano anche delle immagini animate in formato gif. Queste immagini si inseriscono esattamente come delle comuni immagini gif (utilizzando)

4. Formattazione avanzata ed uso di tabelle

4.1. Margini, rientri e allineamento

4.2. La gestione delle 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 table. 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 rimarrà invariata, cioè 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 sarà diversa a seconda della risoluzione video di colui, o colei, che ne esamina il contenuto.

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

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

Cella 1	Cella 2	Cella 3
---------	---------	---------

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.

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

Mentre il codice sottostante fa uso di un nuovo tag `<tr>` che è una specie di ritorno a capo per le tabelle, quindi corrisponde ad una nuova riga.

```
<TABLE BORDER=1 WIDTH=300>  
<TR>  
<TD WIDTH=100>1</TD>  
<TD WIDTH=100>2</TD>
```

```
<TD WIDTH=100>3</TD>
</TR>
```

```
<TR>
<TD WIDTH=100>A</TD>
<TD WIDTH=100>B</TD>
<TD WIDTH=100>C</TD>
</TR>
</TABLE>
```

Il risultato della visualizzazione con il browser:

1	2	3
A	B	C

Lo spazio tra i vari campi di una tabella è definito all'interno del comando `</TABLE>` con **CELLSPACING=X** e **CELLPADDING=X** (dove X è 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 può essere modificata in diversi modi come nei pezzi di codice che seguono:

```
<TABLE WIDTH=300 HEIGHT=200>
<TD WIDTH =100 VALIGN=TOP>top </TD>
<TD WIDTH=100 VALIGN=BOTTOM>bottom </TD>
<TD WIDTH=100 VALIGN=MIDDLE>middle </TD>
</TABLE>
```

```
<TABLE WIDTH=300 HEIGHT=200>
<TD WIDTH =100 ALIGN=RIGHT>destra </TD>
<TD WIDTH=100 ALIGN=CENTER>centro </TD>
<TD WIDTH=100 ALIGN=LEFT>sinistra</TD>
</TABLE>
```

```
<TABLE WIDTH=300 HEIGHT=200>
<TD WIDTH =100 BGCOLOR="red">rosso</TD>
<TD WIDTH=100 BGCOLOR="yellow">giallo</TD>
<TD WIDTH=100 BGCOLOR="blue">azzurro</TD>
</TABLE>
```

```
<TABLE WIDTH=300 HEIGHT=230>
<TD WIDTH =100 BACKGROUND="sfondo1.jpg">prova </TD>
<TD WIDTH=100 BACKGROUND="sfondo2.jpg">Prova </TD>
<TD WIDTH=100 BACKGROUND="sfondo3.jpg">Prova </TD>
</TABLE>
```

Attributi del tag TABLE

WIDTH Specifica la larghezza della tabella, in pixel o in percentuale rispetto alla finestra del browser.

HEIGHT Specifica l'altezza.

ALIGN Specifica l'allineamento del contenuto delle celle.

VALIGN Ne specifica l'allineamento verticale rispetto ai bordi delle celle.

BGCOLOR Definisce, con le solite modalità (`BGCOLOR="#rrggbb"`), il colore di sfondo della tabella.

BORDERCOLOR Definisce il colore che dovranno avere i bordi.

BORDERCOLORDARK Specifica il colore della parte illuminata dei bordi (effetto 3D). Solo Internet Explorer.

BORDERCOLORLIGHT Specifica il colore della parte buia dei bordi (effetto 3D). Solo Internet Explorer.

BORDER Specifica la larghezza del bordo esterno della tabella; "0" per non avere bordi.

CELLPADDING Serve per imporre un contorno interno alle celle che deve essere lasciato vuoto.

CELLSPACING Impone la distanza fra le celle, ovvero la larghezza dei bordi interni.

Attributi dei tag `<TR>`, `<TH>` e `<TD>`

ALIGN Permette di specificare l'allineamento dei dati all'interno di una cella. Può assumere i valori **RIGHT** (per l'allineamento a destra), **LEFT** (per l'allineamento a sinistra) e **CENTER** (per la centratura nella casella). Se utilizzato con il comando `<TR>` permette di associare un allineamento a tutti gli elementi di una stessa riga. Il default è **LEFT** per le celle dati e **CENTER** per le celle d'intestazione.

VALIGN Funziona esattamente come **ALIGN**, ma permette di stabilire l'allineamento verticale dei paragrafi nelle celle. I valori accettati sono **TOP** (allineamento in alto), **MIDDLE** (centratura) e **BASELINE** (allineamento in basso). Il default è **CENTER** per tutte le celle.

COLSPAN Controlla il numero di colonne su cui una cella si può estendere. Utilizzandolo insieme a **ROWSPAN** è possibile creare delle celle larghe 2 o più righe/colonne.

ROWSPAN Definisce il numero di righe su cui incide la casella.

HEIGHT Definisce l'altezza della cella, o dell'intera tabella.

WIDTH Definisce la larghezza della cella, o dell'intera tabella.

NOWRAP Indica che il testo contenuto nella cella non dovrà essere spezzato su più linee.

BGCOLOR Specifica quale deve essere il colore dell'intera tabella, o della riga, o della singola cella.

BORDERCOLOR Specifica il colore che devono avere i bordi.

BACKGROUND Attributo riconosciuto solo da Internet Explorer, serve per specificare una immagine da utilizzare come sfondo.

Dopo aver letto il significato di questi comandi e dei relativi parametri con poche prove è possibile creare tabelle di notevole effetto.

Le celle della tabella possono contenere un qualsiasi elemento HTML: testo, immagini, collegamenti. Utilizzando la tabella è quindi possibile, ad esempio, creare una griglia di pulsanti in cui sono inserite immagini accanto al testo.

E' molto importante notare che il comando <TABLE> va chiuso con il comando </TABLE> alla fine della tabella. Se così non viene fatto è molto probabile che il browser non riesca ad interpretare correttamente la pagina e la visualizzi completamente alterata.

Un'altra cosa interessante è la possibilità di nidificare le TABLE: una cella può contenere a sua volta un'altra TABLE. Come al solito si deve controllare però se le cose funzionano... e ricordarsi che se in locale va tutto bene non è detto che sia così anche con altri browser!

Infine una piccola applicazione: per generare una cornice intorno ad un testo si può utilizzare il bordo di una tabella inserendo tutto il testo in una sola cella.

5. Mappe sensibili

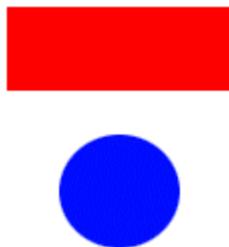
Ormai si sa che è possibile "ancorare" ad un'altra pagina un qualsiasi testo o una qualsiasi immagine; è sufficiente racchiudere il tutto fra i tag `<A...> *** `.

Ma si può fare di più.

Sarebbe bello e d'effetto avere la possibilità di inserire nelle pagine WEB una figura, un disegno, una foto, che ci collega ad altre pagine, a seconda di **DOVE** si va a cliccare col cursore. Ebbene, ciò è possibile con le cosiddette "Mappe sensibili". Una mappa, non è altro che un file di testo (.map, in genere) associato ad una immagine di tipo GIF, in cui sono contenute delle informazioni sugli "hot spot", ovvero sui punti significativi dell'immagine. Un hot spot può avere forma rettangolare, circolare, o poligonale qualsiasi. Le informazioni vengono date in forma di coordinate.

Il primo passo nell'utilizzo di mappe consiste ovviamente nella scelta dell'immagine originale da utilizzare. L'immagine può essere di qualsiasi formato supportato dal browser (anche se come visto in precedenza le immagini più comuni sono le gif) e devono contenere delle aree con significato diverso alle quali si possa associare un evento diverso al click del mouse.

Come esempio si può considerare un immagine molto semplice dove sono evidenti due aree una rettangolare e una circolare:



L'immagine può essere inserita nella pagina html con il classico tag `` ma ancora non si osservano effetti particolari al passaggio del mouse.

Se l'immagine è inserita in un tag `` allora un eventuale click del mouse quando il cursore si trova sopra l'immagine ha come effetto l'apertura della pagina "destinazione". Se invece l'utente volesse configurare link diversi per zone diverse della immagine allora si deve aggiungere un file di configurazione e si devono effettuare opportune modifiche al tag .

Il file di configurazione è un file di testo con un'estensione "map" e contiene delle istruzioni per il server web riguardanti la destinazione del link.

Un esempio di file .map può essere:

```
default    http://www.unifi.it
rect       http://www.dsi.unifi.it/ 10,10 90,90
circle     http://www.die.unifi.it/ 150,150 120,150
```

Che chiamiamo map1.map.

La prima riga imposta il server dove andare se è stata attivata una regione non impostata (default) mentre la seconda riga informa il server che se durante l'evento click il cursore del mouse si trovasse dentro l'area rettangolare di coordinate 10,10 90,90 allora la destinazione è <http://www.dsi.unifi.it/> .

Analogamente funziona la terza riga del file di configurazione.

Si possono inserire tante righe rect o circle quante sono necessarie e l'utente deve solo preoccuparsi di non inserire aree sovrapposte.

Sono disponibili vari tool che permettono di inserire in modo più veloce le aree nelle immagini tramite la selezione con il mouse.

Una volta creato il file deve essere salvato nella stessa directory dell'immagine.

L'ultimo passo da effettuare per rendere attiva l'immagine consiste nella modifica del tag `<a href>` in modo da ridirigere il link verso il file .map. Nel caso dell'esempio considerato il tag diventa:

```
<A HREF="map1.map"><IMG SRC="map1.gif" ISMAP></A>
```

Si osserva inoltre l'attributo ISMAP che indica al server che l'immagine considerata è tipo mappa.

Le righe del file .map possono contenere quattro figure geometriche:

circle *URL* *Xc,Yc* *Xp,Yp*

specifica un'area circolare, di centro *Xc,Yc* e il cui confine passa per il punto *Xp,Yp*.

rect *URL* *X1,Y1* *X2,Y2*

area rettangolare i cui vertici alto-sinistro e basso-destro hanno coordinate *X1,Y1* e *X2,Y2* rispettivamente.

poly *URL* *X1,Y1* *X2,Y2* *X3,Y3* *Xn,Yn*

area poligonale i cui vertici consecutivi sono specificati dalle coordinate *Xi,Yi*.

default *URL*

questa opzione individua tutta la parte di immagine che rimane al di fuori delle aree specificate con circle, rect e poly. Se omessa, cliccare al di fuori delle aree specificate non avrà alcun effetto.

NOTA: l'origine delle coordinate (0,0) è l'angolo in alto a sinistra dell'immagine; l'asse delle X va verso destra, quello delle Y verso il basso.

*NOTA: le mappe **NON** possono essere provate localmente: è necessario che il file .map sia messo nella corretta directory del vostro server, e che siate on-line. (protocollo http attivato)*

Un modo per ovviare a questo problema c'è, e consiste nell'inserire le specifiche delle vostre mappe direttamente nel codice HTML, secondo le modalità che seguono.

Innanzitutto inserite l'immagine (.gif) che volete che sia una mappa cliccabile, aggiungendo al solito tag, l'indicazione di una mappa da utilizzare, mediante l'attributo USEMAP, il cui valore è un nome di fantasia a vostra scelta preceduto da un cancelletto (#).

Subito di seguito, indicate la specifica delle aree che dovranno rappresentare gli hot spots, secondo la sintassi seguente:

```
<IMG SRC="map1.gif" USEMAP="#pippo">
```

```
<MAP NAME="pippo">
```

```
<AREA SHAPE="rect" COORDS="X1,Y1,X2,Y2" HREF=" http://www.unifi.it">
```

```
<AREA SHAPE="circle" COORDS="Xc,Yc,R" HREF=" http://www.unifi.it">
```

```
</MAP>
```

In questo caso le coordinate del rettangolo e del poligono sono come nel caso precedente, mentre i parametri dei cerchi vengono dati mediante le coordinate del centro, seguite dal raggio; notare che qui tutte le coordinate sono separate da virgole, mentre nel caso dei file .map si usavano anche degli spazi.

6. I frame

6.1. Utilizzo, navigazione e formattazione di 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'inutilità 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 può portare a traguardi di pessima impostazione grafica, giungendo ad un risultato opposto rispetto a quello preventivato. Un punto certamente a sfavore dei frames è la loro incompatibilità con i programmi di navigazione dei non vedenti che davanti a tali strumenti si bloccano, non permettendo la lettura delle pagine. Una buona soluzione è quella di creare una versione frame ed una versione senza frame (come questa stessa guida ha fatto).

Come si creano i frames?

Prima di tutto diamo uno sguardo ai comandi HTML per la gestione dei frame.

N2.0	Documento Frame	<FRAMESET></FRAMESET>	(al posto di <BODY>)
N2.0	altezza in righe	<FRAMESET ROWS= , , , ></FRAMESET>	(pixel o %)
N2.0	altezza in righe	<FRAMESET ROWS=* ></FRAMESET>	(* = misura relativa)
N2.0	larghezza in colonne	<FRAMESET COLS= , , , ></FRAMESET>	(pixel o %)
N2.0	larghezza in colonne	<FRAMESET COLS=* ></FRAMESET>	(* = misura relativa)
N3.0b	larghezza della cornice	<FRAMESET BORDER=? >	
N3.0b	cornice	<FRAMESET FRAMEBORDER="yes no" >	
N3.0b	colore della cornice	<FRAMESET BORDERCOLOR="#\$\$\$\$\$\$" >	
N2.0	Definizione del Frame	<FRAME >	(contenuto di ogni singolo riquadro)
N2.0	documento da visualizzare	<FRAME SRC="URL" >	
N2.0	denominazione del frame	<FRAME NAME="****" _blank _self _parent _top >	
N2.0	larghezza dei margini	<FRAME MARGINWIDTH=? >	(margine destro e sinistro)
N2.0	altezza dei margini	<FRAME MARGINHEIGHT=? >	(margine alto e basso)
N2.0	barra di scorrimento?	<FRAME SCROLLING="YES NO AUTO" >	
N2.0	dimensione non modificabile	<FRAME NORESIZE >	
N3.0b	cornice	<FRAME FRAMEBORDER="yes no" >	
N3.0b	colore della cornice	<FRAME BORDERCOLOR="#\$\$\$\$\$\$" >	
N2.0	contenuto in assenza di frame	<NOFRAMES></NOFRAMES>	(per i vecchi browsers)

Per creare una pagina divisa in frame è necessario creare più files HTML richiamati da un file principale che è 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 dovrà visualizzare. Come scritto sopra questi due frames devono essere gestiti da un terzo file il quale dovrà richiamarli assegnando loro una parte della pagina. Ecco il codice di questa pagina:



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

Come si vede il codice del frame è 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) è stabilita dal comando `rows="80,*`, che sta a significare che il frame alto (che in questo caso è 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. È necessario dare un nome al frames (`name="alto"`) e indicare il file HTML che dentro al frame dovrà 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 me name="centrale" src="central.htm">
  <frame me name="alto" src="top.htm">
</FRAMESET>
```

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

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

```
<FRAMESET cols="100,*" >
<frame me name="sx" src="sx.htm">
<frame me 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, è utile inserire del codice che avverta della presenza di frames e della impossibilità da parte del browser di visualizzarle.

Il codice da inserire:

```
<noframe>

<HTML>
<body>

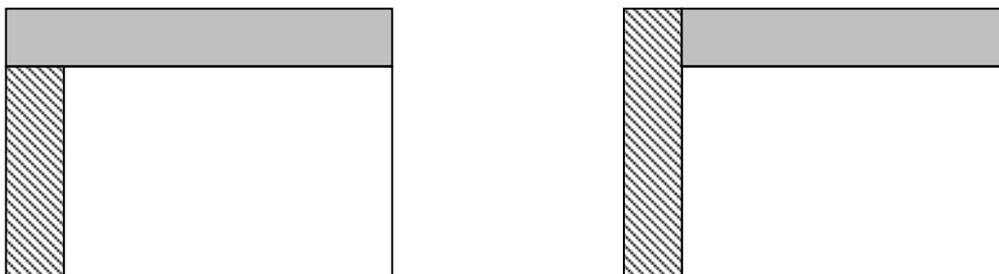
  Attenzione. Il tuo browser non supporta l'opzione dei frame. Per
  visualizzare queste pagine e' necessario scaricare un browser
  che supporti tali opzioni. Ti consiglio Netscape 3.0 o
  superiore.

</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.

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



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

```
</frameset>
```

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

Per eliminare sempre le barre di scorrimento (scrollbars):

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

Per renderle sempre visibili:

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

Per renderle visibili solo quando servono:

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

Per quanto riguarda i links all'interno dei frames (cioè 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 è stato assegnato ai vari frames. Nome che non si riferisce al file ma a quanto scritto dopo "name=". In questo caso, per esempio:

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

il nome assegnato e' "alto".

Consideriamo la seguente pagina divisa in frames:

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



Considerando 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 è:

```
<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>
```

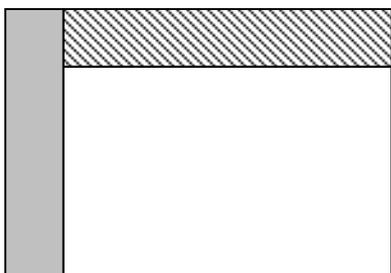
Se inserisci il codice:

```
<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 me name="sx" src="sx.htm">  
<frameset rows="100,*">  
  <frame me name="alto2" src="top.htm">  
  <frame me 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>
```

7. Form ed interazioni con il server

Tutte le belle cose descritte aiutano a costruire una pagina che contiene informazioni che possano essere lette. Ma come fare se si vuole che il lettore possa anche fornire informazioni, oltre che leggerle? La risposta, in una parola, si chiama forms.

L'idea è piuttosto semplice. Per prima cosa, nella pagina si specificano, attraverso opportuni tags, delle zone di input, in cui chi accede alla pagina potrà scrivere, o comunque interagire: i form, quindi, specificano sia zone di input alfanumerico, di scelta guidata (tipicamente, una lista di possibilità), oppure di input univoci (un bottone di OK, per esempio, o dei radio button)

7.1. Controlli, pulsanti e campi nascosti

Tags per dichiarare i form più comuni:

Questi tags potrebbero essere istanziati in modo differente da browser a browser.

- Il tag **FORM**
...permette di dichiarare un form. Il formato è

```
<FORM ACTION="..url.." METHOD="..metodo..">  
  ..altri tags e testo vario...  
</FORM>
```

Ogni zona interattiva all'interno della vostra pagina verrà specificata da una coppia <FORM> .. </FORM>.

Altri tags servono per definire che tipo di form sarà.

ACTION specifica il queryserver da utilizzare.

L'attributo METHOD può valere "GET" o "POST". In breve, con GET i dati vengono inviati al queryserver sulla "linea di comando", mentre con POST vengono inviati separatamente come pacchetto dati. Per input brevi, GET va bene, ma se la riga di comando supera un certo numero di caratteri (128?,256? - non ricordo) viene troncata, quindi POST è più sicuro.

I tag FORM non si possono innestare (anche se ce ne possono essere tanti nello stesso documento..)

- Come specificare una lista di opzioni ?
All'interno di un tag FORM, con

```
<SELECT NAME="nome" >  
<OPTION> Opzione 1  
<OPTION> Opzione 2  
  ...  
<OPTION> Opzione n  
</SELECT>
```

- Come specificare una linea di input libero, bottoni, radio button, checkboxes, etc ?
All'interno di un tag FORM, con

```
<INPUT> TYPE="..tipo.." NAME="..nome.." VALUE="..valore.." CHECKED  
SIZE=larghezza,altezza MAXLENGHT=n>
```

Il tipo può essere uno tra "text", "checkbox", "radio", "submit", "reset".
Ovviamente, gli attributi vanno usati con buon senso: CHECKED ha senso solo se si specifica "checkbox" o "radio" come tipo, etc.

NAME (dev'esserci per tutti tranne "submit" e "reset") indica il nome simbolico con cui il contenuto dell'input sarà indicato quando i dati saranno passati al queryserver.

VALUE specifica il valore per default o l'etichetta di un bottone.

Un tag di tipo "submit" deve esserci perchè è il solo mezzo per permettere all'utente di dire al suo client "ho finito".

- Come specificare un'area di input grande?
Con

```
<TEXTAREA NAME="..nome.." ROWS=n1 COLS=n2>  
..contenuti di defaults..  
</TEXTAREA>
```

7.2. *Letture delle form.*

Quando un utente capita in una pagina con un form, probabilmente vorrà inserirvi dei dati, in particolare, uno dei form è di solito un bottone che manda al client il messaggio "Ok, ho finito". Quando il client riceve questo messaggio, prende tutto ciò che l'utente ha scritto o selezionato, impacchetta questi dati con un formato preciso, e li spedisce al server.

A questo punto, il server passa i dati ad un altro che li analizza e specifica un documento (html) di risposta, che viene inviato al client, che lo mostra. Questo programma si chiama query-server in quanto serve le richieste relative ad una specifica serie di form.

Esempio pratico:

supponiamo che la pagina di Pippo contenga un form che permette di scrivere un messaggio: l'utente scrive, poi "schiaaccia" il pulsante "send"; il client guarda nella pagina, scopre che queryserver va utilizzato, e manda al server web i dati più quest'ultima informazione; il server web chiama il queryserver, passandogli i dati; il queryserver aggiunge l'header SMTP (quello delle mail, per intenderci) e chiama il programma di posta; poi genera una pagina temporanea con dentro scritto " Il tuo form è stato spedito!", la manda al client che la visualizza.

"Query-server" è solo un nome altisonante per un programma che accetta dei dati in un formato specifico, fa qualcosa, e poi rimanda indietro un documento di risposta.

Questo programma si può scrivere in molti linguaggi diversi, da C, a Perl, oppure come script di shell .

Le specifiche di come interfacciare il proprio programma, scritto nel linguaggio prediletto, con il server web (ovvero: come accettare i dati, come "spacchettarli", come rispondere) costituiscono la cosiddetta CGI (Common Gateway Interface). In pratica, per il C, si tratterà di linkare una libreria che contiene delle funzioni apposite; per gli script di shell, di leggere il contenuto di alcune variabili di environment;etc.

Un unico dettaglio in più: ricordate che i queryserver devono essere visibili al server web (cioè, in pratica, al httpd) e che spesso quindi devono risiedere in una directory particolare, che a volte, per motivi di sicurezza, non è direttamente scrivibile dagli utenti di un sistema.

7.3. Cenni a CGI e ASP

CGI significa "*Common Gateway Interface*", ed è il metodo con cui un server agisce con database, documenti, e altri programmi inviando o ricevendo dati; questi dati saranno poi visualizzati sul web. In pratica, saper scrivere CGI significa programmare il web per il web. Il linguaggio "principe" per scrivere i CGI è il [Perl](#), anche se è possibile utilizzare altri linguaggi (C su tutti).

[.....*****.....]

Le Active Server Pages (**ASP**) sono inserite nei file con estensione .asp.

Un file .asp è un file testuale che contiene le seguenti proprietà:

- Testo
- Tag HTML
- Comandi di Script. Un comando di script istruisce il computer a fare qualcosa, come assegnare un valore ad una variabile.

E' alquanto facile creare un file .asp: Basta rinominare un qualsiasi file HTML, sostituendo i file con le estensione esistenti .htm o .html con l'estensione .asp. Per creare uno script .asp disponibile agli utenti del Web, bisogna salvare il file in una directory appartenente al Web server ed essere sicuri di avere il permesso d'esecuzione degli script. Quando il file verrà chiamato dal browser il processo ASP restituirà codice HTML.

Uno **Script** è composto da una serie di comandi di Script.

L'esecuzione di uno script invia una serie di comandi ad un motore di scripting (scripting engine), che lo interpreta. Gli Script sono scritti in linguaggi aventi precise regole; tra queste: per poter usare un linguaggio di scripting, il server deve disporre di uno scripting engine che conosca il linguaggio. ASP contiene gli scripting engines per i linguaggi di scripting VBScript and JScript. Il primo linguaggio di scripting che ASP assume, se non altrimenti stabilito, è VBScript.

ASP in realtà non è un linguaggio di scripting ma offre un environment (ambiente) che processa gli scripts che vengono incorporati nelle pagine HTML.

Delimitatori ASP

I Tag HTML sono differenziati dal testo dai delimitatori. Un delimitatore è un carattere o una sequenza di caratteri che marcano l'inizio e la fine di una unità.

Nel caso dell'HTML, questi delimitatori sono il simbolo minore (<) e maggiore (>).

In modo del tutto simile, i comandi e le espressioni di output degli script ASP sono differenziati sia dal testo che dai Tag HTML dai delimitatori. ASP usa i delimitatori <% e %> per includere i comandi di script. Ad esempio, il comando <% *nMioNumero* = 5 %> assegna il valore numerico 5 alla variabile *nMioNumero*.

ASP usa i delimitatori <%= e %> per racchiudere le espressioni di output.

Ad esempio, l'espressione di output <%= *nMioNumero* %> invia il valore 5 (valore corrente della variabile) al browser.

7.4. Cenni a Javascript

8. APPENDIX

Questa sezione contiene i codici per caratteri particolari, non disponibili nel set di caratteri dell'HTML. Essi possono essere di due tipi: mnemonici, e numerici (Equivalenti ASCII).

Quelli mnemonici devono iniziare con una e commerciale (&) e finire con un punto e virgola (;).

Quelli numerici iniziano con una e commerciale seguita da un cancelletto (&#) e finiscono anch'essi con un punto e virgola (;).

& Cod. mnemonico ;

oppure

&# Cod. numerico ;

Lettere maiuscole			
Cod. mnemonico	Cod. numerico	Come appare	Descrizione
AElig	198	Æ	Dittongo AE
Aacute	193	Á	A con accento acuto
Acirc	194	Â	A con accento circonflesso
Agrave	192	À	A con accento grave
Aring	197	Å	A cerchiata
Atilde	195	Ã	A con tilde
Auml	196	Ä	A con dieresi o umlaut
Ccedil	199	Ç	C con cediglia
ETH	208	Ð	E con th
Eacute	201	É	E con accento acuto
Ecirc	202	Ê	E con accento circonflesso
Egrave	200	È	E con accento grave
Euml	203	Ë	E con dieresi o umlaut
Iacute	205	Í	I con accento acuto
Icirc	206	Î	I con accento circonflesso
Igrave	204	Ì	I con accento grave
Iuml	207	Ï	I con dieresi o umlaut
Ntilde	209	Ñ	N con tilde
Oacute	211	Ó	O con accento acuto
Ocirc	212	Ô	O con accento circonflesso
Ograve	210	Ò	O con accento grave
Oslash	216	Ø	O con barra
Otilde	213	Ö	O con tilde
Ouml	214	Û	O con dieresi o umlaut
THORN	222	Þ	Thorn
Uacute	218	Ú	U con accento acuto
Ucirc	219	Û	U con accento circonflesso
Ugrave	217	Ù	U con accento grave
Uuml	220	Ü	U con dieresi o umlaut
Yacute	221	Ý	Y con accento acuto

Lettere minuscole			
Cod. mnemonico	Cod. numerico	Come appare	Descrizione

aelig	230	æ	Dittongo ae
aacute	225	á	a con accento acuto
acirc	226	â	a con accento circonflesso
agrave	224	à	a con accento grave
aring	229	å	a cerchiata
atilde	227	ã	a con tilde
auml	228	ä	a con dieresi o umlaut
ccedil	231	ç	c con cediglia
eacute	233	é	e con accento acuto
ecirc	234	ê	e con accento circonflesso
egrave	232	è	e con accento grave
eth	240	ð	e con th
euml	235	ë	e con dieresi o umlaut
iacute	237	í	i con accento acuto
icirc	238	î	i con accento circonflesso
igrave	236	ì	i con accento grave
iuml	239	ï	i con dieresi o umlaut
ntilde	241	ñ	n con tilde
oacute	243	ó	o con accento acuto
ocirc	244	ô	o con accento circonflesso
ograve	242	ò	o con accento grave
oslash	248	ø	o con barra
otilde	245	õ	o con tilde
ouml	246	ö	o con dieresi o umlaut
szlig	223	ß	s tedesca
thorn	254	þ	thorn
uacute	250	ú	u con accento acuto
ucirc	251	û	u con accento circonflesso
ugrave	249	ù	u con accento grave
uuml	252	ü	u con dieresi o umlaut
yacute	253	ý	y con accento acuto
yuml	255	ÿ	y con dieresi o umlaut

Oltre alle lettere, bisogna usare dei codici anche per quei caratteri che hanno un significato particolare in HTML, per evitare ambiguità. Essi sono:

Caratteri HTML			
Cod. mnemonico	Cod. numerico	Come appare	Descrizione
lt	60	<	Simbolo di minore
gt	62	>	Simbolo di maggiore
amp	38	&	E commerciale
quot	147	"	Doppio apice
nbsp	32		Spazio

NOTA: usando il codice numerico, è possibile riprodurre qualsiasi carattere secondo la corrispondenza ASCII.