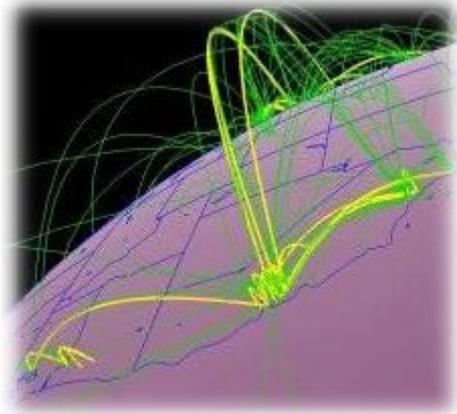


La combinazione di questi tre elementi ha determinato la nascita del Web e, di conseguenza, l'esplosione della popolarità di Internet presso il grande pubblico. Ufficializzato nei primi anni Novanta (1992-1994), il Web ha reso possibile quel concetto di **navigazione ipertestuale** che oggi viene ormai dato per scontato: pagine illustrate e animate, collegamenti su cui fare clic per spostarsi istantaneamente da un sito all'altro e da un argomento all'altro. Era nata la Ragnatela (Web) mondiale dell'informazione.



Le varie versioni del linguaggio HTML

Come ogni standard informatico, anche il linguaggio HTML si è evoluto attraverso una serie di versioni successive.

- Le **prime versioni** si limitavano a consentire la creazione di semplici pagine di testo con collegamenti ipertestuali. Le potenzialità di impaginazione grafica e di sviluppo multimediale erano scarse o nulle.
- La **versione 3.2** (gennaio 1997) ha introdotto numerose novità e ha finalmente consentito un salto di qualità, con l'introduzione di vari attributi grafici e il supporto di un'ampia gamma di materiali multimediali.
- La **versione 4.0**, ufficializzata solo alla fine degli anni Novanta, ha esteso le potenzialità creative del linguaggio HTML, accogliendo e formalizzando alcuni aspetti che già erano stati anticipati in via ufficiosa e non standard (i frame, gli script, i fogli di stile, i layer e così via).

In effetti, uno dei maggiori problemi per chi crea pagine HTML è la **compatibilità**.

Da un lato, il linguaggio HTML è nato all'insegna della più assoluta universalità (indipendenza dalla piattaforma): i file HTML sono di per sé riconosciuti dalla quasi totalità dei sistemi informatici, indipendentemente dalle differenze di hardware e software.

D'altra parte, sussistono differenze (talvolta significative) nel modo in cui i singoli sistemi, e in particolare i singoli browser, visualizzano uno stesso documento HTML.

Nel seguito del modulo saranno volta per volta evidenziati gli elementi (tag) che possono porre ancora oggi problemi di compatibilità.

Come **regola generale**, si tenga presente che se si sfruttano a fondo tutte le caratteristiche più aggiornate del linguaggio HTML si rischia di escludere dal proprio target un certo numero di destinatari: tutti coloro che non dispongono di strumenti di consultazione (browser) altrettanto aggiornati.

Sezione 1. Il linguaggio HTML e le pagine Web

Come funziona una pagina Web

Sullo schermo di chi naviga, una pagina Web appare come un unico documento multimediale. Nella finestra del browser prende forma una struttura unitaria e armonica, in cui si combinano testi, immagini, animazioni ed eventuali altri componenti. Alcuni di questi elementi sono "cliccabili": sono cioè associati in modo ipertestuale ad altri documenti e siti Web.

Ciò che sullo schermo appare come un tutt'uno, in realtà non è un unico file, ma il frutto di un assemblaggio effettuato in tempo reale dal browser.

La visualizzazione di una pagina Web dipende infatti da:

- un **file principale, in formato HTML**: questo file rappresenta il nucleo della pagina Web: contiene il **testo** del documento e le **istruzioni** (tag) che ne determinano l'aspetto e i collegamenti con altri file e indirizzi;
- altri **file multimediali**, che concorrono a completare il documento: immagini JPEG e/o GIF, animazioni, suoni, filmati, applet Java ecc.; questi elementi sono dunque **esterni** rispetto al file principale, e vengono richiamati al suo interno (in fase di visualizzazione) mediante apposite istruzioni inserite nel suo codice sorgente;
- eventuali **file "di servizio", come i fogli-stile**, che rappresentano però un caso particolare (se ne parla nell'ultima parte del modulo).

Il formato HTML

Il nucleo fondamentale di ogni pagina Web è dunque un file scritto in linguaggio HTML. L'acronimo HTML sta per **HyperText Markup Language**: "linguaggio per la marcatura (definizione) degli ipertesti". Si tratta di un linguaggio di programmazione elaborato all'inizio degli anni Novanta, e che ha segnato una svolta epocale nel funzionamento di Internet: con l'HTML è nato infatti il Web, ossia la navigazione ipertestuale e multimediale tra i contenuti della Rete.

In effetti, è un po' vago e non del tutto esatto definire l'HTML come un **linguaggio di programmazione**.

Ai fini di questo modulo, vale la pena di precisare un'importante distinzione concettuale. Un linguaggio di programmazione può essere di due tipi:

- **compilato**: la sequenza di istruzioni (codice sorgente), una volta completata, viene codificata in modo irreversibile, creando un file eseguibile che non può essere decompilato se non mediante strumenti e tecniche particolari; tra i linguaggi di questo tipo vi sono C++, Visual Basic e tutti gli altri strumenti della programmazione "tradizionale";
- **interpretato**: il codice sorgente rimane immutato, e resta sempre leggibile in modo "trasparente"; le istruzioni vengono eseguite passo passo solo quando vengono lette da un'applicazione in grado di interpretarle (per esempio, un browser); l'HTML appartiene a questa categoria.

Da questo fatto derivano alcune importanti implicazioni per chi crea le pagine HTML: per esempio, la difficoltà (o meglio l'impossibilità) di proteggere i propri documenti dalla copia. E' molto semplice copiare e riutilizzare il codice sorgente di una pagina Web, opportunamente modificato: la tutela del copyright è in vigore sul piano legale ed etico, ma è facilmente aggirabile sul piano pratico.

Un file HTML non è altro che un **file di testo** (in particolare, di "puro testo" o di "testo ASCII"), sostanzialmente identico ai file di tipo TXT che si possono scrivere con un comunissimo editor di testi, come il Blocco Note di Windows o Simple Text di Mac OS.

Per poter funzionare come pagina Web, tuttavia, questo file di testo deve avere due prerogative:

- deve avere il **soffisso .HTML (o .HTM)**, anziché .TXT;

- deve contenere, oltre al testo vero e proprio, anche le **istruzioni** che consentono al browser di riconoscerlo e gestirlo correttamente.

I tag

Queste istruzioni sono chiamate **tag ("marcatori")**, e presentano alcune caratteristiche fondamentali che è bene conoscere sin dall'inizio:

- Un tag è un comando **racchiuso tra i segni < (minore) e > (maggiore)**, chiamati anche parentesi angolari; per esempio: ****, ****, **<HEAD>**, **<BODY>** ecc. Quando il browser incontra questi simboli in un documento, "capisce" che si tratta di istruzioni da eseguire e non di testo da visualizzare sullo schermo.
- La maggior parte dei tag (ma non tutti) funziona **"in coppia"**, secondo questo schema:

<TAG_DI_INIZIO> testo o elemento a cui si applica l'istruzione **</TAG_DI_FINE>**

Per esempio, il tag **** serve a formattare un testo in grassetto (Bold) e deve essere espresso con un'istruzione di inizio e una di fine:

**** questa frase è in grassetto ****

- I tag, in generale, possono contenere **attributi** che ne definiscono determinate proprietà. Ogni attributo può assumere differenti **valori**.

Comprendere i tag

A differenza dei normali linguaggi di programmazione, il linguaggio HTML può essere facilmente appreso da chiunque abbia una minima familiarità con l'uso del computer. Le istruzioni tipiche del linguaggio HTML, i tag, sono infatti abbastanza semplici da padroneggiare.

Principi di funzionamento

Per comprendere il funzionamento dei tag, basta ricorrere a un banale paragone. Immaginando di dover **dettare per telefono** un documento (per esempio una lettera, completa di formattazione), probabilmente ci si esprimerebbe in un modo simile a questo: Ecco l'inizio della lettera:

"Gentile signore, (A CAPO)
riguardo al nostro prodotto (GRASSETTO) **Pincopallino** (FINE GRASSETTO),
Le comunichiamo che..."

Ebbene, i tag HTML funzionano secondo un principio analogo.

Per esempio:

- per andare a capo si può inserire i tag **
** (Break) o **<P>** (Paragraph);
- per far apparire una parola in grassetto si utilizza il tag **** (in inglese Bold significa grassetto): **** Pincopallino ****

Sintassi

Come si può notare, in base agli esempi di cui sopra:

- alcuni tag sono **singoli**, ossia vanno inseriti uno per volta (come il tag **
**);
- altri tag funzionano a **coppie**, richiedono cioè un'istruzione di **apertura** (per esempio: **** per "inizio grassetto") e un'istruzione di **chiusura** (per esempio: **** per "fine grassetto").

Spesso è possibile includere **coppie di tag all'interno di altre coppie di tag**: in tal caso occorre però rispettare una sequenza simmetrica, o (in parole povere) "a sandwich".

Per esempio, il tag identifica il grassetto e il tag <I> (Italic) il corsivo. Ebbene, per formattare in grassetto corsivo una parola si deve scrivere:

 <I> Pincopallino </I>

oppure:

<I> Pincopallino </I>

ma **non**: <I> Pincopallino </I>.

Quasi tutti i tag prevedono **attributi** (facoltativi o, più raramente, obbligatori), ossia specifiche aggiuntive che ne influenzano il risultato.

La sintassi generica è:

<TAG ATTRIBUTO="valore">

Per esempio: <TABLE BORDER="1"> oppure .

In generale, occorre fare attenzione alla **differenza tra maiuscole e minuscole**.

- Per i **tag** e i rispettivi **attributi**, la differenza tra maiuscole e minuscole è ininfluente. Per esempio, il tag per il grassetto può essere scritto sotto la forma oppure , senza alcun vincolo di sorta. Tuttavia, è comune l'abitudine di scrivere in maiuscolo tutti i tag, per far risaltare più facilmente il codice HTML rispetto al testo del documento, quando si revisiona una pagina.
- Invece, per quanto riguarda i **valori** degli attributi (che possono essere numeri ma anche **parole, sigle, nomi di file** e così via), la differenza tra maiuscole e minuscole è di fondamentale importanza. Per esempio, scrivere **non** è la stessa cosa che scrivere . Questo tag serve a visualizzare nella pagina un'immagine, il cui file si chiama appunto "foto1.jpg": se nel tag il nome del file viene scritto in lettere maiuscole anziché minuscole (o viceversa) si rischia con ogni probabilità di incorrere in un errore.

Come si è detto, è molto facile accedere al codice HTML, cioè alla sequenza di testo e tag che è all'origine di ogni pagina Web. Ciascun browser contiene un comando grazie al quale si può **visualizzare il codice sorgente** della pagina attualmente caricata.

Ciò permette di effettuare subito un primo esercizio pratico. Sebbene ci si trovi all'inizio del modulo (e il codice sia ancora in gran parte incomprensibile), tale esercizio può essere utile per cominciare a familiarizzare con l'HTML e verificare concretamente cosa vi sia "dietro le quinte" di una pagina.

Visualizzare il codice sorgente HTML di una pagina

Ogni browser contiene un comando che permette di visualizzare il codice HTML relativo alla pagina Web correntemente caricata.

Visualizzando il codice (chiamato anche "sorgente") si possono scoprire tutti i dettagli e le impostazioni che hanno generato la pagina così come la vediamo in Internet.

Supponiamo di essere collegati a un sito Web (per esempio: <www.fineco.it>).

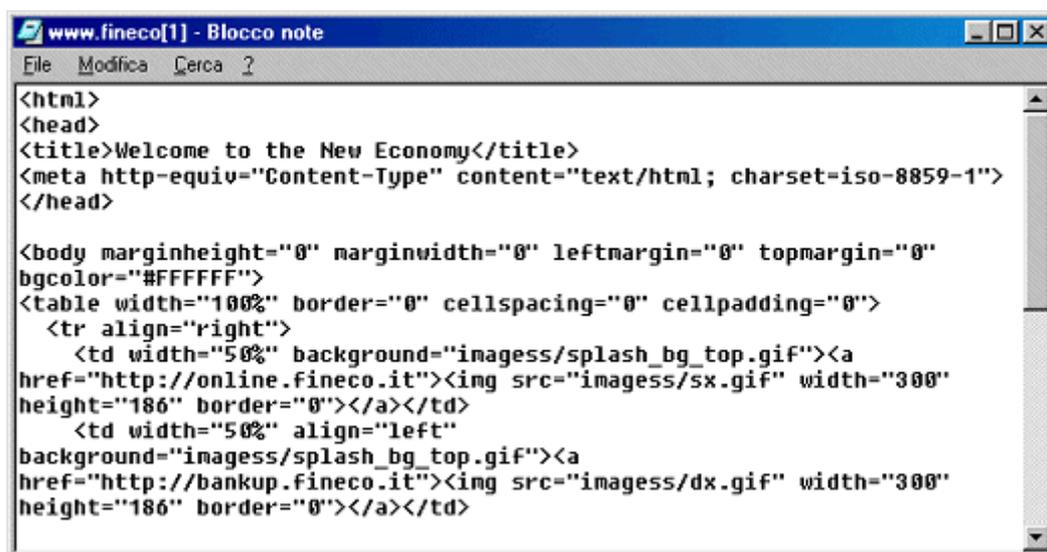
La pagina appare così nella finestra di Internet Explorer:



Ora si può aprire il menu **Visualizza** e cliccare sul comando **HTML**:



A questo punto, se si usa Windows 95/98, si aprirà automaticamente una finestra dell'editor di testo predefinito (il Blocco note) in cui appare integralmente il codice HTML della pagina; gli utenti Mac vedranno invece l'HTML in una nuova finestra di Explorer, simile a quella prodotta da Netscape.



Come scrivere un file HTML

Un file HTML può essere scritto a mano, parola dopo parola e tag dopo tag, oppure ricorrendo a un apposito programma (l'editor HTML).

- **Scrittura manuale del codice HTML** - Per creare un file HTML non servono necessariamente strumenti specializzati. Per i motivi appena spiegati, è sufficiente un banale editor di testo come il Blocco note di Windows o Simple Text di Mac OS. Nel seguito del modulo si farà riferimento a questo metodo. E' un ottimo sistema ai fini didattici (imparare l'HTML), ma nella pratica presenta sia vantaggi che svantaggi. Il vantaggio è che l'autore mantiene un controllo totale sul codice, creando file dalla struttura snella, razionale e "pulita". Lo svantaggio, abbastanza ovvio, è che si tratta di una soluzione faticosissima e in fin dei conti impraticabile se si deve costruire un sito complesso, con numerose pagine e un'impostazione grafica sofisticata.
- **Utilizzo di un editor HTML** - Esistono programmi specializzati nell'elaborazione di pagine Web: i cosiddetti editor di tipo WYSIWYG (What You See Is What You Get: "quello che vedi è quello che ottieni"). Questi programmi funzionano in modo analogo ai più potenti software di videoscrittura (come MS Word) e di impaginazione. Per usarli non serve conoscere l'HTML: si opera in modo grafico e intuitivo, manipolando oggetti e pulsanti tramite il mouse: il programma genera automaticamente il codice HTML necessario. Lo svantaggio è che non tutti questi editor generano un codice ottimizzato e perfettamente standard (cioè in grado di essere visualizzato nello stesso modo da tutti i browser).

Gli editor WYSIWYG

Gli editor HTML di tipo WYSIWYG sono strumenti potenti e intuitivi per la creazione di pagine e di siti Web.

La sigla WYSIWYG sta per What You See Is What You Get: "quello che vedi è quello che ottieni".

Ciò significa che, anziché scrivere a mano il codice HTML di ogni pagina, riga per riga e tag per tag, si può agire sullo schermo manipolando testo e immagini tramite il mouse.

Chiunque sappia utilizzare un buon programma di videoscrittura, di impaginazione e/o di grafica sicuramente può imparare molto in fretta a utilizzare un editor Web.

Gli editor WYSIWYG sono la scelta ideale se si devono costruire pagine complesse o siti molto articolati.

Tipi di editor Web

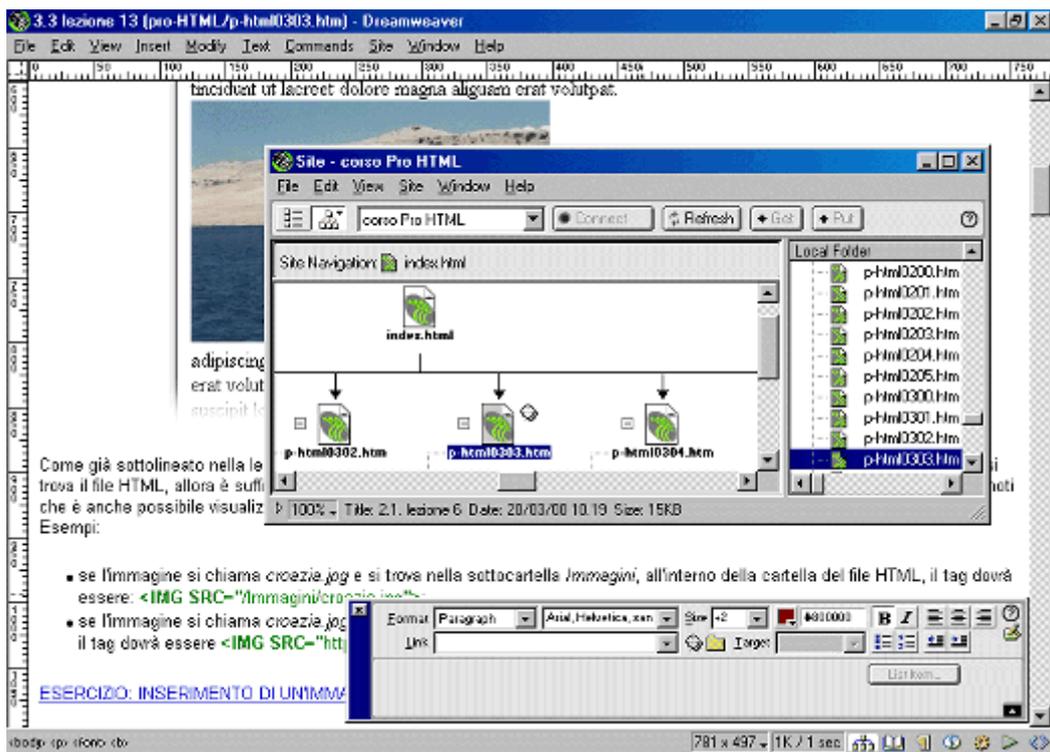
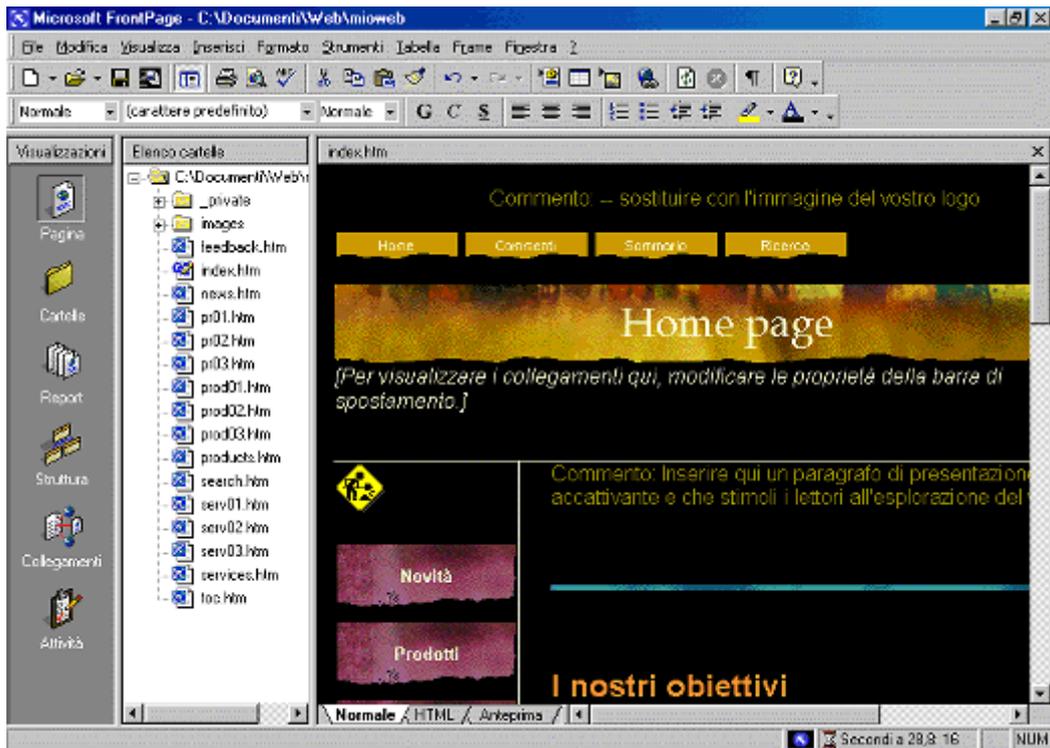
Gli editor Web si possono suddividere in due famiglie di strumenti:

- gli editor di **siti**, che permettono di gestire in modo automatico e coerente interi set di pagine Web collegate tra loro;
- gli editor di **pagine**, che consentono di creare e modificare una singola pagina per volta.

1. Nella prima categoria rientrano gli editor più potenti, sofisticati e costosi: autentici strumenti professionali per lo sviluppo Web.

Il vantaggio di questi strumenti è che l'autore può impostare e modificare rapidamente l'intera struttura o interfaccia grafica di un sito con poche semplici operazioni. I collegamenti, gli sfondi e altri elementi comuni a tutte le pagine del sito vengono aggiornati automaticamente dal programma.

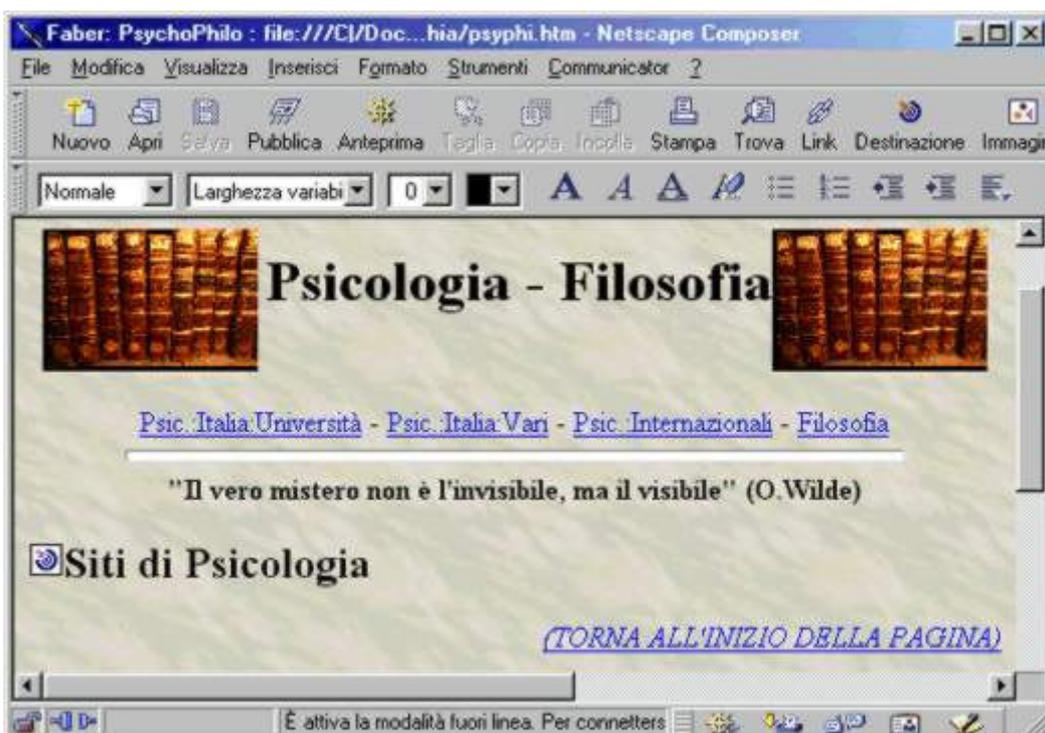
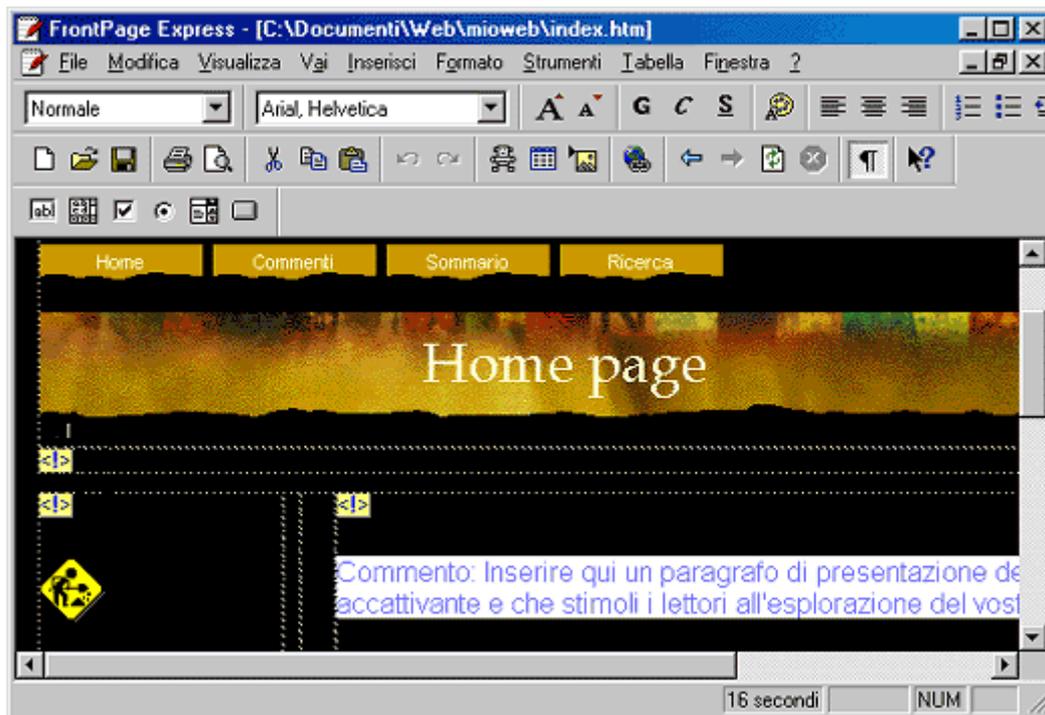
Tra questo primo gruppo di editor si possono citare per esempio **Microsoft FrontPage** e **Macromedia DreamWeaver**:



2. Nella seconda categoria rientrano editor più limitati ma pur sempre vantaggiosi.

Gli editor che operano a livello di singola pagina non possiedono strumenti per la gestione automatizzata di un intero sito. Tuttavia essi consentono ugualmente di creare e modificare le pagine in modo immediato e intuitivo, senza ricorrere alla stesura del codice HTML. Si tratta di strumenti affidabili ma di livello amatoriale più che professionale, e come tali sono per lo più gratuiti.

Tra i più diffusi vi sono **Microsoft FrontPage Express** e **Netscape Composer**.



Allora perché è importante imparare il linguaggio HTML?

Se gli editor Web offrono tanti vantaggi, verrebbe spontaneo chiedersi perché abbia ancora senso studiare il linguaggio HTML. Ecco alcune delle risposte possibili.

- Quasi tutti gli editor Web non generano un codice HTML perfettamente **ottimizzato**, ossia essenziale e rispondente agli standard internazionali. Spesso un editor fa ricorso (in modo automatico e invisibile per l'utente) a istruzioni proprietarie, non standard, che possono rivelarsi controproducenti in determinate circostanze di visualizzazione.
- Conoscere il linguaggio HTML consente di **utilizzare ancora meglio un editor**, poiché

si è in grado di comprenderne a fondo il funzionamento e di ovviare a eventuali dubbi o incongruenze.

- Tutti i più abili progettisti Web ritengono doveroso conoscere bene il linguaggio HTML, perché solo così si è in grado, dopo aver abbozzato una serie di pagine con un editor, **apportare a mano tutti i ritocchi necessari** per raffinare e perfezionare ulteriormente i risultati.

Qualunque sia lo strumento prescelto, prima di mettersi all'opera conviene formulare un piano di lavoro organico e completo.

Il **metodo** corretto comprende le seguenti macro-fasi:

1. schema/layout della pagina o del sito: abbozzo dell'idea fondamentale e del tipo di contenuti;
2. progettazione della struttura (numero di pagine, gerarchia, collegamenti, inserti multimediali);
3. progettazione dell'interfaccia (gabbia grafica della pagina-tipo, disposizione di menu e collegamenti sullo schermo, scelta di caratteri e colori);
4. realizzazione del/i documento/i HTML;
5. verifica del funzionamento (off-line);
6. pubblicazione on-line.

Poiché questo modulo è dedicato strettamente al linguaggio HTML, non ci si sofferma ulteriormente sugli aspetti strategici. Da qui in avanti si prende in considerazione soltanto la **fase esecutiva** (realizzazione dei documenti HTML), utilizzando come strumento un semplice **editor di testo** per la scrittura manuale del codice.

Per chi desidera approfondire l'argomento, analizzando metodi e fasi di lavoro, nonché i dettagli circa gli editor HTML professionali, è disponibile il modulo "**Come si crea un sito Web**".

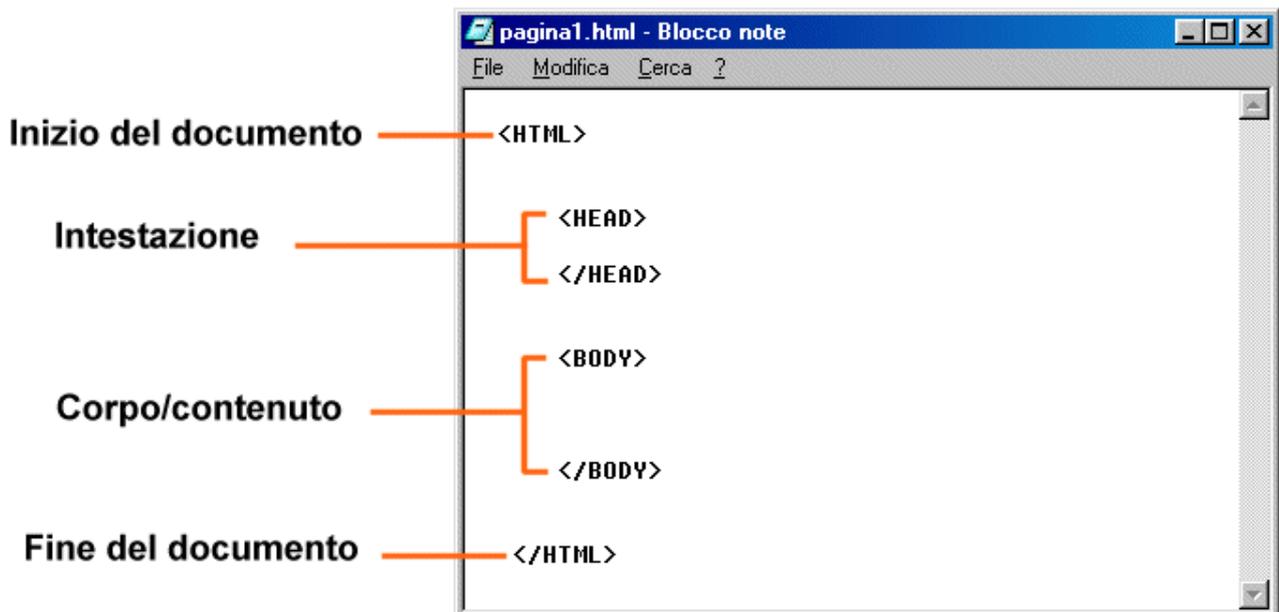
Sezione 2. Creazione, modifica, salvataggio dei file

Creazione della struttura-base

La struttura essenziale di ogni pagina Web comprende alcuni elementi costanti, che ne rappresentano (per così dire) l'ossatura.

- All'**inizio** e alla **fine** del documento si trovano i tag **<HTML>** e **</HTML>**: essi indicano al browser che tutto ciò che è compreso nel mezzo è una pagina Web. Tra questi due tag, si inseriscono due sezioni del documento, contraddistinte da appositi tag.
- La sezione compresa tra **<HEAD>** e **</HEAD>** è l'**intestazione** del documento. Qui si possono facoltativamente inserire informazioni che non vengono visualizzate nella pagina (una volta caricata nel browser), ma che ne descrivono alcune caratteristiche: titolo, autore, parole-chiave e così via. Queste informazioni sono utili per la gestione della pagina da parte del browser dei visitatori e da parte dei motori di ricerca Web.
- La sezione compresa tra **<BODY>** e **</BODY>** è il **corpo** del documento: qui si deve inserire il **contenuto da visualizzare** nella pagina (testo, collegamenti a immagini o indirizzi Web e via dicendo).

Figura 1



Nella figura qui sopra i tag sono ben distanziati e allineati allo scopo di facilitarne la lettura.

Tuttavia, è importante sottolineare che gli **a capo** e gli **allineamenti** impostati manualmente all'interno del codice HTML **non** hanno alcuna influenza sull'aspetto della pagina.

Teoricamente, i tag e il contenuto della pagina potrebbero anche essere scritti tutti di seguito, su un'unica lunga riga.

Per far sì che, durante la visualizzazione nel browser, una frase vada a capo o segua un certo tipo di allineamento si devono inserire nel codice appositi tag (v. sezione 8).

Inoltre, i tag visibili nella pagina precedente sono scritti in **caratteri maiuscoli**: non è un obbligo, potrebbero anche essere scritti minuscoli. Si tratta di un accorgimento per renderli più immediatamente evidenti quando si ritocca il codice HTML della pagina.

Caccia all'errore

Osservare il seguente codice HTML: dovrebbe generare una semplice pagina Web, ma contiene errori tali per cui non può funzionare.

Cosa c'è di sbagliato?

```
</HTML>

  </HEAD>

<HEAD>

  </BODY>

<BODY>

<HTML>
```

Individua la risposta corretta:

- I tag iniziali e finali sono tutti invertiti. Dovrebbe sempre apparire prima il tag senza barra (per esempio: <HTML>) e poi quello con la barra (esempio: </HTML>)
- I tag BODY dovrebbero essere i più "esterni" e contenere tutti gli altri.
- I tag HEAD dovrebbero essere i più "esterni" e contenere tutti gli altri

Nota

La risposta giusta è la prima.

-
1. Il primo passo per creare una pagina Web, perciò, consiste nell'aprire il proprio editor di testo (come il Blocco note di Windows) e scrivere uno dopo l'altro i tag sopra citati.
 2. A questo punto **la pagina è formalmente pronta** e può essere salvata. Naturalmente, se in seguito la si caricasse nel browser, apparirebbe totalmente vuota.
 3. Perciò, prima di salvarla, si può inserire una semplice frase nel corpo del documento:

```
<HTML>

  <HEAD>

  </HEAD>

  <BODY>

    Questa è una pagina Web.

  </BODY>

</HTML>
```

In realtà, il codice di base di ogni pagina Web può contenere altri elementi (tag), che non sono strettamente indispensabili: a volte utili, a volte superflui.

Altri tag strutturali e "pulizia" del codice HTML

Come specificato nella sezione, gli unici tag obbligatori, che costituiscono la struttura di base di una pagina Web, sono:

<HTML> ... </HTML>

<HEAD> ... </HEAD>

<BODY> ... </BODY>

Tutti gli eventuali altri tag sono inseriti liberamente dall'autore per determinare il contenuto e l'aspetto della pagina.

Tuttavia, per maggiore precisione, occorrerebbe citare altri tag generici che possono far parte della struttura di una qualsiasi pagina Web.

Tag strutturali e/o opzionali

Nella struttura di base di una pagina HTML possono essere inseriti altri tag, in aggiunta a quelli essenziali (HTML, HEAD e BODY).

Tra questi tag aggiuntivi, a titolo di esempio, si possono citare i seguenti:

- **<!DOCTYPE "attributi">**: questo tag, da inserire all'inizio del file (prima ancora di **<HTML> ... </HTML>**) serve a dichiarare la versione di HTML su cui il documento è basato.
Per esempio: **<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2//EN">** (questo tag indica che la pagina corrisponde allo standard HTML 3.2). La sintassi completa di questo tag è in effetti complessa, e non è il caso di approfondirla in questa sede.
- **<!-- commento -->**: questo tag, che può essere inserito in qualsiasi punto del documento, consente di commentare alcune parti del codice della pagina: il testo del commento, compreso tra **<!--** e **-->**, non viene visualizzato dai browser.
Per esempio: **<!-- Questo è un commento -->**

In generale, nessuno dei tag di questo genere è obbligatorio: si tratta di opzioni che i progettisti Web possono sfruttare a scopo di promemoria o per facilitare la gestione delle pagine quando il lavoro è svolto in équipe.

Attenzione!

Questa regola prevede alcune significative eccezioni: per esempio, in particolari circostanze il tag **<BODY>** viene sostituito dal tag **<FRAMESET>**. Alcune di queste eccezioni verranno presentate nel seguito del modulo (v. per esempio le sezioni 19 e 20 per **<FRAMESET>**).

Il concetto di "pulizia" del codice HTML

Quando nel codice HTML si inseriscono elementi aggiuntivi come:

- tag opzionali;
- commenti;
- spaziature supplementari tra i tag (per rendere più "leggibile" il codice),

si deve comunque fare attenzione a non accrescere esageratamente e inutilmente le dimensioni della pagina.

Infatti, ogni carattere inserito nel codice (compresi gli spazi vuoti) rappresenta un byte, che concorre ad aumentare le dimensioni complessive del file HTML.

Poiché le dimensioni di un file incidono sensibilmente sui tempi di trasmissione, caricamento e visualizzazione nel browser, è bene mantenere, entro limiti ragionevoli, l'ingombro di ogni pagina.

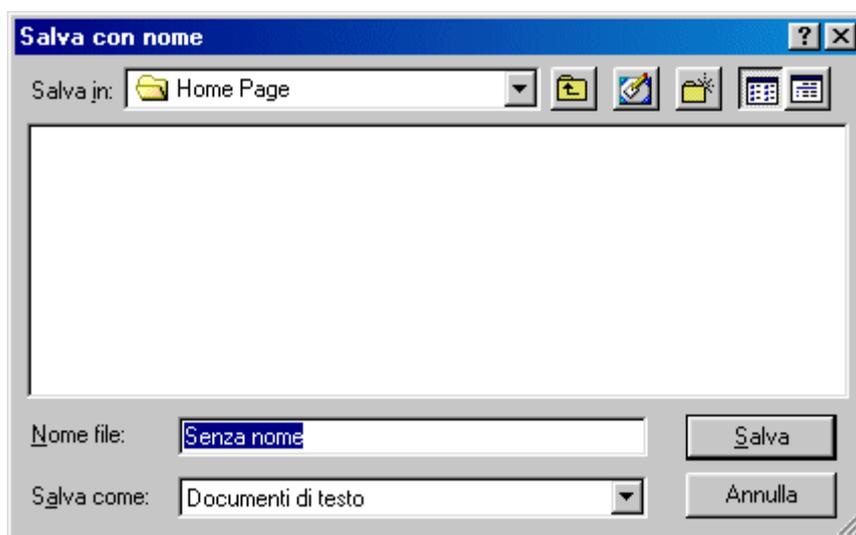
Se da un lato è vero che sono soprattutto i file delle immagini a pesare sui tempi di caricamento, dall'altro sarebbe consigliabile ripulire anche il codice HTML di ogni singola pagina, eliminando tutti i tag e i caratteri non necessari.

Ecco dunque un altro motivo per cui vale la pena conoscere bene il linguaggio HTML, anche qualora si utilizzi un editor WYSIWYG per costruire pagine e siti. Tutti gli autori Web più smaliziati sanno come revisionare il codice riga per riga e alleggerirlo dagli elementi inessenziali.

Salvataggio

Per salvare la pagina HTML si devono rispettare alcune condizioni. Ecco la procedura corretta:

1. aprire il menu **File** del Blocco note e fare clic su **Salva con nome**;
2. nella finestra di salvataggio, scegliere innanzitutto la posizione (disco e cartella) in cui salvare il file;



3. nella stessa finestra, assegnare un nome al file, scrivendolo nella casella **Nome file**: ad esempio, **pagina1.html**; è necessario specificare (scrivendolo per esteso) il suffisso .html o .htm;

Nome file:

4. quindi, in corrispondenza dell'opzione **Salva come** (che specifica il tipo di file), occorre selezionare la voce **Tutti i file (*.*)**;

Salva come:

5. infine, si può fare clic sul pulsante **Salva**.

Alcune raccomandazioni e osservazioni di ordine pratico:

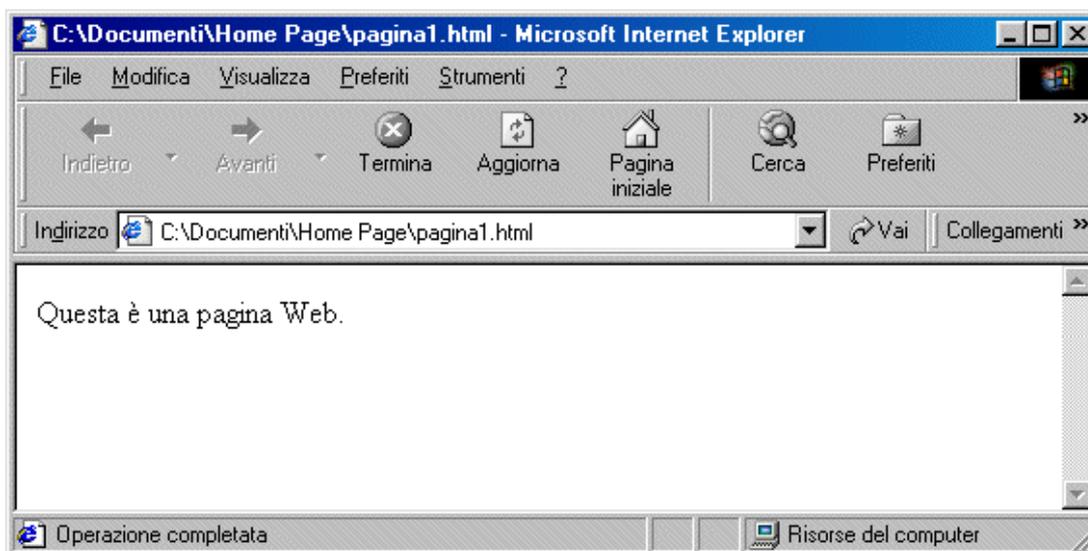
- la prima pagina di un sito di solito viene chiamata **index.html** oppure **home.html**: in questo modo essa può essere più facilmente riconosciuta e gestita dal server (il computer su cui viene pubblicata);
- quando si assegna un nome al file, occorre fare attenzione alla differenza tra **maiuscole e minuscole**: alcuni server Web sono sensibili a tale differenza; come abitudine, converrebbe utilizzare sempre e soltanto i caratteri minuscoli per il nome e il suffisso dei file;
- attenzione alla **posizione** (disco e cartella, comprese eventuali sottocartelle) in cui si

salvano le pagine HTML e i file di immagini: la posizione dei file è di fondamentale importanza al momento di impostare i rispettivi collegamenti ipertestuali (v. sezioni 5 e 13); per ora, sarebbe consigliabile salvare le pagine in una stessa cartella (per motivi di semplicità).

Apertura e modifiche successive

Dopo aver salvato la pagina, è possibile aprirla con il browser per verificarne l'aspetto, in questo modo:

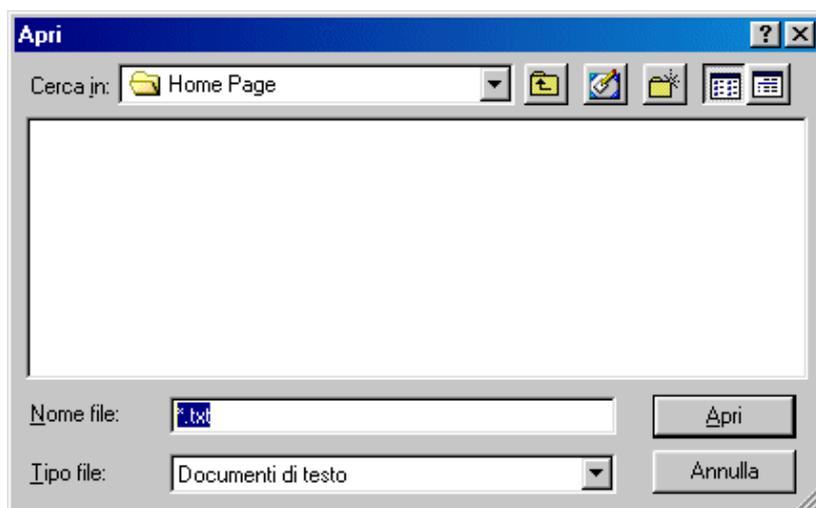
1. sfogliare i dischi e le cartelle (mediante Esplora risorse o uno strumento equivalente), fino a individuare il file;
2. fare doppio clic sull'icona del file;
3. la pagina viene quindi caricata nel browser.



Se invece si desidera riaprire la pagina con l'editor, allo scopo di modificarne il codice HTML, si può procedere in vari modi.

Il metodo più "classico" è il seguente:

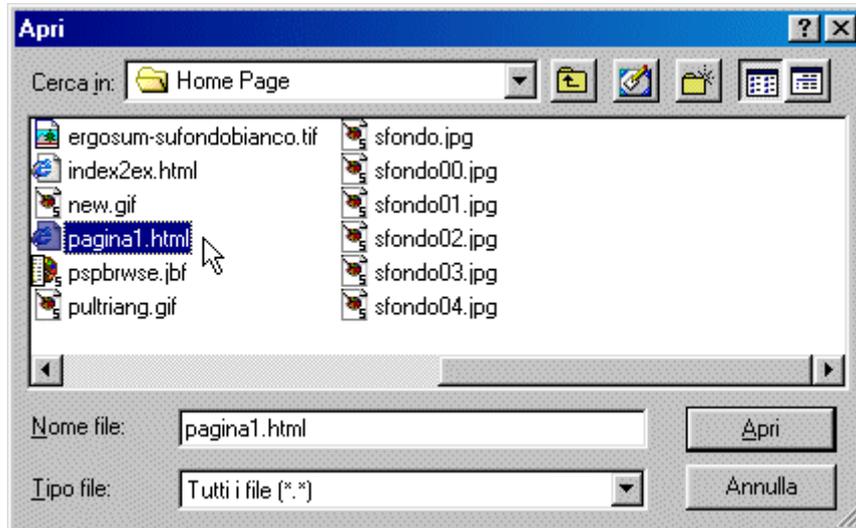
1. lanciare il Blocco note e, nella sua finestra, selezionare il comando **File/Apri**;
2. nella finestra per l'apertura dei file, selezionare la posizione (disco e cartella) in cui si trova la pagina HTML;



3. in corrispondenza dell'opzione **Tipo file** selezionare la voce **Tutti i file (*.*)**: in caso contrario apparirebbero in elenco solo i file di testo (TXT);



4. selezionare l'icona della pagina da aprire e fare clic sul pulsante **Apri**.



Se si trova scomodo questo metodo, è possibile ricorrere a qualche scorciatoia.

Scorciatoie per la modifica di una pagina HTML

Per aprire una pagina HTML esistente e modificarla, esistono alcune scorciatoie. Le seguenti indicazioni valgono in particolare per i sistemi Windows:

Per esempio, è possibile:

1. visualizzare la pagina in Internet Explorer;
2. fare clic sul comando **Visualizza/HTML**;



3. la pagina viene automaticamente aperta nell'editor di testo (Blocco note) e il codice può essere modificato.

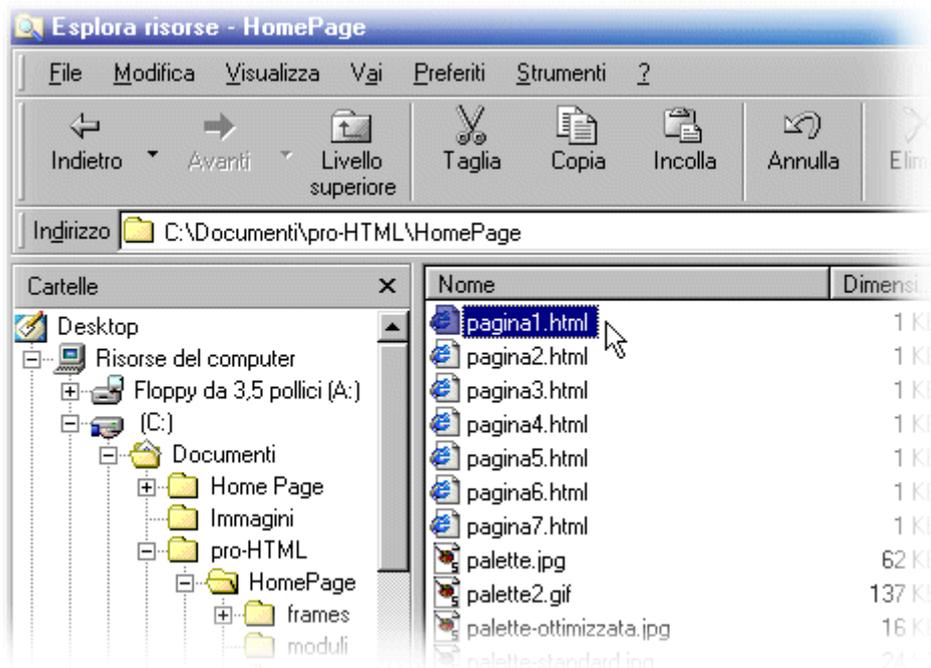
Attenzione

Ovviamente, se la pagina è un file locale (cioè residente sul computer dell'utente) il codice può tranquillamente essere modificato e salvato. Se invece la pagina appartiene a un sito remoto (ossia è visualizzata on-line) le modifiche non avranno alcun effetto sulla versione presente sul

server (se ne potrà tutt'al più salvare una copia modificata sul proprio computer).

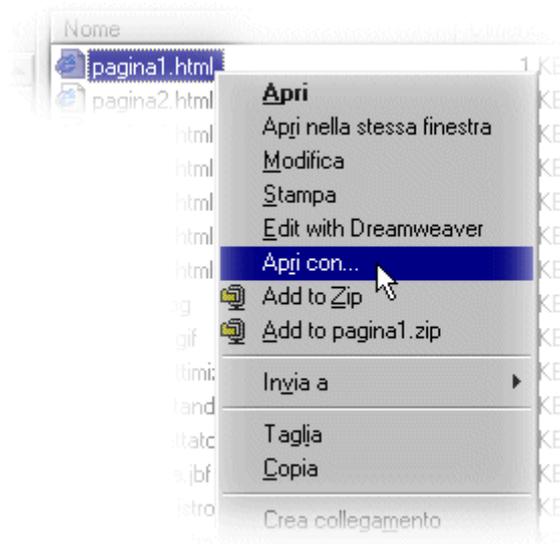
Un'altra scorciatoia, sebbene leggermente più laboriosa, è la seguente:

1. individuare il file HTML nella finestra di Esplora risorse (Gestione risorse in Windows 95) oppure di Risorse del computer, e selezionarne l'icona con un clic;

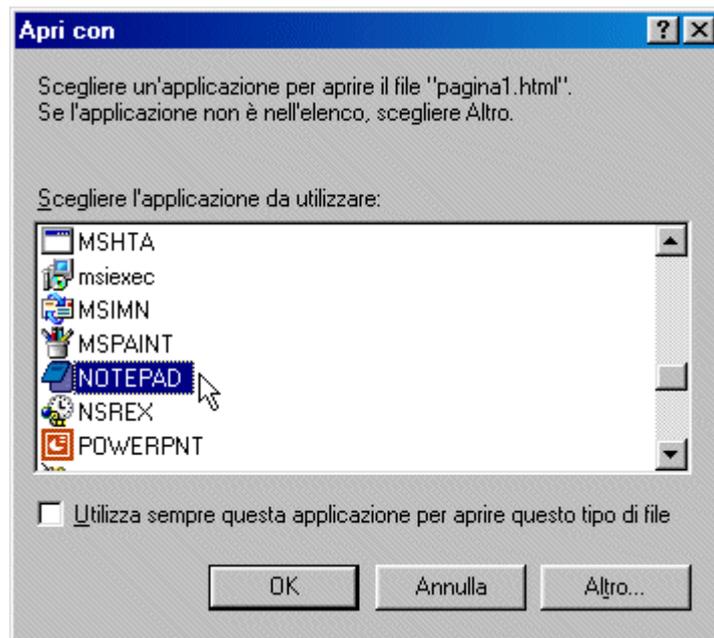


2. quando il file è selezionato (evidenziato), tenere premuto il tasto MAIUSC (SHIFT) e cliccare con il tasto destro del mouse sull'icona del file;

3. nel menu di scelta rapida, apparirà così il comando **Apri con...** (che normalmente non è presente): cliccare su tale comando;



4. nella finestra successiva, selezionare il programma con cui si desidera aprire il file: in questo caso, selezionare il Blocco Note (la voce in elenco è **NOTEPAD**); attenzione: **non** si deve attivare la casella **Utilizza sempre questa applicazione...**, altrimenti i file HTML non sarebbero più associati al browser Web!



5. cliccando su **OK** la pagina viene aperta nella finestra del Blocco Note, dove si può modificare il codice.

Una volta riaperta la pagina nella finestra del Blocco note, si possono apportare tutte le modifiche desiderate.

Successivamente è sufficiente salvare le modifiche (comando **File/Salva**) e chiudere il Blocco note.

Giunti a questo punto, si è in grado di creare una prima pagina HTML: certamente rudimentale, ma dotata di tutti i necessari elementi strutturali.

Sezione 3. L'intestazione

Titolo del documento

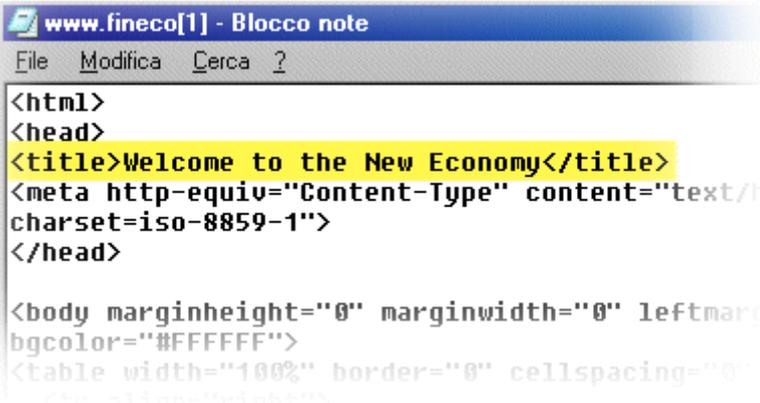
Un primo tag che è possibile collocare nell'intestazione è il tag `<TITLE>`, con cui si definisce il titolo della pagina Web. Il titolo appare sulla barra superiore della finestra del browser quando la pagina viene visualizzata durante la navigazione.



Questo tag richiede un'istruzione di apertura e una di chiusura.
La sintassi corretta è perciò:

`<TITLE> Titolo_della_pagina </TITLE>`

Per esempio, nella figura seguente è evidenziato il tag `<TITLE>`, che determina il titolo della pagina sopra illustrata:

A screenshot of a Notepad window titled "www.fineco[1] - Blocco note". The menu bar shows "File", "Modifica", "Cerca", and "?". The text area contains the following HTML code:

```
<html>
<head>
<title>Welcome to the New Economy</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body marginheight="0" marginwidth="0" leftmargin="0" topmargin="0" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0">
```

The `<title>Welcome to the New Economy</title>` line is highlighted in yellow.

Meta-informazioni

Nell'intestazione si possono specificare, mediante il tag `<META>`, alcune meta-informazioni riguardanti il documento.

Le meta-informazioni sono dati che non vengono visualizzati nella pagina, ma servono al browser e ai motori di ricerca per gestirne i contenuti in modo più efficace. Le meta-informazioni consentono, in particolare, di specificare i seguenti dettagli:

- il nome dell'**autore** della pagina;
- una breve **descrizione** dei contenuti (l'argomento) della pagina;
- un elenco schematico di **parole-chiave** legate ai contenuti.

Il tag <META> non richiede chiusura (ossia va utilizzato da solo), ma l'aggiunta di alcuni attributi e dei rispettivi valori. Questa è la sintassi generica:

<META ATTRIBUTO1=VALORE ATTRIBUTO2=VALORE>

- La voce qui indicata come **ATTRIBUTO1** (che può essere **HTTP-EQUIV** oppure **NAME**) serve a specificare il tipo di informazione: per esempio **NAME=author** significa che ciò che segue (**ATTRIBUTO2**) è il nome dell'autore.
La differenza tra **HTTP-EQUIV** e **NAME** è che il primo attributo è più utile per la gestione della pagina da parte del server Web, mentre il secondo serve a indicare "oggettivamente" il ruolo della meta- informazione. Senza scendere nei dettagli tecnici, si può consigliare di ricorrere abitualmente a **NAME**.
- La voce **ATTRIBUTO2** (che è sempre **CONTENT**) serve a specificare il contenuto concreto della meta-informazione: per esempio, l'effettivo nome dell'autore.

In pratica, per inserire nell'intestazione del documento il nome dell'**autore** si utilizzerà questo tag:

<META NAME=author CONTENT="nome_e_cognome">

Per esempio:

<META NAME=author CONTENT="Fabrizio Comolli">

Attenzione alle **virgolette** che racchiudono il valore di un attributo: a volte sono obbligatorie e a volte no. Per esempio, sono obbligatorie quando il valore è espresso da più parole (come nel caso del nome e cognome).

Poiché in ogni caso sono **sempre** accettabili (non fanno nulla di male anche se non sono necessarie), è meglio utilizzarle in ogni circostanza.

Per inserire nell'intestazione una **descrizione** del documento si utilizzerà questo tag:

<META NAME=description CONTENT="breve_descrizione_dell'argomento_della_pagina">

Per esempio:

<META NAME=description CONTENT="Questa è la prima prova di scrittura di una pagina HTML">

oppure:

<META NAME=description CONTENT="Home page della XYZ S.p.A.: prodotti per ufficio, computer, telefonia">

Questa descrizione appare di solito nell'elenco dei siti trovati, a fianco dell'indirizzo Web della pagina, a chi effettua una ricerca con i motori di ricerca Web. E' uno strumento utile per far capire a colpo d'occhio l'argomento della pagina, e attirarvi così i visitatori.

Infine, si raccomanda di inserire nell'intestazione un elenco di **parole-chiave**, che specifichino l'argomento della pagina in ogni aspetto o variante:

<META NAME=keywords CONTENT="parola1 parola2 parola3">

Per esempio:

<META NAME=keywords CONTENT="ufficio computer telefonia XYZ">

Le parole-chiave vengono utilizzate dai motori di ricerca per catalogare la pagina e segnalarla a chi sta effettuando ricerche su argomenti affini. Esse sono dunque un mezzo di importanza fondamentale per promuovere la visibilità della propria home page e ottenere un'adeguata attenzione da parte del pubblico di Internet.

Ecco alcune considerazioni e consigli circa l'impostazione delle parole-chiave:

- nell'elenco, le parole-chiave possono essere separate da **spazi vuoti** oppure da **virgole** (alcuni motori prediligono la prima tecnica, altri la seconda);
- è bene non esagerare con la **quantità** di parole-chiave: una ventina (al massimo) può essere più che sufficiente (alcuni motori di ricerca ignorano le parole-chiave in eccesso);
- nello stesso tempo, è bene cercare di ampliare il più possibile la **gamma** delle parole-chiave: per esempio, se la pagina Web parla di telefonia, tra le parole-chiave converrebbe specificare "telefonia" ma anche "cellulari", "centralini", "fax" o qualsiasi altro termine pertinente; inoltre, è raccomandabile indicare le parole-chiave **in italiano e anche in inglese** (o in qualunque altra lingua desiderata, se ci si rivolge in particolare a un determinato paese).

Più in generale, si possono riassumere alcuni consigli essenziali per aumentare la reperibilità e la popolarità delle proprie pagine Web.

Come promuovere la "trovabilità" di un sito

Non basta pubblicare sul Web una pagina o un sito dedicati a un certo argomento. È necessario che il proprio potenziale pubblico (il target dei lettori interessati) sappia che la pagina o il sito esistono.

Come si fa a pubblicizzare l'esistenza del proprio sito, o meglio come si fa a promuoverne la visibilità e la reperibilità?

La soluzione più immediata è facilitare la catalogazione del sito da parte dei motori di ricerca. Ciò comporta due operazioni distinte:

1. **impostare in modo opportuno il codice HTML** di ogni pagina, in modo che siano adeguatamente evidenziate e valorizzate le parole-chiave che ne descrivono il contenuto;
2. **segnalare l'indirizzo** del sito ai motori di ricerca.

Impostazione del codice HTML

Come spiegato nella sezione, esistono appositi tag di tipo <META> da inserire nell'intestazione di ogni pagina per specificarne:

1. una sintetica descrizione (con **<META NAME="description">**);
2. un elenco di parole-chiave (con **<META NAME="keywords">**).

Ecco alcune considerazioni e consigli circa l'impostazione delle parole-chiave:

- nell'elenco, le parole-chiave possono essere separate da **spazi vuoti** oppure da **virgole** (alcuni motori prediligono la prima tecnica, altri la seconda);
- è bene non esagerare con la **quantità** di parole-chiave: una ventina (al massimo) può essere più che sufficiente; alcuni motori di ricerca ignorano le parole-chiave in eccesso;
- nello stesso tempo, è bene cercare di ampliare il più possibile la **gamma** delle parole-chiave: per esempio, se la pagina Web parla di telefonia, tra le parole-chiave converrebbe specificare telefonia ma anche cellulari, centralini, fax o qualsiasi altro termine pertinente; inoltre, è raccomandabile indicare le parole-chiave **in italiano ma anche in inglese** (o in qualunque altra lingua desiderata).

Attenzione

Oltre ai tag **<META>** dell'intestazione, si hanno a disposizione altre opportunità per ribadire le parole-chiave associate alla pagina.

Con un minimo di malizia, si possono ripetere le parole-chiave più importanti in altri elementi della pagina identificati da appositi tag, come:

- il titolo del documento HTML (tag **<TITLE>**, illustrato in questa stessa sezione);
- i titoli nel corpo del testo (v. sezione 10);
- le immagini e le relative didascalie (v. sezione 13).

Molti motori di ricerca, infatti, analizzano anche tali elementi al fine di catalogare correttamente l'argomento di una pagina.

Segnalazione del sito ai motori di ricerca

Dopo aver impostato adeguatamente il codice HTML delle pagine e averle pubblicate on-line, è opportuno segnalare ai principali motori di ricerca l'esistenza del sito, il suo indirizzo e l'argomento trattato.

Tutti i motori di ricerca prevedono questa possibilità: basta cercare nella loro home page un collegamento chiamato **Add URL, Aggiungi URL, Segnala un sito** o qualcosa del genere. Anche alcuni siti indipendenti offrono un servizio (a volte gratuito, a volte a pagamento) per segnalare contemporaneamente lo stesso indirizzo Web a molteplici motori.

A titolo di esempio, si può visitare l'ottimo sito www.motoridiricerca.it.

Altri contenuti dell'intestazione

Nell'intestazione si possono inserire anche altri elementi, più o meno superflui o utili in base al tipo di pagina Web.

Per esempio, si può automatizzare la frequenza di **aggiornamento** della pagina nel browser con il seguente tag:

```
<META HTTP-EQUIV=refresh CONTENT="frequenza_in_secondi">
```

In questo modo, la visualizzazione della pagina nel browser viene aggiornata automaticamente: la pagina viene ricaricata dopo il numero di secondi specificato, senza che chi naviga debba cliccare volontariamente sul pulsante **Aggiorna** o **Ricarica**. Si tratta di un accorgimento utilissimo per le pagine Web di carattere informativo, il cui contenuto cambia rapidamente: notiziari, indici borsistici e così via.

Il tag **<META>** può essere utilizzato anche per definire (sebbene con un valore meramente formale) il **copyright** della pagina:

```
<META NAME=copyright CONTENT="Copyright 2000, nome_del_titolare">
```

Altri ancora sono i possibili contenuti dell'intestazione: nel seguito del modulo ne verranno presentati alcuni, tra cui il rimando a un foglio di stile CSS esterno tramite il tag **STYLE** (v. sezioni 24 e 25).

Per il momento, ci si può fermare qui.

Nota: errori da evitare

Prima di proseguire, un'ultima raccomandazione.

Come è stato detto all'inizio della sezione, il contenuto dell'intestazione è facoltativo. Se però si decide di sfruttare lo spazio dell'intestazione in uno dei modi descritti, definendone alcuni contenuti, è importante rispettare la **corretta sintassi** HTML. E' facile incorrere in errori banali, che sarebbe bene evitare

Errori da evitare nell'intestazione

Oltre a raccomandare in linea di principio il rispetto della sintassi HTML, per quanto riguarda l'intestazione va sottolineato che si riscontrano spesso errori banali.

Quasi sempre gli errori hanno a che vedere con il titolo del documento.

- È caldamente consigliabile **specificare il titolo** della pagina, inserendo nell'intestazione un tag <TITLE>: senza il tag <TITLE>, una pagina risulta piuttosto anonima e poco professionale, specie se il visitatore la memorizza nel suo elenco di siti preferiti (apparirebbe l'indirizzo Web anziché un nome intuitivo).
- Il titolo deve essere **ben impostato**: non troppo breve ed ermetico (non va bene una sigla, meglio una breve frase descrittiva), ma neanche troppo lungo (considerando che deve apparire nella barra superiore della finestra del browser).
- Non bisogna commettere una classica svista (più comune di quanto si possa credere), inserendo per disattenzione più tag <TITLE> nella stessa intestazione. E' ammesso **un solo titolo** per ogni documento.

Quest'ultimo punto merita un breve commento.

A volte, infatti, non si tratta di una svista ma di un maldestro tentativo di sfruttare caratteristiche ormai sorpassate dei browser.

Una delle prime versioni del browser Netscape (per esattezza, Netscape 1.0) manifestava un comportamento particolare: in presenza di molteplici tag <TITLE> in uno stesso documento, mostrava i vari titoli in rapida successione, l'uno dopo l'altro.

Perciò era possibile, all'epoca, sfruttare questa anomalia e scrivere un codice di questo genere:

```
<HEAD>
<TITLE> Q </TITLE>
<TITLE> Qu </TITLE>
<TITLE> Que </TITLE>
<TITLE> Ques </TITLE>
<TITLE> Quest </TITLE>
<TITLE> Questa </TITLE>
<TITLE> Questa è </TITLE>
<TITLE> Questa è la </TITLE>
<TITLE> Questa è la mia </TITLE>
<TITLE> Questa è la mia home page </TITLE>
```

Il risultato era che il titolo completo "Questa è la mia home page" sarebbe stato visualizzato progressivamente nella barra della finestra di Netscape.

Attenzione!

Tutto ciò non è più valido con i browser più recenti: una porzione di codice come quella sopra riportata rappresenta soltanto una inutile violazione della corretta sintassi HTML.

I browser attuali reagirebbero in modo imprevedibile: Netscape 4.0 visualizzerebbe solo il primo titolo, Internet Explorer 3.01 solo l'ultimo e così via.

Si raccomanda perciò di attenersi scrupolosamente a quanto indicato nelle sezioni del modulo.

Sezione 4. Corpo del documento

Sfondo

Lo sfondo della pagina può essere personalizzato, impostandone il colore o inserendo un'immagine.

È anche possibile definire un suono che accompagni in sottofondo la visualizzazione della pagina, sebbene sussistano alcune limitazioni.

- Il colore di sfondo va impostato con l'attributo BGCOLOR, secondo la seguente sintassi:

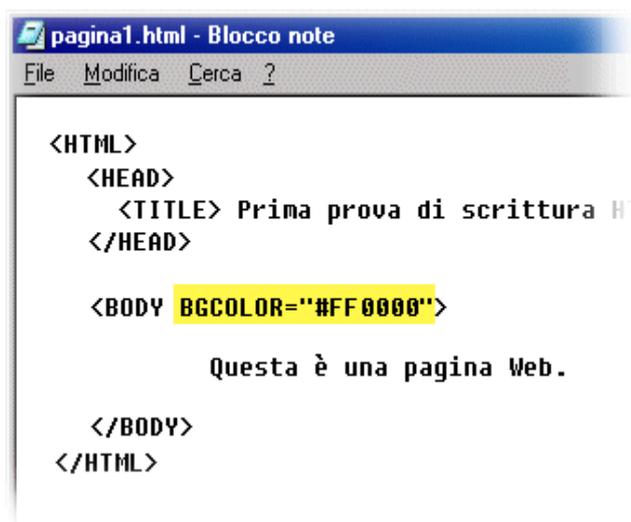
```
<BODY BGCOLOR="#codice_del_colore">
```

Il codice del colore deve essere specificato mediante sei caratteri secondo il formato **esadecimale**, ossia con le cifre da 0 a 9 e/o le lettere dalla A alla F.

Per esempio, il codice FFFFFFFF equivale al bianco, il codice 000000 al nero, il codice FF0000 al rosso e così via.

Per impostare uno sfondo rosso, perciò, si dovrà scrivere questa istruzione:

```
<BODY BGCOLOR="#FF0000">
```



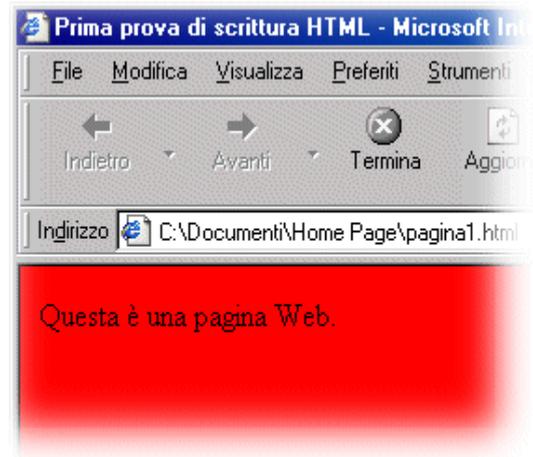
```
pagina1.html - Blocco note
File  Modifica  Cerca  ?

<HTML>
  <HEAD>
    <TITLE> Prima prova di scrittura HT
  </HEAD>

  <BODY BGCOLOR="#FF0000">

    Questa è una pagina Web.

  </BODY>
</HTML>
```



I colori nel linguaggio HTML

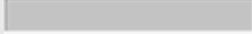
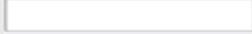
Questa scheda anticipa alcune delle nozioni che verranno presentate in modo più approfondito nella sezione 11.

In linea di principio, si tenga presente che ai fini della codifica HTML:

- ogni colore viene definito mediante una **combinazione di componenti (colori primari): rosso (Red), verde (Green), blu (Blue)**;
- ciascuna componente può assumere una proporzione variabile, così da generare un ampio numero di tonalità cromatiche intermedie; il valore di ciascuna componente, in un dato colore, viene espresso mediante una coppia di caratteri (lettere e/o numeri) secondo il **metodo di codifica esadecimale**, dove il valore minimo (nullo) è "00" e il valore massimo è "FF";
- di conseguenza, ogni colore è rappresentato dalla **formula generica #RRGGBB** (dove RR, GG e BB sono i valori delle tre componenti: rispettivamente Red, Green e Blue); per esempio, il bianco corrisponde al codice #FFFFFF, il nero a #000000, il rosso a #FF0000, il

verde a #00FF00, il blu a #0000FF, il viola a #FF00FF e così via.

Come sintetico promemoria (in attesa della sezione 11), ecco una tabella con i codici dei 16 colori più elementari:

Colore (campione)	Nome (in inglese)	Codice esadecimale RGB
	aqua	00FFFF
	black	000000
	blue	0000FF
	fuchsia	FF00FF
	gray	808080
	green	008000
	lime	00FF00
	maroon	800000
	navy	000080
	olive	808000
	purple	800080
	red	FF0000
	silver	C0C0C0
	teal	008080
	white	FFFFFF
	yellow	FFFF00

- Al posto di un colore omogeneo, lo sfondo della pagina Web potrebbe contenere un'immagine, in formato GIF o JPEG (v. anche sezione 12 e seguenti). Per impostare un'immagine di sfondo si utilizza l'attributo BACKGROUND, secondo la seguente sintassi:

```
<BODY BACKGROUND="percorso_e_nome_del_file">
```

Più avanti nel modulo (v. sezioni 5 e 13) si troveranno dettagli e consigli circa l'impostazione dei percorsi (posizione dei file).

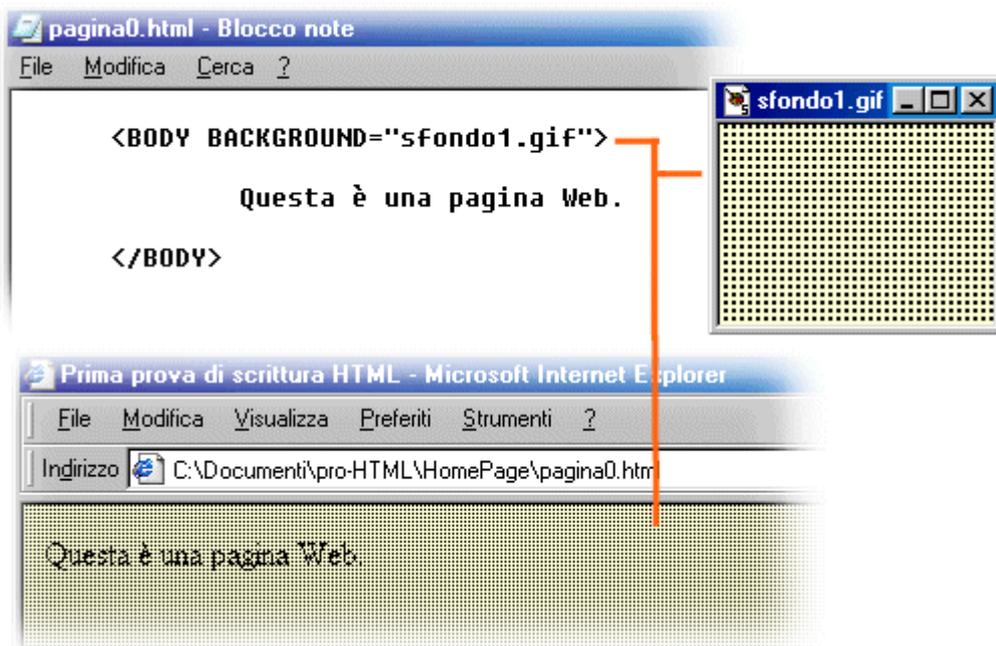
Per semplificare, si supponga per ora che il file dell'immagine si trovi nella stessa cartella della pagina HTML. In questo caso, è sufficiente inserire come valore dell'attributo BACKGROUND soltanto il nome del file. Per esempio:

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

Le immagini di sfondo

Se si inserisce un'immagine come sfondo della pagina Web, tale immagine viene automaticamente replicata dal browser in modo tale da coprire l'intero spazio visibile. L'immagine di sfondo, in altre parole, si comporta come un mattoncino o, meglio ancora, come la tessera di un mosaico.

Per esempio:



Infine, è possibile associare un **suono** alla visualizzazione della pagina. Il file audio, preferibilmente in formato MIDI, può essere eseguito una sola volta, oppure ripetuto per N volte o anche a ciclo continuo (loop). Tuttavia, riguardo ai suoni di sottofondo, bisogna tenere presenti alcune limitazioni e controindicazioni. Si legga la seguente scheda di approfondimento.

Suoni in sottofondo

Sebbene il linguaggio HTML consenta di impostare un suono di sottofondo, che accompagna l'apertura di una pagina, questa opzione è in realtà **sconsigliabile**. I motivi per cui si sconsiglia il ricorso a tale opzione sono sostanzialmente i seguenti:

- Il risultato dell'operazione sarebbe quasi inevitabilmente **povero sul piano qualitativo**; la multimedialità in ambito Web soffre ancora di alcune limitazioni, e viene implementata con modalità più complesse rispetto a quelle consentite dai tag per il suono in sottofondo: di solito si ricorre a tecniche di collegamento esterno per richiamare file audiovisivi salvati in formati particolari (un classico esempio sono i cosiddetti formati di streaming, cioè per la riproduzione in diretta, come quelli RealAudio e RealVideo, oppure le animazioni multimediali in formato Macromedia Flash).
- I tag per inserire i suoni in sottofondo non sono standard: occorre adottarne uno per Internet Explorer e uno per Netscape. Si pongono dunque **problemi di compatibilità**, e tali (pur secondarie) difficoltà non sono giustificate dal livello qualitativo dei risultati prodotti (vedi il punto precedente).

Ad ogni modo, per completezza, ecco le nozioni essenziali.

Innanzitutto, per questa finalità sono supportati due **formati audio** principali:

- il formato **.WAV (Wave)**, tipico dell'ambiente Windows; i file .WAV possono essere di discreta qualità sonora, ma in compenso hanno dimensioni rilevanti, tali da rallentare notevolmente il caricamento;
- il formato **.MID (Midi)**, di diffusione pressoché universale; i file .MID sono generalmente di piccole dimensioni (dunque abbastanza rapidi da caricare), ma anche di bassa qualità sonora (ciò dipende, in realtà, anche dal tipo di scheda audio di cui è dotato ogni utente finale).

Affinché il suono in sottofondo venga caricato da un utente dotato di Internet Explorer occorre inserire nel codice della pagina questo tag:

<BGSOUND SRC="nome_file" LOOP="numero_di_ripetizioni">

Invece, affinché il suono venga caricato da chi utilizza il browser Netscape, il tag è:

<EMBED SRC="nome_file" LOOP="numero_di_ripetizioni/">

In entrambi i casi, si deve specificare:

- il nome (ed eventualmente il percorso) del file audio, tramite l'attributo SRC; per esempio: **SRC="volare.mid"**;
- il numero di ripetizioni desiderate, tramite l'attributo LOOP; per esempio: **LOOP="3"**; se si indica **LOOP="-1"** o **LOOP="infinite"**, il brano verrà riprodotto all'infinito (fino a che l'utente non abbandona la pagina).

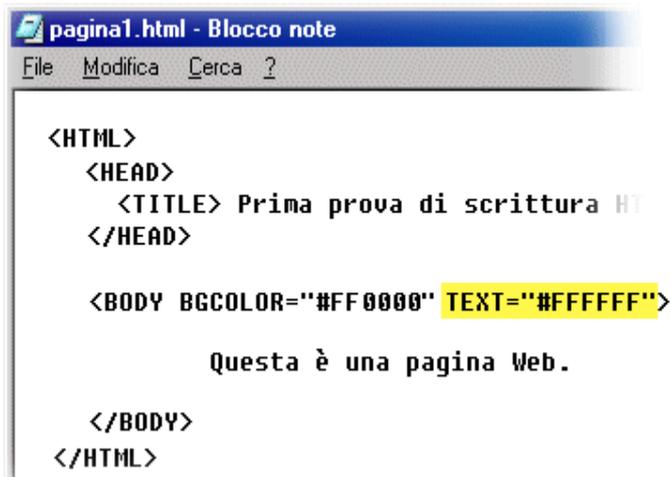
Colore del testo

Si può impostare a priori il colore del testo della pagina, mediante l'attributo TEXT:

<BODY TEXT="#codice_del_colore">

Valgono le stesse considerazioni già espresse per il colore di sfondo (v. anche sezione 11). Per esempio, per far sì che tutto il testo del documento risulti bianco, si può utilizzare questa istruzione:

<BODY TEXT="#000000">

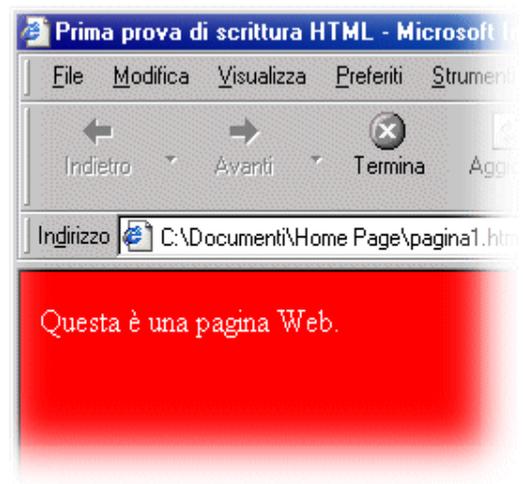


```
<HTML>
<HEAD>
  <TITLE> Prima prova di scrittura HT
</HEAD>

<BODY BGCOLOR="#FF0000" TEXT="#FFFFFF">

  Questa è una pagina Web.

</BODY>
</HTML>
```



Questa impostazione può essere in seguito aggirata, definendo un colore specifico per una certa porzione di testo, mediante il tag **** (v. sezione 6 e successive).

Bisogna sottolineare che le **altre caratteristiche** del testo (come il tipo di carattere, le dimensioni, gli allineamenti eccetera) **non** possono essere definite qui, come attributi generali della sezione **<BODY>**. Per questo genere di impostazioni si deve ricorrere:

- ai tag per la formattazione di singole porzioni di testo, come ****, **<P>** o **<DIV>** (v. sezione 6 e successive);
- ai fogli di stile (v. Sezioni 24 e 25).

Colore dei collegamenti

Un'ultima caratteristica generale, da definire tramite attributi del tag <BODY>, è il colore dei collegamenti ipertestuali (v. anche sezione 5). Si possono definire tre colori:

- un colore per i collegamenti ipertestuali **in condizioni normali**: si usa l'attributo **LINK="#codice_del_colore"**;
- un colore per i collegamenti **attivi**, che appare nel momento in cui vi si clicca sopra: si usa l'attributo **ALINK="#codice_del_colore"**;
- un colore per i collegamenti **già visitati**: si usa l'attributo **VLINK="#codice_del_colore"**.

In pratica, per personalizzare il colore dei collegamenti ipertestuali si possono abbinare questi tre attributi:

<BODY LINK="#colore1" ALINK="#colore2" VLINK="#colore3">

Ecco, a titolo di esempio, una comparazione tra i colori standard e alcuni colori personalizzati. Si noti che, in base alle impostazioni standard, i collegamenti attivi (nel momento del clic) non cambiano colore.

Colori standard



Collegamento <LINK>

Collegamento attivo <ALINK>

Collegamento visitato <VLINK>

Colori personalizzati



Naturalmente, per scegliere opportuni abbinamenti di colori (sfondo, testo, collegamenti) è bene avere una certa competenza grafica, oppure consultare guide e siti dedicati al Web Design.

In linea di massima, è raccomandabile evitare contrasti troppo marcati, che penalizzerebbero la leggibilità delle pagine.

Sezione 5. Collegamenti ipertestuali

Introduzione

Gli utenti che si sono avvicinati da poco a Internet e al Web restano affascinati soprattutto dagli aspetti multimediali e interattivi dei siti e dei servizi on-line.

Eppure, il fattore che all'origine ha caratterizzato la rivoluzione del World Wide Web e ne ha determinato l'irresistibile successo è stato la sua **ipertestualità**: la possibilità, cioè, di sfogliare l'immensa quantità di informazioni disponibili secondo "salti logici", associazioni, percorsi di lettura trasversali e liberi.

Con un clic il lettore/navigatore può saltare da una pagina all'altra, da un argomento all'altro, assecondando flessibilmente i propri interessi senza essere vincolato a una trama predefinita. In termini tecnici, ogni pagina Web è un **nodo** della Rete, che contiene un numero variabile di **collegamenti** ipertestuali verso altri nodi (esterni o interni).

Cosa sono gli ipertesti

Nelle scienze della comunicazione, il termine **testo** indica in generale un qualsiasi messaggio, indipendentemente dalla forma (mezzo o medium) in cui viene diffuso.

In questo senso, un "testo" può essere un documento scritto, un filmato, un brano musicale e via dicendo.

Tutti i testi tradizionali hanno una struttura relativamente rigida: ogni testo ha un inizio e una fine, e la direzione in cui può essere consultato è lineare. Per esempio, un libro può essere sfogliato pagina dopo pagina; un film viene proiettato fotogramma dopo fotogramma e così via.

Se il lettore (o lo spettatore) non rispetta queste rigide regole di consultazione, non riesce a cogliere il significato originario del messaggio.

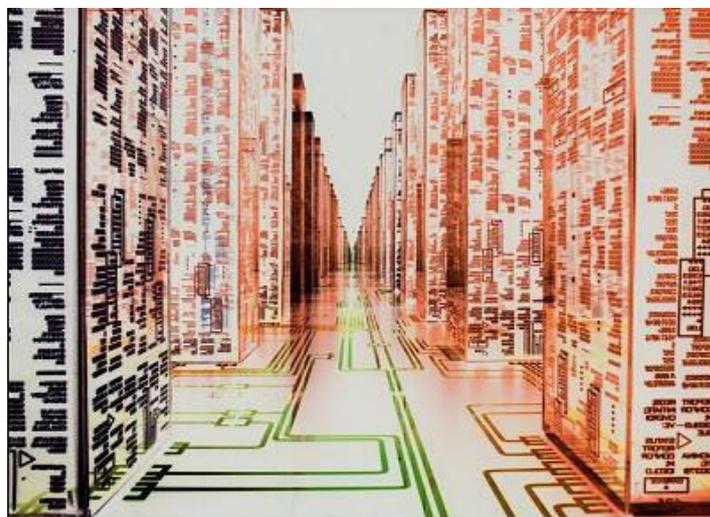
Un **ipertesto** invece è un messaggio o documento privo di struttura, o meglio privo di un ordine di consultazione prefissato.

L'autore dell'ipertesto crea soltanto una serie di:

- **nodi**, cioè nuclei di informazione, blocchi di contenuti comunicativi;
- **collegamenti** che predispongono una rete di passaggi (salti) da un nodo all'altro; si pensi, per esempio, a un'enciclopedia in cui ogni voce può essere correlata a molte altre, in modo "trasversale".

L'utente che consulta un ipertesto è libero di scegliere, in totale (o quasi) autonomia, quali collegamenti seguire caso per caso, quali nodi visualizzare e in quale ordine di successione.

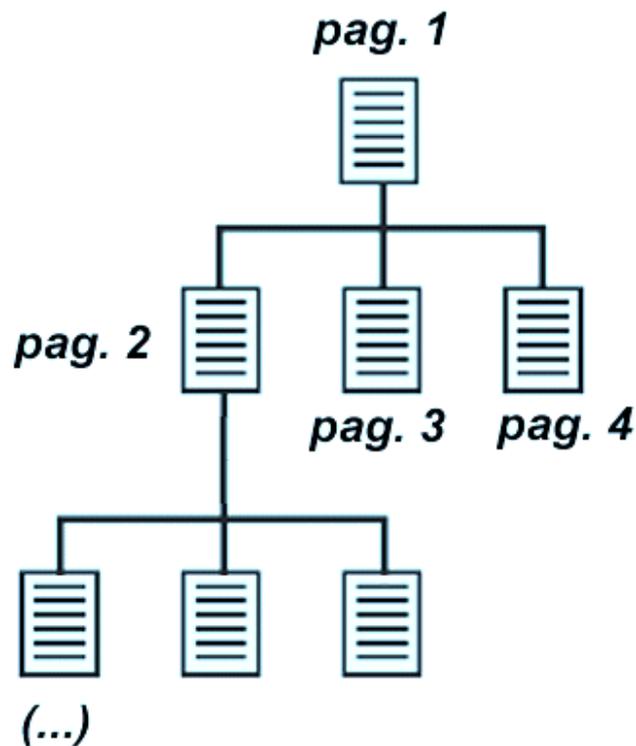
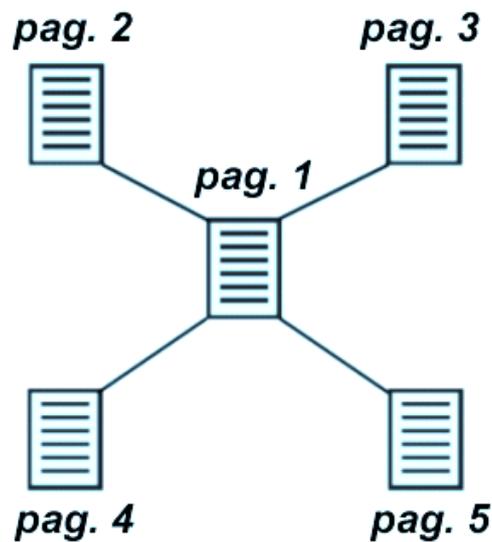
Un ipertesto è dunque un documento ampio e aperto a differenti piani di navigazione: una **rete** di informazioni e rimandi incrociati.



La rete di nodi e collegamenti che costituisce un ipertesto può assumere svariate configurazioni:

- può svilupparsi in senso prevalentemente gerarchico o verticale, come un catalogo di prodotti suddiviso in categorie o un organigramma che descrive la struttura di un'azienda;
- può svilupparsi in modo ampio e multidirezionale (circolare, reticolare), assegnando al lettore il più alto grado di libertà nella navigazione.

In generale, più numerosi e fitti sono i collegamenti incrociati predisposti dall'autore dell'ipertesto, maggiore è la libertà di navigazione/consultazione concessa al lettore. L'unico **svantaggio** è che una libertà di navigazione "troppo" ampia può comportare un certo grado di confusione: se la rete ipertestuale è troppo estesa e troppo fitta, il lettore può fare fatica a costruirsi una mappa mentale (un indice) del documento complessivo.



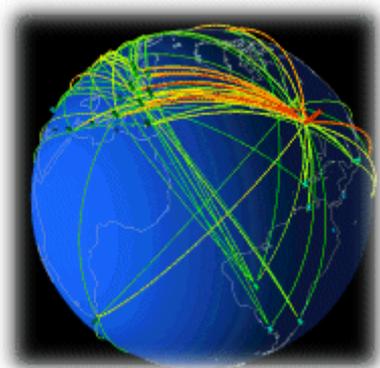
Al di là di questa breve introduzione teorica, bisogna notare che per lungo tempo il concetto di ipertesto è rimasto un progetto puramente teorico. Vi sono alcuni famosi personaggi che hanno anticipato e propugnato tale concetto, in epoche in cui la tecnologia non era ancora in grado di tradurlo in realtà operativa.

1. Negli anni Trenta/Quaranta, **Vannevar Bush** immaginò un apparecchio che sostenesse e facilitasse la memorizzazione e l'archiviazione delle informazioni. Tale apparecchio, chiamato MEMEX (Memory Extender), avrebbe dovuto funzionare in conformità con lo stile della memoria umana, basata sulla categorizzazione gerarchica, sull'associazione di idee e sui riferimenti incrociati.

2. Negli anni Sessanta/Settanta, **Ted Nelson** coniò il termine "ipertesto" e ne approfondì le potenzialità, elaborando un progetto chiamato Xanadu (anch'esso arrestatoosi allo stadio teorico).

3. Solo negli anni Ottanta/Novanta, grazie all'evoluzione dell'**informatica** e della telematica, il concetto di ipertestualità trovò un'attuazione pratica. Il programma HyperCard per Macintosh, gli Help delle applicazioni, le enciclopedie multimediali su CD-ROM sono tutti esempi di ipertesti: documenti interattivi, che consentono all'autore e al lettore di definire (più o meno liberamente) la struttura della comunicazione e la navigazione tra le informazioni.

Grazie a **Tim Berners Lee**, padre del progetto **World Wide Web** (v. sezione 1), anche l'accesso ai contenuti e ai servizi di Internet ha infine assunto una forma ipertestuale.



La navigazione on-line si svolge cliccando su parole e immagini per saltare da un argomento all'altro, da un sito all'altro, con la possibilità di seguire volta per volta infiniti percorsi concettuali.

Progettare un sito Web contenente varie pagine (nodi) correlate tra loro (tramite collegamenti o link), significa appunto sviluppare un documento di tipo ipertestuale.

Principi di funzionamento

I collegamenti ipertestuali vengono creati associando l'indirizzo (percorso/nome file) del nodo di destinazione a un elemento visualizzato nella pagina corrente (stringa di testo o immagine). Questa associazione si basa sull'utilizzo del tag **<A HREF>**, dove **A** sta per "ancora (o ancoraggio)" e **HREF** è l'abbreviazione di **H**ypertext-**R**EFerence ("riferimento ipertestuale"). Ecco la sintassi generica:

**** testo o immagine ****

Il testo o l'immagine diventano così un **elemento "attivo"** (in gergo: cliccabile):

- il puntatore del mouse, passando sopra tale elemento, assume la forma di una manina con il dito puntato;
- mentre il puntatore si trova sopra l'elemento attivo, nella barra di stato del browser (lungo il bordo inferiore della finestra) appare in anteprima l'indirizzo di destinazione;
- facendo clic si salta all'indirizzo di destinazione.

La **destinazione** del collegamento può essere esterna o interna **rispetto alla pagina**

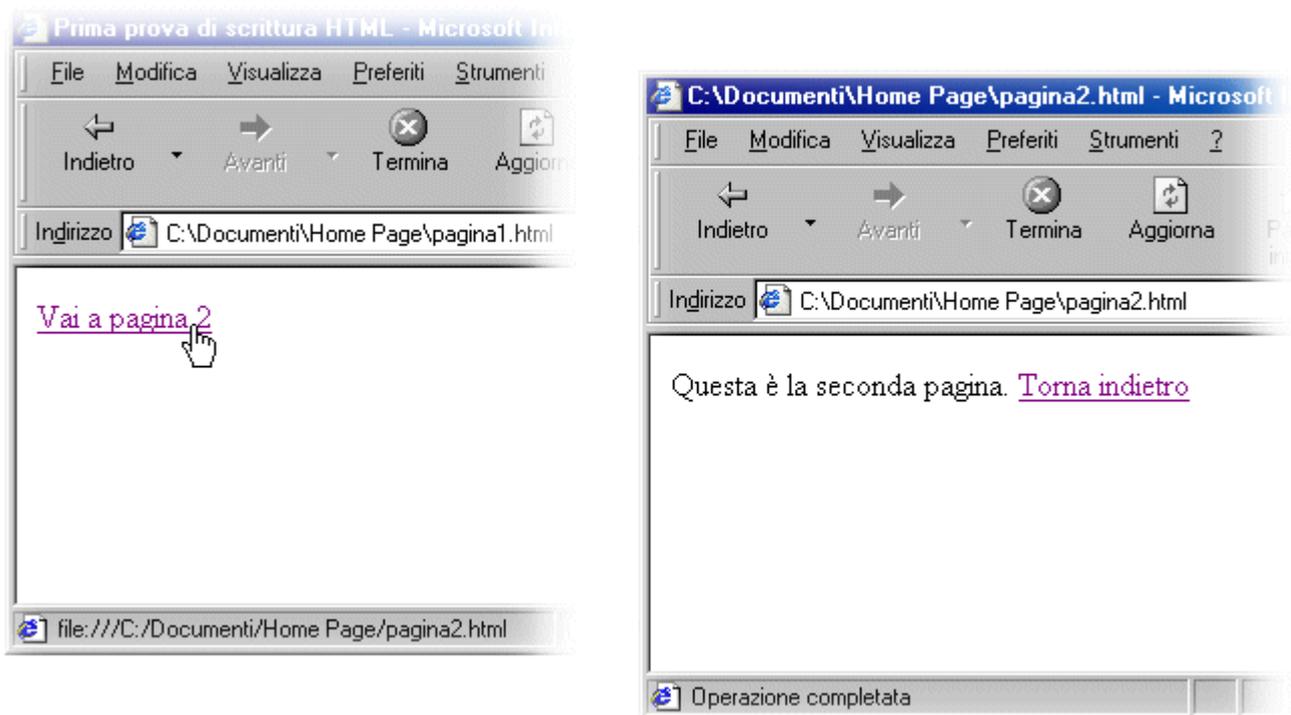
corrente (origine del collegamento):

- **esterna**: un'altra pagina dello stesso sito; un differente sito Web; un indirizzo di posta elettronica; un file da visualizzare (documento, immagine o filmato) oppure da scaricare mediante download;
- **interna**: un segnalibro invisibile (chiamato "ancora"), per saltare istantaneamente da un punto all'altro dello stesso documento.

Collegamenti esterni

Per creare un collegamento verso **un'altra pagina dello stesso sito** la sintassi è di questo tipo:

` Vai a pagina 2 `



Nell'impostazione del tag `<A HREF>` vanno tenuti in considerazione i seguenti aspetti:

- il **nome**: il nome del file di destinazione (nell'esempio: *pagina2.html*) deve essere scritto correttamente, rispettando le maiuscole e minuscole;
- il **percorso**: se il file di destinazione si trova nella stessa posizione (cartella) del file di partenza, è sufficiente indicarne il nome, come nell'esempio sopra riportato; altrimenti, è necessario indicare nel tag il percorso completo oltre al nome del file.

Riferimenti relativi e assoluti

Molto spesso in un tag si fa riferimento a un file esterno alla pagina Web; ecco alcuni dei casi più frequenti:

- quando si imposta un collegamento ipertestuale; per esempio: ` ... `;
- quando si richiama la visualizzazione di un'immagine; per esempio: ``;

- quando si creano segnalibri (ancore) all'interno delle pagine.

In tutte queste situazioni, è importantissimo considerare la **posizione (server, disco e cartella)** in cui si trova effettivamente il file a cui si fa riferimento.

Tale posizione viene chiamata tecnicamente **percorso (path)**.

- Se il file in questione si trova su un server remoto (cioè in un altro sito Web), non ci sono problemi: è ovvio che va indicato il relativo indirizzo completo (URL).
Per esempio: ``.
Questo tipo di collegamento viene definito come riferimento assoluto.
- Se invece il file fa parte del sito che l'autore sta preparando, dunque si trova (o si troverà) **sullo stesso server** della pagina HTML corrente, allora è necessario fare attenzione.

Potrebbero verificarsi le seguenti circostanze:

- il file si trova **nella stessa cartella** della pagina HTML corrente: in tal caso è sufficiente indicare nei tag soltanto il nome del file; per esempio, se nella pagina 1 si vuole inserire un collegamento alla pagina 2, e le due pagine si trovano nella stessa posizione (cartella), allora basta impostare un tag di questo genere: `` vai a pagina 2 ``; questo tipo di collegamento viene definito **riferimento relativo**;
- il file si trova **in un'altra cartella** rispetto alla pagina HTML corrente: in tal caso occorre specificare nei tag il percorso completo; per esempio, se nella pagina si vuole inserire l'immagine foto1.jpg che si trova nella sottocartella Immagini, allora il tag dovrà essere di questo genere: ``; questo tipo di collegamento è un **riferimento assoluto**.

Per maggiore sicurezza, quando si devono impostare riferimenti e percorsi, si può utilizzare lo speciale tag `<BASE>`. Questo tag, inserito nell'intestazione del documento tra `<HEAD>` e `</HEAD>`, identifica l'indirizzo (URL) predefinito a cui fare riferimento ogni volta che si crea un collegamento a un file esterno alla pagina corrente.

Per esempio:

`<BASE HREF="http://www.miosito.it">`

Tutti i collegamenti ipertestuali all'interno della pagina daranno per scontato il riferimento a questo URL (salvo il caso in cui nel tag si indichi espressamente un URL differente).

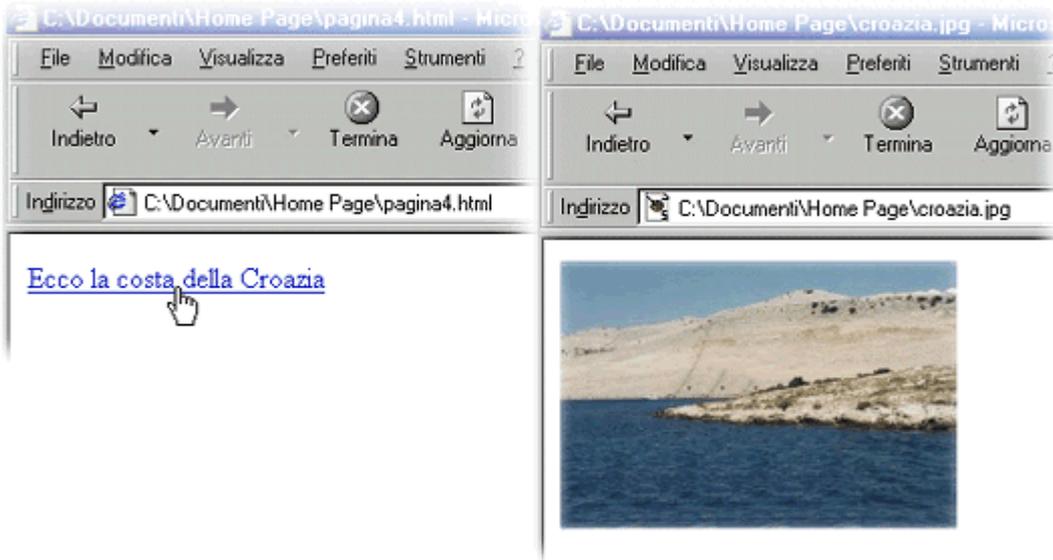
Per creare un collegamento verso **un altro sito Web**, basta inserire come destinazione nel tag `<A HREF>` l'indirizzo completo (URL) del sito in questione. Per esempio:

`` Vai al sito Kataweb ``



Se si indica soltanto il nome di dominio (nell'esempio: `http://www.kataweb.it`) il collegamento porta alla home page predefinita del sito. In alternativa, è possibile creare un collegamento verso **una specifica pagina del sito**, impostandone il percorso completo (per esempio: `http://www.kataweb.it/aiuto/index.html`).

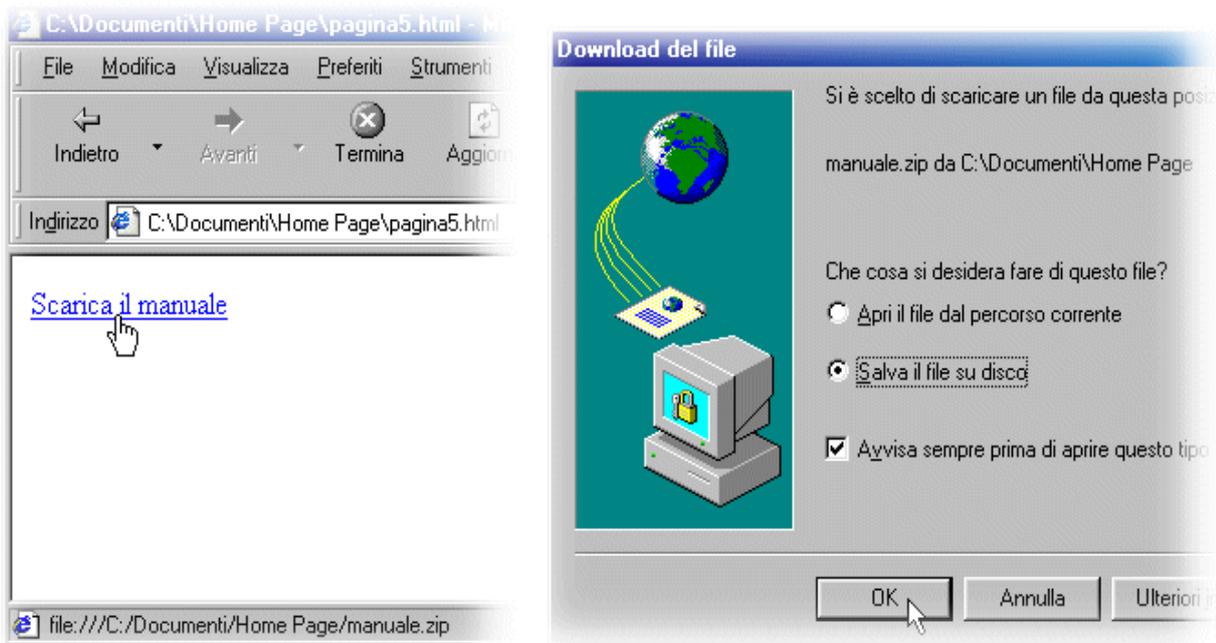
E' possibile impostare un collegamento che rimanda a **un'immagine** (in formato GIF o JPEG).
In tal caso, facendo clic sul collegamento, nella finestra del browser viene visualizzata esclusivamente l'immagine, anziché una pagina Web. Per esempio:



** Visualizza questa foto **

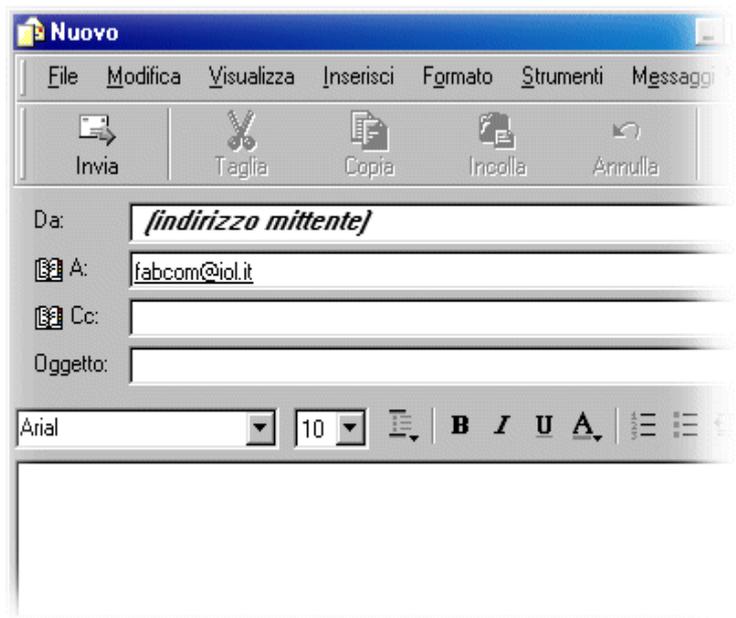
Per proseguire la navigazione bisognerà cliccare su uno dei pulsanti del browser (Indietro, Avanti, Home/Pagina iniziale) o digitare a mano un altro indirizzo Web.
In modo analogo, si può creare un collegamento a **un file esterno** (di qualunque genere: testi, filmati, pacchetti compressi e così via). Se sul computer di chi naviga è installato un programma che riconosce quel formato di file, il file potrà essere **aperto** direttamente. Altrimenti il file verrà **scaricato** mediante un procedimento automatico di download. Per esempio:

** Scarica la relazione in formato compresso .ZIP **



Infine, è possibile creare un collegamento a un indirizzo di posta elettronica (e-mail), indicando la destinazione secondo questo schema: **mailto:utente@dominio.com**.
Per esempio:

** Scrivimi un messaggio **



Collegamenti interni ("ancore")

Si possono creare collegamenti anche all'interno di una stessa pagina Web. In questo modo, il lettore può agevolmente saltare da un punto all'altro del testo: una soluzione molto comoda se la pagina è particolarmente lunga e suddivisa in sezioni distinte.

La predisposizione di collegamenti interni va effettuata secondo la seguente procedura.

- Innanzitutto, si devono **definire le destinazioni ("ancore")** nei vari punti prescelti mediante il tag

****.

Per esempio (supponendo che la pagina sia suddivisa in vari argomenti o paragrafi):

** Qui inizia l'argomento 1 (o il paragrafo 1, e così via) **

(...)

** Qui inizia l'argomento 2 **

(...)

- Successivamente si impostano i **collegamenti** ipertestuali con il tag ****.

** Vai all'argomento 1 **

** Vai all'argomento 2 **

Volendo, si possono **combinare** collegamenti esterni e collegamenti interni: è cioè possibile definire un collegamento ipertestuale che punta a un'ancora specifica all'interno di un'altra pagina Web.

Per esempio:

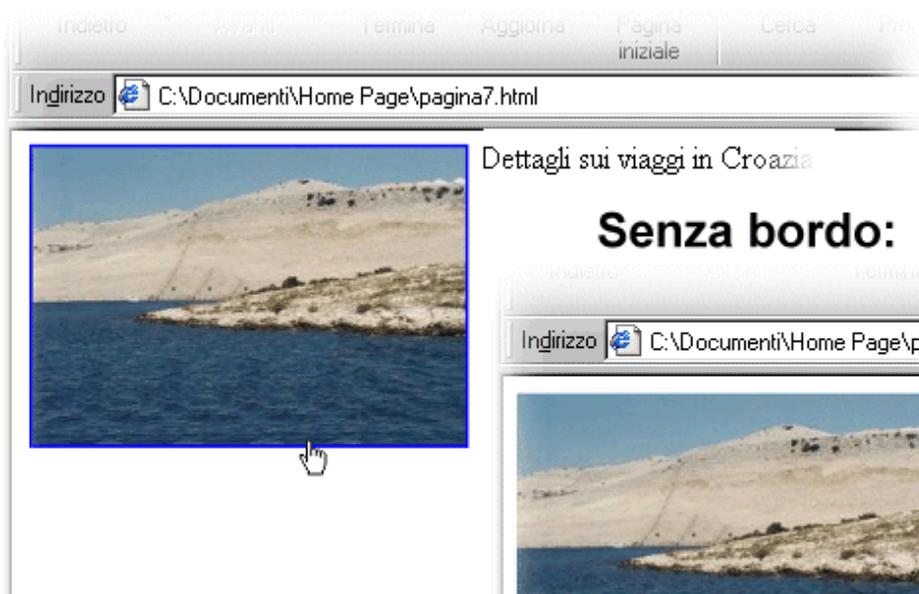
** Vai all'argomento 1 della pagina 2 **

Anche in questo caso, naturalmente, bisogna che in precedenza sia stata definita l'ancora di destinazione all'interno del file *pagina2.html*.

Personalizzare l'aspetto dei collegamenti

- Nella sezione precedente (sezione 4) si è visto come si possono definire a priori i **colori** dei collegamenti ipertestuali, mediante gli attributi **LINK**, **ALINK** e **VLINK** del tag **<BODY>**. In questo modo, si possono adottare colori personalizzati in sostituzione del colore blu standard.
- Se il collegamento è associato a un'**immagine** (anziché a un testo), l'immagine appare contornata da un profilo colorato: una soluzione non molto elegante dal punto di vista grafico. Occorre allora specificare l'attributo **BORDER=0** all'interno del tag **** che richiama l'immagine:

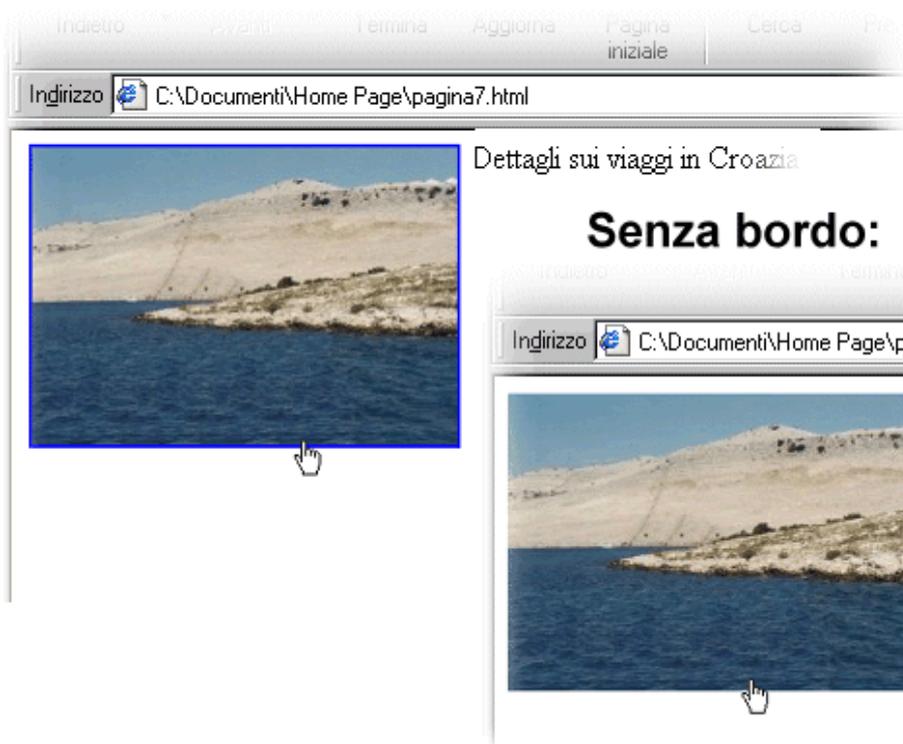
Con bordo:



Senza bordo:



Con bordo:



Il tag `` è spiegato più avanti (v. sezione 13).

Per quanto riguarda i collegamenti ipertestuali, si tenga presente che il puntatore del mouse cambia forma quando passa sopra un elemento attivo: ciò è sufficiente a segnalare che all'immagine è associato un collegamento.

Perciò, è consigliabile nascondere sempre il bordo colorato.

A questo punto, sono stati esaminati i concetti di base e i tag essenziali per la costruzione di una prima pagina Web completa e funzionante (compresi i collegamenti ipertestuali), sebbene ancora assolutamente grezza sul piano grafico.

Esercitazione

A conclusione di questa prima lezione del modulo, ecco un esercizio più complesso (ma anche più libero) per saggiare le competenze maturate.

Creare **tre pagine Web** semplici ma complete, rispettando i seguenti vincoli:

- la prima pagina deve essere chiamata **index.html**; i nomi delle due pagine successive possono essere scelti liberamente;
- ciascuna pagina deve avere un **titolo**;
- ciascuna pagina deve contenere un **collegamento ipertestuale verso le altre due**;
- almeno una delle pagine deve avere uno **sfondo colorato**;
- almeno una delle pagine deve avere il **testo colorato**;
- almeno una delle pagine deve contenere un **collegamento ipertestuale a un qualunque sito Web**;
- almeno una delle pagine deve contenere un **collegamento a un indirizzo di posta elettronica**.

Il **contenuto** delle pagine può essere scelto liberamente:

- si può inserire anche soltanto una semplice frase all'interno di ciascuna pagina;
- oppure si può studiare un contenuto sensato e articolato, così che le tre pagine rappresentino davvero un abbozzo di sito Web. Per esempio, la prima pagina potrebbe contenere un messaggio di benvenuto e una descrizione del sito; la seconda pagina potrebbe contenere il proprio curriculum vitae; la terza pagina potrebbe contenere un piccolo elenco di collegamenti a siti Web che l'autore consiglia in base ai propri interessi e così via.

Sezione 6. Testo e caratteri

Introduzione

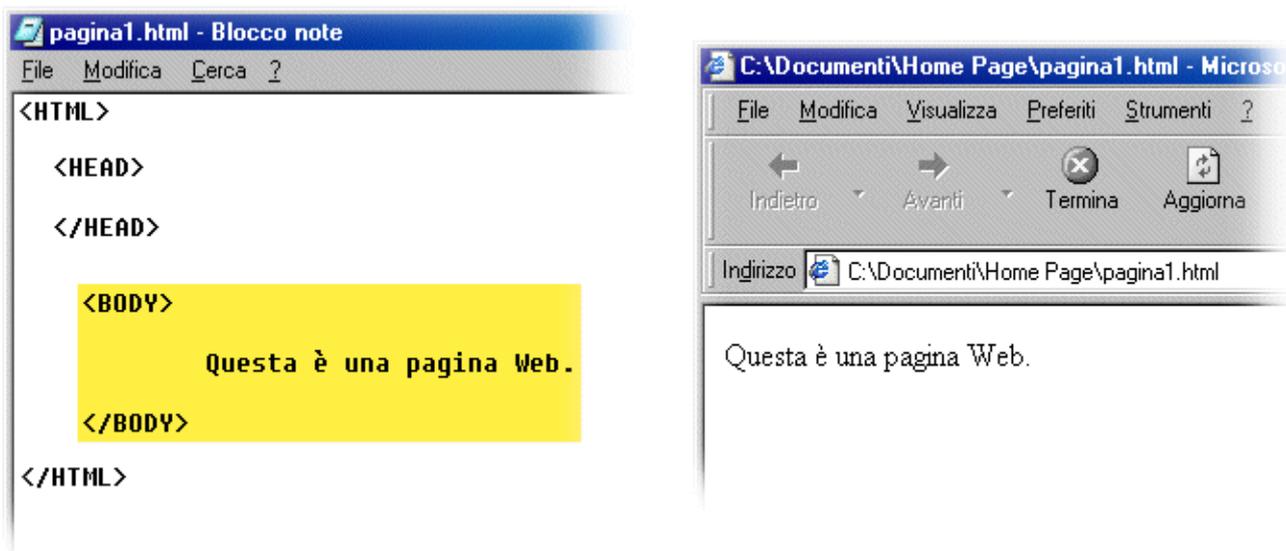
Per chi usa il computer, scrivere è probabilmente l'attività più semplice e consolidata. Di solito, il primo programma che si impara a utilizzare è un software di videoscrittura. Questa attività non comporta particolari difficoltà se, come è normale, si dispone di una tastiera italiana con le lettere accentate e tutti i simboli tipici della nostra lingua. Ebbene, quando si scrive il testo di una pagina Web, operando direttamente all'interno del codice HTML, occorre fare attenzione.

Il testo può essere digitato nel modo consueto, ma in certi casi (lettere accentate e simboli vari) si deve ricorrere ad appositi artifici.

Molto meno immediata, poi, è la personalizzazione dell'aspetto del testo (colori, dimensioni e tipo di carattere).

Digitazione del testo e caratteri speciali

Come si è già visto (a partire dalla sezione 2), il testo inserito nella sezione **<BODY>** ... **</BODY>** del documento appare sullo schermo quando la pagina viene visualizzata. In linea di massima, perciò, è sufficiente digitare il testo normalmente. Per esempio:

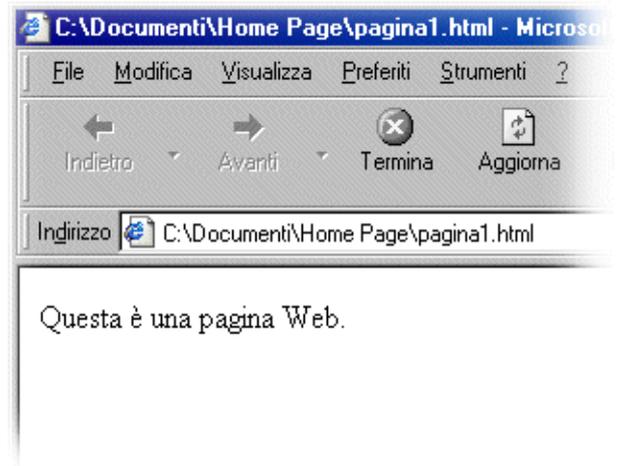


Tuttavia, bisogna fare due precisazioni.

- Gli "a capo" ottenuti con il tasto **Invio** mentre si scrive il codice HTML **non** vengono mantenuti in fase di visualizzazione nel browser. Per mandare a capo il testo visibile nella pagina bisogna inserire nel codice HTML appositi tag, che verranno spiegati tra breve (v. sezione 8).
Per esempio:

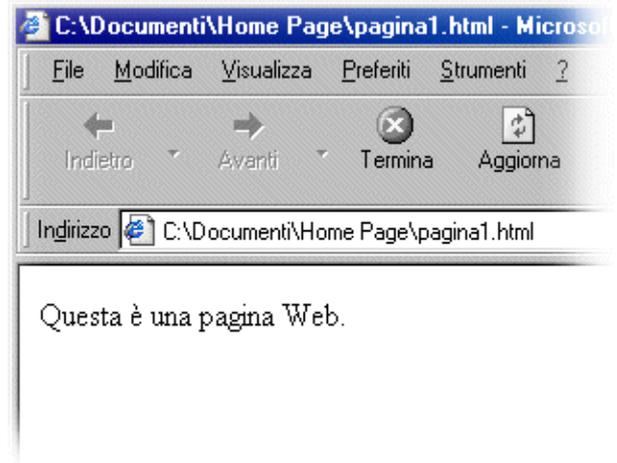
```
pagina1.html - Blocco note
File Modifica Cerca ?
<HTML>
  <HEAD>
  </HEAD>
  <BODY>
    Questa è una
    pagina Web.
  </BODY>
</HTML>
```

a capo
(Invio)



Per visualizzare nella pagina le **lettere accentate** e alcuni **simboli** particolari, non bisogna scriverli direttamente tramite tastiera bensì si devono inserire nel codice HTML apposite stringhe (sequenze di caratteri) chiamate **entity**. Negli esempi precedenti, scrivere a mano la lettera "è" rappresenta una scelta impropria, non del tutto corretta secondo la sintassi HTML (sebbene i browser più recenti la interpretino correttamente). Invece di scrivere è, nel codice HTML si dovrebbe digitare la corrispondente entity **è**, come nell'esempio seguente:

```
pagina1.html - Blocco note
File Modifica Cerca ?
<HTML>
  <HEAD>
  </HEAD>
  <BODY>
    Questa &egrave;; una
    pagina Web.
  </BODY>
</HTML>
```



Le entity (lettere accentate e caratteri speciali), infatti, si ottengono con la sintassi generica **&codice;** (dove *codice* deve essere sostituito da un certo valore numerico o da una certa espressione). Esiste una **tabella di riferimento standard** per ottenere le entity corrispondenti ai caratteri desiderati.

Caratteri speciali, entity e codici HTML

Ecco una tabella riassuntiva con i principali caratteri speciali (entity) presenti su una tastiera italiana e/o di uso comune, e i rispettivi codici HTML.

Carattere	Codice
"	"
&	&
<	<
>	>
[spazio]	
£	£
	¦
©	©
®	®
°	°
±	±
¼	¼
½	½
¾	¾
À	À
Á	Á
Â	Â
Ã	Ã
Ä	Ä
Å	Å
Æ	Æ
Ç	Ç
È	È
É	É
Ê	Ê
Ë	Ë
Ì	Ì
Í	Í
Î	Î
Ï	Ï
Ñ	Ñ
Ò	Ò
Ó	Ó
Ô	Ô
Õ	Õ
Ö	Ö
Ø	Ø
Ù	Ù
Ú	Ú
Û	Û
Ü	Ü
Ý	Ý
à	à
á	á
â	â
ã	ã

ä	ä
å	å
æ	æ
ç	ç
è	è
é	é
ê	ê
ë	ë
ì	ì
í	í
î	î
ï	ï
ñ	ñ
ò	ò
ó	ó
ô	ô
õ	õ
ö	ö
÷	÷
ø	ø
ù	ù
ú	ú
û	û
ü	ü
ý	ý
þ	þ
ÿ	ÿ

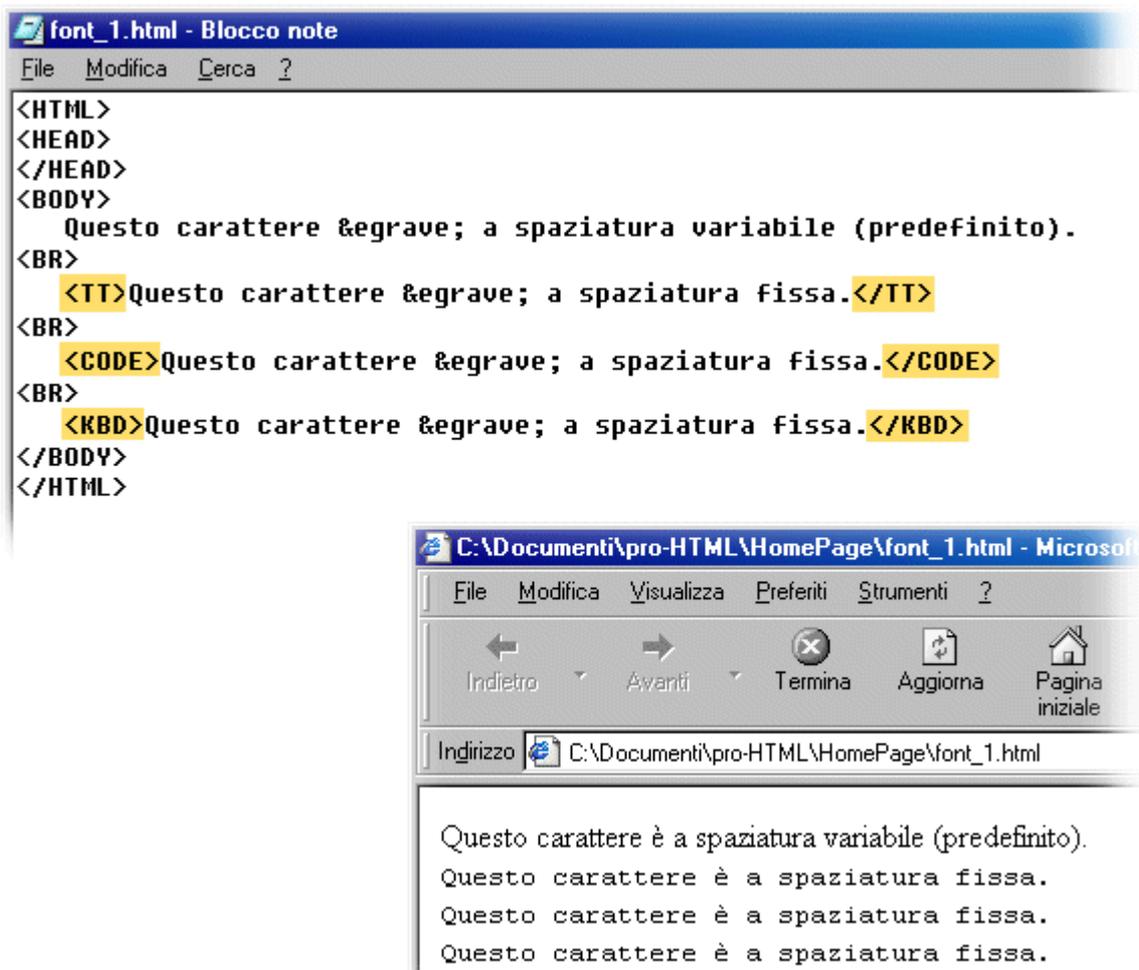
Tipi di carattere: avvertenze generali

Lavorando con il testo, il linguaggio HTML non consente la stessa libertà creativa di un software di videoscrittura o di impaginazione. Chi, è abituato a scegliere tra decine o centinaia di tipi di carattere (font) rischia di restare un po' deluso.

Le vecchie versioni del linguaggio HTML (e le vecchie versioni dei browser) consentivano di scegliere tra soli due tipi di carattere, entrambi tutt'altro che raffinati e originali:

- un carattere **a spaziatura variabile, analogo al Times**: questo è il tipo di carattere predefinito, visualizzato dai browser in mancanza di ulteriori indicazioni (v. esempi precedenti);
- un carattere **a spaziatura fissa, analogo al Courier**: questo tipo di carattere si può ottenere con vari tag, equivalenti tra loro, come `<TT> ... </TT>` ("TeleType": carattere da telescrivente), oppure `<CODE> ... </CODE>`, o anche `<KBD> ... </KBD>`.

Per esempio:



Personalizzazione del tag ****

Il tag ** ... ** permette di personalizzare l'aspetto del testo con una certa flessibilità.

Alcuni suoi attributi vengono correttamente riconosciuti da tutti i browser, altri soltanto dalle versioni più recenti di Internet Explorer e di Netscape Navigator. Gli attributi "universali" del tag **** sono:

- **COLOR**, per definire il colore del testo;
- **SIZE**, per definire le dimensioni (corpo) del testo.

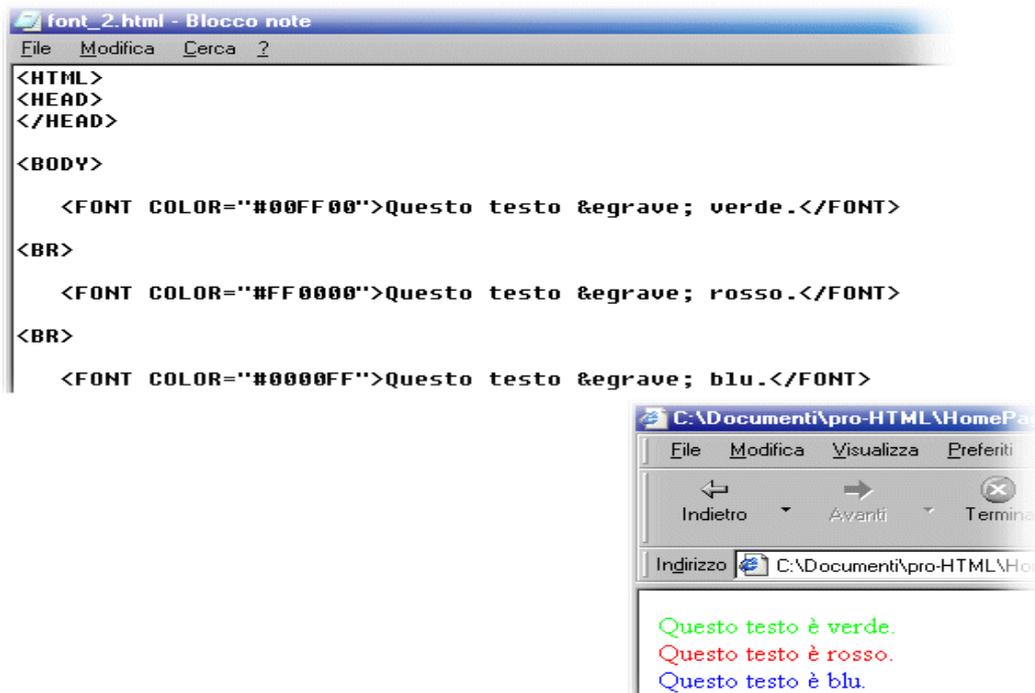
Per quanto riguarda il **colore** del testo, la sintassi è:

**** testo da colorare ****

Questa istruzione prevale sul colore eventualmente definito nel tag generale **<BODY TEXT="#colore">**.

Per i dettagli circa la codifica HTML dei colori si rimanda all'anticipazione fornita nella sezione 4 e, più avanti, alla sezione 11.

Esempio:



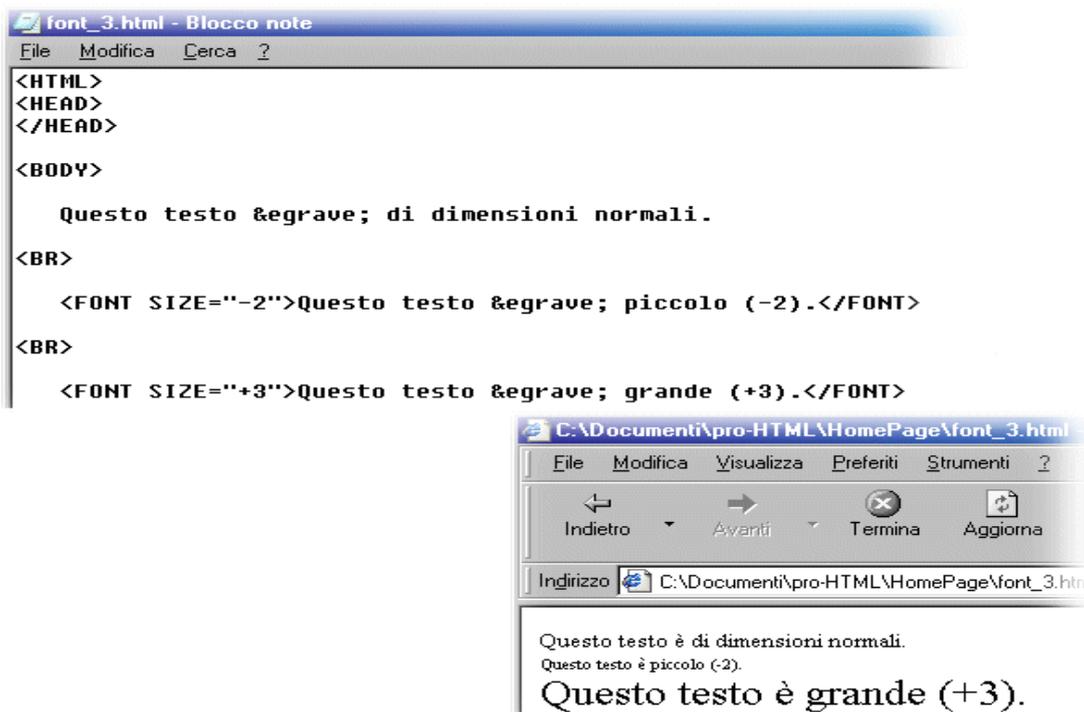
Per quanto riguarda le **dimensioni** dei caratteri, la sintassi è:

**** testo da ridimensionare ****

Il **valore** delle dimensioni del testo viene espresso da un numero intero, che può essere indicato in senso relativo o assoluto:

- in senso relativo: valori da **-3** (molto piccolo) a **+3** (molto grande);
- in senso assoluto: valori da **1** (molto piccolo) a **7** (molto grande).

Per esempio:



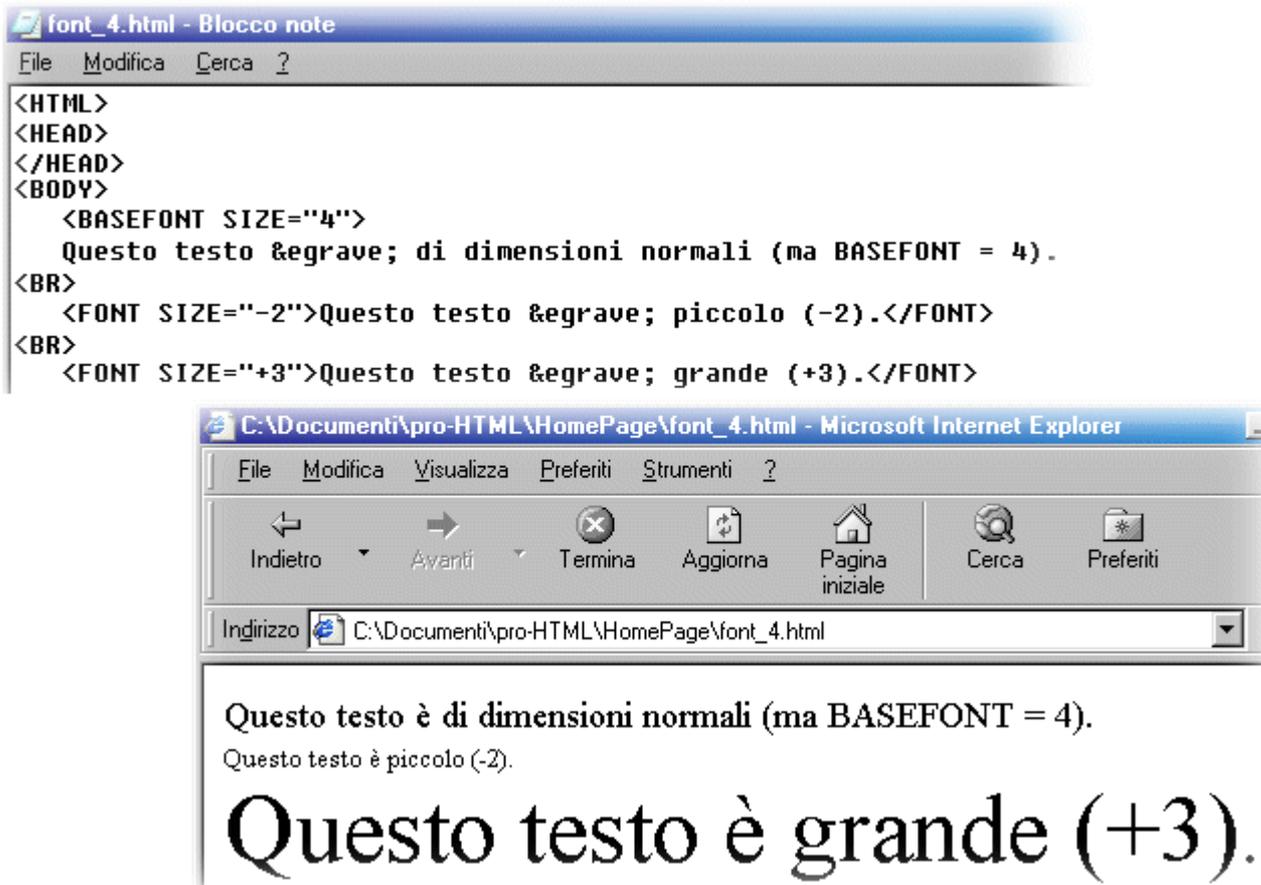
Se lo si desidera, si può fissare un valore di riferimento valido per tutto il documento mediante il seguente tag:

<BASEFONT SIZE="valore">

Questo tag è singolo, non richiede cioè un'istruzione di chiusura. Il valore va espresso in senso assoluto (un numero da 1 a 7).

Da qui in poi, le dimensioni espresse in senso relativo con **** si riferiranno alle dimensioni di base stabilite in **<BASEFONT>**.

Per esempio:



Per inciso, si tenga presente che le dimensioni del testo possono essere definite anche mediante altri due tag:

- **<BIG> ... </BIG>** per ottenere un testo di dimensioni più grandi del normale;
- **<SMALL> ... </SMALL>** per ottenere, viceversa, un testo di dimensioni più piccole del normale.

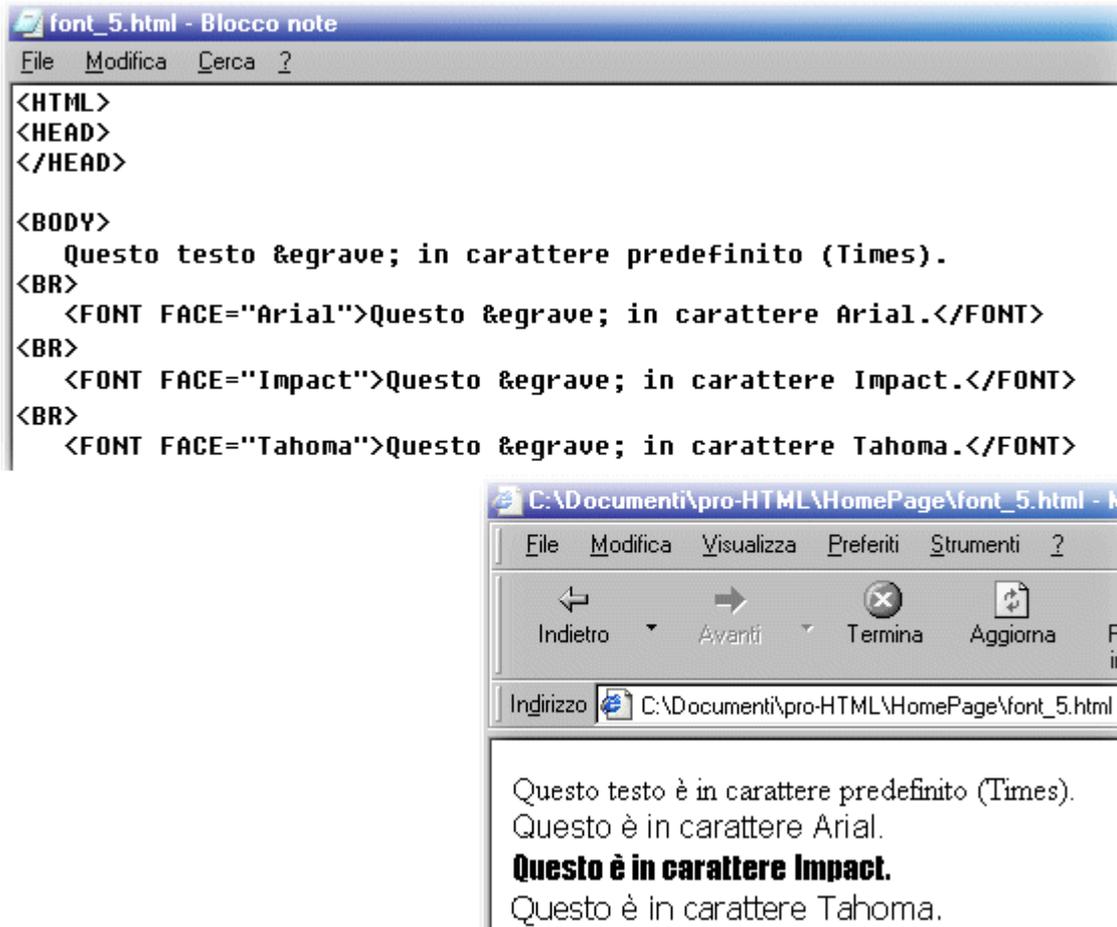
Tuttavia, dato l'evidente carattere approssimativo di questa soluzione, è preferibile ricorrere al tag ****, che consente un controllo molto più preciso sui risultati effettivi.

Come preannunciato, il tag **** consente anche di scegliere il **tipo di carattere**, mediante l'attributo **FACE**.

La sintassi generica è:

**** testo ****

Per esempio:



Occorre però fare attenzione alle seguenti **limitazioni**:

- questa istruzione viene interpretata correttamente solo da alcuni browser (Netscape versione 3.x o superiore; Internet Explorer versione 4.x o superiore);
- è necessario che sul computer di chi visualizza la pagina sia installato il tipo di carattere previsto.

Per aggirare quest'ultima difficoltà, basta indicare alcuni **tipi di carattere alternativi**: se il primo della lista non è presente, verrà visualizzato il secondo; se anche questo manca, verrà visualizzato il terzo e così via.

Esempio:

**** testo in carattere Arial (oppure Helvetica, o Sans-Serif) ****

Una raccomandazione: quando si personalizza l'aspetto del testo, è bene rispettare alcune norme o consuetudini della grafica editoriale. Nella seguente scheda sono riassunti alcuni semplici consigli.

Per concludere, naturalmente tutti gli attributi del tag esaminati in questa sezione possono essere **combinati tra loro**:

**testo personalizzato **

Sezione 7. Attributi ed effetti

Differenza tra stili logici e stili fisici

I tag per la formattazione del testo si suddividono in due tipologie: stili logici e stili fisici. Alcuni di questi tag sono stati citati di sfuggita nelle sezioni precedenti, ma ora vale la pena di esaminarli in un quadro d'insieme.

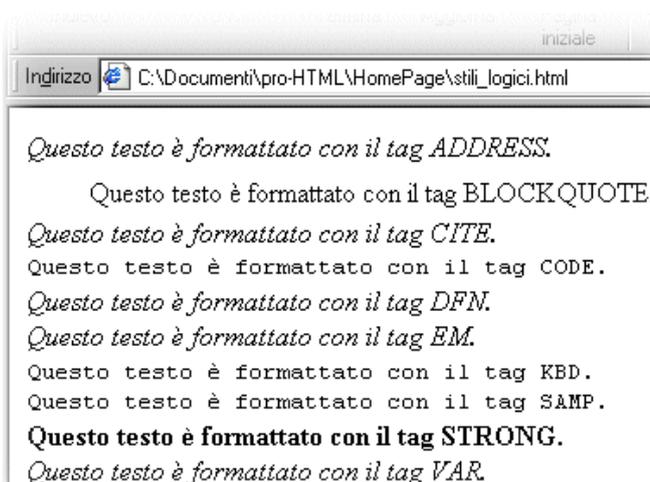
- Gli **stili logici** sono tag che determinano l'aspetto del testo in base a una definizione funzionale, in base cioè al presunto scopo della formattazione (o alla funzione del testo selezionato). Ecco alcuni esempi:

- **<ADDRESS> ... </ADDRESS>**: "indirizzo per contatti": il testo viene generalmente visualizzato in corsivo;
- **<BLOCKQUOTE> ... </BLOCKQUOTE>**: "citazione a blocchetto (come frase a sé stante)": il testo viene rientrato di circa 1 cm dal margine sinistro;
- **<CITE> ... </CITE>**: "citazione (all'interno di una frase)": il testo viene generalmente visualizzato in corsivo;
- **<CODE> ... </CODE>**: "codice, stringa di comandi": il testo viene visualizzato in un carattere a spaziatura fissa;
- **<DFN> ... </DFN>**: "definizione": il testo viene generalmente visualizzato in corsivo;
- ** ... **: "enfasi": il testo viene visualizzato in corsivo;
- **<KBD> ... </KBD>**: "digitazione da tastiera": il testo viene generalmente visualizzato in carattere a spaziatura fissa;
- **<SAMP> ... </SAMP>**: "testo esemplificativo": generalmente visualizzato in un carattere a spaziatura fissa;
- ** ... **: "rafforzamento": il testo viene generalmente visualizzato in grassetto;
- **<VAR> ... </VAR>**: "variabile (logica/programmazione)": il testo viene generalmente visualizzato in corsivo.

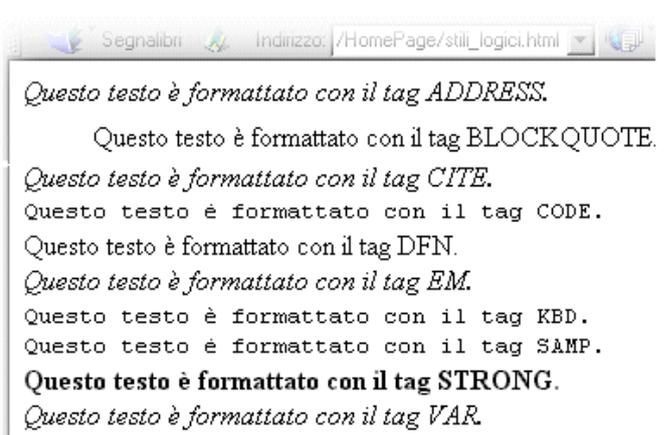
Gli stili logici sono raccomandati da molti manuali di HTML, in quanto rappresentano una soluzione elegante dal punto di vista formale (teorico). Tuttavia, comportano alcune controindicazioni.

Il loro risultato in fase di **visualizzazione**, infatti, può variare a seconda del browser utilizzato dall'utente. Per esempio, si osservi la seguente figura, in cui la visualizzazione in Internet Explorer viene comparata con quella in Netscape Navigator:

Internet Explorer



Netscape



Come si può notare nella figura, il risultato estetico dei vari stili logici corrisponde a semplici variazioni di corsivo, grassetto e carattere a spaziatura fissa. Inoltre, almeno in un caso (tag DFN), il risultato è diverso a seconda del browser utilizzato.

- Gli **stili fisici**, a differenza da quelli logici, forzano il testo ad assumere attributi grafici "oggettivi", assoluti, senza alcun riferimento alla presunta funzione dei singoli elementi. Il testo, mediante questi tag, viene impostato come "puro" grassetto, "puro" corsivo e così via.

Sebbene siano forse meno raffinati dal punto di vista concettuale (rispetto agli stili logici), questi tag non soffrono di imprevisti e controindicazioni di alcun genere e rappresentano pertanto una scelta molto valida nella pratica. Inoltre, sono più facili da ricordare, in quanto il loro significato è immediato e intuitivo e il loro effetto è direttamente prevedibile.

Per giunta, a differenza degli stili logici, i tag di formattazione fisica possono essere impiegati **all'interno di altri elementi** della pagina (titoli, elenchi puntati o numerati e così via) allo scopo di personalizzarne l'aspetto.

Nel prossimo paragrafo, perciò, essi vengono descritti in modo dettagliato, incoraggiandone l'utilizzo.

Attributi di formattazione (stili fisici)

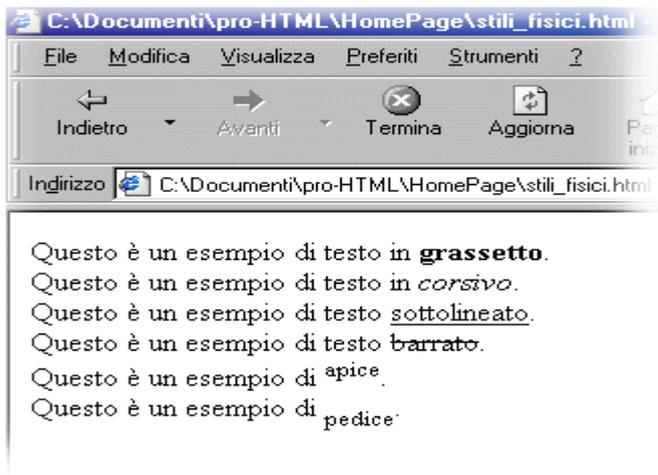
I tag per la formattazione fisica del testo definiscono in modo univoco gli attributi grafici dei caratteri.

Molti di questi tag risulteranno intuitivi per chi ha una certa familiarità con i programmi di videoscrittura.

Ecco l'elenco dei principali stili fisici:

- ** ... **: testo in grassetto (Bold);
- **<I> ... </I>**: testo in corsivo (Italic);
- **<U> ... </U>**: testo sottolineato (Underlined); formattazione da sconsigliare, perché è poco elegante (meglio usare il corsivo) e soprattutto perché può creare confusione con i collegamenti ipertestuali (che solitamente appaiono sottolineati);
- **<STRIKE> ... </STRIKE>**: testo barrato o depennato; formattazione poco raccomandabile e di scarsa utilità;
- **^{...}**: apice, esponente; formattazione molto utile, per esempio, per note, commenti e formule chimiche/matematiche;
- **_{...}**: pedice; formattazione molto utile per le formule chimiche/matematiche.

Ecco un esempio di visualizzazione degli stili fisici e dei relativi attributi di formattazione:



Anche il tag **<TT>** (carattere a spaziatura fissa), descritto nella sezione precedente, può essere considerato uno stile fisico. Lo stesso vale per il tag **<PRE>**, che verrà presentato nella prossima sezione.

Altri effetti decorativi

Al di fuori della sintassi HTML standard, esistono due tag per produrre effetti animati sul testo. Un tag è supportato soltanto da Internet Explorer, l'altro soltanto da Netscape Navigator:

- **<MARQUEE>** ... **</MARQUEE>**: il testo scorre orizzontalmente sullo schermo, se la pagina è visualizzata in **Internet Explorer**;
- **<BLINK>** ... **</BLINK>**: il testo lampeggia sullo schermo, se la pagina è visualizzata in **Netscape Navigator**.

Per esempio:

Se state utilizzando Internet Explorer, questa frase **SCORRE** orizzontalmente. Se state usando Netscape no.

Se state utilizzando Netscape, questa frase **LAMPEGGIA**. Se state usando Internet Explorer no.

Come si può vedere, si tratta di effetti abbastanza rozzi e per giunta vincolati all'utilizzo di uno specifico browser.

Si sconsiglia pertanto di adottarli.

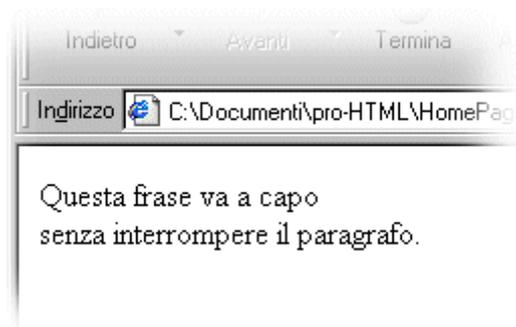
Sezione 8. Paragrafi, blocchi e linee

Andare a capo

Per mandare a capo forzatamente il testo della pagina Web si devono inserire specifici tag all'interno del codice HTML: `
`, `<P>` o `<DIV>`.

- Il tag `
` equivale a una semplice interruzione di riga (che non spezza il paragrafo).
Il testo che segue va a capo, mantenendo però tutte le caratteristiche del testo precedente e senza che si crei uno spazio vuoto tra le righe.
Per esempio:

Questa frase va a capo `
` senza interrompere il paragrafo.

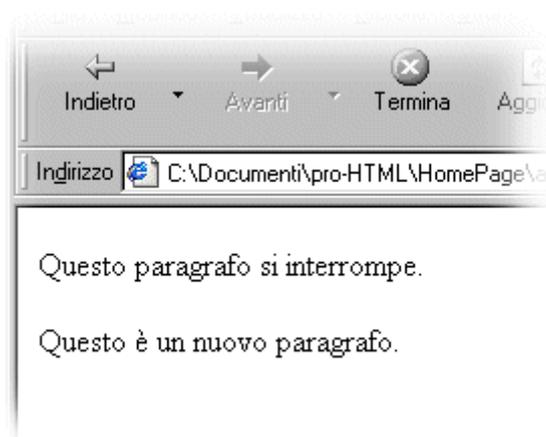


- Il tag `<P>` (che può essere usato sia da solo sia nella forma `<P> ... </P>`) crea invece una vera interruzione di paragrafo.
Il testo successivo va a capo, lasciando una riga di spazio vuoto. Il nuovo paragrafo può essere formattato con attributi e allineamenti diversi rispetto a quello precedente.
Per esempio:

Questo paragrafo si interrompe. `<P>` Questo è un nuovo paragrafo.

Oppure:

`<P>` Questo paragrafo si interrompe. `</P>` `<P>` Questo è un nuovo paragrafo. `</P>`



- Il tag `<DIV>`, che serve a creare un blocco di testo con formattazione omogenea

(v. più avanti in questa sezione), ha un comportamento "ibrido" rispetto a
 e <P>.

Il testo racchiuso tra i tag <DIV> ... </DIV> va a capo e può assumere caratteristiche di formattazione diverse da quelle del testo precedente: rappresenta a tutti gli effetti un nuovo paragrafo (come accade con <P>). Tuttavia, non si crea uno spazio vuoto tra il nuovo testo e quello precedente (come accade con
).

Attenzione!

Per **inserire più righe vuote** (ossia andare a capo ripetutamente), **non** bisogna inserire molteplici tag <P>. Questa sarebbe infatti una violazione della corretta sintassi HTML; il risultato inoltre non sarebbe garantito, perché molti browser ignorano i tag <P> in eccesso.

La soluzione ideale è inserire un tag <P> (o <DIV>) seguito da tanti tag
 quante sono le righe vuote da ottenere.

Allineamenti, giustificazioni, rientri

Chi lavora abitualmente con i programmi di videoscrittura o di impaginazione riesce a gestire allineamenti e rientri senza alcuna difficoltà. Ecco alcuni esempi tipici, illustrati in modo schematico:



Operando con il linguaggio HTML, la gestione di questi aspetti è un po' meno semplice e meno flessibile.

Per impostare gli **allineamenti** (che valgono paragrafo per paragrafo) si può ricorrere a uno dei seguenti metodi:

- **attributo ALIGN** del tag **<P>**: mediante la sintassi **<P ALIGN="valore"> ... </P>** è possibile ottenere i principali tipi di allineamento; ecco alcuni esempi:

- **<P ALIGN="left"> ... </P>**: allineamento a **sinistra**; in realtà, questo è l'allineamento predefinito e non ha bisogno di essere specificato dall'attributo ALIGN.
- **<P ALIGN="center"> ... </P>**: allineamento al **centro**.
- **<P ALIGN="right"> ... </P>**: allineamento a **destra**.
- **<P ALIGN="justify"> ... </P>**: allineamento **giustificato**.

Attenzione!

questo valore **non** è contemplato dallo standard HTML ufficiale, quindi solo pochi browser sono in grado di interpretarlo correttamente.

tag <CENTER> ... </CENTER>: questo tag, ormai obsoleto e dunque sconsigliabile, consente di allineare al **centro** qualunque elemento (blocco di testo o immagine);

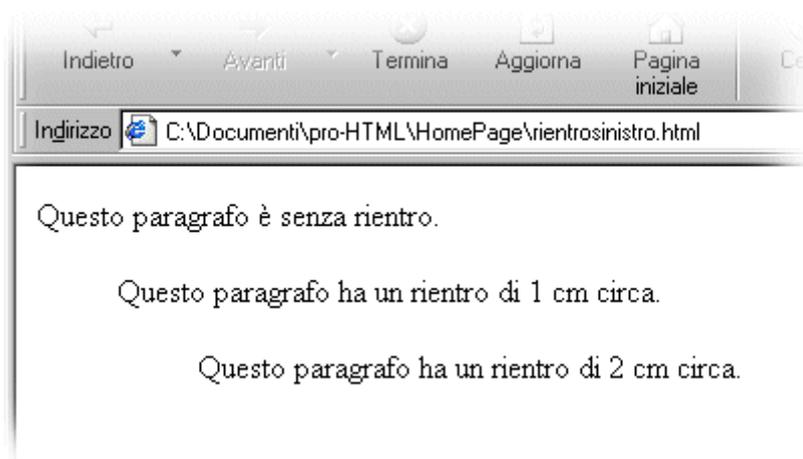
- in alternativa ai metodi precedenti, la soluzione raccomandata è il ricorso al **tag <DIV ALIGN="valore">** (v. più avanti in questa sezione).

Per quanto riguarda i **rientri**, le possibilità sono ancora più limitate.

- Per impostare un **rientro sinistro** si può utilizzare il tag **<BLOCKQUOTE>**, già citato in precedenza.
Si possono sommare più tag **<BLOCKQUOTE>** consecutivi: ciascuno di essi provoca un rientro di circa 1 cm rispetto al margine sinistro.

Per esempio:

```
<P>Questo paragrafo è senza rientro.</P>
<BLOCKQUOTE> Questo paragrafo ha un rientro di 1 cm circa. </BLOCKQUOTE>
<BLOCKQUOTE><BLOCKQUOTE> Questo paragrafo ha un rientro di 2 cm circa.
</BLOCKQUOTE></BLOCKQUOTE>
```



- Non è possibile, invece, definire un **rientro destro**.
L'unica soluzione può essere il ricorso a una tabella (con bordi invisibili) per personalizzare l'allineamento dei paragrafi (v. sezione 15 e successive).

Blocchi o sezioni di testo con formattazione omogenea

Alcuni tag consentono di identificare e formattare in modo omogeneo interi blocchi (sezioni) di testo.

Ciò non equivale necessariamente a creare altrettanti paragrafi: in certi casi il testo va automaticamente a capo e in altri no.

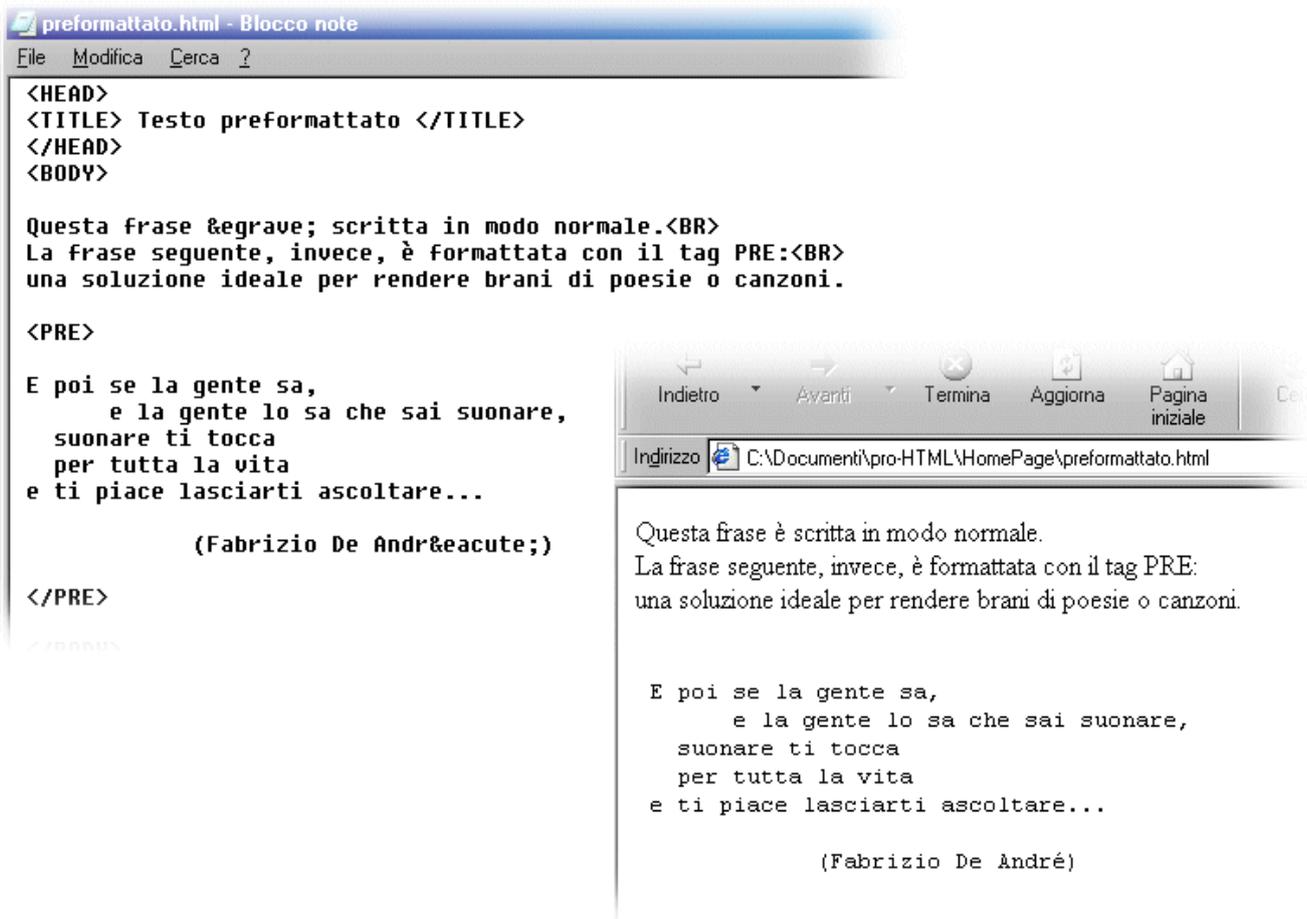
Si tratta piuttosto di una scelta stilistica, per differenziare alcune parti del testo rispetto al resto della pagina.

- Il tag **<DIV> ... </DIV>** serve per contrassegnare una **sezione** di testo, costituita da uno o più paragrafi. In pratica, all'interno di **<DIV> ... </DIV>** si possono racchiudere molteplici paragrafi **<P> ... </P>**.
Di per sé, comunque, il tag **<DIV>** manda **a capo** il testo che segue (senza aggiungere spazio vuoto).
Con questo tag si può definire l'**allineamento** di tutto il testo della sezione, aggiungendovi l'attributo **ALIGN**:
 - **<DIV ALIGN="left"> ... </DIV>**: allineamento a sinistra.
 - **<DIV ALIGN="center"> ... </DIV>**: allineamento al centro.
 - **<DIV ALIGN="right"> ... </DIV>**: allineamento a destra.
- Il tag ** ... ** serve esclusivamente per modificare lo **stile di formattazione** di una parte del testo (un intero paragrafo, oppure una breve frase, o una parola, o un singolo carattere), senza mandare a capo il testo. Viene utilizzato soprattutto in combinazione con i **fogli di stile**: se ne riparlerà più avanti (v. Sezioni 24 e 25).

Rientrano in questa categoria anche altri due tag, di utilizzo meno frequente, ma adeguati a esigenze particolari.

- Il tag **<BLOCKQUOTE> ... </BLOCKQUOTE>** è già stato esaminato: esso consente di contrassegnare una citazione, o di ottenere comunque un nuovo paragrafo con un **rientro sinistro** di circa 1 cm.
- Il tag **<PRE> ... </PRE>** è un caso particolare: permette di definire un paragrafo di **testo preformattato**.
In pratica, il testo racchiuso tra **<PRE>** e **</PRE>** viene visualizzato come un nuovo paragrafo, in carattere a spaziatura fissa e rispettando il modo in cui il testo è stato digitato nel codice HTML del documento: vengono mantenuti (diversamente dal solito) tutti gli spazi, gli a capo e gli allineamenti così come appaiono nel codice sorgente.

Per esempio:



Linee di separazione orizzontali

Per esigenze grafiche, è possibile inserire una linea orizzontale che evidenzi la demarcazione tra differenti blocchi di testo.

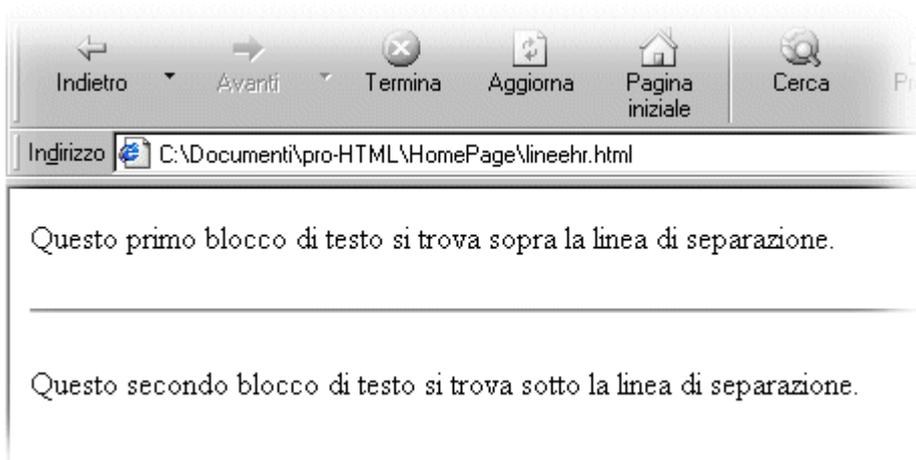
A questo scopo si utilizza il tag **<HR>**.

Per esempio:

<P>Questo primo blocco di testo si trova sopra la linea di separazione.**</P>**

<HR>

<P> Questo secondo blocco di testo si trova sotto la linea di separazione. **</P>**

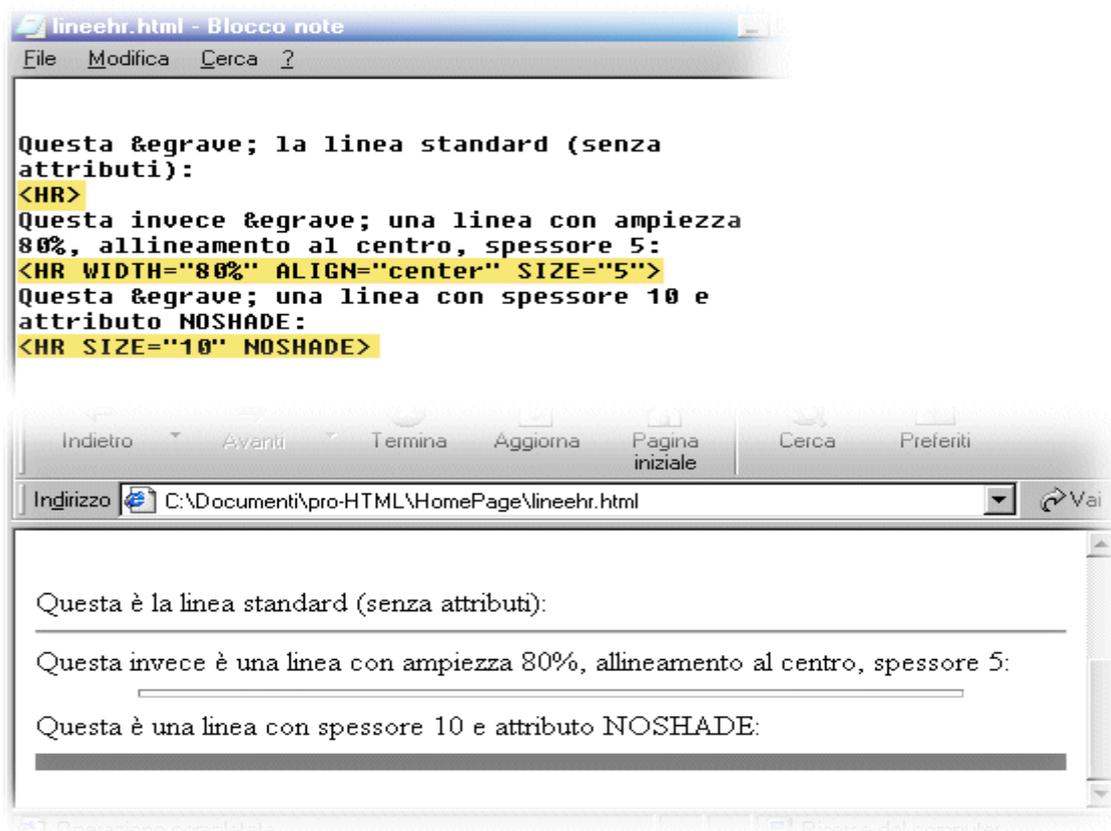


Il tag **<HR>** (da usare da solo, cioè senza istruzione di chiusura) prevede i seguenti attributi:

- **WIDTH="valore"**: si può stabilire la **larghezza** della linea, ossia la sua estensione orizzontale; il valore può essere espresso in **pixel** (un qualsiasi numero intero), per esempio **<HR WIDTH="500">**, oppure in **percentuale** rispetto alla larghezza della finestra del browser (bisogna esplicitare il simbolo **%**), per esempio **<HR WIDTH="80%">**;
- **ALIGN="valore"**: se la linea non occupa l'intera larghezza della pagina, se cioè si è fissato un valore WIDTH (vedi sopra), allora è possibile definirne l'**allineamento**; i valori previsti sono **left**, **center** o **right** (per esempio **ALIGN="center"** per posizionare la linea al centro);
- **SIZE="valore"**: questo attributo determina lo **spessore** della linea; il valore è espresso con un numero intero di **pixel**, a partire da 1 (predefinito);
- **HEIGHT**: in alternativa a **SIZE** si può usare l'attributo **HEIGHT**, che funziona in modo identico;
- **NOSHADE**: se si aggiunge questo attributo (che non richiede di specificare alcun valore), la linea apparirà **piana e piatta**, anziché ombreggiata o tridimensionale;
- **COLOR="#codice_del_colore"**: si può definire il **colore** della linea; tuttavia questo attributo funziona solo se si visualizza la pagina con **Internet Explorer**; il suo utilizzo è dunque sconsigliabile.

Come sempre, i vari attributi possono essere combinati in un unico tag **<HR>**, per ottenere una linea con le caratteristiche desiderate di larghezza, spessore, allineamento e così via.

Per esempio:



Sezione 9. Elenchi puntati e numerati

Elenchi puntati:

Il primo tipo di elenco è quello puntato, il cui tag è (Unordered List, "lista non ordinata").

Si tratta di un semplice elenco di voci (argomenti) senza un ordine di successione; ogni voce dell'elenco viene evidenziata da un simbolo (chiamato "punto" o *bullet*).

I tag ... contrassegnano l'inizio e la fine dell'elenco.

Nel mezzo, un tag identifica ciascuna voce.

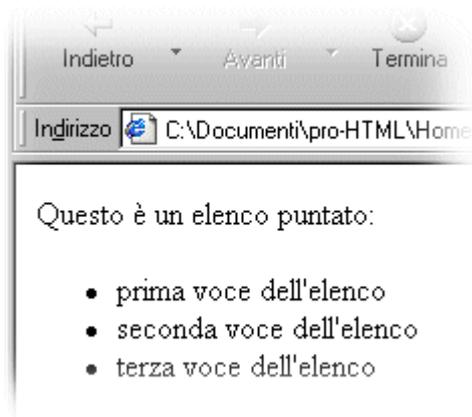
Per esempio:

 prima voce dell'elenco

 seconda voce dell'elenco

 terza voce dell'elenco

(...)



Sono disponibili alcune **opzioni**, nella forma di attributi da aggiungere al tag iniziale:

- **TYPE="tipo_di_simbolo"**: questo attributo permette di scegliere un simbolo personalizzato; i valori previsti sono:
 - **disc** (punto pieno: v. figura precedente);
 - **circle** (punto vuoto, bianco);
 - **square** (quadrato nero).
- **COMPACT**: questo attributo dovrebbe ottimizzare la visualizzazione dell'elenco, ma è scarsamente supportato dai browser e perciò sconsigliabile.

Elenchi numerati:

Un secondo tipo di elenco è quello numerato, il cui tag è (Ordered List, "lista ordinata").

Si tratta di un elenco di voci con un preciso ordine di successione; ogni voce dell'elenco viene evidenziata da un numero (oppure, opzionalmente, da una lettera).

I tag ... contrassegnano l'inizio e la fine dell'elenco.

Nel mezzo, un tag identifica ciascuna voce.

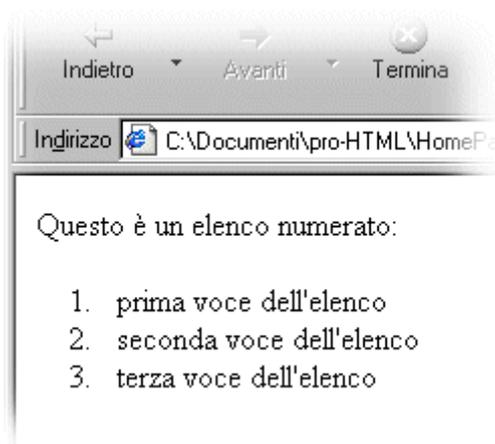
Per esempio:

**** prima voce dell'elenco

**** seconda voce dell'elenco

**** terza voce dell'elenco

(...)



Sono disponibili alcune **opzioni**, nella forma di attributi da aggiungere al tag **** iniziale:

- **TYPE="tipo_di_numerazione"**: questo attributo permette di scegliere i numeri o le lettere per evidenziare le voci dell'elenco; i valori previsti sono:
 - **1** (numeri arabi: v. figura precedente);
 - **I** (numeri romani maiuscoli);
 - **i** (numeri romani minuscoli);
 - **A** (lettere maiuscole);
 - **a** (lettere minuscole);
- **START="numero"**: con questo attributo si può far iniziare il conteggio da un qualsiasi valore; è una soluzione comoda quando si deve interrompere un elenco (inserendo una frase o un'immagine) e lo si vuole riprendere più avanti nella pagina;
- **COMPACT**: questo attributo dovrebbe ottimizzare la visualizzazione dell'elenco, ma è scarsamente supportato dai browser e perciò sconsigliabile.

Nota: elenchi annidati

Gli elenchi puntati o numerati possono anche essere annidati l'uno dentro l'altro: si creano così **elenchi multi-livello**.

È sufficiente inserire un elenco ** ... ** (o ** ... **) all'interno di un altro, facendo attenzione a chiudere correttamente i tag. Ecco un esempio di elenchi puntati annidati:

Questo è l'occorrente per disegnare:

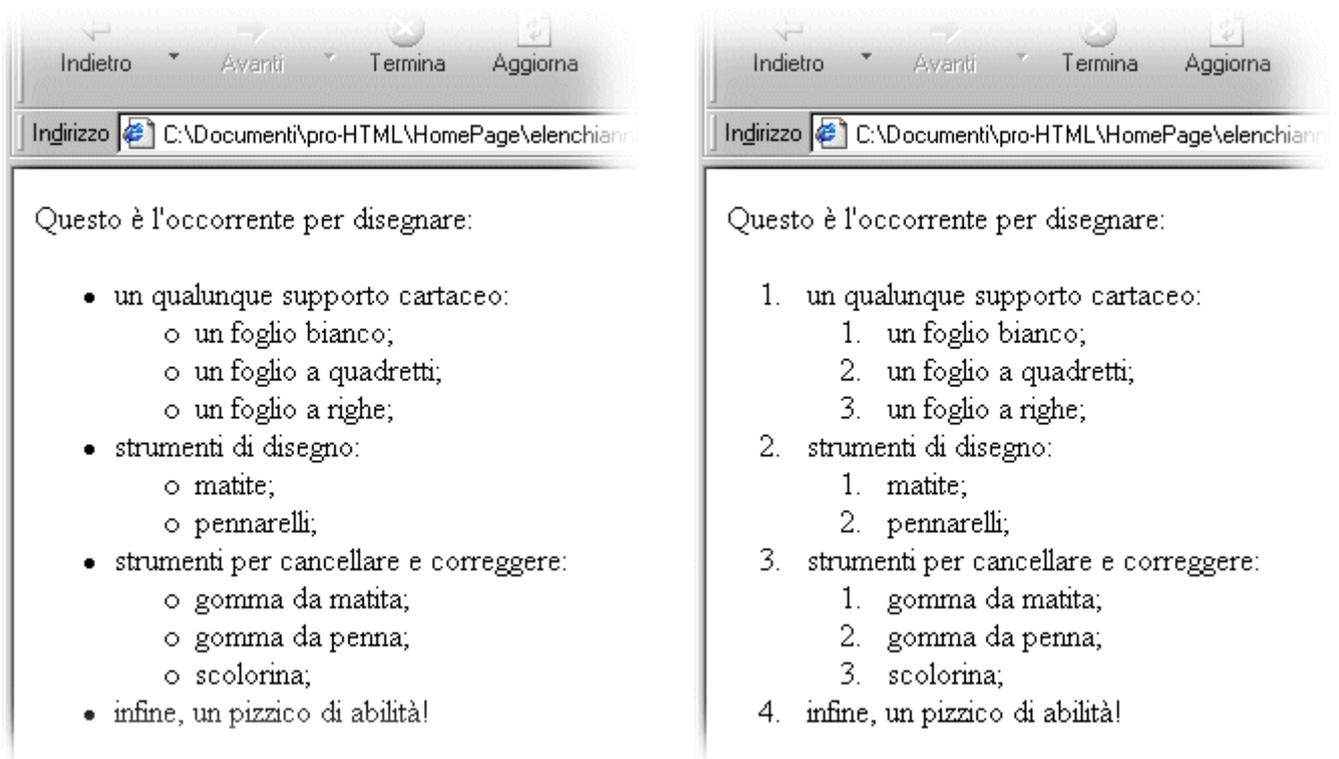
```

<LI>un qualunque supporto cartaceo:
  <UL>
  <LI>un foglio bianco;
  <LI>un foglio a quadretti;
  <LI>un foglio a righe;
  </UL>
<LI>strumenti di disegno:
  <UL>
  <LI>matite;
  <LI>pennarelli;
  </UL>
<LI>strumenti per cancellare e correggere:
  <UL>
  <LI>gomma da matita;
  <LI>gomma da penna;
  <LI>scolorina;
  </UL>
<LI>infine, un pizzico di abilità!
</UL>

```

Seguendo questo stesso schema si possono realizzare elenchi numerati annidati l'uno nell'altro.

Basta utilizzare il tag **** anziché quello ****:



Si noti che gli allineamenti e i punti (o numeri) si adattano automaticamente ai vari sotto-livelli.

Per il resto, valgono le opzioni di personalizzazione già illustrate nei paragrafi precedenti.

Elenchi di definizioni: <DL>

Oltre agli elenchi puntati e numerati, il linguaggio HTML prevede un terzo tipo di elenco, l'**elenco di definizioni**, il cui tag è **<DL>** (Definition List).

In questo caso si elenca una lista di termini, ciascuno accompagnato da una definizione (come in un glossario o in un dizionario).

I tag **<DL> ... </DL>** contrassegnano l'inizio e la fine dell'elenco.

Nel mezzo, un tag **<DT>** (Definition Term) identifica ogni termine da definire e un tag **<DD>** (Definition Data) la rispettiva definizione.

Per esempio:

<DL>

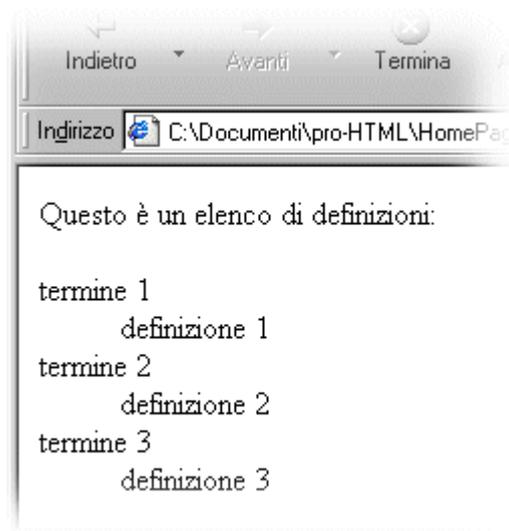
<DT> termine 1 **<DD>** definizione 1

<DT> termine 2 **<DD>** definizione 2

<DT> termine 3 **<DD>** definizione 3

(...)

</DL>



La formattazione dei termini e delle definizioni è automatica.

Alcuni browser evidenziano in grassetto i termini da definire. Le definizioni hanno sempre un rientro sinistro di circa 1 cm.

Sezione 10. Titoli

Livelli di titolazione

Il tag per definire un titolo è `<Hn> ... </Hn>`: il valore n può essere un numero da 1 a 6.

Sono previsti, infatti, **sei livelli di titolazione**, da `<H1>` (il livello superiore) a `<H6>` (il livello inferiore).

Per esempio, il titolo generale del documento potrebbe essere formattato come `<H1>`, i vari capitoli del testo come `<H2>`, i paragrafi di ogni capitolo come `<H3>` e così via.

In fase di visualizzazione, i titoli appaiono come paragrafi a sé stanti (a capo) con caratteristiche di formattazione automaticamente legate al livello di titolazione.

Per esempio:

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

Questo ` un paragrafo di testo normale.

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

Questo ` un paragrafo di testo normale.

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

Questo ` un paragrafo di testo normale.

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

Questo ` un paragrafo di testo normale.

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

Questo ` un paragrafo di testo normale.

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

Questo ` un paragrafo di testo normale.

Come si può notare, la formattazione dei titoli varia automaticamente in base al rispettivo livello gerarchico.

Dal punto di vista pratico si deve considerare che:

- l'aspetto specifico dei titoli può variare leggermente da browser a browser;
- per ragioni di leggibilità, è sconsigliabile l'utilizzo di titoli di livello inferiore a `<H4>`.

Personalizzazione dei titoli

È possibile personalizzare, entro certi limiti, l'aspetto dei titoli per quanto riguarda l'allineamento e la formattazione.

- Il tag `<Hn>` prevede l'attributo **ALIGN="valore"** per definire l'**allineamento** orizzontale del titolo. I valori possibili sono: **left** (predefinito), **center** o **right**. Per esempio:

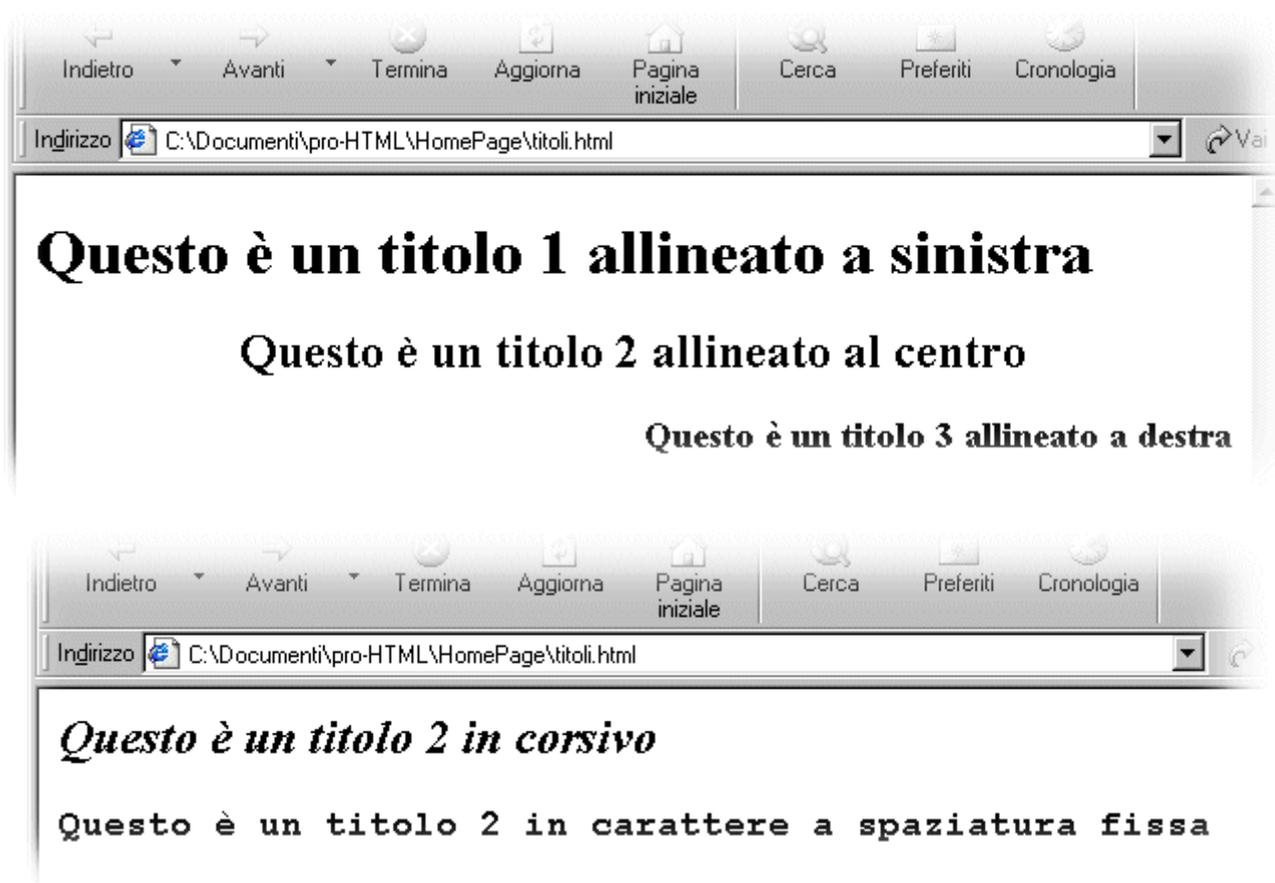
`<H1 ALIGN="left">` Questo ` un titolo 1 allineato a sinistra `</H1>`

`<H2 ALIGN="center">` Questo ` un titolo 2 allineato al centro

`</H2>`

`<H3 ALIGN="right">` Questo ` un titolo 3 allineato a destra

`</H3>`



- All'interno di un tag `<Hn> ... </Hn>` è possibile definire **stili fisici di formattazione** (v. sezione 7): corsivo, grassetto eccetera. È però necessario ricordarsi di chiudere i tag di formattazione prima del tag di chiusura del titolo.

Esempi:

```
<H2><I> Questo è un titolo 2 in corsivo </I></H2>
<H2><TT> Questo è un titolo 2 in carattere a spaziatura fissa
</TT></H2>
```

Si tenga presente che, come molti altri elementi della pagina, i titoli e la loro formattazione possono essere gestiti efficacemente anche tramite i fogli di stile (v. sezioni 24 e 25).

L'importanza dei titoli

Prima di concludere, è bene ribadire che un intelligente utilizzo dei titoli può avere molta importanza ai fini della comunicazione via Web, sotto diversi punti di vista.

- Un'opportuna e ben equilibrata gerarchia di titoli e sottotitoli può incrementare notevolmente la **leggibilità** della pagina, aiutando il visitatore a comprenderne e a memorizzarne i contenuti salienti.
- I **motori di ricerca**, quando catalogano i siti Web, spesso si basano anche sui titoli presenti nella pagina per classificarne gli argomenti. Sarebbe bene, perciò, cercare di formulare il testo dei titoli in modo tale che contenga le parole-chiave più

significative, per segnalare l'argomento del sito. Ne deriverà una maggiore visibilità delle proprie pagine agli occhi di chi effettua ricerche on-line.

A questo punto, si possiedono tutte le nozioni necessarie per creare pagine HTML contenenti testi completi e accuratamente formattati.

Esercitazione

A conclusione di questo secondo modulo del modulo, ecco un esercizio per saggiare le competenze maturate.

In ambito professionale, può capitare di dover convertire in formato HTML un documento cartaceo al fine di pubblicarne un equivalente all'interno di un sito Web. Si consideri per esempio il documento qui raffigurato:

I motori di ricerca

"Una vita senza ricerca
non merita di essere vissuta"
(Platone)

Non basta saper navigare sul Web.
Per utilizzare Internet in modo intelligente e produttivo è necessario conoscere i motori di ricerca e le strategie per interrogarli efficacemente.

Tipi di motori

Esistono vari tipi di motori di ricerca:

- motori "a catalogo" o directory, come *Yahoo*;
- motori per la ricerca libera di parole-chiave, come *Altavista*;
- meta-motori per effettuare ricerche incrociate, come *MetaCrawler*;
- banche-dati specializzate in particolari settori.

Modalità di ricerca

Ad esempio, per effettuare una ricerca libera occorre:

1. collegarsi al sito Web di un apposito motore;
2. inserire una o più parole-chiave, combinate mediante operatori logici come AND oppure NOT;
3. fare clic sul pulsante che avvia la ricerca...

pagina a cura di Lapalisse Search Services - © 2000

Supponendo che si tratti di un normale documento stampato, si crei una pagina HTML che lo riproduca nel modo più fedele possibile. È possibile ottenere risultati qualitativamente validi se si applicano opportunamente i tag e gli attributi studiati in questo modulo del modulo. Non esiste una soluzione ottimale in assoluto: ciascuno può sperimentare differenti strategie e opzioni fino al raggiungimento di un risultato soddisfacente.

Sezione 11. I colori nella codifica HTML

La composizione dei colori e il sistema RGB

Per riprodurre nel modo più fedele possibile la gamma dei colori visibili in natura si ricorre normalmente alla miscelazione di alcune tinte di base: dalla loro combinazione si ottengono tutte le sfumature intermedie.

Questa miscelazione si basa su due diverse tecniche o metodi:

- la **sintesi sottrattiva** (basata sull'assorbimento e dunque sulla sottrazione della luce da parte dei corpi fisici); questo metodo si applica per la **stampa** dei colori su carta o su altri supporti; i colori di base in questo caso sono il ciano, il magenta, il giallo e il nero: la sintesi sottrattiva viene perciò indicata con la sigla **CMYK** (Cyan, Magenta, Yellow, black);
- la **sintesi additiva** (basata sull'emissione e dunque sull'addizione o sovrapposizione della luce grazie a dispositivi di proiezione); questo metodo si applica per la **visualizzazione** dei colori su schermi e monitor; i colori di base in questo caso sono il rosso, il verde e il blu: la sintesi additiva viene perciò indicata con la sigla **RGB** (Red, Green, Blue).

Poiché le **pagine Web** hanno come applicazione primaria la visualizzazione sui monitor, il metodo di riferimento per la gestione dei colori è la **sintesi additiva (RGB)**. Ogni possibile tonalità di colore è definita da una combinazione di valori delle tre componenti di base: rosso, verde e blu.

In pratica, ogni colore è identificato da **tre valori** (numeri): il primo indica la "quantità" di rosso presente nel colore in questione; il secondo la "quantità" di verde"; il terzo la "quantità" di blu.

Ciascuno di questi tre valori può variare da **0** (minimo: componente non presente) a **255** (massimo: componente dominante).

Secondo questo schema, per esempio:

- il **nero** corrisponde ai valori RGB **0, 0, 0** (nessuna luce, assenza di colore);
- il **bianco** corrisponde ai valori RGB **255, 255, 255** (luce "completa");
- il "puro" **rosso** corrisponde ai valori RGB **255, 0, 0**;
- il "puro" **verde** corrisponde ai valori RGB **0, 255, 0**;
- il "puro" **blu** corrisponde ai valori RGB **0, 0, 255**;
- i valori RGB **80, 0, 0** corrispondono a un **rosso scuro/amaranto**;
- i valori RGB **255, 0, 255** corrispondono a una tonalità di **viola/fucsia**.

La codifica esadecimale in HTML

Nel linguaggio HTML i colori vengono definiti nel modo appena illustrato, ma con una differenza: per indicare i valori RGB (da 0 a 255) si utilizza infatti il sistema di codifica esadecimale anziché quello decimale. Nella codifica esadecimale ogni valore è espresso da **una coppia di caratteri**, che possono essere **numeri (da 0 a 9)** o **lettere (da A a F)**.

Ecco qualche esempio di valori decimali con i loro corrispettivi esadecimali:

Valore decimale	Codifica esadecimale
0	00
25	19
50	32
75	4B
100	64
125	7D
150	96
175	AF
200	C8
255	FF

Calcolare i codici RGB esadecimali

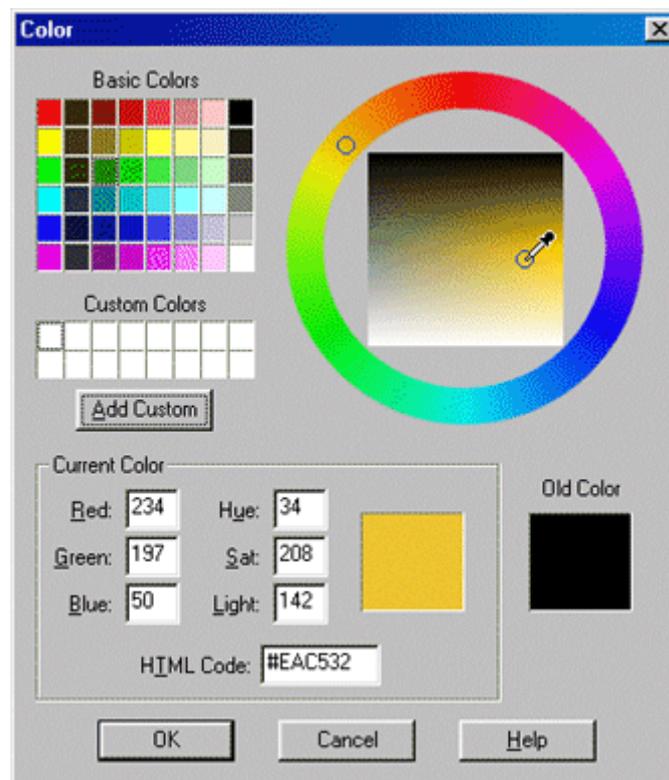
Come spiegato nella sezione, ogni possibile colore è identificato da un gruppo di tre valori (componenti RGB: rosso, verde e blu), ciascuno dei quali può variare da 0 a 255.

Tuttavia, nel codice HTML questi valori devono essere espressi in formato esadecimale, anziché decimale.

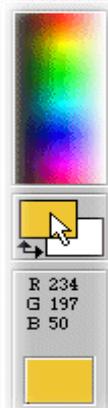
Si pone quindi questo problema: supponendo di conoscere i tre valori che identificano un certo colore, o magari di volerli "dosare" liberamente (creando miscele personalizzate), come si fa a tradurli nei loro corrispettivi esadecimali?

Normalmente, un buon programma di grafica è in grado di visualizzare una tavolozza completa dei colori, evidenziandone i valori tanto in formato RGB decimale quanto RGB esadecimale (HTML).

Si osservi per esempio la seguente schermata (ricavata da Paint Shop Pro): selezionando un colore nella tavolozza se ne ottengono tutte le informazioni necessarie, compresa la codifica esadecimale HTML (HTML Code).



Tuttavia, non sempre si ha a disposizione un programma di grafica così completo. In altri casi, si potrebbe disporre soltanto della codifica RGB decimale corrispondente a un certo colore:

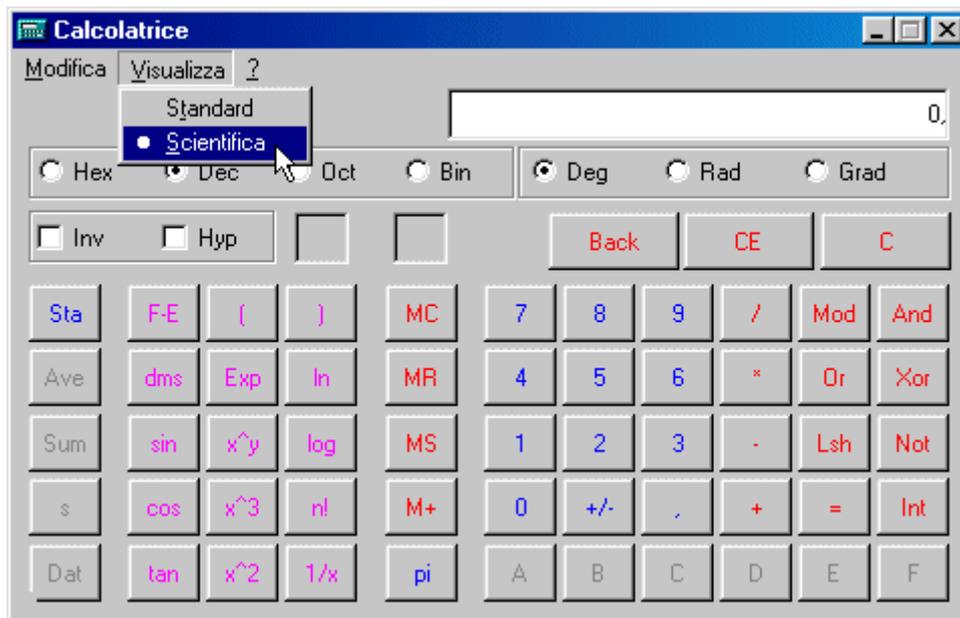


In base a questo esempio, i valori RGB decimali del colore prescelto sono: **234, 197, 50**.

Se si vuole sperimentare liberamente qualunque combinazione di valori RGB, i valori decimali possono essere utilizzati in modo abbastanza intuitivo, calibrando le variazioni tra 0 e 255 per ogni componente.

In tutte queste circostanze sorge appunto il problema del passaggio dal valore decimale a quello esadecimale.

A questo scopo, basta utilizzare una calcolatrice scientifica. Va bene anche un software che simuli una calcolatrice: nel caso di Windows, per esempio, si può attivare il programma **Calcolatrice** (che si trova nel menu **Start/Programmi/Accessori**) e selezionare nel menu **Visualizza** la modalità **Scientifica**.



Verificare che sia selezionato il metodo di codifica decimale (Dec): non ci sono difficoltà; è sempre quello predefinito, dunque dovrebbe essere già attivo in partenza. Si digiti quindi il primo dei tre valori (nell'esempio di cui sopra: **234**).



Il valore indicato viene istantaneamente convertito in termini esadecimali (nell'esempio: **EA**). Ripetere questa operazione (passi 4 e 5) per gli altri due valori. Si otterranno i risultati **C5** (corrispondente al decimale 197) e **32** (corrispondente al decimale 50).

Ecco quindi il codice RGB del colore prescelto, tradotto in formato HTML: **#EAC532**.

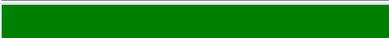


Ecco ora due tabelle (anzi, due tavolozze) in cui sono riassunti i principali colori da utilizzare nelle pagine Web.

Nelle tabelle sono riportati i **nomi in inglese** dei colori (per motivi che saranno spiegati nel seguito della sezione) e i corrispondenti **codici** esadecimali RGB.

Si deve tenere presente che, per una serie di ragioni tecniche, la scelta è limitata a **un certo numero di colori** (mentre le combinazioni possibili in assoluto sono milioni).

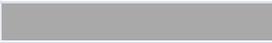
La seguente tabella mostra i **primi 16 colori** universalmente riconosciuti da tutti i browser:

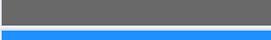
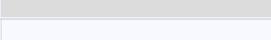
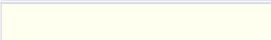
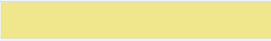
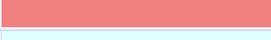
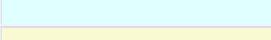
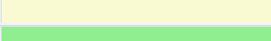
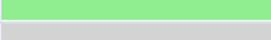
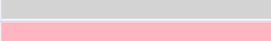
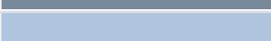
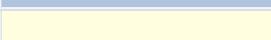
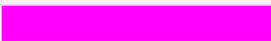
Colore (campione)	Nome (in inglese)	Codice esadecimale RGB
	aqua	00FFFF
	black	000000
	blue	0000FF
	fuchsia	FF00FF
	gray	808080
	green	008000
	lime	00FF00
	maroon	800000

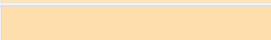
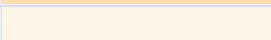
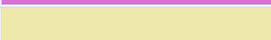
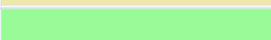
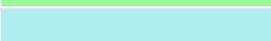
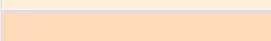
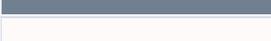
	navy	000080
	olive	808000
	purple	800080
	red	FF0000
	silver	C0C0C0
	teal	008080
	white	FFFFFF
	yellow	FFFF00

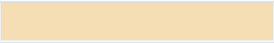
Tabella dei 124 colori riconosciuti dai browser più diffusi

Questa seconda tabella mostra **altri 124 colori**, riconosciuti solo dai browser più diffusi (sia Netscape sia Internet Explorer, dalla versione 3.x in poi)

Colore (campione)	Nome (in inglese)	Codice esadecimale RGB
	aliceblue	F0F8FF
	antiquewhite	FAEBD7
	aquamarine	7FFFD4
	azure	F0FFFF
	beige	F5F5DC
	bisque	FFE4C4
	blanchedalmond	FFEBCD
	blueviolet	8A2BE2
	brown	A52A2A
	burlywood	DEB887
	cadetblue	5F9EA0
	cadetblue	5F9EA0
	chartreuse	7FFF00
	chocolate	D2691E
	coral	FF7F50
	cornflowerblue	6495ED
	cornsilk	FFF8DC
	crimson	DC1436
	cyan	00FFFF
	darkblue	00008B
	darkcyan	008B8B
	darkgoldenrod	B8860B
	darkgray	A9A9A9
	darkgreen	006400
	darkkhaki	BDB76B
	darkmagenta	8B008B
	darkolivegreen	556B2F
	darkorange	FF8C00
	darkorchid	9932CC
	darkred	8B0000
	darksalmon	E9967A

	darkseagreen	8FBC8F
	darkslateblue	483D8B
	darkslategray	2F4F4F
	darkturquoise	00CED1
	darkviolet	9400D3
	deeppink	FF1493
	deepskyblue	00BFFF
	dimgray	696969
	dodgerblue	1E90FF
	firebrick	B22222
	floralwhite	FFFAF0
	forestgreen	228B22
	gainsboro	DCDCDC
	ghostwhite	F8F8FF
	ghostwhite	F8F8FF
	gold	FFD700
	goldenrod	DAA520
	greenyellow	ADFF2F
	honeydew	F0FFF0
	hotpink	FF69B4
	indianred	CD5C5C
	indigo	4B0082
	ivory	FFFFF0
	khaki	F0E68C
	lavender	E6E6FA
	lavenderblush	FFF0F5
	lawngreen	7CFC00
	lemonchiffon	FFFACD
	lightblue	ADD8E6
	lightcoral	F08080
	lightcyan	E0FFFF
	lightcyan	E0FFFF
	lightgoldenrodyellow	FAFAD2
	lightgreen	90EE90
	lightgray	D3D3D3
	lightpink	FFB6C1
	lightsalmon	FFA07A
	lightseagreen	20B2AA
	lightskyblue	87CEFA
	lightslategray	778899
	lightsteelblue	B0C4DE
	lightyellow	FFFFE0
	limegreen	32CD32
	linen	FAF0E6
	magenta	FF00FF
	mediumaquamarine	66CDAA

	mediumblue	0000CD
	mediumorchid	BA55D3
	mediumpurple	9370DB
	mediumseagreen	3CB371
	mediumslateblue	7B68EE
	mediumspringgreen	00FA9A
	mediumturquoise	48D1CC
	mediumvioletred	C71585
	midnightblue	191970
	mintcream	F5FFFA
	mistyrose	FFE4E1
	moccasin	FFE4B5
	navajowhite	FFDEAD
	oldlace	FDF5E6
	olivedrab	6B8E23
	orange	FFA500
	orangered	FF4500
	orchid	DA70D6
	palegoldenrod	EEE8AA
	palegreen	98FB98
	paleturquoise	AFEEEE
	palevioletred	DB7093
	papayawhip	FFefd5
	peachpuff	FFDAB9
	peru	CD853F
	pink	FFC0CB
	plum	DDA0DD
	powderblue	B0E0E6
	rosybrown	BC8F8F
	royalblue	4169E1
	saddlebrown	8B4513
	salmon	FA8072
	sandybrown	F4A460
	seagreen	2E8B57
	seashell	FFF5EE
	sienna	A0522D
	skyblue	87CEEB
	slateblue	6A5ACD
	slategray	708090
	snow	FFFAFA
	springgreen	00FF7F
	steelblue	4682B4
	tan	D2B48C
	thistle	D8BFD8
	tomato	FF6347
	turquoise	40E0D0

	violet	EE82EE
	wheat	F5DEB3
	whitesmoke	F5F5F5
	yellowgreen	9ACD32

Modalità di applicazione: un riepilogo

Alcune nozioni pratiche circa l'impostazione dei colori nella pagina sono già state anticipate nelle sezioni precedenti; altre saranno specificate nelle prossime sezioni. Riepilogando, è possibile definire il colore dei seguenti elementi:

- **sfondo** della pagina (v. sezione 4): attributo **BGCOLOR** del tag **<BODY>**;
- **testo** della pagina (v. sezione 4): attributo **TEXT** del tag **<BODY>**;
- **collegamenti** ipertestuali (v. sezione 4): attributi **LINK**, **ALINK** e **VLINK** del tag **<BODY>**;
- **testo** di una parte della pagina (v. sezione 6): attributo **COLOR** del tag ****;
- **sfondo** di tabelle e singole celle (v. più avanti, sezioni 17 e 18);
- **sfondo** di singole porzioni di testo, mediante i fogli di stile (v. più avanti, sezioni 24 e 25).

Ecco ora una piccola aggiunta a quanto anticipato nelle sezioni precedenti. Per indicare un colore nel codice HTML della pagina si può utilizzare l'espressione **"#codice_esadecimale"** oppure **"nome_del_colore"**. Questo è il motivo per cui nelle tabelle precedenti è stato indicato il nome (inglese) di ogni colore. Per esempio:

** ... **

oppure

** ... **

Attenzione!

La seconda forma non viene riconosciuta correttamente da tutti i browser ed è perciò sconsigliata: meglio ricorrere sempre al codice esadecimale.

Visualizzazione nei browser: avvertenze e consigli

Per concludere, ecco due brevi osservazioni circa la visualizzazione dei colori nei browser durante la navigazione e la lettura delle pagine.

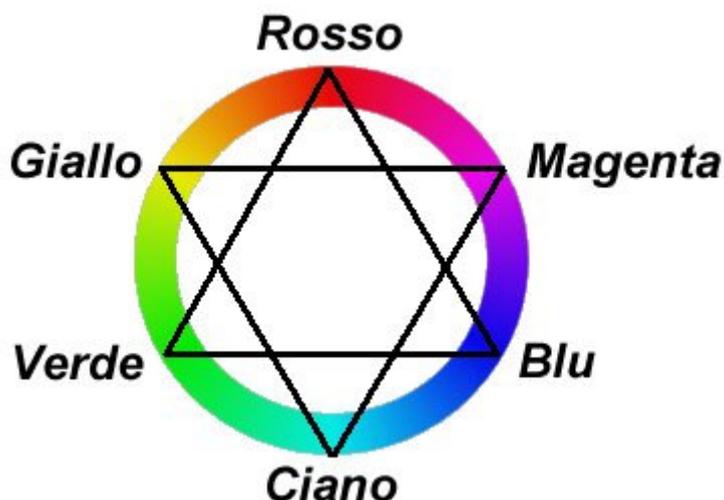
- Innanzitutto, è bene sapere che i risultati sul piano grafico/estetico non sono mai totalmente affidabili.
- Rispetto alla progettazione della pagina HTML e alle prove preliminari di visualizzazione potrebbero sempre insorgere effetti imprevisti o non del tutto fedeli. Il fatto è che sussistono piccole **differenze tra browser e browser, nonché tra sistema operativo e sistema operativo**.
La stessa pagina (con gli stessi colori) potrebbe apparire diversa se visualizzata da un PC o da un Mac, o da un browser piuttosto che da un altro.
Questo è un limite intrinseco delle tecnologie Web e non c'è modo di garantirsi un'assoluta fedeltà di riproduzione dei colori. .
- In secondo luogo, per progettare **opportuni abbinamenti** cromatici (soprattutto nel rapporto **testo/sfondo**) sarebbe bene basarsi su alcune elementari norme grafiche. Si tratta di un argomento che indubbiamente esula da questo modulo, ma converrebbe almeno leggere alcune note introduttive al riguardo.

Abbinamenti e contrasti cromatici

Il tema dei corretti abbinamenti cromatici è affascinante, ma troppo esteso per essere approfondito in questa sede: chi desidera informazioni dettagliate dovrebbe rivolgersi a specifici manuali o corsi di grafica professionale.

Tuttavia, vale la pena di accennare alcuni concetti fondamentali, che possono risultare utili per chi si avvicina alla progettazione grafica di pagine Web.

Come principio generale per definire l'armonia degli abbinamenti tra i colori, si osservi questo schema:



In questa ruota sono evidenziati i colori di riferimento del sistema RGB (rosso, verde e blu) e quelli del sistema CMYK (ciano, magenta, giallo - ovviamente si omette il nero, che non pone particolari problemi).

Nel mezzo si trovano tutte le tonalità intermedie. Oltre ai colori "puri", inoltre, bisognerebbe considerare le sfumature prodotte dal variare della luminosità (per esempio: verde chiaro, verde scuro ecc.), ma per semplificare si può sorvolare su questo aspetto.

Ebbene, prendendo come spunto le linee che collegano i colori di riferimento si tenga presente che tra un colore e l'altro si possono stabilire due tipi di rapporto:

- **complementarietà**, tra colori diametralmente opposti sulla ruota (per esempio: giallo e blu);
- **contrasto**, tra colori posti alle estremità di una stessa linea (per esempio: rosso e blu).

In linea di massima (sebbene si tratti di una semplificazione) l'abbinamento tra colori complementari può essere considerato armonioso o quanto meno accettabile.

Viceversa, l'abbinamento tra colori fortemente contrastati tende ad affaticare l'occhio del lettore e andrebbe dunque evitato a tutti i costi, salvo il caso in cui si voglia attirare violentemente l'attenzione.

In ogni caso, un contrasto eccessivo e disarmonico rende estremamente difficoltosa la lettura e deve perciò essere sempre evitato nel caso di un testo relativamente lungo su uno sfondo omogeneo.

Ciò vale a maggior ragione nel caso delle pagine Web, in quanto i colori visualizzati su un monitor (che emette attivamente la luce) sono mediamente più vividi e brillanti dei corrispettivi risultati stampati su carta.

Nonostante queste indicazioni possano sembrare intuitive e dettate dal buon senso, in realtà è molto facile imbattersi in **errori di progettazione grafica** che appesantiscono la consultazione delle pagine Web.

Si osservi per esempio questa porzione di pagina Web, tratta da un sito altamente professionale:

Copyright 2000 internet.com Corp.
All Rights Reserved. [Legal Notices](#) , [Reprints](#)



Sebbene gli sviluppatori di questo sito siano professionisti dei servizi Web, la scelta degli abbinamenti cromatici da essi operata è quanto mai infelice.
Chi riesce agevolmente a leggere le scritte blu su sfondo verde alza la mano...

Consiglio

È bene affinare il proprio senso estetico e studiare nei limiti del possibile le soluzioni più opportune per scegliere i colori dello sfondo e del testo.

Molti siti Web trattano questo argomento. Un buon punto di partenza per l'esplorazione personale potrebbe essere il seguente URL: <<http://www.two4u.com/color>>

Sezione 12. Immagini: formati e ottimizzazione

Caratteristiche generali delle immagini digitali

Ogni immagine digitale è caratterizzata da tre aspetti:

- **risoluzione:** dimensioni/qualità visiva dell'immagine; la risoluzione comprende due aspetti:

quantità di punti che compongono effettivamente l'immagine; si misura in **pixel**; esempio: 640x480 (640 pixel orizzontali x 480 verticali), 1024x768 e così via;

densità dei punti nella visualizzazione o stampa; si misura in **dpi** (dot per inch, punti per pollice): per esempio: 72 dpi, 300 dpi ecc.;

- **profondità di colore:** numero massimo di colori possibili; si misura in **bit**; i valori tipici sono: 8 bit (256 colori), 16 bit (65mila colori), 24 bit (milioni di colori);
- **formato/compressione:** formato di salvataggio o **codifica** del file; alcuni formati permettono di comprimere i dati, ottenendo file di dimensioni ridotte: certi metodi di compressione (detti "lossy", o distruttivi) possono influire negativamente sulla qualità dell'immagine;

esempio di formato **non compresso:** BMP (Windows/OS2 Bitmap);

esempi di formati **compressi:** TIFF, GIF, JPEG, PNG.

Ciascuno di questi tre aspetti concorre a determinare le **dimensioni complessive del file** dell'immagine: a parità di risoluzione, una maggiore profondità di colore produce un file più ingombrante e così via.

I formati di file per le immagini sul Web

Nella seguente tabella sono illustrati i formati di file utilizzabili nelle pagine Web:

Formato:	Compressione:	Prof. di colore:	Applicazione ottimale:	Note aggiuntive:
GIF (CompuServe Graphics Interchange Format)	Sì (automatica)	256 colori	disegni, icone, schermate computer	supporta sfondo trasparente e animazione
JPG, JPEG (Joint Photograph Expert Group)	Sì (regolabile)	milioni di colori	fotografie	ottima qualità visiva
PNG (Portable Network Graphics)*	Sì (automatica)	milioni di colori	come GIF e JPEG	supporta sfondo trasparente

* Il formato PNG è stato proposto in tempi relativamente recenti e non è ancora molto diffuso.

Il concetto di ottimizzazione

Per chi naviga sul Web il fattore tempo è rilevante. Se una pagina viene caricata troppo lentamente, quasi sempre il visitatore interrompe l'attesa e si allontana dal sito prima di averla ricevuta e letta.

Il **tempo di caricamento** delle pagine Web dipende strettamente dalle **dimensioni dei file**: più un file è ingombrante e più lenta sarà la sua trasmissione attraverso la Rete.

È perciò molto importante ridurre il più possibile le dimensioni dei file, in modo da renderne più rapido il caricamento.

Non si possono stabilire criteri assoluti: molto dipende dalla velocità della connessione dei visitatori e dalle condizioni di traffico sulla Rete. Tuttavia, almeno indicativamente, di solito si suggerisce come limite massimo **la soglia dei 50 KB**: se i file che compongono un documento Web (**pagina HTML + immagini + eventuali altri componenti**) hanno un ingombro inferiore o pari ai 50 KB complessivi, allora il tempo di caricamento sarà accettabile nella maggior parte delle circostanze.

Partendo da queste premesse, bisogna notare che le pagine HTML (essendo file di puro testo) generalmente non raggiungono dimensioni rilevanti: anche se il testo è molto lungo (più di 5-6 schermate verticali) l'ingombro del file resta modesto.

Dunque, l'elemento che più "pesa" in termini di ingombro e tempi di caricamento sono senz'altro **le immagini**.

Qui entra in gioco il concetto di **ottimizzazione**: ottimizzare un'immagine significa "limarne" al massimo le dimensioni, raggiungendo il miglior compromesso tra qualità visiva e dimensioni del file.

Per ottimizzare un'immagine digitale, in generale, si può intervenire su differenti fattori:

- ridurre la risoluzione, ossia le dimensioni reali dell'immagine;
- ridurre la profondità di colore;
- salvare il file in un formato con elevato livello di compressione.

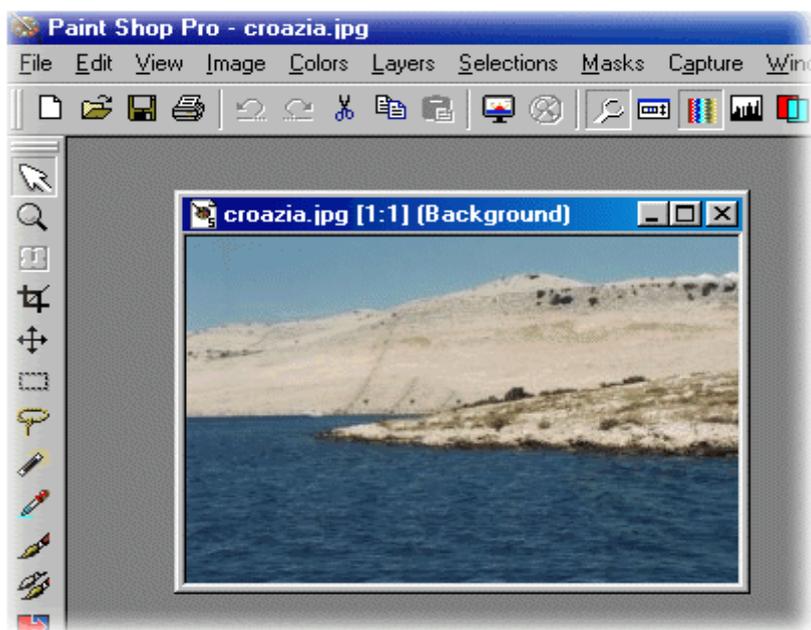
Tuttavia, per quanto riguarda il Web la scelta è limitata: come si è detto, i formati standard sono GIF e JPEG, ciascuno vincolato a una specifica profondità di colore. Restano quindi praticabili solo le soluzioni 1 (diminuire la **risoluzione**: sempre possibile, ma stando attenti a mantenere una qualità accettabile) e 3 (aumentare il livello di **compressione**: opzione disponibile per il formato JPEG).

Per modificare a piacere i vari parametri di un'immagine, compresa la risoluzione, si possono usare vari programmi di ritocco grafico. Alcuni di questi sono shareware, ossia gratuiti per un periodo di prova; per esempio:

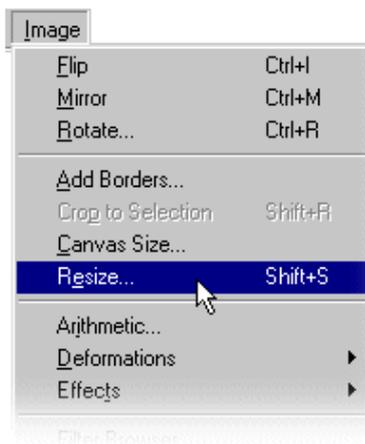
- Paint Shop Pro, in ambiente Windows (si può trovare sui CD allegati alle maggiori riviste di informatica per Windows oppure prelevare dalla Rete all'indirizzo <<http://it.jasc.com/>>);

Ridurre la risoluzione di un'immagine con Paint Shop Pro

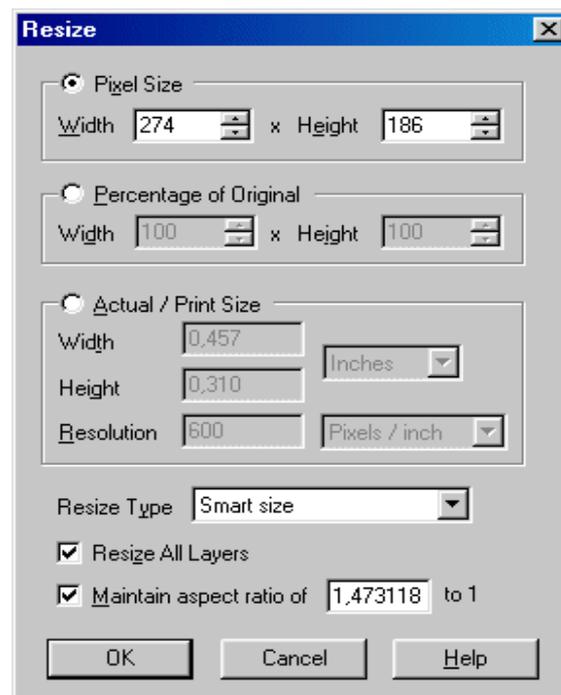
Aprire l'immagine originale nell'area di lavoro del programma (in questi esempi si fa riferimento alla versione 5 di Paint Shop Pro):



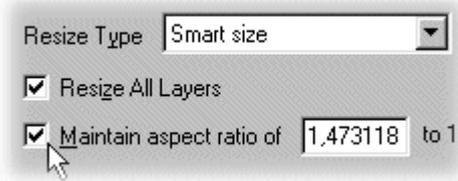
Cliccare sul menu **Image** e selezionare il comando **Resize**:



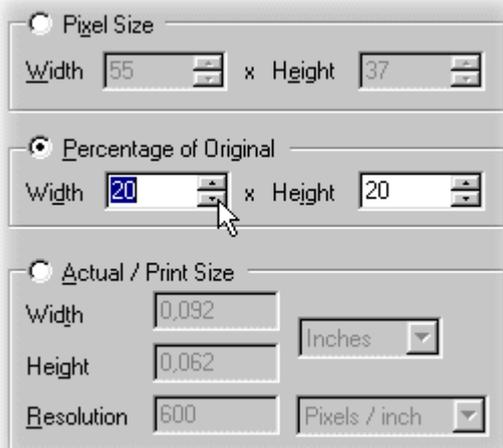
Si apre così la finestra **Resize**: qui si trovano tutte le opzioni necessarie per intervenire sulle dimensioni e sulla risoluzione dell'immagine. Si noti che esistono tre criteri alternativi per modificare la risoluzione: **Pixel Size** (si definiscono direttamente le nuove dimensioni in pixel: larghezza x altezza); **Percentage of Original** (si stabilisce un rapporto in percentuale rispetto alle dimensioni originali); **Actual/Print Size** (si impostano le dimensioni "effettive" a video e in stampa: in pollici/cm o punti per pollice/cm).



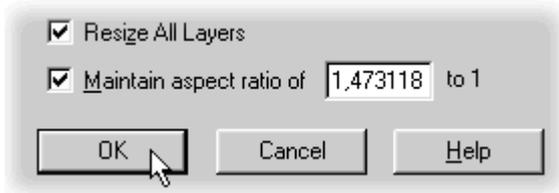
Prima di modificare i valori originali, è importante controllare che in basso siano selezionate queste due voci: **Resize Type: Smart Resize** (salvaguardia automatica della qualità visiva dell'immagine) e **Maintain aspect ratio** (salvaguardia automatica delle proporzioni larghezza/altezza, in modo da non deformare l'immagine).



A questo punto si può finalmente modificare la risoluzione dell'immagine: per semplicità, si può scegliere il criterio percentuale (Percentage of Original). Per ottenere una miniatura adatta agli scopi descritti nella sezione (per esempio: pagina-catalogo), si può ridurre l'immagine al 20% di quella originale. I valori sono comunque indicativi e bisogna scegliere il più opportuno caso per caso.



Confermare l'operazione cliccando sul pulsante **OK**.



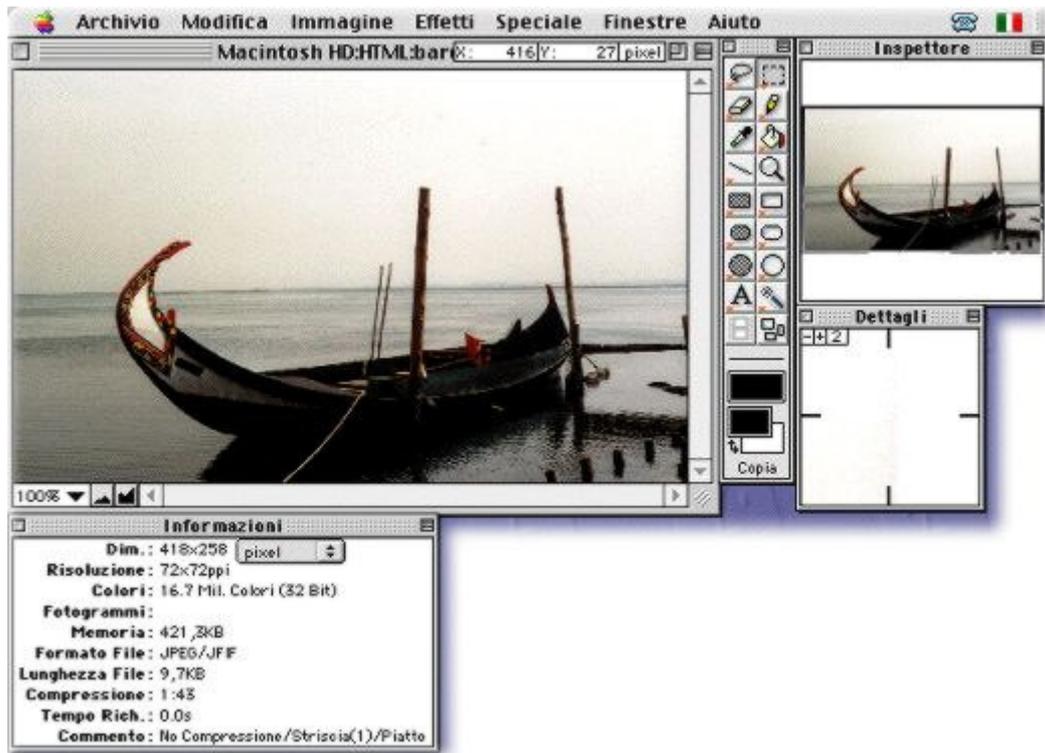
Ecco a confronto l'immagine originale e il risultato del ridimensionamento:



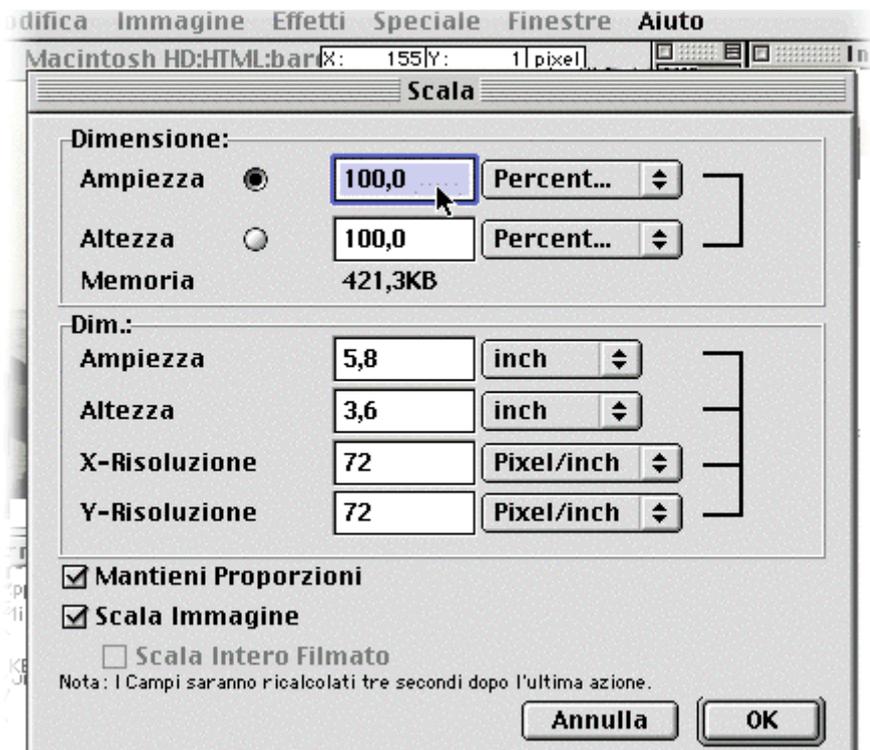
GraphicConverter, in ambiente Mac (si può trovare sui CD allegati alle maggiori riviste di informatica per Mac oppure prelevare dalla Rete all'indirizzo <www.lemkesoft.de/en/>).

Ridurre la risoluzione di un'immagine con Graphic Converter

Aprire l'immagine originale nell'area di lavoro del programma (in questi esempi si fa riferimento alla versione 3.7.2 di GraphicConverter):



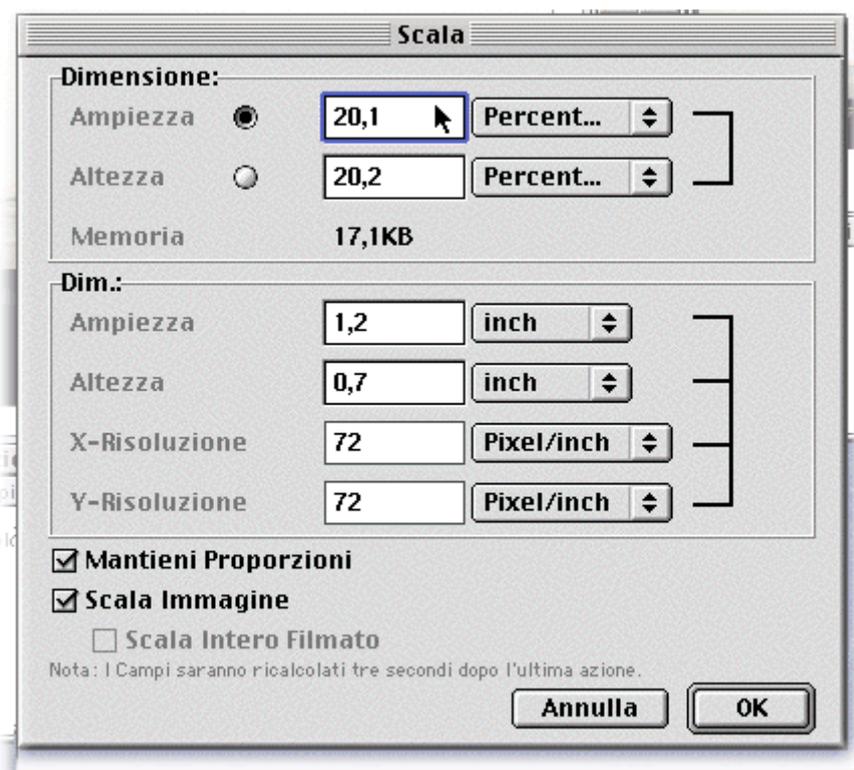
Dal menu **Immagine** selezionare il menu **Dimensione**, e da quest'ultimo il comando **Scala...**. Si apre così la finestra **Scala**: qui si trovano tutte le opzioni necessarie per intervenire sulle dimensioni e sulla risoluzione dell'immagine. Si noti che esistono diversi criteri alternativi per modificare la risoluzione: nel primo riquadro è possibile sia modificare le dimensioni come rapporto in percentuale rispetto all'originale, sia definire direttamente le nuove dimensioni in pixel; nel secondo riquadro si possono definire direttamente le dimensioni delle immagini in pollici (inch) o in centimetri, oppure modificare la risoluzione in pixel per pollice (pixel/inch).



Prima di modificare i valori originali, è importante controllare che in basso siano selezionate queste due voci: **Mantieni Proporzioni** (salvaguardia automatica delle proporzioni ampiezza/altezza, in modo da non deformare le immagini) e **Scala Immagine**.



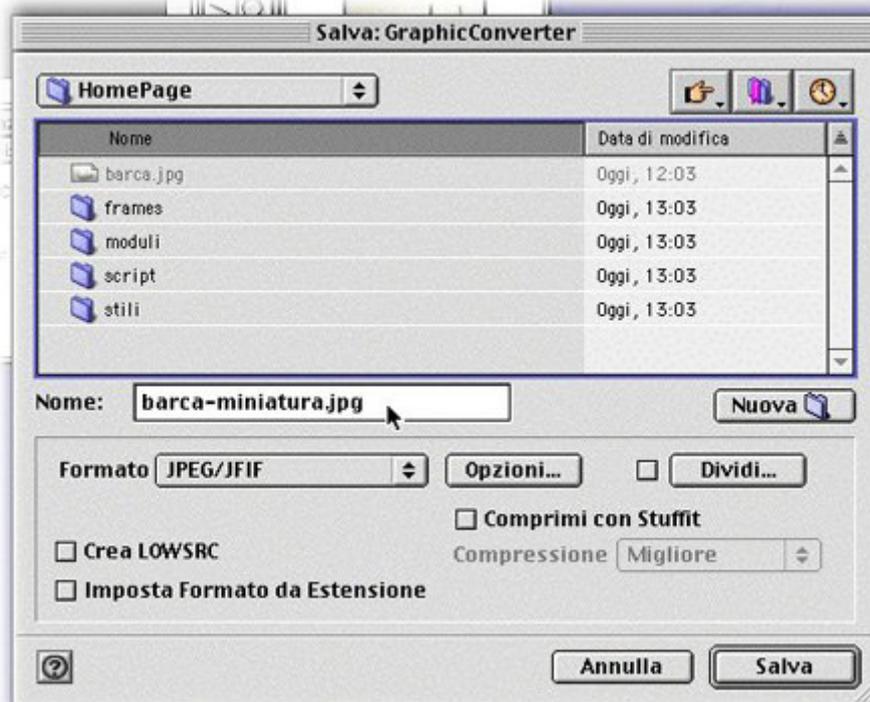
A questo punto si può finalmente modificare la risoluzione dell'immagine: per semplicità, si può scegliere il criterio percentuale (nel primo riquadro). Per ottenere una miniatura adatta agli scopi descritti nella sezione (per esempio: una pagina- catalogo), si può ridurre l'immagine al 20% di quella originale. I valori sono comunque indicativi e bisogna scegliere il più opportuno caso per caso (GraphicConverter modifica leggermente i valori percentuali immessi; le differenze nei risultati sono comunque trascurabili).



Confermare l'operazione cliccando sul pulsante **OK**. Ecco a confronto l'immagine originale e il risultato del ridimensionamento:



Salvare la versione ridotta dell'immagine con un altro nome, tramite il comando **Archivio/Salva come**. Per esempio, se l'immagine originale si chiama **barca.jpg**, salvare la versione a bassa risoluzione con un nome del tipo **barca- miniatura.jpg**: in questo modo si potrà scegliere di volta in volta il file più adatto alle circostanze, in base ai criteri illustrati nella sezione.



A proposito della profondità di colore, un possibile intervento di ottimizzazione riguarda la **palette (tavolozza)** associata all'immagine.

- In ogni file di immagine sono incorporati una serie di dati che indicano il numero massimo di colori previsto: in pratica, una tavolozza contenente tutti i colori possibili in base al parametro della profondità di colore (per esempio, 256 per le immagini GIF). I dati relativi a questa tavolozza influiscono sulle dimensioni complessive del file dell'immagine.
- Non è detto però che tutti i colori **possibili** siano effettivamente **presenti** nella specifica immagine: in una certa immagine GIF, per esempio, potrebbero essere concretamente utilizzati (visibili) solo 100 o 180 dei 256 colori totali consentiti dalla tavolozza.
- In questi casi (tramite un programma di grafica) è possibile eliminare dal file dell'immagine le informazioni relative ai colori non utilizzati, trasformando la **tavolozza standard** in una **tavolozza ottimizzata**, che contiene soltanto i colori presenti nell'immagine. Tutti i principali programmi di grafica (come PhotoShop o Paint Shop Pro) prevedono questa possibilità, anzi spesso eseguono automaticamente l'ottimizzazione.
- Le dimensioni del file possono così essere ridotte sensibilmente.

A titolo di esempio, ecco due versioni di una stessa immagine, una con tavolozza standard e l'altra con tavolozza ottimizzata:

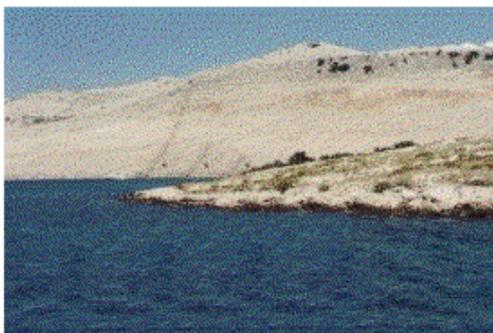
Versione 1 (21 KB)



Tavolozza standard



Versione 2 (19 KB)



Tavolozza ottimizzata



Come si può notare, nell'esempio proposto si ottiene un risparmio di circa il 10% sulle dimensioni del file. In alcuni casi il risparmio può essere notevolmente superiore. Nella seguente scheda sono riportati alcuni suggerimenti pratici per applicare le nozioni apprese in questa sezione.

Una pagina con catalogo miniature

Come si è detto, il numero e il "peso" (in termini di dimensioni dei file) delle immagini incidono significativamente sui tempi di caricamento delle pagine Web.

Quando possibile, è sempre meglio limitare al minimo entrambi i parametri (numero di immagini e ingombro di ciascun file).

Tuttavia, in particolari circostanze, una pagina Web **deve** contenere **numerose immagini**.

Ecco alcuni esempi pratici:

- un'azienda pubblica sul Web il catalogo dei suoi prodotti;
- un autore presenta sul Web il proprio curriculum con l'elenco dei libri pubblicati.

In questi casi, è normale che la pagina Web includa una serie di immagini: per esempio, le foto dei prodotti o le copertine dei libri.

Per motivi di impaginazione grafica e di leggibilità, ovviamente, è preferibile che tali immagini siano presentate in dimensioni ridotte, ossia sotto forma di **miniature** (in inglese: thumbnail).

Per esempio:



Si possono poi impostare opportuni **collegamenti ipertestuali** in modo che il lettore, cliccando su una miniatura, possa accedere a un'altra pagina contenente l'immagine ingrandita della copertina, del prodotto e così via.



Ebbene, per preparare la pagina iniziale, quella con l'elenco delle miniature, si hanno a disposizione due differenti soluzioni.

- **Soluzione (A):** inserire nella pagina le immagini vere e proprie, riducendone soltanto le dimensioni visibili mediante gli attributi di visualizzazione HEIGHT e WIDTH (v. prossima sezione).
 - *Vantaggi:* per ogni immagine è presente un solo file, dunque l'ingombro complessivo del sito resta invariato.
 - *Svantaggi:* il caricamento della pagina-catalogo risulta notevolmente appesantito, poiché i file delle immagini hanno un ingombro effettivo che resta inalterato (legato alla rispettiva risoluzione) anche se le dimensioni visibili appaiono ridotte.
- **Soluzione (B):** per ogni immagine, preparare un secondo file con risoluzione ridotta (miniatura); per esempio: per il file foto1.jpg preparare la versione in miniatura foto1_miniatatura.jpg e così via. A questo scopo si opera con un programma di grafica, come spiegato nell'esercizio precedente. Nella pagina-catalogo, inserire quindi solo le miniature (collegando ciascuna di esse, in modo ipertestuale, a un'altra pagina in cui caricare facoltativamente la versione originale).
 - *Vantaggi:* la pagina-catalogo viene caricata molto rapidamente (rispetto all'altra soluzione) in quanto le immagini che contiene sono effettivamente più piccole degli originali.
 - *Svantaggi:* per ogni immagine esistono due file (quello a piena risoluzione e quello in miniatura), dunque l'ingombro complessivo del sito aumenta.

Consiglio

Poiché ciò che conta è la rapidità di caricamento di ogni singola pagina del sito, è chiaro che **la soluzione (B) è altamente preferibile.**

Se si è interessati all'argomento, è consigliabile approfondirlo tramite corsi e manuali dedicati alla grafica digitale.

13. Immagini: inserimento nella pagina

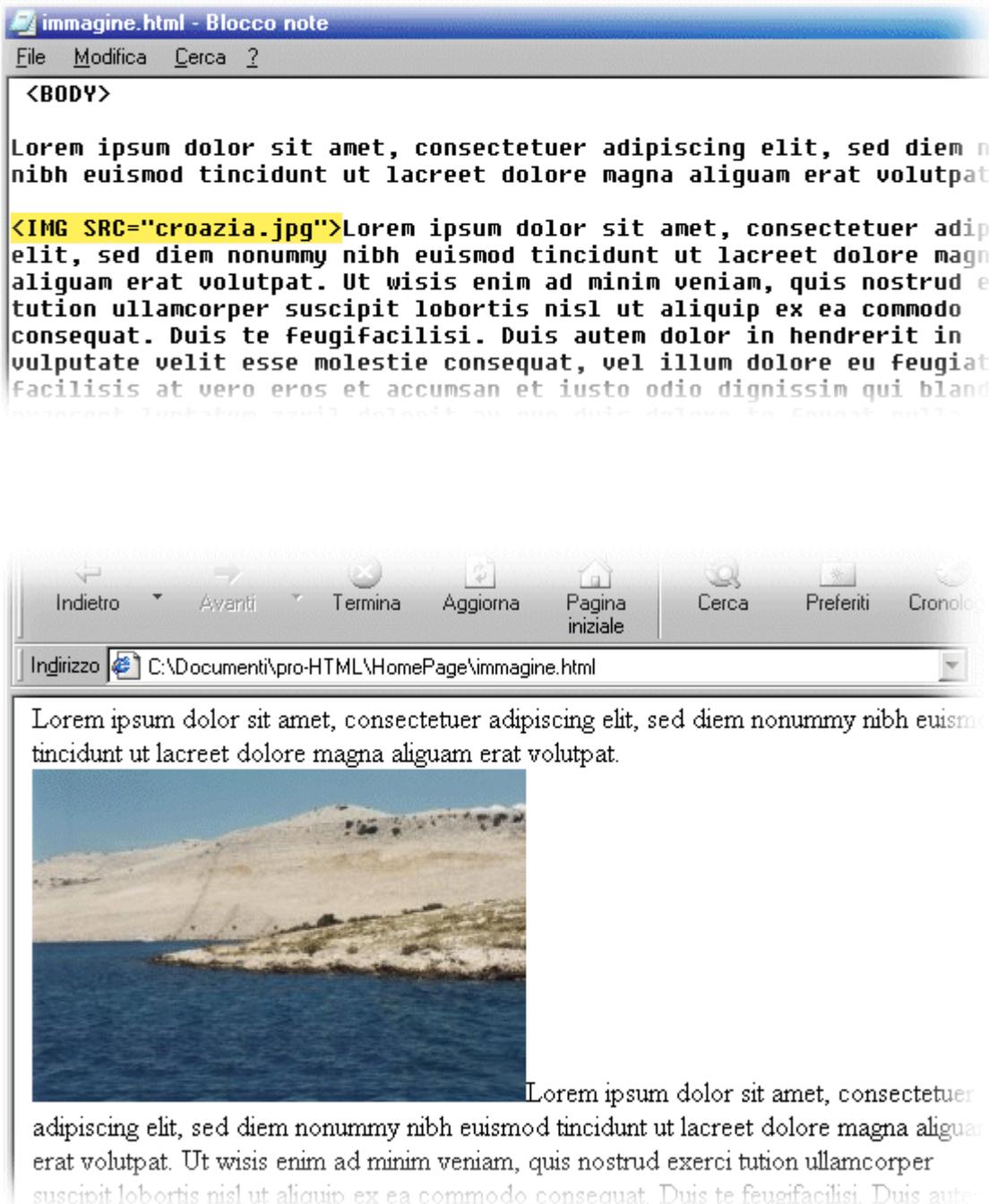
Inserimento di un'immagine con il tag

Per far apparire nella pagina un'immagine è sufficiente inserire nel codice HTML il tag ****.

Questo tag è singolo: **non** richiede cioè un'istruzione di chiusura del tipo **</fine_del_tag>**.

Al posto della stringa generica *nome_file* bisogna scrivere il nome del file dell'immagine.

Esempio:



Come già sottolineato nella sezione 5, se il file dell'immagine si trova nella stessa posizione (disco e cartella del server) in cui si trova il file HTML, allora è sufficiente indicarne il **nome**: in caso contrario è necessario specificarne il **percorso completo**. Si noti che è anche possibile

visualizzare nella pagina immagini che si trovano presso **altri server (siti Web)**.
Esempi:

- se l'immagine si chiama *croazia.jpg* e si trova nella sottocartella *Immagini*, all'interno della cartella del file HTML, il tag dovrà essere: ****;
- se l'immagine si chiama *croazia.jpg* e si trova nella cartella principale di un altro server (sito), il cui URL è *www.croazia.com*, il tag dovrà essere ****.

Opzioni e attributi

Osservando la figura precedente, si può notare che l'immagine viene inserita in modo "meccanico" e poco elegante: mantiene le sue dimensioni assolute (risoluzione) e viene allineata alla base della riga di testo corrispondente, senza spazi aggiuntivi.

Si hanno a disposizione vari attributi del tag per **personalizzare l'impaginazione delle immagini**.

Dimensioni: è possibile definire a piacere le dimensioni di visualizzazione di un'immagine, specificando gli attributi WIDTH (larghezza, in pixel) e HEIGHT (altezza, in pixel).
Esempio: ****.

Naturalmente occorre conoscere le dimensioni effettive (risoluzione) dell'immagine e mantenere inalterato il rapporto larghezza/altezza, se non si vuole deformare l'immagine. Un trucco molto utile: anche se si vuole visualizzare l'immagine a grandezza naturale, conviene *sempre* specificare i valori WIDTH e HEIGHT: in questo modo infatti il browser "sa" come collocare l'immagine e il caricamento della pagina è molto più rapido.

Allineamento: è possibile definire l'allineamento dell'immagine rispetto al testo circostante, mediante l'attributo ALIGN.

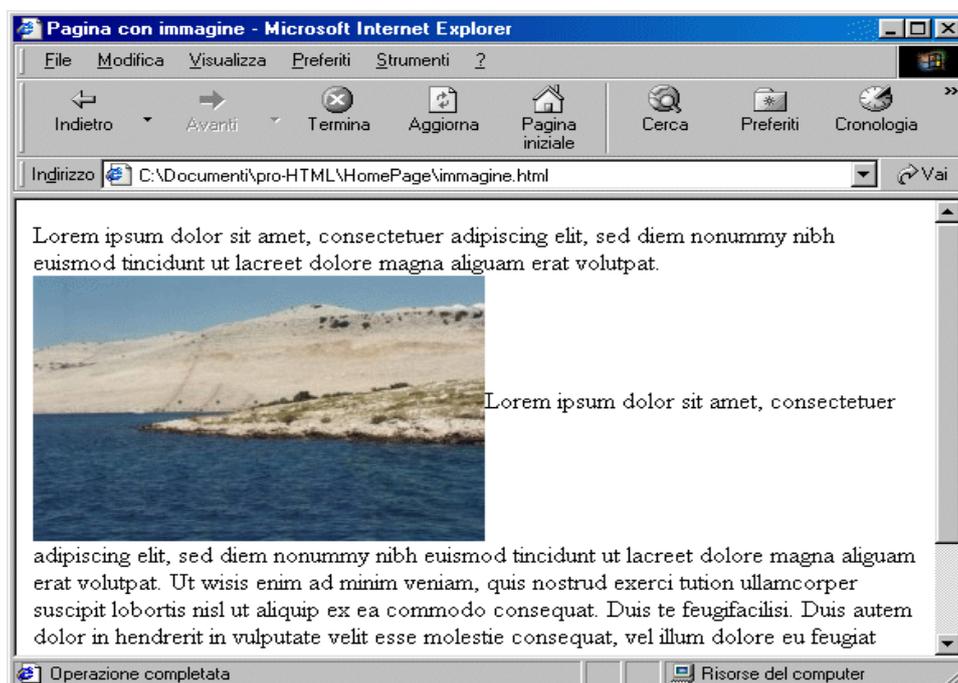
Esempio: ****.

I valori possibili sono: *bottom, middle, top, left, right*.

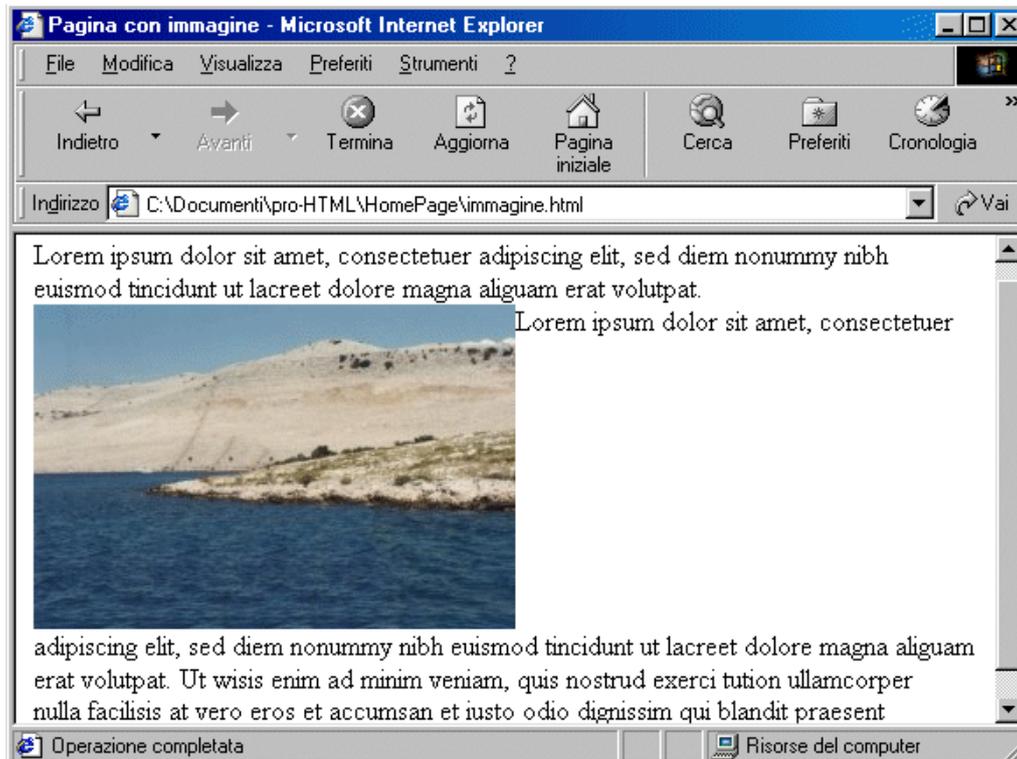
- **ALIGN="bottom"**: il lato inferiore dell'immagine appare allineato alla base della riga di testo;

è il valore predefinito in mancanza di altre indicazioni (v. figura precedente);

- **ALIGN="middle"**: il punto mediano dell'altezza dell'immagine si allinea alla base della riga di testo; al termine della riga il testo si spezza e prosegue sotto l'immagine;



- **ALIGN="top"**: il lato superiore dell'immagine si allinea alla parte superiore della riga di testo; al termine della riga il testo si spezza e prosegue sotto l'immagine;



- **ALIGN="left"**: l'immagine si posiziona sul lato sinistro della pagina e il testo scorre intorno a lei sul lato destro; in fase di visualizzazione tuttavia possono manifestarsi problemi di allineamento, a seconda degli attributi di impaginazione del testo circostante;



- **ALIGN="right"**: l'immagine si posiziona sul lato destro della pagina e il testo scorre intorno a lei sul lato sinistro; in fase di visualizzazione tuttavia possono manifestarsi problemi di allineamento, a seconda degli attributi di impaginazione del testo circostante.



- **Spazio circostante**: si può aumentare lo spazio vuoto che circonda l'immagine, allontanandola dal testo: basta specificare gli attributi VSPACE (spazio verticale, in pixel) e HSPACE (spazio orizzontale, in pixel).
Esempio: **** (sopra e sotto l'immagine resterebbero 20 pixel vuoti; a destra e a sinistra dell'immagine resterebbero 10 pixel vuoti).
- **Bordo**: si può aggiungere un bordo intorno all'immagine tramite l'attributo **BORDER="spessore"**. Lo spessore del bordo deve essere espresso in pixel.
Esempio: ****.



- **Etichetta/didascalia**: mediante l'attributo ALT (testo alternativo) si può definire un'etichetta o didascalia associata a un'immagine. La sintassi è: ****.

Questa breve descrizione testuale appare durante il caricamento della pagina (finché l'immagine non viene visualizzata) o quando si passa con il mouse sopra l'immagine.

Esempio: ****



Questa etichetta di testo andrebbe *sempre* specificata; rappresenta infatti un accorgimento molto utile da più punti di vista:

- i **visitatori** non sono costretti ad attendere il completo caricamento delle immagini, ma possono capirne in anticipo il contenuto;
- gli **speciali browser per non-vedenti** sono in grado di interpretare correttamente la pagina;
- i **motori di ricerca** hanno a disposizione un ulteriore elemento per catalogare con precisione gli argomenti-chiave della pagina.

Grazie a tutti questi attributi, che possono anche essere combinati in uno stesso tag ****, è possibile controllare con una certa flessibilità l'impaginazione delle immagini.

Immagini e collegamenti ipertestuali

A un'immagine può essere associato un collegamento ipertestuale, secondo la sintassi standard descritta nella sezione 5: è sufficiente includere il tag **** all'interno della coppia di tag **<A HREF>** e **** che definiscono il collegamento.

Esempio:

** **

In questo caso, facendo clic sull'immagine *croazia.jpg* si salterebbe alla pagina *descrizione_croazia.html*.

Quando un'immagine è attiva o cliccabile (è cioè associata a un collegamento), appare generalmente circondata da un bordo di colore blu.

Poiché questo effetto può risultare anti-estetico, è consigliabile **nascondere il bordo**

colorato: basta specificare all'interno del tag l'attributo **BORDER="0"**, in modo che il bordo resti invisibile (0 pixel).

Esempio:

```
<A HREF="descrizione_croazia.html"> <IMG SRC="croazia.jpg" BORDER="0" > </A>
```

Pagine con molte immagini/miniature

Prima di concludere la sezione, un suggerimento pratico.

Se si devono inserire molte immagini nella stessa pagina, si profilerebbe il rischio di rallentare eccessivamente il tempo di caricamento. Inoltre la pagina diventerebbe in sé molto lunga da sfogliare (sviluppo verticale).

Per l'impaginazione del documento, come già accennato nella sezione precedente, si può scegliere tra **due soluzioni**:

- inserire nella pagina tutte le immagini vere e proprie, riducendone però le **dimensioni visibili** grazie agli attributi del tag , specificando cioè: **** (v. sopra);
- creare delle **copie in miniatura** di tutte le immagini (dunque nuovi file, **a bassa risoluzione**), e inserire nella pagina queste miniature mediante il tag ****.

In entrambi i casi, si potrebbe poi associare a ogni immagine un collegamento ipertestuale che ne richiami la versione a dimensioni reali in una pagina a parte (v. sezione 5).

Tuttavia tra i due metodi c'è una grossa differenza:

- nel primo caso, nonostante le dimensioni *visibili* delle immagini siano ridotte, le *effettive* dimensioni dei file (in KB) restano inalterate: la necessità di visualizzare nella pagina tutte le immagini renderebbe perciò molto lento il caricamento;
- nel secondo caso, sebbene si debbano preparare appositamente i duplicati delle immagini originali, questi duplicati in miniatura hanno un "peso" (in KB) ridottissimo: il caricamento della pagina sarebbe perciò molto più rapido che nel caso precedente. Questa dunque è la soluzione da preferire.

Sezione 14. Immagini: mappe cliccabili

Ambiti di applicazione

Le mappe cliccabili possono risultare utili in varie circostanze.

Molto spesso, poi, rappresentano una soluzione persino elegante sul piano estetico, preferibile rispetto all'impiego di tante singole immagini slegate tra loro.

Ecco un paio di esempi pratici che possono rendere l'idea circa i possibili ambiti di applicazione delle mappe.

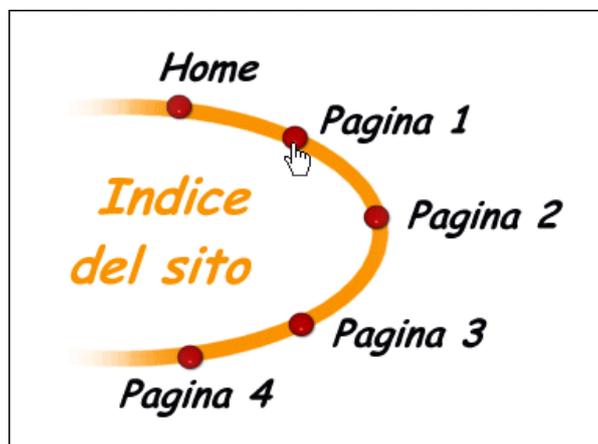
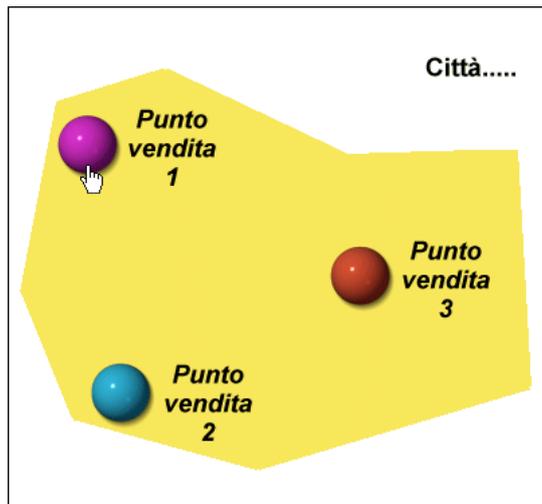
Come suggerisce ovviamente il nome stesso, una mappa cliccabile è l'ideale quando si devono trasmettere **informazioni di carattere geografico**.

Supponiamo di dover realizzare il sito di una catena commerciale dotata di vari punti vendita in una certa città.

Nella prima pagina del sito si potrebbe inserire una mappa cliccabile che riproduce la piantina della città. Ogni punto vendita è evidenziato da un simbolo sulla piantina.

Si potrebbe allora associare a ciascun punto della mappa un collegamento ipertestuale che rimanda a una pagina con i dettagli relativi al singolo punto vendita (orari di apertura, indicazioni stradali e così via). Molto spesso le mappe cliccabili vengono utilizzate per realizzare i **menu di navigazione** dei siti Web, approfittando del fatto che un'immagine di questo genere consente artifici grafici altrimenti impossibili.

Si osservi ad esempio la figura seguente: si tratta di un'unica immagine, con varie aree cliccabili (una per ogni voce del menu). Sarebbe impossibile ottenere questo risultato con tante immagini separate: l'unica soluzione è una mappa cliccabile.



Definizione di un'immagine come mappa: attributo USEMAP

Per costruire una mappa cliccabile occorre innanzitutto inserire nella pagina l'immagine di base, mediante il consueto tag (v. sezione precedente). All'interno del tag , però, bisogna aggiungere l'attributo USEMAP. La sintassi generica è:

```
<IMG SRC="nome_file" USEMAP="#nome_mappa">
```

Al posto di *nome_file* si deve scrivere il nome del file dell'immagine (con l'eventuale percorso). Al posto di *nome_mappa* si può scegliere un qualsiasi nome che convenzionalmente descriverà la mappa: servirà più avanti come riferimento (v. prossimo paragrafo).

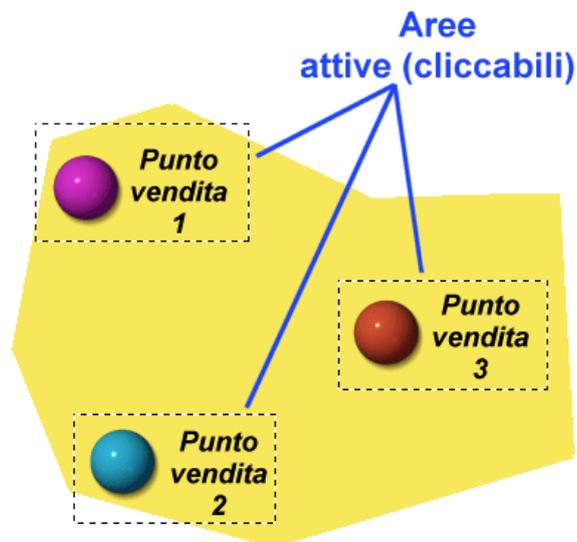
Ad esempio, per realizzare l'ipotetica mappa dei punti vendita sopra illustrata, supponiamo che l'immagine di base si chiami *mappa_pv.gif*. Nel codice HTML della pagina si dovrà allora inserire un tag come questo:

```
<IMG SRC="mappa_pv.gif" USEMAP="#punti_vendita">
```

In questo modo si stabilisce che:

- in un certo punto della pagina va visualizzata l'immagine *mappa_pv.gif*;
- tale immagine dovrà servire come mappa cliccabile;
- il nome convenzionale della mappa è *punti_vendita*.

Ora bisogna specificare il funzionamento della mappa, ossia quante e quali sono le aree cliccabili e a quali collegamenti ipertestuali devono essere associate.



Articolazione della mappa e dei collegamenti: tag <MAP> e <AREA>

Per specificare il tipo di funzionamento della mappa cliccabile si utilizza il tag <MAP> ... </MAP>.

Questa coppia di tag deve essere inserita dopo il tag .

Nel tag di apertura si deve indicare il **nome della mappa**: <MAP NAME="*nome_mappa*">.

La stringa *nome_mappa* deve corrispondere a quella dell'attributo USEMAP (v. sopra).

Per proseguire l'esempio, perciò, dopo il tag che richiama l'immagine *mappa_pv.gif* bisogna inserire questi tag:

```
<MAP NAME="punti_vendita">
```

</MAP>

In questo modo è stata predisposta la definizione della mappa.

Ora si devono individuare **le aree cliccabili**.

A tale scopo si devono inserire, tra <MAP> e </MAP> tanti tag **<AREA>** quante sono le aree desiderate, secondo questo schema semplificato:

```
<MAP NAME="punti_vendita">  
<AREA... --- definizione della prima area cliccabile>  
<AREA... --- definizione della seconda area cliccabile>  
<AREA... --- definizione della terza area cliccabile>  
</MAP>
```

All'interno di ogni tag <AREA>, per definire la corrispondente area cliccabile, si devono specificare tre **attributi**:

SHAPE="forma_dell'area": il valore da inserire al posto di *forma_dell'area* può essere **rect** (area rettangolare), **circle** (area circolare) oppure **poly** (area poligonale, irregolare);

- **COORDS="coordinate_pixel"**: le coordinate servono a fissare la posizione di ogni area cliccabile all'interno dell'immagine.

La posizione è definita, in generale, dalle **coordinate X e Y di alcuni punti di riferimento (pixel)**.

A seconda della forma dell'area, si assumono i seguenti punti di riferimento:

- se l'area è rettangolare (SHAPE="rect"), si devono indicare le coordinate del primo pixel in alto

a sinistra e dell'ultimo in basso a destra;

- se l'area è circolare (SHAPE="circle"), si devono indicare le coordinate del pixel centrale

e la lunghezza (sempre in pixel) del raggio;

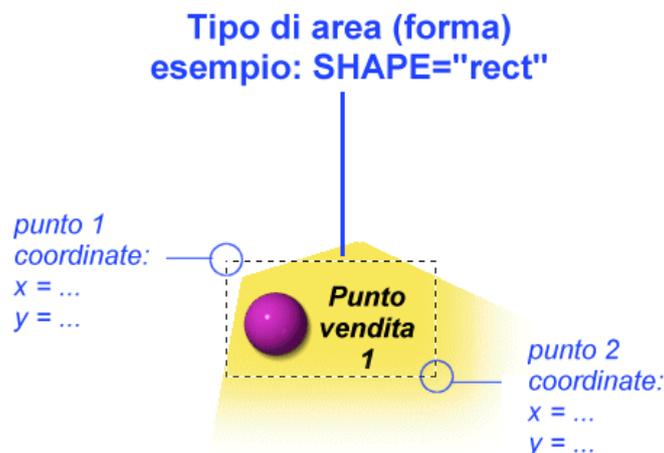
- se l'area è un poligono irregolare (SHAPE="poly"), si devono indicare le coppie di coordinate

di tutti i suoi vertici;

- **HREF="destinazione_collegamento"**: qui va indicata la destinazione del collegamento ipertestuale: un'altra pagina dello stesso sito Web, l'indirizzo di un differente sito Web e via dicendo (v. sezione 5).

Riprendendo il solito esempio, nell'ipotetica mappa dei punti vendita si devono creare tre aree cliccabili, di forma rettangolare.

Ciascuna area rettangolare viene identificata dalle coordinate di due punti.



Supponendo che in questa immagine le coordinate del punto 1 siano 30, 80, che quelle del punto 2 siano 172, 160 e che il collegamento debba rimandare alla pagina puntovendita1.html, il tag sarà allora:

```
<AREA SHAPE="rect" COORDS="30,80 , 172,160" HREF="puntovendita1.html">
```

Estendendo il ragionamento, l'intera mappa cliccabile dell'esempio potrebbe essere così costruita (i valori delle coordinate sono puramente indicativi):

```
<IMG SRC="mappa_pv.gif" USEMAP="#punti_vendita">
<MAP NAME="punti_vendita">
<AREA SHAPE="rect" COORDS="30,80 , 172,160" HREF="puntovendita1.html">
<AREA SHAPE="rect" COORDS="54,277 , 190,350" HREF="puntovendita2.html">
<AREA SHAPE="rect" COORDS="233,186 , 374,260" HREF="puntovendita3.html">
</MAP>
```

Se si è seguita la spiegazione fino a questo punto, il meccanismo di funzionamento generale dovrebbe essere abbastanza chiaro. Tuttavia resta legittimo un dubbio fondamentale: **come si calcolano le coordinate delle aree?**

E' qui che insorgono i problemi pratici.

Una mappa in azione...

A differenza dai consueti esercizi, questo si svolge all'inverso:

1. osservare e sperimentare un esempio di mappa cliccabile in azione;
2. visualizzare e studiare il codice HTML con cui è stata generata.

Si tratta quindi di un esempio pratico e interattivo, finalizzato a comprendere meglio il funzionamento delle mappe.

Questa scelta è dettata dal fatto che **non** conviene procedere nel modo tipico degli altri esercizi, ossia scrivere a mano il codice HTML necessario per l'impostazione di una mappa. La creazione delle mappe, nella pratica professionale, viene sempre effettuata mediante strumenti WYSIWYG: potenti editor Web (come quelli citati all'inizio del modulo: v. sezione 1) o programmi di grafica specializzati per tale funzione.

Ecco perciò i passi dell'esercizio.

Osservate la parte del codice evidenziata in giallo che inserisce l'**immagine** e ne definisce la funzione di **mappa**:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF">

<p><a name="inizio"></a>Fare clic sull'immagine per accedere alle varie se
(&quot;punti&quot;;) della pagina:</p>

<p align="center">

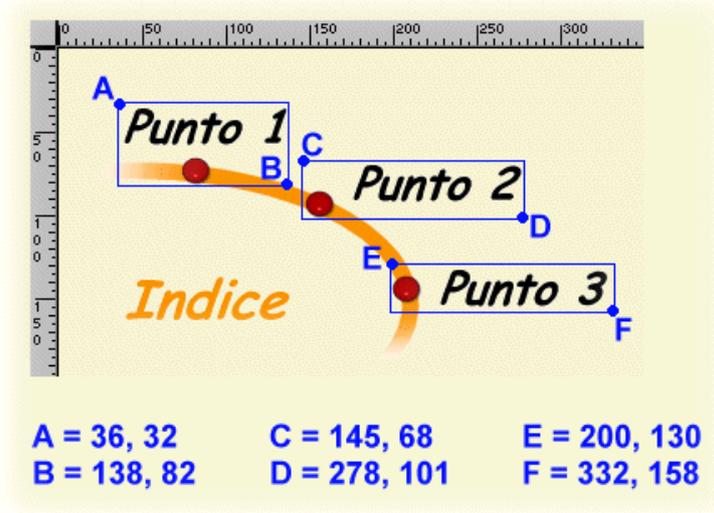


<map name="Map">
<area shape="rect" coords="36,32,138,82" href="#punto1">
<area shape="rect" coords="145,68,278,101" href="#punto2">
<area shape="rect" coords="200,130,332,158" href="#punto3">
</map>
```

Come sono state definite le **aree** della mappa?

Mediante un qualsiasi programma di grafica (come per esempio Paint Shop Pro) si calcola la

posizione dei punti-chiave che definiscono ciascuna area. Le **coordinate** di questi punti vengono quindi inserite come valore dell'attributo COORDS di ogni tag <AREA> (vedi sopra).



Per completezza, studiando il codice HTML dell'esempio converrebbe analizzare anche il modo in cui a ogni area della mappa è associato un punto particolare all'interno della stessa pagina. In teoria, i **collegamenti** associati alle aree della mappa possono puntare a **qualsiasi riferimento** (come ad altre pagine dello stesso sito oppure a indirizzi Web di altri siti). In questo caso, invece, si è sfruttata l'opzione delle **ancore ipertestuali all'interno della stessa pagina** (v. sezione 5).

Sono state inserite nella pagina tre ancore associate ai vari punti (chiamate rispettivamente "#punto1", "#punto2" e "#punto3"), più un'ancora per consentire il ritorno immediato all'inizio della pagina (chiamata "#inizio").

Problemi pratici

Effettivamente, la costruzione delle mappe cliccabili non è un'operazione complessa di per sé, se non fosse per l'ostacolo appena citato.

Infatti, mentre per impostare il codice HTML della mappa si può agire nel modo consueto, con il normale editor di testo, è materialmente impossibile calcolare le coordinate delle aree cliccabili se non si ricorre a un **programma esterno**.

Per calcolare le coordinate delle aree di un'immagine si può utilizzare:

- un **editor di immagini**, come PhotoShop o Paint Shop Pro: aprendo l'immagine nell'editor, la barra di stato mostra istante per istante le coordinate del pixel su cui si trova il puntatore del mouse;
- un **apposito programma di utilità**, specializzato in questo compito: se ne trovano parecchi nel circuito shareware, come ad esempio MapThis; un programma del genere consente di definire in modo grafico e intuitivo le aree cliccabili dell'immagine e genera poi automaticamente i tag HTML da copiare e incollare nel codice della pagina;
- un **editor Web WYSIWYG** (come ad esempio DreamWeaver o FrontPage): quasi tutti i migliori software commerciali di tale categoria prevedono qualche opzione in proposito. Molti anzi automatizzano quasi completamente la creazione delle mappe.

Con queste avvertenze si conclude il presente modulo del modulo.

Prima di procedere, ecco ora un esercizio che consente di ricapitolare e mettere a punto le nozioni fin qui apprese.

Esercitazione

Per ricapitolare gli argomenti salienti di questa lezione, svolgere il seguente esercizio:

1. preparare **una** pagina Web, contenente un testo a piacere;
2. definire **colori** personalizzati per ogni elemento della pagina: sfondo, testo, collegamenti ipertestuali; cercare di scegliere abbinamenti cromatici il più possibile adeguati, in base alle raccomandazioni fornite nella sezione 11;
3. inserire nella pagina una o più **immagini** a piacere, personalizzandone l'allineamento; le immagini possono essere personali oppure scaricate da un sito Web (in caso di dubbio, chiedere consiglio al tutor);
4. associare un **collegamento** ipertestuale (con riferimento a un qualunque indirizzo di sito Web) almeno a una delle immagini; evitare che venga visualizzato il bordo dell'immagine, adottando gli opportuni attributi.

Sezione 15. Le tabelle: uno strumento flessibile

Rappresentazione schematica di dati

In primo luogo, ovviamente, le tabelle HTML assolvono alla loro tipica funzione di base: rappresentare in modo ordinato e schematico dati di ogni genere. Per esempio:

Codice	Articolo	Quantità
100.256.0000A	Giunto verticale sinistro	1000
100.256.0000B	Giunto verticale destro	2500
100.374.0022X	Snodo a gomito trasversale	867

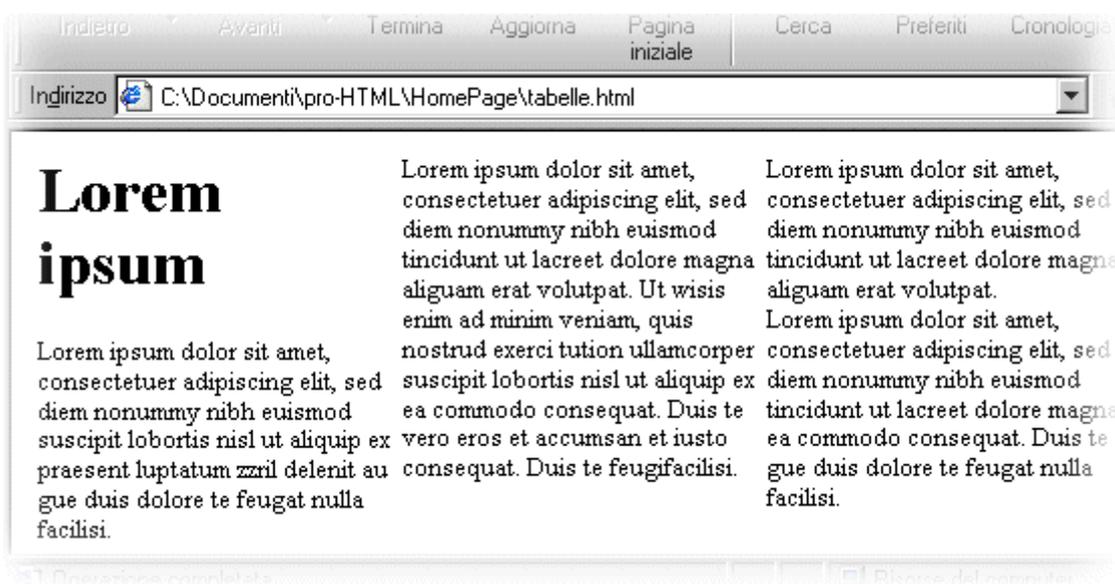
Questo genere di applicazione non comporta particolari difficoltà. Sarà sufficiente:

1. impostare la tabella mediante i tag fondamentali (v. sezione 16);
2. definire la visibilità/invisibilità dei bordi e il loro spessore (v. sezione 17);
3. formattare liberamente i contenuti (v. sezione 18);
4. eventualmente, adottare alcuni tag e attributi supplementari per definire meglio l'intestazione delle colonne.

Testo: colonne e rientri

Grazie alle tabelle, si possono superare i limiti dei tag HTML legati alla formattazione del testo. Come si è visto nelle sezioni da 6 a 10, il linguaggio HTML non offre grande flessibilità per la gestione del testo rispetto ai programmi di videoscrittura. Ebbene, ricorrendo alle tabelle è possibile ottenere risultati altrimenti impraticabili.

- Per esempio, solo in questo modo, si può disporre il testo su **colonne multiple**, con un'impaginazione simile a quella delle riviste su carta:



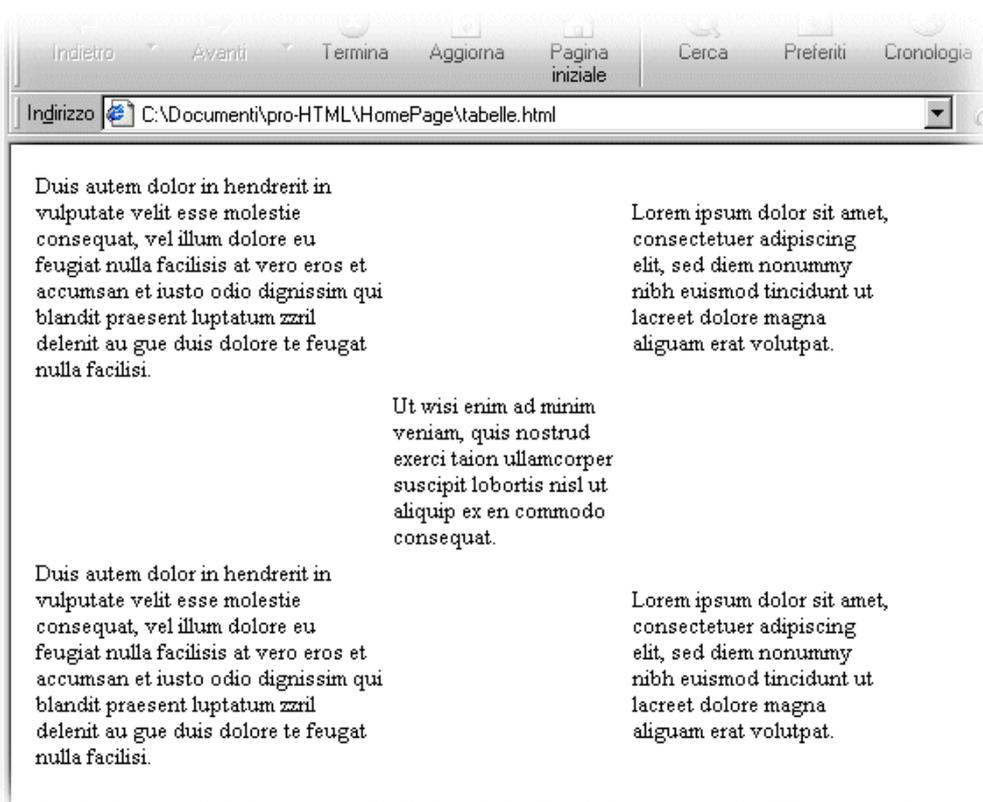
Per ottenere un effetto del genere, basta creare una tabella con bordi invisibili:

<h1> Lorem ipsum</h1> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi. Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit au gue duis dolore te feugiat nulla facilisi.</p>	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi. Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit au gue duis dolore te feugiat nulla facilisi. Ut wisi enim ad minim veniam, quis nostrud exerci taion ullamcorper suscipit lobortis nisl ut aliquip ex en commodo consequat. Duis te feugifacilisi.</p>	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi. Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit au gue duis dolore te feugiat nulla facilisi.</p>
--	--	--

L'unico limite, in questo caso, è che il testo non scorre automaticamente da una colonna all'altra: occorre ripartire a mano i blocchi di testo tra le varie celle della tabella.

- Un'altra interessante opportunità offerta dalle tabelle, è la realizzazione di **rientri variabili**, in alternativa a un utilizzo improprio e forzato di tag come <BLOCKQUOTE> e simili.

Per esempio:



Per raggiungere questo risultato basta impostare una tabella con il numero necessario di colonne (celle) e inserire il testo solo in quelle desiderate, lasciando vuote le altre.

Duis autem dolor in hendrent in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisis.		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
	Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.	
Duis autem dolor in hendrent in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisis.		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Per maggiore precisione, però, sarebbe bene inserire comunque nelle celle vuote un contenuto/segnaposto: un testo o un'immagine formattati in modo tale da renderli invisibili. In gergo tecnico, questi elementi sono chiamati **spaziatori** (spacer). Solo così si ha la garanzia che gli allineamenti previsti vengano rispettati durante la visualizzazione nel browser.

Gli spaziatori

Per distanziare con precisione gli elementi contenuti in una tabella, si può ricorrere ai cosiddetti spaziatori.

In pratica, poiché le opzioni HTML per l'allineamento non offrono sufficienti garanzie di precisione, è consigliabile ricorrere ai seguenti artifici.

- Per la spaziatura verticale nelle celle (ossia per lasciare una quantità fissa di spazio vuoto in rapporto all'altezza di una cella), si possono inserire **caratteri "invisibili"**, come un punto a cui viene assegnato il colore di sfondo; se lo sfondo della cella è bianco, lo spaziatore potrebbe essere di questo tipo:

** .
**

Un tag come questo introduce un punto bianco (dunque invisibile) e un ritorno a capo (interruzione di linea).

Se lo sfondo della cella non è bianco, basta modificare l'attributo di colore del punto. In alternativa, si può ricorrere anche per la spaziatura verticale all'impiego di immagini invisibili (v. punto seguente).

- Per la spaziatura orizzontale nelle celle (ossia per regolare con precisione la larghezza di una cella di per sé priva di contenuti), si può utilizzare **un'immagine "invisibile"**, come una .GIF trasparente (o riempita dello stesso colore dello sfondo); ecco come procedere:

1. con un programma di grafica, creare una semplice immagine che consiste in un quadratino di piccole dimensioni (per esempio: 1x1 pixel o 10x10 pixel);
2. riempire il quadratino con il colore dello sfondo della pagina (o della cella); meglio ancora sarebbe salvare l'immagine in formato .GIF e definire come trasparente il suo riempimento (tutti i migliori programmi di grafica offrono questa opzione, tipica del formato .GIF);
3. salvare l'immagine con un nome che la renda facilmente riconoscibile (per esempio: "spaziatore1x1.gif" o "spaziatore10x10.gif");
4. nel codice HTML della pagina, là dove si vuole creare una certa quantità di spazio vuoto, inserire questa immagine mediante il tag `` o ``; se necessario, inserire più copie della stessa immagine per aumentare la quantità di spazio.

Consiglio

È meglio ricorrere alle immagini anziché ai caratteri di testo "invisibili". I caratteri (punti, linee o altro) non offrono infatti sufficienti garanzie di regolarità nell'allineamento; tuttavia, rappresentano una soluzione accettabile nel caso della spaziatura verticale.

In ogni caso, l'aspetto interessante di questa soluzione è che gli spaziatori (testo o piccole immagini) sono pressoché **ininfluenti riguardo all'ingombro** e dunque ai tempi di caricamento della pagina.

Allineamento preciso delle immagini

Come si è detto nella sezione 13, l'allineamento delle immagini in una pagina Web presenta alcune difficoltà e restrizioni: i tag e gli attributi di base, infatti, non consentono adeguata libertà né precisione.

Utilizzando le tabelle, invece, è possibile disporre le immagini nel punto esattamente desiderato e allinearle al testo con una certa flessibilità.

Per esempio, se un'immagine deve essere affiancata al testo, la soluzione ideale è proprio il ricorso a una tabella (con bordo invisibile):



Il ricorso alle tabelle consente inoltre di ottenere **allineamenti complessi**, altrimenti impossibili o estremamente difficili da realizzare.

Per esempio:

Duis autem dolor in hendrent in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisis.		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
	Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex commodo consequat.	
Duis autem dolor in hendrent in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisis.		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Layout di pagina

Infine, le tabelle possono essere utilizzate (e lo sono spesso) per costruire l'intera struttura di una pagina Web.

In termini tecnici, questa struttura viene chiamata **layout** o **gabbia**.

Tipico, per esempio, è il ricorso a una tabella per posizionare nella pagina un'area **menu** (o **barra di navigazione**), visivamente distinta e separata dall'area principale del documento (testo e immagini):

Si tenga presente che in questo caso, in alternativa alle tabelle, si può anche ricorrere ai **frame** (v. sezione 19 e successive).

L'impiego delle tabelle, rispetto ai frame, presenta vantaggi e svantaggi che saranno in seguito discussi.

Personalizzando la struttura e gli attributi delle tabelle, si possono anche costruire **layout molto sofisticati**.

Una volta comprese le potenzialità di utilizzo delle tabelle HTML, è giunto il momento di imparare a gestirle nella pratica.

<p>ErgoSum an ERGONOMICS SUMMARY</p> <p><i>Federico Guzzi</i> <small>English abstract here</small></p> <p>Contenuti: Ergonomia Ergonomia psicologica Manualistica Le mie attività</p> <p>Divagazioni: Il mio vecchio sito personale Moto Guzzi custom</p>	<p>Benvenuti al sito ErgoSum. Lo scopo di queste pagine è fornire una panoramica sul rapporto tra ergonomia, psicologia e informatica (per studenti o semplici curiosi). Per informazioni più approfondite consultare il sito della Società Italiana di Ergonomia. Pubblico qui anche una presentazione delle mie attività professionali, come occasione di scambio e contatto.</p> <p>Ergonomia La disciplina che promuove l'adattamento del lavoro all'uomo (e non viceversa). Informazioni generali, cenni storici e indirizzi Web. ▶</p> <p>Ergonomia psicologica Il punto di contatto tra psicologia, scienze cognitive e informatica. Informazioni generali, cenni storici, bibliografia e indirizzi Web. ▶</p> <p>Manualistica Considerazioni sulla progettazione di manuali e guide (technical writing). Informazioni generali, bibliografia, indirizzi Web. ▶</p> <p>Le mie attività Il mio profilo professionale: libri di informatica, manuali, guide on-line. Curriculum vitae. ▶</p>
---	---

Sezione 16. Impostazione di una tabella

Creazione della tabella

Per creare una tabella si utilizza la coppia di tag **<TABLE> ... </TABLE>**.

Per definire la struttura della tabella, nei programmi di videoscrittura i parametri sono: numero di righe e numero di colonne.

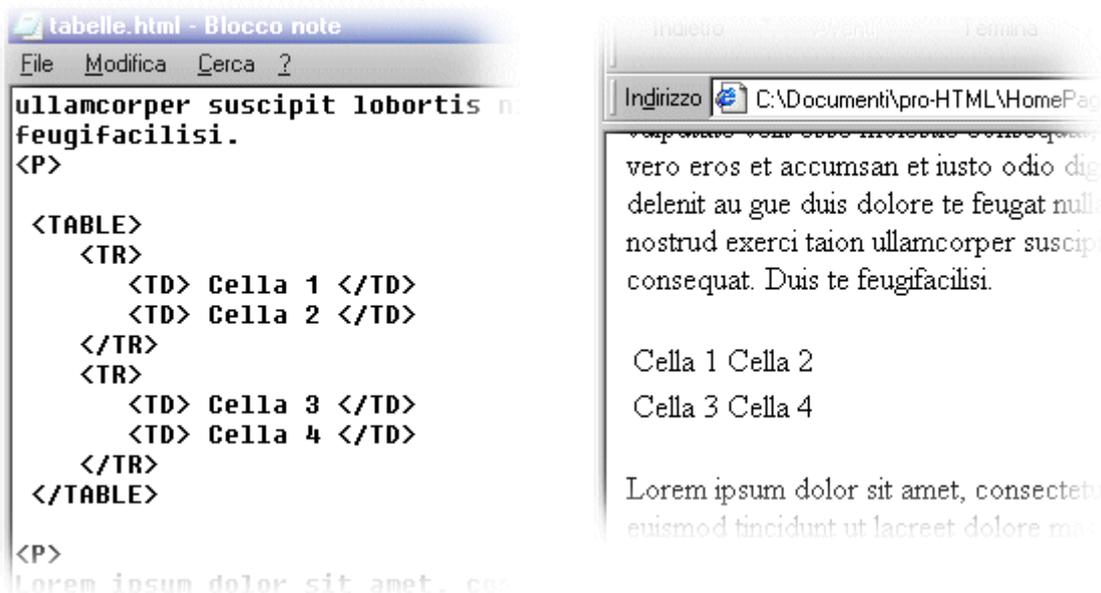
Nel linguaggio HTML, invece, si usa un criterio diverso. Nell'area compresa tra i tag **<TABLE>** e **</TABLE>** occorre infatti:

1. creare le **righe** desiderate: ogni riga viene definita tramite la coppia di tag **<TR> ... </TR>** ("TR" sta per "table row": riga di tabella);

2. all'interno di ogni riga, creare le **celle** desiderate: ogni cella viene definita tramite la coppia di tag **<TD> ... </TD>** ("TD" sta per "table data": dati/celle di tabella).

Normalmente ogni riga contiene un uguale numero di celle: in questo modo la tabella comprende un **numero fisso di colonne**. Tuttavia è possibile, facoltativamente, impostare un numero variabile di celle per ogni riga della tabella, costruendo così tabelle dalla struttura "irregolare". Lo si vedrà al termine della prossima sezione.

Ecco un esempio di semplice tabella:



Il **bordo** delle celle di solito resta invisibile (ma dipende anche dalle impostazioni dei singoli browser).

Per renderlo visibile si deve specificare un apposito attributo (v. prossima sezione).

Larghezza

Se non si specifica diversamente, la tabella si estende orizzontalmente in modo da occupare l'intera larghezza della pagina.

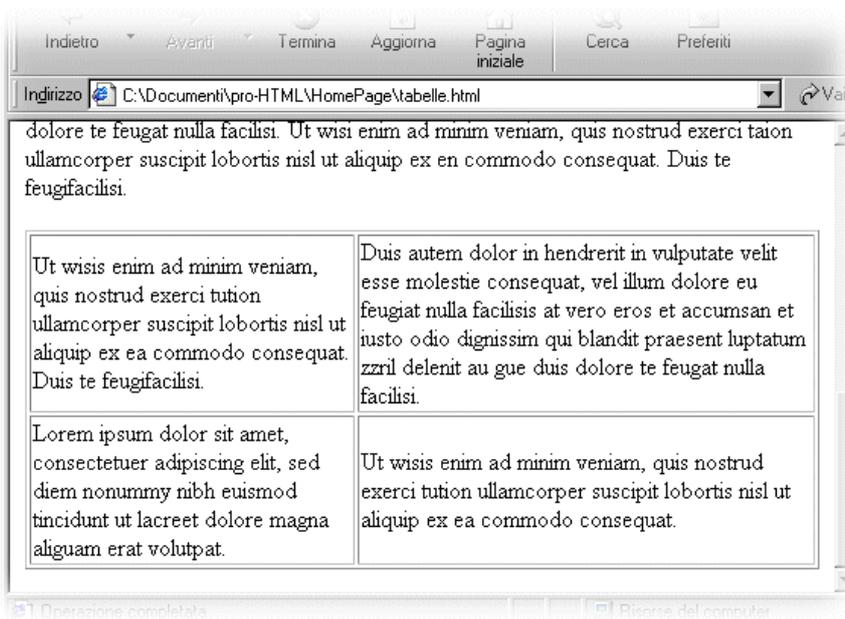
Per definire a piacere la larghezza della tabella si può utilizzare l'attributo **WIDTH="valore"**, da inserire all'interno del tag di apertura **<TABLE>**.

Il valore dell'attributo può essere espresso:

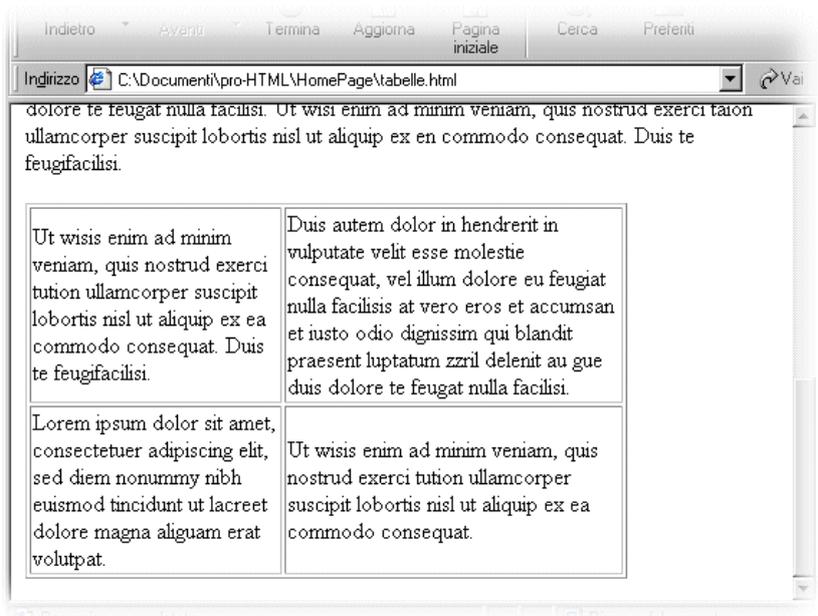
- in **pixel** (un numero intero, con valore assoluto); per esempio: **<TABLE WIDTH="500"> ... </TABLE>**;
- in **percentuale** (rispetto alla larghezza della pagina); per esempio: **<TABLE WIDTH="80%"> ... </TABLE>**.

Ecco alcuni esempi (per comodità viene qui visualizzato il bordo della tabella):

Larghezza standard:



Larghezza 80%:



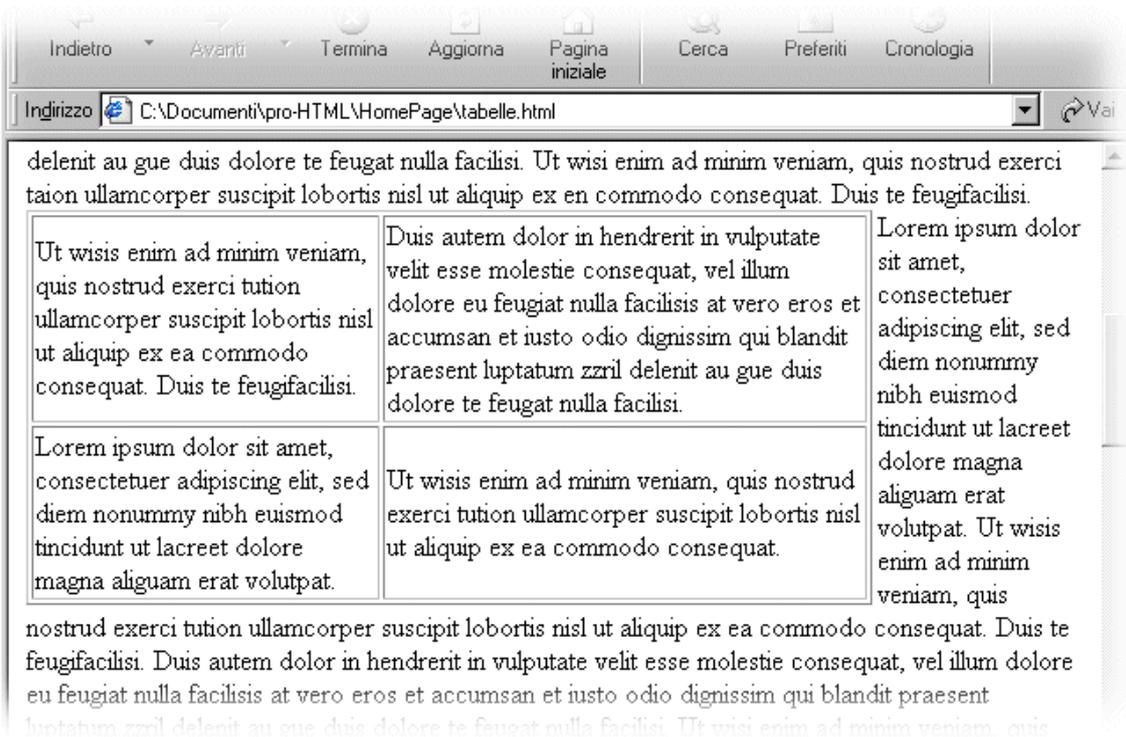
Allineamento nella pagina

Se la tabella ha una larghezza inferiore a quella della finestra, è possibile sceglierne l'allineamento rispetto ai margini: a sinistra, al centro o a destra.

A questo scopo si utilizza l'attributo **ALIGN="valore"**, da inserire all'interno del tag di apertura **<TABLE>**.

Supponendo di avere una tabella con larghezza 80% rispetto alla pagina, ecco come impostarne l'allineamento (anche in questo caso, per comodità, si mostra il bordo della tabella):

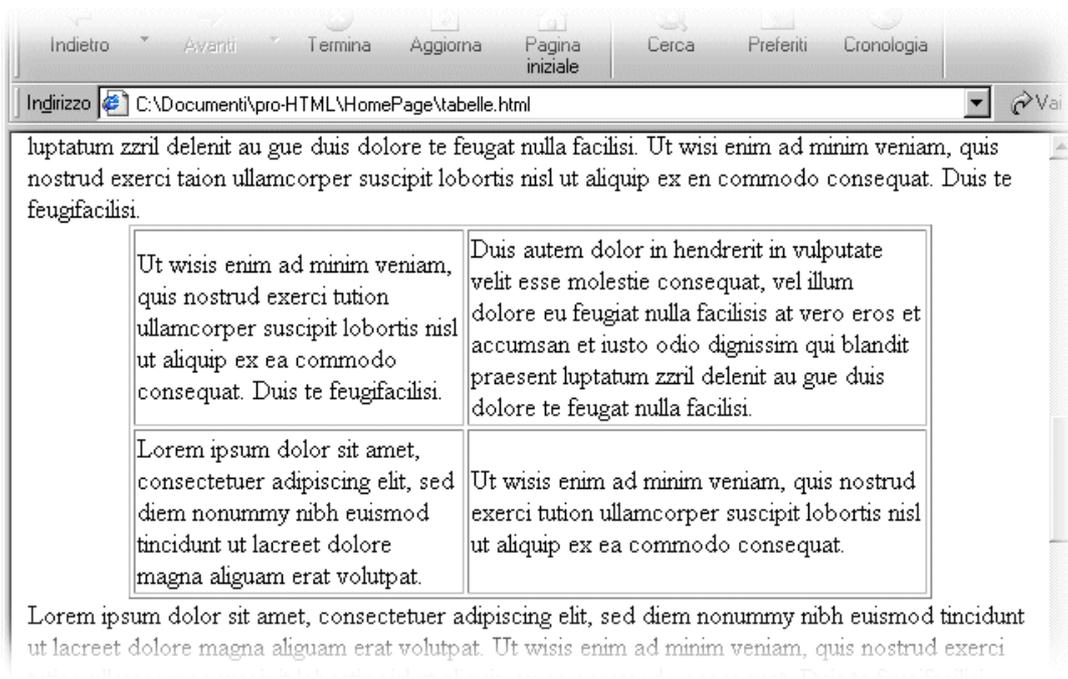
- allineamento a **sinistra** (allineamento predefinito):
<TABLE WIDTH="80%" ALIGN="left"> ... </TABLE>
il testo della pagina scorre automaticamente intorno al bordo destro della tabella;



- allineamento al **centro**:

<TABLE WIDTH="80%" ALIGN="center"> ... </TABLE>

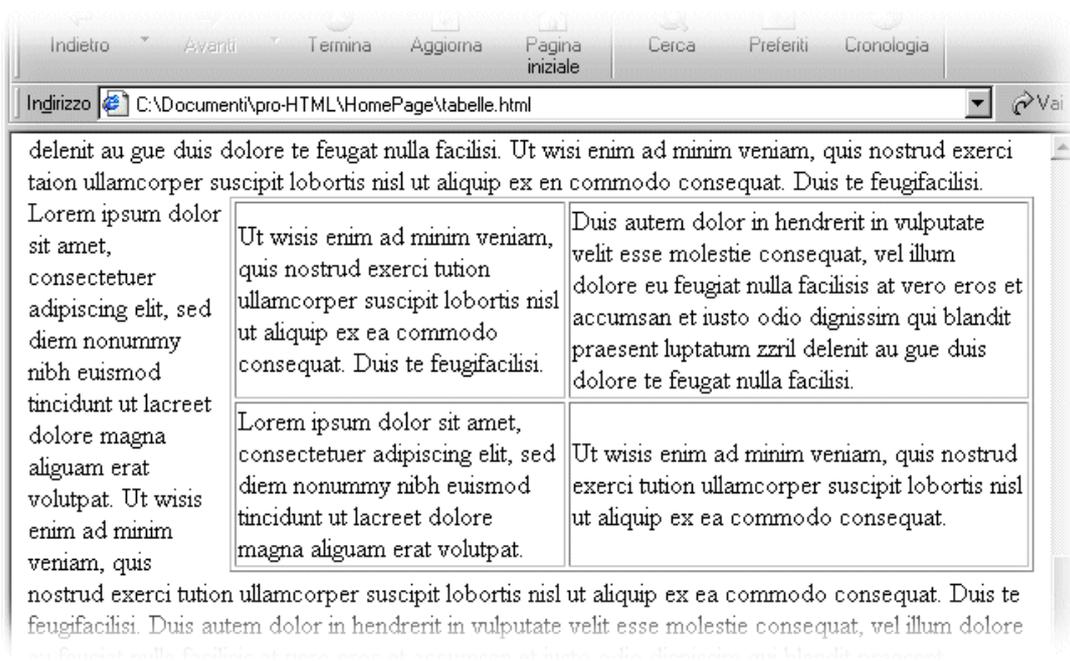
il testo si interrompe prima della tabella e prosegue sotto la tabella;



- allineamento a **destra**:

<TABLE WIDTH="80%" ALIGN="right"> ... </TABLE>

il testo scorre automaticamente intorno al bordo sinistro della tabella.



Opzioni aggiuntive

Secondo la sintassi HTML in una tabella si possono aggiungere elementi particolari, come una didascalia per l'intera tabella e un'intestazione per ciascuna colonna.

Nella pratica, queste opzioni sono poco utilizzate (spesso si ricorre ad altre soluzioni di formattazione). Esse sono:

- **Didascalia** della tabella: subito dopo il tag di apertura `<TABLE>` si può aggiungere la coppia di tag `<CAPTION> ... </CAPTION>`.

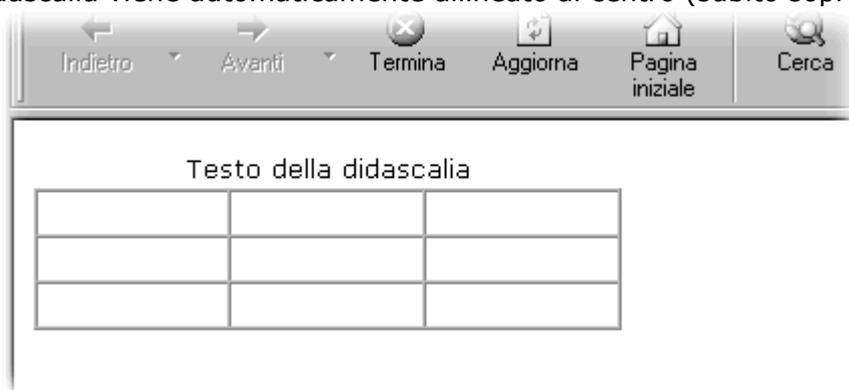
Per esempio:

```
<TABLE>
<CAPTION> Testo della didascalia </CAPTION>
```

...

```
</TABLE>
```

Il testo della didascalia viene automaticamente allineato al centro (subito sopra la tabella).



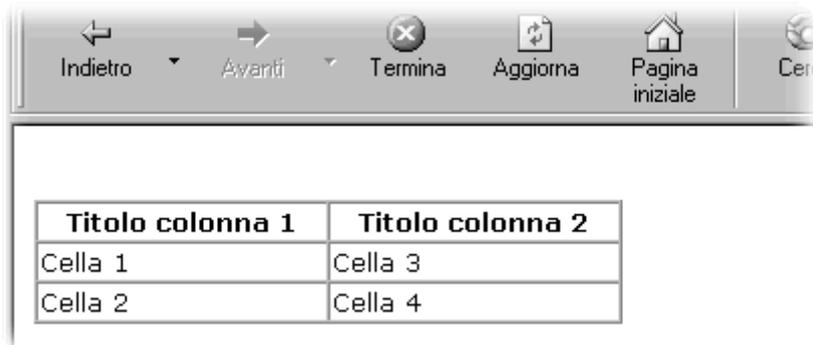
- **Intestazione** delle colonne: in una riga della tabella (di solito nella prima), al posto della normali celle `<TD> ... </TD>` si possono inserire le celle speciali `<TH> ... </TH>`, che fungono da intestazione delle rispettive colonne.

Per esempio:

...

```
<TR>
<TH> Titolo colonna 1 </TH> <TH> Titolo colonna 2 </TH>
```

```
</TR>
<TR>
<TD> Cella 1 </TD> <TD> Cella 2 </TD>
</TR>
<TD> Cella 3 </TD> <TD> Cella 4 </TD>
</TR>
...
```



The image shows a browser toolbar with buttons for 'Indietro', 'Avanti', 'Termina', 'Aggiorna', 'Pagina iniziale', and 'Cerca'. Below the toolbar is a table with two columns and two rows. The first row contains the headers 'Titolo colonna 1' and 'Titolo colonna 2'. The second row contains 'Cella 1' and 'Cella 3'. The third row contains 'Cella 2' and 'Cella 4'.

Titolo colonna 1	Titolo colonna 2
Cella 1	Cella 3
Cella 2	Cella 4

La formattazione del testo di queste celle di intestazione viene gestita automaticamente dal browser (generalmente con un grassetto); per il resto si possono utilizzare gli stessi attributi validi per <TD> (v. prossima sezione).

Oltre alle opzioni precedenti, sono disponibili attributi per l'impostazione dei **bordi** e dello **sfondo** delle celle.

Sebbene gran parte di questi attributi sia legata al tag generale <TABLE>, essi vengono presentati nelle prossime sezioni, che riguardano la personalizzazione e la formattazione delle tabelle.

Sezione 17. Personalizzazione di bordi e celle

Bordo

Se non si imposta alcun valore, il bordo delle celle resta invisibile.

Per definire il bordo della tabella si utilizza l'attributo **BORDER**, da inserire nel tag di apertura <TABLE>:

<TABLE BORDER="spessore"> ... </TABLE>

Lo spessore deve essere espresso da un numero intero di pixel (da 1 in su): l'effetto visivo in realtà dipende anche da altri parametri (come la spaziatura tra le celle: v. più avanti nella sezione).

Esempio:

Cella 1	Cella 2
Cella 3	Cella 4

BORDER=1

Cella 1	Cella 2
Cella 3	Cella 4

BORDER=3

Cella 1	Cella 2
Cella 3	Cella 4

BORDER=5

Cella 1	Cella 2
Cella 3	Cella 4

BORDER=10

Alcuni browser supportano anche l'attributo **BORDERCOLOR="colore"**, per assegnare al bordo un colore, espresso mediante i consueti codici esadecimali (v. sezione 11).

Tuttavia questa aggiunta non rientra nella sintassi HTML standard e, considerando i problemi di compatibilità tra browser, è sconsigliabile.

Sfondo

Per impostare uno sfondo personalizzato si hanno a disposizione due attributi, identici a quelli già citati a proposito del corpo del documento (v. sezione 4):

- **BGCOLOR="colore"**: per definire un colore di sfondo omogeneo; il colore deve essere espresso, come sempre, tramite il suo valore esadecimale (v. sezione 11);
- **BACKGROUND="nome_file"**: per collocare sullo sfondo un'immagine; se necessario, oltre al nome del file occorre specificare il suo indirizzo o percorso (v. sezioni 4 e 13).

Questi due attributi possono essere inseriti in due differenti posizioni:

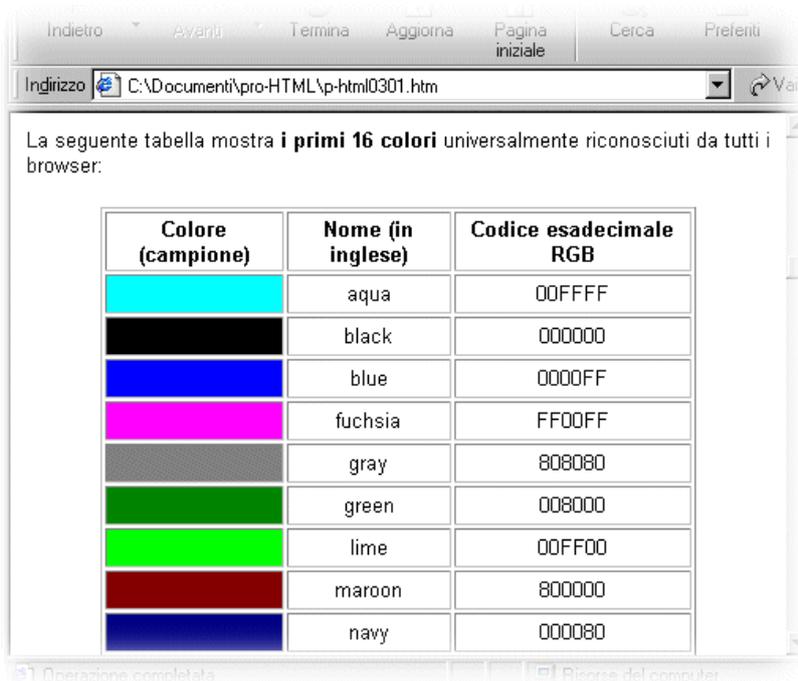
- all'interno del tag di apertura <TABLE>, per assegnare tale caratteristica all'intera tabella;
- oppure all'interno di un tag di apertura <TD>, per assegnare tale caratteristica a una singola cella.

Attenzione, però: sia i due attributi (BGCOLOR e BACKGROUND) che la loro collocazione

(tabella o singola cella), pongono **problemi di compatibilità**.

Solo alcuni browser, in versione recente, sono in grado di interpretare correttamente queste istruzioni.

Un esempio dovrebbe essere apparso nella sezione 11: la tavolozza dei colori è appunto una tabella con sfondi differenziati per le singole celle.



Spaziatura delle celle

Tramite appositi attributi del tag <TABLE> si può regolare a piacere la spaziatura tra le celle e all'interno delle celle.

In mancanza di tali indicazioni, il browser dà per scontato un valore predefinito in entrambi i casi.

- L'attributo **CELLSPACING** permette di definire la **spaziatura tra una cella e quelle adiacenti** (e quindi lo spessore aggiuntivo del bordo, se visibile). La sintassi generica è:

<TABLE CELLSPACING="numero_di_pixel"> ... </TABLE>

Il valore predefinito è 2.

Esempi:

CELLSPACING=2 (predefinito)

Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi.	Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit au gue duis dolore te feugat nulla facilisi.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diem nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.	Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CELLSPACING=5

Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi.	Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisi.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.	Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

- L'attributo **CELLPADDING** permette di definire la spaziatura interna delle celle, ossia la quantità di **spazio vuoto tra il margine della cella e i suoi contenuti** (testi e/o immagini).
La sintassi generica è:

<TABLE CELLPADDING="numero_di_pixe"> ... </TABLE>

Il valore predefinito è 1.
Esempi:

CELLPADDING=1 (predefinito)

Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi.	Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisi.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.	Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CELLPADDING=5

Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi.	Duis autem dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugiat nulla facilisi.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.	Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Estensione delle celle

Gli attributi **HEIGHT** e **WIDTH** permettono di definire l'estensione di ogni cella in altezza e larghezza.

Tali attributi devono essere inseriti nel tag di apertura <TD> di ogni cella:

- **<TD HEIGHT="numero_di_pixel"> ... </TD>**: questa sintassi generica fissa l'**altezza in pixel** della cella.
Si noti che tutte le celle di una stessa riga devono avere la stessa altezza.
Se vengono indicati valori differenti, il browser si basa automaticamente su quello più elevato.
- **<TD WIDTH="numero_di_pixel"> ... </TD>**: questa sintassi generica fissa la **larghezza in pixel** della cella.
Si noti che tutte le celle di una stessa colonna devono avere la stessa larghezza.
Se vengono indicati valori differenti, il browser si basa automaticamente su quello più elevato.

Ovviamente questi attributi possono essere abbinati all'interno di un medesimo tag <TD>. Anziché in pixel, i valori possono anche essere espressi **in percentuale** (rispetto alla larghezza dell'intera tabella).

Esempio: **<TD WIDTH="50%"> ... </TD>**.

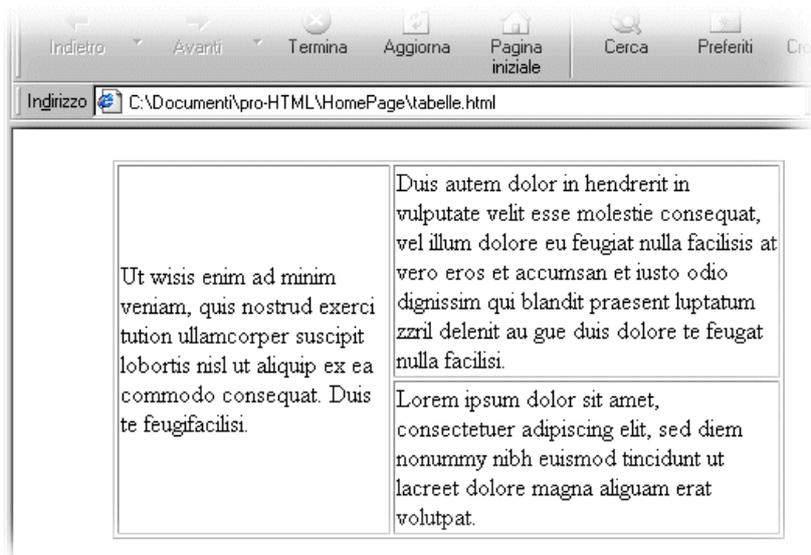
Attenzione al seguente dettaglio: quando si fissano specifici valori di altezza e larghezza delle celle, bisogna assicurarsi che il contenuto di ogni cella non ecceda lo spazio disponibile (altrimenti si incorre in errori di visualizzazione).

Un caso particolare è la possibilità di **spanning**, ossia di estendere una cella per più righe o colonne: si crea in questo modo una tabella con un numero difforme, non costante, di celle per ogni riga/colonna.

Per ottenere questo risultato basta inserire nel tag <TD> desiderato uno dei seguenti attributi:

- **ROWSPAN="numero"** per estendere la cella attraverso il numero di righe specificato; esempio:

```
tabelle.html - Blocco note
File Modifica Cerca ?
<TABLE BORDER=1 WIDTH="80%" ALIGN="center">
<TR>
<TD ROWSPAN="2">Ut wisis enim ad minim veniam, quis
nostrud exerci tution ullamcorper suscipit lobortis
ut aliquip ex ea commodo consequat. Duis te
feugifacilisi.</TD>
<TD>Duis autem dolor in hendrerit in vulputate velit
molestie consequat, vel illum dolore eu feugiat nulla
facilisis at vero eros et accumsan et iusto odio
dignissim qui blandit praesent luptatum zzril delenit
gue dui dolore te feugiat nulla facilisi.</TD>
</TR>
<TR>
<TD>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed diam nonummy nibh euismod tincidunt ut laoreet
dolore magna aliquam erat volutpat.</TD>
</TR>
</TABLE>
```

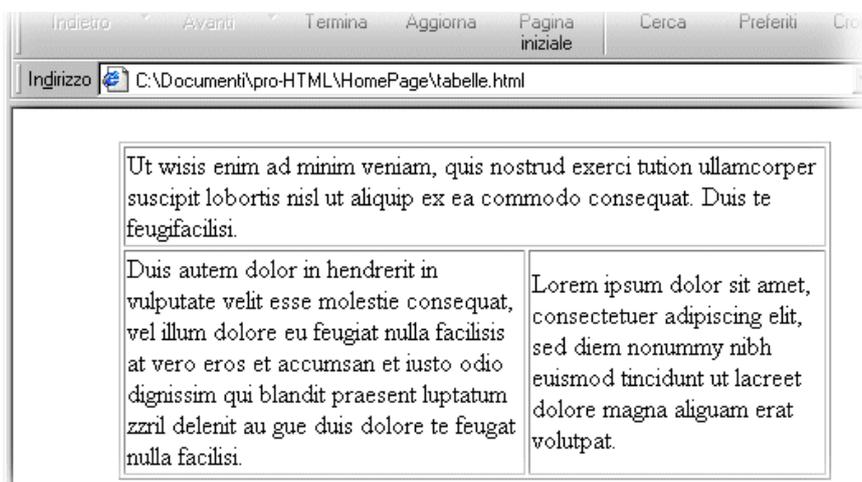


- **COLSPAN="numero"** per estendere la cella attraverso il numero di colonne specificato; esempio:

```

tabelle.html - Blocco note
File Modifica Cerca ?
<TABLE BORDER=1 WIDTH="80%" ALIGN="center">
<TR>
<TD COLSPAN="2">Ut wisis enim ad minim veniam, quis
nostrud exerci tution ullamcorper suscipit lobortis nisl
ut aliquip ex ea commodo consequat. Duis te
feugifacilisi.</TD>
</TR>
<TR>
<TD>Duis autem dolor in hendrerit in vulputate velit e
molestie consequat, vel illum dolore eu feugiat nulla
facilisis at vero eros et accumsan et iusto odio
dignissim qui blandit praesent luptatum zzril delenit
gue duis dolore te feugiat nulla facilisi.</TD>
<TD>Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed diem nonummy nibh euismod tincidunt ut lacree
dolore magna aliquam erat volutpat.</TD>
</TR>
</TABLE>

```



Naturalmente, se si utilizzano questi attributi, occorre tenere presente che **il numero complessivo di celle sulle varie righe/colonne** deve rispettare lo spanning:

- se la cella si estende su più righe, le righe successive devono contenere una cella in meno;
- se la cella si estende su più colonne, la sua riga deve contenere una cella in meno rispetto alle altre righe, e così via.

Osservando bene le figure precedenti si può capire facilmente questo concetto.

A questo punto si è in grado di impostare una tabella con sufficienti possibilità di personalizzazione.

Restano da esaminare soltanto gli attributi riguardanti il contenuto di ogni cella: ecco l'argomento della prossima sezione.

Sezione 18. Formattazione dei contenuti

Allineamento orizzontale

Per determinare l'allineamento orizzontale dei contenuti di una cella si ricorre all'attributo **ALIGN**:

<TD ALIGN="valore"> ... </TD>

Questo attributo può assumere tre differenti valori:

- **ALIGN="left"** (predefinito): allineamento a sinistra;
- **ALIGN="center"**: allineamento al centro;
- **ALIGN="right"**: allineamento a destra.

Per esempio:

ALIGN="left"	ALIGN="center"	ALIGN="right"
Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis.	Duis autem dolor in hendrerit in vulputate velit esse molestie consequat.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diem nonummy nibh euismod tincidunt.

Allineamento verticale

Per determinare l'allineamento verticale dei contenuti di una cella si ricorre all'attributo **VALIGN**:

<TD VALIGN="valore"> ... </TD>

Questo attributo può assumere quattro differenti valori:

- **VALIGN="top"**: allineamento al margine superiore della cella;
- **VALIGN="middle"** (predefinito): allineamento al centro dell'altezza della cella;
- **VALIGN="bottom"**: allineamento al margine inferiore della cella;
- **VALIGN="baseline"**: il contenuto della cella si allinea alla base della prima riga di testo delle celle adiacenti.

Per esempio:

VALIGN="top"	VALIGN="middle"	VALIGN="bottom"
Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis.	Duis autem dolor in hendrerit in vulputate velit esse molestie consequat.	Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diem nonummy nibh euismod tincidunt.

Scorrimento a capo del testo

Normalmente il testo contenuto in una cella scorre a capo automaticamente (wrap) a seconda delle dimensioni della cella stessa, della tabella e dell'intera finestra del browser.

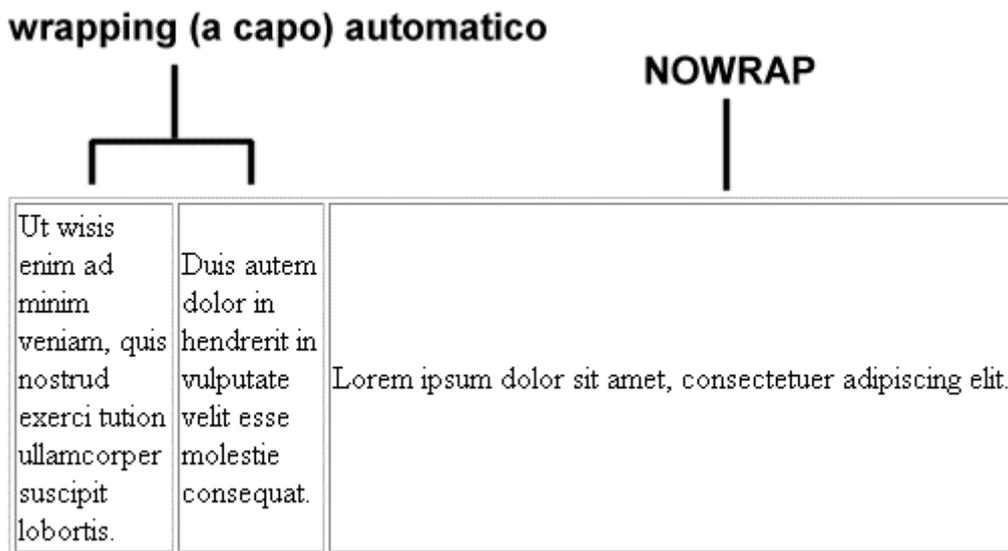
Tuttavia è possibile forzare il testo di una cella affinché non vada mai a capo.

Questa soluzione può essere utile in quelle rare occasioni in cui conviene mantenere fissa la disposizione delle scritte.

A tale scopo è sufficiente aggiungere nel tag <TD> l'attributo **NOWRAP**:

<TD NOWRAP> ... </TD>

Per esempio:



Attenzione

Le dimensioni delle celle adiacenti variano automaticamente in base all'allargamento della cella con l'attributo **NOWRAP** (nell'esempio qui sopra, le due celle a sinistra diventano alte e molto strette).

Formattazione generica

Per qualsiasi altra esigenza, ai contenuti di una cella si possono applicare senza restrizioni **tutti i comuni tag di formattazione** esaminati nelle prime sezioni del modulo: grassetto, corsivo, tipo e dimensione dei caratteri e così via.

L'unica avvertenza importante è la seguente: ricordarsi di aprire e chiudere i tag di formattazione all'interno della singola cella.

In altre parole, occorre che qualsiasi formattazione venga inclusa all'interno della coppia di tag **<TD> ... </TD>** della cella.

Se in un'altra cella si desidera applicare un'analogha formattazione, occorre inserire nuovamente i tag corrispondenti.

Esercitazione

Per sperimentare le competenze maturate in questa sezione della lezione, ecco un altro esercizio in stile "copia dal vero".

Si supponga di partire da questo documento in formato stampato e di doverlo realizzare in formato HTML per la pubblicazione on-line, si cerchi di costruire una pagina Web il più possibile simile al modello che segue, utilizzando in particolare i tag relativi alle tabelle.

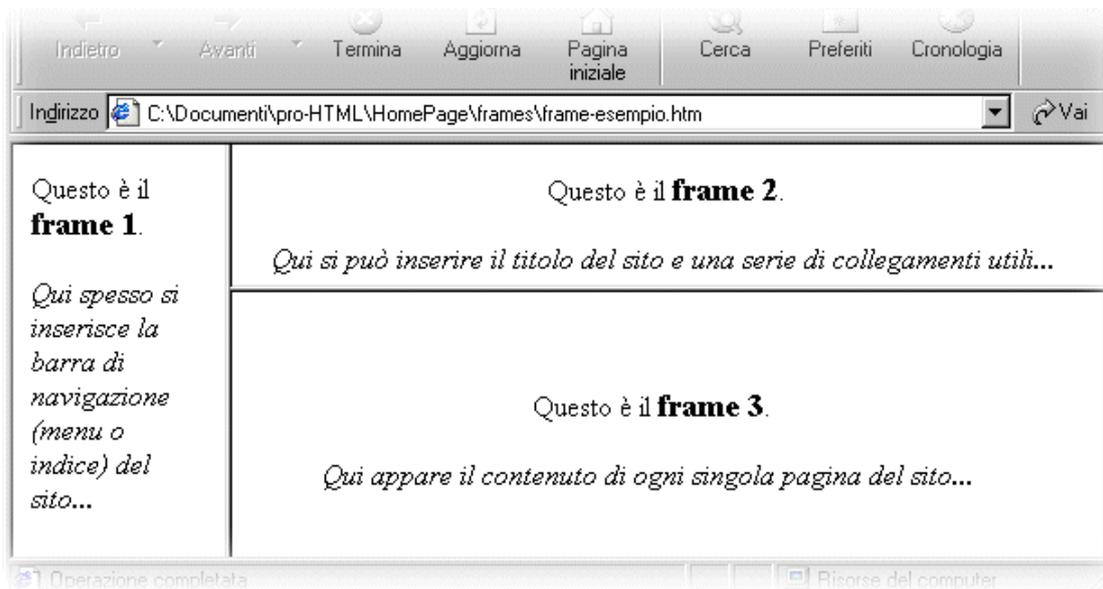
<h1>I motori di ricerca</h1>	
<p><i>Non basta saper navigare sul Web. Per utilizzare Internet in modo intelligente e produttivo è necessario conoscere i motori di ricerca e le strategie per interrogarli efficacemente.</i></p> <p style="text-align: right;"><i>"Una vita senza ricerca non merita di essere vissuta" (Platone)</i></p>	
Tipi di motori	Modalità di ricerca
<p>Esistono vari tipi di motori di ricerca:</p> <ul style="list-style-type: none">• motori "a catalogo" o directory, come Yahoo;• motori per la ricerca libera di parole-chiave, come Altavista;• meta-motori per effettuare ricerche incrociate, come MetaCrawler;• banche-dati specializzate in particolari settori. <p>Esistono poi motori di portata internazionale e motori specifici per la ricerca tra i siti italiani o di altre nazionalità.</p>	<p>Ad esempio, per effettuare una ricerca libera occorre:</p> <ol style="list-style-type: none">1. collegarsi al sito Web di un apposito motore;2. inserire una o più parole-chiave, combinate mediante operatori logici come AND oppure NOT;3. fare clic sul pulsante che avvia la ricerca... <p>Il risultato sarà un elenco di siti Web in cui sono presenti le parole-chiave indicate. Visitando ogni sito si potrà verificare se corrisponde alle proprie esigenze.</p>

Sezione 19. I frame: vantaggi e svantaggi

Cosa sono i frame

I frame (letteralmente: "cornici") sono un tipo particolare di struttura HTML, che consente di suddividere la finestra del browser in vari **riquadri fisicamente distinti**.

Per esempio:

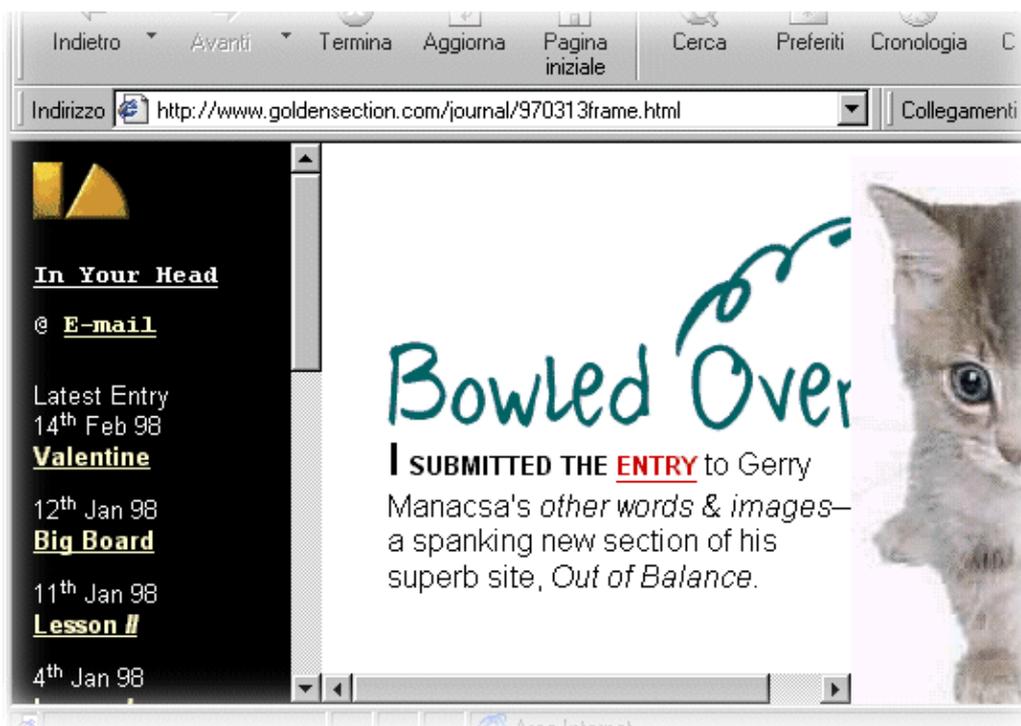


I frame possono essere individuati a colpo d'occhio, se sono circondati da un **bordo** visibile (come nella figura precedente).

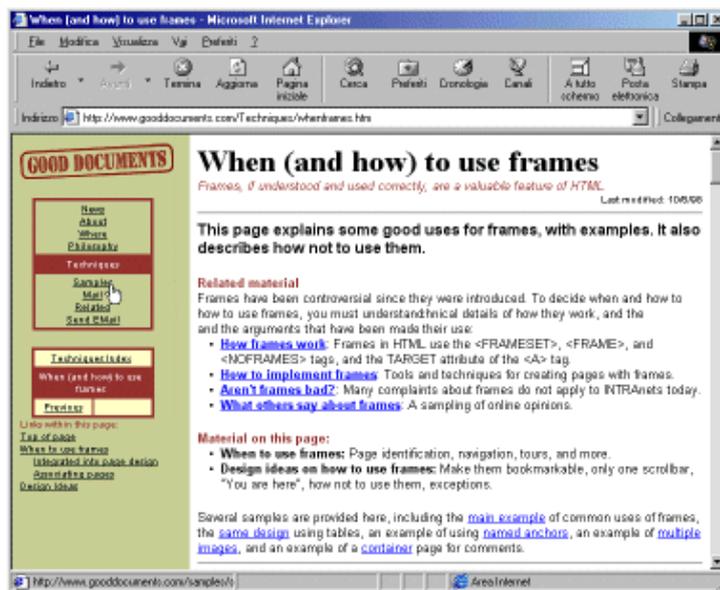
Tuttavia, a volte, il bordo è invisibile: dipende dalle scelte grafiche del progettista Web.

In questi casi, un altro criterio che consente di evidenziare i frame è la presenza di una **barra laterale di scorrimento**.

Per esempio:



La barra di scorrimento appare soltanto se il contenuto del frame fuoriesce dallo spazio disponibile: in certi casi, solo ridimensionando la finestra del browser (e dunque riducendo lo spazio visibile) ci si può accorgere della presenza dei frame. Per esempio:



A che scopo si utilizzano i frame?

- La principale caratteristica dei frame è che possono essere **aggiornati in modo indipendente**: il contenuto di un frame può restare fisso sullo schermo, mentre il contenuto degli altri frame varia a seconda dei comandi impartiti dal visitatore.
- L'applicazione più tipica e frequente, perciò, consiste nel creare un frame (di solito in alto o a sinistra) contenente il menu del sito Web, chiamato anche **barra di navigazione**. Questa parte della finestra resta fissa (fintanto che il visitatore naviga all'interno del sito).
- Ogni volta che il visitatore clicca su una voce del menu, nel frame a fianco appare il **contenuto** della pagina corrispondente.

Come funzionano

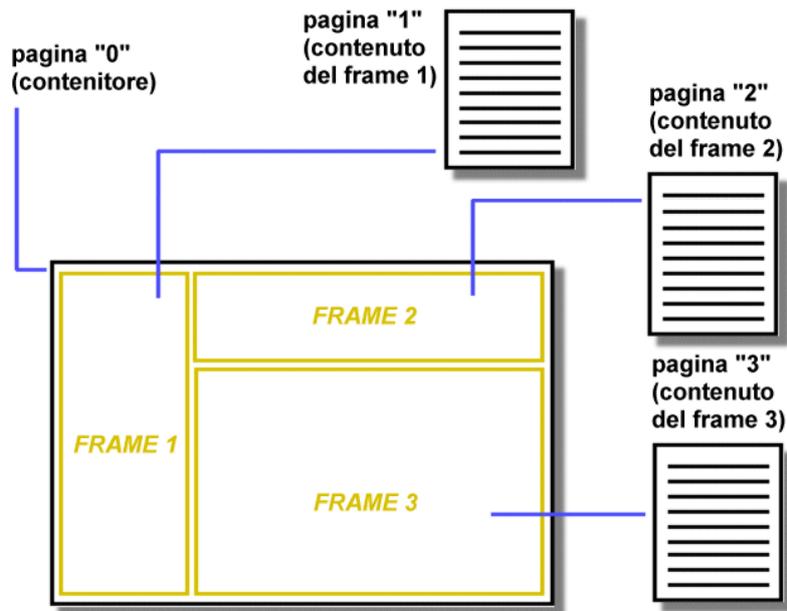
I frame si basano su una sintassi HTML diversa da quella delle normali pagine Web (che abbiamo considerato fino a qui).

Nel caso di una struttura a frame, ciò che appare nella finestra del browser **non è una singola pagina** HTML.

Un insieme di frame viene generato creando, con opportuni tag HTML (v. prossima sezione):

- **una pagina "contenitore"**: questa pagina non contiene alcunché (né testo né immagini, salvo eccezioni), ma indica al browser soltanto la presenza e la disposizione dei vari frame;
- **altre pagine**, ciascuna delle quali corrisponde al contenuto da visualizzare in ogni frame.

Ecco un esempio schematico:



Vantaggi

I frame offrono alcuni innegabili vantaggi:

- un sito costruito con opportune combinazioni di frame si può presentare in modo **graficamente molto elegante** e persino sofisticato;
- una corretta impostazione e suddivisione dei frame può aiutare il navigatore a comprendere istantaneamente la **struttura logica** del sito e delle sue varie sezioni (argomenti e sottoargomenti);
- la navigazione può risultare molto **rapida**, e dunque comoda per il visitatore, in quanto solo una parte dello schermo viene ricaricata o aggiornata (mentre uno o più frame restano fissi).

Svantaggi

D'altra parte, l'utilizzo dei frame comporta numerosi svantaggi, alcuni dei quali significativi:

- **accessibilità/compatibilità**: i frame sono un'innovazione abbastanza recente, quindi le **vecchie versioni dei browser** potrebbero non essere in grado di visualizzarli; si rischia di escludere una parte dell'audience potenziale del sito;
- **facilità d'uso**: i frame hanno un comportamento "anomalo" rispetto alle normali pagine Web, almeno agli occhi di un navigatore non esperto; è probabile che insorgano difficoltà se l'utente vuole **salvare, stampare o memorizzare in un segnalibro** il contenuto di uno specifico frame;
- **motori di ricerca**: alcuni motori di ricerca non riescono a catalogare efficacemente i siti basati su frame; si rischia perciò di diminuire la popolarità e la reperibilità delle proprie pagine;
- **sicurezza**: almeno fino a poco tempo fa, le pagine con frame erano particolarmente vulnerabili agli attacchi degli hacker: un hacker poteva facilmente sostituire il contenuto di un frame con una pagina fittizia, costruita ad arte.

Frame e "usabilità" del Web

Come dichiarato nella sezione, i frame pongono non poche difficoltà per i navigatori non esperti o dotati di browser non aggiornati.

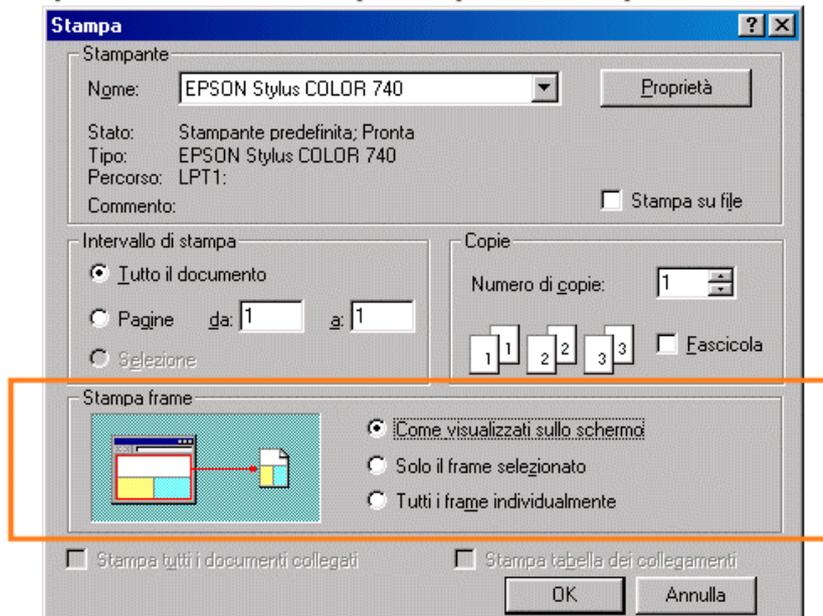
Il fatto è che l'utente medio probabilmente ignora questi due aspetti:

- una schermata suddivisa in più frame **non è una singola pagina**;
- può essere attivo solo un frame per volta; per **frame "attivo"** si intende quello su cui l'utente ha cliccato per ultimo (con ogni probabilità, perciò, si tratta del frame contenente l'eventuale menu/barra di navigazione).

Di conseguenza, in molte circostanze il comportamento della (presunta) pagina Web può sorprendere e disorientare l'ignaro visitatore. Si considerino le seguenti operazioni.

- **Salvataggio e segnalibri:** quando si cerca di salvare sul proprio computer una copia della (presunta) pagina Web, o quando si cerca di memorizzarne l'indirizzo nel proprio elenco di siti preferiti, si rischia di ottenere un riferimento inutile, in quanto parziale o inesatto.
- **Stampa:** le versioni più recenti di Internet Explorer, dalla 4.0 in su, consentono di scegliere se stampare solo il frame attivo o tutti i frame così come appaiono sullo schermo (tale opzione va comunque ricercata nel menu *File/Stampa*; non è invece disponibile se l'utente clicca direttamente sul pulsante *Stampa*). Le vecchie versioni di Internet Explorer e tutte le versioni di Netscape fino alla 4.5 non prevedono tale opzione.

Opzioni di Internet Explorer per la stampa dei frame:



Consiglio

Per questi motivi, oltre che per l'incompatibilità dei frame con i browser più datati, quasi tutti i professionisti della progettazione Web tendono a sconsigliare l'utilizzo dei frame.

A titolo di esempio, si possono trovare molte informazioni e riferimenti utili nell'ottimo sito **Usable Web**: <<http://www.usableweb.com>>.

Frame e tabelle a confronto

Dal punto di vista grafico, una certa disposizione (impaginazione) di testo e immagini sullo schermo può essere ottenuta sia con un insieme di frame sia con una tabella (v. sezioni da 15 a 18).

La differenza fondamentale è che:

- utilizzando i **frame**, una parte dello schermo può davvero rimanere fissa e immutabile (a tutto vantaggio dei tempi di realizzazione e di quelli di caricamento/navigazione);
- utilizzando le **tabelle**, invece, ogni contenuto visibile (compresa l'ipotetica barra di navigazione) deve essere inserito in ogni singola pagina Web e ricaricato ogni volta durante la navigazione: i tempi si allungano, almeno teoricamente.

In realtà, l'allungamento dei tempi di caricamento legato all'uso delle tabelle può risultare quasi inavvertibile.

Una volta caricata la prima pagina, infatti, i contenuti (comprese le immagini e quindi anche i pulsanti della barra di navigazione) vengono memorizzati nella cache del browser del visitatore.

L'accesso alle pagine successive, in cui tali elementi si ripetono identici, può essere perciò molto rapido.

Considerando poi gli svantaggi tipici dei frame (sopra accennati), si può concludere con una leggera preferenza a favore delle tabelle.

Consiglio

Si raccomanda di ricorrere ai frame solamente in questi casi:

- quando sono assolutamente necessari per **esigenze grafiche o tecniche**;
- quando si ha la garanzia che gli utenti dispongono degli strumenti e delle conoscenze necessarie per gestirli correttamente (per esempio in una Intranet aziendale, ossia in un **ambiente omogeneo** in cui si può supporre che tutti gli utenti abbiano lo stesso browser e lo stesso grado di competenza).

Sezione 20. Impostazione di una struttura a frame

La pagina "contenitore"

Innanzitutto occorre creare la pagina "contenitore", quella che imposta la combinazione dei frame.

Questa pagina differisce dalla consueta sintassi HTML: al posto del tag <BODY> si utilizza infatti il tag <FRAMESET>.

Il codice di questa pagina sarà dunque così configurato:

```
<HTML>
.
.. <HEAD>
... (...) contenuto dell'intestazione (...)
.. </HEAD>
.
.. <FRAMESET attributi>
... (...) altri tag per definire le caratteristiche dei singoli frame (...)
.. </FRAMESET>
.
</HTML>
```

Mediante un'opportuna ripetizione del tag <FRAMESET> e l'aggiunta di alcuni attributi è possibile configurare a piacere il numero e la disposizione dei frame sullo schermo.

Numero, posizione e dimensioni dei frame

In primo luogo è **obbligatorio** specificare se i frame devono essere disposti in orizzontale (righe) oppure in verticale (colonne).

A questo scopo, si deve aggiungere al tag <FRAMESET> di apertura l'attributo **ROWS** (righe) oppure **COLS** (colonne).

Inoltre, associato a tale attributo si deve specificare un valore (o meglio, un gruppo di valori) che definisca le dimensioni di ogni frame.

La sintassi generica è:

```
<FRAMESET COLS="larghezza_colonna_1, larghezza_colonna_2">
```

oppure:

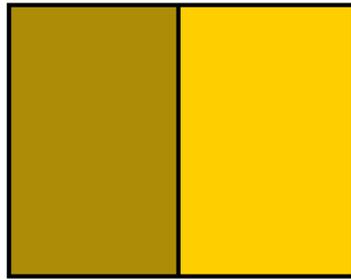
```
<FRAMESET ROWS="altezza_riga_1, altezza_riga_2">
```

Il **numero** di valori indicati determina automaticamente il numero di righe (o colonne), cioè di frame: per costruire tre frame verticali basta specificare l'attributo COLS accompagnato da tre valori.

Le **dimensioni** dei frame possono essere indicate:

- in **percentuale** (rispetto alle dimensioni della finestra); per esempio: **COLS="20%,80%"** oppure **ROWS="10%,90%"**;
- in **pixel** (valore assoluto); per esempio: **COLS="100,400"** oppure **ROWS="50,600"**;
- in senso **relativo**, con il simbolo * (**asterisco**); per esempio, l'attributo **COLS="30%,*"** produce due frame di cui il primo occupa il 30% della larghezza e l'altro tutta la parte restante della finestra; l'attributo **COLS="2*,3*,*"** produce tre frame di cui il primo occupa i 2/6 della finestra, il secondo i 3/6 (metà) e il terzo la larghezza rimanente.

Ricapitolando, ecco alcuni esempi schematici di strutture a frame:



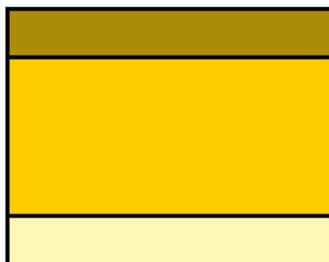
```
<FRAMESET COLS="50%,50%">  
<FRAME SRC="sinistro.html">  
<FRAME SRC="destro.html">  
</FRAMESET>
```



```
<FRAMESET COLS="25%,*">  
<FRAME SRC="sinistro.html">  
<FRAME SRC="destro.html">  
</FRAMESET>
```



```
<FRAMESET ROWS="50%,*">  
<FRAME SRC="sopra.html">  
<FRAME SRC="sotto.html">  
</FRAMESET>
```



```
<FRAMESET ROWS="20%,60%,20%">  
<FRAME SRC="sopra.html">  
<FRAME SRC="centro.html">  
<FRAME SRC="sotto.html">  
</FRAMESET>
```

Contenuto dei frame

Come si è appena visto, per impostare un insieme di frame bisogna innanzitutto specificare l'attributo COLS o ROWS del tag <FRAMESET>.

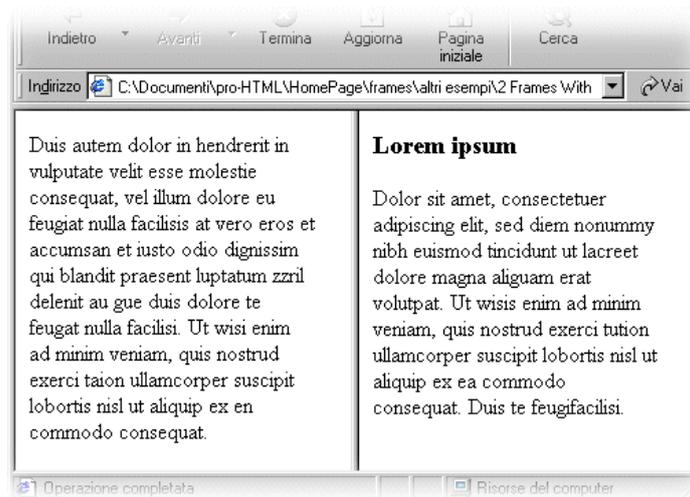
Ma c'è un altro elemento che si deve **necessariamente** inserire, affinché la pagina funzioni. Si devono infatti specificare, tra <FRAMESET> e </FRAMESET>, i documenti che riempiranno i singoli frame.

Per definire il contenuto dei frame si utilizza il tag <FRAME SRC="nome_file.html">.

Occorre quindi:

1. preparare a parte le singole pagine che andranno collocate in ciascun frame;
2. salvare queste pagine come altrettanti file HTML;
3. nella pagina "contenitore", predisporre la struttura a frame e specificare per ogni frame la rispettiva pagina di contenuto.

Per esempio, si supponga di voler ottenere una struttura di questo tipo:



Bisognerà allora:

1. preparare un file HTML con il testo da inserire nel frame sinistro, e salvarlo con un nome come **pagina_sinistra.html**;
2. preparare un file HTML con il testo da inserire nel frame destro, e salvarlo con un nome come **pagina_destra.html**;
3. preparare la pagina "contenitore" e salvarla con qualsiasi nome (ricordando che, se si tratta della pagina iniziale del sito, è opportuno chiamarla **index.html** o **home.html**);
4. in questa pagina "contenitore" inserire il seguente codice HTML:

```
<FRAMESET COLS="50%,50%">
<FRAME SRC="pagina_sinistra.html">
<FRAME SRC="pagina_destra.html">
</FRAMESET>
```

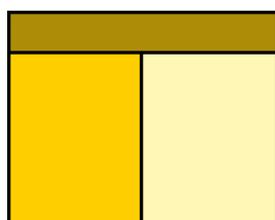
A questo punto, visualizzando nel browser la pagina "contenitore", si dovrebbe ottenere un risultato pressoché identico a quello della figura precedente.

Frame annidati

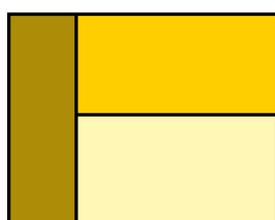
È anche possibile creare strutture con un numero irregolare di frame: per esempio, un frame orizzontale superiore e due inferiori affiancati, oppure un frame verticale sinistro e due frame sovrapposti a destra.

A questo scopo, è sufficiente impostare **vari blocchi di tag <FRAMESET> ... </FRAMESET> uno dentro l'altro**.

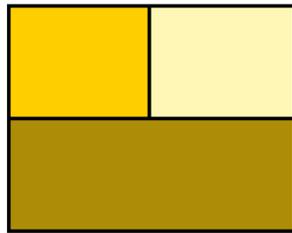
Ecco alcuni esempi schematici:



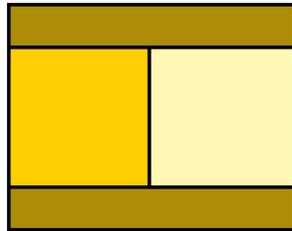
```
<FRAMESET ROWS="20%,80%">
<FRAME SRC="sopra.html">
<FRAMESET COLS="50%,50%">
<FRAME SRC="sotto_sinistra.html">
<FRAME SRC="sotto_destra.html">
</FRAMESET>
</FRAMESET>
```



```
<FRAMESET COLS="25%,75%">
<FRAME SRC="sinistro.html">
<FRAMESET ROWS="50%,50%">
<FRAME SRC="destro_sopra.html">
<FRAME SRC="destro_sotto.html">
</FRAMESET>
</FRAMESET>
```



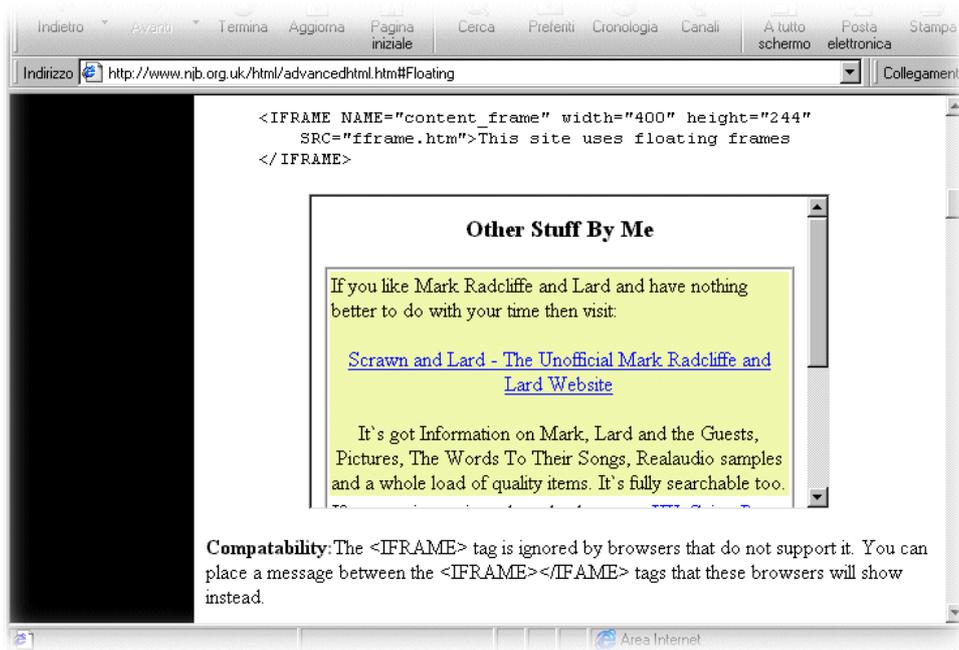
```
<FRAMESET ROWS="50%,50%">
  <FRAMESET COLS="50%,50%">
    <FRAME SRC="sopra_sinistro.html">
    <FRAME SRC="sopra_destro.html">
  </FRAMESET>
  <FRAME SRC="sotto.html">
</FRAMESET>
```



```
<FRAMESET ROWS="20%,60%,20%">
  <FRAME SRC="sopra.html">
  <FRAMESET COLS="50%,50%">
    <FRAME SRC="centro_sinistro.html">
    <FRAME SRC="centro_destro.html">
  </FRAMESET>
  <FRAME SRC="sotto.html">
</FRAMESET>
```

Frame interni (flottanti)

Un caso particolare è rappresentato dai cosiddetti frame interni (o flottanti). Si tratta di frame che appaiono "immersi" all'interno di una normalissima pagina HTML, come riquadri a sé stanti. Per esempio:



I frame interni non richiedono l'impostazione di una pagina "contenitore": possono essere aggiunti a una comune pagina Web al pari di testo, immagini, tabelle eccetera. Per realizzare questi frame, infatti, basta inserire nel codice di una comune pagina HTML, all'interno della sezione <BODY>, questa coppia di tag:

```
<IFRAME>
.. (...) contenuto del frame (...)
</IFRAME>
```

Attenzione!

Il tag <IFRAME> non è nato nell'ambito dello standard HTML: è stato introdotto a suo tempo

da Microsoft, con **compatibilità limitata** a Internet Explorer. Di conseguenza, il suo utilizzo è sconsigliabile.

IL TAG **<NOFRAMES>**

Come si è sottolineato nella sezione precedente, i frame possono incorrere in problemi di compatibilità.

Non tutti i browser sono in grado di riconoscerli e visualizzarli.

Se un utente si collega a un sito con frame mediante un browser non compatibile, non riuscirà a visualizzare alcun contenuto.

Per evitare questo rischio, la soluzione più semplice è aggiungere, alla fine della pagina "contenitore", il tag **<NOFRAMES>**.

Questo tag permette di specificare un **contenuto alternativo** (testo e/o immagini) che verrà visualizzato dai browser non compatibili con i frame.

Per esempio:

<FRAMESET attributi

(...) struttura a frame (...)

</FRAMESET>

<NOFRAMES>

. . Se leggete questo messaggio,

. . il vostro browser non è

. . compatibile con i frame.

. . Fate clic qui:

. . **** versione senza frame ****

</NOFRAMES>

Con questo accorgimento, il sito è ben impostato: la struttura a frame apparirà ai visitatori opportunamente attrezzati, mentre gli altri troveranno almeno un esplicito avvertimento (e, se possibile, una soluzione alternativa per accedere ugualmente ai contenuti del sito).

Sezione 21. Personalizzazione dei frame

Spaziatura

È possibile regolare la spaziatura tra i frame e quella all'interno di ogni frame (tra il bordo e i contenuti), analogamente a quanto si è visto per le celle di una tabella.

- **Spaziatura tra i frame:** si può impostare, nel tag di apertura `<FRAMESET>`, l'attributo **- FRAMESPACING="numero"**.

In questo modo si definisce lo spazio vuoto (in pixel) tra un frame e l'altro.

Attenzione!

L'attributo `FRAMESPACING` non è riconosciuto da alcuni browser (specie se in vecchie versioni).

- **Spaziatura all'interno di un frame:** per regolare lo spazio vuoto intorno ai contenuti di un frame si hanno a disposizione due attributi del tag `<FRAME>`:
 - **MARGINHEIGHT="numero"**, che permette di definire la quantità di spazio (in pixel) da lasciare vuoto **sopra e sotto** i contenuti del frame;
 - **MARGINWIDTH="numero"**, che permette di definire la quantità di spazio (in pixel) da lasciare vuoto **a destra e a sinistra** dei contenuti del frame.

Bordo

Si può scegliere se visualizzare o meno il bordo dei frame, utilizzando l'attributo

FRAMEBORDER.

Questo attributo deve essere inserito nel tag di apertura `<FRAMESET>` (quello di livello gerarchico più elevato, ossia il più "esterno", se vi sono frame annidati: v. sezione precedente) e può assumere il valore "yes" oppure "no":

- **<FRAMESET FRAMEBORDER="yes"> ... </FRAMESET>**: i frame appaiono circondati da un sottile bordo tridimensionale (ombreggiato);
- **<FRAMESET FRAMEBORDER="no"> ... </FRAMESET>**: i frame appaiono privi di bordo.

Vi sono anche altri due attributi relativi ai bordi:

- **BORDER="numero"**: questo attributo consente di regolare a piacere lo spessore del bordo (in pixel);
- **BORDERCOLOR="#colore"**: questo attributo consente di definire il colore del bordo con un valore esadecimale (v. sezione 11).

Attenzione!

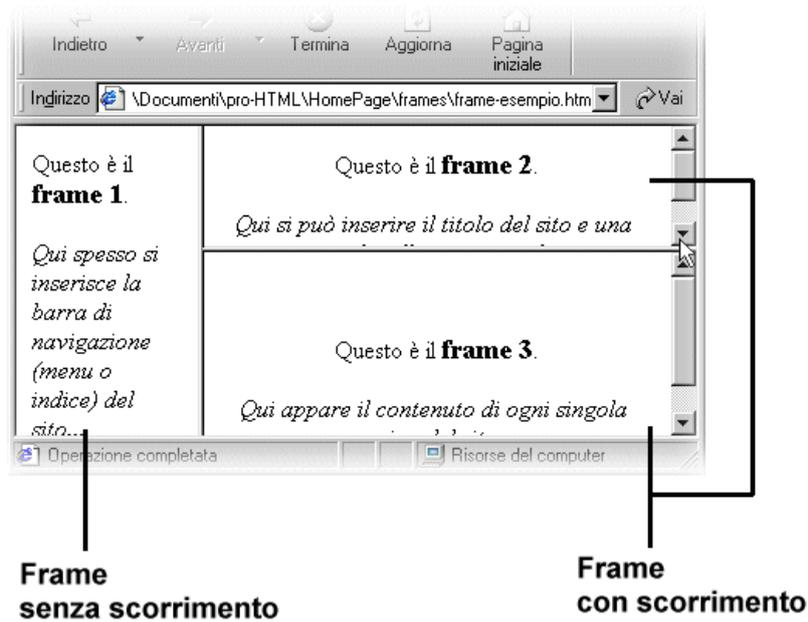
Questi due attributi non sono standard e non tutti i browser sono in grado di riconoscerli: se ne sconsiglia perciò l'utilizzo.

Ridimensionamento e scorrimento (scrolling)

- In fase di visualizzazione, di solito il lettore può **ridimensionare** a piacere un frame, trascinandone il bordo con il mouse.
- Se si desidera impedire che il frame possa essere ridimensionato liberamente, basta inserire nel tag `<FRAME>` l'attributo **NORESIZ**. In questo modo il frame specificato manterrà sempre costanti le sue dimensioni, indipendentemente dalle operazioni dell'utente e dal ridimensionamento della finestra del browser.
- Normalmente, se il contenuto di un frame fuoriesce dallo spazio visibile (rispetto alle dimensioni della finestra del browser), appare automaticamente una **barra di scorrimento**

che permette di effettuare lo scrolling del frame.

In caso di esigenze particolari, è possibile impedire a priori lo scrolling. Basta inserire nel tag <FRAME> l'attributo **SCROLLING="no"**.



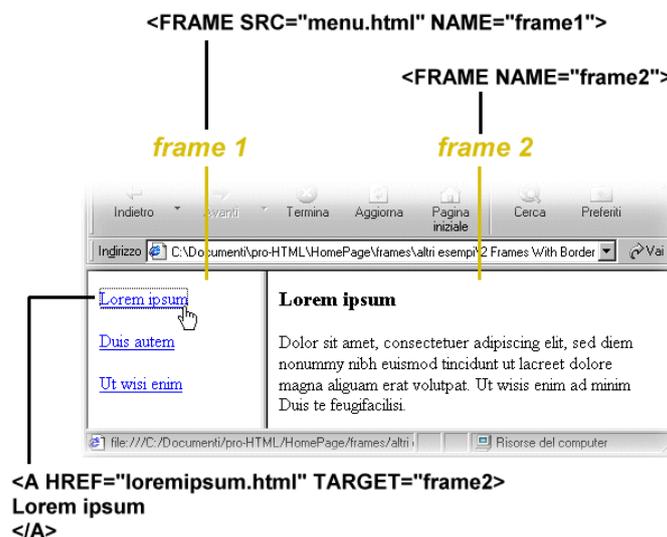
Collegamenti e target

Come si è detto, una delle applicazioni più tipiche dei frame è la creazione di un menu o barra di navigazione in un frame che resta fisso sullo schermo, mentre in un altro frame vengono visualizzati di volta in volta i contenuti selezionati.

Per ottenere questo risultato, occorre:

1. identificare ogni frame con un nome convenzionale mediante l'attributo **NAME="nome"**, da inserire nel tag <FRAME> corrispondente;
2. quando si imposta il collegamento ipertestuale, aggiungere al tag <A HREF> l'attributo **TARGET="nome"**.

Per esempio:



In questo esempio, quando si fa clic su uno dei collegamenti ipertestuali (a sinistra) il frame del menu resta fisso mentre la pagina corrispondente viene caricata nel frame a fianco. A questo punto si sono apprese tutte le nozioni essenziali per gestire con efficacia i frame.

Lezione 5 - Frame

Esercitazione

Per sperimentare le competenze maturate in questo modulo del modulo, ecco un esercizio di riepilogo relativo ai frame.

1. Creare una prima pagina Web (da chiamare **index.html**) del tipo <FRAMESET>: una **struttura con due frame verticali**.

Larghezza del frame sinistro: 30%.

Larghezza del frame destro: 70%.

2. Creare altre tre pagine Web, con i contenuti per ciascun frame:

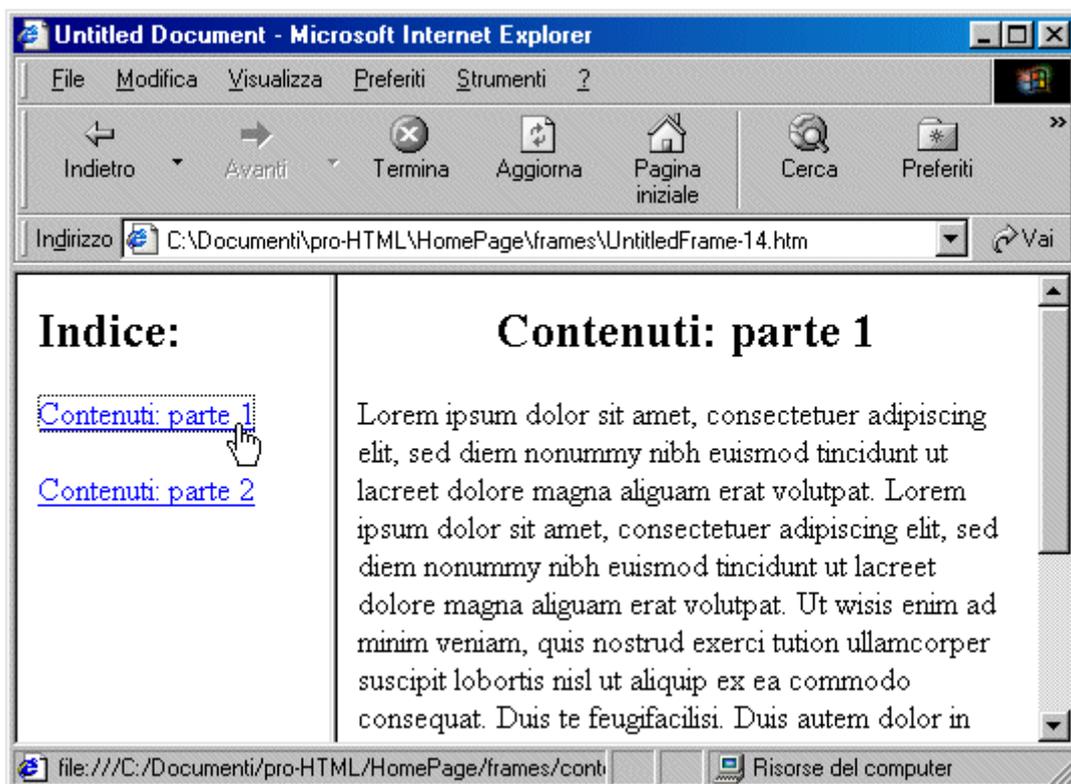
- una pagina chiamata **menu.html**: questa pagina dovrà occupare in modo fisso il frame di sinistra;
- due pagine chiamate rispettivamente **contenuti1.html** e **contenuti2.html**: queste pagine dovranno simulare due sezioni di contenuti del sito (si può inserire un testo a caso oppure, a scelta, inventare un testo sensato).

3. Nella pagina menu.html, inserire **collegamenti ipertestuali** verso le due pagine di contenuti.

4. Impostare i frame e i collegamenti in modo tale che:

- nel frame di sinistra resti sempre visibile il menu;
- facendo clic su un collegamento del menu la corrispondente pagina di contenuti venga caricata nel frame di destra.

Il risultato dovrebbe essere simile a questo:





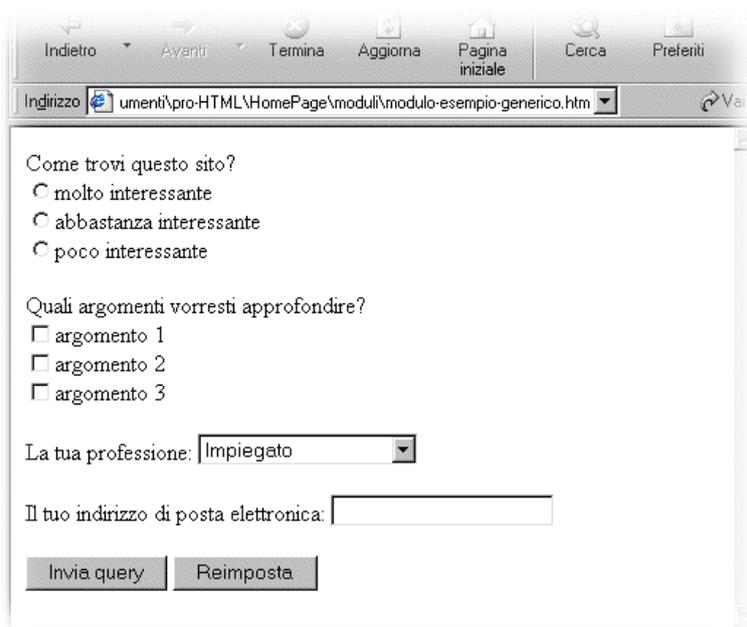
Si cerchi di costruire una struttura Web il più possibile simile a tale modello. Sarebbe preferibile (ma non obbligatorio) **personalizzare** al massimo il risultato, sfruttando tutte le opzioni disponibili descritte in questo modulo e nei precedenti: sfondi, formattazione del testo e così via.

Sezione 22. I moduli interattivi (form)

Principi di funzionamento

Un modulo interattivo è un insieme di testi e di funzioni che consente ai visitatori di comunicare, al gestore del sito, dati di varia natura: risposte a questionari, dati necessari per stipulare un contratto, richieste di informazioni e via dicendo.

Per esempio:



The screenshot shows a web browser window with a form. The browser's address bar displays the URL: `ument\pro-HTML\HomePage\moduli\modulo-esempio-generico.htm`. The form contains the following elements:

- Question: "Come trovi questo sito?" with three radio button options: "molto interessante", "abbastanza interessante", and "poco interessante".
- Question: "Quali argomenti vorresti approfondire?" with three checkbox options: "argomento 1", "argomento 2", and "argomento 3".
- Text: "La tua professione:" followed by a dropdown menu showing "Impiegato".
- Text: "Il tuo indirizzo di posta elettronica:" followed by a text input field.
- Buttons: "Invia query" and "Reimposta".

Un modulo può essere inserito in una normale pagina HTML grazie alla coppia di tag **<FORM>** ... **</FORM>**.

All'interno di questi tag, mediante altri tag specifici, si possono impostare i seguenti elementi:

- **testo** del modulo (domande, commenti, istruzioni per la compilazione);
- **elementi di input** (caselle, pulsanti, campi di testo);
- **pulsanti per l'elaborazione finale** del modulo: un pulsante per la conferma e l'inizio dei dati; un pulsante per la cancellazione e l'eventuale reimpostazione dei dati.

Ecco i principali elementi di un modulo interattivo:



The diagram illustrates typical form elements on a yellow background:

Tipici elementi di Input:

- radio button (opzioni esclusive: "A oppure B"):
- checkbox (opzioni multiple "sì/no"):
- select box (menu/elenco a discesa):
- text box (campo di testo; 1 riga):
- text area (campo di testo; dimensioni a piacere):

Pulsanti di Invio e Reimpostazione:

- submit (conferma e spedizione dati):
- reset (cancellazione e reimpostazione dati):

Attenzione!

L'elaborazione dei dati deve appoggiarsi a funzioni esterne: in altre parole, i dati del modulo non vengono gestiti direttamente dalla pagina HTML.

Nel tag iniziale <FORM>, perciò, occorre specificare la destinazione e il tipo di elaborazione dei dati: "chi" li dovrà gestire e "come".

- Normalmente i dati vengono elaborati da **appositi script o programmi (di tipo CGI)** presenti sul server del sito.
- In tal caso bisogna inserire nel codice HTML del modulo il percorso e il nome della funzione corrispondente.
- In alternativa, è possibile impostare il modulo in modo tale che i dati vengano automaticamente spediti a un **indirizzo di posta elettronica**.

Definizione di un modulo

Per creare un modulo, bisogna inserire nel codice HTML della pagina la coppia di tag **<FORM attributi> ... </FORM>**.

Gli attributi del tag di apertura servono a definire le modalità di gestione ed elaborazione dei dati.

Il tag **<FORM>** ha **due attributi**, che sono **entrambi obbligatori**:

ACTION="destinazione": questo attributo specifica la destinazione dei dati, che può essere uno script/programma CGI residente sul server (per esempio: **ACTION="/cgi-bin/nome_dello_script"**) oppure un indirizzo di posta elettronica, associato al comando mailto: (per esempio: **ACTION="mailto:nome@dominio.com"**);

- **METHOD="tipo_di_elaborazione"**: questo attributo specifica il trattamento da effettuare, che può essere **GET** (i dati vengono elaborati immediatamente e viene fornita in risposta una certa pagina HTML o un altro tipo di output) oppure **POST** (i dati vengono "accumulati" e trasmessi alla destinazione sopra indicata).

In questa sede non si entra nei dettagli specialistici, ma si tenga presente almeno questo aspetto: se la destinazione dei dati è un indirizzo e-mail (anziché uno script del server) è obbligatorio utilizzare il metodo POST.

Per esempio, con la seguente sintassi si può impostare un modulo in modo tale che, quando l'utente clicca sul pulsante di conferma e invio dei dati, questi vengano spediti all'indirizzo e-mail specificato:

```
<FORM ACTION="mailto:fabcom@iol.it" METHOD="post">  
.. (...) elementi di input e testo del modulo (...)  
</FORM>
```

Una volta definita la struttura di base del modulo, è possibile impostarne i contenuti:

1. il testo di accompagnamento, digitato come quello di una comune pagina HTML;
2. gli elementi di input associati al testo, ciascuno dei quali è identificato da una variante del tag **<INPUT>**;
3. i due pulsanti per l'invio o per la reimpostazione del modulo.

Tipi di <INPUT>

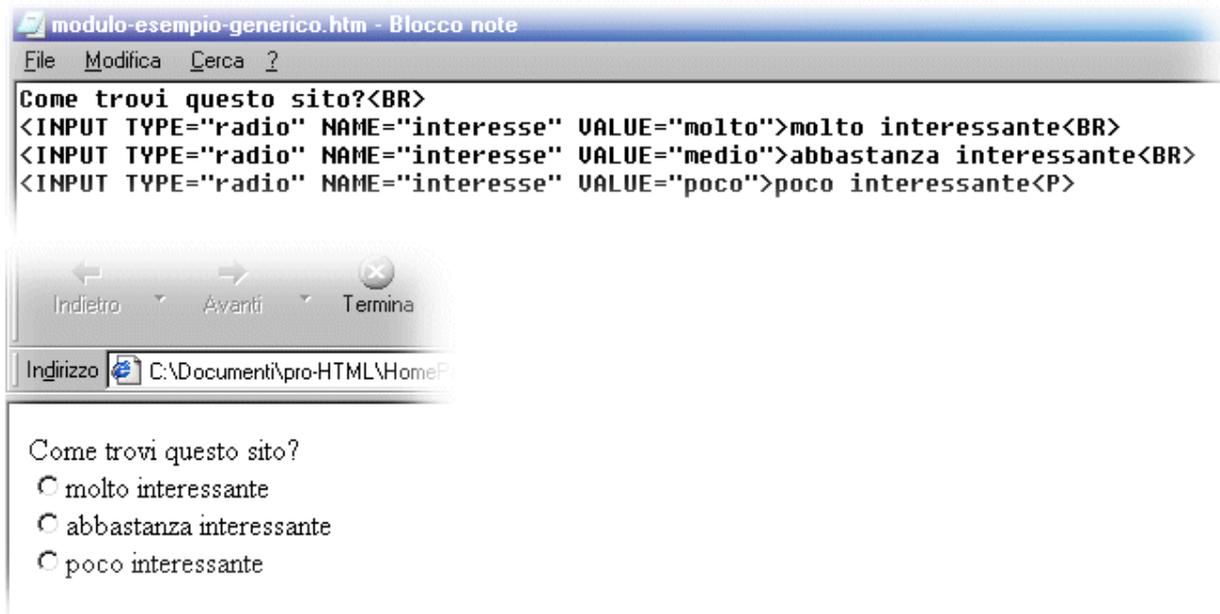
Per impostare gli elementi di input (caselle, pulsanti, campi di testo) si utilizza il tag <INPUT>. Questa è la sintassi generica:

```
<INPUT TYPE="tipo_di_elemento" NAME="nome/gruppo" VALUE="valore">
```

- L'attributo **TYPE** specifica il tipo di elemento: opzione esclusiva del tipo "A oppure B" (**TYPE="radio"**), opzione multipla del tipo "sì/no" (**TYPE="checkbox"**), campo di testo da compilare (**TYPE="text"**).

- L'attributo **NAME** identifica (con un nome a piacere) il singolo elemento o il gruppo di elementi omogenei a cui esso appartiene.
- L'attributo **VALUE** (se previsto) indica il valore dell'elemento.

In particolare, nel caso delle caselle associate alla **risposte a una stessa domanda** è importante identificare tutti questi elementi con lo stesso attributo NAME. Questo esempio dovrebbe aiutare a chiarire il concetto:



Vi sono poi altre varianti del tag `<INPUT>`, che permettono di includere in un modulo **ulteriori elementi interattivi**: immagini con funzioni di pulsante (`<INPUT TYPE="image">`), campi nascosti (`<INPUT TYPE="hidden">`), campi per l'immissione di password (`<INPUT TYPE="password">`) eccetera.

Oltre a questi elementi, si possono inserire campi di testo con dimensioni variabili, mediante il tag `<TEXTAREA>`, ed elenchi di opzioni (menu a discesa) mediante il tag `<SELECT>`. Tali argomenti, tuttavia, esulano dalle finalità di questo modulo

Pulsanti di Invio/Reimpostazione

Per aggiungere in fondo al modulo i due pulsanti di invio o reimpostazione si ricorre ancora al tag `<INPUT>`.

- Pulsante di **Invio** dei dati: `<INPUT TYPE="submit" VALUE="etichetta">` dove al posto di *etichetta* si può scrivere il testo da visualizzare sul pulsante (per esempio: "Spedisci i dati").
- Quando l'utente clicca su questo pulsante, i dati del modulo vengono inviati alla destinazione specificata (ACTION) ed elaborati nel modo previsto (METHOD).
- Pulsante di **Reimpostazione** dei dati: `<INPUT TYPE="reset" VALUE="etichetta">` dove al posto di *etichetta* si può scrivere un testo personalizzato (per esempio: "Cancella e ricomincia").
- Quando l'utente clicca su questo pulsante, i dati precedentemente inseriti vengono azzerati e il modulo può essere ricompilato da capo.

Si noti, comunque, che non è obbligatorio inserire questi due pulsanti alla fine del modulo. Essi possono essere collocati **in qualsiasi posizione**, purché nel codice HTML si trovino compresi all'interno della coppia di tag `<FORM> ... </FORM>` che identificano il modulo.

Sezione 23. Cenni su script ed effetti dinamici

Cosa sono gli script

Gli script sono sequenze di istruzioni scritte in linguaggi specifici (JavaScript o VBScript), ma incorporate direttamente nel codice HTML di una normale pagina Web.

- Il linguaggio **JavaScript** deriva dal linguaggio di programmazione **Java**, inizialmente introdotto da Sun e da Netscape ma ormai universalmente diffuso nell'ambito Web.
- La differenza è che il linguaggio Java serve a realizzare veri e propri programmi (chiamati "applet" se sono installati sui computer degli utenti e "servlet" se installati sui server): questi programmi possono essere richiamati all'interno di una pagina Web mediante appositi tag e collegamenti. Il linguaggio Javascript, invece, permette di incorporare tutti i comandi di uno script direttamente all'interno del codice HTML della pagina.
- Il linguaggio **VBScript** ha una funzione analoga a Javascript, ma è stato sviluppato da Microsoft: deriva dal linguaggio di programmazione **Visual Basic**. Un'altra tecnologia Microsoft che permette di ottenere risultati analoghi è costituita dai cosiddetti **controlli ActiveX** (librerie di oggetti dinamici).

Tutti i browser più recenti (come Internet Explorer e Netscape dalle rispettive versioni 4 in poi) sono in grado di interpretare correttamente queste istruzioni e mostrarne a schermo gli effetti. Possono comunque verificarsi occasionali problemi di compatibilità.

Gli effetti e gli ambiti di applicazione degli script possono essere così riassunti:

- **multimedialità**: gli script possono produrre piccole animazioni, suoni o altri effetti multimediali;
- **interattività**: gli script, per certi aspetti, possono svolgere le stesse funzioni dei moduli interattivi o form (v. sezione precedente).

Come si imposta uno script

Per inserire uno script in una pagina Web si utilizza il tag **<SCRIPT>**.

Questo tag può essere incluso tanto nella sezione <HEAD> quanto nella sezione <BODY> della pagina:

- **se lo script è inserito in <HEAD>**, viene eseguito automaticamente e istantaneamente, prima che si completi il caricamento della pagina nel browser;
- **se lo script è inserito in <BODY>**, la sua esecuzione avviene durante il caricamento, in modo automatico, oppure in risposta a un'azione dell'utente (come il clic su un collegamento o su un pulsante).

La sintassi generica di uno script è:

```
<SCRIPT LANGUAGE="linguaggio">
```

```
.. (...) sequenza di istruzioni secondo la sintassi del linguaggio specifico (...)
```

```
</SCRIPT>
```

Al posto della stringa **linguaggio** bisogna indicare il linguaggio in cui sono effettivamente scritte le istruzioni: **LANGUAGE="javascript"** oppure **LANGUAGE="vbscript"**.

Entrambi i linguaggi hanno una sintassi complessa, sebbene non troppo difficile anche per chi non è un programmatore professionista. Il loro apprendimento richiede comunque uno studio specifico.

Tuttavia, nel prossimo paragrafo si vedrà qualche esempio pratico e si imparerà a copiare e a personalizzare il codice di uno script esistente.

Esempi di script

In questo paragrafo vengono illustrati alcuni esempi in linguaggio **Javascript** (il più diffuso e il più ampiamente compatibile).

Una delle applicazioni più semplici da realizzare con gli script è l'apertura di **finestre di avvertimento** e la visualizzazione di **messaggi** vari.

Esempi di script

Ecco alcuni esempi pratici di script realizzati in linguaggio Javascript.

Script di tipo "Alert"

Il seguente script apre una finestra di tipo "Alert", contenente un messaggio e il pulsante OK. Per vederne l'effetto cliccare su questo collegamento: [esempio di script "Alert"](#).

Ecco il codice di base per uno script di questo tipo:

```
<SCRIPT LANGUAGE="javascript">
window.alert("testo_del_messaggio")
</SCRIPT>
```

Script di tipo "Confirm"

Il seguente script apre una finestra di tipo "Confirm", contenente un messaggio, il pulsante OK e il pulsante Annulla.

Per verificarne l'effetto fare clic su questo collegamento: [esempio di script "Confirm"](#).

Ecco il codice di base per uno script di questo tipo:

```
<SCRIPT LANGUAGE="javascript">
window.confirm("testo_della_domanda")
(...) altre istruzioni per gestire la risposta "OK" e la risposta "Annulla" (...)
</SCRIPT>
```

Script di tipo "Prompt"

Il seguente script apre una finestra di tipo "Prompt", contenente un messaggio, un campo di testo che l'utente può/deve compilare e il pulsante OK.

Per verificarne l'effetto fare clic su questo collegamento: [esempio di script "Prompt"](#).

Ecco il codice di base per uno script di questo tipo:

```
<SCRIPT LANGUAGE="javascript">
window.prompt(testo_del_messaggio, risposta_predefinita)
(...) altre istruzioni per gestire la risposta (...)
</SCRIPT>
```

Nota sulla modalità di attivazione degli script

I codici sopra illustrati provocherebbero in realtà un'attivazione automatica e istantanea dei rispettivi script, a mano a mano che le istruzioni vengono caricate dal browser insieme al corpo della pagina HTML.

Ai fini di questa scheda di approfondimento, invece, si è ritenuto più opportuno rendere **opzionale e volontaria l'attivazione** degli script: ogni script è attivato quando l'utente clicca sul testo che lo richiama.

Come si ottiene questo risultato?

- **Nella sezione <HEAD>** della pagina è stato predisposta un'unica sequenza di istruzioni, comprendente tutti e tre gli script:

```

<SCRIPT LANGUAGE="JavaScript">
<!-- hide script from old browsers

var string1="Ecco un messaggio!"

var string2="Clic su OK se siete d'accordo, altrimenti su Annulla."

var string3="Siete d'accordo."

var string4="Non siete d'accordo."

var string5="Come ti chiami?"

var string6="Tizio"

var string7="OK"

function OpenAlert(string) {
    window.alert(string);
}

function OpenConfirm(string) {
    var agree = window.confirm(string);
    if (agree == true) {
        window.alert(string3);

    }
    else {
        window.alert(string4);
    }
}

function OpenPrompt(question, defaultName) {
    var name = window.prompt(question, defaultName)
    if (name == null) {
        window.alert(string7);
    }
    else {
        window.alert("Piacere di conoscerti, " + name);
    }
}

//-->
</SCRIPT>

```

Nella sezione <BODY> della pagina, a ogni punto di attivazione (per esempio: la scritta **esempio di script "Alert"**) è stato associato un collegamento ipertestuale che richiama la parte di script corrispondente:

```

<a href="javascript:OpenAlert(string1)">
    esempio di script "Alert";
</a>

```

Non bisogna preoccuparsi se l'argomento risulta troppo complesso: lo scopo di questa sezione è solo quello di accennare a livello intuitivo i principi di funzionamento e le potenzialità degli script.

Gli script di esempio appena proposti possono essere facilmente copiati e personalizzati: basta sostituire il testo dei messaggi che appaiono nelle finestre.

Tornando alla sezione, un esercizio appositamente predisposto aiuterà a sperimentare questa operazione in modo facilitato, attraverso la personalizzazione di un semplice script "Alert".

DHTML ed effetti dinamici

Il cosiddetto HTML dinamico (DHTML: "Dynamic HTML") è un'estensione del linguaggio HTML standard, che consente effetti dinamici applicati al contenuto di una pagina Web.

Gli effetti dinamici sono associati a eventi e/o oggetti specifici.
Per esempio:

- **onMouseOver="effetto"**: quando il mouse passa sopra un elemento della pagina (testo o immagine), questo può cambiare colore o aspetto;
- **onClick="effetto"**: quando il visitatore fa clic sull'elemento specificato, si verifica un comportamento dinamico (come un'animazione);
- **onMouseOut="effetto"**: quando il mouse si allontana da un elemento, questo manifesta un certo comportamento dinamico.

Anche la gestione degli effetti DHTML richiede l'apprendimento di un'apposita sintassi. Rispetto agli script, la sintassi DHTML può apparire relativamente più semplice, in quanto il significato delle istruzioni è spesso intuitivo.

In realtà, per utilizzare al meglio le opzioni DHTML occorre completare la pagina Web con opportuni script e fogli di stile.

Si tratta perciò di funzioni riservate a progettisti esperti.

In ogni caso, sussistono controindicazioni all'utilizzo del codice DHTML: al momento, le istruzioni DHTML sono compatibili soltanto con Internet Explorer vers. 4 (o superiore).

Le più recenti versioni del browser Netscape mostrano una tendenza alla compatibilità DHTML, ma per ora è sconsigliabile sviluppare pagine DHTML, se non si ha la certezza che tutti i visitatori utilizzino Internet Explorer vers. 4 (o superiore).

Per questo motivo, sarebbe superfluo approfondire ulteriormente l'argomento in questa sede: si rimandano i lettori interessati ad appositi corsi o siti Web. Nella seguente scheda si può comunque osservare qualche esempio di effetti DHTML applicati al testo della pagina.

Esempi di effetti DHTML

Ecco alcuni semplici esempi di effetti dinamici applicati al testo mediante un utilizzo congiunto di:

- istruzioni DHTML;
- fogli di stile;
- script.

Si ricorda che questi effetti possono essere visualizzati solo se si sta utilizzando Internet Explorer versione 4 (o superiore).

**Al passaggio del mouse,
il testo di questa frase
diventa blu.**

**Al passaggio del mouse,
questa frase diventa rossa
e il puntatore del mouse cambia aspetto.**

**Lo sfondo di questa frase
cambia colore
al passaggio del mouse.**

**Questa frase invece rimane immutata,
ma il puntatore del mouse cambia aspetto.**

Come è stato detto nella sezione, gli effetti DHTML presentano vantaggi e svantaggi:

- il vantaggio principale è la grande ricchezza e varietà di risultati ottenibili: animazioni, menu interattivi e così via;
- gli svantaggi sono la difficoltà di realizzazione (è richiesto uno studio specialistico) e la scarsa compatibilità a livello di browser.

Sezione 24. I fogli di stile: concetti fondamentali

Cosa sono i fogli di stile

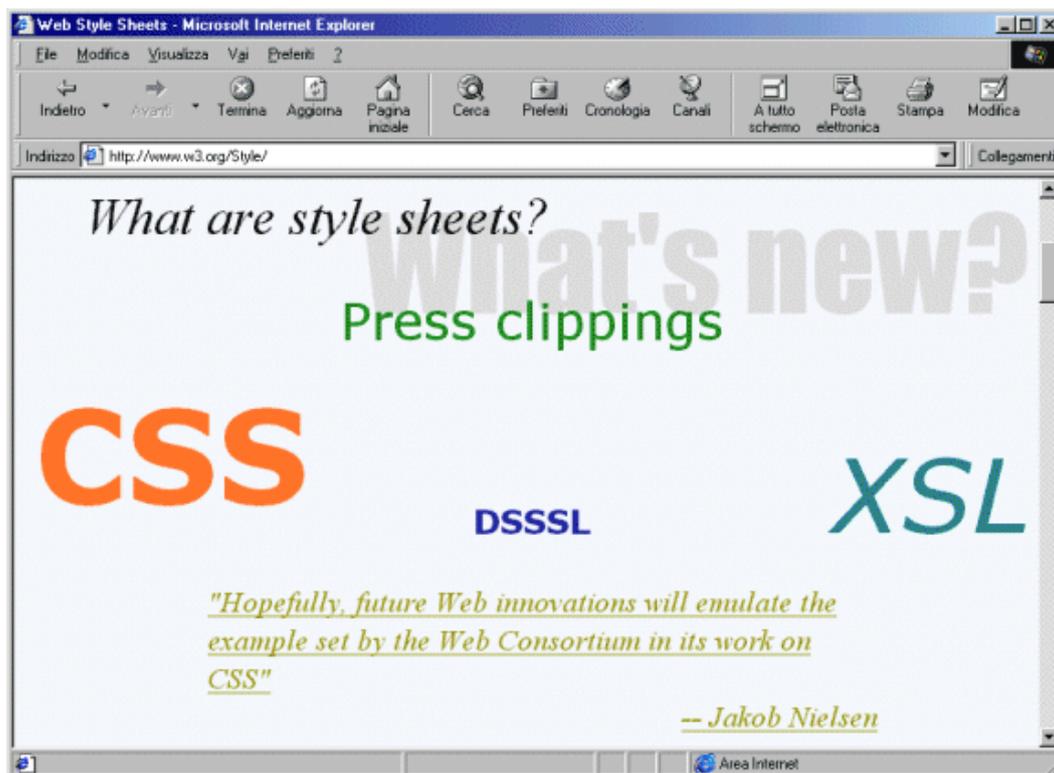
I fogli di stile sono particolari documenti privi di contenuto. Più esattamente, sono modelli di documento, che contengono soltanto una serie di impostazioni stilistiche (criteri di formattazione) liberamente definite dall'autore.

Quando vengono associati a un documento (una pagina Web) o a un gruppo di documenti (un sito Web), tutti i rispettivi contenuti si adeguano automaticamente a tali caratteristiche di formattazione.

È facile comprendere i **vantaggi** straordinari offerti dai fogli di stile, rispetto alla normale creazione di pagine HTML:

- quando si costruiscono a mano le pagine HTML, scrivendone il codice caso per caso, è necessario ripetere i tag di formattazione in corrispondenza di ogni singolo contenuto (paragrafo di testo, immagine ecc.);
- impostando un foglio di stile, **le caratteristiche di formattazione vengono fissate una volta per tutte**: ogni documento associato al foglio di stile adatta automaticamente i suoi contenuti ai criteri stilistici progettati dall'autore;
- inoltre, un sapiente utilizzo dei fogli di stile può produrre **risultati grafici molto eleganti**, che sarebbe difficile ottenere mediante una formattazione manuale con i normali tag HTML.

Ecco un esempio di pagina Web formattata con i fogli di stile:



I fogli di stile sono stati introdotti nella sintassi HTML già da alcuni anni, ma almeno all'inizio ponevano spiccati **problemi di compatibilità**:

- i fogli di stile hanno iniziato a essere correttamente riconosciuti e gestiti solo a partire dalla versione 3.0 di Internet Explorer e dalla versione 4.0 di Netscape Communicator;
- ancora oggi possono verificarsi limiti di compatibilità e risultati indesiderati legati ai tipi di carattere (la visualizzazione dipende dai font effettivamente installati nel computer del visitatore) e ai colori di sfondo della pagina.

Ciononostante, si può prevedere una diffusione sempre maggiore dell'utilizzo dei fogli stile, a mano a mano che la comunità degli utenti Web aggiornerà le versioni dei browser in proprio possesso.

Come funzionano

Il principio di funzionamento di un foglio di stile è il seguente:

1. nel foglio di stile si attribuiscono le caratteristiche desiderate (**stili**) ai vari elementi delle pagine HTML, facendo riferimento ai comuni tag che identificano ciascun elemento: per esempio, si può attribuire ai titoli <H1> un certo colore, ai paragrafi <P> un certo allineamento, al testo normale un certo tipo di carattere e così via;
2. a questo scopo, all'interno del foglio di stile si utilizza una **sintassi** diversa da quella HTML fin qui considerata: per esempio, gli attributi sono inseriti tra parentesi graffe e al posto del segno "=" (uguale) si utilizza il segno ":" (due punti); maggiori dettagli saranno forniti nella prossima sezione;
3. **quando la pagina HTML viene caricata** in un browser compatibile con i fogli di stile, ai vari elementi standard viene automaticamente applicata la formattazione definita dagli stili. Ma come fa il browser ad **associare gli stili a una pagina HTML**? Verrebbe spontaneo pensare che gli stili siano inseriti direttamente all'interno del codice HTML della pagina. Tuttavia, non sempre è così. Infatti, il vantaggio fondamentale dei fogli di stile è appunto quello di stabilire la formattazione di un intero sito senza ripetere le relative istruzioni (tag e attributi) caso per caso. Si hanno dunque a disposizione tre opzioni di utilizzo dei fogli di stile: in linea, incorporati oppure esterni/collegati.

- **Fogli di stile in linea:** le istruzioni di stile vengono inserite volta per volta all'interno dei **singoli tag** di ogni pagina HTML, sotto forma di attributo **STYLE="stile_di_formattazione"**; per formattare un intero blocco di testo, per esempio, si può utilizzare il tag ** ... **.
- **Fogli di stile incorporati:** le istruzioni di stile vengono inserite all'inizio di ogni pagina HTML, tramite la coppia di tag **<STYLE> ... </STYLE>** posti tra la sezione **<HEAD>** e la sezione **<BODY>**; in questo caso, gli stili vengono automaticamente applicati all'**intero documento**.
- **Fogli di stile esterni/collegati:** le istruzioni di stile vengono inserite in **un file a parte con suffisso .css** (senza i tag **<STYLE> ... </STYLE>**); in ogni pagina HTML si aggiunge un collegamento a tale file mediante il tag **<LINK>**.

Fra questi tre tipi di fogli di stile sussiste una gerarchia di tipo funzionale:

1. se in un sito è presente un foglio di stile di tipo **esterno/collegato**, esso determina la formattazione di tutte le pagine collegate;
 2. se però è presente un foglio di stile **incorporato** in una pagina, questo prevale sulla formattazione precedente;
 3. se inoltre, all'interno di una pagina, è presente un foglio di stile **in linea** (singoli attributi di stile), queste istruzioni di formattazione prevalgono su quelle precedenti.
- La strategia più intelligente, perciò, sarebbe quella di predisporre:

- un foglio di stile esterno/collegato per impostare la formattazione generale del sito;
- un foglio di stile incorporato, per le singole pagine che devono assumere una formattazione omogenea al loro interno ma diversa da quella generale del sito;
- facoltativamente, fogli di stile (attributi di stile) in linea per modificare la formattazione di singole parti del testo, qualora lo si ritenga necessario.

Tecniche di realizzazione

Dal punto di vista tecnico, i fogli di stile possono essere costruiti con due metodologie alternative.

Esistono infatti:

- **fogli di stile a cascata (CSS: "Cascading Style Sheet")**: corrispondono esattamente alle descrizioni precedenti; sono riconosciuti dalle versioni recenti di Internet Explorer e Netscape;
- **fogli di stile Javascript**: dal punto di vista logico corrispondono sostanzialmente alle descrizioni precedenti, ma dal punto di vista pratico sono costruiti in modo differente; sono infatti realizzati in linguaggio Javascript (in questo senso, sono analoghi agli script: v. sezione precedente); sono riconosciuti da Netscape ma non da Internet Explorer.

I fogli di stile CSS rappresentano di fatto lo standard nella progettazione Web più sofisticata.

Sezione 25. I fogli di stile a cascata (CSS)

Foglio di stile in linea

Impostare un foglio di stile in linea significa semplicemente **associare attributi di stile a singoli elementi (tag)** della pagina.

L'attributo da aggiungere è **STYLE="stile_di_formattazione"**.

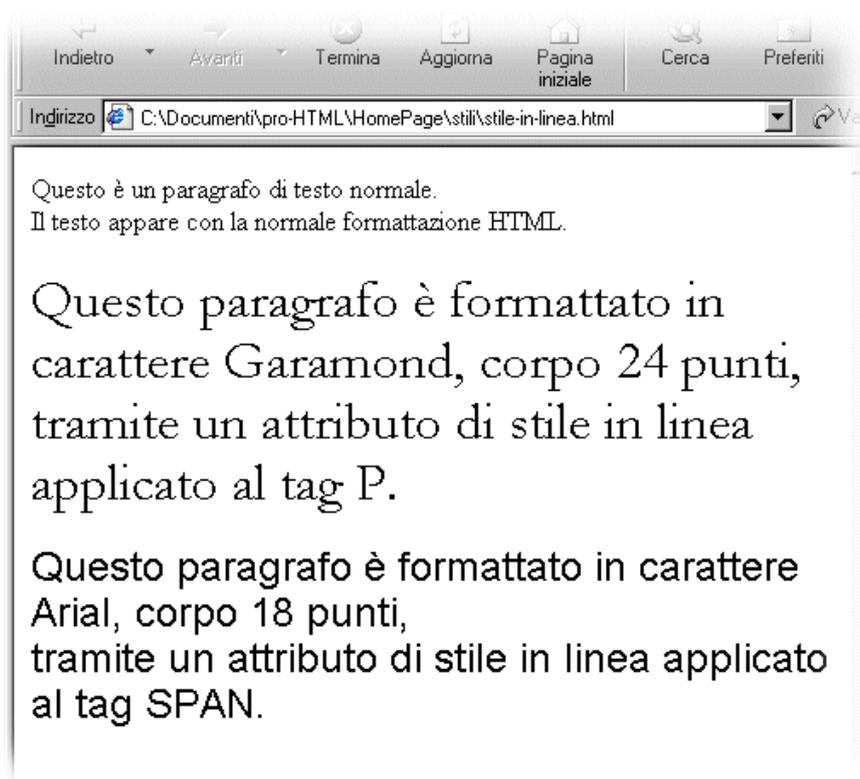
Per esempio:

<P STYLE="font: 24pt garamond">

Questo paragrafo è formattato in carattere Garamond, corpo 24 punti, **
** tramite un attributo di stile in linea applicato al tag P.

</P>

Questo paragrafo è formattato in carattere Arial, corpo 18 punti, **
** tramite un attributo di stile in linea applicato al tag SPAN.



L'attributo STYLE può essere aggiunto ai seguenti tag:

- al tag **<P>**, per applicare lo stile a un singolo paragrafo;
- al tag **<DIV>**, per applicare lo stile a un blocco di testo di qualsiasi dimensione (specie se molto lungo e comprendente più paragrafi);
- al tag ****, per applicare lo stile a una qualunque porzione di testo (paragrafi, parole o anche singole lettere).

Negli esempi precedenti si è applicato uno stile legato al tipo di carattere.

Le opzioni di stile a disposizione sono però numerose: un elenco più dettagliato viene presentato al termine di questa sezione.

Foglio di stile incorporato

Un foglio di stile incorporato è un insieme unitario di impostazioni di formattazioni, poste all'inizio del codice HTML di una pagina Web. Tali impostazioni, inserite una tantum, valgono per **l'intero documento corrente**.

Il foglio di stile incorporato è definito dalla coppia di tag **<STYLE> ... </STYLE>** ed è collocato **tra la sezione <HEAD> e la sezione <BODY>** della pagina:

```
<HTML>
.. <HEAD>
.. (...) intestazione del documento (...)
.. </HEAD>
<STYLE>
.. (...) definizioni degli stili (...)
</STYLE>
.. <BODY>
.. (...) corpo/contenuti del documento (...)
.. </BODY>
</HTML>
```

Nello spazio compreso tra <STYLE> e </STYLE> si possono elencare tutti gli elementi (tag) della pagina a cui si vuole attribuire uno specifico stile. Gli stili sono espressi sotto forma di attributi, racchiusi tra parentesi graffe.

La sintassi generica è:

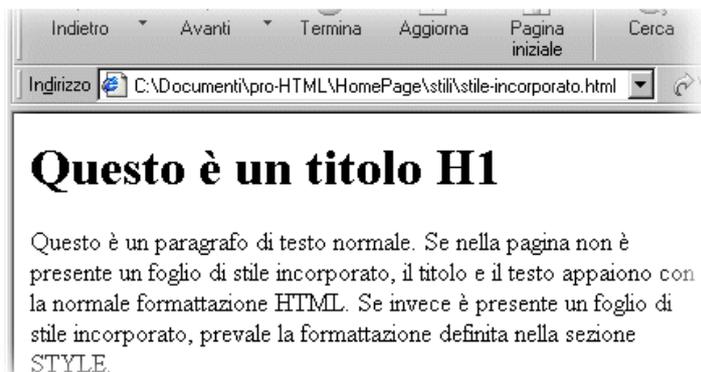
```
TAG<attributo: valore; attributo: valore>
```

Nel corpo della pagina, a tutti gli elementi identificati dal tag vengono applicati gli attributi di stile definiti in questa sede, annullando la formattazione standard prevista dal tag stesso. Per esempio:

```
<HTML>
.. <HEAD>
.. </HEAD>
<STYLE>
.. BODY <background: #606000; color: #000000>
   H1 font: <14pt arial; color: #000080>
   P font: <12pt garamond; text-indent: 0.5in>
</STYLE>
.. <BODY>
.. <H1>Questo è un titolo H1 </H1>
.. <P>Questo è un paragrafo di testo normale.
.. Se nella pagina non è presente un foglio di stile incorporato,
.. il titolo e il testo appaiono con la normale formattazione HTML.
.. Se invece è presente un foglio di stile incorporato,
.. prevale la formattazione definita nella sezione STYLE.</P>
.. </BODY>
</HTML>
```



La stessa pagina SENZA foglio di stile incorporato (formattazione standard HTML) apparirebbe così:



Foglio di stile esterno/collegato

Si può realizzare anche un foglio di stile esterno/collegato.

In questo modo le impostazioni di stile possono essere automaticamente applicate a **tutte le pagine di un sito**.

Ecco la procedura:

1. creare un **file di testo** e inserire al suo interno le istruzioni riguardadanti gli stili; la sintassi è identica a quella dei fogli di stile incorporati (v. sopra) ma **non** si usa la coppia di tag **<STYLE> ... </STYLE>**;
2. salvare questo file con il **suffisso ".css"** (per esempio: **mio_stile.css**);
3. nel codice HTML di **ogni pagina** del sito Web, all'interno della sezione **<HEAD> ... </HEAD>**, inserire un collegamento al foglio di stile esterno con la seguente sintassi: **<LINK REL="stylesheet" HREF="mio_stile.css" TYPE="text/css">**.

.css.

Tutte le pagine del sito recepiscono automaticamente le nuove impostazioni di formattazione.

Attributi di stile

Una volta compresi questi principi generali, per costruire fogli di stile è sufficiente apprendere la sintassi dei vari possibili attributi di formattazione.

Si tratta di un argomento piuttosto vasto, che esula dagli obiettivi di questo modulo. Tuttavia, per chi fosse interessato è disponibile la seguente scheda di approfondimento, in cui vengono presentati alcuni degli attributi più comuni.

I principali attributi di stile

Dopo aver assimilato i meccanismi fondamentali per la gestione dei fogli di stile, ecco una rapida rassegna dei principali attributi di stile che è possibile utilizzare.

Testo

Ecco i principali attributi di stile relativi alla formattazione del testo:

- **{font-family}** definisce il tipo di carattere;
esempio: `{font-family: arial, helvetica, sans-serif}`
- **{font-size}** definisce le dimensioni dei caratteri in:
punti: ad esempio `{font-size: 11pt}`
pollici: ad esempio `{font-size: 1in}`
centimetri: ad esempio `{font-size: 1cm}`
pixel: ad esempio `{font-size: 18px}`
percentuale: ad esempio `{font-size: 80%}`
- **{font-style}** definisce lo stile grafico del testo, ossia gli attributi di formattazione;
esempio: `{font-style: bold}`
- **{text-decoration}** definisce eventuali effetti ornamentali, come la sottolineatura; i valori possibili sono: `none`, `underline`, `italic`, `line-through`;
esempio: `{text-decoration: underline}` si noti che: tramite questo attributo si può far sì che i collegamenti ipertestuali non appaiano mai sottolineati;
 - se si usa un foglio di stile incorporato o collegato basta inserire nel foglio questa stringa:
`A {text-decoration: none}`
 - se si ricorre a un foglio di stile in linea, basta aggiungere il corrispondente attributo STYLE a ogni singolo collegamento; esempio:
``
vai a pagina 2 ``

Margini e allineamento

Per impostare i margini e gli allineamenti si possono utilizzare questi attributi di stile:

- **{margin-left}** definisce il margine sinistro in pollici, centimetri, pixel o punti (la sintassi delle unità di misura è analoga a quella appena descritta per le dimensioni dei caratteri: v. sopra);
esempio: `{margin-left: 0,5cm}`
- **{margin-right}** definisce il margine destro; la sintassi è sempre la stessa;
esempio: `{margin-right: 1cm}`
- **{margin-top}** definisce il margine superiore rispetto all'inizio del documento; la sintassi è sempre la stessa;
esempio: `{margin-top: 24pt}`.
- **{text-indent}** definisce un rientro standard per ogni paragrafo di testo (ma non, ad esempio, per le immagini e gli altri elementi della pagina); la sintassi è sempre la stessa;
esempio: `{text-indent: 2cm}`

Colori e sfondo

Per impostare i colori e/o lo sfondo di specifiche parti del documento si hanno a disposizione i seguenti attributi di stile:

- **{background}** definisce lo sfondo di un qualsiasi elemento (intero documento, paragrafo, titolo eccetera); basta aggiungere l'attributo alla opportuna voce del foglio di stile e specificarne il valore, che può essere:
un'immagine; esempio: `{background: sfondo01.gif}`
un colore; esempio: `{background: #FF00CC}`
- **{color}** definisce il colore del testo;
esempio: `{color: #80EAFB}`

Nota sull'utilizzo degli stili

Naturalmente, vari attributi di stile possono essere combinati in una stessa voce del foglio di stile.

Ad esempio, si può in un colpo solo definire il tipo di carattere, le dimensioni e il colore del testo per i paragrafi normali, in questo modo:

```
P {font-family: arial; font-size: 14pt; color: #800000}
```

Personalizzazione di un foglio di stile incorporato

Nel seguente campo di testo si trovano già i tag essenziali che definiscono la struttura di una pagina Web.

Nella pagina è preimpostato un **foglio di stile incorporato**:

- sono definiti **stili** per vari elementi della pagina (sfondo, corpo del testo, titoli H1);
- i **valori** inseriti sono fittizi ("**XXXXXX**").

Personalizzare il foglio di stile, sostituendo le stringhe **XXXXXX** con opportuni valori, in modo tale che:

- lo **sfondo** della pagina sia di colore marrone-bordeaux (codice esadecimale: #666633);
- i **titoli** siano di colore azzurro-acqua (codice esadecimale: #00FFFF);
- i **paragrafi** di testo normale siano di colore bianco (codice esadecimale: #FFFFFF).

Nota

NON sono ammesse variazioni.

Esercitazione

Per sperimentare le competenze maturate in questa lezione del modulo, ecco un esercizio di riepilogo.

Creare una pagina Web che contenga:

- un foglio di stile incorporato (con impostazioni di formattazione a libera scelta dell'autore);
- uno script di tipo Alert con il messaggio: **Questa pagina contiene un modulo e un foglio di stile!**
- un titolo e un breve testo a cui applicare le formattazioni del foglio di stile;
- un semplice modulo interattivo per la spedizione di un messaggio e-mail.

Per proseguire...

Giunti al termine del modulo, ecco un riepilogo di **tutto ciò che si è imparato** attraverso i vari moduli di sezioni.

- Si è costruita innanzitutto una visione d'insieme circa la natura e la storia del linguaggio HTML e del Web. Si sono chiariti i principi di funzionamento delle pagine HTML e dei tag.
- Sono state apprese le **tecniche di base** per creare, salvare, modificare i file HTML. Si è appreso come impostarne le caratteristiche fondamentali: struttura intestazione/corpo, personalizzazione di alcuni parametri fondamentali (ad esempio lo sfondo), definizione dei collegamenti ipertestuali.
- Al termine del primo gruppo di sezioni, si era già in grado di produrre le prime pagine HTML, semplici ma funzionanti.
- Sono stati quindi approfonditi gli aspetti riguardanti la gestione del **testo**: caratteri e simboli, attributi grafici, formattazione di paragrafi e blocchi di testo.
- Sono stati analizzati elementi di testo particolari, come gli elenchi e i titoli.
- Successivamente si è sviluppata la competenza nella gestione dei **colori** e delle **immagini**. In primo luogo, è stato esplorato in modo approfondito il metodo di codifica dei colori nel linguaggio HTML.
- Si è visto il significato del concetto di "ottimizzazione" delle immagini e sono stati esaminati alcuni esempi applicativi.
- Sono state apprese le basi per la creazione e la gestione delle mappe cliccabili. Si sono scoperte in seguito le potenzialità di utilizzo delle **tabelle**, anche come strumento di impaginazione.
- Si è imparato a gestire ogni aspetto dell'impostazione e personalizzazione delle tabelle e dei loro contenuti.
- Passando ad aspetti più complessi, si è scoperto cosa sono i **frame** e come funzionano.
- È stata studiata e sperimentata l'impostazione di strutture basate sui frame e la personalizzazione delle loro caratteristiche.
- L'ultima parte del modulo ha guidato un'esplorazione di alcune caratteristiche avanzate del mondo HTML, che richiederebbero un superiore livello di competenza. Sono stati innanzitutto studiati e sperimentati i concetti essenziali relativi ai **moduli** e agli **script**.
- Particolare attenzione, infine, è stata dedicata ai **fogli di stile**. Si è assimilato un inquadramento generale dei fogli di stile e si è studiato da vicino il funzionamento dei fogli CSS (in particolare quelli incorporati).
- Tutte le funzioni principali studiate nel modulo sono state verificate passo per passo mediante singoli **esercizi pratici**.
- Al termine di ogni fase (modulo o gruppo di sezioni), esercizi di più ampio respiro hanno consentito di sperimentare con maggiore libertà le tecniche apprese e hanno stimolato la capacità di realizzare in formato HTML **progetti** via via più complessi.

Da qui in poi, **per proseguire** si potranno approfondire singoli argomenti o sviluppi specialistici adottando varie modalità e strumenti:

- corsi o manuali su funzioni HTML avanzate e dinamiche (DHTML, pagine ASP, scripting, interfacce Web per l'interrogazione di database e così via);
- corsi o manuali sui principali linguaggi di programmazione per il Web: Java, Perl, altri ambienti di sviluppo professionali.

Ora resta da svolgere un **esercizio conclusivo** in cui mettere a frutto tutte le competenze acquisite nell'intero modulo.

ESERCIZIO FINALE

Costruzione di un piccolo sito web a tema libero