

PRINCIPI DI HTML 4.0

1	ELEMENTI FONDAMENTALI.....	2
1.1	INTRODUZIONE.....	2
1.2	SINTASSI DI UNA REGOLA E DI UN TAG	4
2	FORMATTAZIONE DEL TESTO.....	5
2.1	CORPO DEL TESTO.....	5
2.2	TITOLI e PARAGRAFI.....	5
2.3	STRUTTURA DEL TESTO.....	6
2.4	LISTE	7
3	TABELLE	9
4	FORMATTAZIONE DEI CARATTERI E DEI PARAGRAFI	11
4.1	CARATTERI SPECIALI.....	11
4.2	I FONT.....	12
5	COLLEGAMENTI	13
6	IMMAGINI E MAPPE SENSIBILI.....	15
6.1	IMMAGINI.....	15
6.2	MAPPE SENSIBILI	16
7	FRAMES	17
8	MODULI	21
9	FOGLI DI STILE.....	23

1 ELEMENTI FONDAMENTALI

1.1 INTRODUZIONE

Un documento HTML è un file in cui il testo in esso contenuto è formattato e strutturato grazie all'inclusione di speciali comandi di markup (o tag) delimitati dai simboli “maggiore” (iniziatore del comando) e “minore” (terminatore del comando). Ciò sta a significare che qualunque sequenza di caratteri compresa tra un iniziatore ed un terminatore non fa parte del testo bensì della sua struttura. In pratica testo e metatesto convivono nel medesimo documento.

Un siffatto documento, quindi, necessita di qualche tipo di programma che lo legga e ne interpreti i comandi in modo da “calare” il testo all'interno della struttura specificata. Tale programma viene definito “Parser”, ovvero “Valutatore grammaticale” perché, in effetti, un documento HTML possiede una grammatica intrinseca che necessariamente bisogna rispettare (come in qualunque linguaggio di programmazione).

Dato che un parser probabilmente è in grado di interpretare diversi linguaggi, deve anche essere in grado di determinare quali regole grammaticali utilizzare per l'interpretazione. Affinché ciò avvenga è necessario, prima di tutto, scrivere nel documento un comando che segnali, appunto, al parser che il linguaggio utilizzato è proprio l'HTML.

Immaginiamo di avere un file di testo all'interno del quale scriviamo solamente due righe come le seguenti:

```
file: prova.htm
```

```
<HTML>
</HTML>
```

Questo è un primo essenziale documento HTML. Il parser legge la prima riga e capisce di dover utilizzare le regole per l'interpretazione di documenti HTML e lo farà finché il tag non verrà “chiuso”. Ecco una nuova norma: una regola HTML è composta da due tag che delimitano il testo a cui applicare la regola stessa; il tag di chiusura è composto dallo stesso tag di apertura ma preceduto da una barra. In questo caso tra i due tag non c'è alcun testo e il tag si dice “vuoto”¹.

L'HTML nasce come linguaggio di formattazione logica e quindi possiede molte regole per la strutturazione del documento, ma col passare del tempo e l'ampio diffondersi del linguaggio si è notato che una medesima regola logica poteva avere diverse rappresentazioni grafiche (ancora oggi Netscape e Internet Explorer conservano queste differenze di rappresentazione). Nacque così l'esigenza di creare nuove regole per la formattazione del testo (quindi non più a livello di struttura ma a livello di layout). E' sempre necessario sapere se stiamo strutturando il documento a livello logico o a livello di layout per essere consapevoli dei punti in cui parser differenti possano interpretare diversamente le nostre istruzioni.

All'interno, una pagina HTML è suddivisa principalmente in due parti: l'intestazione (che conterrà il titolo della pagina, le indicazioni sul contenuto e gli stili) e il corpo, con il documento vero e proprio. Si provi a costruire, con un editor di testo, un file come il seguente:

¹ In XML (un altro linguaggio parente stretto dell'HTML) i tag vuoti del tipo <nometag></nometag> (e quindi senza alcun testo tra il tag iniziale e finale) vengono sostituiti dalla scrittura più breve <nometag/>.

file: prova.htm

```
<HTML>

<HEAD>
  <TITLE>
    Questo è il titolo della mia pagina di prova
  </TITLE>
</HEAD>

<BODY>
  Questo è il testo del documento
</BODY>

</HTML>
```

Il documento HTML contiene due regole: HEAD e BODY. A sua volta HEAD contiene una regola TITLE. Spostarsi a destra ogni qualvolta si inserisce una regola all'interno di un'altra e spostarsi a sinistra alla sua chiusura è facoltativo ma aumenta la comprensione del testo. Questo modo di scrivere il codice viene definito “con indentatura” e l'atto si dice di “indentare” o, in inglese, “indent”.

Salvando questo file con il nome “prova.htm” ed eseguendolo con un qualunque parser otteniamo una pagina in cui compare il testo “Questo è il testo del documento”, ed il titolo della barra del parser è “Questo è il titolo della mia pagina di prova”. Modificando in maniera significativa il testo del documento si nota che:

- anche andando a capo nel testo il parser rappresenterà il documento ignorando queste modifiche;
- anche se si inseriscono più spazi di seguito o caratteri di tabulazione il parser continuerà ad ignorarli;
- i tag vengono interpretati sia se scritti in minuscolo che in maiuscolo (o, al limite, metà e metà);
- si possono inserire spazi vuoti tra il nome del tag ed il suo carattere terminatore (“>”);
- ecc.

1.2 SINTASSI DI UNA REGOLA E DI UN TAG

- Abbiamo già osservato che una regola (che noi, più o meno impropriamente chiameremo talvolta “comando”) è composta da due tag: uno iniziale ed uno finale:

```
<nometag> testo a cui applicare la regola </nometag>
```

- Le regole si possono annidare:

```
<nometag1> solo regola 1
  <nometag2> valgono entrambe le regole </nometag2>
  vale ancora solo la regola 1
</nometag1>
```

- Sebbene caldamente sconsigliato (e in XML assolutamente scorretto), le regole si possono incastrare:

```
<nometag1>
  qui vale la regola 1
  <nometag2>
    qui valgono entrambe le regole
  </nometag2>
  qui vale solo la regola 2
</nometag1>
```

E’ da notare che l’indent nel caso di regole “incastrate” non è più così significativo; in effetti questo modo di programmare viola le regole della programmazione strutturata, e ciò “è male”.

- I tag possono essere corredati di “attributi”

```
<nometag1 attributo1="valore1" attributo2="valore2">
  testo a cui applicare la regola con attributi
</nometag1>
```

L’uso degli attributi sarà chiarito più avanti. Al momento basti sapere che non sempre è obbligatorio inserire il valore di un attributo tra virgolette, ma è caldamente consigliato².

² Ogni “consiglio” viene dato non solo per indurre a produrre secondo le regole di buona programmazione, ma anche perché l’HTML ha subito una notevole espansione, anche tra i non esperti; il prezzo di questa popolarità è stato, appunto, l’involgarimento della grammatica. Ogni altro linguaggio (sia esso di markup oppure no) non supporta questa flessibilità, compreso l’XML a cui molti dei programmatori HTML dovranno inevitabilmente rivolgersi tra qualche tempo. Adottando queste norme il passaggio sarà, dunque, più morbido.

2 FORMATTAZIONE DEL TESTO

Seguono i comandi principali per la formattazione del documento. Laddove possibile si è accostato ad uno stile fisico il relativo stile logico.

2.1 CORPO DEL TESTO

Per il tag BODY sono disponibili alcuni attributi particolarmente utili, come:

- TEXT che consente di specificare il colore di default del testo all'interno del corpo;
- BACKGROUND che consente di specificare l'indirizzo di una immagine da utilizzare come sfondo;
- BGPROPERTIES con il valore FIXED che consente di far rimanere l'immagine di sfondo fissa durante lo scrolling della pagina.

2.2 TITOLI e PARAGRAFI

Anche se in maniera poco corretta si usano alcuni tag senza porsi il problema di chiuderli. I due più importanti sono:

 che consente di andare a capo

<HR> che consente di inserire una riga orizzontale nel documento

Esempi di comandi completi sono invece i seguenti:

<H1> Titolo grande </H1>

<H2> Titolo più piccolo </H2>

...

<H6> Titolo davvero piccolo </H6>

<NOBR>Qualunque cosa accada in questo blocco non si va a capo</NOBR>

<P> Paragrafo di testo.</P>



La prima riga di un paragrafo è in interlinea doppio mentre ogni paragrafo inizia su una riga nuova e termina in modo che il testo seguente vada a capo. Un paragrafo si può allineare a destra, a sinistra o centrarlo. Ciò avviene attraverso l'utilizzo dell'attributo ALIGN che può assumere i valori rispettivamente "right", "left" e "center".

2.3 STRUTTURA DEL TESTO

Grassetto (bold)

Stile fisico: Questo testo è scritto in grassetto

Stile logico: Ma anche questo (STRONG sta per forte enfasi)

Corsivo (italic)

Stile fisico: <I>Questo, invece, è in corsivo</I>

Stile logico: e, naturalmente anche questo, (EM sta per enfasi)

Stile logico: <CITE>compreso quest'altro</CITE> (CITE sta per citazione)

Sottolineato (underlined)

Stile fisico: <U>Testo sottolineato</U>

Margini

Stile logico: <BLOCKQUOTE>Questo testo ha dei margini più stretti</BLOCKQUOTE>

Nota: i comandi <BLOCKQUOTE> si possono annidare per rendere i margini progressivamente più stretti.

...

Tutti i comandi possono essere annidati, così un file HTML come quello seguente:

```
<HTML>

<HEAD>
  <TITLE>Prova della formattazione annidata</TITLE>
</HEAD>

<BODY>
  <H1>PROVA FORMATTAZIONE</H1>
  Questo testo contiene diversi comandi.<BR>Ad ogni comando corrisponde
  uno stile del testo.<BR>Ad esempio,
  parte del testo è in <B>grassetto</B>, parte in <I>corsivo</I> e
  parte viene <U>sottolineato</U>; inoltre c'è anche del testo
  <U><B>sottolineato in grassetto, <I>sottolineato in grassetto e corsivo</U> e
  in grassetto corsivo</B></I>.
</BODY>

</HTML>
```

Il risultato, nel browser, è il seguente:

PROVA FORMATTAZIONE

Questo testo contiene diversi comandi.

Ad ogni comando corrisponde uno stile del testo.

Ad esempio parte del testo è in **grassetto**, parte in *corsivo* e parte viene sottolineato; inoltre c'è anche del testo **sottolineato in grassetto, sottolineato in grassetto e corsivo e in grassetto corsivo**.

Come già accennato il testo formattato precedentemente non è un buon esempio di programmazione in quanto “incastra” le regole e non le annida. Il testo potrebbe essere riscritto come segue:

```
<BODY>
  <H1>PROVA FORMATTAZIONE</H1>
  Questo testo contiene diversi comandi.<BR>Ad ogni comando corrisponde
  uno stile del testo.<BR>Ad esempio,
  parte del testo è in <B>grassetto</B>, parte in <I>corsivo</I> e
  parte viene <U>sottolineato</U>; inoltre c'è anche del testo
  <B>
  <U>sottolineato in grassetto,
  <I>sottolineato in grassetto e corsivo
  </I>
  </U>
  <I>
  e in grassetto corsivo
  </I>
  </B>.
</BODY>
```

Il risultato è il medesimo, ma è possibile indentare il codice aumentandone la leggibilità.

2.4 LISTE

E' possibile creare degli elenchi automaticamente. Sia puntati che numerati.

Elenco puntato:

```
<B>COSE DA COMPERARE</B>
<UL>
  <LI>Pane</LI>
  <LI>Latte (parzialmente <B>scremato</B>)</LI>
  <LI>Birra</LI>
  <LI>Pasta (Barilla, <U>spaghetti n°4</U>)</LI>
</UL>
```

Il risultato è il seguente:

COSE DA COMPERARE

- Pane
- Latte (parzialmente **scremato**)
- Birra
- Pasta (Barilla, spaghetti n°4)

Elenco numerato:

```
<B>ELENCO NUMERATO</B>
<OL>
  <LI>prima voce</LI>
  <LI>seconda voce</LI>
</OL>
```

Il cui risultato è, invece, il seguente:

ELENCO NUMERATO

1. prima voce
2. seconda voce

Esistono altri tipi di elenchi, come gli elenchi di definizioni (<DL>) in cui ogni voce è composta da un “Definition Term” (<DT>) e da una “Definition Defined” (<DD>); oppure le liste di directory (<DIR>). Il tag possiede l’attributo “TYPE” i cui valori permettono di scegliere il tipo di numerazione dell’elenco. Ad esempio la lista seguente:

```
<B>ELENCO NUMERATO</B>
<OL TYPE="a">
  <LI>prima voce</LI>
  <LI>seconda voce</LI>
</OL>
```

Produce il seguente effetto:

ELENCO NUMERATO

- a. prima voce
- b. seconda voce

I valori che si possono assegnare all’attributo TYPE sono i seguenti:

Valore	Numerazione
1	Numeri arabi 1. prima voce 2. seconda voce ...
a	Lettere minuscole a. prima voce b. seconda voce ...
A	Lettere maiuscole A. prima voce B. seconda voce ...
i	Numeri romani minuscoli i. prima voce ii. seconda voce iii. terza voce iv. quarta voce ...
I	Numeri romani maiuscoli I. prima voce II. seconda voce III. terza voce IV. quarta voce ...

3 TABELLE

E' possibile definire la struttura di tabelle utilizzando il comando TABLE. Questo consente di specificare, al suo interno, righe e colonne e, in ogni cella risultante sarà possibile inserire qualunque cosa, come del testo comunque formattato, immagini (come vedremo) e altre tabelle. I tag utili alla creazione di una tabella sono i seguenti:

```
<TABLE> per la definizione della tabella
<TR> (table row) per l'aggiunta di una riga
<TD> (table data) per la creazione delle colonne all'interno di ogni riga
```

TABLE può possedere alcuni attributi, tra cui:

- BORDER seguito da un numero (ad esempio BORDER="4") che ne indica lo spessore (in pixel).
- WIDTH seguito da un numero (ad esempio WIDTH="200" oppure WIDTH="80%") o da una percentuale che indicano la larghezza della tabella. Il numero indica la larghezza in pixel mentre la percentuale consente di specificare la larghezza della tabella rispetto all'area occupabile
- Anche <TR> e <TD> possono possedere l'attributo WIDTH; in particolare è molto interessante poter specificare larghezze relative (in percentuale) e assolute (in pixel) all'interno della stessa tabella.

Facciamo un esempio completo.

```
<TABLE BORDER="1" WIDTH="80%">
<TR> <B> FATTURA </B>
</TR>

<TR>
  <TD           ><B>PRODOTTO           </B></TD>
  <TD WIDTH="90" ><B>QUANTITA'         </B></TD>
  <TD WIDTH="150"><B>PREZZO UNITARIO    </B></TD>
  <TD WIDTH="100"><B>TOTALE             </B></TD>
</TR>

<TR>
  <TD>Involtino plimavela</TD>
  <TD>4</TD>
  <TD>1.500</TD>
  <TD>6.000</TD>
</TR>

<TR>
  <TD>Budello maiale</TD>
  <TD>2</TD>
  <TD>8.000</TD>
  <TD>16.000</TD>
</TR>

<TR>
  <TD>TOTALE</TD>
  <TD></TD>
  <TD></TD>
  <TD>22.000</TD>
</TR>

</TABLE>
```

Il risultato sarà simile al seguente:

FATTURA			
PRODOTTO	QUANTITA'	PREZZO UNITARIO	TOTALE
Involto plimavela	4	1.500	6.000
Budello maiale	2	8.000	16.000
TOTALE			22.000

Come si può notare sarebbe opportuno allineare le cifre d destra e, magari, centrare i titoli delle colonne. Queste operazioni si potranno effettuare alla luce delle informazioni apprese nel prossimo capitolo.

4 FORMATTAZIONE DEI CARATTERI E DEI PARAGRAFI

4.1 CARATTERI SPECIALI

Ci si potrebbe chiedere come sia possibile inserire in un documento HTML i caratteri “<” o “>”; o più spazi consecutivi o ancora caratteri speciali non presenti sulla tastiera (come, ad esempio, “½”): L’HTML consente di inserire caratteri attraverso i loro “codici”; cioè ad ogni carattere è associato un codice numerico che lo identifica univocamente. L’inserimento avviene in questo modo. Supponiamo di voler far comparire nel documento la frase:

tutti sanno che 4 < 3

Il codice HTML da scrivere sarà il seguente:

tutti sanno che 4 P 3

In cui “80” è il codice del carattere “minore” e la scritta “P” (con la “e” commerciale ed il punto e virgola obbligatori) indica proprio al browser di visualizzare il carattere di codice 80. Molti di questi caratteri, inoltre, sono talmente frequenti che vi è stato attribuito un codice mnemonico in sostituzione del numero. Proprio a “minore di” corrisponde anche il codice mnemonico “<”.

Segue la lista dei principali caratteri speciali con i relativi codici numerici e mnemonici.

Carattere	Codice numerico	Codice mnemonico
&	&	&
<	<	<
>	>	>
(spazio)	 	
...

4.2 I FONT

il tag FONT consente di specificare il tipo e le caratteristiche del carattere o del testo da formattare. I suoi principali attributi sono:

- SIZE a cui segue la dimensione del testo desiderata;
- FACE seguito dal nome del carattere;
- COLOR seguito dal colore.

Ad esempio il codice seguente:

```
Questa è una prova:
<FONT COLOR="#FF0000" FACE="Algerian" SIZE="5"> CIAO A TUTTI.</FONT>
Fine della prova
```

Produce il risultato seguente:

Questa è una prova: **CIAO A TUTTI.** Fine della prova

I colori fondamentali si possono indicare direttamente con il loro nome: blue, red, yellow, black, white, ...Gli altri si indicano con il carattere “#” seguito da un numero esadecimale a 6 cifre. Ogni coppia indica la “quantità” di colore primario da utilizzare secondo lo schema “RRGGBB” (R=rosso, G=verde, B=blu). #FF0000 significa: “255 parti di rosso (il massimo), zero parti di verde e zero parti di blu”. In pratica è un rosso pieno; secondo quanto detto prima si sarebbe potuto scrivere: COLOR=“red”.



I tipi dei caratteri, come Arial, Times, Courier, ecc. dipendono estremamente dal sistema operativo su cui la pagina viene interpretata. Con il MAC, ad esempio, gran parte dei font comuni in Windows non esistono (e viceversa). Il Browser decide autonomamente che font utilizzare in sostituzione a quello mancante.

5 COLLEGAMENTI

Veniamo, ora, a definire uno degli elementi fondamentali di una pagina HTML: la possibilità di inserire collegamenti verso altre pagine. Prima di tutto definiamo “Uniform Resource Locator” (abbreviato URL) qualsiasi tipo di indirizzo internet. Questa definizione è importante in quanto i collegamenti all’interno di una pagina possono contenere o riferimenti locali o URL di altri documenti; inoltre, oltre ai collegamenti esterni (quindi di un documento verso altri documenti) è possibile inserire collegamenti interni (ossia del documento verso altre parti dello stesso documento). Il tag che consente di realizzare i collegamenti è A. Possiede fundamentalmente due attributi:

- HREF seguito dall’URL o dal riferimento locale o interno a cui dirigersi una volta cliccato sul testo chiuso nei tag A e /A;
- NAME che attribuisce un nome al collegamento cosicché possa essere referenziato da altri collegamenti.

L’esempio seguente si riferisce ad una pagina contenente tre collegamenti a pagine esterne puntando direttamente al loro URL

```
<H1>INDICE DEGLI ARGOMENTI</H1>

<A HREF="http://www.miosito.com/doc/cap1.htm">Capitolo primo </A> <BR>
<A HREF="http://www.miosito.com/doc/cap2.htm">Capitolo secondo </A> <BR>
<A HREF="http://www.miosito.com/doc/cap3.htm">Capitolo terzo </A> <BR>
```

Il cui risultato è il seguente:

INDICE DEGLI ARGOMENTI

[Capitolo primo](#)
[Capitolo secondo](#)
[Capitolo terzo](#)



Tutti i tipi di collegamenti sono da considerarsi case-sensitive; cioè una stessa lettera viene interpretata diversamente se scritta in maiuscolo o in minuscolo. Ad esempio anche se ciò non è del tutto vero “http://www.miosito.com” e “http://www.MioSito.com” sono da considerarsi indirizzi differenti.

Abbiamo imparato ad utilizzare gli URL. Nel caso in cui ci si volesse riferire ad un file presente sul web server corrente è possibile specificare solo il percorso relativo. Immaginiamo di implementare un sito che avrà l’indirizzo www.miosito.com e, nella cartella “doc” di questo sito ci sono i file “doc1.htm”, “doc2.htm”, “doc3.htm”. L’ipotetico file index.htm posizionato nella root del server avrà una struttura analoga a quella precedente in cui, però, gli URL potranno essere sostituiti con i riferimenti locali:

```
<H1>INDICE DEGLI ARGOMENTI</H1>

<A HREF="/doc/cap1.htm">Capitolo primo </A> <BR>
<A HREF="/doc/cap2.htm">Capitolo secondo </A> <BR>
<A HREF="/doc/cap3.htm">Capitolo terzo </A> <BR>
```

I collegamenti interni, come anticipato, consentono di riferirsi ad una parte specifica di un documento. Scriviamo l'ipotetico codice del capitolo 1 a cui ci si è riferiti negli esempi precedenti:

```
<H1>CAPITOLO PRIMO</H1>
<A HREF= "#par1">Paragrafo uno</A><BR>
<A HREF= "#par2">Paragrafo due</A><BR>
<A HREF= "#par3">Paragrafo tre</A><BR>

<BR>

<H2><A NAME="par1">PARAGRAFO UNO</A></H2>
<P>Testo del paragrafo 1</P>

<H2><A NAME="par2">PARAGRAFO DUE</A ></H2>
<P>Testo del paragrafo 2</P>

<H2><A NAME= "par3">PARAGRAFO TRE</A ></H2>
<P>Testo del paragrafo 1</P>
```

Cliccando su una dei tre collegamenti sotto il titolo "CAPITOLO PRIMO" il browser muoverà la pagina in modo che la riga il cui nome del collegamento corrisponde a quello cliccato diventerà la prima a partire dall'alto. In pratica un collegamento interno è preceduto dal carattere "#".

Collegamenti interni ed esterni possono essere combinati:

```
<A NAME="su" HREF="giu">vai giù</A>
...
<A NAME="giu" HREF="su">torna su</A>
```

6 IMMAGINI E MAPPE SENSIBILI

6.1 IMMAGINI

L'inserimento di immagini è molto semplice. Si utilizza il comando IMG unitamente ad alcuni suoi attributi. IMG è uno dei tag HTML a non avere il relativo tag di chiusura (come BR e HR). Laddove sia necessario includere un'immagine (all'interno di una tabella, nel corpo del testo o in qualunque altra parte) è necessario scrivere almeno il codice seguente:

```
<IMG SRC="http://www.miosito.com/immagini/fotomiocane.jpg">
```

in cui "http://www.miosito.com/immagini/fotomiocane.jpg" è l'URL dell'immagine da visualizzare. l'attributo SRC può anche indicare un riferimento relativo all'interno del web server. Altri attributi importanti di IMG sono i seguenti:

- ALT che consente di specificare una stringa che compare in sostituzione dell'immagine nei browser che non riescono o non desiderano caricare l'immagine specificata;
- ALIGN i cui valori possibili consentono di posizionare in modo specifico l'immagine sulla riga di testo corrente (seguono i valori possibili)

Valori possibili di ALIGN:	
top	il lato superiore è allineato con il testo della stessa riga
bottom	il lato inferiore è allineato con il testo della stessa riga
middle	per le immagini allineate lungo la linea mediana

Altri valori possibili sono: texttop, absmiddle, baseline, absbottom.

- HEIGHT che consente di specificare l'altezza in pixel dell'immagine (se viene omissa questo attributo l'immagine assume l'altezza originale);
- WIDTH è analogo a HEIGHT ma consente di impostare la larghezza dell'immagine;
- LOWSRC consente di impostare un URL di un'immagine a bassa risoluzione che verrà caricata immediatamente; successivamente, una volta terminato il caricamento dell'immagine vera e propria, il browser si occuperà della sostituzione;
- BORDER permette di impostare lo spessore del bordo dell'immagine.

E' possibile assegnare un collegamento ad un'immagine semplicemente chiudendola all'interno di un comando anchor (<A>). In questo caso cliccando ovunque nell'immagine verrà caricato il collegamento specificato dall'attributo HREF.

6.2 MAPPE SENSIBILI

Una mappa sensibile è una immagine in cui alcune parti sono connesse ad un indirizzo. Queste aree cliccabili possono essere rettangolari, circolari o poligonali. L'esempio seguente è abbastanza esplicativo.

```
<IMG SRC="http://www.miosito.com/05.jpg" USEMAP="#01" BORDER="0">

<MAP NAME="01">
  <AREA SHAPE="rect"    COORDS="102,38,242,138" HREF="lombardia.htm" TITLE="">
  <AREA SHAPE="circle" COORDS="173,232,45"      HREF="piemonte.htm"  TITLE="">
  <AREA SHAPE="poly"
    COORDS=" 37,191,
            26,221,
            60,273,
            102,272,
            123,225,
            99,189,
            37,191" HREF="liguria.htm" TITLE="">
  <AREA SHAPE="default" NOHREF>
</MAP>
```

Si procede come segue:

- Si attribuisce all'immagine definita in un tag IMG un attributo USEMAP al quale si assegna il nome della mappa da utilizzare;
- si crea, all'interno di una regola MAP la struttura della mappa.

Una regola MAP si riferisce ad una determinata immagine attraverso il riferimento dell'attributo NAME; successivamente all'interno della regola è possibile definire un elenco di regole AREA che consentono, ognuna, di definire, all'interno della mappa, un'area cliccabile collegata ad un determinato indirizzo.

Una regola AREA, come appena descritto, consente di specificare un'area cliccabile. I suoi attributi sono:

- SHAPE che specifica la forma dell'area. Può essere: RECTANGLE, CIRCLE, POLY e DEFAULT. Quest'ultima si riferisce all'area dell'intera immagine non coperta da altre regole AREA;
- COORDS consente di specificare le coordinate dell'area.
 - nel caso di RECTANGLE contiene quattro numeri nella forma X1,Y1,X2,Y2 che rappresentano le coordinate dei due punti che definiscono il rettangolo;
 - nel caso di CIRCLE contiene tre numeri nella forma X,Y,R in cui X,Y rappresentano il centro del cerchio ed R il suo raggio;
 - nel caso di POLY si ha un elenco di punti nel formato X1,Y1,X2,Y2,... che rappresentano i vertici del poligono.
- HREF indica il collegamento al documento da caricare nel caso si clicchi nell'area specificata;
- TITLE è il titolo da dare alla nuova finestra;
- NOHREF usato solo per NON collegare un area ad un documento.

7 FRAMES

In HTML (a partire dalla versione 3) è possibile dividere la pagina del browser in diverse sezioni e caricare in ognuna di esse diverse pagine. Ogni sezione viene definita FRAME e l'insieme delle sezioni FRAMESET. Si desidera costruire una pagina suddivisa in tre parti. Un frame orizzontale in alto della larghezza massima consentita, contenente il titolo della pagina e due frame verticali sotto quello del titolo: quello di sinistra contenente il menu dei capitoli, quello a destra contenente il capitolo da illustrare. In pratica si hanno queste pagine:

barra.htm

```
<HTML>
<HEAD>
  <TITLE>BARRA ORIZZONTALE DEL SITO</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H1><FONT COLOR="red" FACE="ARIAL">SITO DI ESEMPIO</FONT></H1>
  </CENTER>
</BODY>
</HTML>
```

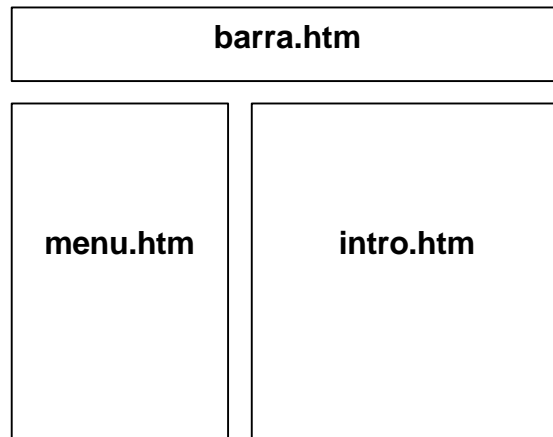
menu.htm

```
<HTML>
<HEAD>
  <TITLE>MENU VERTICALE DEL SITO</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <A HREF="intro.htm">Introduzione </A><BR>
    <A HREF="cap1.htm" >Capitolo primo </A><BR>
    <A HREF="cap2.htm" >Capitolo secondo</A><BR>
    <A HREF="cap3.htm" >Capitolo terzo </A><BR>
    <A HREF="cap4.htm" >Capitolo quarto </A><BR>
  </CENTER>
</BODY>
</HTML>
```

intro.htm

```
<HTML>
<HEAD>
  <TITLE>INTRODUZIONE</TITLE>
</HEAD>
<BODY>
  <H2>INTRODUZIONE</H2>
  <P ALIGN="left">Testo dell'introduzione</P>
</BODY>
</HTML>
```

La struttura dei frame vuole essere pressappoco la seguente:



Cliccando su uno dei collegamenti dei menu i diversi capitoli vanno a sostituire “intro.htm” o il capitolo visualizzato precedentemente nello stesso frame. Il file HTML che consente di fare questo è il seguente:

index.htm

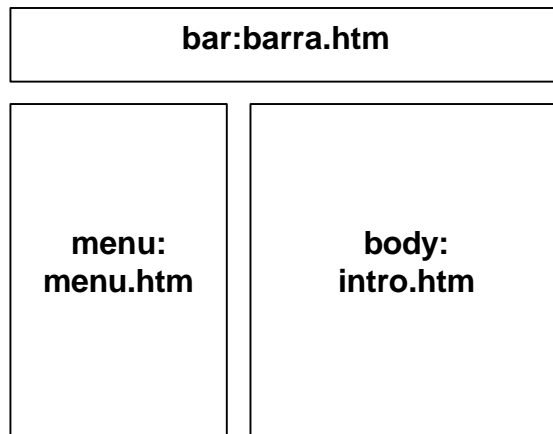
```
<HTML>
<HEAD></HEAD>

<FRAMESET ROWS="43,*" FRAMEBORDER="0" BORDER=0>
  <FRAME SRC="barra.htm" SCROLLING="no"
    NORESIZE=""
    NAME="bar"
    FRAMEBORDER="1"
    TARGET="_self"
    MARGINWIDTH="0"
    MARGINHEIGHT="0"
    BORDER="0">
  <FRAMESET COLS="169,*">
    <FRAME SRC="menu.htm" SCROLLING="no" NORESIZE="" NAME="menu"
      FRAMEBORDER=0 TARGET="body" MARGINWIDTH="0" MARGINHEIGHT="0"
      BORDER="0">
    <FRAME SRC="intro.htm" NAME="body" FRAMEBORDER=0 MARGINWIDTH="1"
      MARGINHEIGHT="1" SCROLLING="auto" BORDER="1" TARGET="_self">
  </FRAMESET>
</FRAMESET>
<NOFRAMES>
<BODY>
</BODY>
</NOFRAMES>
</FRAMESET>
```

La pagina è divisa principalmente in due frame orizzontali (ROWS). Il primo viene chiamato “bar”, è alto 43 pixel e vi viene caricata la pagina “barra.htm”; il secondo occupa tutto lo spazio rimanente (*):



Questo “secondo spazio” è a sua volta un FRAMESET ma, questa volta, verticale (COLS). IL primo frame è largo 169 pixel, si chiama menu e contiene la pagina “menu.htm”, il secondo occupa lo spazio rimanente, si chiama “body” e carica all’attivazione la pagina “intro.htm”.



Il frame “menu” ha come TARGET il frame “body”; ciò significa che ogni collegamento nella pagina contenuta nel frame “menu” se non specificato diversamente, caricherà le pagine correlate nel frame “body”. A prescindere da alcuni attributi dall’ovvio significato (BORDER, MARGINWIDTH, MARGINHEIGHT, FRAMEBORDER, NAME già spiegato) è importante sottolineare che NORESIZE elimina all’utente la possibilità di ridimensionare il frame cliccando sulla linea divisoria tra un frame ed un altro e spostarla; mentre l’attributo SCROLLING può assumere i valori NO, HORIZONTAL, VERTICAL, BOTH, AUTO a seconda delle barre di scorrimento che si desiderano visualizzare nella pagina. AUTO consente di farle visualizzare solo nel caso in cui il documento HTML contenuto nel frame ecceda le dimensioni del frame stesso.



All'interno del tag A di qualunque collegamento è possibile specificare l'attributo TARGET seguito dal nome del frame in cui si desidera visualizzare il documento da caricare. Esistono alcuni valori speciali come "_blank" che apre una nuova finestra e "_self" che carica nello stesso frame; oppure "_top" che carica nel documento principale della finestra. E' possibile, inoltre, specificare il comando BASE nella sezione HEAD anch'esso con l'attributo TARGET. Tutti i collegamenti il cui frame di destinazione non verrà specificato faranno riferimento a quello contenuto in questo attributo.

8 MODULI

L'HTML è un linguaggio definito "stateless"; cioè non è in grado di tenere in memoria alcun dato "storico" rispetto a documenti caricati precedentemente, collegamenti visitati, valori impostati, ecc. E' altresì vero che delle regole consentono di inviare ai Web server, o al limite ad altre pagine HTML, dei valori impostati dall'utente. Sebbene al momento non sapremmo come utilizzare tali valori nel caso vengano ricevuti da una nostra pagina (serve un linguaggio di script, come JavaScript o VBScript) sarà opportuno imparare ad inviarli proprio per consentirne una rielaborazione da parte di chi sarà in grado di manipolarli.

L'invio di dati a partire dall'HTML avviene per mezzo dei FORM. Un form è delimitato dal rispettivo tag iniziale e finale (/FORM). Al suo interno si possono definire diversi oggetti manipolabili dall'utente, come caselle di testo, checkbox, radiobutton, pulsanti, liste e caselle combinate. Facciamo un esempio. Il codice seguente definisce un modulo (a pagina seguente c'è la rappresentazione nel browser).

```
<form method="GET" ACTION="http://www.miosito.com/destinazione.htm">
  <p>Nome:   <input type="text" name="T1" size="20">           </p>
  <p>Cognome: <input type="text" name="T2" size="20">         </p>
  <p>Città:   <select size="1" name="D1">
                <option value="Roma">Roma   </option>
                <option value="MI" >Milano </option>
                <option value="TO" >Torino </option>
            </select>
  <p>Commento:<textarea rows="4"      name="S1" cols="58"></textarea> </p>
  <p>Maschio <input type="radio"      name="R1" value="V1"      > </p>
  <p>Femmina <input type="radio"      name="R1" value="V2"      > </p>
  <p>Sposato <input type="checkbox"      name="C1" value="ON"      > </p>
  <p>
    <input type="submit" name="B1" value="Invia" >
    <input type="reset" name="B2" value="Reimposta" > </p>
</form>
```

Gli attributi principali di un form sono:

- ACTION contiene l'URL o il collegamento relativo alla pagina a cui si vogliono mandare i dati. Solitamente è un URL di un CGI-BIN o di una pagina ASP, ma può anche essere una normale pagina HTML.
- METHOD che può essere GET o POST. Nel primo caso invia i dati impostati nel form (vedi oltre) accodandoli all'URL di destinazione specificato da ACTION, nel secondo li cripta in una intestazione http sempre inviandoli alla pagina di destinazione.

All'interno di un form si possono definire diversi oggetti, ognuno dei quali possiede un nome (attributo NAME) e, laddove non è possibile per l'utente scriverci, un valore (VALUE).

Il codice precedente (a parte gli allineamenti) corrisponde alla seguente pagina HTML:

Nome:	<input type="text"/>
Cognome:	<input type="text"/>
Città:	<input type="text" value="Roma"/>
Commento:	<div style="border: 1px solid black; height: 60px; width: 100%;"></div>
Maschio	<input type="radio"/>
Femmina	<input checked="" type="radio"/>
Sposato	<input type="checkbox"/>
	<input type="button" value="Invia"/> <input type="button" value="Reimposta"/>

9 FOGLI DI STILE

I CSS (Fogli di Stile a Cascata) consentono una formattazione strutturata. La definizione degli stili all'interno di una pagina HTML avviene secondo lo schema seguente.

```
<HTML>
<HEAD><TITLE>Formattazione con CSS</TITLE>
<STYLE TYPE="text/css">
  H1 {font-family: Arial, Helvetica; font-weight: bold}
  CITE {font-family: Courier}
</STYLE>
</HEAD>
<BODY>
...
</BODY>
```

Gli attributi principali di uno stile sono i seguenti.

color:	#rrggbb
display:	Inline/Block/List-item/none
font-family:	Arial, Helvetica, Courier, ...
font-size:	12pt/1in/small/larger/smaller
font-style:	normal/italic oblique
font-variant:	Normal/Small-caps
font-weight:	Normal/Bold/Lighter
white-space:	Normal/pre/nowrap
text-decoration:	overline/underline/line-through/blink
background-color:	#rrggbb
background-image:	URL immagine
text-align:	right/left/center/justify
text-indent:	rientro nella prima riga
...	...

La formattazione può essere inserita anche nei tag SPAN e DIV (ma anche in P e FONT) i quali possiedono (come gli altri tag) gli attributi ID e CLASS. Il primo(ID) identifica univocamente un'istanza di un tag da un'altra, cioè in un documento non possono esserci due ID uguali. Al contrario CLASS identifica una classe di tag, per cui è possibile (anzi auspicabile) che uno stesso identificativo di classe si ripeta in uno stesso documento.

Ad esempio se volessi formattare un tag DIV specifico dovrei farlo attraverso il suo ID:

```
DIV#NomeID {...}
```

Altrimenti, se decidessi di formattare con uno stile una classe di tag DIV dovrei procedere come qui di seguito:

```
DIV.NomeClasse {...}
```

Si possono creare file in cui vengono racchiusi i diversi stili. Il metodo per includere tali librerie di stili è il seguente (riga da inserire nel codice HTML nella sezione HEAD):

```
<LINK REL="stylesheet" TYPE="text/css" HREF="cambiacolorelink.css">
```

Immaginiamo di avere un foglio denominato `cambiacolorelink.css` il cui contenuto è il seguente:

`cambiacolorelink.css`

```
<STYLE>
  A      {color : blue ; }
  A:hover {color : red  ; }
</STYLE>
```

Ed un file HTML come il seguente nella stessa cartella:

`cambiacolorelink.htm`

```
<HTML>
<HEAD>
  <LINK REL="stylesheet" TYPE="text/css" HREF="cambiacolorelink.css">
</HEAD>
<BODY>

<H1>PROVA UTILIZZO DI UN CSS ESTERNO</H1>

<A HREF="http://www.pippo.it">Collegamento</A>

</BODY>
</HTML>
```

Per quanto contenuto nella pagina CSS passando con il mouse sopra il link, esso diventa rosso.