

Appunti di HTML

Il concetto di ipertesto

Un ipertesto è formato da un insieme di documenti, collegati tra loro tramite riferimenti ipertestuali, denominati link. Questi documenti possono essere costituiti da testo, immagini, suono, ecc. A differenza del normale testo, che deve essere letto sequenzialmente, i link permettono a chi *naviga* l'ipertesto di accedere alle sue componenti in un ordine non rigido, ma seguendo un percorso a piacere tra quelli possibili al suo interno.

Anche se i documenti che si trovano sul Web hanno una natura ipertestuale, l'uso di questo termine non è necessariamente collegato all'accessibilità di un documento sul Web. Un ipertesto può infatti essere memorizzato su un supporto (un disco fisso, un floppy) a cui si accede localmente.

I documenti HTML permettono di scrivere gli ipertesti per il Web. Si tratta di documenti memorizzati sui server Web che vengono visualizzati dai client Web, ovvero dai browser (Netscape Navigator, Microsoft Internet Explorer, Opera, ...).

Richiami su Internet

Le pagine HTML sono trasferite attraverso la rete Internet usando un protocollo particolare **HTTP** (**H**yper **T**ext **T**ransfer **P**rotocol) costruito al di sopra della coppia di protocolli **TCP/IP** che costituiscono la base di Internet. Sono localizzate mediante un ben preciso sistema di indirizzamento noto con l'acronimo **URL** (**U**niform **R**esource **L**ocator).

Internet è una rete geografica mondiale che collega le reti di diversi paesi, con milioni di nodi collegati. È organizzata a domini, secondo un ordinamento gerarchico.

Es.

disi.unige.it dominio Dip. di Informatica (**disi**)
nel dominio Univ. di Genova (**unige**)
nel dominio Italia (**it**)

doc.ic.ac.uk Department of Computing (doc)
Imperial College (ic)
rete academic (ac)
dominio United Kingdom (uk)

li.univ-paris13.fr Laboratoire d'Informatique
Université de Paris 13
France

cs.ubc.ca Computer Science
University of British Columbia
Canada

ics.mit.edu

Laboratory for Computer Science
MIT
rete education (edu)
il dominio USA è implicito

Internet fornisce vari **servizi** tra cui ricordiamo

- trasferimento dati
- posta elettronica
- collegamento remoto
- liste di discussione
- basi di dati in rete
- motori di ricerca
- navigazione ipertestuale

Caratteristiche di HTML

HTML (**H**yper **T**ext **M**arkup **L**anguage) è un linguaggio per la costruzione di ipertesti. È costituito da un insieme di comandi che permettono di definire la struttura e l'aspetto di un ipertesto, di inserire oggetti (immagini, tabelle, moduli) e di creare link ipertestuali interni al documento oppure ad altri documenti.

Un documento HTML è costituito da un semplice testo, che in quanto tale può essere visualizzato attraverso qualsiasi editor di testo. Per vedere il documento HTML come ipertesto è necessaria invece un'applicazione (il browser) che interpreta i comandi contenuti nel testo producendo la visualizzazione definita rispetto a questi comandi.

Alla base di HTML esiste una standardizzazione che garantisce che ad ogni comando (testuale) corrisponda una certa visualizzazione da parte del browser.

Per creare un file HTML è possibile usare due strumenti:

- un editor di testo (ad esempio WordPad)
- un'applicazione per la preparazione di documenti HTML (Microsoft Frontpage, Netscape Composer, Macromedia Dreamweaver) del tipo WYSIWYG ("what you see is what you get")

Mentre nel primo caso occorre conoscere HTML e scrivere i comandi tramite l'editor di testo, nel secondo caso la conoscenza di HTML non è necessaria, in quanto l'applicazione permette all'utente di comporre il documento a partire dall'aspetto risultante: i comandi vengono aggiunti man mano dall'editor a seconda degli elementi (testo, grafica, ecc) inseriti dall'utente.

Il file HTML creato tramite l'editor deve essere salvato come documento testuale con estensione .html o .htm; in caso contrario, il documento verrà visualizzato come testo e non come documento HTML.

Comandi HTML

I comandi HTML sono detti **tag** (o marcatori) e devono sempre essere scritti all'interno di parentesi uncinate (<NOME-TAG>). I tag – tranne alcune eccezioni – devono essere aperti e chiusi, e la loro influenza si esercita sul testo contenuto al loro interno (tra l'apertura e la chiusura del tag, che avviene facendo precedere il comando dal carattere “/”).

Inoltre, i tag possono avere degli attributi che assumono dei valori (scritti tra virgolette). Gli attributi (scelti tra un insieme predefinito) sono scritti di seguito al nome del comando, all'interno delle parentesi.

Struttura di un documento

Un documento HTML è interamente contenuto all'interno del tag <HTML> e si compone di due parti essenziali (che vedremo in seguito in maggiore dettaglio)

- head (intestazione)
- body (corpo del documento)

L'intestazione (head) può contenere il titolo, oltre ad alcune informazioni generali (tipo di editor usato, contenuto del documento, ecc.).

Il corpo del documento (body) contiene il documento vero e proprio, che verrà visualizzato nella finestra del browser.

All'interno del body è possibile inserire

- testo (e la relativa formattazione)
- liste
- tabelle
- immagini
- collegamenti
- moduli

Layout della pagina

È possibile rendere più gradevole una pagina inserendo nel body degli attributi che modificano l'aspetto del testo e lo sfondo della pagina. Ad esempio, si può colorare lo sfondo della pagina, modificare il colore del testo, e l'aspetto dei link.

Testo e formattazione

Tutto ciò che viene scritto dentro il corpo del documento, tranne naturalmente i comandi e i loro attributi, viene visualizzato dal browser come testo. Per inserire la formattazione (a capo, corsivo, grassetto, centratura del testo, ecc.) si utilizzano i comandi appositi.

Annidamento dei comandi

A questo punto è opportuno notare come i comandi di HTML possano essere *annidati*, cioè “in scatolati” uno nell'altro. Così facendo si possono ottenere effetti sofisticati

- un'immagine all'interno di un link crea un collegamento associato all'immagine
- i punti di una lista possono contenere testo formattato, immagini, altre liste

I tag del linguaggio

Vediamo ora più in dettaglio i tag che HTML mette a disposizione del progettista di pagine per il Web.

Come abbiamo già visto, i tag hanno una forma sintattica particolare

```
<NOME-TAG>  
    informazioni  
</NOME-TAG >
```

Le informazioni tra <NOME-TAG> e </NOME-TAG> vengono visualizzate secondo quanto specificato dal tag stesso.

- I tag hanno nomi mnemonici.
- Molti tag hanno degli **attributi** che servono per fornire ulteriori caratteristiche alle porzioni del documento. I valori degli attributi dovrebbero essere scritti tra gli virgolette, l'ordine con cui si scrivono è irrilevante.
- Per aumentare la leggibilità dei file HTML si possono *indentare* i tag.
- I nomi dei tag possono essere scritti usando le lettere maiuscole o minuscole (è consigliabile essere consistenti per facilitare le correzioni). Fino a qualche anno fa, i tag venivano scritti con i caratteri maiuscoli per aumentare la leggibilità dei documenti HTML. Oggi si usano anche i caratteri minuscoli, principalmente per compatibilità con le estensioni più recenti dei linguaggi di markup (es. XML).
- HTML non è “sensibile” ai *blank* (spazi) e alle linee vuote, ovvero indipendentemente dal numero di spazi e di linee lasciati vuoti, il browser visualizzerà sempre un solo spazio.

Tutti i documenti iniziano con il tag <HTML> e terminano con </HTML>

Un documento HTML è costituito da due parti

```
<HTML>  
    <HEAD>  
        descrizione delle caratteristiche del documento 1  
    </HEAD>  
    <BODY>  
        documento vero e proprio 2  
    </BODY>  
</HTML>
```

HEAD

L'elemento fondamentale nella parte di head è il titolo del documento che verrà visualizzato come titolo nella finestra del browser.

Attn. poiché il titolo viene usato anche per costruire gli indici automatici usati dai motori di ricerca è importante che esso sia significativo.

Es.

```
<HEAD>
  <TITLE>Scienze Ambientali e Naturali, Università di Genova</TITLE>
</HEAD>
```

Oss. Il contenuto del tag <TITLE> viene scritto automaticamente nell'elenco dei siti preferiti quando si salvano usando il browser.

Esistono poi altri tag, detti META tag, che permettono di associare delle informazioni al documento HTML. Le "keywords" (parole chiave), per esempio, sono i termini che descrivono il contenuto di una pagina Web e sono gli stessi termini che vengono poi usati dai motori di ricerca per l'indicizzazione.

Es.

```
<META name="keywords"
  content="Scienze Ambientali, biologia marina, Genova, STAN">
```

Altri META tag riguardano l'autore della pagina Web o il nome dell'editor con cui il documento HTML è stato generato.

Es.

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

Mettendo tutto insieme avremo per esempio:

```
<HTML>
<HEAD>
  <TITLE>Scienze Ambientali e Naturali, Università di Genova</TITLE>
  <META name="keywords"
    content="Scienze Ambientali, biologia marina, Genova, STAN">
  <META name="author" content="Nome Cognome">
  <META name="generator" content="WordPad">
</HEAD>
.....
.....
.....
</HTML>
```

BODY

Contiene il documento (pagina) vero e proprio, cioè il contenuto che viene visualizzato nel browser.

Documento di testo

Il testo può essere inserito liberamente nella parte BODY e verrà visualizzato secondo le direttive di formattazione.

Titoli

`<H1> titolo1 </H1>`

`<H2> titolo2 </H2>`

...

`<H6> titolo6 </H6>`

permettono di indicare quali parti di testo vengono usate come titoli.

Oss. `<H1>`, ..., `<H6>` sono usati anche per controllare le dimensioni dei caratteri: `<H1>` corrisponde a caratteri grandi, ..., `<H6>` a caratteri piccoli.

Comandi per andare a capo

`
 {a capo}`

`<P> {a capo e inizio nuovo paragrafo}`

Es.

`<BODY>`

`<H1> Scienze Ambientali e Naturali </H1>`

`<P>Questo documento descrive il corso di laurea.`

`
Il documento è diviso in più capitoli.`

`<P>`

`<H2> Durata del corso </H2>`

....

`</BODY>`

Oss. la grandezza dei caratteri in fase di visualizzazione dipende dalle opzioni selezionate nel browser.

Oss. per ottenere i caratteri speciali (non ASCII standard) si devono usare dei comandi speciali.

Es.

`à -> à`

`è -> è`

`é -> é`

`ä -> ä`

....

Colore

Il comando <BODY> ha degli attributi che permettono di stabilire il colore dello sfondo e il colore del testo.

```
<BODY      bgcolor="colore sfondo"
          text="colore testo"
          background="pathname del file con l'immagine per lo sfondo"
          link="colore link da visitare"
          vlink="colore link visitati"
>
```

Il colore può essere specificato con

- parole chiave: red, yellow, ...
- codice esadecimale: esprimendo la sua codifica RGB

Oss. per default si ha sfondo grigio, testo nero, link da visitare blu, link visitati rossi.

Dimensione dei caratteri

Si usano i tag <BASEFONT> e con l'attributo **size** che permette di definire le dimensioni dei caratteri da 1 (piccolo) a 7 (grande)

Es.

```
<BASEFONT size="4">
```

```
<FONT size="3"> testo a dimensione 3 </FONT>
... si ritorna alla dimensione di basefont
```

```
<FONT size="+1"> dim+1 del precedente </FONT>
... si ritorna al precedente
```

Il comando FONT ha in realtà tre attributi

- **size** *{per cambiare la dimensione dei caratteri}*
- **color** *{per cambiare il colore dei caratteri}*
- **face** *{per cambiare il tipo di font}*

Oss. se si sceglie un font poco comune può darsi che il calcolatore sul quale verrà visualizzato il vostro documento non lo abbia. In questo caso, l'effetto non sarà lo stesso che avete progettato perché il browser userà i suoi font predefiniti.

Es.

```
<FONT size="5" color="red" font="Courier">
```

```
Questo testo viene visualizzato in rosso, ha dimensione 5 e font Courier
</FONT>
```

...

Si ritorna a colore, dimensione, font precedenti

Formato dei caratteri

Esistono vari modi per cambiare il formato dei caratteri.

Stili fisici

` testo ` *{testo in grassetto}*
`<I> testo </I>` *{testo in corsivo}*
`<TT> testo </TT>` *{testo typewriter}*

Stili logici

` testo ` *{enfasi (di solito corsivo)}*
`<CITE> testo </CITE>` *{citazioni (di solito in corsivo)}*
`<CODE> testo </CODE>` *{per codice di computer}*
` testo ` *{forte enfasi (grassetto)}*

Allineamento

Il comando `<P>` ha un attributo **align** per l'allineamento del testo nel paragrafo.

`<P align="left"> testo allineato a sinistra </P>`
`<P align="right"> testo allineato a destra </P>`
`<P align="center"> testo allineato al centro </P>`

Oss. la centratura del testo si può ottenere anche con il comando `<CENTER>`.

`<CENTER> testo da centrare </CENTER>`

Indici e pedici

a`₁` produce a₁
b`²` produce b²

Es.

```
<HTML>
<BODY bgcolor="white" text="red">
<BASEFONT size="4">
<H1>
<CENTER> Scienze Ambientali e Naturali</CENTER>
</H1>
<P align="right">
<FONT size="+1">Laurea triennale nuovo ordinamento</FONT>
</P>
Presentazione del <EM> Corso di laurea </EM>
<P>
<H2> Durata </H2>
La durata &grave; di <STRONG> 3 </STRONG> anni. <BR>
Il corso prevede <STRONG> 180 </STRONG> crediti.
<BR>
<FONT size="-1" color="blue">
La laurea specialistica non &grave; ancora attivata.
</FONT>
<BR>
<ADDRESS>
pagina creata da X Y
</ADDRESS>
</BODY>
</HTML>
```

Liste di elementi

Può essere utile poter costruire liste di elementi come quella seguente

Il corso di laurea ha i seguenti indirizzi indirizzi

- aaaaaa
- bbbbbb

HTML fornisce vari comandi per creare le liste.

1) Liste non numerate: e

Es.

Il corso di laurea ha i seguenti indirizzi indirizzi

```
<UL>
<LI> aaaaa </LI>
<LI> bbbbbb </LI>
</UL>
```

 ha l'attributo **type** per modificare la forma del “pallino” nella lista.

```
<UL type="disc">
<UL type="circle">
<UL type="square">
```

2) Liste numerate: e

Es.

Il corso di laurea ha i seguenti indirizzi indirizzi

```
<OL>
<LI> aaaaa </LI>
<LI> bbbbbb </LI>
</OL>
```

Produce:

Il corso di laurea ha i seguenti indirizzi indirizzi

1. aaaaa
2. bbbbbb

 ha gli attributi **type** e **start**

```
type="1" | "A" | "a" | "I" | "i"
start="numero"
```

Es.

Il corso di laurea ha i seguenti indirizzi indirizzi

```
<OL type="i" start="3">
<LI> aaaaa</LI>
<LI> bbbbb</LI>
</OL>
```

Produce:

Il corso di laurea ha i seguenti indirizzi indirizzi

```
iii  aaaaa
iv   bbbbb
```

3) Liste di definizioni: <DL>, <DT> e <DD>

Per scrivere delle definizioni si può usare il tag <DL>

Es.

Alcune definizioni:

```
<DL>
<DT> HTML
    <DD> ling. per scrivere pagine Web</DD>
</DT>
<DT> Internet
    <DD> rete mondiale di calcolatori</DD>
</DT>
</DL>
```

Produce:

Alcune definizioni:

```
HTML
    ling. per scrivere pagine Web
Internet
    rete mondiale di calcolatori
```

Si possono *annidare* liste, non necessariamente dello stesso tipo

Es.

```
<UL>
<LI> primo elemento
<OL>
    <LI> primo sottolista</LI>
    <LI> secondo sottolista</LI>
</OL>
</LI>
<LI> secondo elemento
    ...
</UL>
```

Produce:

- primo elemento
 1. primo sottolista
 2. secondo sottolista
- secondo elemento
- ...

Testo preformattato

Con il tag `<PRE>` è possibile dare disposizioni affinché una parte di testo non venga formattata.

`<PRE>`

```
questo testo sarà lasciato           come scritto
con gli stessi spazi e le stesse righe
che ci sono                           qui
```

`</PRE>`

Citazioni

Per inserire citazioni nel testo si usa il comando

`<BLOCKQUOTE>` *citazione* `</BLOCKQUOTE>`

Linee orizzontali

Per separare parti di testo si può usare il comando `<HR>` che produce linee orizzontali. Questo comando ha tre attributi:

```
<HR size="numero"           {spessore della linea in pixel}
width="numero" | "numero%"  {lunghezza in pixel o in percentuale}
align="left" | "right">
```

Testo lampeggiante

Testo, in qualunque forma, che lampeggia in fase di visualizzazione

`<BLINK>` *questo testo lampeggia* `</BLINK>`

Oss. questo tag funziona solo su Netscape ed è sconsigliato.

Testo scorrevole

Testo, in qualunque forma, che scorre in fase di visualizzazione

`<MARQUEE>` *questo testo scorre* `</MARQUEE>`

Oss. questo tag funziona solo su Explorer ed è sconsigliato.

Commenti

Può essere utile mettere dei commenti nel documento che non sono visualizzati dal browser
<!-- questo è un commento -->

Immagini

Fino ad ora abbiamo visto come si può formattare il testo; HTML permette anche di inserire delle immagini usando il comando .

- l'immagine deve essere su un file a parte; i browser supportano formati quali GIF, JPEG, PNG, BITMAP
- il comando IMG ha vari attributi

```
<IMG
  src="pathname o URL dell'immagine"
  align="left" | "right" | "center"
  alt="text"           {testo alternativo all'immagine}
  border="numero"     {larghezza in pixel del bordo}
  height="numero"     {altezza in pixel dell'immagine}
  width="numero"      {larghezza in pixel dell'immagine}
  hspace="numero"     {spazio in pixel a destra e sinistra dell'immagine}
  vspace="numero"     {spazio in pixel sopra e sotto l'immagine}
  usemap="nome"       {permette di creare image map}
>
```

Es.

```
<IMG src="images/foto.gif" align="left" width="100" height="120" vspace="20">
```

Oss. gli attributi **height** e **width** *istruiscono* il browser sullo spazio che deve essere lasciato per la visualizzazione dell'immagine. Il caricamento delle pagine diventa quindi più veloce perché il browser può cominciare a scrivere il testo successivo prima ancora di aver caricato completamente un'immagine (tanto *sa* quanto spazio occuperà l'immagine).

Tuttavia, l'uso di questi attributi è sconsigliato per cambiare le dimensioni delle immagini perché queste verranno visualizzate in forma distorta o sgranata. Se si vuole ingrandire o rimpicciolire un'immagine è conveniente lavorare sul file dell'immagine, usando un programma di grafica, piuttosto che forzarne le dimensioni con gli attributi **height** e **width**.

Oss. è necessario scrivere esplicitamente l'estensione del file dell'immagine (GIF → .gif
JPEG → .jpg BITMAP → .bmp – ma non funziona su Netscape Navigator).

Oss. il testo alternativo all'immagine – valore dell'attributo **alt** – è importante per garantire l'accessibilità delle vostre pagine anche ai ciechi che usano i browser vocali. Quando un browser vocale incontra un'immagine legge il valore dell'attributo **alt** che dovrebbe contenere una descrizione dell'immagine stessa.

Suoni

È possibile associare dei suoni alla presentazione di una pagina HTML usando il comando **<BGSOUND>** (che funziona solo in Explorer).

Anche i suoni devono essere memorizzati su file a parte; esistono vari formati (AU, WAV, AIFF, MIDI).

```
<BGSOUND
    src="pathname del suono" | "URL del suono"
    loop="numero" | "infinite"
>
```

L'attributo **loop** serve per stabilire quante volte si deve ripetere il suono. Se assume valore **infinite** si ripete il suono fino a quando non si esce dalla pagina.

Es.

```
<BGSOUND src="suoni/pippo.wav" loop="3">
```

Legami ipertestuali

Fino ad ora abbiamo visto come creare singole pagine HTML. Per avere un ipertesto sono necessari dei collegamenti (link) tra le pagine. Prima di vedere come si definisce un collegamento è necessario capire come è formato un indirizzo URL per il Web.

Un URL è costituito da varie parti

protocollo:// indirizzo internet /pathname#label

Il **protocollo** descrive il tipo di collegamento da realizzare

http:	<i>{per file remoti da trasferire con http}</i>
ftp:	<i>{attiva trasferimento dati con ftp}</i>
news:	<i>{attiva collegamento a server news}</i>
telnet:	<i>{attiva sessione di collegamento remoto}</i>
mailto:	<i>{attiva spedizione di mail}</i>

Il protocollo più importante per i legami ipertestuali è http

L'**indirizzo internet** è l'indirizzo del calcolatore remoto sul quale risiedono i documenti (server Web).

Il **pathname** indica il nome del file che si vuole scaricare, con il relativo percorso.

#label serve per indicare una porzione all'interno di un documento HTML (come vedremo più avanti).

Es.

1) <http://www.disi.unige.it/index.html> *{file index.html sul server www.disi.unige.it}*

Oss. www in questo caso è il nome di un calcolatore nel dominio disi.unige.it
Di solito alle macchine server web viene assegnato il nome www.

2) <mailto:ribaudo@disi.unige.it>

{attiva il programma di invio mail}

3) <ftp://ftp.disi.unige.it/didattica/materiale.pdf>

{attiva trasferimento del file materiale.pdf che si trova nella cartella didattica dal server ftp.disi.unige.it}

Oss. anche in questo caso ftp è il nome del calcolatore server per ftp

4) <telnet:elios.disi.unige.it>

{attiva la finestra di interazione remota}

Creazione del legame ipertestuale

Bisogna considerare

- il testo (immagine) che si comporterà come **hotword** o **bottone**
- il **servizio** da attivare

ANCHOR <A>

Il tag <A> permette di creare un link. Il suo attributo più importante è **href** cui si associa l'URL del documento da aprire

`testo o immagine`

Es.

`vai al sito del DISI `

`spedisci una mail a Ribaldo `

` `

Es.

```
<HTML>
<HEAD>
  <TITLE> ACME </TITLE>
</HEAD>

<BODY>
  <IMG src="images/logo.jpg">
  <H1> <B> ACME Società per Azioni </B> </H1>
  <HR>
  <H3>ACME è una società leader nella produzione di
  componenti per calcolatori</H3>
  <P>
  La sua produzione attuale comprende:
  <UL>
  <LI>componenti di
  <A href="http://www.acme.com/prod/mem.html">memoria </A></LI>
  <LI>componenti per
  <A href="http://www.acme.com/prod/col.html"> collegamento in rete </A></LI>
  </UL>
  <P>
  Il <A href="ftp://www.acme.com/cat.zip">catalogo</A> può essere trasferito via
  rete.
  <P>
  ACME collabora con <A href="http://www.unige.it">l'Università di Genova</A>
  <P>
  La sede:
  <A href="http://www.acme.com/sede.html"><IMG src="images/sede.jpg"> </A>
  <P>

  <HR>
  Pagina a cura di <A href="mailto:maria@www.acme.com">Maria Rossi</A>

</BODY>
</HTML>
```


Etichette (label) e salti a sezioni

Con il comando <A> è anche possibile etichettare un punto di un documento con un nome (una **label**) e quindi saltare direttamente a quel punto grazie ad un link.

```
<A name="nome label"> testo a cui saltare </A>
```

Es. nel documento manuale.html possiamo avere

```
....  
<H1><A name="cap1">Capitolo 1</A></H1>  
...
```

In un altro punto dello stesso file manuale.html si avrà

```
<A href="#cap1">vai al Capitolo 1</A>
```

In un altro file sullo stesso server si avrà

```
<A href="manuale.html#cap1">vai al Capitolo 1</A>
```

In un file su un altro server si avrà

```
<A href="http://www.manuali.it/manuale.html#cap1">vai al Capitolo 1</A>
```

Link a immagini, suoni e animazioni esterne

Si possono visualizzare immagini, suoni, animazioni specificando nell'URL il nome di un file corrispondente. I principali formati riconosciuti sono

- per le immagini: GIF, JPEG, PNG, BITMAP
- per i suoni: AIFF, AU, WAV
- per le animazioni: MOV (QuickTime), AVI, MPEG

Funzioni più sofisticate

Vedremo ora delle funzioni più sofisticate per

- creare delle tabelle all'interno di un documento (ogni elemento di una tabella può contenere testo, immagini, suoni, link, ...)
- rendere attive delle porzioni di immagini (image map)
- suddividere una pagina in più parti, ognuna delle quali contiene una diversa pagina HTML gestita indipendentemente dalle altre (frame)
- creare moduli in rete che possono essere spediti e gestiti da altre applicazioni (form)

Tabelle

Per creare una tabella si usa il comando

<TABLE> *descrizione della tabella* **</TABLE>**

<TABLE> ha vari attributi che servono per stabilire le caratteristiche della tabella

<TABLE

border="numero" *{larghezza in pixel dei bordi}*
align="left" | "right" | "center" *{allineamento della tabella nella pagina}*
cellspacing="numero" *{spazio in pixel tra le celle}*
cellpadding="numero" *{spazio tra bordo e contenuto delle celle}*
width="numero" | "numero%" *{larghezza della tabella in pixel o in %}*

>

Comandi per la formattazione della tabella

<TR> *nuova riga* **</TR>**

<TR

align="left" | "right" | "center" *{allineamento nella tabella}*
valign="top" | "middle" | "bottom" *{allineamento verticale del testo}*
bgcolor="colore dello sfondo della riga"

>

<TD> *nuova cella in una riga* **</TD>**

<TH> *titolo della tabella (grassetto e centrato)* **</TH>**

<CAPTION> *titolo della tabella* **</CAPTION>**

Es.

```
<TABLE border="2">
<CAPTION align="bottom"> una tabella </CAPTION>
<TR>
  <TH> prima </TH>
  <TH> seconda </TH>
  <TH> terza </TH>
</TR>
<TR align="left">
  <TD> 1 </TD>
  <TD> 2 </TD>
  <TD> 3 </TD>
</TR>
<TR align="center">
  <TD> a </TD>
  <TD> b </TD>
  <TD> c </TD>
</TR>
<TR align="right">
  <TD> i </TD>
  <TD> ii </TD>
  <TD> iii </TD>
</TR>
</TABLE>
```

Produce:

prima	seconda	terza
1	2	3
a	b	c
i	ii	iii

una tabella

I comandi <TD> e <TH> hanno vari attributi

<TD (TH)

width="numero" | "numero%" {larghezza della cella in pixel o in %}
colspan="numero" {numero di colonne su cui si distribuisce una cella}
rowspan="numero" {numero di righe su cui si distribuisce una cella }
nowrap {non andare a capo nelle celle }

>

Es.

```
<CENTER>
<TABLE border="1">
<CAPTION align="bottom"> esempio complesso </CAPTION>
<TR> <TH rowspan="2" colspan="2"> </TH>
      <TH colspan="2"> Media </TH>
</TR>
<TR> <TH> Altezza </TH>
      <TH> Peso </TH>
</TR>
<TR> <TH rowspan="2" valign="top" > Genere </TH>
      <TH> maschi </TH>
      <TD> 1.75 </TD>
      <TD> 70 </TD>
</TR>
<TR> <TH> femmine </TH>
      <TD> 1.65 </TD>
      <TD> 58 </TD>
</TR>
</TABLE>
</CENTER>
```

Produce:

		Media	
		Altezza	Peso
Genere	maschi	1.75	70
	femmine	1.65	58

esempio complesso

Oss. le tabelle possono essere annidate (una cella può a sua volta contenere una tabella), se non sono annidate allora </TD> e </TH> possono essere omissi.

Image map

In un documento può essere utile rendere **attive** delle porzioni di immagini, ossia realizzare dei link ipertestuali con delle parti di immagini.

Per creare una image map è necessario avere

1. il file dell'immagine
2. una specifica di quali parti sono attive e di quali azioni devono essere svolte quando si clicca su di esse

L'immagine viene caricata con il comando usando l'attributo **usemap**

```
<IMG src="nome del file dell'immagine" usemap="#mappatura">
```

Per scrivere la specifica della mappatura si usano i tag <MAP> e <AREA>; si scriverà

```
<MAP name="mappatura"> specifica della mappatura </MAP>
```

Per la specifica della mappatura si usa il tag <AREA>; si specificano delle aree dell'immagine e ad ogni area si associa l'URL corrispondente. Il tag <AREA> ha vari attributi

```
<AREA
  shape="rect" | "circle" | "polygon" | "default"      {forma dell'area}
  coords="x1,y1,x2,y2" | {rect: coord. del vertice in alto a sx e di quello in basso a dx}
                    "x, y, r" | {circle: coordinate del centro e del raggio}
                    "x1,y1, x2,y2, x3,y3, ..." {polygon: coordinate dei vertici}
  href="URL"      {URL associato all'area}
  nohref          {per le aree non attive}
>
```

Oss. come si possono scoprire le coordinate?

Si può leggere il file dell'immagine con un programma di grafica e, muovendosi con il mouse, leggere le coordinate visualizzate nella barra di stato del programma.

Oss. l'origine delle coordinate è in alto a sinistra sullo schermo.

Es.

```
<MAP name=" mappatura ">
  <AREA shape="rect" coords="0,0,90,90" href="http://www.unige.it/index.html">
  <AREA shape="circle" coords="200,200,10" href="mailto:ribaudo@disi.unige.it">
  <AREA shape="circle" coords="400,100,5" href="http://www.google.com">
  <AREA shape="default" nohref>
</MAP>
```

Oss. il valore dell'attributo **name** nel tag <MAP> e il valore dell'attributo **usemap** nel tag devono coincidere (escludendo il simbolo #).

Oss. l'area di tipo *default non* funziona in Explorer.

Frame

In molti casi può essere utile dividere un documento HTML in più parti, ognuna delle quali può a sua volta contenere un documento HTML gestibile in modo separato. Per fare questo si devono usare i comandi **<FRAMESET>** e **<FRAME>**.

Si deve creare un documento HTML, di solito chiamato index.html che

- non ha il tag **<BODY>** (**</BODY>**) che è rimpiazzato dal tag **<FRAMESET>** (**</FRAMESET>**)
- ha una parte finale **<NOFRAMES>** ... **</NOFRAMES>** che viene visualizzata nel caso in cui il browser non supporti i frame (quasi tutti i browser odierni supportano i frame)

Il comando **<FRAMESET>** permette di partizionare un documento attraverso una suddivisione in righe e colonne

<FRAMESET

```
rows= "numero1, ..., numeroN" |           {altezza in pixel delle righe}
      " numero1%, ..., numeroN %" |       {altezza in % (somma<100%!!)}
      *                                   {indica il resto della pagina}
cols="numero1, ..., numeroN" | " numero1%, ..., numeroN %" | *
frameborder="yes" | "no"                 {per i bordi delle varie righe e colonne}
```

>

Oss. ogni frameset può usare solo uno tra i due attributi rows o cols

Es.

```
<FRAMESET rows="50,50,80"> ..... </FRAMESET>
<FRAMESET rows="20%,50%,30%"> ..... </FRAMESET>
<FRAMESET rows="30,*,20,20"> ..... </FRAMESET>
<FRAMESET rows="2*,*"> ...      {la prima riga è il doppio della seconda}
```

Il comando **<FRAME>** si usa per indicare i frame creati con **<FRAMESET>**

<FRAME

```
src="URL"                                {URL associato al frame}
name="nome della finestra"               {usato per la navigazione}
scrolling="yes" | "no" | "auto"          {scrollbar associate al frame}
noresize
marginwidth="numero"                     {larghezza dei margini in pixel}
marginheight="numero"                    {altezza dei margini in pixel}
```

>

Es.

```
<FRAME src="esempio.html" name="finestra1" scrolling="yes" noresize>  
<FRAME src="http://www.disi.unige.it" name="finestra2" scrolling="auto">
```

Es. pagina divisa orizzontalmente in due frame

```
<HTML>  
<HEAD>  
<TITLE> prova di frame </TITLE>  
</HEAD>  
  
<FRAMESET rows="40%,60%">  
    <FRAME src="esempio.html" name="finestra1" scrolling="yes" noresize>  
    <FRAME src="http://www.disi.unige.it" name="finestra2" scrolling="auto">  
</FRAMESET>  
  
<NOFRAMES>Il tuo browser non supporta i frame</NOFRAMES>  
</HTML>
```

Es. pagina divisa in due righe: la prima contiene tre frame, la seconda contiene due frame

```
<HTML>  
<HEAD>  
<TITLE> prova di frame </TITLE>  
</HEAD>  
  
<FRAMESET rows="50%,50%">  
<FRAMESET cols="*,2*,*">  
    <FRAME src="a1.html" name="finestra1" scrolling="yes" noresize>  
    <FRAME src="http://www.disi.unige.it/index.html" name="disi">  
    <FRAME src="a2.html" name="finestra2" noresize>  
</FRAMESET>  
  
<FRAMESET cols="50%,50%">  
    <FRAME src="a3.html" name="finestra3" scrolling="no">  
    <FRAME src="http://www.mfn.unige.it/index.html" name="mfn">  
</FRAMESET>  
  
</FRAMESET>  
  
<NOFRAMES> Mi spiace </NOFRAMES>  
</HTML>
```

Se da un link in una pagina HTML si passa ad un altro documento, questo per default viene aperto nello stesso frame.

Es. se nel documento a1.html inseriamo un link, il file corrispondente verrà aperto nella stessa finestra in cui si trova a1.html.

Si può scegliere in quale finestra aprire un documento usando i **nomi** delle finestre e l'attributo **target** all'interno di <A>.

Es. in a1.html aperto nella finestra1 possiamo scrivere

```
<A target="finestra2" href="http://www.lastampa.it">La Stampa</A>
```

e il sito de La Stampa verrà visualizzato nella finestra di nome finestra2.

In **target** si possono usare alcuni nomi speciali per aprire i documenti

- target="_self" *{la finestra stessa: default}*
- target="_top" *{nella finestra intera}*
- target="_blank" *{in una nuova finestra}*
- target="_parent" *{nel frameset in cui il frame è contenuto}*

Form (moduli)

In alcuni documenti HTML può essere utile creare dei moduli (form) che possono essere riempiti da chi consulta le pagine stesse.

Le informazioni inserite possono poi essere spedite per e-mail oppure inviate ad opportuni programmi per elaborazioni successive.

Si deve usare il comando **<FORM>** *descrizione del modulo* **</FORM>**

```
<FORM action="mailto:indirizzo internet" method="post">
```

qui la descrizione del modulo

```
    <INPUT type="submit">
    <INPUT type="reset">
</FORM>
```

I valori dell'attributo **type**, submit e reset, servono per creare due pulsanti:

- il pulsante submit, quando cliccato, spedisce i dati inseriti nel form
- il pulsante reset, quando cliccato, cancella i dati inseriti nel form

All'interno del modulo è necessario creare degli spazi per l'input dell'utente.

Ogni spazio deve avere un nome che viene usato nella mail creata dal form per spedire le informazioni. Gli spazi per l'input possono essere costituiti da

- testo libero
- check box
- radio box
- menu

Il comando **<TEXTAREA>** crea una zona per il testo libero

```
<TEXTAREA
  name="nome"      {nome usato nella risposta}
  rows="numero"   {numero di righe}
  cols="numero"   {numero di colonne}
>
testo di default
</TEXTAREA>
```

Es.

```
<FORM>
....
<P> Dimmi cosa ne pensi di questa pagina:
<TEXTAREA name="valutazione" rows="3" cols="40">
capolavoro!!!
</TEXTAREA>

<INPUT type="submit">
<INPUT type="reset">
</FORM>
...
```

Il comando **<INPUT>** permette di creare altri tipi di oggetti all'interno di un modulo. Il suo attributo **type** può assumere valori diversi.

```
<INPUT type = "text" |      {input di testo}
              "radio" |     {crea radio box}
              "checkbox" |    {crea check box}
              "password" |  {spazio per la password}
              "hidden" |   {campo nascosto}
              "submit" |   {pulsante di invio}
              "reset" |    {pulsante di cancellazione}
              "button" |   {pulsante generico}
>
```

Il tag **< INPUT>** ha anche altri attributi che dipendono dal valore dell'attributo **type**

1) **type="text"** *{crea un campo di testo di una sola riga}*

```
<INPUT type="text"
      name="nome del campo di input"
      size="numero"            {dimensione del campo}
      maxlength="numero"      {numero massimo di caratteri per il campo}
      default="testo"          {testo di default}
>
```

Es.

```
<INPUT type="text" name="nazionalita?" size="20" maxlength="40" default="italiana">
```

2) **type="radio"** *{crea dei pulsanti circolari a scelta singola (radio button)}*

```
<INPUT type="radio"
      name="nome del gruppo di bottoni"
      value="valore del bottone"    {valore che viene restituito}
      checked                    {il bottone è selezionato di default}
>
```

Es.

```
<FORM>
```

```
....
<P>
Quali di questi mezzi di trasporto utilizzi?
<INPUT type="radio" name="trasp" value="auto" checked> automobile
<INPUT type="radio" name="trasp" value="moto"> motocicletta
<INPUT type="radio" name="trasp" value="treno"> treno
<INPUT type="radio" name="trasp" value="aereo"> aereo
....
```

Oss. per avere la scelta singola i bottoni di tipo radio devono avere tutti lo stesso nome.

3) **type="checkbox"** *{crea dei pulsanti rettangolari a selezione multipla (check button)}*

```
<INPUT type="checkbox"
      name="nome del gruppo di checkbox"
      value="valore del box"      {valore restituito}
      checked                    {bottone selezionato per default}
>
```

Es.

<FORM>

....

<P>

Quali di questi browser utilizzi?

<INPUT type="checkbox" name="browser1" value="1"> Netscape

<INPUT type="checkbox" name="browser2" value="2"> MS Explorer

<INPUT type="checkbox" name="browser3" value="3"> Opera

....

4) **type="password"** *{crea un campo per inserire una password (sullo schermo vengono visualizzati degli *****)}*

```
<INPUT type="password"
      name="nome dell'input"
      size="numero"        {spazio che viene lasciato}
      value="testo"        {il testo è la password}
>
```

5) **type="submit"**
type="reset" *{creano pulsanti per inviare o cancellare i dati inseriti nel modulo}*

```
<INPUT type="submit"
      value="testo"    {il testo permette di personalizzare le scritte sui bottoni}
>
```

6) **type="button"** *{crea un pulsante generico (push button)}*

```
<INPUT type="button"
      value="testo"    {il testo permette di personalizzare la scritta sul bottone}
>
```

Oss. questo pulsante non ha un comportamento predefinito, comportamento che di solito viene creato mediante la programmazione in JavaScript.

Creazione di menu

Con i tag **<SELECT>** e **<OPTION>** si possono creare dei menu.

Per racchiudere il menu si usa **<SELECT>** *descrizione del menu* **</SELECT>**

Per ogni riga del menu si usa **<OPTION>** *valore dell'elemento* **</OPTION>**

```
<SELECT
  name="nome del menu"
  size="numero"      {numero delle righe visibili del menu}
>

<OPTION
  value="valore dell'elemento"      {valore restituito}
  selected                          {elemento selezionato di default}
>
testo che compare nel menu
</OPTION>
```

Es.

```
<FORM>
```

```
...
<P> Seleziona uno tra questi browser
<SELECT name="browser" size="2">
<OPTION value="1" selected>Navigator </OPTION>
<OPTION value="2"> Explorer </OPTION>
<OPTION value="3"> Opera </OPTION>
.....
</SELECT>
```

```
.....
</FORM>
```

Combinando opportunamente tutti questi elementi si possono costruire moduli complessi per creare delle pagine HTML interattive, nelle quali l'utente può inserire i propri dati e commenti. Queste informazioni possono essere usate da altri programmi. Ad esempio, i form sono utili in combinazione con i database: ogni form produce un record che viene memorizzato in un database per elaborazioni successive.

È buona norma costruire allineare gli elementi che fanno parte dei moduli. Per questo motivo quasi sempre i campi, i bottoni, i menu, ... vengono inseriti all'interno delle celle delle tabelle.