

Prerequisiti:

- **Oggetti multimediali.** Essi sono: le icone, le immagini animate, i suoni, i video, le animazioni in flash, banner, contatori di visite, ecc.
- **Pagina web.** Essa è una pagina contenente oltre al testo oggetti multimediali che è letta da un browser. Esse sono dei file aventi estensione html o htm come ad es. musica.html, cinema.htm.
- **Browser.** Esso, letteralmente sfogliatore o navigatore, è un programma che permette di leggere e sfogliare le pagine web. Esempi sono: Internet Explorer, Netscape Navigator, Mozilla, Opera, ecc.
- **HTML.** E' il linguaggio usato per scrivere le pagine web. Esso è una sigla: Hyper Text Markup Language ("linguaggio di contrassegno per gli Iper testi").
- **TAG.** Essi sono le parole chiavi che costituiscono il lessico del linguaggio HTML. Essi dicono al browser come impaginare e visualizzare il testo e gli oggetti multimediali nella pagina web.
- **Home page.** Essa è la pagina iniziale di un sito internet. Essa è denominata per convenzione index.html o pagina web indice ovvero pagina web che contiene l'indice di tutti i principali link alle pagine web interne al sito.
- **Sito internet.** Esso altro non è che una cartella contenente un insieme finito di pagine web e oggetti multimediali tra loro collegati attraverso i cosiddetti link ipertestuali.
- **Script¹.** Esso è un piccolo programma inserito tra i tag di una pagina web, interpretato ed eseguito da un altro programma detto interprete² Gli interpreti usati sono es. il browser (per gli script lato client) e il web server (per gli script lato server).
- **Linguaggi di scripting.** Essi sono i linguaggi usati per scrivere gli script. Essi sono dei linguaggi intermedi tra l'HTML e un linguaggio di programmazione vero e proprio come Java, C++, e Visual Basic, poichè la sintassi di un linguaggio di scripting è meno rigida e complicata di quella del corrispondente linguaggio di programmazione d'origine. Es. sono: VBscript (ha la sintassi simile al Visual Basic), JavaScript, Jscript (entrambi hanno la sintassi simile al java che è poi quella del C), CGI, Perl, Php (sintassi simile al C).³
- **WebServer o server web** è un programma che gira ininterrottamente su una macchina, in attesa di richieste del client (il nostro browser) da soddisfare. Ad es. quando clicchiamo su un link come <http://www.libero.it/index.html>, innanzitutto il nostro client (il nostro browser) attiva una connessione tra il nostro computer e il server web www.libero.it, richiedendo la pagina web di nome index.html; a questo punto il server riceve la richiesta e ricerca la pagina. Se il server web la trova, la consegna al browser, altrimenti invia un messaggio di errore molto noto: '404 not found' e poi chiude la connessione con il client (browser)

Gli strumenti software che useremo sono:

Il blocco note o meglio HAP-Edit (Editor testuale freeware), Paint o meglio Irfanview (tool di fotoritocco freeware), Frontpage o WS_FTP per l'upload delle pagine web del sito, il browser Internet Explorer.

Lezione 1 - Usabilità di un sito web

Definizione di usabilità di un sito

Un sito internet è usabile se può essere usato e visualizzato in modo semplice ed immediato dalla più ampia percentuale di utenti (anche disabili) e con il più ampio numero di browser.

¹ Nota al prof.: Script, Linguaggi di scripting e WebServer sono concetti necessari per la lezione otto dedicata ai form e quindi sono da spiegare solo a queglii alunni a cui interessano le pagine web dinamiche.

² Dalla programmazione sappiamo che i programmi per poter essere eseguiti devono essere dati in input ad un programma compilatore o ad un programma interprete.

³ ASP in realtà non è un linguaggio di scripting ma uno strumento che offre un ambiente che processa gli script di tipo Jscript (essa è la versione di JavaScript targata Microsoft con sintassi C-like usata anche dal PHP) o VBscript integrati all'interno delle pagine HTML. ASP supporta in modo nativo due linguaggi di scripting, *VBScript* e *JScript* purché essi vengano supportati dal server che dovrà ospitare ed elaborare le pagine realizzate. Tramite la direttiva `@ LANGUAGE` è possibile specificare il linguaggio da utilizzare all'interno di una determinata pagina ASP. Così, ad esempio, se una pagina ASP contiene come prima linea la direttiva `<% @ LANGUAGE = "JScript" %>` ASP interpreterà il codice contenuto all'interno di quella pagina come codice JScript

Sei regole dell'usabilità.

- 1) La grafica deve essere realizzata in modo da non penalizzare prestazioni. Spesso i siti sono lenti non perché eccessivamente grafici ma perché la grafica non è stata ottimizzata (immagini troppo grandi in termini di byte) o si sono usati componenti grafici (animazioni *in flash*) ove lo stesso effetto si poteva ottenere facilmente con codice html.
- 2) Poiché alcuni accostamenti possono non essere colti appieno da persone con difetti visivi, come ad es. : un testo rosso su sfondo nero, il testo rosso sull'arancio e peggio il testo arancio su rosso, bisogna evitare di usare colori dello sfondo e degli oggetti in primo piano troppo simili per tonalità.
- 3) I font da usare devono essere abbastanza grandi per essere letti senza difficoltà anche dagli ipovedenti e devono senza grazie come ad es. arial, microsoft sans serif, verdana, ecc.
- 4) La barra dei links principali nella home page del sito deve essere posta a sinistra o in alto poiché da sempre leggiamo (in occidente) da sinistra a destra e dall'alto al basso. Un utente che naviga in internet si aspetterà di vedere i links in questo ordine.
- 5) Ogni link o oggetto multimediale del sito deve essere evidenziata sempre anche con la sua descrizione. Ciò si realizza usando sempre nel tag `<a href>` e `` gli attributi `alt` e `title` utili per essere letti dagli screenreader (programmi di lettura della pagine web per i non vedenti).
- 6) Una pagina web deve fornire sempre una risposta alle seguenti domande: dove mi trovo? Dove sono stato? Dove posso andare? Occorre, pertanto, progettare una buona struttura di navigazione, in cui i link mostrano chiaramente i percorsi ipertestuali possibili.

Altre caratteristiche del linguaggio HTML⁴

- E' un pseudo-linguaggio di programmazione perché non possiede ad es. i cicli e i costrutti tipici di un linguaggio di programmazione; Inoltre, L'HTML è un linguaggio interpretato. Per questo motivo può succedere che la stessa pagina web venga interpretata e quindi visualizzata in modo diverso da due browser diversi.
- Fa uso dei tag o marcatori che in genere sono in un numero pari per poter rappresentare l'inizio (start-tag) e la fine (end-tag) di un elemento; da notare che non tutti i tag sono supportati dai browser: in tal caso il tag viene semplicemente ignorato dal browser senza segnalare alcun errore;
- Non è case-sensitive cioè ad es. `<TITLE>` è equivalente a `<title>` ed a `<TiTLe>`;

Altre regole importanti sull'usabilità.

- 7) se scrive con la tastiera e tutte le funzioni da impiegare durante la scrittura devono essere attivabili dalla tastiera per evitare la continua perdita di concentrazione data dall'andarsi a cercare il mouse ed i pulsanti a video (o peggio le mini-icone) e per consentire ai non vedenti di non dipendere dal mouse. Usare a tal proposito il tag `accesskey` nei links.
- 8) i messaggi e le registrazioni vocali devono essere accompagnati in alternativa da pagine web alternative per i non udenti.
- 9) per le pagine che contengono script, applet o altri oggetti di programmazione deve essere fornita informazione equivalente in un'altra pagina web accessibile e aggiornata con la stessa frequenza.
- 10) Attenzione anche ai diversi browser (Explorer, Netscape, Mozilla, ecc) e alle diverse versioni (Internet Explorer 5, Internet Explorer 6, ecc). Un sito dovrebbe essere visibile allo stesso modo con tutti i tipi di browser e non solo con le ultime versioni (ma almeno con quelle immediatamente precedenti): non usate plug-in⁵ che non siano assodati; non usate tecniche trendy, che potrebbero, in breve, rivelarsi dei flop.

⁴ Nota al prof.: tale caratteristiche dell'HTML e dell'usabilità successive comunicarle solo ai programmatori ITC e ITIS

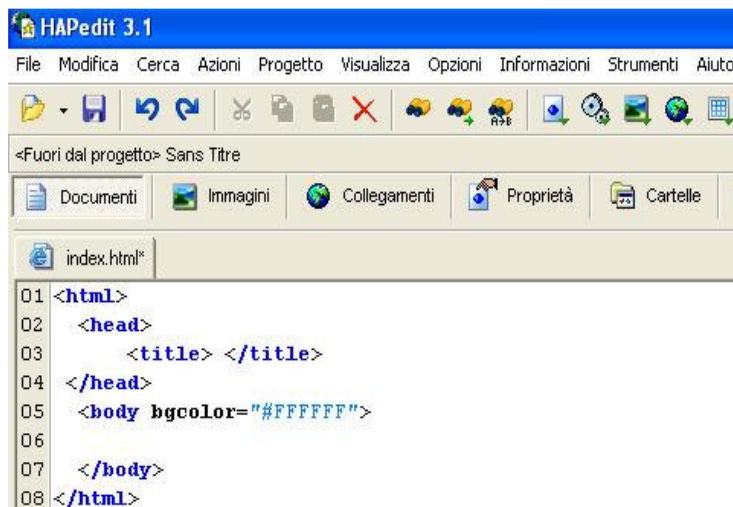
⁵ programma che viene installato "nel" browser e serve per eseguire per esempio Applet Java o pagine Flash. Ad es. il plug-in flash o il plug-in Java Run Time Enviroment.

Bisogna introdurre un motore di ricerca alle pagine web del sito, infatti: più del 50% degli utenti utilizza i motori di ricerca interni ai siti, solo il 15% degli utenti segue solo i link disponibili, circa il 35% degli utenti utilizza link e ricerca. Molti siti, infatti, presentano oltre che un motore di ricerca sul web anche un motore di ricerca all'interno del sito ed esso poiché è molto importante è spesso posizionato in alto a sinistra o a destra.

Un altro requisito che rende un sito navigabile è la presenza di una mappa del sito che permette di avere una visione ampia e per argomenti delle pagine web del sito e quindi di riuscire a trovare subito quello che si sta cercando.

6Lezione 2 - Il tag <BODY>

Un esempio di editor testuale un po' più evoluto del blocco note di Windows: HAP-Edit



Tra i vari editor testuali un po' più evoluti del blocco note di Windows consideriamo l'editor HAP-Edit, editor freeware utile per costruire una pagina html oltre che file CSS, Javascript, ASP e PHP da inserire eventualmente in una pagina web. Tale editor non è auto installante. Esso si avvia semplicemente cliccando due volte sul file denominato Cliccami.exe. Successivamente dal menù File ☐ nuovo ☐ selezionare l'opzione nuova pagina web.

La struttura HTML di una pagina web vista in figura

- i simboli come ad es. **<HTML>** e **</HTML>** si chiamano tag di apertura e tag di chiusura. Quindi un tag di chiusura si ottiene aggiungendo una / (slash) ad un tag di apertura. Molti tag ma non tutti, devono avere un tag di chiusura (vedi più avanti). Poiché i tag dicono al browser cosa fare, scrivendo solo i tag **<html>** e **</html>** è come se si dicesse al browser "questo è l'inizio di un documento HTML " (**<HTML>**) e "questa è la fine di un documento HTML (**</HTML>**).
- Ogni documento HTML necessita dei due tag (testa) **HEAD** e **/HEAD**, dei due tag (titolo) **TITLE** e **/TITLE**, annidati tra i tag HEAD.
- Ogni documento HTML necessita dei due tag (corpo) **BODY** e **/BODY** che delimitano il corpo di una pagina web⁷.
- Tra i tag HEAD sono presenti anche i tag **<META http-equiv=.....>** detti tag meta ovvero tag nascosto. Essi servono ad indicare al browser ad es.: il set di caratteri deve utilizzare quello occidentale (giapponese) se la pagina è in italiano (giapponese), le parole chiavi per ritrovare il sito nei motori di ricerca se precedentemente registrato, ecc. In seguito vedremo altri tipi di tag meta.
- La scritta **BGCOLOR** vicino al tag **<BODY>** definisce una **proprietà o attributo**. E' la proprietà del tag BODY responsabile del colore dello sfondo che vogliamo dare alla pagina web. Il colore è espresso in codice esadecimale. Tali codici sono su tutti i manuali e editor html, quindi non bisogna ricordarli a memoria.
- Il tag **<!-- -->** serve al programmatore di pagine web per inserire un suo commento sulla riga di codice html digitata. Tale tag di commento viene ignorato dal browser nel senso che tutto quello che noi scriviamo tra tale tag non viene interpretato dal browser e quindi non viene visualizzato.

⁶ Per convenzione scriveremo i nomi dei tag e quelli degli attributi maiuscoli e minuscoli valori degli attributi . In Html si può scrivere tutto maiuscolo, minuscolo o misto poiché esso non fa differenza. HAP-Edit, vedi fig., scrive tutto in minuscolo ma per distinguere i tag dagli attributi e dai valori usa il colore nero in grassetto per i tag, il colore blu in grassetto per gli attributi e il colore celestino per i valori degli attributi.

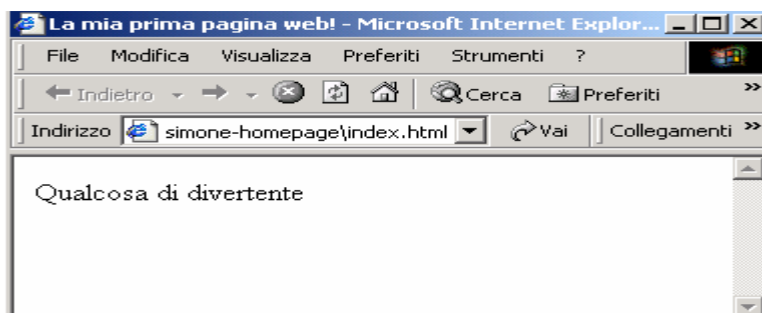
⁷ Nota al prof.: fai es. indicando in una pagina web di prova quale è il corpo

La mia prima pagina web!

- Scriviamo tra i tag TITLE “la mia prima pagina web !” e salviamo la pagina con il nome **index1**⁸ ed estensione html o Htm. Con l’editor HAP-Edit non è necessario specificare il formato⁹ html o Htm ma se noi utilizzassimo quale editor il blocco note di Windows dovremmo specificare il formato.

Con HAP-Edit quindi a noi non resta che scegliere il nome index1 e salvarla nella cartella che conterrà il nostro sito che si chiamerà cognome-homepage la quale sarà per comodità salvata a sua volta sul desktop o nella cartella assegnata alla vostra classe o meglio a ciascuno di voi alunni dall’amministratore di rete.

- Andiamo nella cartella cognome-homepage e visualizziamo la pagina index1 con il browser Internet Explorer □ compare:



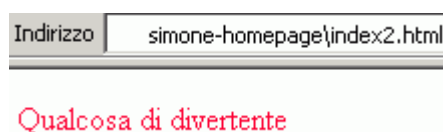
- la scritta la mia prima pagina web è comparsa sulla barra dei titoli.
- il testo “qualcosa di divertente scritto nel tag BODY è comparso nel corpo della pagina web.
- lo sfondo della pagina web è bianco □ vorrà dire che il codice esadecimale FFFFFFFF corrisponde al colore bianco (fai guardare gli allievi sul manuale). Nota bene il testo è di colore nero anche se non l’abbiamo impostato noi, questo significa che per default il testo sarà sempre nero. Saremo poi noi a settare, cambiare il colore del testo, se vogliamo, con l’attributo seguente:

L’attributo text del tag body

Aggiungendo l’attributo text = “#FF0033” al tag <BODY> come nella figura seguente:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <title> La mia prima pagina web! </title>
5     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
6   </head>
7   <body BGCOLOR="#FFFFFF" text="#ff0033">
8     <!-- codice esadecimale per il colore dello sfondo (bianco e del testo (rosso) -->
9     Qualcosa di divertente
10  </body>
11 </html>
```

il testo del corpo della pagina rispetto alla pagina index1 precedente cambia in rosso e quindi anche la scritta qualcosa di divertente. Chiaramente al posto del codice esadecimale #ff0033 si poteva inserire più comodamente la costante red che sta per rosso senza il simbolo di cancelletto ovvero così: text=red”. Per vedere il risultato apri la pagina di **index2** che il browser interpreta così:



⁸ Nota al prof. : E’ possibile anche utilizzare le pagine web di prova già pronte scaricabili dal sito personale.

⁹ Per cosa sia esattamente il formato di un file vedi più avanti. Comunque es. di formati sono: .doc, .ppt, .xls, ecc.

L'attributo background del tag body

Nella cartella icone-sito contenuta nella cartella cognome-homepage consideriamo l'icona linea.gif:

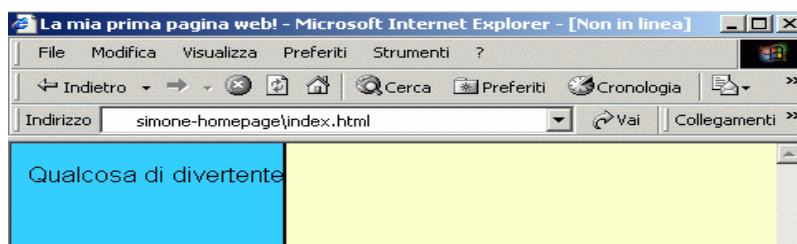


Sfruttando l'icona linea.gif suddetta scriviamo il codice HTML seguente:

```
<BODY TEXT="#FF0099" BACKGROUND="icone-sito/linea.gif">
```

Notiamo che nell'espressione "icone-sito/linea.gif" noi abbiamo specificato tre cose importanti:

- **il percorso dove si trova l'icona linea.gif** ovvero nella cartella icone-sito;¹⁰
- **il formato della icona** che è il .gif (si legge "punto gif") di cui parleremo più avanti.
- Inoltre, chiaramente, se usiamo l'attributo background, **l'attributo bgcolor non serve a più usarlo** perché rimane "coperto" dall'attributo background. Per vedere il risultato apri la pagina di **index3** che il browser interpreta così:



Da essa si capisce subito che l'icona linea.gif scelta come sfondo viene interpretata dal browser in modo ciclico ovvero l'una accanto all'altra sino a riempire completamente lo sfondo della pagina web.¹¹

L'attributo BGPROPERTIES del tag body

Nella cartella icone-sito consideriamo l'icona mago.gif:



il codice `<BODY TEXT="#FF0099" BACKGROUND="icone-sito/mago.gif" BGPROPERTIES="fixed">` con l'attributo `BGPROPERTIES="fixed"` permette di far scivolare il testo della pagina web sull'immagine di sfondo mantenuta fissa. Apri il file **index4** come esempio.

L'attributo BGSOUND SRC inserito subito dopo il tag body

Il codice `<BGSOUND SRC="vasco-midi.mid" loop="2">`. Questo tag permette di inserire una musica di sottofondo di qualsiasi formato come ad es. : .mid, .kar, .mp3, ecc ripetuta loop volte. In realtà i valori di loop = 0 e 1 eseguono il file musicale 1 sola volta; i valori di loop= -1 e infinite eseguono il file musicale un numero infinito di volte; infine i valori di loop = 2, 3 ,4 ecc eseguono il file musicale rispettivamente 2, 3, 4 volte, ecc. Apri il file **index5** come esempio.

¹⁰ vedi più avanti concetto di pathname assoluto e relativo.

¹¹ Fai vedere agli allievi prima il risultato nel browser e poi la sottigliezza dell'icona icona2.gif

Lezione 3 - I tag di formattazione del testo

Si parla di formattazione del testo quando si vuole mettere in grassetto, in corsivo, ecc. un testo. A tal proposito esistono nel linguaggio HTML i seguenti tag:

I tag , <I> e <U>

- pippo e pluto vanno al mare mettono **pippo e pluto vanno al mare** in grassetto.
- <I> pippo e pluto vanno al mare </I> mettono *pippo e pluto vanno al mare* in corsivo.
- <U> pippo e pluto vanno al mare </U> mettono la sottolineatura sotto pippo e pluto vanno al mare.

Questi tag possono essere combinati. In tal caso devono essere tra loro annidati cioè scritti così:
pippo e pluto <I> vanno <U> al mare </U> </I> cioè in modo che l'ultimo tag aperto (ad esempio <U> deve essere il primo ad essere chiuso). Apri il file **index6** come esempio. Esso mostra: in html il codice a sinistra e nel browser la figura a destra:

```
<BODY bgcolor="#FFFFFF">
```

```
pippo e pluto vanno al mare      <br>  
<b> pippo e pluto vanno al mare </b>      <br>  
<i> pippo e pluto vanno al mare </i>      <br>  
<u> pippo e pluto vanno al mare </u>      <br>  
<b> pippo e <i> pluto <u> vanno al mare </u> </i> </b> <br>  
  
</BODY>
```

Indirizzo | simone-homepage\index6.html

pippo e pluto vanno al mare
pippo e pluto vanno al mare
pippo e pluto vanno al mare
pippo e pluto vanno al mare
pippo e pluto vanno al mare

ove il tag
 , come si sarà capito, serve semplicemente a far andare da capo.

Il tag font con gli attributi size, face e color

- Il font rappresenta il tipo di carattere usato per rappresentare un carattere. I font si classificano in font con grazie come: Times New Roman e Monotype Corsiva e font senza grazie come: Arial; Microsoft Sans Serif, Lucida Sans Unicode, ecc.

Ogni browser ha come font di default il Times New Roman, ha come colore del testo di default il black o nero avente codice 000000 ed ha come grandezza del carattere di default il valore 3.¹² Infatti ad es. la frase “Qualcosa di divertente” scritta a pag 4 senza specificare il tipo di font che il browser doveva usare, il tipo di colore del font da usare e il tipo di grandezza del font è stata interpretata dal browser con il font Times New Roman con il colore nero e con la grandezza 3.

A che servono tali valori di default? Se il programmatore di pagine web ha previsto per una determinata pagina ad es. o un font o un colore oppure una dimensione non supportata dal browser che legge la pagina allora essa viene interpretata lo stesso ma usando il font di default Times New Roman, il colore di default nero e la grandezza del carattere di default che è il 3.

L'attributo FACE

 è la scritta usata per modificare il font di default. Tra virgolette noi specifichiamo in ordine da sinistra verso destra il tipo di font che il browser deve usare per leggere la pagina web. Il browser cerca nella cartella fonts di windows il font Arial. Se lo trova lo utilizza, se non lo trova, cerca nella medesima cartella il font Comic Sans. E così di seguito, se non trova nemmeno questo usa il font di default ovvero il font Times New Roman per leggere la pagina web.¹³

¹² I valori di default sono quelli che il browser assume se non gli si è detto niente di specifico.

¹³ Da ciò discende che se noi decidiamo di usare per il testo della pagina web sempre ad es. il font arial è chiaro che tale tag conviene aprirlo subito dopo il tag <body> e chiuderlo appena prima del tag </body>

L'attributo SIZE

`` è la scritta usata per specificare la grandezza del font.

Supponiamo di voler far comparire nel browser la scritta pippo e pluto vanno al mare.

Esempio 1

Tra i tag `<BODY>` e `</BODY>` scriviamo il codice HTML seguente:

` pippo e pluto vanno al mare `. Esso viene interpretato dal browser così:

Indirizzo | simone-homepage\index7.html

pippo e pluto vanno al mare

ovvero nel modo corretto perché prima di aprire un nuovo tag `` viene chiuso il tag font size precedente. Apri il file **index7** come esempio.

Questo tag chiude il secondo tag font ma tiene aperto ancora il primo.

Esempio 2

Tra i tag `<BODY>` e `</BODY>` scriviamo il codice HTML:

` pippo e pluto vanno al mare ` viene interpretato dal browser così: pippo e pluto vanno al mare. Esso viene interpretato dal browser così:

Indirizzo | simone-homepage\index8.html

pippo e pluto vanno al mare

Quindi questa forma, sebbene sia sintatticamente corretta, è sbagliata rispetto a ciò che ci avevamo posto come obiettivo. Apri il file **index8** come esempio.

L'attributo COLOR

`` permette di specificare per una parte di testo un colore particolare diverso da quello impostato nel tag `<body TEXT="#FF0099">`.

Nomi di colori predefiniti

Ci sono alcuni colori che sono individuabili anche attraverso dei nomi predefiniti oltre che dalla combinazione simbolo # e codice esadecimale di sei cifre. Alcuni esempi sono:

- fucsia, beige, black, blue, brown, olive, yellow, salmon, ecc. In tal caso la sintassi sarà:

- ``

- `<BODY BGCOLOR="white" text="black">` perché anche con il tag `<BODY>` è possibile usare tali nomi predefiniti. Apri il file **index9** come esempio per visualizzare a sinistra il codice HTML e a destra quello che viene interpretato dal browser:

```
<BODY bgcolor="white" text="black">
```

```
I colori di Pippo e Pluto
```

```
<br>
```

```
<font color="blue"> Pippo: " io sono dell'inter!" </font>
```

```
<br>
```

```
<font color="red"> Pluto: " io sono del milan!" </font>
```

```
<br>
```

```
Chi l'avrebbe mai detto che anche Pippo e pluto erano tifosi!  
</BODY>
```

Indirizzo | simone-homepage\index9.html

I colori di Pippo e Pluto

Pippo: " io sono dell'inter!"

Pluto: " io sono del milan!"

Chi l'avrebbe mai detto che anche Pippo e pluto erano tifosi!

Deduzioni:

- il colore del testo stabilito dall'attributo text del tag body è quello che viene usato sempre.
- se noi desideriamo dare ad una frase particolare del corpo della pagina web un colore diverso è possibile farlo attraverso l'uso dell'attributo color del tag font, aperto e chiuso con al loro interno la frase di cui vogliamo cambiare colore.

I tag
, <P>, <HR> e <CENTER>

Scriviamo nel corpo della pagina web tra i tag <BODY> e </BODY> la seguente frase: pippo e pluto vanno al mare a pescare su più linee come mostra la figura seguente:

```
<BODY bgcolor="#FFFFFF">
  pippo
  e pluto
  vanno al
  mare a
  pescare
</BODY>
```

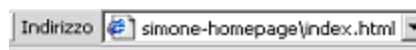
Che succede se visualizziamo la pagina web nel browser? Succede che la frase scritta su più linee viene invece riportata su una sola linea, perché il browser non riconosce la formattazione della frase scritta su più linee. Infatti, se non viene indicato diversamente, il browser visualizza i caratteri uno dietro l'altro. Quindi se vogliamo andare a capo dobbiamo utilizzare un tag specifico: **il tag
**. Da notare che il tag
 è un es. di tag che non ha bisogno del tag di chiusura.

I caratteri speciali

Scriviamo tra i tag <BODY> la frase “Qualcosa di divertente” separando le parole (vedi fig.1)

```
<BODY bgcolor="#FFFFFF">
  Qualcosa      di      divertente
</BODY>
```

(fig.1)



Qualcosa di divertente

(fig. 2)

e lanciamo l'anteprima della pagina web. Quello che notiamo è rappresentato nella fig. 2 ovvero notiamo che il browser non riconosce più di uno spazio.

Allora come dobbiamo fare per inserire tra le parole di un testo più di uno spazio? Bisogna usare il carattere speciale: ** **; uno per ogni spazio che vogliamo inserire.¹⁴

Altri caratteri speciali sono:

- **<**; (per inserire un segno minore, infatti, lt sta per less then) ;
- **>**; (per inserire un segno maggiore, infatti, gt sta per great then)
- **&**; (per inserire &);
- **"**; (per inserire virgolette)
- **­** (per inserire un trattino);
- **©**; (per inserire il simbolo di copyright ©)

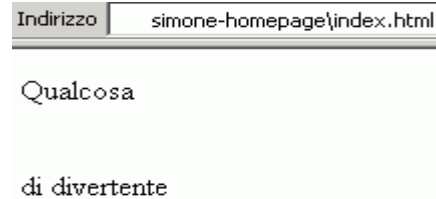
Lasciare più di una riga vuota nella pagina web

Per lasciare più di una riga vuota nella pagina web basta usare in modo combinato il carattere speciale ** ** e il tag **
** tante volte per quante righe vuote si vogliono inserire. Ad esempio scrivendo (vedi fig.1) tre volte **
**

¹⁴ Come esempio inserisci nel codice html tra le parole qualcosa, di e divertente alcuni simboli ** **;

```
<BODY bgcolor="#FFFFFF">  
Qualcosa  
&nbsp; <BR>  
&nbsp; <BR>  
&nbsp; <BR>  
di divertente  
</BODY>
```

(fig. 1)



(fig. 2)

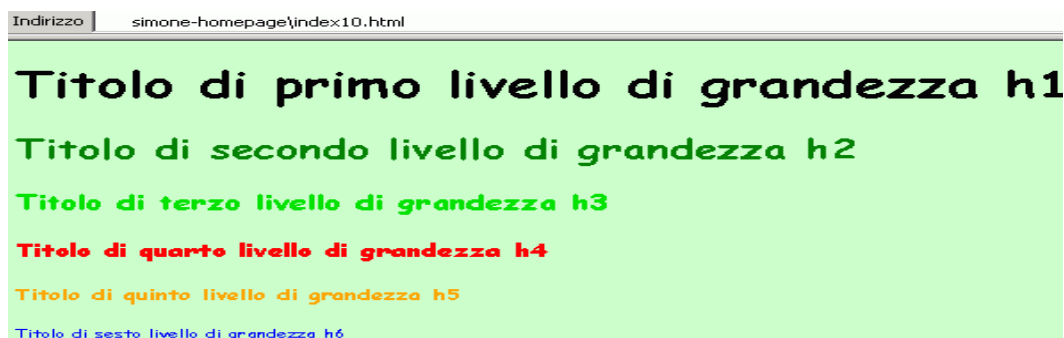
riesco a scrivere tre righe vuote sotto la parola “di divertente” dalla parola “Qualcosa ” (fig. 2)

- Simile a **
** è il **tag <P>**. E' esattamente la stessa cosa, ma oltre ad andare a capo, aggiunge un'interlinea. I tag **<P>** possono essere usati quindi per il titolo di un paragrafo essendo che esso è in genere seguito da un ritorno a capo e da una interlinea.

Il tag **<P>** possiede inoltre l'**attributo ALIGN** che può assumere i valori: **LEFT, CENTER e RIGHT** che rispettivamente spostano il testo contenuto tra i tag **<P>** e **</P>** tutto a sinistra dello schermo, al centro dello schermo o tutto a destra dello schermo.

Lezione 4 - Altri tag di formattazione, il tag <MARQUEE> e il pathname

- Il tag permette di porre in grassetto marcato il testo compreso tra esso e il suo tag di chiusura. In alcuni browser esso ha la stessa funzione del tag
- Il tag <HR> semplicemente permette di inserire una linea di separazione nel corpo della pagina nella posizione in cui si trova il tag <HR> avente lo stesso colore del testo impostato o nell'attributo text del tag <BODY> o nell'attributo color del tag .
- Il tag <CENTER> permette di centrare il testo scritto tra esso e il suo tag di chiusura.
- I tag <H1,2,3,4,5 e 6> sono usati in genere per i titoli dei paragrafi, pertanto anche essi in automatico mandano il testo a capo, senza usare il tag specifico
. Il tag <H?> fornisce sei possibili grandezze che aumentano di dimensioni al decrescere del numero. Quindi il titolo più grande lo fornisce il tag <H1>. Anche i tag <H?> necessitano dei tag di chiusura. Apri il file **index10** che mostra la figura seguente:



Il tag <MARQUEE>

```
<BODY BGCOLOR="#FFFFFF">  
<H1 ALIGN="CENTER"> Benvenuto su </H1>  
<CENTER> <IMG SRC="icone-sito/logoindex.gif"> </CENTER>  
  <MARQUEE = "100" ALIGN="MIDDLE" WIDTH="600" HEIGHT="40">  
    <STRONG>  
      <BIG>  
        Il portale del materiale didattico da scaricare Gratis e delle news sulla scuola  
      </BIG>  
    </STRONG>  
  </MARQUEE>  
</BODY>
```

Tale tag possiede gli attributi:

- SCROLLDELAY che definisce la velocità di scorrimento del testo;
- l'attributo ALIGN già visto per il tag <P> e che può assumere i valori LEFT, CENTER e RIGHT.
- gli attributi WIDTH e HEIGHT che definiscono la larghezza e l'altezza dello spazio nel quale deve scorrere il testo.

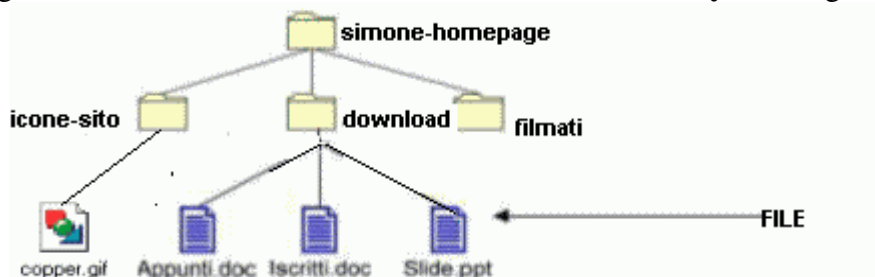
Come esempio apri il file index11 che mostra la figura seguente:



Concetti di file, di formato di un file, di cartella o directory, di file system

Sappiamo che le informazioni nel computer sono memorizzate sull'hard disk in "contenitori" logici chiamati **file**. Ciascun file è caratterizzato da un nome e da una estensione detta formato. L'estensione è in genere composta da 3 o 4 caratteri come ad es. html, doc, gif, ecc ed essa è solo una convenzione stabilita dall'utente al fine di distinguere due file con lo stesso nome ma con formato diverso e dipende dalla natura delle informazioni che esso contiene. Tale estensione è necessaria esplicitarla nella costruzione delle pagine web.

Oltre ai file esiste un altro tipo di "contenitore" detto cartella o directory che altro non sono che contenitori logici¹⁵ di file raggruppati secondo criteri significativi per l'utente: per dimensione, per data di ultima modifica, per nome, ecc. Le cartelle possono a loro volta contenere altre sottocartelle che possono contenere altre cartelle e così via. In tal caso si parla di gerarchia di cartelle che dà vita alla struttura logica cosiddetta ad "albero" delle cartelle chiamata **file system** seguente:



Pathname (o percorso) relativo di un file.

Il percorso di un file è definito relativo se fa riferimento solo alla cartella ove è contenuto il file. Ad es. consideriamo il file **appunti.doc** della struttura a file system precedente contenuto nella cartella **download** interna alla cartella **simone-homepage**.

Quale percorso devo fare per raggiungere tale file? Il percorso relativo da fare per recuperare il file di esempio **appunti.doc** dipende dalla posizione corrente in cui si trova il file index che lo richiama.

- se il file index si trovasse nella cartella **simone-homepage** dovrei entrare nella cartella **download** e selezionare il file **appunti.doc**. In HTML ciò si ottiene scrivendo "**download/appunti.doc**"
- se il file index si trovasse nella cartella **filmati** dovrei prima uscire da tale cartella. In HTML ciò si ottiene scrivendo "**../**" e poi dovrei entrare nella cartella **download** e selezionare il file **appunti.doc**. In HTML il percorso totale relativo si scrive: "**../download/appunti.doc**"

Pathname (o percorso) assoluto (o completo) di un file.

Il percorso di un file è detto assoluto nei seguenti casi di esempio:

- **c:/documents and settings/all users/desktop/simone-homepage** (caso a)
- **http://www.libero.it/** (caso b)

ovvero quando viene specificato quale percorso da fare un percorso o che è strettamente legato all'hard disk del nostro computer (caso a) o ad un sito esterno alla nostra pagina web (caso b).

Quando usare il pathname relativo e assoluto

Chiaramente quando si costruiscono pagine web è consigliabile usare per le immagini e per i links tra le pagine web del sito esclusivamente pathname relativi, in quanto l'uso di pathname assoluti farebbe "funzionare" le pagine web solo ed esclusivamente nel disco locale in cui sono state create. L'uso di pathname assoluti come mostrato anche nell'esempio si esaurisce ai link a siti esterni al nostro.

¹⁵ logici perché nella realtà sul disco fisso i file sono raccolti in modo diverso.

Lezione 5 - Il tag e gli attributi src, width, height, border, align e alt

Consideriamo l'icona copper.gif inserita nella cartella icone-sito seguenti:



dopodiché tra i tag <BODY> e </BODY> digitiamo il codice HTML seguente:

`` ove:

- icone-sito/copper.gif è il percorso relativo ove si trova l'icona;
- WIDTH e HEIGHT sono la larghezza e l'altezza dell'icona in numero di pixel ricavate selezionando l'icona □ proprietà □ scheda riepilogo. Inserire le dimensioni WIDTH e HEIGHT di un'icona è importante perché in questo modo il browser predispone già lo spazio necessario ad ospitarla e rende lo scaricamento della pagina più veloce.

Gli attributi WIDTH e HEIGHT del tag

Consideriamo l'icona red-dot.gif (puntino rosso) contenuta nella cartella icone-sito e scriviamo la righe di codice seguente:

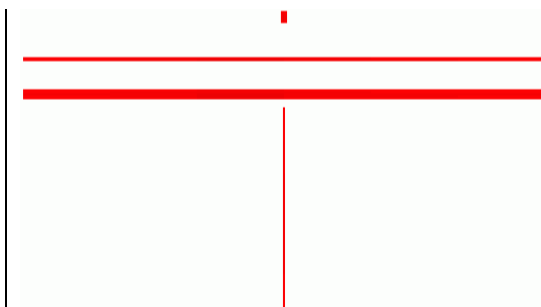
```
<CENTER>  <!-- tag center per disegnare al centro della finestra -->
  <IMG src="icone-sito/red-dot.gif" width="6" height="6"> <BR>

  <IMG src="icone-sito/red-dot.gif" width="510" height="2"> <BR>

  <IMG src="icone-sito/red-dot.gif" width="510" height="5"> <BR>

  <IMG src="icone-sito/red-dot.gif" width="2" height="100"> <BR>

</CENTER>
```



Quindi grazie ad un piccolo puntino rosso è possibile creare e tracciare delle linee orizzontali e verticali e di diverso spessore specificando opportunamente gli attributi width e height. Apri il file **index12** come esempio.

L'attributo BORDER del tag

- L'attributo BORDER inserito nel tag come dice la parola serve semplicemente a specificare lo spessore del bordo che vogliamo dare all'immagine. Esso assume i valori interi: 0,1,2,3,4,5,6 e 7 ove scrivendo ad esempio:

`` significa che vogliamo che l'immagine abbia un bordo spesso pari a 0, cioè nullo ovvero senza nessun bordo. Difatti la riga di codice precedente equivale alla seguente:

`` che è la stessa di prima senza proprio l'attributo border. Questo significa che per default nel tag è specificato l'attributo **BORDER = 0**.

L'attributo align del tag

Normalmente per default il fondo di un'immagine e' allineato con il testo. L'immagine e' come se fosse "poggiata" sulla linea del testo. Con l'attributo **ALIGN**¹⁶ è possibile modificare tale allineamento.

- Con l'attributo **ALIGN** e i valori **TOP**, **MIDDLE** E **BOTTOM** è possibile modificare l'allineamento dell'immagine facendo visualizzare il testo rispettivamente con la parte alta dell'immagine, con la parte media (centrale) dell'immagine, con la parte bassa dell'immagine.

- Con l'attributo **ALIGN** e i valori **LEFT** e **RIGHT** è possibile far visualizzare rispettivamente l'immagine a sinistra o a destra dello schermo. Apri file **index13** contenente le opzioni possibili.

L'attributo alt del tag utile per i browser vocali dei non vedenti

Innanzitutto che cosa i browser vocali? Sono dei particolari browser usati dai non vedenti esclusivamente durante la visita di un sito web oppure per scaricare la posta elettronica. Un esempio di browser vocale attualmente in commercio è **Ibm Home Page Reader** dal costo intorno ai 170€. Come funzionano i browser vocali? Essi fatti girare su computer che possiedono una scheda audio e delle casse trasformano in voce sintetizzata il testo che appare sullo schermo e per questo motivo interpretano solo il testo e quindi quando incontrano una immagine se essa non possiede una breve descrizione alternativa (da qui il nome dell'attributo **ALT**) viene ignorata.

D'altra parte l'utilità dell'attributo alt la si può notare anche quando la pagina web viene interpretata da un browser classico che per un motivo o un altro non riesce ad aprire e a caricare l'immagine. In tal caso se è stato inserito l'attributo alt, al posto dell'immagine che non si visualizza compare un messaggio alternativo che è quello scritto nell'attributo alt.

Esempio

Inserisci il codice html seguente:

```
<BODY BGCOLOR="#FFFFFF" TEXT="red">  
<IMG ALT="icona poliziotto" SRC="icone-sito/copper.gif" WIDTH="100" HEIGHT="80">  
<HR> <!-- inea di separazione tra le due icone -- >  
<IMG ALT="icona mago" SRC="icone-sito/mago.gif" WIDTH="120" HEIGHT="100">  
</BODY>
```

poi vai su Internet Explorer ☐ Menù Strumenti ☐ Opzioni Internet ☐ Avanzate ☐ togliamo la spunta su mostra immagini (ovvero “settiamo” Internet Explorer in modo che non visualizzi le immagini). Quello che appare è nella figura seguente:



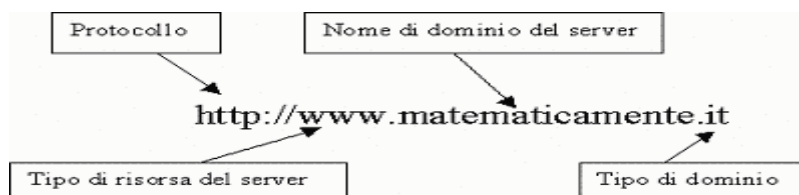
Apri il file **index14** come esempio.

¹⁶ To align significa, infatti, allineare.

Lezione 6 - Il tag <A> e gli attributi HREF, NAME, TITLE E ACCESSKEY

La principale potenza dell'HTML deriva dalla sua abilità e flessibilità nel linkare (collegare tra loro) regioni di testo (e anche immagini) ad:

- a qualunque link esterno specificato tramite URL (Uniform Resource Locator)
- un altro documento dello stesso sito;
- ad una sezione dello stesso documento;
- ad una sezione di un secondo documento.



ove esempi di URL sono: **ftp://ftp.nevada.edu/pub/guitar** (che indica l'indirizzo dell'account guitar su un server ftp); **http://www.ascuoladi.135.it** (che indica l'indirizzo di un sito web); **mailto:info@linux.it** (che indica un indirizzo e-mail); **news:soc.culture.italian** (indica il nome di un gruppo di discussione al quale è possibile accedere).

Esempio 1 - Link ad un URL esterno al nostro sito

Per creare dalla nostra pagina index un link esterno al nostro sito basta usare il tag <A> e l'attributo href uguagliato al percorso (path) esterno assoluto da fare per raggiungere il file nella rete Internet o per richiamare il client mail, ftp o news installato nel nostro computer. Ad esempio tra i tag <BODY> e </BODY> inseriamo le righe di codice seguenti:

```
<P> <A href="http://www.google.it"> google home page </A> </P>
<P> <A href="mailto:profsimone@katamail.com"> e-mail prof &nbsp;simone </A> </P>
<P> <A href="ftp://ftp.nevada.edu/pub/guitar"> server ftp </A> </P>
<P> <A href="news:soc.culture.italian"> server news </A> </P>
```

ove i tag <P> e </P> servono per inserire tra un link ed un altro una linea vuota. In queste quattro linee di codice abbiamo creato quattro link a quattro URL esterni. Apri il file **index-link1** come esempio.

Caso particolare dell'esempio 2 – link da una pagina web ad un file interno al nostro sito

Per creare dalla nostra pagina index un link ad un file interno al nostro sito basta usare il tag <A> e l'attributo href uguagliato al percorso (path) relativo da fare per raggiungere il file nella cartella del nostro sito.

Ad esempio tra i tag <BODY> e </BODY> inseriamo le righe di codice seguenti:

- ** appunti di accessibilità ** per creare dalla pagina web index-link2 un collegamento al file appunti.doc contenuto nella cartella download del nostro sito.
- ** pantera rosa ** per creare un dalla pagina web index-link2 un collegamento al file video prosa.exe da vedere direttamente nel sito man mano che è scaricato contenuto nella cartella download del nostro sito cliccando sul pulsante: "esegui l'applicazione dalla posizione corrente" nella finestra che compare.
- ** pantera rosa-256KB ** per creare un dalla pagina web index-link2 un collegamento al file video prosa.zip contenuto nella cartella download del nostro sito da scaricare prima sul proprio PC cliccando sull'opzione: "salva il file su disco" e poi dopo averlo unzippato di poterlo vedere.

Apri il file **index-link2** come esempio.

Esempio 3 - link ad una sezione di una pagina web individuata da un'ancora

Una pagina web può essere più lunga dell'altezza del nostro monitor e per questo motivo per poterla sfogliare nella sua interezza dovremmo muovere la barra di scorrimento laterale sino alla fine della pagina. In alternativa all'inizio della pagina web è possibile creare dei link a particolari sezioni della stessa in modo da farci collegare ad esse direttamente senza sfogliare tutta la pagina. Per fare ciò come, come accade anche in Word quando vogliamo creare un ipertesto, dobbiamo inserire nei punti in cui vogliamo che ci sia un salto **un segnalibro o ancora o riferimento**, cosiddetta appunto perché crea un punto di ancoraggio alla sezione della pagina a cui vogliamo saltare.

Supponiamo che stia scrivendo una guida sull'HTML avente le seguenti sezioni (fig. 1 a sinistra):

- lezione 1;
- lezione 2;
- lezione 3;
- lezione 4;

(fig. 1)

` lezione1 `

` lezione2 `

` lezione3 `

` lezione4 `

(fig. 2)

allora l'obiettivo nostro è creare delle ancore a tali sezioni in modo da facilitare l'utente nel reperimento delle sole lezioni che gli necessitano. Per inserire le ancore suddette basta inserire all'interno dei tag `<BODY>` e `</BODY>` nella posizione scelta il tag usato per i link con l'attributo `name` nel seguente modo ` lezione numero ` (vedi fig. 2).

Nel file di esempio **index-ancora**, supponendo di voler creare un'ancora alla lezione 1 della guida HTML, come esempio di ancora abbiamo creato la seguente:

` lezione1 `

Per vederla vai nel codice HTML un po' prima del tag di chiusura `</BODY>`. Faccio notare che la sola creazione di un'ancora non è interpretata dal browser quindi l'apertura di tale file **index-ancora** non produrrà nulla.

A questo punto creiamo un link all'ancora inserita. Basterà usare il tag `<A>` ove l'attributo `href` punterà al link che ha lo stesso nome dell'ancora definita su. Quindi scriveremo il codice seguente:

` Lezione 1 `

Quindi per creare un link all'ancora sezione 1 inserita bisogna usare l'attributo `href` uguagliato al simbolo di cancelletto nome dell'ancora. Apri il file di esempio **index-ancora-link** per vedere l'effetto creato dall'ancora.¹⁷

Combinazione dei tag `<A>` e ``

Nei link anziché usare un testo per collegare una pagina web possiamo usare un'immagine. La costruzione del link è infatti la stessa. Ma che significa usare una immagine come link?

Guardando l'esempio 1 precedente per creare un link esterno al sito al motore di ricerca google, abbiamo scritto al centro tra i tag `<A>` e ``:

` google home page `

con l'obiettivo di far comparire nel browser quale testo del link: "google home page". Ebbene, noi possiamo far comparire anche una immagine al posto del testo suddetto inserendo il tag `` al centro tra i tag `<A>` e `` nel seguente modo:

``

``

``

Apri il file **index-link3-image** come esempio.

¹⁷ La stessa cosa usando Frontpage si ottiene come in Word inserendo prima un segnalibro dal menù Inserisci e poi inserendo un collegamento ipertestuale con l'opzione inserisci nel documento stesso → selezionare il segnalibro scelto.

Esempio 4 - link con uso degli attributi title e accesskey

- L'attributo **title** serve per specificare un testo esplicativo quando si clicca il link. Questo testo addizionale o etichetta favorisce l'accessibilità del sito anche ai disabili, poiché esso viene letto dagli screenreader. L'attributo "title" è anche utilissimo per migliorare la propria presenza nei motori di ricerca, che ne vanno a leggere il contenuto.

```
<a href="http://www.ascuoladi.altervista.org" title="questo è il sito del materiale didattico gratis"
Homepage </a>
```

- L'attributo **accesskey** permette di raggiungere un link da tastiera usando la combinazione dei tasti "ALT + carattere+ invio"¹⁸, usando, cioè delle scorciatoie "da tastiera" che possiamo utilizzare nel nostro sito. Si tratta di scegliere delle lettere della tastiera, in genere le iniziali del link, che quando vengano digitate dall'utente permettono di andare direttamente a determinate pagine. La sintassi per la realizzazione di un link è la seguente:

1° Esempio

```
<a
href="http://www.ascuoladi.altervista.org/matematica.html"
accesskey="m"><b>m</b>atematica
</a>
<!--1° versione: grassetto della prima lettera -->
```

Il codice a fianco produce il seguente risultato nel browser:

[**m**atematica](http://www.ascuoladi.altervista.org/matematica.html)

Questo link permette, indicando la lettera associata, di saltare alla pagina web matematica.html utilizzando la combinazione Alt + m per mettere il focus sul link ed [Invio] per concretizzare il link. E' consigliabile evidenziare la lettera associata alla chiave di accesso. A tal fine, nell'es. visto si è decisi mettere in grassetto la prima lettera m del link matematica.

2° Esempio

```
<a
href="http://www.ascuoladi.altervista.org/informatica.html"
accesskey="i"><b>( i )</b>nformatica
</a>
<!--2° versione: parentesi tonde della prima lettera -->
```

Il codice a fianco produce il seguente risultato nel browser:

[**\(i \)**nformatica](http://www.ascuoladi.altervista.org/informatica.html)

Questo link permette, indicando la lettera associata, di saltare alla pagina web informatica.html utilizzando la combinazione Alt + i per mettere il focus sul link ed [Invio] per concretizzare il link. E' consigliabile evidenziare la lettera associata alla chiave di accesso. A tal fine, nell'es. visto si è decisi mettere in la prima lettera i tra parentesi tonde del link informatica

In genere gli attributi accesskey devono essere riservate per la navigazione dei menu che portano alle parti principali del sito. Comunque, si tratta, insieme all'attributo title di un'altra tecnica per migliorare l'accessibilità.

Apri il file **index-link4-accessibilità** come esempio.

¹⁸ Non in tutti i browser, soprattutto quelli più vecchi, funzionano gli accesskey, inoltre con Mozilla e Netscape Navigator è sufficiente la combinazione Alt + carattere.

Lezione 7 - Le tabelle e Uso delle tabelle con HAP-Edit


Per quanto riguarda la lezione delle tabelle rimando ai tantissimi manuali che ci sono in rete, anche perché le tabelle usano molti attributi come ALIGN, BORDER, WIDTH, HEIGHT già visti per gli altri tag fin qui analizzati e che hanno per le tabelle lo stesso significato. D'altra parte i tag principali delle tabelle sono tre: <TABLE> e </TABLE>, <TR> e </TR>, <TD> e </TD>. La cosa più importante da ricordare è che poiché una tabella per essere definita tale deve avere almeno una riga e quindi almeno una cella allora anche per definire una sola cella occorre introdurre tutti e tre i tag suddetti come ad esempio:

```
<TABLE >
  <TR>  <!-- Occorre un tag <TR> per ogni riga della tabella che si vuole considerare -->
    <TD>EZIO GREGGIO  </TD>
    <TD>ENZO IACCHETTI </TD>
    <!-- Ci sono due celle all'interno della stessa riga della tabella - ->
  </TR>
</TABLE>
```



Come esempio apri il file **index-table** seguente che mostra l'uso di tre tabelle: 1x1, 1x2 e 2x2:

| | |
|------------------------|----------------|
| e-mail | |
| Ezio Greggio | Enzo Iacchetti |
| Ezio Greggio | Enzo Iacchetti |
| Le veline | Il Gabibbo |

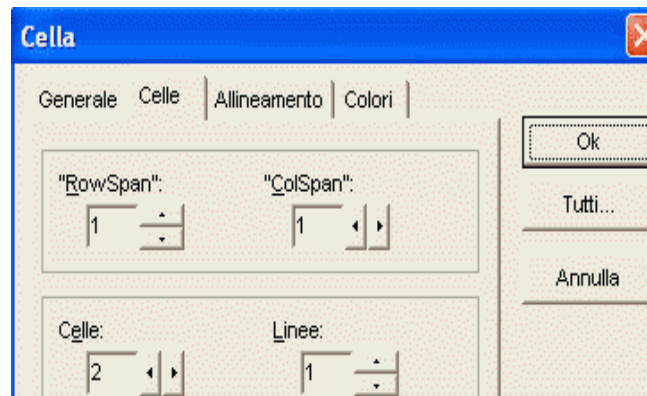
Creazione di una tabella con HAP-Edit

In HAP-Edit una tabella è possibile crearla in due semplici passaggi. Nel primo passaggio cliccando sul pulsante  della barra degli strumenti compare la figura seguente:



Nella scheda Generale è possibile specificare quale deve essere l'altezza (HEIGHT) e la larghezza (WIDTH) che deve avere la tabella in termini di numero di pixel o in percentuale.¹⁹ Nella scheda Frame è possibile specificare quanto deve essere lo spessore del bordo della tabella. Nella scheda Colori è possibile specificare il colore del bordo e dello sfondo della tabella. Passiamo al secondo passaggio. Cliccando di nuovo sul pulsante  e poi sul pulsante  compare la finestra seguente:

¹⁹ La misura in percentuale è quella consigliabile e per questo motivo essa è l'opzione settata per default da HAP-Edit



Partiamo dalla scheda Cella la più importante perché permette di specificare il numero di righe e di colonne che dovrà avere la tabella. Il numero di colonne è indicato con il nome Cella e il numero di righe con il nome Linee (vedi fig.), quindi, ad es., per creare una riga (ovvero una coppia di tag `<TR>` e due colonne (ovvero due coppie di tag `<TD>`) bisogna dare il valore 2 al nome Cella e il valore 1 al nome Linee come mostrato in figura. Nella stessa scheda compaiono le opzioni RowSpan e ColSpan. Essi si riferiscono agli omonimi attributi delle celle di una tabella che vedremo più avanti.

Nella scheda generale è possibile specificare l'altezza (HEIGHT) e la larghezza (WIDTH) delle celle in pixel o in percentuale.²⁰ Nella scheda Allineamento è possibile specificare l'allineamento delle celle: a sinistra (ALIGN="LEFT"), a destra (ALIGN="RIGHT") o al centro (ALIGN="CENTER"). Nella scheda Colori è possibile specificare il colore del bordo e dello sfondo delle celle anche in modo molto preciso cliccando sul pulsante Altro che permette di scegliere qualsiasi tipo di colore oltre a quelli predefiniti.

²⁰ Anche per le celle la misura in percentuale è quella consigliabile e per questo essa è l'opzione settata per default

Lezione 8 - I Form

I form sono moduli on line riempibili dai visitatori del sito che possono essere usati per commenti al sito, questionari, prenotazioni, ordinazioni, immissione di dati, registrazione di account e un'ampia varietà di altre applicazioni. Il modulo si trova all'interno del corpo della pagina HTML. L'utente lo riempie e poi lo invia, attraverso il browser, all'URL indicato dall'autore. L'URL può essere un indirizzo e-mail o uno script CGI, ASP, PHP lato server residente in una cartella del WebServer che provvederà a processarlo per risalire alle informazioni inserite dal visitatore.

La struttura HTML di un form è la seguente:

```
<HTML>
  <HEAD>
    <TITLE>          </TITLE>
  </HEAD>
<BODY>
<FORM>  <!-- inizio del modulo o form -->
<INPUT>  <!-- per caselle di tipo: testo, password, spunta singola o multipla, per pulsanti -->

<SELECT> <!-- per selezioni singole o multiple -->

    <OPTION> </OPTION>
    <OPTION> </OPTION>

</SELECT>

<TEXTAREA> <!-- per aree per raccogliere commenti e descrizioni --> </TEXTAREA>

</FORM>  <!-- fine del modulo o form -->

</BODY>
</HTML>
```

<FORM> è l'elemento che dichiara l'apertura del form o modulo.

Gli attributi del tag **<FORM>**

I principali sono tre: **ACTION**, **METHOD** e **NAME** e servono rispettivamente:

1) **ACTION** serve per inviare il contenuto del modulo riempito ad un URL specifico che può essere l'indirizzo di uno script CGI, ASP, PHP, Javascript ecc lato server residente in una cartella del WebServer installato oppure, più semplicemente, l'indirizzo e-mail del webmaster.

Esempio1: ad es. scrivendo:

```
ACTION="http://www.macchina.it/cgi-bin/script"
```

In questo caso abbiamo "inviato" i dati ad un programma "script", contenuto all'interno di una directory "cgi-bin" del webserver²¹ denominata "www.macchina.it", ossia ad un programma che provvederà a decodificare i dati, a manipolarli ed a fornire l'output del suo lavoro al server.

²¹ L'indirizzo di un webserver inizia sempre per http://, come analogamente l'indirizzo di un server ftp inizia sempre con ftp://

Esempio2: ad es. scrivendo:

```
ACTION="mailto:user@macchina.it"
```

In questo caso abbiamo inviato i dati nella mail del webmaster, senza elaborarli.

2) METHOD specifica la regola usata per spedire i contenuti della form. Può assumere i due valori GET (di default) e POST. Con GET, i dati del moduli sono inviati all'URL indicato attraverso una query²². Con POST, invece, i dati vengono codificati, impacchettati ed inviati all'URL. E' quindi possibile inviare una maggior quantita' di dati con maggior sicurezza. Per questo motivo e' consigliabile l'uso di POST piuttosto che GET.

3) NAME è il nome che preferiamo assegnare al modulo. Esso diventa opzionale in HTML ma indispensabile se si dovesse far uso di JavaScript o altro linguaggio di programmazione.

4) ENTYPE specifica il tipo di formattazione che dovrà avere il testo inviato via e-mail. Ad es. associando ENTYPE="application/x-www-form-urlencoded" si provoca un messaggio di posta elettronica nel quale i dati associati alla form vengono spediti come attachment (allegato); invece, associando ENTYPE="text/plain", il messaggio non viene inviato come attachment, ma come corpo del messaggio stesso.

5) TARGET: che consente di far aprire i dati del form in una pagina web differente rispetto a quella corrente.

All'interno del tag **<FORM>** per creare i cosiddetti campi del form si utilizzano, come visto, tre nuovi tag: **<INPUT>**, **<SELECT>**, **<OPTION>** e **<TEXTAREA>**. Partiamo con il primo.

Il tag **<INPUT>**

Esso non possiede tag di chiusura ed è caratterizzato dai seguenti attributi principali: **NAME**, **TYPE**, **VALUE**, **MAXLENGTH**, **SIZE**, **DISABLED** e **ERROR** e servono rispettivamente:

- **NAME** per definire il nome del campo che raccoglie il dato introdotto.
- **TYPE** per definire il tipo del campo. I principali tipi sono: **TEXT**, **SUBMIT**, **BUTTON**, **PASSWORD**, **RESET**, **FILE**, **RADIO** (scelta singola) e **CHECKBOX** (scelta multipla)
- **VALUE** per inserire un valore predefinito all'interno del campo e resterà tale nel form a meno che l'utente che compilerà il form non lo modifichi.
- **MAXLENGTH** stabilisce la lunghezza massima del numero di caratteri che è possibile digitare all'interno del campo di testo (se non si omette questo attributo all'interno di un campo di testo sarà possibile scrivervi anche la Divina Commedia!)
- **SIZE** stabilisce la lunghezza del campo di testo (se si omette questo attributo ogni sistema operativo stabilirà una lunghezza di default, in genere una 20ina di caratteri).

Se il numero dei caratteri che posso inserire è molto alto (ossia se ho specificato un valore alto per l'attributo **MAXLENGTH** e se la lunghezza del campo di testo è minima (ossia se ho specificato un

²² Una query è una stringa tipo: `http://www.macchina.it/cgi-bin/programma?nome1=valore1&nome2=valore1+valore2` in cui la prima parte e' l'URL specificato in ACTION, seguita, dopo il "?" che indica l'inizio della query, da coppie nome=valore separate da "&"; quando uno stesso nome ha piu' valori, essi sono separati da "+".

valore basso per l'attributo SIZE il testo scritto all'interno della casella si potrà lo stesso scrivere solo che non potendo essere visibile tutto scomparirà verso sinistra a mano a mano che si digitano altri caratteri sulla destra)

- DISABLED fa rappresentare il campo al browser, ma impedisce all'utente di interagire con esso, disabilitandolo. Il browser rappresenta i campi disabilitati "screndoli" o cambiandone colore.

- ERROR indica il messaggio di errore da visualizzare se qualcosa va male.

1° ESEMPIO²³: Il tipo text e submit dell'attributo NAME del tag INPUT

Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es1-form-input-text.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il primo campo seguente:

```
<STRONG>Inserisci il tuo nome</STRONG> <BR>
<INPUT NAME="nome" TYPE="text" VALUE="il tuo nome" MAXLENGTH="20" SIZE="10">
```

Bene, inseriti il campo di testo, dobbiamo includere un bottone che consenta di inviare le informazioni appena introdotte all'autore del sito. Per il pulsante di invio si procede creando un altro elemento <INPUT> questa volta però metteremo come TYPE l'attributo submit che serve per creare il pulsante di invio, aggiungeremo anche VALUE per la scritta che dovrà comparire sul pulsante. Il codice da scrivere è:

```
<INPUT TYPE="submit" VALUE="Spedisci">
```

La pagina web es1-form.html nel browser produce il seguente risultato:

Inserisci il tuo nome

Osservazioni

- Nel piccolo campo di testo, di figura, poiché abbiamo dichiarato MAXLENGTH="20" posso inserire fino a 20 caratteri, chiaramente non riuscirò a visualizzarli tutti contemporaneamente, succederà dunque che (poichè size="10") dal 11° carattere in poi inserito i caratteri scorreranno verso sinistra (dando l'impressione di scomparire, ma non bisogna preoccuparsi perchè è solo un'impressione) per far posto ai nuovi, e questo finchè non sarà raggiunto il limite massimo di 20 caratteri che abbiamo fissato.

- Et voila! Ecco la nostra prima form! Beh, a voler essere pignoli si potrebbe notare che per come è stato costruito, questo form, non è un granchè, ad essere più precisi risulta del tutto inutile, difatti se si prova a schiacciare il bottone Spedisci si otterrà come risultato: il grande nulla! Il modulo infatti pur funzionando non sa ancora a chi inviare i dati appena raccolti e neppure in quale formato. E allora badiamo alla sostanza, ossia, una volta realizzata la mia piccola form come faccio a farmi arrivare queste informazioni via e-mail?

I metodi per ricevere le informazioni inserite on line dai visitatori

²³ Per semplicità d'ora in poi tralasceremo di scrivere, per ogni es. di form, la struttura completa dei tag dell'HTML, limitandoci solo a scrivere quello tra i tag <FORM> e </FORM>. Inoltre per convenzione scriveremo i nomi dei tag e quelli degli attributi maiuscoli e minuscoli valori degli attributi. In Html si può scrivere tutto maiuscolo, minuscolo o misto poiché esso non fa differenza. HAP-Edit ad es. scrive tutto in minuscolo ma per distinguere i tag dagli attributi e dai valori usa il colore nero in grassetto per i tag, il colore blu in grassetto per gli attributi e il colore celestino per i valori degli attributi.

- il primo è semplice e consiste nell'inviare i dati attraverso la posta elettronica;
- il secondo è più difficile e consiste nell'inviare i dati ad un webserver che per ora lo tralascieremo.

Inviare i dati raccolti via e-mail: uso degli attributi ACTION e METHOD

Procediamo dunque con il 1) metodo. E' ovvio che se inserisco un form all'interno del mio sito voglio fare in modo che queste informazioni, una volta inserite possano giungere all'autore del sito. Il metodo più semplice consiste nel spedire le informazioni via e-mail. Per fare questo bisogna aggiungere gli attributi **ACTION** e **METHOD**, suddetti, oltre all'attributo **NAME** opzionale, al tag di apertura.

L'attributo **ACTION** specificherà l'indirizzo di posta elettronica al quale voglio che siano inviate le informazioni raccolte, mentre l'attributo **METHOD**, che può assumere due diversi valori, ossia **GET** e **POST**, specificherà il modo in cui le informazioni devono essere inviate: ci basti sapere che, per questo primo metodo, è sufficiente usare il valore **POST**. Quindi in luogo del tag di apertura `<FORM>` dovrò scrivere:

(*) `<FORM ACTION="mailto: gabriele.gigliotti@lamezianet.it" METHOD="post">`

E' possibile aggiungere un soggetto all'e-mail? La risposta è sì. Basterà cambiare la riga (*) con la seguente:

`<FORM ACTION="mailto:gabriele.gigliotti@lamezianet.it?subject=...." METHOD="post">`

facendo, quindi, seguire all'indirizzo e-mail, la scritta `?subject=nome del soggetto`

2° ESEMPIO: Il tipo text, submit e reset dell'attributo NAME del tag INPUT

Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es2-form-input-submit-reset.html. Tra i tag `<BODY>` e `</BODY>` digitiamo la coppia di tag `<FORM>` e `</FORM>` e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

```
<STRONG>Inserisci il tuo nome</STRONG> <BR>
<INPUT NAME="nome" TYPE="text" VALUE="nome" MAXLENGTH="50" SIZE="5">
<STRONG>Inserisci il nome del tuo cantante preferito</STRONG><BR>
<INPUT NAME="cantante" TYPE="text" VALUE="cantante preferito">
<STRONG>Inserisci la tua e-mail</STRONG><BR>
<INPUT NAME="e-mal" TYPE="text" VALUE="e-mail" >
<INPUT TYPE="submit" VALUE="Spedisci">
```

Se riempiendo i campi del modulo ci siamo accorti di aver sbagliato a inserire delle informazioni abbiamo due possibilità in genere: o modificare le info sbagliate o cancellare tutte le info inserite e ripartire da capo. Ebbene, il tag `<INPUT>` prevede un tipo **RESET** che permette di ripulire il modulo ripristinando i valori di default. A tal fine aggiungiamo al codice scritto la seguente linea di codice:

```
<INPUT TYPE="reset" VALUE="Cancella">
```

La pagina web es2-form-input-submit-reset.html nel browser produce questo risultato:

Inserisci il tuo nome

Inserisci il nome del tuo cantante preferito

Inserisci la tua e-mail

3° ESEMPIO: Il tipo password e reset dell'attributo NAME del tag INPUT

Molto simile al TYPE=text è il TYPE=password. Si tratta esattamente della stessa cosa tranne per il fatto che vengono visualizzati degli asterischi al posto dell'input reale. Il browser poi ci spedisce i dati reali, soltanto che non li visualizzerà. Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es3-form-input-password.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

 Area Riservata

Login:

<INPUT NAME="login" TYPE="text" SIZE="20">

Password:

<INPUT NAME="pass" TYPE="password" SIZE="20">

<INPUT NAME="entra-pass" TYPE="submit" VALUE="entra">

<INPUT TYPE="reset" VALUE="Cancella">

Il codice a fianco produce il seguente risultato nel browser:

Area Riservata

Login:

Password:

Tale codice, come mostra la fig. a destra crea quattro campi. Un campo di tipo testo (denominato login), uno di tipo submit (denominato entra-pass), un terzo campo di tipo password (denominato pass) e un quarto campo di tipo reset.

Osservazione

Anche in tale es. non inserendo gli attributi ACTION e METHOD nel tag INPUT la pressione del pulsante entra non produce nulla.

4° ESEMPIO: Il tipo file dell'attributo NAME del tag INPUT

Il tipo file permette all'utente on line di inviare un file del suo PC sul webserver, inserendo il path assoluto del file da inviare nell'apposita casella di testo o cliccando sul pulsante sfoglia che permette di raggiungere il file richiesto sfogliando le varie cartelle e sottocartelle in cui il file è contenuto. Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es4-form-input-file.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

Inserisci nella casella di testo il percorso assoluto del file del tuo PC da inviare oppure clicca sul pulsante sfoglia

<INPUT name="file-hardisk" type="file" size="30">

Il codice a fianco produce il seguente risultato nel browser:

Inserisci nella casella di testo il percorso assoluto del file del tuo PC da inviare oppure clicca sul pulsante sfoglia

Come suddetto, anche qui l'attributo size indica la larghezza del campo della casella di testo.

5° ESEMPIO: Il tipo radio dell'attributo NAME del tag INPUT (per la scelta singola)

Il tipo radio permette all'utente on line di specificare una scelta singola tra diverse possibilità. Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es5-form-input-radio.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

```
1° <b> Come si chiama il tuo migliore amico? </b> <BR>
2° <INPUT NAME="il mio migliore amico" TYPE="radio" VALUE="marco "> Marco <BR>
3° <INPUT NAME=" il mio migliore amico" TYPE="radio" VALUE="pietro"> Pietro <BR>
4° <INPUT NAME=" il mio migliore amico" TYPE="radio" VALUE="luca"> Luca <BR>
5° <INPUT NAME=" il mio migliore amico" TYPE="radio" VALUE="stefano"> Stefano
```

Tale codice produce il seguente risultato nel browser:

Come si chiama il tuo migliore amico?

- ☐ Marco
- ☐ Pietro
- ☐ Luca
- ☐ Stefano

Osservazioni

- Il valore dell'attributo NAME in tutte e quattro le righe rimane sempre uguale, mentre il valore dell'attributo VALUE cambia sempre.²⁴ Più precisamente, il valore dell'attributo NAME nella 2°, 3°, 4° e 5° è sempre lo stesso e cioè "il mio migliore amico" in quanto, nel form, i tag INPUT non importa distinguerli visto che la scelta sarà sempre una sola.
- Gli attributi VALUE per ognuno dei quattro input hanno lo stesso valore del testo scritto fuori il tag <INPUT> e questo perché il testo fuori serve per essere visualizzato nella pagina web, mentre quello dentro vicino all'attributo VALUE è usato per trasmetterlo.
- La scelta effettuata dall'utente on line sarà spedita come coppia "valore di NAME = valore di VALUE". Ad es. supponendo che l'utente on line scelga la 1° opzione, cioè Marco, allora in tal caso l'informazione trasmessa sarebbe la coppia "il mio migliore amico = Marco" che è, quindi, l'unica informazione importante da annotare.
- Alla fine di ognuna delle cinque righe ci sono i tag
 per permettere di visualizzare nel browser le diverse scelte su linee diverse.
- Nella 1° riga abbiamo inserito la domanda a cui seguirà una sola risposta scelta tra le quattro possibili.

E' possibile specificare una scelta di default nell'elenco delle scelte? La risposta è sì.

Basta inserire la parola CHECKED all'interno del tag <INPUT> contenente la scelta di default. Ad es. supponendo che nell'esempio precedente la scelta di default sia Marco allora basterà solo modificare la 2° riga di codice con la seguente:

```
2° <INPUT NAME="il mio migliore amico" TYPE="radio" VALUE="Marco" CHECKED> Marco <BR>
```

Questa modifica produrrà il seguente risultato:

²⁴ Essa è una regola importante da ricordare poiché col tipo checkbox dell'attributo TYPE sarà esattamente il contrario.

Come si chiama il tuo migliore amico?

- ☒ Marco
☐ Pietro
☐ Luca
☐ Stefano

6° ESEMPIO: Il tipo checkbox dell'attributo NAME del tag INPUT (per la scelta multipla)

Il tipo checkbox permette all'utente on line di specificare più di una scelta tra diverse possibilità. Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es6-form-input-chekbox. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

- 1° Quali di questi ragazzi sono tuoi amici?

2° <INPUT NAME="marco" TYPE="checkbox" VALUE="si" CHECKED>Marco

3° <INPUT NAME="pietro" TYPE="checkbox" VALUE="si">Pietro

4° <INPUT NAME="luca" TYPE="checkbox" VALUE="si">Luca

5° <INPUT NAME="stefano" TYPE="checkbox" VALUE="si">Stefano

Tale codice produce il seguente risultato nel browser:

Quali di questi ragazzi sono tuoi amici?

- ☒ Marco
☐ Pietro
☐ Luca
☐ Stefano

Osservazioni

- Il valore dell'attributo NAME in tutte e quattro le righe cambia sempre, mentre il valore dell'attributo VALUE rimane sempre lo stesso. Più precisamente, i valori dell'attributo NAME nella 2°, 3°, 4° e 5° cambia sempre in quanto, nel form, i tag INPUT importa distinguerli visto che la scelte dell'utente on line possono essere anche tutte quelle disponibili.²⁵
- Nella 1° riga abbiamo inserito la domanda a cui seguirà una o più risposte scelte tra le quattro possibili.
- Gli attributi VALUE per ognuno dei quattro input hanno valore diverso rispetto al testo scritto fuori il tag <INPUT> e questo perché il testo fuori serve per essere visualizzato nella pagina web, mentre quello dentro vicino all'attributo VALUE è usato per trasmetterlo.
- Le scelte, da una a quattro in tal caso, effettuate dall'utente on line saranno spedite come coppia "valore di NAME = valore di VALUE". Ad es. supponendo che l'utente on line scelga la 1° opzione, cioè Marco, e la 4° opzione, cioè Stefano, allora in tal caso l'informazione trasmessa sarebbe la coppia "Marco=si" e la coppia "Stefano=si" che sono, quindi, le uniche informazioni importanti da annotare.
- Alla fine di ognuna delle cinque righe ci sono i tag
 per permettere di visualizzare nel browser le diverse scelte su linee diverse.


²⁵ In un singolo form, per il tipo checkbox, è bene che non si ripetano mai gli stessi valori NAME. Quando dico mai, non intendo che se lo farai il tuo computer si pianterà irrimediabilmente, alla peggio potrebbe incasinarsi il browser, o il parser (Mailto Formatter è un parser), o lo script CGI o ASP o PHP, o forse il "poveraccio" che dovrà cavare un senso logico dal form che ha ricevuto.

- Nella 1° riga abbiamo inserito la domanda a cui seguirà una o più risposte scelte tra le quattro possibili.
- La scelta del valore “si” per l’attributo VALUE è a discrezione del webmaster. Si avrebbe potuto anche scegliere: “yes”, “1”, “on” ecc.
- Come nell’es. 3 anche qui è possibile specificare una o più scelte di default che l’utente on line può liberamente cambiare, attraverso l’inserimento della parola CHECKED all’interno del tag <INPUT> contenente la scelta di default. Nel nostro es. noi l’abbiamo inserita nella 2° riga.

Creazione dei tag INPUT in modo visuale con HAP-Edit


Adesso che abbiamo capito bene la sintassi del tag INPUT scrivendola carattere per carattere, per velocizzare la scrittura del codice conviene, secondo me, utilizzare per le prossime esercitazioni le procedure visuali del nostro editor HAP-Edit.

Aperto HAP-Edit ☐ Nuova pagina html ☐ tra i tag <BODY> e </BODY> inserire la coppia di tag <FORM> e </FORM> ☐ cliccare sul pulsante  e poi:

1) **per inserire un tag INPUT con attributo TYPE uguale a text o a password** basta cliccare sull’opzione  Inserisci campo di testo del menù a discesa che compare. Essa mostra la finestra seguente:



Nelle righe Nome e Fornisci un valore bisogna inserire il nome e il valore da associare rispettivamente all’attributo NAME e all’attributo VALUE del tag INPUT. Nella parte sinistra è possibile specificare la larghezza da associare all’attributo SIZE del tag INPUT (per default è settato il valore 20 come già detto in precedenza). Nella parte destra in basso è possibile specificare il valore da associare all’attributo TYPE scelto tra testo e password del tag INPUT. Tralasciamo per ora l’uso delle opzioni “Disabled”²⁶ e “ReadOnly”.


2) **per inserire un tag INPUT con attributo TYPE uguale a submit o a reset** basta cliccare sull’opzione  Inserisci pulsante del menù a discesa che compare. Essa mostra la finestra seguente:

²⁶ Disabled corrisponde all’attributo omonimo del tag INPUT. Esso, se selezionato, fa rappresentare il campo al browser, ma impedisce all’utente di interagire con esso, disabilitandolo. Il browser rappresenta i campi disabilitati “scurandoli” o cambiandone colore.

The dialog box 'Componente del modulo' has a blue title bar with a close button. It contains the following fields and options:

- Nome:** A text input field.
- Fornisci un valore:** A checked checkbox followed by a text input field containing the word 'Invia'.
- Options:** Three radio buttons: 'Pulsante "Invia"' (selected), 'Pulsante "Reimposta"', and '"Disabled"'.
- Buttons:** 'Ok', 'Tutti...', and 'Annulla' on the right side.

Nelle righe Nome e Fornisci un valore bisogna inserire il nome e il valore da associare rispettivamente all'attributo NAME e all'attributo VALUE del tag INPUT. Nella parte sinistra in basso è possibile specificare il valore da associare all'attributo TYPE scelto tra submit (Pulsante "Invia") e reset (Pulsante "Reimposta") del tag INPUT. Tralasciamo per ora l'uso dell'opzione "Disabled" già spiegata.

3) **per inserire un tag INPUT con attributo TYPE uguale a radio** basta cliccare sull'opzione  Inserisci pulsante di scelta del menù a discesa che compare. Essa mostra la finestra seguente:

The dialog box 'Componente del modulo' has a blue title bar with a close button. It contains the following fields and options:

- Nome:** A text input field.
- Fornisci un valore:** A checked checkbox followed by an empty text input field.
- Options:** Two checkboxes: 'Radio button selezionato' and '"Disabled"'.
- Buttons:** 'Ok', 'Tutti...', and 'Annulla' on the right side.

Nelle righe Nome e Fornisci un valore bisogna inserire il nome e il valore da associare rispettivamente all'attributo NAME e all'attributo VALUE del tag INPUT. Nella parte sinistra in basso è possibile specificare se l'opzione corrente deve essere di default mettendo un check alla voce Radio Button selezionato. Tralasciamo per ora l'uso dell'opzione "Disabled" già spiegata.

4) **per inserire un tag INPUT con attributo TYPE uguale checkbox** basta cliccare sull'opzione  Inserisci casella di controllo del menù a discesa che compare. Essa mostra la finestra seguente:

The dialog box 'Componente del modulo' has a blue title bar with a close button. It contains the following fields and options:

- Nome:** A text input field.
- Fornisci un valore:** A checked checkbox followed by an empty text input field.
- Options:** Two checkboxes: 'Checkbox selezionato' and '"Disabled"'.
- Buttons:** 'Ok', 'Tutti...', and 'Annulla' on the right side.

Nelle righe Nome e Fornisci un valore bisogna inserire il nome e il valore da associare rispettivamente all'attributo NAME e all'attributo VALUE del tag INPUT. Nella parte sinistra in basso è possibile specificare se l'opzione corrente deve essere di default mettendo un check alla voce Checkbox selezionato. Tralasciamo per ora l'uso dell'opzione "Disabled" già spiegata.

Il tag <SELECT>

SELECT e' utilizzato per creare menù con scelte singole o multiple. E' generalmente usato come un menù a tendina e rappresenta un'alternativa più compatta rispetto ai radio-buttons per le scelte singole, o ai checkboxes per le scelte multiple.

Esso possiede il tag di chiusura </SELECT> e contiene al suo interno una coppia di tag <OPTION> </OPTION> per ognuna delle possibili scelte indicate. Esso possiede come principali attributi seguenti:

- NAME specifica il nome del menu'.
- SIZE specifica quante opzioni devono essere visualizzate assieme nella finestra. Per default è specificato il valore di size =1 necessario per creare un menù a discesa. Per valori di size > 1 il menù a tendina dovendo visualizzare più opzioni si trasforma in una lista scorrevole.
- MULTIPLE, SELECTED e DISABLED sono tre attributi che non possiedono valori. MULTIPLE abilita le scelte multiple attivabili da tastiera con il tasto CTRL, se inserito all'interno del tag di apertura <SELECT>. SELECTED seleziona l'opzione di default, cioè quella che deve essere visualizzata per prima nel menù a discesa, se inserito all'interno del tag di apertura <OPTION>.
- DISABLED fa in modo che il menu' a discesa creato non possa essere modificato dall'utente e resta appunto disabilitato se se inserito all'interno del tag di apertura <SELECT>.
- ERROR specifica il messaggio di errore da visualizzare se qualcosa va male.

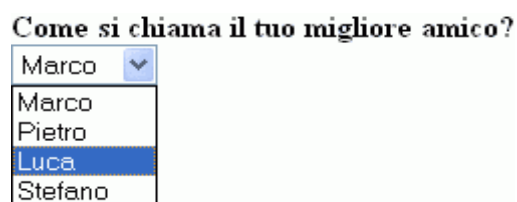
IL tag <OPTION> possiede come unico attributo VALUE. Esso contiene il valore usato per trasmetterlo.

7° ESEMPIO: Tag Select e gli attributi NAME, OPTION, VALUE, SIZE=1 (selezione singola)

Aperto HAP-Edit □ Nuova pagina Html □ salviamola con il nome es7-form-select.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

```
<b> Come si chiama il tuo migliore amico? </b> <BR>
<SELECT NAME="amico" SIZE="1">
  <OPTION VALUE="Marco"> Marco
  <OPTION VALUE="Pietro">Pietro
  <OPTION VALUE="Luca">Luca
  <OPTION VALUE="Stefano">Stefano
</SELECT>
```

Il codice a fianco produce il seguente risultato nel browser:



8° ESEMPIO: Tag Select e NAME, OPTION, VALUE, SIZE=3 (selezione multipla), SELECTED

Aperto HAP-Edit □ Nuova pagina Html □ salviamola con il nome es8-form-select-multiplo.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

```
<b> Come si chiamano i tuoi migliori amici? </b> <BR>
<SELECT MULTIPLE NAME="amici" SIZE=4>
```



Il codice a fianco produce il seguente risultato nel browser:

```
<OPTION VALUE="Marco">Marco  
<OPTION VALUE="Pietro">Pietro  
<OPTION VALUE="Luca">Luca  
<OPTION VALUE="Stefano">Stefano  
<OPTION VALUE="Guido" SELECTED>Guido  
<OPTION VALUE="Alfredo">Alfredo  
<OPTION VALUE="Enrico">Enrico  
</SELECT>
```

Come si chiamano i tuoi migliori amici?

Creazione del tag SELECT in modo visuale con HAP-Edit

Adesso che abbiamo capito bene la sintassi del tag INPUT scrivendola carattere per carattere, per velocizzare la scrittura del codice conviene, secondo me, utilizzare per le prossime esercitazioni le procedure visuali del nostro editor HAP-Edit.

Aperto HAP-Edit ☐ Nuova pagina html ☐ tra i tag <BODY> e </BODY> inserire la coppia di tag <FORM> e </FORM> ☐ cliccare sul pulsante  e poi basta cliccare sull'opzione  Inserisci elenco/menu del menù a discesa che compare. Essa mostra la finestra seguente:

Il tag <TEXTAREA>

Esso permette all'utente di inserire più di una linea di testo. Esso possiede il tag di chiusura </TEXTAREA> e possiede i seguenti attributi: ROWS (numero righe visibili nell'area) e COLS (numero colonne visibili nell'area) oltre ai già menzionati NAME, DISABLED e ERROR. Il testo tra i tag di apertura e chiusura serve ad inizializzare il valore del campo con un valore di default.

9° ESEMPIO: Tag <TEXTAREA>

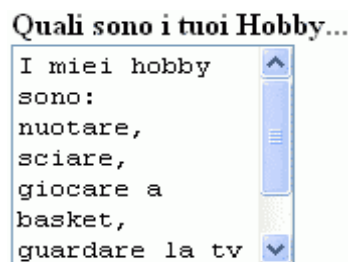
Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es9-form-textarea.html. Tra i tag <BODY> e </BODY> digitiamo la coppia di tag <FORM> e </FORM> e posizionandoci al centro tra essi diamo qualche invio per creare qualche riga vuota. Dopodichè inseriamo il codice seguente:

```
<b>Quali sono i tuoi Hobby... </b> <BR>  
<TEXTAREA NAME="commenti" ROWS=15 COLS=65>
```

I miei Hobby sono: nuotare, sciare, giocare a basket,
guardare la tv



```
</TEXTAREA>
```

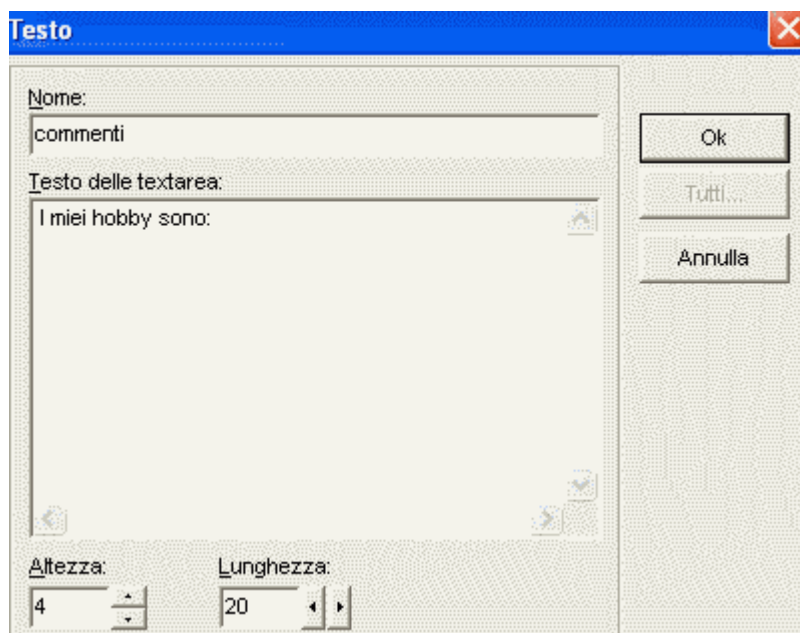
Il codice a fianco produce il seguente risultato nel browser:



Creazione del tag <TEXTAREA> in modo visuale con HAP-Edit

Adesso che abbiamo capito bene la sintassi del tag INPUT scrivendola carattere per carattere, per velocizzare la scrittura del codice conviene, secondo me, utilizzare per le prossime esercitazioni le procedure visuali del nostro editor HAP-Edit.

Aperto HAP-Edit ☐ Nuova pagina html ☐ tra i tag <BODY> e </BODY> inserire la coppia di tag <FORM> e </FORM> ☐ cliccare sul pulsante  e poi basta cliccare sull'opzione  Inserisci area di testo del menù a discesa che compare. Essa mostra la finestra seguente:



10° ESEMPIO: Creazione di un questionario i cui dati sono spediti al webmaster via e-mail

Utilizza il tag <INPUT> per semplici campi di testo di una sola linea, il tag <TEXTAREA> per immissioni multilinea e in più ci sono tutti gli altri elementi di form visti, in congiunzione con i classici tag HTML. Aperto HAP-Edit ☐ Nuova pagina Html ☐ salviamola con il nome es10-form-questionario.html. Dopodichè inseriamo il codice seguente:

```
<body>  
<H2>Questionario on line</H2>  
  
<P>Riempi i campi seguenti:  
  
<FORM METHOD=post ACTION="mailto:ascuoladi@yahoo.it">  
  
<P>Il tuo Nome: <input name="nome" size="20">
```

```
<P><input name="maschio" type="radio"> Uomo
<P><input name="femmina" type="radio"> Donna
<P> Componenti della famiglia: <input name="famiglia" type="text" size="1">
<P>Titolo di studio:
<UL>
<LI><input name="titolo-1" type="checkbox" value="licenza media"> Licenza Media
<LI><input name="titolo-2" type="checkbox" value="Diploma"> Diploma <P>
<LI>Altre <textarea name="altri-titoli" cols="20" rows="4"></textarea>
</UL>
<P>Nickname: <INPUT NAME="nickname" size="20">
<P>Grazie per aver risposto al questionario! <P>
<INPUT TYPE="submit" VALUE="Invia"> <INPUT TYPE="reset" VALUE="cancella">

</FORM>
</body>
```

Il codice di sopra produce il seguente risultato nel browser:

Questionario on line

Riempi i campi seguenti:

Il tuo Nome:

☒ Uomo

☐ Donna

Componenti della famiglia:

Titolo di studio:

- ☐ Licenza Media
- ☐ Diploma
- Altre

Nickname:

Grazie per aver risposto al questionario!