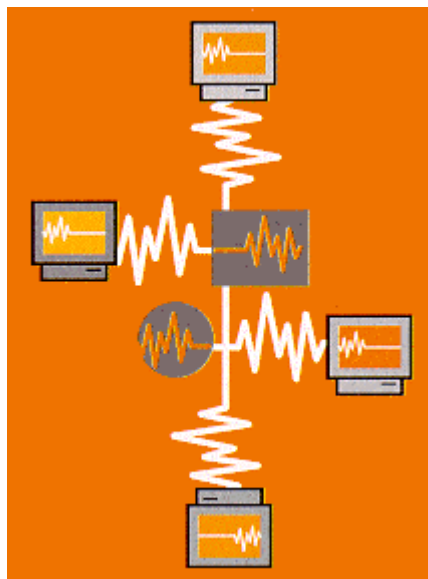


Guida HTML



Progetto di formazione

Tele-PMI

Iniziativa Comunitaria ADAPT/bis

1 Indice

1	Indice	2
2	Introduzione	3
3	Il Testo	4
3.1	<i>Grandezza</i>	5
3.2	<i>Formattazione</i>	5
3.3	<i>Allineamento</i>	5
3.4	<i>Colori</i>	5
3.5	<i>Internazionalizzazione</i>	6
4	Le Immagini	6
5	I Collegamenti	8
6	Le Liste	9
7	Le Tabelle	10
8	Frame	13
8.1	<i>Frame in linea</i>	15
9	Form	16
10	Fogli di Stile (CSS)	18

2 Introduzione

Molte persone si domanderanno come possono fare per realizzarne una propria Home Page. Ecco quindi una piccola guida per creare le proprie pagine.

Per far questo c'è bisogno di:

- Conoscere il linguaggio HTML cioè il linguaggio che sta alla base di tutti i documenti che girano nel WWW.
- Avere uno spazio WEB: infatti voi dopo aver creato le proprie pagine c'è bisogno di metterle sulla rete. Per far questo bisogna parlare con il Webmaster e chiedergli di pubblicarle. Infatti solitamente i provider riservano uno spazio nel server dedicato proprio alle pagine dei loro utenti.

La più grande caratteristica dei documenti HTML è l'essere multiplatforma. Infatti il vostro file che create su PC e mettete in rete, potrà essere letto anche da utenti di Macintosh o di UNIX.

Per realizzare documenti HTML sono in circolazione per la rete una quantità spropositata di HTML editor; inizialmente, però, è meglio prendere dimestichezza con il linguaggio imparando a conoscerlo. Potete utilizzare invece un programma che avete già nel vostro computer: l'editor del DOS o il Blocco Note (alias notepad) di WINDOWS. Create con uno di questi programmi un nuovo file, seguendo le istruzioni contenute in questo documento, realizzate il vostro documento HTML e ricordatevi alla fine di salvare il file non con l'estensione .TXT ma .HTM (se usate il DOS o Win3.1x) o .HTML (se usate Windows 95). A questo punto caricate Netscape o il vostro browser preferito e, dal menù File, scegliete Open File o Open Local File e lì inserite il percorso del vostro documento (es. C:\Internet\HTML\miofile.htm).

Nel linguaggio HTML (HyperText Markup Language) i comandi (detti TAG) vengono immessi tra le parentesi "<" e ">". Normalmente tutti i comandi cominciano con <comando> e finiscono con </comando>. E' possibile inserire i comandi sia in lettere maiuscole che in lettere minuscole, quindi hanno lo stesso effetto <COMANDO> e <comando>.

Tutti i documenti HTML iniziano con <HTML> e finiscono con </HTML>.

Vengono poi suddivisi in testa (<HEAD>, </HEAD>) e corpo (<BODY>, </BODY>).

All'interno della testa bisogna inserire il titolo <TITLE> , </TITLE>; non c'è limite per la lunghezza del titolo, ma consiglio di non farlo troppo lungo perché potrebbe non essere visualizzato

bene nella barra del titolo del browser. Sempre all'interno del tag HEAD vanno inseriti, se ce ne fossero, anche i fogli di stile (CSS – Cascading Style Sheets).

All'interno del corpo va inserito tutto quello che volete contenga il vostro documento, come ad esempio:

- Testo
- Immagini
- Collegamenti ad altri documenti
- Liste
- Frame
- Form
- Fogli di stile

A questo punto possiamo fare un piccolo schema riassuntivo su quanto appena visto. Questa è la bozza di un documento HTML:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Titolo del Documento </TITLE>
```

```
<BODY>
```

Questo è il corpo del documento dove il corpo del documento.

```
</BODY>
```

```
</HTML>
```

Un documento HTML ha due facce; infatti il documento che si realizza con l'editor, contenente i vari tag, viene poi interpretato dal browser che lo visualizza senza i tag.

3 Il Testo

Un documento HTML molte volte può sembrare «piatto» per questo sono stati introdotti dei TAG per dare più stile alle nostre parole.

3.1 *Grandezza*

Possiamo aumentare la grandezza del testo semplicemente racchiudendolo tra il tag <H1> e </H1>. E' possibile sostituire a 1 un numero che varia da 1 a 6 ricordandosi che con 1 il carattere sarà più grande e con 6 più piccolo.

3.2 *Formattazione*

E' possibile formattare il testo in vari modi. Possiamo renderlo:

- Corsivo con il tag <I> e </I>
- Grassetto con il tag e
- Intermittente con il tag <BLINK> e </BLINK>
- TASTIERA con il tag <KBD> e </KBD>

3.3 *Allineamento*

Possiamo allineare il testo che abbiamo racchiuso nel tag <H?> e </H?> (dove ? è un numero da 1 a 6). Inserendo <H? ALIGN=CENTER> e </H?> il testo verrà centrato. A CENTER è possibile sostituire LEFT per l'allineamento a sinistra o RIGHT per quello a destra.

3.4 *Colori*

E' possibile impostare il colore del testo e di sfondo nella maniera più semplice attraverso l'attributo BGCOLOR dell'elemento BODY precisamente <BODY BGCOLOR="#value" ...BACKGROUND="nomefile.gif"> usando i nomi dei colori o i codici RGB con cifre esadecimali:

```
<BODY BG COLOR="#000000" TEXT="#FFFFFF" LINK="#CC0000" ALINK="#777777"
VLINK="#990099" BACKGROUND="nomefile.gif">
```

Tuttavia, l'HTML 4.0 sconsiglia tale uso di questo attributo. Usando un foglio di stile (i fogli di stile forniscono alcuni vantaggi agli autori Web e risultano particolarmente potenti quando vengono combinati con gli script), si può ottenere lo stesso effetto con le seguenti definizioni:

```
BODY {background: #000 url (nomefile.gif)}
```

```
BODY {color: #FFF}
```

```
A: link {color: #C00}
```

A: visited {color: #555}

A: active {color: #909}

In questo caso i colori sono espressi usando tre cifre RGB invece di sei cifre RGB.

3.5 *Internazionalizzazione*

La maggior parte dei documenti HTML dall'avvento del WWW è stata scritta in inglese. Con la diffusione dell'uso del Web sono iniziati ad apparire documenti scritti in altre lingue. Poiché ci troviamo lingue diverse che usano caratteri differenti, le nuove specifiche dell'HTML 4.0 offrono ora una maggiore flessibilità nella visualizzazione e nella specifica delle varie lingue del pianeta. L'HTML 4.0 attraverso l'uso dell'attributo LANG specifica la lingua usata per scrivere un documento nel tag <HTML> all'inizio del documento.

```
<HTML LANG= "IT">
```

ove «IT» sta per Italian (italiano).

Un altro attributo relativo alle lingue da utilizzare nei tag è DIR, che specifica la direzione che il browser dovrà usare per visualizzare il testo. DIR= "LTR" da sinistra a destra (impostazione standard), mentre DIR= "RTL" da destra a sinistra. Per capirci meglio se vogliamo specificare un documento scritto in ebraico e che venga visualizzato da destra a sinistra, l'esempio è il seguente:

```
<HTML LANG= "HE" DIR= "RTL">
```

E' inoltre possibile inserire delle righe orizzontali con <HR> per segnare la fine di un sezione nel documento. Non ha bisogno di una istruzione finale.

Se, invece, volete andare a capo o lasciare delle righe bianche non dovete far altro che inserire, una o più volte il tag
, tuttavia, l'HTML 4.0 consiglia di evitare l'uso di elementi di stile come questi, proponendo come alternativa i fogli di stile, che forniscono un maggiore controllo sull'aspetto dei documenti e una maggiore flessibilità, dato che non interferiscono con il contenuto della pagina.

4 Le Immagini

E' possibile inserire immagini nei documenti HTML. Attenzione: le immagini possono arricchire la vostra pagina e rendere più gradevole la visione di un documento ma, rallentano di MOLTO la navigazione a causa del tempo(a volte anche molto lungo !!) che ci vuole per caricarle.

Quindi usatele con molta cautela; spetta a voi saperle gestire e calibrare in modo che non siano d'intralcio con la navigazione.

I formati per le immagini più utilizzati sono il GIF (.GIF) e il JPEG (.JPG). Per inserire delle immagini il tag è ``. Visto che non tutti i navigatori utilizzano l'opzione di caricare in automatico le immagini (sempre per cercare una maggior velocità) potreste fargli una cosa gradita dicendogli il "titolo" dell'immagine o comunque il suo contenuto.

Con l'HTML 4.0 l'attributo ALT è diventato obbligatorio. Per far questo dovete leggermente modificare il TAG precedente in ``. Con `` il testo immediatamente successivo all'immagine verrà allineato in alto. Se a TOP sostituiamo BOTTOM verrà allineata in basso; se invece usiamo MIDDLE in mezzo.

A volte si potrebbe aggiungere un oggetto multimediale come suoni, filmati, direttamente in un documento HTML. L'HTML 4.0 permette di inserire questi oggetti con il nuovo elemento oggetto `<OBJECT>` il quale può sostituirsi al tag `` nell'inserimento delle immagini nei documenti. L'elemento oggetto deve iniziare con il tag `<OBJECT>` in cui si possono inserire gli attributi che servono per specificare la posizione del file multimediale (attributo DATA) e il tipo di file che si vuole includere (attributo TYPE). Dopo il tag `<OBJECT>` si deve inserire il testo alternativo oppure un'immagine nel caso in cui gli oggetti in questione non siano supportati dal browser. L'elemento oggetto deve essere quindi chiuso con il tag di fine `</OBJECT>`. Se il browser utilizzato riconosce l'elemento oggetto e dispone, per esempio, di un plug-in per filmati QuickTime, il video "pinco pallino" verrà riprodotto sullo schermo. Se il browser non riconosce l'elemento oggetto e non dispone del plug-in specifico, verrà visualizzato del testo alternativo. Al posto del testo alternativo si può mettere un'immagine fissa se il browser non supporta il formato QuickTime. Con il codice seguente il tag `` specifica l'immagine da visualizzare al posto del filmato QuickTime:

```
<OBJECT DATA="nomefile.mov" TYPE="video/quicktime">
```

```
<IMG SRC="nomefile.gif" ALT="[ descrizione filmato oppure mettere il nome del video]">
```

```
</OBJECT>
```

Tuttavia, la maggior parte dei plug-in che gestisce un certo tipo di oggetto, come file MIDI, richiede dei parametri. Per esempio, un oggetto MIDI o un video non vengono normalmente riprodotti automaticamente, ma in conseguenza a un comando impartito dall'utente. Comunque, in certi casi si può cambiare il comportamento del plug-in usando un parametro. Un parametro è costituito da un nome di proprietà e da un valore che può cambiare il comportamento del plug-in.

Se prendiamo come esempio il parametro "AUTOSTART", questo determina se un oggetto deve essere eseguito automaticamente. Per attivare un parametro, si deve includere l'elemento PARAM nell'elemento oggetto. L'elemento PARAM è un elemento vuoto, del tipo:

```
<PARAM NAME="nome parametro" VALUE="valore parametro">
```

Per esempio, il plug-in LiveAudio di Netscape non riproduce automaticamente un file audio. Quindi, per riprodurre il file si può attivare il parametro AUTOSTART:

```
<OBJECT DATA="nomefile.mid" TYPE="audio/midi">
```

```
<PARAM NAME="AUTOSTART" VALUE="TRUE">
```

```
[musica di sottofondo]
```

```
</OBJECT>
```

Il rovescio della medaglia in tutto questo è che non tutti i browser gestiscono l'elemento oggetto (è ancora nuovo) nel caricamento delle immagini in un documento HTML, e quelli che lo possono fare attualmente pongono diversi problemi. Per cui è consigliabile usare l'elemento immagine ancora più semplice per inserire immagini nel documento HTML.

5 I Collegamenti

Una delle caratteristiche principali del documento HTML sono i collegamenti. Infatti alcune parole, come avrete notato, sono sottolineate e cliccandoci sopra vi possono portare in un qualsiasi punto del WEB. E' possibile, partendo da una qualsiasi pagina e solo usando i collegamenti, spostarsi da una parte all'altra del mondo senza nemmeno che ve ne possiate accorgere.

Il tag per inserire un collegamento è e . Quello che verrà scritto tra questi due TAG diventerà sottolineato e cliccandoci sopra vi porterà alla pagina stabilita. Se la pagina a cui volete fare il collegamento si troverà nella stessa directory della prima pagina basterà inserire il nome del file. In caso contrario, se volete fare il collegamento a qualcos'altro al di fuori della vostra directory, dovrete inserire il percorso completo.

Esempio. Se volete fare il collegamento al sito di Netscape il tag sarà:

```
<A HREF="http://www.netscape.com">Server di Netscape</A>
```

Oltre che ad un server HTTP è possibile fare un collegamento a:

FTP

La sigla FTP sta per File Transfer Protocol ed è il protocollo dedicato alla trasmissione di FILE nella rete. Quindi sono possibili collegamenti di questo tipo:

```
<A HREF="ftp://ftp.protos.it">Server FTP di Protos</A>
```

MAIL TO

Tradotto in italiano sarebbe «posta a». Viene data la possibilità di spedire un messaggio ad un dato destinatario. Se si volesse spedire posta all'indirizzo menghini@geocities.com il tag da usare sarebbe:

```
<A HREF="mailto:menghini@geocities.com">menghini@geocities.com</A>
```

6 Le Liste

Vi potrà capitare molto spesso mentre scrivete documenti HTML di aver bisogno di fare una lista o un elenco di qualcosa: le vostre pagine preferite, l'elenco dei CD che ascoltate con più piacere, etc, etc ...

L'HTML ha pronti dei tag proprio per questo scopo. L'argomento potrebbe essere molto lungo e complesso; ci limitiamo a dirvi che le liste principali sono di due tipi:

- Liste non ordinate
- Liste numerate

In tutti i tipi di liste le istruzioni sono tre:

1. Istruzione di inizio lista

2. Elemento della lista

3. Istruzione di conclusione lista

Nelle liste non ordinate, dette anche libere, i tag di inizio e fine sono `` e ``, mentre gli elementi della lista sono identificati con ``. Se inseriamo `<UL TYPE=DISC >` potremo sostituire i simboli della lista con dei "dischi". Se a DISC sostituiamo CIRCLE avremo dei cerchi, se invece utilizziamo SQUARE avremo dei quadrati. Con `<UL SRC="ball.gif">` è possibile sostituire i "puntini" della lista con delle immagini.

Nelle liste numerate, i tag di inizio e fine sono `` e ``, mentre gli elementi della lista sono identificati sempre con ``. Anche per le liste numerate è possibile modificare il tipo di numerazione attraverso l'attributo TYPE= con i seguenti valori: I per i numeri romani maiuscoli, i per quelli minuscoli, A per le lettere maiuscole e a per quelle minuscole.

7 Le Tabelle

All'interno di un documento HTML è possibile inserire dei dati di qualunque tipo (numeri, nomi, percentuali) all'interno di tabelle. Il tag che identifica l'inizio di una tabella è **<table>**, accompagnato sempre al tag **</table>** che indica invece la fine della suddetta tabella. Il tag `table` prevede una opzione *BORDER* che ci consente di definire la grandezza del bordo che deve essere associato alla tabella. Tale grandezza va rappresentata con un numero positivo intero: ponendo **<table border =0>** si intende una tabella priva di bordo.

All'interno della tabella stessa saranno presenti altri tag che mi permetteranno di inserire i dati nei vari campi della tabella stessa.

Questi sono:

<TR></TR> che identificano inizio e fine di ogni riga della tabella

<TH></TH> che identificano l'intestazione della tabella. Viene associato questo tag ad ogni termine utilizzato per identificare un insieme di dati (riga o colonna) inseriti nella tabella

<TD></TD> che identificano ogni dato singolo che è stato inserito della tabella.

Applicando le semplici regole ora introdotte è possibile realizzare una tabella come quella sottostante:

Studenti	maschi	femmine
Italiani	120	65
Stranieri	17	13
Totale	137	78

Essa è stata realizzata utilizzando le seguenti istruzioni:

```
<table border=2>
```

```
<tr>
```

```
<th>Studenti</th><th>maschi</th><th>femmine</th>
```

```
<tr>
```

```
<th>Italiani</th><td>120</td><td>65</td>
```

```
<tr>
<th>Stranieri<th><td>17</td><td>13</td>
<tr>
<th>Totale<th><td>137</td><td>78</td>
</table>
```

Vediamo quali sono gli attributi dell'elemento TABLE:

ALIGN

Specifica la posizione della tabella relativamente ai margini della finestra; i possibili valori sono LEFT (default), RIGHT e CENTER.

WIDTH

Specifica la grandezza della tabella in pixel o in percentuale (rispetto alla grandezza della finestra).

BORDER

Indica lo spessore del bordo della tabella; per non avere bordo si scriverà BORDER=0.

CELLSPACING

Specifica, in pixel, lo spazio tra le celle.

CELLPADDING

Specifica, in pixel, lo spazio tra il contenuto di una cella ed il suo bordo.

Analizziamo adesso quali sono gli attributi dell'elemento TR:

ALIGN

Posiziona, orizzontalmente, il contenuto all'interno di una cella; i possibili valori sono LEFT, CENTER e RIGHT.

VALIGN

Posiziona, verticalmente, il contenuto all'interno di una cella; i possibili valori sono TOP, MIDDLE e BOTTOM.

Per mostrare l'uso dei principali attributi descritti, possiamo creare le seguenti tabelle:

Nome	Paolino Paperino
Indirizzo	Via del Sicomoro, 254 Paperopoli

utilizzando questa sequenza di comandi

```
<TABLE BORDER =2 WIDTH=70%>
<TR>
<TH width=30%> Nome </TH>
<TD width=70%> Paolino Paperino</TD>
<TR>
<TH WIDTH=30%> Indirizzo </TH>
<TD WIDTH=70%> Via del Sicomoro, 254 Paperopoli</TD>
</TABLE>
```

oppure la seguente tabella:

Questa è una tabella di prova per il corso di Poggibonsi

1	1	1	1
---	---	---	---

utilizzando questa sequenza di comandi:

```
<TABLE BORDER=2 CELLSPACING=10 CELLPADDING=10>
<TR ALIGN=LEFT VALIGN=MIDDLE>
<TD COLSPAN=5 NOWRAP>Questa è una tabella di prova per il corso di Poggibonsi
<TR ALIGN=LEFT VALIGN=MIDDLE>
<TD>1
<TD>1
<TD>1
<TD ALIGN=RIGHT>1
```

```
<TD>1
</TABLE>
```

8 I Frame

Un'altra delle grosse innovazioni delle ultime versioni dell'HTML sono i frame (supportati da Netscape v. 2 e successive e Internet Explorer v. 3 e successive). In pratica si tratta di documenti divisi al loro interno in sotto documenti totalmente indipendenti fra loro; i frame dividono infatti la finestra del documento in più aree logiche, contenenti file HTML, aggiornabili separatamente, permettendo perciò la creazione di documenti molto complessi e spesso molto funzionali (sempre che l'utente disponga di un browser adatto alla loro visualizzazione). Ma attenzione perché l'abuso dei frame può dare origine a risultati d'aspetto grafico indesiderati. Per rendere il tutto ancora più versatile, si possono includere dei FRAMESET al FRAMESET principale. I documenti contenenti frame sono di un tipo un po' particolare in quanto all'interno del loro sorgente non è più presente il tag <BODY> e </BODY>, ma un sostitutivo <FRAMSET> e </FRAMSET>. Il documento base si presenterà quindi così:

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

Qui puoi metterci tutto il contenuto dei frame attraverso i tag descritti in avanti

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

All'interno del tag FRAMSET potranno essere presenti solo questi tag: <FRAME>, <FRAMESET>, <NOFRAMES>. Con le ultime disposizioni dell'HTML 4.0 è stato aggiunto un nuovo elemento chiamato IFRAME il quale permette di disporre di frame interni alla pagina, ma questo lo vedremo dopo. Vediamo adesso la sintassi:

<FRAMESET> viene personalizzato con il seguente schema:

```
<FRAMSET ROWS= 'value' COLS= 'value' BORDER= 'value' BORDERCOLOR=
'#xxxxxx' FRAMEBORDER= 'value' >
```

i cui flag corrispondono a:

ROWS è una lista di valori separati da virgole, che definiscono la dimensione delle divisioni verticali della pagina, i valori possono essere espressi in percentuale, in pixel, ed un * (detto frame relativa), che occuperà il restante spazio se la somma dei pixel non è sufficiente a coprire l'ampiezza della pagina o la somma percentuale è inferiore a 100%;

- COLS funziona come ROWS, ma divide la pagina in colonne;
- BORDER definisce lo spessore del bordo in pixel (Solo Netscape);
- BORDERCOLOR definisce il colore del bordo (Solo Netscape);
- FRAMEBORDER definisce lo spessore globale per i bordi che separano i frame (Solo Netscape).

Una volta definito il FRAMESET si definiscono all'interno i vari frame e le loro caratteristiche con il tag <FRAME>. L'esatta sintassi per il tag <FRAME> è:

```
<FRAMESET>
```

```
<FRAME...>
```

```
</FRAMESET>
```

```
<FRAME SRC="url" NAME="nome" MARGINWIDTH="value"
MARGINHEIGHT="value" SCROLLING="yes-no-auto" NORESIZE FRAMEBORDER="value"
FRAMESPACING="value" BORDERCOLOR="#xxxxxx">
```

- SRC l'url del file che verrà mostrato nel frame;
- NAME il nome a cui riferirsi come target se si dovesse usare un link, per aggiornare un unico frame;
- MARGINWIDTH e MARGINHEIGHT è la distanza che il documento HTML contenuto nel frame avrà dai bordi del frame stesso;
- SCROLLING identifica la presenza di barre di scorrimento (si, no oppure automatiche);
- NORESIZE permette al frame di non essere ridimensionabile (attraverso il trascinarsi dei bordi);
- FRAMEBORDER è supportato da Explorer in base al valore (0 non mostra bordi), mentre Netscape lo interpreta come yes/no; inoltre Netscape lo accetta anche nel <FRAMESET>;
- FRAMESPACING è accettato solo da Explorer e definisce lo spazio in pixel che si deve trovare attorno al frame;

- `BORDECOLOR` è accettato solo da Netscape (anche nel `<FRAMESET>`) e definisci il colore del bordo del frame in base al codice esadecimale.

Dopo aver inserito i vari frame è possibile utilizzare il tag `<NOFRAMES>` e `</NOFRAMES>`, al cui interno andrà digitato il documento alternativo per quei browser che ancora non supportano i frames.

```
<HTML>
<HEAD>...</HEAD>
<FRAMESET...>
<FRAME...>
<FRAME...>
...
<NOFRAMES>
...
</NOFRAMES>
</FRAMESET>
</HTML>
```

8.1 *Frame in linea*

L'elemento `IFRAME`, accennato precedentemente, si sostituisce al tag `FRAMESET` permettendoci di costruire frame interni alla struttura HTML, cosa che al momento solo Internet Explorer dalla v.3 e successive riesce a mettere in pratica.

```
<div align=center>
  <iframe src="Url" name="nome iframe" frameborder=0 scrolling=yes width=value
height=value align=middle></iframe>
</div>
```

DIV ALIGN con questo il frame può essere posizionato (a destra, a sinistra e al centro della struttura);

SRC l'url del file che verrà mostrato nel frame;

NAME il nome a cui riferirsi come target se si dovesse usare un link, per aggiornare un unico frame;

FRAMEBORDER è supportato da Explorer in base al valore (0 non mostra bordi) mentre Netscape lo interpreta come yes|no;

SCROLLING identifica la presenza di barre di scorrimento (si, no oppure automatiche);

WIDTH indica la dimensione del frame in altezza;

HEIGHT indica la dimensione del frame in larghezza;

ALIGN allineamento verticale o orizzontale;

9 I Form

Una delle caratteristiche più interessanti del WEB è la natura interattiva delle sue pagine. L'interattività viene fornita anche dalla possibilità di richiedere dati all'utente, di inviare commenti e di controllare l'azione di un programma. Per tutto questo viene usata una funzione HTML chiamata form. Esistono diversi controlli per i form, ma l'argomento potrebbe essere troppo lungo e ci limitiamo solamente a una breve descrizione di base. Ogni form deve usare un elemento form (tag <FORM> e </FORM>), insieme ad alcuni elementi speciali per ogni controllo, come l'elemento input, l'elemento select e l'elemento textarea. Nell'esempio si userà un URL mailto per elaborare il form. Il codice qui sotto trasferisce i dati inseriti nel form inviando un messaggio all'indirizzo e-mail specificato:

```
<HTML LANG=«IT»>
```

```
<HEAD>
```

```
<TITLE> Modulo per commenti </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<p>Inviate i vostri commenti sul sito.
```

```
<FORM ACTION="mailto:bianchi@dominio.it" METHOD="POST"  
ENCTYPE="TEXT/PLAIN">
```

```
<p>Nome: <INPUT TYPE="TEXT" NAME="nome">
```

```
<p>Indirizzo e-mail: <INPUT TYPE="TEXT" NAME="email">
```

```
<p>La URL del sito WEB preferito:<BR>
```

```
<INPUT TYPE="TEXT" SIZE="60" NAME="url" VALUE="http://">
```

```
<p>Commenti:<BR>
```

```
<TEXTAREA ROWS="5" COLS="60" NAME="commenti">Digitate qui i vostri
commenti.</TEXTAREA>
```

```
<p><INPUT TYPE="SUBMIT" VALUE="Invia">
```

```
<INPUT TYPE="RESET" VALUE="Annulla">
```

```
</FORM>
```

Nell'esempio riportato abbiamo due pulsanti invia e annulla generati dall'elemento input. Quando sono selezionati i due bottoni, l'attributo METHOD dell'elemento form indica al browser di inviare (POST) i dati compilati all'indirizzo email specificato dall'attributo ACTION. Poi l'attributo ENCTYPE dell'elemento form indica di inviare i dati del form in formato testo. Dopo aver inviato i dati questi saranno inviati dal programma di email. Navigator invia le informazioni automaticamente dopo aver mostrato un messaggio di avvertimento. Internet Explorer 4 invece utilizza il programma di email dell'utente. Volendo si può visualizzare un testo qualsiasi sui bottoni cambiando l'impostazione dell'attributo VALUE. In più l'HTML 4.0 supporta l'attributo TARGET, per definire una nuova finestra per output di un programma CGI, nonché il nuovo attributo ACCEPT, che può essere usato se il server WEB accetta l'invio (upload) di un file.

Invece un esempio di form semplice come la creazione di un bottone collegato ad un URL è il seguente:

```
<FORM ACTION="http://www.netscape.com" METHOD="GET">
```

```
<INPUT TYPE="SUBMIT" VALUE="Vai a Netscape">
```

```
</FORM>
```

In questo caso l'attributo METHOD dell'elemento form indica al browser di ottenere (GET) la pagina specificata dall'attributo ACTION.

Inoltre si può usare un'immagine come pulsante. Un pulsante immagine funziona come pulsante submit, ad eccezione del fatto che viene visualizzata l'immagine specifica al posto del pulsante di comando. Gli attributi per un pulsante immagine sono simili a quelli usati nel tag . Bisognerà specificare l'origine (SRC) dell'immagine e un testo alternativo usando l'attributo ALT:

```
<INPUT TYPE="IMAGE" SRC="nomefile.gif" ALT="[Invia]" ALIGN="RIGHT">
```

HTML 4.0 ha introdotto un nuovo elemento chiamato **BUTTON** che fornisce una maggiore flessibilità dell'elemento input, poiché può contenere immagini, elenchi, oggetti, paragrafi multipli, e altri ancora. Quello che viene inserito tra i tag `<BUTTON>` e `</BUTTON>` appare come un pulsante. L'elemento **BUTTON** supporta l'attributo **TYPE** che può essere usato solo come `TYPE="BUTTON"`, `TYPE="RESET"`, e `TYPE="SUBMIT"`.

HTML 4.0 ha introdotto inoltre l'elemento **LABEL** per consentire ai browser di sintesi vocale di descrivere i controlli in un form. Questo elemento deve iniziare con il tag `<LABEL>`, contenere un controllo del form con il relativo testo descrittivo e finire con il tag `</LABEL>`.

```
<LABEL>Nome: <INPUT TYPE="TEXT" NAME="nome"></LABEL>
```

10 I Fogli di Stile (CSS)

Il foglio di stile è un insieme di definizioni che agiscono sull'aspetto di un documento. Attualmente il tipo di foglio di stile più comune è Cascading Style Sheet (CSS), che viene identificato come foglio di stile di livello 1 perché è stato il primo, e finora l'unico, a essere creato. CSS1 serve per determinare come deve apparire sullo schermo un documento quando viene usato un browser grafico come IE 4 e Navigator 4. Sfortunatamente, poiché i fogli di stile rappresentano una nuova ed emergente tecnologia, browser differenti reagiscono ai fogli di stile in modi diversi. Alcune definizioni del foglio stile che vengono interpretate correttamente da Navigator 4 non sono interpretate nello stesso modo da IE 4, e viceversa. CSS1 utilizza delle definizioni per controllare circa 50 proprietà differenti, come i colori, lo sfondo, i tipi di carattere, l'aspetto dei bordi, i margini, l'allineamento e la spaziatura dei caratteri. Un esempio di foglio di stile CSS1 con una definizione è il seguente:

```
H1 {text-align: center}
```

Questa definizione indica di centrare tutte le intestazioni di primo livello presenti in un documento. Come si vede, i fogli di stile usano una sintassi completamente diversa da quella di HTML.

Comunque il metodo più semplice per apprendere qualcosa è quello di fare pratica. Per cui presenteremo un semplice esempio che mostra come i fogli di stile possono cambiare l'aspetto di un documento definendo la sintassi e la terminologia di CSS1. Si crea una definizione CSS1 usando un elemento stile (`<STYLE>` e `</STYLE>`) nella sezione **HEAD** di un documento HTML per cambiare l'aspetto di tutte le intestazioni di primo livello:

```
<HTML LANG="IT">
```

```
<HEAD>
<TITLE>Esempio foglio di stile </TITLE>
<STYLE TYPE= "text/css">
H1 {border : thick solid blue}
</STYLE>
</HEAD>
<BODY>
<H1> Questo è un esempio di foglio di stile.</H1>
<p>Questa pagina mostra un bordo spesso che racchiude il testo frutto del foglio di stile.
</BODY>
</HTML>
```

Questo esempio mostra un documento assolutamente normale ad eccezione del fatto che il testo racchiuso tra i tag `<H1>` e `</H1>` viene incorniciato con un bordo spesso. Uno dei vantaggi offerti dai fogli di stile è che questo documento non contiene nessun codice per il tipo di carattere o per i bordi; il bordo viene specificato dal foglio di stile all'interno della sezione `HEAD`, con una sola definizione: `H1 {border: thick solid blue}`. Questa semplice definizione indica che tutte le intestazioni di primo livello devono avere un bordo spesso blu pieno. Le definizioni sono costituite da selettori e dichiarazioni, e le dichiarazioni consistono di proprietà e valori. L'elemento `STYLE` usato nell'esempio ha un attributo `TYPE` (`TYPE="text/css"`), che indica che il foglio di stile è stato scritto con il linguaggio Cascading Style Sheet. Questo attributo o uno equivalente è necessario per indicare al browser quale linguaggio di foglio stile utilizzare.

L'argomento meriterebbe un maggiore approfondimento, ma in questa sede ci limitiamo a dire che vi sono:

1. I fogli di stile all'interno dell'elemento stile nella sezione `HEAD` di un documento (fogli di stile incorporati) come nel primo esempio accennato prima. Il tag `<STYLE>` dovrebbe sempre contenere un attributo `TYPE` per indicare il tipo di foglio stile usato.

Una considerazione importante relativa ai fogli stile incorporati è che i browser più datati, che non riconoscono l'elemento `STYLE`, potrebbero essere confusi dalla presenza dell'elemento `STYLE` e visualizzare così le definizioni sullo schermo. Per evitare questo inconveniente, si dovrebbe sempre inserire il foglio di stile in una struttura di commento. In questo modo, i browser che

supportano CSS1 riconoscono ugualmente il foglio di stile, mentre gli altri browser ignorano il contenuto del commento e procedono oltre. Per inserire il foglio di stile come commento, dovete digitare i caratteri `<!--` prima del foglio stile, e i caratteri `-->` dopo il foglio stile, come vi mostrerò qui sotto.

```
<HEAD>
<TITLE>titolo da mettere</TITLE>
<STYLE TYPE="text/css">
<!--BODY {color: white; background: black}
TD {color: white; background: black}-->
</STYLE>
</HEAD>
```

2. Collegamenti a fogli di stile esterni usando l'elemento LINK per associare un foglio stile a un documento HTML. La possibilità di collegarsi a un foglio stile esterno è l'opzione più potente, poiché permette di definire un foglio stile standard per l'intero sito WEB. Ogni file HTML nel sito può contenere un collegamento a quel foglio stile standard. Se si vuole cambiare l'aspetto di ogni pagina del sito contemporaneamente, è sufficiente modificare il foglio stile esterno, invece di intervenire sui singoli file HTML. Ci sono tre attributi obbligatori da impostare, insieme ad altri due attributi opzionali come vedrete sotto.

- L'attributo HREF deve specificare la URL del file del foglio esterno.
- L'attributo REL deve avere il valore "STYLESHEET" o "ALTERNATE STYLESHEET".
- L'attributo TYPE deve essere "text/css", a meno che il linguaggio di default per il foglio di stile non sia già stato specificato nella sezione HEAD con il tag META.
- L'attributo opzionale TITLE può indicare il nome del foglio stile, se si vuole consentire agli utenti di disabilitarlo.
- L'attributo opzionale MEDIA serve per indicare il dispositivo a cui viene associato il foglio stile.

Per capire meglio guardiamo l'esempio sotto di un codice che imposta un collegamento a un foglio stile esterno chiamato random.css.

```
<LINK HREF="random.css" REL="STYLESHEET" TYPE="text/css" TITLE="Il mio lavoro">
```

Un altro esempio è quello di un elemento HEAD che mostra come impostare il linguaggio di default per il foglio stile puntato da un URL assoluta.

```
<HEAD>
<TITLE>Mia home page</TITLE>
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
<LINK HREF="http://www.software.com/~mauri/nomefile.css" REL="STYLESHEET"
TITLE="Foglio stile di gioco">
</HEAD>
```

Qualsiasi foglio stile può usare la notazione speciale @import per fare riferimento a un foglio stile esterno. Le definizioni del foglio stile esterno vengono importate, cioè caricate, insieme alle definizioni del foglio stile principale, ed entrambi i fogli stile vengono associati agli elementi del documento HTML. L'enunciato @import deve apparire all'inizio del foglio stile, prima di qualsiasi definizione. Si deve usare la parola url con delle parentesi intorno alla URL del foglio stile esterno per indicare la posizione.

```
@import url (nomefile.css)
```

In questo caso è stata usata una URL relativa. Il browser carica il foglio stile contenuto nel file chiamato nomefile.css. Questo file deve trovarsi nella stessa directory del file del foglio stile che contiene l'enunciato @import.

Quando si crea un foglio stile esterno, si dovrebbe creare un file di testo che contiene definizioni e gruppi di definizioni senza nessun elemento HTML. Ma, attenzione bisogna ricordarsi che l'eventuale enunciato @import deve essere il primo.

Si possono usare anche URL assolute.

```
@import url (http://www.software.com/~mauri/css/nomefile.css)
```

CSS1 usa per i commenti un formato diverso da quello di HTML. I commenti di CSS1 iniziano con /* e finiscono con */.

```
/* Esempio commento di CSS1 */
```

Ora date una occhiata sotto a questo foglio stile con un commento che spiega la sua funzione.

```
nome.css
```

```
/* CSS1 Style Sheet by HTML point */
```

```
@import url (ourdefaultfonts.css)
```

```
*/ The ourdefaultfonts.css style sheet contains font settings */
```

```
BODY {background: black; color: white; margin-left: 10%; margin-left: 10% }
```

```
H1 {text-align: center}
```

```
TD{color: white; background: black}
```

```
/* Since Navigator 4 doesn't inherit properties to tables, you must separately define the body rule for a table cell */
```

```
WARNING {font-size: larger ; font-weight: bolder ; text-align: center; color: red; background : Yellow; border: thick groove gray;}
```

```
HR {text-align: center; margin-left: 25%; width: 50%; margin-right: 25%;}
```