

CORSI PER ADULTI

Le basi del Web: il linguaggio HTML e correlati

Corso: 14.03.2011 – 05.05.2011

Docente: Natan Colombini
Professione: Consulente Informatico
Specializzazione: Progettista tecnico WEB
Tecnico sistemi e reti
E-mail : n.colombini@gmail.com

1. INTRODUZIONE

1.1 COSA È L'HTML

L'HTML (HyperText Markup Language - linguaggio di descrizione per ipertesti) è il principale codice di programmazione utilizzato dal mondo internet. Esistono molti altri codici di sviluppo per le pagine internet (ad esempio php), ma tutti devono essere affiancati dall'HTML.

Quindi possiamo affermare che: **tutti i siti internet sono sviluppati su una base di HTML**¹

1.2 COSA È UN SITO INTERNET

Un sito internet non è nient'altro che **un'insieme di file** contenenti codice HTML.

Ogni file corrisponde ad una pagina internet (o ad una parte di pagina). I file di codice HTML hanno un'estensione² .htm oppure .html, aprendoli con editore di testi (esempio il *blocco note* di Windows) al suo interno si trova il codice sorgente. Oltre ai file HTML a dipendenza possiamo trovare altri file di immagini o pdf o qualsiasi altro file compatibile con il web.

1.3 COME SI CREA UN SITO INTERNET

Per la creazione di siti internet con codice HTML **non servono programmi particolari**, ci basta il semplice *blocco note* di Windows. Poi per la visualizzazione migliore delle pagine, o meglio della struttura del codice, vi sono molti programmi free (gratis) che aiutano il webmaster³ a leggere e scrivere tale codice (basta scrivere in google la stringa *editor html*).

Esistono diversi software grafici (esempio FrontPage del pacchetto MS Office Professional) che permettono al webmaster di creare un sito, o una pagina, internet in modo più veloce e molto più semplice che scrivere a mano il codice HTML. Però questi programmi, a parte che sono quasi tutti a pagamento, inseriscono nel codice parti non necessarie così da rendere i nostri file più pesanti (con ovvii rallentamenti nella visualizzazione), ma soprattutto utilizzando questi programmi in caso di errori il webmaster si può ritrovare molto in difficoltà a risolverli. Questi programmi sono molto utili solo se si conosce alla perfezione cosa sta dietro (il codice HTML).

Un sito internet HTML **viene quindi creato e modificato sempre localmente** sul proprio computer per poi essere infine messo su internet. Non è necessario avere una connessione a internet per sviluppare un sito, la connessione si rende necessaria unicamente a lavoro completato per il trasferimento dei file sul web.

1.4 COSA SONO I BROWSER

I Browser non sono altro che programmi che permettono di visualizzare le pagine (i file) HTML in modo grafico (e non ha codice). Esistono oggi diversi browser, tra cui troviamo il più conosciuto tra essi, e cioè Internet Explorer della Microsoft, oppure l'open source⁴ Mozilla Firefox. Un buon webmaster deve testare le proprie pagine internet su più browser, questo perché il codice HTML può risultare corretto su di un browser e nello stesso tempo "sballare" la pagina su altri browser, normalmente questo problema non dovrebbe sussistere se le pagine vengono scritte in modo

¹ Da tempo a questa parte (anno 2001 circa) è stato rilasciato il codice XHTML che comunque è un'evoluzione basata sempre su HTML

² Non esiste differenza tra un'estensione .htm e una .html, le estensioni a 3 lettere sono derivate dal sistema operativo DOS

³ Con il termine webmaster si definisce colui che crea, gestisce e progetta il sito internet

⁴ Open source è un programma (creato da uno o più programmatori) di libero accesso (ed utilizzo) alla propria sorgente

corretto. Per riagganciarci al capitolo precedente (1.3): tramite i programmi grafici questi errori sono molto spesso presenti.

Il funzionamento dei browser è il seguente: prendono dal nostro server web (dove sono situate le pagine) i nostri file di codice HTML, li scaricano sul computer di chi sta navigando, ed infine li visualizzano a schermo sotto forma grafica.

Quindi ricordate che: **le nostre pagine internet, ed i loro contenuti, vengono salvate sul PC di chi lo visualizza.**

2. LA PRIMA PAGINA HTML

2.1 CHE NOME DARE ALLE PAGINE

Prima di tutto, come accennato nel capitolo 1.4, le pagine vengono salvate sul computer di chi le visualizza, quindi bisogna **evitare** nomi che hanno un **linguaggio scurrile**.

Più importante: non mettete **mai spazi nel nome della pagina**, infatti anche se i nostri sistemi operativi (Windows, MAC, Linux, ...) danno questa possibilità i browser non la possono gestire. Ad esempio per chiamare la pagina web della mia descrizione la chiamerò **chi_sono.html**, sostituendo quindi lo spazio con il segno di *underscore* (_).

Infine bisogna **evitare** di chiamare le nostre pagine con **caratteri maiuscoli**, alcuni vecchi browser infatti non riescono a gestirli, ed è possibile che navigando tra le pagine queste non vengano visualizzate e restituiscano un errore al visitatore.

2.2 LA STRUTTURA BASE

Tutte le istruzioni del codice HTML sono chiamate TAG. Questi elementi definiscono la struttura della pagina, possono quindi rappresentare paragrafi, link², elenchi, immagini, tabelle ed altro.

Tutti i tag sono racchiusi da parentesi uncinata, maggior parte di questi elementi hanno un'apertura (< ... >) ed una chiusura (</ ... >).

Qui di seguito alcuni punti molto importanti:

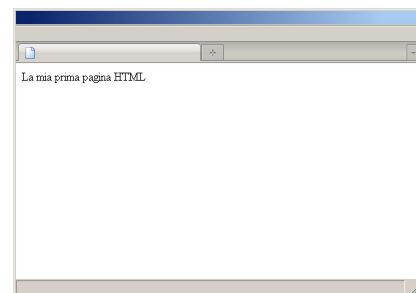
- Tutte le pagine HTML devono avere come primo ed ultimo TAG <html>
- Quasi tutti i TAG devono avere un'apertura ed un chiusura (eccezioni a parte)
- Tutti i TAG seguono la regola della nidificazione
- Tutti i TAG devono essere scritti con caratteri minuscoli

2.2 Esempio

a) La struttura base di una pagina HTML con testo

```
<html>
<head>
</head>
<body>
  <p>
    La mia prima pagina HTML
  </p>
</body>
</html>
```

risultato



Il codice HTML può essere scritto anche tutto su di una riga, ma per comodità di lettura si consiglia di seguire la forma sopra descritta.

² Link, ovvero collegamenti ipertestuali, servono per lo spostamento tra documenti (o pagine)

⁶ I motori di ricerca sono quei siti internet che permettono di ricercare altri siti internet, un esempio di motore di ricerca è google ma ve ne sono molti altri

2.3 I TAG HEAD

Questi TAG racchiudono le informazioni riguardanti il sito, quali potrebbero essere le parole chiave (*keywords*) che caratterizzano il sito stesso (fondamentali per una buona posizione nei motori di ricerca⁶), oppure l'autore di tale pagina (copyright), oppure ancora molte altre istruzioni, quali ad esempio le funzioni javascript.

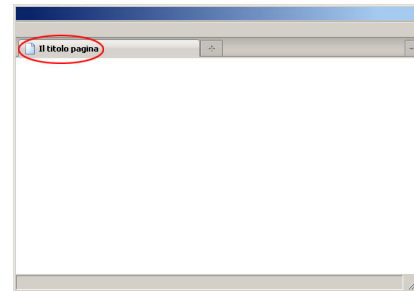
Ma per ora limitiamoci a sapere che **in questi TAG può essere annidato** il TAG <title> il quale permette la visualizzazione di **un titolo sulla nostra pagina**.

2.3 Esempio

a) La struttura base di una pagina HTML con titolo

```
<html>
<head>
  <title>Il titolo pagina</title>
</head>
<body>
</body>
</html>
```

risultato



b) La struttura base di una pagina HTML con titolo, keyword, description e copyright

```
...
<head>
  <title>Il titolo pagina</title>
  <meta name="keywords" content="primo,sito,pagina,internet">
  <meta name="description" content="Questa è la mia prima pagina internet">
  <meta name="copyright" content="Natan Colombini">
</head>
...
```

Il risultato a video di tale codice è lo stesso dell'esempio precedente, infatti i TAG meta name servono unicamente ai motori di ricerca.

2.4 I TAG BODY

Nei TAG <body> **viene racchiuso tutto ciò di visibile del nostro sito internet**, qualsiasi elemento visualizzato a schermo dovrà infatti essere inserito in questi TAG (eccezione fatta per il titolo).

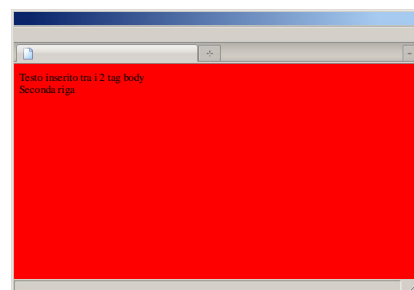
Tra i 2 TAG body infatti possiamo inserire testi, immagini, link, tabelle e molto altro. Mentre al suo interno possiamo parametrizzare lo stile della pagina (capitolo 3).

2.4 Esempio

a) La struttura base di una pagina HTML con TAG body parametrizzato

```
<html>
<head>
</head>
<body bgcolor="red">
  <p>
    Testo inserito tra i 2 tag body<br />
    Seconda riga
  </p>
</body>
</html>
```

risultato



2.5 I COMMENTI

Fino ad ora le nostre pagine internet risultano avere poche righe di codice, ed un primo impatto possiamo trovare subito le informazioni che ci necessitano. Però una pagina internet un po' più complessa può avere svariate righe e ritrovare quanto ricercato diventa un'impresa non semplice.

Qui vengono in aiuto i commenti. I commenti servono unicamente al webmaster, questi infatti non vengono visualizzati a schermo ma solamente a codice.

2.5 Esempio

a) Per inserire un commento questa la sintassi

```
<!-- questo è un commento -->
```

2.6 IL W3C

Il World Wide Web Consortium (W3C) è un'istituzione che crea gli standard per il mondo internet. Per ulteriori informazioni a riguardo, visitate il sito internet <http://www.w3.org>.

A noi interessa però sapere che la nostra pagina internet (funzionante) non è ancora valida per il w3c, infatti mancano alcune parti di codice obbligatorie per tale validazione.

2.6 Esempio

a) Struttura base di una pagina HTML valida per il W3C

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Il titolo pagina</title>
  <meta name="keywords" content="primo,sito,pagina,internet">
  <meta name="description" content="Questa è la mia prima pagina internet">
  <meta name="copyright" content="Natan Colombini">
</head>
<body>
  <p>
  La mia prima pagina HTML
  </p>
</body>
</html>
```

3. PARAMETRIZZARE IL TAG BODY

3.1 IMPOSTARE IL COLORE

Come anticipato precedentemente per impostare il colore di sfondo di una pagina internet viene utilizzata l'istruzione **bgcolor** (abbreviazione di background color) all'interno del TAG <body>.

Il colore può essere inserito nella forma inglese (*bgcolor="red"*), ma questa è sconsigliata per il semplice motivo che per inserire un colore rosso tendente al bianco la formula in lingua inglese non è disponibile, e anche se il colore rosa può essere scritto "pink" se volessi che il colore fosse leggermente più rosso sarei impossibilitato. Ecco perché per scrivere ad esempio il colore viene utilizzata la forma esadecimale basata su RGB (Rosso Verde Blu), il colore viene così inserito: *bgcolor="#FF0000"*.

È consigliabile **sempre inserire un colore di sfondo per le nostre pagine web**, anche se il colore che vogliamo visualizzare è il bianco (#FFFFFF). Il motivo è che l'utente (il visitatore) sul proprio computer può avere diverse impostazioni personalizzate, tra le quali può modificare anche i colori predefiniti del suo browser. *Ad esempio in internet explorer 8 selezionando la voce opzioni dal menu strumenti è possibile modificare i colori predefiniti. Un'impostazione (del visitatore) di sfondo nero, un testo (della pagina) di colore nero e un sito internet senza impostazione dello sfondo e questo risulterà illeggibile.*

Il cervello umano percepisce ed elabora i colori di un sito internet in un tempo molto breve, e ovviamente il colore di sfondo di una pagina web è la prima cosa che il visitatore nota. È **molto importante scegliere bene i colori** (sia di sfondo che del testo). I colori forti (rosso, giallo, blu o colori fosforescenti) possono sembrare carini al primo impatto ma a lungo stancano gli occhi.

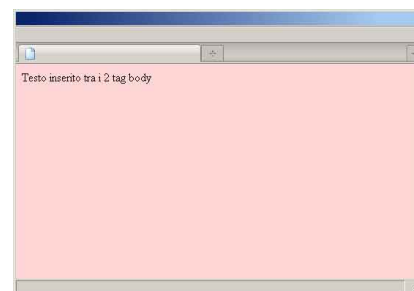
La miglior scelta per un colore di sfondo è quella di **utilizzare** dei **colori con un basso contrasto** (bianco, giallino, rosa, verdino, ... chiamati anche *colori pastello*), mentre il testo deve avere un forte contrasto rispetto allo sfondo, ma non troppo forte da disturbarne la lettura.

Per conoscere i codici esadecimali dei colori bisogna (a meno che non conosciate a memoria tutti i colori ed i rispettivi codici) utilizzare un programma di grafica, oppure cercarli in un qualche motore di ricerca.

3.1 Esempio

a) Una pagina HTML con sfondo di colore rosa e testo di colore nero

```
<html>
<head>
</head>
<body bgcolor="#FFD6D4" text="#000000">
  <p>
    Testo inserito tra i 2 tag body
  </p>
</body>
</html>
```



3.2 IMPOSTARE UN'IMMAGINE DI SFONDO

Per impostare un'immagine di sfondo bisogna utilizzare (sempre all'interno del TAG <body>) l'istruzione background (<body background="immagine">).

Come anche per il colore dello sfondo, bisogna fare molta attenzione a quale immagine viene scelta, per i seguenti motivi:

- Non scegliere un'immagine con tonalità dello stesso colore del testo
- Non deve occupare troppo spazio su disco, per non rendere lenta l'apertura delle pagine
- Il mondo web supporta unicamente i formati: .jpg, .gif, .png

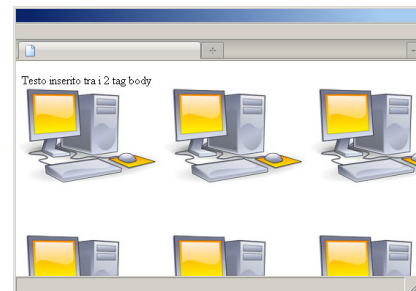
Per inserire un'immagine di sfondo, ad esempio di un computer, vi sono i 3 modi seguenti:

- <body background="computer.jpg">
L'immagine computer.jpg si trova nella stessa directory dove si trova la pagina web
- <body background="../immagini/computer.jpg">
L'immagine computer.jpg si trova nella directory immagini posizionata ad un livello superiore
- <body background="http://www.motogastro.ch/computer.jpg">
L'immagine computer.jpg si trova sul sito internet www.motogastro.ch

3.2 Esempio

a) Una pagina HTML che ha come sfondo un'immagine

```
<html>
<head>
</head>
<body background="computer.jpg">
  <p>
    Testo inserito tra i 2 tag body
  </p>
</body>
</html>
```



Come si può notare l'immagine di sfondo viene ripetuta per tutta la pagina web, per questo un'immagine come quella da noi scelta non va bene. Un'immagine di sfondo ripetuta viene chiamata *pattern*. Per poter avere l'immagine di sfondo fissa (quindi non ripetuta) bisogna utilizzare i CSS, quindi per il momento non vediamo come fare.

Tramite l'attributo seguente possiamo però rendere l'immagine fissa in caso di barre (orizzontali e/o verticali): <body background="computer.jpg" bgproperties="fixed">

3.3 I MARGINI

Tutte le pagine internet hanno dei margini (spazio libero tra la pagina ed il suo contenuto) a sinistra e in alto. Questi margini possono dare fastidio nel caso necessitiamo di far partire il testo da inizio pagina, per ovviare a questo "problema" bisogna parametrizzarli (all'interno del TAG <body>) nel seguente modo: <body leftmargin="0" topmargin="0">.

I margini (come del resto tutte le unità di misura di HTML) sono espressi in pixel.

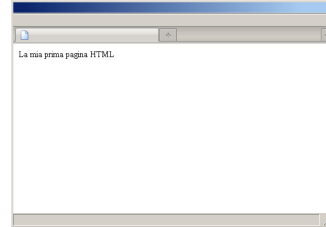
Ad oggi non più tutti i browser supportano tale struttura.

3.3 Esempio

a) Vediamo nella pagina successiva, alcune pagine HTML con e senza margini (con solo codice del TAG <body>)

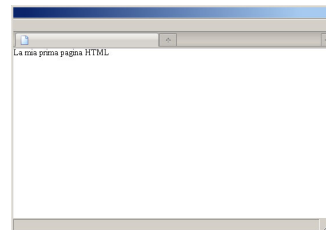
PAGINA SENZA IMPOSTAZIONI

```
...  
<body>  
La mia prima pagina HTML  
</body>  
...
```



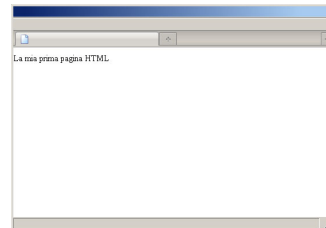
PAGINA SENZA MARGINI

```
...  
<body leftmargin="0" topmargin="0" >  
La mia prima pagina HTML  
</body>  
...
```



PAGINA SENZA MARGINE SINISTRO

```
...  
<body leftmargin="0" >  
La mia prima pagina HTML  
</body>  
...
```



3.4 LA LINGUA DEL DOCUMENTO

Impostare la lingua del documento non è indispensabile, infatti questa operazione non cambierà nulla nella visualizzazione della pagina, però il W3C (capitolo 2.4) consiglia di inserire tale parametro.

Potrebbe comunque essere utile anche ai motori di ricerca nel caso il vostro sito internet abbia pagine in diverse lingue.

Ecco come impostare la lingua della pagina: **<body lang="it">**

4. PARAMETRIZZARE I TESTI

4.1 SOSTITUIRE IL TIPO DI CARATTERE

Se non impostato diversamente il carattere inserito nelle nostre pagine è quello di default selezionato dal browser che utilizziamo.

Per modificare il carattere di un nostro testo bisogna parametrizzare il TAG **** con l'attributo **face** seguito dal nome del carattere.

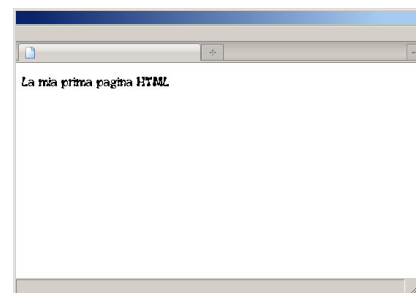
Con HTML bisogna però fare attenzione che il carattere da noi scelto sia un font presente su molte delle macchine dei nostri visitatori, infatti se utilizziamo un carattere di tipo *Arial* questo sarà verosimilmente presente su tutti i computer (con sistema operativo Windows) e quindi la nostra pagina sarà visualizzata nello stesso modo da tutti i visitatori. Se però utilizziamo un carattere tipo *flubber* questo sarà presente sicuramente su pochi computer e quindi il nostro visitatore non vedrà il testo come noi vorremmo fargli apparire.

Per ovviare a questo "problema" è consigliabile **inserire più font** all'interno della nostra parametrizzazione **face**.

4.1 Esempio

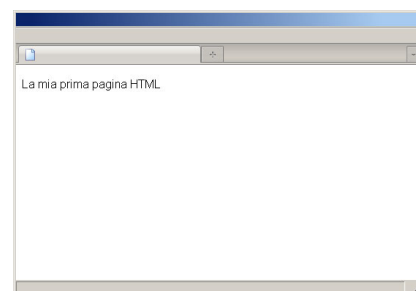
a) Una pagina HTML testo di tipo `flubber` e con computer testo **supportato**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type">
</head>
<body>
  <p>
    <font face="flubber,arial">
      La mia prima pagina HTML
    </font>
  </p>
</body>
</html>
```



b) Una pagina HTML testo di tipo `flubber` e con computer testo **NON supportato**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//IT">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type">
</head>
<body>
  <p>
    <font face="flubber,arial">
      La mia prima pagina HTML
    </font>
  </p>
</body>
```



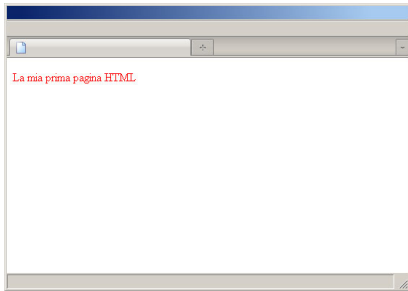
</html>

4.2 PARAMETRIZZARE IL CARATTERE

Oltre ad impostare un tipo diverso di carattere, abbiamo la possibilità di sostituirne il colore () e/o le dimensioni ().

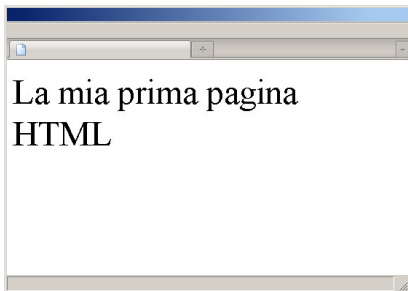
4.2 Esempio

a) Una pagina HTML con testo di colore rosso



```
...  
<body>  
<p>  
<font color="#FF0000">  
  La mia prima pagina HTML  
</font>  
</p>  
</body>  
...
```

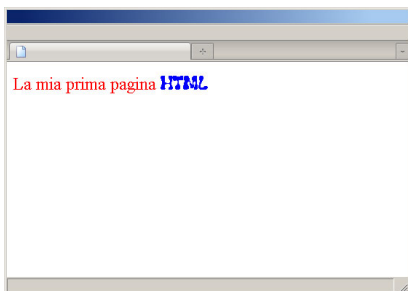
b) Una pagina HTML con testo grande



```
...  
<body>  
<p>  
<font size="7">  
  La mia prima pagina HTML  
</font>  
</p>  
</body>  
...
```

Il parametro size è impostabile con valori da 1 a 7 (1 piccolo - 7 grande)

c) Una pagina HTML con diversi stili di testo



```
...  
<body>  
<p>  
<font size="5" color="#FF0000">  
  La mia prima pagina  
<font face="flubber,arial"  
  color="#0000FF">  
  HTML  
</font>  
</font>  
</p>  
</body>  
...
```

4.3 GLI STILI DI TESTO

In questo capitolo vediamo i TAG che permettono di definire degli stili particolari per il nostro testo. Ad esempio per scrivere un testo in grassetto non è necessario impostare dei parametri all'interno del TAG font, ma basta utilizzare il TAG .

Esistono molti stili per i testi, ma qui ne visualizzeremo solo qualcuno (i più usati), per avere la lista di tutti gli stili di testo visitate l'URL: <http://www.w3schools.com/>

Tutti gli esempi di TAG per lo stile di testo sono visualizzabili nella pagina seguente.

4.3.1 Testi in grassetto, corsivo e sottolineati

<code>Qui il nostro testo</code>	Qui il nostro testo
<code><i>Qui il nostro testo</i></code>	<i>Qui il nostro testo</i>
<code><u>Qui il nostro testo</u></code>	<u>Qui il nostro testo</u>
<code>Qui il nostro <i>testo</i></code>	Qui il nostro <i>testo</i>

4.3.2 Testi apice, pedice e barrati

<code>Qui il nostro <sup>testo</sup></code>	Qui il nostro ^{testo}
<code>Qui il nostro <sub>testo</sub></code>	Qui il nostro _{testo}
<code><strike>Qui il nostro testo</strike></code>	Qui il nostro testo
<code><strike>Qui il nostro <sup>testo</sup></strike></code>	Qui il nostro ^{testo}

4.4 CARATTERI SPECIALI

A volte il webmaster ha la necessità di inserire caratteri più o meno speciali. Ad esempio inserire il minore (<) in testo non è semplice come sembra, infatti il segno < in HTML vuol dire che si sta aprendo un TAG e quindi il codice si aspetta che dopo tale segno vi sia scritto un TAG, ma se questo non avviene la nostra pagina non sarà visualizzata come ci aspettavamo.

Per inserire un carattere speciale (ASCII) bisogna quindi intervenire in un modo un po' più complicato ed all'interno del nostro codice per rappresentare il segno < dobbiamo inserire il codice: <

Vediamo qui sotto una tabella ASCII to HTML presa dal sito <http://www.w3schools.com/> e adattata, eventualmente per visualizzarla tutta visitate il sito internet.

ASCII	HTML	Descrizione
"	"	virgolette
#	#	cancelletto
\$	$	dollaro
%	%	per cento
&	&	e commerciale
'	'	apostrofo
*	*	asterisco
+	+	segno più
/	/	slash
<	<	minore
=	=	Uguale
>	>	maggiore
@	@	chiocciola
\	\	backslash
^	^	segno di omissione
`	`	accento grave
	|	barra verticale
~	~	circa

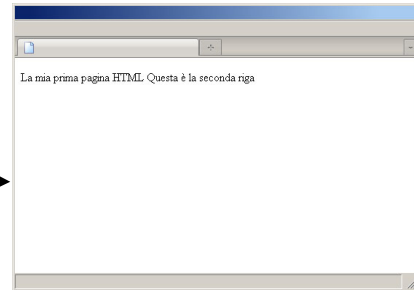
4.5 ANDARE A CAPO

Nel codice HTML quando si scrive, ad esempio un testo, e si va a capo, questo non viene riportato nella visualizzazione. Per andare a capo bisogna utilizzare il TAG `
`, questo è TAG particolare infatti non è scritto come gli altri `<TAG>` ed ha solo un apertura.

4.5 Esempio - andare a capo

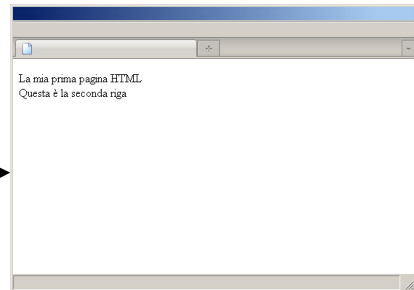
MODO NON CORRETTO

```
...  
<body>  
<p>  
  La mia prima pagina HTML  
  Questa è la seconda riga  
</p>  
</body>  
...
```



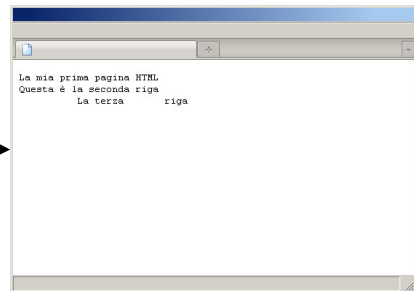
MODO CORRETTO

```
...  
<body>  
<p>  
  La mia prima pagina HTML<br />  
  Questa è la seconda riga  
</p>  
</body>  
...
```



Vi è comunque un'altra possibilità per andare a capo senza utilizzare il TAG `
`, utilizzando i TAG `<pre>` (e chiusura `</pre>`) possiamo impostare il carattere come da HTML. Ma **sconsiglio** vivamente **di utilizzare questo TAG**, per la non compatibilità con tutti i browser, vediamo comunque un esempio.

```
...  
<body>  
<p>  
<pre>  
La mia prima pagina HTML  
Questa è la seconda riga  
  La terza    riga  
</pre>  
</p>  
</body>  
...
```



4.6 I PARAGRAFI

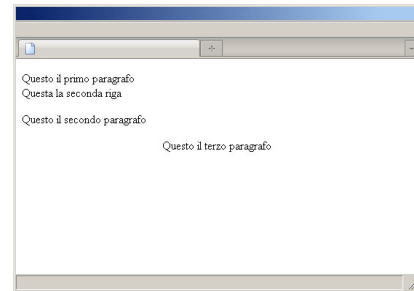
Ogni volta che si comincia a scrivere un testo (o anche inserire un'immagine) in un codice HTML, dopo il TAG `body` viene aperto un TAG `<p>`, per poi essere chiuso (`</p>`) prima di `body`.

Questa forma è stata resa necessaria per validare la pagina con gli standard W3C. Però il TAG `<p>` serve in molte altre situazioni, infatti se volessimo andare a capo di 2 righe potremmo scrivere `

`, ma la forma corretta è quella di chiudere l'attuale paragrafo e riaprirne uno nuovo. Tramite i paragrafi possiamo anche scegliere l'allineamento (sinistra, centro, destra o giustificato) del testo. Vediamo alcuni esempi nella pagina successiva.

4.6 Esempio

```
...  
<body>  
  <p>  
    Questo il primo paragrafo<br />  
    Questa la seconda riga  
  </p>  
  <p>  
    Questo il secondo paragrafo  
  </p>  
  <p align="center">  
    Questo il terzo paragrafo  
  </p>  
</body>  
...
```



4.7 I TITOLI

In HTML esistono i cosiddetti titoli (rappresentati dal TAG <h>), questi sono una sorta di scorciatoia per creare dei testi formattati in carattere di grandezza diversa senza l'utilizzo del TAG . Se ad esempio vogliamo inserire un testo che funga da titolo, possiamo inserire:

```
<h1>Qui il testo di dimensioni maggiori</h1>
```

Questo TAG funziona (oltre che da scorciatoia per la dimensione del testo) anche da paragrafo, quindi all'inserimento del TAG viene creato uno spazio di 2 righe e alla chiusura anche (proprio come un paragrafo).

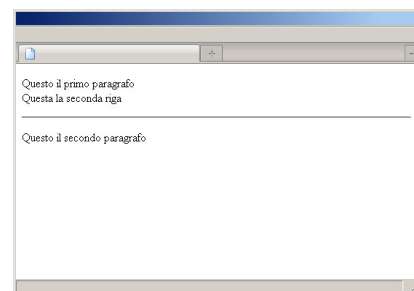
Il TAG <h> può essere settato da 1 a 6. Un testo formattato con il TAG <h1> ha una dimensione di 36 pixel (<h2> = 24 pixel; <h3> = 21 pixel; <h4> = 18 pixel; <h5> = 16 pixel; <h6> = 14 pixel).

4.8 LA LINEA DIVISORIA

Un altro TAG utile per separare la nostra pagina è il TAG <hr /> che viene utilizzato per tracciare una linea orizzontale nella pagina. Anche questo come il TAG
 ha solo l'apertura.

4.8 Esempio

```
...  
<body>  
  <p>  
    Questo il primo paragrafo<br />  
    Questa la seconda riga  
  </p>  
  <hr />  
  <p>  
    Questo il secondo paragrafo  
  </p>  
</body>  
...
```



Questo TAG è anche parametrizzabile in altezza (attributo size), in lunghezza (attributo width) e in allineamento (attributo align), come da esempio:

```
<hr size="3" width="100" align="right" />
```

5. GLI ELENCHI

5.1 COSA SONO GLI ELENCHI

A volte capita di aver la necessità di descrivere voci sottoforma di elenchi puntati. Tramite il codice HTML abbiamo le 2 seguenti possibilità:

- Elenchi non ordinati, tramite TAG
 1. Elenchi ordinati, tramite TAG

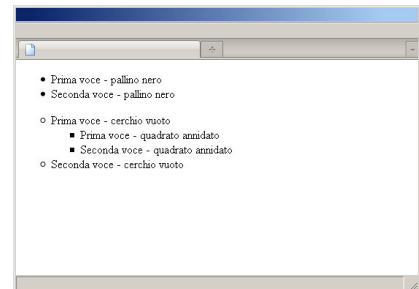
5.2 ELENCHI NON ORDINATI

Gli elenchi non ordinati sono quegli elenchi senza numeri e/o lettere come ad esempio il pallino nero, in HTML comunque vi è la possibilità di inserire anche altri tipi di segno, come il quadrato nero (square) o il cerchio vuoto (circle).

È buona cosa **impostare sempre il tipo di elenco non ordinato** che si vuole inserire, questo perchè se non impostato a dipendenza del browser utilizzato dal visitatore gli elenchi potrebbero essere diversi.

5.2 Esempio

```
...  
<body>  
<ul type="disc">  
  <li>Prima voce - pallino nero</li>  
  <li>Seconda voce - pallino nero</li>  
</ul>  
<ul type="circle">  
  <li>Prima voce - cerchio vuoto</li>  
  <ul type="square">  
    <li>Prima voce - quadrato annidato</li>  
    <li>Seconda voce - quadrato annidato</li>  
  </ul>  
  <li>Seconda voce - cerchio vuoto</li>  
</ul>  
</body>  
...
```



Come avrete notato nell'esempio gli elenchi (sia ordinati o non) possono tra di loro essere annidati. I più attenti avranno anche notato che alla chiusura (e non all'apertura) del TAG viene lasciato uno spazio come il paragrafo (questa regola non funziona per i TAG annidati).

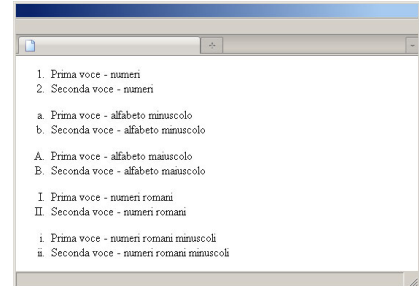
5.3 ELENCHI ORDINATI

Gli elenchi ordinati, come dice la parola stessa, sono gli elenchi che vanno in sequenza di numeri, ma anche di lettere e/o altro. In questo caso se non impostato diversamente, gli elenchi ordinati prendono come type il numero (da 1), questo in tutti i browser, quindi è inutile inserirlo, sappiate però che il tipo numerico viene impostato con type="1".

Anche con questa implementazione esistono diversi elenchi ordinati, vediamo degli esempi nella pagina successiva.

5.3 Esempio

```
...  
<body>  
<ol>  
  <li>Prima voce - numeri</li>  
  <li>Seconda voce - numeri</li>  
</ol>  
<ol type="a">  
  <li>Prima voce - alfabeto minuscolo</li>  
  <li>Seconda voce - alfabeto minuscolo</li>  
</ol>  
<ol type="A">  
  <li>Prima voce - alfabeto maiuscolo</li>  
  <li>Seconda voce - alfabeto maiuscolo</li>  
</ol>  
<ol type="I">  
  <li>Prima voce - numeri romani</li>  
  <li>Seconda voce - numeri romani</li>  
</ol>  
<ol type="i">  
  <li>Prima voce - numeri romani minuscoli</li>  
  <li>Seconda voce - numeri romani minuscoli</li>  
</ol>  
</body>  
...
```



Per partire ad esempio dal numero 5, nulla di più semplice, ci basterà inserire nel TAG l'attributo start. Vediamo la sintassi:

```
<ol type="a" start="5">
```

L'elenco partirà dalla lettera e

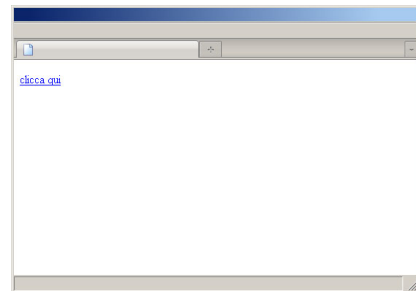
6. I LINK

6.1 COSA SONO I LINK

I Link, o collegamenti ipertestuali, vengono utilizzati per lo spostamento tra pagine e o altri siti internet. I link in HTML sono formati da 2 parti, la prima è quella visibile a schermo, quale potrebbe essere un testo oppure un'immagine cliccabili, la seconda (ma non meno importante) è il collegamento a dove. Mettiamo di avere una pagina con un testo *clicca qui* che si collega al sito internet *http://www.motogastro.ch*. La parte visibile è *clicca qui* mentre la parte di collegamento è *http://www.motogastro.ch*.

6.1 Esempio

```
...  
<body>  
<p>  
<a href="http://motogastro.ch">clicca qui</a>  
</p>  
</body>  
...
```



Vi è la possibilità di collegarsi a diversi file, o meglio, vi è la possibilità di creare un link per tutto quanto presente nel mondo del web (file html, siti internet, immagini, ...)

6.2 APERTURA SU NUOVA PAGINA

Come visto precedentemente quando viene premuto il link il risultato (in questo caso il sito internet *http://www.motogastro.ch*) viene aperto nella stessa pagina del nostro sito internet.

Nel caso si volesse far aprire la pagina in una nuova finestra, bisogna aggiungere l'attributo `target="_blank"`. L'attributo `target` viene utilizzato anche in altri casi, ma per il momento limitiamoci a quanto detto, poi nel capitolo inerente ai frames (capitolo 9) vedremo altri casi di funzionamento.

La nostra pagina di esempio diventerà quindi così:

```
<a href="http://motogastro.ch" target="_blank">clicca qui</a>
```

Per aprire tutti i link della nostra pagina su un determinato target dobbiamo inserire nella testa del documento html (quindi tra i tag `<head>`) la seguente stringa:

```
<head>  
...  
<base target="_blank" />  
</head>
```

Questa funzionalità ci eviterà di inserire per ogni link l'attributo `target`, che potremo omettere.

6.3 I LINK ANCORA

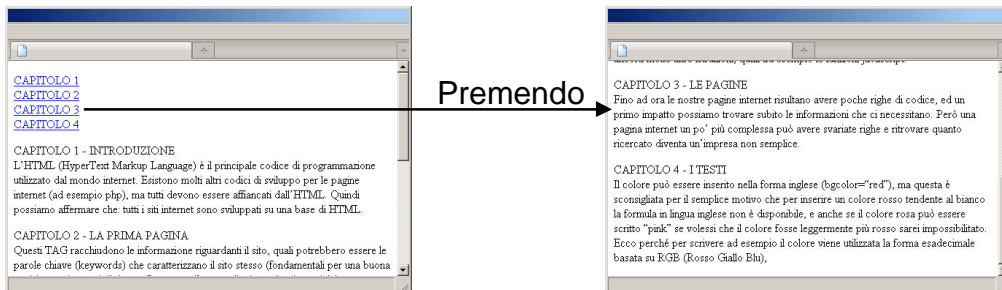
Mettiamo di avere una pagina internet di svariate righe (in visualizzazione) con al suo inizio un menù per selezionare i diversi capitoli, in questo caso il link come visto fino ad ora non andrebbe bene, perchè non necessitiamo di aprire una nuova pagina (in una nuova finestra, o all'interno della stessa) ma necessitiamo di **spostarci nella nostra attuale pagina** al capitolo XY.

Qui vengono dunque utilizzati i link detti ancora. Il puntamento (il link) viene aperto come un link normale: `capitolo 1`, ma dopo il segno di = viene inserito il cancelletto (#), questo simbolo informa il codice HTML che il collegamento deve venir puntato sulla stessa pagina. A questo punto HTML ricerca (nella stessa pagina) l'ancora chiamata `capitolo1`, quindi da parte nostra in un determinato punto del documento dobbiamo inserire tale parametro, così facendo `Capitolo 1 - Introduzione`.

Vediamo un esempio nella pagina successiva:

6.3 Esempio

```
...  
<body>  
<p>  
<a href="#capitolo1">CAPITOLO 1</a><br />  
<a href="#capitolo2">CAPITOLO 2</a><br />  
<a href="#capitolo3">CAPITOLO 3</a><br />  
<a href="#capitolo4">CAPITOLO 4</a><br />  
</p>  
<p>  
<a name="capitolo1">CAPITOLO 1 - INTRODUZIONE</a><br />  
  L'HTML è il principale codice ... (continua)  
<a name="capitolo2">CAPITOLO 2 - LA PRIMA PAGINA</a><br />  
  La prima pagina HTML ... (continua)  
<a name="capitolo3">CAPITOLO 3 - LE PAGINE</a><br />  
  Il TAG body è quanto possiamo ... (continua)  
<a name="capitolo4">CAPITOLO 4 - I TESTI</a><br />  
  I testi, o meglio i font, sono quelli ... (continua)  
</p>  
</body>
```



6.4 INVIARE UNA EMAIL

Tramite i link abbiamo la possibilità di far inviare una e-mail tramite il programma di posta elettronica predefinito installato sul nostro computer. Mettiamo di voler scrivere una e-mail a `info@motogastro.ch`, procederemo a creare un link nel seguente modo:

```
<a href="mailto:info@motogastro.ch">mandami una e-mail</a>
```

Quindi inserendo prima dell'indirizzo e-mail l'istruzione *mailto* avremo questa funzionalità.

6.5 I COLORI DEI LINK

È possibile modificare il colore dei link e/o la loro sottolineatura. Questa operazione deve essere eseguita nel TAG `<body>`. Infatti i link possono avere 2 stati:

- Il link da visitare - di colore blu (#0000FF) /istruzione link
- Il link visitato - di colore viola (#800080) /istruzione vlink

Vediamo quindi il codice del TAG <body> per la modifica dei link e dei vlink:

```
<body link="#00FF00" vlink="#FF0000">
```

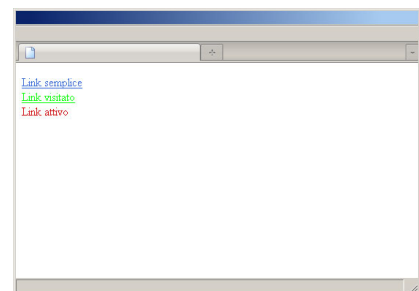
6.6 I LINK ATTIVI

In molti siti internet al passaggio del mouse il link cambia colore, oppure cambia la formattazione. Bene, questi effetti con il solo codice HTML non è possibile riprodurli, però tramite i CSS (Cascading Style Sheets) integrati possiamo riprodurlo.

I CSS sono parti di codice che ci permettono di modificare una parte di pagina con una sola stringa di codice, cioè possiamo inserire un blocco di istruzioni (normalmente nei TAG <head>) le quali non dovranno più essere riscritte all'interno del TAG <body> e tutti gli elementi associati prenderanno tali modifiche. Ulteriori informazioni inerenti i CSS saranno descritte nel capitolo 12.

6.6 Esempio

```
...
<head>
  ...
  <style type="text/css">
    <!-- a:link dei CSS corrisponde a link nel TAG body -->
    a:link {
      color: #336add;
      text-decoration: underline;
      background-color: #B2FF99;
    }
    <!-- a:visited dei CSS corrisponde a vlink nel TAG body -->
    a:visited {
      color: #00FF00;
      text-decoration: underline;
    }
    <!-- a:hover dei CSS non può essere riprodotto solo con HTML -->
    a:hover {
      color: #d20c0c;
      text-decoration: none;
    }
  </style>
</head>
<body>
<p>
  <a href="001.html">Link semplice</a><br />
  <a href="002.html">Link visitato</a><br />
  <a href="003.html">Link attivo</a><br />
</p>
</body>
...
```



Oltre a queste impostazioni per i link tramite CSS possiamo modificare anche il colore di background, ci basta inserire la seguente stringa: background-color: "colore";

7. LE IMMAGINI

7.1 PREMESSA

Abbiamo già visto come inserire un'immagine di sfondo (capitolo 3.2), ma in questo capitolo vedremo come inserire un'immagine tra il testo con la possibilità di parametrizzarne l'altezza o la larghezza, di effettuare un'immagine che funga da link, e daremo un'occhiata alle mappe di immagini.

Anche per le immagini inserite nel testo valgono le regole da rispettare per le immagini di sfondo (estensione e dimensioni in MB).

7.2 INSERIMENTO IMMAGINE

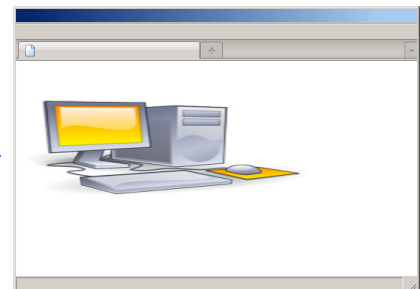
Per inserire un'immagine il TAG da utilizzare è ``, dove *percorso* sta per il "luogo" dove si trova la nostra immagine, come già visto questa può trovarsi sul nostro sito internet o su un sito internet determinato.

Inserendo l'immagine `computer.jpg` (utilizzata nel capitolo 3.2) nel modo come sopra descritto (e mettendo che l'immagine sia allo stesso livello della nostra pagina) questa verrà riprodotta nelle sue dimensioni reali. Possiamo però modificare la grandezza dell'immagine direttamente con HTML, utilizzando gli attributi *width* (per la larghezza) e *height* (per l'altezza).

Consiglio: è sempre meglio avere l'immagine più piccola di quello che si vuole visualizzare a schermo e non viceversa, ridurre le dimensioni (in pixel) dell'immagine con HTML non ne riduce il peso (in MB).

7.2 Esempio

```
...  
<body>  
<p>  
    
</p>  
</body>  
...
```



Inserendo l'attributo `alt=""` si ha la possibilità di visualizzare un messaggio al passaggio del mouse.

7.3 IMMAGINI E LINK

Come per il testo anche sulle immagini è possibile inserire un collegamento ipertestuale (link). Per default le immagini quando diventano dei link hanno un bordo di 1 pixel e questo può essere fastidioso, quindi bisogna utilizzare ed impostare a 0 il bordo, come da esempio sotto riportato:

```
<a href="..."></a>
```

Il bordo dell'immagine può essere ovviamente impostato anche ad un numero superiore ad 1, in questo caso anche se l'immagine non "fa" da link il bordo viene visualizzato.

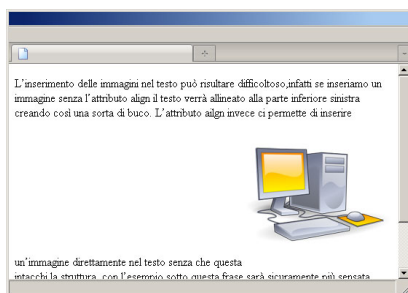
7.4 IMMAGINI E TESTO

L'inserimento delle immagini nel testo può risultare difficoltoso, infatti se inseriamo un'immagine senza l'attributo align il testo verrà allineato alla parte inferiore sinistra creando così una sorta di buco. L'attributo align invece ci permette di inserire un'immagine direttamente nel testo senza che questa intacchi la struttura, con l'esempio sotto questa frase sarà sicuramente più sensata.

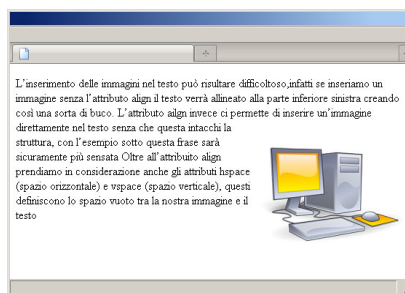
Oltre all'attributo align prendiamo in considerazione anche gli attributi hspace (spazio orizzontale) e vspace (spazio verticale), questi definiscono lo spazio vuoto tra la nostra immagine e il testo.

7.4 Esempio

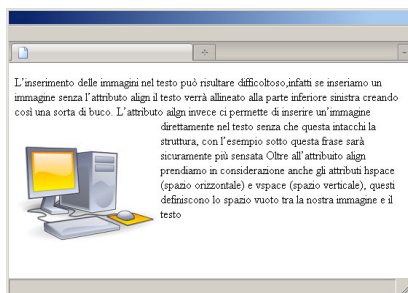
.. ...



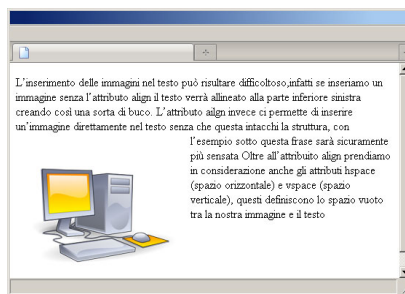
.. ...



.. ...



.. ...



7.5 LE MAPPE D'IMMAGINI

Le mappe d'immagini possono risultare molto utili nel caso avessimo un'immagine nella quale vogliamo far sì che al passaggio del mouse su di una determinata area questa sia linkabile.

Mettiamo di avere una fotografia dei nostri colleghi di ufficio ed al passaggio del mouse sulla faccia di uno di questi vi sia il proprio nome con collegamento al suo sito internet.

Questo capitolo è stato inserito più che altro per rendervi attenti che vi è anche questa possibilità però non spiegherò più di così il loro funzionamento perché risulta troppo laborioso.

Potete comunque scaricare un programma apposito (esempio MAPedit) e provare a "giocare" un po' con questa possibilità d'implementazione.

8. LE TABELLE

8.1 PREMESSA

I programmatori di HTML sanno bene che le tabelle sono una parte portante del buon funzionamento del codice. Infatti tramite il TAG **<table>** abbiamo la possibilità, non solo di creare dei dati strutturati (come ad esempio utilizzando microsoft excel), ma possiamo impaginare il layout in modo da rendere la nostra pagina strutturata come l'abbiamo pensata (ad esempio con un menu verticale sulla parte sinistra ed con il testo sulla parte destra).

Senza l'utilizzo delle tabelle queste operazioni sarebbero impossibili, o meglio, molto più complicate da riprodurre con il solo linguaggio HTML. Con l'arrivo di XHTML l'utilizzo delle tabelle per questo scopo (strutturare il layout) sta pian piano "morendo", questo perché tramite i CSS possiamo riprodurre la medesima struttura con molte meno righe di codice.

Comunque ancora oggi, molti sviluppatori utilizzano le tabelle per strutturare il layout, e tutti i programmatori utilizzano ancora questo TAG (**<table>**) per creare delle tabelle di dati.

8.2 STRUTTURA BASE

Per inserire una tabella all'interno della nostra pagina HTML viene utilizzato il TAG **<table>**. Ogni tabella che si rispetti deve avere delle righe: le quali vengono riprodotte con il TAG **<tr>** (abbreviazione di table row); e delle celle che "tradotte" in HTML vengono inserite con il TAG **<td>** (table data), le celle sono infatti il risultato tra righe e colonne.

Una tabella se inserita senza l'attributo *border* a video (nel browser) avrà un bordo invisibile, a dipendenza di cosa si vuole riprodurre questo può essere anche un bene, comunque per l'esempio sotto riportato utilizzeremo un bordo di 2 pixel.

8.2 Esempio

```
...  
<body>  
  <table border="2">  
    <tr>  
      <td>Cella 1</td>  
      <td>Cella 2</td>  
    </tr>  
    <tr>  
      <td>Cella 3</td>  
      <td>Cella 4</td>  
    </tr>  
  </table>  
</body>  
...
```



Come possiamo notare con il TAG **<table>** non è necessario inserire il TAG di paragrafo **<p>**.

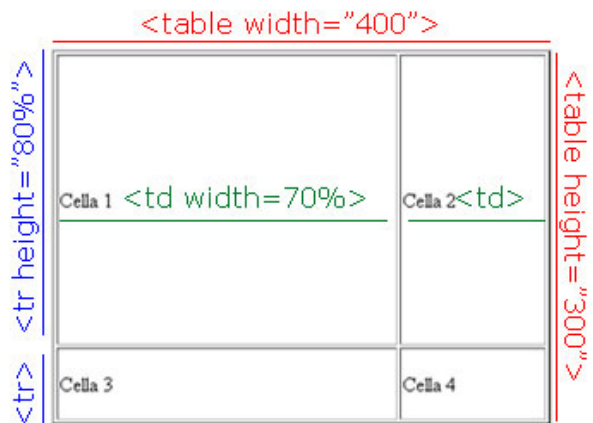
8.3 DIMENSIONI

Come visto nell'esercizio alla pagina precedente, se non impostato, sia la tabella (<table>), sia la riga (<tr>), sia le celle (<td>) prendono la dimensione del contenuto (il testo).

Queste dimensioni ovviamente possono essere sostituite, sia per la larghezza (attributo **width**) sia per altezza (attributo **height**). Le dimensioni possono essere espresse in pixel (esempio <table width="200">) sia in percentuale rispetto alla dimensione della finestra del browser (esempio <table width="80%">).

8.3 Esempio

```
...  
<table border="2" width="400" height="300">  
  <tr height="80%">  
    <td width="70%">Cella 1</td>  
    <td>Cella 2</td>  
  </tr>  
  <tr>  
    <td>Cella 3</td>  
    <td>Cella 4</td>  
  </tr>  
</table>  
...
```



Come si può notare nell'esempio superiore:

- <tr height="80%"> corrisponde a 240 pixel (300/100*80)
- <tr> corrisponde a 60 pixel (300-240)
- <td width="70%"> corrisponde a 280 pixel(400/100*70)
- <td> corrisponde a 120 pixel (400-280)

Regole importanti:

- Le celle (<td>) della 2a riga (<tr>) prendono **sempre** le dimensioni delle celle della 1a riga
- Le dimensioni che valgono sono **sempre** quelle superiori, prima <table>, poi <tr> e poi <td>
- L'attributo **height** (altezza) non in tutti i browser è supportato

8.4 L'ALLINEAMENTO DELLE CELLE

Come visto precedentemente l'allineamento del contenuto delle celle se non impostato è in orizzontale a sinistra (left) e in verticale al centro (middle).

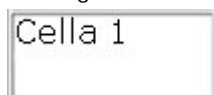
Per allineare in **orizzontale** viene usato l'attributo **align="..."** (possibili: left, center, right)

Per allineare in **verticale** viene usato l'attributo **valign="..."** (possibili: top, middle, bottom)

Ogni cella può essere allineata a piacimento.

8.4 Esempi

```
<td align="left" valign="top">
```



```
<td align="center" valign="middle">
```



```
<td align="right" valign="bottom">
```



8.5 IL RAGGRUPPAMENTO

A volte possiamo avere la necessità di creare una tabella con una sola cella superiore e più celle inferiori, in questo caso utilizzeremo il TAG **<colspan>**. Oppure potremmo avere necessità di creare una tabella con una cella verticale e al suo lato più righe di celle, quindi utilizzeremo il TAG **<rowspan>**. Vediamo il funzionamento con gli esempi sottostanti.

8.5.1 Esempio TAG **<colspan>**

```
...  
<table border="2">  
  <tr>  
    <td colspan="2">Cella 1 - riga 1 </td>  
    <td>Cella 2 - riga 1 </td>  
  </tr>  
  <tr>  
    <td>Cella 3 - riga 2 </td>  
    <td>Cella 4 - riga 2 </td>  
    <td>Cella 5 - riga 2 </td>  
  </tr>  
  <tr>  
    <td colspan="3">Cella 6 - riga 3 </td>  
  </tr>  
</table>  
...
```

Cella 1 - riga 1		Cella 2 - riga 1
Cella 3 - riga 2	Cella 4 - riga 2	Cella 5 - riga 2
Cella 6 - riga 3		

8.5.2 Esempio TAG **<rowspan>**

```
...  
<table border="2">  
  <tr>  
    <td>1 - colonna 1 </td>  
    <td rowspan="2">colonna 2 </td>  
    <td>1 - colonna 3 </td>  
  </tr>  
  <tr>  
    <td>2 - colonna 1 </td>  
    <td>2 - colonna 3 </td>  
  </tr>  
</table>  
...
```

1 - colonna 1	colonna 2	1 - colonna 3
2 - colonna 1		2 - colonna 3

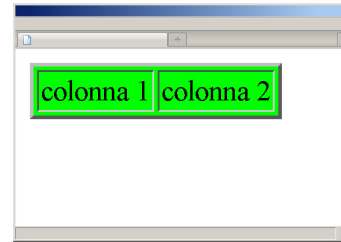
8.6 I COLORI

Oltre a quanto abbiamo visto fino ad ora, abbiamo la possibilità dare alle nostre tabelle un colpo di classe aggiungendo dei colori. Possiamo definire un colore per i bordi ed un colore di sfondo (sia per la tabella, sia per la riga, sia per la cella), oppure anche impostare un'immagine di sfondo (che come per il TAG **<body>** viene ripetuta per tutta la tabella). Ovviamente le celle se non impostato prendono il colore di sfondo della tabella.

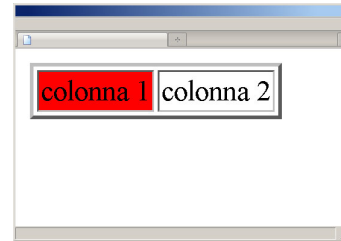
Gli esempi nella pagina successiva

8.6.1 Esempi colori di sfondo

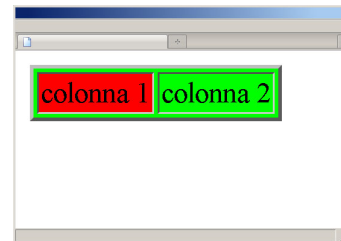
```
...  
<table border="2" bgcolor="#00ff00" >  
  <tr>  
    <td>colonna 1</td>  
    <td>colonna 2</td>  
  </tr>  
</table>  
...
```



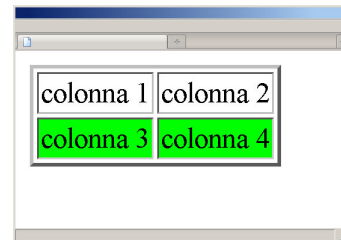
```
...  
<table border="2" >  
  <tr>  
    <td bgcolor="#ff0000" >colonna 1</td>  
    <td>colonna 2</td>  
  </tr>  
</table>  
...
```



```
...  
<table border="2" bgcolor="#00ff00" >  
  <tr>  
    <td bgcolor="#ff0000" >colonna 1</td>  
    <td>colonna 2</td>  
  </tr>  
</table>  
...
```

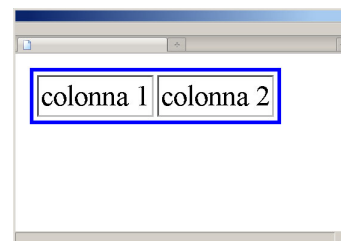


```
...  
<table border="2" >  
  <tr>  
    <td>colonna 1</td>  
    <td>colonna 2</td>  
  </tr>  
  <tr bgcolor="#00ff00" >  
    <td>colonna 3</td>  
    <td>colonna 4</td>  
  </tr>  
</table>  
...
```



8.6.2 Esempi colori di bordo

```
...  
<table border="2" bordercolor="#0000ff" >  
  <tr>  
    <td>colonna 1</td>  
    <td>colonna 2</td>  
  </tr>  
</table>  
...
```



Anche in questo caso è possibile applicare il colore dei bordi ai TAG <table>, <tr> e <td>

8.7 ALTRI ATTRIBUTI DI TABLE

Oltre agli attributi visti fino ad ora ve ne sono altri 2 molto importanti.

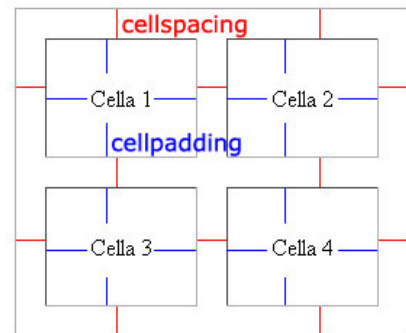
L'attributo **cellpadding** viene utilizzato per distanziare il contenuto della cella dal suo bordo, questo se non specificato ha un valore uguale a 0.

L'attributo **cellspacing** invece serve a distanziare una cella dall'altra, o da cella a bordo, di default questo parametro è settato ad 1.

Ma vediamo il tutto con l'esempio seguente.

8.7 Esempio cellpadding e cellspacing

```
...  
<table border="1" cellspacing="20" cellpadding="30">  
  <tr>  
    <td>Cella 1</td>  
    <td>Cella 2</td>  
  </tr>  
  <tr>  
    <td>Cella 3</td>  
    <td>Cella 4</td>  
  </tr>  
</table>  
...
```



9. I FRAMES

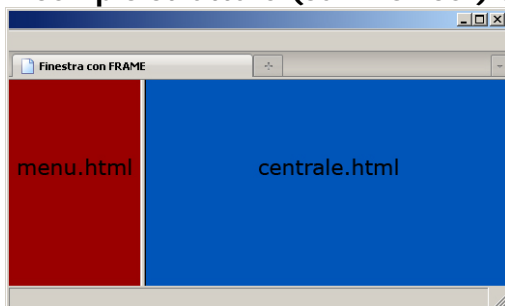
9.1 COSA SONO

Tramite i frames (o cornici) abbiamo la possibilità di visualizzare più pagine HTML in una sola finestra del browser. Mettiamo di voler creare una pagina web, che sul lato sinistro abbia una sorta di menu fisso e sulla parte destra-centrale abbia il contenuto variabile a seconda della voce di menu selezionata. Per far questo dobbiamo utilizzare i frames (e quindi 2 pagine distinte).

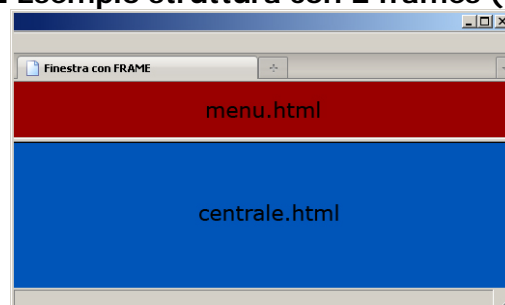
Un layout del genere, possiamo rappresentarlo anche tramite le tabelle (come visto nel capitolo precedente), ma a differenza usando i frames non dobbiamo inserire in ogni pagina le stesse righe di codice (per esempio per visualizzare i menu). Bensì creeremo 2 pagine distinte, 1 per il menù (menu.html) e una per la parte variabile (centrale.html).

Ovviamente se con il menu vogliamo visualizzare più pagine, creeremo più pagine per la centrale.

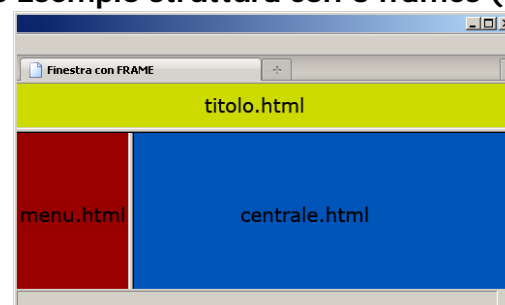
9.1.1 Esempio struttura (sul Browser) con 2 frames (orizzontali)



9.1.2 Esempio struttura con 2 frames (verticali)



9.1.3 Esempio struttura con 3 frames (orizzontali e verticali)



9.2 COME INSERIRE I FRAMES

A differenza di quanto visto fino ad ora, una pagina contenente dei frame non avrà la formula:

```
<html><head><head><body></body></html>
```

bensì i TAG *body* vengono sostituiti dai TAG *frameset*:

```
<html><head><head><frameset></frameset></html>
```

Questa formula è l'unica eccezione nella quale una pagina HTML è definita in modo diverso da quanto visto fino ad ora.

Il TAG *<frameset>* è uno di quei TAG che senza attributi non serve a nulla (come anche visto per il TAG **). Gli attributi che devono essere specificati sono **cols**= "... " e/o **rows**= "...".

L'attributo **rows** (righe) viene utilizzato per suddividere la finestra del browser in senso orizzontale, mentre l'attributo **cols** (colonne) viene utilizzato per suddividere la finestra del in senso verticale.

All'interno (= "...") degli attributi rows e cols dobbiamo specificare le dimensioni delle suddivisioni nelle quali inserire le pagine. Mettiamo di voler suddividere la finestra in due pagine, inseriremo:

```
<frameset cols="dimensione1, dimensione2">      suddivisione verticale  
<frameset rows="dimensione1, dimensione2">      suddivisione orizzontale
```

Le dimensioni possono essere espresse in pixel oppure in percentuale:

```
In pixel: <frameset="250,*">          * = lo spazio mancante della finestra  
In percentuale: <frameset="15%,85%">  dove la finestra equivale al 100%
```

Una volta definite le divisioni (nell'esempio superiori: 2) e le dimensioni sulle quali vogliamo operare, dobbiamo specificare quali pagine inserire in queste cornici (frame). Quindi dobbiamo già conoscere il nome delle pagine, perciò queste devono essere già state create precedentemente. Supponiamo di aver creato una pagina di nome *menu.html* ed un'altra di nome *centrale.html*:

```
<frame name="menu" src="menu.html" />  
<frame name="centrale" src="centrale.html" />
```

L'attributo **name** (non obbligatorio ma consigliato) specifica il nome della divisione. Mentre **src** viene utilizzato per specificare il percorso della pagina da visualizzare.

9.2.1 Esempio codice con 2 frames (orizzontali)

```
<html>  
<head>  
  <title>Finestra con FRAME orizzontali</title>  
</head>  
<frameset rows="15%,85%">  
  <frame name="menu" src="menu.html" />  
  <frame name="centrale" src="centrale.html" />  
</frameset>  
</html>
```

9.2.2 Esempio codice con 2 frames (verticali)

```
<html>  
<head>  
  <title>Finestra con FRAME verticali</title>  
</head>  
<frameset cols="150,*">  
  <frame name="menu" src="menu.html" />  
  <frame name="centrale" src="centrale.html" />  
</frameset>  
</html>
```

9.2.3 Esempio codice con 3 frames (orizzontali e verticali) e standard w3c

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//IT"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Finestra con diversi FRAME</title>
</head>
<frameset rows="15%,85%">
  <frame name="titolo" src="titolo.html" />
  <frameset cols="20%,80%">
    <frame name="menu" src="menu.html" />
    <frame name="centrale" src="centrale.html" />
  </frameset>
</frameset>
</html>
```

9.3 PARAMETRIZZARE I FRAME

Anche i TAG <frameset> e rispettivamente i TAG <frame> hanno ulteriori parametrizzazioni a quanto visto fino ad ora. Vediamone per punti.

9.3.1 TAG <frameset>

- Colorare il bordo del frame: utilizzare l'attributo **bordercolor**
<frameset cols="..." **bordercolor="#01f5d8"**>
- Eliminare il bordo:
<frameset cols="..." **frameborder="no"**>

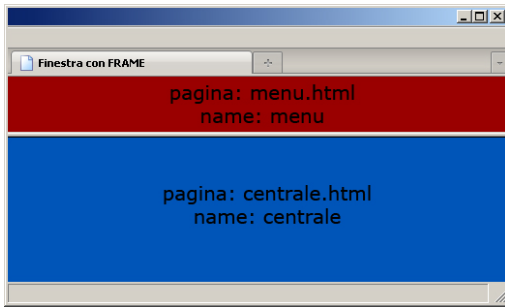
9.3.2 TAG <frame>

- Togliere la barra verticale o orizzontale (anche se il contenuto è più grande della conica)
<frame src="..." **scrolling="no"**>
- Impedire il ridimensionamento del frame:
<frame src="..." **noresize**>
- Impostare una distanza orizzontale dall'estremità sinistra del frame al suo contenuto
<frame src="..." **marginwidth="20"**> *dimensioni in pixel*
- Impostare una distanza verticale dall'estremità superiore del frame al suo contenuto
<frame src="..." **marginheight="20"**> *dimensioni in pixel*

9.4 L'ATTRIBUTO DI FRAME: NAME

Al capitolo 9.2 è stato consigliato di impostare sempre l'attributo **name** del TAG frame. Questo consiglio è stato dato perché tra i frame deve (quasi sempre) esserci la possibilità di spostarsi tra di essi.

Mettiamo di avere un frame con nel suo contenuto un menu, e un altro frame con il contenuto descrittivo del link premuto nel menu, come immagine a pagina seguente.



A questo punto per caricare un link (da menu.html) al frame centrale (pagina centrale.html) l'espressione che bisogna inserire è la seguente:

```
<a href="005.html" target="centrale">Apri la pagina 005</a>
```

Vediamo qui che tramite l'attributo **target**, diciamo al nostro link dove far apparire la pagina 005.html nel frame nominato *centrale*. Se questo attributo viene omesso la pagina 005.html viene aperta nel nostro frame nominato *menu*, questo perché il link senza l'attributo **target** apre la pagina nella stessa pagina di dove si trova il collegamento.

Ecco qui perché è stato consigliato di impostare per tutti i frame un attributo **name**.

9.5 L'ATTRIBUTO TARGET DEI TAG <A>

Abbiamo già visto nel capitolo 6.2 riguardante i link, che tramite l'attributo **target** impostato a **_blank** abbiamo la possibilità di aprire una nuova finestra del browser.

Nel capitolo 9.4 abbiamo invece visto che impostando il nome del frame la nostra pagina viene aperta nella cornice con quel nome (*target="nome del frame"*).

L'attributo **target** ha comunque altre impostazioni interessanti, ad esempio possiamo avere una finestra con frame, e se vogliamo che in questa finestra venga aperta una nuova pagina senza frame dovremo settare l'attributo **target** nel seguente modo:

```
<a href="005.html" target="_top">Apri la pagina 005</a>
```

Mettiamo ora di avere un menu con 10 voci (link) e tutte questi 10 collegamenti devono essere aperti nel frame "centrale", inserire per ogni link l'attributo **target="centrale"** non è la miglior soluzione (anche se funzionante), in questo caso possiamo specificare tra i TAG <head> la seguente stringa: `<base target="centrale" />`

Così facendo tutti i link verranno aperti nel target nominato *centrale*.

Questa funzione funziona anche per le pagine senza frame. Se per esempio volessimo che tutti i collegamenti della nostra pagina vengano aperti in nuova finestra inseriremmo (sempre tra i TAG <head>) la stringa: `<base target="_blank" />`

9.5 Esempio TAG <base>

```
<html>
<head>
  <title></title>
  <base target="centrale" />
</head>
<body>
<p>
  <a href="005.html">Apri la pagina nel frame centrale</a>
  ...
```

9.6 COMPATIBILITÀ CON I MOTORI DI RICERCA

Inizio dicendo che vi sono dei TAG da inserire nelle nostre pagine HTML per rendere possibile la visualizzazione nei motori di ricerca del nostro sito internet, ma in questo capitolo non ne parleremo ancora.

Oltre a questi TAG i motori di ricerca analizzano il testo del nostro sito internet ed estrapolano delle parti che poi visualizzeranno dopo la ricerca. Purtroppo il contenuto di un sito internet strutturato con dei frame non viene preso in considerazione dai motori di ricerca. Questo "problema" a fatto si che molti sviluppatori (o webmaster) hanno discriminato i frame e questi su molti siti internet sono stati rimossi e rimpiazzati con un layout strutturato con delle tabelle.

In verità il problema può essere risolto utilizzando i TAG `<no frames>`, infatti **i motori di ricerca** in caso "incontrino" un TAG `<frameset>` spesso (non sempre) ignorano quanto viene inserito tra i 2 TAG. **NON ignorano però quanto descritto tra i TAG `<no frames>`.**

9.5 Esempio TAG `<no frames>`

```
<html>
<head>
  <title></title>
</head>
<frameset cols="200,*">
  <frame name="menu" src="menu.html" />
  <frame name="centrale" src="centrale.html" />
  <no frames>
  Questa è la mia pagina HTML
</no frames>
</frameset>
</html>
```

Quindi in questo caso quanto descritto nelle pagine `menu.html` e/o `centrale.html` non viene riconosciuto (o meglio: rilevato) dai motori di ricerca, ma questi rileveranno unicamente il contenuto tra i due TAG `<no frames>`.

Personalmente non utilizzo più un layout strutturato su frame, proprio perché sono uno di quei webmaster che tempo fa hanno discriminato quest'implementazione; ma con le nuove innovazioni del codice HTML e delle funzioni javascript (capitolo 9.7), questo problema è stato risolto definitivamente (per il momento).

A proposito, **il contenuto dei TAG `<no frames>` non viene visualizzato quando si naviga sul browser.**

9.7 DA FARE IN CASO DI UTILIZZO DI UNA STRUTTURA A FRAME

Nel caso vogliate creare un sito web utilizzando i frame fate però attenzione che i motori di ricerca potrebbero estrapolare la pagina `centrale.html` senza visualizzare la `menu.html` (e viceversa). Quindi bisogna impostare una funzione javascript che non permetta l'apertura di una pagina `centrale.html` senza una pagina `menu.html` (quindi senza frame), come segue:

```
<script language="javascript">
  if (window.location == parent.location) { location.href = "iniziale.html"; }
</script>
```

Questa funzione javascript è da inserire tra i TAG `<head>`, e in poche parole dice all'incirca così: *se la pagina corrente non ha frames, apri questa pagina all'interno della pagina iniziale.html*

Quindi questa funzione non è da inserire nella pagina contenente i `<frameset>` (in questo caso `iniziale.html`), ma al contrario è da inserire in tutte le altre pagine.

10. I FORM

10.1 COSA SONO

I form (o moduli) permettono a chi visita il nostro sito internet di inviare dei dati, predefiniti da parte del webmaster, tramite e-mail.

La definizione sopra riportata non è del tutto corretta, infatti tramite i form abbiamo la possibilità di inviare dei dati anche ad altre parti (ad esempio ad un database), però con il solo linguaggio HTML la possibilità unica è quella di inviare delle e-mail. Ovviamente questo solo se sul computer del visitatore è configurato un client di posta (esempio Outlook).

Solitamente una struttura con i form ha al suo interno 2 pagine, ovvero:

- Una pagina che serve per l'inserimento dei dati da inviare
- Una pagina che serve per inviare tali dati tramite veri e propri linguaggi di programmazione WEB (ad esempio php)

In questo capitolo però vedremo unicamente la pagina 1, con al suo interno i parametri che permettono di inviare una e-mail. Non faremo accenno ad altre impostazioni proprio perché sarebbe inutile spiegarle senza poterle mettere in pratica.

10.1 Esempio: una pagina con form

Nome *	<input type="text"/>
Cognome *	<input type="text"/>
E-mail *	<input type="text"/>
Moto	<input type="text"/>
Testo	<input type="text"/>

10.2 COME SI INSERISCONO

Come molti altri TAG di HTML (vedi capitolo sugli elenchi) per l'inserimento dei form è necessario utilizzare più elementi (TAG). Prima di tutto bisogna definire che stiamo utilizzando i moduli, quindi apriamo il seguente TAG `<form>`, in questo specifichiamo già da subito l'attributo che permette l'invio di e-mail come da esempio:

```
<form action="mailto:info@sito.com?subject=mail dal sito" name="modulo" method="post">  
...  
</form>
```

Bene qui possiamo vedere che questo form invierà una mail a info@sito.com con oggetto della e-mail [mail dal sito](#) e sarà inviata tramite il metodo `post`. Esiste la possibilità di inviare un form tramite il metodo `get` ma per le e-mail viene utilizzato il metodo `post`, quindi sappiate unicamente che in una **form bisogna aggiungere l'attributo `method="post"`**.

All'interno dei TAG `<form>` possiamo (dobbiamo) inserire altri TAG chiamati `<label>` e `<input>`, dove `label` sta per la descrizione del TAG `input`, come da esempio:

10.2 Esempio: `<label>` e `<input>`

<code><label></code>	<code><input></code>
Nome *	<input type="text"/>

10.3 I TAG INPUT

Come dice lo stesso TAG `<input>`, questo serve per inserire dei dati. I TAG `<input>` devono essere associati all'attributo `type`, perché senza attributi questo TAG è inutilizzabile.


Bisogna fare attenzione perché un TAG `<input>` può avere molte personalizzazioni (tramite attributo `type`). Ad esempio se volessimo creare un casella per l'inserimento di testo dobbiamo utilizzare l'attributo `type` settato a `text` (esempio capitolo 10.3.1), ma sempre con `<input>` possiamo creare un bottone, con `type="button"` (capitolo 10.3.4).

10.3.1 Input type text

Tramite `input` settato a `text` abbiamo la possibilità (come specificato precedentemente) di creare un sorta di casella di testo per l'inserimento di dati di tipo testo. Con un codice come il seguente:

```
<input type="text" name="campo_nome" value="Inserisci il nome" />
```

Abbiamo un risultato a video che permette, appunto, a chi visita la nostra pagina WEB di inserire del testo, come immagine sotto riportata:

A screenshot of a web browser showing a single text input field. The field is rectangular with a thin border and contains the placeholder text "Inserisci il nome" in a light gray font.

Con il settaggio di `type` a `text` possiamo specificare i seguenti parametri:

- `<input type="text" name="campo_nome"`
serve quando riceviamo il mail (quanto inserito fa riferimento al `campo_nome`)
- `<input type="text" value="Inserisci il nome"`
Il testo inserito nell'attributo `value` viene riportato all'interno della casella
- `<input type="text" maxlength="50"`
Il numero massimo di caratteri che possono essere inseriti
- `<input type="text" size="250"`
La larghezza (in pixel) della casella

10.3.2 Input type submit

Il settaggio a `submit` del TAG `<input>` crea un pulsante. Questo settaggio in un form è ovviamente obbligatorio (quasi sempre, con una funzione javascript questo può essere tralasciato, ma vediamo più tardi come integrare javascript), infatti senza un `<input type="submit" />` il visitatore non ha la possibilità di inviare i dati del form alla mail specificata.

Quindi quando creiamo una form, bisogna settare anche un TAG `<input>` con attributo `submit` che permette appunto di inviare quanto richiesto. Vediamo un semplice, ma obbligatorio settaggio a `submit`: `<input type="submit" value="Invia la mail" />`

Questo non necessita di spiegazioni. L'attributo `size` non funziona con questo `type`, per dimensionarlo bisogna utilizzare i CSS.

10.3.3 Input type reset

Poche spiegazioni sono necessarie per il settaggio **reset**; questo infatti crea un pulsante simile al `submit` ma a differenza di quest'ultimo permette di cancellare i contenuti inseriti. Ecco qui un semplice esempio: `<input type="reset" value="Cancella tutto" />`

10.3.4 Input type button

Il settaggio di `type="button"` crea un bottone come i precedenti (`submit` e `reset`) ma a differenza di questi senza una funzione javascript apposita (vedere capitolo 10.6) è inutile. Vediamo un semplice esempio: `<input type="button" value="Bottone" />`

10.3.5 Input type image

Questo settaggio ha lo stesso funzionamento del settaggio a *button*, quindi inutile senza una funzione javascript apposita. Come dice l'attributo, questo permette di attribuire ad un'immagine lo stesso funzionamento di un bottone. Vediamo un semplice esempio:

```
<input type="image" src="invia.jpg" alt="invia il form" width="120" height="30" />
```

10.3.5 Input type radio

Gli input type radio vengono utilizzati per permettere all'utente di effettuare una scelta di opzioni definite dal webmaster. Vediamo un esempio:

Signora
 Signor

```
<input type="radio" name="cortesia" value="Signora" /><label>Signora</label>  
<br />  
<input type="radio" name="cortesia" value="Signor" /><label>Signor</label>
```

10.3.5 Input type checkbox

A differenza dei type radio tramite i type checkbox il visitatore può selezionare più di una voce e quindi anche l'attributo name deve essere diverso (infatti nei type radio il name è uguale per tutte le voci). Vediamo un esempio:

```
<input type="checkbox" name="patate" value="patate" /><label> Patate</label>  
<input type="checkbox" name="carote" value=" carote"/><label>Carote</label>  
<input type="checkbox" name="pasta" value="Pasta" checked="checked" />  
<label>Pasta</label>
```

Patate
 Carote
 Pasta

10.4 IL TAG TEXTAREA

In un form oltre ad inserire un dei dati "anagrafici" quali nome, cognome, ... possiamo inserire un campo che serve per l'inserimento del testo del messaggio che vogliamo inviare. Dobbiamo quindi creare un campo di inserimento testo più grande (e magari su più linee), utilizziamo quindi il TAG <textarea>:

```
<textarea name="messaggio" rows="6" cols="50">  
Inserisci qui il tuo messaggio  
</textarea>
```

L'attributo rows indica il numero di righe (orizzontali) presenti nel campo textarea, mentre l'attributo cols specifica quante colonne (o caratteri) possono essere inseriti in una riga.

10.5 I TAG SELECT

Oltre ai campi input e textarea esistono i TAG select, questi riproducono un campo con menu a tendina, vediamo un'esempio:

```
<select name="animale">  
<option value="cane" selected="selected">Cane</option>  
<option value="gatto">Gatto</option>  
<option value="pesce">Pesce</option>  
</select>
```



10.6 CONTROLLO SUI CAMPI (Javascript)

Mettiamo di avere una pagina con un modulo che viene utilizzato per l'invio dei propri dati e di un messaggio personale alla nostra mail info@sito.ch. Come da esempio della pagina seguente.

10.6.1 Pagina per invio di un messaggio (con form)

```
<html>
<head>
  <title>Inviami un messaggio</title>
</head>
<body>
<p>
<form action="mailto:info@sito.ch" name="modulo" method="post">
  <label>Nome</label><input type="text" name="nome" value="" /><br />
  <label>Cognome</label><input type="text" name="cognome" value="" /><br />
  <label>Messaggio</label><textarea name="msg" rows="3" cols="10"></textarea><br />
  <input type="submit" value="Invia" />
</form>
</p>
</body>
</html>
```

10.6.2 Pagina per invio di un messaggio e controllo campi vuoti (javascript)

```
<html>
<head>
  <title>Inviami un messaggio</title>
  <script language="javascript" type="text/javascript">
    <!--
      function controllo()
      {
        if (document.modulo.nome.value == "")
        {
          alert ('Attenzione. Il campo "Nome" è obbligatorio');
          document.modulo.nome.focus();
          return false ;
        }
        if (document.modulo.msg.value == "")
        {
          alert ('Attenzione. Il campo "Messaggio" è obbligatorio');
          document.modulo.msg.focus();
          return false ;
        }
      }
    </script>
</head>
<body>
<p>
<form action="mailto:info@sito.ch" onsubmit="return controllo()" name="modulo" method="post">
  <label>Nome</label><input type="text" name="nome" value="" /><br />
  <label>Cognome</label><input type="text" name="cognome" value="" /><br />
  <label>Messaggio</label><textarea name="msg" rows="3" cols="10"></textarea><br />
  <input type="submit" value="Invia" />
</form>
</p>
</body>
</html>
```

Tramite questa funzione javascript all'invio del form (`<input type="submit" />`), questo prima di inviare la mail verifica la funzione javascript controllo (grazie all'attributo inserito nel TAG di apertura `<form>`: `onsubmit="return controllo()"`).

La funziona javascript inserita tra i TAG di testa (`<head>`) non fa altro che verificare che i 2 campi (nome e msg) della stessa pagina (`document.`) del form `modulo` non siano nulli (`value == ""`).

Se non sono nulli parte la mail, in caso contrario tramite la funzione `focus()` il mail non parte e il puntatore del mouse si "inserisce" nel campo vuoto.

11. I TAG META

11.1 COSA SONO

I TAG meta (o metatag) sono degli elementi di HTML che permettono di "aiutare" i motori di ricerca a trovare le informazioni delle vostre pagine web.

Ho detto "aiutare" perché i motori di ricerca sono in continua evoluzione ed oltre a leggere i metatag cercano le informazioni nel codice (ricordate l'attributo di body lang="it" ?), anche per questo una pagina web deve essere il più possibile redatta con buon codice HTML.

I TAG <meta /> **vengono inseriti nella testa del nostro documento**, quindi tra gli <head> (e non tra i <body>).

I metatag non cambiano l'aspetto della nostra pagina WEB, ed è consigliabile inserirli in più pagine possibili (e possibilmente diversamente impostati a dipendenza dei contenuti).

11.2 UNA PAGINA CON METATAG

È inutile spiegare passo passo tutti i metatag, quindi qui sotto vi inserisco una lista (compilata) di alcuni attributi di questi TAG. Esistono inoltre molti programmi che generano il codice dei metatag basta scrivere in un qualsiasi motore di ricerca la stringa: *metatag generator* e scegliere uno dei risultati. Comunque la lista riportata in seguito è già abbastanza (anche troppo) esaustiva.

11.2 Esempio METATAG

```
<html>
<head>
  <meta name="description" content="Appare come descrizione nei risultati dei motori di ricerca" />
  <meta name="keywords" content="inserire alcune parole chiave separate da una ," />
  <meta name="copyright" content="es: Club Motogastro" />
  <meta name="author" content="Nome Cognome" />
  <meta name="email" content="es: info@motogastro.ch" />
  <meta name="Charset" content="UTF-8" />
  <meta name="Distribution" content="Global" />
  <meta name="Rating" content="General" />
  <meta name="Revisit-after" content="31 Days" />
</head>
...
```

11.3 CONCLUSIONE

Oltre a questi metatag, esistono molti altri modi per permettere ai vostri siti internet di comparire nei risultati forniti dai motori di ricerca. Il più importante è quello della registrazione del vostro sito internet ai principali motori di ricerca. *Ad esempio in google digitate Add URL e aprite la prima pagina proposta*. Alcuni di questi hanno anche un pannello di controllo che permette di specificare ulteriori informazioni inerenti il vostro sito, sempre in google (esempio) esiste una sezione chiamata "strumenti per i webmaster".

Tutte queste regole raramente vi permetteranno di essere tra i primi risultati di un motore di ricerca. La miglior strategia è quella che il vostro sito sia visitato, infatti più un sito internet è visitato e più i motori di ricerca lo analizzeranno.

Quindi, se lo scopo è quello di essere presente nei risultati forniti dai motori di ricerca, **pubblicizzate il più possibile il vostro sito internet**.

Senza dimenticare che esistono anche ulteriori servizi offerti dai motori di ricerca per permettervi di scalare ulteriormente la vetta dei risultati, ma questi sono ovviamente a pagamento.

12. PUBBLICARE IL SITO

12.1 INTRODUZIONE

Una volta che abbiamo creato il nostro sito internet (le nostre pagine HTML) sul nostro computer (o MAC), possiamo finalmente inserirle in internet; questo ci permetterà di esporle al mondo intero. Ricordandoci comunque che in caso volessimo modificare una determinata parte delle nostre pagine, queste dovranno essere modificate sempre sulla nostra postazione e in seguito riscritte (o rimesse) in internet.

Esistono 2 modi per comparire nel web, uno a pagamento e uno gratuitamente. Perché scegliere il metodo a pagamento? Ovviamente perché questo ci permetterà di avere servizi aggiuntivi, e soprattutto il nostro sito potrà avere un URL (ovvero un indirizzo) semplice e come piacerà a noi (se questo indirizzo non è già utilizzato).

Tutti i siti internet hanno (devono avere) le 2 seguenti particolarità:

- **Un nome** (es. <http://www.motogastro.ch> oppure <http://web.ticino.com/cheng-on>)
- **Dello spazio** per l'inserimento dei nostri file HTML (ed. eventuali immagini, ...)

12.2 REGISTRAZIONE (A PAGAMENTO)

13.2.1 Il nome

Il nome del sito internet, viene chiamato **nome di dominio** (web domain), e nell'opzione a pagamento questi hanno la forma seguente: www.proprionome.com.

Dove *proprio nome* è quanto viene da noi scelto, mentre [.com](http://www.com) sta per l'estensione del nostro nome di dominio, di regola in svizzera vengono utilizzate le estensioni di tipo [.ch](http://www.ch), ma nessuno vieta di creare e registrare un sito con estensione [.de](http://www.de) (Germania) oppure [.com](http://www.com) (Commerciale) oppure Questa scelta sta a voi, ricordate però che se registrate un sito internet con, un'estensione [.fr](http://www.fr) nei risultati forniti dai motori di ricerca figurerà che il vostro sito è situato in Francia e non in Svizzera.

Per quanto riguarda le estensioni di tipo [.ch](http://www.ch) il fornitore ufficiale è la SWITCH (Serving Swiss Universities) e potete trovarla all'URL <http://www.switch.ch> oppure <http://www.nic.ch>. Ad oggi (2011) i costi annuali per un nome a dominio (manca ancora lo spazio, capitolo successivo) di tipo [.ch](http://www.ch) variano da 17.00 a 34.00 franchi svizzeri.

12.2.2 Lo spazio

Oltre ad un nome, necessitiamo di spazio dove inserire i nostri file. Lo spazio viene fornito da diverse ditte (es. Swisscom) a noi basta inserire in un qualsiasi motore di ricerca la dicitura **Web hosting** ed ecco che ci verranno proposti diversi risultati e diversi costi.

Quando si acquista spazio sul web bisogna fare molta attenzione a cosa stiamo acquistando, infatti questi servizi hanno costi diversi non unicamente per quanto spazio (MB o GB) lasciano al cliente, ma anche e soprattutto quali servizi web mettono a disposizione.

Con quanto visto in questa documentazione, e cioè l'HTML, qualsiasi opzione scelta andrà bene (spazio a parte) ma nel caso volessimo un giorno creare il nostro sito internet con un codice ad esempio come php che si appoggia ad una banca dati MySQL, ecco che dovremo scegliere il nostro servizio di web hosting adatto a quanto necessitiamo.

Infine ricordiamoci cosa vogliamo inserire in internet. Se ad esempio ho creato delle pagine per un moto club, mi andrà benissimo scegliere un servizio di web hosting meno caro possibile con i suoi eventuali problemi di stabilità e/o di servizio. Ma nel caso stessi creando un sito internet per la mia ditta, sicuramente preferisco spendere di più ma avere un'ottima stabilità e un buon servizio.

Infine ricordiamoci che possiamo utilizzare un servizio di web hosting situato ad esempio in USA, e nello stesso tempo usufruire di un nome di dominio .ch.

I costi relativi ad un servizio web hosting a pagamento variano da pochi franchi fino ad arrivare a svariati costi mensili (anche superiori a 100.00 franchi svizzeri).

12.2.3 Come fare

Prima di tutto cerchiamo un servizio di web hosting adatto a le nostre esigenze (spazio, servizi, serietà), una volta trovato, registriamoci e a questo punto abbiamo, in molti casi, la possibilità di chiedere direttamente alla ditta fornitrice di registrare il nostro nome di dominio presso la SWITCH. Il funzionamento è molto più semplice di quello che si possa credere.

12.3 REGISTRAZIONE (GRATUITAMENTE)

12.3.1 Il nome

Un nome gratuito viene fornito anche in questo caso da molte ditte, ma a differenza di uno pagamento non avremo la formula www.propri nome.com, bensì ci appoggeremo ad altri siti internet i quali ci daranno dei nomi a dominio del genere: www.ditta.com/propri nome oppure www.propri nome.ditta.com. Ecco qual è la differenza tra un nome a pagamento ed un nome gratuito.

Voglio aprire una piccola parentesi e informarvi che esiste una piccola nazione di nome Tokelau che per pubblicizzarsi a reso gratuita la registrazione della propria estensione .tk. Questo vuol dire che gratuitamente avrete la possibilità di avere il vostro sito internet www.propri nome.tk. Visitate il sito internet <http://www.dot.tk/>.

12.3.2 Lo spazio

Come per il nome (gratuito), lo spazio viene fornito dagli stessi enti (dominio tk a parte). Infatti non funziona come per il metodo a pagamento, dove dobbiamo avere sia un dominio che uno spazio, qui ci registriamo e ci forniscono nome e spazio.

Dove sta il guadagno per chi ci fornisce tale servizio? Semplice, prima di tutto qualsiasi persona visita il vostro sito internet deve per forza scrivere (nell'indirizzo) anche il nome di chi ci ha fornito lo spazio (e il nome), e in molti casi nelle nostre pagine vengono visualizzate della pubblicità (in molti casi, ma non in tutti).

12.3.3 Come fare

Semplicissimamente scriviamo in un qualsiasi motore di ricerca la stringa **web hosting free** e lasciamoci sbalordire dai numerosissimi enti che offrono tale servizio. Dopo la registrazione saremo pronti ad inserire le nostre pagine.

12.4 TRASFERIRE LE NOSTRE PAGINE

Una volta che abbiamo registrato il nostro nome e il nostro spazio ci verranno forniti dei dati di FTP (File Transfert Protocol), questi vengono utilizzati proprio per inserire i nostri file dal computer (o mac) al web.

Per trasferire i dati (i file), necessitiamo di un programma apposito, anche in questo caso cerchiamo in qualsiasi motore di ricerca, scarichiamo ed installiamo. Vi consiglio il programma **FileZilla FTP**, gratuito e funzionale.

Ricordiamoci che tutte le nostre pagine devono essere modificate sempre sulla postazione di lavoro, non è possibile modificarle direttamente da internet (con HTML).

13. I CSS

13.1 INTRODUZIONE

I CSS (Cascading Style Sheets - Fogli di stile a cascata) ci permettono di modificare le nostre pagine HTML con più semplicità. *Cosa vuol dire questa frase?* Mettiamo di avere in tutte le pagine del nostro sito internet un colore di sfondo blu. Se un giorno vogliamo sostituire tale colore dovremmo editare tutte le pagine HTML ed impostare (tramite l'attributo bgcolor del TAG <body>) il nuovo colore, quindi in un sito di 100 pagine, dovremmo editare 100 pagine. Con l'utilizzo dei CSS invece, editeremo solo la pagina contenente codice CSS e tutte le nostre 100 pagine prenderanno gli attributi da questo singolo file.

Quindi **i CSS ci permettono di centralizzare degli attributi** (uguali in tutte le pagine) in una sola, o meglio: **in un solo foglio** CSS. Questi vengono chiamati css esterni, ma vi è la possibilità di inserire anche dei CSS interni (vedi capitolo 13.3).

Una pagina HTML strutturata tramite fogli CSS, viene quindi definita una pagina XHTML. In questo corso vedremo unicamente alcuni parametri dei CSS, quindi la nostra struttura sarà sempre HTML (con aggiunta di alcuni parametri CSS), questo perché una pagina XHTML deve avere molti attributi dei CSS che purtroppo in questo corso non vedremo. L'XHTML è molto più evoluto di quanto possiate pensare leggendo questi pochi fogli, ma almeno avrete una buona base per cominciare, e sicuramente farete buon uso dei CSS.

13.2 CSS Esterni

I CSS esterni vengono così definiti, perché alla nostra pagina HTML verrà collegato (tramite apposito TAG <link />) un foglio di stile CSS. Questo foglio di stile avrà un'estensione .css (quindi una novità confronto alle nostre pagine .html).

Le nostre pagine prenderanno quindi le informazioni (esempio il colore di sfondo) dal foglio esterno CSS.

13.2 Esempio

Mettiamo di avere un foglio esterno css per l'impostazione del colore di sfondo blu delle nostre pagine HTML, ecco qui il codice della nostra pagina HTML (notate che nel TAG <body> non vi è l'attributo bgcolor):

```
<html>
  <head>
    <title>Pagina HTML</title>
    <link rel="stylesheet" type="text/css" href="sfondo.css" />
  </head>
  <body>
    Questa pagina ha un colore di sfondo blu
  </body>
</html>
```

Come si può notare, inseriamo un TAG <link /> tra i TAG <head>, a questo inseriamo gli attributi **rel** e **type** (che per tutti CSS che vedremo devono essere impostati come da esempio). Infine inseriamo un ulteriore attributo, **href** nel quale scriviamo il percorso del nostro file css.

Ora invece vediamo l'interno del nostro file [sfondo.css](#):

```
body { background-color: #0000ff; }
```

Come notiamo il nostro file sfondo.css è formato unicamente da una sola riga, dove specifichiamo all'interno del TAG <body> l'attributo background-color (per l'impostazione del colore). La sintassi dei CSS verrà spiegata nel capitolo 13.4.

13.3 CSS Interni

Tramite i CSS interni abbiamo la possibilità di specificare un parametro unico per questa pagina. Con l'esempio visto precedentemente (capitolo 13.2) abbiamo impostato il colore di sfondo blu tramite il TAG `<link />`. Quindi (se impostato il TAG) tutte le pagine del nostro sito internet avranno un colore di sfondo blu; mettiamo ora invece di voler impostare per un'unica pagina un colore di sfondo rosso, definiamo quindi all'interno di tale pagina un CSS interno.

13.3 Esempio

Questa pagina HTML dovrebbe avere un colore di sfondo blu, ma con un CSS interno possiamo definire un colore di sfondo rosso:

```
<html>
  <head>
    <title>Pagina HTML</title>
    <link rel="stylesheet" type="text/css" href="sfondo.css" />
    <style type="text/css">
      body { background-color: #ff0000; }
    </style>
  </head>
  <body>
    Questa pagina ha un colore di sfondo blu
  </body>
</html>
```

Notiamo che, anche se abbiamo specificato che tale pagine prende gli attributi del foglio sfondo.css, il risultato è una pagina con sfondo rosso (e non blu), questo grazie al CSS interno, dove tra i TAG `<style>` abbiamo inserito del codice CSS.

Ovviamente questo esempio, è inutile, avrei potuto impostare il colore di sfondo nel TAG `<body>` (con il solo utilizzo di HTML). Ma sappiate che in un foglio CSS possiamo impostare molti più attributi (e quindi non solo lo sfondo della pagina, come fatto qui).

13.4 La sintassi (semplice)

La sintassi, o struttura, di come scrivere del codice CSS è molto semplice. In questo esempio (e anche nei futuri) mettiamo di lavorare su di un CSS esterno.

A proposito anche **i CSS possono essere redatti con un semplice editor di testi** come il blocco note di Windows. L'unica differenza è che nel salvataggio, invece di impostare un'estensione di tipo `.html` ne imposteremo una di tipo `.css`.

Il codice dei CSS può anch'esso essere scritto tutto su una riga, ma per una miglior lettura verrà redatto su più righe (come per l'HTML).

Vediamo una semplice foglio CSS che ci permette di redigere alcune delle regole inerenti il TAG `<body>` ed il TAG `<p>` (paragrafo).

```
body { background-color: #0000ff; color: #ff0000; }
p { font-size: 15px; }
```

Quindi nelle 2 righe sopra stanti notiamo che per definire lo stile di un TAG HTML inseriamo il suo nome senza impostare questo tra i segni `<>`. Dopo aver inserito il nome apriamo (e infine chiudiamo) le parentesi graffe `{ }`. All'interno di queste parentesi specifichiamo degli attributi, questi a differenza di HTML, che ha la formula *tipo=valore*, seguono la formula ***tipo:valore***;

Ovviamente all'interno delle parentesi graffe possiamo inserire più parametri, questi devono essere separati da uno spazio (notate che la formula per la specificazione di un attributo comprende già il segno del punto e virgola `;`). Quindi la sintassi completa:

```
ELEMENTO_HTML { TIPO_ATTRIBUTO1:VALORE1; TIPO_ATTRIBUTO2:VALORE2; }
```

13.5 I commenti

Anche per il codice CSS esistono i commenti che ci aiutano con la lettura (ricordate in HTML un commento viene inserito con la formula `<!-- commento -->`). Nei CSS i commenti devono essere inseriti tra i segni `/*` (apertura) e `*/` (chiusura).

Quindi il nostro foglio con commenti:

```
/* Questo è un commento */  
body { background-color:#0000ff; color:#ff0000; } /* Coloriamo lo sfondo e il testo */  
p { font-size:15px; } /* Impostiamo la grandezza del testo tra i TAG <p> */
```

I commenti possono essere inseriti in qualsiasi parte del codice CSS.

13.6 La sintassi (complessa)

Tramite il nostro foglio CSS abbiamo quindi, fino ad ora, impostato dei colori per tutte le pagine HTML e una grandezza per tutto il testo inserito tra i paragrafi (TAG `<p>`).

Mettiamo ora di volere che nella nostra pagina HTML vi sia un particolare paragrafo che a differenza degli altri abbia il testo di una grandezza di 16 pixel (e non 15) e che sia scritto in *corsivo*. In questo caso utilizzando i CSS interni non risolveremmo il problema perché tutti i nostri paragrafi prenderanno la dimensione e lo stile sopra indicato.

Dobbiamo quindi creare una sorta di nuovo paragrafo aggiuntivo a quello standard, vediamo quindi dapprima il nostro foglio CSS parametrizzato con un paragrafo di nome *speciale* (tralasciamo i commenti):

```
body { background-color:#0000ff; color:#ff0000; }  
p { font-size:15px; }  
p.speciale { font-size:16px; font-style:italic; }
```

Come da esempio aggiungiamo un'ulteriore regola nominata, sempre `p` (essendo un paragrafo) e in più aggiungiamo un segno di punto `.` e un nome scelto da noi.

Ora in HTML se inseriamo il testo tra i TAG `<p>` il risultato sarà simile a:

```
[HTML] <p>il mio testo</p> [risultato] il mio testo
```

Se invece aggiungiamo un attributo all'interno del TAG `<p>` il risultato sarà diverso:

```
[HTML] <p class="speciale">il mio testo</p> [risultato] il mio testo
```

13.7 Il raggruppamento

Mettiamo di voler definire lo sfondo della pagina con colore blu, lo sfondo del paragrafo di colore rosso e lo sfondo del paragrafo di nome *trasparente* di colore blu (come lo sfondo di pagina), potrei scrivere il mio codice CSS nel seguente modo:

```
body { background-color:#0000ff; }  
p { background-color:#ff0000; }  
p.trasparente { background-color:#0000ff; }
```

Ma qui possiamo utilizzare il raggruppamento fornito dai CSS, questo per rendere il nostro foglio il più leggero e pulito possibile:

```
body, p.trasparente { background-color:#0000ff; }  
p { background-color:#ff0000; }
```

Inserendo quindi dopo il tag (e prima dell'apertura delle parentesi graffe) il secondo tag separato da un segno di virgola `,` raggruppiamo i 2 in uno solo. Quindi sia `body` che `p.trasparente` prendono l'attributo dello sfondo di colore blu.

13.8 Gli pseudo-elementi

Con i CSS possiamo impostare anche degli pseudo-elementi, cosa sono questi? Semplice, mettiamo di voler dire che al passaggio del mouse su di un link questo venga colorato diversamente, oppure che tutte le lettere iniziali del mio paragrafo vengano scritte in grassetto e con colore blu, ecco qui quale sarebbe il codice del mio file CSS:

```
p:first-letter { font-weight: bold; color: #0000ff; }  
a:hover { color: #d20c0c; }
```

Ecco quindi qui di seguito **alcuni** degli pseudo-elementi disponibili:

:link	Aggiunge uno stile ai collegamenti non ancora visitati
:active	Aggiunge uno stile ai collegamenti attivi
:visited	Aggiunge uno stile ai collegamenti visitati
:hover	Aggiunge uno stile ai collegamenti su passaggio del mouse
:first-letter	Modificare il primo carattere
:first-line	Modifica la prima riga dell'elemento con tale impostazione

13.8 I campi div

Mettiamo di voler colorare di blu unicamente un pezzo di testo all'interno della nostra frase, in questo caso dobbiamo utilizzare un elemento dei CSS chiamato **div**, attenzione perché se non impostato con attributo (*display:inline;*) un div funge da paragrafo.

```
div.colore_blu { color: #0000ff; display: inline; }
```

Quindi in html scriveremo la seguente istruzione:

Codice HTML: Oggi <div class="colore_blu">è una bella</div> giornata.

Risultato: Oggi è una bella giornata.

13.8 Esempio di un file

Ecco qui un esempio di come può essere un file CSS, questo ha delle impostazioni per i TAG <body> e <p> (uguali) e altre per il TAG <a> (ovvero il tag dei link)

```
body, p {  
    background-color: #000000; /* Colore di sfondo */  
    color: #ffffff; /* Colore del testo */  
    font-size: 12px; /* Dimensione carattere */  
    font-family: Tahoma,Verdana; /* Tipo di carattere */  
    margin: 10px 20px 30px 20px; /* Margini esterni, [top=10 right=20 down=30 left=20] */  
    padding: 20px 5px 20px 5px; /* Margini interni, come per gli esterni */  
    background-image: url('sfondo.jpg'); /* Immagine di sfondo */  
    background-repeat: no-repeat; /* Immagine di sfondo non ripetuta */  
    background-position:center top; /* Immagine ... center (orizzontale) e top (verticale) */  
}  
  
a:link { color: #336add; text-decoration: underline; background-color:#B2FF99; }  
a:visited { color: #00FF00; text-decoration: underline; }  
a:hover { color: #d20c0c; text-decoration: none; }
```

13.9 Alcune funzioni

Come detto inizialmente, grazie a questa guida avrete un'idea di cosa possiamo fare con i CSS, ma ricordate che questi hanno moltissime altre funzioni e/o opzioni. Siccome però questo è un corso base, io vi do gli strumenti e alcune informazioni, la specializzazione spetta a voi.

Ecco quindi qui sotto e **alcune funzioni** dei CSS che potranno esservi utili.

13.9.1 Sfondo (background)

background-color: #000000;	/* Colore */
background-image: url('sfondo.jpg');	/* Immagine */
background-repeat: no-repeat;	/* - no-repeat Immagine non ripetuta - repeat Immagine ripetuta - repeat-x Immagine ripetuta orizzontalmente - repeat-y Immagine ripetuta verticalmente */
background-position:center top;	/* Immagine ... center (orizzontale) e top (verticale) */

13.9.2 Carattere (font)

font-family: Tahoma,Verdana;	/* Tipo */
font-size: 12px;	/* Grandezza */
font-style:normal;	/* normal, italic, oblique, inherit */
font-weight :bold;	/* normal, bold, bolder, lighter, inherit */

13.9.3 Testo (text)

color: #000000;	/* Colore */
text-align:center;	/* Allineamento: left, center, right, justify */
line-height: 10px;	/* Distanza tra una riga e la successiva */
letter-spacing: 3px;	/* Distanza tra una lettera e l'altra */
word-spacing: 30px;	/* Distanza tra una parola e l'altra */
text-decoration:none;	/* Decorazione: underline, overline, line-through, blink */
text-trasform:none;	/* Trasforma: capitalize, uppercase, lowercase */

13.9.4 Elenchi (list)

list-style-type:none;	/* none: nessun elemento per l'elenco */ disc: cerchio pieno circle: cerchio vuoto square: quadrato pieno disc: cerchio pieno */
list-style-image:url('punto.png');	/* Imposta un immagine come puntatore */
list-style-position:inside;	/* Posizionamento: inside, outside */

13.9.5 Altro

margin: 10px 20px 30px 20px;	/* Margini esterni, [top=10 right=20 down=30 left=20] */
padding: 20px 5px 20px 5px;	/* Margini interni, come per gli esterni */
width: 400px;	/* Anche in percentuale, utile con le tabelle */
height: 100px;	/* Anche in percentuale, utile con le tabelle */
border-width: 10px;	/* Bordo tabelle, ma anche di <body> o <p> o ... */
border-style:solid;	/* Stile: solid, double, dotted, dashed, groove, ... */
border-color:#000000;	/* Colore del bordo */
display:inline;	/* Permette ai campi div di non andare a capo */

13.10 Conclusione

Con i CSS possiamo personalizzare tutti gli attributi delle nostre pagine!

Moltissime altre funzioni per i CSS possono essere trovate qui: <http://www.w3schools.com/css/>