

Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Muovere i primi passi in XHTML con Style.

Benvenuti su questa guida freeware, cioè completamente gratuita che può essere copiata e distribuita liberamente a patto che non siano alterati in alcun modo i riferimenti all'autore ed al suo sito di riferimento.

Se siete arrivati fin qui e non per caso, significa che siete interessati a fare pagine web usando XHTML (Extensible HyperText Markup Language) ovvero le più recenti tecniche di programmazione web che fanno uso di regole decisamente più rigide perché meno tolleranti del buon "vecchio" HTML (HyperText Markup Language).

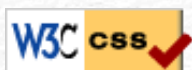
Per quale motivo soltanto adesso su Web-Link si parla di XHTML? Semplicemente perché soltanto adesso si hanno riscontri concreti che questo sia lo standard di regole da seguire: di sigle e di proposte alternative ad HTML, fermatosi alla sua versione 4.01 rilasciata nel lontano 1997, se ne sono viste e sentite molte, ma nessuna ha avuto il successo che prometteva, hanno però tutte quante contribuito notevolmente all'attuale XHTML che altro non sarebbe che una riformulazione di HTML 4.x come applicazione XML.

Lo scopo di questa rubrica è, come sempre, dare la spinta giusta fornendo lo stimolo per iniziare; nessuna presunzione visto l'argomento decisamente più complesso di quello che sembra e che in quanto tale difficilmente riuscirebbe ad essere trattato in maniera esauriente in così poche pagine.

I tempi attuali impongono una certa evoluzione anche da parte degli addetti ai lavori, dalla nascita di HTML ad oggi di cose ne sono cambiate, basti pensare agli stessi dispositivi per accedere alle pagine web (client) che si sono notevolmente evoluti e ai quali se ne sono aggiunti di nuovi; non è più tanto raro accedere a pagine web con palmari o telefoni cellulari ed anche webTV.

A chi si rivolge questa guida? Ovviamente a tutti: chi inizia da zero e chi ha già un sito fatto in HTML e vuole adeguarlo ottimizzandolo nel rispetto delle regole XHTML per renderlo corrispondente a questa definizione. Il passaggio non è complesso, si tratta di introdurre la regola della [struttura a moduli](#) ed essere consapevoli che non si potrà più tralasciare nemmeno quei piccoli particolari pensando che tanto il browser sarebbe poi comunque in grado di visualizzare il codice adoperato, al contrario: la rigidità di XML impedirà di fare come è accaduto fino ad oggi, contribuendo con disattenzioni ed inesattezze al degrado di un codice pulito e semanticamente corretto. Non è infatti una rarità trovare pagine con guazzabugli di codice dovuti a tags ripetuti o inseriti in posti sbagliati, spesso sintatticamente scorretti, e constatare con una certa sorpresa che il browser adoperato è comunque in grado di visualizzare quella pagina in modo decente.

XHTML pretende che tutti i tags ed i loro attributi siano espressi in minuscolo; è obbligatorio inserire sempre il tag di chiusura (quando esiste), ad esempio per il paragrafo `<p>` dove la sua chiusura `</p>` era facoltativa diventerà adesso obbligatoria, mentre per quei tags che non hanno un relativo tag di chiusura, come ad esempio il ritorno a capo `
`, si dovrà fare uso della barra di chiusura `/` all'interno dello stesso così da farlo diventare `
`.



I valori dei vari attributi dovranno essere specificati tra doppi apici o virgolette " " ; si dovrà utilizzare l'attributo id insieme o al posto di name per identificare gli elementi di un documento.

A questo punto il principiante che non conosce nulla di HTML, XLM, XHTML si starà forse chiedendo cosa sia stato detto fino ad ora, ne avrebbe tutte le ragioni, ma vorrei tranquillizzarlo: questa premessa è stata fatta soltanto per chi ha già un sito e volesse convertire le proprie pagine web da HTML a XHTML.

Dalla prossima lezione inizieremo partendo proprio da zero e sarà così possibile apprendere con facilità cosa e come fare per iniziare finalmente a creare la nostra pagina web. Nessuna presunzione, lo ripeto, XHTML è qualcosa di decisamente più complesso di come sarà illustrato in questa guida, ma questa è un "primi passi" e come tale introdurrà l'argomento, fornendo però tutto il necessario e rimandando alla letteratura specializzata per ogni eventuale approfondimento.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Questa opera è pubblicata sotto una Licenza Creative Commons.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Muovere i primi passi in XHTML con Style.

Benvenuti su questa guida freeware, cioè completamente gratuita che può essere copiata e distribuita liberamente a patto che non siano alterati in alcun modo i riferimenti all'autore ed al suo sito di riferimento.

Se siete arrivati fin qui e non per caso, significa che siete interessati a fare pagine web usando XHTML (Extensible HyperText Markup Language) ovvero le più recenti tecniche di programmazione web che fanno uso di regole decisamente più rigide perché meno tolleranti del buon "vecchio" HTML (HyperText Markup Language).

Per quale motivo soltanto adesso su Web-Link si parla di XHTML? Semplicemente perché soltanto adesso si hanno riscontri concreti che questo sia lo standard di regole da seguire: di sigle e di proposte alternative ad HTML, fermatosi alla sua versione 4.01 rilasciata nel lontano 1997, se ne sono viste e sentite molte, ma nessuna ha avuto il successo che prometteva, hanno però tutte quante contribuito notevolmente all'attuale XHTML che altro non sarebbe che una riformulazione di HTML 4.x come applicazione XML.

Lo scopo di questa rubrica è, come sempre, dare la spinta giusta fornendo lo stimolo per iniziare; nessuna presunzione visto l'argomento decisamente più complesso di quello che sembra e che in quanto tale difficilmente riuscirebbe ad essere trattato in maniera esauriente in così poche pagine.

I tempi attuali impongono una certa evoluzione anche da parte degli addetti ai lavori, dalla nascita di HTML ad oggi di cose ne sono cambiate, basti pensare agli stessi dispositivi per accedere alle pagine web (client) che si sono notevolmente evoluti e ai quali se ne sono aggiunti di nuovi; non è più tanto raro accedere a pagine web con palmari o telefoni cellulari ed anche webTV.

A chi si rivolge questa guida? Ovviamente a tutti: chi inizia da zero e chi ha già un sito fatto in HTML e vuole adeguarlo ottimizzandolo nel rispetto delle regole XHTML per renderlo corrispondente a questa definizione. Il passaggio non è complesso, si tratta di introdurre la regola della struttura a moduli ed essere consapevoli che non si potrà più tralasciare nemmeno quei piccoli particolari pensando che tanto il browser sarebbe poi comunque in grado di visualizzare il codice adoperato, al contrario: la rigidità di XML impedirà di fare come è accaduto fino ad oggi, contribuendo con disattenzioni ed inesattezze al degrado di un codice pulito e semanticamente corretto. Non è infatti una rarità trovare pagine con guazzabugli di codice dovuti a tags ripetuti o inseriti in posti sbagliati, spesso sintatticamente scorretti, e constatare con una certa sorpresa che il browser adoperato è comunque in grado di visualizzare quella pagina in modo decente.

XHTML pretende che tutti i tags ed i loro attributi siano espressi in minuscolo; è obbligatorio inserire sempre il tag di chiusura (quando esiste), ad esempio per il paragrafo `<p>` dove la sua chiusura `</p>` era facoltativa diventerà adesso obbligatoria, mentre per quei tags che non hanno un relativo tag di chiusura, come ad esempio il ritorno a capo `
`, si dovrà fare uso della barra di chiusura / all'interno dello stesso così da farlo diventare `
`.

I valori dei vari attributi dovranno essere specificati tra doppi apici o virgolette " " ; si dovrà utilizzare l'attributo id insieme o al posto di name per



identificare gli elementi di un documento.

A questo punto il principiante che non conosce nulla di HTML, XLM, XHTML si starà forse chiedendo cosa sia stato detto fino ad ora, ne avrebbe tutte le ragioni, ma vorrei tranquillizzarlo: questa premessa è stata fatta soltanto per chi ha già un sito e volesse convertire le proprie pagine web da HTML a XHTML.

Dalla prossima lezione inizieremo partendo proprio da zero e sarà così possibile apprendere con facilità cosa e come fare per iniziare finalmente a creare la nostra pagina web. Nessuna presunzione, lo ripeto, XHTML è qualcosa di decisamente più complesso di come sarà illustrato in questa guida, ma questa è un "primi passi" e come tale introdurrà l'argomento, fornendo però tutto il necessario e rimandando alla letteratura specializzata per ogni eventuale approfondimento.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

[Clicca per informazioni !!!](#)

www.web-link.it



[per stampare questa pagina](#)



[Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.](#)



Questa opera è pubblicata sotto una
[Licenza Creative Commons.](#)



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Gli Strumenti

Dopo la [prefazione](#) rivolta in particolar modo a chi conosce già HTML possiamo ora rivolgere particolare attenzione al principiante che non sa praticamente nulla e che vuole iniziare da zero a costruire la sua prima pagina web nel pieno rispetto delle regole degli attuali standard.

Non è assolutamente necessario aver letto [Primi passi HTML](#) mentre diventa obbligatorio leggere prima o poi la [Guida CSS](#) (entrambe raggiungibili dal menù laterale del sito <http://www.Web-Link.it>

Una pagina web è una sorta di ipertesto, ovvero testo apparentemente "normale" generato da un comune editor, uno di quei programmi che permettono di scrivere testo, in questo caso il codice, a volte lo stesso che si adopera per scrivere le normali lettere; la differenza consiste nello scrivere "parole" particolari che diventano vere e proprie istruzioni per il browser di navigazione che le legge e le interpreta di conseguenza, queste parole sono chiamate **<tags>** (elementi), racchiusi fra parentesi angolari **< >**, caratteristica che consente poi la corretta interpretazione da parte del browser.

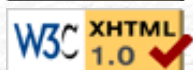
Non c'è nulla di particolarmente difficile, non serve avere alcuna precedente esperienza di programmazione; si deve semplicemente seguire le spiegazioni con una certa attenzione e poi provare, provare e riprovare fino ad ottenere il risultato voluto.

Come ho detto non si tratta di un linguaggio vero e proprio di programmazione per il quale è di solito richiesta una certa preparazione di base, si tratta molto più semplicemente di una combinazione di **tags** (elementi) piuttosto semplici da inserire e direi con non troppe regole da rispettare.

Fare delle buone pagine web non è per niente difficile, i problemi potrebbero nascere quando non ci si accontenta più di semplici pagine o quando queste debbano produrre "effetti speciali", ma questo lo vedremo in seguito; una panoramica più completa su quelli che sono gli elementi (tags) di html è possibile averla consultando la [guida HTML](#) dedicata sempre ai principianti, anch'essa di facile consultazione con possibilità di essere scaricata in formato compresso (zip) per essere stampata o consultata comodamente off-line.

Ci siamo dilungati fin troppo, passiamo all'azione e vediamo cosa serve e quali sono gli strumenti necessari per la creazione della nostra prima pagina web in XHTML. Abbiamo detto un editor, cioè uno di quei programmi che si adoperano per scrivere testi, il wordpad o il blocco notes che di solito si trovano a corredo di windows, tanto per fare un esempio, potrebbero andare bene ma anche qualsiasi altro elaboratore di testi potrebbe fare al caso nostro; unica raccomandazione: ricordarsi sempre di salvare il tutto in formato ASCII, cioè puro testo senza aggiunta di formattazioni particolari. Alcuni di questi programmi infatti, word di microsoft per citarne uno, aggiungono dei caratteri particolari che soltanto loro riescono poi a riconoscere e ad interpretare in modo corretto, questi caratteri non vanno per niente d'accordo con il codice xhtml e di conseguenza **non** vanno adoperati.

Non lasciatevi ingannare dal fatto che word possa esportare in HTML o XML, lasciate perdere, se proprio volete adoperare word e seguire questo corso,



salvate in ascii (txt) e rinominate poi il file così salvato in htm o html.

Personalmente vi consiglio di procurarvi un editor specifico per HTML e magari anche un editor per i fogli di stile CSS visto che se ne trovano di potenti ed anche di gratuiti non vedo perché se ne dovrebbe fare a meno. Dovendo parlare di editor HTML, due sono le categorie in cui si suddividono: visuali e testuali.

Quelli visuali non vanno bene per scrivere codice in XHTML per cui li eliminiamo fin da subito, chi avesse fatto pagine web usando questo tipo di editor difficilmente potrà passare a XHTML o convertire quanto già fatto se non mettendo mano in modo testuale al codice prodotto. Su questa pagina troverete invece gli editors CSS che agevolano molto la creazione e la modifica degli stili offrendo un'ampia panoramica su quello che gli stili stessi sono in grado di fare. Esistono poi editors html come Home Site che hanno a corredo un editor di style come Top Style offrendo così in un solo programma entrambi gli strumenti necessari.

E con questo le premesse sono davvero finite, nella prossima lezione potremo finalmente iniziare a fare qualcosa di concreto.

Se avete domande da fare potete scrivere sul forum di supporto, gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

[Clicca per informazioni !!!](#)

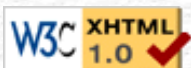
www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Regole, struttura base e DOCTYPE

Dopo aver visto quali sono [gli strumenti](#) necessari possiamo iniziare a fare qualcosa di veramente concreto, creiamo nel nostro PC una nuova cartella dove andremo a salvare tutti i nostri files ed assegnamogli un nome, quello che volete, magari xhtml.

I **TAGS**, o elementi, che compongono XHTML devono essere racchiusi fra parentesi angolari **<queste>**

Si tratta di parole chiave, termini riservati in lingua inglese, da qui la quasi totale inutilità del volersi procurare a tutti i costi un editor html in italiano. L'elemento (tag) smette di produrre il suo effetto nel momento in cui lo stesso viene ripetuto facendolo precedere da una barra laterale / contenuta sempre all'interno delle parentesi angolari. Tutti i tags necessitano di essere chiusi nessuno escluso.

L'inizio di un documento XHTML avrà come elemento iniziale un DTD:

```
<!DOCTYPE PUBLIC "-//W3C//DTD XHTML 1.0 xxxxx//EN"
```

dove al posto di quelle xxxxx dovrà essere usata una delle tre possibili specifiche: Strict, Transitional, Framest. Non è un tag vero e proprio e serve per impartire al browser direttive precise sullo standard al quale dovrà fare riferimento il codice che seguirà, è obbligatorio usarlo per essere certi che il risultato sia il più possibile uguale per qualsiasi tipo di browser adoperato, non trattandosi di un vero elemento (tag) è anche l'unico che non necessita di alcuna chiusura finale ed è scritto in maiuscolo, cosa che non deve essere fatta per nessuno degli altri elementi.

```
<html>
```

e come elemento (tag) finale di chiusura

```
</html>
```

come ho detto sopra dovranno essere scritti in lettere minuscole, tutto ciò che sarà contenuto all'interno di questi due elementi (tags) sarà interpretato dal browser come codice HTML e grazie al DTD specificato precedentemente seguirà le definizioni appropriate di XML, meglio allora se aggiungiamo il riferimento preciso in modo che diventi:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">
```

Un documento XHTML è composto da due sezioni ben distinte: una parte iniziale denominata testa (head)

```
<head>
```

ed una centrale denominata corpo (body)

```
<body>
```

Nella testa (head) vanno messi quei comandi che inviano informazioni al server (lo spazio web che si trova in rete) come il titolo, i comandi Meta Tags e quel codice che deve essere letto con una certa priorità: scripts e fogli di style interni o esterni che decideremo di adoperare. Nel corpo (body) tutto il



resto.

Tutti gli elementi (tags) necessitano sempre del relativo elemento di chiusura: `</head>` e `</body>`

Riassumendo, una struttura XHTML tipo potrebbe essere questa:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

Chi conosce già html penserà a qualche errore da parte mia perché praticamente non è cambiato nulla fra questa struttura XHTML ed una classica di HTML se non l'aggiunta di quel **DOCTYPE** iniziale, ed è infatti questa la prima grande novità; un documento XHTML deve necessariamente avere quella dichiarazione (DTD) e ne esistono di tre tipi, dipende dal grado di ristrettezza e/o della struttura che si intende realizzare. Vediamole qui di seguito:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Specifica l'uso più rigoroso di XHTML, il documento XHTML scritto utilizzando questa doctype non deve fare uso di alcun tag e/o attributo relativo all'aspetto grafico del documento stesso (font, colori, ecc.), l'aspetto grafico è lasciato esclusivamente ai fogli di stile (CSS).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Ammette l'utilizzo dei tags e/o attributi relativi all'aspetto grafico del documento XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Prevede l'utilizzo dei frames (cornici) nei documenti XHTML.

Noi adopereremo per i nostri esempi un DTD Transitional ed un foglio di style esterno che vedremo più avanti, per aiutare e non confondere il principiante ho usato un box con una cornice di colore verde per le parti di codice xhtml mentre quelle relative al codice CSS saranno racchiuse in un box con cornice di colore marrone così facendo si saprà subito a quale delle due parti sto

facendo riferimento.

Col nostro editor creiamo la struttura sopra, usiamo magari il mouse facendo l'operazione di copia ed incolla, andiamo poi all'interno degli elementi (tags) `<head> ... </head>` per inserire il titolo che ha come elementi dichiaranti `title` e relativa chiusura `/title`

```
<title> Titolo della pagina </title>
```

Assegnamo un titolo alla nostra pagina, questo titolo si leggerà nella cornice superiore del browser, è molto importante ai fini di una buona posizione (indicizzazione) nei motori di ricerca e va usato bene per descrivere al meglio il contenuto della pagina, si dovrebbero usare da un minimo di 5 ad un massimo di 15 parole.

Andiamo adesso all'interno dei tags `<body> ... </body>` e scriviamo la nostra prima frase, per farlo useremo l'elemento `p` che significa paragrafo. Apriamo dunque un paragrafo `<p>` scriviamo un qualsiasi testo: "questa è la mia prima pagina web in xhtml". Chiudiamo il paragrafo `</p>`. Se tutto è stato fatto correttamente dovremo avere una struttura identica a questa :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
  </head>
  <body>
    <p>Questa è la mia prima pagina web in xhtml</p>
  </body>
</html>
```

Non importa come è stato disposto il testo nella frase, possiamo lasciare spazi vuoti o ritornare a capo quante volte vogliamo, nessuna importanza; il browser ignorerà il nostro modo di scrivere limitandosi a leggere ed interpretare soltanto gli elementi validi di xhtml e CSS. (più avanti si vedrà come lasciare spazi e come andare a capo)

A questo punto non resta che salvare il tutto nella nostra cartella creata in precedenza appositamente per questo corso. Se il vostro editor non fosse specifico per html fate in modo di salvare in formato ascii, cioè normalissimo testo puro (txt). Assegnategli un nome, quello che preferite, e salvatelo nel vostro PC. Trattandosi della pagina iniziale sarebbe meglio chiamarla fin da subito **index.html** (tutto minuscolo e con la elle finale e non il solo htm) perché è questo il nome esatto che di solito il server in rete va a ricercare come pagina iniziale del sito quando viene introdotto il solo URL.

Se non avete un editor che salva direttamente in html con molta probabilità il vostro file avrà un'estensione di tipo .txt, ovvio direte voi, visto che è stato scritto con un normale editor per testi. Rinominate questo file in modo che abbia come estensione html altrimenti il browser non riuscirà a capire che si tratta di ipertesto html.

A proposito di questa operazione di rinomina file dovrei fare alcune precisazioni: l'operazione di per se è molto semplice basta cliccare sull'icona del file e rinominare ma chi adopera Windows potrebbe non vedere l'estensione finale del file appena salvato, in questo caso rinominarlo servirebbe a poco dal momento che windows aggiungerebbe sì html al nome appena assegnato ma lasciando di fatto il file nel formato originario txt e di conseguenza non funzionerebbe.

Mi rendo conto che non sia molto semplice capire che cosa è stato appena detto specialmente se a leggere è un principiante in senso generale, per chi avesse problemi di questo tipo vi rimando ad un esempio che credo possa

chiarire meglio.

Se siete riusciti a salvare il file in html ha come icona quella del browser e questo significa che il sistema operativo lo riconosce come pagina web e lo ha associato al programma per navigare in internet. Fate doppio click o trascinatelo sul browser aperto e se tutto è stato fatto come si deve si vedrà la sola scritta:

Questa è la mia prima pagina web in xhtml

Difficile ?

No, direi proprio di no, sono certo che pensavate a qualcosa di diverso e decisamente più complesso. Per essere la prima lezione introduttiva credo possa bastare, esercitatevi, soltanto dopo aver compreso questo meccanismo ed aver acquisito quel minimo di familiarità con la creazione ed il salvataggio dei files in formato html potrete passare alle lezioni successive.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Il testo

Abbiamo visto [nella lezione precedente](#), come creare la nostra prima pagina web usando una struttura base e mettendo al suo interno una qualunque scritta generica, allo stesso identico modo avremo potuto scrivere un intero libro se solo avessimo voluto farlo. Ma un testo ha bisogno di essere formattato per essere facilmente leggibile; ed è proprio di questo che ci occuperemo in questa lezione.

Apriamo il nostro editor qualunque esso sia, e richiamiamo la pagina che avevamo creato e salvato, quella della lezione precedente per intenderci. Posizioniamo il cursore nel punto immediatamente prima della chiusura dell'elemento (tag) `</head>` ed inseriamo il richiamo ad un foglio di style esterno che poi andremo a realizzare.

```
<link rel="stylesheet" type="text/css" href="nome_del_foglio_di_style.css" />
```

Dove al posto di `nome_del_foglio_di_style.css` possiamo mettere il nome che preferiamo, io nel mio caso lo chiamerò `foglio.css`

adesso la struttura base sarà come questa:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
    <link rel="stylesheet" type="text/css" href="foglio.css" />
  </head>
  <body>
    <p> Questa è la mia prima pagina web in xhtml</p>
  </body>
</html>
```

Creiamo ora questo foglio di stile e per farlo useremo lo stesso editor adoperato per creare la pagina web, questa volta però anziché salvare in formato html salveremo in formato .css che indica al nostro sistema che si tratta di un documento CSS (cascading style sheet).

Vi ricordo che in questo corso i due codici: html e css saranno distinti fra loro da box con cornici colorate per essere più facilmente riconoscibili, verde per html e marrone per il foglio di stile, così facendo si dovrebbe evitare qualsiasi possibile confusione, vi ricordo anche che la [Guida sui CSS](#) spiega molto meglio e più dettagliatamente come funzionano i css (fogli di stile).

Apriamo l'editor ed inseriamo nel nuovo documento (vuoto) questo codice:

```
p { text-align: center; }
```



Le parentesi graffe si ottengono premendo i tasti AltGr + maiuscole + parentesi quadra [o] che generano { }

Salviamo questo documento col nome foglio.css e per fare questa operazione valgono le stesse regole spiegate nella [lezione precedente](#), se tutto ha funzionato a dovere dovremo avere adesso nella stessa cartella due files: index.html e foglio.css mentre per il file .html si dovrebbe vedere associata l'icona grafica del browser, per l'icona del file .css è probabile che ci sia l'icona col bandierino di windows che indica file sconosciuto, è perfettamente normale perché nel PC non c'è alcun programma in grado di gestire fogli di stile (a meno che non se ne sia installato uno) ma tutto funzionerà perché entrambi i files saranno riconosciuti e gestiti dallo stesso browser.

Vediamo adesso il risultato di questo foglio css facendo doppio clic sul file index.html relativo alla nostra prima pagina web, la scritta non si trova più a sinistra perché nel foglio di style abbiamo definito che il tag `<p>` usato per scrivere il paragrafo ha il testo allineato al centro (`text-align: center`)

Testo allineato al centro (center)

I possibili attributi che possono essere assegnati a `text-align` sono: left (sinistra), right (destra), center (centrato) e justify (giustificato)

Questi i risultati che si ottengono:

Testo allineato a sinistra (left)

Testo allineato a destra (right)

Testo allineato al centro (center)

Testo allineato in modo giustificato (justify) per far vedere l'effetto su entrambi i margini destro e sinistro il testo deve essere di qualche riga per questo che sto scrivendo, per raggiungere una quantità di testo tale che permetta di vedere e capire questo tipo di allineamento.

Se avete capito il meccanismo è facilmente intuibile che in quel foglio di stile è possibile assegnare al tag `<p>` paragrafo usato nell'esempio non soltanto l'allineamento del testo ma molti altri parametri, tutti quelli che i css supportano e che potete vedere sulla pagina parametri della guida css.

Assegnamo una dimensione del font di 10 punti, scegliamo come font per il testo Arial e se questo non fosse disponibile nel PC di chi ci visita allora andrà bene Verdana e se anche questo non ci fosse si adopererà uno della famiglia serif, come colore il rosso. Queste le modifiche da apportare al foglio di style:

```
p { font-size: 10 pt;
```



```
font-family: Arial, Verdana, sans-serif;
color: red;
text-align: center; }
```

questo il risultato:

Testo con font 10 punti di tipo Arial colore rosso allineamento centrale.

Abbiamo fatto la modifica soltanto nel foglio di style senza toccare la pagina web, nella pagina web c'è soltanto il testo ma il colore, le dimensioni, il font e l'allineamento si trovano nel foglio di stile, un foglio che sarà richiamato da tutte le pagine web e se decidessimo di cambiare il colore o il tipo di font a tutto il nostro sito non dovremo andare pagina per pagina, paragrafo per paragrafo ma fare una sola modifica al foglio di stile comune a tutte le pagine per vedere tutto il sito cambiare all'istante.

Usando i fogli di style si fa spesso uso di un elemento contenitore che si chiama `<div>` e che da solo non avrebbe ragione di esistere ma utilizzato con testi ed immagini è in grado di posizionare i vari elementi con precisione assoluta perché è possibile passargli le coordinate orizzontali e verticali dello schermo o lasciare che queste siano calcolate automaticamente dal browser, lo stesso risultato visto sopra lo si sarebbe potuto ottenere usando il solo `div` al posto di `<p>` con una sintassi leggermente diversa, è preferibile `<p>` (paragrafo) perché semanticamente più corretto nel rispetto di modularità xhtml.

Per lo stesso motivo di modularità html mette a disposizione tags per i titoli che vanno da `<h1>` a `<h6>` possono essere ridefiniti grazie ai fogli di stile.

Digitando del testo vi accorgete presto che a differenza degli editor normali, ogni volta che volete tornare a capo non basta la semplice pressione del tasto invio. Nelle pagine web per tornare a capo si fa uso dell'elemento `
` oppure aprendo e chiudendo un nuovo paragrafo `<p></p>` Vi accorgete anche che inserendo più spazi fra una parola e l'altra questi non saranno visualizzati ma ignorati tutti quanti lasciandone uno solo.

Provate voi stessi per rendervene conto. Più avanti nella [lezione dedicata alla cosmesi del testo](#) vedremo anche altre cose che non è possibile fare in html o almeno non così come siamo abituati a fare con un normale editor di testi.

Bene con questa lezione avete imparato a creare una pagina web e ad introdurvi del testo, nella [prossima lezione](#) vedremo come gestire il testo organizzandolo in liste ordinate.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Le Liste

Abbiamo visto [nella lezione precedente](#), come inserire del testo e formattarlo a seconda delle proprie esigenze; un testo però a volte necessita di elenchi o liste più o meno ordinate e xhtml, grazie ai css, offre diverse possibilità per gestire e personalizzare queste liste, vediamo alcune, quelle maggiormente adoperate.

Tutte le liste possono essere nidificate fra loro, all'interno di queste liste ci sono tre elementi base:

istruzione iniziale della lista

istruzione di di inizio e termine dell'elemento della lista

istruzione finale della lista

Prendiamo la nostra struttura html, quella usata nelle lezioni precedenti, ed inseriamo le istruzioni per definire una lista, in questo caso ordinata (ol = Ordered List):

```
<body>
  <ol id="prova_lista" dir="ltr">
    <li>Descrizione prima voce della lista</li>
    <li>Descrizione seconda voce della lista</li>
    <li>Descrizione terza voce della lista</li>
    <li>Descrizione quarta voce della lista</li>
    <li>Descrizione quinta voce della lista</li>
  </ol>
</body>
```

Questo il risultato:

1. Descrizione prima voce della lista
2. Descrizione seconda voce della lista
3. Descrizione terza voce della lista
4. Descrizione quarta voce della lista
5. Descrizione quinta voce della lista

Le nostre frasi hanno adesso un numero progressivo inserito dalle istruzioni lista, notare l'attributo **dir** che serve per indicare la direzione: ltr (left to right) da sinistra a destra. La stessa lista con attributo rtl (right to left) da destra a sinistra:

1. Descrizione prima voce della lista
2. Descrizione seconda voce della lista
3. Descrizione terza voce della lista
4. Descrizione quarta voce della lista
5. Descrizione quinta voce della lista



Vediamo anche come definire una lista non ordinata. (ul= Unordered List)

```
<body>
  <ul id="prova_lista" dir="ltr">
    <li>Descrizione voce lista 1</li>
    <li>Descrizione voce lista 2</li>
    <li>Descrizione voce lista 3</li>
    <li>Descrizione voce lista 4</li>
    <li>Descrizione voce lista 5</li>
  </ul>
</body>
```

Questo il risultato a destra con dir ltr:

- Descrizione prima voce della lista
- Descrizione seconda voce della lista
- Descrizione terza voce della lista
- Descrizione quarta voce della lista
- Descrizione quinta voce della lista

Questo il risultato a sinistra con dir rtl:

- Descrizione prima voce della lista
- Descrizione seconda voce della lista
- Descrizione terza voce della lista
- Descrizione quarta voce della lista
- Descrizione quinta voce della lista

E' possibile usare i css per controllare l'aspetto e la posizione dei contenuti della lista stessa. le proprietà: `list-style-type`, `list-style-image`, `list-style-position` e `list-style` servono proprio a questo, vediamo alcuni esempi:

Adoperiamo lo stesso foglio di stile usato in precedenza. Apriamolo con l'editor ed inseriamo al suo interno questo codice:

```
ol { list-style-type: square; }
```

Abbiamo definito come tipo di lista ordinata `square` che significa quadrato

1. Descrizione prima voce della lista
2. Descrizione seconda voce della lista
3. Descrizione terza voce della lista

Con lo stesso sistema si possono avere diversi simboli usando le seguenti parole chiave da assegnare alla proprietà `list-style-type`

1. Descrizione con **square**
2. Descrizione con **disc**
3. Descrizione con **circle**
4. Descrizione con **lower-roman**
5. Descrizione con **upper-roman**
6. Descrizione con **lower-alpha**
7. Descrizione con **upper-alpha**
8. Descrizione con **none**

Da notare che il numero romano minuscolo riporta IV = 4 perchè è il quarto componente di quella lista, quello successivo infatti essendo il quinto (V) segue un incremento cronologico, stessa cosa vale per le lettere dell'alfabeto, la sesta voce corrisponde alla sesta lettera in ordine alfabetico e cioè la lettera f. upper e lower stanno ad indicare se usare lettere maiuscole o minuscole. none viene usato per nessun marcatore.

Vediamo la proprietà **list-style-image** che permette di usare una immagine grafica.

```
ol { list-style-image: url(file:nome_immagine.gif); }
```

1. Descrizione 1 con immagine grafica
2. Descrizione 2 con immagine grafica
3. Descrizione 3 con immagine grafica
4. Descrizione 4 con immagine grafica
5. Descrizione 5 con immagine grafica

Dove è possibile dopo **url(file:** specificare un url valido che si trova in rete, una cartella del proprio spazio web o direttamente il nome dell'immagine da adoperare se questa si trova nella stessa cartella dove risiede la pagina web che la richiama.

Vediamo la proprietà **list-style-position**: che accetta due parametri: inside o outside

```
ol { list-style-position: inside; }
```

- Descrizione testo nella lista non ordinata con parametro inside
Descrizione testo nella lista non ordinata con parametro inside
Descrizione testo nella lista non ordinata con parametro inside
- Descrizione testo nella lista non ordinata con parametro inside
Descrizione testo nella lista non ordinata con parametro inside
Descrizione testo nella lista non ordinata con parametro inside
- Descrizione testo nella lista non ordinata con parametro inside
Descrizione testo nella lista non ordinata con parametro inside
Descrizione testo nella lista non ordinata con parametro inside

```
ol { list-style-position: outside; }
```

- Descrizione testo nella lista non ordinata con parametro outside

Descrizione testo nella lista non ordinata con parametro outside
Descrizione testo nella lista non ordinata con parametro outside

- Descrizione testo nella lista non ordinata con parametro outside
Descrizione testo nella lista non ordinata con parametro outside
- Descrizione testo nella lista non ordinata con parametro outside
Descrizione testo nella lista non ordinata con parametro outside

Descrizione testo nella lista non ordinata con parametro outside
Descrizione testo nella lista non ordinata con parametro outside

Notare la differenza che risulta evidente: nel caso di `inside` il marcatore di lista si trova all'interno del testo che compone la lista stessa. Adoperando invece `outside` risulta all'esterno.

La proprietà `list-style` permette di attribuire contemporaneamente una o più proprietà direttamente, come in questo esempio:

```
ol { list-style: url(immagine.gif) square outside; }  
ul { list-style: circle inside; }
```

Per questa lezione credo possa bastare... Avete imparato a creare una pagina web e ad introdurre del testo usando all'occorrenza delle liste, nella prossima lezione vedremo come inserire le immagini.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Le Immagini

Abbiamo visto [nella lezione precedente](#), come organizzare il testo in liste ordinate e non, ma una pagina web che si rispetti non può essere priva di immagini ed è proprio di queste che ci occuperemo in questa lezione.

Aprirete il vostro editor e richiamate la pagina che avevate creato, quella della lezione precedente per intenderci. Posizionate il cursore nel punto in cui dovrà essere inserita l'immagine, questo il **tag** o elemento, da adoperare:

```

```

Dove xxxxxxxxxxxx è il nome dell'immagine e yyy la sua estensione grafica. Le immagini consentite dal linguaggio html sono **.gif .jpg .png**

Si adoperano quelle di tipo **gif** quando servono immagini animate o che abbiano delle trasparenze (quelle che lasciano vedere lo sfondo), mentre per tutti gli altri casi è preferibile adoperare quelle di tipo **jpg** essendo questo un formato compresso risulta molto più snello e di conseguenza più veloce da caricare; un aspetto quest'ultimo da non trascurare mai quando si adoperano immagini. Ricordatevi che dimensioni esagerate significa anche peso esagerato; sulla rete un'immagine da 200 Kb spesso induce il visitatore ad abbandonare il sito prima ancora che questa sia stata completamente visualizzata, ricordatevi anche che nonostante le attuali linee a larga bande super veloci (ADSL) ci sono ancora molti utenti che accedono alla rete con modem analogici da 56 Kb, fate dunque in modo da evitare sempre immagini pesanti ed anzi, al contrario, preoccupatevi che queste siano le più "leggere possibili".

Il formato **.png** nato abbastanza recentemente più per problemi legali che non per reali esigenze tecniche, è adoperato raramente in quanto non risulta poi così "ottimale", ma questo è un mio modesto parere per altro molto discutibile.

Detto questo, passiamo ad inserire la nostra prima immagine. Supponiamo che questa si chiami web-link.gif, sarà inserita nel listato in questa maniera:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
    <link rel="stylesheet" type="text/css" href="foglio.css" />
  </head>
  <body>
    <p>Questa è la mia prima pagina web in xhtml</p>
    
  </body>
</html>
```

Notate la sintassi del comando img che è leggermente diversa da quella

adoperata per inserire il testo con <p>, infatti il tag per le immagini non ha il relativo tag di chiusura, non serve mettere , ma come tutti i tags di xhtml deve essere chiuso ed è possibile farlo con la barra prima della parentesi angolare che determina la fine del tag stesso.

Noterete anche che dopo la parola chiave: img, che indica "immagine", c'è un: "src" che significa: source, ovvero sorgente, cioè quale immagine deve essere visualizzata. Questo nome va inserito all'interno dei doppi apici, ed è praticamente il nome dell'immagine stessa. Tutto chiaro?

Se adesso salvate il file sovrascrivendo quello che avevate richiamato e lo andate poi a visualizzare, noterete che l'immagine non c'è; o meglio c'è qualcosa di nuovo ma non si vede. Come mai? Semplice, vi serve un'immagine che si chiami "web-link.gif" e che si trovi nella stessa cartella dove si trova anche il file html, cioè questo file che avete appena modificato e salvato.

A questo punto vi consiglio di lasciare perdere l'immagine web-link.gif e procuratevi una qualsiasi immagine vostra, quella che preferite, a patto che sia in uno dei formati validi gif jpg o png, mettetela nella stessa cartella dove avete il file html e cambiate il nome all'interno di src=" " inserendo il nome esatto della vostra immagine.

Adesso portatevi una riga sotto il tag che è servito per inserire l'immagine e aprite un nuovo paragrafo scrivendo: La mia prima pagina web ha finalmente anche la sua prima immagine.

Se tutto quanto è stato fatto correttamente, la vostra pagina comincerà a prendere una certa consistenza, con la stessa procedura potete inserire tutti i testi e tutte le immagini che vorrete ma senza mai dimenticare che non si deve esagerare. Il peso in Kb di tante piccole immagini viene sommato e potrebbe creare lo stesso problema di una sola immagine da molti kb. Per evitare questo conviene suddividere in più pagine i documenti lunghi o ricchi di immagini.

Vediamo cosa dobbiamo o potremo aggiungere al tag:

```

```

Molto importante ai fini dell'accessibilità è l'attributo alt che significa alternate cioè testo alternativo che descrive l'immagine nel caso in cui questa non fosse visibile.

```

```



Provate a portare il cursore del mouse sull'immagine e lasciatelo fermo per un momento, a seconda del browser adoperato si potrà leggere il testo inserito con l'attributo alt e questo è un comportamento anomalo perchè l'immagine è visibile. In effetti non servirebbe a nulla dal momento che alt dovrebbe essere in alternativa all'immagine mancante ma il browser IE si comporta in questo strano modo.

Se è un testo descrittivo che vogliamo visualizzare passando sopra all'immagine si deve adoperare l'attributo **title** riconosciuto da tutti i browser e serve proprio per dare un titolo o una descrizione testuale all'immagine grafica, si possono adoperare entrambi (alt e title) per i browser come IE avrebbe comunque priorità l'attributo title.

```

```



Se la nostra immagine non si trovasse nella stessa cartella dove c'è il file che la richiama dovremo necessariamente farla precedere dal "path", cioè il percorso che specifica la sua posizione all'interno del nostro PC, questo potrebbe essere anche un URL valido, cioè un indirizzo di un sito Internet che la contiene, in questo caso però l'immagine sarà visibile soltanto nel caso in cui ci si trovi collegati alla rete internet.

Questi alcuni probabili validi percorsi:

```

```

```

```

```

```

Se vengono specificate le dimensioni dell'immagine, il browser sarà molto più rapido nel generare tutta la pagina per cui sarebbe opportuno inserirle sempre. Per farlo si adoperano i due attributi in lingua inglese **width** per la larghezza ed **height** per l'altezza.

```

```

I numeri 165 e 120 sono pixel, l'unità di misura per le dimensioni dell'immagine. E' anche possibile specificare se l'immagine deve avere un bordo perimetrale e di quale spessore; per ottenere questo si adopera l'attributo: **border** dove è ammesso anche valore zero nel caso in cui non si volesse alcun bordo.

Avrete certamente notato che tutti i vari attributi restano all'interno delle parentesi angolari, <queste> potrebbero essere disposti anche in ordine diverso da quello che ho adoperato io a patto che sia rispettata la corretta sintassi e per sintassi si intende: spazio fra un parametro e l'altro, segno di = uguale, le virgolette " (doppi apici) " la chiusura /> del tag img.

In xhtml strict non è ammesso specificare il tag immagine al di fuori di un elemento blocco o di un contenitore on line come: paragrafo, titoli, tabelle. ecc.ecc.

Ci sono molti altri attributi che potrebbero essere inseriti allo stesso modo, vi rimandando alla [guida HTML](#) nel caso si volesse conoscere l'elenco completo.

I fogli di style introducono molti nuovi parametri per la gestione ed il

posizionamento delle immagini e gestiscono alcuni di quelli tradizionali, per esempio è possibile specificare le dimensioni in **em** in modo che varino col variare delle dimensioni del testo impostate dall'utente, o in percentuale **%** in modo che la variazione dipenda dalla risoluzione grafica e dalle dimensioni della finestra del browser lasciando a questo il compito di calcolare le dimensioni mantenendo le giuste proporzioni. Aggiungete il codice sotto al vostro foglio di style esterno:

```
img#ridimensiona { width: 80%; }
```

Un esempio chiarirà meglio la funzione del codice css riportato sopra, su [questa pagina](#) è possibile vederlo in azione.

Il suo richiamo nel codice xhtml dovrà essere fatto in questo modo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
    <link rel="stylesheet" type="text/css" href="foglio.css" />
  </head>
  <body>
    <p>
      
  </body>
</html>
```

Nota: il nome dell'immagine deve essere scritto esattamente come è realmente, rispettando cioè eventuali maiuscole/minuscole/spazi, questo perché a differenza di windows che non farebbe alcuna distinzione, una volta trasferito il lavoro sul server web l'immagine di nome Web-link.gif sarà considerata un'immagine diversa dalla stessa immagine che si chiamasse WEB-link.gif (differenza dovuta alle maiuscole e minuscole).

Esercitatevi per capire questo meccanismo, la guida CSS potrà essere di particolare aiuto per i principianti. Nella [prossima lezione](#) vedremo come inserire le immagini col testo e come posizionarle all'interno dello schermo.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Immagini e Testo

Abbiamo visto [nella lezione precedente](#), come inserire immagini nella nostra pagina web, avevamo la [lezione ancora precedente](#) come inserire il testo, adesso vedremo come si possono combinare testo ed immagini.

La prima difficoltà che si incontra quando si inserisce una immagine è posizionarla in un punto preciso che sia diverso dal margine sinistro, se nella pagina c'è del testo diventa difficile amalgamare al meglio i due elementi (immagini e testo). Per nostra fortuna esistono gli attributi di allineamento ed il box model che vedremo in seguito.

Le immagini sono in linea col testo i Fogli di Style permettono di cambiare questo allineamento tramite l'attributo **vertical-align** che accetta i seguenti parametri: *baseline, top, middle, bottom, sub, super, text-top* e *text-bottom*. Personalmente non trovo che il risultato sia soddisfacente ma essendo un mio parere non va preso in considerazione. Trovo invece decisamente utili **display: block;** che serve per avere il testo sopra e sotto all'immagine o **float: left;** che permette al testo di disporsi intorno all'immagine.

Vediamoli in pratica, aprite il vostro editor e richiamate il foglio di style creato in precedenza. Inseriamo una definizione **display: blok** per il tag `img` che sarà adoperato da tutte le immagini presenti nella nostra pagina.

```
img {
  display: block;
}
```

Questo il risultato:

questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di `display: block`



questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di `display: block`

Mentre con **float: left**





se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left

float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left

E' stato ridefinito il tag `img` ma così facendo tutte le immagini della pagina seguirebbero le stesse direttive, per poter creare definizioni diverse da applicare ad ogni singola immagine si usa l'attributo `id` (identificativo) che richiama il selettore specificato.

```
img#sinistro {
float: left;
margin: 15px 15px 15px 15px;
}

img#destro {
float: right; margin: 15px 15px 15px 15px;
}
```

Così facendo abbiamo adesso due identificativi `id` con nomi: `destro` e `sinistro` applicati entrambi al tag `img` per le immagini.

Nel codice html saranno richiamati inserendo l'attributo `id` corrispondente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
<head>
<title> Titolo della mia pagina Web</title>
<link rel="stylesheet" type="text/css" href="foglio.css" />
</head>
<body>

<p>tutto il testo che serve tutto il testo che serve </p>

<p>tutto il testo che serve tutto il testo che serve </p>


</body>
</html>
```


Notate la disposizione del testo, l'immagine anche se apparentemente rotonda ha una forma rettangolare o quadrata, non è possibile infatti creare immagini rotonde, se inseriamo il bordo si vedrà meglio:



A questo punto cambieremo nel nostro foglio di style l'allineamento impostando il giustificato al tag `p`

```
p { font-size: 10 pt;  
font-family: Arial,Verdana,sans-serif;  
color: #000000;  
text-align: justify; }
```

Ottenendo un risultato decisamente più elegante:

questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left questo testo non serve a nulla se non a far vedere l'effetto con testo e immagini facendo uso di float : left



Nell'immagine è stato impostato un valore leggermente diverso per il margine inferiore, è infatti possibile dichiarare i bordi nel foglio di style usando quattro dichiarazioni distinte, una per ogni margine, si inizia dal margine superiore e si prosegue in senso orario quindi margine laterale destro, margine inferiore e margine laterale sinistro. A causa dell'ombra grafica sotto l'immagine il suo margine inferiore è stato ridotto a 3 pixel anziché 15 come negli altri casi, per dare l'impressione che lo spazio sia

distribuito in parti uguali: **margin: 15px 15px 3px 15px;**

Anche per questa lezione credo possa bastare... Abbiamo imparato ad introdurre testi ed immagini, nella prossima lezione conosceremo i links, la vera forza delle pagine web.

Se avete domande da fare potete scrivere sul forum di supporto, gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

I Links o le Ancore

Dopo aver visto come inserire [testo](#) e [immagini](#), non resta che affrontare l'argomento che dà vita alla grande rete: i links o ancore, cioè quelle scritte interattive che se cliccate col cursore del mouse danno modo di richiamare punti precisi di una pagina o passare da una pagina all'altra e come spesso succede da un intero sito all'altro. Sono proprio queste Ancore o Links ipertestuali che hanno decretato il successo del web.

Aprirete il vostro editor e richiamate la pagina che avevate creato, quella della lezione precedente per capirci. Posizionate il cursore nel punto esatto in cui dovrà essere inserita un'ancora, (link) questo il **tag** o elemento, da adoperare:

```
<a> </a>
```

a come ancora... così da solo, senza altri attributi è praticamente inutilizzabile, ma vediamo combinato con l'attributo: **href** che è quello maggiormente adoperato.

href significa hypertext reference, indica che ciò che si trova fra le ancore di apertura e di chiusura sarà un collegamento ipertestuale, ovvero permetterà di richiamare un'altra pagina, una immagine o un punto preciso della stessa pagina, quello che sarà specificato nell'attributo stesso.

Vediamo un esempio pratico, supponiamo che nella pagina web si voglia avere un link al sito: www.Web-Link.it

Sarà sufficiente specificarlo in questo modo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
    <link rel="stylesheet" type="text/css" href="foglio.css" />
  </head>
  <body>
    <p>Questa è la mia prima pagina web in xhtml</p>
    
    <a href="http://www.web-link.it"> clicca qui </a>
  </body>
</html>
```

Notate la sintassi del comando, la frase *clicca qui* sarà la sola cosa che si vede sulla pagina, si trova fra i due tags ancora, dove il primo **<a>** fa uso di attributi, in questo caso di **href** che indica quale indirizzo URL valido richiamare nel momento in cui sarà fatto clic sulla scritta, segue la sua chiusura **** che delimita la frase divenuta link (collegamento ipertestuale), guai a dimenticarlo, se accadesse tutto il resto della pagina, da quel punto in avanti, diventerebbe un link!



Possiamo sostituire la scritta *clicca qui* con una qualsiasi altra parola o frase, oppure adoperare delle immagini grafiche, il risultato sarebbe comunque lo stesso, tutto ciò che si trovasse fra i due tags ancora, se cliccato, porterebbe all'apertura del link specificato, in questo caso il sito www.Web-Link.it perché è quello l'indirizzo web specificato dopo href.

Se al posto dell'indirizzo web <http://...> avessi inserito il nome di un'altra pagina contenuta in questa stessa cartella (pagina2.html per esempio) avrei dato vita ad un sito interattivo vero e proprio formato da due pagine web.

```
<a href="pagina2.html"> vai alla pagina due </a>
```

Attenzione, se introducessimo una immagine grafica al posto della scritta si darebbe vita ad un meccanismo chiamato nidificazione degli elementi (tags), cioè si possono inserire elementi html all'interno di altri elementi html, questo perché ogni elemento ha i suoi attributi ed i suoi effetti e trovandosi all'interno di un altro elemento eredita eventuali proprietà ed attributi dell'elemento che lo incorpora. Da qui la modularità del linguaggio

```
<a href="pagina2.html"></a>
```

In questo caso l'immagine grafica adoperata come esempio nella pagina precedente ha ereditato le sue proprietà diventando di fatto un link e se cliccata porterà alla pagina 2.



Il suo bordo adesso potrebbe essere decisamente antiestetico ma è una caratteristica dei links, cambia di colore a seconda che il link sia già stato visitato oppure no. E' possibile sopprimerlo inserendo all'immagine l'attributo `border="0"` visto in precedenza.

```
<a href="pagina2.html"></a>
```



Continua con i links...

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Links o Ancore

[Pagina precedente...](#)

Considerate che la pagina richiamata dal link andrà a sostituire la pagina sulla quale si trovava il link richiamante, se non è questo l'effetto desiderato, è possibile aggiungere un attributo che specifica dove aprire il link o meglio su quale finestra del browser di navigazione dovrà essere aperto.

Si tratta di **target** bersaglio, destinazione e fra i vari parametri è possibile inserire la parola riservata **_blank** che significa apri il link in una nuova finestra del browser in modo da mantenere anche la finestra dalla quale è stato fatto clic, per cui ci saranno due finestre del browser aperte: una con la nostra pagina e l'altra con la pagina richiamata dal link.

Questo metodo non piace a molti, forse perché lo ritengono un ostacolo ai fini di una più facile accessibilità, io credo invece che sia indispensabile in certe situazioni ma ancora una volta si tratta di un mio parere molto personale e come tale discutibile.

Se avete un sito strutturato con frames, cioè diversi riquadri all'interno di una finestra principale, è possibile specificare il nome del riquadro nel quale vedere il risultato del link.

Vediamo un esempio:

```
<a href="pagina2.html" target="_blank"></a>
```



Facendo clic sull'immagine grafica si richiamerà la pagina2.html ma aprendo una nuova finestra del browser.

Altri possibili attributi per il tag `<a>` sono:

accesskey="..." Assegna un tasto di attivazione immediata (hotkey) all'elemento

tabindex="..." il valore che definisce in che ordine attivare il presente elemento tra tutti i collegamenti quando si usa il tasto di tabulazione



Volendo richiamare un punto preciso in una pagina che potrebbe anche essere della stessa pagina, si adopera l'attributo **id** (in html era il solo **name**) entrambi servono per creare una specie di etichetta, questa viene collocata nel punto esatto da richiamare.

Il tag **<a** con attributo **href** questa volta adopera uno strano simbolo **#** (si chiama pound o cancelletto), seguito dal nome assegnato all'etichetta creata in precedenza, facendo clic su questo link si raggiungerà il punto esatto in cui è stata posizionata l'etichetta creata con id e/o name, sembra complicato ma un esempio chiarirà molto meglio questo semplice meccanismo:

Creiamo una etichetta nel punto preciso da richiamare usando i due attributi id e name:

```
<a id="qui" name="qui"></a>
```

Non serve inserire nulla fra l'apertura e la chiusura del tag ancora perché non c'è nulla da visualizzare trattandosi di una etichetta che serve solo al meccanismo di spostamento, risulterà essere completamente trasparente alla pagina web il non vedere nulla sarà a tutto vantaggio dell'estetica.

Adesso creiamo il link che richiama il punto preciso in cui si trova l'etichetta creata in precedenza:

```
<a href="#qui">vai a fondo pagina</a>
```

Ho messo l'etichetta **qui** a fondo pagina, proviamo il link appena creato sopra:
[vai a fondo pagina](#)

Se il punto da richiamare si trova su una pagina diversa dalla stessa in cui avviene il richiamo è sufficiente inserire il nome della pagina seguito dal nome dell'etichetta:

```
<a href="nome_pagina.html#qui">vai in un punto preciso di una pagina</a>
```

I più attenti a questo punto si saranno già accorti che i links usati da queste pagine hanno un colore diverso da quello standard e sono privi della sottolineatura, si tratta infatti di una personalizzazione ad opera dei CSS.

Possiamo definire il colore, il font, lo sfondo, le dimensioni e molto altro ancora specificando questi parametri nel nostro foglio di style sul quale andremo ad inserire questo codice:

```
A:link{ text-decoration: none; color: #0000cc; }  
A:visited{ text-decoration: none; color: #009900; }  
A:hover{ Color: #3366ff; text-decoration: none; }
```

Vanno messi in questa sequenza: link, visited, hover e si riferiscono rispettivamente ai links ancora da visitare, a quelli visitati, e quando il mouse si trova sopra ad un link visitato o meno. Nelle'esempio sopra ci siamo limitati a togliere la sottolineatura **text-decoration: none;** e ad assegnare un colore diverso e personalizzato per ognuna delle condizioni possibili **Color: #3366ff;** Non importa la sequenza dei parametri, se prima il colore e poi tex-

decoration, si veda a questo proposito la [guida sui CSS](#) relativa alla voce i links.

Non resta che concludere questa lezione dedicata ai links elencando le altre funzioni che un link può richiamare, oltre ad un indirizzo web preceduto da http:// l'attributo href può infatti servire per richiamare:

```
<a href="ftp://...">  
<a href="mailto:...">  
<a href="nomefile.zip">  
<a href="nomepagina.html">
```

che servono rispettivamente per:

```
<a href="ftp://...">descrizione</a>
```

collegamento ad un sito ftp.

```
<a href="mailto: nome@server.suffisso">descrizione</a>
```

indirizzo di posta elettronica per la spedizione di e-mail all'indirizzo specificato richiamando il proprio programma di posta elettronica.

```
<a href="nomefile.zip">descrizione</a>
```

prelievo, scaricamento (download) di un file in formato compresso zip.

```
<a href="nomepagina.html">descrizione</a>
```

collegamento ad altra pagina html contenuta all'interno della stessa cartella.

Ce ne sarebbero altri ancora quali: [news](#), [gopher](#), [nntp](#), [telnet](#), [wais](#), [newsrsc](#); ma visto il loro raro utilizzo ritengo che in questa fase possano servire soltanto per creare confusione e questo non soltanto ad un principiante.

A questo punto conoscete tutto ciò che serve per creare un sito, si ho detto sito e non più pagina web, infatti con l'inserimento del tag ancora è possibile passare da una pagina all'altra dando vita ad un vero e proprio sito: il vostro.

Sono ancora molte le cose da sapere per fare un buon sito, ma già con questa lezione potete cominciare a fare qualcosa di concreto, [dalla prossima](#) torneremo sulla struttura, sul testo e sulle immagini aggiungendo alcuni particolari e precisazioni che vi serviranno soltanto se nel frattempo avrete cominciato a fare qualcosa di vostro.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

[torna su](#)

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Caratteri speciali e cosmesi del testo

Dopo aver visto come si adoperano i [links](#) torniamo adesso sul testo. Se avete fatto delle prove, e mi auguro proprio di sì, vi sarete sicuramente già accorti che scrivendo in XHTML non sempre è possibile ottenere lo stesso risultato che si ottiene con la normale scrittura di un testo. Certi caratteri infatti hanno regole precise, si tratta semplicemente di conoscerle ed applicarle.

Una delle prime cose che si notano è la difficoltà di poter lasciare più di uno spazio vuoto fra una parola e l'altra. Xhtml infatti ignora tutti gli spazi inseriti con il metodo normale e ne lascia di fatto uno soltanto, provate dunque a scrivere una frase mettendo più spazi vuoti fra le varie parole, provate poi a guardare il risultato sul vostro browser, noterete con sorpresa che tutti gli spazi sono stati ignorati e sostituiti da uno soltanto, questa è una caratteristica di XHTML, come fare allora per avere più di un solo spazio nel caso in cui servisse distanziare maggiormente le parole?

Con appositi codici che servono esclusivamente ad introdurre spazi:

** **; Questo simbolo significa non-breaking space, (da notare che non va inserito fra parentesi angolari non essendo un elemento o tag), se ne possono mettere tanti quanti sono gli spazi desiderati, così facendo il browser eseguirà la richiesta. Anche il **;** punto e virgola finale fa parte del comando

Esempio: lascio tre spazi fra questa e le parole che seguono.

Anche tornare a capo diventa difficile all'interno di uno stesso paragrafo; la semplice pressione del tasto invio (o return a seconda del tipo di tastiera), non è sufficiente per forzare un ritorno a capo, per questa operazione si adopera il tag **
** (break) questo tag forza un ritorno a capo, più **
** consecutivi equivalgono a più ritorni a capo, saltare tante righe quanti sono i br inseriti.

I due metodi sopra sono di gran lunga i più adoperati durante la scrittura di un testo xhtml, da qui la logica considerazione: per ottenere un semplice spazio vuoto c'è bisogno di un'operazione macchinosa come digitare quella sequenza di 5 caratteri? Se adoperate un normale editor di testi sì, se invece vi siete procurati un editor html, probabilmente si tratta di fare click su un pulsantino grafico che introdurrà quella sequenza di caratteri automaticamente, quasi come premere il tasto spazio sulla tastiera.

E' possibile inserire una riga divisoria adoperando il tag **<hr />** (horizontal rule), il risultato sarà questo sotto:



Ci sono anche altri caratteri che conviene introdurre con le corrette regole che XHTML impone, si tratta di caratteri particolari ma che spesso si adoperano nella comune scrittura; per esempio noi Italiani abbiamo le vocali accentate, se scritte come da tastiera queste potrebbero mettere in crisi il browser che le visualizza, è perciò preferibile adoperare il relativo codice per

ottenerle :

à = à
è = è
é = é
ì = ì
ò = ò
ù = ù

Anche le doppie virgolette, le parentesi angolari e tutto quello che potrebbe essere male interpretato conviene scriverlo con il suo codice, ecco qui di seguito l'elenco dei caratteri di uso più comune:

" = "
& = &
< = <
> = >
€ = €

Per concludere, alcuni caratteri speciali che potrebbero risultare utili:

© = ©
® = ®
® = ®
Ø = Ø
£ = £
¼ = ¼
½ = ½
¾ = ¾
¹ = ¹
º = º

Per rendere una pagina più leggibile si fa spesso ricorso ad una specie di cosmesi del testo, si usa infatti sottolinearlo, enfatizzarlo, inclinarlo, ridimensionarlo, ecc. ecc.

Tutte cose affidate al foglio di style nel quale sarà possibile definire parametri, titoli o elementi a blocco da utilizzare per il testo, ma oltre a questo potrebbe essere utile evidenziare una parola o una frase, vediamo quali sono gli elementi (tags) che producono questi effetti.

** = bold o grassetto.**

Tutto quello che sarà scritto all'interno dei tag **** e **** sarà visualizzato in grassetto, facciamo un esempio: dovendo scrivere " www.Web-Link.it tutto per la tua pagina web" e volendo la scritta Web-Link.it in grassetto, si dovrà racchiudere questa fra i due tag **** e ****, in questo modo:

www.web-link.it tutto per la tua pagina web.

Questo il risultato:

www.**web-link.it** tutto per la tua pagina web

Avremo potuto ottenere lo stesso risultato adoperando **** e **** che sono equivalenti ma che in futuro potrebbero essere deprecati.

** = Italico o inclinato.**

Tutto quello che sarà scritto all'interno dei tag **** e **** sarà visualizzato in italico, facciamo ancora lo stesso esempio: dovendo scrivere

" www.Web-Link.it tutto per la tua pagina web"
e volendo la scritta Web-Link.it in italico, si dovrà racchiudere questa fra i due tag `` e ``, in questo modo:

```
www.<em>web-link.it</em> tutto per la tua pagina web.
```

Questo il risultato:

www.*web-link.it* tutto per la tua pagina web

Avremo potuto ottenere lo stesso risultato adoperando `<i>` e `</i>` che sono equivalenti ma che in futuro potrebbero essere deprecati.

Ho riportato questi due perchè a volte risultano più pratici da adoperare nel caso in cui si volesse evidenziare una piccola parte del testo, è anche possibile ridefinire strong e em all'interno del foglio di style assegnando loro magari diversi parametri per esaltare meglio la loro modifica sul testo.

Ci sono una serie di "titoli" sei livelli di intestazioni, da h1 (il più importante) ad h6 (il meno importante) che è possibile adoperare con o senza una ridefinizione di style: `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>`

```
<h1>...</h1>
```

Grassetto, carattere molto grande, una o due righe vuote sopra e sotto.

```
<h2>...</h2>
```

Grassetto, carattere grande, una o due righe vuote sopra e sotto.

```
<h3>...</h3>
```

Corsivo, carattere grande leggermente rientrato sul margine sinistro, una o due righe vuote sopra e sotto.

```
<h4>...</h4>
```

Grassetto, carattere normale, rientro maggiore rispetto a H3, una riga vuota sopra e sotto.

```
<h5>...</h5>
```

Corsivo, carattere normale, rientro normale come H4, una riga vuota sopra.

<h6>...</h6>

Grassetto, rientro come testo normale ma più di H5, una riga vuota sopra.

Personalmente li trovo orrendi esteticamente parlando, ma visto che piacciono dal punto di vista strutturale ai motori di ricerca ai quali danno una certa importanza e visto che tutto quello inserito in un titolo ha sicuramente un maggiore peso e quindi maggiore importanza, il consiglio è di usarli magari inserendo al loro interno semplicemente una immagine grafica o altri elementi di html o dopo averli ridefiniti a proprio gusto e piacere nel foglio di style, come dall'esempio che segue.

```
h1 { font-size:1.50em }  
h2 { font-size:1.25em }  
h3, h4, h5, h6 { font-size:1em }
```

Ottenendo un risultato leggermente più accettabile:

Grassetto, carattere molto grande, una o due righe vuote sopra e sotto.

Grassetto, carattere grande, una o due righe vuote sopra e sotto.

Corsivo, carattere grande leggermente rientrato sul margine sinistro, una o due righe vuote sopra e sotto.

Grassetto, carattere normale, rientro maggiore rispetto a H3, una riga vuota sopra e sotto.

Corsivo, carattere normale, rientro normale come H4, una riga vuota sopra.

Grassetto, rientro come testo normale ma più di H5, una riga vuota sopra.

Con questa lezione avete imparato ad introdurre caratteri speciali e modificare il testo in modo da renderne più gradevole la sua lettura, nella prossima lezione vedremo come usare i colori e le immagini grafiche di sfondo.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

I colori e gli sfondi

Dopo aver visto come si adoperano i [links](#), le [immagini grafiche](#) ed il [testo](#) siamo praticamente in grado di produrre un documento completo in [xhtml](#) ma anche l'occhio vuole la sua parte per questo che vedremo come abbellire il documento con colori e/o immagini di sfondo facendo uso del foglio di [style](#).

Il colore del testo è definito dalla proprietà [color](#) alla quale è possibile assegnare uno qualsiasi dei 16 milioni di colori disponibili. Stessa cosa per il colore di sfondo definito dalla proprietà [background-color](#) che accetta gli stessi valori assegnabili a [color](#) con l'aggiunta del valore [transparent](#).

Se non vengono specificati i colori, siano questi per il testo e/o per lo sfondo, saranno usati quelli scelti dalle impostazioni del sistema operativo dell'utente, a questo proposito è bene specificare sempre entrambi i due parametri per evitare che magari un testo (o uno sfondo) di un colore possa risultare uguale o simile ad un colore di un testo o di uno sfondo impostato dall'utente nel proprio sistema operativo. Specificandoli entrambi si avrà la certezza di una buona leggibilità sempre, a prescindere dalle impostazioni personalizzate. Ritengo che non ci sia cosa peggiore che visitare un sito che costringe ad uno sforzo visivo per leggere testi scritti magari in rosso o in blu su sfondo nero, oppure giallo su sfondo bianco, personalmente chiudo la pagina e non ritorno su quel sito neppure se fosse l'unico a trattare l'argomento di mio interesse.

Vediamo come inserire nel nostro foglio di [style](#) il colore blu per il testo su sfondo bianco:

```
body {  
  color: Blue;  
  background: White;  
}
```

Ad inizio corso si è accennato alla eredità dei vari elementi ([tags](#)) inseriti all'interno di altri elementi, in base a questa regola tutto quello che viene assegnato all'elemento [body](#), che è padre di tutti gli altri elementi inseriti al suo interno, ne consegue che i parametri assegnati saranno applicati anche a tutti gli altri elementi, a meno che non abbiano una loro definizione personalizzata. Per come inserire i parametri e per i vari attributi e loro valori vi rimando alla [Guida CSS](#).

Oltre ad un colore a tinta unita è possibile usare come sfondo una immagine grafica usando la proprietà [background-image](#) che accetta come parametro un url con percorso (assoluto o relativo) seguito dal nome dell'immagine da adoperare che dovrà essere di tipo gif, jpg o png, gli stessi formati grafici ammessi dal web per le immagini.

```
body {  
  background-image: url(immagine.gif);  
  color: Blue;
```



}

Le immagini usate come sfondo si ripetono fino a riempire completamente l'intera pagina di qualsiasi dimensione questa venga visualizzata e scorrono insieme al testo a meno che non si intervenga su apposite proprietà che ne cambiano il comportamento di default, vediamo un esempio, questa l'immagine grafica che useremo come sfondo:



e su [questo link](#) il risultato dell'immagine che riempie l'intera pagina.

Gli sfondi colorati a tinta unita o fatti da immagini grafiche sono possibili anche a singole parti di codice all'interno dello stesso documento, praticamente quello che vedete nei vari box di questo corso oppure un esempio più marcato come questo:

```
div {  
  background-image: url(immagine2.jpg);  
  color: #ff0000;  
}
```

Ho voluto usare un colore rosso su uno sfondo variopinto per evidenziare meglio e maggiormente quanto detto all'inizio di questa lezione: la difficoltà di leggere un testo quando non c'è il giusto contrasto cromatico e la giusta differenza di luminosità con lo sfondo.

Ci sono altre proprietà che è possibile adoperare insieme alla proprietà background-image, si tratta di:

background-attachment: scroll

background-attachment: fixed

Con valori **scroll** o **fixed**

Servono per tenere fermo lo sfondo mentre si scorre lungo la pagina, per default lo sfondo scorre con la pagina, **scroll** se si adopera **fixed** lo sfondo resta fermo.

(su i due links è possibile vedere un esempio pratico)

background-repeat: repeat-y;

background-repeat: repeat-x;

background-repeat: no-repeat

Con valori **repeat-y** o **repeat-x** o **no-repeat**

Servono per avere una ripetizione dell'immagine di sfondo non su tutta la pagina ma soltanto in senso verticale: **repeat-y** o soltanto in senso orizzontale: **repeat-x** o nessuna ripetizione: **no-repeat**

(su i tre links è possibile vedere un esempio pratico)

Con background-repeat: no-repeat è possibile adoperare un'altra proprietà: background-position: xxx yyy

con i seguenti possibili valori:

xxx

background-position: top

background-position: center

background-position: bottom

yyy

background-position: left

background-position: center

background-position: right

o posizioni numeriche con

background-position: 100px, 50px

background-position: 50%,30%

Con valori **top**, **center**, **bottom**, **left**, **right**, **valori numerici**, **valori in percentuale**

che servono per posizionare l'immagine di sfondo in alto, al centro in basso, a sinistra, a destra, ad una precisa distanza dai bordi, ad una distanza dai bordi che varia in percentuale della finestra del browser.

E' possibile adoperare uno solo valore di x o di y o combinarli entrambi come in questi esempi: **bottom left**, **top right** e tutte le altre possibili combinazioni. (su i vari links è possibile vedere un esempio pratico)

Con questa lezione avete imparato ad usare i colori e le immagini grafiche di sfondo, nella prossima lezione vedremo cosa si deve inserire nella parte head e l'importanza dei meta comandi per essere reperiti meglio dai motori di ricerca.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

head & Meta Tags

Dopo aver visto come si adoperano i colori e le immagini grafiche di sfondo, siamo praticamente in grado di produrre un documento completo in xhtml ma dal momento che questo deve essere messo in rete necessita di alcune informazioni che non cambieranno il suo aspetto estetico ma contribuiranno a renderlo più facilmente reperibile dai motori di ricerca.

Si tratta dei meta comandi che a differenza di tutto quello visto fino ad ora non vanno inseriti nel body ma nella parte head del documento.

Questi metacomandi sono diversi e servono per svariati utilizzi, è possibile vederli raccolti su questa pagina: [Meta Tags](#), credo che soltanto alcuni siano veramente importanti ed utili ai fini di essere rintracciati ed indicizzati al meglio dai motori di ricerca.

Cosa sono i motori di ricerca ?

Quando siete in rete e cercate qualche argomento neppure vi rendete conto che lo fate consultando proprio un motore di ricerca, se questo non sapesse che esiste anche la pagina web che state cercando non potrebbe elencarla insieme a tutte le altre che trattano lo stesso argomento da ricercare.

Per questo motivo è opportuno specificare nella pagina iniziale (di solito index.html) almeno quel minimo di informazioni relative al contenuto del nostro sito, lo si fa con questi comandi che hanno una sintassi simile a quella degli altri tags adoperati fino ad ora e cioè fra le parentesi angolari viene inserito il metacomando con la seguente sintassi:

```
<meta name="xxxxxx" content="yyyyyy" />
```

Dove al posto di xxxxxx andrà messo il nome del metacomando ed al posto di yyyyyy il suo contenuto o descrizione.

Questo tag, come `
` `<hr />` `` non ha il relativo tag di chiusura ed allora vale la regola di chiuderlo prima della sua parentesi angolare finale `>`.

Come detto in precedenza a mio parere i metacomandi realmente utili ed importanti sono pochi: titolo (title), descrizione (description), parole chiave (key words), codifica caratteri (charset).

Vediamo la loro sintassi e a cosa servono:

```
<meta http-equiv="title" content="Mio Titolo" />
```

Si adopera per il titolo del documento, al posto di "Mio Titolo" mettete il titolo della vostra pagina o del vostro sito.



```
<meta name="description" content="La descrizione del sito o della pagina" />
```

Si adopera per la descrizione, è questo che vi identifica realmente, siate precisi nel descrivere cosa tratta il vostro sito.

```
<meta name="keywords" content=" parole, chiave, separate, da, virgole" />
```

Si adopera per dare ulteriori informazioni sull'argomento trattato, spesso si adoperano parole diverse che hanno però lo stesso significato cercando di interpretare ed intuire quello che il visitatore digiterà per effettuare la ricerca, faccio un esempio: chi cerca web-link potrebbe cercarlo facendo uso della parola "weblink" ma anche con "web_link" oppure con "web link". Tutte queste varianti possono essere inserite come keywords (parole chiave) e quindi aiutare nella ricerca. Se ne possono inserire fino a 1000 e dovranno semplicemente essere separate da una virgola e da uno spazio vuoto. Ritengo questo meta tag il più importante in assoluto per certi motori di ricerca.

Per chiarire meglio ed ulteriormente questo concetto vi riporto alcune keywords inserite nei mie metacomandi: [XHTML](#), [CSS](#), [XML](#), [html](#), [dhtml](#), [DTD](#)

A differenza dei tags che devono essere scritti con le sole lettere minuscole le parole chiave possono essere inserite adoperando anche maiuscole.

```
<meta http-equiv="Content-type" content="text/html; charset=windows-1252" />
```

Si adopera per la codifica del set di caratteri adoperato, in questo caso windows-1252.

Questi metacomandi dovranno essere inseriti soltanto nella prima parte della struttura e precisamente fra i tags `<head>` e `</head>`, e non nel body come è stato fatto fino ad ora. Rivediamo adesso lo schema base completo di tutto quello fino a questo momento trattato e relativo alla nostra pagina web in xhtml:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
    <meta .... />
    <meta .... />
    <meta .... />
    <meta .... />
    <meta .... />
    <link rel="stylesheet" type="text/css" href="foglio.css" />
  </head>
```

```
<body>
  <p>Questa è la mia prima pagina web in xhtml</p>
  
  <a href="http://www.web-link.it"> clicca qui </a>
</body>
</html>
```

All'interno della sezione head trovano posto anche altri elementi (tags) relativi al codice di programmazione sotto forma di **scripting** , non ritengo il caso di parlarne adesso, non in questa rubrica almeno, ma se visitate il link relativo troverete comunque tutte le informazioni necessarie al loro utilizzo.

Concluderei la sezione head col tag che permette di visualizzare un'icona nella barra degli indirizzi del browser, non c'entra molto con xhtml ma so che prima o poi vi chiederete come fare, questo il tag per ottenerla:

```
<link rel="shortcut icon" href="/favicon.ico" />
```

Ovviamente dovrete avere un file grafico in formato icona che rappresenti il vostro simbolo e dovrà chiamarsi esattamente **favicon.ico** ed essere depositato sul server insieme alla pagina che lo richiama.

Con questa lezione avete imparato ad introdurre i meta comandi utili per essere visti da molti motori di ricerca, nella **prossima lezione** vedremo il box model per gestire blocchi di testo ed immagini, praticamente la struttura portante per impaginare tutto quello che è stato spiegato fino a questo momento.

Se avete domande da fare potete scrivere sul **forum di supporto**, gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul **Forum** gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

il Box Model.

Una delle cose più importanti introdotte con i CSS è il box model che permette di creare una struttura (layout) fatta da rettangoli e/o quadrati, praticamente colonne e righe con o senza bordo, per poter impaginare i nostri dati, siano essi: testi, immagini o qualsiasi altra cosa che potrebbe far parte di una pagina web.

Vediamolo un box, questo il suo codice in css per definirlo:

```
#box {
width: 350px;
height: 150px;
padding: 20px;
border: 5px solid;
margin: 20px;
}
```

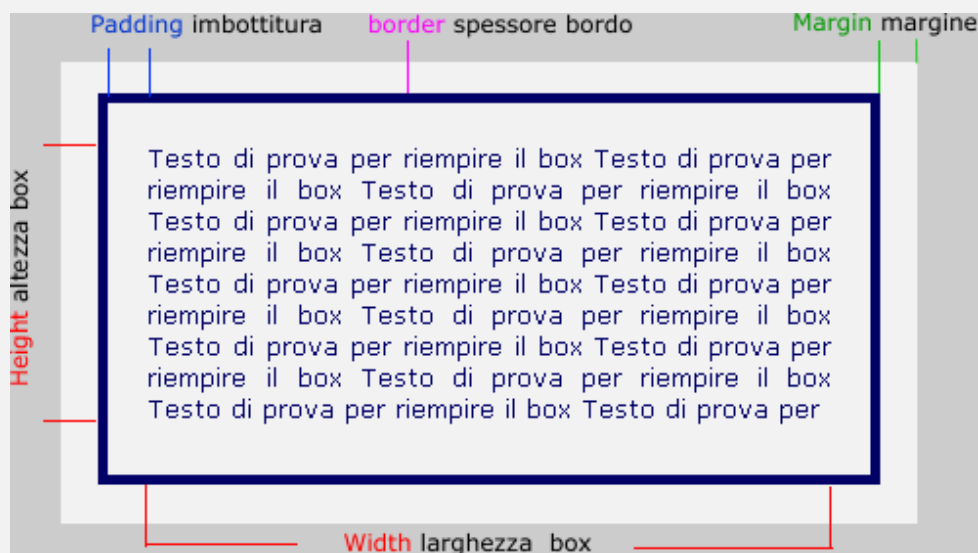
Dove:

width ed **height** sono rispettivamente larghezza ed altezza dei contenuti che possono essere inseriti all'interno del box.

padding lo spazio fra i contenuti ed il bordo del box.

border lo spessore del bordo.

margin la distanza del box dagli altri elementi della pagina.



Questo il codice html per richiamarlo all'interno della pagina:

```
<div id="box">Testo di prova per riempire il box... </div>
```

Questo il risultato a video:



di prova per riempire il box Testo di prova per
riempire il box Testo di prova per riempire il box Testo
di prova per riempire il box Testo di prova per
riempire il box Testo di prova per riempire il box Testo
di prova per riempire il box Testo di prova per

Da notare che la larghezza e l'altezza totale di un box sono dati dalla somma dei contenuti più quello del padding più quello dello spessore del bordo. Nel nostro caso lo spazio occupato realmente sarà: $350 + 20 + 20 + 5 + 5 = 400$ pixel per la larghezza e $150 + 20 + 20 + 5 + 5 = 200$ pixel in altezza. Praticamente le misure specificate (350 x 150) si riferiscono ai contenuti.

Questo è vero per tutti i browser tranne che per IE, il famoso browser della Microsoft, il più adoperato ma anche quello che meno di altri rispetta gli standards o se preferite le direttive del W3C. Li chiameremo bugs e di questi ce ne sono diversi che si sommano a molte proprietà dei CSS che non sono supportate. Pazienza, noi navigatori confidiamo nelle versioni future e nel frattempo chiediamoci come mai noi ci diamo da fare per adeguarci ai nuovi metodi quando poi certe case produttrici di browser possono fare come gli pare. Ma questo è un discorso che qui non c'entra molto.

Se si tratta di un solo box probabilmente nessuno si accorgerebbe di nulla ma dovendo creare un layout formato da più box ecco che le differenze si noterebbero, qualche pixel in più o qualche pixel in meno sarebbe sufficiente per scombinare tutta la nostra impaginazione, per fortuna si può rimediare, vediamo come.

Internet Explorer nelle versioni 5 e 5.5 ma anche per la versione 6 (se opera in *quirks mode*) considera la larghezza e l'altezza dei contenuti sottraendo dalle misure specificate quelle dello spessore del bordo e dello spazio padding. Nel nostro caso il box esterno sarà realmente 350 x 150 Pixel ma il box dei contenuti si rimpicciolisce diventando: $350 - 20 - 20 - 5 - 5 = 300$ pixel per la larghezza e $150 - 20 - 20 - 5 - 5 = 100$ pixel per l'altezza.

In questo esempio la differenza sarebbe notevole se visto con un browser come Firefox ed un IE V.5 e dal momento che IE sottrae bordo e padding dalla larghezza e dall'altezza totale, si potrebbe risolvere creando due box uno interno all'altro dove nel box esterno non si specificano né bordi né padding che metteremo nel box interno.

```
#box-esterno {  
width: 400px;  
height: 200px;  
}  
  
#box-interno {  
padding: 20px;  
border: 5px solid;  
margin: 20px;  
}
```

Questo il codice html per richiamarlo all'interno della pagina:

```
<div id="box-esterno">  
  <div id="box-interno">  
    Testo di prova per riempire il box...  
  </div>  
</div>
```

Questo il risultato a video:

Testo di prova per riempire il box Testo di prova per
riempire il box Testo di prova per riempire il box Testo
di prova per riempire il box Testo di prova per
riempire il box Testo di prova per riempire il box Testo

di prova per riempire il box Testo di prova per
riempire il box Testo di prova per riempire il box Testo
di prova per riempire il box Testo di prova per
riempire il box Testo di prova per riempire il box Testo
di prova per riempire il box Testo di prova per

Il risultato a video è lo stesso per tutti i browser ed il bug di IE non assegnerà al box dei contenuti misure differenti perchè il box interno non ha misure alle quali sottrarre bordi e padding e nel box esterno non ci sono bordi e padding da sottrarre.

PS: se vedete i due box diversi frà di loro il vostro browser è uno di quelli affetti da bug del box model.

Nella prossima lezione dedicata sempre al box model vedremo come posizionarli e quali siano i parametri principali.

Se avete domande da fare potete scrivere sul forum di supporto, gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

Home Page Web-Link
Prefazione
Gli Strumenti
La Struttura base
Il Testo
Le Liste
Le Immagini
Immagini e Testo
I links
Links e CSS
Caratteri speciali
Colori e Sfondi
Head e MetaTags
Box Model
Posizionare il Box
Box liquido
Il Layout
Le Tabelle - Struttura
Le Tabelle - Attributi
I Moduli o Forms
I Frames o Cornici
Iframe
Multimedia Audio
Multimedia Applet
Multimedia Script
On line con FTP
Alcuni Esempi
Conclusioni
Forum di supporto
Download .zip

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Posizionare il Box Model.

Abbiamo visto nella [lezione precedente](#) che un box è praticamente un contenitore di codice html che inserito in un punto qualsiasi della pagina ne segue il suo flusso naturale generando un ritorno a capo come tutti gli elementi di blocco, ne consegue che più box sarebbero disposti verticalmente uno di seguito all'altro. A causa di questa caratteristica diventa impossibile creare dei layout elaborati usando più box senza fare uso di una proprietà dei fogli di style dal nome **position** che permette di posizionare il box in modo diverso da quello naturale.

Gli attributi ammessi per la proprietà position sono: **static**, **relative**, **absolute** e **fixed**.

Vediamoli per scoprire e capire a cosa servono.

static è il posizionamento predefinito, quello naturale che segue il flusso nella disposizione della pagina. Il suo posizionamento è nel punto preciso in cui viene creato o richiamato.

blocco 1 inserito nel normale flusso
blocco 2 inserito subito dopo l'istruzione per il blocco 1
blocco 3 inserito subito dopo l'istruzione per il blocco 2

Notate come i blocchi inseriti nella pagina seguendo il flusso naturale del codice disponendosi verticalmente pur non avendo inserito alcun ritorno a capo e questo perchè il tag <div> come tutti gli elementi di blocco provoca automaticamente un ritorno a capo.

Non avendo impostato alcun **margin** i bordi risultano incollati fra loro. L'esempio sotto è lo stesso box con margin impostato a 5 pixel. Ricordate quanto detto nella lezione precedente riguardo al box model? si fa riferimento al box interno per cui i 5 pixel che distanziano i vari box sono 3 pixel più 2 pixel esterni che fanno da bordo.

blocco 1 con margin inserito nel normale flusso
blocco 2 con margin inserito subito dopo l'istruzione per il blocco 1
blocco 3 con margin inserito subito dopo l'istruzione per il blocco 2

questo il codice per definirlo nel foglio di style CSS.

```
#box1 {
width: 400px;
height: 20px;
border: solid 1px;
margin: 5px;
color: black;
background-color: #ffff99;
}
```

Il codice si riferisce al box1 ma la definizione per gli altri due box è praticamente la stessa, varia soltanto per il nome: #box2 e #box3 ed il relativo colore di sfondo: background-color.

L'attributo margin accetta anche come valore la parola **auto** che posizionerebbe il blocco al centro del flusso dal momento che assegnerebbe automaticamente lo stesso margine ad entrambi i lati.

blocco 1 con margin auto inserito nel normale flusso



Da notare che per avere un bordo nel foglio di style non basta specificare il solo spessore (1px), si deve specificare anche un possibile attributo che identifica il tipo, in questo caso **solid** o il bordo non sarà visibile.

Per richiamare i vari box all'interno del codice html:

```
<div id="box1">blocco 1 con margin inserito... </div>  
<div id="box2">blocco 2 con margin inserito... </div>  
<div id="box3">blocco 3 con margin inserito... </div>
```

Quando il contenuto del box supera le sue dimensioni, per esempio inserendo un'immagine più larga, si hanno comportamenti differenti a seconda del tipo di browser, tanto per cambiare ognuno di questi interpreta a modo suo. Alcuni allargano il box per adattarlo alla dimensione del suo contenuto, altri fanno uscire l'immagine dal box.

E' possibile regolare questo comportamento al fine di unificarlo su tutti i browser grazie alla proprietà **overflow** che accetta i seguenti possibili attributi: **visible** predefinito, **hidden** nascosto, **auto** appaiono le barre di scorrimento soltanto se necessario, **scroll** appaiono le barre di scorrimento sempre anche se non necessitano.



XHTML & CSS di Web-Link.it

overflow visible: predefinito, interpretato in modo differente dai vari browser.



XHTML & CSS di Web-Link.it

overflow hidden: nascondi le barre e taglia l'immagine alle dimensioni del blocco.



XHTML & CSS di Web-Link.it

overflow auto: taglia l'immagine ed inserisci soltanto le barre di scorrimento necessarie.



XHTML & CSS di Web-Link.it

overflow scroll: taglia l'immagine ed inserisci entrambe le barre di scorrimento anche se non necessarie.

relative è il posizionamento modificato rispetto alla posizione predefinita, praticamente un offset che consente di spostare il box rispetto alla sua naturale origine senza che questo spostamento incida o alteri il posizionamento degli altri elementi (oggetti) facenti parte della stessa pagina.

blocco 1 con margin inserito nel normale flusso

blocco 2 position relative 12 px top e 20 px left

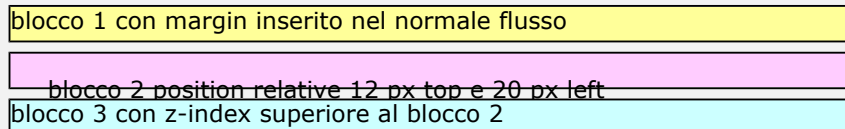
blocco 3 con margin inserito subito dopo l'istruzione per il blocco 2

Notate come il blocco 2 sia spostato di 12 pixel a partire dall'alto e di 20 pixel a partire dalla sinistra rispetto alla sua posizione naturale senza che gli altri due box subiscano alcuna variazione di posizione, questo perché *position relative* agisce solo sul box che ne fa uso senza alterare il normale flusso. E' possibile specificare la posizione *relative* usando gli attributi **top**, **left**, **bottom** e **right** rispettivamente per posizionarlo a partire dall'alto, da sinistra, dal basso e da destra. Sono ammessi anche numeri negativi che posizionerebbero il blocco in direzione contraria da quella dei numeri positivi.

Questo il codice nel css per definirlo.

```
#box2 {  
  position: relative;  
  top: 12px;  
  left: 20px;  
  width: 450px;  
  height: 20px;  
  border: solid 1px;  
  margin: 5px;  
  color: black;  
  background-color: #ffccff;  
}
```

A questo punto il blocco 2 risulta sovrapposto al blocco 3 oscurandolo in parte, è possibile stabilire l'ordine di sovrapposizione con la proprietà **z-index** che accetta solo valori numerici positivi, il numero più alto è quello in primo piano e copre il numero più basso.



Adesso il blocco 3 risulta sovrapposto al blocco 2 perchè ha un valore **z-index** di **1** che è superiore al valore dello z-index (di default impostato a 0) del blocco 2.

questo il codice nel css per definirlo.

```
#box3 {  
  position: relative;  
  z-index: 1;  
  width: 450px;  
  height: 20px;  
  border: solid 1px;  
  margin: 5px;  
  color: black;  
  background-color: #ccffff;  
}
```

absolute è il posizionamento assoluto cioè è possibile posizionare il blocco in un qualsiasi punto della pagina, a differenza del posizionamento relativo che abbiamo visto non alterava gli altri oggetti pur condizionandoli con la sua presenza, *absolute* farà in modo che il blocco così definito risulti come se non facesse parte della pagina, praticamente non inciderà in alcun modo con gli altri oggetti, una specie di sovrapposizione alla pagina stessa.

E' possibile specificare la posizione *absolute* usando gli stessi attributi visti per *relative* **top**, **left**, **bottom** e **right** rispettivamente per posizionarlo a partire dall'alto, da sinistra, dal basso e da destra ma questa volta il punto di riferimento è al primo blocco progenitore non statico o se non esiste all'elemento `<html>` praticamente i margini della finestra del browser.

Se non viene specificato alcun valore di posizionamento il blocco assumerà valore assoluto riferendosi al punto in cui viene inserito dal normale flusso del codice Su [questa pagina](#) è possibile vedere un esempio.

fixed Non supportato dai browser IE è simile al posizionamento assoluto ma il riferimento è sempre e solo alla finestra del browser e quando la pagina scorre i box così definiti restano fissi nella loro posizione specificata.

Nella [prossima lezione](#) dedicata sempre al box model vedremo come centrarlo automaticamente qualunque sia la dimensione della finestra del browser, praticamente un box

liquido.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

il Box Liquido.

Abbiamo visto nelle [lezioni precedenti](#) come creare dei box e come posizzionarli correttamente, esiste però una tecnica che prende il nome di *box liquido* che consentirà al box di adattarsi alla pagina ridimensionandosi automaticamente, proprio come fosse un liquido versato in un recipiente, da qui il nome box liquido.

Dovendo centrare un box abbiamo visto che è sufficiente impostare il valore **auto** alla proprietà **margin**

```
#box-centrato{
margin: auto;
}
```

```
<div id="box-centrato">Testo di prova per riempire il box... </div>
```

Su [questo link](#) il risultato a video

E' possibile eliminare i bordi perimetrali impostando nel body un margine ed un padding a 0 pixel

```
body {
padding: 0;
margin: 0;
}

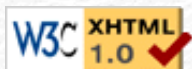
#box-centrato{
margin: auto;
}
```

```
<div id="box-centrato">Testo di prova per riempire il box... </div>
```

Su [questo link](#) il risultato a video.

Oppure decidere un bordo fisso o in percentuale agendo sul padding nel body per un box liquido e flottante.

```
body {
padding: 5% 5%;
margin: 0;
}
```



```
#box-centrato{
margin: 0;
width: 100%;
}
```

```
<div id="box-centrato">Testo di prova per riempire il box... </div>
```

Su [questo link](#) il risultato a video.

Purtroppo però come abbiamo avuto modo di constatare certi valori sono male interpretati o per nulla supportati da parte di alcuni browser. Internet Explorer, quello maggiormente adoperato è anche quello che meno di altri rispetta le regole. Per evitare o arginare questa errata interpretazione è possibile adoperare certi "trucchetti", che rendono la visualizzazione quanto più possibile uguale per tutti i tipi di browser.

Per esempio il browser IE V.5.5 centra il box assegnando il valore center alla proprietà text-align riservata al testo, è possibile allora impostare questa proprietà nel body per poi ripristinare il normale allineamento del testo nel box con text-align: left come nell'esempio sotto.

```
body {
padding: 0;
margin: 0;
text-align: center; /* per il browser IE 5.5 */
}

#box-centrato{
margin: auto;
text-align: left; /* ripristinato allineamento del testo */
}
```

```
<div id="box-centrato">Testo di prova per riempire il box... </div>
```

Su [questo link](#) il risultato a video.

Si conclude con la centratura verticale di un box anche se non è consigliabile per via di un tipo di browser che taglia la parte superiore del box che avesse dimensioni maggiori della finestra del browser che lo visualizza.

```
#box-centrato{
position: absolute;
height: 200px;
top: 50%;
margin-top: -100px;
}
```

```
<div id="box-centrato">Testo di prova per riempire il box... </div>
```

Si noti la tecnica utilizzata nel css: il box ha una dimensione in altezza (height) di 200 pixel, viene posizionato il suo margine superiore al centro della finestra del browser (top 50%) e poi viene impostato il suo margine superiore con un numero negativo che ne provoca lo spostamento verso l'alto e per l'esatta centratura deve essere della metà esatta dei pixel usati per la dimensione del box (200), in questo caso -100px ed il box risulterà perfettamente centrato in senso verticale come è possibile vedere su [questo link](#)

Nella [prossima lezione](#) vedremo in pratica il box model fisso e liquido combinato per creare layout che servono ad impaginare tutti i nostri dati.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

I Layout fissi e liquidi.

Abbiamo visto nelle [lezioni precedenti](#) come creare dei box e come posizionarli nello schermo, adesso facendo uso di queste tecniche possiamo assemblarli creando di fatto un layout, cioè una struttura per contenere i nostri dati.

Si tratta in pratica di usare box fissi e/o box flottanti per impaginare il tutto, dai testi alle immagini dai moduli alle liste o qualsiasi altro oggetto che possa far parte di una pagina web.

Ne vediamo uno a scopo didattico: due colonne; una per il menù a sinistra e l'altra per la pagina principale, una testata con titolo ed un margine inferiore. Lo possiamo vedere a [questo link](#).

Questo il codice nel foglio di style CSS:

```
body {
text-align: center; /* IE5.x */
}

#telaio {
width:100%;
margin: 0px;
border:1px solid gray;
line-height:150%;
color: black;
background-color: #ffffcc;
text-align: left; /* IE5.x */
}

#testata {
font-size: 130%;
}

#fondo {
font-size: 85%;
text-align: center;
}

#testata, #fondo {
padding:0.5em;
color:white;
background-color: #cc9900;
clear:left;
}

#sinistro {
float:left;
width:160px;
margin: 0px;
padding: 5px;
}

#centrale {
```




```
margin-left: 170px;  
border-left:1px solid gray;  
padding:10px;  
}
```

Questo il codice HTML per ottenerlo:

```
<div id="telaio">  
<div id="testata"> ...Testata... </div>  
  
<div id="sinistro">  
... Sinistro...  
</div>  
  
<div id="centrale">  
... Centrale ...  
</div>  
  
<div id="fondo"> ...Fondo... </div>  
</div>
```

Su [questo link](#) il risultato a video

Sono state introdotte molte delle cose trattate fino a questo momento ed anche i parametri sono stati espressi volutamente con unità di misura differenti al solo scopo di rendere meglio l'idea di cosa sia possibile adoperare con l'uso dei fogli di style.

Nella [prossima lezione](#) vedremo le tabelle che non servono soltanto per impaginare dati ma sono a volte adoperate come fossero layout.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).

Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Le Tabelle: la loro struttura.

Abbiamo visto nella [lezione precedente](#) come creare un layout usando dei box posizionati nella pagina. Prima di questa possibilità erano le tabelle il solo strumento ad essere usato in html per impaginare gli oggetti, praticamente venivano private del loro bordo e sfruttando la sola versatile struttura venivano adoperate per impaginare. Così facendo si era quasi perduto lo scopo originale per il quale erano state inventate che era quello di presentare dati ben ordinati e tabulati, questo scopo è stato riscoperto e ritrovato con Xhtml.

Vediamo come si definiscono: `<table>` e relativo elemento (tag) di chiusura `</table>` questo elemento di definizione da solo servirebbe a poco, una tabella è infatti formata da celle disposte su righe e colonne che per essere create necessitano di altri elementi, almeno due: `<tr>` e `<td>` che si trovano annidati all'interno di `<table>` e che servono rispettivamente per suddividere la tabella in righe e/o in colonne, entrambi hanno e richiedono il relativo elemento di chiusura `</tr>` e `</td>`.

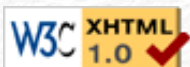
Intere tabelle possono essere contenute a loro volta all'interno di celle di altre tabelle, da qui la definizione di "tabelle nidificate" una specie di scatole cinesi: una dentro l'altra per dare vita a tabelle piuttosto complesse.

Iniziamo con la struttura più semplice: una sola cella. Negli esempi faremo uso dell'attributo opzionale `border` che ne evidenzia il contorno perimetrale, lo useremo anche per le celle alle quali assegneremo un colore di sfondo che ci consentirà di vedere e capire meglio quanto andremo ad illustrare. Potremo assegnare l'attributo `border` direttamente all'elemento `table` ma trattandosi di html con style lo assegneremo invece tramite un foglio di style sul quale imposteremo anche il tipo di bordo, il suo spessore ed il colore. Questo il codice CSS:

```
table {
border: solid 2px #000000;
text-align: center;
}
td {
border: solid 1px #000000;
color: Black;
background: #F1EBAD;
}
```

Questo il codice html:

```
<table>
<tr>
  <td> contenuto della tabella </td>
</tr>
</table>
```



Questo il risultato a video:

contenuto della tabella

Da notare che per `<table>` il bordo è definito da 2 pixel mentre per `<td>` da un solo pixel, questa scelta dipende soltanto dal gusto personale l'importante è sapere che i CSS permettono anche questo. Si noti inoltre che ogni elemento che riguarda la tabella necessita del relativo elemento (tag) di chiusura dove `<tr>` decreta l'inizio di una nuova riga, mentre `<td>` è la cella vera e propria, analogamente `</tr>` decreta la fine di una nuova riga, mentre `</td>` la fine della cella. Dovendo creare adesso una tabella con due celle affiancate, praticamente due colonne, sarà sufficiente affiancare due elementi `<td>` all'interno di un `<tr>`

```
<table>
<tr>
  <td> cella sinistra </td><td> cella destra </td>
</tr>
</table>
```

Questo il risultato :

cella sinistra	cella destra
-------------------	-----------------

Con la stessa logica è possibile aggiungere tutte le righe e le celle che ci servono, quindi:

```
<table>
<tr>
  <td> cella alto sinistra </td><td> cella alto destra </td>
</tr>
<tr>
  <td> cella basso sinistra </td><td> cella basso destra </td>
</tr>
</table>
```

Questo il risultato :

cella alto sinistra	cella alto destra
cella basso sinistra	cella basso destra

Oltre a `<tr>` e `<td>` che sono gli elementi base di una tabella esiste anche `<th>` che definisce il contenuto della intestazione di una tabella; per essere adoperato necessita dell'elemento `<thead>` (in cima) o `<tfoot>` (in fondo), praticamente la tabella viene suddivisa in due parti principali, proprio come succede con la pagina web dove c'è un head (testa) ed un body (corpo) anche in una tabella è possibile avere un `<thead>` ed un `<tbody>` se

dovesse servire avere l'intestazione a fine tabella invece che al suo inizio sarà sufficiente adoperare `<tfoot>` (piede) al posto di `<tbody>` (testa). Potrà sembrare complicato ma vi assicuro che è di una semplicità unica, un paio di esempi chiariranno tutto e meglio:

```
<table>
  <thead>
    <tr>
      <th>titolo head sinistra</th><th>titolo head destra</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> cella alto sinistra </td><td> cella alto destra </td>
    </tr>
    <tr>
      <td> cella basso sinistra </td><td> cella basso destra </
td>
    </tr>
  </tbody>
</table>
```

Ricordatevi che facendo uso di `<thead>` o di `<tfoot>` sarà necessario usare anche `<tbody>`.
Questo il risultato :

titolo head sinistra	titolo head destra
cella body alto sinistra	cella body alto destra
cella body basso sinistra	cella body basso destra

La scritta in grassetto è una caratteristica dell'elemento `<th>` mentre inserendo `<tfoot>` al posto di `<thead>`:

```
<table>
  <tfoot>
    <tr>
      <th>titolo foot sinistra</th><th>titolo foot destra</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td> cella alto sinistra </td><td> cella alto destra </td>
    </tr>
    <tr>
      <td> cella basso sinistra </td><td> cella basso destra </
td>
    </tr>
  </tbody>
</table>
```

Si avrà questo il risultato :

titolo foot sinistra	titolo foot destra
-----------------------------	---------------------------

cella body alto sinistra	cella body alto destra
cella body basso sinistra	cella body basso destra

Per avere una didascalia si adopera l'elemento `<caption>` con relativo elemento di chiusura `</caption>` che deve essere inserito immediatamente sotto l'elemento `<table>`. Serve per descrivere la natura della tabella, descrizione che risulterà essere esterna alla tabella stessa pur facendone parte integrante. E' possibile inserire anche `summary` come attributo dell'elemento `table` e sarà letto soltanto da quei dispositivi non visuali quali per esempio sintesi vocale e Braille.

```
<table summary"descrizione tabella">
  <caption>Descrizione tabella </caption>
  <thead>
    <tr><th>titolo head sinistra</th><th>titolo head destra</th></tr>
  </thead>
  <tbody>
    <tr><td> cella alto sinistra </td><td> cella alto destra </td></tr>
    <tr>
      <td> cella basso sinistra </td><td> cella basso destra </td></tr>
  </tbody>
</table>
```

Questo il risultato:

Descrizione Tabella

titolo head sinistra	titolo head destra
cella body alto sinistra	cella body alto destra
cella body basso sinistra	cella body basso destra

Come meccanismo direi che è abbastanza semplice data la sua logica e di conseguenza di facile comprensione. Una tabella avrà sempre una forma regolare quadrata o rettangolare che sia, è possibile fondere le celle fra di loro sia in modo orizzontale che verticale (righe e colonne) per dare vita a tabelle irregolari, per farlo si adoperano gli attributi: `rowspan` e `colspan` che servono rispettivamente per occupare più di una sola cella in senso orizzontale e più di una sola cella in senso verticale, come tali sono applicabili al solo elemento cella `<td>`.

Vediamo un esempio: supponendo di dover occupare con una sola cella più colonne della tabella regolare adopereremo l'attributo `colspan` specificando il numero di colonne da occupare che nel nostro caso potrà essere al massimo 2. (ovvio, la nostra tabella in questo caso è formata da 2 sole colonne):

```
<table>
  <tr>
    <td colspan="2"> cella alto sinistra </td></tr>
</table>
```

```

<tr>
  <td colspan="2"> Questa cella occupa lo spazio di 2 celle </
td>
</tr>
</table>

```

Questo il risultato:

cella alto sinistra	cella alto destra
Questa cella occupa lo spazio di 2 celle praticamente 2 colonne	

Allo stesso identico modo usando l'attributo rowspan si potranno occupare più righe della tabella regolare, anche in questo caso il valore di rowspan potrà essere al massimo 2 (ovvio, la nostra tabella anche in questo caso è formata da 2 sole righe):

```

<table>
  <tr>
    <td rowspan="2"> questa cella occupa 2 righe </td><td>
    cella alto destra </td>
  </tr>
  <tr>
    <td> cella basso destra </td>
  </tr>
</table>

```

Questo il risultato :

questa cella occupa lo spazio di 2 celle praticamente 2 righe	cella alto destra
	cella basso destra

Combinando i vari attributi è possibile ottenere strutture ottimali, dipende da come vogliamo impaginare i nostri dati.

Nella prossima lezione vedremo altri attributi che è possibile usare con le tabelle.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Le Tabelle: i vari attributi.

Abbiamo visto nella [lezione precedente](#) come creare una tabella e come sia possibile modificare la sua struttura, vediamo adesso alcuni degli attributi più interessanti:.

Oltre al già citato border una tabella ha molti altri attributi, per esempio per regolare lo spazio fra una cella e l'altra o lo spazio fra il bordo ed il contenuto della cella si adoperano **border-spacing** (non supportato dai browser IE) e **padding**.

Questo il codice nel foglio di style:

```
table {
border: solid 2px #000000;
text-align: center;
}
td {
border: solid 1px #000000;
color: Black;
background: #F1EBAD;
padding: 18px;
}
```

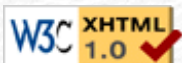
Questo il codice html:

```
<table>
<tr>
<td colspan="3"> riga 1 su 3 colonne </td>
</tr>
<tr>
<td> colonna 1 </td>
<td> colonna 2 </td>
<td> colonna 3 </td>
</tr>
<tr>
<td colspan="3"> fine con riga su 3 colonne </td>
</tr>
</table>
```

Questo il risultato :

riga 1 su 3 colonne		
colonna 1	colonna 2	colonna 3
fine con riga su 3 colonne		

Ad una tabella è possibile assegnare una larghezza predefinita impostando dei



valori fissi (numero di pixel) o in qualsiasi altro valore ammesso dai fogli di style come per esempio "%" in percentuale rispetto alla larghezza della finestra del browser. Si raccomanda di usare questo attributo per evitare che la tabella si ridimensioni dinamicamente con risoluzioni video elevate sconvolgendo di fatto l'impaginazione iniziale.

L'attributo per impostare la larghezza è **width** che oltre alla tabella è possibile assegnare anche alle singole celle, a questo proposito si consideri che in una colonna di celle con valori impostati diversamente per ogni singola cella avrà prevalenza la cella più larga che costringerà tutte le altre facenti parte della stessa colonna ad adeguarsi alla sua misura massima.

Senza una definizione di dimensioni la tabella risulterà essere dinamica, ovvero si allargherà a seconda del contenuto delle sue celle, qui sotto lo stesso esempio visto sopra con del testo aggiuntivo sulle varie celle per dare l'idea di cosa succede.

cella 1 aggiungo del testo in questa cella in modo che si noti meglio cosa succede alla tabella che non avendo dimensioni impostate si allarga fino a raggiungere il massimo dello spazio disponibile, in questo esempio lo spazio è dato dall'elemento progenitore che contiene questa tabella o avrebbe raggiunto i bordi della finestra del browser.		
cella 2 colonna 1 aggiungo del testo così da verificare l'effetto su tre colonne	cella 3 colonna 2	cella 4 colonna 3
cella 5 colonna 1	cella 6 colonna 2	cella 7 colonna 3
cella 8 con riga su 3 colonne		

La tabella si è allargata fino alle sue misure massime per via del testo scritto nella cella 1 che con colspan 3 occupa lo spazio di tre colonne, la cella 2 che forma la colonna 1 ha molto testo che ne allarga le dimensioni adeguando alle stesse anche la colonna 5 che ha meno testo.

Vediamo cosa succede assegnando adesso un valore di width 450 pixel a table ed un valore del 33% a td. La stessa tabella risulta con una larghezza fissa e con celle delle stesse dimensioni (33% = un terzo delle dimensioni della tabella), in questo caso è il testo ad adattarsi dinamicamente alle misure impostate.

cella 1 aggiungo del testo in questa cella in modo che si noti meglio cosa succede alla tabella che non avendo dimensioni impostate si allarga fino a raggiungere il massimo dello spazio disponibile, in questo esempio lo spazio è dato dall'elemento progenitore che contiene questa tabella o avrebbe raggiunto i bordi della finestra del browser.		
cella 2 colonna 1 aggiungo del testo così da verificare l'effetto su tre colonne	cella 3 colonna 2	cella 4 colonna 3
cella 5 colonna 1	cella 6 colonna 2	cella 7 colonna 3
cella 8 con riga su 3 colonne		

Il bordo è possibile definirlo singolarmente e persino differentemente per ognuno dei 4 lati di una cella, abbiamo visto negli esempi precedenti che è possibile impostare il valore numerico in pixel per il suo spessore, adesso con la proprietà **border-color**: inserita nel CSS è possibile definire il colore ma esiste anche il tipo di bordo.

```
td {  
border: solid 1px #ff0000;  
border-color: #ff0000;}
```


Quello da noi adoperato fino a questo momento è sempre stato `solid` ma ne esistono altri che però non sono tutti supportati dai vari browser (IE è uno di questi). Vediamoli descritti con il relativo esempio, ovviamente chi non dovesse vedere differenze sarà soltanto a causa del proprio browser.

Queste le parole chiave per definire il tipo di bordo:

<code>solid</code>	linea continua.
<code>dotted</code>	punteggiato.
<code>dashed</code>	tratteggiato.
<code>double</code>	doppia linea continua.
<code>groove</code>	scavato.
<code>ridge</code>	sbalzato; l'opposto di <code>groove</code> .
<code>inset</code>	premuto.
<code>outset</code>	sollevato; l'opposto di <code>inset</code> .
<code>hidden</code>	nascosto.

Ecco il risultato per i browser che supportano i vari tipi.

bordo solid 1 pixel | bordo solid 1 pixel

bordo dotted 1 pixel | bordo dotted 1 pixel

bordo dashed 1 pixel | bordo dashed 1 pixel

bordo double 5 pixel | bordo double 5 pixel

bordo groove 5 pixel | bordo groove 5 pixel

bordo ridge 5 pixel | bordo ridge 5 pixel

bordo inset 5 pixel | bordo inset 5 pixel

bordo outset 5 pixel | bordo outset 5 pixel

Abbiamo detto che IE non supporta la proprietà `border-spacing` che regola lo spazio fra i bordi delle celle e fra il bordo delle celle e quello della tabella. Per eliminare questo spazio esiste la proprietà `border-collapse` che accetta due valori: `separate` e `collapse`, rispettivamente per separare o unire le celle:

`border-collapse: separate 10px;` | IE separa ma ignora i 10 pixel

`border-collapse: collapse;` | i bordi collassano

Le celle possono essere allineate oltre che orizzontalmente (con le stesse proprietà

e modalità viste per l'allineamento del testo) anche verticalmente con la proprietà: **vertical-align**: che oltre ai valori numerici accetta le seguenti parole chiave: top, middle, bottom, super, sub, baseline, text-top, text-bottom.

colonna 1, aggiungo del testo al solo scopo di allungare la cella e risaltare l'allineamento verticale delle altre celle.	vertical-align: top;	vertical-align: middle;	vertical-align: bottom;
--	--------------------------------	-----------------------------------	-----------------------------------

Si possono cambiare i colori dei bordi ed anche quelli degli sfondi questi ultimi possono essere fatti oltre che da tinte unite anche da immagini grafiche di tipo gif, jpg o png, proprio come abbiamo visto in precedenza per lo sfondo della pagina web e per questo valgono le stesse regole.

colore verde come sfondo della cella e bordo color rosso	colore giallo come sfondo della cella e bordo color blue	colore rosso come sfondo della cella e bordo color verde	colore blue come sfondo della cella e bordo color giallo con testo color bianco
---	---	--	--

immagine jpg come sfondo della cella e bordo color rosso	immagine gif come sfondo della cella e bordo color blue	immagine jpg come sfondo della cella e bordo color verde	immagine jpg come sfondo della cella e bordo color giallo
---	--	--	--

Con questa lezione abbiamo imparato ad usare le tabelle per mostrare i dati in modo ordinato ed elegante.

Nella prossima lezione vedremo i moduli, meglio conosciuti come "forms" che servono per comunicare ed interagire direttamente col visitatore del nostro sito.

Se avete domande da fare potete scrivere sul forum di supporto, gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

I Moduli o Forms.

Abbiamo visto nella [lezione precedente](#) come impaginare i dati facendo uso delle tabelle. Vediamo ora come poter interagire col visitatore ricevendo dati ed informazione tramite un modulo che è possibile compilare e spedire al nostro indirizzo e-mail direttamente dalla pagina web.

Un modulo (form) può essere costituito da diversi elementi quali: caselle per l'introduzione dei dati, menù a tendina dove effettuare scelte preimpostate, caselle di selezione multipla tipo spunta, radio caselle per singole o multiple selezioni, aree di testo più o meno ampie per raccogliere commenti e descrizioni, caselle nascoste per password, pulsanti di invio, pulsanti di annullamento o reset.

Tutti questi elementi sono raccolti e visibili sulla [guida html](#), in questa lezione ci limiteremo all'essenziale per poter creare un piccolo modulo e sarete voi stessi, una volta appresa la tecnica, ad arricchirlo di tutti quegli elementi che riterrete utili al vostro fabbisogno.

I forms possono essere utilizzati per svariate forme di comunicazione, il solo codice html ha però delle limitazioni, nel senso che non è possibile effettuare controlli sui dati introdotti e meno ancora interagire con le scelte dell'utente che li compila. Queste limitazione sono però superabili adoperando in abbinamento un qualsiasi linguaggio di programmazione (scripting) come: php, asp, javascript, con i quali è possibile ricevere o inviare dati a/da pagine diverse, eseguire operazioni matematiche, effettuare controlli sui dati introdotti, anche di tipo piuttosto complesso come verificare se un campo destinato a ricevere un indirizzo e-mail è stato compilato correttamente, se ad esempio al suo interno si è dimenticata la @ che sappiamo essere fondamentale per un indirizzo di posta elettronica e molto altro ancora.

Vediamo come si creano e quali siano gli elementi indispensabili al suo funzionamento: Un modulo ha come elemento base `<form>` tutto quello che sarà inserito al suo interno e fino alla chiusura `</form>` farà parte del modulo e come tale seguirà le azioni previste nel modulo stesso.

L'elemento (tag) form che dichiara il modulo da solo dice poco e niente, necessita di alcuni attributi, almeno due per essere spedito: **action** e **method** ed un terzo opzionale **id** che servono rispettivamente:

action per indirizzare il modulo al server, all' URL, al CGI o all'indirizzo di posta elettronica.

method per dichiarare con quale dei due possibili metodi: **post** o **get** si devono trasmettere i dati. (get è disapprovato)

id Il nome che preferite assegnare al modulo, diventa indispensabile facendo uso di JavaScript o altro linguaggio di programmazione che diversamente non saprebbe a quale modulo fare riferimento nella pagina web. Per ragioni di compatibilità con html precedenti si fa uso anche di **name** che deve avere lo stesso dato di **id**.

Così come abbiamo visto con le tabelle dove il solo `<table>` serviva per creare la struttura affidando a `<tr>` e `<td>` il compito di creare righe e colonne, nel form è indispensabile almeno un elemento **input** che a seconda dell'attributo **type** ad esso associato diventa un campo di testo da compilare



(text), un pulsante per spedire (submit) un pulsante per resettare (reset) o altro ancora.

Vediamo con un esempio pratico quanto detto fino a questo momento.

```
<body>
  <form action="pagina.html" method="post" id="miomodulo">
    <label for="parere">Inserisci il tuo parere:</label>
    <input type="text" id="parere" size="20" />
    <input type="submit" value="Invia" /><input type="reset"
    value="Cancella" />
  </form>
</body>
```

Si tratta di un ipotetico modulo per ricevere un parere e questo sotto il risultato a video:

Inserisci il tuo parere:

Il modulo è perfettamente funzionante come è possibile constatare compilandolo, anche se non è ancora in grado di fare nulla di concreto, vediamo di analizzarlo per capire le varie istruzioni.

form definisce il modulo. **action** (pagina.html) richiama una pagina che in questo caso è in html e come tale impossibilitata ad elaborare dati o prendere decisioni (sarebbe possibile con pagine in ASP o in PHP). **method** (post) il metodo usato per la spedizione. **id** (miomodulo) il nome dell'oggetto modulo al quale poter fare riferimento nel caso in cui servisse.

label for l'etichetta che identifica il campo (parere).

input type="text" id="parere" size="20" un campo di tipo testo (text) con id parere (lo sesso specificato in label for) size 20 sono le dimensioni del campo a video che in questo caso occupa 20 caratteri.

input type="submit" value="Invia" un pulsante di spedizione (submit) con scritta (value) invio.

input type="reset" value="Cancella" un pulsante di tipo annulla (reset) con scritta (value) cancella.

Da notare che il campo parere e i due pulsanti invia e cancella sono stati tutti creati dallo stesso elemento input cambiando semplicemente il valore del suo attributo type. Per dovere di informazione questi i valori che è possibile assegnare, sono in ordine alfabetico: Checkbox, Button, File, Hidden, Image, Password, Radio, Reset, Submit, Text.

Soffermiamoci sul metodo di spedizione, nel nostro esempio sopra basterebbe inserire come action `mailto:nostro_indirizzo@di_posta`, lo stesso attributo visto per i links, ed il modulo sarebbe in grado di farci pervenire i dati inseriti adoperando il programma (client) di posta elettronica di chi lo compila. Questo è sicuramente il metodo più semplice perchè ogni visitatore ha sicuramente un suo programma di posta elettronica installato nel computer, non tutti però sono disposti a divulgare l'indirizzo e-mail, e questo per svariati motivi oltre che per la propria privacy.

Ci sono in rete script gratuiti in CGI ai quali poter inviare il modulo, questi script prendono il nome di sendmail o formmail e non necessitano di dati sensibili come l'indirizzo e-mail di chi lo compila. Purtroppo per ragioni di sicurezza, al fine di contenere anche lo SPAM sempre più dilagante, questa operazione è divenuta ormai possibile soltanto dallo stesso server sul quale si ha il proprio sito. Gli script in cgi provvedono a ricevere i dati, effettuano i

controlli previsti e dopo averli formattati passano all'immediata spedizione (send-mail). Questo servizio è di solito compreso con lo spazio web dove si ha il proprio sito, verificatelo negli accessori o nei servizi messi a vostra disposizione.

Altro metodo, sempre più usato ultimamente, è sfruttare il linguaggio PHP o ASP per chi possiede uno spazio web che supporti questi linguaggi.

Mi rendo conto che per un principiante non sia molto semplice capire questo meccanismo, vi garantisco però che non è neppure troppo difficile, vediamo un paio di esempi che certamente chiariranno meglio ogni cosa, vediamo come farci inviare il form sopra adoperando entrambi i sistemi. Visto che ci siamo aggiungiamo un paio di campi e gli diamo un aspetto grafico per renderlo più professionale.

Spedizione tramite posta elettronica:

```
<body>
  <form action="mailto:vostro_nome@server_di_posta.it"
  method="post" id="miomodulo" enctype="text/plain">

  <label for="nome">Nome: </label>
  <input type="text" id="nome" name="nome" size="20" />

  <label for="cognome">Cognome: </label>
  <input type="text" id="cognome" name="cognome"
  size="20" />

  <label for="parere">Inserisci il tuo parere: </label>
  <input type="text" id="parere" name="parere" size="20" />

  <input type="submit" value="Invia" />
  <input type="reset" value="Cancella" />
  </form>
</body>
```

Abbiamo introdotto un altro attributo per l'elemento form, si tratta di **enctype** che specifica il tipo di contenuto usato per inoltrare il modulo, questo attributo ci garantisce una corretta interpretazione da parte del programma di posta elettronica che dovrà spedire i dati.

Questo il risultato a video facendo uso di una tabella per impaginare gli elementi del form e della stessa tabella vista nella lezione precedente abbiamo usato lo style.

Modulo per ricevere dati e parere	
Nome:	
Cognome:	
Inserisci il tuo parere:	
Il modulo sarà inviato tramite posta elettronica	

Decisamente più elegante... ah, queste tabelle, se non ci fossero andrebbero inventate.

Il modulo funziona, se volete provarlo fate pure ma non sappiamo a chi arriveranno i dati ho infatti dovuto alterare il mio indirizzo e-mail per ragioni

di spam, ma questo è un rischio che chiunque abbia un sito web dovrà correre.

Se questo sistema vi soddisfa non resta che creare tutti i moduli che servono pronti per essere compilati e spediti... è possibile aggiungere anche l'oggetto (o subject) alla e-mail che vi consegnerà i dati che altrimenti avrebbe come oggetto un messaggio di questo genere: *Modulo inviato da Microsoft Internet Explorer* o *Invio modulo da Firefox*, dipende da quale browser di navigazione state adoperando.

Per aggiungere l'oggetto è sufficiente aggiungere `?subject` all'elemento form:

```
<form action="mailto:vostro_nome@server_di_posta.it?
subject=Quello che volete" enctype="text/plain"
name="mio_form">
```

L'altro sistema è invece quello di affidare il tutto ad un server CGI o ad uno script PHP o ASP, la struttura resterebbe la stessa ma al posto di "mailto:vostro_nome@server_di_posta.it" sarebbe inserito URL relativo o assoluto dello script cgi: "/cgi-bin/form-mail.pl" o "http:www.web-link.it/cgi-bin/form-mail.pl"

Ricordatevi che dovremo adoperare un indirizzo di CGI valido, ripeto: informatevi sulla home page del server dove avete il sito, vi assicuro che questo servizio è comune a molti degli spazi web offerti gratuitamente. Spesso non basta conoscere il solo indirizzo CGI, servono anche altri parametri ma sarà lo stesso fornitore del servizio a fornirveli.

Con questa lezione abbiamo imparato come si costruiscono i moduli, una delle cose più difficili per un principiante. Siamo quasi alla fine, non arrendetevi proprio adesso che manca davvero poco. Nella prossima lezione, vedremo i Frames, come gestire più pagine all'interno della stessa.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.





Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

I Frames (le cornici).

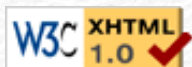
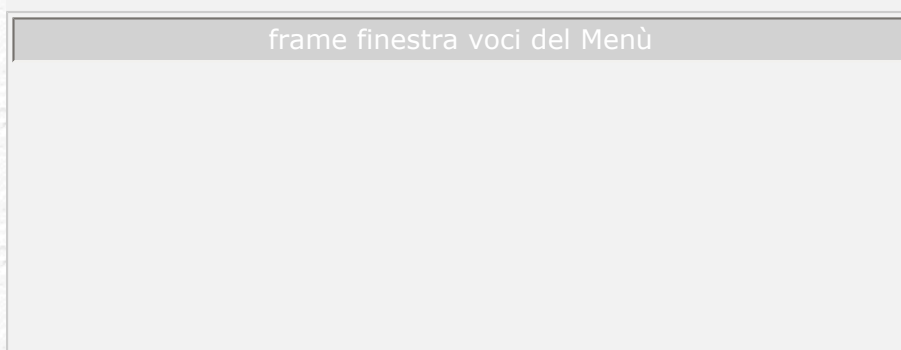
Abbiamo visto nella [lezione precedente](#) come interagire con i visitatori del sito tramite un modulo per la ricezione dei dati in mailbox. Vediamo adesso di trattare un argomento piuttosto discusso e contrastato dalle varie filosofie del fare web, si tratta dei frames: croce e delizia di molti.

I frames permettono di suddividere la stessa pagina web in più pagine indipendenti fra di loro, questa indipendenza consente di avere parti fisse come per esempio i menù o le intestazioni e parti mobili come le varie pagine che si alternano all'interno della stessa struttura. Dicevo croce e delizia perchè sui frames ne ho sentite dire davvero molte, alcune delle quali non corrispondono assolutamente a verità.

Se i frames sono stati inventati a partire da html 3.2 ed ancora oggi in xhtml 1.1 sono ammessi un motivo ci dovrà pur essere o sarebbero stati deprecati come è accaduto a molti altri elementi di html.

I Frames (o cornici) altro non sono che una suddivisione della finestra principale del browser in modo tale che nella stessa ci siano due o più finestre. Quante e come disporle dipenderà soltanto da noi, dal nostro progetto e dalla sua organizzazione. Non esistono limiti se non quelli del buon gusto e della praticità d'uso, che tradotto in termini pratici significa semplicità di navigazione.

In molti casi sono sufficienti due sole cornici (frames), per esempio una laterale (o superiore) dove inserire il menù di navigazione e l'altra la vera e propria finestra di navigazione, quella principale. Per dare meglio l'idea vediamo un paio di possibili esempi, con menù laterale e con menù superiore:



frame Finestra Principale

Quale dei due schemi adottare dipenderà soltanto da noi e da come si è deciso di organizzare il sito, in qualsiasi caso sarà necessario creare una nuova pagina html che contenga al suo interno il **solo codice** per la struttura dei frames. Questa nuova pagina richiamerà all'interno di se stessa tutte le altre pagine (in questo caso due) che si disporranno affiancate verticalmente oppure orizzontalmente a seconda del tipo di struttura impostata. Di solito questa pagina si chiama index.html che è poi la pagina iniziale del sito.

Per prima cosa si preparano le due pagine che saranno richiamate dalla struttura: **menu.html** e **principale.html**, potete adoperare qualsiasi altro nome l'importante è ricordarsi di cambiare poi i riferimenti laddove io negli esempio avrò adoperato questi due nomi.

Se vi ricordate all'inizio di questo tutorial si è parlato della DOCTYPE DTD di inizio pagina che dovremo cambiare da transitional o strict a **frameset**.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web con frames</title>
  </head>
  <body>
  </body>
</html>
```

Portatevi col cursore subito sotto il tag `</head>` ed inserite il nuovo elemento `<frameset>` che serve per dichiarare una struttura a frames.

Avete capito bene, proprio dopo `/head` e prima di `body`, i frames sono gli unici elementi che non vanno inseriti ne dentro a `<head>` testa, ne dentro a `<body>` corpo, della pagina web ma in mezzo, esattamente dove finisce la testa ed inizia il suo corpo.

L'elemento `<frameset>` è uno di quei tags che da soli non dicono niente, necessita di almeno uno dei suoi fondamentali attributi: **rows** o **cols** che servono rispettivamente per suddividere la struttura in righe o in colonne. Al suo interno e prima della relativa chiusura `/frameset` sono contenuti altri elementi di cui uno fondamentale: `<frame>` che serve per richiamare la pagina da incorporare nella struttura.

Supponendo di dover dividere la finestra in modo verticale adopereremo: `<frameset cols="130, *">` assegnando così 130 pixel della finestra del browser al frame piccolo laterale per il menù, ed il restante spazio (* significa tutto il resto) al frame principale, potremo adoperare anche valori espressi in percentuale `<frameset cols="25%, 75%">` nel caso in cui si debbano rispettare delle misure proporzionate alla risoluzione video del visitatore.

Una volta stabilite nel frameset le dimensioni della suddivisione, si dovranno

richiamare i due files preparati precedentemente che in questo caso sono menu.html e principale.html Questo il codice completo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web con frames</title>
  </head>
  <frameset cols="130, *">
    <frame name="menu" id="menu" src="menu.html" />
    <frame name="princ" id="princ" src="principale.html" />
  </frameset>
  <body>
  </body>
</html>
```

L'elemento `<frame>` ha diversi attributi, alcuni di questi sono indispensabili: `name` o `id` che attribuiscono un nome al frame in modo da poterlo usare per richiamare al suo interno una pagina da un link, vedremo più avanti come fare con `target`. `src` (source) che richiama la pagina html vera e propria. `scrolling` con `auto` o `yes` o `no` per la barra di scorrimento laterale, se non specificato si intende `auto`. `frameborder` con valori `0` o `1` per nascondere o mostrare il bordo. `marginwidth` e `marginheight` per la distanza in pixel dal margine laterale sinistro e distanza in pixel dal margine superiore. Vediamo un esempio completo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web con frames</title>
  </head>
  <frameset cols="130, *">
    <frame name="menu" id="menu" src="menu.html" />
    <frame name="princ" id="princ" src="principale.html" />
    <noframe>
  <body>
    <p> Testo visto da chi non supporta i frames</p>
  </body>
  </noframe>
  </frameset>
</html>
```

Abbiamo introdotto l'elemento `<noframe>` che si adopera all'interno di `frameset` e serve per quei browser che non supportassero i frames, soltanto in quel caso sarebbe interpretato il codice al suo interno: `body` ed il suo contenuto altrimenti ignorati.

Se avete preparato le tre pagine non resta che provare a vedere cosa succede, voi richiamate soltanto quella con la struttura, le altre due saranno gestite dalla struttura stessa.

Se avete fatto tutto correttamente dovrete avere le due pagine visualizzate contemporaneamente, una di fianco all'altra nella stessa finestra come nell'esempio che ho messo in [questo link](#).

Ricordatevi i nomi che avete assegnato ai vostri frames, e per nomi intendo quelli specificati nell'attributo `name` e `id`, saranno questi nomi che se

dichiarati all'interno dell'attributo target che abbiamo visto per i links, decideranno su quale dei due frames aprire o richiamare quanto associato al link stesso.

Su [questo link](#) è possibile vedere ben 18 schemi di frames diversi e selezionare il codice html già pronto per quello che potrebbe essere la vostra struttura ideale, attenzione però che quel codice fa riferimento a html 4.01 e certi attributi non sarebbero validi per xhtml ma potrà sempre essere utile per avere un'idea.

Con questa lezione avete imparato a gestire il vostro sito adoperando più pagine web contemporaneamente. Molte persone manifestano una certa diffidenza ad una gestione del sito con i frames, capisco che sia meno accessibile di uno senza frames e trovo qualsiasi altra motivazione non giustificata, c'è chi dice che i siti con frames non vengono indicizzati dai motori di ricerca, è vero ma a causa di una cattiva progettazione, provate ad andare su google e digitare nel box ricerca "fare pagine web" e guardate in quale posizione si trova web-link.it che è sempre stato fatto con frames.

Nella prossima lezione, vedremo come usare iframe.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Iframe (cornice interna).

Abbiamo visto nella [lezione precedente](#) come poter suddividere all'occorrenza una pagina web in più finestre (cornici) ognuna delle quali richiamava una propria pagina web risultando di fatto come una sola ed unica pagina. Vediamo adesso come richiamare sempre una pagina esterna facendo uso di iframe, praticamente un frame interno che non altera la struttura principale perché consente semplicemente un richiamo a questa pagina più o meno come avviene per l'inserimento di una immagine.

Gli Iframes altro non sono che dei frames in linea, in qualsiasi punto della pagina è possibile creare un iframe e richiamare così al suo interno un'altra pagina.

Nel caso di iframe non è necessario usare una DOCTYPE DTD con dichiarazione frameset perché non si tratta di una struttura a frames vera e propria ma di una normale pagina transitional o strict che incorpora o include un frame.

L'elemento `<iframe>` col relativo elemento di chiusura `</iframe>` ha diversi attributi, alcuni di questi sono indispensabili: `src` (source) che richiama la pagina html vera e propria. `name` o `id` che attribuiscono un nome all'iframe in modo da poterlo usare per richiamare al suo interno una pagina da un link, vedremo più avanti come fare con `target`. `scrolling` con `auto` o `yes` o `no` per la barra di scorrimento laterale, se non specificato si intende auto. `frameborder` con valori `0` o `1` per nascondere o mostrare il bordo. `marginwidth` e `marginheight` per la distanza in pixel dal margine laterale sinistro e distanza in pixel dal margine superiore. Vediamo un esempio, questo il codice xhtml:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it"
lang="it">
  <head>
    <title> Titolo della mia pagina Web</title>
    <link rel="stylesheet" type="text/css" href="foglio.css" />
  </head>
  <body>
    <p>Questa è la mia prima pagina web in xhtml</p>
    <iframe src="0intro.html" id="intro" name="intro"
width="300" height="150" marginwidth="3"
marginheight="3">
    </iframe>
  </body>
</html>
```

Questo il risultato:

Vediamo lo stesso codice adoperato per l'esempio sopra ma con l'attributo `frameborder` impostato a `0`

Ancora lo stesso codice adoperato per l'esempio sopra con `frameborder = 0` e `scrolling = no`

Ricordatevi i nomi che avete assegnato ai vostri iframes, e per nomi intendo quelli specificati con l'attributo `name` e `id`, saranno questi nomi che se dichiarati all'interno dell'attributo `target`, che abbiamo visto per i links, decideranno su quale degli iframes aprire e richiamare quanto associato al link stesso. Provate a fare clik su [questo link](#), la pagina richiamata si aprirà sull'iframe immediatamente sopra. Con [questo link](#) rimetteremo la pagina originale. L'uso di `target` è deprecato nella forma strict di xhtml.

Con questa lezione avete imparato ad introdurre all'interno di una vostra pagina web un'altra pagina web facendo in modo che risulti come se fosse una sola ed unica pagina.

Ormai siamo alla fine e nella prossima lezione, vedremo come inserire effetti speciali quali: audio, applets, scripting.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Multimedia: Audio, Applets, Script

Nella [lezione precedente](#) abbiamo appreso l'uso di Iframe e con quello abbiamo concluso avendo praticamente visto quasi tutto ciò che serve realmente per creare un sito web formato da immagini, sfondi e testo. E' però possibile inserire anche effetti speciali come l'audio o delle animazioni particolari.

Audio

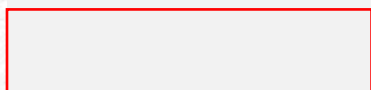
Da premettere che fino ad ora tutto il codice visto che abbiamo avuto modo di adoperare era letto ed interpretato dal browser di navigazione, essendo questo conforme allo standard XHTML ci garantiva la sua visualizzazione ottimale da parte di qualsiasi browser e sistema operativo adoperato.

Con l'audio le cose cambiano perchè non basta il solo browser ne il fatto che il codice sia corretto, questa volta si deve fare uso di "programmi accessori" che di solito sono definiti come plug-in e si basano su programmi esterni veri e propri che sono di regola già installati nei vari sistemi operativi. Per ascoltare un brano audio per esempio, sia questo in formato midi o in MP3, ma anche in uno degli altri possibili formati, si deve avere installato nel proprio PC un programma come il Multimedia Player di Windows, il QuickTime di Apple o altri e tutti quanti in grado di riconoscere e gestire il formato audio adoperato, senza questi plug-in non sarà possibile vedere o sentire nulla nonostante la sintassi magari più che corretta.

Prendiamo la nostra struttura html, quella usata nelle lezioni precedenti, ed inseriamo le istruzioni per richiamare e gestire un brano audio che in questo caso è formato da un file di tipo midi:

```
<body>
  <object type="audio/x-mid" data="nome_midi.mid"
    height="40" width="170">
    <param name="src" value="nome_midi.mid"></param>
    <param name="autostart" value="true"></param>
    <param name="repeat" value="true"></param>
    <param name="loop" value="true"></param>
  </object>
</body>
```

Questo il risultato:



Si dovrebbe visualizzare una parte dell'interfaccia grafica adoperata dal vostro sistema operativo e si dovrebbe udire il brano musicale di sottofondo,



uso il condizionale perchè in realtà questo dipende da molti fattori come ho appena spiegato sopra.

Dovendo illustrare il codice adoperato: si fa uso di **object** e se ne specifica il **type** (audio/x-mid) ed il **data** (titolomidi.mid) in questo caso titolomidi.mid che dovrà essere il nome del brano usato in formato midi ripetuto poi nel parametro sotto relativo al sorgente **src**.

Ci sono anche **height** e **width** per impostare le dimensioni visibili dell'interfaccia grafica del programma adoperato, in questo esempio 40 pixel in altezza e 170 pixel in larghezza. (ottimi per chi ha windows col multimedia player a corredo)

Seguono una serie di parametri relativi a: **autostart**, **repeat**, **loop** e servono per stabilire se il brano deve avviarsi da solo, se deve essere ripetuto e per quante volte, i valori ammessi sono true per si e false per no.

Come dicevo ad inizio di questo argomento molto dipende dal tipo di browser che visualizzerà la pagina, per esempio in certe versioni il comando object potrebbe non essere riconosciuto o male interpretato, si dovrebbe prevedere un riconoscimento del tipo di browser e richiamare così o un codice o l'altro.

Anche se obsoleto per XHTML il comando embed è al momento la sola alternativa per molti tipi e versioni di browser, è possibile adoperarlo in questo modo:

```
<body>
  <embed src="nome_midi.mid" autostart="false" loop="true"
  height="40" width="170">
</embed>
</body>
```

Questo il risultato:

Anche in questo caso si dovrebbe vedere parte dell'interfaccia grafica (170 x 40 px), la muisca non si sente perchè ho usato false in autostart, se state ascoltando il brano è per via del codice precedente, fermate la musica cliccando sul pulsantino di stop dall'interfaccia grafica vista sopra per l'uso di object, premete adesso sul pulsantino play di questo codice e se il vostro browser riconosce embed dovrete ascoltare lo stesso identico brano avviato automaticamente in precedenza.

Uso sempre il condizionale perchè purtroppo le cose sono meno semplici di quello che sembra, il motivo non è il codice adoperato ma le differenti interpretazioni da parte dei browser ed i vari programmi installati nei sistemi operativi, inizialmente c'era persino un comando proprietario dei soli browser Internet Explorer, si trattava di bgsound, poi è stata la volta di embed ed infine object.

Si potrebbe creare codice html che operi delle scelte a seconda del browser adoperato in modo da fargli leggere ed eseguire soltanto il codice più adeguato ma non vi sono certezze sulla garanzia di buon funzionamento.

Questo sotto il codice che opera la scelta del tipo: se il browser è IE e la sua versione è inferiore alla 5 leggi ed esegui embed in qualsiasi altro caso leggi ed esegui object.

```
<body>
  <!--[if gte IE 5]>
  <embed src="nome_midi.mid" autostart="false" loop="true"
  width="170" height="40"></embed>
  <![endif]-->

  <object type="audio/x-mid" data="nome_midi.mid"
  height="40" width="170">
  <param name="src" value="nome_midi.mid"></param>
  <param name="autostart" value="true"></param>
  <param name="repeat" value="true"></param>
  <param name="loop" value="true"></param>
  </object>
</body>
```

Con questa lezione le vostre pagine si sono arricchite di un opzionale sottofondo musicale, vorrei ricordare che sarebbe buona norma lasciare al visitatore la possibilità di avviare e fermare l'audio per non costringerlo ad ascoltare la nostra musica che non sempre è gradita.

Nella prossima lezione impareremo ad inserire un'applet Java.

Se avete domande da fare potete scrivere sul forum di supporto, gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

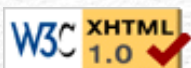
www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Multimedia: Audio, Applets, Script

Nella [lezione precedente](#) abbiamo appreso come inserire una musica o meglio come riprodurre un qualsiasi file di tipo audio all'interno della nostra pagina web, vediamo adesso come poter inserire un applet Java, (programma scritto col linguaggio java) che fa uso oltre al codice html anche di files esterni e che necessita, proprio come per l'audio, di un programma che dovrà essere presente nel sistema operativo.

Questo programma si chiama JVM: Java Virtual Machine e lo si dovrebbe trovare già installato nel proprio sistema operativo ma ultimamente non è più così per note controversie legali sui diritti d'autore, per questo motivo la Microsoft ha deciso di non distribuirlo più a corredo del suo sistema operativo Windows, nessun problema, se manca è possibile reperirlo ed installarlo gratuitamente dal [sito di SUN](#), la casa madre del Java.

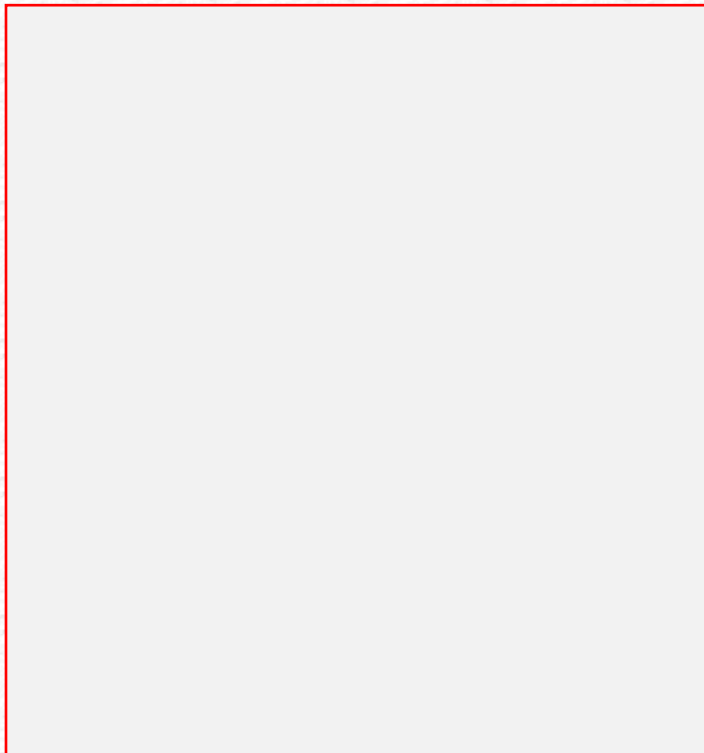
Java Applets

Prendiamo la nostra struttura html, quella usata nelle lezioni precedenti, ed inseriamo le istruzioni per richiamare e gestire un Applet, il quale è sempre formato da almeno un file esterno che ha come estensione .class e che dovrà essere presente nella stessa cartella in cui si trova il file html che lo richiama, per questo motivo dovrà essere trasferito sul nostro spazio web esattamente come tutti gli altri files. (in realtà il file .class potrebbe risiedere anche su una cartella diversa o addirittura su uno spazio web diverso ma non è il caso in questo momento e contesto di complicarsi ulteriormente la vita).

```
<body>
  <object classid="java:Lake.class" type="application/x-java-
  applet" height="350" width="330">
    <param name="image" value="img2.jpg" />
  </object>
</body>
```

Questo il risultato:





Se non vedete nulla non preoccupatevi, è solo perchè state adoperando IE e più sotto capirete il perchè.

Dovendo spiegare il codice adoperato: si fa uso di **object** e se ne specifica la **classid** (java:Lake.class) dove Lake.class è un file che serve a generare l'effetto riflesso e deve trovarsi nella stessa cartella dove c'è questa pagina con il codice html che lo richiama, il **type** (application/x-java-applet) informa il browser che si tratta di un java-applet, quindi un **param name** che dice trattasi di una immagine (quella alla quale applicare l'effetto riflesso) e specifica il nome di questa immagine (img2.jpg)

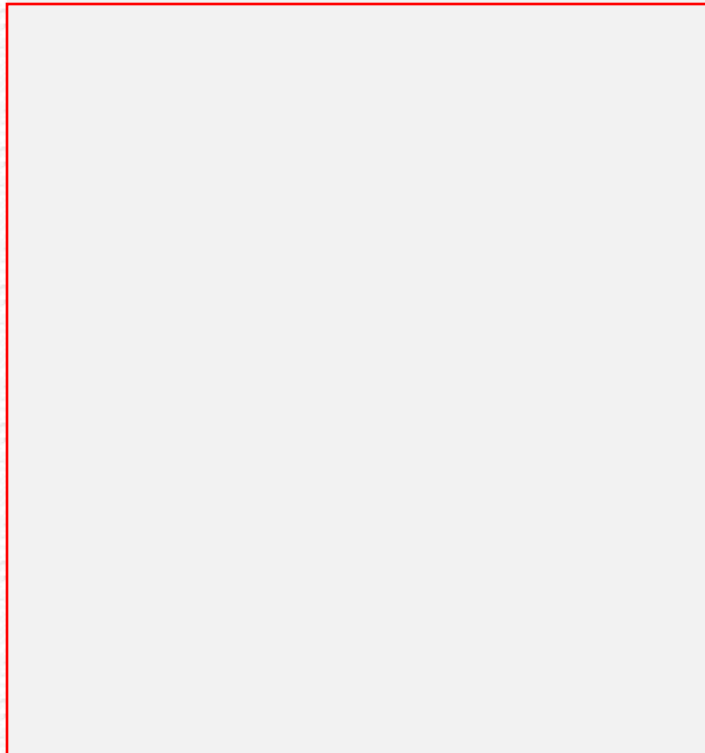
Ci sono anche **height** e **width** per impostare le dimensioni che in questo caso sono relative all'immagine adoperata e alla porzione di questa sulla quale ottenere l'effetto riflesso.

Come dicevo ad inizio di questo argomento: molto dipende dal tipo di browser che visualizzerà la pagina, per esempio in certe versioni il comando **object classid** potrebbe essere diverso ed in alcuni casi non riconosciuto o male interpretato, si dovrebbe prevedere un riconoscimento del tipo di browser e richiamare così o un codice o l'altro; proprio come è stato fatto nella [lezione precedente](#) per i files di tipo audio.

```
<body>
  <!--[if !IE]>-->
  <object classid="java:Lake.class" type="application/x-java-
  applet" height="350" width="330">
  <param name="image" value="img2.jpg" />
  <!--<![endif]-->
  <object classid="clsid:8AD9C840-044E-11D1-B3E9-
  00805F499D93" height="350" width="330">
  <param name="image" value="img2.jpg" />
  <param name="code" value="Lake" />
  </object>
  <!--[if !IE]>-->
  </object>
  <!--<![endif]-->
```

</body>

Questo sotto il risultato e questa volta dovrebbe essere visibile ai vari browsers, diversamente potrebbe mancare nel PC la JVM accennata ad inizio di questa lezione.



Vorrei concludere dicendo che le applets non dovrebbero essere mai usate se il sito tiene conto dell'accessibilità non essendo queste visibili in alcun modo alternativo, spesso si trovano applicazioni java-applet come i menù che sono alla base della navigazione di un sito e non si tiene in considerazione che non tutti riusciranno a vederli e di conseguenza impossibilitati a navigare lo stesso sito.

Per questa lezione credo possa bastare... Avete imparato ad inserire eventuali applets [prossima lezione](#) vedremo come inserire gli script.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto.](#)



Menù Argomenti

Home Page Web-Link
Prefazione
Gli Strumenti
La Struttura base
Il Testo
Le liste
Le Immagini
Immagini e Testo
I links
Links e CSS
Caratteri speciali
Colori e Sfondi
Head e MetaTags
Box Model
Posizionare il Box
Box liquido
Il Layout
Le Tabelle - Struttura
Le Tabelle - Attributi
I Moduli o Forms
I Frames o Cornici
Iframe
Multimedia Audio
Multimedia Applet
Multimedia Script
On line con FTP
Alcuni Esempi
Conclusioni
Forum di supporto
Download .zip

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Multimedia: Audio, Applets, Script

Nella lezione precedente abbiamo appreso come inserire nella pagina web un applet Java adesso vediamo come inserire uno Script java, anzi meglio dire un Javascript, non lasciatevi trarre in inganno del nome Java che li accomuna perchè di fatto hanno davvero poco per non dire nulla in comune, a parte quel pezzo di nome.

Mentre per le applets si doveva usare uno o più files esterni il javascript necessita solo del suo codice che viene letto ed interpretato direttamente dal browser, si tratta di un utilissimo linguaggio di programmazione che non è poi così pericoloso come qualcuno lo definisce, sebbene i soliti male intenzionati tentino di sfruttarlo per creare insidie. Questo linguaggio essendo stato studiato esclusivamente per il browser non necessita di alcuna installazione ne di programmi esterni ne sul server ne sul proprio PC.

JavaScript

Perchè fare uso di javascript?

Semplicemente per rendere dinamiche o interattive le nostre pagine web, anche in questo caso come per le applets si dovrebbe evitarne l'uso nel caso di siti che tengano in alta considerazione l'accessibilità ma credo sia quasi impossibile non farne uso prima o poi.

Ci sono moltissimi siti in rete che distribuiscono script di ogni genere, anche su web-link.it esiste una intera sezione ben nutrita che ne distribuisce per le più svariate esigenze, potete consultarla [su questo link](#). A differenza delle istruzioni che troverete in quelle pagine facendo uso di XHTML il codice non sempre è gradito al validatore e per ottenere una validazione della pagina pur avendo degli script al suo interno conviene mettere il codice javascript in un file esterno e per farlo è sufficiente copiare tutto il suo codice in un file che sarà salvato con estensione **.js**

Facciamo un esempio, prendiamo lo script per la data che si trova qui: <http://www.web-link.it/scripting/3data.htm> lasciamo fuori soltanto le dichiarazioni di `<script>` e `</script>` ed incolliamo il codice in un nuovo file che chiameremo data.js

```

/* Original: Andrea Bianchi
Web Site: http://www.web-link.it */

function WebDate() {
var now = new Date();
var year= now.getYear();if (year < 1000) year += 1900;

nameDay = new Array
('Domenica','Lunedì','Martedì','Mercoledì','Giovedì','Venerdì','Sabato');

nameMth = new Array('Gennaio','Febbraio','Marzo','Aprile','Maggio', 'Giugno',
'Luglio', 'Agosto', 'Settembre', 'Ottobre', 'Novembre','Dicembre');

document.write('<p> Oggi: '+nameDay[now.getDay()]+ ' ' + now.getDate()
+ " " + nameMth[now.getMonth()] + " " + year+'</p>');}

```

Salviamo il tutto e richiamiamo il file appena creato con questa semplice riga di codice che dovrà essere posizionata in head proprio dove avrebbe dovuto trovarsi il codice completo così come da istruzioni in quella pagina.

```

<head>
<script type="text/javascript" src="data.js"></script>
</head>

```



Certi script come questo sono formati da due o più parti di codice e devono essere inseriti in body o in head e questo dipende da quello che ha previsto il suo programmatore il quale avrà fornito istruzioni in merito. Nel nostro esempio una parte del codice dello script andava messa in head e noi abbiamo messo proprio in head il richiamo al file .js che contiene lo script vero e proprio, queste istruzioni vengono poi richiamate dalla parte di codice che si trova nel body, praticamente in questo esempio quello che si trova in body serve soltanto per stabilire il punto esatto della pagina web sulla quale far apparire la data odierna.

```
<body>  
  <span id="webpos">  
    <script type="text/javascript">WebDate();</script>  
  </span>  
</body>
```

Qui non serve capire il codice ma soltanto fare molta attenzione ad inserirlo in modo giusto e corretto, trattandosi di un linguaggio di programmazione è sufficiente una sola virgola, un ritorno a capo non previsto ed il tutto potrebbe non funzionare a dovere, in alcuni casi persino paralizzare il browser che tenterebbe di eseguire il codice sbagliato.

Questo sotto il risultato di quello script:

Oggi: Sabato 19 Agosto 2006

Finisce qui questa rubrica dedicata a XHTML con Style, nella prossima lezione vedremo come trasferire il tutto nel nostro spazio web facendo uso di un programma di FTP.

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Trasferire il vostro lavoro on-line

Ovvero come adoperare un programma per FTP

Nelle lezioni precedenti abbiamo imparato tutto ciò che serve per creare un proprio sito, lo abbiamo fatto usando il più moderno e recente dei metodi: XHTML con i suoi fogli di style, ci siamo limitati all'essenziale è vero ma presunzione a parte sono certo che vi sia tutto quello che serve per costruire un buon sito web.

Non resta adesso che renderlo pubblico così da poterlo condividere con il grande popolo della rete Internet, per farlo abbiamo bisogno per prima cosa di uno spazio web, sia questo gratuito o a pagamento, lo spazio web gratuito che a mio parere è in assoluto il migliore è: altervista.org che offre molto in cambio di niente, non impone nemmeno banner pubblicitari.

Ho parlato di un servizio esterno come quello di [altervista](http://altervista.org) perchè spesso insieme alla connessione ad internet ci viene dato anche uno spazio web che purtroppo e troppo spesso si rivela inadeguato per mille motivi che non sto qui a descrivervi, insomma va bene che è gratis, si fa per dire, ma non per questo deve essere lento e limitato.

Che abbiate già un vostro spazio web o che decidiate di averne uno la procedura è sempre la stessa e cioè si accede a questo spazio tramite una connessione FTP, meglio se ci si affida ad un programma di FTP, non importa se non sapete cosa sia questo ftp, di fatto vi basta conoscere tre dati fondamentali per poter accedere al vostro spazio web:

nome del server ftp
User ID
Password

Vediamo cosa sono, a cosa servono e come si usano:

nome del server ftp lo decide chi vi offre lo spazio web, di solito è il nome dello stesso spazio, in alcuni casi potrebbe essere preceduto dalla sigla ftp, vi faccio degli esempi validi, sono quelli reali che ho io:

weblink.altervista.org
ilmioforum.com
ftp.web-link.it


User ID Lo decidete voi in fase di registrazione, fate particolare attenzione perchè una volta scelto questo nome non sarà più possibile cambiarlo. Ritornando all'esempio sopra io scelsi: [weblink](http://weblink.altervista.org), [ilmioforum](http://ilmioforum.com), e [web-link](http://ftp.web-link.it)

Password Come per la User ID è un dato che decidete voi, a volte viene assegnato automaticamente dal gestore dello spazio web al quale vi iscrivete ma in questo caso potrete poi modificarlo con un nome di vostra scelta. Un consiglio: se possibile evitate password tutte uguali, se da un lato è vero che sono più semplici da ricordare, dall'altro è altrettanto vero che una volta scoperta una si sono praticamente scoperte tutte.

Se siete in possesso di questi tre dati non resta che inserirli in uno dei tanti programmi FTP che la rete offre, anche lo stesso browser di navigazione



diventa all'occorrenza un programma per il trasferimento dei files ftp, è sufficiente digitare nella barra del browser ftp://ftp.web-link.it/ anzichè http://www.web-link.it ovviamente serve poi conoscere nome utente e password. Dal momento che la rete offre appositi programmi FTP e molti dei quali sono gratuiti credo convenga usare uno di questi che ci mette a disposizione diverse opzioni più o meno utili.

A questo punto non importa cosa avete deciso di adoperare, in ognuno di questi ci sarà comunque una maschera dedicata alla configurazione dei tre dati per l'accesso, ed è proprio su quella maschera che dovrete inserirli, se non avete alcun programma FTP e non sapete quale scegliere, potete scaricarne uno da questo mio sito reperibile a [questo link](#) .

Se invece avete già un qualsiasi altro programma FTP fate in modo che con l'introduzione dei tre dati questo possa collegarsi al vostro spazio web.

Una volta connessi vi si presenta una schermata divisa in due finestre: a sinistra il vostro disco rigido stile gestione risorse di windows, a destra il vostro spazio sul server web, quello al quale vi siete appena collegati.

fate un doppio click sui files o sulle cartelle da trasferire e questi automaticamente saranno copiati nello spazio web.

Vi ricordo che la vostra pagina iniziale dovrà necessariamente chiamarsi **index.html** (tutto minuscolo e con la elle finale di html), a meno che non abbiate uno spazio web su server windows dove di solito il nome della prima pagina è **Default.htm** (con la D maiuscola e senza la elle finale di htm)

Se nel vostro PC avete creato delle cartelle con all'interno files o immagini dovrete fare in modo di ricreare le stesse cartelle anche sullo spazio web del server, ci sono programmi che lo fanno in automatico nel momento in cui copiate la cartella, altri invece hanno una voce MAKEDIR o MKDIR che serve per creare una nuova cartella; attenzione ai nomi che dovranno essere perfettamente uguali a quelli che avete nel vostro disco rigido e per uguali intendo qualsiasi carattere: maiuscole, minuscole, spazi vuoti e tutto quello che è relativo ai nomi che li compongono. I sistemi in rete sono molto sensibili a queste differenze, anche alle più impercettibili, da qui il termine "case sensitive".

Se non avete alcuna esperienza in fatto di files, cartelle e relativi percorsi, vi consiglio di lasciare tutto su una sola ed unica cartella e di trasferire il suo contenuto dal vostro disco al vostro spazio web senza dover creare niente, praticamente tutto ciò che è relativo al vostro sito e che avete sul vostro HD lo copiate pari pari sullo spazio web e siete praticamente a posto così.

Se in futuro farete modifiche o aggiornamenti, continuate a farli sulla copia nel vostro disco fisso, non preoccupatevi di cancellare o modificare il file che si trova on-line sul server, questo sarà sovrascritto nel momento in cui invierete i nuovi files modificati visto che dovranno necessariamente avere lo stesso nome di quelli da aggiornare.

Con questa lezione avete imparato a trasferire il vostro lavoro sulla rete, se tutto è andato a buon fine non vi resta che digitare sul vostro browser l' URL che vi è stato assegnato così da vedere finalmente in linea il risultato. Se tutto è ok, potete cominciare a distribuire il vostro indirizzo agli amici, ai parenti e a tutti i conoscenti; cominciate anche ad iscriverlo nei vari motori di ricerca in modo che possa essere trovato da tutti comprese le persone che non vi conoscono.

Ho finito...

o quasi, nella [prossima ed ultima lezione](#) non mi resta che darvi alcuni consigli e salutarvi...

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Alcuni esempi in XHTML con Style.

A chi non piacerebbe avere dei contenitori con bordi smussati come questi?

Questo il codice CSS per ottenerlo:

```

/* CONTENITORI DEI BORDI E DEL CONTENUTO DEL
BOX */

div.BORDI {width: 120px;}

div.CONTENUTO{background-color: #b1d6a7;
color: #000000;
width: 120px;
padding: 5px;}

/* CREAZIONE DEGLI ANGOLI SMUSSATI */

span.TOP, span.BOTTOM{display: Block;
background-color: #FFFFFF;}

span.TOP span, span.BOTTOM span{display: Block;
overflow: Hidden;
background-color: #b1d6a7;
height: 1px;}

span.RIGA1{margin: 0 5px;}

span.RIGA2{margin: 0 3px;}

span.RIGA3{margin: 0 2px;}

span.TOP span.RIGA4, span.BOTTOM span.RIGA4
{margin: 0 1px;
height: 2px;}

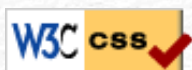
```

Questo il codice HTML per ottenerlo:

```

<div class="BORDI">
<span class="TOP">
<span class="RIGA1"></span>
<span class="RIGA2"></span>
<span class="RIGA3"></span>
<span class="RIGA4"></span>
</span>

```



```
</div>
```

```
<div class="CONTENUTO">  
<b>XHTML con Style</b><br /><br /> Corso gratuito  
e completo per principianti e non<br /><br />  
disponibile su http://www.web-link.it  
</div>
```

```
<div class="BORDI">  
<span class="BOTTOM">  
<span class="RIGA4"></span>  
<span class="RIGA3"></span>  
<span class="RIGA2"></span>  
<span class="RIGA1"></span>  
</span>  
</div>
```

O questo menù di navigazione:

per i quali poter scegliere il colore preferito:

Questo il codice CSS per ottenerlo:

```
#navlist {  
padding: 3px 0;  
margin-left: 0;  
border-bottom: 1px solid #778;  
font: bold 12px Verdana, sans-serif;}  
  
#navlist li {  
list-style: none;  
margin: 0;  
display: inline;}  
  
#navlist li a {  
padding: 3px 0.5em;  
margin-left: 3px;  
border: 1px solid #778;  
border-bottom: none;  
background: #b1d6a7;  
text-decoration: none;}  
  
#navlist li a:link { color: #448;}  
  
#navlist li a:visited { color: #667;}  
  
#navlist li a:hover {  
color: #000;  
background: #009900;  
border-color: #227;}  
  
#navlist li a#current {  
background: white;  
border-bottom: 1px solid white;}
```

Questo il codice HTML per ottenerlo:

```
<div id="navcontainer">
<ul id="navlist">
<li id="active"><a href="#" id="current">pagina uno</a></li>
<li><a href="#">pagina due</a></li>
<li><a href="#">pagina tre</a></li>
<li><a href="#">pagina quattro</a></li>
<li><a href="#">pagina cinque</a></li>
</ul>
</div>
```

Per questo menù trovate maggiori spiegazioni su [questo link](#)

Entrambi gli script non sono di mia invenzione ma frutto anonimo della rete.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Conclusioni

Siamo giunti al termine di questa forse breve ma sicuramente esauriente rubrica dedicata a tutti i principianti (e non) che hanno deciso di imparare a creare un proprio sito facendo uso delle più recenti tecniche in fatto di linguaggio per il web: XHTML e CSS.

Nelle lezioni che hanno formato questa rubrica credo di aver detto tutto quello che c'era da dire per permettervi di creare un vostro sito nel modo più semplice e rapido possibile.

Ho volutamente evitato alcuni argomenti che a mio parere rischiavano soltanto di confondere il principiante ma che sono certo, se vi appassionerete, andrete voi stessi a ricercare e questa volta con la consapevolezza di cosa state cercando.

Le tecniche apprese contano molto ma il successo di un sito dipenderà in gran parte dal vostro buon gusto e dalla vostra fantasia nel saper scegliere e dosare sapientemente i colori, gli sfondi, le immagini, i fonts e quant'altro possa far parte di una pagina web, senza trascurare i contenuti ovviamente.

Tanto per fare un esempio trovo che non ci sia cosa peggiore di una scritta blu o rossa su sfondo nero o comunque scuro ed anche viceversa. Ricordatevi che trattandosi di XHTML si dovrebbe avere un occhio di riguardo all'accessibilità del sito che in questa rubrica è stata appena accennata.

Vi suggerirei inoltre di evitare pagine di lunghe dimensioni e magari farcite da immagini animate, applets, scripts, scritte in movimento e chi più ne ha più ne metta. Tutte cose che appesantiscono la visualizzazione infastidiscono e rendono il tutto molto confusionario. Una pagina web dovrebbe essere la più leggera possibile che si traduce poi in velocità e facilità di lettura dei suoi contenuti.

Forse non tutti sanno che una pagina prima di essere visualizzata dal browser deve essere scaricata nella memoria cache del proprio PC, per cui il tempo per la sua visualizzazione cambia molto a seconda di cosa avete introdotto in quella pagina. Tenete conto anche del tipo di connessione che il vostro visitatore potrebbe avere, non tutti dispongono infatti di una linea a larga banda, se il vostro visitatore ha tempi d'attesa esagerati sarà facilmente indotto ad abbandonare il sito a favore di un altro che offre lo stesso vostro argomento ma difficilmente tornerà a farvi visita.

Dividete le vostre pagine più lunghe facendo in modo che diventino più di una, se l'argomento è interessante saranno sfogliate comunque tutte quante esattamente come si farebbe con un libro.

Pagine che contengono molto testo dovrebbero avere delle piccole immagini inserite nel testo stesso, così facendo si renderebbe meno pesante e faticosa la lettura e si darebbe un insieme di aspetto estetico più gradevole.

Offrite la possibilità di navigare il sito con menù e/o con rimandi a fine pagina, il navigatore dovrebbe muoversi nel modo più agevole possibile senza dover adoperare i tasti del browser di pagina avanti/indietro, mettete voi stessi, in caso di testi su più pagine, la possibilità di sfogliare queste in entrambi i sensi.

Evitate immagini pesanti, se le acquistate voi stessi con uno scanner fate in



modo che siano a 75 dpi. più che sufficienti per la grafica web. Se ne avete di già pronte sottoponetele ad una elaborazione da parte di quei programmi che le "strizzano" un poco riducendone il peso senza alterarne la qualità. Ricordate che se non vi servono immagini animate, o con trasparenze, quelle di tipo .jpg sono da preferirsi essendo in assoluto le più leggere.

Potete controllare che il vostro lavoro sia corrispondente a XHTML inserendo l'URL o il nome della pagina, anche se questa non fosse ancora on line. direttamente da questo sito: <http://validator.w3.org/>

Ho finito e questa volta ho finito sul serio, se come credo farete un vostro sito comunicatemi il suo URL, sul forum c'è un'apposita area denominata check site e mi farà piacere vederlo.

Se trovate errori di qualsiasi genere (anche semplici dimenticanze e/o imprecisioni) comunicatemele e sarò ben contento di poterlo correggere, in fondo si sa: nessuno è perfetto o infallibile ed io faccio parte degli esseri umani :)

Andrea Bianchi

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Struttura elementi e moduli

Questa la suddivisione dei vari moduli e relativi elementi di xhtml, alcuni possono essere inseriti all'interno di altri come fossero scatole cinesi, così facendo si combineranno perché ogni elemento ha i suoi attributi ed i suoi effetti e trovandosi all'interno di un altro elemento eredita eventuali proprietà ed attributi dell'elemento stesso che lo incorpora. Ci sono elementi di blocco come `<p>`, `<div>`, `<h1>` ... `<h6>` ed elementi in linea come `<a>`, ``, ``, `` è scorretto e quindi non consentito introdurre un elemento di blocco all'interno di elementi in linea è invece corretto e quindi consentito il contrario.

Non è logico ne corretto inserire un titolo `<h1>` all'interno di un link `<a>` ma è logico inserire un link `<a>` all'interno di un titolo `<h1>`, oppure un paragrafo `<p>` all'interno di una parte di testo evidenziata con `` ma è corretto evidenziare una parte di testo con `` all'interno di un paragrafo `<p>` tanto per citare alcune probabili combinazioni.

Modulo Struttura

`body, head, html, title`

Modulo Testo

`abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var`

Modulo Hypertext

`a`

Modulo Lista

`dl, dt, dd, ol, ul, li`

Modulo Forms

`form, input, label, select, option, textarea`

Modulo Tabelle

`caption, table, td, th, tr`

Modulo Immagini

`img`

Modulo Object

`object, param`

Modulo Meta comandi

meta

Modulo Link

link

Modulo Base

base

Se avete domande da fare potete scrivere sul [forum di supporto](#), gratuito e aperto a tutti in apposita area XHTML.

Andrea Bianchi

Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul [Forum gratuito di supporto](#).



Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Rinomina del file da Windows

Dal nostro editor adoperato per creare il codice xhtml salviamo il file col nome "index.txt". Se windows non è stato impostato per vedere le estensioni dei files a video potremo vedere soltanto index, come già detto infatti, windows nella sua configurazione originale nasconde le estensioni dei files riconosciuti dal sistema e noi non vediamo questa estensione ma di fatto l'estensione c'è e lo stesso sistema la interpreta come tale.

Adesso se noi facciamo l'operazione di rinomina file (tasto destro del mouse o F2) otterremo una cosa di questo tipo: "index.html". A questo punto sarebbe facile credere di essere riusciti a modificare il tipo di file, purtroppo però non è così; per il sistema windows di fatto quel file è rimasto di tipo txt con la sola differenza che il suo nome adesso è diventato: "index.html.txt". Noi però non lo vediamo a causa dell'impostazione nascondi l'estensione dei files riconosciuti dal sistema operativo.

Complicato vero? Credo che per uno che non ha dimestichezza con i nomi dei files e con le operazioni di sistema sia molto più complesso che non creare una pagina web.

E' possibile rendersi conto che la modifica al nome non ha avuto alcun effetto guardando l'icona grafica che il sistema ha assegnato al nostro files, anche dopo l'operazione rinomina, di fatto è rimasta la stessa di prima, quella cioè associata ai files di testo e invece avrebbe dovuto cambiare per essere associata a quella del browser dal momento che i files di tipo html vengono aperti col browser e non con l'elaboratore di testi.

Si ma allora come fare ?

Bisogna disattivare dalle impostazioni di windows questa caratteristica di nascondere le estensioni ai nomi dei files che se per certi aspetti potrebbe risultare comoda per certi altri (è il nostro caso) non lo è affatto.

Non temete, non si faranno danni al sistema, se proprio non vi piace questa opzione potrete sempre ripristinarla una volta rinominato il file in modo corretto, vi basterà rimettere il segno di spunta laddove lo andrete a togliere.

Prima però fate questo tentativo: slavate il nome del file mettendolo all'interno delle doppie virgolette "index.html" e salvate, se funziona non vi serve altro e potete evitare di continuare a leggere queste istruzioni.

Non ha funzionato?

Pazienza, vediamo come procedere, questi i passaggi per windows XP ma le operazioni sono se non proprio le stesse sono molto simili per qualsiasi altro sistema windows, si tratta di individuare la voce:

"[] nascondi le estensioni per i tipi di file conosciuti"

che in Windows XP è raggiungibile aprendo una qualsiasi cartella, quindi alla voce

- strumenti -
- opzioni cartella -
- visualizzazione -

Nell'elenco di voci una delle prime dice proprio: nascondi le estensioni per i



tipi di file conosciuti; togliete il segno di spunta dalla casella quadrata sulla sinistra della descrizione.

Perfetto, un clic sulla voce applica o ok e adesso il vostro file si lascerà vedere in modo integrale e soltanto adesso sarà possibile rinominarlo in modo corretto.

Cliccate sopra il file (una sola volta) e premendo il tasto F2 oppure col tasto destro del mouse scegliete la voce - rinomina -

Portatevi col cursore sulla sigla .txt e cancellatela in modo da poterla sostituire con .html, nel momento in cui confermerete questa operazione windows vi avviserà della possibilità che il file così rinominato non sia più utilizzabile, non fate caso a questo msg e confermate. Se tutto è ok l'icona è cambiata, adesso c'è quella del vostro browser.

Non resta che cliccarci sopra due volte o trascinarla sul programma di navigazione, potrete così vedere finalmente la vostra prima pagina web, che spero, dopo tutto questo lavoro, funzioni almeno correttamente.

Se non vedete nulla, o il risultato non è quanto vi aspettavate, non scoraggiatevi!

Probabilmente si tratta di un piccolissimo e banale errore di distrazione, riguardate attentamente quello che avete fatto, sono certo che troverete la causa.

Andrea Bianchi

Clicca per informazioni !!!

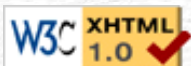
www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.





Ingrandite o rimpicciolite la finestra del browser, il logo occuperà sempre l' 80% della dimensione della finestra

```
body {  
background-image: url(immagine2.jpg);  
color: Blue;  
}
```

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina.

```
body {  
background-image: url(immagine2.jpg);  
background-attachment: scroll;  
color: Blue;  
}
```

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina lasciata libera **scroll** di muoversi durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina lasciata libera **scroll** di muoversi durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina lasciata libera **scroll** di muoversi durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina lasciata libera **scroll** di muoversi durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina lasciata libera **scroll** di muoversi durante lo scorrimento della pagina.


```
body {  
background-image: url(immagine2.jpg);  
background-attachment: fixed;  
color: Blue;  
}
```

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina e tenuta ferma da **fixed** durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina e tenuta ferma da **fixed** durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina e tenuta ferma da **fixed** durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina e tenuta ferma da **fixed** durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina e tenuta ferma da **fixed** durante lo scorrimento della pagina.

Immagine grafica per lo sfondo ripetuta fino al riempimento dell'intera pagina e tenuta ferma da **fixed** durante lo scorrimento della pagina.

```
body {  
background-image: url(immagine2.jpg);  
background-repeat: repeat-y;  
color: Blue;  
}
```

Immagine grafica per lo sfondo ripetuta soltanto in senso verticale.

```
body {  
background-image: url(immagine2.jpg);  
background-repeat: repeat-x;  
color: Blue;  
}
```

Immagine grafica per lo sfondo ripetuta soltanto in senso orizzontale.

```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina.



```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: top;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata **in cima** alla pagina.


```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: center;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata al **centro** della pagina.



```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: bottom;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata a **fondo** della pagina.



```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: left;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata a **sinistra** della pagina.



```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: right;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata a **destra** della pagina.



```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: 200px 200px;  
color: Blue;  
}
```

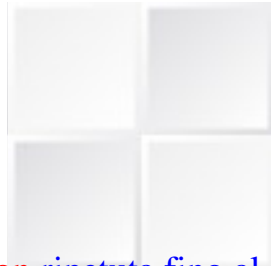


Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata ad una distanza di 200 pixel dal margine sinistro e 200 pixel dal bordo superiore della pagina.

Se variate le dimensioni della finestra del browser la posizione dell'immagine non cambia.


```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: 30% 50%;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata ad una distanza del 30% dal margine sinistro ed al 50% dal bordo superiore della pagina.

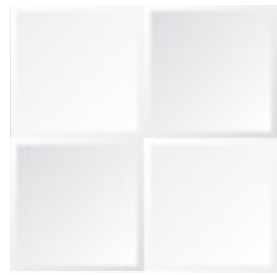
Provate a variare le dimensioni della finestra del browser l'immagine si riposiziona rispettando le percentuali impostate.



```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: bottom left;  
color: Blue;  
}
```

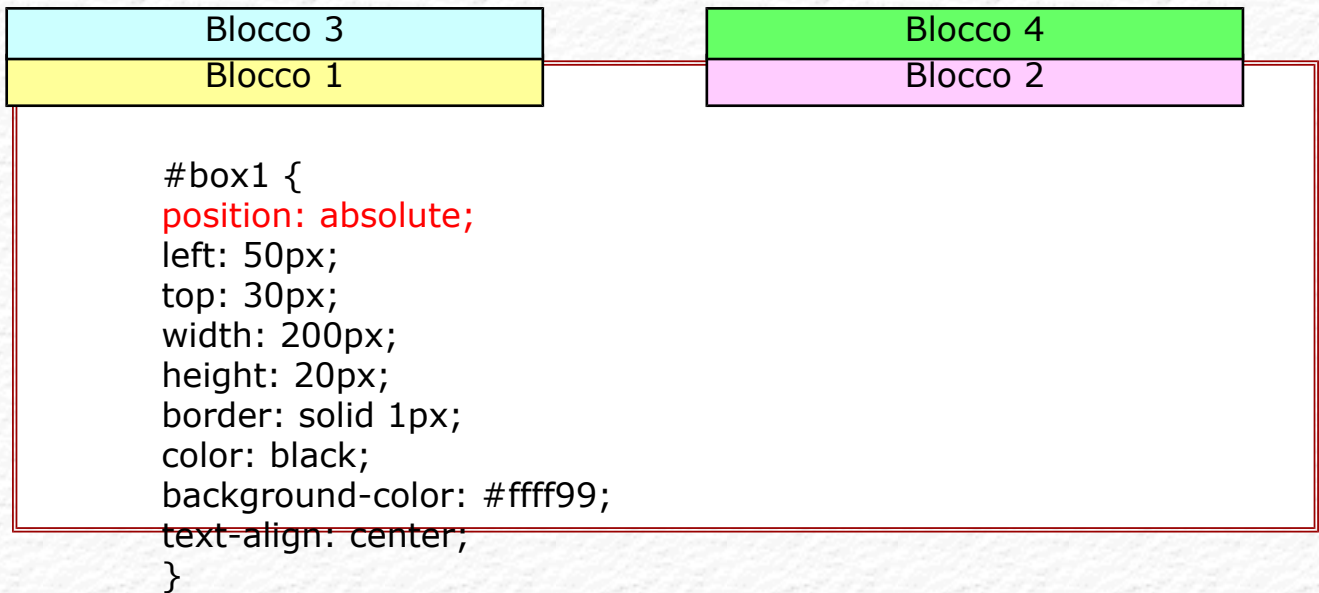
Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata a fondo pagina a **sinistra**.





```
body {  
background-image: url(immagine2.jpg);  
background-repeat: no-repeat;  
background-position: top right;  
color: Blue;  
}
```

Immagine grafica per lo sfondo **non** ripetuta fino al riempimento dell'intera pagina e posizionata **in cima** alla pagina a **destra**.



position: absolute Il codice CSS sopra è relativo al blocco 1, gli altri blocchi hanno praticamente lo stesso codice con coordinate di posizionamento differenti:

```
box2: right: 50px; top: 30px;  
box3: left: 50px; bottom: 30px;  
box4: right: 50px; bottom: 30px;
```

Provate ad allargare o a restringere la finestra del browser e noterete con ei 4 box mantengono la loro posizione assoluta rispetto alla finestra del browser.

Questo box è centrato orizzontalmente e se provate a ridimensionare la finestra del browser il box e quanto contenuto al suo interno cambieranno posizione adattandosi perfettamente alle nuove dimensioni.

C'è però un bordo perimetrale che è possibile togliere come vedremo nel prossimo esempio.

Questo box è centrato orizzontalmente e se provate a ridimensionare la finestra del browser il box e quanto contenuto al suo interno cambieranno posizione adattandosi perfettamente alle nuove dimensioni.

Non ci sono spazi fra il box e la finestra del browser perchè margin e padding sono stati impostati a 0 nel body.

Questo box è centrato orizzontalmente e se provate a ridimensionare la finestra del browser il box e quanto contenuto al suo interno cambieranno posizione adattandosi perfettamente alle nuove dimensioni.

Ci sono spazi fra il box e la finestra del browser perchè al padding nel body sono stati impostati dei margini a 5% della finestra del browser. Nel box la dimensione width è stata impostata al 100% anche se con certi browser non sarebbe necessario.

Questo box è centrato verticalmente e se provate a ridimensionare la finestra del browser il box e quanto contenuto al suo interno cambieranno posizione centrandosi perfettamente alle nuove dimensioni.

Questa la parte sinistra
destinata a contenere i links
che fanno da menù per le
informazioni visualizzate
poi nel box contenitore
centrale.

voce 1
voce 2
voce 3
voce 4
voce 5
ecc.ecc.

XHTML con Style: il layout

Testo dimostrativo inserito solo per riempire la pagina

Testo dimostrativo inserito solo per riempire la pagina

Testo dimostrativo inserito solo per riempire la pagina

Testo dimostrativo inserito solo per riempire la pagina

Testo dimostrativo inserito solo per riempire la pagina

Testo dimostrativo inserito solo per riempire la pagina

Testo dimostrativo inserito solo per riempire la pagina



frame laterale
per il menù

frame centrale

questo è un link
che richiamerà
una nuova
pagina sul frame
qui di lato.

Menù Argomenti

[Home Page Web-Link](#)[Prefazione](#)[Gli Strumenti](#)[La Struttura base](#)[Il Testo](#)[Le Liste](#)[Le Immagini](#)[Immagini e Testo](#)[I links](#)[Links e CSS](#)[Caratteri speciali](#)[Colori e Sfondi](#)[Head e MetaTags](#)[Box Model](#)[Posizionare il Box](#)[Box liquido](#)[Il Layout](#)[Le Tabelle - Struttura](#)[Le Tabelle - Attributi](#)[I Moduli o Forms](#)[I Frames o Cornici](#)[Iframe](#)[Multimedia Audio](#)[Multimedia Applet](#)[Multimedia Script](#)[On line con FTP](#)[Alcuni Esempi](#)[Conclusioni](#)[Forum di supporto](#)[Download .zip](#)

Rev. 0.9 - Agosto 2006

Ver. 0.1 - Anno 2005

Argomento e pagina in preparazione.

Visita il [sito di riferimento](#) per vedere se questo argomento è stato portato a termine o per eventuali aggiornamenti.



Clicca per informazioni !!!

www.web-link.it



per stampare questa pagina



Potete fare domande e discutere l'argomento sul Forum gratuito di supporto.

Iframe (cornice interna). Target nome dell' iframe

Questo è un esempio per dimostrare il richiamo di una pagina all'interno di un iframe grazie all'attributo target nome dell'iframe.

Iframe (cornice interna).

Questo è un esempio scritto su una pagina e richiamato all'interno di questa pagina grazie all'elemento `<iframe>` che ha diversi attributi, alcuni di questi sono indispensabili: `name` o `id` che attribuiscono un nome all'iframe in modo da poterlo usare per richiamare al suo interno una pagina da un link, vedremo più avanti come fare con `target`. `src` (source) che richiama la pagina html vera e propria. `scrolling` con `auto` o `yes` o `no` per la barra di scorrimento laterale, se non specificato si intende `auto`. `frameborder` con valori `0` o `1` per nascondere o mostrare il bordo. `marginwidth` e `marginheight` per la distanza in pixel dal margine laterale sinistro e distanza in pixel dal margine superiore.

Pagina richiamata dal link sul frame laterale, come potete vedere quel frame è rimasto lo stesso ed è cambiato soltanto questo, il tutto è possibile grazie al suo nome (princ) usato nel target del link stesso.

[Torna alla pagina precedente](#)

[Torna alla lezione sui frames](#)

frame centrale