

TIPO: CSS

ATTRIBUTI DI MARGINI ED ALLINEAMENTO

I fogli di stile permettono di giustificare il testo senza l'uso improprio di tabelle o espedienti simili. I CSS creano degli elementi visualizzati all'interno di box. Ogni box ha delle proprietà che si possono dividere in tre categorie:

* Le proprietà dei margini: "margin" che assegnano un bordo esterno al box;

* le proprietà di riempimento: "padding" che assegnano uno spazio interno al box che separa il contenuto dai margini;

* le proprietà dei bordi: "border" che definiscono le linee grafiche intorno al box.

I valori di questi attributi sono espressi in pollici (in), centimetri (cm), pixel (px), punti (pt) e unità (em).

Semplificando il concetto, i fogli di stile sono impostati come se si trattasse di finestre indipendenti all'interno della pagina. Hanno, quindi, una loro grandezza e valori di impostazione. Tra questi, oltre al testo ed al background, sono presenti i valori relativi ai bordi di tali pseudo-finestre.

TIPO: CSS

ATTRIBUTI WIDTH / HEIGHT

Si riferisce alla larghezza od altezza del box. Può essere espresso nelle unità di misura finora analizzate. Il valore "auto" permette al box di adattarsi al contenuto degli elementi:

```
BODY { width: 300px;
        height: 40px; }
```

TIPO: CSS

ATTRIBUTO FLOAT

Questo attributo ha una funzione simile ad ALIGN in HTML classico. I valori a disposizione sono left, right e none:

```
BODY { float: right }
```

Un esempio :

Ad esempio la seguente riga di comando allinea l'immagine a sinistra ed imposta il margine sinistro a 0 :

```
IMG.icon {float: left;
           margin-left: 0;}
```

oppure:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN" >
<HTML>
<HEAD>
  <TITLE>Float example</TITLE>
  <STYLE type="text/css">
    IMG { float: left }
    BODY, P, IMG { margin: 2em }
  </STYLE>
</HEAD>
<BODY>
  <P><IMG src=img.gif alt="This image will illustrate floats">
    Some sample text that has no other...
</BODY>
```

</HTML>

TIPO: CSS

ATTRIBUTO BORDER

border-top,
border-bottom,
border-right,
border-left

Questi attributi definiscono lo stile e il colore di ogni elemento e vengono impostati con alcune parole chiave:

none, dotted, dashed, solid, double, groove, ridge, inset e outset:

```
BODY { border-top: dotted green;  
        border-left: dashed black;  
        border-right: solid blue;  
        border-bottom: groove yellow }
```

border-top-width,
border-bottom-width,
border-right-width,
border-left-width

La funzione di questi attributi e' di generare un effetto simile ai bordi delle table HTML. E' possibile stabilire misure percentuali o di riferimento (em), oltre a parole chiave quali thin, medium e thick:

```
BODY { border-top-width: 5px;  
        border-left-width: 4pt;  
        border-right-width: thick;  
        border-bottom-width: 2em }
```

border-width

Come per gli attributi margin e padding, anche i quattro valori visti in precedenza possono essere definiti con un solo attributo border-width:

```
BODY { border-width: 10pt 10px 1em 2in }
```

border-color

Questo attributo definisce i colori dei quattro bordi del CSS nella loro globalita', cioe' senza possibilita' di stabilire colori diversi per ognuno di essi:

```
BODY { border-color: green }
```

border-style

Questo attributo definisce lo stile dei quattro bordi del CSS nella loro globalita', cioe' senza possibilita' di stabilire stili diversi per ognuno di essi.

I valori sono specificati da alcune parole chiave:

none, dotted, dashed, solid, double, groove, ridge, inset, outset.

Gli ultimi quattro valori sono tridimensionali, sempre che il browser li supporti:

```
BODY { border-style: groove }
```

Attributo border

Si tratta di una sorta di attributo omnicomprensivo degli ultimi visti in questa pagina. Racchiude tutte le varianti relative ai bordi, agli stili, al colore e alla larghezza. Assegna gli stessi valori a tutti i

lati, annullando gli effetti di altri attributi simili in precedenza inseriti:

```
BODY { border: 12px groove green }
```

TIPO: CSS

ATTRIBUTO MARGIN

I quattro valori visti in precedenza possono essere definiti con il solo attributo margin. Quando viene impostato questo valore, il browser assume come ordine:

(1) il primo margine in alto (margin-top), (2) il destro (margin-right), (3) l'inferiore (margin-bottom) e (4) il sinistro (margin-left).

Se ne vengono forniti solo due o tre, quelli mancanti vengono automaticamente assegnati in base al valore del lato regolarmente impostato:

```
BODY { margin: 10px 10px 10px 10px }
```

In questo primo esempio tutti gli elementi hanno uguale distanza rispetto agli elementi vicini, e ognuno di essi e' correttamente indicato.

```
BODY { margin: 10px 10px }
```

In questo secondo esempio sono impostati solo due valori (margin-top e margin-right). I mancati margini sinistro e inferiore vengono desunti rispettivamente dal valore del margine destro e superiore.

Un' ulteriore esempio :

```
BODY { margin: 5em } /* all margins 5em */  
P { margin: 2em 4em } /* top and bottom margins 2em,  
left and right margins 4em */  
DIV { margin: 1em 2em 3em 4em } /* top margin 1em,  
right margin 2em,  
bottom margin 3em,  
left margin 4em */
```

TIPO: CSS

ATTRIBUTO PADDING

Attributo padding
padding-top,
padding-bottom,
padding-right,
padding-left

Tali attributi indicano la distanza tra i lati del box e gli elementi al suo interno. In altre parole hanno una funzione opposta a quella vista in precedenza per l'attributo margin. E' possibile usare le misure standard (pollici, centimetri, punti, pixel ecc.) o valori percentuali:

```
<STYLE>  
BODY { padding-top: 10pt;  
padding-right: 10px;  
padding-bottom: 110%;  
padding-left: 2in }  
</STYLE>
```

Attributo padding

Come per l'attributo margin, anche i quattro valori visti in precedenza possono essere definiti con un solo attributo padding. Quando viene impostato questo valore, il browser assume come ordine:

(1) il primo valore di riempimento in alto (padding-top), (2) il destro (padding-right), (3) l'inferiore (padding-bottom) e (4) il sinistro (padding-left).

Se ne vengono forniti solo due o tre, quelli mancanti vengono automaticamente assegnati in base

al valore del lato impostato:

```
BODY { padding: 10pt 10px 110% 2in }
```

In questo primo esempio tutti gli elementi hanno uguale distanza rispetto agli elementi vicini, e ognuno di essi e' correttamente indicato.

```
BODY { padding: 10pt 10px }
```

TIPO: CSS

CSS: COME INGRANDIRE UNA IMMAGINE AL PASSAGGIO DEL MOUSE

```
#contenitore {  
position: relative;  
top: 10px;  
left: 50px;  
width: 120px;  
height: 105px;  
background-color: transparent;  
z-index: 100;  
}
```

```
#contenitore a img {  
border: 0;  
}
```

```
#contenitore a.miniatura, #contenitore a.miniatura:visited {  
display: block;  
width: 120px;  
height: 105px;  
text-decoration: none;  
background-color: transparent;  
border: 0;  
}
```

```
#contenitore a .grande {  
display: block;  
position: absolute;  
width: 1px;  
height: 1px;  
top: -1px;  
left: -1px;  
background-color: transparent;  
}
```

```
#contenitore a.miniatura:hover .grande {  
display: block;  
position: absolute;  
top: 130px;  
left: 180px;  
width: 350px;  
height: 307px;  
}
```

Il codice HTML potrebbe essere:

```
< div id="contenitore" >
< a class="miniatura" href="#" title="Immagine in miniatura" >
< img src="http://www.nomesito.com/miniatura.gif" title="Immagine in miniatura" alt="Immagine in miniatura" >
< img class="grande" src="http://www.nomesito.com/grande.gif" title="Immagine a dimensione reale" alt="Immagine a dimensione reale" >
< /a >
< /div >
```

TIPO: CSS

DISTANZA TRA BORDO SUPERIORE E INFERIORE

height

Questa proprietà definisce la distanza tra il bordo superiore e quello inferiore di un elemento. Non è ereditata e si applica a tutti gli elementi tranne:

- * colonne di tabelle
- * elementi inline non rimpiazzati

Sintassi

```
selettore {height: <valore>;}
```

Valori

- * un valore numerico con unità di misura.
- * un valore in percentuale. Il valore in percentuale si riferisce sempre all'altezza del blocco contenitore, purché esso abbia un'altezza esplicitamente dichiarata. Diversamente, la percentuale viene interpretata come auto.
- * auto. L'altezza sarà quella determinata dal contenuto.

L'uso di height va sempre valutato con attenzione e non pensando di farne una scorciatoia per avere layout precisi al pixel. Il caso più importante da valutare è quando il contenuto dovesse superare i limiti imposti tramite la proprietà. Il comportamento dei vari browser è al riguardo notevolmente diverso, come si può osservare nell'esempio. Explorer non fa altro che ignorare l'altezza impostata estendendola. Mozilla e Opera rispettano la regola, ma il contenuto eccedente va a sovrapporsi agli elementi vicini o sottostanti. Ricordatelo: molte volte si può ottenere lo stesso risultato visivo desiderato usando proprietà come padding e margin. E considerate sempre che l'altezza è indirettamente influenzata anche dalla larghezza di un elemento.

Esempi

```
div {height: 250px;}  
p.blocco {height: 50%;}
```

TIPO: CSS

ESEMPIO

PARAGRAPH CODE

```
p:first-letter{  
    text-transform: uppercase;  
}  
p {  
    color: #424242;  
    font-family: "Adobe Caslon Pro", "Hoefler Text", Georgia, Garamond, Times, serif;  
    letter-spacing: 0.1em;  
    text-align: center;  
    margin: 40px auto;  
    text-transform: lowercase;  
    line-height: 145%;  
    font-size: 14pt;  
    font-variant: small-caps;  
}
```

TWITTER HEADLINE

```
font-family: "Adobe Caslon Pro", "Hoefler Text", Georgia, Garamond, Times, serif;  
font-style: italic;  
color: #424242;}
```

```
a {      font-style: normal;
font-variant: small-caps;
text-decoration: none;
color: #afafaf;
font-size: 14px;
}
```

TIPO: CSS

ESEMPIO

HEADLINE

```
font-family: Georgia,"Times New Roman",serif;
font-size: 12px;
font-weight: bold;
color: #600;
line-height: 22px;
margin: 0;
text-transform: uppercase;
letter-spacing: 1px
```

DATE

```
font-family: Georgia,"Times New Roman",serif;
font-size: 10px;
line-height: 22px;
text-transform: uppercase;
letter-spacing: 2px;
```

TIPO: CSS

ESEMPIO CSS

HEADLINES

```
font-family:georgia, serif;
color:#381704;
font-size:10px;
letter-spacing:0.1em;
line-height:200%;
padding-top:11px;
```

NUMBER

```
font-family:georgia, serif;
color:#3B200F;
font-size:16px;
font-weight:bold;
line-height:125%;
text-align:center;
```

QUIPPED SECTION

```
font-family:georgia, serif;
color:#786E69;
font-size:10px;
font-weight:bold;
letter-spacing:.1em;
text-transform:uppercase;
padding-bottom:3px;
```

```
font-family:georgia, serif;
color:#786E69;
font-size:10px;
font-weight:bold;
font-style:italic;
letter-spacing:.1em;
padding-bottom:35px;
```

PARAGRAPH

```
font-family:georgia,serif;
color:#381704;
font-size:12px;
font-weight:normal;
line-height:150%;
padding:0px;
```

TIPO: CSS

ESEMPIO CSS

HEADLINE

```
font-family:Georgia,serif;
color:#4E443C;
font-variant: small-caps; text-transform: none; font-weight: 100; margin-bottom: 0;
```

PARAGRAPH

```
font-family: "Helvetica Neue", "Lucida Grande", Helvetica, Arial, Verdana, sans-serif;
font-size: 14px;
margin-top: .5em; color: #666;
```

PARAGRAPH START

```
font-family:Georgia,serif;
font-size: .8em;
font-weight: bold;
text-transform:uppercase;
letter-spacing:2px;
```

TIPO: CSS

ESEMPIO MENU ROSA

```
htm: <div id="menù" style="width:150px">
<a href="#" class="stra2">Home Page</a>
<a href="#" class="stra2">Biografia</a>
<a href="#" class="stra2">Contatti</a>
<a href="#" class="stra2">Foto</a>
</div> CSS:A.stra2{DISPLAY: block; text-align:left; padding-left:4px; width:auto; line-height:25px;
height:25px; border-bottom:1px solid #fff; text-decoration:none; background-color:#F18195;
color:#fff}
A.stra2:hover {TEXT-DECORATION: none; background-color:#F52B63; font-weight:bold; }
```

TIPO: CSS

I BORDI

Stile dei bordi
dotted

Codice:

```
#div1 {
border : 2px dotted Black;
background: silver;
width: 300px;
}
```

dashed

Codice:

```
#div2 {
border : 2px dashed Black;
```



```
background: silver;
width: 300px;
}
```

solid

Codice:

```
#div3 {
border : 3px solid Black;
background: silver;
width: 300px;
}
```

double

Codice:

```
#div4 {
border : 5px double Black;
background: silver;
width: 300px;
}
```

groove

Codice:

```
#div5 {
border : 5px groove gray;
background: silver;
width: 300px;
}
```

ridge

Codice:

```
#div6 {
border : 5px ridge gray;
background: silver;
width: 300px;
}
```

inset

TIPO: CSS

IMMAGINI CHE SI INGRANDISCONO AL PASSAGGIO DEL MOUSE

Desiderate ottenere l'effetto che al passaggio del mouse l'immagine della vostra tabella si ingrandisca? Seguite il tutorial e riuscirete a farlo.

Anteprima e codice disponibili qui: [Dynamicdrive](#) , passare col mouse sulle due immagini.

Procediamo spiegando i vari passi da seguire.

Potete scegliere indifferentemente il Procedimento 1 o il Procedimento 2, il passo finale sarà poi il medesimo.

Procedimento 1:

Inserire in Modifica colori e stili il seguente codice, la posizione è indifferente, ma suggerisco di metterlo alla fine, per trovarlo più facilmente in futuro.

CODICE/*Credits: Dynamic Drive CSS Library */

/*URL: [URL=http://www.dynamicdrive.com/style/]http://www.dynamicdrive.com/style/[URL] */

```
.thumbnail{
position: relative;
z-index: 0;
}
```

```
.thumbnail:hover{
background-color: transparent;
z-index: 50;
}
```

```
.thumbnail span{ /*CSS for enlarged image*/
position: absolute;
background-color: lightyellow;
padding: 5px;
left: -1000px;
border: 1px dashed gray;
visibility: hidden;
color: black;
text-decoration: none;
}
```

```
.thumbnail span img{ /*CSS for enlarged image*/
border-width: 0;
padding: 2px;
}
```

```
.thumbnail:hover span{ /*CSS for enlarged image on hover*/
visibility: visible;
top: 0;
left: 60px; /*position where enlarged image should offset horizontally */
}
```

N.B.: Al primo cambio di skin (cambio del CSS) tale codice sparirà.

Procedimento 2:

Inserire il codice seguente in Gestione codici HTML, nel box Codice html che verrà visualizzato in ogni pagina in cima al forum:

CODICE<style type="text/css">

/*Credits: Dynamic Drive CSS Library */

/*URL: [URL=http://www.dynamicdrive.com/style/]http://www.dynamicdrive.com/style/[URL] */

```
.thumbnail{
position: relative;
z-index: 0;
}
```

```
.thumbnail:hover{
background-color: transparent;
z-index: 50;
```

TIPO: CSS

INGRANDIRE UN'IMMAGINE AL PASSAGGIO DEL MOUSE

```
position: absolute;
background-color: none;
padding: 5px;
left: -1000px;
border: none;
visibility: hidden;
color: black;
text-decoration: none;
```

Se volete aggiungere un colore di contorno all'immagine in grande dovrete cambiare questo attributo `background-color: none;` e mettere al posto di `none` il colore scelto [meglio se esadecimale].

Se Se volete aggiungere un contorno all'immagine in grande dovrete cambiare questo attributo `border: none;` e mettere al posto di `none` la larghezza del bordo [es: 1px; 2px; 3px; ecc]

TIPO: CSS

MARGIN

margin

E' una proprietà a sintassi abbreviata. Con essa è possibile specificare i valori per tutti e quattro i lati di un elemento. Si applica a tutti gli elementi e non è ereditata.

Valori

- * un valore numerico con unità di misura.
- * un valore in percentuale.
- * auto. Per la proprietà margin il valore auto significa che la distanza sarà automaticamente calcolata rispetto alla larghezza dell'elemento contenitore.

Per usare al meglio questa proprietà è fondamentale conoscere le regole che ne gestiscono il funzionamento.

In prima istanza è ovvio usare per essa quattro valori, uno per ciascun lato:

```
div {margin: 10px 15px 10px 20px;}
```

L'ordine di lettura va inteso in senso orario. Per cui: il primo valore si riferisce al lato superiore, il secondo a quello destro, il terzo al lato inferiore, il quarto a quello sinistro. In pratica usare la sintassi vista nell'esempio equivale a scrivere:

```
div {
margin-top: 10px;
margin-right: 15px;
margin-bottom: 10px;
margin-left: 20px; }
```

Nella definizione dei valori, inoltre, è possibile mischiare percentuali con valori assoluti in unità di misura.

Un'ulteriore abbreviazione della sintassi si può ottenere usando tre, due o un solo valore. Queste le regole:

- * se si usano tre valori, il primo si riferisce al margine superiore, il secondo a quelli sinistro e

destra, il terzo a quello inferiore

* se si usano due valori, il primo si riferisce ai lati superiore e inferiore, il secondo al sinistro e al destro

* se si usa un solo valore, un uguale distanza sarà ai quattro lati

Un uso interessante del valore auto per i lati sinistro e destro è quello che consente di centrare in tal modo un elemento rispetto alla pagina o al box contenitore

Esempi

```
p {margin: 20px 10px;}
div {margin: 20px;}
h1 {margin: 10px auto;}
```

TIPO: CSS

OMBRA SU BOX

```
<style type="text/css">
.shadow1{
box-shadow: 5px 5px #818181;
-webkit-box-shadow: 5px 5px #818181;
-moz-box-shadow: 5px 5px #818181;
filter: progid:DXImageTransform.Microsoft.dropShadow(color=#818181, offX=5, offY=5,
positive=true);
}
</style>
```

```
<div class="shadow1" style="width:500px; height:50px; padding:5px; background:navy; border:
1px solid black;"> </div>
```

TIPO: CSS

OMBREGGIATURA SUL TESTO

text-shadow

Crea un'ombreggiatura per il testo specificato. La proprietà non è supportata da nessun browser e non è ereditata.

Sintassi

```
selettore {text-shadow: <valore colore> <valore x> <valore y> <valore blur>;}
```

Esempi

```
h1 {text shadow: blue 2px 2px 3px;}
```

TIPO: CSS

POSITION

1. position: static;

Il valore static (statico) è il valore di default e sarebbe uguale alla posizione dell'elemento nell'esempio del flusso normale.

L'elemento 'statico' non può essere mosso dalla sua posizione e ignora le dichiarazioni top, bottom, left e right.

2. position: absolute;

Il valore absolute (assoluto) permette di rimuovere l'elemento dallo flusso normale della pagina per poterlo posizionare con le dichiarazioni top, bottom, left e right in unità di misura.

L'elemento assoluto è posizionato in rapporto al suo div 'contenitore' solo se lui stesso ha la sua

proprietà position impostato con un valore altro che 'static'.

Se no, l'elemento in posizione assoluta si posiziona in base a l'elemento HTML (normalmente, l'area visiva della finestra del browser) sempre in alto a sinistra.

TIPO: CSS

POSIZIONARE UN BOX AL CENTRO

```
...<body>
<div class="centrato">
  Testo contenuto nel box.
</div>
</body>
```

Come puoi vedere utilizziamo solamente un semplice elemento DIV il cui contenuto può essere totalmente arbitrario... ad esempio un form di login.

Per il nostro DIV abbiamo utilizzato la classe CSS "centrato" che andiamo a definire di seguito. Vorrei farti notare che sarebbe stato possibile utilizzare anche in ID... cambia solamente il tipo di sintassi utilizzata nella seguente porzione CSS.

Il foglio di stile CSS è altrettanto semplice:

```
div.centrato {
  display:
  block;
  width:300px;
  height:200px;
  position:absolute;
  top:50%;
  left:50%;
  margin:-150px 0 0 -100px;
  background: #900;
}
```

Le prime 3 righe definiscono la dimensione del box. Questo è un dato fondamentale affinché il successivo posizionamento abbia successo.

Le successive 3 righe definiscono il posizionamento assoluto del box. Utilizzando la misura percentuale riusciamo ad ottenere il posizionamento centrato senza conoscere le dimensioni esatte della finestra.

Ti faccio però notare che in questo modo è l'angolo in alto a sinistra del box a risultare centrato. Questo si deve al fatto che i posizionamenti vengono calcolati proprio da questa posizione per qualsiasi elemento della pagina.

Entrando nel vivo di questa tecnica di posizionamento utilizziamo l'attributo margin per "correggere" il posizionamento portando il punto notevole al posizionamento al centro del box. E' importante notare che utilizziamo dei margini negativi per "arretrare" di metà dimensione sia la larghezza che l'altezza del box.

L'ultima riga serve semplicemente a definire un colore di sfondo per visualizzare meglio il box posizionato.

Visualizza l'esempio completo »

Conclusioni:

Questa è una semplicissima tecnica per ottenere un box centrato basata unicamente su definizioni CSS. Sicuramente è una più che valida alternativa all'utilizzo di tabelle per ottenere effetti di questo tipo.

Il punto debole di questa soluzione è la necessità di conoscere a priori le dimensioni esatte del box da centrare.

Per finire vorrei farti notare che questa soluzione può essere adottata anche per posizionare immagini o altri elementi XHTML.

TIPO: CSS

SFONDI SFUMATI CON IMMAGINE

```
background: url(grad_white.png) repeat-x;
```

TIPO: CSS

SFONDO

```
BODY {  
font-family: Arial, Verdana, Geneva, Helvetica, sans-serif;  
text-align: left;  
background-color: #C0CDDE;  
font-variant: normal;  
font-size: 11px;  
background-image: url(bkhtm.jpg);  
background-repeat: no-repeat;  
background-position: 300px 0px;  
  
}
```

TIPO: CSS

SPAZIO TRA LE LETTERE

letter-spacing

Aumenta lo spazio tra le lettere di una parola. Proprietà ben supportata ed ereditata.

Sintassi

```
selettore { letter-spacing: <valore>; }
```

Valori

* normal. Valore di default. Le lettere mantengono il loro spazio normale.

* un valore numerico con unità di misura. Le lettere saranno spaziate secondo la distanza impostata.

E' possibile anche impostare valori negativi. Ciò farà sì che le lettere appaiano sempre più compresse, con il rischio ovvio dell'illeggibilità.

Esempi

```
h1 { letter-spacing: 5px; }
```

TIPO: CSS

SPAZIO TRA LE PAROLE

word-spacing

Proprietà complementare a letter-spacing. Serve ad aumentare lo spazio tra le parole comprese in un elemento. Proprietà ben supportata ed ereditata.

Sintassi

```
selettore { word-spacing: <valore>; }
```

Valori

* normal. Valore di default. Le parole mantengono il loro spazio normale.

* un valore numerico con unità di misura. Le parole saranno spaziate secondo la distanza impostata.

Esempi

```
p { word-spacing: 1.2em; }  
div.testo { word-spacing: 5px; }
```

TIPO: CSS

TRASFORMARE IL TESTO DA MIN A MAI

text-transform

La proprietà serve a cambiare gli attributi del testo relativamente a tre aspetti: maiuscolo, minuscolo, prima lettera maiuscola. Proprietà ben supportata e comodissima se avete lunghe porzioni di testo tutto in maiuscolo da trasformare (o viceversa). Ereditata.

Sintassi

```
selettore {text-transform: <valore>;}
```

Valori

- * none. Valore di default. Nessuna trasformazione viene applicata.
- * capitalize. La prima lettera di ogni parola viene trasformata in maiuscolo.
- * uppercase. Tutto il testo è maiuscolo.
- * lowercase. Tutto il testo è minuscolo.

Esempi

```
p {text-transform: capitalize;}  
h1 {text-transform: uppercase;}
```