

Copyright (c) 2008, 2009, 2010 Fabio Proietti

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Authors and contributors
Fabio Proietti

Feedback
Please direct any comments or suggestions about this document to
fabio.proietti AT istruzione DOT it

Publication date and version
2008-11-15, v.0.4
2009-11-15, v.0.5
2010-04-21, v.0.6

last modified 2012-02-11

Indice

Introduzione

Attributo style

Elemento <style>

Elemento <link>

Proprietà dello sfondo

Proprietà del testo

Collegamenti

Box model

Flusso del testo

Proprietà position

Proprietà float

Selettore class

Selettore id

Elementi e <div>

Introduzione

Quando si usa il linguaggio HTML o XHTML per costruire pagine web, non ci si cura dell'aspetto estetico delle pagine, ma solo dei contenuti.

Per personalizzare l'aspetto esteriore della pagina, come è il colore dello sfondo o l'allineamento del testo, esiste uno specifico linguaggio chiamato CSS.

Il CSS, come il linguaggio HTML, viene interpretato dal browser.

CSS significa Cascade Style Sheet (fogli di stile a cascata) e permette di controllare in modo organico, coerente, semplice ed economico l'aspetto delle pagine.

Il linguaggio CSS non è l'unico modo di controllare l'aspetto estetico di una pagina ma chi non lo utilizza, fatica di più ed impiega più tempo.

Il CSS, come il linguaggio HTML, ha subito una evoluzione ed è giunto oggi alla versione 3.

Nei seguenti paragrafi si vedranno alcuni esempi per comprendere dove si può inserire il codice CSS. Solo successivamente si analizzerà la sintassi del linguaggio e le sue caratteristiche.

Attributo style (Inline style)

Ci sono alcuni attributi standard che possono essere utilizzati su più tag.

Uno di questi è l'attributo **title**, che può essere usato sia su `` che su `<a>`.

Ad esempio:

```

```

```
<a href="file.mp3" title="scarica il file mp3">clicca</a>
```

L'attributo `title="..."` non va confuso con il tag `<title>`, che invece si trova racchiuso entro `<head>`.

Un altro attributo standard è `style="..."`

Questo attributo consente di definire delle regole di stile, ad esempio:

```
<h1 style="color:blue; text-align:center">Capitolo 1</h1>
```

```
<p style="color:green">Primo paragrafo.</p>
```

Tra virgolette e separate da punto e virgola, ci sono le regole di stile per h1 e p.

Quando le regole di stile sono messe all'interno di un tag sono dette anche "inline styles".

Ogni regola è composta da due elementi (proprietà e valore) separati dal simbolo "due punti", inoltre, come detto, la regola termina con un punto e virgola.

proprietà:valore;

Ad esempio questa regola imposta il colore di un testo in blu:

color:blue;

Inline style: questo modo di inserire le regole è scomodo perché:

- si deve ripetere la stessa regola su tutti i paragrafi
- si deve ripetere la regola su tutte le pagine di un sito web.

Elemento <style> (Internal style sheet)

Oltre all'attributo `style="..."` esiste anche il tag `<style>` che si usa dentro `<head>` e che va usato come nel seguente esempio:

```
<head>
  <style type="text/css" >
    h1 {color:blue; text-align:center;}
    p {color:green;}
  </style>
</head>
```

Si ricorda che tutto ciò che si trova dentro gli elementi `<head>` di solito non viene visualizzato dal browser.

Come si può notare, ogni regola è espressa nella seguente forma:

```
selettore { proprietà:valore;... }
```

In questo caso le regole sono dette anche "Internal style sheet".

Internal style sheet: questo modo di inserire le regole è un po' più comodo perché:

- non si deve ripetere la stessa regola su tutti i paragrafi.
- però è ancora necessario ripetere la regola su tutte le pagine di un sito web.

Elemento <link> (External style sheet)

All'interno del tag `<head>` è possibile inserire collegamenti ad altri documenti esterni, come file `.css` (fogli di stile) usando il tag `<link>`.



Tutto quello specificato dentro `<head>`, compreso il tag `<link>`, non viene visualizzato dentro la pagina web. Nel prossimo esempio è possibile leggere il codice html necessario a collegare `pagina.html` con `foglio.css`

```
<head>
  <meta http-equiv="Content-Type" content="text/html" />
  <link rel="stylesheet" type="text/css" href="./foglio.css" />
</head>
```

attributo **rel**: specifica la relazione tra la pagina web e il documento esterno

attributo **type**: specifica il tipo di contenuto (MIME type) del documento esterno

attributo **href**: specifica il percorso del documento esterno

In questo caso le regole vengono chiamate anche "External style sheet". A causa del fatto che si hanno due file di testo, bisogna preoccuparsi della codifica di entrambi i file. Per specificare la codifica del `foglio.css` si deve inserire:

```
@charset "UTF-8";
```

External style sheet: questo modo di inserire le regole è molto più comodo perché:

- non si deve ripetere la stessa regola su tutti i paragrafi.
- non si deve ripetere la stessa regola su tutte le pagine di un sito web.

Conclusione

Riassumendo ci sono tre modi di controllare lo stile di un elemento (tag) HTML:

- inline
- internal
- external

Questi modi possono essere usati anche tutti e tre contemporaneamente, ma in generale valgono i seguenti principi:

- il primo si usa per controllare lo stile di un solo elemento nella pagina
- il secondo si usa per controllare lo stile di più elementi simili nella stessa pagina
- il terzo si usa per controllare lo stile di più elementi in diverse pagine

Nei prossimi esempi si utilizzeranno esclusivamente fogli di stili esterni

Specificare un colore

Si è visto un esempio in cui si modifica il colore del testo, con la seguente regola

```
p { color:green; }
```

Un altro qualsiasi colore può essere ottenuto sommando tre componenti di colore: rosso, verde e blu (red, green, blue)

L'intensità di ogni colore si esprime con un numero in base dieci (da 0 a 255) oppure in base sedici (da 00 a ff)

```
p { color:rgb(10,119,255); }
```

```
body { color:#0a77ff; }
```

Proprietà dello sfondo

La seguente regola consente di impostare nell'elemento **<body>**, un'immagine di sfondo (albero.jpg) allineata in alto a destra, sopra uno sfondo celeste (#ccccff):

```
body {  
    background:#ccccff url('albero.jpg') no-repeat right top;  
}
```

In alternativa possono essere utilizzate le seguenti singole proprietà:

```
body {  
    background-color: #ccccff;  
    background-image: url('albero.jpg');  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Proprietà	Valori permessi	
background-attachment	scroll fixed inherit	
background-color	<i>VALORE COLORE</i> transparent inherit	
background-image	url(<i>VALORE URL</i>) none inherit	
background-position	left top left center left bottom right top right center right bottom	center top center center center bottom <i>x% y%</i> VALOREX VALOREY inherit
background-repeat	repeat repeat-x repeat-y no-repeat inherit	

Proprietà del testo

Il seguente gruppo di regole consente di impostare nell'elemento `<p>`, un testo di colore blu (`#0000ff`), allineato a destra e sottolineato

```
p {
  color: #0000ff;
  text-align: right;
  text-decoration: underline;
}
```

Proprietà	Valori permessi	
color	<i>VALORE COLORE</i>	
letter-spacing	normal	<i>VALORE IN PIXEL</i>
text-align	left right	center justify
text-decoration	none underline	overline line-through blink
text-indent	<i>VALORE IN PIXEL</i>	<i>VALORE IN %</i>
text-shadow	<i>VALORE COLORE</i>	<i>VALORE IN PIXEL</i>
text-transform	none capitalize	uppercase lowercase
vertical-align	baseline sub super top text-top	middle bottom text-bottom <i>VALORE IN PIXEL</i> <i>VALORE IN %</i>
white-space	normal pre	nowrap

Per **esercizio** provare a colorare di rosso un testo e poi ad allinearlo a destra: è possibile colorare alcune parole nere e altre rosse? È possibile allineare alcune parole a destra e altre parole a sinistra?

Anche il carattere (o font) utilizzato è una proprietà CSS, ma può accadere che il browser non supporti il font richiesto nella regola o che tale font non sia installato nel sistema di un utente che visualizza la pagina. È necessario quindi specificare dei font generici come "Serif", "Sans-Serif" o "Monospace".

Nella seguente tabella sono messi a confronto quattro font:

E	E	E	E
DejaVu Sans	DejaVu Serif	FreeSans	FreeSerif

Nel seguente esempio, la regola richiede il font "freesans" o, in alternativa a questo, il font "Arial" o, in alternativa ai primi, un qualsiasi font della **generica** famiglia "sans-serif"

```
p { font-family: freesans Arial sans-serif; }
```

Quando il programmatore deve scegliere il font da utilizzare nella pagina web, deve tenere conto sia della disponibilità e del supporto del font da parte di chi visualizzerà la pagine, sia della licenza d'uso con sui può essere utilizzato il font. La licenza d'uso del font viene fornita da chi ha creato il font, come la licenza di un'immagine viene fornita da chi ha creato l'immagine.

La seguente regola modifica lo stile del font (utilizzando il corsivo) e il peso del font (utilizzando il grassetto).

```
p {
  font-family: freemono "Courier New" monospace;
  font-style: italic;
  font-weight: bold;
}
```

Proprietà	Valori permessi	
font-family	<i>family-name</i> <i>generic-family</i> inherit	
font-size	xx-small x-small small medium large x-large	xx-large smaller larger VALORE IN PIXEL VALORE IN % inherit
font-style	normal italic	oblique inherit
font-weight	normal bold bolder lighter 100 200	300 400 ... 900 inherit

Le dimensioni dei font

La leggibilità del testo su un monitor è una cosa importante per una pagina web ben riuscita. Tuttavia, specificare le dimensioni del font non è facile perché, specificando questa dimensione in pixel si potrebbe ottenere un effetto visivo diverso a seconda della risoluzione del monitor di chi visualizza la pagina.

Per questo motivo è consigliabile specificare la dimensione dei caratteri in percentuale invece che in pixel, in modo che sia proporzionale alla dimensione predefinita dal browser di ogni utente.

Tuttavia le impostazioni di un utente potrebbero avere un font più grande rispetto a quello che possiamo immaginare e in tal caso il testo potrebbe finire in una posizione inaspettata, inusuale, poco gradevole o scomoda da leggere.

I collegamenti e le pseudoclassi

I collegamenti (o link) possono avere le stesse proprietà viste anche per il testo. I link hanno quattro tipi di selettori (che si chiamano anche pseudoclassi):

a:link collegamento non visitato

a:visited collegamento visitato

a:hover collegamento sotto al puntatore del mouse

a:active collegamento mentre viene premuto il tasto per selezionarlo

in questo esempio i link vengono colorati di rosso e cambiano colore quando si avvicina il puntatore del mouse

NOTA: è importante specificare le quattro pseudoclassi sempre nell'ordine indicato

```
a:link {
    color: #ff0000;
}
a:hover {
    background-color: #00ff00;
}
```

Il selettore id

Come si possono fare due titoli usando due stili diversi?

Una possibilità è quella di usare gli stili inline.

Un'altra possibilità è quella di identificare in modo diverso ogni titolo. Ad esempio, per creare un identificatore si può scrivere:

```
<h1> primo paragrafo</h1>
<h1 id="speciale"> secondo paragrafo</h1>
```

```
h1#speciale
  {   color: red;
    }
```

Per utilizzarli nel codice HTML si aggiunge l'attributo id="..." mentre nel codice CSS si aggiunge il simbolo cancelletto (#) seguito dall'identificatore.

NOTA: Ogni identificatore può essere utilizzato per identificare un SOLO elemento <h1> in tutta la pagina HTML. Non posso riutilizzare la stessa regola per un altro titolo...

Immaginando di applicare un titolo con identificatore "speciale" in tutte le pagine di un sito web. Se al posto dell'identificatore "speciale" si fosse usata una regola inline, come nel seguente esempio, si avrebbe avuto uno svantaggio: perché, quando si vuole cambiare il colore da rosso a verde, si deve modificare il codice html di tutte le pagine del sito.

```
<h1> primo paragrafo</h1>
<h1 style="color:red;"> secondo paragrafo</h1>
```

Il selettore class

L'identificatore assicura che ci sia un solo elemento con tale nome, ma alcune volte si vuole riutilizzare la stessa regola per più elementi. Come si può ripetere la stessa regola su più elementi?

Una possibilità sarebbe quella di usare gli stili inline, con tutti i suoi svantaggi.

Un'altra possibilità è quella di usare due classi diverse per i paragrafi.

Ad esempio, per creare due classi si può scrivere:

```
<p class="blu"> primo paragrafo</p>
<p class="verde"> secondo paragrafo</p>
<p class="blu"> terzo paragrafo</p>
<p class="verde"> quarto paragrafo</p>
```

```
p.blu
  {   color: blue;  }

p.verde
  {   color: green; }
```

Per utilizzarle nel codice HTML si usa l'attributo class="..." mentre in CSS si usa il punto (.) seguito dal nome della classe.

NOTA: a differenza degli identificatori, una classe può essere utilizzata anche più volte dentro la stessa pagina web.

È possibile definire più classi sullo stesso elemento, ma con proprietà diverse (come colore ed allineamento) e combinarle insieme.

```
<p class="blu destra"> la prima frase</p>
<p class="blu sinistra"> la seconda frase</p>
<p class="rosso destra"> la terza frase </p>
<p class="rosso sinistra"> la quarta frase </p>
```

```
p.blu
  {   color: blue;   }

p.rosso
  {   color: red;    }

p.destra
  {   text-align:right;  }

p.sinistra
  {   text-align:left;  }
```

Una pseudoclasse con classe

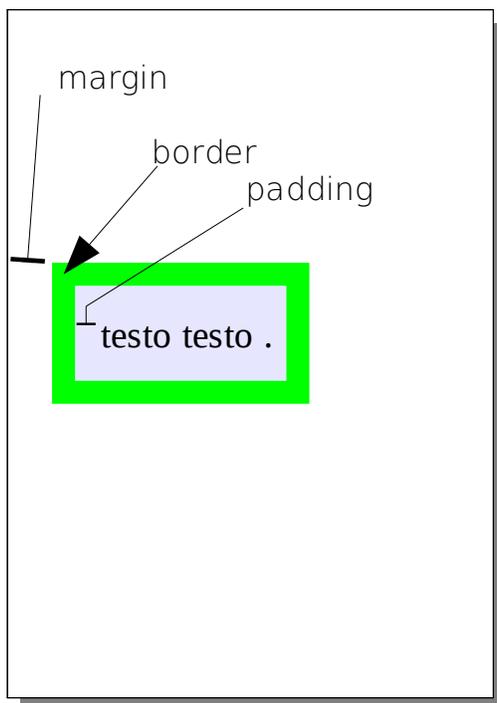
Nel caso in cui si voglia usare una classe con le pseudoclassi dei collegamenti, si deve indicare prima la classe e poi la pseudoclasse

```
a.interno:link
  {
    color: #ff0000;
  }
```

```
a.esterno:link
  {
    color: #00ff00;
  }
```

Box Model

Ogni elemento HTML può essere considerato come un contenitore a cui si possono assegnare caratteristiche diverse (come il bordo, il colore o la posizione).



Proprietà	Valori permessi	
margin	<i>auto</i>	<i>VALORE IN PIXEL</i> <i>VALORE IN %</i>
border-style	none hidden dotted dashed solid	double groove ridge inset outset inherit
border-color	<i>VALORE COLORE</i> transparent	inherit
border-width	thin medium	thick <i>VALORE IN PIXEL</i> inherit
padding	<i>VALORE IN PIXEL</i> <i>VALORE IN %</i>	

Ognuna delle precedenti proprietà (margin, border e padding) può specificare un solo valore per tutti i 4 lati del box, oppure 4 valori diversi per i 4 lati, nel seguente ordine: superiore, destro, basso e sinistro.

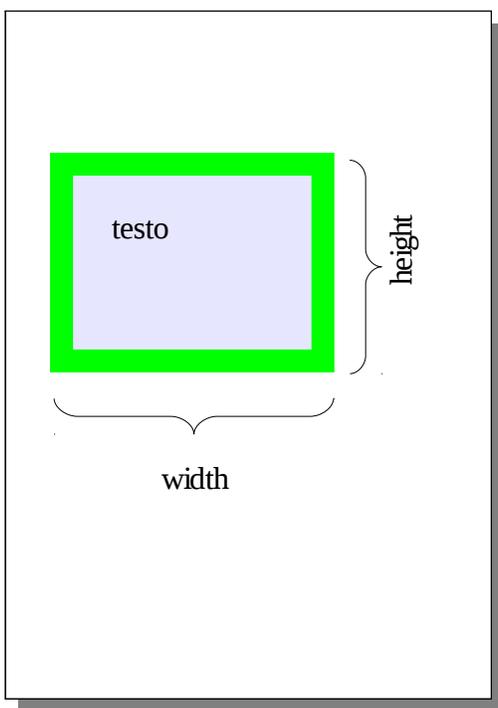
Nel seguente esempio viene creato un contenitore che ha un bordo più grande sul lato superiore.

```
p {
  border-style:solid;
  border-color: black;
  border-width: 2 1 1 1;
}
```

Nel seguente esempio viene creato un contenitore con un margine di 3 pixel, un bordo rosso di 3 pixel e uno sfondo rosa.

```
p {
  margin: 5px;
  border-style: solid;
  border-color: red;
  border-width: 3px;
  background-color: pink;
}
```

Ogni box ha di solito una dimensione che si adatta al suo contenuto. Esistono comunque delle regole che, oltre al margine e al bordo, possono specificare anche delle dimensioni maggiori (o minori) di quelle che avrebbe avuto automaticamente



Proprietà	Valori permessi	
width / height	<i>auto</i> <i>inherit</i>	VALORE IN PIXEL VALORE IN %

Esercizio

creare una tabella come quelle indicate nelle seguenti figure, utilizzando le classi in css in fogli di stile esterni

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

soluzione

prima	seconda	terza	quarta
a	b	c	d
a	b	c	d
a	b	c	d
a	b	c	d

soluzione

Il flusso del testo

Il flusso normale del testo scorre normalmente nella pagina web da sinistra verso destra e dall'alto verso il basso. Esistono due tipi di elementi:

- gli elementi "inline" (ad esempio `<a>` e ``) che si affiancano a sinistra e vanno a capo solo quando è necessario;
- gli elementi "block" (come `<p>` e `<h1>`) che interrompono il flusso del testo e vanno sempre a capo.

La proprietà `display` consente di modificare questa caratteristica.

Esercizio

Provare ad utilizzare su due immagini `display:inline`, `display:block` e `display:none`

Proprietà	Valori permessi	
display	none block inline	table table-row inherit
visibility	visibile	hidden

Insieme a `display`, esiste una proprietà leggermente diversa: `visibility`

parola con  immagine	display: inline; visibility: hidden;	valori predefiniti per le immagini
parola con  immagine	visibility: hidden;	rende trasparente (cioè nascosto o invisibile) l'elemento a cui è applicata.
parola con  immagine	display: none;	rimuove completamente dalla pagina web l'elemento a cui viene applicata.

Le proprietà "**position**" e "**float**" permettono di modificare il flusso del testo, modificando il metodo di posizionamento di un elemento, rispetto al bordo della pagina o rispetto ad un elemento HTML padre (contenitore).

Proprietà	Valori permessi	
position	<i>static</i> <i>absolute</i>	<i>fixed</i> <i>relative</i> <i>inherit</i>
float	left right	none inherit

Position non indica "dove" mettere l'elemento, ma solo il sistema da adottare. La differenza tra "**position**" e "**float**" si nota quando due utenti osservano la stessa pagina utilizzando una finestra del browser con dimensioni diverse oppure modificando la dimensione della finestra del browser.

- Utilizzando la proprietà "**position**" l'elemento può essere messo ovunque, anche sopra altri elementi, addirittura nascondendoli.
- Usando la proprietà "**float**" gli altri oggetti possono scorrere liberamente a destra (o a sinistra). Ad esempio, la proprietà "**float**" permette al testo di scorrere ordinatamente intorno ad un'immagine.

Le due proprietà potrebbero essere usate anche contemporaneamente, ma nel caso in cui venga specificata la proprietà **position: absolute** qualsiasi valore assegnato a **float** verrà ignorato.

Proprietà position

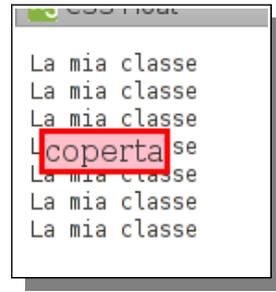
Prima di vedere i valori che può assumere la proprietà **position**, è necessario conoscere altre quattro proprietà che si usano INSIEME a position. Infatti, dopo aver specificato il valore di position, si deve specificare il valore della distanza dai margini della pagina web. Se si indica la distanza dal margine superiore, ovviamente, non si può specificare anche quella dal margine inferiore: top è alternativa a bottom, e right è alternativa a left.

Proprietà	Valori permessi	
top / left / right / bottom	<i>auto</i> <i>inherit</i>	VALORE IN PIXEL VALORE IN %

Valori position	Descrizione
static	Default. Non specifica la posizione e l'elemento appare nel normale flusso.
absolute	Fissa la posizione <u>rispetto</u> al contenitore superiore (padre) che <u>non sia static</u> . Il valore deve essere specificato nelle proprietà: "left", "top", "right" e "bottom".
fixed	Fissa la posizione rispetto alla pagina web. Il valore è specificato come sopra in "left", "top", ecc.
relative	Fissa la posizione <u>rispetto</u> alla posizione normale (static) dell'elemento. Il valore è specificato come sopra.
inherit	Eredita la proprietà position dall'elemento contenitore (padre)

Nel seguente esempio si usa il valore "fixed" per `<p>` per poterlo poi disporre vicino alla parte bassa rispetto all'elemento `<body>`, nel seguente modo:

```
p {  
  position: fixed;  
  bottom: 50px  
  right: 50px  
  margin: 5px;  
  border: 3px solid red;  
  background-color: pink;  
}
```



Esercizio

inserire un box con bordo rosso sulla parte destra in basso della pagina e al suo interno un testo che sia allineato a destra.

Proprietà float

Un elemento "float" si posiziona adiacente agli elementi successivi, ma non "sopra" come accade usando la proprietà "position"

NOTA: un elemento "float" non influenza gli elementi precedenti

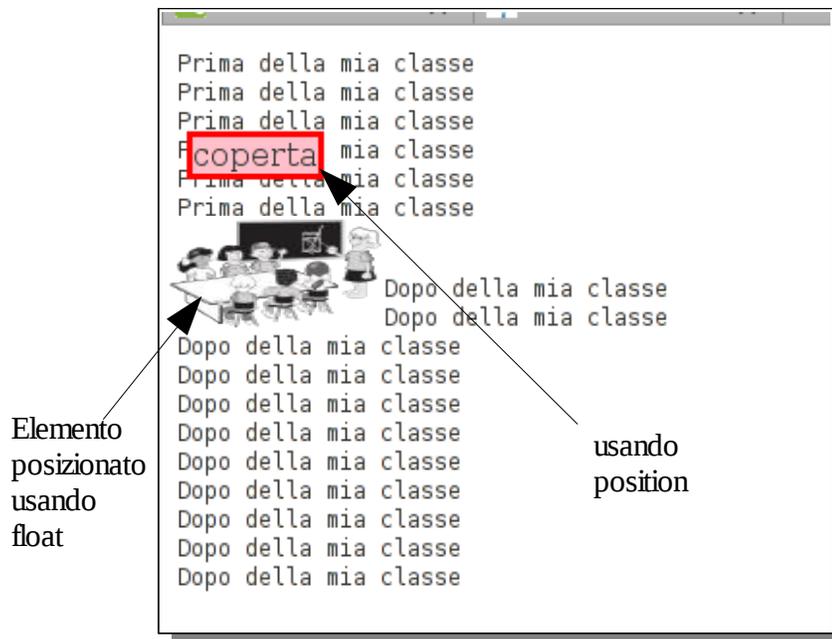
Valori float	Descrizione
left	l'elemento si posiziona a sinistra e tutto quello che lo segue gli scorre intorno (a destra)
right	l'elemento si posiziona a destra e tutto quello che lo segue gli scorre intorno (a sinistra)

Per il prossimo esempio è stato usato il seguente codice

```
<p> Prima della mia classe </p>
<p> Prima della mia classe </p>

<p> Dopo la mia classe </p>
<p> Dopo la mia classe </p>
```

```
img {
  float:left
}
```



La precedente regola posiziona l'elemento (un'immagine) allineandolo completamente a sinistra. Come sarebbe stato il risultato senza usare float? Di solito gli altri elementi scorrono dalla parte opposta (a destra).

Per "interrompere" il flusso del testo a fianco di un oggetto float, come il precedente, si può applicare su alcuni elementi, la regola **clear:left;** (o **clear:right;**)

Un elemento che possiede la proprietà clear, si disporrà subito **sotto** un eventuale elemento float.

Sintassi dei selettori

In CSS, all'inizio di ogni regola può essere specificato l'elemento, o tag, a cui verrà applicata. Ad esempio, la seguente regola verrà applicata agli elementi **<p>**:

```
p {
  text-decoration:underline;
}
```

La seguente regola verrà applicata sia agli elementi **<h1>** che **<p>**

```
h1, p {
  text-decoration:underline;
}
```

La seguente regola verrà applicata solo agli elementi **** interni a **<p>** mentre non a quelli che sono interni ad altri elementi (come nella seconda riga di codice HTML , dove **** è interno all'elemento **<a>**)

```
p > img{
  background-color:yellow;
}
```

codice HTML **<p> </p>**

codice HTML **<p> <a> </p>**

La seguente regola verrà applicata a tutti gli elementi **** che si trovano all'interno di elementi **<p>**

```
p img{
  background-color:yellow;
}
```

codice HTML **<p> </p>**

codice HTML **<p> <a> </p>**

Esempio

Esempio che fa apparire e scomparire un'immagine nascosta

```
img.nascosta
```

```
{  
  display:none;  
}
```

```
a.sorpresa
```

```
{  
  border-style: solid;  
  width:20;  
  height:20;  
}
```

```
a.sorpresa:hover img.nascosta
```

```
{  
  display: inline  
}
```

Questo esempio permette di osservare una combinazione di tutte le proprietà CSS viste fino ad ora per creare un menu a tendina in css (senza usare programmi in javascript...)

```
ul.mainmenu, ul.mainmenu ul {  
    width: 85px;  
    border-width: 1px;  
    border-style: solid;  
    border-color: black;  
    background-color: lightyellow;  
}
```

```
ul.mainmenu li {  
    position: relative;  
}
```

```
ul.mainmenu li:hover {  
    background-color: yellow;  
}
```

```
ul.mainmenu li > ul {  
    display: none;  
}
```

```
ul.mainmenu li:hover > ul {  
    display: block;  
    /*  
    position: absolute;  
    top: 2px;  
    left: 80px;  
    */  
}
```

Elementi `` e `<div>`

Se si vuole attribuire una proprietà ad un gruppo di oggetti o ad un testo che non è racchiuso da nessuno elemento HTML, può essere usato il tag `` che viene usato esclusivamente per aggiungere stili.

Quando si vuole interrompere anche il flusso del testo, invece di `` si usa `<div>`. Questo perché `` è un elemento di tipo "inline", come `` e `<a>`, mentre `<div>` è un elemento di tipo "block", come `<p>` e `<table>`.

Ad esempio, si può racchiudere una parola di un paragrafo:

```
<p>La mia classe mi piace <span>molto</span> </p>
```

Ovviamente, per poter usare diversi stili per `` è necessario usare il selettore class (o il selettore id):

```
<p>La mia classe mi piace <span class="grassetto">molto</span> </p>
```

