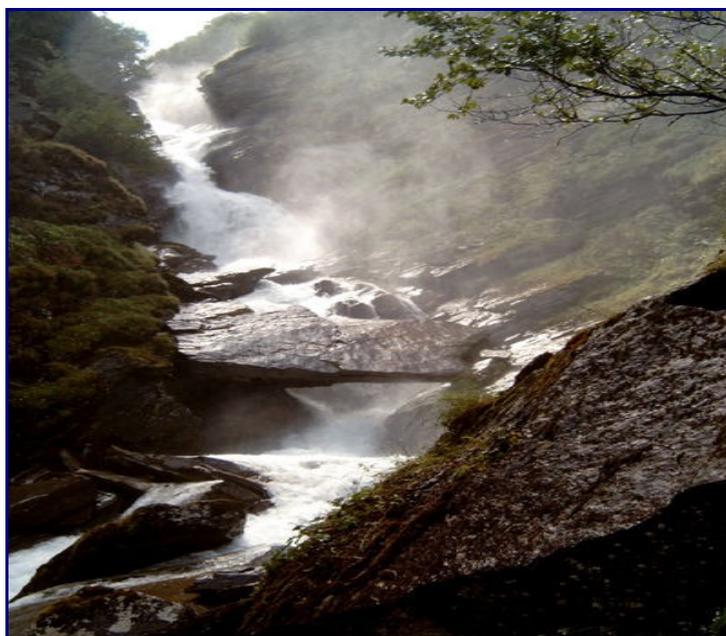


Benvenuto nel wikibook:

# CSS



[Versione attuale del libro>>](#)

**© Copyright 2007, utenti di Wikibooks. Questo libro è stato pubblicato dagli utenti di Wikibooks.**

GFDL 2007

E' permesso copiare, distribuire e/o modificare questo documento secondo i termini della GNU Free Documentation License, Versione 1.2 o qualsiasi versione successiva pubblicata dalla Free Software Foundation, senza le Sezioni Invarianti costituite dalla prima di copertina e dal paragrafo "Licenza". Una copia della licenza è contenuta nella sezione intitolata "Licenza".

**Autori:** Diablo, Otrebla86, Pietrodn, Ramac, Sky.

**Copertina:** Cascade, si tratta infatti di fogli di stile a cascata, ed è rilasciata secondo i termini della GFDL.

## Indice generale

Premessa.....	3
Finalità.....	3
Libri correlati.....	3
Altri progetti.....	3
Collegamenti esterni.....	4

Regole e sintassi.....	4
CSS esterni e interni.....	4
Stili esterni.....	4
Stili interni.....	4
Style.....	5
Sintassi fondamentale.....	5
Commenti.....	5
Selettori.....	6
Elementi HTML.....	6
ID e CLASSI.....	6
Le pseudo-classi.....	7
Gli pseudo-elementi.....	7
Altri tipi di selettori.....	7
Cascade.....	8
L'origine di un foglio di stile.....	8
La specificità di un selettore.....	8
!important.....	9
Attribuire uno stile ad un elemento.....	9
Unità di misura, ereditarietà e box model.....	9
Definire i valori.....	9
Numeri interi e reali.....	9
Grandezze.....	10
Percentuale.....	10
URI.....	10
Stringhe.....	10
Colori.....	11
Ereditarietà.....	11
Il box model.....	12
Proprietà di base.....	12
Sfondo.....	12
Colore.....	12
background-image.....	13
background-repeat.....	13
background-attachment.....	13
Posizione.....	14
Riepilogo.....	14
Testo base.....	14
color.....	14
letter-spacing e word-spacing.....	15
text-align.....	15
text-decoration.....	16
text-indent.....	16
text-transform.....	16
Testo avanzato.....	16
font-family.....	17
font-size.....	17
font-weight.....	17
font-style.....	18
font-variant.....	18
La proprietà riassuntiva font.....	18
Liste.....	18
list-style-type.....	19

list-style-image.....	19
list-style-position.....	19
list-style.....	20
Disegnare la struttura della pagina.....	20
Gestione delle dimensioni e dell'overflow.....	20
I margini.....	21
Il padding.....	21
I bordi.....	21
Proprietà riassuntive dei bordi.....	22
Posizionamento.....	22
Posizionarsi relativamente agli altri elementi della pagina.....	23
Posizionamento assoluto e fisso.....	23
Usare il floating.....	23
La proprietà clear.....	24
Contenuto generato.....	25
content.....	25
Uso delle virgolette.....	26
Lavorare con i contatori.....	26
Altre proprietà.....	27
cursor.....	28
display.....	28
Compatibilità.....	29
L'affidabilità dei diversi browser.....	29
Soluzioni.....	29
Licenza.....	30

## Premessa

Il linguaggio [CSS](#) è un linguaggio che permette di impostare tutti gli aspetti di una visualizzazione di una pagina web, quali allineamento della pagina, il layout (la disposizione cioè dei contenuti rispetto allo schermo), la formattazione del testo e i colori.

Esso è nato dalla necessità di separare il contenuto HTML della pagina dalla grafica e dalla visualizzazione. Le regole per la buona stesura di un codice CSS sono state dettate dal [W3C](#).

## Finalità

Questo wikibook ha come obiettivo quello di far apprendere al lettore buone conoscenze del linguaggio CSS. Il libro può essere consultato da chi possiede un fondamento di [HTML](#), essendo il CSS un linguaggio di applicazione grafica alla struttura preesistente.

## Libri correlati

- [HTML](#)

## Altri progetti

-  [Wikipedia](#) contiene una voce riguardante [CSS](#)

## Collegamenti esterni

- [css.html.it](http://css.html.it)
- ([EN](#)) [la sezione dedicata al CSS del World Wide Web Consortium](#)

# Regole e sintassi

## CSS esterni e interni

Gli stili di una pagina possono essere definiti sia all'interno del file nel quale devono operare, sia in un file a parte.

- Gli **stili esterni** solitamente vengono usati se si hanno molte pagine a cui applicare il medesimo stile. Questo permette di non fare confusione nel lavoro di modifica e revisione.
- Gli **stili interni** vengono usati se le pagine sui cui operare sono veramente poche o se il sito è semplice (questo sistema è sconsigliato perché nasconde una delle potenzialità dei CSS)

Si noti che i CSS vengono validati dal W3C solamente se sono esterni.

### Stili esterni

È il sistema più usato ed è compatibile con tutti i browser. Per inserire un collegamento ad uno stile esterno alla pagina, usiamo il tag `<link />` nell'intestazione della nostra pagina:

```
<html>
<head>
  <title>Inserire i fogli di stile in un documento</title>
  <link rel="stylesheet" type="text/css" href="stile.css">
</head>
<body>...
```

Il seguente sistema di importazione è invece più adatto a risolvere problemi di compatibilità tra vecchi e nuovi browser:

```
<style>
  @import url(stile.css);
</style>
```

### Stili interni

```
<html>
<head>
  <title>Inserire i fogli di stile in un documento</title>
  <style type="text/css">
    ...codice...
  </style>
</head>
<body>...
```

Si ricordi che questo sistema non permette la validazione da parte del [W3C Validator](#)

## Style

L'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'attributo `style` applicabile ad ogni elemento HTML. Questo sistema è utilizzato all'interno degli elementi del codice HTML della pagina e può risultare utile se c'è bisogno di aggiungere uno stile unico per l'elemento.

```
<elemento style="regole_di_stile">
```

Se ad esempio si vuole formattare l'elemento `Fieldset`, bisogna operare come segue:

```
<fieldset style="color:red;font-size:14px">
```

In questo caso questo elemento, solo questo nella pagina, avrà il testo al suo interno rosso e grande 14px

## Sintassi fondamentale

Un foglio CSS è semplicemente un insieme di definizioni dello stile da applicare a determinati elementi della pagina. La loro sintassi è:

```
selettore { dichiarazioni }
```

Le dichiarazioni tra parentesi graffe racchiudono coppie di *proprietà* : *valore* separate da un punto e virgola. Ad esempio:

```
<style>
  p { font-size: 12px; }
</style>
```

è un foglio di stile con una sintassi valida. In particolare, il valore per la proprietà *font-size* sarà *12px*.

## Commenti

All'interno di un foglio di stile è possibile inserire dei commenti, ossia porzioni di testo che verranno ignorate dal browser, delimitati da `/*` e `*/`, che si possono estendere su una o più righe. Questo può risultare utile soprattutto nei casi di collaborazione a coloro che non hanno scritto di prima mano il codice; può però risultare utile anche all'autore stesso, ad esempio per poter riprendere il proprio codice tra le mani dopo un lungo periodo di tempo; inoltre può risultare comodo per eliminare temporaneamente porzioni di codice durante la fase di test.

```
selettore {
  /* questo verrà ignorato*/
}

/* anche_questo_selettore {
  e_questa: proprietà
} */

/* *attenzione*: mai annidare i commenti
/* il browser farà come se questo testo non esistesse */ <-- qui si chiude il
commento
  ma questo testo verrà letto
```

```
*/ <-- questo genera un errore
```

# Selettori

I CSS possono contenere diverse indicazioni sullo stile dei vari elementi della pagina, i quali devono essere identificati dai **selettori**. I selettori possono riferirsi a:

- elementi HTML
- ID
- classi

Esistono oltre a questi anche particolari elementi o classi, chiamate pseudo-classi e pseudo-elementi.

## Elementi HTML

È il caso più semplice di selettore, che definisce la visualizzazione di tutti gli elementi HTML con un determinato nome. Ad esempio:

```
p { dichiarazioni }
```

definisce lo stile per tutti gli elementi p (i paragrafi) della pagina.

## ID e CLASSI

Un id è un selettore CSS che può essere assegnato ad un solo elemento nella pagina, identificato univocamente dall'attributo `id`. Una classe è un insieme di proprietà che può essere invece assegnato a uno o più elementi HTML tramite l'attributo `class`. La sintassi è:

```
#nome_id { dichiarazioni }  
.nome_classe { dichiarazioni }
```

e sono richiamate in questo modo:

```
<elemento_html id="nome_id">...</elemento_html>  
<elemento_html class="nome_classe">...</elemento_html>
```

Per quanto riguarda le classi, è possibile stabilire dichiarazioni diverse a seconda essa sia attribuita ad un elemento piuttosto che ad un'altra:

```
p.evidenziato { dichiarazioni }  
h1.evidenziato { dichiarazioni }
```

imposta uno stile diverso a seconda che la classe `evidenziato` sia attribuita a elementi H1 e elementi P. È possibile inoltre stabilire sottoclassi, la cui sintassi è

```
elemento.classe.sottoclasse { dichiarazioni }
```

Nell'esempio precedente avremmo potuto quindi aggiungere:

```
p.evidenziato.grassetto { dichiarazioni }
```

e, nella pagina HTML:

```
<p class="evidenziato grassetto">...</p>
```

## Le pseudo-classi

Esistono particolari classi che identificano le proprietà di un selettore in base al proprio stato, chiamate **pseudo-classi**. La sintassi è

```
elemento:stato { dichiarazioni }
```

Il loro uso è legato soprattutto ai tag `<a>`; gli stati definibili per i link sono:

- `link` indica lo stato di un collegamento non ancora visitato
- `visited` indica lo stato di un collegamento già visitato
- `hover` indica lo stato di un collegamento quando vi passa sopra il mouse. Una volta che il mouse esce dal link, viene ripreso lo stile definito da `link` o da `visited`
- `active` indica lo stato di un collegamento quando l'utente fa clic su di esso

Un altro stato definibile è `focus`, utilizzato soprattutto per i campi dei form, che identifica lo stato di un elemento che ha ottenuto il focus da parte dell'utente. Una ulteriore pseudo-classe è associata alla lingua dell'utente. Ad esempio:

```
elemento:lang(it) //definizione per utenti italiani  
elemento:lang(en) //definizione per utenti anglosassoni
```

## Gli pseudo-elementi

Gli **pseudo-elementi** identificano elementi specifici ma che non sono identificati da tag o attributi. La sintassi è:

```
elemento:nome_pseudoelemento { dichiarazioni }
```

I principali pseudo-elementi sono:

- `first-child` identifica il primo elemento figlio contenuto nell'elemento
- `first-letter` identifica la prima lettera del testo contenuto nell'elemento
- `first-line` identifica la prima riga del testo contenuto nell'elemento
- `begin` e `end` usati in relazione con la proprietà `content`

## Altri tipi di selettori

Basandosi sulla gerarchia degli elementi HTML nella pagina è possibile definire alcuni stili particolari. In questi casi è però meglio che la pagina sia strutturata secondo gli standard [XHTML](#).

- **elementi figli:** è possibile definire uno stile per tutti gli elementi contenuti in un altro. La sintassi è:

```
elemento_genitore elemento_figlio { dichiarazioni }
```

Ad esempio:

```
h1 b { dichiarazioni }
```

identifica tutti gli elementi `b` contenuto anche non direttamente in un elemento `h1`

- elementi fratelli: il primo elemento allo stesso livello di un altro che abbia quindi lo stesso legame di parentela. La sintassi è

```
elemento1 + elemento2 { dichiarazioni }
```

e identifica il primo elemento1 fratello di un elemento2

## Cascade

Per *cascade* si intendono le regole con le quali il browser assegna ad un elemento una dichiarazione di stile. Questo è importante nel caso di selettori diversi che indichino lo stesso elemento fornendo dichiarazioni contrastanti.

Nel caso di fogli di stile importati, vengono considerate gli stili del foglio che li importa.

## L'origine di un foglio di stile

Prima di analizzare il *cascade*, è bene vedere le possibili origini di un foglio di stile:

- **autore**: è lo stile definito nel sorgente HTML della pagina dal webmaster o dal grafico del sito
- **browser**: i browser conformi agli standard del World Wide Web Consortium devono avere un foglio di stile predefinito da assegnare alle pagine
- **utente**: taluni browser mettono a disposizione all'utente la possibilità di creare un foglio di stile personalizzato, magari tramite un'interfaccia grafica e traducendo poi le impostazioni dell'utente in dichiarazioni CSS

È possibile quindi che questi fogli di stile si sovrappongano e in questo caso vengono applicati in base al *cascade*.

## La specificità di un selettore

Altro elemento importante è la **specificità** di un selettore. Vediamo come calcolarla:

- calcolare il numero di attributi ID nel selettore (= a)
- calcolare il numero degli altri attributi (es. classi) e pseudoclassi del selettore (= b)
- calcolare il numero dei nomi degli elementi HTML (= c)

Concatenare ora i tre numeri nell'ordine  $a + b + c$ . Si faccia attenzione a scegliere una base di numerazione opportuna: se, ad esempio,  $b$  vale 12, si può scegliere una base esadecimale in modo tale che il numero risulti, ad esempio, 0-c-7 invece che 0-12-7.

Si noti che l'attributo `style` degli elementi HTML possono essere considerati selettori ID, in

quanti hanno un riferimento univoco con il tag a cui si riferiscono.

## **!important**

Ultima regola da vedere prima di analizzare le regole del cascade è la dichiarazione ***!important*** la quale indica al browser che una regola è particolarmente importante. Vediamo la sintassi:

```
selettore { [dichiarazioni...]  
proprietà: valore !important;  
[dichiarazioni...]  
}
```

## **Attribuire uno stile ad un elemento**

Il CSS assegna un peso a ciascun blocco di dichiarazioni di stile; nel caso di sovrapposizione di dichiarazioni, vince quella con maggior peso.

Analizziamo finalmente le regole seguite dal browser per attribuire uno stile ad un elemento HTML nella pagina:

- dapprima cerca tutte le dichiarazioni identificate dal selettore che corrisponde all'elemento in questione
- la prima analisi riguarda il peso e l'origine degli stili. Per quanto riguarda le normali dichiarazioni, hanno la precedenza, in ordine, gli stili dell'autore, dell'utente e del browser. Nel caso di dichiarazioni seguite da *!important*, invece, hanno la precedenza, in ordine, gli stili dell'utente, dell'autore e del browser; questo per permettere la piena accessibilità dei contenuti da parte di utenti con handicap visivi, ad esempio. Le dichiarazioni *!important* superano sempre le dichiarazioni normali.
- la seconda distinzione riguarda la specificità del selettore: ha la precedenza il selettore con specificità maggiore
- nel caso due dichiarazioni abbiano stessi peso, specificità e origine vince quella fornita per ultima. Le dichiarazioni dei fogli di stile importati sono considerate come precedenti qualsiasi dichiarazione del foglio che importa.

## **Unità di misura, ereditarietà e box model**

Parleremo in questa pagina di alcune delle proprietà più importanti delle pagine di stile, ovvero le unità di misura, la definizione dei valori, il concetto di ereditarietà e i box.

### **Definire i valori**

In alcuni casi, una proprietà accetta solo valori definiti tra una serie di parole chiave; negli altri casi il CSS accetta diversi di valori.

### **Numeri interi e reali**

I **numeri** devono essere indicati senza separatori per le migliaia, usando un punto (.) come separatore decimale. È possibile inoltre specificare il segno del numero (+ o -) anche se la maggior

parte delle volte non serve in quanto molte proprietà accettano solo valori positivi.

## Grandezze

Le **grandezze** sono usate per definire lunghezze orizzontali e verticali. e sono espresse da un numero (intero o reale) seguito da una unità di misura.

Le principali **unità di misura** si dividono in:

- **relative**: specificano una grandezza che dipende da un'altra grandezza. Sono:
  - **em**: è relativa all'altezza font in uso. Se, ad esempio, il font è alto 12pt, 1em varrà 12pt, 2em varranno 24pt.
  - **ex**: funziona come em, ma è relativa all'altezza della lettera x nel set di caratteri in uso.
  - **px**: [pixel](#), sono relativi al dispositivo di output (solitamente lo schermo) e alle impostazioni del computer dell'utente
- **assolute**: le unità di misura assolute sono equivalenti a quelle usate nella realtà. Sono:
  - **in**: [pollici](#); un pollice vale circa 2.54 centimetri
  - **cm**: [centimetri](#)
  - **mm**: [millimetri](#)
  - **pt**: [punti](#) — un pt per CSS vale 1/72 di pollice
  - **pc**: [pica](#) — 1 pica vale 12 punti

## Percentuale

I valori in **percentuale** servono per esprimere percentuali del valore che assume la proprietà stessa dell'elemento padre e vanno espresse con un numero seguito dal simbolo di percentuale (%)

## URI

I valori [uri](#) specificano un percorso assoluto (es. `http://esempio/dir/file`) oppure uno relativo (nell'esempio di prima, `dir/file`).

I valori URI vanno usati tramite la notazione `url (percorso) . percorso` può essere delimitato da apici (').

## Stringhe

Le [stringhe](#) sono delle sequenze alfanumeriche di caratteri, ovvero del testo.

In CSS vanno delimitate da una coppia di virgolette singole o doppie, finendo con lo stesso apice con cui si è iniziato. Ad esempio:

```
'questa è una stringa'  
"questa anche"  
'questa no (è chiusa in modo scorretto)'
```

Nel caso si voglia inserire in una stringa delimitata, ad esempio, da apici un apice è necessario usare il carattere di commutazione \ (barra retroversa). Ad esempio:

```
'questa non è una 'stringa'  
'questa invece è una \'stringa'
```

## Colori

Un **colore** può essere identificato con tre metodi differenti:

- tramite un numero esadecimale che indica le componenti [RGB](#). Per una lista completa dei colori esadecimali, vedi [la tavolozza dei colori](#)
- tramite la funzione

```
rgb (<rosso>, <verde>, <blu>)
```

dove <rosso>, <verde> e <blu> sono le componenti RGB del colore espresse con un valore intero da 0 a 255.

- tramite una delle parole chiave predefinite, che sono:

•		black
•		silver
•		gray
•		white
•		maroon
•		red
•		purple
•		fuchsia

•		green
•		lime
•		olive
•		yellow
•		navy
•		blue
•		teal
•		aqua

- tramite un colore associato alle proprietà di sistema. Ad esempio:

•	background (il colore di sfondo del desktop)
•	buttonFace (il colore di sfondo dei pulsanti)
•	buttonText (testo dei pulsanti)
•	captionText (testo delle etichette)
•	grayText (testo disabilitato)

## Ereditarietà

Nei fogli di stile l'ereditarietà, come suggerisce la parola, si verifica quando, definito uno stile per un elemento HTML, a tutti i suoi elementi figli (cioè contenuti anche indirettamente in esso) verrà applicato lo stesso stile (si dice che la proprietà viene *ereditata*).

Per esempio, se all'elemento `<body>` applichiamo un colore rosso tramite la dichiarazione `color: red`, tutti i suoi elementi discendenti (quindi tutti gli elementi della pagina) erediteranno questa proprietà, a meno che non venga stabilito diversamente nelle dichiarazioni di altri elementi discendenti del tag `body`. Infatti l'ereditarietà è l'ultima possibilità per attribuire uno stile ad un elemento.

Non tutte le proprietà sono ereditabili: non avrebbe senso, ad esempio, forzare l'ereditarietà delle dimensioni di un elemento o i suoi margini, in quanto sono normalmente diversi per ogni singolo elemento; saranno invece ereditate proprietà che rendano la pagina omogenea, come appunto il colore o le dimensioni del testo.

Per forzare l'ereditarietà di una proprietà per una particolare dichiarazione, è possibile usare il valore speciale **inherit** (eredita), che è un valore valido per qualsiasi proprietà e imposta il valore dell'elemento appena genitore.

## Il box model

Per box model si intende l'insieme delle regole per la definizione degli stile degli elementi blocco. Ogni box comprende alcuni elementi di base:

- un'area del contenuto vero e proprio (il testo, un'immagine, ecc...), di cui è possibile definire l'altezza e la larghezza (**width e height**)
- un **padding**, ossia uno spazio vuoto tra il contenuto e il bordo dell'elemento
- un bordo (**border**), del quale è possibile impostare colore, stile e spessore
- un margine (**margin**) che identifica uno spazio intercorrente tra l'elemento e gli altri elementi presenti nella pagina. Nel caso di due box con margini allineati in verticale, la loro distanza sarà data dalla somma dei due margini. Se sono allineati orizzontalmente, intervergono le regole del *margin collapsing* e la distanza effettiva tra i due elementi sarà pari al valore del margine maggiore.

L'altezza e la larghezza effettive di un elemento saranno quindi date dalla somma delle dimensioni del contenuto orizzonti/verticali (width e height), dal valore del padding orizzontale/verticale e dallo spessore dei bordi orizzontali/verticali.

## Proprietà di base

Nei moduli seguenti verranno descritte le proprietà di base di questo linguaggio. Si parlerà delle proprietà del testo e dello sfondo.

### Sfondo

Grazie ai CSS possiamo facilmente assegnare un colore o un'immagine di sfondo, ripetuta o meno, ad un elemento nel web: dal body al paragrafo, dalla tabella all'input form.

### Colore

La proprietà **background-color** permette di definire il colore di sfondo di un elemento.

#### Sintassi:

```
selettore {background-color: <valore>;}
```

Il valore deve essere un colore. Il colore verrà inoltre usato per riempire gli spazi non coperti da un'eventuale immagine di sfondo.

#### Esempio:

```
body { background-color: white; }  
p { background-color: #FFFFFF; }  
.classel { background-color: rgb(0, 0, 0)}
```

## background-image

La proprietà **background-image** definisce un'eventuale immagine da usare per lo sfondo dell'elemento.

### Sintassi:

```
selettore { background-image: <valore> }
```

dove <valore>:

- un URL assoluto o relativo che punti ad un'immagine (in questo caso bisognerà usare la notazione url (percorso))
- **none** (valore di default): non verrà applicata nessuna immagine allo sfondo

### Esempio:

```
body {background-image: url(immaginesfondo.gif); }  
.class2 {background-image:  
url(<nowiki>http://www.wikibooks.it/immagine.gif</nowiki>); }
```

## background-repeat

Imposta il modo in cui l'immagine viene ripetuta sullo sfondo.

### Sintassi:

```
selettore { background-repeat: <valore>; }
```

I valori possibili sono i seguenti:

- **repeat**: l'immagine di sfondo viene ripetuta sia in senso orizzontale sia in senso verticale
- **repeat-x**: l'immagine di sfondo viene ripetuta in senso orizzontale
- **repeat-y**: l'immagine di sfondo viene ripetuta in senso verticale
- **no-repeat**: l'immagine di sfondo viene visualizzata solo una volta, senza ripetizioni

## background-attachment

Questa funzione consente di scegliere se l'immagine dovrà essere fissa sullo sfondo, o si muove con la pagina.

### Sintassi:

```
selettore { background-attachment: <valore> }
```

dove <valore> può assumere i seguenti valori:

- **scroll** L'immagine si muove con la pagina
- **fixed** L'immagine è fissa sullo sfondo

## Posizione

Questa funzione decide come deve essere posizionata l'immagine all'interno della pagina.

### Sintassi:

```
selettore { background-position: <valore> }
```

dove <valore> può assumere questi valori:

- **top left** in alto a sinistra
- **top center** in alto centrata
- **top right** in alto a destra
- **center left** al centro a sinistra
- **center center** al centro centrata
- **center right** al centro a destra
- **bottom left** in basso a sinistra
- **bottom center** in basso centrata
- **bottom right** in basso a destra
- <valori in percentuale>

Per quanto riguarda i valori in percentuale, essi decidono il valore x e y in percentuale rispetto alla pagina.

### Esempio:

```
selettore { background-position: 50% 50% }
```

L'immagine è così mostrata al centro.

## Riepilogo

Per una maggiore compattezza e semplificazione del codice, possiamo riassumere tutte le proprietà relative allo sfondo scrivendo semplicemente **background** e tutti i 5 valori possibili a seguire.

### Esempio:

```
selettore {  
    background: blue  
               url(immaginedisfondo.gif)  
               no-repeat  
               scroll  
               center center  
}
```

## Testo base

Verranno adesso descritte le proprietà di base del CSS inerenti al testo:

### color

La proprietà **color** specifica il colore del testo contenuto in un qualsiasi elemento HTML, ad

esempio una pagina intera, o una tabella, o il contenuto di un fieldset, etc.

### **Sintassi:**

```
selettore { color: <colore>; }
```

Dove <colore> è un valore di colore valido.

Ad esempio:

```
<style>
  body {color: red;}
  table {color: green;}
</style>
```

Con l'esempio di cui sopra, assegneremo il colore rosso al testo di tutta la pagina, tranne quello contenuto nelle tabelle.

## **letter-spacing e word-spacing**

Definiscono lo spazio rispettivamente tra i caratteri e le parole del testo

### **Sintassi:**

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

Dove <valore> può essere **normal**, che definisce la spaziatura normale tra le lettere o le parole, un valore espresso in percentuale o una grandezza seguita da un'unità di misura valida.

## **text-align**

Definisce l'allineamento del testo.

### **Sintassi:**

```
selettore { text-align: <pos>; }
```

Dove <pos> può assumere i valori :

- **left**
- **center**
- **right**

che allineano il testo rispettivamente a sinistra, al centro e a destra, e

- **justify**

che giustifica il testo ai margini dell'elemento.

## text-decoration

Definisce eventuali decorazioni in aggiunta al testo.

### Sintassi:

```
selettore { text-decoration: <valore> }
```

Dove <valore> può assumere uno dei seguenti valori:

- **none**: al testo non viene applicata nessuna decorazione
- **underline**: il testo viene sottolineato da una riga continua
- **overline**: ogni riga del testo ha una linea continua sopra
- **line-through**: il testo risulta barrato da una linea continua
- **blink**: il testo lampeggia (da colore a trasparente). Non tutti i browser supportano questo valore

## text-indent

Imposta il rientro per la prima riga del testo

### Sintassi:

```
selettore { text-indent: <valore> }
```

Dove <valore> è un valore percentuale o affiancato da un'unità di misura (in questo caso è preferibile usare em) che definisce lo spazio tra il margine dell'elemento e la prima lettera del testo. È possibile assegnare a questa proprietà anche valori negativi.

## text-transform

Permette di forzare la capitalizzazione del testo.

### Sintassi:

```
selettore { text-transform: <cap> }
```

Dove <cap> può assumere uno dei seguenti valori:

- **none**: la capitalizzazione del testo rimane quella definita nel codice HTML
- **capitalize**: forza la prima lettera di ogni parola alla capitalizzazione maiuscola
- **uppercase**: forza il testo ad essere maiuscolo
- **lowercase**: forza il testo alla capitalizzazione minuscola

# Testo avanzato

In questo modulo verranno trattate le proprietà per definire il font di un testo in una pagina, che sono principalmente:

## font-family

Permette di definire il tipo di carattere usato per la visualizzazione del testo.

### Sintassi:

```
selettore { font-size: <valore>; }
```

dove <valore> è una sequenza di uno o più tipi di carattere indicati tra apici nell'ordine in cui verranno usati nel caso non siano installati sul sistema operativo dell'utente. È possibile inoltre definire una famiglia generale di caratteri tramite i seguenti valori:

- **serif**: le [serif](#) sono le grazie, ovvero gli abbellimenti delle lettere, che presentano caratteri come il Times New Roman o il FreeSerif. È usato soprattutto per la stampa, in quanto la lettura delle grazie su schermo risulta più faticosa. Esempio: Testo con serif.
- **sans-serif**: sono [sans-serif](#) (senza grazie) font come il Verdana, l'Arial, il FreeSans, l'Helvetica, ecc... Sono usati prevalentemente per lo schermo. Esempio: **Testo senza serif**.
- **cursive**: i font cursive presentano delle caratteri più simili alla calligrafia umana piuttosto che a quelli di stampa. Esempio: **Cursive**.
- **fantasy**: caratteri che uniscono la visualizzazione del testo ad elementi decorativi. Esempio: Testo fantasioso.
- **monospace**: caratteri tipo telescrivente o macchina da scrivere e quindi monospaziati, come il Courier. Esempio: Testo monospaziato.

Ad esempio possiamo definire così il font di una pagina web:

```
body {font-family: 'Verdana', 'Arial', 'Helvetica', sans-serif; }
```

## font-size

Definisce l'altezza del testo che dipenderà anche dal tipo di carattere usato. Normalmente questa proprietà è definita in em oppure in pt.

### Sintassi:

```
selettore {font-size:<valore>; }
```

## font-weight

Permette di definire il "peso" del testo, ad esempio se vada visualizzato in grassetto o meno.

### Sintassi:

```
selettore {font-weight:<valore>; }
```

Dove *valore* è un valore in centinaia compreso tra 100 e 900 inclusi (900 corrisponderà ad un testo più grassetto di 100) oppure uno tra i seguenti valori:

- normal
- bold
- bolder

- lighter

che definiscono un testo normale, grassetto, grassetto pesante e più "leggero". Normalmente il valore `normal` corrisponde al peso numerico 500.

## font-style

Permette di definire un font in corsivo italico o obliquo.

### Sintassi:

```
selettore {font-style: valore;} }
```

dove `valore` può essere **normal**, **cursive** o **oblique**.

## font-variant

Permette di impostare un testo maiuscoletto.

### Sintassi:

```
selettore {font-variant:valore;} }
```

dove `valore` può assumere il valore **normal** o **small-caps** (maiuscoletto). Normalmente il testo maiuscoletto presenta i caratteri minuscoli come delle riduzioni dei corrispettivi caratteri minuscoli.

## La proprietà riassuntiva font

La proprietà **font** permette di riassumere le proprietà viste in questo modulo.

### Sintassi:

```
selettore {font: <font-style> <font-variant> <font-weight> <font-size> <line-height> <font-family>;}
```

È possibile inoltre usare al posto dei diversi valori `font`, indicare una delle seguenti famiglie usate dal sistema in uso:

- **caption**: il font usato per le etichette ad esempio dei pulsanti
- **icon**: il font usato per le etichette delle icone
- **menu**: il font usato nei menù
- **message-box**: il font usato per le finestre di dialogo
- **small-caption**: il font usato per le etichette dei piccoli controlli
- **status-bar**: il font usato per la barra di stato delle finestre

## Liste

In questo modulo verrà trattata la formattazione delle liste che rispetto a quella fornita dall'HTML è sicuramente più gestibile.

## list-style-type

Questa proprietà permette di stabilire, analogamente all'attributo `type` delle liste HTML, il simbolo usato come punto elenco, sia di una lista ordinata sia di una non ordinata. Può assumere i seguenti valori:

- liste ordinate:
  - **decimal**: numeri decimali, partendo da 1
  - **decimal-leading-zero**: numeri decimali con zeri aggiuntivi (01...99)
  - **lower-roman** e **upper-roman**: numeri romani minuscoli (i, ii, iii, iv, v, ecc...) e maiuscoli (I, II, III, IV, V, ecc...)
  - **lower-latin/lower-alpha** e **upper-latin/upper-alpha**: lettere dell'alfabeto latino minuscole (a, b, c, ... z) e maiuscole (A, B, C, ... Z)
  - **lower-greek**: lettere dell'alfabeto greco minuscole ( $\alpha$ ,  $\beta$ ,  $\gamma$ , ...)
- liste non ordinate: sono disponibili i normali valori **disc**, **square** e **circle**, che funzionano come l'attributo `type` delle liste HTML

Impostando la proprietà al valore **none** la lista apparrà, se non viene specificata un'immagine, senza punto elenco.

## list-style-image

CSS permette di definire come punto elenco un'immagine; nel caso sia stato impostato un valore sia per `list-style-type` sia per questa proprietà, viene mostrata solo l'immagine (se è valida).

**Sintassi:**

```
list-style-image: url(<percorso>);
```

<percorso> è quindi un percorso che identifica un'immagine.

## list-style-position

Questa proprietà permette di impostare la posizione del punto elenco di ogni elemento `li`. Può assumere i seguenti valori:

- **outside**: il punto elenco risulterà esterno al blocco del testo dell'elemento
- **inside**: il punto elenco risulterà interno al blocco del testo dell'elemento

Il concetto risulta forse più chiaro con un esempio:

- La proprietà è settata qui su  
inside
- Anche qui
  
- La proprietà è invece settata qui su  
outside
- Anche qui

## list-style

Esiste inoltre una proprietà riassuntiva `list-style` che riassume la formattazione delle liste. La sintassi è:

```
list-style: <list-style-type> <list-style-position> <list-style-image>;
```

# Disegnare la struttura della pagina

In questo modulo verranno trattate le diverse proprietà legate al box model, già analizzato [precentemente](#) nel corso del libro.

## Gestione delle dimensioni e dell'overflow

Le proprietà **width** e **height** permettono di impostare rispettivamente la larghezza e l'altezza del contenuto del box in questione. Il valore di queste proprietà può essere una grandezza oppure un valore in percentuale riferito all'elemento genitore; è possibile inoltre usare un valore **auto**, che funziona in modo diverso per le due proprietà:

- per quanto riguarda l'altezza, con un valore auto verrà determinata dal contenuto dell'elemento stesso
- per quanto riguarda la larghezza, con un valore auto assume le dimensioni dell'elemento genitore

La gestione delle dimensioni del contenuto di un elemento era tuttavia già possibile in HTML attraverso gli attributi `width` e `height`, anche se non era applicabile a tutti i tag.

Possono risultare utili inoltre le proprietà **min-height**, **max-height**, **min-width** e **max-width** che servono per impostare le dimensioni minime o massime che può assumere l'elemento in questione

La proprietà **overflow** permette di impostare il comportamento che deve avere un elemento nel caso il suo contenuto superi le dimensioni previste dalle proprietà `width` e `height` (questo fenomeno è appunto chiamato con il termine tecnico *overflow*). Può assumere uno tra i seguenti valori:

- **hidden** il contenuto in eccedenza viene ignorato e non viene quindi visualizzato
- **visible** le dimensioni del box vengono riadattate in modo tale da mostrare il contenuto per intero
- **scroll** vengono aggiunte delle barre di scorrimento orizzontali e/o verticali che permettono all'utente di vedere per intero il contenuto semplicemente scorrendolo
- **auto** il browser si comporta secondo le sue impostazioni

## I margini

La gestione dei margini prevede l'uso di quattro proprietà `margin-top`, `margin-bottom`, `margin-left`, `margin-right` che specificano rispettivamente il margine superiore, inferiore, sinistro e destro dell'elemento. Possono assumere una grandezza, un valore in percentuale o **auto** (che normalmente imposta margini nulli). Si noti che i valori per i margini possono essere anche negativi; ad esempio:

```
margin-left: -10px;
```

mostra l'elemento spostato di 10px a sinistra rispetto alla posizione che dovrebbe assumere normalmente.

Esiste inoltre una proprietà riassuntiva `margin` che permette di impostare insieme le quattro proprietà relative ai margini. In base a quanti valori vengano forniti, si comporta in maniere differenti:

- se ne viene indicato solo uno, questo si riferirà a tutti e quattro i margini
- se ne vengono indicati due, il primo si riferisce ai margini orizzontali e il secondo a quelli verticali
- se ne vengono indicati tre, il primo e il terzo sono riferiti rispettivamente ai margini superiore e inferiore, il secondo a quelli verticali
- se ne vengono indicati quattro, il primo si riferirà al margine superiore, il secondo a quello destro, il terzo a quello inferiore e il quarto a quello sinistro

## Il padding

Il padding è gestito tramite quattro proprietà `padding-top`, `padding-bottom`, `padding-left`, `padding-right` che specificano rispettivamente il padding superiore, inferiore, sinistro e destro dell'elemento. Possono assumere una grandezza o un valore in percentuale.

Esiste inoltre una proprietà riassuntiva `padding` che permette di impostare insieme le quattro proprietà relative al padding che si comporta allo stesso modo di `margin`.

## I bordi

I bordi in CSS sono gestibili tramite le proprietà:

```
border-<lato>-style  
border-<lato>-color
```

che specificano rispettivamente lo stile, lo spessore e il colore del lato (top, bottom, left, right) indicato.

Per quanto riguarda lo stile dei bordi, è possibile specificare uno dei seguenti valori:

- **none**: il bordo non viene visualizzato
- **dotted**: una successione di punti
- **dashed**: il bordo è visualizzato con dei trattini
- **solid**: il bordo è una linea continua

- **double**: il bordo è costituito da due linee separate da uno spazio bianco
- **groove** e **ridge**: il bordo risulta incassato o in rilievo tramite un gioco di colori
- **inset** e **outset**: l'intero box risulta incassato o in rilievo tramite un gioco di colori

Per quanto riguarda il colore, è possibile specificare un valore di colore valido oppure il valore **transparent** (bordo trasparente che influisce però sulla visualizzazione della struttura).

Per quanto riguarda lo spessore dei bordi, la proprietà `border-<lato>-width` può assumere i valori:

- **thin** (un bordo sottile)
- **medium** (un bordo normale)
- **thick** (un bordo spesso)
- un valore di grandezza, normalmente espressa in px, che deve essere positiva

## Proprietà riassuntive dei bordi

Anche per la definizione dei bordi esistono delle proprietà riassuntive. Sono:

```
border-style
border-width
border-color
```

Queste tre proprietà specificano lo stile, lo spessore e il colore dei quattro bordi assieme. Possono avere da uno a quattro valori e il loro funzionamento è simile a quello di `margin`.

Esistono inoltre le proprietà:

```
border-<lato>: <border-<lato>-width> <border-<lato>-style> <border-<lato>-color>
```

che definiscono i diversi aspetti dei bordi per un singolo lato.

Infine possiamo utilizzare anche una proprietà la cui sintassi è:

```
border: <border-width> <border-style> <border-color>
```

che specifica insieme i valori delle tre proprietà dei bordi per tutti e quattro i lati.

## Posizionamento

CSS permettono di posizionare all'interno di una pagina oggetti che varino o restino fissi indipendentemente dalla dimensione in cui viene visualizzata la pagina e/o allo scorrere di questa.

Nella pagina, un elemento può essere posizionato secondo tre modi differenti:

- seguendo il normale andamento della pagina previsto dall'HTML
- senza alcun legame con il resto del flusso degli elementi
- allineandosi in modo tale che gli elementi si dispongano al lato (questa tecnica, chiamata in gergo tecnico *floating*, è simile al [funzionamento dell'attributo align delle immagini](#))

## Posizionarsi relativamente agli altri elementi della pagina

Per posizionare un elemento rispetto agli elementi della pagina secondo quanto previsto dal puro HTML, usiamo la proprietà **position** con questi valori:

- **static** (valore di default per quasi tutti gli oggetti HTML)
- **relative**: l'elemento è posizionato relativamente il normale flusso con l'uso di quattro proprietà (left, right, top e bottom) che assumono come valore una grandezza e specificano la distanza (positiva o negativa) dell'elemento rispetto alla posizione che questo dovrebbe occupare normalmente nella pagina. Ad esempio:

```
b {position: relative; top: -5px;}
```

Con questo codice, tutti gli elementi bold appariranno spostati di 5 pixel verso l'alto rispetto alla linea del paragrafo.

In particolare:

- **top** indica la distanza del bordo superiore dell'oggetto e il bordo superiore della pagina
- **bottom** indica la distanza del bordo inferiore dell'oggetto e il bordo inferiore della pagina
- **left** indica la distanza del bordo sinistro dell'oggetto e il bordo sinistro della pagina
- **right** indica la distanza del bordo destro dell'oggetto e il bordo destro della pagina

## Posizionamento assoluto e fisso

Per posizionamento assoluto (dal latino *ab solutum*, sciolto dal resto) o fisso si intende un posizionamento che non segue il flusso degli elementi. È fissato tramite la proprietà `position` con l'uso dei valori:

- **absolute**: l'elemento è posizionato rispetto al suo blocco contenitore (ad esempio la pagina) tramite le proprietà left, right, top e bottom. L'elemento risulterà quindi legato allo scorrere della pagina
- **fixed**: l'elemento è posizionato rispetto alla finestra sempre tramite le quattro proprietà left, right, top e bottom e non è legato dallo scorrimento della pagina. Ad esempio:

```
div.bann {  
  position: fixed;  
  bottom: 0px;  
  right: 0px;  
}
```

mostra l'elemento fisso nell'angolo in basso a destra dello schermo.

## Usare il floating

Il *floating* è una tecnica CSS che, tramite la proprietà **float**, permette agli elementi HTML di allinearsi a destra o a sinistra della pagina (o dell'elemento contenitore). La caratteristica più interessante è però la possibilità che gli elementi della pagina scorrano a destra o a sinistra dell'oggetto in questione. Il funzionamento è analogo a quello della proprietà `align` dell'elemento HTML `img`. Può assumere i valori **left**, **right** e **none**. Vediamo due esempi:

Questo elemento `div` è flottante a sinistra (`float: left;`) e il paragrafo si dispone alla sua destra

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene

non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a restringersi, e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia costiera dall'altra parte; e il ponte, che ivi congiunge le due rive, par che renda ancor più sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda ricomincia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni... (da [I promessi sposi, capitolo primo](#), A. Manzoni)

Vediamo alcuni dettagli su questa tecnica:

Questo elemento `div` **non** è flottante (`float: none;`) e il paragrafo non si dispone alla sua destra

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a restringersi, e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia costiera dall'altra parte; e il ponte, che ivi congiunge le due rive, par che renda ancor più sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda ricomincia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni... (da [I promessi sposi, capitolo primo](#), A. Manzoni)

- il *floating* può essere usato solo su elementi con una larghezza fissa (impostata dalla proprietà `width` oppure, ad esempio per le immagini, dalle dimensioni in pixel del file di origine)
- la proprietà `float` non viene considerata se l'elemento ha un posizionamento assoluto o fisso
- gli elementi flottanti diventano necessariamente elementi *block*.

## La proprietà `clear`

La proprietà `clear` sposta l'elemento a cui viene applicata sotto a tutti gli elementi flottanti dal lato specificato. Può assumere i valori:

- **none**: non viene effettuato alcuno spostamento
- **left** o **right**: l'elemento viene spostato sotto a tutti gli elementi *floatanti* a destra o a sinistra
- **both**: vengono considerati tutti gli elementi flottanti sia a destra sia a sinistra.

Vediamo un esempio:

In questo caso il paragrafo qui a fianco, invece di disporsi a destra del blocco, si dispone allineandosi con il suo lato inferiore: ciò è causato dall'uso della dichiarazione CSS `clear: left`. L'effetto è simile a quello del blocco senza floating ma è stato ottenuto con due modalità differenti. Ovviamente la proprietà `clear` risulta comodo con molti elementi flottanti nella pagina. La proprietà `clear` è usata spesso in relazione all'elemento `br` in questo modo:

Questo elemento `div` è flottante a sinistra (`float: left;`) e il paragrafo si dispone alla sua destra

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene

non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia costiera dall'altra parte; e il ponte, che ivi congiunge le due rive, par che renda ancor più sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda ricomincia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni... (da [I promessi sposi, capitolo primo](#), A. Manzoni)

In questo caso il paragrafo qui a fianco, invece di disporsi a destra del blocco, si dispone allineandosi con il suo lato inferiore: ciò è causato dall'uso della dichiarazione CSS `clear: left`. L'effetto è simile a quello del blocco senza floating ma è stato ottenuto con due modalità differenti. Ovviamente la proprietà `clear` risulta comodo con molti elementi flottanti nella pagina.

La proprietà `clear` è usata spesso in relazione all'elemento `br` in questo modo:

```
<br style="clear: both" />
```

che interrompe il testo effettuando il clear in maniera semplice e pulita.

## Contenuto generato

Una delle grandi funzionalità del CSS consiste nella possibilità di generare automaticamente il contenuto degli elementi della pagina.

Per fare ciò si usa la proprietà **content**, che purtroppo non è supportata sempre correttamente da tutti i browser.

### content

La proprietà **content** imposta il contenuto dell'elemento. Può assumere come valore:

- **none**: non viene generato alcun contenuto
- una combinazione di:
  - una **stringa**
  - un **percorso** che indica una risorsa da inserire (es. un'immagine)
  - un **contatore** (vedi nel prossimo paragrafo)
  - **open-quote** o **close-quote**: virgolette di apertura o di chiusura aumentando di uno il livello di indentazione
  - **no-open-quote** o **no-close-quote**: non mostra virgolette ma aumenta il livello di indentazione

Questa proprietà è usata soprattutto con le pseudoclassi `:before` e `:after`, ad esempio in questo modo:

```
body:after {  
  content: 'The end';  
}
```

```
}
```

imposta il contenuto finale del corpo della pagina. Le pseudoclassi `:before` e `:after` non sono però supportate correttamente da IE.

## Uso delle virgolette

Per specificare i diversi tipi di virgolette da usare per i diversi livelli di indentazione usiamo la proprietà **quotes** la cui sintassi è:

```
quotes: [apertura1 chiusura1] [apertura2 chiusura2] [aperturan chiusuran]
```

Per spiegare l'uso facciamo un esempio:

```
/* questo è lo stile */
p { quotes: '<<' '>>' '"' '>'; }
blockquote:before { content: open-quote; }
blockquote:after { content: close-quote; }

<!-- nell'HTML -->
<p>Luigi disse:
  <blockquote>
    Lorem ipsum dolor sit amet, ...<br/>
    Infine disse:
      <blockquote>
        Questo è un altro discorso
      </blockquote>
    E finì
  </blockquote>
  Questo è quello che disse Luigi
</p>
```

fornisce come output

Luigi disse:

```
<< Lorem ipsum dolor sit amet, ...
Infine disse:
```

```
"Questo è un altro discorso"
```

```
E finì >>
```

Questo è quello che disse Luigi

## Lavorare con i contatori

CSS permette inoltre di generare automaticamente contenuti numerati. Per fare ciò è possibile usare due proprietà:

```
counter-reset: <nome1> <valore1> <nome2> <valore2> <nomen> <valoren>
counter-increment: <nome1> <valore1> <nome2> <valore2> <nomen> <valoren>
```

La prima proprietà resetta il contatore chiamato `<nomen>` al valore `<valoren>`. Se omissa il

valore, il contatore viene impostato a 0. La seconda proprietà incrementa il valore del contatore chiamato `<nome>` di `<valore>` posizioni. Se omissso il valore, il contatore viene incrementato di 1.

Per accedere al valore corrente del contatore si usa la funzione

```
counter(<nome>, <stile>)
```

dove `nome` è il nome del contatore e `<stile>` è lo stile del contatore (funziona in maniera identica alla proprietà `list-style-type`).

Vediamo un esempio per chiarire il concetto:

```
/*nel foglio di stile*/
body { counter-reset: capitolo; }
h1:before {
  content: "Capitolo " counter(capitolo) ". ";
  counter-increment: capitolo; /* incrementa il capitolo*/
  counter-reset: sezione; /* resetta la sezione */
}
h2:before {
  content: "Sezione " counter(capitolo) "." counter(sezione) ". ";
  counter-increment: sezione; /* incrementa la sezione */
}

<!-- nel codice html -->
<h1>La pagina</h1>
<p>...</p>
<h2>L'intestazione</h2>
<p>...</p>
<h2>Il corpo</h2>
<p>...</p>
<h1>La formattazione del testo</h1>
<p>...</p>
<h2>I tag di formattazione</h2>
<p>...</p>
<h2>Il tag font</h2>
<p>...</p>
```

produrrà come output

```
Capitolo 1. La pagina
...
Sezione 1.1. L'intestazione
...
Sezione 1.2. Il corpo
...
Capitolo 2. La formattazione del testo
...
Sezione 2.1. I tag di formattazione
...
Sezione 2.2. Il tag font
...
```

## Altre proprietà

In questo modulo analizzeremo alcune proprietà CSS che non sono state presentate nel corso del

libro ma sono comunque rilevanti.

## cursor

Questa proprietà serve a modificare l'aspetto del cursore quando questo si posiziona sull'elemento in questione. Può assumere come valore una lista di:

- **auto**: il cursore automatico per il tipo di elemento
- **default**: il cursore predefinito di sistema (normalmente una freccetta)
- **pointer**: il puntatore usato ad esempio per i link (normalmente la manina)
- **text**: il cursore usato per il testo selezionabile (solitamente il caret I)
- **wait**: indica di aspettare, solitamente è una clessidra
- **progress**: l'applicazione sta lavorando, ma può comunque reagire ai comandi (solitamente una freccia con accanto una clessidra)
- **crosshair**: la croce usata per puntare con precisione (a forma di "+")
- **help**: usato per indicare la possibilità di visualizzare un aiuto relativo all'elemento in questione (spesso un punto interrogativo)
- **move**: il cursore visualizzato su elementi spostabili
- **e-resize**, **ne-resize**, **nw-resize**, **n-resize**, **se-resize**, **sw-resize**, **s-resize**, **w-resize**: le frecce usate per i ridimensionamenti. Le lettere che precedono `-resize` indicano il verso (N = nord (north), S = sud (south), E = est (east), W = ovest (west))
- un percorso indicante un file di cursore

Il browser scorrerà la lista di valori fino a quando non incontrerà un cursore valido o rappresentabile. Altrimenti userà il cursore automatico.

## display

La proprietà **display** serve a definire la visualizzazione e il *rendering* dell'elemento da parte del browser. Può assumere, tra gli altri, i valori:

- **inline**: l'elemento viene visualizzato come in linea
- **block**: l'elemento è visualizzato come blocco
- **list-item**: l'elemento viene visualizzato come un oggetto di una lista ("formattabile" tramite le proprietà `list-style`)
- **none**: l'elemento non viene visualizzato e non influisce sulla visualizzazione della pagina

Questa proprietà può servire:

- nell'ambito di pagine [XML](#), i quali elementi non hanno associato un *rendering*
- ad esempio per poter posizionare e gestire come un blocco (quindi impostare dimensioni, posizioni, ecc...) elementi in-line come i link

E' invece molto interessante l'uso della dichiarazione `display: none;` che, soprattutto se utilizzata insieme a linguaggi dinamici come [JavaScript](#), permette di nascondere o visualizzare elementi della pagina con un semplice clic dell'utente.

# Compatibilità

Come pure molte altre tecnologie usate per la programmazione web, anche il linguaggio CSS soffre di problemi di compatibilità tra i diversi browser sul mercato.

Infatti, sebbene ormai sia possibile adottare soluzioni accettabili e compatibili, persistono ancora nei diversi browser differenze con le specifiche ufficiali del WWW Consortium.

I problemi sono legati soprattutto:

- alla compatibilità tra versioni precedenti dei browser (**retro-compatibilità**)
- alle differenze di *rendering* (visualizzazione) tra i diversi browser

## L'affidabilità dei diversi browser

Attualmente nessuno dei browser in commercio offre il supporto completo agli standard W3C, tuttavia molti vi si avvicinano:

- [Internet Explorer](#), il browser attualmente più usato, è stato forse uno dei maggiori freni alla diffusione del CSS, a causa della scarsa attinenza agli standard. Presenta numerosi problemi di rendering (ad esempio il valore **medium** della proprietà **font-size** è interpretato in maniera scorretta) e, maggiore lacuna, non offre alcun supporto al contenuto generato (proprietà **content**, **quotes**, **counter**, ecc...).
- presentano un ottimo supporto agli standard, anche se non completo, [Firefox](#), [Opera](#), [Safari](#) e [Konqueror](#). In particolare gli ultimi due presentano un parziale supporto ai CSS3.

## Soluzioni

Una delle tecniche più usate è il cosiddetto *browser-sniffing* (*annusamento* del browser) tramite JavaScript, che consiste nella determinazione del browser in uso dal client e poter usare di volta in volta un foglio di stile differente.

Internet Explorer mette inoltre a disposizione nell'ambito dell'HTML il *commento condizionale*, una tecnica che consiste nell'inserimento nella pagina di un commento HTML distinto dalla particolare sintassi:

```
<!--[if IE]>
  codice esclusivamente per Internet Explorer
<![endif]-->
```

In questo modo i browser normali non interpreteranno il commento mentre solo IE leggerà il codice al suo interno. In questo modo è possibile definire un foglio di stile unicamente per IE:

```
<!--[if IE]>
  <link rel="stylesheet" type="text/css" href="stileIE.css">
<![endif]-->
```

L'ultimo suggerimento è quello di testare sempre le proprie pagine su diversi [motori di rendering](#); è infatti oramai possibile sviluppare, come già detto, soluzioni accettabili e compatibili con più browser.

# Licenza

[GNU Free Documentation License](#)

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is

available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque

copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

### [How to use this License for your documents](#)

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document

under the terms of the GNU Free Documentation License, Version 1.2

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".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the

Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two

alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.