

Introduzione ai CSS

I Cascade Style Sheets (CSS), noti in Italia come "Fogli di stile a cascata", sono lo strumento definito dal W3C per definire l'aspetto delle pagine web.

Prima dell'introduzione dei CSS, le opportunità per i webmaster di definire l'aspetto delle pagine web erano limitate a tag HTML (ad esempio il tag FONT) e si basavano sull'uso di tabelle per la definizione del layout. Il limite alla diffusione dei CSS è stato, inizialmente, lo scarso supporto dei CSS da parte dei browser più diffusi. A distanza di anni, i browser hanno finalmente maturato una buona capacità di interpretare i CSS e il limite al loro utilizzo è stato esclusivamente, la pigrizia e l'incapacità dei webmaster, come pure la non conoscenza dei vantaggi offerti dai CSS.

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
- ~ Gli **STILI IN LINEA** consistono nell'uso dell'attributo **style** applicabile a ogni elemento HTML. Questo sistema può risultare utile se c'è bisogno di aggiungere uno stile unico per l'elemento.

Nei primi due casi le regole devono essere specificate come segue:

```
selettore { proprietà: valore }
```

Se per uno stesso selettore si volessero specificare diverse proprietà, ogni proprietà dovrà essere separata da un punto e virgola:

```
selettore { proprietà1: valore1; proprietà2: valore2 }
```

Per maggiore comodità è possibile andare a capo.

Spesso si ha la necessità di associare ad uno stesso tag HTML diversi stili, per fare ciò è possibile ricorrere alle classi.

```
p.nota { font-size: small }
```

```
p.evidenzia { background-color: yellow; }
```

Le due classi sono **nota** ed **evidenzia** e vengono specificate separando il nome del tag e il nome della classe con un punto. La classe verrà specificata nel codice HTML attraverso l'attributo **CLASS**:

```
<p class="nota">Questo paragrafo è una nota</p>
```

```
<p class="evidenziato">Questo paragrafo è evidenziato in giallo</p>
```

Proprietà di base

I fogli di stile sono essenzialmente un **insieme di regole** per istruire il browser su come debba essere presentata la pagina web.

Le **regole** dei CSS possono essere specificate in una pagina esterna al documento a cui le regole devono essere applicate, possono essere specificate nel documento stesso, possono essere incorporate nei tag (X)HTML.

Per ogni elemento, possono essere definite diverse **proprietà**. Ogni proprietà può avere un **valore** che, in associazione con la proprietà, descrive come il browser debba rappresentare il selettore.

In alcuni casi, una proprietà accetta solo valori definiti tra una serie di parole chiave; in altri casi le **grandezze** 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.

- ~ **em**: è relativa all'altezza font in uso. Se, ad esempio, il font è alto 12pt, 1em varrà 12pt, 2em varranno 24pt.
- ~ **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.

- ~ **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 (%)

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, e può essere identificato con tre metodi differenti:

- ~ tramite un **numero esadecimale** che indica le componenti **RGB**
- ~ 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 (ad esempio green, lime, olive, yellow)

LETTER-SPACING E WORD-SPACING

Definiscono lo spazio rispettivamente tra i caratteri e le parole del testo; il valore associato può essere **normal**, che definisce la spaziatura normale, un valore espresso in percentuale o una grandezza seguita da un'unità di misura valida.

TEXT-ALIGN

Definisce l'allineamento del testo e 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; 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 attraverso un valore percentuale o numerico (in questo caso è preferibile usare em) che definisce lo spazio tra il margine dell'elemento e la prima lettera del testo.

E' possibile assegnare a questa proprietà anche valori negativi.

TEXT-TRANSFORM

Permette di forzare la capitalizzazione del testo. 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

FONT-FAMILY

Permette di definire il tipo di carattere usato per la visualizzazione del testo; è possibile utilizzare 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. E' 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. E' usato soprattutto per la stampa, in quanto la lettura delle grazie su schermo risulta più faticosa
- ~ **sans-serif**: sono sans-serif (senza grazie) font come il Verdana, l'Arial, il FreeSans, l'Helvetica, ecc...
- ~ **cursive**: i font cursive presentano delle caratteri più simili alla calligrafia umana piuttosto che a quelli di stampa
- ~ **fantasy**: caratteri che uniscono la visualizzazione del testo ad elementi decorativi
- ~ **monospace**: caratteri tipo telescrivente o macchina da scrivere e quindi monospaziati, come il Courier

FONT-SIZE

Definisce l'altezza del testo e dipende anche dal tipo di carattere usato. Normalmente tale proprietà è espressa in em, px o pt.

FONT-WEIGHT

Permette di definire il "peso" del testo, ad esempio se vada visualizzato in grassetto o meno; il valore può essere espresso 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; il valore può essere **normal**, **cursive** o **oblique**.

FONT-VARIANT

Permette di impostare un testo maiuscoletto; il valore può assumere, tra gli altri, il valore **normal** o **small-caps** (maiuscoletto). Normalmente il testo maiuscoletto presenta i caratteri minuscoli come delle riduzioni dei corrispettivi caratteri minuscoli.

Liste

LIST-STYLE-TYPE

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 e maiuscole
- ~ **lower-greek**: lettere dell'alfabeto greco minuscole (α , β , γ , ...)

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 apparirà, se non viene specificata un'immagine, senza punto elenco.

LIST-STYLE-IMAGE

- ~ permette di definire come punto elenco un'immagine; a tal proposito la sintassi prevede l'utilizzo della proprietà **url** seguita dal

percorso che identifica un'immagine.

LIST-STYLE-POSITION

Questa proprietà permette di impostare la posizione del punto elenco di ogni item della lista. 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

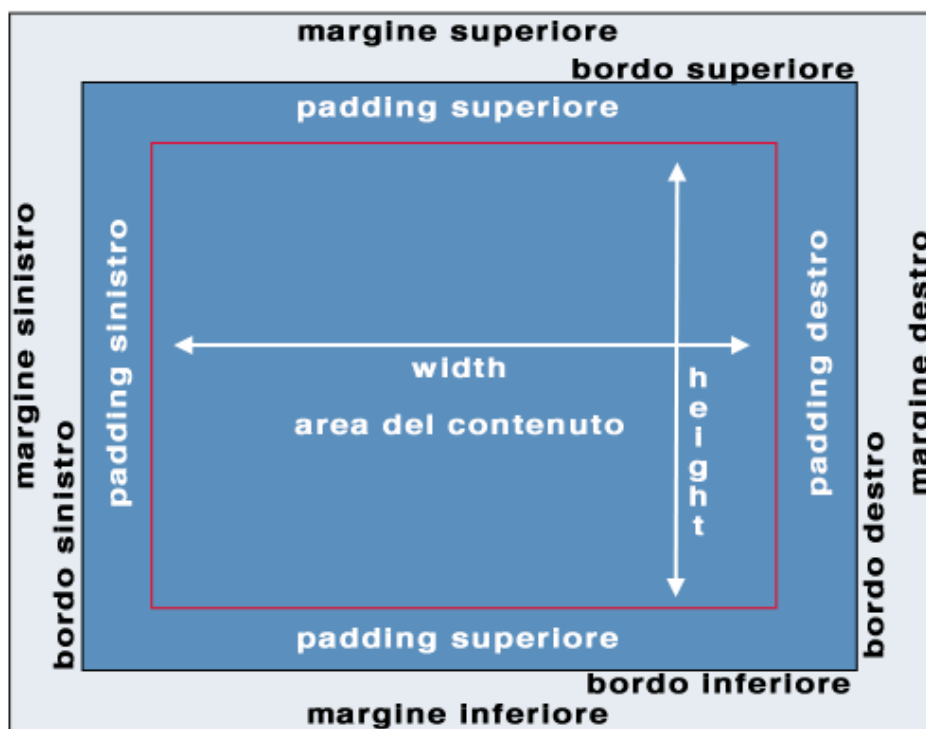
LIST-STYLE

E' una proprietà riassuntiva che riassume la formattazione delle liste.

Disegnare la struttura della pagina

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 in esame 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, intervengono 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.

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.

Esiste inoltre una proprietà riassuntiva **MARGIN** che permette di impostare insieme le quattro proprietà relative ai margini. In base a quanti valori vengono 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

Posizionamento

I 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)

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 al 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;}` indica che 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

Per posizionamento assoluto 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 proprie 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.

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.

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 pseudoelementi.

Elementi HTML

E' 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 }](#)

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. E' 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**. 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