

Introduzione al Cascading Style Sheet language

a cura di Paolo Malesci - www.paolomalesci.it

Lo scopo del CSS language è quello di controllare lo stile dei vari elementi che concorrono a formare un documento HTML.

Si può operare in tre modi:

1. collegamento ad un foglio di stile esterno;
2. definizione degli stili all'inizio del documento;
3. utilizzo della proprietà **style** all'interno di un Tag HTML (embedded style).

Occorre tenere presente che presupposto essenziale per il corretto funzionamento del CSS è la mancanza di errori nei Tag HTML. Prestare quindi attenzione alla chiusura dei vari Tag ed all'ordine in cui avviene; tenere inoltre presente che il tag **<p>** viene interrotto dalla presenza al suo interno di altre strutture come ad esempio **<div>**.

1. Link ad un Foglio di Stile esterno

Inserire nella sezione HEAD del documento la riga di codice:

```
<link rel="stylesheet" href="miofoglio.css">
```

Le definizioni degli stili saranno contenute nel file *miofoglio.css* che dovremo preoccuparci di creare con le modalità che vedremo nella parte finale.

Operando in questo modo si possono avere dei vantaggi quale, ad esempio, l'utilizzo dello stesso stile per più pagine di un sito Web oppure di abbinare Javascript e css per testare le caratteristiche del computer dell'utente ed in base a quelle scegliere fogli di stile diversi.

2. Utilizzo del Tag **<STYLE>**

Inserire nella sezione HEAD del documento:

```
<style type="text/css">
  elem1 { attrib1:valore1;[attrib2:valore2;....] }
  ripetere per ogni elemento da definire
</style>
```

qualora si scriva un elemento su più righe interrompere la riga solo dopo il ;.

3. Elementi ed Attributi

Entriamo nel dettaglio:

elemento è ogni Tag HTML al quale vogliamo attribuire **elementi di stile**; si possono indicare anche più elementi separati da virgole. Se vogliamo che gli elementi H1-H4 siano con il testo rosso anziché nero (default) scriveremo:

```
h1, h2, h3, h4 { color: red; }
```

Abbiamo anche visto un attributo (color) che si scrive (ed ha anche lo stesso significato) come la proprietà color di HTML.

Gli spazi dopo la punteggiatura non sono obbligatori, ma si usano per una migliore leggibilità. Oltre ai vari Tag HTML vi sono anche 3 elementi particolari corrispondenti alle proprietà LINK, VLINK e ALINK dell'elemento <BODY>; **:link**, **:visited** ed **:active** ed in più **:hover** (passaggio del mouse sul link). È inoltre possibile definire **Classi** a piacere antepoendo # al nome ed utilizzando nei Tag **class="nomeclasse"**.

È possibile attribuire più classi ad una parte del documento utilizzando il Tag ; ad esempio:

```
<div class="note">testo a piacere stile 1
<span class="rosso">testo a piacere stile 1 rosso
</span>testo a piacere stile 1
</div>
```

Attributi del testo

Corrispondono alle proprietà del tag di HTML più quelle di controllo della formattazione:

color	colore del testo	vedi nota sull'uso dei colori
font-family	tipo di font	uno o più nomi separati da virgole
font-weight	spessore del font	lighter, normal, bold, bolder (default=normal)
font-style	stile del font	normal, italic, oblique (default=normal)
font-size	dimensione del font	in varie unità di misura
text-align	allineamento del testo	left, right, center, justify
text-indent	rientro 1° rigo in un blocco testo	in varie unità di misura
text-decoration	decorazione del testo	none, underline, blink, line-through

Per quanto riguarda il Font è buona norma specificare anche uno dei font di default riconosciuti da tutti i Browser in tutte le piattaforme (anche se i risultati saranno diversi); ad esempio

font-family: verdana, arial, sans-serif;

in modo che se Verdana non è installato userà Arial e se proprio ci va male almeno userà un font "senza grazie" (ovvero a bastoncino).

Per quanto riguarda lo stile **oblique** è reso come italic da MS IE e non è supportato da Netscape.

Per le dimensioni del Font si possono usare misure assolute o relative; tra le misure assolute le dimensioni in punti (es: 10pt) devono essere usate con cautela in quanto le dimensioni saranno indipendenti dalla risoluzione dello schermo ed il carattere potrà essere quindi troppo piccolo o troppo grande (ricordatevi che esistono ancora schermi 640x480 e che 1024x768 su un 15" non è poi tanto raro); dovremo ovviamente usarle nelle parti grafiche a dimensioni fisse (ad esempio una scritta sovrapposta ad un'immagine). Le dimensioni equivalenti ai SIZE (da 1 a 7) del Tag FONT sono nell'ordine: xx-small, x-small, small, medium, large, x-large, xx-large.

Le dimensioni relative sono smaller | larger o espresse in percentuale (es:80%); tenere presente che l'effetto può combinarsi: se avete assegnato font-size:80%; al tag **p** ed al tag **a** e mettete un link all'interno di un paragrafo il testo del link sarà a grandezza 64% anziché 80%.

Per quanto riguarda l'indentazione si possono usare sia misure assolute (px = pixel, pt = punti) sia relative espresse in **em** ed in **ex**; queste particolari misure sono la larghezza della lettera m e l'altezza del carattere del font in uso in quel punto del documento.

Attributi della zona

Si applicano ai tag che definiscono una zona o una cella, a cominciare dal più esteso di essi (**body**) fino a **div** ed ai vari **h1-h6**.

background	colore di sfondo	vedi nota sull'uso dei colori
background-color	colore di sfondo	equivalente al precedente
background-image	immagine di sfondo	none, url(<i>file immagine</i>)
margin-left	margine sinistro	in tutte le unità di misura
margin-right	margine destro	idem
margin-top	margine superiore	idem
margin-bottom	margine inferiore	idem
padding	spazio dai bordi	in varie unità di misura
border-style	tipo di bordo	none, solid, double, groove, inset, outset, ridge
border-width	spessore del bordo	thin, medium, thick oppure misura
border-color	colore del bordo	vedi nota sull'uso dei colori
border	parametri del bordo	nel formato: style[width[color]]

per **padding** e **border** è possibile usare i suffissi -left, -right, -top e -bottom se si vogliono attribuire loro stili diversi (es: border-left-color); sarebbe possibile farlo anche specificando più parametri anziché uno solo ma non tutti i browser lo supportano.

Uso dei colori

Vi sono tre modi di indicare un colore: per nome, valori rgb, esadecimale; i nomi accettati sono:

colore	nome	rgb()	hex	colore	nome	rgb()	hex
	black	0,0,0	000000		silver	255,255,255	C0C0C0
	navy	0,0,128	000080		blue	0,0,255()	0000FF
	green	0,128,0	008000		lime	0,255,0	00FF00
	maroon	128,0,0	800000		red	255,0,0	FF0000
	olive	128,128,0	808000		yellow	255,255,0	FFFF00
	teal	0,128,128	008080		aqua	0,255,255	00FFFF
	purple	128,0,128	800080		fuchsia	255,0,255	FF00FF
	gray	128,128,128	808080		white	255,255,255	FFFFFF

In CSS2 il range nomi è più esteso ma visto che già con il CSS ci sono alcuni problemi con i browser non compliciamoci l'esistenza.

È possibile usare qualsiasi colore a 24 bit tramite la funzione rgb(val1,val2,val3) o tramite il valore esadecimale, ma si potrebbero ottenere risultati diversi a seconda dei browser, delle schede video e

delle piattaforme; questo problema è ovviamente tanto più importante quanto più vasta è l'utenza a cui ci si rivolge: ad esempio se il nostro sito si occupa di software per Windows 2000 possiamo essere quasi sicuri che gli utenti useranno un PC Windows e come browser MS IE 5 o superiore, ma se il nostro sito si occupa delle problematiche relative all'utilizzo dei sensori CCD nella fotografia astronomica non dovremo meravigliarci se ci saranno anche utenti Mac ed Unix e magari anche qualche browser di cui ignoravamo persino l'esistenza.

Perciò useremo solo i colori che tutti i browser supportano; oltre ai 16 colori base citati possiamo usare la cosiddetta "browser safe palette", composta da valori equidistanziati per le tre componenti. I valori da usare sono i seguenti:

Decimale	00	51	102	153	204	255
Esadecimale	00	33	66	99	CC	FF

potremo quindi per esempio usare #CCFF33 per un verde pisello non saturo o #993366 per un prugna scuro oppure usare rispettivamente `rgb(204,255,51)` e `rgb(153,51,102)`.

Attributi di posizione

Sono i seguenti:

position	modo posizionamento	absolute, relative
left	posizione margine sin.	in varie unità di misura
top	posizione margine sup.	idem
width	larghezza	idem
height	altezza	idem
visibility	visibilità	visible, hidden, inherit
z-index	livello	numero intero (default=0)

L'attributo **z-index** è particolarmente importante per controllare la visualizzazione di zone sovrapposte: se ad esempio vogliamo sovrapporre una scritta ad una immagine assegneremo `z-index:0`; all'immagine e `z-index:1`; alla scritta; occorre tenere presente che l'azione del mouse è sul livello in primo piano per evitare sorprese nel funzionamento dei link.

Il "subclassing"

Abbiamo già visto come definire una classe; è possibile definire anche sottoclassi operando nel modo seguente: abbiamo già definito lo stile per l'elemento DIV però vogliamo in alcuni casi renderlo con uno sfondo celeste pallido; scriveremo allora

```
div.celeste { background: #CCFFFF; }
```

e quando lo vogliamo usare scriveremo

```
<div class="celeste">
```

4. Come utilizzare la proprietà STYLE e come scrivere un foglio di stile

La risposta al primo quesito è molto semplice; dopo STYLE= inserite la stringa di stile tra virgolette (è una stringa!) nella forma *attributo1: valore1; attributo2: valore2; etc.*, ad esempio il codice:

```
<div style="width:14em; padding:0.2ex; border-left:solid 3px red; border-right:solid 3px red; color:fuchsia; text-align:center;">
Vediamo cosa viene fuori
</div>
```

produce questo risultato

Vediamo cosa viene fuori

Anche la risposta al secondo quesito è semplice: il file .css contiene in formato testo quello che avreste scritto tra i tag <STYLE type="text/css"> e </STYLE> se aveste inserito lo stile nel documento con in più eventuali righe di commento in testa (autore, data creazione, etc.) nel formato /* commento */; se non avete un editor appropriato potete scriverlo usando il Notepad, salvarlo come .txt e poi rinominarlo .css ed il gioco è fatto!

Per ulteriori informazioni potete visitare la [W3C Home page for CSS](#) (in inglese). Come guida iniziale vi consiglio "CSS e DHTML" di D.Livingston e M.Brown ed. italiana di [Tecniche Nuove](#); se poi volete sapere proprio tutto e non avete problemi con l'inglese ["Cascading Style Sheets"](#) di Håkon Wium Lie & Bert Bos, 1999 Addison Wesley, \$29.95.