Principali proprietà dei CSS

Box Model e proprietà di base del testo

Sviluppo di siti web – UD10

prof. Giovanni Borga

Il BOX MODEL

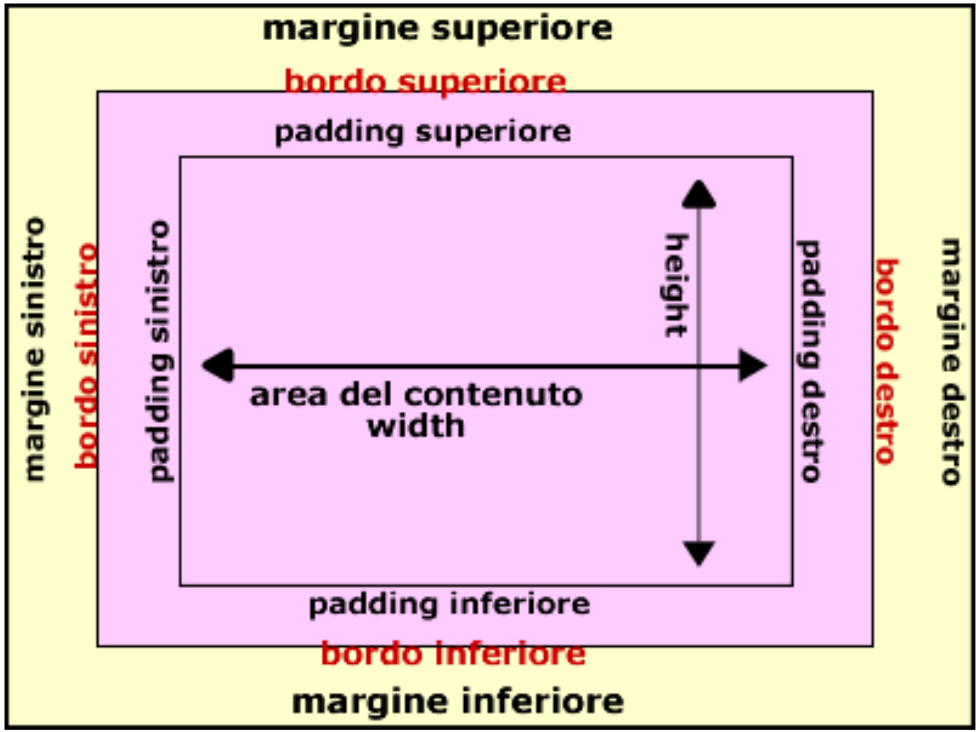
Il BOX MODEL

Se si vuole usare i CSS per scopi che vadano oltre la semplice gestione di sfondo e testo occorre avere ben chiaro il meccanismo che governa la presentazione dei vari elementi di una pagina.

Abbiamo visto che una pagina (X)HTML non è altro che un insieme di box rettangolari (che si tratti di elementi blocco o di elementi inline non fa differenza).

Tutto l'insieme di regole che gestisce l'aspetto visuale degli
elementi blocco viene in genere riferito al cosiddetto
box model.

Ogni box comprende un certo numero di componenti di base, ciascuno modificabile con proprietà dei CSS.



Componenti del box model

- **Area del contenuto** - È la zona in cui trova spazio il contenuto vero e proprio, testo, immagini, animazioni Flash. Le dimensioni orizzontali dell'area possono essere modificate con la proprietà width. Quelle verticali con height.
- **Padding** - È uno spazio vuoto che può essere creato tra l'area del contenuto e il bordo dell'elemento. Come si vede dalla figura, se si imposta un colore di sfondo per un elemento questo si estende dall'area del contenuto alla zona di padding.
- **Bordo** - È una linea di dimensione, stile e colore variabile che circonda le zone del padding e del contenuto.
- **Margine** - È uno spazio di dimensioni variabili che separa ogni elemento da quelli adiacenti.

NB: queste componenti non sono state introdotte con i CSS, ma fanno parte del normale meccanismo di rendering di un documento. Quando realizziamo una pagina (X)HTML senza fogli di stile è il browser ad applicare per alcune di queste proprietà le sue impostazioni predefinite. Per esempio, introdurrà un certo margine tra un titolo e un paragrafo o tra due paragrafi.

La novità è che con i CSS possiamo controllare con precisione al pixel tutti questi aspetti. Il box model è governato da una serie di regole di base concernenti la definizione di un box e il suo rapporto con gli altri elementi.

Larghezza del BOX

Bisogna distinguere tra la larghezza dell'area del contenuto e la larghezza effettiva di un box .

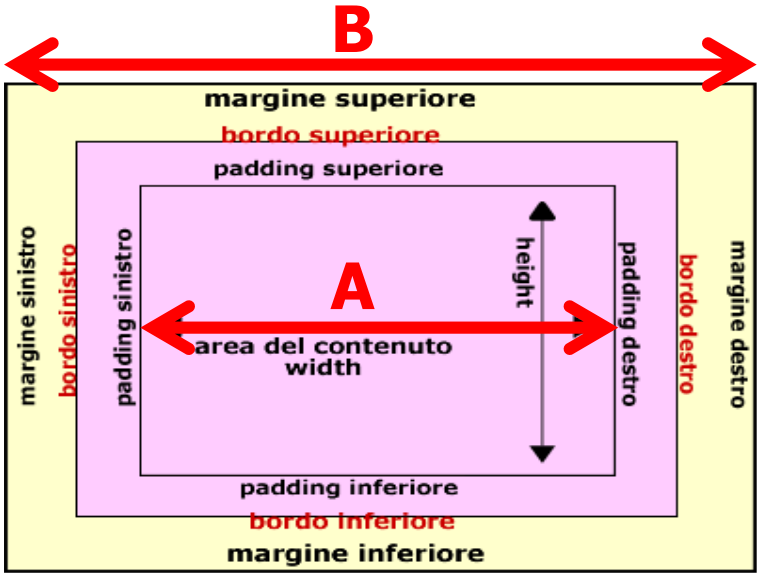
A - LARGHEZZA DELL'AREA DEL CONTENUTO = valore della proprietà width.

B - LARGHEZZA DEL BOX = margine sinistro + bordo sinistro + padding sinistro + area del contenuto + padding destro + bordo destro + margine destro

Margini, padding e bordi devono considerarsi a tutti gli effetti parte dell'area complessiva dell'elemento.

Se non si imposta WIDTH, o se si imposta a «auto» la larghezza di un box è uguale a quella dell'area del contenuto dell'elemento contenitore.

Cioè, ogni elemento «invade» automaticamente lo spazio in larghezza a sua disposizione che corrisponde allo spazio del contenuto dell'elemento che lo contiene.



Il valore AUTO

Per **tre** proprietà del box model è possibile utilizzare il valore «**auto**»: Le proprietà sono:

margini, altezza e larghezza

L'effetto è quello di lasciar calcolare al browser l'ammontare di ciascuna di esse in base alla risoluzione dello schermo e alle dimensioni della finestra.

Valori negativi

Sono ammesi valori negativi ma **solo per i margini**, Non è consentito per padding, bordi, altezza e larghezza.

Margini verticali e orizzontali tra gli elementi

Per due box adiacenti in senso verticale, che abbiano impostato un margine inferiore e uno superiore, **la distanza non è data dalla somma** delle due distanze.

A prevalere sarà invece la distanza maggiore tra le due.

Il meccanismo è detto del «**margin collapsing**» e non si applica nel senso orizzontale.

Proprietà di base del testo

Proprietà COLOR

La proprietà color si applica a tutti gli elementi ed è ereditata.

Se si imposta il colore per un elemento esso sarà ereditato da tutti gli elementi discendenti per cui non si definisca esplicitamente un altro colore.

I valori possibili sono:

- qualunque **valore di un colore** definito con i metodi descritti più avanti
- la parola chiave **inherit**. *(Con essa si dice esplicitamente al browser di ereditare le impostazioni definite per l'elemento parente)*.

Nell'esempio che segue dichiariamo che tutti i paragrafi di un documento devono avere il testo in rosso:

```
p {color: red }
```

PROPRIETA' COLOR

La definizione dei colori ha la sintassi di base dell'HTML. Ecco il riepilogo.

PAROLE CHIAVE

BLACK	FUCHSIA
NAVY	OLIVE
BLUE	GRAY
MAROON	LIME
PURPLE	AQUA
GRENN	SILVER
RED	YELLOW
TEAL	WHITE

CODICI ESADECIMALI

Esempio:

#CC0000 =

Alcuni colori:

Colore	Hex	Colore	Hex
black	000000	silver	C0C0C0
navy	000080	blue	0000FF
green	008000	lime	00FF00
teal	008080	aqua	00FFFF

VALORI RGB: sintassi abbreviata

E' possibile usare per essi una sintassi abbreviata in cui i valori per il rosso, il verde e il blue sono definiti solo dalla prima lettera o numero. Il colore dell'esempio precedente può essere definito anche così: #C00

Proprietà COLOR: specificità dei CSS

Con i CSS è possibile utilizzare anche una tecnica specifica per definire un codice colore, utilizzando la parola chiave RGB seguita da valori in percentuale o in numeri interi da 0 a 255.

PERCENTUALI RGB

#RGB

rgb(rrr%, ggg%, bbb%)

rgb(0%,0%,0%)

rgb(100%,100%,100%)

VALORI RGB

#RGB

rgb(rrr, ggg, bbb)

rgb(0,0,0)

rgb(255,255,255)

Per ogni elemento si possono definire almeno tre colori:
il colore di **primo piano** (foreground); il colore del **bordo** (border), il colore di **sfondo** (background).

Ma con la proprietà COLOR possiamo definire solo i primi due, ovvero il colore del testo e dei bordi.

Per altre caratteristiche esistono proprietà apposite che vedremo successivamente.

Gestione dello sfondo

Si tratta di una delle più importanti novità introdotte dai CSS.

Lo sfondo si gestisce con 5 proprietà specifiche che **possono essere applicate a tutti gli elementi**:

1. **background-color**
2. **background-image**
3. **background-repeat**
4. **background-attachment**
5. **background-position**

Sono tutte proprietà NON ereditate. Ciascuna di essa definisce un solo, particolare aspetto relativo allo sfondo di un elemento.

La proprietà **background**, invece, è la proprietà a sintassi abbreviata con cui possiamo definire sinteticamente e con una sola dichiarazione tutti i valori per lo sfondo.

Proprietà singole dello sfondo

background-color (colore di sfondo)

Valori: *un qualunque colore* | transparent (corrisponde al colore dell'elemento parente)

Esempi:

body { background-color: white }
p { background-color: #FFFFFF }

---

background-image (URL di un'immagine come sfondo)

Valori: *una URL assoluta o relativa che punti ad un'immagine* | none (default)

Esempi:

body { background-image: url(sfondo.gif) }
div { background-image: url(http://www.server.it/images/sfondo.gif) }

---

background-repeat (direzione in cui l'immagine di sfondo viene ripetuta)

Valori: repeat (default) | repeat-x | repeat-y | no-repeat

Esempi:

body { background-image: url(sfondo.gif) ; background-repeat: repeat }
div { background-image: url(sfondo.gif) ; background-repeat: repeat-x }

Proprietà singole dello sfondo

background-attachment (scorrimento dello sfondo rispetto al testo).
Valori: scroll | fixed
Esempio: **body { background-image: url(back.gif) ; background-attachment: fixed }**

---

background-position (punto di inserimento dell'immagine di sfondo)
Valori: coordinate di un punto sugli assi verticale e orizzontale con tre modalità:
valori in percentuale | *valori espressi con unità di misura* | top | left | bottom | right
Esempi:
body {background-image: url(back.gif) ; background-repeat: no-repeat ; background-position: 50px 50px }
(l'immagine apparirà a 50px dal bordo superiore e a 50px da quello sinistro della finestra)

body {background-image: url(back.gif) ; background-repeat: no-repeat ; background-position: center center }
(l'immagine apparirà centrata in entrambe le direzioni)

NB: se si imposta un solo valore esso verrà usato sia per l'asse orizzontale sia per quello verticale.

Proprietà a sintassi abbreviata per lo sfondo

Con la proprietà **background** possiamo definire in una unica dichiarazione tutti gli aspetti dello sfondo.
Per essere valida, la dichiarazione non deve contenere necessariamente riferimenti a tutte le proprietà viste finora, ma deve contenere almeno la definizione del colore di sfondo.

La struttura sintattica è la seguente:

selettore {background:background-color background-image background-repeat background-attachment background-position; }

I valori non vanno separati da virgole. L'ordine delle caratteristiche non è influente.

Esempio: **body { background:white url(sfondo.gif) repeat-x fixed }**

Proprietà di base del testo

Un aspetto essenziale dei CSS riguarda il nuovo approccio alla gestione del testo. Con i CSS viene sostanzialmente **abolito il tag <font>**.

Vengono introdotte molte nuove proprietà che potremmo dividere tra proprietà di base e proprietà estese.

Le proprietà di base sono quelle che permettono di gestire i seguenti aspetti:

- Il **font** da usare
- La **dimensione** del testo
- La **consistenza** del testo
- L'**interlinea** tra i paragrafi
- L'**allineamento** del testo
- La cosiddetta «**decorazione**» del testo, ovvero sottolineature, barrati ecc.

Carattere del testo

Per impostare il tipo di carattere di una porzione di testo si utilizza la proprietà **font-family**.
Font-family è applicabile a tutti i tag ed è ereditata. La sintassi permette di impostare non solo un singolo font ma un elenco di font alternativi; nell'esempio che segue:

```
p { font-family: arial, Verdana, sans-serif }
```

il browser tenterà di usare il primo font della lista. Se questo non è disponibile userà il secondo. In mancanza anche di questo verrà utilizzato il font principale della famiglia sans-serif presente sul sistema.

Quando si imposta la proprietà font-family si possono usare tutti i font che si vuole, ma **è opportuno inserire alla fine l'indicazione di una famiglia generica** tra le cinque elencate: (tra parentesi riportiamo i caratteri predefiniti sui sistemi Windows):

- **serif** (Times New Roman)
- **sans-serif** (arial)
- **cursive** (Comic Sans)
- **fantasy** (Allegro BT)
- **monospace** (Courier)

I nomi nella lista vanno separati dalla virgola.
I caratteri con nomi composti da più parole vanno inseriti tra virgolette.

Esempio: `div { font-family: Georgia, "Times New Roman", serif }`

Dimensione del testo

Font-size, insieme a font-family è la proprietà considerata essenziale nella definizione dell'aspetto del testo, di cui definisce le dimensioni. Applicabile a tutti gli elementi ed ereditata.

Le dimensioni possono essere espresse in senso **assoluto** o in senso **relativo**.

(*Assoluto significa che essa non dipende da nessun altro elemento ed è quella definita dall'unità di misura usata. Relativo significa che essa viene calcolata in base alla dimensione del testo dell'elemento parente*).

I valori assoluti ammessi sono:

- le sette parole chiave **xx-small, x-small, small, medium, large, x-large, xx-large**
- quelli espressi con le seguenti unità di misura: pixel (**px**), centimetri (**cm**), millimetri (**mm**), punti (**pt**), picas (**pc**), pollici (**in**), x-height(**ex**). Per i testi a video si consigliano però sono punti e pixel. Le altre sono più adatte per i CSS destinati alla stampa.

I valori relativi ammessi sono:

- le parole chiave **smaller** e **larger**
- quelli espressi in **em** (em-height)
- quelli espressi in **percentuale**

Esempi: **p { font-size: 12px }** ; **div.titolo { font-size: 50% }** ;
#div1 { font-size: large } ; **h2 { font-size: 1.2em }**

Consistenza del testo

Font-weight serve a definire la consistenza (il «peso visivo») del testo.
Si applica a tutti gli elementi ed è ereditata.

I valori ammessi per font.weight possono appartenere ad una scala numerica o essere delle parole chiave; anche in questo caso abbiamo valori assoluti o relativi:

- valori numerici **100** - **200** - **300** - **400** - **500** - **600** - **700** - **800** - **900** ordinati in senso crescente (dal leggero al pesante)
- **normal**. Valore di default. E' l'aspetto normale del font ed equivale al valore 400
- **bold**. Il carattere acquista l'aspetto che definiamo in genere grassetto. Equivale a 700
- **bolder**. Misura relativa. Il testo sarà più pesante rispetto al testo dell'elemento parente
- **lighter**. Misura relativa. Il testo sarà più leggero di quello dell'elemento parente

Esempi:

```
p { font-weight: 900 } ;  div { font-weight: bold }
```

Stile del testo

La proprietà **font-style** imposta le caratteristiche del testo in base ad uno di questi tre valori:

- **normal**: il testo mantiene il suo aspetto normale
- **italic**: formatta il testo in corsivo
- **oblique**: praticamente simile a italic

La proprietà si applica a tutti gli elementi ed è ereditata.

L'esempio è semplicissimo:

```
p { font-style: italic }
```

Interlinea

Grazie a **line-height** è possibile impostare dimensioni di interlinea in modo molto flessibile.

La proprietà, in realtà, serve a definire l'altezza di una riga di testo all'interno di un elemento blocco, ma l'effetto ottenuto è appunto quello ricercato da molti editor, ovvero un modo per impostare uno spazio tra le righe. La proprietà si applica a tutti gli elementi ed è ereditata.

Valori ammessi:

- **normal**. Il browser separerà le righe con uno spazio «standard».
  Generalmente corrispondente a un valore numerico compreso tra 1 e 1.2
- **valore numerico intero o decimale**. Usando valori numerici con dei decimali (es. 1.2, 1.3, 1.5) si un risultato in cui l'altezza della riga è uguale alla dimensione del font moltiplicata il valore espresso.
- **valore numerico con unità di misura**. L'altezza della riga sarà uguale alla dimensione specificata.
- **valore percentuale**. L'altezza della riga viene calcolata come una percentuale della dimensione del font.

E' sconsigliato l'utilizzo dei valori in percentuale e in unità esplicite; in generale è più controllabile la resa con valori numerici.

Esempi:

```
p { line-height: 1.5}
```

```
body { line-height: 15px }
```

Sintassi abbreviata per il testo

La proprietà **font** può essere paragonata a background. Si tratta di una proprietà a sintassi abbreviata che serve a impostare con una sola dichiarazione tutte le principali caratteristiche del testo.

Le proprietà definibili in forma abbreviata con font sono quelle che abbiamo descritto finora, ovvero:

font-family* | *font-size* | *line-height* | *font-weight* | *font-style* | *font-variant* | *font di sistema

Valgono le seguenti regole:

- I **valori** delle singole proprietà NON vanno separati da virgole.
- I valori definiti per la **font-family** VANNO separati da virgole *(anche in questo caso i nomi dei font costituiti da più parole vanno racchiusi tra virgolette)*
- Per quanto riguarda l'**ordine** delle proprietà, la dichiarazione dovrebbe sempre finire con la coppia font-size > font-family.
- Se si vuole indicare un valore di **line-height** è necessario mettere in valore dopo quello di font-size separandolo da uno slash. *(es. 10px/1.2, 24pt/1.5 ecc.)*

Nell'esempio:

```
p { font: bold 12px/1.5 Georgia, "Times New Roman", serif }
```

Nell'ordine abbiamo definito: font-weight, dimensione/ interlinea, font-family.

Allineamento del testo

La proprietà **text-align** serve a impostare l'allineamento del testo. E' ereditata e si applica a tutti gli elementi.

Sono ammessi i valori:

- **left**. Allinea il testo a sinistra
- **right**. Allinea il testo a destra
- **center**. Centra il testo
- **justify**. Giustifica il testo

Esempio:

```
p { text-align: center }
```

«decorazione» del testo

Text-decoration imposta particolari decorazioni e stili per il testo.
Ereditabile e applicabile a tutti gli elementi.

Valori ammessi:

- **none**. Il testo non avrà alcuna decorazione particolare
- **underline**. Il testo sarà sottolineato
- **overline** Il testo avrà una linea superiore
- **line-through**. Il testo sarà attraversato da una linea orizzontale al centro
- **blink**. Testo lampeggiante *(molto sconsigliato perché poco supportato dai vecchi browser e per il fatto che va valutato con molta attenzione l'inserimento di elementi «animati» in una pagina perché se sono troppi rendono poco efficace la comunicazione)*

Esempi:

```
p { text-decoration: overline }
```

```
a { text-decoration: none }
```