



## LEZIONE 6 APPROFONDIMENTI CSS



Completiamo i css necessari per la home page.

Gli elementi della pagina (header,nav,article,aside e footer) sono racchiusi in un elemento <div> che è un contenitore vuoto a cui, però, assoceremo un id che chiameremo **container** da scrivere nel file style-home-page.css della cartella risorse. La regola CSS potrebbe essere questa:

```
.container {  
    background: #eef;  
    margin: 0 auto;  
    width: 800px;  
  
    box-shadow: 5px 0px 30px #000; }
```

Nel file index.html dobbiamo solamente inserire nel tag di apertura <div> l'id di nome container senza il simbolo di cancelletto.

```
<div class="container" >
```

Approfondiamo nel dettaglio le dichiarazioni.

**background: #eef;**

background è il colore di sfondo; quando compaiono tre simboli invece di sei è come se ci fosse scritto **#eeeff**: **ee** è il numero esadecimale del canale rosso, **ee** è il numero esadecimale del canale verde e **ff** è il numero esadecimale del canale blu. Possiamo anche scrivere come indicato nella lezione precedente anche il colore scritto in inglese da una lista di 147 nomi o il colore in decimale, per esempio lo stesso colore corrisponde a:

**background: rgb(238,238,255);** perché **ee** in esadecimale corrisponde a 238 in decimale e **ff** a 255.

**margin: 0 auto;** è equivalente a **margin: 0 auto 0 auto;**

Il primo zero è il margine in alto il secondo zero è il margine in basso, auto è il margine di destra e di sinistra. Quando il margine è impostato ad auto il contenitore viene centrato orizzontalmente.

**width: 800px;**

E' la larghezza del contenitore.

**box-shadow: 5px 5px 10px 2px #333;**

**box-shadow** può essere applicata a tutti gli elementi.

La sintassi di base prevede la definizione di quattro valori numerici:

Il primo valore (5px) imposta lo spostamento dell'ombra in senso orizzontale. Essendo il valore positivo, l'ombra viene collocata sul lato destro del box. Se si usano valori negativi l'ombra viene collocata sul lato sinistro.

Il secondo valore (5px) imposta lo spostamento in senso verticale. Dal momento che anche in questo caso si è usato un valore positivo, l'ombra viene collocata sul lato inferiore del box. Usando valori negativi l'ombra viene collocata sul lato superiore del box.

Il terzo valore (10px) definisce il livello di sfocatura (**blur radius**) dell'ombra: più alto è questo valore, più sfocata apparirà l'ombra. Si usi 0 se si desidera un'ombra netta e senza sfocatura. Non sono permessi valori negativi.

Il quarto valore (2px) imposta il livello di diffusione (**spread radius**) dell'ombra. Più i valori sono alti, più l'ombra tenderà ad espandersi, in tutte le direzioni. Se invece si usano valori negativi, l'ombra tende a contrarsi, fino a scomparire del tutto.

Di questi quattro valori, gli unici obbligatori sono i primi due. I valori della sfocatura e della diffusione, invece, possono essere omessi. Se non specificati, tali valori sono pari per default a 0. La nostra regola si chiude con l'indicazione del colore dell'ombra (#333333).

Analizziamo ora ogni riquadro della home page iniziando dall'elemento **<header>** che racchiude l'intestazione della pagina, con la sola scritta "IL COLORE".

```
header {  
    text-align:center;  
    text-shadow:2px 2px 3px red;  
    text-size:20px;  
    padding:2px;  
    border: 2px groove blue;  
    color:black;  
}
```

### **text-align:center;**

Questa proprietà centra il testo; teniamo presente che header è racchiuso in container che ha una larghezza di 800 pixel. Oltre al centro si può scegliere di allineare a destra o a sinistra.

### **text-shadow:2px 2px 3px red;**

Una novità dei CSS3 è la proprietà text-shadows che crea un'ombreggiatura al testo: il primo valore (2px) definisce lo spostamento dell'ombra sull'asse orizzontale (x); il secondo valore (2px) definisce lo spostamento dell'ombra sull'asse verticale (y); il terzo valore (3px) imposta il livello di sfocatura (blur) dell'ombra: più alto è questo valore, più sfocata apparirà l'ombra; se si usa 0 otterremo un'ombra netta e senza sfocatura; il quarto valore (red) definisce il colore dell'ombra.

### **text-size:20px;**

Questa proprietà definisce la grandezza del carattere;

### **padding:2px;**

Il padding è la distanza del testo rispetto al bordo; con un solo valore impostato, la distanza di 2 pixel riguarda tutti e 4 lati: sopra, a destra, sotto e a sinistra.

## **border: 2px groove blue;**

Questa proprietà assegna un bordo all'elemento <header> di spessore 2 pixel di stile groove e di colore blu. I possibili stili dei bordi sono:

Stile bordo      Descrizione

**none**      l'elemento non presenta alcun bordo e lo spessore equivale a 0

**hidden**      equivalente a none

**dotted**      bordo a puntini

**dashed**      bordo a lineette

**solid**      bordo solido e continuo

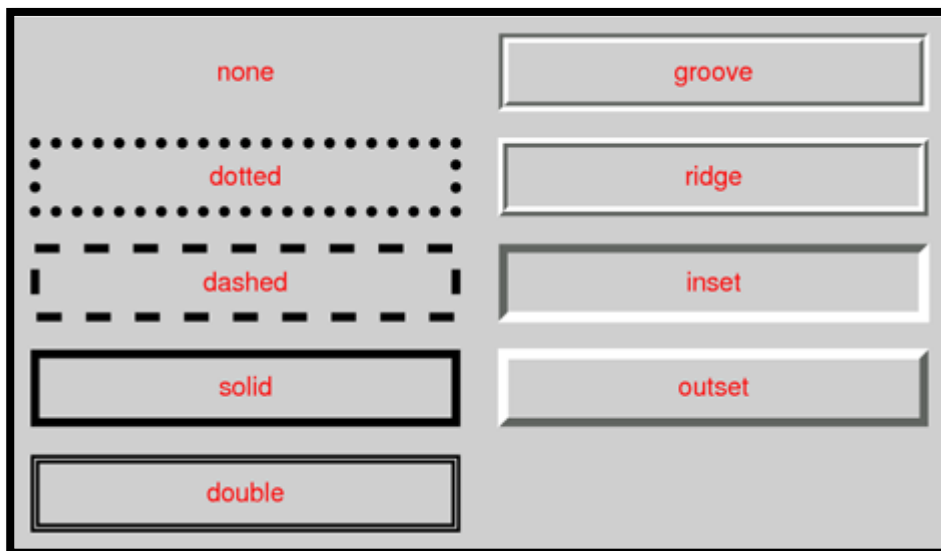
**double**      bordo solido, continuo e doppio

**groove**      tipo di bordo in rilievo

**ridge**      altro tipo di bordo in rilievo

**inset**      effetto 'incastonato'

**outset**      effetto 'sbalzato'



Con i CSS3 vi è anche la possibilità di arrotondare gli angoli dei bordi con la proprietà **border-radius**, si rimanda alla rete ulteriori approfondimenti. (es. <http://www.html.it/pag/19451/border-radius/> ).

## **color:black;**

E' il colore di primo piano.


Analizziamo adesso la regola CSS applicata all'elemento `<nav>` quello che racchiude gli elementi di navigazione, i link veri e propri. (**teniamo presente che si tratta di una barra di navigazione orizzontale**).

```
nav {  
    width:796px;  
    border: 2px groove red;  
    list-style:none;  
    margin:0;  
    padding:0;  
    text-align:center;  
    background-color:#3FF }  
}
```

**Una cosa fondamentale da sapere è come calcolare la misura del riquadro: per esempio se il contenitore ha una larghezza (width:800 pixel) di 800 pixel, l'elemento `<nav>` contenuto in esso non deve superare tale numero. Ma come si calcola la larghezza?**

**La larghezza è data dal contenuto misurato dalla proprietà width + il padding sinistro + il padding destro + il bordo destro + il bordo sinistro.**

**Nel caso specifico la larghezza del riquadro è:**

796(content)	}	
+2 bordo destro		
+2 bordo sinistro		
+0 padding		

Il container ha 800px ma nessun bordo e nessun padding. Se dovessimo mettere 1 pixel per il bordo destro e 1 pixel per il bordo sinistro dovremmo scalare di due il content (width 798px) per avere la lunghezza complessiva di 800 pixel.

**list-style:none;**

Questa proprietà serve a non visualizzare i simboli ordinati o non degli elementi della lista.

```
nav li {  
    display: inline;  
}
```

Un'altra regola importante da sapere è che gli elementi della lista sono considerati **elementi di blocco e vengono visualizzati in colonna, uno per riga**. E' bene ricordare che la distinzione fra elementi in blocco e in linea con la versione HTML5 è venuta meno, grazie ai nuovi tag semantici, e gli elementi in linea sono categorizzati come elementi a contenuto frasale (phrasing content), tuttavia il ricorso ai tag vuoti `<div>` non semantici a volte è utile ed è bene conoscere anche questa vecchia distinzione.

Nel nostro caso vogliamo visualizzare gli elementi sulla stessa riga e la proprietà **display:inline;** serve a questo. Se non indicato è come se ci fosse **display:block;** .

Quando gli elementi sono in linea non vengono visualizzati i punti elenco; se fossero in righe separate con **display:block;** e non volessimo visualizzare i punti elenco avremmo dovuto indicare anche la proprietà **list-style:none;**. Questa proprietà serve ad indicare quale punto elenco vogliamo visualizzare tra i tanti proposti (si rimanda alla rete per ulteriori approfondimenti).

```
ul {  
    padding:0px; margin:13px auto;  
  
}
```

### **padding:0px;**

Questa proprietà applicata all'elemento <ul> imposta a 0 il padding e così facendo possiamo gestire meglio il padding degli elementi <a> (link).

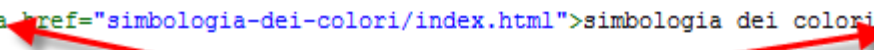
### **margin:13px auto;**

I margini in questo caso sono legati alla prossima regola, 13px è uguale al padding di a (sopra e sotto) e auto serve a centrare gli elementi li all'interno di ul che è inserito in nav.

```
a {  
    color:black;  
    padding:13px 40px;  
    font-weight:bold;  
    text-decoration:none;  
}
```

Nella prossima lezione tratteremo meglio i link (il tag <a> ) per ora lo applichiamo ai tre elementi della barra di navigazione nel file index.html

```
<li><a href="metodi-di-colore/index.html"> metodo colore </a></li>  
<li><a href="colore-nella-arte/index.html"> colore nell'arte</a></li>  
<li><a href="simbologia-dei-colori/index.html">simbologia dei colori</a></li>  
..
```



L'elemento <a> contiene l'attributo obbligatorio href per collegare la pagina.

### **color:black;**

Questa proprietà imposta il colore del collegamento a nero.

### **padding:13px 40px;**

Questa proprietà imposta il padding.

### **font-weight:bold;**

Questa proprietà imposta il grassetto al testo.

### **text-decoration:none;**

Questa proprietà elimina la sottolineatura al collegamento che altrimenti i browser inseriscono automaticamente.

### **a:hover {**

**color:white;**

**background-color:blue; }**


Queste proprietà impostano il colore bianco di primo piano e blu di sfondo nello stato hover, cioè quando il puntatore del mouse è posizionato sul collegamento.

```
article {
    width:576px;
    height:700px;
    padding:10px;
    float:left;
    border: 2px groove red;
}
```

Analizziamo gli ultimi tre elementi (article,aside e footer). La prima cosa da notare è che nel layout che ci siamo prefissati di creare, article e aside devono essere accostati e footer invece visualizzato sotto. Se notiamo le proprietà di article (nelle prime tre non c'è niente di nuovo) ma la quarta **float:left;**

è importante per capire che se usiamo la proprietà float, l'elemento che seguirà article, ovvero **aside** fluttuerà a sinistra e non sotto come sarebbe se non ci fosse il floating.

```
aside {
    padding-bottom:20px;
    width: 196px;
    height: 700px;
    float:left;
    border: 2px groove red;
}
```



Anche aside deve contenere la proprietà float:left.

```
footer {
    background-color:black;
    color:white;
    clear:both;
    text-align:center;
    padding:5px;
}
```

Quando invece vogliamo interrompere l'accostamento degli elementi e riprendere il normale flusso di visualizzazione (un elemento su una riga nuova) dobbiamo inserire la proprietà **clear:both;** che annulla il floating.