



I FOGLI DI STILE

Antonio Mercurio

I fondamentali del CSS con accenni al CSS3

antonio mercurio

Questo libro è in vendita presso <http://leanpub.com/ifondamentaldelcssconaccennialcss3>

Questa versione è stata pubblicata il 2016-09-26



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2016 antonio mercurio

Also By antonio mercurio

Php le basi: per principianti

Laravel 5.2 : Per principianti

Guida a GIT

Dedico questo libro a tutti gli sviluppatori di software libero.

Indice

| | |
|--|------|
| La sintassi | ii |
| Come inserire il codice | iii |
| Come applicare i colori al testo | vi |
| Le classi | viii |
| Background | xii |
| Bordi | xiv |
| I margini | xix |

#Cosa bisogna sapere prima di leggere questo libro

Per capire questo libro basta possedere le basi del linguaggio html in quanto il Css si basa proprio sugli elementi html che dovrete conoscere. Inoltre avrete bisogno di un editor html o multilinguaggio come per esempio i seguenti: - **Komodo edit** scaricabile da qui [Komodo edit](http://komodoide.com/komodo-edit/)¹ - **Brackets** scaricabile da qui [Brackets](http://brackets.io/)²

Entrambi sono opensource, multilinguaggio e free. Inoltre sono cross-platform perchè girano su Linux, Windows e Mac. Questi editor sono ottimi perchè sono estensibili con altri plugins e poi hanno la funzione di auto-completamento del codice, quindi vi aiutano nella scrittura del codice necessario. Anche perchè senza di essi si potrebbe programmare ma sarebbe difficile a volte ricordare a memoria i nomi di proprietà.

¹<http://komodoide.com/komodo-edit/>

²<http://brackets.io/>

La sintassi

Il css o (fogli di stile a cascata) consiste in una serie di proprietà e valori applicati sugli elementi html che conoscete, e che consentono di formattare e stilizzare i contenuti dei nostri siti o pagine senza doverlo riscrivere singolarmente per ognuna di esse. Immaginate di aver scritto 100 pagine web e di voler ad un tratto applicare una formattazione unica a tutte. Dovreste farlo nel codice di ogni pagina, mentre con i fogli di stile possiamo creare un file in formato .css ed importarlo semplicemente in ogni pagina nella sezione `<head></head>` e così tutte le pagine saranno formattate allo stesso modo. In pratica il linguaggio html serve ad inserire testo ed altri elementi sulle pagine, mentre il Css consente di stilizzare e curare la formattazione dei contenuti. Ecco come si presenta un codice css tipico

```
p{  
font-size:18px;  
}
```

Abbiamo la parte esterna alle parentesi graffe che sarebbe l'elemento html su cui applichiamo la regola e in questo caso il paragrafo. Questo viene definito selettore. La parte font-size è la proprietà che in questo caso sarebbe la grandezza di un carattere e dopo i due punti abbiamo il suo valore che è la misura del font in pixel. Ogni istruzione css termina con un ;.

Come inserire il codice

Le istruzioni css possono essere inserite in un file css esterno. Per iniziare questo corso create nel vostro editor una cartella di nome css e al suo interno create un file di tipo css e nominatelo stile in modo da avere un file stile.css. All'interno di esso inserite questo codice

```
p{
font-size:26px;
}
```

poi salvate. Ora create nella cartella del progetto un file html e lo nominate test, quindi ora abbiamo un file stile.css e uno test.html. Non cancellate questi due files mai perchè li useremo sempre durante il nostro studio. Ora nella sezione <head> del file html inserite questa istruzione <link rel="stylesheet" type="text/css" href="stile.css"> .

Spieghiamo questa istruzione. Link rel fa in modo che il link si riferisca ad una risorsa di tipo css, infatti stylesheet sta per foglio di stile. Poi segue type=text/css che rappresenta la tipologia di testo e href indica il percorso dove si trova il file. In questo caso non abbiamo dovuto indicare un percorso perchè si trova già nella stessa cartella del progetto e non in sottocartelle. Questa modalità di inserire il css è quella che useremo sempre nel libro e quindi non cancellate il file e non spostatelo di posizione. Si definisce questa modalità css esterno. Ora provate a scrivere uno o più paragrafi nel vostro file html come segue

css esterno

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6
7     <link rel="stylesheet" href="stile.css" type="text/css">
8
9 </head>
10 <body>
11
12
13 Testo senza formattazione
14
15 <p>Testo con la proprietà font.weight</p>
16
```



```
17 <p>Testo con la proprietà font.weight</p>
18
19 <p>Testo con la proprietà font.weight</p>
20
21
22
23 </body>
24 </html>
```

Aprire la pagina in un browser e noterete che tutti i paragrafi hanno un font con grandezza 26 pixels. I 26 pixels noi li indichiamo con 26px. Un secondo modo per inserire il css è all'interno della sezione <head> ma usando il tag <style>, così

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6
7     <style>
8
9         p{
10
11             font-size:26px;
12
13         }
14
15
16
17     </style>
18
19 </head>
20 <body>
21
22
23 Testo senza formattazione
24
25 <p>Testo con la proprietà font.weight</p>
26
27 <p>Testo con la proprietà font.weight</p>
28
29 <p>Testo con la proprietà font.weight</p>
```

```
30
31
32
33 </body>
34 </html>
```

In pratica inseriamo nella sezione <head> una sezione <style> e al suo interno il codice. Si definisce css interno alla pagina. Se lo usate uguale in cento pagine e poi decidete di apportare una modifica sarete costretti a ricopiarlo uguale in tutte le pagine, quindi questa modalità va bene quando una o più pagine del vostro sito si distinguono nella formattazione e vogliono applicare una standard distinto dal css esterno. Infine esiste un terzo modo per inserire css ed è definito in linea. Il nostro esempio di prima potrebbe riprodursi così

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7
8
9  </head>
10 <body>
11
12
13 Testo senza formattazione
14
15 <p style="font-size: 26px;">Testo con la proprietà font.weight</p>
16
17 <p style="font-size: 26px;">Testo con la proprietà font.weight</p>
18
19 <p style="font-size: 26px;">Testo con la proprietà font.weight</p>
20
21
22
23 </body>
24 </html>
```

In pratica se volete applicare il font con grandezza 26 pixels dovete applicarle per ogni paragrafo che vi interessa. Dovete inserire nello stesso paragrafo style e poi tra apici l'istruzione css. Questi sono i tre modi di inserire css ma noi useremo sempre il css esterno e vi conviene insistere su questo.

Come applicare i colori al testo

Immaginiamo di voler applicare un colore al testo normale nella sezione <body> ed un colore diverso ai paragrafi. Supponiamo di scegliere il colore giallo per il testo normale ed il colore rosso per il testo in paragrafo.

```
1  body{
2
3  color:yellow;
4
5  }
6
7  p{
8
9
10 color:red;
11
12
13
14 }
```

Ricordate da questo momento in poi il codice va sempre inserito nel file stile.css e nella sezione <head> dovete importarlo come abbiamo visto. Provate a scrivere del testo così

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7
8
9  </head>
10 <body>
11
12
13 Testo con colore giallo
14
```

```
15 <p>Paragrafo rosso</p>  
16  
17  
18  
19 </body>  
20 </html>
```

Nel codice abbiamo usato su body la proprietà color con il valore yellow. Cioè color sta per colore e yellow per giallo. Il valore ricordate sempre dopo i due punti. Mentre la stessa proprietà ma con valore red (rosso) per i paragrafi. Ogni selettore (elemento html) su cui impostiamo le proprietà deve essere inserito fuori le parentesi graffe. Se usate uno degli editor che vi ho consigliato, i vari colori vengono suggeriti, e comunque sono gli equivalenti dell'inglese. Non cancellate questo esempio che possiamo utilizzare nel prossimo argomento.

Le classi

Nell'esempio di prima abbiamo quattro paragrafi e tutti utilizzano il colore rosso. Questo succede perchè abbiamo definito le proprietà per body e p come selettori. Possiamo però utilizzare uno strumento di css che sono le classi. Con esse noi possiamo creare regole da applicare solo a parti della pagina pur mantenendo le regole create in precedenza. Quindi nel nostro esempio manteniamo le due regole per i colori di body e di p ma creiamo una classe per applicare il colore blue ai soli paragrafi. Ecco il codice modificato

```
1  body{
2
3
4
5      color:yellow;
6
7
8  }
9
10
11  p{
12
13
14
15      color:red;
16
17
18
19  }
20
21
22  .blu{
23
24
25      color:blue;
26
27
28
29  }
```

Abbiamo creato una classe nominata blu preceduta da un punto. Per creare una classe la sintassi è .nomeclasse. Poi abbiamo specificato un colore blue. Ora questa classe non è applicabile solo ai paragrafi ma anche al testo normale basta richiamarla nella pagina come segue

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7      <link rel="stylesheet" href="stile.css" type="text/css">
8
9
10
11 </head>
12 <body>
13
14
15 Testo con colore giallo
16
17 <p>Paragrafo rosso</p>
18
19 <p class="blu">Testo con colore blu</p>
20
21
22
23 </body>
24 </html>
```

Come vedete al terzo paragrafo abbiamo applicato la classe usando l'attributo class con il simbolo = e tra apici il nome. Ora se ricaricate la pagina nel browser dovrete avere del testo giallo, poi un paragrafo rosso e uno solo blu, proprio quello su cui abbiamo applicato la classe. Sempre mantenendo questo esempio possiamo creare un'altra classe di nome paragrafo che possa applicare del testo di colore arancione. Provate a modificare il codice così

```
1  body{
2
3
4
5      color:yellow;
6
7
8  }
9
10
11  p{
12
13
14
15      color:red;
16
17
18
19  }
20
21
22  .blu{
23
24
25      color:blue;
26
27
28
29  }
30
31  #paragrafo{
32
33
34      color:orange;
35
36
37  }
```

Abbiamo creato una classe di nome paragrafo usando la seguente sintassi alternativa #nomeclasse ed abbiamo definito la proprietà del colore. Ora per applicare la classe seguite il codice seguente

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7      <link rel="stylesheet" href="stile.css" type="text/css">
8
9
10
11 </head>
12 <body>
13
14
15 Testo con colore giallo
16
17 <p>Paragrafo rosso</p>
18
19 <p class="blu">Testo <span id="paragrafo">con colore blu</span></p>
20
21
22
23
24 </body>
25 </html>
```

Come vedete abbiamo applicato la classe di nome blu ad una parte del testo e alla parte restante dello stesso la classe paragrafo richiamandola con la parola chiave id ed il nome della stessa ed usando il tag span che dovrete conoscere.

Background

Possiamo applicare un colore di sfondo per esempio alla pagina intera come ad un elemento <div> oppure a del testo. Scrivete questo codice

```
1  body{
2
3
4      background-color: aquamarine;
5
6
7  }
8
9  p{
10
11
12      background-color: yellow;
13
14
15
16  }
17
18
19  div{
20
21
22
23
24      width: 300px;
25      height: 300px;
26      border: 1px solid red;
27      background-color: grey;
28
29
30  }
```

Abbiamo applicato la proprietà background-color:aquamarine alla pagina intera. Il body si riferisce alla pagina intera. E questa proprietà specifica il colore di sfondo della pagina. Poi abbiamo definito un colore di sfondo per i contenitori div ed uno per i paragrafi. Modificate poi la pagina html così

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7      <link rel="stylesheet" href="stile.css" type="text/css">
8
9
10
11 </head>
12 <body>
13
14 <p>Paragrafo giallo</p>
15
16 <div>
17
18
19
20
21
22 </div>
23
24
25
26
27
28 </body>
29 </html>
```

Nella pagina html abbiamo inserito un paragrafo semplice che avrà un colore giallo, mentre abbiamo creato anche un div che sapete in html serve a creare un contenitore ed abbiamo applicato un bordo che vi invito a ricopiare uguale perchè lo spiegheremo in seguito. Ora se caricate la pagina dovreste avere lo sfondo aquamarine per la pagina, giallo per il paragrafo e grigio per il div. Abbiamo terminato con i colori di sfondo.

Bordi

Nell'esempio precedente abbiamo creato un contenitore div sul quale è stato applicato un colore di sfondo. Su di esso possiamo applicare dei bordi. Immaginiamo di voler creare più div ognuno con dei bordi diversi. Possiamo usare in questo caso le classi ed ecco un esempio

```
1  div{
2
3
4      border: 1px solid red;
5
6      width:300px;
7      height: 300px;
8      background-color: grey;
9
10
11
12 }
13
14
15 .div1{
16
17
18     border: 2px solid blue;
19
20     width:200px;
21     height: 200px;
22     background-color: red;
23
24
25
26
27
28
29 }
30
31
32 .div2{
33
```

```
34
35     border: 3px solid red;
36
37     width:100px;
38     height: 100px;
39     background-color: yellow;
40
41
42
43
44
45
46 }
```

Analizziamo il codice. Abbiamo stabilito una proprietà che è valida per ogni div della pagina e cioè **border** seguita da un valore numerico che rappresenta lo spessore del bordo poi lo stile solid (cioè colore pieno) ed il colore. Quindi scrivendo `border:1px solid red;` stiamo specificando un bordo da applicare al div con spessore 1 pixel e colore a tratto continuo rosso. Poi seguono le misure del div contenitore e cioè width che è la larghezza e height l'altezza sempre in pixels. Infine abbiamo specificato il colore di sfondo che già conosciamo come proprietà. Successivamente abbiamo creato due classi div1 e div2 con le stesse proprietà ma con valori diversi. Questo codice lo abbiamo usato nel file html che qui mostriamo

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7      <link rel="stylesheet" href="stile.css" type="text/css">
8
9
10
11 </head>
12 <body>
13
14
15 <div>
16
17
18 <h1>Div</h1>
19
```

```
20
21 </div>
22
23
24 <div class="div1">
25
26
27     <h1>Div1</h1>
28
29
30
31 </div>
32
33 <div class="div2">
34
35
36     <h1>div2</h1>
37
38
39
40 </div>
41
42
43
44 </body>
45 </html>
```

In pratica abbiamo creato tre div di cui il primo non ha classe mentre gli altri due ereditano le classi rispettive div1 e div2. Ora se aprite la pagina nel browser noterete tre div con un loro testo interno e sono diversi sia nel bordo, nella grandezza e nel colore di sfondo. Avete notato che abbiamo inserito un tag <h1> all'interno dei div. Voi nei div potete inserire qualsiasi cosa anche una immagine. Possiamo anche specificare un bordo diverso superiore, inferiore e laterale del nostro div e con colori diversi. Ecco come

```
1  div{
2
3
4
5
6      border-top: 2px solid blue;
7      border-left: 2px dotted red;
8      border-right: 2px dashed yellow;
9      border-bottom: 2px solid orange;
10
11     width: 300px;
12     height: 300px;
13     background-color: yellow;
14
15
16
17
18
19 }
```

Analizziamo il codice. Abbiamo specificato per qualsiasi div della pagina un bordo superiore (border-top) con pixel pari a 2 e di colore blue, a sinistra (border-left) di colore rosso e a puntini (dotted) e a destra di colore giallo e tratteggiato (dashed), ed infine inferiore (border-bottom) di colore arancione.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7      <link rel="stylesheet" href="stile.css" type="text/css">
8
9
10
11 </head>
12 <body>
13
14
15 <div>
16
17
18 <h1>Div</h1>
```

19

20

21 `</div>`

22

23

24

25 `</body>`26 `</html>`

I margini

Inserite nel file css il seguente codice che poi analizziamo

```
1  div{
2
3
4
5
6      border: 1px solid red;
7
8      width:300px;
9      height: 300px;
10     background-color: yellow;
11
12
13
14
15
16 }
17
18
19
20 .div-margin{
21     border: 1px solid red;
22
23     width:300px;
24     height: 300px;
25     background-color: yellow;
26
27     margin-left: 25px;
28
29
30
31
32
33 }
```

Abbiamo specificato per tutti i div della pagina larghezza ed altezza e un bordo di colore rosso e spessore 1 pixel. Mentre per i div che useranno la classe div-margin abbiamo aggiunto anche un

marginale sinistro con la proprietà `margin-left: 25 px;`. In pratica abbiamo specificato uno spazio di 25 pixel del div rispetto al margine sinistro della pagina. Ora modificate la pagina html così

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6
7      <link rel="stylesheet" href="stile.css" type="text/css">
8
9
10
11 </head>
12 <body>
13
14
15 <div>
16
17
18 <h1>Div</h1>
19
20
21 </div>
22
23
24
25 <div class="div-margin">
26
27     <h1>div2</h1>
28
29
30
31 </div>
```

Caricate la pagina e vedrete due div con bordo rosso ma il secondo che usa la classe `div-margin` dista rispetto al margine sinistro della pagina di 26 px. Notate che il secondo div risulta attaccato al primo. Ora modificate il css così

```
1  div{
2
3
4
5
6    border: 1px solid red;
7
8    width:300px;
9    height: 300px;
10   background-color: yellow;
11
12   margin-top:26px;
13   margin-left: 25px;
14
15   margin-bottom:15px;
16
17
18 }
19
20
21
22 .div-margin{
23   border: 1px solid red;
24
25   width:300px;
26   height: 300px;
27   background-color: yellow;
28
29   margin-left: 25px;
30
31
32
33
34
35 }
```

Notate che al div generale abbiamo aggiunto un margine superiore con la proprietà `margin-top:26px;` e un margine sinistro `margin-left:25px;` e un margine inferiore con `margin-bottom:15px;`. Ora stabilendo un margine bottom in pratica abbiamo specificato che vogliamo staccarci da qualsiasi elemento che si trova subito in fondo al nostro div. Se ricaricate la pagina noterete che adesso i due div con bordo rosso risultano staccati ugualmente dal margine sinistro, e il primo div in alto risulta staccato di 15 pixel rispetto al div in basso.