

Applicazioni Web 2013/14

Lezione 5: Cascading Style Sheets

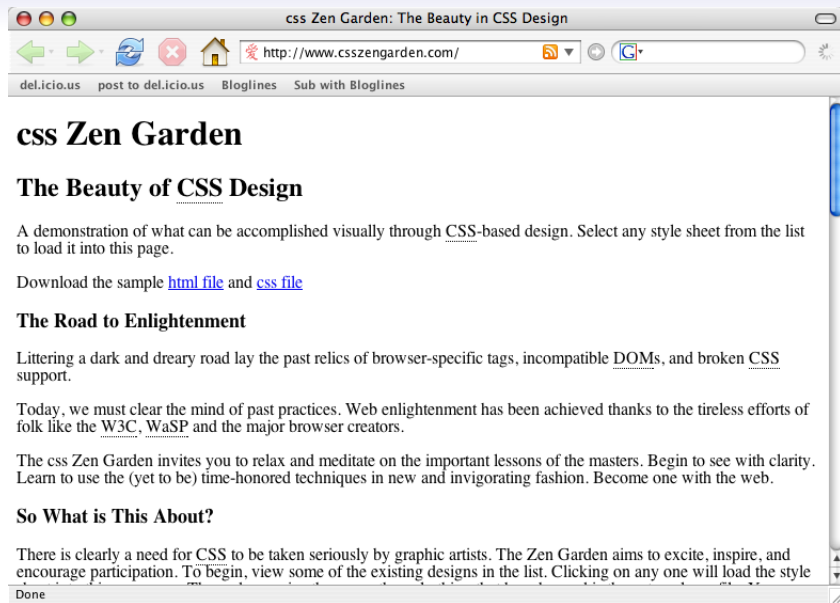
Matteo Vaccari

<http://matteo.vaccari.name/>
matteo.vaccari@uninsubria.it

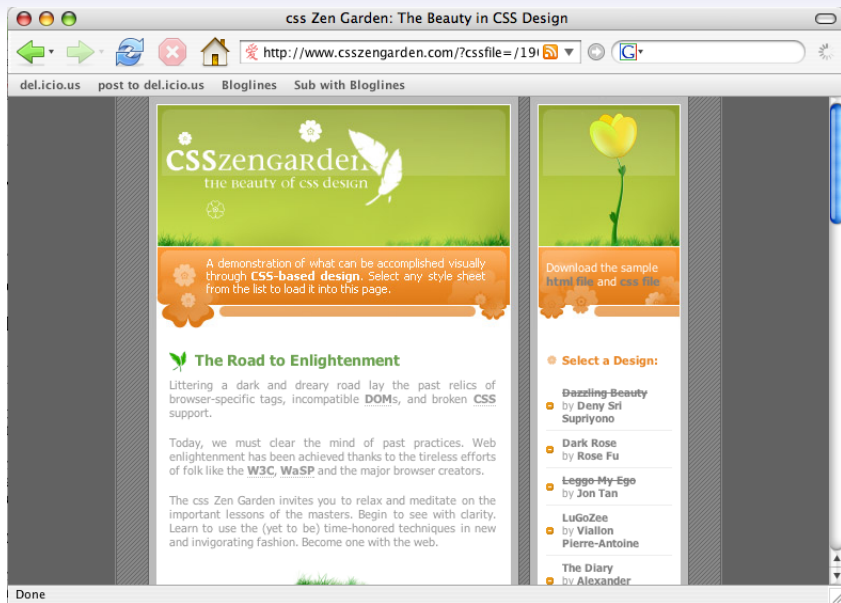


(cc) Matteo Vaccari. Published in Italy.
Attribution – Non commercial – Share alike 2.5









Cascading Style Sheets, (CSS)

Scopo: separare la *presentazione* dal *contenuto*

Standard W3C

Bandwidth friendly: riduce il “peso” dei doc html

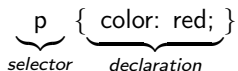
Facilita la condivisione di stili fra diversi documenti

Permette al grafico di modificare *totalmente* l'aspetto di un sito web senza toccare i documenti HTML

CSS

CSS: una lista di *regole*

Regola: *selector* seguito da una lista di *declaration*


p { *color: red;* }

Declaration: *property* ":" *value*

Significa: tutti i **p**aragrafi usino testo color rosso

Versioni alternative:

```
p { color: #ff0000; }
```

```
p { color: #f00; }
```

```
p { color: rgb(255, 0, 0); }
```

```
p { color: rgb(100%, 0%, 0%); }
```

Dichiarazioni multiple

```
p {  
  font-family: sans-serif;  
  font-size: 14px;  
  color: gray;  
  background: white;  
}
```


Valori *alternativi* e *generici*

```
p {  
  font-family: "Lucida Grande", Verdana, Lucida, Helvetica, sans-serif;  
}
```

Vengono usati nell'ordine: se non è disponibile il primo, si usa il secondo, ...

“sans-serif” è un generico font “senza bastoni” ed è *sempre* disponibile

Selettori multipli

```
h1, h2, h3, h4, h5, h6 {  
    font-family: Verdana, Arial, sans-serif;  
    font-weight: bold;  
}
```

equivale a

```
h1 {  
    font-family: Verdana, Arial, sans-serif;  
    font-weight: bold;  
}  
h2 {  
    font-family: Verdana, Arial, sans-serif;  
    font-weight: bold;  
}  
...
```

Le regole si “sommano”

```
h1 {  
  font-family: Verdana, Arial, sans-serif;  
  font-weight: bold;  
}
```

equivale a

```
h1 {  
  font-family: Verdana, Arial, sans-serif;  
}  
  
h1 {  
  font-weight: bold;  
}
```

Contextual Selectors (Descending Selectors)

```
em    { text-style: italic;    text-weight: normal; }  
em em { text-style: normal;    text-weight: bold;   }
```

Che significa?

Sono `corsivo grassetto di nuovo corsivo`
normale

Sono *corsivo* **grassetto** *di nuovo corsivo* normale

Commenti in CSS

I commenti in CSS sono come in C:

```
/*  
    Questo e' un commento per CSS  
*/
```

Come si “collegano” gli stili a un documento? (i)

Prima maniera: l'elemento *style*

```
<style type="text/css">  
  body {  
    font-family: Verdana, sans-serif;  
    font-size:   small;  
  }  
</style>
```

Nella *head* del documento

Come si “collegano” gli stili a un documento? (ii)

Seconda maniera: in un file esterno (lo *stylesheet*)

```
<link rel="stylesheet" href="style.css" type="text/css"
      media="screen" />
```

Deve stare nella *head*

Media: video `media="screen"`, `media="print"`, `media="all"`

Questa è la maniera **preferita**

Condividere gli stili fra più documenti

Risparmiare banda!

Come si “collegano” gli stili a un documento? (iii)

Terza maniera: l'*attributo* style

```
<p style="font-size: x-small; color: gray;">  
  Una notarella piccina piccina...  
</p>
```

Modifica **un** elemento

Da **evitare**; ci sono maniere più eleganti

id selectors

Se ho un elemento marcato con l'attributo *id*

```
<h1 id="titolone">Notizia bomba!</h1>
```

Posso scrivere regole *apposta per lui*

```
#titolone { font-size: xx-large; }
```

oppure

```
h1#titolone { font-size: xx-large; }
```

“#” indica un *id* selector

Class selectors

Se ho degli elementi marcati con l'attributo *class*

```
<li class="important">Diario delle lezioni</li>
```

Posso scrivere regole *apposta per loro*

```
.important { color: black; background: #f88; }
```

oppure

```
li.important { color: black; background: #f88; }
```

“.” indica un *class selector*

Pseudo-classi

```
a:link      {color:#FF0000;} /* unvisited link */
a:visited   {color:#00FF00;} /* visited link */
a:hover     {color:#FF00FF;} /* mouse over link */
a:active    {color:#0000FF;} /* selected link */

p:first-child {
  color:blue;
}
```

Ereditarietà

Una dichiarazione viene *ereditata* da tutti gli elementi contenuti:

```
body { font-family: Verdana; }
```

Significa “Verdana dappertutto”...

A meno che non ci sia un'altra regola *piú specifica*

```
h1 { font-family: Georgia; }
```

Tutte le proprietà ereditabili vengono ereditate

Cascade

Che succede quando due dichiarazioni sono in conflitto per un elemento?

- Importance
- Specificity
- Source order

Cascade

- Importance

1. User agent style sheets
2. Normal declarations in user style sheets
3. Normal declarations in author style sheets
4. Important declarations in author style sheets
5. Important declarations in user style sheets

- Specificity

- Source order

```
a {  
  text-decoration: none ! important;  
}
```

Cascade

- Importance
- Specificity
- Source order

Il selettore

`ul#navigation`

è più specifico di

`ul`

e anche di

`#navigation`

Cascade

- Importance
- Specificity
- Source order

```
p {  
  color: cyan;  
}  
p {  
  color: black; /* “vince” questa */  
}
```


Come specificare la dimensione del testo?

Font Size

- xx-small | x-small | small | medium | large | x-large | xx-large
- es. 12pt (punti tipografici: ok)
- es. 16px (pixel: non ok: dipende dalla risoluzione)
- es. 1.4em (1.4 volte la larghezza di una "M")
- relative size: larger | smaller
- percentage (in relation to parent element) es. 120%

Block o inline?

Gli elementi si dividono in due *categorie*:

- block
- inline

Se di tipo “block” allora ha un riquadro (box) che lo racchiude

Esempio: `<p> ... </p>`

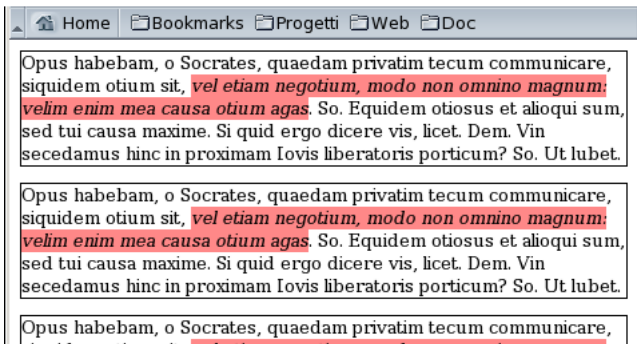
Se un elemento è di tipo “inline” allora si applica a una regione di testo *non riquadrata*

Esempio: ` ... `

Se ho una box, allora posso applicare *stili* alla box

Dimostrazione di block e inline

```
<style type="text/css">
  p { border: 1px solid black; }
  em { color: black; background: #f88; }
</style>
```



Yesterday the keynote was by Zed Shaw, the author of the Mongrel web server. His laptop wouldn't work with the beamer so he had to improvise without his slides—We did get to see the original presentation the next day. Anyway I was not disappointed: the topic of his talk was programming ethics, “by which I do not mean ‘do not throw Chinese dissidents in jail’...”. Rather, he presented us with three rules that he follows when he's coding.

margin

padding

content

border

The first rule is “You cannot communicate sufficiently”, which means that even if you put a lot of care into making your code clear, intention-revealing and understandable, it's still possible that your code will not be perceived as such by others. Just imagine what would happen then if you didn't put care into making it understandable!

Zed also said that Agile people tend to take the DRY principle too rigidly, coming up with very complex ways of removing duplication; yet it's much more important to communicate clearly what the code is doing than removing every bit of duplication. Sure, we should not copy-and-paste code; but a little duplication is better than complexity.

Box model

border: un *elemento grafico* che può essere stilizzato

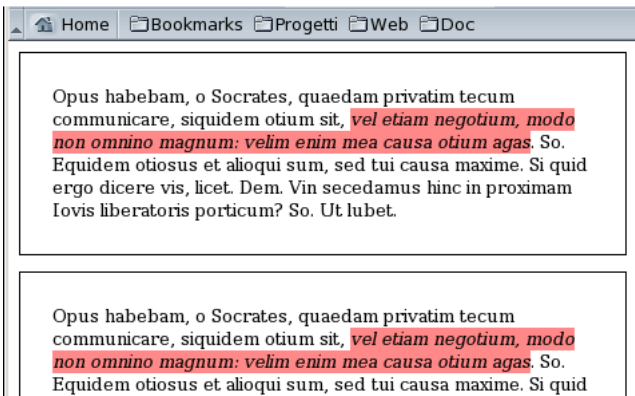
padding: *distanza* fra il contenuto e il *border*

margin: *distanza* fra un box e gli altri

Di una *distanza* posso stilizzare solo la *dimensione*...

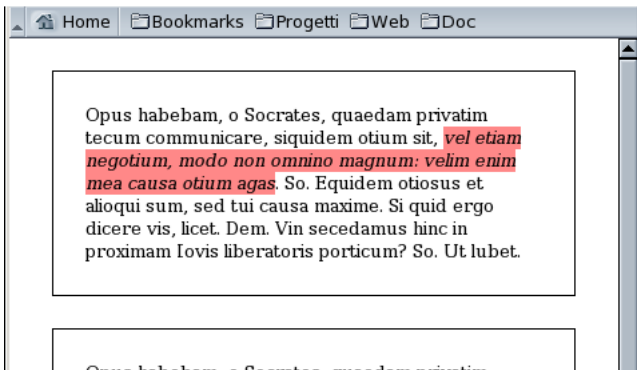
Aumentiamo il *padding*

```
<style type=text/css>
p {
  border: 1px solid black;
  padding: 2em;
}
em { color: black; background: #f88; }
</style>
```



Aumentiamo il *margin*

```
<style type=text/css>
p {
  border: 1px solid black;
  padding: 2em;
  margin: 2em;
}
em { color: black; background: #f88; }
</style>
```



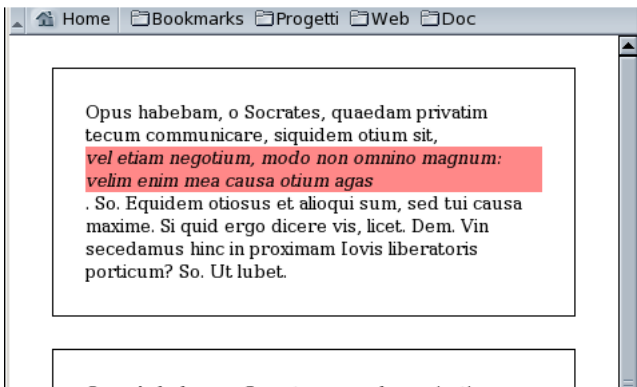
$$2 + 2 = 4?$$

I margini verticali *non si sommano*: si prende il max dei due.

I margini *orizzontali* invece si sommano

Display property

```
em {  
    color: black;  
    background: #f88;  
    display: block;  
}
```

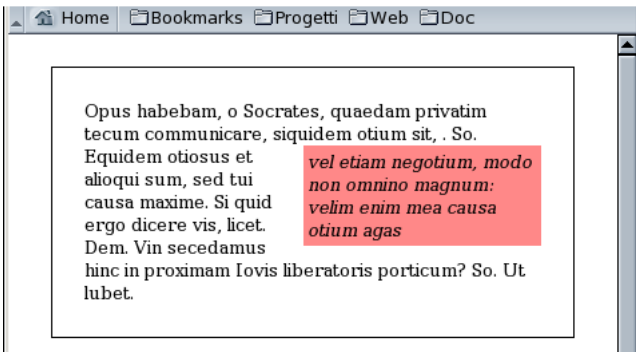


Display property

- **inline**: senza box
- **block**: con box
- **none**: l'elemento non viene mostrato

La proprietà *float*

```
em {  
    color: black; background: #f88;  
    display: block;  
    float: right;           /* fluttua a destra */  
    width: 50%;            /* restringiamola un po' */  
    padding: .3em;         /* diamo un po' di respiro */  
}
```



Span e Div

span: una *porzione* di testo

div: una *divisione* del documento

Gli elementi `` e `<div>` non hanno nessuna proprietà grafica

Tranne che `` è *inline* e `<div>` è *block*

La loro unica (e grande) utilità è che possono ricevere l'attributo *id* e/o l'attributo *class* a cui poi applicare stili

“Posizionare” gli elementi

Fixed: rispetto alla *finestra*

```
p.pos_fixed {  
  position: fixed;  
  top: 30px;  
  right: 5px;  
}
```

Relative: rispetto alla sua *posizione normale*

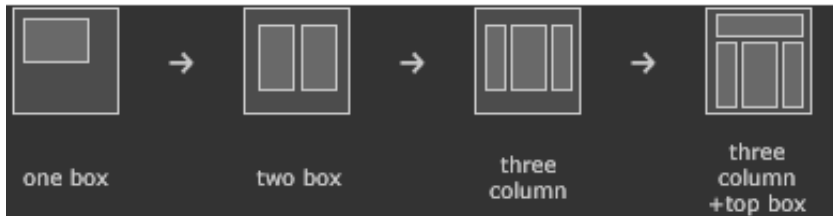
```
h2.pos_left {  
  position: relative;  
  left: -20px;  
}
```

Absolute: rispetto al primo genitore che ha una posizione non statica (o body)

```
h2 {  
  position: absolute;  
  left: 100px;  
  top: 150px;  
}
```

Page layout

Vogliamo progettare la “architettura” di una pagina (gabbia tipografica)



Due maniere:

- ibrido: tabella + CSS (antico e **deprecato**)
- puro: solo CSS

```
<html>
  <head>
    <title>Layout ibrido due colonne + header</title>
  </head>
  <body>
    <table border='1'>
      <tr>
        <td colspan='2'><p>This is the header</p></td>
      </tr>
      <tr valign='top'>
        <td width='30%'><p>This is the sidebar</p></td>
        <td>
          <p>This is the main content</p>
          <p>
            Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
            eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
          </p>
        </td>
      </tr>
    </table>
  </body>
</html>
```

E il risultato è...

This is the header	
This is the sidebar	<div>This is the main content</div> <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut</div>

Layout CSS con framework *Blueprint*

```
<html>
<head>
  <title>Prova</title>
  <link rel='stylesheet' href='css/blueprint/screen.css' type='text/css' media='screen, projection'>
  <link rel='stylesheet' href='css/blueprint/print.css' type='text/css' media='print'>
  <!--[if lt IE 8]><link rel='stylesheet' href='css/blueprint/ie.css' type='text/css' media='screen, projection'>
</head>
<body>
  <div class='container'>
    <div class='span-24 last'>
      <p>The <strong>Blueprint</strong> framework defaults to 950px wide content and 24 columns of 30px + 10px
    </div>
    <div class='span-6'>
      <p>
        This is a left sidebar
      </p>
    </div>
    <div class='span-18 last'>
      <p>This is the main content</p>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore e
      </p>
    </div>
  </div>
</body>
</html>
```

Layout CSS con framework *Blueprint*

The **Blueprint** framework defaults to 950px wide content and 24 columns of 30px + 10px margin

This is a left sidebar

This is the main content

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

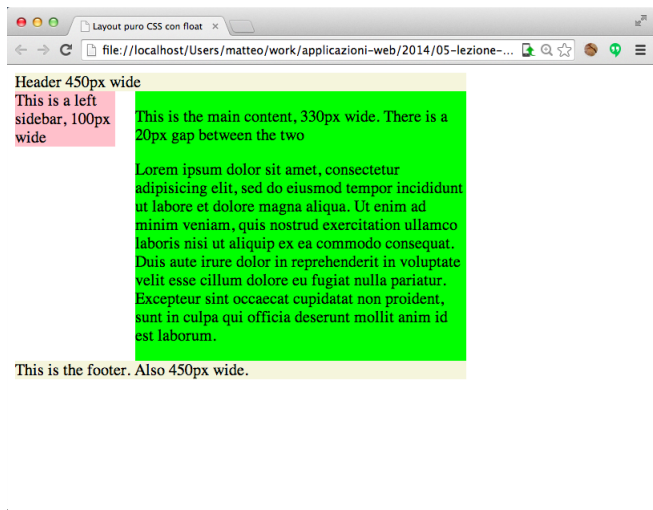
Funziona ... ma abbiamo descritto il layout nel markup HTML!!!

Layout puro CSS con *float*

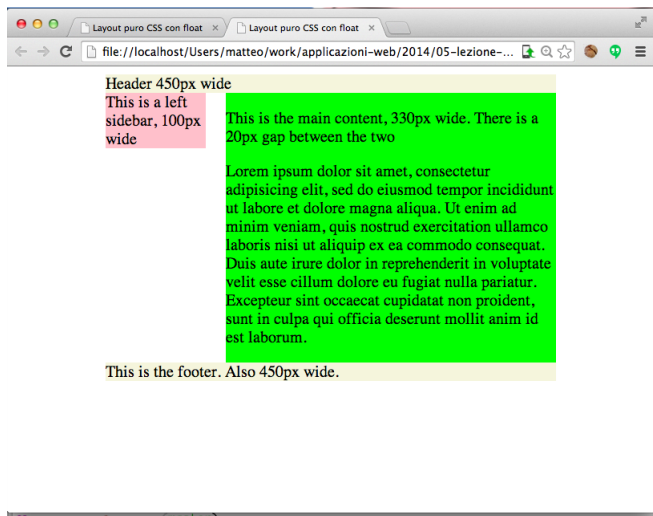
```
<html>
<head>
  <title>Layout puro CSS con float</title>
  <style type="text/css" media="screen">
    header {
      width: 450px;
      background-color: beige;
    }
    aside {
      width: 100px;
      float: left;
      margin-right: 20px;
      background-color: pink;
    }
    #main {
      width: 330px;
      float: left;
      margin-left: 0px;
      background-color: lime;
    }
    footer {
      width: 450px;
      clear: both;
      background-color: beige;
    }
  </style>
</head>
```

```
<body>
  <header>
    Header 450px wide
  </header>
  <aside>
    This is a left sidebar, 100px wide
  </aside>
  <section id='main'>
    <p>
      This is the main content, 330px wide.
      There is a 20px gap between the two
    </p>
    <p>
      Lorem ipsum dolor sit amet, ...
    </p>
  </section>
  <footer>
    This is the footer. Also 450px wide.
  </footer>
</body>
</html>
```

Layout puro CSS con *float*



Centrare orizzontalmente la pagina nella finestra



Centrare orizzontalmente la pagina nella finestra

```
<html>
<head>
  <title>Layout puro CSS con float</title>
  <style type="text/css" media="screen">
    #page {
      width: 450px;
      margin: auto;
      padding: 0px;
    }
    /* ... */
  </style>
</head>

  <body>
    <div id="page">
      ...
    </div>
  </body>
</html>
```

Esercizio: riproduci questo design

Romanzi e racconti

Romanzi e racconti è semplicemente una base dati di romanzi e racconti. Puoi aggiungere i libri che ti piacciono, recensirli, etichettarli e condividerli. E puoi andare a vedere che cosa piace agli altri.

chi sei?

[forgot password?](#)

registrati!

☐ I agree to the Terms of Use.

Scopri

Cerca nella base dati per titolo, autore...

Esplora

[Libri aggiornati di recente](#) | [Tag popolari](#) | [Libri consigliati](#)

Tom De Marco, Timothy Lister: *Peopeware*

scifi, action, sci-fi, lucas, great, space, starwars, adventure, boring, sequel, ...

in 16 lo raccomandano

in 26 lo hanno letto

Douglas Coupland, Eleanor Rigby

drugs, drama, weird, addiction, love, drug, ignorant, bliss, tragic, morale, ...

in 16 lo raccomandano

in 26 lo hanno letto

J.R.R. Tolkien, *Il signore degli anelli*

drama, prison, thriller, quality, over-rated, injustice, added, jailbreak, this, escape, ...

in 16 lo raccomandano

in 26 lo hanno letto

C. S. Lewis, *Lontano dal pianeta silenzioso*

pirates, action, adventure, comedy, depp, fantasy, pirate, curse, jack, caribbean, ...

in 116 lo raccomandano

in 216 lo hanno letto