

INTRODUZIONE AI CSS

giorgio.beggiora@artigianidelweb.com



giorgiobeggiora

CANSEI DE SER SEXY

SONO UN GRUPPO MUSICALE
BRASILIANO DI SAN PAOLO
PRODOTTO DALLA SUB POP

<https://www.youtube.com/watch?v=7agPOt1XZz8>



CSS

Quel linguaggio che serve a fare la grafica

Giorgio Beggiora - 2001

CASCADING STYLE SHEETS

CSS specifica uno schema di priorità per determinare quali regole di stile si applicano, se più di una regola corrisponde ad un particolare elemento. In questa cosiddetta "cascata", priorità o pesi sono calcolati e assegnati alle regole, in modo che i risultati siano prevedibili.

PERCHÈ USARE I CSS?

- Consentono di separare la grafica dal contenuto e dalla logica
- Sono scalabili e riutilizzabili da più pagine
- Sono molto più veloci e performanti di HTML e JavaScript
- La loro efficienza può migliorare l'esperienza mobile

COME FUNZIONANO I CSS

SEMPLICE PAGINA HTML

```
<!doctype html>
<html lang="it">
  <head>
    <meta charset="utf-8" />
    <title>Demo CSS</title>
  </head>
  <body>
    <ul>
      <li>uno</li>
      <li>due</li>
      <li>tre</li>
      <li>quattro</li>
    </ul>
  </body>
</html>
```

COME APPLICARE LE REGOLE

- Inline

```
<a href="/" style="color:lime;background:gray;">
  Home
</a>
```

- tag <style> (preferibilmente nel tag <head>)

```
<style>
  body{margin: 0; background-color: silver;}
  a{text-decoration: none;}
</style>
```

- tag <link> (preferibilmente nel tag <head>)

```
<link rel="stylesheet" href="/path/to/file.css">
```

- @import (come prima istruzione di un file css)

```
@import "/path/to/file.css";
```


SELETTORI

NOME DEL TAG

```
<style>
span{color: red;}
</style>
```

```
<ul>
  <li>uno</li>
  <li><span>due</span></li>
  <li>tre</li>
  <li><span>quattro</span></li>
</ul>
```

- uno
- **due**
- tre
- **quattro**

CLASSE

```
<style>
.evidenza{color: red;}
</style>
```

```
<ul>
  <li class="wp">uno</li>
  <li class="wp evidenza"><span>due</span></li>
  <li class="wp">tre</li>
  <li class="wp evidenza"><span>quattro</span></li>
</ul>
```

- uno
- **due**
- tre
- **quattro**

ID

```
<style>
#evidenza{color: red;}
</style>
```

```
<ul>
  <li>uno</li>
  <li><span>due</span></li>
  <li>tre</li>
  <li id="evidenza"><span>quattro</span></li>
</ul>
```

- uno
- due
- tre
- quattro

ATTRIBUTO

```
<style>
[title="Numero 2"]{color: red;}
</style>
```

```
<ul>
  <li title="Numero 1">uno</li>
  <li title="Numero 2"><span>due</span></li>
  <li title="Numero 3">tre</li>
  <li title="Numero 4"><span>quattro</span></li>
</ul>
```

- uno
- **due**
- tre
- quattro

PSEUDO-CLASSI

POSIZIONE

```
<style>
li:nth-child(3){color: red;}
</style>
```

```
<ul>
  <li>uno</li>
  <li><span>due</span></li>
  <li>tre</li>
  <li><span>quattro</span></li>
</ul>
```

- uno
- due
- tre
- quattro

STATO

```
<style>
li:hover{color: red;}
</style>
```

```
<ul>
  <li>uno</li>
  <li><span>due</span></li>
  <li>tre</li>
  <li><span>quattro</span></li>
</ul>
```

- uno
- due
- tre
- quattro

ALTRI ESEMPI

:empty

:checked

:enabled

:disabled

:focus

:active

:link

:visited

LA CASCATA

PARENTELA

```
<style>
li > .evidenza{color: red;}
</style>
```

```
<span>
  <ul>
    <li class="evidenza">uno</li>
    <li><span class="evidenza">due</span></li>
    <li>tre</li>
    <li>quattro</li>
  </ul>
</span>
```

- uno
- **due**
- tre
- quattro

VINCE IL PIU' SPECIFICO

```
<style>
span{color: yellow;}
span{color: green;}
ul li span{color: purple;}
ul li:nth-child(2) span{color: lime;}
ul span{color: red !important;}
</style>
```

```
<span>
  <ul>
    <li>uno</li>
    <li><span>due</span></li>
    <li>tre</li>
    <li><span>quattro</span></li>
  </ul>
</span>
```

- uno
- **due**
- tre
- **quattro**

CONDIVISO

```
<style>
.evidenza, .errore{color: red;}
</style>
```

```
<span>
  <ul>
    <li>uno</li>
    <li><span class="evidenza">due</span></li>
    <li>tre</li>
    <li><span class="errore">quattro</span></li>
  </ul>
</span>
```

- uno
- **due**
- tre
- **quattro**

UN ESEMPIO PIÙ COMPLESSO

```
<style>
#lista .evidenza span:first-child{color: red;}
</style>
```

```
<span>
  <ul id="lista">
    <li>uno</li>
    <li class="evidenza">
      <span>due</span>
      <span>bis</span>
    </li>
    <li>tre</li>
    <li><span>quattro</span></li>
  </ul>
</span>
```

- uno
- **due** bis
- tre
- quattro

SENZA SPAZI = NELLO STESSO TAG

```
<style>
span#due.evidenza{color: red;}
</style>
```

```
<span>
  <ul id="lista">
    <li class="evidenza">uno</li>
    <li class="evidenza">
      <span id="due" class="evidenza">due</span>
    </li>
    <li class="evidenza">tre</li>
    <li class="evidenza"><span>quattro</span></li>
  </ul>
</span>
```

- uno
- **due**
- tre
- quattro

CI VEDIAMO SU:

- MDN
developer.mozilla.org
- CSS-TRICKS
css-tricks.com
- CAN I USE
caniuse.com
- ...e su quel forum
che non serve neanche
nominare...

CSS E JS

CAMEL SYNTAX

```
<script>
var element = document.getElementById('contenuto');
element.style.fontSize = '12px';
element.className = 'hello ciao salut hola';
element.setAttribute('style', 'font-weight: bold; color: red;');
var style = window.getComputedStyle(element);
</script>
```


COMPATIBILITÀ BROWSER

Una volta c'erano gli hack, filter, conditional comments.

Oggi ci sono i vendor prefix:

```
<style>  
-webkit-opacity: 0.7;  
-moz-opacity: 0.7;  
-ms-opacity: 0.7;  
opacity: 0.7;  
</style>
```

Se un browser non supporta una funzione si tende a non implementarla: è veramente così importante che un sito si veda identico su tutti i browser, a scapito della velocità?

CSS NELLE EMAIL

BACK TO THE FUTURE

Siamo fermi da 15 anni

USARE I FONT

WOFF

@font-face

LAYOUT

BOX MODEL

```
div {  
  width: 100px;  
  height: 100px;  
  padding: 10px;  
  border: 1px;  
  margin: 10px;  
}
```

```
html{  
  box-sizing: border-box;  
}  
*, *:before, *:after{  
  box-sizing: inherit;  
}
```

FLOAT LAYOUT

```
div {  
  display: block;  
  float: left;  
}
```

FLEXIBLE LAYOUT

```
div {  
  display: flex;  
}
```

TABLE LAYOUT

```
div {  
  display: table;  
}
```

RESPONSIVE

```
@media (max-width: 600px) {  
  .sidebar {  
    display: none;  
  }  
}
```

TRANSIZIONI E ANIMAZIONI

```
a:hover{  
  color: red;  
  transition-delay: 0.6s;  
}
```

```
p {  
  animation-duration: 3s;  
  animation-name: slidein;  
}  
@keyframes slidein {  
  from {  
    margin-left: 100%;  
    width: 300%;  
  }  
  to {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```


LESS & SASS

PREPROCESSORI

Servono per sopperire alle mancanze di CSS se paragonato ad un vero linguaggio di programmazione.

Tipicamente si scrive codice simil-css mentre un software monitora la cartella che contiene i file e ad ogni salvataggio li elabora per produrre file css validi.

LESS e SASS sono due sintassi diverse.

VANTAGGI

<http://sass-lang.com/guide>

- Variable
- Nesting
- Import & Partial
- Mixin (functions)
- Extend (inheritance)
- Operation, Logic, Loop
- Utility (funzioni, nuovi selettori, data type, ecc)

SVANTAGGI

- Serve un software che crei i file css
- È necessario mantenere i file sorgenti

DIFFERENZE

<https://css-tricks.com/sass-vs-less/>

- Sono parecchio "tecniche". Si dice che...
- LESS sia leggermente più facile da imparare
- LESS sia più veloce nella compilazione
- SASS sia più strutturato
- Il mio consiglio per chi comincia è...
- SASS!

SEMPLICI PREPROCESSORI GRATUITI ED OPEN-SOURCE

windows, linux, mac

- Koala - <http://koala-app.com/>
Less, Sass, Compass, CoffeeScript
- Scout - <http://scout-app.io/>
Sass



GRAZIE

giorgio.beggiora@artigianidelweb.com



[giorgiobeggiora](#)