



Web Design

Matteo Baccan



CSS

CSS

Cascading Style Sheets : in italiano fogli di stile a cascata

È un linguaggio usato per definire la formattazione di documenti

Descrive lo stile degli elementi di pagina

La sua interpretazione necessita di un browser

CSS – strumenti

Lo strumento che useremo durante il corso è

<https://codepen.io>

CodePen is a social development environment. At its heart, it allows you to write code in the browser, and see the results of it as you build. A useful and liberating online code editor for developers of any skill, and particularly empowering for people learning to code. We focus primarily on front-end languages like HTML, CSS, JavaScript, and preprocessing syntaxes that turn into those things.

Iscrivetevi e seguite il profilo creato apposta per il corso

<https://codepen.io/matteobaccan>

CSS – strumenti

Editor

Codepen.io

Notepad

Notepad++

VisualStudio Code

Va bene qualsiasi editor, non visuale, meglio se con syntax highlighter e code completion

Le slide e i sorgenti del corso, liberamente ispirati a <https://www.w3schools.com> e costantemente aggiornati, sono disponibili a questo indirizzo

<https://github.com/matteobacchan/CorsoCSS>

CSS – esempio

```
body {  
  background-color: red;  
}
```

```
h1 {  
  color: black;  
  text-align: center;  
}
```

```
p {  
  font-family: courier;  
  font-size: 24px;  
}
```

CSS

Cos'è il CSS?

CSS è acronimo di **Cascading Style Sheets**, sono fogli che vengono utilizzati per formattare le pagine web.

Con i CSS è possibile controllare il colore, il carattere, la dimensione del testo, la spaziatura tra gli elementi, il modo in cui gli elementi sono posizionati e disposti, quali immagini di sfondo o colori di sfondo devono essere utilizzati, o le diverse visualizzazioni in base alle dimensioni dello schermo

Da notare che **cascading** identifica il fatto che uno stile applicato a un elemento padre si applicherà anche a tutti gli elementi figli all'interno dell'elemento padre

CSS

I CSS possono essere aggiunti ai documenti HTML in 3 modi:

Inline - utilizzando l'attributo `style` all'interno degli elementi HTML

Interno - utilizzando un elemento `<style>` nella sezione `<head>`

Esterno: utilizzando un elemento `<link>` per collegarsi a un file CSS esterno

CSS Inline

`<h1 style="color:blue;">Una intestazione blue</h1>`

`<p style="color:red;">Un paragrafo rosso</p>`

CSS Interno

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {background-color: powderblue;}
      h1  {color: blue;}
      p   {color: red;}
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

CSS Esterno

```
<!DOCTYPE html>  
<html>
```

```
  <head>  
    <link rel="stylesheet" href="styles.css">  
  </head>
```

```
  <body>  
    <h1>Intestazione</h1>  
    <p>Paragrafo</p>  
  </body>
```

```
</html>
```

CSS styles.css

```
body {  
  background-color: powderblue;  
}
```

```
h1 {  
  color: blue;  
}
```

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

CSS sintassi

La sintassi di base dei CSS è data dalla dichiarazione di un **selector**, seguito da una lista di **proprietà** e del loro relativo **valore**

```
selector {  
  proprietà: valore;  
}
```

In questo caso il selettore è **body** e la proprietà impostata è **background-color** seguita dal suo valore **red**

```
body {  
  background-color: red;  
}
```

CSS selector

I **selector** permettono di identificare in modo preciso gli elementi HTML che vogliamo personalizzare.

Esistono 5 categorie diverse di **selector**

selector semplici : selezionano gli elementi in base a nome, id, classe

selector combinatori o di relazione : selezionano gli elementi in base alla loro relazione

selector di pseudo-classe : selezionano gli elementi in base a uno stato

selector di pseudo-elementi : selezionano e definiscono lo stile di una parte di un elemento

selector di attributo : selezionano gli elementi in base a un attributo o al valore di un attributo

CSS selector semplici

I **selector** semplici selezionano gli elementi in base a nome, id, classe

```
tag {  
  color: green;  
}  
#idtag {  
  color: red;  
}  
.classenome {  
  color: magenta;  
}  
tag.classenome {  
  color: magenta;  
}
```

CSS selector universale

Per convenzione esiste il selector `*` che indica che le proprietà indicate devono essere applicate a qualsiasi **tag**

```
* {  
  color: green;  
}
```

In questo modo, qualsiasi elemento contenuto in pagina, avrà una colorazione di default impostata sul verde

CSS selector raggruppamenti

Per ridurre la prolissità dei CSS è stata introdotta la sintassi per raggruppamento che permette di mettere, in un'unica dichiarazione, più direttive CSS. Per questo motivo scrivere

```
h1 {  
  color: green;  
}  
h2 {  
  color: green;  
}
```

Equivale a scrivere

```
h1, h2 {  
  color: green;  
}
```


CSS selector combinator

I **selector combinatori** : selezionano gli elementi in base alla loro relazione. Per determinare la relazione viene usato un **combinator**. I combinator possono essere di 4 tipi

discendente (spazio)

figlio (>)

fratelli adiacenti (+)

fratelli generali (~)

CSS selector combinator

discendente (spazio)

`div p { background-color: red; }`

figlio (>)

`div > p { color: white; }`

fratelli adiacenti (+)

`h3 + span { color: white; }`

fratelli generali (~)

`h2 ~ h3 { border: 1px solid black; }`

CSS selector pseudo classe

Una pseudo classe identifica uno stato speciale di un tag.

La sintassi di utilizzo è simile alla sintassi base, con l'aggiunta di : e il tipo di pseudoclasse

```
selector:pseudoclasse {  
  proprietà: valore;  
}
```

CSS selector pseudo classe

Esistono una trentina di pseudoclassi.

Di seguito alcune classi

:hover è attiva quando il puntatore del mouse è sopra l'elemento

:first-child è il primo elemento di una lista

:focus un input che riceve il fuoco

:read-only un input con l'attributo **readonly**

Sul sito dei developer Mozilla è possibile averne un elenco completo

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

CSS selector pseudo elemento

Un pseudo elemento viene utilizzato per applicare uno stile a una parte specifica di un elemento.

La sintassi di utilizzo è simile alle pseudo classi, con l'aggiunta di un doppio : e il tipo di pseudo elemento

```
selector::pseudoelemento {  
  proprietà: valore;  
}
```

CSS selector pseudo elemento

Un pseudo elemento viene utilizzato per applicare uno stile a una parte specifica di un elemento.

::after prima dell'elemento

::before dopo l'elemento

CSS attribute selector

Tramite i selector di attributi è possibile applicare uno stile agli elementi HTML che hanno attributi o valori di attributo specifici.

```
selector[attributo] {  
  proprietà: valore;  
}
```

```
selector[attributo=valore] {  
  proprietà: valore;  
}
```

CSS attribute selector

Seleziono solo il tag **p** che ha un attributo chiamato **test1**

```
p[test1] {  
  color: red;  
}
```

Seleziono solo il tag **p** che ha un attributo chiamato **test2** col valore **pippo**

```
p[test2=pippo] {  
  color: green;  
}
```


CSS attribute selector

Seleziono un attributo che contiene la parola valore

```
selector[attributo~=valore] {  
  proprietà: valore;  
}
```

Seleziono un attributo che inizia per valore (uguale o uguale seguito dal trattino)

```
selector[attributo|=valore] {  
  proprietà: valore;  
}
```

CSS attribute selector

Seleziono un attributo che inizia valore

```
selector[attributo^=valore] {  
  proprietà: valore;  
}
```

Seleziono un attributo che finisce per valore

```
selector[attributo$=valore] {  
  proprietà: valore;  
}
```

CSS attribute selector

Seleziono un attributo che contiene la sottostringa valore

```
selector[attributo*=valore] {  
  proprietà: valore;  
}
```