



corso di

WEB DEVELOPER

CSS di Base

IL CSS è uno strumento che serve per definire la parte grafica di una pagina web: l'idea di base, è quella di mostrare una parte "abbellita" di esso, ed è un meccanismo per stilizzare i documenti, per far fronte alle esigenze di utenti e web designer. I fogli di stile, indicano quindi come mostrare i documenti attraverso schermi e stampanti, definendo delle serie di proprietà per ogni elemento HTML, e dette proprietà, tecnicamente, devono avere un nome e un valore, separati da due punti (:) e devono essere separate da dei punti e virgola (;).

Esistono due tipologie di approcci ai fogli di stile CSS:

APPROCCI DIVERSI: FOGLIO DI STILE ESTERNO VS. FOGLIO DI STILE INTERNO

- Foglio di stile *esterno*: definisce le regole di stile in un file .css che verrà incluso nel documento HTML tramite il tag HTML `<link>`;
- Foglio di stile *interno*: definisce le regole di stile nella sezione header nel documento HTML utilizzando il tag `<style>`.



Figura 1 - gli approcci al foglio di stile

Vediamo ora due esempi, partendo dal foglio di stile esterno:

ESEMPIO FOGLIO DI STILE ESTERNO

Consideriamo di avere un file .css con le seguenti regole:

```
external.css
1 .red{
2   color:red;
3 }
4
5 .thick{
6   font-size: 28px;
7 }
8
9 .green{
10  color:green;
11 }
12
```



Figura 2 - foglio di stile esterno, parte 1

ESEMPIO FOGLIO DI STILE ESTERNO

```

<!DOCTYPE html>
<html>
  <head>
    <title>Foglio di stile esterno</title>
    <link href="style.css" type="text/css" rel="stylesheet"/>
  </head>
  <body>
    <p>Il risultato è uguale a quello dell'esempio con il foglio di stile esterno</p>
    <p class="red">Questo è red nel file css</p>
    <p class="thick">Questo è .thick nel file css</p>
    <p class="green">Questo è green nel file css</p>
    <p class="thick green">Questo è .thick e .green</p>
  </body>
</html>

```

Questo è red nel file css
Questo è .thick nel file css
Questo è green nel file css
Questo è .thick e .green



Figura 3 - foglio di stile esterno, parte 2

Nell'esempio del foglio di stile interno, il risultato è lo stesso della slide precedente:

ESEMPIO FOGLIO DI STILE INTERNO

Il risultato è uguale a quello dell'esempio con il foglio di stile esterno.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Foglio di stile interno</title>
    <style type="text/css">
      .red {
        color: red;
      }
      .thick {
        font-size: 24px;
      }
      .green {
        color: green;
      }
    </style>
  </head>
  <body>
    <p>Il risultato è uguale a quello dell'esempio con il foglio di stile esterno</p>
    <p class="red">Questo è red nel file css</p>
    <p class="thick">Questo è .thick nel file css</p>
    <p class="green">Questo è green nel file css</p>
    <p class="thick green">Questo è .thick e .green</p>
  </body>
</html>

```



Figura 4 - foglio di stile interno

Nei diversi approcci da adottare, dobbiamo distinguere quello **inline** da quello **incorporato**:

APPROCCI DIVERSI: INLINE VS. STILE INCORPORATO

- *Inline*, definisce le regole di stile all'interno dell'elemento HTML utilizzando l'attributo `style`. Questo tipo dovrebbe essere utilizzato quando si ha la necessità di cambiare lo stile di un singolo elemento HTML. Le regole definite con il metodo inline sovrascrivono sia le regole definite in CSS esterni, sia quelle definite nell'elemento `<style>` dell'header.
- *Stile incorporato*, quello che abbiamo già visto con il metodo del foglio di stile interno ed esterno: tutto in un unico blocco.



Figura 5 - inline vs incorporato

Infine, eccoci arrivati ad un elemento fondamentale in CSS, ovvero i **selettori**. Esistono quattro tipi di selettori:

I SELETTORI

- **Selettore universale**, si indica con l'asterisco (*); le seguenti regole vanno applicate a tutti gli elementi;
- **Selettore di tipo**, ad esempio richiamando il nome del tag;
- **Selettore di classe**, tramite l'attributo `class`: ad esempio `<p class=«esempio»>...</p>`
- **Selettore di ID**: permettono di assegnare una regola CSS ad un unico elemento nella pagina HTML.



Figura 6 - i selettori

Dopo questa carrellata veloce, passiamo al vero e proprio CSS, partendo dai quattro elementi seguenti: **margin, border, padding e content**.

- **Margin** viene usato per creare spazio attorno agli elementi, al di fuori di ogni confine definito;
- **Border**, invece, permette di specificare lo stile, la larghezza e il colore del bordo dell'elemento;

- **Padding** è usato per creare spazio al di fuori del contenuto dell'elemento, all'interno di ogni bordo definito;
- **Content** viene usato con gli pseudo elementi `::before` e `::after` per inserire il contenuto generato.

Vediamoli nel dettaglio:

MARGIN

L'elemento Margin, prende in considerazione:

- `margin-top`, per il margine superiore;
- `margin-right`, per il margine destro;
- `margin-bottom`, per il margine inferiore;
- `margin-left`, per il margine sinistro.



Figura 7 – margin

BORDER

L'elemento border, ha tre attributi principali:

- `border-width`, specifica lo spessore dei quattro bordi;
- `border-color`, per impostare il colore dei quattro bordi
- `border-style`, specifica il tipo di display da visualizzare



Figura 8 – border

PADDING

L'elemento `border`, specifica il padding per ogni lato dell'elemento;

- *padding-top*, per il padding superiore;
- *padding-right*, per il padding destro;
- *padding-bottom*, per il padding inferiore;
- *padding-left*, per il padding sinistro.

Figura 9 – padding



CONTENT

L'elemento `border`, utilizza due diversi attributi:

- *::before*, il quale inserisce del contenuto prima di un elemento HTML;
- *::after*, il quale inserisce del contenuto dopo un elemento HTML.

Figura 10 – content



Altri elementi molto importanti in CSS sono il **colore**, la **posizione**, il **background** e il **font**:

- Il **colore**, che può essere specificato in vari modi: esadecimali, RGB, HSL, o con nomi predefiniti;
- La **posizione**, che assume il valore static (default), absolute, fixed, relative, ed altri ancora;
- Il **background**, che si occupa dello sfondo degli elementi HTML;
- Il **font**, che identifica stile, variante, tipologia di carattere e famiglia.

Infine, una piccola carrellata al programma **Bootstrap**: è un framework di sviluppo gratuito e open source, che serve a creare siti web responsive e mobile-first, quindi ottimizzati per i cellulari, attraverso il CSS e Javascript.



www.miacademy.it

info@miacademy.it



seguici su facebook
[@miacademyformazione](https://www.facebook.com/miacademyformazione)