

I Fogli di Stile (CSS 3)

Trattasi di semplici documenti modificabili anche con un editor di testo ai quali si assegna l'estensione **.CSS**.

Abbellire le proprie pagine utilizzando istruzioni all'interno del file HTML è una cosa, realizzare pagine di stile invece esterne è tutt'altra cosa.

Un foglio di stile si dice **interno** quando il suo codice è compreso in quello del documento HTML. A seconda che si lavori con un CSS esterno o interno variano sintassi e modalità di inserimento. Rispetto a queste diverse modalità si parla di fogli di stile **collegati**, **incorporati** o **in linea**.

Collegato

Per caricare un foglio di stile esterno in un documento esistono due possibilità. La prima è quella che fa uso dell'elemento HTML **<link>**. La dichiarazione va sempre collocata **all'interno della sezione <head>** del documento HTML:

```
<html>
<head>
  <link href="stile.css" rel="stylesheet" type="text/css">
</head>
<body>
  [...]
</html>
```

Potremo usare anche la direttiva **@import** all'interno di **<style>** :

```
<html>
<head>
  <style>
    @import url(stile.css);
  </style>
</head>
<body>
  [...]
</html>
```

Lo stile delle pagine WEB

Ricordarsi in questo caso di chiudere l'istruzione con un punto e virgola.

Incorporato

Infine un altro modo è quello dei fogli incorporati dove le istruzioni sono inserite direttamente nel documento HTML tramite il tag **<style>**.

```
<html>
  <head>
    <style type="text/css">
      body {background: white;}
      p {color: black;}
      [...]
    </style>
  </head>
  <body>
    [...]
  </body>
</html>
```

Come si vede, la parte di codice che ci interessa inizia con l'apertura del tag **<style>**.

In Linea

L'ultimo modo per formattare un elemento con i CSS consiste nell'uso dell'attributo HTML **style**. Esso fa parte della collezione di attributi HTML definiti **globali**: si tratta di quegli attributi applicabili a tutti gli elementi. La dichiarazione avviene a livello dei singoli tag contenuti nella pagina e per questo si parla di fogli di stile in linea.

```
<elemento style="regole_di_stile">
```

Se, ad esempio, si vuole formattare un titolo h1 in modo che abbia il testo di colore rosso e lo sfondo nero, scriveremo:

```
<h1 style="color: red; background: black;">...</h1>
```

NOTA IMPORTANTE

Le cose da osservare nel codice sono due. Come valore agli style si possono dichiarare più regole di stile, separate dal punto e virgola. I due punti si usano invece per introdurre il valore della proprietà da impostare, esattamente come si fa con i CSS esterni e incorporati.

SUGGERIMENTO

Usare sempre fogli di stili esterni, ricorrere a CSS incorporati per particolari esigenze di formattazione su elementi specifici di una certa pagina.

Struttura di un istruzione CSS

La parola parla da sé: si tratta di una semplice dichiarazione che serve a **selezionare** la parte o le parti di un documento soggette ad una specifica regola.



Selettore universale

Anche nei CSS abbiamo un jolly. Il **selettore universale** serve infatti a selezionare **tutti gli elementi di un documento**. Si esprime con il carattere * (asterisco).

```
* {color: red;}
```

La regola assegna il colore rosso a tutti i caratteri/elementi della pagina.

Selettore di tipo (o selettore di elementi)

È il più semplice dei selettori. È costituito dal nome di uno specifico elemento HTML. Serve a selezionare tutti gli elementi di quel tipo presenti in un documento.

```
h1 {color: green;}  
p {background-color: yellow;}
```

Raggruppare i selettori

È possibile nei CSS raggruppare diversi selettori al fine di semplificare il codice e per poterlo fare essi vanno separati da una **virgola**.

Il raggruppamento è un'operazione molto conveniente. Pensate a questo scenario:

```
h1 {background: white;}  
h2 {background: white;}  
h3 {background: white;}
```

Tutti e tre gli elementi hanno uno sfondo bianco quindi potremmo scrivere così:

```
h1, h2, h3 {background: white;}
```

La regola per il raggruppamento vale per qualsiasi tipologia di selettori anche tra tipi diversi.

Impostare il body

Il primo passo nella realizzazione del foglio di stile per la stampa è quello di dare un'impostazione preliminare al body, impostando un font adatto ed eliminare eventuali background. E' quindi altamente consigliato utilizzare un font 'serif' per testi quindi dobbiamo scegliere un carattere come Times, Georgia, etc:

```
body {  
  background: white;  
  color: black;  
  font-family: "Times New Roman", Times, serif;  
  font-size: 12px;  
}
```

Anche la dimensione è importante infatti in fase di stampa una dimensione di 12 pixel è sufficiente, mentre dimensioni come 14 o 16 pixel potrebbero risultare difficili da leggere quando i testi sono lunghi.

CLASS

Per assegnare gli attributi a gruppi di oggetti si utilizza la notazione **class**

```
.centro { text-align:center; color:red; }
```

dove centro verrà assegnato alla variabile **class** all'interno del file HTML e quindi qualsiasi oggetto contrassegnato con **class="centro"** riceverà gli attributi specificati, in questo caso l'allineamento al centro.

```
<h1 class="centro">Allineamento al centro </h1>
```

```
<p class="centro"> Allineamento al centro</p>
```

Lo stile delle pagine WEB

Lo stesso meccanismo è valido per i selettori di tipo **id**. In un documento HTML l'attributo **id** è usato per identificare in **modo univoco** un elemento. In pratica, se assegno ad un paragrafo l'id **centro**, non potrò più usare questo valore nel resto della pagina. Di conseguenza, l'id **#centro** dichiarato nel CSS trasformerà solo quel paragrafo specifico.

Una singola classe, al contrario, può essere assegnata a più elementi, anche dello stesso tipo.

In un documento potremo avere il seguente scenario:

```
p {color: white;}
```

Tutto il paragrafo sarà bianco ma se volessi diversificare allora dovrei agire in questo modo :

```
.testobianco {color: white;}  
.testonero {color: black;}
```

```
<p class="testobianco">Il testo di questo paragrafo è bianco.</p>  
<p class="testonero">Il testo di questo paragrafo è nero.</p>
```

Così da avere una parte del testo di colore bianco e l'altra nera:

NOTA

La strategia dovrà essere questa, se lo stile va applicato ad un solo specifico elemento usate un **id**, se prevedete di usarlo più volte e su più elementi, allora definite una classe nel CSS.

Selettore di classi

Per definire una *classe* si usa far precedere il nome da un semplice punto “.”:

```
.nome_della_classe
```

Questa è la sintassi di base. Un selettore di classe così definito può essere applicato a tutti gli elementi di un documento HTML.

Esiste un secondo tipo di sintassi più restrittivo rispetto alla sintassi generica

```
p.testobianco {color: white;}
```

lo stile verrà applicato solo ai paragrafi che presentino l'attributo **class**="testobianco".

Anche qui è importante stabilire un minimo di strategia. Il secondo tipo di sintassi va usato solo se pensate di applicare una classe ad uno specifico tipo di elemento (solo paragrafi o solo div, e così via). Se invece ritenete di doverla applicare a tipi diversi usate la sintassi generica.

Selettore di id

La sintassi per il selettore **id** è semplicissima. Basta far precedere il nome dal simbolo di cancelletto #:

```
#nome_id
```

Con questa regola

```
#titolo {color: blue;}
```

assegniamo il colore blue all'elemento che presenti questa definizione nel codice HTML:

```
<h1 id="titolo">...</h1>
```

I selettori di attributo

I selettori di attributo servono a selezionare gli elementi in base ai loro attributi e/o al valore di tali attributi, un esempio ci chiarirà meglio le idee :

```
elemento[attributo] {dichiarazioni;}
```

```
<style>
```

```
  p#parametro {
```

```
    text-align:center;
```

```
    color:red;
```

```
  }
```

```
</style>
```

dove #parametro è l'**id** dell'oggetto a cui si vogliono applicare gli attributi oppure un gruppo di oggetti definiti da quell'id (parametro)

ES : <p **id**="parametro">Ciao Ragazzo!</p>

Oppure potremmo tra tutti selezionarne solo uno così :

```
<p> Ciao Ragazzo! </p>
```

```
<p> Ciao Classe! </p>
```

```
<p id="parametro"> Ciao Scuola! </p>
```


Il nome dell'attributo va posto tra parentesi quadre, senza racchiuderlo tra virgolette. Tra il nome dell'elemento e la definizione dell'attributo non va lasciato spazio.

assegna il colore bianco (**white**) e la sottolineatura a tutti gli elementi a che presentino un attributo **title**:

```
a[title] {color: white; text-decoration: underline;}
```

Dunque, in questa sequenza HTML, solo il primo e il terzo link corrispondono e solo ad essi sarà applicata la regola:

```
<p>Ecco un <a href="#" title="link">link</a>.</p>  
<p>Ancora un <a href="#">link</a> ma senza l'attributo title.</p>  
<p>Un altro <a href="#" title="">link</a>, con title vuoto.</p>  
<p>Ultimo <a href="#" titles="link">link</a>.</p>
```

osservazioni:

- il secondo link non corrisponde al selettore perché non ha un attributo **title**;
- il terzo link corrisponde anche se l'attributo non ha un valore;
- il quarto non corrisponde perché il nome dell'attributo non è identico a quello definito nella regola CSS.

Selezione con valore corrispondente

Questo tipo di selettore individua tutti gli elementi che abbiano come valore dell'attributo specificato la stringa di testo impostata nella regola CSS.

```
elemento [attributo="valore"] {dichiarazioni;}
```

La stringa corrispondente al valore va messa tra virgolette se composta da più di una parola. Posso dunque scrivere così:

```
a[title=lorem]
```

Lo stile delle pagine WEB

Ma non così, perché la regola non verrebbe applicata:

```
a[title=lorem ipsum]
```

Andiamo subito a creare una regola CSS per la dimostrazione pratica:

```
a[title="lorem ipsum"] {color: white; text-decoration: underline;}
```

Partendo da questo codice HTML, solo il primo link corrisponderà:

```
<p>Ecco un <a href="#" title="lorem ipsum">link</a>.</p>
<p>Ancora un <a href="#" title="link">link</a>.</p>
<p>Un altro <a href="#" title="loremipsum">link</a>.</p>
<p>Ultimo <a href="#" title="LOREM IPSUM">link</a>.</p>
```

Perché solo il primo corrisponde?

- Nel primo link il valore dell'attributo nel codice HTML coincide perfettamente con quello definito nel CSS;
- nel secondo il valore è del tutto diverso;
- nel terzo le due parole della stringa sono scritte tutte attaccate;
- nel quarto sono scritte in maiuscolo, mentre nel CSS abbiamo usato il minuscolo (nella definizione del valore, dunque, si fa distinzione tra caratteri maiuscoli e minuscoli).

Selezione in base a valori che contengono una stringa

Questo tipo seleziona tutti gli elementi con un attributo che contenga una lista di parole separate da spazi, una delle quali corrisponde esattamente al valore definito nella regola CSS.

A primo impatto non è molto intuitivo il suo funzionamento, ma cercherò di renderla comprensibile con un esempio.

```
elemento[attributo~="valore"] {dichiarazioni;}
```

Lo stile delle pagine WEB

La differenza principale con il precedente selettore è che il segno = va preceduto dalla tilde (~) si guardi l'esempio qui sotto

```
a[title~="lorem"] {color: white; text-decoration: underline;}
```

Per assegnare la proprietà ad un elemento a è necessario che l'attributo **title** contenga una lista di parole separate da spazi e che nella lista sia presente la stringa *lorem*.

Dato questo codice HTML, dunque, la regola verrà applicata solo al primo e secondo link. Il terzo, infatti, non contiene la stringa *lorem*, il quarto la presenta in maiuscolo mentre nel CSS è stata scritta in minuscolo:

```
<p>Ecco un <a href="#" title="lorem ipsum sit">link</a>.</p>
<p>Ancora un <a href="#" title="link lorem">link</a>.</p>
<p>Un altro <a href="#" title="ipsum sit">link</a>.</p>
<p>Ultimo <a href="#" title="LOREM link">link</a>.</p>
```

Selezione in base a valori che iniziano con una certa stringa

Si tratta di un selettore pochissimo utilizzato e di scarsa utilità. Individua tutti gli elementi in cui uno specifico attributo contiene una lista di parole separate da trattini, una delle quali corrisponde al valore definito nella regola CSS.

```
elemento[attributo]="valore" {dichiarazioni;}
```

Riprendiamo modificandolo l'esempio visto poc'anzi:

```
a[title="lorem"] {color: white; text-decoration: underline;}
```

Solo il primo link corrisponde alla regola appena vista:

















```
<p>Ecco un <a href="#" title="lorem-ipsum-dolor">link</a>.</p>
<p>Ancora un <a href="#" title="ipsum-lorem-dolor">link</a>.</p>
```

Il cursore







Ci sono alcuni casi, però, in cui si ha la necessità di modificare il cursore di default adattandolo ad una particolare interfaccia. I CSS ci consentono di **personalizzare il cursore del mouse** in maniera molto semplice, è sufficiente una sola riga di codice:

```
button { cursor: pointer; }
```

La proprietà **cursor** consente di scegliere il tipo di cursore da utilizzare per il selettore. In questo caso è stato scelto il classico cursore a forma di mano per i campi di tipo **button**.

 pointer	 default	 crosshair
 text	 help	 move
 n-resize	 ne-resize	 nw-resize
 s-resize	 se-resize	 sw-resize
 e-resize	 w-resize	 wait
 progress	browser-determined auto	custom image url("url")

Nel ultima versione (CSS3) sono stati introdotti i seguenti cursori:

<i>not-supported</i>	<i>not-supported</i>	<i>not-supported</i>
copy	alias	cell
		
all-scroll	no-drop	not-allowed
		
col-resize	row-resize	vertical-text

Bisogna fare molta attenzione all'utilizzo di cursori personalizzati perché, se utilizzati in circostanze errate, possono causare il disorientamento dell'utente. Soprattutto è sconsigliatissimo modificare i valori di default a cui sono abituati gli utenti, come ad esempio modificare il cursore dei link.

Nascondere

Il prossimo passo, quindi, consiste nel nascondere completamente tutti quegli elementi con una semplice riga di codice come la seguente:

```
#menu, #sidebar, .advertising {
  display: none;
}
```