

# Introduzione ai Fogli di stile

Prof. Accarino

IIS Altiero Spinelli

Via Leopardi 132 Sesto San Giovanni

# Fogli di Stile CSS

Il **CSS** (*Cascading Style Sheets* o *Fogli di stile*) è un linguaggio usato per definire la formattazione di documenti [HTML](#), [XHTML](#) e [XML](#).

Le regole per comporre il CSS sono contenute in un insieme di direttive (*Recommendations*) emanate a partire dal 1996 dal [W3C](#).

L'introduzione del CSS si è resa necessaria per *separare i contenuti dalla formattazione* e permettere una programmazione più chiara e facile da utilizzare

# Fogli di Stile CSS

**Un normale documento Html unisce insieme contenuti, struttura e stile.**

**Infatti volendo visualizzare una normale riga di testo con font verdana, grandezza 3 e colore blu avremmo:**

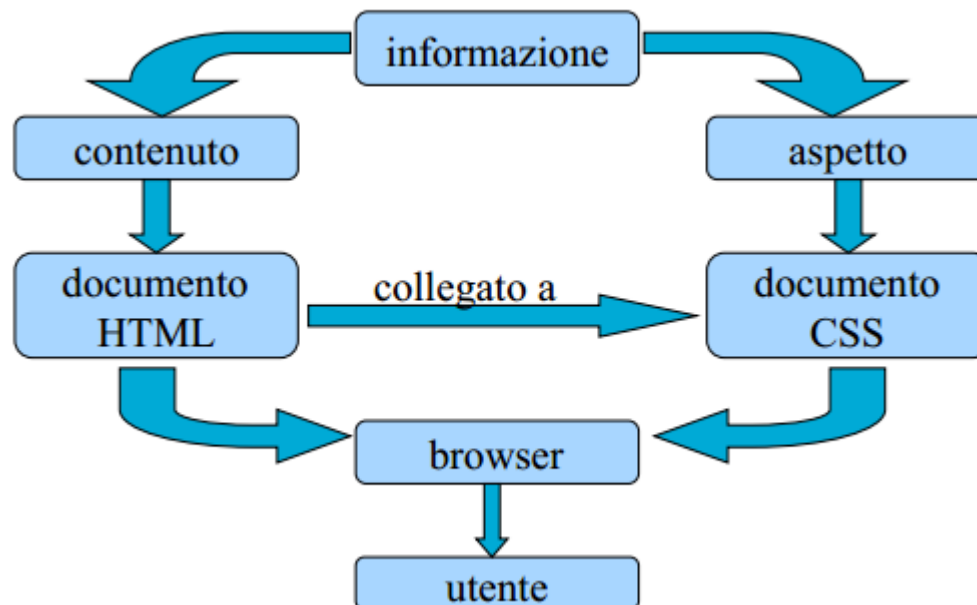
```
<FONT FACE=VERDANA SIZE=3 COLOR=BLUE>Testo comune</FONT>
```

**come si può vedere, il contenuto (Testo comune) e' contenuto nello stile (font, size e color).**

**Se, si volesse modificare il font di un sito formato da 200 documenti, l'operazione, oltre che lunga, Potrebbe risultare snervante.**

# Fogli di Stile CSS

- I **CSS** risolvono questo problema perchè ***separano il contenuto dalla presentazione***. Il contenuto è nella pagina html, mentre la presentazione è riservata ai fogli di stile.



Un foglio di stile è un insieme di regole, accompagnate, volendo, da qualche nota di commento.

# CSS esterni e interni

- È **esterno** un foglio di stile definito in un file separato dal documento. Si tratta di semplici documenti di testo editabili anche con il Blocco Note o TextEdit ai quali si assegna l'estensione **.css**.
- Un foglio di stile si dice invece **interno** quando il suo codice è compreso in quello del documento. 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**
  - In linea.**

# Fogli collegati

I fogli collegati sono **esterni**. Si tratta di fogli di stile definiti in files separati dal documento.

```
<html>
```

```
<head>
```

```
  <title>Inserire i fogli di stile in un documento</title>
```

```
  <link rel="stylesheet" type="text/css" href="stile.css">
```

```
</head>
```

```
<body>
```

```
.....
```

# Fogli incorporati

I fogli incorporati sono quelli inseriti direttamente nel documento HTML tramite l'elemento **<style>**. Anche in questo caso la dichiarazione va posta all'interno della sezione **<head>**:

```
<html>
<head> <title>Inserire i fogli di stile in un documento</title>
<style type="text/css">
  body { background: #FFFFCC; }
</style> </head>
<body>
.....
```

# Gestione del colore



Figura 1 - La palette VGA di Windows

black | navy | blue | maroon | purple | green | red | teal |  
fuchsia | olive | gray | lime | aqua | silver | yellow | white

## #RRGGBB

Le semplici 16 parole chiave non esauriscono ovviamente la gamma dei colori visualizzabili su un monitor moderno.

Già in HTML era possibile impostare il colore di un elemento servendosi di codici esadecimali.

In essi, le prime due lettere (o numeri) corrispondono ai valori per il colore rosso (RED), la seconda coppia fa riferimento al verde (GREEN), l'ultima al blue (BLUE).



# La proprietà color

Visti i sistemi per rappresentare i colori, dobbiamo ora chiarire un aspetto importante. Per ogni elemento si possono definire almeno tre colori:

- il colore di primo piano (foreground color)
- il colore di sfondo (background color)
- il colore del bordo (border color)

La proprietà color definisce esclusivamente:

- il colore di primo piano, ovvero quello del testo
- il colore del bordo di un elemento quando non si imposti esplicitamente quest'ultimo con le proprietà border o border-color

[Esempio](#)

# Fogli in linea

L'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'**attributo 'style'**. Esso fa parte della collezione di attributi (X)HTML definita **Common**: 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. La sintassi generica è la seguente:

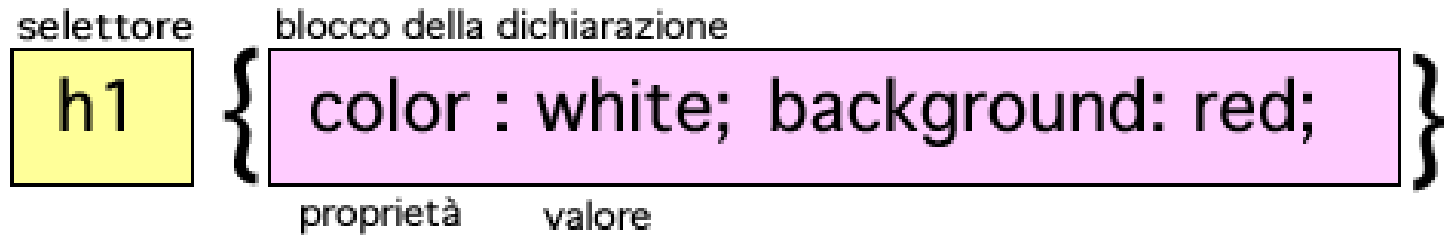
```
<elemento style="regole_di_stile">
```

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>
```

Le cose da osservare nel codice sono due. Come valore di **style** si possono dichiarare più regole di stile. Esse vanno separate dal punto e virgola. I due punti si usano invece per introdurre il valore della proprietà da impostare.

# Com'è fatta una regola



Questa è una tipica struttura di una regola CSS. Essa è composta da due blocchi principali:

- **il selettore**
- **il blocco delle dichiarazioni**

Il selettore serve a definire la parte del documento cui verrà applicata la regola. In questo caso, ad esempio, verranno formattati tutti gli elementi **<H1>**: lo sfondo sarà rosso, il colore del testo bianco.

Il blocco delle dichiarazioni è delimitato rispetto al selettore e alle altre regole da **due parentesi graffe**. Al suo interno possono trovare posto più dichiarazioni. Esse sono sempre composte da una coppia:

- **Proprietà**
- **valore**

La proprietà definisce un aspetto dell'elemento da modificare (margini, colore di sfondo, etc) secondo il valore espresso. Proprietà e valore devono essere separati dai due punti.

# Proprietà singole e a sintassi abbreviata

Le proprietà singole sono la maggior parte: impostano per un dato elemento o selettore un singolo aspetto. Con le proprietà a sintassi abbreviata è possibile invece definire con una sola dichiarazione un insieme di proprietà.

Esempio :

```
p{  
margin-top: 10px;  
margin-right: 5px;  
margin-bottom: 10px;  
margin-left: 5px;  
}
```

Proprietà Singole

Proprietà a sintassi abbreviate

```
p {margin: 10px 5px 10px 5px;}
```

# I selettori

Fondamentalmente una regola CSS viene applicata ad un **selettore**. 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 di elementi

È il più semplice dei selettori. È costituito da uno qualunque degli elementi di HTML.

### Sintassi:

```
h1 {color: red;}
```

```
p {background: white; font: 12px Verdana, arial, sans-serif;}
```

```
table {width: 200px;}
```

[Esempio](#)

È possibile nei CSS raggruppare diversi elementi al fine di semplificare il codice. Gli elementi raggruppati vanno separati da una **virgola**.

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

[Esempio](#)

# I selettori

## Selettore universale

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

## Sintassi

[Esempio](#)

```
*{ color: black; }
```

# I selettori

## Selettori delle classi

Per definire una classe si usa far precedere il nome da un semplice punto: `.nome_della_classe`

Esempio:

```
<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      .verde { color:green;}
      .blu { color:blue;}
      .rosso { color:red;}
    </style>
  </head>
  <body>
    <h1 class="verde">Il titolo è in verde</h1>
    <h1 class="blu">Il titolo è in blu</h1>
    <h1 class="rosso">Il titolo è in rosso</h1>
    <em class="blu">La scritta in corsivo è in blu</em>
    <p class="verde">Il paragrafo normale è in verde</p>
  </body>
</html>
```

Le classi sono importanti perché possono essere applicate più volte ad elementi del documento e ad elementi diversi

# I selettori

## Selettori delle classi

Esiste un secondo tipo di sintassi: **<elemento>.nome\_della\_classe**

Esso è più restrittivo rispetto alla sintassi generica perchè applica la regola sollo all'elemento specificato esempio:

```
<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      body { background:#FFFF99; }
      h1.verde { color:green; background:#FFFF99 }
    </style>
  </head>
  <body>
    <h1>Questo titolo non è verde</h1>
    <h1 class="verde">Questo titolo è verde</h1>
  </body>
</html>
```

[Esempio](#)



# Selettore delle identità ID

Un altro selettore speciale, oltre alle classi, è il selettore delle identità (ID selectors).

Le caratteristiche sono sempre le stesse e ciò che vale per le classi vale anche per i selettori.

L'unica differenza sostanziale è nella sintassi del comando che, anziché avere un punto prima del nome della classe, ha una cilletta **#** e nel fatto che un selettore di identità può essere inserito una volta sola in ogni documento

(ovviamente si possono inserire selettori aventi nomi diversi).

La sintassi di un selettore ID è semplicissima. Basta far precedere il nome dal simbolo di cancelletto #: `#nome_id`

# Selettore delle identità ID

Esempio:

```
<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      #verde { color:green;}
      #blu { color:blue;}
      #rosso { color:red;}
    </style>
  </head>
  <body>
    <h1 id="verde">Il titolo è in verde</h1>
    <h1 id="blu">Il titolo è in blu</h1>
    <h1 id="rosso">Il titolo è in rosso</h1>
  </body>
</html>
```



[Esempio](#)

## Classi e ID

La fondamentale differenza fra classi ed ID riguarda l'uso che viene fatto dell'attributo **id** in un documento HTML.

In un documento HTML l'attributo **id** è usato per identificare in **modo univoco** un elemento.

In pratica, se si assegna ad un paragrafo l'id "formattatesto", non si può più usare questo valore nel resto della pagina.

Di conseguenza, l'ID **#formattatesto** dichiarato nel CSS trasformerà solo quel paragrafo specifico.

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

# Classi e ID

Esempio:

```
<p class="testorosso">....</p>
<div class="testorosso">....</div>
<table class="testorosso">...</table>
<p class="testorosso">....</p>
```

```
<style type="text/css">
  .rosso{color:red;}
</style>
```

La classe **.testorosso** presente nel CSS formatterà allo stesso modo il testo del paragrafo, del div e della tabella.

Questo invece è sbagliato. **(usato due volte)**

```
<p id="rosso">....</p>
<div id="rosso">...</div>
```

```
<style type="text/css">
  #rosso{color:red;}
</style>
```

La strategia dovrà dunque essere questa : se uno stile va applicato ad un solo specifico elemento conviene usare un **ID**, se invece si prevede di usarlo più volte ovvero su più elementi conviene definire nel CSS una **classe**.

# Il contenitore div

La sua definizione è *generic block-level element*  
Cioè contenitore generico block level.

Il fatto che sia un elemento block-level ci garantisce il fatto che possa contenere qualsiasi tipo di elemento html. Inoltre, la sua presentazione naturale (quindi senza fogli di stile) è totalmente neutra:

infatti questo elemento si presenta di default senza margini, bordi o padding.

È quindi il contenitore per eccellenza per realizzare layout senza l'uso di tabelle:

**ad ogni div portante verrà associata una sezione della pagina.**

Esso è utilizzato soprattutto per creare layout senza tabelle sfruttando le caratteristiche del cosiddetto **BOX MODEL**