

# CSS

## Cascading Style Sheets

il linguaggio per **modificare lo stile** delle pagine web

1

### Fogli di stile (CSS)

- Servono per facilitare la creazione di pagine HTML con un **aspetto uniforme**
- Permettono di **separare il contenuto** di un documento da aspetti legati alla sua **presentazione**
- Possiamo "dimenticare" alcuni tag e attributi HTML

2

## Fogli di stile (CSS)

.....  
Ma cosa significa separare lo stile dai contenuti? Consideriamo un sito Web mediamente complesso, con un numero di pagine HTML pari a 100. Poniamo che il testo adottato per l'esposizione degli argomenti sia un "arial" corsivo. Con gli strumenti classici di HTML il codice per ottenere questo risultato e':

```
<font face="Arial"><i>Testo della pagina</i></font>
```

ripetuto in tutte le 100 pagine del sito, a chiusura e apertura del testo da formattare. Semplice, ovvio e per molti versi banale. Ma cosa succede nel momento in cui scegliamo di modificare il tipo di carattere a tutte le pagine? Non c'e' altra soluzione che aprire le 100 pagine e procedere ad altrettante modifiche, che sostituiscano il nome "Arial" con il nuovo font scelto. Un webmaster di medie capacita' puo' impiegare solo qualche decina di minuti, che diventano ore per siti di grandi dimensioni. Bene, questa perdita di tempo e' diretta conseguenza della promiscuita' tra stile e contenuto, laddove il primo (il tag FONT) non e' separato dal secondo (il testo della pagina).

Viene da se' che la soluzione a questo problema e' nella separazione tra i due elementi sopracitati, che nella pratica si risolve adottando i Cascading Style Sheets. D'ora in poi ci serviremo dell'acronimo CSS per richiamare questi "fogli di stile a cascata" che da tempo sono stati introdotti nel Web publishing, ma che solo negli ultimi mesi hanno conosciuto una grande diffusione.

.....

dal sito [www.html.it](http://www.html.it)

3

## Fogli di stile (CSS)

- Vantaggi
  - Permettono di modificare il **look & feel** di un documento in modo efficiente
  - Lo stesso stile può essere applicato a più documenti
  - Il sorgente HTML è più "pulito" e il download dei documenti è più veloce
  - Si possono progettare pagine HTML per più browser

4

## Fogli di stile (CSS)

- Il Consorzio che si occupa della standardizzazione del linguaggio per il web (<http://www.w3c.org>) suggerisce di usare i CSS perché facilitano la progettazione e la modifica delle pagine HTML
- 1996, CSS1: servono per modificare l'**aspetto degli elementi** nelle pagine HTML (colore, dimensione caratteri, ...)
- 1998, CSS2: permettono funzioni più sofisticate e introducono la possibilità di **posizionamento** per gli elementi nelle pagine HTML

5

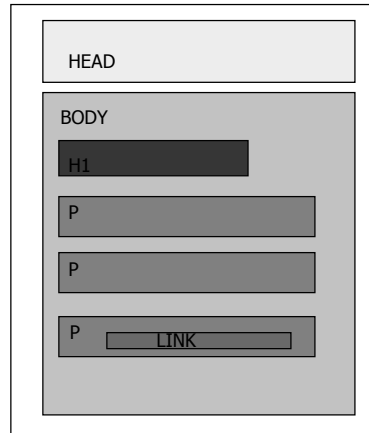
## CSS: blocchi

- Ogni documento HTML può essere visto come un **insieme di blocchi** (contenitori) sui quali si può agire con stili diversi
- I tag HTML definiscono i blocchi
- Per ogni blocco si può specificare uno stile che ne descrive l'aspetto

6

## CSS: blocchi

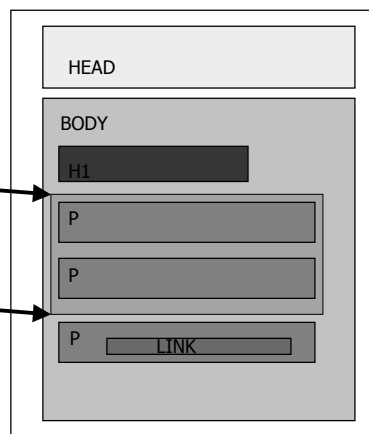
```
<html>
<head>...</head>
<body>
<h1>titolo</h1>
<p> primo paragrafo </p>
<p> secondo paragrafo </p>
<p> terzo paragrafo
<a href="pippo.html">link</a>
</p>
</body>
</html>
```



7

## CSS: <div> e <span>

```
<html>
<head>...</head>
<body>
<h1>titolo</h1>
<div>
<p> primo paragrafo </p>
<p> secondo paragrafo </p>
</div>
<p> terzo paragrafo
<a href="pippo.html">link</a>
</p>
</body>
</html>
```



8

## CSS: <div> e <span>

- Permettono di strutturare le pagine HTML
- <div> definisce un blocco generico (block level)
- <span> permette di specificare elementi in linea (in line)
- Associando degli stili ai tag <div> e <span> si possono ottenere effetti grafici sofisticati

9

## CSS: regole

- Un foglio di stile è una **collezione di regole** stilistiche che definiscono il look & feel degli elementi
- Ogni regola è costituita da un **selettore** cui è associato un elenco di **proprietà**. Ad ogni proprietà si associa un **valore**

```
selector {  
    property1 : value1;  
    property2 : value2;  
    ...  
    propertyN : valueN;  
}
```

10

## CSS: regole

- Un selettore può essere un tag HTML

```
h1 { color:#dd5500; font-family:Arial; }  
  
p {  
    margin-top:50;  
    color:blue;  
    font-style:italic;  
}
```

11

## CSS: proprietà (esempi)

Per lo sfondo  
background-color  
background-image  
...

Per i margini  
margin-left  
margin-right  
margin-top  
margin-bottom  
...

Per il testo  
font-style  
font-weight  
font-size  
font-family  
text-align  
text-transform  
text-color  
text-decoration  
...  
...

12

## Esempi

```
body { color:black; background-color:yellow; }
p     { font-size:120%; font-style:italic; color:green; }
h1    { margin-left:10%; margin-right:10%; }
h2    { font-family: "Times New Roman", Arial; }
```

```
table, tr, td, th, li
    {   color:blue;
        font-family: "sans serif";
        font-size:12pt;
    }
```

13

## Esempi

```
body {
    background-image:url(sfondo.gif);
    background-attachment:fixed;
    background-repeat:no-repeat;
    background-position:center center;
}
```

**esempio**

[per lo sfondo]

```
body {
    scrollbar-base-color:blue;
    scrollbar-face-color:red;
    scrollbar-arrow-color:yellow;
}
```

[per la barra di scorrimento]

## CSS: dove si scrivono? [embed]

- Le regole stilistiche possono essere scritte nella parte <head> usando il tag <style>

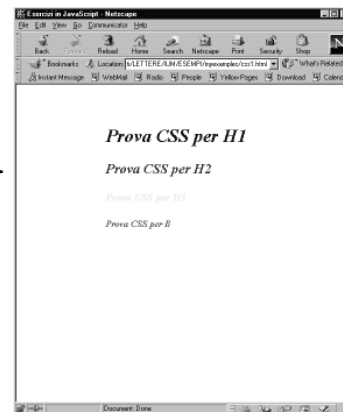
```
<style type="text/css">
<!--
    regole per definire gli stili
-->
</style>
```

**NB:** i commenti servono per evitare errori con i browser che non supportano i fogli di stile

15

## CSS: dove si scrivono? [embed]

```
<html>
<head>
<style type="text/css">
<!--
  h1 {color:blue; font-style:italic;}
  h2 {color:red; font-style:italic;}
  h3 {color:yellow; font-style:italic;}
  b {color:green; font-style:italic;}
-->
</style>
</head>
<body bgcolor="#ffffff">
<center>
<h1> Prova CSS per H1 </h1>
<h2> Prova CSS per H2 </h2>
<h3> Prova CSS per H3 </h3>
<b> Prova CSS per B </b>
</center>
</body>
</html>
```



16



## CSS: dove si scrivono? [inline]

- <style> può anche essere usato come **attributo** di altri tag HTML

```
<h1 style="color:red; text-transform:capitalize;">  
  Prova CSS per h1  
</h1>
```

**NB:** ci vogliono gli apici " ... " e non le parentesi graffe!

17

## CSS: dove si scrivono? [external]

- Il metodo migliore è quello di scrivere gli attributi di stile in un **file esterno**, che può essere condiviso da più file HTML
- Il file che contiene gli stili deve essere un file in **formato testo** (ASCII) con estensione **.css**

file stile.css

```
h1 {color:red; text-transform:capitalize;}  
p {color:blue;}
```

**NB:** nel file .css non si devono inserire tag HTML ma solo le regole di stile

18

## CSS: dove si scrivono? [external]

- Nel file HTML si deve usare il tag <link>

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

19

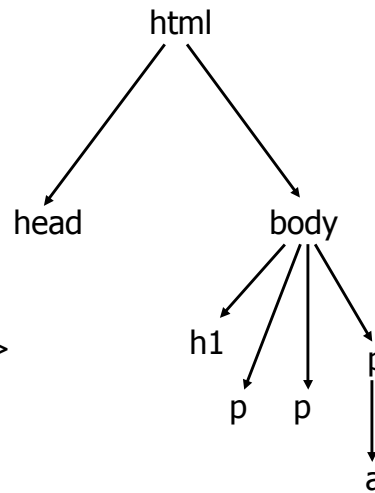
## CSS: dove si scrivono?

- Un file HTML può importare più file di stile esterni
- Se per lo stesso tag ci sono più stili, le regole di ereditarietà - cascade rules - ne determinano l'aspetto
- È difficile prevedere a priori come il browser risolverà i conflitti tra stili diversi
- Si dà la priorità alle regole definite all'interno del documento

20

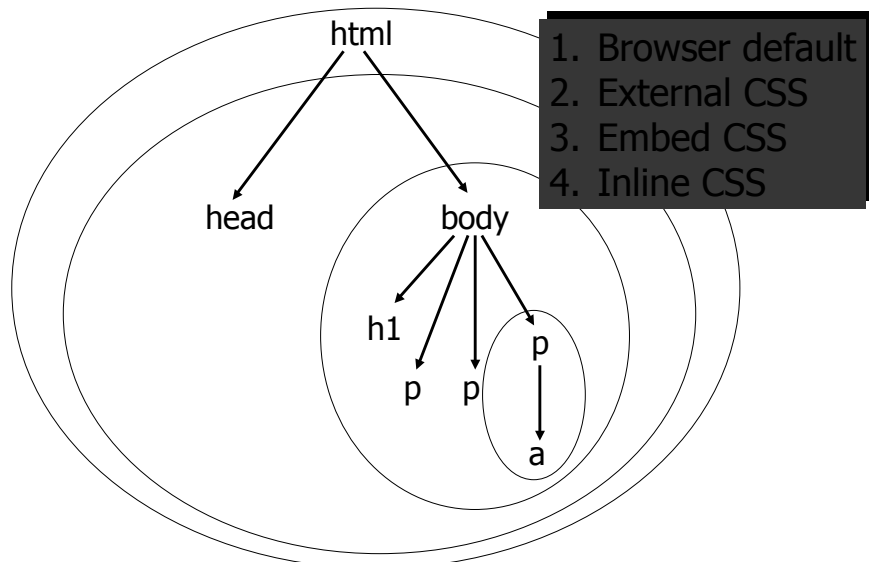
## CSS: ereditarietà

```
<html>
<head>...</head>
<body>
<h1>titolo</h1>
<p> primo paragrafo </p>
<p> secondo paragrafo </p>
<p> terzo paragrafo
<a href="pippo.html">link</a>
</p>
</body>
</html>
```



21

## CSS: ereditarietà



**NB:** non tutti gli attributi vengono ereditati

22

## CSS: class selector

- Si può assegnare lo stesso stile a tag di tipo diverso introducendo una **classe** che definisce delle proprietà

nel file stile.css

```
...  
h1 { font-style:italic; }  
.imp { color:red; text-decoration:underline; }  
...  
....
```

→ Identificatore preceduto da un "."

23

## CSS: class selector

nel file .css

```
h1 { font-style:italic; }  
.imp { color:red; text-decoration:underline; }
```

nel file .html

```
...  
<body>  
<h1> Primo titolo </h1>  
<h1 class="imp"> Titolo da evidenziare </h1>  
...  
...  
</body>  
</html>
```



24

## CSS: pseudo-class

- Permettono di associare stili diversi allo stesso selettore

```
selector : pseudo-class  
    { property1 : value1; property2 : value2; ... }
```

```
a:link      {color:red; text-decoration:none;}  
a:visited  {color:blue; text-decoration:none;}  
a:hover    {color:green; text-decoration:underline;}
```

[per i link]

**esempio**

25

## CSS: id selector

- Gli identificatori permettono di definire delle regole che si applicano ad **un solo** elemento nella pagina

nel file stile.css

```
...  
#speciale { color:red; text-transform:uppercase; }  
...
```

→ Identificatore preceduto da un "#"

26

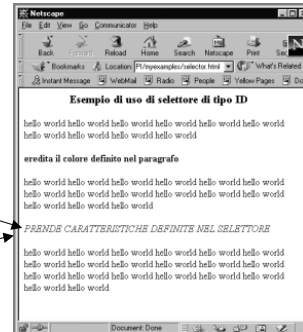
## CSS: id selector

nel file .css

```
P { color:blue; }  
#speciale { color:red; text-transform:uppercase; }
```

nel file .html

```
<body>  
...  
...  
<i id="speciale">prende caratteristiche definite  
nel selettore</i>  
...  
...
```



27

## CSS: posizionamento

- Permette di specificare **dove** verranno visualizzati gli elementi di una pagina HTML
- Il posizionamento è una delle parti più critiche dei CSS (non c'è completa compatibilità tra i browser)
- Ogni **blocco** (porzione di documento compresa tra due tag) può diventare un **elemento posizionabile**

28

## CSS: posizionamento

- Per avere un elemento posizionabile si deve usare la proprietà **position**

nel file stile.css

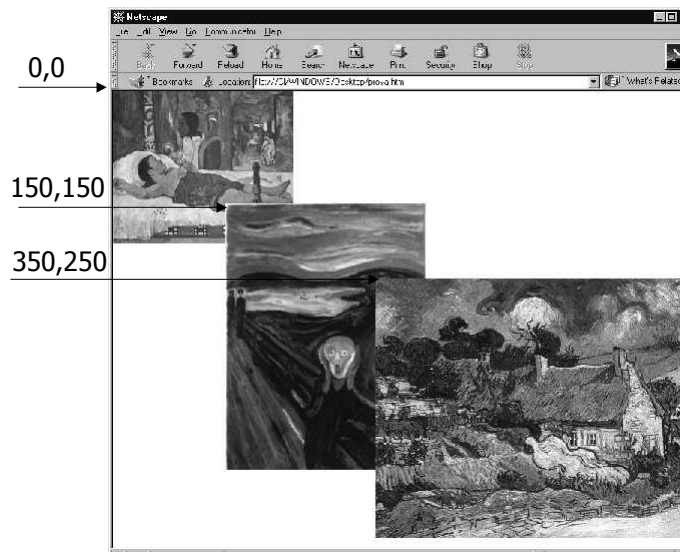
...

```
#posiz {position:absolute; left:250; top:250;}
```

...

29

## CSS: posizionamento



30

## CSS: posizionamento

nel file .css

```
#posiz1 {position:absolute; top:0; left:0;}
#posiz2 {position:absolute; top:150; left:150;}
#posiz3 {position:absolute; top:350; left:250;}
```

nel file .html

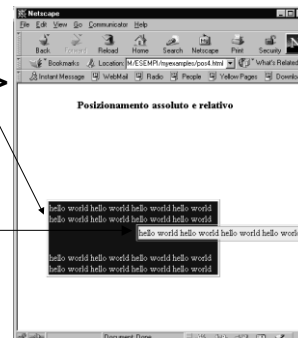
```
...
<body bgcolor="white">
<div id="posiz1"></div>
<div id="posiz2"></div>
<div id="posiz3"></div>
....
```

**NB:** se si usano gli stessi valori di posizione le immagini risultano sovrapposte

31

## CSS: posizionamento (relativo)

```
<div style="position:absolute; left:50; top:200;">
<table width="300" bgcolor="blue" border="1">
<tr>
<td><font COLOR="#ffffff">
hello world hello world hello world hello world
hello world hello world hello world hello world
<div style="position:relative; left:150;">
<table bgcolor="yellow" border="1">
<tr>
<td>hello world hello world hello world hello world</td>
</tr>
</table>
</div>
hello world hello world hello world hello world
hello world hello world hello world hello world
</font>
</td>
</tr>
</table>
</div>
```



32



## CSS: posizionamento

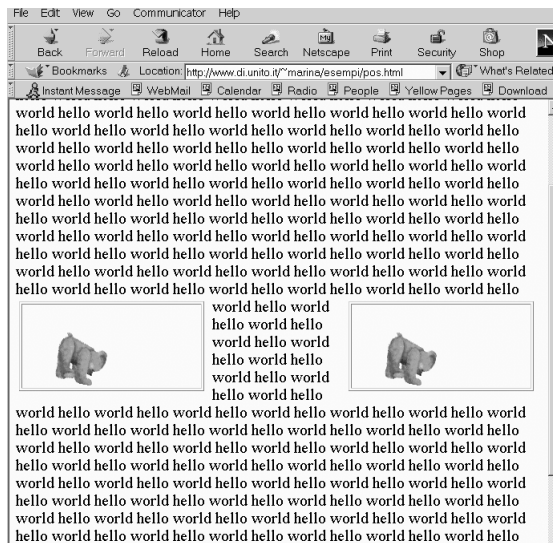
- Se un elemento deve "avvolgere" un altro elemento si può usare la proprietà **float**

```
<span style="float:right;">  
      
</span>
```

esempio

33

## CSS: posizionamento



34

## CSS: z-index

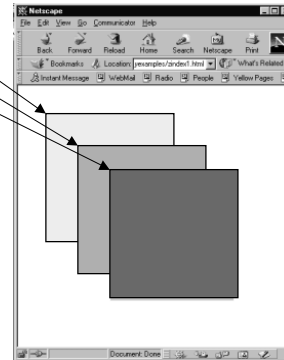
- Si può modificare l'ordine di visualizzazione degli elementi usando z-index

nel file .css

```
#posiz1 { position:absolute; left: 50; top:50; }  
#posiz2 { position:absolute; left:100; top:100; }  
#posiz3 { position:absolute; left:150; top:150; }
```

nel file .html

```
...  
<div id="posiz1"></div>  
<div id="posiz2"></div>  
<div id="posiz3"></div>  
...
```



35

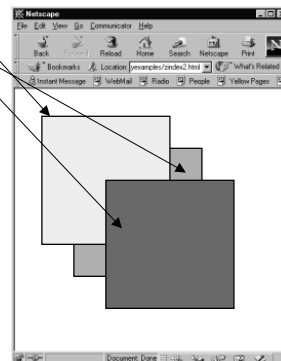
## CSS: z-index

nel file .css

```
#posiz1 { position:absolute; z-index:2; left: 50; top:50; }  
#posiz2 { position:absolute; z-index:1; left:100; top:100; }  
#posiz3 { position:absolute; z-index:3; left:150; top:150; }
```

nel file .html

```
...  
<div id="posiz1"></div>  
<div id="posiz2"></div>  
<div id="posiz3"></div>  
...
```



36

## CSS: pseudo-element

- Permettono di associare stili diversi allo stesso selettore

```
selector : pseudo-element  
  { property1 : value1; property2 : value2; ... }
```

```
p { font-size: 12 pt; }  
  
p : first-line { color:blue; font-variant small-caps; }  
  
p : first-letter { font-size: xx-large; }
```

**esempio**

37

## CSS: media types

- Permettono di specificare come devono essere visualizzati i documenti su media diversi

@media all | aural | braille | print | screen

```
@media screen {  
    font-size : 12pt;  
    font-family : Verdana; }  
  
@media print {  
    font-size : 10pt;  
    font-family : Times; }
```

38