

Umberto Emanuele

HTML & CSS 1

Giorno 3

Novembre 2023



Introduzione ai CSS e sintassi base

Cascading Style Sheet

Fogli di stile a cascata.

Regole di formattazione nate per sostituire gli attributi di formattazione del vecchio HTML4.

Vantaggi

Controllare la formattazione di più pagine.

Snellire l'HTML.



CSS level 1 - 2 - 2.1 - 3

Verifica compatibilità: <https://caniuse.com/>

Selettore { proprietà: valore; } – Sintassi CASE - SENSITIVE

```
* { font-style: italic; } /* SELETTORE UNIVERSALE */  
p { font-size: 30px; } /* SELETTORE BASE */  
em { font-weight: bold; color: gray; } /* SELETTORE BASE CON PROPRIETA' MULTIPLE */  
p strong { color: hsla(240, 100%, 25%, 0.507); } /* SELETTORE DISCENDENTE */  
ul, ol { margin-left: 45px; } /* SELETTORE DI GRUPPO */
```

CSS Inline

Metodologia deprecata.

Pressoché simile ai vecchi attributi

HTML con tutti i relativi svantaggi.

```
<h1 style="color: ■ navy; text-align:center;">Titolo principale</h1>  
✓ <p style="font-size: 14px;">Testo della pagina con  
  |   <strong style="color: ■ orange;">elemento in evidenza</strong>  
  </p>
```

CSS incorporati

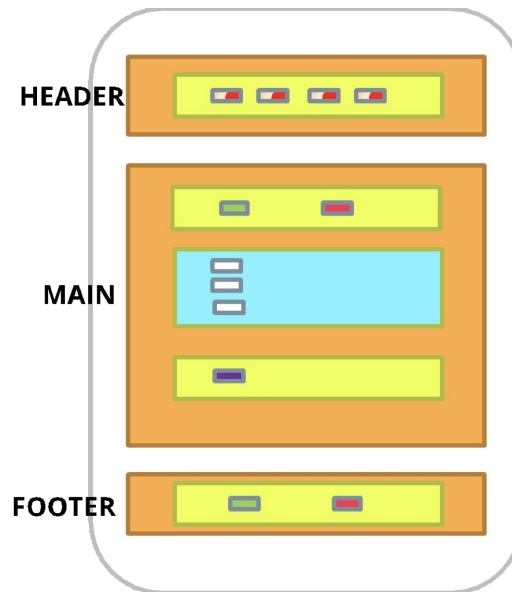
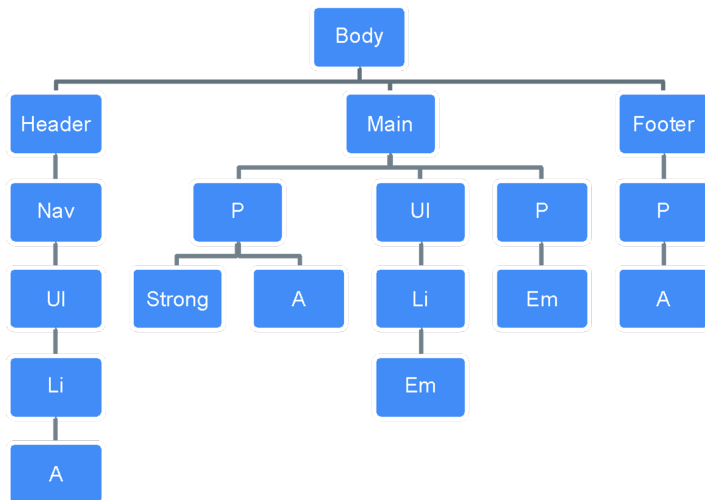
Metodologia sconsigliata, utilizzata solo in casi eccezionali. Il codice è inserito all'interno della pagina specifica (sezione HEAD).

```
<head>
  <title>Pagina prodotti azienda</title>
  <style>
    h1 {color: navy; text-align:center;}
    p {font-size: 14px;}
    strong {color: orange;}
  </style>
</head>
```

CSS esterni

Metodologia raccomandata. Le regole sono contenute in fogli esterni, collegati a TUTTE le pagine che necessitano di fruire di quelle formattazioni.

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




```
strong { color: orange !important; } /* OK su TUTTO*/
```

Priorità a cascata

1. Dichiarazioni inline
2. Dichiarazioni incorporate /
esterne
3. Dichiarazioni del browser

Priorità specifica

L'ultima dichiarazione sovrascrive
la precedente.

```
p { color: red; }  
p { color: navy; } /* OK */
```

Priorità gerarchica

1. Stili inline (1000 punti)
2. ID (100 punti)
3. Classi, Pseudo -classi,
Attributi (10 punti)
4. Elementi (1 punto)

```
p strong { color: red; } /* OK */  
strong { color: green; }
```

Colori ed unità di misura

Assolute

Misure non relative ad alcun riferimento,
pressappoco uguali per tutti.

Ormai sconsigliate:

- Pixel (px)
- Punti (pt)

```
p { font-size: 30px; }
```

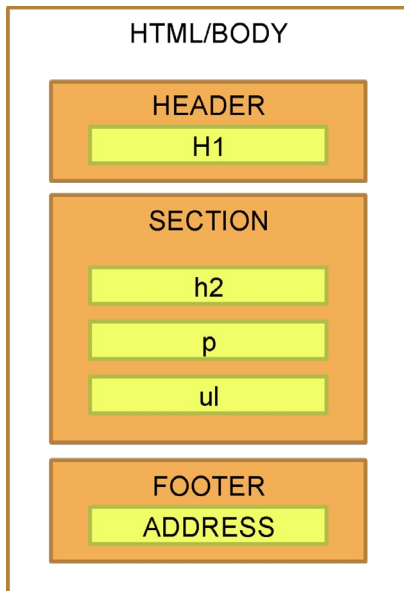
Relative

Misure relative al contenitore parent o al viewport. Sono raccomandate per favorire la User Experience dell'utente.

| | |
|------------|---|
| % | proporzionali all'elemento parent |
| em | proporzionali all'elemento parent |
| rem | proporzionali all'elemento Root (radice) |
| vw | proporzionali alla larghezza del viewport |
| vh | proporzionali all'altezza del viewport |

```
<main>
  <article>
    <p>Testo</p>
  </article>
</main>
```

| | |
|-------------------------|-------------------------------|
| p {font -size: 90%;} | Article: 16px / P: 14.4px |
| p {font -size :0.9em;} | Article: 16px / P: 14.4px |
| p {font -size: 0.9rem;} | Article: ... / P: 14.4px |
| p {font -size: 0.9vw;} | Viewport W: 350px / P: 3.15px |
| p {font -size: 0.9vh;} | Viewport H: 500px / P: 4.5px |



```
html {font-size: 62.5%;} /* 10px */  
header h1 {font-size: 2rem;} /* 20px */  
section {font-size: 1.1rem;} /* 11px */  
section h2 {font-size: 1.2em;} /* 13.2px */  
/* section p/ul = 11px */  
footer {font-size: 1rem;} /* 10px */  
footer address {font-size: 0.8em;} /* 8px */
```

Testuali

Valori letterali, semplici da ricordare ma con palette limitate rispetto a sintassi numeriche

- CSS 1 Color Keywords (es: blue, orange, red)
- CSS 3 Color Keywords (es: darkcyan, lightgray)
- Transparent (Keyword specifica)



https://developer.mozilla.org/en-US/docs/Web/CSS/color_value

Numerici

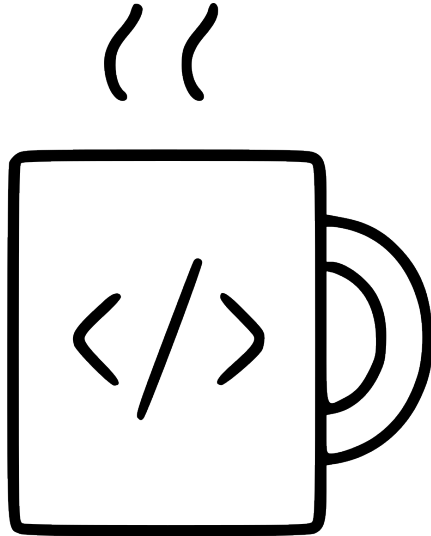
Valori numerici, permettono maggiore varietà cromatica e uso delle trasparenze

| | | |
|-------------|--|-----------------------|
| RGB | Red Green Blue (0 - 255) | rgb(0,0,128) |
| RGBA | Red Green Blue Alpha (0 - 255) + (0 - 1) | rgb(0,0,128,0.5) |
| HEX | Esadecimale (00 - FF) | #000080 |
| HEXA | Esadecimale Alpha (00 - FF) + (FF-00) | #000080ae |
| HSL | Hue Saturation Lightness | hsl(240,100%,25%) |
| HSLA | Hue Saturation Lightness Alpha | hsl(240,100%,25%,0.5) |

Proprietà CSS per i testi

Serif, Sans-serif o Monospaced

| | |
|-------------------|--|
| Serif | Caratteri graziati, con abbellimenti alle estremità |
| Sans-serif | Caratteri non graziati (o bastoni), senza abbellimenti |
| Monospace | Caratteri con uguale spaziatura |
| Cursive | Caratteri che simulano la scrittura manuale |
| Fantasy | Caratteri dall'estetica particolare |



PAUSA

Ci vediamo alle ore 11.25

- Arial (sans-serif)
- Verdana (sans-serif)
- Helvetica (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

```
article {  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
body {  
  font-family: 'Edward Script', 'Brush Script MT', cursive;  
}
```

```
@font-face {  
  font-family: FontPersonale;  
  src: url('railway.woff2') format('woff2'), /* Browser recenti */  
       url('railway.woff') format('woff'), /* Browser moderni */  
       url('railway.ttf') format('truetype'), /* Safari, Android, iOS */  
}  
  
p {  
  font-family: FontPersonale;  
}
```



```
article {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 1.1rem; /* px, em, %, vw */  
  font-style: italic; /* normal */  
  font-weight: bold; /* normal, 100-900 */  
  font-variant: small-caps; /* normal */  
  line-height: 1.5; /* da zero in su */  
}  
  
body {  
  font: italic small-caps bold 1.1rem/1.4 Arial, sans-serif;  
}
```

```
section {  
  color: ■darkgray;  
  letter-spacing: 5px; /* -5px */  
  text-align: center; /* left, right, justify - allinea il CONTENUTO */  
  text-indent: 30px; /* fa rientrare solo la prima linea - anche in % o em */  
  text-transform: uppercase; /* lowercase, capitalize */  
}
```

```
h1 {  
  text-shadow: 4px 4px 8px 2px ■ #666; /* singola */  
  text-shadow: 4px 4px 8px 2px ■ #666, -4px -4px 8px 2px ■ #333; /* multipla */  
}
```

| | |
|-----------|--------------------------|
| 1° valore | Disposizione orizzontale |
| 2° valore | Disposizione verticale |
| 3° valore | Raggio di sfocatura |
| 4° valore | Spread dell'ombra |
| 5° valore | Colore dell'ombra |

Selettori: ID e classi

Elemento unico

L'ID identifica in maniera univoca un elemento nella pagina. E' case -sensitive.

Il simbolo che precede l'ID è il # (cancellotto).

```
#banner {  
  background-color: ■ #c10019;  
}
```

Elemento ripetuto

La classe identifica uno o più elementi nella pagina. E' case-sensitive.

Il simbolo che precede la classe è il . (punto)

Possiamo applicare più classi allo stesso elemento.

```
.special {  
    font-weight: bold;  
    text-transform: capitalize;  
}  
#banner .special {  
    font-weight: bold;  
    text-transform: capitalize;  
}  
p.special {  
    font-weight: bold;  
    text-transform: capitalize;  
}
```



Get vector icons and social logos on your website with Font Awesome, the web's most popular icon set and toolkit.

<https://fontawesome.com>



<https://icomoon.io>



<https://fontello.com>

Selettori avanzati e pseudo-classi

Identificano determinati elementi basandosi sulla loro relazione con altri elementi

| | |
|---|----------------|
| Discendenti – tutti i p discendenti di article | article p |
| Figli (Child Selectors) – tutti i div figli diretti di section | section > div |
| Adiacenti (Adjacent Sibling Selectors) – il primo p messo dopo un blockquote | blockquote + p |
| Seguenti (General Sibling Selectors) – tutte le ul precedute da un p | p ~ ul |

Identificano determinati elementi basandosi sul loro stato, posizionamento o interazione.

I nomi sono predefiniti. Alcuni esempi:

| | |
|---|---|
| :hover | Quando il puntatore passa su un elemento |
| :playing / :paused | Quando un elemento multimediale è messo in play o pausa |
| :first -child / :last -child | Primo o ultimo elemento figlio |
| :first -of -type / :last -of -type | Primo o ultimo elemento figlio specifico |
| :not(selector) | Elemento che non è il selettore indicato |

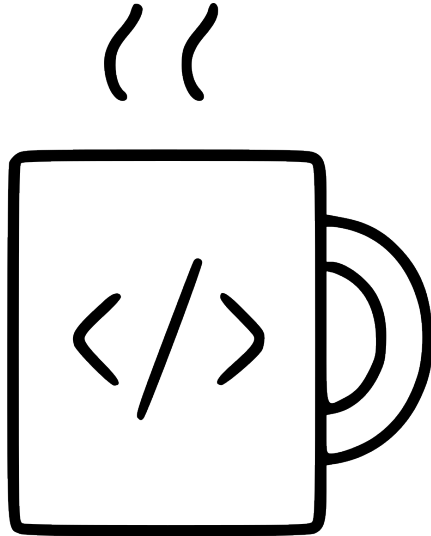
Identificano determinati elementi basandosi sulle loro caratteristiche, che non sono specificate tramite HTML. I nomi sono predefiniti. Alcuni esempi:

| | |
|---|---|
| :hover | Quando il puntatore passa su un elemento |
| :playing / :paused | Quando un elemento multimediale è messo in play o pausa |
| :first -child / :last -child | Primo o ultimo elemento figlio |
| :first -of -type / :last -of -type | Primo o ultimo elemento figlio specifico |
| :not(selector) | Elemento che non è il selettore indicato |

Tramite la proprietà content si possono inserire caratteri, attributi o immagini

```
p::before {content: "Articolo: ";} /* testo */
section::before {counter-increment: capitolo; content: "sezione: " counter(capitolo);} /* contatore */
a::before {content: attr(href);} /* valore dell'attributo */
h2::after {content: url(assets/images/freccia.png);} /* immagine */
span::before {content: open-quote;} /* doppi apici di apertura */
span::after {content: close-quote;} /* doppi apici di chiusura */
```





PAUSA

Ci vediamo alle ore 14.00



shaping the skills of tomorrow

challengenetwork.it

