

CSS (Cascading style sheets)

- Utilizzati per stilizzare delle pagine web
- Passano in stato «Recommended» del W3C nel 1996
- Si inseriscono all'interno del tag <head> di una pagina HTML e collegano un file esterno

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

- Questa soluzione garantisce il massimo riutilizzo di codice possibile potendo essere richiamato in ciascuna delle pagine web HTML desiderate.



Regole CSS

E' costituito da una sequenza di *regole*, costituite da un *selettore* e da un *blocco di dichiarazioni*, racchiuso tra parentesi graffe

```
selettore {  
    proprietà1:valore1;  
    proprietà2:valore2;  
}
```

Le proprietà CSS sono numerose, le più utilizzare sono circa 60, tra cui:

background-color: definisce il colore di sfondo di un elemento;

color: definisce il colore del testo di un elemento;

font-size: definisce la dimensione di un testo;

float: definisce un blocco *flottante*, spostabile su uno dei lati sx/dx del suo contenitore

margin e *padding*: definiscono lo spazio circostante agli elementi

text-align: definisce l'allineamento degli elementi, tra cui il testo.

Stile per tag

```
h1 {  
    color: royalblue; /* tutti gli elementi h1 avranno  
questo stile*/  
}  
  
h1, p {  
    text-align: center;  
}  
/*tutti gli elementi h1 e p avranno questo stile*/
```

```
* {  
    color: ■ green;  
}
```

* Applica lo stile specificato a tutti i tag
Viene chiamato ***selettore universale***

Selettori di attributi

Consentono di selezionare un elemento da un suo attributo.

Selettore di attributo nella form

```
input[type="button"] {  
    color: ■ red;  
    background: □ black;  
}  
  
input[type="checkbox"], input[type="radio"] {  
    color: □ black;  
    background: transparent;  
}  
  
input[disabled], input[readonly] {  
    background: ■ gray;  
}
```

Selettore di attributo class o id

```
/*Selettore tramite id*/
#par {
    color: cornflowerblue;
}

/*Selettore tramite CLASS name*/
.paragrafo {
    font-weight: bold;
}
```

```
<p id="par"> ciao </p>
```

```
<p class="paragrafo"> ciao </p>
```

Classe di stile

definisce uno stile richiamabile da ogni tag con l'attributo class.

Permette di definire uno stile per tipologie di tag differenti.

Mescolare i selettori

```
p, .warning{  
  font-weight: bold;  
}
```

Mescolando i selettori per tag e per classe, lo stile verrà applicato se il tag:



- è un p
- ha class warning
- è entrambi

Selettori Combinati

È possibile combinare più selettori tra loro.

Tag1 Tag2

TUTTI gli elementi strong in elementi p avranno colore red


```
p strong {  
  color:  red;  
}  
div * p {  
  background:  yellow;  
}
```

Tag1 * Tag2

Esempio: tutti i tag p all'interno di un tag (anche se innestati) hanno colore di background giallo

Tag1 > Tag2

Esempio: tutti i tag strong 'figli' di p (non innestati)

```
p > strong {  
  color:  green;  
}
```

Selettori Combinati

```
tag1 + tag2 {
```

```
  proprietàZ: value;
```

```
}
```

per selezionare tag2 che si trova DOPO un elemento tag1

```
tag1 ~ tag2 {
```

```
  proprietàZ: value;
```

```
}
```

per selezionare OGNI elemento <tag2> preceduto da un elemento <tag1>

```
div + h3 {
```

```
  text-align: right;
```

```
  color: firebrick;
```

```
}
```

```
/*seleziona il tag h3 DOPO un tag  
div di chiusura*/
```

```
h2 ~ h3 {
```

```
  background-color: aqua
```

```
}
```


```
/*seleziona tutti i tag h3  
preceduti da un tag h2*/
```


Pseudo-Classi (:)

Le pseudo-classi sono usati per definire uno stato particolare di un tag.

Esempio: tag <a>

Possono essere anche combinate con classi CSS!


`a.classNameElemento:hover {
 color: yellowgreen;
}`

```
a:link {  
    color: darkblue;  
} /*Link non visitato*/  
  
a:visited {  
    color: mediumpurple;  
} /*Link visitato*/  
a:hover {  
    color: yellow;  
} /*Mouse sul link */  
  
a:active {  
    color: magenta;  
} /*Link selezionato (clic con  
il destro sopra per es.)*/
```

Pseudo-Elementi

Gli pseudo-elementi sono elementi per consentire una formattazione più semplice.

Pseudo Elemento	Descrizione
::first-name	Prima riga di un elemento
::first-letter	Prima lettera di un elemento
::before	Inserire del contenuto prima dell'elemento
::after	Inserire del contenuto dopo dell'elemento

Esempio

```
p::first-letter {  
  color: blueviolet;  
  font-size: xx-large;  
}
```

Disposizione degli elementi

La disposizione degli elementi viene gestita tramite tre proprietà:

- Float
- Visibility: visibilità dell'elemento (hidden per nascondere)
- Display: gestisce rendering dell'elemento (inline, block, inline-block, none)

Proprietà Display

Ricorda:

- In-line: elementi in linea (img, span, a)
- Block: contenitori che delimitano spazio (div, section, p, footer, h1...h6, form...)

Mostrare un elemento in-line come block:

```
tag {  
    display: block;  
}
```

Mostrare un elemento block come in-line:

```
tag {  
    display: in-line;  
}
```

`display: none;` → elemento nascosto

Proprietà Float e Clear

- **Float** : per ridisporre l'elemento con altri

```
tag {  
  float: left; /*right, none*/  
}
```

- **Clear**: “pulire” lo spazio attorno all'elemento

```
tag {  
  clear: left; /*right, none, both*/  
}
```

Posizionamento Assoluto

Elementi potrebbero essere sovrapposti.

Le istruzioni sono:

- Position: tipo di posizionamento (static, absolute, relative, fixed)

```
tag{  
position: relative;  
top: -20px;  
}
```

- Z-Index: gestione degli elementi in primo piano, secondo piano.

Box Model

Ogni elemento ha una dimensione data dalla somma dei seguenti parametri:

- **Border:** bordo dell'elemento
- **Padding:** spazio tra contenuto dell'elemento e il bordo
- **Margin:** spazio tra il bordo dell'elemento e quelli circostanti



Gestione del Border

È possibile impostare le seguenti caratteristiche:

Proprietà	Descrizione
border-width	Dimensione del bordo
border-style	Stile del bordo (dotted line, solid line)
border-color	Colore del bordo
border-radius	Curvatura degli angoli

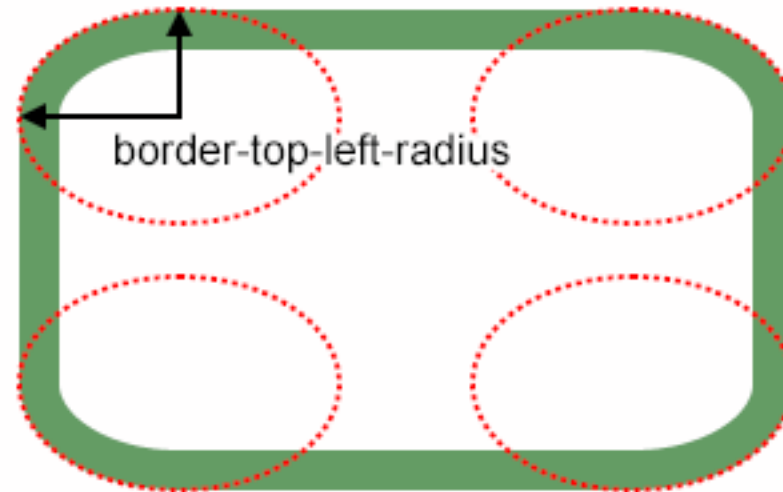
Esempio

```
p {  
  border-style: dotted;  
  border-width: 5px;  
}
```


Bordi arrotondati

Attraverso l'attributo border-radius

```
.border {  
    border-radius: 50px;  
}
```



Gestione Tipografica

Alcune delle principali impostazioni per la formattazione sono le seguenti:

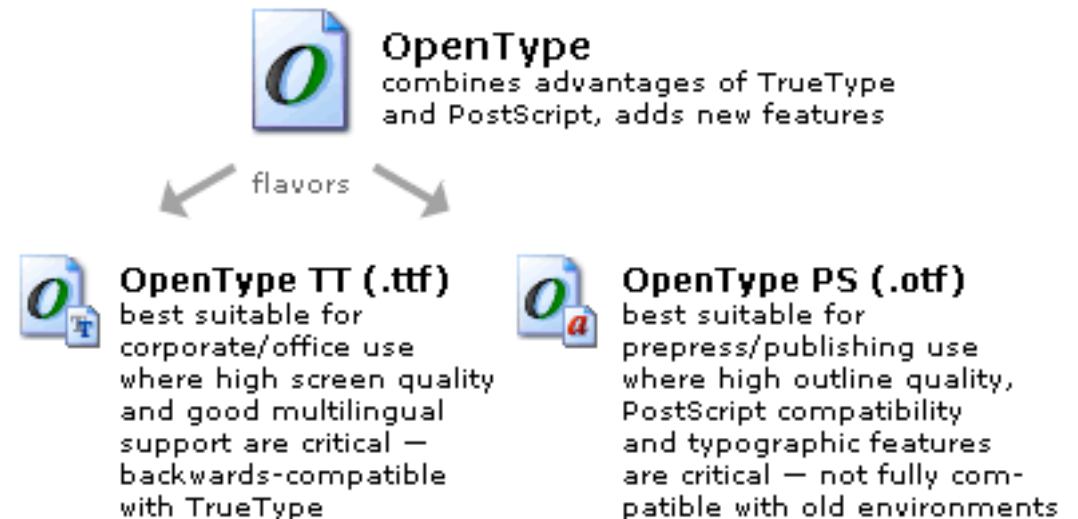
Proprietà	Descrizione
Font-family	Tipo di carattere (font)
Font-size	Dimensione del carattere
Color	Colore del testo
Font-weight	Grassetto
Font-style	Stile del font (italic)
Text-transform	Trasformazione testo (capitalize, uppercase...)
Text-decoration	Sottolineatura e Barrato
Text-Align	Allineamento testo

Supporto tipografico

Font in formato OTF (Open Type Font, evoluzione di True Type Font)

[<http://en.wikipedia.org/wiki/OpenType>]

```
@font-face {  
    font-family: 'MyFont';  
    src: url(MyFont.otf);  
}
```



Suddivisione testo in colonne

È possibile definire il testo in più colonne tramite alcune proprietà di stile del contenitore testuale quali:

- Column-count: conteggio delle colonne
- Column-gap: spazio tra colonne
- Column-rule: stile per la divisione in colonne

Troncamento di testo

Già supportato da IE, diventa parte delle specifiche:

```
.text {  
    text-overflow: ellipsis;  
}
```

Without overflow: Looooooooooooooooooooooooooooo text
.overflow-box: Looooooooooooooooooooo...

Media Query

Utilizzate per cambiare lo stile e adattare il contenuto in base al dispositivo che lo utilizza.

L'idea è creare un CSS per ogni tipo di visualizzazione che si intende supportare creando delle regole in base alle quali sarà il browser a decider quale CSS utilizzare.

Tali regole includono la definizione del tipo di device e delle sue caratteristiche.