

CASCADING STYLE SHEETS

CSS

Cascading Style Sheet

- I **fogli di stile** o **CSS** (acronimo di *Cascading Style Sheets*) sono uno strumento di formattazione delle pagine web che consente di personalizzare l'aspetto della pagina senza agire direttamente sul codice HTML della stessa.
- E' il linguaggio standard per la stilizzazione di documenti HTML che supplisce ai tag di formattazione dell'HTML stesso e conferisce maggiore potenzialità di stilizzazione ad una pagina Web.
- L'introduzione dei fogli di stile si è resa necessaria per separare i contenuti dalla formattazione.

Cosa si può fare con i CSS

Grazie ai CSS è possibile:

- definire i margini del documento;
- definire lo stile del testo;
- dare l'allineamento al testo;
- gestire lo sfondo della pagina o di singoli elementi;
- gestire elementi testuali e d'immagine;
- creare effetti di transizione;
- rendere dinamici i link testuali;
- ecc.
- Grazie all'uso di **CSS esterni**, inoltre, si può modificare l'aspetto di tutte le pagine web di un sito modificando il solo file di stile.

Sintassi CSS

- La più semplice regola css è composta di tre parti: un selettore, una proprietà ed un valore:

`selettore{proprietà: valore}`

- Il selettore può essere **l'elemento html** che si vuole definire
- La proprietà è l'attributo che si vuole cambiare
- Alla proprietà è associato un valore
- La proprietà e il valore sono separati da due-punti : e racchiusi tra **parentesi graffe**

`body {color: black}`

- Se ci sono più attributi sono separati da un **punto e virgola ;**

Come inserire un foglio di stile

Fondamentalmente sono tre i modi di usare i css, quale sia il migliore dipende soltanto da voi, dalla vostra organizzazione mentale e dalle vostre reali esigenze:

- Fogli di stile esterni
- Fogli di stile interni
- Fogli di stile in linea (inline)

Fogli di stile esterni

- I fogli di stile esterni sono quelli più utilizzati in quanto meglio interpretano la filosofia dei fogli di stile.
- ideale se l'applicazione ha molte pagine, in effetti si può cambiare l'aspetto di un intero sito modificando un solo file
- Ogni pagina deve essere collegata al foglio di stile
- Si deve inserire il tag `<link>` all'interno della sezione head della pagina HTML

`<head>`

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

`</head>`

- L'attributo **REL** indica che il file collegato è un foglio di stile (stylesheet)
- L'attributo **HREF** richiama il percorso del file esterno
- L'attributo **TYPE** definisce il linguaggio del foglio di stile.
- Il browser leggerà le definizioni di stile dal file `miostile.css`, e formatterà il documento di conseguenza

Scrivere un css esterno

- Un foglio di stile esterno può essere scritto in qualsiasi editor di testo
- Il file non deve contenere alcun tag html
- Il foglio di stile deve essere salvato con estensione **.CSS**
- Un esempio di foglio di stile:

```
body {  
background-image: url("immagini/miosfondo.gif")  
}  
p {margin-left: 20px}
```

Fogli di stile interni

- Un foglio di stile interno può essere usato quando un documento html ha uno stile unico
- Si definisce nella sezione head con il tag `<style>`
`<head>`
`<style type="text/css">`
`hr {color: red}`
`p {margin-left: 20px}`
`body {background-image: url("immagini/foto.jpg")}`
`</style>`
`</head>`
- Il browser leggerà le definizioni di stile, e formatterà il documento in accordo ad esse

Stile inline

- Uno stile inline fa perdere molti dei vantaggi dei fogli di stile, perché mischia il contenuto con la presentazione
- Metodo da usare quando uno stile deve essere applicato ad una sola occorrenza di un elemento in un documento
- Per definire uno stile inline si usa l'attributo style dell'elemento da formattare
- L'attributo style può contenere qualsiasi proprietà css
- L'esempio mostra come cambiare il colore e il margine sinistro di un singolo paragrafo

```
<p style="color: green; margin-left: 20px"> Testo del paragrafo </p>
```

Regole

- Se il **valore** è composto da più parole, bisogna racchiuderlo tra virgolette doppie " "
- Se si vuole specificare **più di una proprietà**, bisogna separarle con un **punto-e-virgola** ;
- L'esempio mostra come definire un paragrafo con allineamento centrato e colore del testo rosso

```
p {  
  text-align: center;  
  color: red;  
  font-family: "sans serif";  
}
```

Raggruppare i selettori

- Si possono **raggruppare i selettori**: bisogna separare i selettori con una **virgola**
- Nell'esempio sotto, gli elementi di titolo sono raggruppati; ogni elemento di titolo sarà verde

```
h1, h2, h3, h4, h5, h6  
{  
  color: green;  
}
```

Classi

Le classi servono per creare parametri alternativi a quelli indicati nei fogli di style.

Un esempio pratico sono i titoli o gli esempi, che di solito hanno un formato diverso da quello scritto normalmente.

Le classi si possono inserire in fogli esterni (richiamabili da ogni pagina), oppure in una sola pagina (fra i tag head).

La sintassi per l'inserimento è uguale agli altri parametri di style

Classi

SINTASSI:

Per creare una classe (sia in foglio esterno che interno) dobbiamo solo scrivere il **nome della classe preceduto da un punto** e seguito dalle parentesi graffe.

.esempi { font-family : Arial, verdana, Helvetica; font-size:12px; color:#031D81; font-weight:bold; font-style:italic; }

Nota: i nomi delle classi devono iniziare con un carattere alfabetico e non con un numero e non devono contenere spazi vuoti.

Per richiamare una classe dobbiamo solo scrivere:

**

Testo.....

**

I selettori ID (Identificatori)

- L'uso degli ID, così come quello delle classi, può essere associato a qualsiasi elemento valido di HTML anche se di norma lo si adopera con l'elemento vuoto `<div>`.
- Un selettore ID può essere adoperato una sola volta all'interno della stessa pagina, mentre una classe non ha praticamente alcun limite.

I selettori ID (Identificatori) /2

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>
```

Per **applicare l'attributo ID** ad un elemento basta inserirlo subito dopo il tag di apertura dell'elemento stesso.

L'uso degli ID, così come quello delle classi, può essere associato a qualsiasi elemento html valido anche se di norma lo si adopera con l'elemento **<DIV>**. L'elemento (div) non ha molto significato in html, ed infatti viene adoperato proprio con i css, specialmente per la gestione delle immagini e dei blocchi di testo.

Ereditarietà

Secondo questo meccanismo le impostazioni di stile applicate ad un elemento vengono ereditate anche dai suoi discendenti. Almeno fino a quando, per un elemento discendente, non si imposti esplicitamente un valore diverso per quella proprietà.

Vediamo di chiarire con un esempio. Impostiamo il colore grigio scuro per il testo a livello dell'elemento body:

```
body {color: #222;}
```

Tutti gli elementi discendenti di body, erediteranno questa impostazione. Ma se ad un certo punto definiamo nel codice del CSS un selettore con la proprietà color: white; l'ereditarietà viene spezzata:

```
body {color: #222;}
```

```
li {color: white;}
```

Gli elementi li avranno perciò il testo bianco.

ereditarietà

- In HTML un documento qualsiasi presenta una struttura fatta ad albero, il quale si compone di elementi "padre" e "figli"; Ci sono molte ragioni per avere i documenti così strutturati; nel caso particolare dei fogli di stile c'è una particolare ragione : "**L'ereditarietà**". Così come fanno nel mondo umano i figli ad ereditare determinate proprietà dai loro padri, allo stesso modo gli elementi in HTML ereditano determinate proprietà; lo stesso discorso vale per Css in questo caso ciò che viene ereditato sono delle proprietà stilistiche. **Il meccanismo** che consente questo passaggio di proprietà è proprio l'ereditarietà.
- Come si è potuto capire, **l'ereditarietà è un veicolo di trasporto che distribuisce le proprietà stilistiche agli elementi discendenti.**

Peso e origine

- Con peso si definisce la diversa importanza di una regola rispetto ad un'altra. Sono varie le cause del diverso **peso delle regole**.
-
- **Origine**
 - Un criterio di importanza è dato dall'origine del foglio di stile:
 - foglio dell'autore
 - foglio dell'utente
 - foglio predefinito del browser
 - Il foglio dell'autore è più importante del foglio dell'utente che a sua volta prevale sul foglio predefinito del browser.
-
- **Specificità**
 - Criteri di importanza in funzione della tipologia degli elementi, in ordine di importanza:
 - id (#titolo)
 - classi e pseudo-classi (.titolo)
 - elementi (h1)
-
- **Collocazione**
 - A secondo di dove la regola è definita, in ordine di importanza:
 - in linea
 - in CSS incorporati
 - in CSS collegati
-
- **Importanza**
 - La parola chiave !important applicata ad una dichiarazione la porta al primo posto nella gerarchia di importanza, a prescindere da origine, specificità e collocazione

concetto di cascade

- Visto che siamo nel mondo dei fogli di stile a cascata, introduciamo adesso il concetto di cascade, o meglio tentiamo di capire il procedimento di un browser quando incontra un foglio di stile e lo fa visualizzare sul monitor del nostro computer. Il browser quando riceve una pagina dal server, esegue un "filtraggio" degli stili. Inizialmente valuta il valore dell'attributo media (se è presente) e quindi imposta le prime regole di visualizzazione. Successivamente verifica la presenza di eventuali stili definiti dall'utente, (come la stampa) e comincia ad ordinare i vari elementi in base al loro peso e alla loro origine. Infine ordina le varie dichiarazioni in questo modo: gli stili in linea hanno più importanza di quelli incorporati che a loro volta prevalgono su quelli collegati
- Bisogna inoltre sapere che se dopo la dichiarazione è presente la parola !important, essa balza al primo posto nell'ordine di applicazione a prescindere da peso, origine, specificità e ordine. In questo [esempio](#) è chiaro il concetto.

Commenti

- Si possono inserire nel css commenti che possono essere d'aiuto quando il codice sorgente deve essere in seguito modificato
- I commenti vengono ignorati dal browser
- Un commento comincia con /* e termina con */

```
/* commento*/  
p {  
  text-align: center;  
  /* altro commento */  
  color: black;  
  font-family: arial;  
}
```

Sfondo

- **background-color** – Imposta il colore di sfondo di un elemento
- **background-image** – Imposta un'immagine come sfondo – `url(...)`
- **background-repeat** – Imposta la ripetizione di una immagine di sfondo – **repeat**, **repeat-x**, **repeat-y**, **no-repeat**
- **background-attachment** – Immagine di sfondo fissa o che scorre con il testo – `scroll`, `fixed`
- **background-position** – Posizione di partenza di una immagine di sfondo – `top left`, `top center`, `top right`, `center left`, `center center`, `center right`, `bottom left`, `bottom center`, `bottom right`, `x-% y-%`, `x-pos y-pos`

Sfondo

background-repeat

- **background-repeat: repeat | repeat-x | repeat-y | no-repeat**

Questo attributo serve per specificare come disporre l'immagine di sfondo quando non si tratta di uno sfondo omogeneo a tinta unita che riempie tutta la pagina:

repeat indica che l'immagine deve essere replicata in **orizzontale** ed in **verticale**.

repeat-x deve essere replicata soltanto in **orizzontale**.

repeat-y deve essere replicata soltanto in **verticale**.

no-repeat indica che l'immagine **non** deve essere replicata.

Logicamente questo attributo dovrà necessariamente essere adoperato in abbinamento con **background-image** es.:

```
{background-image: url(logo.gif); background-repeat: repeat;}
```

Sfondo

background-position

- **background-position: *coordinate***

Questo attributo definisce il punto da cui iniziare a posizionare l'immagine di sfondo e funziona se abbinato con `background-image` e `background-repeat`. Si possono adoperare parole chiave per la posizione orizzontale: *left*, *center*, *right* e parole chiave per la posizione verticale: *top*, *center*, *bottom*.

E' anche possibile esprimere valori percentuali, dove: *0% 0%* indica l'angolo in alto a sinistra (default) e: *100% 100%* indica l'angolo in basso a destra. Per maggior precisione si possono specificare anche coordinate espresse in mm, cm, inc. es.:

```
{background-image: url(logo.gif); background-repeat:  
repeat-y; background-position: 50% 50%;}
```

Sfondo

background-attachment

- `background-attachment: scroll | fixed`

Questo attributo definisce se l'immagine usata come sfondo deve scorrere sullo schermo insieme al testo (`scroll`) oppure restare fissa sullo sfondo (`fixed`) nel momento in cui si scorre la pagina verticalmente.

Esempi:

```
{background-image: url(logo.gif); background-attachment: scroll;} 
```

```
{background-image: url(logo.gif); background-attachment: fixed;} 
```


Testo

- **color** – Colore del testo
- **direction** – Direzione del testo – **ltr, rtl**
- **text-align** – Allineamento – **left, right, center, justify**
- **text-decoration** – Decorazione – **none, underline, overline, line-through, blink**
- **text-indent** – Indentazione della prima riga – **length, %**
- **text-shadow** – Ombra
- **text-transform** – Controlla le lettere – **none, capitalize, uppercase, lowercase**

Testo

text-align

- `text-align: left | right | center | justify`
- Allineamento del testo, valori possibili: sinistra, destra, centrato, giustificato.

```
p {text-align: left;}
```

```
p {text-align: right;}
```

```
div {text-align: center;}
```

```
div {text-align: justify;}
```

esempi →

Testo

text-align - esempi

`text-align: left;` paragrafo allineato a sinistra.

`text-align: right;` paragrafo allineato a destra.

`text-align: center;` paragrafo allineato al centro.

`text-align: justify;` paragrafo giustificato. Che cosa è la giustificazione? Guardate attentamente i due margini: destro e sinistro, potrete notare che il testo è allineato sia a destra che a sinistra; per fare questo il comando aggiunge automaticamente degli spazi vuoti fra le parole.

direction

- L'elemento **direction** imposta la direzione con cui il testo dev'essere stampato a video (da sinistra verso destra o viceversa).

La sintassi è la seguente:

- **p { direction: *valore*; }**
- I possibili **valori** che può assumere sono i seguenti:
- **ltr** - di default, da sinistra a destra;
- **rtl** - da destra a sinistra.

Testo

text-decoration

text-decoration: overline | underline | line-through | blink | none

Decorazione del testo, valori possibili: overline= soprallineato, underline= sottolineato, line-through= sbarrato, none= nessuno. E' anche possibile combinarne più di uno, per esempio inserendo sia underline che overline. None lo si adopera in particolar modo con i link in quanto toglie la sottolineatura che altrimenti avrebbero per default.

```
h1 {text-decoration: overline;}
p {text-decoration: underline;}
div {text-decoration: overline underline;}
p {text-decoration: line-through;}
p {text-decoration: blink;}
p {text-decoration: none;}
```

text-decoration - esempi

text-decoration: overline; - sopra lineato (qui non si vede perché non supportato da powerpoint)

text-decoration: underline; - sotto lineato

text-decoration: overline underline; - sopra e sotto lineato (qui non si vede perché non supportato da powerpoint)

~~text-decoration: line-through;~~ - barrato

text-decoration: blink; - lampeggiante (non supportato da IE)

text-decoration: none; - nessuna decorazione.

Testo

text-indent

text-indent: lunghezza | percentuale

Indentazione, rientro della prima riga di un blocco di testo dal margine sinistro. E' possibile fare uso anche di numeri negativi per un rientro esterno.

```
p {text-indent: 12px;}  
div {text-indent: 20%;}
```

text-indent: 5px

text-indent: 25px

Testo

text-transform

text-transform: capitalize | uppercase | lowercase

Capitalize trasforma in maiuscola la prima lettera di ogni parola, **uppercase** e **lowercase** trasformano in maiuscolo o minuscolo l'intero blocco di testo.

P {text-transform: capitalize;} P {text-transform: uppercase;} P {text-transform: lowercase;}

text-transform: capitalize; testo di prova

TEXT-TRANSFORM: UPPERCASE; TESTO DI PROVA

text-transform: lowercase; testo di prova

line-height

line-height: normal | numero | percentuale

Interlinea, altezza della riga del testo, valori possibili: normal, numero, percentuale. Con il valore **normal** il testo non avrà alcuna variazione. Il **numero** può essere espresso in em (lunghezza lettera m) o in **percentuale**.

p {line-height: 3em;} div {line-height: 50%;}

line-height: 3em

line-height: 50%

Font

- **font-style** – Stile, corsivo – normal, italic
- **font-variant** – Maiuscoletto o normale – normal, small-caps
- **font-weight** – Peso, grassetto – normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900
- **font-size** – Dimensione – xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %
- **font-family** – Lista con priorità di nomi di famiglie di font e/o nomi di famiglie generiche – family-name, generic-family (serif, sans-serif, cursive, fantasy, monospace)
- **font** – Imposta tutte le proprietà del font

font-family

`font-family: famiglia1, famiglia2, famiglia3;`

Si possono elencare più famiglie di caratteri per dare la possibilità al browser di selezionare quella presente sul PC del visitatore. Si tenterà di visualizzare la prima specificata per poi passare a quelle successive nel caso in cui non fossero presenti quelle precedenti. A questo proposito è opportuno chiudere la lista con una famiglia generica come sans-serif per essere certi di trovare un riscontro sempre e comunque.

Le famiglie si separano da una virgola e da uno spazio vuoto opzionale. Nel caso in cui il nome del font fosse composto da più di una sola parola, per esempio *MS Sans Serif*, questo dovrà essere racchiuso fra doppie virgolette "MS Sans Serif" nelle dichiarazioni sul foglio esterno e nella pagina, e da singoli apostrofi nel caso di dichiarazione in linea che farebbe già uso di doppie virgolette per accettare la dichiarazione stessa di style.

`p {font-family: verdana, arial, sans-serif;} div {font-family: "MS Sans Serif", verdana, sans-serif;}`

`<b style="font-family: 'MS Sans Serif', verdana, sans-serif;"> ... `

font-size

font-size: lunghezza | valore assoluto | valore relativo | percentuale

Si possono esprimere le dimensioni del font facendo riferimento ad uno dei quattro possibili attributi. Vedi anche diapositive 40, 41, 42 (**controllare il numero**).

Di solito si adoperano i punti (pt) o i pixel (px), ma qualsiasi altro parametro andrà sintatticamente bene.

```
p {font-size: 9pt;} i {font-size: larger;} b {font-size: 18px;}  
h1 {font-size: 150%;}
```

font-style

font-style: normal | italic | oblique

Si può eseguire il rendering (resa grafica) dei caratteri in corsivo.

```
h1 {font-style: normal;}
```

```
p {font-style: italic;}
```

```
b {font-style: oblique;}
```

font-variant

`font-variant: normal | small-caps`

Assegna il maiuscoletto con `small-caps` mentre `normal` per rimuovere il tutto.

```
p {font-variant: small-caps;}  
b {font-variant: normal;}
```

font-weight

`font-weight: bold | bolder | lighter | normal | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900`

Per impostare lo spessore del testo. Oltre alle parole riservate è possibile introdurre anche un valore numerico da 100 a 900 con incrementi di 100. Soltanto a partire dal valore 700 si avvertirà l'effetto grassetto.

```
P {font-weight: bold;}  
h1 {font-weight: 700;}
```

font

Usando la proprietà font è possibile raggruppare tutte le diverse proprietà dei font in una sola.

Per esempio immagina queste quattro linee di codice usate per descrivere le proprietà del font per il tag <p>:

```
p {  
font-style: italic;  
font-weight: bold;  
font-size: 30px;  
font-family: arial,  
sans-serif;  
}
```

Usando la proprietà per il raggruppamento il codice sarà semplificato:

```
p { font: italic bold 30px arial, sans-serif; }
```

L'ordine dei valori per font è:

font-style | font-variant | font-weight | font-size | font-family

Visibilità

- z-index – Ordine nello stack (rilievo, valori più alti sono in primo piano) – auto, number
- overflow – Impostazione per contenuto più ampio dell'area disponibile – visible, hidden, scroll, auto
- visibility – Visibilità – visible, hidden (occupa spazio)
- display – Visualizzazione – none (non occupa spazio), block, inline, ...

I Parametri

- I Parametri Sono quei valori che vanno specificati negli attributi e variano a seconda dell'attributo usato. Nelle tabelline riportate nelle pagine seguenti è possibile rendersi conto di come possono essere espressi, alcuni di essi sono equivalenti, producono cioè lo stesso risultato, in questo caso sarete voi a decidere quale adoperare in base ai vostri gusti necessità ed abitudini.

Unità di misura:

font-size: *esempio*;

Unità di misura	Descrizione	Esempio	Risultato
cm	Centimetri	font-size: 1cm;	Risultato
mm	Millimetri	font-size: 7mm;	Risultato
in	Pollici	font-size: 0,25in;	Risultato
pt	Punti	font-size: 12pt;	Risultato
pc	Pica(=12 punti)	font-size: 2pc;	Risultato
px	Pixel	font-size: 25px;	Risultato
em	Lun. lettera m	font-size: 2,5em;	Risultato
ex	Altezza carattere x	font-size: 3ex;	Risultato

Valori assoluti:

font-size: *valore*;

Valore	Sintassi descrizione	Risultato
xx-small	font-size: xx-small ;	Risultato
x-small	font-size: x-small ;	Risultato
small	font-size: small ;	Risultato
medium	font-size: medium ;	Risultato
large	font-size: large ;	Risultato
x-large	font-size: x-large ;	Risultato
xx-large	font-size: xx-large ;	Risultato

Valori relativi:

- **font-size:** *valore;*

Valore	Sintassi descrizione	Risultato
smaller	font-size: smaller ;	Risultato
larger	font-size: larger ;	Risultato

Colori:

color: *valore*;

Valore	Sintassi descrizione	Risultato
nome	color: green	Risultato
<i>valore</i> esadecimale	color: #008000;	Risultato
<i>valore</i> RGB	color: rgb(0,128,0);	Risultato
<i>percentuale</i> RGB	color: rgb(0%,50%,0%);	Risultato

Posizionamento Box

Attributi e Parametri per definire e posizionare un box, esempio di riferimento:

`<div style="position: absolute; left: 100px; top: 50px; z-index: 5;"> contenuto </div>`
con valori e parametri riportati nella tabella sotto.

Attributo	Parametro	Descrizione
position	<i>absolute - static - relative</i>	Il modo in cui può essere posizionato un box contenitore.
left	<i>cm - mm - in - pt - pc - px - ex - em</i>	la distanza del contenitore dal margine sinistro della finestra del browser.
top	<i>cm - mm - in - pt - pc - px - ex - em</i>	la distanza del contenitore dal margine superiore della finestra del browser.
height	<i>cm - mm - in - pt - pc - px - ex - em</i>	Altezza del contenitore se posizionato in modo absolute.
width	<i>cm - mm - in - pt - pc - px - ex - em</i>	Larghezza del contenitore se posizionato in modo absolute.
overflow	<i>visible - hidden - scroll</i>	Contenuto che supera Larghezza Altezza del box contenitore.
visibility	<i>visible - hidden -</i>	Visibilità o meno di un box contenitore.
Z-index	<i>xx (numeri positivi)</i>	L'ordine di priorità di visualizzazione di un box contenitore.

Note e chiarimenti!!!

- Le pagine che seguono sono di chiarimento

L'origine di un foglio di stile

- Prima di analizzare il *cascade*, è bene vedere le possibili origini di un foglio di stile:
- **autore**: è lo stile definito nel sorgente HTML della pagina dal webmaster o dal grafico del sito
- **browser**: i browser conformi agli standard del World Wide Web Consortium devono avere un foglio di stile predefinito da assegnare alle pagine
- **utente**: taluni browser mettono a disposizione all'utente la possibilità di creare un foglio di stile personalizzato, magari tramite un'interfaccia grafica e traducendo poi le impostazioni dell'utente in dichiarazioni CSS
- È possibile quindi che questi fogli di stile si sovrappongano e in questo caso vengono applicati in base al *cascade*.

Ereditarietà

In base al principio di ereditarietà le regole stilistiche applicate ad un elemento vengono ereditate dai suoi discendenti fino a quando non interviene una nuova regola. Esempio, se impostiamo il colore blu per il testo nell'elemento BODY (`color: 0000ff;`), tutti gli altri elementi suoi discendenti erediteranno questa impostazione. Non tutte le proprietà però sono ereditate.

L'importanza di una regola CSS (CascadingStyleSheet) è data prima di tutto dall'origine del foglio di stile, l'ordine è il seguente:

- foglio dell'autore
- foglio dell'utente
- foglio predefinito del browser

A seguire **l'ordine di importanza** tra gli stili definiti dall'autore è:

- stile in linea
- stile incorporato
- stile collegato

A seguire l'ordine di applicazione di una regola CSS viene influenzato da **peso e specificità**.

La specificità descrive il peso relativo delle varie regole all'interno di un foglio di stile, in