



# **Introduzione ai Fogli di Stile**

## ***Cascading Style Sheet (CSS) – logica e sintassi***

Sviluppo di siti web – UD09

*prof. Giovanni Borga*

# Cascading Style Sheet - CSS

Dietro il semplice acronimo CSS (Cascading Style Sheets - Fogli di stile a cascata) si nasconde **uno dei fondamentali linguaggi standard del W3C.**

E' l'ideale complemento dell'HTML. Infatti, nelle intenzioni del W3C Consortium, l'HTML, così come la sua evoluzione, XHTML, deve essere visto semplicemente come un linguaggio strutturale privo di qualsiasi elemento riguardante la presentazione dei contenuti. Per questo obiettivo, ovvero definire l'aspetto visuale ed grafico di una pagina, lo strumento designato sono appunto i CSS.

Questo approccio è noto anche mediante l'espressione: **separare il contenuto dalla formattazione.**

Le versioni del linguaggio:

- **CSS1**: dicembre 1996.
- **CSS2**: maggio 1998 (non porta grandi stravolgimenti, ma molte aggiunte rispetto alla 1)
- **CSS3**: pubblicato come raccomandazione ma già ampiamente supportato

## Vantaggi dell'uso dei fogli di stile

Utilizzando i CSS si capisce ben presto che le possibilità offerte da questo linguaggio sono numerosissime.

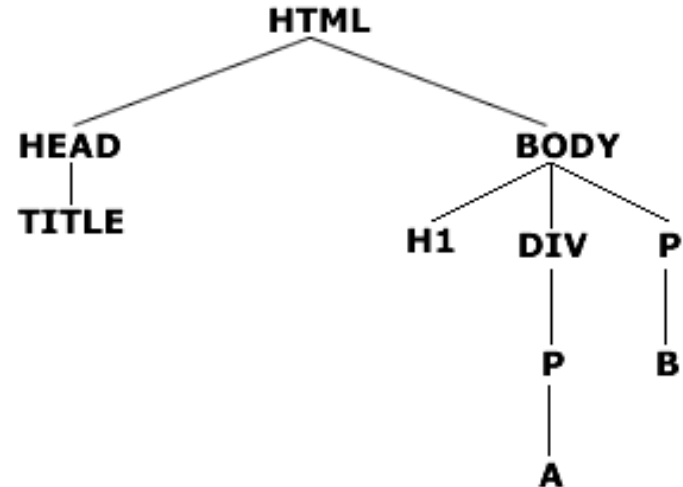
Sinteticamente vediamo i principali **punti di forza**:

- Quantità di **opzioni grafiche** e di layout estremamente più ampia di quanto offerto dall'HTML
- Gestione efficiente e versatile del **DOM** (Document Object Model)
- **Ottimizzazione del codice** (si definisce uno stile e lo si applica a tutti gli elementi – le modifiche possono poi essere effettuate solo sulle definizioni)
- **Interazione con lo scripting**

# Struttura del documento ed ereditarietà

Osserviamo il codice a sinistra; la sua rappresentazione gerarchica può essere l'immagine a destra:

```
<html>  
  <head>  
    <title>Struttura del documento</title>  
  </head>  
  <body>  
    <h1>Titolo</h1>  
    <div>  
      <p>  
        Primo <a href="pagina.htm">paragrafo</a>  
      </p>  
    </div>  
    <p>Secondo<b>paragrafo</b></p>  
  </body>  
</html>
```





# Gerarchia degli elementi di un documento

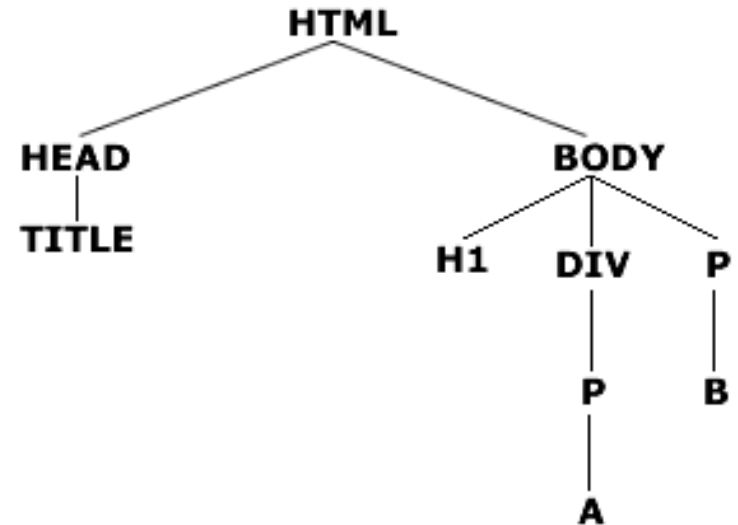
Il documento è in effetti una perfetta forma di gerarchia ordinata in cui tutti gli elementi hanno tra di loro una relazione del tipo **genitore-figlio**.

**PARENT-CHILD** in inglese; dicitura importante perchè utilizzata nei linguaggi come DOM o Javascript: **ogni elemento è genitore e/o figlio di un altro**.

- Un elemento si dice genitore (parent) quando contiene altri elementi.
- Si dice figlio (child) quando è racchiuso in un altro elemento.

In base a queste indicazioni il nostro documento ha le seguenti caratteristiche:

1. *BODY è figlio di HTML,*
2. *BODY è anche genitore di H1, DIV e P.*
3. *P è a sua volta genitore di un elemento B.*



# Gerarchia ed ereditarietà

Anche BODY è genitore di B? Non proprio.

Serve un'altra distinzione, quella tra **antenato** (*ancestor*) e **discendente** (*descendant*).

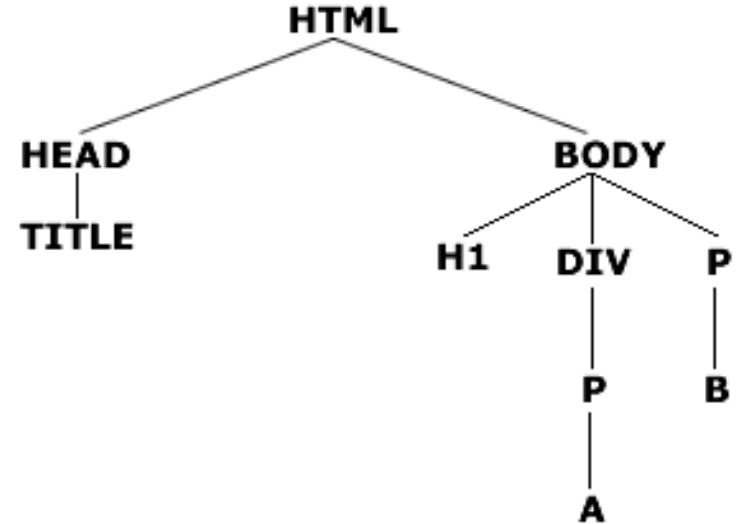
**La relazione parent-child** è valida solo se tra un elemento e l'altro si scende di **UN livello**. (Esattamente come in un albero familiare si indica la relazione tra padre e figlio).

- HEAD è figlio di HTML
- A è figlio di P
- ... ecc.

Tra **DIV e A**, invece si scende **di due livelli**: diciamo allora che **DIV è un antenato di A** e che questo è rispetto al primo un discendente.

C'è un solo elemento che racchiude tutti e non è racchiuso: HTML. Si dice infatti che HTML è l'elemento radice (root).

*NB: attenzione a non fraintendere: l'elemento radice di un documento (X)HTML non è BODY. HTML inoltre, non è una semplice dichiarazione ma a tutti gli effetti un TAG alla stregua di tutti gli altri.*



## Fogli di stile interni ed esterni

Abbiamo diversi modi per inserire i fogli di stile CSS in un documento.

Per capire il meccanismo è necessario chiarire la fondamentale distinzione tra **fogli esterni** e **interni**.

**E' esterno un foglio di stile definito in un file separato dal documento**

Come per gli HTML, si tratta di semplici documenti di testo ai quali si assegna l'**estensione .css**.

**Un foglio di stile è 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 di TRE DIVERSE TIPOLOGIE:

- a) **Collegati**
- b) **Incorporati**
- c) **In linea**

# CSS collegati

Per collegare un foglio esterno ad un documento esistono due possibilità. La prima e più compatibile è quella che fa uso del tag <LINK>. **La dichiarazione va sempre collocata all'interno della sezione <HEAD> del documento HTML:**

**<html>**

**<head>**

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

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

**</head>**

**<body>**

**...**

Attributi del tag <link>:

- **rel:** descrive il tipo di relazione tra il documento e il file collegato. (*per i CSS due sono i valori possibili: «stylesheet» e «alternate stylesheet»*).
- **href:** indica l'URL del foglio di stile.
- **type:** indica il tipo di dati da collegare. (*Per i CSS l'unico valore possibile è text/css*).
- **media:** indica il supporto (schermo, stampa, etc) cui applicare un particolare foglio di stile. *L'attributo è opzionale.*



## CSS collegati

Il secondo modo per collegare CSS esterni è usare la **direttiva @import** all'interno del tag <style>:

```
<html>  
  <head>  
    <title>Inserire i fogli di stile in un documento</title>  
    <style>  
      @import url(stile.css);  
    </style>  
  </head>  
<body>  
  ...
```

Questa modalità è una delle più sicure per risolvere problemi di compatibilità tra vecchi e nuovi browser. Si noti il CSS va collegato definendo un URL (assoluto o relativo) da racchiudere tra **parentesi tonde**, non servono doppi apici e che la dichiarazione deve chiudersi con un **punto e virgola**.

# CSS incorporati

I fogli incorporati sono quelli inseriti direttamente nel documento (X)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>
  ...
```

*(in rosso il tag, in blu le regole di stile)*

Attributi del tag <style>:

- **type** (obbligatorio)
- **media** (opzionale)

(Valgono le indicazioni viste in precedenza)

Dopo l'apertura di <style> si scrivono le regole del CSS seguite dalla normale chiusura </style>.

## CSS 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 attributes**», ovvero la famiglia di attributi applicabili a tutti i tag HTML.

La dichiarazione avviene a livello dei singoli tag contenuti nella pagina e per questo si parla di fogli di stile in linea (inline).

La sintassi generica è la seguente: **<tag style="regola\_di\_stile1; regola\_di\_stile2; ...">**

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>**

NB: 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.

# Note sul collegamento/incorporamento dei CSS

## Quando usare l'una o l'altra soluzione?

i risultati nella formattazione del documento non cambiano. La giusta soluzione sarà quindi quella richiesta dalla nostra applicazione.

Occorre dunque **progettare**, pensare in anticipo a quella che dovrà essere la struttura delle pagine del sito.

Le fasi da seguire per la progettazione possono essere le seguenti:

1. Si costruisce un **foglio di stile generico ed esterno**, da applicare a tutte le pagine del sito. Esso conterrà le regole per formattare gli elementi o le sezioni presenti in tutte le pagine.
2. Si analizzano sezioni ed elementi presenti solo in certe pagine o che vogliate modificare solo in determinati casi. (ad esempio, se occorre cambiare in rosso il colore di un titolo iniziale solo in una pagina delle decine che formano il sito).  
In questo caso si usa uno **stile incorporato solo in quella pagina**.
3. Infine se in particolari punti ci sono da introdurre formattazioni «spot» ed eccezioni si utilizza la tecnica dei **fogli di stile inline**.

# Note sul collegamento/incorporamento dei CSS

Se prendiamo come esempio un foglio di stile incorporato:

```
<style type="text/css">  
  h1 {color: red; }  
</style>
```

Per la legge che **regola il meccanismo del cascading** questo stile prevarrà su quello del CSS esterno.

Nel caso in cui le pagine siano relativamente numerose, l'operazione può essere più complicata perché in caso di modifiche agli stili già inseriti bisognerebbe ri-editarle ad una ad una.

È più efficace **collegare un secondo foglio di stile**. Se le regole del secondo foglio entrano in conflitto con quelle del principale si può utilizzare uno stratagemma: collegare il secondo CSS con la tecnica @import:

```
<link rel="stylesheet" type="text/css" href="stile.css">  
<style type="text/css">  
  @import url(nuovo_stile.css);  
</style>
```

**Le regole dello stile collegato con @import prevalgono su quelle ereditate dal tag <link>.**



# L'attributo MEDIA

Abbiamo accennato all'attributo MEDIA (opzionale). Grazie ad esso è possibile **impostare un foglio di stile per ogni supporto** su cui la nostra pagina verrà visualizzata.

Uno degli utilizzi più diffusi è quello di differenziare gli stili di visualizzazione da quelli di stampa.

L'attributo media può essere applicato sia a <link> sia a <style>:

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

```
<style type="text/css" media="screen">...</style>
```

E' possibile usare più di un valore, separando i nomi della lista con una virgola:

```
<link rel="stylesheet" type="text/css"
media="print, tv, aural" href="print.css" />
```

## Valori validi per MEDIA:

- **all** (default). *Il CSS si applica a tutti i dispositivi di visualizzazione.*
- **screen**. *Stili per i normali browser web.*
- **print**. *Stili per le stampe del documento.*
- **projection**. *Stili per la presentazione / proiezione del documento.*
- **aural**. *Stili per dispositivi a sintesi vocale.*
- **braille**. *Stili per supporti braille.*
- **embossed**. *Stili per stampanti braille.*
- **handheld**. *Stili per dispositivi mobili.*
- **tty**. *Stili per dispositivi a carattere fisso.*
- **tv**. *Stili per le Web-tv.*

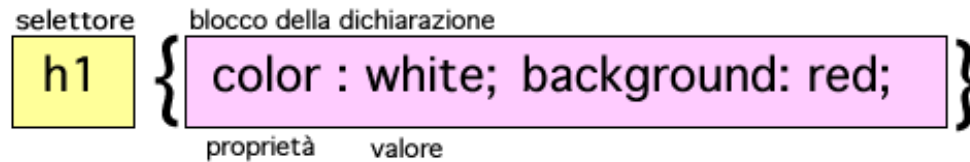


# Sintassi dei fogli di stile

# Struttura di una REGOLA

Questa è la tipica struttura di una **REGOLA** CSS.

Una regola esprime come deve essere l'aspetto di un elemento della pagina HTML.



Il SELETTORE indica **quale elemento** deve essere renderizzato con la regola.

Il BLOCCO delle dichiarazioni indica **come** renderizzare l'elemento. Il blocco può contenere una o più proprietà dell'elemento e relativo aspetto.

**E' importante imparare fin da subito il ruolo dei TRE DELIMITATORI: la parentesi graffa, i due punti e il punto e virgola.**

# Blocchi delle dichiarazioni

Il blocco delle dichiarazioni è **delimitato rispetto al selettore da due parentesi graffe**.  
Al suo interno possono trovare posto più dichiarazioni che sono sostanzialmente delle coppie:

- **proprietà**
- **valore**

La proprietà definisce un aspetto dell'elemento (margini, colore di sfondo, etc) mentre il valore definisce il carattere di quell'aspetto.

**Proprietà e valore devono essere separati dai due punti.**

Valgono inoltre le seguenti regole sintattiche:

1. Non è possibile indicare più di una proprietà con riferimento allo stesso valore,
2. E' ammesso invece, per determinati casi, specificare più valori per la stessa proprietà:

```
body {color background: black;}
```

```
p {font: 12px Verdana;}
```

**Più dichiarazioni vanno separate dal punto e virgola.** (il punto e virgola dopo l'ultima dichiarazione è facoltativo. Alcuni browser meno recenti lo richiedono quindi è in generale consigliato metterla).

Gli spazi lasciati all'interno di una regola non influiscono sul risultato.

# Commenti

Per inserire parti di commento in un CSS si utilizzano i seguenti segni:

```
/* come segno di apertura  
*/ come segno di chiusura
```

Questa sintassi vale sia per commenti di una sola riga sia per più righe di commento. Ecco un esempio:

```
<style type="text/css">  
  /* questo è lo stile per il body */  
  body {  
    background: #FFFFCC;  
  }  
</style>
```

**NB: nei fogli di stile incorporati questa sintassi è ammessa solo all'interno del tag `<style>`, al di fuori si deve utilizzare la sintassi dei commenti HTML.**



## Proprietà singole e a sintassi abbreviata

Nelle definizioni delle regole è possibile, per alcune proprietà, utilizzare il riferimento ai **singoli valori** oppure utilizzare una **sintassi abbreviata**. Vediamo ad esempio la proprietà `margin`; la sintassi a proprietà singola è la seguente:

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

Ma è anche possibile quest'altra sintassi (attenzione la sequenza: top-right-bottom-left) :

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

***Le proprietà per le quali è possibile utilizzare la sintassi abbreviata sono le seguenti:***

*background | border | border-top | border-right | border-bottom | border-left | cue | font | list-style |  
| margin | outline | padding | pause |*

# Selettori di elementi (type selector)

Fondamentalmente una regola CSS viene applicata ad un selettore. Il selettore è una semplice dichiarazione che serve a **selezionare la parte** o le parti di un documento soggette ad una specifica regola.

Il più semplice dei selettori è il **TYPE SELECTOR**. E' una dichiarazione che coincide con i nomi dei tag HTML. Vediamo un esempio:

```
h1 {color: #000000;}
```

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

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

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;}
```

Questa regola attribuisce il colore bianco ai primi tre livelli di heading.

## Selettori speciali (classi e ID)

Esistono due attributi fondamentali applicabili a tutti gli elementi HTML: **CLASS** e **ID**.

Dichiarare questi attributi a prescindere dai CSS non ha alcun senso e non modifica in alcun modo la presentazione della pagina. Dobbiamo infatti definire i corrispondenti selettori in un foglio di stile.

**I valori per CLASS e ID sono etichette che possiamo liberamente definire.** Vediamo un esempio chiarificatore; un paragrafo caratterizzato da un attributo CLASS valorizzato con «testorosso»:

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

Nell'HEAD del documento potremmo trovare un CSS incorporato con il seguente codice:

```
<style type="text/css">
.testorosso {
    font: 12px arial, Helvetica, sans-serif;
    color: #FF0000;
}
</style>
```

In tutto il documento i paragrafi avranno il testo di colore rosso.

**ID ha lo stesso comportamento di CLASS;** la scelta tra i due dipende dalla logica del documento.

## Il selettore speciale CLASS

Per definire una classe si usa far precedere il nome da un semplice **punto**:

**.nome\_della\_classe**

Questa è la sintassi di base. Un selettore classe così definito può essere applicato a tutti gli elementi di un documento HTML.

Il nome della classe può essere combinato con un type selector; ad esempio:

**p.testorosso {color: red;}**

**lo stile verrà applicato solo ai paragrafi che presentino l'attributo class="testorosso".**

E' importante stabilire delle strategie.

Questo secondo tipo di sintassi combinata va usata solo se si pensa di applicare una classe ad uno specifico tipo di elemento (solo paragrafi o solo div, e così via).

Se invece si intende applicare la classe a tipi diversi va utilizzata la sintassi base.

E' ammessa l'attribuzione di più classi allo stesso elemento utilizzando lo spazio come separatore:

**<p class="testorosso testobold testoitalic">....</p>**

## Il selettore speciale ID

La sintassi di un selettore ID è semplice come la precedente: è sufficiente far precedere il nome dal simbolo di cancelletto #:

**#nome\_id**

Con questa regola usiamo il selettore ne modo classico:

**#titolo {color: blue;}**

assegniamo il colore blue ad un qualsiasi tag che abbia attributo id="titolo".

Ma come per le classi è possibile usare una sintassi combinata:

**p#nome\_id {color: blue;}**

**In realtà il selettore ID è stato pensato per essere utilizzato in modo univoco in un documento. Serve cioè per identificare in modo preciso un elemento di una pagina HTML.**

Di conseguenza la sintassi combinata perde sostanzialmente di senso e significato.

NB: per lo stesso motivo **NON** è ammessa l'attribuzione di più ID allo stesso elemento!





# Valori e unità di misura

## Regole di base per i valori e le unità di misura

1. I **valori** di una proprietà **non vanno mai messi tra virgolette**.  
*Uniche eccezioni i valori espressi da stringhe di testo e i nomi dei font formati da più di una parola (esempio: "Times New Roman").*
2. Quando si usano **valori numerici con unità di misura, non bisogna lasciare spazio tra numero e sigla** dell'unità.  
*E' corretto quindi scrivere 15px oppure 5em.  
E' invece sbagliato usare 15 px o 5 em. In questi casi la regola sarà ignorata o mal interpretata.*

# Unità per le dimensioni

Le unità di misura usate per definire dimensioni, spazi o distanze sono 8. (Nella pratica comune solo alcune di esse sono realmente usate).

- **in** (inches - pollici) classica misura del sistema metrico americano, quasi mai utilizzato per definire il render a video, molto più utile per gli stili orientati alla stampa.
- **cm** (centimetri): come per i pollici, si tratta di un'unità molto poco utilizzata.
- **mm** (millimetri): valgono le considerazioni fatte per pollici e centimetri.
- **pt** (points - punti): unità di misura tipografica destinata essenzialmente a definire la dimensione dei font.
- **pc** (picas): unità poco usata. 1 pica equivale a 12 punti.
- **em** (em-height): unità di misura molto usata dagli autori CSS. 1 em equivale all'altezza media di un carattere per un dato font. E' un unità di misura relativa.
- **ex** (ex-height): poco usata. 1 ex equivale all'altezza del carattere x minuscolo del font scelto.
- **px** (pixels): unità di misura ideale su monitor. E' quella più usata e facile da comprendere.

# Percentuali, colori, stringhe, URI

## Percentuali

Un valore espresso in percentuale è da considerare **sempre relativo rispetto ad un altro valore**, in genere quello espresso per l'elemento parente. Si esprime con un valore numerico seguito (senza spazi) dal segno di percentuale: **60%** è pertanto corretto, 60 % no.

## Colori

I diversi modi per esprimere il valore di un colore sono esattamente corrispondenti alle **regole dell'HTML**.

## Stringhe

Alcune proprietà dei CSS possono avere come valore una stringa di testo. I valori espressi da stringhe vanno sempre racchiusi **tra virgolette**. Le proprietà in questione sono tre: content, quotes, text-align (ma solo per le tabelle definite con i CSS).

## Valori URL

Si tratta di indirizzi che puntano a documenti esterni (in genere immagini, come negli sfondi). Possono essere assoluti o relativi. Nel secondo caso il path fa sempre riferimento alla posizione del foglio di stile e non del documento HTML. La definizione dell'indirizzo è sempre introdotta dalla **parola chiave url** e va inserita tra parentesi tonde **senza virgolette**. Esempio: **url(immagini/sfondo.gif)**.



# Ereditarietà, cascade, conflitti tra stili



# Le leggi fondamentali dei fogli di stile

## *Le leggi di*

**Ereditarietà**  
**Cascade**  
**Conflitti tra stili**

*sono alla base del meccanismo di funzionamento dei fogli di stile.*

*Sono le leggi che definiscono l'interdipendenza delle regole di stile  
all'interno di un documento HTML*

# Ereditarietà

Secondo il meccanismo dell'ereditarietà:

**le impostazioni stilistiche applicate ad un elemento ricadono anche sui suoi discendenti.**

Questo **fino a quando**, per un elemento discendente, non si imposti esplicitamente un valore diverso per quella proprietà.

*Se impostiamo ad esempio il colore rosso per il testo (`color: red;`) a livello dell'elemento `BODY` tutti gli altri elementi suoi discendenti ereditano questa impostazione. Ma se ad un certo punto definiamo nel codice del CSS un selettore con la proprietà `color: black;` l'ereditarietà viene spezzata.*

**Non tutte le proprietà sono ereditate;** in genere sono quelle attinenti la formattazione del box model: margini, bordi, padding, background le più importanti.

*I motivi sono in generale abbastanza logici: nel caso del bordo, ereditarlo è semplicemente senza senso. Se ne imposto uno, diciamo, per un paragrafo sarebbe assurdo che un elemento `<A>` o un testo in grassetto vengano circondati dallo stesso bordo!*

## Conflitti tra stili: peso e origine

Vediamo ora alcuni concetti fondamentali riconducibili all'ambito dei conflitti possibili tra gli stili e le regole. Osserviamo la situazione in cui esistono le seguenti regole in un CSS:

```
p {color: black;}  
.testo {color: red;}
```

E in una pagina HTML scrivo questo codice: `<p class="testo">Testo del paragrafo</p>`  
Avremo il testo del paragrafo rosso, non nero.

**Questo perchè il selettore CLASS prevale sul TYPE SELECTOR.**

Introduciamo dunque il concetto di **peso** che è di fatto la maggiore o minore importanza da assegnare a ciascuna regola.

Un secondo concetto è quello di **origine del foglio di stile**. Noi utilizzeremo dei fogli di stile creati come autori, che vengono appunto definiti «fogli dell'autore». Occorre sapere che questi prevalgono su altri due tipi detti «dell'utente» e «predefinito del browser» di cui non ci occuperemo.

**I fogli di stile dell'autore prevalgono su quelli utente e del browser.**

# Specificità

La specificità descrive il peso relativo delle varie regole all'interno di un foglio di stile. Quando il peso relativo di una regola è maggiore di una seconda regola che è in conflitto con la prima fa sì che la prima regola prevalga sulla seconda. Ma come calcoliamo questo peso specifico?

Per ottenere la specificità di una regola si compone una tripletta di numeri: a-b-c, dove a è il numero di selettori ID, b il numero di selettori CLASS, c il numero di TYPE SELECTOR.

Ad esempio:

**#titolo {color: black;}** ha un ID, 0 classi, 0 elementi. La tripletta è dunque 1-0-0

**.classe1 {color: green;}** ha invece 0 ID, 1 classe, 0 elementi. La tripletta è 0-1-0

**h1 {color: red;}** ha infine 0 ID, 0 classi, un elemento. La tripletta è: 0-0-1

**Vale la regola per cui gli ID pesano più delle CLASS che pesano più dei TYPE SELECTOR.**

Per cui il tag **<h1 id=class="titolo" class="classe1">** apparirà in nero.

NB: il calcolo della specificità si effettua a prescindere dalla numerosità degli elementi di ordine inferiore. La regola **#paragrafo {color: green;}** presenta la seguente specificità 1-0-0 ed è più importante di **div p {color: red;}** che ha la tripletta 0-0-2.

## Il concetto di «cascade»

Il concetto di «cascade» è forse quello **più importante** tra quelli che sono alla base dei fogli di stile, non a caso forma lo stesso nome dei CSS.

Per comprenderlo facciamo una piccola analisi del comportamento di un browser durante il rendering di una pagina con fogli di stile; ecco le operazioni:

1. **Scelta del target** (es. se c'è un CSS per lo screen e uno per il print, nel rendering a schermo verrà scartato il CSS per il print.).
2. **Matching delle regole** (vengono scartate tutte le regole che si riferiscono a elementi non presenti nel documento).
3. **Ordinamento di pesi e origine** (se il CSS è definito dall'autore, verrà usato quello altrimenti quello utente o quello del browser, inoltre si ordinano le regole in base al peso)
4. **Calcolo della specificità**. (calcola la specificità dei selettori e in caso di conflitto tra regole usa i criteri di prevalenza).

NB: in generale c'è assenza di conflitti. In questi casi, oppure quando c'è equivalenza di peso, origine e specificità, viene applicata la regola più vicina all'elemento nel codice del documento con il criterio:

**gli stili in linea prevalgono su quelli incorporati che a loro volta prevalgono su quelli collegati.**



# Importanza

Infine il concetto di importanza. Si tratta di una legge semplice e lineare:

Se una dichiarazione viene accompagnata dalla **parola chiave !important** essa balza al primo posto nell'ordine di applicazione a prescindere da peso, origine, specificità e ordine.

La sintassi di utilizzo è la seguente:

```
p { color: red !important }
```

La parola chiave va posizionata dopo il valore della proprietà separata da uno spazio. Attenzione a non dimenticare il punto esclamativo!