

Cascading Style Sheet

- I fogli di stile a cascata (detti anche semplicemente fogli di stile) vengono usati per definire la **rappresentazione** di documenti HTML, XHTML e XML.
- L'introduzione dei fogli di stile si è resa necessaria per **separare** i contenuti dalla formattazione.

Wikipedia

Alberto Ferrari

Separazione contenuto/layout

- I tag html erano stati in origine studiati per definire il contenuto di un documento html
- Il layout del documento doveva essere preso in carica dal browser, senza usare tag di formattazione

Alberto Ferrari

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}`

Alberto Ferrari

Come inserire un foglio di stile

- Fogli di stile esterni
- Fogli di stile interni
- Stile inline

Alberto Ferrari

Fogli di stile esterni

- Foglio esterno: ideale se l'applicazione ha molte pagine
 - Si può cambiare l'aspetto di un intero sito modificando un solo file
- Ogni pagina deve essere collegata al foglio di stile
 - Tag `<link>` all'interno della sezione `head`

```
<head>  
  <link rel="stylesheet" type="text/css"  
        href="mystile.css" />  
</head>
```
 - Il browser leggerà le definizioni di stile dal file `mystile.css`, e formatterà il documento di conseguenza
- 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}
```

Alberto Ferrari

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("imagini/foto.jpg")}
</style>
</head>
```
- Il browser leggerà le definizioni di stile, e formatterà il documento in accordo ad esse

Alberto Ferrari

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>
```

Alberto Ferrari

Ordine di priorità (cascata)

- I fogli di stile permettono di specificare informazioni di stile in vario modo
 - Gli stili possono essere specificati in un singolo elemento html, nella sezione *<head>* di una pagina html, o in un file css esterno
 - È possibile fare riferimento pure a diversi fogli esterni in una stessa pagina html
 - Quale stile sarà usato quando è specificato più di uno stile per un elemento html?
- Tutti gli stili vengono raccolti a *cascata* in un nuovo foglio di stile *virtuale* secondo le seguenti regole
 - Default del browser (priorità più bassa)
 - Foglio di stile esterno
 - Foglio di stile interno (nella sezione *<head>*)
 - Stile inline (nell'elemento html – priorità più alta)

Alberto Ferrari

Sintassi Css - 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";
}
```

Alberto Ferrari

Sintassi Css - 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;
}
```

Alberto Ferrari

Selettore di classe

- Con il selettore di classe si possono definire stili diversi per uno stesso tipo di elemento html
 - Es. due tipi di paragrafo: uno allineato a destra e uno allineato al centro

```
p.firma {text-align: right}
p.importante {text-align: center}
```
 - Bisogna impostare l'attributo *class* nel documento html

```
<p class="firma">
  Paragrafo che sarà allineato a destra
</p>
<p class="importante">
  Paragrafo che risulterà centrato.
</p>
```

Alberto Ferrari

classe

- Si può anche omettere il nome di tag nel selettore per definire uno stile da applicare a tutti gli elementi html di una certa classe
 - Allineare al centro tutti gli elementi di classe *importante*
- ```
.importante {text-align: center;}

```
- Sia l'elemento *h1* che quello *p* sono di classe *importante*
  - Entrambi gli elementi seguiranno le regole del selettore *.importante*
- ```
<h1 class="importante">
  Titolo centrato
</h1>
<p class="importante">
  Anche questo paragrafo centrato
</p>

```

Alberto Ferrari

Selettore id

- Il selettore *id* è diverso dal selettore *class*
 - Mentre un selettore *class* può applicarsi a **multi** elementi su una pagina, il selettore *id* si applica ad **un solo** elemento
 - Un attributo *id* deve essere unico all'interno del documento
 - La regola di stile qui sotto sarà applicata ad un elemento *p* che ha il valore di *id* posto a *para1*
- ```
p#para1 {
 text-align: center;
 color: red;
}

```
- La regola di stile qui sotto sarà applicata all'elemento avente come *id* il valore *xyz*
- ```
#xyz {color: green}
<h1 id="xyz">Some text</h1>

```

Alberto Ferrari

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;
}

```

Alberto Ferrari

Sfondo

- *background-color* – Imposta il colore di sfondo di un elemento
- *background-image* – Imposta una 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*
- *background* – Imposta tutte assieme le proprietà dello sfondo

Alberto Ferrari

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*

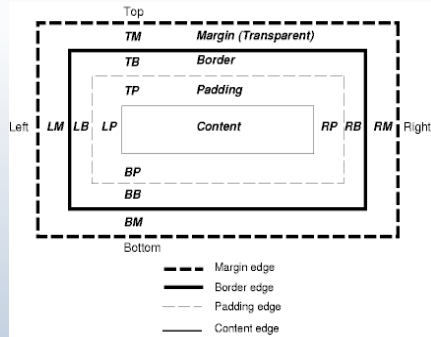
Alberto Ferrari

Font

- *font-style* – Stile, corsivo – *normal, italic*
 - *font-variant* – Maiuscolo 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 a tutte le proprietà del font
- Dimensioni
- px: misura in pixel (fissa rispetto a preferenze utente e risoluzioni monitor)
 - pt: punti tipografici, 1/72 pollice (fissa rispetto a preferenze utente)
 - em: rispetto alla dimensione di font attuale per elemento
 - %: rispetto alla dimensione di font dell'elemento genitore

Alberto Ferrari

Box model



Bordi

- **border-width** – Ampiezza dei quattro bordi – 1-4 valori ()
- **border-style** – Quattro bordi – 1-4 valori – *none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset*
- **border-color** – Colore – 1-4 valori – *color*
- **border** – Imposta tutte le proprietà (uguali) per i quattro bordi
 - `.important { border: 2px dashed red; }`
- **border-top, border-right, ...** – Imposta tutte le proprietà per il bordo superiore, destro, ...
 - `blockquote { border-left: 2px solid black; }`
- I bordi *top, right, bottom, left* possono essere cambiati indipendentemente usando proprietà separate: es. *border-top-width*

Alberto Ferrari

Margini esterni e interni

- **margin** – Spazio attorno agli elementi – 1-4 valori
 - È possibile usare valori negativi per sovrapporre il contenuto
 - I margini *top, right, bottom, left* possono essere cambiati indipendentemente usando proprietà separate, es. *margin-top*
 - Possibile usare scorciatoia *margin* per impostare tutti i margini
- **padding** – Spazio tra il bordo e il contenuto degli elementi – 1-4 valori
 - Non sono ammessi valori negativi
 - I padding *top, right, bottom, left* possono essere cambiati indipendentemente usando proprietà separate, es. *padding-top*
 - Possibile usare la scorciatoia *padding* per impostare tutti i padding

Alberto Ferrari

Dimensioni

- **width** – Ampiezza – *auto, length, %*
- **height** – Altezza – *auto, length, %*

Alberto Ferrari

Posizione

- **bottom** – Distanza del limite inferiore di un elemento rispetto all'elemento che lo contiene – *auto, %, length*
- **left** – Sinistra ... – *auto, %, length*
- **right** – Destra ... – *auto, %, length*
- **top** – Superiore – *auto, %, length*
- **vertical-align** – Allineamento verticale – *top, middle, bottom, baseline, sub, super, text-top, text-bottom, length, %*
- **position** – Modalità di posizionamento – *static, relative, absolute, fixed, inherit*
- **float** – Elementi posizionati a lato, circondati da testo – *left, right, none, inherit*

Alberto Ferrari

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, ...*

Alberto Ferrari