



Introduzione a CSS

Prof. Ing. Andrea Omicini

II Facoltà di Ingegneria, Cesena

Alma Mater Studiorum, Università di Bologna

andrea.omicini@unibo.it

Web Style Sheets

- ◉ Fogli di stile per il Web
- ◉ Scopi
 - ◉ descrivere come gli elementi di un documento devono essere "esposti", presentati
 - ◉ su diversi media, come stampa su carta, video, audio, medium per disabili, ecc.
 - ◉ separare la descrizione dello stile dal contenuto e dalla sua struttura
- ◉ <http://w3c.org/Style/>
 - ◉ Numerose specifiche: CSS1, CSS2, XPath, XSLT, XSL-FO
 - ◉ Due linguaggi: CSS & XSL

Perché due linguaggi?

- CSS
 - può essere usato con HTML e XML
 - ma ha una sua sintassi propria, e non è generale a sufficienza per essere un linguaggio di trasformazione
- XSL (combinazione di XSLT & XSL-FO)
 - è un linguaggio di trasformazione
 - p.e., per trasformare una pagina XML in HTML/CSS
 - con sintassi XML
 - ma può essere usato solo con XML, e non con HTML
- In effetti, condividono lo stesso modello di formattazione
- e possono essere usati insieme

Dynamic HTML

- Pagine HTML con contenuto dinamico
- Tre tecnologie componenti
 - HTML / XHTML
 - CSS
 - JavaScript / ECMAScript
- che condividono il DOM
 - Document Object Model
 - che descrive la struttura generale concettuale del documento DHTML
 - a cui fanno riferimento i browser
 - che però poi hanno specifiche di DOM più dettagliate
 - che dobbiamo conoscere e evitare

Specifiche CSS

- CSS1, CSS2, e successive
- Noi ci limitiamo a lavorare su CSS1
 - oltre ai tutorial, guardate bene CSS1
 - <http://www.w3c.org/TR/REC-CSS1>
 - perché le domande all'esame saranno basate su quello
 - quindi vi conviene imparare a cercare lì velocemente le informazioni che vi servono

Perché “cascading”?

- Perché ci possono essere tante specifiche di stile diverse per uno stesso documento
 - in cascata
 - per ragioni di
 - modularità
 - equilibrio autore / lettore
- Una cosa da imparare è l'ordine di priorità della “cascata”

Incorporare CSS in (X)HTML

- 1 Referenziando un documento CSS esterno (dentro <head>)

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

- 2 Specificando l'elemento <style> (dentro <head>)

```
<style type="text/css"><!--  
  @import url(style.css)  
  a.smalllink, a.medlink, a.biglink {  
    font-family: Tahoma, Verdana, Myriad Web, Syntax, sans-serif;  
    font-weight: bold; text-decoration: none; white-space: nowrap; }  
  a.smalllink { font-size: 8pt; }  
  a.medlink { font-size: 9pt; }  
  a.biglink { font-size: 10pt; }  
--></style>
```

- 3 Specificando l'attributo style dentro un tag

```
<p style="color: green">Questo testo sia verde</p>
```

Dichiarazioni CSS

- Dichiarazione

```
h1 { font-size: 14pt; }
```

- Gruppi

```
h1, h2, h3 { font-family: helvetica; }  
h1 { font-weight: bold;  
      font-style: normal; }
```

- Ereditarietà

- tutte le proprietà non specificate sono ereditate dal "contenitore più esterno"

```
<h1>Se il tag di enfasi non specifica il font <em>questo</em>  
rimane Helvetica</h1>
```

Un foglio di stile CSS

- È un file di testo
 - si crea con le solite modalità
 - nuovo file in text editor o word processor
 - poi si salva solo testo
- con estensione .css
- Contiene solo
 - dichiarazioni CSS
 - commenti
- Nessun prologo nè struttura

Classi come Selettori

- Classi
 - nomi definiti dall'utente per raggruppare elementi
 - tramite l'attributo class
- Stile per classi in notazione puntata
 - ```
.smalllink { font-size: 8pt; }
```

    - classe "generica"
  - ```
a.smalllink { color: blue; }
```

 - classe "regolare"
 - fanno sì che
 - ```
<p class="smalllink">Testo piccolo</p>
```
    - sia 8 punti, mentre
      - ```
<a class="smalllink">Link piccolo</a>
```
 - sia 8 punti e blu

ID come Selettori

- Anche l'attributo ID può essere specificato per qualunque elemento
 - e usato come selettore di stile
 - usando # invece del punto
- ```
#esempioID { font-size: 8pt; }
```
- La differenza, più che sintattica o semantica, è concettuale
    - le classi raggruppano elementi omogenei
    - l'ID serve a definire caratterizzazioni individuali
      - l'ID è unico in una pagina XHTML
      - utile in pagine generate dinamicamente per cambiare uno stile

# Selettori contestuali

- Si può sfruttare l'ereditarietà per definire stili innestati
  - tipo "l'enfasi dentro un titolo di livello 1 è verde"

```
h1 em { color: green; }
```
  - modello "a stack", senza limiti (a parte il buon senso)
  - che si adatta al modello di ereditarietà
- Si mischia senza problemi con classi e ID

# Commenti

```
/* Questo è un commento */
```

# Pseudo-classi

- Pseudo-classi di ancora

```
a.link { color: red; }
a.visited { color: blue; }
a.active { color: green; }
```

- specificano, rispettivamente, il colore del link quando viene visualizzato, dopo che è stato visitato, e quando il puntatore gli è sopra
- Ci sono anche pseudo-elementi come first-line o first-letter
  - guardateci voi :)

# Cascading

- Più specifiche possono insistere sulla stessa proprietà
- Algoritmo di risoluzione
  - trovare tutte le dichiarazioni, e i valori ereditati e di default
  - ordinare le dichiarazioni per importanza

```
h1 { color: green ! important; }
```
  - ordinare per origine: autore > lettore > browser
  - ordinare per specificità: più specifico > meno specifico
  - ordinare per ordine d'arrivo: l'ultimo vince

# Modello di formattazione

- Due tipi di elementi
  - in-line
    - non hanno "a capo" prima e dopo, sono per default la maggioranza dei tag come <p>, <em>, <b>, ...
  - block
    - è come se stessero su una riga per conto loro
    - per default, i titoli di vario livello, e gli elementi delle liste
- La proprietà del DOM che definisce ciò è "display"
  - quindi, la posso cambiare con una dichiarazione CSS
  - valori: "inline", "block", "none"

# Cosa dovete imparare?

- Come minimo
  - Sintassi CSS, e combinazione con XHTML
  - Le basi di CSS: fonti, testo, liste, colori
  - Classi, ereditarietà e cascading
  - Tabelle con CSS
  - Formattare pagine Web con CSS
- Ma la sintassi è tanto semplice in struttura quanto complessa in quantità e dettaglio
  - meglio imparare l'accesso rapido alle fonti