

Sistemi Distribuiti

Cascading Style Sheets (CSS): Concetti Fondamentali

Sistemi Distribuiti a.a. 2002/03

Questi lucidi sono basati sul materiale didattico preparato
dal prof. G. Mecca dell'Università della Basilicata
(<http://www.difa.unibas.it/users/gmecca/corsi/sviluppoWeb>)

CSS >> Sommario

Concetti Fondamentali

- Introduzione
 - ⇒ storia di CSS
 - ⇒ uno studio di caso
 - ⇒ la filosofia di CSS
- Esempio di riferim.
 - ⇒ modulo "LINK" di XHTML
- Sintassi
 - ⇒ regole
- ⇒ classificazione delle proprietà
- ⇒ selettori
- Semantica
 - ⇒ ereditarietà e cascata
- "Box Model"
- Lunghezze
- Tipi di Dispositivo
- Compatibilità

Sistemi Distribuiti a.a. 2002/03

3

CSS >> Concetti Fondamentali >> Introduzione

Introduzione

- Cascading Style Sheets (CSS)
 - ⇒ fogli di stile in cascata
- Tecnologia per la presentazione
 - ⇒ associa elementi di grafica ad un documento
es: tipo di font, colori, spaziature ecc.
- Si applica a linguaggi di marcatura
 - ⇒ HTML, XML, SGML ...
- Vari standard del W3C

Sistemi Distribuiti a.a. 2002/03

4

Storia di CSS

- Idea di separare struttura e presentazione
 - ⇒ Tim Berners Lee, NeXT browser, 1990
 - ⇒ documento HTML
 - ⇒ fogli di stile con sintassi proprietaria
- L'approccio di Mosaic (1993)
 - ⇒ niente fogli di stile
 - ⇒ presentazione standard da parte del browser
 - ⇒ semplici funzionalità per la grafica (colori)

Storia di CSS

- La guerra dei browser
 - ⇒ domanda del mercato
 - ⇒ elementi non standard solo per la grafica
es: b, font, multicol, spacer, blink, marquee...
- Utilizzo abituale di HTML
 - ⇒ utilizzo improprio delle tabelle
 - ⇒ trucchi per la grafica es: `null.gif`
 - ⇒ scarso rispetto per gli standard

Gestione della presentazione: manutenzione e scalabilità

- Un sito con un numero elevato di pagine
- In ogni pagina elementi (intestazioni, titoli, barre di navigazione, ecc.) con precise caratteristiche grafiche (dimensione e colore dei font, bgcolor, etc.)

Manutenzione

- Supponiamo di voler "rinfrescare la grafica" del sito
- Dobbiamo modificare tutti i file
 - ⇒ Costi elevati
 - ⇒ Altissima probabilità di introdurre errori

Scalabilità

- Il sito di una grossa istituzione (ad es. una università)
 - ⇒ Organizzazione in sezioni
 - ⇒ ogni sezione gestita autonomamente da un dipartimento/divisione
- Come conciliare
 - ⇒ Uniformità della presentazione
 - ⇒ Autonomia locale di sviluppo
 - ⇒ Forme locali di personalizzazione

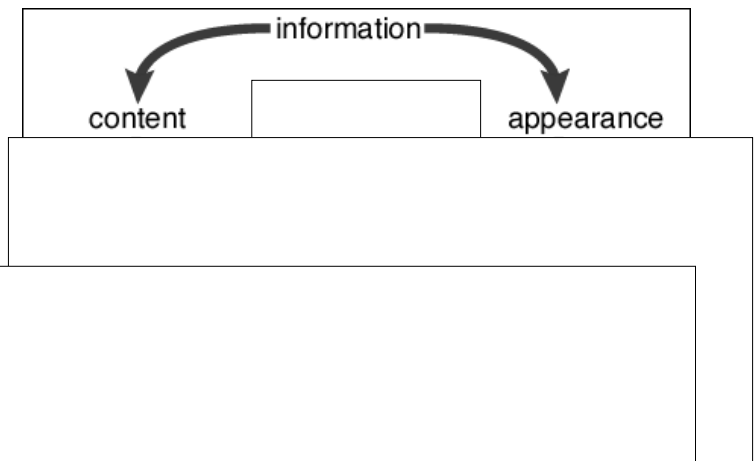
Scalabilità e Manutenzione

- Centralizzo la definizione degli aspetti di presentazione
 - ⇒ Evoluzione: agisco solo sulla definizione della presentazione
 - ⇒ Autonomia: uso la definizione della presentazione comune
 - ⇒ Località: “estendo” localmente la presentazione comune

La Filosofia di CSS

- Idee principali
 - ⇒ separare contenuto e grafica (foglio di stile)
 - ⇒ consentire la “combinazione” di più stili
 - ⇒ consentire di associare stili diversi a dispositivi (“media”) diversi
- E inoltre
 - ⇒ consentire all’utente di scegliere lo stile (combinare stile dell’utente e dell’autore)
 - ⇒ rendere visibile comunque il contenuto (anche ai browser che non supportano CSS)

La Filosofia di CSS



Storia di CSS

- CSS Livello 1
 - ⇒ dicembre 1996
 - ⇒ principali caratteristiche di formattazione (font, colori, spaziature, dimensioni ecc.)
 - ⇒ primi browser: IE3, NN4
- CSS Livello 2
 - ⇒ maggio 1998
 - ⇒ estende CSS1 in varie direzioni (supporto a dispositivi diversi, posizionamento assoluto)

Storia di CSS

- CSS Mobile Profile
 - ⇒ “candidate recommendation”, ottobre 2001
 - ⇒ insieme minimo di funzionalità CSS che un browser (mobile e non) dovrebbe supportare
 - ⇒ essenzialmente CSS livello 1
 - ⇒ alcune funzionalità di CSS

Il Modulo LINK di XHTML

- Visualizzazione standard
 - ⇒ foglio di stile del browser
- Per associare un foglio di stile
 - ⇒ modulo LINK di XHTML
 - ⇒ riferimento al foglio di stile nella testa

elemento	attributo	modello di contenuto
link	rel, title, href, type ...	EMPTY

Il Modulo LINK di XHTML

- Esempio
 - ⇒ `<link rel="stylesheet" type="text/css" href="primo.css" />`
- E' possibile specificare più di uno stile
 - ⇒ `<link rel="alternate stylesheet" type="text/css" href="stili/secondo.css" title="secondo" />`
 - ⇒ `<link rel="alternate stylesheet" type="text/css" href="http://www.w3.org/css/examples.css" title="stile del consorzio" />`

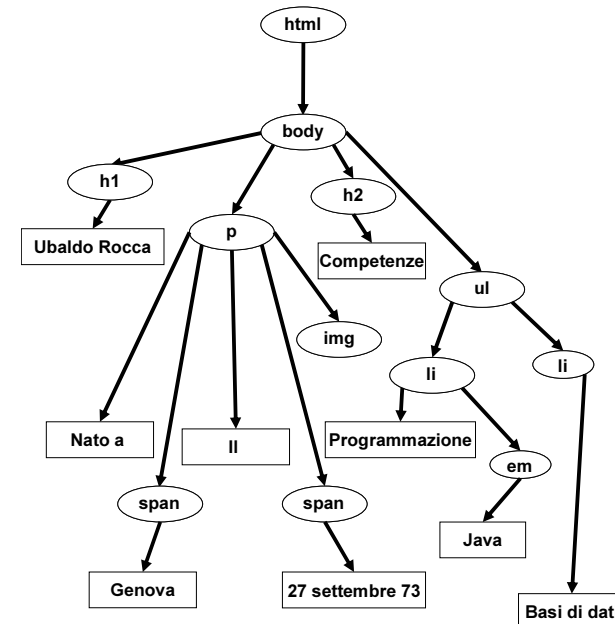
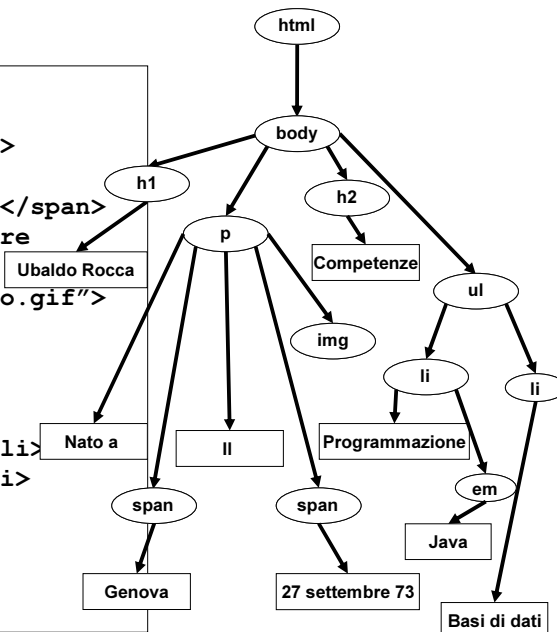
Il Modulo LINK di XHTML: Esempio

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "DTD-xhtmlbasic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it">
<head>
<title>Pagina di Esempio</title>
<link rel="stylesheet" type="text/css" href="primo.css" />
</head>
<body>
...
</html>
```

CSS: idea generale

- I documenti HTML corrispondono ad alberi
- In fase di visualizzazione, la nidificazione dei nodi corrisponde ad una nidificazione di riquadri ("box")
- L'idea di CSS è di associare ai vari riquadri un insieme di regole di visualizzazione (ad es. colore, font, larghezza, etc.)

```
<html>
<body>
<h1>Ubaldo Rocca</h1>
<p>
  Nato a <span>Genova</span>
  il <span>27 settembre
  1973</span>
  ... 
</p>
<h2>Competenze</h2>
<ul>
<li>Programmazione
  <em>Java</em></li>
<li>Basi di Dati</li>
</ul>
</body>
</html>
```



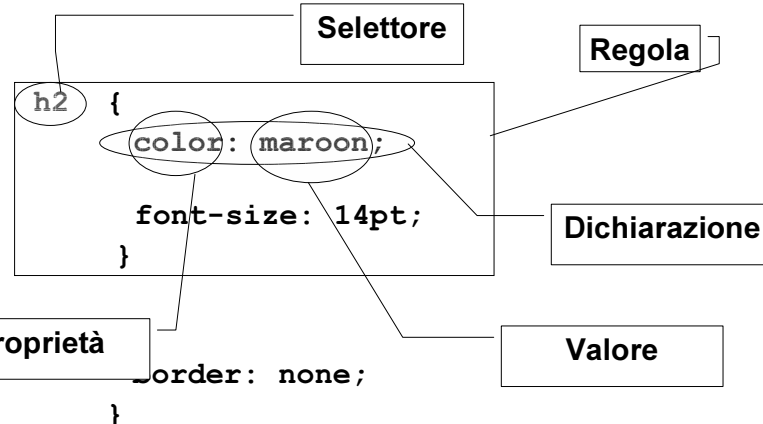
Idea generale

- Per specificare come formattare ogni box abbiamo bisogno di:
 - ⇒ un meccanismo che permetta di selezionare i vari box
 - ⇒ un insieme di dichiarazioni di formattazione da associare ad ogni selettore di box
- Per rendere efficace il meccanismo
 - ⇒ (alcune) proprietà devono poter essere “ereditate” dal box padre (ad esempio il font)

Sintassi

- Foglio di stile: lista di regole
 - ⇒ <selettore> { <lista di dichiarazioni> }
- Regola
 - ⇒ <selettore> { <lista di dichiarazioni> }
- Dichiarazione
 - ⇒ <proprietà>: <valore>;

Foglio di stile: lista di regole



Classificazione delle Proprietà (CSS1)

- Font
 - ⇒ proprietà dei font
 - ⇒ es: font-family
- Color
 - ⇒ caratteristiche dello sfondo
 - ⇒ es: color
- Text
 - ⇒ formato del testo
 - ⇒ es: text-align
- Box
 - ⇒ stabilisce margini e spaziatura
 - ⇒ es: margin-top
- Classification
 - ⇒ tipo di visualizzazione
 - ⇒ formato delle liste
 - ⇒ es: list-style-image
- Totale: 50 proprietà
 - ⇒ 122 in CSS2

Selettori

- Varie forme
- Nome di elemento
 - ⇒ `es: img { border: none; }`
 - ⇒ tutti gli elementi con quel nome
- Altre forme di selettore
 - ⇒ (sotto)-classe di elementi con lo stesso nome
 - ⇒ elementi singoli

Selettori

- Classe di elementi
 - ⇒ attributo "class" degli elementi HTML
 - ⇒ selettore `<nome>.<classe>`
- Esempio: titolo e coda in bianco su rosso
 - ⇒ `<div class="barra"> ... </div>`
 - ⇒ `div.barra { background-color: red; color: white; text-align: center; }`

Selettori

- Elementi singoli
 - ⇒ attributo "id" degli elementi HTML
 - ⇒ selettore `<nome>#<id>`
- Esempio: indirizzo in corsivo
 - ⇒ `<p id="indirizzo"> ... </p>`
 - ⇒ `p#indirizzo { text-style: italic; font-size: 12pt; }`
- Esempio: nome del dipartimento in rosso
 - ⇒ ` ... `
 - ⇒ `span#dip { color: red; }`

Semantica

- Semantica delle regole
 - ⇒ caratteristiche di visualizzazione
 - ⇒ elementi che corrispondono al selettore
- Due meccanismi fondamentali
 - ⇒ **ereditarietà**: un elemento può ereditare caratteristiche dai suoi antenati (DOM)
 - ⇒ **cascata** ("cascade"): la stessa proprietà di un elemento può essere definita più volte

Ereditarietà

- Documento HTML/XHTML
 - ⇒ InfoSet o, correntemente “albero DOM”
- Ereditarietà
 - ⇒ alcune proprietà sono ereditate dai predecessori (es: font), altre no (es: sfondo);
 - ⇒ vedi standard; in CSS2: proprietà “inherit”
 - ⇒ es:


```
body {font-family: Arial;
background-image: url("icons/quadretti.jpg");}
```

Cascata

- Proprietà ridefinite: conflitti
 - ⇒ vari fogli di stile (standard, dell'autore, dell'utente)
 - ⇒ varie definizioni nello stesso foglio

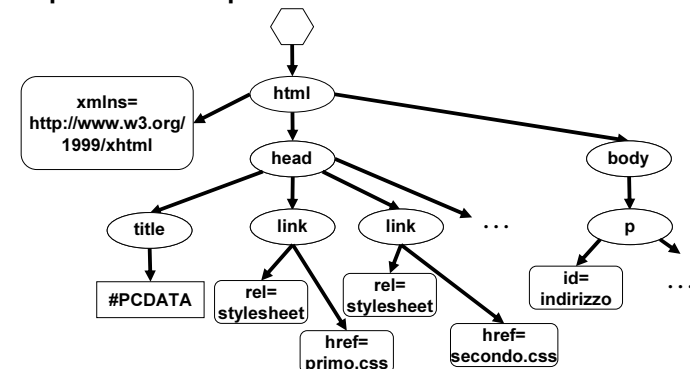

```
ES: p { font-size: 9pt; }
```
- Cascata
 - ⇒ le proprietà hanno precedenze diverse
 - ⇒ si applicano “in cascata”

Cascata

- Algoritmo di risoluzione dei conflitti
 - ⇒ trova tutte le proprietà per un elemento
 - ⇒ se nessuna, ereditarietà oppure standard
 - ⇒ se più di una, pesa per origine (prima autore, poi utente, poi standard)
 - ⇒ se più di una con lo stesso peso, pesa per specificità (prima id, poi classe, poi nome)
 - ⇒ se più di una con lo stesso peso, pesa per ordine di apparizione (prima le ultime)

Esempio

- Proprietà di <p id="indirizzo">



Esempio

- Proprietà di `<p id="indirizzo">`
 - ⇒ proprietà dello stile standard
es: color: black;
 - ⇒ proprietà ereditate da `<body>` es: font-family: Arial;
 - ⇒ proprietà con selettore p; es: font-size: 9pt;
 - ⇒ proprietà con selettore `p#indirizzo`
font-style: italic; font-size: 12pt;
 - ⇒ conflitto sulla dimensione del font
prevale la proprietà con selettore più specifico

ATTENZIONE:
importanza della
correttezza
sintattica

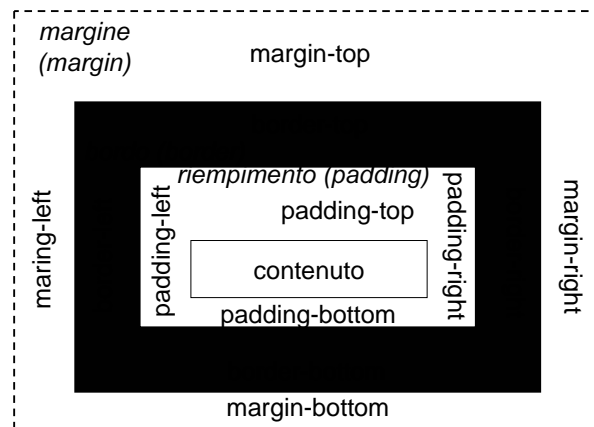
Modello dei Riquadri ("Box Model")

- Classificazione degli elementi HTML
 - ⇒ "heading", "block", "inline", "list"
- Elementi che producono blocchi
 - ⇒ heading, block, list: producono un'interruzione di linea
- Elementi in linea
 - ⇒ inline: non producono interruzioni di linea
- Ad ogni elemento corrisponde un riquadro

Modello dei Riquadri ("Box Model")

- Flusso di visualizzazione
 - ⇒ disposizione dei riquadri nella finestra del br.
 - ⇒ riquadro contenuto nel riquadro del padre
- Riquadro: articolato in vari spazi
 - ⇒ contenuto ("content")
 - ⇒ riempimento ("padding")
 - ⇒ bordo ("border")
 - ⇒ margine ("margin")

Modello dei Riquadri ("Box Model")



Modello dei Riquadri (“Box Model”)

○ Proprietà relative al riquadro

- ⇒ margin-top, margin-bottom, margin-left, margin-right (lunghezze)
- ⇒ padding-top, padding-bottom, padding-left, padding-right (lunghezze)
- ⇒ border-top-width, border-bottom-width, border-left-width, border-right-width (lunghezze)
- ⇒ border-color, border-style
- ⇒ width, height (solo per i “block”)

Modello dei Riquadri: Esempi

```
body {padding-left: 30pt;      h2 {margin-top: 5pt;
padding-right: 30pt;         margin-bottom: 2pt; }
padding-top: 10pt;
padding-bottom: 10pt;}

a {color: white;              p {margin-top: 0pt;
background-color: blue;      margin-bottom: 0pt; }
padding-left: 5pt;
padding-right: 5pt; }

div.barra {
padding-top: 10pt;
padding-bottom: 10pt;
border-color: yellow;
border-style: solid;
border-width: thick; }

a.linkImg {
background-color:
transparent; }
```

Alterazione del Flusso

○ Proprietà “float” e “clear”

○ Float

- ⇒ sposta a destra o a sinistra i riquadri
- ⇒ fuori dal flusso ordinario (“fluttuanti”)
- ⇒ gli altri riquadri scorrono attorno
- ⇒ l’elemento viene considerato di tipo “block”
- ⇒ utilizzo tipico: immagini

Alterazione del Flusso

○ Esempio

```
⇒ <img class="aDestra" ... />
⇒ img.aDestra { float: right;
margin-bottom: 10pt;
margin-top: 10pt; }
```

Alterazione del Flusso

○ Clear

- ⇒ impedisce lo scorrimento
- ⇒ specifica se un elemento può avere elementi flottanti alla sua sn, alla sua ds, or per nulla
 - ⇒ clear: left;
l'elemento non permette floating obj a sn
 - ⇒ clear: right;
 - ⇒ clear: both;
- ⇒ `div.barra {clear: both; width: 400pt; }`

○ Div e clear possono sostituire le tabelle per creare layout di pagina

Lunghezze

○ Vari modi per esprimere lunghezze

- ⇒ absolute; unità: in, cm, mm, pt (1 pt=1/72 in)
- ⇒ relative; unità: em, px (pixel)
- ⇒ percentuali; es: 90%
- ⇒ parole chiave; es: small, x-small

○ In generale

- ⇒ evitare di utilizzare unità assolute
- ⇒ portabilità tra i dispositivi e flessibilità

Lunghezze Relative

○ em

- ⇒ lung. relative alla dimensione del font
- ⇒ font-size: relativo alla dim. del font del padre

○ px

- ⇒ lung. relative alla risoluzione dello schermo

○ Effetto simile alle percentuali

- ⇒ ma uso più complesso

Lunghezze Relative: Esempi

```
body {font-family: Arial;
background-image:
url("icons/quadr.jpg");
font-size: 1em;
padding-left: 2em;
padding-right: 2em;
padding-top: 1em;
padding-bottom: 1em;}

div.barra {color: white;
background-color: maroon;
text-align: center;
padding-top: 0.5em;
padding-bottom: 0.5em;
border-color: navy;
border-style: solid;
border-width: 0.2em;
clear: both; }

h2 {color: maroon;
font-size: 1.5em;
margin-top: 0.5em;
margin-bottom: 0.5em; }

p {font-size: 0.9em;
margin-top: 0;
margin-bottom: 0; }

a { background-color: blue;
color: white;
padding-left: 0.5em;
padding-right: 0.5em; }

p#indirizzo {
font-style: italic;
font-size: 1.1em;} sesto.css
...
```

Tipi di Dispositivo (“Media Types”)

- CSS2 introduce il concetto di “media”
 - ⇒ è possibile specificare il dispositivo di rif.to
 - ⇒ il browser può selezionare lo stile giusto
- Tipi di Media
 - ⇒ screen, handheld, tty, tv
 - ⇒ print, projection
 - ⇒ braille, embossed, aural
 - ⇒ all

Tipi di Dispositivo (“Media Types”)

- Specifica del tipo di dispositivo
 - ⇒ attributo “media” dell’elemento link
 - ⇒ `<link rel="stylesheet" type="text/css" href="primo.css" media="screen" />`
 - ⇒ `<link rel="stylesheet" type="text/css" href="secondo.css" media="all" />`
- Regole “chiocciola” (“at-rules”)
 - ⇒ @import
 - ⇒ @media

Tipi di Dispositivo (“Media Types”)

- @import
 - ⇒ include le regole di un foglio in un altro
 - es: in “secondo.css”
 - `@import url("primo.css");`
 - ⇒ deve essere la prima dichiarazione del foglio
 - ⇒ è possibile specificare il tipo di media
 - es: `@import url("primo.css") handheld;`
- @media
 - ⇒ specifica il dispos. per cui vale un blocco di regole
 - ⇒ es: `@media print { body {font-size: 10pt;}... }`

Compatibilità

- Internet Explorer 5
 - ⇒ primo con compatibilità del 99% con CSS1
- Versioni 6
 - ⇒ ottima compatibilità con CSS1
 - ⇒ alcune proprietà di CSS2
- Principio di visibilità
 - ⇒ il contenuto deve essere visibile anche in assenza del foglio di stile (stile standard del browser)

Compatibilità

- Porre attenzione al supporto dei browser
- Per essere aggiornati:
<http://www.webreview.com/style/css1/charts/mastergrid.shtml>