



An Introduction to CSS

Prof. Ing. Andrea Omicini
II Facoltà di Ingegneria, Cesena
Alma Mater Studiorum, Università di Bologna
andrea.omicini@unibo.it

Web Style Sheets

- Style sheets for the Web
- Aims
 - describing how elements in a document must be presented
 - on different media types, as paper print, video, audio, medium for people with disabilities, etc.
 - separating style's description from content and its structure
- See <http://www.w3c.org/Style/>
- Many specifications: CSS1, CSS2, XPath, XSLT, XSL-FO
- Two languages: CSS & XSL

2

Why two languages?

- CSS
 - can be used with HTML and XML
 - but it has its own syntax, and it's not general enough to be a transformational language
- XSL (union of XSLT / XSL-FO / XPath)
 - it's a transformational language
 - e.g., it can be used to transform an XML page in HTML/CSS
 - featuring an XML syntax
 - but it can be used with XML only, not with HTML
- Indeed, they share the same "formatting model"...
- ...and they can be used together

3

Dynamic HTML

- HTML pages with dynamic content
- composed using three technologies
 - HTML / XHTML
 - CSS
 - JavaScript / ECMAScript
- sharing the DOM
 - Document Object Model
 - which describes the conceptual general structure of a DHTML document
 - which is referenced by browsers
 - which feature their own detailed DOM specifications
 - which we have to know and avoid

4

AJAX

- Asynchronous JavaScript And XML
 - goal: improve interaction between browsers & servers
- composed using three technologies
 - a combination of:
 - XHTML / HTML & CSS
 - JavaScript for DOM manipulation
 - XMLHttpRequest object
 - to exchange data asynchronously with server
 - usually, XML for data transfer
- example: changing a portion of a web page according to some user interaction without reloading a whole page

5

CSS Specifications

- CSS1, CSS2, and above
- We focus our work on CSS1
 - study CSS1 besides tutorials
 - See <http://www.w3c.org/TR/REC-CSS1>
 - because questions in the exam will be based on that specification
 - so you'll benefit from learning how to quickly search needed information in that document

6

Why "cascading"?

- 1 Because there can be many different styles specified for the same document
 - 2 in a cascading flow
 - 3 for different reasons
 - 4 modularity
 - 5 a balance between author and reader
- 6 A thing to learn is the priority order of the "cascade"

7

How to embody CSS in (X)HTML

- 1 Referencing an external CSS document (within <head>)
`<link href="style.css" rel="stylesheet" type="text/css" media="screen" />`
- 2 Specifying the <style> element (within <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 Specifying the style attribute within a tag
`<p style="color: green">Let this text be green</p>`

8

CSS Declarations

- 1 Declaration
`h1 { font-size: 14pt; }`
- 2 Groups
`h1, h2, h3 { font-family: helvetica; }
h1 { font-weight: bold;
font-style: normal; }`
- 3 Inheritance
 - 4 all non-specified properties for an element are inherited by its parent element`<h1>If the emphasis tag does not specify its font this is displayed as Helvetica</h1>`

9

A CSS stylesheet

- 1 It's a text file
 - 2 you can create it in the usual ways
 - 3 a new file in a text editor or word processor
 - 4 then you save it as plain text
- 3 with .css extension
- 4 It only contains
 - 5 CSS declarations
 - 6 comments
- 7 Neither prologue nor structure

10

Classes as selectors

- 1 Classes
 - 2 user defined names to group elements
 - 3 by means of the class attribute
- 2 Dot notation for class styles
`.smalllink { font-size: 8pt; }`
 - 3 "generic" class
`a.smalllink { color: blue; }`
 - 4 "regular" class
 - 5 they make
 - 6 <p class="smalllink">Tiny text</p>
 - 7 to be 8 points, while
 - 8 Tiny link
 - 9 to be 8 points and blue

11

ID as selectors

- 1 Also the id attribute can be specified for every element
 - 2 and used as a style selector
 - 3 using # instead of a dot
 - 4 #exampleID { font-size: 8pt; }
- 2 The difference is conceptual rather than syntactic or semantic
 - 3 classes group homogeneous elements
 - 4 ID is used to define individual characterizations
 - 5 any ID is unique in an XHTML page
 - 6 useful in dynamically generated pages to change a style

12

Contextual selectors

- Inheritance can be exploited to define nested styles
 - e.g. "emphasis within a level 1 header is green"

```
h1 em { color: green; }
```
 - "stack" model, without limits (just use common sense)
 - which fits the inheritance model
- It can be mixed with classes and IDs without any problem

13

Comments

```
/* This is a comment */
```

14

Pseudo-classes

- Anchor pseudo-classes

```
a:link { color: red; }
a:visited { color: blue; }
a:active { color: green; }
```
- specify the link's color, respectively: when the link is visualized; after the link has been visited; and when the pointer hovers on the link
- There are also pseudo-elements as `first-line` and `first-letter`
 - have a look by yourself :)
- Pseudo-classes can be combined with CSS classes

15

Cascading

- Many declarations can be applied to the same property
- Resolution algorithm
 - find all the declarations and their default inheritance values
 - order declaration by importance

```
h1 { color: green ! important; }
```
 - order by source: author > reader > browser
 - order by specificity: more specific > less specific
 - order by appearance: the last one wins

16

Formatting model

- Two kinds of elements
 - in-line
 - they do not have a "newline" after and before, it's the default for most tags as ``, ``, ``, ...
 - block
 - it's as if they are displayed on a line of their own
 - it's the default for headers of all levels, and list elements
- The DOM property defining this behaviour is called `display`
 - so, it can be changed using a CSS declaration
 - values: `inline`, `block`, `none`

17

What should we learn from our lab activity?

- As a minimum
 - CSS syntax, and interoperation with XHTML
 - CSS fundamentals: fonts, text, lists, colors
 - Classes, inheritance, cascading
 - How to manage tables with CSS
 - In general, how to format web pages using CSS
- Syntax is as simple in structure as complex for quantity and details
 - it's better to learn using quick access to knowledge sources

18