

Cascading Style Sheets (CSS)

Prof. Cesare Pautasso

<http://www.pautasso.info>

cesare.pautasso@unisi.ch

19.9.2007

Fall Semester 2007
Software Atelier III – Web Development Lab
©2007 Cesare Pautasso

1

Contents

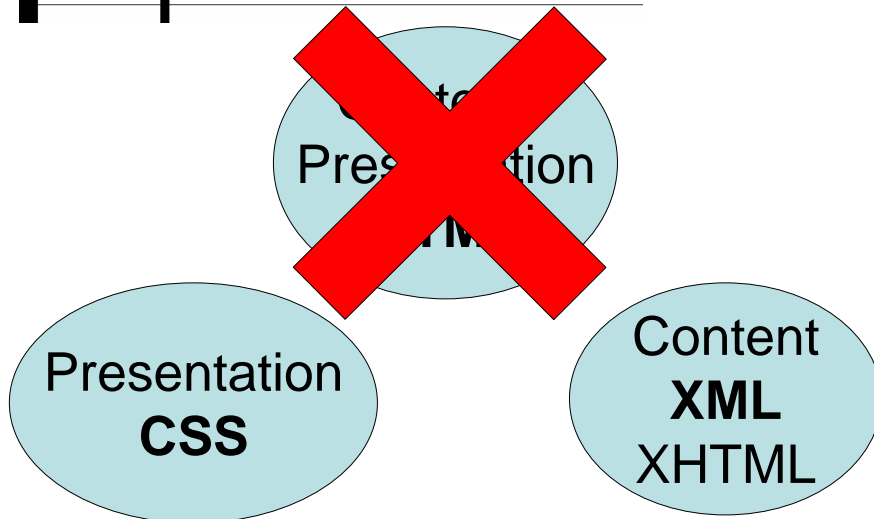
- Presentation vs. Contents
- XHTML (Content)
- Cascading Style Sheets (Presentation)
 - How to style elements
 - Formatting Rules
 - Selectors
 - Why Cascading?
 - Page Layout
- Cross-Browser Compatibility

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

2

Presentation/Content



19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

3

Presentation/Content

- Advantages of Presentation/Content separation:
 - Reuse style sheets
 - one style for the entire website (easier to maintain)
 - spare bandwidth (css can be cached)
 - Different styles for the same content
 - Multi-channel delivery (Web browser, printer, mobile phone, screen reader)
 - User personalization (choose themes)

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

4

XHTML

- XHTML stands for **EX**tensible **HyperText Markup Language** (W3C Standard, 2000)
- Combines:
 - XML (Strict Syntax)
 - HTML (existing tags for Web pages)
- It will gradually replace HTML (legacy)
- Works well together with CSS

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

5

X(HT)ML Rules

1. All element tags and attribute names written in **lowercase** (XML is case sensitive)
2. Always close every opened element
`<tag>...</tag>` or `<tag/>`
3. All elements must be properly nested
`<p></p>`
4. Only one root element per document
`<html><body></body></html>`
5. Attribute values must be quoted
`...`
6. `id` Attribute replaces the name attribute
`<div id="navigation">...</div>`

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

6

XHTML Structure

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
strict.dtd">  
  
<html>  
  <head>  
    <title>My Homepage</title> </head>  
  <body>  
    <h1>My Title</h1>  
    <p>Some paragraph</p>  
    <!-- Comment -->  
  </body>  
</html>
```

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

7

XHTML Types

- **Strict** - this is the one to use to get the most out of CSS

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```
- **Transitional** - more backwards compatible with HTML4, some browsers will not properly render the CSS

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```
- **Frameset** - use only if you REALLY need to use frames in your page

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

8

- <http://csszengarden.com/>
- <http://www.w3.org/StyleSheets/Core/preview>

There are three ways of applying CSS to HTML:

1. Straight into the HTML tags using the **STYLE attribute**.

```
<p style="color: red">text</p>
```

HTML should be presentation free, so in-line styles should be avoided wherever possible.

2. With internal styles that are used for the whole page. They are placed into a **STYLE tag** in the page header.

Like the inline styles, this should be limited to special pages that, for some reasons, should look different than the others.

Applying CSS

3. With external styles, stored in a separate CSS file. This file called 'style.css' will contain a set of declarations like:

```
p { color: red; }
```

Use the `@import` directive or the `<link>` element to use the CSS from a web page:

```
<html><head>
<title>My first CSS-based page</title>
<style type="text/css">@import
url(style.css)</style>
<link type="text/css" rel="stylesheet"
href="style.css"/>
</head>
```

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

11

CSS Rules

CSS is a declarative language. It uses **rules** to specify how parts of the XHTML document should be rendered.

Each CSS rule has

- A *selector*, targeting specific XHTML elements
- One or more pairs of property name and value, to control their formatting

```
selector
{ property: value; }
```

```
body {
font-size: 0.8em;
color: navy;
}
```

Apply this rule to the **body** tag, setting the **font-size** and **color** properties to respectively **0.8em** and **navy**.

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

12

Formatting Values

Depending on which property, rules use the following data types:

- Strings
- Numbers
- Enumerations
- Lengths
- Colors
- URLs

```
h1 {
  font-family: "Arial";
  font-size: 0.8em;
  font-style: italic;
  color: navy;
  background-color: #001122;
  background-image: url(/images/h1.png);
}
```

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

13

Colors

Colors represented with:

- RGB values such as `rgb(255,0,0)`
- Hexadecimal RGB values such as `#fedc10` or `#e2c = #ee22cc`
- enumeration (`red`, `black`, `white`, etc.)
- `transparent`

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

14

Lengths and Sizes

Lengths can be expressed in several units of measures

- **em**, the height/width of a M character
- **px**, for pixels
- **%**, for percentages of the parent's value
- **0** (Zero) does not need a unit.

A web page should be flexible, so pixel widths should be used sparingly

- Pixels recommended for border thickness or other spacing
- Very bad for font sizes (some browsers will not zoom)
- Percentages can be dangerous when used with nested tags

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

15

Main Properties

- background (-color, -image, -position)
- border (-width, -style, -color) x (-top, -right, -bottom, -left)
- font (-family, -size, -weight)
- text (color, direction, letter-spacing, text-align, text-decoration, text-transform, word-spacing, ...)
- tables (border-collapse, caption-side, empty-cells)
- list (-style, -style-image, -style-type)
- margin (-top, -right, -bottom, -left)
- padding (-top, -right, -bottom, -left)
- layout (display, clear, float, position, visibility, cursor)
- positioning (left, right, bottom, top, clip, overflow, vertical-align, z-index)
- dimension (height, width, max-height, max-width, min-height, min-width)

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

16

http://www.w3schools.com/css/css_reference.asp#border

Selectors

- Control which XHTML elements are affected by a certain rule by referring to:
 - Tags **h1**
 - Classes **.navigation**
 - IDs **#home**
 - Wildcards *****
 - Pseudo-Classes **:hover**
- Selectors can be combined, grouped, and nested to make them more selective

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

17

Universal Selector

```
* {  
  color: #000;  
  margin: 0;  
  padding: 0;  
}
```

- Use the * selector to setup default properties of ALL elements of your page that do not have a more specific style
- Browser defaults may differ, so this helps you to start with known formatting defaults

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

18

Tag Selectors

```
body {  
  color: #000;  
  background: #fff;  
  font-family:  
  verdana, arial, san  
  s-serif;  
  font-weight:  
  normal;  
}
```

- This applies the formatting to elements of a certain type (e.g., the <body> of a page)
- Also all children elements within the <body> will inherit the same format (unless there is an overriding rule)

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

19

Class Selectors

```
.navigation {  
  text-decoration:  
  underline;  
}
```

```
<p class="navigation">  
Home - Introduction  
</p>
```

- The same rule is applied to all document elements of a particular class (independent of their tag)
- More than one element can share the same class

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

20

ID Selectors

```
#home {
  margin-top: 2px;
  border-bottom: 1px
  solid;
}
```

```
<p id="home">
Welcome to my Homepage
</p>
```

- The rule applies only to the document element with the given id attribute
- In XHTML Element IDs must have **unique** values within the same page

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

21

Attribute Selectors

```
a[href^="http:"] {
  background: url(external.png)
  no-repeat right top;
  padding-right: 10px;
  border-bottom: 2px solid;
}
a[href^="http://www.mysite.com"] {
  background-image: none;
  padding-right: 0;
  border-bottom: 1px dotted;
}
<a href="http://www.google.com">
External Link
</p>
```

- The rule applies only to the document element with the matching attribute value
- Matching operators:
 - attribute = value (equals)
 - attribute ~= value (contains)
 - attribute ^= value (starts with)
 - attribute \$= value (ends with)
- The example shows how to style external links differently
- Warning: does not work in IE6.

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

22

Nested Selectors

Selectors separated by 'space' are used to target nested tags.

ul li

defines a rule for elements `` within a ``

#top a

defines a rule for element `<a>` within the element with `id="top"`

#top .navbar a

defines a rule for element `<a>` within an item of `class="navbar"` contained within an element with `id="top"`

a.navbar#top

defines a rule for element `<a>` of `class="navbar"` of `id="top"`

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

23

Combined Selectors

Tag, Class and ID Selectors can be combined

ul.publications

defines a rule for elements `` of `class="publications"`

div#top

defines a rule for element `<div id="top">`

table #top

defines a rule for some element of `id="top"` contained within a `<table>`

#top table

defines a rule for `<table>` elements contained within some element of `id="top"`

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

24

Selectors Summary

	CSS	XHTML
ID Selector	#myid	id="myid"
Tag Selector	table	<table>...</table>
Class	.talks	class="talks"
Tag+Class	ul.talks	<ul class="talks">...
Tag+ID	div#nav	<div id="nav">...</div>
Nested Selector	ul li li	Child
Attribute Selector	a[href=]	...


19.9.2007

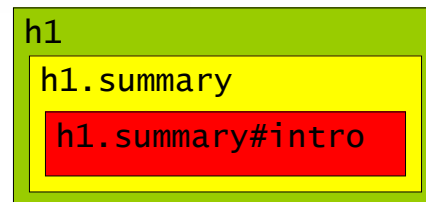
Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

25

Selector Specificity

- Why are style sheets "cascading"?
- Rules with more specific selectors override rules with more generic ones


 Inline Styles
 ID
 Class
 Tag
 Universal Selector



```

<h1>
<h1 class="summary">
<h1 class="summary" id="intro">
<h1 class="summary" id="intro" style="color:navy">
  
```

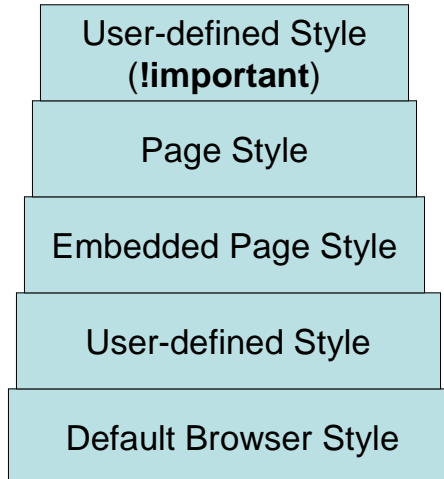
19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

26

Rule Conflicts

- What happens if two rules apply to the same element?
- Rules are prioritized.
 - More specific styles override general styles
 - Page styles replace user-defined styles
- Use the **!important** flag to prioritize styles
- If two styles have the same priority, the one defined “last” takes precedence



19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

27

Pseudo-Classes

Pseudo classes “:” can be applied to selectors and they specify a “dynamic” state of the element.

Pseudo-classes applicable to link elements:

- **a:link** is an unvisited link
- **a:visited** - already visited link
- **a:hover** - is a link (but in Firefox also many other elements) with the mouse cursor over it
- **a:active** - a link while being clicked on

For best results, make sure they are always declared in this order! (LoVeHAte)

```
a {
text-decoration: underline;
}
```

```
a:link { color: blue; }
```

```
a:visited { color: purple; }
```

```
a:hover {
text-decoration: none; }
```

```
a:active { color: red; }
```

```
input:focus { background-
color: yellow; }
```

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

28

Layout with CSS

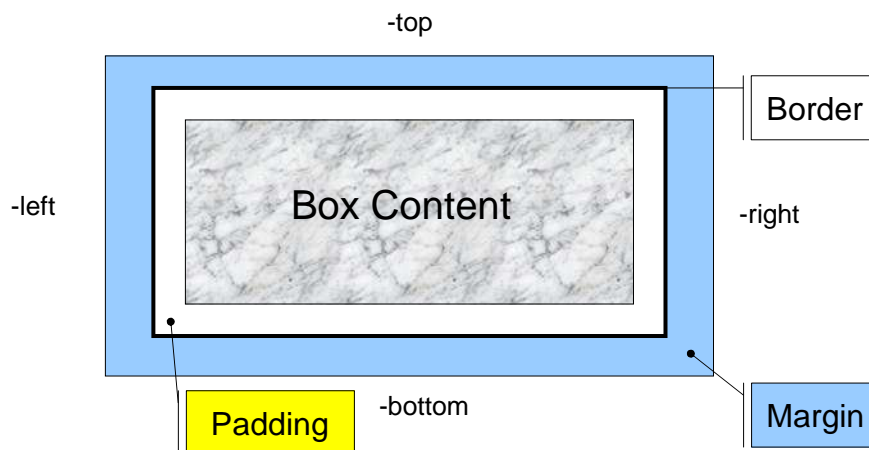
- Much better than using HTML tables (originally it was the only way, but no more!)
 - More control over the positioning and size of each page box element:
 - Margin
 - Padding
 - Border
1. Use `<div id="myid">` elements to logically break down your main page layout boxes.
 2. Specify their layout properties:
 - Display Type
 - Dimension
 - Positioning
 - Floating

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

29

CSS Box Model



19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

30

Display

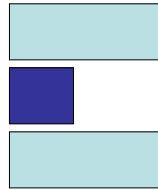
`display:inline`

- Do not break the flow of the text



`display:block`

- Line break before and after the element.



<div>

`display:none`

- The element is hidden



19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

31

Position

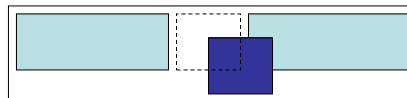
`position:static`

- Automatic layout (ignore left, top, right, bottom)



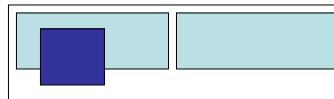
`position:relative`

- Offset the position respect to the automatic one.



`position:absolute`

- Position with respect to the parent box (or the whole page). The element is pulled out of the normal layout flow



`position:fixed`

- Position with respect to the window (no matter how big the page is) – *only works in Firefox and IE7/Strict mode*

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

32

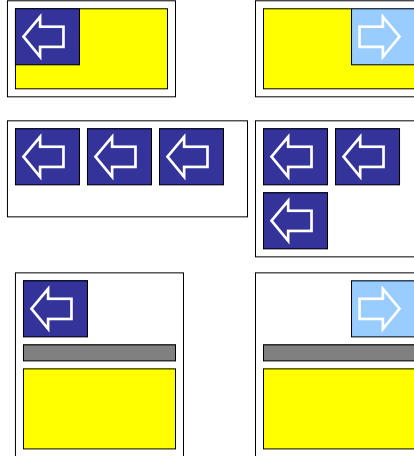
Floating

float: right/left/none

- Shift a box to the right/left of the container box.
If there is no space, boxes will wrap around.

clear: left/right/both

- Force following boxes to continue under the previous highest floated box (like \clearpage in LaTeX – “page” break)



19.9.2007

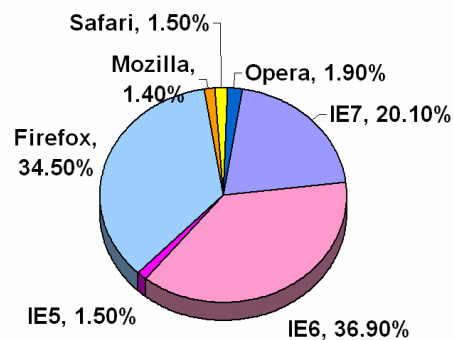
Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

33

CSS2 Web Browsers

- CSS 2.0 was standardized in 1998 but it is still *not fully and correctly* supported by all versions of all Web browser (IE)
- Unfortunately, we cannot ignore some non compliant Web browsers due to their widespread usage
- Some tricks and hacks can be used to write browser-specific CSS code.
- For the exercises, always use the CSS validator to make sure your code is standard compliant:

<http://jigsaw.w3.org/css-validator/>



Browser Market Share (July 2007)

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

34

Supported CSS2 Features

Browser	Internet Explorer					Netscape		Gecko	Safari	Opera
	Windows					Mac	All	All	All	All
Version	4.0	5.0	5.5	6.0	7.0	5.0	4.x	1.0+	1.2+	7.0 +
Type selectors	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Class selectors	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ID selectors	Y	Y	Y	Y	Y	Y	B	Y	Y	Y
Descendent selectors	Y	Y	Y	Y	Y	Y	B	Y	Y	Y
Link pseudo class selectors	Y	Y	Y	Y	Y	Y	B	Y	Y	Y
Dynamic pseudo class selectors	N	N	N	N	P	P	N	Y	Y	Y
Pseudo element selectors										
:first-line	N	N	Y	Y	Y	Y	N	Y	Y	Y
:first-letter	N	N	Y	Y	Y	Y	N	Y	Y	Y
:before	N	N	N	N	N	N	N	Y	Y	Y
:after	N	N	N	N	N	N	N	Y	Y	Y
Selector groups	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Language pseudo class selectors	N	N	N	N	N	Y	N	N	N	N
Child selectors	N	N	N	N	Y	Y	N	Y	Y	Y
First child selectors	N	N	N	N	Y	Y	N	Y	Y	Y
Adjacent selectors	N	N	N	N	Y	Y	N	Y	Y	Y
Attribute selectors	N	N	N	N	Y	N	N	Y	Y	Y

19.9.2007

Fall Semester 2007
Software Atelier III

35

http://westciv.com/style_master/academy/browser_support/selectors.html

References

- <http://www.w3schools.com/xhtml/>
- <http://www.w3schools.com/css/>
- <http://www.alistapart.com/articles/>
- Andy Budd, Cameron Moll, Simon Collison, **CSS Mastery – Advanced Web Standards Solutions**, APress 2006 <http://www.cssmastery.com/>
- Andy Clarke, **Transcending CSS. The Fine Art of Web Design**, New Riders ISBN 0321410971 - <http://www.transcendingcss.com/>
- W3C, **XHTML 1.0: The Extensible HyperText Markup Language** <http://www.w3.org/TR/xhtml1/>
- W3C, **CSS 2.0 Specification** <http://www.w3.org/TR/REC-CSS2/>

19.9.2007

Fall Semester 2007
Software Atelier III
©2007 Cesare Pautasso

36