

# Cascading Style Sheets for XML

Code: xml-css

## Author and version

- Daniel K. Schneider
- Email: Daniel.Schneider@tecfa.unige.ch
- Version: 0.9 (modified 22/1/07 by DKS)

## Prerequisites

- XML editing
- Some CSS with HTML

## Availability (including example files)

url: <http://connections.webster.edu/> COAP 2180 Course Homepage

These slides: Files / teaching\_materials / slides / **css-intro.pdf**

Example files: Files / teaching\_materials / example\_css /



## Disclaimer

- There may be typos (sorry) and mistakes (sorry again)
- Please also consult a textbook and/or a good CSS cheatsheet !

## Objectives

- Recall of some CSS 1 and CSS 2 principles
- Understand how to use CSS with XML

# 1. Table of contents

1. Table of contents	3
2. Cascading Style Sheets principles	4
2.1 Purpose of CSS and status of CSS 2 implementation	4
2.2 Syntax of CSS declarations	5
2.3 Resources on the web	6
3. XML with CSS	7
3.1 Association of a style sheet	7
3.2 Useful CSS2 selectors	8
3.3 Cascading and inheritance	10
3.4 Summary of CSS2 selectors:	11
4. Recall of some CSS properties	12
4.1 Most important typographic element types	12
4.2 Comments	13
4.3 Font properties	13
4.4 Text alignment	14
4.5 CSS Box structure	15
4.6 Some complete examples	17
5. Positioning	19
6. Data-centric XML with CSS	21
6.1 The CSS "content" property	21
6.2 Use XHTML tags to display pictures	23
7. Some advice	24
7.1 First operations when writing a CSS for XML	24
7.2 If your stylesheet doesn't display as it should	25
8. Next steps	26
8.1 Reading	26
8.2 Next modules	26

<b>9. Homework: mini-project 2</b>	<b>27</b>
<b>9.1 Task</b>	<b>27</b>
<b>9.2 Approximate evaluation grid</b>	<b>28</b>
<b>9.3 Submission format and procedure</b>	<b>29</b>

## 2. Cascading Style Sheets principles

### 2.1 Purpose of CSS and status of CSS 2 implementation

- Rendering of HTML and (text-centric) XML contents
- DHTML, dynamic XML, dynamic SVG etc. (in particular: appear/disappear, move, etc.)

#### Advantages

- Separation of content and style: makes web sites easier to maintain
- Multiple rendering: adaptation to media and people (screen size, font size, print, etc.)
- An easy way to render contents of text-centric XML

#### Disadvantages

- lack of text-transformation in CSS1/CSS2 makes CSS rather unsuitable for data-centric XML
- Implementation of CSS 2 is bad in IE 6 / 7. In particular, the content property is missing. It is needed to display attribute values and/or add extra text to output.

#### Implementation

- CSS 1 (1996): ok in Firefox/Opera, more or less ok in IE 6
- CSS 2 (1998): more or less ok in Firefox/Opera, not too good in IE 6/7
- CSS 3 (under construction)

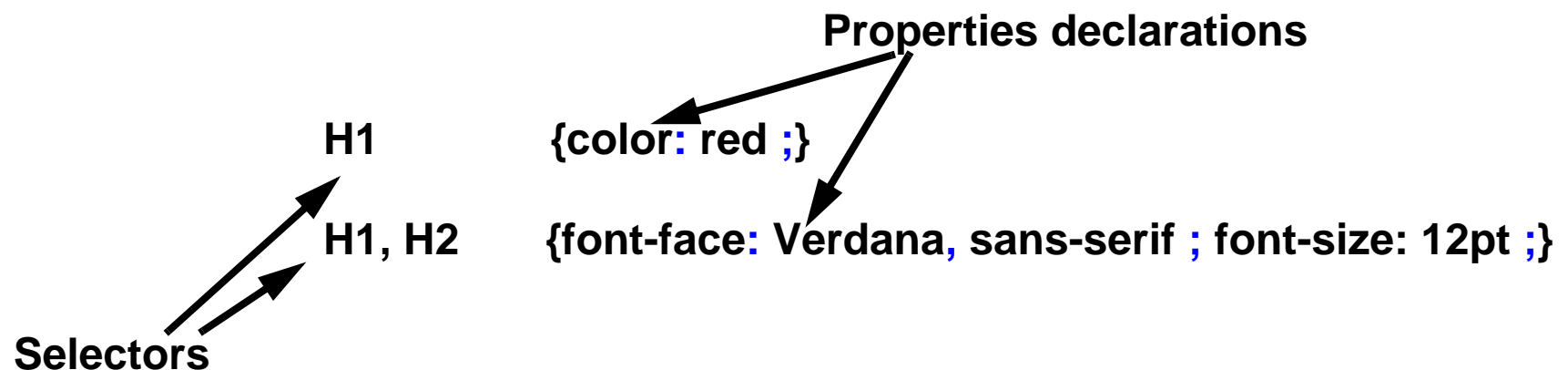
Hint: Use browser compatibility tables when you plan for a larger audience

## 2.2 Syntax of CSS declarations

- Style sheet = set of rules that describe how to render XML or HTML elements
- Each rule has two parts:
  - The **selector**: defines to which elements a rule applies
  - The **declaration**: defines rendering, i.e. values of CSS properties

### Example 2-1: A simple HTML example

```
H1 { color: red }  
P { font-face: Verdana, sans-serif ; font-size: 12pt}  
H1, H2, H3 { color : blue }  
H1.ChapterTOC, H2.PeriodeTOC, H2.ExerciceTOC, H2.SectionTOC {  
    display: block;text-indent: 30pt;  
    text-align: left; font-size: 14.000000pt;  
    font-weight: Bold; font-family: "Times";  
}
```



## 2.3 Resources on the web

### Standards

url: <http://www.w3.org/Style/CSS/> (CSS page of the W3C)

url: <http://www.w3.org/TR/REC-CSS2/> (CSS 2 specification)

### Overviews and cheat sheets

url: <http://refcards.com/download/bj/css2.pdf>

url: <http://www.veign.com/downloads/guides/qrg0007.pdf>

url: <http://www.ilovejackdaniels.com/cheat-sheets/css-cheat-sheet/>

url: <http://www.visibone.com/html/> (good commercial cheat sheets)

url: <http://home.tampabay.rr.com/bmerkey/cheatsheet.htm> (by example)

url: <http://lesliefranke.com/files/reference/csscheatsheet.html>

### XML and CSS Tutorials

- The CSS2 specification covers both HTML and XML !

url: <http://www.zvon.org/xxl/CSS2Tutorial/General/htmlIntro.html> ZVON CSS 2 Tutorial

... there are many more tutorials on the Web.

### Compatibility tables

url: <http://www.quirksmode.org/css/contents.html> (consult this for IE 6/7!)

### CSS Validator (use it please !)

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

## 3. XML with CSS

### 3.1 Association of a style sheet

**CSS stylesheets are associated with the following processing instruction**

```
<?xml-stylesheet type="text/css" href="some_name.css"?>
```

Please note: this is different from HTML !

#### **Example 3-1: Association of a CSS stylesheet to an XML file**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="stepbystep.css" type="text/css"?>
<!DOCTYPE Stepbystep SYSTEM "stepbystep-ex.dtd">

<Stepbystep xmlns:xlink="http://www.w3.org/1999/xlink">
  <Doctitle>Instructions </Doctitle>
  .....
</Stepbystep>
```



## 3.2 Useful CSS2 selectors

- XML needs a navigator that supports at least partially CSS2
- These selectors also work with HTML ...

### *selection of an element (mostly you will use this)*

**Syntax:** element

example:

```
Step {  
    display: list-item;  
    list-style-type: decimal;  
}
```

### *selection of a child element*

**Syntax:** mother\_element > child\_element

Example:

```
Step > Title { .... }
```

### *selection of descendant element (child, great-child, etc.)*

**Syntax:** mother\_element element

example:

```
Step Title { .... }
```

### *combinations*

example:

```
DIV OL>LI P
```

***selection siblings (elements next to each other sharing the same parent)***

**Syntax:** `sister_element + sister_element`

example:

```
H1 + H2 { margin-top: -5mm }
```

***selection of an element that has a certain attribute***

**Syntax:** `element[attribute]`

example:

```
Title[status] { color: blue; }
```

(all titles that have a status attribute are rendered in blue )

***selection of an element that has an attribute with a given value***

**Syntax:** `element[attribute="value"]`

example:

```
Title[status="draft"] { color: red; }
```

***selection of an element that has an attribute with a given value in a comma-sep. list***

```
Title[status~="draft"] { color: blue; }
```

## 3.3 Cascading and inheritance

### Rule ordering

- (Roughly speaking): ***the last rule found will win.***
- E.g. if you define text color in more than one place, the `color:` property found in the last rule encountered will be used

### Inheritance of properties from parents

- ***Child elements usually inherit properties from the parent elements !!!***
- If you don't like this you have to change explicitly these properties

### Example 3-2: Inheritance of properties

#### XML

```
<section>
  <title>Here is a title</title>
  <para>Here is a paragraph</para>
</section>
```

#### CSS

```
section {font-family:Arial}
title {font-family:Helvetica}
/* para will inherit font-family from section, i.e. Arial */
```

## 3.4 Summary of CSS2 selectors:

Pattern	Meaning
*	Matches any element.
E	Matches any E element (i.e., an element of type E).
E F	Matches any F element that is a descendant of an E element.
E > F	Matches any F element that is a child of an element E.
E:first-child	Matches element E when E is the first child of its parent.
E:link E:visited	Matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited).
E:active E:hover E:focus	Matches E during certain user actions.
E + F	Matches any F element immediately preceded by an element E.
E[foo]	Matches any E element with the "foo" attribute set (whatever the value).
E[foo="warning"]	Matches any E element whose "foo" attribute value is exactly equal to "warning".
E[foo~="warning"]	Matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".
E[lang ="en"]	Matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en".
DIV.warning	HTML only. The same as DIV[class~="warning"].
E#myid	Matches any E element ID equal to "myid".

## 4. Recall of some CSS properties

**Syntax:** `property:value;`

**Syntax:** `property:value,alternative_value1,alternative_value2,...;`

### 4.1 Most important typographic element types

1. Blocks, i.e. elements that should start a new paragraph

HTML examples: `<p>`, `<h2>`, `<div>`

2. Lists and list elements

3. Inline elements

HTML examples: `<b>`, `<strong>`, `<span>`

4. Tables

5. Of course, you also can decide to use absolute positioning to place elements ...

### The Display attribute

- The first operation when dealing with XML is to define the display property for each element
- Warning: Some property values (like tables) do not work with IE

Examples that work with most browsers:

```
display: block;  
display: inline;  
display: list-item;
```

## 4.2 Comments

- Comments begin with the characters "/\*" and end with the characters "\*/". They may occur anywhere between tokens, and their contents have no influence on the rendering.
- Comments may not be nested.

### Example:

```
/* Paragraph elements */  
para {display:block;} /* para elements are blocks */
```

## 4.3 Font properties

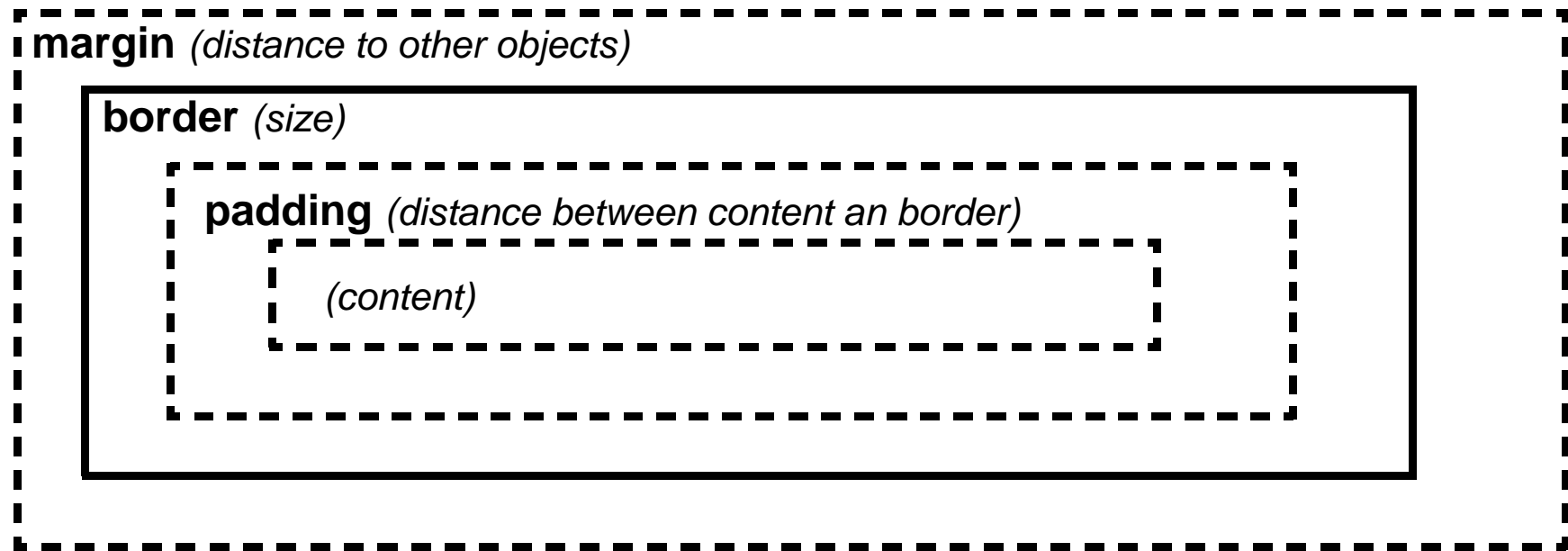
property	Typical values	explanation	example
font-family	font_name	Name of font	font-family: Helvetica;
	font_type	Generic name of font	font-family: serif;
font-size	pt, cm	size	font-size: 14pt;
font-style	normal	normal	
	italic	italic	font-style: italic;
font-weight	number between 100 and 999	thickness	font-weight: 500;
	normal	value = 400	font-weight: normal;
	bold	value = 700	font-weight: bold;

## 4.4 Text alignment

property	values	explanation	example
<b>text-align</b>	left	Paragraph alignment	<code>text-align: left;</code>
	center		<code>text-align: center;</code>
	right		<code>text-align: right;</code>
	justify		<code>text-align: justify;</code>
<b>text-indent</b>	pt, cm	First line indent	<code>text-indent: 1cm;</code>
<b>line-height</b>	pt, cm	line height	<code>line-height: 14pt;</code>
	relative value	font-height * value	<code>line-height: 1.2;</code>

## 4.5 CSS Box structure

Each element is a box



- There are properties for each of these components (see next page for a sampler)
- For some properties, there are shortcuts



## Borders, margins and colors properties (there are more)

property	values	explanation	example
<b>margin</b>	pt, px, cm, %	All 4 margins	<code>body {margin:1cm;}</code>
<b>margin-top</b>		on top	<code>p {margin-top:10px;}</code>
<b>margin-bottom</b>		below	<code>h3 {margin-bottom:3pt;}</code>
<b>margin-left</b>		to the left	<code>img {margin-left:50px;}</code>
<b>margin-right</b>		to the right	<code>p.citation {margin-right:10pt;}</code>
<b>border</b>	pt,px, cm, %	thickness	<code>p {border:5px;}</code>
<b>border-top</b>			<code>h1 {border-top:0.2cm;}</code>
<b>border-style</b>	solid	simple line	<code>p {border-style:solid;}</code>
	double	double line	<code>h1 {border-style:double;}</code>
<b>padding</b>	pt,px,cm,%,etc	padding size	<code>para {padding: 5px;}</code>
<b>color</b>	value hexa or color name	text color	<code>#menu {color:#000000;}</code> <code>body {color:blue;}</code>
<b>background</b>		background color	<code>section, h2 {background:blue;}</code>

## 4.6 Some complete examples

### Example 4-1: Simple page

#### XML: simple-page.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="simple-page.css" type="text/css"?>
<page updated="jan 2007">
  <title>Hello friend</title>
  <content> Here is some content  </content>
  <content> Here is some more content :) </content>
  <comment> Written by DKS/Tecfa </comment>
</page>
```

#### CSS: simple-page.css

```
/* Definitions that apply to the whole hierarchy */
page { font-family:Times; line-height:1.5;}
/* Margins for the box of the root element */
page { margin-top:3cm; margin-left:3cm; margin-right:3cm; }

/* Block elements */

title, content, comment { display:block; }

title { font-family: Arial; font-size:1.5em;}
content { }
comment { font-style:italic; }
```

## Example 4-2: Simple list

### simple-list.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="simple-list.css" type="text/css"?>
<page updated="jan 2007">
  <title>Hello friend</title>
  <list>
    <item> Here is an item that will be somewhat longer. Here is an item that
will be somewhat longer. </item>
    <item> Here is item B</item>
    <item> Here is a C item </item>
  </list>
  <comment> Written by DKS/Tecfa , jan 2007 </comment>
</page>
```

### simple-list.css

```
/* Definitions that apply to the whole heirarchy */
page { font-family:Times; line-height:1.5;}
page { margin-top:3cm; margin-left:3cm; margin-right:3cm; }
title, list, comment { display:block; }

title { font-family: Arial; font-size:1.5em;}
item {display:list-item; list-style-position:outside;
      list-style-type: disc; }
comment { font-style:italic; }
```

## 5. Positioning

- By default elements of an XML (or HTML) file are displayed in sequential order
- It is possible to put an element wherever you wish
- Positioning is not easy (avoid if you are new to CSS)

### Example 5-1: Simple absolute positioning

#### XML: simple-positioning.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="simple-positioning.css" type="text/css"?>
<page updated="jan 2007">
  <title>Hello friend</title>
  <hotstuff>
    <item> Here is an item that will be somewhat longer. Here is an item that
will be somewhat longer. </item>
    <item> Here is item B</item>
    <item> Here is a C item </item>
  </hotstuff>
  <content>
    <para> Here is some standard content. Here is some standard content. Here
is some standard content. Here is some standard content. Here is some standard
content. Here is some standard content. Here is some standard content. Here
is some standard content. </para>
    <comment> Written by DKS/Tecfa , jan 2007 </comment>
  </content>
</page>
```

## CSS: simple-positioning.css

```
/* Definitions that apply to the whole heirarchy */
page { font-family:Times; line-height:1.5;}

/* Margins for the box of the root element */
page { margin-top:3cm; margin-left:3cm; margin-right:3cm; }

/* Block elements */

title, hotstuff, content, comment { display:block; }

title { font-family: Arial; font-size:1.5em;}
content { position: absolute; left: 0; width: 60% }
hotstuff { position: absolute;
    right: 0;
    width: 20%;
    font: 10px/14px verdana, sans-serif;
    color: white;
    margin: 5px 5px 5px 5px;
    padding: 1cm;
    background-color: black; }

item {display:list-item; list-style-position:outside; list-style-type: disc;
}
comment { font-style:italic; }
```

## 6. Data-centric XML with CSS

### CSS isn't made for data-centric XML:

- There is no data transformation (e.g. to add extra text)
- There is no easy way to display attribute values.

... but there are a few tricks for CSS 2 browsers, .e.g use the content or table properties (not implemented in IE 6/7 !)

### 6.1 The CSS "content" property

- Allows to deal somewhat with data-centric XML (not implemented in IE 6/7)

#### Example 6-1: Content property example

#### XML: simple-content.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="simple-content.css" type="text/css"?>
<page updated="jan 2007">
  <title>Hello friend</title>
  <list>
    <item price="10"> White plate </item>
    <item price="20"> Gold plate </item>
    <item price="15"> Silver plate </item>
  </list>
  <comment> Written by DKS/Tecfa , jan 2007 </comment>
</page>
```

## CSS: simple-content.css

```
/* Definitions that apply to the whole heirarchy */
page { font-family:Times; font-size:14pt; line-height:1.5;}

/* Margins for the box of the root element */
page { margin-top:2cm; margin-left:2cm; margin-right:2cm; }

/* Block elements */

title, list, comment { display:block; }
title { font-family: Arial; font-size:1.5em;}

list:before { content:"Products on sale:"; font-size:1.2em; }
item        { display:block; }
item:after  { content:" - Price: " attr(price) " CHF";}

comment { font-style:italic; }
```

### The :before and :after selectors

- can be used to add contents before or after element contents (doesn't work with IE6/7).

### The content property:

- can access attribute values: attr(attribute\_name)
- can add extra information strings.

## 6.2 Use XHTML tags to display pictures

- Pictures inserted into XML can be used to convey extra information to the reader
- I strongly discourage this, since that way data isn't anymore separated from styling
  - rather wait for XSLT !!!

### XML: simple-content-htmlns.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="simple-content-htmlns.css" type="text/css"?>
<page xmlns:html="http://www.w3.org/1999/xhtml" updated="jan 2007">
  <title>Hello friend</title>
  <list>
    <html:img src="photo.jpg"/>
    <item price="10"> White plate </item>
    <item price="20"> Gold plate </item>
    <item price="15"> Silver plate </item>
  </list>
  <comment> Written by DKS/Tecfa , jan 2007 </comment>
</page>
```

### CSS: simple-content-htmlns.css

- same as simple-content.css



## 7. Some advice

### 7.1 First operations when writing a CSS for XML

1. Use the root element to define margins, default font, etc.
2. Decide which elements are blocks and which ones are inline
3. Identify "special elements" like titles and lists

#### Some example CSS rules

```
/* title and para elements are blocks. They have an extra margin */  
title, para {display: block; margin: 0.5em;}  
/* title element font is 1.5 as big */  
title {font-size: 1.5em;}  
/* item elements are list elements, we use bullet style */  
item {display: list-item;list-style-type: disc;}  
/* strong is an inline element. Uses italic style and blue color */  
strong {display: inline; font-style: italic; color: rgb(000,000,128);}
```

- All example XML and CSS files can be found in the "example-css" directory (you may try to play with these a bit before doing your own project)

## 7.2 If your stylesheet doesn't display as it should

1. Validate your CSS (submit the CSS file): <http://jigsaw.w3.org/css-validator/>
  - Missing punctuations in property declaration (": " or "; " or ", ")
  - misspelled property names or values ???
  - missing brace { .... ???
2. Check spelling of element names
  - the on-line CSS validator will not detect this !
3. Check compatibility of your browser or at least check with Firefox
  - remember IE6/7 only partially support CSS 2 (dated 1998)
4. Remember that most properties are inherited from parent elements !
  - you may have to change a property value in a child
5. Do not use the "class" shortcut you learned for HTML
  - in addition, it is very unlikely that your DTD includes a class attribute

## 8. Next steps

### 8.1 Reading

- These slides may not be enough to understand, so please read: Carey (pp. 227- 285), but **ignore** the `html:img src=".."` tricks, it's not XML compliant !  
Optional: 1 or 2 case problems  
Note: This course is not about CSS, you are not required to master all CSS details

### 8.2 Next modules

#### Module 3

- Learning how to write your own DTD
- Some more details about DTDs (we didn't cover everything in module 1)

#### Module 4

- XML Namespaces
- Microsoft XML Data Islands and binding to HTML

#### Module 5

- XSLT introduction (XML data transformations)

## 9. Homework: mini-project 2

Due: Monday jan 29 16:00h

### 9.1 Task

#### Render an XML document with CSS

- **You may reuse the XML document made for project 1 !**
- Alternatively:
  - you can work with an other DTD suggested for mini project 1
  - or use the XML document produced by one of your classmates for homework 1 if the other person agrees (will not affect grading).

#### XML contents: The same rules as for project 1 apply

- Respect the semantics of the elements and the attributes
- Validate your document
- Try to use as many different elements as you can (if appropriate)
- Follow additional directions for each suggested DTD

#### Some advice

- CSS can be time consuming, start by doing a simple version first !
- You can edit your CSS with any XML or HTML editor in principle

## 9.2 Approximate evaluation grid

Minimal work required:	Probable grade
Wellformed (but not valid) document and all elements are styled with CSS	D
Valid XML document, no major CSS mistakes, all elements display.	C
Extra features:	Probable bonus (depends on quality)
Inserted comments <code>/* . . . . . */</code> in the CSS	+
Fully valid CSS	+
Organization and complexity of the CSS (good use of inheritance)	+
The page produced is useful, really looks good and respects some ergonomics	+ .. ++
Write a 1-2 page report that discusses the architecture of the CSS and your opinion of it, e.g. <ul style="list-style-type: none"> <li>• describe architecture of the CSS (without going into detailed description of every rule !)</li> <li>• discuss what you would like to improve, what you liked/disliked, your difficulties, etc.</li> <li>• discuss limits of rendering the document with CSS</li> </ul>	+++

### Examples:

- To get a B+: Firstly produce a valid document, then (a) either write a nice report or (b) document the CSS code and produce a nice document
- To get an A: do all of the above very well

## 9.3 Submission format and procedure

### I require a double submission:

1. Paper copies to be turned in a start of Monday lesson. Don't forget to write your name on it.
2. Electronic copies: <http://connections.webster.edu/>  
Go to the COAP 2180 course. Then **Files / homework/project\_2**.

Please make sure to name files according to the following rules:

File name	Example	when ?
your_name.xml	vasta.xml	This is a mandatory file !
xxx_your_name.css	vasta.css	This is a mandatory file !
your_name.{doc pdf html}	vasta.pdf	only if you decide to write one ....