Separation of content and presentation – XHTML+CSS

Ivana Dusparic
Department of Computer Science
Trinity College Dublin

Separation of structure and presentation

- Using XHTML and CSS – this lecture
- Using XML and XSL – next week

HTML

- HyperText Markup Language (HTML)
  - The language traditionally used to create hypertext pages for use on the Web
  - Supports basic hypermedia document creation and layout
- Why use a special Markup language?
  - Instead of WYSIWYG representation like MS Word
  - Hyperlinks, flexible

Displaying HTML Document

- Not all browsers interpret HTML in the same way
  - Pages may look slightly different depending on what browser is being used
  - Browsers with limited presentation capabilities, e.g., text-only, PDAs, WAP phones, Text To Speech (TTS) converters
- HTML is standardized by the World Wide Web Consortium (W3C) – specification at http://www.w3.org/TR/REC-html40/

Problems with HTML:

- Structure and presentation are mixed
  - Once mixed, it is difficult (often impossible) to separate.
- When we mix content with presentation, we are tailoring the content to be shown in a particular way.
  - Towards a particular web browser
  - Towards a particular OS, fonts, etc.
  - Users with standard devices, e.g., monitor size
- Accessibility issue
  - 1999 lawsuit US National Federation of the Blind vs America Online

XHTML 1.0

- Extensible HyperText Markup Language
  - Stricter and cleaner version of HTML
  - W3C Recommendation January 2000
  - XHTML 2.0 currently in progress
- Allows some separation of structure and presentation
  - Structure expressed in XHTML
  - Presentation expressed as Cascading Style Sheet (CSS)

XHTML 1.0

- Reformulation of HTML 4.0 in XML 1.0
- XHTML documents conform to:
  - XML 1.0
  - HTML 4.0 (if guidelines are followed)
- XHTML documents will work with:
  - Legacy user agents (HTML browsers)
  - Future user agents (XML browsers)#
- XHTML suitable for transition phase to XML

XHTML 1.0

- Contains ***elements*** marked up with ***tags***
- Every document consists of 3 parts:
  - Document type specification
  - Document head
  - Document body
- Document Type Specification – DOCTYPE
  - Describes exactly which type of XHTML the document belongs to - Strict, Transitional, Frameset
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

XHTML 1.0

- Document Head
  - Contains meta-information
  - Title, author, keywords, etc
- Document Body
  - Contains the actual content

XHTML 1.0 tags

- XHTML elements are marked up with tags
  - Always occur in pairs: an opening and a closing tag <p></p>
  - Short-hand notation: <br /> (note the extra space for compatibility with current browsers)
- Some elements have parameters, specified as tag attributes
  - These contain extra information about the element
  - E.g., <a href=“http://www.tcd.ie/”>TCD</a>
  - E.g., <img src=“me.gif” />

XHTML 1.0 tags

- A few basic elements and tags to get you started
  - Headings <h1></h1>, <h2></h2>, …
  - Paragraphs <p>Text</p>
  - Hyperlinks <a href=“…”>Anchor Text</a>
  - Images <img src=“…” />
  - Unordered lists <ul> … </ul>
  - Ordered lists <ol> … </ol>
  - List items <li> … </li>
  - Emphasised text <em> … </em>

XHTML 1.0 tags

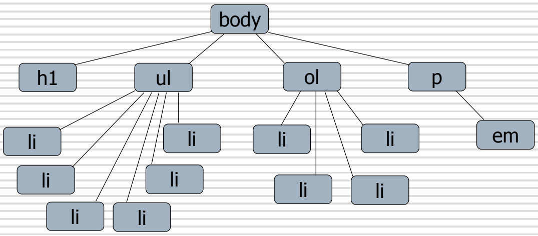
- The <div> tag
  - Used to define logical divisions in XHTML documents
  - Used with CSS to change the style of a portion of text which is not necessarily enclosed in a single tag
  - Can be used to name sections of documents for later referencing (e.g., from JavaScript)
  - Ignored by non-XHTML browsers
  - Acts as a paragraph break and therefore cannot occur inside paragraphs

XHTML 1.0 tags

- The <span> tag
  - Tells browser to apply style rules to a certain portion of a document
  - Can occur anywhere
  - Similar to <div>
- Example

```
<div class=“task”>
  <h2>Assignment 2 is
    <span style="color:red;">
    Important</span></h2>
  <p>Buy ice cream.</p>
</div>
```

XHTML Documents Have Tree Structure



XHTML Comments and Special Characters

- XHTML Documents can also contain comments:
  - Enclosed in <!-- ... --> tags
  - Ignored by the browser
  - <!-- this is a comment -->
- And special characters:
  - <, >, &, é, etc.
  - Use ampersand-semicolon notation
  - Find list of special character notations on:
    - http://www.w3.org/TR/html401/sgml/entities.html#h-24.2.1
    - http://www.cookwood.com/html/extras/entities.html

XHTML vs HTML

- XHTML elements must be properly nested
  - <b><i> this is wrong </b><i>
- XHTML documents must be well-formed
  - all tags closed, properly nested, everything within <html></html> tags
- Tag names must be in lowercase
- All XHTML elements must be closed
- In strict XHTML some HTML elements deprecated
  - <font>, <u> etc

Cascading Style Sheets - CSS

- Defined by the W3C
  - http://www.w3.org/Style/CSS/
- Contain advice to browsers as to how documents should be displayed
- Browser without CSS support simply ignore it - CSS does not break HTML/XHTML conventions
- Multiple style sheets can exist and be supplied to browsers with different capabilities
  - XHTML document remains unchanged

CSS Declarations

- made up of 3 parts:
  - selector, property (attribute), value
  - selector {attribute:value;attribute:value}
- Examples:
  - h1,h2 {text-align:center;color:red}
  - body { font-family: "Times New Roman", Serif; color: green }
- Comments in CSS: /*comment*/

CSS Declarations

- A colon (:) separates the attribute name from the value
- A semi-colon (;) separates attribute and value pairs
- Commas are used to separate lists of values "Arial, Helvetica, Sans-serif". List in order of priority, if first one not available will use second one etc.
- When values of attributes contain spaces - as in Times New Roman – they must be in quotation marks, i.e., "Times New Roman

CSS Inheritance and Scope

- Some properties for a given element are inherited by the element’s children
  - e.g., a font-family property declared for a **p** element will be inherited by any **em** elements inside the **p**.
- Properties declared for the child element override inherited properties
- Not all CSS properties are inherited
  - e.g., borders and margins

CSS Classes

- The class selector can be used to define different styles for the same XHTML element:

```
p.right {text-align: right}
p.center {text-align: center}
```

- Example:

```
<p class=“right”>This paragraph is aligned to the
  right.</p>
<p class=“center”>This paragraph is centered.</p>
```

CSS Classes

- The class selector can be used to define different styles for any element:

```
.green {font-color: green}
.red {font-color: red}
.purple {font-color: purple}
```

- Example:

```
<h1 class=“red”>A Red Heading</h1>
<p class=“green”>A green paragraph</p>
<div class=“purple”>Something purple</p>
```

Using CSS

- Three ways to use CSS declarations in XHTML document
  - External stylesheets
  - Embedded stylesheets
  - Inline stylesheets
- Priority order:
  - Inline style
  - Internal style sheet
  - External style sheet
  - Browser default

Using CSS

- External Stylesheets
  - Style declarations kept in a separate file
  - Link to it by using ***link*** tag inside ***head*** tag
  - Useful when applying same style to multiple pages
- Example:

```
<html>
  <head>
    <title>Page title</title>
        <link rel="stylesheet“
      href="mystyles.css" type="text/css" />
  </head>
  <body></body>
</html>
```

Using CSS

- Embedded stylesheet
  - Placed within the head element of XHTML document
  - Useful for individual web pages
- Example:

```
<html>
   <head>
       <title>This is a title</title>
    <style type="text/css">
         body {background: white; color: blue;}
         h1 {font-family: Arial, Helvetica,
          Sans-serif; }
    </style>
  </head>
<body></body>
</html>
```

Using CSS

- Inline Stylesheets
  - Tied to individual elements within a page
  - Useful for once-off styles
  - Use XHTML syntax (no braces)
- Example:

```
<html>
    <head>
     <title>This is a title</title>
    </head>
    <body style="background: white">
     <h1 style="font-family: Arial,
     Helvetica, Sans-serif; color:
     blue">This is Heading 1 text</h1>
   </body>
</html>
```

Applying CSS

- Cascading style sheets are applied on the client side, i.e., by the browser.
- Client Side
  1. Client requests (and server responds with) XHTML document
  2. Client requests (and server responds with) CSS stylesheet
  3. Client displays document, taking into account advice given in stylesheet

Validation of XHTML and CSS

- The W3C runs a validation service that will validate XHTML and CSS documents on the web
- XHTML validator
  - http://validator.w3.org/
- CSS validator
  - http://jigsaw.w3.org/css-validator/

Problems with HTML and XHTML

- Limited support for mathematical formulae and scientific material
  - Currently: embedded images
  - Future: MathML (an XML application)
- No support for drawing graphics (e.g., figures and diagrams)
  - Currently: inlined images
  - Future: Scalable Vector Graphics (SVG) (another XML application)

References

- Previous years' 4ICT12 notes by Mads Haahr
- W3 Schools. www.w3schools.com
  - XHTML tutorial http://www.w3schools.com/xhtml/default.asp
  - CSS tutorial http://www.w3schools.com/css/default.asp