

XML Transformation

Ramesh Sankaranarayanan

Department of Computer Science
Australian National University

COMP3410 IT in E-Commerce

Outline

- 1 Introduction
 - Transforming XML Documents
- 2 Cascading Style Sheets
 - Introduction
 - CSS Syntax
 - Referencing style sheets
 - Style Properties
 - CSS and XML
 - CSS Limitations

Outline

- 1 Introduction
 - Transforming XML Documents
- 2 Cascading Style Sheets
 - Introduction
 - CSS Syntax
 - Referencing style sheets
 - Style Properties
 - CSS and XML
 - CSS Limitations

Transforming XML Documents

Mainly two ways of formatting or transforming XML documents:

- Cascading Style Sheets (CSS)
- Extensible Style Language (XSL)

Outline

- 1 Introduction
 - Transforming XML Documents
- 2 Cascading Style Sheets
 - Introduction
 - CSS Syntax
 - Referencing style sheets
 - Style Properties
 - CSS and XML
 - CSS Limitations

Cascading Style Sheets

CSS

- designed to separate presentation from content in HTML documents
- Formally proposed to W3C in 1995
- Became a recommendation in 1996. This was called CSS1
- In May 1998, CSS2 reached recommendation status with W3C
- Good support in most browsers for CSS1. Good support for CSS2 in Mozilla and Opera. Not so good in IE
- 23 May 2001, CSS3 working draft released. Pretty good support in some browsers, eg, Firefox, Safari, Opera and Konqueror. Limited support in IE.

Why CSS?

Some advantages:

- Changing the style of an HTML document is much easier.
- CSS style sheets can be reused
- Can use same style sheet for all your web pages to get a uniform look and feel. Can change this easily by just changing the style sheet.

CSS Syntax

An example:

```
p {
  display: block;
  margin-bottom: 10px;
  font-family: Arial, sans-serif;
  font-size: 1.2em;
  color: red;
}
```

<p>This is a test paragraph</p>

All paragraphs rendered using this style will have a 10 pixel margin, a font size 1.2 times the normal, red as the text color and use the Arial font

CSS Syntax

CSS Structure

A CSS style sheet consists of a set of rules that determines how a given set of styles is applied to a document. **Selectors** are used to identify portions of an HTML document to which a specific style is to be applied. There are three kinds of selectors:

- **Element type**
- **Attribute class**
- **Attribute ID**

CSS Syntax

Element Type

The element type selector applies a set of rules to a specific element. The paragraph element style specified in the previous slide is one such example. Such a rule applies to all usage of the specified element, in this case the `<p>` element.

CSS Syntax

Attribute Class

Can get finer control using an attribute class.

```
p.bold {
  display: block;
  color: red;
  font-weight: bold;
}
```

This defines a special class of **bold** paragraphs. And here is how you would use this class:

```
<p class="bold">This is bold.</p>
```

The attribute class could be defined as a generic class by defining it as **.bold** (no element attached to the definition).

CSS Syntax

Attribute ID

An attribute ID is not associated with any specific element. It establishes a style rule that can be applied to any element. An example is:

```
#feelingblue {
  color: blue;
  font-family: Arial;
}
```

And here is an example of how this can be used...

```
<h1 id="feelingblue">My Blues Collection</h1>
```

Referencing style sheets

Two ways of doing this:

Style sheets can be both **internal** and **external**. An internal style sheet is specified in the body of the HTML document itself. An external style sheet is stored in a file (ends with prefix **.css**) and is referenced from an HTML file. It is better to use external style sheets in most cases.

Internal Style Sheets

Specified as follows:

```
<html><head>
  <style type="text/css">
    p {
      font-size: 1.5em;
      color: green;
      font-family: Times, serif;
    }
  </style>
</head>
<body>
  <p>This is a test paragraph</p>
</body></html>
```

External Style Sheets

Assume style sheet is **mystyle.css**:

```
<html>
<head>
  <link rel="stylesheet" type="text/css"
    href="mystyle.css">
</head>

<body>
  <p>This is a test paragraph</p>
</body>
</html>
```

Some Style Properties

Display

This determines how the element is displayed. It can take the following values:

- block** Display the element in a box.
- list-item** Display the element as a list item (bullet) in a box.
- inline** Display the element in an inline box.
- none** The element is not displayed.

Some Style Properties

width and height

These determine the width and height of the box in which the element is displayed. The units that can be used to specify the size include **in**, **cm**, **px** and **em**.

border

This is used to define an element's border. There are other properties available to define each aspect of a border such as **border-width**, **border-color** and **border-style**.

Some Style Properties

margin

This is used to set the margins of an element. You can set the margins for individual sides using properties **margin-left**, **margin-right**, **margin-top** and **margin-bottom**.

color

The **background-color** and **color** properties can be used to set the background and text colors of an element, respectively.

text

The **text-align** and **text-indent** properties can be used to set the alignment and indentation of an element, respectively.

Some Style Properties

font

The font properties are used to set the font of the text. The **font-family** property can be used to set the font type; the **font-size** and **font-style** properties are self-explanatory; and the **font-weight** property can be used to set the font weight to **normal**, **bold**, **bolder**, or **lighter**.

Using CSS with XML

Do the following:

Just specify a style for the root element, and then one for as many sub-elements as required. For example:

```
catalogue {  
  display: block;  
  width: 350px;  
  padding: 10px;  
  border-width: thin;  
  border-style: groove;  
  background-color: #c0ffcc;  
  color: blue;  
  text-align: center;  
}
```

Using CSS with XML

Do the following:

```
title {
  display: block;
  font-family: Times, serif;
  font-size: 1.4em;
  font-weight: bold;
}
```

Using CSS with XML

Referencing the CSS file:

Assume that this is saved in a file called **catalogue.css**. Then, we can refer to this in the xml document as follows:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css"
  href="catalogue.css"?>
  ...
```

Limitations of CSS

In general:

If all that is needed is to simply format XML documents for browsers, then CSS does a reasonable job. On the other hand, if some transformation of the XML document is required, then CSS is not sufficient. The document needs to be transformed first using some other application or technology, and then CSS could be applied to the resulting XML (or HTML) document.