Introduction to Cascading Style Sheets

Cascading Style Sheets

- Allow to assign properties at once to all elements
- Advantages over HTML:
    - Save time
    - Easy to make changes in formatting
    - Easy to create a common style of all pages in the Web site

Anatomy of a Style

- A Style is made up of an HTML tag and definitions
- A Definition determines how an element is marked with the tag
- Definition structure:

    Property:Value(s)

- Each property has one or more values
- Examples: font-size, font-weight, font-family, color *etc.*.

    Values: font-size:12pt; font-weight:bold; font-family:"Lucida"

- *Umbrella property*

    Font: bold 12pt "Lucida"

Common Properties

- Tags with common properties can be defined at once as:

    H1,H3,H5{color:green;font-style:italic;font-family:"Lucida"}

Setting up Style Sheets

- **Local** application - Not centralized
- **Internal** Style Sheet - Centralized within current HTML page
- **External** Style Sheet - Centralized within a group of HTML pages or even Web site

Local application

- Style information is placed in the BODY part of HTML document
- Attribute **STYLE** is added to the tag

STYLE="*property1*:value1;
***property2*:value2;…; *propertyN*:valueN"**

Example: **<H2 STYLE="background:blue;color:blue">Information Technology</H2>**

Internal Style Sheet

- Style information is placed in the HEAD part of an HTML document

  <STYLE>
  ***TagName1* {definition1; definition2; … definitionX}…..**
  ***TagNameY* {definition1; definition2;…definitionZ}**
  </STYLE>

External Style Sheets - ESS

- To create an external style sheet:

  1. Create new plain text file;
  2. Add tags with definitions;
  3. Save the file with extension .css.

- ESS is applied via a link applied to multiple documents
- The link code is placed in the HEAD part of HTML document

  <LINK REL=Stylesheet TYPE="text/css" HREF="URL">

  (where **URL** is relative URL of style sheet file)

Formatting Rules

- External style sheets can be applied to all HTML pages, even holding internal and local style sheets.
- HTML files may hold local and internal styles
- The latest style definition is applied.

Defining Styles for Classes

- HTML elements with the same tag may be divided into classes with different format
- To specify a **class**:

  ParentTag.ClassName{definition(s)}

- To apply a style the attribute CLASS is used in the BODY of HTML Document

  <*ParentTag* **CLASS=ClassName>*...text…*</***ParentTag***>**

Defining Styles for a Particular Tag

- To identify individual tags and appropriate styles:

  ParentTag#IDName{definition(s)}

- To apply the style to a particular tag in the BODY of HTML Document:

  <*ParentTag* **ID=ID***Name***>*...*text*...</***ParentTag***>**

Creating Custom HTML Tags

- **DIV** and **SPAN** can be combined with classes and IDs
- **DIV** may contain paragraphs, text, headers; creates block-level tags
- **SPAN** may contain **FONT, B, I,...**; creates inline tags

Block-Level Tags

- In the Style Sheet:

  DIV.ClassName{definition(s)} or

  DIV#IDName {definition(s)}

- In the BODY of HTML documents:

  <DIV CLASS=ClassName> or

  <DIV ID=IDName> Along with **</DIV>**

Inline HTML Tags

- In the Style Sheet:

  SPAN.ClassName{definition(s)} or

  SPAN#IDName {definition(s)}

- In the BODY of HTML documents:

  <SPAN CLASS=ClassName> or

  <SPAN ID=IDName> Along with **</SPAN>**

Defining Styles for Hyperlinks

- In the style description:

  A:*STATE*{definition(s)**}**

- Link states:
  - **LINK** – hasn't yet been clicked
  - **VISITED** – has been click already
  - **ACTIVE** – is being clicked
  - **HOVER** – is pointed

ESS file example: Plain text; no header; code only.

```
h2{color:white;background:blue}
ol{font-size:8pt;font-weight:bold}
li.odd{color:green}
li.even{font-size:14pt}
div{background:yellow}
span{font-size:16pt;color:red}
a:link{color:green;font-size:8pt}
a:visited{color:blue;background:red;font-size:14pt}
a:hover{color:yellow;background:green;font-size:16pt}
```