Tutorial 3 - Working with Cascading Style Sheet (CSS)

Cascading Style Sheets (CSS)

- a **style** defines the appearance of a document element.
- a **style sheet:** collection of styles for a Web page or Website
- style sheets use common language and syntax
- main style sheet standard: **Cascading Style Sheets (CSS)**

CSS history

- developed by WWW Consortium (www.w3c.org), the same organization that develops standards for HTML
- designed to augment HTML, not replace it
- a whole new way of formatting Web pages
- provides several tools not available with standard HTML
- different versions
  - CSS1 released in 1996
  - CSS2 released in 1998
  - CSS3, latest standard being developed

Browser support for CSS

- Internet Explorer
  - Internet Explorer 4.0 provides support CSS1
  - Internet Explorer 5.0 provides the best support for CSS1
- Netscape
  - Netscape's support for CSS1 has been spotty
  - Netscape has been pushing their own style sheet language over CSS, however, Netscape 5.0 may end up supporting CSS1

Benefits of style sheets

- a design tool
- makes website more flexible
- easier to maintain and modify
- more aesthetically interesting
- consistent look

Style types

- three ways of employing CSS in Web pages:
  - **inline styles**
    - styles are added to each tag within the HTML file
    - style only affects that particular tag
  - **embedded** or **global styles**
    - applied to an entire HTML file
    - allowing the Web designer to modify the appearance of any tag in the document
  - **linked** or **external style sheets**
    - placed in an external file, linked with Web pages
    - allowing Web designer to modify the appearance of tags in several documents

Using inline styles

- format a single section, better use inline style
- syntax

```
<tag style="style declarations">
```

  - ***tag*** is the name of an HTML element (h1, h2, p, etc)
  - ***style declarations***
    - the styles defined for the particular tag
    - must be enclosed within double quotation marks
    - use semi-colon separate multiple attributes

```
<tag style="attribute1:value1; attribute:value2">
```

  - e.g.

```
<h1 style="color:gold; font-family:sans-serif">
```


Creating an embedded style

- **embedded style**, a style applied to various sections within a Web page
- use `<style>` tag within the head section
- within `<style>` tag, enclose style declarations
- syntax

```
<style type="style sheet language">
  style declarations
</style>
```

  - ***style sheet language*** identifies the type of style language used in the document
  - default is "text/css" for using CSS

Style declarations

- syntax for style declaration:

```
selector{attribute1:value1; attribute2:value2; ...}
```

  - collection of attributes and values also referred to as ***declaration*** of the selector
- **selector**
  - identifies an element in your document, e.g., a heading
  - identifies attributes and values within the braces for that element
- example

```
<style type="text/css">
  h1 {color: gold; font-family: sans-serif}
</style>
```


Defining a embedded style

Grouping selectors

- apply the same declaration to a group of selectors by including all of the selector names separated by commas
- example:

```
<style type="text/css">
  h1, h2, h3, h4, h5, h6 {color:gold; font-
  family:sans-serif}
</style>
```


Using an external style sheet

- creating a text file containing style declarations
- have the extension "**.css**", though this is not a requirement
- within a style sheet, **<style>** tag is NOT needed, only need style declarations

Linking to style sheets with <link>

- general syntax for using the <link> tag

```
<link href="URL" rel="relation_type"
  type="link_type" />
```

  - ***URL*** is the URL of the linked document
  - ***relation_type*** establishes relationship between linked document and Web page
    - rel="stylesheet" for linking to a style sheet
  - ***link_type*** indicates language used in linked document
    - type="text/css" for linking to a style sheet
- example

  link to a style sheet named "**farm.css**,"

```
<link href="farm.css" rel="stylesheet" type="text/
  css" />
```


Linking to style sheets with @import

- enclose the **@import** command within the embedded **<style>** tags

```
<style type="text/css">
   @import url(company.css)
   @import url(company.css)
</style>
```

  - ***company.css*** is the URL of the style sheet file
- limited browsers support @import
- better off using the **<link>** tag

Style precedence

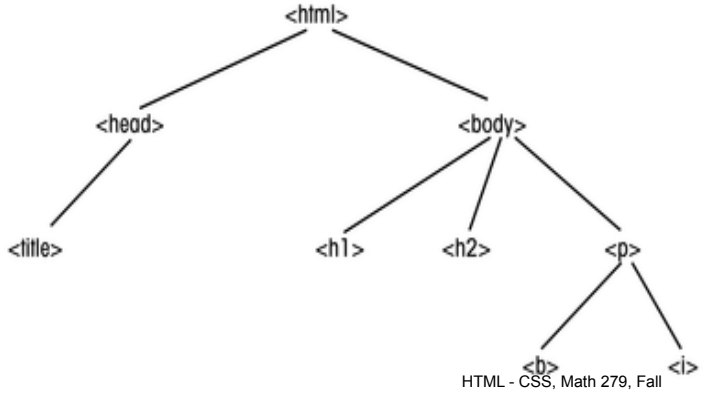
- in cases styles conflict, precedence is determined in the following order:
  - an **inline style** overrides any **embedded style** or **external style sheet**
  - an **embedded style** overrides an **external style sheet**
  - an **external style** sheet overrides the **internal style rules** set by the Web browser

Changing styles

- as a change is made to a style at one level, the changes are cascaded through to the other levels ➔ **cascading style sheets**
- need to keep track of the inline, embedded, and external style sheets to correctly predict the impact that style changes have on the appearance of each page

Working with style inheritance

- Web pages invariably have elements (e.g., tags) placed within other elements
- sample tree structure of Web elements



Parent and descendant elements

- an element that lies within another element is called a **descendant** or **descendant element**
  - e.g., in previous slide, <b> tag is a descendant of <p> tag
- an element that contains another element is called the **parent** or **parent element**.
  - e.g., **<body>** tag is the parent of all other tags used to format the Web contents
- using the principle of **inheritance**, *styles defined for a parent tag are transferred to its descendants tags*

Contextual selectors

- use tree structure to better control styles
  - apply a style only to *direct descendant* of a parent element, use syntax: `e1 e2`
  - ***e1*** and ***e2*** are the names of HTML elements (i.e. tags) and e2 is directly below e1 in the tree structure of elements
  - example: `li b {color:blue}`

    only changes the color of the *boldface* text residing within a <li> tag to *blue*
- not all browsers support contextual selectors

Generally used attributes and their values

Using font families

- The **font-family** attribute allows you to choose a font face for Web content
- CSS works with two types of font faces:
  - **specific font**, which is a font such as **Arial**, **Garamond**, or **Times New Roman** that is actually installed on a user's computer
  - **generic font**, which is a general description of a font, allowing the operating system to determine which installed font best matches it
  - CSS supports five **generic** font types: **serif, sans-serif, monospace, cursive, and fantasy**

Generic fonts

Generic Names	Font Samples		
serif	defg	defg	defg
sans-serif	defg	defg	defg
monospace	defg	defg	defg
cursive	defg	defg	defg
fantasy	DEFG	DEFG	defg

within each generic font there can be a wide range of appearances

Generic, specific fonts issues

- using generic fonts cannot be sure which specific font used
- specific fonts are preferred
- provide several fonts to choose from
- list specific font names first, followed by a generic font name, in case none of specific fonts can be found

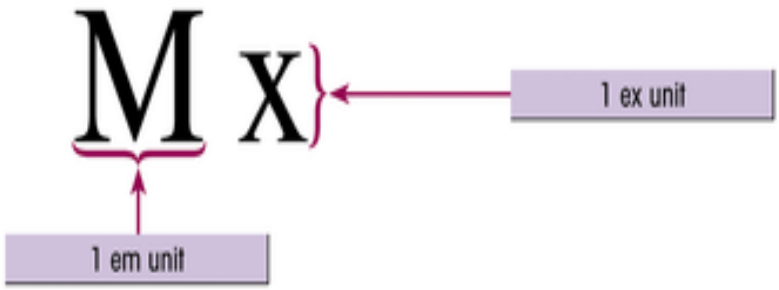
Managing font size

- in CSS, use `font-size` attribute to manage font sizes
- font sizes can be expressed
  - as a unit of length
  - with a keyword description
  - with a keyword expressing the size relative to the font size of parent element
  - as a percentage of the parent element

Absolute and relative units

- use unit of length, **absolute units** or **relative units**
  - **absolute units** define the font size based on standard units of measurement:
    - mm (millimeter),
    - cm (centimeter),
    - in (inch),
    - pt (point)
    - pc (pica)
  - **relative units** express the font size relative to a size of a standard character
    - **em unit** is equal to the *width* of capital letter "**M**" in browser's default font size
    - **ex unit** is equal to the *height* of a small "**x**" in default font size

em, ex units



Pixels

- a pixel is the smallest element recognized by the monitor
- text that is 10 pixels high may be perfectly readable at a low-resolution (e.g., 640 x 480) monitor, but it can become unreadable at high-resolution (e.g., 1024 x 768) monitor

Descriptive keywords

- seven descriptive keywords for font size
  - xx-small
  - x-small
  - small
  - medium
  - large
  - x-large
  - xx-large
- Example

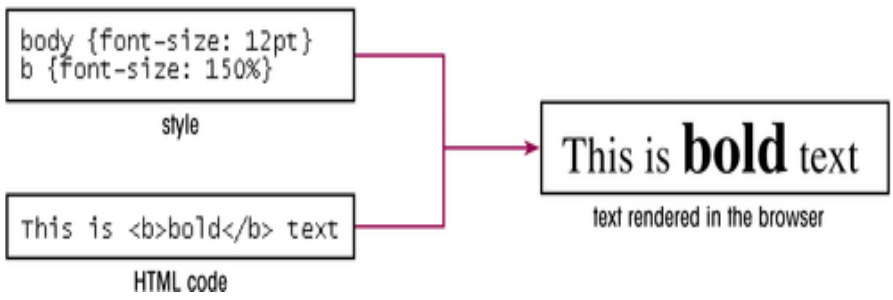
```
body {font-size: medium}
```

Keywords: smaller, larger

- using keywords "**larger**" and "**smaller**,"
  - makes the font **one size** larger or smaller than the size of parent element
  - example: to make h2 heading one size larger than the body text, you could use the following style

```
body {font-size: medium}
h2 {font-size: larger}
```


Font size as percentage of parent tag



The font size of bold content (defined by <b> tag) is 150% of the size of surronding text (where the font size is defined by <body> tag)

Specifying letter, word spacing

- set the space between individual letters, **letter-spacing** attribute

  `letter-spacing: size`

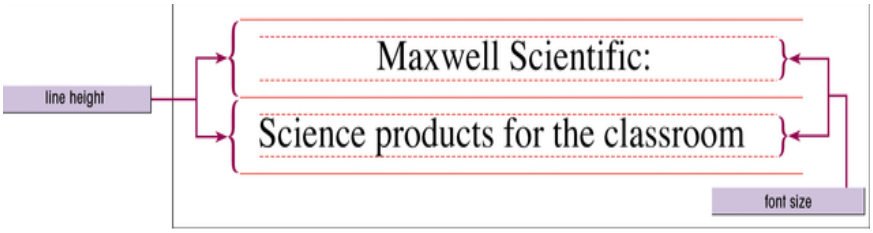
- change the spacing between individual words, **word-spacing**

  `word-spacing: size`

- ***size*** can be
  - the value "**normal**", which allows the browser to determine letter spacing
  - a number expressed in *inches, millimeters, centimeters, em units*, etc.
    - the same measuring units used to describe font size

Specifying line height

- **line-height** attribute modifies vertical space between lines of text, specifies the minimum distance between the baselines of adjacent lines.
- example



Specifying line height (cont.)

- syntax

  `line-height: size`

  - ***size*** is either a specific length, a percentage of the font size, or a number representing the ratio of the line height to the font size
  - standard ratio is 1.2, which means that the line height is 1.2 times the font size
  - example, to make paragraphs double-spaced use the style definition

    ```
    p {line-height: 2}
    ```


Font weights

- CSS considers "**bold**" to be an aspect of the font's weight, or line thickness.
- **font weights** can be expressed as *an absolute number ranging in intervals of 100, going from 100 (the lightest) up to 900* (the heaviest or "most bold").
- for most fonts,
  - a weight of 400 corresponds to normal text
  - a weight of 700 can be used for bold text
  - a weight of 900 for "extra" bold text
- example

  ```
  h2 {font-weight: 700}
  ```

Aligning text horizontally and vertically

- **text-align** attribute

```
text-align: alignment
```

  - ***alignment*** can be left, center, right, or justify
  - setting the text-align value to "**justify**" stretches the text, extending it from the left to the right margin
- **vertical-align** attribute

```
vertical-align: alignment
```

  - ***alignment*** has one of the keyword values

Values for vertical-align

ATTRIBUTE VALUE	DESCRIPTION
baseline	Aligns the element with the baseline
bottom	Aligns the bottom of the element with the bottom of the lowest element (text or image) in the line
middle	Aligns the element in the middle of the text
sub	Aligns the element as a subscript
super	Aligns the element as a superscript
text-bottom	Aligns the element with the font's bottom
text-top	Aligns the element with the top of the tallest letter
top	Aligns the element with the top of the tallest element (text or image) in the line

Special text attributes

- CSS provides three attributes for special text effects:
  - text-decoration
  - text-transform
  - font-variant

text-decoration attribute

- attribute name: **text-decoration**
- values
  - none
  - underline
  - overline
  - line-through
- examples

Maxwell Scientific teaches science
text-decoration:none

Maxwell Scientific teaches science
text-decoration:underline

Maxwell Scientific teaches science
text-decoration:overline

~~Maxwell Scientific teaches science~~
text-decoration:line-through

text-transform attribute

- attribute name: **text-transform**
- attribute value:
  - **capitalize**
    capitalize the first letter of each word in a paragraph
  - **uppercase**
    display the text in all capital letters
  - **lowercase**
    display the text in all lowercase letters

text-transform attribute, example

Maxwell Scientific teaches science
text-transform:none

Maxwell Scientific Teaches Science
text-transform:capitalize

MAXWELL SCIENTIFIC TEACHES SCIENCE
text-transform:uppercase

maxwell scientific teaches science
text-transform:lowercase

font-variant attribute

- **font-variant** attribute create small caps
- small caps are capital letters that are the same size as lowercase letters
- syntax

```
font-variant: small-caps
```

- **Netscape** does not support the font-variant attribute in versions prior to 6.0

font-variant attribute, example

Maxwell Scientific teaches science

font-variant:normal

MAXWELL SCIENTIFIC teaches science

font-variant:small-caps