Cascading Style Sheets (CSS)

CSS

- Cascading Style Sheets (CSS) is the dominant styling technology

  - Versions of CSS:
    - CSS1 - 1996
      - basic typography (font, colors)
    - CSS2 - 1998
      - now fairly well supported across browsers except IE
    - CSS3 is under development

CSS Levels

- There are three levels of Cascading Style Sheets:

<table>
<thead>
<tr>
<th>Level</th>
<th>Applies To</th>
<th>Defined In</th>
</tr>
</thead>
<tbody>
<tr>
<td>external</td>
<td>multiple documents</td>
<td>separate text file</td>
</tr>
<tr>
<td>internal</td>
<td>one document</td>
<td>head section</td>
</tr>
<tr>
<td>inline</td>
<td>one element</td>
<td>opening tag</td>
</tr>
</tbody>
</table>

When there are conflicts, inline styles override internal styles, and internal styles override external styles.

CSS Levels

- The value of CSS comes from consistency and maintenance
  - By defining a style once at a high level, it can be applied over multiple documents consistently
  - Changes occur in one place – the entire look and feel of a site can be changed quickly
  - Multiple levels allow you to override global styles when needed
  - Internal styles and inline styles should be used sparingly
**Style Rules**

- A style rule is made up of a selector and a list of property/value pairs.
- This syntax applies to external and internal levels only.

**General Format:**

```
selector {
  property1: value1;
  property2: value2;
  etc.
}
```

**Example:**

```
h2 {
  text-align: center;
  color: red;
}
```

**External Style Sheets**

- External CSS files are specified using the `<link>` element in the head section of a document:
  ```html
  <link rel="stylesheet" type="text/css" href="/include/envirocorp.css">
  ```
- The file referenced includes a list of style rules; any file that links to a style sheet will adopt those styles.
- Multiple style sheets can be linked into one document.
- The `style` element with an `import` can also be used to link to an external style sheet.

**Internal Style Sheets**

- Internal styles are specified using the `style` element in the head section of a document:
  ```html
  <style type="text/css">
    hr {
      color: sienna;
    }
  </style>
  ```

**Inline Style Sheets**

- Inline styles are specified using the `style` attribute of the appropriate tag:
  ```html
  <p style="padding-left: 50px; color: red">
    whatever
  </p>
  ```
- Note that this approach essentially mixes content and presentation, and should be avoided.
Selectors

- A selector can be a single element name or a list of names separated by commas:

```css
h1, h2, h3, h4 {
    margin: 0;
    padding: 0;
}
```

All elements in the list will have the same styling properties

Class Selectors

- A selector can also refer to a class of elements
- The class attribute establishes that a particular element belongs to a particular class

```html
<p class="intro">
  whatever
</p>
```

- The selector uses a dot notation to refer to a class

```css
p.intro {
    text-indent: 10px;
}
```

Class Selectors

- Multiple elements can be part of the same class
- Omitting the element name in a selector defines a style for all elements in the class

```html
<h3 class="intro">
  A Big Introduction
</h3>
```

Id Selectors

- An id is similar to a class except that an id can be used only once per page
- A hash mark (#) is used to indicate ids in a selector
- As with a class, the element name can be omitted

```html
<p id="book">
  Design Patterns
</p>
```

```css
p#book {
    font-weight: bold;
}
```
CSS Philosophies

- Find the appropriate, highest level at which to define a style
- Provide general rules for all HTML elements used on the site
- Use internal styles sparingly, and inline styles even less
- Share style rules when possible across elements
- Don't use div and span elements when styling basic elements (perhaps with classes and ids) will suffice

CSS Comments

- CSS comments should be used to group and explain the purpose of style rules
- Comments begin with /* and end with */

```html
/* Primary Navigation Styles */
a.primary {
  color: red;
}
```

Property Values

- CSS properties are divided into various categories:
  - Fonts, text, colors and backgrounds, etc.
- There are multiple ways to specify a property value, depending on the particular property
- Some are absolute and some are relative to the current value of the property
- Some are expressed numerically and some using keywords (red, bigger, bold)

Property Values

- Length / size can be expressed in various units:
  - px  pixel  monitor resolution pixels
  - pt  point  font point size
  - pc  pica   1 pica = 12 points = 1/6 inch
  - em  Em     size of M in current font
  - ex  x      size of x in current font (half of M)
  - in  inches U.S. inches
  - mm  millimeters Metric millimeters
  - cm  centimeters Metric centimeters
Property Values

- Percentages can be used in many situations as well, relative to the current value or container size
- URLs are specified using the following notation:
  
  ```
  url(logo.jpg)
  url(../index.html)
  url(images/bullet.gif)
  url(http://www.cnn.com)
  ```

Property Values

- Color can be specified using color names, RGB values, or hex codes
- RGB values can be expressed in the 0-255 range, or as percentages
  
  ```
  rgb(0, 255, 255)
  rgb(50%, 50%, 50%)
  ```

  Hex (rgb) values are preceded with a # sign:
  
  ```
  #FF3300
  #A40075
  ```

Font Properties

- A variety of properties can be set to specify the characteristics of fonts:
  - Font family
  - Size
  - Variant (small caps)
  - Style (italics)
  - Weight (boldness)

  These can be set individually or using a combined (shortcut) property

Font Properties

- The font-family property can be set to a specific font name such as times or courier
- Or it can be set to a generic family: serif, sans-serif, monospace, cursive, or fantasy
- A list of font-family values can be provided in order of preference:
  
  ```
  h1 {
      font-family: arial, helvetica, sans-serif;
  }
  ```
Font Properties

- Other font properties, which generally default to normal:

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>font-style</td>
<td>italic, oblique</td>
</tr>
<tr>
<td>font-weight</td>
<td>bold, bolder, lighter, 100-900</td>
</tr>
<tr>
<td>font-variant</td>
<td>small-caps</td>
</tr>
<tr>
<td>font-size</td>
<td>specific size (example: 12pt)</td>
</tr>
<tr>
<td></td>
<td>xx-small, ... xx-large</td>
</tr>
<tr>
<td></td>
<td>relative (examples: 1.4em, 50%)</td>
</tr>
<tr>
<td></td>
<td>larger, smaller</td>
</tr>
</tbody>
</table>

Font Properties

- Several font properties can be expressed at the same time using the font property.
- If present, the order must be: weight, style, variant, size, and font families.

```html
h3 {
  font: bold 12pt arial, 'times new roman';
}
```

Text Properties

- Some text properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>color name or value</td>
</tr>
<tr>
<td>text-decoration</td>
<td>none, underline, overline, line-through, block</td>
</tr>
<tr>
<td>text-align</td>
<td>center, justify, left, right</td>
</tr>
<tr>
<td>text-indent</td>
<td>length (example: 10px)</td>
</tr>
<tr>
<td></td>
<td>relative to font (example: 3em) percentage of parent (example: 5%)</td>
</tr>
<tr>
<td>text-transform</td>
<td>none, capitalize, lowercase, uppercase</td>
</tr>
</tbody>
</table>

Background Properties

- Some properties affecting an element's background:

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>background-color</td>
<td>color name or value</td>
</tr>
<tr>
<td>background-image</td>
<td>url(...)</td>
</tr>
<tr>
<td>background-repeat</td>
<td>repeat, repeat-x, repeat-y, no-repeat</td>
</tr>
<tr>
<td>background-attachment</td>
<td>scroll, fixed</td>
</tr>
<tr>
<td>background-position</td>
<td>length (example: 30px)</td>
</tr>
<tr>
<td></td>
<td>percentage (example: 25%) top, center, bottom left, center, right</td>
</tr>
</tbody>
</table>
Background Properties

The value of the background-position property is made up of a vertical / horizontal pair:

```css
body {
  background-image: url(map.gif);
  background-position: 20px 45px;
}
```

The values are relative to the top and left edges of the element,
Word values (like right and top) and percentages keep the image within the element, taking the image size into account.

Background Properties

- Like the font property, the background property can be used to specify several values for a background at one time
- The order, if present, is color, image, repeat, attachment, position

```css
body {
  background: black url(nightSky.gif) no-repeat fixed;
}
```

List Properties

- Properties can be set to affect the look of unordered and ordered lists, as well as individual list items
- The list-style-type property can be applied to unordered lists with the values disc (default), circle, square, or none
- The list-style-image property can be used to set the bullet image using a url value
- Both of these properties can be set for individual list items

<table>
<thead>
<tr>
<th>Property Value</th>
<th>Sequence Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>decimal</td>
<td>1, 2, 3, 4, ...</td>
</tr>
<tr>
<td>upper-alpha</td>
<td>A, B, C, D, ...</td>
</tr>
<tr>
<td>lower-alpha</td>
<td>a, b, c, d, ...</td>
</tr>
<tr>
<td>upper-roman</td>
<td>I, II, III, IV, ...</td>
</tr>
<tr>
<td>lower-roman</td>
<td>i, ii, iii, iv, ...</td>
</tr>
</tbody>
</table>
**Box Model**

- Style sheets treat each element as if it were contained in a series of boxes.
- The element content is surrounded by padding, which is surrounded by the border, which is surrounded by the margin.

![Diagram of Box Model]

This represents the element content.

**Margins**

- Margins, borders, and padding are all optional and often have a default size (width) of zero.
- The size of all three features can be set per side.
- Various shorthand properties can also be used.
- Any background applied to an element extends into the padding and under the border, but not into the margin.
- The margin is transparent, and therefore the background of the parent element is visible in the margin area.

**Margins**

- Margin sizes are set using the margin-top, margin-right, margin-bottom, and margin-left properties.
- The values for the right and left margins can be:
  - a specific length
  - a percentage of the width of the containing element
  - auto, based on other elements on the page
- The values for the top and bottom margins can be a specific length or auto, but not percentage.
- Negative values can be used for margins to overlap elements.

```css
img {
  margin: 5px 8px 12px 0;
}
```

If all four values are given, they apply in a clockwise order:

- top
- right
- bottom
- left
Margins

- The margin shortcut property may also be used with one, two, or three values.
- If only one value is given, it applies to all four sides.
- If two values are given, the first applies to the top and bottom, the second applies to the right and left.
- If three values are given, the second applies to both the right and left sides.

Padding

- Padding sizes are specified using the padding-top, padding-right, padding-bottom, and padding-left properties.
- The values are specified as they are for margins.
- The padding shortcut property can be used to specify all padding sizes, and follows the same roles as the margin property.
- Exception: padding cannot be negative.

Borders

- The only important characteristic of padding and margins is their size.
- For a border, however, we can specify its size (width), style, and color.
- Properties:
  - border-top-width
  - border-top-style
  - border-top-color
  - border-right-width
  - border-right-style
  - border-right-color
  - border-bottom-width
  - border-bottom-style
  - border-bottom-color
  - border-left-width
  - border-left-style
  - border-left-color

Borders

- The width value of any border can be a specific length, or it can be thin, medium, or thick.
- The color can be specified using a color name or rgb value.
- The style specifies the way the border looks using the following values:
  - none
  - dotted
  - dashed
  - solid
  - double
  - groove
  - ridge
  - inset
  - outset
Borders

- In addition, shortcut properties for each border characteristic can be used:
  - border-width, border-color, and border-style
  - They specify the values for a particular characteristic for all four sides
  - One, two, three, or four values may be specified, following the same rules as the margin property

```css
ing {  
  border-color: red yellow blue;  
}
```

Borders

- Furthermore, shortcut properties for each side can be used:
  - border-top, border-right, border-bottom, border-left
  - The order, if present, must be width, style, color
  - The border shortcut sets them all for all four sides

```css
h2 {  
  border: 1px dotted red;  
  border-left: 1px solid blue;  
}
```