Introduction To Web Programming I

(CSC 211)

Lecture Note 4

How to Code Selectors

20th January, 2020

- To style any element in a web, you need to select that element and then apply style to it.
- You code a selector for all elements of a specific type by naming the element. This is referred to as a type selector.
- You code a selector for an element with a class attribute by coding a period followed by the class name. Then, the rule set applies to all elements with that class name.
- You code a selector for an element with an id attributed by coding a pound sign(#) followed by the id value.
- **HTML that can be selected by element type, id, or class**
-

```
<main>
        <h1>This Season's Speaker Lineup</h1>
        <p class="blue">October: Francis Ho</p>
        <p class="blue">November: SA Ahmed</p>
  </main>
 <footer>
        <p id="copyright" class="blue right">Copyright 2020</p>
  </footer>
```

- CSS rule sets that select by element type, id, and class
- All elements

```
*{ margin: .5em 1em;}
```

- Elements by type

```
main {
  border: 2px solid black;
  padding: 1em; }
h1 { font-family: Arial, sans-serif; }
p { margin-left: 3em; }
```

- One element by ID

```
#copyright { font-size: 80%; }
```

- Elements by class

```
.blue { color: blue; }
.right { text-align: right; }
```

How to code relational selectors

- *Relational* selectors can be often used to avoid the need for id or class attributes.
- To select elements only when they are ***descendants*** of a higher-level element, use a descendant selector that consists of the higher element, a **space**, and the descendant element.
- To select a ***sibling*** element that is adjacent to another element, use an adjacent sibling selector that consists of the first element, a plus (**+**), and the sibling element.
- To select elements only when they are ***child*** elements of the parent element, you can use a child selector that consists of the parent element, the greater than sign (**>**), and the child element.
- To select any elements that are siblings to another element, you can use a ***general*** sibling selector that consists of the first element, tilde (**~**), and the sibling element, but this can only be used by browsers that support CSS3.

- HTML that can be selected by relationship

```
<main>
        <h1>This Season's Town Hall speakers</h1>
        <ul>
            <li>January: <a href="#">David Day</a></li>
            <li>February: <a href="#">Francis Ho</a</li>
            <li>March: <a href="#">Teo Kim Heng</a></li>
          <li>April: <a href="#">Neil Richardson</a></li>
        </ul>
        <h2>Post-lecture luncheons</h2>
        <p>Extend the excitement by going to the luncheon</p>
        <p>A limited number of tickets are available.</p>
        <p><em>Contact us by phone </em> at (+234) 802 4362 345.</p>
</main>
```

- **CSS rule sets with relations selectors**
- Descendant

  main li { font-size: 90%; }

  ul a { color: green; }
- Adjacent sibling

  h2+p { margin-top: .5em; }
- Child

  main>p { font-size: 80%; }

  li>a { color: green; }
- General sibling(CSS3)

  h2~p { margin-left: 2em; }

How to code combinations of selectors

- To code a selector for an element and class, code the element name, a period, and the class name.
- To code multiple selectors for the same rule set, use commas to separate the selectors.
- To select all elements with a specific attribute, you can use an attribute selector that consists of the universal selector followed by the attribute name in brackets. You can also omit the universal select when you code this type of selector.
- To select elements with a specific attribute, you can use an attribute selector that consists of the element followed by the attribute name within brackets.
- To select an element with a specific attribute value, you can use an attribute selector that consists of the element followed by the attribute name, an equal sign, and a value within quotation marks.
- When you are coding the CSS for an HTML page, you usually don't need attribute selectors, They are mostly more useful when you are using JavaScript.

- Combinations of selectors
  - A selector for a class within an element

    ul.speakers { list-style-type: square; }
  - Multiple selectors

    h1, h2, h3 { color: blue; }
- Attribute selectors
  - All elements with href attribute

    *[href] { font-size: 95%; }
  - All <a> elements with href

    a[href] { font-family: Arial, sans-serif; }
- All input elements with type attributes that have a value of "submit"

```
input[type="submit"] {
    border: 1px solid black;
    color: #ef9c00;
    background-color: #facd8a; }
```

- Common CSS pseudo-classes
  - **:link** - A link has not been visited. By default, blue, underlined text.
  - **:visited** - A link has been visited. By default, purple underlined text.
  - **:active** - The active link (mouse button down but not released). By default, red, underlined text.
  - **:hover** - An element with the mouse hovering over it Code this after :link and :visited.
  - **:focus** - An element like a link or form control that has the focus.
  - **:first-child** - The first child of an element.
  - **:last-child** - The last child of an element.
  - **:only-child** - The only child of an element.
- Common pseudo-elements
  - **::first-letter** - The first letter of an element.
  - **::first-line** - The first line of an element.

- **HTML that can be used by pseudo-class and pseudo-element selectors**

```
<main>
    <p>Welcome to Faculty of Computing.</p>
    <p>We have some fascinating speakers for you this semester!</p>
    <ul>
        <li>January: <a href="#">David Day</a></li>
        <li>February: <a href="#">Francis Ho</a</li>
        <li>March: <a href="#">Teo Kim Heng</a></li>
        <li>April: <a href="#">Neil Richardson</a></li>
    </ul>
</main>
```

- The CSS for pseudo-class and pseudo-element selectors

```
a:link { color: green; }
a:hover, a:focus { color: fuchsia; }
main p:first-child { font-weight: bold; }
main p:first-child::first-letter { font-size: 150%; }
```

- Accessibility guideline
  - Apply the same formatting to the **:hover** and **:focus** pseudo-classes for an element. That way, those who cannot use the mouse will have the same experience as those who can.
  - Pseudo-classes are predefined classes that apply to conditions. In contrast, pseudo-elements let you select a portion of text.

- The term CSS refers to the fact that more than one style sheet can be applied to a single web page. Then, if two or more rules for the same property are applied to the same element, the cascade order and other rules determine which rule takes precedence.
- The cascade order for applying CSS rule sets
  - Search for the rule sets that apply to an element in the sequence that follows and apply the rule set from the first group in which it is found:
    - !important rules in a user style sheet
    - !important rules in a web page
    - Normal rules in a web page
    - Normal rules in a user style sheet
    - Default rules in the web browser

- If more than one rule set at a cascade level is applied to an element...
- Use the rule set with the highest specificity. for example, the p.highlight selector is more specific than the .highlight selector.
- If the specificity is the same for two or more rule sets in a group, use the rule set that is specified last.

- How to determine the specificity of a selector
  - An id is the most specific
  - A class, attribute selector, or pseudo-class selector is less specific.
  - An element or pseudo-element selector is least specific.
- When two or more rule sets are applied to an HTML element, CSS uses the cascade order and rules shown above to determine which rule set to apply.
- A user can create a user style sheet that provides a default set of rules for web pages. Users with poor vision often do this so the type for a page is displayed in large font.
- Since most users do not create user style sheets, you usually can control the way the rules are applied for your websites. But you should keep in mind how your web pages could be affected by user style sheets.

- The fonts specified for the font-family property are searched in the order listed If you can include a font name that contains spaces, the name must be enclosed in quotes.
- If you specify a generic font last and the web browser cannot find any of the other fonts in the list, it will use its default font for the generic font that you specified.
- The font properties that you set for an element are inherited by all of its descendants.
- If you use relative font sizes, the users will be able to vary the sizes by using their browsers. If you use pixels, the font size will vary based on the screen resolution.
- **The five generic font families**
  - **serif** - Fonts with tapered, flared, or slab stroke ends.
  - **sans-serif** - Fonts with plain stroke ends.
  - **monospace** - Fonts that use the same width for each character.
  - **cursive** - Fonts with connected, flowing letters that look handwriting.
  - **fantasy** - Fonts with decorative styling.

- How to specify a font family
  - font-family: Arial, Helvetica, sans-serif;
  - font-family: "Times New Roman", Times, serif;
  - font-family: "Courier New", Courier, monospace;
- How to specify the font size
  - font-size: 12pt;
  - font-size: 150%;
  - font-size: 1.5em; /* same as 150% */
- A font-family rule in the body element that is inherited by all descendants

```
body {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 100; }
```

- A font-family rule in a descendant that overrides the inherited font family

```
p { font-family: "Times New Roman", Times, serif; }
```

Other properties for styling fonts

- You can set the font-style, font-weight, and font-variant properties to a value of "normal" to remove any formatting that has been applied to theses properties.
- The line-height property determines the spacing between lines within a block element.
- If you specify just a number for the line-height property, the font size is multiplied by that value to determine the line height, and the multiplier is inherited by child elements. If you specific an absolute or relative size for the line-height property, the actual line height is inherited by child elements.
- You can use the shorthand property for a font to set all six font properties with single rule. When you use this property, the font-size and font-family properties are required.
- **font-style** - A keyword that determines how the fonts is slanted: normal, italic, and oblique.
- **font-weight** - A keyword or number that determines the boldness of the font: normal, bold, bolder, lighter, or multiples of 100 from 100 through 900, with 400 equivalent to normal. Bolder and lighter are relative to the parent element.

- **font-variant** - A keyword that specifies whether small caps will be used: normal and small-caps.
- **line-height** - A relative or absolute value or a number that specifies the amount or vertical space for each line. The excess space is divided equally above and below the font.
- How to specify font styles and variants
  - font-style: italic;
  - font-style: normal; /* remove style */
  - font-variant: small-caps
- How to specify font weights
  - font-weight: 700;
  - font-weight: bold; /* same as 700 */
  - font-weight: normal; /* same as 400 */
  - font-weight: lighter; /* relative to the parent element */

- How to specify line height

  line-heigh: 14pt;

  line-heigh: 140%;

  line-heigh: 1.4em; /* same as 140% */

  line-heigh: 1.4; /* same as 140% and 1.4em */
- The syntax for the shorthand font property

  font:[style] [weight] [variant] size[/line-height] family;
- How to use the shorthand font property

  font: italic bold 14px/19px Arial, sans-serif;

  font: small-caps 150% "Times New Roman", Times, serif;

  font: 90%/120% "Comic Sans MS", Impact, sans-serif;

How to set properties for formatting text

- **text-indent** - A relative or absolute value that determines the indentation for the first line of text. This property is inherited.
- **text-align** - A keyword that determines the horizontal alignment of text. Possible values are left, center, right, and justify. This property is inherited.
- **vertical-align** - A relative or absolute value or a keyword that determines the vertical alignment of text. Possible keywords are baseline, bottom, middle, top, text-bottom, text-top, sub, and super.
- **text-decoration** - A keyword that determines special decorations that are applied to text. Possible values underline, overline, line-through, and none.
- The **text-indent** and **text-align** properties are often used with text, and the **vertical-align** property is often used with tables.
- The **text-decorat**ion property is often set to "none" to remove the underlines from links.

- **The HTML for a web page**

```
<header>
    <h1>San Joaquin Valley Town Hall</h1>
</header>
<main>
    <p>Welcome to San Joaquin Valley Town Hall. We have some fascinating speakers for you this season!</p>
</main>
<footer>
    <p>&copy; Copyright 2020 San Joaquin Valley Town Hall.</p>
</footer>
```

- **CSS that specifies a text indent and horizontal alignment**

```
body {
        font-size: 100%;
        margin: 2em; }
h1 { font-size: 180% }
        main p { text-align: 2em; }
footer p {
        font-size: 80%;
        text-align: right; }
```

How to use CSS3 to add shadows to text

- Positive values offset the shadow to the right or down. Negative values offset the shadow to the left or up.
- The blur radius determines how much the shadow is blurred.
- The text-shadow property is supported by all modern browsers.
- If this property is not supported by a browser, it is ignored so there is no shadow, which is usually fine.
- The syntax of the text-shadow property
- text-shadow: *horizontalOffset, verticalOffset, blurRadius, shadowColor*;
  - The h1 element

```
<h1>San Joaquin Valley Town Hall</h1>
```

  - The CSS

```
h1 {                    /* color: blue; text-shadow: -2px -2px 4px red; */
    color: #ef9c00;
    text-shadow: 4px 4px; }
```

How to float an image so text flows around it

- The HTML

```
<img src="images/logo.png" alt="Town Hall Logo" width="80">
<h1>San Joaquin Valley Town Hall</h1>
<h2>Bringing cutting-edge speakers to the valley</h2>
```

- The CSS

```
img {
    float: left;
    margin-right: 1em; }
```

- The property that will stop the floating before a subsequent element

```
main { clear: left; }
```

- To float an image, you use the float property, and to set the margins around it, you use the margin property.
- When you float an image to the left, the block elements that follow it fill the space to the right of it. When the element that follow get past the height of the images and its top and bottom margins, they flow it not the space below the element.
- You can use the clear property to stop an element from flowing into the space alongside a floated element.
- For now, you can experiment with the size of the image to get the effect that you want, but in the next coming lessons you will learn the right ways to get the same results.