Web Site Design and Development

Lecture 5

CS 0134
Fall 2018
Tues and Thurs
1:00 – 2:15PM

CSS

- CSS stands for Cascading Style Sheets.
- These style sheets define the look and layout of your HTML elements.
- A CSS file is made up of a series of style rules that contain a selector and set of declarations.

```
h1 {
    color: blue;
    text-decoration: underline;
}
p {}
```

CSS selectors

- **Type – This is the name of the html element. This is written as is, like the h1 from our previous example.**
- **ID – This is the id assigned to an element with the id attribute. This is written as '#' followed by the id.**
- **Class – This is a class value assigned to an element with the class attribute. This is written as '.' followed by the class.**
- *** - This is the universal selector. This allows you to select all elements.**

Example CSS selectors

```
<article id="thesis">
  <h1>Study of the affects of pizza on the retention of information during exam preparation</h1>
  <p class="drop-cap article-body">
    Lorem ipsum...
  </p>
  <p class="conclusion article-body">
    Lorem ipsum...
  </p>
</article>
```

- *
- article
- h1
- p

- #thesis
- .drop-cap
- .article-body
- .conclusion

Example CSS selectors

```
<header>
  <img src="img/logo.png" alt="Company Co Logo" id="logo" class="left">
  <h2 class="right">Company Co</h2>
  <nav id="top-links">
    <a href="about.html" class="active">About</a>
    <a href="contact.html">Contact</a>
  </nav>
</header>
```

- *
- header
- img
- h2
- nav
- a

- #logo
- #top-links
- .left
- .right
- .active

Relational selectors

- **Relational selectors are selectors that are based on the relationship between two elements.**
- **You use the terms child, sibling and descendant in the same way that you would use them for a family tree.**

Types of relational selectors

- **Descendant – This selects an element if and only if that element is a descendant of another element. This is written as “ancestor descendant”.**
- **Adjacent sibling – This selects an element that is adjacent to another element. This is written as “element one+element two”.**
- **Child – This selects an element that is a child of another element. This is written as “parent>child”.**
- **General sibling – This selects any element that is a sibling of another element. This is written as “element one~element two”.**

Example relational CSS selectors

```
<header>
  <img src="img/logo.png" alt="Company Co Logo" id="logo" class="left">
  <h2 class="right">Company Co</h2>
  <nav id="top-links">
    <a href="about.html" class="active">About</a>
    <a href="contact.html">Contact</a>
  </nav>
</header>
```

- **Descendant**
  - header img
  - header a
- **Adjacent sibling**
  - img+h2
  - h2+nav
- **Child**
  - header>nav
  - nav>a
- **General Sibling**
  - img~nav
  - a~a

Combinations of selectors

- **You can combine type selectors with class and id selectors.**
  - For example, if you want all paragraphs of class article-body, you can combine those selectors as p.article-body.
- **You can also code multiple selectors in one style rule by listing each selector separated by a comma.**
  - For example, if you want a style rule that affects all heading elements, you can write

```
h1, h2, h3, h4, h5, h6 {
    font-weight: normal;
}
```

Example combination CSS selectors

```
<header>
  <img src="img/logo.png" alt="Company Co Logo" id="logo" class="left">
  <h2 class="right">Company Co</h2>
  <nav id="top-links">
    <a href="about.html" class="active">About</a>
    <a href="contact.html">Contact</a>
  </nav>
</header>
```

- img#logo
- h2.right
- img, a
- .left, .right

Attribute selectors

- **You can select elements based on the attributes that element has as well as the value of those attributes.**
  - For example
    - *[type] or [type] will select all elements with the type attribute
    - a[href] will select all <a> elements that have the href attribute
    - input[type=”text”] will select all input elements that have the type attribute with the value of text

Example attribute CSS selectors

```
<header>
  <img src="img/logo.png" alt="Company Co Logo" id="logo" class="left">
  <h2 class="right">Company Co</h2>
  <nav id="top-links">
    <a href="about.html" class="active">About</a>
    <a href="contact.html">Contact</a>
  </nav>
</header>
```

- *[src]
- [src]
- img[alt]
- a[href="about.html"]

Psuedo-classes

- **A pseudo-class is is a prefined “class” that meets a certain condition.**
  - Common pseudo-classes include:
    - :link – a link that has not been visted
    - :visited – a link that has been visited
    - :active – a link that is actively being clicked on
    - :hover – an element that the mouse cursor is hovering over
    - :focus – an element that has focus
    - :first-child – the first child of an element
    - :last-child – the last child of an element
    - :only-child – matches if an element only has one child

Examples of selectors with psuedo-classes

- a:link { text-decoration: none; }
- a:hover, a:focus { text-decoration: overline underline; }
- input:focus { border-color: red; }
- For accessibility, you should always apply the same formatting to :hover and :focus for an element. In this way, no matter how the user accesses that element, they see the same styling.

Pseudo-elements

- **A pseudo-element lets you select a portion of text.**
  - Two common pseudo-elements are
    - ::first-letter – the first letter of an element
    - ::first-line – the first line of an element
- **Notice that the pseudo-element starts with '::'. This started in CSS3. Prior to CSS3, you would just use ':'. You will need to use ':' if you need your web page to work in version of Internet Explorer prior to 9.**

Example of pseudo-element

```
.drop-cap::first-letter {
    float:left;
    font-size: 1.5em;
}

p::first-line {
    margin-left: 2em;
}
```

Questions?

CSS Units of measurement

- **For many CSS properties, you will need to set a size for the property**
- **There are two types of measurements**
  - Absolute: the size of 1 unit of this measurement is always the same size
  - Relative: the size of 1 unit of this measurement is relative to some other size

Absolute measurements

- **px: this is pixels, this is the size of one dot on a monitor**
- **pt: this is points, a point is 1/72 of an inch**
- **Examples**
  - font-size: 16pt;
  - width: 960px;

Relative measurements

- **em: this is ems, one em is equal to the font size of the current font**
- **rem: this is rems, one rem is equal to the font size of the root element**
- **%: this is percent, this will give you a size that is a percentage of the current value.**
- **Examples**
  - font-size: 150%;
  - margin-left: 2em;

Colors

- **There are 4 ways to specify a color**
  - Color name
  - RGB (red, green, blue)
  - Hexadecimal RGB
  - HSL (hue, saturation, lightness)
- **For RGB and HSL, you can also add a fourth value that specifies the opacity of the color.**
- **For accessibility, place dark text on a light background as this is easier to see than light text on a dark background.**

Color name

- There are 16 color names supported by all browsers: black, silver, white, aqua, gray, fuchsia, red, lime, green, maroon, blue, navy, yellow, olive, purple and teal
- To use color name, you simply type the name as the property value, for example, color: red.

RGB

- RGB is defined as the amount of red, green and blue that makes up a color.
- You can define that amount either using percentages or a value from 0 to 255. For example, rgb(20%, 60%, 40%) and rgb(51, 153, 102) are the same color with the former using percents and the latter using 0-255.
- You will typically find this color by using a color chart or color picker.

Hexadecimal RGB

- **Hexadecimal RGB (often referred to as Hex) is like RGB except it is represented as a '#' followed by 3 hexadecimal numbers put together. For example, #000000 for black.**
- **The hexadecimal numbers represent the amount of red, green and blue that make up the color, in that order.**
- **These numbers range from 00 for 0% of a color to FF for 100% of a color.**
- **Like RGB, you will typically find this value by using a color chart or picker**
- **This is the most common way to specify a color**

HSL

- **HSL is defined as the hue-degrees, saturation% and lightness% of a color**
  - hue-degrees: this is the color represented in degrees from 0 to 359
  - saturation%: this is how saturated a color is from 0% to 100%
  - Lightness%: this is the lightness of the color from 0% to 100%. 0% is black, 100% is white and 50% is normal
- **An example of HSL is hsl(300, 50%, 50%)**
- **Like RGB and Hex, you will typically find this value by using a color chart or picker**

Color opacity

- **To specify color opacity in RBG and HSL, you have rgba and hsla respectively.**
- **Opacity is specified as a number between 0 and 1 with 0 being completely transparent and 1 being completely opaque. For example, 20% opacity would be represented as 0.2.**
- **Examples**
  - rgba(0, 122, 230, 0.4)
  - hsla(240, 40%, 80%, 0.9)

Questions?

The cascading part of a cascading style sheet

- We can have multiple rules match the same element, so there has to be some way of determining which rule will be applied.
- The cascade, with regards to a cascading style sheet, refers to the priority in which rules are applied to elements.
- You can influence this priority by adding '! important' to a CSS declaration.
- Users can also set their own styles.

The cascade order

- **If more than one selector matches and element, the styles are applied in the following order, least priority to most priority**
  - Default styling supplied by the browser
  - Declarations from a user style sheet
  - Declarations from the web page
  - !important declarations from the web page
  - !important declarations from a user style sheet

What happens if multiple selectors of the same priority match?

- **If multiple selectors match at the same priority level, one of two things will happen**
  - If one of the selectors is more specific, e.g. h1.title vs .title, the more specific rule is used.
    - Specificity from most specific to least
      - The id for an element
      - Class, attribute selector or pseudo-class
      - Element or pseudo-element
  - If both selectors are the same level of specificity, the last rule that matches will be applied

Questions?

Adding styles to your web page

- **There are three ways to add styles to your web page**
  - Adding a style attribute to your html element
  - Using the style element in the head section of your HTML file
  - Using an external style sheet. This is the preferred way as it lets us keep our HTML and styling separate.

In the style attribute

- You can add CSS declarations directly to a element using the style attribute, this is called an inline style.
- This way of adding styles is hard to keep track of and may cause unexpected results if you forget that you styled an element this way.

Code

```
<h1 style=”text-decoration:underline”>
   Super Important Heading
</h1>
```

The style element

- **The style element belongs in the head section of your website.**
- **Styling this way is better than with inline styles but you will have to copy the style element and styles to every HTML file you need to style.**

Code

```
<style>
    h1 {
        text-decoration: underline
    }
</style>
```

External style sheet

- **You can place all of your style rules in a <filname>.css file on your website and refer to them from all your HTML files using the link element.**
- **This is the best way as it keeps your content and formatting separate and makes it easy to apply consistent style on all your web pages.**
- **If you have multiple link elements pointing in the same HTML file, the files are processed one at a time and when rules in later files refer to elements styled in earlier files, the later files rules are used.**

Example with external style sheet

HTML File

```
<html lang=”en”>
    <head>
        <title>Some Title</title>
        <link rel=”stylesheet”
href=”style.css”>
    </head>
    <body>
        <h1>Some Title</h1>
    </body>
</html>
```

CSS File

```
h1 {
    text-decoration:underline;
}
```