HTML5 Authoring with Mark Lassoff

Section 4: Getting Started with CSS Text Rules

> Note: Framework Television offers a full certification course in CSS. The included material with in the HMTL5 Authoring program is designed to give you enough CSS knowledge to start working with the language, but is not designed to be comprehensive.

CSS (Cascading Style Sheet language) provides the design layer to web based, and much mobile content. It is a mark up language that allows you, as the developer, to set rules governing how content should appear on the users' screens.

This has become more challenging of late because we no longer live in a world where screen size, resolution, bandwidth and other factors are predictable. People viewing your content may range from someone in an office with a fiber internet connection and a giant screen, to someone in subsaharan Africa with a cell phone, slow connection and tiny screen.

Fortunately CSS has both the power and the flexibility to produce styles that look great across the range of devices in use on today's internet.

> Understanding the "mechanicals" of CSS are just one aspect of producing digital content. You also have to understand the rules of digital design and usability. Designing interfaces that are usable and look good is a separate, related field to web development known as UX (User eXperience). You my find UX is a good direction to move in if you want to spend more time creating designs than writing code. You won't escape the coding entirely, however, as today, almost all UX designers have a firm grip of HTML5 and CSS.

After completing this section you will:

- ☐ Understand the basic CSS syntax.
- ☐ Use External and Internal Style Sheets
- ☐ Understand Fundamental Typographical CSS Rules
- ☐ Demonstrate CSS Override

Understand Basic CSS Syntax

CSS uses a different syntax than HTML. It is, however, designed to work easily with HTML.

CSS statements essentially have two parts: The selector and the style rule. The selector determines which elements to apply the CSS to and the style rule determines which styles to apply. Here's a typical block of CSS code:

```
p       {
                color: #121212;
                font-size: 14pt;
        }
```

The selector in this case is `p`. This means that any `<p>` tags would be selected and styled. (There are numerous other types of selectors that we'll get in to later in the course).

The style rules should be somewhat understandable just from reading them. In the case above, we're applying two style rules to the `<p>` tags. The first rule applies a deep gray color to the text. The second sets the size of the text to 14 points. Note that each CSS style rule is terminated by a semicolon.

Let's take a look at a second example:

```
h3      {
        text-decoration: underline;
        }
```

This rule would apply an underline to any h3 tags appearing in the document.

Use Inline, Internal and External Style Sheets

> The terms "CSS" and stylesheet are being used interchangeably throughout this section.

As is true with many aspects of digital development, there are several methods you can use to apply CSS code. Let's look at the three primary methods individually:

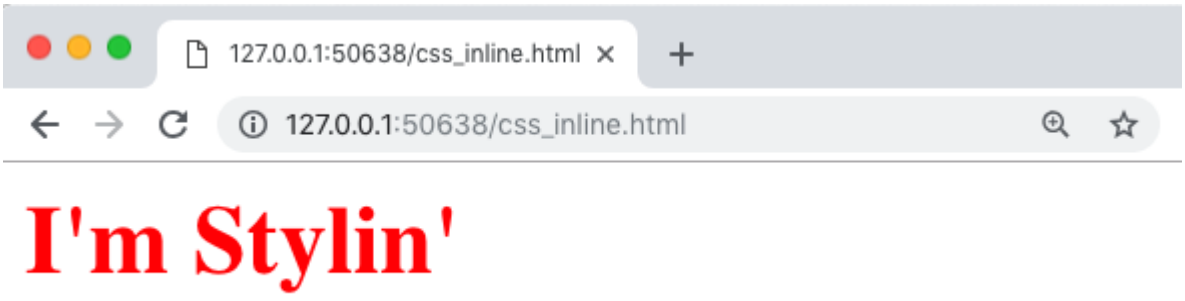
Inline CSS

Inline CSS is applied via the HTML `<style>` tag. This method is commonly used with HTML emails because an external style sheet is not possible.

Internal CSS looks like this:

```
<h1 style="color: red; font-size: 24pt;">I'm Stylin'</h1>
```

The `<h1>` element here would appear red in a 24 pt size. The result would look like this:



Outside of use with HTML emails, the use of inline styles is not considered a best practice. Well formed CSS should be able to be easily:

- ☐ Edited within a document. With inline CSS each individual instance of CSS must be located and edited.
- ☐ Applied to multiple elements on multiple pages lending to style consistency across a site.

Internal Style Sheets

Internal stylesheets appear in the `<head>` section of an HTML document. The CSS itself will appear within a `<style>` element. This is advantageous because all the CSS is in one location and easily editable. Perhaps, more importantly, CSS style rules can be applied to multiple elements, unlike in the case of inline CSS. Here's an example of an internal style sheet.

```
<head>
<title>Example Internal CSS</title>
<style>
        h1 {
                font-size: 28pt;
                font-family: Arial;
                font-weight: bold;
                color: #121212;
                }
        h2  {
                font-family: 20px;
                font-family: Arial;
                font-weight: normal;
                }
</style>
</head>
```

In this brief style sheet we've created style rules that apply to `<h1>` and `<h2>` tags within our document.

External Style Sheets

External style sheets appear in their own file and are attached to the HTML document using the `<link>` tag.

External Style Sheets are often the preferred method of placing CSS because the document can be applied to multiple HTML files giving related documents consistent styling and allowing style changes across multiple pages to be completed with a single edit.

An external style sheet should be saved with the `.css` file extension. Here's a brief external style sheet saved as `myStyles.css`. **Note that in the case of external stylesheets no style tags are used.**

```
p       {
        font-family: Arial;
        font-size: 10pt;
        }

strong {
        color: red;
        }

em      {
        color: blue;
        }
```

You should be able to tell from context what the style rules do in this case. To attach the `myStyles.css` file to the HTML document the following line should appear in the `<head>` section of the HTML document:

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

There are a couple of things to note here about the `<link>` tag. First the `href` attribute should point to the path and filename for your CSS. In the case above the CSS appears in the same folder as the HTML. The `rel` attribute and the `type` attribute always appear as shown here and are required.

Like the `<br/>` tag, the `<link />` tag is an empty tag and uses the `/` character at the end of the tag to self-close.

Understand Fundamental Typographical CSS Rules

You've already seen a number of fundamental CSS rules applied. These typographical rules are all fairly straight forward. They are all applied in the manner we've demonstrated here.

Here are the most important CSS text properties. (The term property and the term rule are interchangeable).

Rule	
color	Sets the color of text in rgb, hex or color-name format
letter-spacing	Sets space between characters in text
line-height	Sets the height of an individual line.
text-align	Right, Center, Left Align or Justify your text
text-decoration	Sets underline, or overline on text
text-indent	Specifies indent on first line of text
text-shadow	Specifies a drop shadow on text
text-transform	Controls the Capitalization of text
vertical-align	Set the vertical alignment of your text
font-family	Set the typeface for text
font-size	Sets the font-size of text
font-style	Sets the style in which font is presented. "Normal", "Italic" and "Oblique" are the options.
font-variant	Allows use of small caps (Where lower case letters are represented by smaller upper-case letters)
font-weight	Sets the boldness of text.

Some of these rules, like `font-size` require a measurement. CSS is very flexible when it comes to measuring type. Below is a chart of the most common types measurements that are available.

Unit	Description	Example
%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	`p {font-size: 16pt; line-height: 125%;}`
cm	Defines a measurement in centimeters.	`div {margin-bottom: 2cm;}`
em	A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.	`p {letter-spacing: 7em;}`
in	Defines measurements in inches	`p {word-spacing: .15in;}`
mm	Defines measurements in millimeters	`p {word-spacing: 15mm;}`
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	`body {font-size: 18pt;}`
px	Defines a measurement in screen pixels.	`p {padding: 25px;}`

Despite the flexibility afforded by CSS, most developers stick to `px`, `pt`, and `em`. Because it's a relative measurement that respects the users preferences `em` is becoming the most accepted type measurement unit.

Debug This: Semantic HTML

You know what to do here. We've got two separate files here.

`westport_debug.html`

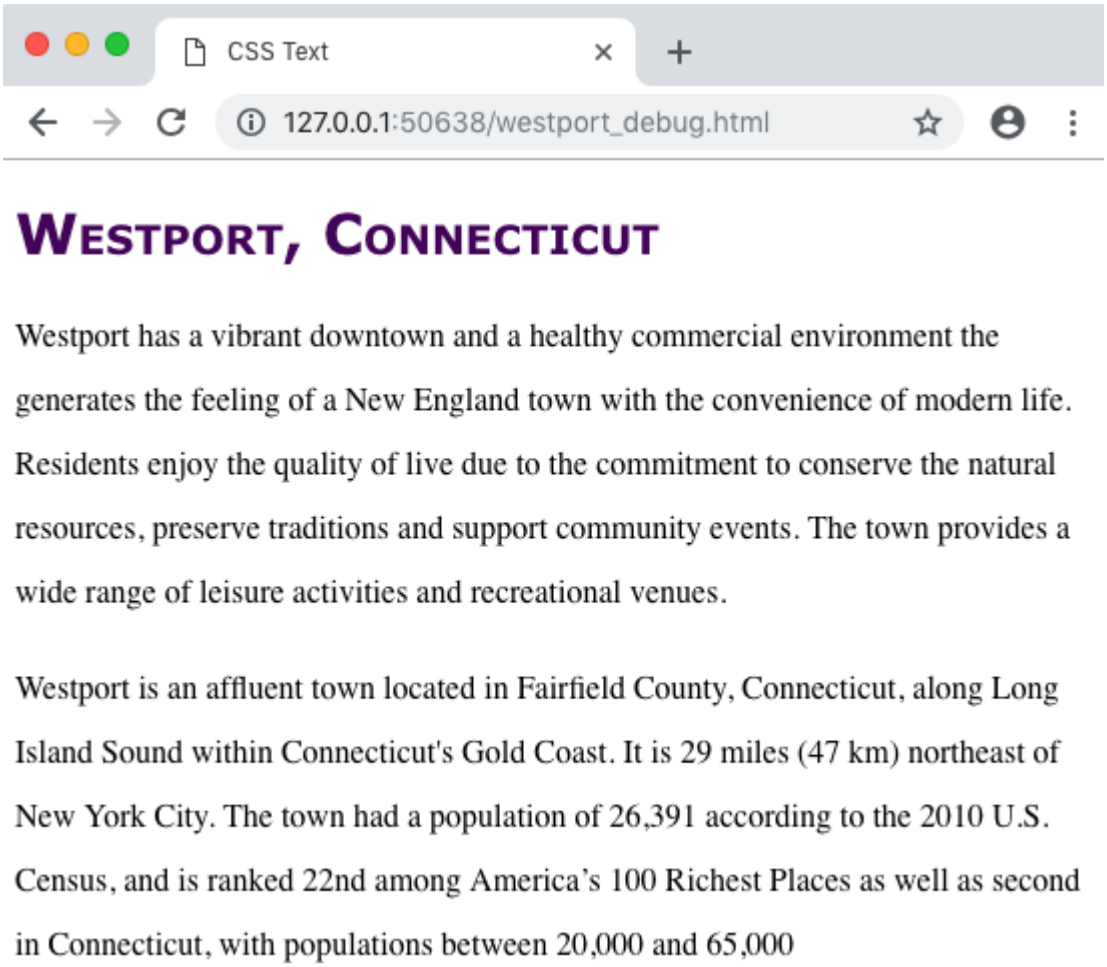
```
<!DOCTYPE html>
<html>
    <head>
        <title>CSS Text</title>
        <link href="westportcss" relationship="StyleSheet" />
    </head>
    <body>
        <h1>Westport, Connecticut</h1>
        <p>Westport has a vibrant downtown and a healthy
         commercial environment the generates the feeling of
         a New England town with the convenience of modern life.
         Residents enjoy the quality of live due to the commitment
```

```
          to conserve the natural resources, preserve traditions and
          support community events. The town provides a wide range
          of leisure activities and recreational venues. </p>

         <p>Westport is an affluent town located in Fairfield County,
          Connecticut, along Long Island Sound within Connecticut's
           Gold Coast. It is 29 miles (47 km) northeast of New
           York City. The town had a population of 26,391 according
           to the 2010 U.S. Census, and is ranked 22nd among
           America’s 100 Richest Places as well as second in
           Connecticut, with populations between 20,000 and 65,000</p>
    </body>
</html>
```

westportcss.css

```
hl
{
    font-family: Verdana
    font-size: 1.75mem
    font-weight: bold
    text-decoration: none
    font-variant: mall-caps
    color: rgb(75,0,100)
}

p
{
    line-height: 2em
    text-align: left
}
```

With errors corrected your file should appear like this:



Submit This:

Find or write a paragraph or two about your hometown or about a location that you like. (Wikipedia is a good source for this-- Make sure you give appropriate credit in your document!) Create correctly styled HTML5 and use appropriate markup

Using some of the CSS properties we've discussed in this section, create a pleasing and readable design for your document. Use an external CSS file to create your typography. (There is no correct or incorrect for this lab as long as your CSS correctly styles your HTML document!)

Have some fun with this!

Remember, when submitting the work please use the following naming convention for your file: `HTMLAUTHORING_LastName_SectionNumber.html`. So if your last name is Smith and your submitting section 8, you file name should be `HTMLAUTHORING_Smith_8.html`.

For this course visit https://www.dropbox.com/request/RhW9kBDXtisq2Fsvg3hY to submit your assignments.