2

INTRODUCTION TO WEB DESIGN

Module 2 Styling With CSS

Instructor Tonya D. Wright

Introduction to CSS

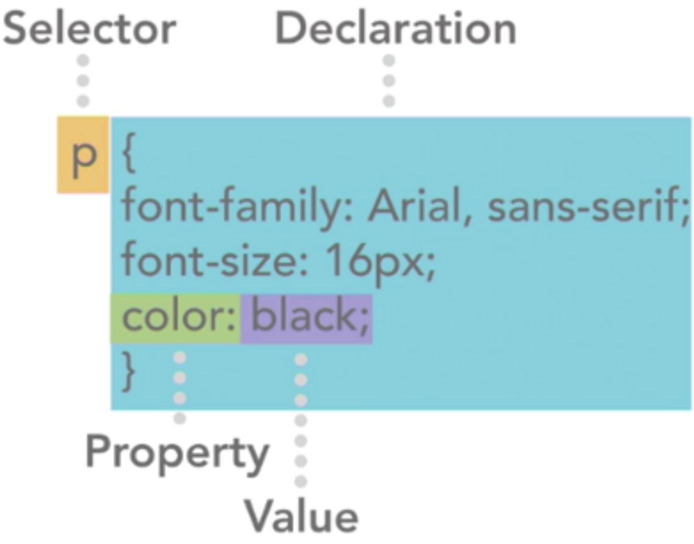
CSS stands for Cascading Stylesheets. CSS is a language that allows you to change the appearance of elements on the page: the size, style, and color of text; background colors; border styles and colors; even the position of elements on the page. HTML is for structure, while CSS is for presentation.

Since the very beginning, HTML tags came right out of the box with one or more default formats, characteristics, or behaviors. So even if you did nothing, much of your text would already be formatted in a certain way in most browsers. One of the essential tasks in mastering CSS is learning and understanding these defaults and how they may affect your content.

Common HTML Defaults

ITEM	DESCRIPTION
Background	In most browsers, the page background color is white. The background of the elements <div>, <table>, <td>, <th> and most other tags is transparent.
Headings	Headings <h1> through <h6> are bold and align to the left. The six heading tags apply differing font size attributes, with <h1> the largest and <h6> the smallest. Apparent sizes may vary between browsers. Headings and other text elements may also display additional spacing (margins) above or below.
Body text	Outside of a table cell, paragraphs—<p>, <li>, <dd>, <dt>—align to the left and start at the top of the page.
Table cell text	Text within table cells, <td>, aligns horizontally to the left and vertically to the center.
Table header	Text within header cells, <th>, aligns horizontally and vertically to the center (this is not standard across all browsers).
Fonts	Text color is black. Default typeface and font are specified and supplied by the browser, which in turn can be overridden by the user using the preference settings in the browser itself.
Margins	Spacing external to the element border/boundary is handled by margins. Many HTML elements feature some form of margin spacing. Margins are often used to insert additional space between paragraphs and to indent text, as in lists and blockquotes.
Padding	Spacing within the box border is handled by padding. According to the default HTML 4 style sheet, no elements feature default padding.

The basics of CSS



```
p {
font-family: Arial, sans-serif;
font-size: 16px;
color: black;
}
```

The basics of CSS are pretty simple. Stylesheets are made up of rules that control the styling and positioning of HTML elements. Each rule is made up of selectors which are used to target HTML elements and declarations, which contain properties and values that the browser then uses to style the element.

CSS styles are made up of two parts: the selector and the declaration. The selector, here **p** for paragraph, tells the browser which element or elements in the document to style.

Using this selector, all paragraphs within a style document would be formatted. While this element selector is pretty simple, selectors can be very complex depending upon which elements on the page you're trying to target. Selectors can be grouped together or even combined to allow for more precise element targeting. The declaration, which is enclosed in these curly braces, uses the formatting instructions for the style.

The declaration in this example has several rules. The rules themselves are made up of two parts: the property and the value. The property and value are separated by a colon and use a semicolon at the end to tell the browser to stop evaluating and move on to the next rule.

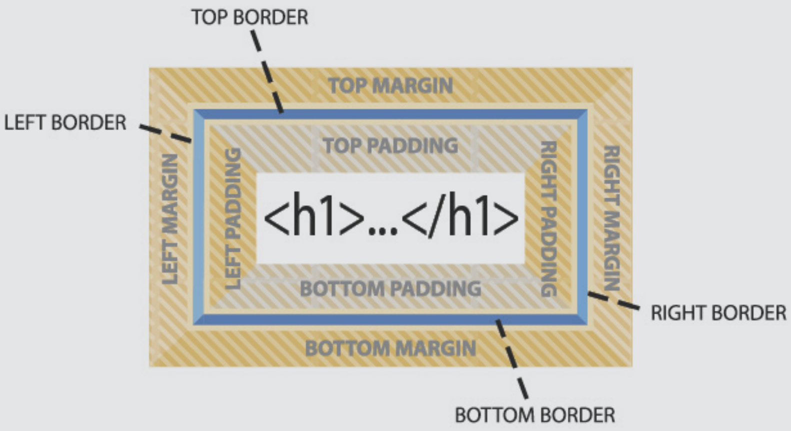
While this syntax is simple, you will need to learn the myriad selectors and selector combinations which make it possible to exercise a greater amount of control over page elements. You'll also need to learn the various properties you can set for each of the elements and the values that are allowed for that property.

Although there are some additional syntax rules to learn, for the most part that's as complicated as CSS gets.

CSS Box Model

Browsers normally read the HTML code, interpret its structure and formatting, and then display the webpage. CSS does its work by stepping between HTML and the browser, redefining how each element should be rendered. It imposes an imaginary box around each element and then enables you to format almost every aspect of how that box and its contents are displayed.

CSS permits you to specify fonts, line spacing, colors, borders, background shading and graphics, margins, and padding, among other things. Most of the time, these boxes are invisible, and although CSS gives you the ability to format them, it doesn't require you to do so.



you will see this box model often.

Methods of adding CSS to HTML

There is more than one way to add a Cascading Stylesheet (CSS) to your HTML document. We will cover the advantages and disadvantages of the 4 main methods.

1. Linking to a separate CSS file:

is the most common method of attaching CSS rules to an HTML document. With this method all of your style rules are contained in a single text file that is saved with the .css extension. This file is saved on your server and you link to it directly from each HTML file.

The link is just a simple line of HTML that you put in the <head> section of your HTML document, it looks like this:

```
<link rel="stylesheet" type="text/css" href="css/nameoffile.css" media="screen" />
```

Make sure you include the correct path to your CSS file in the href.

If the CSS file is in the same folder as your HTML file then no path is required (1) but if it's saved in a folder (2):

(1) href="/nameoffile.css" **OR** **(2)** href="css/nameoffile.css"

We will using (2).

The media parameter specifies when the CSS rules are to be used. "screen" indicates for use on the computer screen. If you specify "print" then the rules will only be followed when the page is printed. You can include multiple CSS files if required.

Advantages - If you need to make a style change across your whole website then you only need to make the change once in your single CSS file. If you want to completely change the look of your website, again, you only need to update this one file.

CSS Zen Garden - A demonstration of what can be accomplished through CSS-based design.

Perhaps the best reason to have a separate CSS file is for speed. When a person first visits your website their browser downloads the HTML of the current page plus the linked CSS file. Then when they navigate to another page their browser only needs to download the HTML of that page, the CSS file is cached so does not need to be downloaded again.

>>>>>This can significantly **decrease** browsing speeds on a website.

2. Embedding CSS into the HTML

You can also embed CSS rules directly into any HTML page. To do this you need to add the following code to the <head> of your HTML document. All of your CSS rules go between the style tags. As before, the media can be "screen" for your computer screen or "print" for printing.

```
<style media="screen" type="text/css">

Add style rules here

</style>
```

Disadvantage with this approach the styles must be downloaded every time someone visits the page, this can cause a slightly slower browsing experience. If you need to make an adjustment you must visit every page of your website.

Advantage - Because the CSS is part of the HTML document, the whole page exists as just one file. This can be useful if you are sending the page to someone via email or if it will be used as a template. Another advantage of using this method is with dynamic content. If you are using a database to generate the page content you can also generate dynamic styles at the same time.

3. Adding Inline CSS to HTML tags

Style rules can also be added directly to any HTML element. To do this you simply add a style parameter to the element and enter your style rules as the value. Here is an example of a heading with red text and a black background:

```
<h2 style="color:red;background:black;">This is a red heading with a black background</h2>
```

This is not a good method to use because it will bloat your HTML and make website maintenance a pain. However it can be useful in some situations. One example could be if you are using a system where you don't have access to the CSS file - simply add your style rules directly to the elements instead.

4. Importing a CSS file from within CSS

Another interesting way to add CSS to a HTML page is with the import rule. This lets us attach a new CSS file from within CSS itself.

Even the most complex style sheet starts out with a single rule. But when you're working on a particularly massive and complex website, over time your style sheet will inevitably start to reflect the site's size and complexity. And even if you employ every trick for organizing your CSS in the book, you might find that the sheer size of the file is simply overwhelming. At that point, you might want to consider splitting your style sheet up into several smaller CSS files. That's when the @import rule can come in quite handy.

To import a new CSS file from within CSS simply use the following rule:

```
<style>
        @import url('/css/styles.css');
</style>
```

Or this more cleaner technique:

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

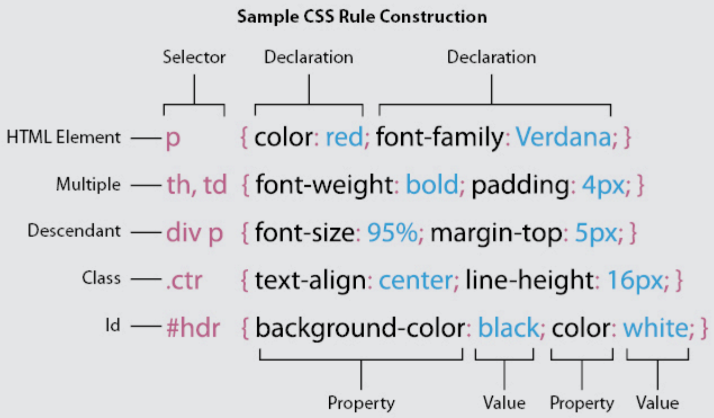
an extra tip:
Your @imports must come before all other rules in your CSS. So be sure to do your imports before you do anything else.
We will be doing this in our code, so you can see how it works.

Applying CSS styling

A CSS formatting instruction is known as a rule. A rule consists of two parts”a selector and one or more declarations. The selector specifies what element, or combination of elements, is to be formatted; declarations contain the styling information. CSS rules can redefine any existing HTML element, as well as define two custom element modifiers, named “class” and “id.”

A rule can also combine selectors to target multiple elements at once or to target specific instances within a page where elements appear in unique ways, such as when one element is nested within another.

These sample rules demonstrate some typical constructions used in selectors and declarations. The way the selector is written determines how the styling is applied and how the rules interact with one another.


Sample CSS Rule Construction
Selector
Declaration
Declaration
HTML Element — p { color: red; font-family: Verdana; }
Multiple — th, td { font-weight: bold; padding: 4px; }
Descendant — div p { font-size: 95%; margin-top: 5px; }
Class — .ctr { text-align: center; line-height: 16px; }
Id — #hdr { background-color: black; color: white; }
Property
Value
Property
Value

Applying a CSS rule is not a simple matter of selecting some text and applying a paragraph or character style, as in Adobe InDesign or Adobe Illustrator. CSS rules can affect single words, paragraphs of text, or combinations of text and objects. A single rule can affect an entire webpage, a single paragraph, or just a few words or letters. Basically, anything that has an HTML tag on it can be styled, and there is even an HTML tag specifically intended to style content that has no tag.

CSS rule syntax: VERY IMPORTANT

CSS is a powerful adjunct to HTML. It has the power to style and format any HTML element, but the language is sensitive to even the smallest typo or syntax error. Miss a period, comma, or semicolon and you may as well have left the code out of your page entirely. Even worse, an error in one rule may cancel all the styling in subsequent rules or the entire style sheet.

For example, take the following simple rule:

```
p { padding: 1px;
    margin: 10px; }
```

EXAMPLE WE WILL USE - *Note there will be times when spacing is necessary, you will see.*

It applies both padding and margins to the paragraph <p> element.

This rule can also be written properly without spacing as:

```
p{padding:1px;margin:10px;}
```

The spaces and line breaks used in the first example are unnecessary, merely accommodations for the humans who may write and read the code. Removing excess spacing is known as minification and is often used to optimize stylesheets. Browsers and other applications processing the code do not need this extra space, but the same cannot be said of the various punctuation marks sprinkled throughout the CSS. - *Note there will be times when the spacing is important, you will see.*

FOR THIS COURSE, WE WILL FOLLOW THE FIRST EXAMPLE. MAKES IT EASIER FOR BOTH OF US

Moving forward with CSS:

- Learn as much as you can about selectors. There are multiple types of selectors, and knowing the rules behind them and how they work is critical to writing efficient CSS. Without the ability to write selectors, you won't be able to target specific page elements or regions.
- Learn which properties can be set for each type of element and which values are accepted for specific properties.
- Also, learn the **three** core rendering principles of CSS - **Specificity, Inheritance, and the Cascade Theory**. These principles control when rules are applied, the order in which they're applied, and how any styling conflicts between rules should be resolved.
- Another great way to learn syntax is to examine the CSS found in other sites. Often the authors will comment the CSS in a way that helps you understand the syntax and why it's written a certain way. We talk about viewing source code on the next page.

one more tip:

There are some differences between how browsers render specific properties; it's really important that you have a general idea of what is supported and how rendering might differ from browser to browser. Occasionally, you can create fallbacks or workarounds for browsers, or simply design around the browsers' differences.

So download the top leading browsers and it helps to have access to Mac and PC so you can view the differences all around.

VALIDATING HTML AND CSS

After we have coded HTML and CSS, we can go online and validate our code to ensure we are meeting compliance standards. This is also a quick way to see if we have any errors.

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, MobileOK content, or to find broken links, there are other validators and tools available.

HTML Validator
https://validator.w3.org/

CSS Validator
https://jigsaw.w3.org/css-validator/

OTHER VALIDATOR AND TOOLS
https://w3c.github.io/developers/tools/

In Case You Were Wondering

CSS: Adding Icons to Your Web page

In our final video for this unit we added in a script to pull in a minified CSS stylesheet for icons.

Icons make the content on your website more visual. Generally, icons are used in buttons alongside text, in navigation menus, or alongside some important information. Earlier, whenever someone had to use an icon on their webpage, they used to first download the icon, and then add an img tag with the icon's path as src to show the icon.

Not anymore, thanks to some amazing Icon Fonts like Font Amazing, Google Material Icons and Bootstrap Icons, we can directly use the icons without downloading or installing anything.

Using Icon Font libraries, through CDN's we can adjust the size, color, shadow of the icon, just like we do for any other text on our webpage.

WHAT IS A CDN?
content delivery network
A content delivery network (CDN) is a system of distributed servers (network) that deliver pages and other web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server.

FONT AWESOME

https://fontawesome.com/
This is the website I used to support our buttons with icons. I also have a free account here to use icons in other projects.

To use Font awesome library into our webpage, we need to add the following code inside the <head></head> tag.

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

The above line of code, includes the CSS file for Font Awesome from the CDN server, and now we can add icons as follows:

```
<i class="fa fa-cab">TEXT CAN GO HERE</i>
<i class="fa fa-refresh fa-spin">TEXT CAN GO HERE</i>
<i class="fa fa-home">TEXT CAN GO HERE</i>
```

MODULE 2 RESOURCES

Checks browser for CSS compatibility
http://css3test.com

https://jigsaw.w3.org/css-validator/

http://www.cssdesk.com/

http://csslayout.news/

https://css-tricks.com/

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors

https://www.smashingmagazine.com/mastering-css-principles-comprehensive-reference-guide/

https://www.w3.org/TR/css-cascade-3/