

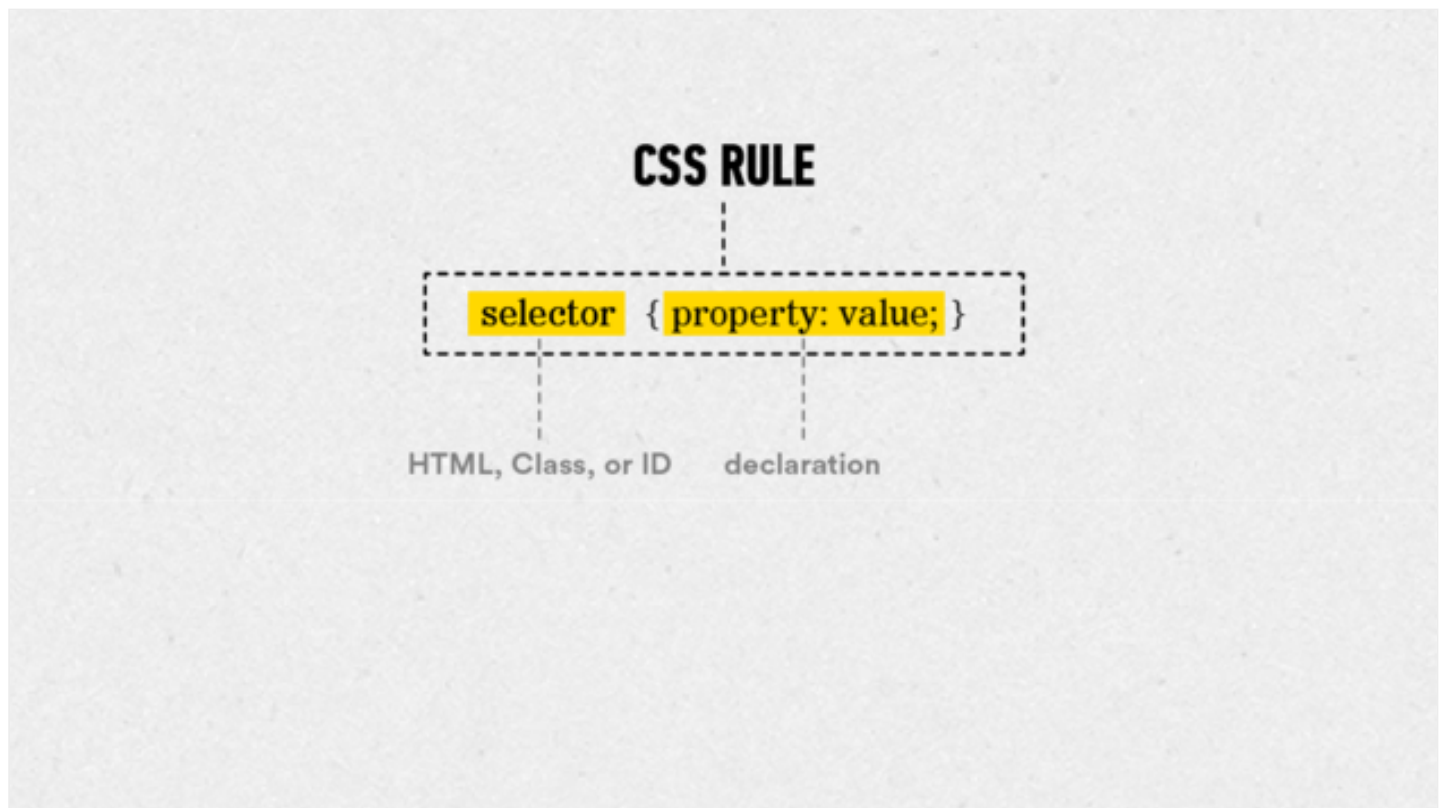


Review Guide: Principles of HTML & CSS

Let's review some of the key ideas introduced in this unit. You can also download all this content for future reference by clicking on the attached file.

Key Definitions

- **Rule:** The building block of a CSS stylesheet. A rule consists of a selector and a declaration block (one or more declarations).
- **Declaration:** A declaration is made up of a property and a value, separated by a colon and punctuated by a semi-colon."
- **Selector:** The actual HTML object the declaration(s) apply to.
- **Property:** The characteristic of the selector that will be changed.
- **Value:** The amount or type of change to be applied to the corresponding property of the matched selector.



- **Marking-Up:** The process of assigning HTML tags to given text content in order to indicate its relation to the rest of the text or dictate how it should be displayed.
- **Serif Font:** One of two general categories of fonts (typefaces) that uses marks (called “serifs”) to embellish characters. A common serif font is “Times New Roman”.
- **Sans-Serif Font:** One of two general categories of fonts that have cleaner line due to not using marks to embellish characters. (Sans Serif literally meaning “Without serif”) A common sans-serif font is “Helvetica”.
- **Class:** A class attribute is added to an HTML element in order to give you a “hook” to refer to that element in your CSS. CSS class selectors begin with a “.”. Classes can be used multiple times per page.
- **ID:** An id attribute is added to an HTML element in order to give you a “hook” to refer to that element in your CSS. CSS id selectors begin with a “#”. IDs can only be used one time per page.
- **Horizontal Rule:** Add a horizontal line across your page using the hr tag.
- **Line Break:** Break up a block of text using the br tag.
- **Image:** Add images to your HTML using the img tag. Tell the browser the source of the image file with an src attribute.
- **Absolute File Path:** A path to a website or file that includes a full web address (starting with “http”) that the browser loads from the remote location directly. For example:

```

```

- **Relative File Path:** A path to a website or file that gives you the path to the resource you are looking for as it relates to your website's local file structure. For example, if you wanted to retrieve an image called *newlogo.png* that resides in a file called *img*, you would enter the following relative address:

```

```

Why Separate HTML from CSS?

Separating HTML from CSS offers you scalability and versatility. If you separate how your site looks from what your site says, things become more flexible.

By separating your HTML and CSS, you can make the change in one place and have it apply to your whole site. If you separate what your site says from how it looks, you can apply any number of different styles to the same content.

CSS Color Treatment

While color names are fine when you're just beginning, there's a number of reasons you'll want to switch over to something more advanced.

First, color names are rendered differently by different browsers. Secondly, there are only 147 color names accepted as standard, meaning your options are going to be pretty limited.

Instead, you'll want to use either RGB or hexadecimal codes. Both of these are built on a system of entering values for the colors red, green, and blue.

By mixing different intensities of these three colors, you can create millions of different colors and shades. Intensity values range from 0 (no intensity) to 255 (full intensity) in the RGB system.

In hex, they range from 0-9, then continue from A-F, with two characters each for red, green, and blue. This is clearer with examples, so here are some common colors with their RGB and Hex equivalents.

The format for color names, RGB, and hexadecimal should look like the following, respectively:

```
p {  
    color: red;  
}  
  
p {  
    color: (255,0,0);  
}  
  
p {  
    color: #FF0000;  
}
```

CSS Text Treatment

▪ font-family

To adjust the font of your selected text element, use the font-family property. For the value, enter the name of the font to which you'd like to alter your text.

To be safe, try putting a comma after your selected font and enter a generic family as a fallback. If the web browser doesn't support the font you selected, it will choose the fallback.

```
h1 {  
    font-family: Arial, sans-serif;  
}
```

If your selected font is more than one word, capitalize both words and put them in quotation marks.

```
h1 {  
    font-family: "Courier New", monospace;  
}
```

▪ font-size

To increase or decrease the spacial dimensions of your chosen text, use the font-size property. As a beginner, you'll want to enter pixel values for your font-size values.

```
h3 {  
    font-size: 24px;  
}
```

As you become more advanced, try using percentages or ems instead of pixels. These can be pretty tricky, though, so you may want to wait until we cover them in Unit 9.

▪ font-weight

To adjust the thickness of your selected text, use the font-weight property.

As a beginner, you can enter values like “normal” to make your text thin and “bold” to make your text thick. As these values aren’t very specific, different browsers may interpret their display with slightly different outputs.

```
h1 {  
    font-weight: normal;  
}  
  
h1 {  
    font-weight: bold;  
}
```

As you become more advanced, try using the numbers 100, 200, 300, 400, 500, 600, 700, 800, and 900 as values to gain more granular control. With this system, 400 is roughly equivalent to “normal” and 700 roughly equals “bold.”

```
h1 {  
    font-weight: 400;  
}  
  
h1 {  
    font-weight: 700;  
}
```

▪ font-style

To make normal text italic, use the property font-style and the value “italic.” To reverse this effect, use the value “normal.”

```
a {  
    font-style: italic;  
}  
  
a {  
    font-style: normal;  
}
```

▪ text-align

To adjust the positioning of a text element, use the CSS property `text-align` and one of the following values: `left`, `right`, `center`, or `justify`.

```
body {  
  text-align: center;  
}
```

▪ text-decoration

To add an underline to normal text, use the CSS property `text-decoration` and the value `underline`.

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

To remove underlines, use the value `none`. This declaration is often applied to anchor tags.

```
a {  
  text-decoration: none;  
}
```

Less commonly used values include `overline` to add a line above text and `line-through` which strikes a line through your text.

▪ text-transform









To adjust capitalization in a selected text element, use the `text-transform` property.

Values for this property include `uppercase` to make every letter capitalized, `lowercase` to make every letter uncapitalized, and `capitalize` to make the first letter of every word in the selected text uppercase.

```
h1 {  
  text-transform: uppercase;  
}
```

Sublime Text Shortcuts

- **Quick Save:** `⌘` + `S` (for Mac) or `CTRL` + `S` (PC)
- **Instant Boilerplate:** Type `<html>` + `TAB` (Mac and PC)
- **Instant Lorem Ipsum:** Type `<Lorem>` + `TAB` (Mac and PC)
- **Add Comment:** `⌘` + `/` (for Mac) or `CTRL` + `/` (for PC)

- **Undo:**  +  (for Mac) or  +  (for PC)
- **Redo:**  +  (for Mac) or  +  (for PC)

Classes vs. IDs

Classes and IDs, also called selectors, are ways of targeting the style of specific HTML elements on your page.

So what's the difference between them?

In short, classes allow you to style many elements with a particular style, while IDs are only capable of styling a single element.

Using these selectors allows you flexibility and control of styling individual, as well as groups, of elements on your page.

HTML and Images

When you'd like to add images to your site, you should use the tag with the "src" attribute. "src" stands for "Source" and works just like the tag's "href" attribute. It tells the image tag where to find the image you'd want to include on your page.

Relative vs. Absolute Addressing

Relative addressing basically tells your browser: "Look in our project folder for the file specified. Or, if a folder is specified, look within that folder for the file specified."

On the other hand, when we link to a source outside of our project, it is referred to as an absolute address.

Sublime Text Shortcuts

Bookmark [this guide](#) or download it for review.

Questions to Ask Your Mentor

1.
Why is it important to create separate HTML and CSS documents? Can you explain separation of concerns in greater detail?
- 2.

How do I get started with creating an external style sheet? What elements should I define the style for first?

3.

Where can I find examples of external style sheets to use as templates?

4.

What are some best practices for organizing my files and folders on my computer? Should I be saving my work elsewhere?

5.

Which fonts should I use for my project?

Further Reading

How the RGB Color System Works

http://en.wikipedia.org/wiki/RGB_color_model

<http://www.rgbworld.com/color.html>

More on Colors

http://www.w3schools.com/cssref/css_colornames.asp

<http://www.color-hex.com/>

Adobe Color CC (formerly known as Adobe Kuler)

<https://color.adobe.com/create/color-wheel/>

More on Fonts

http://www.w3schools.com/css/css_font.asp

http://www.w3schools.com/cssref/css_websafe_fonts.asp

Difference Between Serif and Sans-Serif

<http://www.urbanfonts.com/blog/2013/02/serif-vs-sans-the-final-battle>

Classes and IDs

<http://css-tricks.com/the-difference-between-id-and-class/>

File Paths

<http://css-tricks.com/quick-reminder-about-file-paths/>

Absolute vs. Relative Paths

<http://www.coffeecup.com/help/articles/absolute-vs-relative-pathlinks>

http://www.boogiejack.com/server_paths.html

DOWNLOADS

Summary

© 2017 General Assembly. All rights reserved. [Regulatory Information](#) [Terms](#) [Privacy](#) [Assembled in NYC](#)