

# Week 6: Introduction to Cascading Style Sheets (CSS)

## Agenda

- Overview
  - Project Proposal
  - Previous Participation Questions/Responses
  - New Resources
  - What's Due?
- Introduction to Cascading Style Sheets (Lecture/Class Activity)
  - What is CSS?
  - How can we add styles to a page?
  - CSS Examples
    - Web Developer Tools (Chrome)
- First CSS Exercise (CW/HW)
- Participation Questions (4)
- Next Week

## Overview

Last week, we reviewed validation and why it's important for our HTML documents. We also covered search engine optimization and general usability/accessibility best practices. This week, we will learn about Cascading Style Sheets (CSS) and how they can be used to add styles to our pages. By the end of this class, students will be able to link CSS to their HTML files and apply basic styles to elements on the page.

## New Resources

I've made a PowerPoint for CSS available on Blackboard. It's located in Blackboard → Lecture Notes → Additional Resources and it's titled "Intro to CSS." Besides this PowerPoint, there are two HTML files with associated CSS that were released with today's lecture notes. Feel free to review the content/code for additional help with the material we cover today.

## Project Proposal

The first part of the final project (Project Proposal) is **due tonight at midnight**. You will need to submit 3 wireframes, 1 style tile, and 1 needs assessment for the website you plan to build for your final project. If you aren't sure about your topic or the quality of your proposal, please reach out to me.

Overall, you need to make sure you are explaining your work as much as possible. For the wireframes that means using headings/labels to make your potential layout more understandable. For the style tile, you should include colors, fonts, adjectives and anything else that's visual to convey a look/feel. The needs assessment will be used to explain your ideas, goals, etc. for the website.

## Previous Participation Questions/Responses

These are the participation questions from last week:

<https://forms.gle/W6dBh1R4NCAfwN7T8>

This link has the responses to the fun question from last week. Feel free to look over the responses from myself and your peers.

<https://docs.google.com/spreadsheets/d/1QLoQzbO-Er7ARIECl0aPvY1lKUyZTdYeuwsn4g0XEAK/edit?usp=sharing>

### What's Due?

- Participation Questions (4)
- First CSS Exercise (CW/HW)
- Project Proposal (tonight at midnight)

## Introduction to Cascading Style Sheets (Lecture/Class Activity)

### What is CSS?

CSS stands for Cascading Style Sheets. Unlike HTML, which is used to represent the structure of a web page, CSS is the language for determining the presentation of a page in terms of colors, fonts, positioning, etc. We use CSS to select specific HTML elements to give them styles.

After selecting specific elements (either directly or based on certain criteria), we can then choose whatever properties we'd like and give them different values. An example would be targeting paragraph tags and giving them a red font color. We can do that with the code below:

```
p {  
  color: red;  
}
```

“p” is the selector (all paragraph tags), “color” is the property, and “red” is the value we are assigning to the property. The selector comes first followed by curly brackets “{}”. The curly brackets represent a block of CSS declarations. Inside of the curly brackets is where we choose our properties and values, and each set is a CSS declaration. The properties and values are always separated by a colon and the whole statement (color: red) is ended with a semicolon.

It is **best practice** to separate your HTML and CSS by using an **externally placed style sheet**. The reason for this is because it is easier to update your code if it is in a different location. If you have a consistent banner style on many pages of your website and use inline/internal styling, you would have to go into the code for each page and edit it. If you use external styling, you can edit the code in one place and the changes would persist across all pages.

Just like HTML and other languages, we can add comments to our files in addition to regular code. In CSS, commenting is simple and requires only slashes and asterisks. To add a comment, use the syntax below:

```
/* This is a comment */
```

Anything between the asterisks is the comment content and won't affect styles being applied. Comments are useful if you want to keep a style, but prevent it from affecting the page. They're

also helpful for general documentation purposes and I recommend using them to distinguish different sections of your page.

### **How can we add styles to a page?**

There are three main ways for implementing CSS on a web page and each one is explained below with a link to an example. To see the code in the example, right-click the page and select “View Source” or “Inspect Element” and look around the window that appears.

#### Inline CSS

This is typed directly into the HTML tag as part of the style attribute. The line below will do the same thing as the CSS demonstrated above. However, it will only apply to the HTML tag where it resides.

```
<p style="color: red">This is a red paragraph.</p>
```

#### Internal CSS

This is CSS typed between <style></style> tags which are located inside of the <head></head> tags of your HTML document. I’ve used this in some of the assignments this semester to give our tables styles.

In the CSS declaration below, I select the table, tr, th, and td tags to give them all a solid black border that is 1px in width. I then collapse the border, so it appears as one line and then add a little padding, so the text is easier to read.

```
<style>
    table, tr, th, td {
        border: 1px solid black;
        border-collapse: collapse;
        padding: 5px;
    }
</style>
```

#### External CSS – BEST PRACTICE

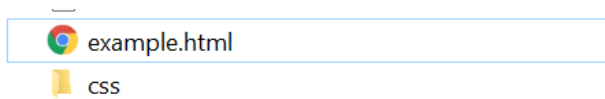
This is CSS placed in a separate document with a “.css” extension. It is connected to our HTML through the usage of a <link> tag which is placed inside of the <head></head> tags. The link tag uses two attributes which are rel and href.

The rel is required to explain the relationship between our resource and the HTML. Here, we are saying it’s a style sheet. The href section is where you link your CSS style sheet. Like working with local images, you need to include folder/file names accurately to link them correctly. In the example below, my CSS file is located in a css folder which I refer to before the file itself.

```
<head>
    <link rel="stylesheet" href="css/style.css">
```

</head>

View of my folder structure (style.css is inside of the css folder):



Since I use “css/style.css” my CSS file should be inside of that folder. If I wanted to have my HTML and CSS files be in the same folder, the link tag would be changed so that the href had style.css as shown below.

```
<link rel="stylesheet" href="style.css">
```

View of my folder structure (they’re blue because I selected both items in my folder):



A lot of students tend to put both files in the same folder. I would recommend keeping them separated like they are in the first screenshot. This will keep your files organized and is helpful for assignments/the final project. When you move your files to FileZilla, the folder organization there should be the same as it is on your local machine. **If your CSS doesn’t work at all with your HTML, it’s probably because the href value is incorrect.**

## CSS Examples

Open a text editor and follow along as I demonstrate how we can use CSS on a page. I will go over the general syntax and show you how to apply styles and check them with web developer tools in the browser. For my demo, I will be using Brackets and Google Chrome.

Online Resources (Not required reading, only meant to help you if you need it)

- <https://www.htmldog.com/guides/css/> - CSS Tutorials for all levels
- [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps) - Excellent resource with guides/tutorials for CSS
- [https://www.tutorialspoint.com/css/css\\_syntax.htm](https://www.tutorialspoint.com/css/css_syntax.htm) - Specific syntax examples
- <https://css-tricks.com/> - Cool website for code snippets and articles

## First CSS Exercise (CW/HW)

**There are two parts to this assignment.** For the first part, you will add CSS styles to a web page so that it matches a provided image as closely as possible. In part two, you will open an existing web page and inspect it. After inspecting the page, you will alter the styles on the page and take a screenshot of the changes.

### Part 1

Download the “First-CSS-Exercise.zip” folder from Blackboard under today’s Lecture Notes or the Assignment folder. Extract everything from inside of this zip folder to some place where you

will be able to find them easily. There will be an HTML file, a css folder with a style.css file in it, and an image with an example of a finished assignment.

Based on the instructions below, add CSS to the style.css file to alter the way the page looks. The image I've provided is an example of how the assignment should look after you are done with it. Your page should look exactly like mine if you follow my instructions listed below.

To get started, extract all the files from the Zip folder on Blackboard or get them from the link mentioned above. Open the HTML/CSS files inside of a text editor and add code as necessary. All CSS should go inside of the style.css file. You don't need to include any tags, only CSS should be inside of that document.

For all assignments/project work in this course, your CSS must be **externally placed**. This is because it will be easier to manipulate styles on your page if the styles are separated. I would recommend leaving the ".css" file inside of the folder provided. Otherwise, you will have to change the href value in your link tag. The instructions for the styles I'd like to see are listed below.

#### Instructions for Adding CSS

- Give the body a font-family of verdana, a font size of 16px, and a background of black
- Give the h1 a background of black, a color of white, and a font size of 2em
- Give the paragraph right after the h1 a color of white ("+" selector" is good if you want to use it)
- Give each div a background color of "#ccc"
- Each h2 should be centered aligned and have a font size of 1.4em
- Each paragraph should have a line height of 1.3
- Content inside of Div with the class of "firstDiv"
  - The h2 and span tags should have a color of blue
  - The span tags should have a white background color and a 1px solid blue border
  - The unordered list should have its list-style-type property changed to square
  - The unordered list items should have the odd list items appear with a blue background and white text (nth-child selector on li tag – 1<sup>st</sup>, 3<sup>rd</sup>, 5<sup>th</sup> list items targeted)
- Content inside of Div with the class of "secondDiv"
  - The h2 and span tags should have a color of green
  - The span tags should have a white background color and a 1px dashed green border
  - The table, th, and td tags should have a black background, white color for the text, and a solid white 1px border. The th and td tags should have 4px of padding added to them.

### **Extra Credit (+.5pt on your final grade)**

Complete the assignment without using any classes in your CSS file. This means you would only use element names and no classes or ids in your CSS selectors. You will need to utilize various pseudo-classes such as nth-child, nth-of-type, etc. in your CSS selectors to do this. This is a bit tougher and requires that you be creative with your selector choices. There are also many routes to getting it done, feel free to use whatever selectors you need.

### **Part 2**

Open a web page of your choosing and inspect it as shown earlier in class. To inspect a page, you can right-click the page and select “Inspect” or you can use F12 on a PC. If you’re in Safari, you might have to activate the developer tools options in your preferences.

After you’ve opened a web page and inspected it, a panel should appear which displays HTML and CSS. Your task for this part is to alter the CSS for various elements on the page so that it is significantly different than the original version. There is no right or wrong process here, your goal is simply to play with the styles you see and change the page. If you see a style you don’t understand, look it up and try something new.

The main thing I want to see is that you’ve experimented with different CSS property/value combinations and have documented the changes. After you’re done making your changes, take a screenshot of the new version of the page and attach it with your submission. You can write a list to document your changes instead of in a paragraph.

### **Tips**

- Change the values for properties (feel free to try the ones listed here)
  - Font-size
  - Font-family
  - Color
  - Background-color
  - Border
  - Margin
  - Padding
  - Text-align
  - Text-decoration
  - Line-height
- Uncheck some boxes for styles
- Target large items on the page for more visible changes
- Use the selector tool (box with an arrow – furthest left icon) to pick items to change
- Don’t refresh the page or your changes will be lost

Your web page and screenshot are **due October 7<sup>th</sup> at 6pm**. To submit the work for this exercise, go to Blackboard → Course Materials → Lecture Notes for this week’s class. You can also visit the “Assignments” folder and the submission area will be titled “First CSS Exercise”

**You must submit a live link to your web page and one image.** The work submitted will be evaluated based on the rubric explained below.

### **First CSS Exercise – 10pts**

- Part 1: 8pts
  - Live link was submitted: 1pt
  - CSS is externally placed: 1pt
  - CSS is organized/nested neatly: 1pt
  - All required styles are included: 5pts
- Part 2: 2pts
  - Numerous changes were made: .5pt
  - Explanation of changes/website name included: .5pt
  - Screenshot of changes included: 1pt

## Participation Questions 9/30

To receive credit for participation in today's class, please answer all the questions in the Google Form linked below. If you don't want to answer the final question (fun), simply put N/A (not applicable) for your answer and that will count.

### **This Week's Participation Questions:**

<https://forms.gle/rMaLpfDhz1gT7UgJ7>

### **Previous Participation Questions/Responses**

These are the participation questions from last week:

<https://forms.gle/W6dBh1R4NCAfwN7T8>

This link has the responses to the fun question from last week. Feel free to look over the responses from myself and your peers.

<https://docs.google.com/spreadsheets/d/1QLoQzbO-Er7ARIECl0aPvY1lKUyZTdYeuwsn4g0XEak/edit?usp=sharing>

## Next Week

Next week we will look at more complicated CSS properties that will allow us to manipulate the positioning of the content on our page. This is one of the tougher areas of web development because manipulating positioning on a web page can be difficult based on your content and layout needs. I would recommend reading up on the CSS Box Model and Floats at the very least. Below are some links to get you started. You don't have to read through them all but sifting through some of the material will help you for next week.

1. <https://www.htmldog.com/guides/css/intermediate/layout/>
2. <https://css-tricks.com/all-about-floats/>
3. [https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/The\\_box\\_model](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model)
4. <https://css-tricks.com/the-css-box-model/>