# Quick Guide – Cascading Style Sheets (CSS)

Now that you know html, it’s time to add some style! to your website. Style involves adding colors, fonts, size of text, etc. to your content.

You can add styling inline to the html tags themselves - i.e.

<h1 style="color:red;margin-left:30px;">This is a heading</h1>

However, this can become onerous when you have multiple webpages and you want them to all have the same look and feel. To give that consistent look to a website, all h1 tags should include this style code. A cascading style sheet (CSS) is an easy way to apply the same look and feel – i.e., style, across your website. Rather than coding colors, fonts, etc. on each page and to each type of tag, a .CSS file can be defined once, and then applied throughout your website.

The name cascading style sheets comes from the idea that the web browser will apply the style in a cascade using the following priorities:

1. Inline styling (such as the h1 tag above
2. .css sheet
3. Browser default

Styling can have a dramatic effect on how a webpage looks. Check out these examples from W3Schools:

<https://www.w3schools.com/css/css_intro.asp>

## Creating a .CSS file

Similar to an .HTML file, a .CSS file is a text file that indicates what style should be applied to an HTML tag – or more specifically, applied to the content within that tag.

You can create a CSS file in notepad or in your IDE – **just make sure your file’s extension is .css**

## General Format

The general format of a CSS command is:

selector { property:value; property:value; . . . property:value; }

* selector – generally refers to the HTML element (i.e., tag) you want the style to impact.
* { } – inside the curly brackets is the declaration block; each declaration is separated by a semi-colon
* property – the style attribute/property type
* value – the property value

## Examples

Example 1 – set all H1 (header 1) text to red with a font size of 22pt

h1 {color: red; font-size: 22pt;}

Example 2 – all paragraphs should use navy font and be bolded

p {color: navy; font-weight: bold;}

Example 3 – the background color of the page should be lightyellow, and the text in 14pt navy

Body { background: lightyellow;

color: navy;

font-size: 14pt;

}

See w3schools for a list of colors: <http://www.w3schools.com/colors/colors_names.asp>

## Adding a .CSS file to your website

In order for a webpage to recognize your .CSS file, add the following code to the head section of each html file. It can go right after your <title> tag:

<head>

<title>page title</title>

**<link href="styles.css" rel="stylesheet"> 🡨 Add this line**

</head>

## Setting cool fonts

Google has a set of great fonts – have a look and select your favorite to include on your website:

<https://fonts.google.com/>

In order to use these fonts you must include a reference in your html file. You may also include the font in your .CSS file. For example, if you liked the Felipa font:

Include the following in your <head> tag, just before the link to your .css file:

<link href="https://fonts.googleapis.com/css?family=Felipa" rel="stylesheet">

You can reference the font in a .css file as follows:

body { background: aliceblue;

color: indigo;

font-size: 14pt;

font-family: 'Felipa';

}

## Other Stuff

### Size of fonts

You can set the font size based on “pt” (points), “px” (pixals), or as a percentage of the base font. The advantage of using percentages is if the base font size changes, the other fonts will change proportionally. Here is an example of styling H1 using a font size percentage:

H1 {font-size: 250%;

font-weight: bold;

color: red}

### Setting styles for hovering

You can set a style when the user hovers over content. For example, to change a link to a different color when it is hovered over, use the following CSS code:

a:hover { color: red }

## Focusing Styles

There will be instances where you don’t want a style to apply to every instance of an element. For example, you may **not** want to have all paragraphs (element <p> use the same style). You might have two *types* or *classification* of paragraphs – maybe something like school and personal. For example, you may want all paragraphs relating to school to be green text, and all paragraphs relating to personal items to be orange text.

There are a couple of ways to go about this using CSS. You can add ID# to elements in html, and then style that ID# in CSS. You can set up different classes in your html, and then style each class differently. Another easy way is to enclose the content you want styled a bit differently in your own tags.

#### EXAMPLE

In your html file:

<work>

<p> This is a paragraph about work. I’ve added my own work tag to designate it as work related. </p>

</work>

<me>

<p> This is a paragraph about me. I’ve added my own me tag to designate it as personal.</p>

</me>

Then in your CSS file:

work p { color: green }

me p { color: orange }

### Setting up navigation bars

You can set up snazzy navigation bars too!

In your html file add <nav> tags around your menu items:

<nav>

<a href="aboutme.html">About Me</a>

<a href="CV.pdf">CV</a>

<a href="https://www.linkedin.com/in/tanya-beaulieu-13131756">linkedin</a>

</nav>

In your css file, style the a elements within the nav tags:

nav a { display: inline-block;

border: 2px solid navy;

padding: 10px;

margin: 5px;

background: yellow;}

In this example, the display property indicates the a elements (i.e., links) should be displayed as an inline-block. Content can be displayed inline (does not start on a new line, it just follows the previous content on the same line), or block (starts on a new line and takes the entire width available. An inline-block is the best of both worlds in that we can apply only some styles to a block (i.e., borders), but the content (i.e., our links) will display one right after the other.

Border can be used on blocks to add a border around the block. In our example, we added a border that was 2 pixels wide, a solid line (as opposed to dashed, or other styles), and the border is navy color.

Padding specifies the amount of space between the border lines and the content inside. We specified 10 pixels of padding. Margin is the amount of space between blocks. In this example we said our a elements (links) should be 5 pixels apart.

Background lets you style the background of the block. I choose a color style to set the background to yellow.

## Adding Comments

It is a good idea when coding to use comments to document what your code does and/or leave notes to yourself. It is amazing what you will forget when you go back and look at your code in 6 months.

A CSS comment begins with /\* and ends with \*/ and can span multiple lines. The browser will ignore everything with the comment tags.

/\* This is a single line comment \*/ /\* This is a

a multiline

comment \*/