**Personal Reference Document**

**Styling in CSS – How to**

**Written By:** Philip Anderegg

**On:** Saturday, January 27th, 2024

Table of Contents

[CSS Styling 3](#_Toc157256350)

[Type Selectors 3](#_Toc157256351)

[Background color 4](#_Toc157256352)

[Margins 6](#_Toc157256353)

[Background Image 7](#_Toc157256354)

# CSS Styling

In order to style html elements in css, first we must have some html written. We can start with the following:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Cafe Menu</title>

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

  </head>

At the very bottom of the html document, there is a </html> tag to close the document. We must designate the character set, and the view, setting it to the device’s width of the screen. We then designate a title called Café Menu and finally, we link our document to the CSS file via <link href=”styles.css” rel=”stylesheet” />

\*Note: meta and link are self-closing tags

\*\*In html we can use the <style> </style> and directly imbed the css styling within the html document; however, it is good practice to simply have a separate document called styles.css and take care of it there

Now add a body so that the title appears on screen:

<body>

        <main>

            <h1>CAMPER CAFE</h1>

            <p>Est. 2020</p>

            <section>

                <h2>Coffee</h2>

            </section>

        </main>

    </body>

</html>

Nice! Let’s move on to CSS

## Type Selectors

The syntax for CSS is

  element {

    property: value;

  }

Right now this is our webpage:

A white background with black text

Description automatically generated

If we want to center all the text, we can do it as follows in CSS:

  h1, h2, p {

    text-align: center;

  }

We can use multiple selectors and attribute the same styling. This gives us this:

A white background with black text

Description automatically generated

Great!

## Background color

We can set the background color by using the following:

element {

  background-color: color;

}

For the color, we can use strings such as red, blue, burlywood etc; we can use rgb values: rgb(0-255, 0-255, 0-255); rgba values (rgb with “a” element that defines transparency from 0.0 to 1.0). You can also use hsl (hue, saturation, light) and hsla, to define colors. We’ll do burlywood instead of brown for a café vibe:

  body {

    background-color: burlywood;

  }

Result:

A sign with text on it

Description automatically generated

Since we are making a menu, we should place everything in a div for design layout purposes, with id value “menu”:

        <div id="menu">

            <main>

                <h1>CAMPER CAFE</h1>

                <p>Est. 2020</p>

                <section>

                    <h2>Coffee</h2>

                </section>

            </main>

        </div>

In the CSS, we can target the menu id using the # symbol followed by the id. If we add this to CSS:

  #menu {

    width: 300px;

    background-color: burlywood;

  }

And comment out the background color in the body, we get this:

A close up of a sign

Description automatically generated

We can determine width using pixels or percentages:

#menu {

    width: 80%;

    background-color: burlywood;

  }

A sign with text on it

Description automatically generated

## Margins

Almost there, now let’s fix the margins:

  margin-left: auto;

  margin-right: auto;

doing this to the menu will center it. We will also change the menu id selector to a class selector using .class-name syntax.

In the html: <div class="menu">

In the CSS: .menu {…}

## Background Image

We can add a background image using a url, within the body selector:

background-image: url(https://cdn.freecodecamp.org/curriculum/css-cafe/beans.jpg);

This adds coffee beans to the page:

A pile of coffee beans

Description automatically generated

Now our café will have items and prices, we will use an article for that.

          <article>

            <p>French Vanilla</p>

            <p>3.00</p>

          </article>

Doing this results in the following:

A sign with text on it

Description automatically generated

We can add more things the same way, but we want them on the same line. What we’ll do is define the p element as a class called flavor and in the CSS, align the text to the left. We will also do something similar to the price, and add class=”price”, and align it to the right.

HTML:

<article>

            <p class="flavor">French Vanilla</p><p class="price">3.00</p>

          </article>

          <article>

            <p class="flavor">Caramel Macchiato</p><p class="price">3.75</p>

          </article>

          <article>

            <p class="flavor">Pumpkin Spice</p><p class="price">3.50</p>

          </article>

          <article>

            <p class="flavor">Hazelnut</p><p class="price">4.00</p>

          </article>

          <article>

            <p class="flavor">Mocha</p><p class="price">4.50</p>

          </article>

CSS:

  .flavor {

    text-align: left;

  }

  .price {

    text-align: right;

  }

Problem, they are still not side-by-side! \*\*Note bg image is still there

A sign with text on it

Description automatically generated

Add class=”item” to the article elements. Then we can modify the p elements within the item class directly:

## Multiple p elements in-line

.item p {

    display: inline-block;

}

A sign with text on it

Description automatically generated

\*\*Inline block only takes up its content width. To spread them, use text-align on the individual flavor and price classes and set the widths to 75%, 25%:

\*Note if you want both <p> elements on the same line, you must have them on the same line, as shown in the HTML above.

  .flavor {

    text-align: left;

    width: 75%;

  }

  .price {

    text-align: right;

    width: 25%;

  }

A sign with text on it

Description automatically generated

Now that we know it works for one, we can do it for the rest of the items

<article class="item">

            <p class="flavor">French Vanilla</p><p class="price">3.00</p>

          </article>

          <article class="item">

            <p class="flavor">Caramel Macchiato</p><p class="price">3.75</p>

          </article>

          <article class="item">

            <p class="flavor">Pumpkin Spice</p><p class="price">3.50</p>

          </article>

          <article class="item">

            <p class="flavor">Hazelnut</p><p class="price">4.00</p>

          </article>

          <article class="item">

            <p class="flavor">Mocha</p><p class="price">4.50</p>

          </article>

A menu of coffee beans

Description automatically generated

## Setting a Border

.frame {

  border: 50px solid black;

}

You can round the border using border-radius:

border-radius: 8px 10px 8px 10px;

The arguments are: top-left, top-right, bottom-right, bottom-left. IF you only put one argument, the border is the same for all corners.

An example:

border-radius: 30px 25px 60px 12px;

A red rectangular object with black border

Description automatically generated

We can rotate elements:

transform: rotate(-0.6deg);

A brown rectangular frame with a black border

Description automatically generated