

Week-2 Basic CSS - Recap

1. Introduction to CSS

CSS which stands for Cascading Styles Sheet, is for styling content written in HTML. With CSS we can change the color of a text, apply background color or image on an element, control the size, space and layout of elements.

2. How to add CSS styles on your webpage

We can apply CSS to our webpages in 3 main ways:

1. **Inline CSS** - the use of the style attribute on an element
2. **Embedded (Internal CSS)** - the use of the style tags on the head part of our HTML document.
3. **External CSS** - the use of a separate file that contain CSS codes which is then linked to our HTML document.

3. CSS Syntax

CSS styles are written in rules.

- A **CSS Rule** is a combination of of a *Selector*, and *Declarations* in a declaration block.
- A **Selector** is a way of selecting HTML element(s) that you want to style.
- A **Declaration** is a combination of a *property* and *value*, separated by a colon, and usually ends with a semicolon.
- These declaration(s) are put inside curly brackets called **Declaration Block**

Example of a CSS Syntax

```
body {  
  font-size: 16px;  
  background-color: gray;  
  width: 800px;  
}
```

3. CSS Selectors

CSS Selectors are ways of selecting HTML elements we want to style. We have learned four main types of selectors.

1. Tagname Selector - This is also called Element name selector, or Simple selector. This type of selector selects an element based on its tag name. Eg:

```
h2 {  
  
}
```

above will select all the *h2* elements on the page.

2. Class Selector - This selects an element based on its class name. Eg:

```
.blue-text {  
  
}
```

above will select all the elements on the page which has the class *blue-text*.

3. ID Selector - This selects an element based on its ID value. Elements are expected to have only one ID, and are meant to be unique on page as well. Eg:

```
#logo {  
  
}
```

above will select an element on the page which has the ID *logo*.

4. Attribute Value Selector - This selects an element based on its attribute value. Eg:

```
input[type='checkbox'] {  
  
}
```

above will select all **input** elements on the page which has the **type** attribute with a value **checkbox**.

4. Common CSS properties

We learned about CSS properties. CSS properties are keywords we normally used in our CSS declarations to apply particular styles to our elements. Remember a CSS declaration is a combination of a CSS *property* and *value*

1. **color** - to set the foreground color of an element
2. **background-color** - to set the background color an element
3. **font-size** - to set the font-size of an element
4. **font-family** - to set the font family of an element. There are non standard fonts we normally add to our webpages. Example of a custom font provider is **Google Fonts**
5. **width** - to se the width of an element. Remember that elements on our webpage are like rectangles. So you can set the width and height of an element.
6. **height** - to se the height of an element.
7. **border-width** - to set the width of the border of an element. Remember elements are like rectanglges, so they can have borders.
8. **border-style** - to set the style of the border of an element.
9. **border-color** - to set the color of the border of an element.
10. **border** - border is a shorthand form of *border-width*, *border-style*, and *border-color*.
11. **border-radius** - We can make rounded corners to an element by using the *border-radius* property.
12. **padding** - This refers to the inner space of an element. The space between the actual content and border of an element.
13. **margin** - This refers to the outer space of an element. The space between the border of an element and other surrounding elements.

5. Inheritance

Inheritance in CSS, is when styles are passed down from a parent to child elements. Eample

```
body {  
    color: blue;  
}  
  
h1 {  
    text-transform: uppercase;  
}
```

/*

Above, we set all text colors on the body to blue. This style will be

```
passed down to all children of the body. So even though we didn't set a
color for the h1, the h1 will inherit its color from the body
*/
```

6. Specificity (Priotizing Styles)

This is when the browser decides which CSS declaration to use when there are conflicting declarations. We learned that *Class selectors* are more specific than *Tagname selectors*, *ID selectors* are more specific than *Class Selectors*, and *inline-styles* are more specific than *ID Selectors*.

There is also this *!important* keyword which when used on a declaration, will overrule any other declarations.

7. Working with Colors

We learned that there are 3 main ways of specifying colors in CSS:

1. Color Names. Eg: red, blue, green
2. Hexadecimal (Hex) Color Codes. Eg: #0000FF (Blue hex code)
3. RGB (Red, Green, Blue) Color Codes. Eg: rgb(0, 255, 0) (Green hex code)

8. CSS Variables (CSS Custom Properties)

CSS custom properties allow us to use variables in CSS. A variable is a custom keyword/name that stores a value. In CSS, a variable is defined starting with two dashes (--) followed by the name, a colon and then the value. Example *--primary-color: blue*

9. Basic Compatibiilty issues with Browser Fallbacks

We learned that not all CSS features especially modern features are supported by all browsers. This is where compatibility comes in. For example, Internet Explorer does not support CSS variables. So for example if you are using a CSS variable to set the background color of an element, you may provide a fallback for internet explorer like below:

```
div {
    background-color: gray; /* this is the fallback for Internet
Explorer and other browsers that don't support variables */

    background-color: var(--background-color, 'gray') /* the second
value here 'gray' is used as a fallback for the --background-color
```

```
variable incase if it's not set */  
}
```

10. CSS Variables with Media Query

We also explored some other use cases of CSS variables by updating the value of a variable in media query, thereby changing the styles of the elements on which the variable is used.

By Omar Jasseh

Lead Instructor, JCC

I hope you enjoyed :) reading this material.

Feel free to shoot me an email to jassehomar99@gmail.com or WhatsApp message to 2203100948 if you found any grammatical or content-based error(s) on this document.

I do normally share coding tips and resources on a few social platforms. Follow me on:

[Facebook](#), [Linkedin](#), [Twitter](#), or [Github](#)

to learn from the resources I share with the world.