Introduction to Web Development

A sneak peek into CSS - part 1

CSS

THE STYLING PART

What is CSS

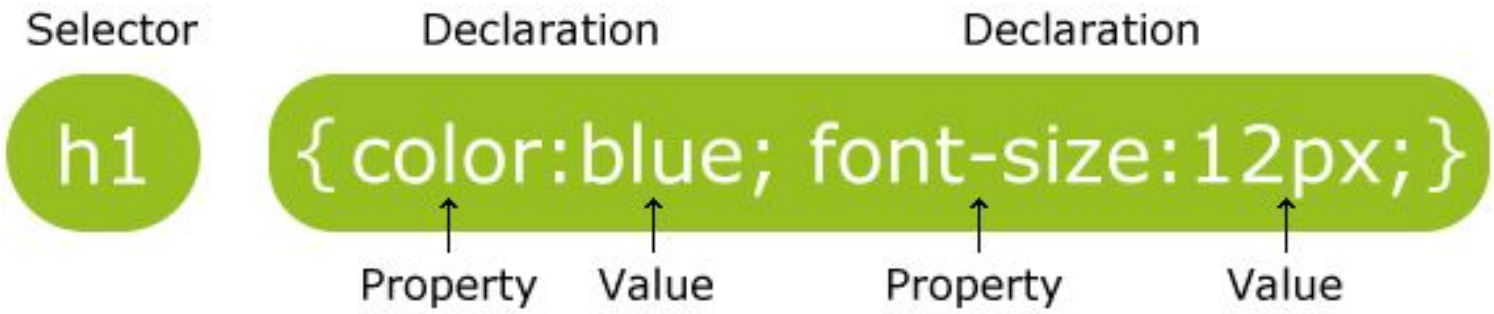
Codetrain

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External stylesheets are stored in **CSS files (** with the extension **‘.css’)**

CSS Syntax

Codetrain

A CSS rule-set consists of a selector and a declaration block:



CSS Syntax continued

Codetrain

- The **selector** points to the HTML element you want to style.
- The **declaration block** contains one or more declarations separated by **semicolons**.
- Each declaration includes a CSS **property name** and a **value**, separated by a **colon**.
- A CSS declaration always ends with a **semicolon**, and declaration blocks are surrounded by **curly braces**.

CSS Syntax continued

Codetrain

Example:

In the following example all **<p>** elements will be center-aligned, with a red text color:

```
p {
    color: red;
    text-align: center;
}
```

CSS Element Selector

Codetrain

The element selector selects elements based on the element name.

Example:
You can select all **<p>** elements on a page like this (in this case, all **<p>** elements will be center-aligned, with a red text color):

```
p {
    color: red;
    text-align: center;
}
```

The id Selector

Codetrain

- The **id** selector uses the id attribute of an HTML element to select a **specific element**.

- To select an element with a specific **id**, write a hash **(#)** character, followed by the id of the element.

```
#unique-item {
    color: red;
    text-align: center;
}
```

The class Selector

Codetrain

- The class selector selects elements with a specific class attribute.

- To select elements with a specific class, write a period (.) character, followed by the name of the class.

```
.related-items {
   color: red;
   text-align: center;
}
```

CSS Selectors - specificity

Codetrain

- You can also specify that only specific HTML elements should be affected by a class or id.

- In the example below, only **<p>** elements with **class="center"** will be center-aligned:

```
p.center {
    color: red;
    text-align: center;
}
```

Grouping Selectors

Codetrain

- If you have elements with the same style definitions, you can group them.

```
h1 {
    color: red;
    text-align: center;
}
p {
    color: red;
    text-align: center;
}
```

```
h1 , p {
    color: red;
    text-align: center;
}
```

CSS background properties

Codetrain

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

```
h1 {
    background-color: green;
}
```

```
body {
    background-image: url("nice_image.png");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
}

body {
    background: url("ugly_image.png") no-repeat right top fixed;
}
```

CSS border properties

Codetrain

The CSS border properties allow you to specify the style, width, and color of an element's border.

- Border-style e.g dotted, solid etc
- Border-width e.g 5px
- Border-color e.g red
- Border-radius e.g 5%

```
div {
    border: 5px solid grey;
    border-radius:5%;
}
```

CSS margins

Codetrain

The CSS margin properties are used to create space around elements, outside of any defined borders.

- margin-top
- margin-right
- margin-bottom
- margin-left

```
Html element {
    margin: top right bottom left;
}
```

```
div {
    margin: 25px 10px 30px 15px;
}
```

Ask me what **margin: auto** does!

CSS paddings

Codetrain

The CSS padding properties are used to create space around elements, inside of any defined borders.

- padding-top
- padding-right
- padding-bottom
- padding-left

```
Html element {
    padding: top right bottom left;
}
```

```
div {
    padding: 25px 10px 30px 15px;
}
```

CSS Height and Width

Codetrain

The height and width properties are used to set the height and width of an element.

```
div {
    width: 50%;
    height: 100vh;
    background-color: purple;
}
```

Three ways to insert CSS

Codetrain

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

Codetrain

- With an **external style sheet**, you can change the look of an entire website by changing just one file!

- Each page must include a **reference** to the external style sheet file inside the **<link>** element. The **<link>** element goes inside the **<head>** section:

```
<head>
   <link rel=”stylesheet” type=”text/css” href=”styles.css” />
</head>
```

Internal Style Sheet

Codetrain

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the **<style>** element, inside the **<head>** section of an HTML page:

```
<head>
   <style>
      div {
        margin: 25px 10px 30px 15px;
      }
      p {
        color: green;
      }
   </style>
</head>
```

Inline Styles

Codetrain

- An **inline style** may be used to apply a unique style for a single element.

- To use **inline styles**, add the **style attribute** to the relevant **element**. The style attribute can contain any **CSS property.**

- The example below shows how to change the color and the left margin of a **<h1>** element:

```
<h1 style=”color: #fff; background-color: grey”>
   This is some text about something interesting
</h1>
```

Now let's practise some CSS!

EXERCISE 1: INSTRUCTIONS

1. In your previous project folder from the HTML class(html-practice), create a new file in your text editor and save it as *styles.css* (make sure it's saved in the html-practice folder you created from the first lesson)
2. Let's start with linking our newly created css file to our first html document we created (from the first class).
3. Apply styles to your html using all the css properties we looked at in this lesson. Remember to use a variation of element selectors, class and id selectors.
4. Also, apply some inline css styles to your html.

EXERCISE 2: INSTRUCTIONS

1. In *forms.html* (this is from your second html lesson), create an internal style tag in the head section of your html document
2. Style your input fields(width, height, border-radius, etc).
3. Style the title of your form
4. Style the labels of your form
5. Import a font from google fonts and use that in your project
6. Freestyle from here… pun intended :)

Thanks!

Codetrain
Mahama Block,
Accra Digital Centre - Accra Central

info@codetraingh.com
www.codetraingh.com