Advanced CSS

Section 1: CSS Basics

1.1 What is CSS and How to write CSS?

CSS stands for Cascading Style Sheets. This allows us to create rules that specify how HTML elements should appear. CSS is used to make websites/web apps to appear aesthetically pleasing to the user.

Interesting Facts: CSS1 was first created in 1996. CSS2 was released in 1998. Finally, CSS3 first draft was released in 1999. Since 1999 it has continued developing and is still the current version as at April 2020.
CSS is split in separate modules like colours, backgrounds, animations, flex box, etc. and CSS creators/developers are working on developing those modules instead of creating another version i.e. CSS4. This is why the current CSS3 is hugely different from when it was released in 1999.

There are three methods of writing CSS which are inline, internal CSS and external CSS. It is best practice to use the external CSS method when writing CSS code because it is more readable, reusable and easier to maintain. The external CSS can be used across multiple HTML files by adding a link element tag in the HTML file referencing the .css file location to import the stylesheet.

The syntax below demonstrates how to write CSS:

```
h3 {
  background-color: blue;
  color: white;
}
```

The text represented in red is called the selector. This can be a HTML element and/or element attributes (such as id and classes) and/or pseudo selectors.

The opening and closing brackets wraps all the styles to be applied to the selector.

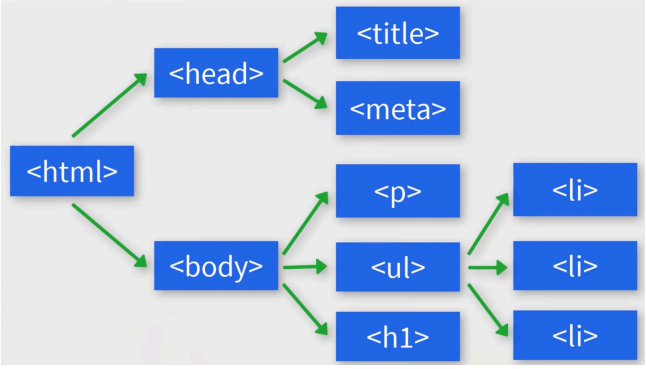
The text in blue is called properties which we want to style for the selector. Finally, the text in yellow is the values for the property.

The property and value pairs are actually called declarations. After each declaration we need to place a semi-colon (exception for the last declaration which does not require a semi-colon but it is recommended to add a semi-colon to all declarations).

1.2 HTML Elements Tree?

The HTML document is represented as a family tree. The family tree describes the relationships between family members and uses terms such as parent, child and sibling. A member of the family can be a parent to other while being a child of another family

member and a sibling of another family member. Below is a example representation of a HTML Elements tree for a .html document.



All HTML documents open with a <html> tag and all other element tags are contained within this element. Therefore, this is a parent element. The <html> element tag has not parents itself as well as no siblings.

Moving one level deeper we have the <head> and <body> element tags. These tags exists side by side which makes them siblings. They both also have the same parent <html> but they also contain children elements which make them parents themselves.

The <head> element has two children <meta> and <title> which are siblings to one another.

The <body> element has children of <p>, <ul> and <h1> which are all siblings of one another. Moving further down a level the <ul> has three children of <li> of its own making it a parent.

This is how we can view the structure of HTML documents as an HTML Element Tree. It is vital to have this knowledge when dealing with different types of selectors in CSS.

1.3 CSS Selectors

To style an element we require to select the element and there are multiple ways to access elements in CSS. Below is example syntax for selecting elements:

```
h1, h2 {
    color: green;
}

ul li {
    color: green;
}

#list-item {
    color: red;
}

.item {
    color: blue;
}

input[type="email"] {
    background-color: orange;
}
```

Comma (,) is used to select multiple elements and style them with the same declarations i.e. styles.

We can be more specific to reach child elements by using its parent as seen with the <li> element using its parent <ul> to select it.

The pound (#) symbol is used to select elements by their id attribute.

The period (.) symbol is used to select elements by their class attribute.

We can select an attribute value using square bracket ([ ]) notation.

It is important to understand the precedence of selectors. Execution of the code occurs line by line from top to bottom. However, specific selectors have higher precedence over less specific selectors and therefore the top to bottom rule does not always apply. In the example to the right, the ul li block of code has higher precedence over the li block of code and therefore the link items will have a text colour of green and not yellow.

Classes allow us to target multiple elements to share the same style while id allow us to target a specific element only because id must be unique.
Class and id attributes have a higher precedence over element selectors while the id selector has higher precedence over class selectors.

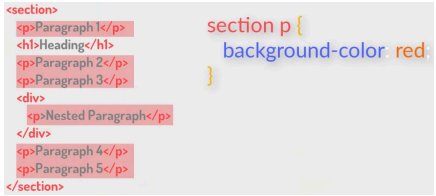
Disclaimer: It is possible to add the same id on multiple elements and they will be styled the same as we would have with classes. However, it is good practice to keep id attribute values unique while using classes to share attribute values.

Note: There are different ways of selecting elements but the common practice is to select using classes or ids instead of using the element names. In some cases a combination can be used.

1.3 CSS Combinators

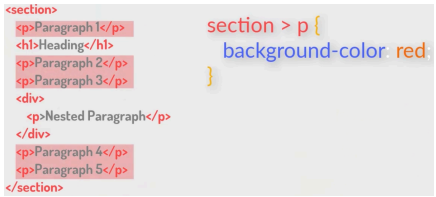
Combinators defines the relationship between selectors. In CSS there are four different combinators. Using the example HTML markup to the right we can explore the different combinator selectors in more detail:

Descendant (space) selector - matches all descendant elements of specified element.



All of the paragraph elements including the nested paragraph will be selected using the descendant selector and will be styled the same with a red background colour i.e. selecting all descendants.

Child (>) selector - selects all immediate children of the specified element.



The background colour for all paragraph elements will change to red except for the nested paragraph because it is not an immediate child of the <selector> parent element.

Adjacent Sibling (+) selector - selects the immediately following sibling of the specified element.

```
<section>
  <p>Paragraph 1</p>
  <h1>Heading</h1>
  <p>Paragraph 2</p>
  <p>Paragraph 3</p>
  <div>
    <p>Nested Paragraph</p>
  </div>
  <p>Paragraph 4</p>
  <p>Paragraph 5</p>
</section>
```

```
h1 + p {
  background-color: red;
}
```

The background colour will change only for the paragraph element which is placed immediately after the <h1> element.

General Sibling (~) selector - selects all siblings placed after a specified element.

```
<section>
  <p>Paragraph 1</p>
  <h1>Heading</h1>
  <p>Paragraph 2</p>
  <p>Paragraph 3</p>
  <div>
    <p>Nested Paragraph</p>
  </div>
  <p>Paragraph 4</p>
  <p>Paragraph 5</p>
</section>
```

```
h1 ~ p {
  background-color: red;
}
```

The background colour will be changed for all paragraphs which are placed after the <h1> element except for the nested paragraph element.

1.4 CSS Colours

There are multiple ways to choose/select colours in CSS.

The first method is by the colour name and all modern browsers support 140 standard colour names.

```
background-color: red;
```

The second method is by choosing rbg values which stands for red, blue and green and specifying a number between 0-255 for the three colours.

```
rgb(red, green, blue);
rgb(255, 0, 0);
```

The third method is selecting the rgba value . This is similar to the rgb values but adds a fourth value which is the alpha channel value. This allows us to select an opacity for the colour. The alpha value ranges between 0 and 1 i.e. 0-100% opacity.

```
rgba(255, 0, 0, 0.5)
```

The fourth method is to use hexadecimal values which start with a pound sign (#) followed by three values in pairs. The intensity of the colour can be controlled by values between 0 - f.

```
#ff0000 - red;
#00ff00 - green;
#0000ff - blue;
```

```
(#f00)
(#0f0)
(#00f)
```

Hexadecimals can be written in a shorten range using only three values but this will provide the least amount of ranged colours to select from.

Note: Colours have a huge effect on people's mood for example red is seen as a colour that grabs the user's attention and has an irritative effect. Green is seen as a colour related to nature and therefore is restful for the user's eyes. Blue is seen as a colour that relaxes, refreshes and produces peaceful feelings and moods. It is not recommended to use many colours on a webpage but a selection of colours that complement each other.

To colour your websites/web apps special attention should be taken to the:

- Perfect dominant colour
- Perfect colour scheme
- Perfect background colour
- Colours in the correct places

1.5 Inheritance

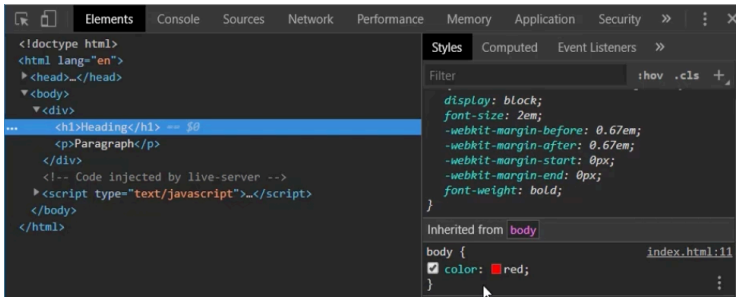
Inheritance is an important topic in CSS. The term itself describes the relationship between parent and child elements. In CSS each child element inherits its style from the parent element.

```
<body>
    <div>
        <h1>Heading</h1>
        <p>Paragraph</p>
    </div>
</body>
```

The example HTML markup to the left can be used to demonstrate the inheritance principle.

```
body {
    color: red;
}
```

The heading and paragraph elements will change text colour to red despite not being reference directly. This is because they inherited the <body> element because they are placed inside of the <body> element. This does not matter whether <body> is the direct or indirect parent.

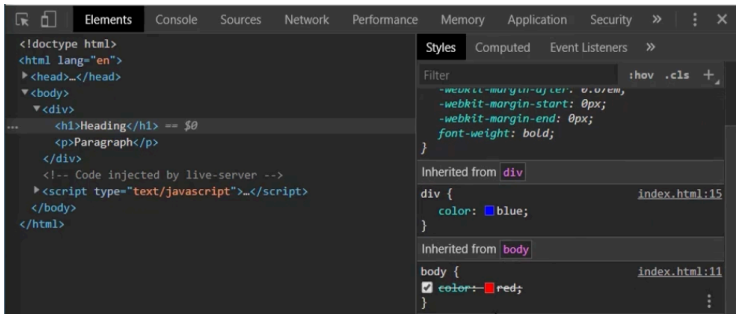


This can also be seen in the developer tools where the colour has been inherited from the <body> parent element for the , <h1> and <p> element.

```
body {
    color: red;
}

div {
    color: blue;
}
```

In this example the <h1> and <p> elements will inherit the blue text colour from their parent element style which will replace the red text colour style from the <body> element. This is due to CSS specificity and precedence.



This can also be seen in the developer tools Elements Styles tab where the colour: red now has a strike through which indicates that it is no longer applied. All the elements now inherit from their direct parent which is the element.

Therefore, closer parents have higher precedence to inherit its styles to their child elements. However, if the child element has its own style it will not inherit from any parent elements.

This is how inheritance operates in CSS i.e. child element inherits styles from its parent element. By default it is the closer parent element that has higher precedence to inherit styles to the child element, unless the child element has its own styles then the inherited style is overwritten and no longer applied.