Basic CSS Review

CSS Basics

- **What is CSS?**: Cascading Style Sheets (CSS) is a markup language used to apply styles to HTML elements. CSS is used for colors, background images, layouts and more.

- **Basic Anatomy of a CSS Rule**: A CSS rule is made up of two main parts: a selector and a declaration block. A selector is a pattern used in CSS to identify and target specific HTML elements for styling. A declaration block applies a set of styles for a given selector or selectors.

Here is the general syntax of a CSS rule:

```
selector {

  property: value;

}
```

- **meta name="viewport" Element**: This meta element gives the browser instructions on how to control the page's dimensions and scaling on different devices, particularly on mobile phones and tablets.

- **Default Browser Styles**: Each HTML element will have default browser styles applied to them. This usually includes items like default margins and paddings.

Inline, Internal, and External CSS

- **Inline CSS**: These styles are written directly within an HTML element using the style attribute. Most of the time you will not be using inline CSS due to separation of concerns.

Here is an example of inline CSS:

```
<p style="color: red;">This is a red paragraph.</p>
```

- **Internal CSS**: These styles are written within the <style> tags inside the head section of an HTML document. This can be useful for creating short code examples, but usually you will not need be using internal CSS.

- **External CSS**: These styles are written in a separate CSS file and linked to the HTML document using the link element in the head section. For most projects, you will use an external CSS file over internal or inline CSS.

Working With the width and height Properties

- **width Property**: This property specifies the width of an element. If you do not specify a width, then the default is set to auto. This means the element will take up the full width of its parent container.

- **min-width Property**: This property specifies the minimum width for an element.

- **max-width Property**: This property specifies the maximum width for an element.

- **height Property**: This property specifies the height of an element. Similarly, the height is auto by default, which means it will adjust to the content inside.

- **min-height Property**: This property specifies the minimum height for an element.

- **max-height Property**: This property specifies the maximum height for an element.

Different Types of CSS Combinators

- **Descendant Combinator**: This combinator is used to target elements that are descendants of a specified parent element. The following example will target all li items inside ul elements.

<ul>

  **
Example item one
**

  **
Example item two
**

  **
Example item three
**

**
**

ul li {

  background-color: yellow;

}

- **Child Combinator (>)**: This combinator is used to select elements that are direct children of a specified parent element. The following example will target all p elements that are direct children of the container class.

.container > p {

```
   background-color: black;

   color: white;

}
```

- **Next-sibling Combinator (+)**: This combinator selects an element that immediately follows a specified sibling element. The following example will select the paragraph element that immediately follows the h2 element.

`<h2>I am a sub heading</h2>`

`<p>This paragraph element will get a red background.</p>`

```
h2 + p {

   background-color: red;

}
```

- **Subsequent-sibling Combinator (~)**: This combinator selects all siblings of a specified element that come after it. The following example will style only the second paragraph element because it is the only one that is a sibling of the ul element and shares the same parent.

```
ul ~ p {

   background-color: green;

}
```

Inline, Block, and Inline-Block Level Elements

- **Inline Level Elements**: Inline elements only take up as much width as they need and do not start on a new line. These elements flow within the content, allowing text and other inline elements to appear alongside them. Common inline elements are span, anchor, and img elements.

- **Block Level Elements**: Block-level elements start on a new line and take up the full width available to them by default, stretching across the width of their container. Some common block-level elements are div, paragraph, and section elements.

- **Inline-Block Level Elements**: You can set an element to inline-block by using the display property. These elements behave like inline elements but can have a width and height set like block-level elements.

Margin and Padding

- **margin Property**: This property is used to apply space outside the element, between the element's border and the surrounding elements.

- **padding Property**: This property is used to apply space inside the element, between the content and its border.

- **margin Shorthand**: You can specify 1–4 values to set the margin sides. One value applies to all four sides; two values set top and bottom, then right and left; three values set top, horizontal (right and left), then bottom; four values set top, right, bottom, left.

- **padding Shorthand**: You can specify 1–4 values to set the padding sides. One value applies to all four sides; two values set top and bottom, then right and left; three values set top, horizontal (right and left), then bottom; four values set top, right, bottom, left.

CSS Specificity

- **Inline CSS Specificity**: Inline CSS has the highest specificity because it is applied directly to the element. It overrides any internal or external CSS. The specificity value for inline styles is (1, 0, 0, 0).

- **Internal CSS Specificity**: Internal CSS is defined within a style element in the head section of the HTML document. It has lower specificity than inline styles but can override external styles.

- **External CSS Specificity**: External CSS is linked via a link element in the head section and is written in separate .css files. It has the lowest specificity but provides the best maintainability for larger projects.

- **Universal Selector (*)**: a special type of CSS selector that matches any element in the document. It is often used to apply a style to all elements on the page, which can be useful for resetting or normalizing styles across different browsers. The universal selector has the lowest specificity value of any selector. It contributes 0 to all parts of the specificity value (0, 0, 0, 0).

- **Type Selectors**: These selectors target elements based on their tag name. Type selectors have a relatively low specificity compared to other selectors. The specificity value for a type selector is (0, 0, 0, 1).

- **Class Selectors**: These selectors are defined by a period (.) followed by the class name. The specificity value for a class selector is (0, 0, 1, 0). This means that class

selectors can override type selectors, but they can be overridden by ID selectors and inline styles.

- **ID Selectors**: ID selectors are defined by a hash symbol (#) followed by the ID name. ID selectors have a very high specificity, higher than type selectors and class selectors, but lower than inline styles. The specificity value for an ID selector is (0, 1, 0, 0).

- **!important keyword**: used to give a style rule the highest priority, allowing it to override any other declarations for a property. When used, it forces the browser to apply the specified style, regardless of the specificity of other selectors. You should be cautious when using !important because it can make your CSS harder to maintain and debug.

- **Cascade Algorithm**: An algorithm used to decide which CSS rules to apply when there are multiple styles targeting the same element. It ensures that the most appropriate styles are used, based on a set of well-defined rules.

- **CSS Inheritance**: The process by which styles are passed down from parent elements to their children. Inheritance allows you to define styles at a higher level in the document tree and have them apply to multiple elements without explicitly specifying them for each element.