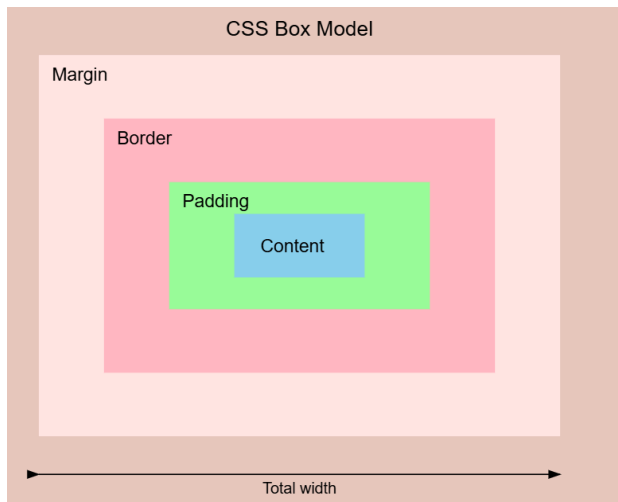


Take Home Assignment 2

Q1. With the help of a diagram, explain what the box model in CSS is. Which CSS properties are a part of the box model? Explain the concept of CSS Alternative Box Model. Outline the CSS code used to initialize the Website to use the CSS Alternative Box Model. (20 points)

Here is a model:



The CSS Box Model is, simply put, a way to help visualize the concept of how certain elements like, content, padding, etc., are set up and spaced, when it comes to web layouts. Every HTML element is treated as a box with four main components:

Content:

Content is the actual content of the element, this could include text boxes, images and videos. Essentially the main course of the website

Padding:

Think of padding as an internal breathing room that prevents content from touching the borders. Padding is particularly useful when you need to create whitespace around text, images, or other content within a container.

Border:

Border defines the visible boundary around an HTML element, positioned between the padding and margin. The border property allows you to create a variety of lines and styles, to help create clear distinction between other visual aspects of your website

Margin:

Margin works very similarly to padding but the clear distinction is that it creates spacing around HTML elements and how far apart they are. Margin unlike padding which is locked to the border is free to create as much distance as it would like.

Concept of CSS Alternative Box Model (CSSABM):

CSSABM changes how the width and height are calculated. In the standard box model, the width and height are defined by the content box size and padding and border being added to that. On the other hand, CSSABM makes width and height include the padding and border. One benefit of the CSSABM is that it can help with consistency but can make some things a little trickier.

This is the code that initializes CSSABM:

```
{  
  box-sizing: border-box;  
}
```

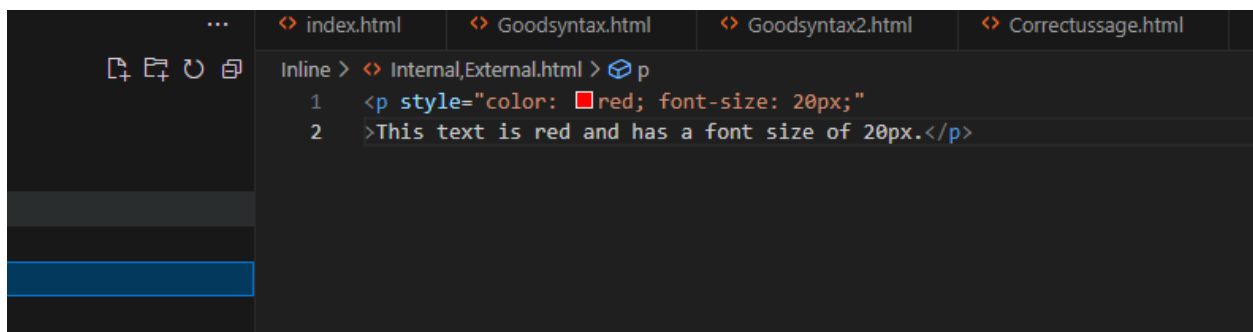
Q2. What are the three distinct ways in which you can embed CSS in your webpage. Explain your answer with the help of HTML-CSS code syntax. (15 points)

The three distinct ways of embedding CSS into your Webpage is Inline, Internal, and External

Inline:

This method applies CSS directly to the element using the style. It's useful for making quick changes to a single element but is not recommended for large projects because it can be hard to manage.

Example Code:

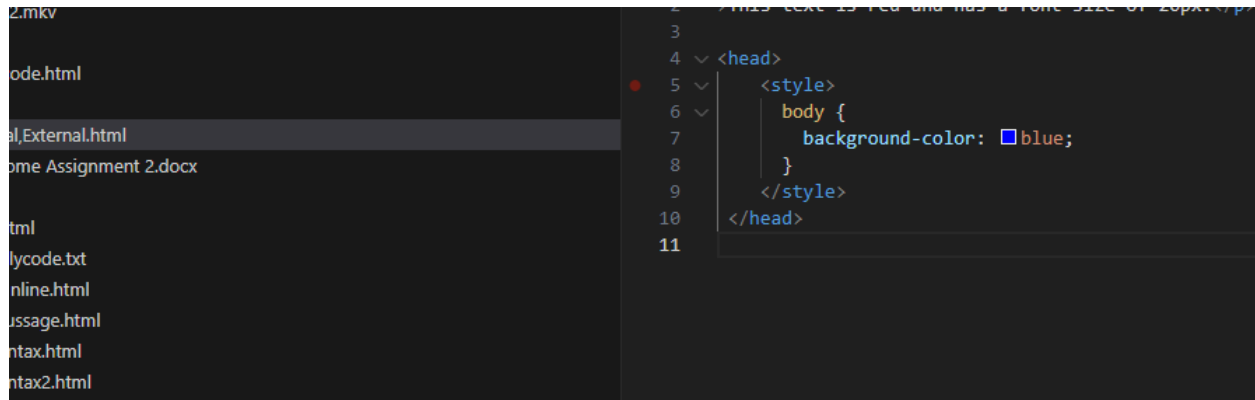
A screenshot of a code editor with a dark theme. The top of the editor shows four tabs: 'index.html', 'Goodsyntax.html', 'Goodsyntax2.html', and 'Correctusage.html'. The 'index.html' tab is active. On the left side, there is a sidebar with icons for file explorer, search, and other editor functions. The main code area shows the following HTML code:

```
Inline > < Internal, External.html > p  
1 <p style="color: red; font-size: 20px;"  
2 >This text is red and has a font size of 20px.</p>
```

The text 'This text is red and has a font size of 20px.' is displayed in red and a larger font size in the editor's preview area.

Internal:

Internal CSS is written inside the style tag within the head section of an HTML file. This is useful when styling a single webpage since all the styles are all located in one area.

A screenshot of a code editor with a dark theme. The left sidebar shows a file explorer with several files, including 'code.html'. The main editor area shows the content of 'code.html'. Lines 4 through 11 are visible, showing the HTML head section. A <style> tag is present, containing a CSS rule for the body element: 'body { background-color: blue; }'. The text 'blue' is highlighted in blue. Line numbers 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11 are visible on the left side of the editor area.

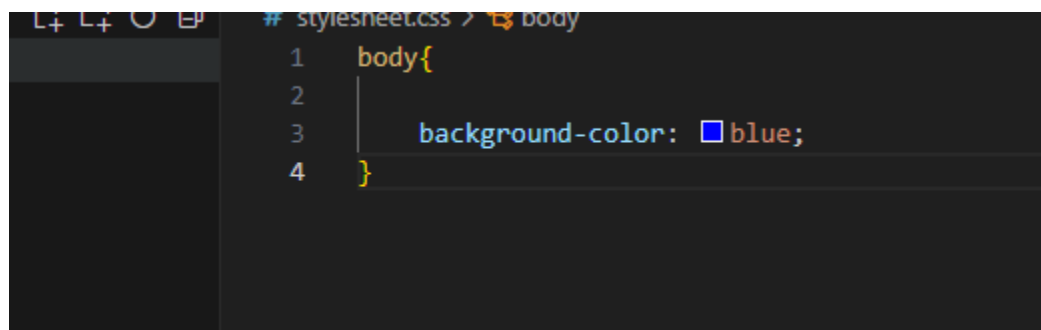
```
2 </p>This text is red and has a font size of 20px.</p>
3
4 <head>
5   <style>
6     body {
7       background-color: blue;
8     }
9   </style>
10 </head>
11
```

External:

External CSS is thought to be the best way to style a website because the CSS code is written in a separate file and linked to the HTML file using the link tag. This allows the coder to use CSS across multiple pages instead of having to rewrite the same code over and over again.

A screenshot of a code editor with a dark theme. The main editor area shows the HTML head section. Lines 4 through 7 are visible, showing a <link> tag that links to an external CSS file named 'styles.css'. The text 'styles.css' is underlined. Line numbers 3, 4, 5, 6, and 7 are visible on the left side of the editor area.

```
3
4
5 <head>
6   <link rel="stylesheet" href="styles.css">
7 </head>
```

A screenshot of a code editor with a dark theme. The left sidebar shows a file explorer with a file named 'stylesheet.css'. The main editor area shows the content of 'stylesheet.css'. Lines 1 through 4 are visible, showing a CSS rule for the body element: 'body { background-color: blue; }'. The text 'blue' is highlighted in blue. Line numbers 1, 2, 3, and 4 are visible on the left side of the editor area.

```
1 body{
2
3   background-color: blue;
4 }
```

Q3. Which HTML elements the following CSS codes style? (20 points)

- i. `*{color:"red";}`
 - a. Targets: All elements on the page

- ii. `#container{margin: 0px;}`
 - a. Targets: The element with `id="container"`
- iii. `.container{background-color: "purple";}`
 - a. Targets: All elements with `class="container"`
- iv. `ul li {color: "red";}`
 - a. Targets: All `` (list items) inside a `` (unordered list).
- v. `.container>p{color: "white";}`
 - a. Targets: `<p>` tags that are direct children of elements with `class="container"`.
- vi. `p + p {margin-top: 0;}`
 - a. Targets: A `<p>` that immediately follows another `<p>`
- vii. `h2 ~ p {margin-left: 5px;}`
 - a. Targets: All `<p>` tags that are siblings of an `<h2>` and appear after it.
- viii. `input [type="text"]{width: 30px;}`
 - a. Targets: `<input>` elements with `type="text"`
- ix. `.container, .menu {background-color: "gray";}`
 - a. Targets: Both elements with `class="container"` and `class="menu"`

Q4. Write a CSS code snippet that will make a webpage styling responsive to different devices i.e. the webpage will present itself in a particular way depending on the screen size of the device that is accessing the webpage (for e.g. viewing using a standard desktop computer vs. viewing using a mobile device). (10 points)

```
<> MultipleScreens.html
1  body {
2      font-size: 18px;
3      background-color: white;
4      text-align: left;
5  }
6
7  @media (max-width: 768px) {
8      body {
9          font-size: 16px;
10         background-color: lightgray;
11         text-align: center;
12     }
13 }
14
15 @media (max-width: 480px) {
16     body {
17         font-size: 14px;
18         background-color: lightblue;
19     }
20 }
21
```

Using the Media tag, we can look at the available size of the screen that is viewing our webpage and based on what is the size available, if it matches the conditions it applies the respective style based on the respective media Query.

Q5. Explain in details, the usage of the following CSS properties implemented together:

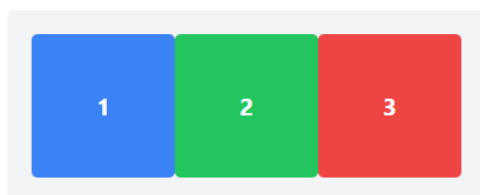
```
div{  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
    flex-direction: row;  
    flex-wrap: wrap;  
}
```

Include snapshots for the implementation effect of this code to elaborate and explain your answer. (20 points)

1. Div:

- a. This creates a <div>, flex container, which means that every element created inside it will follow a different set of rules than the normal rules for blocks set by like inline behavior

display: flex

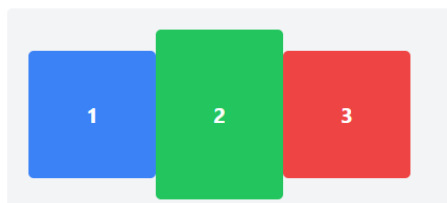


b.

2. align-items: center.

- a. This property aligns the item along the vertical axis
- b. If the container height is larger, all items will be centered vertically

+ align-items: center



c.

3. justify-content: space-between;

- a. It spreads the elements evenly inside the container, placing the first item at the start, the last item at the end, and equal spacing between them.

+ **justify-content: space-between**

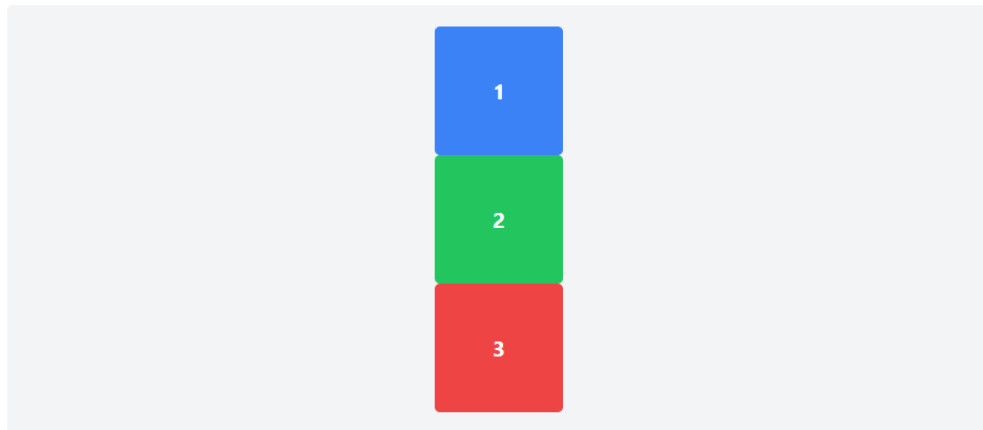


b.

4. flex-direction: row;

- a. Defines the main axis as left to right, Items appear in a horizontal row from left to right.

flex-direction: column

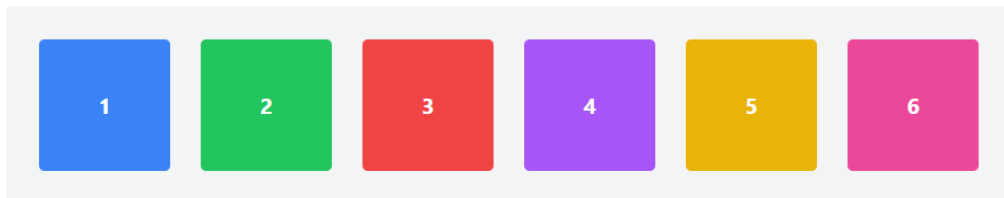


b.

5. flex-wrap: wrap;

- a. Allows items to move to the next line if there is not enough space in one row.

+ **flex-wrap: wrap**



b.

Q6. Explain the concept of cascading in cascading style sheets. How can you override the default cascading behavior in CSS? (5 points)

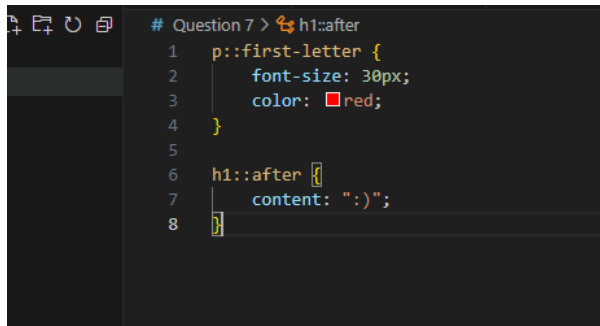
Cascading in CSS just means that styles are applied in a certain order based on specificity, order, importance, and inheritance. If multiple rules affect the same thing, the browser

picks the most specific or the last one written. You can change this by using stronger selectors, !important, inline styles, or by writing the rule later in the CSS. !important wins over everything, but it's better to avoid using it too much so your CSS stays easy to edit.

Q7. What are pseudo elements and pseudo classes in CSS? Give a syntactically correct example of a use-case scenario for each concept. (10 points)

Pseudo-elements and pseudo-classes are special CSS tools that help us style specific parts of elements or how they behave in different situations.

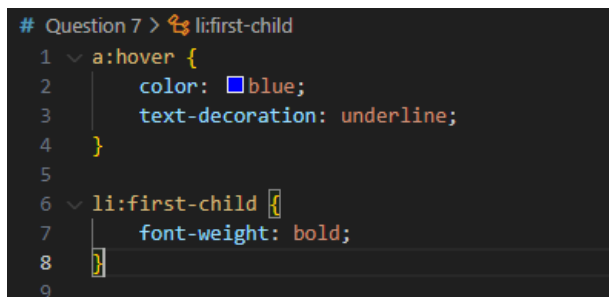
Pseudo-elements use double colons (::) and let us style specific bits of an element, like making the first letter of a paragraph bigger or adding something before or after content. Here's how we use pseudo-elements:



```
# Question 7 > h1::after
1 p::first-letter {
2   font-size: 30px;
3   color: red;
4 }
5
6 h1::after {
7   content: " :)";
8 }
```

Pseudo-classes use just one colon (:) and help us style elements when they're in different states - like when you hover over a link or when you want to style the first item in a list.

Here's some common examples:



```
# Question 7 > li:first-child
1 a:hover {
2   color: blue;
3   text-decoration: underline;
4 }
5
6 li:first-child {
7   font-weight: bold;
8 }
9
```

Citations

Questions 2: <https://dev.to/amaan56/3-ways-to-add-css-to-your-html-web-page-3k8j>

Questions 1-7: <https://developer.mozilla.org>