**HTMl & CSS**

**Understanding questions:**

1. Block-level elements:

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

For example: <p> and <div>

Inline-block elements:

Like an inline element except they have padding and margin added.

For example: <li>

Inline elements:

An inline element does not start on a new line. An inline element only takes up as much width as necessary.

For example: <span>

1. The DOCTYPE declaration instruct the web browser about the HTML version of the page.
2. The box model is used when we do the design layout of the page. The box wraps around every element. Padding is the amount of space from the element to the border and margin is the amount of space from the border to other elements.
3. The principles of responsive web design are:

Fluid layouts and Flexible images – a fluid grid lets the content to dynamically resize and re arrange to fit different devices. It ensure that the website stays consistent.

Enhance visual hierarchy – enhancing it ensures that users can easily navigate and understand the content regardless of their device.

Select appropriate typography – we must choose fonts that work well on different devices.

Don’t hide content – because mobile devices are the primary source of internet access, they need to be reliable and give all the information.

Prioritize accessibility – should be accessible to all users.

|  |  |  |  |
| --- | --- | --- | --- |
| **Absolute lengths** | | **Relative lengths** | |
| cm | centimeters | em | Relative to the font size of the element |
| mm | millimeters | ex | Relative to the x-hieght of the current font |
| px | pixels | ch | Relative to the width of the 0 |
| in | inches | rem | Relative to font-size of the root element |
| pt | Points | vw | Relative to 1% of the width of the viewport |
| pc | picas | Vh | Relative to 1% of height of the viewports |
|  |  | vmin | Relative to 1% of viewports smaller dimension |
|  |  | vmax | Relative to 1% of viewports larger dimension |
|  |  | % | Relative to the parent element |

1. Position:

The position CSS property sets how an element is positioned in a document. There are five different position values:

1. Static - The element is positioned according to the normal flow of the document.
2. Relative - The element is positioned according to the normal flow of the document, and then offset relative to itself based on the values of top, right, bottom, and left.
3. Absolute - The element is removed from the normal document flow, and no space is created for the element in the page layout.
4. Fixed - An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
5. Sticky - An element with position: sticky; is positioned based on the user's scroll position.

The top, right, bottom, and left properties determine the final location of positioned elements.

1. justify-content - used to align the flex items. Have many settings.

align-items - used to align the flex items.

align-self - specifies the alignment for the selected item inside the flexible container.

align-content - used to align the flex lines.

flex-grow - specifies how much a flex item will grow relative to the rest of the flex items.

flex-wrap – specifies whether the flex items should wrap or not.

Example:

<!DOCTYPE html>

<html>

<style>

.flex-container {

display: flex;

justify-content: center;

align-content: space-between;

}

</style>

</head>

<body>

<h1>Create a Flex Container</h1>

<div class="flex-container">

<div style=”flex-grow: 1”>1</div>

<div style=”flex-grow: 1”>>2</div>

<div style=”flex-grow: 1”>>3</div>

</div>

</body>

</html>

1. the difference between class and id is that id us specific to a single element and class can be assigned to multiple elements on a website.
2. The difference between div and span is that div is used for block-level organization and span is used for inline organization.
3. Examples for semantic elements:

<section> - defines a section in a document. It groups content together for example a chapter in a document.

<header> - represents a container for introductory content or navigational links.

<footer> - defines a footer of the document. Foe example, contact information.

1. ::before – inserts something before the content of each selected element.

::after – insert something after the content.

|  |  |  |
| --- | --- | --- |
|  | **CSS** | **SCSS** |
| **Definition** | a stylesheet language used to define the presentation and layout of a webpage written in HTML or XML. CSS follows a cascading hierarchy | a preprocessor scripting language that is a superset of CSS. It provides additional features and functionalities that are not available in regular CSS. |
| **Syntax** | plain-text syntax | structured syntax with additional features |
| **Variables** | Doesn’t allow to define variables | allows you to define variables to store values |
| **Nesting** | requires you to write each selector separately | allows you to nest selectors within other selectors |
| **mixins** | does not provide this functionality | allows you to create reusable code snippets using mixins |
| **File extention** | use the .css file extension | use the .scss file extension |
| **compilation** | files are interpreted by web browsers directly | files must be preprocessed into standard CSS files using a preprocessor |