**MITBOTS: TASK 1**

**SUBMITTED BY: SANIA MACTELIN A P**

**HTML:**

**1.What are Semantic Tags? Why they are better than just using <div>?**

Semantic Tags are tags which define the meaning of their content. They describe the structure of the webpage to both the browser and search engines. They display the title, logo and copyright details using header and footer tags.

Examples:

* <header> : contains logo, title etc..
* <footer> : usually contains copyright information.
* <section> : divides the webpage into sections.
* <nav> : contains navigation links.

Semantic tags are better than using <div> because they doesn’t give any meaning of the contents. However, semantic tags makes the code easier to understand and interpret the webpage’s structure.

**2. What is a <div>? Why and when do we use it?**

A<div> tag stands for division tag. It is used to group several elements together into a single container. It has no specific meaning and hence it’s a non-semantic element.

It is used when no specific meaning is needed and it is mostly used with CSS & JavaScript to style parts of a webpage.

**3. Role of <div> in layout:**

* A <div> acts like a container.
* It is used to group contents as they hold multiple elements.
* They are useful for styling with CSS.

**4. Difference between <div> and semantic tags?**

|  |  |
| --- | --- |
| **<div> tag** | **Semantic tags** |
| * A <div> tag is a generic container which doesn’t give meaning of the contents. * It is used for styling & layout purposes. * Example:   <div class = “nav”>…</div> | * Semantic tags give specific meaning to the content and make the code easier . * It is used for describing page structure. * Example:   <nav>, <header>,etc.. |

**CSS:**

**RESPONSIVENESS IN CSS**

**1. What is responsiveness?**

Responsiveness ensures that the website looks good and works well on any device. The layout of the website automatically adjusts to fit in different screen sizes and devices.

**2. How to make a page responsive?**

By using media queries, we can make a webpage responsive. By using relative units like %, em, rem instead of px and by using flexbox, grid for flexible layouts.

**3. Introduction to media queries:**

Media queries is a CSS feature that apply styles based on screen size and resolution of the devices. They make the webpages responsive.

**Example:**

@media (max-width: 300px){

body{

background-color: lightgreen;

}

}

**POSITIONING IN CSS**

**1. Static**

Static refers to the default position for all elements. It is used when we don’t need any special positioning.

Real life Example:

People standing in a queue.

**2. Relative**

Relative refers to the normal position however we can slightly move it with the help of top, bottom, left, right. They are used for small adjustments.

Real life Example:

A painting tilted on a wall for style.

**3. Absolute**

Absolute refers to the nearest positioned element. It uses top, bottom, left, right for its movement with respect to the nearest positioned element.

Real life Example:

A balloon tied to a cart, moves along with the cart. Here, the cart is the relative positioned element.

**4. Fixed**

It is always positioned relative to the screen. It doesn’t move even when the page is scrolled.

Real life Example:

A sticky note stuck to the book, it always stays in the same position.

**5. Sticky**

It is the combination of relative and fixed. At first, its relative but when it reaches a certain point it acts as fixed.

Real life Example:

In a bus, first we’re walking (relative), but when the bus moves you grab a handle (fixed).

**FLEXBOX**

**1.What is Flexbox?**

Flexbox is a special tool in CSS which is used to arrange items in rows and columns. It helps to align, space and position elements.

**2. When and why to use it?**

Flexbox is used when we want to arrange items either in rows or columns. To center and adjust elements according to our screen size. For easy alignment, sizing and responsive layout flexbox is used.

**3. Key properties (display: flex, justify-content, align-items, etc.)**

1. display : flex

It tells the container that we are using flexbox.

1. flex-direction

It chooses whether the items go left-to-right (row) or top-to-bottom (column).

1. justify-content

Moves items side to side (horizontally).

flex-start : Items go to the left.

center : Items go to the middle.

flex-end : Items go to the right.

space-between : Provides equal space between items.

space-around : Provides space around each item.

1. align-items

Moves items up and down (vertically).

flex-start : Items go to the left.

center : Items go to the middle.

flex-end : Items go to the right.

stretch : Items stretch to fill the space.

**CSS GRID**

**1. What is Grid Layout?**

It is a two dimensional layout with rows and columns. It is used to create complex web layouts.

**2. Basic syntax and Use cases:**

Syntax:

**HTML**

<div class = “container”>

<div>1</div>

<div>2</div>

</div>

**CSS**

.container{

display: grid;

}

**Use Cases:**

They are used in

* Photo galleries
* Dashboards & Webpage layouts.

**3. Difference between Flexbox and Grid?**

|  |  |
| --- | --- |
| **Flexbox** | **Grid** |
| * Flexbox is a one dimensional layout model. * Either horizontal or vertical. * Simple layouts. * Nesting is possible but complex. | * Grid is a two dimensional layout model. * Both horizontal or vertical. * Complex layouts. * Supports nesting grids. |