**TDC WEV DEV WEEK-01 DAY-04**

**Explanation of topics covered:**

**1. Box Model in CSS**

The CSS box model is a fundamental concept that explains how elements on a webpage are structured and displayed. It consists of:

1. **Content**: The actual content of the box, such as text or an image.
2. **Padding**: Space between the content and the border.
3. **Border**: A border surrounding the padding (if any) and content.
4. **Margin**: Space outside the border.

Here’s an example code and its output:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.box {

width: 200px;

padding: 20px;

border: 5px solid #000;

margin: 15px;

}

</style>

<title>Box Model Example</title>

</head>

<body>

<div class="box">This is a box.</div>

</body>

</html>

**Output**: A box with a width of 200px, 20px padding, 5px border, and 15px margin around it.

**2. CSS Positioning and Layout Basics**

CSS positioning allows you to control the placement of elements on a webpage. The primary positioning types are:

1. **Static**: Default position; elements are placed in the normal document flow.
2. **Relative**: Elements are positioned relative to their normal position.
3. **Absolute**: Elements are positioned relative to their nearest positioned ancestor.
4. **Fixed**: Elements are positioned relative to the viewport.
5. **Sticky**: Elements toggle between relative and fixed, depending on the scroll position.

Here’s an example of using absolute and relative positioning:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.relative-box {

position: relative;

width: 300px;

height: 150px;

border: 2px solid #000;

}

.absolute-box {

position: absolute;

top: 20px;

left: 50px;

width: 100px;

height: 100px;

background-color: #f00;

}

</style>

<title>Positioning Example</title>

</head>

<body>

<div class="relative-box">

<div class="absolute-box"></div>

</div>

</body>

</html>

**Output**: A larger box with a smaller red box positioned 20px from the top and 50px from the left within the larger box.

**3. Using Flexbox for Layout**

Flexbox is a powerful layout module that allows for easy alignment and distribution of space among items in a container. Here's a basic example:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.container {

display: flex;

justify-content: space-between;

align-items: center;

height: 200px;

border: 2px solid #000;

}

**Output**: A container with three blue boxes, evenly spaced and centered vertically.