**Department of Computer Science & Engineering**

**Sub: Web Technology Lab**

**Lab Worksheet2**

**Upload the document with correct code:** [**https://forms.gle/QKdmDDEUqT74EiTq5**](https://forms.gle/QKdmDDEUqT74EiTq5)

**Name: Hariprasad B R**

**USN: 1MS22CS057**

1. You have a simple web page with a header, a main content area, and a footer. The layout is supposed to be responsive, but on smaller screens, the content spills out of the viewport, causing a horizontal scrollbar. Your task is to identify and fix the issue using CSS debugging techniques.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Debugging Exercise</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header {

background-color: #333;

color: #fff;

padding: 1em;

text-align: center;

}

main {

display: grid;

grid-template-columns: repeat(2, 1fr);

gap: 10px;

padding: 20px;

}

.content-box {

background-color: #f0f0f0;

padding: 20px;

border: 1px solid #ddd;

}

footer {

background-color: #333;

color: #fff;

padding: 1em;

text-align: center;

clear: both;

}

</style>

</head>

<body>

<header>

<h1>Debugging Exercise</h1>

</header>

<main>

<div class="content-box">

<h2>Content Box 1</h2>

<p> CSS is the language we use to style an HTML document.CSS describes how HTML elements should be displayed.</p>

</div>

<div class="content-box">

<h2>Content Box 2</h2>

<p>CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.</p>

</div>

</main>

<footer>

<p>&copy; 2025 Debugging Exercise</p>

</footer>

</body>

</html>

#### Problem:

On smaller screens, the main grid layout causes content to overflow because it is always set to two columns (grid-template-columns: repeat(2, 1fr);), even when the screen is too narrow.

#### Solution:

Use @media queries to make the grid layout adapt to smaller screens by switching to a single-column layout. Also, add max-width: 100% to ensure no element exceeds the screen width.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Debugging Exercise</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

overflow-x: hidden; /\* Prevent horizontal scroll \*/

}

header {

background-color: #333;

color: #fff;

padding: 1em;

text-align: center;

}

main {

display: grid;

grid-template-columns: repeat(2, 1fr);

gap: 10px;

padding: 20px;

}

.content-box {

background-color: #f0f0f0;

padding: 20px;

border: 1px solid #ddd;

max-width: 100%;

}

footer {

background-color: #333;

color: #fff;

padding: 1em;

text-align: center;

clear: both;

}

@media (max-width: 600px) {

main {

grid-template-columns: 1fr; /\* Change to one column on small screens \*/

}

}

</style>

</head>

<body>

<header>

<h1>Debugging Exercise</h1>

</header>

<main>

<div class="content-box">

<h2>Content Box 1</h2>

<p>CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.</p>

</div>

<div class="content-box">

<h2>Content Box 2</h2>

<p>CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.</p>

</div>

</main>

<footer>

<p>&copy; 2025 Debugging Exercise</p>

</footer>

</body>

</html>

1. You have a navigation bar with a logo and a list of menu items. The logo and the menu items should be vertically aligned, but currently, they are not. Your task is to identify and fix the issue.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Vertical Alignment Exercise</title>

<style>

nav {

display: flex;

justify-content: space-between;

align-items: flex-start; /\* Change this to fix the issue \*/

padding: 1em;

background-color: #333;

color: #fff;

}

nav img {

width: 50px;

height: 50px;

}

nav ul {

list-style: none;

padding: 0;

margin: 0;

display: flex;

}

nav li {

margin-right: 20px;

}

</style>

</head>

<body>

<nav>

<img src="logo.png" alt="Logo">

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</body>

</html>

#### Problem:

The align-items: flex-start; property causes the logo and menu items to be misaligned.

#### Solution:

Change align-items: center; to ensure both elements align properly.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Vertical Alignment Exercise</title>

<style>

nav {

display: flex;

justify-content: space-between;

align-items: center; /\* Fix: Align items vertically \*/

padding: 1em;

background-color: #333;

color: #fff;

}

nav img {

width: 50px;

height: 50px;

}

nav ul {

list-style: none;

padding: 0;

margin: 0;

display: flex;

}

nav li {

margin-right: 20px;

}

nav a {

color: white;

text-decoration: none;

}

</style>

</head>

<body>

<nav>

<img src="logo.png" alt="Logo">

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</body>

</html>

1. The button below has a hover effect that is not working correctly. Fix the transition and animation issues.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Debug Animation</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<button class="animated-btn">Hover Me</button>

</body>

</html>

.animated-btn {

background: #007BFF;

color: white;

padding: 10px 20px;

border: none;

transition: background-color ease-in; /\* Issue: Missing duration \*/

}

.animated-btn:hover {

background-color: red;

transform: scale(1.5); /\* Issue: Should be smooth \*/

}

#### Problem:

1. The transition property is missing a duration.
2. The transform: scale(1.5); does not have a smooth transition.

#### Solution:

Add a 0.3s duration to transition, and ensure all properties (including transform) are included in the transition.

.animated-btn {

background: #007BFF;

color: white;

padding: 10px 20px;

border: none;

transition: background-color 0.3s ease-in, transform 0.3s ease-in-out;

}

.animated-btn:hover {

background-color: red;

transform: scale(1.1);

}

1. Fix Content Overflow:

A div with specified dimensions overflows its container despite padding/margin declarations.

<div class="container">

<div class="content">Overflowing content</div>

</div>

<style>

.container {

width: 300px;

border: 2px solid black;

}

.content {

width: 300px;

padding: 20px;

margin: 15px;

background: lightblue;

}

</style>

#### Problem:

1. The content div has a width of 300px, but additional padding and margin cause it to exceed 300px, leading to overflow.

#### Solution:

Use box-sizing: border-box; to include padding and border within the defined width

<div class="container">

<div class="content">Overflowing content</div>

</div>

<style>

.container {

width: 300px;

border: 2px solid black;

overflow: hidden; // Prevent overflow

}

.content {

width: 100%; // Adjusted width

padding: 20px;

margin: 0; // Removed margin

background: lightblue;

box-sizing: border-box; // Prevent overflow

}</style>