🧑‍💻 Day 4 Lab Sheet – CSS Layouts & Positioning

Objective

By the end of this lab, you will:

- Understand CSS display types (block, inline, inline-block, none)
- Practice CSS positioning (static, relative, absolute, fixed, sticky)
- Use float for simple layouts
- Build a basic multi-section webpage layout

Task 1: Display Property Practice

1. Create a file named day4_display.html.
2. Add the following HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Display Practice</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>CSS Display Example</h1>
  <p>This is a paragraph.</p>
  <span>This is inline text.</span>
  <button>Click Me</button>
</body>
</html>
```

3. In style.css, try these rules:

```
p { display: block; background: lightblue; }
span { display: inline; background: yellow; }
button { display: inline-block; padding: 10px; }
```

✅ Observe the difference between block, inline, and inline-block.

Task 2: Positioning Practice

1. Create a file day4_position.html.
2. Add three <div> elements:

```
<div class="box static">Static Box</div>
<div class="box relative">Relative Box</div>
<div class="box absolute">Absolute Box</div>
```

3. In style.css:

```
.box {
  width: 150px;
  height: 100px;
  margin: 20px;
  background: lightgray;
  border: 2px solid black;
}

.relative {
  position: relative;
  top: 20px;
  left: 40px;
  background: lightgreen;
}

.absolute {
  position: absolute;
  top: 50px;
  left: 200px;
  background: lightcoral;
}
```

✅ Notice how relative shifts slightly, while absolute moves independently.

Task 3: Sticky & Fixed Elements

1. Add a navigation bar at the top of your HTML page:

```
<nav class="navbar">
  <a href="#">Home</a> | <a href="#">About</a> | <a href="#">Contact</a>
</nav>
<p>Scroll down to see effect...</p>
<p style="height:1000px;">Lots of content here...</p>
```

2. In style.css:

```
.navbar {
  position: fixed;
  top: 0;
  width: 100%;
  background: navy;
  color: white;
  padding: 10px;
}
```

✅ The navbar stays fixed when scrolling.
👉 Change fixed to sticky and compare behavior.

Task 4: Float Practice

1. Add an image and text:

```
<img src="https://via.placeholder.com/150" class="float-img">
<p>
  This paragraph text should wrap around the image.
  CSS float is useful for placing images inside text.
</p>
```

2. In style.css:

```
.float-img {
  float: right;
  margin: 10px;
}
```

✅ The text should wrap around the image.

Task 5: Mini Layout Project – Two Column Blog

👉 Create a file day4_blog.html with this structure:

```
<header>My Blog</header>
<div class="container">
  <div class="sidebar">Sidebar Menu</div>
  <div class="content">Main Content Area</div>
</div>
<footer>© 2025 My Blog</footer>
```

In style.css:

```
header, footer {
  background: black;
  color: white;
  text-align: center;
  padding: 10px;
}

.container {
  width: 100%;
}

.sidebar {
  width: 30%;
  float: left;
  background: lightgray;
  padding: 20px;
}

.content {
  width: 70%;
  float: left;
  background: white;
  padding: 20px;
}
```

✅ You now have a simple **2-column blog layout** with a header and footer.

✅ **Deliverables**

- day4_display.html + style.css
- day4_position.html + style.css
- day4_blog.html + style.css

Your output should demonstrate correct use of **display, position, float, and layout**.