

Aim: To demonstrate CSS box model, flex, grid and various other properties

Properties covered :

- 1) flex and grid part
- 2) position property
- 3) margin and padding
- 4) the box model
- 5) others

Results: (Program printout with output)

```
index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Properties Demo</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <header class="header">Fixed Header</header>

  <div class="static-box">Static Position</div>
  <div class="relative-box">Relative Position</div>
  <div class="absolute-box">Absolute Position</div>

  <div class="relative-container">
    <div class="absolute-inside">Absolute Inside Relative</div>
  </div>

  <div class="two-columns">
    <div class="left-column">Left</div>
    <div class="right-column">Right</div>
  </div>

  <div class="float-container">
```

```

    <div class="float-left">Float Left</div>
    <div class="float-right">Float Right</div>
</div>

<div class="clearfix">
    <div class="float-box">Float 1</div>
    <div class="float-box">Float 2</div>
</div>

<!-- Flex and Grid Layout -->
<div class="flexbox-container">
    <div class="flex-item">Flex 1</div>
    <div class="flex-item">Flex 2</div>
    <div class="flex-item">Flex 3</div>
</div>

<div class="grid-container">
    <div class="grid-item">Grid 1</div>
    <div class="grid-item">Grid 2</div>
    <div class="grid-item">Grid 3</div>
    <div class="grid-item">Grid 4</div>
</div>

<div class="border-box">Border Box</div>
<div class="box">Width, Height, Margin & Padding</div>
<div class="box-model">Box Model Example</div>
<div class="quirks-mode">IE Quirks Mode</div>

<span class="inline-element">Inline Element</span>
<div class="clear-example">Clear Property</div>
<div class="opacity-box">Opacity Example</div>
<div class="visibility-box">Hidden Element</div>
<div class="display-box">Block Display</div>
<div class="overflow-box">Overflow Hidden</div>

<footer class="footer">Sticky Footer</footer>

</body>
</html>

```

style.css

```

/* Position Property */
.static-box {

```

```
    position: static;
    background: lightgray;
    padding: 10px;
}
```

```
.relative-box {
    position: relative;
    left: 20px;
    background: lightblue;
    padding: 10px;
}
```

```
.absolute-box {
    position: absolute;
    top: 50px;
    left: 50px;
    background: pink;
    padding: 10px;
}
```

```
.relative-container {
    position: relative;
    height: 100px;
    background: lightgreen;
}
```

```
.absolute-inside {
    position: absolute;
    top: 10px;
    left: 10px;
    background: coral;
    padding: 5px;
}
```

```
.two-columns {
    position: relative;
}
```

```
.left-column, .right-column {
    position: absolute;
    width: 45%;
    padding: 10px;
}
```

```
.left-column {
    left: 0;
    background: lightyellow;
}
```

```
.right-column {
    right: 0;
    background: lightcyan;
}
```

```

}

/* Flex and Grid Layout */
.flexbox-container {
    display: flex;
    justify-content: space-around;
    background: lightgray;
    padding: 10px;
}
.flex-item {
    background: lightcoral;
    padding: 20px;
    flex: 1;
    text-align: center;
}

.grid-container {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 10px;
    background: lightblue;
    padding: 10px;
}
.grid-item {
    background: lightseagreen;
    padding: 20px;
    text-align: center;
}

/* Margin and Padding */
.box {
    width: 200px;
    height: 100px;
    margin: 10px;
    padding: 10px;
    background: lavender;
}

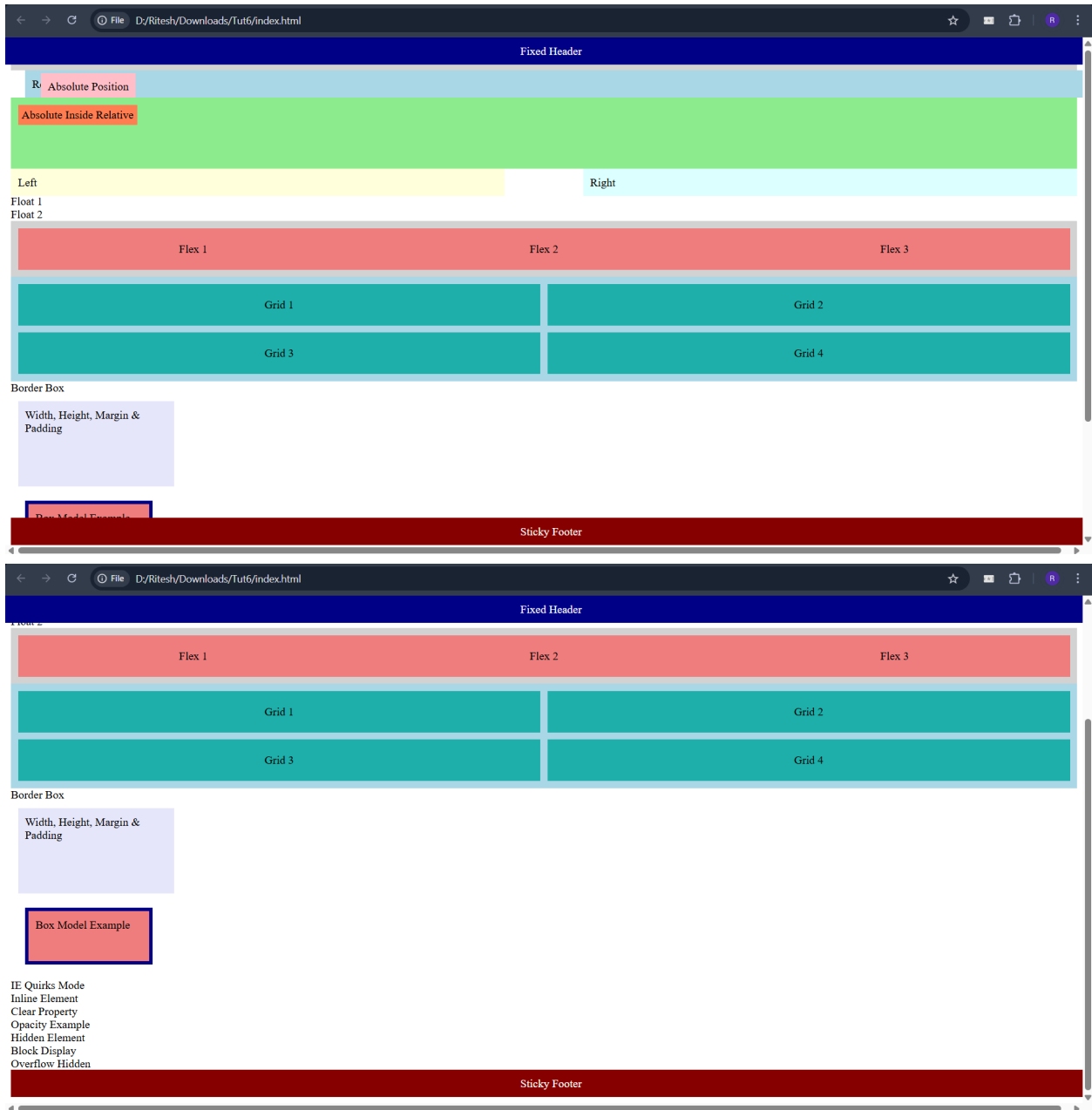
/* The Box Model */
.box-model {
    width: 150px;
    height: 50px;
    border: 5px solid darkblue;
    padding: 10px;
    margin: 20px;
    background: lightcoral;
}

```

```
/* Fixed header */
.header {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  background: darkblue;
  color: white;
  text-align: center;
  padding: 10px;
  z-index: 1000;
}

/* Sticky footer */
.footer {
  position: sticky;
  bottom: 0;
  width: 100%;
  background: darkred;
  color: white;
  text-align: center;
  padding: 10px;
}
```

Output WebPage :



Outcomes: CO2: Create Web pages using HTML 5 and CSS

Conclusion: (Conclusion to be based on the outcomes achieved)

From this experiment, I learned how different CSS properties affect web page layout and styling. By using positioning (static, relative, absolute), float, flexbox, and grid, I understood how elements can be arranged dynamically. I also explored the box model, margins, padding, borders, visibility, opacity, and display properties, which control spacing and appearance. This experiment helped me see how these concepts work together to create well-structured and visually appealing web pages.