4. Write a program to apply different types of selector forms
i. Simple selector (element, id, class, group, universal)
ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
iii. Pseudo-class selector
iv. Pseudo-element selector
v. Attribute selector

DESCRIPTION:

This program demonstrates the use of various **CSS selector forms** to apply styles to HTML elements.

1. **Simple Selectors** are used to select elements based on element name, id, class, group of elements, and the universal selector. These selectors help apply basic styling to specific or multiple elements efficiently.
2. **Combinator Selectors** are used to select elements based on their relationship with other elements. Descendant, child, adjacent sibling, and general sibling selectors define how elements are styled depending on their position and hierarchy in the HTML structure.
3. **Pseudo-class Selectors** are used to style elements based on their state or user interaction, such as hovering over a link, focusing on an input field, or selecting odd and even list items.
4. **Pseudo-element Selectors** are used to style specific parts of an element, such as the first letter or first line of text, or to insert content before or after an element.
5. **Attribute Selectors** are used to select and style elements based on the presence or value of their attributes, such as styling input fields based on their type.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>CSS Selectors Experiment</title>
  <style>
    /* Simple Selectors */
    p {
      color: purple;
      padding: 15px;
      font-size: 18px;
    }

    #highlighted {
      font-style: italic;
    }

    .focus {
      background-color: lightgreen;
      border-radius: 10px;
      padding: 10px;
    }

    section, article {
      margin: 10px;
    }
```

```css
* {
   font-family: Verdana, sans-serif;
}

/* Combinator Selectors */
section p {
   color: orange;
}

article > p {
   border: 2px dashed red;
}

h2 + p {
   font-weight: bold;
}

h2 ~ p {
   background-color: lightblue;
}

/* Pseudo-class Selectors */
a:hover {
   color: green;
}

input:focus {
   outline: 3px solid red;
}

li:nth-child(odd) {
   background-color: #ffeb99;
}

li:nth-child(even) {
   background-color: #99d9ff;
}

/* Pseudo-element Selectors */
p::first-letter {
   font-size: 2em;
   font-weight: bold;
}

section p::before {
   content: "★ ";
   color: gold;
}

/* Attribute Selectors */
```

```
        [type="text"] {
            border: 3px solid purple;
            padding: 5px;
            border-radius: 8px;
        }

        [type="submit"] {
            background-color: darkorange;
            padding: 8px;
            color: white;
            border: none;
            border-radius: 5px;
        }
    </style>
</head>

<body>

<h1>CSS Selector Experiment</h1>

<!-- SIMPLE SELECTORS -->
<h3>Simple Selectors (Element, ID, Class, Universal)</h3>
<div id="highlighted">
    <p>Element selector (p) and ID selector </p>
    <p class="focus">Class selector </p>
</div>

<hr>

<!-- COMBINATOR SELECTORS -->
<h3>Combinator Selectors</h3>

<h4>Adjacent & General Sibling</h4>
<h2>Heading</h2>
<p>Adjacent sibling selector </p>
<p>General sibling selector </p>

<hr>

<!-- PSEUDO-CLASS SELECTORS -->
<h3>Pseudo-Class Selectors</h3>
<ul>
    <li>Odd item</li>
    <li>Even item</li>
    <li>Odd item</li>
    <li>Even item</li>
</ul>

<a href="#">Hover pseudo-class </a>
```

```
<hr>

<!-- PSEUDO-ELEMENT SELECTORS -->
<h3>Pseudo-Element Selectors</h3>
<p>Pseudo-element applied </p>

<hr>

<!-- ATTRIBUTE SELECTORS -->
<h3>Attribute Selectors</h3>
<form>
  <label for="username">Username:</label>
  <input type="text" id="username">
  <br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

5. CSS with Color, Background, Font, Text and CSS Box Model
a. Write a program to demonstrate the various ways you can reference a color in CSS.
DESCRIPTION:
This program clearly demonstrates the various ways of referencing colors in CSS by applying different color formats to HTML elements. It uses named colors, hexadecimal values, and short hexadecimal values to represent colors in a simple and readable manner. The program also includes RGB and RGBA color formats, where RGBA allows the use of transparency through an alpha value. In addition, HSL and HSLA color formats are used to define colors based on hue, saturation, and lightness, with HSLA supporting opacity control. CSS variables (custom properties) are used to store reusable color values, improving consistency and maintainability.
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>CSS Color References</title>
  <style>
    /* Defining CSS variables */
    :root {
      --custom-blue: #4A90E2;
      --custom-green: rgb(34, 197, 94);
    }

    body {
      display: flex;
      flex-wrap: wrap;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      background-color: #f4f4f4;
      font-family: Arial, sans-serif;
    }

    .color-box {
      width: 200px;
      height: 200px;
      display: flex;
      align-items: center;
      justify-content: center;
      color: white;
      font-weight: bold;
      font-size: 14px;
      text-align: center;
      margin: 10px;
      border-radius: 10px;
      box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.2);
    }

    /* Different ways to define colors */
    .named-color { background-color: red; }
    .hex-color { background-color: #00ff00; color: black; } /* Green */
    .short-hex-color { background-color: #0af; } /* Light Blue */
    .rgb-color { background-color: rgb(255, 165, 0); } /* Orange */
    .rgba-color { background-color: rgba(255, 0, 0, 0.5); } /* Semi-transparent Red */
    .hsl-color { background-color: hsl(240, 100%, 50%); } /* Blue */
    .hsla-color { background-color: hsla(300, 76%, 72%, 0.8); color: black; } /* Semi-transparent Pink */
    .css-variable { background-color: var(--custom-blue); } /* Custom variable */
    .css-variable-green { background-color: var(--custom-green); color: black; } /* Custom Green */

  </style>
</head>
<body>

  <div class="color-box named-color">Named Color: Red</div>
  <div class="color-box hex-color">Hex: #00ff00</div>
  <div class="color-box short-hex-color">Short Hex: #0af</div>
  <div class="color-box rgb-color">RGB: rgb(255, 165, 0)</div>
  <div class="color-box rgba-color">RGBA: rgba(255, 0, 0, 0.5)</div>
  <div class="color-box hsl-color">HSL: hsl(240, 100%, 50%)</div>
  <div class="color-box hsla-color">HSLA: hsla(300, 76%, 72%, 0.8)</div>
  <div class="color-box css-variable">CSS Variable: var(--custom-blue)</div>
  <div class="color-box css-variable-green">CSS Variable: var(--custom-green)</div>

</body>
</html>
```

b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.

DESCRIPTION:

This program demonstrates how to apply a background image in CSS with precise control over its position and orientation. The image is placed halfway down the page and tilted horizontally to create a visually angled effect. Using the background-attachment: fixed property ensures that the image stays in the same position even when the user scrolls up or down. This example illustrates how to combine **background positioning, transformation, and fixed attachment** to create a stationary and visually appealing background effect on a webpage.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Fixed Background Image Halfway Down</title>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
      color: white;
      text-align: center;
      background-color: #333;
    }

    /* Ensures enough scrolling space */
    .content {
      height: 200vh;
      display: flex;
      justify-content: center;
      align-items: center;
      font-size: 24px;
    }

    /* Background Image Section */
    .background-image {
      position: fixed;
      top: 50%;
      left: 0;
      width: 100%;
      height: 50vh;
      background-image: url('https://www.tutorialkart.com/img/mountains.jpg');
      background-size: cover;
      background-position: center;
      transform: scaleX(-1); /* Flip horizontally */
      z-index: -1; /* Keeps it in the background */
    }

    /* Text Content on Top */
    .text-overlay {
      position: relative;
      z-index: 1;
      font-size: 28px;
```

```
            font-weight: bold;
        }
    </style>
</head>
<body>

    <div class="content">
        <div class="text-overlay">
            Scroll to see the effect. The background stays fixed!
        </div>
    </div>

    <div class="background-image"></div>

</body>
</html>
```

c. Write a program using the following terms related to CSS font and text:
i. font-size ii. font-weight iii. font-style iv. text-decoration v. text-transformation vi. text-alignment

<u>DESCRIPTION:</u>

This program demonstrates various CSS properties used to style fonts and text on a webpage. It shows how to change the **font size** using font-size to make text larger or smaller, and how to apply **font weight** using font-weight to make text bold or light. The **font style** property is used to make text italic or normal, while **text-decoration** adds effects such as underline, overline, or line-through. The **text-transform** property changes the case of the text to uppercase, lowercase, or capitalize, and **text-align** is used to align text to the left, right, center, or justify it.

<u>CODE:</u>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>CSS Font and Text Effects</title>
    <style>
        h2 {
            text-align: center;
        }

        /* Different Text Effects */
        .large-text {
            font-size: 24px;
        }

        .bold-text {
            font-weight: bold;
        }

        .italic-text {
            font-style: italic;
        }
        .underline-text {
            text-decoration: underline;
        }

        .uppercase-text {
            text-transform: uppercase;
        }

        .capitalize-text {
            text-transform: capitalize;
        }

        .center-text {
            text-align: center;
        }

        .justify-text {
            text-align: justify;
        }
    </style>
</head>
<body>

    <h2>CSS Font and Text Properties</h2>

    <p class="large-text">This paragraph has a larger font size.</p>

    <p class="bold-text">This paragraph has bold text.</p>

    <p class="italic-text">This paragraph is in italic style.</p>

    <p class="underline-text">This paragraph is underlined.</p>

    <p class="uppercase-text">this text is in uppercase.</p>

    <p class="capitalize-text">this text is capitalized.</p>

    <p class="center-text">This text is center-aligned.</p>

    <p class="justify-text">This paragraph has justified text. Justified text ensures that both
edges of the paragraph are aligned, creating a neat block of text.</p>

</body>
</html>
```

d. Write a program, to explain the importance of CSS Box model using
i. Content ii. Border iii. Margin iv. Padding

DESCRIPTION:
The CSS Box Model is a fundamental concept used to control the layout and spacing of elements on a web page. Every HTML element is represented as a rectangular box, which consists of four main parts: content, padding, border, and margin. Understanding the box model helps in designing well-structured and visually balanced web pages.

1. **Content**
   The content area is the innermost part of the box. It contains the actual text, images, or other media inside an element. The size of the content area determines how much space the element needs to display its information clearly.
2. **Padding**
   Padding is the space between the content and the border. It creates internal spacing within an element, making the content easier to read and preventing it from touching the border.
3. **Border**
   The border surrounds the padding and content. It defines the boundary of an element and can be styled using different colors, widths, and styles to visually separate elements from each other.
4. **Margin**
   Margin is the outermost space around an element. It controls the distance between an element and neighboring elements, helping to create proper spacing and alignment on the web page.

CODE:
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Box Model Demonstration</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      background-color: #e0f7fa;
    }

    .box {
      width: 280px;
      background-color: #ffffff;
      padding: 25px; /* space inside the border */
      border: 8px solid #00796b; /* box border */
      margin: 15px auto; /* space outside the box */
      text-align: center;
      box-shadow: 3px 3px 12px rgba(0, 0, 0, 0.2);
      border-radius: 8px;
    }

    .label {
      font-weight: bold;
      color: #004d40;
      margin-bottom: 8px;
```

```
        }
        .content {
           background-color: #b2dfdb;
           padding: 8px;
        }

        /* Modified individual boxes for demonstration */
        .padding-demo {
           padding: 40px;
           background-color: #e0f2f1;
        }

        .border-demo {
           border: 8px dashed #d32f2f;
           background-color: #ffebee;
        }

        .margin-demo {
           margin: 60px auto;
           background-color: #fce4ec;
        }

        .content-demo {
           width: 200px;
           margin: 0 auto;
           background-color: #c8e6c9;
              padding: 10;
        }
    </style>
</head>
<body>

    <h2>CSS Box Model Demonstration</h2>

    <div class="box content-demo">
        <div class="label">Content</div>
        <div class="content">This is the content area</div>
    </div>

    <div class="box padding-demo">
        <div class="label">Padding</div>
        <div class="content">This box has extra padding</div>
    </div>

    <div class="box border-demo">
        <div class="label">Border</div>
        <div class="content">This box has a thick dashed border</div>
    </div>

    <div class="box margin-demo">
```

```
        <div class="label">Margin</div>
        <div class="content">This box has extra margin</div>
    </div>

</body>
</html>
```