

## **WTMA LAB MANUAL – CSS PROGRAMS**

### **Experiment – CSS Stylesheets & Animation**

---

## **Program 1: Develop and demonstrate the usage of Inline, Internal and External style sheet using CSS**

---

### **Aim**

To design a simple web page and demonstrate the usage of:

1. Inline CSS
  2. Internal CSS
  3. External CSS
- 

### **Software Required**

- OS: Windows / Linux
  - Editor: VS Code / Notepad++
  - Browser: Chrome / Firefox
- 

### **Theory**

CSS (Cascading Style Sheets) is used to control presentation of HTML elements such as:

- Colors
- Fonts
- Layout
- Spacing

Types of CSS:

1. **Inline CSS** – applied directly inside an HTML tag
  2. **Internal CSS** – written in `<style>` tag inside `<head>` section
  3. **External CSS** – written in a separate .css file and linked using `<link>` tag
-

## Steps / Procedure

### A) Inline CSS

1. Open **VS Code / Notepad++**.
  2. Create a new file and save as **inline.html**.
  3. Write basic HTML structure.
  4. Apply CSS directly inside tags using the **style=""** attribute:
    - o Example: `style="color: blue; text-align: center;"`
  5. Save the file.
  6. Open **inline.html** in a browser (Chrome/Firefox).
  7. Verify that styles are applied inside individual elements.
- 

### B) Internal CSS

1. Create a new file and save as **internal.html**.
  2. Write basic HTML structure.
  3. Inside `<head>` section, add `<style> ... </style>` block.
  4. Write CSS rules for elements like:
    - o `body, h1, p, button`
  5. Save the file.
  6. Open **internal.html** in browser.
  7. Verify that styles are applied through internal stylesheet.
- 

### C) External CSS

1. Create a new file and save as **external.html**.
2. Create another file and save as **style.css** in the same folder.
3. In **external.html**, link CSS using:
4. `<link rel="stylesheet" href="style.css">`
5. In **style.css**, write the CSS rules for:
  - o `body, h1, p, button`

6. Save both files.
7. Open **external.html** in browser.
8. Confirm webpage formatting is applied from external file.

## **Program Code**

### **A) Inline CSS Example**

**File: inline.html**

```
<!DOCTYPE html>

<html>
  <head>
    <title>Inline CSS Example</title>
  </head>
  <body>
    <h1 style="color: blue; text-align: center;">WTMA Inline CSS</h1>

    <p style="font-size: 18px; color: green;">
      This paragraph is styled using Inline CSS.
    </p>

    <button style="background: orange; padding: 10px; border: none;">
      Click Me
    </button>
  </body>
</html>
```

---

### **B) Internal CSS Example**

**File: internal.html**

```
<!DOCTYPE html>
```

```
<html>  
<head>  
    <title>Internal CSS Example</title>
```

```
    <style>  
        body {  
            font-family: Arial;  
            background-color: #f4f4f4;  
        }
```

```
        h1 {  
            color: darkred;  
            text-align: center;  
        }
```

```
        p {  
            font-size: 18px;  
            color: navy;  
            margin: 20px;  
        }
```

```
        button {  
            background-color: green;  
            color: white;  
            padding: 10px 15px;  
            border: none;  
            border-radius: 5px;  
        }
```

```
</style>

</head>

<body>

<h1>WTMA Internal CSS</h1>

<p>This content is styled using Internal CSS.</p>

<button>Submit</button>

</body>

</html>
```

---

### C) External CSS Example

#### **File 1: external.html**

```
<!DOCTYPE html>

<html>

<head>

<title>External CSS Example</title>

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

</head>

<body>

<h1>WTMA External CSS</h1>

<p>This page uses External CSS file.</p>

<button>Login</button>

</body>

</html>
```

#### **File 2: style.css**

```
body {

font-family: Verdana;

background-color: #e8f0ff;

}
```

```
h1 {  
    color: purple;  
    text-align: center;  
    margin-top: 30px;  
}
```

```
p {  
    font-size: 18px;  
    color: #333;  
    margin: 20px;  
}
```

```
button {  
    background-color: #ff0066;  
    color: white;  
    padding: 10px 20px;  
    border: none;  
    border-radius: 6px;  
}
```

---

## Output

- Inline styles visible directly in HTML tags
  - Internal CSS applied through <style> section
  - External CSS applied through .css file
- 

## Result

Thus, Inline, Internal and External stylesheets were implemented and demonstrated successfully using CSS.

---

## **Program 2: Design a Student Record Web Page Using CSS Keyframe Animation**

---

### **Aim**

To design a Student Record webpage and apply **CSS keyframe animation** for text/card effects.

---

### **Software Required**

- VS Code / Notepad++
  - Browser: Chrome
- 

### **Theory**

**CSS Keyframe Animation** is used to create animations by defining key points of style changes over time.

Syntax:

```
@keyframes animationName {  
    0% {}  
    100% {}  
}
```

---

### **Steps / Procedure**

1. Open **VS Code / Notepad++**.
2. Create a new file and save as **student\_record\_animation.html**.

3. Write the HTML structure for:
  - o Student Record Title
  - o Student details displayed using a <table>
4. Create a container <div> with class **card** to show the student record.
5. Inside <style> tag, apply CSS for:
  - o body (background + center alignment)
  - o card (width, padding, border-radius, shadow)
  - o table (borders + spacing)
6. Create keyframe animations using:
  - o @keyframes slideIn (card comes from top with fade-in)
  - o @keyframes glowText (title color glow effect)
7. Apply animations using:
8. animation: slideIn 2s ease;

and

animation: glowText 2s infinite alternate;

9. Save the file.
10. Run the HTML file in browser.
11. Verify:
  - o Card slides in smoothly
  - o Title text glows continuously

## **Program Code**

**File: student\_record\_animation.html**

```
<!DOCTYPE html>

<html>
<head>
<title>Student Record - Animation</title>
<style>
```

```
body {  
    font-family: Arial, sans-serif;  
    background: #f2f2f2;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
}
```

```
.card {  
    width: 450px;  
    padding: 25px;  
    background: white;  
    border-radius: 12px;  
    box-shadow: 0px 0px 15px rgba(0,0,0,0.2);  
    animation: slideIn 2s ease;  
}
```

```
h2 {  
    text-align: center;  
    color: darkblue;  
    animation: glowText 2s infinite alternate;  
}
```

```
table {  
    width: 100%;  
    margin-top: 15px;  
    border-collapse: collapse;
```

```
}

td {
    padding: 10px;
    border: 1px solid #ccc;
}

.label {
    font-weight: bold;
    background: #e0e0ff;
    width: 40%;
}

/* Keyframe Animations */

@keyframes slideIn {
    from {
        transform: translateY(-60px);
        opacity: 0;
    }
    to {
        transform: translateY(0);
        opacity: 1;
    }
}

@keyframes glowText {
    from {
        color: darkblue;
    }
}
```

```
text-shadow: 0 0 5px skyblue;  
}  
  
to {  
color: red;  
text-shadow: 0 0 10px orange;  
}  
  
}  
  
</style>  
</head>
```

```
<body>  
<div class="card">  
  <h2>Student Record</h2>  
  
<table>  
  <tr>  
    <td class="label">Student Name</td>  
    <td>Rahul Sharma</td>  
  </tr>  
  <tr>  
    <td class="label">Roll Number</td>  
    <td>CS101</td>  
  </tr>  
  <tr>  
    <td class="label">Course</td>  
    <td>B.Tech CSE</td>  
  </tr>
```

```
<tr>
  <td class="label">Semester</td>
  <td>4</td>
</tr>
<tr>
  <td class="label">Email</td>
  <td>rahul@gmail.com</td>
</tr>
<tr>
  <td class="label">Phone</td>
  <td>9876543210</td>
</tr>
</table>
</div>
</body>
</html>
```

---

## Output

## Student Record

<b>Student Name</b>	Rahul Sharma
<b>Roll Number</b>	CS101
<b>Course</b>	B.Tech CSE
<b>Semester</b>	4
<b>Email</b>	rahul@gmail.com
<b>Phone</b>	9876543210

---

### Result

Thus, a Student Record Web Page was designed and CSS keyframe animation was implemented successfully.