Manav Rachna International Institute of Research and Studies



School of computer Applications

Web Technologies(4.5CA152C01)

Assignment

|  |  |
| --- | --- |
| Submitted By | |
| Student Name | Mayank Aggarwal |
| Roll NO | 24/SCA/BCA/024 |
| Program | Bachelor of Computer Application |
| Semester | 1st Semester |
| Department | Computer Application |
| Section | 1A |
| Batch | 2024-25 |

Q1: What does CSS stand for? Name three types of CSS. List three common properties used in CSS for styling text.

1. CSS Definition:

CSS, or Cascading Style Sheets, is a language that styles an HTML webpage. It helps control how things look on the page, like the layout, colors, fonts, and overall design, making websites look nice and consistent across different devices.

2. Types of CSS:

Inline CSS: This type of CSS is added directly to HTML elements using the style attribute. It’s written inside the opening tag of an element, so it’s good for quick styling of single items. However, it can be hard to manage in bigger documents.

Example: <h1 style="color: blue;">Hello World</h1>

Internal CSS: This style is written inside a <style> tag in the <head> section of the HTML file. Internal CSS adds styles to the whole document, but it only works within that single HTML file.

Example:

html

<head>

<style>

h1 { color: green; }

</style>

</head>

External CSS: An external CSS file is a separate file that contains styling instructions for a webpage. To use it, you link this CSS file to the HTML document using the <link> tag inside the <head> section. External CSS is helpful for keeping a consistent look across multiple pages, as one stylesheet can style several HTML files at once.

Example:

html

<head>

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

</head>

In `styles.css`: `h1 { color: red; }`

3. Common Properties for Styling Text in CSS:

Color: This property sets the color of the text.

Example: color: blue;

Font-size: This property sets the size of the text. It can be specified in pixels (`px`), em units, percentages, or other CSS units.

Example: font-size: 18px;

Text-align: This property aligns the text within its container. It can be set to values such as `left`, `right`, `center`, or `justify`.

Example: text-align: center;

Q2: Explain the difference between internal, external, and inline CSS.

1. Internal CSS:

Definition: Internal CSS is when you put the CSS code directly in the HTML file, inside a <style> tag in the <head> section.

Use Case: It’s handy for adding unique styles to one specific webpage that you don’t plan to use anywhere else.

Advantages:

* Keeps the styling in one place within the HTML file.
* Good for quick custom styles specific to that page.

Disadvantages:

* Can’t be reused on other pages, so if you need the same style on multiple pages, you’d have to copy the code each time.

Example:

html

<head>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

</style>

</head>

2. External CSS:

Definition: External CSS is stored in its own file (with a .css extension) and linked to an HTML document using the <link> tag inside the <head> section.

Use Case: Great for big websites where you want a consistent look on many pages.

Advantages:

* Keeps the HTML file focused on content, while CSS handles styling.
* You can link the same CSS file to multiple pages, making it easy to update styles in one place.
* Makes HTML files smaller and, if cached, can help pages load faster.

Disadvantages:

* Needs an extra file, which must load before styles show up.
* If the CSS file is big, it can slightly slow down the first page load.

Example:

html

<head>

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

</head>

- In `styles.css` file:

css

body {background-color: #ffffff;

font-size: 16px; }

3. Inline CSS:

Definition: Inline CSS is when you add styles directly to an HTML element by using the style attribute in the element’s opening tag.

Use Case: It’s useful when you want to quickly style a single element with unique settings, especially for small or one-time changes. Advantages:

* Quick to use for small tweaks.
* Lets you style one element without affecting others.

Disadvantages:

* Makes the HTML code harder to read and organize.
* Can lead to repetitive code if you need the same styles for multiple elements.
* Makes the HTML file bigger in size.Example:

html

<p style="color: blue; font-weight: bold;">This is an inline-styled paragraph.</p>

Q3: How does the position: absolute differ from position: relative?

**1. position: relative:**

Definition: When an element has a `relative` position, it is placed based on where it would normally be on the page.

Behavior: The element stays in its usual spot in the layout, so it still takes up space as usual.

If you use `top`, `right`, `bottom`, or `left` to move it, it will shift \*from\* its normal spot, but other elements around it won’t be affected since it still holds its original space.

Use Case: This is useful when you want to nudge an element a little without changing the positions of nearby elements.

Example:

html

<div style="position: relative; top: 20px; left: 10px;">

This is positioned 20px down and 10px right of its normal position.

</div>

**2. position: absolute:**

Definition: When an element has position: absolute, it is taken out of the regular page layout, so it won’t affect the positioning of other elements around it. It is positioned relative to its nearest ancestor element that has a position set (like relative, absolute, or fixed). If no such ancestor is found, it will be positioned relative to the main <body> or the whole page.

Behaviour:Since it’s out of the layout flow, other elements act as if it isn’t there. You can use properties like top, right, bottom, and left to control where it appears. It can overlap other elements and be placed exactly where you want it.

Use Case: absolute positioning is helpful for things like floating elements, pop-ups, overlays, and elements you want to place precisely within a container.

Example:

html

<div style="position: relative;">

<div style="position: absolute; top: 0; left: 0;">

This is positioned at the top-left corner of its parent container.

</div>

</div>

Q4: Describe the difference between id and class selectors in CSS.

1. id Selector:

Definition: The id selector in CSS is used to style one unique element on a webpage. Each id should be different from all others in the HTML document to avoid conflicts.

Syntax: An `id` selector is defined with a `#` symbol followed by the `id` name.

Usage: The `id` attribute is ideal for applying specific styles to one unique element on the page, such as a header, footer, or a specific section that requires unique styling.

Specificity: The `id` selector has a high specificity, which means it overrides styles applied by classes or tag selectors.

Example:

html

<div id="header">This is the header</div>

css

#header {

background-color: lightblue;

font-size: 24px;

}

2. class Selector:

Definition: The class selector is a way to style many elements at once. You can use it on as many elements as you want on a webpage, making it easy to reuse styles.

Syntax: A `class` selector is defined with a `.` symbol followed by the class name.

Usage: Classes help you apply the same styles to different elements, like buttons or sections, ensuring they look consistent if they serve similar purposes.

Specificity: The class selector has lower priority than an id selector. This means that if both are used on the same element, the styles from the id will take precedence over those from the class.

Example:

html

<div class="content">This is content 1</div>

<div class="content">This is content 2</div>

css

.content {

color: black;

padding: 10px;

border: 1px solid gray;

}

Q5: Use CSS to create a hover effect that changes the text color of a button when a user hovers over it.

Code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Button Hover Effect</title>

<style>

.hover-button {

background-color: #4CAF50; /\* Green background \*/

color: white; /\* Initial text color \*/

padding: 10px 20px;

border: none;

cursor: pointer;

font-size: 16px;

border-radius: 5px;

transition: color 0.3s ease; /\* Smooth transition \*/

}

.hover-button:hover {

color: yellow; /\* Change text color on hover \*/

}

</style>

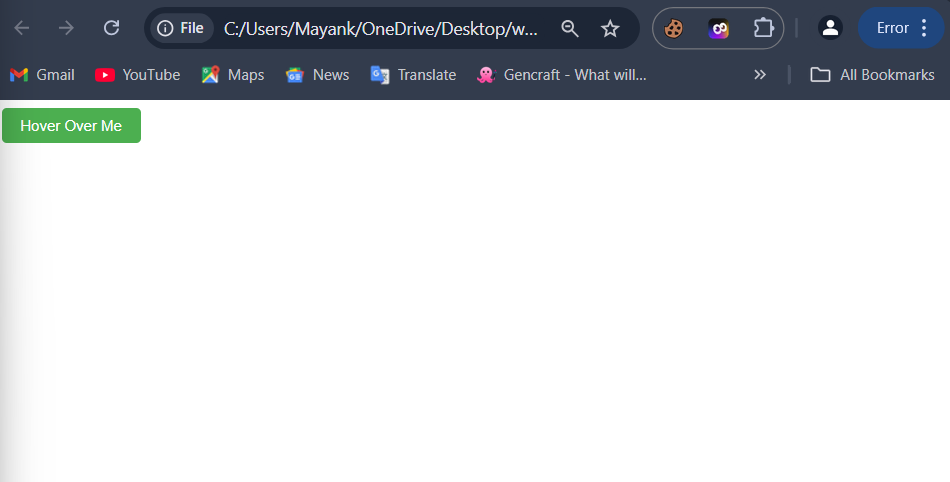
</head>

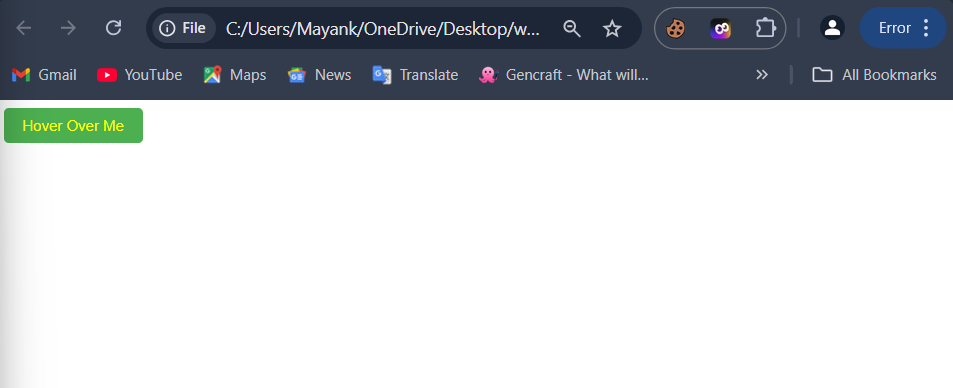
<body>

<button class="hover-button">Hover Over Me</button>

</body>

</html>





Q6: Style a form using CSS so that the input fields have a specific width, border color, and padding.

Code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Styled Form</title>

    <style>

        .form-container {

            max-width: 400px;

            margin: 0 auto;

            padding: 20px;

            background-color: #f9f9f9;

            border-radius: 8px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

        }

        .form-input {

            width: 100%;

            padding: 10px;

            margin-bottom: 15px;

            border: 2px solid #4CAF50;

            border-radius: 5px;

            font-size: 16px;

        }

        .submit-button {

            width: 100%;

            padding: 10px;

            background-color: #4CAF50;

            color: white;

            border: none;

            font-size: 16px;

            cursor: pointer;

            border-radius: 5px;

        }

        .submit-button:hover {

            background-color: #45a049;

        }

</style>

</head>

<body>

<div class="form-container">

<form>

<label for="name">Name:</label>

<input type="text" id="name" class="form-input" placeholder="Enter your name">

<label for="email">Email:</label>

<input type="email" id="email" class="form-input" placeholder="Enter your email">

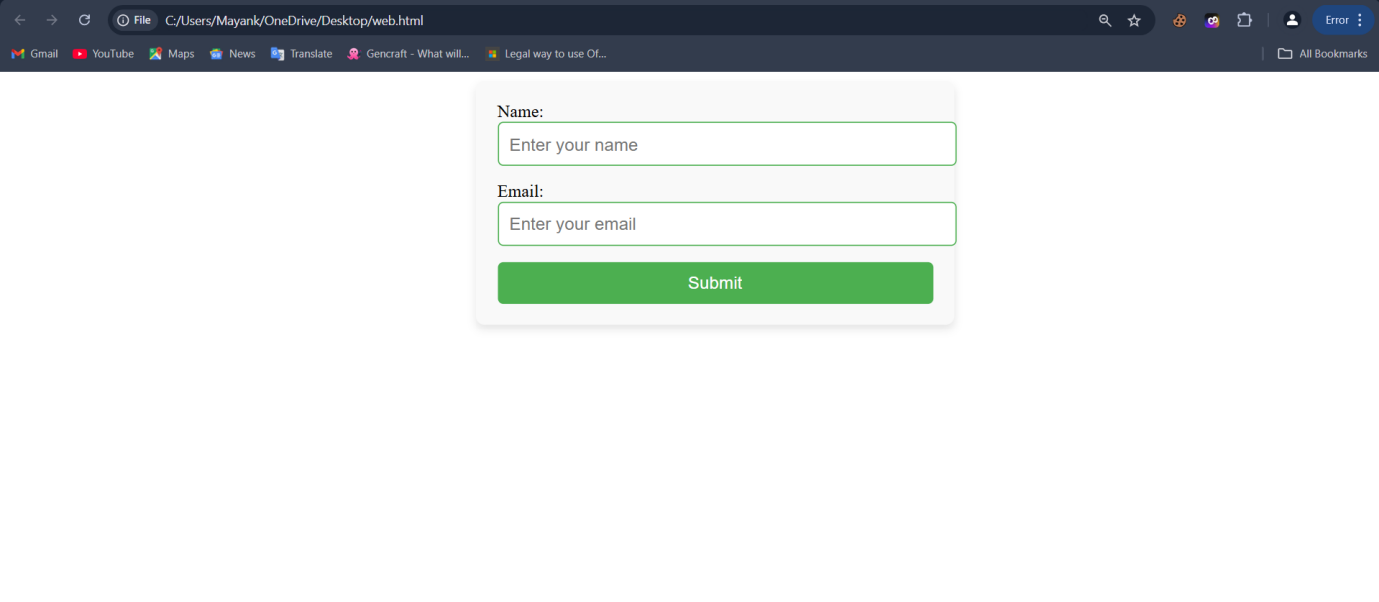
<button type="submit" class="submit-button">Submit</button>

</form>

</div>

</body>

</html>



Q7: Break down the structure of a CSS rule and explain each part of it.

A CSS rule consists of two main parts: the selector and the declaration block. The declaration block is further divided into properties and values.

Structure of a CSS Rule

css

selector {

property: value;

}

Example CSS Rule

css

p {

color: blue;

font-size: 16px;

}

Breakdown of Each Part

**1. Selector:**

**Definition:** The selector tells the browser which HTML element(s) to apply a style rule to.

**Example:** In the example above, p is the selector, which means that the rule affects all <p> (paragraph) elements in the HTML document.

**Types of Selectors:** There are different types of selectors in CSS, including:

* **Class selectors:** Start with a dot (.) and apply styles to elements with a specific class name (e.g., .className).
* **ID selectors:** Start with a hashtag (#) and apply styles to a specific element with that ID (e.g., #idName).
* **Element selectors:** Target specific HTML elements (e.g., div or h1).
* **More complex selectors:** Include combinations of these, like selecting elements that are inside other elements (descendant selectors) or elements in certain states (pseudo-class selectors).

**2. Declaration Block:**

**Definition:** The declaration block is where you write the styles for an element. It can have one or more style rules and is surrounded by curly braces {}.

3. Declaration:

A declaration tells the web browser how to style an element. It has two parts: a property and a value. They are separated by a colon (:) and end with a semicolon (;). For example, if you want to change the text color and size, you would write those as declarations.

4. Property:

Definition: A property is a part of CSS that describes what part of the element you want to change. Examples of properties include color, font-size, and margin.

Example: In the example, `color` and `font-size` are properties.

5. Value:

Definition: The value tells the browser what specific setting to use for the property. For example, if you set the property color to blue, then blue is the value. Similarly, if you set font-size to 16px, then 16px is the value for that property.

Q8: Construct a CSS stylesheet for a form that uses advanced selectors, pseudo-classes, and media queries.

Code:

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Advanced Form</title>

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

</head>

<body>

<form class="signup-form">

<h2>Sign Up</h2>

<label for="name">Name:</label>

<input type="text" id="name" name="name" placeholder="Enter your name" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="Enter your email" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" placeholder="Enter your password" required>

<label for="confirm-password">Confirm Password:</label>

<input type="password" id="confirm-password" name="confirm-password" placeholder="Confirm your password" required>

<button type="submit">Submit</button>

</form>

</body>

</html>

CSS:

body, html {

    height: 100%;

    font-family: Arial, sans-serif;

    background-color: #f3f4f6;

    display: flex;

    align-items: center;

    justify-content: center;

}

.signup-form {

    background-color: #ffffff;

    padding: 30px;

    max-width: 400px;

    width: 100%;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

.signup-form h2 {

    text-align: center;

    color: #333333;

    margin-bottom: 20px;

}

.signup-form label,

.signup-form input[type="text"],

.signup-form input[type="email"],

.signup-form input[type="password"] {

    display: block;

    width: 100%;

}

.signup-form label {

    font-weight: bold;

    color: #333333;

    margin-top: 15px;

    margin-bottom: 5px;

}

.signup-form input[type="text"],

.signup-form input[type="email"],

.signup-form input[type="password"] {

    padding: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;

    margin-bottom: 15px;

    transition: border-color 0.3s;

}

.signup-form input[type="text"]:focus,

.signup-form input[type="email"]:focus,

.signup-form input[type="password"]:focus {

    border-color: #007bff;

    outline: none;

}

.signup-form button {

    width: 100%;

    padding: 12px;

    background-color: #007bff;

    color: #ffffff;

    border: none;

    border-radius: 5px;

    font-size: 16px;

    cursor: pointer;

    transition: background-color 0.3s;

}

.signup-form button:hover {

    background-color: #0056b3;

}

.signup-form input::placeholder {

    color: #aaaaaa;

}

.signup-form input:required:invalid {

    border-color: #ff6666;

}

@media (max-width: 480px) {

    .signup-form {

        padding: 20px;

    }

    .signup-form h2 {

        font-size: 1.5em;

    }

    .signup-form button {

        font-size: 0.9em;

    }

}

