MANAV RACHNA INTERNATIONAL INSTITUTE OF RESEARCH AND STUDIES

Name - Ambuj Yadav
Program - Bachelor of Computer Applications
Department - School of Computer Applications
Roll No. - 24/SCA/BCA/001
TRN No. - 12404401N032

COURSE NAME: Web Technologies
(4.5CA152C01)

ASSIGNMENT: II

Q1: What does CSS stand for? Name three types of CSS. List three common properties used in CSS for styling text. L-1

Ans : CSS, or Cascading Style Sheets, is a stylesheet language which is used for describes the presentation of HTML documents. It allows developers to separate content from design, making web pages more flexible and easier to maintain.

Three Types of CSS :

1 Inline CSS

2. Internal CSS

3. External CSS

Three Common CSS properties for styling text :

1. ***Color***

2. Font-Size

3.Font-weight

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

Ans : 1.**Internal CSS :** Internal CSS is defined within a `<style>` tag in the `<head>` section of an
HTML document.

Example

```
<head>
  <style>
    body {
      background-color: lightblue;
    }
  </style>
</head>
```

2.**External CSS** : External CSS is written in a separate `.css` file, which is linked to the HTML document using a `<link>` tag.

Example :

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

Contents of Styles.CSS :

```
body {
  background-color: lightblue;
}
```

3.Inline CSS : Inline CSS is applied directly within an HTML element the 'style' attribute.

Example :

```
<body>
    <p style="color: blue; font-size: 18px;">
        This is a paragraph.</p>
</body>
```

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

Ans: The position: absolute and position:relative properties in CSS control how elements are positioned on a page, but they behave differently in terms of reference points and movement.

Postion: **absolute** - An element with `position: absolute` is positioned relative to the nearest positioned ancestor (i.e., an ancestor with a position other than `static`). If no such ancestor exists, it is positioned relative to the initial containing block (usually the `<html>` element).

•**Impact on Layout**: The element is removed from the document flow, meaning it does take up any space in the layout, and other elements will behave as if it does not exist.

•**Use *Case***: Ideal for creating overlays, tooltips, or any element that should be precisely placed regardless of the surrounding elements.

Example:

```
css

.absolute {

 position: absolute;

 top: 20px; /* Positions the element 20px from the top of the

nearest positioned ancestor

*/

 left: 30px; /* Positions the element 30px from the left */

}
```

Position: **relative** - An element with `position: relative` is positioned relative to its normal position in the document flow.
•**Impact on Layout**: The space for the element is still preserved in the layout. It can be moved using the `top`, `right`, `bottom`, or `left` properties, but this movement is offset from where it would normally be placed.
•**Use Case**: Often used to fine-tune the position of elements while maintaining their place in the flow of the document.

Example:

```
css
.relative {
position: relative;
 top: 10px; /* Moves the element down by 10px from its original position */
}
```

Q4: Describe the difference between id and class selectors in CSS.
Ans: In CSS, 'I'd and 'class' selectors are both used to style HTML elements, but they differ in purpose and usage :
In CSS, id and class selectors are both used to style HTML elements, but they differ in purpose and usage:

1. Uniqueness:

id Selector: Targets a single, unique element on a page. Each id should be unique across the entire HTML document.

class Selector: Can be applied to multiple elements. Multiple elements can share the same class to apply the same style.

2. Syntax:

id Selector: Prefixed with a #. Example: #header { color: blue; }

class Selector: Prefixed with a .. Example: .button { color: green; }

3. Specificity:

id Selector: Has higher specificity, so it overrides styles from class selectors if both are applied to the same element.

class Selector: Lower specificity compared to id, meaning it can be overridden by an id selector.

4. Use Cases:

id Selector: Best for styling unique elements, like a single navigation bar, header, or footer.

class Selector: Ideal for reusable styles, like buttons, form fields, or any group of elements that share the same look.

Summary :

In general, class selectors are more flexible and are commonly used for reusable styles, while id selectors are reserved for unique, one-off styling.

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

Ans: You can create a hover effect that changes the text color of a button using CSS. Here's a **simple example:**

```
 /* Initial button style */
button {
 color: black;      /* Initial text color */
 background-color: lightgray;
 border: none;
 padding: 10px 20px;
 cursor: pointer;
}

/* Hover effect */
button:hover {
 color: white;      /* Text color on hover */
 background-color: darkgray; /* Optional background color change */
}
```

Explanation :

button:hover: This selector applies styles when the user hovers over the button.

color: white;: This changes the text color to white when hovering.

You can customize the colors and other styles to suit your design. This effect works with any clickable element, not just <button> tags.

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

Ans :

IN Html :

```
<!DOCTYPE html>
<html lang="en">
<body>
 <form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <input type="submit" value="Submit">
 </form>
</body>
</html>
```

IN CSS :

```
    /* styles.css */
form {
 width: 300px;
 margin: 0 auto;
}
input[type="text"],
input[type="email"] {
 width: 100%;        /* Makes the input fields take full width of the form */
 padding: 10px;      /* Adds padding inside the input fields */
```

```
 border: 2px solid #4CAF50; /* Sets a specific border color */
 border-radius: 4px;   /* Adds rounded corners */
 margin-bottom: 15px;  /* Adds space below each input */
 box-sizing: border-box; /* Ensures padding doesn't affect overall width */
}
input[type="submit"] {
 background-color: #4CAF50;
 color: white;
 padding: 10px;
 border: none;
 border-radius: 4px;
 cursor: pointer;
}
input[type="submit"]:hover {
 background-color: #45a049;
}
```

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

Ans : A CSS rule is made up of two main parts: the selector and the declaration block. Each part has a specific role in defining the style to be applied to an HTML element or group of elements. Here's the breakdown:

1. Selector

The selector determines which HTML element(s) the CSS rule will apply to. It can be as simple as an HTML tag (like p for paragraphs), a class (.classname), an ID (#idname), or a more complex combination (like div > p to select paragraphs directly inside a div).

Example: p { ... } or .container { ... } — the selector here specifies the target element(s).

2. Declaration Block

The declaration block is enclosed in curly braces { ... } and contains one or more declarations. Each declaration defines a property and a value that will be applied to the selected element(s).

Property: The property is the specific style attribute you want to modify, like color, font-size, margin, etc.

Value: The value is the setting for the property. For example, if the property is color, the value could be red or #ff0000.

Each declaration is written as property: value; with a semicolon (;) separating multiple declarations within the block.

Example:

```
p {
   color: blue;
   font-size: 16px;
}
```

In this example:

color is a property, and blue is its value.

font-size is a property, and 16px is its value.

Complete Example

Here's a complete CSS rule:

```
h1 {
   color: green;
   text-align: center;
   margin: 20px;
}
```

In this rule:

h1 is the selector, targeting all <h1> elements.

• The declaration block includes :
* color:green;(change text color to green)
* text-align:center;(centers the text)
*margin:20px; (adds space around the element)

Together, the selector and declaration block from a CSS rule that defines styling for specified elements in the HTML document.

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

Ans : Here is a CSS stylesheet for a responsive and accessible form. It uses advanced CSS selectors, pseudo-classes, pseudo-elements, and media queries to create a modern, visually appealing form.

```
/* Basic reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body
{
  font-family: Arial, sans-serif;
  background-color: #f0f0f5;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
}

/* Form container styling */
form
  {
  width: 100%;
  max-width: 400px;
  background-color: #ffffff;
  padding: 2rem;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
```

```
/* Label styling with focus effects */
label {
  font-weight: bold;
  display: block;
  margin-bottom: 0.5rem;
}
input[type="text"],
input[type="email"],
input[type="password"],
textarea {
  width: 100%;
  padding: 0.8rem;
  margin-bottom: 1rem;
  border: 1px solid #ccc;
  border-radius: 4px;
  transition: border-color 0.3s ease;
}
input[type="text"]:focus,
input[type="email"]:focus,
input[type="password"]:focus,
textarea:focus {
  border-color: #0073e6;
outline: none;
}
/* Pseudo-class for required fields */
input:required {
 border-left: 4px solid #0073e6;
 padding-left: calc(0.8rem - 4px);
}
```

```
/* Placeholder styling */
input::placeholder,
textarea::placeholder {
 color: #999;
}
/* Pseudo-class for invalid fields */
input
textarea:
 border-radius: 4px;
 margin-bottom: 1rem;}
.message.success {
 color: #2a9d8f;
 background-col
 font-weight: normal;
}
input[type="checkbox"]:checked + label,
input[type="radio"]:checked + label {
 color: #0073e6;
 font-weight: bold;
}
/* Button styling with*/
{ background-color: #0073e6;
border: none;
 color: #ffffff;
 font-size: 1rem;
 font-weight: bold;
 cursor: pointer;
 border-radius: 4px;
 transition: background-color 0.3s ease;
}
```

```
button:hover {
 background-color: #005bb5;
}
button:active {
 background-color: #004494;
}
/* Styling for specific fieldset and legend */
fieldset {
 border: 1px solid #ccc;
 padding: 1rem;
 margin-bottom: 1rem;
 border-radius: 4px;
}
legend {
 font-weight: bold;
 font-size: 1.2rem;
}
/* Responsive design for smaller screens */
@media (max-width: 480px) {
 form {
 padding: 1.5rem;
 }
```

```
button {
 font-size: 0.9rem;
 }

 input[type="text"],
 input[type="email"],
 input[type="password"],
 textarea {
 padding: 0.6rem;
 }
}
```

Explanation

Advanced Selectors:

input[type="checkbox"] + label and input[type="radio"] + label style labels following checkboxes or radio buttons.

input:required applies a left border for required fields.

Pseudo-classes:

•focus styles form elements when focused.

•invalid and :valid provide feedback on validation status

Pseudo-elements:

•placeholder styles the placeholder text for inputs and text areas.

Media Queries:

For screens below 480px, font size, padding, and form spacing adjust for readability and a comfortable mobile experience.