UNIT III

Cascading Style Sheets : CSS Basics: CSS Rule, Applying CSS Rules (Selectors), Embedding CSS code in HTML page Inline,internal,external style sheets. **CSS Properties:** Font, Color, Types of CSS Color values, Background, CSS Box Model, Display properties, Styling Pseudo Elements, Positioning properties, Layering, Styling Lists and tables.

CSS:-

CSS(cascading style sheets) is a style sheet language used to describe the presentation and appearance of web pages written in HTML. It is used to control the layout, colors, fonts, spacing, and overall design of a website.

Using CSS, developers can apply styles to multiple web pages at once, which saves time and ensures consistent design across a website. CSS also helps in creating responsive web pages that adjust properly on different devices like computers, tablets, and mobile phones.

CSS works by selecting HTML elements and applying style rules to them using selectors, properties and values.

Role of web designing:-

Web designing refers to the process of creating and designing the layout, structure and visual appearance of websites. It focuses on making website attractive, user-friendly and easy to navigate. CSS plays a very important role in web designing because it controls the style, layout, colors, fonts, spacing and responsiveness of web pages. In web development HTML is used to create the structure, while to design and style the webpage.

1. improving visual appearance :-

CSS helps improve the visual appearance of web pages by adding colors, fonts, backgrounds and styles. Without CSS, web pages look plain and simple. With CSS headings , paragraphs, and buttons can be styled to look attractive.

Ex:-

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

2. controlling Layout of web pages:-

CSS helps organize different sections of a webpage such as header, navigation menu, content area, and footer. It allows developers to control the size, position, and spacing of elements. Proper layout makes the website look structured and easy to navigate. It also helps display content neatly on the screen.

Ex:-

```
div{
   width: 300px;
```

```
  height: 200px;
  border: 2px solid black;
}
```

3. Maintaining Consistency

CSS allows the same design to be applied to **multiple web pages of a website**. This ensures that fonts, colors, and layouts remain the same across all pages. Consistency makes the website look **professional and well organized**. It also saves time because styles can be reused.

Example:

```
p{
  font-family: Arial;
  font-size: 18px;
}
```

4. Separating Content from Design

One of the important roles of CSS is separating **HTML content from design and styling**. HTML is used to create the structure of a webpage, while CSS is used for design. This separation makes the code **clean, easy to understand, and easier to maintain**. Developers can change the design without modifying the HTML content.

Example:

HTML

```
<h1>Welcome</h1>
```

What is this?

CSS

```
h1{
  color: green;
}
```

5. Creating Responsive Design

CSS helps create **responsive websites** that work well on different devices like **mobile phones, tablets, and computers**. Developers can adjust font sizes, layouts, and images according to screen size. Responsive design improves the **usability and accessibility** of websites. It ensures that the website looks good on all devices.

Types of CSS:-

1.inline CSS
2.Internal CSS
3. External CSS

1.inline CSS:-

Inline CSS is applied **directly inside an HTML element** using the **style attribute**. It affects only the specific element where it is written.
Inline CSS is useful for **quick styling of a single element**, but it is not recommended for large websites because it makes the code difficult to maintain.

Ex:-

```
<body>
<h1 style=”Background-color:Blue, color:white, font-size:20px”> Hello world </h1>
</body>
```

2. Internal CSS:-

Internal CSS is written inside the **<style> tag** in the **<head> section** of an HTML document. It is used to style **multiple elements on the same webpage**.
Internal CSS is useful when a **single webpage requires a unique style**.

Ex;-

```
<html>
<head>
<style>
H1{
Color:red;
}
P{
Color:blue;
Font-size: 20px;
}
</style>
<body>
<h1> welcome </h1>
<p< this is CSS </p>
</body>
</html>
```

3. External CSS:-

External CSS is written in a **separate file with the extension .css**. This file is linked to the HTML document using the **<link> tag**.
External CSS is the **most recommended method** because it allows the same styles to be used across **multiple web pages**.

Test.html

```
<head>
<link rel=”stylesheet” href=”style.css”>
</head>
```

Style.css

```
H1{
Color: red;
}
P{
Color: green;
Font-size: 20px;
}
```

CSS Rule:-

A CSS Rule is a statement used in cascading style sheets to apply styles to specific HTML elements on a web page. It tells the browser which element to select and what style should be applied to it. CSS rules help in controlling the appearance of web pages such as colors, fonts, spacing, layout and alignment.

A CSS rule consists of two main parts: Selector and Declaration Block.

1. structure of a CSS rule :-

The general syntax of CSS rule is:

Ex:-

```
selector {
  property: value;
}
```

Here, the **selector** identifies the HTML element to be styled, and the **declaration block** contains one or more declarations that define the style.

2. Components of a CSS Rule

A CSS rule contains the following components:

1. Selector

A **selector** specifies the HTML element that you want to style.

Example:

h1

This selector targets all <h1> headings in the webpage.

2. Declaration Block

The declaration block is enclosed inside **curly braces { }** and contains one or more declarations.

Example:

{
  color: red;
  font-size: 30px;
}
Each declaration defines a style property and its value.

3. Property

A **property** defines the aspect of the element you want to change.
Examples of properties:

- color
- font-size
- background-color
- text-align
- margin
- padding

Example:
color

4. Value

A **value** specifies the setting for the property.
Example:
red
So the declaration becomes:
color: red;

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Rule Example</title>

<style>
body {
  background-color: lightyellow;
}

h1 {
  color: red;
  text-align: center;
  font-size: 40px;
}
```

```
p {
  color: blue;
  font-size: 18px;
}

button {
  background-color: green;
  color: white;
  padding: 10px;
}
</style>

</head>

<body>

<h1>Welcome to CSS</h1>

<p>This is a paragraph showing how CSS rules style HTML elements.</p>

<p>CSS makes webpages colorful and attractive.</p>
<button>Click Here</button>
</body>
</html>
```

A CSS rule is used to apply styles to HTML elements using selectors and declarations. In the above example, CSS rules are applied to body, h1, p, and button elements to control the appearance of the webpage.

CSS Selectors

In CSS, selectors are used to identify and select the HTML elements that we want to apply styles to in a webpage created with HTML.

A CSS selector tells the browser which elements should receive the styling rules defined in CSS. After selecting the element, different CSS properties and values can be applied to change its appearance such as color, font size, background, spacing, and layout.

Types of CSS selectors:-

1. Element Selector (Tag Selector):-

The **element selector** selects all HTML elements with a specific **tag name**. It applies the same style to every element of that type on the webpage. For example, if we use the p selector, all paragraph elements will receive the same style. This selector is simple and commonly used for basic styling. It is useful when we want to apply a style to all elements of the same type.

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1{
color:red;
}
p{
color:blue;
font-size:18px;
}
</style>
</head>
<body>
<h1>Welcome to CSS</h1>
<p>This is paragraph one.</p>
<p>This is paragraph two.</p>
</body>
</html>
```

2. ID Selector:-

The **ID selector** is used to select a specific element based on its **unique id attribute**. It is written using the **# symbol** followed by the id name. Since an id is unique, this selector styles only one element in the webpage. It is useful when a particular element needs special styling different from others.

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
#header{
color:green;
text-align:center;
}
#content{
font-size:20px;
color:blue;
}
</style>
</head>
<body>
<h1 id="header">My Website</h1>
<p id="content">Welcome to CSS tutorial.</p>
</body>
</html>
```

3. Class Selector:-

A **Class Selector** is used to apply styles to **one or more HTML elements that share the same class name** in a webpage created with HTML.

Unlike the **ID selector**, a **class selector can be used for multiple elements**. This makes it very useful when we want to apply the **same style to many elements** in a webpage.

In CSS, the **dot symbol (.)** is used to represent a class selector.

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>
.text{
color:blue;
font-size:18px;
}
</style>
</head>
<body>
<p class="text">This is paragraph one.</p>
<p class="text">This is paragraph two.</p>
</body
</html>
```

4. Universal Selectors:-

The **universal selector** selects **all elements** in the HTML document. It is represented by the **asterisk (*) symbol**. This selector is commonly used to apply general styles such as removing default margins or padding from all elements. It helps maintain consistency in the webpage design.

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>
*{
font-family:Arial;
}
</style>
</head>
<body>
<h1>Universal Selector Example</h1>
<p>This paragraph uses Arial font.</p>
<h2>This heading also uses Arial font.</h2>
</body>
</html>
```

5. Group Selector:-

A **Group Selector** is used to **apply the same CSS style to multiple HTML elements at the same time** in a webpage created with HTML.

Sometimes different HTML elements need the **same styling**. Instead of writing separate CSS rules for each element, we can **group them together using a comma (,)** and apply the style once.

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p{
color:green;
text-align:center;
}
</style>
```

```
</head>
<body>
<h1>Main Heading</h1>
<h2>Sub Heading</h2>
<p>This is a paragraph example</p>
</body>
</html>
```

CSS Properties:-

CSS properties are used to **style and design HTML elements** on a webpage. Some of the most commonly used CSS properties are **font properties, color properties, and background properties**. These properties control the **appearance and layout** of web page content.

Font Properties in CSS

Font properties in CSS are used to control the appearance, style, size, and formatting of text in a web page. These properties help developers make text more readable, attractive, and properly formatted. By using font properties, we can change the font type, size, weight, style, and spacing of text.

CSS provides several font-related properties to style text elements in HTML.

1. font-family

The **font-family** property is used to specify the **type of font** for text.

Example:

```
p {
  font-family: Arial;
}
```

2. font-size

The **font-size** property is used to set the **size of the text**.

Example:

```
h1 {
  font-size: 30px;
}
```

3. font-style

The **font-style** property is used to make text **normal, italic, or oblique**.

Example:

```
p {
  font-style: italic;
}
```

4. font-weight

The **font-weight** property controls the **thickness (boldness)** of text.

Example:

```
h1 {
  font-weight: bold;
}
```

5. font-variant

The **font-variant** property is used to display text in **small capital letters**.

Example:

```
p {
  font-variant: small-caps;
}
```

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>

h1 {
  font-family: Arial;
  font-size: 36px;
  font-weight: bold;
}

p {
  font-family: Times New Roman;
  font-size: 18px;
  font-style: italic;
}

</style>
</head>

<body>

<h1>Welcome to CSS</h1>
```

```
<p>This paragraph shows the use of CSS font properties.</p>

</body>
</html>
```

2. Color Property in CSS & Types of CSS Color Values

The **color property in CSS** is used to **set the color of text** in HTML elements. It controls the **foreground color** of text content such as headings, paragraphs, links, and other text elements on a web page. By using the color property, web developers can make web pages more **attractive, readable, and visually appealing**.

Ways to Specify Color in CSS

CSS provides several ways to define colors.

1. Color Names

CSS supports many **predefined color names**.

Example:

```
p {
  color: blue;
}
```

Some common color names include:

- red
- blue
- green
- yellow
- black
- white
- orange
- purple

Example:

```
h1 {
  color: green;
}
```

This displays the heading text in **green color**.

2. RGB Values

RGB stands for **Red, Green, and Blue**.

Each value ranges from **0 to 255**.

Syntax:

```
rgb(red, green, blue)
```

Example:

```
p {
  color: rgb(255,0,0);
}
```

Explanation:

- 255 → Red
- 0 → Green
- 0 → Blue

3. HEX (Hexadecimal) Values

HEX color values are represented using **# followed by six hexadecimal digits**.

Syntax:

```
#RRGGBB
```

Example:

```
h1 {
  color: #FF0000;
}
```

Explanation:

- FF → Red
- 00 → Green
- 00 → Blue

4. RGBA Values

RGBA stands for **Red, Green, Blue, and Alpha (transparency)**.

The alpha value ranges from **0 to 1**.

Example:

```
p {
  color: rgba(255,0,0,0.5);
}
```

Explanation:

- Red color
- Transparency level = **0.5 (semi transparent)**

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>

h1 {
  color: red;
```

```
}

p {
  color: rgb(0,0,255);
}

h2 {
  color: #008000;
}

</style>
</head>

<body>

<h1>Welcome to CSS</h1>
<h2>This is a heading</h2>
<p>This paragraph shows different color values in CSS.</p>

</body>
</html>
```

3. Background Property in CSS

Background properties in CSS are used to control the background appearance of HTML elements such as webpages, sections, headings, or paragraphs. These properties allow developers to set background colors, background images, image positions, repetition, and size. By using background properties, we can make web pages more attractive and visually appealing.

1. Background Color

The **background-color** property is used to set the **background color** of an HTML element.

Syntax

```
selector {
  background-color: value;
}
```

Example

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

2. Background Image

The **background-image** property is used to set an **image as the background** of an element.

Syntax

```
selector {
  background-image: url("image.jpg");
}
```

Example

```
body {
  background-image: url("background.jpg");
}
```

3. Background Repeat

By default, background images **repeat both horizontally and vertically**.

The **background-repeat** property controls how the image repeats.

Values

- repeat
- no-repeat
- repeat-x
- repeat-y

Example

```
body {
  background-repeat: no-repeat;
}
```

Explanation:

The background image appears **only once**.

Example:

```
body {
  background-repeat: repeat-x;
}
```

The image repeats **horizontally**.

4. Background Position

The **background-position** property specifies the **position of the background image**.

Example

```
body {
  background-position: center;
}
```

Other values include:

- top

- bottom
- left
- right
- center

Example:

```
body {
  background-position: top right;
}
```

The image appears at the **top right corner**.

5. Background Size

The **background-size** property controls the **size of the background image**.

Example

```
body {
  background-size: cover;
}
```

Explanation:

The image covers the **entire webpage**.

Other values include:

- contain
- auto
- specific width and height

Example:

```
body {
  background-size: 500px 300px;
}
```

6. Background Attachment

The **background-attachment** property specifies whether the background image **scrolls with the page or remains fixed**.

Values

- scroll (default)
- fixed
- local

Example

```
body {
  background-attachment: fixed;
}
```

Explanation:

The background image **remains fixed while scrolling the page**.

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightyellow;
  background-image: url("bg.jpg");
  background-repeat: no-repeat;
  background-position: center;
  background-size: cover;
}
h1 {
  background-color: lightgreen;
}
</style>
</head>
<body>
<h1>Welcome to CSS</h1>
<p>This example demonstrates background properties in CSS.</p>
</body>
</html>
```

Text Properties in CSS (

Text properties in CSS are used to control the appearance, alignment, spacing, and decoration of text in web pages. These properties help improve the readability, formatting, and presentation of text content such as headings, paragraphs, and links. Using text properties, developers can control text alignment, decoration, spacing, case conversion, and indentation.

1. text-align

The text-align property is used to control the horizontal alignment of text within an element.

Values

- left
- right
- center
- justify

Example:-

```
p {
  text-align: center;
}
```

2. text-decoration

The **text-decoration** property is used to **add or remove decorations** such as underline or line-through.

Values

- none
- underline
- overline
- line-through

Example

```
h1 {
  text-decoration: underline;
}
```

3. text-transform

The **text-transform** property is used to **change the case of text**.

Values

- uppercase
- lowercase
- capitalize

Example

```
p {
  text-transform: uppercase;
}
```

4. text-indent

The **text-indent** property is used to specify the **indentation of the first line of text** in a paragraph.

Example

```
p {
  text-indent: 50px;
}
```

Explanation:

The first line of the paragraph starts **50 pixels away from the left margin**.

5. letter-spacing

The **letter-spacing** property controls the **space between characters**.

Example

```
h1 {
  letter-spacing: 5px;
}
```

Explanation:

The space between letters increases by **5 pixels**.

6. word-spacing

The **word-spacing** property controls the **space between words**.

Example

```
p {
  word-spacing: 10px;
}
```

Explanation:

The space between words increases by **10 pixels**.

7. line-height

The **line-height** property controls the **space between lines of text**.

Example

```
p {
  line-height: 25px;
}
```

Ex:-

```
<!DOCTYPE html>
<html>
<head>
<style>

h1 {
  text-align: center;
  text-decoration: underline;
  letter-spacing: 3px;
}

p {
  text-indent: 40px;
  text-transform: capitalize;
  line-height: 25px;
  word-spacing: 5px;
}

</style>
```

```
</head>
<body>
<h1>CSS Text Properties</h1>
<p>this paragraph demonstrates different text properties in css for formatting text content in a webpage.</p>
</body>
</html>
```

Different CSS Display Properties and Their Uses

The display property in CSS is used to control how an HTML element is displayed on a webpage. It determines whether an element behaves like a block element, inline element, or other layout types. By using the display property, developers can control the layout and arrangement of elements in a web page.

The common values of the CSS display property are:

- block
- inline
- inline-block
- none
- flex
- grid

1. display: block

A block element always starts on a new line and takes up the full width available. Block elements create a vertical layout on the webpage.

Examples of block elements:

-
- <p>
- <h1> to <h6>

Example:

```
div{
  display: block;
}
```

Use:

It is used when we want elements to appear one below another and occupy full width.

2. display: inline

Inline elements do not start on a new line. They only take up as much width as necessary and appear in the same line with other elements.

Examples of inline elements:

-

- <a>
- <b>

Example:

```
span{
  display: inline;
}
```

Use:

It is used when we want elements to appear side by side in the same line.

3. display: inline-block

The inline-block value is a combination of inline and block.

Elements appear in the same line, but we can also set width and height like block elements.

Example:

```
div{
  display: inline-block;
  width: 150px;
  height: 100px;
}
```

Use:

It is used when elements should appear side by side but still allow width and height settings.

4. display: none

The none value hides the element completely from the webpage.

Example:

```
p{
  display: none;
}
```

Use:

It is used to hide elements temporarily, such as menus, messages, or buttons.

5. display: flex

The flex display property is used to create flexible layouts where elements can easily align and distribute space.

Example:

```
.container{
  display: flex;
}
```

Use:
It is used to align elements horizontally or vertically and create responsive layouts.

6. display: grid

The grid display property is used to create two-dimensional layouts with rows and columns.

Example:

```
.container{
  display: grid;
}
```

Use:

It is used to design complex layouts like dashboards, image galleries, and website structures.

Example Program

```
<!DOCTYPE html>
<html>
<head>
<style>

div{
  display: inline-block;
  width: 100px;
  height: 100px;
  background-color: lightblue;
}

</style>
</head>
<body>
<div>Box1</div>
<div>Box2</div>
<div>Box3</div>
</body>
</html>
```

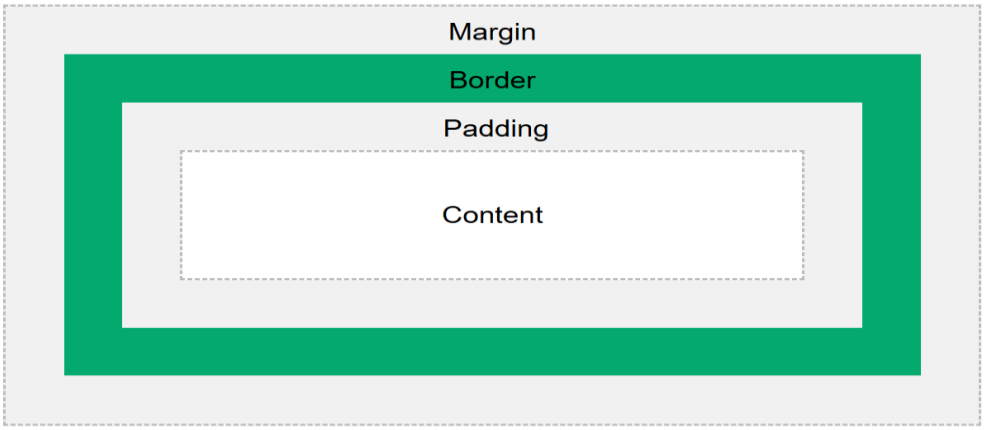
CSS Box Model:-

The **CSS Box Model** is a fundamental concept in web design that describes the **layout and structure of elements in a webpage**. According to the box model, every HTML element is treated as a **rectangular box**. This box consists of different layers that control the **spacing, borders, and size of elements**.

The CSS box model helps developers control the **layout, spacing, and positioning of elements** on a webpage.

The CSS box model has **four main components**:

1. Content
2. Padding
3. Border
4. Margin



1. Content

The **content** is the innermost part of the box where the **actual text, images, or other data** of the element appears. The size of the content area can be controlled using the **width and height properties** in CSS.

Example:

```
p{
  width: 300px;
  height: 100px;
}
```

This defines the **size of the content area**.

2. Padding

Padding is the space between the **content and the border** of an element. It creates space inside the element around the content, which improves readability and design.

Example:

```
p{
  padding: 20px;
}
```
This creates **20 pixels of space inside the element around the content**.

3. Border

The **border** surrounds the padding and content area. It creates a visible boundary around the element and can be styled with different **colors, widths, and styles**.

Example:

```
p{
  border: 2px solid black;
}
```
This creates a **2px solid black border around the element**.

4. Margin

Margin is the outermost part of the box model. It creates space **outside the border** and controls the distance between one element and another element.

Example:

```
p{
  margin: 15px;
}
```
This creates **15 pixels of space outside the element**.

Complete Example:-

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Box Model Example</title>

<style>

.box{
   width: 300px;          /* Content width */
   height: 150px;         /* Content height */
   padding: 20px;         /* Space inside the box */
   border: 5px solid blue;   /* Border around padding */
   margin: 40px;          /* Space outside the box */
   background-color: lightyellow;
}
</style>
```

```
</head>
<body>
<h2>CSS Box Model Example</h2>
<p>The box below shows how margin, border, padding, and content work.</p>
<div class="box">
This is the content area of the box.
Padding creates space inside the box.
Border surrounds the padding.
Margin creates space outside the box.
</div>
</body>
</html>
```

Styling Pseudo Elements in CSS :-

Pseudo-elements in CSS are used to style specific parts of an HTML element. They allow developers to apply styles to parts of the content such as the first letter, first line, or before and after the content without adding extra HTML elements. Pseudo-elements are written using double colon (::) before the element name.

Syntax:

```
selector::pseudo-element{
  property: value;
}
```

Pseudo-elements are useful for improving the design and appearance of text and content on a webpage.

1. ::first-letter

The ::first-letter pseudo-element is used to style the first letter of a text inside a block element.

Example:

```
p::first-letter{
  font-size: 30px;
  color: red;
}
```

Explanation:

The first letter of the paragraph becomes larger and red in color.

2. ::first-line

The ::first-line pseudo-element is used to style the first line of a paragraph.

Example:

```
p::first-line{
  color: blue;
```

```
  font-weight: bold;
}
```
Explanation:
The first line of the paragraph will appear blue and bold.

3. ::before

The ::before pseudo-element is used to insert content before an element.
Example:
```
h2::before{
  content: "★ ";
  color: green;
}
```
Explanation:
A star symbol appears before the heading text.

4. ::after

The ::after pseudo-element is used to insert content after an element.
Example:
```
h2::after{
  content: " ✓";
  color: blue;
}
```
Explanation:
A check symbol appears after the heading.

5. ::selection

The ::selection pseudo-element is used to style the text selected by the user (when highlighting with the mouse).
Example:
```
::selection{
  background-color: yellow;
  color: red;
}
```
Explanation:
When the user selects text, it will appear with a yellow background and red color.

Complete Example

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>

p::first-letter{
  font-size: 30px;
  color: red;
}

p::first-line{
  color: blue;
}

h2::before{
  content: "★ ";
}

h2::after{
  content: " ✓";
}

::selection{
  background-color: yellow;
}

</style>
</head>

<body>

<h2>CSS Pseudo Element Example</h2>

<p>
CSS pseudo-elements allow you to style specific parts of an element
such as the first letter and first line of text.
</p>

</body>
</html>
```

Layering in CSS

Layering in CSS refers to the technique of placing HTML elements on top of each other in different layers on a webpage. It controls how elements overlap and appear in front or behind other elements.

Layering is mainly controlled using the z-index property along with the position property. By using layering, developers can create complex layouts, overlapping images, menus, popups, and other visual effects.

Position Property

To apply layering, elements must use the position property. Some common position values are:

- static – Default position of elements.
- relative – Element is positioned relative to its normal position.
- absolute – Element is positioned relative to its nearest positioned parent.
- fixed – Element stays fixed on the screen even when scrolling.

Example:

```
.box1{
  position: relative;
}
```

z-index Property

The z-index property controls the vertical stacking order of elements.

Elements with higher z-index values appear in front, while elements with lower values appear behind.

Example:

```
.box1{
  position: absolute;
  z-index: 1;
}

.box2{
  position: absolute;
  z-index: 2;
}
```

In this example, box2 appears above box1 because it has a higher z-index value.

Example Program

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>

.box1{
  width:200px;
  height:200px;
  background-color:red;
  position:absolute;
  left:50px;
  top:50px;
  z-index:1;
}

.box2{
  width:200px;
  height:200px;
  background-color:blue;
  position:absolute;
  left:100px;
  top:100px;
  z-index:2;
}

</style>
</head>

<body>
<h2>CSS Layering Example</h2>
<div class="box1"></div>
<div class="box2"></div>
</body>
</html>
```

Explanation

- Two boxes overlap each other.
- box1 has z-index:1.
- box2 has z-index:2.
- Since box2 has a higher z-index, it appears above box1.

Styling Lists and Tables in CSS

CSS allows developers to style lists and tables to improve the appearance and readability of web pages. By using CSS properties, lists and tables can be customized with colors, spacing, borders, alignment, and layout styles.

1. Styling Lists in CSS

Lists in HTML are mainly of two types:

- Ordered List (ol)
- Unordered List (ul)

CSS properties are used to change the list markers, position, color, and spacing.

Important List Styling Properties

1. list-style-type

This property changes the type of marker or bullet used in the list.

Example:

```
ul{
  list-style-type: square;
}
```

Common values:

- disc (default bullet)
- circle
- square
- decimal
- none

2. list-style-position

This property defines the position of list markers.

Example:

```
ul{
  list-style-position: inside;
}
```

Values:

- inside – marker appears inside the list item
- outside – marker appears outside (default)

3. list-style-image

This property allows the use of an image as a list marker.

Example:

```
ul{
  list-style-image: url("star.png");
}
```

This replaces the bullet with an image icon.

Example for Styling Lists

```
<!DOCTYPE html>
<html>
<head>
<style>

ul{
  list-style-type: square;
  color: blue;
}

</style>
</head>

<body>
<h2>Example of Styled List</h2>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JavaScript</li>
</ul>
</body>
</html>
```

2. Styling Tables in CSS

Tables are used to display data in rows and columns. CSS can be used to improve table appearance by adding borders, spacing, colors, and alignment.
Important Table Styling Properties

1. border

This property adds a border around the table and cells.
Example:

```
table, th, td{
  border: 1px solid black;
}
```

2. border-collapse

This property controls whether table borders collapse into a single border or remain separate.
Example:

```
table{
  border-collapse: collapse;
}
```

3. padding

Padding adds space inside table cells.

Example:

```
th, td{
  padding: 10px;
}
```

4. text-align

This property controls the alignment of text inside table cells.

Example:

```
th{
  text-align: center;
}
```

Example for Styling Tables

```
<!DOCTYPE html>
<html>
<head>
<style>

table{
  border-collapse: collapse;
  width: 50%;
}

table, th, td{
  border: 1px solid black;
}

th, td{
  padding: 10px;
  text-align: center;
}
```

5 Marks

1. Define CSS and list its advantages.
2. What is the CSS Box Model?
3. What are pseudo-elements? Give examples.
4. Define CSS selectors.
5. What are CSS color values? List their types.
6. Differentiate between internal and external CSS.

10 Marks

1. Define CSS and Explain its role in web designing with suitable examples.
2. Explain the different types of CSS selectors and their importance in styling web pages.
3. explain the use of background properties in CSS with suitable examples.
4. Explain different CSS display properties and their uses.
5. Explain different ways of applying CSS rules to an HTML document. Describe inline, internal, and external style sheets with suitable examples.