Unit-2: Working with CSS3

1. What is CSS? What are the benefits of CSS? List out the different ways to write CSS.

(3 CO-2)

- **Definition:**
  CSS stands for Cascading Style Sheets. It is a styling language used to describe the presentation of a document written in HTML or XML *(CSS ka full form Cascading Style Sheets hai. Yeh ek styling language hai jo HTML ya XML mein likhe document ke presentation ko describe karta hai).*

- **Use/Purpose:**
  CSS is used to control the layout of web pages, set fonts, colors, margins, padding, etc., and make websites visually appealing and user-friendly *(CSS ka use web pages ke layout ko control karne, fonts, colors, margins, padding, etc. ko set karne ke liye hota hai aur websites ko visually appealing aur user-friendly banata hai).*

- **Syntax:**
  ```
  selector {  property: value;}
  ```

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

- **Benefits of CSS:**
  - **Separation of Content and Style:** It allows you to separate content (HTML) from presentation (CSS), making the website easier to maintain *(Yeh content (HTML) aur presentation (CSS) ko separate karta hai, jo website ko maintain karna asaan bana deta hai).*
  - **Improved Page Load Speed:** Since styles are defined separately, they can be cached by browsers, reducing the loading time *(Jab styles alag se define kiye jaate hain, browsers inhe cache kar lete hain, jis se page ka loading time kam ho jata hai).*
  - **Consistent Design:** Using CSS ensures that design elements like fonts and colors are consistent throughout the website *(CSS ka use ensure karta hai ki design elements jaise fonts aur colors har page pe consistent rahe).*
  - **Responsive Design:** CSS helps create responsive designs that adapt to different screen sizes and devices *(CSS responsive designs banane mein madad karta hai jo alag-alag screen sizes aur devices ke liye adapt ho sakte hain).*
- **Different Ways to Write CSS:**
  - **Inline CSS:** The style is applied directly within the HTML tag using the `style` attribute *(Inline CSS style directly HTML tag ke andar `style` attribute ka use karke apply kiya jata hai).*
    Example

```
<p style="color: red;">This is a red paragraph.</p>
```

- **Internal CSS:** The style is written inside the `<style>` tag in the `<head>` section of the HTML document *(Internal CSS style `<style>` tag ke andar HTML document ke `<head>` section mein likha jata hai).*
  Example:`html`

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

- **External CSS:** The style is written in an external `.css` file, which is linked to the HTML document *(External CSS ek alag `.css` file mein likha jata hai, jo HTML document se link kiya jata hai).*
  Example:`html`

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

- **Real-life Application:**
  - **Web Design:** CSS is widely used to design and layout web pages. For instance, websites like Google, Facebook, and Amazon use CSS for styling their pages *(CSS ka use web pages ko design aur layout karne ke liye hota hai. Jaise Google, Facebook, aur Amazon websites apne pages ko style karne ke liye CSS use karte hain).*
  - **Mobile Apps:** CSS is used for styling mobile web applications, making them responsive and mobile-friendly *(CSS mobile web applications ko style karne ke liye use hota hai, jo unhe responsive aur mobile-friendly banata hai).*

2. List and Explain types of CSS with Example.

(4 CO-2)

- **Definition:**
  There are three types of CSS that determine where the styles are applied: Inline CSS, Internal CSS, and External CSS *(CSS ke teen types hote hain, jo decide karte hain ki styles kaha apply honge: Inline CSS, Internal CSS, aur External CSS).*

- **Use/Purpose:**
  Each type is used in different scenarios based on the requirements of a project. The use of these types ensures flexible and maintainable code *(Har type alag-alag scenarios mein use hota hai project ki requirements ke hisaab se. In types ka use flexible aur maintainable code ensure karta hai).*

- **Types of CSS:**

  1. **Inline CSS:**
     Inline CSS is used when you want to apply styles directly within an HTML element using the `style` attribute *(Inline CSS tab use hota hai jab aap directly ek HTML element ke andar `style` attribute ka use karke styles apply karna chahte ho)*.
     Example:

```
<p style="color: blue;">This paragraph is blue.</p>
```

2.Internal CSS:
Internal CSS is used when you need to style a single HTML document. The CSS code is written inside the `<style>` tag in the `<head>` section of the HTML document *(Internal CSS tab use hota hai jab aapko ek single HTML document ko style karna ho. CSS code `<style>` tag ke andar `<head>` section mein likha jata hai).*
Example:

```
<head>
  <style>
    p {
      font-size: 18px;
    }
  </style>
</head>
<body>
  <p>This paragraph has a font size of 18px.</p>
</body>
```

3.External CSS:
External CSS is used when you want to apply styles across multiple web pages. The styles are written in a separate `.css` file, which is linked to the HTML document *(External CSS tab use hota hai jab aapko styles multiple web pages par apply karne ho. Styles ek alag `.css` file mein likhe jate*

hain, jo HTML document se link kiye jate hain).
Example:

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

- **Real-life Application:**

1. **Inline CSS:** Used when applying a quick style to a single element, like changing the text color of one heading on the page *(Inline CSS ka use jab ek single element ko quickly style karna ho, jaise ek heading ka text color change karna).*
2. **Internal CSS:** Ideal for styling a single web page when the styles are specific to that page *(Internal CSS ka use ek single web page ko style karne ke liye hota hai jab styles uss page tak limited hon).*
3. **External CSS:** Most commonly used for large websites where multiple pages share the same styles *(External CSS ka use large websites mein hota hai jahan multiple pages ko same styles share karni hoti hain).*

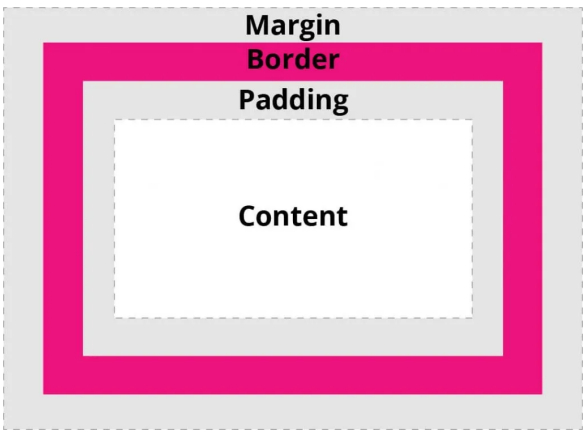
3. Explain CSS box model (border, margin, padding) with neat diagram.

(8 CO-2)

- **Definition:**
The CSS box model is a concept that defines the rectangular boxes generated for elements in the document layout. It consists of four parts: content, padding, border, and margin *(CSS box model ek concept hai jo document layout mein elements ke liye rectangular boxes define karta hai. Isme chaar parts hote hain: content, padding, border, aur margin).*

- **Use/Purpose:**
The box model is essential for controlling the size, spacing, and alignment of elements on a webpage. It helps to design the layout of elements, including their spacing, and ensures that they fit together correctly *(Box model website ke elements ke size, spacing, aur alignment ko control karne ke liye important hai. Yeh elements ke layout ko design karne mein madad karta hai, unke spacing ko adjust karta hai, aur ensure karta hai ki sab elements sahi tarike se fit ho).*

- **Parts of the Box Model:**



1.Content: The actual content of the element, like text or images *(Content woh asli content hota hai element ka, jaise text ya images).*
2.Padding: The space between the content and the border. Padding adds space inside the element *(Padding content aur border ke beech ka space hota hai. Yeh element ke andar space add karta hai).*
3.Border: The border that surrounds the element,

between the padding and margin *(Border woh line hoti hai jo element ke around hoti hai, padding aur margin ke beech).*

4. **Margin:** The space outside the border, which separates the element from others *(Margin woh space hota hai jo border ke bahar hota hai, jo element ko doosre elements se separate karta hai).*

Syntax Example:

```
div {

  width: 200px;
  height: 100px;
  padding: 20px;
  border: 5px solid black;
  margin: 30px;
}
```

- **Real-life Application:**

1. **Web Layouts:** The box model is used in web design for creating layouts, adjusting the space between elements, and ensuring that the content fits properly inside containers *(Box model web design mein layouts banane, elements ke beech space adjust karne, aur content ko containers ke andar fit karne ke liye use hota hai).*
2. **Responsive Design:** Understanding the box model is crucial in creating responsive designs that adjust the content properly on different screen sizes *(Box model ko samajhna responsive designs banane ke liye zaroori hota hai jo content ko alag-alag screen sizes par sahi tarike se adjust kare).*

4. Explain Class and ID Selector in CSS with examples.

(4 CO-2)

- **Definition:**
  The class and ID selectors are used to select elements in HTML and apply styles to them *(Class aur ID selectors ka use HTML mein elements ko select karne aur unpe styles apply karne ke liye hota hai).*

- **Use/Purpose:**

  - **Class Selector:** The class selector is used when you want to apply the same styles to multiple elements *(Class selector tab use hota hai jab aapko same styles multiple elements par apply karne ho).*
  - **ID Selector:** The ID selector is used to apply styles to a single unique element *(ID selector tab use hota hai jab aapko ek unique element par styles apply karne ho).*

- **Syntax Example:**

  - **Class Selector:**
    ```
    css
    ```

```
.example {
  color: red;
}
```

HTML:

```
<div class="example">This is a red text.</div>
<p class="example">This is also red.</p>
```

ID Selector:

```
#unique {
  color: blue;
}
```

HTML:

```
<div id="unique">This is blue text.</div>
```

- **Real-life Application:**
  - **Class Selector:** Class selectors are commonly used for styling groups of elements like buttons, paragraphs, or sections *(Class selectors ka use commonly groups of elements ko style karne ke liye hota hai, jaise buttons, paragraphs, ya sections).*
  - **ID Selector:** ID selectors are often used to style specific elements like a header, footer, or unique elements that need individual styling *(ID selectors ka use specific elements ko style karne ke liye hota hai, jaise header, footer, ya koi unique element jo individual styling chahiye).*

5. What is CSS Display? Explain it with example.

(4 CO-2)

- **Definition:**
  The `display` property in CSS defines the display behavior of an element. It specifies how an element should be displayed in the layout *(CSS mein `display` property element ke display behavior ko define karti hai. Yeh specify karti hai ki ek element ko layout mein kaise display kiya jana chahiye).*

- **Use/Purpose:**
  The `display` property is used to control the layout of elements on the page. It can make an element behave as a block, inline, or other types based on the value set *(Yeh property page par elements ke layout ko control karne ke liye use hoti hai. Yeh element ko block, inline ya kisi aur type ke roop mein set kar sakti hai).*

Syntax:

```
css
element {

  display: value;
}
```

Example:

```
css
```

```
.block-element {

  display: block;
  width: 100%;
  background-color: lightgrey;
}

.inline-element {
  display: inline;
  color: red;
}
```

HTML:

```
html
<div class="block-element">This is a block element.</div>
<span class="inline-element">This is an inline element.</
span>
```

- **Real-life Application:**
  - **Block-level Elements:** Used for elements like paragraphs (`<p>`) and divisions (`<div>`) that need to take up the entire width of the container *(Block-level elements jaise paragraphs (`<p>`) aur divisions (`<div>`) ko use kiya jata hai jo container ka pura width lete hain).*
  - **Inline Elements:** Used for small elements like links (`<a>`) or spans (`<span>`) that should not interrupt the flow of the content *(Inline elements chote elements jaise links (`<a>`) ya spans (`<span>`) ke liye use kiye jate hain jo content ke flow ko interrupt nahi karte).*

6. Explain the use of Media Query in CSS with example.

(4 CO-2)

- **Definition:**
  A media query in CSS is used to apply styles depending on the characteristics of the device (screen size, resolution, orientation, etc.). It allows you to create responsive designs that adapt to different devices *(CSS mein media query ka use styles ko device ki characteristics (screen size, resolution, orientation, etc.) ke hisaab se apply karne ke liye hota hai. Yeh responsive designs banane mein madad karta hai jo alag-alag devices par adjust ho sakte hain).*

- **Use/Purpose:**
  Media queries are used to apply different styles for different screen sizes or devices. For example, you can change the layout for mobile devices compared to desktop *(Media queries ka use different screen sizes ya devices ke liye alag-alag styles apply karne ke liye hota hai. Jaise aap mobile devices ke liye layout ko change kar sakte hain desktop ke comparison mein).*

Syntax:

```
css
@media screen and (max-width: 600px) {

  /* Styles for devices with max-width 600px */
  body {
    background-color: lightblue;
  }
}
```

Example:

```
css
@media screen and (max-width: 768px) {

  /* Change the background color for screens smaller than
768px */
  body {
    background-color: lightyellow;
  }
}
```

- HTML:

```
html
```

```
<p>This background color will change based on screen
size.</p>
```

- **Real-life Application:**
    - **Mobile-First Design:** Media queries are used in mobile-first web design to ensure the website looks good on mobile devices first and then adapts for larger screens *(Mobile-first design mein media queries ka use hota hai jisse website sabse pehle mobile devices pe achha dikhe aur phir larger screens ke liye adjust ho jaye).*
    - **Responsive Layouts:** Media queries are essential for creating flexible layouts that adjust to the size of the screen, such as grids that rearrange themselves on smaller devices *(Responsive layouts banane mein media queries zaroori hote hain jo screen size ke hisaab se adjust hote hain, jaise grids jo chhote devices par apne aap ko rearrange karte hain).*

7. Demonstrate the use of CSS for Colors and Background.

(4 CO-2)

- **Definition:**
  In CSS, colors and backgrounds are used to enhance the appearance of elements by applying color properties to text, borders, and backgrounds *(CSS mein colors aur backgrounds ka use elements ki appearance ko enhance karne ke liye hota hai, jisme text, borders, aur backgrounds pe color properties apply ki jati hain).*

- **Use/Purpose:**
  Color properties are used to change the color of text, borders, and background areas to improve the design and make the content more visually appealing *(Color properties ka use text, borders, aur background areas ka color badalne ke liye hota hai jisse design improve hota hai aur content visually appealing hota hai).*

Syntax:

```
css
```

```
element {
  color: value;  /* For text color */
  background-color: value;  /* For background color */
}
```

Example:

```
css
h1 {
  color: darkblue;  /* Set the text color */
```

```
  background-color: lightyellow;  /* Set the background color
*/
}
```

HTML:

```
html
<h1>This is a heading with color and background.</h1>
```

- **Real-life Application:**
    - **Website Design:** Colors are used in web design to create a mood or theme, such as blue for trustworthiness or green for calmness *(Website design mein colors ka use mood ya theme create karne ke liye hota hai, jaise blue trustworthiness ke liye aur green calmness ke liye).*
    - **Background Images:** CSS allows you to set images as backgrounds, enhancing the visual appeal of sections or entire pages *(CSS background images ko set karne ki suvidha deta hai, jo sections ya poori pages ke visual appeal ko enhance karte hain).*

8. Define viewport, CSS, CSS selector.

(3 CO-2)

- **Definition:**
    - **Viewport:** The viewport is the visible area of the web page that can be seen on the user's screen *(Viewport woh visible area hota hai jo web page ka user ke screen par dikhta hai).*
    - **CSS:** Cascading Style Sheets (CSS) is a styling language used to define the presentation of a web page *(Cascading Style Sheets (CSS) ek styling language hai jo web page ke presentation ko define karta hai).*
    - **CSS Selector:** A CSS selector is used to select HTML elements and apply styles to them *(CSS selector HTML elements ko select karne aur unpe styles apply karne ke liye use hota hai).*
- **Use/Purpose:**
    - **Viewport:** The viewport is crucial for responsive design as it helps to adjust the layout of the page to different screen sizes *(Viewport responsive design ke liye zaroori hai kyunki yeh page ke layout ko alag-alag screen sizes ke hisaab se adjust karne mein madad karta hai).*
    - **CSS:** CSS is used to style the appearance of elements like text, colors, backgrounds, and more *(CSS ka use elements ke appearance ko style karne ke liye hota hai jaise text, colors, backgrounds, aur zyada).*
    - **CSS Selector:** CSS selectors help target specific HTML elements to apply desired styles *(CSS selectors specific HTML elements ko target karne mein madad karte hain jisse desired styles apply kiye ja sake).*
- **Real-life Application:**
    - **Viewport:** In mobile design, the viewport meta tag is used to control the scaling and zooming of the webpage *(Mobile design mein viewport meta tag ka use webpage ke scaling aur zooming ko control karne ke liye hota hai).*

- **CSS:** Every website uses CSS to design layouts, change fonts, set colors, and create visually appealing interfaces *(Har website CSS ka use apne layout ko design karne, fonts ko change karne, colors set karne, aur visually appealing interfaces banane ke liye karti hai).*
- **CSS Selector:** Web designers use selectors to style specific sections or elements on the page *(Web designers selectors ka use specific sections ya elements ko style karne ke liye karte hain).*

9. Enlist the various attributes of font?

(3 CO-2)

- **Definition:**
  Font attributes in CSS define how the text should be displayed. These include font size, family, style, weight, etc. *(CSS mein font attributes define karte hain ki text kaise display hoga. Ismein font size, family, style, weight, aur aur bhi cheezein hoti hain).*

- **Use/Purpose:**
  Font attributes are used to make text readable and visually appealing. These attributes allow you to customize the appearance of text *(Font attributes ka use text ko readable aur visually appealing banane ke liye hota hai. Yeh attributes text ke appearance ko customize karne mein madad karte hain).*

- **Font Attributes:**

  1. **font-family:** Specifies the font of the text *(Yeh attribute text ke font ko specify karta hai).*
  2. **font-size:** Defines the size of the text *(Yeh attribute text ke size ko define karta hai).*
  3. **font-weight:** Controls the thickness of the text *(Yeh attribute text ke thickness ko control karta hai).*
  4. **font-style:** Specifies whether the text should be normal, italic, or oblique *(Yeh attribute specify karta hai ki text normal, italic, ya oblique hona chahiye).*
  5. **line-height:** Sets the amount of space between lines of text *(Yeh attribute text ki lines ke beech ka space set karta hai).*

Example:

```
css
p {
  font-family: Arial, sans-serif;
  font-size: 16px;
  font-weight: bold;
  font-style: italic;
  line-height: 1.5;
}
```

- **Real-life Application:**

  1. **Typography in Websites:** Designers use font attributes to ensure that the text is legible and matches the style of the website *(Designers font attributes ka use karte hain taaki text readable ho aur website ke style ke saath match kare).*
  2. **Branding:** Companies use specific fonts to maintain a brand's identity *(Companies specific fonts ka use karti hain apne brand ki identity ko maintain karne ke liye).*

10. Explain Flex Layout in detail with example.

(8 CO-2)

- **Definition:**
  The Flexbox layout is a one-dimensional layout model in CSS that allows elements to be arranged either horizontally or vertically, making it easier to design flexible and responsive web layouts *(Flexbox layout ek one-dimensional layout model hai CSS mein, jisme elements ko horizontally ya vertically arrange kiya ja sakta hai, aur yeh flexible aur responsive web layouts design karna asaan banata hai).*

- **Use/Purpose:**
  Flexbox is used to create responsive layouts where elements can grow, shrink, or be spaced out evenly in relation to each other *(Flexbox ka use responsive layouts banane ke liye hota hai jahan elements ek dusre ke relation mein grow, shrink ya evenly spaced ho sakte hain).*

Syntax:

```
css

.container {

  display: flex;
  justify-content: space-between; /* Spacing between items */
  align-items: center; /* Align items vertically in the
center */
}

.item {
  flex: 1; /* Allow each item to grow equally */
}
```

Example:

```
css
.container {

  display: flex;
  justify-content: center; /* Center align items horizontally
*/
  align-items: center; /* Center align items vertically */
}

.item {
  flex: 1; /* All items will take equal space */
  padding: 10px;
```

```
  background-color: lightcoral;
}
```

HTML:
```
html
```

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

- **Real-life Application:**
    - **Responsive Navigation Bars:** Flexbox is widely used for creating responsive navigation bars, where menu items align horizontally on larger screens and vertically on smaller screens *(Flexbox ka use responsive navigation bars banane ke liye hota hai, jahan menu items large screens par horizontally aur chhote screens par vertically align hote hain).*
    - **Grid Layouts:** Flexbox is helpful for creating flexible grid layouts, where items adjust based on screen size *(Flexbox grid layouts banane mein madad karta hai, jahan items screen size ke hisaab se adjust hote hain).*

11. What is Grid Layout? Explain Grid Layout in detail with example.

(8 CO-2)

- **Definition:**
  The CSS Grid Layout is a two-dimensional layout system in CSS that allows for the creation of complex layouts with rows and columns *(CSS Grid Layout ek two-dimensional layout system hai CSS mein jo rows aur columns ke saath complex layouts banane mein madad karta hai).*

- **Use/Purpose:**
  Grid Layout is used to create grid-based designs with rows and columns, providing better control over the placement of elements *(Grid Layout ka use rows aur columns ke saath grid-based designs banane ke liye hota hai, jisse elements ke placement ko achhe se control kiya ja sakta hai).*

Syntax:
```
css
```

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

```
  grid-template-columns: repeat(3, 1fr); /* Create 3 equal
columns */
  grid-gap: 10px; /* Space between grid items */
}

.item {
  background-color: lightgreen;
  padding: 20px;
}
```

Example:

css

```
.container {

  display: grid;
  grid-template-columns: 1fr 2fr 1fr; /* 3 columns, with the
middle one being twice as wide */
  grid-gap: 20px; /* Add space between items */
}

.item {
  background-color: lightblue;
  padding: 10px;
  text-align: center;
}
```

HTML:

html

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

- **Real-life Application:**

- **Website Layouts:** Grid Layout is ideal for creating complex page layouts, such as magazine or newspaper-style designs with multiple columns *(Grid Layout complex page layouts banane ke liye ideal hai, jaise magazine ya newspaper-style designs jisme multiple columns ho).*
- **Dashboard Interfaces:** It is also used in creating dashboard interfaces where different content blocks align in rows and columns *(Yeh dashboard interfaces banane mein bhi use hota hai jahan content blocks rows aur columns mein align hote hain).*

12. What is the difference between CSS and CSS3?

(4 CO-2)

- **Definition:**
  - **CSS:** CSS (Cascading Style Sheets) is a styling language that defines how HTML elements are displayed on a web page *(CSS (Cascading Style Sheets) ek styling language hai jo define karti hai ki HTML elements web page par kaise display honge).*
  - **CSS3:** CSS3 is the latest version of CSS that introduced new features like animations, transitions, and more advanced selectors *(CSS3 CSS ka latest version hai jo naye features introduce karta hai jaise animations, transitions, aur advanced selectors).*
- **Use/Purpose:**
  - **CSS:** CSS is used to style the appearance of a website *(CSS ka use website ke appearance ko style karne ke liye hota hai).*
  - **CSS3:** CSS3 is used to enhance the styling capabilities of websites with advanced features like rounded corners, shadows, gradients, and animations *(CSS3 ka use websites ke styling capabilities ko enhance karne ke liye hota hai, jaise rounded corners, shadows, gradients, aur animations).*
- **Key Differences:**
  - **CSS:** Lacks features like animations, transitions, and advanced media queries *(CSS mein animations, transitions, aur advanced media queries jaise features nahi hote).*
  - **CSS3:** Introduces advanced features like animations, transitions, shadows, and gradients *(CSS3 mein advanced features jaise animations, transitions, shadows, aur gradients hote hain).*
- **Real-life Application:**
  - **CSS:** Used for basic styling, such as setting colors, fonts, and layouts *(CSS ka use basic styling ke liye hota hai, jaise colors, fonts, aur layouts set karna).*
  - **CSS3:** CSS3 is used to create modern and dynamic web pages with enhanced effects *(CSS3 ka use modern aur dynamic web pages banane ke liye hota hai jisme enhanced effects hote hain).*

13. Compare Hexadecimal color codes with RGB values? What does ‘a’ in the RGBA mean?

(4 CO-2)

- **Definition:**
  - **Hexadecimal Color Code:** A hexadecimal color code represents a color using a 6-digit combination of letters and numbers *(Hexadecimal color code ek color ko represent karta hai 6-digit combination se jo letters aur numbers ka use karta hai).*

    - **RGB:** RGB (Red, Green, Blue) is a color model that uses the combination of red, green, and blue light to create colors *(RGB (Red, Green, Blue) ek color model hai jo red, green, aur blue light ke combination ka use karke colors create karta hai).*
    - **RGBA:** RGBA is an extension of RGB, where 'A' stands for Alpha, representing opacity *(RGBA ek extension hai RGB ka, jisme 'A' ka matlab Alpha hota hai, jo opacity ko represent karta hai).*
- **Use/Purpose:**
    - **Hexadecimal Color Code:** Used in web design for specifying colors *(Hexadecimal color codes ka use web design mein colors specify karne ke liye hota hai)*.
    - **RGB:** RGB is widely used for defining colors in web design *(RGB ka use web design mein colors define karne ke liye hota hai).*
    - **RGBA:** RGBA is used when you need to set transparency or opacity along with color *(RGBA ka use tab hota hai jab aapko color ke saath transparency ya opacity set karni ho).*
- **Example:**
    - **Hexadecimal Color Code:** `#FF5733` (A bright orange color).
    - **RGB:** `rgb(255, 87, 51)` (A bright orange color using RGB values).
    - **RGBA:** `rgba(255, 87, 51, 0.5)` (A semi-transparent orange color with 50% opacity).
- **Real-life Application:**
    - **Hexadecimal:** Often used in HTML and CSS for defining colors *(Hexadecimal ka use HTML aur CSS mein colors define karne ke liye hota hai).*
    - **RGB and RGBA:** Used in creating colors for backgrounds, borders, and text in web design *(RGB aur RGBA ka use web design mein backgrounds, borders, aur text ke liye colors create karne mein hota hai).*

14. List some of the CSS frameworks.

(3 CO-2)

- **Definition:**
  CSS frameworks are pre-written CSS code that provides ready-to-use styles and layout components for faster web development *(CSS frameworks pre-written CSS code hote hain jo ready-to-use styles aur layout components provide karte hain, taaki web development jaldi ho sake).*
- **Use/Purpose:**
  CSS frameworks help developers create professional, consistent, and responsive websites more efficiently by offering reusable code and components *(CSS frameworks developers ko professional, consistent, aur responsive websites banane mein madad karte hain, jo reusable code aur components offer karte hain).*
- **Popular CSS Frameworks:**
    - **Bootstrap:** A highly popular framework that offers ready-to-use components, grid system, and responsive design tools *(Bootstrap ek popular framework hai jo ready-to-use components, grid system aur responsive design tools provide karta hai).*
    - **Foundation:** A responsive front-end framework providing flexible and customizable grid systems *(Foundation ek responsive front-end framework hai jo flexible aur customizable grid systems provide karta hai).*
    - **Bulma:** A modern CSS framework based on Flexbox, easy to use for building responsive web designs *(Bulma ek modern CSS framework hai jo Flexbox par based hai, aur responsive web designs banane mein aasaan hai).*

    - **Tailwind CSS:** A utility-first framework that provides low-level CSS utilities for fast design creation *(Tailwind CSS ek utility-first framework hai jo fast design creation ke liye low-level CSS utilities provide karta hai)*.
- **Real-life Application:**
    - **Website Development:** These frameworks are used by developers to quickly build responsive websites with predefined layouts and components *(Yeh frameworks developers dwara websites banane ke liye use hote hain jahan predefined layouts aur components hote hain)*.

15. What is Z-index?

(3 CO-2)

- **Definition:**
  The `z-index` property in CSS determines the stacking order of elements on the webpage, allowing some elements to be placed in front of or behind others *(CSS mein `z-index` property elements ke stacking order ko determine karti hai, jisse kuch elements dusre elements ke aage ya peeche rakhe ja sakte hain)*.
- **Use/Purpose:**
  `z-index` is used to control the overlapping of elements, such as when one element is positioned over another *(Z-index ka use elements ke overlapping ko control karne ke liye hota hai, jaise jab ek element dusre ke upar positioned ho)*.

Syntax:`css`

```
.element {
  position: absolute;
  z-index: 10;
}
```

Example:`css`

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

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

-

In this example, `.box2` will be stacked on top of `.box1` because of the higher `z-index` value.

- **Real-life Application:**
  - **Layered UI Elements:** `z-index` is used to control the visibility of elements like modals, dropdowns, or tooltips that need to appear above other content *(Z-index ka use UI elements jaise modals, dropdowns, ya tooltips ko control karne mein hota hai jo dusre content ke upar dikhne chahiye).*

16. What does margin: 40px 100px 120px 80px signify?

(4 CO-2)

- **Definition:**
  The `margin` property in CSS defines the space around an element. It can be set for all four sides in a specific order: top, right, bottom, and left *(CSS mein `margin` property element ke around space define karti hai. Yeh chaar sides ke liye ek specific order mein set hoti hai: top, right, bottom, aur left).*

- **Use/Purpose:**
  The `margin` property is used to control the space between elements, preventing them from being too close or overlapping *(Margin ka use elements ke beech mein space control karne ke liye hota hai, taaki woh ek dusre ke bahut kareeb ya overlap na karein).*

- **Syntax:**
  ```
  css
  margin: top right bottom left;
  ```

Example:

```
css

.container {

  margin: 40px 100px 120px 80px;
}
```

This means:

- `40px` is the margin for the top.
- `100px` is the margin for the right.
- `120px` is the margin for the bottom.
- `80px` is the margin for the left.

- **Real-life Application:**
  - **Spacing Between Sections:** Used to create space between elements or sections in web layouts, making the content more visually appealing *(Yeh elements ya sections ke beech mein space create karne ke liye use hota hai, taaki content visually appealing ho).*

17. What is responsive web design?

(3 CO-2)

- **Definition:**
  Responsive web design refers to creating web pages that automatically adjust their layout and content based on the screen size of the device being used *(Responsive web design ka matlab hai aise web pages banana jo apne layout aur content ko device ke screen size ke hisaab se automatically adjust karte hain).*

- **Use/Purpose:**
  It ensures that websites provide an optimal viewing experience across different devices, such as smartphones, tablets, and desktops *(Yeh ensure karta hai ki websites alag-alag devices jaise smartphones, tablets, aur desktops par optimal viewing experience de).*

Syntax (Using Media Queries):

```
css
```

```
@media (max-width: 768px) {

  .container {
    width: 100%;
  }
}
```

Example:

```
css
@media (max-width: 600px) {

  .navbar {
    flex-direction: column; /* Stack the navigation items
vertically */
  }
}
```

- **Real-life Application:**
  - **Mobile-Friendly Websites:** Ensures that websites are mobile-friendly and work seamlessly across various devices *(Yeh ensure karta hai ki websites mobile-friendly ho aur alag-alag devices par seamlessly kaam karein).*

18. What are the limitations of CSS?

(4 CO-2)

- **Definition:**
  CSS is a powerful styling language, but it has its limitations, such as lack of support for complex animations, limited browser compatibility, and limited control over layout behavior

(CSS ek powerful styling language hai, lekin iski kuch limitations hain, jaise complex animations ka lack, limited browser compatibility, aur layout behavior par limited control).

- **Use/Purpose:**
  Understanding CSS limitations helps developers decide when to use it alongside other technologies like JavaScript or CSS preprocessors *(CSS ki limitations ko samajhna developers ko madad karta hai yeh decide karne mein ki kab use CSS ke saath dusri technologies jaise JavaScript ya CSS preprocessors ki zarurat hai).*

- **Limitations:**
  - **Limited Layout Control:** CSS alone cannot provide complex layouts without using frameworks like Flexbox or Grid *(CSS khud complex layouts provide nahi kar sakta bina Flexbox ya Grid frameworks ke).*
  - **Cross-Browser Compatibility:** CSS may not always render consistently across all browsers *(CSS har browser mein consistently render nahi hota).*
  - **Animation Limitations:** CSS animations are limited compared to JavaScript *(CSS animations JavaScript ke comparison mein limited hote hain).*
- **Real-life Application:**
  - **Combining CSS with JavaScript:** Developers often combine CSS with JavaScript for more dynamic interactions and behaviors *(Developers zyada dynamic interactions aur behaviors ke liye CSS ko JavaScript ke saath combine karte hain).*

19. How is padding and margin different from one another in CSS with example?

(4 CO-2)

- **Definition:**
  - **Padding:** Padding is the space between the content of an element and its border *(Padding wo space hota hai jo element ke content aur border ke beech hota hai).*
  - **Margin:** Margin is the space outside the border, between the element and other surrounding elements *(Margin wo space hota hai jo border ke baahar hota hai, element aur surrounding elements ke beech).*
- **Use/Purpose:**
  - Padding is used to create space inside an element, ensuring the content doesn't touch the edges of the box *(Padding ka use element ke andar space create karne ke liye hota hai, taaki content box ke edges se na lage).*
  - Margin is used to create space between elements, ensuring they don’t overlap *(Margin ka use elements ke beech mein space create karne ke liye hota hai, taaki woh overlap na karein).*

Syntax:

```
css
.box {

  padding: 20px;
  margin: 10px;
}
```

Example:
css

```
.container {
  padding: 20px;
  margin: 10px;
}
```

In this example:

- Padding adds space inside the box, between the content and the border *(Padding box ke andar space add karta hai, content aur border ke beech)*.
- Margin adds space outside the box, pushing other elements away *(Margin box ke baahar space add karta hai, doosre elements ko door push karta hai)*.

- **Real-life Application:**
  - **Form Layouts:** Padding is used to give space between text and input fields, while margin is used to create space between different form elements *(Form layouts mein padding text aur input fields ke beech mein space dene ke liye hota hai, aur margin alag form elements ke beech mein space create karne ke liye hota hai)*.

20. How are block-level and inline elements different from one another in CSS?

(4 CO-2)

- **Definition:**
  - **Block-level Elements:** Block-level elements take up the full width of their container, starting on a new line *(Block-level elements apne container ki poori width occupy karte hain aur naye line se shuru hote hain)*.
  - **Inline Elements:** Inline elements take up only the space required by their content and do not start on a new line *(Inline elements sirf apne content ki zarurat ke mutabik space lete hain aur naye line pe start nahi hote)*.
- **Use/Purpose:**
  - Block-level elements are used for creating large structural components like headers, paragraphs, and divs *(Block-level elements ka use bade structural components jaise headers, paragraphs, aur divs banane ke liye hota hai)*.
  - Inline elements are used for smaller content like links or text that should appear within other content without breaking the layout *(Inline elements ka use chhote content jaise links ya text ke liye hota hai, jo doosre content ke andar bina layout break kiye appear hote hain)*.

Syntax:
css

```
.block {

  display: block;
}

.inline {
  display: inline;
}
```

Example:
`html`

```
<div class="block">This is a block-level element.</div>

<span class="inline">This is an inline element.</span>
```

- **Real-life Application:**
    - **Text and Paragraph Layout:** Block-level elements like paragraphs create sections in text content, whereas inline elements like links fit naturally within text without breaking the flow *(Text aur paragraph layouts mein block-level elements jaise paragraphs text content mein sections create karte hain, jabki inline elements jaise links text ke andar naturally fit ho jaate hain bina flow break kiye)*.

21. Using HTML, CSS create a hover and focus effect for navigation items, using CSS transformations.

(8 CO-2)

- **Definition:**
  The hover and focus effects in CSS allow developers to add interactive styles when a user hovers over or focuses on an element *(CSS mein hover aur focus effects developers ko interactive styles add karne ki suvidha dete hain jab user kisi element par hover ya focus karta hai)*.
    - **Hover:** Applied when a user moves the cursor over an element *(Hover tab apply hota hai jab user kisi element par cursor move karta hai)*.
    - **Focus:** Applied when a user focuses on an element *(Focus tab apply hota hai jab user kisi element par focus karta hai)*.
- **Use/Purpose:**
  Hover and focus effects are used to provide visual feedback to users, improving user

interaction with navigation items *(Hover aur focus effects ka use visual feedback dene ke liye hota hai, taaki users ka interaction navigation items ke saath behtar ho sake).*

Syntax (Hover and Focus):

```
css
```

```
nav a {
  text-decoration: none;
  transition: transform 0.3s ease;
}

nav a:hover {
  transform: scale(1.1); /* Scales up the item on hover */
}

nav a:focus {
  outline: none;
  transform: scale(1.1); /* Same effect when focused */
}
```

Example:

```
html
```

```
<nav>
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Services</a>
  <a href="#">Contact</a>
</nav>
```

-

  In this example:
  - On hover, the navigation items will scale up *(items hover par scale ho jaayenge).*
  - On focus (when tabbed to), the items will also scale up *(focus par bhi items scale ho jaayenge).*
- **Real-life Application:**

- **Navigation Menus:** Used to enhance the interactivity of navigation menus, making them more engaging and user-friendly *(Navigation menus mein yeh effects use kiye jaate hain, taaki unki interactivity enhance ho aur user-friendly ho sake).*

22. How would you select all paragraphs within a div with the class "container" using CSS selectors?

(4 CO-2)

- **Definition:**
  CSS selectors are patterns used to select elements on the web page so that styles can be applied to them *(CSS selectors patterns hote hain jo web page ke elements ko select karne ke liye use kiye jaate hain, taaki unhe styles apply kiya ja sake).*

- **Use/Purpose:**
  The purpose of using CSS selectors is to target specific elements and apply styles to them *(CSS selectors ka use specific elements ko target karne aur unhe styles apply karne ke liye hota hai).*

Syntax:
css

```
.container p {

  /* styles for all paragraphs inside .container */
}
```

Example:
css

```
.container p {

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

In this example, all <p> elements inside a div with the class "container" will have blue text and a font size of 16px.

- **Real-life Application:**

- **Styling Content Sections:** This is useful for styling specific sections of a website, such as paragraphs within a certain container *(Yeh specific sections, jaise kisi container ke andar paragraphs ko style karne ke liye useful hota hai).*

23. Compare and contrast Flexbox and Grid Layouts in CSS, highlighting their main differences and use cases.

(8 CO-2)

- **Definition:**
  - **Flexbox:** Flexbox is a layout model that allows items to be arranged in one-dimensional rows or columns, where the items can adjust their size to fill the space *(Flexbox ek layout model hai jisme items ko ek-dimensional rows ya columns mein arrange kiya jaata hai, aur items apne size ko space ko bharne ke liye adjust kar sakte hain).*
  - **Grid Layout:** Grid Layout is a two-dimensional layout system that allows items to be placed into rows and columns simultaneously *(Grid Layout ek two-dimensional layout system hai jo items ko rows aur columns dono mein ek saath place karne ki suvidha deta hai).*
- **Use/Purpose:**
  - **Flexbox** is ideal for simpler layouts where items need to adjust based on the space available *(Flexbox simple layouts ke liye ideal hai, jahan items ko available space ke hisaab se adjust karna hota hai).*
  - **Grid Layout** is best suited for complex layouts where items need to be positioned both vertically and horizontally *(Grid Layout complex layouts ke liye best hai, jahan items ko vertical aur horizontal dono directions mein position karna hota hai).*
- **Main Differences:**
  - **Flexbox:**
    - One-dimensional (rows or columns) *(Ek-dimensional: rows ya columns).*
    - Items are distributed along the main axis *(Items main axis ke along distribute hote hain).*
  - **Grid Layout:**
    - Two-dimensional (both rows and columns) *(Do-dimensional: rows aur columns dono).*
    - Allows precise placement of items within both rows and columns *(Dono rows aur columns mein items ki precise placement allow karta hai).*

Example (Flexbox):

```
css
```

```
.container {
  display: flex;
  justify-content: space-between;
}

.item {
  width: 30%;
}
```

Example (Grid):
css

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 10px;
}

.item {
  background-color: lightblue;
}
```

- **Real-life Application:**
    - **Flexbox:** Used for navigation bars, card layouts, and simple forms *(Flexbox ka use navigation bars, card layouts, aur simple forms mein hota hai).*
    - **Grid Layout:** Used for complex web pages like magazine layouts, photo galleries, or dashboards *(Grid Layout complex web pages jaise magazine layouts, photo galleries, ya dashboards ke liye use hota hai).*

24. Provide an example of how you would create a simple two-column layout using Flexbox.

(8 CO-2)

- **Definition:**
  Flexbox can be used to create layouts that adjust dynamically. A simple two-column layout can be achieved by using Flexbox to distribute space between two elements *(Flexbox ka use layouts create karne ke liye hota hai jo dynamically adjust hote hain. Ek simple two-column layout Flexbox ka use karke do elements ke beech space distribute karke achieve kiya ja sakta hai).*

- **Use/Purpose:**
  The two-column layout is common for websites and blogs, where content is divided into two sections *(Two-column layout websites aur blogs mein common hai, jahan content ko do sections mein divide kiya jaata hai).*

Syntax:
css

```
.container {

  display: flex;
  justify-content: space-between;
}

.left-column, .right-column {
  flex: 1; /* Both columns take up equal space */
}
```

Example:
`html`

```
<div class="container">

  <div class="left-column">
    <p>Left Column Content</p>
  </div>
  <div class="right-column">
    <p>Right Column Content</p>
  </div>
</div>
```

- **Real-life Application:**
    - **Blog or News Website:** This layout is widely used on blogs and news websites to display main content and related articles side by side *(Yeh layout blog ya news websites mein widely use hota hai taaki main content aur related articles ko side by side dikhaya ja sake).*

25. Explain how CSS Grid Layout differs from traditional layout methods like floats and positioning.

(8 CO-2)

- **Definition:**
    - **CSS Grid Layout:** Grid Layout is a two-dimensional system, allowing both rows and columns to be used simultaneously *(CSS Grid Layout ek two-dimensional system hai, jo rows aur columns dono ko simultaneously use karne ki suvidha deta hai).*
    - **Traditional Layouts (Floats/Positioning):** Floats and positioning methods are one-dimensional, requiring more manual control for positioning elements and managing

their space *(Floats aur positioning methods ek-dimensional hote hain, aur elements ko position karne aur unke space ko manage karne ke liye zyada manual control ki zarurat hoti hai).*

- **Use/Purpose:**
  - CSS Grid is best used for complex layouts, whereas floats and positioning are better suited for simpler layouts *(CSS Grid complex layouts ke liye best hai, jabki floats aur positioning simpler layouts ke liye better suited hote hain).*
  - Grid Layout automatically adjusts the size of items based on their placement, whereas float-based layouts require clearing the floats and are prone to issues with overlapping *(Grid Layout apne items ke size ko automatically adjust kar leta hai unki placement ke hisaab se, jabki float-based layouts ko floats ko clear karna padta hai aur overlapping issues bhi ho sakte hain).*

Example (CSS Grid):

```
css
```

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 20px;
}

.item {
  background-color: lightgreen;
}
```

Example (Traditional Floats):

```
css
```

```
.container {
  width: 100%;
}

.item {
  float: left;
  width: 48%;
  margin-right: 4%;
}
```

- **Real-life Application:**
    - **Grid Layout:** Used for complex page layouts, like dashboards or websites with multiple sections *(Grid Layout complex page layouts jaise dashboards ya websites ke liye use hota hai jahan multiple sections hote hain).*
    - **Float and Positioning:** Used for simpler layouts or when working with legacy projects that require specific placements *(Float aur positioning simpler layouts ke liye ya legacy projects mein use hoti hai jahan specific placements ki zarurat hoti hai).*

26. Explain the display property in CSS and its various values.

(8 CO-2)

- **Definition:**
  The `display` property in CSS defines how an element is displayed on the page *(Display property CSS mein ek element ko page par kaise display kiya jaayega, yeh define karta hai).*
  It determines the layout behavior of an element and can change how elements interact with other elements around them *(Yeh element ke layout behavior ko determine karta hai aur yeh decide karta hai ki wo element doosre elements ke saath kaise interact karega).*

- **Use/Purpose:**
  The `display` property is used to define how an element behaves in the document flow, whether it should take up space or not, and whether it should be inline or block *(Display property ka use yeh decide karne ke liye hota hai ki element document flow mein kaise behave karega, kya wo space lega ya nahi, aur kya wo inline ya block hoga).*

- **Values of `display` Property:**
    - `block`: The element takes up the full width of its container, starting on a new line *(Block element container ki full width occupy karta hai, aur naye line se start hota hai).*
    - `inline`: The element only takes up as much width as it needs and doesn't start on a new line *(Inline element sirf utni width leta hai jitni zarurat hoti hai, aur naye line par start nahi hota).*
    - `inline-block`: The element is treated like an inline element but allows setting width and height *(Inline-block element inline element ki tarah treat hota hai, lekin width aur height set karne ki suvidha deta hai).*
    - `none`: The element is completely removed from the document flow *(Element ko document flow se completely remove kar diya jaata hai).*

Syntax:

```
css
```

```
.element {

  display: block;  /* Can be inline, block, inline-block, or
none */
}
```

Example:
`css`

```
div {
  display: block;  /* Takes full width of container */
}

span {
  display: inline;  /* Takes only required width */
}
```

- **Real-life Application:**
    - **Navigation Menus:** Block display is often used for menu items and headers, whereas inline display is used for links and buttons *(Block display navigation menus aur headers ke liye use hota hai, jabki inline display links aur buttons ke liye use hota hai).*
    - **Form Elements:** Form labels and buttons may use inline-block for flexible layout and control over dimensions *(Form elements jaise labels aur buttons flexible layout aur dimensions ko control karne ke liye inline-block use karte hain).*

27. Describe how you would create rounded corners for a box element using CSS.

(3 CO-2)

- **Definition:**
  To create rounded corners for a box element in CSS, the `border-radius` property is used *(CSS mein box element ke rounded corners banane ke liye `border-radius` property use hoti hai).*
- **Use/Purpose:**
  The `border-radius` property is used to create rounded corners for elements, making the design smoother and more visually appealing *(Border-radius property elements ke liye rounded corners create karti hai, jo design ko smoother aur visually appealing banata hai).*

Syntax:
`css`

```
.box {
```

```
  border-radius: 10px;  /* Creates rounded corners */
}
```

Example:
css

```
.container {
  border-radius: 15px;  /* Rounded corners for the box */
  background-color: lightblue;
  padding: 20px;
}
```

- **Real-life Application:**
    - **Buttons:** Rounded corners are commonly used in buttons to make them look more modern and appealing *(Buttons mein rounded corners ka use unhe modern aur appealing banane ke liye hota hai).*
    - **Cards:** Cards, which are often used in design systems, can have rounded corners to create a softer, more user-friendly interface *(Design systems mein jo cards hote hain, unmein rounded corners ka use interface ko softer aur user-friendly banane ke liye hota hai).*

28. Discuss the use of border images in CSS and provide an example.

(4 CO-2)

- **Definition:**
  Border images in CSS allow an image to be used as the border of an element *(CSS mein border images allow karti hain ek image ko element ke border ke roop mein use karna).*
- **Use/Purpose:**
  The `border-image` property allows you to apply an image as a border, giving you more design flexibility *(Border-image property aapko image ko border ke roop mein apply karne ki suvidha deti hai, jo design mein zyada flexibility deti hai).*

Syntax:
css

```
.element {
  border-image: url('border-image.png') 30 round;
}
```

Example:

```
css

.box {
  border: 10px solid transparent;
  border-image: url('border-pattern.png') 30 round;
}
```

- **Real-life Application:**
    - **Decorative Borders:** Border images are often used for adding decorative elements around boxes, such as image frames or custom borders *(Border images ka use boxes ke around decorative elements, jaise image frames ya custom borders, add karne ke liye hota hai).*

29. Discuss the role of CSS frameworks like Bootstrap in facilitating responsive web design.

(4 CO-2)

- **Definition:**
  CSS frameworks like Bootstrap provide pre-built CSS components and grid systems to help developers create responsive websites quickly *(CSS frameworks jaise Bootstrap pre-built CSS components aur grid systems provide karte hain jo developers ko responsive websites jaldi banane mein madad karte hain).*
- **Use/Purpose:**
  These frameworks help speed up the development process and ensure consistency across different screen sizes *(Yeh frameworks development process ko fast karte hain aur different screen sizes par consistency ensure karte hain).*
- **Key Features of Bootstrap:**
    - **Grid System:** A 12-column grid system that allows responsive layout design *(Grid system: Ek 12-column grid system jo responsive layout design ki suvidha deta hai).*
    - **Pre-designed Components:** Provides pre-designed components like buttons, modals, navbars, etc. *(Pre-designed components: Pre-designed components jaise buttons, modals, navbars, etc. provide karta hai).*
    - **Mobile-first Approach:** Bootstrap is built with a mobile-first approach, meaning the design is optimized for smaller screens first and then scaled up for larger screens *(Mobile-first approach: Bootstrap mobile-first approach ke saath built hota hai, iska matlab hai ki design pehle chhote screens ke liye optimize hota hai aur phir larger screens ke liye scale up hota hai).*
- **Syntax:**
  Bootstrap's syntax involves using pre-built classes *(Bootstrap ka syntax pre-built classes ka use karta hai).*

Example:
`html`

```
<div class="container">

  <div class="row">
    <div class="col-md-6">
      <p>This is a responsive column!</p>
    </div>
    <div class="col-md-6">
      <p>This is another responsive column!</p>
    </div>
  </div>
</div>
```

- **Real-life Application:**
    - **Responsive Websites:** Bootstrap is widely used for creating responsive websites that work on multiple devices like desktops, tablets, and smartphones *(Bootstrap ka use responsive websites banane ke liye hota hai jo multiple devices par kaam karte hain, jaise desktops, tablets, aur smartphones).*

Unit-3: Working With JavaScript

1. What is JavaScript? What are the benefits of JavaScript? (3 CO-3)

Definition:

JavaScript is a lightweight, interpreted programming language used to create interactive web pages. *(JavaScript ek halka aur turant execute hone wala programming language hai jo interactive web pages banane ke liye use hota hai.)*

Use/Purpose:

- It helps in adding dynamic content to websites. *(Ye websites me dynamic content add karne me madad karta hai.)*
- JavaScript runs on the client side, reducing server load. *(Ye client side pe run hota hai, jisse server ka load kam hota hai.)*
- It allows event-driven programming for interactive user experiences. *(Ye event-driven programming ko support karta hai jo user experience ko interactive banata hai.)*

Syntax:

```
console.log("Hello, JavaScript!");
```

(Yahaan console.log() ka use output dikhane ke liye kiya gaya hai.)

Example:

```
let name = "Amit";

alert("Hello, " + name);
```

(Yeh ek simple example hai jo ek pop-up box me "Hello, Amit" dikhayega.)

Real-life Application:

- Used in web development to create interactive elements like forms and buttons. *(Web development me interactive elements jaise ki forms aur buttons banane ke liye use hota hai.)*
- Used in game development and mobile applications. *(Game development aur mobile apps me bhi iska use hota hai.)*

Keywords Explanation:

- Client-side scripting – JavaScript runs in the browser, not on the server. *(JavaScript browser me run hota hai, server pe nahi.)*
- Dynamic content – Content that changes without reloading the page. *(Aisa content jo bina page reload kiye change ho sakta hai.)*
- Event-driven – Code execution depends on user actions like clicks. *(Code*

execution user ke actions jaise clicks pe depend karta hai.)

---

2. Differentiate Client-side scripting and Server-side scripting (4 CO-3)

Feature	Client-side Scripting	Server-side Scripting
Execution	Runs in the browser	Runs on the server
Speed	Faster	Slower (due to server communication)
Examples	JavaScript, HTML, CSS	PHP, Python, Node.js
Security	Less secure	More secure
Page Load Speed	Faster	Slower

Explanation:

- Client-side scripting executes on the user's web browser. *(Client-side scripting user ke web browser me execute hota hai.)*
- Server-side scripting executes on the server before sending the response to the client. *(Server-side scripting pehle server pe execute hota hai aur fir client ko response bhejta hai.)*

---

3. Explain pop-up boxes in JavaScript with example. (3 CO-3)

Definition:

Pop-up boxes in JavaScript are used to display messages or take user input. *(JavaScript me pop-up boxes ka use messages dikhane ya user input lene ke liye hota hai.)*

Types of Pop-up Boxes:

1. Alert Box: Displays a message. *(Ek simple message dikhata hai.)*
2. Confirm Box: Asks for confirmation (OK/Cancel). *(User se OK ya Cancel choose karne ko kehta hai.)*
3. Prompt Box: Takes user input. *(User se koi input leta hai.)*

Example:

```
alert("This is an alert box!");  // Simple alert box

confirm("Are you sure?");  // Confirmation box

let name = prompt("Enter your name:");  // Prompt box

console.log("Hello, " + name);
```

(Ye example teen tareeke ke pop-ups dikhata hai: Alert, Confirm, aur Prompt.)

Real-life Application:

- Used to show warnings and notifications. *(Warnings aur notifications dikhane ke liye use hota hai.)*
- Used in form validations. *(Form validation me use hota hai.)*

---

4. Explain DOM in JavaScript. (4 CO-3)

Definition:

DOM (Document Object Model) is a programming interface for web documents. *(DOM ek programming interface hai jo web documents ko access karne aur modify karne deta hai.)*

Use/Purpose:

- Helps manipulate HTML and CSS dynamically. *(HTML aur CSS ko dynamically change karne me madad karta hai.)*

- Allows JavaScript to change web page content. *(JavaScript ke through web page ka content change kiya ja sakta hai.)*

Example:

```
document.getElementById("demo").innerHTML = "Hello, DOM!";
```

(Yeh code id=demo wale element ka content change karega.)

Real-life Application:

- Used in dynamic web pages to update content without refreshing. *(Dynamic web pages me bina reload kiye content update karne ke liye use hota hai.)*
- Used in interactive applications like online quizzes. *(Interactive applications jaise online quizzes me use hota hai.)*

Keywords Explanation:

- DOM Tree – A hierarchical structure of the web page. *(Web page ka ek tree structure hota hai.)*
- getElementById() – Used to select an HTML element. *(HTML element select karne ke liye use hota hai.)*
- innerHTML – Property to change content of an element. *(Element ka content change karne ke liye use hoti hai.)*

---

5. Explain Callback function in JavaScript with example. (4 CO-3)

Definition:

A callback function is a function passed as an argument to another function. (Callback function ek aisi function hoti hai jo kisi doosre function me argument ki tarah pass hoti hai.)

Use/Purpose:

- Used to execute code after a function finishes execution. *(Ek function ke execute hone ke baad doosra code run karne ke liye use hota hai.)*
- Used in asynchronous programming. *(Asynchronous programming me use hota hai, jisme kuch tasks delay se complete hote hain.)*

Example:

```
function greet(name, callback) {
```

```
    console.log("Hello, " + name);

    callback();

}

function sayGoodbye() {

    console.log("Goodbye!");

}

greet("Amit", sayGoodbye);
```

(Is example me sayGoodbye() function ko greet() function ke andar callback ki tarah use kiya gaya hai.)

Real-life Application:

- Used in handling asynchronous operations like API calls. *(API calls aur asynchronous operations handle karne ke liye use hota hai.)*
- Used in event handling. *(Event handling ke liye use hota hai, jaise button click pe function execute karna.)*

Keywords Explanation:

- Callback – A function passed to another function. *(Ek function jo doosre function ko diya jata hai.)*
- Asynchronous – Code execution without blocking the main program. *(Code execution jo main program ko block nahi karta.)*
- Event-driven – Execution depends on events like clicks. *(Execution events jaise clicks pe depend karta hai.)*

6. What is JavaScript event handling? List the major events and show the use of at least one event by writing JavaScript code. (8 CO-3)

Definition:

Event handling in JavaScript refers to the process of responding to user actions such as clicks, key presses, or mouse movements.

(JavaScript me event handling ka matlab hai user ke actions jaise click, key press ya mouse movement ka response dena.)

Use/Purpose:

- Helps in creating interactive web pages. *(Interactive web pages banane me madad karta hai.)*
- Allows execution of specific functions when an event occurs. *(Koi event hone par specific function execute karne ki suvidha deta hai.)*
- Common in form validation, animations, and UI interactions. *(Form validation, animations, aur UI interactions me use hota hai.)*

Major Events in JavaScript:

1. onclick – Runs code when an element is clicked. *(Jab element pe click hota hai to ye code execute karta hai.)*
2. onchange – Runs when an input value changes. *(Input value badalne par ye execute hota hai.)*
3. onmouseover – Runs when mouse is over an element. *(Mouse kisi element ke upar aane par ye trigger hota hai.)*
4. onmouseout – Runs when mouse leaves an element. *(Mouse element se bahar jane par trigger hota hai.)*
5. onkeypress – Runs when a key is pressed. *(Jab koi key press hoti hai tab trigger hota hai.)*
6. onload – Runs when the page is fully loaded. *(Jab page poori tarah load ho jata hai tab execute hota hai.)*
7. onfocus – Runs when an input field is focused. *(Jab kisi input field pe click hota hai to trigger hota hai.)*

Syntax:

javascript

```
element.event = function() {

    // Code to execute when event occurs

};
```

(Event ka naam event hota hai aur function define karta hai ki kya execute hoga.)

Example (onclick event):

javascript

```
<button onclick="showMessage()">Click me</button>
```

```
<script>
function showMessage() {
    alert("Button clicked!");
}
</script>
```

(Yeh example ek button pe click hone par "Button clicked!" message dikhayega.)

Real-life Application:

- Used in form validation (e.g., checking if a field is empty). *(Form validation me use hota hai, jaise ki koi field empty hai ya nahi.)*
- Used in gaming for detecting user interactions. *(Gaming applications me user ke interactions detect karne ke liye use hota hai.)*

Keywords Explanation:

- Event Listener – A function that waits for an event to occur. *(Ek function jo kisi event hone ka intezar karta hai.)*
- Trigger – The action that causes an event to occur. *(Jo action kisi event ko activate karta hai.)*
- Handler – The function executed when an event happens. *(Jo function event hone par execute hota hai.)*

---

7. Demonstrate the use of onchange, onmouseover, onmouseout, onkeypress, onload, onfocus events with proper example. (8 CO-3)

Definition:

JavaScript provides various events to handle user interactions dynamically.
(JavaScript user ke interactions ko handle karne ke liye different events provide karta hai.)

Use/Purpose:

- Helps in form validation. *(Form validation me madad karta hai.)*
- Enhances user experience with interactive UI. *(UI ko interactive banane ke liye use hota hai.)*

Examples:

html

```
<!DOCTYPE html>

<html>

<body>


<!-- onchange Event -->

<input type="text" onchange="alert('Value changed!')"
placeholder="Type something">


<!-- onmouseover & onmouseout Events -->

<p onmouseover="this.style.color='red'"
onmouseout="this.style.color='black'">Hover over me</p>


<!-- onkeypress Event -->

<input type="text" onkeypress="console.log('Key
Pressed!')" placeholder="Press a key">


<!-- onload Event -->
```

```
<script>

window.onload = function() {

    alert("Page loaded successfully!");

};

</script>


<!-- onfocus Event -->

<input type="text"
onfocus="this.style.backgroundColor='yellow'"
placeholder="Click to focus">


</body>

</html>
```

(Yeh example different events ka use dikhata hai jaise onchange, onmouseover, onmouseout, onkeypress, onload, aur onfocus.)

Real-life Application:

- Used in login forms to highlight active fields. *(Login forms me active fields highlight karne ke liye use hota hai.)*
- Used in dynamic tooltips when hovering over elements. *(Hover karne par tooltip dikhane ke liye use hota hai.)*

---

8. List out JavaScript's inbuilt Objects. Explain any three JavaScript's inbuilt Objects with proper example. (8 CO-3)

Definition:

JavaScript provides built-in objects that help in handling data efficiently.
(JavaScript me kuch built-in objects hote hain jo data ko efficiently handle karne me madad karte hain.)

Common JavaScript Inbuilt Objects:

1. Math – Performs mathematical operations. *(Mathematical calculations karne ke liye use hota hai.)*

2. Date – Handles date and time. *(Date aur time manage karne ke liye use hota hai.)*
3. String – Manipulates text. *(Text manipulate karne ke liye use hota hai.)*
4. Array – Stores multiple values. *(Multiple values store karne ke liye use hota hai.)*
5. Object – Stores key-value pairs. *(Key-value pairs store karne ke liye use hota hai.)*

Examples:

1. Math Object:

javascript

```
console.log(Math.sqrt(16)); // Output: 4

console.log(Math.random()); // Output: Random number
between 0 and 1
```

(Math object ka use square root aur random number generate karne ke liye kiya gaya hai.)

2. Date Object:

javascript

```
let today = new Date();

console.log(today.toDateString()); // Output: Current date
in readable format
```

(Date object current date dikhane ke liye use hota hai.)

3. String Object:

javascript

```
let str = "Hello, JavaScript!";

console.log(str.length); // Output: 18

console.log(str.toUpperCase()); // Output: HELLO,
JAVASCRIPT!
```

(String object ka use text manipulate karne ke liye hota hai.)

Real-life Application:

- Used in calculations, date formatting, and text processing. *(Calculations, date formatting, aur text processing me use hota hai.)*
- Used in building dynamic and interactive websites. *(Dynamic websites banane me madad karta hai.)*

---

9. How user-defined Objects are created in JavaScript? Explain with proper example. (8 CO-3)

Definition:

A user-defined object is a custom object created by the programmer.
(User-defined object wo hota hai jo programmer khud banata hai.)

Syntax:

javascript

```
function Student(name, age) {
    this.name = name;
    this.age = age;
}
```

(Is syntax me ek function ka use object create karne ke liye kiya gaya hai.)

Example:

javascript

```
function Car(brand, model) {
    this.brand = brand;
    this.model = model;
}
```

```
let myCar = new Car("Toyota", "Corolla");

console.log(myCar.brand); // Output: Toyota
```

(Yeh example ek custom object Car banata hai jisme brand aur model properties hoti hain.)

Real-life Application:

- Used in defining reusable data structures like Employee, Product, etc. *(Employee, Product jaise reusable structures banane ke liye use hota hai.)*

10. Explain the concept of an Array in JavaScript. Also, Demonstrate with an example. (8 CO-3)

Definition:

An array in JavaScript is a special variable used to store multiple values in a single variable.
(JavaScript me array ek special variable hai jo ek hi variable me multiple values store karne ke liye use hota hai.)

Use/Purpose:

- Helps in storing and managing multiple values efficiently. *(Multiple values ko efficiently store aur manage karne ke liye use hota hai.)*
- Used in loops and data manipulation. *(Loops aur data manipulation me madad karta hai.)*
- Reduces the need for multiple variables. *(Multiple variables use karne ki zaroorat kam karta hai.)*

Syntax:

javascript

```
let arrayName = [value1, value2, value3];
```

(Ye syntax ek array banata hai jisme multiple values hoti hain.)

Example:

javascript

```
let fruits = ["Apple", "Banana", "Mango"];
```

```
console.log(fruits[0]); // Output: Apple

console.log(fruits.length); // Output: 3

fruits.push("Orange"); // Adds a new element

console.log(fruits); // Output: ["Apple", "Banana",
"Mango", "Orange"]
```

(Yeh example ek array create karta hai jisme fruits store kiye gaye hain aur ek new element add kiya gaya hai.)

Real-life Application:

- Used in e-commerce websites to store product lists. *(E-commerce websites me product list store karne ke liye use hota hai.)*
- Used in social media platforms to store user comments and messages. *(Social media apps me user ke comments aur messages store karne ke liye use hota hai.)*

Keywords Explanation:

- Index – The position of an element in an array (starting from 0). *(Array ke elements ka position.)*
- Push() – Adds an element at the end of the array. *(Array ke last me element add karta hai.)*
- Pop() – Removes the last element from the array. *(Array ke last element ko remove karta hai.)*
- Length – Returns the total number of elements in the array. *(Array me kitne elements hain ye batata hai.)*

11. Demonstrate JavaScript Form Validation with a proper example. (8 CO-3)

Definition:

Form validation in JavaScript is used to check whether user input in a form is correct before submitting it to the server.
(JavaScript form validation ka use user input ko check karne ke liye hota hai taaki galat data server pe submit na ho.)

Use/Purpose:

- Ensures correct data is submitted. *(Data ko sahi format me submit karne ke liye use hota hai.)*
- Prevents empty fields and incorrect values. *(Empty fields aur galat values submit hone se roknay ke liye.)*
- Improves user experience by giving immediate feedback. *(User ko turant feedback dene ke liye.)*

Syntax:

javascript

```
function validateForm() {
    let name = document.getElementById("name").value;
    if (name == "") {
        alert("Name must be filled out");
        return false;
    }
}
```

(Yeh function check karega ki user ne name enter kiya hai ya nahi.)

Example:

html

```
<form onsubmit="return validateForm()">
    Name: <input type="text" id="name">
    <input type="submit" value="Submit">
</form>
<script>
    function validateForm() {
        let name = document.getElementById("name").value;
        if (name == "") {
            alert("Name must be filled out");
            return false;
        }
    }
```

```
</script>
```

(Yeh form validation check karega ki name field khali to nahi hai.)

Real-life Application:

- Used in login and signup forms. *(Login aur signup forms me use hota hai.)*
- Used in payment forms to ensure correct details are entered. *(Payment forms me details check karne ke liye.)*

Keywords Explanation:

- Validation – Checking if input data is correct. *(Data sahi hai ya nahi yeh check karna.)*
- onSubmit – Runs JavaScript when the form is submitted. *(Form submit hone pe JavaScript execute hota hai.)*
- alert() – Displays a warning message. *(User ko warning message dikhane ke liye.)*

---

12. Write a JavaScript code to display Fibonacci Series of given number. Number should be entered by user through a text box. (8 CO-3)

Definition:

Fibonacci Series is a sequence of numbers where each number is the sum of the two preceding ones.
(Fibonacci series ek sequence hai jisme har number pehle do numbers ka sum hota hai.)

Use/Purpose:

- Used in mathematical calculations. *(Mathematical problems solve karne me madad karta hai.)*
- Important in computer algorithms. *(Computer algorithms me use hota hai.)*

Syntax:

javascript

```
function fibonacci(n) {
    let a = 0, b = 1, sum;
    for (let i = 0; i < n; i++) {
        console.log(a);
        sum = a + b;
```

```
        a = b;
        b = sum;
    }
}
```

(Yeh function Fibonacci series generate karega.)

Example:

html

```
<input type="number" id="num" placeholder="Enter number">
<button onclick="showFibonacci()">Generate</button>
<p id="result"></p>

<script>
function showFibonacci() {
    let n = document.getElementById("num").value;
    let a = 0, b = 1, sum;
    let series = [a, b];
    for (let i = 2; i < n; i++) {
        sum = a + b;
        series.push(sum);
        a = b;
        b = sum;
    }
    document.getElementById("result").innerText =
series.join(", ");
```

```
}
</script>
```

(Yeh code user input lega aur Fibonacci series dikhayega.)

Real-life Application:

- Used in search algorithms. *(Search algorithms me use hota hai.)*
- Used in financial predictions. *(Financial calculations me use hota hai.)*

Keywords Explanation:

- Fibonacci series – Sequence of numbers where each term is the sum of the previous two. *(Ek sequence jo pehle do numbers ke sum se banta hai.)*
- Loop – Used to repeat a process multiple times. *(Ek process ko repeat karne ke liye loop ka use hota hai.)*

---

13. Write a JavaScript code to change the background color of the page at a specific time interval. (4 CO-3)

Definition:

JavaScript can be used to change the background color of a webpage dynamically at set intervals.
(JavaScript ka use background color ko automatically change karne ke liye hota hai.)

Use/Purpose:

- Improves user experience. *(User experience ko interactive banata hai.)*
- Used in animations and effects. *(Web animations aur effects ke liye use hota hai.)*

Example:

html

```
<script>
function changeColor() {
    let colors = ["red", "blue", "green", "yellow",
"pink"];
    let index = 0;
```

```
    setInterval(function () {
        document.body.style.backgroundColor =
colors[index];
        index = (index + 1) % colors.length;
    }, 1000);
}
changeColor();
</script>
```

(Yeh code har second background color change karega.)

Real-life Application:

- Used in advertisement banners. *(Ad banners me color change dikhane ke liye.)*
- Used in digital clocks for color effects. *(Digital clocks me color animation ke liye.)*

Keywords Explanation:

- setInterval() – Repeats a task after a fixed interval. *(Ek task ko repeat karne ke liye use hota hai.)*
- style.backgroundColor – Changes background color dynamically. *(Background color ko change karta hai.)*

---

14. Write a JavaScript code that takes an integer number as input and tells whether the number is prime or not. (8 CO-3)

Definition:

A prime number is a number that is only divisible by 1 and itself.
(Prime number wo hota hai jo sirf 1 aur khud se divide hota hai.)

Example:

html

```
<input type="number" id="num">
<button onclick="checkPrime()">Check Prime</button>
```

```
<p id="result"></p>


<script>
function checkPrime() {
    let num = document.getElementById("num").value;
    let isPrime = true;


    if (num < 2) isPrime = false;
    for (let i = 2; i < num; i++) {
        if (num % i == 0) {
            isPrime = false;
            break;
        }
    }
    document.getElementById("result").innerText = isPrime
? "Prime Number" : "Not a Prime Number";
}
</script>
```

(Yeh code check karega ki diya gaya number prime hai ya nahi.)

---

15. Write JavaScript code that displays the text “SILVER OAK UNIVERSITY” with increasing font size in intervals of 1 second in blue color. When font size reaches 50px, it should stop. (8 CO-3)

Example:

```
<p id="text" style="color:blue;">SILVER OAK
```

```
UNIVERSITY</p><!-- The text which will increase in size
-->

 // Function to increase font size every 1 second

<script>

let size = 10; // Initial font size

let interval = setInterval(function () {

    if (size >= 50) // If font size reaches 50px, stop{

        clearInterval(interval)// Stops the interval
execution;

    } else {

        document.getElementById("text").style.fontSize =
size + "px";

        size += 2;// Increase size by 2px every second

    }

}, 1000); // 1000ms = 1 second

</script>
```

(Yeh text size har second badhata rahega jab tak 50px tak nahi pahunchta.)

16. Difference between == and === operators (3 CO-3)

Definition:

In JavaScript, == is the equality operator that checks for value equality, whereas === is the strict equality operator that checks for both value and type equality.
(JavaScript me == sirf value check karta hai, jabki === value aur type dono check karta hai.)

Use/Purpose:

== converts data types if needed before comparing. *(Ye automatic type conversion karta hai.)*

=== does not perform type conversion, making it more precise. *(Ye bina conversion ke exact comparison karta hai.)*

Syntax:

javascript

```
console.log(5 == "5");  // true (because type conversion happens)

console.log(5 === "5"); // false (because types are different)
```

(Yahaan == wale case me number string me convert ho gaya, jabki === me aisa nahi hua.)

Example:

javascript

```
console.log(0 == false);  // true

console.log(0 === false); // false
```

(== me false ko 0 ke barabar maana gaya, jabki === me nahi.)

Real-life Application:

Used in form validation to compare user inputs.
(Form validation me user input compare karne ke liye use hota hai.)

Keywords Explanation:

Type coercion – == automatically converts types.

Strict equality – === checks both value and type.

features	**== (Equality Operator)**	**===(strict operator)**
Definition	Compares values after type conversion.	Compares both value and data type.

Type Conversion	Yes, it converts types if needed before comparison.	No, it does not convert types.
Comparison	Only checks if values are equal.	Checks if values and types are both equal.
Usage	Used when type conversion is acceptable.	Used when strict comparison is required.
Syntax	5 == "5" → true (Type conversion happens)	5 === "5" → false (Different types)
Example	0 == false → true	0 === false → false
Real-life Application	Used in loose comparisons, e.g., form validation.	Used in strict comparisons, e.g., checking user roles in authentication.

---

17. Difference between var and let keyword in JavaScript (4 CO-3)

Definition:

var and let are used to declare variables, but let has block scope while var has function scope.
(var aur let dono variables declare karne ke liye use hote hain, lekin let ka scope block hota hai aur var ka function.)

Use/Purpose:

var variables can be re-declared in the same scope. *(Ek hi scope me var dobara declare ho sakta hai.)*

let variables cannot be re-declared in the same scope. *(let wale variables ek hi scope me dobara declare nahi ho sakte.)*

let prevents accidental overwriting of variables. *(let variable overwrite hone se bachata hai.)*

Syntax:

javascript

```
var x = 10;

var x = 20; // Allowed


let y = 10;

let y = 20; // Error
```

(var me variable dubara declare ho sakta hai, lekin let me error aayega.)

Example:

javascript

```
if (true) {
    var a = 10;
    let b = 20;
}
console.log(a); // Works
console.log(b); // Error
```

(var ka scope function tak hota hai, let ka block tak hi hota hai.)

Real-life Application:

Used in modern JavaScript development to avoid errors.
(Modern JavaScript me bugs se bachne ke liye let use hota hai.)

Keywords Explanation:

Block scope – let is limited to the block {}.

Function scope – var is limited to the function.

features	== (Equality Operator)	===(strict operator)
Definition	Used to declare variables with function scope.	Used to declare variables with block scope.
Scope	Function-scoped (accessible within the function).	Block-scoped (only accessible within {} block).
Re-declaration	Allowed within the same scope.	Not allowed within the same scope (throws an error).
Hoisting	Hoisted with an initial value of undefined.	Hoisted but not initialized (gives an error if used before declaration).
Usage	Used in older JavaScript versions.	Used in modern JavaScript to prevent accidental overwrites.
Syntax	var x = 10; var x = 20; (No error)	let y = 10; let y = 20; (Error)
Example	js if (true) { var x = 10; } console.log(x); // Works	js if (true) { let y = 10; } console.log(y); // Error
Real-life Application	Used in old JavaScript codebases.	Preferred in modern JavaScript to avoid scope-related issues.

---

18. What is NaN property in JavaScript? (3 CO-3)

Definition:

NaN (Not a Number) is a special value in JavaScript that represents an invalid number. *(NaN ek special value hai jo batata hai ki koi calculation invalid hai.)*

Use/Purpose:

Used to check if a value is not a valid number. *(Check karne ke liye ki value number hai ya nahi.)*

Occurs when mathematical operations fail. *(Mathematical operations fail hone par NaN aata hai.)*

Syntax:

javascript

```
console.log(0 / 0); // NaN

console.log(parseInt("Hello")); // NaN
```

(Yahaan 0 / 0 aur invalid string parsing me NaN aaya.)

Example:

javascript

```
let x = "abc" * 5;

console.log(isNaN(x)); // true
```

*("abc" * 5 invalid hai, isliye NaN return hota hai.)*

Real-life Application:

Used in form validation to check if the input is a valid number.
(Form validation me check karne ke liye ki user ne sahi number enter kiya hai ya nahi.)

Keywords Explanation:

Invalid number – A result that is not a valid numeric value.

isNaN() function – Used to check if a value is NaN.

---

19. Explain call(), apply(), and bind() methods. (8 CO-3)

Definition:

These are methods in JavaScript used to control the execution context of functions.
(Yeh methods functions ka execution context control karne ke liye use hote hain.)

Use/Purpose:

call() – Calls a function with a given this value and arguments.

apply() – Similar to call(), but takes arguments as an array.

bind() – Returns a new function with a fixed this value.

Syntax:

javascript

```
const person = { name: "Amit" };
function greet(msg) {
    console.log(msg + ", " + this.name);
}
greet.call(person, "Hello"); // "Hello, Amit"
greet.apply(person, ["Hi"]); // "Hi, Amit"
const newFunc = greet.bind(person);
newFunc("Hey"); // "Hey, Amit"
```

(call() aur apply() function ko turant call karte hain, jabki bind() ek naye function me store karta hai.)

Example:

javascript

```
const student = { name: "Ravi" };
function showDetails(age) {
    console.log(this.name + " is " + age + " years old.");
}
```

```
showDetails.call(student, 20);

showDetails.apply(student, [22]);

const boundFunc = showDetails.bind(student);

boundFunc(25);
```

(Yahaan call(), apply(), aur bind() se this ko student object se bind kiya gaya hai.)

Real-life Application:

Used in event handling and object inheritance.
(Event handling aur object ke methods reuse karne ke liye use hota hai.)

Keywords Explanation:

Execution context – Defines this inside a function.

Function borrowing – Using methods from other objects.

---

20. What is DOM? Explain in detail.

Definition:

DOM (Document Object Model) is a programming interface for web documents that represents the structure of a webpage.
(DOM ek programming interface hai jo web page ka structure represent karta hai.)

Use/Purpose:

Allows JavaScript to interact with and manipulate HTML elements. *(JavaScript ko HTML ke elements se interact karne ki permission deta hai.)*

Makes web pages dynamic and interactive. *(Web pages ko dynamic aur interactive banata hai.)*

Syntax:

javascript

```
document.getElementById("demo").innerHTML = "Hello!";
```

(Yeh code HTML element ki value change karega.)

Example:

javascript

```
document.querySelector("h1").style.color = "red";
```

(Yeh code h1 tag ka color red kar dega.)

Real-life Application:

Used to create interactive elements like buttons, forms, and animations.
(Interactive buttons, forms aur animations banane ke liye use hota hai.)

Keywords Explanation:

Nodes – Elements in the DOM structure.

Manipulation – Changing HTML or CSS dynamically.

21. What are JavaScript Data Types? (4 CO-3)

Definition:

A data type in JavaScript specifies the kind of data a variable can store.
(Data type batata hai ki variable kis type ka data store karega.)

Use/Purpose:

- Helps in organizing and processing different types of data efficiently.
  (Data ko efficiently organize aur process karne me madad karta hai.)

Ensures that operations are performed correctly on values.
(Operations ko sahi tarike se execute hone me help karta hai.)

Syntax:

javascript

```
let x = 10;       // Number
let name = "Ali"; // String
let isActive = true; // Boolean
let value = null; // Null
let y;            // Undefined
```

(Yeh syntax different data types ke example dikhata hai.)

Example:

javascript

```
let person = {name: "Rahul", age: 22}; // Object
let numbers = [1, 2, 3, 4]; // Array
let add = function(a, b) { return a + b; }; // Function
```

(Object, Array aur Function ka use dikhata hai.)

Real-life Application:

- Used in storing user data like name, age, and email in applications.
  (User details jaise naam, umar, email store karne ke liye.)
- Helps in mathematical calculations like e-commerce billing.
  (Shopping cart billing ya price calculations ke liye.)

Keywords Explanation:

- Primitive Data Types: Simple values (Number, String, Boolean, Null, Undefined).
- Non-Primitive Data Types: Complex structures (Object, Array, Function).

---

22. What would be the result of 3 + 2 + "7"? (3 CO-3)

Definition:

When numbers and strings are combined in JavaScript, type conversion occurs.
(JavaScript automatically kuch values ko convert karta hai jab unhe add kiya jata hai.)

Use/Purpose:

- Helps in understanding how JavaScript handles numbers and strings in operations.
  (JavaScript ka type conversion samajhne ke liye.)

Syntax & Example:

javascript

```
console.log(3 + 2 + "7");  // Output: "57"
```

(Pehle 3 + 2 hoga jo 5 dega, fir "7" string ke saath concatenate ho jayega.)

javascript

```
console.log("3" + 2 + 7);   // Output: "327"
```

(Pehle "3" string hai, toh JavaScript 2 aur 7 ko bhi string bana dega.)

Real-life Application:

- Used in string-based calculations, like showing prices with currency symbols.

(Price calculation ke liye jisme "₹" + 500 format hota hai.)

Keywords Explanation:

- Type Conversion: Changing a value from one type to another.
- Concatenation: Joining two strings together.

---

23. Write a JavaScript code for adding new elements dynamically. (4 CO-3)

Definition:

Dynamically adding elements means creating and inserting new HTML elements using JavaScript.
(JavaScript ka use karke bina page reload kiye naye elements add karna.)

Use/Purpose:

- Used in interactive web applications where content updates dynamically.
  (Web pages me naye elements add karne ke liye jaise comment section ya list items.)

Syntax:

javascript

```
let newElement = document.createElement("p");

newElement.innerHTML = "New Paragraph";

document.body.appendChild(newElement);
```

(Yeh code ek naya `<p>` element create karke webpage me insert karega.)

Example Code:

javascript

```
function addElement() {
    let newDiv = document.createElement("div");
    newDiv.innerHTML = "This is dynamically added";
    document.body.appendChild(newDiv);
}
```

(Function call karne par ek naya element add hoga.)

Real-life Application:

- Live notifications in apps like WhatsApp, Gmail.
- Dynamically adding items to a shopping cart.

Keywords Explanation:

- DOM Manipulation: Modifying web page content using JavaScript.
- appendChild(): Adds an element inside another element.

---

24. What are all the looping structures in JavaScript? Explain in detail with example. (8 CO-3)

Definition:

Loops help in executing a block of code multiple times.
(Loop ek code block ko baar-baar chalane ka tareeka hai.)

Use/Purpose:

- Helps in reducing code repetition.
  (Same code likhne ki zaroorat nahi padti.)
- Useful for iterating over arrays, lists, and objects.
  (List ya object ke har item par action lene ke liye.)

Syntax & Example:

1. for loop

javascript

```
for (let i = 0; i < 5; i++) {
    console.log(i);
}
```

(Loop `i` ko 0 se 4 tak print karega.)

2. while loop

javascript

```
let i = 0;
while (i < 5) {
    console.log(i);
    i++;
}
```

(Condition `true` hone tak loop chalega.)

3. do-while loop

javascript

```
let i = 0;
do {
    console.log(i);
    i++;
} while (i < 5);
```

(Pehla execution bina condition check kiye hoga.)

Real-life Application:

- Used in fetching multiple records from a database.
- Automating repeated actions like scrolling a webpage.

Keywords Explanation:

- Iteration: Repeating a block of code.
- Condition: The logic that decides how many times the loop runs.

---

25. What are all the control structures in JavaScript? Explain in detail with example. (8 CO-3)

Definition:

Control structures control the flow of execution of a program.
(Control structures decide karte hain ki program ka flow kaise chalega.)

Use/Purpose:

- Helps in making decisions in a program.
  (Code ko alag-alag conditions ke according execute karne ke liye.)

Syntax & Example:

1. if-else statement

javascript

```
let age = 18;
if (age >= 18) {
    console.log("You are an adult.");
} else {
    console.log("You are a minor.");
}
```

(Agar age 18 ya usse zyada hai toh "You are an adult" print hoga.)

2. switch statement

javascript

```
let day = "Monday";
switch (day) {
    case "Monday":
        console.log("Start of the week!");
        break;
    case "Friday":
        console.log("Weekend is near!");
        break;
    default:
        console.log("Regular day.");
}
```

(Yeh day variable ke hisaab se alag messages print karega.)

Real-life Application:

- Used in handling invalid user inputs.
- Used in building calculators, chatbots, and AI-based decision-making.

Keywords Explanation:

- Conditional Statement: Controls which block of code should execute.
- Break Statement: Stops the execution of a loop or switch case.

Unit-1: Working with HTML5

1. What is the purpose of semantic elements in HTML5? (2 CO-1)

Definition: Semantic elements are HTML tags that provide meaning about the content inside them, making the code more readable and easier to understand.
(Semantic elements wo HTML tags hote hain jo content ka meaning batate hain aur code ko readable banate hain.)

Use/Purpose: These elements help both developers and browsers understand the structure and purpose of the content. They improve SEO (Search Engine Optimization) and make webpages more accessible to screen readers for visually impaired users.
(Ye elements developers aur browsers ko content ka structure samajhne mein madad karte hain. Yeh SEO aur accessibility ko improve karte hain.)

Syntax:

- `<article>`: Defines an independent piece of content.
- `<section>`: Represents a section of the document, often used to group related content.
- `<nav>`: Represents a navigation section.
- `<header>`: Defines a header for a section or the entire document.
- `<footer>`: Represents the footer of a section or the entire document.
  (Yeh kuch important semantic elements hain jo HTML5 mein use hote hain.)
  Example:

```
<article>
  <h2>Article Title</h2>
  <p>This is an article content. It provides valuable
information.</p>
</article>
<section>
  <h2>Section Title</h2>
  <p>This is a section containing related information.</
p>
</section>
```

(Yeh example semantic elements ka use dikhata hai jaise article aur section.)

Real-life Application: Websites like blogs and news sites use these tags to structure content like articles, sections, and navigation.
(Blogging aur news websites mein yeh tags use hote hain content ko structure karne ke liye.)

Keywords Explanation:

- `<article>`: A self-contained piece of content that could stand alone (e.g., blog post, news article).
- `<section>`: A thematic grouping of content, typically with a heading.
- `<nav>`: A container for links that help users navigate through the site.

2. Name two multimedia elements supported by HTML5. (2 CO-1)

Definition: HTML5 supports multimedia elements like audio and video, which can be embedded directly into the webpage without the need for plugins.
(HTML5 mein audio aur video jaise multimedia elements ko directly webpage mein embed kiya ja sakta hai bina plugins ke.)

Use/Purpose: These elements make it easier to add media content like music and videos directly into the webpage. They provide built-in controls for the user to interact with the media (e.g., play, pause).
(Yeh elements media content jaise music aur videos ko directly webpage pe add karne mein madad karte hain aur user ko controls dete hain jaise play, pause.)

Syntax:

- For audio:`html`
  ```
  <audio src="audiofile.mp3" controls></audio>
  ```

- For video:`html`

  ```
  <video src="videofile.mp4" controls></video>
  ```

(Yeh syntax audio aur video files ko embed karne ke liye hain.)

Example:

```
<audio src="song.mp3" controls></audio>
<video src="movie.mp4" controls></video>
```

(Yeh example audio aur video ko embed karta hai webpage mein.)

Real-life Application: Audio and video elements are used in websites for streaming music, videos, or podcasts (e.g., YouTube, Spotify).
(Music streaming websites jaise YouTube aur Spotify mein yeh elements use hote hain.)

Keywords Explanation:

- `<audio>`: Used to embed sound or music on a webpage.
- `<video>`: Used to embed a video file on a webpage.

3. How do you define a required field in an HTML form with an example? (4 CO-1)

Definition: The `required` attribute in an HTML form ensures that a user must fill out a particular input field before the form is submitted.
(HTML form mein `required` attribute ensure karta hai ki user ko form submit karne se pehle ek specific field ko fill karna hoga.)

Use/Purpose: This is used to make certain fields mandatory, such as the name, email, or password fields in a form.
(Yeh use hota hai kuch fields ko mandatory banane ke liye, jaise name, email, ya password.)

Syntax:

```
<input type="text" required>
```
(Yeh syntax ek input field ko required banata hai.)

Example:

```
<form>
  <label for="name">Name:</label>
  <input type="text" id="name" required><br>
  <label for="email">Email:</label>
  <input type="email" id="email" required><br>
  <input type="submit">
</form>
```
(Yeh form name aur email fields ko required banata hai.)

Real-life Application: Used in registration and login forms to ensure essential information is entered.
(Yeh registration aur login forms mein use hota hai taaki zaroori information bharna pade.)

Keywords Explanation:

- `required`: Makes the field mandatory to fill out before submitting the form.

4. List all HTML tags which are new and removed in HTML5. (4 CO-1)

Definition: HTML5 introduced several new tags that improve the structure and accessibility of a webpage. It also removed obsolete tags that were previously used but are no longer needed.
(HTML5 mein naye tags aaye hain jo webpage ki structure aur accessibility ko improve karte hain, aur purane obsolete tags ko remove kar diya gaya hai.)

Use/Purpose: New tags help organize content semantically and improve the user experience. Removed tags were outdated and replaced by newer, more efficient elements.
(Naye tags content ko better organize karte hain aur user experience ko improve karte hain. Purane tags ko replace kiya gaya hai.)

New Tags:

- `<article>`: Represents independent content.

- `<section>`: Represents a section of content.
- `<nav>`: Represents a navigation block.
- `<header>`: Defines the header of a document.
- `<footer>`: Defines the footer of a document.
- `<aside>`: Represents content that is tangentially related to the main content.
- `<mark>`: Highlights text.
  (Yeh kuch naye tags hain HTML5 mein.)

  Removed Tags:
- `<font>`: Used to define font styles, now replaced by CSS.
- `<center>`: Used to center content, now replaced by CSS.
- `<big>`: Used to increase text size, now replaced by CSS.
- `<frame>`: Used to create frames, now deprecated in favor of `<iframe>`.
  (Yeh kuch purane tags hain jo ab remove kar diye gaye hain.)

  Example:

```
<header>
  <h1>My Website</h1>
</header>
```

(Yeh `<header>` tag ka use ek website ke header ko define karne ke liye hota hai.)

Real-life Application: Websites now use semantic elements like `<article>` and `<section>` for better content structure, while older tags like `<font>` are no longer supported.

(Websites ab `<article>` aur `<section>` tags use karti hain better structure ke liye, aur purane tags jaise `<font>` ab supported nahi hain.)

Keywords Explanation:

- `<article>`: A self-contained block of content that could stand alone.
- `<section>`: A group of content with a heading.
- `<header>`: The top section of a page or content area.
- `<footer>`: The bottom section of a page or content area.

5. What is the significance of using placeholder attributes in form inputs with example? (3 CO-1)

Definition: The `placeholder` attribute provides a short hint or description about the expected value in an input field. It appears inside the input field before the user types anything.

(`placeholder` attribute ek chhoti si hint deta hai jo input field mein user ke likhne se pehle dikhayi deti hai.)

Use/Purpose: It helps users understand what kind of information should be entered in

a field, without needing to add extra labels.
(Yeh user ko input field mein kis type ka data daalna chahiye, yeh batata hai bina extra labels ke.)
Syntax:

```
<input type="text" placeholder="Enter your name">
```

(Yeh syntax ek input field dikhata hai jisme placeholder "Enter your name" hai.)
Example:

```
<form>
  <label for="email">Email:</label>
  <input type="email" id="email"
placeholder="example@example.com"><br>
  <input type="submit">
</form>
```

(Yeh example email field ko dikhata hai jisme placeholder "example@example.com" hai.)
Real-life Application: Commonly used in forms for email addresses, phone numbers, or search bars (e.g., Google search).
(Yeh commonly forms mein use hota hai jaise email address, phone number, ya search bar mein.)
Keywords Explanation:

- `placeholder`: A short text inside an input field that provides a hint about the field's content.

6. What is the difference between a canvas and SVG element in HTML5? (4 CO-1)

Feature	Canvas	SVG
Definition	A bitmap-based drawing area that allows dynamic rendering of graphics using JavaScript.	A vector-based format that represents graphics using XML-based markup.
Rendering	Renders graphics pixel by pixel.	Uses mathematical equations to render graphics.
Scalability	Not scalable; resizing causes pixelation.	Scalable; maintains quality when resized.
Performance	Faster for complex graphics and animations.	Slower for complex graphics due to DOM updates.
Use Case	Suitable for games, real-time graphs, and animations.	Suitable for static images like logos, charts, and maps.
Editing	Requires JavaScript for modifications.	Can be edited using CSS and JavaScript.
Interactivity	Limited interactivity; needs scripting.	More interactive; elements can have event handlers.
File Size	Smaller for complex and frequent updates.	Can be larger for complex graphics due to XML structure.
Example	`<canvas id="myCanvas"></canvas>`	`<svg width="100" height="100"><circle cx="50" cy="50" r="40" /></svg>`

7. Discuss the benefits of using semantic elements over traditional HTML elements. (3 CO-1)

Definition: Semantic elements are HTML tags that convey meaning about the content they contain, unlike traditional elements that only focus on structure.

(Semantic elements wo HTML tags hote hain jo content ka meaning bhi batate hain, jabki traditional elements sirf structure pe focus karte hain.)

Use/Purpose:

- Improves readability for developers.
- Enhances accessibility for screen readers.
- Provides better SEO, as search engines understand the structure and context better.

- *Developers ke liye readability improve hoti hai.*
- *Screen readers ke liye accessibility badhati hai.*
- *SEO ko better banata hai kyunki search engines ko structure aur context samajhne mein madad milti hai.)*

  Example:

```
<header>
  <h1>Welcome to My Website</h1>
</header>
<article>
  <h2>Blog Post</h2>
  <p>This is a blog post.</p>
</article>
```

(Yeh example semantic elements ka use dikhata hai, jisme `<header>` aur `<article>` tags use kiye gaye hain.)

Real-life Application: Websites like news sites and blogs use semantic tags like `<header>`, `<footer>`, and `<article>` for better content structure and SEO optimization.

(Websites jaise news aur blog sites semantic tags ka use karte hain content ko structure karne ke liye aur SEO ko optimize karte hain.)

Keywords Explanation:

- `semantic elements`: HTML tags that provide meaning about the content, like `<header>`, `<footer>`, `<article>`.
- `SEO`: Search Engine Optimization, improving visibility on search engines.

8. List and briefly describe at least five different input types in HTML5. (5 CO-1)

Definition: HTML5 introduces various input types to make forms more efficient and user-friendly. Each input type allows for better validation and input handling.

(HTML5 mein kai input types introduce kiye gaye hain jo forms ko zyada efficient aur user-friendly

banate hain.)

Use/Purpose: Input types allow for better data validation, ensuring that the right kind of information is entered.

(Input types data validation ko improve karte hain, taaki sahi type ki information enter ho.)

Syntax and Examples:

- `<input type="text">`: A single-line text field.`html`

  ```
  <input type="text" placeholder="Enter your name">
  ```

- `<input type="email">`: Accepts only email addresses.`html`

  ```
  <input type="email" placeholder="Enter your email">
  ```

- `<input type="password">`: Hides the text entered, typically used for passwords.`html`

  ```
  <input type="password" placeholder="Enter your password">
  ```

- `<input type="date">`: Allows the user to pick a date.`html`

  ```
  <input type="date">
  ```

- `<input type="number">`: Accepts only numeric input.`html`

  ```
  <input type="number" placeholder="Enter your age">
  ```

(Yeh examples different input types ko dikhate hain, jaise text, email, password, date, aur number fields.)

Real-life Application: Used in forms for entering names, email addresses, dates, numbers, and passwords (e.g., online registration forms, login pages).

(Yeh forms mein use hota hai jaise online registration aur login pages mein data enter karne ke liye.)

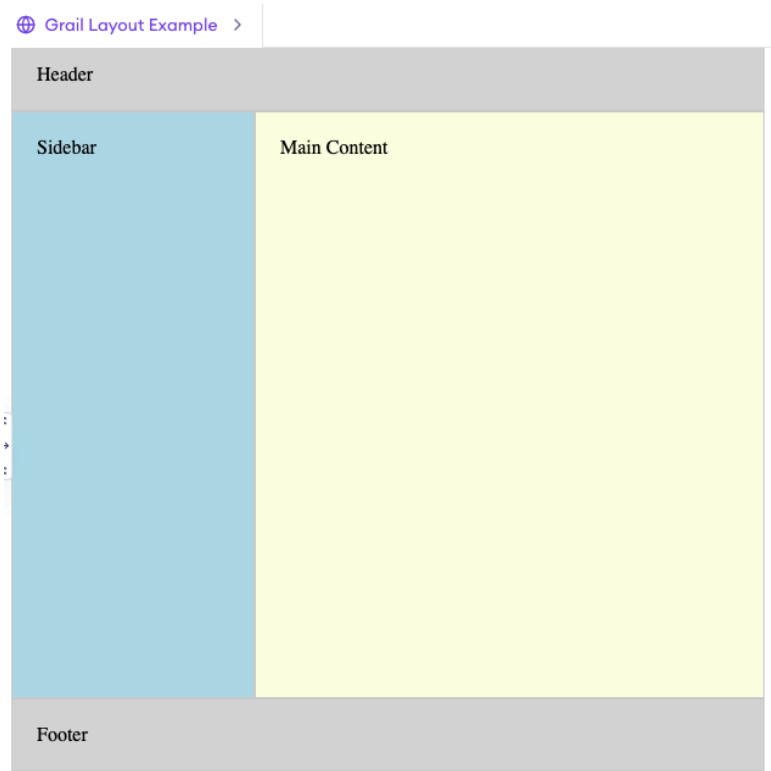
Keywords Explanation:

- `type="text"`: A field for entering text.
- `type="email"`: A field that accepts only email format.
- `type="password"`: A field for entering hidden text like passwords.
- `type="date"`: A field for selecting a date.
- `type="number"`: A field for entering numeric values.

9. What is Grail layout in HTML? (8 CO-1)

Definition: The Grail layout is a simple yet flexible web design layout used for creating a modern web page structure. It consists of a header, main content area, sidebar, and footer, typically arranged in a grid-like structure.
(Grail layout ek simple aur flexible web design layout hai jo modern webpage structures ke liye use hota hai. Ismein header, main content area, sidebar, aur footer hote hain.)

Use/Purpose: It helps in designing web pages with a clear structure and makes it easier to create responsive layouts with a header and footer that remain in place while content changes.
(Yeh web pages ko clear structure dene mein madad karta hai aur responsive layouts banane ko aasan banata hai, jisme header aur footer fixed rehte hain jabki content change hota hai.)



Syntax:

```
<header>Header content</header>
<main>
  <article>Main content</article>
  <aside>Sidebar content</aside>
</main>
<footer>Footer content</footer>
```

(Yeh syntax ek simple Grail layout ko dikhata hai, jisme header, main content, sidebar aur footer hote hain.)

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Grail Layout Example</title>
  <style>
    body {
      display: grid;
      grid-template-rows: 60px 1fr 60px;
      grid-template-columns: 200px 1fr;
      grid-template-areas: "header header" "sidebar main"
"footer footer";
      height: 100vh;
    }
    header, footer, main, aside {
      padding: 20px;
      border: 1px solid #ccc;
    }
    header { grid-area: header; background-color:
lightgray; }
    footer { grid-area: footer; background-color:
lightgray; }
    aside { grid-area: sidebar; background-color:
lightblue; }
    main { grid-area: main; background-color:
lightyellow; }
  </style>
</head>
<body>
  <header>Header</header>
  <aside>Sidebar</aside>
  <main>Main Content</main>
  <footer>Footer</footer>
</body>
</html>
```

(Yeh example Grail layout ka ek basic implementation dikhata hai jisme header, sidebar, main content aur footer ko grid layout mein organize kiya gaya hai.)

Real-life Application: Commonly used in modern websites, particularly blogs and

news websites, to provide a well-structured and easy-to-navigate layout.
(Yeh layout modern websites, jaise blogs aur news websites mein use hota hai, jisme structure clear aur navigation aasan hota hai.)

Keywords Explanation:

- `grid layout`: A system for creating a layout using rows and columns.
- `grid-template-areas`: Defines areas within the grid for different elements.
-

10. Explain HTML `<FORM>` element with all form attributes with example. (8 CO-1)

Definition: The `<form>` element is used to collect user input through fields such as text, radio buttons, checkboxes, and more. It sends the collected data to the server for processing.
(`<form>` element user input collect karne ke liye use hota hai jaise text, radio buttons, checkboxes, aur aur bhi fields, aur collected data ko server par bhejne ke liye.)

Use/Purpose: Forms are essential in allowing users to interact with websites by submitting data, for tasks like registrations, logins, feedback, etc.
(Forms websites ke liye zaroori hote hain jisme user data submit kar sake, jaise registrations, logins, feedback, etc.)

Syntax:

```
<form action="/submit_form" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br>
  <input type="submit" value="Submit">
</form>
```

(Yeh example ek basic form ko dikhata hai jisme name aur email ko collect karne ka option diya gaya hai.)

Attributes:

- `action`: Specifies where to send the form data when the form is submitted.
- `method`: Defines the HTTP method (GET or POST) used to send form data.
- `name`: Identifies the form or input field uniquely.
- `enctype`: Specifies the encoding type for form data (used for file uploads).
- `target`: Determines where to display the response (same window, new window, etc.).
- *`action`: Yeh batata hai ki form data kaha bhejna hai.*
- *`method`: Yeh batata hai ki data send karne ke liye GET ya POST method ka use hoga.*

- `name`: *Form ya input field ko unique identify karta hai.*
- `enctype`: *File upload ke liye form data ko kaise encode karna hai, yeh batata hai.*
- `target`: *Yeh batata hai ki response ko kis window mein dikhana hai.)*

Real-life Application: Forms are used in almost every website for tasks such as creating accounts, signing in, making purchases, or submitting feedback.
(Forms har website mein use hote hain jaise accounts banane, sign in karne, purchases karne, ya feedback submit karne ke liye.)

Keywords Explanation:

- `action`: The URL where form data is sent.
- `method`: The method used to send the form data (GET or POST).
- `input`: An element for entering data.

11. Create a basic student detail form that will take input data, including First Name, Last Name, Gender, Enrollment ID, Designation, and Phone Number. (8 CO-1)

Definition: This task asks to create a simple form where users can enter personal details such as name, gender, enrollment ID, etc.
(Yeh task ek simple form banane ko keh raha hai jisme user apne personal details jaise name, gender, enrollment ID, etc. enter karega.)

Use/Purpose: It's used to collect personal details from students or users for registration or profiles.
(Yeh form students ya users se unke personal details collect karne ke liye use hota hai, jaise registration ya profiles ke liye.)

Example:

```
<form>
  <label for="first-name">First Name:</label>
  <input type="text" id="first-name" name="first-
name"><br>

  <label for="last-name">Last Name:</label>
  <input type="text" id="last-name" name="last-name"><br>

  <label for="gender">Gender:</label>
  <input type="radio" id="male" name="gender"
value="male"> Male
  <input type="radio" id="female" name="gender"
value="female"> Female<br>

  <label for="enrollment-id">Enrollment ID:</label>
  <input type="text" id="enrollment-id" name="enrollment-
id"><br>
```

```
  <label for="designation">Designation:</label>
  <input type="text" id="designation"
name="designation"><br>

  <label for="phone-number">Phone Number:</label>
  <input type="tel" id="phone-number" name="phone-
number"><br>

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

(Yeh example ek student detail form ka hai, jisme first name, last name, gender, enrollment ID, designation, aur phone number fields hai.)

Real-life Application: Forms like this are used for student registrations, sign-ups, or surveys.

(Aise forms student registration, sign-ups, ya surveys ke liye use hote hain.)

Keywords Explanation:

- `radio`: A form element that allows the user to select one option from multiple choices.
- `tel`: A field for entering a phone number.

12. Explain any five Git commands with syntax and example. (5 CO-1)

Definition: Git is a version control system used to track changes in code and manage source code history. Several commands are used to interact with Git repositories.

(Git ek version control system hai jo code mein changes track karne aur source code history manage karne ke liye use hota hai.)

Use/Purpose: Git commands help in managing repositories, checking status, committing changes, etc.

(Git commands repository manage karne, status check karne, aur changes commit karne mein madad karti hain.)

Commands:

1. `git init`
    - **Purpose**: Initializes a new Git repository.
    - **Syntax**: `git init`

- **Example**:bash

```
git init
```

- **Real-life Application**: Used when starting a new project.
  (: Jab naya project shuru karte hain, tab yeh command use hoti hai.)

2. `git clone`
   - **Purpose**: Clones an existing repository to your local machine.
   - **Syntax**: `git clone <repository-url>`
   - **Example**:bash

```
git clone https://github.com/username/repo.git
```

   - **Real-life Application**: When you want to work on an existing project, you clone the repository.
     (Agar aap existing project par kaam karna chahte hain, toh aap repository clone karte hain.)
3. `git status`
   - **Purpose**: Displays the current state of the repository, such as staged, unstaged, or untracked changes.
   - **Syntax**: `git status`
   - **Example**:bash

```
git status
```

   - **Real-life Application**: Useful for checking what changes have been made and what is ready to be committed.
     (Yeh command kaam aati hai jab aapko yeh dekhna ho ki kaunse changes kiye gaye hain aur kaunse commit hone wale hain.)

4. `git commit`
    - **Purpose**: Commits the changes to the local repository.
    - **Syntax**: `git commit -m "Commit message"`
    - **Example**:bash

      ```
      git commit -m "Updated readme file"
      ```
    - **Real-life Application**: Saves the changes made to the repository along with a message describing the change.
      (Yeh command changes ko local repository mein save karti hai aur ek message ke saath jo change kiya gaya ho.)
5. `git push`
    - **Purpose**: Pushes committed changes to a remote repository.
    - **Syntax**: `git push origin <branch-name>`
    - **Example**:bash

      ```
      git push origin main
      ```
    - **Real-life Application**: Used when you want to send your local changes to the GitHub repository.
      (Jab aap apne local changes ko GitHub repository pe bhejna chahte hain, tab yeh command use hoti hai.)

Real-life Application: Git commands are widely used in software development for collaboration, tracking changes, and managing version control.
(Git commands software development mein collaboration, changes track karne aur version control manage karne ke liye use hoti hain.)

Keywords Explanation:

- `repository`: A storage space for your code and files.
- `commit`: To save your changes with a message.
- `push`: To send local changes to a remote repository.

13. How do you create a new repository on GitHub? (3 CO-1)

Definition: A GitHub repository is where you can store your code and collaborate with others. You can create a new repository to start a project.

(GitHub repository wo jagah hai jahan aap apna code store kar sakte hain aur dusron ke saath collaborate kar sakte hain. Nayi repository banani hoti hai jab aap project shuru karte hain.)

Use/Purpose: Creating a repository helps to organize your code and track changes.

(Repository banane se aapka code organize hota hai aur changes track kiye jaate hain.)

Steps:

1. Go to GitHub and log in to your account.
2. Click on the "New" button or "+ New repository" on the homepage.
3. Give a name to your repository.
4. Select whether it should be public or private.
5. Initialize the repository with a README (optional).
6. Click "Create repository."

   (Yeh steps aapko GitHub pe nayi repository banane ke liye follow karne padte hain.)

   Real-life Application: This is used by developers to host and share code projects.

   (Yeh step developers use karte hain apne code projects ko host aur share karne ke liye.)

Keywords Explanation:

- `repository`: A place to store your code.
- `README`: A file that describes the repository's purpose and how to use it.

14. What are branches in Git and why are they used? (3 CO-1)

Definition: Branches in Git allow you to create separate lines of development within the same project, so you can work on different features or bug fixes without affecting the main codebase.

(Git mein branches aapko ek hi project mein alag-alag development lines banane ki suvidha deti hain, taaki aap different features ya bug fixes par kaam kar sakein bina main code ko affect kiye.)

Use/Purpose: They are used to isolate work on different tasks, allowing parallel development without conflicts.

(Yeh different tasks pe kaam ko alag rakhne ke liye use hoti hain, taaki parallel development possible ho aur conflicts na ho.)

Syntax:

```
git branch <branch-name>
git checkout <branch-name>
```

(Yeh commands new branch banane aur us branch pe switch karne ke liye use hoti hain.)

Example:

```
git branch feature-1
git checkout feature-1
```
(Yeh example ek branch `feature-1` banane aur uspe switch karne ka hai.)

Real-life Application: Branches are essential for teams working on multiple features at once, allowing them to merge their work later without disrupting others' code.
(Jab teams ek saath kai features pe kaam karti hain, tab branches unko apna kaam alag rakhne aur baad mein merge karne mein madad karti hain.)

Keywords Explanation:

- `branch`: A separate line of development.
- `checkout`: To switch to a specific branch.

15. Explain the concept of Pull and Push Requests in GitHub. (4 CO-1)

Definition: Pull and push requests are processes for contributing changes to a repository. Pull requests allow you to propose changes, and push requests are used to upload changes to a repository.
(Pull aur push requests wo processes hain jisme aap apne changes ko repository mein contribute karte hain. Pull request changes propose karne ke liye hota hai aur push request changes ko repository mein upload karne ke liye use hota hai.)

Use/Purpose:

- **Pull Request**: Used to propose code changes to be merged into the main repository.
- **Push Request**: Used to upload local changes to the remote repository.
  (Pull request code changes ko main repository mein merge karne ke liye propose karte hain, aur push request local changes ko remote repository mein upload karne ke liye hota hai.)
  Syntax:
- **Pull Request**: Create a pull request on GitHub after pushing changes.
  (Pull request banane ke liye GitHub pe changes push karne ke baad request create karte hain.)
- **Push Request**:

```
git push origin <branch-name>
```
(Push request mein local branch ko remote repository mein push karte hain.)

Example:

- **Pull Request**:
  After pushing your changes to a new branch, you can create a pull request on GitHub to merge your changes into the main branch.
- **Push Request**:

```
git push origin feature-branch
```

Real-life Application: Pull requests are commonly used in open-source projects where contributors propose changes and the project owner reviews and merges them.
(Pull requests open-source projects mein commonly use hoti hain jahan contributors apne changes propose karte hain aur project owner unko review karke merge karta hai.)

Keywords Explanation:

- `Pull Request`: A proposal to merge code changes into another branch.
- `Push Request`: Uploading your changes to a remote repository.

16. Define the following terms: (5 CO-1)

1. **Canvas**
    - **Definition**: The HTML `<canvas>` element is used to draw graphics on the web, such as shapes, images, or animations.
      (HTML `<canvas>` element ka use web par graphics draw karne ke liye hota hai jaise shapes, images ya animations.)
    - **Use/Purpose**: Used for creating dynamic graphics, like games or charts.
      (Yeh dynamic graphics banane ke liye use hota hai, jaise games ya charts.)
    - **Example**:
2.

```
<canvas id="myCanvas" width="500" height="500"></
canvas>
```

3.
    - **Real-life Application**: Games, data visualization, and interactive animations use the canvas.
      (Games, data visualization aur interactive animations mein canvas ka use hota hai.)
4. **Figma**
    - **Definition**: Figma is a cloud-based design tool used for creating web and mobile user interfaces.
      (Figma ek cloud-based design tool hai jo web aur mobile user interfaces banane ke liye use hota hai.)
    - **Use/Purpose**: Used for collaborative design work, prototyping, and creating UI/UX.
      (Yeh collaborative design work, prototyping aur UI/UX design ke liye use hota hai.)
    - **Real-life Application**: UI/UX designers use Figma to design and prototype websites and apps.
      (UI/UX designers Figma ka use websites aur apps design karne ke liye karte hain.)
5. **Hierarchy**

- **Definition**: Hierarchy refers to the system of organizing elements in a structured way based on their importance.
  (Hierarchy ka matlab hai elements ko ek structured tareeke se organize karna unki importance ke hisaab se.)
- **Use/Purpose**: Helps in organizing content in web design, ensuring a logical flow.
  (Yeh content ko web design mein organize karne mein madad karta hai, taaki logical flow ban sake.)
- **Example**: Using headings (H1, H2) to create a hierarchy of content.
  (*Headings (H1, H2) ka use karke content ki hierarchy create karte hain.)*

6. Components

- **Definition**: In web design, components are reusable UI elements such as buttons, cards, or navigation bars.
  (Web design mein components wo reusable UI elements hote hain jaise buttons, cards ya navigation bars.)
- **Use/Purpose**: Components help to create consistent and modular interfaces.
  (Components consistency aur modular interfaces banane mein madad karte hain.)
- **Example**: A button or card component that can be reused across pages.
  (Button ya card component jo pages mein reuse kiya jaa sakta hai.)

7. Prototyping

- **Definition**: Prototyping is the process of creating a preliminary version of a product or interface to test and validate ideas.
  (Prototyping wo process hai jisme product ya interface ka ek preliminary version banaya jata hai taaki ideas test aur validate kiye ja sakein.)
- **Use/Purpose**: Used for visualizing and testing the functionality of a design.
  (Yeh design ki functionality ko visualize aur test karne ke liye use hota hai.)
- **Example**: Creating a prototype of a website before starting the full development.
  (Full development shuru karne se pehle website ka prototype banana.)