RAJ’S ASSIGNMENT

HTML

1.Are the HTML tags and elements the same thing?

HTML Tags: HTML tags are the building blocks of HTML documents. They are used to define the structure and appearance of content on a webpage. For example, `<div>` is an opening tag for a paragraph, and `</div>` is a closing tag for a paragraph.

HTML Elements: An HTML element consists of an opening tag, the content within the tag, and a closing tag. It represents a complete unit of content on a webpage. The opening tag and closing tag, along with the content in between

HTML tags are individual instructions that define the type and behavior of content, while HTML elements are complete units that encompass the opening tag, content,

2.What are tags and attributes in HTML?

HTML tags are the basic building blocks of an HTML document. They are used to define the structure and semantics of the content within a webpage.

For Example:

```
<span>This is a paragraph.</span>
```

HTML attributes provide additional information about an HTML element. They are placed within the opening tag of an element and are used to modify the element's behavior or appearance.

For Example

```
<script src=""> </script>
```

3.Void elements, also known as self-closing or empty elements

Void elements, also known as self-closing or empty elements.

```
<p>This is the first line.<br>This is the second line.</p>
```

4.What are HTML Entities? With Example.

HTML entities are special codes used to represent characters that have special meanings in HTML but need to be displayed as their actual characters on a webpage.

For Example:

```
<p>This is an & example.</p>
```

5.What are different types of lists in HTML? With Example.

In HTML, there are three main types of lists.

Unordered Lists

```
<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ul>
```

Ordered Lists

<ol>
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
</ol>

Definition Lists

<dl>
<dt>HTML</dt>
<dd>Hypertext Markup Language</dd>
<dt>CSS</dt>
<dd>Cascading Style Sheets</dd>
</dl>

6.What is the ‘class’ attribute in HTML? With Example.

CSS classes define a set of styles that can be applied to multiple elements on a webpage.

By using the `class` attribute.

For Example

HTML

<p class=”one”>lorem ipsum……….</p>

CSS

.one{

color:red;

}

7.What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements? With Example

ID:The `id` attribute is used to uniquely identify a single HTML element on a webpage

For Example:

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

class: The `class` attribute is used to define a group of elements that share a common styling or behavior.Multiple elements can have the same `class` value

For Example:

<p class="two">This is an important paragraph.</p>

8.What are the various formatting tags in HTML?

HTML provides various formatting tags that allow you to control the appearance and presentation of text and content on a webpage.

Bold Text

<p>This is <b>lorem ipsum</b>.</p>

Italic Text

<p>This is <i>lorem ipsum</i>.</p>

Underlined Text

<p>This is <u>lorem ipsum</u>.</p>

Strikethrough Text

```
<p>This is <s>strikethrough</s> text.</p>
```

9.How is Cell Padding different from Cell Spacing? With Example

Cell padding is the space between the content of a table cell and its borders.

For Example

```
<table cellpadding="10">
<tr>
<td>one</td>
<td>two</td>
</tr>
</table>
```

10.How can we club two or more rows or columns into a single row or column in an HTML table? With Example.

To group two or more rows or columns into a single row or column in an HTML table

rowspa

`colspan`

```
<table>
<tr>
<td rowspan="2">Merged Cell</td>
<td>Row 1</td>
<td>Row 1</td>
</tr>
<tr>
<td>Row 2</td>
<td>Row 2</td>
</tr>
</table>
```

Merging Columns

```
<table>
<tr>
<td>Column 1</td>
<td colspan="2">Merged Cell</td>
<td>Column 4</td>
</tr>
<tr>
<td>Row 2</td>
<td>Row 2</td>
<td>Row 2</td>
<td>Row 2</td>
</tr>
</table>
```

11.What is the difference between a block-level element and an inline element?

Block-level elements are HTML elements that create a block of content that spans the entire width of its parent container.

```
<div>lorem ipsum…………….</div>
```

Inline elements are HTML elements that do not create a new line and only take up as much width as necessary to fit their content.

```
<a href="#">Link</a>
```

12.How to create a Hyperlink in HTML? With Example.

The `<a>` element allows you to link to other web pages, documents, or resources on the internet.

o create a hyperlink (or a link) in HTML the `<a>` element.

For Example:

```
<a href="https://www.google.com">Visit Example Website</a>
```

13.What is the use of an iframe tag? With Example.

The `<iframe>` (inline frame) tag in HTML is used to embed another HTML document or web page within the current document.

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/8JhdiDiZMNE" title="YouTube video player"
frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>
```

13.What is the use of a span tag? Explain with example?

The `<span>` tag in HTML is a inline-level element that is used to apply styling or scripting to a specific portion of text within a larger block of content. `<span>` tag doesn't create a new line or change the layout.

```
<p>This is a <span style="font-size: 10px;">red</span> word in a sentence.</p>
```

14.How to insert a picture into a background image of a web page? With Example.

To insert a picture (an image) into the background of a web page, you can use CSS to set the background image property.

For Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url('XYZ.jpg');
background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed;
}
</style>
</head>
<body>
<h1>Hello world</h1>
<p>lorem ipsum.</p>
</body>
</html>
```

15.How are active links different from normal links?

Active links refer to the appearance of a hyperlink when it is being actively clicked or interacted with.This state is often temporary and is triggered while a user is clicking on the link.

For Example:

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

```
<head>
<style>
a {
color: blue;
text-decoration: none;
}
```

```
a:active {
  color: red;
}
```

```
</style>
</head>
<body>
<a href="#">Home</a>
</body>
</html>
```

16.What are the different tags to separate sections of text.

there are various tags and elements that you can use to separate sections of text and structure your content in a meaningful way.

```
<div class="one">
<h2>Section Title</h2>
<p>This is the content of the section.</p>
</div>
```

17.What is SVG?

SVG stands for Scalable Vector Graphics.It is a markup language used to create two-dimensional vector graphics.

18.What is difference between HTML and XHTML?

HTML (Hypertext Markup Language) and XHTML (Extensible Hypertext Markup Language) are both markup languages used to structure and present content on the web.

19.What are logical and physical tags in HTML?

In HTML, the terms "logical tags" and "physical tags" are often used to describe two different approaches to structuring content within a webpage.

<header>

<nav>

<main>

<article>

<section>

<aside>

<footer>

Physical Tags (Presentational Tags):

Physical tags, also referred to as presentational tags.

- `<b>`
- `<i>`
- `<u>`
- `<font>`
- `<center`

CSS

1.What are the benefits of using CSS?

Cascading Style Sheets (CSS) is a powerful technology used in web development to style and format the presentation of HTML content.CSS offers numerous benefits that enhance the design, usability, and maintainability of websites.

Responsive Design

Flexibility and Customization

Accessibility

2.What are the disadvantages of CSS ?

CSS, while essential for styling web content, has its share of challenges. Cross-browser compatibility issues can lead to inconsistent rendering, and its specificity rules can cause unexpected style conflicts.Debugging problems can be intricate, particularly when styles are inherited or overridden multiple times.

3.What is the **CSS3:** CSS3 introduced properties for box shadows, border images, and flexible box layouts (Flexbox), which significantly improved layout capabilities.

difference between CSS2 and CSS3?

CSS2: CSS2 had a limited set of properties for styling borders and backgrounds.

CSS2: The box model in CSS2 was relatively basic, and it didn't account for things like box shadows, border images, or flexible box layouts.

CSS2: CSS2 didn't have built-in support for responsive design and adapting layouts to different screen sizes.

CSS3: CSS3 introduced additional properties for creating complex border styles, rounded corners, gradients, and more advanced background effects.

CSS3: CSS3 introduced additional properties for creating complex border styles, rounded corners, gradients, and more advanced background effects.

4.Name a few CSS style components ?

CSS style components, also known as CSS properties.

Color

Font Family

Font Size

Margin

Padding

5.What do you understand by CSS opacity?

CSS opacity is a property that allows you to control the transparency level of an element.

For Example:

```
.demo{
opacity: 0;
transition: opacity 1s;
}
```

6.How can the background color of an element be changed?

You can change the background color of an element using CSS by applying the `background-color` property to that element.

For Example:

```
demo {
background-color: red;
}
```

7.How can image repetition of the backup be controlled?

Image repetition in the background can be controlled using the `background-repeat` property in CSS.

1. **repeat:** The background image is repeated both horizontally and vertically (default behavior).
2. **repeat-x:** The background image is repeated only horizontally.
3. **repeat-y:** The background image is repeated only vertically.
4. **no-repeat:** The background image is not repeated in either direction.

8.What is the use of the background-position property?

The `background-position` property is a CSS (Cascading Style Sheets) property used to control the positioning of a background image within an HTML element.

```
background-position: 100px;
```

9.Which property controls the image scroll in the background

The property that controls the scrolling behavior of a background image is the `background-attachment` property in CSS. This property determines whether the background image scrolls with the content or remains fixed as the content is scrolled.

10.Why should background and color be used as separate properties?

Using background-related properties (like `background-color` and `background-image`) and color properties (like `color`) separately is advantageous because it provides greater flexibility and control over the visual appearance of an element.

Layering and Stacking: Separating background properties from color properties allows you to layer different visual elements more effectively.

Fallbacks and Progressive Enhancement: Separation of color and background properties makes it simpler to provide fallbacks and enhance user experience.

Accessibility: Keeping color and background properties separate can help improve accessibility.

11.How to center block elements using CSS1?

1. The `position: absolute` property positions the element absolutely within its containing element.
2. `top: 50%;` and `left: 50%;` move the top-left corner of the element to the center of its containing element.

3. `margin-top: -50px;` and `margin-left: -50px;` adjust the element's position by moving it back half of its height and width respectively. This effectively centers the element because you are taking into account its dimensions.
4. `width` and `height` set the dimensions of the centered element.

```
<!DOCTYPE html>
<html>
<head>
<style>
.centered {
position: absolute;
top: 50%;
left: 50%;
margin-top: -50px;
margin-left: -50px; width: 100px;
height: 100px;
background-color: lightblue;
}
</style>
</head>
<body>

<div class="centered">
Centered Block
</div>

</body>
</html>
```

12.How to maintain the CSS specifications?

1. **Follow Best Practices**: Adhere to CSS best practices to ensure maintainability, performance, and compatibility. These practices include using external stylesheets, avoiding inline styles, using meaningful class and ID names, and optimizing CSS selectors.
2. **Separation of Concerns**: Follow the principle of separation of concerns by separating your HTML structure, CSS styles, and JavaScript functionality. This makes it easier to update and maintain each aspect independently.
3. **Version Control**: Use version control systems like Git to track changes to your CSS files. This allows you to revert to previous versions if needed and collaborate with team members effectively.

13.What are the ways to integrate CSS as a web page?

Inline Styles: Inline styles are added directly to HTML elements using the `style` attribute. This method applies styles only to that specific element.

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

nternal Stylesheet: An internal stylesheet is placed within the `<head>` section of the HTML document using the `<style>` tag. It applies styles to the entire page and keeps them separate from the HTML content.

```
<head>
<style>
p {

font-size: 16px;
}
</style>
```

```
</head>
<body>
<p>lorem ipsum……….</p>
</body>
```

External Stylesheet: An external stylesheet is a separate `.css` file that contains all the CSS rules. It's linked to the HTML document using the `<link>` element within the `<head>` section.

```
<head>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<p>lorem ipsum……….</p>
</body>
```

14.What is embedded style sheets?

An embedded stylesheet, also known as an internal stylesheet, is a method of including CSS (Cascading Style Sheets) rules directly within the `<style>` element of an HTML document. This allows you to define styles that will be applied to the HTML elements on that specific web page.

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

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

```
    .highlight {
        background-color: yellow;
    }
</style>
```

```
</head>
<body>
<p>This is a paragraph with blue text.</p>
<p class="highlight">This paragraph has a yellow background.</p>
</body>
</html>
```

15.What are the external style sheets?

External stylesheets are separate CSS (Cascading Style Sheets) files that contain all the styling rules for a website. Instead of including styles directly within HTML documents, you link to these external stylesheet files from your HTML.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
```

```
<p>This is a paragraph with blue text.</p>
<p class="highlight">This paragraph has a yellow background.</p>
</body>
</html>
```

css

```
/* styles.css */
p {
color: blue;
font-size: 16px;
}

.highlight {
background-color: yellow;
}
```

16.What are the advantages and disadvantages of using external style sheets?

Advantages

1. **Reusability:** Once you create a well-designed external stylesheet, you can reuse it across multiple web pages. This ensures consistent styling and reduces duplication of code.
2. **Consistency:** External stylesheets allow you to apply a uniform look and feel to your entire website, ensuring that design elements are consistent across pages.
3. **Ease of Maintenance:** Making changes to the design or layout of your website becomes much simpler when using external stylesheets. You only need to update one file, and the changes will be reflected across all linked pages.

disadvantages

1. **Additional HTTP Request:** Loading an external stylesheet requires an additional HTTP request, which can slightly increase page load times, especially on slower connections.
2. **Dependency:** If the external stylesheet fails to load or is mistakenly linked, the styling for your web pages may be compromised.
3. **Rendering Delay:** In some cases, a large external stylesheet may cause a slight delay in rendering the page, particularly if it's placed in the `<head>` section and blocks rendering.

16.What is the meaning of the CSS selector?

A CSS selector is a pattern that is used to select and target specific HTML elements in order to apply styling rules to them. Selectors are a fundamental part of CSS and play a crucial role in determining which elements on a web page should be affected by the defined styles.

```
p {

color:red;
}

#header {
background-color:blue;
}
```

17.What are the media types allowed by CSS

1. **all**: This is the default media type. It applies styles to all devices.

2. **screen**: This media type is used for computer screens, tablets, and smartphones. It's the most common media type for web styles.
3. **print**: Styles with this media type are intended for printed pages. It removes unnecessary elements like navigation bars and backgrounds, focusing on content.

```
@media screen {
body {
font-size: 16px;
}
}
```

```
@media print {
body {
font-size: 12pt;
}
}
```

18.What is the rule set

A rule set, often referred to as a CSS rule or a style rule, is a fundamental concept in Cascading Style Sheets (CSS). It defines how a specific HTML element or a group of elements should be styled.

```
p {
color: blue;
font-size: 16px;
margin-top: 10px;
}
```

HTML 5

1.What are the new tags added in HTML5?

HTML5 introduced a variety of new elements and attributes to enhance the structure and functionality of web pages. Here are some of the notable new elements introduced in HTML5:

1. **<header>**: Represents a container for introductory content or a set of navigational links.
2. **<nav>**: Defines a section of navigation links, often used for menus, navigation bars, or similar content.
3. **<section>**: Represents a thematic grouping of content within a document, such as chapters, tabs, or blocks of related content.
4. **<article>**: Represents a self-contained composition in a document, such as a blog post, news story, or forum post.
5. **<aside>**: Defines content that is tangentially related to the content around it, often used for sidebars, pull quotes, or advertisements.
6. **<footer>**: Represents a container for the footer of a section or the document as a whole, typically containing metadata, copyright information, and contact details.
7. **<main>**: Specifies the main content of a document, excluding headers, footers, and sidebars.
8. **<figure>**: Represents any content that is referenced from the main content, such as images, videos, charts, and code examples.
9. **<figcaption>**: Provides a caption or description for a <figure> element.
10. **<time>**: Represents a specific period in time, such as dates, times, or durations.

11. **<mark>**: Highlights text within the context of its surrounding content.
12. **<progress>**: Represents the completion progress of a task.
13. **<meter>**: Represents a scalar measurement within a known range, such as disk usage or ratings.
14. **<details>**: Creates a disclosure widget that can be used to show or hide additional content.
15. **<summary>**: Provides a visible heading for a <details> element.
16. **<datalist>**: Defines a predefined list of options for input elements, typically used with <input> elements of type "text" or "search."

2.How to embed audio and video in a webpage?

Embedding Audio:

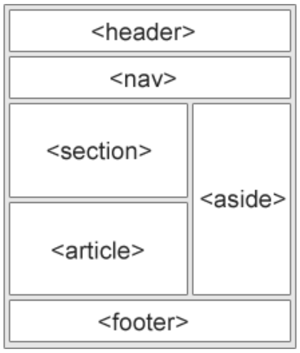
```
<audio controls>
<source src="your-audio-file.mp3" type="audio/mpeg">
</audio>
```

Embedding Video:

```
<video controls>
<source src="your-video-file.mp4" type="video/mp4">
Your browser does not support the video element.
</video>
```

3.Semantic element in HTML5?

In HTML5, semantic elements are HTML elements that carry a meaning or convey a specific type of structural information about the content they enclose.



4.Canvas and SVG tags

`<canvas>` element, you use JavaScript to draw and update the content. This provides a lot of flexibility and control, making it suitable for creating complex animations and interactive graphics.

`<svg>` element, on the other hand, is a vector graphics format that allows you to create scalable, resolution-independent graphics. It uses XML-based markup to define shapes, lines, text, and other graphical elements.

JAVASCRIPT

1.What is JavaScript

JavaScript is a versatile and widely-used programming language primarily known for its role in web development. It allows developers to add interactivity, dynamic behavior, and advanced functionality to websites and web applications. JavaScript is a scripting language, which means it's executed by the browser (or other runtime environments) on the client side, directly in response to user interactions or other events.

Web Development: JavaScript is a fundamental part of front-end web development. It's used to manipulate and modify the content of web pages in real-time, create interactive user interfaces, handle form submissions,

Dynamic Content: JavaScript allows developers to change and update the content of a webpage without requiring the user to navigate away.

2.What is the use of isNaN function?

The `isNaN()` function in JavaScript is used to determine whether a value is "Not-a-Number" (NaN). NaN is a special value in JavaScript that represents an undefined or unrepresentable value resulting from arithmetic operations.

For Example

```
isNaN(value);
```

3.What is negative Infinity?

In JavaScript, negative Infinity is a special numeric value that represents negative infinity. It is one of the values in the IEEE 754 floating-point standard, which JavaScript uses to represent numbers. Infinity, whether positive or negative, is used to represent values that are beyond the limits of representable numbers.

```
const negative_value= -Infinity;
```

4.Which company developed JavaScript?

JavaScript was developed by Netscape Communications Corporation, which is now known as Netscape Communications or simply Netscape. It was created by Brendan Eich, who was an engineer at Netscape. The language was originally developed under the name "Mocha," which was later renamed to "LiveScript," and finally to "JavaScript" to capitalize on the popularity of Java at the time.

5.What are undeclared and undefined variables?

"Undeclared" and "undefined" are terms used to describe different states of variables in programming languages like JavaScript.

1. **Undeclared Variables:**
   An undeclared variable is a variable that has been used in the code without being explicitly declared using a `var`, `let`, or `const` statement.
2.

```
x = 10;
console.log(x);
```

result 10;

Undefined Variables:

An undefined variable is a variable that has been declared but has not been assigned a value. In JavaScript, when a variable is declared but not assigned a value, it automatically takes the value `undefined`.

```
let demo;
console.log(demo); // Outputs undefined
```

6.Write the code for adding new elements dynamically

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <ul id="myList">
        <li>Apple</li>
        <li>banan</li>
        <li>Graps</li>
      </ul>

      <button id="button">Add Item</button>
<script>
    // Get references to the elements
const myList = document.getElementById("myList");
const button = document.getElementById("button");

// Add a click event listener to the button
button.addEventListener("click", function () {
  // Create a new list item element
  const newItem = document.createElement("li");

  // Set the text content of the new item
  const newItemText = document.createTextNode("pineApple");
  newItem.appendChild(newItemText);

  // Append the new item to the list
  myList.appendChild(newItem);
});

</script>
</body>
</html>
```

7.What is the difference between ViewState and SessionState?

`ViewState` is a feature in ASP.NET that allows developers to persist state information across postbacks (when a form is submitted to the server and the page is reloaded).

`ViewState` data is encoded and included in the HTML output of the page. When the page is submitted back to the server, the `ViewState` data is sent along with the request.

`SessionState` is another feature in ASP.NET that allows you to store and manage user-specific data across multiple requests. Unlike `ViewState`, `SessionState` data is stored on the server, and a unique session ID is associated with each user's session.

8.What is === operator?

The `===` operator in JavaScript is called the "strict equality operator." It is used to compare two values for equality without performing type coercion.

9.How can the style/class of an element be changed?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
```

```
<button id="myButton">Lorem ipsum dolor sit amet consectetur adipisicing elit. A dignissimos aliquam vero aut ad voluptatem enim quos? Hic


    <script>
    const myButton = document.getElementById("myButton");
    myButton.addEventListener("click", function() {
        myButton.style.backgroundColor = "blue";
        myButton.style.color = "white";
        myButton.style.fontSize = "20px";
    });
    </script>

</body>
</html>
```

10.How to read and write a file using JavaScript?

JavaScript has limited access to the local file system due to security concerns. However, you can interact with files using the File API when working with files that are selected by the user through input elements like `<input type="file">`. Additionally, you can use JavaScript on the server side (e.g., with Node.js) to perform more extensive file operations.

11.What are all the looping structures in JavaScript?

JavaScript provides several looping structures that allow you to repeat a block of code multiple times. Here are the main looping structures in JavaScript:

for (initialization; condition; increment/decrement) {
// code to be executed
}

12.How can you convert the string of any base to an integer in JavaScript?

Keep in mind that `parseInt()` will attempt to convert the provided string to an integer in the specified base. If the string contains non-numeric characters that are not valid for the specified base, `parseInt()` will stop parsing and return the converted part of the string.

```
const string = "42";
const int = parseInt(int);
console.log(int);
```

13.What is the function of the delete operator?

let demo = {
firstName: "Raj",
lastName: "Vishwakarma",
salary: 15000
}

console.log(delete emp.salary);
console.log(demo);

console.log(delete emp.salary);
console.log(demo);

14.What are all the types of Pop up boxes available in JavaScript?

alert

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

  <style>
    p{

    }
  </style>
</head>
<body>
  <p id="one"></p>
  <button  onclick = "a()">Enter ME</button>
  <script>
    function a(){
      alert("404");

    }
  </script>
</body>
</html>
```

confirm

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
<p id="one"></p>
<button onclick="a()">Hit ME</button>
<script>
function a(){
  let name;
  if (confirm("please enter between tow suggetion")) {
    name = "Yes you pressed Ok"
  } else {
    name = "There is not pressed Ok"

  }
  document.getElementById("one").innerHTML = name;
}
</script>
</body>
</html>
```

prompt

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <p id="one"></p>
  <button onclick="a()">Hit Me</button>

  <script>
    function a(){
      let demo;
      let letterbox = prompt("please enter your name");
      if (letterbox == " " || letterbox == null) {
        demo = "please enter your name";
      } else {
        demo = "Hii "+letterbox+" how are you";
            }
            document.getElementById("one").innerHTML = demo;
    }
  </script>
```

```
</body>
</html>
```

15.What is the use of Void (0)?

The usage of `void(0)` in JavaScript is primarily related to preventing the navigation of a page when clicking on a hyperlink or a button. This technique is used to create "javascript:void(0)" links or buttons that perform JavaScript actions without causing the browser to navigate to a new page or trigger a page refresh.

```
<nav>
     <div class="container main-nav flex">
      <a class="company-logo" href="javascript:void(0)"> <img  src="asset/logo icon.svg" alt=""></a>

      <div class="nav-links" id="nav-links">
       <ul class="flex">
        <li><a class="hover-link" href="javascript:void(0)">Product</a></li>
        <li><a class="hover-link" href="javascript:void(0)">Customers</a></li>
        <li><a class="hover-link" href="javascript:void(0)">Pricing</a></li>
        <li><a class="hover-link" href="javascript:void(0)">Resources</a></li>
        <li><a class="hover-link secondary-btn" href="javascript:void(0)">Sign in</a></li>
        <li><a class="hover-link primary-btn" href="javascript:void(0)">Sign up</a></li>
       </ul>
      </div>

      <a href="javascript:void(0)" class="nav-toggle hover-link" id="toggle-icon"><i class="fa-solid fa-bars"></i></a>

     </div>

    </nav>
```

16.How can a page be forced to load another page in JavaScript?

// Navigate to a new page
window.location.href = "https://www.amazon.in/";

17.What are the disadvantages of using innerHTML in JavaScript?

While the `innerHTML` property in JavaScript is a convenient way to manipulate the content of HTML elements.

Using `innerHTML` to directly insert or modify HTML content from untrusted sources can lead to security vulnerabilities like Cross-Site Scripting (XSS).

Changing an element's `innerHTML` can potentially remove or overwrite existing event handlers attached to child elements.