1. What is HTML (Hyper Text Markup Language) ?

HTML is the standard language used to create and design web pages & web applications. It provides the structure of a webpage by using a series of elements and tags to denote different types of content and their respective formatting.

2. What are web pages and about its structure ?

A web page is a document accessible on the World Wide Web, typically written in HTML and designed to be viewed in a web browser. Web pages can contain text, images, videos, links, and other multimedia content. They are the fundamental building blocks of websites and can be static (displaying the same content to every visitor) or dynamic (content can change based on user interaction or other factors).

3. Use of HTML ?

HTML (Hyper Text Markup Language) is fundamental for creating and designing web pages and web applications.

4. What is Static and Dynamic web Pages ?

Static web pages display fixed content that doesn't change unless manually updated. Dynamic web pages display content that can change based on user interaction, database updates, or other factors, often using server-side scripting languages like PHP or JavaScript to generate content dynamically.

5. What is web accessibility ?

Web accessibility ensures websites are usable by people with disabilities, providing equal access to information and functionality.

6. What is the Web Content Accessibility Guidelines (WCAG) ?

The Web Content Accessibility Guidelines (WCAG) are a set of international standards developed by the W3C to make web content more accessible to people with disabilities. They provide guidelines for creating content that is perceivable, operable, understandable, and robust, ensuring inclusivity across diverse user needs and technologies.

7. Describe the HTML API ?

HTML APIs (Application Programming Interfaces) are built into the browser and provide functionalities for web applications. Those are:-

i. Geo location API:-

The Geolocation API allows web applications to access the geographical location of the user's device. It is commonly used in mapping applications, location-based services, and for providing personalized content based on location.

Example Usage:-

```
if (navigator.geolocation) {
     navigator.geolocation.getCurrentPosition(function(position) {
          console.log("Latitude: " + position.coords.latitude);
          console.log("Longitude: " + position.coords.longitude);
         });
           } else {
           console.log("Geolocation is not supported by this browser.");
        }
```

Key Features: -

a. getCurrentPosition(): Retrieves the device's current position.

b. watchPosition(): Monitors the device's position and updates when it changes.

c. clearWatch(): Stops monitoring the position.

ii. Web Workers API

The Web Workers API allows web applications to run scripts in background threads, independent of the main execution thread. This helps improve performance and responsiveness by offloading intensive tasks from the main thread.

Example Usage:-

```
// main.js
  if (window.Worker) {
        const worker = new Worker('worker.js');
        worker.postMessage('Hello, worker');
        worker.onmessage = function(event) {
                        console.log('Received from worker:', event.data);
                         };}
  // worker.js
  onmessage = function(event) {
     console.log('Received from main script:', event.data);
     postMessage('Hello, main script');
  };
```

Key Features:

a. Worker(): Creates a new web worker.
b. postMessage(): Sends a message to the worker.
c. onmessage: Event handler for receiving messages from the worker.

iii. Server-Sent Events (SSE) API

The Server-Sent Events (SSE) API enables web applications to receive automatic updates from a server via a persistent HTTP connection.It is useful for real-time applications like news feeds, live sports scores, and stock price updates.

Example Usage:

```
// client-side JavaScript
if (typeof(EventSource) !== "undefined") {
       const source = new EventSource('server-endpoint');
       source.onmessage = function(event) {
           console.log('New message:', event.data);
       };
   } else {
     console.log("SSE not supported by this browser.");
  }
// server-side (Node.js example using Express)
  const express = require('express');
  const app = express();
  app.get('/server-endpoint', (req, res) => {
         res.setHeader('Content-Type', 'text/event-stream');
         res.setHeader('Cache-Control', 'no-cache');
         res.setHeader('Connection', 'keep-alive');
  setInterval(() => {
    res.write(`data: ${new Date().toISOString()}\n\n`);
  }, 1000); });     app.listen(3000);
```

Key Features:

a. EventSource(): Creates a new SSE connection.

b. onmessage: Event handler for receiving messages from the server.

c. Server-side streaming: The server sends updates as a continuous stream.

8. Describe the Attribute in HTML ?

- ❖ Attributes in HTML provide additional information about elements and are used to modify their behavior or appearance.
- ❖ They are specified within the opening tag of an element and consist of a name and a value. Here's an overview of attributes in HTML:

Syntax:

- ➢ Attributes are written as name-value pairs within the start tag of an element.
- ➢ The syntax is name="value", where name is the attribute name and value is its corresponding value.
- ➢ Multiple attributes can be included in a single element, separated by spaces.

A. Common Attributes:

1. **id**: Provides a unique identifier for an element within the document.
2. **class**: Assigns one or more class names to an element, allowing styling with CSS.
3. **src**: Specifies the URL of an external resource, such as an image or script.
4. **href**: Specifies the URL of a hyperlink.
5. **alt**: Provides alternative text for images, useful for accessibility.
6. **title**: Provides additional information about an element, often displayed as a tooltip.
7. **style**: Applies inline CSS styles to an element.

8. **disabled**: Disables an input element or button.

9. **checked**: Pre-selects a checkbox or radio button.

10. **aria-*:** Attributes used for accessibility (e.g., aria-label, aria-labelledby).

B. Custom Attributes:

- ➢ Custom attributes can be created for storing additional data or for JavaScript purposes.
- ➢ While not inherently invalid, it's best practice to prefix custom attributes with data- to ensure compliance with HTML standards.

Example: <div data-custom="value">

C. Boolean Attributes:

- ➢ Some attributes do not require a value and are considered true if present, and false if absent.
- ➢ Examples include checked, disabled, readonly, and required.

9. Graphics in HTML ?

Graphics in HTML can be added using various elements and techniques.Here are some common methods for including graphics:

1. < img >:-

The <img> tag is used to embed images in HTML documents.

eg:-

< img src="image.jpg" alt="Description of the image">

2. < svg >

SVG (Scalable Vector Graphics) allows for the creation of vector-based graphics directly within HTML.

eg:-

```
 <svg width="100" height="100">
      <circle cx="50" cy="50" r="40" fill="red" />
</svg>
```

3. <canvas>

The <canvas> element provides a drawing surface for graphics via JavaScript.

eg:-

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

4. <iframe>

The <iframe> element can embed content from another website or source, including graphics.

eg:-

```
<iframe src="https://example.com/image.html" width="300"
    height="200"></iframe>
```

5. CSS Background Images:

Background images can be added to HTML elements using CSS

eg:-

```
*header {
      background-image: url('background.jpg');
      background-size: cover;    }
```

10. what is tags in HTML ?

In HTML, tags are the fundamental building blocks used to create the structure and content of web pages.

A. Head Tags:-

The <head> tag in HTML is used to define the head section of an HTML document. It contains meta-information about the document.

i. <title>

Defines the title of the HTML document, which appears in the browser's title bar or tab.

eg:-

<title>This is the Title</title>

ii. <meta>

Provides metadata about the HTML document, such as the character set, author, and viewport settings.

eg:-

<meta charset="UTF-8">

<meta name="description" content="Description of the webpage">

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

iii. <link>

Links to external resources, such as stylesheets, favicon, or other documents.

eg:-

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

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

iv. <script>

Defines client-side JavaScript code or links to external JavaScript files.

eg:-

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

v. Other Meta Tags:

Additional meta tags can be used for search engine optimization (SEO),specifying keywords, author information, etc.

eg:-

```
<meta name="keywords" content="HTML, CSS, JavaScript">

<meta name="author" content="John Doe">
```

vi. <style>

Contains internal CSS styles that apply specifically to the current document.

eg:-

```
<style>

  body {

    font-family: Arial, sans-serif;

  }

</style>
```

vii. <base>

Specifies a base URL to be used for all relative URLs within the document.

eg:-

```
<base href="https://www.example.com">
```

viii. <noscript>

Provides content to be displayed if the browser does not support JavaScript or if JavaScript is disabled.

eg:-

```
<noscript>Your browser does not support JavaScript!</noscript>
```

B. Containt / Container Tags:-

i. <div>

The <div> tag is a generic container used to group related elements together for styling purposes or to apply common behaviors.

eg:-

```
<div>
   <p>This is some text inside a div.</p>
   <img src="image.jpg" alt="Image">
</div>
```

ii. <span>:

The <span> tag is an inline container typically used to apply styles or behaviors to a specific portion of text.

eg:-

```
<p>This is <span style="color: red;">red</span> text.</p>
```

iii. <section>

The <section> tag defines a section of content within a document, often representing a thematic grouping of content.

eg:-

```
<section>
    <h2>Section Title</h2>
    <p>This is the content of the section.</p>
</section>
```

iv. <article>

The <article> tag represents a self-contained piece of content that could be distributed and reused independently.

eg:-

```
<article>
            <h2>Article Title</h2>
            <p>This is the content of the article.</p>
</article>
```

v. <header>

The <header> tag typically contains introductory content or navigational elements at the beginning of a document or section.

eg:-

```
<header>
  <h1>Page Title</h1>
  <nav>
    <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
    </ul>
  </nav>
</header>
```

vi. <footer>

The <footer> tag contains content typically found at the end of a document or section, such as copyright information or contact details.

eg:-

```
<footer>
  <p>&copy; 2024 My Website</p>
</footer>
```

vii. <nav>

The <nav> tag defines a section of navigation links, usually placed at the beginning or end of a document.

eg:-

```
<nav>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
    </ul>  </nav>
```

viii. <h1 ~ h6>

The <h1> tag in HTML is used to define the most important heading in a document. It represents the highest

level of heading and is typically used to indicate the main topic or title of the page.

eg:-

<h1>This is the Main Heading</h1>

ix. <p>

The <p> tag in HTML is used to define a paragraph of text. It is one of the most commonly used tags for organizing and formatting textual content on a webpage.

eg:-

<p>This is a paragraph of text.</p>

x. <strong>

The <strong> tag in HTML is used to indicate strong importance or emphasis on text. It typically renders the enclosed

text in a bold font style by default, though its primary purpose is to convey importance rather than dictate specific styling.

eg:-

<p>This is <strong>important</strong> text.</p>

xi. <em>

The <em> tag in HTML is used to denote emphasis on text, typically rendering the enclosed text in italics by default. It indicates a level of emphasis that is different from the surrounding text, often used to highlight a word or phrase for added significance.

eg:-

```
<p>This is <em>emphasized</em> text.</p>
```

C. List Tags:-

i. <ol>:-

Ordered List <ol>Defines a list where items are ordered with numbers or letters by default.

eg:-

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

ii. <ul>:-

Unordered List <ul> Defines a list where items are unordered with bullet points by default.

eg:-

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

iii. <dl>:-

Definition List <dl> Defines a list of terms and their corresponding descriptions.

Each term is marked with a <dt> (definition term) tag, and each description is marked with a <dd> (definition description) tag.

eg:-

```
<dl>
   <dt>Term 1</dt>
   <dd>Description 1</dd>
   <dt>Term 2</dt>
   <dd>Description 2</dd>
</dl>
```

iv. <li>:-

The <li> tag in HTML is used to define an item in a list. It is a child element of either <ul> (unordered list) or <ol> (ordered list) tags, which represent unordered and ordered lists, respectively.

eg:-

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

D. Semantic Elements in HTML:-

i. <article>:-

Represents a self-contained piece of content that can be distributed and reused independently. Commonly used for blog posts, news articles, comments, etc.

ii. <aside> :-

Defines content tangentially related to the surrounding content, often represented as a sidebar, pull quote, or similar element.

iii. <details>

Represents additional details or information that can be toggle open or closed. Typically used to create collapsible sections of content.

iv. <figcaption>

Represents a caption or legend for a figure or image. It provides context or explanation for the associated content.

v. <figure>

Represents any content that is referenced as a single unit, often including images, diagrams, illustrations, etc., along with their captions (<figcaption>).

vi. <footer>

Represents the footer of a document or section, typically containing metadata, copyright information, links to related documents, etc.

vii. <header>

Represents the header of a document or section, containing introductory content, headings, navigation links, etc.

viii. <main>

Represents the main content of the document, excluding headers, footers, sidebars, and other content that is peripheral to the main content.

ix. <mark>

Represents text highlighted for reference or emphasis, typically highlighted with a yellow background.

x. <nav>

Represents a section of navigation links, typically used for site navigation menus, table of contents, or other navigation-related content.

xi. <section>

Represents a thematic grouping of content within a document, typically with a heading (<h1>-<h6>) to denote the section's title.

xii. <summary>

Represents a summary, caption, or legend for a <details> element, providing a brief overview or explanation of the content.

xiii. <time>:-

Represents a specific point in time or a duration. It can be used to mark up dates, times, or time ranges.

E. Empty tag:-

i. <img>:-

The <img> tag is used to embed images in HTML documents. It is an empty element and does not require a closing tag.

eg:-

```
<img src="image.jpg" alt="Description">
```

ii. <br>:-

The <br> tag inserts a line break, forcing the content after the break to appear on a new line. Like <img>, it is an empty element and does not have a closing tag.

eg:-

```
Line 1<br>Line 2
```

iii. <hr>:-

The <hr> tag represents a thematic break or horizontal rule, dividing content into sections. It is also an empty element.

eg:-

```
<p>Content above</p>
    <hr>
<p>Content below</p>
```

F. Multimedia Tags:-

i. <audio>:-

This tag is used to embed audio content in an HTML document. It can contain one or more <source> tags to specify alternative audio file formats

eg:-

```
<audio controls>

  <source src="audio.mp3" type="audio/mpeg">

  Your browser does not support the audio element.

</audio>
```

ii.<video>:-

The <video> tag is used to embed video content in an HTML document. Like <audio>, it supports multiple <source> tags for different video formats.

eg:-

```
<video controls>

     <source src="video.mp4" type="video/mp4">

     Your browser does not support the video tag.

</video>
```

iii. <source>:-

The <source> tag specifies media resources for <audio>, <video>, or <picture> elements. It provides alternative media files in different formats or resolutions.

eg:-

```
<video controls>
    <source src="video.mp4" type="video/mp4">
    <source src="video.webm" type="video/webm">
    Your browser does not support the video tag.
</video>
```

iv. <embed>:-

The <embed> tag embeds external content, such as multimedia content or plugins, into an HTML document. It is commonly used for embedding Flash content.

eg:-

```
<embed src="movie.swf" width="400" height="300">
```

v. <track>:-

The <track> tag is used to specify text tracks for <audio> or <video> elements, such as subtitles, captions, or descriptions.

eg:-

```
<video controls>
    <source src="video.mp4" type="video/mp4">
    <track src="subtitles_en.vtt" kind="subtitles" srclang="en"
            label="English">
   <track src="subtitles_fr.vtt" kind="subtitles" srclang="fr"
             label="French">
    Your browser does not support the video tag.
</video>
```

G. HTML <a> tag attributes

- The <a> tag defines a hyperlink, which is used to link from one page to another.
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
-

i. href:-

Specifies the URL (Uniform Resource Locator) of the linked document or resource.

eg:-

<a href="https://www.example.com">Visit Example</a>

ii. target:-

Specifies where to open the linked document. Possible values include:-

1. **_self:** Opens the link in the same window/tab (default).
2. **_blank:** Opens the link in a new window/tab.
3. **_parent:** Opens the link in the parent frame or window.
4. **_top:** Opens the link in the full body of the window.

eg:-

<a href="https://www.example.com" target="_blank">Visit Example</a>

iii. download:-

Specifies that the target will be downloaded when a user clicks on the hyperlink. The value of the attribute specifies the filename.

eg:-

```
<a href="image.jpg" download>Download Image</a>
```

iv. rel:-

Specifies the relationship between the current document and the linked document. Common values include nofollow, noopener, noreferrer, etc.

eg:-

```
<a href="https://www.example.com" rel="nofollow">Visit
        Example</a>
```

v. title:-

Specifies extra information about the linked document or resource, typically displayed as a tooltip when the mouse hovers over the link.

eg:-

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

vi. aria-*:-

Attributes for accessibility, such as aria-label, aria-labelledby, aria-describedby, etc., to provide additional information to assistive technologies.

eg:-

```
<a href="https://www.example.com" aria-label="Visit Example
        Website">Visit Example</a>
```