

PW SKILL ASSIGNMENT

RAJKAMAL YADAV

Question1: List out the feature of HTML 5 ?

Answer:

HTML stands for Hypertext Markup Language, the standard language for creating web pages and web applications. HTML5, the 5th version of HTML, introduces several new features that enhance the creation of both static and dynamic websites. Let's explore the top 10 new features of HTML5 that make it a significant improvement over previous versions:

1. New Semantic Elements

- **<header>**: Defines the header section of a document or a section.
- **<footer>**: Defines the footer section of a document or a section.
- **<article>**: Represents a self-contained piece of content.
- **<section>**: Defines a section in a document.
- **<nav>**: Represents a navigation section.
- **<aside>**: Represents content aside from the main content.
- **<main>**: Represents the main content of the document.

2. Multimedia Support

- **<audio>**: Allows embedding audio files with native controls.
- **<video>**: Allows embedding video files with native controls.
- **<source>**: Specifies multiple media resources for video or audio elements.
- **<track>**: Provides text tracks for media elements, like subtitles.

3. Graphics and Interactive Content

- **<canvas>**: Used for drawing graphics on the fly via scripting (usually JavaScript).
- **<svg>**: Supports Scalable Vector Graphics for creating vector-based images directly in the browser.
- **<math>**: Enables mathematical notations.

4. New Form Controls

- **<datalist>**: Provides a list of predefined options for an **<input>** element.

- **<keygen>**: Facilitates key pair generation (now deprecated in favor of other methods).
- **<output>**: Represents the result of a calculation or user action.
- **New Input Types**: New types like email, url, number, range, date, time, color, search, etc.
- **Placeholder Attribute**: Allows for placeholder text in input fields.

5. Improved Accessibility and ARIA Roles

- **role attribute**: Used in conjunction with ARIA (Accessible Rich Internet Applications) to improve accessibility for users with disabilities.
- **<figure> and <figcaption>**: Used for associating images and their captions.

6. Local Storage and Offline Capabilities

- **localStorage and sessionStorage**: Allow data to be stored in the user's browser with no expiration date and session-based expiration, respectively.
- **IndexedDB**: A more complex form of local storage for storing large amounts of structured data.
- **Application Cache**: Enables web applications to be cached and run offline (now largely deprecated in favor of Service Workers).

7. Enhanced Scripting Capabilities

- **JavaScript APIs**: New APIs like Geolocation API, Drag and Drop API, Web Workers for background processing, and WebSockets for real-time communication.
- **<script> Improvements**: Ability to place scripts at the end of the document and use async or defer attributes to improve loading performance.

8. New Attributes and Global Attributes

- **data-* attributes**: Allow embedding custom data attributes within HTML elements.
- **contenteditable**: Allows an element's content to be editable by the user.
- **draggable**: Specifies whether an element is draggable.

9. Enhanced Browser Support

- **Cross-document messaging**: Allows sending messages between different origins.
- **CORS (Cross-Origin Resource Sharing)**: Improved handling for cross-origin requests.

10. Improved Error Handling and Parsing

- HTML5 parsers handle invalid markup more gracefully and consistently across browsers.

11. Mobile-Friendly and Responsive Design Features

- **Viewport Meta Tag**: Helps control layout on mobile devices.
- **Responsive images**: Using srcset and picture elements for serving appropriate image sizes.

12. Web Components

- **Custom Elements**: Allows developers to define their own HTML elements.

- **Shadow DOM:** Encapsulates the internal structure of web components, preventing styling and scripting interference.
- **Templates:** Allows for defining reusable HTML templates.

Question2: What are the html entities. list out 5 commonly use entities?

Answer:

HTML entities are special codes that represent characters that are either reserved in HTML or have no corresponding character in the keyboard. These entities are used to display characters that would otherwise be interpreted as HTML code, or to include characters that are not easily typed.

Each HTML entity starts with an ampersand (&) and ends with a semicolon (;)

5 Commonly Used HTML Entities:

1. **<;** Represents the "less than" sign (<).
 - Example: < is written as < in HTML.
2. **>;** Represents the "greater than" sign (>).
 - Example: > is written as > in HTML.
3. **&;** Represents the ampersand (&).
 - Example: & is written as & in HTML.
4. **";** Represents a double quotation mark (").
 - Example: " is written as " in HTML.
5. **'** or **';** Represents a single quotation mark or apostrophe (').
 - Example: ' is written as ' or ' in HTML.

Question3: define accessibility in the context of web development. discuss why it's essential to create accessible website and how it benefits different user group?

Answer:

What is Accessibility in Web Development?

Accessibility in web development refers to the practice of designing and building websites and web applications in a way that ensures they can be used by as many people as possible, including those with disabilities. This involves making sure that all users, regardless of their physical or cognitive abilities, can perceive, understand, navigate, and interact with the web content. Accessibility is often achieved through adherence to standards such as the Web Content Accessibility Guidelines (WCAG).

Why Accessibility is Essential

1. Legal Compliance:

- Many countries have laws and regulations that require websites to be accessible to people with disabilities. Non-compliance can result in legal action, fines, and damage to a company's reputation.

2. Inclusive User Experience:

- Accessibility ensures that everyone, including people with disabilities, can access information and services on the web. This includes people with visual, auditory, motor, and cognitive impairments.

3. Wider Audience Reach:

- By making a website accessible, businesses can reach a broader audience, including those who might otherwise be excluded. This can lead to increased engagement, customer satisfaction, and conversion rates.

4. Improved Usability for All:

- Accessibility often leads to better usability for all users, not just those with disabilities. Features like larger buttons, clearer navigation, and more readable text benefit everyone, including older adults or users in environments with poor lighting.

5. SEO Benefits:

- Accessible websites are often more search engine friendly because they follow good semantic practices, such as using proper heading structures, alt text for images, and descriptive link text. This can improve a site's search engine rankings.

How Accessibility Benefits Different User Groups

1. People with Visual Impairments:

- **Screen Readers:** Ensuring that websites are compatible with screen readers allows visually impaired users to navigate the site using audio feedback.
- **Text Alternatives:** Alt text for images and labels for form elements help those who rely on screen readers or have low vision.

2. People with Hearing Impairments:

- **Transcripts and Captions:** Providing text alternatives for audio and video content ensures that deaf or hard-of-hearing users can access the same information as others.
- **Visual Cues:** Using visual alerts in place of sound notifications ensures that important messages are conveyed to all users.

3. People with Motor Disabilities:

- **Keyboard Navigation:** Ensuring that all functionality is accessible via the keyboard helps users who cannot use a mouse due to motor impairments.

- **Accessible Forms:** Designing forms that are easy to navigate and complete with assistive devices, like sip-and-puff systems or head pointers, benefits users with limited motor control.

4. People with Cognitive Disabilities:

- **Clear Content Structure:** Simple language, clear instructions, and logical content structure help users with cognitive disabilities understand and navigate the site.
- **Consistent Navigation:** Providing a consistent layout and navigation throughout the site reduces confusion and cognitive load.

5. Older Adults:

- **Readable Fonts:** Larger, readable fonts and high-contrast color schemes help older adults who may have declining vision.
- **Simplified Interaction:** Simplified and intuitive interaction models benefit older adults who may be less familiar with technology.

Question4: List any 3 ways which help us in improving the accessibility of HTML?

Answer:

Improving the accessibility of HTML involves several best practices that ensure a website is usable by people with various disabilities. Here are three key ways to enhance accessibility:

1. Use Semantic HTML Elements

- **Why it helps:** Semantic HTML elements, such as <header>, <nav>, <main>, <article>, and <footer>, provide meaning to the content structure and help assistive technologies like screen readers understand the content better.
- **Example:** Instead of using <div> for everything, use <nav> for navigation menus and <article> for independent content sections. This makes it easier for users to navigate the page.

```
<header>
```

```
<h1>Website Title</h1>
```

```
<nav>
```

```
<ul>
```

```
<li><a href="#home">Home</a></li>
```

```
<li><a href="#about">About</a></li>
```

```
<li><a href="#contact">Contact</a></li>
```

```
</ul>
```

```
</nav>
```

</header>

2. Provide Text Alternatives for Non-Text Content

- **Why it helps:** Non-text content, like images, videos, and icons, may not be accessible to users who are blind or have low vision. By providing text alternatives (e.g., alt text for images), you ensure that all users can understand the content.
- **Example:** Use the alt attribute in tags to describe the image.

```

```

- **For complex images:** If an image is complex, such as a chart or graph, provide a more detailed description in a nearby text or use the <figure> and <figcaption> elements.

3. Ensure Keyboard Accessibility

- **Why it helps:** Some users cannot use a mouse and rely on keyboard navigation to interact with web content. Ensuring that all interactive elements, such as links, buttons, and form fields, are accessible via keyboard is essential.
- **Example:**
 - Ensure that all interactive elements are focusable and can be accessed using the Tab key.
 - Use the :focus CSS pseudo-class to indicate when an element is focused, helping users who navigate with the keyboard.

```
<a href="contact.html" class="button">Contact Us</a>
```

```
<style>
```

```
.button:focus {
```

```
outline: 2px solid #000; /* Provides a visible outline when focused */
```

```
}
```

```
</style>
```

- **Additional tip:** Avoid using elements that are not natively focusable (like <div> or) for interactive purposes. If you do, ensure you add appropriate ARIA roles and make them keyboard accessible.

Question5: Create a webpage that highlight the feature of HTML 5. use appropriate semantic tags to structure the content and showcase at list 3 key feature of HTML5 with explanation?

Answer:

HTML5 Semantics refers to the use of specific tags like <header>, <footer>, <nav>, <article>, <section>, etc., to provide clearer structure and meaning to web content. This improves accessibility, and SEO, and facilitates better understanding by both humans and machines.

Why Do I Need to Use Semantic HTML Tags?

Using semantic HTML tags enhances website accessibility, improves search engine optimization (SEO), and provides clearer structure and meaning to content. This fosters better understanding by both users and machines, leading to improved usability and user experience.

Exapmle:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Key Features of HTML5</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      line-height: 1.6;

      margin: 0;

      padding: 0;

      background: #f4f4f4;

    }

    header, footer {

      background: #333;

      color: #fff;

      padding: 10px 20px;

      text-align: center;

    }

    section {

      padding: 20px;

      margin: 10px;
```

```
background: #fff;
border-radius: 5px;
}
h2 {
color: #333;
}
article {
margin-bottom: 20px;
}
</style>
</head>
<body>

<header>
  <h1>Features of HTML5</h1>
</header>

<main>
  <section>
    <article>
      <h2>1. Semantic HTML Elements</h2>

      <p>HTML5 introduced several new semantic elements that provide meaning to the structure of
web documents. Examples include:</p>

      <ul>
        <li><strong>&lt;header&gt;</strong>: Defines the header section of a document or
section.</li>
        <li><strong>&lt;footer&gt;</strong>: Defines the footer section of a document or
section.</li>
        <li><strong>&lt;article&gt;</strong>: Represents a self-contained piece of content that could
be distributed independently.</li>
      </ul>

      <p>These elements improve the organization of content and make it easier for search engines
and assistive technologies to understand the structure of the page.</p>
```


</article>

<article>

<h2>2. Multimedia Support</h2>

<p>HTML5 provides native support for embedding multimedia elements like audio and video without needing third-party plugins. Examples include:</p>

<audio>: Allows for the embedding of audio files with native controls for play, pause, and volume.

<video>: Allows for the embedding of video files with native controls for playback, such as play, pause, and full-screen.

<p>This built-in support ensures that multimedia content is more accessible and easier to use across different devices and browsers.</p>

</article>

<article>

<h2>3. Local Storage</h2>

<p>HTML5 introduced new methods for storing data on the client side, making web applications more powerful and capable. Examples include:</p>

localStorage: Provides a way to store data persistently across sessions without expiration.

sessionStorage: Provides a way to store data for the duration of the page session (until the page or browser is closed).

<p>These storage options allow for saving user preferences, form data, and other information, improving the user experience by reducing the need to re-enter data.</p>

</article>

</section>

</main>

<footer>

<p>© 2024 HTML5 Features. All rights reserved.</p>

</footer>

</body>

</html>

Explanation:

- **Semantic HTML Elements:** Uses <header>, <footer>, and <article> to structure the content meaningfully.
- **Multimedia Support:** Demonstrates the use of <audio> and <video> tags (although the actual multimedia files are not included in this example, they should be added as needed).
- **Local Storage:** Mentions localStorage and sessionStorage as methods for client-side data storage.

Question6: Create a simple webpage which has a table. the table must have 2 columns HTML and HTML5. the table should include a minimum of tree rows describing between HTML and HTML5?

Answer:

Explanation:

- **HTML and HTML5 Columns:** The table has two columns comparing HTML and HTML5 features.
- **Three Rows:** Each row describes a difference or improvement from HTML to HTML5.
- **Styling:** Basic CSS is used to style the table, including borders, padding, and alternating row colors for readability.

code

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>HTML vs HTML5 Comparison</title>
```

```
<style>
```

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 20px;  
    background: #f4f4f4;  
}  
  
table {  
    width: 100%;  
    border-collapse: collapse;  
    background: #fff;  
    margin-top: 20px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
th, td {  
    padding: 10px;  
    text-align: left;  
    border-bottom: 1px solid #ddd;  
}  
  
th {  
    background: #333;  
    color: #fff;  
}  
  
tr:nth-child(even) {  
    background: #f9f9f9;  
}  
  
</style>  
</head>  
<body>
```

<h1>HTML vs HTML5 Comparison</h1>

<table>

<thead>

<tr>

<th>HTML</th>

<th>HTML5</th>

</tr>

</thead>

<tbody>

<tr>

<td>HTML does not have native support for multimedia elements.</td>

<td>HTML5 introduces native support for multimedia with the
<code><audio></code> and <code><video></code> elements.</td>

</tr>

<tr>

<td>HTML relies on third-party plugins like Flash for interactive content.</td>

<td>HTML5 provides native support for interactive content through elements like
<code><canvas></code> for drawing graphics and animations.</td>

</tr>

<tr>

<td>HTML lacks semantic elements for better content structuring.</td>

<td>HTML5 introduces semantic elements like <code><header></code>,
<code><footer></code>, <code><article></code>, and
<code><section></code> for better content organization and readability.</td>

</tr>

</tbody>

</table>

</body>

</html>

Out Put

