

## ANSWER 1-

HTML5 introduced several new features and improvements over its predecessor, HTML4. Some of the key new features introduced in HTML5 include:

**Semantic Elements:** HTML5 introduced new semantic elements like `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`, etc. These elements provide clearer structure and meaning to the content, making it easier for search engines and assistive technologies to understand the page.

**Audio and Video:** HTML5 introduced `<audio>` and `<video>` elements, which allow native embedding and playback of audio and video content without requiring third-party plugins like Flash.

**Canvas:** The `<canvas>` element introduced in HTML5 provides a drawing API that allows dynamic rendering of 2D and 3D graphics, animations, charts, and interactive visualizations directly within the browser.

**New Input Types:** HTML5 introduced new input types like date, time, email, number, range, color, search, etc. These input types provide improved user experience and better input validation for specific data formats.

**Form Enhancements:** HTML5 introduced new attributes and features to enhance form handling, such as placeholder attribute, required attribute, autofocus attribute, and more.

**Local Storage:** HTML5 introduced the `localStorage` and `sessionStorage` APIs, which allow web applications to store data locally on the user's device, providing a persistent and efficient way to store user preferences and application state.

**Geolocation:** HTML5 introduced the Geolocation API, which allows web applications to access the user's geographical location, enabling location-based services and personalized experiences.

**Drag and Drop:** HTML5 introduced native support for drag and drop operations, allowing users to drag elements and drop them onto designated areas or targets within a web page.

**Responsive Images:** HTML5 introduced the `srcset` and `sizes` attributes, which enable developers to provide different image sources and sizes based on the device's screen resolution, helping to optimize image loading and improve the performance of responsive web design.

**Web Storage:** HTML5 introduced the Web Storage API, which provides a way to store larger amounts of structured data on the client-side, persistently across sessions, improving the performance and scalability of web applications.

These are just a few examples of the many new features introduced in HTML5. HTML5 brought significant advancements to web development, empowering developers to create more powerful and interactive web applications with improved semantics, multimedia support, offline capabilities, and more.

## **ANSWER 2-**

Semantic tags in HTML are elements that carry meaning and convey the structure of the content they contain. They provide contextual information to both the browser and developers, making it easier to understand and interpret the purpose of different sections of a web page. Semantic tags improve accessibility, search engine optimization (SEO), and overall code readability. Here are a couple of examples of semantic tags introduced in HTML5: `<header>`: The `<header>` tag represents the introductory or navigational section of a document or

a specific section within a document. It typically contains a logo, site title, navigation menu, or other related content. Example usage:

```
<header>

  <h1>My Website</h1>

  <nav>

    <ul>

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

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

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

    </ul>

  </nav>

</header>
```

**<nav>:** The `<nav>` tag represents a section of a document that contains navigation links, such as menus, lists of links, or a table of contents. It helps identify the primary navigation area within a page. Example usage:

```
<nav>

  <ul>

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

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

    <li><a href="#">Services</a></li>

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

  </ul>

</nav>
```

**<main>:** The `<main>` tag represents the main content area of a document, excluding header, footer, navigation, and sidebar content. It should contain unique content related to the document or the central topic of the page. Example usage:

```
<main>
```

```
<h1>Welcome to my blog</h1>
```

```
<article>
```

```
  <h2>Article Title</h2>
```

```
  <p>Article content goes here.</p>
```

```
</article>
```

```
</main>
```

**<article>:** The `<article>` tag represents a self-contained composition within a document. It could be a blog post, a news article, a forum post, or any other independent piece of content. Example usage:

```
<article>
```

```
  <h2>Article Title</h2>
```

```
  <p>Article content goes here.</p>
```

```
  <footer>Author, Date</footer>
```

```
</article>
```

These are just a few examples of semantic tags. By utilizing semantic tags appropriately, you enhance the structure and meaning of your HTML, making it more accessible, SEO-friendly, and maintainable

## ANSWER 3-

Semantic tags in HTML have several uses and benefits. Here are some key uses of semantic tags:

**Improved Accessibility:** Semantic tags help improve the accessibility of web content. Screen readers and assistive technologies can better understand and interpret the structure and meaning of the content, providing a more accessible experience for users with disabilities.

**Search Engine Optimization (SEO):** Search engines rely on semantic structure to understand the content and context of web pages. By

using semantic tags, you can provide clearer signals to search engines about the purpose and relevance of different sections of your content, potentially improving your search engine rankings.

**Code Readability and Maintainability:** Semantic tags make your HTML code more readable and self-explanatory. By using tags like `<header>`, `<nav>`, `<main>`, and `<article>`, it becomes easier for developers to understand the purpose and structure of different sections of the page, making code maintenance and collaboration more efficient.

**Structure and Document Outline:** Semantic tags contribute to the overall structure and outline of the document. They help define the hierarchy of the content, making it easier to navigate and comprehend the document's structure, especially for long-form content or documents with complex layouts.

**Styling and CSS Targeting:** Semantic tags provide meaningful hooks for CSS styling and targeting. By using semantic tags, you can apply specific styles to different sections of the document easily. This allows for more modular and maintainable CSS code.

**Future-Proofing:** Semantic tags provide a forward-compatible approach to web development. They are designed to withstand future changes and developments in HTML standards. By utilizing semantic tags, your code is more likely to remain valid and compatible with future versions of HTML.

**Improved Communication and Collaboration:** Semantic tags promote a common understanding and consistent terminology among developers when discussing web page structures. They facilitate effective communication and collaboration between team members, making it easier to discuss and work on different sections of a web page.

Overall, using semantic tags in HTML brings clarity, structure, and meaning to your web content, benefiting accessibility, search engines, code maintainability, and effective communication among developers.