**HTML5 – Assignment**

**• Question 1: Difference b/w HTML & HTML5?**

**Answer:**

|  |  |  |
| --- | --- | --- |
| Feature | HTML | HTML5 |
| Doctype Declaration | <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 ..."> | <!DOCTYPE html> (simpler and shorter) |
| Semantic Elements | Limited, used <div>, <span>, etc. | New semantic elements like <header>, <footer>, <article>, <section>, <nav>, etc. |
| Multimedia Support | No native support for audio/video; plugins (e.g., Flash) required | Native support for <audio> and <video> elements |
| Form Elements | Basic form elements like text, password, checkbox | New input types like email, url, date, range, number, etc., and additional attributes like placeholder, autofocus |
| Canvas | No support for graphics and drawing | <canvas> element for 2D graphics and drawing |
| SVG Support | Basic support for SVG | Improved support for scalable vector graphics (SVG) |
| APIs (Application Programming Interfaces) | Limited JavaScript APIs | Extensive APIs, including Geolocation, Web Storage, Web Sockets, Offline Web Apps, Drag-and-Drop, etc. |
| Local Storage | No local storage for web apps | Local Storage and Session Storage for storing data on the client side |
| Cross-Origin Resource Sharing (CORS) | Limited support for cross-origin requests | Native CORS support for secure cross-origin data sharing |
| Deprecated Elements | Elements like <font>, <center>, <big>, <strike> | Deprecates or removes old elements like <font>, <center>, etc. |
| Mobile & Responsive Design | No built-in support for mobile or responsive design | Built-in support for responsive design using the viewport meta tag, media queries, etc. |
| Offline Support | Not supported | Built-in support for offline applications using Service Workers |
| JavaScript APIs | Limited or no support for advanced JavaScript APIs | Support for advanced JavaScript APIs such as Web Workers, File API, Notification API, etc. |
| Browser Compatibility | Standard across all browsers, with less support for modern features | Supported by modern browsers, but older browsers may need polyfills |
| Accessibility | Lacked semantic structure, harder to create accessible content | Improved accessibility with semantic elements and ARIA support |

**• Question 2: What are the additional tags used in HTML5?**

**Answer:**

**1. <header>**

* Represents the introductory content or navigational links for a document or section.
* Often used for logos, navigation menus, and introductory text.

Example:

<header>

            <h1>My Website</h1>

            <nav>

              <ul>

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

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

              </ul>

            </nav>

</header>

Output:

**My Website**

* [Home](file:///C:\Users\Nisha\Desktop\task-js\html5.html)
* [About](file:///C:\Users\Nisha\Desktop\task-js\html5.html)

**2. <footer>**

* Defines the footer for a document or section, usually containing copyright information, links, or related documents.

Example:

<footer>

            <p> 2024 My Website. All Rights Reserved. </p>

</footer>

Output:

2024 My Website. All Rights Reserved.

**3. <article>**

* Represents a self-contained, independent piece of content, like a blog post or news article.

Example:

<article>

            <h2>HTML5: A Guide</h2>

            <p>This article explains the key features of HTML5.</p>

</article>

Output:

**HTML5: A Guide**

This article explains the key features of HTML5.

**4. <section>**

* Represents a section of content, typically with a heading. Used to group related content.

Example:

 <section>

            <h2>Introduction</h2>

            <p>This section introduces HTML5 elements.</p>

</section>

Output:

**Introduction**

This section introduces HTML5 elements.

**5. <nav>**

* Defines a navigation block, typically used for links to other sections or pages.

Example:

   <nav>

            <ul>

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

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

            </ul>

          </nav>

Output:

* [Home](file:///C:\Users\Nisha\Desktop\task-js\html5.html)
* [Contact](file:///C:\Users\Nisha\Desktop\task-js\html5.html)

**6. <aside>**

* Represents content that is tangentially related to the content around it, like a sidebar or related links.

Example:

 <aside>

            <h3>Related Articles</h3>

            <ul>

              <li><a href="#">HTML5 Features</a></li>

              <li><a href="#">CSS3 Tips</a></li>

            </ul>

          </aside>

Output:

**Related Articles**

* [HTML5 Features](file:///C:\Users\Nisha\Desktop\task-js\html5.html)
* [CSS3 Tips](file:///C:\Users\Nisha\Desktop\task-js\html5.html)

**7. <figure>**

* Represents content (like an image, chart, or video) that is self-contained and often accompanied by a caption.

Example:

<figure>

            <img src="https://t4.ftcdn.net/jpg/07/29/45/59/360\_F\_729455984\_0BLpankzqOhGPVorxT5Tu7VuLNPvx2ur.jpg" alt="A beautiful landscape">

            <figcaption>A stunning view of the mountains.</figcaption>

          </figure>

Output:

A stunning view of the mountains.

**8. <mark>**

* Highlights text, often used to emphasize important content, such as search results.

Example:

<p>The <mark>HTML5</mark> specification is now official.</p>

Output:

The HTML5 specification is now official.

**9. <video>**

* Embeds video content in the document. It can include controls for play, pause, and volume adjustment.

Example:

 <video width="320" height="240" controls>

            <source src="../task-js/video/WhatsApp Video 2024-11-14 at 11.48.52 AM.mp4" type="video/mp4">

        </video>

Output:

Your browser does not support the video element

**10. <audio>**

* Embeds audio content such as music or sound effects, and it also supports playback controls.

Example:

 <audio controls>

            <source src="../task-js/video/WhatsApp Audio 2024-11-14 at 11.51.18 AM.mpeg" type="audio/mp3">

          </audio>

Output:

Your browser does not support the audio element

**11. <canvas>**

* Allows for dynamic graphics, drawing shapes, images, and animations using JavaScript.

Example:

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

<script>

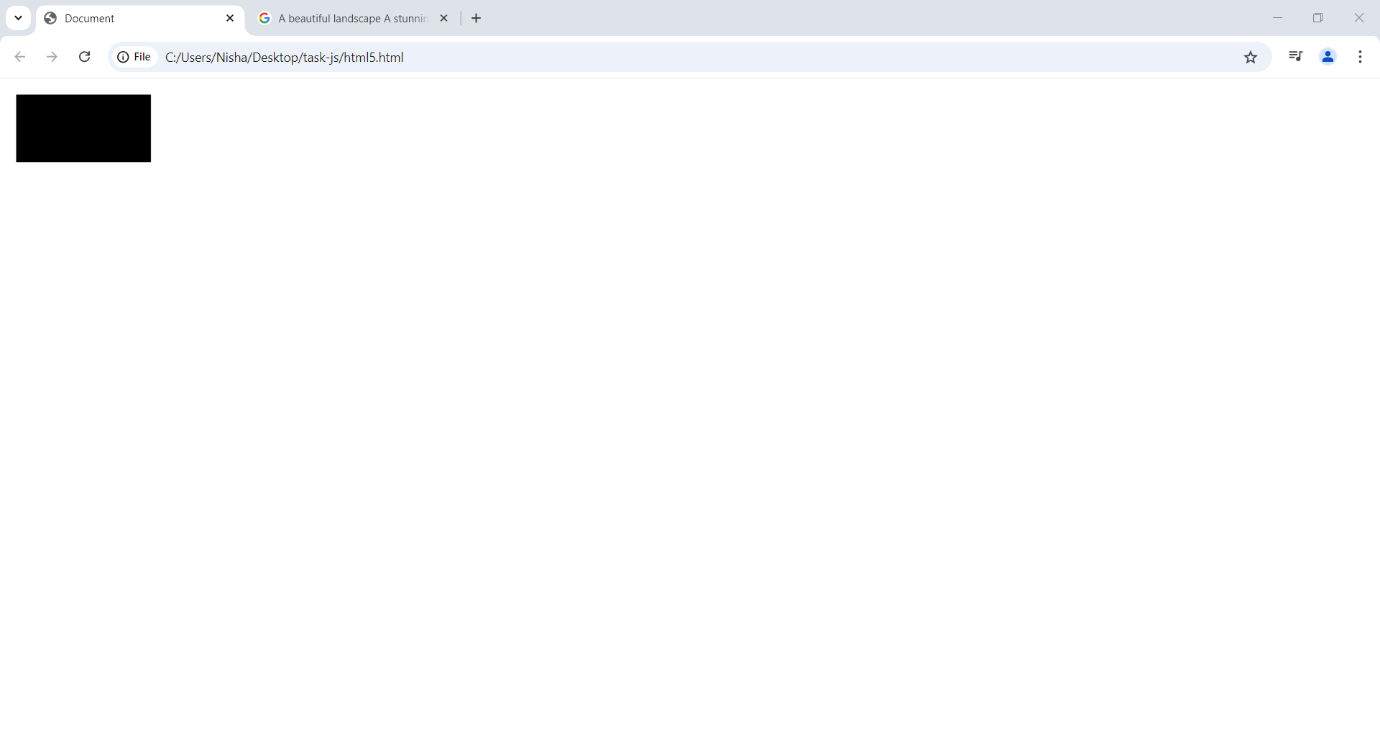
  var canvas = document.getElementById("myCanvas");

  var ctx = canvas.getContext("2d");

  ctx.fillRect(10, 10, 150, 75);

</script>

Output:



**12. <svg**

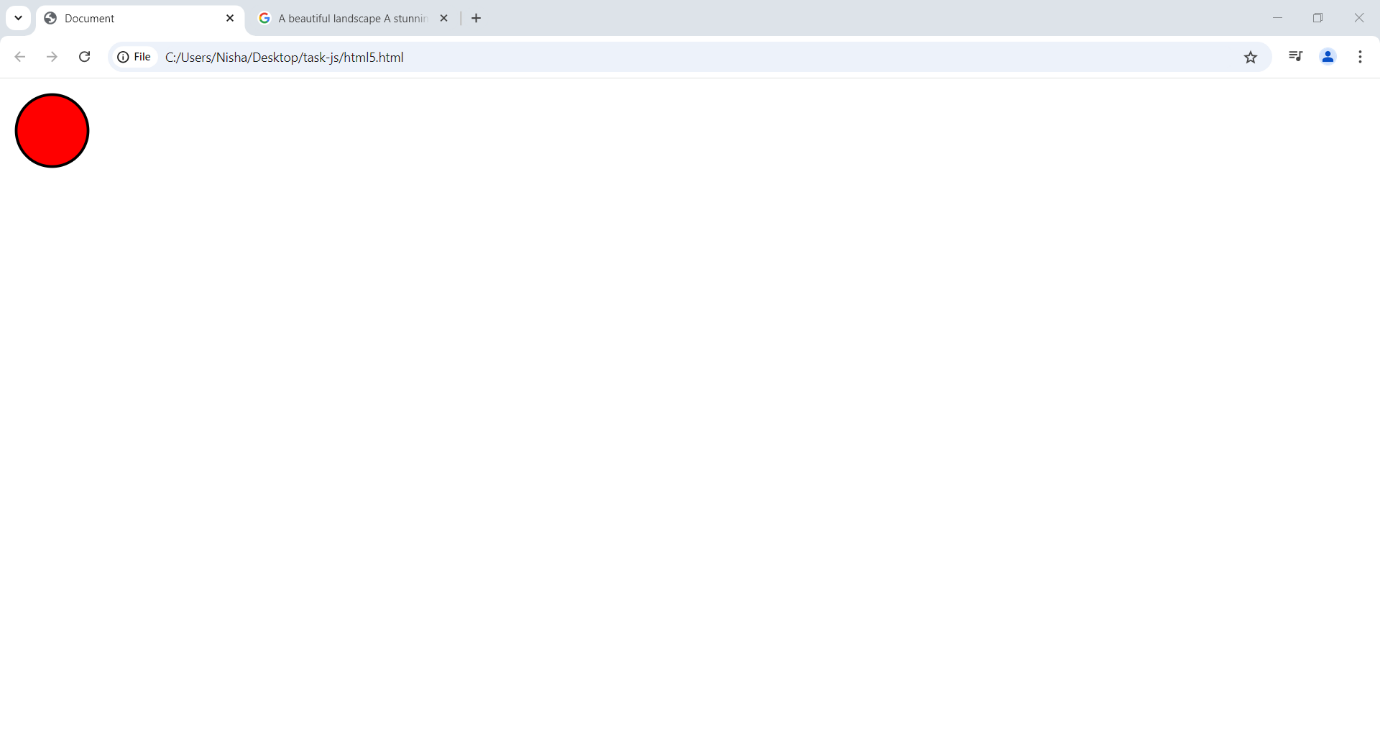
* Defines scalable vector graphics (SVG). It is used for drawing shapes, paths, and other vector-based graphics.

Example:

   <svg width="100" height="100">

            <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />

          </svg>

**Output”** 

**13. <picture>**

* Allows for specifying multiple image sources based on different conditions, like screen size or resolution, providing better image optimization.

Example:

  <picture>

  <source srcset="https://img.freepik.com/premium-photo/beautiful-landscape-house-mountains-house-is-surrounded-by-trees-has-stunning-view-valley\_14117-263951.jpg" media="(max-width: 500px)">

  <source srcset="https://img.freepik.com/premium-photo/beautiful-landscape-house-mountains-house-is-surrounded-by-trees-has-stunning-view-valley\_14117-263951.jpg" media="(min-width: 501px)">

  <img src="https://img.freepik.com/premium-photo/beautiful-landscape-house-mountains-house-is-surrounded-by-trees-has-stunning-view-valley\_14117-263951.jpg" alt="A sample image">

</picture>

Output:

