ASSIGNMENT Module (HTML5) – 3

UDAYRAJPUT

What are the new tags added in HTML5?

ANS :-

**HTML5 introduced several new tags and attributes to enhance the structure and functionality of web documents. Some of the notable new tags in HTML5 include:**

**1. <article>: Represents a self-contained piece of content that could be distributed and reused independently, such as a news article or blog post.**

**2. <section>: Defines a section in a document, grouping together related content. It's often used to create a thematic grouping of content within an `<article>`.**

**3. <nav>: Represents a section of the document intended for navigation links, such as menus or tables of contents.**

**4. <header>: Represents a group of introductory or navigational aids, often containing heading elements (`<h1> - <h6>`), logos, and navigation elements.**

**5. <footer>: Represents a footer for a section or a page, containing metadata, copyright information, and other related content.**

**6. <aside>: Represents content that is tangentially related to the content around it, such as sidebars or pull quotes.**

**7. <figure> and <figcaption>: <figure> is used to encapsulate media content (such as images, videos, and diagrams), and <figcaption> provides a caption or description for the content within <figure>.**

**8. <mark>: Highlights text within a document, typically used to indicate a part of the text that has been marked or highlighted for reference.**

**9. <progress>: Represents the completion progress of a task. It's often used to show the progress of file uploads or form submissions.**

**10. <time>: Represents a specific period in time, allowing for machine-readable date and time information.**

**11. <datalist>: Contains a set of <option> elements that represent the permissible or suggested options available to users in other controls, such as <input> elements with the list attribute.**

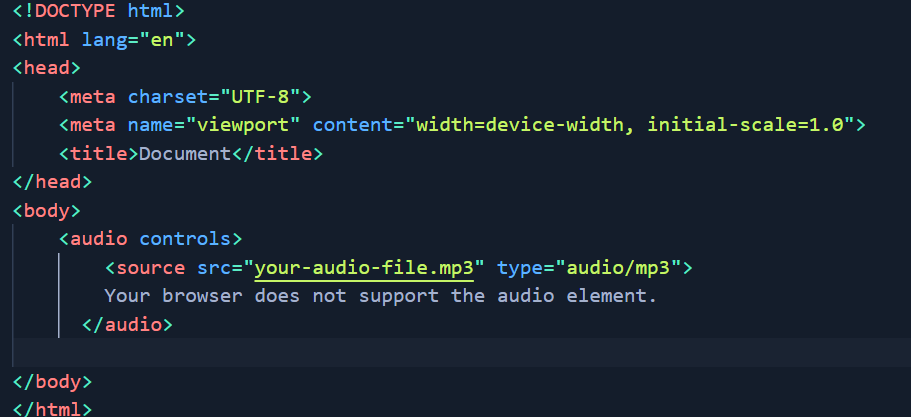
**12. <output>: Represents the result of a calculation or user action. It's often used in conjunction with scripting languages like JavaScript.**

**These are just a few examples of the new tags introduced in HTML5.**

How to embed audio and video in a webpage?

**ANS :- To embed audio and video in a webpage, you can use the <audio> and <video> elements in HTML5. Here's a basic guide on how to do it:**

**### Embedding Audio:**

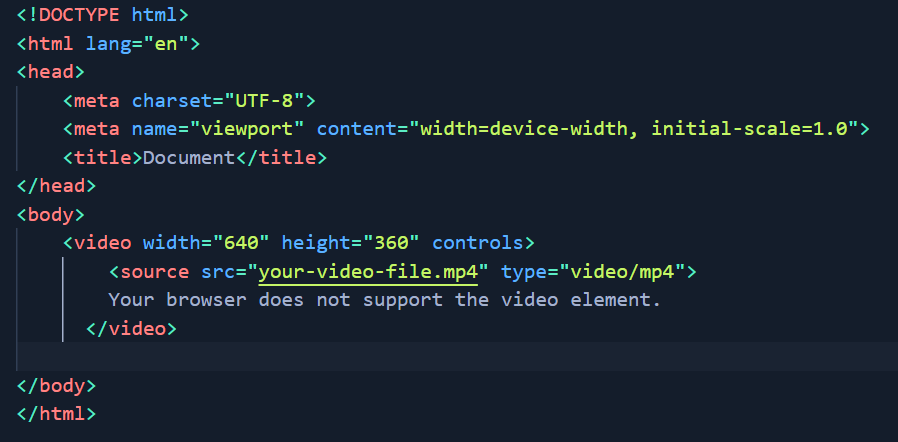
****

**In this example:**

**- The controls attribute adds play, pause, and volume controls to the audio player.**

**- The <source> element specifies the source file and its type.**

**### Embedding Video:**

****

**In this example:**

**- The controls attribute adds play, pause, and volume controls to the video player.**

**- The <source> element specifies the source file and its type.**

**- The width and height attributes define the dimensions of the video player.**

**### Optional Attributes:**

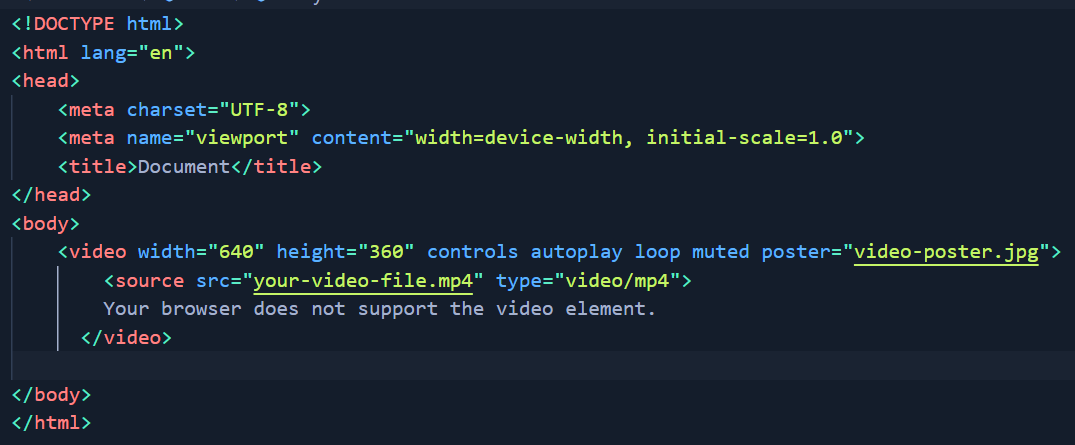
**- autoplay: Starts playing the audio or video automatically when the page loads.**

**- loop: Loops the audio or video playback.**

**- muted: Starts the audio or video with the sound muted.**

**- poster: Specifies an image to be displayed while the video is downloading or until the user hits play.**

**### Example with Optional Attributes:**

****

**### Browser Compatibility:**

**Make sure your audio and video files are in formats supported by major browsers (e.g., MP3 and MP4). Additionally, different browsers may support different formats, so providing multiple source elements with different file types increases compatibility.**

**Remember to replace "your-audio-file.mp3" and "your-video-file.mp4" with the actual file paths or URLs of your audio and video files.**

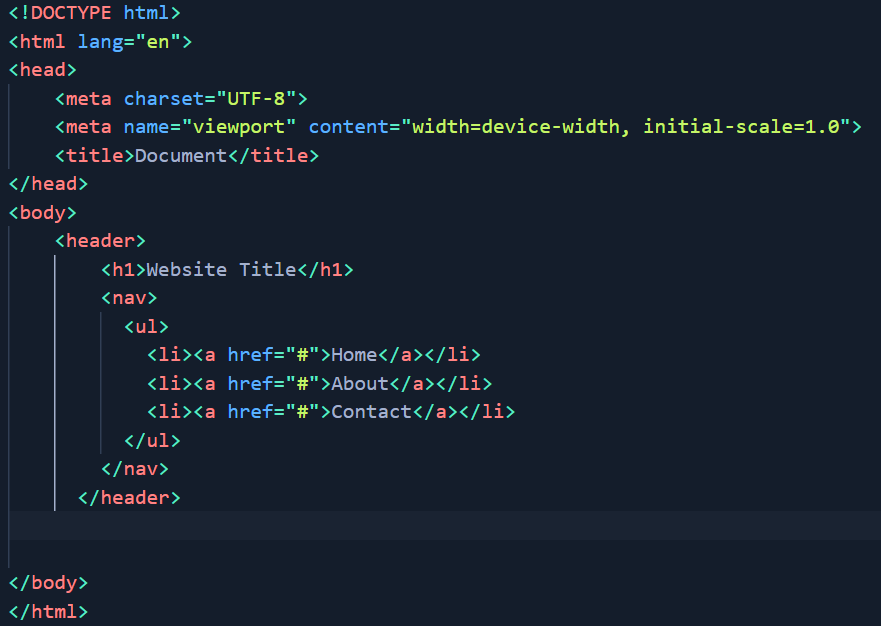
**Always consider accessibility by providing alternative content (e.g., text or links) for users who cannot access or play the audio or video content.**

Semantic element in HTML5?

ANS :- Semantic elements in HTML5 are tags that carry meaning about the structure and content of a web page. They provide more information to both browsers and developers about the purpose and importance of the content they enclose. Using semantic elements makes your HTML code more readable, understandable, and accessible. Here are some key semantic elements introduced in HTML5:

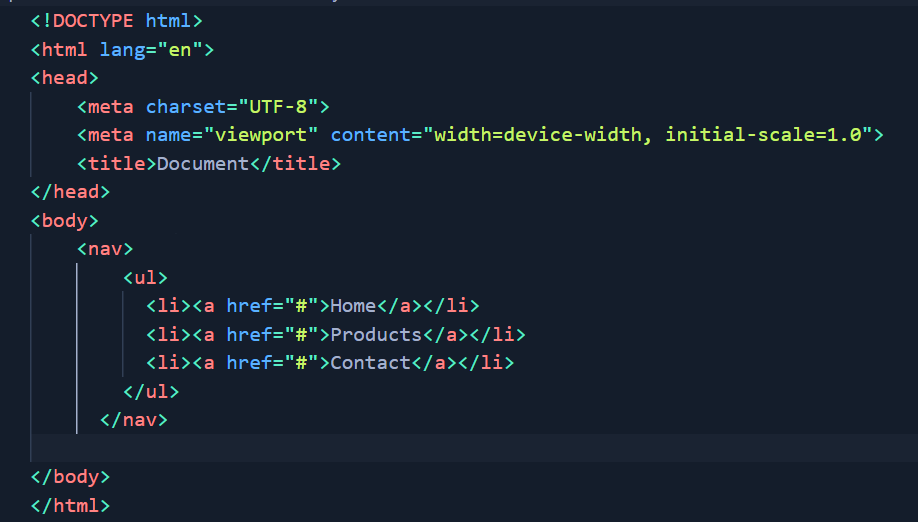
1. <header>:

Represents a group of introductory or navigational aids and can include headings, logos, and navigation elements.



2. <nav>:

Represents a section of the document intended for navigation links, such as menus or tables of contents.



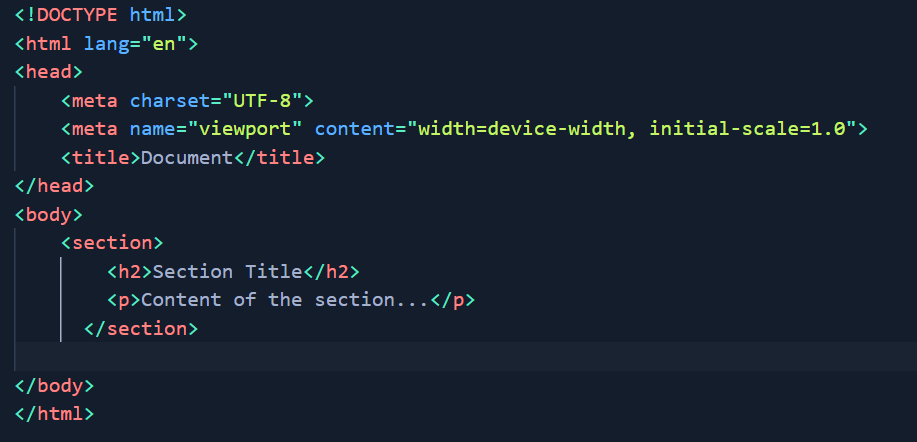
3. <article>:

Represents a self-contained piece of content that could be distributed and reused independently, such as a news article or blog post.



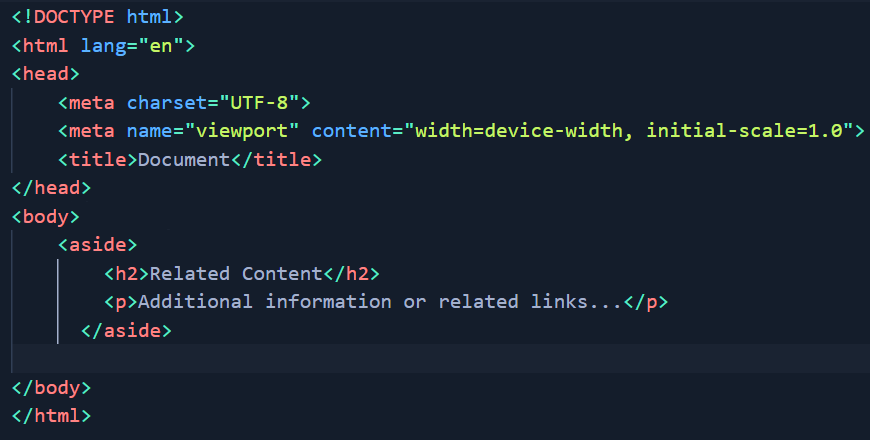
4. <section>:

Defines a section in a document, grouping together related content. It is often used to create a thematic grouping of content within an `<article>`.



5. <aside>:

Represents content that is tangentially related to the content around it, such as sidebars or pull quotes.



6. <footer>:

Represents a footer for a section or a page, typically containing metadata, copyright information, and other related content.



7. <main>:

Represents the main content of the `<body>` element in a document. It excludes content that is repeated across multiple pages, such as headers and footers.



These semantic elements help convey the structure and meaning of your content, improving accessibility and search engine optimization. Using them appropriately can enhance the overall clarity and maintainability of your HTML code.

Canvas and SVG tags ??

ANS :- The <canvas> and <svg> tags in HTML are both used to create graphics and visual content on web pages, but they have different approaches and use cases.

### <canvas> Element:

The <canvas> element provides a bitmap-based rendering surface that you can draw on using JavaScript. It is part of the HTML5 specification and is well-suited for dynamic graphics, animations, and interactive content.

#### Example of <canvas>:



In this example, a <canvas> element is created with a specified width and height. The JavaScript code gets the 2D rendering context of the canvas and draws a blue rectangle on it.

### <svg> Element:

The <svg> element is a vector graphics format based on XML (Scalable Vector Graphics). SVG is more declarative and can be created directly in HTML or as standalone XML files. It's suitable for static graphics, icons, diagrams, and scalable images.

#### Example of <svg>:



In this example, an <svg> element is used to draw a blue rectangle. The <rect> element is a child of the <svg> element, and its attributes define the position, size, and color of the rectangle.

### Choosing Between <canvas> and <svg>:

- Use <canvas> when:

- You need dynamic and interactive graphics.

- You want pixel-based rendering.

- You are creating complex animations or real-time graphics.

- Use <svg> when:

- You want to create scalable graphics.

- You are working with static images, icons, or diagrams.

- You prefer a declarative approach to graphics.

Both <canvas> and <svg> have their strengths, and the choice between them depends on the specific requirements of your project. You can also use them together in a single web page if needed, depending on the use cases for each.