**Module - 5**

**HTML5**

**• What are the new tags added in HTML5?**

HTML5 introduced several new tags to enhance the structure and semantics of web documents. Some of the key new tags introduced in HTML5 include:

<header>: Defines a header for a document or section.

<footer>: Defines a footer for a document or section.

<nav>: Defines a set of navigation links.

<article>: Defines an independent, self-contained content piece.

<section>: Defines a section v a document.

<aside>: Defines content aside from the content (like a sidebar).

<figure>: Specifies self-contained content, such as illustrations, diagrams, photos, code listings, etc.

<fig caption>: Provides a caption or legend for a <figure> element.

<main>: Represents the main content of the <body> of a document or application.

<time>: Represents a specific period in time or a range of time.

<mark>: Defines marked or highlighted text.

<progress>: Represents the progress of a task.

<meter>: Represents a scalar measurement within a known range.

<details>: Defines additional details that the user can view or hide.

<summary>: Defines a visible heading for the <details> element.

<data grid>: Defines a region that contains data for a data grid.

<dialog>: Represents a dialog box or other interactive component.

<template>: Defines content that should be hidden when the page loads but can be instantiated later on.

<input type="date">, <input type="time">, <input type="email">, <input type="url">, <input type="range">, etc.: Various new input types introduced for form elements.

<canvas>: Provides a space on which you can use JavaScript to draw graphics and animations.

<audio> and <video>: Embeds audio or video content respectively.

<source>: Specifies multiple media resources for <audio> and <video> elements.

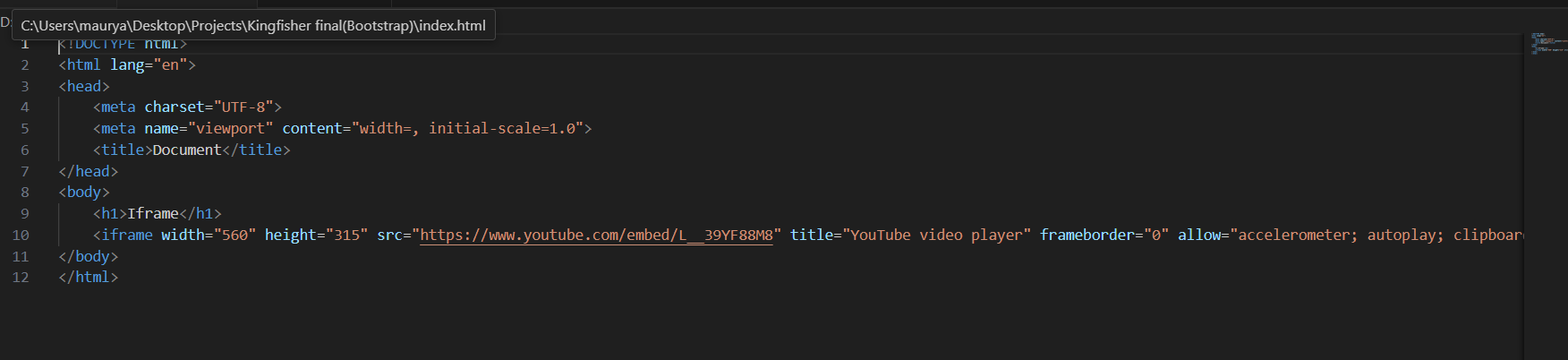
<track>: Specifies text tracks (such as subtitles or captions) for <audio> and <video> elements.

These are just a few of the most commonly used new tags introduced in HTML5. They aim to improve the semantics, accessibility, and structure of web documents.

**• How to embed audio and video in a webpage?**

To embed audio in HTML, we use the <audio> tag. Before HTML5, audio cannot be added to web pages in the Internet Explorer era. To play audio, we used web plugins like Flash. After the release of HTML5, it is possible. This tag supports Chrome, Firefox, Safari, Opera, and Edge in three audio formats – MP3, WAV, OGG. Only Safari browser doesn’t support OGG audio format.

To embed video in HTML, we use the <video> tag. It contains one or more video sources at a time using <source> tag. It supports MP4, WebM, and Ogg in all modern browsers. Only Ogg video format doesn’t support in Safari browser.

****

**• Semantic element in HTML5?**

HTML5 introduced several new semantic elements to help developers create more meaningful and structured web pages. Semantic elements are tags that clearly describe their meaning to both the browser and the developer, making the HTML code more understandable and improving accessibility and search engine optimization (SEO). Some of the key semantic elements introduced in HTML5 include:

<header>: Represents introductory content, typically containing navigation links, logos, headings, etc. It usually appears at the top of a page or section.

<footer>: Represents the footer of a page or section, containing information such as copyright notices, contact information, etc. It typically appears at the bottom of a page or section.

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

<article>: Represents a self-contained piece of content that could be distributed and independently reusable, such as blog posts, news articles, forum posts, etc.

<section>: Represents a thematic grouping of content, typically with a heading. It's a generic container for content that doesn't have a more specific semantic element available.

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

<main>: Represents the main content of the document, excluding any content that is repeated across multiple pages, such as headers, footers, navigation menus, etc.

<figure> and <figcaption>: <figure> represents self-contained content, such as images, videos, diagrams, etc., while <figcaption> represents a caption or legend for the content inside the <figure> element.

<details> and <summary>: <details> represents a disclosure widget from which the user can obtain additional information or controls, and <summary> represents a summary or label for the content inside the <details> element.

These semantic elements help developers create well-structured HTML documents that are easier to understand, maintain, and navigate for both humans and machines. They also improve accessibility by providing clearer semantics for assistive technologies.

**• Canvas and SVG tags**

SVG (Scalable Vector Graphics) is an XML-based vector image format, suitable for creating graphics and interactive elements. [HTML5 Canvas](https://www.geeksforgeeks.org/html-canvas-tag/) is more like a blank canvas where you can draw whatever you want using pixels. [SVG](https://www.geeksforgeeks.org/svg/) is great for logos and icons, while Canvas is good for dynamic animations and games.

**SVG:**

**Scalable Vector Graphics (SVG)** is an XML-based image format used to define two-dimensional vector-based graphics for the web. Unlike raster images (Ex .jpg, .gif, .png, etc.), a vector image can be scaled up or down to any extent without losing the image quality. An SVG image is drawn out using a series of statements that follow the XML schema — that means SVG images can be created and edited with any text editor, such as Notepad. There are several other advantages of using SVG over other image formats like JPEG, GIF, PNG, etc.

**Example:**In this example, we will see the code for drawing a circle using SVG elements.

**Canvas:**

The HTML element is used to draw graphics on the fly, via scripting (usually JavaScript). The element is only a container for graphics. You must use a script to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

**Example:**In this example, we will see the code for drawing a square using Canvas elements.