# **Web Development Module (HTML5) - 3**

**Q1: What are the new tags added in HTML5?**

**Ans**: HTML3 is not a commonly used version of HTML. The most widely known and used version of HTML is HTML4, which was succeeded by HTML5. HTML5 introduced many new elements and features compared to HTML4. To clarify, there is no HTML3. Here are some of the new tags and elements introduced in HTML5 compared to HTML4:

1. **<header> and <footer>:** These tags define the header and footer sections of a document or a section of a document.
2. **<nav>:** This tag is used to define navigation menus, making it easier to structure website navigation.
3. **<section>:** The <section> element is used to define sections within a document, such as chapters, tabs, or content groupings.
4. **<article>:** This element represents a self-contained composition in a document, such as a blog post or a news article.
5. **<aside>:** The <aside> tag is used for content that is tangentially related to the content around it, such as sidebars, pull quotes, or advertising.
6. **<figure> and <figcaption>:** These elements are used to associate a caption with an image, illustration, diagram, or other media content.
7. **<main>:** The <main> element is used to mark the main content of a document, providing semantic significance to search engines and assistive technologies.
8. **<time>:** This tag is used to represent a specific time or a range of time, which can be helpful for date and time information.
9. **<mark>:** The <mark> element is used to highlight or mark a specific portion of text within a document.
10. **<meter>:** This tag is used to represent a measurement within a known range, such as a progress bar or a gauge.
11. **<progress>:** The <progress> element is used to create a progress bar, indicating the completion status of a task.
12. **<datalist>:** It provides a list of predefined options for an <input> element with the list attribute.
13. **<output>:** The <output> element is used to display the result of a calculation or script.
14. **<canvas>:** This element provides a way to draw graphics and animations directly within the browser using JavaScript.
15. **<video> and <audio>:** These elements allow embedding video and audio content directly into a web page, with support for various formats.
16. **<source>:** Used in conjunction with the <video> and <audio> elements, it allows you to specify multiple sources for media content to ensure cross-browser compatibility.
17. **<details> and <summary>:** These elements create a disclosure widget that allows you to show or hide additional information on user interaction.
18. **<dialog>:** The <dialog> element is used to create a dialog or modal window for user interactions.

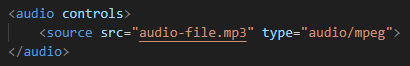
These are just some of the new tags and elements introduced in HTML5 compared to HTML4. HTML5 also brought improvements in form elements, multimedia support, and better handling of semantic markup, making it a more versatile and powerful language for web development.

**Q2: How to embed audio and video in a webpage?**

**Ans:** To embed audio and video in a webpage in simple terms, you can use the <audio> and <video> HTML elements. Here's how to do it:

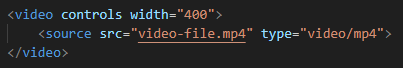
1. Audio:

To embed audio (like music or a sound clip), use the <audio> element. You specify the audio file's source (URL) using the src attribute. Here's an example:



2. Video:

To embed video (like a movie or video clip), use the <video> element. You also specify the video file's source using the src attribute. Here's an example:



Make sure to provide alternative text (the text between the opening and closing tags) for users whose browsers don't support the audio or video element. This text will be displayed as a fallback message.

Remember to choose the appropriate file formats (e.g., MP3, Ogg for audio, and MP4, WebM for video) and ensure that your web server is correctly configured to serve these files.

**Q3: Semantic element in HTML5?**

**Ans:** In simple terms, a semantic element in HTML5 is a special tag that provides meaning to the structure and content of a web page. It helps both humans and machines (like search engines or assistive technologies) understand the purpose of different parts of a webpage. Instead of using generic tags like <div> or <span> for everything, semantic elements offer clarity by describing the content's role.

For example, in HTML5, you can use:

1. **<header>:** To mark the header section of your page, which often includes the site's logo and main navigation.
2. **<nav>:** To indicate a navigation menu, making it clear that these links are for navigating around the website.
3. **<main>:** To identify the primary content area of your page, helping search engines and screen readers focus on the main content.
4. **<article>:** To mark self-contained content, like a blog post or news article.
5. **<section>:** To group related content within a document, such as chapters in a book.
6. **<footer>:** To define the footer section, which often includes copyright information and links to related content.

Using these semantic elements improves the accessibility, search engine optimization, and overall structure of your web page. It makes it easier for both people and machines to understand and interact with your content.

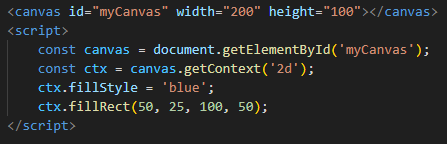
**Q4: Canvas and SVG tags**

**Ans:** Canvas and SVG are two different ways to create and manipulate graphics in HTML5. Here's a brief overview of both:

**1. Canvas:**

* **Canvas Element:** In HTML5, the <canvas> element allows you to draw graphics using JavaScript. It provides a low-level, immediate-mode 2D drawing API.
* **Drawing Context:** You use JavaScript to get a drawing context for the canvas, and then you can draw shapes, lines, text, and images on the canvas using various methods provided by the context object.
* **Pixel-Based:** The canvas is pixel-based, meaning you manipulate individual pixels to create graphics. It's best suited for things like games, charts, or other dynamic graphics.
* **Scalability:** Canvas graphics do not scale well, and they do not have built-in support for interactivity, which you need to implement manually.

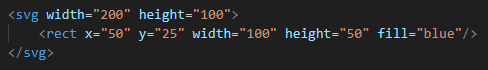
**Example:**



**2. SVG (Scalable Vector Graphics):**

* **SVG Element:** SVG is a language for describing two-dimensional vector graphics. In HTML5, you can embed SVG graphics directly using the <svg> element.
* **Vector Graphics:** SVG is resolution-independent and uses mathematical descriptions of shapes, making it ideal for icons, logos, illustrations, and any graphics that need to scale smoothly without loss of quality.
* **Structured XML:** SVG is based on XML, so you can create and manipulate SVG graphics using HTML or XML, and you can also style and animate them using CSS and JavaScript.
* **Interactivity:** SVG supports interactivity features such as event handling, making it suitable for creating interactive diagrams and maps.

**Example:**



In summary, the choice between <canvas> and SVG depends on your specific use case. Use <canvas> for pixel-based, dynamic graphics like games, and SVG for scalable, vector-based graphics like icons and illustrations. Each has its strengths and weaknesses, and the choice depends on what you want to achieve in your web application.