***Module (HTML5)\_3***

**1). What are the new tags added in HTML5?**

**Ans.** HTML5 introduced many new tags that were not available in previous versions of HTML. Some of the new tags in HTML5 are:

1. **<article>:** Defines an independent article or section of content within a webpage.
2. **<header>:** Defines the header section of a webpage or a section within a webpage.
3. **<footer>:** Defines the footer section of a webpage or a section within a webpage.
4. **<nav>:** Defines a section of a webpage that contains navigation links.
5. **<section>:** Defines a section within a webpage.
6. **<aside>:** Defines content that is related to the main content of a webpage, but can be considered tangential or supplementary.
7. **<main>:** Defines the main content of a webpage.
8. **<figure>:** Defines a self-contained content block, such as an image, diagram, or chart.
9. **<figcaption>:** Defines a caption or description for a <figure> element.
10. **<progress>:** Defines a progress bar or indicator for the completion of a task or process.
11. **<time>:** Defines a specific time or date.
12. **<meter>:** Defines a scalar measurement within a known range, such as disk usage or memory usage.

These are just a few examples of the new tags in HTML5. There are many more new tags and attributes available in HTML5 that can help developers create more semantic and accessible webpages.

**2). 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 are the basic steps to embed audio and video:

1. Choose the audio or video file you want to embed on your webpage and make sure it's in a compatible format (e.g. MP3, WAV, MP4, etc.).
2. Create an <audio> or <video> element in your HTML code, and specify the source of the audio or video file using the src attribute.

The controls attribute adds audio or video controls to the player, allowing users to play, pause, rewind, and adjust the volume. The type attribute specifies the MIME type of the file, and the text "Your browser does not support the audio/video element." will be displayed in case the user's browser doesn't support the element.

You can also add other attributes to customize the appearance and behavior of the audio or video player, such as autoplay, loop, preload, poster, and more.

**3). Semantic element in HTML5?**

**Ans.** In HTML5, semantic elements are those that give meaning to the content they contain. They help to describe the structure of a webpage in a more meaningful and organized way, making it easier for search engines, screen readers, and other tools to understand and parse the content. Some examples of semantic elements in HTML5 are:

1. **<header>:** Defines the header section of a webpage or a section within a webpage.
2. **<nav>:** Defines a section of a webpage that contains navigation links.
3. **<main>:** Defines the main content of a webpage.
4. **<article>:** Defines an independent article or section of content within a webpage.
5. **<section>:** Defines a section within a webpage.
6. **<aside>:** Defines content that is related to the main content of a webpage, but can be considered tangential or supplementary.
7. **<footer>:** Defines the footer section of a webpage or a section within a webpage.
8. **<figure>:** Defines a self-contained content block, such as an image, diagram, or chart.
9. **<figcaption>:** Defines a caption or description for a <figure> element.
10. **<time>:** Defines a specific time or date.
11. **<address>:** Defines the contact information for the author or owner of a webpage.
12. **<mark>:** Defines text that is highlighted for emphasis.

Using semantic elements in HTML5 not only makes the content more accessible and understandable for assistive technologies, but it also improves the search engine optimization (SEO) of a webpage, as search engines can better understand the content and context of the webpage.

**4). Canvas and SVG tags**

**Ans.** Canvas and SVG are two different ways to draw graphics on a webpage using HTML5.

The <canvas> tag provides a blank canvas on which you can use JavaScript to draw graphics and animations. You can draw lines, shapes, text, images, and more using the canvas API. The canvas is essentially a rectangular bitmap that can be manipulated using JavaScript. The <canvas> element has attributes such as width and height that define the size of the canvas. Here is an example of a simple canvas drawing:

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

<script>

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

var ctx = canvas.getContext("2d");

ctx.fillStyle = "#FF0000";

ctx.fillRect(0, 0, 150, 75);

</script>

On the other hand, the **<svg>** tag defines a vector-based graphics image that is scalable and can be styled using CSS. SVG stands for Scalable Vector Graphics, and it uses XML syntax to define vector graphics. You can create shapes, text, and images using SVG elements and attributes. SVG graphics can be interactive and animated using JavaScript. Here is an example of a simple SVG drawing:

<svg width="100" height="100">

<rect x="10" y="10" width="80" height="80" style="fill:red;" />

</svg>

In this example, we create a red square using the **<rect>** element and style it using the **style** attribute. SVG also has a range of other elements and attributes that can be used to create more complex graphics.

In summary, **<canvas>** is a bitmap-based graphics system that requires JavaScript to draw on, while **<svg>** is a vector-based graphics system that uses XML syntax and can be styled with CSS. Both have their own strengths and can be used to create a wide range of graphics and animations.