Module 5 – Frontend – HTML5

**Question 1: Difference Between HTML and HTML5**

| **Feature** | **HTML (Older Versions)** | **HTML5 (Latest Version)** |
| --- | --- | --- |
| **Version** | HTML 4.01 or earlier | HTML5 |
| **Doctype Declaration** | Long & complicated  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"...> | Simple  <!DOCTYPE html> |
| **Multimedia Support** | No built-in support  (needed Flash/Java plugins) | <audio> and <video> tags for native support |
| **Semantic Elements** | Mostly non-semantic: <div>, <span> | Added semantic tags like <header>, <footer>, <nav>, <article>, <section> |
| **Graphics Support** | No support | Built-in <canvas> and SVG support |
| **Form Enhancements** | Basic controls only | New inputs: <date>, <email>, <range>, <color>, etc. |
| **APIs** | Not supported | Supports new APIs: Geolocation, Web Storage, WebSockets, Drag and Drop, etc. |
| **Mobile-Friendly** | Not responsive by default | Designed with mobile in mind (responsive, touch support) |

**✅ Question 2: Additional Tags Introduced in HTML5**

Here are some **important HTML5 tags** that were **not in older HTML versions**:

**📌 Semantic Tags (Structure)**

* <header> – Defines a page or section header
* <footer> – Defines a page or section footer
* <nav> – Navigation links
* <article> – Self-contained content (e.g., blog post)
* <section> – Generic section of a document
* <aside> – Sidebar content
* <main> – Main content of the page
* <figure> – Used with <figcaption> to group media and captions

**🎥 Media Tags**

* <audio> – Embed audio files
* <video> – Embed video files
* <source> – Specify media file formats for audio/video

**🎨 Graphics and Drawing**

* <canvas> – For drawing graphics via JavaScript
* <svg> – For scalable vector graphics (also used in older XML, but integrated better in HTML5)

**🔠 Form Input Types**

* <input type="email">, <input type="date">, <input type="range">, <input type="color">, <input type="url">, etc.

**🔐 Others**

* <datalist> – Suggest input options
* <output> – Display output from calculations
* <progress> – Progress bar
* <meter> – Display scalar measurement (e.g., disk usage)

Lab Assignment(Task)

Create a audio video tag

⇒ Also applied properties like muted loop autoplay

⇒ Create some shape using canvas tag in html

⇒ Create some shape using svg tag in html

**✅ 1. Audio & Video Tags with Properties**

**🎵 Audio Tag:**

html

CopyEdit

<audio controls autoplay muted loop>

<source src="your-audio-file.mp3" type="audio/mp3">

Your browser does not support the audio element.

</audio>

**🎬 Video Tag:**

html

CopyEdit

<video width="400" height="300" controls autoplay muted loop>

<source src="your-video-file.mp4" type="video/mp4">

Your browser does not support the video element.

</video>

**Attributes used:**

* autoplay: Starts automatically
* muted: Starts muted (required for autoplay in most browsers)
* loop: Repeats automatically
* controls: Shows play/pause, volume, etc.

**✅ 2. Canvas Tag: Create Shapes Using JavaScript**

**🖌️ Canvas with Rectangle and Circle:**

html

CopyEdit

<canvas id="myCanvas" width="300" height="200" style="border:1px solid #000000;"></canvas>

<script>

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

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

// Draw a rectangle

ctx.fillStyle = "blue";

ctx.fillRect(20, 20, 100, 50);

// Draw a circle

ctx.beginPath();

ctx.arc(200, 70, 40, 0, 2 \* Math.PI);

ctx.fillStyle = "red";

ctx.fill();

</script>

3. SVG Tag: Create Shapes Without JavaScript

<!DOCTYPE html>

<html>

<head>

<title>SVG Shapes Example</title>

</head>

<body>

<h2>SVG Shapes</h2>

<svg width="400" height="300" style="border: 1px solid #000">

<!-- Rectangle -->

<rect x="10" y="10" width="100" height="50" fill="skyblue" stroke="black" stroke-width="2" />

<!-- Circle -->

<circle cx="200" cy="40" r="30" fill="lightgreen" stroke="black" stroke-width="2" />

<!-- Ellipse -->

<ellipse cx="320" cy="40" rx="50" ry="30" fill="lightpink" stroke="black" stroke-width="2" />

<!-- Line -->

<line x1="0" y1="100" x2="400" y2="100" stroke="red" stroke-width="2" />

<!-- Polygon (Triangle) -->

<polygon points="50,150 100,200 0,200" fill="orange" stroke="black" stroke-width="2" />

<!-- Polyline (Zigzag) -->

<polyline points="150,150 180,180 210,150 240,180" fill="none" stroke="blue" stroke-width="2" />

<!-- Text -->

<text x="10" y="280" font-size="20" fill="purple">SVG Shape Example</text>

</svg>

</body>

</html><!DOCTYPE html>

<html>

<head>

<title>SVG Shapes Example</title>

</head>

<body>

<h2>SVG Shapes</h2>

<svg width="400" height="300" style="border: 1px solid #000">

<!-- Rectangle -->

<rect x="10" y="10" width="100" height="50" fill="skyblue" stroke="black" stroke-width="2" />

<!-- Circle -->

<circle cx="200" cy="40" r="30" fill="lightgreen" stroke="black" stroke-width="2" />

<!-- Ellipse -->

<ellipse cx="320" cy="40" rx="50" ry="30" fill="lightpink" stroke="black" stroke-width="2" />

<!-- Line -->

<line x1="0" y1="100" x2="400" y2="100" stroke="red" stroke-width="2" />

<!-- Polygon (Triangle) -->

<polygon points="50,150 100,200 0,200" fill="orange" stroke="black" stroke-width="2" />

<!-- Polyline (Zigzag) -->

<polyline points="150,150 180,180 210,150 240,180" fill="none" stroke="blue" stroke-width="2" />

<!-- Text -->

<text x="10" y="280" font-size="20" fill="purple">SVG Shape Example</text>

</svg>

</body>

</html>