Module No. 1

Introduction to HTML 5, syntax, attributes, events, SVG, Web storage, Introduction to Canvas, Audio & Video, Geolocations, Drag & Drop, Web workers, working with Fonts, working with other graphics.

Introduction to HTML 5

- ✓ Html5 is a language developed by Ian Hickson (Google) and David Hyatt (Apple) in 22 January 2008.
- ✓ Filename extension .html , .htm
- ✓ Open Source Hyper Text Resources
- ✓ HTML5 =HTML + CSS+JS
- ✓ HTML= Next Generation Features for Modern web Development.
- ✓ Html5=WHATWG+W3C
- ✓ It is Web Platform. It will be the new standard for HTML.
- ✓ HTML5 is currently under development as the next major revision of the HTML standard.
- ✓ HTML5 is a markup language for structuring and presenting World Wide Web and a core technology of the Internet.

HTML5 Introduction

- HTML5 is the fifth version of Hypertext Markup Language (HTML), a standard language used to structure webpages.
- It defines how content on a webpage should be structured and displayed. Here are some key points of HTML5.
- Provides multimedia support by embedding audio and video without plugins.
- Adds new form controls like date and email input types.
- Enables web storage to store data offline for better performance.
- Uses semantic elements such as <header> and <footer> for clearer structure.
- Delivers improved performance, especially on mobile devices.

1991–1999

- **HTML was created by web legend Tim Berners-Lee in 1991, and HTML versions 1–4 were developed throughout the 1990s by W3C. In these early days of widespread internet use, HTML efficiently displays the vast majority of web content, since at this time it largely consists of static, non-interactive sites.**

2000

- **W3C recommends XHTML 1.0 – an XML-based markup language that mirrors/extends HTML. Previous versions of HTML are now showing their age, struggling to handle the latest generation of multimedia, and interactive sites. To get the best results, developers are resorting to third-party plugins.**

2004

- **Development of HTML is closed by W3C, who instead decide to focus on XHTML. WHATWG is formed to develop HTML further, with the aim of reflecting the modern dynamic web, while keeping backwards compatibility with existing HTML code.**

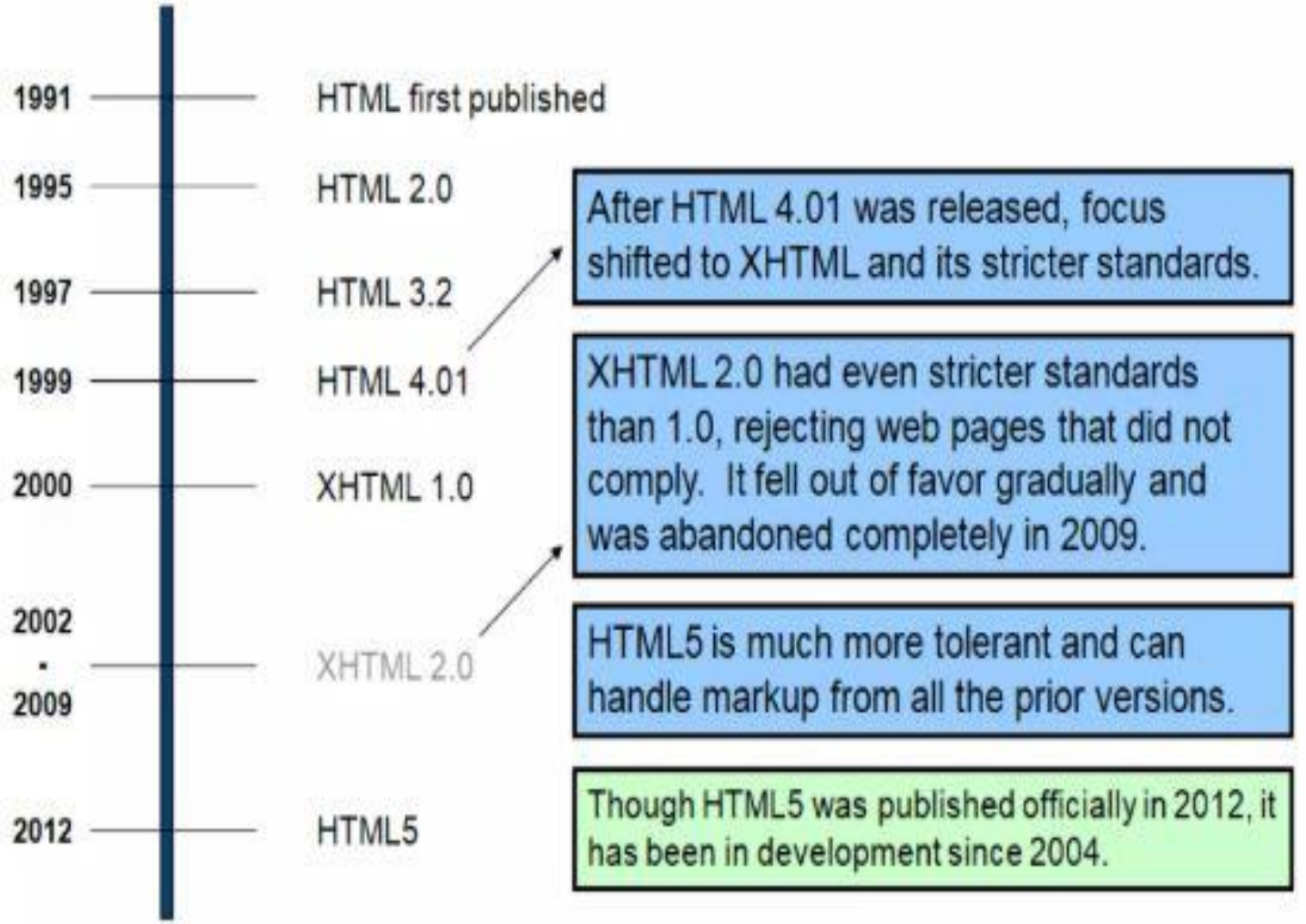
2004–2006

- **WHATWG gains support from major web browser developers. In 2006, W3C also announced its support for the project.**

2008

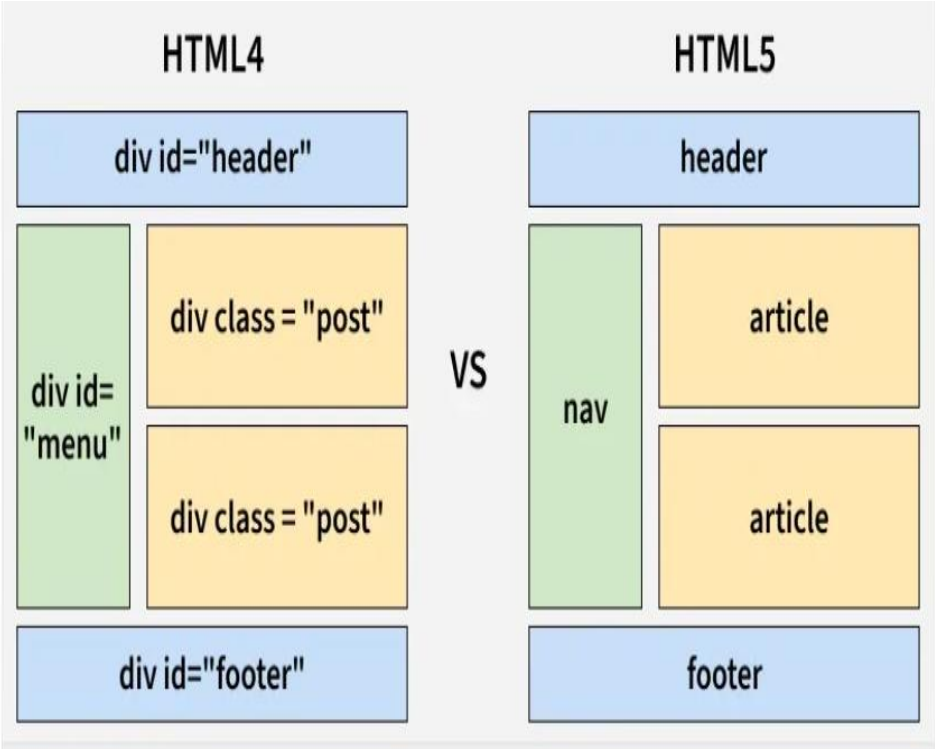
- **The first public draft of HTML5 is released by WHATWG. Web Hypertext Application Technology Working Group**

HTML Evolution:



Why use HTML5

- It allows you to play a video and audio file.
- It allows you to draw on a canvas.
- It facilitate you to design better forms and build web applications that work offline.
- It provides you advance features for that you would normally have to write JavaScript to do.
- The most important reason to use HTML 5 is, we believe it is not going anywhere.
- It will be here to serve for a long time according to W3C recommendation.


HTML4
HTML5
div id="header"
header
div id= "menu"
div class = "post"
div class = "post"
VS
nav
article
article
div id="footer"
footer


<Audio>
<3D>
<Video>
New Features of HTML5
HTML
<Games>
<Storage>
<Type>
<Motion>

HTML5-Syntax

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>My Web Page</title>
</head>
<body>
<!-- This is a comment in HTML -->
<h1>welcome to web technology</H1>
</body>
</html>
```

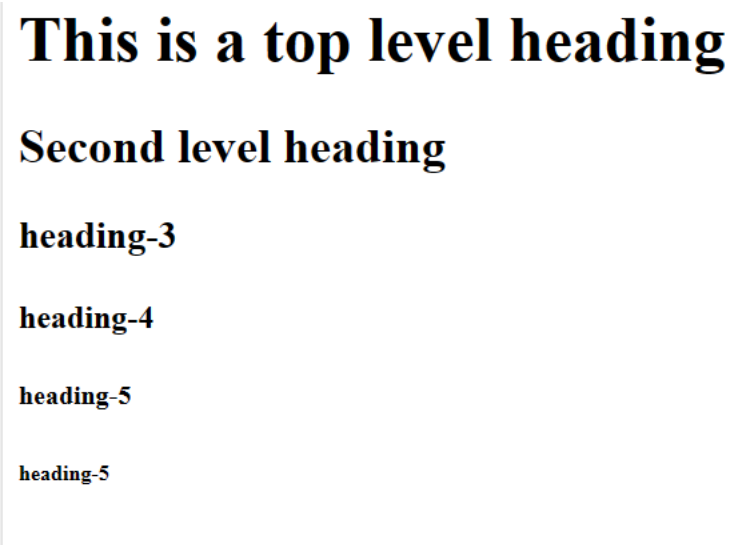
Text Formatting Tags

- Text formatting tags modify the text between the opening tag and the closing tag

<b>	**Used to display bold text**
<i>	*Used to display italic text*
<u>	Used to display underlined text
<sup>	used to superscript a text(Sample$^{superscript)}$
<sub>	used to subscript a text(Sample$_{subscript\ )}$
<MARK>	Defines marked/highlighted text
font	*To change style of text*
<pre></pre>	defines preformatted text.
<del></del>	Deleted text – strike through
<hr>	Defines horizontal line

Heading Tags…

```
<HTML>
<BODY>
<H1>This is a top level heading</H1>
<H2>Second level heading</H2>
<h3> heading-3</h3>
<h4>heading-4</h4>
<h5> heading-5</h5>
<h6> heading-5</h6>
</BODY>
</HTML>
```



New Added Elements in HTML 5

- HTML5 introduced several new elements that help structure a webpage better and make it more interactive.
- These elements not only improve the design but also make the code more readable and accessible.
- Here are some key elements of HTML5:
- **<article>:** The <article> tag represents independent, self-contained content like articles or posts.
- **<aside>:** The <aside> tag holds secondary or related information supporting the main content.
- **<figcaption>:** The <figcaption> tag provides a caption or description for a <figure> element.
- **<figure>:** The <figure> tag groups self-contained media such as images, diagrams, or code.
- **<header>:** The <header> tag contains introductory content like headings or navigation links.
- **<footer>:** The <footer> tag includes closing content such as author info or copyright notes.
- **<main>:** The <main> tag defines the central, most important content of the webpage.
- **<mark>:** The <mark> tag highlights or emphasizes specific text.

New Added Elements in HTML 5

.

- **<nav>:** The <nav> tag groups navigation links for moving around the site.
- **<section>:** The <section> tag organizes related content into thematic blocks.
- **<details>:** The <details> tag creates expandable or collapsible content sections.
- **<summary>:** The <summary> tag provides the clickable heading for a <details> element.
- **<time>:** The <time> tag represents dates or times in a readable and machine-usable format.
- **<bdi>:** The <bdi> tag isolates text with unknown or mixed text direction.
- **<wbr>:** The <wbr> tag adds an optional line-break position in long words.
- **<datalist>:** The <datalist> tag provides autocomplete suggestions for an <input> element.
- **<output>:** The <output> tag displays results of calculations or script actions.
- **<progress>:** The <progress> tag visually represents the progress of a task.
- **<svg>:** The <svg> tag embeds scalable vector graphics into a webpage.
- **<canvas>:** The <canvas> tag draws dynamic graphics using JavaScript.
- **<audio>:** The <audio> tag embeds audio content like music or sound effects.
- **<embed>**: The <embed> tag inserts external resources such as media players or apps.
- **<source>:** The <source> tag defines media sources for <audio> and <video> elements.
- **<track>:**The <track> tag provides captions or subtitles for videos.
- **<video>:** The <video> tag embeds video content directly into a webpage.

HTML5 Advantages

- HTML5 offers several benefits that enhance modern web development like:
- Works on all major browsers and devices (cross-platform).
- Lets websites access user location (geolocation).
- Supports offline use through Web Storage or AppCache.
- Uses semantic tags for cleaner, simpler code.
- Optimized for smooth performance on mobile devices.

HTML5 Disadvantages

Despite its strengths, HTML5 comes with some drawbacks:

- Older browsers may not support all HTML5 features (browser inconsistencies).
- Some older mobile devices struggle with HTML5 (mobile compatibility issues).
- Web Storage and similar features can pose security risks if misused.
- Advanced APIs introduce complexity for beginners.
- IE 8 and below show poor support for HTML5.

Example of HTML5

- **Adding a Video Using the <video> Tag**

```
<html>
<body>
   <video width="320" height="100" controls>
      <source src="movie.mp4" type="video/mp4">
      Your browser does not support the video tag.
   </video>
</body>
</html>
```

- The <video> tag is used to embed a video in the HTML page.
- The controls attribute adds play, pause, and volume control buttons.
- The <source> tag specifies the video file and format. If the video is not supported, the text inside the <video> tag will be displayed.

Best Practices of Using HTML5

- Use Semantic Tags: Improve SEO and accessibility by using tags like <header>, <footer>, <article>, and <section>.
- Ensure Cross-Browser Compatibility: Test your code across multiple browsers, especially older versions.
- Use <meta charset=”UTF-8″>: Always specify the character encoding at the start to avoid encoding issues.
- Mobile Responsiveness: Use the <meta name=”viewport” content=”width=device-width, initial-scale=1.0″> tag for mobile-friendly designs.

HTML Attributes

- **What are Attributes?**
- Attributes provide **extra information** about an element
- Written in the **opening tag**

Format:

- <tagname attribute="value">

Common HTML5 Attributes

(a) id Attribute

- Unique identification of an element
- Paragraph

(b) class Attribute

- Used to apply styles to multiple elements
- Hello

(c) style Attribute

- Applies inline CSS
- Red Text

(d) title Attribute

- Displays tooltip text
- Hover me

(e) href Attribute (Links)

- Google

(f) src Attribute (Images, Media)

- <img src="photo.jpg" alt="My Photo">

(g) alt Attribute

- Alternative text for images
- <img src="img.jpg" alt="Image not found">

HTML5 Form Attributes

(a) type Attribute

```
<input type="text">
<input type="email">
<input type="date">
```

(b) name Attribute

```
<input type="text" name="username">
```

(c) placeholder Attribute

```
<input type="text" placeholder="Enter name">
```

(d) required Attribute

```
<input type="email" required>
```

(e) value Attribute

```
<input type="text" value="Default Text">
```

Introduction to Canvas

The HTML5 **<canvas>** element is used to draw graphics on the fly via JavaScript. It's a powerful feature for rendering shapes, images, animations, games, and data visualizations—*all directly in the browser.*

Basic Syntax:

```
<canvas id="myCanvas" width="400" height="200"></canvas>
```

- The <canvas> tag creates a rectangular drawing area.
- By itself, it’s just a container; all drawing must be done with JavaScript.

Using the <canvas> Element for Drawing

```
<html>
<body>
  <canvas id="myCanvas" width="100" height="100"></canvas>
  <script>
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "blue";
    ctx.fillRect(50, 50, 100, 100);
  </script>
</body>
</html>
```

- The <canvas> tag is used to create an area for drawing graphics using JavaScript.
- The getContext("2d") method is used to set up a 2D drawing context on the canvas.
- The fillRect method draws a blue rectangle within the canvas.

Purpose	Method
Draw rectangle	fillRect(x, y, w, h)
Draw circle	arc(x, y, radius, startAngle, endAngle)
Draw line	moveTo(x, y) + lineTo(x, y) + stroke()
Set color	fillStyle, strokeStyle
Clear canvas	clearRect(x, y, w, h)
Draw text	fillText(text, x, y)

When to Use <canvas>

Use canvas when you need:

- **Dynamic, pixel-based graphics**
- **Games**
- **Image editing**
- **Real-time visualizations**

Using the <progress> Element to Show Task Progress

```
<html>
<body>
   <h1>HTML5 Progress Bar Example</h1>
   <progress value="70" max="100"></progress>
</body>
</html>
```

- The <progress> tag is used to display a progress bar on the webpage.
- The value attribute sets the current progress, and the max attribute defines the maximum value.

EVENTS

- HTML5 events are actions or occurrences that happen in the browser, such as a user clicking a button, a page finishing loading, or a video starting to play. These events can be "listened" for by JavaScript code, which can then execute specific functions in response. This allows for dynamic and interactive web pages.
- Here's a breakdown of common HTML5 event categories and some examples:

1. Window Events: These events are triggered on the window object.

- onload: Triggered once the entire page (including all resources like images and scripts) has finished loading.
- onunload: Triggered just before the page is unloaded (e.g., when the user navigates away or closes the window).
- onresize: Triggered when the browser window has been resized.
- onscroll: Triggered when the user scrolls the window.
- onerror: Triggered when a JavaScript error occurs on the page.

2. Form Events: These events are related to HTML forms and their elements.

- onsubmit: Triggered when a form is submitted.
- onreset: Triggered when a form is reset to its default values.
- onfocus: Triggered when an element (like an input field) gains focus.
- onblur: Triggered when an element loses focus.
- onchange: Triggered when the value of an element (like an input, select, or textarea) has been changed and the element loses focus.
- oninput: Triggered when the value of an input, textarea, or select element is changed.
- onselect: Triggered when the user selects some text in an input field or textarea.

3. Mouse Events: These events occur due to user interaction with the mouse.

- onclick: Triggered when an element is clicked.
- ondblclick: Triggered when an element is double-clicked.
- onmousedown: Triggered when a mouse button is pressed down over an element.
- onmouseup: Triggered when a mouse button is released over an element.
- onmousemove: Triggered when the mouse pointer moves while it is over an element.
- onmouseover: Triggered when the mouse pointer moves onto an element.
- onmouseout: Triggered when the mouse pointer moves out of an element.
- onwheel: Triggered when the mouse wheel is rotated over an element.

4. Keyboard Events: These events are related to user interaction with the keyboard.

- onkeydown: Triggered when a key is pressed down.
- onkeypress: Triggered when a key is pressed and released (for character-producing keys).
- onkeyup: Triggered when a key is released.

5. Media Events: These events are specific to HTML5 <audio> and <video> elements.

- onplay: Triggered when the media starts playing.
- onpause: Triggered when the media is paused.
- onended: Triggered when the media has reached the end.
- onvolumechange: Triggered when the volume of the media changes.
- onseeking: Triggered when the user starts moving to a new position in the media.
- onseeked: Triggered when the seeking operation is complete.
- ondurationchange: Triggered when the duration of the media changes.
- ontimeupdate: Triggered when the current playback position of the media changes.

SVG (Scalable Vector Graphics)

In HTML5, **SVG (Scalable Vector Graphics)** is used to define vector-based graphics for the web. SVG is an XML-based markup language that can be embedded directly into HTML documents. It allows for resolution-independent, scalable, and interactive graphics, making it ideal for responsive web design, data visualizations, icons, and more.

Basic Example of SVG in HTML5

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>SVG Example</title>
</head>
<body>
  <h1>My SVG Circle</h1>
  <svg width="200" height="200">
    <circle cx="100" cy="100" r="80" stroke="blue" stroke-width="4" fill="lightblue" />
  </svg>
</body>
</html>
```

Element	Description
<svg>	The container element
<circle>	A circle
<rect>	A rectangle
<line>	A line
<ellipse>	An ellipse
<polygon>	A closed shape with multiple sides
<path>	A complex shape using commands
<text>	Text inside SVG

Embedding SVG

1.Inline SVG (recommended for dynamic or styled content):

<svg>...</svg>

2. Object tag:

<object type="image/svg+xml" data="image.svg"></object>

3.Image tag (no interactivity):

Difference between SVG and Canvas

Feature	SVG	Canvas
Type	Vector-based	Pixel-based
Scaling	Scales without loss of quality	Loses quality when scaled
DOM access	Each shape is part of DOM	No DOM for shapes
Interactivity	Easy (events per element)	Harder (manual handling)
Performance	Better for static graphics	Better for animations & games
CSS support	Yes	No
JavaScript	Less JS needed	Heavy JS required

SVG

- Uses XML-based vector graphics
- Each shape is an element (`<circle>`, `<rect>`)
- Best for **logos, icons, diagrams, charts**
- Use **SVG** → static, scalable, interactive graphics

Canvas

- Draws graphics using JavaScript
- Works at pixel level
- Best for **games, animations, image editing**
- Use **Canvas** → fast, dynamic, real-time graphics

Geolocations

- **HTML Geolocation** is used to get the geographical position of a user. Due to privacy concerns, this position requires user approval.
- **Geo-location** in HTML5 is used to share the location with some websites and be aware of the exact location. It is mainly used for local businesses, and restaurants, or showing locations on the map. It uses JavaScript to give latitude and longitude to the backend server.
- Most browsers support Geolocation API. Geo-location API uses a global navigator object. **HTML Geolocation**, various properties, methods & their implementation through examples.

Geolocations

```
<html>
<body>
<p>Click on the button to get your coordinates.</p>
<button onclick="getLocation()">Try It</button>
<p id="demo"></p>


<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(success);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";  }}
```

```
function success(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}</script>

</body>
</html>
```

Geolocations