

## Module 5 – Mernstack – HTML5

### Theory

#### 1. Difference b/w HTML & HTML5?

**Ans:**

##### 1. DOCTYPE Declaration

- HTML: Long and complex:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- HTML5: Simple and short:

```
<!DOCTYPE html>
```

##### 2. New Semantic Tags

- HTML: Limited semantic tags. Used mostly <div> and <span>.
- HTML5: Introduced semantic tags like:
  - <header>, <footer>, <article>, <section>, <nav>, <aside>, etc.

##### 3. Multimedia Support

- HTML: Requires plugins (like Flash) for audio and video.
- HTML5: Native support for:
  - <audio> and <video> tags.

##### 4. Form Controls & Attributes

- HTML: Basic input types and limited validation.
- HTML5: New input types and attributes, like:
  - type="email", type="date", type="range", etc.
  - New attributes: placeholder, required, autofocus, pattern.

##### 5. Graphics Support

- HTML: No built-in support for drawing.
- HTML5: Supports graphics with:
  - <canvas> for drawing 2D graphics.
  - <svg> for vector graphics.

##### 6. Storage Capabilities

- HTML: Relies on cookies.
- HTML5: Introduced local storage options:
  - localStorage and sessionStorage (client-side storage).

##### 7. Browser Support

- HTML: Older standard, not optimized for modern browsers.

- HTML5: Designed to work across modern browsers and devices.

## 8. Offline Support

- HTML: No offline capabilities.
- HTML5: Added offline features through:
  - Application Cache (deprecated) and Service Workers (modern approach).

## 2. What are the additional tags used in HTML5?

**Ans:**

### ❖ Semantic Tags

HTML5 introduced tags that give meaning to page structure, making code more readable and accessible:

- `<header>` – Defines the top section of a page or a section, often includes logos or navigation.
- `<footer>` – Defines the bottom section, often includes copyright or contact info.
- `<nav>` – Represents a section with navigation links.
- `<section>` – Groups related content in a document.
- `<article>` – Defines independent, self-contained content like blog posts or news articles.
- `<aside>` – Content indirectly related to the main content, like sidebars.
- `<main>` – Represents the main content of a page.
- `<figure>` – Used to group media like images with a caption.
- `<figcaption>` – Provides a caption for the `<figure>`.
- `<mark>` – Highlights or emphasizes part of the text.
- `<time>` – Represents a specific date or time.
- `<details>` – Creates a collapsible section that can be toggled open or closed.
- `<summary>` – Provides a heading for the `<details>` element.

### ❖ Multimedia Tags

These allow embedding audio and video without needing plugins:

- `<audio>` – Embeds audio files.
- `<video>` – Embeds video content.
- `<source>` – Specifies media sources inside `<audio>` or `<video>`.
- `<track>` – Adds subtitles or captions to media.

### ❖ Graphics and Visuals

For drawing and vector graphics:

- `<canvas>` – Creates a space to draw 2D graphics using JavaScript.
- `<svg>` – Embeds scalable vector graphics directly in HTML.

### ❖ Form Enhancements

New input types and attributes (not tags, but worth noting):

Examples:

`<input type="email">`

`<input type="date">`

`<input type="range">`

`<input type="color">`

New attributes include: placeholder, required, autofocus, pattern.

### ❖ **Interactive Elements**

- `<dialog>` – Creates modal dialogs or pop-ups.
- `<menu>` – Represents a list of commands or context menu.
- `<menuitem>` – Represents a command within a menu (now deprecated).