

What is HTML?

HTML (HyperText Markup Language) is the standard markup language used to create and design **web pages**. It provides the **basic structure** of a web page by defining elements such as text, images, links, tables, forms, and multimedia.

HTML is **not a programming language**; it is a **markup language** that uses tags to describe the structure and content of a web page.

Basic Structure of an HTML Document

An HTML document consists of the following main parts:

1. <!DOCTYPE html> – Declares the document type
2. <html> – Root element of the document
3. <head> – Contains metadata (title, styles, scripts)
4. <title> – Defines the page title
5. <body> – Contains visible content of the web page

HTML Tags

- HTML uses **tags** enclosed within <>
- Tags usually come in **pairs**
 - Opening tag: <p>
 - Closing tag: </p>
- Some tags are **self-closing**
 - ,
, <hr>

Common HTML Elements

- Headings: <h1> to <h6>
- Paragraph: <p>
- Links: <a>
- Images:
- Lists: , ,

- Tables: <table>, <tr>, <th>, <td>
 - Forms: <form>, <input>, <textarea>, <select>
-

Advantages of HTML

- Simple and user-friendly
 - Free and open standard
 - Easy integration with CSS and JavaScript
 - Forms the foundation of all websites
-

Limitations of HTML

- Cannot perform logic or calculations
- Static by nature
- Requires CSS and JavaScript for styling and interactivity

Difference Between HTML and HTML5

Feature	HTML	HTML5
Version	Older version	Latest version
Doctype	Long and complex	Simple <!DOCTYPE html>
Multimedia support	Requires plugins	Built-in audio and video
Graphics	No native support	Supports Canvas and SVG
Semantic elements	Limited	New semantic tags introduced
Form controls	Basic	Advanced form input types
Storage	Cookies only	Local Storage & Session Storage
Mobile support	Limited	Optimized for mobile devices
Error handling	Poor	Better error handling
Browser compatibility	Partial	Fully supported by modern browsers

Forms in HTML (Core Programming Concept)

Form Structure

```
<form action="submit.html" method="post">
```

Input Types

Type	Purpose
text	Name
email	Email
password	Password
radio	Gender
checkbox	Skills
submit	Submit form
reset	Clear form

Validation Attributes

- required
 - maxlength
 - pattern
 - placeholder
-

10. Frames and Iframes

Frames

- Divide browser window
- Obsolete but still in syllabus

Iframe

```
<iframe src="page.html" width="400" height="300"></iframe>
```

Used to embed:

- Web pages
- Maps
- Videos

HTML5 vs HTML – Meta & Viewport Support

Feature	HTML	HTML5
Charset declaration	Complex	Simplified
Viewport support	Not standard	Fully supported
Mobile responsiveness	Limited	Built-in support
SEO support	Basic	Improved

New Features in HTML5

- Semantic elements: <header>, <footer>, <nav>, <section>, <article>
- Multimedia elements: <audio>, <video>
- Graphics: <canvas>, SVG
- New form inputs: email, date, number, range
- Client-side storage
- Offline web application support

Advantages of HTML5 Over HTML

- Eliminates need for third-party plugins
- Improves website performance
- Better accessibility and SEO
- Enhanced user experience
- Suitable for modern web applications

Meta Tags and Viewport in HTML5

Meta Tags in HTML5

What are Meta Tags?

Meta tags are special HTML tags used to provide **metadata** about a web page.

Metadata is information about the document that is **not displayed on the web page**, but is used by:

- Web browsers
- Search engines
- Other web services

Meta tags are placed **inside the <head> section** of an HTML document.

Commonly Used Meta Tags in HTML5

1. Character Encoding Meta Tag

```
<meta charset="UTF-8">
```

Explanation:

- Specifies the character encoding for the document
 - UTF-8 supports almost all characters and symbols
 - Prevents text display issues
-

2. Viewport Meta Tag

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Explanation:

- Controls how a web page is displayed on different devices
- Ensures proper rendering on mobile phones and tablets

Attributes:

- width=device-width – sets page width to device screen width

- initial-scale=1.0 – sets default zoom level
-

3. Description Meta Tag

```
<meta name="description" content="Introduction to HTML5 and web development">
```

Explanation:

- Provides a short summary of the web page
 - Used by search engines in search results
-

4. Keywords Meta Tag

```
<meta name="keywords" content="HTML, HTML5, Web Development">
```

Explanation:

- Specifies keywords related to the page content
 - Helps in basic search engine indexing
-

5. Author Meta Tag

```
<meta name="author" content="Web Technology Department">
```

Explanation:

- Specifies the author of the document
-

6. Refresh Meta Tag (Optional)

```
<meta http-equiv="refresh" content="5">
```

Explanation:

- Automatically refreshes the page after a given number of seconds
-

Importance of Viewport Meta Tag in HTML5

- Makes websites **responsive**
- Improves **mobile user experience**
- Prevents horizontal scrolling

- Ensures correct scaling on different screen sizes
 - Essential for **mobile-first design**
-

Complete HTML5 Head Section Example

```
<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <meta name="description" content="HTML5 basics and examples">

  <meta name="keywords" content="HTML5, Web Technology">

  <meta name="author" content="WT Lab">

  <title>Introduction to HTML5</title>

</head>
```

Embedding CSS in an HTML File

1. What is CSS?

CSS (Cascading Style Sheets) is used to control the presentation of HTML elements such as colors, fonts, layout, and spacing.

HTML defines structure, CSS defines style.

2. Why Do We Embed CSS?

- Improves appearance of web pages
- Maintains design consistency
- Separates content and design
- Allows reusability of styles

3. Ways to Embed CSS in HTML

3.1 Inline CSS

Definition:

CSS is written directly inside HTML tags using the style attribute.

Example:

```
<h1 style="color: blue;">Hello</h1>  
  
<h3 style="color: purple; font-family: Arial; text-decoration: underline;">  
    Multiple styles applied  
</h3>
```

Advantages:

- Quick styling
- Useful for testing

Disadvantages:

- Not reusable
- Hard to maintain

3.2 Internal (Embedded) CSS

Definition:

CSS is written inside the <style> tag in the <head> section.

Example:

```
<style>  
h1 { color: green; }  
</style>
```

Advantages:

- Cleaner than inline CSS

- Suitable for single-page sites

Disadvantages:

- Not reusable across pages

3.3 External CSS

Definition:

CSS is written in a separate .css file and linked using <link> tag.

Example:

```
<link rel="stylesheet" href="style.css">
```

Advantages:

- Best practice
- Easy maintenance
- Reusable

4. CSS Priority Order

1. Inline CSS
2. Internal CSS
3. External CSS

5. Best Practice

Use external CSS for real-world projects.