
1. Need and Benefits of HTML

HTML (HyperText Markup Language) is the standard language used to create web pages.

Need of HTML

- To structure web content (text, images, links, forms)
- To create webpages that browsers can understand
- Acts as the backbone of websites

Benefits

- Easy to learn and use
- Supported by all browsers
- Free and open standard
- Works with CSS and JavaScript
- Platform independent

2. Setup for HTML

You only need:

- A **text editor** (Notepad, **Visual Studio Code**)
- A **web browser** (Chrome, Edge, Firefox)

Steps

1. Create a file with .html extension
 2. Write HTML code
 3. Open the file in a browser
-

3. Browser

A browser:

- Reads HTML code
- Renders it visually
- Executes JavaScript
- Applies CSS styles

Examples: Chrome, Firefox, Edge, Safari

4. BOM and DOM

BOM (Browser Object Model)

- Represents the **browser window**
- Allows interaction with browser features
- Example: window, navigator, location

DOM (Document Object Model)

- Represents the **HTML document as a tree**
 - Allows JavaScript to manipulate HTML elements
 - Example: change text, styles, attributes
-

5. DOCTYPE

- Declares the **HTML version**
- Helps browser render the page correctly

HTML5 DOCTYPE

<!DOCTYPE html>

6. Character Encoding

- Defines how characters are displayed
- Prevents symbol/language issues

Common encoding

<meta charset="UTF-8">

7. <script> Tag

- Used to add JavaScript

<script src="app.js"> </script>

Can be:

- Inline
 - External
-

8. <link> Tag

- Used to connect external resources (mostly CSS)

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

9. HTML5 Document Structure

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>My Page</title>

</head>

```
<body>
  <h1>Hello World</h1>
</body>
</html>
```

10. Comments

Used to explain code (not shown in browser)

```
<!-- This is a comment -->
```

11. Visual Studio Code Features

Visual Studio Code features:

- Syntax highlighting
 - Auto-complete (IntelliSense)
 - Extensions
 - Live Server
 - Git integration
 - Lightweight and fast
-

12. Google Chrome Developer Tools

Google Chrome DevTools

- Inspect HTML/CSS
 - Debug JavaScript
 - View console errors
 - Analyze performance
 - Edit live DOM/CSS
-

13. Inspect Document

- Right click → **Inspect**
 - Opens DevTools
 - Helps debug and understand page structure
-

14. Formatting Tags

Examples:

- ** – bold**
 - *<i> – italic*
 - <u> – underline
 - ** – important text**
 - ** – emphasized text**
-

15. Lists

Ordered List

```
<ol>  
  <li>HTML</li>  
</ol>
```

Unordered List

```
<ul>  
  <li>CSS</li>  
</ul>
```

16. Table

```
<table>
  <tr>
    <th>Name</th>
  </tr>
  <tr>
    <td>Alice</td>
  </tr>
</table>
```

17. Form & Input Tags

```
<form>
  <input type="text" placeholder="Enter name">
  <input type="submit">
</form>
```

Common input types:

- text
 - password
 - email
 - radio
 - checkbox
 - submit
-

18. Images

```

```

19. Styles

Inline CSS

`<p style="color:red;">Text</p>`

Internal CSS

```
<style>
p { color: blue; }
</style>
```

External CSS

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

20. Placeholder

- Temporary hint inside input fields

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

21. Inline and Block Elements

Block Elements

- Start on new line
 - Take full width
- Examples: `<div>`, `<p>`, `<h1>`

Inline Elements

- Do not start new line
- Examples: ``, `<a>`, ``
-

22. id vs class Attributes

id

- Unique
- Used once

```
<div id="header"> </div>
```

class

- Reusable

```
<div class="box"> </div>
```

Difference

id	class
Unique	Multiple
#id	.class

1. Basic HTML + DOCTYPE + Character Encoding

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Basic HTML Page</title>
</head>
<body>
  <h1>Hello World</h1>
  <p>This is my first HTML page.</p>
</body>
</html>
```

2. Browser + Setup Example

Save this as index.html and **open it in Chrome / any browser**

```
<!DOCTYPE html>
<html>
<body>
  <h2>Open me in a browser</h2>
</body>
</html>
```

3. BOM Example

```
<!DOCTYPE html>
<html>
<body>
  <button onclick="showAlert()">Click Me</button>

  <script>
    function showAlert() {
      alert("Hello from BOM!");
      console.log(window.innerWidth);
    }
  </script>
</body>
</html>
```

4. DOM Example

```
<!DOCTYPE html>
<html>
<body>
  <p id="text">Original Text</p>
```

```
<button onclick="changeText()">Change</button>

<script>
  function changeText() {
    document.getElementById("text").innerHTML = "Text Changed!";
  }
</script>
</body>
</html>
```

5. <script> Tag (External JS)

index.html

```
<!DOCTYPE html>
<html>
<body>
  <h2>External JavaScript</h2>
  <button onclick="sayHello()">Click</button>

  <script src="app.js"></script>
</body>
</html>
```

app.js

```
function sayHello() {
  alert("Hello from external JS file");
}
```

6. <link> Tag (External CSS)

index.html

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Styled Page</h1>
</body>
</html>
```

style.css

```
h1 {
  color: blue;
}
```

7. HTML5 Semantic Document

```
<!DOCTYPE html>
<html>
<body>

  <header>
    <h1>My Website</h1>
  </header>

  <nav>
    <a href="#">Home</a> |
    <a href="#">About</a>
  </nav>
```

```
<section>
  <article>
    <h2>Article Title</h2>
    <p>Article content here.</p>
  </article>
</section>

<footer>
  <p>© 2026</p>
</footer>

</body>
</html>
```

8. Comments

```
<!DOCTYPE html>
<html>
<body>
  <!-- This is a comment -->
  <p>Hello</p>
</body>
</html>
```

9. Formatting Tags

```
<!DOCTYPE html>
<html>
<body>
  <b>Bold</b> <br>
  <i>Italic</i> <br>
  <u>Underline</u> <br>
  <strong>Important</strong> <br>
```

```
<em>Emphasized</em>
</body>
</html>
```

10. Lists

```
<!DOCTYPE html>
<html>
<body>

<ul>
  <li>HTML</li>
  <li>CSS</li>
</ul>

<ol>
  <li>Login</li>
  <li>Dashboard</li>
</ol>

</body>
</html>
```

11. Table

```
<!DOCTYPE html>
<html>
<body>

<table border="1">
  <tr>
    <th>Name</th>
    <th>Marks</th>
```

```
</tr>
<tr>
  <td>Alice</td>
  <td>90</td>
</tr>
</table>
```

```
</body>
</html>
```

12. Form & Input Tags

```
<!DOCTYPE html>
<html>
<body>

<form>
  <input type="text" placeholder="Enter name"> <br> <br>
  <input type="email" placeholder="Enter email"> <br> <br>
  <input type="password" placeholder="Password"> <br> <br>
  <input type="submit">
</form>

</body>
</html>
```

13. Images

```
<!DOCTYPE html>
<html>
<body>
  
</body>
```

</html>

14. Styles (Inline, Internal, External)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p { color: green; }
  </style>
</head>
<body>

<p>Internal CSS</p>
<p style="color:red;">Inline CSS</p>

</body>
</html>
```

15. Placeholder

```
<!DOCTYPE html>
<html>
<body>
  <input type="text" placeholder="Search here">
</body>
</html>
```

16. Inline vs Block Elements

```
<!DOCTYPE html>
<html>
<body>
```

```
<div style="background:lightblue;">Block Element</div>
<span style="background:yellow;">Inline Element</span>
<span style="background:pink;">Inline Element</span>

</body>
</html>
```

17. id vs class

```
<!DOCTYPE html>
<html>
<head>
<style>
  #header { color: red; }
  .box { color: blue; }
</style>
</head>
<body>

<h1 id="header">Unique Header</h1>
<p class="box">Box 1</p>
<p class="box">Box 2</p>

</body>
</html>
```

18. Inspect Document (Practice Code)

```
<!DOCTYPE html>
<html>
<body>
  <button class="btn">Inspect Me</button>
```

```
</body>  
</html>
```

Right-click → **Inspect** → see HTML & CSS live.
