

# Web Programming : HTML

By

Narasimha Rao T

***Microsoft.Net FSD Trainer***

Professional Development Trainer

[tnrao.trainer@gmail.com](mailto:tnrao.trainer@gmail.com)

## 1. Structure of an HTML Document

- An HTML (HyperText Markup Language) document defines the **structure and layout** of a webpage. HTML is not a programming language — it's a **markup language** that uses tags to describe elements like text, images, and links.
- Every HTML file follows a standard structure so browsers (like Chrome, Edge, or Firefox) can interpret and display it correctly.

## Basic HTML Document Structure

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>My First Web Page</title>
  </head>
  <body>
    <h1>Welcome to My Page!</h1>
    <p>This is my first paragraph.</p>
  </body>
</html>
```

## 2. Common HTML Tags

HTML tags define the **building blocks** of your webpage. Tags are usually written in pairs: an **opening tag** (e.g., `<p>`) and a **closing tag** (e.g., `</p>`).

### Headings (`<h1>` – `<h6>`)

Headings define titles or subtitles.

```
<h1>Main Title</h1>  
<h2>Section Heading</h2>  
<h3>Subsection Heading</h3>
```

- `<h1>` is the most important, `<h6>` is the least.
- Use them **hierarchically** — one `<h1>` per page is ideal for accessibility and SEO.

## Paragraphs (<p>)

Used to display blocks of text.

```
<p>This is a paragraph of text.</p>
```

Paragraphs automatically include margin space for readability.

## Lists

Lists organize information clearly.

### Unordered List (bulleted):

```
<ul>  
  <li>Apples</li>  
  <li>Bananas</li>  
</ul>
```

### Ordered List (numbered):

```
<ol>  
  <li>First Step</li>  
  <li>Second Step</li>  
</ol>
```

## Images (<img>)

Displays pictures.

```

```

- `src` : path or URL to the image.
- `alt` : alternative text shown if the image cannot load — also read by screen readers.

Accessibility note: Always use descriptive `alt` text.

## Links (<a>)

Create clickable hyperlinks.

```
<a href="https://www.example.com">Visit Example</a>
```

- `href` : specifies the target URL or file path.
- You can also link within the same page using an **anchor**:

```
<a href="#section2">Go to Section 2</a>
```



### 3. Semantic HTML

- Semantic HTML introduces **meaningful tags** that describe the role of content rather than just its appearance.
- Before HTML5, developers used non-semantic tags like `<div>` everywhere, which made structure unclear. Semantic tags make the HTML **easier to read, improve accessibility, and help search engines understand content**.

## Common Semantic Elements

```
<header>
  <h1>My Blog</h1>
  <nav>
    <a href="#home">Home</a>
    <a href="#articles">Articles</a>
  </nav>
</header>

<main>
  <section id="articles">
    <h2>Recent Posts</h2>
    <article>
      <h3>Learning HTML</h3>
      <p>HTML gives structure to web content...</p>
    </article>
  </section>
</main>

<footer>
  <p>&copy; 2025 My Blog. All rights reserved.</p>
</footer>
```

## Explanation of Key Semantic Tags

- `<header>` — Top area of a page or section, often contains a title, logo, or navigation bar.
- `<nav>` — Specifically contains navigation links.
- `<main>` — Represents the main content unique to the page.
- `<section>` — Groups related content into logical sections (e.g., an “About” or “Services” section).
- `<article>` — Self-contained piece of content like a news article or blog post.
- `<footer>` — Bottom section, usually for contact info, copyright, or site links.
- `<aside>` — For sidebars, advertisements, or additional context.

## Why Use Semantic HTML?

1. **Accessibility:** Screen readers can navigate the page more easily.
2. **SEO Benefits:** Search engines can better index your content.
3. **Maintainability:** The structure is clearer for other developers.

**Visit to know more details :** <https://www.youtube.com/watch?v=20SHvU2PKsM>

## 4. Forms in HTML

Forms allow users to **input data** that can be sent to a server (e.g., for login, feedback, or search).

### Basic Structure

```
<form action="/submit" method="post">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <label for="country">Country:</label>
  <select id="country" name="country">
    <option value="">--Select--</option>
    <option value="us">USA</option>
    <option value="uk">UK</option>
  </select>
```

```
<label for="message">Message:</label>  
<textarea id="message" name="message" rows="4"></textarea>  
  
<button type="submit">Send</button>  
</form>
```

## Explanation of Attributes

- **action** — URL to which the form data will be sent.
- **method** — How the data is sent:
  - **get** → appends data to the URL (useful for searches).
  - **post** → sends data securely in the body (used for sensitive data).
- **name** — Used to identify form fields when data is submitted.
- **id** and **for** — Connect `<label>` s to inputs for accessibility.
- **required** — Makes a field mandatory.
- **type** — Defines input kind ( `text` , `email` , `password` , `checkbox` , etc.).

## Common Form Elements

- `<input>` → single-line text fields, checkboxes, radio buttons, etc.
- `<select>` → dropdown lists.
- `<textarea>` → multi-line text input.
- `<button>` → triggers form submission or JavaScript actions.



## Bonus: Best Practices

- Always use **semantic** elements where possible.
- Keep indentation consistent — it improves readability.
- Validate HTML using the **W3C Markup Validation Service**.
- Use **meaningful attribute names** and **accessible labels**.
- Always provide **alt text** for images.

## Self-Check Questions:

1. What is the purpose of using semantic tags in HTML?
2. How does `article` differ from `section` ?
3. What type of content is usually placed inside `<header>` and `<footer>`?
4. Why is semantic HTML beneficial for accessibility and SEO?
5. How can you make a piece of text link to another webpage?

## Self-Check Questions:

6. What is the difference between the `get` and `post` methods?
7. What does the `required` attribute do?
8. How does the browser use the `name` attribute when submitting a form?
9. What tag should you use for a multi-line comment box?
10. Why do we need `autofocus` attribute?

## Q & A

---

Narasimha Rao T

[tnrao.trainer@gmail.com](mailto:tnrao.trainer@gmail.com)

