

# HannaCode HTML Book

---

## Learn HTML from the Ground Up – A Complete Beginner's Guide

---

Author: Oladapo A.O (Hannacode Founder)

# THE HANNACODE HTML BOOK

LEARN HTML ○ THE GROUND UP -  
UP - COMPLETE BEGINNERS'  
GUIDE



OLADAPO A.O

(HANNACODE FOUNDER)

# Table of Contents

---

1. [Introduction to HTML](#)
2. [Basic HTML Tags](#)
3. [Text Formatting & Lists](#)
4. [Links & Images](#)
5. [Tables & Forms](#)
6. [Semantic HTML & Media](#)
7. [Best Practices & Mini Projects](#)
8. [Conclusion](#)

# Chapter 1: Introduction to HTML

---

## What is HTML?

---

HTML, or HyperText Markup Language, is the standard language used to create web pages. It provides the structure and content of a web page, defining elements like headings, paragraphs, images, and links. Think of HTML as the skeleton of a website; it gives the page its basic form and organization.

## Role of HTML in Web Development

---

HTML is fundamental to web development. While other technologies like CSS (for styling) and JavaScript (for interactivity) enhance a web page, HTML is always the starting point. It tells the browser what content to display and how to arrange it. Without HTML, there would be no web pages as we know them.

## HTML Document Structure

---

Every HTML document follows a basic structure. It begins with a `<!DOCTYPE html>` declaration, which tells the browser which version of HTML is being used. The entire document is enclosed within `<html>` tags. Inside the `<html>` tags, there are two main sections: `<head>` and `<body>`.

- The `<head>` section contains meta-information about the HTML document, such as its title (which appears in the browser tab), links to stylesheets, and scripts. This content is not directly visible on the web page itself.
- The `<body>` section contains all the visible content of the web page, including text, images, videos, and other elements that users interact with.

# Writing Your First HTML Page

---

Let's create a simple HTML page. Open a plain text editor (like Notepad on Windows, TextEdit on Mac, or VS Code) and type the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First HTML Page</title>
</head>
<body>
  <h1>Hello, HannaCode!</h1>
  <p>This is my very first HTML page. I'm learning HTML!</p>
</body>
</html>
```

Save this file as `index.html` (or any name ending with `.html`). Now, open this file in your web browser. You should see a heading that says

"Hello, HannaCode!" and a paragraph below it. Congratulations, you've just created your first HTML page!

# Chapter 2: Basic HTML Tags

---

HTML tags are the building blocks of web pages. They usually come in pairs, an opening tag and a closing tag, like `<p>` and `</p>`. The content goes between these tags.

## `<html>`, `<head>`, `<body>`

---

As seen in Chapter 1, these are the fundamental tags that define the overall structure of an HTML document:

- `<html>` : The root element that encloses all other HTML elements.
- `<head>` : Contains meta-information about the HTML document, not visible on the page itself.
- `<body>` : Contains all the visible content of the web page.

## Headings ( `<h1>` to `<h6>` )

---

Headings are used to define titles and subtitles on a web page. HTML provides six levels of headings, from `<h1>` (the most important) to `<h6>` (the least important). Search engines use headings to understand the structure and content of your web page.

```
<h1>This is a Main Heading</h1>  
<h2>This is a Subheading</h2>  
<h3>This is a Smaller Subheading</h3>
```

## Paragraphs, Line Breaks, Horizontal Rules

---

- **Paragraphs** ( `<p>` ): Used to define blocks of text. Browsers automatically add some space before and after a paragraph.

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

- **Line Breaks ( <br> )**: Used to insert a single line break. It's an empty tag, meaning it has no closing tag.

```
html <p>This is a line.<br>This is a new line.</p>
```

- **Horizontal Rules ( <hr> )**: Used to create a thematic break in content, often displayed as a horizontal line. It's also an empty tag.

```
html <p>Content above the line.</p> <hr> <p>Content below the line.</p>
```



# Chapter 3: Text Formatting & Lists

---

## Text Formatting

---

HTML provides several tags to format text, giving it different meanings or appearances:

- `<strong>` and `<b>`: Both make text bold. `<strong>` indicates strong importance, while `<b>` is for stylistic bolding without extra importance.
- `<em>` and `<i>`: Both italicize text. `<em>` indicates emphasis, while `<i>` is for stylistic italics (e.g., technical terms).
- `<mark>`: Highlights text.
- `<small>`: Makes text smaller.

```
<p>This is <strong>important</strong> text and this is <b>bold</b> text.</p>
<p>This is <em>emphasized</em> text and this is <i>italic</i> text.</p>
<p>This is <mark>highlighted</mark> text and this is <small>small</small> text.
</p>
```

## HTML Entities and Quotes

---

HTML entities are used to display reserved characters (like `<` or `>`) or special symbols (like `&copy;` for copyright) that are not easily typed on a keyboard.

- `&lt;` for `<` (less than)
- `&gt;` for `>` (greater than)
- `&amp;` for `&` (ampersand)
- `&copy;` for `©` (copyright symbol)

For quotes, you can use the `<q>` tag for short inline quotations or `<blockquote>` for longer, block-level quotations.

```
<p>The symbol for less than is &lt;.</p>
<blockquote>
  <p>"The only way to do great work is to love what you do." - Steve Jobs</p>
</blockquote>
```

## Lists

---

Lists are essential for organizing information on a web page.

### Ordered Lists ( `<ol>` )

Ordered lists are used for items that need to be in a specific sequence, like steps in a recipe. Items are typically numbered.

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

### Unordered Lists ( `<ul>` )

Unordered lists are used for items where the order doesn't matter, like a list of ingredients. Items are typically bulleted.

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Cherry</li>
</ul>
```

### Nested Lists

You can nest lists within each other to create hierarchical structures.

```
<ul>
  <li>Fruits
    <ul>
      <li>Apple</li>
      <li>Banana</li>
    </ul>
  </li>
  <li>Vegetables
    <ul>
      <li>Carrot</li>
      <li>Broccoli</li>
    </ul>
  </li>
</ul>
```

# Chapter 4: Links & Images

---

## Links and Anchors

---

Links are what make the web interconnected. The `<a>` (anchor) tag is used to create hyperlinks.

- **href attribute:** Specifies the destination URL.
- **target attribute:** Specifies where to open the linked document ( `_blank` for a new tab, `_self` for the same tab).

```
<p><a href="https://www.hannacode.com" target="_blank">Visit HannaCode</a></p>  
<p><a href="#top-of-page">Go to Top</a></p>
```

## Images

---

Images are crucial for making web pages visually appealing. The `<img>` tag is used to embed images.

- **src attribute:** Specifies the path to the image file.
- **alt attribute:** Provides alternative text for the image, important for accessibility and when the image cannot be displayed.

# Chapter 5: Tables & Forms

---

## Tables

---

Tables are used to display data in a tabular format (rows and columns). The main tags are:

- `<table>` : Defines the table.
- `<tr>` : Defines a table row.
- `<td>` : Defines a table data cell.
- `<th>` : Defines a table header cell.

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>30</td>
  </tr>
</table>
```

**Adding Captions and Borders:** Use `<caption>` for a table title and CSS for borders (e.g., `<table style="border: 1px solid black;">`).

## Forms and Inputs

---

Forms are used to collect user input. The main elements are:

- `<form>` : Defines the form.
- `<input>` : Defines various input fields (text, email, password, etc.).
- `<label>` : Provides a label for an input field.
- `<textarea>` : Defines a multi-line text input area.

- `<button>`: Defines a clickable button.

**Form Attributes:** `action` (where to send the form data) and `method` (how to send the data, e.g., `GET` or `POST`).

**Input Types:** Common types include `text`, `email`, `password`, `number`, `checkbox`, `radio`, `submit`, `date`, etc.

```
<form action="/submit_form" method="post">
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Submit">
</form>
```

# Chapter 6: Semantic HTML, Media & Best Practices

---

## Semantic HTML

---

Semantic HTML elements clearly describe their meaning to both the browser and the developer. They improve accessibility and SEO.

- `<header>` : Introductory content or a set of navigational links.
- `<nav>` : Navigation links.
- `<main>` : The dominant content of the `<body>` .
- `<article>` : Independent, self-contained content.
- `<section>` : A standalone section within a document.
- `<aside>` : Content tangentially related to the content around it.
- `<footer>` : Footer for its nearest sectioning content or the root element.

## Media and Embeds

---

HTML5 introduced tags for embedding audio and video directly.

- `<audio>` : Embeds sound content.
- `<video>` : Embeds video content.

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

**Embedding YouTube Videos:** Use an `<iframe>` tag with the YouTube embed URL.

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/yourvideoid" frameborder="0"
allowfullscreen></iframe>
```

## Best Practices

---

- **Clean and Readable Code:** Use consistent indentation (2 or 4 spaces) and logical structuring.
- **Comments:** Use `<!-- This is a comment -->` to explain complex parts of your code.
- **Accessibility:** Use semantic HTML, provide `alt` text for images, and ensure proper contrast for text.

## Mini Projects (Concepts)

---

To practice, try building these:

- **Simple Personal Homepage:** Use headings, paragraphs, links, and an image.
- **Contact Form:** Implement a basic form with text inputs, email, and a submit button.
- **Resume Layout:** Structure a simple resume using headings, lists, and paragraphs.



# Conclusion

---

## Summary

---

Congratulations! You've completed the foundational journey into HTML. You've learned about the basic structure of an HTML document, how to use essential tags for text, links, and images, and how to organize content with lists and tables. We also touched upon forms for user input, the importance of semantic HTML for better structure and accessibility, and how to embed media. Remember, HTML is the bedrock of the web, and mastering it opens the door to creating engaging and functional websites.

## More Resources

---

To continue your learning, explore these resources:

- **MDN Web Docs (Mozilla Developer Network):** An excellent and comprehensive resource for web technologies. <https://developer.mozilla.org/en-US/docs/Web/HTML>
- **W3Schools HTML Tutorial:** A popular and beginner-friendly tutorial site. <https://www.w3schools.com/html/>
- **freeCodeCamp:** Offers interactive coding lessons and projects. <https://www.freecodecamp.org/>

## HannaCode Community

---

Join the HannaCode community to connect with other learners, ask questions, and share your projects. Stay updated with the latest in web development and continue your coding journey with us!

- **HannaCode Website:** <https://www.hannacode.com>
- **HannaCode Community:** <https://www.hannacode.com>