**HTML Page Structure**

An HTML document is structured using a set of nested tags. Each tag is enclosed within <…> angle brackets and acts as a container for content or other HTML tags. Let's take a look at a basic HTML document structure:

<!DOCTYPE html>

<html>

<head>

<title>Document</title>

</head>

<body>

<!-- content -->

</body>

</html>

Copy

This is how the title appears on an HTML page:

A typical HTML page looks like this:

<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</body>

</html>

Almost every website uses this structure. The main content goes inside the body tag. No worries if this looks complicated; let's break it down!

**Note:** These are the essential elements for a basic HTML document: **<!DOCTYPE html>, <html>, <head>, <title>, </head>, <body>, </body>, </html>**

**DOCTYPE Declaration**

<!DOCTYPE html>

Copy

The <!DOCTYPE html> declaration informs the web browser about the HTML version being used. The latest version is HTML5. But if this changes in the future (maybe 10 years down the line), the doctype declaration will be helpful!

**HTML Root Element**

<html>

Copy

The <html> tag is the root element that encapsulates all the content on the page.

</html>

Copy

The </html> tag marks the end of the <html> section.

**Head Section**

<head>

Copy

The <head> tag contains metadata and links to external resources like CSS and JavaScript files.

</head>

Copy

The </head> tag marks the end of the <head> section.

**Title Tag**

<title>Document</title>

Copy

The <title> tag sets the title of the web page, which is displayed in the browser's title bar or tab.

**Body Tag**

<body>

Copy

The <body> tag contains the visible content of the web page. This is where text, images, and other elements go.

</body>

Copy

The </body> tag marks the end of the visible content of the web page.

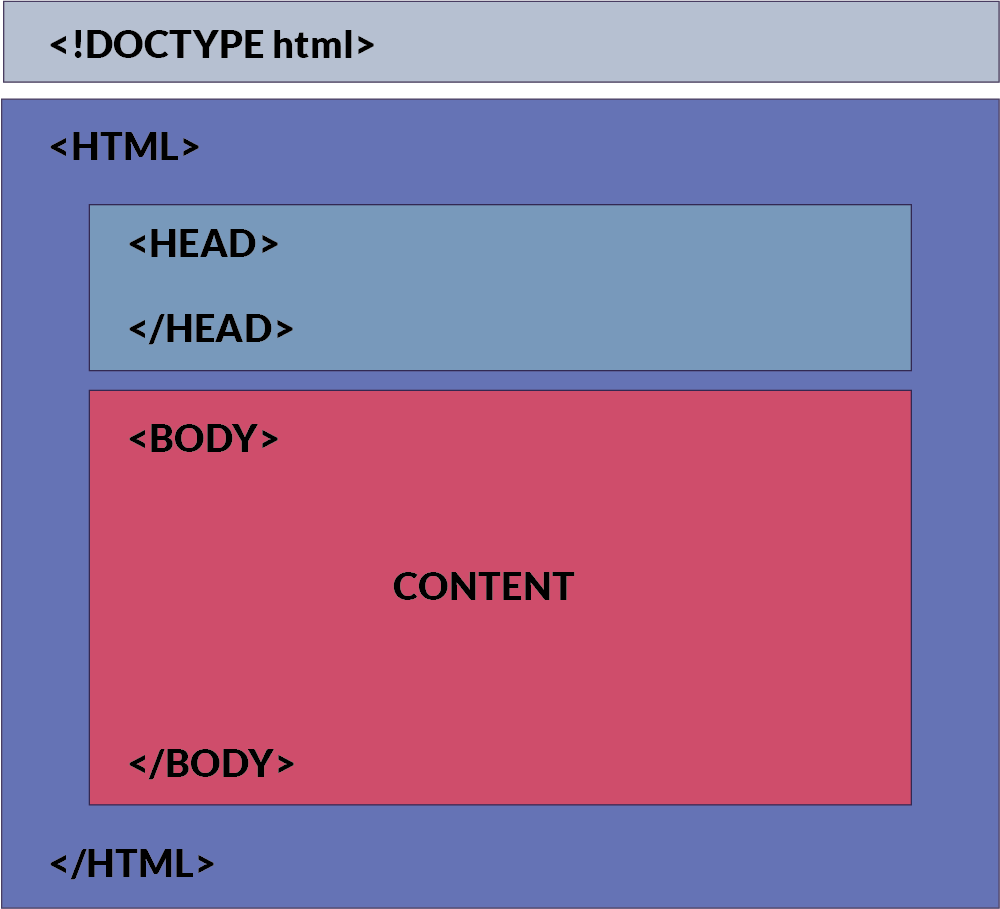
Every HTML page should include at least these essential elements to define the basic layout. In upcoming tutorials, we'll dive deeper into the fascinating world of HTML.

**Summary**

* The <!DOCTYPE html> tag specifies that the document is an HTML5 document.
* The <html lang="en"> tag defines the document to be in English.
* The <head> section contains metadata and the title of the webpage, which appears in the browser's title bar.
* The <body> section contains the content that will be displayed on the webpage.
* The h1 and p are two types of tags. We will learn about more tags in the later section

**Visualization of an HTML Document:**

The following image provides a visual representation of the HTML structure:



**How This Content Appears in a Web Browser:**

Consider this html code:

<!DOCTYPE html>

<html>

<head>

<title>Document</title>

</head>

<body>

<h1> This is a heading</h1>

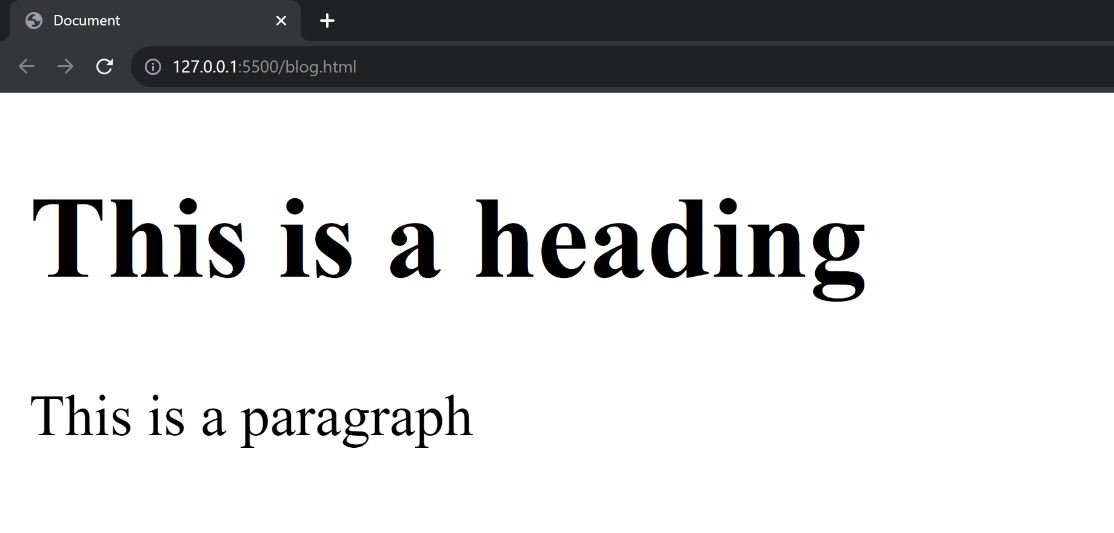
<p>This is a paragraph</p>

</body>

</html>

Copy

Below is an image showing how this HTML document will be rendered in a web browser:



In the browser, the title bar will display the content from the <head> section, specifically the <title> tag. The main area of the browser window (usually a white background) will display the content inside the <body> tag.

In the upcoming sections, we will learn about html tags and elements.

# HTML Tags

If you want to build a beautiful website, tags are essential elements that help you achieve that.

An HTML tag acts as a container for content or other HTML tags. Tags are words enclosed within < and > angle brackets.

They serve as keywords that instruct the web browser on how to format and display the content.

## Commonly used tags in HTML

Here are some commonly used tags in HTML. These are the only tags used 70% of the time

### Document Structure Tags

1. <!DOCTYPE html>: Specifies the document type.
2. <html>: Encloses the entire HTML document.
3. <head>: Contains meta-information and links to scripts and stylesheets.
4. <body>: Contains the content of the web page.

### Metadata Tags

1. <title>: Sets the title of the web page.
2. <meta>: Provides metadata such as character set, author, and viewport settings.
3. <link>: Links external resources like stylesheets.

### Text Formatting Tags

1. <p>: Paragraph.
2. <h1>, <h2>, <h3>, <h4>, <h5>, <h6>: Headings.
3. <strong>: Strong emphasis (typically bold).
4. <em>: Emphasis (typically italic).
5. <br>: Line break.
6. <hr>: Horizontal rule.

### List Tags

1. <ul>: Unordered list.
2. <ol>: Ordered list.
3. <li>: List item.

### Hyperlink and Media Tags

1. <a>: Anchor (used for links).
2. <img>: Image.
3. <audio>: Audio content.
4. <video>: Video content.

### Form Tags

1. <form>: Form.
2. <input>: Input field.
3. <textarea>: Text area.
4. <button>: Button.
5. <select>: Dropdown list.
6. <option>: Options within a <select> or <datalist>.

### Table Tags

1. <table>: Table.
2. <tr>: Table row.
3. <td>: Table data cell.
4. <th>: Table header cell.
5. <thead>: Table header group.
6. <tbody>: Table body group.
7. <tfoot>: Table footer group.

### Semantic Tags

1. <header>: Header section.
2. <footer>: Footer section.
3. <article>: Article.
4. <section>: Section.
5. <nav>: Navigation.
6. <aside>: Sidebar content.

## Paired and Unpaired HTML Tags

Well, that was a really long list. Don't worry we will study these in detail. In HTML, tags can be broadly categorized into two types:

### 1. Paired Tags (Container Tags)

These are tags that come in pairs, consisting of an opening tag and a corresponding closing tag. The content goes between these two tags.

* **Opening Tag**: The opening tag starts with < and ends with >. For example, <p>.
* **Closing Tag**: The closing tag also starts with < but includes a forward slash / before the tag name, and ends with >. For example, </p>.

#### Examples:

* Paragraphs: <p>This is a paragraph.</p>
* Headings: <h1>This is a heading.</h1>

### 2. Unpaired Tags (Self-Closing Tags or Stand-Alone Tags)

These are tags that don't require a closing tag. They are self-contained, encapsulating all the information within a single tag.

* **Self-Closing Tag**: A self-closing tag starts with < and ends with /> (though the / is optional in HTML5). For example, <img /> or <br>.

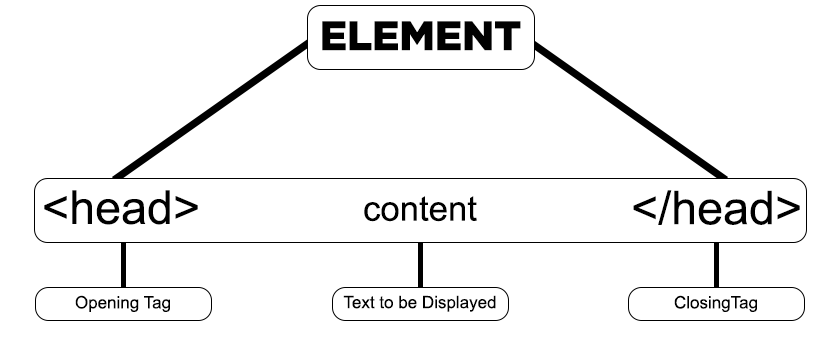
**Note:** Later if you happen to use react or a framework like Next.js, you will have to close the tag like this <br/> <hr/>. So it is better to cultivate the habit!

#### Examples of self-closing tags:

* Line Break: <br/>
* Horizontal Rule: <hr/>
* Image: <img src="image.jpg" alt="An example image"/>

## Pictorial Representation of Tags

The image below offers a visual representation of how tags are structured in HTML. As you can see, an element can contain other elements, which may also contain additional elements, forming a tree-like structure. This hierarchy can include self-closing tags as well as nested tags, as illustrated in the picture



# HTML Elements

Beginners often get confused between HTML elements, nested elements, and tags. Let's clarify the difference by understanding each one step-by-step.

## What is an HTML Element?

An HTML element is a complete set that consists of a start tag (or opening tag), content, and an end tag (or closing tag).

**HTML Element = Start Tag + Content + End Tag**

For example:

<h1>This is our first heading</h1>

Copy

In this example, <h1> is the start tag, "This is our first heading" is the content, and </h1> is the end tag. Together, they form an HTML element.

## What is a Nested HTML Element?

A nested HTML element is an HTML structure where one element is placed inside another element.

The enclosing element is often referred to as the "parent," while the enclosed element is called the "child."

**Nested HTML Element = One HTML Element Inside Another HTML Element**

For example:

<h1><b>This is our first heading</b></h1>

Copy

In this example, the <b> element (child) is nested inside the <h1> element (parent).

## What is an Empty HTML Element?

An empty HTML element is one that does not have a closing tag or content. These elements are also known as "void elements" or "self-closing elements."

**Empty HTML Element = Tags with No Content**

For example:

<br />

Copy

This is a **break tag**, which has no content and no closing tag. It's used to insert a line break within text or other inline elements. The <hr /> tag, used for horizontal rules, is another example of an empty or void element.

## HTML Tags vs. Elements

### HTML Tags

HTML tags are the markers that define the start and end of an element. They are wrapped in angle brackets, like <p> and </p>.

### HTML Elements

An HTML element includes an opening tag, content, and a closing tag, forming a complete set. For example, <p>This is a paragraph.</p>.

#### **Key Takeaways**

* Tags set boundaries; elements include tags plus content.
* Tags come in pairs (most of the time), whereas elements may include nested elements.
* Self-closing or void elements like <br /> are still considered elements, even though they don't have a closing tag or content.
* Elements can be "parent" or "child" when nested, but tags cannot.

**HTML Attributes**

HTML attributes are used to define the characteristics of an HTML element. They are placed within the element's opening tag and consist of two parts: the **name** and the **value**.

* **Name**: Specifies the property for that element.
* **Value**: Sets the value of that property for the element.

**Types of HTML Attributes**

There are three main types of HTML attributes:

1. **Core Attributes**: These are basic attributes that can be applied to most HTML elements. Examples include id, class, and style.
2. **Internationalization Attributes**: These attributes help adapt the document to different languages and regions. Examples include lang and dir.
3. **Generic Attributes**: These attributes provide additional information about the element but don't necessarily affect its appearance or behavior. Examples include data-\* attributes for storing custom data private to the page or application.

Core attributes are some of the most widely used attributes in HTML. There are four main types:

* id
* class
* title
* style

**ID Attribute**

The ID attribute is used to assign a unique identifier to an HTML element. Each element with an ID has its own unique identity, similar to how each individual has a unique identity. Multiple elements cannot have the same ID.

Example:

<p id="html">This is an HTML tutorial</p>

<p id="python">This is a Python tutorial</p>

Copy

In this example, the ID attribute helps to distinguish between two paragraphs by having different values: "html" and "python".

**Class Attribute**

The class attribute is used to associate an HTML element with a particular class, typically for styling or JavaScript manipulation. Unlike the ID attribute, the class attribute is not unique, and multiple elements can share the same class.

**Title Attribute**

The title attribute provides additional information about an element and is often displayed as a tooltip when the mouse hovers over it.

Example:

<h4 title="hello, motto">Title attribute</h4>

Copy

Output:

**Style Attribute**

The style attribute allows for inline styling of HTML elements. It is used in conjunction with CSS properties to directly style individual elements within the HTML code.

**Case Sensitivity**

The HTML standard is flexible about the case of attribute names, allowing them to be written in either uppercase or lowercase, such as "title" or "TITLE." However, for best practices and compatibility with stricter document types like XHTML, the W3C recommends using lowercase attributes.

# HTML Comments

Comments in HTML are like little notes you leave in your code for yourself or other people. These notes help make the code easier to understand but don't show up on the actual website. The web browser just skips over them!

### Key Points About HTML Comments

* Comments are ignored by web browsers.
* They aid in code readability and documentation.
* HTML comments are denoted by <!-- content -->.
* The shortcut key for commenting out code is Ctrl + /.
* HTML supports both single-line and multi-line comments.

### Types of Comments in HTML

HTML primarily supports two types of comments:

#### Single-line Comments

Single-line comments are contained within one line. They are useful for short annotations.

Example:

<!-- This is a single-line comment -->

Copy

As you can see in the video below, the text inside the comment is not rendered

#### Multi-line Comments

Multi-line comments span across multiple lines, making them ideal for detailed explanations or temporarily disabling blocks of code.

Example:

<!--

This is a multi-line comment.

It spans multiple lines.

-->