**HTML Introduction**

Today, I'm writing this tutorial to create a resource that will help you learn HTML in less than 30 days. Here's a recommended timeline for learning HTML, based on your educational background:

* High School students and younger: Around 25 days
* Those beyond High School: Around 20 days
* College students and above: Around 10-20 days

You may be wondering why I'm discussing these timelines. It's important for me to set expectations before you start your journey of learning html with me.

My name is Harry (the same [CodeWithHarry](https://www.youtube.com/channel/UCeVMnSShP_Iviwkknt83cww" \t "_blank) guy from YouTube), and today I'm writing this comprehensive HTML tutorial for all of you. Let's jump right into HTML!

**What is HTML?**

HTML (HyperText Markup Language) was created by Tim Berners-Lee in 1991 as a standard for creating web pages. It's a markup language used to structure content on the web, defining elements like headings, paragraphs, links, and images. HTML forms the backbone of web content. In layman's terms, HTML is like the skeleton of a website. It's a set of instructions that tells a web browser how to display text, images, videos, and other elements on a webpage. Think of it as the building blocks that create the structure and look of a website, similar to how bricks and mortar are used to build a house.

In a nutshell:

* HTML is the language of the web, used to create websites.
* HTML defines the barebone structure or layout of web pages that we see on the Internet.
* HTML consists of a set of tags contained within an HTML document, and the associated files typically have either a "**.html**" or "**.htm**" extension.
* There are several versions of HTML, with HTML5 being the most recent version.

**Features of HTML**

* It is platform-independent. For example, Chrome displays the same pages identically across different operating systems such as Mac, Linux, and Windows.
* Images, videos, and audio can be added to a web page (For example - YouTube shows videos on their website)
* HTML is a markup language and not a programming language.
* It can be integrated with other languages like CSS, JavaScript, etc. to show interactive (or dynamic) web pages

**Why the Term HyperText & Markup Language?**

The term 'Hypertext Markup Language' is composed of two main words: 'hypertext' and 'markup language.' 'Hypertext' refers to the linking of text with other documents, while 'markup language' denotes a language that utilizes a specific set of tags.

Thus, HTML is the practice of displaying text, graphics, audio, video etc. in a certain way using special tags.

**Note:**Tags are meaningful texts enclosed in angle braces, like '<...>'. For example, the '<head>' tag. Each tag has a unique meaning and significance in building an HTML page, and it can influence the web page in various ways.

**Quick Exercise:**

Open a webpage of your choice, right-click on the browser, and select 'View Page Source,' and then you will see the HTML code for that page.

This is the code that the server sent to display the page you're currently viewing. In this tutorial, we'll learn how to write this type of code using HTML.

**A beautiful analogy to understand HTML, CSS, and JavaScript:**



In building a webpage, think of HTML, CSS, and JavaScript as different parts of a car. HTML is like the car's skeleton, forming the basic structure and frame. CSS adds the paint and finishing touches, making the car look appealing with color, style, and design. JavaScript is similar to the engine and mechanical parts, infusing the car with functionality, movement, and interactive features. Similarly, when developing a website, HTML lays out the structure, CSS enhances its visual appeal, and JavaScript provides interactivity and dynamic content

**History of HTML:**

* In 1989, Tim Berners-Lee established the World Wide Web (www), and in 1991, he created the first version of HTML.
* From 1995 to 1997, further work was done to develop and refine different versions of HTML.
* In 1999, a committee was organized that standardized HTML 4.0, a version still used by many today.
* The latest and most stable version of HTML is 5, also known as HTML5.

Feel free to read more history of HTML here on the [HTML Wikipedia page](https://en.wikipedia.org/wiki/HTML#:~:text=HTML%20version%20timeline) but I will move ahead and cover important aspects of HTML.

In the next tutorial, we'll continue our journey and understand how websites work

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Re](https://www.codewithharry.com/refund/)

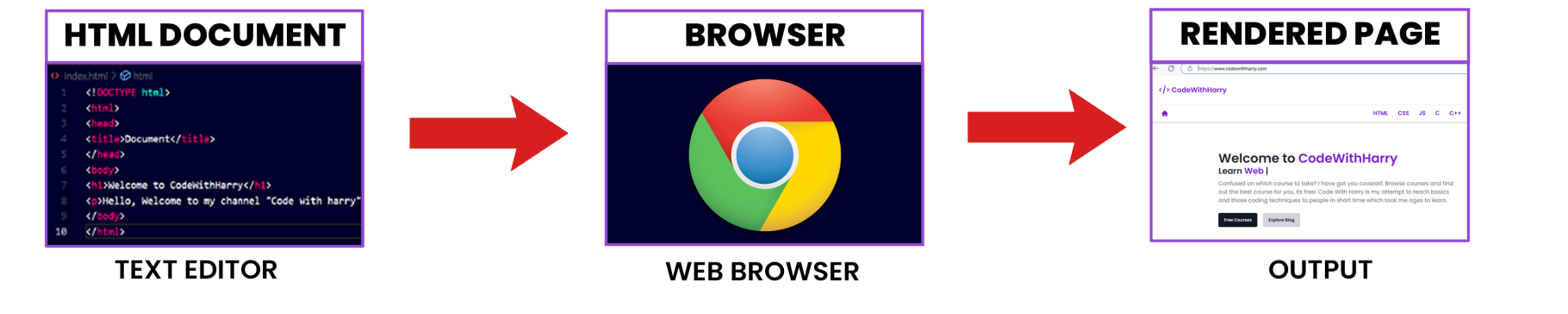
**HTML Working**

You must have heard of frontend and backend. Frontend refers to the visible part of a website or app that users interact with, like the tables, images, and buttons. It's built using languages like HTML, CSS, and JavaScript. The backend, on the other hand, handles behind-the-scenes operations like storing and processing data when users interact with the frontend. It uses languages like Python, Ruby, and Java. In essence, frontend is what users see, while backend manages the functionality.

**How do websites work?**

When we want to access any information on the internet, we search for it using a web browser. The web browser retrieves the content from web servers, where it is stored in the form of HTML documents.

An HTML document is created by writing code with specific tags in a code editor of your choice. The document is then saved with the**'.html'**extension. Once saved, the browser interprets the HTML document, reads it, and renders the web page.



The text editor has the HTML code of a website. This website can now be viewed in a beautifully rendered format using a computer program known as a web browser.

**What is a Web Browser?**

A web browser is a program that understands HTML tags and renders them in a human-readable format that is easily viewable by people visiting the website. Developers write code in HTML because it's a straightforward way to instruct the web browser on what to display. In the next section, I'll show you how to set up VS Code for writing your own HTML code and rendering it in a web browser.

**What is an HTML Document?**

An HTML document is a text document saved with the '.html' or '.htm' extension, containing text and specific tags enclosed in '< >'. These tags provide the necessary instructions for configuring the web page. The tags themselves are standardized and fixed. The structure of an HTML document will be explained later in this HTML tutorial.

**What is a Rendered Page:**

The rendered page is the output screen of our HTML Document which is the page displayed on the browser.

**How does a basic website work?**

* Web Browser(client) requests websites like [www.codewithharry.com](https://www.codewithharry.com/) from the web server.
* Web server in return sends HTML, CSS, and JavaScript files to it.
* HTML, CSS, and JavaScript files are parsed by a web browser which is responsible for showing you a beautiful website.

**How does a browser work?**

A web browser plays a crucial role in parsing and rendering HTML code to the client. A web browser is a complex program that performs many tasks behind the scenes. Here's a summary of how it works:

* A browser is an application that reads HTML documents and displays them as web pages. Browsers can't access the content directly from where it's stored; this is where servers come into play.
* A server acts as an intermediary, listening to browser requests and fulfilling them by delivering the HTML document to the browser.
* Web browsers perform two main tasks: parsing and rendering.
* During the parsing stage, the browser receives raw bytes, which are converted into characters. These characters are then converted into tokens, which in turn are transformed into nodes. These nodes are organized into a tree-like data structure known as the DOM (Document Object Model).
* Once the DOM tree is constructed, the browser moves on to the rendering stage. At this point, each node in the DOM tree is rendered and displayed on the screen.

Dont worry about how browser exactly works just yet. Rather focus on learning HTML. In the next tuotrial we will install VS Code and some extensions for writing our HTML code.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)

**HTML Installation**

Let's get our hands dirty and start preparing to write some code. In this tutorial, we will install VS Code and some related extensions for faster and more efficient HTML development.

**What are the prerequisites to learning HTML?**

I can safely say that there are **no prerequisites to learning HTML**. HTML is the language of the web and is often the first step that web developers take in learning to code.

**Tools needed to make an HTML page:**

1) **HTML Editor:** It's a straightforward tool where every piece of HTML content must be written. You can use any text editor of your choice. In this tutorial, we're using Visual Studio Code because it's lightweight and open-source.

Popular editors for HTML development include text editors like [Notepad++](https://notepad-plus-plus.org/) and [TextEdit](https://support.apple.com/en-in/guide/textedit/welcome/mac" \t "_blank), code editors such as [Sublime Text](https://www.sublimetext.com/) and [Visual Studio Code](https://code.visualstudio.com/), and full-fledged IDEs like [WebStorm](https://www.jetbrains.com/webstorm/" \o "jetbrains webstorm" \t "_blank) and [Eclipse](https://www.eclipse.org/downloads/). Online platforms like [CodePen](https://codepen.io/" \o "codepen" \t "_blank) and [JSFiddle](https://jsfiddle.net/" \o "jsfiddle" \t "_blank) are also commonly used for quick HTML editing and testing.

**Note: You can write HTML even in a Notepad. Text editors like VS code make these things easier.**

2) **Browser:** HTML tags are not displayed by browsers; instead, they are read and interpreted to render the web page. In a web browser, HTML structures are displayed in a styled and visually appealing form. In this tutorial, we are using Google Chrome. Other commonly used browsers include Chromium, Firefox, Safari on Mac, and Microsoft Edge.

**Installation & Setup of Visual Studio Code for HTML:**

We will install and set up HTML to optimize its utility for creating web pages. Additionally, we'll install extensions in Visual Studio Code to enhance its functionality. If you're unsure about which editor to use, you can confidently start with Visual Studio Code. You won't regret it; it's one of the best free code editors available in the market.

* Search for "Visual Studio Code download" on Google
* Download [Visual Studio Code](https://code.visualstudio.com/download) for your Operating System. I am using Windows so I will install it for Windows

Here is a quick video showing VS Code Installation:

**Live Server Extension**

In addition, we'll be installing the Live Server extension in our Visual Studio Code editor to view live reload pages.

The Live Server extension is used to launch a local development server with a live reload feature for HTML pages.

Here is a quick video showing the installation of the VS Code 'Live Server' extension:

In the next lesson, we will start writing some code and review the document structure of HTML pages.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Refund](https://www.codewithharry.com/refund/)

# HTML Execution

## Your Journey to Creating Your First Website Begins Here!

Let's mark this as an important milestone: the creation of your first website! And what's a better way to start than with the traditional "Hello, World!"?

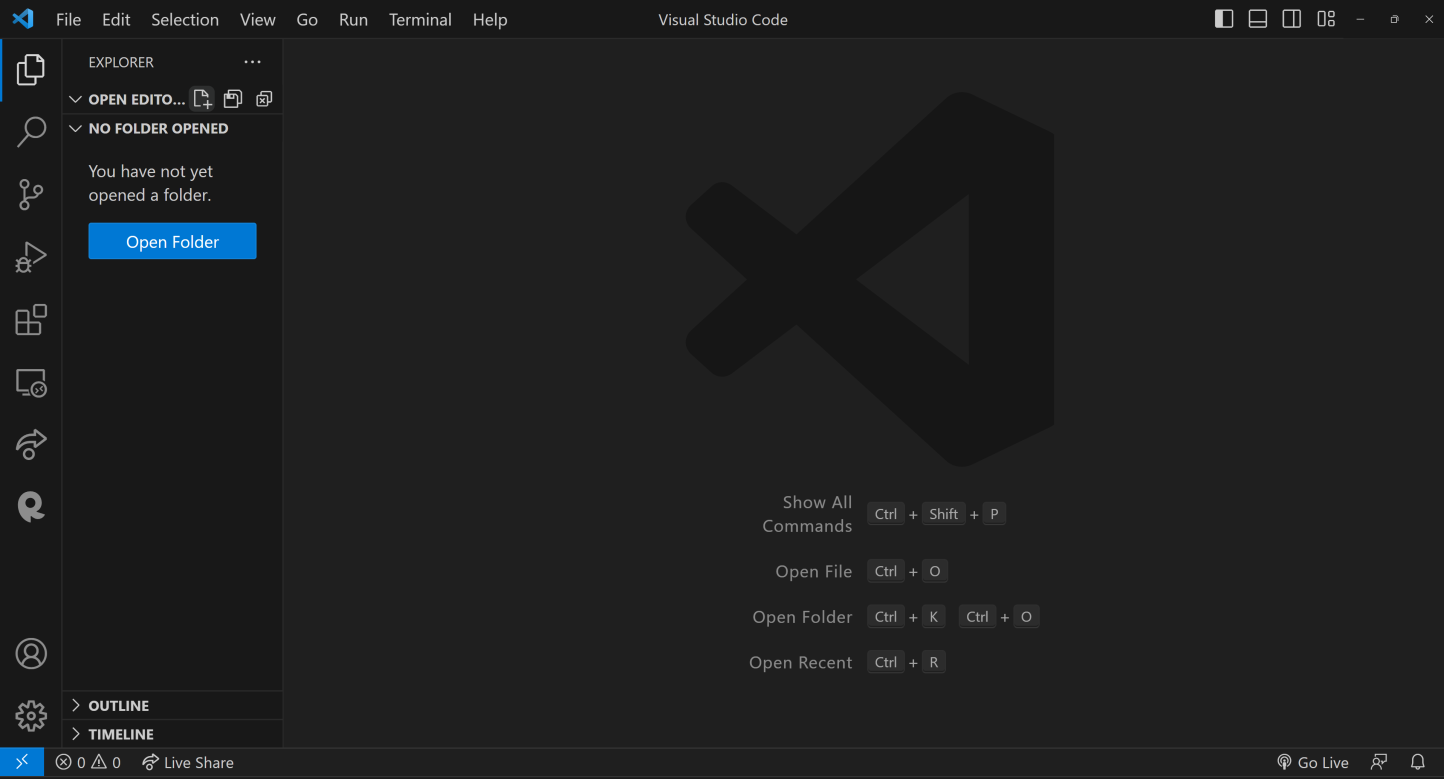
### Why "Hello, World!"?

In the programming world, "Hello, World!" is more than just a phrase. It's a tradition, an emotion, a simple program that teaches you the syntax and gets you started. And guess what? HTML is no different!

Our first website will display the text 'Hello World'

### Let's Get Started: Setting Up Your VS Code

If you haven't already set up your environment, let's begin by opening [Visual Studio Code (VS Code).](https://code.visualstudio.com/download)



#### Creating a New File

Click on "Open Folder" and open a folder somewhere on your computer. I am opening a folder named html-tutorial

Once VS Code is open, you'll want to create a new file:

1. Click on the "New File" icon in VS Code.
2. Type the filename as "index.html" and hit Enter.

#### Pasting the Code

Now that your file is ready, copy the following code and paste it into your "index.html" file.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

Hello World

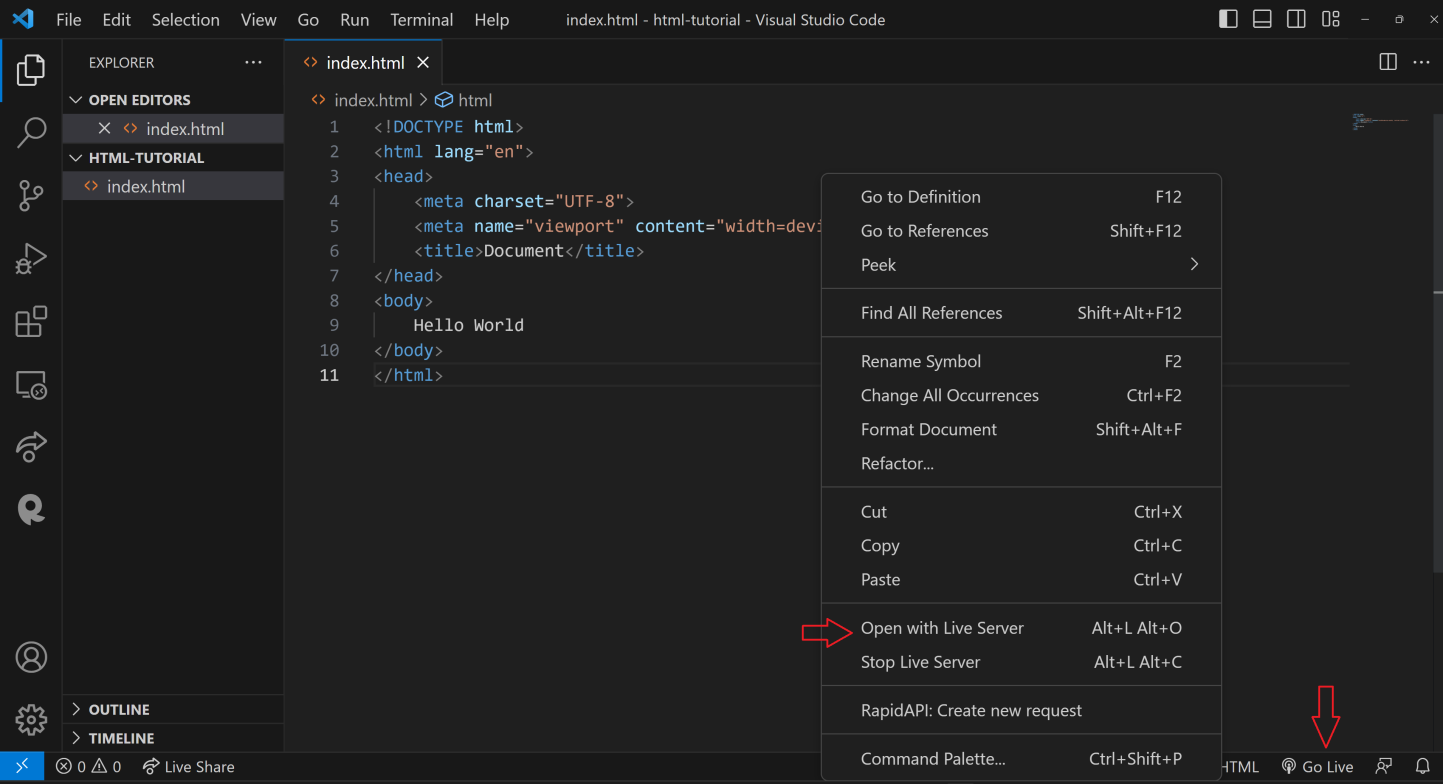
</body>

</html>

Copy

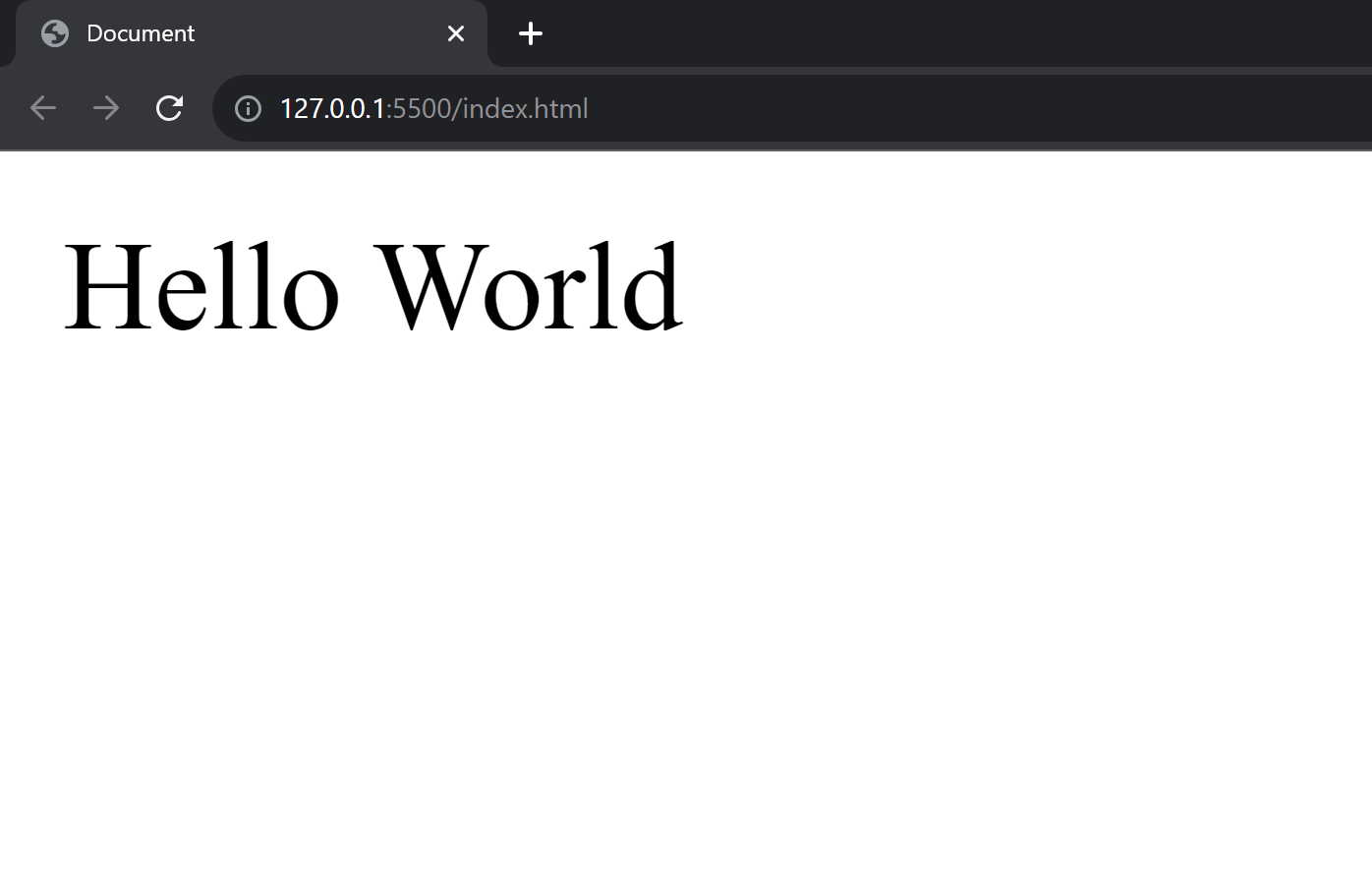
#### Going Live using the "live server" extension

To see your webpage in action, locate the "Go Live" icon at the bottom-right corner of your VS Code window and click it. If you don't see this icon, you probably haven't installed the Live Server extension, which we discussed in a previous tutorial.



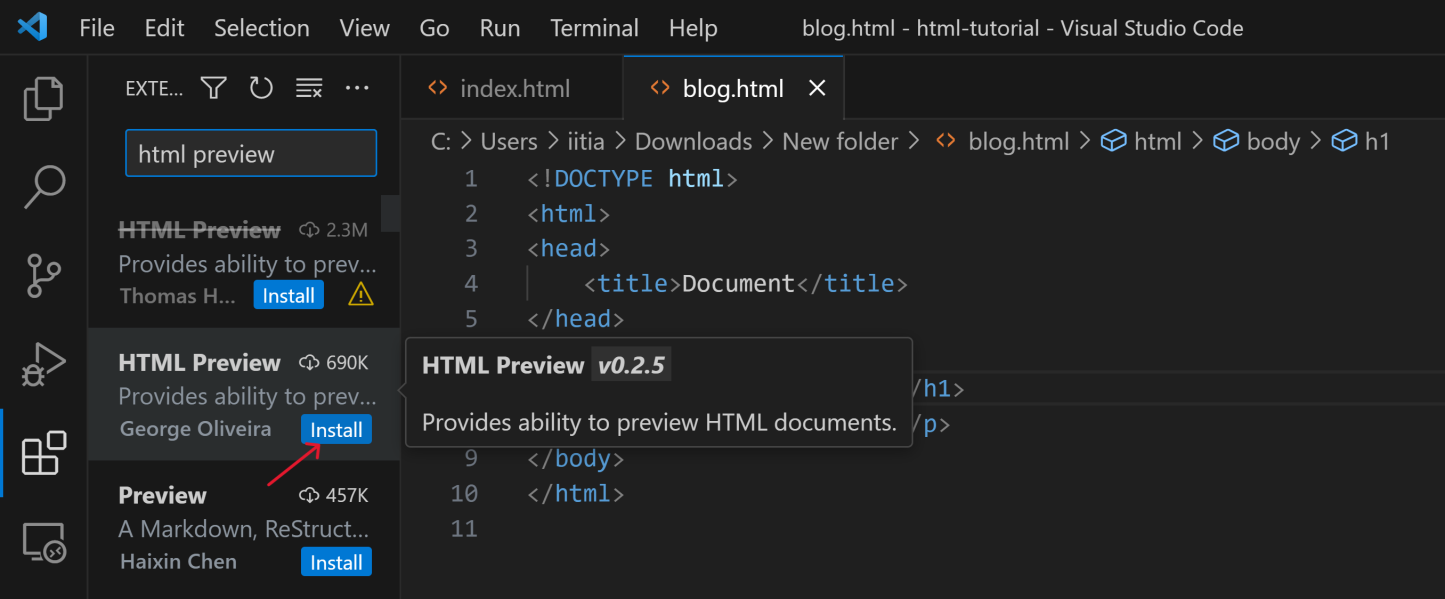
#### Your First Website is Live!

Congratulations! If you've followed along, you should now see your very first website displaying the classic "Hello, World!" message.

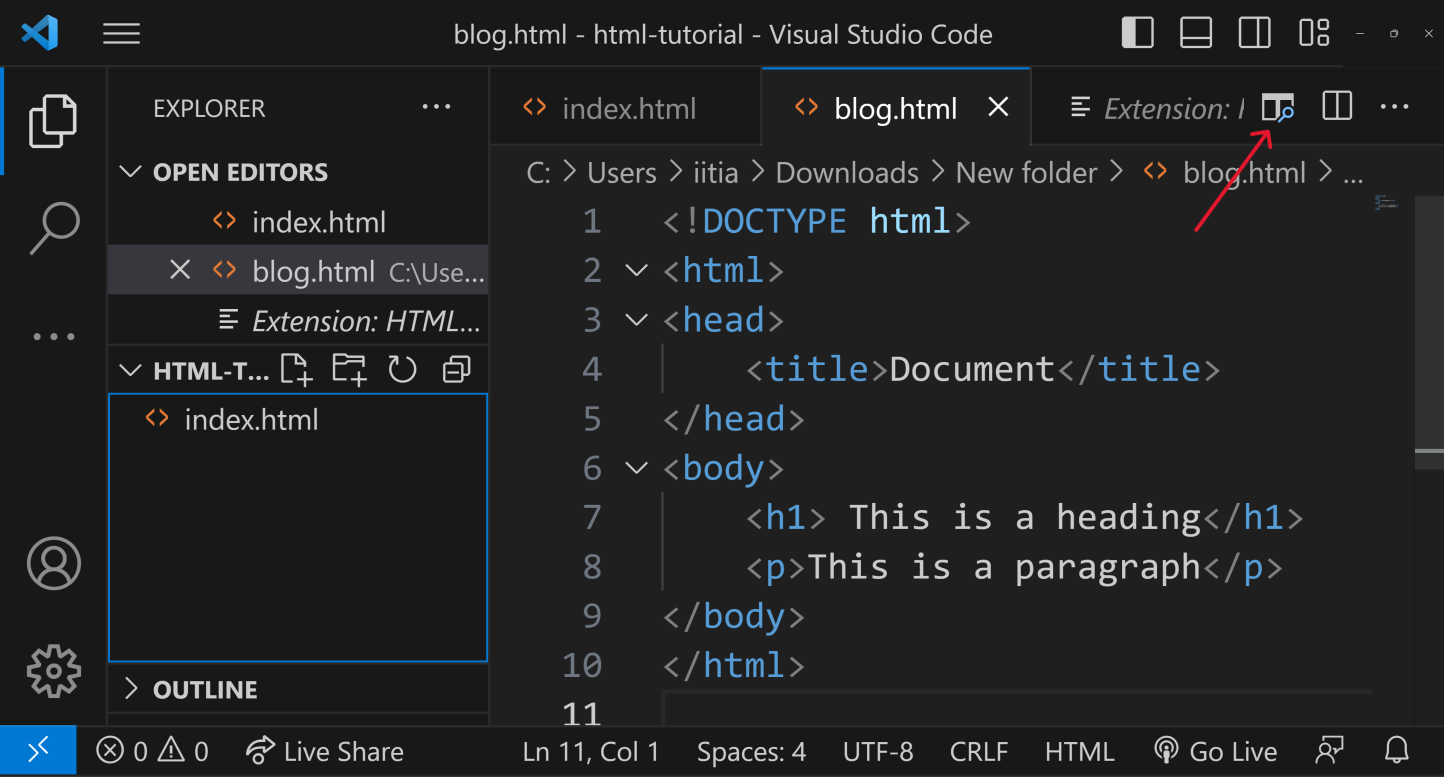


## Live Preview Extension

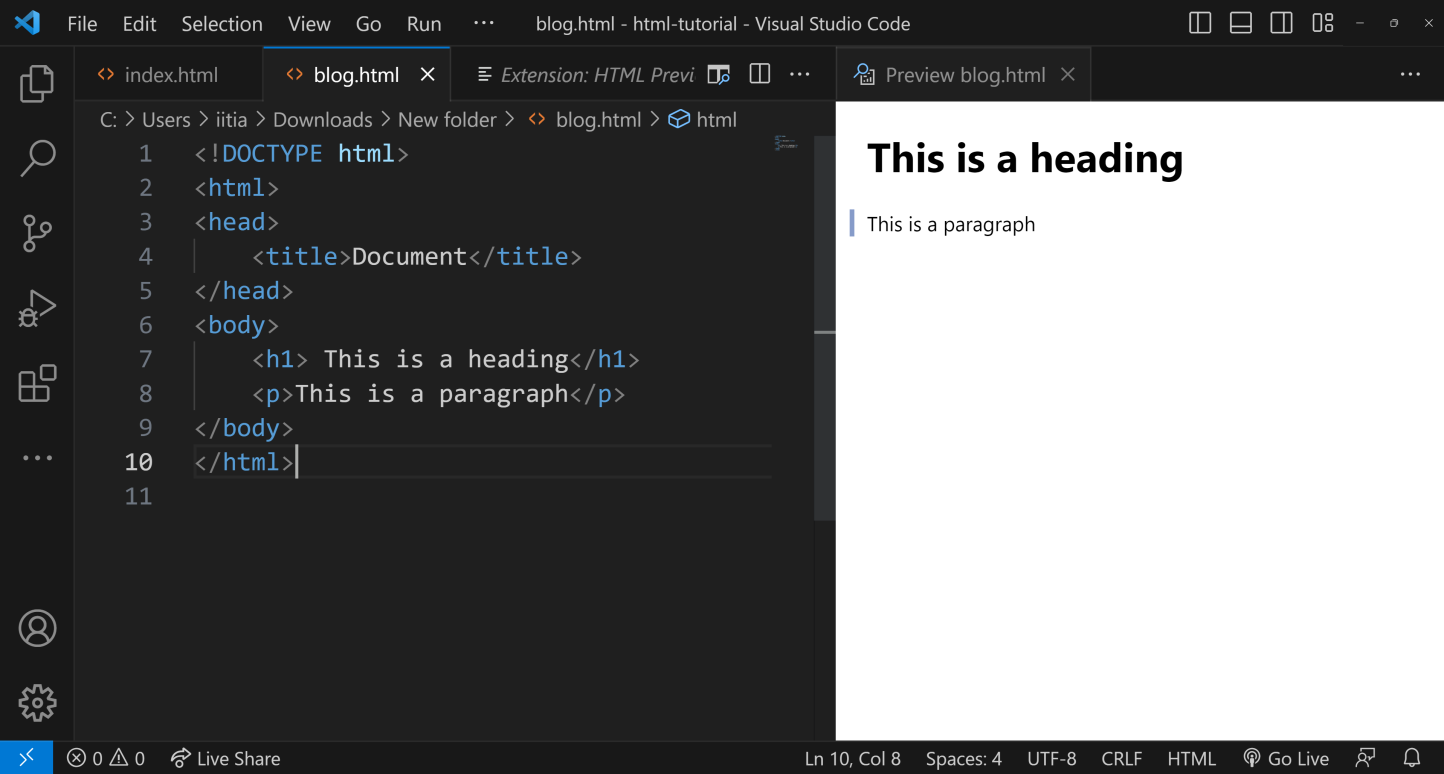
Another useful extension for working with HTML in VS Code is 'HTML Preview.' To install it, simply click on the extensions icon in VS Code and type 'HTML Preview' in the search bar. Install it!



Now, you will see a button within VS Code. Clicking on this button will allow you to preview your HTML right within the editor.



Once you click the button, you'll see a live preview of your HTML directly within VS Code.



You don't even need a browser to render plain HTML. This live preview feature in VS Code is perfect for this HTML tutorial, allowing you to build and preview your entire HTML website without ever leaving the editor.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Refund](https://www.codewithharry.com/refund/)

**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.

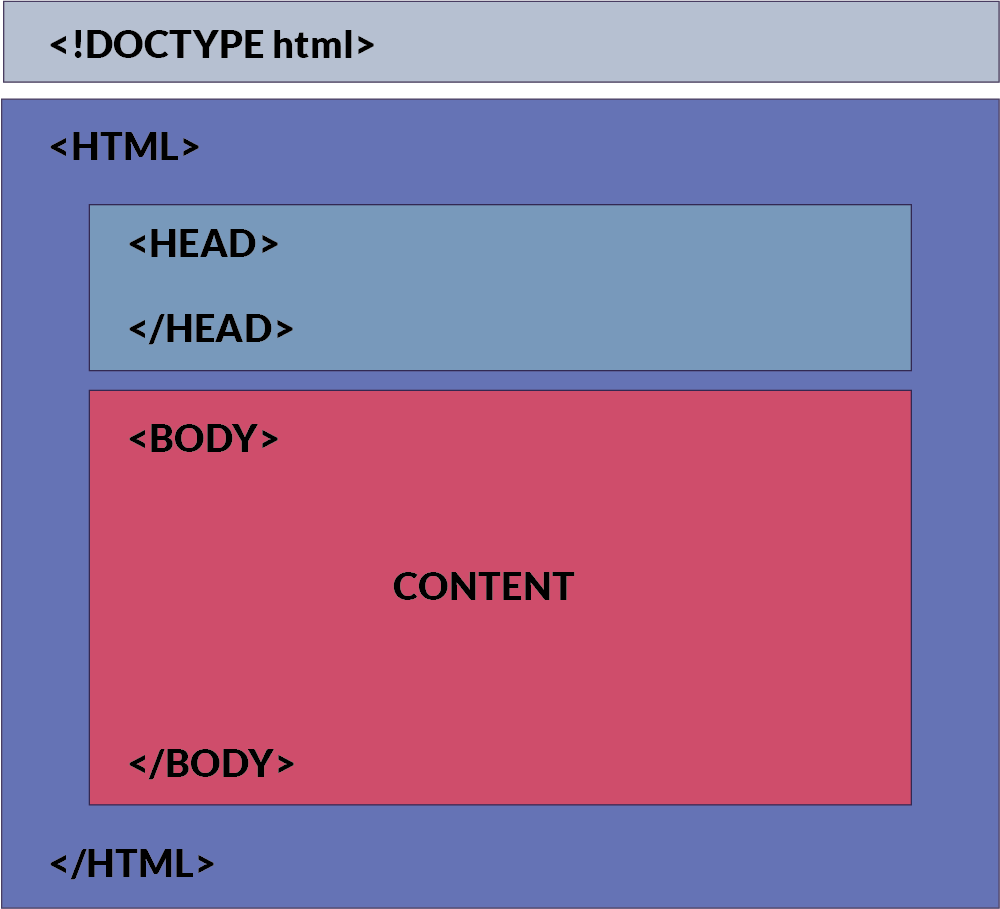
Every HTML page should include at least these essential elements to define the basic layout. In upcoming tutorials, we'll dive deeper into the fascinating world of HTML.

**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>

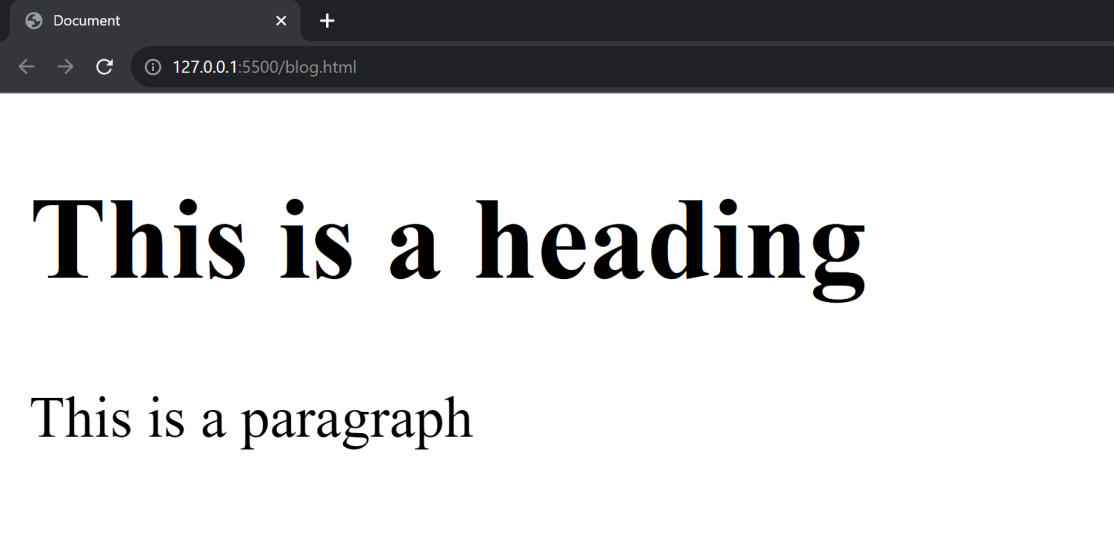
<p>This is a paragraph</p>

</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.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Refund](https://www.codewithharry.com/refund/)

# 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>.

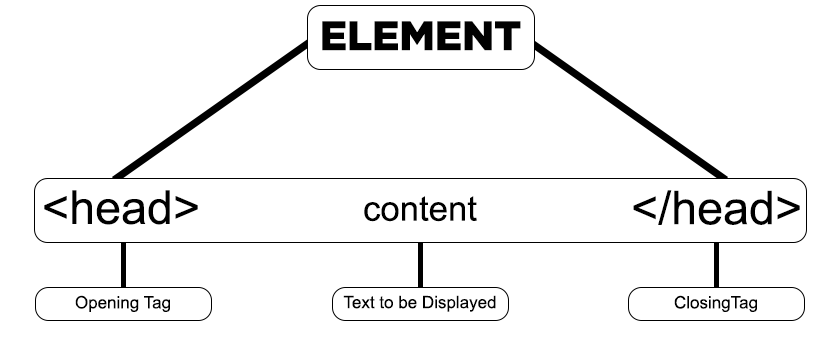
**Note:** Later if you happen to use react or a framework like Next.js, you will have to close the tag like this <br/> <hr/>. So it is better to cultivate the habit!

#### 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



< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Refund](https://www.codewithharry.com/refund/)

# 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.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Refund](https://www.codewithharry.com/refund/)

**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.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

# [Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)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.

-->

Copy

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

# [Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)HTML Id & Classes

HTML offers multiple ways to select and style elements. Two of the most commonly used selectors are IDs and Classes. This blog explores what they are, how to use them, and their differences.

**What is an ID?**

An ID is an attribute, a unique identifier assigned to only one HTML element within a page. It is often used for unique styling and JavaScript manipulations.

<div id="myUniqueID">This is a div with an ID.</div>

Copy

**What are Classes?**

The class attribute lets you give the same name to multiple HTML elements. That way, you can easily change their look or behavior all at once. Classes are not unique and can be assigned to multiple elements. They are generally used for applying the same styles or behaviors to a group of elements.

<div class="myClass">This is a div with a class.</div>

<p class="myClass">This is a paragraph with the same class.</p>

Copy

**The Style Tag**

The style tag in HTML is used to include embedded CSS (Cascading Style Sheets) within an HTML document. It is commonly placed within the <head> section of the HTML file, although it can technically be used in the <body> as well. The style tag allows you to define the look and feel of various HTML elements on the page, including their colors, sizes, margins, and other visual styles.

Here's a simple example:

<!DOCTYPE html>

<html>

<head>

<style>

/\* CSS rules go here \*/

p {

color: blue;

font-size: 18px;

}

.highlight {

background-color: yellow;

}

</style>

</head>

<body>

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

<p class="highlight">This paragraph has a yellow background.</p>

</body>

</html>

Copy

In this example, we have targetted the second paragraph by its class name in CSS. The style tag is used to add CSS right into HTML. We will learn about CSS and selectors later in the CSS tutorial

**Using IDs and Classes in CSS**

In CSS, elements with IDs are selected using a hash (#) symbol before the ID, and elements with classes are selected using a dot (.) before the class name.

/\* CSS for ID \*/

#myUniqueID {

background-color: yellow;

}

/\* CSS for Class \*/

.myClass {

font-size: 18px;

}

Copy

**Differences Between IDs and Classes**

* **Uniqueness:** IDs are unique, and classes can be reused.
* **JavaScript:** IDs are often used for JavaScript operations.
* **Styling:** Both can be used for styling, but IDs have higher specificity.

**Conclusion**

Understanding the difference between IDs and Classes is crucial for effective web development. While IDs are for unique elements, classes are for grouping elements.

< < PreviousNext > >



**CodeWithHarry**Managed by CWH Solutions

[Privacy](https://www.codewithharry.com/privacy/)[Terms](https://www.codewithharry.com/terms/)[Shop](https://www.codewithharry.com/shop/)[Contact](https://www.codewithharry.com/contact/)[Refund](https://www.codewithharry.com/refund/)