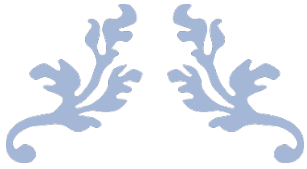


HTML

Hyper **T**ext **M**arkup **L**anguage
Become A Master

Muhammad Umair Ahmad
www.umairahmad.net

Introduction to HTML

What is HTML?

HTML is the standard markup language for creating Web pages.

- **HTML** stands for **H**yper**T**ext **M**arkup **L**anguage
- **HTML** describes the structure of web pages using markup
- **HTML** is not case sensitive

What is Markup Language?

Markup languages are languages used by a computer to annotate a document. These languages are readable by humans, which means that they are usually written using standard words, instead of technical programming language terminology.

Markup languages define the style and structure of a document so that a computer knows how you want that document to appear.

The History of HTML

HTML was created by **Sir Timothy John Berners-Lee** in late *1991* and released in 1993, then **HTML** 2.0 was published in 1995. **HTML** 4.01 was published in late 1999 and was a major version of **HTML**.

- **HTML** 1.0 was released in 1993 with the intention of sharing information that can be readable and accessible via web browsers. But not many of the developers were involved in creating websites. So, the language was also not growing.

- Then comes the **HTML** 2.0, published in 1995, which contains all the features of **HTML** 1.0 along with that few additional features, which remained as the standard markup language for designing and creating websites until January 1997 and refined various core features of **HTML**.

- Then comes the **HTML** 3.0, where Dave Raggett who introduced a fresh paper or draft on **HTML**. It included improved new features of **HTML**, giving more powerful characteristics for webmasters in designing web pages. But these powerful features of new **HTML** slowed down the browser in applying further improvements.

- Then comes **HTML** 4.01 in 1999, which is widely used and was a successful version of **HTML** before **HTML** 5.0, which is currently released and used worldwide. **HTML** 5 can be said for an extended version of **HTML** 4.01, which was published in the year 2008.

HTML5

At the time of writing this book, **HTML5** is the fifth and current major version of the **HTML** standard that is a World Wide Web Consortium **(W3C)** recommendation. The current specification is known as the **HTML** Living Standard. It is maintained by the Web Hypertext Application Technology Working Group **(WHATWG)**, a consortium of the major browser vendors (Apple, Google, Mozilla, and Microsoft).

HTML5 was first released in a public-facing form on 22 January 2008, with a major update and "**W3C** Recommendation" status in October 2014. Its goals were to improve the language with support for the latest multimedia and other new features; to keep the language both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc.

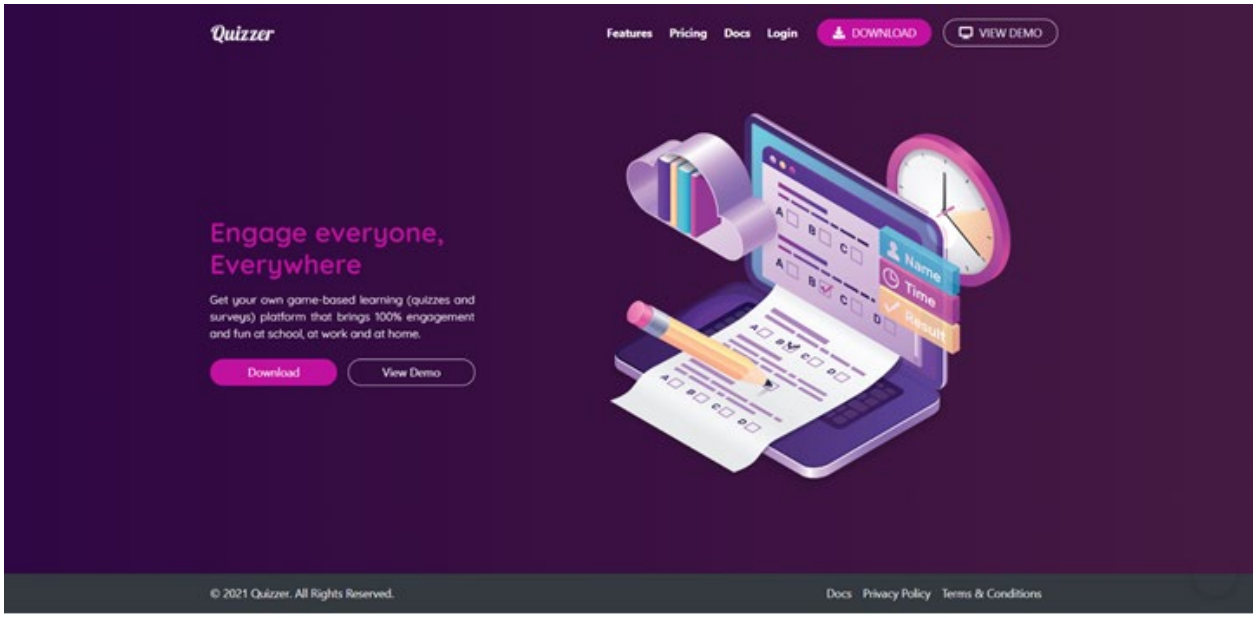
Why Learn HTML5?

It is essential to learn **HTML** if you want to build website, you can't build one if you don't know **HTML** because it's one of the prerequisites in learning other languages used for web development.

HTML5 has added support for many new features that will make it possible to do more with **HTML**, without relying on non-standard proprietary technologies.

Type of content	HTML 1.2	HTML 4.01	HTML5	Purpose
Heading	Yes	Yes	Yes	Organize page content by adding headings and subheadings to the top of each section of the page
Paragraph	Yes	Yes	Yes	Identify paragraphs of text
Address	Yes	Yes	Yes	Identify a block of text that contains contact information
Anchor	Yes	Yes	Yes	Link to another web content
List	Yes	Yes	Yes	Organize items into a list
Image	Yes	Yes	Yes	Embed a photograph or drawing into a web page
Table	No	Yes	Yes	Organize data into rows and columns
Style	No	Yes	Yes	Add CSS to control how objects on a web page are presented
Script	No	Yes	Yes	Add JavaScript to make pages respond to user behaviors (more interactive)
Audio	No	No	Yes	Add audio to a web page with a single tag
Video	No	No	Yes	Add video to a web page with a single tag
Canvas	No	No	Yes	Add an invisible drawing pad to a web page, on which you can add drawings (animations, games, and other interactive features) using JavaScript

Structure


Quizzer
Features
Pricing
Docs
Login
DOWNLOAD
VIEW DEMO
Engage everyone, Everywhere
Get your own game-based learning (quizzes and surveys) platform that brings 100% engagement and fun at school, at work and at home.
Download
View Demo
Name
Time
Result
© 2021 Quizzer. All Rights Reserved.
Docs
Privacy Policy
Terms & Conditions


<body>
<header>Your header content</header>
<main>Your main content</main>
<footer>Your footer content</footer>
</body>

Tags and Elements

What are HTML Tags?

Tags are used to mark up the start of an HTML element and they are usually enclosed in angle brackets.

HTML Tags are element names surrounded by angle brackets.

An example of a tag is: **<h1>**.

Most tags must be opened **<h1>** and closed **</h1>** in order to function. For example:

<h1>This is a heading**</h1>**

What is an HTML element?

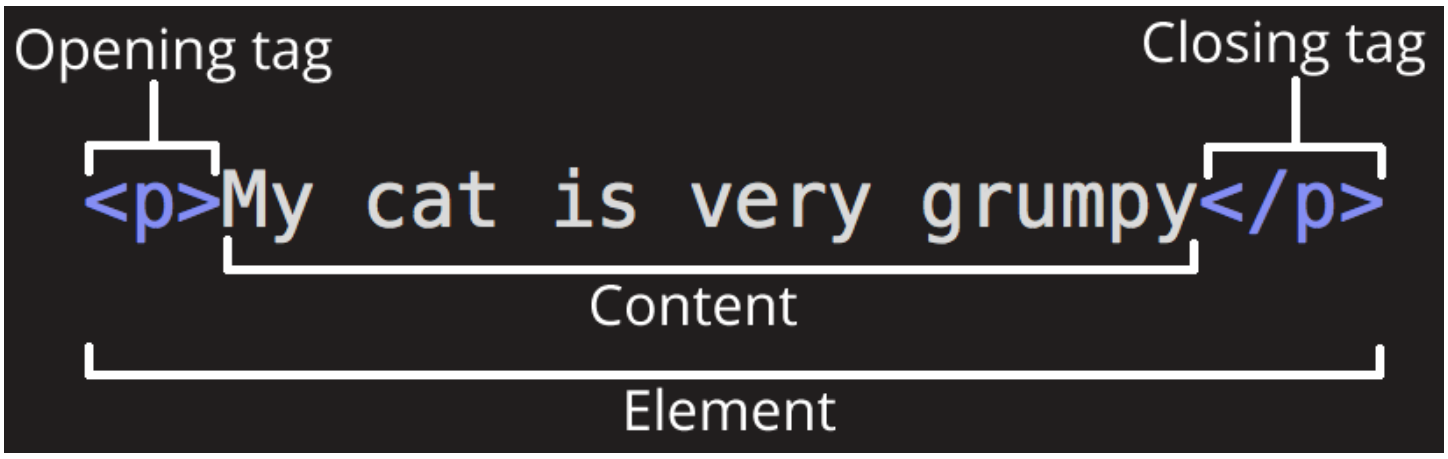
An HTML element is defined by a start tag, some content, and an end tag:

<tagname>Content goes here...**</tagname>**

The HTML element is everything from the start tag to the end tag:

<h1>My First Heading**</h1>**

** **My first paragraph** **



- HTML tags normally come in pairs like and
- The first tag in a pair is called the opening tag or starting tag, the second tag is called the closing tag or ending tag.
- The closing tag is written like the opening tag, but with a forward slash inserted before the tag name.
- The characters in the brackets indicate the tag's purpose.
- For example, in the tags above the p stands for paragraph.
- HTML elements are the building blocks of HTML pages.
- HTML elements are represented by tags.
- Browsers do not display the HTML tags, but use them to render the content of the page.

A very basic structure of a web page

Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Webpage</title>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph</p>
</body>
</html>
```

Output:

Code Explained:

- **`<!DOCTYPE html>`**: this declares the document of type which is **HTML5**.
- **`<html>`**: this element encloses everything inside of an html document; it includes tags, elements, style sheets, scripts, text, multimedia, and a lot more.
- **`<head>`**: this element encloses the metadata of a document which will not be displayed on the main content of a web page; this could include style sheets, scripts, **`<title>, <meta>`** tags and a lot more.
- **`<title>`**: this element defines the title of a web page; it appears on the upper-part of a browser.
- **`<body>`**: this element encloses elements like **`<h1>, <p>,<img>`** and a lot more.
- **`<h1>`**: this element defines the largest heading.
- **`<p>`**: this element defines a paragraph.

```
<html>
    <head>
        <title>This Is Your Title </title>
    </head>
    <body>
        <h1> This Is Your Header </h1>
        <p> This is your paragraph. </p>
    </body>
</html>
```

Nested HTML Elements

HTML elements can be nested (this means **HTML** element can contain one or more **HTML** elements).

For you to better understand it look at the example code below.

** **This text is italicized** **

Output

This text is italicized

Never Skip the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

Code:

```
<!DOCTYPE html>
<html>
<body>
    <h1>This is a heading
    <p>This is a paragraph
</body>
</html>
```

Output:

This is a heading

This is a paragraph

Empty Elements

Empty Elements are elements that do not have an element content and an end tag.

The
 tag defines a line break, and is an empty element without a closing tag:

Code:

```
<p>This is a <br /> paragraph with a line break</p>
```

Output:

This is a

Paragraph with a line break

A list of commonly used Empty Elements:

1. `<meta />`
2. `<link />`
3. `<img />`
4. `<br />`
5. `<hr />`
6. `<input />`

The best practice in **HTML** Empty Elements is to always put a forward slash **/** sign before the grater than **>** sign.

In this way they are closed at their start tags. These tags are also called self-closing tags.

HTML is Not Case Sensitive

HTML tags are not case sensitive: **<P>** means the same as **<p>**. **<H1>** means the same as **<h1>**.

The HTML standard does not require lowercase tags, but W3C recommends lowercase in **HTML**, and demands lowercase for stricter document types like XHTML.

We will always use lowercase tag names.

Creating First HTML Page

Where to write/edit HTML?

A simple text editor is all you need to learn **HTML**.

Web pages can be created and modified by using professional **HTML** editors.

However, for learning **HTML** let's start with a simple text editor like Notepad (Windows), TextEdit (macOS) or Text Editor (Linux).

Follow the steps below to create your first web page with Notepad, TextEdit or Text Editor.

1. Open Notepad on Windows or TextEdit on macOS or Text Editor on Linux.
2. Write some HTML.

```
<!DOCTYPE html>
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph</p>
</body>
</html>
```

3. Save the HTML page.
    a. Save the file on your computer. Select File > Save in the Notepad menu.
    b. Name the file "index.html" and set the encoding to UTF-8 (which is the preferred encoding for HTML files).
4. View the HTML page in your browser.

Text Editors / Integrated Development Environment (IDE)

Editing **HTML** code can be done without any specific tools. In fact, if you have a simple text editor, you are good to go. However, just because you can do something doesn't mean it is the best way to do it – and that applies to web development as well.

If you use proper tools for your work, not only will you make things easier for yourself, but you will also step it up on the quality level.

An **integrated development environment (IDE)** is a software application that provides comprehensive facilities to computer programmers for software development. An **IDE** normally consists of at least a source code editor, build automation tools and a debugger.

In a text editor! Creating HTML files is free you don't need to download expensive applications to do so.

Look at the list of below for some free apps you could use to create/edit HTML files.

- Notepad
- TextEdit
- Text Editor
- Notepad++
- Visual Studio Code
- Brackets
- Sublime Text
- Atom
- Intellij IDEA

HTML Basic

- All the **HTML** documents must start with a document type declaration **<!DOCTYPE html>.**
- The **HTML** document itself begins with **<html>** and ends with **</html>**.
- The visible part of the **HTML** document is between **<body>** and **</body>.**

Code:

```
<!DOCTYPE html>
<html>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph</p>
</body>
</html>
```

What is <!DOCTYPE>?

- The **<!DOCTYPE>** declaration represents the document type, and helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any **HTML** tags).
- The **<!DOCTYPE>** declaration is not case sensitive.
- The **<!DOCTYPE>** declaration for the **HTML5** is **<!DOCTYPE html>**.

Attributes

The HTML attributes are used to provide more information to an **HTML** element.

- **HTML** attributes only appear at start tags. It will never be on end tags.
- All **HTML** elements can have attributes.
- **HTML** elements can have multiple attributes.
- **HTML** attributes usually come in name/value pairs like: `name="value"`
    - An attribute `name`
    - An equal `=` sign
    - A value surrounded by quotation marks `"value"`

You can also use single quotation marks depending on the situation esp. when the value contains double quotes.

For you to better understand it look at the example code below.

```
<p title="I serve as a tooltip">This is a paragraph</p>

<p title="I'm a tooltip">This is a paragraph</p>

<p title='It's going to break'>This is a paragraph</p>
```

```
<a href="login.html">Login</a>

<a href="https://umairahmad.net" target="_blank">Muhammad Umair Ahmad</a>
```

There are two ways to specify the URL in the **href** attribute.

1. **Absolute URL**: Links to an external website that is hosted on another server.
2. **Relative URL**: Links to an **html** page that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: `href="login.html"`.

Block Level Elements & Inline Elements

All the HTML elements can be categorized into two categories.

- Block-level Elements
- Inline Elements

Block-level Elements

A block-level element is any element that starts a new line and uses the full width of the page or container. For example: **<h1>, <p>, **.

A block-level element can take up one line or multiple lines and has a line break before and after the element.

The ** ** element is a block-level element.

Code:

```
<div>This is a div element</div>
```

Output:

This is a div element

Inline Elements

Alternatively referred to as in-line, inline is any element that displays in a line.

An inline element only takes up as much width as necessary.

The element is an inline element.

Code:

```
<p>I love my <span>country</span></p>
```

Output:

I love my country

Semantic Elements

When building web pages, we use a combination of non-semantic **HTML** and Semantic **HTML**. The word semantic means ***"relating to meaning"***, so semantic elements provide information about the content between the opening and closing tags.

By using Semantic **HTML**, we select **HTML** elements based on their meaning, not on how they are presented. Elements such as **`<div>`** and **`<span>`** are not semantic elements since they provide no context as to what is inside of those tags.

For example, instead of using a **`<div>`** element to contain our header information, we could use a **`<header>`** element, which is used as a heading section. By using a **`<header>`** tag instead of a **`<div>`**, we provide context as to what information is inside of the opening and closing tag.

Why Use Semantic Elements?

- **Accessibility:** Semantic **HTML** makes webpages accessible for mobile devices and for people with disabilities as well. This is because screen readers and browsers are able to interpret the code better.
- **SEO:** It improves the website **SEO**, or **S**earch **E**ngine **O**ptimization, which is the process of increasing the number of people that visit your webpage. With better **SEO**, search engines are better able to identify the content of your website and weight the most important content appropriately.
- **Easy to Understand:** Semantic **HTML** also makes the website's source code easier to read for other web developers.

To better understand this, you can think of comparing non-semantic **HTML** to going into a store with no signs on the aisles. Since the aisles aren't labeled, you don't know what products are in those aisles. However, stores that do have signs for each aisle make it a lot easier to find the items you need, just like Semantic **HTML**.

Many web sites contain **HTML** code like: **`<div id="header">, <div id="nav">, <div id="main">, <div id="footer">`** to indicate header, navigation, main content and footer.

In **HTML5** there are some new semantic elements that can be used to define different parts of a web page.

- **<article>**
- **<aside>**
- **<details>**
- **<figcaption>**
- **<figure>**
- **<footer>**
- **<header>**
- **<main>**
- **<mark>**
- **<nav>**
- **<section>**
- **<summary>**
- **<time>**