https://dumudigitikakenya.com/

LEARN HTML IN ONE HOUR

ONE HOUR CRASH COURSE: BASICS OF HTML PROGRAMMING

INTRODUCTION

HTML is an acronym for Hypertext Markup Language. It's the most basic building block of the web, allowing you to add, change, and remove website content, which is essential knowledge for web developers and beneficial for software developers, marketing and sales professionals needing to edit listings, business owners wanting to create their own websites, and freelancers who can earn good money for quality websites. Adding HTML to your resume can't hurt. Learning HTML is super easy; this full course is only about an hour read (I will also be linking sections to my detailed articles). Anyone can learn HTML, including you, so welcome.

Building a website is akin to constructing a house. First, we need a solid foundation and a skeletal structure. HTML serves as the fundamental building block of the web. Then, we can enhance the appearance with CSS (Cascading Style Sheets), which allows us to style, color, and change the appearance of a webpage. Finally, there's JavaScript, which adds functionality comparable to plumbing, heating, air conditioning, and electricity in a house.
Combining these three levels yields a well-functioning webpage.

CHOOSING A TEXT EDITOR

A text editor is a software used for editing plain text files. You can choose any text editor based on your preference. Some popular choices include Visual Studio Code, Sublime Text, Atom, and Notepad++.

I will be using Visual Studio Code.

CREATING AN HTML FILE

To create an HTML file, follow these steps:

- Open your chosen text editor.
- Create a new file.
    - Save the file with a .html extension (e.g., index.html - like mine).

BASIC HTML STRUCTURE

Every HTML document consists of the following basic structure:

```
<!DOCTYPE html>
<html>
<head>
  <title>Title of the Document</title>
</head>
<body>
<!-- Content goes here -->
</body>
</html>
```

INTRODUCTION TO HTML ELEMENTS

HTML (Hypertext Markup Language) provides a set of elements that define the structure and content of web pages. Understanding HTML elements is crucial for creating well-structured and semantic web documents.

ANATOMY OF AN HTML ELEMENT

HTML elements consist of the following parts:

Opening Tag: Begins the element and contains the element's name.

Closing Tag: Ends the element and contains a forward slash before the element name.

Content: The content of the element.

Attributes: Additional information about the element.

* This is a paragraph. *

In this example:

*** *** is the opening tag.

*** *** is the closing tag.

This is a paragraph. is the content.

BASIC TAGS

HTML tags are keywords surrounded by angle brackets (< and >), which are used to define the structure of web pages. Some basic tags include:

- ***<html>***: Defines the root of an HTML document.
- ***<head>***: Contains meta-information about the document.
- ***<title>***: Sets the title of the document.
- ***<body>***: Defines the content of the document.
    - ***<h1>*** to ***<h6>***: Defines headings of different levels, where <h1> is the highest level and <h6> is the lowest level.

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

- **<p>**: Defines a paragraph.
- ***<a>***: Defines a hyperlink.
- ***<img>***: Defines an image.
- ***<ul>***: Defines an unordered list.
- **<ol>**: Defines an ordered list.
- **<li>**: Defines a list item.
- ***
***: Defines a line break.
- ***<hr>***: Defines a thematic break or horizontal rule.
- *** ***: Defines a division or section in an HTML document.
- ******: Defines an inline container for a section of text.
- ****** or <b>: Defines bold text.
- ****** or <i>: Defines italic text.
- ***<u>***: Defines underlined text.
- ***<table>***: Defines a table.
- ***<tr>***: Defines a row in a table.
- ***<th>***: Defines a header cell in a table.
- ***<td>***: Defines a standard cell in a table.
- ***<form>***: Defines an HTML form for user input.
- ***<input>***: Defines an input control.
- ***<select>***: Defines a dropdown list.
- ***<option>***: Defines an option in a dropdown list.
- ***<textarea>***: Defines a multiline input control (text area).
- ***<iframe>***: Defines an inline frame for embedding external content.
- ***<meta>***: Defines metadata about an HTML document.

COMMENTS

HTML comments are used to add notes to your code, which are not displayed in the web browser. Comments start with <!-- and end with -->.

Example:

<!-- This is a basic HTML comment. -->

Why Use HTML Comments?

Documentation

Debugging

Collaboration

FORMATTING A PAGE

HTML provides various tags for formatting text, such as:

<h1> to <h6>: Heading tags for different levels of headings.

*** ***: Paragraph tag.

***
***: Line break tag.

<!DOCTYPE html>

<html>

<body>

<h1>Main Heading</h1>

* This is a paragraph. *

<h2>Subheading</h2>

* This is another paragraph. *

</body>

</html>

LINKS

Links are used to navigate between web pages. They are created using the <a> tag with an *href* attribute.

Example:

<!DOCTYPE html>

<html>

<body>

Visit Example

</body>

</html>

IMAGES

You can display images on a web page using the <img> tag with a src attribute, which specifies the URL of the image.

Example:

<!DOCTYPE html>

<html>

<body>

<img src="image.jpg" alt="Description of the image">

</body>

</html>

VIDEOS & YOUTUBE IFRAMES

To embed videos or YouTube videos, you can use the <video> tag for native videos or <iframe> tag for embedding YouTube videos.

Example (Native Video):

<!DOCTYPE html>

<html>

<body>

<video width="320" height="240" controls>

<source src="movie.mp4" type="video/mp4">

Your browser does not support the video tag.

</video>

</body>

</html>

Example (YouTube iFrame):

```
<!DOCTYPE html>
<html>
<body>
<iframe width="560" height="315" src="https://www.youtube.com/embed/video_id" frameborder="0" allowfullscreen></iframe>
</body>
</html>
```

LISTS

HTML provides two types of lists: ordered lists (<ol>) and unordered lists (<ul>).

Example:

```
<!DOCTYPE html>
<html>
<body>
<h2>Unordered List</h2>
<ul>
<li>Item 1</li>
<li>Item 2</li>
</ul>
```

<h2>Ordered List</h2>

<ol>

*
First item
*

*
Second item
*

*
*

</body>

</html>

TABLES

Tables are created using the <table>, <tr>, <th>, and <td> tags.

Example:

<!DOCTYPE html>

<html>

<body>

<table border="1">

<tr>

<th>Heading 1</th>

<th>Heading 2</th>

</tr>

<tr>

<td>Row 1, Col 1</td>

<td>Row 1, Col 2</td>

</tr>

<tr>

<td>Row 2, Col 1</td>

<td>Row 2, Col 2</td>

</tr>

</table>

</body>

</html>

DIVS & SPANS

 and are container elements used to group HTML elements for styling and scripting purposes.

Example:

<!DOCTYPE html>

<html>

<body>

* *

This is a div.

* *

This is a span.

</body>

</html>

INPUT & FORMS

HTML provides various input types and form elements for user interaction.

Example:

```
<!DOCTYPE html>
<html>
<body>
<form action="/submit" method="post">
<label for="fname">First Name:</label>
<input type="text" id="fname" name="fname"><br><br>
<label for="lname">Last Name:</label>
<input type="text" id="lname" name="lname"><br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

HTML FORM ELEMENTS

HTML forms are used to collect user input and send it to a server for processing. Understanding HTML form elements is essential for creating interactive and dynamic web applications that gather data from users.

Form Structure

A basic HTML form consists of one or more form elements enclosed within a `<form>` tag.

Example:

html

<form>

<!-- Form elements go here -->

</form>

Input Elements

Text Input

The `<input>` element with `type="text"` creates a single-line text input field.

Example:

<input type="text" name="username">

Password Input

The `<input>` element with `type="password"` creates a password input field.

Example:

<input type="password" name="password">

Checkbox

The `<input>` element with `type="checkbox"` creates a checkbox input field.

Example:

<input type="checkbox" name="subscribe" value="yes">

Radio Button

The `<input>` element with `type="radio"` creates a radio button input field.

Example:

<input type="radio" name="gender" value="male"> Male

<input type="radio" name="gender" value="female"> Female

Select Dropdown

The `<select>` element creates a dropdown list, and `<option>` elements define the options within the dropdown.

Example:

<select name="country">

<option value="USA">United States</option>

<option value="UK">United Kingdom</option>

<option value="Canada">Canada</option>

</select>

Textarea

The `<textarea>` element creates a multiline text input field.

Example:

<textarea name="message" rows="4" cols="50"></textarea>

Form Submission

The `<input>` element with `type="submit"` or `<button>` element with `type="submit"` is used to submit the form data to the server.

Example:

<input type="submit" value="Submit">

Form Attributes

HTML forms can have various attributes such as `action`, `method`, `target`, and `enctype` for controlling form submission behavior.

Example:

```
<form action="/submit-form" method="post" target="_blank" enctype="multipart/form-data">
<!-- Form elements go here -->
</form>
```

IFRAMES

iFrames are used to embed another HTML document within the current document.

Example:

```
<!DOCTYPE html>
<html>
<body>
<iframe src="anotherpage.html" width="600" height="400"></iframe>
</body>
</html>
```

META TAGS

Meta tags provide metadata about the HTML document, such as author, description, keywords, etc.

Example:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="description" content="This is a description of the web page.">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
</head>
<body>
<!-- Page content goes here -->
</body>
</html>
```

HTML FILE PATHS

In HTML, file paths are used to link external resources such as images, stylesheets, scripts, and other HTML files. Understanding file paths is essential for creating well-structured and maintainable web projects.

Absolute vs. Relative Paths

- Absolute Paths: Start from the root directory of a website. They include the full URL to the resource.
- Relative Paths: Start from the current location of the HTML file. They are shorter and more flexible but depend on the file's location within the directory structure.

Types of HTML File Paths

1. Path to an HTML File: Used to link to other HTML files within the same project.

About Us

2. Path to an External Website: Used to link to external websites.

Visit Example

3. Path to an Image File: Used to embed images in HTML documents.

<img src="images/logo.png" alt="Logo">

4. Path to a Stylesheet File: Used to link CSS files to HTML documents.

<link rel="stylesheet" href="styles/main.css">

5. Path to a Script File: Used to link JavaScript files to HTML documents.

<script src="scripts/main.js"></script>

Understanding Relative Paths

Relative paths are relative to the location of the current HTML file. They can be:

- Relative to the Current Directory: Use a filename or folder name to reference resources in the same directory.
- Relative to the Root Directory: Start with a forward slash (/) to reference resources from the root directory.
- Relative to the Parent Directory: Use "../" to reference resources in the parent directory.

Example:

<!-- Relative to the Current Directory -->

<link rel="stylesheet" href="styles/main.css">

<!-- Relative to the Root Directory -->

<img src="/images/logo.png" alt="Logo">

<!-- Relative to the Parent Directory -->

Home

Best Practices for File Paths

- Use relative paths whenever possible for better project portability.
- Ensure that file names and folder names are case-sensitive, especially on web servers.
- Use forward slashes (/) as directory separators, even on Windows systems, for cross-platform compatibility.

HTML STYLES

In HTML, styles are used to enhance the appearance of web pages by applying colors, fonts, layouts, and other visual effects. Understanding HTML styles is essential for creating visually appealing and user-friendly web designs.

Inline Styles

Inline styles are applied directly to individual HTML elements using the `style` attribute.

Example:

```
<p style="color: red; font-size: 20px;">This is a paragraph with inline styles.</p>
```

Internal Styles

Internal styles are defined within the `<style>` element in the `<head>` section of an HTML document.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
color: blue;
font-size: 18px;
}
</style>
</head>
<body>
<p>This is a paragraph with internal styles.</p>
</body>
</html>
```

External Styles

External styles are defined in separate CSS (Cascading Style Sheets) files and linked to HTML documents using the `<link>` element.

Example:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<p>This is a paragraph with external styles.</p>
</body>
</html>
```

CSS Selectors

CSS selectors are used to target HTML elements and apply styles to them. Common selectors include:
- Element Selector: Targets HTML elements by their tag name.
- Class Selector: Targets HTML elements with a specific class attribute.
- ID Selector: Targets HTML elements with a specific ID attribute.
- Attribute Selector: Targets HTML elements with a specific attribute value.

Example:

```
/* Element Selector */
p {
color: green;
}

/* Class Selector */
.my-class {
font-weight: bold;
}

/* ID Selector */
#my-id {
text-decoration: underline;
}

/* Attribute Selector */
input[type="text"] {
border: 1px solid #ccc;
}
```

HTML TEXT FORMATTING

In HTML, text formatting allows you to change the appearance of text on a web page. Understanding text formatting is crucial for creating visually appealing and well-structured content.

Bold Text

To make text bold, use the `<strong>` or `<b>` tags.

Example:

This is bold text.

Italic Text

To make text italic, use the `<em>` or `<i>` tags.

Example:

This is italic text.

Underlined Text

To underline text, use the `<u>` tag.

Example:

This is <u>underlined</u> text.

Strikethrough Text

To strikethrough text, use the `<s>` or `<strike>` tags.

Example:

This is <s>strikethrough</s> text.

Superscript and Subscript Text

To create superscript and subscript text, use the `<sup>` and `<sub>` tags respectively.

Example:

This is ^{superscript} text.

This is _{subscript} text.

Text Color

To change the color of text, use the `color` attribute within the `<font>` tag (deprecated in HTML5) or use CSS.

Example using `<font>` (deprecated):

This is red text.

Example using CSS:

This is blue text.

Text Size

To change the size of text, use the `size` attribute within the `<font>` tag (deprecated in HTML5) or use CSS.

Example using `<font>` (deprecated):

This is large text.

Example using CSS:

This is large text.

Preformatted Text

To display text exactly as it is written, use the `<pre>` tag.

Example:

<pre>

This is

preformatted

text.

</pre>

HTML QUOTATION AND CITATION

In HTML, quotation and citation elements are used to mark up quotations and citations within a document. Understanding how to properly use these elements is important for creating well-structured and semantically meaningful content.

Quotation Elements

Blockquote

The `<blockquote>` element is used to indicate a section of quoted content that is separate from the main text.

Example:

<blockquote>

* This is a quoted text. *

<footer>Author</footer>

</blockquote>

Inline Quotation

The `<q>` element is used to mark up inline quotations within a paragraph.

Example:

* This is an <q>inline quotation</q> within a paragraph. *

Citation Elements

Citation

The `<cite>` element is used to mark up the title of a work (e.g., book, article, or webpage) being cited.

Example:

<cite>Title of the Work</cite> by Author

Author Name

The `<author>` element is used to mark up the name of the author of a cited work.

Example:

* Title of the Work <author>Author Name</author> *

Additional Attributes

`cite`

The `cite` attribute is used to specify the URL of the source being cited.

Example:

<blockquote cite="https://www.example.com">

* This is a quoted text. *

<footer>Author</footer>

</blockquote>

HTML FAVICON

In HTML, a favicon is a small icon associated with a website that appears in the browser's address bar, tabs, and bookmarks. Understanding how to add a favicon to your HTML document is essential for branding and improving the user experience of your website.

Adding a Favicon

Using `<link>` Element

To add a favicon to your HTML document, you can use the `<link>` element within the `<head>` section.

Example:

<head>

<link rel="icon" href="favicon.ico" type="image/x-icon">

</head>

Providing Different Sizes

You can provide multiple favicon sizes for different devices using the `sizes` attribute.

Example:

<head>

<link rel="icon" href="favicon.ico" type="image/x-icon" sizes="16x16 32x32">

</head>

Using PNG Format

You can also use PNG format for the favicon and specify the `type` attribute accordingly.

Example:

<head>

<link rel="icon" href="favicon.png" type="image/png">

</head>

HTML JAVASCRIPT

JavaScript is a programming language that enables interactive and dynamic behavior on web pages. Understanding how to integrate JavaScript into HTML documents is essential for creating rich and interactive web applications.

Inline JavaScript

You can add JavaScript code directly within HTML elements using the `onclick` attribute or within `<script>` tags.

Example:

<button onclick="alert('Hello, World!')">Click Me</button>

<script>

alert('Hello, World!');

</script>

External JavaScript

You can also include JavaScript code from an external file using the `<script>` tag with the `src` attribute.

Example:

<script src="script.js"></script>

JavaScript Events

JavaScript can respond to various events such as clicks, mouse movements, keyboard inputs, and more. These events can trigger JavaScript functions to perform specific actions.

Example:

<button onclick="myFunction()">Click Me</button>

<script>

function myFunction() {

alert('Button clicked!');

}

</script>

JavaScript Functions

JavaScript functions are blocks of reusable code that can be called to perform specific tasks. You can define functions inline or in external JavaScript files.

Example:

<button onclick="myFunction()">Click Me</button>

<script>

function myFunction() {

alert('Hello, World!');

}

</script>

JavaScript Variables

JavaScript variables are used to store data values. They can hold various types of data such as numbers, strings, arrays, objects, etc.

Example:

```
<script>
  var message = 'Hello, World!';
alert(message);
</script>
```