LEARN EASILY

0-100 HTML

START PROGRAMMING TODAY

by Edgar Hovhannisyan

HTML 0-100

Learn programming from scratch, create landing page and deploy to the server (hosting) for free.
The version 1.0 was published on 2023-10-05.

Tweet This Book! Please help by spreading the word about this book on Twitter! The suggested tweet for this book is: **I am going to learn #html with HTML0-100 by @edgarsblog** .
Join me on my journey https://edgarsblog.com
The suggested hashtag for this book is #HTML0-100. Find out what other people are saying about the book by clicking on the hashtag.

Beginner's Guide to HTML

5. HTML lists

- Unordered lists: ul (unnembered)
- Ordered lists: ol (numbered)
- Description lists: dl
- Nested lists

6. Images

- HTML Images Syntax
- The src (stands for source) attribute
- The alt attribute
- Image size: width and height
- Images on another server / website
- Animated Images
- Image as a Link
- HTML favicon image

7. HTML links (hyperlinks)

- Text links (websites, social media)
- Image as a link
- The target attribute
- Email, phone, messenger links
- Button as a Link
- The Download attribute

8. CSS styles: inline, internal and external

- Inline CSS
- Internal CSS
- External CSS

9. HTML tables

- What are tables, and why we need them?
- Table row (tr)
- Table head (th)
- Table data (td)
- Table borders
- Table styling
- Colspan and rowspan

10. HTML form

- Form elements
- The input element
- The label element
- The submit button
- The select element
- The textarea element

11. HTML File paths

- Absolute File Paths
- Relative File Paths

12. Website layout elements

- header
- navbar
- main
- section
- article
- sidebar
- footer

13. The landing page

- The layout
- Hosting (server)
- Happy End

Foreword

The HTML 0-100 teaches the fundamentals of Web Development.
You will build a real-world landing page from scratch and learn such useful things as: setting-up work environment, writing your first code, SEO and UI UX principles, manipulating with images and links, choosing free hosting and domain provider, deploying your files to the server.
Everything will be explained for you step by step.
The book comes with additional referenced reading material.
After studying with this book, you will be able to build simple landing pages and email marketing templates.
The material is kept up to date by myself and the community. In the HTML 0-100 I provide examples of code for designing your web page and deploying it to the free server with default free domain.
Besides paying for this book, you won't need to spend money for having your website online.
You can see in the book the best practices of real-world websites. Essentially, you will learn to build your own website from scratch, with features like styling, linking, creating navbar, header, main sections and footer.
I hope this book captures my enthusiasm for HTML, and, that it will help you to get started with programming.
With the Best Wishes, Edgar

About the author

I am an Armenian self-taught software and web engineer, dedicated to learning and teaching programming in HTML, CSS, JavaScript, PHP, WordPress. I bought my first web development course and started learning for my own.

After 3 months of learning HTML, CSS and Bootstrap I was able already to earn money as a freelancer, creating simple landing pages and web templates. Then, I went deeper into programming and started JavaScript, apparently developing my HTML and CSS skills. After working at a startup company for almost 2 years, I started my own courses of HTML and CSS, teaching online students more than a year. I noticed, that the way of my education program and communication with the students gives them motivation, and they are getting better each day. When I saw my students having success in freelance or finding a job as a junior programmer, I decided to create digital products, that will help people from all around the world to do what they really want and to have a solid income, working in IT. That's why you are reading this eBook now. This is the first and most important part of my Web Development Course. It will give you a great and stable basis for moving forward in programming and choose, which direction you want to go further.

The name HTML 0-100 means, that you start HTML with 0 knowledge about this language and finish this course with 100% of knowledge about HTML.

1. Introduction and Setup

What is HTML

HTML abbreviation means - Hyper Text Markup Language. It is the standard markup language for creating Web pages. The World Wide Web Consortium, or W3C recommends it.
HTML describes the structure of a Web page. It consists of a series of elements. HTML elements tell the browser how to display the content, such as texts and images, with the help of codes and symbols.
HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc. For example, if we have <h1>Text<h1> element, it will be displayed as a heading, because h1 tag stands for the heading text and it tells the browser to display itself as a heading. If we have Text element, it will be displayed as a simple paragraph text, because due to the markup language rules, p tag stands for simple paragraph text, and the browser understands that, with the help of HTML language. More about headings and paragraphs you will learn in the related section of this book.
The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.
A browser does not display the HTML tags, but uses them to determine how to display the document.

HTML role in cooperation with CSS and JS

For understanding the role of HTML better, let’s have a little grammatical example of HTML role in cooperation with CSS and JavaScript:
Let’s say we have such sentence: The white car moved:
It consists of 3 parts adjective – white, noun – the car, verb – moved.
Our noun is - HTML (the main thing)
Adjective – CSS (styles)
Verb – JS (action). So, HTML on the page shows the main thing on the page, CSS helps us to describe things from adjective side, how does the thing look (color, sizes, borders and so on). JavaScript gives the thing an action (functionality) – moving, closing, openning, etc.

The history of HTML

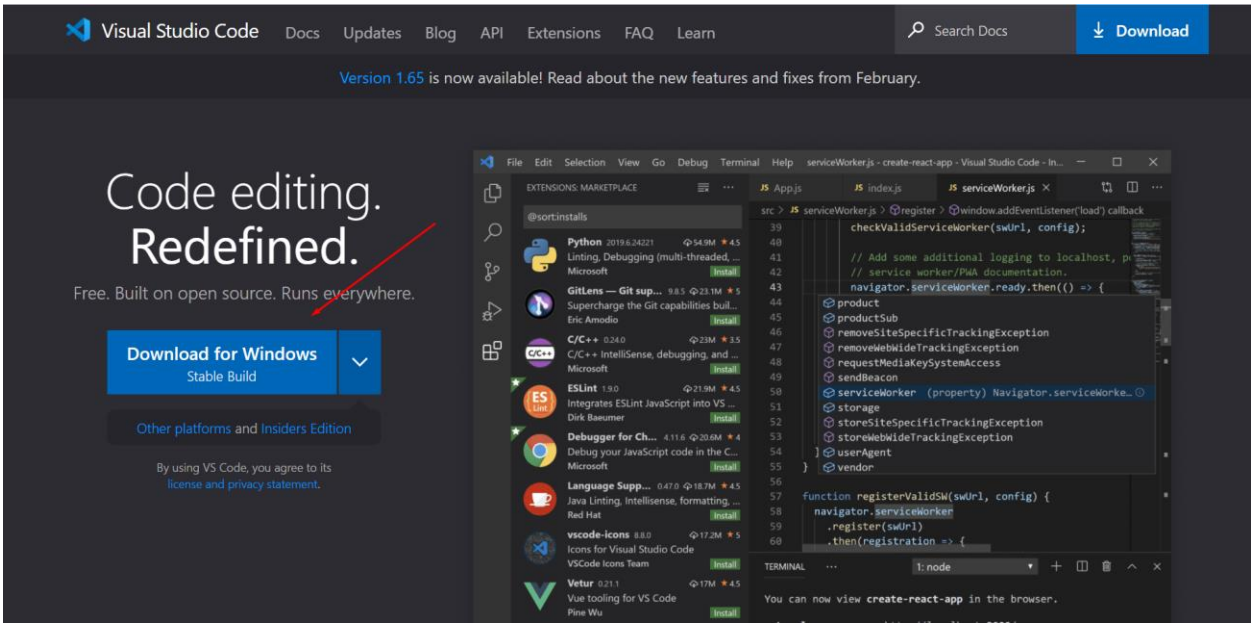
Since the early days of the World Wide Web, there have been many versions of HTML: in 1989 Tim Berners-Lee invented www (World Wide Web). 2 years later, in 1991, he invented already the HTML. The language's last version update was in 2017 - W3C Recommendation: HTML5.2.
This course follows the latest HTML5 standard.
More about the history of HTML you can learn by this link

Why learn HTML?

These days there are a lot of website builders, artificial intelligence tools for creating web pages and so on. But for understanding them right, manipulating with the code, bugs, or making some changes in your web pages, you need to know HTML. Although, HTML is not a programming language, but it is the core of creating, inspecting and understanding the web. All the page builders, contain HTML elements.
For becoming a programmer, digital marketing specialist or even UI UX designer, you will need HTML. The syntax of programming languages, or design flows also contain HTML elements.
HTML is simple and easy to learn, especially with this course.

Environment setup:
IDE, extensions, Emmet

As an instrument for coding and learning HTML you will need PC or laptop and a code editor, which is also called IDE - an **integrated development environment** (IDE). It is a software application, that helps programmers develop their code efficiently. IDEs have key features, such as editing, building, testing, and packaging websites or applications. One of the best choices is The Visual Studio Code Editor.



Visual Studio Code is a lightweight, but powerful source code editor, which runs on your desktop and is available for Windows, macOS and Linux. So, let's get started. On its official website https://code.visualstudio.com/ you can download the package, then locally install it on your PC or Laptop.

If you run macOS or Linux, press the dropdown menu at the right side of the button "Download for Windows" (shown with red arrow) and choose your operating system. You can also find more detailed information about vs code in introductory videos (link of the videos).

For making your work better, the VS Code comes with a lot of extensions, that help programmers to write code faster and easier. One of the recommended such extensions is Emmet. We will install it in IDE in the next chapter.

HTML page structure

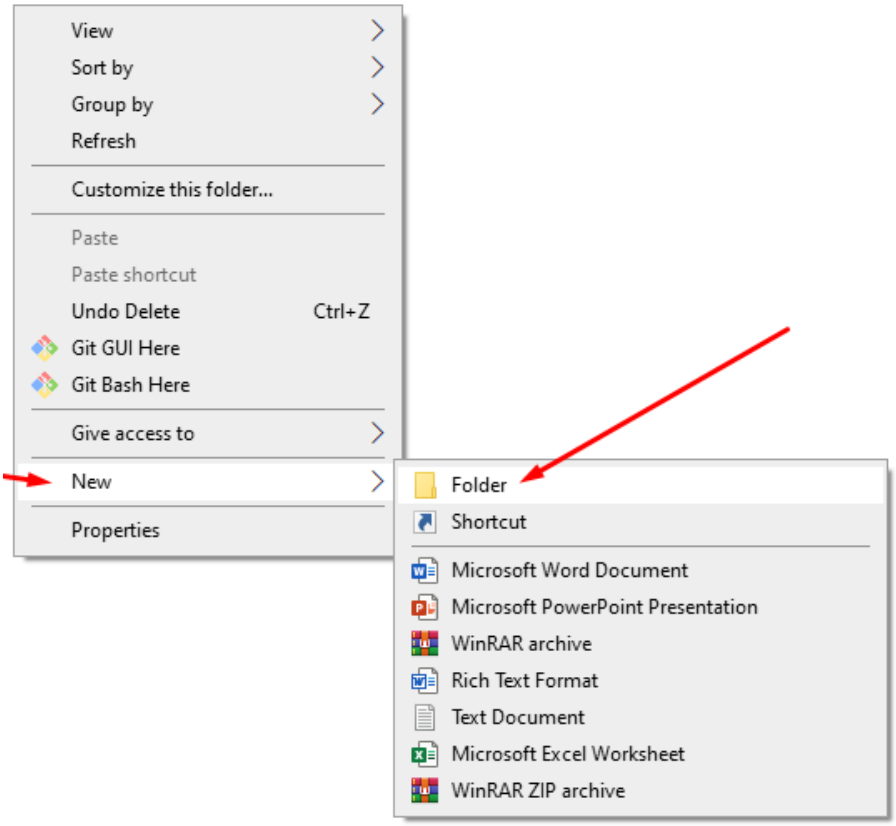
Below is a visualization of an HTML simple page structure without meta tags and links:

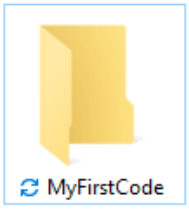
```
<!DOCTYPE html>
<html>
<head>
   <title>Document</title>
</head>
<body>
   <h1>I learn html</h1>
</body>
</html>
```

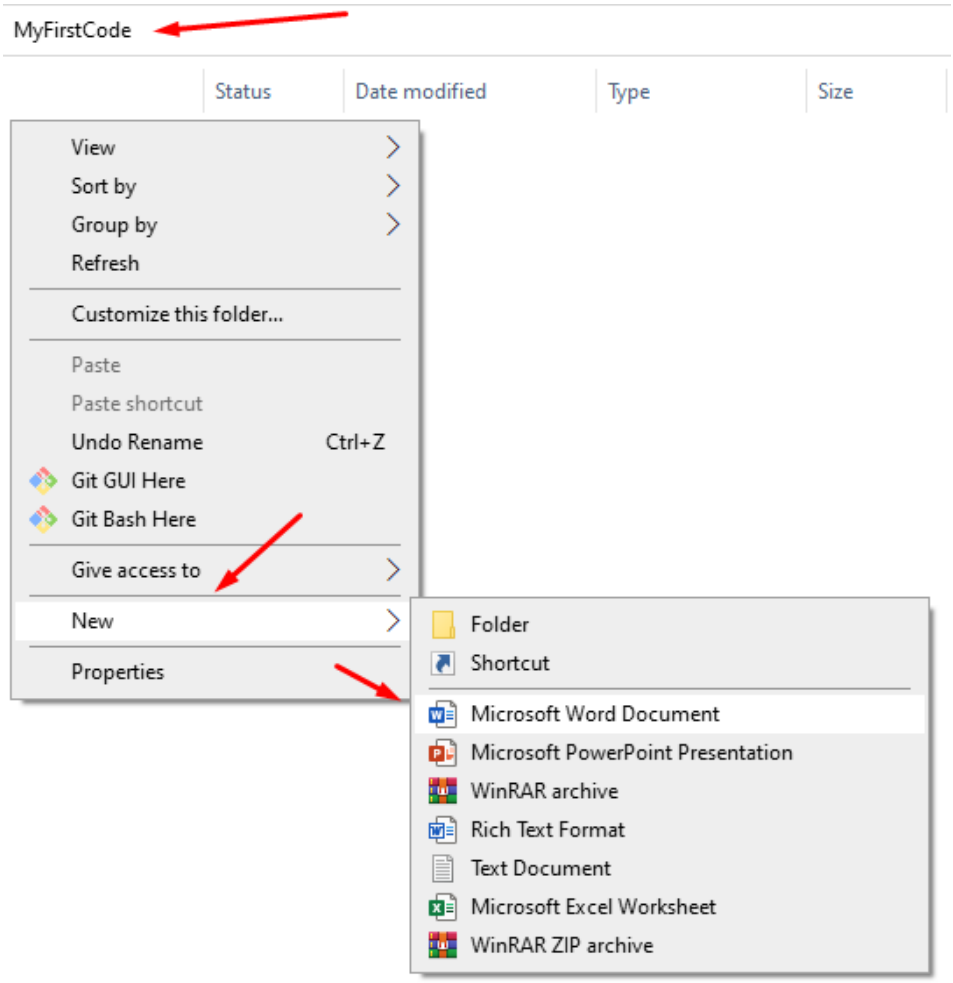
All HTML documents must start with a document type declaration: <!DOCTYPE html>. It represents the document type, and helps browsers to display web pages correctly. It must only appear once, at the top of the page (before any HTML tags). A browser starts reading an HTML page from the top, going down, and from the left to the right.
The HTML document itself begins with <html> and ends with </html> tags.
The visible part of the HTML document is between the <body> and </body> tags.
The content inside the <body> section will be displayed in the browser (what people will see on your website). The content inside the <title> element will be shown in the browser's title bar or in the page's tab. More detailed information about the head section will be provided in the 3-rd chapter.

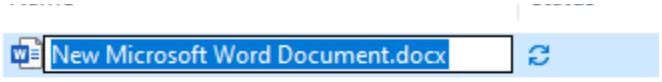
Your first code: index.html file

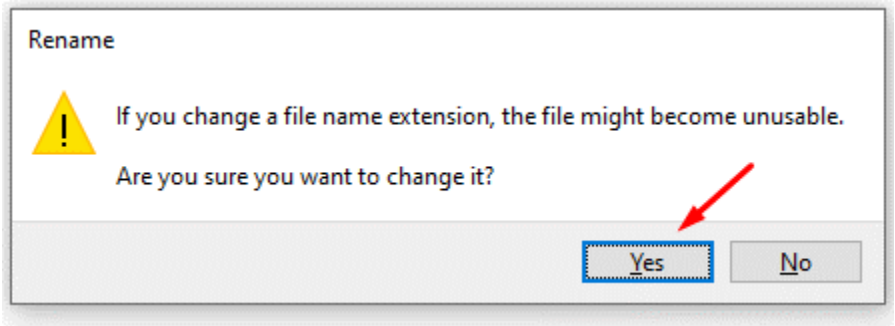
Now it's time to launch your first index.html file. For that purpose, open a new folder, name it whatever you want (in my case it's MyFirstCode) and inside that folder create a new file and rename it to index.html (see the images below).

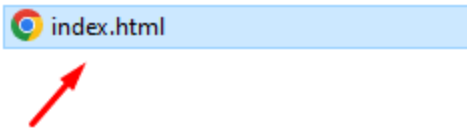




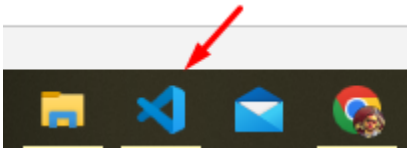

MyFirstCode
Status
Date modified
Type
Size
View
Sort by
Group by
Refresh
Customize this folder...
Paste
Paste shortcut
Undo Rename
Ctrl+Z
Git GUI Here
Git Bash Here
Give access to
New
Properties
Folder
Shortcut
Microsoft Word Document
Microsoft PowerPoint Presentation
WinRAR archive
Rich Text Format
Text Document
Microsoft Excel Worksheet
WinRAR ZIP archive


New Microsoft Word Document.docx

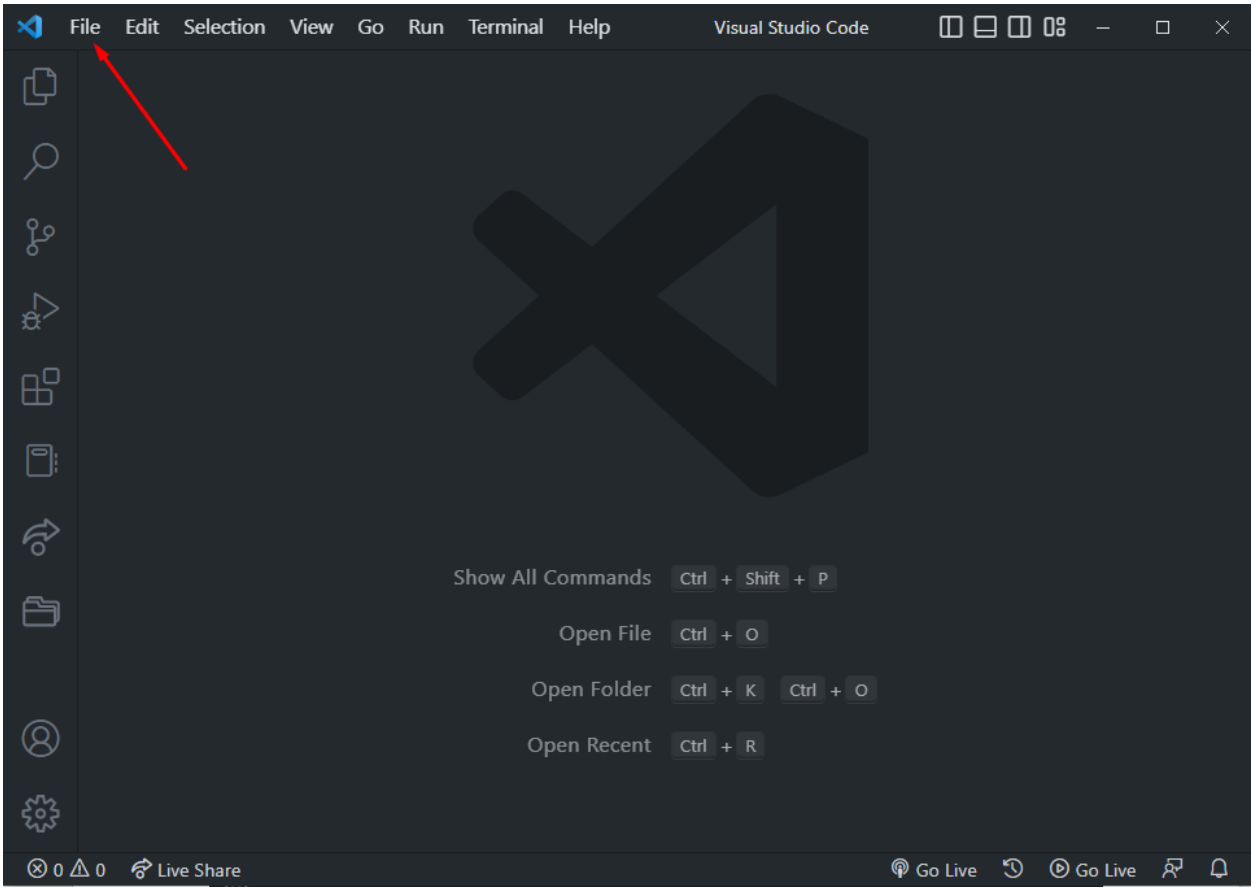

Rename
If you change a file name extension, the file might become unusable.
Are you sure you want to change it?
Yes
No



The file name index.html is HTML standard, it's important to have your main page under that name.
Now, when you have created the folder and the index.html file, it's time to launch your IDE and write some code.
For this, click on the VS Code IDE icon

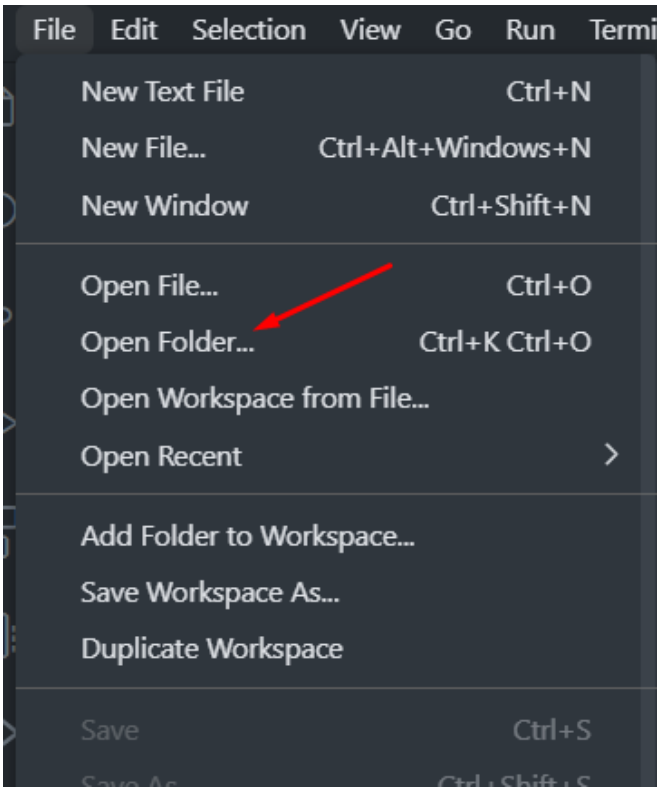


After launching, you will see a blank page in your code editor. You have to enter your folder through the IDE.

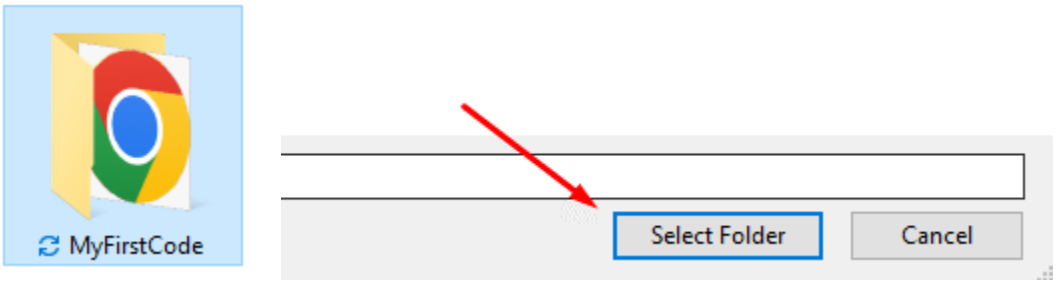


Click on the top left File as on image above

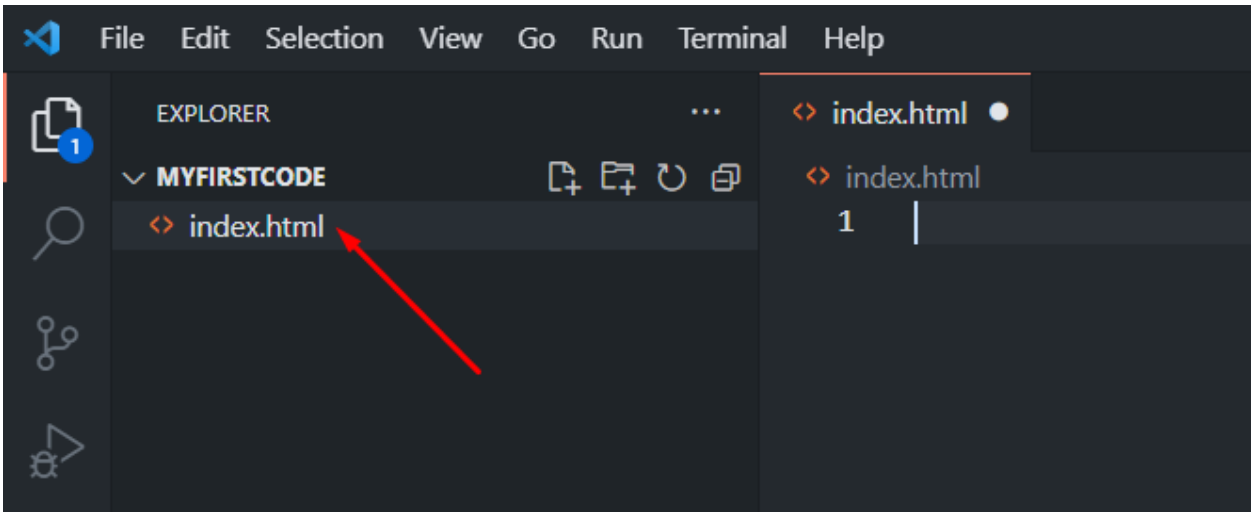
Then click on Open Folder



Find and click on your folder and press Select Folder



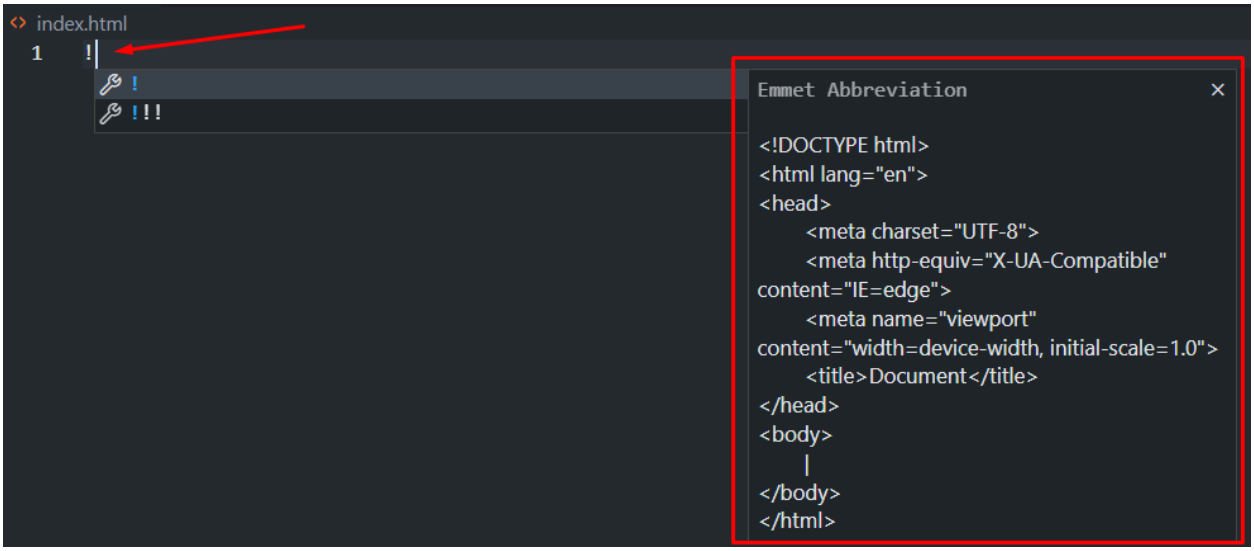
Now you have to see your folder and file inside the IDE like on image below.



Click on index.html file, to tell the code editor, that you want to work with that file. After, we have to write the HTML standard boilerplate code. We can do it in two ways: write it manually, which is time consuming,

```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>

</body>
</html>
```

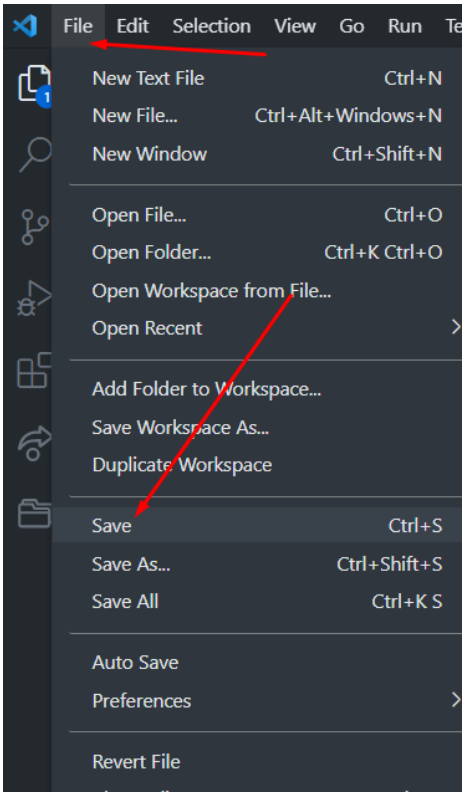
or to do it with the help of inbuilt extension Emmet, that helps us to have the HTML boilerplate in 2 clicks. Press exclamation mark !



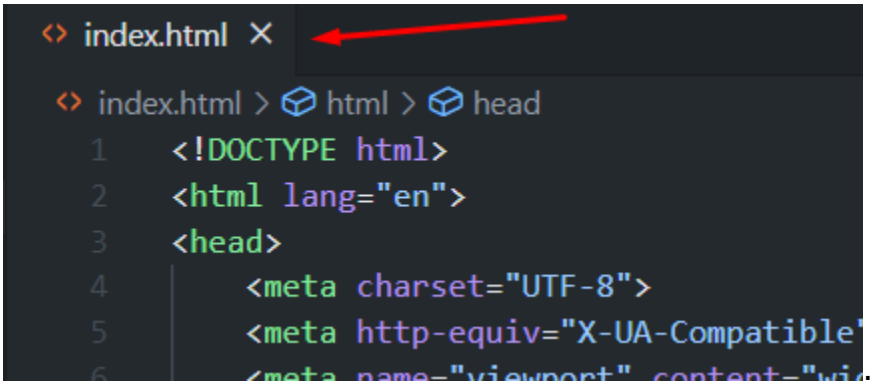
and click tab or enter. You will get the boilerplate code.
Inside the body tag you can write your first code. Type this sample expression:
<h1>I learn html<h1>

When you see the white rounded sign, as shown on the image above, it means, that your recent changes haven't been saved.
Now you have to save your changes in the code, in order to see the result in the browser. You can do it in two ways:

1. with ctrl + s (***recommended***).
2. press at the top left File button then the Save button.

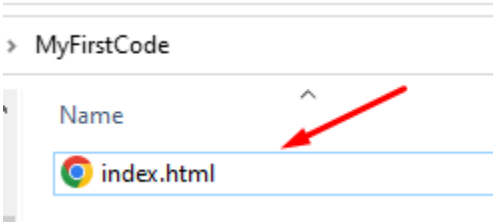


After saving the file, the white rounded sign will become a cross sign, which means, that your recent changes have been saved, like on the image below:



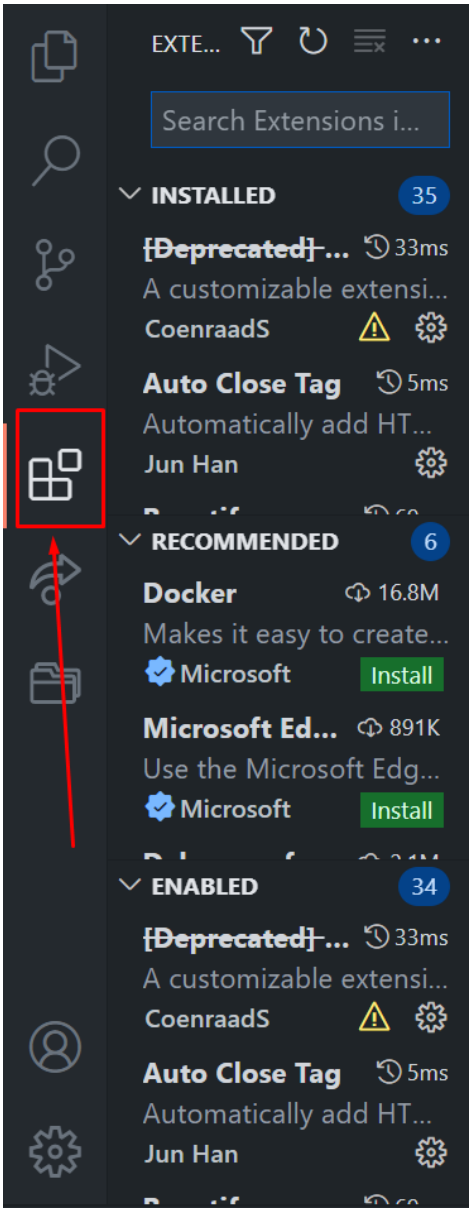
To see the results of your changes you can do it in two popular ways:

1. Go to your local folder and double-click on the index.html file.

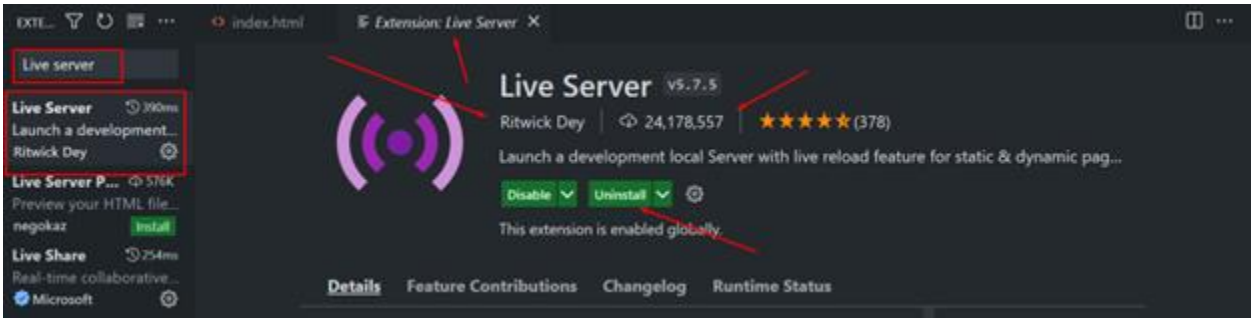


   After each change in your code, you will have to refresh your browser window to see the result.

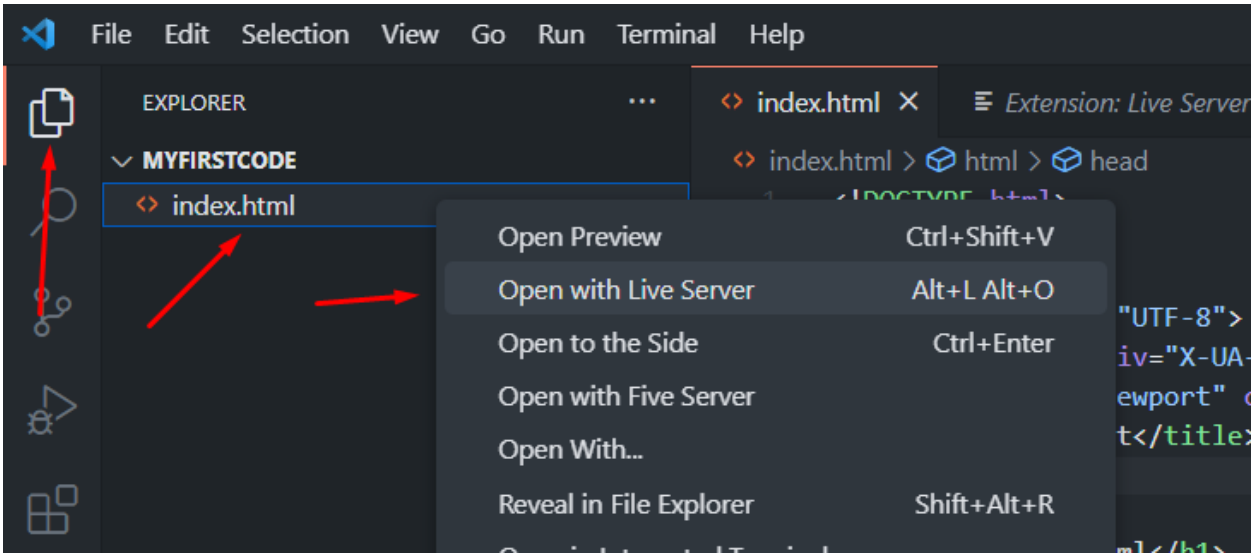
2. The second and ***recommended*** way to see your change results, you have to search for Visual Studio Code extension called **Live Server** and install it.
   For that purpose, click on the extensions icon in VS Code editor like in photo below:



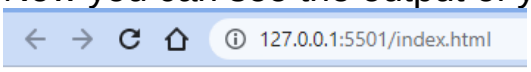
Then type in “Live server” in the search bar and click install, as in screenshots below:



In my case it is already installed.
After you need to launch your server. For that, inside code editor press right click on the index.html file like on image, and the live server will open a new window in your preferred browser, with your file results.



Now you can see the output of your first code below:



I learn html

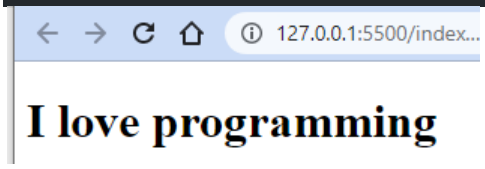
After you have learned, how to launch the server and check your saved changes in the code, let’s jump to the next sections.

2.Tags, Elements and Attributes

Heading tags

HTML headings are titles or subtitles, that you want to display on a webpage.
There are 6 types of heading tags in HTML: h1, h2, h3, h4, h5, h6. Heading tags are important in HTML for having structured and SEO-friendly page. Search engines use the headings to index the structure and content of your web pages.
From h1 to h6 heading tags have decreasing default styles (but also you can set your custom ones). You can use them according to importance. Let's say you use <h1> for your most important headings and <h2> to <h6> for your less important ones. It's better to use them by priority. For example, you can have the main title of your page in <h1>, subtitles in <h2>, sections' titles in <h3>, subsections' titles in <h4>, contact form's heading in <h5> and footer links' titles in <h6>.
Now, inside the body content (after openning <body> tag) write simple heading code <h1>I love programming</h1>, save your changes, pressing **ctrl + s** and see the result in the browser (by double-clicking on your index.html file in the folder, or launching the server).

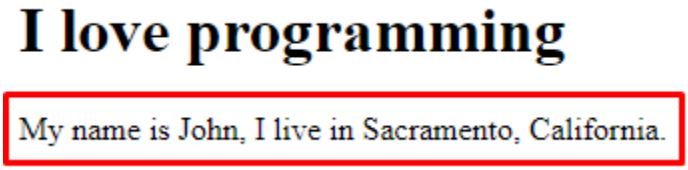
```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>

<body>
    <h1>I love programming</h1>
</body>

</html>
```



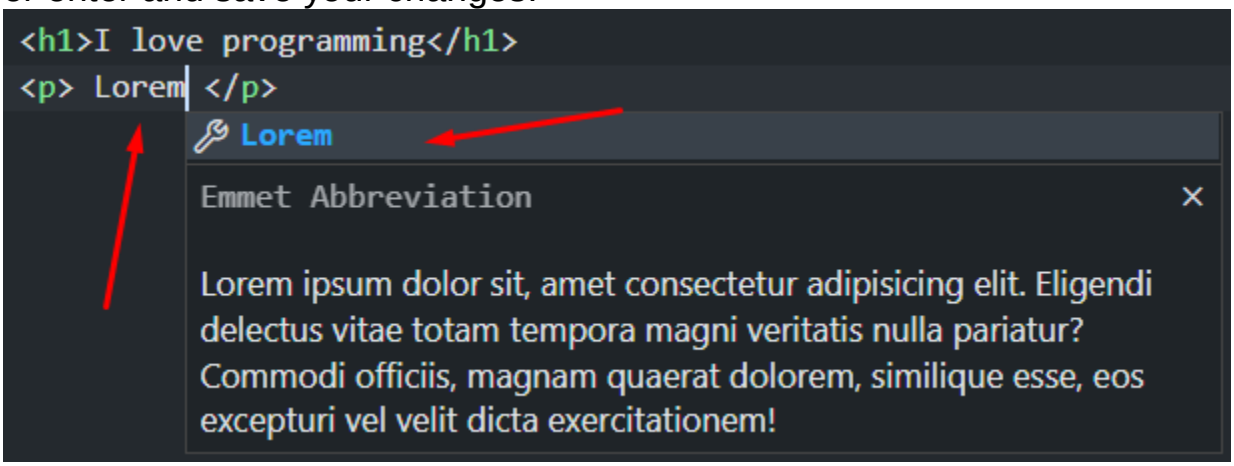
You have to see this output in your browser.

Paragraph tags

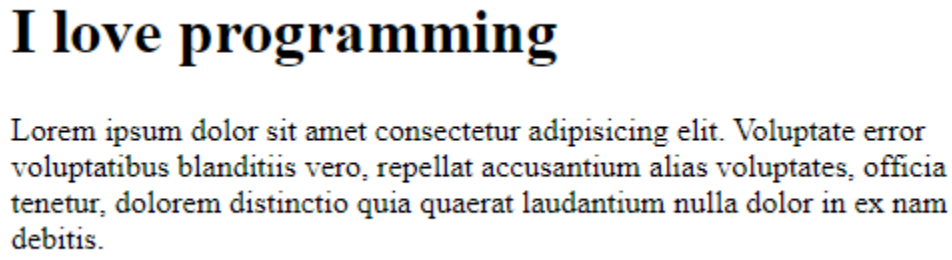
The HTML tag defines a paragraph. It always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.
Paragraphs are used to display the simple text on your website (story, description, large content, etc).
For example, type My name is John, I live in Sacramento, California. with openning and closing tags. **Save. See** in the browser.



the browser output.
For creating simple default texts to see your page's layout, you can use **lorem ipsum** practice. Just type in your IDE lorem, and the Emmet extension will automatically show you the probable option for you text, then press tab or enter and save your changes.



The browser output will be:



HTML tags and elements: what's the difference

HTML tags are the markup language you use. These are, generally the **start tag <>** (or **openning tag**, enclosed in angle brackets), and the **end tag </>** (or **closing tag**, enclosed in angle bracket with a **slash mark**).
The HTML elements include the content inside the openning and closing tags. They are **defined by a start tag, some content, and an end tag.**
So, let's see the difference between tags and element with code example: ** I am learning HTML .** The tags here are two p tags . While the whole HTML element is everything from the start tag to the end tag.
HTML elements can be nested (this means that elements can contain other elements).
All HTML documents consist of nested HTML elements.
The following example contains four HTML elements (<html>, <body>, <h1> and):

```
<!DOCTYPE html>
<html>
 <body>
  <h1>I am John</h1>
  <p>I learn programming</p>
 </body>
</html>
```

The <html> element is the root element and it defines the whole HTML document.
It has a start tag <html> and an end tag </html>.
Then, inside the <html> element there is a <body> element:

```
<body>
  <h1>I am John</h1>
  <p>I learn programming</p>
</body>
```

The <body> element defines the document's body.
It has a start tag <body> and an end tag </body>.
Then, inside the <body> element there are two other nested elements: <h1> and :

```
<h1>I am John</h1>
<p>I learn programming</p>
```

The <h1> element defines a heading.
It has a start tag <h1> and an end tag </h1>:

```
<h1>I am John</h1>
```

The element defines a paragraph.
It has a start tag and an end tag :

```
<p>I learn programming</p>
```

Empty HTML Elements

HTML elements with no content are called empty elements.
The
 tag defines a line break, and is an empty element without a closing tag.
The <hr> tag defines a thematic break in an HTML page and is most often displayed as a horizontal rule. It is used to separate content (or define a change) in an HTML page. The <hr> tag is an empty tag, which means, that it has no end tag.

HTML attributes

HTML attributes provide additional information about HTML elements. Attributes are name-value pairs, which are separated by the equal sign. They usually follow this syntax: <tag attribute='value of attribute'>content</tag>. Attributes are always specified in **the start tag.**
Let's see some sample codes:
The title Attribute:
The title attribute defines some extra information about an element.
The value of the title attribute will be displayed as a tooltip when you mouse over the element: <h2 **title="I'm a header"**>The title Attribute</h2>
The href Attribute:
The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to: <a **href="https://www.edgarsblog.com"**>Visit my Blog.
You will learn more about links in our HTML Links chapter.

The lang Attribute:
You should always include the lang attribute inside the <html> tag, to declare the language of the Web page. This is meant to assist search engines and browsers.
The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

Useful snippets

A clean and structured code makes it easier for you and others to read and understand your code.
Here are some guidelines and tips for creating good HTML code:

- Always declare the document type as the first line in your document - **<!DOCTYPE html>**
- **Use lowercase tag names.** The HTML standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.
- **Never skip the end </>** tag (<p>This is a paragraph). Some HTML elements will display correctly, even if you forget the end tag, however **never rely on this!** It may cause unexpected results and errors, if you forget the end tag.

3. Head section:
in-built title and meta tags

The HTML <head> element is a container for the following elements: <title>, <style>, <meta>, <link>, <script>, and <base>.
It is used for metadata and is placed between the <html> and the <body> tags.
HTML metadata is data about the HTML document. Metadata is not displayed.
Typically it defines the document title, character set, styles, scripts, and other meta information.
A good SEO directly depends on having properly written metadata (we will learn later in SEO chapter).
Nowadays, due to such extensions like Emmet, most IDE's have in-built metadata. And we shall discuss their meaning one by one:

```
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

The HTML Charset Attribute

<meta charset="UTF-8">
To display an HTML page correctly, a web browser must know the character set used in the page.
This is specified in the <meta> tag:
This attribute declares the page's character encoding. It must contain a standard IANA MIME name for character encodings. Although the standard doesn't request a specific encoding, it suggests, to use UTF-8.

HTML http-equiv Attribute

<meta http-equiv="X-UA-Compatible" content="IE=edge">
The http-equiv attribute provides an HTTP header for the information/value of the content attribute.
The http-equiv attribute can be used to simulate an HTTP response header.

Setting the Viewport

The browser's viewport is the area of the window in which web content can be seen. This is often not the same size as the rendered page, in which case the browser provides scrollbars for the user to scroll around and access all the content. The viewport gives the browser instructions on how to control the page's dimensions and scaling.
The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

The Title

The <title> HTML element defines the document's title, that is shown in a browser's title bar or a page's tab.
<title>Document</title>
The usage of content inside the title tags can have direct positive impact on search engines and help related users to find your website easier. A longer, descriptive title performs better than short one. The content of the title is one of the components used by search engine algorithms to decide the order in which to list pages in search results. It is one of the initial elements for your website, to appear in search results (for example on Google). Try to make sure your titles are as unique as possible within your own site. For SEO It would be better, if your title’s and main h1 first heading text’s contents will be the same.

4. Formatting, quotes, comments

Formatting elements

Formatting elements display the text in special design types:
In HTML you can use the following elements:

- - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

HTML and Elements

The HTML element defines bold text, without any extra importance.
Whether element also defines bold, but with importance.
For example:
This is a bold text, or This is a bold text.

HTML <i> and Elements

These elements display text in italic styles, for example if you want to define phrases or names to differentiate from the rest of the content.

HTML <small> Element

The HTML <small> element defines smaller text from the rest of the content

HTML <mark> Element

The HTML <mark> element defines text that should be marked or highlighted:

HTML <del> Element

The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:

HTML <ins> Element

The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text:

HTML <sub> Element

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H2O:

HTML <sup> Element

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1]:

Quote elements

In this chapter we will go through the <blockquote>,<q>, <abbr>, <address>, <cite>, and <bdo> HTML elements.

Here is a quote from Edgarsblog website:
<blockquote cite="http://www.edgarsblog.com">
I run this blog to help freelancers.
</blockquote>

HTML <q> for Short Quotations

The HTML <q> tag defines a short quotation.
Browsers normally insert quotation marks around the quotation.
My goal is to: <q>Build a future where people live in harmony together.</q>

HTML <abbr> for Abbreviations

The HTML <abbr> tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

The <abbr title=“HyperText Markup Language">WHO</abbr> was founded in 1948.

HTML <address> for Contact Information

The contact information can be an email address, URL, physical address, phone number, social media handle, etc.
The text in the <address> element usually renders in *italic,* and browsers will always add a line break before and after the <address> element.

<address>
Written by John Doe.
Visit us at:
Example.com
Barley street 77
USA
</address>

HTML <cite> for Work Title

The HTML <cite> tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

<cite>The Scream</cite> by Unknown artist. Painted in 1677.

HTML <bdo> for Bi-Directional Override

BDO stands for Bi-Directional Override.
The HTML <bdo> tag is used to override the current text direction:
<bdo dir="rtl">This line will be written from right to left</bdo>
for example in Arabic language

Comment elements

You can add comments to your HTML source by using the following syntax:
<!-- Write your comments here -->
Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

Note: Comments are not displayed by the browser, but they can help document your HTML source code.
With comments you can place notifications and reminders in your HTML code:

<!-- This is a comment -->

This is a paragraph.

<!-- Remember to add more information here -->

Comments can be used to hide content.
This can be helpful if you hide content temporarily:

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors.

5. HTML Lists

Lists allow developers to group items in a structured list :
there are several types of lists:

- ul
- ol
- dl
- nested lists

Unordered lists: ul (unnumbered)

ul is an abbreviation, that stands for unordered lists.

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag (li stands for - list item).
The list items will be marked with bullets (small black circles) by default:

An unordered HTML list code
```
<ul>
  <li>Coffee</li>
```

```
  <li>Tea</li>
</ul>
```

The browser output:

- Coffee
- Tea

Ordered lists: ol (numbered)

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.
The list items will be marked with numbers by default:

An ordered HTML list code:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

The browser output:

1. Coffee
2. Tea
3. Milk

Description Lists: dl

HTML also supports description lists.
A description list is a list of terms, with a description of each term.
The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
```

```
  <dd>- white cold drink</dd>
</dl>
```

A Description List

Coffee
- black hot drink
Milk
- white cold drink

6. Images

HTML Images Syntax

The HTML <img> tag is used to embed an image in a web page.
Images are not technically inserted into a web page. Images are linked to web pages. The <img> tag creates a holding space for the referenced image.
The <img> tag is empty, it contains attributes only and does not have a closing tag.
The <img> tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

The src (stands for source) attribute

The required src attribute specifies the path (URL) to the image. When the webs page loads, the browser gets the image from a web server and shows it on the page. That's why, it is important to be sure, that the image link path is set properly, otherwise it will show a broken link icon on the browser. The broken link icon and the alt text are shown, if the browser cannot find the image.

For example: you have the "my_image.png" in your main folder. In this case it's path will be <img src="my_image.png" alt="personal image">

Here the source tag tell the browser the name (source) of your image.

If you have your images in a sub-folder, you must include it's name in the src attribute. For example your sub-folder's name is "images", then the source will be <img src="./images/my_image.png" alt="personal image">

So, as you can see, you always have to show the proper path to the image, otherwise it won't work.

The alt attribute

The required alt attribute provides an alternate text for an image, if for some reason your image is not shown (because of slow connection, an error in the src attribute, or if the user uses a screen reader), it will show the user what is your image about. It is also a great tool for having a good SEO for your website. You can put in the alt attribute the name of your image and it's a good practice to put also the word "image". For example, if you load a car image on your web page, you can write in the alt attribute "car image".

Image size: width and height

You can use the style attribute to specify the width and height of an image.

<img src="my_image.png" alt="personal image"
style="width:500px;height:600px;">

Alternatively, you can use the width and height attributes:

<img src="my_image.png" alt="personal image" width="500" height="600">

The width and height attributes always define the width and height of the image in pixels.

Note: Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

Images on another server / website

Some web sites point to an image on another server.

To point to an image on another server, you must specify an absolute (full) URL in the src attribute:

```
<img src=" https://www.edgarsblog.com/wp-content/uploads/2023/02/workWith.jpg" alt="agreement image">
```

Notes on external images: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images: they can suddenly be removed or changed.

Animated Images

HTML allows animated GIFs:

```
<img src="programming.gif" alt="Computer Man" style="width:48px;height:48px;">
```

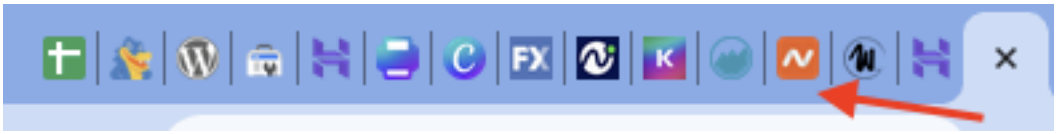
Image as a Link

To use an image as a link, put the <img> tag inside the <a> tag:

```
<a href="https://www.edgarsblog.com">
  <img src="sample.png" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

HTML favicon image

A favicon is a small image displayed next to the page title in the browser tab, like these:



To add a favicon to your website, either save your favicon image to the root directory of your webserver, or create a folder in the root directory called images and save your favicon image in this folder. A common name for a favicon image is "favicon.ico".

Next, add a <link> element to your "index.html" file, after the <title> element, like this:

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

So, the favicon image, has to be inside the link element.

7. HTML links (hyperlinks)

HTML links are called hyperlinks. You can click on a link and jump to another document. When you move the mouse over a link, the mouse arrow will turn into a little hand icon.

Text links: websites, social media

The HTML <a> tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

the most important attribute of the <a> element is the href attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

Visit Edgarsblog website !

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Tip: Links can of course be styled with CSS, to get another look!

Link are a crucial part in web development, that you will use very often. If you want the users to go to you Instagram, Facebook or other social media links or personal website, you will use the
My Instagram account tag.

Image as a link

To use an image as a link, just put the <img> tag inside the <a> tag:

<img src="website.png" alt="website
image" style="width:42px;height:42px;">

Don't forget to close the image with a tag .

The target attribute

By default, the linked page will be displayed in the current browser window.
To change this, you must specify another target for the link.
Here comes the target attribute, which specifies where to open the linked document.
The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

For example: Visit my website!

Email, phone, messenger links

One of the advantages of using HTML is the easy functionality to make a fast call to action links to you email address, phone number, Telegram, WhatsApp or other messengers directly from you webpage. This opportunity is widely used and loved by everyone nowadays.

Use mailto: inside the href attribute to create a link that opens the user's email program (to let them send a new email):
Send email

Use tel: inside the href attribute to create a link, that users can make a phone call with:
Call me

Use https://wa.me/ + the WhatsApp number inside the href attribute to create a link, that users can jump directly into the WhatsApp messenger and send a message:
Send me a WhatsApp message

Use https://telegram.me/ + the Telegram number inside the href attribute to create a link, that users can jump directly into the Telegram messenger and send a message:
Connect on Telegram

Button as a Link

As a link, you can use the buttons as well. <button></button> tag is a great helper, when it comes to creating a CTA (call to action) links with button appearance, such as “Learn more”, “Get in touch”, etc.
For this purpose, again, you have to include the button inside the a tag.

```
<a  href=”https://www.edgarsblog.com/blog/”>
<button>My blog</button>
</a>
```

The Download attribute

For having a downloadable content on your website (for example PDF files or images) the HTML download attribute is a great helper.
You have to specify the path of your file in your folder or subfolder inside the href attribute, after put the download attribute. For example, you have a PDF file in the “documents” subfolder inside your main folder:

```
<a href="./documents/myFile.pdf" download>My document</a>
```

In this way, the browser understands, that clicking on the download link, you want to download a file, attached to that link.

8. CSS styles: inline, internal and external

For styling your web pages in programming you will need CSS styles (Cascading Style Sheets).

The HTML style attribute is used to add styles to an element, such as color, font, size, and more.
Setting the style of an HTML element, can be done with the style attribute.
You can style your HTML files in 3 different ways:

1. Inline
2. Internal
3. External

Inline CSS

In this example, we will change the color and font size of a paragraph element with the help of the “style” attribute, using **the inline method.**

```
<p style="color: green; font-size: 14px;">
        This is some text by inline style
</p>
```

Output: This is some text by inline style
Here, the “style” attribute has been added to the <p> tag and includes two CSS rules: “color” and “font-size”. The “color” rule is set to “green” and the “font-size” rule is set to “14px”. These rules will only apply to the specific paragraph element in which the “style” attribute has been added.

Internal CSS

Is also known as embedded CSS. This is a method of adding CSS rules directly to the head section of an HTML document. It involves using the <style> element to enclose the CSS code within the HTML document itself. Internal CSS is placed within the <head> section of the HTML document, typically after the document’s title and before any other content.

```
<!DOCTYPE html>
<html>
<head>

<title> Internal CSS </title>

<style>

h1 {
color: blue;
font-size: 24px;
font-weight: bold;
}
```

```
p {
color: green;
font-size: 16px;
}
</style>
</head>
<body>
<h1>Edgarsblog</h1>
<p>Some text</p>
</body>
</html>
```

Output:

Edgarsblog

Some text

in this example, the <style> element is used to enclose the CSS rules that apply to the h1 and p elements. Any styling applied with internal CSS is specific to that HTML document and cannot be used on other pages or websites.

Here the color, size, font-weight are CSS properties. The 24px, green, blue, 14px – are CSS values.

CSS is a large separate world in coding, which is very exciting and useful to know. You can learn more about CSS in my another related eBook (for inquiries, please contact me on www.edgarsblog.com .

External CSS

Is used to place CSS code in a separate file and link to the HTML document (in your case – index.html). To use external CSS, create a separate file with the .css file extension (style.css), that contains your CSS rules. You can link this file to your HTML document using the “link” tag in the head section of your HTML document.

```
<head>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
```

In this book we won't pass the separate CSS. But you can learn more about CSS in another related eBook (for inquiries, please contact me on www.edgarsblog.com .

9. HTML Tables

What are tables, and why we need them?

HTML tables allow web developers to arrange data into rows and columns.

An HTML table is defined with the opening and closing <table></table> tags.

Table row (tr)

Each table row is defined with the <tr></tr> tags (stands for *table row*). You can have as many rows as you like in a table; just make sure that the number of cells are the same in each row.
Note: There are times when a row can have less or more cells than another. You will learn about that later.

Table head (th)

A table header is defined with the <th></th> tags (stands for *table head*) . By default, the text in <th> element is bold and centered, but you can change that with CSS.

Table data (td)

A table data/cell is defined with the <td></td> tags (stands for *table data*). Everything between <td> and </td> is the content of the table cell.

Note: a table data can contain all sorts of HTML elements: text, images, lists, links, other tables, etc

So, inside the Table, the table rows (tr) can contain headers (th) and content (td).

```
<table>
<tr>
<th>Book Name</th>
<th>Author Name</th>
</tr>

<tr>
<td>The Book Thief</td>
<td>Markus Zusak</td>
</tr>
</table>
```

The output:

Book Name	Author Name
The Book Thief	Markus Zusak

Table Borders

HTML tables can have borders of different styles and shapes. A border is set using the CSS border property. If you do not specify a border for the table, it will be displayed without borders.

In this example we add an internal CSS styles for our table border, by setting borders for table, row and data properties. You have separate the properties by commas. Or set them separate.

```
<style>
table, th, td {
  border: 1px solid black;
}
</style>
```

The output:

Book Name	Author Name
The Book Thief	Markus Zusak

Collapsed Table Borders.
To avoid having double borders like in the example above, set the CSS border-collapse property to collapse.

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

This will make the borders collapse into a single border:
The output:

Book Name	Author Name
The Book Thief	Markus Zusak

With the border-style property, you can set the appearance of the border.

The following values are allowed:

- dotted 
- dashed 
- solid 
- double 
- groove
- ridge 

- inset 
- outset
- none
- hidden

```
th, td {
  border-style: dotted;
}
```

The output:

Book Name	Author Name
The Book Thief	Markus Zusak

Table styling

You can style your tables, giving them colors, width, height, background, sizes, padding, margin, radius. All this you can learn in the CSS course. Now we will see, how to add some spacing (cell padding) inside the tables, to make them look nicer and user-friendly.
Cell padding specifies the space between the cell content and its borders. If we do not specify a padding, the table cells will be displayed without padding.
If we want to give some space to our headers and content inside the table, we have to set a padding value for the td (table data) property in CSS styling.

```
th, td {
padding: 20px;
}
```

The output will look like this:

Book Name	Author Name
The Book Thief	Markus Zusak

Colspan and rowspan

As you know already from this book, HTML tables can have headers for each column or row, or for many columns/rows. There can also be headers for **multiple columns:**
You can have a header, that spans over two or more columns. As in the example below:

Full name		Age
John	Smith	30
Eva	Harrison	25

Colspan

For this purpose, we use the colspan attribute, which stands for column span. You concatenate columns due to this property, and how many columns you want to join together depends on the value, like this:
<th colspan="2">Name</th>

So, here we tell the browser with our code, that we have one joint header for 2 columns – colspan = “2”.

```
<table>
  <tr>
    <th colspan="2">Full name</th>
    <th>Age</th>
  </tr>
  <tr>
```

```
    <td>John</td>
    <td>Smith</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Eva</td>
    <td>Harrison</td>
    <td>25</td>
  </tr>
</table>
```

The output:

Full name		Age
John	Smith	30
Eva	Harrison	25

Note: The value of the colspan attribute represents the number of columns to span.

Rowspan

To make a cell span over multiple rows, use the rowspan attribute:

```
<table>
  <tr>
        <th rowspan="2">User's data</th>
        <th>Name</th>
        <th>Age</th>
  </tr>

  <tr>
    <td>John</td>
    <td>30</td>
```

```
  </tr>
</table>
```
The output:

User's data	Name	Age
	John	30

Note: The value of the rowspan attribute represents the number of rows to span.

10. HTML Form

An HTML form is used to collect user data through different types of input elements, such as: text fields, email, number, passwords checkboxes, radio buttons, submit buttons, etc. The form element consists of opening and closing <form></form> tags.
It is a perfect method for creating contact forms for websites, or different forms for users.

For example, a user wants to buy a shirt online, so he/she has to first enter their shipping address in the address form and then add their payment details in the payment form to place an order.

Forms are created by placing form elements - input fields within paragraphs, preformatted text, lists and tables.

Note: The form itself is not visible. Also note that the default width of an input field is 20 characters.

<form> <!--form elements--> </form>

Form elements

The HTML <form> element is used to create an HTML form for user input:
These are the following HTML <form> elements:

<label>: It defines label for <form> elements.
<input>: It is used to get input data from the form in various types such as text, password, email, etc by changing its type.
<button>: It defines a clickable button to control other elements or execute a functionality.
<select>: It is used to create a drop-down list.
<textarea>: It is used to get input long text content.
<fieldset>: It is used to draw a box around other form elements and group the related data.
<legend>: It defines a caption for fieldset elements.
<datalist>: It is used to specify pre-defined list options for input controls.
<output>: It displays the output of performed calculations.
<option>: It is used to define options in a drop-down list.
<optgroup>: It is used to define group-related options in a drop-down list.

The input Element

The HTML <input> element is the most used form element.
An <input> element can be displayed in many ways, depending on the type attribute. Here are the different input types you can use in HTML:

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">

- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">
- <input type="week">

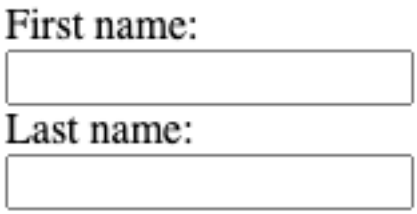
The default value of the type attribute is "text".

<input type="text"> defines a **single-line text input field**:

Example

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

The output:



Besides text type, there are also several popular input types:

<input type="radio"> Displays a radio button (for selecting one of many choices)

<input type="checkbox"> Displays a checkbox (for selecting zero or more of many choices)

<input type="submit"> Displays a submit button (for submitting the form)

<input type="button"> Displays a clickable button

The label element

Notice the use of the <label> element in the code example above.
The <label> tag defines a label for many form elements.

The <label> element helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

The Submit Button

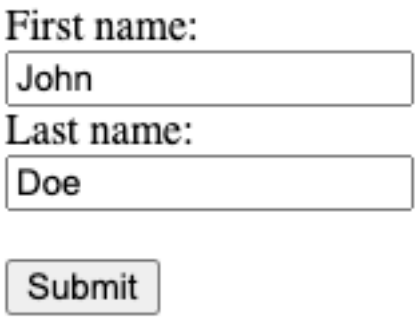
The <input type="submit"> defines a button for submitting the form data to a form-handler. The form-handler is typically a file on the server with a script for processing input data and is specified in the form's action attribute.
In the example below, the form data is sent to a file called "action_page.php". This file contains a server-side script, that handles the form data:

Example
A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
```

```
  <input type="submit" value="Submit">
</form>
```

The output:



You can practice more with forms and create your own professional contact form, without having deep coding skills, learning more from this website: https://formsubmit.co/ .

The select element

The <select> element defines a drop-down list:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <option> element defines an option that can be selected.

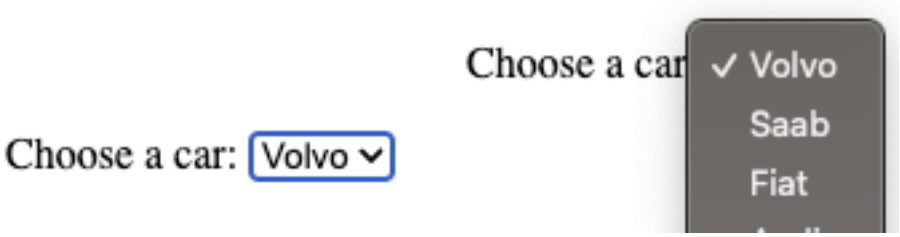
By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the selected attribute to the option:

Example

```
<option value="fiat" selected>Fiat</option>
```

The output:



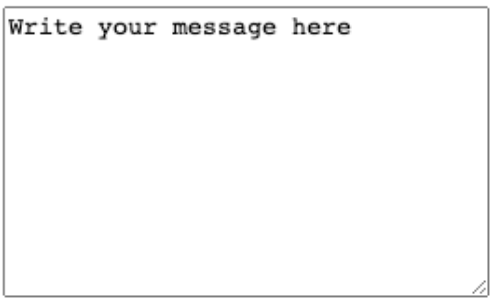
The textarea element

The <textarea> element defines a multi-line input field (a text area). It's a great way to fill in some long text:

Example

```
<textarea name="message" rows="10" cols="30">
Write your message here
</textarea>
```

The rows attribute specifies the visible number of lines in a text area.
The cols attribute specifies the visible width of a text area.
This is how the HTML code above will be displayed in a browser:



You can also define the size of the text area by using CSS:

Example

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

11. HTML File Paths

A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files, like:

- Web pages
- Images
- Style sheets
- Programming languages' code files

There are 2 main types of file paths in HTML: absolute and relative.

Absolute File Paths

An absolute file path is the full URL to a file:

```
<img src="https://www.edgarsblog.com/wp-content/uploads/2023/02/workWith.jpg" alt="agreement image">
```

Relative File Paths

A relative file path points to a file relative to the current page.

In the following example, the file path points to a file in the images sub-folder located at the root of the current web:

```
<img src="/images/web.png" alt="website image">
```

In the following example, the file path points to a file in the images sub-folder located in the current folder:

```
<img src="images/web.png" alt="website image">
```

In the following example, the file path points to a file in the images folder located in the folder one level up from the current folder:

<img src="../images/web.png" alt="website image">

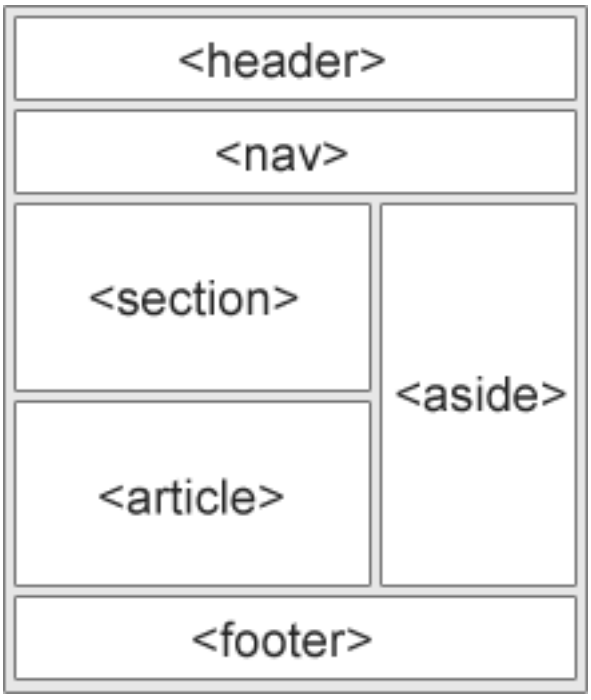
It is best practice to use relative file paths (if possible).

When using relative file paths, your web pages will not be bound to your current base URL. All links will work on your own computer (localhost) as well as on your current public domain and your future public domains.

12. Website layout elements

HTML has several semantic elements, that define the different parts of a web page:

- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section



It is widely used when the header or the navbar are at the top.

Then come the main: section, aside or article elements.
After comes the footer at the end.
We will go one by one over this elements.
After you understand the purpose of their use, we will create locally a simple web page, then we will upload your files to the hosting.
After you will have a website online.
Also I will show you how to buy domain, hosting plans for your future use.

Navbar:
This element is used to have logo, links, call to action buttons (such as call us, ask a quote and so on).

Header: this element is used for demonstrating the main purpose or value of the website. The most famous types of header are 2:
Hero section header (with media, title, text, button) and one line header: with text, button and sometimes background image.

The main part (element) of each website includes aside, article, section elements.

They give you a great opportunity to count how many sections you have (for example if you want to calculate your web deisgn project cost for your client, sections are really helpful.

Footer: Is the place where you can implement your social media follow links, privacy policy, copyright and other useful links.

13. The landing page. Building your first hand-coded and real website

We will have here navbar, one line header: text and button, main section and footer.
In navbar we shall have a logo from Flaticon.com and CTR (call to action) link.
In One line header we will have: heading title, description text and button link / CTR (call to action).

In the main part we will have section elements - with image, heading and paragraph texts.
In footer we will have the copyright part.
The website will be deployed on free and powerful hosting: Netlify.

So, let's start:
At first, as we have already mentioned at the beginning of this e-book, set up your code editor, with local folder, where you will have a simple index.html file.
After, type the HTML boilerplate code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Let's say we name our website - Portfolio. Inside the title tags type "Portfolio", in order to show on the browser window tab your website's name, while someone hovers on it.

```
<title>Portfolio</title>
```

The Layout

Inside the body tags will be our website's main code.
Let's have at first navbar, then header, then main content, inside it sections and at the end only the footer.
Inside nav element we have logo image and Contact links.

```
<nav>
```

```
<img src="" alt="logo image">
<ul>
   <li><a href="">Contact</a></li>
</ul>
   </nav>
```

Without styling it looks simply, so let's give a little style to it. As we didn't learn CSS (main styling language in coding), we will write some inline code in the same index.html file.

The logo and the ul element by default will be under each other, cause html elements take the whole row by default. To put them all 3 together, we have to use such popular flexbox style, as – display: flex; . It will put our elements together on one row. For that, in the parent nav element we write our style, so that it will stack the flex items horizontally (from left to right). It will look like this:

```
nav style="display: flex;">
```

Now, we have to do the same inside ul elements itself.

```
<ul style="display: flex;">
```

After, you will see the ul elements with dotes at the start of each link, for removing them, write the list-style: none; code inside the each li element:

```
<li style="list-style: none;">
```

After, you will see, that anchor links are underlined (by default styles), you can remove that by using text-decoration: none; style.

```
<a href="" style="text-decoration: none;">
```

But, as we can see the logo and the links are not on equal lines. For making them equal we must use align-items; center; property inside nav element.

```
<nav style="display: flex; align-items: center;">
```

Now, let's make logo and the links at the start and end of the webpage accordingly. For that we use justify-content: space-between; property.

Now we have to make the layout a little nicer, by adding a space inside from both: left and right sides, aligning center and giving a maximum width. For that, we can use the padding (spaces), margin-auto (align center) and max-width (set maximum width) properties (about all the style properties you can learn more deeply in my eBook about CSS styling).

```
<body style="padding: 0px 15px; max-width: 800px; margin: auto">
```

After, let's add the rest of the elements and discuss them one by one:

```
<main>
        <section>
<h2>Suubtitle 1</h2>
<img src="" alt="">
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ex hic quis quasi repellat magni libero et doloribus! Minus, nulla enim sit consequatur nobis quod, temporibus, mollitia doloremque deserunt nesciunt sed.</p>
        </section>
        <section>
            <h3>Subtitle 2</h3>
            <img src="" alt="">
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vel repellendus ipsa dignissimos esse quis, eveniet, molestiae error necessitatibus ipsum, eius unde. Modi deserunt cum similique blanditiis, dolores suscipit non voluptas?</p>

        </section>
    </main>
    <footer>
<p>Copyright 2023. All rights reserved.</p>
    </footer>
```

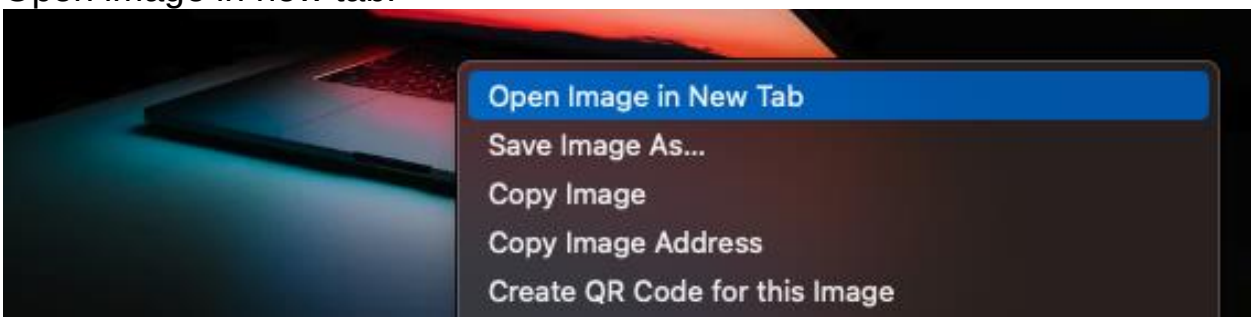
As you can see, we have the <main></main> element, inside it we have 2 <section></section> elements. And the last ones, have inside heading (h2 and h3), image <img> and paragraph elements.

After, you can see the footer element. But before, let's go through the main layout and it's content.

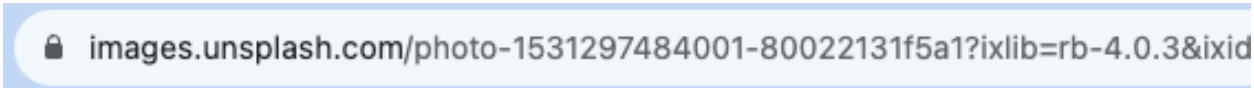
Inside the heading tags we have Subtitle 1 and Subtitle 2, which you can also change to 'About Me' and 'My Skills' heading texts. It's just an example for you to see the difference in default size of heading tags. But of course you can give them your custom sizes with font-size: ..px property (for example set the value to 18px).

Then, we have the image element. As we have learned before, you can have a local image in you folder and use it's relative path, or external link (absolute path), which we will use now.

For example, go to the https://unsplash.com/, find some free image, hover on it, press right mouse click on the image you want the path to use, select Open image in new tab.



After, copy the image path in the browser window, in my case it's https://images.unsplash.com/photo-1531297484001-80022131f5a1?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=1120&q=80



And paste it in the image source. Also add an alt tag, for example if it's a laptop image, you can write alt="laptop image"

```
<img src="https://images.unsplash.com/photo-1531297484001-80022131f5a1?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=1120&q=80" alt="laptop image">
```

After saving and refreshing your page, you have to see the result. But for making nicer your webpage, it will be good to give a width to your image. In my case, I gave 250px – width="400px".

```
<img src="https://images.unsplash.com/photo-1531297484001-80022131f5a1?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3
```

```
D%3D&auto=format&fit=crop&w=1120&q=80" width="250px" alt="laptop image" >
```

Now we have the paragraph part, where you can write a small biography or other description text, depends on the topic of your webpage.
In my case, I have just put Lorem Ipsum sample text.
The same with title, image and paragraph you can try in the second section.

After we have the <footer></footer> element.
Inside we have a simple copyright text with element.

```
Copyright 2023. All rights reserved.
```

For having your footer at the bottom, it's useful to add a minimum height to your main content, otherwise, if your content is short the footer won't be at the bottom by default.
Here we have added:

```
<main style="min-height: 600px;">
```

Now let's add the logo image from https://www.flaticon.com/
I have downloaded into my coding folder an image with 64px size (you can choose on Flaticon th sizes and download for free) and will use it's path locally.

```
<img src="coding.png" alt="logo image">
```

Now let's add call and email functionality to our CTR (links).
Under the contact link let's add phone call functionality:
HTML gives you a great opportunity to do it with the code: tel:+123456789 your desired phone number inside the href tag.

```
<li style="list-style: none;"><a href="tel:+123456789" style="text-decoration: none;">Contact</a></li>
```

Now let's add email functionality to the "Email me" button:
For this you have to use mailto:youremail@gmail.com (put the email you want inside the href tag).

```
<a href="mailto:youremail@gmail.com"><button>Email me</button></a>
```

At the end, let's have a little space between for two sections. For that, one of the simplest ways to do it, is using
 break tag before and after the first section.

```
<br>
        <section>
<h2>Subtitle 1</h2>
<img src="https://images.unsplash.com/photo-1531297484001-80022131f5a1?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=1120&q=80" width="250px" alt="laptop image" >
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ex hic quis quasi repellat magni libero et doloribus! Minus, nulla enim sit consequatur nobis quod, temporibus, mollitia doloremque deserunt nesciunt sed.</p>
        </section>
        <br>
```

Contact

Hi, this is my portfolio

Here I deal with my story, achievements and some blog content

Email me

Subtitle 1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ex hic quis quasi repellat magni libero et doloribus! Minus, nulla enim sit consequatur nobis quod, temporibus, mollitia doloremque deserunt nesciunt sed.

Subtitle 2

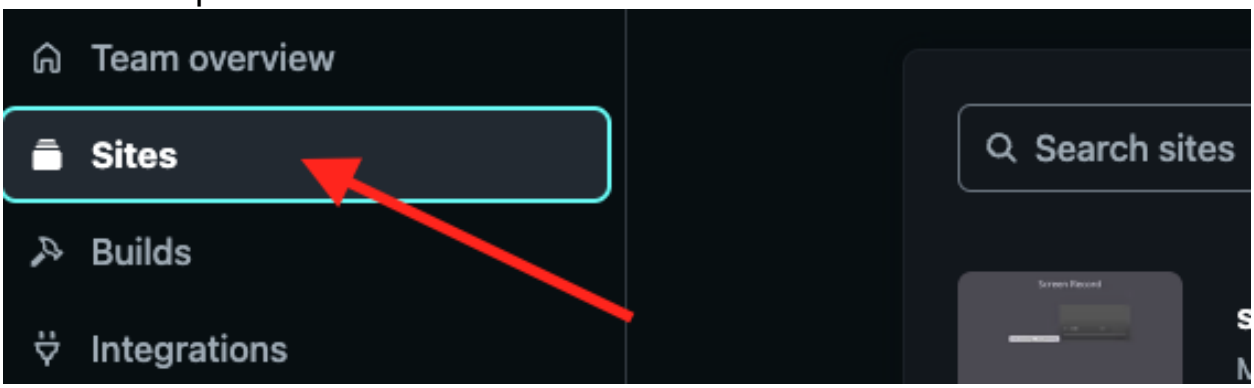
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vel repellendus ipsa dignissimos esse quis, eveniet, molestiae error necessitatibus ipsum, eius unde. Modi deserunt cum similique blanditiis, dolores suscipit non voluptas?

In result we have this simple webpage, which we will deploy now to the hosting platform and have a default free domain for live tracking of our website.

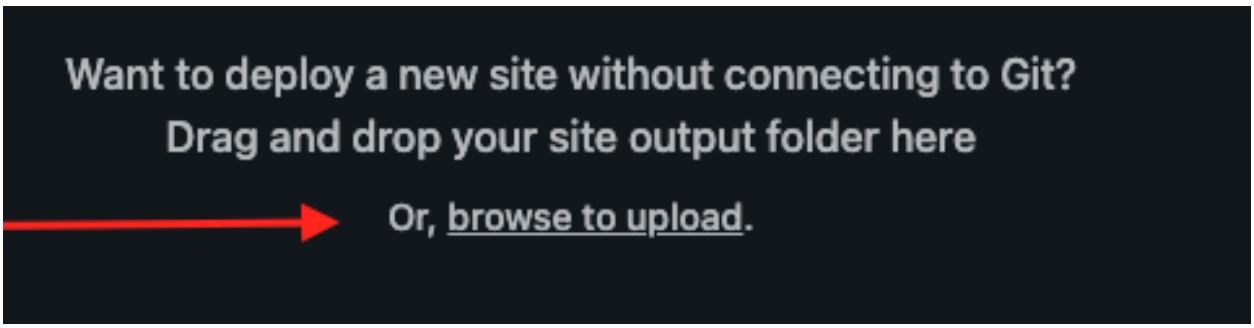
Hosting (server)

Few words about hosting (server). You can consider them as a parking place for your website code (files), which you connect with your domain and have the full website online.
Domain and hosting management is a different topic, which we will discuss here only from the point of the free hosting platform – Netlify.
Now, when you are ready to deploy (host) your files onto the hosting online, you need to do the following:

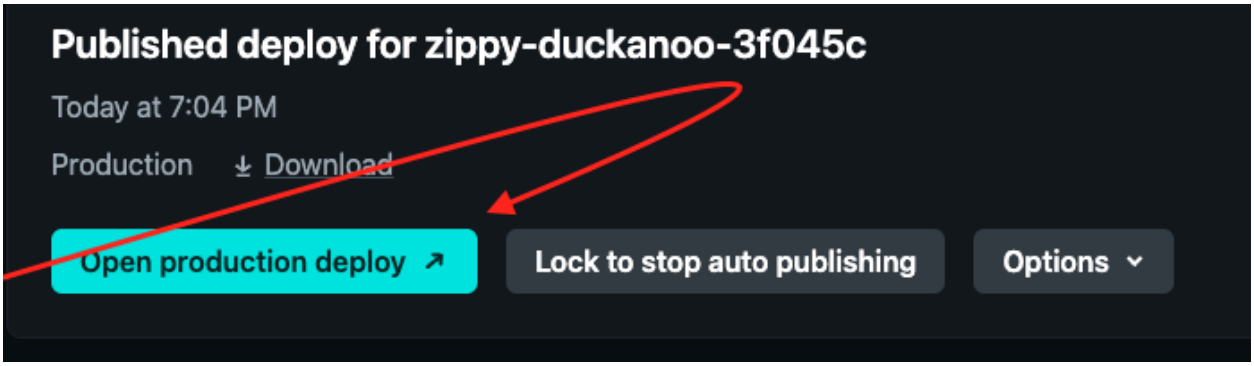
1. go to https://www.netlify.com/ website
2. sign up
3. press on Sites



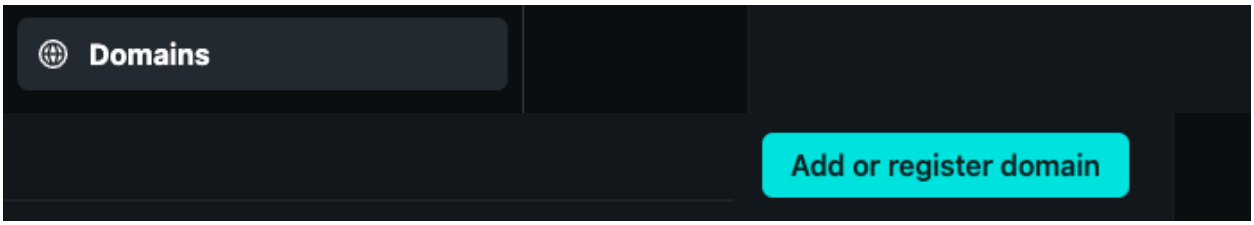
4. Then go to the bottom and press browse to upload link:



5. After, choose your folder, without zipping it (you will have there index.html file and images) and select it.
6. Click on Open production to deploy and you will se your website with a free default domain. In my case it's https://6514446ecab9c4031ea29cf6--zippy-duckanoo-3f045c.netlify.app/



That's all, congratulations. You made your own simple website with this e-book. If you want to change in future some data in your code, you can change it locally then deploy again, it will give you a new default domain. For having your custom domain you can buy a domain for example on https://www.namecheap.com/ and link to your files on Netlify in domains section.



Now, if you want to go deeper into the programming, you have to learn CSS (the styling) and the some programming language (for example JavaScript or PHP) for having a great functionality for your website.

The full code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Portfolio</title>
</head>
<body style="padding: 0px 15px; max-width: 1290px; margin: auto;">
    <nav style="display: flex; align-items: center; justify-content: space-
between;">
<img src="coding.png" alt="logo image">
<ul style="display: flex; justify-content: space-between;">
```

```
    <li style="list-style: none;"><a href="tel:+123456789" style="text-
decoration: none;">Contact</a></li>
</ul>
    </nav>
    <header>
<h1>Hi, this is my portfolio</h1>
<p>Here I deal with my story, achievements and some blog content</p>
<a href="mailto:youremail@gmail.com"><button>Email me</button></a>


    </header>
    <main style="min-height: 600px;">
        <br>
        <section>
<h2>Subtitle 1</h2>
<img src="https://images.unsplash.com/photo-1531297484001-
80022131f5a1?ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3
D%3D&auto=format&fit=crop&w=1120&q=80" width="250px" alt="laptop
image" >
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ex hic quis
quasi repellat magni libero et doloribus! Minus, nulla enim sit consequatur
nobis quod, temporibus, mollitia doloremque deserunt nesciunt sed.</p>
        </section>
        <br>
        <section>
            <h3>Subtitle 2</h3>
            <img src="https://images.unsplash.com/photo-1507238691740-
187a5b1d37b8?ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3
D%3D&auto=format&fit=crop&w=910&q=80" width="250px" alt="">
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vel
repellendus ipsa dignissimos esse quis, eveniet, molestiae error
necessitatibus ipsum, eius unde. Modi deserunt cum similique blanditiis,
dolores suscipit non voluptas?</p>

        </section>
    </main>
    <footer>
<p>Copyright 2023. All rights reserved.</p>
```

```
    </footer>
</body>
</html>
```

Happy End

I am happy to tell you, that by buying this e-book you will have a one-month support from me. So, if you are facing any issues, you can send me an email. I would be happy to help you.