

Subscribe to  Updates

code.makery (/)

Learning how to code.

Home (/)

Library (/library/)

Paths (/paths/)

Blog (/blog/)

About (/about/)

Edit on GitHub

https://github.com/css-tutorial/lib/code.makery.ch/edit/master/content/library/html-css/part1/_index.en.md

Part 1: Your First Website

Aug 10, 2018

Articles in this Series ▾

Downloads ▾

Languages ▾

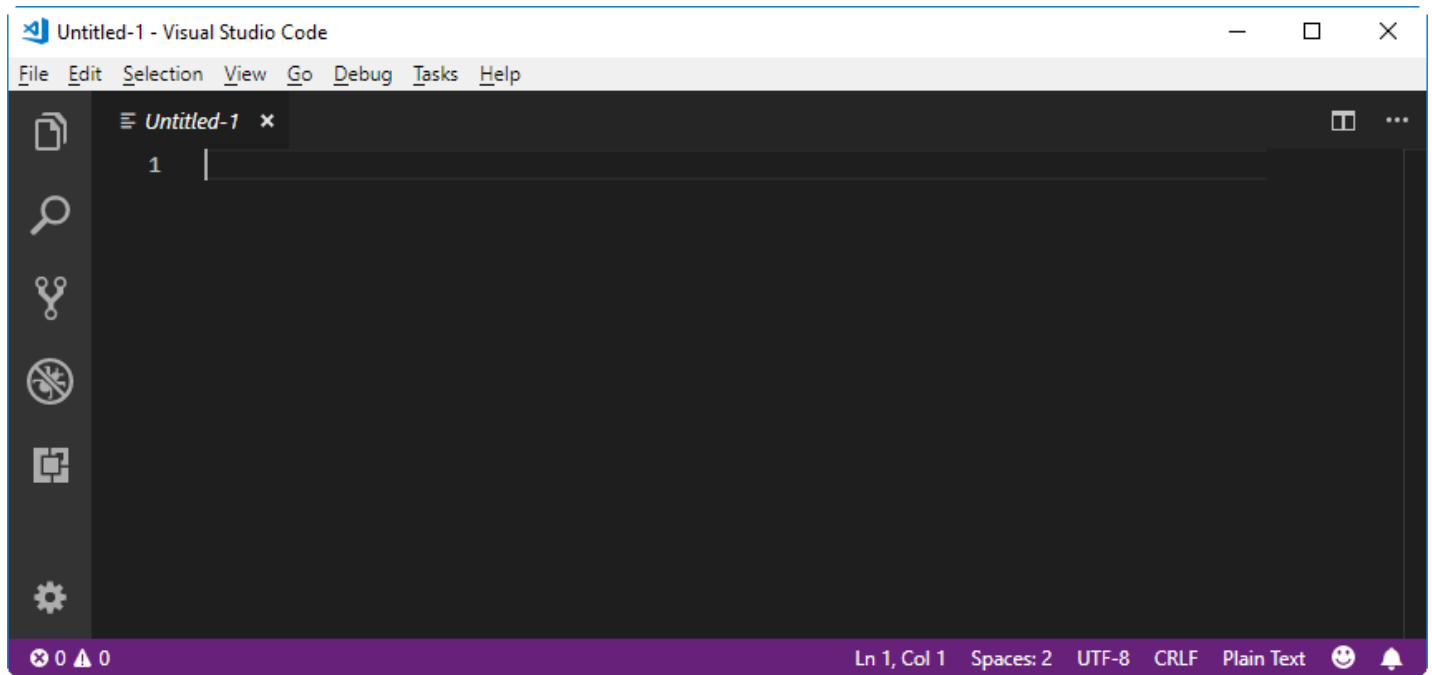
Let's dive right in and create our first web page.

Our Tools

For web development, you need two programs: an **editor** to create the files for the website and a **browser** (for example Edge, Firefox, Safari, or Chrome) to view and test your website.

Editor

For creating web pages, a normal text editor would be sufficient. But there are editors that greatly simplify your work as a programmer. Therefore, I recommend that you use a modern code editor.



My current favorite among the many editors is Visual Studio Code (<https://code.visualstudio.com/>) (free). You may use an editor of your choice but I will occasionally refer to some functions of Visual Studio Code (VS Code). Good alternatives are Atom (<https://atom.io>) or Brackets (<http://brackets.io>).



(<https://code.visualstudio.com/>)

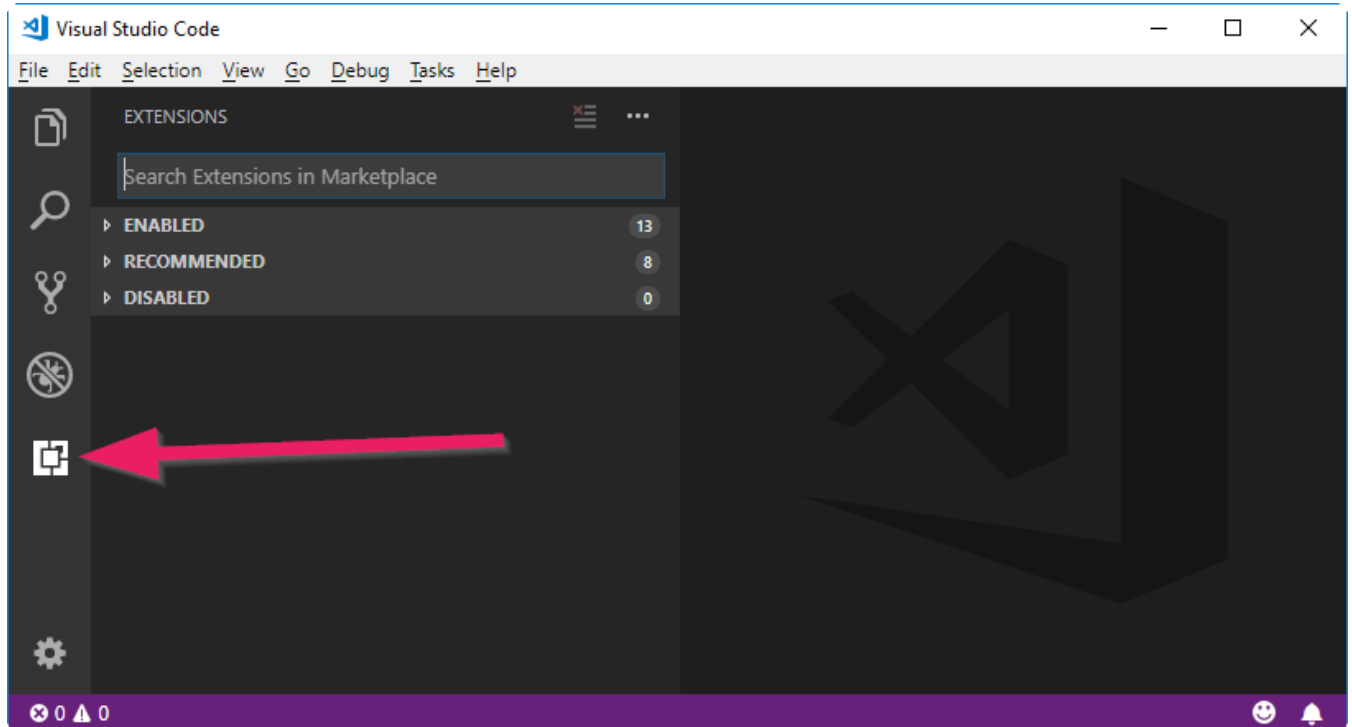
Go ahead and install an editor. If you do not know which one, use Visual Studio Code (<https://code.visualstudio.com/>) for now.

Editor Extensions

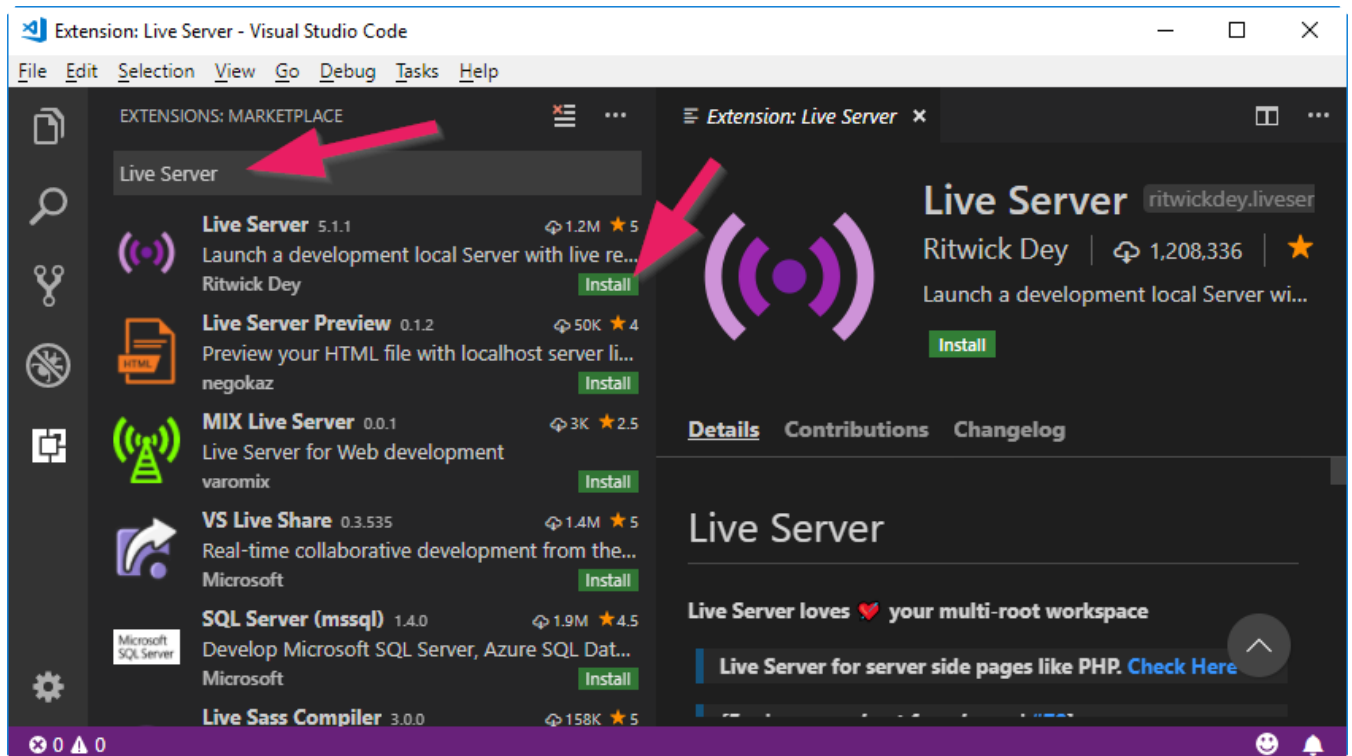
While we're at it we'll add an extension to VS Code that will give superpowers to our editor: Live Server (<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>).

1. Open Visual Studio Code.

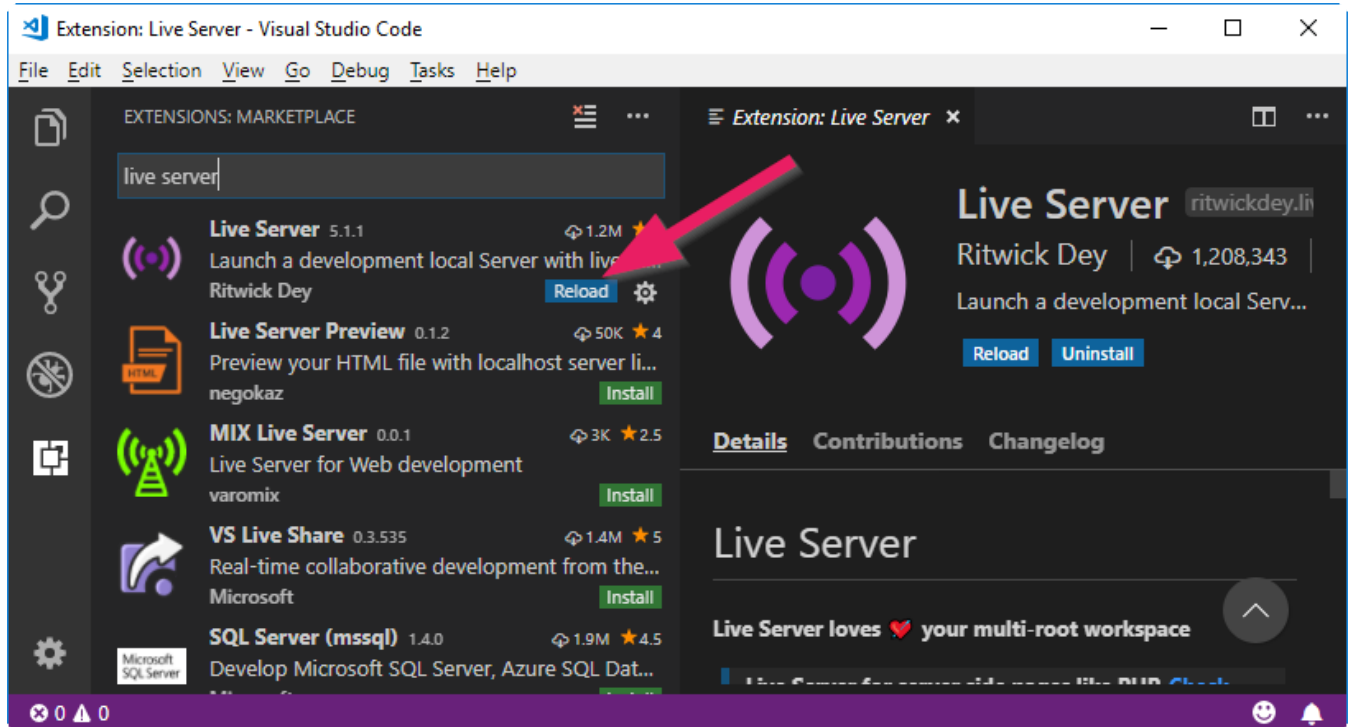
2. Open the menu "Extensions".



3. Type "Live Server" into the search field and install the extension.



4. Reload Visual Studio Code to activate the extension.



Browser

Our website should, of course, run in all major browsers (Edge, Firefox, Chrome and Safari). However, for web development I recommend you use Chrome (<https://www.google.com/chrome/>). Chrome includes very useful tools for developers which you will use often.



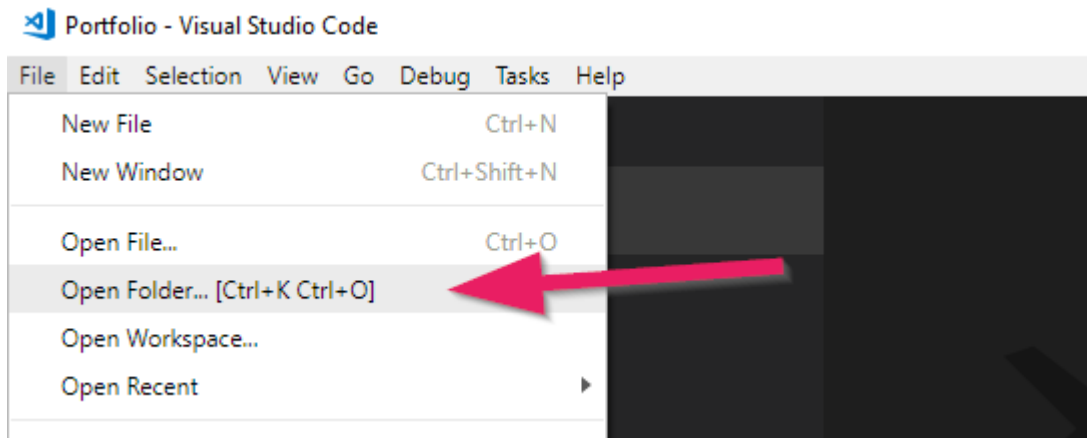
If you do not have Chrome on your computer, you can install it here (<https://www.google.com/chrome/>).

Creating an HTML Document

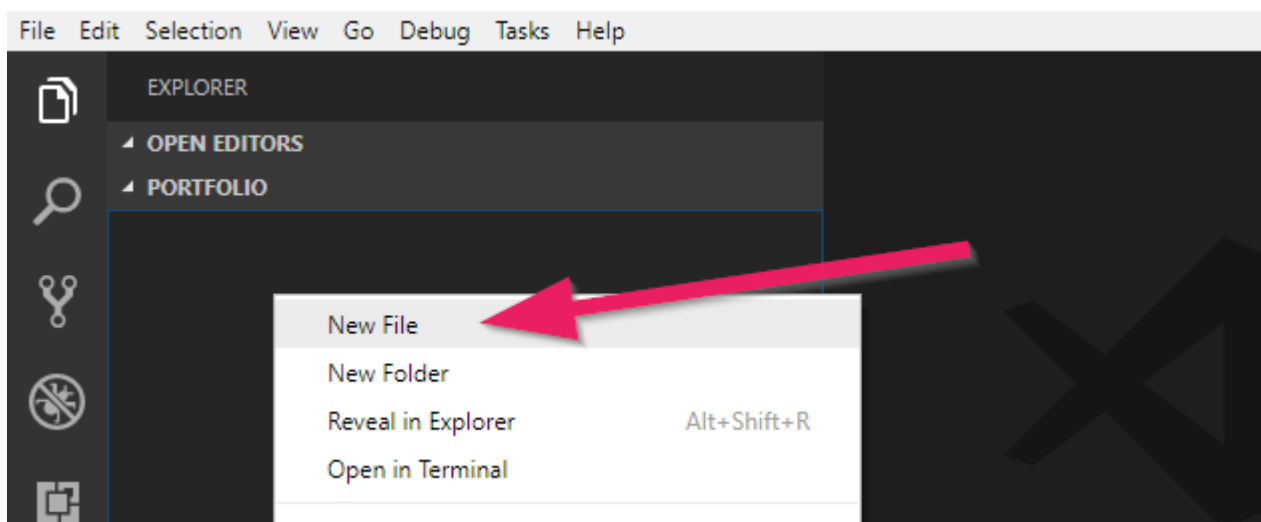
Equipped with editor and browser, let's create the first HTML document for our website.

1. Create a folder on your computer for your project. Name the folder *Portfolio* (or anything you want).
2. Open VS Code.

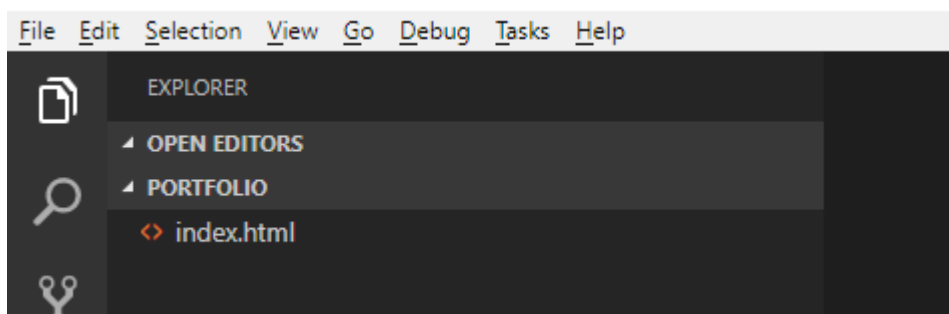
3. Open the **File** menu and select **Open Folder** Browse for the folder you created and open it.



4. Right-click below the folder and select **New File**. Name the file `index.html` .



5. Now you have a blank text file named `index.html` .

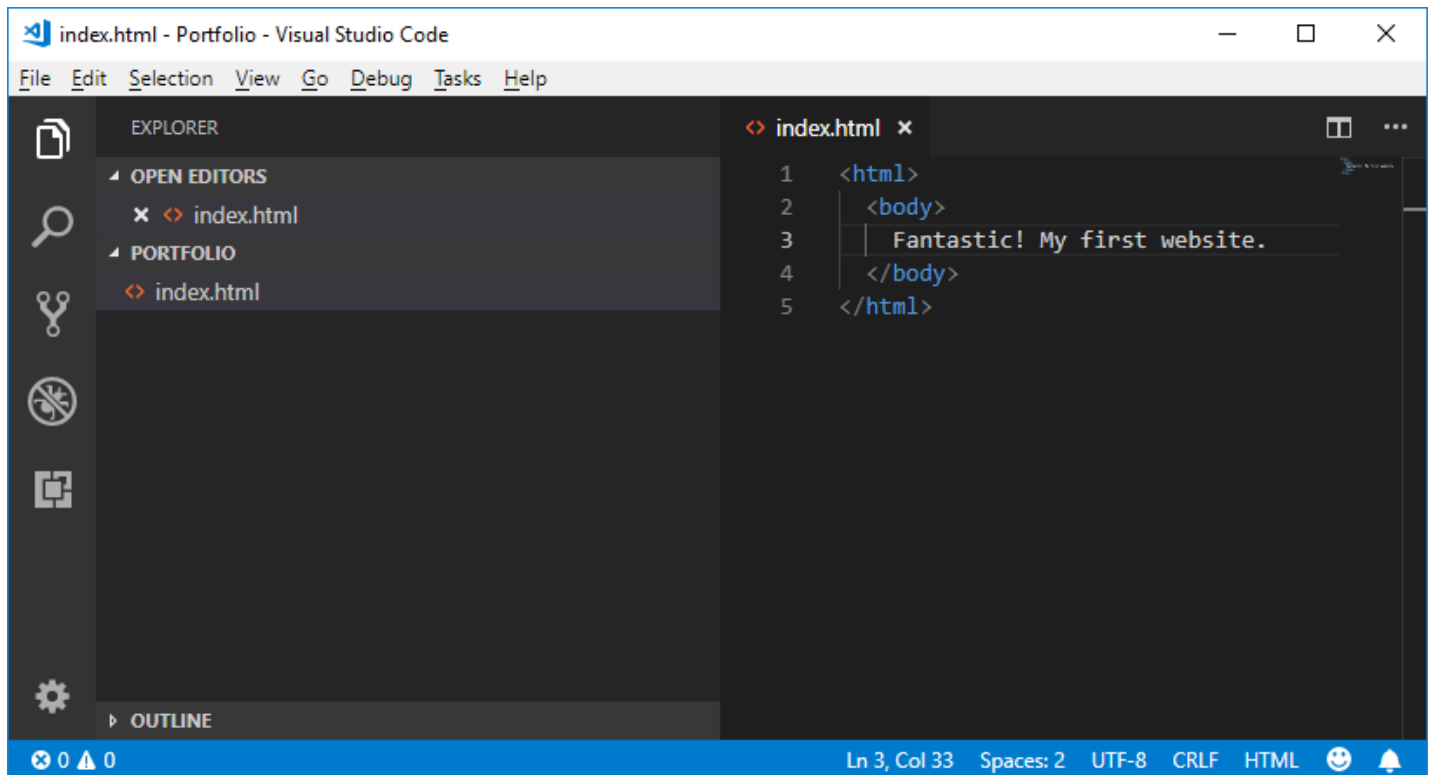


Why index.html?

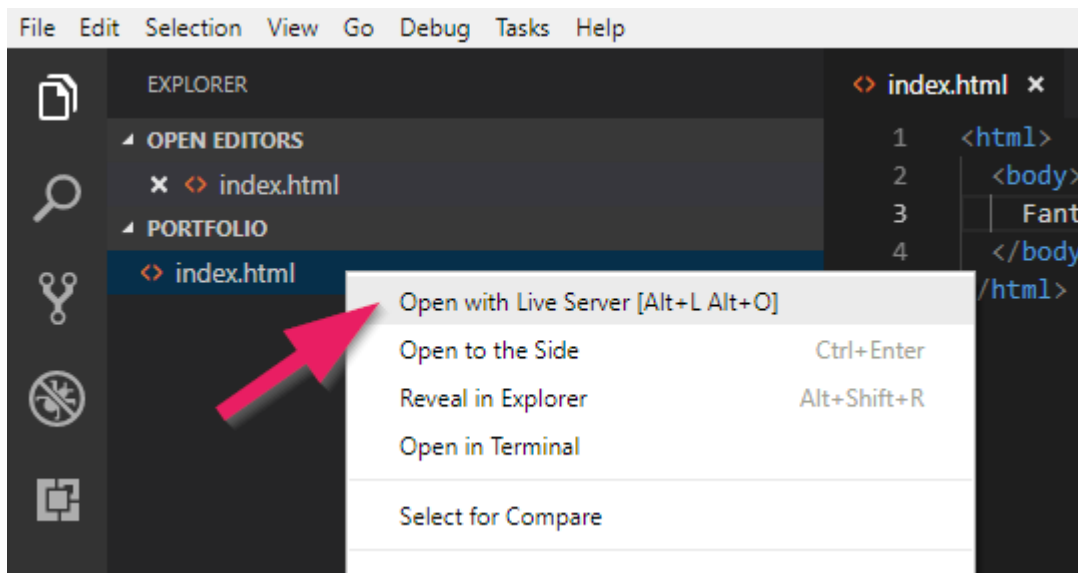
You could probably assume, the name `index.html` does have special meaning. If a website address is called, for example `https://code.makery.ch` , the `index.html` file is automatically displayed first, which in this case is `https://code.makery.ch/index.html` . For us, our first `index.html` will become our home page.

Viewing and Refreshing

Now you can fill the document with content. Type the following lines in your HTML.



To view the page, we'll use the **Live Server** extension that we installed above. Right-click on your `index.html` (the file below the Portfolio folder and not the one above under "Open Editors"). Then click **Open with Live Server**. A browser window should open up and show your first website. Now, whenever you save any changes they are automatically updated by the live server.



If the page does **not open in your preferred browser**: Change the settings on your computer to make another browser your standard browser. Alternatively you can copy the address (for example `http://127.0.0.1:5500/index.html`) from one browser and insert it into another.

If the the page is **not updated** after changes: Save all your files in the editor and then - inside the browser - refresh with `F5` or `cmd+r` .

Congratulations! You have just created your first website!

Indentation

In order to keep your code clean, it is important that you correctly indent the lines with the **tab key**. Pay close attention to the example code and indent accordingly. The browser does not care but it is important for us as programmers to be able to have a clear view.

Tip 1: Use `Shift+Tab` keys to move indentation to the left.

Tip 2: You can indent multiple lines at once if you select them and click `Tab` or `Shift+Tab` .

Tip 3 (automatically): Right-click somewhere in your program code and choose `Format Document` .

HTML Elements

Tags

In the above example you have already seen the typical HTML characters with angle brackets. These are called **tags**.

HTML elements consist usually (but not always) of two tags, one opening and one closing. In our example, `<html>` is an opening tag and `</html>` with the slash is a closing tag.

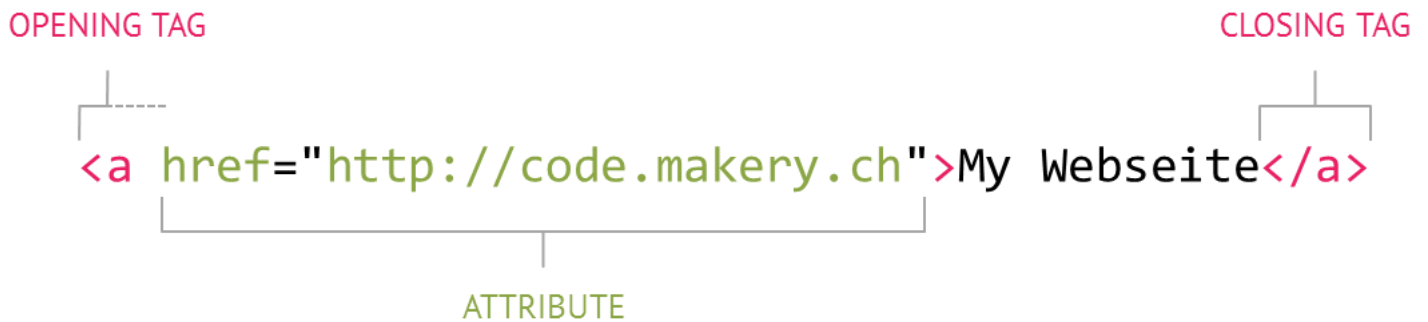
The text between the opening and the closing tag is the content of the HTML element. With `<html>` and `</html>` we tell the browser about the beginning and end of the HTML of our web page.

The second tag we have seen is the `<body>` tag. It states that all content between the opening `<body>` and the closing `</body>` tags are to be shown in the main area of the browser.

Attributes

Attributes declare additional information for an item. Attributes are part of the opening tag of an element and always have a **name** and a **value**.

As an example, let's look at the HTML element for a link. It is probably one of the most important elements — what would the internet be without links?



The `<a>` element above includes an `href` attribute (short for “hypertext reference”) with the value `https://code.makery.ch`. The browser knows that it must show the link as My Website (`https://code.makery.ch`).

Basic Structure of an HTML Page

We have already seen the two elements `<html>` and `<body>`. But the basic structure of an HTML page usually contains a few more. Adjust your website to the following code. Afterwards, we will discuss each element it.

HTML Structure


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Web Portfolio of Marco</title>
  </head>
  <body>
    <h1>Web Portfolio of Marco</h1>
    <p>Write anything you want to tell the world.</p>
  </body>
</html>
```

Explanations

- Always put `<!DOCTYPE html>` on the first line. It tells the browser about the type of the document.
- The `<html>` tag indicates the start and `</html>` the end of the document.
- The `<head>` element contains additional information about the page. In contrast to the `<body>` element, this information does not appear in the main area of the browser.
 - Within the `<head>`, there should be an indication about the character set: `<meta charset="utf-8">`. If you do not specify the character set some special characters may not get displayed correctly. You might have noticed that the `<meta>` element has no closing tag. There are a few elements without closing tags (`
`, ``, etc.), but they are the exception.
 - Next you'll see the `<title>` element. The title is displayed in the title bar at the top of your browser window.
- Everything inside the `<body>` element is displayed in the main area of the browser.
 - A `<h1>` defines the main heading. Sub headings can be created with `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`.
 - Text between `<p>` and `</p>` is a paragraph.
- After each opening tag, the next element should be indented (with a tab or two spaces) for better overview. Make sure you follow this habit.

Tip 1: Keep this basic HTML structure handy. You can use it for any new HTML page!

Tip 2: Use the keyboard shortcut `Ctrl+S` to save the current file.

Tip 3: Use the keyboard shortcut `Ctrl+Z` for undo.

With these basic HTML elements we are well prepared to bring our website to a new level. First, let's add an image so that the home page of our portfolio looks a bit more interesting.

Inserting an Image

To insert an image we use the `` element. The following example will insert my picture:

```

```

The `` element only has one opening but no closing tag. It contains a `src` and an `alt` attribute. The `src` attribute specifies the URL, that is the location and file name of the image. The `alt` attribute is an "alternative text", which describes the content of the image. This text is used by search engines and if for some reason the image can not be displayed, for example on a screen reader for the blind.

Relative and Absolute URLs

URLs are used for the `src` attribute of images and also for the `href` attribute of links. The URL specifies the "address" of a file (for example another web page or an image). Depending on the location of the file, either a *relative* or *absolute* URL must be used.

If a file is **part of the same web site**, then a **relative URL** can be used. As we have seen in the example above, this is only the name of the file.

A **relative URL** is either *relative* to the current HTML page or *relative* to the root of our website. To reference a file in the same folder, we can just use the filename like `marco.jpg`. If the target file is located in another folder, this must be taken into account. If the image from the example above were in a subfolder called `images`, the *relative URL* would be `images/marco.jpg`. If the file is in a parent folder you can reach it with `../`. The URL for the image in this case would be `../marco.jpg`.

Instead of navigating from the current file we can also start from the root of our website by adding a leading slash `/marco.jpg`. We can use this from any file in your website and in any subfolder and it will always start at the root. This is helpful for navigation links as we will see in a following part of the tutorial.

If the file is located **on another website**, an **absolute URL** must be used. Absolute URLs contain the entire domain name and path. An example would be `https://code.makery.ch/images/marco.jpg`.

Remember the following about URLs:

- Inside the same folder we just use the filename, for example `marco.jpg`
- Two dots (`..`) refer to the parent directory.
- If we want to start in the root directory we add an `/` before the path of the file, for example `/marco.jpg`.

Examples of Relative and Absolute URLs

```
<!-- Relative URLs -->
<a href="image-gallery.html">Image Gallery</a>
<a href="blog/first-blog-entry.html">My First Blog Entry</a>
<a href="../image-gallery.html">Back to Image Gallery</a>

<!-- Absolute URLs -->
<a href="http://www.my-colleague.com/blog.html">Blog of a Colleague</a>
```

Portfolio with an Image

If you have not already done so, try and insert an image in your portfolio. You must copy an image to your *Portfolio* folder on your computer. Make sure that you specify the exact file name, including the file extension.

All of your code could now look something like this (I've added a subheading and some more text):

index.html with Finished Portfolio Code

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Web Portfolio of Marco</title>
  </head>
  <body>
    <h1>Web Portfolio of Marco</h1>

    <h2>Welcome!</h2>

    <p>Thanks for stopping by.</p>

    <p>Please have a look around. In the blog section I document my experiences in programmi

    <p>Marco :-)</p>
  </body>
</html>
```

That's how the portfolio currently looks like in the browser:



What's next?

→ In the next section you will learn how to publish your website on the internet. Continue with Part 2: Publishing Your Website (</library/html-css/part2/>)

Intro (/library/html-css/)	1 (/library/html-css/part1/)	2 (/library/html-css/part2/)
3 (/library/html-css/part3/)	4 (/library/html-css/part4/)	
5 (/library/html-css/part5/)	6 (/library/html-css/part6/)	7 (/library/html-css/part7/)
Next (/library/html-css/next/)		

Comments

Palobrea • 5 years ago



This was very helpful for an absolute newbie like me! looking forward to the others. thank you

^ | ▾ • Reply • Share ›



Aquagreenjay ➔ Palobrea • a year ago

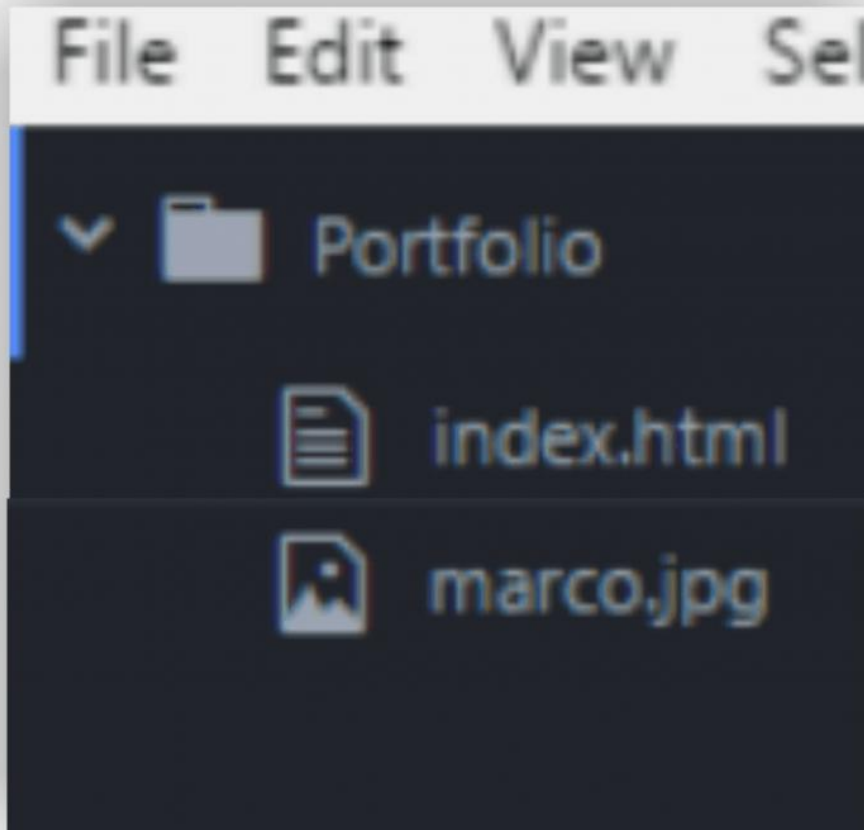
Agreed

^ | ▾ • Reply • Share ›



Δυόσμος • 5 years ago

I couldn't put my foto in the document. Until I noticed that I should drag the foto, from my desktop to the Portfolio file!



1 ^ | ▾ • Reply • Share ›



Barry L. Nelson • 4 years ago

This is the first explanation of how to create an html webpage in great detail that I have found so far and it is fully functional. So far I have completed instruction from page #1 and it works.

1 ^ | ▾ • Reply • Share ›



Marco Jakob Mod ➔ Barry L. Nelson • 4 years ago

Hi Barry

Thanks for the feedback. I'm glad you like it!

Marco

4 ^ | ▾ • Reply • Share ›



Reni Yeno • 3 years ago

Is it possible to modify the address or host of the website URL <http://127.0.0.1:5500/index.html> to something like <https://HelloWorld.html>?

1 ^ | v • Reply • Share ›



Xephonine → Reni Yeno • a year ago

i also wanna know that

1 ^ | v • Reply • Share ›



Qw • 3 years ago

Can i create a login/signin box?

1 ^ | v • Reply • Share ›



Qw • 3 years ago

Can i create login/signin and a reserved settings to admin

2 ^ | v • Reply • Share ›



Marco Jakob Mod → Qw • 3 years ago

You could use Netlify's (paid) visitor access control: <https://www.netlify.com/doc...>

^ | v • Reply • Share ›



Racheal • 3 years ago

Exactly what I was looking for, thank you!

^ | v • Reply • Share ›



JavaSolution • 2 years ago

Such wonderful post

Thanks for your helpful post

[HTML USE](#)

^ | v • Reply • Share ›



HEHEHE • a year ago

rhnx

^ | v • Reply • Share ›



HEHEHE • a year ago

Sponsored

Ultra-Wealthy Farmer Says “Wake Up America”

Investing Outlook

Homeowners Are Trading in Their Doorbell Cams for This.

PureCare

If You Have Gum Disease Or Tooth Decay, Do This Immediately (It's Genius)

Healthbay

Gronk's Favorite "Dressy" Shoes Feel Like Walking On Clouds

Wolf & Shepherd

These Are The 22 Coolest Gifts For 2022

Top Trending

Shop Now

We are selling off our remaining magic metal windmills. Great garden decor.

ballshot



Subscribe to receive updates by Email (<https://tinyletter.com/codemakery>).

Enter email

Get updates

