Creating a front-end webpage

Suppose you wanted to have a corner of the internet dedicated to your creativity. Perhaps you have an interest in pets, and you wanted to write a web page about that. Or, you wanted to showcase your skills and make a personal webpage.

In this user instruction manual, you will be able to do just that. You will learn how to make a simple webpage using HyperText Markup Language, or HTML. You don't have to be proficient at computers; this manual is designed for anyone to follow!

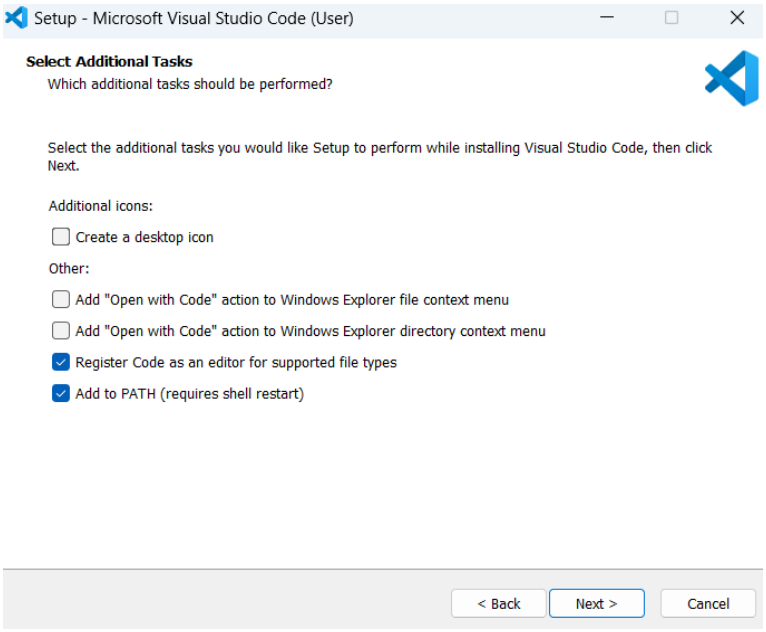
Set up

Code Editor

Before we begin, we will need an environment to edit the code. These code editing environments are known as **code editors**. A popular favorite for web development is Visual Studio Code (VSCode). I would highly advise this editor because of the additional add-ons that you could install tol make the coding process more efficient and simple.

Go ahead and give it a Download!

The software installation process is pretty straightforward:



After you click on the download button, run through the setup.

When you arrive at this screen, make sure to leave the bottom two checkmarks checked.

The other check marks are less essential.

After that, proceed with the install button.

Figure 1: VSCode Installer default setup options

The Project Folder

Once the editor has been downloaded, you will need a folder to save all your project work. To do this, make a folder anywhere you would like to have the project saved in. Give it a name, for example, “Website Tutorial.”

1. Give VSCode a run. The landing page should look like the following:

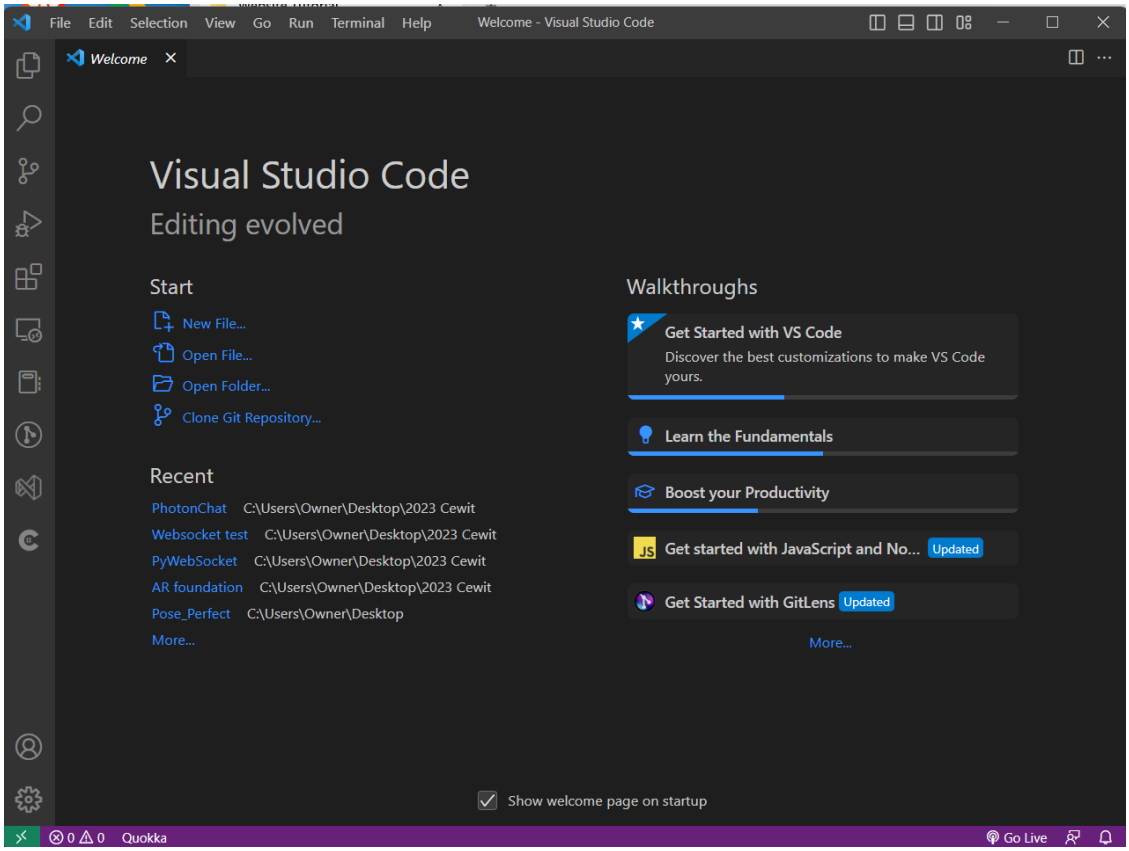


Figure 2: The home screen VSCode. Notice the options under the “Start” header.

   a. My window may look slightly different from that of a freshly installed VSCode. Don’t worry, since that won’t interfere much with the project setup.

2. Go ahead and click the open folder button. A file explorer window should open up.
3. Select the folder you just created. This should then be opened into VSCode, and now you can begin creating files!
4. We will create two files in the folder: Click the very first icon, which looks like two pieces of paper on top of each other.

5. This button is called the **Explorer**. It lists all the files in your project. Currently, it should be empty because we have none! Let's make our first file:
    - Right click the empty space, and click New File
    - Name it: index.html
    - Create another file, and name it: styles.css

Note: This tutorial will not go over how to use css. You may look into it if you would like to.

6. The explorer should look like this:

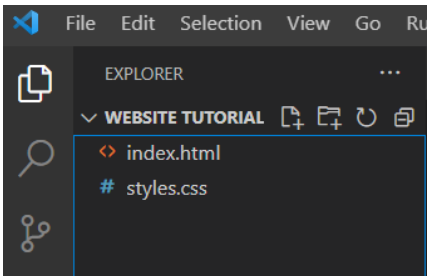


Figure 3: The current explorer list

We are almost ready to code. Before we do that, we will need to download extensions.

Additional (Essential) Extensions for VSCode

Extensions, or add-ons, are great for customizing the way you code. They mostly aid in making repetitive tasks, such as typing common phrases, automatic, thus saving precious time for development!

To download extensions, click on the icon on the left hand bar of the screen that resembles four cubes, with one detached cube on the upper right. You will then see a search bar. Install these following extensions:

- **Auto Rename Tag**, by Jun Han
- **HTML Boilerplate**, by sidthesloth
- **Live Server**, by Ritwick Dey

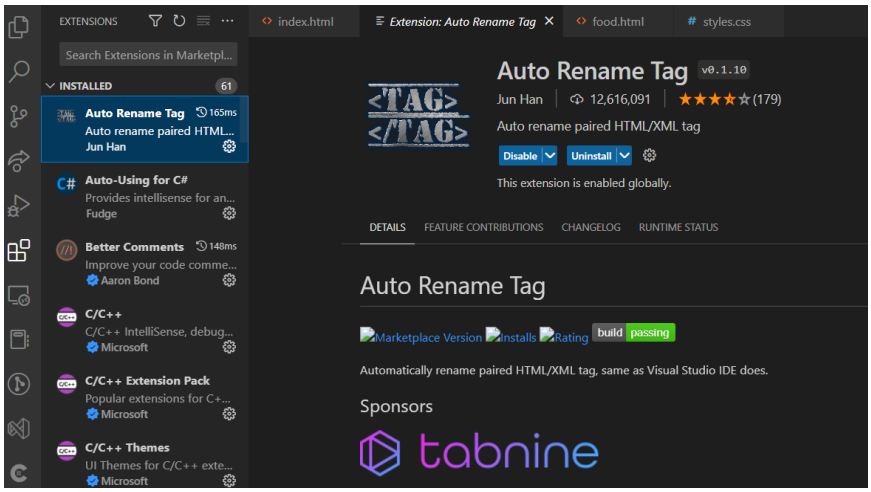


Figure 4: Note the search bar in the upper left corner, which says "Search Extensions in Marketplace"

Essential Terminology:

HTML - Some call it a programming language, but really, it's like a grammar that lets the browser know what to draw.

CSS - Whereas HTML tells the browser what to draw, this tells the browser how to draw it.

Elements - These are the bread and butter of HTML grammar and make up the body of the web page. An HTML element **must** have an open and closing tags, as well as the type of the element. Here's an example:

- Hello World!
- <> is an **open tag symbol**
- </> is a **close tag symbol**
- The "p" is the element type. Here, "p" stands for paragraph, making this element a paragraph element!

Figure 5: a visual showing open and close tags | source

Tags - There are a whole bunch of different tags that create different looking text. Some don't even create text–they create input fields and checkboxes! We won't be covering those however.

The Home Page

1. We'll begin with the home page. This is the page that will connect us to all other pages in the website, and is the most important one. Having a meaningful home page will allow for a great first impression.

Note: All text that is blue should be copied into the code editor.

2. Open up the index.html file, and type: doc. Then, click enter. The extension you previously downloaded should create a framework of code. Here is what you should see:

Figure 6: A lot of the text here is a bunch of code that doesn't concern us! Just make sure it's there (or else your website will break!)

Inside the open and close tags of the body is where our website code will be.

3. In line 6 of the code, you can see the title of our web page. I'll be changing it to: Website Tutorial. You can name it whatever you like.

4. Let's now edit the body with tags (refer back to terminology for a reminder). Here will be the first two tags that we will learn about:
    a. This is the paragraph tag.
    b. <h1></h1> This is the header tag. It's like the tag, but makes bigger text.
    - The 1 can be replaced with any number up to 5, with the heading growing smaller the larger the number is. This is useful for subheadings and sub-subheadings etc.

5. In the body, type the following:

```
<h1>Website Tutorial</h1>
<h2>Home page</h2>
<p>This is a paragraph element! <b>This text is bolded!</b></p>
<p>Welcome to the website!</p>
```

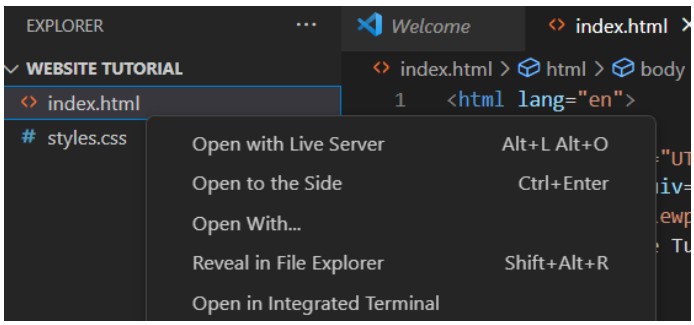
You will notice that as you're typing, some text is being automatically suggested and/or filled in. Those are the extensions hard at work letting you type less by filling in text for you!

Don't forget to continuously save your work by pressing ctrl+s on your keyboard! If you don't save your work, your code won't be updated when you run it!

Running the website

1. To get the website running, we will be using another extension yet again: Live Server. In the project explorer, right click the index.html file. You should see an option that says "open with live server".

2. In the photo to the right (Figure 7), it is the first option. Go ahead and click it. After waiting for a few seconds, you will now see a website open!



    a. Alternatively, you can open up your folder in its location saved on your computer, then click the index. html file.

b. This, however, will require you to refresh the page to see any changes–the Live Server extension refreshes for you automatically when you save your work.

3. The web page should now look something like this:

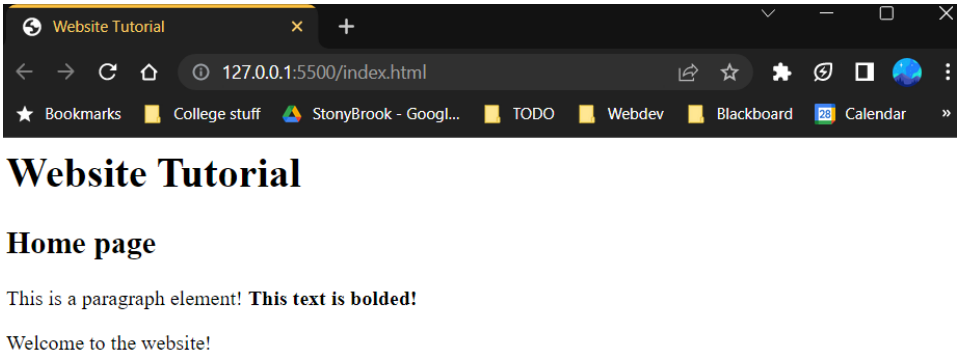


Figure 8: The result of our newly created home page!

4. Here's what you can notice that follows correctly from our code:
   - The tab is called "Website Tutorial". Conveniently, that's what we typed in the Title tags!
   - The text with the h2 tags is smaller than the text with the h1 tags
   - The paragraph texts are separated by vertical spacing.
   - The text surrounded by the <b> tags are bolded.

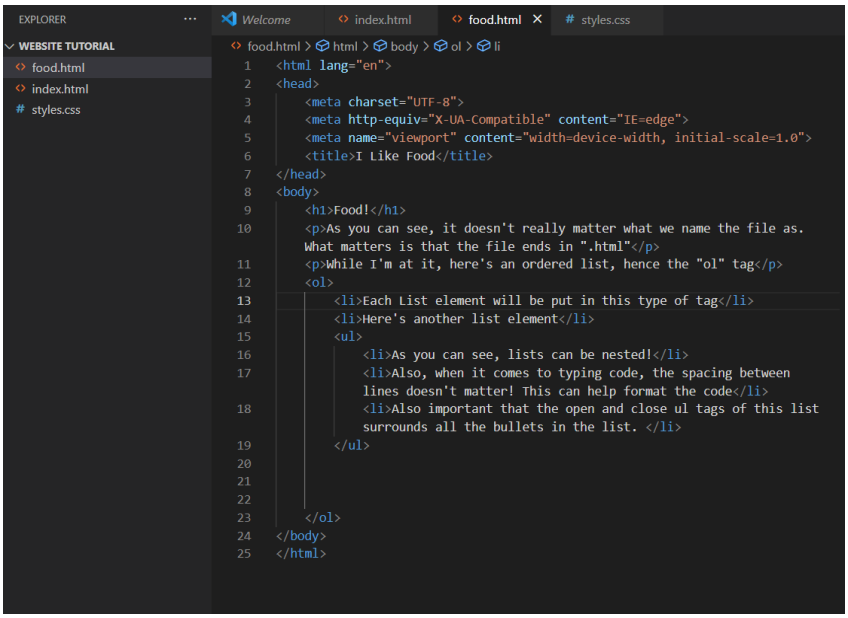
5. You can go ahead and add some more HTML elements of your own to populate the web page. Some tags you can try and research for yourself are:
   -
, <i>, , , <ol>, <ul>,

Linking to other pages

A website that is only made up of a home page is quite boring. So, let's make some more.

1. In the explorer, right click, and make some new HTML files. You can name them whatever you want, as long as you end the file name with: ".html"
2. Each of these HTML files will be a new page. In each of these files, go through the first few steps as with what we did for the index.html file.

Figure 9: Code example for the new webpage



3. You can try reading through the content of this example as I try to explain what each line does. To encourage some creativity, I didn't include this as text to copy.

Now, the question arises: how do I navigate to this page? You can try opening another live server tab for it. But that doesn't make sense if you just want to switch between pages in a single tab. To do this, we will need a new tag.

4. Head over back to your index.html file and add this line of code right before the end of the body:

```
<a href="food.html">Link to the other tab!</a>
```

Note: Instead of food.html, you would have the name of your new file.

5. This is known as the "anchor tag." It links your files together.
   - href = "" is like a question where the tag is asking you to fill in the quotation with the name of the destination file
   - The text in between the tags is the text that will be displayed. This can be different than the destination text

6. On your home page, you will see a link that you can click. When you click it, you will be brought to the page called, in my case, "I Like Food"!

Uh oh. We now seem stuck on this new page. How do I navigate back to the home page? You actually already know the answer: another anchor tag, with "index.html" in the quotation marks:

```
<a href="index.html">Back Home</a>
```

The new web page will look like this:

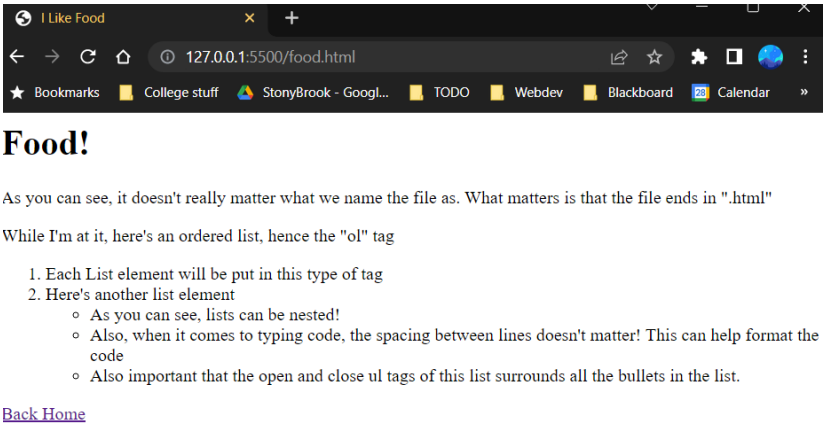
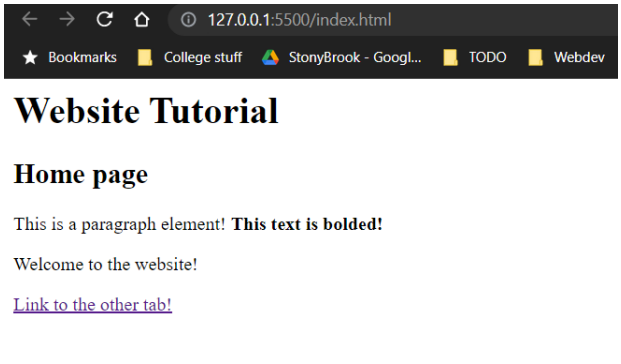
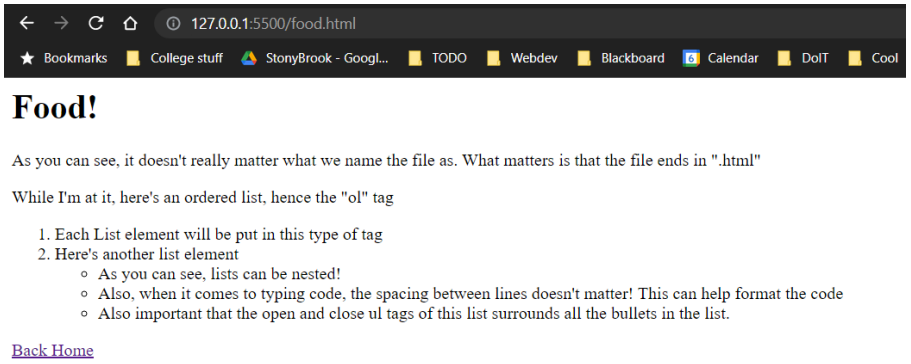


Figure 10: The new completed second webpage, with a link back to the home page.

Conclusion

Congratulations! You have now learned to create web pages and link them together. This has only been a brief introduction on how to make a website. There's so much more to learn, such as the other tags mentioned before. You can also add images, with the "img" tag, to spruce up your website.





Figures 11 & 12: The two completed web pages side to side

For now though, this little website will create your own corner of the internet!