

Deploy your projects on Github using Visual Studio Code and Git



Power Up Web Dev

[Follow](#)

Apr 27, 2018 · 10 min read

The screenshot shows the Visual Studio Code interface with three code editors open:

- tags.html.js**:
A module export function that generates HTML for a list of tags. It uses map to iterate over tags, and then joins them into a single string with class names.
- article.html.js**:
A module export function that generates HTML for an article. It includes a header with a title link, a require statement for tags.html.js, and a body section.
- video.html.js**:
A module export function that generates HTML for a video, starting with a header.

```
3 <h1><a href="#">article>
4 </header>
5 ${requirement}>
```

Photo by Pankaj Patel via Unsplash

Visual Studio Code is the new black. Everywhere I go, whether it's youtube or one of the popular social media sites, Visual Studio is being held in high esteem. I wondered why? So I checked out what it had to offer and what converted me to become a regular user is the way VSC approached version control.

What's not to love? It has integrated terminal. It presents version control in a visually appealing manner, making it easy to track file changes, and not to mention the plethora of commands organized in a drop down menu.

In this article, I will hold your hand from beginning to end and show you how you can start using Visual Studio Code and Git to deploy your projects on Github.

(If you haven't done so yet, check out my previous article, "*How to understand git and github for beginners*" in which I light-heartedly explain the *what* and the *why* to version control and its fundamental importance for anyone aspiring to be a web developer.)

What you need:

1. Download Git
2. Download Visual Studio Code
3. Create a Github Account

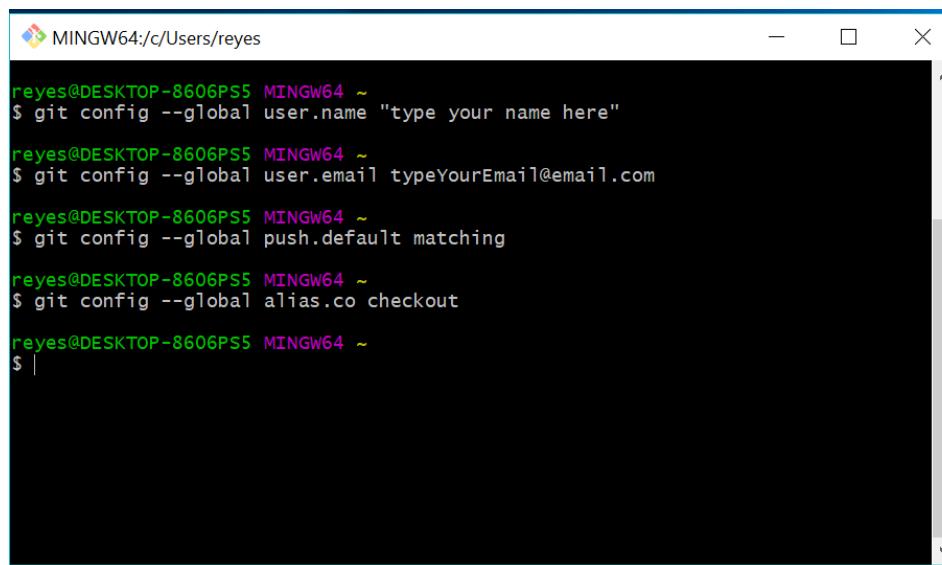
Installing them is pretty straight forward, but if you like reading manuals then docs are provided within the links I provided.

Once everything is installed, we begin with the first step.

1. Open the Gitbash terminal and configure git settings

Run these four commands:

```
git config --global user.name "type your name here"  
git config --global user.email typeyour@email.com  
git config --global push.default matching  
git config --global alias.co checkout
```



The screenshot shows a terminal window titled 'MINGW64:/c/Users/reyes'. It contains the following command history:

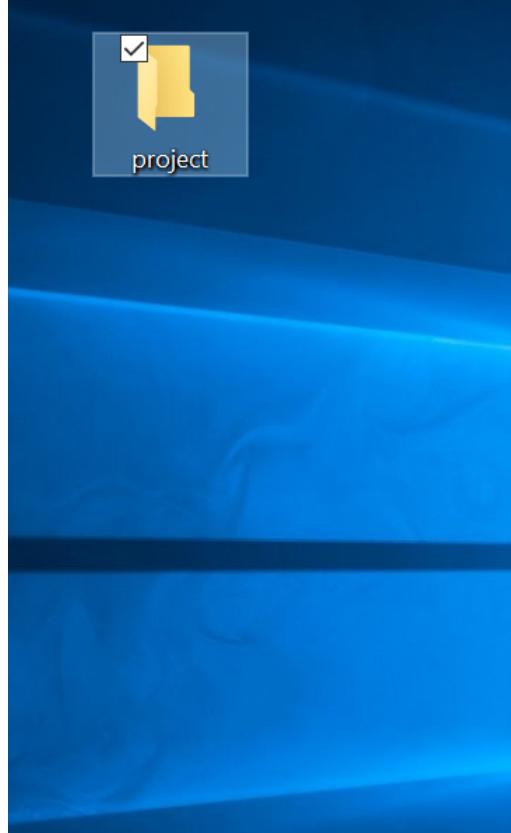
```
reyes@DESKTOP-8606PS5 MINGW64 ~  
$ git config --global user.name "type your name here"  
reyes@DESKTOP-8606PS5 MINGW64 ~  
$ git config --global user.email typeYourEmail@email.com  
reyes@DESKTOP-8606PS5 MINGW64 ~  
$ git config --global push.default matching  
reyes@DESKTOP-8606PS5 MINGW64 ~  
$ git config --global alias.co checkout  
reyes@DESKTOP-8606PS5 MINGW64 ~  
$ |
```

To see if you did this right, you can type:

```
git config --global user.name and git config --global user.email
```

Because there's a lot to cover, you can read the docs here, if you want to find out what they do.

2. Create a folder for your project



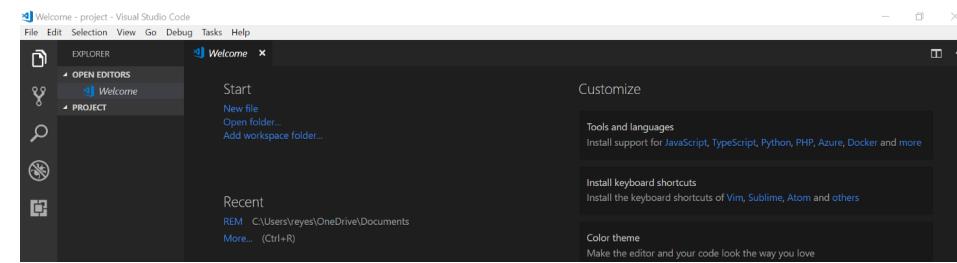
3. Open your folder project in Visual Studio Code

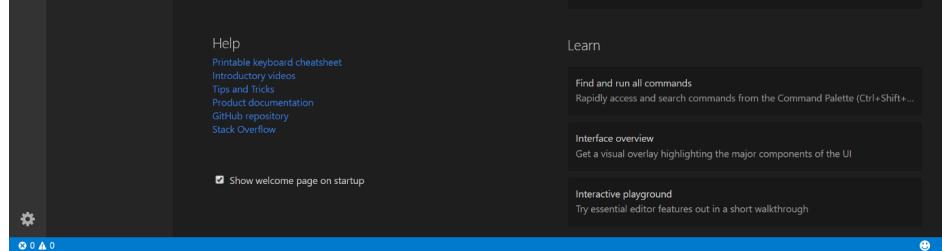
Click file → Click open folder → Highlight your folder → Click Select Folder

or

You can just drag your folder and drop it inside VSC

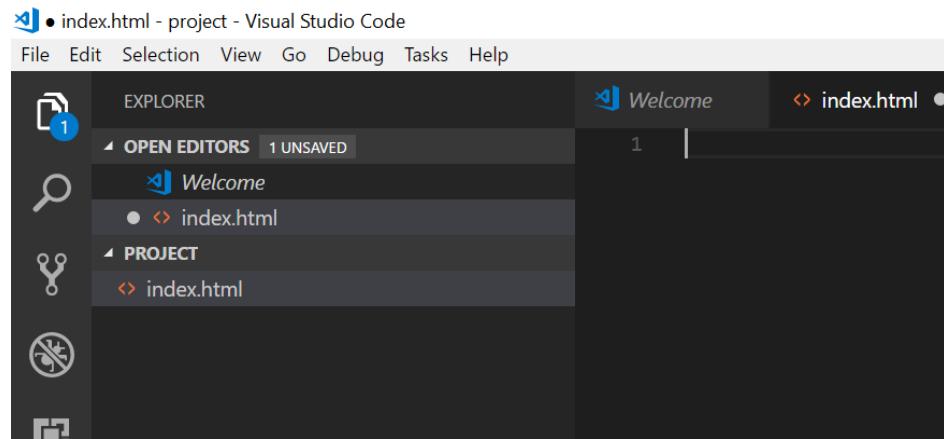
At this point, it should look like this:





4. Create a new file

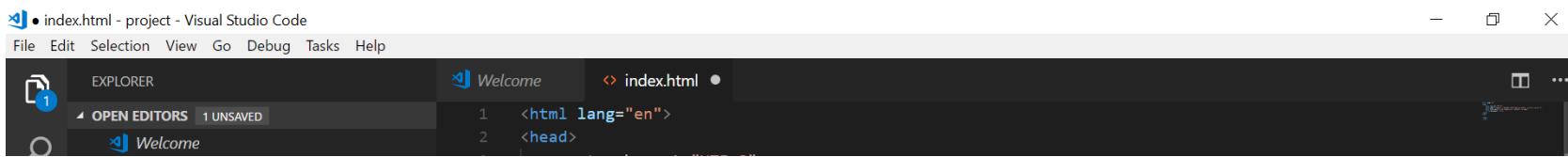
Click file → Click new file → Click file again → Click save → Name your file
index.html



5. Set up your index.html

Make sure you're on the index.html — Click on the blank canvas → Type
doc then press Tab immediately after

This will automatically create a simple HTML template for your index file,
one of the many built-in functions that make VSC awesome.



The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the Explorer sidebar, which lists a project folder containing an index.html file. The main editor area displays the following HTML code:

```
3 <meta charset="UTF-8">
4 <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 <meta http-equiv="X-UA-Compatible" content="ie=edge">
6 <title>Document</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

The status bar at the bottom indicates "Ln 4, Col 54 (12 selected) Spaces: 4 UTF-8 CRLF HTML".

6. Type a simple text

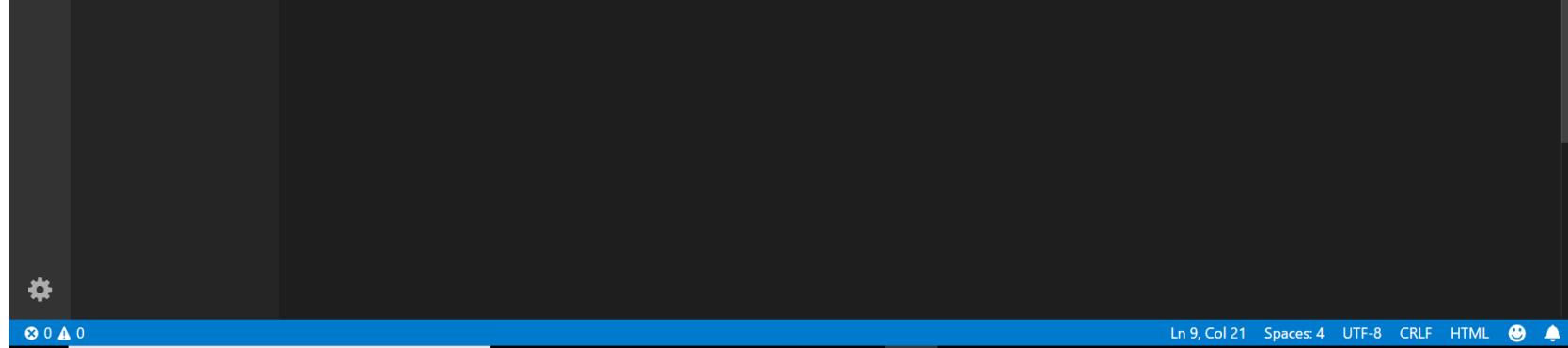
Click in between the <body></body> tags →

Type <h1>'Hello World'</h1> →

Click file → Click save

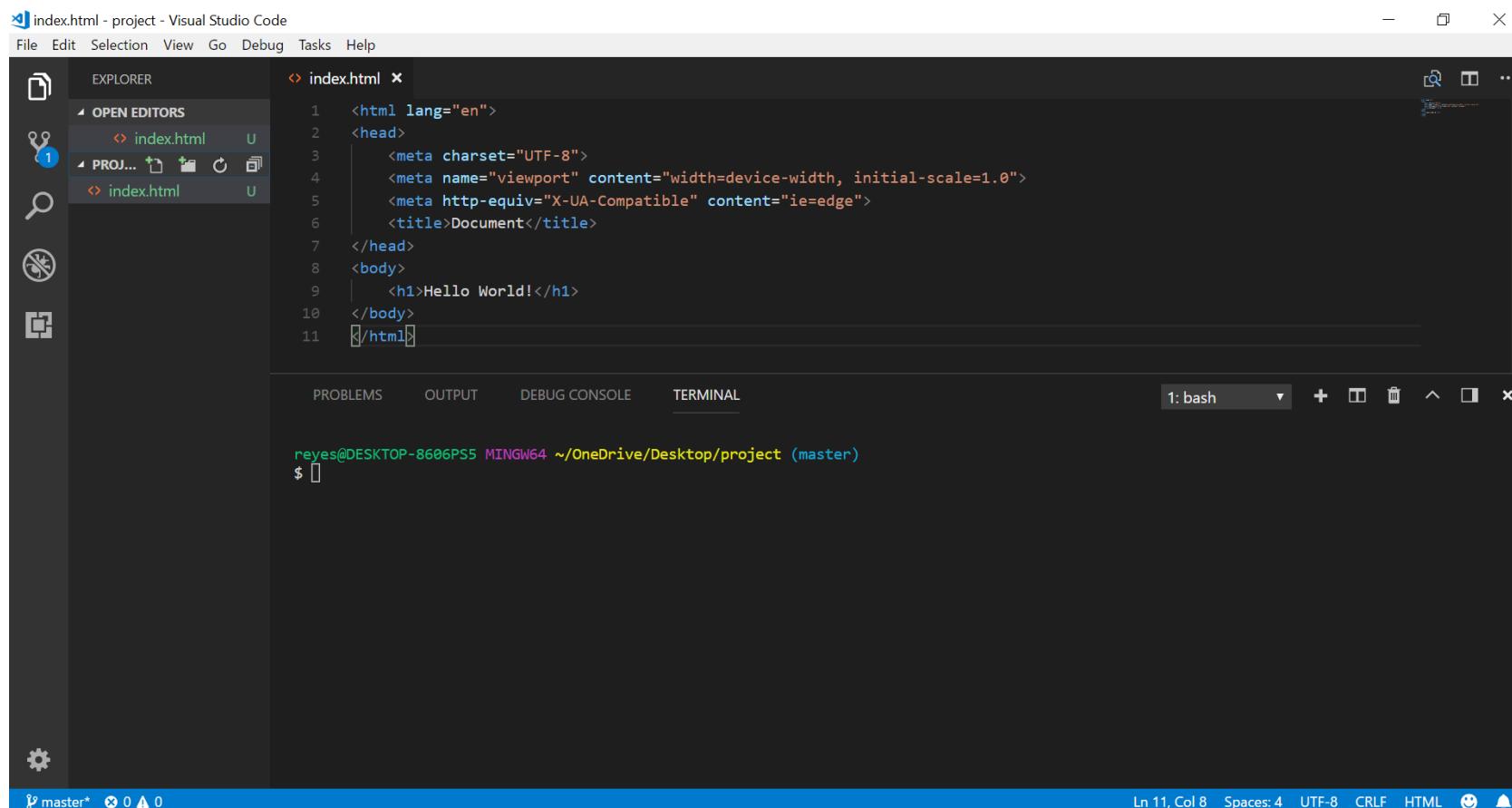
The screenshot shows the Visual Studio Code interface with a light theme. The Explorer sidebar shows the project structure with files named "Welcome" and "index.html". The main editor area shows the same HTML code as before, but with an additional line added between the body tags:

```
3 <meta charset="UTF-8">
4 <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 <meta http-equiv="X-UA-Compatible" content="ie=edge">
6 <title>Document</title>
7 </head>
8 <body>
9 <h1>Hello World!</h1>
10 </body>
11 </html>
```



7. Access the integrated terminal (git)

Click view → Integrated terminal



8. Change terminal default to git

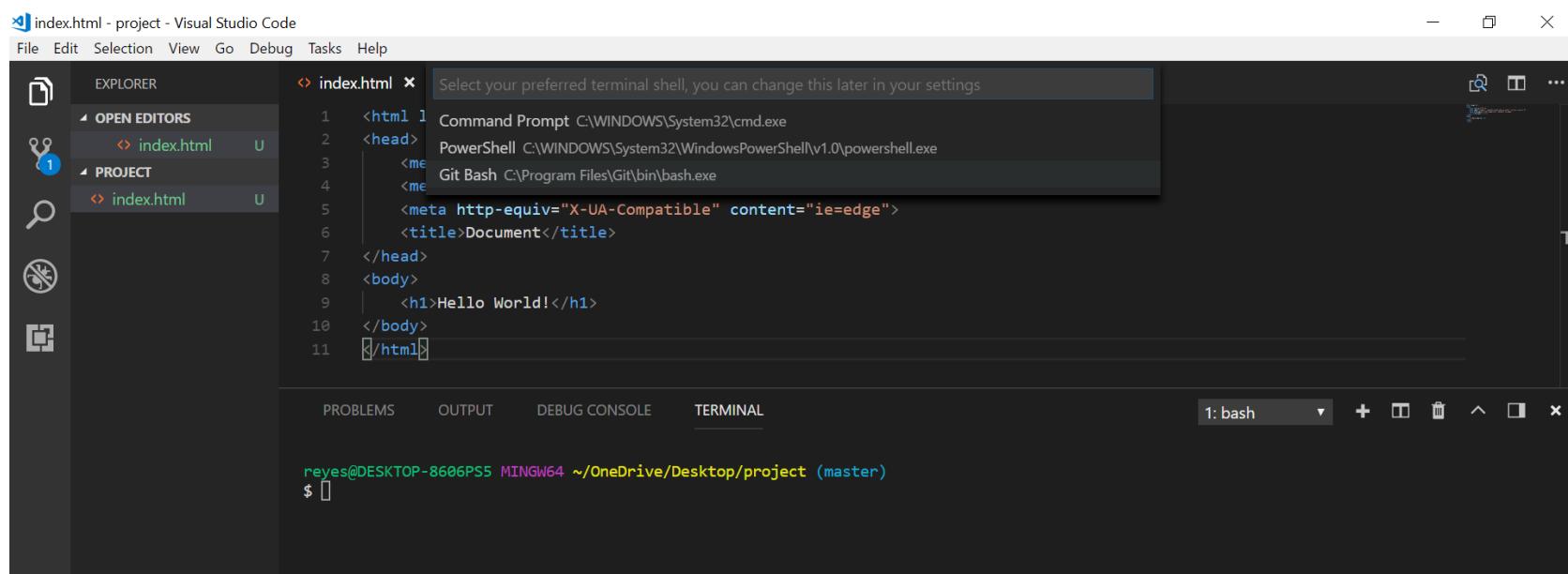
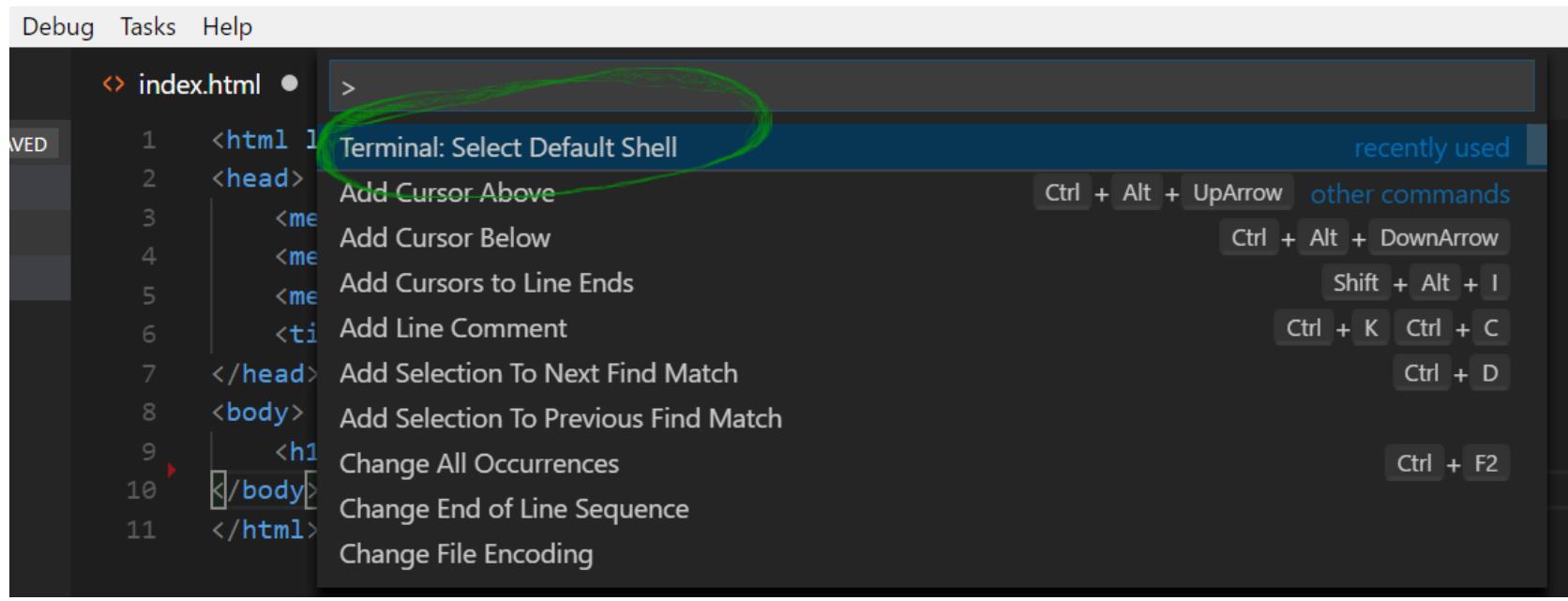
After the integrated terminal pops up →

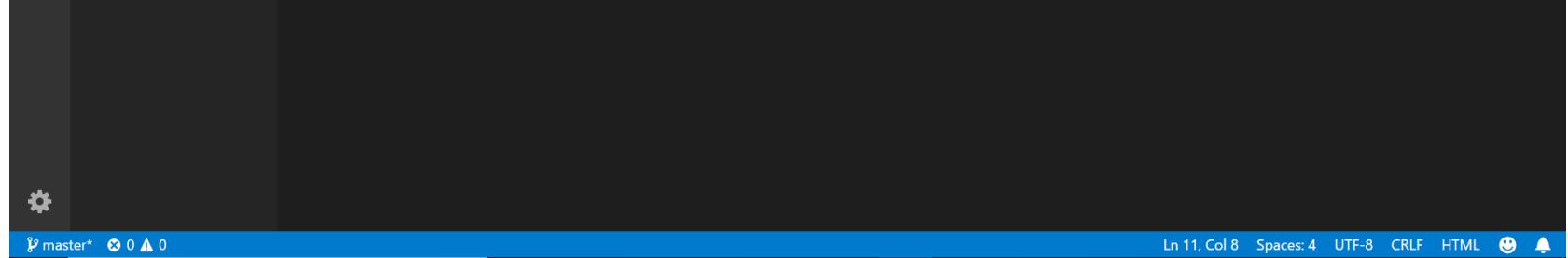
Press the f1 key →

Type the following: Terminal: Select Default Shell →

Click Terminal →

Then select git bash





You know this worked when the dropdown menu in the terminal says
“bash”

You should be seeing something like this:

A screenshot of a terminal window in a dark-themed code editor. The status bar at the top shows 'master*' with 0 errors, 0 warnings, and 0 info messages. It also shows file statistics: Ln 11, Col 8, Spaces: 4, UTF-8, CRLF, HTML, and icons for smiley face and bell notifications. The main area displays a portion of an HTML file with code like '<h1>Hello World!</h1>' and '</body>'. Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. A green oval highlights the TERMINAL tab, which contains a command-line interface showing the user's path 'reyes@DESKTOP-8606PS5 MINGW64 ~/OneDrive/Desktop/project (master)' and a prompt '\$'. The terminal tab is labeled '1: bash'.

8. Create a repository in the root of your project

Click in the blank canvas of the terminal → Type the words: git init

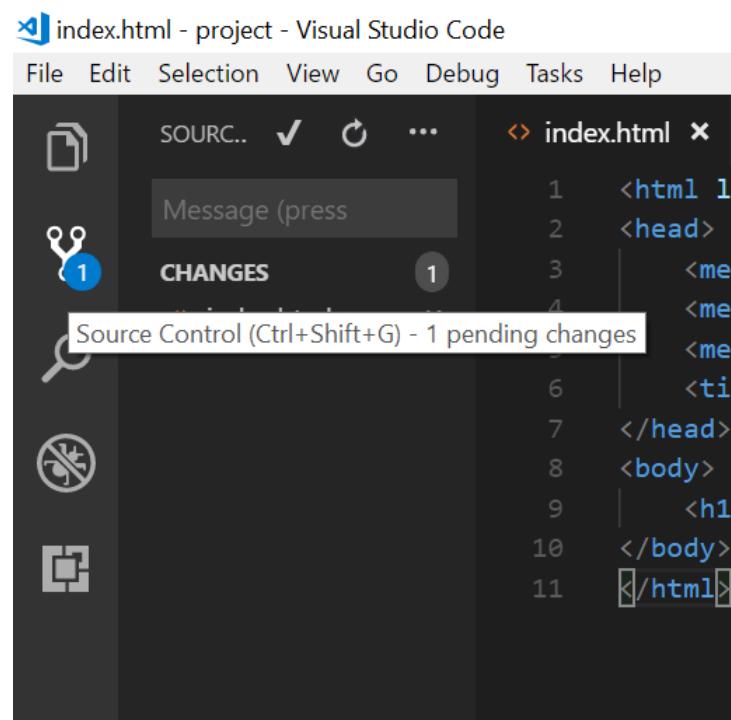
This creates a repository in the root of your project. It should look like this:

A screenshot of a terminal window in a dark-themed code editor. The status bar at the top shows 'master*' with 0 errors, 0 warnings, and 0 info messages. It also shows file statistics: Ln 11, Col 8, Spaces: 4, UTF-8, CRLF, HTML, and icons for smiley face and bell notifications. The main area displays a portion of an HTML file with code like '<h1>Hello World!</h1>' and '</body>'. Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal tab contains a command-line interface showing the user's path 'reyes@DESKTOP-8606PS5 MINGW64 ~/OneDrive/Desktop/project (master)' and the command '\$ git init'. The output of the command is 'Initialized empty Git repository in C:/Users/reyes/OneDrive/Desktop/project/.git/'.

9. Go to the Source Control Section of VSC

One of the many great features of Visual studio code, is it's ability to give you a visualization of version control.

We can check any changes made to our project files by clicking the prong symbol on the left side bar of VSC. You should have something like this:

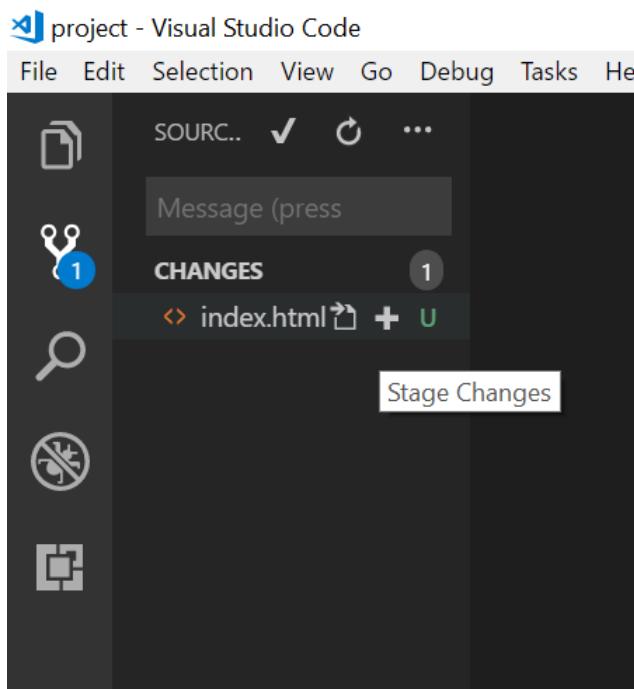


You'll notice that when you hover on the prong symbol, it says pending changes. This indicates that there were recent changes made to your project folder / files. In our case, the reason that the "pending changes" message is there is because we recently created a index.html file.

Is there anything else that would give us hints of recent change? if you look at the sidebar to the right of the prong symbol, you will see that your

index.html file has a letter U, which stands for *Untracked*.

Untracked files are newly created project files that are yet to be staged.



The other letter you will see frequently would be the letter M, which stands for modified. It indicates — just like its name suggests — that a project file has been modified.

A screenshot of the Visual Studio Code interface, similar to the previous one but with a green circle highlighting the "index.html" entry in the Changes list. The list item now has a blue plus sign and a green 'M' icon. The code editor on the right shows the contents of index.html:

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello Worlds!</h1>
</body>
```

Analogy to the rescue

Before we go further, here is an analogy that will make these concepts easier to understand:

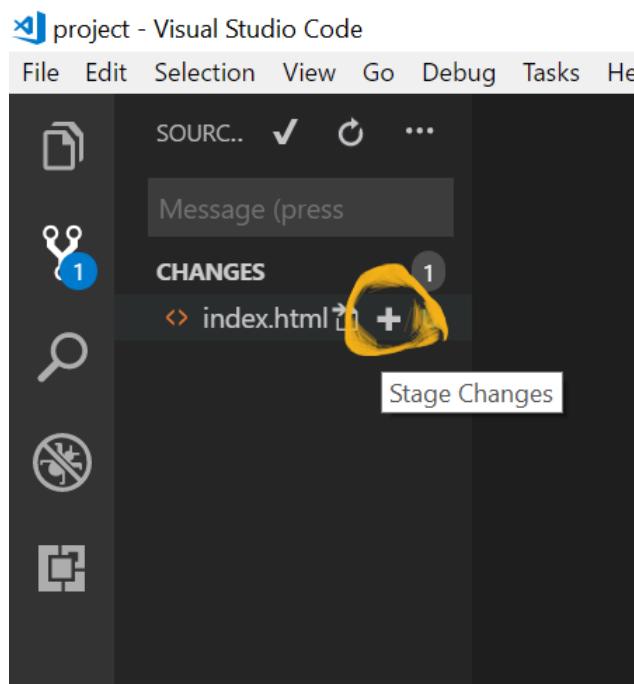
Think of this whole process like a space launch analogy.

In this step, your index.html file is an astronaut preparing to be deployed to a different planet (github site) and right now he is putting on his spacesuit (untracked file waiting to be staged).

10. Add untracked file to stage area

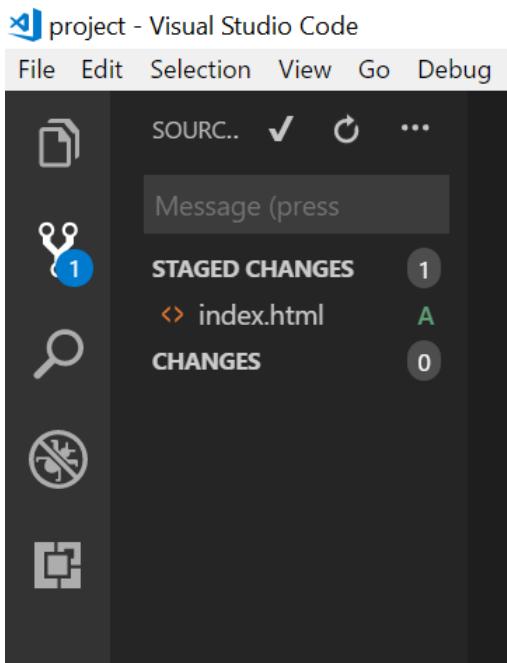
Your astronaut is all fitted and ready. We will now take him to the elevator that leads to the spaceship door.

Hover over the index.html file and a + symbol with the words “stage changes” should appear → Click the + symbol



The index.html file should now be under ‘**Staged Changes**’ and has a letter A with green font beside it, which stands for *Added*.

This means that your astronaut has reached the spaceship door and is ready to enter.

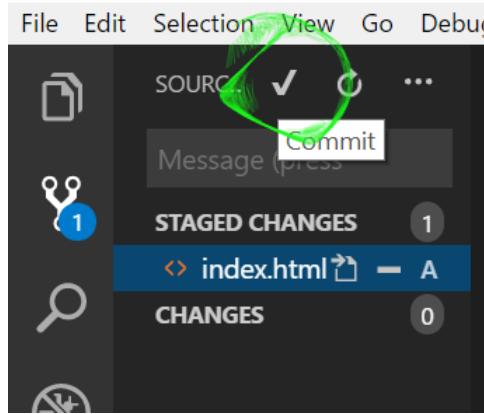


11. Commit the staged files

Now the astronaut has to check that everything is okay, sit down, fasten his seat-belt and wait for launch.

Likewise, you can check your changes by hovering over index.html and clicking open file. Once you’ve determined that the code looks good and you’re happy with the changes, we’re then ready to **commit** our changes (the last step before launch).

In the left sidebar, just above ‘*staged changes*’, click the check symbol →



You will be prompted to create a commit message → type “This is my first commit message” or anything you want → Press Enter

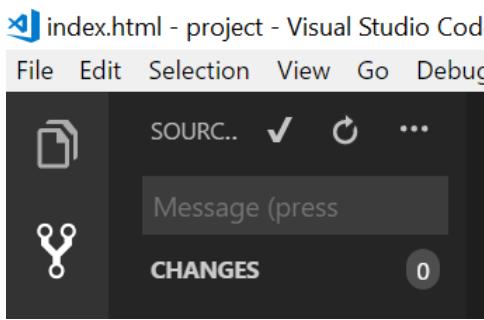
```
This is my first commit message
Please provide a commit message (Press 'Enter' to confirm or 'Escape' to cancel)
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
```

A screenshot of the Visual Studio Code editor showing a commit message being typed into the message input field. The message reads "This is my first commit message". Below the message, a placeholder text says "Please provide a commit message (Press 'Enter' to confirm or 'Escape' to cancel)". The code editor shows the beginning of an HTML file with meta tags and a title tag.

The main purpose of the message is to provide a written documentation of the changes you are making, which will make for easy reference for your future self or any developers you might be collaborating with.

Click here if you want to know the best practices on writing good commit messages.

Now the source control sidebar should be clear and say 0 changes, meaning to say that all astronauts being deployed, have boarded and buckled up.

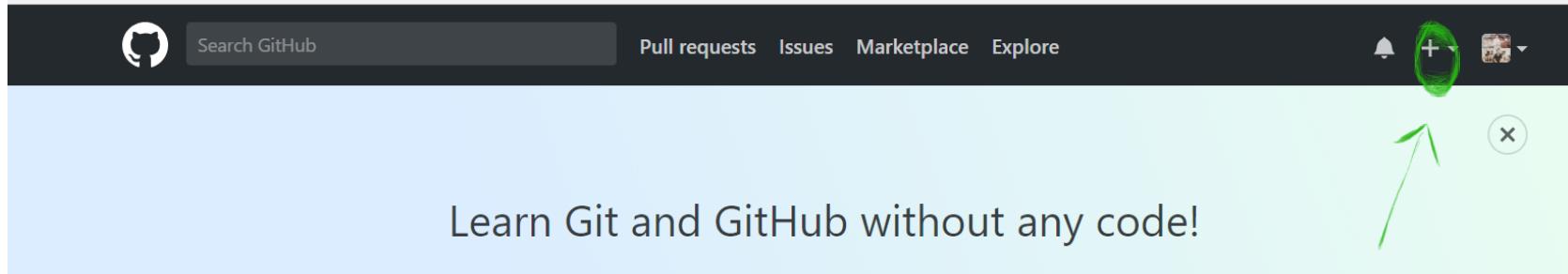




13. Create a new repository on Github

Now that your astronaut is buckled up and ready to go, we need the coordinates to the planet he is landing on.

At the right side of the Github website, click the + icon → click ‘New repository’



14. Set up your repository settings

Create a repository name of your choice →

Scroll down, you can fill in or skip the description (optional) →

Skip, Public repositories cannot be changed to private (it's a paid add-on) →

Check Initialize this repository with a README (optional, I prefer to create my own README file) →

Click ‘Create repository’

M Editing Deploy your project Create a New Repository

GitHub, Inc. [US] https://github.com/new

Owner Repository name
project ✓

Great repository names are short and memorable. Need inspiration? How about cautious-carnival.

Description (optional)
this an example project

Public Anyone can see this repository. You choose who can commit.
 Private You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None

Create repository

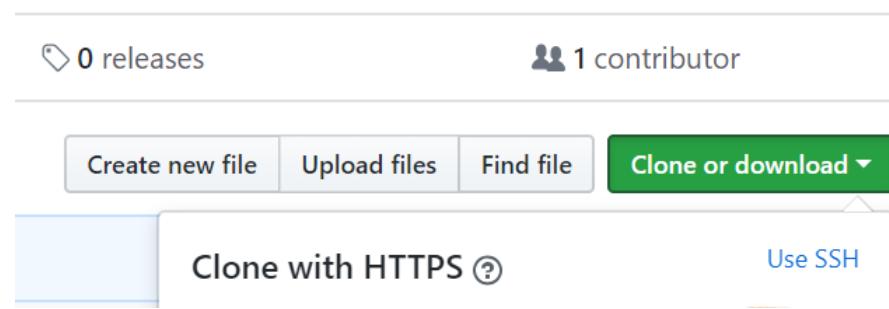
Congratulations you just made your first github repository!!! This calls for a dance break

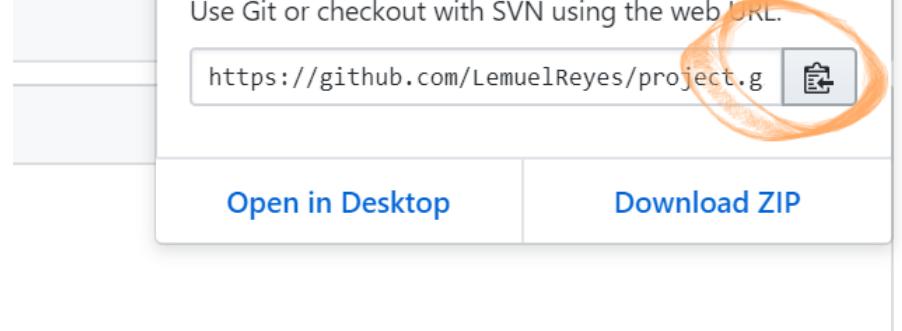


15. Copy the repository "coordinates"

To the right of the screen, click Clone or download →

Click the clipboard icon to copy the repository link





16. Add a remote

Now that we have the planet coordinates, we can input them into the spaceship autopilot system to direct the spaceship's trajectory to the right planet (repository).

Go back to Visual Studio Code → Click View → Click integrated terminal → Run the command: git remote add origin [paste your repository url]

To check if it worked, type git remote -v and it should be pointing to your repository like so:

A screenshot of Visual Studio Code. The code editor shows a simple HTML file with three lines of code: line 9 has '<h1>Hello World!</h1>', line 10 has '</body>', and line 11 has '</html>'. Below the code editor are four tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, displaying a terminal session:

```
reyes@DESKTOP-8606PS5 MINGW64 ~/OneDrive/Desktop/project (master)
$ git init
Initialized empty Git repository in C:/Users/reyes/OneDrive/Desktop/project/.git/
reyes@DESKTOP-8606PS5 MINGW64 ~/OneDrive/Desktop/project (master)
$ git remote add origin https://github.com/LemuelReyes/project.git
reyes@DESKTOP-8606PS5 MINGW64 ~/OneDrive/Desktop/project (master)
$ git remote -v
origin  https://github.com/LemuelReyes/project.git (fetch)
origin  https://github.com/LemuelReyes/project.git (push)

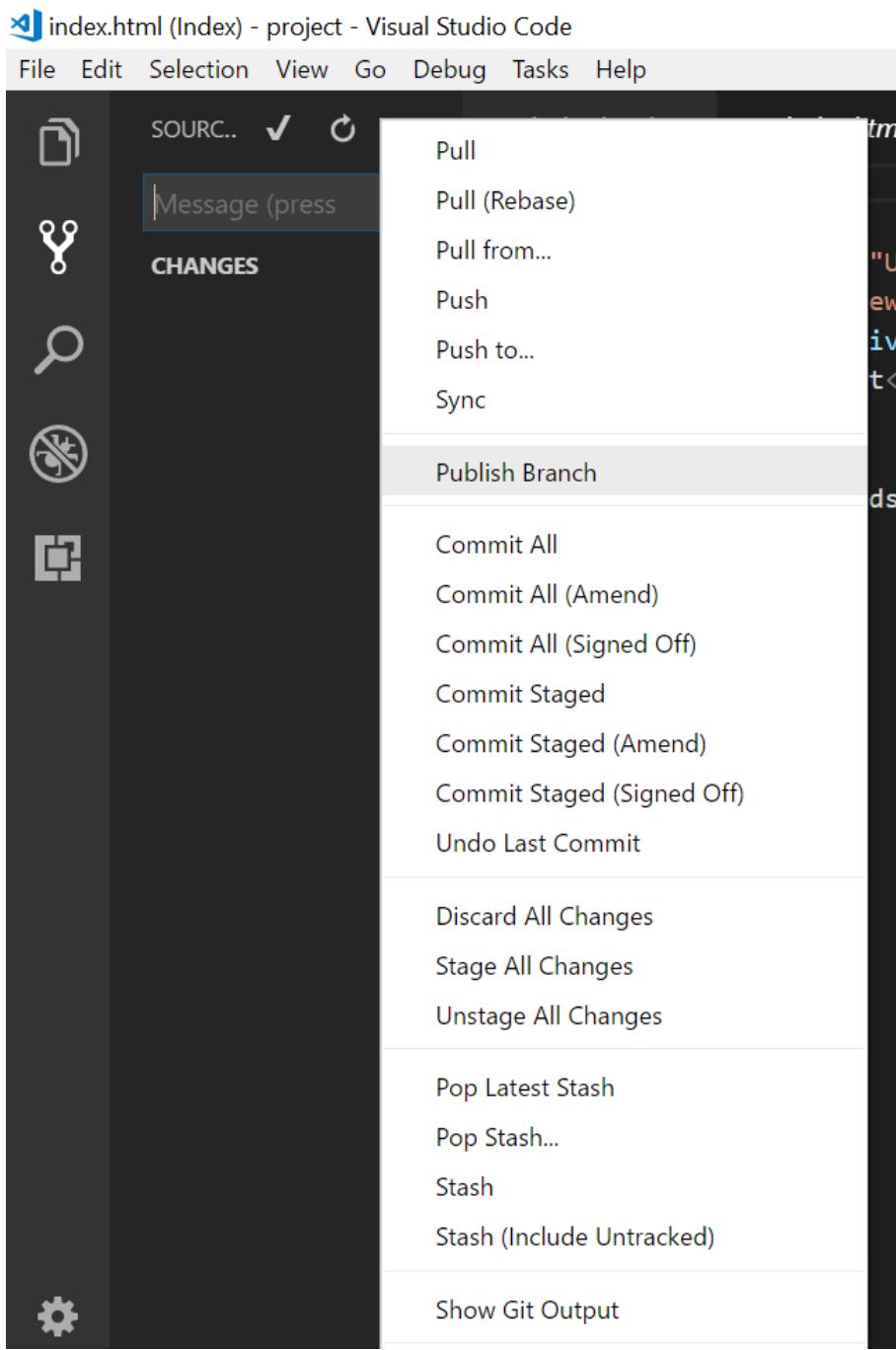
reyes@DESKTOP-8606PS5 MINGW64 ~/OneDrive/Desktop/project (master)
$
```

17. Push your project to Github

Go back to Source Control section →

On the top of the left side bar, there is three dots (...) click on it →

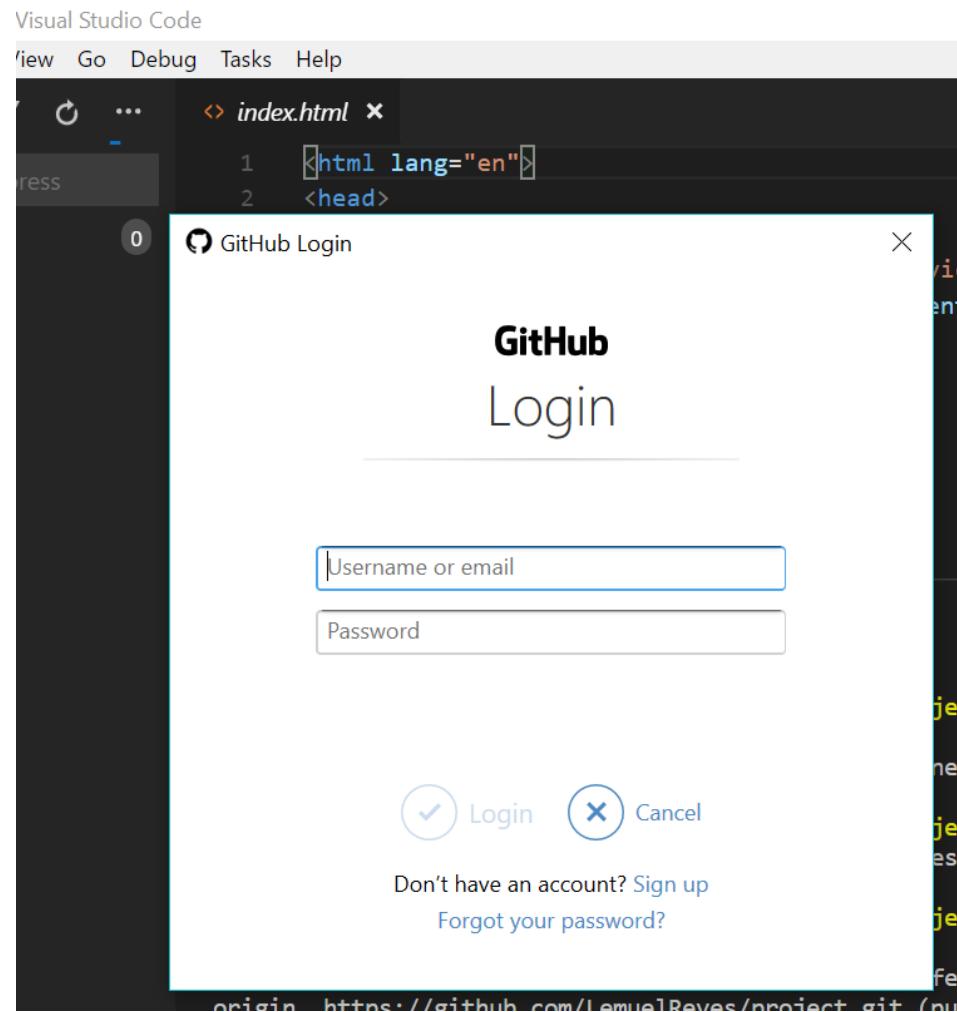
Click Publish Branch→



master 0 1 0 0

Install Additional SCM Providers...

You will be prompted to log in your github account credentials →
Enter credentials



10...9...8...7...6...5...4...3...2...1

And lift off!!

Our astronaut is blasting his way toward planet Github and is projected
(pun intended) to land on your newly created repository!

18. Chest out, chin high because you just published your glorious project

Go back to Github → Scroll down →

On the right sidebar, under Your Repositories, click your project name

You should see this:

This repository

Search

Pull requests Issues Marketplace Explore

Unwatch 1 Star 0 Fork 0

LemuelReyes / project

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Add topics

1 commit 1 branch 0 releases 0 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Lemuel Reyes My first commit message Latest commit 98614b9 18 seconds ago

index.html My first commit message 17 seconds ago

Help people interested in this repository understand your project by adding a README. Add a README

There it is! The commit message that we entered earlier on Step 11.

If you click the commit message in the middle, it will show you the files you published, like so:

My first commit message

Browse files

master

Lemuel Reyes committed a minute ago 0 parents commit 98614b9b47e98007e7451be38f471924f66adf4e

Showing 1 changed file with 11 additions and 0 deletions. Unified Split

11 index.html

```
@@ -0,0 +1,11 @@
1 <html lang="en">
2 <head>
3   <meta charset="UTF-8">
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <meta http-equiv="X-UA-Compatible" content="ie=edge">
6   <title>Document</title>
7 </head>
8 <body>
9   <h1>Hello World!</h1>
10 </body>
11 </html>
```

0 comments on commit 98614b9

Lock conversation

Write Preview

Your astronaut has landed safely!!!

This is a cause to celebrate, let's dance one more time.



19. Add another element on your index.html file

Let's send another astronaut to accompany the one we just deployed.

Below your “Hello World” text, add an `<h2></h2>` tag →

Type “How are you” in the h2 tag→

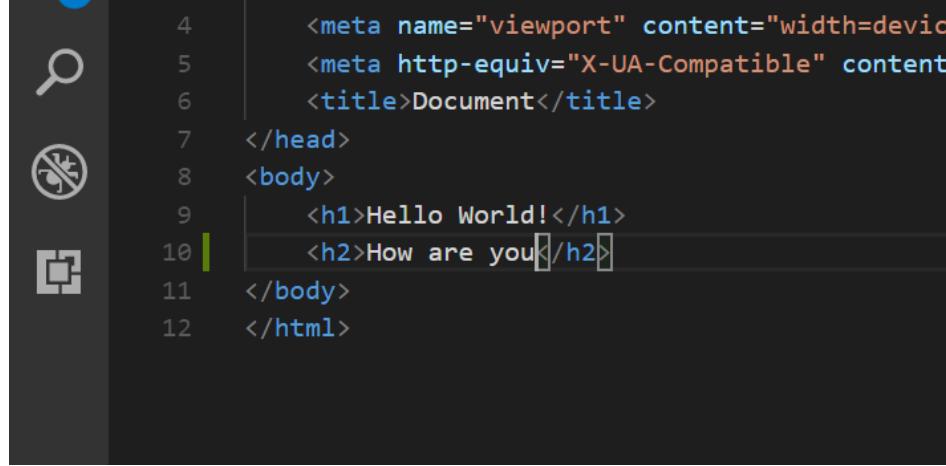
Save →

index.html - project - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

Welcome index.html

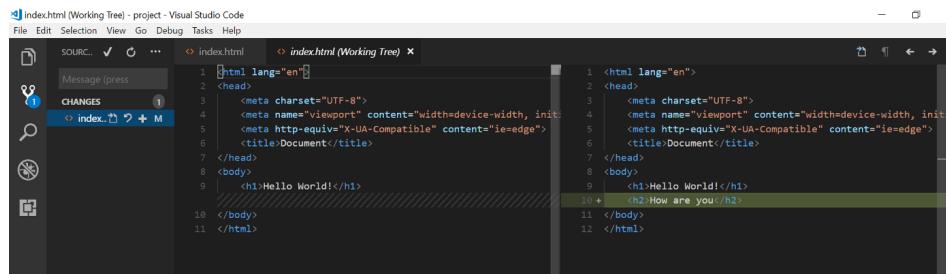
```
1 <html lang="en">
2 <head>
3   <meta charset="UTF-8">
```



```
4 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
5 <meta http-equiv="X-UA-Compatible" content="ie=edge">
6 <title>Document</title>
7 </head>
8 <body>
9 <h1>Hello World!</h1>
10 <h2>How are you</h2>
11 </body>
12 </html>
```

Go to Source Control section →

Review changes (the right side panel is the recent change, you will notice that the h2 tag is in green, this means that you added it recently) →



SOURCE CHANGES index.html (Working Tree)

```
index.html lang="en"
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello World!</h1>
<h2>How are you</h2>
</body>
</html>
```

Stage your newly edited file →

Commit →

Click the three dots →

This time you have to click “push” (anytime you make a change, use push.

Publish branch is only done initially when setting up your repository) →

Check your project and you will see that the new changes are there. Your h2 tag in light green.

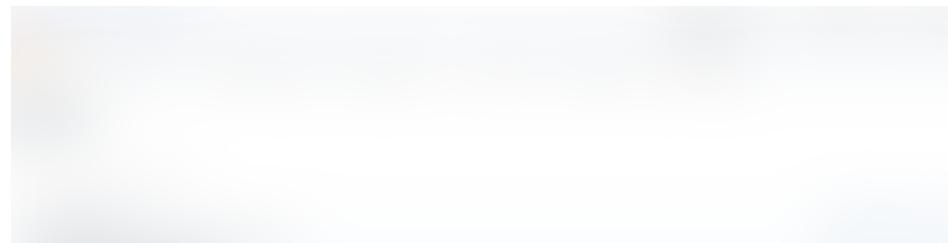
20. Review your project history

In your main project folder there is a clock icon with the word “commits” →
Click on it



You will see each commit you've created and pushed. These are different saved “progress” files which you can revert to — should you ever wish to revert your work to a different point in time. Or you can read your project progress in all its entirety, from beginning to end.

Additionally, if you are collaborating with other developers, they will see the changes you make here and you will see theirs.



There you have it. Congratulations on using Visual Studio, Git, and Github to publish your work online! Pat yourself in the back for me!

Conclusion

Visual Studio Code, Git, and Github are powerful and essential tools to add to your arsenal. They enable you to save your code, create a linear history of your work from beginning to end, collaborate with others and showcase your projects to employers — just to name a few.

My name is Lemuel and I am an aspiring Web Developer. I've created a series titled: The Road To Become A Web Developer. It is the narrative that tells of my personal efforts, struggles, and journey in landing a career in Web Development.

In it you will find personal stories, tutorials of things I've learned, and thoughts on anything tech related. It is with hope, that by documenting the process, that I'll be able to look back and see the remarkable progress I've made. And if it helps someone else, then that's even better!

You can support me in different ways: clapping, sharing my posts, inviting me to guest feature in your Medium, commenting or hitting that follow button. Or If you're feeling generous, keep me awake on this long journey by buying me a cup of coffee. :)

Connect with me on Twitter

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Explore your membership

Thank you for being a member of Medium. You get unlimited access to insightful stories from amazing thinkers and storytellers. Browse

[About](#)

[Help](#)

[Legal](#)