CSY1063
Web Development
Week 4

Chris Rafferty
Chris.Rafferty@northampton.ac.uk

Learning Objectives

- This week we will be covering
  - Version control with Git
  - Branches
  - Merging
  - GitHub
  - Commits

Workflow

- Development on a live website is a bad idea
- Although it’s possible to edit the HTML/CSS live on the website, users will see your changes as you make them
- If something breaks, real people using the website will see it. This is incredibly unprofessional!

Workflow pt2

- Instead, you should run a copy of the live website on your own computer
- This has a lot of benefits:
  - Speed: You don’t need to upload files to a website to see the changes, just save the file and refresh the page
  - You can break things and it doesn’t matter, if a page stops working, it doesn’t matter, nobody other than you will see it.
  - You can test different approaches

Workflow pt3

- Once you have made all the changes you want, you can then upload them to the real website
- Using this methodology you have copies of the code in two different places

Workflow pt4

- Each time you make a change to your local copy it needs uploading to the server
  - More on this later!

Version Control

- When developing software you need to frequently make changes to it
- Sometimes you want to completely replace functionality in the code

Version Control pt2

- One technique for managing this is making a backup of the original file.
- For example back up index.html by copying it to index2.html
- Work on index.html
- If needed restore index2.html

Version Control pt3

- On larger projects this can be very difficult to manage
- Especially when multiple people want to work on the code.

Versions

- During development you'll usually have two versions of any given code:
  - The last known "good configuration" where everything works as intended and is currently running online with real visitors.
  - The "in-development" version that may contain missing code and thing that need to be fixed, this would be running on your computer only

Versions pt2

- In real projects you end up with different scale tasks:
  - Quick updates (e.g. changing some text on the website)
  - New developments (e.g. adding new features like a shopping cart)

Workflow pt5

- In industry, clients will understand the difference between these requests
  - Some big developments can take weeks or months and will be deployed (made live) when it’s ready
  - Some changes need to be done quickly

Making changes

Live Version

```
<main>
    <p>Welcome to our website.</p>
</main>
```

Development Version

```
<main>
    <p>Welcome to our website.</p>
    <p>
        <a href="newproduct.html">
            Click here to see our new product launch!
        </a>
    </p>
</main>
```

Making changes pt2

- Here the development version has some unfinished code that’s not ready to go live (the new product page is not ready yet)
- However, the client might want you to urgently add a notice to the home page

Making changes pt3

Client Request

```
<main>
    <p>Welcome to our website.</p>
</main>
<aside>
    <h2>Closed today!</h2>
    <p>Please note: Due to heavy snow we
    will be unable to open today. You
    can still contact us by email if you
    have any enquires</p>
</aside>
```

Development Version

```
<main>
    <p>Welcome to our website.</p>
    <p>
        <a href="newproduct.html">
            Click here to see our
            new product launch!
        </a>
    </p>
</main>
```

Making changes pt4

- To make this change you need to:
  - Back up your changes to the file e.g. rename it index2.html
  - Download the original index.html
  - Make the quick change to the file
  - Upload the new file
  - Restore your changes (put index2.html back to index.html in your development version)
  - Make the same change to your development version

Making changes pt5

- This can be incredibly difficult to manage in real projects, especially when multiple people are working on the same code
  - What happens with most software!
- You end up with several versions of the file and when you’re ready to “go live” with a feature you will need to manually apply the changes to the most recent version of the file
- This is both time consuming and difficult
- The more frequently things change, the bigger problem this becomes

Version Control Software

- To overcome these problems, software companies use version control software.
- There are several different tools to do this job
- Subversion (SVN) was popular in the early 2000s however it’ use has heavily diminished.

Version Control Software pt2

- In recent years most companies now use a program called Git
- Git was developed by the developer of Linux Kernel to solve version related issues where hundreds of developers were working on the same code at any one time.
- Its more advanced features can be overkill for small projects, however it is still incredibly useful even on tiny projects.

GitHub

- Knowing how to use Git is increasingly more important for software developers due to GitHub
- GitHub is a code hosting platform that allows you to manage code via Git
- The popularity of GitHub has helped Git become the standard Version Control Software in use today
- Most open source projects are hosted on GitHub
- If you want to work on an open source project or release your own code on GitHub you’ll need to understand how to use Git

Git

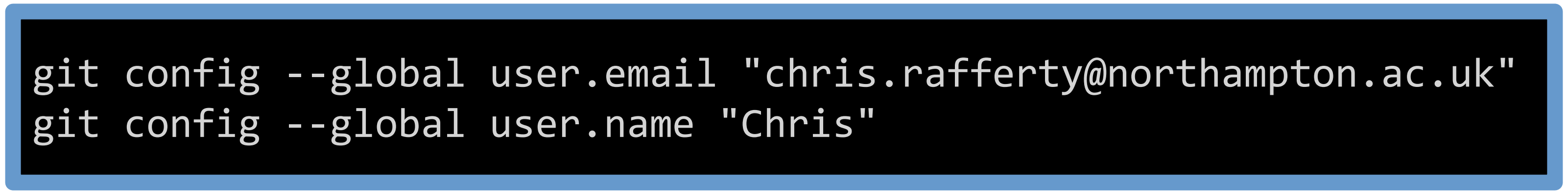
- Git can be confusing at first
- Git is command line driven but there are GUIs available (some are better than others!)
- It’s useful to learn the command line to see what is happening behind the scenes but you can get a basic understanding through a GUI
- Visual Studio Code contains a relatively simple GUI

Git Basics

- Each project/website is placed in a git repository (or repo for short)
- This is a directory that contains all the project files
- In our case, the repo will be the folder that contains your HTML and CSS files

Git Basics pt2

- When you add files to a repository you must be identified as a user
- This is so when you look through a list of changes to the code, you can see who made the change
- Using PowerShell (in any location) you must run 2 commands before saving any files to the repository
  - git config --global user.email “your@email.com“
  - git config --global user.name “Your name”
- With your name/email inside the quotes



```
git config --global user.email "chris.rafferty@northampton.ac.uk"
git config --global user.name "Chris"
```

Git statuses

- There are 3 types of file in git:
  - Unstaged files that have changed, but will not be committed
  - Staged (will be added to the repository)
  - Committed (Currently stored inside the repository)

Adding files

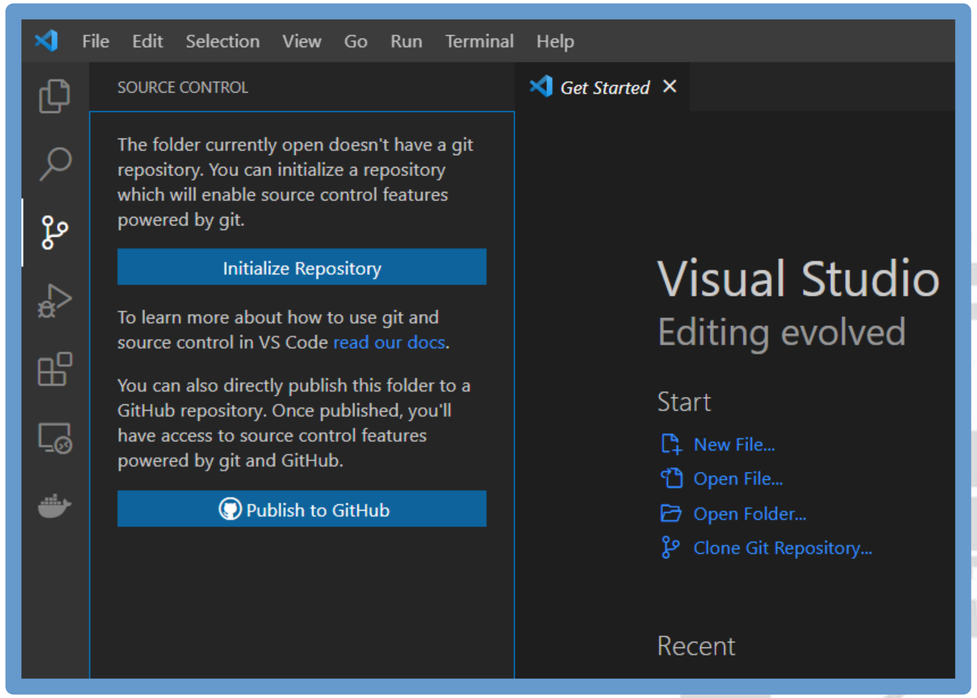
- "Committing" a file (or group of files!) saves a snapshot of those files as they are at that current point in time
- Each "commit" is a snapshot of how the project looks at a particular date/time
- You can go back to different snapshots
  - (in theory… there is actually a better way to manage this)
- It is useful to be able to see file histories

Downloading git

- Download and install git from https://git-scm.com

Initializing a repository

- Firstly we need to initialize a repository
- We can achieve this by using the source control tab in Visual Studio Code
- You then need to click on Initialize Repository
  - You need to be in a folder for this option to be available.



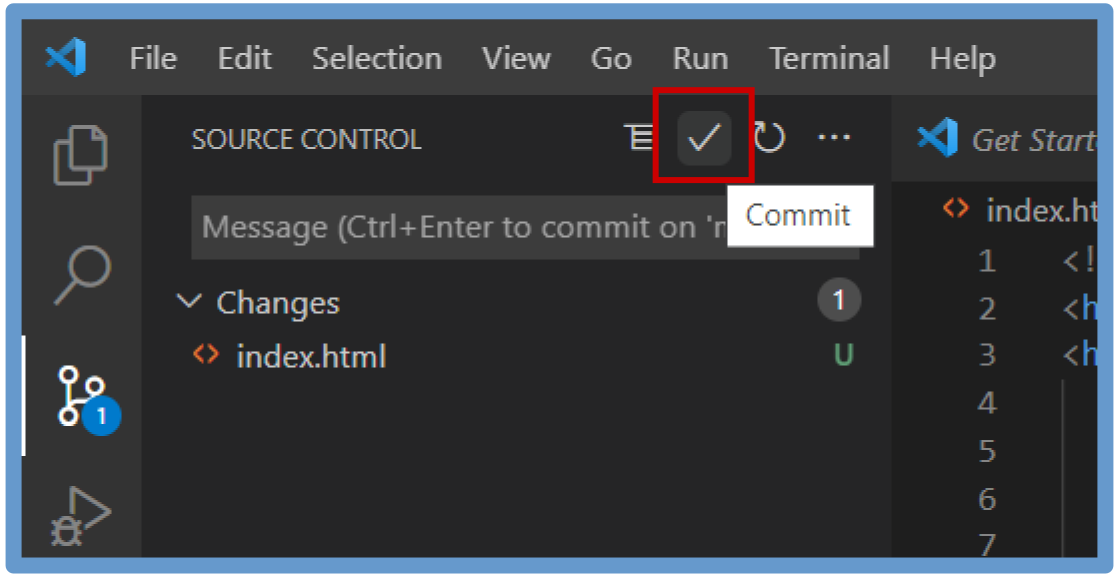
The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.

To learn more about how to use git and source control in VS Code read our docs.

You can also directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

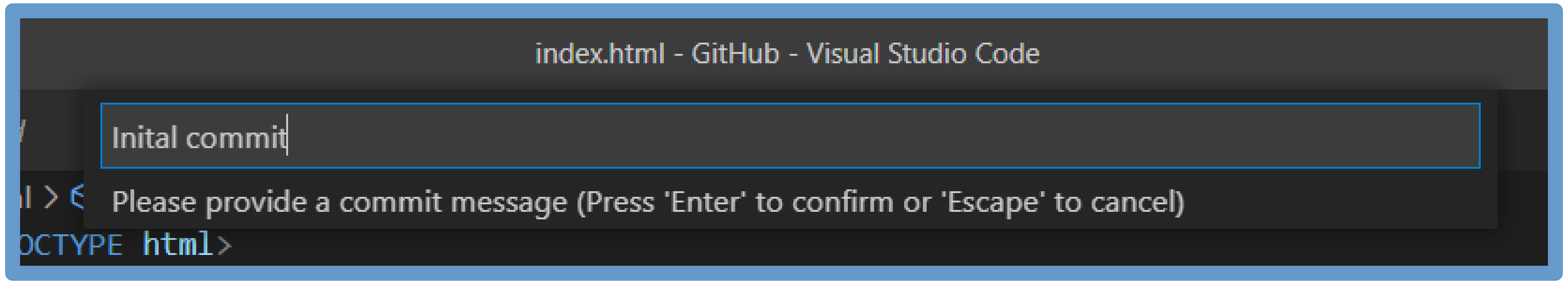
Taking a snapshot

- You need to have set your email and name in PowerShell before attempting this step (slide 23)
- You can take a snapshot of the code as it is right now by clicking the tick icon
  - This will commit that file and take a snapshot



Taking a snapshot pt2

- You will then be asked to provide a commit message
- This needs to be a meaningful message that describes the changes that were added
- As this is the first commit a meaningful name could be “Initial commit”



index.html - GitHub - Visual Studio Code

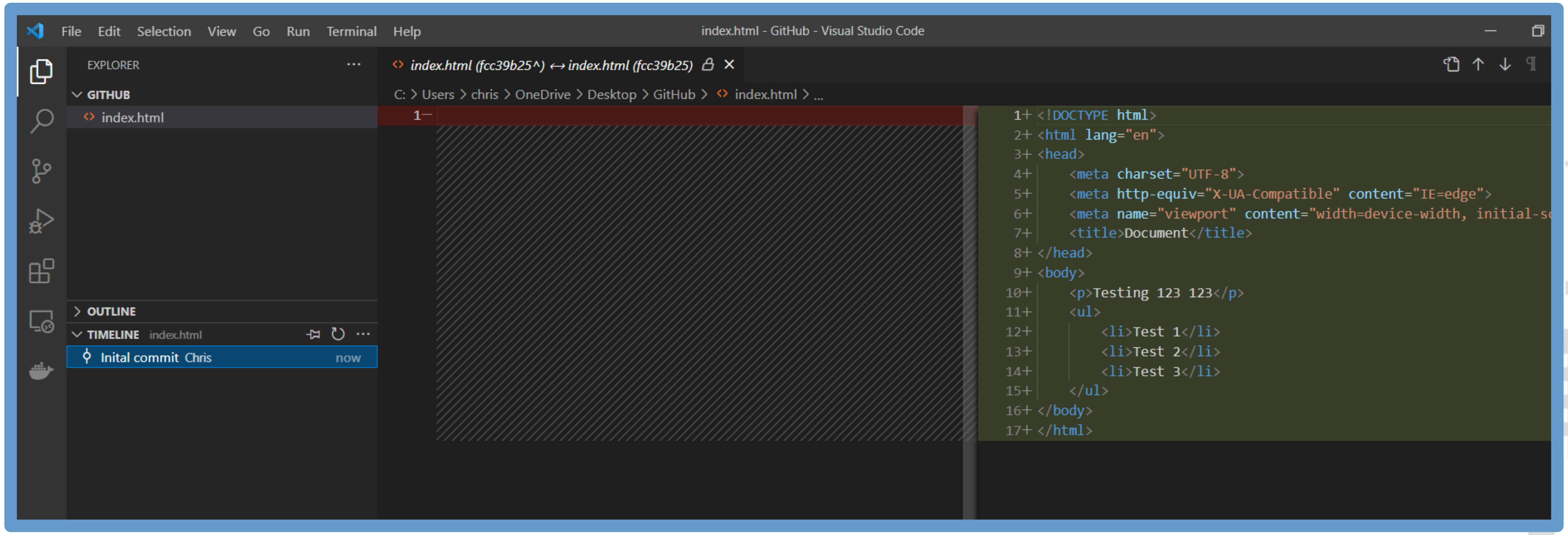
Inital commit

Please provide a commit message (Press 'Enter' to confirm or 'Escape' to cancel)

```
OCTYPE html>
```

Timeline

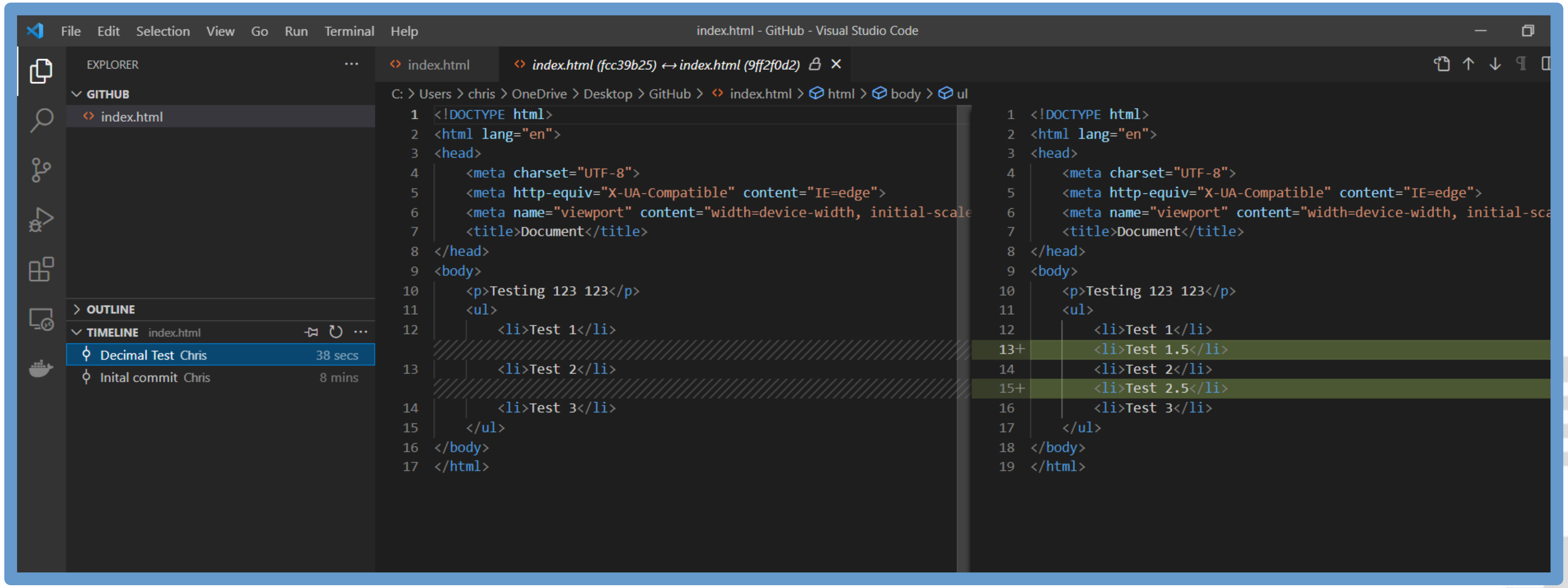
- If you go to the Explorer tab and look at the timeline you can see what was changed in that commit, in this example we went from having no code to then adding our index file (initial commit)


File Edit Selection View Go Run Terminal Help
index.html - GitHub - Visual Studio Code
EXPLORER
GITHUB
index.html
OUTLINE
TIMELINE index.html
Inital commit Chris now
index.html (fcc39b25^) ↔ index.html (fcc39b25)
C: > Users > chris > OneDrive > Desktop > GitHub > index.html > ...

```
<!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-s
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```

Timeline pt2

- The timeline shows us exactly what was added in each commit
- You can see that in this new commit (Decimal Test) the new list items are highlighted in green.


File Edit Selection View Go Run Terminal Help
index.html - GitHub - Visual Studio Code
EXPLORER
GITHUB
index.html
OUTLINE
TIMELINE index.html
Decimal Test Chris 38 secs
Inital commit Chris 8 mins
index.html
index.html (fcc39b25) ↔ index.html (9ff2f0d2)
C: > Users > chris > OneDrive > Desktop > GitHub > index.html > html > body > ul

```
<!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
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <ul>
        <li>Test 1</li>

        <li>Test 2</li>

        <li>Test 3</li>
    </ul>
</body>
</html>
```

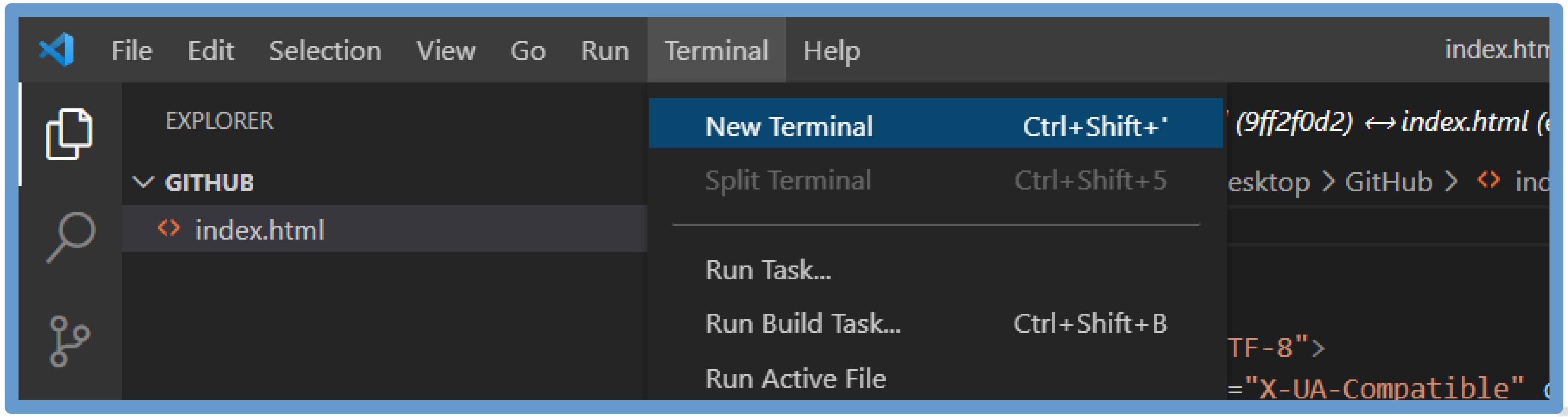
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-sca
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <ul>
        <li>Test 1</li>
+       <li>Test 1.5</li>
        <li>Test 2</li>
+       <li>Test 2.5</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```

Timeline pt3

- With each new commit you are able to see how the files change over time and you can do this for each and every file
- Although we are able to view each of these snapshots there is not an easy way to go back to that version of the code.
- There is a better approach called branches (more on that later!)

Revert

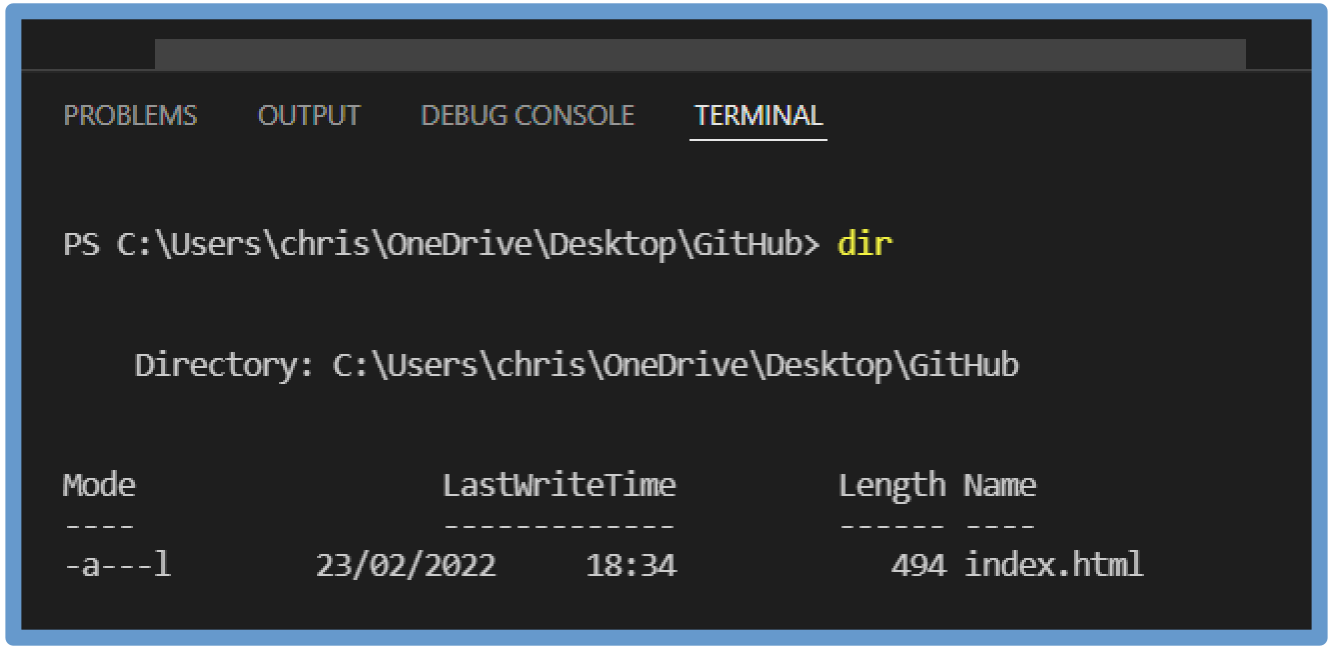
- If we wanted to remove code added in a commit we can use a revert
- This will remove that portion of code while leaving the rest intact (any code added in future commits will remain intact as well)
- We first need to open a terminal



```
TF-8">
="X-UA-Compatible"
```

Revert pt2

- If you type ‘dir’ you can see all the files in the repository
- You can do this to check your in the right directory
- In this example its just the index file

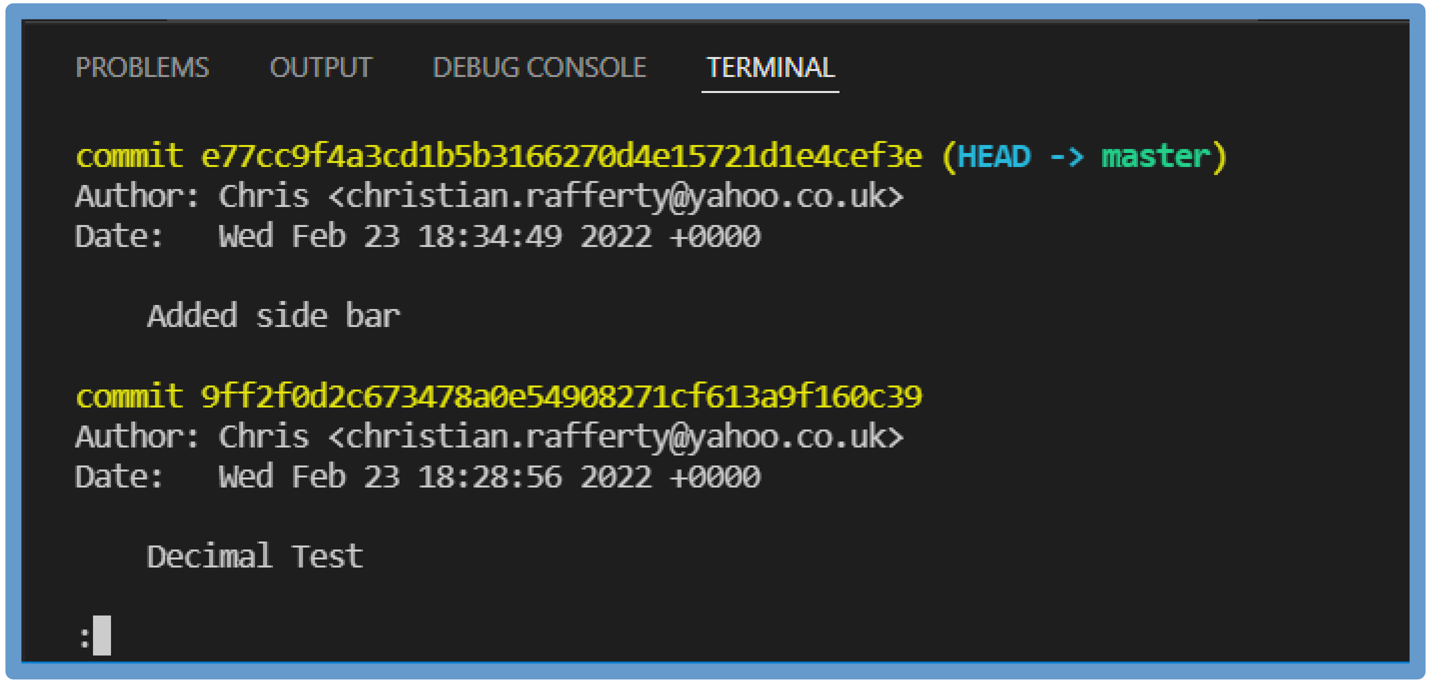


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\chris\OneDrive\Desktop\GitHub> dir


    Directory: C:\Users\chris\OneDrive\Desktop\GitHub


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---l        23/02/2022     18:34            494 index.html
```

Revert pt3

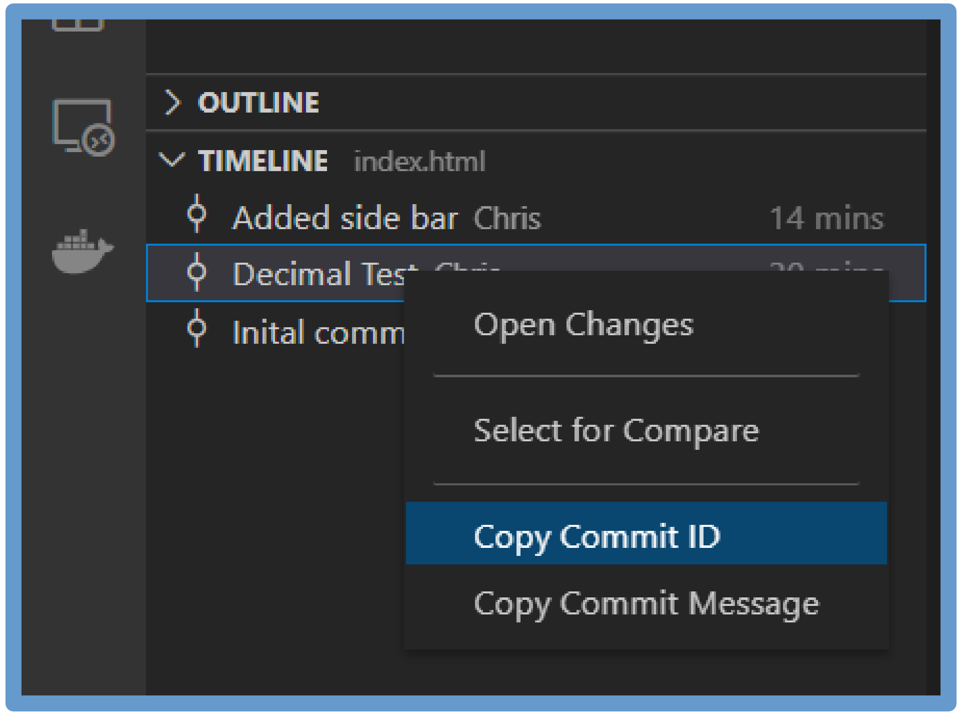
- You can type ‘git log’ to see all the commits you have made
  - If you get stuck in this section press q and it will bring you back to the terminal



```
commit e77cc9f4a3cd1b5b3166270d4e15721d1e4cef3e (HEAD -> master)
Author: Chris <christian.rafferty@yahoo.co.uk>
Date:   Wed Feb 23 18:34:49 2022 +0000

    Added side bar

commit 9ff2f0d2c673478a0e54908271cf613a9f160c39
Author: Chris <christian.rafferty@yahoo.co.uk>
Date:   Wed Feb 23 18:28:56 2022 +0000

    Decimal Test

:
```

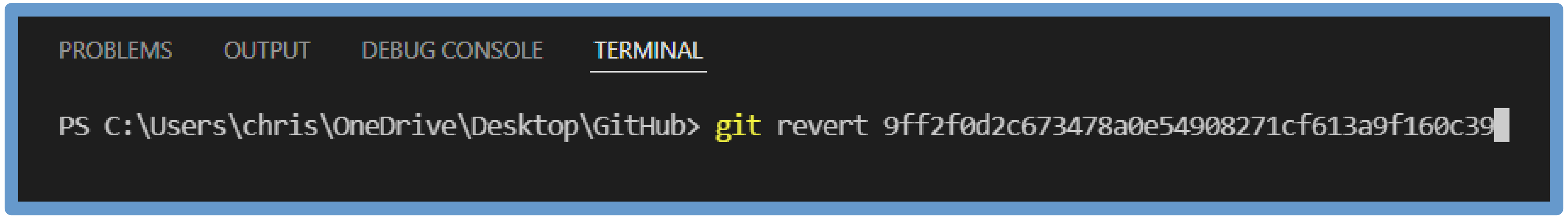
Revert pt4

- We need to get the ID of the commit we want to revert and the easiest way to do this is to right click on the commit in the timeline



Revert pt5

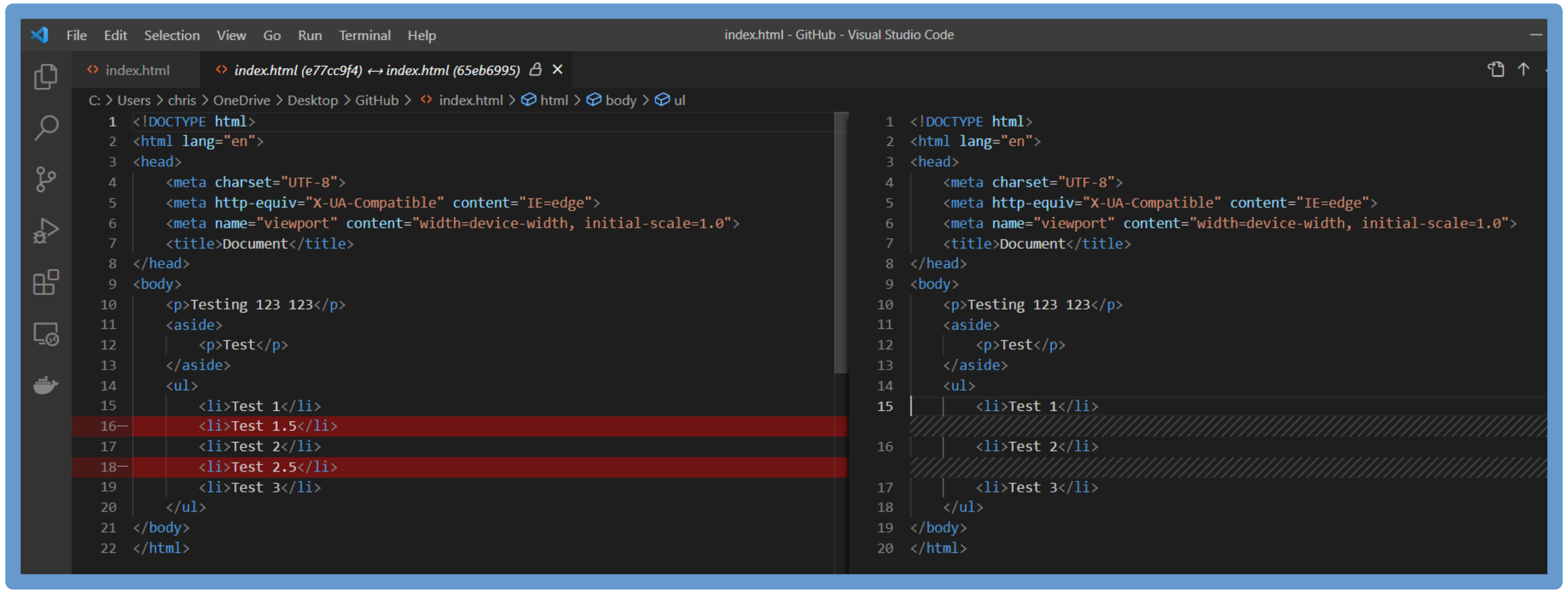
- Once you have the ID of the commit you can then type in
  - git revert theID



```
PS C:\Users\chris\OneDrive\Desktop\GitHub> git revert 9ff2f0d2c673478a0e54908271cf613a9f160c39
```

Revert pt6

- By using revert you can see that it has removed the code
- It’s important to note that the code we added after that commit has remained
- Any code added in other commits will remain unaffected


File Edit Selection View Go Run Terminal Help
index.html - GitHub - Visual Studio Code
index.html
index.html (e77cc9f4) ↔ index.html (65eb6995)
C: > Users > chris > OneDrive > Desktop > GitHub > index.html > html > body > ul

```html
<!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>
    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 1.5</li>
        <li>Test 2</li>
        <li>Test 2.5</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```

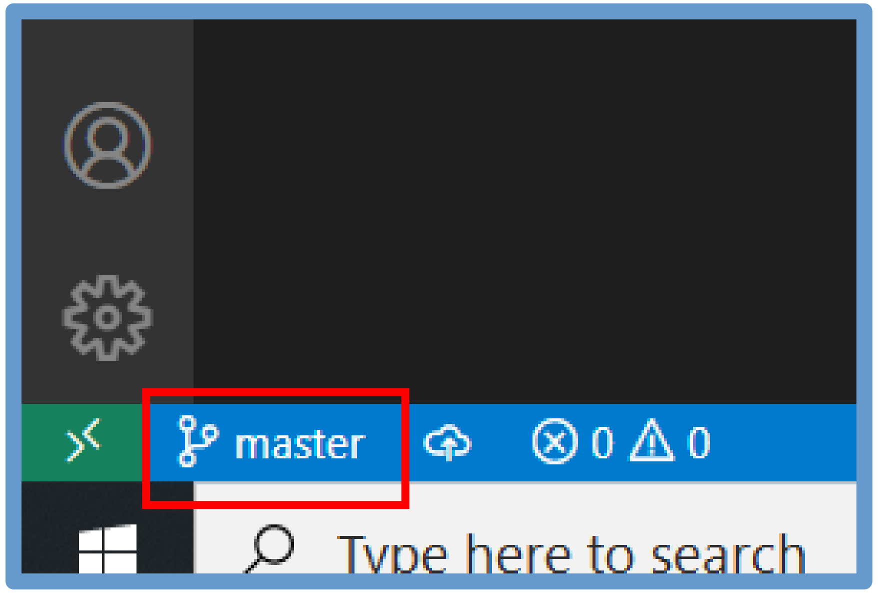
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>

        <li>Test 2</li>

        <li>Test 3</li>
    </ul>
</body>
</html>
```

Revert pt7

- This means we can make reversions regardless of the order of commits
- If you have 10 commits you could revert the second while leaving the rest of the code from the later commits unaffected.
- Revert is not undo

Branches

- Most of the time you don’t want to revert and redo commits and this is why Visual Studio Code doesn’t have functionality for that
- There is a much better way of solving this problem called branches
- You can manage this using the button at the bottom that says master

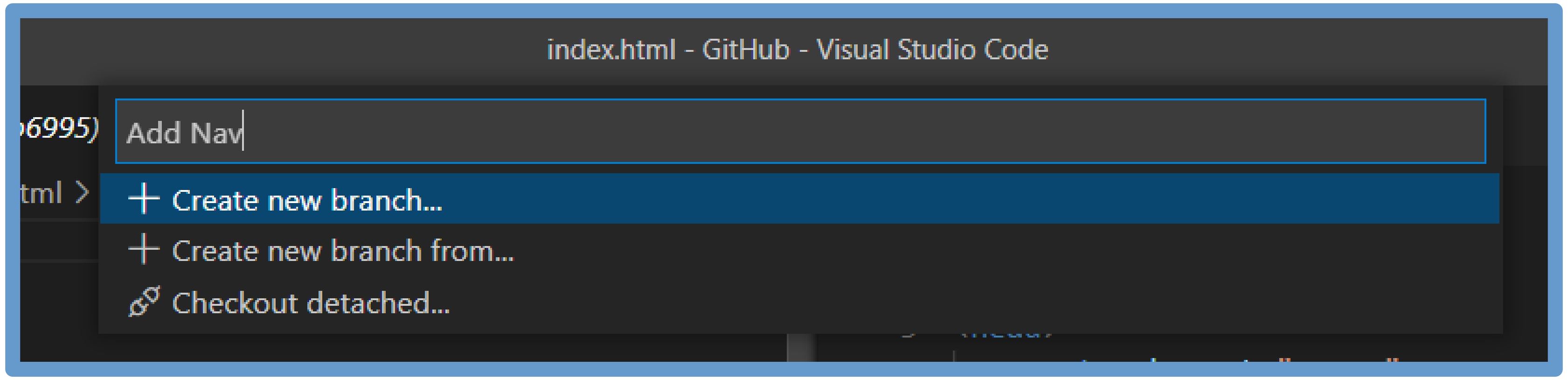


Branches pt2

- Master is the main branch of this code
- A branch is like a different folder of the code
  - Its similar to copying all the code and then working on that copy of the code
  - The original version will remain unaffected while you work on the copied version
  - You can have as many different branches (versions of the code) as you like

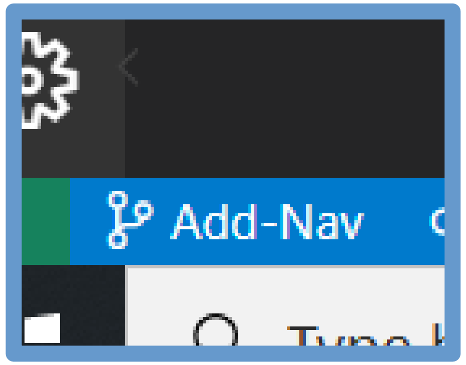
Branches pt3

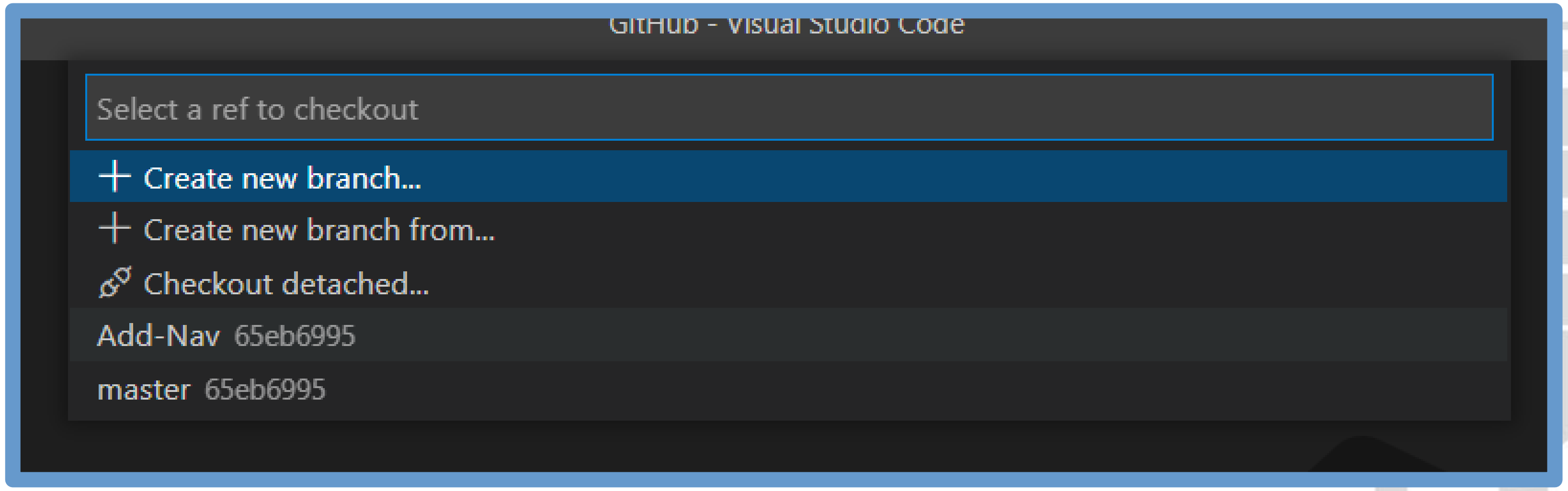
- To add a new branch click on the master button
- You then need to give the new branch a meaningful name
  - In this example we will be adding navigation to our website so the branch could be called Add Nav


index.html - GitHub - Visual Studio Code
Add Nav
+ Create new branch...
+ Create new branch from...
Checkout detached...

Branches pt4

- If you click on the branches button again you can see all the branches in this code
  - You can see the master (main/original code) and the Add Nav (the new branch)
  - You can see which branch you are currently working on by looking at the branch button



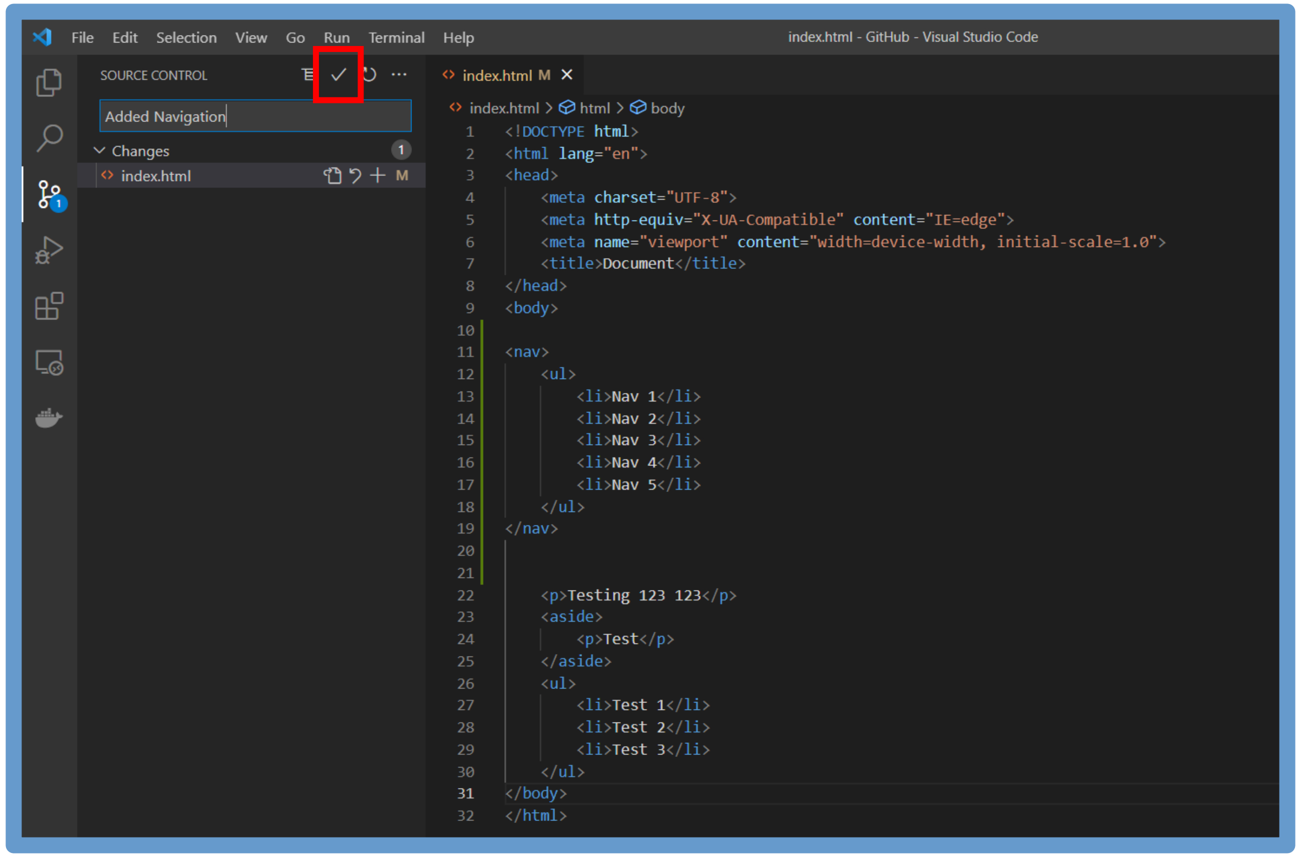

GitHub - Visual Studio Code

Select a ref to checkout

+ Create new branch...
+ Create new branch from...
Checkout detached...
Add-Nav 65eb6995
master 65eb6995

Branches pt5

- Using the Add Nav branch we are going to add some new lines of code and commit



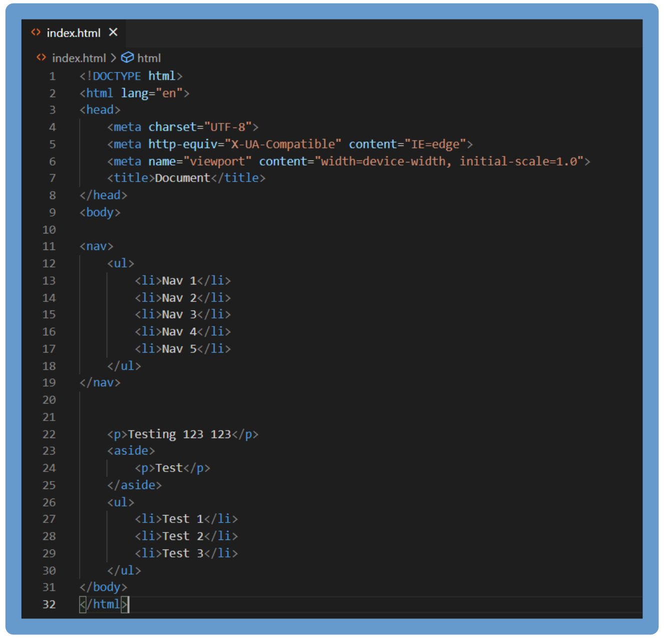
File Edit Selection View Go Run Terminal Help
index.html - GitHub - Visual Studio Code
SOURCE CONTROL
Changes
index.html
index.html M
index.html > html > body

Added Navigation

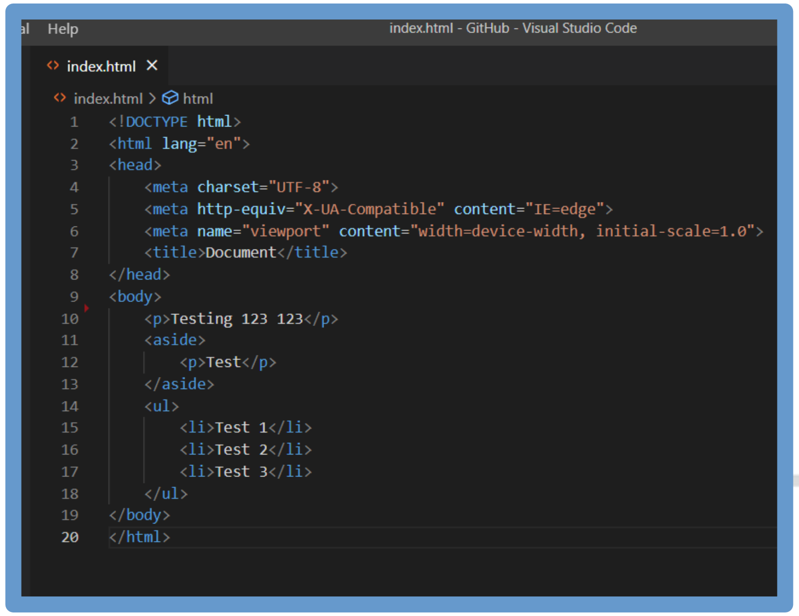
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

<nav>
    <ul>
        <li>Nav 1</li>
        <li>Nav 2</li>
        <li>Nav 3</li>
        <li>Nav 4</li>
        <li>Nav 5</li>
    </ul>
</nav>


    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```

Branches pt6

- Because we are working in the new branch that code hasn't been added to the master branch and we can easily go from one branch to the other to see the code that's been added



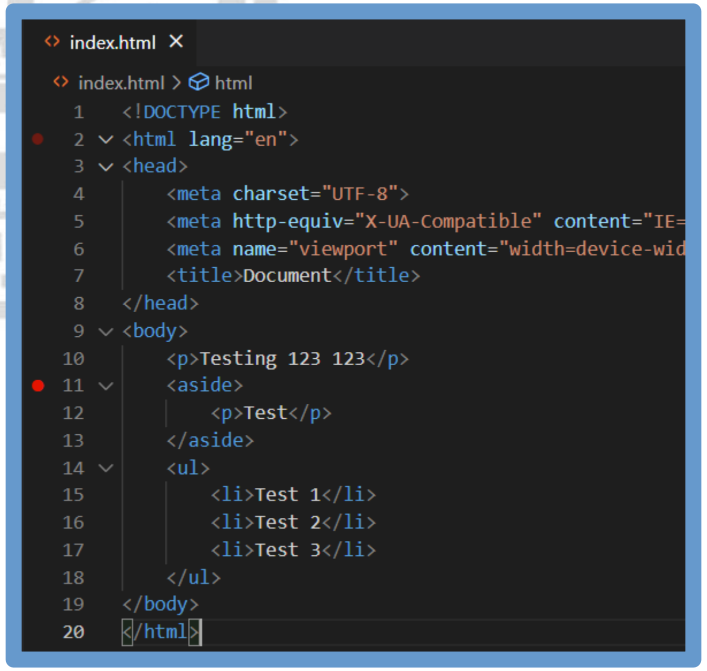
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

<nav>
    <ul>
        <li>Nav 1</li>
        <li>Nav 2</li>
        <li>Nav 3</li>
        <li>Nav 4</li>
        <li>Nav 5</li>
    </ul>
</nav>


    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```



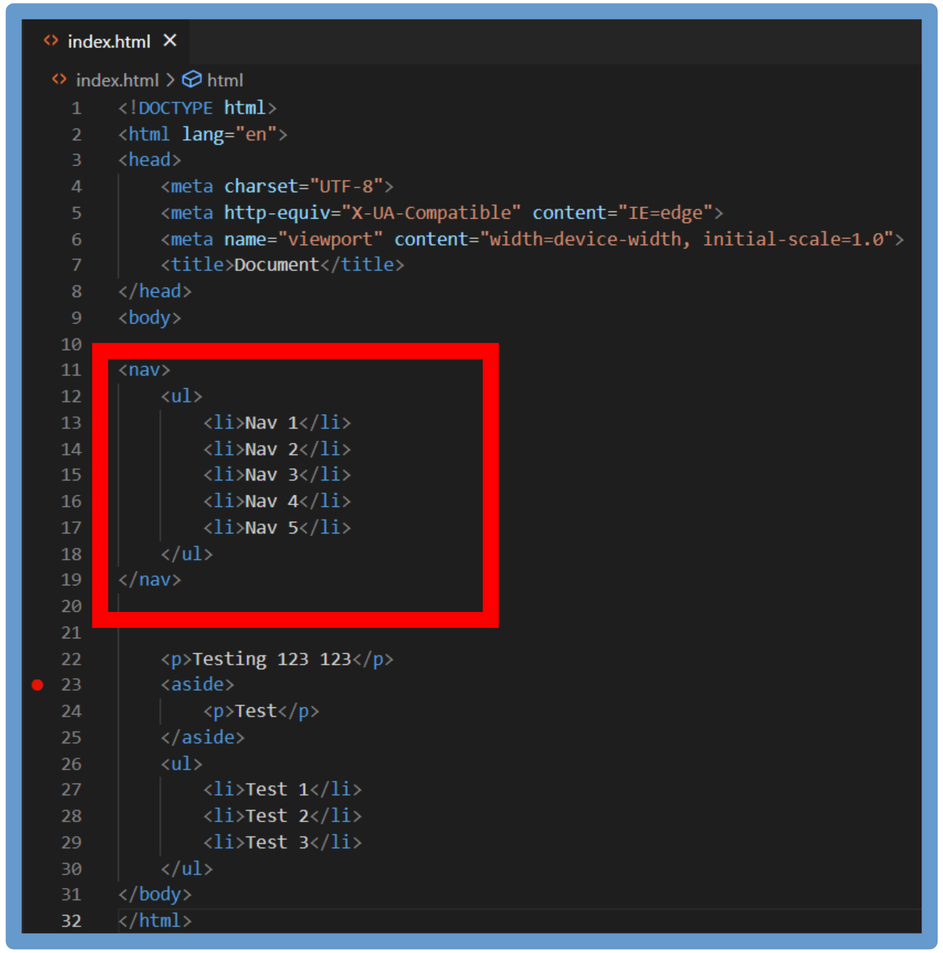
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```

Branches pt7

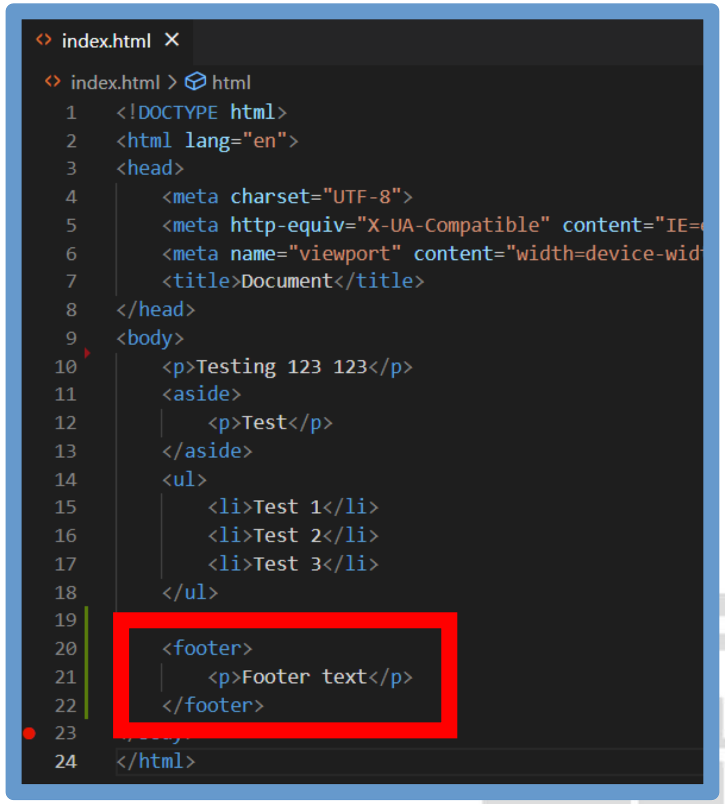
- If we added another branch copied from the master branch (doesn’t include the nav code) we now have three different versions of the code



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=
    <meta name="viewport" content="width=device-wid
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```



```html
<!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>

<nav>
    <ul>
        <li>Nav 1</li>
        <li>Nav 2</li>
        <li>Nav 3</li>
        <li>Nav 4</li>
        <li>Nav 5</li>
    </ul>
</nav>


    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=
    <meta name="viewport" content="width=device-wid
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>

    <footer>
        <p>Footer text</p>
    </footer>
</body>
</html>
```

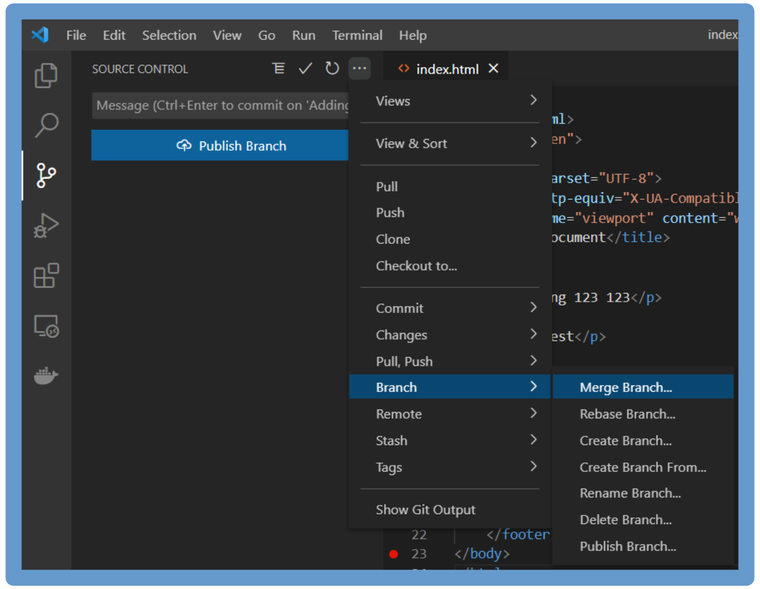
master

Add-Nav

Adding-Footer

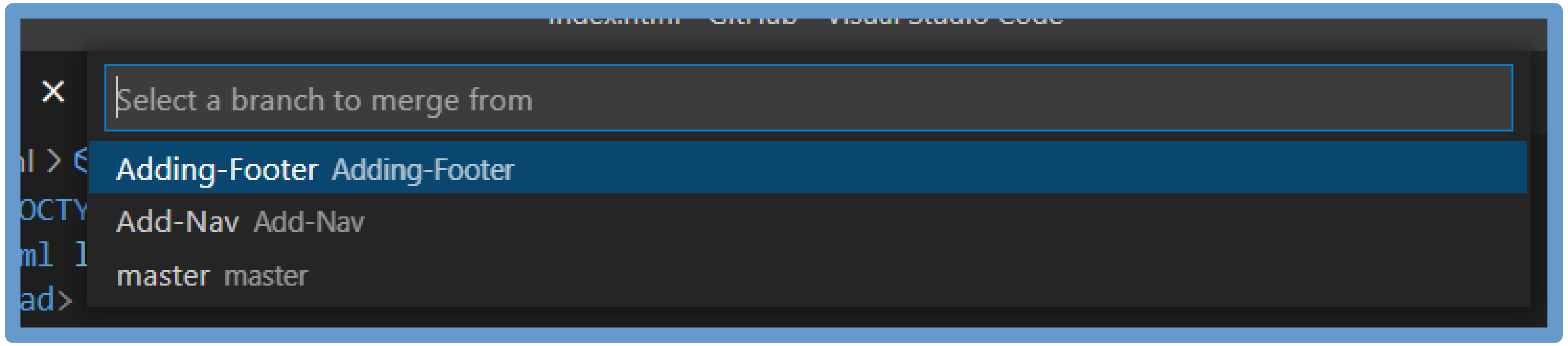
Branches pt8

- The problem now is each branch has different changes
- If we want both of these changes (nav and footer) in the master branch (original code) we can do something called merging the branches.
- To do this we go to the source control tab



Branches pt9

- We should first be in the master branch so we can then merge one of the other branches into master
- Once we select the merge option we then select the branch to merge with



Select a branch to merge from

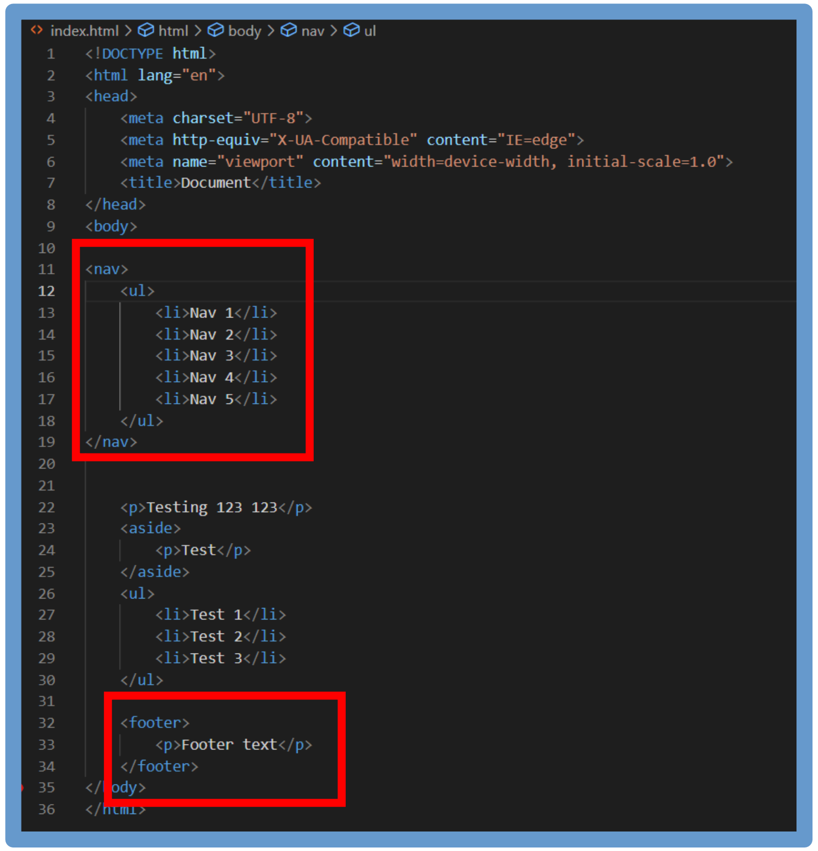
Adding-Footer Adding-Footer

Add-Nav Add-Nav

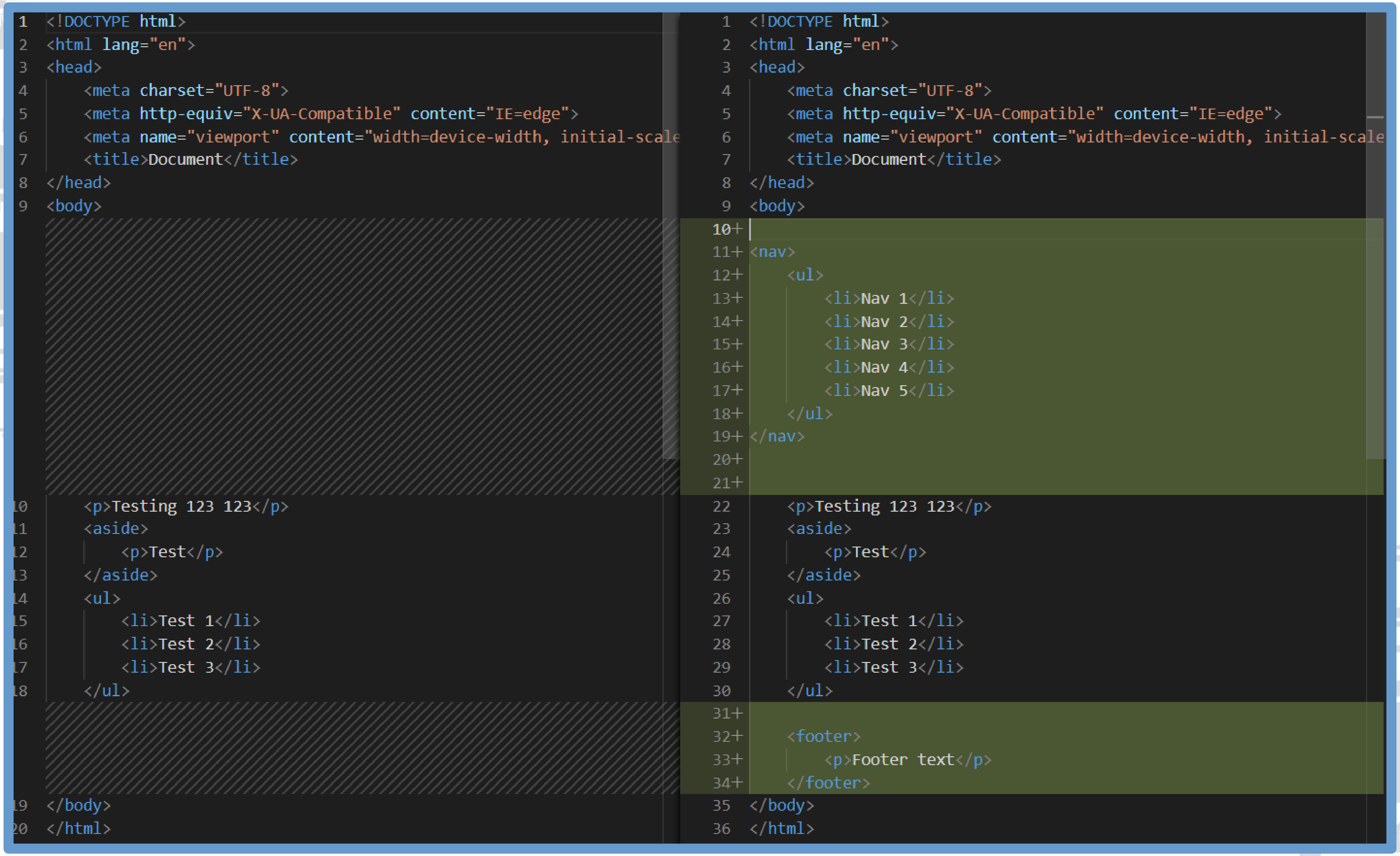
master master

Branches pt10

- If you merge both branches into master you can see that both the changes have been added



```html
<!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>

<nav>
    <ul>
        <li>Nav 1</li>
        <li>Nav 2</li>
        <li>Nav 3</li>
        <li>Nav 4</li>
        <li>Nav 5</li>
    </ul>
</nav>


    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>

    <footer>
        <p>Footer text</p>
    </footer>
</body>
</html>
```



```html
<!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
    <title>Document</title>
</head>
<body>
    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>
</body>
</html>
```

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

<nav>
    <ul>
        <li>Nav 1</li>
        <li>Nav 2</li>
        <li>Nav 3</li>
        <li>Nav 4</li>
        <li>Nav 5</li>
    </ul>
</nav>


    <p>Testing 123 123</p>
    <aside>
        <p>Test</p>
    </aside>
    <ul>
        <li>Test 1</li>
        <li>Test 2</li>
        <li>Test 3</li>
    </ul>

    <footer>
        <p>Footer text</p>
    </footer>
</body>
</html>
```

Branches pt11

- It takes some time to get used but this is the industry standard for working on projects
- It is recommended you use this approach in your assignment
- When you need to add a new page/feature create a new branch and then merge with the master once it has been completed
- This is a good way of keeping track of all the changes to that code

Exercises

- Install git (https://git-scm.com)
- See slide 23. Open a terminal and run the two commands to set your name and email address
  - ➢git config --global user.email "YOUR EMAIL"
  - ➢git config --global user.name "YOUR NAME"
- Initialise a git repository in your website folder
- Commit your initial project
- Add some new HTML (like a p tag or a h1 tag)

Exercise pt2

- From the master branch create a new branch called “changes”
- Add something new to the HTML like a p tag or a h1 tag
- Commit the change
- Switch between the master and “changes” branches so you can see the different versions
- Switch to the master branch and create another new branch called “more-changes”
  - You should now have the master branch and 2 branches you have created yourself

Exercise pt3

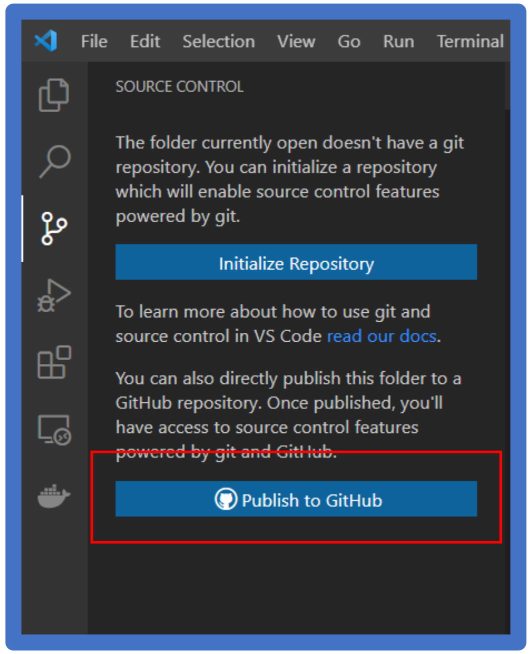
- Add another change to the HTML in your new more-changes branch
- Commit your changes
- Switch between all three branches so you can see how it works
  - All three branches should have different bits of code
- Switch back to master
- Merge the changes from the other two branches into the master branch

GitHub introduction

- GitHub allows you to host the code you create online including all the changes (branches & commits)
- https://github.com/
- You will need to create an account
- If you already have an account just sign in

Connecting to GitHub

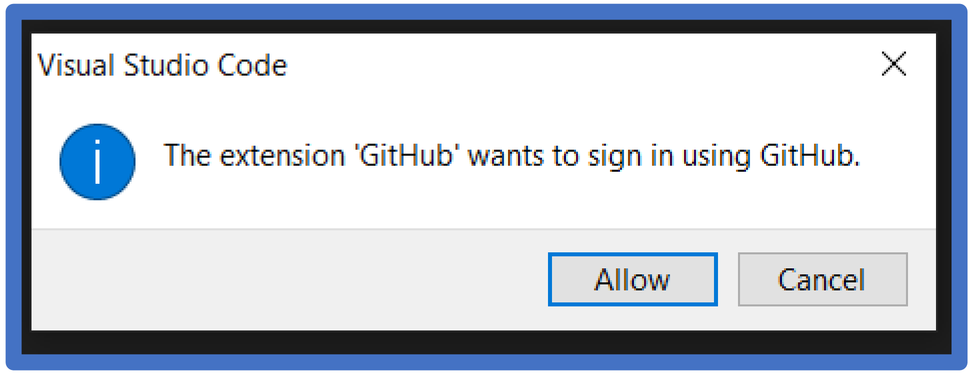
- Open Visual Studio Code and create a new folder
- You will need to have a file (index.html)
- We then need to go to the Source Control tab and select the Publish to GitHub option



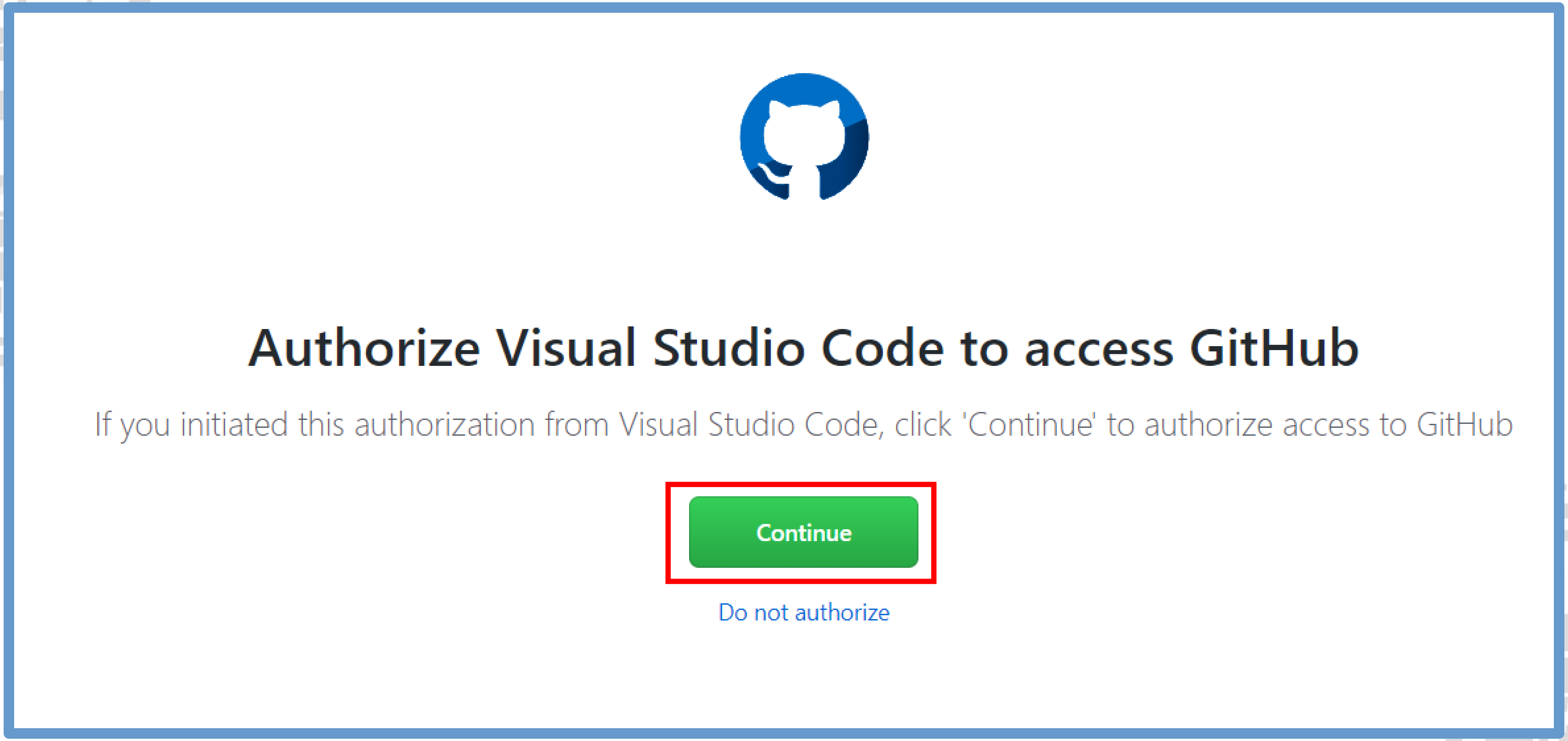
The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.

To learn more about how to use git and source control in VS Code read our docs.

You can also directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

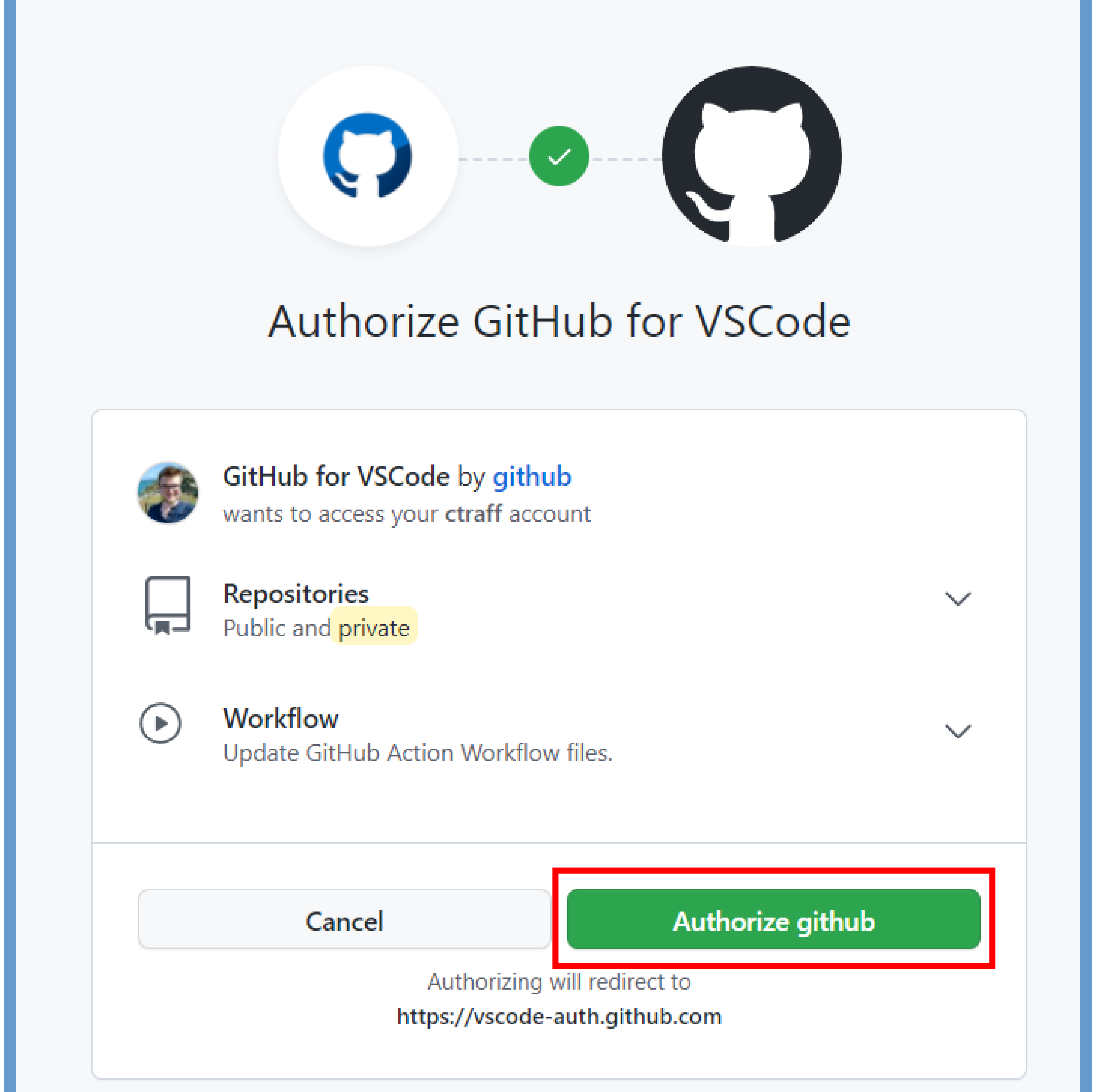


The extension 'GitHub' wants to sign in using GitHub.



Authorize Visual Studio Code to access GitHub

If you initiated this authorization from Visual Studio Code, click 'Continue' to authorize access to GitHub

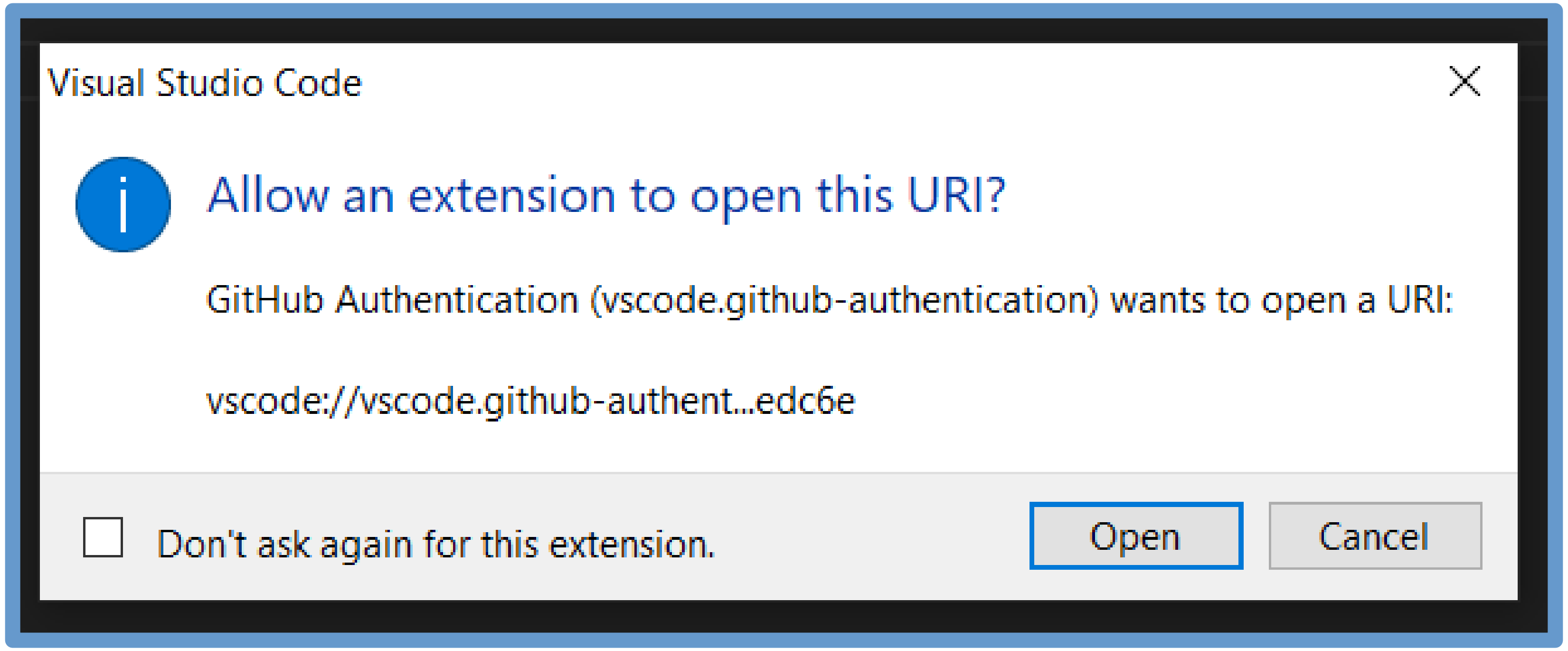

Authorize GitHub for VSCode

GitHub for VSCode by github
wants to access your **ctraff** account

Repositories
Public and private

Workflow
Update GitHub Action Workflow files.

Authorizing will redirect to
https://vscode-auth.github.com



Visual Studio Code

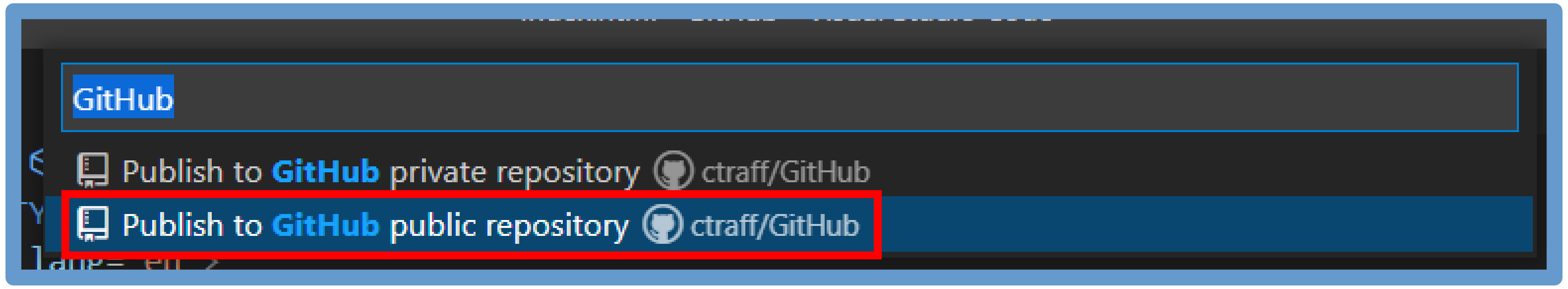
Allow an extension to open this URI?

GitHub Authentication (vscode.github-authentication) wants to open a URI:

vscode://vscode.github-authent...edc6e

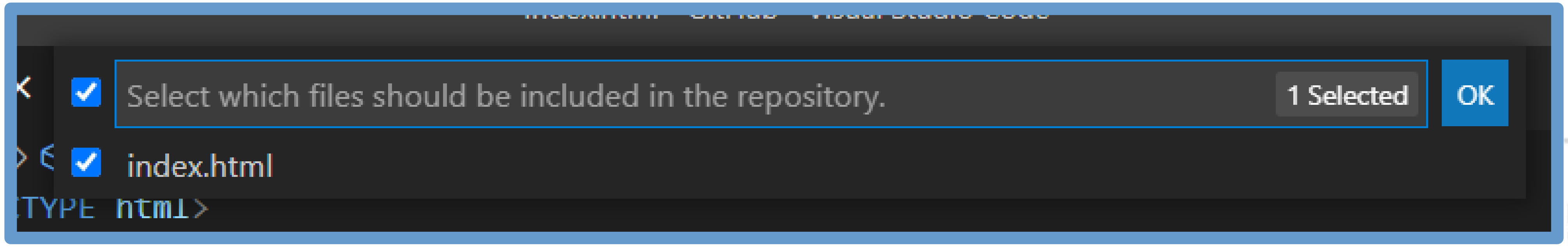
Creating a repository

- You will then have two options
  - Private: Only accessible to you and those you share access with
  - Public: Anyone in the world can see your repository (your code)
    - Good if you wanted to share your code with a colleague (group project year 2)
- We are going to set our repository to public
- You can see the name of the repository is USERNAME/FOLDER
  - In this example its my username / (the folder name which is GitHub)


GitHub
Publish to GitHub private repository ctraff/GitHub
Publish to GitHub public repository ctraff/GitHub

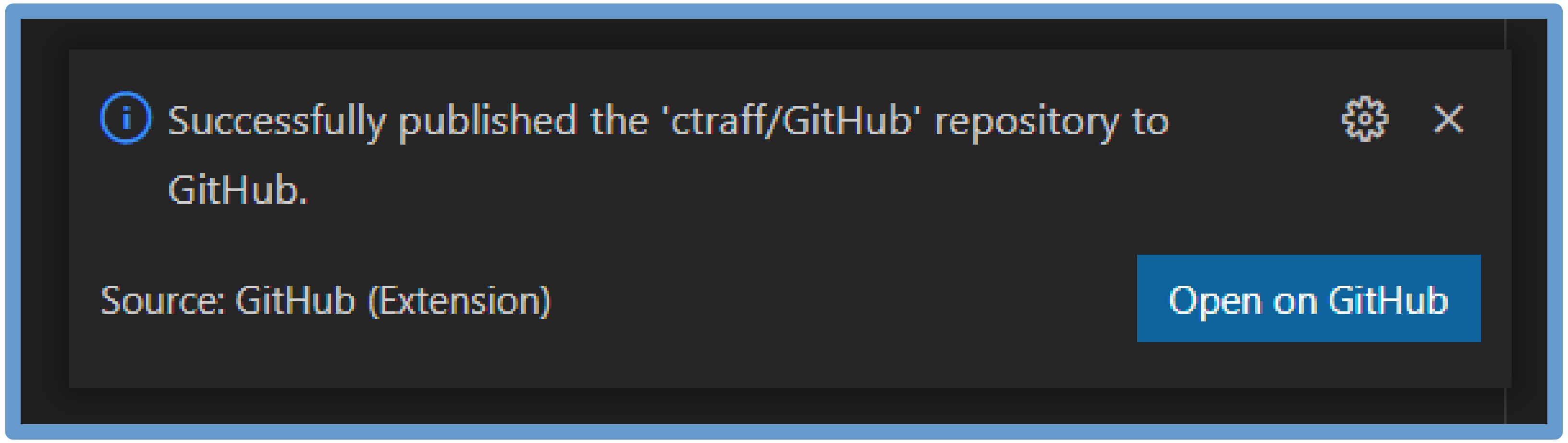
Creating a repository pt2

- You then need to select which files to upload
- In our case we only have the index.html file



Creating a repository pt3

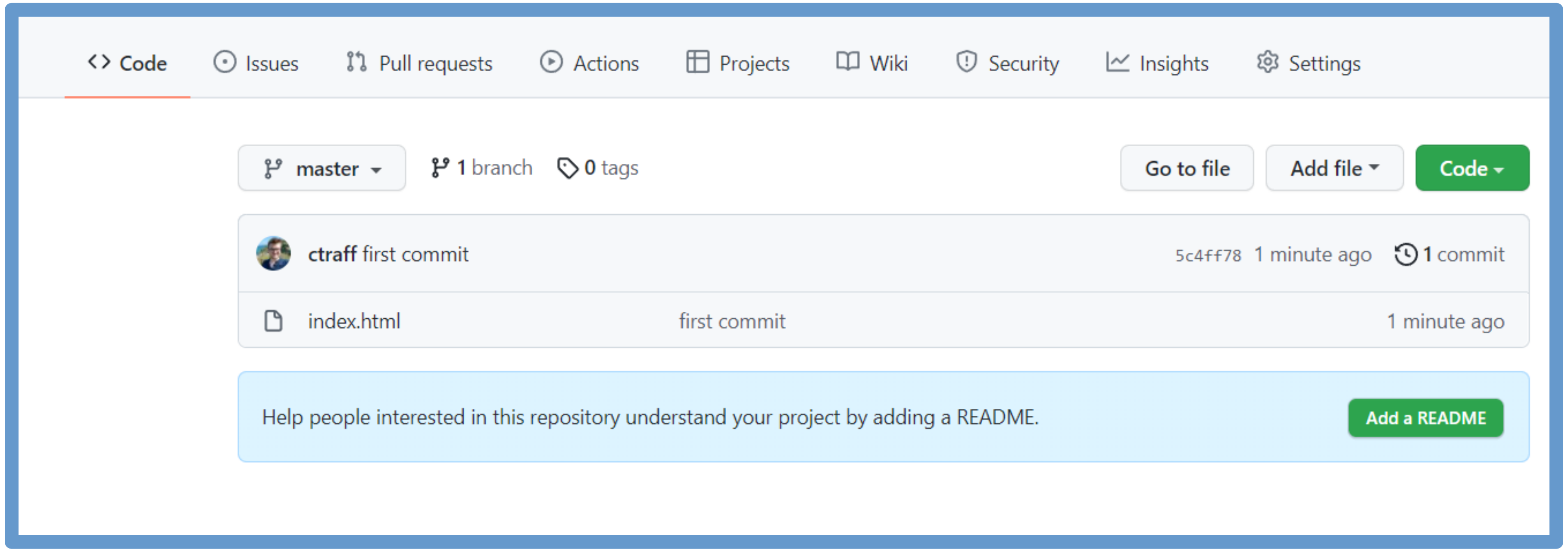
- We can then view the repository on GitHub


Successfully published the 'ctraff/GitHub' repository to GitHub.

Source: GitHub (Extension)

Creating a repository pt4

- You can see that the files we selected are now in our GitHub repository



ctraff first commit 5c4ff78 1 minute ago 1 commit

index.html first commit 1 minute ago

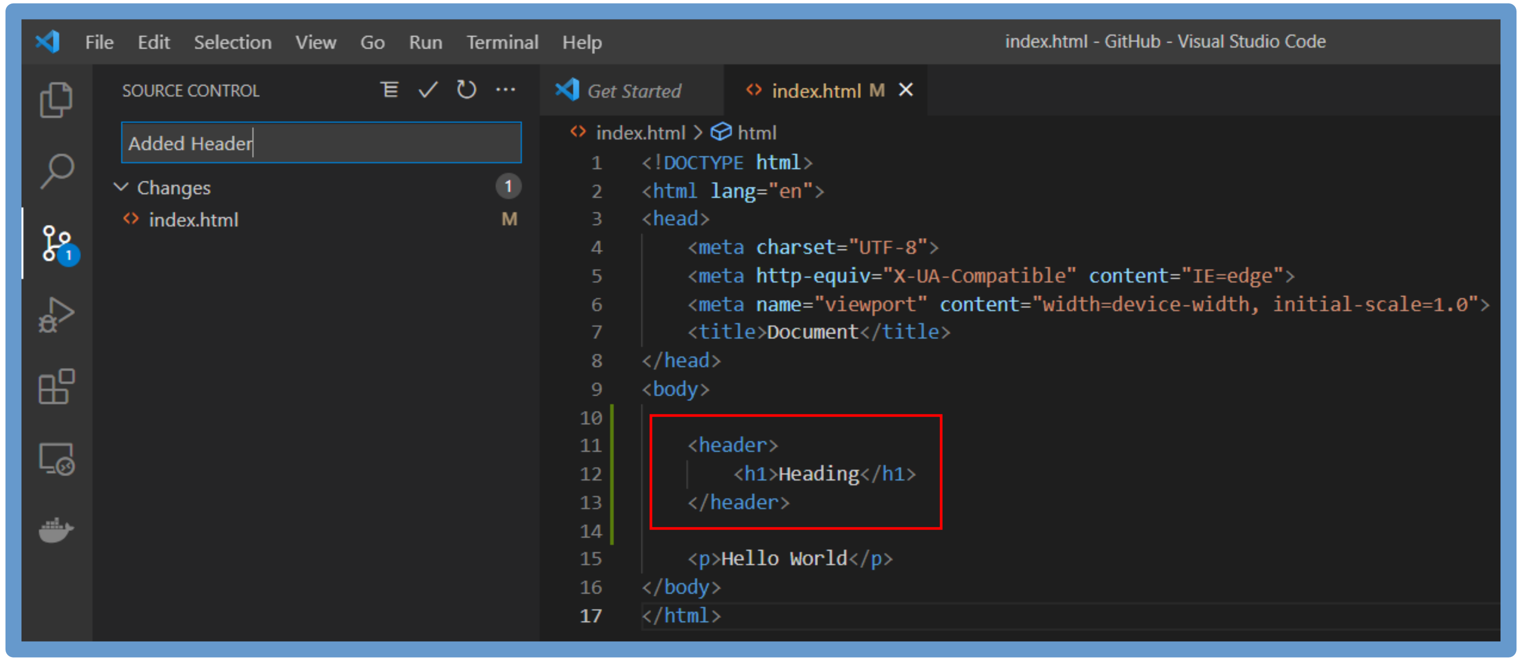
Help people interested in this repository understand your project by adding a README.

Creating a repository pt5

- This is useful because if we take the URL and give it to some one else they can view the code in that repository

Commit

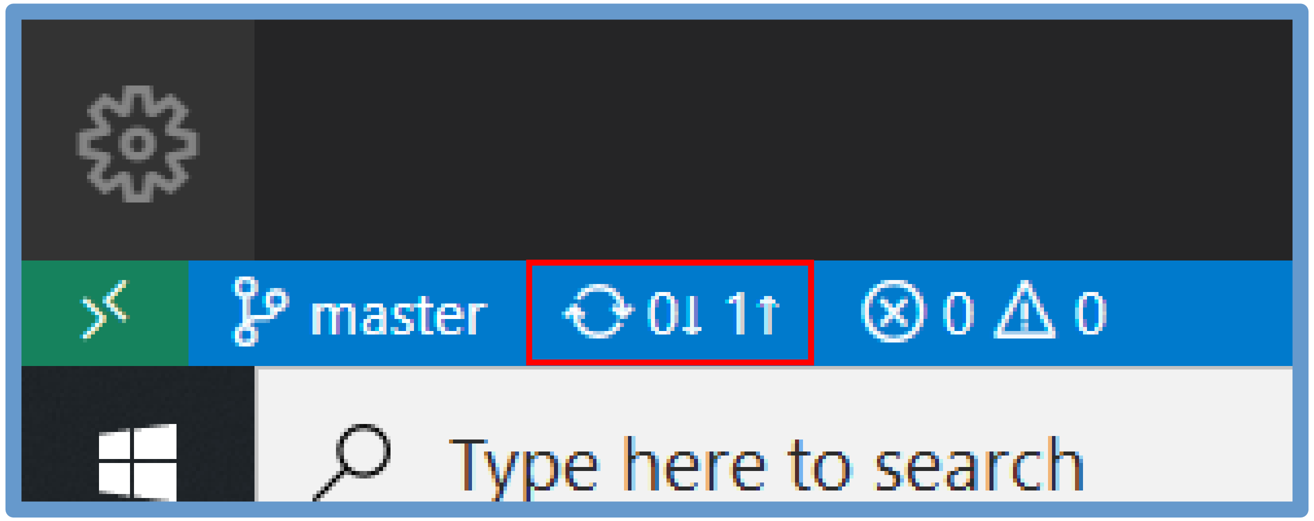
- If we now create a new piece of code and save it
- We can commit that change
  - In this example we are adding a header and calling the commit “Added Header”

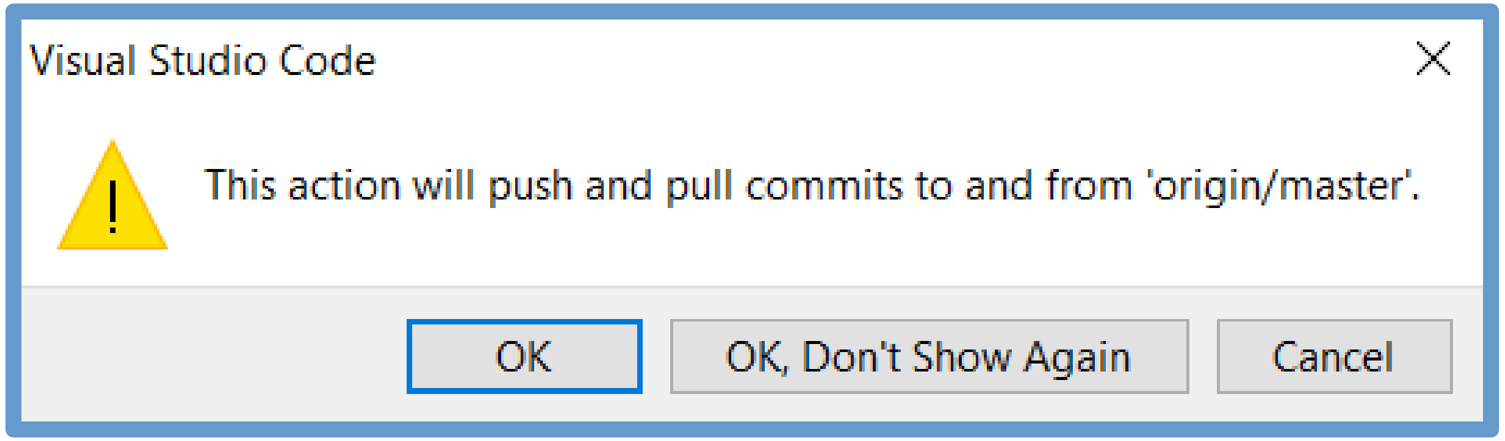


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

    <header>
        <h1>Heading</h1>
    </header>

    <p>Hello World</p>
</body>
</html>
```

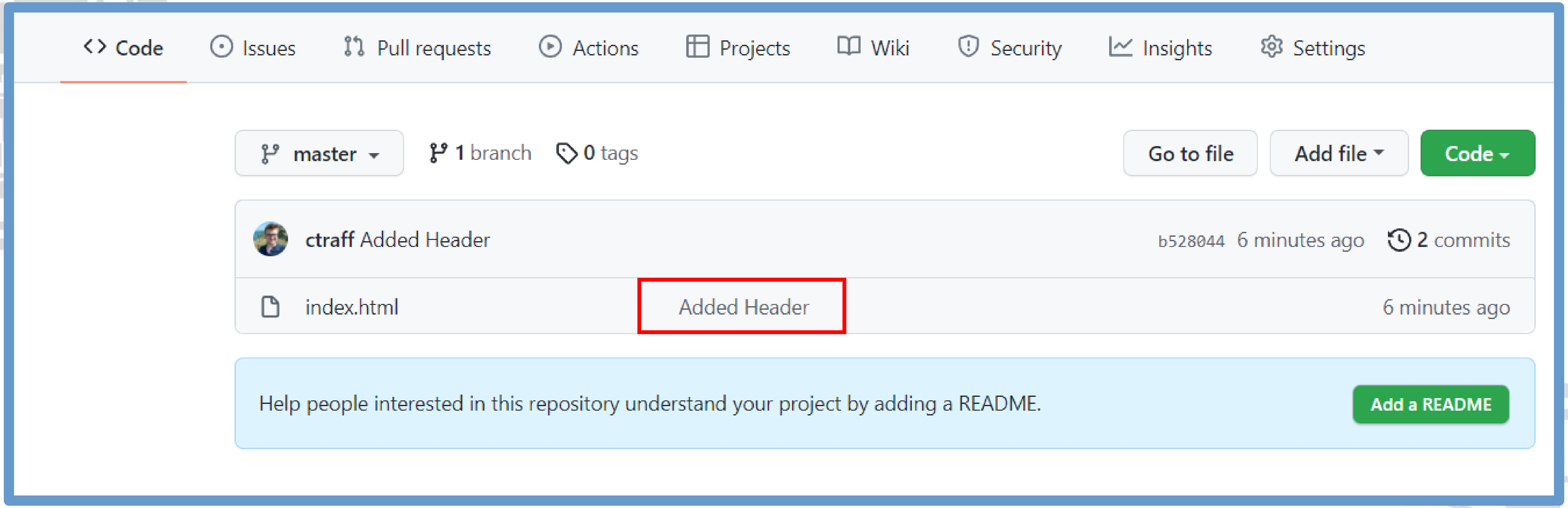
Commit pt2

- Before this commit can be seen on GitHub we need to do what is called Push
- Git has saved that commit locally on your machine to save it to GitHub we need to Push the commit
  - To do this we need to press this button at the bottom




Visual Studio Code

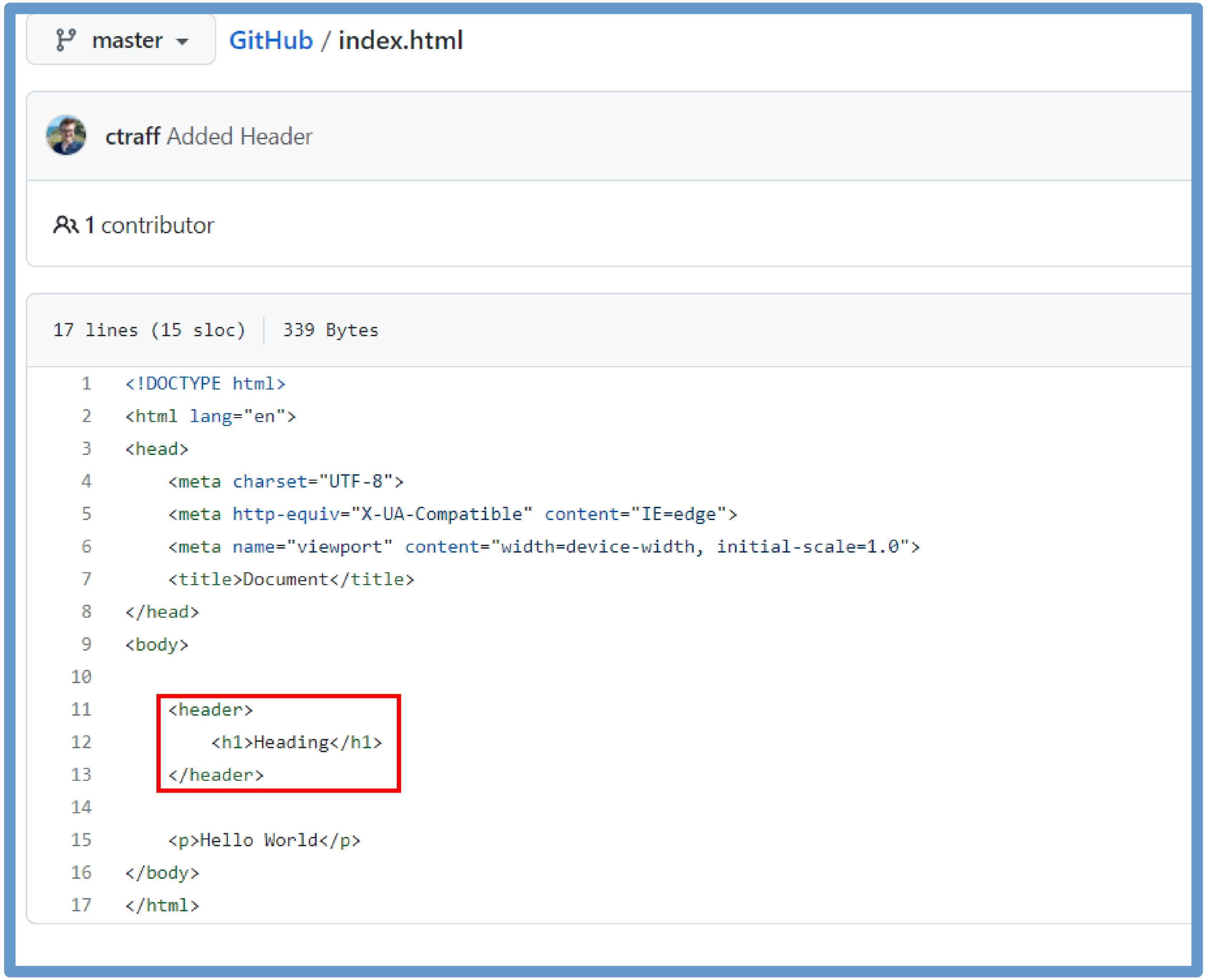
This action will push and pull commits to and from 'origin/master'.



ctraff Added Header b528044 6 minutes ago 2 commits

index.html Added Header 6 minutes ago

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


master
GitHub / index.html

ctraff Added Header

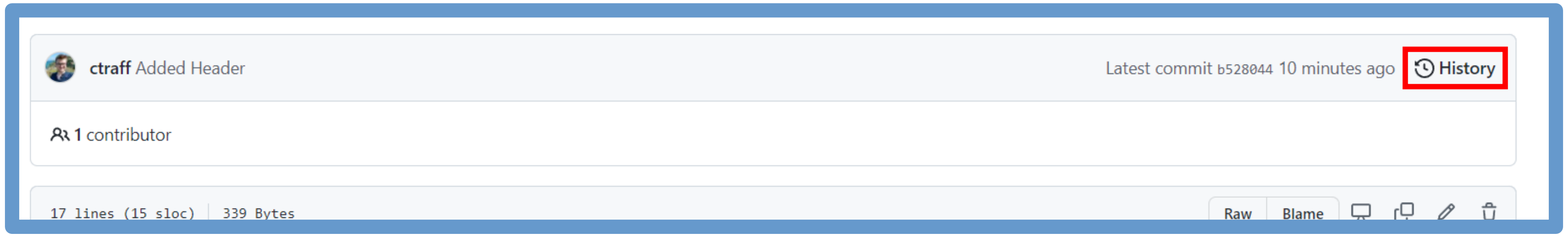
1 contributor

17 lines (15 sloc) | 339 Bytes

```html
<!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>

    <header>
        <h1>Heading</h1>
    </header>

    <p>Hello World</p>
</body>
</html>
```

History

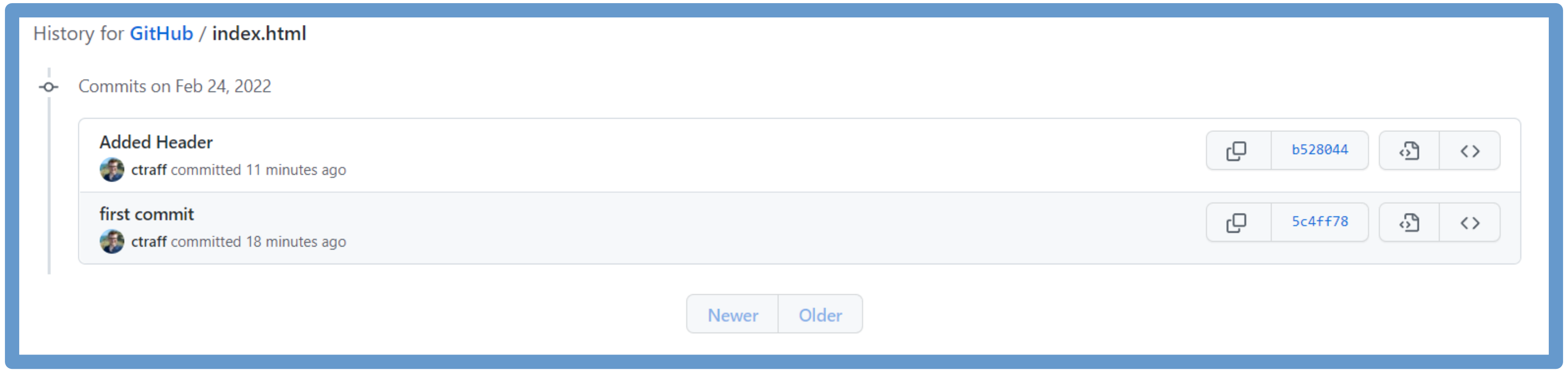
- If you click on the History button
- You can see all the commits that have been made to that file
  - It’s a lot nicer to use than the timeline in Visual Studio Code


ctraff Added Header

Latest commit b528044 10 minutes ago

1 contributor

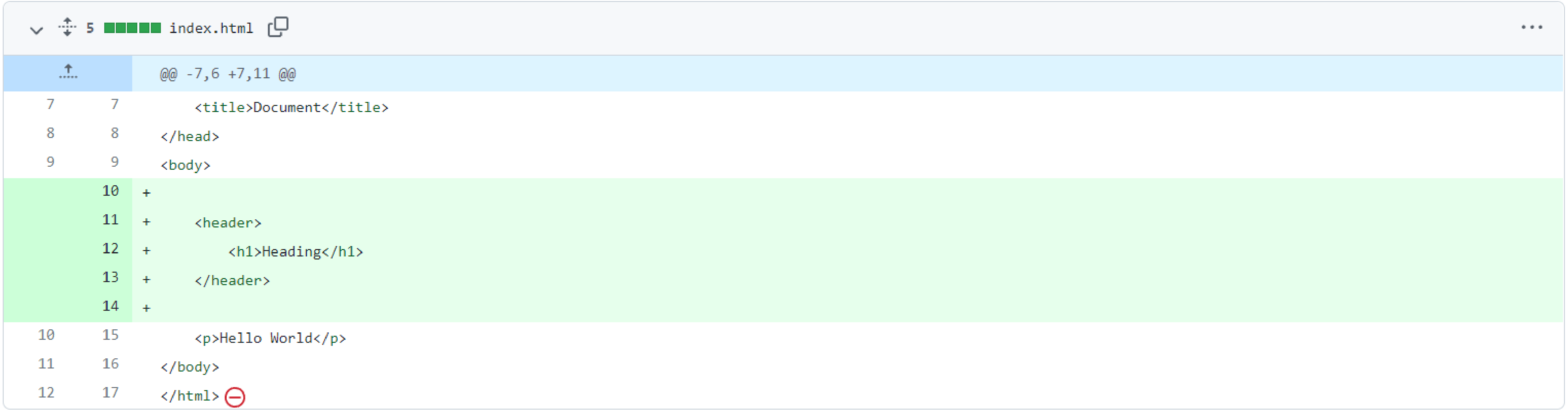
17 lines (15 sloc) 339 Bytes


History for GitHub / index.html

Commits on Feb 24, 2022

Added Header
ctraff committed 11 minutes ago
b528044

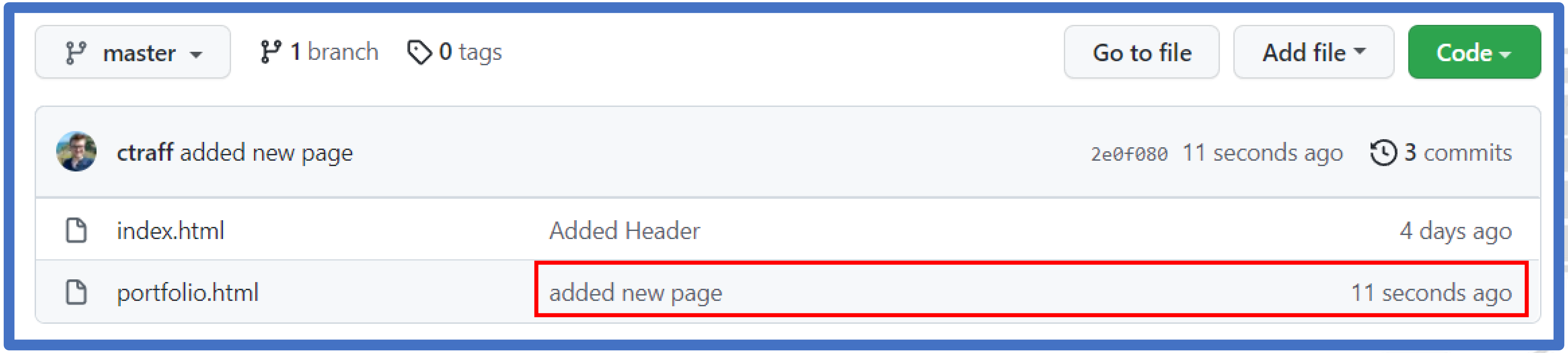
first commit
ctraff committed 18 minutes ago
5c4ff78



```
@@ -7,6 +7,11 @@
         <title>Document</title>
     </head>
     <body>
+
+        <header>
+            <h1>Heading</h1>
+        </header>
+
         <p>Hello World</p>
     </body>
 </html>
```

History pt2

- You can see the history of every commit in all the files in your project
  - This is useful because you can see the exact point when a piece of code was added
  - For example, if we added a new page to our website you can see the exact commit along with the time it was added

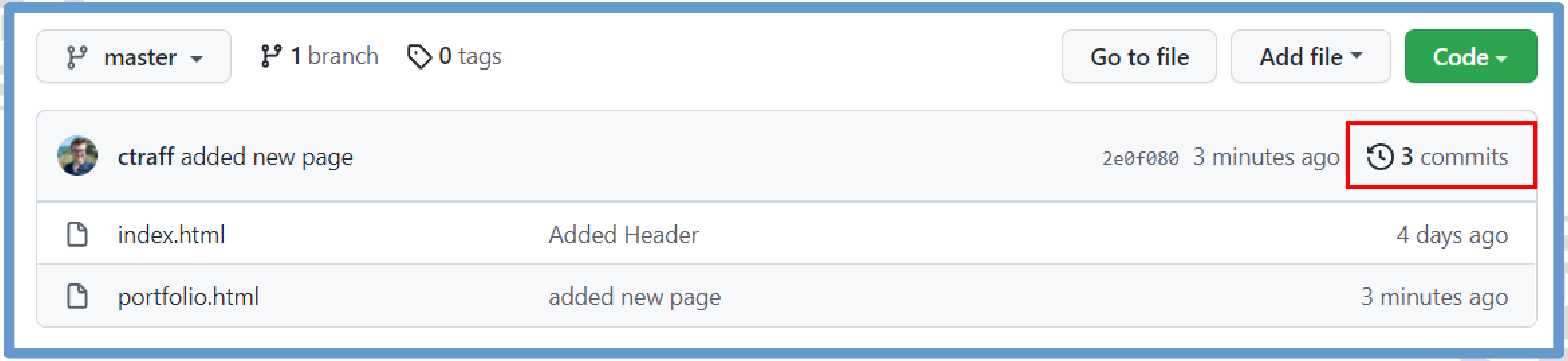


ctraff added new page 2e0f080 11 seconds ago 3 commits

index.html	Added Header	4 days ago
portfolio.html	added new page	11 seconds ago

History pt3

- You can see all the commits in the project

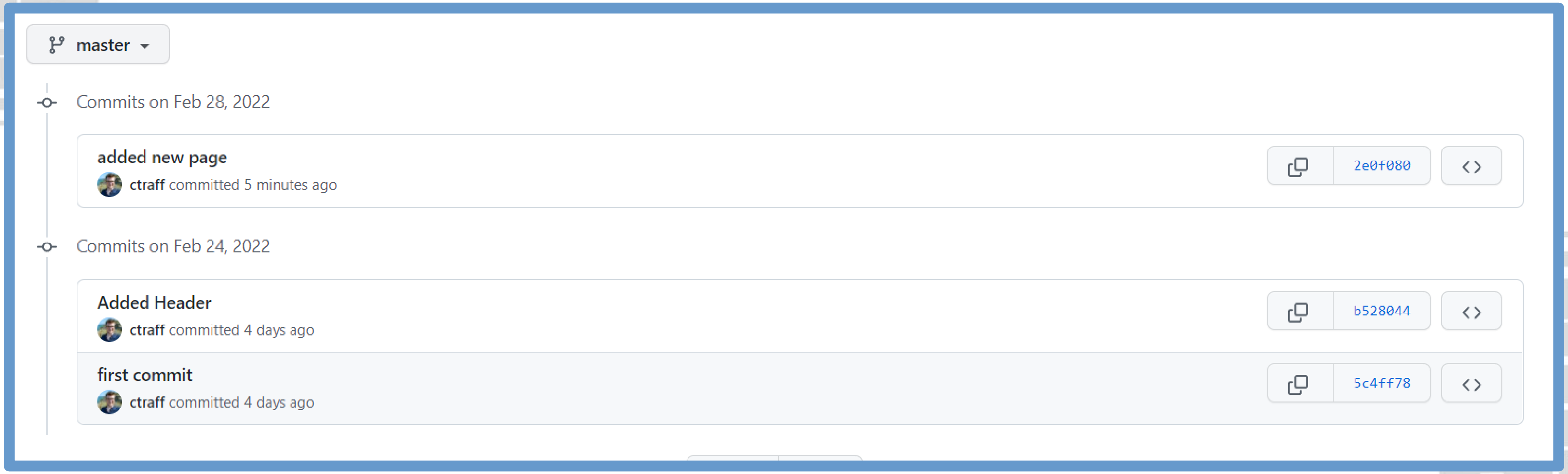


ctraff added new page 2e0f080 3 minutes ago 3 commits

index.html	Added Header	4 days ago
portfolio.html	added new page	3 minutes ago

History pt4

- You can view all the snapshots of the code at that point in time



master

Commits on Feb 28, 2022

added new page
ctraff committed 5 minutes ago
2e0f080

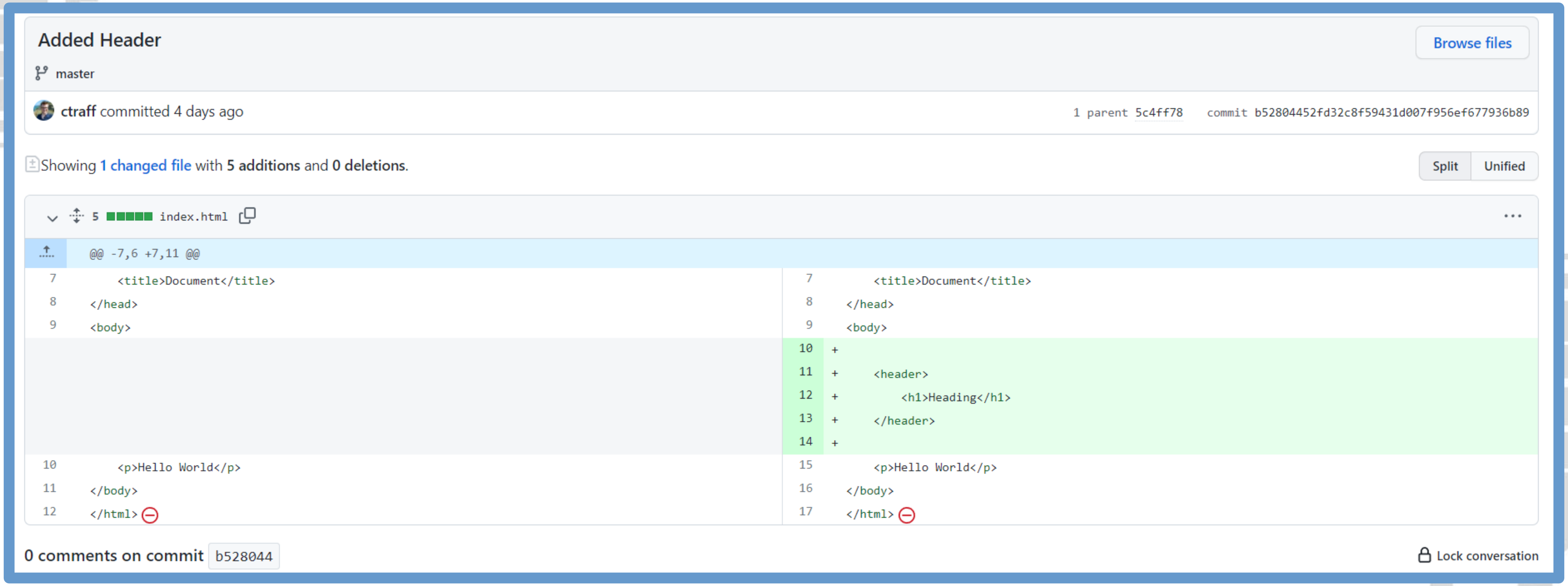
Commits on Feb 24, 2022

Added Header
ctraff committed 4 days ago
b528044

first commit
ctraff committed 4 days ago
5c4ff78

History pt5

- We can view what was changed in our code



Added Header

master

ctraff committed 4 days ago

1 parent 5c4ff78 commit b52804452fd32c8f59431d007f956ef677936b89

Showing 1 changed file with **5 additions** and **0 deletions**.

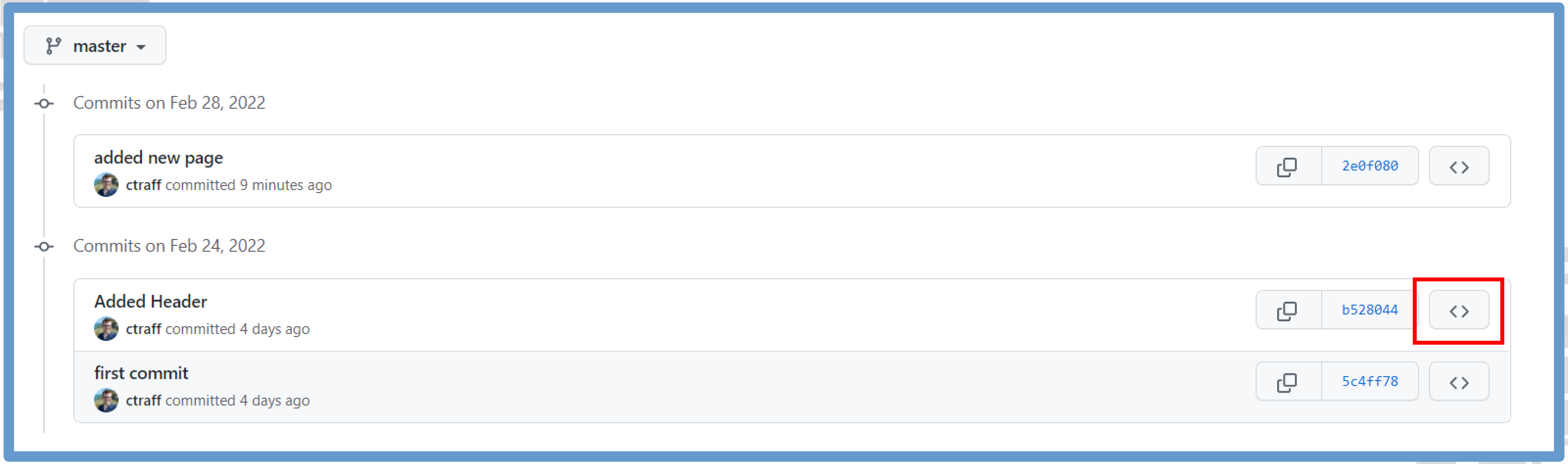
5 index.html

```
@@ -7,6 +7,11 @@
 7      <title>Document</title>        7      <title>Document</title>
 8  </head>                            8  </head>
 9  <body>                             9  <body>
                                      10 +
                                      11 +    <header>
                                      12 +        <h1>Heading</h1>
                                      13 +    </header>
                                      14 +
10      <p>Hello World</p>            15      <p>Hello World</p>
11  </body>                           16  </body>
12  </html>                           17  </html>
```

0 comments on commit b528044

History pt6

- You can browse the repository at that point in the history



master

Commits on Feb 28, 2022

added new page
ctraff committed 9 minutes ago
2e0f080

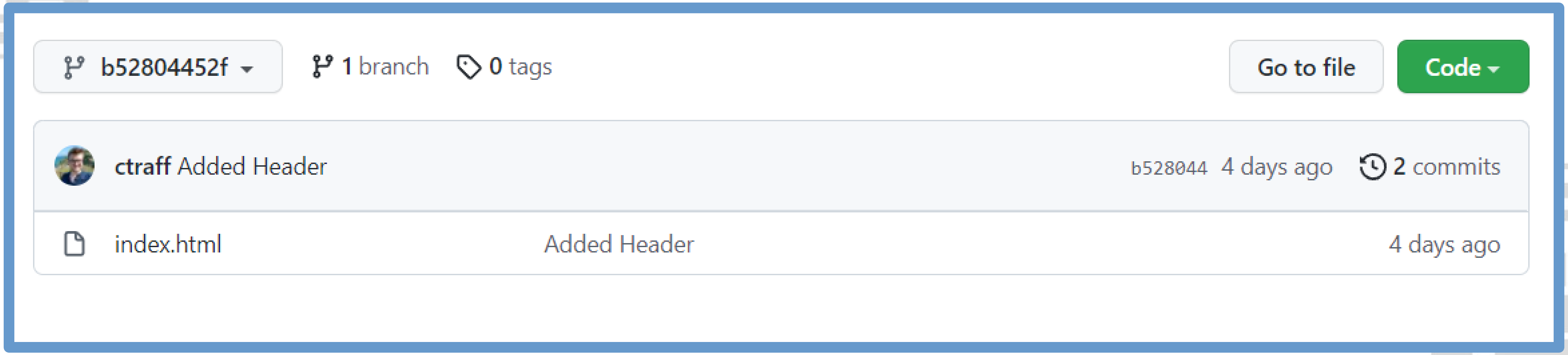
Commits on Feb 24, 2022

Added Header
ctraff committed 4 days ago
b528044

first commit
ctraff committed 4 days ago
5c4ff78

History pt7

- The portfolio.html page didn't exist in this commit
- We can view how the repository looked at this point in time

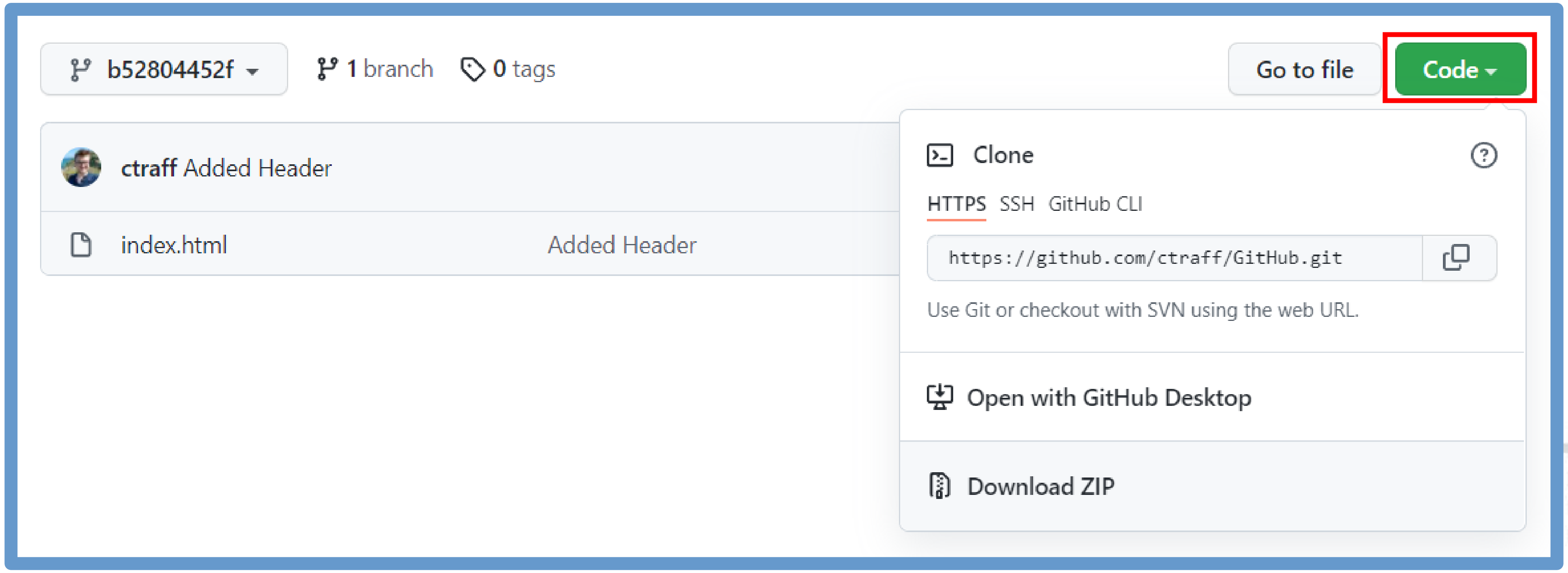


ctraff Added Header b528044 4 days ago 2 commits

index.html	Added Header	4 days ago

History pt8

- We can even download the repository or clone it to a folder on our computer



```
https://github.com/ctraff/GitHub.git
```

Use Git or checkout with SVN using the web URL.

History pt9

- This acts as a backup of our website taking snapshots throughout development
  - If we ever had an issue with the website we can easily see the commit it was added and revert to a previous snapshot
  - This is most popular form of version control used in industry today

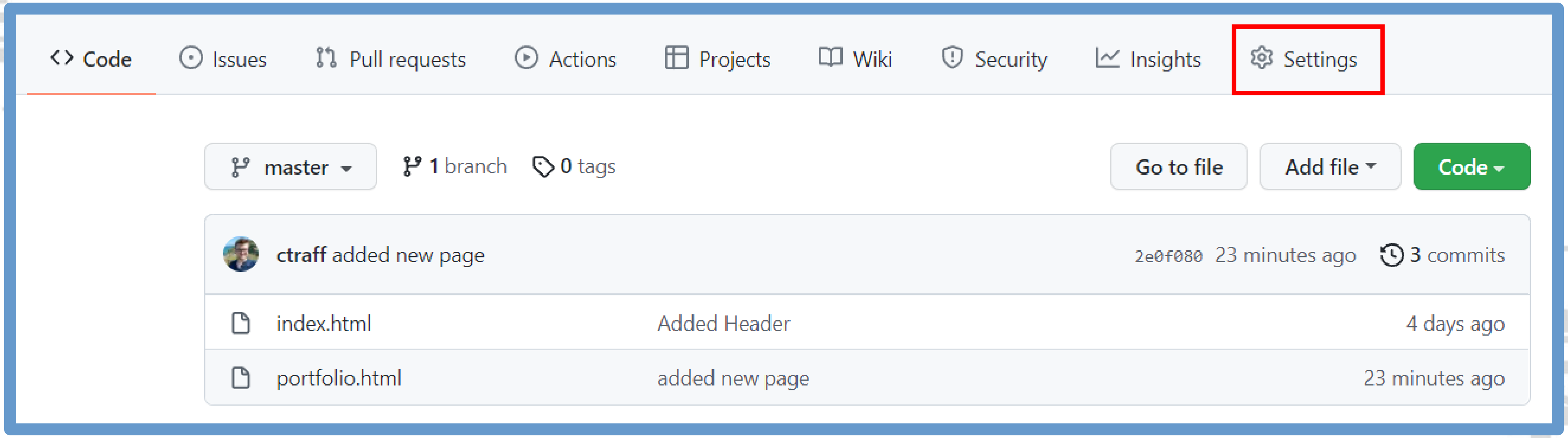
- It might look a bit complicated at first but after using it a few times you can never go back

GitHub Pages

- If you want to make your pages live on the web (viewable using a URL) you can do that through GitHub
- When you normally create a page its only viewable to you on your computer
  - You can only access it by having physical access to your computer
- One of the really useful features of GitHub is making your pages live on the web so you can test them

GitHub Pages pt2

- To do this you will need to got to settings

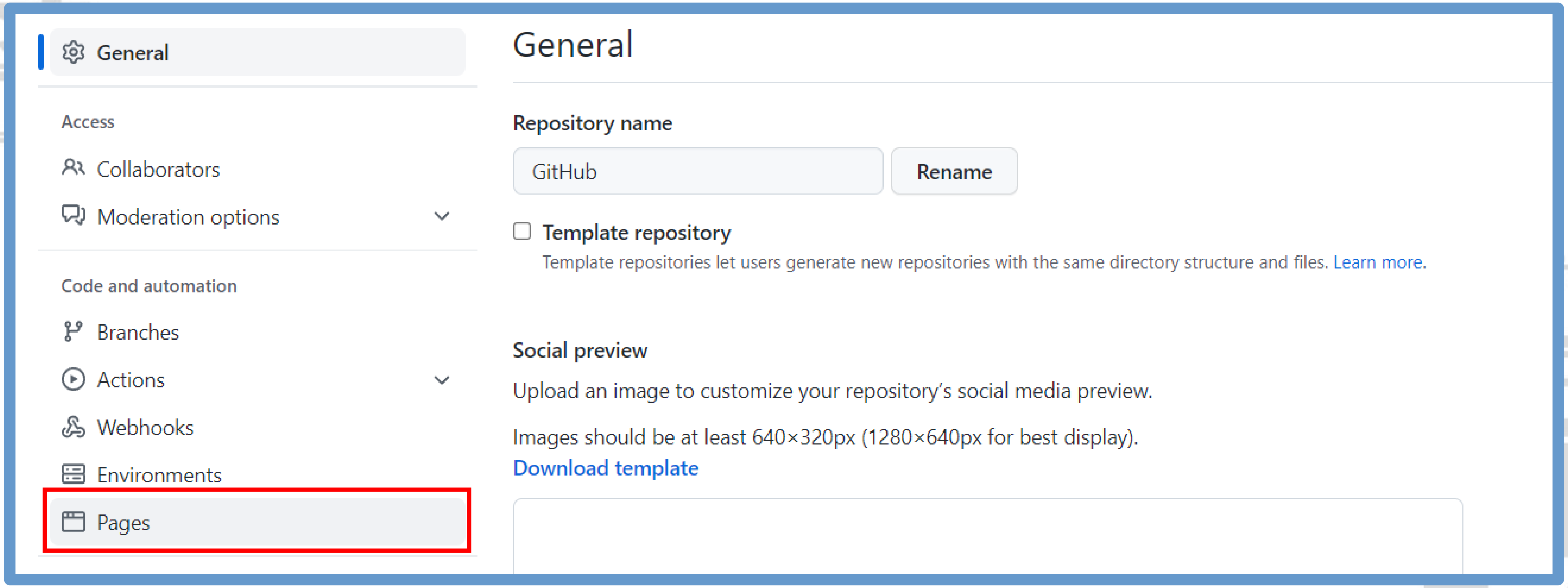


ctraff added new page 2e0f080 23 minutes ago 3 commits

index.html	Added Header	4 days ago
portfolio.html	added new page	23 minutes ago

GitHub Pages pt3

- Using the side bar navigate to Pages



General

Repository name

GitHub

Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more.

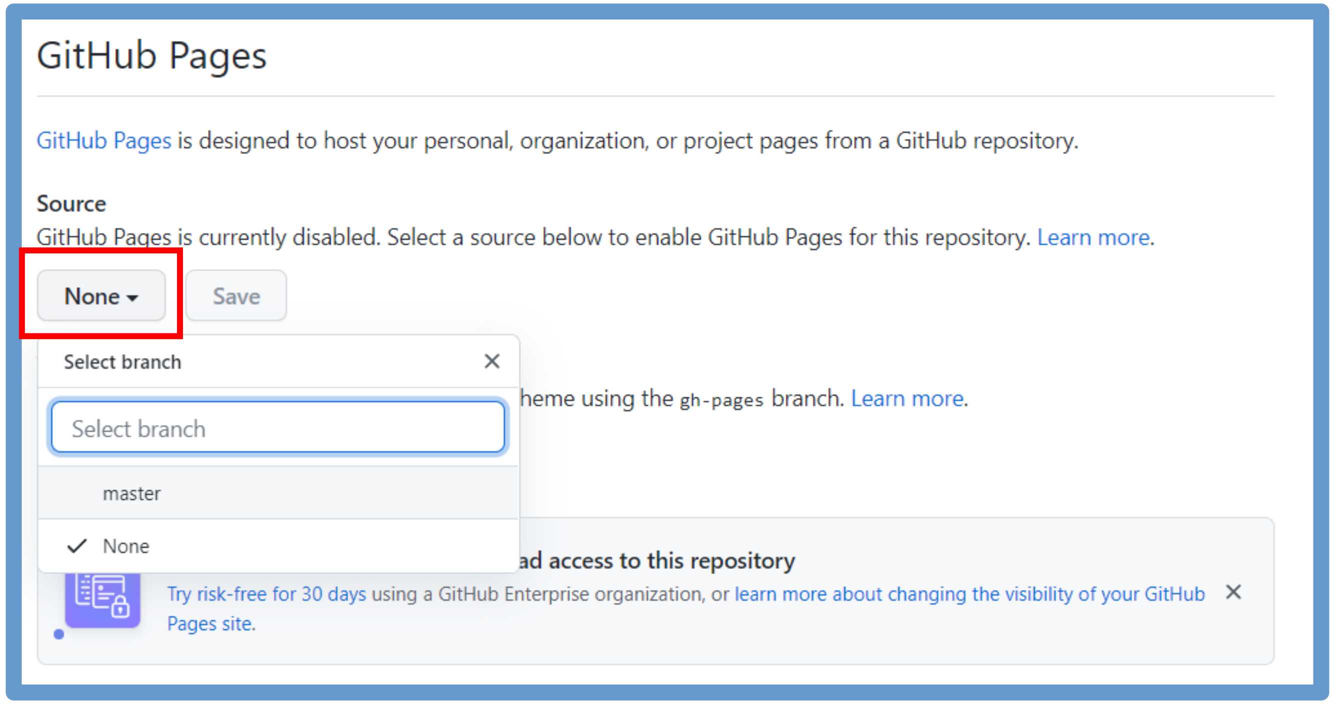
Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).
Download template

GitHub Pages pt4

- You then need to select the source button (None)
  - At this point you will only have the master branch however when you start using multiple branches you can select which branch you want to publish



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

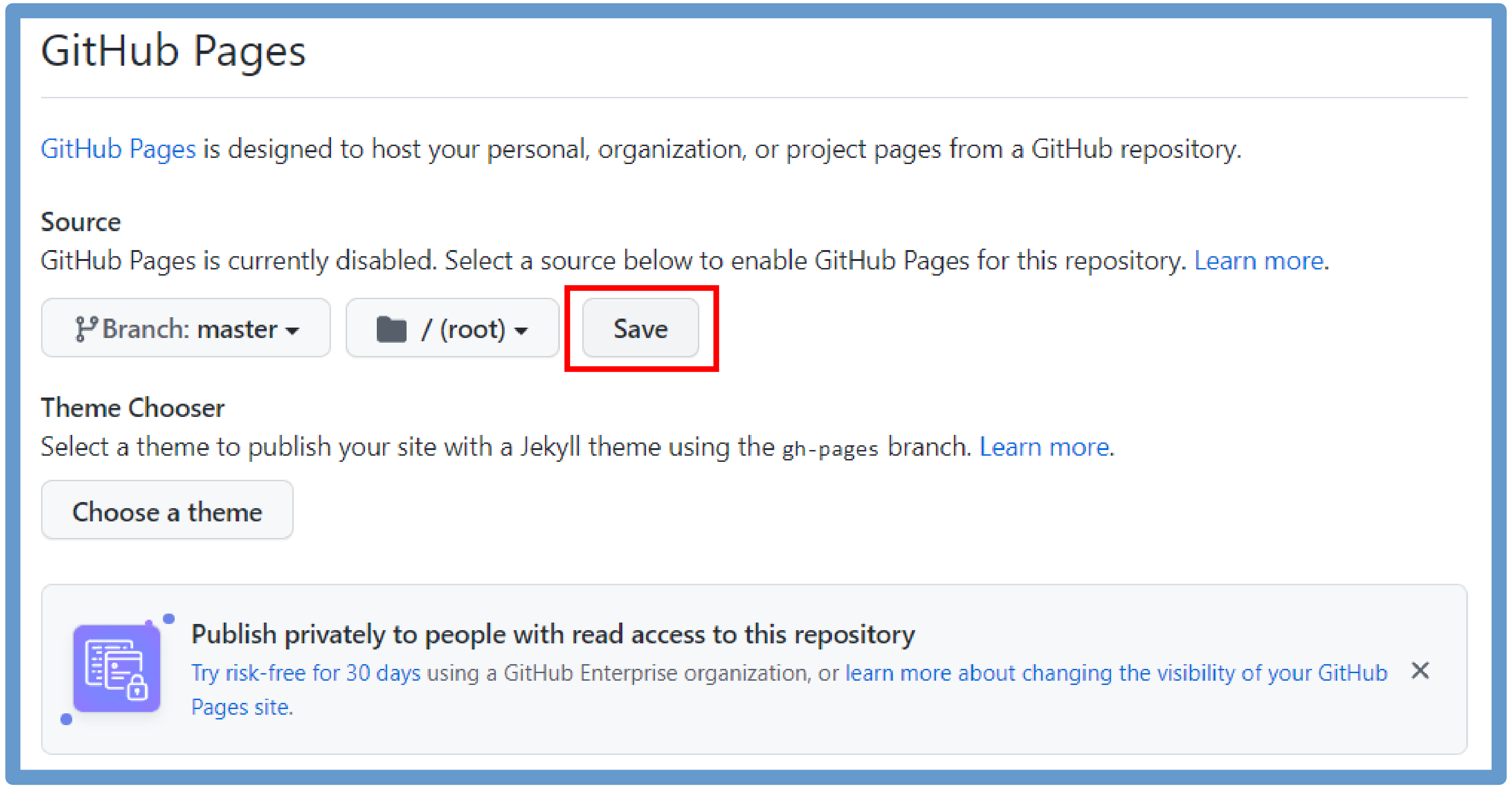
heme using the `gh-pages` branch. Learn more.

ad access to this repository

Try risk-free for 30 days using a GitHub Enterprise organization, or learn more about changing the visibility of your GitHub Pages site.

GitHub Pages pt5

- Then all we need to do is click save



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

Theme Chooser

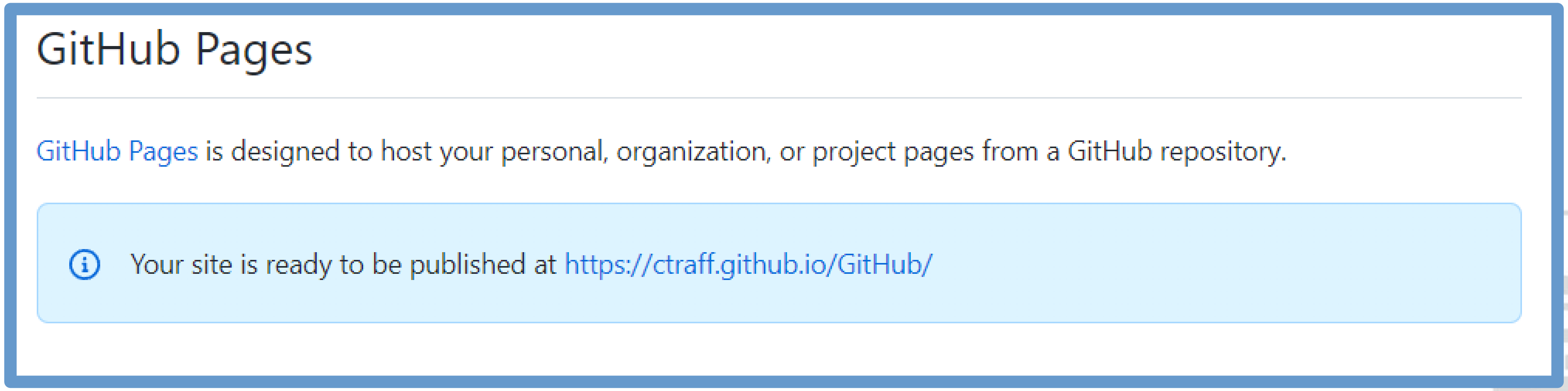
Select a theme to publish your site with a Jekyll theme using the `gh-pages` branch. Learn more.

Publish privately to people with read access to this repository

Try risk-free for 30 days using a GitHub Enterprise organization, or learn more about changing the visibility of your GitHub Pages site.

GitHub Pages pt6

- The page will refresh and you will see this message saying your site is ready to be published



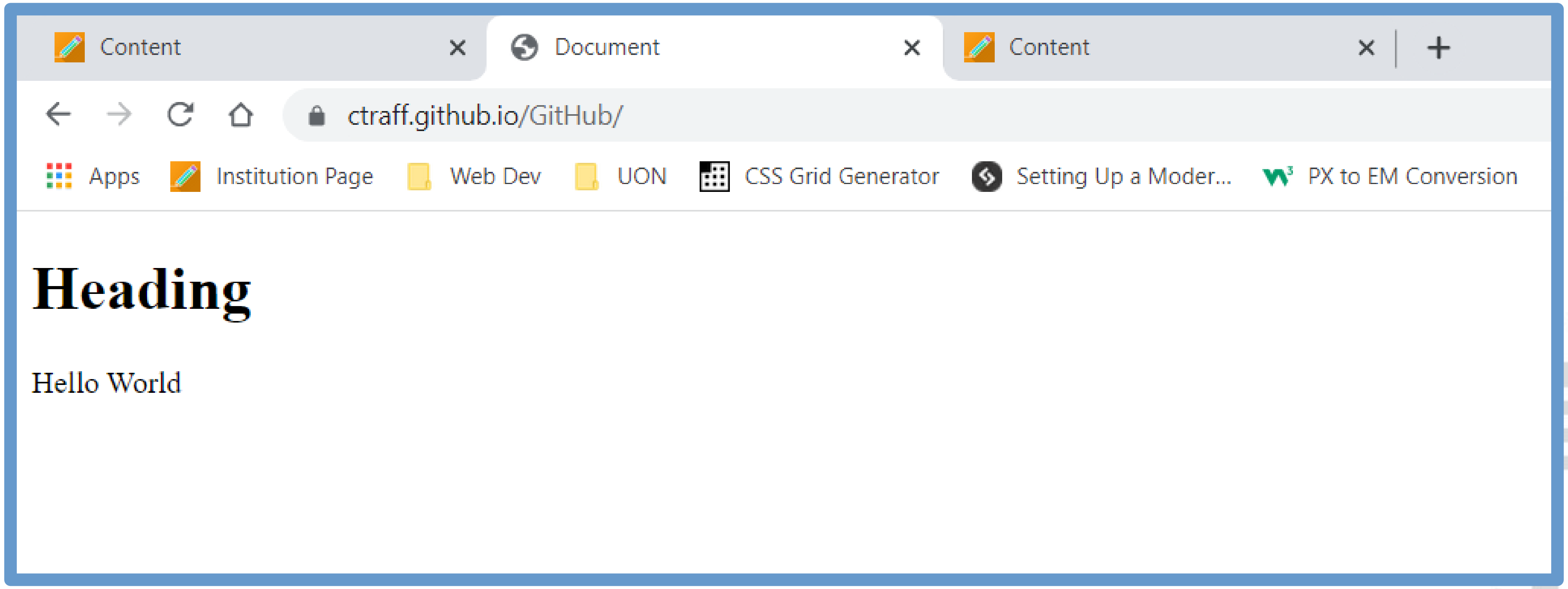
GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://ctraff.github.io/GitHub/

GitHub Pages pt7

- Click the link and you will then be taken to the live version of your site
  - You can send the URL to anyone and they will now be able to view your website



Heading

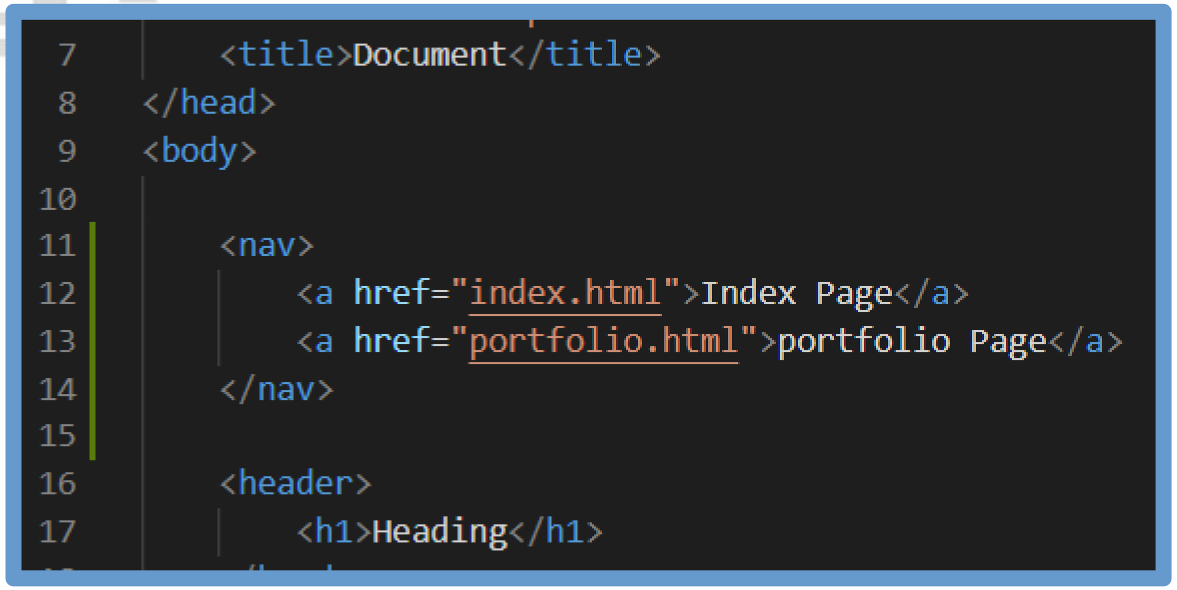
Hello World

GitHub Pages pt8

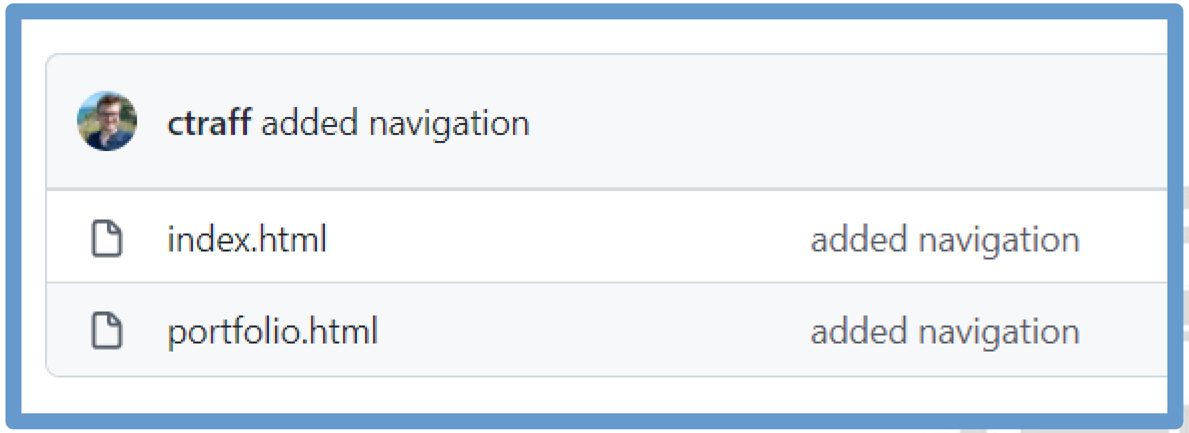
- The first page you are taken too is index.html
  - This is why it is important to have your homepage called index
  - It’s the first page the website will look for
- If your website is not displaying correctly make sure do the following
  - Wait a few minutes (sometimes it can take up to 20)
  - Make sure your index file is in your repository
  - See if you can view it locally on your computer
- The URL is broken down into
  - https://USERNAME.github.io/FOLDER_NAME/FILE_NAME.html

GitHub Pages pt9

- If we add some anchor tags (<a>) to act as the navigation to our pages
  - Remember to commit the changes and push to GitHub



```
    <title>Document</title>
</head>
<body>

    <nav>
        <a href="index.html">Index Page</a>
        <a href="portfolio.html">portfolio Page</a>
    </nav>

    <header>
        <h1>Heading</h1>
```

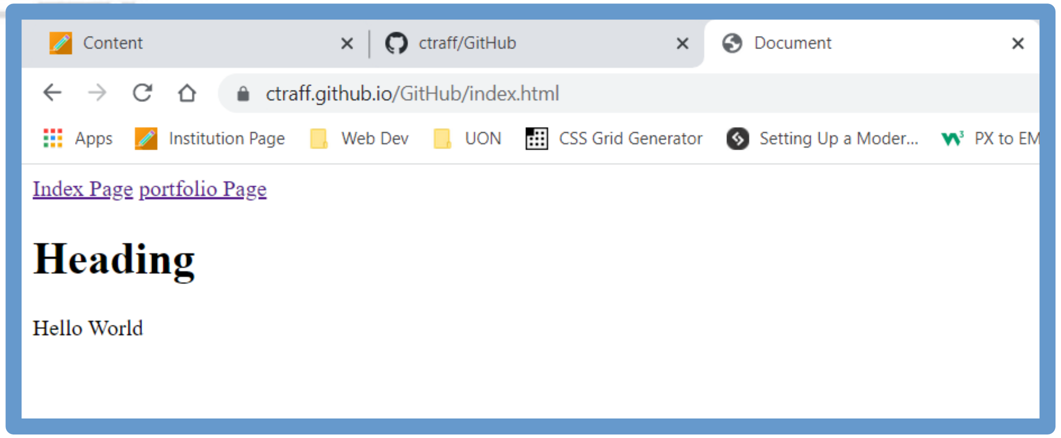


ctraff added navigation

index.html	added navigation
portfolio.html	added navigation

GitHub Pages pt10

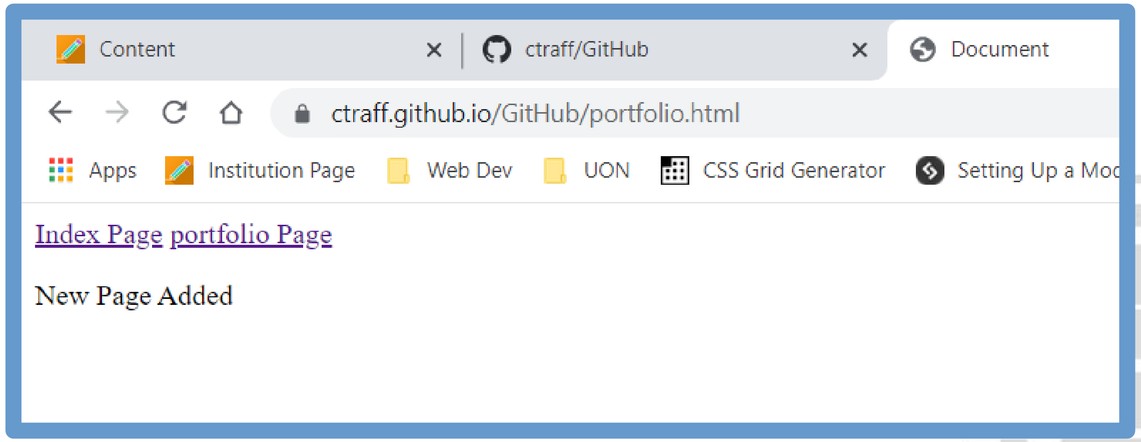
- If you refresh the page on your live site you will see that the code has been update
  - This can take a few minutes to sync up (up to 20 minutes)



Index Page portfolio Page

Heading

Hello World



Index Page portfolio Page

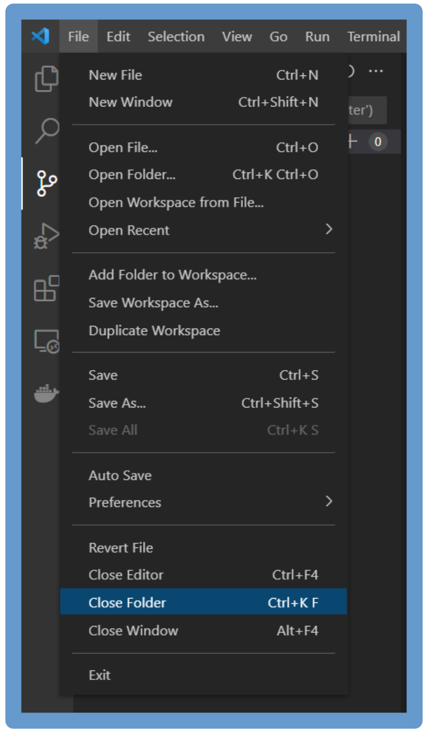
New Page Added

GitHub Pages pt11

- You can have multiple website live
- There is a size limit of 1GB per website
  - You shouldn’t run into this issue but if you do it will most likely be image size based (try optimizing the images to be a smaller size https://imagecompressor.com/)
- Only HTML, CSS & JavaScript work with GitHub Pages (PhP that you learn in the second year will not)

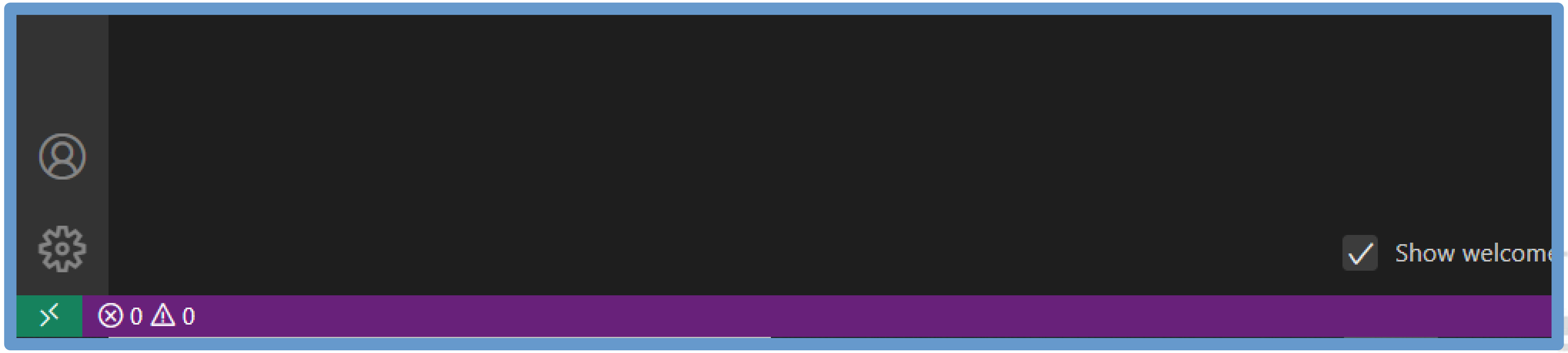
Forking

- If you want to use a repository you have found online and edit the code (on GitHub)
- First thing you need to do is go to File -> Close Folder



Forking pt2

- The bar at the bottom should now be purple instead of blue

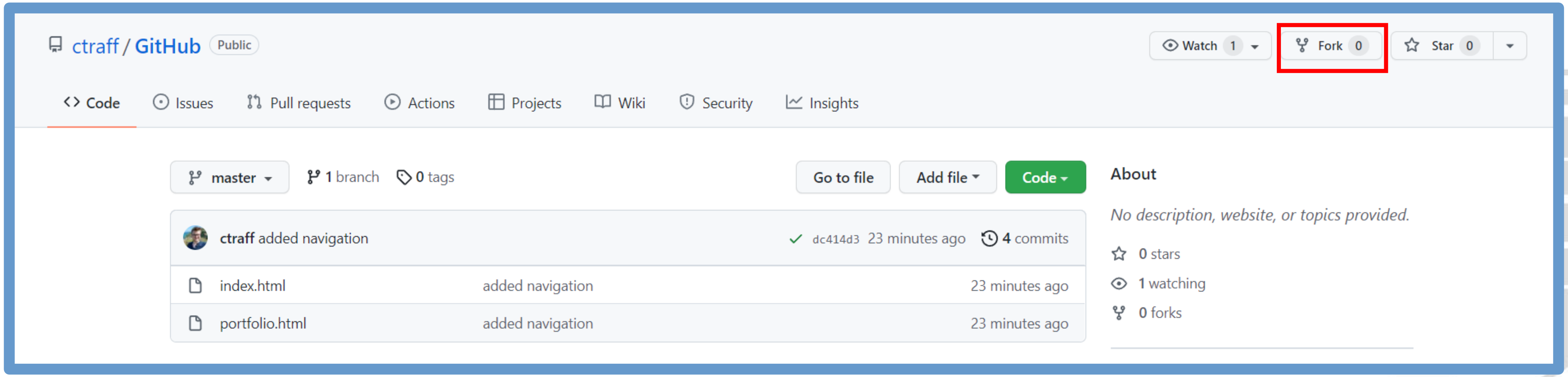


Forking pt3

- I have signed into a different GitHub account
- This is to demonstrate how you can access and edit someone else's repository
  - Using a different account we are going to change the repository we just created

Forking pt4

- Using the alternative account we can view the repository (needs to be public)
- We need to get access to these files and to do that we click on the Fork button.



ctraff / GitHub Public

Watch 1 | Fork 0 | Star 0

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 0 tags | Go to file | Add file | Code

ctraff added navigation ✓ dc414d3 23 minutes ago 4 commits

index.html	added navigation	23 minutes ago
portfolio.html	added navigation	23 minutes ago

About

No description, website, or topics provided.

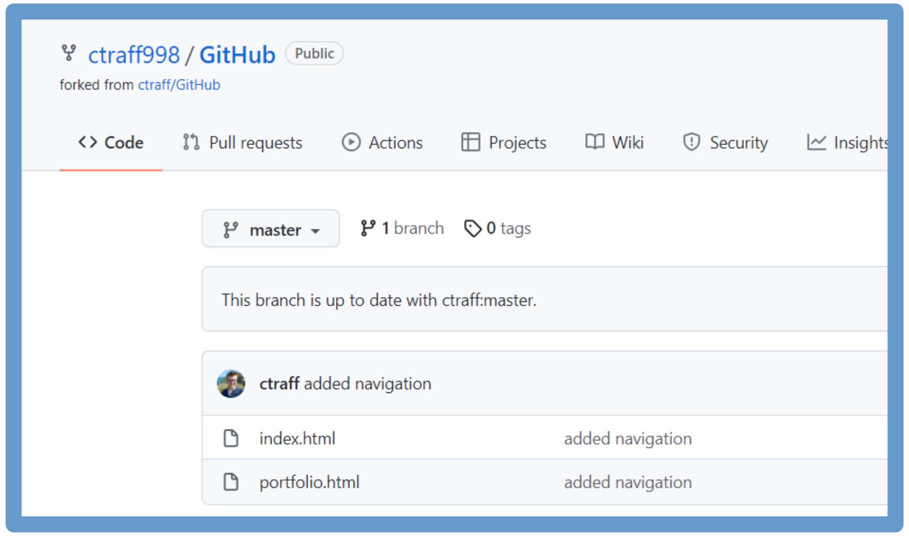
0 stars

1 watching

0 forks

Forking pt5

- This will copy all the code from that repository to your account
  - There are now two repositories for this code
    - The original account (the one we created the repository with)
    - The new account we have just used to copy the code



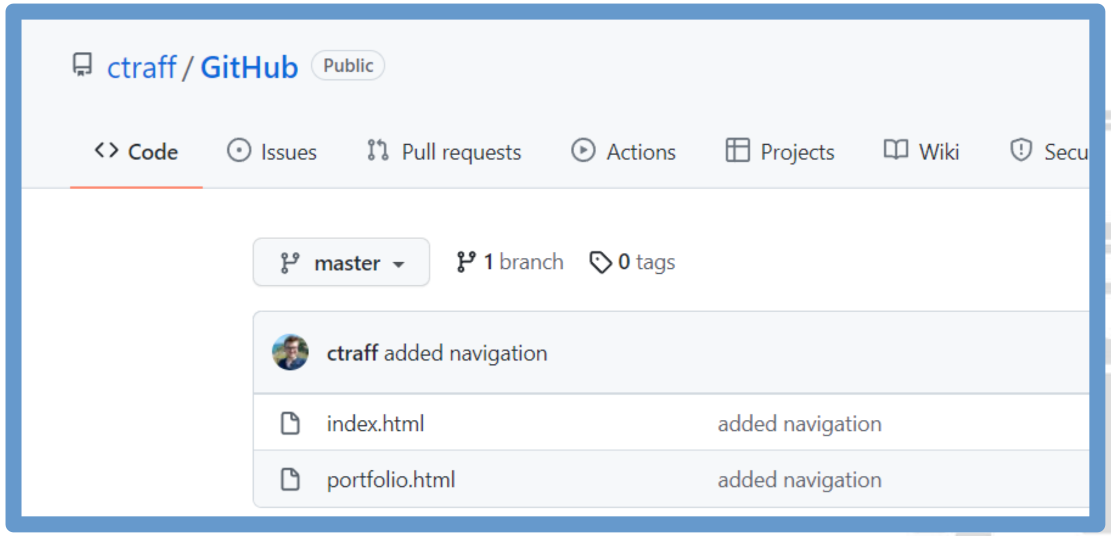
ctraff998 / GitHub Public
forked from ctraff/GitHub

This branch is up to date with ctraff:master.

ctraff added navigation

index.html added navigation
portfolio.html added navigation

New Account



ctraff / GitHub Public

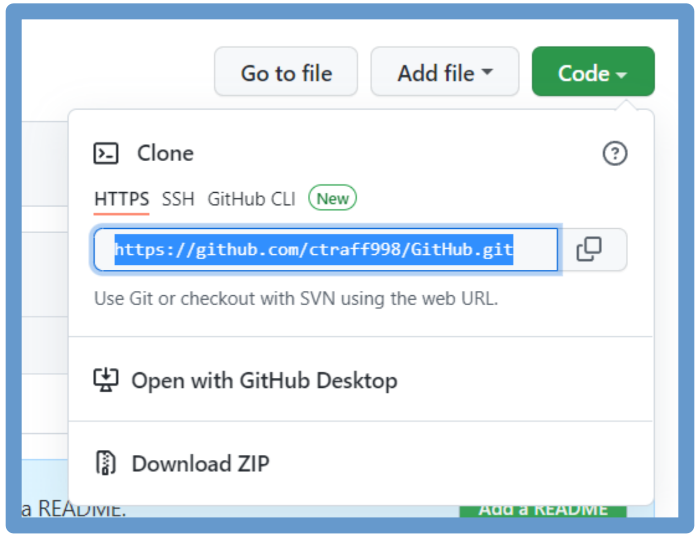
ctraff added navigation

index.html added navigation
portfolio.html added navigation

Old Account

Forking pt6

- A fork is a separate version of the code in a different GitHub account
- You can have as many forks as you like in many different accounts
  - In big scale projects each person can have a different fork of the code
- To download the fork onto our account and use it in Visual Studio Code we click on the code button and copy the link

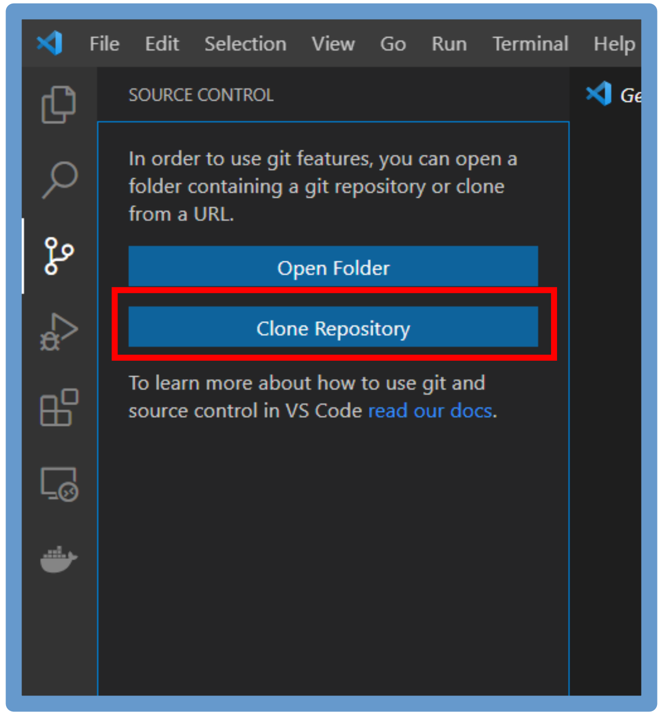


https://github.com/ctraff998/GitHub.git

Use Git or checkout with SVN using the web URL.

Forking pt7

- Go back to Visual Studio Code and click the Source Control tab
- There are two options the one we want is Clone Repository



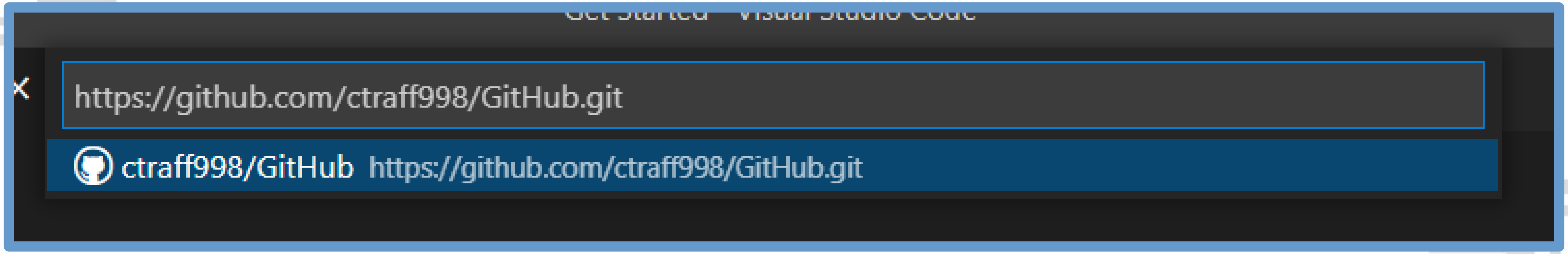
SOURCE CONTROL

In order to use git features, you can open a folder containing a git repository or clone from a URL.

To learn more about how to use git and source control in VS Code read our docs.

Forking pt8

- You will then need to paste in the link you copied from the repository

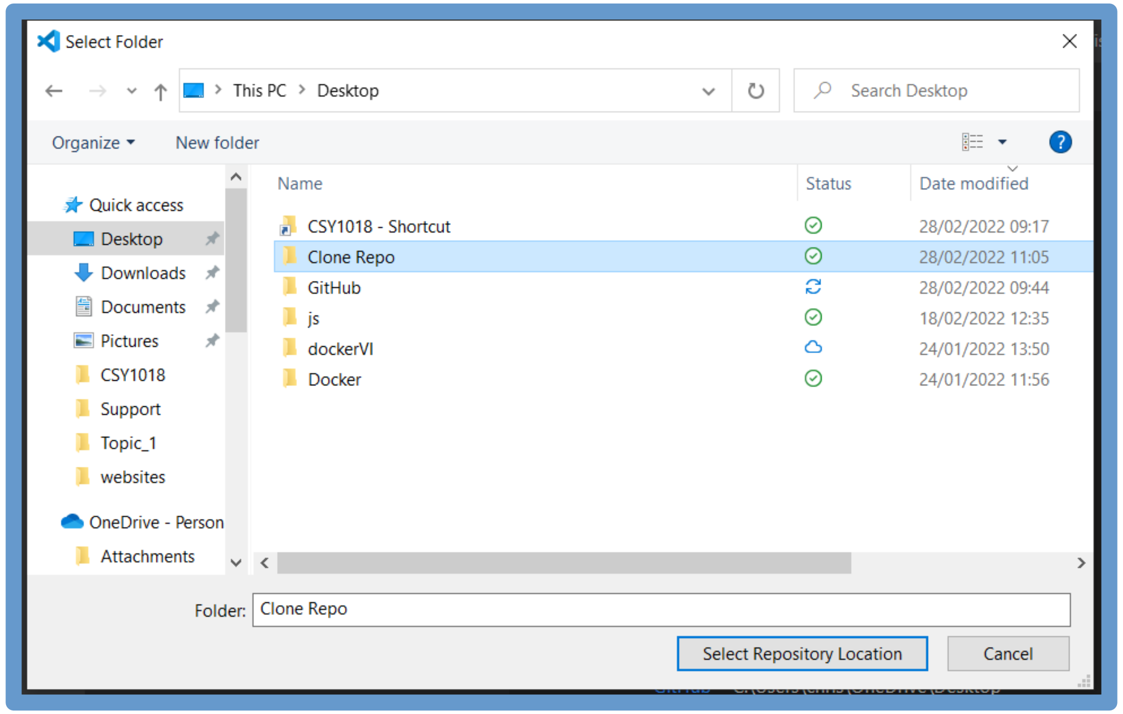


https://github.com/ctraff998/GitHub.git

ctraff998/GitHub https://github.com/ctraff998/GitHub.git

Forking pt9

- You will then be asked where you want to save the files
- In this example I have created a new folder on my Desktop called Cloned Repo but you can save this anywhere


Select Folder

This PC > Desktop

Search Desktop

Organize ▾ New folder

Name	Status	Date modified
CSY1018 - Shortcut	✓	28/02/2022 09:17
Clone Repo	✓	28/02/2022 11:05
GitHub	⟳	28/02/2022 09:44
js	✓	18/02/2022 12:35
dockerVI	☁	24/01/2022 13:50
Docker	✓	24/01/2022 11:56

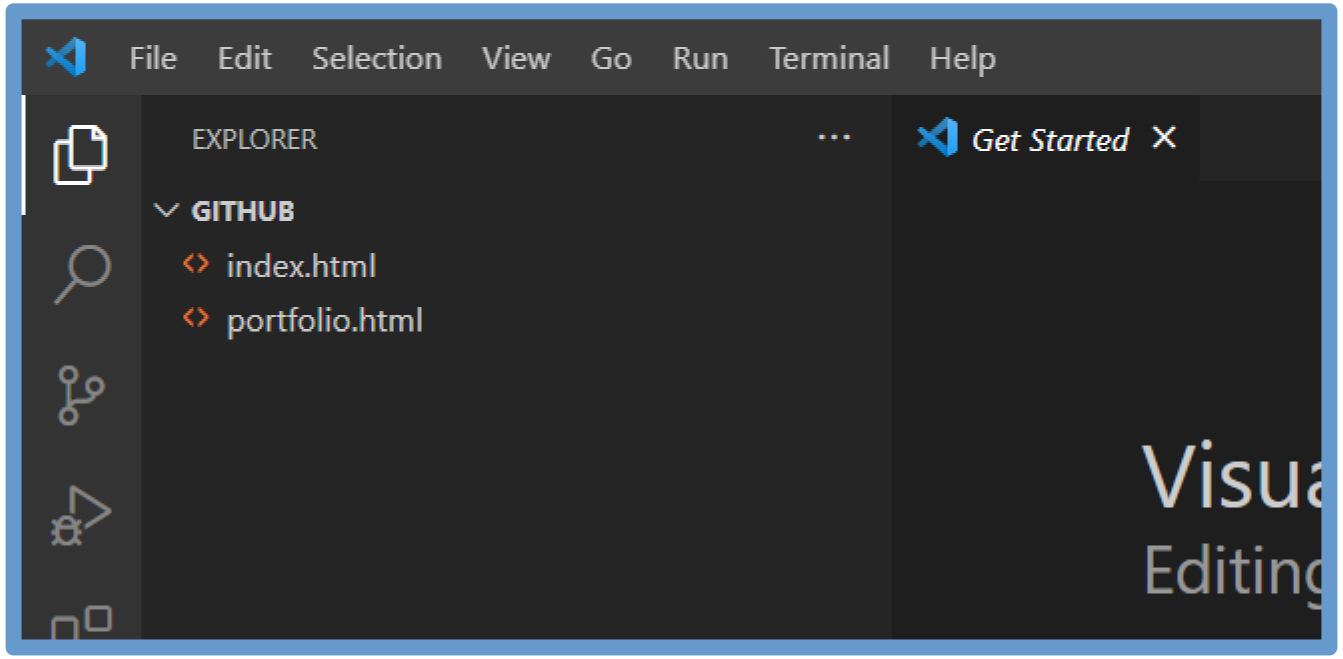
Quick access
Desktop
Downloads
Documents
Pictures
CSY1018
Support
Topic_1
websites
OneDrive - Person
Attachments

Folder: Clone Repo

Select Repository Location | Cancel

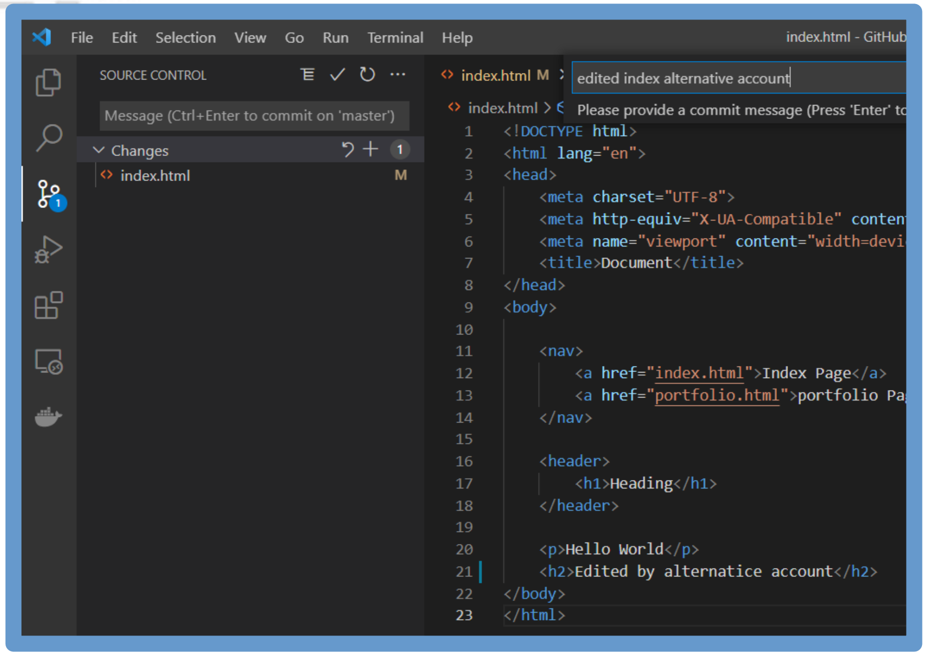
Forking pt10

- We can now see the files from the repository in Visual Studio Code



Forking pt11

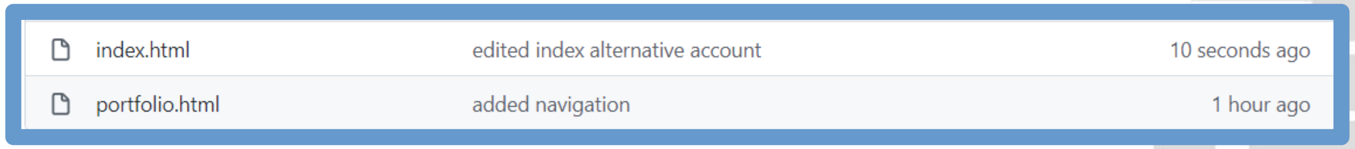
- If we make a change to our code (then commit and push)
- We can see that the change has appeared in our repository (still on the forked repository)


File Edit Selection View Go Run Terminal Help
index.html - GitHub
SOURCE CONTROL
Message (Ctrl+Enter to commit on 'master')
Changes
index.html M
index.html M

edited index alternative account

Please provide a commit message (Press 'Enter' to

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" conten
    <meta name="viewport" content="width=devi
    <title>Document</title>
</head>
<body>

    <nav>
        <a href="index.html">Index Page</a>
        <a href="portfolio.html">portfolio Pa
    </nav>

    <header>
        <h1>Heading</h1>
    </header>

    <p>Hello World</p>
    <h2>Edited by alternatice account</h2>
</body>
</html>
```



index.html	edited index alternative account	10 seconds ago
portfolio.html	added navigation	1 hour ago

Exercises pt2

- Create an account on GitHub (https://github.com/)
- Create a new project in Visual Studio Code
- Initialize it on GitHub (set to public)
- Add a new file and commit it
- Sync it to GitHub
- Check the repository on GitHub to see if the new file has been added
- Set up GitHub pages and make your website live (check you can access it using the URL)