

FIT3179 DATA VISUALISATION

Week 4 Lab Activity: Introduction to HTML and GitHub

1.	Installing Visual Studio Code	2
2.	Installing Add-ons for Visual Studio Code	6
2.1	Live Server Add-on	6
2.2	Vega Preview Add-on	7
2.3	Vega Viewer Add-on	7
3.	Introduction to HTML	8
3.1	Understanding the structure of HTML	9
3.2	Creating an HTML project in Visual Studio Code	10
3.3	Understanding some basic HTML elements	12
3.4	Additional Resources for other HTML elements.	15
4.	Introduction to GitHub	16
4.1	Setting up a GitHub account	16
4.2	Updating a GitHub Project	18
4.3	Using GitHub Pages to make the webpage publicly accessible.	20

1. Installing Visual Studio Code

Visual Studio Code is a code editor which is optimised for creating and debugging modern websites and cloud applications. There are many benefits of using a code editor for HTML programming:

- Syntax Highlighting: highlights content when you make syntax errors
- Auto-completion: suggests elements based on what is being typed
- Sensible formatting: easy formatting so all nested code is clearly understood

To download Visual Studio Code on your local machine, visit <https://code.visualstudio.com>.

For **Windows users**,

1. Click on the button mentioning “Download for Windows – Stable build” (Figure 1). If you have a 32-bit Windows Operating System, click on the arrow next to it and select “other downloads” option (Figure 2).

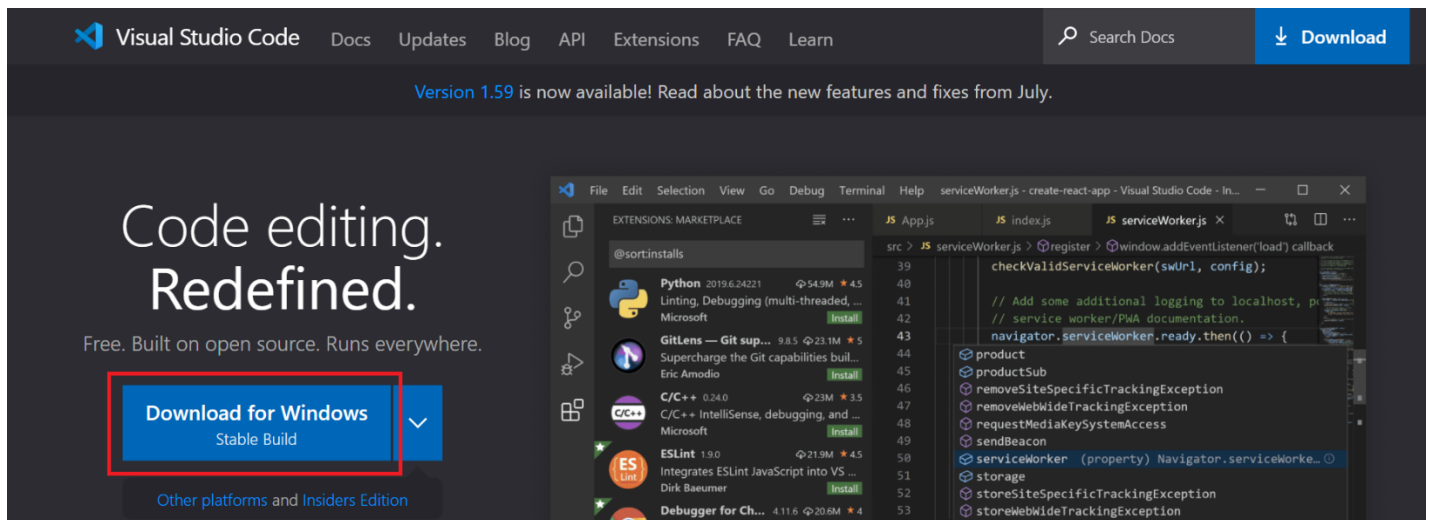


Figure 1. Download for Windows 64-bit OS

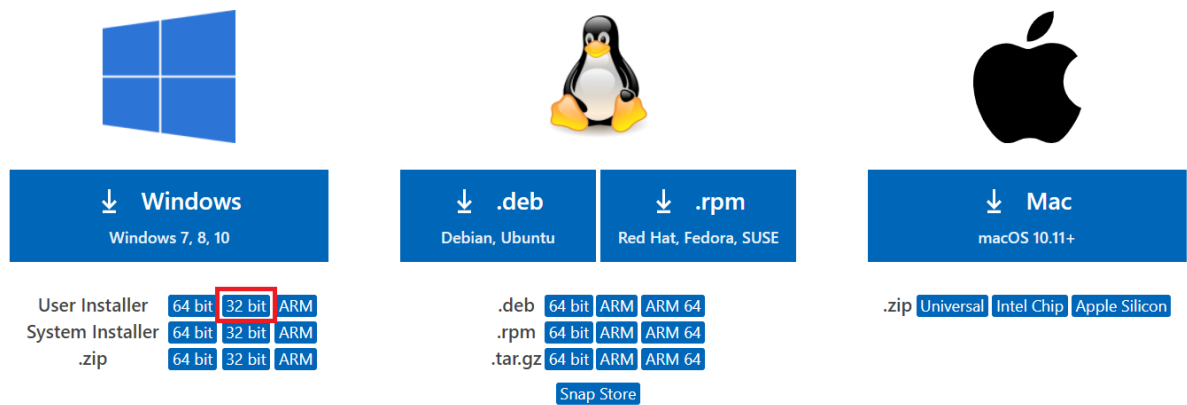


Figure 2. VS code download for Windows 32-bit OS

2. Open the downloaded **VSUserSetup[version].exe** file and setup the configuration as following:
 - a. Accept the agreement and click Next.
 - b. In the **Select Destination Location** panel, keep the folder structure as it is and click Next. This will be the location where your Visual Studio Code will be installed.

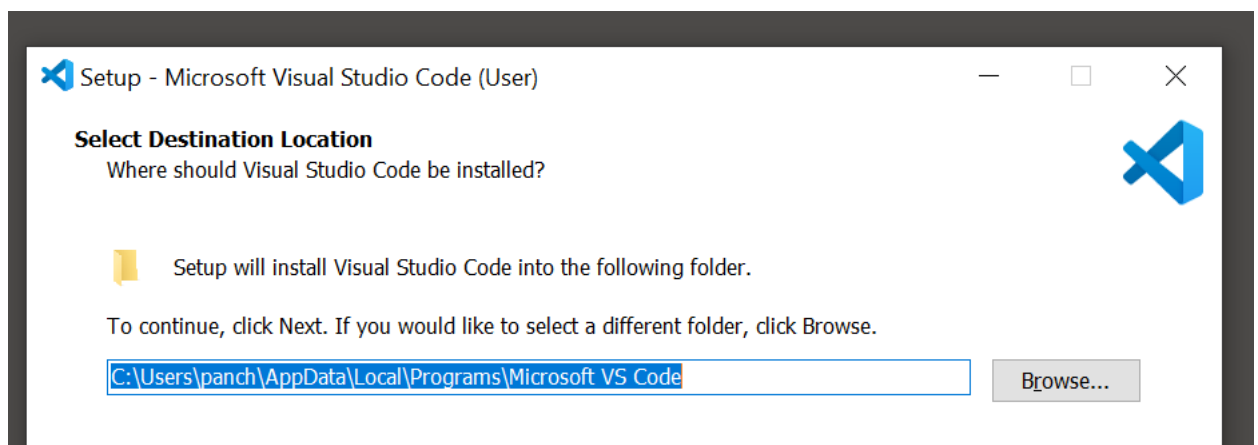


Figure 3. Selecting Location to install Visual Studio Code

- c. No changes are to be made in the **Select Start Menu Folder** panel. Click Next.
- d. In the **Select Additional Tasks** panel, check on all the options and click Next. If you do not want a shortcut icon for the application to be created on your desktop, uncheck the "Create a desktop icon."

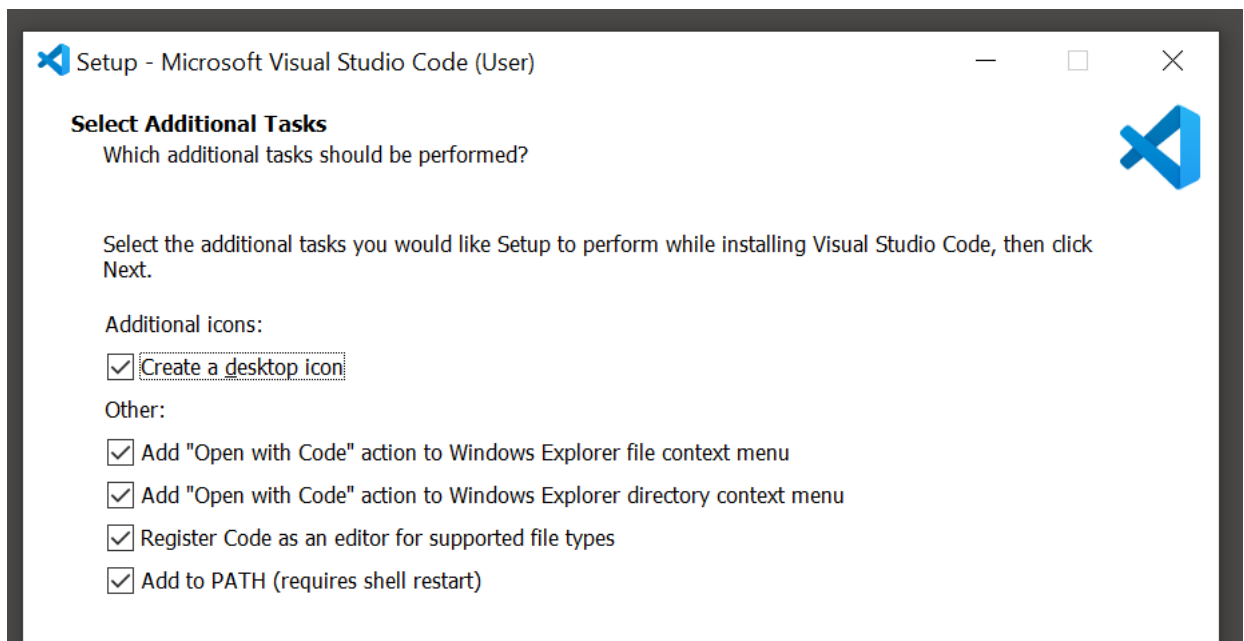


Figure 4. Options to perform additional tasks during installation

e. Click on Install. This will now install the Visual Studio Code on your machine.

For **Mac** users,

1. Click on the button mentioning "Download for Mac – Stable build". If the option is not available visit <https://code.visualstudio.com/#alt-downloads>. Then click on Mac download as shown in (Figure 5).

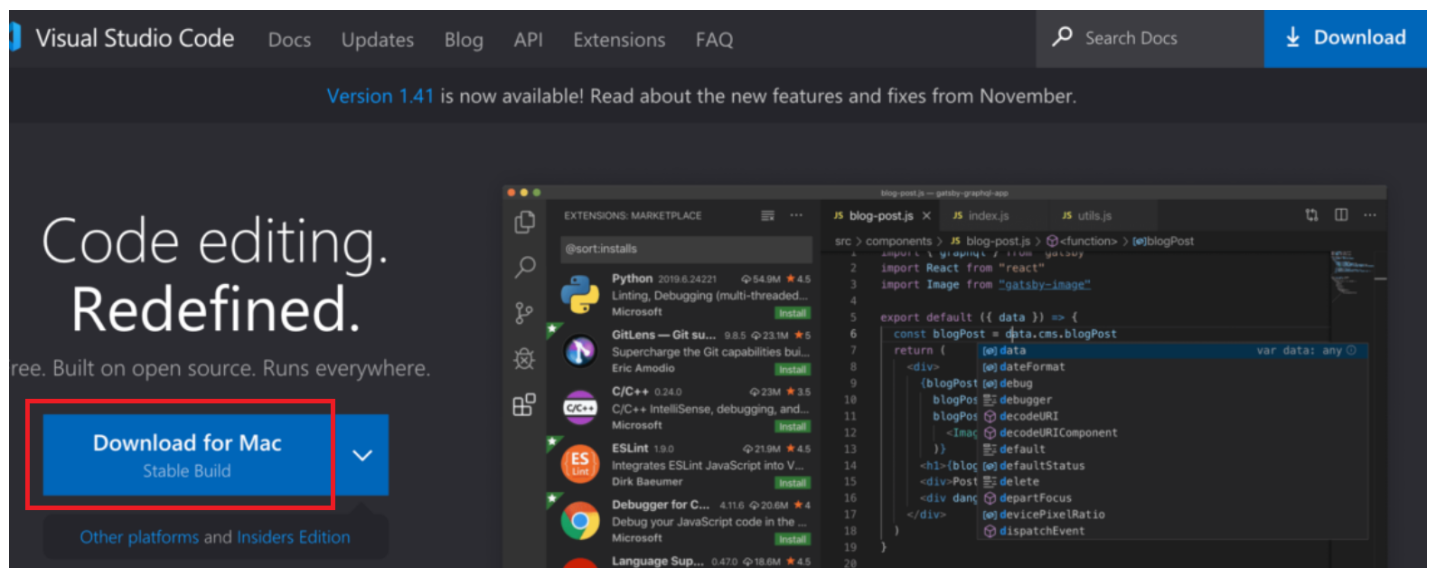


Figure 5. Universal download for Mac

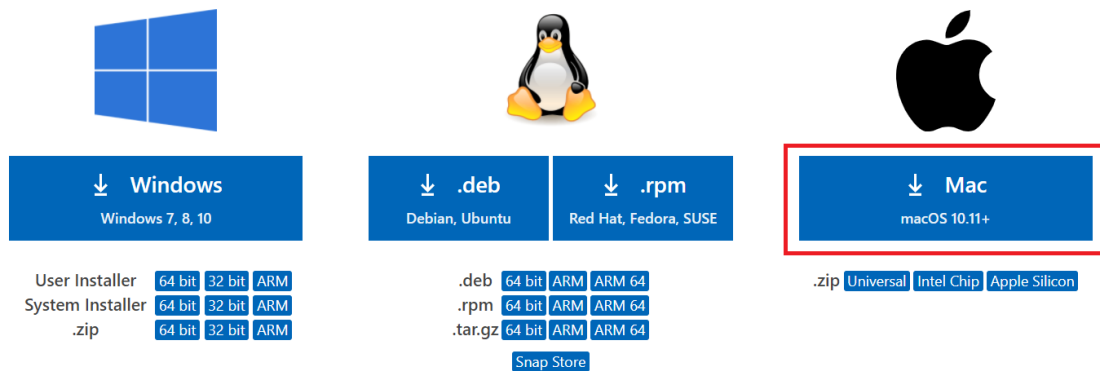


Figure 6. Alternative download for Mac

2. Once the zip file is downloaded, click to open it. It will give a prompt as shown in (Figure 7). Click on Open and it will open the application on your machine.

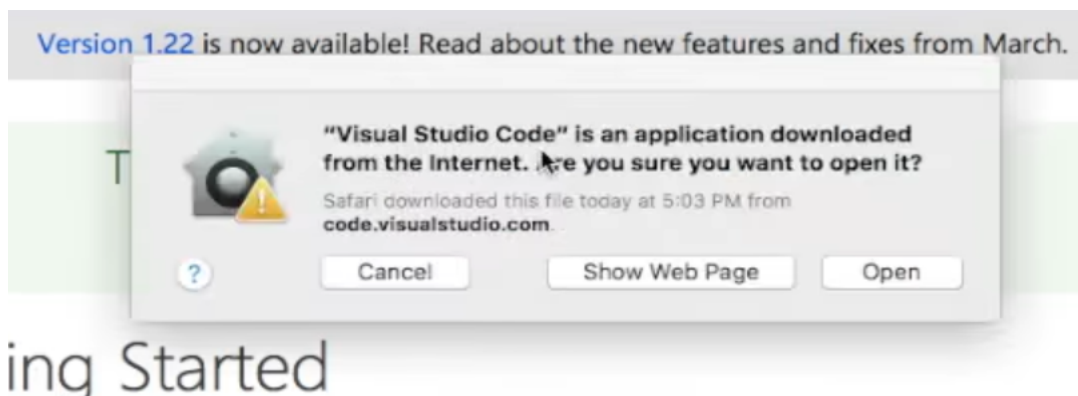


Figure 7. Prompt after clicking on the extracted application

3. The application might still not be visible on your launchpad. To view it in your launchpad, go to your Downloads folder and drag and drop the application into your Applications folder (Figure 8).

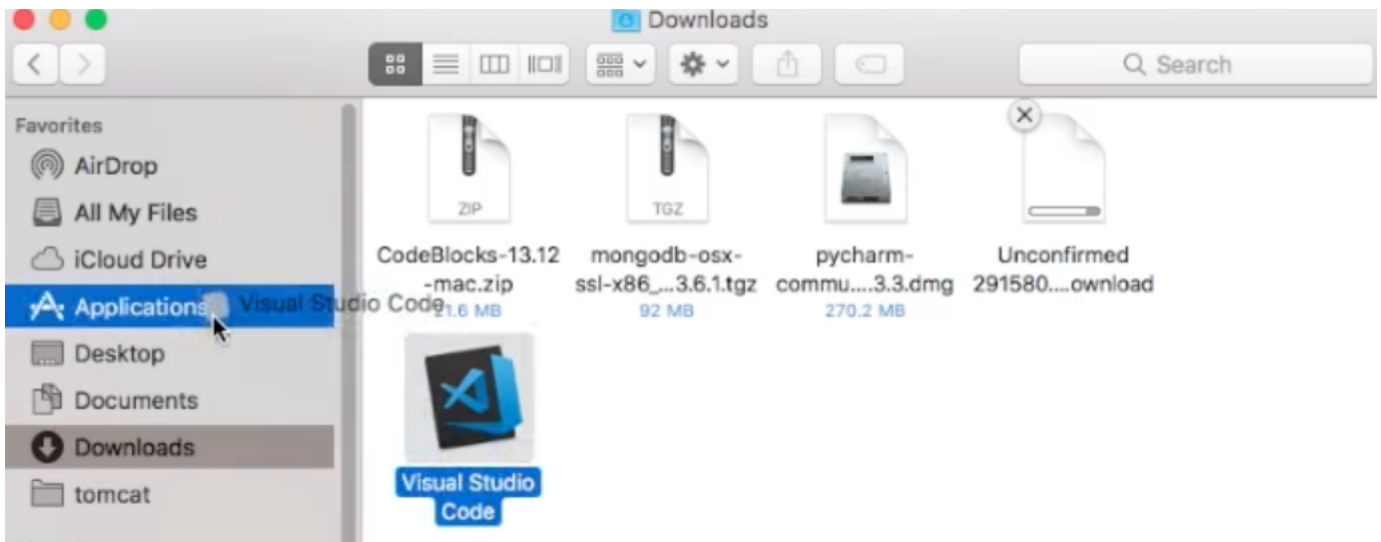



Figure 8. Dragging and dropping the application to the Applications folder

2. Installing Add-ons for Visual Studio Code

A. Live Server Add-on

This add-on is available in Visual Studio code and will make it easy for you to **see the reflected code on a web page** for the changes made in HTML code.

- i. Open your Visual Studio Code application.
- ii. In your Activity panel on the left-hand side, click on the **extensions**  button as shown in (Figure 9).

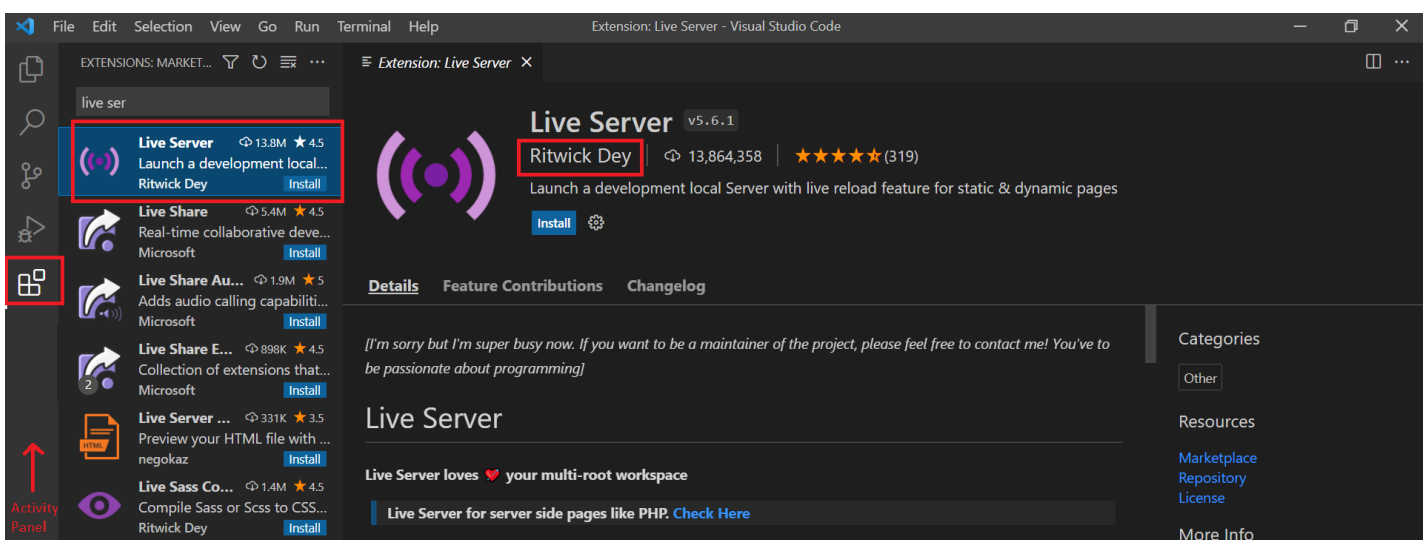


Figure 9. Downloading Live Server Extension on Visual Studio Code.

- iii. Select the extension as highlighted in Figure 9. And click on Install.

- iv. After the installation is complete, if the extension has not been enabled, click on the “Enable” button.
- v. We will see how to run this extension in Activity 2.

B. Vega Preview Add-on

This **Vega Preview** extension is available on Visual Studio code and can be utilised to see a **preview** of your **Vega and Vega-Lite JSON files**.

- i. Open your Visual Studio Code application.
- ii. In your Activity Panel on the left side, click on the extensions button as shown in (Figure 10).
- iii. Select the extension as highlighted in Figure 10. Click Install.

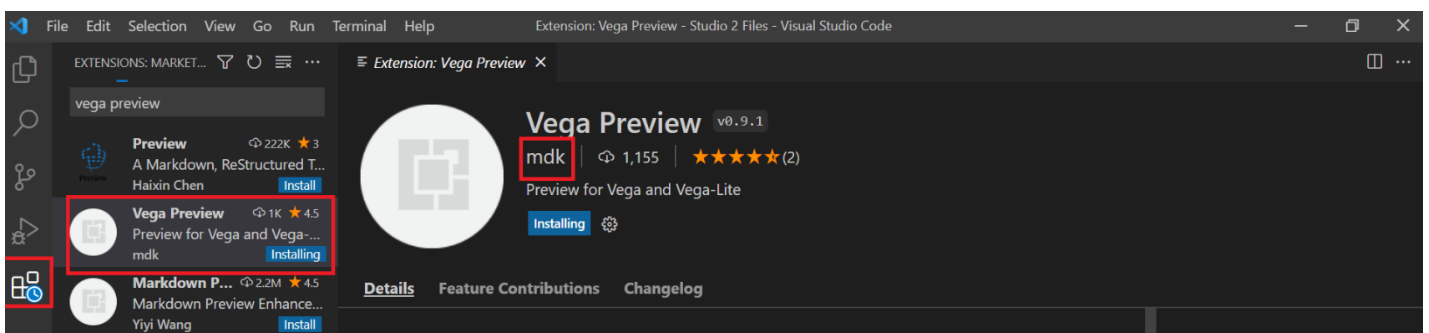


Figure 10. Installing Vega Preview add-on

B. Vega Viewer Add-on

This extension can be used to **view graphs** based on the Vega and Vega-Lite JSON files. It also **allows exporting the graphs in SVG and PNG formats**.

- i. Follow the same process as 2.1 and 2.2.
- ii. Click on the extensions button and search for Vega Viewer by the author Random Fractals inc.

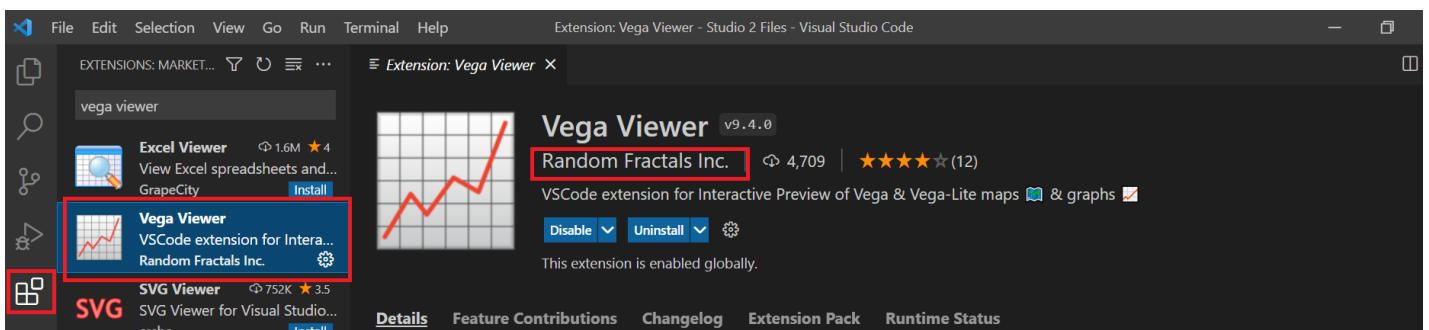


Figure 11. Installing the Vega Viewer add-on

3. Introduction to HTML

HTML (Hyper Text Markup Language) is a markup language **used for creating web pages**. It is considered as the skeleton of web pages. It provides a **structure** for the content appearing on web pages, such as textual content, images or videos.

This activity will help you to understand the structure of HTML and create your first web page using some basic HTML elements. A limitation with creating web pages using HTML is that it only allows you to **create basic elements** but does not allow you to style your content or make it interactive.

A. Understanding the structure of HTML

HTML code is written by creating a series of HTML elements. An HTML element tells the browser the format in which the content should be displayed. The element can be defined by the following:

- A start tag
- Some textual content (optional)
- An end tag

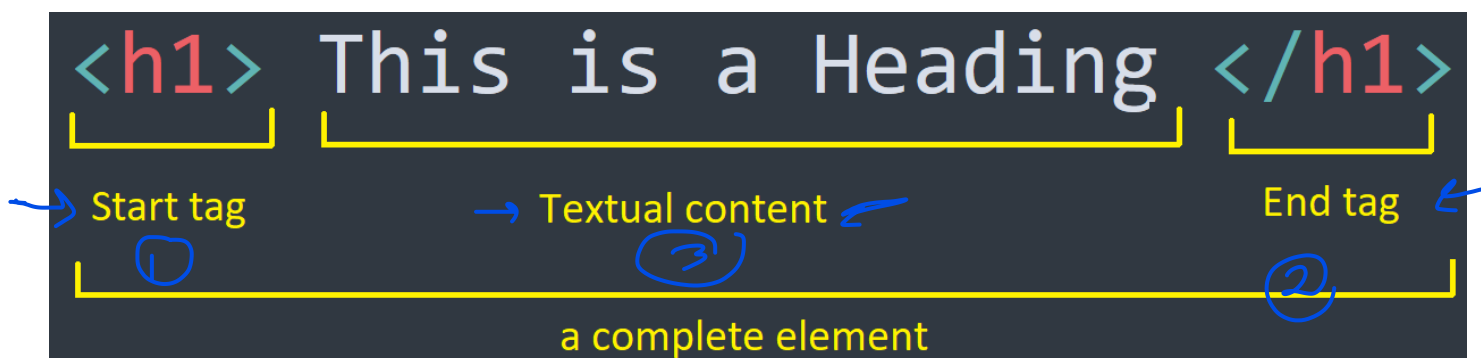


Figure 12. Example of an HTML element

The start tag and the end tag are written in `< >` notation to make it easy for the browser to understand where an element begins and ends when an HTML document is processed. You will have multiple elements of the same type in an HTML document. For this reason, it is important to add a **`/`** **in the end** tag before the element name.

A simple HTML document can be created by using the format as shown in (Figure 13):

```
<!DOCTYPE html>
<html>
<head>
  <title>Document Title</title>
</head>
<body>
  <h1>This is a Heading </h1>
  <p> This is a paragraph. </p>
</body>
</html>
```

Figure 13. Basic HTML structure

- **!Doctype element** lets the browser know that this document should be processed as an HTML document.
- **Html element** is the **root of an HTML document**. It will have two children elements nested in it (head and body element)
- **Head element** will be used to **link your CSS and JS** files to the HTML document and to give a **title** for this web page.
- **Body element** is the **container** for all your visible content, such as images, videos or textual content.

B. Creating an HTML project in Visual Studio Code

1. Create a new folder on your Desktop (alternatively, you can create it anywhere you would like to create your HTML project). Name it appropriately without any spaces.
2. Open your Visual Studio Code application. Go to File -> Open Folder. Browse the locations where you have created your folder and click Select folder.

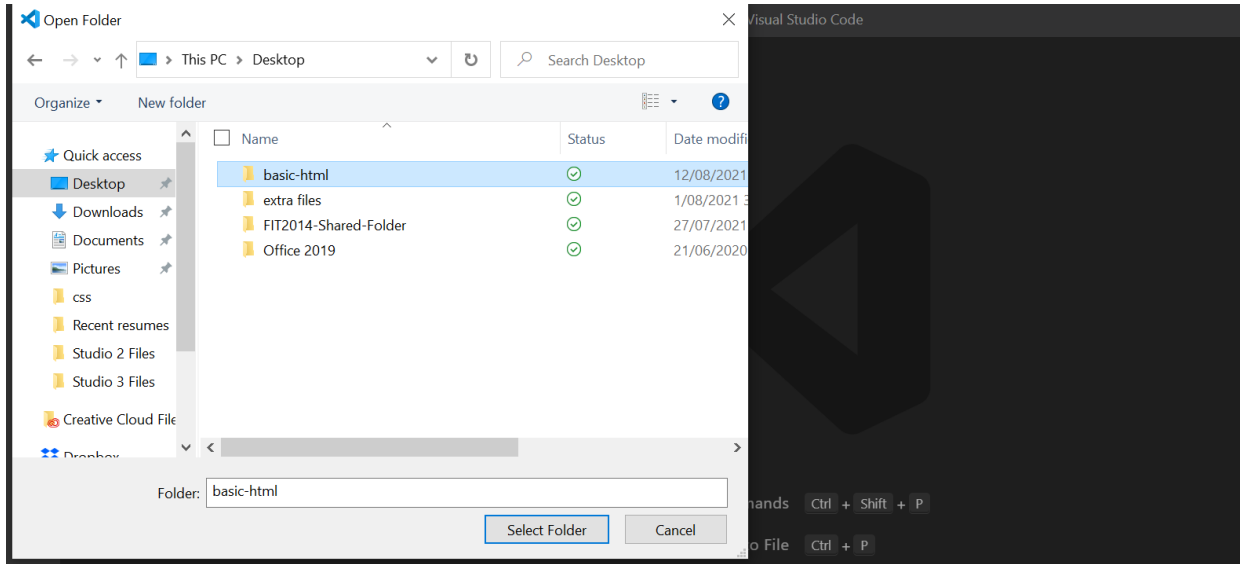


Figure 14. Open a folder

3. In your Explorer Panel, click on the “New file” button to create a new file. All HTML files are saved with the extension .html otherwise, the document will not be processed as an HTML document. Save this new file as `index.html` and press enter.

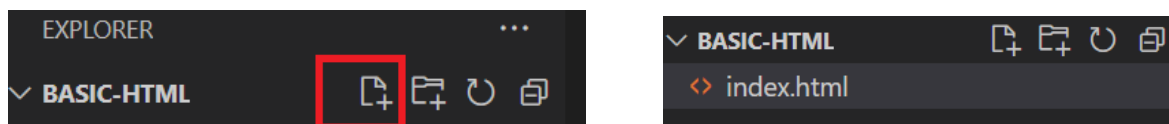


Figure 15. Creating a new HTML file

4. In this HTML file, create the structure as shown in Figure 13. A shortcut to creating the basic structure of an HTML document is to type `html:5` and select the suggestion, as shown in Figure 16. Remove the meta elements as they are not required for this unit.

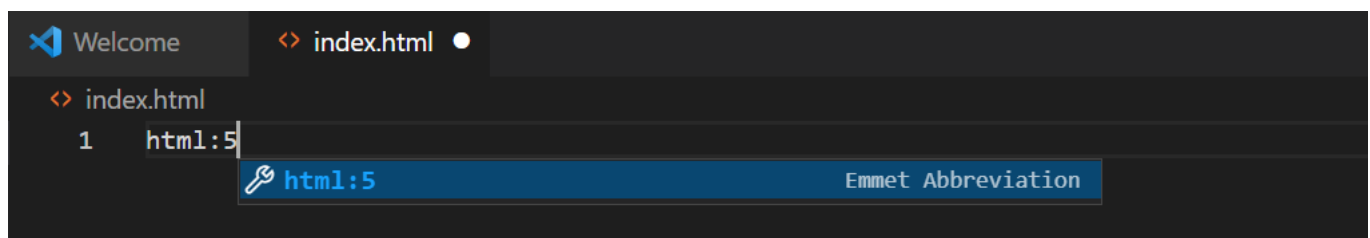
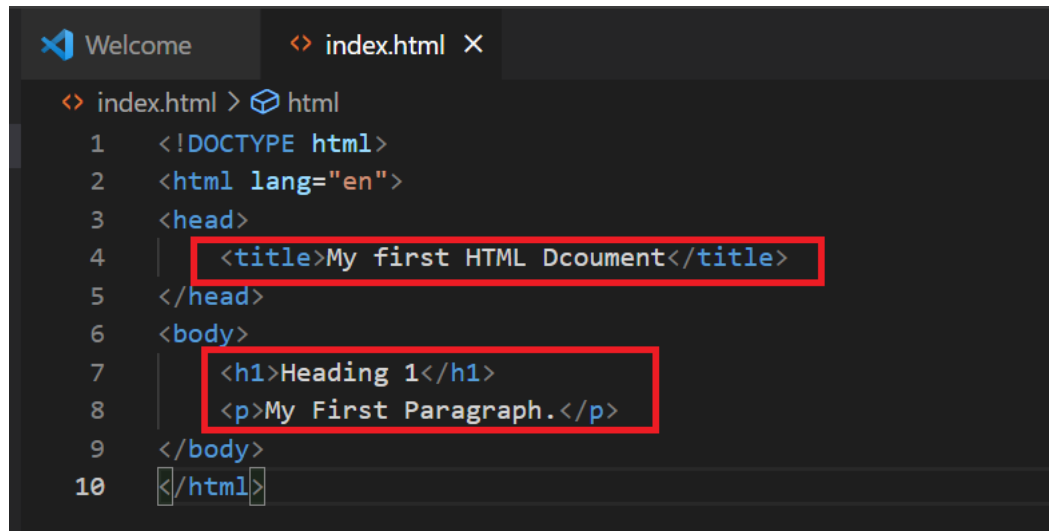


Figure 16. HTML shortcut to create basic HTML structure

5. Change the content in the `<title>` element to rename your webpage to something relevant.

6. Create a new HTML element inside the body element. We will be creating our first heading and first paragraph, as shown in Figure 17.



```
<?xml version="1.0" encoding="UTF-8" ?>
<?DOCTYPE html>
<html lang="en">
  <head>
    <title>My first HTML Document</title>
  </head>
  <body>
    <h1>Heading 1</h1>
    <p>My First Paragraph.</p>
  </body>
</html>
```

Figure 17. First HTML document

7. Every time a change is made in the HTML document, it **needs to be saved** in order for the change to be reflected on the web page.
8. If you have installed and enabled the **Live Server Add-on** from the previous activity, right-click on the file name and click on “Open with Live Server”. It will open the file in your browser. When a file is opened with the Live Server add-on, you can see your changes getting reflected automatically on the web page.
9. Alternatively, if Live Server has not been installed, go to Run -> Run Without Debugging.

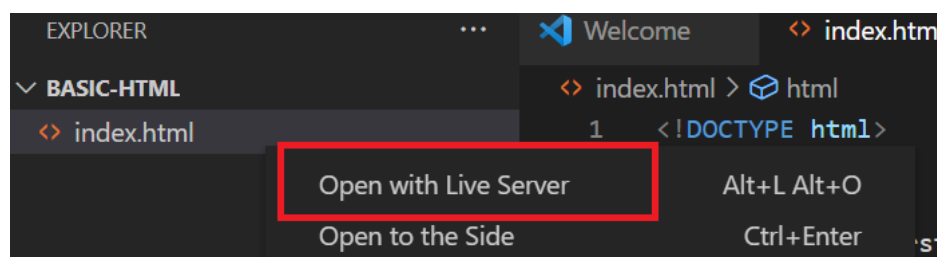


Figure 18. Opening file with Live Server Add-on

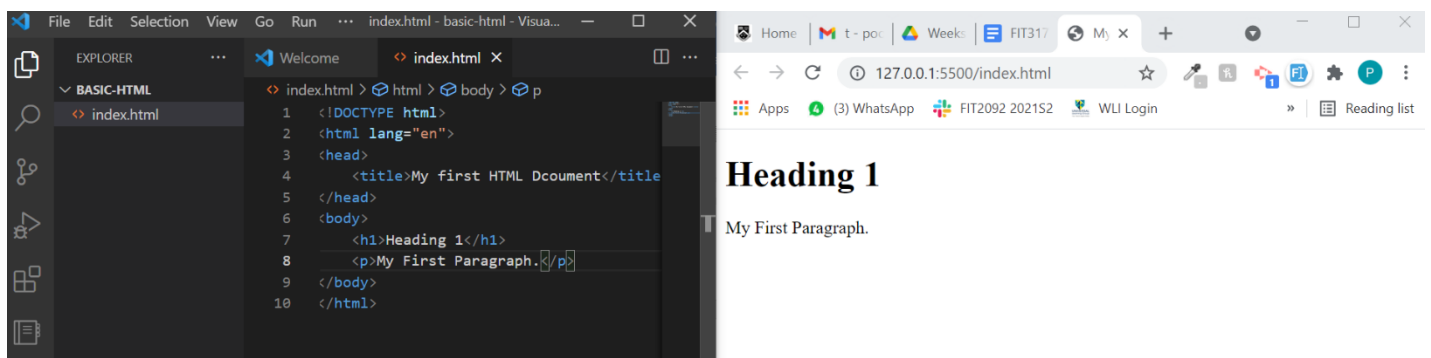


Figure 19. Saving and viewing the HTML document in the browser

C. Understanding some basic HTML elements

1. Creating Paragraphs and Headings in HTML

<p> elements is used to define paragraphs in HTML. For each paragraph that is added to the HTML document, a separate element needs to be created.

Six levels of headings could be used in an HTML document. **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** and **<h6>**. The default styles of the headings vary in font size. The heading should be used to logically structure the content of your document.

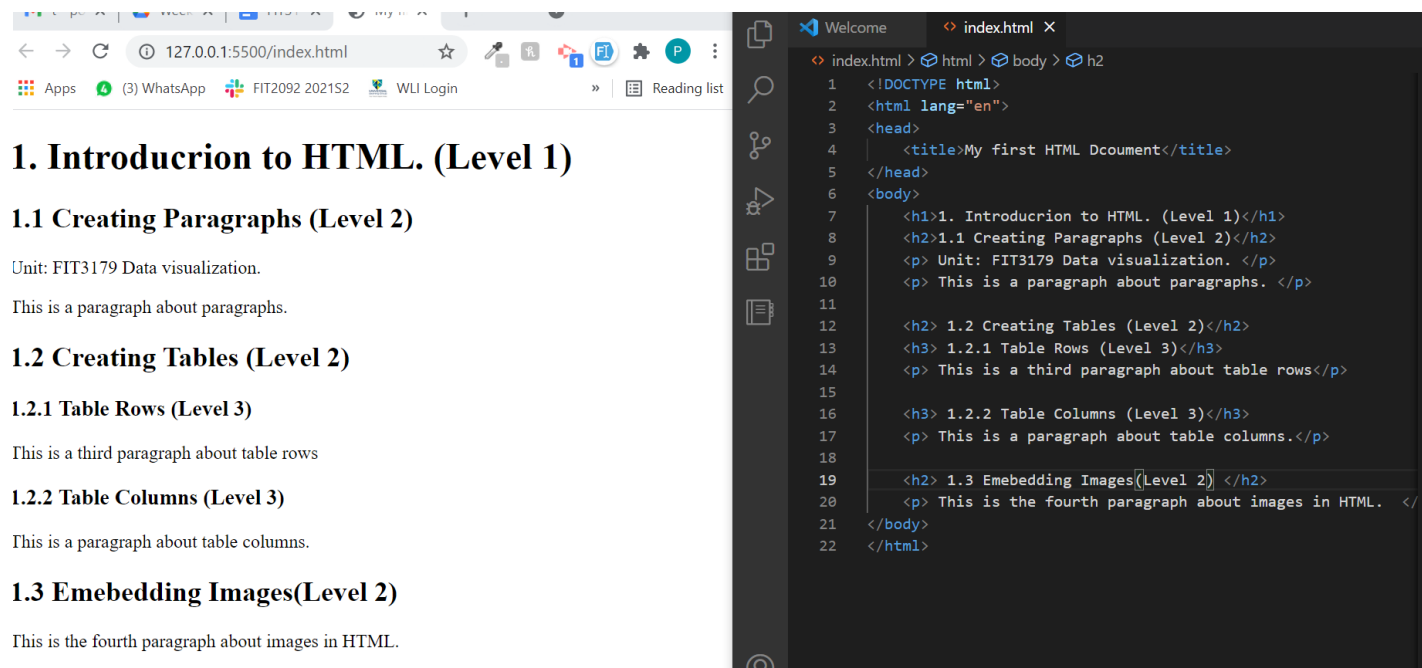


Figure 20. Example of Paragraphs and Headings in an HTML document.

2. Creating Tables in HTML

A table in HTML is a combination of 4 different elements.

- **<table>** element represents the outer container for creating a table
- **<th>** element represents the first heading row in a table
- **<tr>** element represents a row
- **<td>** element represent columns in a single row

The **<table>** element should be the outermost element in this hierarchy. **<tr>** element is used to create a single row in the table. In the example in the (Figure 21), **<tr>** element is the outer element and inside the **<tr>** element, **<th>** element is used to create the heading row. Each **<th>** element represents one column in a particular row. **<td>** element is used to create a single column that is not a table heading.

Table of Student and Grade

First Name	Last Name	Grade
John	Doe	HD
Jill	Jackson	D
Jane	Smith	C

tr
td

```
6 <body>
7 <h1> Table of Student and Grade </h1>
8
9 <table border="1" cellpadding="10%">
10 <tr>
11 <th>First Name</th>
12 <th>Last Name</th>
13 <th>Grade</th>
14 </tr>
15
16 <tr>
17 <td>John</td>
18 <td>Doe</td>
19 <td>HD</td>
20 </tr>
21 <tr>
22 <td>Jill</td>
23 <td>Jackson</td>
24 <td>D</td>
25 </tr>
26 <tr>
27 <td>Jane</td>
28 <td>Smith</td>
29 <td>C</td>
30 </tr>
31 </table>
32 </body>
```

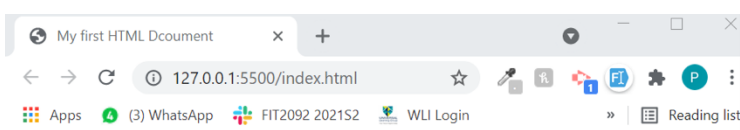
Figure 21. HTML code for tables and output on the browser

3. Embedding Images in HTML

Images can be embedded into a HTML document by using an `` element. **Note:** This element is a self-closing element (an element that does not require an end tag.)

Some HTML elements require attributes to be added to the elements to modify their structure. For example, an image element can be created, but unless a path is given to the image file, it cannot be viewed. For example: ``. Src is the attribute that takes in the path to the image file.

Visit shorturl.at/opLY5 and save this image file into the same folder as your index.html file.



Embedding Images in HTML.



```
File Edit Selection View Go Run ... index.html - basic-html - Visual ...
Welcome index.html x
index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>My first HTML Dcoument</title>
5 </head>
6 <body>
7 <h2> Embedding Images in HTML. </h2>
8
9 
10 </body>
11 </html>
```

Figure 22. Embedding images in an HTML document

4. Creating Hyperlinks in HTML

Hyperlinks are clickable links that redirect users to another HTML web page or another online resource. Hyperlinks can be created in HTML by using the `<a>` element. The basic structure of a hyperlink can be seen in the Figure below. The `href` attribute is used to define a path to an online resource.

```
<a href="LINK_TO_ANOTHER_RESOURCE"> Text to be viewed in browser </a>
```

Figure 23. Example of a hyperlink

Copy the link <https://www.monash.edu/> and paste it in the `href` attribute of an `<a>` element. Modify the text between the start and end tags to “Click here to view the Monash website”.

Hyperlinks in HTML.

[Click here to view the Monash Website.](https://www.monash.edu/)

```
6  <body>
7  <h2> Hyperlinks in HTML. </h2>
8
9  <a href="https://www.monash.edu/">
10  Click here to view the Monash Website.
11  </a>
```

Figure 24. Hyperlink in HTML document with output

5. Creating sections in HTML

A `<div>` element is used to **define a section or division** in a HTML document. It can be used to group multiple elements together. A `<div>` element does not represent any visible structure on the web page. It can be used to **create complex layouts** in an HTML document. Without CSS applied to the `<div>` element, there is no significant change visible in the structure on the HTML page. As shown in Figure 25, all elements in group 1 `<div>` elements had red coloured font (Done using CSS and will be covered in coming labs), and all the elements grouped in the second `<div>` element had green coloured font.

Hyperlinks in HTML.

Heading of first group

This paragraph is in 1st group

Heading of second group

This paragraph is in 2nd group.

} div
{ div

```
6  <body>
7  <h2> Hyperlinks in HTML. </h2>
8
9  <div style="color: red;">
10  <h2>Heading of first group</h2>
11  <p>This paragraph is in 1st group</p>
12  </div>
13
14  <div style="color: green;">
15  <h2>Heading of second group</h2>
16  <p>This paragraph is in 2nd group.</p>
17  </div>
18  </body>
19  </html>
```

Figure 25. Use of `<div>` tag to group elements in 2 sections

D. Additional Resources for other HTML elements.

1. Lists in HTML (https://www.w3schools.com/html/html_lists.asp)
2. Formatting Text in HTML (https://www.w3schools.com/html/html_formatting.asp)
3. Writing comments in HTML (https://www.w3schools.com/html/html_comments.asp)

4. Introduction to GitHub

1. Setting up a GitHub account

To create an account on GitHub, visit <https://github.com/>. Enter your Monash student email to create an account on GitHub. Choose a unique username. If the username already exists, GitHub will not allow you to proceed further.

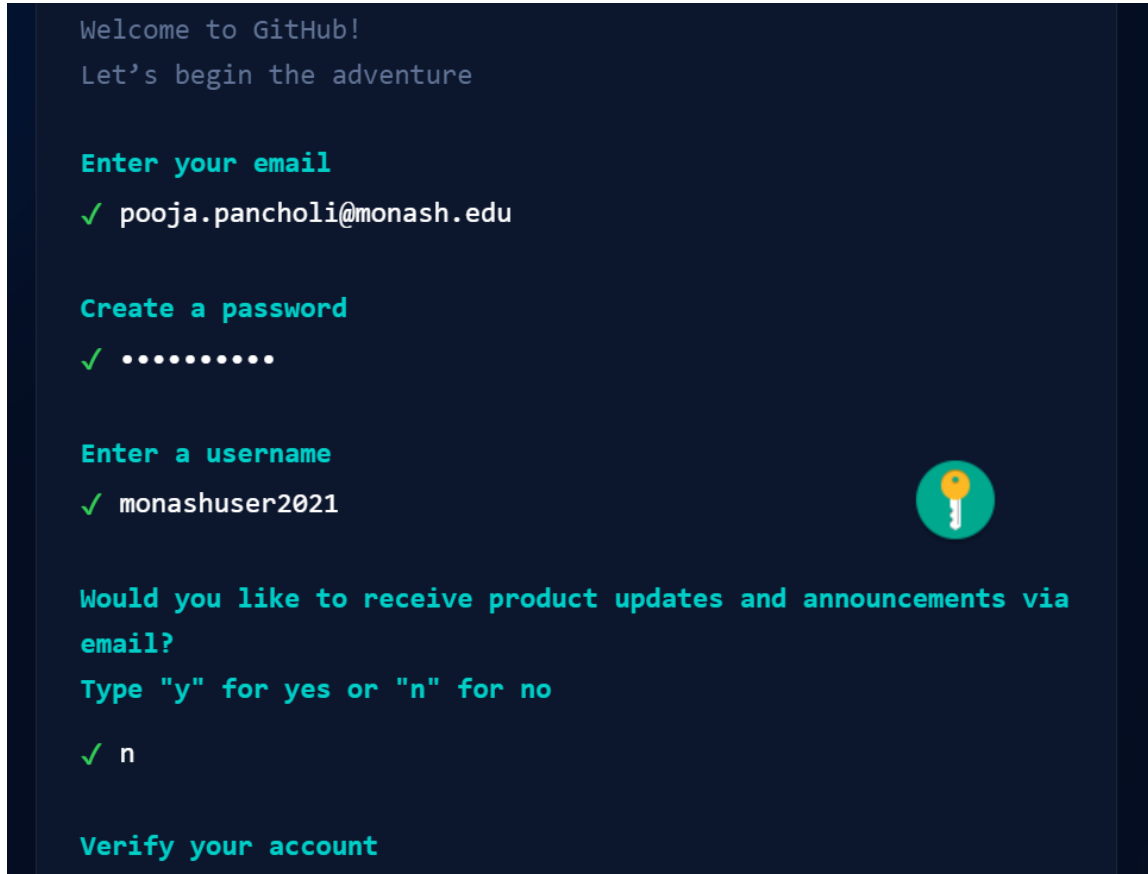
A screenshot of the GitHub sign-up process on a dark blue background. The text is in a light blue, monospace-style font. It starts with 'Welcome to GitHub!' and 'Let's begin the adventure'. Then it asks to 'Enter your email', showing a green checkmark and the email 'pooja.pancholi@monash.edu'. Next is 'Create a password', showing a green checkmark and a series of dots. Then 'Enter a username', showing a green checkmark and the username 'monashuser2021'. To the right of this step is a circular icon with a key. Below that, it asks 'Would you like to receive product updates and announcements via email?' and 'Type "y" for yes or "n" for no', showing a green checkmark and the response 'n'. At the bottom, it says 'Verify your account'.

Figure 26. Signing up for a GitHub account

Once the sign-up process is complete, GitHub will automatically log you in and redirect the user to the GitHub dashboard interface, as shown in (Figure 27).

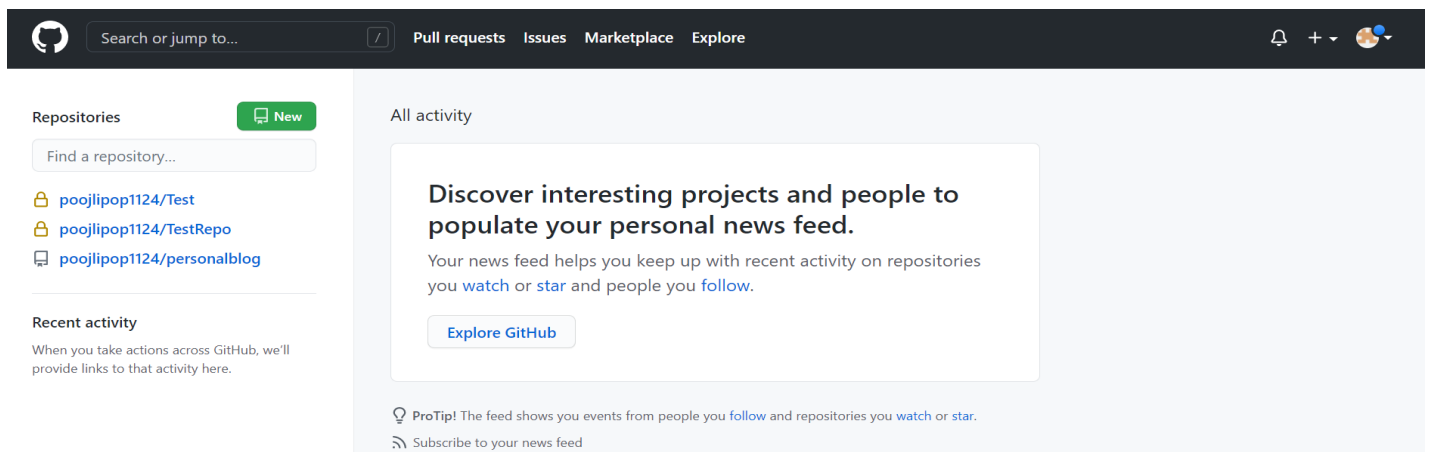



Figure 27. GitHub Interface

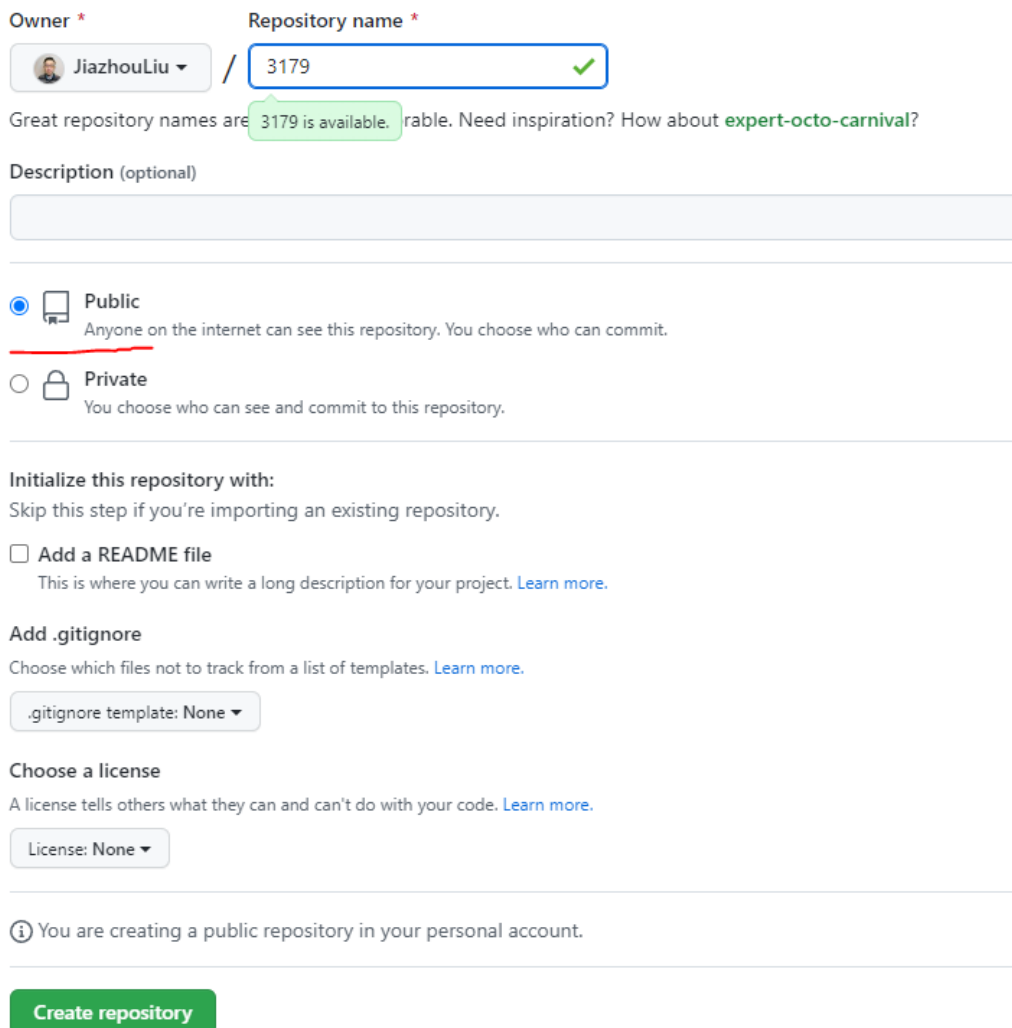
2. GitHub Repository

For working on a project, we can create repositories on GitHub which can be updated whenever there are changes in the project.



- 2.1. To create a GitHub repository, click on the  button on the side of the dashboard. Make sure that you select the **Public** option.


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)





Owner ^{*} / Repository name ^{*}

 JiazhouLiu / 3179 

Great repository names are  rable. Need inspiration? How about [expert-octo-carnival](#)?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

 You are creating a public repository in your personal account.

[Create repository](#)

Figure 28. Creating a new repository on GitHub

- 2.2. To upload your existing project on GitHub, click on the “upload an existing file” hyperlink as shown in (Figure 29).

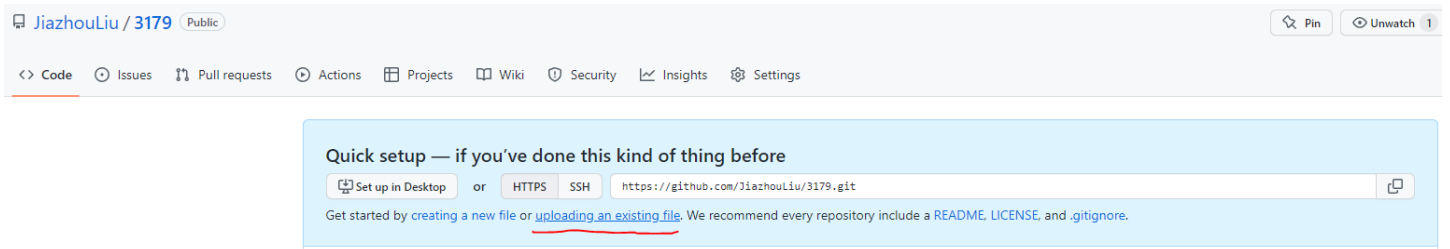


Figure 29. Uploading an existing file to the GitHub repository

- 2.3. Drag the index.html from your local HTML folder and drop it into the drop box as shown in (Figure 30).

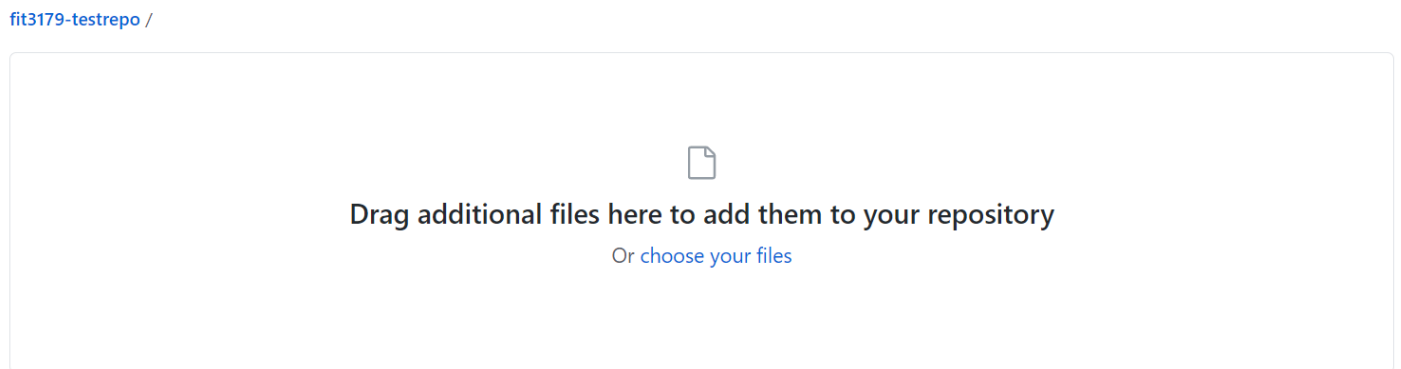


Figure 30. Dropbox to upload files on GitHub Repository

- 2.4. In the Commit Changes box, add an appropriate message to summarise the files that were uploaded. For example, database files uploaded, new images added, code changes updated and click on the Commit Changes button.
- 2.5. Your first file is now added to your repository. To view the contents of the uploaded file, click on the hyperlink with the file name on it, and it will show the contents of the file.

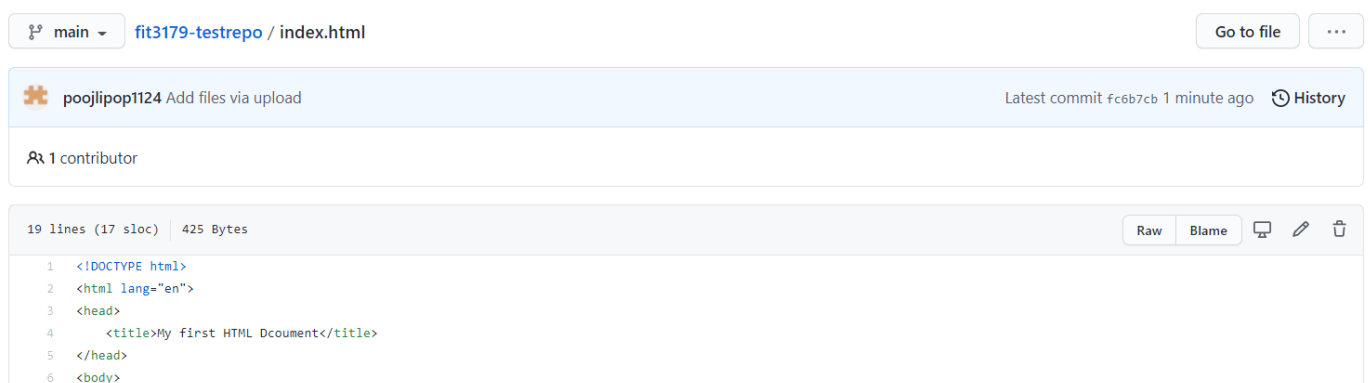


Figure 31. Contents of index.html file

3. Updating a GitHub Project

A GitHub project can be updated in two ways:

3.1. Uploading the updated file and committing the changes.

Step 1: Add a new paragraph and heading to your index.html file on your local machine.

Step 2: Click on your repository name, as shown in (Figure 32) and click on Add file button. Click on the Upload files hyperlink to upload a new file and repeat the process as shown in 4.1.

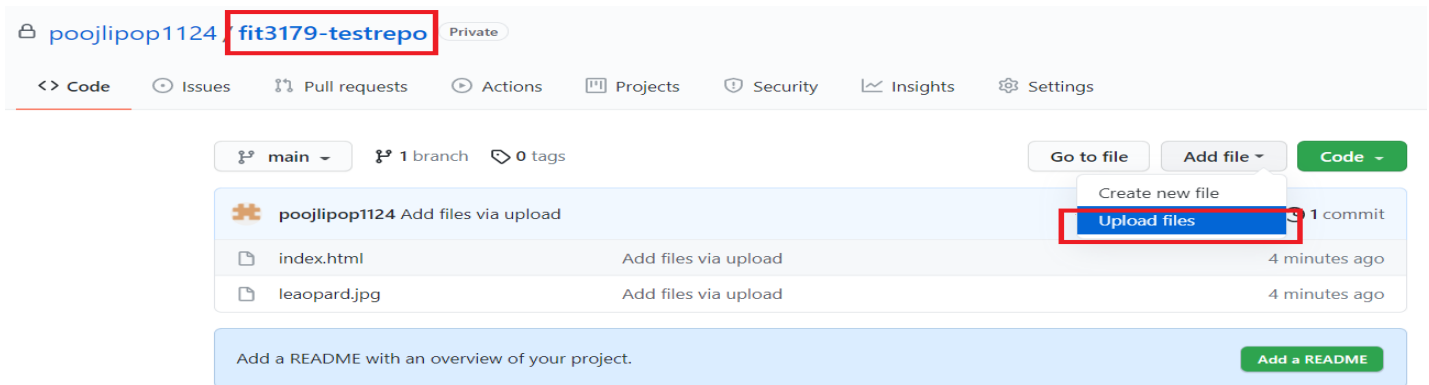


Figure 32. Uploading updated files to GitHub.

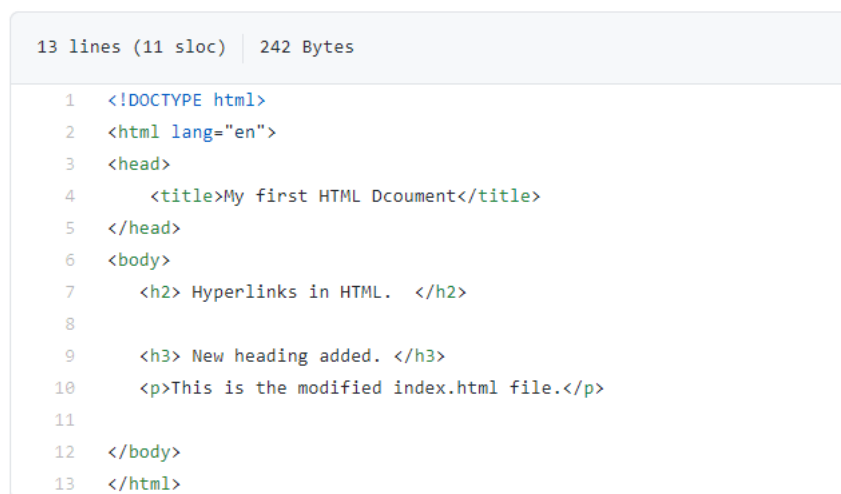


Figure 33. The modified HTML file is reflected on GitHub

3.2. Make changes on GitHub directly.

Step 1: Click on the hyperlink of the file that you wish to modify. In this case, click on index.html.

Step 2: Click on the button  to edit the raw HTML code on GitHub.

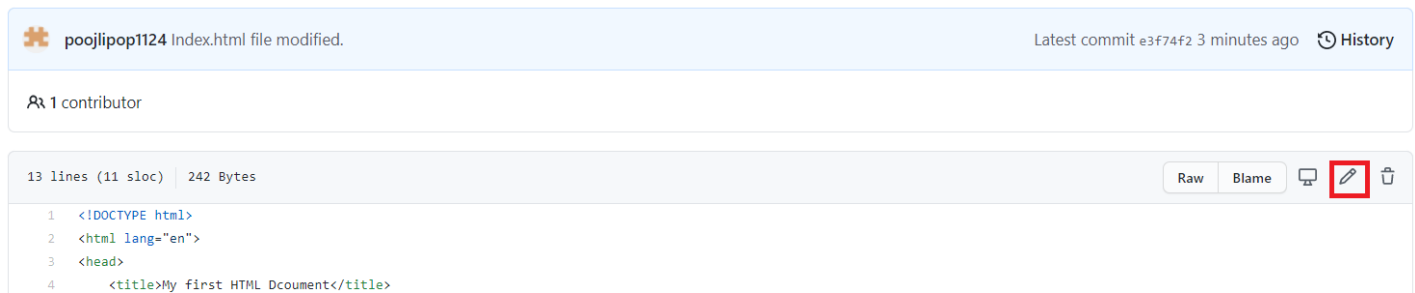


Figure 34. Opening GitHub file in Edit mode.

Step 3: GitHub will open the file in edit mode to add new lines of code to the HTML document. It will also **show the preview** of all the lines that have been modified (added or removed).

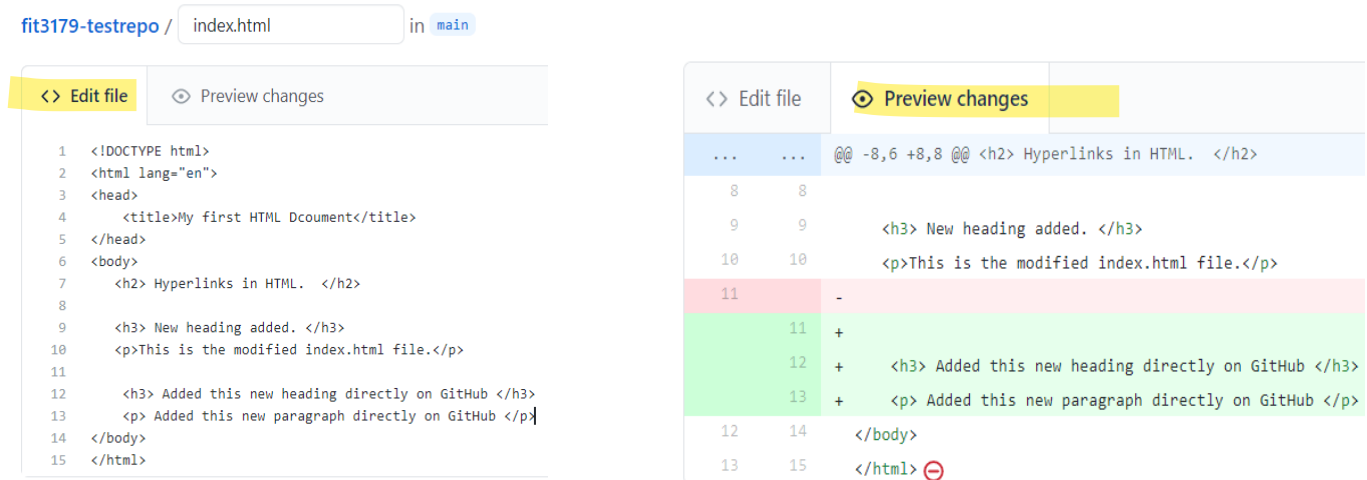


Figure 35. Editing files directly on GitHub and previewing changes

Step 4: Commit the changes and click on the **Commit changes** button.

4. Using GitHub Pages to make the **webpage publicly accessible**.

Step 1: Click on **Settings** in the top bar in the repository. Select the **pages** button on the left panel in the settings menu.

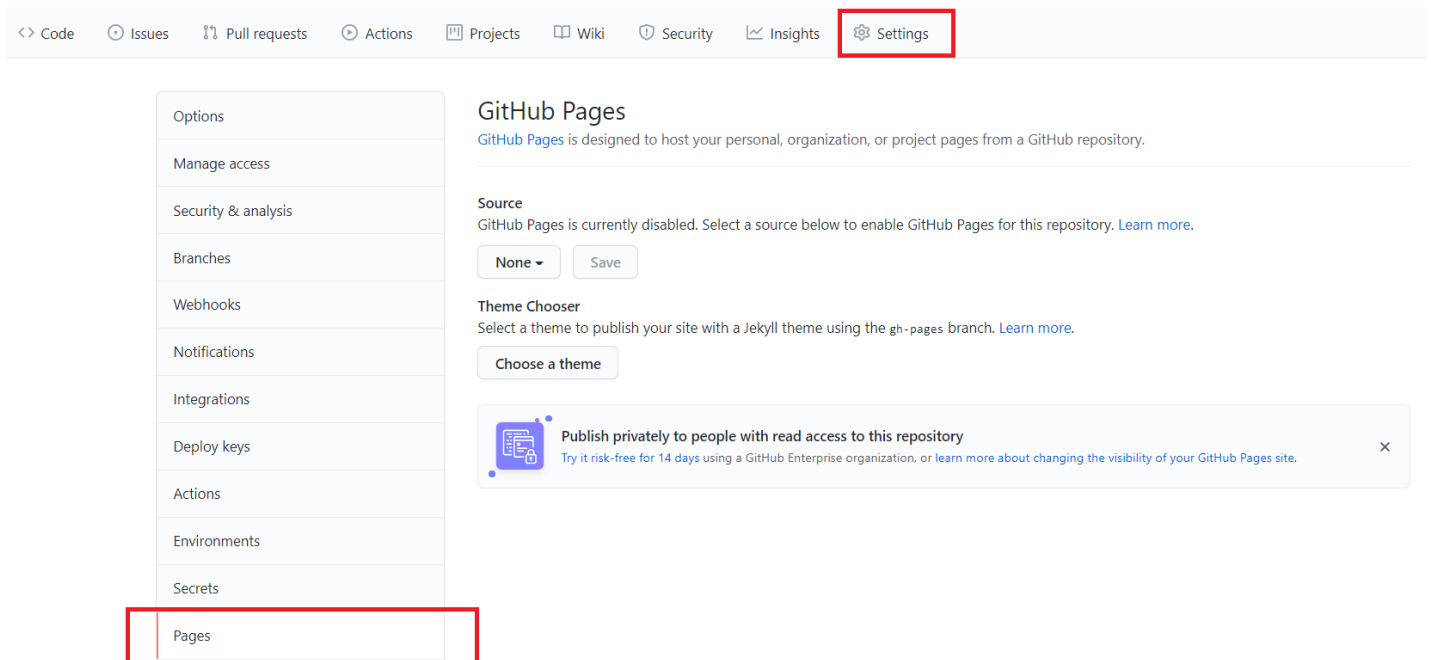


Figure 36. Pages settings to make a web page publicly accessible

Step 2: In the Source section, click on the dropdown menu that says None and select the branch that your **HTML file has been uploaded**. In this case, select the 'main' branch. Keep the second dropdown menu as and click on Save.

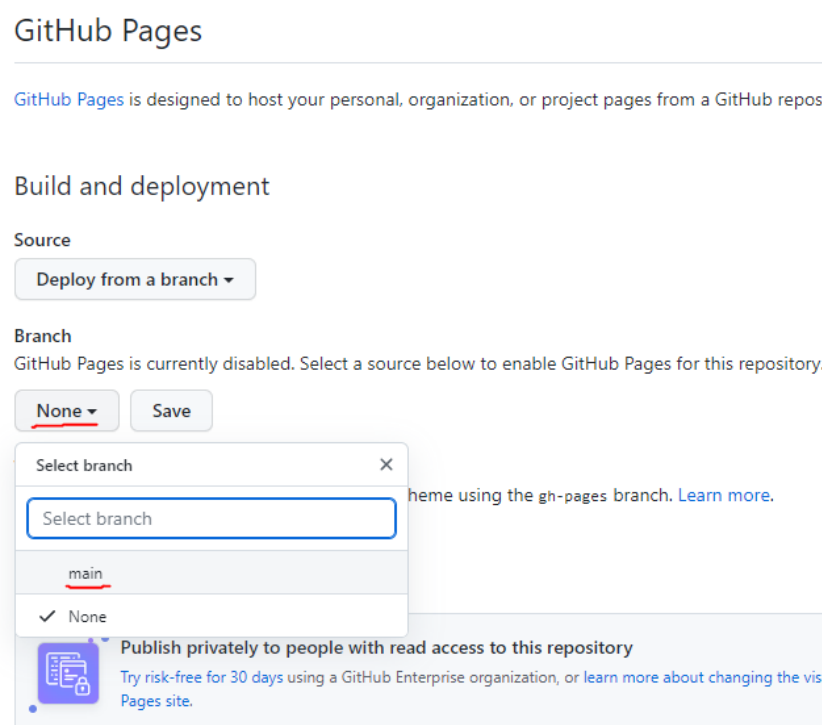


Figure 37. Select a Branch

Step 3: There will be either a green popup or a blue popup saying whether the site source has been saved or not.

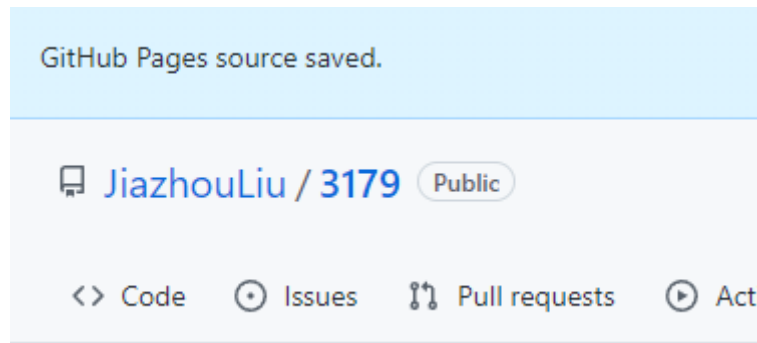


Figure 38. Blue popup – Site page source is saved

Step 4: **Wait for a couple of minutes**, then **refresh** the page; you should be able to see a notification like in Figure 39. Your repository is now public via this URL. You can click on the **Visit site button** to check your website. **If you don't have an index.html file at the base** of your repository, you will see your webpage like in figure 40.

GitHub Pages

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

Your site is live at <https://jiazhouliu.github.io/3179/>

Last deployed by github-pages 3 minutes ago

[Visit site](#) [...](#)

Build and deployment

Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more.](#)

main ▾ / (root) ▾ [Save](#)

Figure 39. Site URL

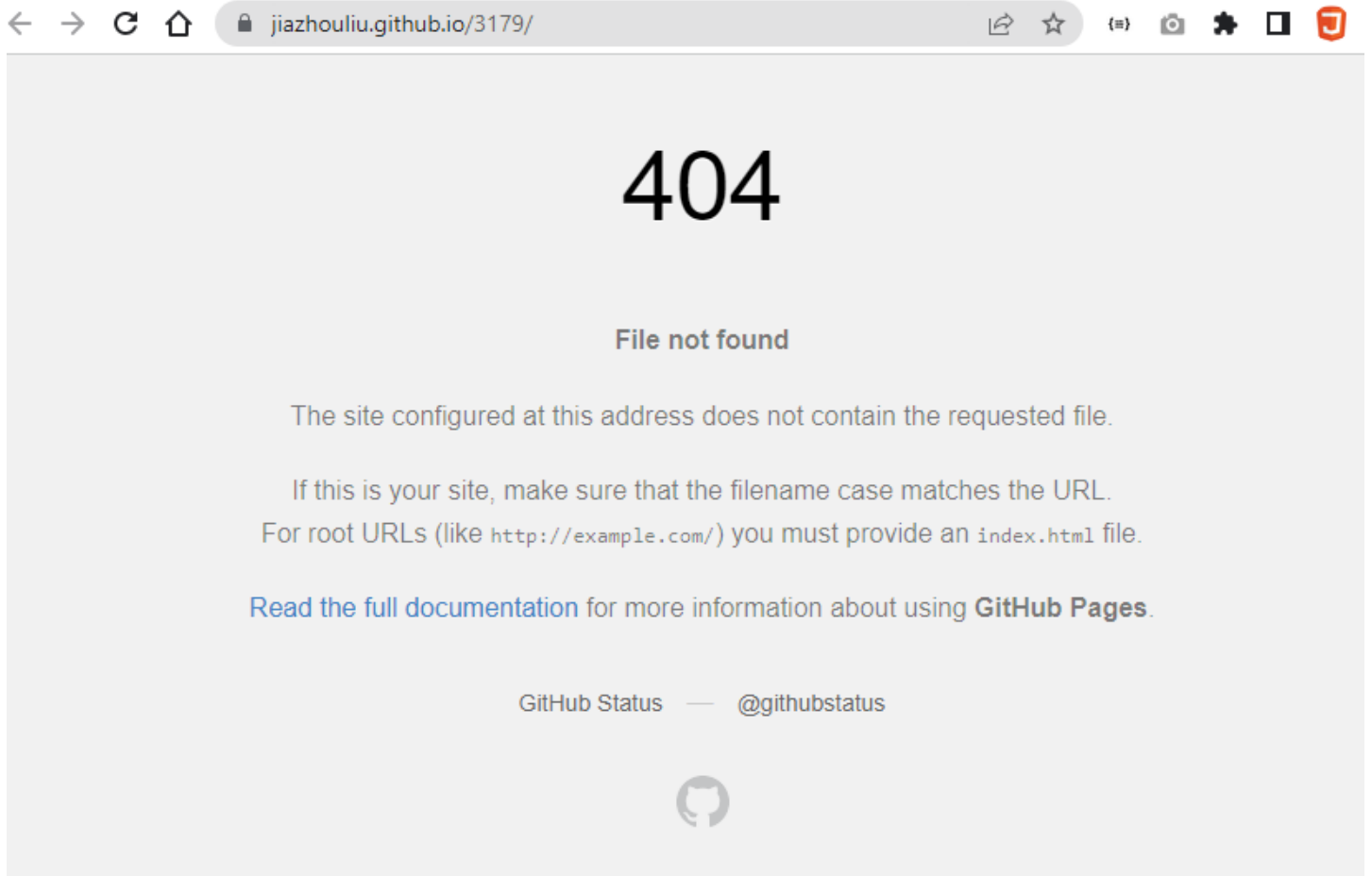


Figure 40. A website without an index.html

Exercise: Now, try to upload an index.html file and see if your website works correctly.

Note: You will have other homework and studio activities that need to be published on your GitHub. One way to manage them is to create/upload folders in the same repository (see figure 41). The other way is to create different repositories for each activity.

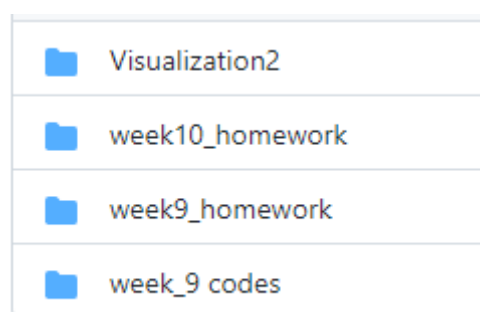


Figure 41. Different folders in the GitHub repository