BACKING UP FILES ON GITHUB AND HOSTING A WEB PAGE ON GITHUB PAGES

Edmond Lascaris and Anne Jessup - Creative Commons – 30 March 2021

OVERVIEW

Today we are going to learn how to create a repository in GitHub and back up some files from our Raspberry into this same repository. As a bonus, we can also configure GitHub Pages so that we can make any web page live on the internet.

LEARNING OBJECTIVES

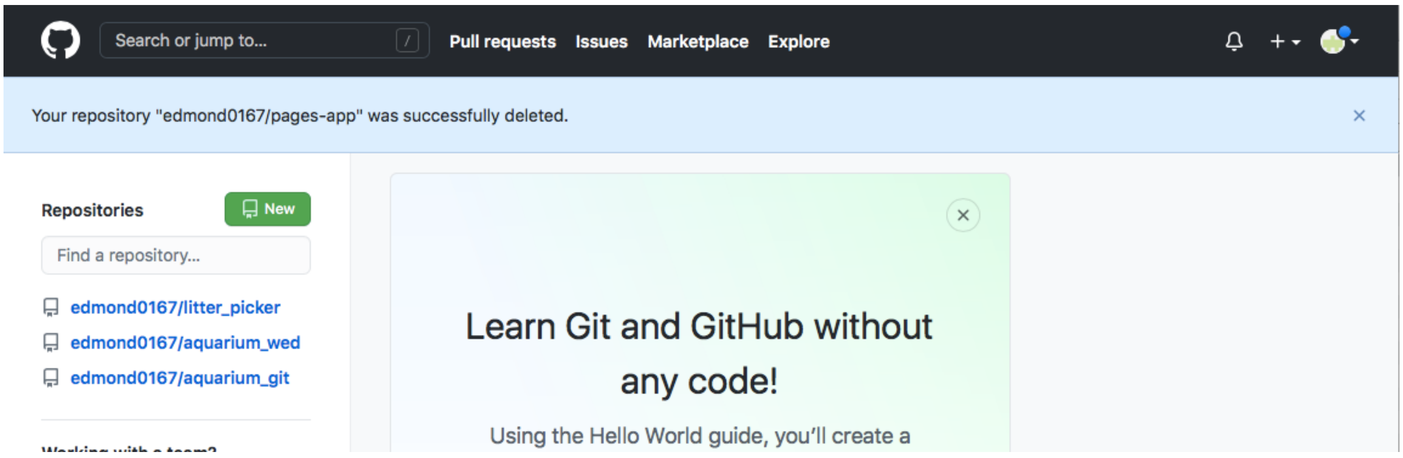
- How to create a GitHub repository
- How to clone a GitHub repository to the Raspberry Pi using the Terminal
- How to push new files from the Raspberry Pi up to the GitHub repository
- How to enable GitHub Pages

HOW TO CREATE A GITHUB REPOSITORY

A GitHub repository is like a directory or folder on your Raspberry Pi, except that it is in the Cloud. Most developers use GitHub to back up their code and share their code. If the data on your Raspberry Pi is ever lost, you can have peace of mind knowing that all your important files are backed up.

1. **Create a GitHub Repository**

- Open GitHub on your home computer or Raspberry Pi and **Sign in**.
- Click on the New or Create a New Repository button.



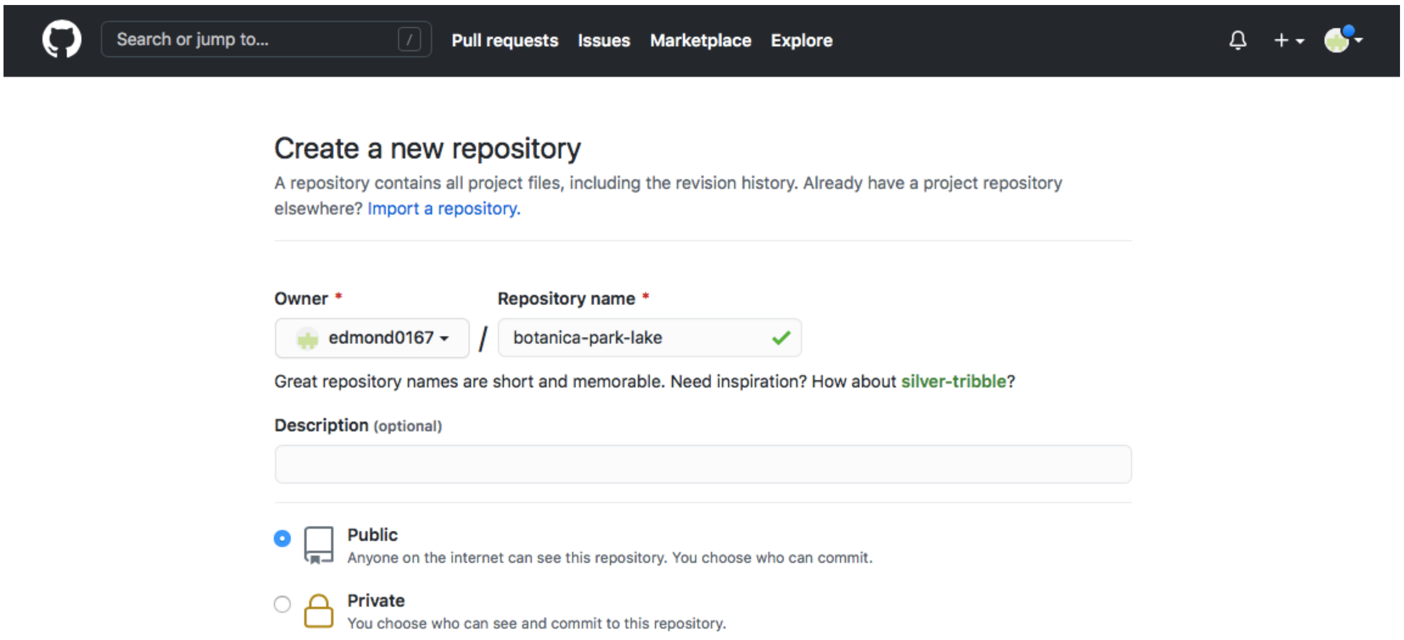
Your repository "edmond0167/pages-app" was successfully deleted.

edmond0167/litter_picker
edmond0167/aquarium_wed
edmond0167/aquarium_git

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a

- When filling out the details to Create a new repository enter the following:
  - **Repository name** (e.g. botanica-park-lake)
  - **Public** – make sure this radio button is selected
  - **Add a README file** – this needs to be selected. It creates a README.md file



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * edmond0167 / Repository name * botanica-park-lake ✓

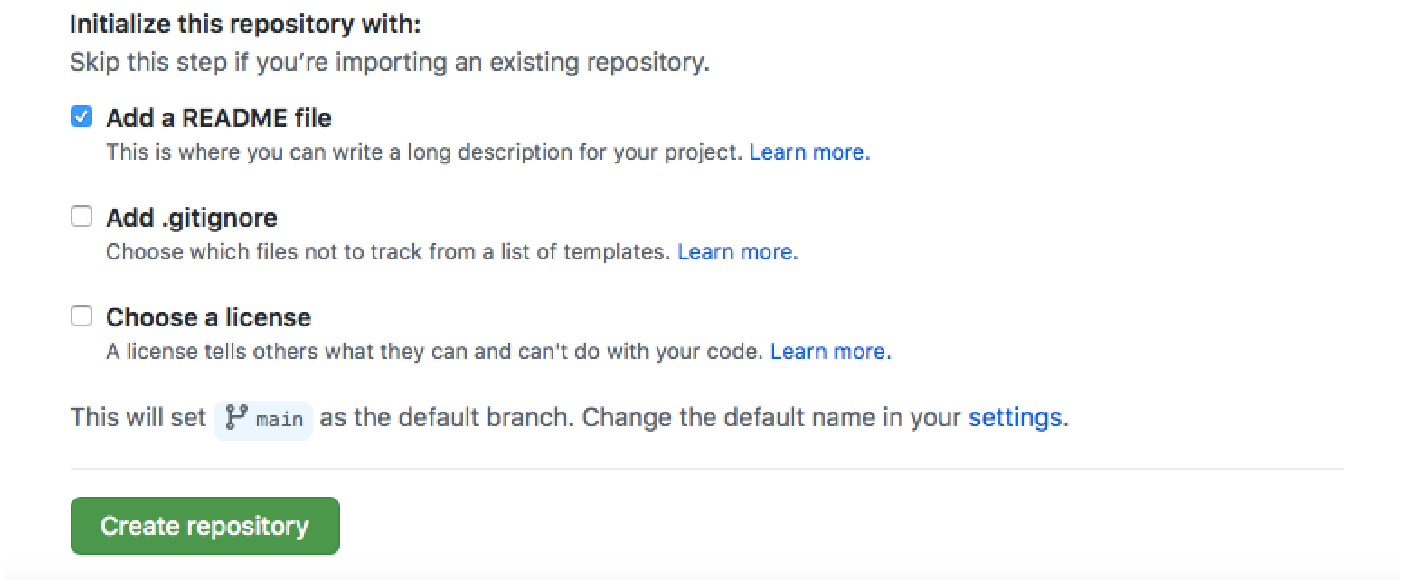
Great repository names are short and memorable. Need inspiration? How about silver-tribble?

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.

- Finally, click on the green **Create repository** button



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

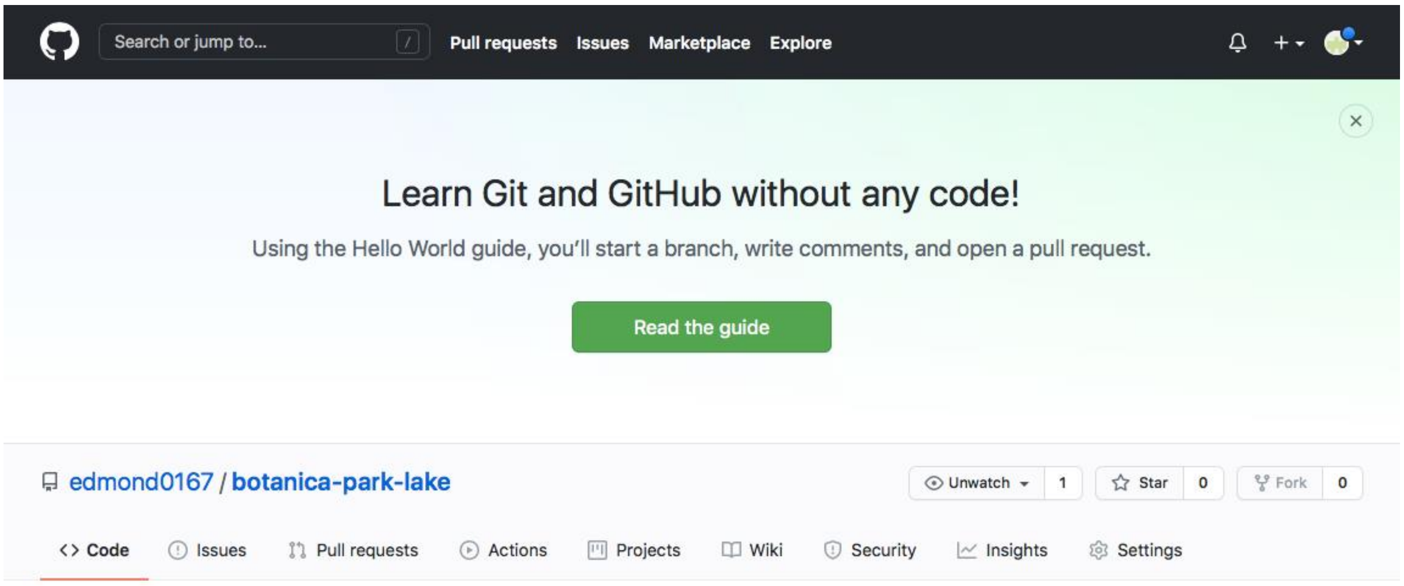
☑ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

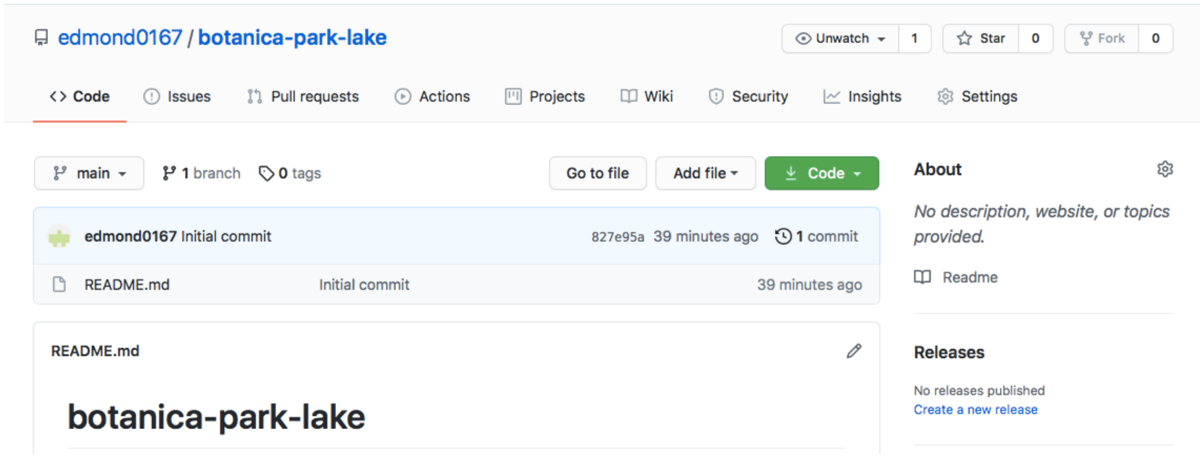
☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

This will set `main` as the default branch. Change the default name in your settings.

- The result will be a new repository.
- The new repository name is presented next to your username (**edmond0167/botanica-park-lake**).
- In this example,
  - my username is **edmond0167**, and
  - the repository name is **botanica-park-lake**


Search or jump to...
Pull requests
Issues
Marketplace
Explore
Learn Git and GitHub without any code!
Using the Hello World guide, you'll start a branch, write comments, and open a pull request.
Read the guide
edmond0167 / botanica-park-lake
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings

- Clicking on the repository name will show you what files are currently in the repository.
- In the example below there is only one file in the repository – README.md



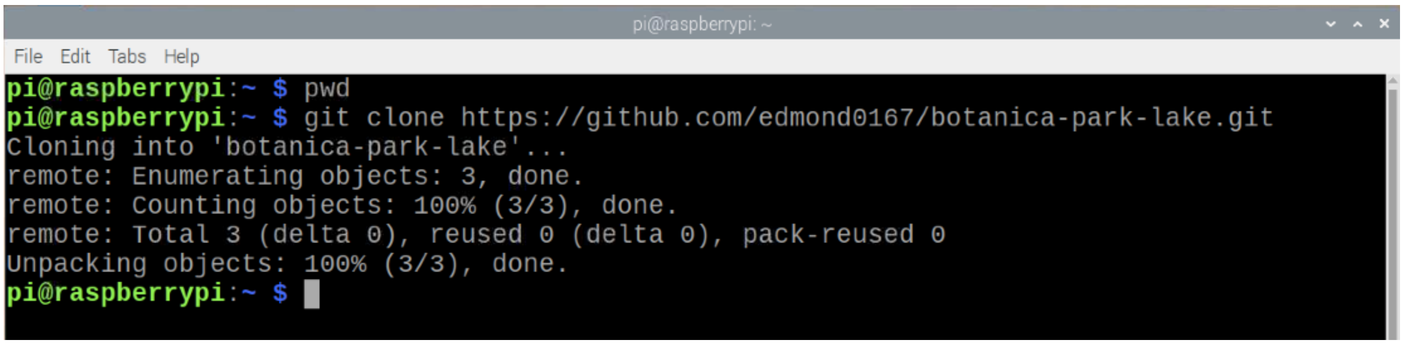
CLONING THE GITHUB REPOSITORY TO THE RASPBERRY PI

Now that we have created a new repository on GitHub we need to create a clone of this repository on our Raspberry Pi (local file system).

Once we have a clone of the repository on our Raspberry Pi we can back up any files to GitHub more easily.

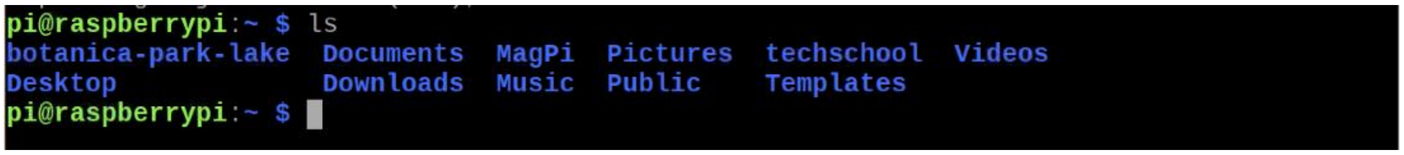
2. Cloning a repository from GitHub

- On the Raspberry Pi, open the Terminal.
- Make sure you are in the pi home directory (enter **pwd** and you should get **/home/pi** which is the same as **~**)
- To clone the repository, enter the command
    - **git clone https://github.com/<*github-username*>/<*repository-name*>.git**
- For my repository the command I would enter is:
    - **git clone https://github.com/edmond0167/botanica-park-lake.git**



```
pi@raspberrypi:~ $ pwd
pi@raspberrypi:~ $ git clone https://github.com/edmond0167/botanica-park-lake.git
Cloning into 'botanica-park-lake'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
pi@raspberrypi:~ $
```

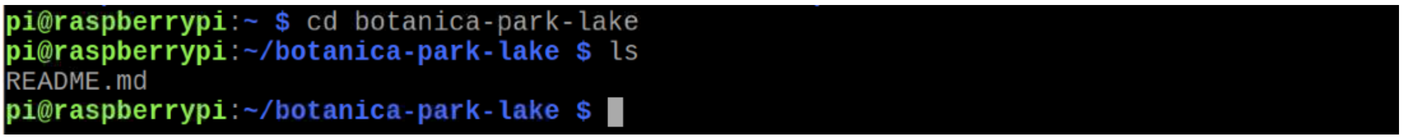
- If you enter the command **ls** in the Terminal you should see your repository directory
- In my case I can see the directory **botanica-park-lake**



```
pi@raspberrypi:~ $ ls
botanica-park-lake  Documents  MagPi  Pictures  techschool  Videos
Desktop             Downloads  Music  Public    Templates
pi@raspberrypi:~ $
```

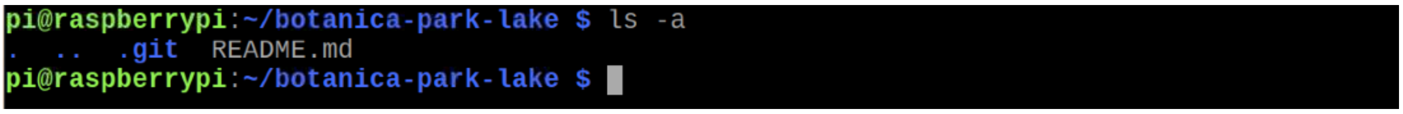
- You can enter the directory with the command **cd <directory-name>**
- In this example I would enter **cd botanica-park-lake**
- Once inside the directory you can enter the command **ls** to examine the files.

- My botanica-park-lake only has one file – **README.md**



```
pi@raspberrypi:~ $ cd botanica-park-lake
pi@raspberrypi:~/botanica-park-lake $ ls
README.md
pi@raspberrypi:~/botanica-park-lake $
```

- However, there are actually some hidden files and directories within the botanica-park-lake directory.
- To reveal these hidden files we add the -a option to the ls command. Enter **ls -a**
- This reveals another hidden directory named **.git** which holds configuration files and other information to help the Raspberry Pi back up files to the GitHub repository on the internet.



```
pi@raspberrypi:~/botanica-park-lake $ ls -a
.  ..  .git  README.md
pi@raspberrypi:~/botanica-park-lake $
```

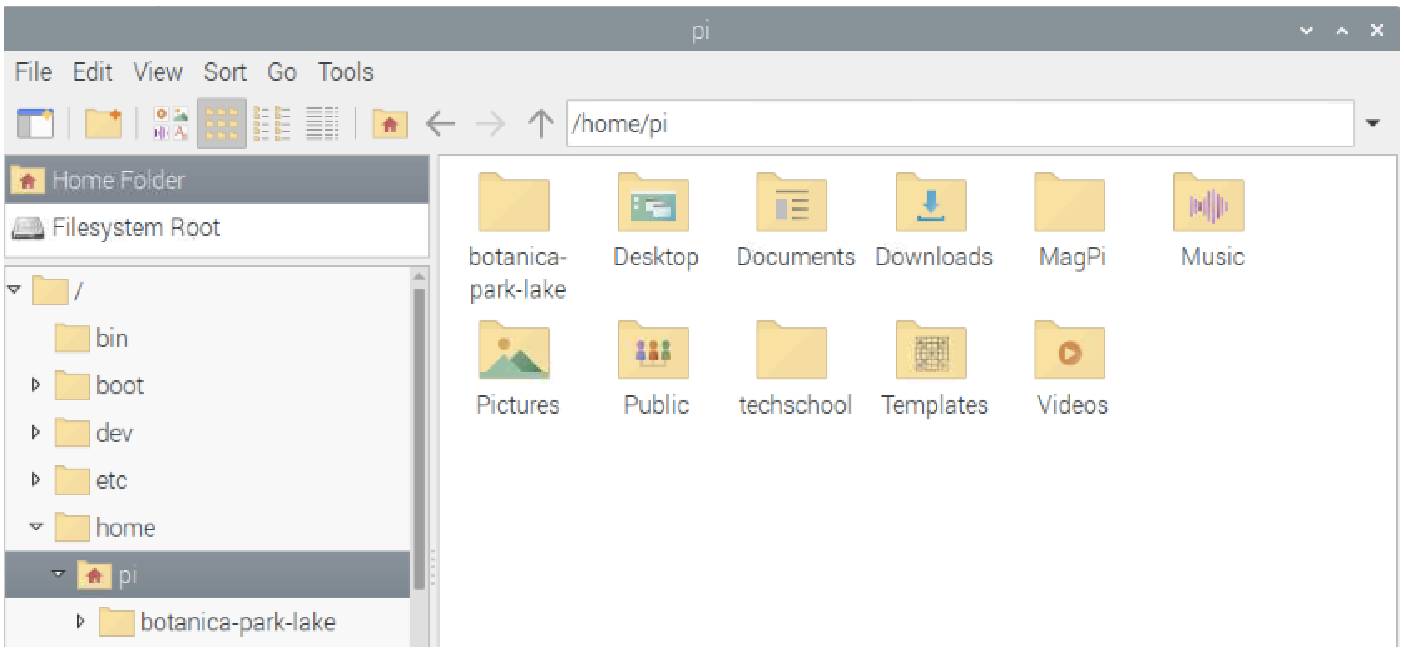
PUSHING FILES FROM THE PI TO THE GITHUB REPOSITORY

GitHub allows us to easily upload (Push) files from the working directory on the Pi to our GitHub account. We do this so that our files in GitHub are always updated with the latest changes in the working directory on the Raspberry Pi. We will do this in two steps:

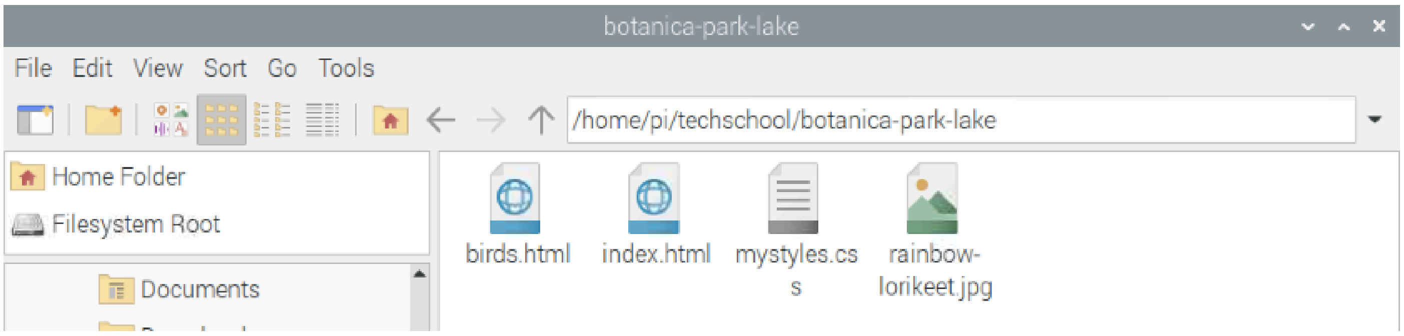
- Add some files to the working directory (/home/pi/botanica-park-lake) on the Raspberry PI
- Push these new files to the GitHub repository

1. Copy files to the new botanica-park-lake directory on the Raspberry Pi

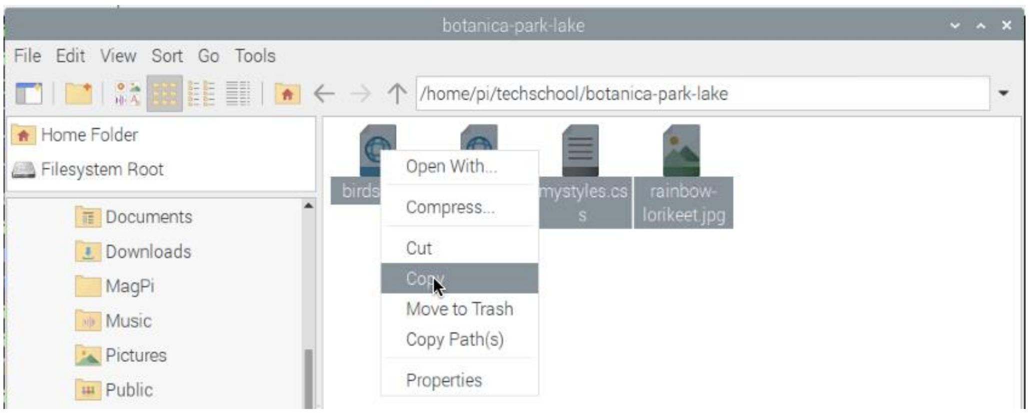
- We will copy some files from the Raspberry Pi
    - from - **/home/pi/techschool/botanica-park-lake**
    - to **/home/pi/botanica-park-lake**
- so that we can upload these to our GitHub repository.
- Open up the File Manager. The File Manager normally opens at the **/home/pi** directory.



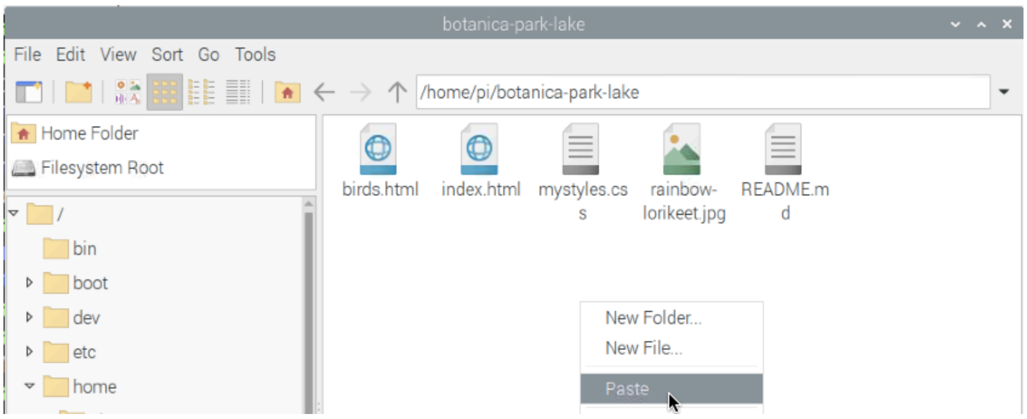
- Navigate to the files we created in a previous lesson in the directory:
    - **/home/pi/techschool/botanica-park-lake**



- Highlight all the files, right-mouse-button click and select Copy.



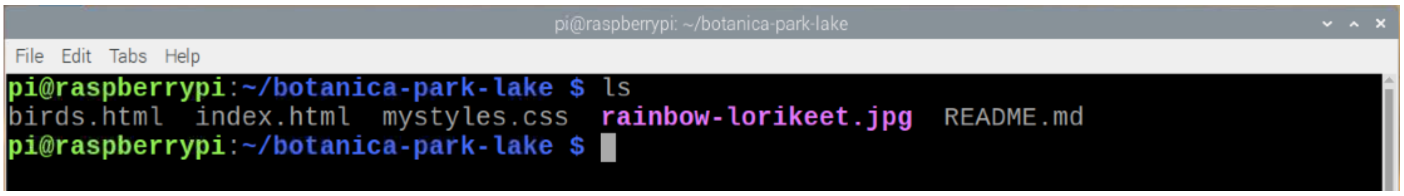
- Navigate to the new repository directory **/home/pi/botanica-park-lake**
- **Paste** the copied files.



2. Pushing files from the Raspberry Pi working directory to the GitHub repository

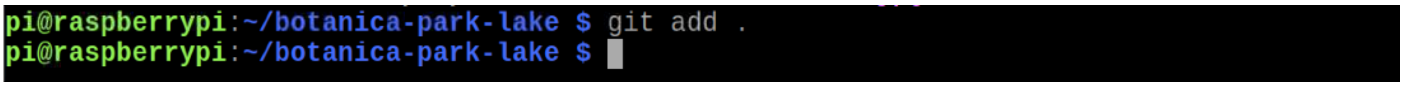
- Open the Terminal on the Raspberry Pi.

- Make sure you are in the local working directory e.g. **~/botanica-park-lake**
- Check that new files have been added to this working directory using the **ls** command.



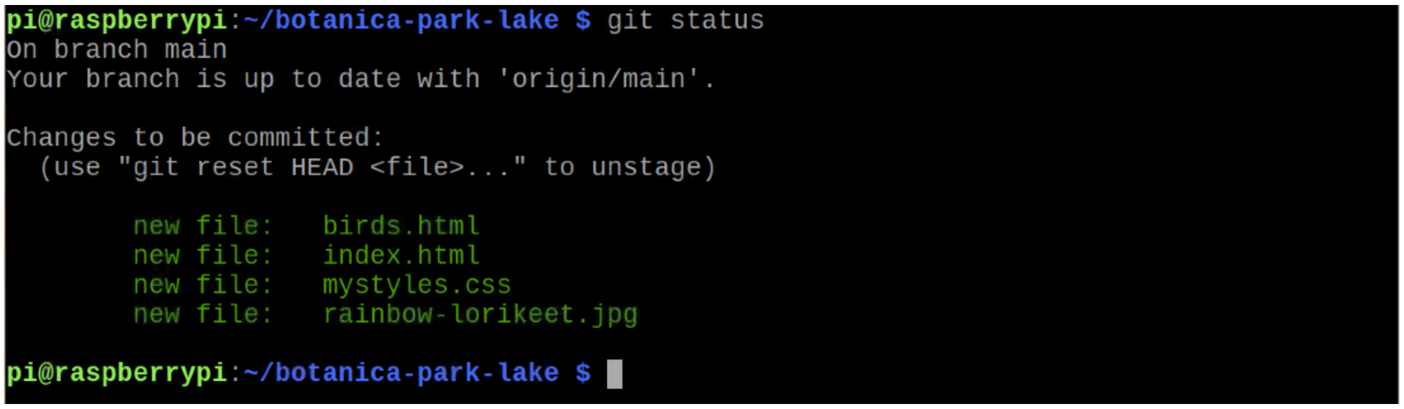
```
pi@raspberrypi:~/botanica-park-lake $ ls
birds.html  index.html  mystyles.css  rainbow-lorikeet.jpg  README.md
pi@raspberrypi:~/botanica-park-lake $
```

- Enter the command **git add .** (yes – the full stop is very important) THsi command adds new files from the (1)-**Working Directory** to the (2)-**Staging Area**.



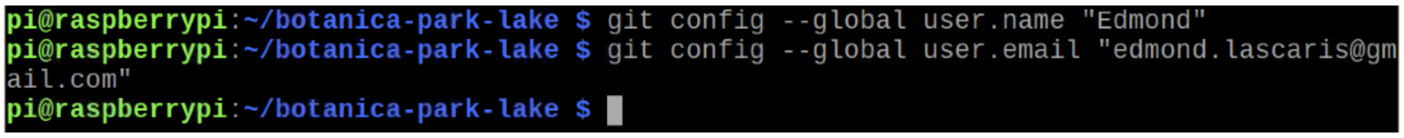
```
pi@raspberrypi:~/botanica-park-lake $ git add .
pi@raspberrypi:~/botanica-park-lake $
```

- You can enter the command **git status** to understand what new files are being added to your repository.



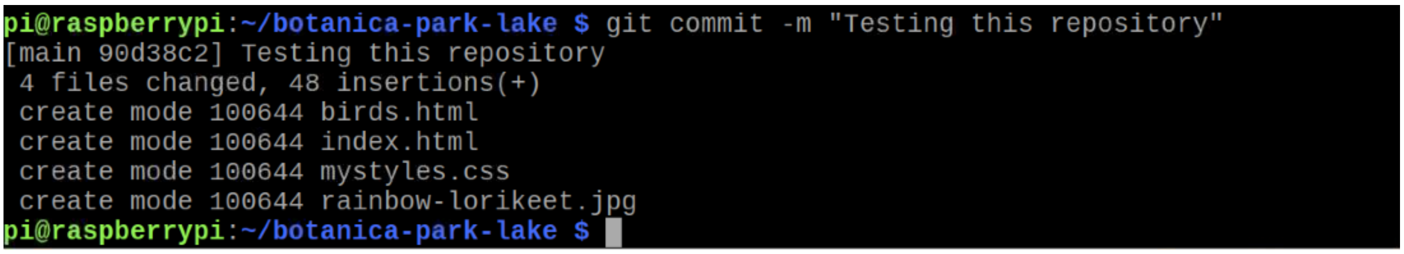
```
pi@raspberrypi:~/botanica-park-lake $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   birds.html
        new file:   index.html
        new file:   mystyles.css
        new file:   rainbow-lorikeet.jpg

pi@raspberrypi:~/botanica-park-lake $
```

- We need to configure our username and email details. Enter the following commands:
  - **git config --global user.name "Edmond"**
  - **git config --global user.email** edmond.lascaris@gmail.com



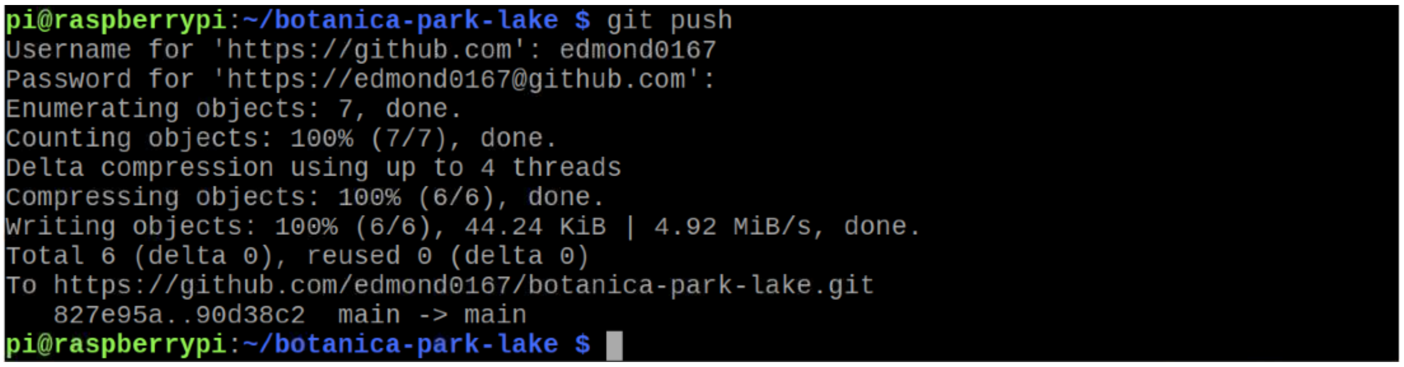
```
pi@raspberrypi:~/botanica-park-lake $ git config --global user.name "Edmond"
pi@raspberrypi:~/botanica-park-lake $ git config --global user.email "edmond.lascaris@gm
ail.com"
pi@raspberrypi:~/botanica-park-lake $
```

- Now we enter the command **git commit -m "Testing the repository"** – This command takes the files in the (2)-**Staging Area** and permanently commits them to the (3)-**Local Git Repository** – which is in the **.git** folder on the Raspberry Pi.



```
pi@raspberrypi:~/botanica-park-lake $ git commit -m "Testing this repository"
[main 90d38c2] Testing this repository
 4 files changed, 48 insertions(+)
 create mode 100644 birds.html
 create mode 100644 index.html
 create mode 100644 mystyles.css
 create mode 100644 rainbow-lorikeet.jpg
pi@raspberrypi:~/botanica-park-lake $
```

- Finally we can enter the command **git push** – this pushes any code updates from the (3)-**Local Repository** to the (4)-**Remote Repository** on GitHub.
- When this command is entered, you will also be asked for your **username** and **password**.
- When entering in your password no characters will be displayed in the Terminal.

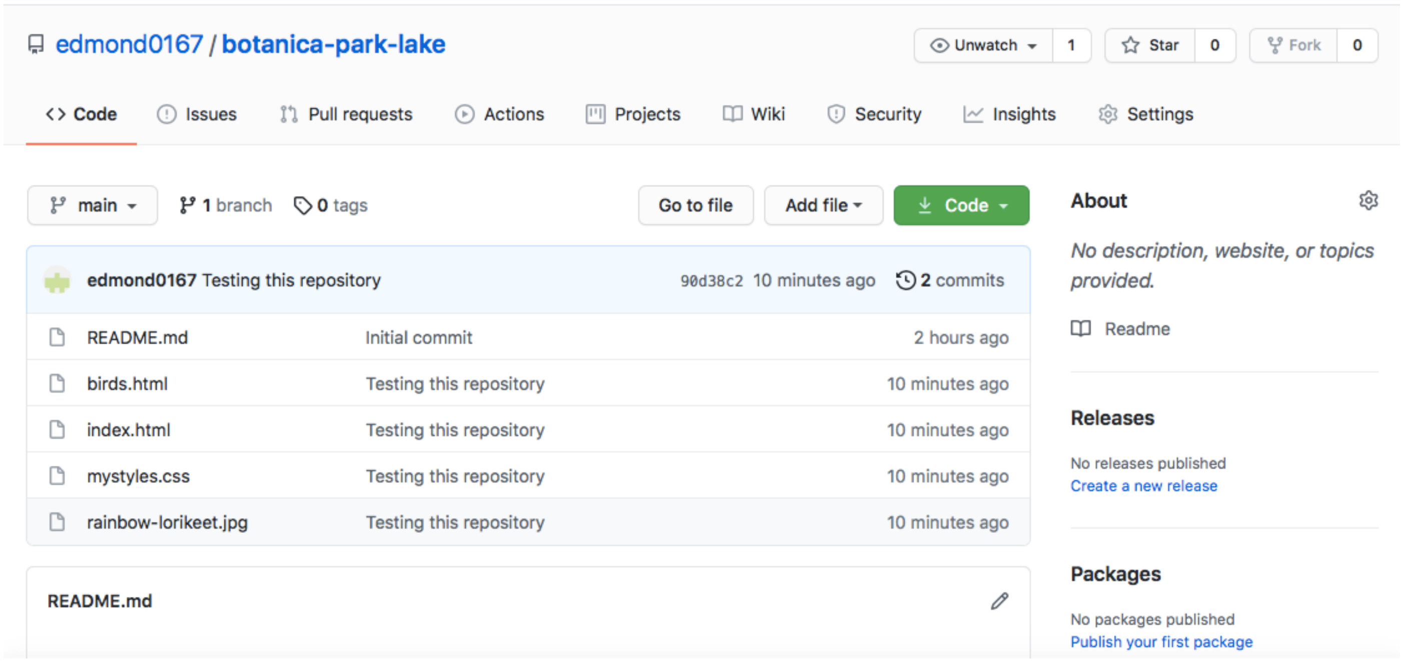


```
pi@raspberrypi:~/botanica-park-lake $ git push
Username for 'https://github.com': edmond0167
Password for 'https://edmond0167@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 44.24 KiB | 4.92 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/edmond0167/botanica-park-lake.git
   827e95a..90d38c2  main -> main
pi@raspberrypi:~/botanica-park-lake $
```

- Any future updates to this working directory will only require the following three commands:
  - **git add .** *- to copy files from Working Directory to the Staging Area*
  - **git commit -m "new comments"** *- commit files from the Staging Area to the Local Repository*
  - **git push** (with a request for username and password) *- push files to the Remote Repository*
- NB. If you have Two-Factor Authentication enabled, you will need to disable this in GitHub Settings.

3. Checking the GitHub repository to see if files have been uploaded

- Go to your GitHub account and the new remote repository.
- Click on the **repository name** – in my case I would click on /edmond0167/botanica-park-lake
- You should see all the files from the Raspberry Pi working directory in the GitHub remote repository.



edmond0167 / botanica-park-lake

Unwatch 1 | Star 0 | Fork 0

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

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

edmond0167 Testing this repository 90d38c2 10 minutes ago 2 commits

README.md	Initial commit	2 hours ago
birds.html	Testing this repository	10 minutes ago
index.html	Testing this repository	10 minutes ago
mystyles.css	Testing this repository	10 minutes ago
rainbow-lorikeet.jpg	Testing this repository	10 minutes ago

README.md

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

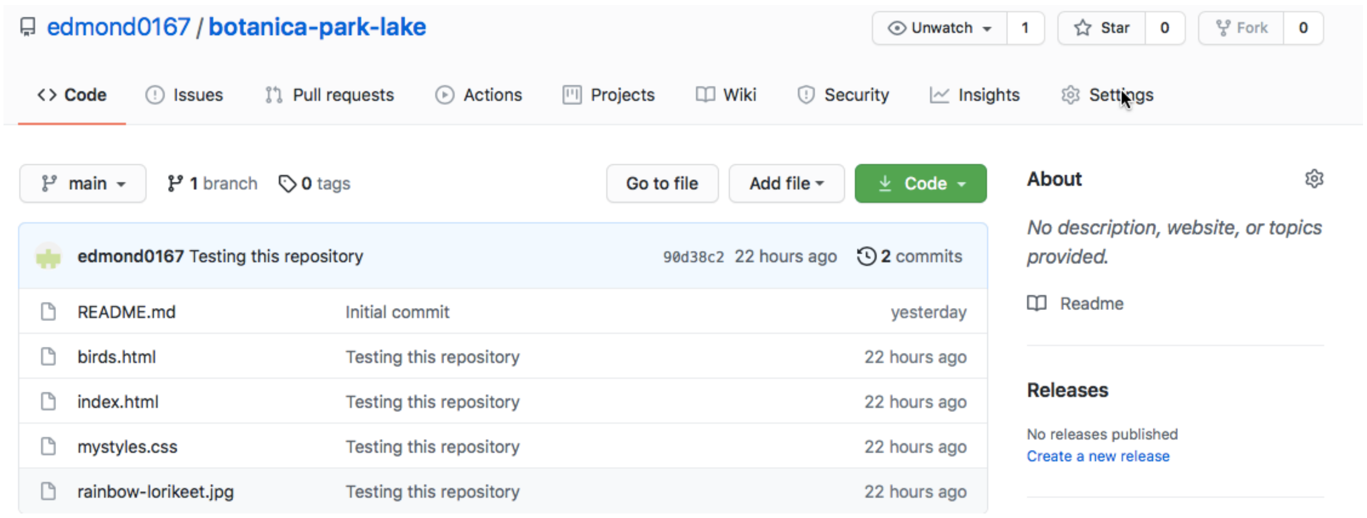
No packages published
Publish your first package

ENABLING GITHUB PAGES

GitHub provides an additional feature that when enabled allows GitHub to act as a web server. All you need to have in place is an index.html file. GitHub will look for this file and deliver it as the default welcome page. GitHub pages lets you turn GitHub repositories into simple web site so that you can present your project to the world.

4. Enabling GitHub Pages for a repository

- Click on the repository name and then click on the **Settings tab** below the repository name.



edmond0167 Testing this repository 90d38c2 22 hours ago 2 commits

README.md	Initial commit	yesterday
birds.html	Testing this repository	22 hours ago
index.html	Testing this repository	22 hours ago
mystyles.css	Testing this repository	22 hours ago
rainbow-lorikeet.jpg	Testing this repository	22 hours ago

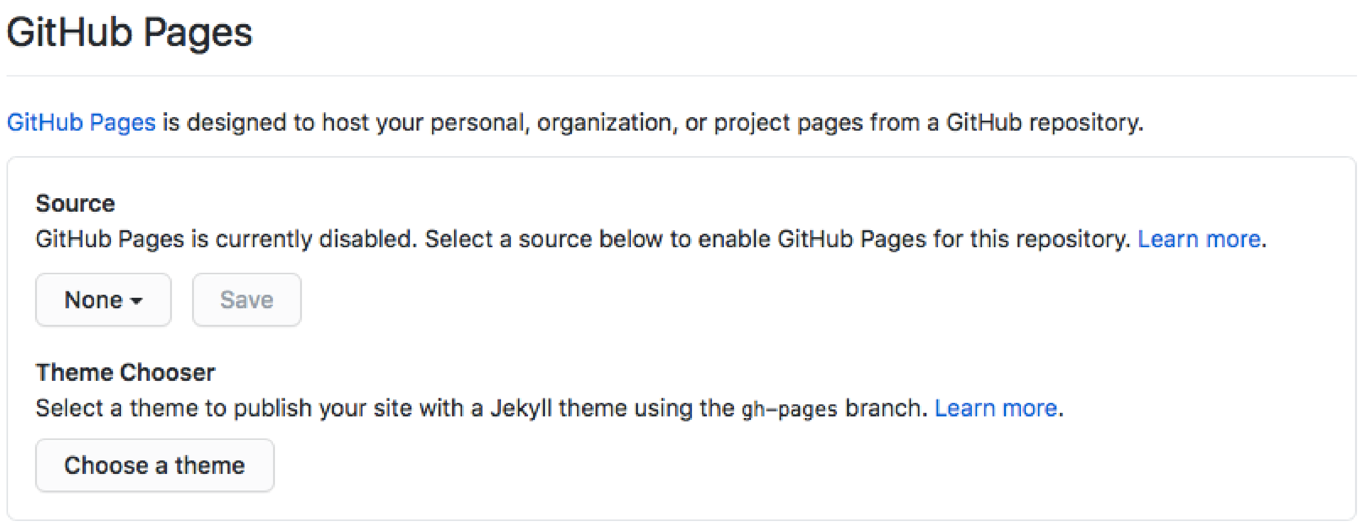
No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

- Once in the Setting tab, scroll down until you get to the sub-title **GitHub Pages**



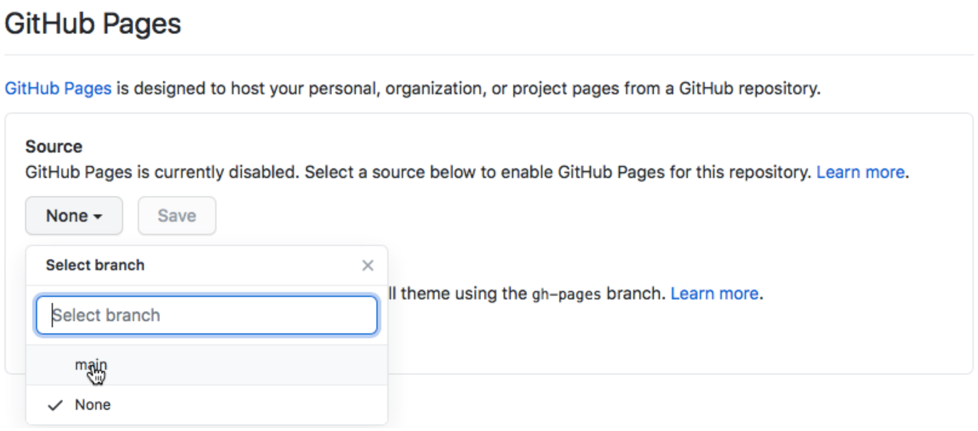
GitHub Pages

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.

- Click on the button currently labelled **None** and select the **main** option.

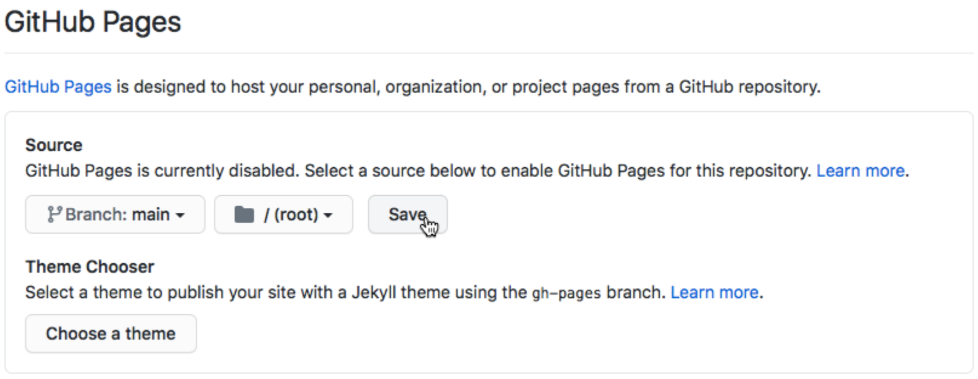

GitHub Pages

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.

- Once **main** is selected, click on **Save**.
- When you click Save you will be returned to the top of the Settings page.


GitHub Pages

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

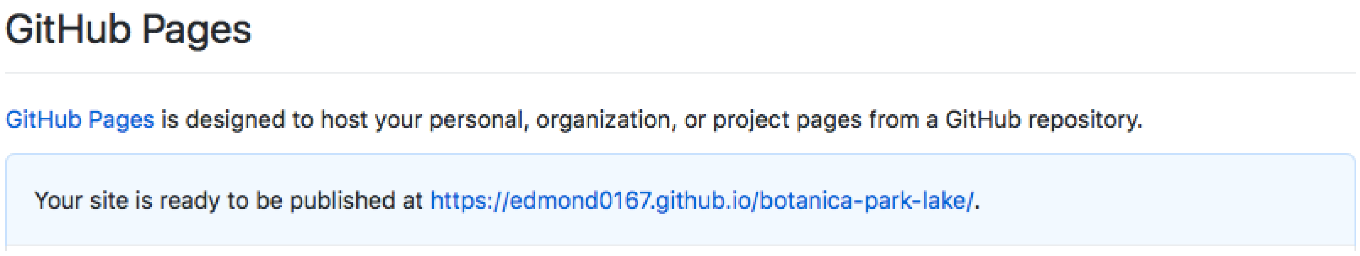
Source

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.

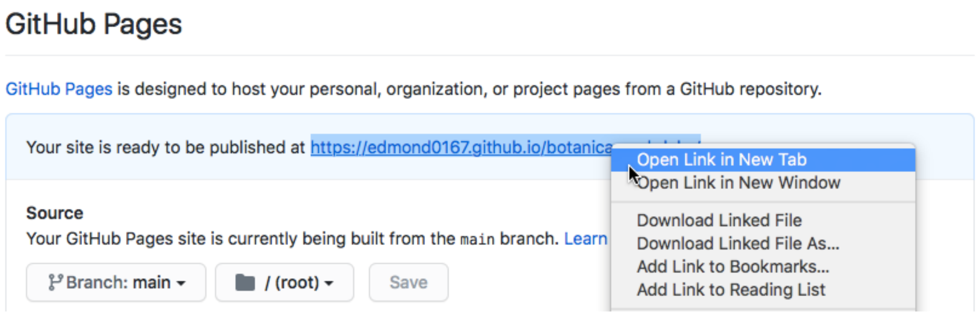
- Scroll down the Setting page to the GitHub Pages section and you should see a notice that **Your site is ready to be pub-lished a https://<user-name>.github.io/<repository-name>**


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://edmond0167.github.io/botanica-park-lake/.

- Right mouse button click on this URL and open up the link in a new Browser tab.


GitHub Pages

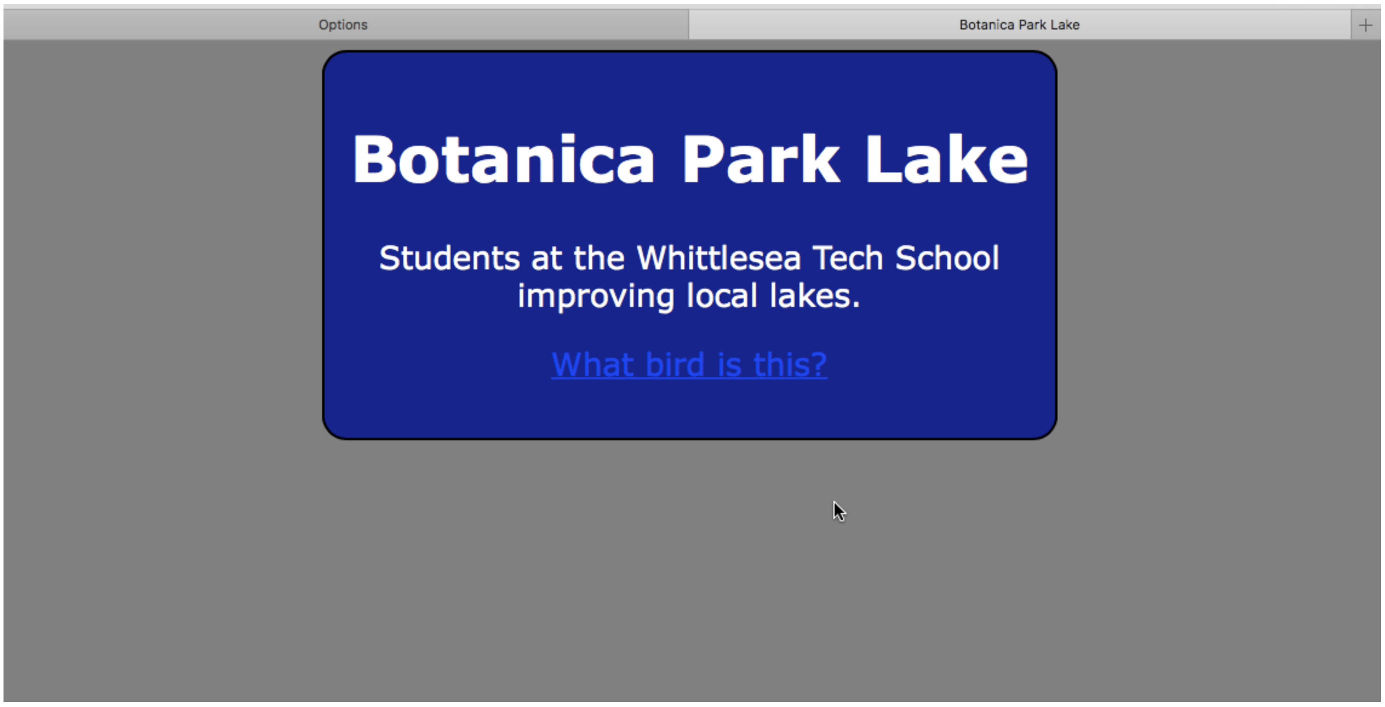
GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://edmond0167.github.io/botanica

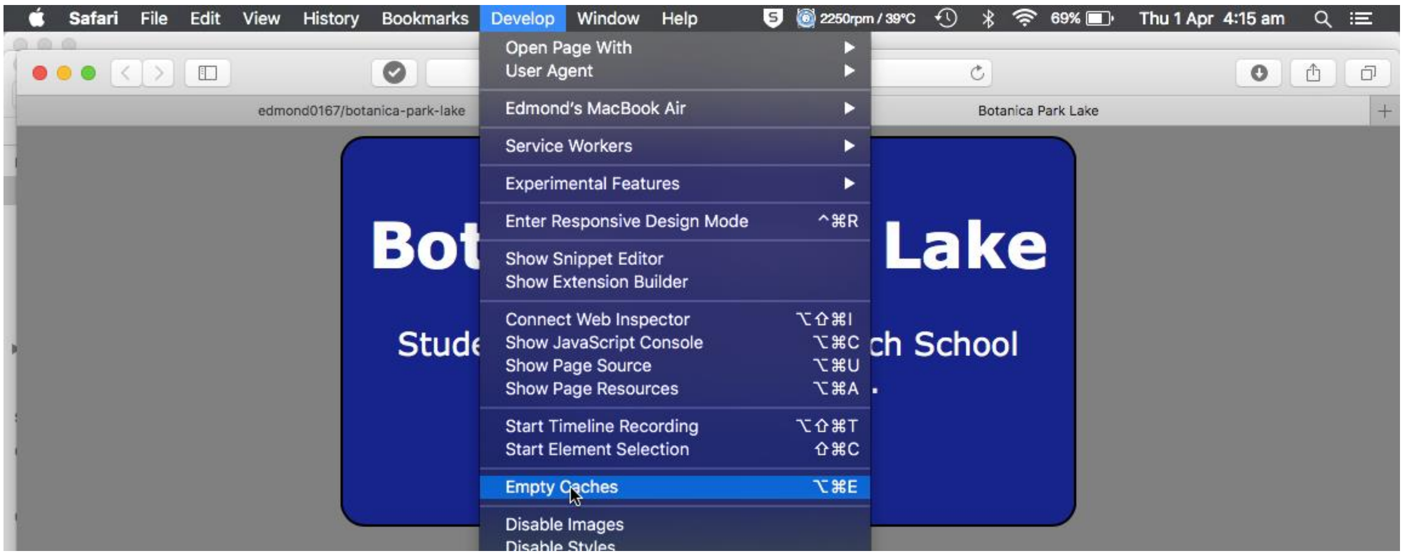
Source

Your GitHub Pages site is currently being built from the `main` branch. Learn

- If everything works then you should see your index.html file being displayed in your Browser.
- You can test that this URL is available on the internet by accessing it from another computer or a mobile phone.
- Changes may not be seen immediately. You may need to wait 1-5 minutes for changes to take effect.

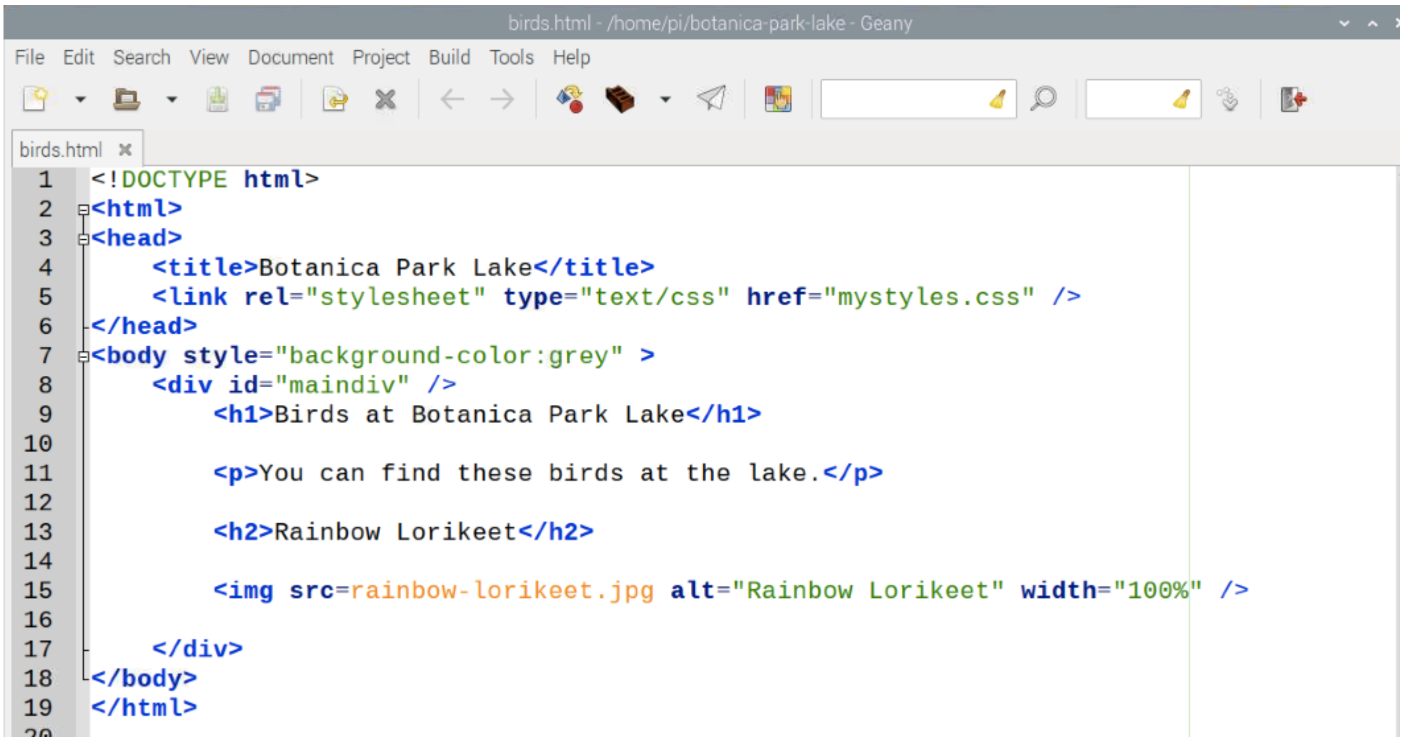

Options
Botanica Park Lake
Botanica Park Lake
Students at the Whittlesea Tech School improving local lakes.
What bird is this?

- If you still don't see a change you may need to **empty the cache** on your Browser.
- Browsers cache (or store) previously accessed web pages to speed up page display response times. Often a Browser will retrieve a saved copy of a web page and not reload the newer version. Clearing the Browser cache forces the Brower to reload the page.
- Click on **Empty Caches** to force the web page to reload. Each Browser has this option. You may need to enable this function before you can use it. You will also need to **Reload** the page from within the Browser.



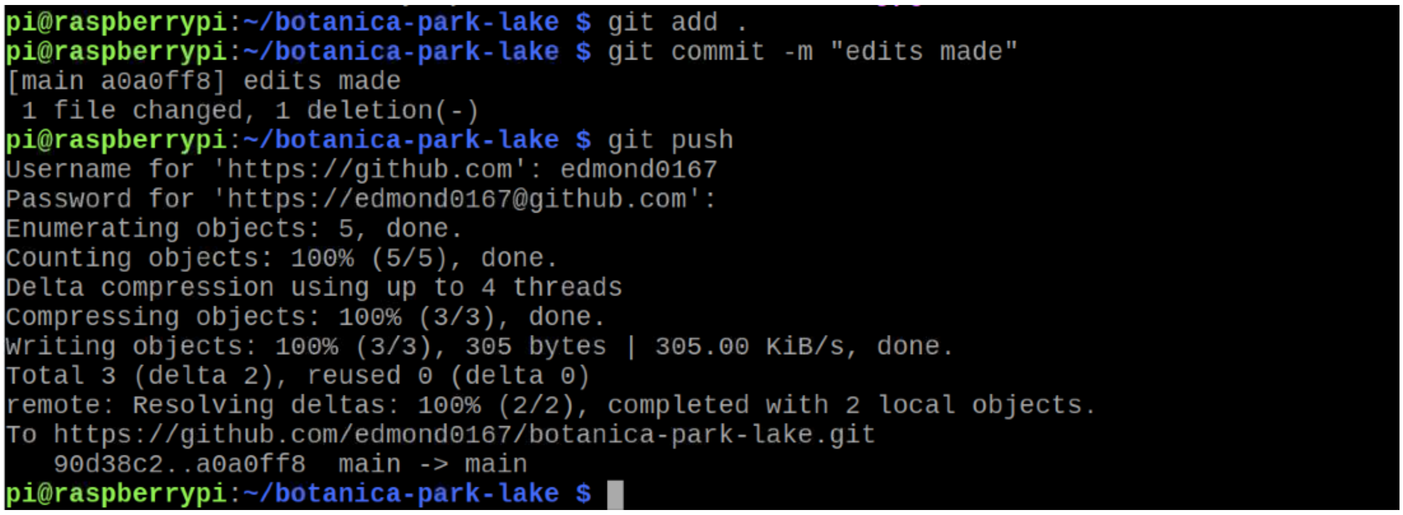
5. **Updating you web page and uploading to GitHub**

- Go back to your Raspberry Pi and open Geany.
- Make any changes to your files using Geany. You can even create a new web page (e.g. waterbugs.html) and save this in the same directory as your other files.


birds.html - /home/pi/botanica-park-lake - Geany
File Edit Search View Document Project Build Tools Help
birds.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Botanica Park Lake</title>
    <link rel="stylesheet" type="text/css" href="mystyles.css" />
</head>
<body style="background-color:grey" >
    <div id="maindiv" />
        <h1>Birds at Botanica Park Lake</h1>

        <p>You can find these birds at the lake.</p>

        <h2>Rainbow Lorikeet</h2>

        <img src=rainbow-lorikeet.jpg alt="Rainbow Lorikeet" width="100%" />

    </div>
</body>
</html>
```

- Once you have finished making changes or additions you need to upload (push) your changes to the GitHub repository.
- Enter the following commands in the Terminal in this sequence.
    - **git add .**
    - **git commit -m "messages and comments here"**
    - **git push**
        - enter **user-name**
        - enter **password**
- Don't forget that you may need to **Empty the Cache** on your Browser and **Reload** the page for all changes to be seen.



```
pi@raspberrypi:~/botanica-park-lake $ git add .
pi@raspberrypi:~/botanica-park-lake $ git commit -m "edits made"
[main a0a0ff8] edits made
 1 file changed, 1 deletion(-)
pi@raspberrypi:~/botanica-park-lake $ git push
Username for 'https://github.com': edmond0167
Password for 'https://edmond0167@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/edmond0167/botanica-park-lake.git
   90d38c2..a0a0ff8  main -> main
pi@raspberrypi:~/botanica-park-lake $
```