

jeKyRequirements:

- Github account (<https://github.com/>)
- Github desktop application installed on your computer (<https://desktop.github.com/>)
- Jekyll installed on your computer (<https://jekyllrb.com/docs/installation/>)
- A text editor (e.g. Atom, <https://atom.io/>, or Sublime Text, <https://www.sublimetext.com/>)

Step 1: Set up a Github repository for Github pages and clone your repository.

1. After logging into Github, under the repositories tab, click 'new' to setup a new repository.

2. IMPORTANT: You **must** name your repository 'yourgithubname.github.io'. That is the way that Github will be able to recognize that your repository is for Github pages. (You can see the example below- I'm only getting the error message because I already have this repository created)

3. Make sure to check the box "Initialize this repository with a README."

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 anteaterpeter ▾

Repository name *

/ anteaterpeter.github.io 

Great repository names are short, lowercase, and unique. The repository anteaterpeter.github.io already exists on this account. b-dollop.

Description (optional)

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

☒ Initialize this repository with a README

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

Add .gitignore: None ▾

Add a license: None ▾



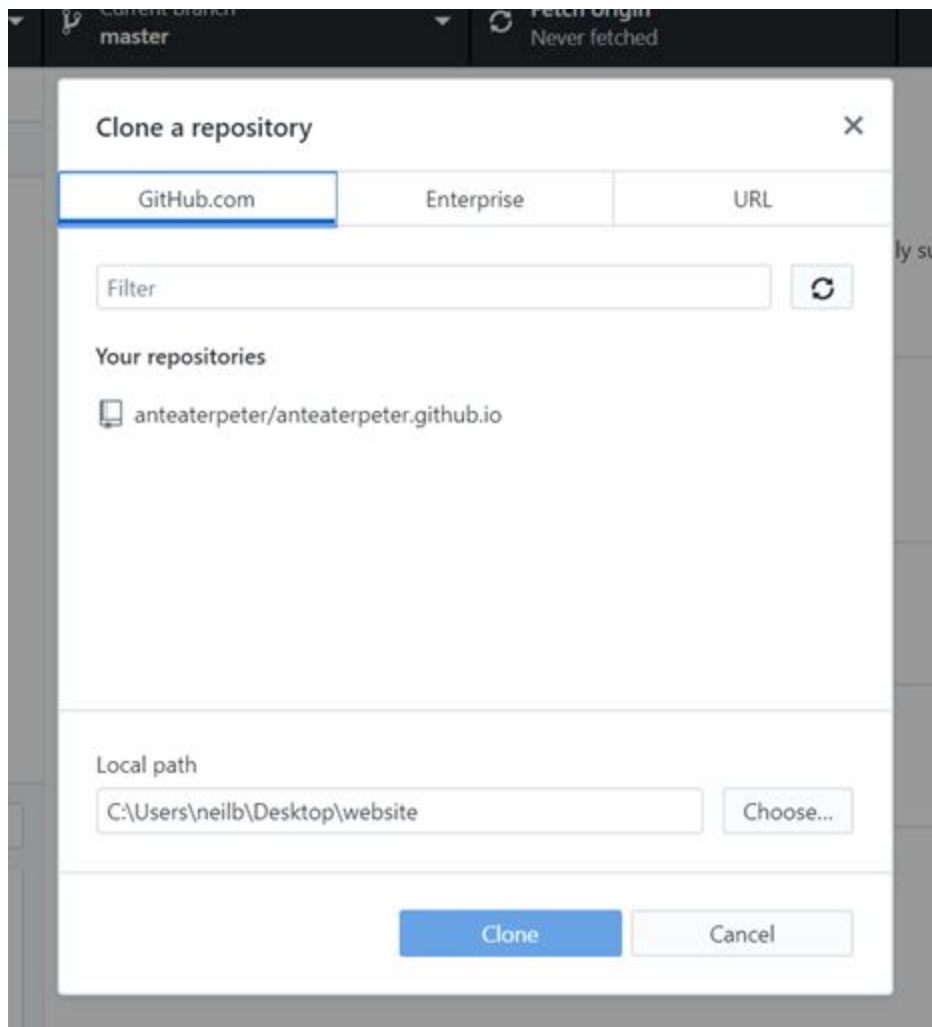
Create repository

3. After your repository is created, head over to Github Desktop to clone your repository into a directory on your computer.

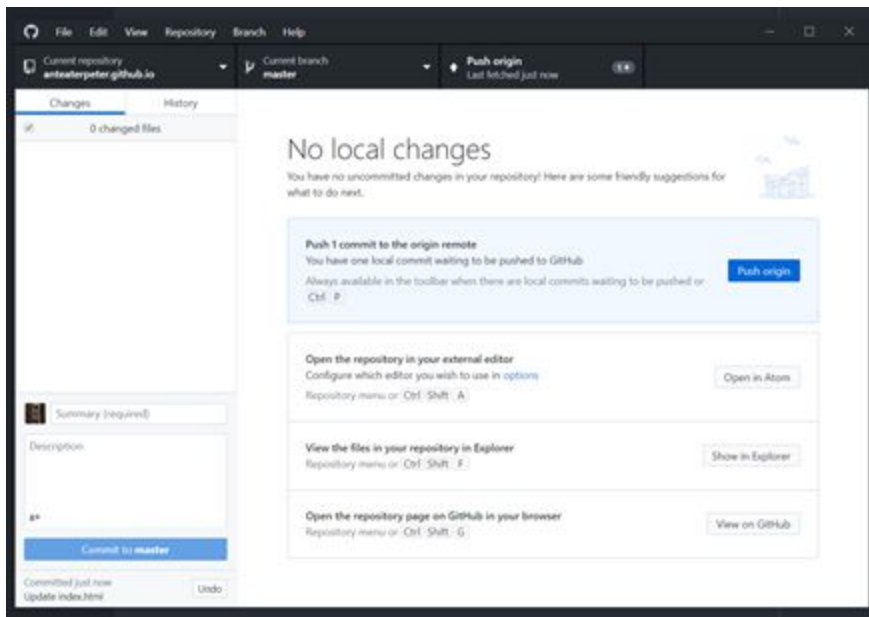
a. Click File>Clone Repository...

b. Then Select 'Browse' to choose the directory that you want your cloned repository to go into (we chose to clone our repository into a new folder called 'website')

c. Finally, select the repository corresponding to your future website and click 'clone'



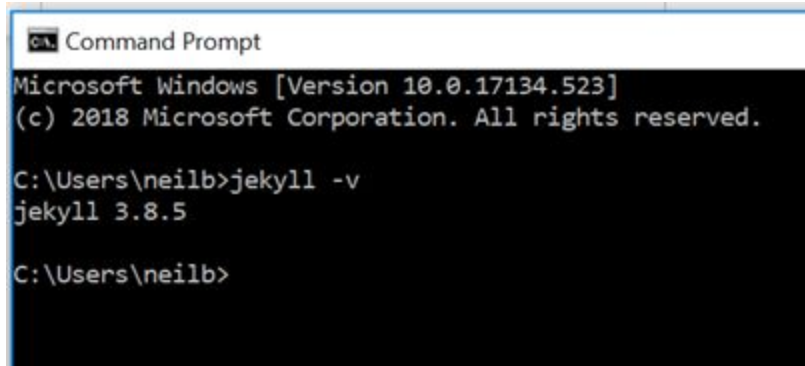
4. After that, open your text editor and create an HTML file
 - a. Inside the file, write 'Hello World'
 - b. Save this file in your new repository as 'index.html' (for the file to be recognized as the frontpage of your website, it will always have to be named 'index')
 - c. Commit your changes in Github Desktop and Push your changes (make sure to give a name and description to your commit)



If all goes well, after a few minutes, you should be able to go to 'USERNAME.github.io' in your browser and see the html file you have created.

Step 2: Set up a site with Jekyll

1. Open your command prompt and first check that Jekyll is installed.
 - a. Type 'jekyll -v' and hit enter. If your command prompt returns the version of Jekyll that you have installed then you have Jekyll installed



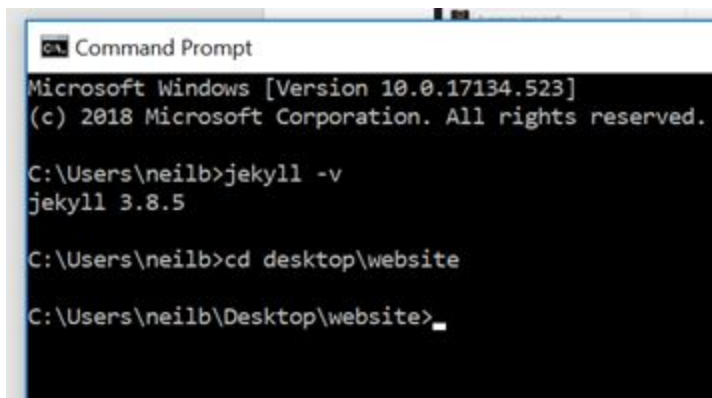
```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\neilb>jekyll -v
jekyll 3.8.5

C:\Users\neilb>
```

b. If your command prompt tells you that it doesn't recognize the command 'jekyll' then you need to follow the installation instructions here: <https://jekyllrb.com/docs/installation/>

2. Change your directory to the directory where your repository is using the command 'cd' in your command prompt (In my case, I'm not in the exact same folder but in the folder that my repository is also located)



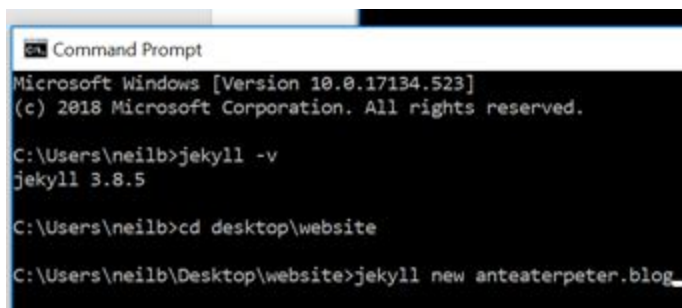
```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\neilb>jekyll -v
jekyll 3.8.5

C:\Users\neilb>cd desktop\website

C:\Users\neilb\Desktop\website>
```

3. Create your Jekyll site by giving the command 'jekyll new USERNAME.blog'



```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\neilb>jekyll -v
jekyll 3.8.5

C:\Users\neilb>cd desktop\website

C:\Users\neilb\Desktop\website>jekyll new anteaterpeter.blog
```

4. This will set up a group of files that you can use for your website. You can see these files in action by changing your directory into the new folder you have created and then giving the command 'bundle exec jekyll serve'

```
Command Prompt - bundle exec jekyll serve
Bundler: Using liquid 4.0.1
Bundler: Using mercenary 0.3.6
Bundler: Using pathutil 0.16.2
Bundler: Using rouge 3.3.0
Bundler: Using safe_yaml 1.0.4
Bundler: Using jekyll 3.8.5
Bundler: Using jekyll-feed 0.11.0
Bundler: Using jekyll-seo-tag 2.5.0
Bundler: Using minima 2.5.0
Bundler: Using tzinfo 2.0.0
Bundler: Using tzinfo-data 1.2018.9
Bundler: Using wdm 0.1.1
Bundler: Bundle complete! 5 Gemfile dependencies, 32 gems now installed.
Bundler: Use 'bundle info [gemname]' to see where a bundled gem is installed.
New jekyll site installed in C:/Users/neilb/Desktop/website/anteaterpeter.blog.

C:\Users\neilb\Desktop\website>cd anteaterpeter.blog
C:\Users\neilb\Desktop\website\anteaterpeter.blog>bundle exec jekyll serve
Configuration file: C:/Users/neilb/Desktop/website/anteaterpeter.blog/_config.yml
Source: C:/Users/neilb/Desktop/website/anteaterpeter.blog
Destination: C:/Users/neilb/Desktop/website/anteaterpeter.blog/_site
Incremental build: disabled. Enable with --incremental
Generating...
Jekyll Feed: Generating feed for posts
done in 2.32 seconds.
Auto-regeneration: enabled for 'C:/Users/neilb/Desktop/website/anteaterpeter.blog'
Server address: http://127.0.0.1:4000/
Server running... press ctrl-c to stop.
```

5. Now open your browser and surf on over to 'localhost:4000'. There you will be able to see your website and any changes you have made before 'committing' those changes to your github page.

Some notes:

1. You can play around with the files in your blog folder (like the 'about.md' and the '_post' file). Once you have saved those changes, you can reload your local webpage to see those changes reflected on your 'site'
2. One exception is if you change the '_config.yml' file. Any change you make here, you will have to close out of your 'server' and re-run the command 'bundle exec Jekyll serve' to restart everything. Then you should be able to see those change reflected on your site.

Step 3: Customize your website with a pre-made HTML5 layout

Note: while you could set up your website using Jekyll entirely, in this tutorial we will only use it in order to see our website locally (step 2.5). Instead we will use pre-made HTML5 layouts.

1. Pick an HTML5 template you like and download it. Note that you will be able to change colors, banners, categories, etc. later on so you should decide based on the overall structure of the layout. Some resources with free templates: <https://html5up.net/>, <https://colorlib.com/wp/free-html5-website-templates/>, <https://templated.co/>.
2. Optional: familiarize yourself with basic HTML language (https://www.w3schools.com/html/html_intro.asp)