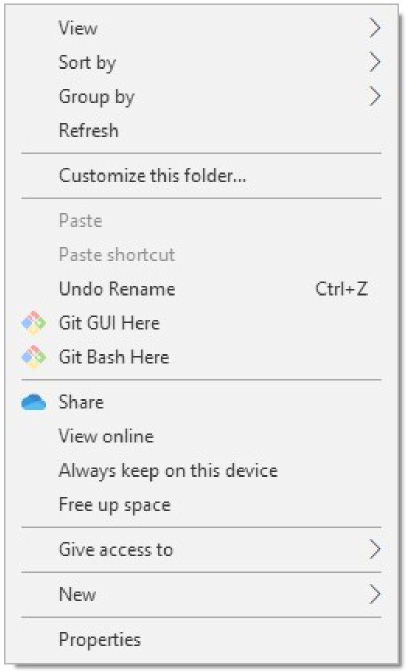
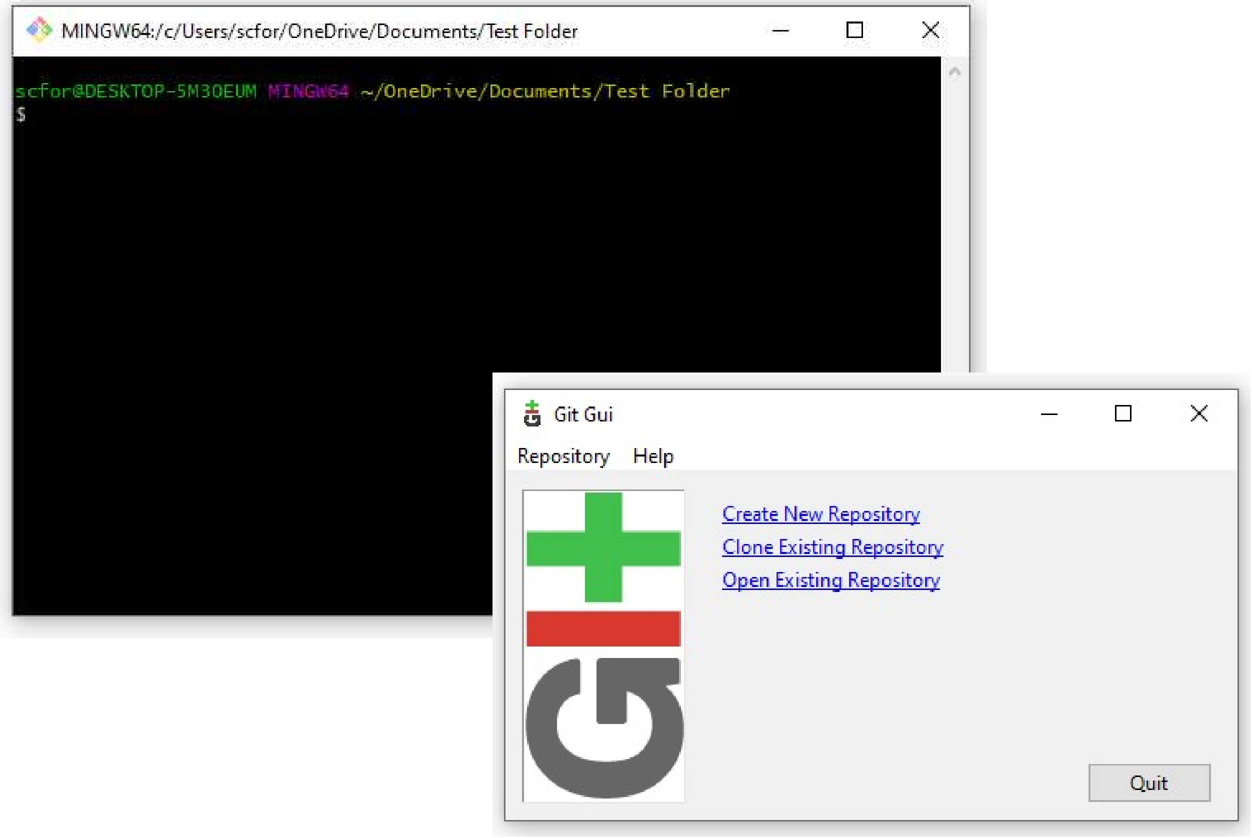
git good

A git crash course

For Windows: new right click options




MINGW64:/c/Users/scfor/OneDrive/Documents/Test Folder

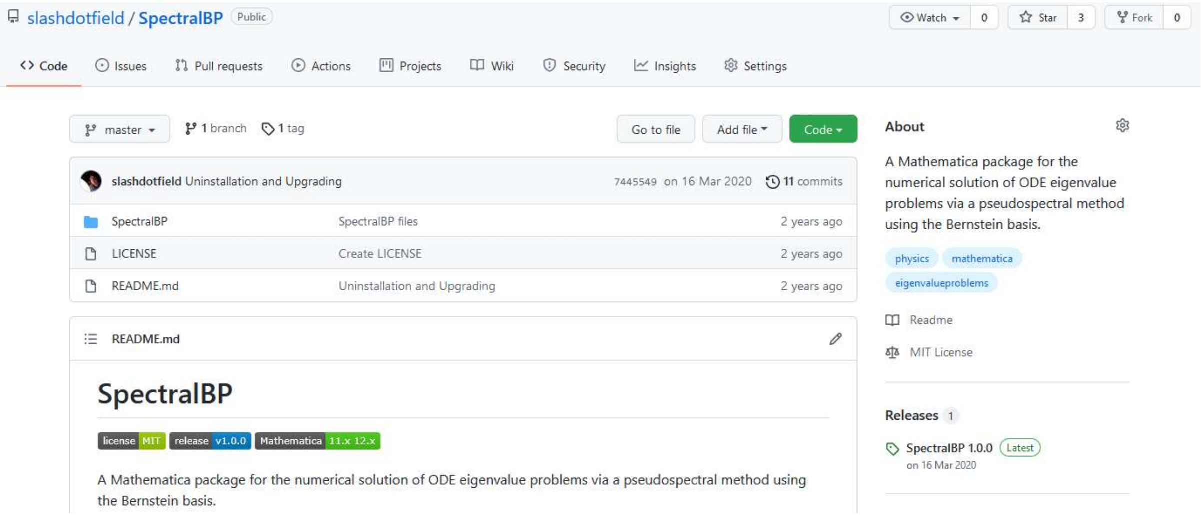
```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder
$
```

Git Gui
Repository
Help
Create New Repository
Clone Existing Repository
Open Existing Repository
Quit

For Linux/MacOS

GitHub vs git

- Git is a version control software
- GitHub is a web-based company that provides many tools that integrate into git



slashdotfield / **SpectralBP** Public

slashdotfield Uninstallation and Upgrading 7445549 on 16 Mar 2020 **11** commits

SpectralBP	SpectralBP files	2 years ago
LICENSE	Create LICENSE	2 years ago
README.md	Uninstallation and Upgrading	2 years ago

README.md

SpectralBP

license MIT release v1.0.0 Mathematica 11.x 12.x

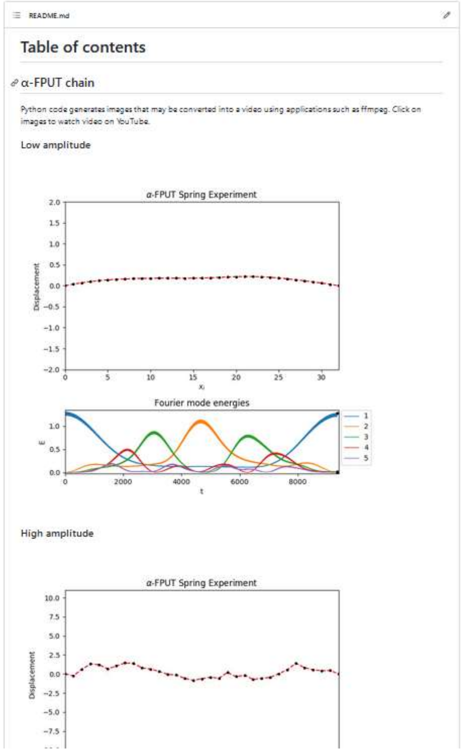
A Mathematica package for the numerical solution of ODE eigenvalue problems via a pseudospectral method using the Bernstein basis.

About

A Mathematica package for the numerical solution of ODE eigenvalue problems via a pseudospectral method using the Bernstein basis.

Releases 1

SpectralBP 1.0.0 Latest
on 16 Mar 2020



README.md

Table of contents

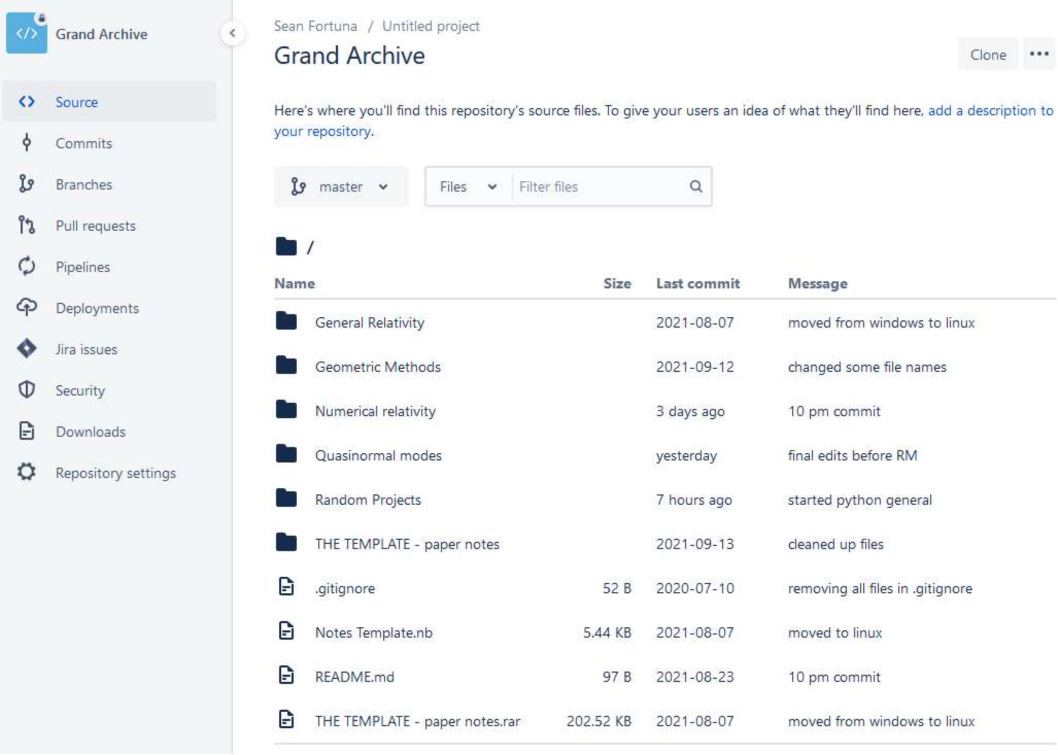
α-FPUT chain

Python code generates images that may be converted into a video using applications such as ffmpeg. Click on images to watch video on YouTube.

Low amplitude

High amplitude

There are others (Bitbucket)



Sean Fortuna / Untitled project

Grand Archive

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, add a description to your repository.

/

Name	Size	Last commit	Message
General Relativity		2021-08-07	moved from windows to linux
Geometric Methods		2021-09-12	changed some file names
Numerical relativity		3 days ago	10 pm commit
Quasinormal modes		yesterday	final edits before RM
Random Projects		7 hours ago	started python general
THE TEMPLATE - paper notes		2021-09-13	cleaned up files
.gitignore	52 B	2020-07-10	removing all files in .gitignore
Notes Template.nb	5.44 KB	2021-08-07	moved to linux
README.md	97 B	2021-08-23	10 pm commit
THE TEMPLATE - paper notes.rar	202.52 KB	2021-08-07	moved from windows to linux

Command Table of Contents

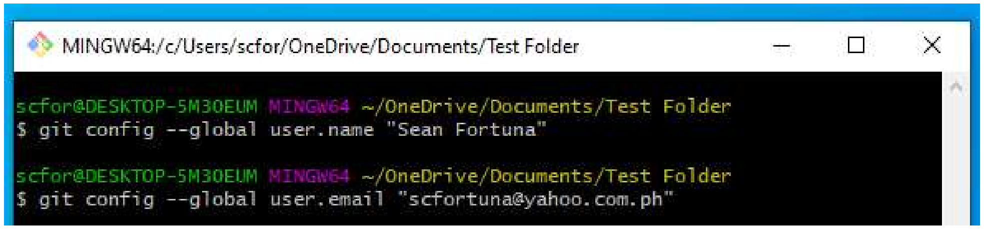
- We shall try to go through the following (arranged in descending order of rarity):
    - Git setup commands
    - Repository setup commands
    - Versioning commands
    - **Collaborative commands (new!)**
- We shall try to do these via the bash shell (it’ll look good on your CV)

Getting started – user name and email

Two commands:

$ git config --global user.name “**###**”

$ git config --global user.email “**###**”



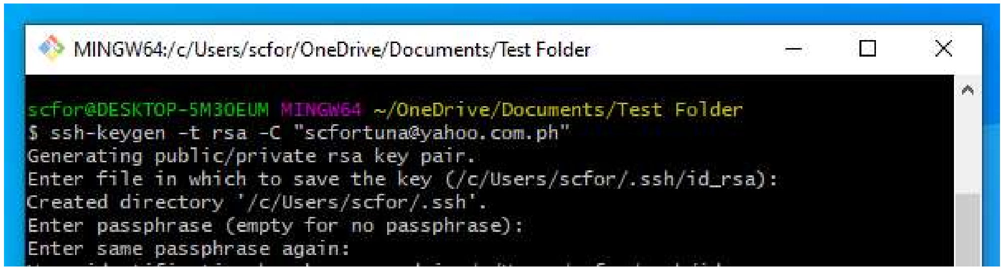
```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder
$ git config --global user.name "Sean Fortuna"

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder
$ git config --global user.email "scfortuna@yahoo.com.ph"
```

This lets git know when we eventually go to versioning, who is responsible for what contribution to the git repository

Getting start – SSH key generation

- One command:

$ ssh-keygen –t rsa –C “**###**”



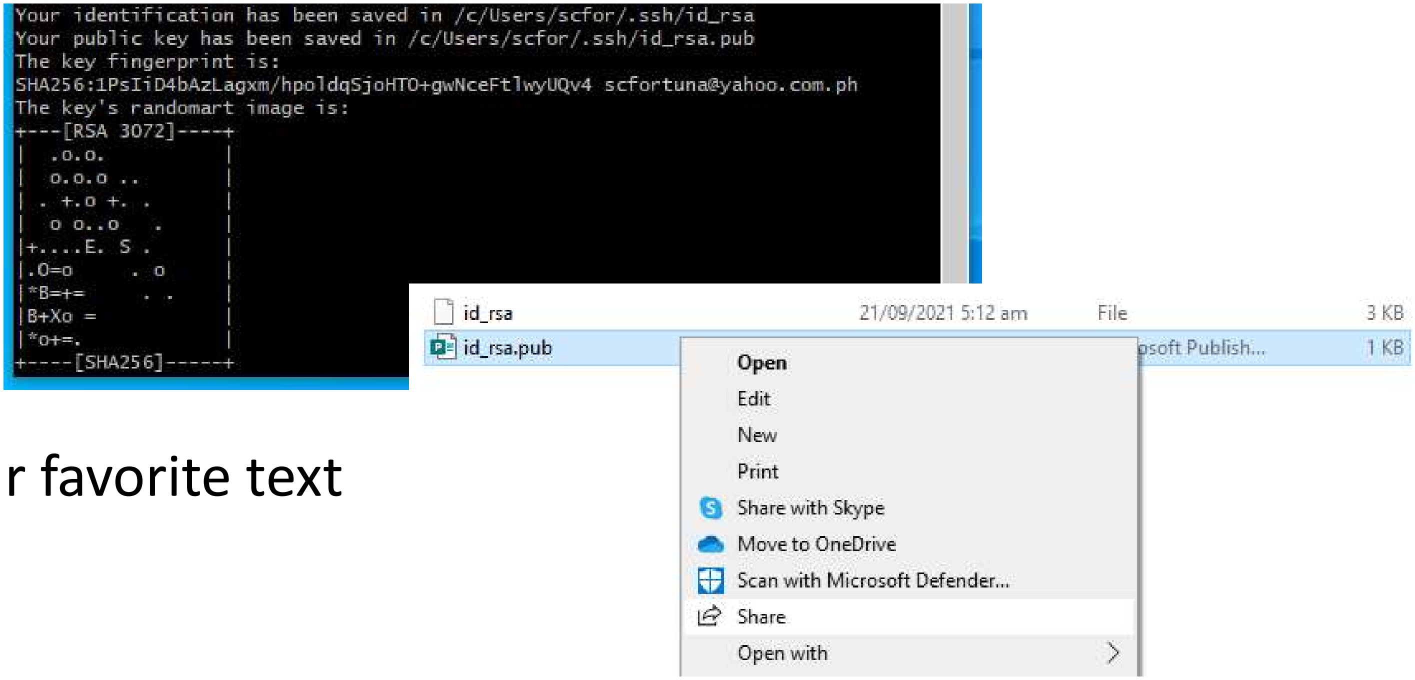
```
scfor@DESKTOP-5M3OEUM MINGW64 ~/OneDrive/Documents/Test Folder
$ ssh-keygen -t rsa -C "scfortuna@yahoo.com.ph"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/scfor/.ssh/id_rsa):
Created directory '/c/Users/scfor/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

No further input required. Just keep pressing “Enter” until it is all over.

This lets you use GitHub without entering your password every time you do something.

Getting started – Linking ssh keys to your github

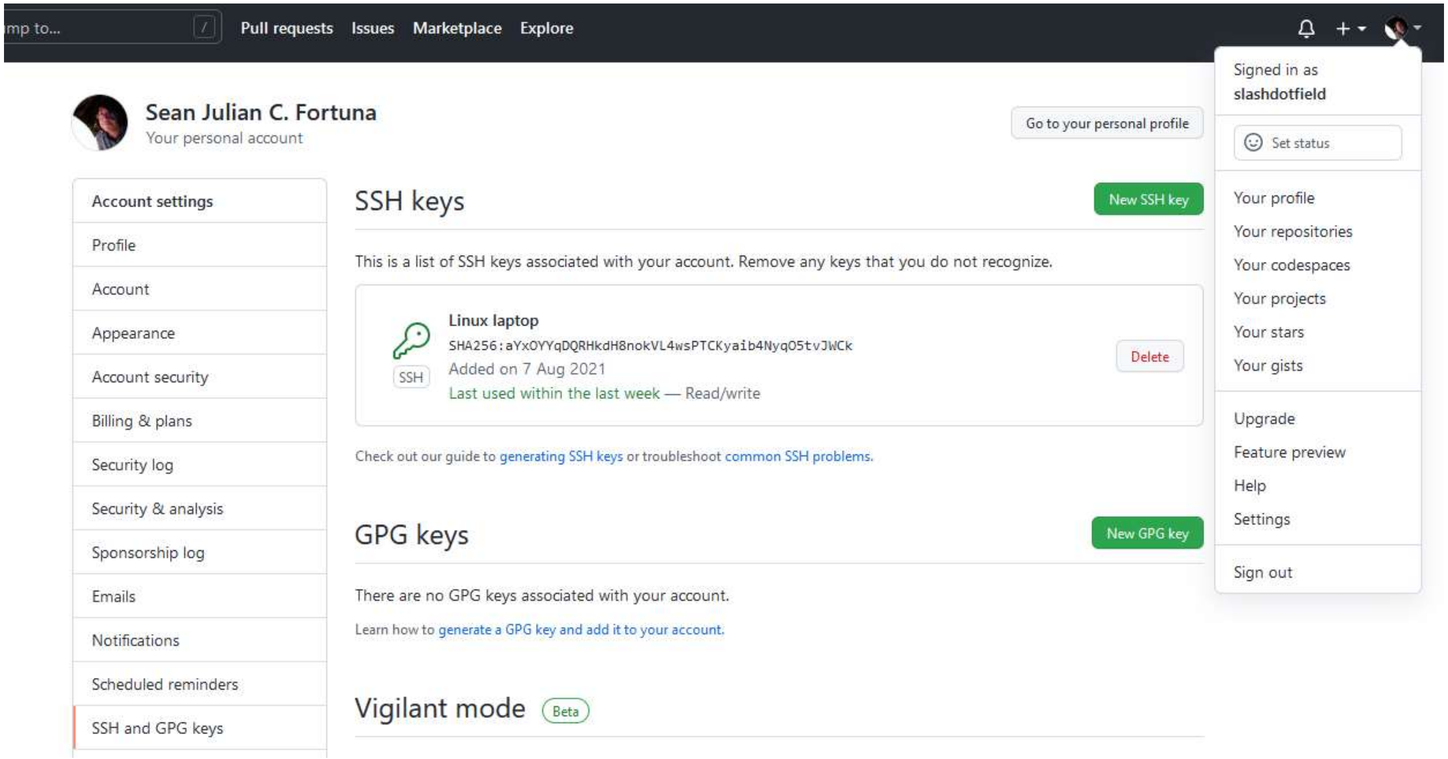
- Navigate to where your public key is



```
Your identification has been saved in /c/Users/scfor/.ssh/id_rsa
Your public key has been saved in /c/Users/scfor/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1PsIiD4bAzLagxm/hpoldqSjoHTO+gwNceFtlwyUQv4 scfortuna@yahoo.com.ph
The key's randomart image is:
+---[RSA 3072]----+
|  .o.o.          |
|  o.o.o ..       |
| . +.o +. .      |
|  o o..o   .     |
|+....E. S .      |
|.O=o     . o     |
|*B=+=     . .    |
|B+Xo =           |
|*o+=.            |
+----[SHA256]-----+
```

- Open with your favorite text editor

Getting started – Linking ssh keys to your github



Sean Julian C. Fortuna
Your personal account

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Linux laptop
`SHA256:aYxOYYqDQRHkdH8nokVL4wsPTCKyaib4NyqO5tvJWCk`
Added on 7 Aug 2021
Last used within the last week — Read/write

Check out our guide to generating SSH keys or troubleshoot common SSH problems.

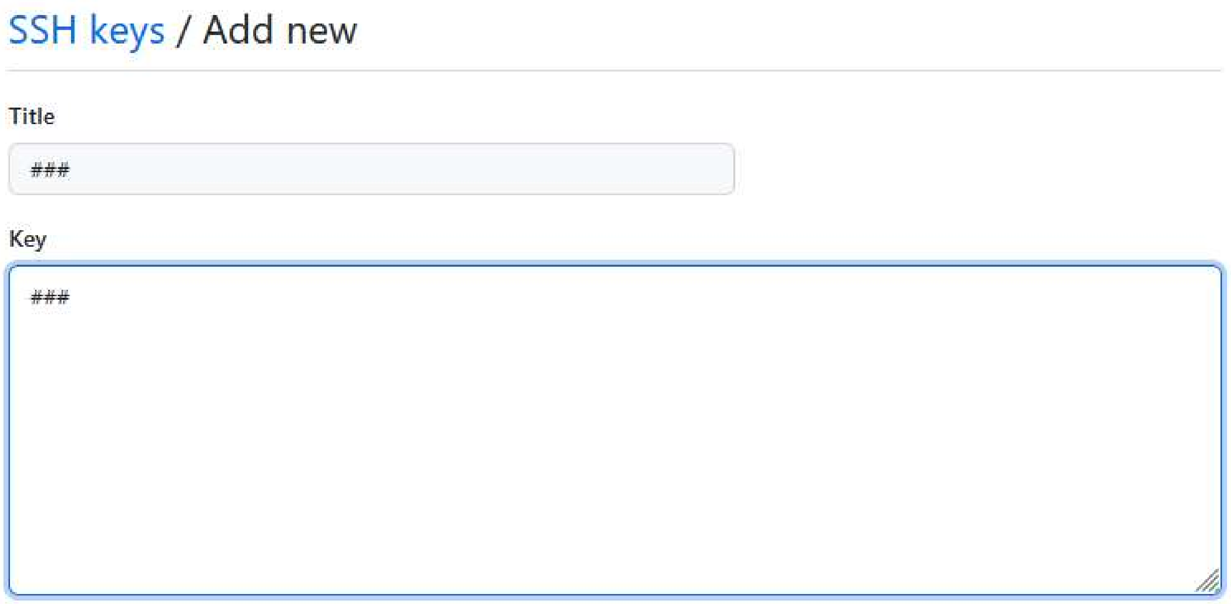
GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

Vigilant mode

Getting started – Linking ssh keys to your github



SSH keys / Add new

Title

###

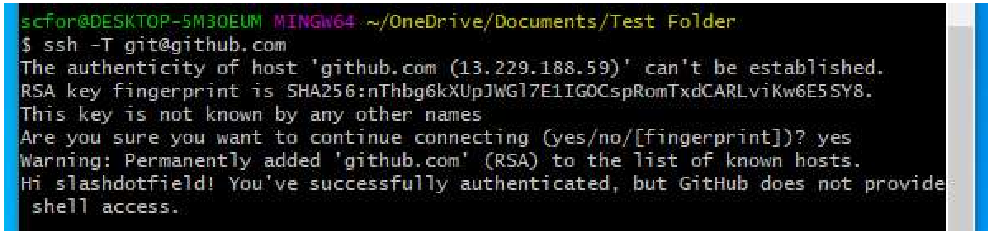
Key

###

Getting started – the truth is revealed

- One command:

$ ssh –T git@github.com

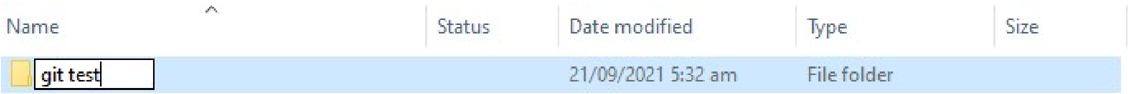


```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder
$ ssh -T git@github.com
The authenticity of host 'github.com (13.229.188.59)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (RSA) to the list of known hosts.
Hi slashdotfield! You've successfully authenticated, but GitHub does not provide
 shell access.
```

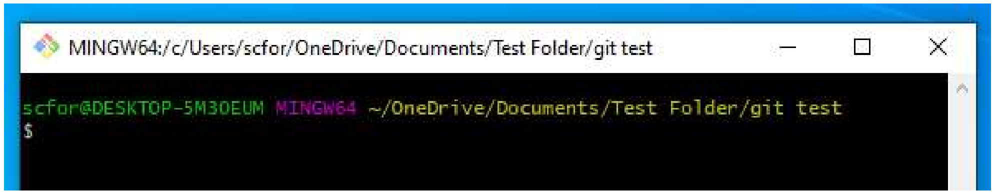
Congratulations! We may begin with creating a repository.

Creating a local repository

- Create a new folder and open the bash terminal there



Name	Status	Date modified	Type	Size
git test		21/09/2021 5:32 am	File folder	



MINGW64:/c/Users/scfor/OneDrive/Documents/Test Folder/git test

```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder/git test
$
```

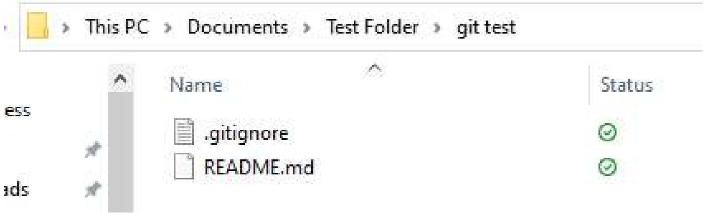
Initializing your local repository

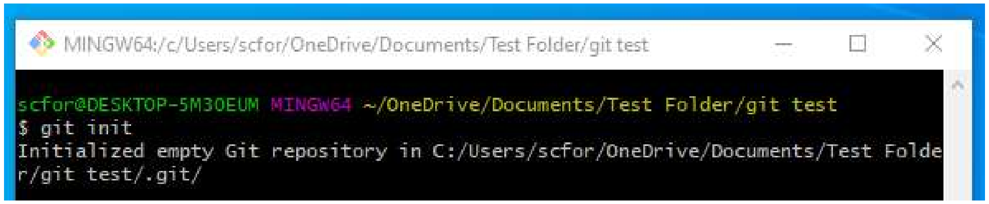
- Three commands:

$ git init

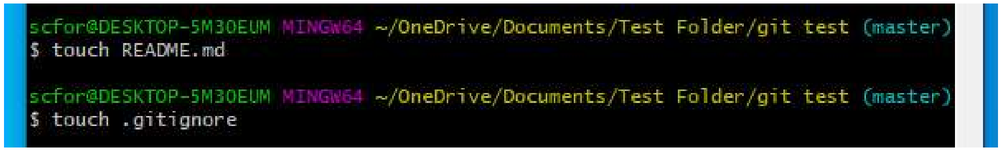
$ touch README.md

$ touch .gitignore




MINGW64:/c/Users/scfor/OneDrive/Documents/Test Folder/git test

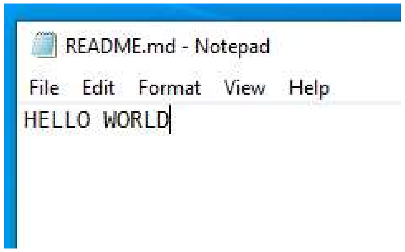
```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder/git test
$ git init
Initialized empty Git repository in C:/Users/scfor/OneDrive/Documents/Test Folde
r/git test/.git/
```



```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder/git test (master)
$ touch README.md

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder/git test (master)
$ touch .gitignore
```

Initializing your local repository

- Open the README.md file with a text editor, and type the most cliché message you can think of

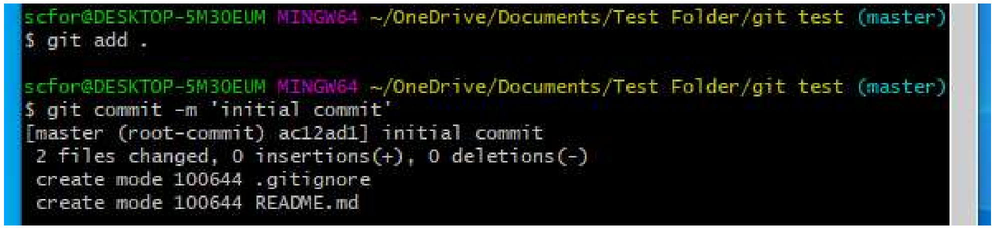


HELLO WORLD

- There are **three** main idioms you need to remember after initializing your repository. **Two** of them are related to your local repository.

$ git add .

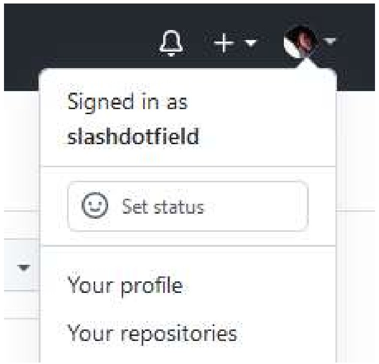
$ git commit –m ‘###’



```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder/git test (master)
$ git add .

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/Test Folder/git test (master)
$ git commit -m 'initial commit'
[master (root-commit) ac12ad1] initial commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
 create mode 100644 README.md
```

Initializing your GitHub repository

- Navigate to Your Repositories and click ‘New’

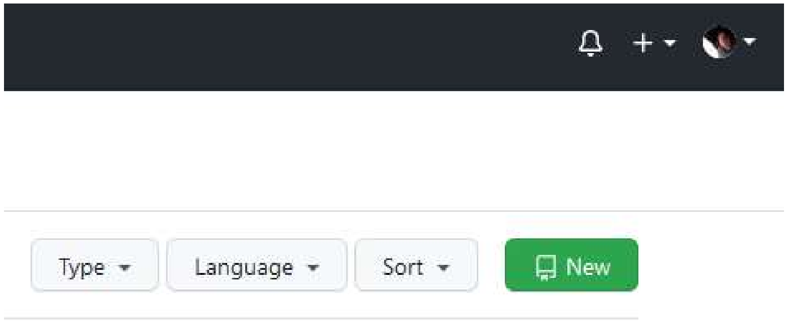


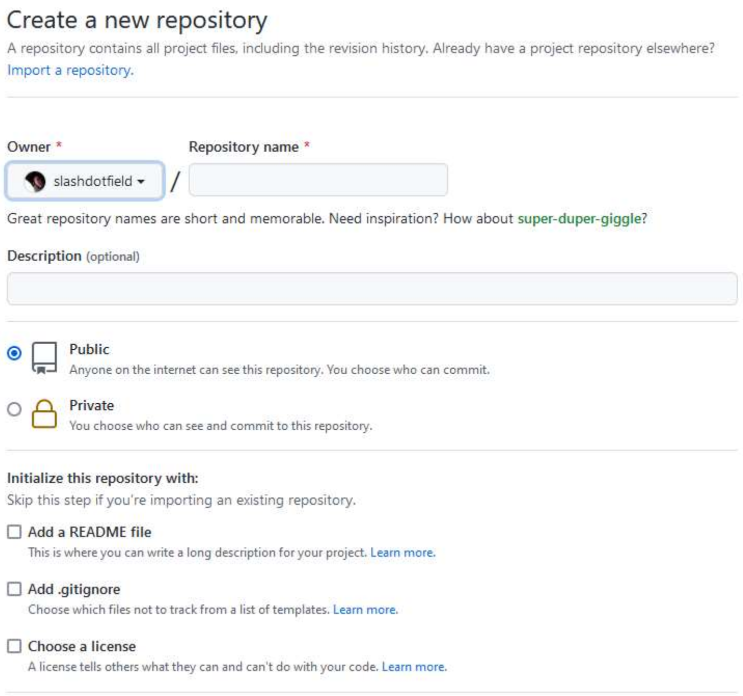
Signed in as
slashdotfield

☺ Set status

Your profile

Your repositories





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 *

slashdotfield ▾ /

Great repository names are short and memorable. Need inspiration? How about super-duper-giggle?

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.

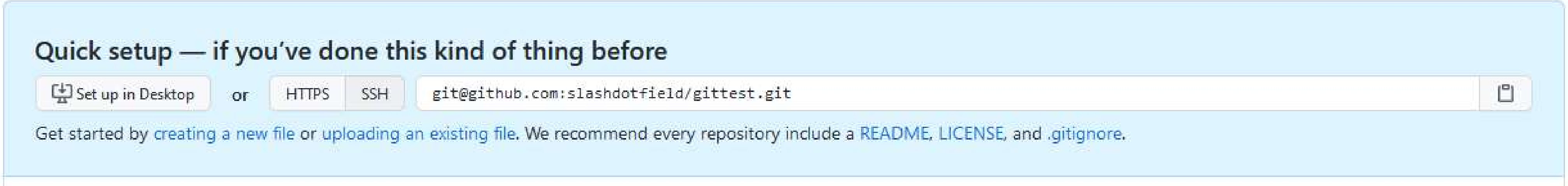
□ Add a README file
This is where you can write a long description for your project. Learn more.

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

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

Initializing your GitHub repository

- Fill in appropriately and copy the SSH quick setup string



Quick setup — if you've done this kind of thing before

```
git@github.com:slashdotfield/gittest.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

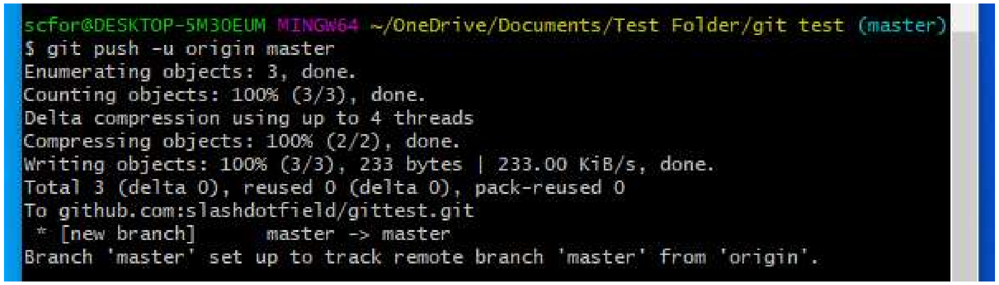
- To link your local repo to the GitHub repo

$ git remote add origin **###**

Initializing your GitHub repository

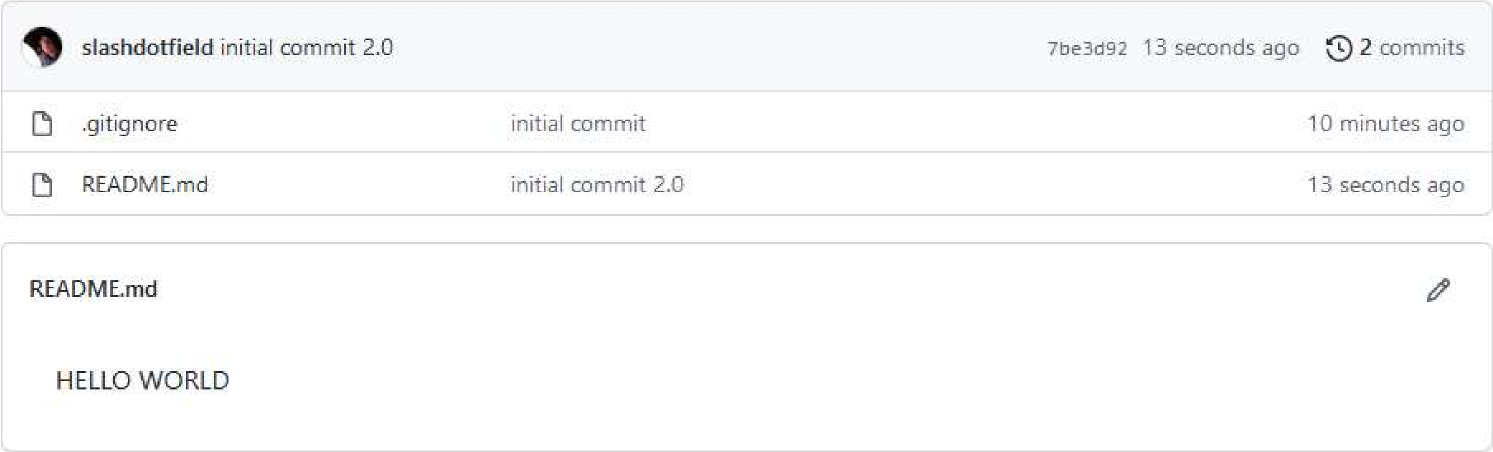
- The final main idiom you need to remember after initialization relates to syncing your local and remote repositories

$ git push –u origin master



```
scfor@DESKTOP-5M3OEUM MINGW64 ~/OneDrive/Documents/Test Folder/git test (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 233 bytes | 233.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:slashdotfield/gittest.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Congratulations!



slashdotfield initial commit 2.0 | 7be3d92 13 seconds ago | 2 commits

.gitignore	initial commit	10 minutes ago
README.md	initial commit 2.0	13 seconds ago

README.md

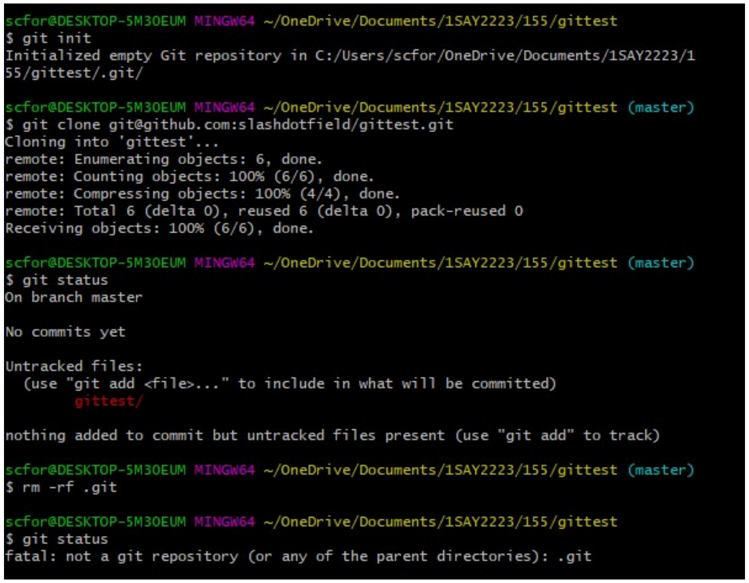
HELLO WORLD

Typical workflow that follows now is

1. Edit code
2. $ git add .
3. $ git commit –m ‘###’
4. $ git push –u origin master

Some remaining questions

- How do you stop a folder from git tracking?
    1. Delete the .git folder manually
    2. rm –rf .git



```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/gittest
$ git init
Initialized empty Git repository in C:/Users/scfor/OneDrive/Documents/1SAY2223/1
55/gittest/.git/

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/gittest (master)
$ git clone git@github.com:slashdotfield/gittest.git
Cloning into 'gittest'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/gittest (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        gittest/

nothing added to commit but untracked files present (use "git add" to track)

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/gittest (master)
$ rm -rf .git

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/gittest
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

Some remaining questions

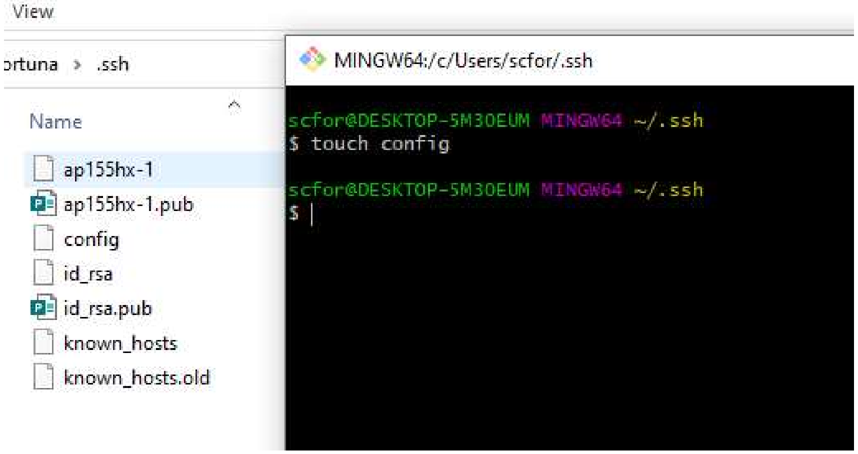
- How about collaboration?
  - clone, pull, conflict and merging
  - Two solutions. Either,
    - a repository gives access to a bunch of public keys (for a large class, hassle)
    - a repository gives access to a single public key, and a private key is distributed to trusted people

We shall be trying the second route.

Permanently adding a new private key

- Go to your ssh folder. Cut and paste the public and private keys there and run

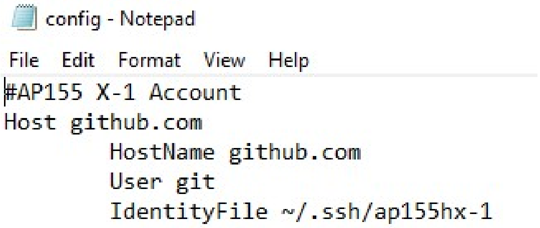
$ touch config



```
scfor@DESKTOP-5M30EUM MINGW64 ~/.ssh
$ touch config

scfor@DESKTOP-5M30EUM MINGW64 ~/.ssh
$ |
```

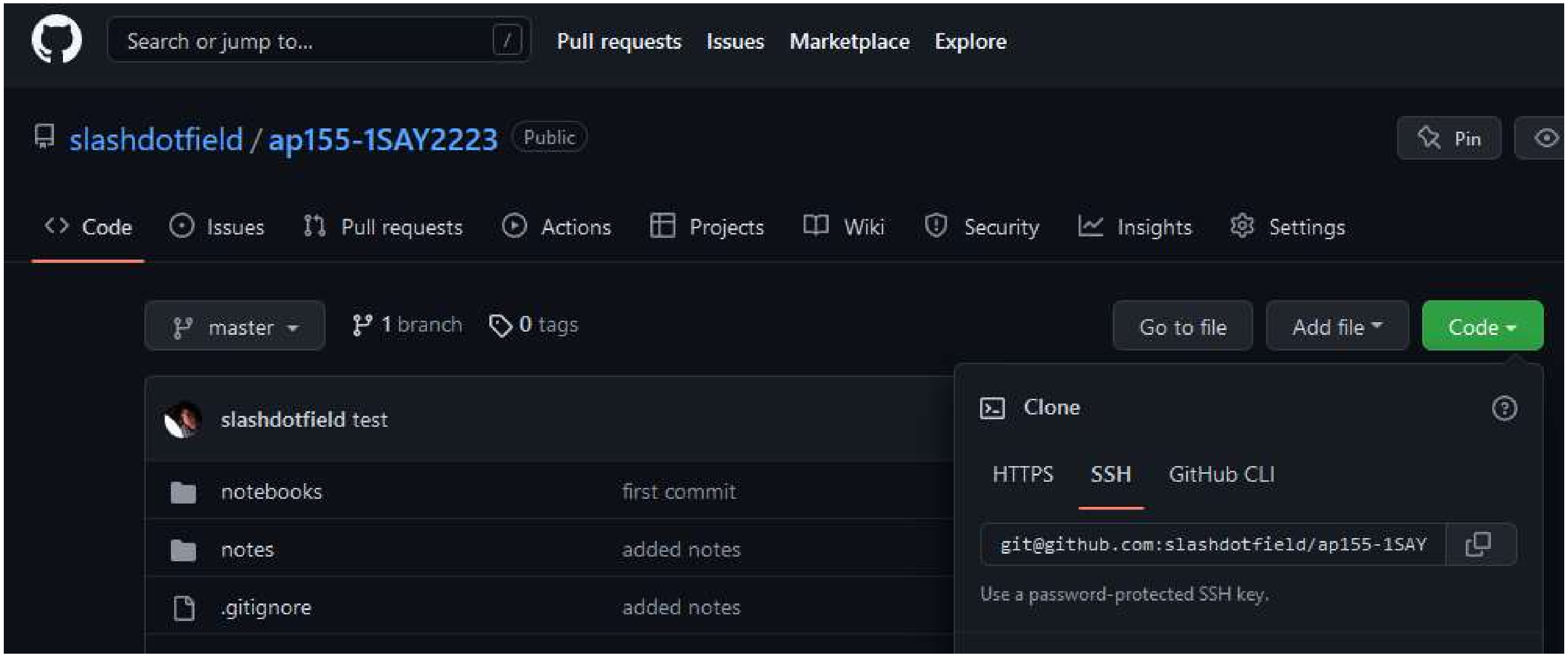
Permanently adding a new private key

- Open the config file with a text editor and put in the following text



```
#AP155 X-1 Account
Host github.com
        HostName github.com
        User git
        IdentityFile ~/.ssh/ap155hx-1
```

Cloning our class repo

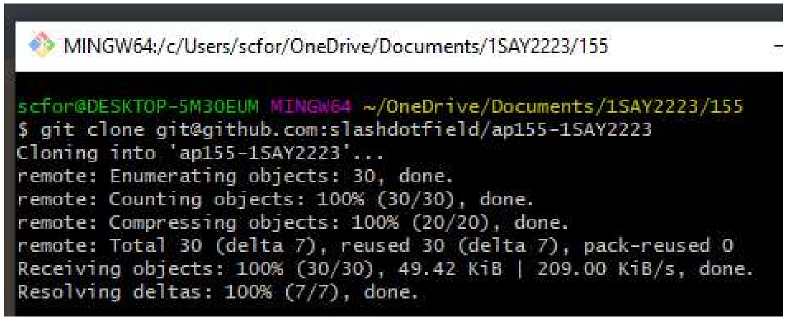


```
git@github.com:slashdotfield/ap155-1SAY
```

Use a password-protected SSH key.

Cloning our class repo

$ git clone git@github.com:slashdotfield/ap155-1SAY2223



```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155
$ git clone git@github.com:slashdotfield/ap155-1SAY2223
Cloning into 'ap155-1SAY2223'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 30 (delta 7), reused 30 (delta 7), pack-reused 0
Receiving objects: 100% (30/30), 49.42 KiB | 209.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.
```

Let’s do some activities

Commands for local repo

```
$ git add .
$ git commit –m “###”
```

Commands for remote repo

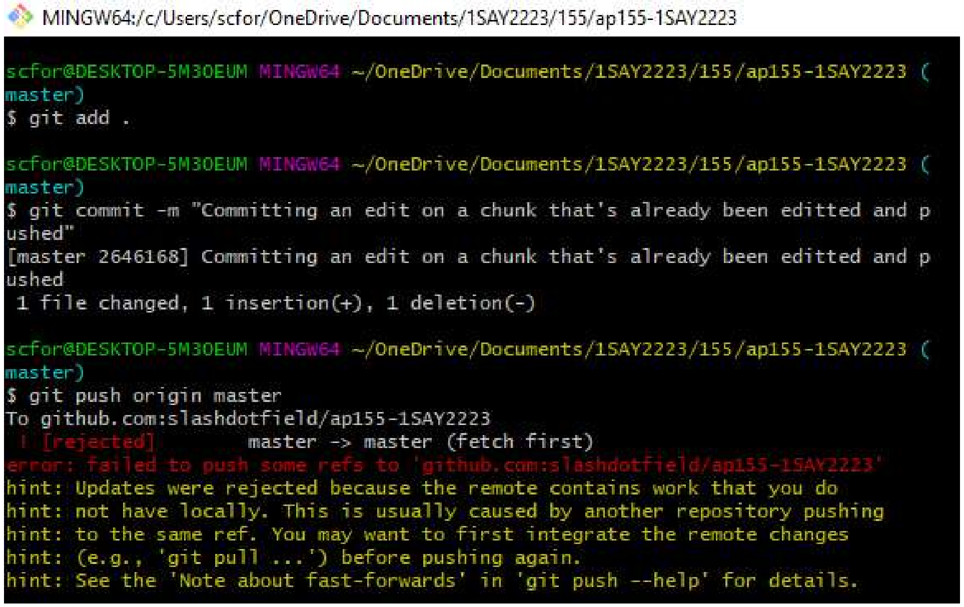
```
$ git push origin master
$ git pull origin master
```

Resolving conflicts and merges

Pushing and pulling edits are all fine if you edit different chunks of data.

Two people can edit the same file, **so long as they do not overlap**.

What happens otherwise? CONFLICT! REJECTION!

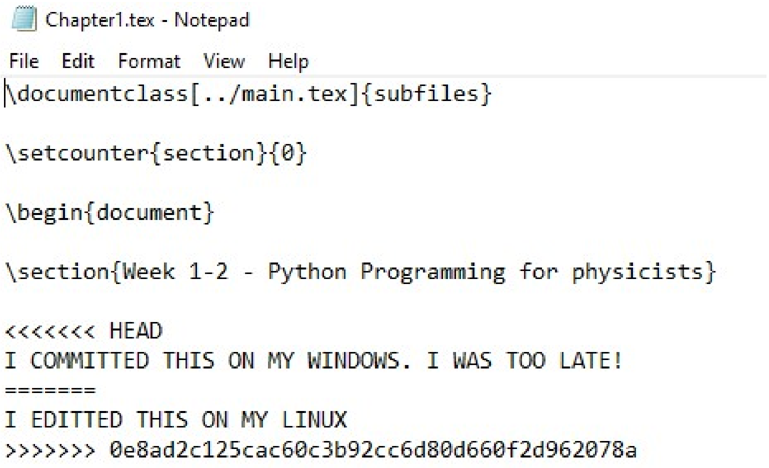


MINGW64:/c/Users/scfor/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223

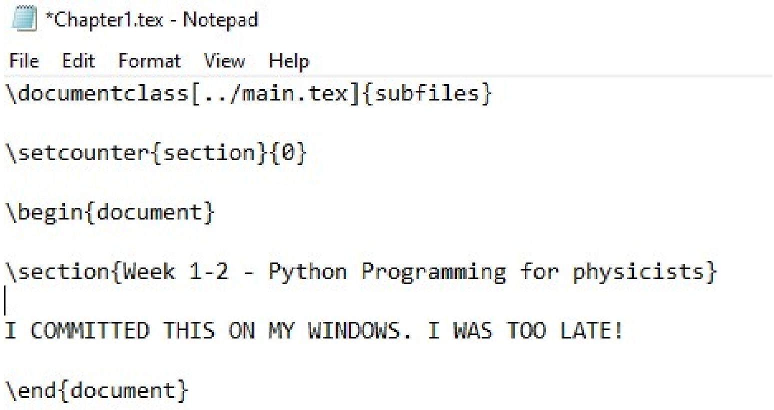
```
scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223 (
master)
$ git add .

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223 (
master)
$ git commit -m "Committing an edit on a chunk that's already been editted and p
ushed"
[master 2646168] Committing an edit on a chunk that's already been editted and p
ushed
 1 file changed, 1 insertion(+), 1 deletion(-)

scfor@DESKTOP-5M30EUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223 (
master)
$ git push origin master
To github.com:slashdotfield/ap155-1SAY2223
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'github.com:slashdotfield/ap155-1SAY2223'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Resolving conflict and merges

One needs to pull first and manually resolve the conflict.

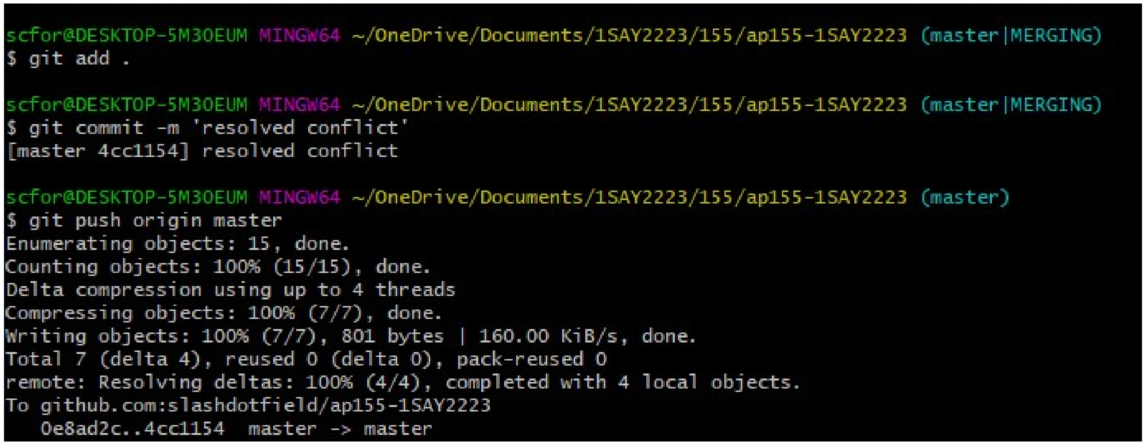


```
\documentclass[../main.tex]{subfiles}

\setcounter{section}{0}

\begin{document}

\section{Week 1-2 - Python Programming for physicists}

<<<<<<< HEAD
I COMMITTED THIS ON MY WINDOWS. I WAS TOO LATE!
=======
I EDITTED THIS ON MY LINUX
>>>>>>> 0e8ad2c125cac60c3b92cc6d80d660f2d962078a
```



```
\documentclass[../main.tex]{subfiles}

\setcounter{section}{0}

\begin{document}

\section{Week 1-2 - Python Programming for physicists}

I COMMITTED THIS ON MY WINDOWS. I WAS TOO LATE!

\end{document}
```

- All is now well with the world



```
scfor@DESKTOP-5M3OEUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223 (master|MERGING)
$ git add .

scfor@DESKTOP-5M3OEUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223 (master|MERGING)
$ git commit -m 'resolved conflict'
[master 4cc1154] resolved conflict

scfor@DESKTOP-5M3OEUM MINGW64 ~/OneDrive/Documents/1SAY2223/155/ap155-1SAY2223 (master)
$ git push origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 801 bytes | 160.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:slashdotfield/ap155-1SAY2223
   0e8ad2c..4cc1154  master -> master
```