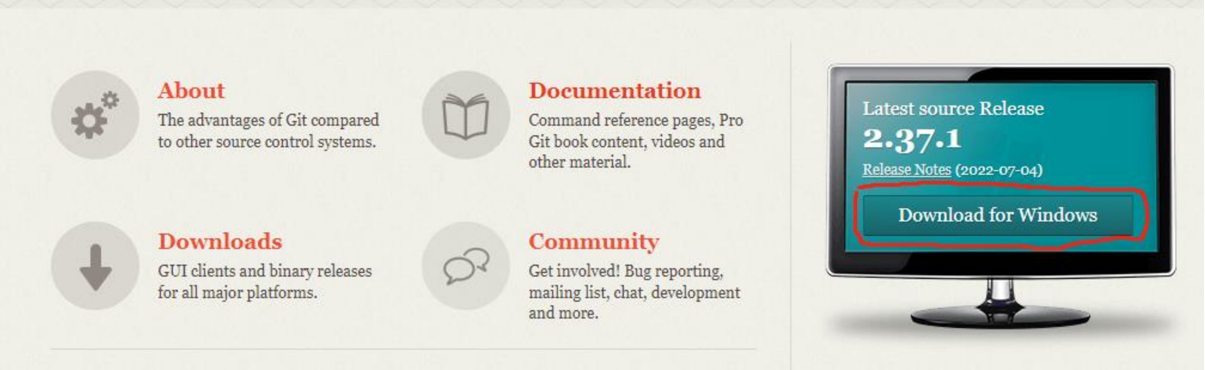
Git guide

------------------------------Download and Set up Environment-------------------------

1. Download git from the link [Git - Downloading Package (git-scm.com)]



About
The advantages of Git compared to other source control systems.

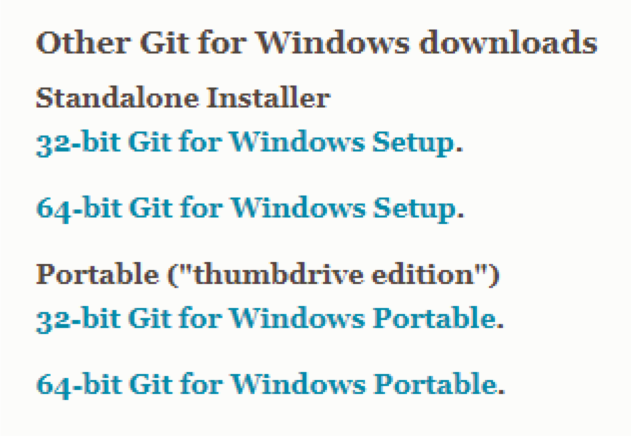
Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.37.1
Release Notes (2022-07-04)

2. Chose the version for your computer



Other Git for Windows downloads

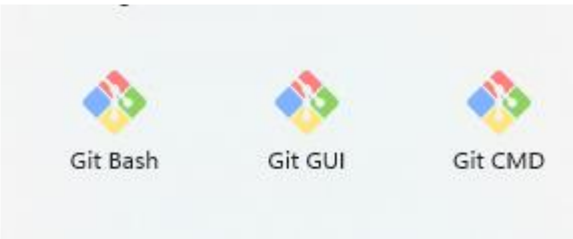
Standalone Installer
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

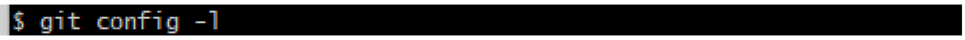
Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

3. Just click on next when you install
4. After install you'll see this 3 app here

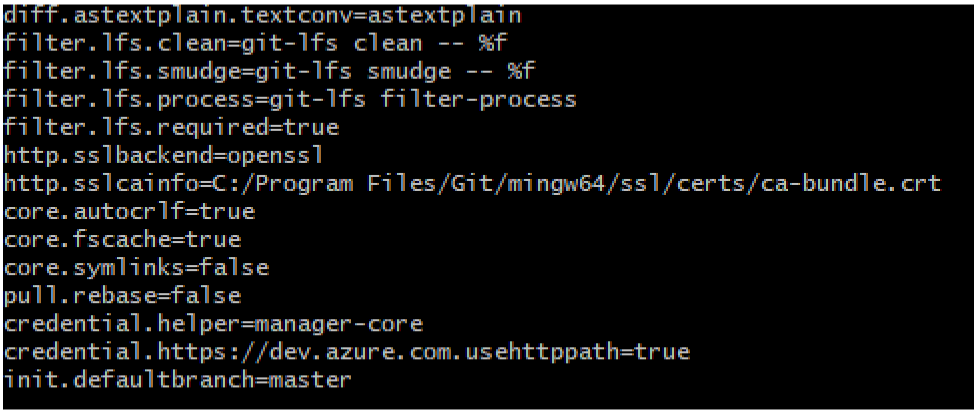


5. This 3 is just 3 different ways to open up the git shell (Terminals). Bash--Linux-style command line (most used, recommend); CMD--Windows-style command line; GUI--graphic chemical tool (unnecessary).
6. Open your git bash, type:



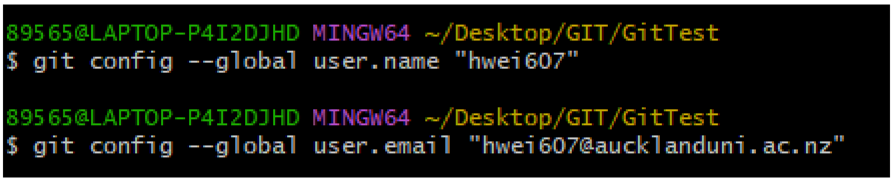
```
$ git config -l
```

To check your configuration



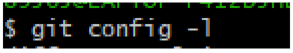
```
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
```

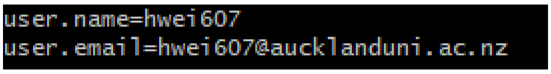
7. Then type these two lines to set up your name and email for your git



```
89565@LAPTOP-P4I2DJHD MINGW64 ~/Desktop/GIT/GitTest
$ git config --global user.name "hwei607"

89565@LAPTOP-P4I2DJHD MINGW64 ~/Desktop/GIT/GitTest
$ git config --global user.email "hwei607@aucklanduni.ac.nz"
```



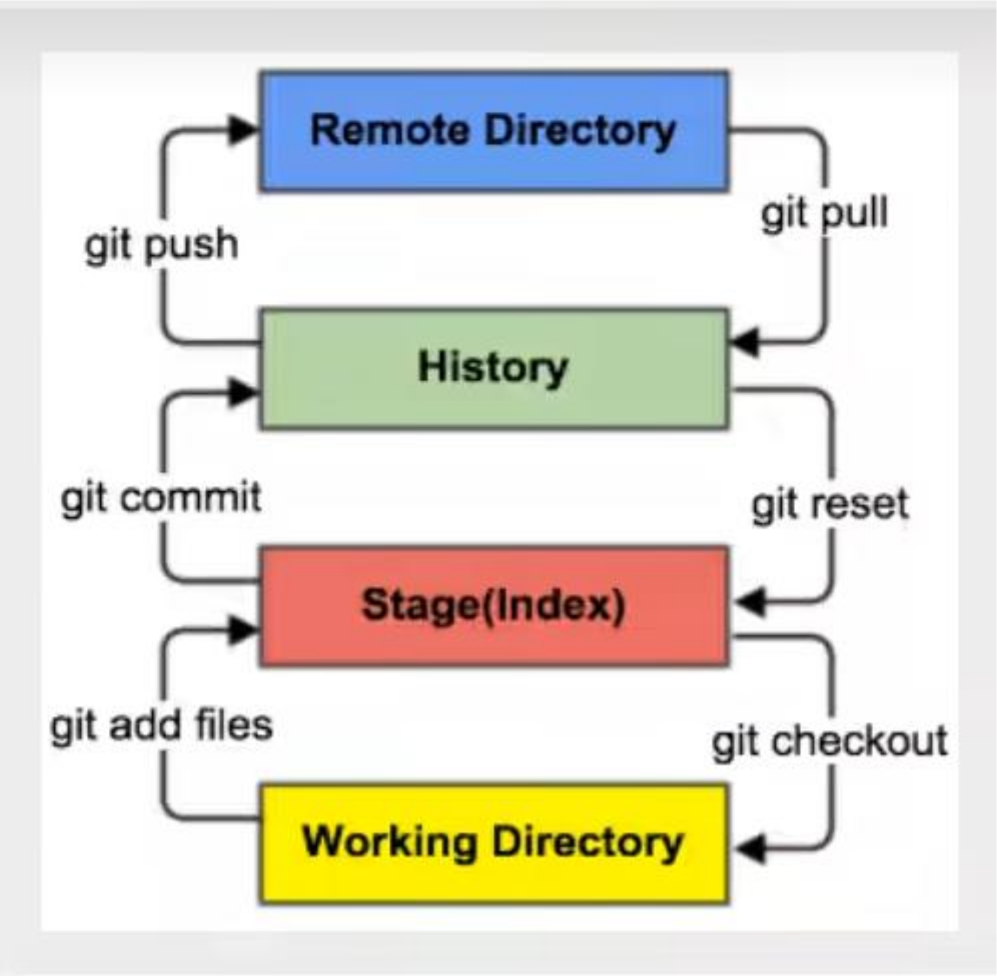


8. Type this line again: `$ git config -l`, and you will see your local information has been set

```
user.name=hwei607
user.email=hwei607@aucklanduni.ac.nz
```

at the bottom of the configuration.

----------------------------------so far, all the set-up has been done-------------------------------------



This is about the git/github basic workflow and theory.

Working Directory: This is the place where you store your project/code. This is on your local disk.

Stage: This is just a hidden file in your work directory use to stored the information about the file you make changes recently.

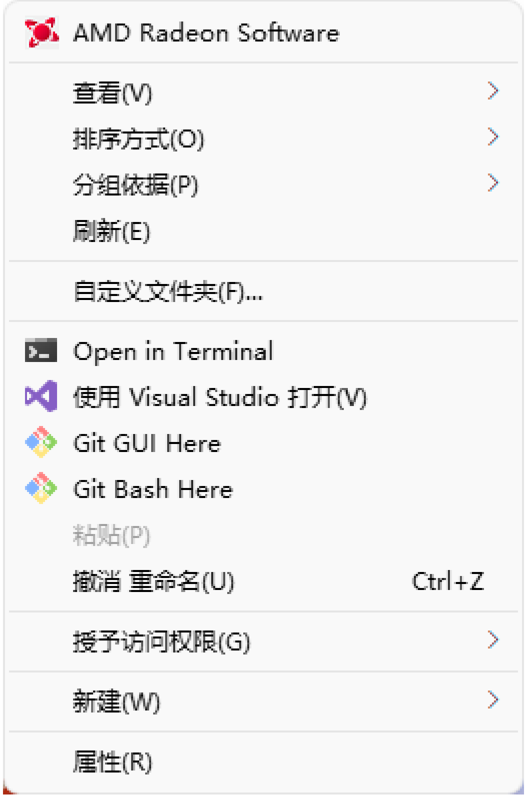
History: repository where actually save all your project

Remote Directory: The remote server like github.

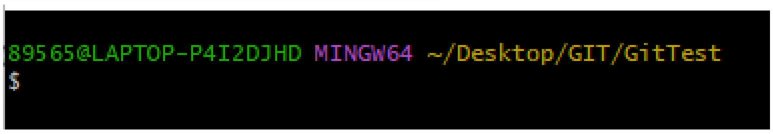
The only thing that you need to manage is just the working directory and the remote directory. The middle two process just need you to use cmd to do some operation.

Now this is the operation instance:

1.Right click on your working directory:



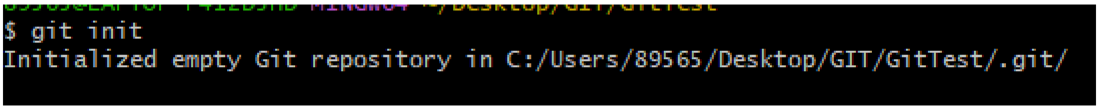
2.click on the “git bash here”



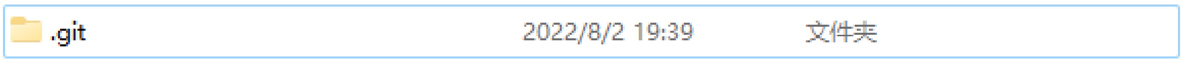
```
89565@LAPTOP-P4I2DJHD MINGW64 ~/Desktop/GIT/GitTest
$
```

You will see the directory is your working directory

3. Type git init to initalise a reisitory on your local file



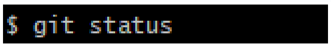
```
89565@LAPTOP-P4I2DJHD MINGW64 ~/Desktop/GIT/GitTest
$ git init
Initialized empty Git repository in C:/Users/89565/Desktop/GIT/GitTest/.git/
```



This will generate this .git file which is the hidden file I have mentioned before where to store your stage files.

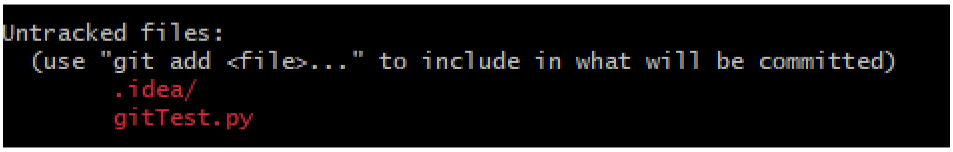
Otherwise, you can use git clone [url] to clone a repository form the remote directory to your own computer.

Then I can use touch cmd or create a .py file called gitTest.py and then use this code



```
$ git status
```

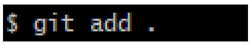
To check the status of the file and you will see



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

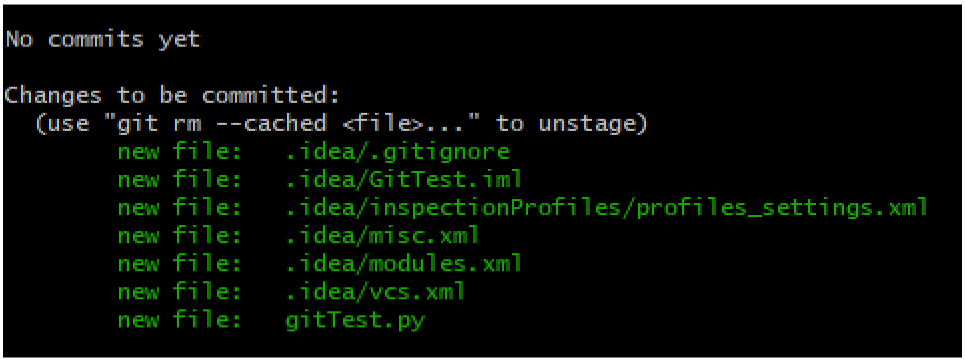
It's now untracked

Then you can use the cmd



```
$ git add .
```

Let's check the status again

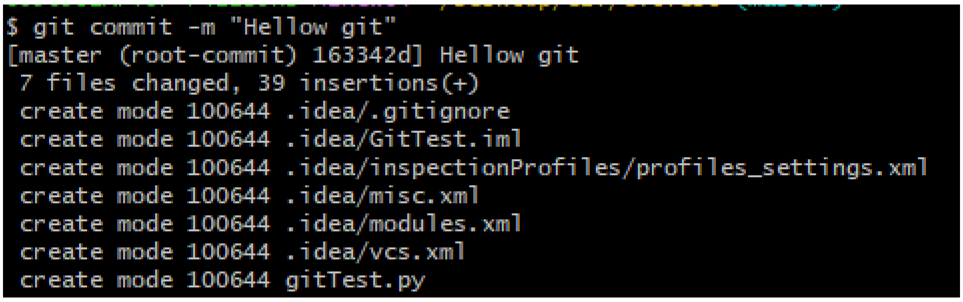


```
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .idea/.gitignore
        new file:   .idea/GitTest.iml
        new file:   .idea/inspectionProfiles/profiles_settings.xml
        new file:   .idea/misc.xml
        new file:   .idea/modules.xml
        new file:   .idea/vcs.xml
        new file:   gitTest.py
```

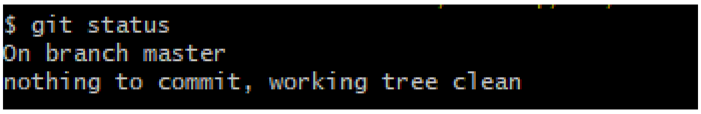
Now you can see the gitTest.py file is waiting you to commit it

Then use git commit -m "[message]" to put your file into your local repository. (-m "[]" this is to add some commit information).



```
$ git commit -m "Hellow git"
[master (root-commit) 163342d] Hellow git
 7 files changed, 39 insertions(+)
 create mode 100644 .idea/.gitignore
 create mode 100644 .idea/GitTest.iml
 create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
 create mode 100644 .idea/misc.xml
 create mode 100644 .idea/modules.xml
 create mode 100644 .idea/vcs.xml
 create mode 100644 gitTest.py
```

Now we check status again

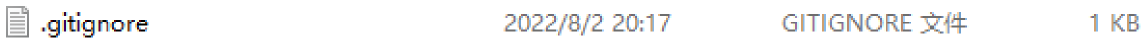


```
$ git status
On branch master
nothing to commit, working tree clean
```

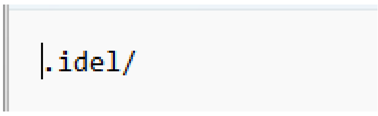
It's saying working tree clean!!! That's good.

After you have a remote directory then you can just use push cmd to update your file onto your github!!!

Extension: If you don't want to put every single file into your repository you either create a .gitignore file or use git add [file-name].



.gitignore 2022/8/2 20:17 GITIGNORE 文件 1 KB



```
.idel/
```

This is the example for I want to ignore the .idea file