Outline

- Since you are required to create a private Git Repo
- AND Git is a very useful tool to manage codes

- We will learn some basics of Git today

Install git

- Usually already installed on MacOS and Linux
- Usually need to be installed on Windows
- Check detailed tutorial for installation on at:
  - https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- Or the download page:
  - https://git-scm.com/download

❖ Notice: you might need to re-open the terminal (CMD) after installing using command lines.

Basic git commands

- Configure your info
- Usually, you only need to set user.name and user.email

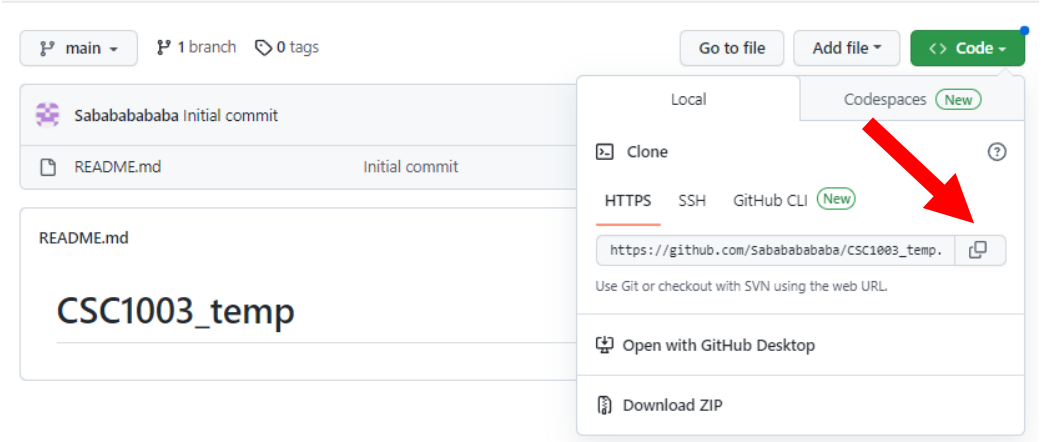
```
git config --global user.name

git config --global user.email
```

- You can use `git config --global credential.helper store` to skip authentication, but your pswd will be saved locally. (I personally don't like it)

Basic git commands

- You can also clone an existing git repository
- Usually you only need git clone URL




```
E:¥git¥p2>git clone https://github.com/Sababababaha/CSC1003_temp.git
Cloning into 'CSC1003_temp'...
info: please complete authentication in your browser...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

.git	2023/1/9 19:05	文件夹	
README.md	2023/1/9 19:05	MD 文件	1 KB

- Will ask you for authentication if it is a private repo (ez for windows and linux)
- (For MacOS, create a token at https://github.com/settings/tokens and use the token as password)

Make changes to the repo

- Suppose you have done your work, and updated some files in the repo (add files/edit codes…)

- You want to post these changes to the server
- Follow these steps for the simplest method:

1. Check
2. Add
3. Commit
4. push

Check

- Use **git status** to check what is changed

```
E:¥git¥p2¥CSC1003_temp>git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

- (If not sure you are at the newest ver., use **git fetch** to update local info)

Add

- Use **git add [files]** to add files to commit
- **[files]** can be file name or directory

```
E:¥git¥p2¥CSC1003_temp>git add hw.java

E:¥git¥p2¥CSC1003_temp>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hw.java

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

add a single file

```
E:¥git¥p2¥CSC1003_temp>git add .

E:¥git¥p2¥CSC1003_temp>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hw.java
        new file:   hw.py
```

add a directory

Commit

- Use **git commit –m "MESSAGE"** to commit all added changes

```
E:¥git¥p2¥CSC1003_temp>git commit -m "add 2 files, 1 java, 1 py"
[main 1395b2c] add 2 files, 1 java, 1 py
 2 files changed, 6 insertions(+)
 create mode 100644 hw.java
 create mode 100644 hw.py
```

- Now check the status it will tells you that you are 1 commit ahead

```
E:¥git¥p2¥CSC1003_temp>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Push

- Use **git push** to post the changes to server
- Might ask for authentication

```
E:¥git¥p2¥CSC1003_temp>git push
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 32 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 439 bytes | 439.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Sababababababa/CSC1003_temp.git
   7b700c1..1395b2c  main -> main

E:¥git¥p2¥CSC1003_temp>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Keep up to the newest

- You may want to commit changes on **device A** and work on **device B**
- Follow these steps to keep **device B** up to date:

1. Fetch the newest version
2. Dealing with conflicts (might not be any)
3. Pull the newest version

Fetch the newest version

- Use **git fetch** to know what's new on the remote side

```
E:¥git¥p1¥CSC1003_temp>git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 419 bytes | 10.00 KiB/s, done.
From https://github.com/Sababababababa/CSC1003_temp
   7b700c1..1395b2c  main       -> origin/main
```

Pull the newest version

- If no conflicts are found, we can simply update the local repo by **git pull**
- May ask for authentication

```
E:¥git¥p1¥CSC1003_temp>git pull
Updating 7b700c1..1395b2c
Fast-forward
 hw.java | 5 +++++
 hw.py   | 1 +
 2 files changed, 6 insertions(+)
 create mode 100644 hw.java
 create mode 100644 hw.py
```

```
E:¥git¥p1¥CSC1003_temp>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Dealing with conflicts

- It is possible that your work on **device A** is based on a earlier version, while the work on **device B** is already pushed.

```
E:¥git¥p1¥CSC1003_temp>git push
To https://github.com/Sababababaaba/CSC1003_temp.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/Sababababaaba/CSC1003_temp.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

- Try **git pull** and see if auto merge works
- If not, manually fix it with the help of **git diff**
  - **Diffs** are separated by ========
- After resolve it, do normal commit and push

```
E:¥git¥p1¥CSC1003_temp>git pull
Auto-merging hw.py
CONFLICT (content): Merge conflict in hw.py
Automatic merge failed; fix conflicts and then commit the result.

E:¥git¥p1¥CSC1003_temp>git diff
diff --cc hw.py
index f33eab3,2f217a8..0000000
--- a/hw.py
+++ b/hw.py
@@@ -1,1 -1,2 +1,6 @@@
- print('what¥'s up')
++<<<<<<< HEAD
++print('what¥'s up')
++=======
+ st='Hello World'
 -print(st)
++print(st)
++>>>>>>> d91544e49ece46ce9742fbb0cc7f6bcf066bf490
```

Ignoring some files

- You might want to ignore some files when synchronizing
- Use the .gitignore file

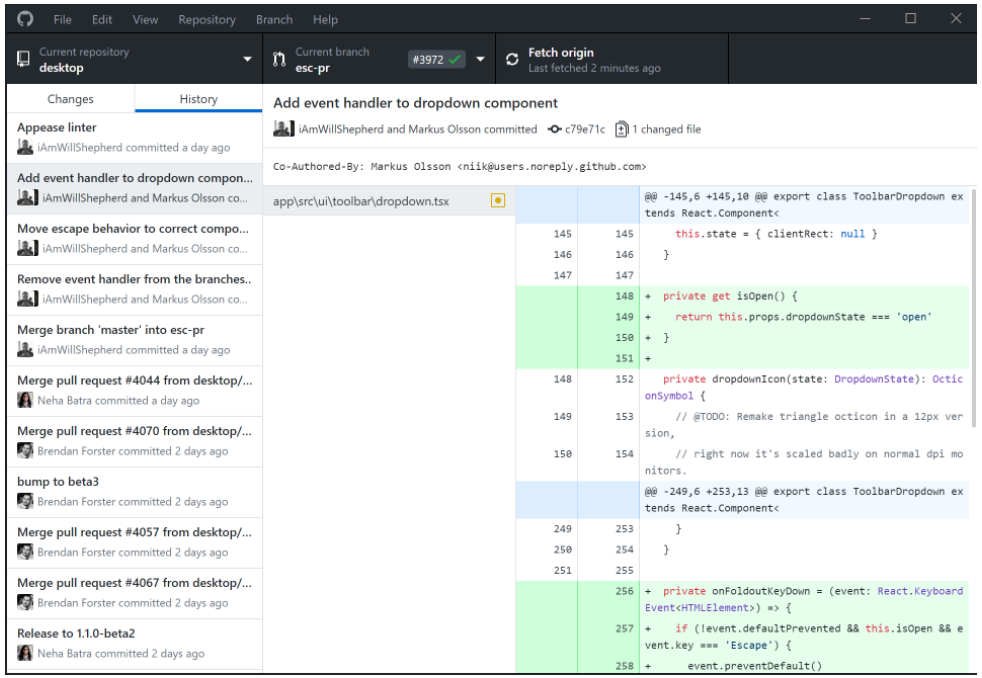
```
# CSC1003_temp
# ignore ALL .log files
*.log

# ignore ALL files in ANY directory named temp
packages/
```

- You can find good tutorials for writing this file
- In this course you may only need to know how to ignore **files** and **directories**

More user-friendly way

- You can also use github-desktop (though it is not elegant)
  - https://desktop.github.com/



  - Sometimes it produces mysterious bugs.