Notes

Sept.9th

- Wildcard usage

Example: use \* and find the place with such wording

>> cd \*541\*

Use: >>man cd

To get help with the cd function

Type *q* to exit the help page

Or --help (?)

To get in the specific path: use /

>>cd Desktop/SUBJECTS/…

>>cd ../Desktop

To go to the name with a space:



Absolute path: full address

Relative path: address starts from where we are

When should I use these two paths?

- use the one that’s easier!

Use git clone *link*

To clone the project in your own computer

Use git code file

To make changes to the file

Use git add file

To add the files

Check git status

Use git push

to upload new changes

see more push instructions on

<https://help.github.com/en/articles/pushing-commits-to-a-remote-repository>

Sep.11th

Why and when would you use the two different ways of adding and committing?

- if reviewing codes/more complicated, use local; if only simple edition/reviewing documents, use the online edition is more convenient.

Viewing history of the project:

- On Github through the repo’s code commit view

- On the local machine using git log

How similar are these two views? Which is easier?

- Easier to navigate in Github, and also review how the files look like in each commit.

Comparing files from different commits:

in Github

<https://github.ubc.ca/YOUR_USERNAME/YOUR_REPO_NAME/compare/shortSHA1..shortSHA2>

open this in the browser and the changes are shown.

In command lines

git diff SHA1 SHA2 file

Merge conflicts:

If two changes in two different places, and same lines are changed

If merged, the same line has been changed - work eliminated

- here is the merge conflict brought up

Sep. 16th

Use *echo* to key in lines

Use *code + file name* to open the coding for the file.

git log -- oneline

shows the changes of one line

Restore:

git restore -s SHORT\_SHA-1 Filename

remember to add and commit after restoration.

Create file called .gitignore:

code .gitignore

then type in the files want to ignore

save and add and commit (no push)  
check git status again -> the files we don’t want to track have disappeared!

small tip:

git add .

-> adds all files that have changed.

Create global git ignore

code .gitignore\_global



Tell git: once you find this, ignore it.

After saving:



(Have to provide the full path of the file name)

ls -a

to show the .gitignore file again

Get a copy of else’s repo

Use Fork at the top right -> to my github

clone it to my local device -> add/commit/push to the repo in my github

Catch up:

git add -u

git commit -m “…”

Intellisense/Contextual Help

use TAB to get information of the functions when typing.

Tiffany gives an absolute path of the working directory, so when we run it on our computer, the pwd is not correct.

R Markdown: use knit to see it in HTML

in Jupyter: click cell - cell toolbar - set to be slideshow