Git Tutorial
Sungho Hong

OCNC 2019

OIST

Plan

- Setting up git,

- Creating a repo,

- Committing changes made by adding, removing, and revising files,

- Reading the history of a repo and looking into past commits,

- Cloning someone else's repo, pulling changes, and resolving conflicts with your work,

- What to learn more, Q&A

Setting up Git

- Here we are mostly following "Version Control with Git" at https://swcarpentry.github.io/git-novice

- If you use git for the first time, set up git with your name and email address:

```
git config --global user.name "Here Your Name"
```

```
git config --global user.email "your.name@awesome.univ"
```

Creating a project

- To make a repo:

    mkdir my_project; cd my_project

    git init

    git status

    gitk

Adding files to a project

- Let's create a text file and add it to a repo:

```
echo "Hello, there" > file1.txt

git status  # check status

git add file1.txt

git status  # check status

git commit -m "file1.txt"
```

Adding files to a project

- Let's create a text file and add it to a repo (cont'd):

```
git commit -m "file1.txt" # ⟵ That was a bad message

git commit --amend -m "Add a file saying hello"

git log # (or gitk)
```

git log

```
ocnc@ocnc-VirtualBox:~/Documents/my_project$ git log
commit 6744f88472d01203ba01047c8a250cc0b2e6fb50 (HEAD -> master)
Author: Sungho Hong <shhong@oist.jp>
Date:   Tue Jun 25 10:37:26 2019 +0900

    Add the first file saying hello
```

- **SHA1** (6744f88472…) is an id of this commit and is useful if you need to access this commit.

  - In most cases, you don't need to type it all but only the beginning part.

  -

Exercise

- Add more files and create longer history.

Committing modified files

- Let's modify an existing file:

```
echo "And good afternoon and good night" >> file1.txt

git status

git diff
```

Committing modified files

- Let's modify an existing file and commit it:

```
git add files1.txt
```

```
git commit -m "Add more greetings in file1.txt"
```

```
git log # (or gitk)
```

Reviewing changes

- Let's modify an existing file and commit it:

git log

git show 6744:file1.txt # or git show 6744:file1.txt > file1_6744.txt

git diff 6744 file1.txt

Going back in time and coming back

```
git checkout 6744
```

```
git checkout master
```

Cloning a repo

```
cd ..

git clone my_project joe_project

cd joe_project

git status

git log
```

git log

```
commit 4ef8f1f5670c65c78f6593b6018a0d67208b2a13 (HEAD -> master, origin/master, origin/HEAD)
Author: Sungho Hong <shhong@oist.jp>
Date:   Tue Jun 25 10:59:10 2019 +0900

    Add more greetings in file1.txt

commit 6744f88472d01203ba01047c8a250cc0b2e6fb50
Author: Sungho Hong <shhong@oist.jp>
Date:   Tue Jun 25 10:37:26 2019 +0900

    Add the first file saying hello
```

git remote show origin

Pulling changes from a remote repo

- Let's go back to "origin" and make and commit changes:

  `cd ../my_project`

- … (making a change and commit it) …

  `echo "Tutorial status: Going well" > status.txt`

  `git add status.txt; git commit –m "Add a file for recording status"`

  `git log`

Pulling changes from a remote repo

- Let's go to the cloned repo and pull changes:

  `cd ../joe_project`

  `git pull origin master`

Inspecting changes before pulling them

- If you want to be more careful and avoid pulling changes immediately:

  git fetch origin master

  git diff origin/master

  git merge origin/master `# or again, git pull origin master`

Conflicts

- Let's change and commit status.txt in both places:

```
cd ../my_project

echo "Tutorial Status: Another Error Occured" > status.txt

git add .; git commit -m "Update the status"



cd ../joe_project

echo "Tutorial Status: Going Well" > status.txt

git add .; git commit -m "Update the status"
```

git pull origin master

```
ocnc@ocnc-VirtualBox:~/Documents/my_project2$ git pull origin master
From /home/ocnc/Documents/my_project
 * branch               master      -> FETCH_HEAD
Auto-merging status.txt
CONFLICT (content): Merge conflict in status.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Resolving conflitcs #1: Choose ours over theirs

```
git merge --abort


git pull origin master -X ours


# …Editing the commit message…


git log


more status.txt
```
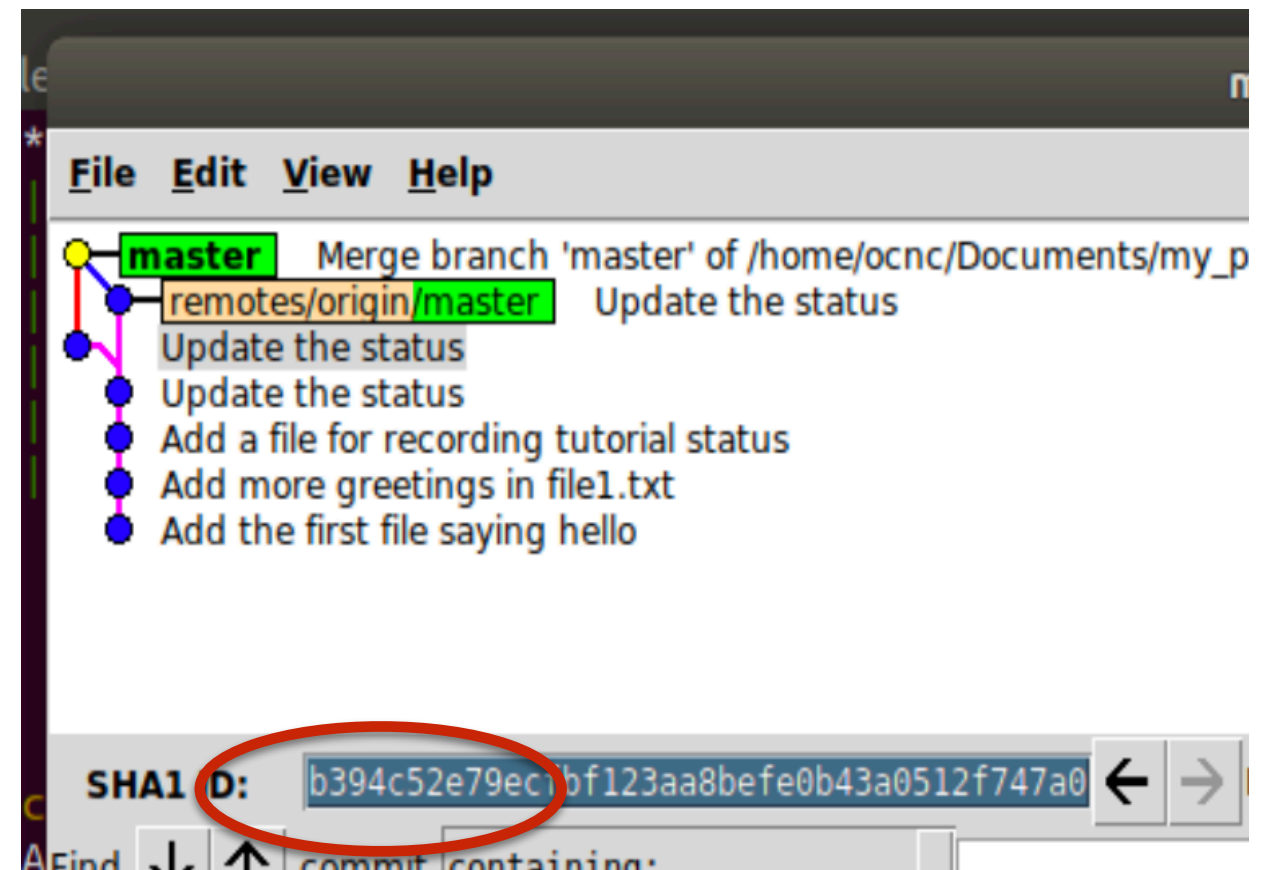
HARD RESET!

- Now you don't like what you just did and want to *reset to a previous commit*:

`gitk # or git log --graph`

`git reset --hard b394`

`more status.txt`

`git log`

Resolving conflitcs #2: Choose theirs over ours

```
git pull origin master -X theirs


# …Editing the commit message…


git log


more status.txt
```

Resolving conflitcs #3: Editing conflicts

```
git reset --hard b394


# … Editing the conflicted file …


git status


git add status.txt


git merge --continue
```

Pushing your changes to a remote

- If a remote repo is where you are *allowed* to push your changes, you can update the remote:

```
git push origin master
```

Cloning a repo from github

git clone http://github.com/shhong/ocnc2019_git_tutorial

github way of contributing to a project

- "Fork" the project repo.

- Clone your fork and work on it, which you can push to your fork.

- Send a "pull request" to ask meging of your work into their repo.

Important topics omitted

- GUIs

  - Sourcetree, GitKraken,…; Visual Studio Code, Atom,…

- Branching and merging: "git branch" (https://learngitbranching.js.org)

  - Rebasing: "git rebase"

- git blame

- Large binary data

- More on using github

Questions?