

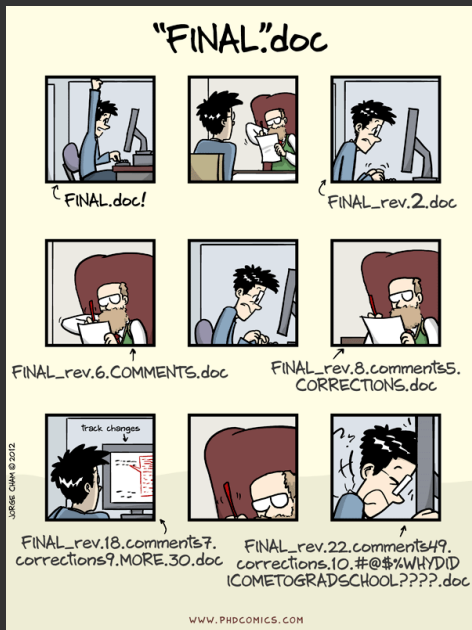
# >>> Git Workshop I: Git the Big Picture!

Name: Paul Z Cheng (Tzu-Yu Hsu Lab)

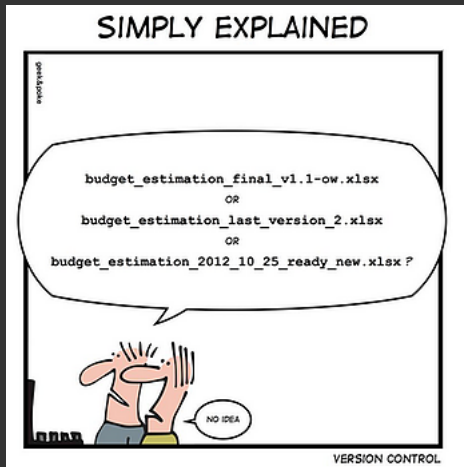
Date: April 12, 2021



# >>> Why use a Version Control System (VCS)?



>>> What is VCS?



## >>> Reproducible and Open Science

- Trend of science's inevitable shift toward "Open Science."

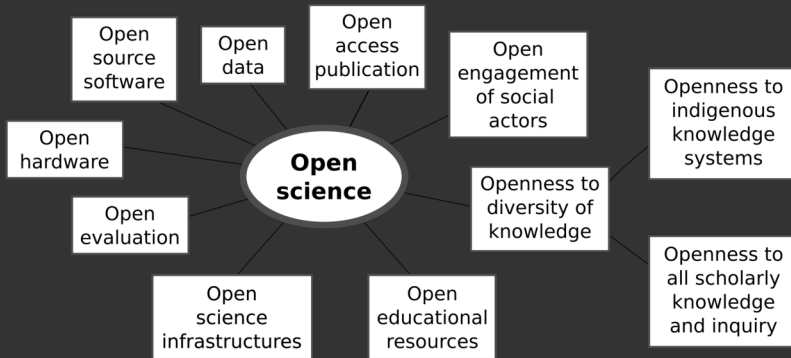
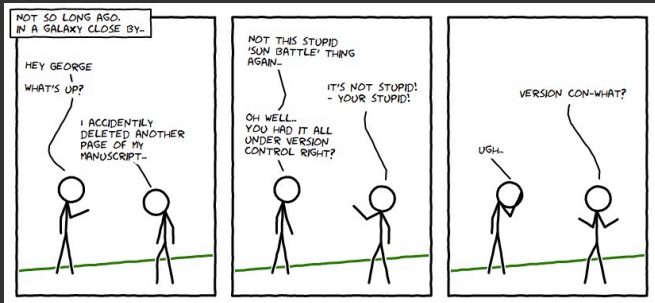


Figure: UNESCO Presentation on Open Science 2021

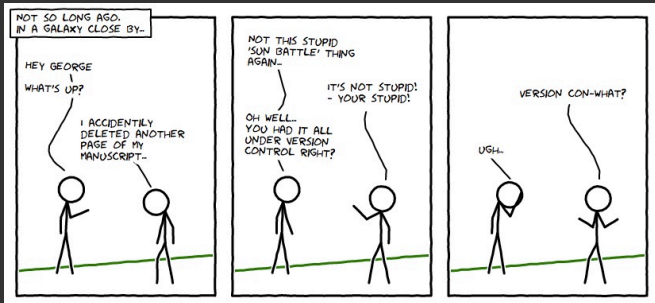
## >>> Feature of a VCS?

1. Save different version of your files in a safe and organized way.



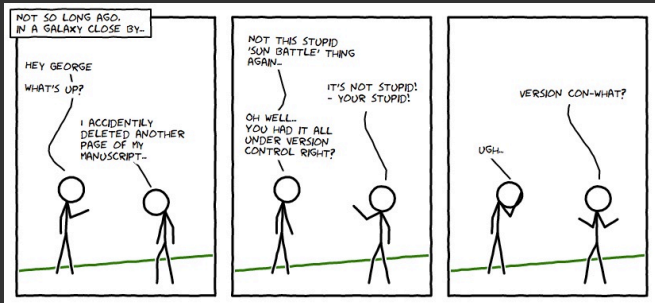
## >>> Feature of a VCS?

1. Save different version of your files in a safe and organized way.
2. Having flexibility to edited and debugged between versions.



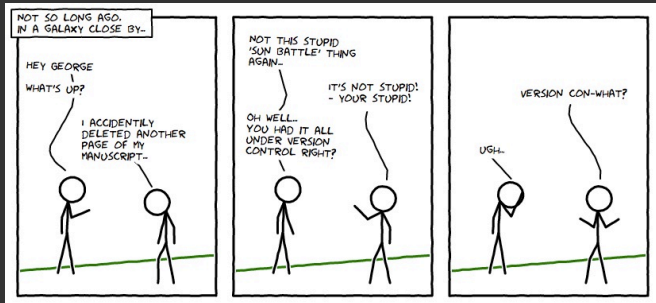
## >>> Feature of a VCS?

1. Save different version of your files in a safe and organized way.
2. Having flexibility to edited and debugged between versions.
3. Easy for collaborator to work on the same project.



## >>> Feature of a VCS?

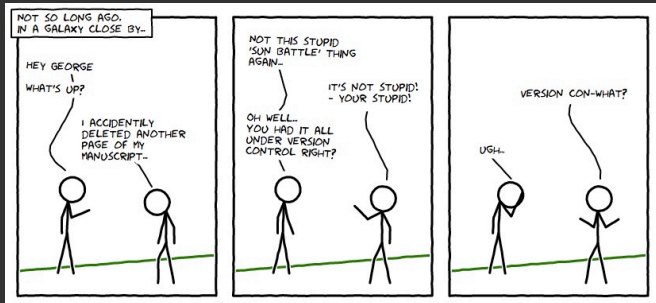
1. Save different version of your files in a safe and organized way.
2. Having flexibility to edited and debugged between versions.
3. Easy for collaborator to work on the same project.
4. Act as a backup.





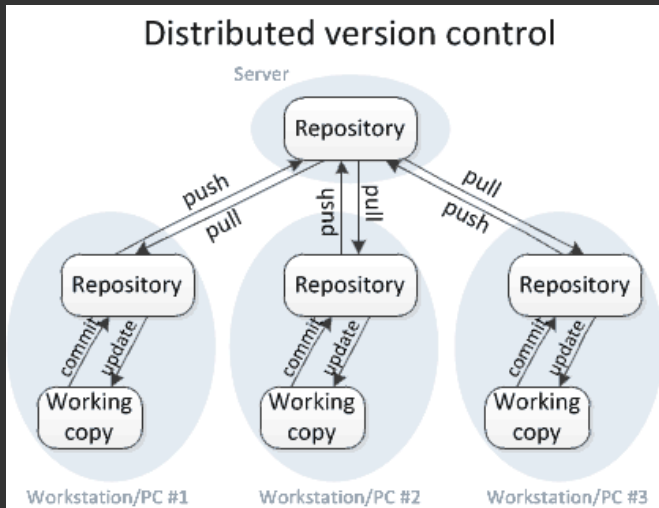
## >>> Feature of a VCS?

1. Save different version of your files in a safe and organized way.
2. Having flexibility to edited and debugged between versions.
3. Easy for collaborator to work on the same project.
4. Act as a backup.
5. Induce effort of sharing and Open science!!



>>> What is Git?

- A powerful distributed revision control system.



>>> What do you git?

Git is easy to use but hard to learn



## >>> Git installation

#For this workshop, we will use the 2.20 and above versions.

#All installation reference is from git main web

#<<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

# Before install go to the command line

git --version

# mac using Homebrew

brew install git

# for more: <https://git-scm.com/download/mac>

# Linux system (Debian or Ubuntu)

sudo apt-get install git-all

# Windows

# <https://git-scm.com/download/win>

# follow gui to set it up

>>> Github Signup

<https://education.github.com/pack> for current TMU student.  
checkout and fork: <https://github.com/paul30402/gitasktic>



Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

**Sign up for GitHub**

>>> **BASH: before began**

# Basic of bash

# change directory

pwd # current directory (cwd for windows)

cd # change current directory to home directory

cd <path> # change current directory to path

cd .. # go up a level

# check out directory items

ls

ls -lrt # list the directories align update timeline

ls -a # check hidden directories i.e. .git ..etc

# Make directory

mkdir # Make a directory

# Create a file

touch <file name>

# checkout [https://tldp.org/LDP/intro-linux/html/sect\\_03\\_03.htm](https://tldp.org/LDP/intro-linux/html/sect_03_03.htm)

[~]\$ \_

[10/18]

```
>>> Git: initiate and Configured
```

```
# git start. note: after travel into your directory  
git init
```

```
# Configure work stations owner of repository  
git config --global user.name "Paul Z. Cheng"  
git config --global user.email paul.z.cheng@gmail.com  
git config --global core.editor atom
```

```
>>> Git: initiate your repository
```

```
# Start a new repository
```

```
# git init set up
```

```
cd to/your/favorite/directory
```

```
git init -b <name of your choosing>
```

```
# if not specified, init branch will be "master."
```

```
# ..... some awesome code session later
```

```
git add .
```

```
git commit -m "First commit of greatness."
```

```
# git remote setup
```

```
git remote add origin <Remote git URL>
```

```
# check the remote setup
```

```
git remote -v
```

```
# push to origin
```

```
git push -u origin master
```



```
>>> Git: clone a repository
```

```
## Clone a directory
```

```
git clone < git URL >
```

```
# ..... some awesome code session later and as above
```

```
# git remote setup
```

```
git remote add origin <Remote URL>
```

```
# check the remote setup
```

```
git remote -v
```

```
# push to origin
```

```
git push -u origin master
```

```
>>> Branching
```

```
# Check out your current listed branches
```

```
git branch
```

```
# make a branch out of the current working area
```

```
git branch new_master
```

```
#switch branches
```

```
git checkout master
```

```
# make and switch
```

```
git checkout -b master_3
```

```
# check the difference between two state
```

```
git diff "branch 1" "branch 2"
```

```
>>> Git: Daily save and editing
```

```
## some awesome coding and editing of your working area later  
# check working area has any need to update  
git status
```

```
# ADD  
git add "file names" # one file  
git add .             # everything that has been changed
```

```
# Commit related to "document your add"  
git commit -m "xx files update"  
# change message  
git commit -amend -m "change"
```

```
# check out commits  
git log
```

>>> Git: Daily short hand

```
# Alias: simplified commands  
git config --global alias.co checkout  
git config --global alias.br branch  
git config --global alias.st status
```

>>> Your Journey Began!!

1. Definition of VCS
2. Git as distributed VCS
3. Workshop objective
4. Installing and basic git commands



>>> Hands on GIT ON!

1. Forked course GitHub directory
2. Clone and setup working git repository on your computer as *master*
3. Create a new branch call *tinker* from your current working directory
4. Check out the folder and try to run the ODDBALL experiment on your computer(install psychopy if you haven't)
5. Lookup git merge and merge *tinker* to *master*
6. Upload your version of *master* to GitHub