



The Carpentries' Version Control with Git with Denisse and Lisa

DATA TAS

TASMANIAN DATA COMMUNITY



Our Code of Conduct

We want to create an inclusive and welcoming place for everyone to learn.

We ask you to:

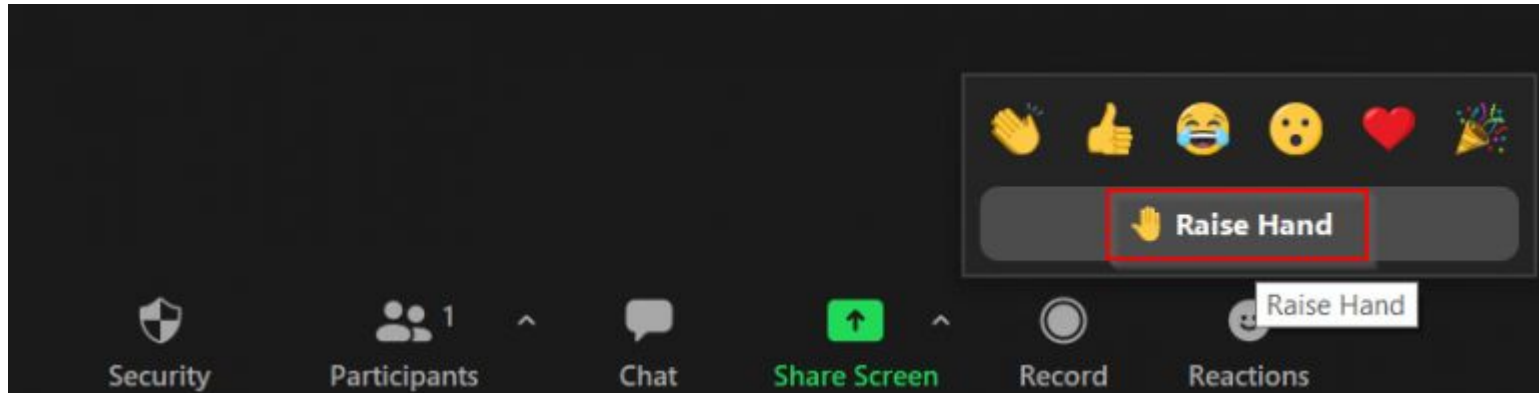
- Treat everyone with respect
- Do not interrupt others
- Get in touch if you feel uncomfortable



Want to ask a question?

Raise your hand! It does not matter if you are in-person or online.

Feeling a little shy? Send us a message on Zoom.





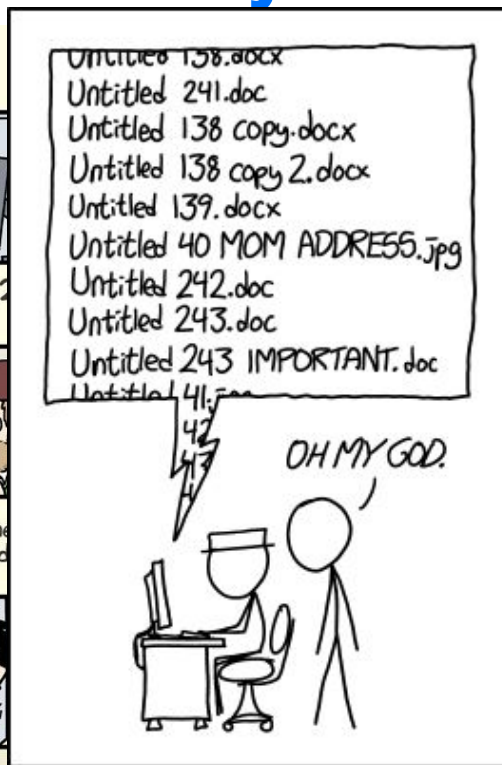
Before starting

1. Check our workshop site:
<https://lidefi87.github.io/2022-03-23-version-control/>
2. Complete the Pre-workshop survey
3. Our schedule
 - 14:00 - 15:05 - Lessons 1 to 5
 - 15:05 - 15:10 - Short break
 - 15:10 - 16:00 - Lessons 6 to 7

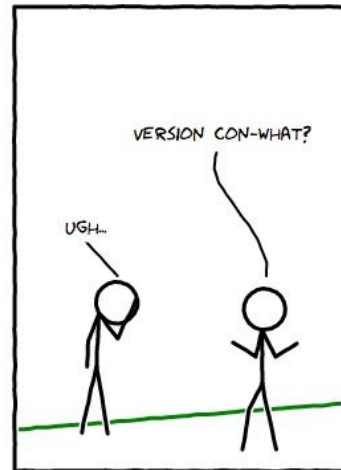
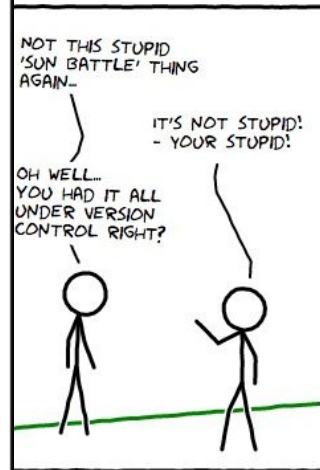
You can find the lessons here:

<https://swcarpentry.github.io/git-novice/>

Version control - Why bother?



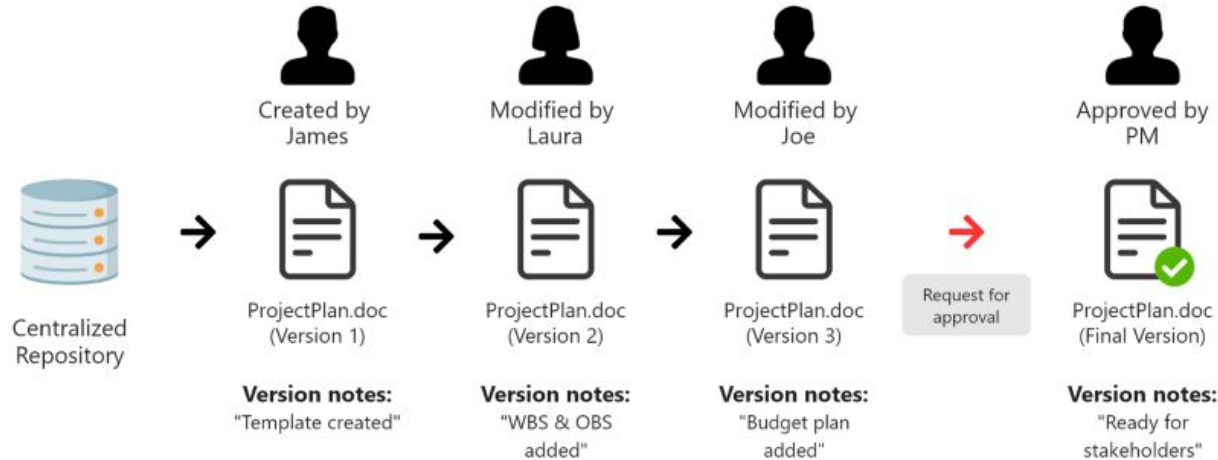
PROTIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.





Version control - Why bother?

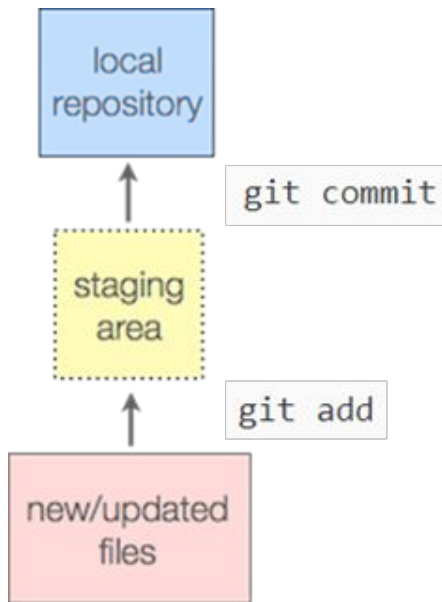
Document Version Control Flow





First steps with Git

1. Introduce yourself to Git and customise it (only once)
 - `git config`
2. Start a repository
 - `git init`
3. Track your files
 - `git add`
4. Save your changes
 - `git commit`

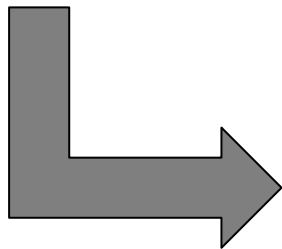




Committing changes to Git

Git is only as useful as the information you provide.

Avoid falling on this trap



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

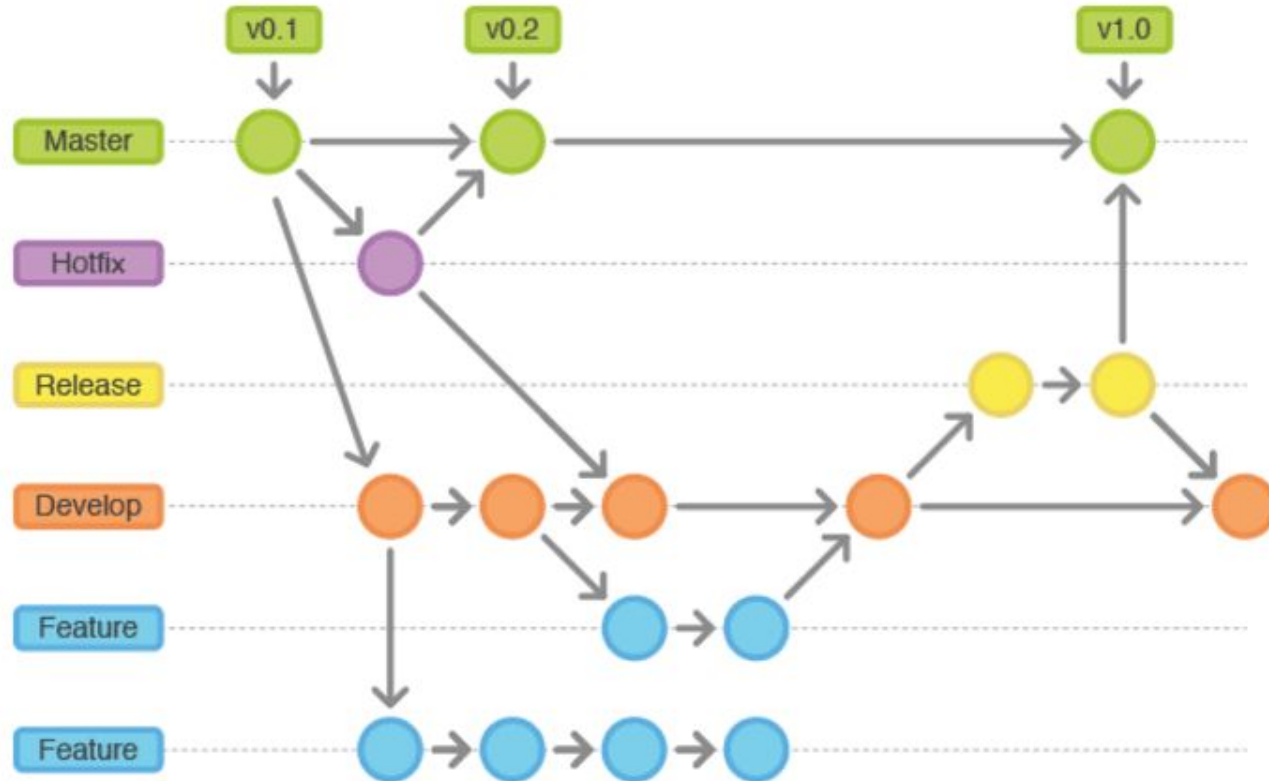


Useful Git commands

1. Check status of your repository
 - `git status`
2. Check differences between commits
 - `git diff`
3. Discarding changes/changing branches/reviewing commits
 - `git checkout`
4. Reverting repository to a previous version
 - `git revert`



Collaborating with Git - Branches





Useful Git commands

1. Check status of your repository
 - `git status`
2. Check differences between commits
 - `git diff`
3. Discarding changes/changing branches/reviewing commits
 - `git checkout`
4. Reverting repository to a previous version
 - `git reset`



Day 2 - Version Control with Git



Undoing changes in Git

We have seen the following commands

- `git checkout`
- `git reset`
- `git revert`

A nice explanation about their differences can be found [here](#).



Connecting GitHub and Git

There are two ways to connect with GitHub

1. Create a personal access token (PAT)
 - [Instructions](#)
2. Set up SSH keys
 - [Instructions](#)

GitHub no longer allows users to use passwords only.



Connecting GitHub and Git

After creating a PAT or SSH key, you can connect your local repository to the cloud

1. Connect remote repositories to your computer
 - `git remote add BRANCH_NAME LINK_GITHUB`
2. Send (push) changes to GitHub
 - `git push REMOTE_BRANCH LOCAL_BRANCH`
3. Receive (pull) changes from GitHub
 - `git pull REMOTE_BRANCH LOCAL_BRANCH`



Copying a remote repository locally

1. Copy (clone) a repository from Github to your computer
 - `git clone LINK_GITHUB`
2. Send (push) changes to GitHub
 - `git push REMOTE_BRANCH LOCAL_BRANCH`
3. Receive (pull) changes from GitHub
 - `git pull REMOTE_BRANCH LOCAL_BRANCH`



What if I do not want to use GitHub?

Some alternatives available include:

1. GitLab: <https://gitlab.com/>
2. Gitbucket: <https://bitbucket.org/>
3. SourceForge: <https://sourceforge.net/>
4. Gitea: <https://gitea.io/en-us/>



Do I need to code to use Git and GitHub?

Not at all. There are several GUIs available now:

1. The Git website offers an extensive list of options:
<https://git-scm.com/downloads/guis>
2. You can also connect to GitHub from:
 - [RStudio](#)
 - [Visual Code](#)
 - [Atom](#)



Some extra references

1. A Quick Introduction to Version Control with Git and GitHub:
<https://doi.org/10.1371/journal.pcbi.1004668>
2. Ten Simple Rules for Taking Advantage of Git and GitHub:
<https://doi.org/10.1371/journal.pcbi.1004947>
3. When it comes to reproducible science, Git is code for success. And the key to its popularity is the online repository and social network, GitHub:
<https://www.natureindex.com/news-blog/when-it-comes-to-reproducible-science-git-is-code-for-success>



We are done!
Thanks for joining us and happy Git!



Follow us!



www.datatas.com



[datatasau](https://www.facebook.com/datatasau)



[@datatas](https://twitter.com/datatas)



[datatas](https://github.com/datatas)



DATATAS

TASMANIAN DATA COMMUNITY