Intro to Git and GitHub

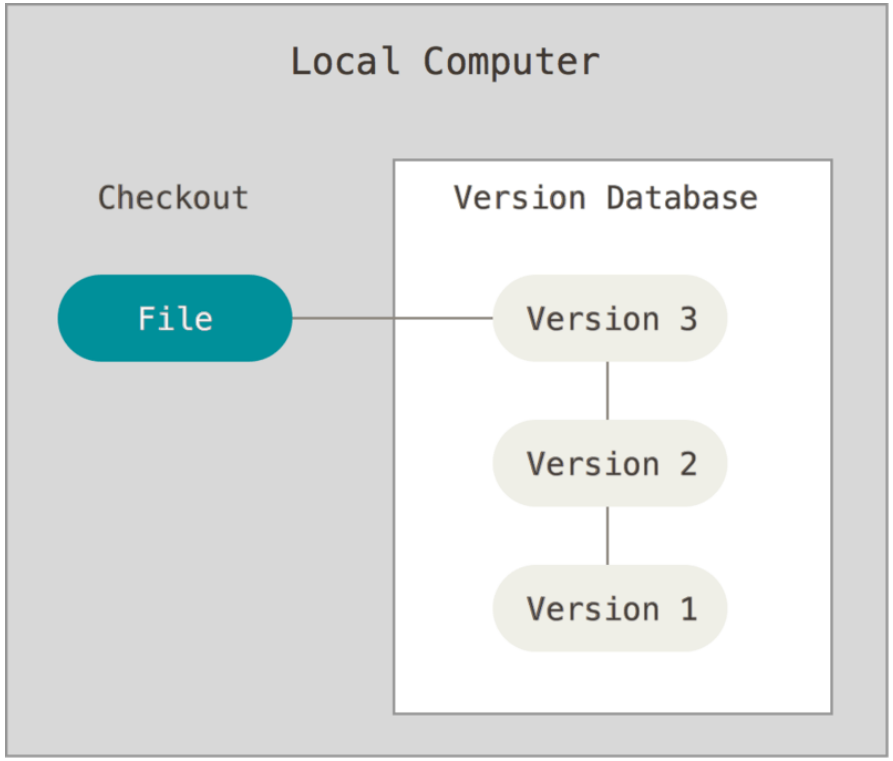
Rory Crean
Researcher
Department of Chemistry - BMC
Uppsala University

9th January 2024

UPPSALA UNIVERSITET

Part 1: Git

Git is a version control system used to track changes



Example local version control diagram,
Taken from: Pro Git book

Advantages over "manual" version control:

- More automated and easier to use.
- More space efficient.
- Much less likely for user error.
- No need to write files like: "final_version3_draft_V3.py"

You start by defining a folder for Git to Monitor

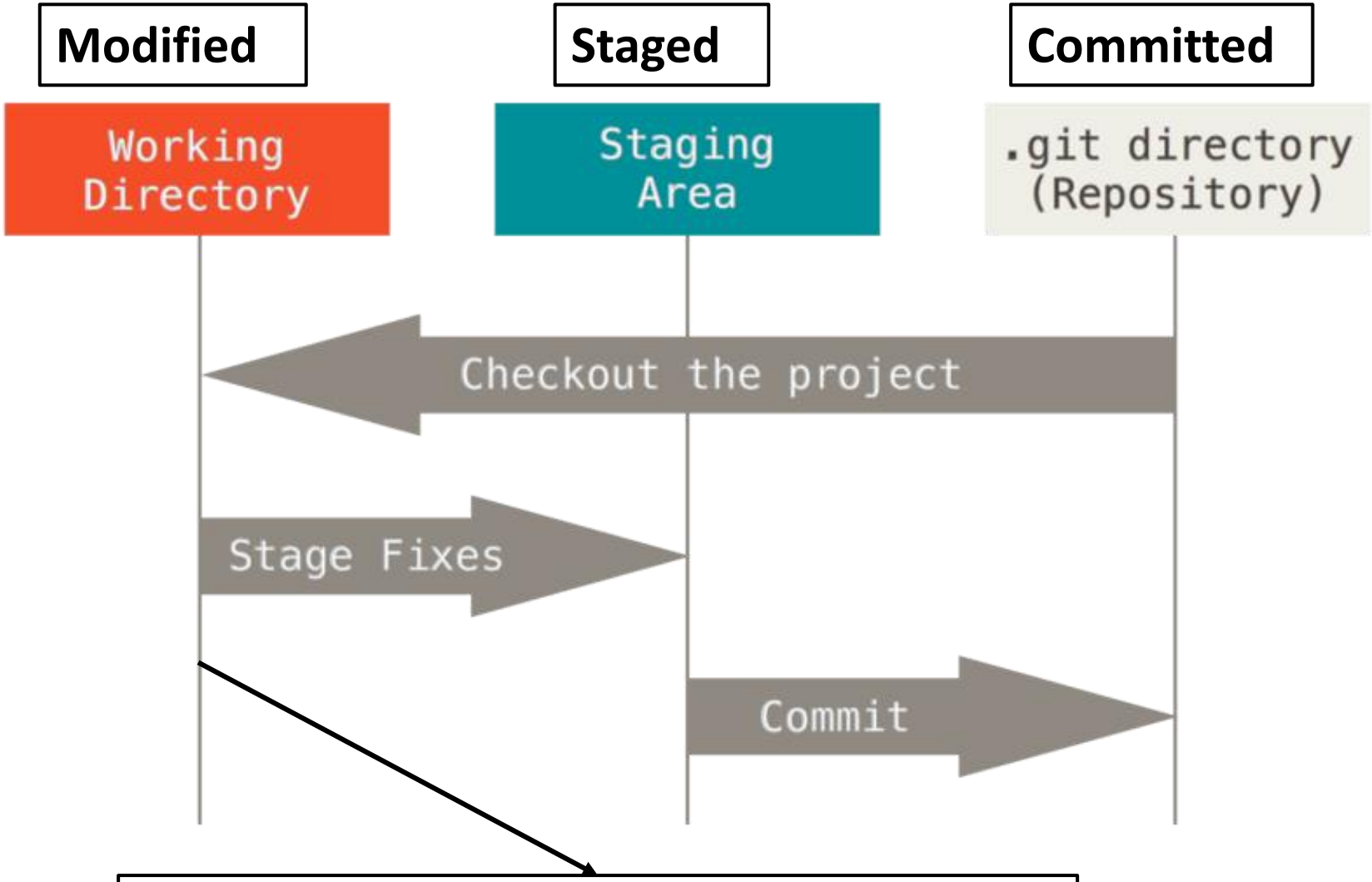
- New project, new folder.
- Store each project/folder in the same general place.
- Don't have spaces in the file path.
- If you use dropbox/onedrive, don't track this location.

My Setup

```
(base) roryc760@UUC-HLFRGY3:~/projects$ pwd
/home/roryc760/projects
(base) roryc760@UUC-HLFRGY3:~/projects$ ls
KIF                  bmc-git-and-github-tutorial      protein-interaction-stats
arjan_codes_course   event-driven-chess               stable-proteins
basel_workshop       kin-gui                          test-repo
bayesian_allostery   practical-python-for-scientists  tools-project
(base) roryc760@UUC-HLFRGY3:~/projects$
```

Each folder above has it's own git repository

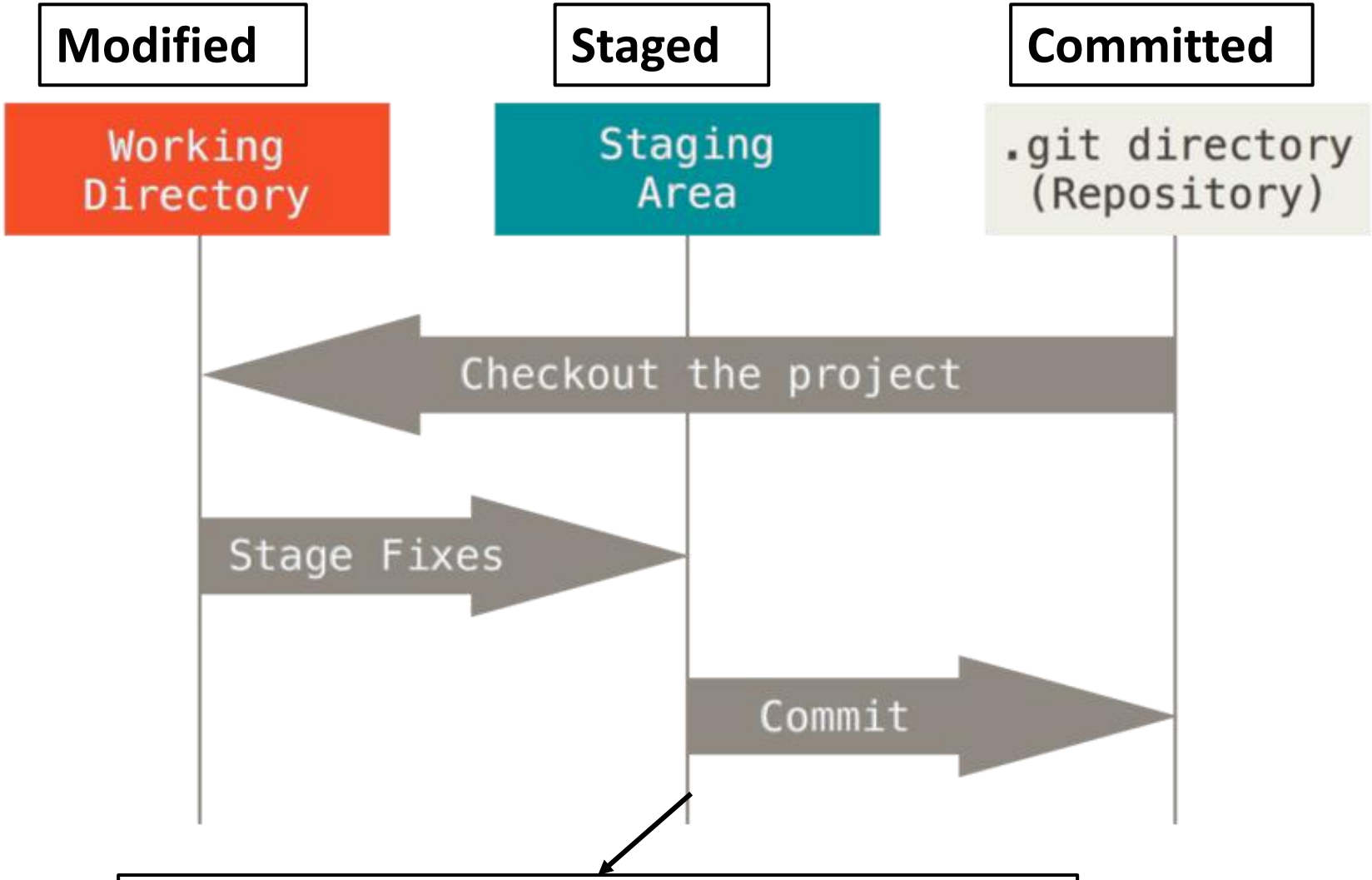
The three states of a file in Git



Git has no record of this file, if you remove it now, Git will never know.

Image taken from: Pro Git book

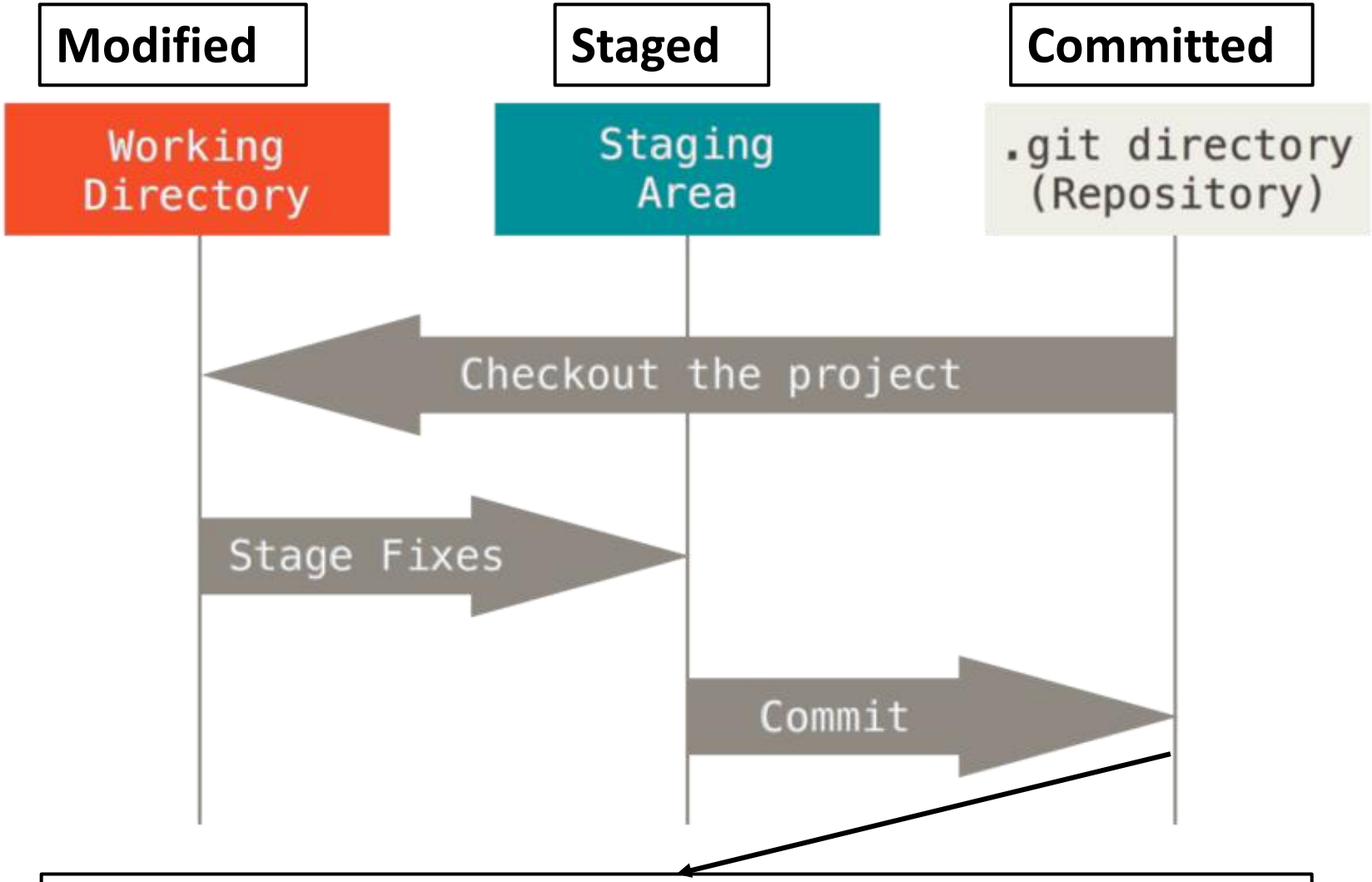
The three states of a file in Git



Use ”git add [file_name]” to move a file to the staging area.

Image taken from: [Pro Git book]

The three states of a file in Git



Use "git commit" to store a new version of the project. Changes in the staging area are used in this commit.

Image taken from: Pro Git book

Tools/IDEs Can Help You Make Use of Git

GitHub Desktop

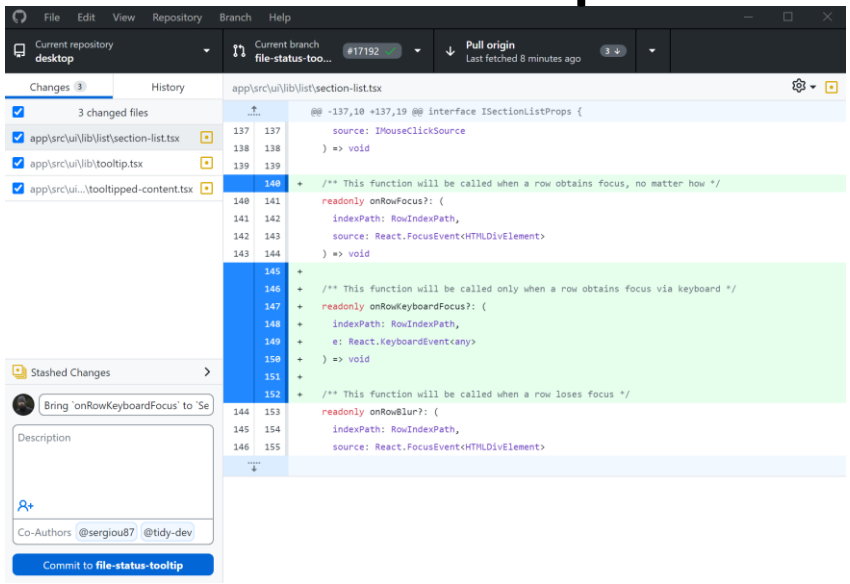


Image taken from: desktop.github.com

GitKraken

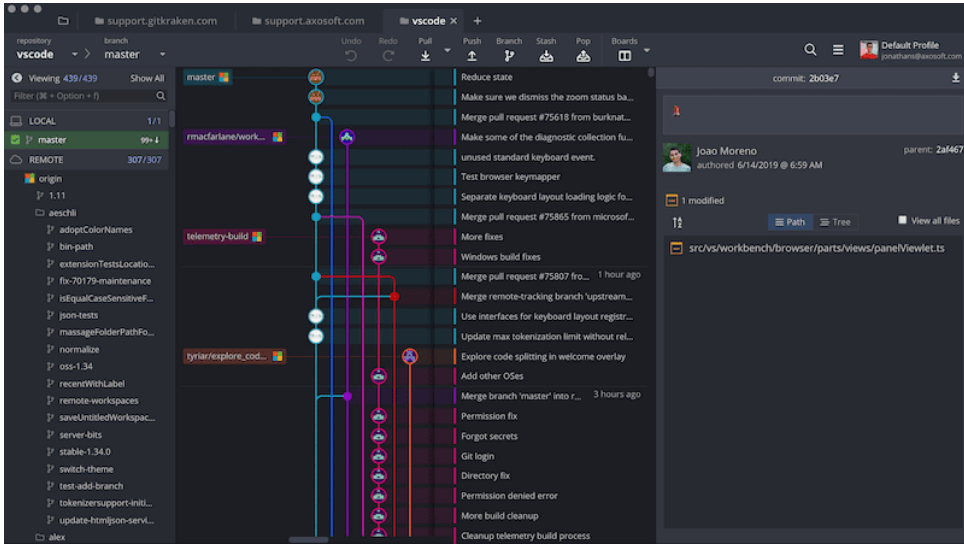
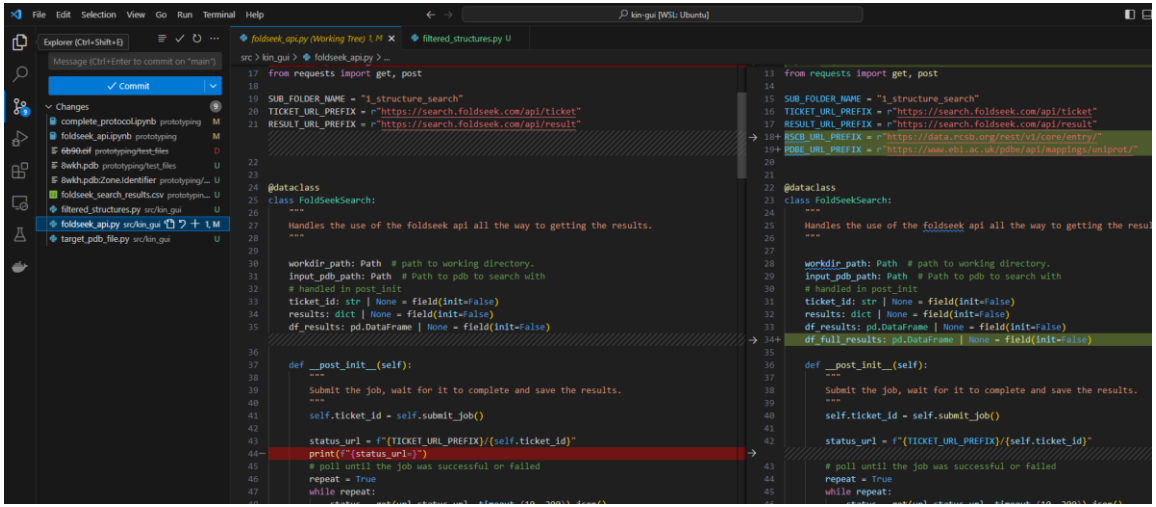


Image taken from: gitkraken.com

VSCode



Hands on Session 1:

Please go to:
https://rmcrean.github.io/bmc-git-and-github-tutorial/
or:
https://github.com/RMCrean/bmc-git-and-github-tutorial (and then click on the link on the left handside.)

Part 2: GitHub and Git Combined

The difference between Git and GitHub

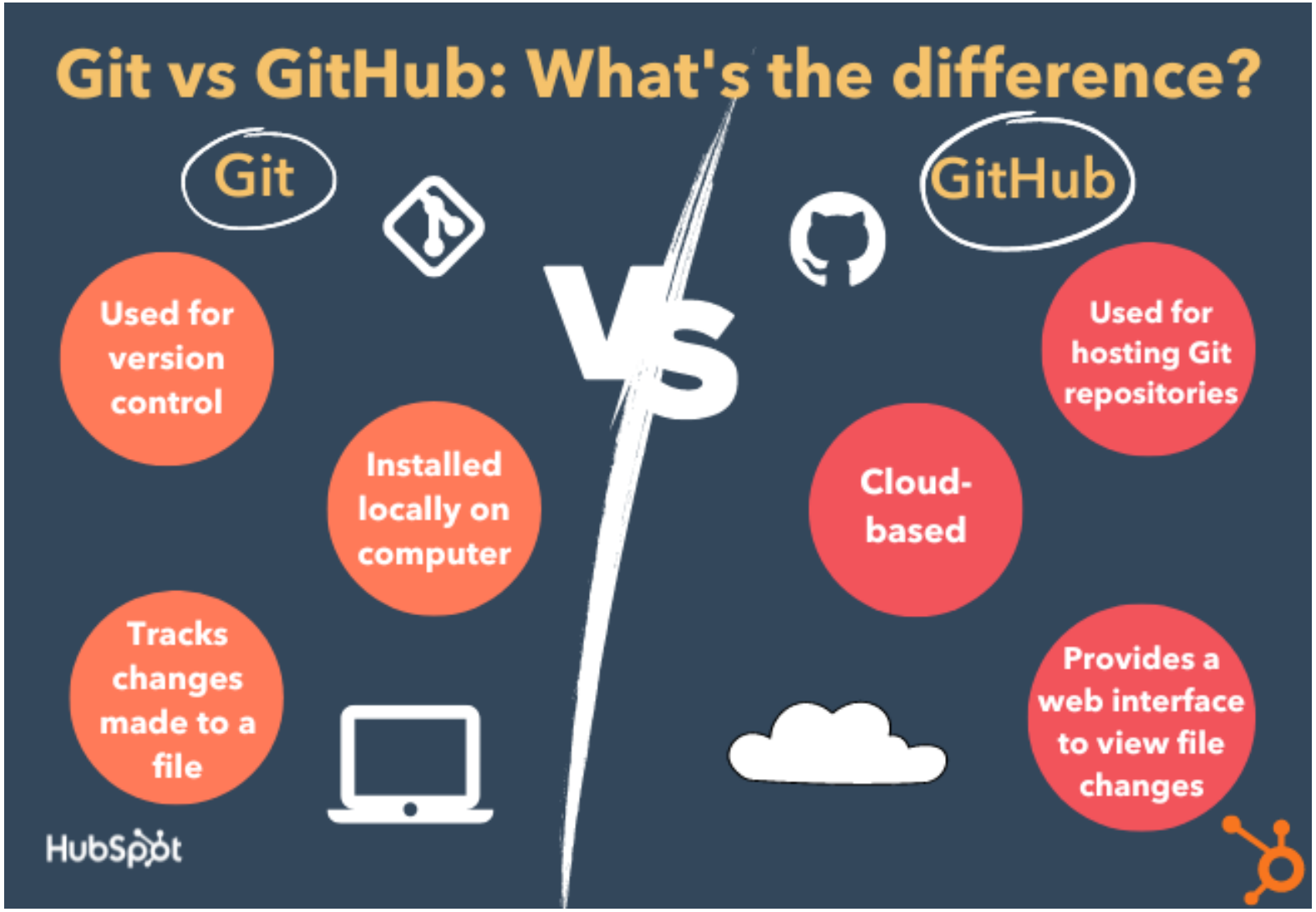
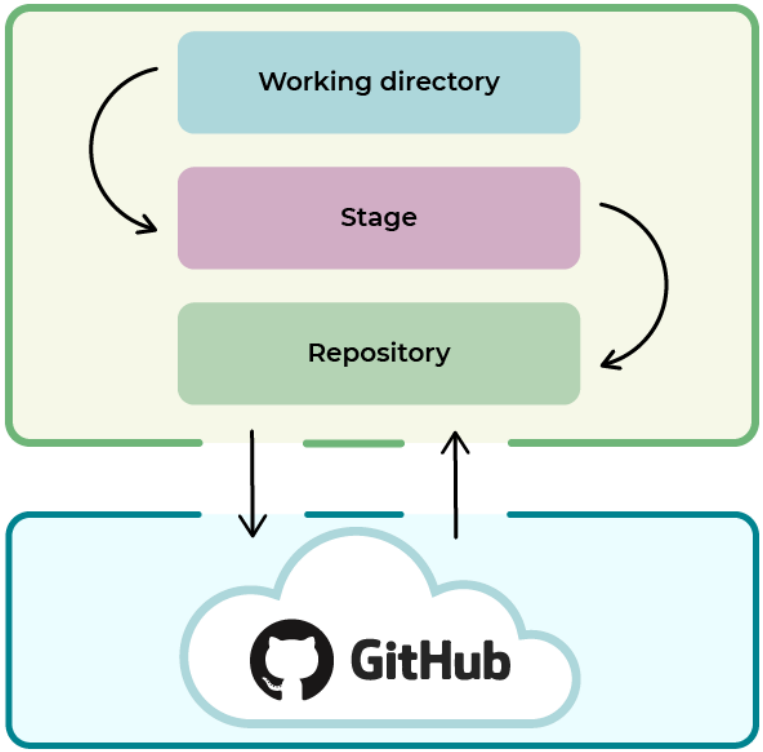


Image taken from: https://blog.hubspot.com/website/git-vs-github

GitHub is a place to store/host remote Repositories

- You can have several versions of the same project, this can be useful both working alone or in a team.



Local repository

Local means it's on your pc (or someone else's pc, in which case that version is local for them).

Remote repository

Remote means you need to connect to it (e.g. hosted on a server).

Image taken from [openclassrooms]

<u>Why Use a Remote:</u>

- Back up your own work.
- To collaborate with other people.
- Share your work.

Not everything should be uploaded to GitHub

Example of things you should not add:

- Large datasets.
- Sensitive/Personal data.
- Passwords/usernames.
- System-specific files, e.g. .DS_Store on a Mac.

How to do this:

- Use a ".gitignore" file and add to it as you need.
- **You should commit your .gitignore file.**

- Use a ".gitignore" template file designed for your programming language.

- Be careful about using "git add ."

.gitignore

practical-python-for-scientists >

```
# Byte-compiled / opt
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / pack
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
```

More Git Vocabulary: Push and Pull

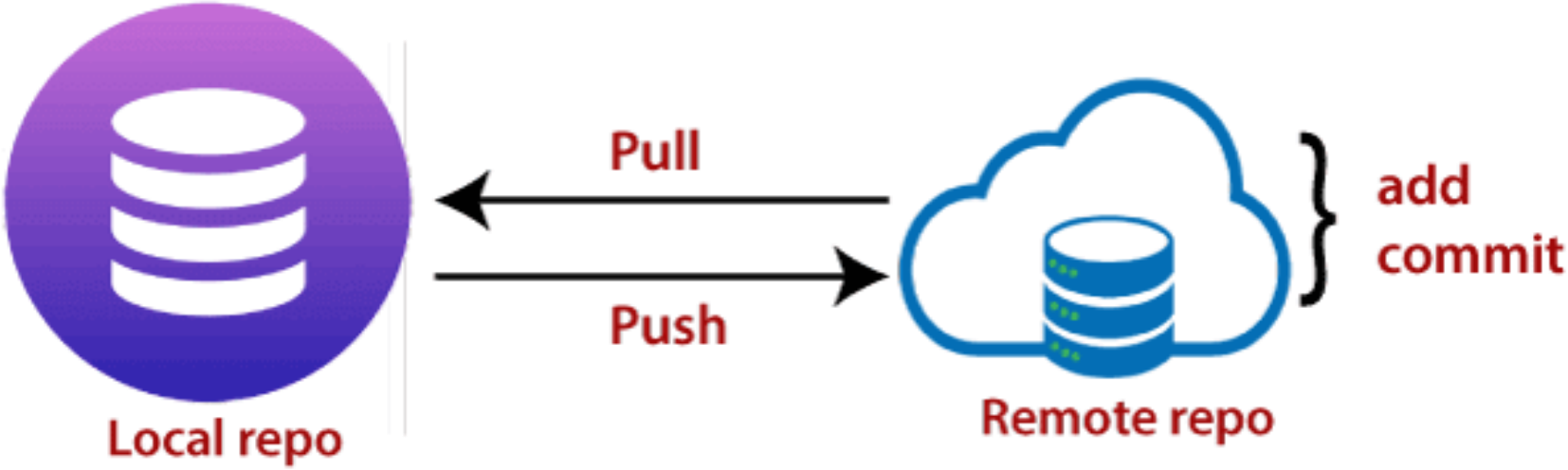


Image taken from: https://www.javatpoint.com/git-push

- "*git push*" – Update local commits to the remote repo.
- "*git pull*" – Get remote commits from your pc to remote repo

And one more:

- "git clone" – Make a local copy of a remote repo.

Hands on Session 2:

Please go to:
https://rmcrean.github.io/bmc-git-and-github-tutorial/
or:
https://github.com/RMCrean/bmc-git-and-github-tutorial (and then click on the link on the left handside.)

Part 3: Branches and Merging

Branches in Git

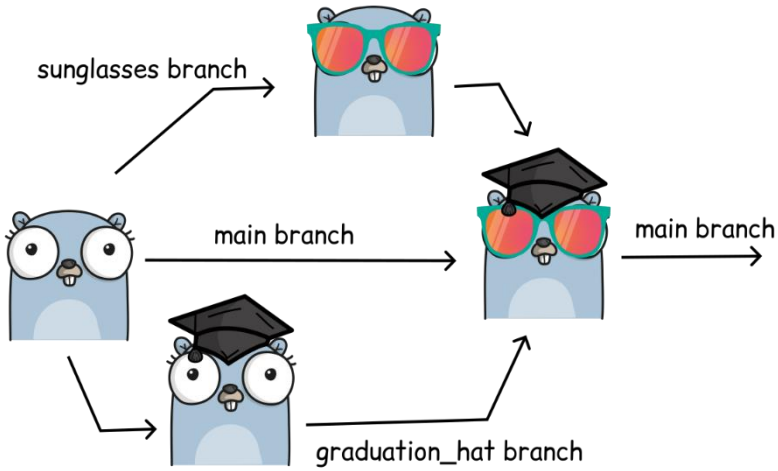


Image taken from: https://coderefinery.github.io/git-intro/branches/

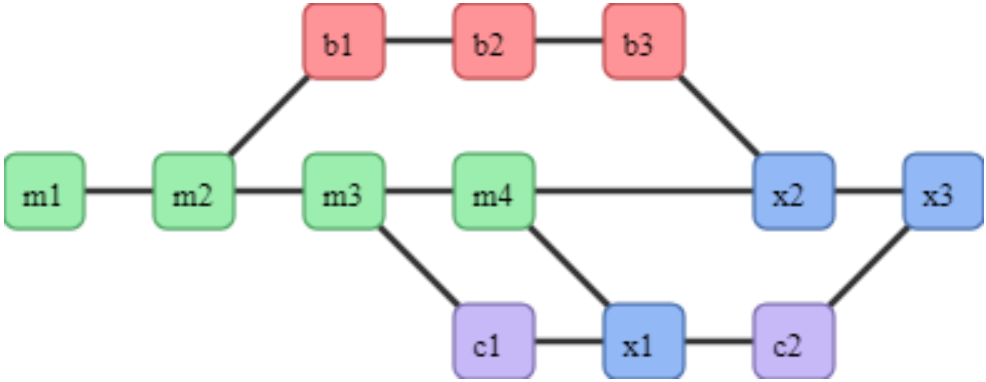


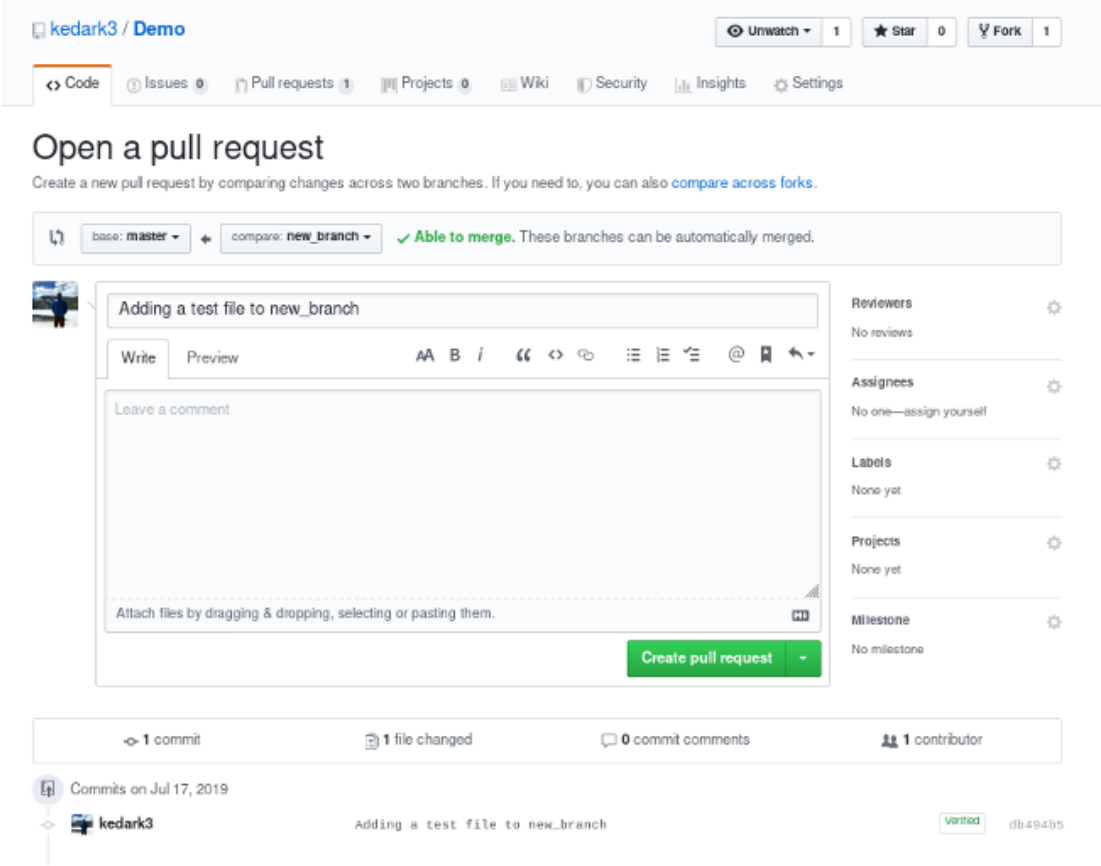
Image taken from: https://coderefinery.github.io/git-intro/branches/

- Branches allow us to separate out different blocks of work.
- Once we're happy with the changes on the branch, we want to **merge** the changes (commits) back onto the main branch.
- If working alone, you can *probably* get away with not using branches.

Merging two branches can be done with either Git or GitHub

Rough Protocol:

1. Make new branch.
2. Add changes to branch
3. Push changes to GitHub
4. Follow Instructions on GitHub



Hands on Session 3:

Please go to:
https://rmcrean.github.io/bmc-git-and-github-tutorial/
or:
https://github.com/RMCrean/bmc-git-and-github-tutorial (and then click on the link on the left handside.)

Summary

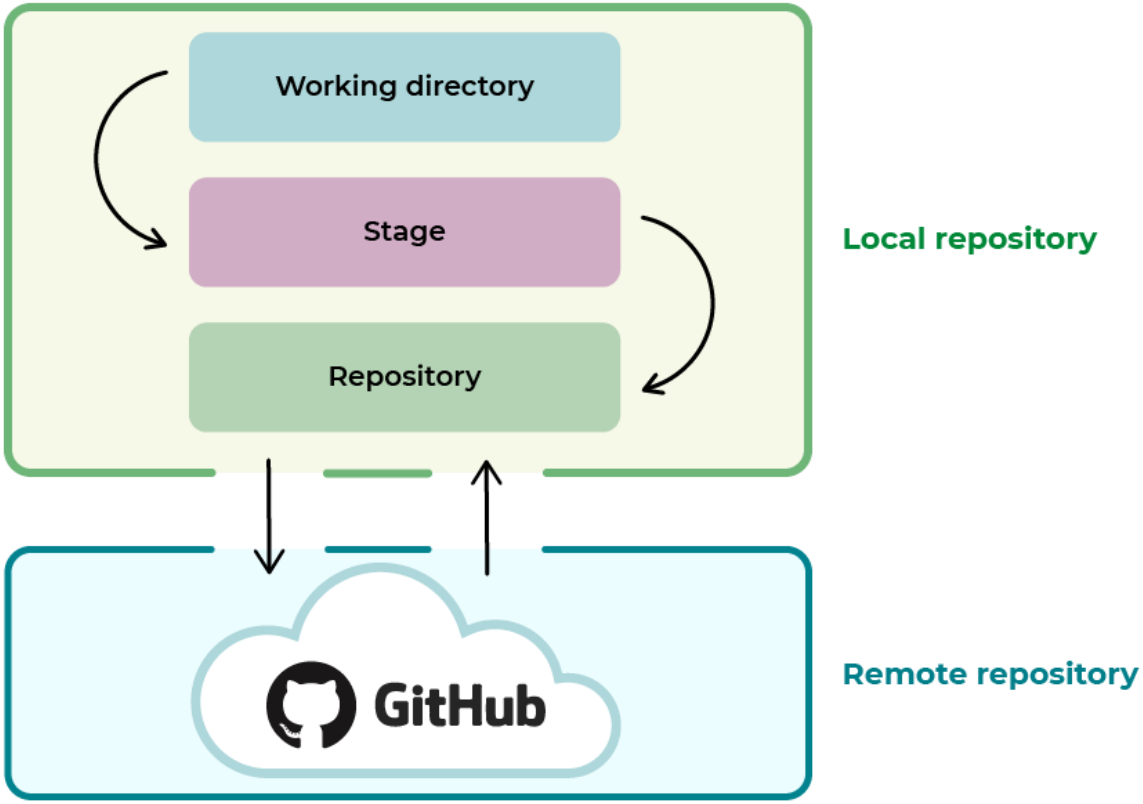


Image taken from [openclassrooms]

- It's easier to keep things simple, especially while learning in the beginning.