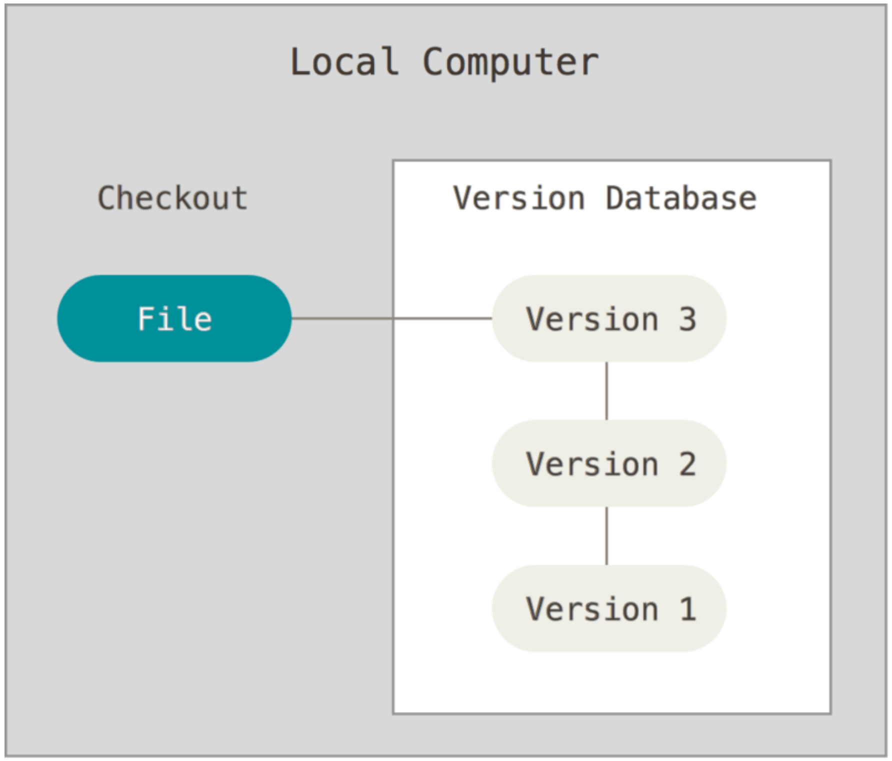
Intro to Git and GitHub

Rory Crean
Researcher
Department of Chemistry - BMC
Uppsala University

18th December 2023

Part 1: Git

Git is a version control system used to track changes



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

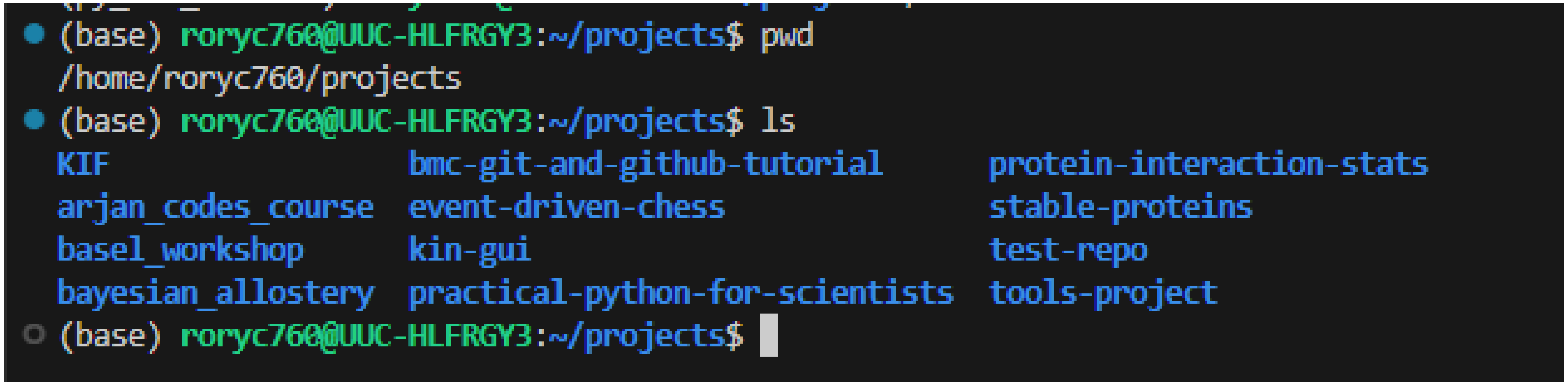
<u>Advantages over "manual" version control:</u>

- 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.

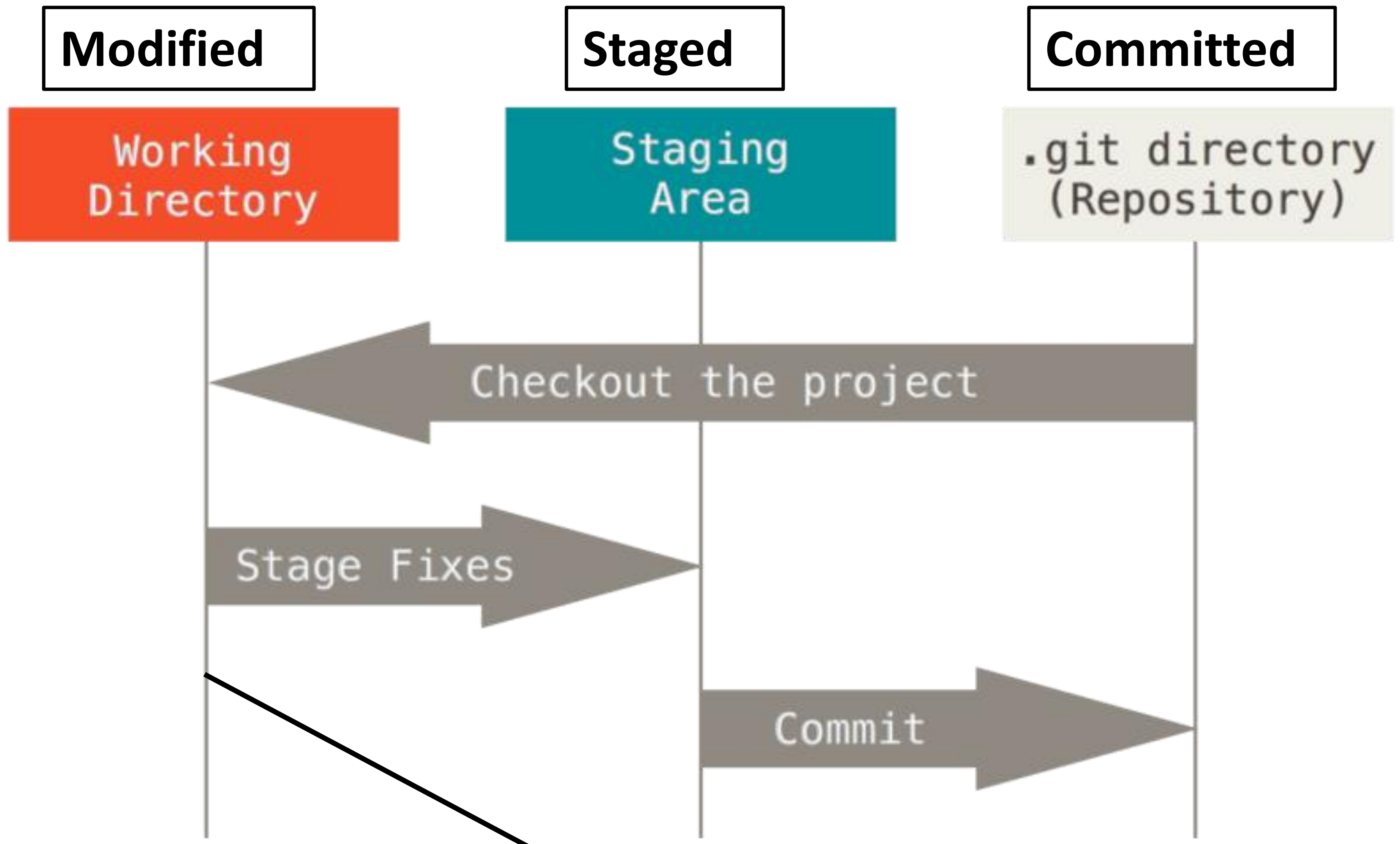
My Setup



```
● (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

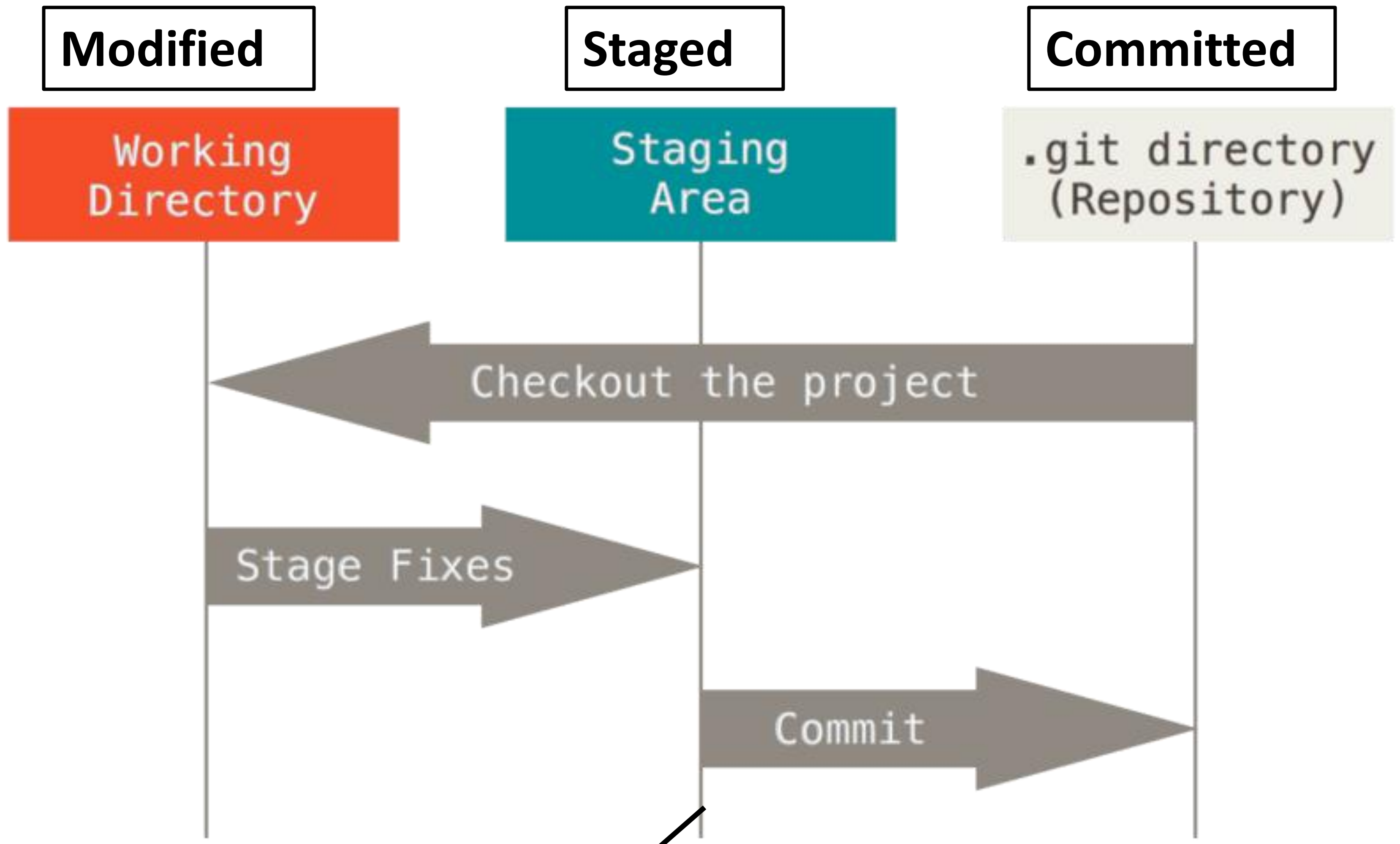
The three states of a file in Git



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

Image taken from: Pro Git book

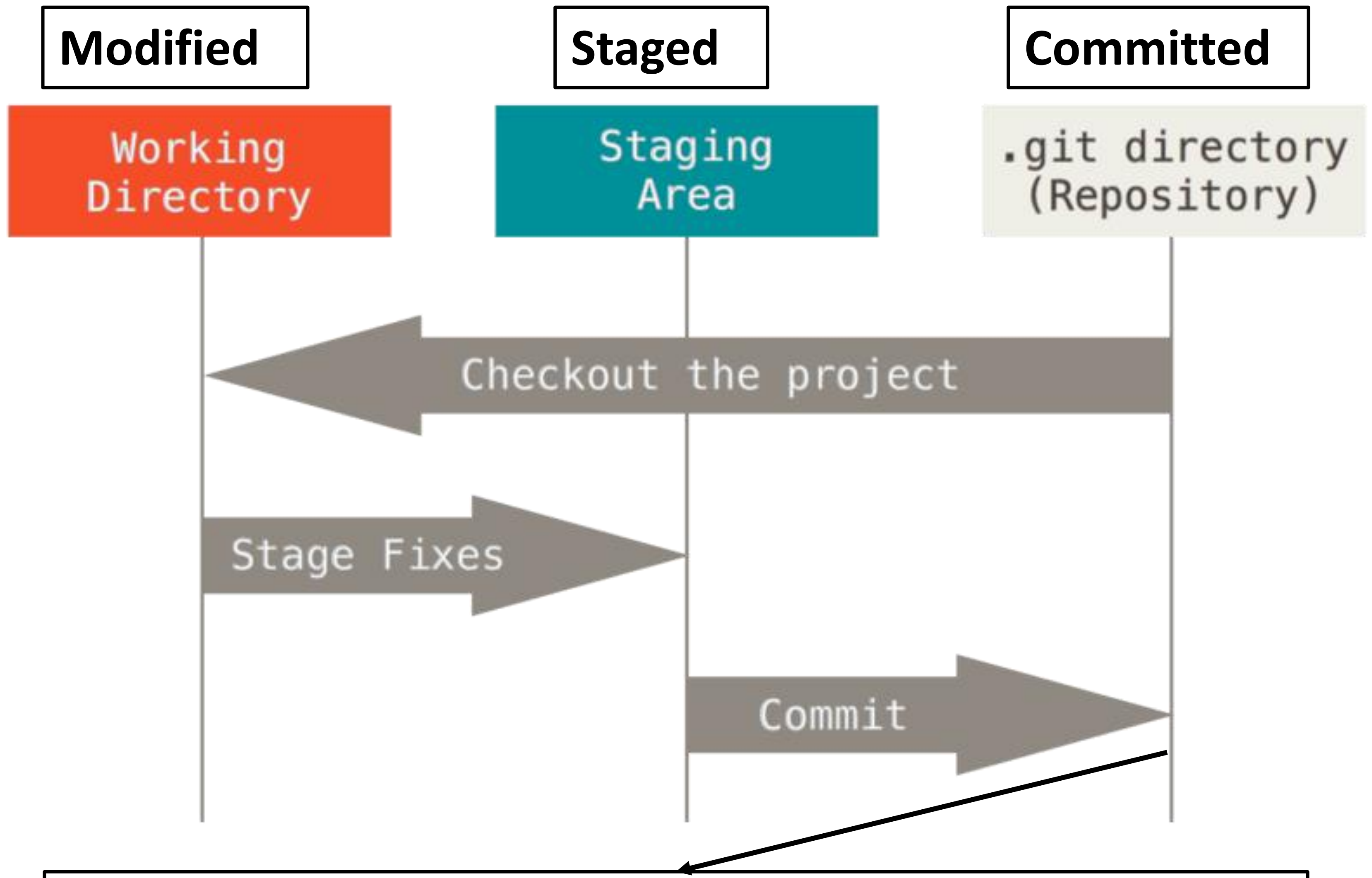
The three states of a file in Git



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

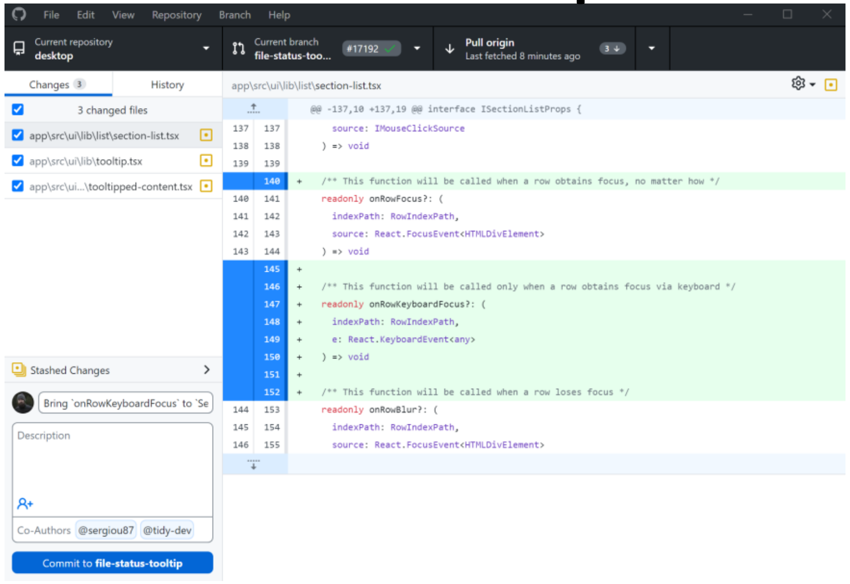


Use “git commit” to store a new version of the project. Changes in the staging area are used in this commit.

Image taken from: [Pro Git book]()

Tools/IDEs Can Help You Make Use of Git

GitHub Desktop



File Edit View Repository Branch Help
Current repository desktop
Current branch file-status-too... #17192
Pull origin Last fetched 8 minutes ago 3
Changes 3
History
3 changed files
app\src\ui\lib\list\section-list.tsx
app\src\ui\lib\tooltip.tsx
app\src\ui\...\tooltipped-content.tsx
Stashed Changes
Bring `onRowKeyboardFocus` to `Se
Description
Co-Authors @sergiou87 @tidy-dev
Commit to file-status-tooltip

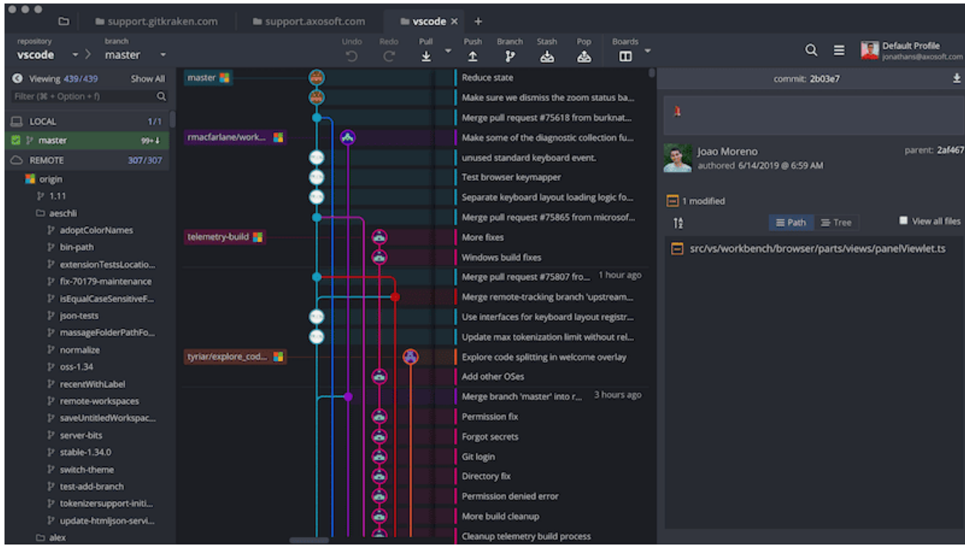
app\src\ui\lib\list\section-list.tsx

```
@@ -137,10 +137,19 @@ interface ISectionListProps {
      source: IMouseClickSource
    ) => void

+   /** This function will be called when a row obtains focus, no matter how */
    readonly onRowFocus?: (
      indexPath: RowIndexPath,
      source: React.FocusEvent<HTMLDivElement>
    ) => void
+
+   /** This function will be called only when a row obtains focus via keyboard */
+   readonly onRowKeyboardFocus?: (
+     indexPath: RowIndexPath,
+     e: React.KeyboardEvent<any>
+   ) => void
+
+   /** This function will be called when a row loses focus */
    readonly onRowBlur?: (
      indexPath: RowIndexPath,
      source: React.FocusEvent<HTMLDivElement>
```

Image taken from: desktop.github.com

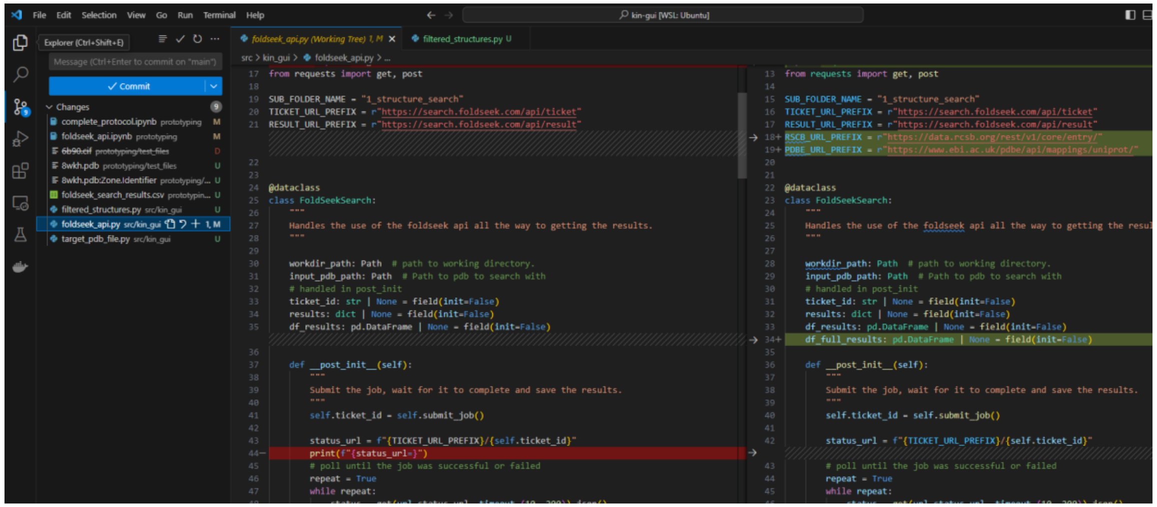
GitKraken



Reduce state
Make sure we dismiss the zoom status ba...
Merge pull request #75618 from burknat...
Make some of the diagnostic collection fu...
unused standard keyboard event.
Test browser keymapper
Separate keyboard layout loading logic fo...
Merge pull request #75865 from microsof...
More fixes
Windows build fixes
Merge pull request #75807 fro... 1 hour ago
Merge remote-tracking branch 'upstream...
Use interfaces for keyboard layout registr...
Update max tokenization limit without rel...
Explore code splitting in welcome overlay
Add other OSes
Merge branch 'master' into r... 3 hours ago
Permission fix
Forgot secrets
Git login
Directory fix
Permission denied error
More build cleanup
Cleanup telemetry build process

Image taken from: gitkraken.com

VSCode



File Edit Selection View Go Run Terminal Help
kin-gui [WSL: Ubuntu]
Explorer (Ctrl+Shift+E)
Message (Ctrl+Enter to commit on "main")
Commit
Changes 9
complete_protocol.ipynb prototyping M
foldseek_api.ipynb prototyping M
6b90.cif prototyping/test_files D
8wkh.pdb prototyping/test_files U
8wkh.pdb:Zone.Identifier prototyping/... U
foldseek_search_results.csv prototypin... U
filtered_structures.py src/kin_gui U
foldseek_api.py src/kin_gui 1, M
target_pdb_file.py src/kin_gui U
foldseek_api.py (Working Tree) 1, M
filtered_structures.py U
src > kin_gui > foldseek_api.py > ...

```
from requests import get, post

SUB_FOLDER_NAME = "1_structure_search"
TICKET_URL_PREFIX = r"https://search.foldseek.com/api/ticket"
RESULT_URL_PREFIX = r"https://search.foldseek.com/api/result"


@dataclass
class FoldSeekSearch:
    """
    Handles the use of the foldseek api all the way to getting the results.
    """

    workdir_path: Path  # path to working directory.
    input_pdb_path: Path  # Path to pdb to search with
    # handled in post_init
    ticket_id: str | None = field(init=False)
    results: dict | None = field(init=False)
    df_results: pd.DataFrame | None = field(init=False)

    def __post_init__(self):
        """
        Submit the job, wait for it to complete and save the results.
        """
        self.ticket_id = self.submit_job()

        status_url = f"{TICKET_URL_PREFIX}/{self.ticket_id}"
        print(f"{status_url=}")
        # poll until the job was successful or failed
        repeat = True
        while repeat:
```

```
from requests import get, post

SUB_FOLDER_NAME = "1_structure_search"
TICKET_URL_PREFIX = r"https://search.foldseek.com/api/ticket"
RESULT_URL_PREFIX = r"https://search.foldseek.com/api/result"
RSCB_URL_PREFIX = r"https://data.rcsb.org/rest/v1/core/entry/"
PDBE_URL_PREFIX = r"https://www.ebi.ac.uk/pdbe/api/mappings/uniprot/"


@dataclass
class FoldSeekSearch:
    """
    Handles the use of the foldseek api all the way to getting the resul
    """

    workdir_path: Path  # path to working directory.
    input_pdb_path: Path  # Path to pdb to search with
    # handled in post_init
    ticket_id: str | None = field(init=False)
    results: dict | None = field(init=False)
    df_results: pd.DataFrame | None = field(init=False)
    df_full_results: pd.DataFrame | None = field(init=False)

    def __post_init__(self):
        """
        Submit the job, wait for it to complete and save the results.
        """
        self.ticket_id = self.submit_job()

        status_url = f"{TICKET_URL_PREFIX}/{self.ticket_id}"

        # poll until the job was successful or failed
        repeat = True
        while repeat:
```

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

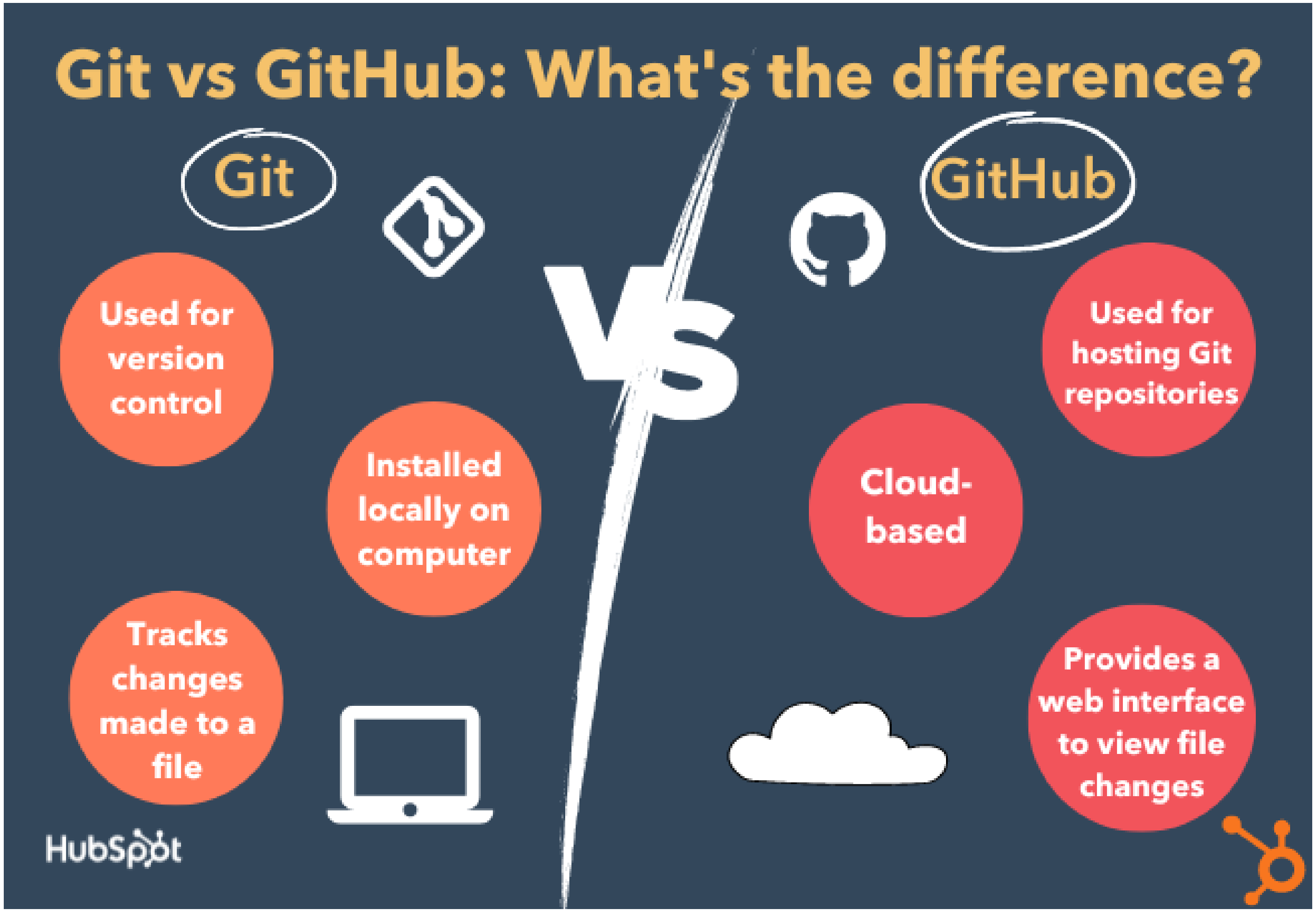

Git vs GitHub: What's the difference?
Git
VS
GitHub
Used for version control
Installed locally on computer
Tracks changes made to a file
Used for hosting Git repositories
Cloud-based
Provides a web interface to view file changes
HubSpot

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.

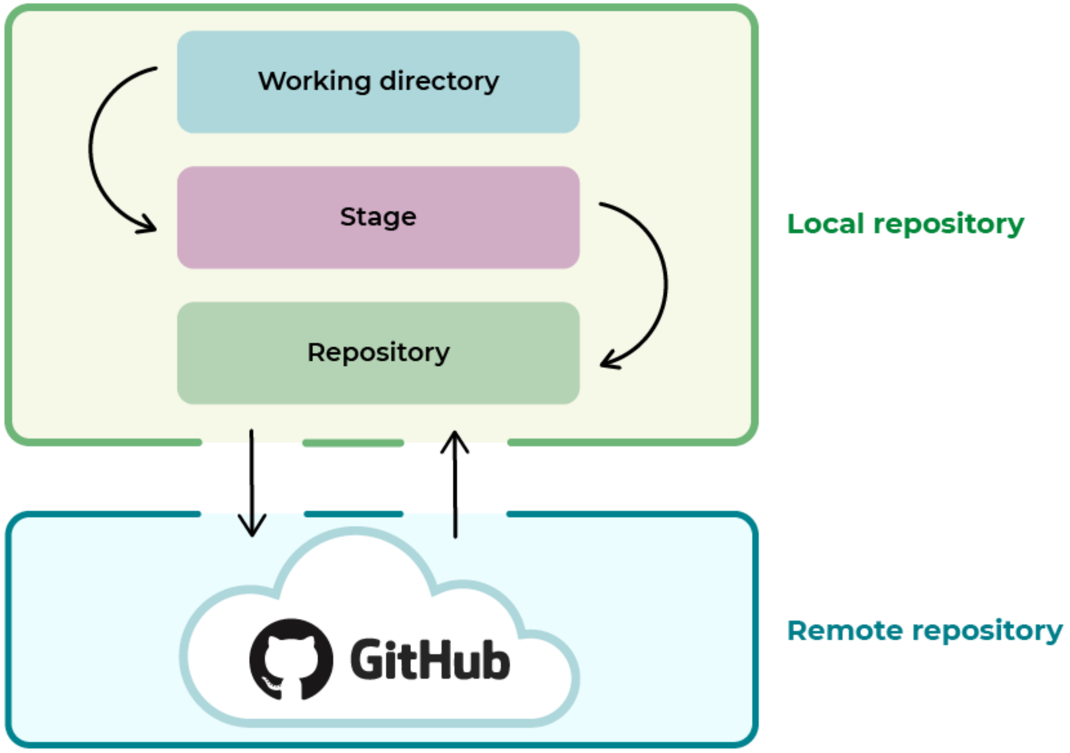


Image taken from openclassrooms

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

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

Why Use a Remote:
- 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:

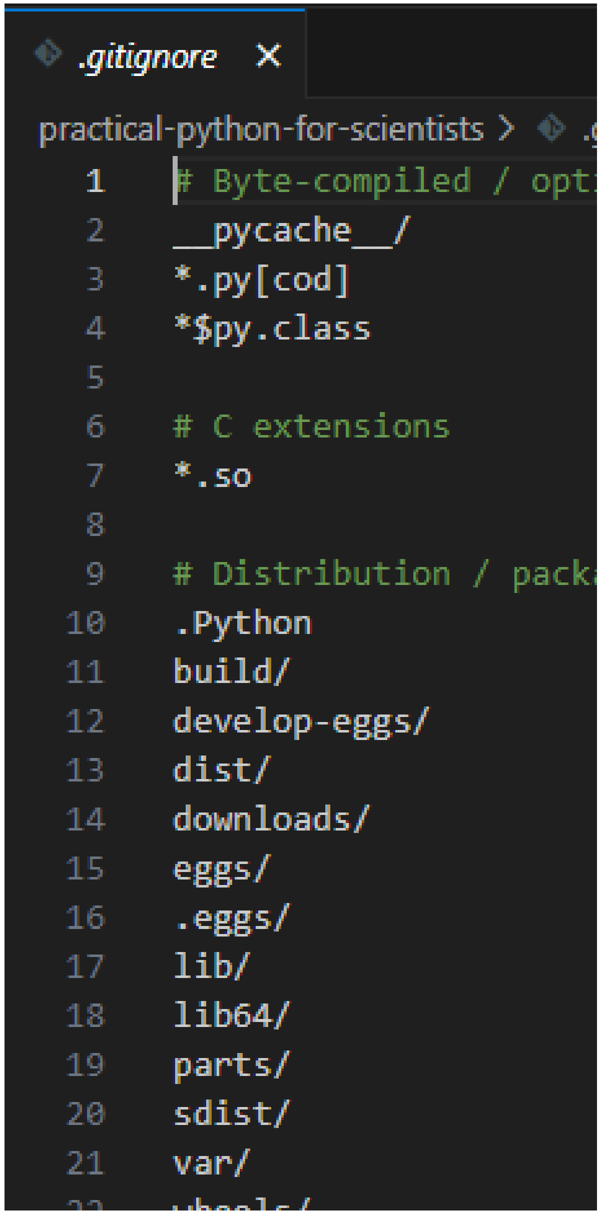
- Large datasets.
- Sensitive/Personal data.
- Passwords/usernames.
- System-specific files, e.g. .DS_Store on a Mac.

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 > .g

```
# 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

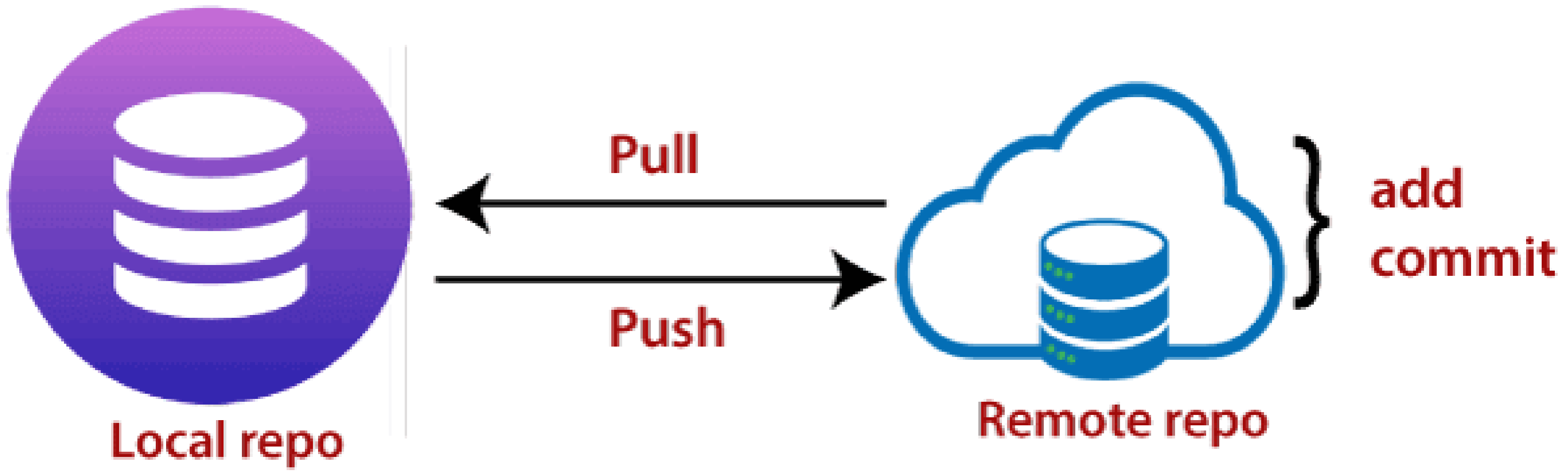


Image taken from: https://www.javatpoint.com/git-push

- *“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

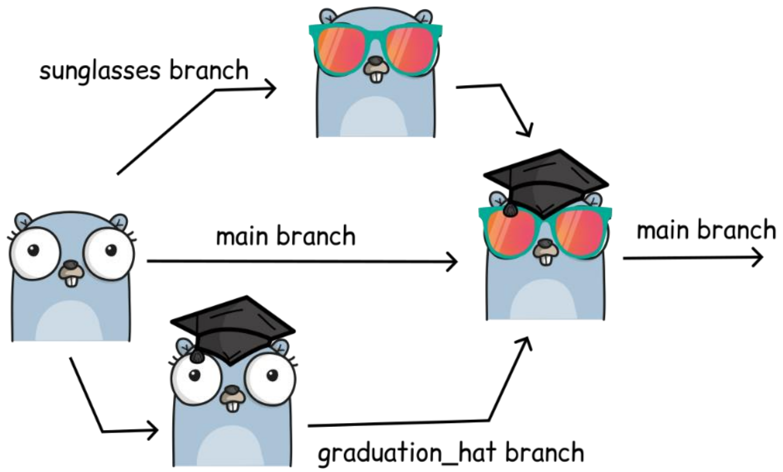


Image taken from: https://coderefinery.github.io/git-intro/branches/

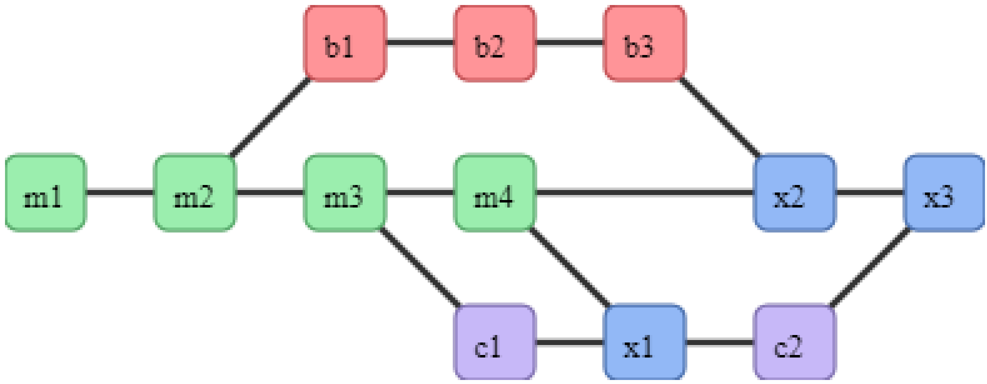


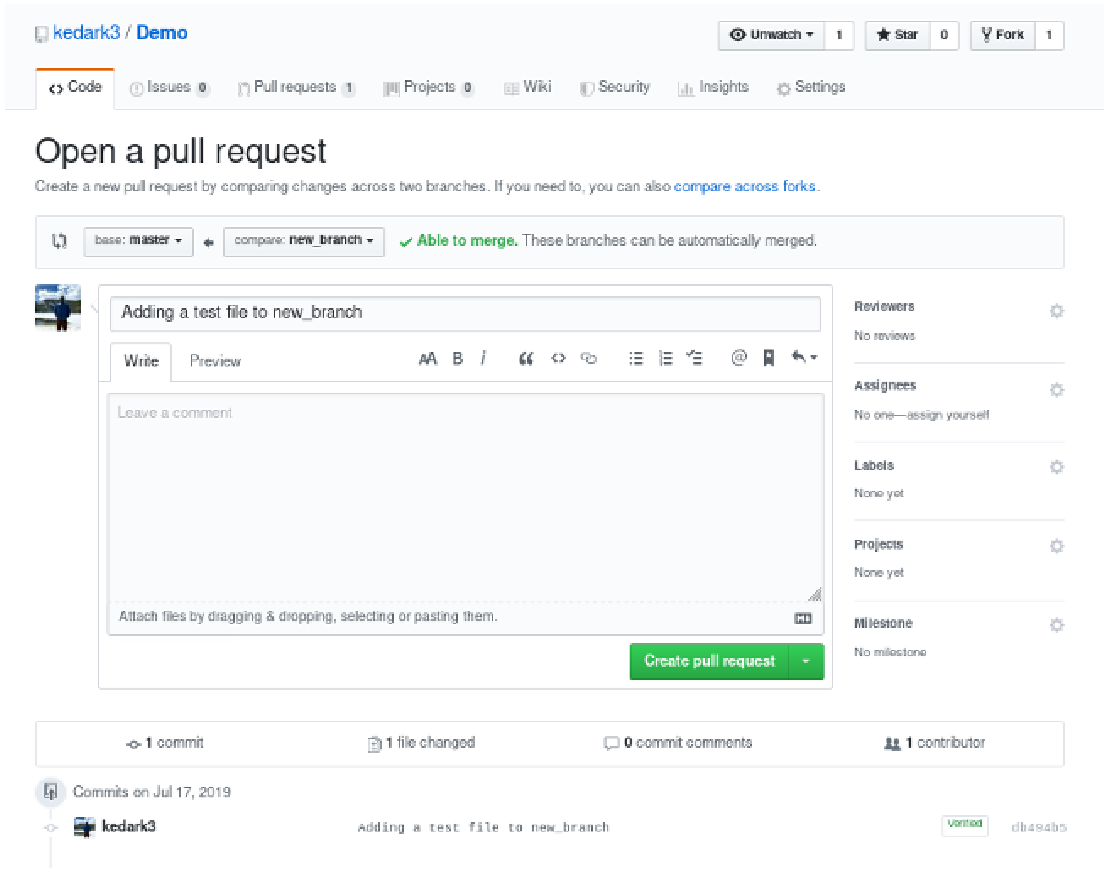
Image taken from: https://coderefinery.github.io/git-intro/branches/

- 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



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ← compare: new_branch ✓ Able to merge. These branches can be automatically merged.

Adding a test file to new_branch

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Reviewers: No reviews
Assignees: No one—assign yourself
Labels: None yet
Projects: None yet
Milestone: No milestone

1 commit · 1 file changed · 0 commit comments · 1 contributor

Commits on Jul 17, 2019

kedark3 `Adding a test file to new_branch` Verified db494b5

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

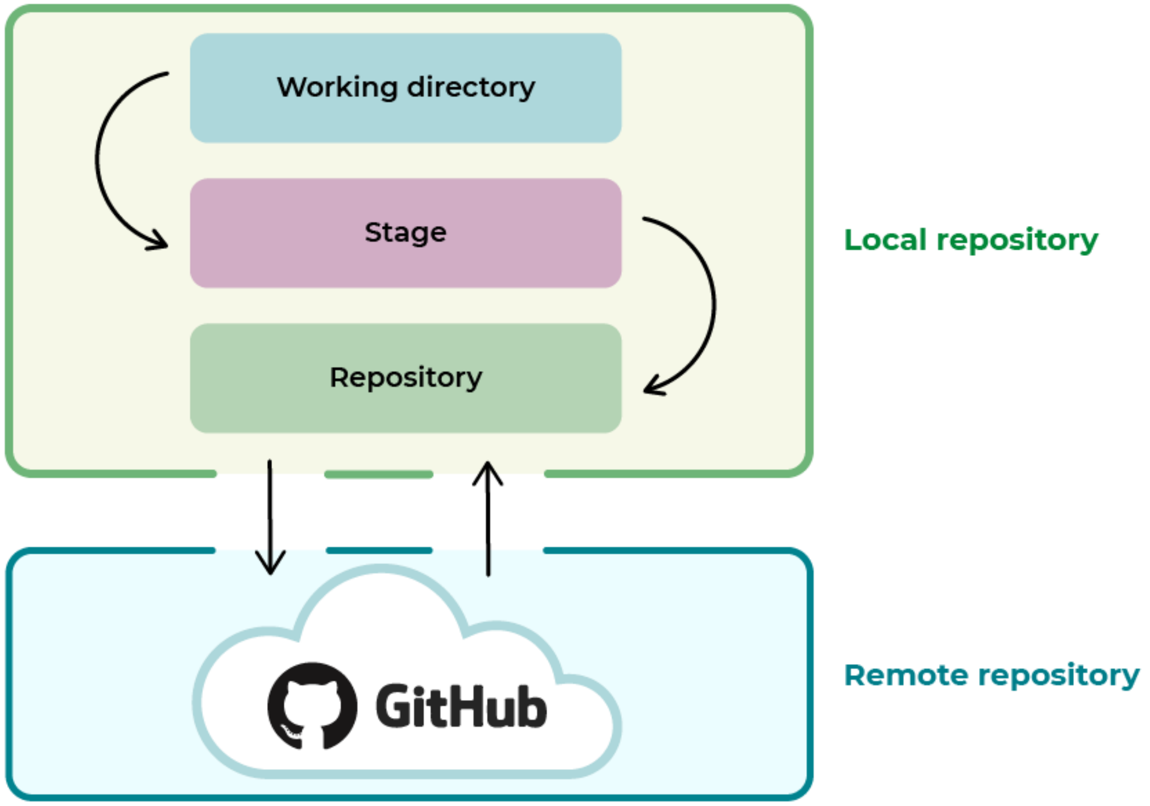


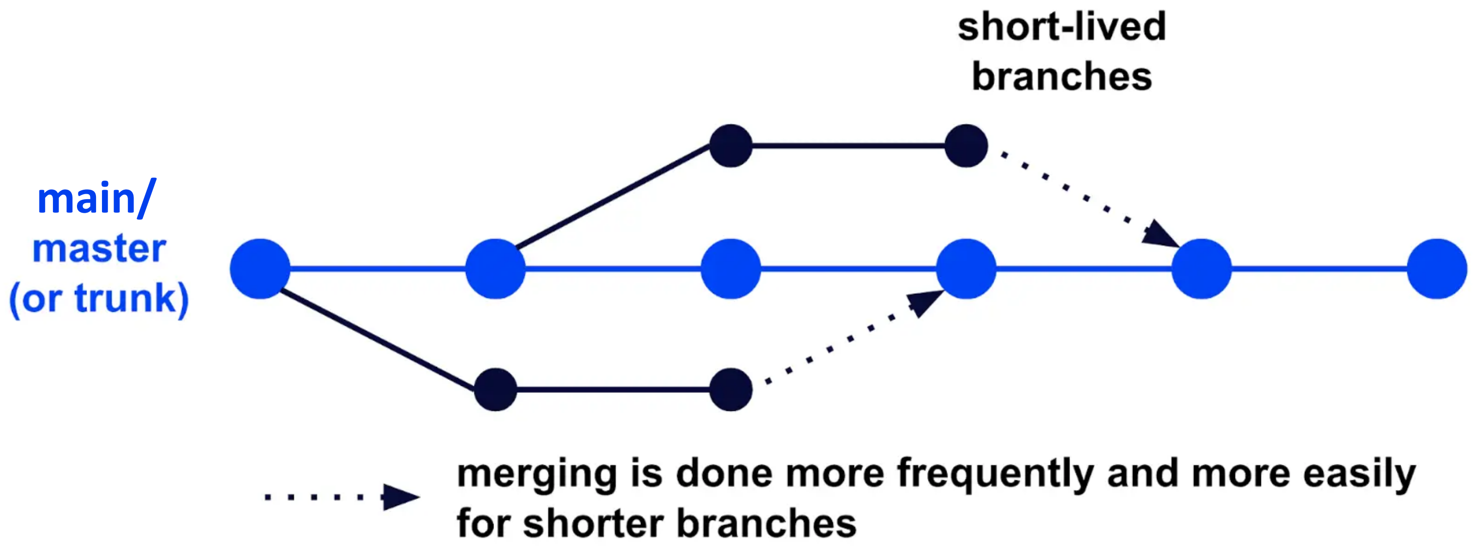
Image taken from [openclassrooms]

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

BELOW ARE SLIDES I CONSIDERED USING BUT DIDN’T DUE TO TIME CONSTRAINTS

Trunk based development can be a good strategy for small groups

- There are a lot of branching strategies…
- Most are inappropriate for small scientific projects involving you and a few colleagues.
- Trunk based development can be a good idea…


short-lived branches
main/ master (or trunk)
merging is done more frequently and more easily for shorter branches

X

Image taken from: https://www.optimizely.com/optimization-glossary/trunk-based-development/

Practicing trunk based development

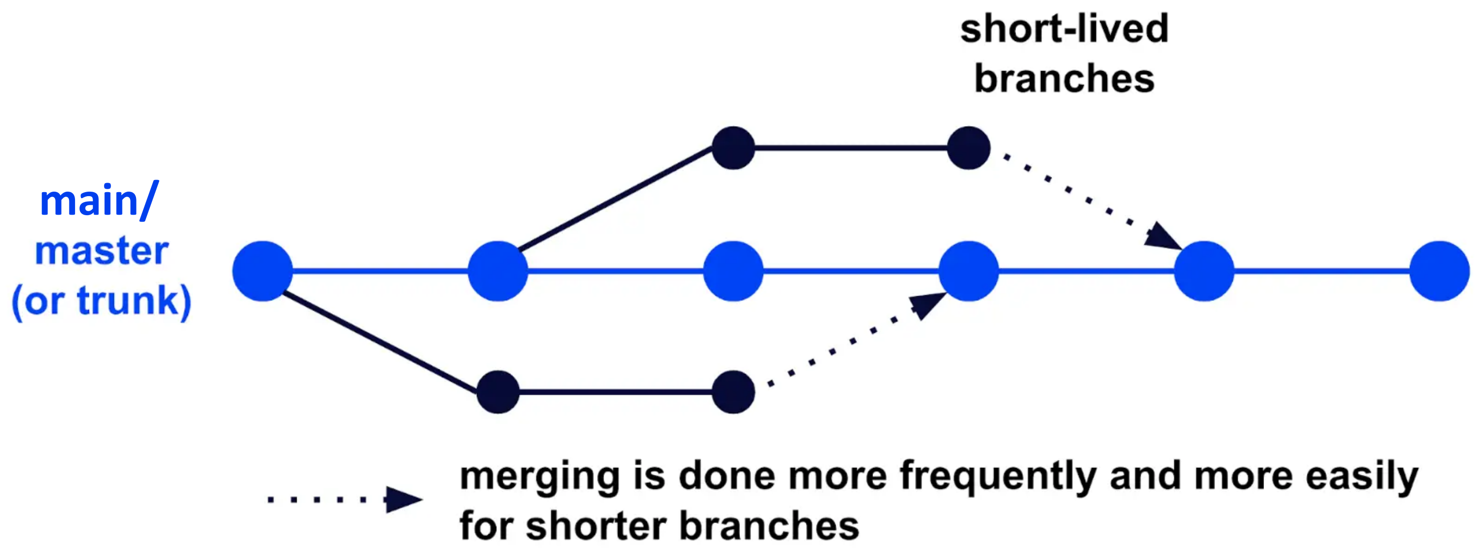

short-lived branches
main/ master (or trunk)
merging is done more frequently and more easily for shorter branches

Image taken from: https://www.optimizely.com/optimization-glossary/trunk-based-development/

- You have **one main branch** which holds code you're all happy with.
- New features/ideas get implemented on a different branch.
- **Once happy** with the new feature, it is merged onto the main branch.
- Don't take too long to merge the new feature.

- **Discuss and plan with co-workers who will do what. Working on different aspects of a project can make the merging process much much much easier**

X