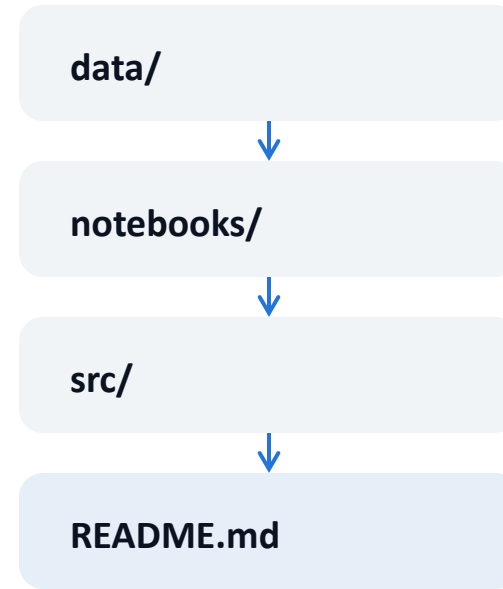


Git & GitHub for Data Science Projects

Track changes. Reproduce experiments.
Collaborate with confidence.



A project is more than one file.

A pile of “final” files.

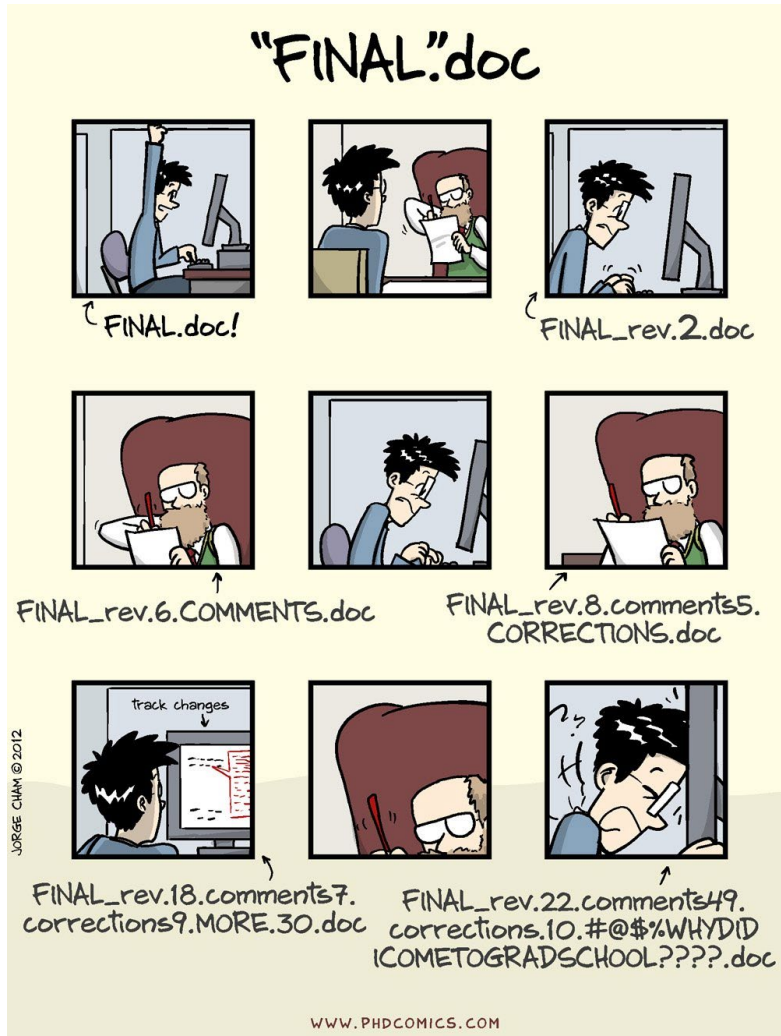
```
analysis.py  
analysis_final.py  
analysis_final_v2.py  
analysis_final_really_final.py
```

When names become your versioning system...

- Which version is correct?
- What changed between them?
- Can you recover an older version?

Git records the answers—without renaming files.

A pile of “final” files.



- Which version is correct?
- What changed between them?
- Can you recover an older version?

Git records the answers—without renaming files.

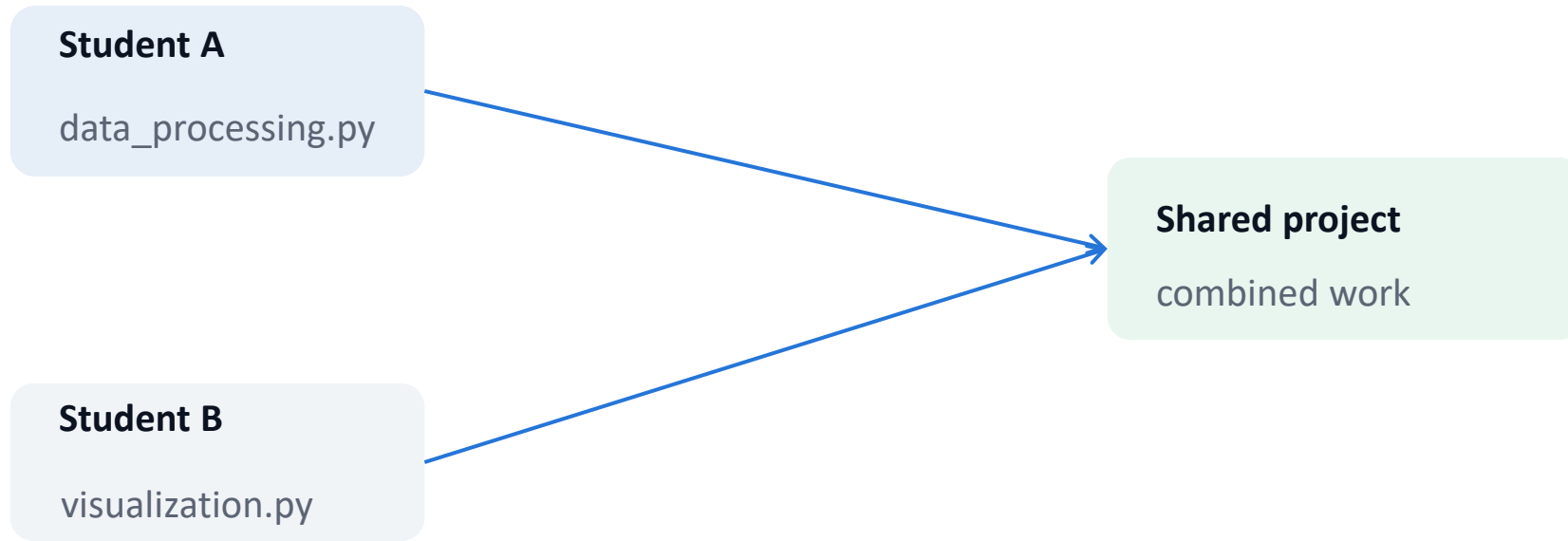
Experiments need a reproducible trail.

MODEL	ALGORITHM	SETTING	ACCURACY
A	Random Forest	100 trees	82%
B	Random Forest + Feature Engineering	500 trees	85%
C	XGBoost + Tuned Parameters	—	87%

Model C performs the best—but can we reproduce the result?

- Which version of the code generated Model C?
- Which parameters and datasets were used?
- Can another student reproduce the same result?

Collaboration works when changes can be combined safely.

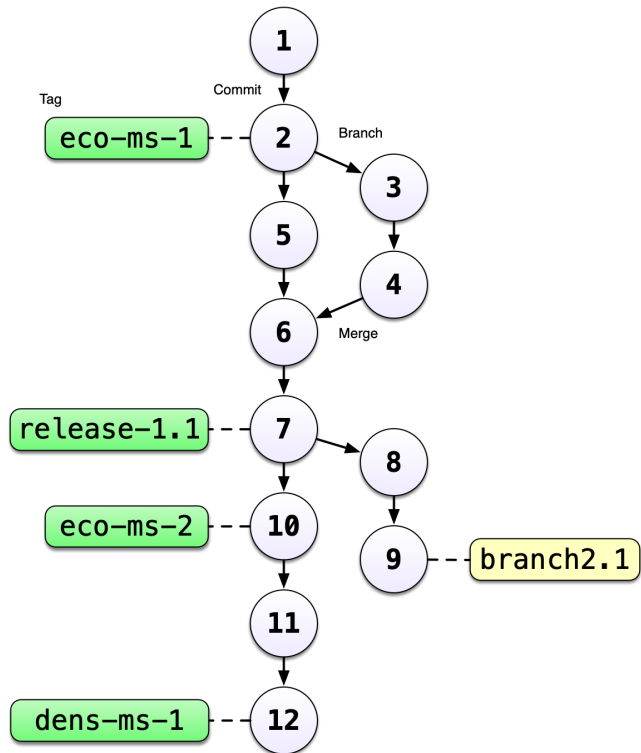


Git makes each contribution visible, reviewable, and mergeable.

Git is a distributed version control system.

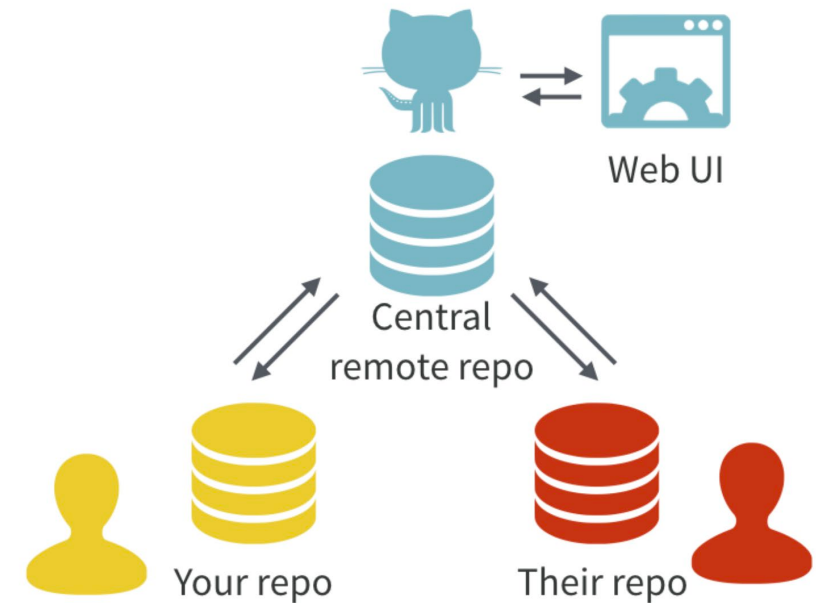
Version control

Manages different versions of a project over time.



Distributed

Each developer has a complete copy of the project history.



They work together, but they are not the same thing.

GIT

Software tool

Runs locally

Tracks changes

Command-line workflow

GITHUB

Online platform

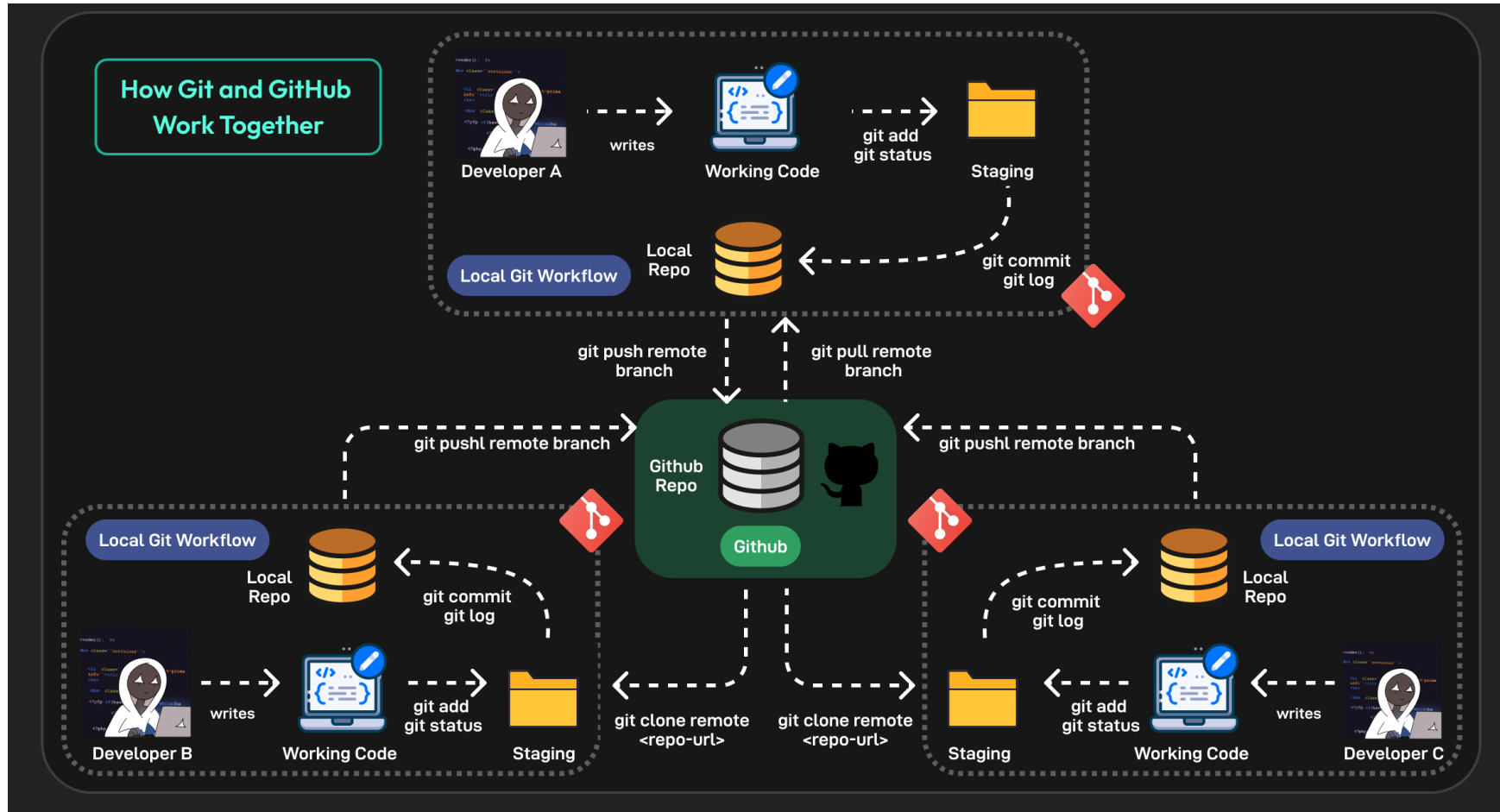
Cloud-based

Stores repositories

Collaboration features

Analogy: Git is “Track Changes”; GitHub is the shared place where the project lives.

They work together, but they are not the same thing.



You can use Git without GitHub but you cannot use GitHub without Git

Many Companies use Git but do not put code to GitHub.



Amazon uses **GitFarm** as its internal Git repository hosting service for managing package source code.

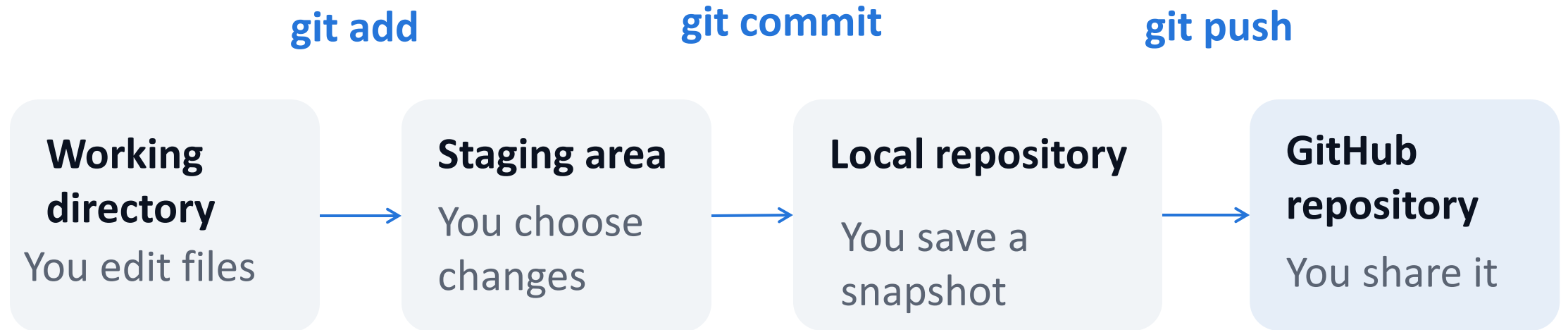


Google uses **Piper** as a centralized version control system for its internal software development.



Alibaba uses **git-repo**, a command-line client tool that wraps native Git to support its centralized AGit-Flow code review workflow.

Git moves selected changes through four places.



```
git commit -m "Add correlation heatmap visualization"
```

Good messages say what changed and why.

A repository is a project folder managed by Git.

```
DataScienceProject/  
├── data/  
├── notebooks/  
├── src/  
├── README.md  
└── requirements.txt
```

Local repository

stored on your computer



Remote repository

stored on GitHub

Same project. Two locations. One shared history.

Install Git, verify it, then tell Git who you are.

1 Installation

<https://git-scm.com/install/windows>

2 Check installation

```
git --version
```

2 Configure identity

```
git config --global user.name "Your Name"  
git config --global user.email "your_email@example.com"
```

Author:

Alice Smith <alice@gmail.com>

Message:

Add data preprocessing

Time:

Aug 25, 2026

Every commit records an author, date, message, and changes.

Git gives you three ways to understand your work.

git status

What is modified, staged, or untracked?

git diff

What lines changed?

git log

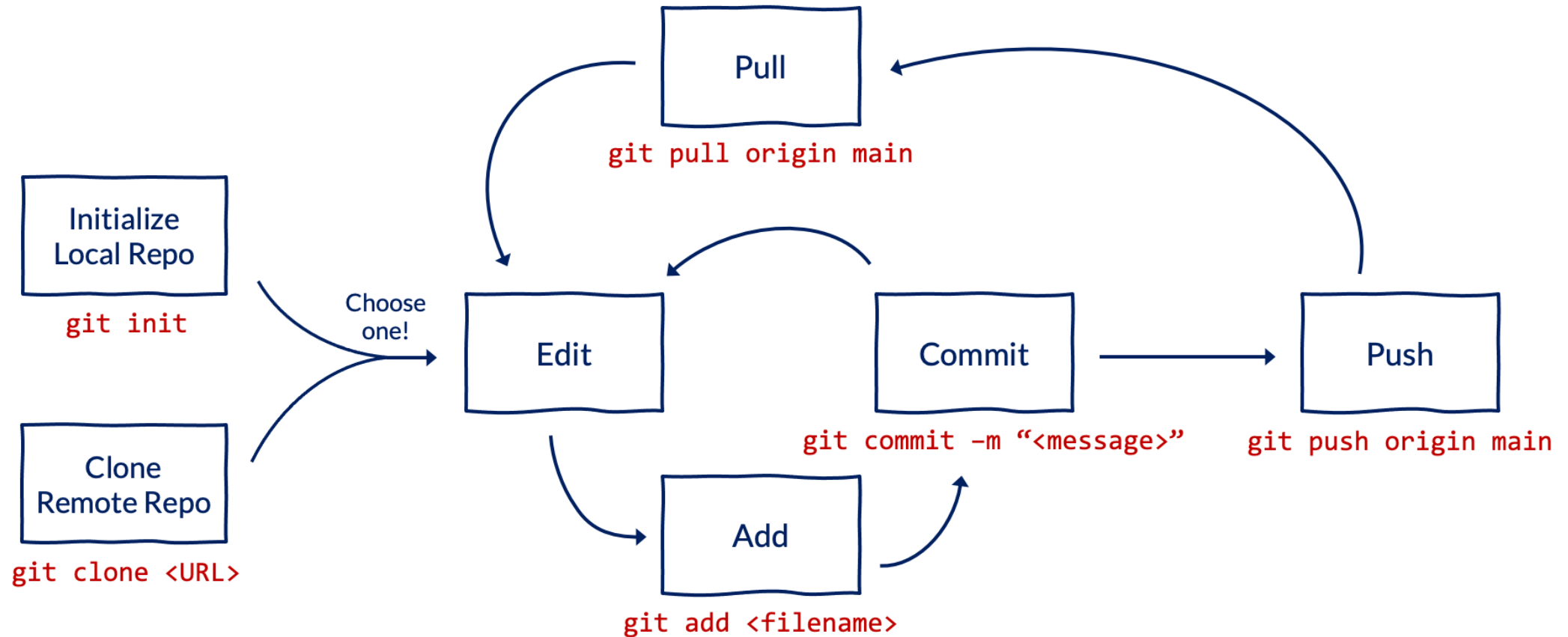
What happened before?

+ added line
- deleted line

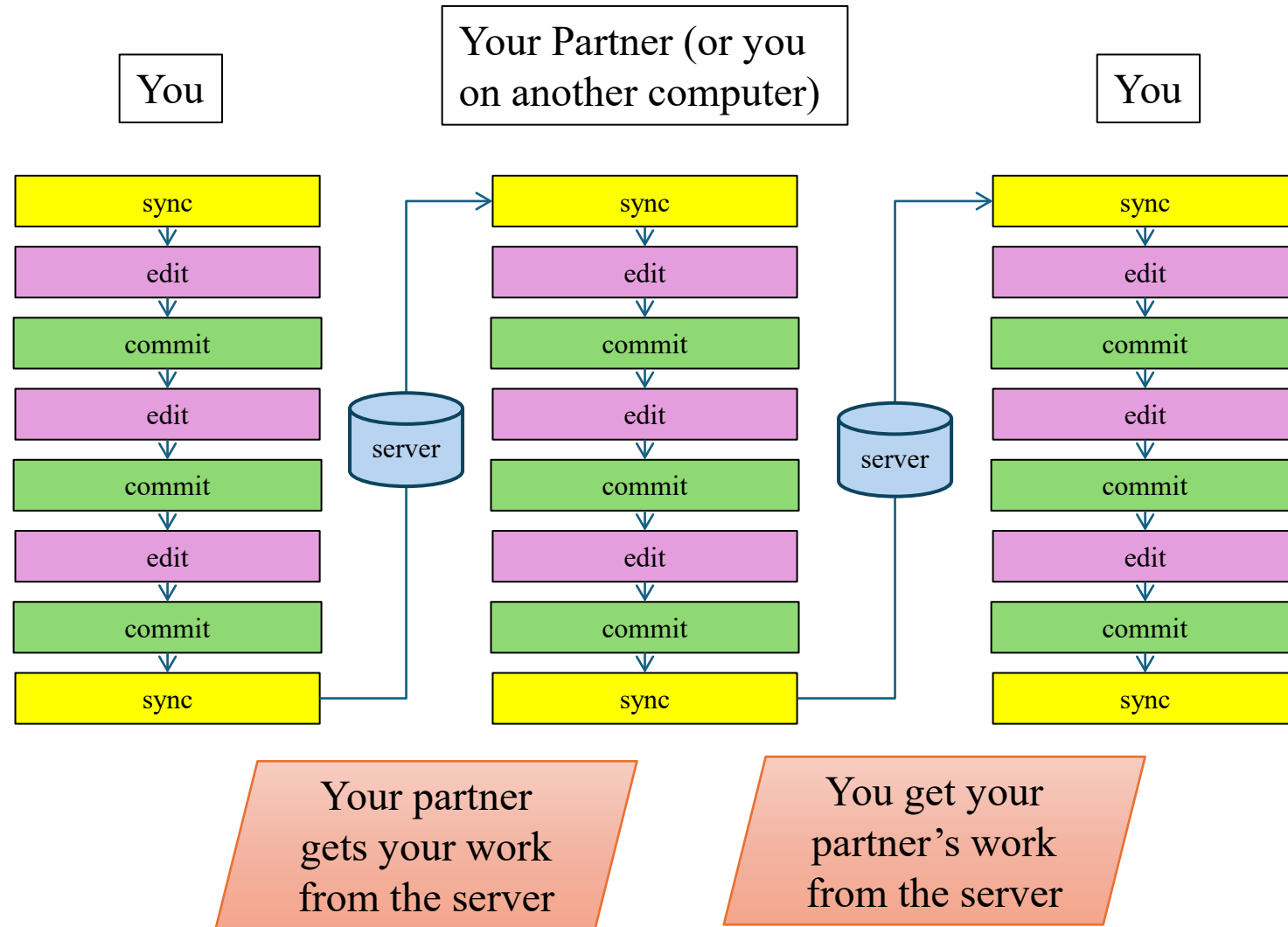
Basic workflow with Git and GitHub.

LOCAL

REMOTE



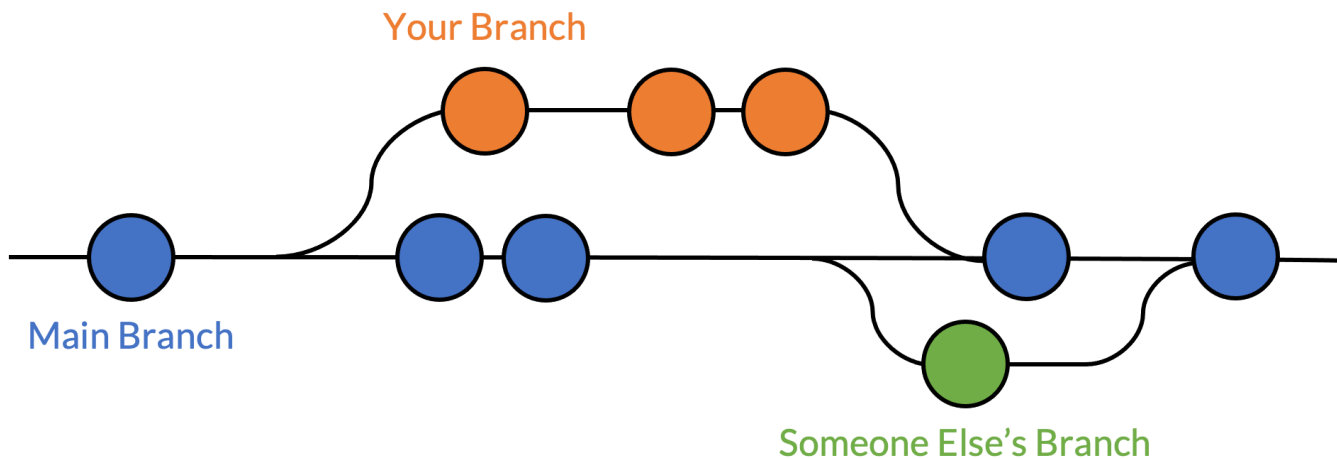
Basic workflow with Git and GitHub.



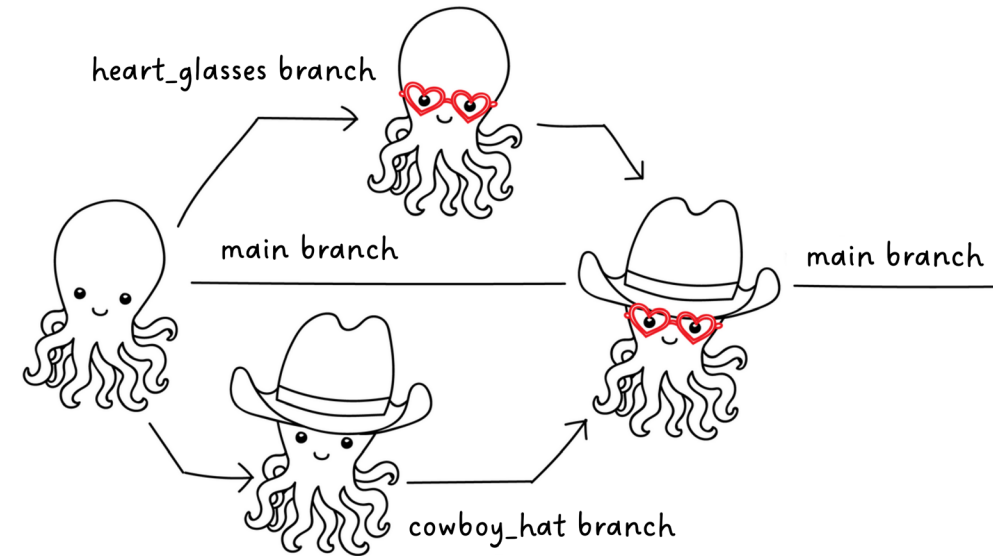
Use a branch to test an idea without disturbing main.

All commits live on a branch.

The default branch in a repository is called the main branch.

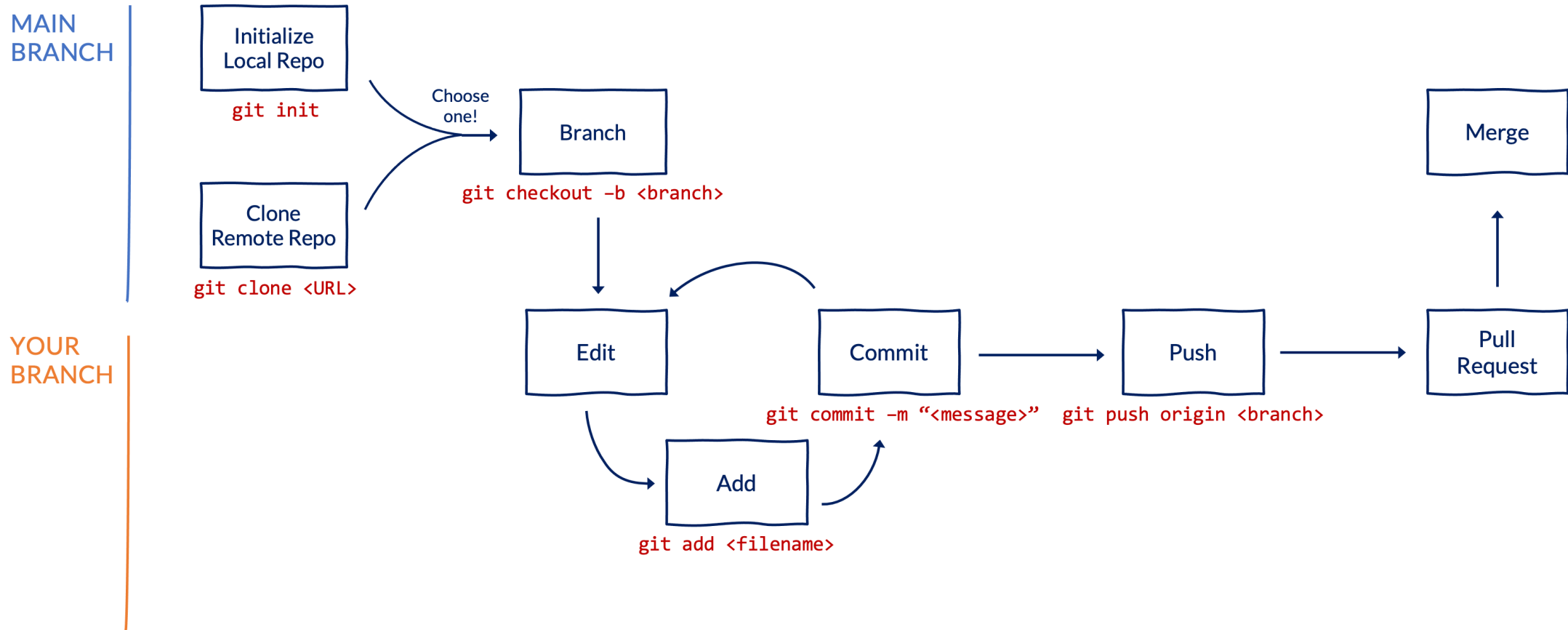


```
git branch your-branch  
git checkout your-branch
```



Use a branch to test an idea without disturbing main.

A more robust collaborative workflow involves using branches and pull requests.



Key Commands

- `git clone <url>`: “Copy this remote project to my local computer.”
- `git add <filename>`: “Look at this file.”
- `git commit -m <message>`: “Take a snapshot of the files I’ve added.”
- `git status`: “Tell me what you’re looking at.”
- `git push`: “Send my snapshots to GitHub so others can see them.”
- `git pull`: “Replace my local version of this project with the remote version.”
- `git checkout -b <new_branch>`: “Create a new branch and switch to it.”
- `git checkout <existing_branch>`: “Switch to a different branch.”

Create a mini project and publish its first history.

-
- | | | |
|---|--------|----------------------|
| 1 | Create | student-data-project |
|---|--------|----------------------|
-
- | | | |
|---|-----|---------------------------------------|
| 2 | Add | README.md, analysis.ipynb, .gitignore |
|---|-----|---------------------------------------|
-
- | | | |
|---|--------|---|
| 3 | Commit | Initial project structure |
|---|--------|---|
-
- | | | |
|---|--------|---|
| 4 | Modify | Add exploratory data analysis |
|---|--------|---|
-
- | | | |
|---|------|-----------|
| 5 | Push | to GitHub |
|---|------|-----------|
-

Success criterion: your GitHub repository shows two meaningful commits.