جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية علوم الحاسب وتقنية المعلومـات
College of Computer Science and Information Technology

Git, GitHub, and Terminal

An Introduction

Imam Abdulrahman Bin Faisal University
College of Computer Science and Information Technology
Teaching Assistant in Computer Engineering Department

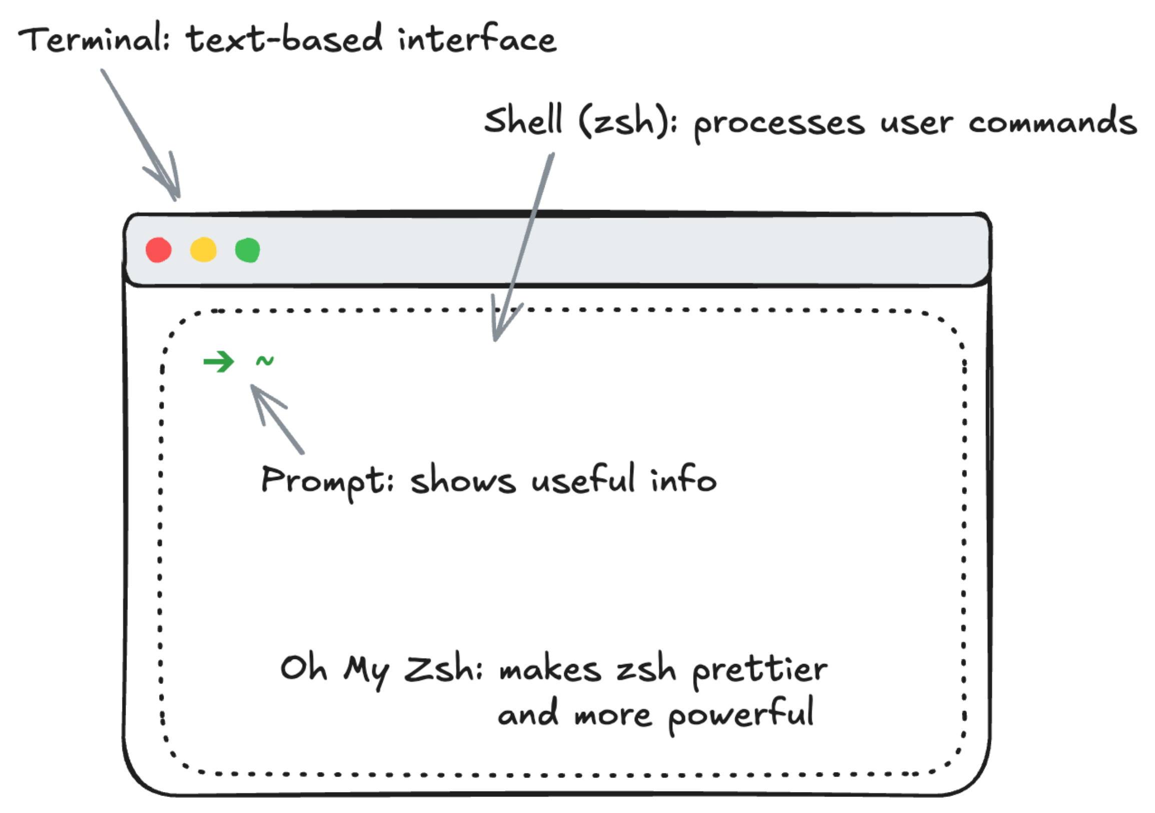
- Ms. Rema Alnssiry
- Ms. Somiah Aljaafary
- Ms. Hind Alzahrani

V1 - 2026

Terminal

Surviving in the terminal:

- `pwd`
- `ls -l`
- `mkdir <directory>`
- `cd <path>`
- `touch <file>`
- `rm <file> # rm -r <directory>`
- `mv <from> <to>`


Terminal: text-based interface
Shell (zsh): processes user commands
→ ~
Prompt: shows useful info
Oh My Zsh: makes zsh prettier and more powerful

Python

pyenv

We used pyenv to install and manage multiple Python versions.

- `pyenv versions`
- `pyenv install <version>`
- `pyenv global <version>`

virtualenv

- `pyenv virtualenvs`

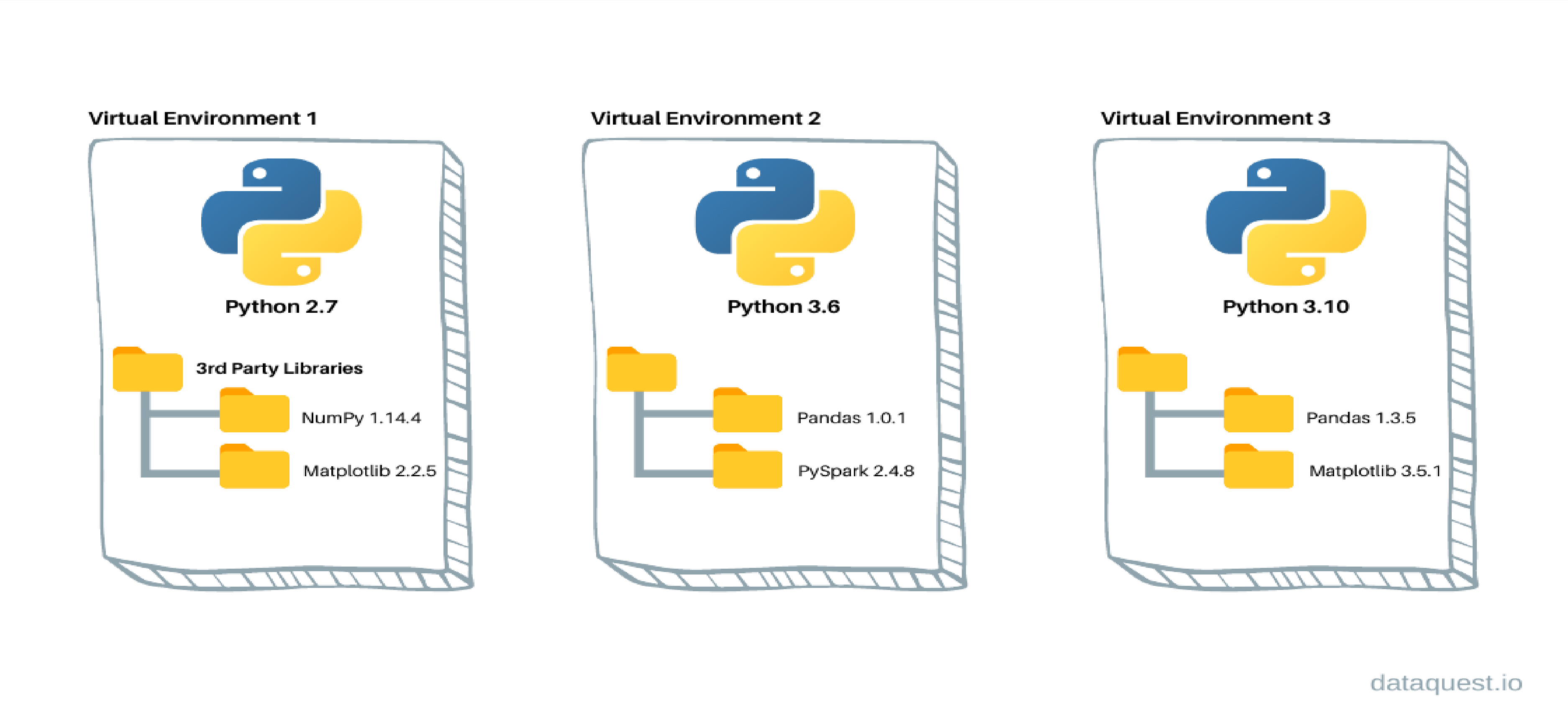
A safe place to `pip install <package>` without interfering with other tools.

Real situation:

Suppose you have Project A depending on pandas==1.4.4 and Project B on pandas==2.2.3, what can we do?

You can isolate those dependencies in two different virtualenvs.

Python



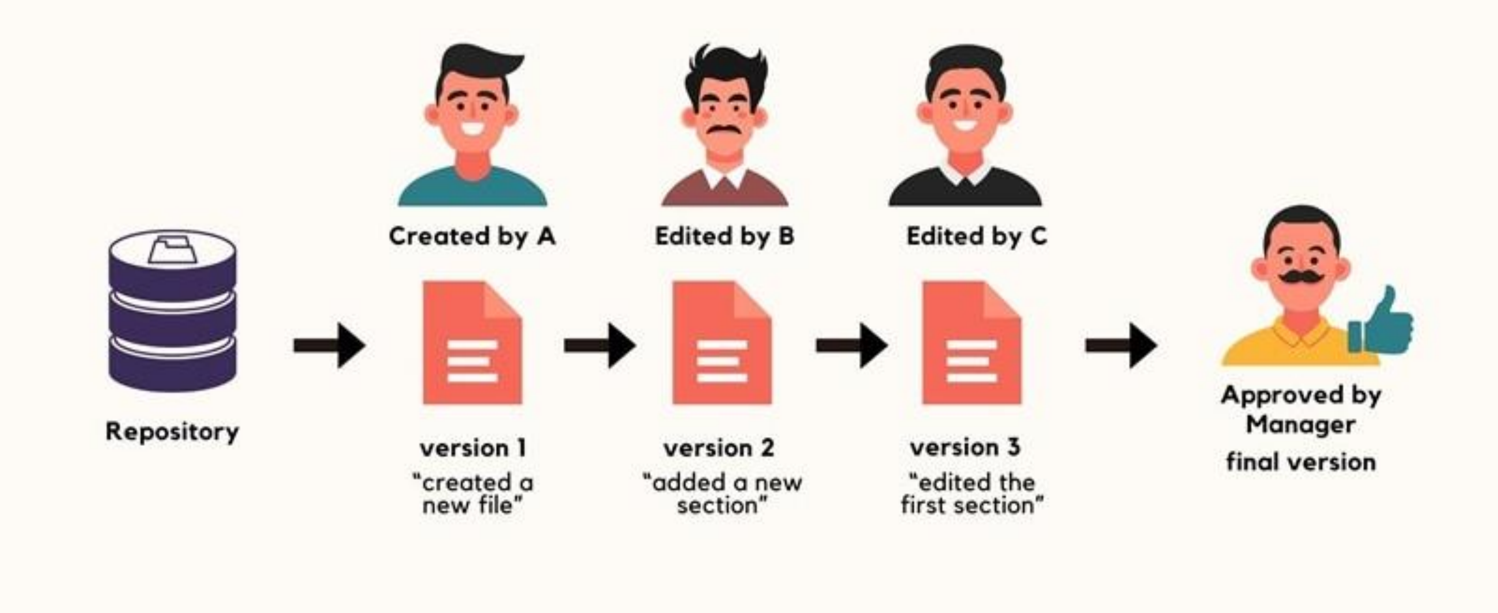
dataquest.io

- In the first weeks we'll use only one virtual environment called ai.
- It will be set by default when starting a new shell.

What Is Version Control?

- **Version Control** is a system that tracks changes to files over time.
- It's like a smart tool that helps people work together on projects, especially on computer programs.

- **Why it is important:**
  - Keeps history of changes
  - Allows reverting to previous versions
  - Supports collaboration
  - Prevents accidental data loss



git

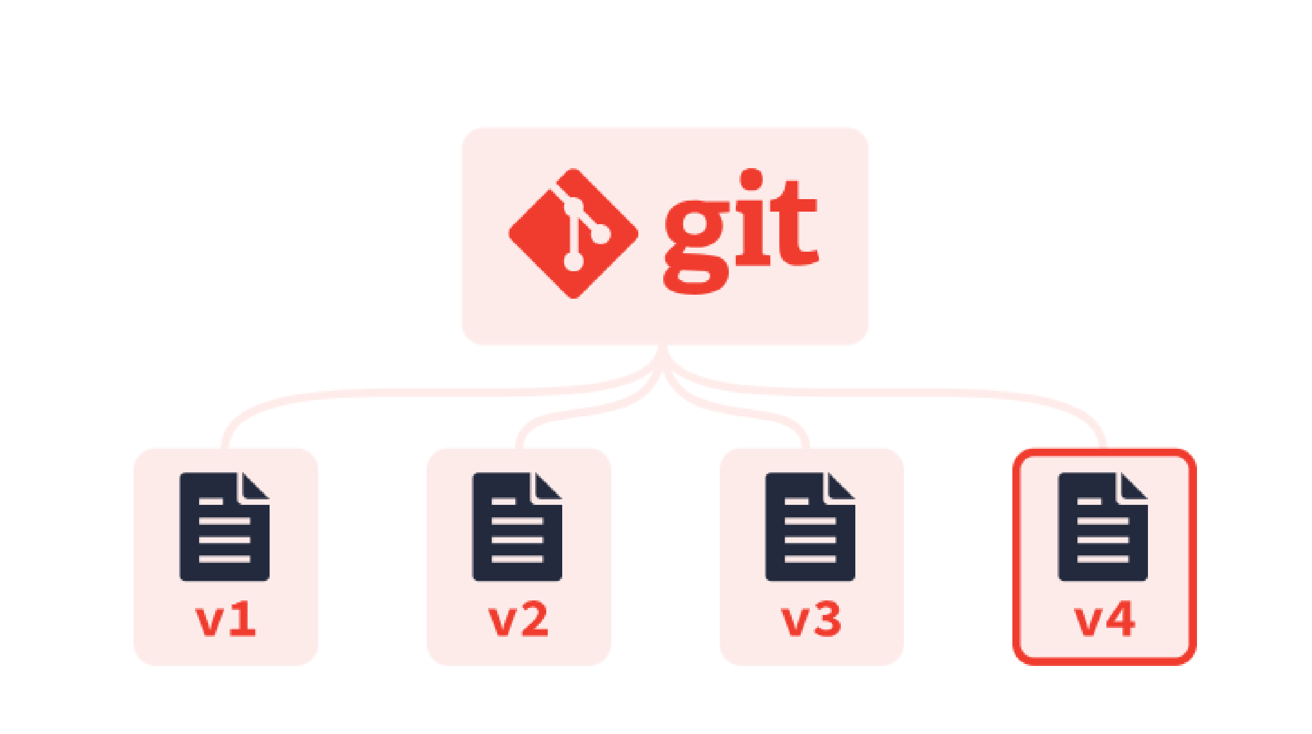
!=

GitHub

What Is Git?

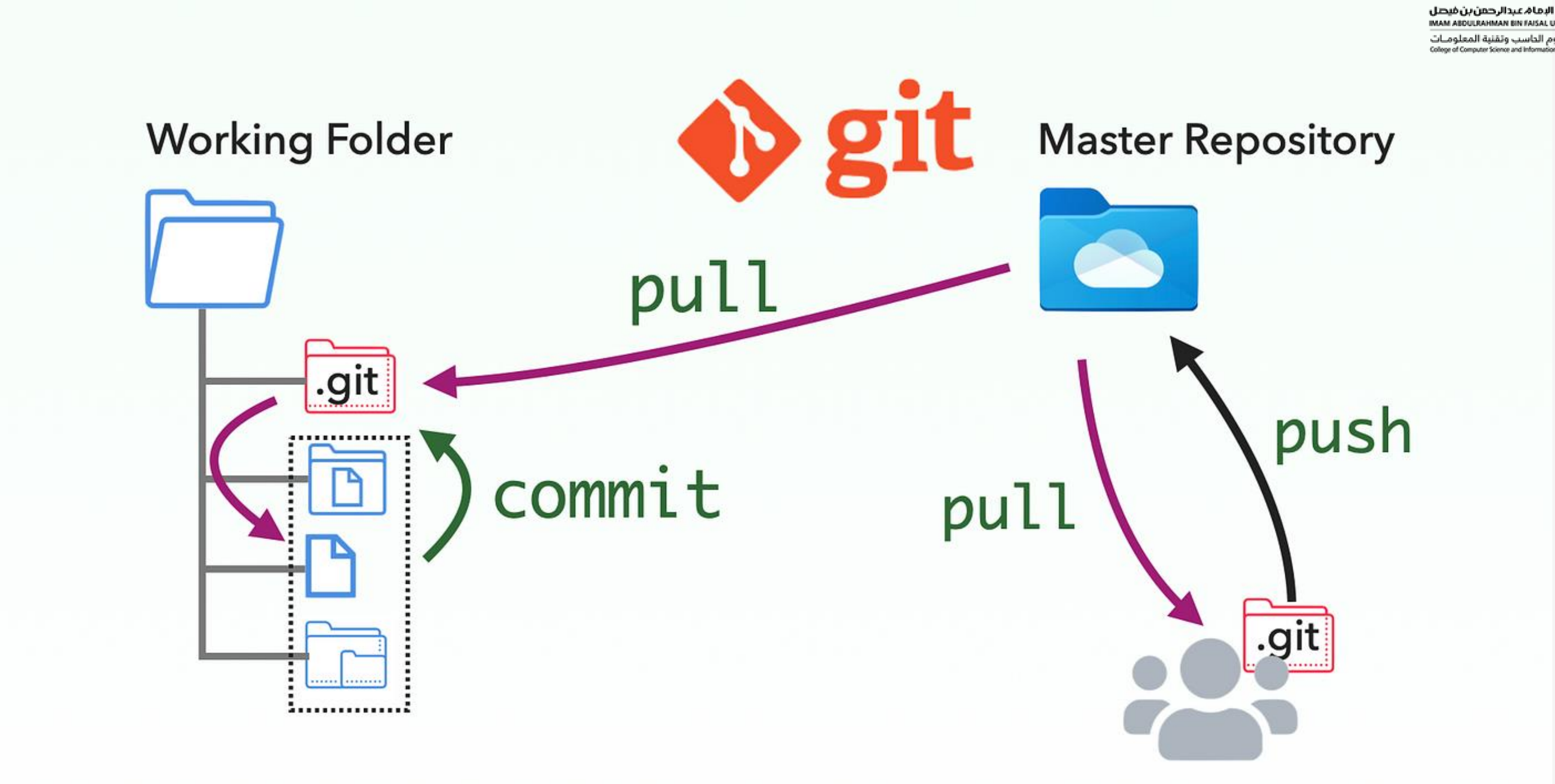
Git is a distributed version control system.

- Tracks changes in files
- Fast and Reliable
- Works Offline
- Used Via Command Line or tools like GitHub Desktop



Key Git Concepts

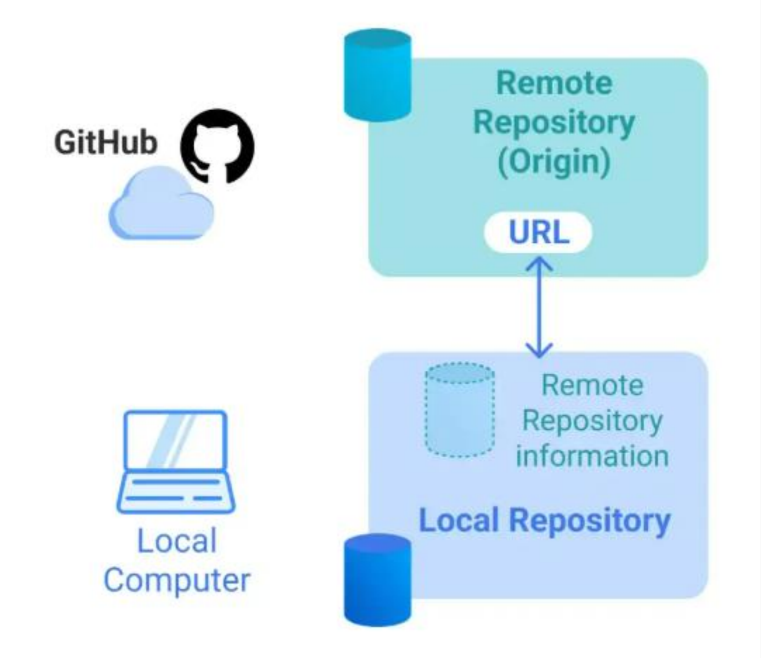
- **Repository (Repo):** Project folder
- **Commit:** Saved snapshot of changes
- **Branch:** Separate line of development
- **Merge:** Combine branches
- **Clone:** Copy a repository
- **Push:** Upload changes
- **Pull:** Download changes


git
Working Folder
Master Repository
.git
pull
commit
pull
push
.git

Repository (Repo)

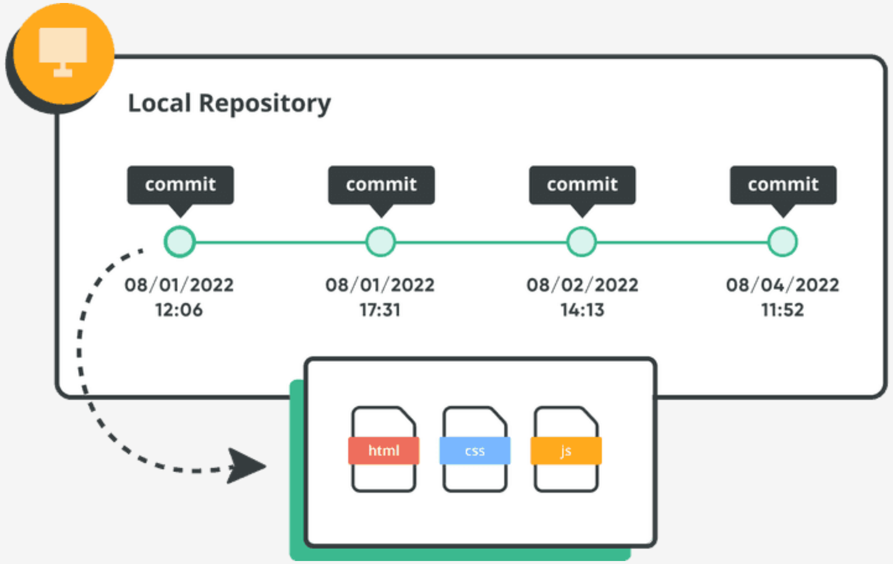
- **A repository** is a central project folder where all files, folders, and their complete change history are stored.
- Contains source code, documents, and configuration files

- **Local Repository:** on your computer
- **Remote Repository:** on GitHub



Commit

- A **commit** is a saved snapshot of changes made to files in the repository at a specific point in time.
- Records what was changed, when, and by whom
- Includes a commit message describing the changes
- Helps restore previous versions if needed

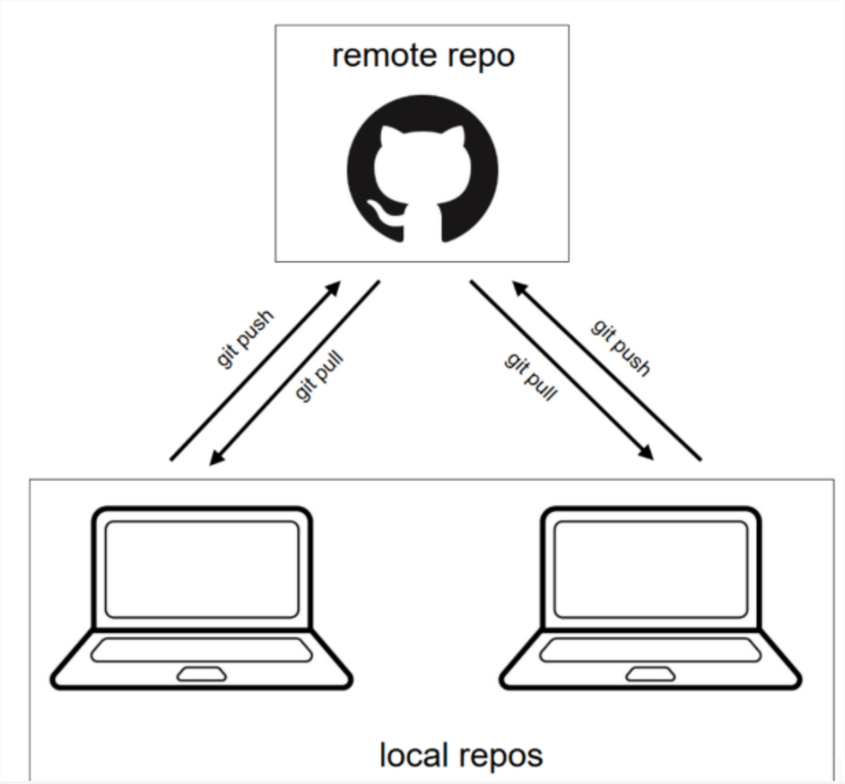


Clone

- A **clone** creates a local copy of a remote repository on your computer.
- Downloads all files and version history
- Allows offline work
- Keeps connection to the remote repository for updates

Push

- A **push** uploads committed changes from a local repository to a remote repository.
- A **pull** downloads and integrates changes from a remote repository into your local repository.



What Is GitHub?

- **GitHub is a cloud-based platform for hosting Git repositories.**
- Stores code online
- Enables collaboration
- Provides issue tracking
- Supports project management
- Widely used in open-source projects

Git vs GitHub

Git	GitHub
Version control tool	Hosting platform
Runs locally	Runs online
Tracks file changes	Shares and manages repos
Command-line based	Web interface

GitHub Features

- Repositories
- Pull Requests
- Issues
- Actions (CI/CD)
- Forking
- Collaboration tools

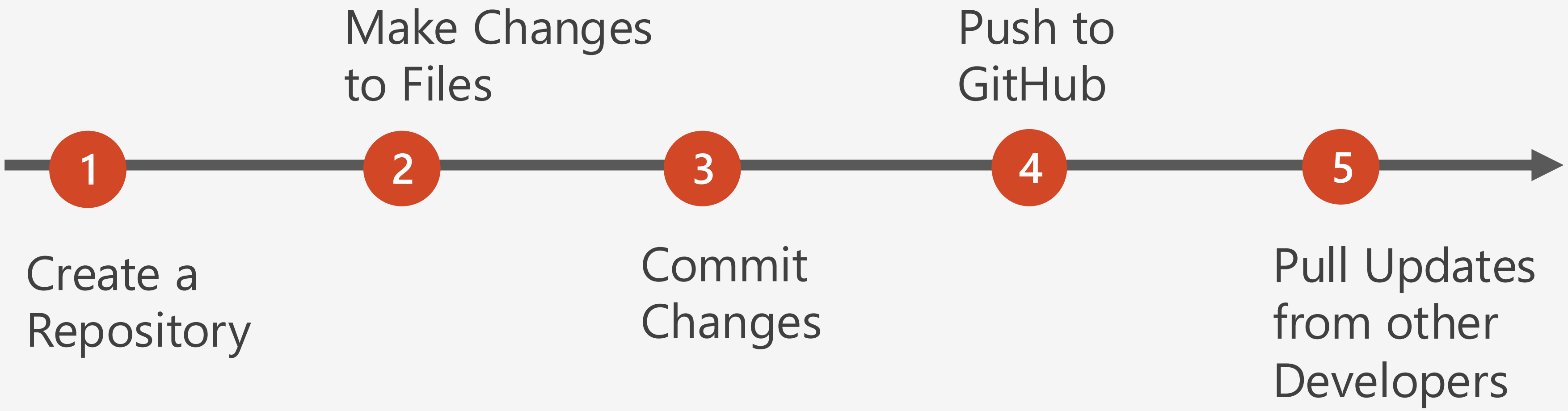
Forking

- **Forking** is the process of creating a personal copy of someone else's repository under your own GitHub account.

- It allows you to freely experiment with changes without affecting the original project.

Forking	Cloning
Creates a copy on GitHub	Creates a copy on your computer
Used for contributing to others' projects	Used for local development
Independent repository	Linked to a remote repository

Basic Git Workflow



1 Create a Repository

2 Make Changes to Files

3 Commit Changes

4 Push to GitHub

5 Pull Updates from other Developers

Git

Surviving with git:

- `cd ~/code/<YOUR_GITHUB_NICKNAME>`
- `git clone <ssh_url> # From GitHub, e.g. github.com/rfalnsari/git`
- `cd <repo>`

Working on master:

- `git status # Should be clean before you start working`
- `code . # Working in VS Code on those files`

- `git status # What files were added/removed/changed since last commit?`
- `git diff # What lines were added/removed since last commit?`

- `# Creating a new commit (2-step process)`
- `git add <file1> <file2>`
- `git commit -m "A MEANINGFUL MESSAGE"`

- `# Pushing commit to GitHub`
- `git push origin master`

Advanced Git Topics

The following Git concepts will be introduced later in the course, during the project weeks, when you will be working in teams:

- Pull
- Merge
- Branches & GitHub Flow
- Resolving merge conflicts

Set Up Your Device:

macOS: https://www.notion.so/Setup-instructions-for-mac-2e75ea763bc88012a4c9f59b5bd92739?source=copy_link

Windows: https://www.notion.so/Setup-instructions-for-Windows-2f15ea763bc880228a34c81ea20380bc?source=copy_link

Resources

GitHub Docs

GitHub Skills

GitHub Deskstop

Markdown Language