Using the command line, Git, and GitHub

Dr. Daniel Genkins
Digital Library Architect
Digital Lab, Heard Libraries

What is the command line?

- A text-based interface used to interact with the computer's operating system
- Allows users to execute commands by typing them
- Command-line interface (CLI) versus graphical user interface (GUI)
- Basic apps: Command Prompt (Windows) and Terminal (MacOS)

Why use the command line?

- Quick execution of tasks without navigating through graphical interfaces
- Automate repetitive tasks with scripts
- Access to a wide range of commands and tools not always available through graphical interfaces
- Perform complex tasks with a combination of simple commands
- Manage remote servers and systems via SSH (Secure Shell)
- Essential for cloud computing and server management
- Gain deeper insight into the operating system and file structure
- Learn fundamental computing concepts and command syntax

macOS Terminal commands

Windows Command Prompt commands

Version control with Git(Hub)

Version control

- A system that records changes to files over time
- Allows you to revert to specific versions later
- Tracks changes and manages multiple versions of a project
- Facilitates collaboration among multiple developers

Types of version control

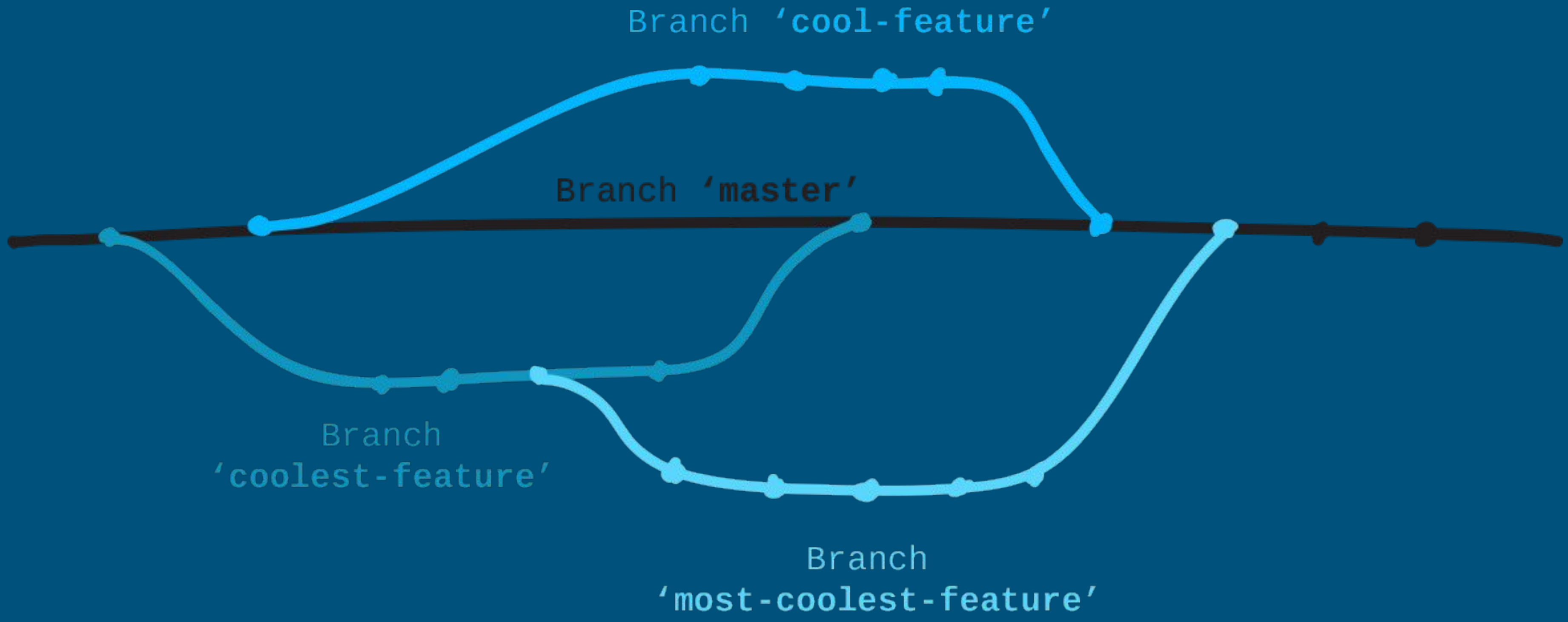
- Local Version Control:
  - Tracks changes on the local machine
  - Simple but not suitable for collaboration
- Centralized Version Control (e.g., Subversion):
  - Uses a central server to store all versions of a project
  - Developers check out and check in changes
- Distributed Version Control (e.g., Git):
  - Each developer has a complete copy of the project repository
  - Enables working offline and facilitates better collaboration

What is Git?

- A distributed version control system designed to handle projects of any size
- Created by Linus Torvalds in 2005
- Speed and efficiency in handling large projects
- Strong support for non-linear development (branching and merging)

Core Git concepts

- Repository
  - A database that stores all versions of project files
  - Can be local or remote
- Commit
  - A snapshot of changes made to the repository
  - Includes a commit message describing the changes
- Branch
  - A parallel version of the repository
  - Allows developers to work on different features independently
- Merge
  - Combines changes from different branches
  - Resolves conflicts that arise from simultaneous edits



Branching with Git

Installing Git on macOS

What is GitHub?

- A web-based platform for hosting Git repositories
- Provides tools for collaboration, project management, and more
- Repository hosting with unlimited public and private repositories
- Integration with other tools and services (CI/CD, issue tracking)
- Social features like following users and starring repositories

CPBP 8306 Python repository

Basic GitHub workflow

Putting it all together

1. Create a repository on GitHub
2. Clone that repository locally
3. Add a file to the repository
4. Stage your change for commitment and add a commit message
5. Push your changes to GitHub

Do steps 2 through 5 using the terminal!