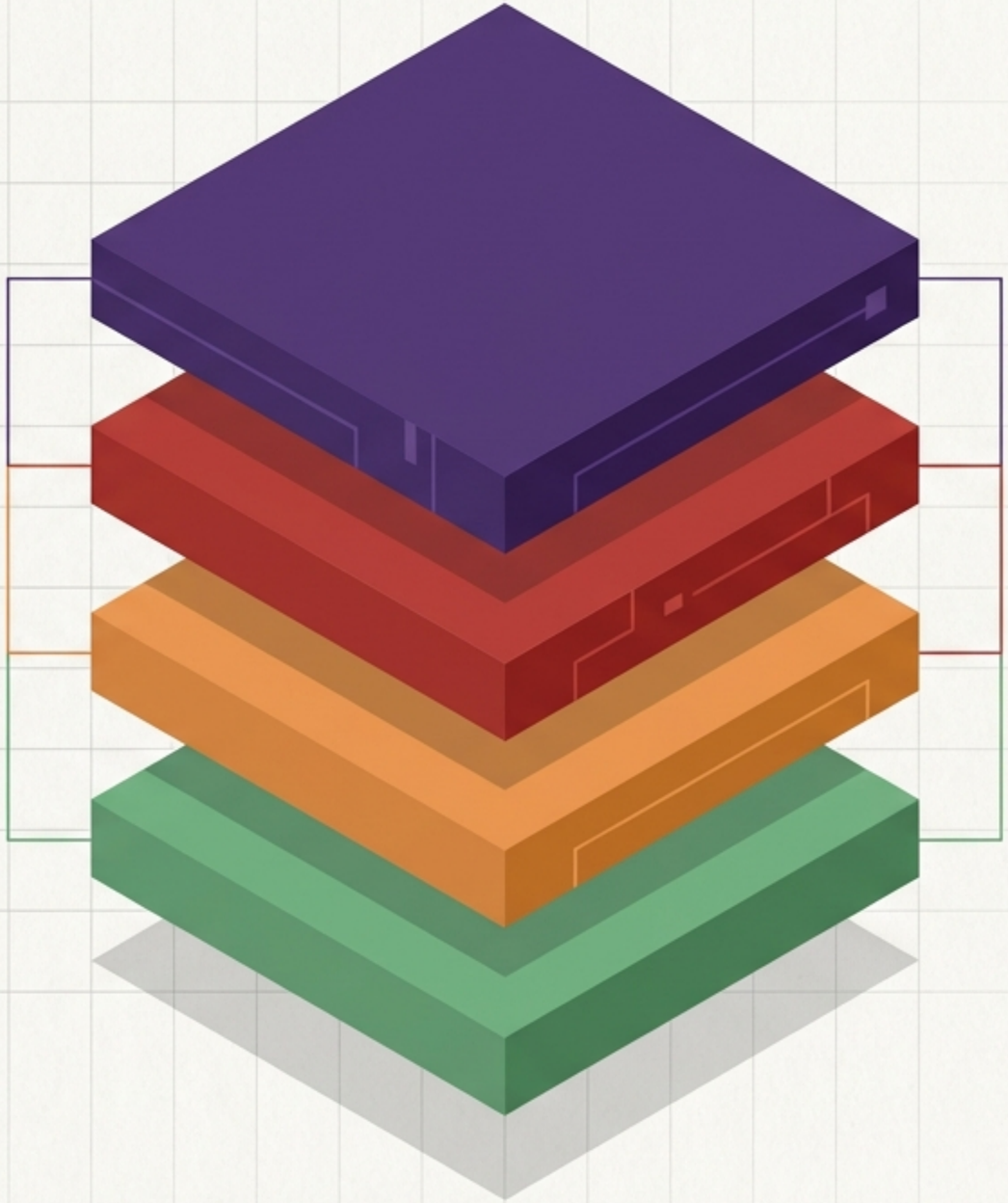
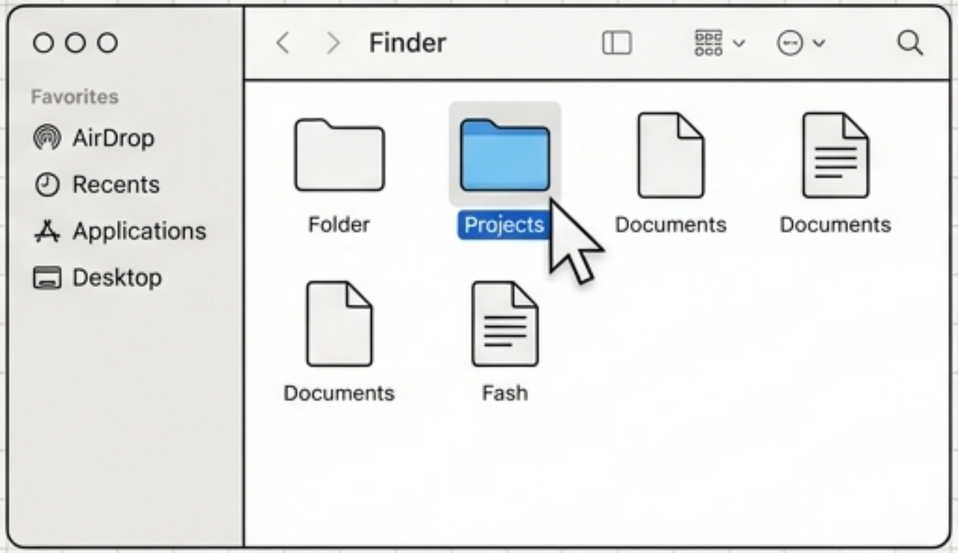
The Developer's Foundation

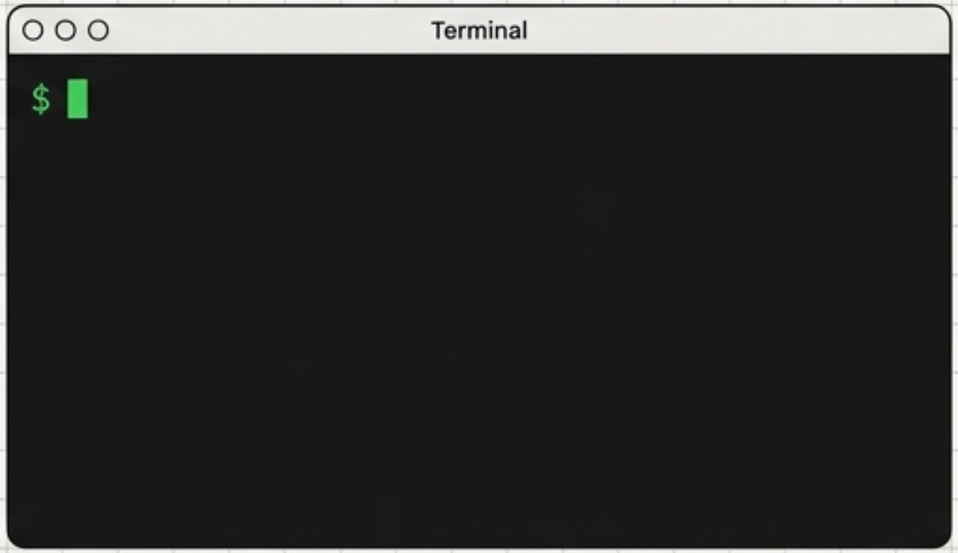
Mastering the Command Line, Homebrew, Git, and GitHub.



Graphical vs. Command Line Interfaces



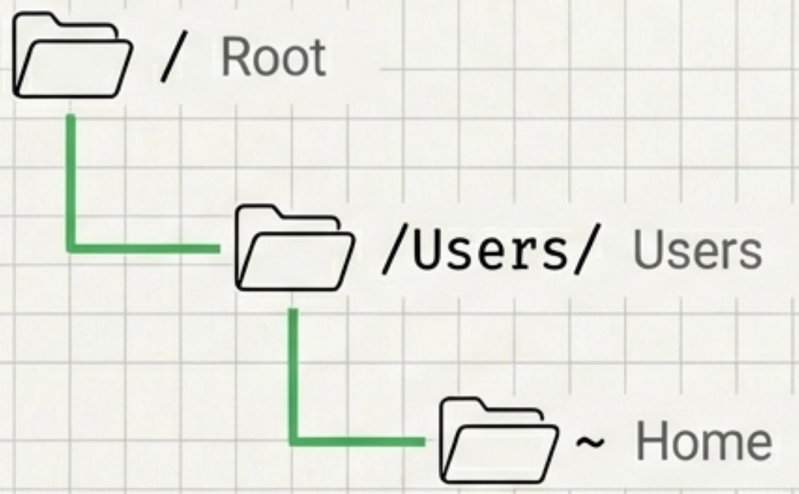
GUI - Graphical User Interface



CLI - Command Line Interface

Both interfaces are shells that manage communication with your computer. A GUI provides a smooth, click-driven experience for end users. A CLI allows developers to directly execute code and automate tasks using text commands.

Navigating the Matrix



`pwd` → **Print Working Directory:** Where am I right now?

`ls` → **List:** What files and folders are in this exact location?

`cd [folder]` → **Change Directory:** Move into this specific folder.

Pro-Tip

Relative vs. Absolute Paths: Absolute paths start from the root (e.g., `/Users/name/project`). Relative paths start from where you currently are. Use `cd ..` to move exactly one folder up the tree, or chain them like `cd ../..` to move two levels up!

File Manipulation Without a Mouse

```
$ mkdir project
```
→ Create a new folder named project

```
$ touch index.html
```
→ Create a new, empty file named index.html

```
$ cp file.txt /backup/
```
→ Copy a file to a new location

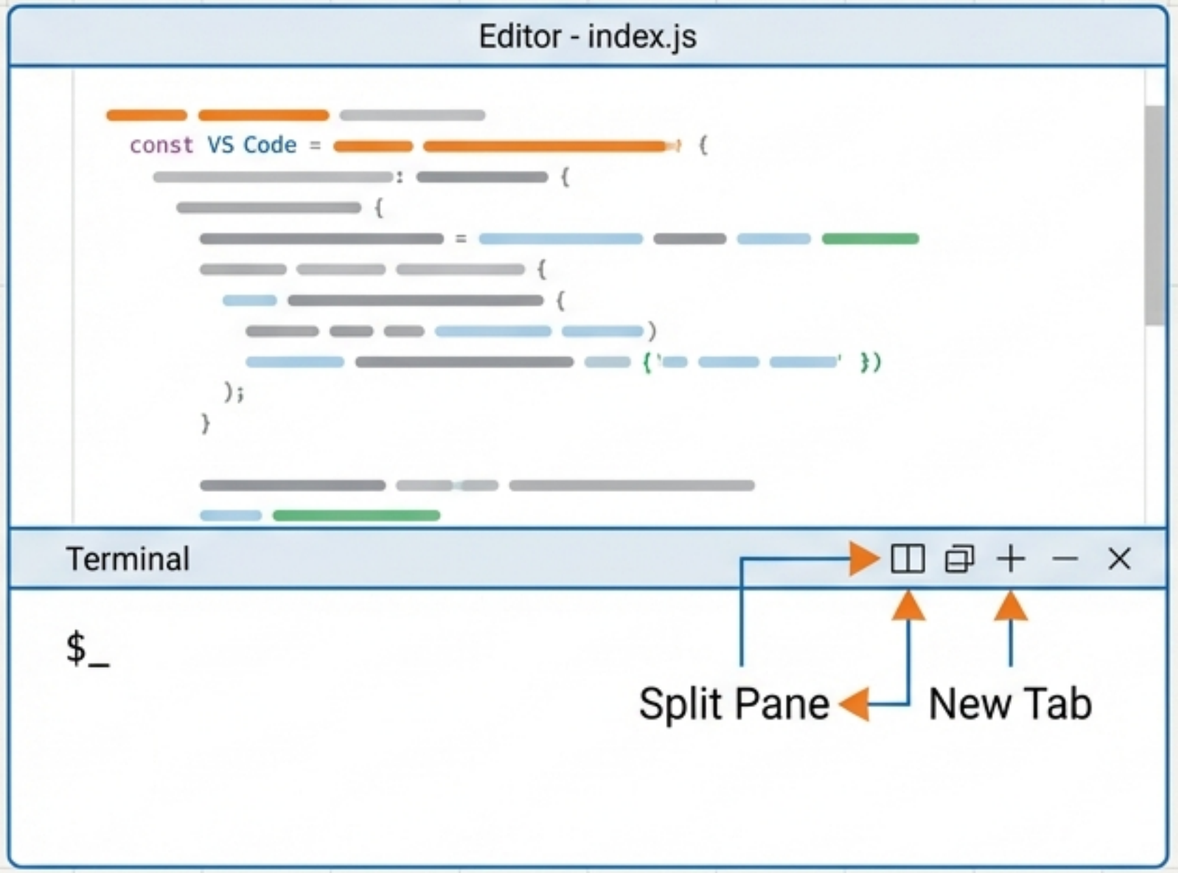
```
$ mv old.txt new.txt
```
→ Move a file (or rename it)

```
$ rm unwanted.txt
```
→ Permanently delete a file

Danger Zone Callout: There is no Trash Can in the terminal. Deletions are permanent.
Never type `rm -rf /` — it will irreversibly delete every file on your computer's hard drive!

The Integrated Workspace

Modern developers rarely leave their editor to use the command line. Visual Studio Code integrates a full-featured terminal directly at the root of your workspace.



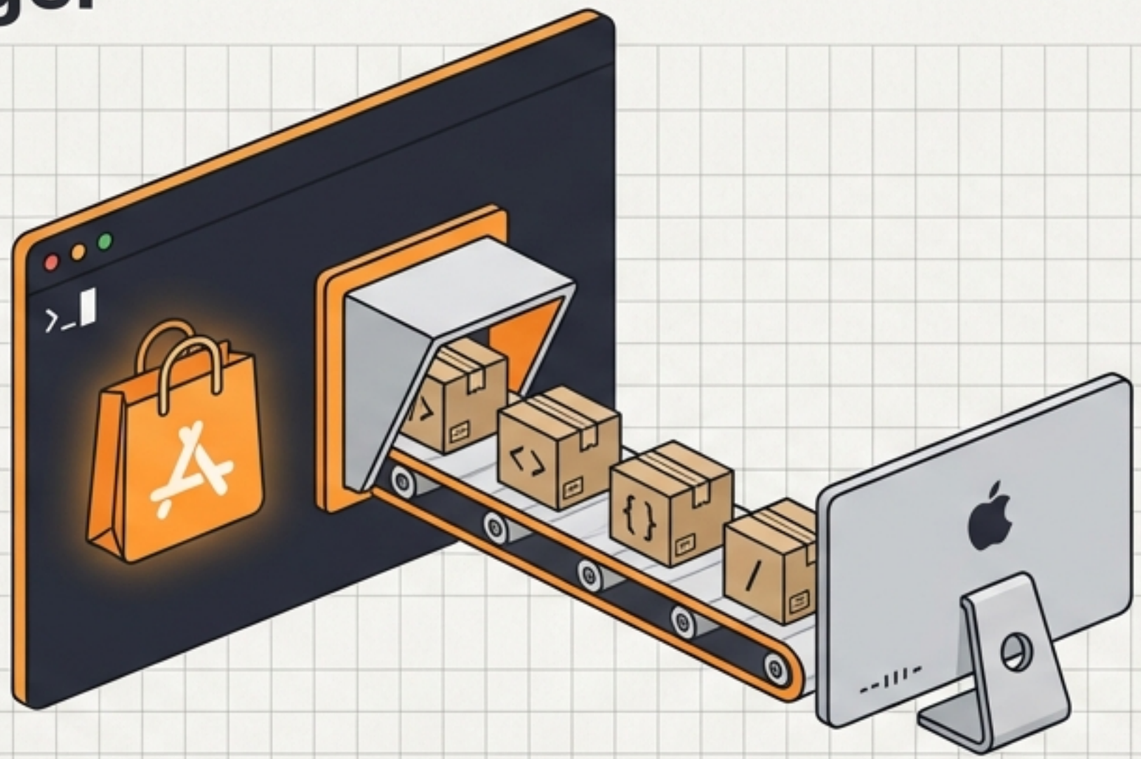
Shortcuts Box

Toggle Terminal:
^ ` (Ctrl + Backtick)

Split Pane:
⌘ \ (Cmd + Backslash)

New Tab:
^ ⇧ ` (Ctrl + Shift + Backtick)

Homebrew: The Missing Package Package Manager



macOS does not ship with a native package manager. Homebrew fills this gap, allowing you to install, update, and manage free and open-source software, developer tools, and desktop applications directly from your terminal.

Brewing Software

Command Line Tools (Formulae)

```
> brew install tree
```

Downloads and installs command-line software.

Desktop Apps (Casks)

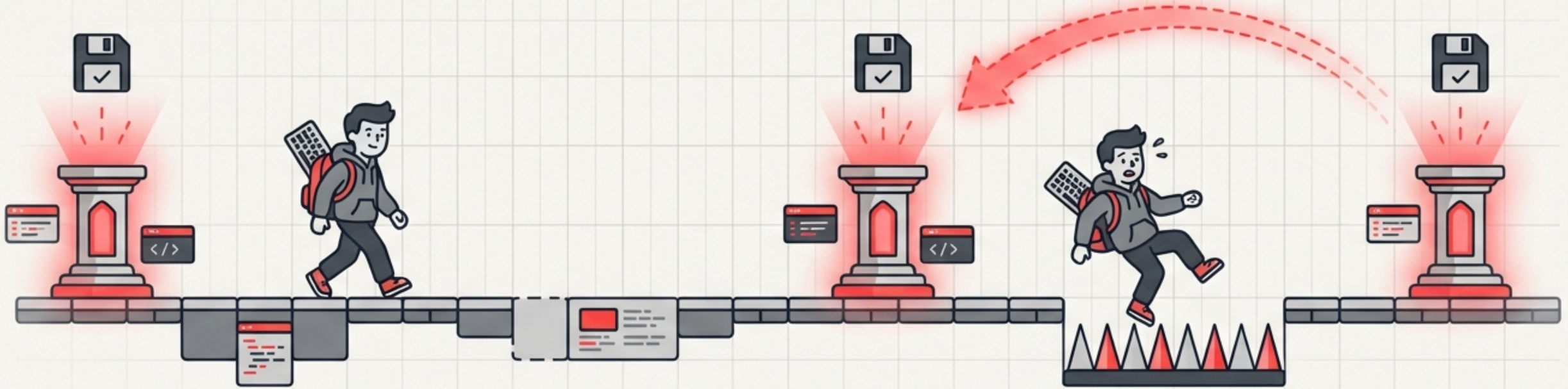
```
> brew install --cask visual-studio-code
```

Automates downloading the .dmg or .zip file and moves the application directly into your macOS Applications folder.

Pro-Tip: Keep your system healthy. Run `brew doctor` to check for potential configuration issues, and `brew upgrade` to update all installed programs to their latest versions.

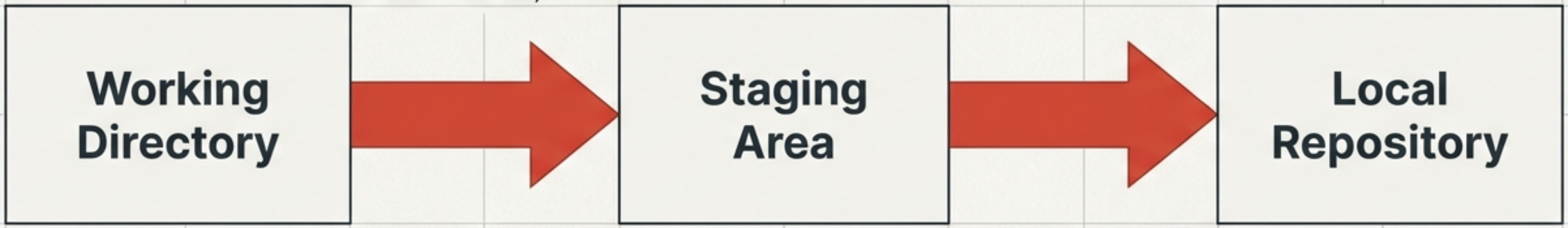
Demystifying Git

Git is a **Version Control System.** Think of it like manual checkpoints in a video game, or a hyper-advanced Google Docs version history. It records all changes you make to your codebase locally, allowing you to revert mistakes and keep track of every iteration of your project.



The Three Stages of Git

2. Stage (`git add .`): You mark specific files to be saved. (Think of this as saying: 'I want to save this character state.')



1. Modify: You edit files in your working directory.

3. Commit (`git commit -m "added jump feature"`): You take a permanent snapshot of the staged files and attach a descriptive message. (Think of this as: 'Confirming the save file.')

Branching & Merging

`git branch [name]` → Create a parallel universe. Safely isolate work on a new feature without breaking the main timeline.

`git checkout [name]` → Switch your active workspace into that parallel universe.

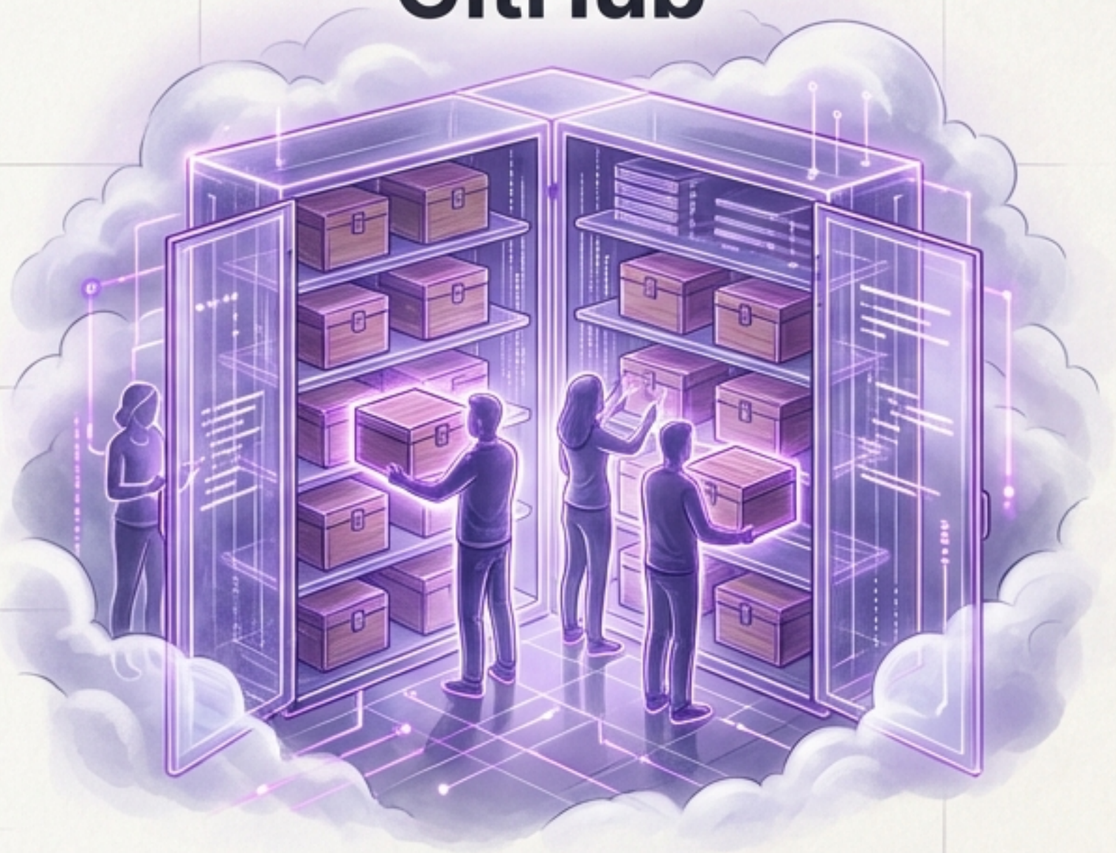
`git merge [name]` → Magically combine the completed feature back into your main timeline, resolving any conflicts.

Git vs. GitHub

Git

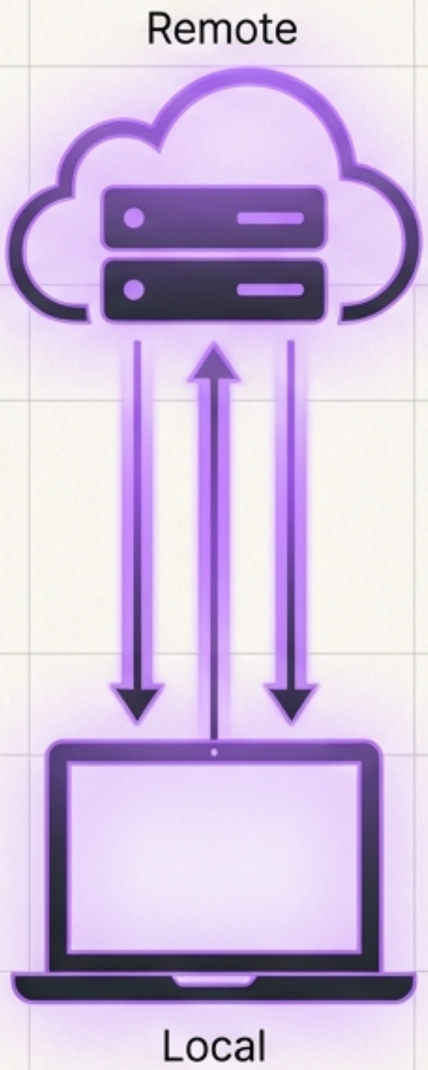
- The local engine running on your machine. It tracks the history of your files.

GitHub



- The cloud hosting service and social collaboration hub. It is the centralized location where developers store their Git repositories to share and back up their work.

Connecting Local to Remote



Starting from scratch

```
gh repo create
```

Use the GitHub CLI to instantly initialize a local repository and push it to the cloud in one interactive step.

Borrowing existing work

```
git clone [URL]
```

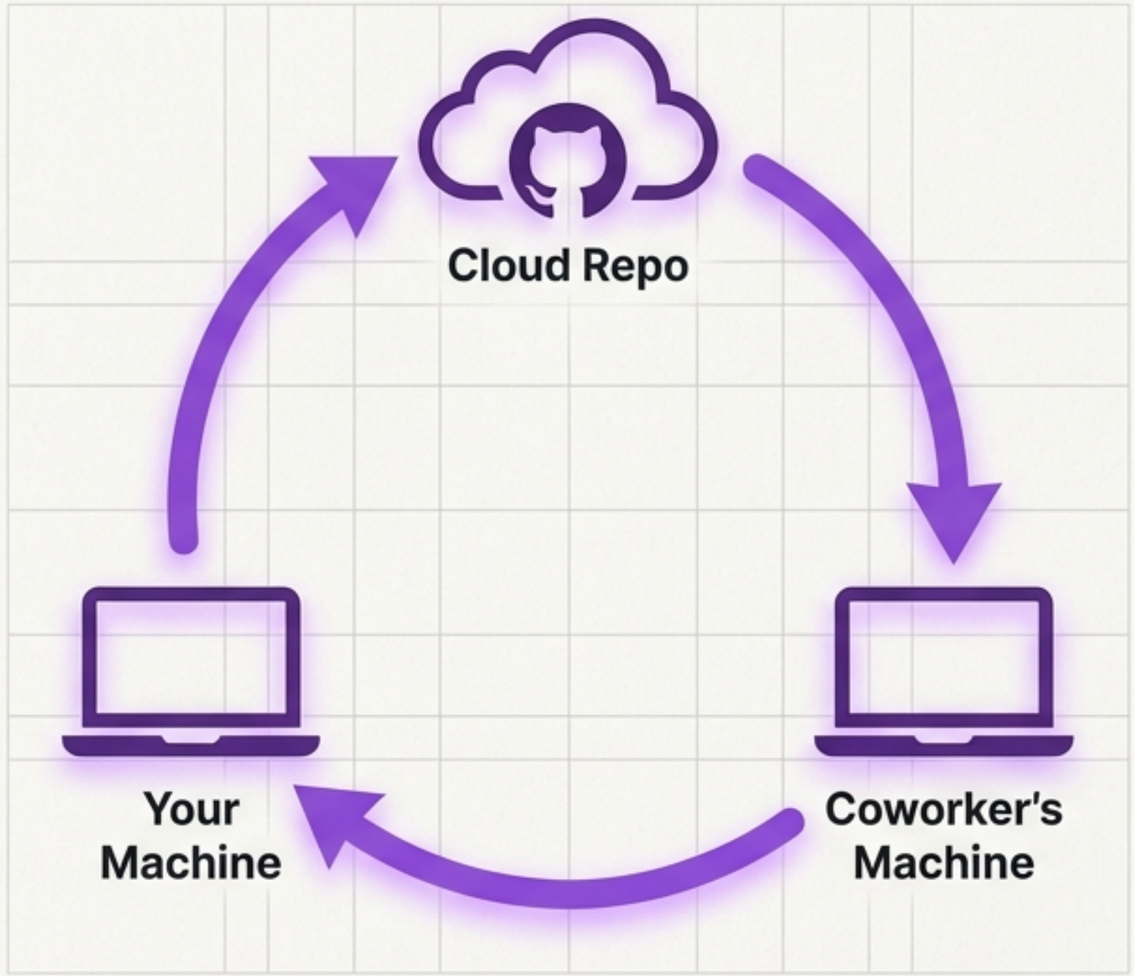
Copy an existing save file from your friend's cloud repository down to your local machine so you can look around and experiment.

Manual connection

```
git remote add origin [URL]
```

Manually link a pre-existing local folder to an empty GitHub repository.

The Collaboration Loop



`git push` → Upload your local commits (save points) to the cloud repository.

`git pull` → Fetch and merge any new updates your teammates have added to the cloud down to your local machine.

The Pull Request (PR):

→ You can't just overwrite your friend's code. A PR is a formal request asking the repository owner to review and accept your proposed changes before they are merged.

Critical Warnings & Troubleshooting

Never git add, commit, or push sensitive information. If you accidentally upload passwords or API keys to a remote repository, they are instantly compromised.

When you get lost:

`git status`	Where the hell am I? What files have I changed? Shows your current branch and staged files.
`git log`	Shows the complete history of your repository, including all commit messages and unique SHA-1 hash identifiers.

The Ultimate Developer's Cheat Sheet

Terminal

`pwd`
print path

`ls -a`
list all

`cd`
change dir

`mkdir` - make dir

`touch` - make file

`rm` - remove

Homebrew

`brew install`
- add CLI package

`brew install --cask`
- add GUI app

`brew doctor`
- diagnostics

`brew upgrade`
- update all

Git (Local)

`git init`
- start repo

`git status`
- check state

`git add .`
- stage all

`git commit -m`
- save snapshot

`git branch`
- list branches

GitHub (Cloud)

`git clone`
- download repo

`git remote add`
- link to cloud

`git push`
- upload changes

`git pull`
- download changes

`gh repo create`
- CLI setup