

GIT / GITHUB / VENVS

ALMOUTHANA TAHA
KHALFALLAH



git

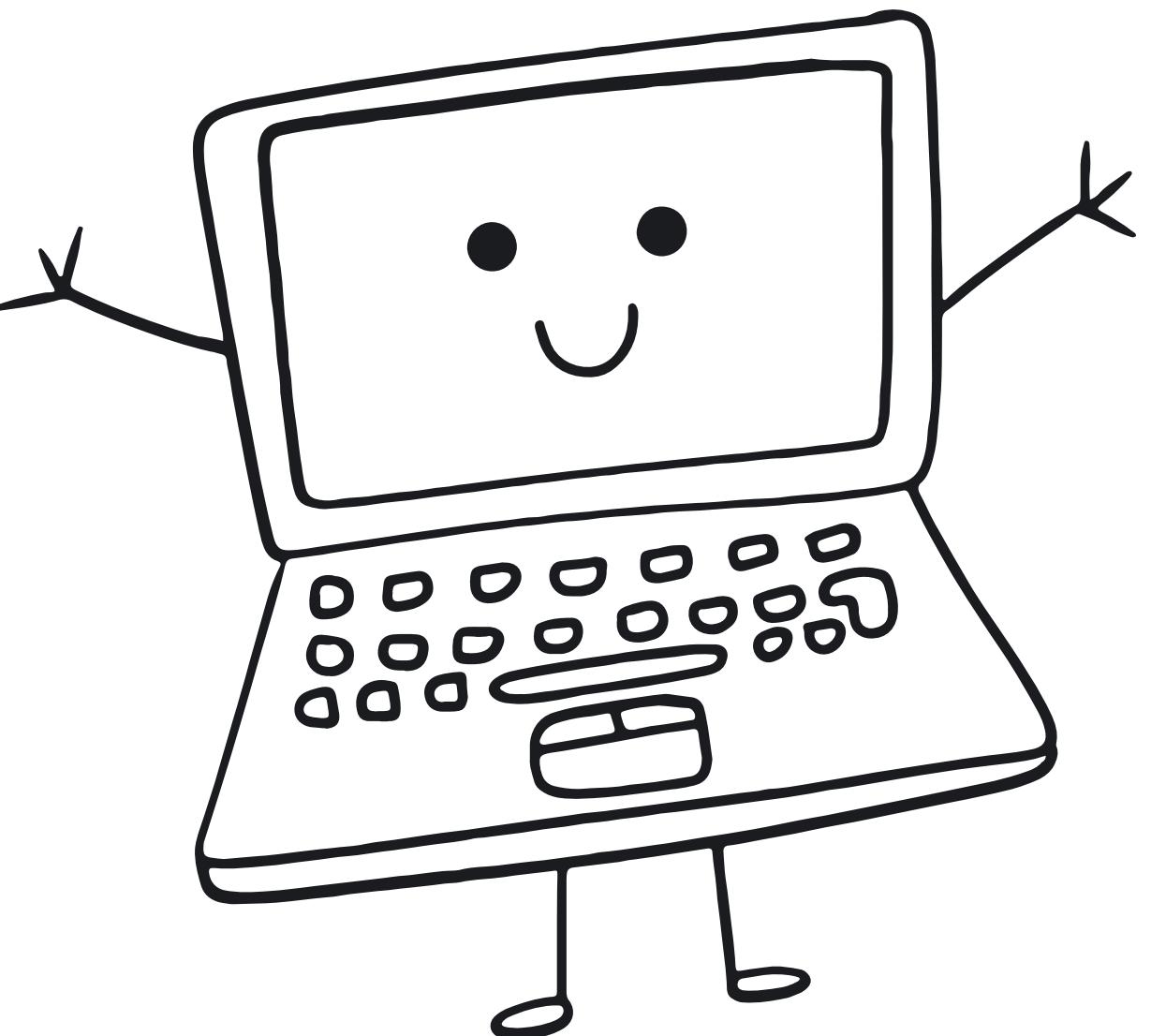


THE WHY BEFORE THE HOW .

WHY THIS MATTERS ?

Pain Points Without Version Control

- **Code breaks when shared**
- **Accidentally overwriting/deleting files**
- **Collaboration chaos**
- **No experiment tracking**
- **Model versioning nightmare**

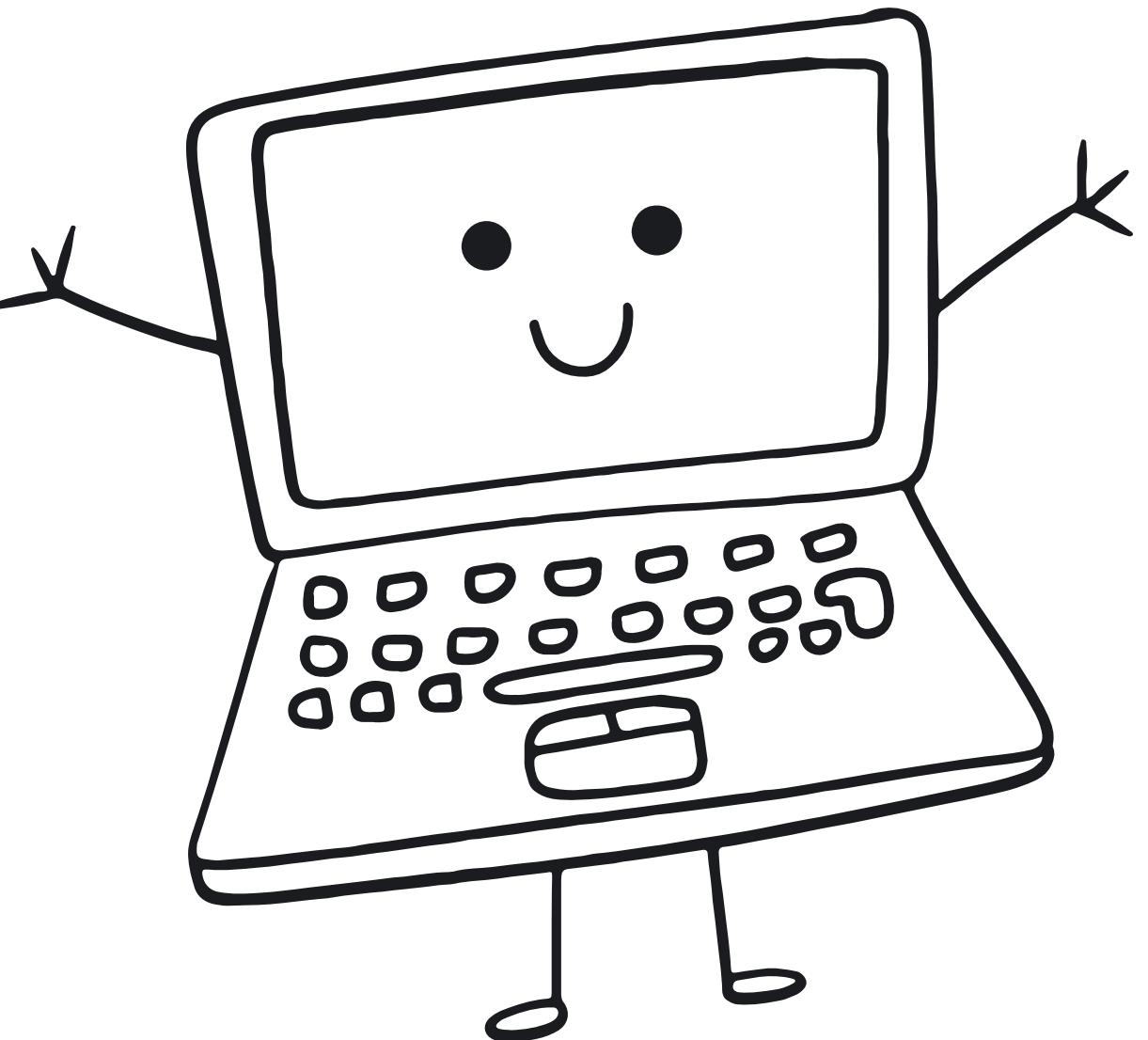


WHY THIS MATTERS ?

(venv)

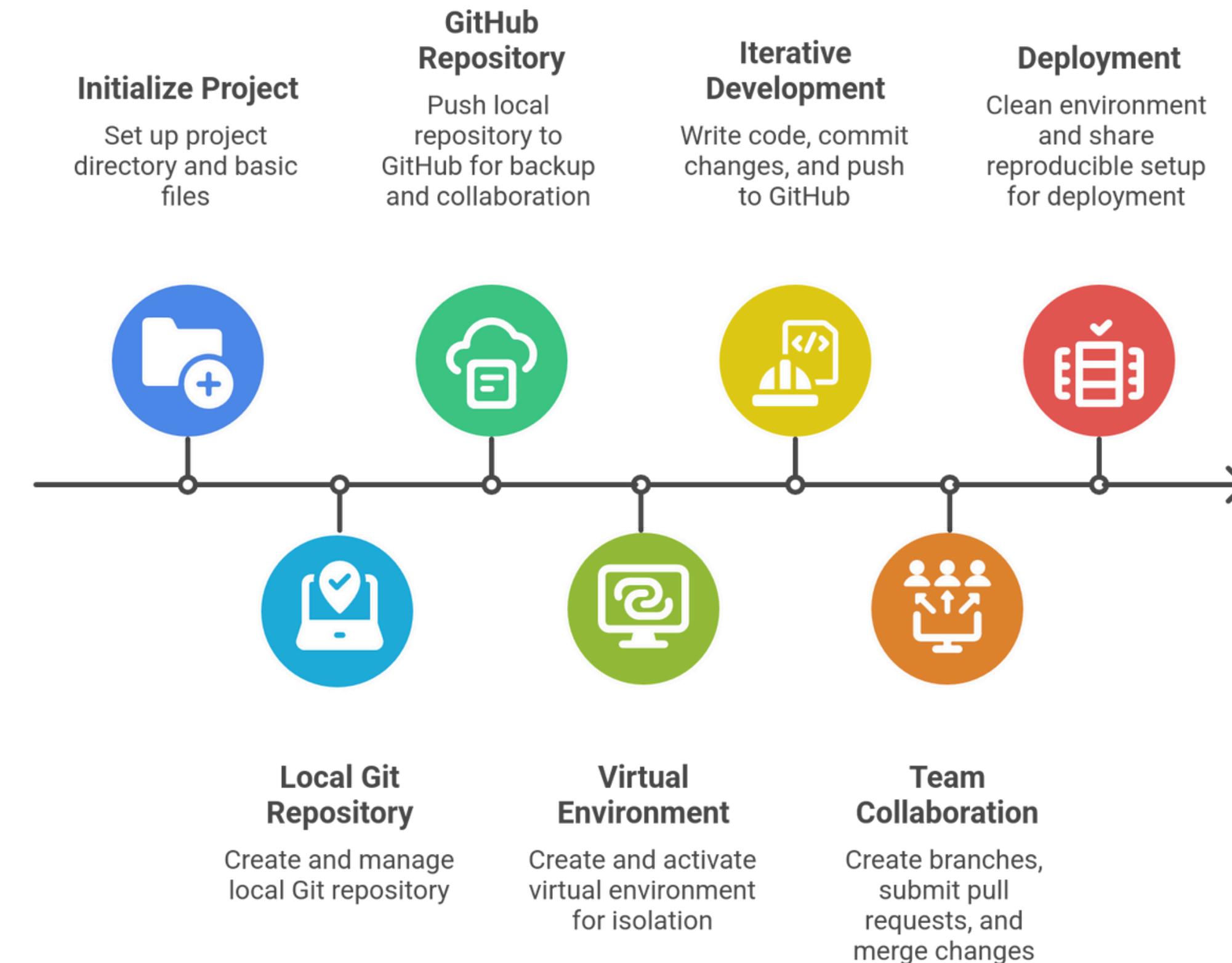
Pain Points Without Virtual Environments

- Dependency hell
- Reproducibility crisis
- Deployment disasters
- Version conflicts



OVERVIEW ON MODERN PYTHON DEVELOPMENT WORKFLOWS :

Software Development Workflow



WHAT WE'LL BUILD TODAY

- Version control concepts & tools
- Git fundamentals + hands-on practice
- GitHub essentials + collaboration workflows
- Linking Git & GitHub together
- Python projects & virtual environments
- Virtual environment managers (focus on UV)
- VSCode integration (terminal + extensions)

Deliverables:

- Git cheat sheet
- GitHub cheat sheet
- Python project best practices guide
- Virtual environment & package management cheat sheet
- Full documentation & practice exercises



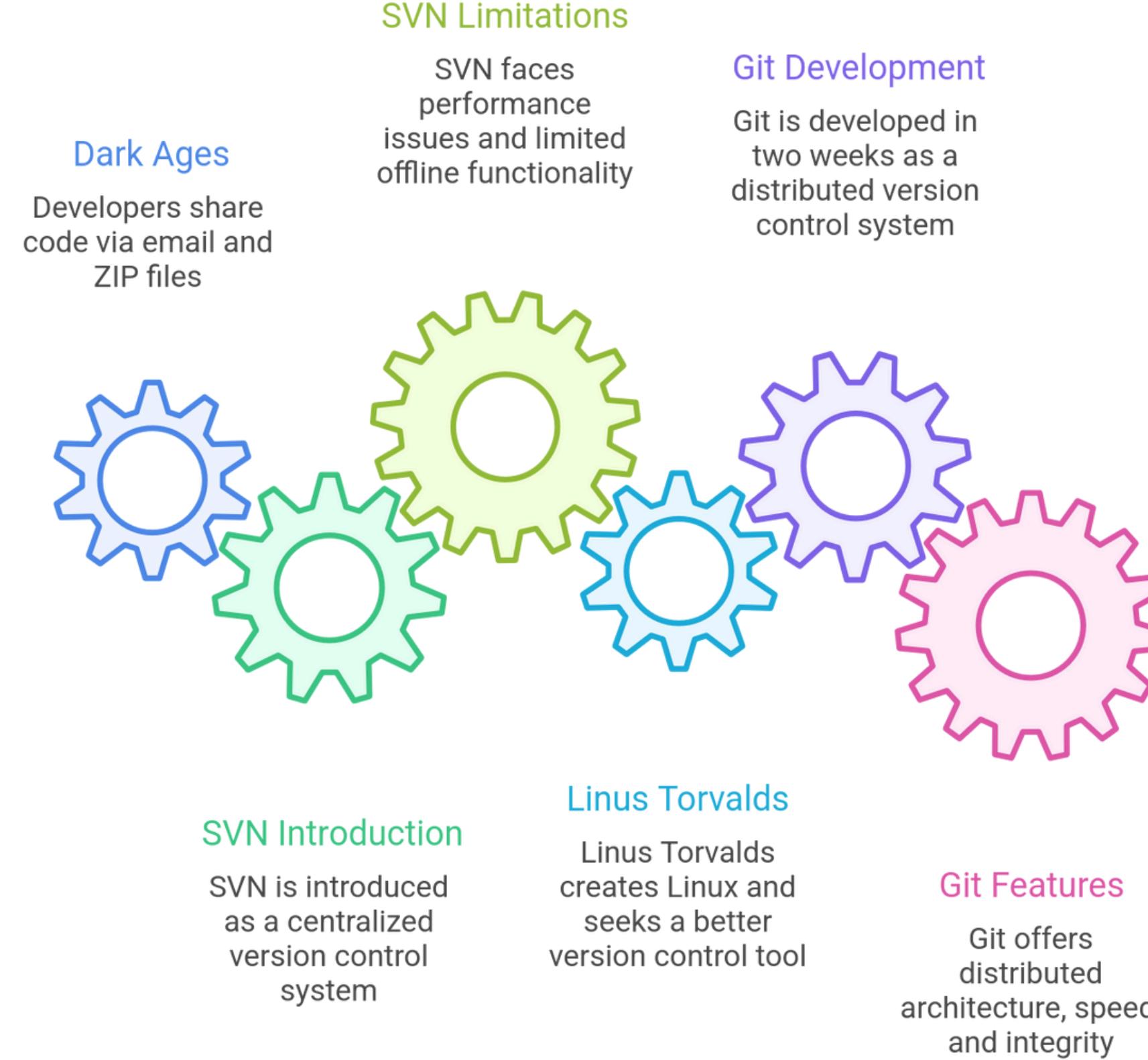


git

GIT - "YOUR TIME MACHINE FOR CODE"



Evolution of Version Control Systems



Git Usage and Repository Growth

Git Commands

Over 150 Git commands available

Linux Kernel Contributions

Over 30,000 contributions to the Linux kernel

Stack Overflow Review

93.87% of developers use Git



Linux Kernel Commits

Over 1 million commits to the Linux kernel

GitHub Repository Growth

Over 500 million repositories on GitHub

GIT VS. ALTERNATIVES

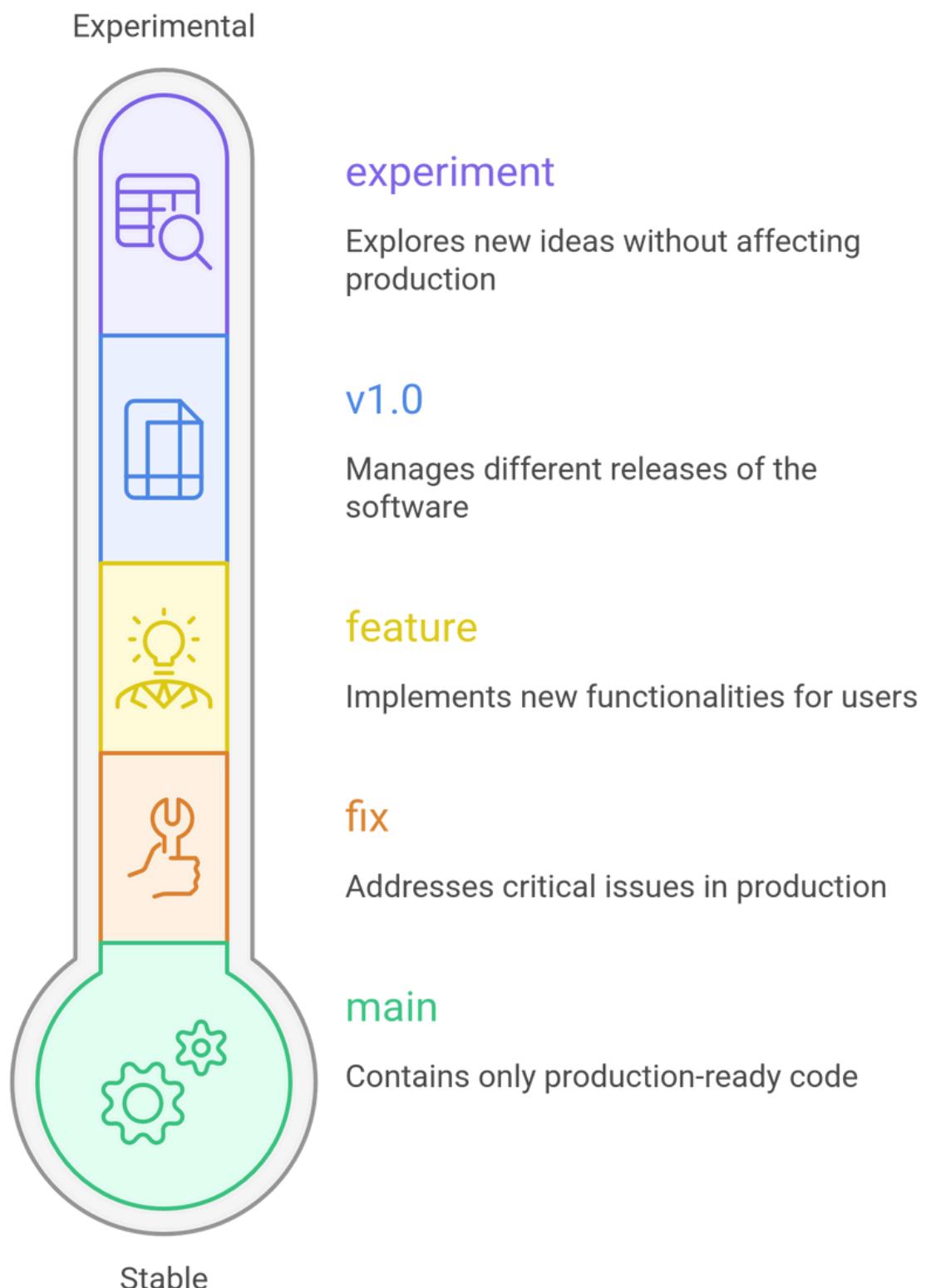
Version Control Tools Comparison

Tool	Git	Mercurial	SVN	Perforce
Type	Distributed	Distributed	Centralized	Centralized
Pros	Fast, powerful, industry standard	Simpler than Git	Simple, linear	Handles huge binary files
Cons	Steep learning curve	Smaller ecosystem	Single point of failure	Expensive, complex
When to Use	Always (99% of cases)	Legacy projects	Rarely (legacy)	Game dev (large assets)



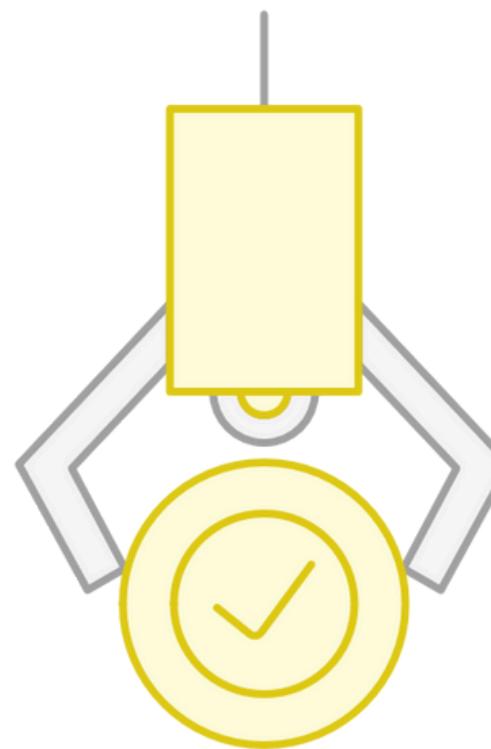
Git branches range from stable to experimental.

WHEN TO BRANCH?



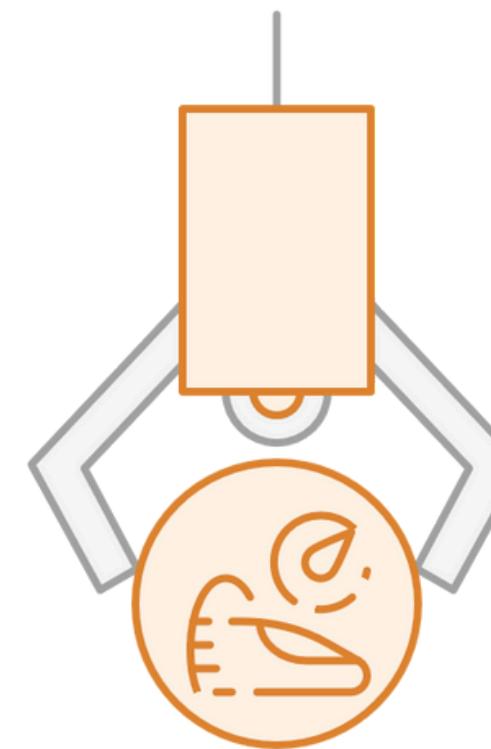
WHEN TO COMMIT ?

Commit guidelines



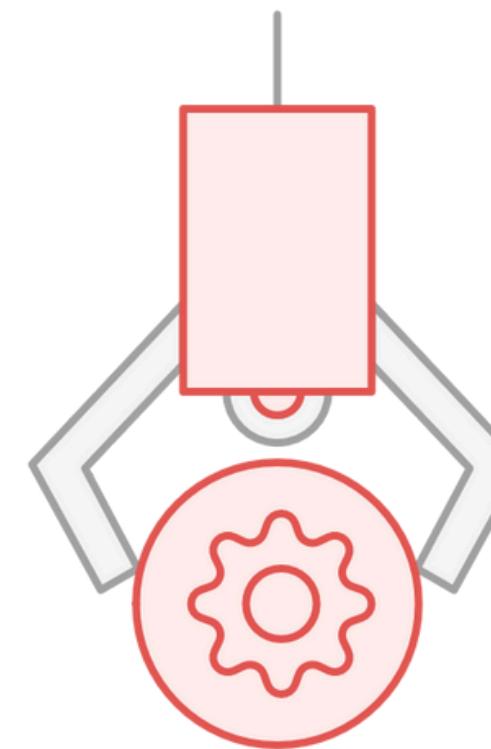
Complete features

Ensure all features are fully implemented.



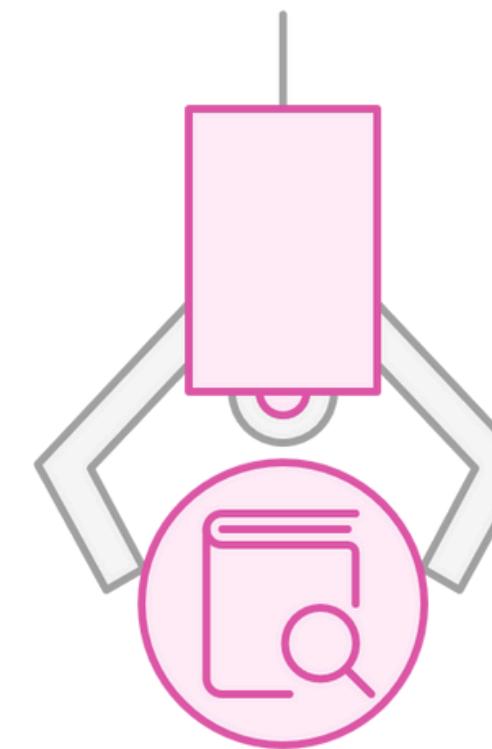
Fixed bugs

Address and resolve any identified bugs.



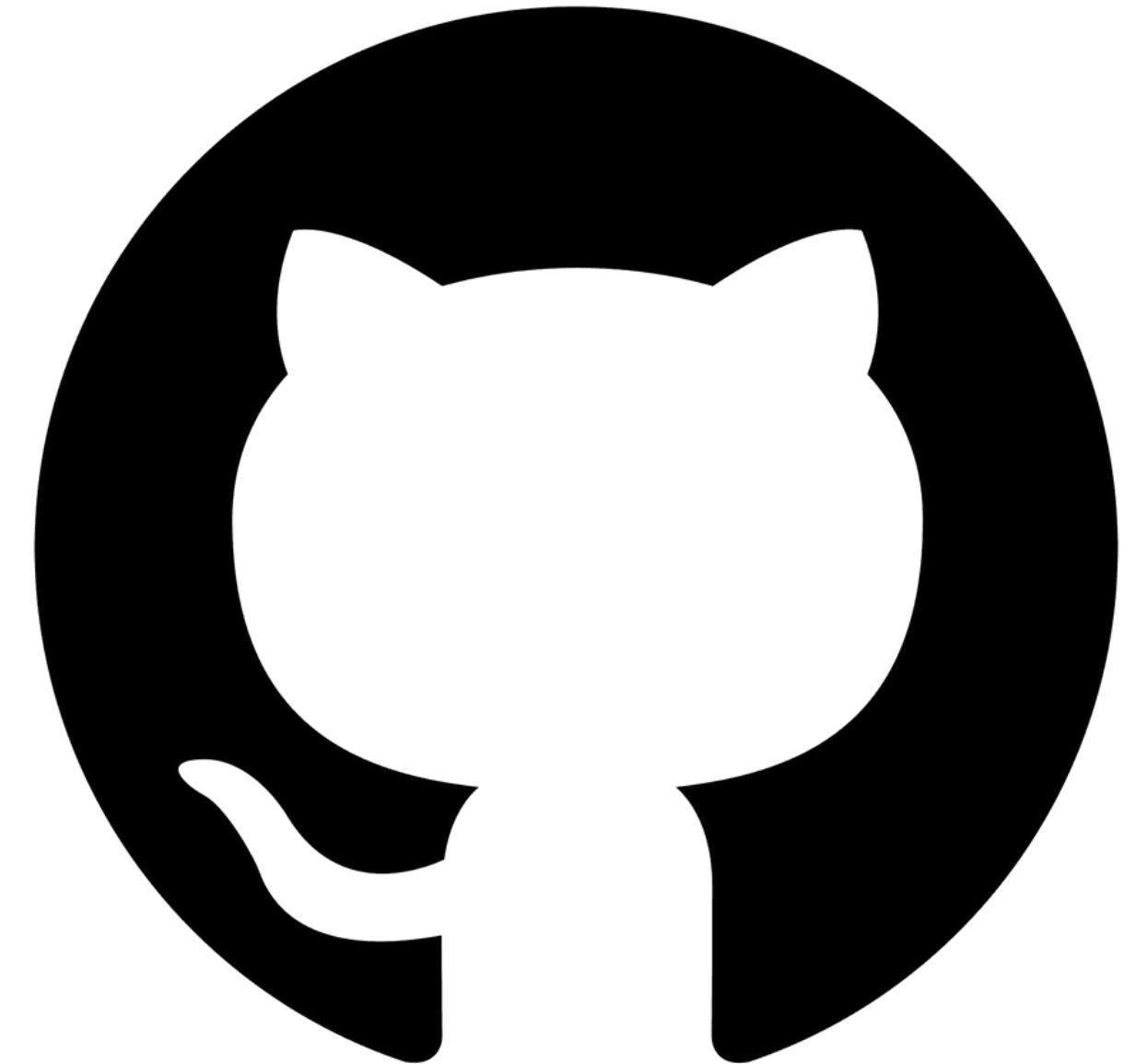
Running code

Verify that the code executes without errors.



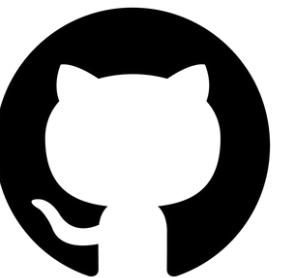
Readme file

Include a `Readme.md` file with project details.



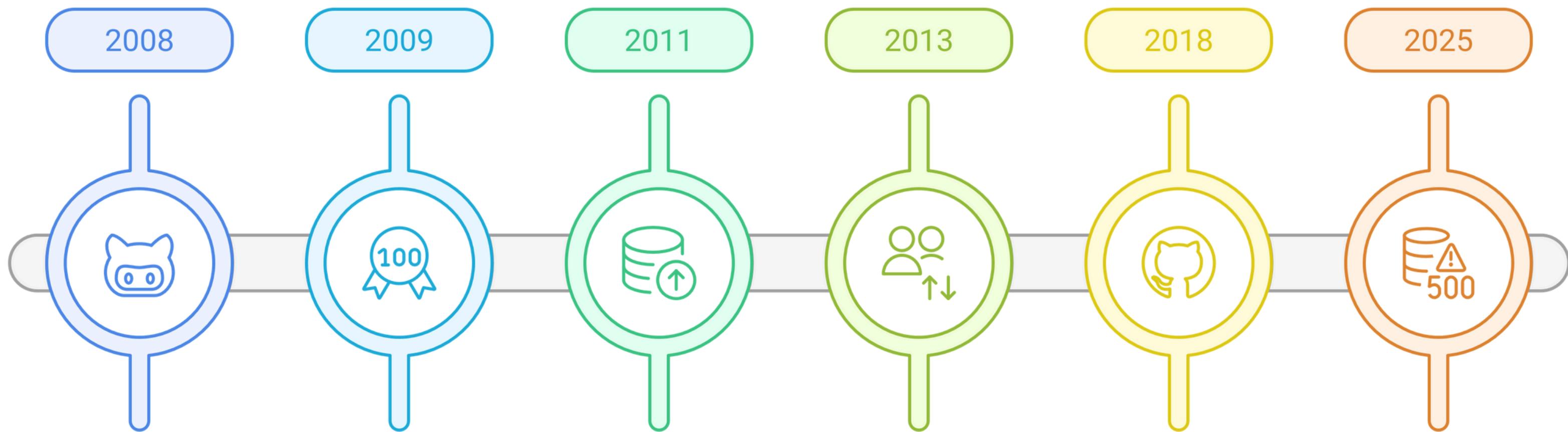
Github

GITHUB - "GIT IN THE CLOUD + COLLABORATION SUPERPOWERS"



GitHub

GitHub's Growth and Impact



GitHub launches

100,000 users /
46,000 repositories

1 million repositories

3 million users

Microsoft acquires
GitHub for \$7.5
billion

180M+ developers,
500M+ repositories

GITHUB VS. ALTERNATI VES



GitHub vs GitLab vs Bitbucket vs Forgejo

Market Share

Best For

Free Tier

CI/CD

Community

UI/UX

Hosting

Philosophy

	GitHub	GitLab	Bitbucket	Forgejo
Market Share	~90% developers	~10%	~5%	Niche (growing)
Best For	Open source, personal projects, portfolios	Enterprise CI/CD, self-hosted	Atlassian ecosystem (Jira)	Self-hosting, privacy, lightweight
Free Tier	Unlimited public/private repos	Generous (self-hosted option)	Limited free features	100% free (self-hosted)
CI/CD	GitHub Actions (powerful)	Built-in (very robust)	Pipelines (integrated)	Forgejo Actions (GitHub-compatible)
Community	Massive (networking effect)	Growing	Smaller	Small but passionate (FOSS-focused)
UI/UX	Clean, intuitive	Feature-rich (can be overwhelming)	Simple	Minimalist, fast
Hosting	Cloud only	Cloud or self-hosted	Cloud or self-hosted	Self-hosted only
Philosophy	Corporate (Microsoft)	Corporate (some FOSS aspects)	Corporate (Atlassian)	100% FOSS, community-driven

WHY GITHUB ?

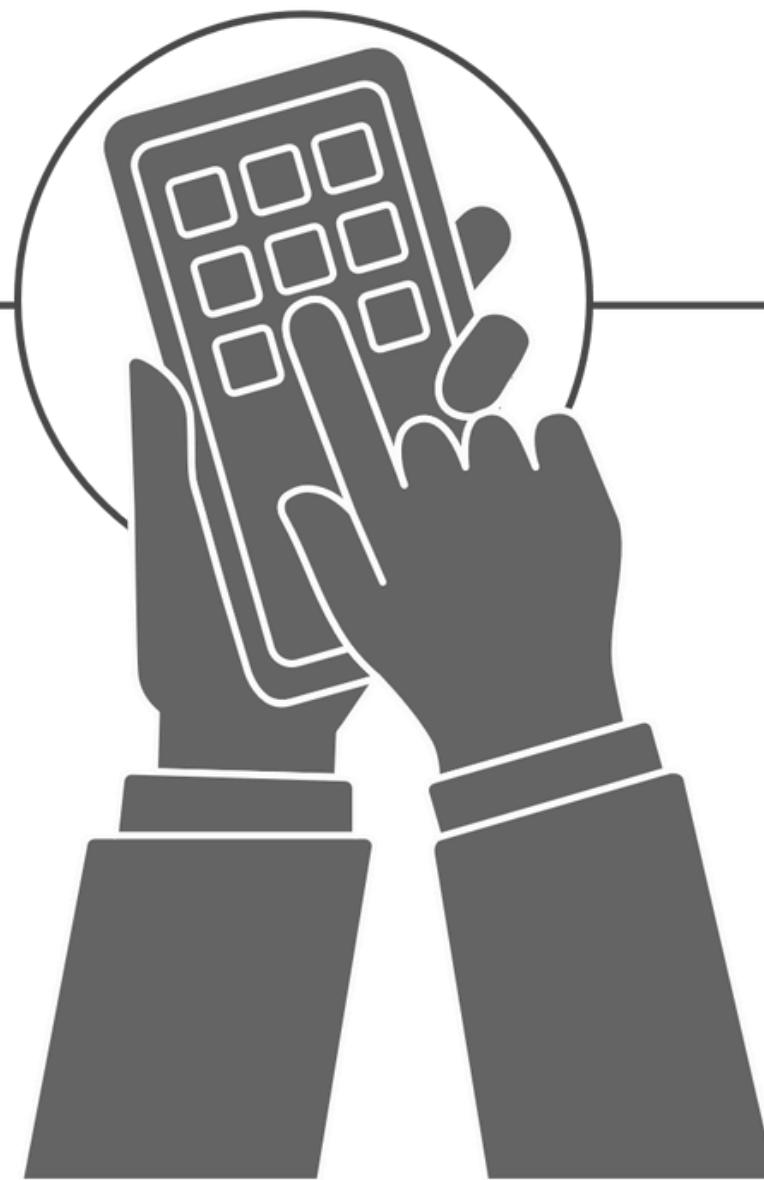


WHY GITHUB ?

Git Commands

git fetch

Downloads changes from a remote repository without applying them to your local branch. Use this when you want to review changes before integrating them.

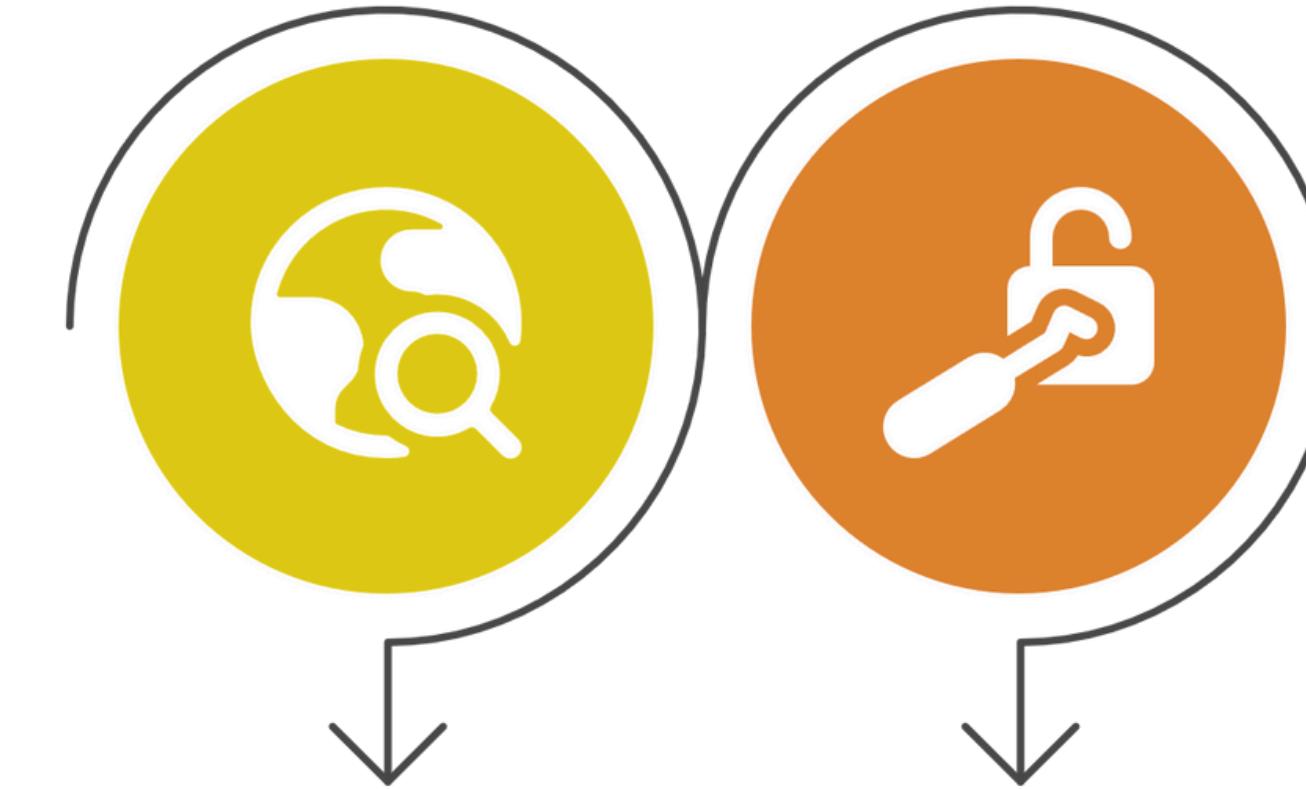


git pull

Downloads changes from a remote repository and immediately applies them to your local branch. This is suitable for solo projects or when you trust the incoming changes.

PUBLIC VS PRIVATE REPOS

Repository Visibility



Public Repositories

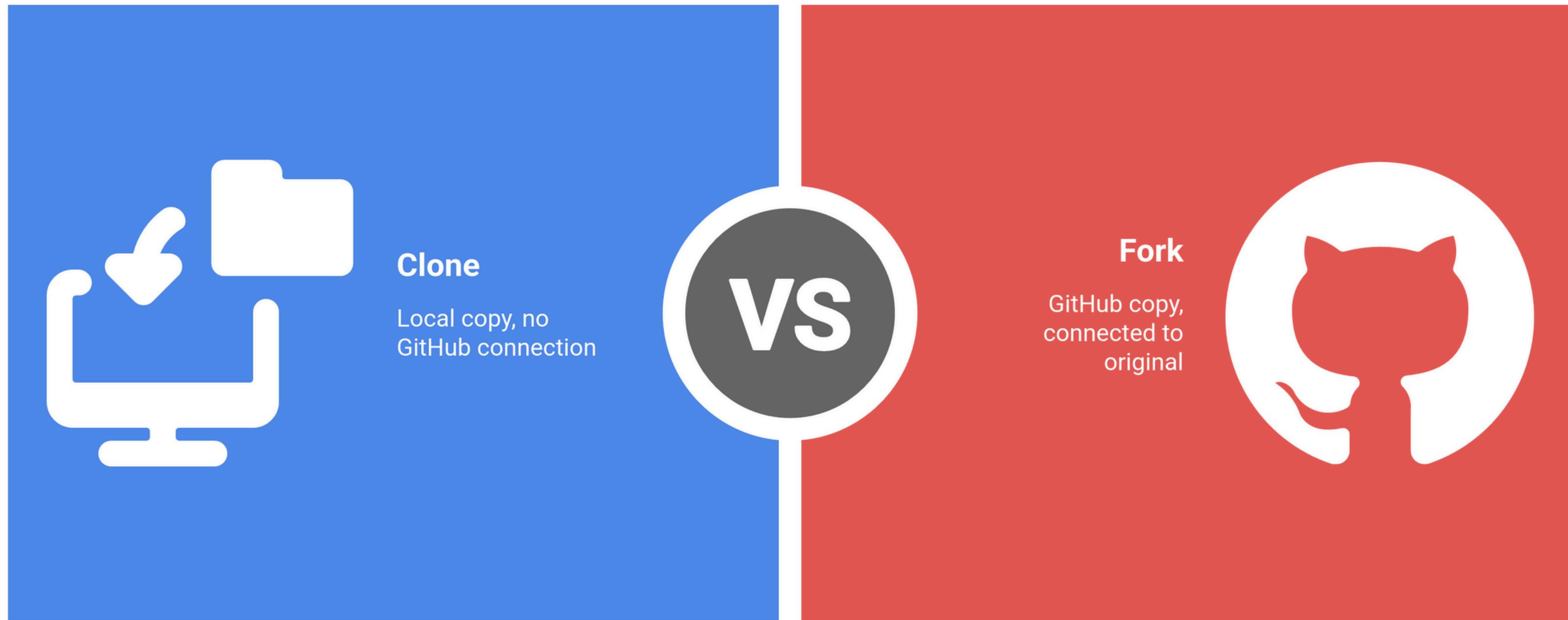
Visible to the entire internet, anyone can clone and view your code. Great for open-source projects.

Private Repositories

Only invited collaborators can access, hidden from public view. Perfect for proprietary code and work projects.

CLONE VS FORK

Choose the appropriate method for copying a repository



REAL WORLD CONTRIBUTION

Contributing to Open Source React

Original React
Repository

Public, controlled by
Facebook

Fork
Repository

Create your own
copy

Clone Locally

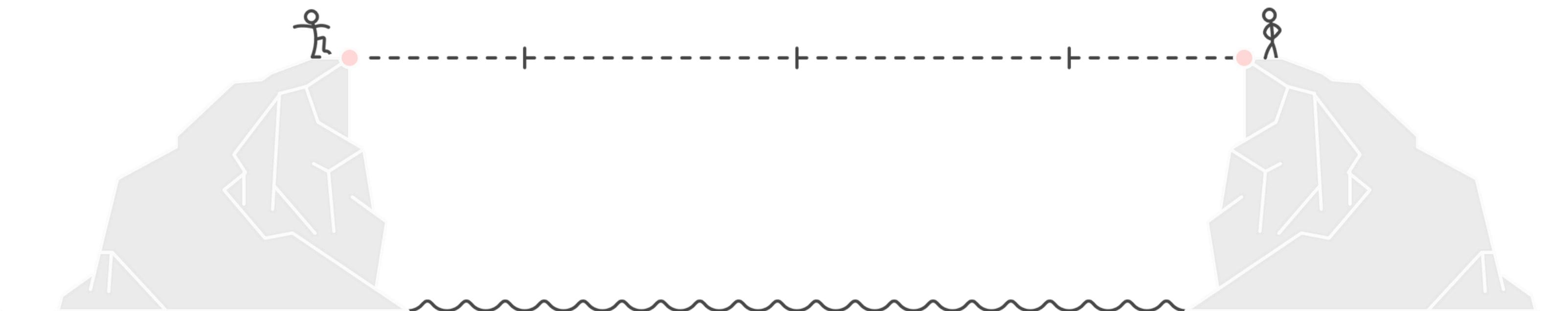
Download to your
computer

Make
Changes

Implement your
contributions

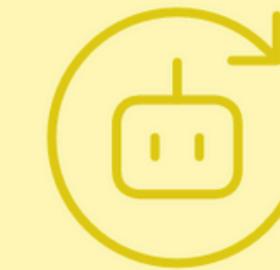
Approved Pull
Request

Contributions merged
into React



GITHUB FEATURES

Github Features



Github Actions

Automate CI/CD workflows for seamless code integration and deployment.



Github Pages

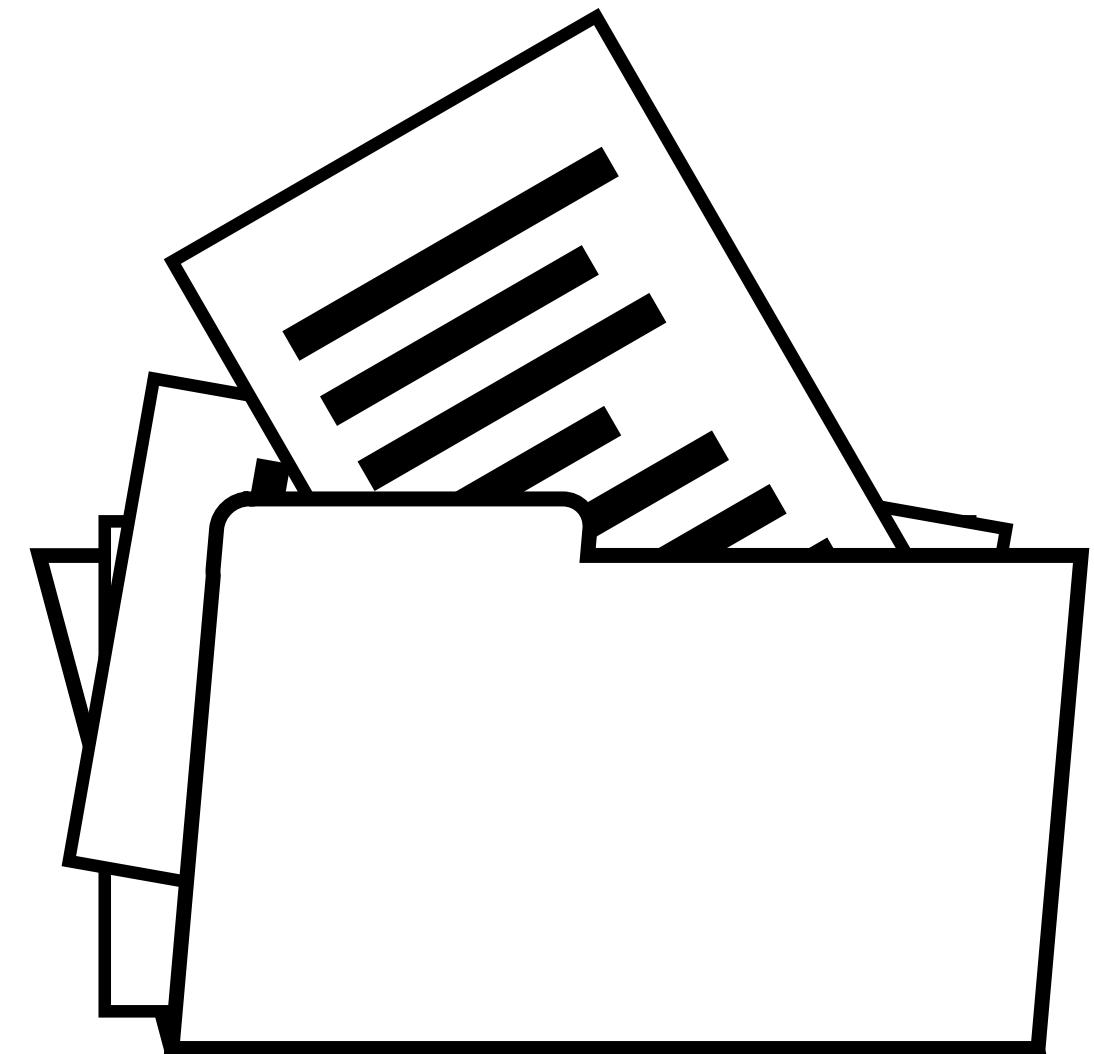
Free hosting service for static websites, ensuring high availability and reliability.



VENV & PACKAGE MANAGEMENT

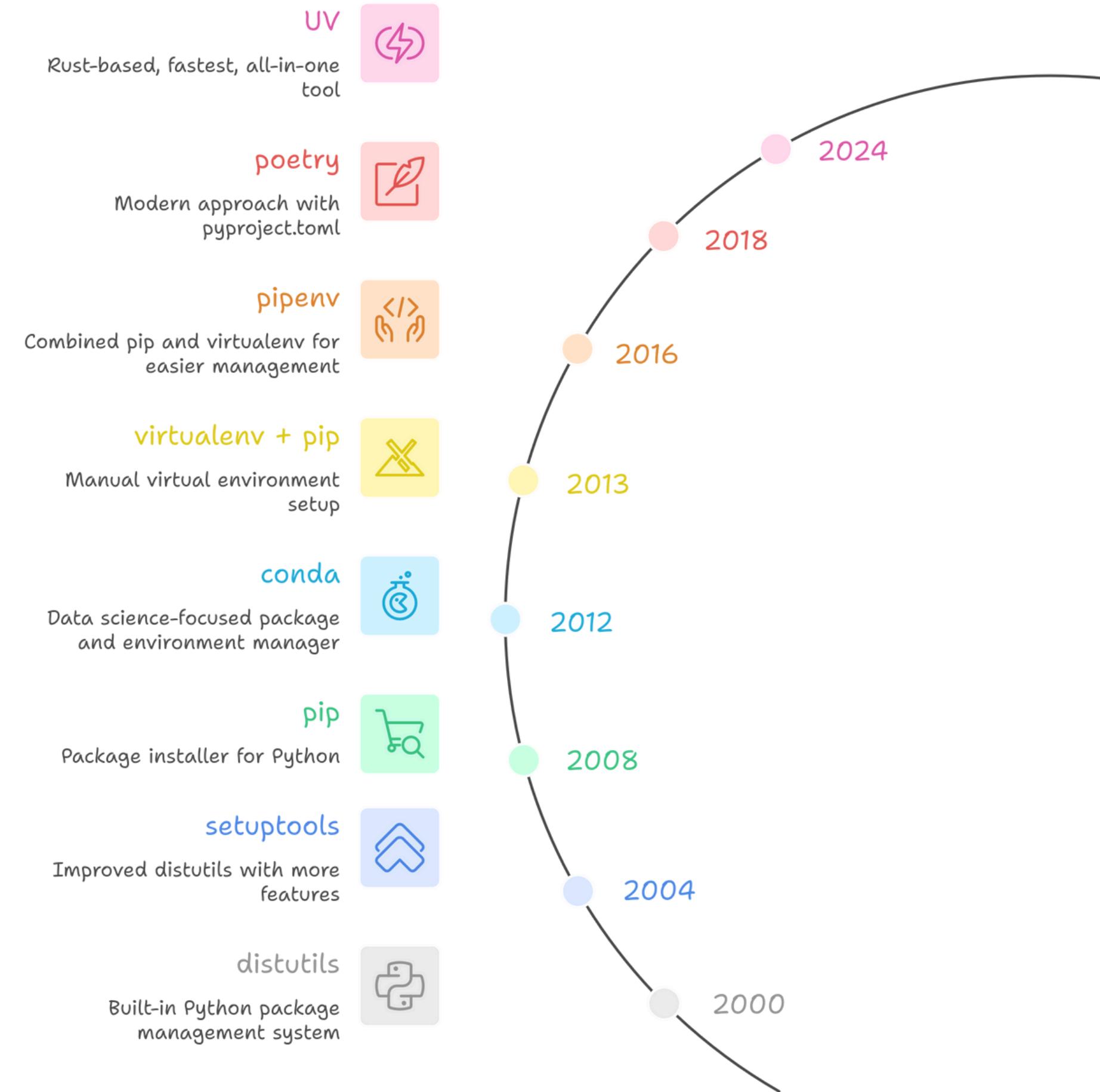
- Python installs packages to ONE global location
- Installing v2.0 overwrites v1.0 completely
- Project A needs pandas 1.0, Project B needs pandas 2.0 = conflict
- Constantly installing/uninstalling packages between projects

The Solution: Virtual Environments



- Isolated "toolboxes" for each project
- Each project has its own packages & versions
- No interference between projects
- Fixes "works on my machine" problem

TIMELINE OF THE EVOLUTION PYTHON MANAGEMENT TOOLS



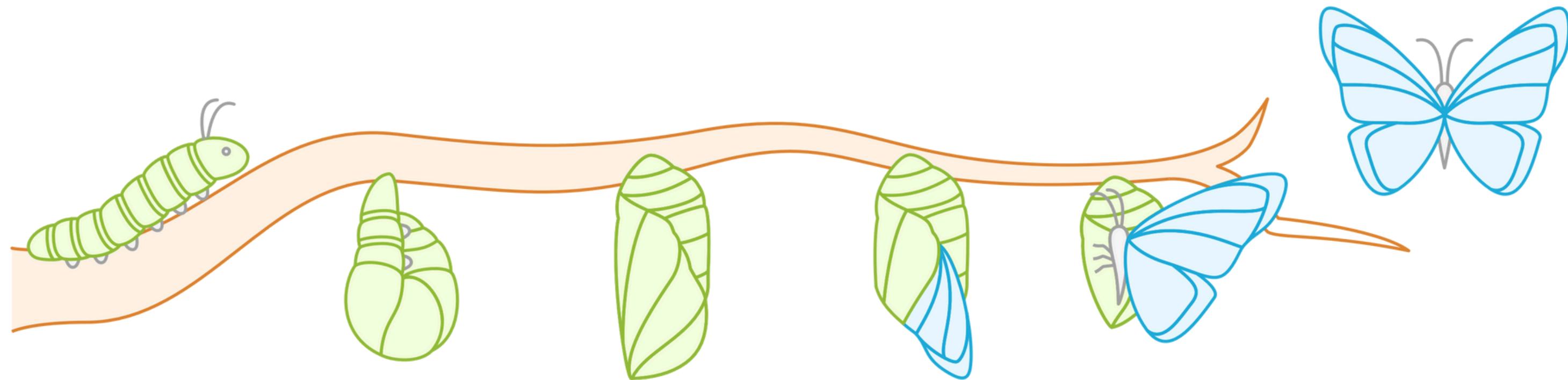
Python Package Manager Comparison

	AVI pip+venv	PIP pipenv	CONDA conda	POETRY poetry	UV uv
Speed	Slow	Very Slow	Slow	Medium	FAST
Lock files	No	Yes	Yes	Yes	Yes
Dependency resolution	Poor	Buggy	Good	Great	Great
Dev dependencies	No	Yes	Partial	Yes	Yes
Auto venv	No	Yes	Yes	Yes	Yes
pyproject.toml	No	No	No	Yes	Yes
Cross-language	No	No	Yes	No	No
Learning curve	Easy	Medium	Hard	Medium	Easy
2026 Recommendation	Legacy	Avoid	Niche	Good	Best



WHY VIRTUAL ENVIRONMENTS FOR AI/DATA SCIENCE?

Virtual Environment Setup



Messy
Dependencies

Project
dependencies are
mixed

Isolate
Dependencies

Create a dedicated
virtual environment

Deploy Exactly

Use production
dependencies for
deployment

Reproduce
Experiments

Ensure consistent
experimental results

Switch Projects

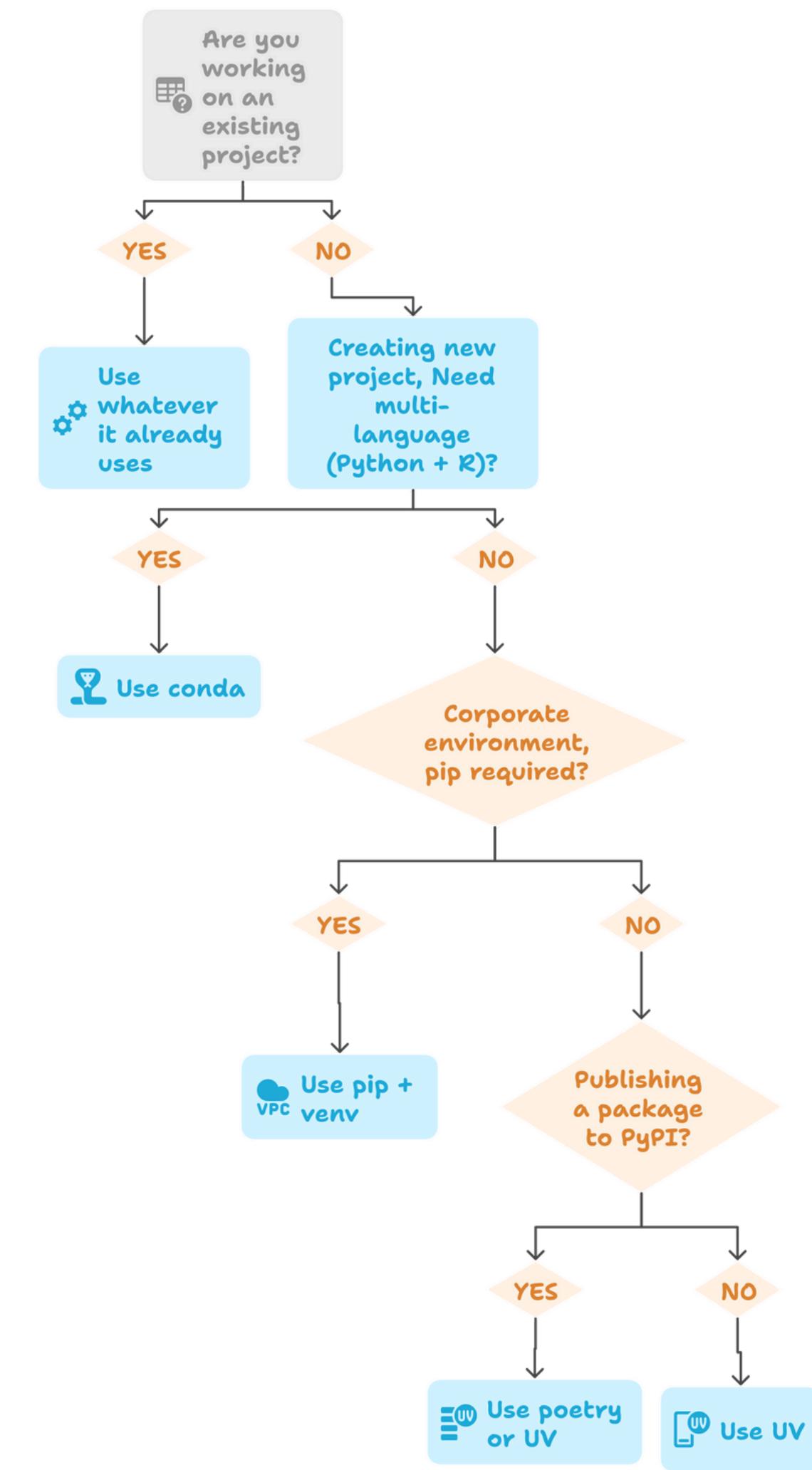
Easily move between
projects without
issues

Clean
Environment

Project
dependencies are
well-managed

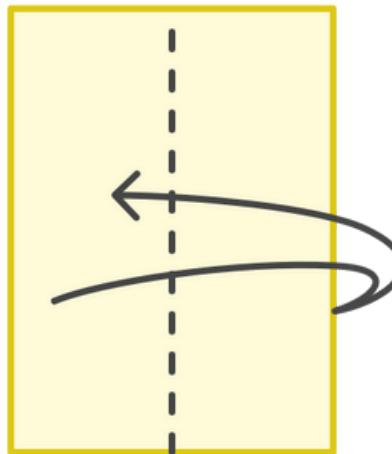
GOLDEN RULE : ONE PROJECT = ONE VIRTUAL ENVIRONMENT

PROJECT SETUP DECISION TREE



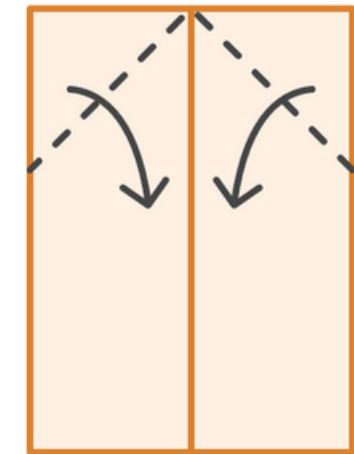
WHAT YOU'VE ACCOMPLISHED ?

From Novice to Git Master



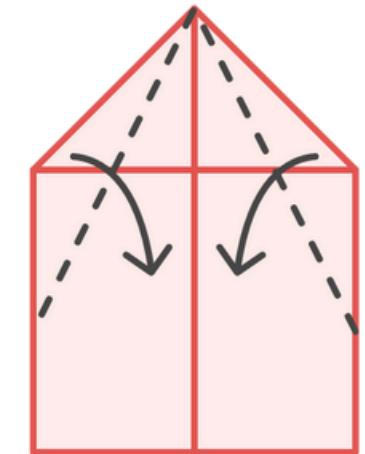
Version Control Novice

Unfamiliar with Git concepts



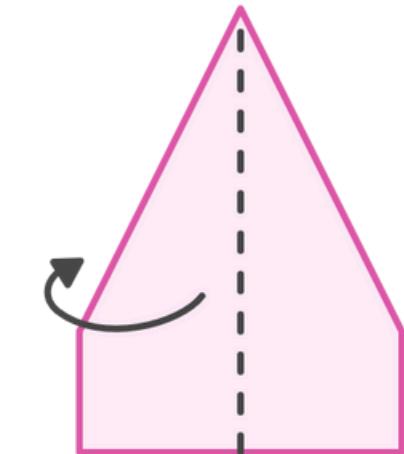
Fundamentals

Learn basic Git commands



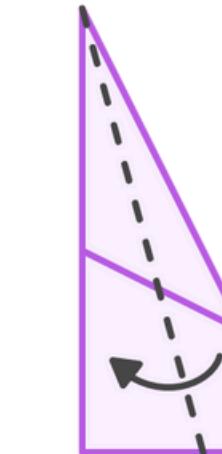
Branching & Collaboration

Work on features and team projects



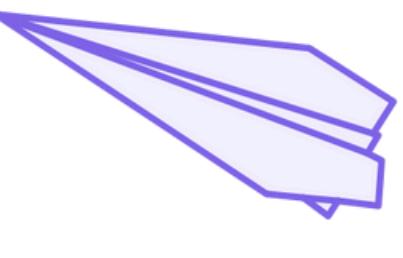
Cloud Collaboration

Use GitHub for remote teamwork



Environment Management

Set up and manage development environments

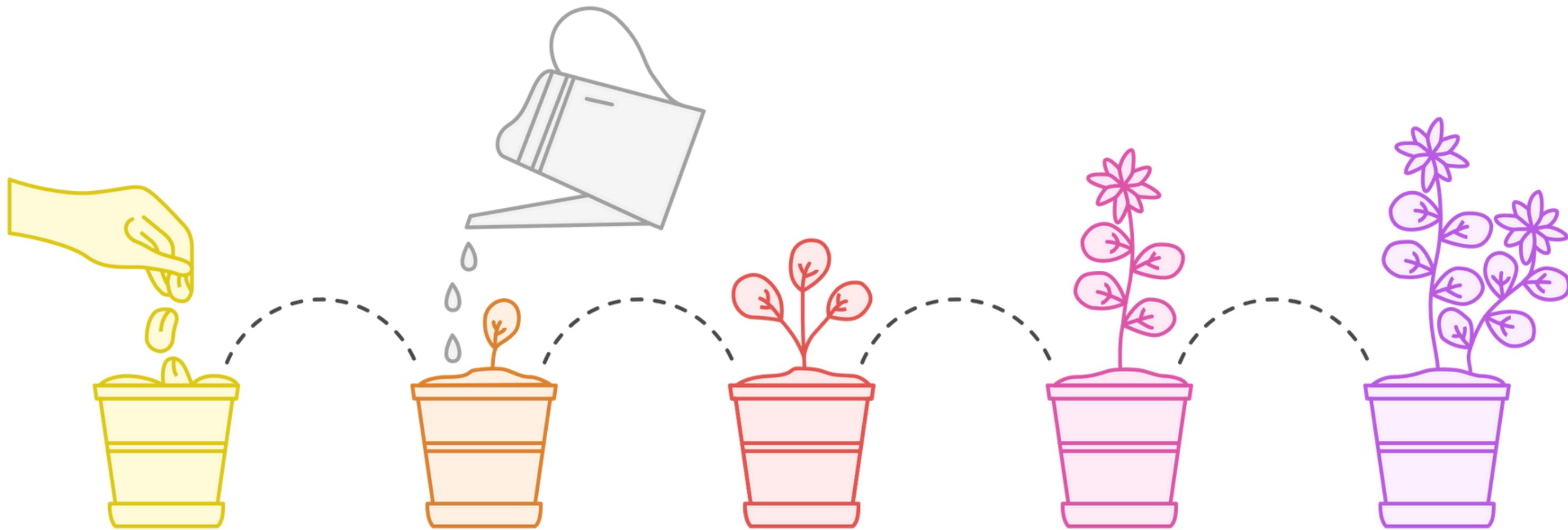


Professional Git User

Confident in Git workflows

THREE PRINCIPLES TO REMEMBER

Git Mastery Journey



Novice

Unfamiliar with Git

Commit Fearlessly

Git is a time
machine

Branch Liberally

Experiments are free

Automate Ruthlessly

Let tools handle
boring stuff

Expert

Confident Git user

WHAT'S NEXT?

Achieving Git Mastery

Git Fundamentals

Basic Git knowledge acquired

Daily Practice

Use Git for all projects

Embrace Mistakes

Learn from errors and conflicts

Read Errors

Understand Git's helpful messages

Git Mastery

Deep understanding and proficiency



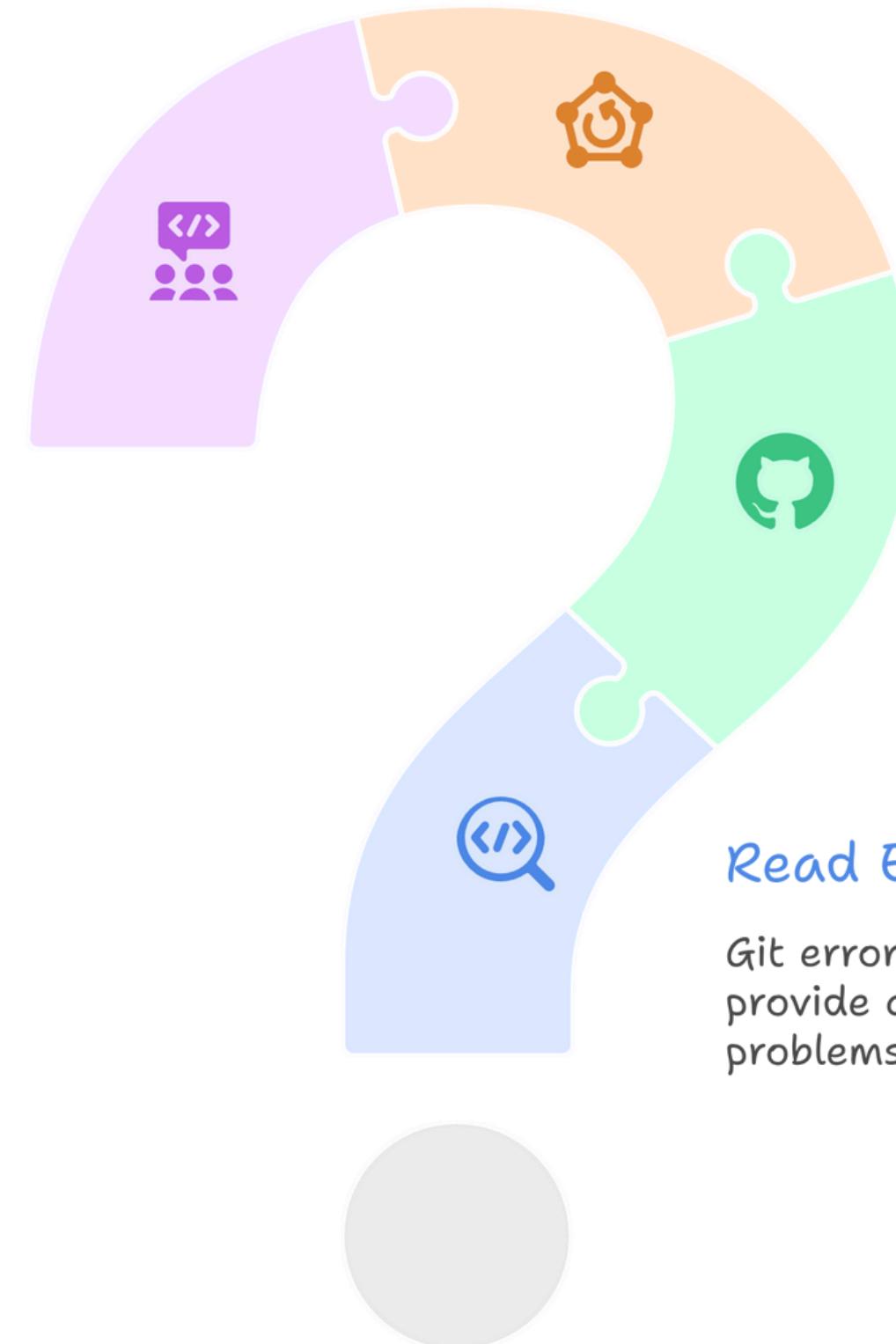
How to handle Git issues?

Create Backup Branch

Backing up your branch before risky actions prevents data loss.

Ask the Community

The Git community is supportive and can offer solutions.



Use 'git status'

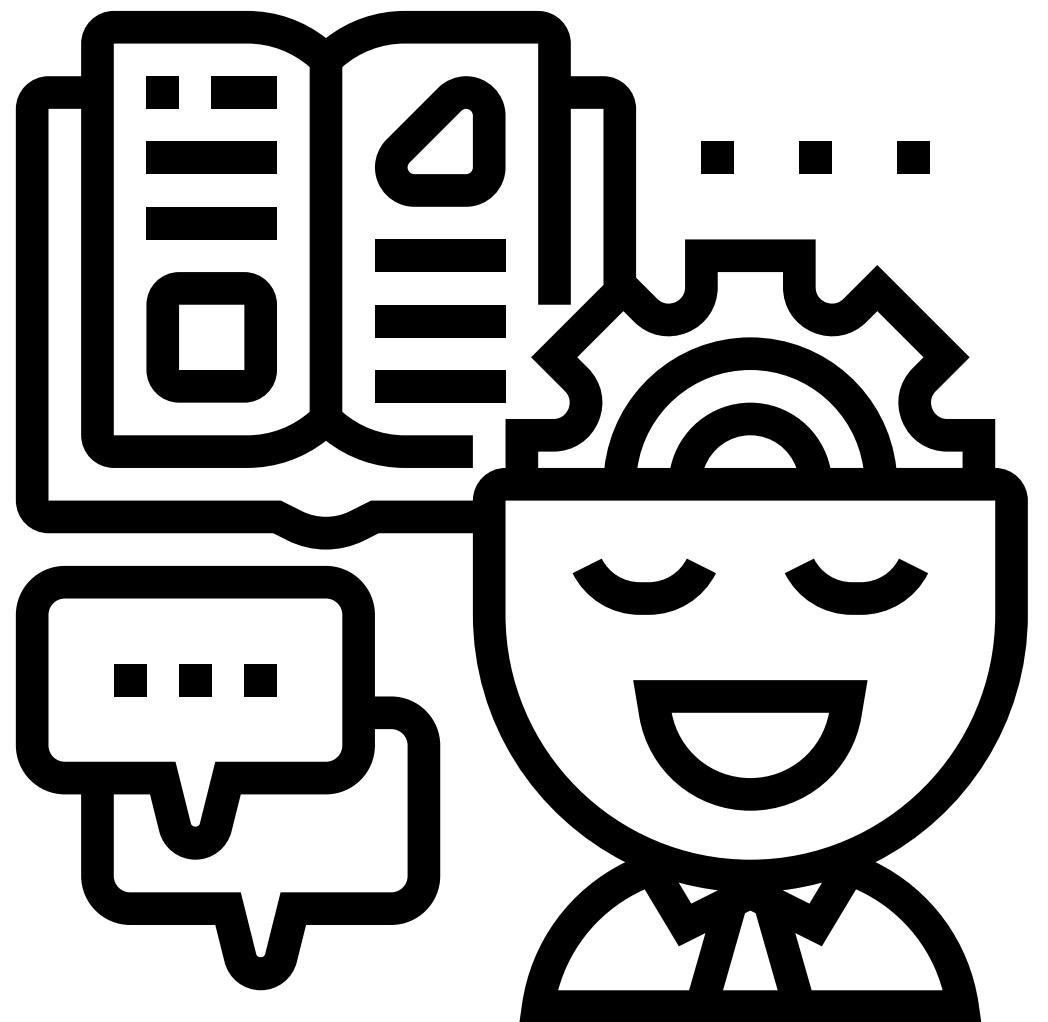
'git status' shows the current state of your repository, helping you track changes.

Read Error Messages

Git error messages often provide clues to solve problems.

The difference between a beginner and an expert isn't that the expert never messes up - it's that they know how to fix it when they do.





**THANK YOU
FOR YOUR
ATTENTION.**

**ANY
QUESTIONS?**