



Git & GitHub: Version Control for Beginners

Master the essentials of version control with Git and GitHub. This presentation will guide you through the fundamentals, helping you collaborate effectively and manage your code like a professional.



by Samuel Kapansa

What is Git & Why Use It?



Version Control System

Git tracks changes to your code over time. It creates snapshots of your entire project.



Parallel Development

Work on different features simultaneously without interference. Merge changes when ready.



Team Collaboration

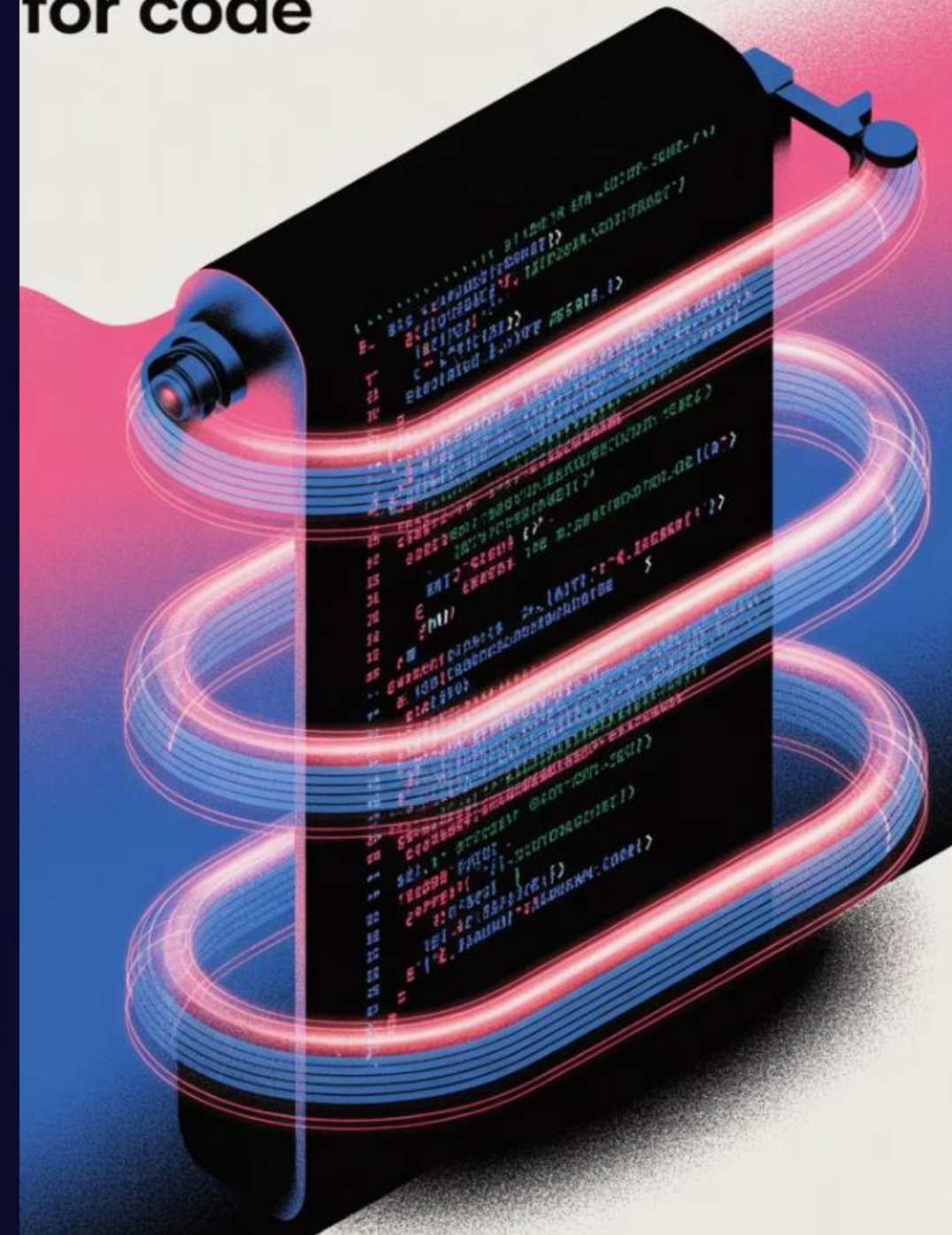
Multiple developers can work on the same project. Changes are coordinated efficiently.



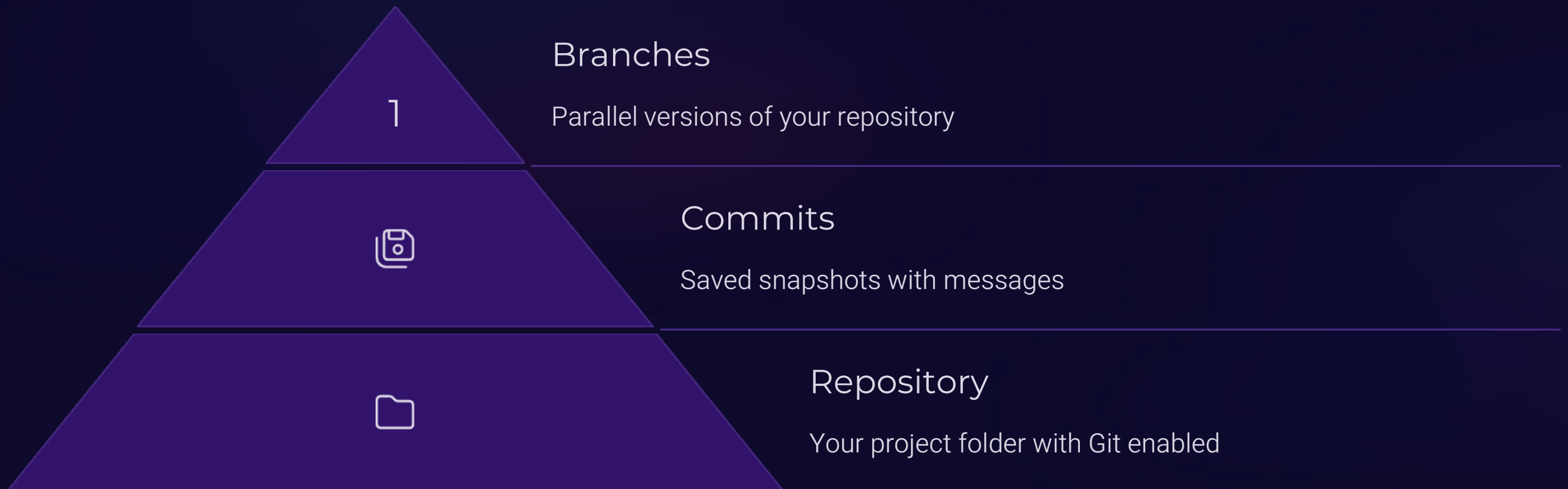
Disaster Recovery

Revert to previous versions when needed. Your history is always preserved.

Git as a time machine for code



Git Basics: Repositories, Commits, and Branches



Introduction to GitHub: Collaboration & Remote Repos

GitHub vs Git

Git is the tool. GitHub is the service hosting your repositories online.

GitHub adds collaboration features on top of Git's core functionality.

Key Benefits

- Project hosting
- Issue tracking
- Pull request reviews
- Team management
- Community contributions



Common Git Commands

git init

Creates a new Git repository. It turns your folder into a Git-managed project.

git commit

Saves your staged changes with a descriptive message. The -m flag adds a message.

git add

Stages changes for commit. Use `git add .` to stage all changes.

git push/pull

Push sends commits to GitHub. Pull fetches changes from GitHub.

Cloning & Forking Repositories



Clone

Copy a repository to your local machine.

Use: `git clone [url]`

2

Fork

Create your own copy on GitHub.

Use GitHub's fork button.



Modify

Make changes to your local or forked copy.



Contribute

Share changes via pull requests.



Pull Requests & Merging



Git Workflow in Teams



Feature Branch Workflow

Create a new branch for each feature.
Keep the main branch always deployable.



Gitflow Workflow

Strict branching model with develop, feature, release, and hotfix branches.



CI/CD Integration

Automate testing and deployment with each commit or merge.

Resolving Merge Conflicts



Identify Conflict

Git notifies you when a merge conflict occurs.



Locate Conflict Markers

Find the <<<<<, =====, and >>>>> in your code.

X₁

Edit Files

Manually resolve by editing or using merge tools.



Add and Commit

Mark as resolved with git add and commit.

Mini Task: Your First Contribution

Fork the Practice Repo

Visit github.com/learn-git/practice and click "Fork".

Clone Your Fork

Use `git clone` with your fork's URL to download it.

Add Your Name

Edit `contributors.md` to add your name to the list.

Commit and Push

Use `git add`, `commit`, and `push` to update your fork.

Create Pull Request

Go to your fork on GitHub and click "Pull Request".