

Git

An introduction to the world's worst
version control

Introduction to Git

- Distributed version control system designed to handle small and large projects with speed and efficiency
- Developed by Linus Torvalds in 2005
- Used for tracking changes in source code during software development.

Why Use Git?

- Version Control: keep track of changes made to the code
- Collaboration: share code without having to send countless .zip files
- Safety: revert to previous states of the code through committing
- Allows simultaneous work on multiple features through branches

Key Concepts in Git

- Repository: storage location for your project, including the entire history of all changes
- Branch: parallel version of the repository. The default branch is usually called “main”

Collaborative Concepts in Git

- Forking: creating a personal copy of someone else's project
- Pull Request: proposal of changes in a branch
- Issues: tracking of tasks and bugs
- Code Review: function of pull requests that allows discussion before merging

Best Practices in Git

- Commit Often
- Use Descriptive Commit Messages
- Limit The Amount Of New Content Per Branch
- Regularly Pull Changes
- Test Before Merging