git-primer

> Introductory level workshop focusing on Git fundamentals.

Rendered Slides: https://tinosibz.github.io/git-primer/

Purpose

- Git allows multiple people to work on the same project and track changes effectively, avoiding conflicts and preserving the history of changes. It also facilitates code review and collaboration, which can lead to better quality and maintainable code.
- **The sooner you make git a habit in your professional career, the better!**
- This workshop's aim is to review the fundamentals and to make sure you are all comfortable with the basics of Git which you will be expected to use throughout the internship.

Overview

1. Introduction to Git and Version Control
2. Setting up Git
3. Creating and Managing Repositories
4. Branching and Merging
5. Collaboration on GitHub
6. Git Best Practices
7. Advanced Git Features
8. Test Your Git Fu

Introduction to Git and Version Control

- Explanation of version control and why it's important for collaboration and tracking changes
- Overview of Git as a specific version control system
- Basic Git terminology (repository, commit, branch, etc.)
- A visual walk through

Setting up Git

- Installing Git on your computer
- Creating a Github account and connecting it to your local Git installation
- Basic configuration settings (username, email, etc.)

Creating and Managing Repositories

- Creating a new repository on Github
- Cloning a repository to your local machine
- Basic Git commands (add, commit, push, pull)
- Understanding the difference between the working directory, staging area, and local repository

Branching and Merging

- Explanation of branches and why they're useful
- Creating and switching between branches
- Merging branches and resolving conflicts

Collaboration on Github

- Understanding Github's pull request workflow
- Collaborating on a project with multiple people
- Managing permissions and access to a repository
- Understanding the concept of "remotes" and how they relate to working with remote repositories on Github

Git Best Practices

- Writing good commit messages
- Keeping a clean repository
- Using .gitignore
- Tips for debugging and troubleshooting
- Best practices for working with multiple people on a project

Advanced Git Features

- Tagging and release management
- Stashing changes
- Reverting changes
- Advanced merging and resolving conflicts

Test Your Git Fu

- Some self-paced exercises to practice your git knowledge.