



Missing Semester - Introduction to Git (Week 1)

Introduction to Git

Welcome to the Introduction to Git lesson! In today's session, we will dive into the world of version control and explore the powerful features of Git. Whether you are a software developer, a data scientist, or simply someone interested in managing your project's codebase efficiently, Git is an essential tool to master.

Git is a distributed version control system that allows multiple people to collaborate on a project simultaneously. It tracks changes made to files, enables easy branching and merging, and provides a reliable history of your project's development. With Git, you can easily revert to previous versions, experiment with new features without affecting the main codebase, and seamlessly collaborate with others.

Throughout this lesson, we will cover the fundamental concepts of Git, including creating a repository, making commits, branching, merging, and resolving conflicts. We will also explore some best practices and useful commands that will help you become proficient in using Git.

Whether you are a beginner or have some experience with Git, this lesson will provide you with a solid foundation to effectively manage your projects and collaborate with others. So, let's get started and unlock the power of Git!

Git is a **distributed version control system** that allows multiple people to collaborate on a project. It is widely used in software development **to track changes in code** and manage different versions of a project.

What is Version Control?

Version control is a system that records changes to a file or set of files over time. It allows you to track modifications, revert to previous versions, and collaborate with others on the same project. Without version control, it can be challenging to manage changes and keep track of different versions of a project.

Why Use Git?

Git offers several advantages over other version control systems. Here are some reasons why you should consider using Git:

1. **Distributed:** Git is a distributed version control system, which means that every user has a complete copy of the entire project history. This allows for offline work and easy collaboration between team members.
2. **Branching and Merging:** Git makes it easy to create branches to work on new features or bug fixes without affecting the main codebase. Branches can be merged back into the main branch once the changes are complete.
3. **History and Tracking:** Git keeps a detailed history of all changes made to a project, allowing you to track who made the changes and when. This makes it easier to identify and fix issues that may arise.
4. **Collaboration:** Git enables multiple people to work on the same project simultaneously. It provides tools for resolving conflicts that may occur when merging changes made by different team members.

Basic Git Workflow

To get started with Git, you need to follow a basic workflow. Here are the essential steps:

1. **Initialize a Repository:** To start using Git, you need to initialize a repository in your project directory. This can be done using the `git init` command.

2. **Add and Commit Changes:** After making changes to your project files, you need to add them to the staging area using the `git add` command. Once the changes are staged, you can commit them to the repository using the `git commit` command.
3. **Create and Switch Branches:** Git allows you to create branches to work on different features or bug fixes. You can create a new branch using the `git branch` command and switch to it using `git checkout`.
4. **Merge Branches:** Once you have completed the changes in a branch, you can merge it back into the main branch using the `git merge` command. Git will automatically handle any conflicts that may arise during the merge process.
5. **Push and Pull:** To collaborate with others, you can push your changes to a remote repository using the `git push` command. Similarly, you can pull changes made by others using the `git pull` command.

Conclusion

Git is a powerful version control system that allows for efficient collaboration and tracking of changes in a project. By understanding the basic concepts and workflow of Git, you can effectively manage your projects and work seamlessly with others.

Resources and Exercises for Further Studies.

1. [BOOK] Black book for learning Git ; Highly recommended - <https://git-scm.com/book/en/v2>
2. [WEB] Oh Shit, Git!?! ; A guide to recover from some common Git mistakes - <https://ohshitgit.com/>
3. [BLOG] Explain how Git works internally ; A must for the nerds - <https://eagain.net/articles/git-for-computer-scientists/>

4. [BLOG] How to explain git in simple words -
<https://smusamashah.github.io/blog/2017/10/14/explain-git-in-simple-words>