



Missing Semester - Basic Git Commands (Week 2)

Introduction to Basic Git Commands

Welcome to the lesson on Basic Git Commands! Git is a powerful version control system that allows developers to track changes in their code, collaborate with others, and easily revert back to previous versions if needed. In this lesson, we will explore some of the fundamental Git commands that every developer should know.

Whether you are a beginner or have some experience with Git, understanding these basic commands will lay a strong foundation for your journey in version control. We will cover essential commands such as `init`, `add`, `commit`, `status`, `log`, and more. By the end of this lesson, you will have a solid understanding of how to use these commands to manage your code effectively.

So, let's dive in and explore the world of Git commands together!# Basic Git Commands

Git is a distributed version control system that allows multiple people to collaborate on a project. It keeps track of changes made to files and allows you to easily revert back to previous versions if needed. Here are some basic Git commands that you will frequently use:

Initializing a Repository

To start using Git, you need to initialize a repository. This can be done by navigating to the project directory in your terminal and running the following command:

```
$ git init
```

This will create a new Git repository in the current directory.

Cloning a Repository

If you want to work on an existing Git repository, you can clone it to your local machine using the `git clone` command. This will create a copy of the repository on your computer. To clone a repository, use the following command:

```
$ git clone <repository-url>
```

Replace `<repository-url>` with the URL of the repository you want to clone.

Checking the Status

To see the current status of your repository, including any changes made to files, you can use the `git status` command. This will show you which files have been modified, added, or deleted.

```
$ git status
```

Adding Files

Before you can commit changes to your repository, you need to add the modified files to the staging area. This can be done using the `git add` command. For example, to add a single file, you can use:

```
$ git add <file-name>
```

To add all modified files, you can use:

```
$ git add .
```

Committing Changes

Once you have added the files to the staging area, you can commit the changes to the repository. A commit is a snapshot of the current state of the project. To commit changes, use the `git commit` command:

```
$ git commit -m "Commit message"
```

Replace `"Commit message"` with a descriptive message that explains the changes made in the commit.

Pushing Changes

If you are working on a remote repository, you can push your local commits to the remote repository using the `git push` command. This will update the remote repository with your changes. To push changes, use the following command:

```
$ git push
```

Pulling Changes

If there are new changes in the remote repository that you want to incorporate into your local repository, you can use the `git pull` command. This will fetch the latest changes from the remote repository and merge them with your local repository. To pull changes, use the following command:

```
$ git pull
```

These are some of the basic Git commands that you will frequently use. There are many more advanced commands and features available in Git, but understanding these basics will give you a good foundation to start using Git for version control.

Resources and Exercises for Further Studies.

1. [READ] Committing is hard(we know), but committing sensitive information is a big NO in Git. Read this blog for an important lesson - <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/removing-sensitive-data-from-a-repository>
2. [TASK] The repository for the project is live at - <https://github.com/yash-srivastava19/missing-semester-bdacc> . Explore the version history as a graph. Find out who was the last person to commit to this repository(plus the time and commit message.)
3. [TASK] The repository has a task associated with it which can be found - <https://github.com/yash-srivastava19/missing-semester-bdacc/tree/main/git> . Try to complete the task to earn a medal from Github, and a certificate from team BDACC(only successful PRs).