# Git

**Version Control System is a tools that helps to track changes in code**

**Git is a Version Control System. It is :**

- *popular*
- *free & Open Source*
- *fast & scalable*

Complete Git and GitHub Tutorial for Beginners

# Github

**Website that allows developers to store and manage their code using Git.**

*https://github.com*


5:39 / 1:15:21

Complete Git and GitHub Tutorial for Beginners

# Setting up Git

**Visual Studio Code**

**Windows (Git Bash)**

**Mac (Terminal)**

*git --version*

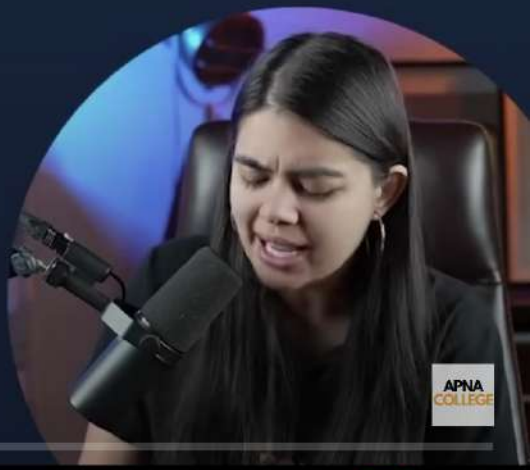

17:50 / 1:15:21

# Configuring Git


*git config --global user.name "My Name"*

*git config --global user.email "someone@email.com"*

*git config --list*



# Clone & Status

remote [ Github ]

local [ laptop / PC ]

**Clone - Cloning a repository on our local machine**

*git clone <- some link ->*

**status - displays the state of the code**

*git status*

Complete Git and GitHub Tutorial for Beginners

***untracked***

*new files that git doesn't yet track*

***modified***

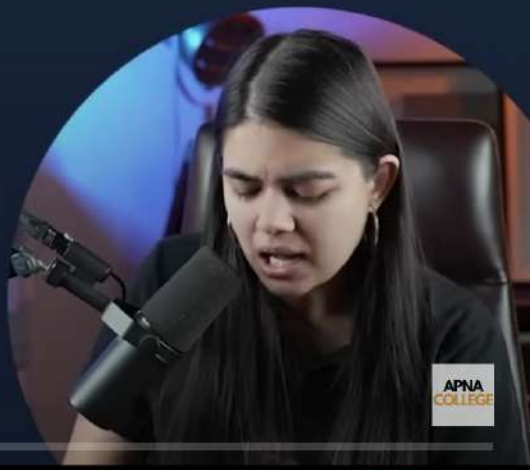
*changed*

***staged***

*file is ready to be committed*

***unmodified***

*unchanged*




31:20 / 1:15:21

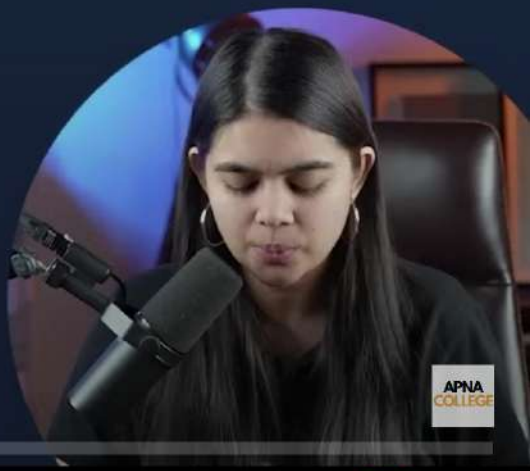
# Add & Commit

**add - adds new or changed files in your working directory to the Git staging area.**

*git add <- file name ->*

**commit - it is the record of change**

*git commit -m "some message"*



# Branch Commands

*git branch* (to check branch)

*git branch -M main* (to rename branch)

*git checkout* *<- branch name ->* (to navigate)

*git checkout -b <- new branch name ->* (to create new branch)

*git branch -d <- branch name ->* (to delete branch)

# Merging Code

## Way 1

*git diff <- branch name->* (to compare commits, branches, files & more)

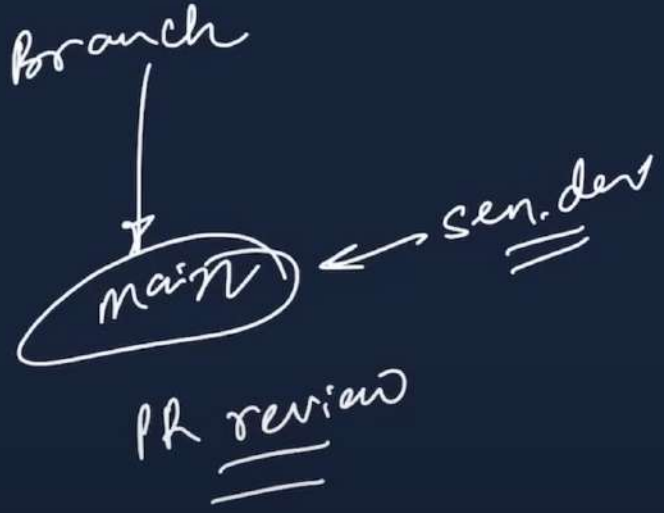
*git merge <- branch name->* (to merge 2 branches)

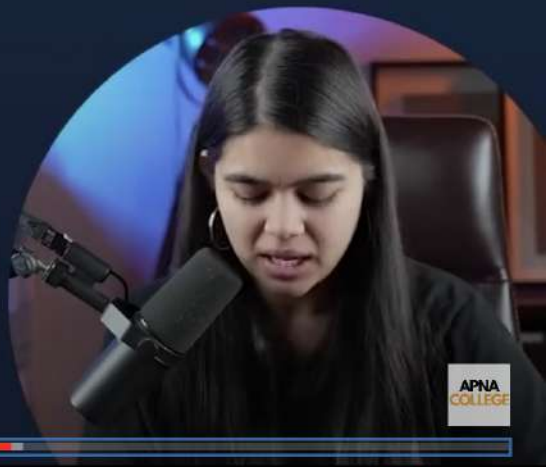
## Way 2

Create a PR

# Pull Request

It lets you tell others about changes you've pushed to a branch in a repository on GitHub.



Complete Git and GitHub Tutorial for Beginners

# Resolving Merge Conflicts

An event that takes place when Git is unable to automatically resolve differences in code between two commits.


1:00:52 / 1:15:21

# Undoing Changes

Case 1 : staged changes (add)

*git reset <- file name ->*

*git reset*

Case 2 : commited changes (for one commit)

*git reset HEAD~1*

Case 3 : commited changes (for many commits)

*git reset <- commit hash ->*

*git reset --hard <- commit hash ->*

Complete Git and GitHub Tutorial for Beginners

# Fork

A fork is a new repository that shares code and visibility settings with the original “upstream” repository.

Fork is a rough copy.


1:14:44 / 1:15:21