Git and Github Cheatsheet

created by: Muhammad Zohaib Abbas

GIT

Git is a distributed version control system that tracks changes in code, allowing multiple developers to collaborate efficiently. It helps manage different versions, making it easy to revert changes and merge code.

GITHUB

GitHub is a cloud-based platform that hosts Git repositories, enabling developers to share, manage, and collaborate on projects remotely. It also provides additional tools like pull requests, issue tracking, and project management.

Basic Git Commands
(for Version Control)

Basic Git Setup Commands

Set your Git username.

- git config --global user.name "Your Name"

Set your Git email address.

- git config -globaluser.email"your.email@example.com"

Verify the configuration settings.

- git config --list

Starting a Repository

Initialize a new Git repository in the current directory.

- git init

Clone a remote repository to your local machine.

- git clone <repository-url>

Basic Commands for Workflow

Check the status of your repository (modified files, staged files, etc.).

- git status

Stage a specific file for commit.

- git add <file>

Stage all changes.

- git add .

Commit staged changes with a descriptive message.

- git commit -m "Your commit message"

Push your local changes to the remote repository.

- git push

Fetch and merge changes from the remote repository to your local copy.

- git pull

Retrieve updates from the remote without merging them.

- git fetch

Branching

List all branches (shows current branch with an asterisk *).

- git branch

Create a new branch.

- git branch <branch-name>

Switch to an existing branch.

- git checkout <branch-name>

Create a new branch and switch to it.

- git checkout -b <branch-name>

Merge another branch into your current branch.

- git merge <branch-name>

Inspecting Changes

View commit history.

- git branch

See the difference between changes not yet staged.

- git branch <branch-name>

View changes between the staging area and the last commit.

- git checkout <branch-name>

Undoing Changes

Unstage a file (but keep changes in the working directory).

- git reset <file>

Discard local changes in a specific file

- git checkout -- <file>

Reset all changes since the last commit (dangerous, will lose changes).

- git reset --hard HEAD

Tags

Create a tag at the current commit.

- git tag <tagname>

Push a tag to the remote repository.

- git push origin <tagname>

Working with Remotes

Link your local repo to a remote repository.

- git remote add origin <remote-url>

View remote connections.

- git remote -v

Push your branch to the remote repository.

- git push origin <branch-name>

Pull changes from a remote branch.

- git pull origin <branch-name>

Stashing Changes

Save your local modifications temporarily to work on something else.

- git stash

Reapply the stashed changes.

- git stash apply

These are the key Git and GitHub commands I use regularly. They'll help you manage your projects easily. Feel free to save this for quick reference. Got any questions or tips? Let's chat!

Happy coding! 👨‍💻

Like ❤

Save 🔖

Repost 🔁

muhammadzohaib abbas

@bugsquashers1.0