# Git and GitHub Tutorial With Commands

Name:-Soumya Ranjan Upadhyaya

Redg. No. :- 2141021005

**1.What is Git?**

Git is a distributed version control system that helps developers track and manage changes in code. It allows multiple developers to work on the same project simultaneously without overwriting each other's work. You can revert to previous versions of your project and work offline as Git is local-first.

**2. What is GitHub?**

GitHub is a cloud-based platform for hosting Git repositories. It provides a web interface for Git, and adds features like issue tracking, pull requests, project boards, and GitHub Actions for automation

**3. What are Git Command Line?**

Git command line tools are command prompt use to interact with Git .They help to create repositories, tract changes ,collaborate with others and manage project history.

Some git commands are-

* mkdir
* cd
* git init
* touch
* git add
* git clone
* git status
* git log
* git commit

**mkdir git-assignment**

This command creates a new directory (folder) named **git-assignment**

**cd git-assignment**

This command moves you into the **git-assignment** directory, so you can start working inside it.

**cat > file.txt**

Creates a new file named **file.txt** and lets you type content into it.

* Pressing Ctrl + D will save and exit.

**echo hello >> file.txt**

Adds the text **hello** to a file named **file.txt**.

* >> means **append** (don’t overwrite if file exists).
* If file.txt doesn’t exist, it will be created.

**git init**

Initializes a new **Git repository** in the current folder (git-assignment).

* It creates a hidden .git folder that tracks version history.

**git add file.txt**

Adds the file **file.txt** to the **staging area**.

* This means Git is now tracking this file for the next commit.

**git commit -m "file created."**

* **Purpose**: Commits the staged changes (in this case, a newly created file filee.txt) to the repository with the message "file created.".

**git branch New-Branch**

* **Purpose**: Creates a new branch named New-Branch from the current branch (main).

**git checkout New-Branch**

* **Purpose**: Switches to the New-Branch branch.
* Now we’ll work in this branch instead of main.

**echo "New\_Branch" >> file.txt**

* **Purpose**: Appends the text "New\_Branch" to the file file.txt.
* This modifies the file content.

**git commit -m "file updated."**

* **Purpose**: To commit changes, but it **fails** because the changes were not staged.

**git config --global user.name "Soumya Ranjan Upadhyaya"**

* **Purpose**: Set the Git global configuration for the user's name.
* This name will be used in all commits made on the system.

**git checkout**

* You switch from your current branch (like main) to another branch(like New\_Branch)

**Adding File**

Add changes in a file to the staging area.

$ git add ‹file>

**Committing Changes**

Commit the staged changes to the local repository with a descriptive message.

$ git commit -m "message"

**Display Commit Changes**

Display the commit history.

$ git log

**Print Working Directory**

Pwd command stands for “Print working directory”.

Displays the full path of your current directory in the terminal code.

$ pwd

**Create A File**

This command is used to create file in git bash

$ touch

**Connect To GitHub**

Links the local repository to a remote one, usually on GitHub, for pushing and pulling code.

$ git remote add origin <https://github.com/username/repo-name>

# Create a new branch

Create a new branch from the current HEAD, Useful for feature development without affecting the main code.

$ git branch <new-branch>

# Checkout

Switches the working directory to the specified branch.

$ git checkout <new-branch>

# Merge Branches

Combines the specified branch into the current branch, integrating changes made in the other branch.

$ Git merge new-branch

# Delete Branches

To delete a branch locally

This will delete the branch only if it has already been fully merged .

$ git branch -d branch\_name

# Lists all the files

Lists all tracked files in the current repository

$ git ls-files