GitHub

❖ GITHUB:

- GitHub is a **code hosting / sharing platform** for **version control** and **collaboration.**
- Anyone can sign up and host a **public code repository** for free, which makes GitHub especially popular with **open-source projects**.

GITHUB IS A WEB-BASED PLATFORM THAT ALLOWS DEVELOPERS TO:

- Store, share, and collaborate: Developers can store their code in a "repository" on GitHub, where they can share their work, track changes, and collaborate with others.

MANAGE PROJECTS:

- GitHub provides tools for managing changes from multiple developers, so that collaborators can work together without interfering with each other's work.

USE GIT:

- GitHub uses the open-source version control software Git, which allows multiple people to make changes to web pages at the same time.

TRACK CHANGES:

- GitHub makes it easy to track changes and navigate revisions.

SHOWCASE WORK:

- Developers can showcase their work on GitHub.

GET FEEDBACK:

- Others can review code and make suggestions to improve it.

BY USING GITHUB, WE WILL:

- Create and use a repository
- Start and manage a new branch
- Make changes to a file and push them to GitHub as commits
- Open and merge a pull request

GITHUB REPOSITORY VISIBILITY:

- You can restrict who has access to a repository by choosing a repository's visibility: public or private.

  ### PUBLIC:

  Public repositories are accessible to everyone on the internet.

  ### PRIVATE:

  Private repositories are only accessible to you, people you explicitly share access with, and, for organization repositories, certain organization members.

➤ **CREATING A NEW REPOSITORY FROM THE WEB UI:**

STEP 1: First sign up / sign in GitHub Account:

STEP 2: Creating a new repository from the web UI:

Repository name: aws-project

Choose a repository visibility: Public

Click Create repository.

STEP 3: Create or upload some files in the repository.

test1 test2 test3

➤ **CLONING A REPOSITORY:**

- Cloning a repository is grabbing all the source files and copying them to your computer
- When you create a repository on GitHub, it exists as a remote repository. You can clone your repository to create a local copy on your computer and sync between the two locations.

STEP 1: On GitHub, navigate to the main page of the repository

Above the list of files, click **<>Code.**

Copy the URL for the repository (HTTPS):

STEP 2: Open Git Bash:

Change the current working directory to the location where you want the cloned directory.

STEP 3: Type git clone, and then paste the URL you copied earlier:

$git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY

To clone along with branch name:

$git clone -b main https://github.com/sysgeeks4u/git-learn.git

$ls

➢ MANAGING REMOTE REPOSITORIES:

- To add a new remote, use the **git remote add** command on the terminal, in the directory your repository is stored at.
- The git remote add command takes two arguments:
  - A remote name, for example, origin
  - A remote URL, for example, https://github.com/OWNER/REPOSITORY.git

To add a Remote repository:

$ git remote add origin https://github.com/OWNER/REPOSITORY.git

$ git remote -v

CHANGING A REMOTE REPOSITORY'S URL:

- The git remote set-url command changes an existing remote repository URL.

  List remote repositories:

  $git remote -v

  Change your remote's URL:

  $git remote set-url origin https://github.com/OWNER/REPOSITORY.git

  $git remote -v

RENAMING A REMOTE REPOSITORY:

- Renaming a remote repository. Use the git remote rename command to rename an existing remote.

  $git remote rename origin destination

  $git remote -v

REMOVING A REMOTE REPOSITORY:

  $git remote -v

  $git remote rm destination

NOTE: git remote rm does not delete the remote repository from the server. It simply removes the remote and its references from your local repository.

GIT PUSH:

- It is used to upload local repository content to a remote repository.
- After git pushes the changes to the remote repository other developers can access the changes and they can contribute their changes by git pulling.
- Before pushing it to the remote repository you need to do a git commit to your local machine.

  SYNTAX: $git push <remote> <branch>

  $git push origin master (or) $git push -u origin main

  NOTE: First time you need to add GitHub credentials / token on browser

- The **--set-upstream or -u** flag sets the upstream for the branch, which means that future git pull commands executed on that branch will automatically know which remote branch to fetch and merge changes from, and future git push commands will know where to push the changes without specifying the remote and branch names.

  $git push --set-upstream origin master

  $git push

GIT PULL:

- It is used to fetch and download content from a remote repository and immediately update the local repository to match that content.
- Git pull is basically combination of git merge and git fetch which is used to update the local branch with the changes available in the remote repository branch.

  SYNTAX: $git pull [remote-name] [branch-name]

  To pull from the default remote and current branch:

  $git pull

  Pulling From a Specific Remote and Branch:

  $git pull <name of the branch>

  $git pull master

GIT FETCH:

- **git fetch** really only downloads new data from a remote repository, but it doesn't integrate any of this new data into your working files.
- Fetch is great for getting a **fresh view** on all the things that happened in a remote repository.

  First create a file in remote repository then run the command in git bash:

  $git fetch

  $git status

  $ls

  List logs into git:

  $git log

  List logs on remote repository:

  $git log origin/master

GIT PULL Vs GIT FETCH:

- The key difference between git fetch and pull is that git pull copies changes from a remote repository directly into your working directory, while git fetch does not.
- The git fetch command only copies changes into your local Git repo.

Git Pull	Git Fetch
Git pull is the combination of two commands git fetch and git merge.	Git fetch command will fetch the remote repository to the local machine.
It will fetch the changes and merges with the repository.	It will only fetch the branch without any merge operation.
A more efficient method for updating your local branch is to use git pull.	Git fetch is frequently used as a first step before using 'git merge' to combine updates.