Git and GitHub

1 What is Git and Github

Git is a a distributed version control system used to track changes in files, particularly in software development. Git is created by Linux Foundation while GitHub is created by GitHub company, which is now owned by Microsoft.

GitHub is the online platform of Git and it allows people to work in group or individually.

Git is installed and located in the local machine. One thing that sets Git and GitHub is the branching model. GitHub is the cloud-based service of Git.

GitHub is suitable for people who are working in projects without conflicting each other.

While GitHub is mostly used for Software Development, it is also used to do other things too such as keeping documents, research,etc. Especially, the projects that require controlling the version.

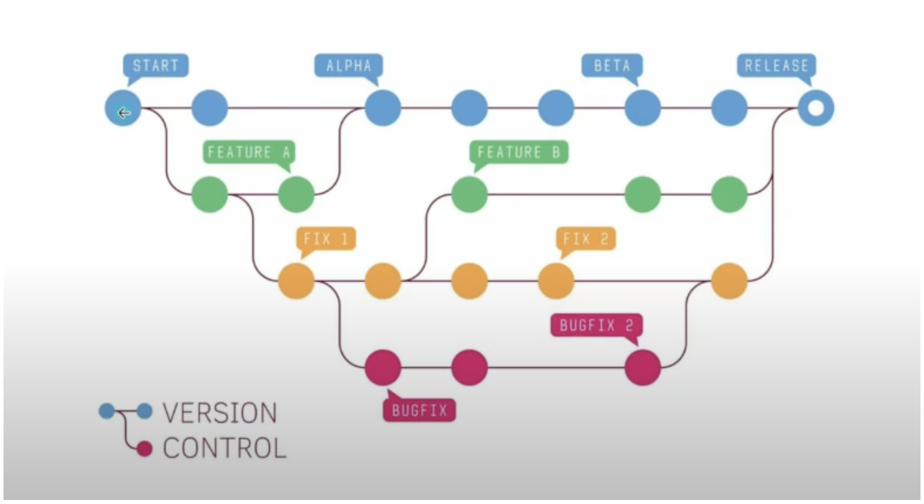
GitHub is really suitable for project management. It helps tracking the work of the team members.

Sometimes, people also GitHub to write books and write research papers

A terminal is an interface that user can interact with the operating system

You can perform many tasks (including the OS GUI) using the terminal. Some tasks may not be performed by the GUI so we need to use the terminal to perform that task and get results.

2 Version control



GitHub is not the only online platform for Git. There are others like Gitlab, Bitbucket, etc. But GitHub is the most popular among all.

We can use GitHub Desktop instead of Git for simplicity.

Git is the best for version control, while using GitHub is the best for using Git on the cloud. GitHub Desktop is the most popular one when we do not want to use the terminal for Git.

3 Introduction to Git

Git is a free and open source version control system for your code. It means that Git manages changes to a project without overwriting any part of that project.

Git is a collection of notes on the most recently worked on project files as well as nice, chronological history of what has changed.

More fomally, Git is a distributed version control system (VCS). Where you can think of as a shared digital diary for computer code, allowing many people to write in it while keeping a record of all the changes that were made over time.

Git lets multiple users track the changes of the project, help maintaining the history of code changes and ensuring traceability. Examples, you can think of the Julia Programming Language, which is an open-source coding language.

A fun fact: Git is created for the development of Linux kernel by Linus Torvalds. He created both Git and Linux, which is really popular in the world right now.

4 Version Control in Git

Git allows people to have a complete copy of the project on their own computer, making it easy to track changes, collaborate, and merge their work.

They start by setting up a Git storage space(repository) for their project. Each person download the repo to their own machine. Git allows people to work on their own and people can revert back to the old version if they want.

Git have different branches for different parts of the project that people are working on.

5 GitHub account and repository

GitHub is a website that helps developer store and manage code using Git. Every project can be called as repository as you can store and manage the project in the repository.

6 Installation of Git

Heads to the page https://git-scm.com/downloads. Choose OS system that is compatible with your computer, and then download the latest version of git.

You can also GitBash but it is not needed as you can use your terminal or use the terminal of VSCode as it is much easier

After downloading you can open the terminal(command prompt) to check the version of the git that you have downloaded Git succesfully. After that you can use Git for your project. You can also launch the Git command through the terminal in your VS Code and manage your project.

After downloading Git, you have configure your Git so that you can when you commit sth it can identify the user who made the commit and the email to associate the commit with your profile. You use Git command to config your profile.

```
git config –global user.name ”Your username”
git config –global user.email ”Your email”
```

7 Steps in Cloning and Doing GitHub project

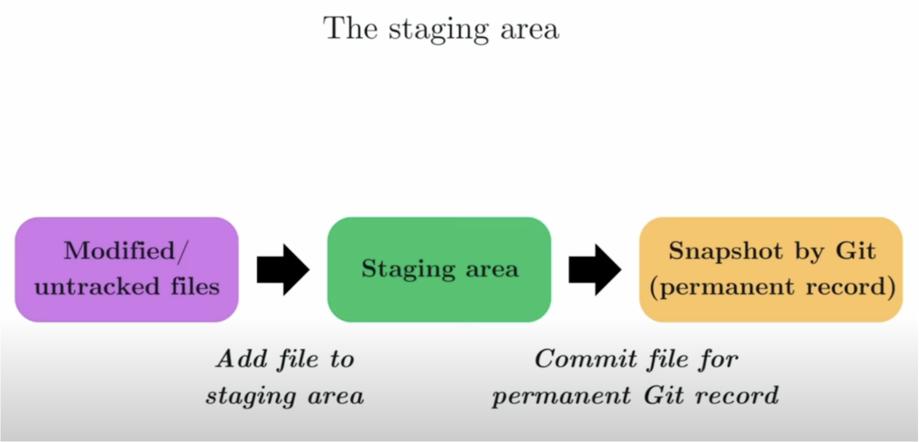
1. Make GitDemo folder
2. Check if git is properly installed
3. Clone the GitHub repository using git clone ”link repo”
4. Check the status of the repo by using git status

5. Add and commit
6. Push command to GitHub and end the session

With git status that we see which branch we are in, and if there is any changes that we have not commited and what we need to commit.

There are 4 stages of Git status:

- Untracked: Git does not know about the file, since it's not commited. New file
- Modified: We made change in the earlier commited file.
- Unmodified: No change in the file.
- Staged: File ready to be commited.


The staging area
Modified/ untracked files
Staging area
Snapshot by Git (permanent record)
Add file to staging area
Commit file for permanent Git record

We will add the file to the staging area and then after that we will commit the code (which is snap snot by Git), which is now ready to be pushed to GitHub.

git add: Add file to the stagin area
git commit: Commit the changes

8 Convert your project from local to GitHub

Using git push. Before that we need to add and commit all the code we have changes. We need to link the repo first using git remote add origin ”link” and then check by using git remote -v.

9 Git branches

When you want to experiment a feature without affecting the main code, you can create a branch to do the experiment without making any changes to the main code. Due to branches, 2 developers can continue work independently without the need to wait for each other. After that you can merge the branches that you have created. Steps in doing the branches:

1. Check the branch where you are in. **git branch**
2. Rename the branch name(not really needed) **git branch -M newname**
3. Create and navigate to the new branch **git check out -b newbranchname**
4. Check the branch where you are in **git branch**. If you are not in the branch you want you can use **git checkout** to change to the branch you want
5. Delete the unnecessary branch (if exist) **git branch -d branchname**. Remember that you can only delete the branch if you are not in that branch
6. After creating the code in another branch, you can push it to github
7. You can check the changes in the branch you have created in Github

10 Merging branches in Git

There are 2 ways to merge the code: Using terminal and Using GitHub

10.1 Using the temrinal

git diff branchname: Check the differences between 2 branches
git merge branchname: Merge the changes between 2 branches.

10.2 Using GitHub

Using the pull request in GitHub. Pull request let you tell others about the changes that you have pushed to a branch in a repository in GitHub.
If your code is accepted to be changed, it means that the branch you have created can be merged into the main branch.
In the industry, the senior developer will be the person to decide if your code can be merged into the main branch or not.
To create a pull request, you need to write a description for your code and why you change it. When you create a pull request, you need to write a detailed change of the code you have made so that your code can be merged into the main branch. You should never write just a line of description for your code.

11 Bring the changes from the remote the local using git pull

After the changes have been merged into the main branch, the important things now is to keep working on the code and to do this you need to pull the code into your machine.
Command to do this:

git pull origin main: Fetch and download the content from the remote repo and immediately update local repo to match the content.

12 Forking a repository

Git fork: Create a new repository that shares code and visibility settings with original ”upstream” repo
You can fork a repo and then make experiment changes without changing the original code.