GIT & GITHUB

**GIT** is a distributed version control system used for tracking changes in source code during software development.

It allows multiple developers to collaborate on projects simultaneously, managing different versions of files, merging changes, and tracking the history of edits.

Git is particularly popular due to its speed, efficiency, and flexibility.

**GITHUB** is a web-based platform that provides hosting for Git repositories.

It offers additional features on top of Git, such as issue tracking, project management tools, and collaboration features like pull requests and code reviews.

GitHub serves as a central hub for developers to store, share, and collaborate on code.

**Key Concepts of Git & GitHub:**

1. **Repository (Repo)**: A repository is a collection of files and folders associated with a specific project, tracked by Git. GitHub hosts these repositories online, allowing users to access them from anywhere.
2. **Clone**: To clone a repository means to create a local copy of it on your own machine. This allows you to work on the project locally, make changes, and synchronize with the remote repository on GitHub.
3. **Commit**: A commit is a snapshot of changes made to files in the repository. Each commit has a unique identifier and includes a message describing the changes made.
4. **Branch**: A branch is a parallel version of the repository that allows you to work on features or fixes without affecting the main codebase (usually called the "master" or "main" branch). Branches are used for experimentation and isolation of changes.
5. **Merge**: Merging is the process of combining changes from one branch into another. This typically occurs when you've completed work on a feature branch and want to incorporate those changes into the main branch.
6. **Pull Request (PR)**: A pull request is a feature in GitHub that allows developers to propose changes to a repository. It's often used for peer review, discussion, and eventual merging of code changes into the main branch.
7. **Fork**: Forking a repository means creating a copy of someone else's repository under your own GitHub account. You can make changes to your forked repository independently and propose them back to the original repository via pull requests.
8. **Merge Conflict**: A merge conflict occurs when Git is unable to automatically resolve differences between two branches being merged. This typically happens when the same part of a file has been modified in different ways in the branches being merged. Developers must manually resolve these conflicts before completing the merge.

**GIT Commands:**

1. **git init**: Initialize a new Git repository in the current directory.
2. **git clone**: Create a local copy of a remote repository.
3. **git add**: Add changes to the staging area in preparation for committing.
4. **git commit**: Record changes to the repository.
5. **git status**: Show the status of changes in the working directory.
6. **git branch**: List, create, or delete branches.
7. **git checkout**: Switch branches or restore working tree files.
8. **git merge**: Merge changes from one branch into another.
9. **git pull**: Fetch from and integrate with another repository or a local branch.
10. **git push**: Update remote branches with local commits.
11. **git log**: Show the commit history.
12. **git remote**: Manage connections to remote repositories.
13. **git fetch**: Download objects and refs from another repository.
14. **git remote**: Manage connections to remote repositories.
15. **git fetch**: Download objects and refs from another repository.

**EXAMPLE:**

1. **Create a Repository on GitHub**:
   * Go to GitHub and log in to your account.
   * Click on the "+" icon in the top right corner and select "New repository".
   * Enter a name for your repository, add a description if needed, choose whether it will be public or private, and then click "Create repository".
2. **Clone the Repository**:
   * Copy the URL of the repository you just created.
   * Open a terminal or command prompt on your local machine.
   * Use the **git clone** command followed by the repository URL to clone it to your local machine.

**git clone <repository\_url>**

1. **Make Changes Locally**:
   * Navigate to the cloned repository on your local machine.
   * Make changes to the files in the repository using your preferred text editor or IDE.
2. **Stage and Commit Changes**:
   * Use the **git add** command to stage the changes you want to commit.

**git add .**

Use the **git commit** command to commit the staged changes with a descriptive message.

**git commit -m “First commit”**

5. **Push Changes to GitHub**:

* Use the **git push** command to upload your committed changes to the GitHub repository.

**git push origin master**

**Create a Pull Request**:

* + If you're collaborating with others or working on a feature branch, you might want to create a pull request to propose your changes.
  + Go to your repository on GitHub.
  + Click on the "Pull requests" tab.
  + Click on the "New pull request" button.
  + Select the branch with your changes as the "compare" branch and the main branch as the "base" branch.
  + Add a title and description for your pull request, then click "Create pull request".

6. **Review and Merge**:

* + Other collaborators can review your pull request, add comments, and suggest changes.
  + Once the changes are approved, a repository maintainer can merge the pull request into the main branch.

7.**Fetch and Pull Changes**:

* + Periodically, you may want to fetch and pull changes from the GitHub repository to keep your local copy up to date.