# Learn GIT Basics

**Git Notes  - TheTestingAcademy (Pramod Sir)**

**Source** - https://sdet.live

Install Git into the System

1. Linux - Simply open up a new terminal and install Git via your distribution's package manager. For Ubuntu, the command is: **sudo apt-get install git**
2. Windows - we recommend git for windows as it offers both a GUI client and a BASH command line emulator.
3. OS X - The easiest way is to install Homebrew, and then just run **brew install git** from your terminal.

What is GIT?

● Git is a version control system that is used for tracking changes to files.
● It does this through a series of **snapshots of your project.** It works with those snapshots to version and manage your source code, and it does this in a simple way.

Why Use Git?

● Can work offline.
● Collaborating with others is easy!
● Branching is easy!
● Branching is fast!
● Merging is easy!
● Git is fast.
● Git is flexible.

## GIT vs Github vs Gitlab

- Git is a version control system, while GitHub and GitLab are web-based Git repositories.
- GitHub is a company that provides Git hosting, and it offers both a cloud-based hosting service and on-premises enterprise versions.
- GitLab is similar to GitHub, but it is an open-source Git hosting platform. It provides a web-based interface for working with Git repositories, as well as a range of tools for collaboration, project management, and continuous integration.

Git is a version control system, GitHub is a Git hosting service, and GitLab is an open-source Git hosting platform.

## Git Architecture



Tier Architecture Workflow
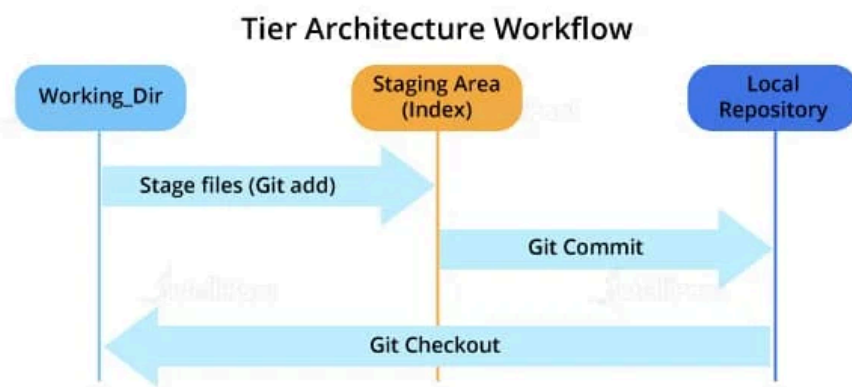
With Remote Repo
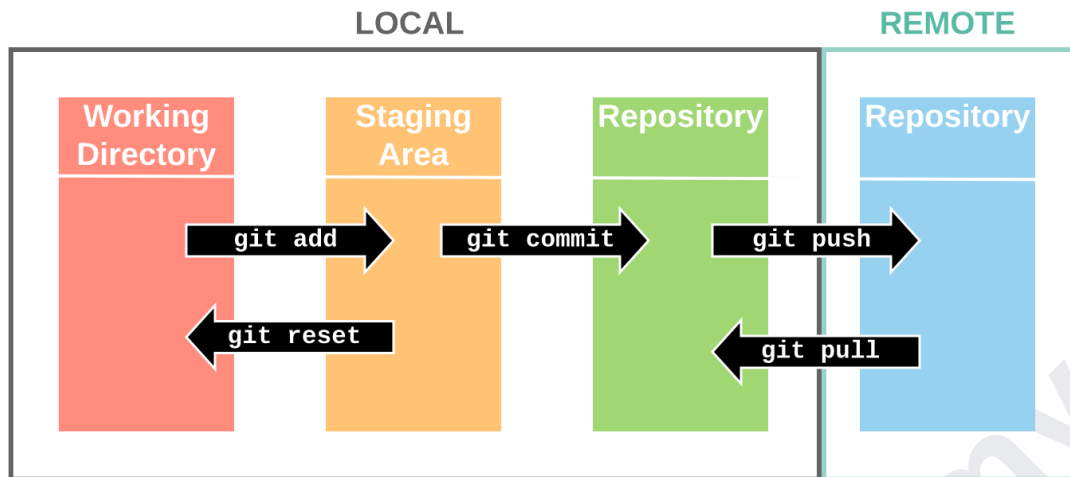
## Configuring Git

```
$ git config --global user.name "My Name"
$ git config --global user.email myEmail@example.com
git config --global --list
```

## .git Directory

The .git directory contains all the configurations, logs, branches, HEAD, and more

## Commands

### Creating a new repository

```
git init
```

### Checking the status

```
git status
```

### Staging

- Git has the concept of a "staging area".
- You can think of this like a blank canvas, which holds the changes which you would like to commit.
- It starts out empty, but you can add files to it (or even single lines and parts of files) with the git add command, and finally commit everything (create a snapshot) with git commit

```
git add
git add hello.txt
git add . // Add everything
```

```
git status
```

Commiting

A commit represents the state of our repository at a given point in time.
It's like a snapshot, which we can go back to and see how thing were when we took it.

```
git commit -m "Initial commit."
```

Blame

Examine specific parts of the code's history and find out who was the last author to modify that line "`bash.
Show what revision and author last modified each line of a file
https://git-scm.com/docs/git-blame

Remote repositories

```
git remote add
```

**Connecting to a remote repository**
1. In order to upload something to a remote repo.
2. Create a Repo at Github.com and Add Remote to it.
3. The Git command to do this is git push and takes two parameters.
   the name of the remote repo (we called ours origin) and the branch to push to

```
git remote add origin https://github.com/PramodDutta/Restfulbooker.git
```

**Add File and Push**

```
git add .
git commit -m "blah blah"
git remote add origin https://github.com/PramodDutta/Restfulbooker.git
git push origin main
```

**Cloning a repository**
● Download locally and have a fully working copy of your project

```
git clone https://github.com/PramodDutta/Restfulbooker.git
```

**Getting changes from a server**
If you make updates to your repository, people can download your changes with a single command

```
git pull origin master
```

## Branches

- Copy of the original project, called a branch.
- Developers can work on their own branch, without the risk of their codebase changing due to someone else's work.
- An already working, stable version of the code won't be broken.
- 
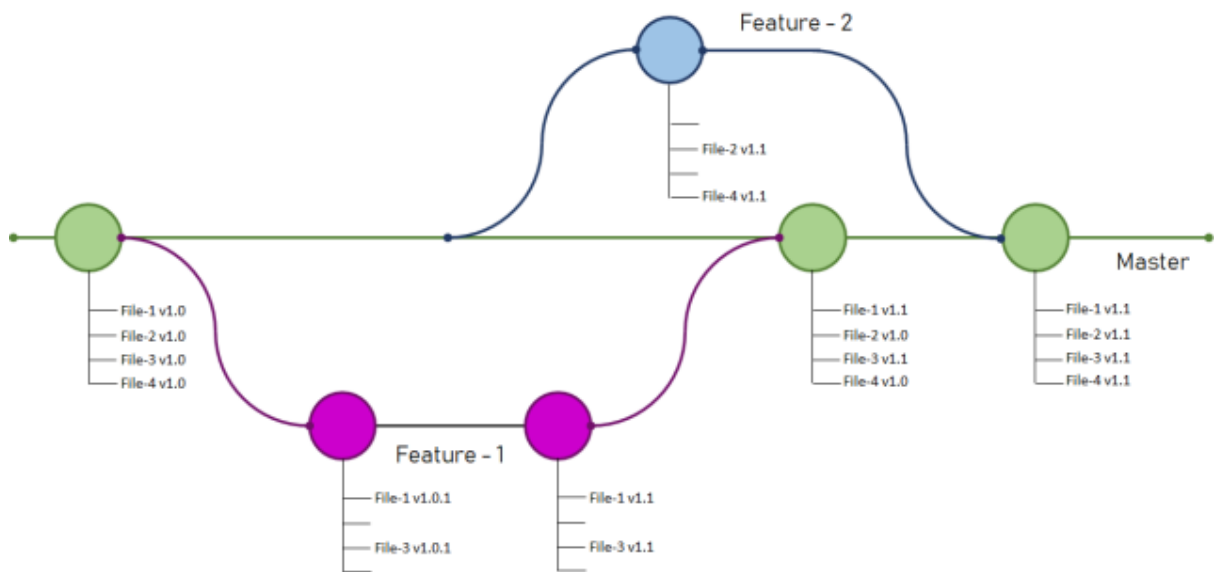


### Creating new branches

```
git branch new_feature
```

### Switching branches

```
git checkout new_feature
```

### Merging branches

```
git add feature.txt
git commit -m "New feature added"
git checkout main
git merge
// Remove old branch
```

```
git branch -d amazing_new_feature
```

## Merge and Rebase

## TEMPORARY COMMITS

```
git stash
//Save modified and staged changes
git stash list
//list stack-order of stashed file changes
git stash pop
//write working from top of stash stack
git stash drop
//discard the changes from top of stash stack
```

## Little Advanced

1. Setting up .gitignore
   a. log files
   b. task runner builds
   c. the node_modules folder in node.js projects
   d. folders created by IDEs like Netbeans and IntelliJ
   e. personal developer notes

2. Check the diff between commits, use the

```
git log
```

3. Want to see in that commit, diff?

```
git show <commit Id>
```

4. Want to see diff from 1 to 2 ids

```
git diff 09bd8cc..ba25c0ff
```

5. Reverting a file to a previous version.

```
git checkout 09bd8cc1 hello.txt
```

6. Fixing a commit and Revert to Head
   a. newest commit can be accessed by the HEAD alias.

```
git revert HEAD
```

```
git revert b10cc123 // Or
```

7. Resolve conflict
   a. git merge tim_branch

```
<<<<<<< HEAD
// Use a for loop to console.log contents.
for(var i=0; i<arr.length; i++) {
    console.log(arr[i]);
}
=======
// Use forEach to console.log contents.
arr.forEach(function(item) {
    console.log(item);
});
>>>>>>> Tim's commit.

$ git add -A
$ git commit -m "Array printing conflict resolved."
```

Git Bisect

- Git bisect is a command in the Git version control system that helps in identifying the commit that introduced a bug or change in the code
- It works by performing a binary search through the commit history to find the commit that caused the change.
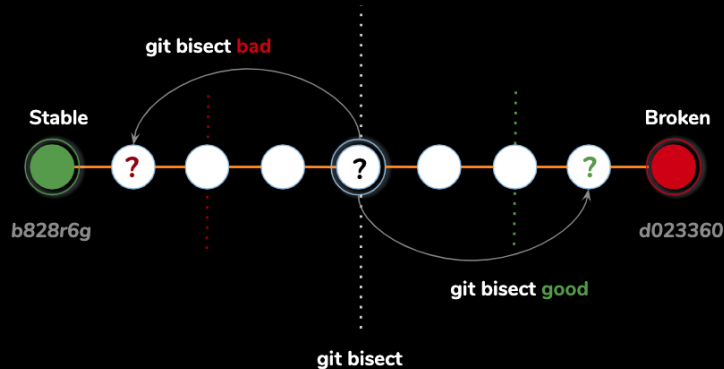
Here is an example of how you can use Git bisect in the command line:

Start the bisect process:

$ git bisect start
#Mark the current state of the code as "bad":

$ git bisect bad
# Mark the known good state of the code:

$ git bisect good <commit-hash>
Git will then checkout a commit in the middle of the good and bad commits. You can test the code to see if it is working as expected. If it is, you can mark it as "good":

$ git bisect good
If the code is not working, you can mark it as "bad":

$ git bisect bad
Git will then checkout another commit and you can repeat steps 4 and 5 until Git bisect has found the specific commit that caused the issue.

Finally, end the bisect process:

$ git bisect reset
This is a simple example of how you can use Git bisect to find the specific commit that caused a change or bug in your code. Bisect is a powerful tool that can save you a lot of time and effort in debugging and fixing issues in your code.

## Reference and Download PDF

- https://education.github.com/git-cheat-sheet-education.pdf
- https://tutorialzine.com/2016/06/learn-git-in-30-minutes
- https://www.geeksforgeeks.org/git-difference-between-merging-and-rebasing/
- https://learnxinyminutes.com/docs/git/

## 50+ Git Interview QnA:

**What is GIT?**
GIT is a version control and source code management (SCM) system that is designed to handle both small and large projects quickly and efficiently.

## What is GIT version control?

- GIT version control enables you to monitor the evolution of a group of files (code files).
- It allows for the creation of several versions of a file collection. Each version saves a snapshot of the files at a specific point in time, and the snapshot can be used to revert the collection of files. (The code can be developed in different versions of Java, and can be merged in Git)
- VCS permits switching between these versions. These versions are often maintained in a location known as a repository. (You can move between Java versions during the development process.)

## What is a Distributed Control System?

Later, the code is sent from our local workstation to the central repository (GitHub). Working does not require a connection to a centralized repository.

## How to configure GitHub repository locally?
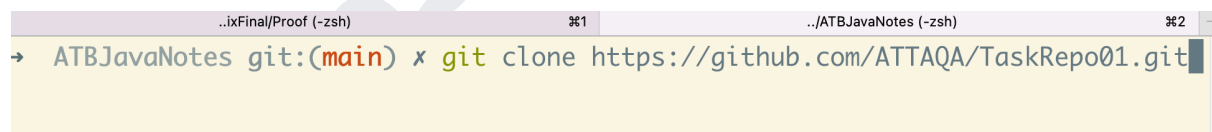
\# git config --global alias.lo "log --oneline" -----> To create an Alias to Command
\# git config --global --unset alias.lo -----> To Remove an Alias
\# git config --global --unset user.name -----> to remove username

## What is the git clone?

To download an existing repository from Centralized (Github) to local system. # git clone



## What is 'git add'?

To add files from work area to Index/staging/cache area. # git add

## What is the use of 'git log'?

To see the commits by content or history, author

- git log -----> To show the Git Commits
- git log -5 -----> To show Recent 5 Commits
- git log --oneline -----> To Display the each commit in one line
- git log --since=2023-01-21
- git log --until=2023-01-01
- git log --author="user_name"

- git log --grep="Index"
- git log --oneline --author="user_name"

```
commit 91680ccea2e4723303377cb64e491f516ea7c948 (HEAD -> m
Author: ipromode <pdutta@tekion.com>
Date:   Sun Oct 30 15:13:59 2022 +0530

    added collections

commit eeb3c63359863b5c135ba83c0881bd6b0f6cd6bd
Author: ipromode <pdutta@tekion.com>
Date:   Wed Oct 26 18:16:13 2022 +0530

    added collections

commit f7f0c47ea1d6edbaae30708a772b58717fe5ca39 (oops, col
Merge: 2111143 2e1ed95
Author: ipromode <pdutta@tekion.com>
Date:   Fri Aug 26 11:03:19 2022 +0530

    Merge branch 'main' of https://github.com/PramodDutta/

commit 2111143a7e96eb6a537a38a19976cc692bd66445
Author: ipromode <pdutta@tekion.com>
```

## How to edit an incorrect commit message in Git? Or How can you fix a broken commit?
 git commit --amend -m "This is your new Git Message"

## How do you undo the last commit?
 git revert <your commit id>

## What is git reset?
Reset the current HEAD state to specific state.

## How to delete a Remote Branch?
git push origin -d

## How to see the difference between 2 branches

git diff <name of branch 1>..<name of branch 2>

## Why GIT better than Subversion (SVN)?
Git is an open-source version control system that allows you to perform a project's'version' command.
Each modification can be linked to a specific developer. Multiple developers can check out and post changes.

**What is git stash?**
Stashing takes the Temporary stored state of your working directory.

```
git stash list [<log-options>]
git stash show [-u | --include-untracked | --only-untracked] [<diff-
options>] [<stash>]
git stash drop [-q | --quiet] [<stash>]
git stash pop [--index] [-q | --quiet] [<stash>]
git stash apply [--index] [-q | --quiet] [<stash>]
git stash branch <branchname> [<stash>]
git stash [push [-p | --patch] [-S | --staged] [-k | --[no-]keep-index] [-
q | --quiet]
          [-u | --include-untracked] [-a | --all] [(-m | --message)
<message>]
          [--pathspec-from-file=<file> [--pathspec-file-nul]]
          [--] [<pathspec>…]]
git stash save [-p | --patch] [-S | --staged] [-k | --[no-]keep-index] [-q
| --quiet]
          [-u | --include-untracked] [-a | --all] [<message>]
git stash clear
git stash create [<message>]
git stash store [(-m | --message) <message>] [-q | --quiet] <commit>
```

**What is Git and how does it work?**
Git is a version control system that tracks changes to files and directories. It allows multiple people to work on the same project simultaneously and keeps track of all changes made to the files.

**What is a Git repository?**
A Git repository is a collection of files and directories that are tracked by Git. It is a place where all the changes to a project are stored and can be accessed by other users.

**What is a commit in Git?**
A commit in Git is a snapshot of the current state of a project. It is a way of saving changes made to the files and directories in a Git repository.

**How do I create a new Git repository?**
To create a new Git repository, you can use the command "git init" in the terminal or command prompt. This will initialize a new repository in the current directory.

**How do I clone a Git repository?**
To clone a Git repository, you can use the command "git clone [repository URL]" in the terminal or command prompt. This will create a copy of the repository on your local machine.

**What is the difference between git pull and git fetch?**
git pull is used to fetch changes from a remote repository and merge them into the current branch. git fetch is used to fetch changes from a remote repository, but it does not merge them into the current branch.

**How do I add a file to a Git repository?**

To add a file to a Git repository, you can use the command "git add [file name]" in the terminal or command prompt. This will add the file to the repository and prepare it for a commit.

## How do I commit changes in Git?

To commit changes in Git, you can use the command "git commit -m [commit message]" in the terminal or command prompt. This will save the changes made to the files and directories in the repository.

## How do I push changes to a remote repository?

To push changes to a remote repository, you can use the command "git push [remote name] [branch name]" in the terminal or command prompt. This will send the changes made to the repository to the remote repository.

## How do I create a new branch in Git?

To create a new branch in Git, you can use the command "git branch [branch name]" in the terminal or command prompt. This will create a new branch in the repository that can be used to make changes without affecting the master branch.

## How do I switch between branches in Git?

To switch between branches in Git, you can use the command "git checkout [branch name]" in the terminal or command prompt. This will switch to the specified branch and make it the active branch.

## How do I merge branches in Git?

To merge branches in Git, you can use the command "git merge [branch name]" in the terminal or command prompt. This will combine the changes made to the specified branch with the current branch.

## How do I resolve conflicts in Git?

To resolve conflicts in Git, you can use the command "git mergetool" in the terminal or command prompt. This will open a tool that allows you to compare and resolve conflicts between different versions of a file.

## How do I view the history of a Git repository?

To view the history of a Git repository, you can use the command "git log" in the terminal or command prompt. This will show a list of all the commits made to the repository.

## How do I revert a commit in Git?

To revert a commit in Git, you can use the command "git revert [commit hash]" in the terminal or command prompt. This will undo the changes made in the specified commit and create a new commit to reflect the changes.

## How do I delete a branch in Git?

To delete a branch in Git, you can use the command "git branch -d [branch name]" in the terminal or command prompt. This will delete the specified branch from the repository.

**How do I stash changes in Git?**
To stash changes in Git, you can use the command "git stash" in the terminal or command prompt. This will temporarily save changes made to the files and directories without committing them.

**How do I apply stashed changes in Git?**
To apply stashed changes in Git, you can use the command "git stash apply" in the terminal or command prompt. This will apply the changes that were previously stashed.

**How do I create a tag in Git?**
To create a tag in Git, you can use the command "git tag [tag name]" in the terminal or command prompt. This will create a new tag that can be used to mark a specific commit in the repository.

**How do I push a tag to a remote repository?**
To push a tag to a remote repository, you can use the command "git push [remote name] [tag name]" in the terminal or command prompt. This will send the specified tag to the remote repository.

**How do I view the status of a Git repository?**
To view the status of a Git repository, you can use the command "git status" in the terminal or command prompt. This will show the current state of the repository, including changes that have been made but not yet committed.

**How do I create a remote repository in Git?**
To create a remote repository in Git, you can use a service like GitHub or GitLab. This will create a new repository on the remote server that can be accessed by other users.

**How do I add a remote repository in Git?**
To add a remote repository in Git, you can use the command "git remote add [remote name] [repository URL]" in the terminal or command prompt. This will add the specified repository as a remote that can be accessed and pushed to.

**How do I remove a remote repository in Git?**
To remove a remote repository in Git, you can use the command "git remote remove [remote name]" in the terminal or command prompt. This will remove the specified repository as a remote that can be accessed and pushed to.

**How do I configure Git?**
To configure Git, you can use the command "git config" in the terminal or command prompt. This allows you to set various settings for your Git installation, such as your name and email.

**How do I check out a specific commit in Git?**
To check out a specific commit in Git, you can use the command "git checkout [commit hash]" in the terminal or command prompt. This will change the current branch to reflect the state of the repository at the time of the specified commit.

**How do I create a patch file in Git?**

To create a patch file in Git, you can use the command "git format-patch [commit hash]" in the terminal or command prompt. This will create a file that contains the changes made in the specified commit.

**How do I resolve a merge conflict in Git?**

To resolve a merge conflict in Git, you can use a text editor to manually edit the conflicting files and resolve any issues. Once the conflicts have been resolved, you can use the command "git add" to stage the changes, and "git commit" to finalize the merge.

**How do I view the commit history in Git?**

To view the commit history in Git, you can use the command "git log" in the terminal or command prompt. This will display a list of all commits made to the repository, along with the commit message, author, and date.

**How do I view the difference between two commits in Git?**

To view the difference between two commits in Git, you can use the command "git diff [commit hash 1] [commit hash 2]" in the terminal or command prompt. This will display the changes made between the two specified commits.

**How do I cherry-pick a commit in Git?**

To cherry-pick a commit in Git, you can use the command "git cherry-pick [commit hash]" in the terminal or command prompt. This will apply the changes made in the specified commit to the current branch.

**How do I rebase a branch in Git?**

To rebase a branch in Git, you can use the command "git rebase [branch name]" in the terminal or command prompt. This will reapply the commits made on the current branch onto the specified branch.

**How do I merge a branch in Git?**

To merge a branch in Git, you can use the command "git merge [branch name]" in the terminal or command prompt. This will combine the changes made on the specified branch with the current branch.

**How do I squash commits in Git?**

To squash commits in Git, you can use the command "git rebase -i [commit hash]" in the terminal or command prompt. This will open an editor where you can specify which commits to squash together.

**How do I rename a branch in Git?**

To rename a branch in Git, you can use the command "git branch -m [old branch name] [new branch name]" in the terminal or command prompt. This will change the name of the specified branch.

**How do I view the branches in a Git repository?**

To view the branches in a Git repository, you can use the command "git branch" in the terminal or command prompt. This will display a list of all branches in the repository.

**How do I view the remote branches in a Git repository?**
To view the remote branches in a Git repository, you can use the command "git branch -r" in the terminal or command prompt. This will display a list of all remote branches in the repository.

**How do I view the current branch in Git?**
To view the current branch in Git, you can use the command "git branch" in the terminal or command prompt. The current branch will be indicated with an asterisk (*) next to its name.

**How do I clone a repository in Git?**
To clone a repository in Git, you can use the command "git clone [repository URL]" in the terminal or command prompt. This will create a copy of the specified repository on your local machine.

# GIT Notes(with Interview QnA) - for Automation Testers / SDET

Important GIT Command Useds which can be used day to day life for managing code on remote repositories.

## ✅ To initialize a repository or create new git repository in your local system -

Command Used: git init

## ✅ What are the new changes and files updated in your code?

Command Used: git status

## ✅ How to add files to the staging area?

Command Used: git add * / git add <filename>

## ✅ How to commit code to local git repo? (local repository not yet pushed to remote)

Command Used: git commit -m "<message>"


## ✅ How to push code to a remote repository?

Ans:
i. Authenticate yourself by using token or basic authentication
ii. Create a repository on your github account
iii. Copy the http url, then use following

Command Used: git remote add origin <remote_url>


## ✅ Pushing the code to remote repository

Command Used: git push origin <branch_name>


## ✅ How to review the current branch?

Command Used: git branch (it reflects all existing branches and current working branch)


## ✅ How to create and update a working branch?

Command Used: git branch -M <branch_name>


## ✅ How to update remote repository URL?

Command Used: git remote set-url origin <url>


## ✅ How to get status and updates made in remote branches?

Command Used: git diff origin/<branch_name>


## ✅ How to update the local repository?

Command Used: git pull origin <branch_name>

## ✅ How to delete a branch on local (after commit)?

Command Used: git branch -d <branch_name>

## ✅ How to force delete a branch?

Command Used: git branch -D <branch_name>

## ✅ How to remove files from the staging area?

Command Used: git reset

## ✅ How to create a git-ignore file?

Command Used: touch .gitignore (works in git-bash)
Mac Terminal: git add .gitignore

## ✅ How to get a new repository in your local system?

Command Used: git clone <remote_repo_url>

## ✅ How to search for text in the current working repo/directory?

Command Used: git grep <text>

## ✅ How to create and move branch working directory (local)?

Command Used: git checkout -b <branch_name>

## ✅ How to fetch the latest changes from remote to local?

Command Used: git pull origin master (branch_name)

## ✅ How to save local changes made without committing or moving them to remote repository?

Command Used: git stash

Note: Used when you want to save local changes temporarily that are not yet committed. Its primary use is when you want to switch a branch or work on different task.

## ✅ How to use stashed changes in git local repo so as to commit to remote repository?

Command Used: git stash apply

Note: Use the changes which were stashed earlier. Please check before applying the changes you switch to the repository for which changes were made.

## ✅ GIT Workflow:

# Git Workflow

{Untracked}    {Index}    {Head}    {Remote Repo}

git add    git commit -m "message"    git push

git commit -a -m "message"

**Working Directory**    **Staging Area**    **Local Repo**    **Remote Repo**

git merge    git fetch

git pull

git diff HEAD

git diff --staged [<commit>]

git diff --cached [<commit>]

git diff <commit or branch>

git diff