**GIT And GITHUB**

✅What is Git?

* Git is a distributed version control system used for tracking changes in source code during software development.
* It allows multiple developers to work on the same codebase simultaneously and keep track of changes made by each developer.

In simple words:

Think of saving a status of a game{checkpoint} and returning back to the checkpoint if you mess up in the future!

✅History of Git

* Git was created in 2005 by Linus Torvalds, the creator of the Linux operating system.
* He created Git as an alternative to the existing version control systems available at the time, which he found to be inefficient and difficult to use.
* Today, Git is widely used by developers and has become the industry standard for version control

✅Who uses Git?

* Git is used by developers and development teams of all sizes, from small startups to large corporations.
* It is used in a variety of industries, including software development, web development, and game development.
* If you're interested in a career in software development, learning Git is a must.

✅Git vs GitHub

* Git is a version control system, while GitHub is a web-based platform that provides hosting for Git repositories.
* While Git allows developers to track changes and collaborate on code, GitHub adds additional features such as issue tracking, pull requests, and collaboration tools.

<https://drive.google.com/file/d/1oSubBQTFECRph_PmV52oeTkzRVyMnfHu/view?usp=share_link>

✅Git Repository:

* A Git repository is a collection of files and folders that are managed by Git.
* It keeps track of all the changes made to the codebase, including when changes were made, who made them, and why they were made.

✅Git Init:

* Git Init is a command used to initialize a new Git repository in your current working directory.
* It creates a .git folder that contains all the necessary files to track changes in your codebase.

✅Git Status:

* Git Status is a command that shows the current state of your working directory and staging area.
* It shows which files have been modified and which files are staged for commit.

✅.git folder:

* The .git folder is the heart of a Git repository.
* It contains all the information needed to track changes to the codebase, including the history of all commits, branches, and tags.

✅Committing Workflow:

* The Git committing workflow involves three stages - staging, committing, and pushing.
* You stage changes with the add command, commit changes with Git Commit, and push changes to a remote repository with Git Push.

✅Staging Changes with Add Command:

* The add command is used to stage changes in your working directory for the next commit.
* You can stage changes one file at a time or all changes at once.

✅Git Commit:

* Git Commit is a command used to save changes to the Git repository.
* It creates a new commit with a unique ID that contains a snapshot of the changes made since the last commit.

✅Git Log:

* Git Log is a command used to view the history of commits in a Git repository.
* It shows the commit ID, author, date, and commit message.

**1. git init:**

This command is used to initialize a new Git repository in the current directory. For

example, you can use the following command to initialize a new Git repository in the myproject directory:

git init my-project

**2. git clone:** This command is used to clone an existing Git repository from a remote source. For

example, you can use the following command to clone the my-project repository from the

https://github.com/user/my-project.git URL:

$ git clone https://github.com/user/my-project.git

**3. git add:**   
  
This command is used to add files to the staging area in a Git repository. For

example, you can use the following command to add the file1.txt and file2.txt files to the

staging area:

$ git add file1.txt file2.txt

4. git commit:   
  
This command is used to commit the changes in the staging area to the Git

repository. For example, you can use the following command to commit the changes in the

staging area with the commit message "Added new files":

1 $ git commit -m "Added new files"

5. git push: This command is used to push the committed changes in the local repository to the

remote repository. For example, you can use the following command to push the committed

Important GIT Commands -

By Naveen AutomationLabs

11/12/2022, 17:03 Box Notes

https://naveenautomationlabs.app.box.com/notes/1087874050730 2/7

changes to the origin remote branch:

1 $ git push origin

6. git pull: This command is used to pull the changes from the remote repository to the local

repository. For example, you can use the following command to pull the changes from the

origin remote branch:

1 $ git pull origin

7. git branch: This command is used to list, create, or delete branches in a Git repository. For

example, you can use the following command to create a new branch called my-branch:

1 $ git branch my-branch

8. git checkout: This command is used to switch between branches in a Git repository. For

example, you can use the following command to switch to the my-branch branch:

1 $ git checkout my-branch

9. git log: This command is used to display the commit history of a Git repository. For example,

you can use the following command to view the commit history with the author and date

information:

1 $ git log --pretty=format:"%h %an %ad"

11/12/2022, 17:03 Box Notes

https://naveenautomationlabs.app.box.com/notes/1087874050730 3/7

10. git diff: This command is used to view the differences between two versions of a file in a Git

repository. For example, you can use the following command to view the differences

between the file1.txt file in the HEAD and the my-branch branch:

1 $ git diff HEAD my-branch file1.txt

11. git stash: This command is used to save the local changes in a Git repository without

committing them. For example, you can use the following command to stash the local

changes:

1 $ git stash

12. git tag: This command is used to add tags to specific commits in a Git repository. For

example, you can use the following command to add a tag called v1.0 to the latest commit:

1 $ git tag v1.0

13. git merge: This command is used to merge one branch into another branch in a Git

repository. For example, you can use the following command to merge the my-branch branch

into the master branch:

1 $ git merge my-branch

14. git reset: This command is used to reset the state of a Git repository to a previous commit.

For example, you can use the following command to reset the repository to the HEAD

commit:

1 $ git reset --hard HEAD

11/12/2022, 17:03 Box Notes

https://naveenautomationlabs.app.box.com/notes/1087874050730 4/7

15. git config: This command is used to configure settings for a Git repository. For example, you

can use the following command to set the user name and email address for the repository:

1

2

16. git remote: This command is used to manage the remote repositories for a Git repository. For

example, you can use the following command to add a new remote repository called origin:

1 $ git remote add origin https://github.com/user/my-project.g

17. git fetch: This command is used to download the objects and references from a remote

repository to the local repository. For example, you can use the following command to fetch

the objects and references from the origin remote repository:

1 $ git fetch origin

18. git gc: This command is used to clean up unnecessary files and optimize the Git repository.

For example, you can use the following command to run the garbage collector:

1 $ git gc

19. git blame: This command is used to view the commit history for each line of a file in a Git

repository. For example, you can use the following command to view the commit history for

the file1.txt file:

1 $ git blame file1.txt

$ git config --global user.name "Naveen Automation"

$ git config --global user.email "naveenautomation@example.com"

11/12/2022, 17:03 Box Notes

https://naveenautomationlabs.app.box.com/notes/1087874050730 5/7

20. git rev-parse: This command is used to parse revision names and extract information from

the revision names. For example, you can use the following command to get the abbreviated

commit hash for the HEAD commit:

1 $ git rev-parse --short HEAD

21. git show: This command is used to show the details of a specific commit in a Git repository.

For example, you can use the following command to show the details of the HEAD commit:

1 $ git show HEAD

22. git clean: This command is used to remove untracked files from a Git repository. For

example, you can use the following command to remove all untracked files from the

repository:

1 $ git clean -df

23. git grep: This command is used to search for a specific pattern in the files in a Git repository.

For example, you can use the following command to search for the foo pattern in the file1.txt

file:

1 $ git grep foo file1.txt

24. git submodule: This command is used to manage submodules in a Git repository. For

example, you can use the following command to add a submodule called my-module from

the URL:

11/12/2022, 17:03 Box Notes

https://naveenautomationlabs.app.box.com/notes/1087874050730 6/7

25. git bisect: This command is used to perform a binary search through the commit history of a

Git repository to find a specific commit. For example, you can use the following command to

start a bisect search for the buggy commit:

$ git bisect start

$ git bisect bad

$ git bisect good buggy

1

2

3

26. git fsck: This command is used to verify the integrity of the objects in a Git repository. For

example, you can use the following command to verify all the objects in the repository:

1 $ git fsck --full

27. git cherry-pick: This command is used to apply the changes from a specific commit to the

current branch in a Git repository. For example, you can use the following command to apply

the changes from the f46f5e5 commit to the current branch:

1 $ git cherry-pick f46f5e5

28. git rebase: This command is used to reapply the commits from a branch on top of another

branch in a Git repository. For example, you can use the following command to rebase the

my-branch branch onto the master branch:

1 $ git rebase master my-branch

29. git mv: This command is used to move or rename a file in a Git repository. For example, you

can use the following command to rename the file1.txt file to file2.txt:

$ git submodule add https://github.com/user/my-module.git my-mo

11/12/2022, 17:03 Box Notes

https://naveenautomationlabs.app.box.com/notes/1087874050730 7/7

1 $ git mv file1.txt file2.txt

30. git ls-files: This command is used to list the files in the index and the working tree of a Git

repository. For example, you can use the following command to list all the files in the

repository:

1 $ git ls-files