

Git

1. Git is open source distributed version control system (DVCS)
2. It is mainly used to collaborate with others.
3. And to track the code of the developers.
4. And to provide security and for more efficiency.

Features :

1. Scalable - If more than one will use it will be efficient.
2. Security - It uses SHA1 (Secure Hash Function) > Algorithm used for cryptography
3. Speed - It is quick compared to others.
4. Open source - Under General Public License (GPL)

Benefits :

1. To Save Time - It is fast.
2. For offline working - It supports
3. To Undo mistakes - we can undo
4. And to track changes - to track status

3 Stages in Git :

1. Working Directory > it is the current working directory after adding in the staging area.
2. Staging Area > It is a preview of commit
3. Repository > After commit it will added in the repository

Command :

1. **\$ ls** → To list.
2. **\$ mkdir <name>** → To create a new directory.
3. **\$ cd <name>** → Change directory.
4. **\$ git init** → Initialize a git repository.
5. **\$ git status** → To check status.
6. **\$ git add <file-name>** → Add file to staging area.
7. **\$ git commit -m "Commit message here"** → Commit file to git repository.
8. **\$ git log** → To check history.



Great job! You are ready to move on to the next lecture.

You got 6 out of 6 correct.

✓ **What you know** ⓘ

Which command do we use to initialize an empty git repository?

Which command do we use to add a file to the staging area?

Which command do we use to commit a file to our git repository (.git directory)?

Which command do we use to check the status of our working directory and staging area?

What are the three states of git?

Which command do we use to check the history of our commits (snapshots)?

- 9. **\$ git add *.html** → To add .html all files or the same type file.
- 10. **\$ touch hello.css** → To create or add new files.
- 11. **\$ ls -a** → list all files.
- 12. **\$ git add -A** → Add all files & folders.
- 13. **\$ git reset HEAD <file name>** → To remove from staging area.



Great job! You are ready to move on to the next lecture.

You got 4 out of 4 correct.

✓ **What you know** ⓘ

Which command do we use to add all files and folders (including hidden ones) to the staging area?

Which command do we use to add all files of the same type?(e.g., all .html files)

Which command do we use to remove helloWorld.js from the staging area?

What is the name of the file that we create in order to list all of the files and folders we want git to ign...

- 14. **\$ touch .gitignore** → Remove from staging area.
- 15. **\$ mv <file name 1> <filename>** → to rename or change files.
- 16. **\$ mv <filename> <foldername>** → To move file to folder.
- 17. **\$ git rm --cached <File name>** → To remove file in folder.
- 18. **\$ cd <folder name> && <filename>** → To create file in folder.
- 19. **\$ echo "folder name" >> .gitignore** → Add the folder to .gitignore.

- 20. **\$ git branch** → To list all branches.
- 21. **\$ git checkout -b <branch_name>** → To add a new branch.
- 22. **\$ git checkout <branch-name>** → To change branch.
- 23. **\$ git merge <branch-name>** → To merge branches from old to new.
- 24. **\$ git branch -d <branch-name>** --> To delete or remove branch

Press Esc to exit full screen

Great job! You are ready to move on to the next lecture.

You got 5 out of 5 correct.

✓ What you know ⓘ

Which command do we use to list our branches?

Which command do we use to add and checkout a new branch, simultaneously?

Which command do we use to switch to another branch?

What are the steps for merging a feature branch into master?

Which command do we use to remove an unwanted branch?

>> Login to github

And use SSH Key from C9 to github add SSH

>> Push an existing repository from the command line

\$ git remote add origin git@github.com:nax3t/intro-to-git.git

\$ git push -u origin master

>> In c9 use these command

\$ git remote add origin <url>([git@github.com](https://github.com/nax3t/intro-to-git.git):nax3t/intro-to-git.git)

25. **\$ git remote add origin <url>** → To add into github.

26. **\$ git remote -v** → To check origin that it is added in github.

27. **\$ git push -u origin <branch>** → To push into github.

 **You've finished the last lesson in this course!**

Find more courses

2. Video

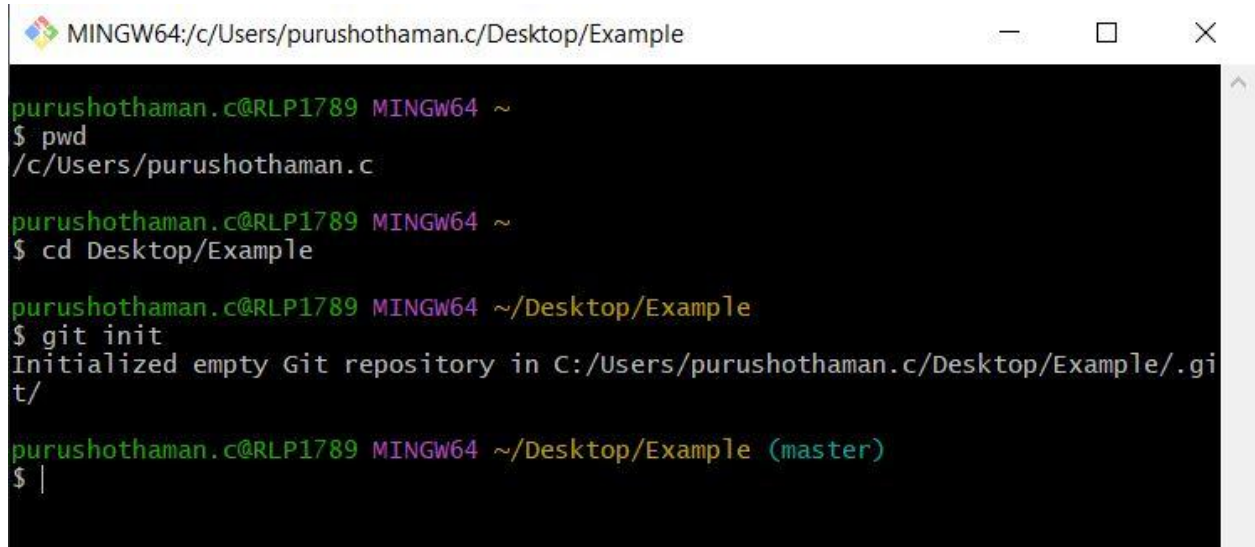
- Source code management tool
- Github is cloud version of git
- Git has 2 types of environment

1. Local > only access through our system 2. Remote > shared or distributed repository

1. Working directory > staging area > local repository > 2. Remote repository
Local Remote

Command :

1. **\$ git init** --> to initialize local repository



```
MINGW64:/c/Users/purushothaman.c/Desktop/Example
purushothaman.c@RLP1789 MINGW64 ~
$ pwd
/c/Users/purushothaman.c
purushothaman.c@RLP1789 MINGW64 ~
$ cd Desktop/Example
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Example
$ git init
Initialized empty Git repository in C:/Users/purushothaman.c/Desktop/Example/.git/
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Example (master)
$ |
```

2. **\$ git clone** --> clone a remote on local

Pull

>> Moving from github(Remote) to local repository
\$ git clone <url>(url of remote repository to pull)

Push

>> Moving from local repository to remote.

\$ git add >> Working directory to staging area

\$ git add -A (or) \$ git add --all >> Two is same to add all files

```
MINGW64:/c/Users/purushothaman.c/Desktop/Sample
nothing added to commit but untracked files present (use "git add" to track)
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git add Two.txt
fatal: pathspec 'Two.txt' did not match any files
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git add -A
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   One.txt
        new file:   folder/Subfolder/Four.txt
        new file:   folder/Three.txt
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ |
```

\$ git commit >> Staging area to local repository

\$ git reset >> To undo changes

```
MINGW64:/c/Users/purushothaman.c/Desktop/Sample
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   One.txt
        new file:   folder/Subfolder/Four.txt
        new file:   folder/Three.txt
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git reset
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        One.txt
        folder/

nothing added to commit but untracked files present (use "git add" to track)
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
```

\$ git add . >> Move to the inside folder. The files in the folder will be committed the files outside the folder are in the staging area.

\$ git commit -m "message" >> To commit Staging area to local repository

\$ git reset HEAD~ >> To reset in Staging area to local repository

\$ git diff Two.txt >> Show the difference made in the file

```
MINGW64:/c/Users/purushothaman.c/Desktop/Sample

changes to be committed:
(use "git rm --cached <file>..." to unstage)
    new file:   One.txt
    new file:   Two.txt
    new file:   folder/Subfolder/Four.txt
    new file:   folder/Three.txt

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git diff Two.txt

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git diff Two.txt
diff --git a/Two.txt b/Two.txt
index e69de29..178dfb2 100644
--- a/Two.txt
+++ b/Two.txt
@@ -0,0 +1 @@
+hjklfjklns,nfns,mm,
\ No newline at end of file

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ |
```

\$ git diff --cached Two.txt >> To track the difference.

```
MINGW64:/c/Users/purushothaman.c/Desktop/Sample

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git add Eight.txt

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ ls
Eight.txt  One.txt  Six.txt  Two.txt  folder/

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git diff --cached Two.txt
diff --git a/Two.txt b/Two.txt
new file mode 100644
index 0000000..345f701
--- /dev/null
+++ b/Two.txt
@@ -0,0 +1,3 @@
+hjklfjklns,nfns,mm,
+
+Lets move
\ No newline at end of file

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ |
```


\$ **git rm -f** <filename> >> delete only one file

```
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git rm -f Two.txt
rm 'Two.txt'

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Eight.txt
    new file:   One.txt
    new file:   Six.txt
    new file:   folder/Subfolder/Four.txt
    new file:   folder/Three.txt
```

\$ **git rm -r** <folder path> >> delete all the files in the folder

\$ **git log** >> Show the history of committed

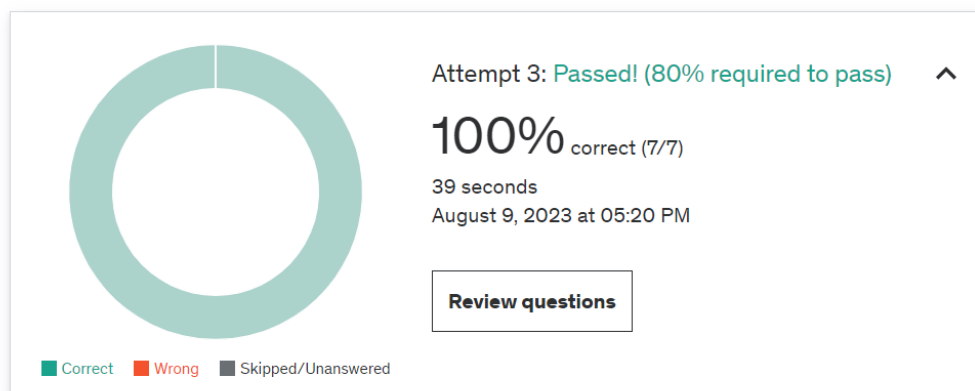
```
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git log
commit a6b0b8c58ab248580a321c2a14ecf5427f6a29f7 (HEAD -> master)
Author: Purushothaman Chinnappan <purushothaman.c@itp.objectfrontier.com>
Date:   Wed Aug 9 17:26:39 2023 +0530

    To be added in local repo

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$
```

Practice Test 1 - Results

7 questions | 15 minutes | 80% correct required to pass



\$ git branch >> To view the branch name

```
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git branch
* master

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git branch new

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git branch
* master
  new

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ |
```

\$ git branch <new branch name> >> To add new branch

\$ git checkout <branch name to switch> >> It is used to switch branch

```
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git checkout new
Switched to branch 'new'

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (new)
$ git branch
  master
* new

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (new)
$ |
```

\$ git merge <branch name> >> It is used to merge to the old one.

```
purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ git merge new
Already up to date.

purushothaman.c@RLP1789 MINGW64 ~/Desktop/Sample (master)
$ |
```