

PRE-JOINING LEARNING PATH

VERSION CONTROL

Q. What is Git?

Ans. Git is a version control system, It is a tool that helps to track changes in code.

Q. Commands used mostly in GitHub.

- (i) `git clone [url]`: retrieve an entire repository from a hosted location via URL
- (ii) `git status`: show modified files in working directory, staged for your next commit
`git`
- (iii) `add [file]`: add a file as it looks now to your next commit (stage)
- (iv) `git commit -m "[descriptive message]"`: commit your staged content as a new commit snapshot
- (v) `git push [alias] [branch]`: Transmit local branch commits to the remote repository branch
- (vi) `git pull`: fetch and merge any commits from the tracking remote branch

Q. How to create Repository from local environment?

Step 1: Run “`git init`” command.

Step 2: go to github.com and create a new repository without `readme.md` file.

Step 3: Run “`git remote add origin [link]`” command.

(Run “`git remote -v`” command for checking the remote origin)

Step 4: Run “`git branch`” to check branch of repository where code is going to push. And then run “`git branch -M main`” command to change name of master branch to main branch.

Q. How to merge branch in github?

=> Follow this step:

git branch (to check branch)

git branch -M main (to rename branch)

git checkout <branch name> (to navigate)

git checkout -b <new branch name> (to create new branch)

git branch -d <branch name> (to delete branch)

git merge [branch] (merge branch)

git diff: diff of what is changed but not staged

Q. How to undo changes in github?

Ans: git reset : clear staging area, rewrite working tree from specified commit

With tutorial:

M : modified - u : unmodified

README.MD -> mark down

cd- change directory

ls:list - md :make directory

Git

Version Control System is a tools that helps to track changes in code

Git is a **Version Control System**. It is :

popular

free & Open Source

fast & scalable

- 1) track the history
- 2) collaborate

Github

Website that allows developers to store and manage their code using Git.

<https://github.com>

New repo:


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).


Owner *

Repository name *

 student-apnacollege ▾


/

apnacollege-demo


 apnacollege-demo is available.

Great repository names are short and memorable. Need inspiration? How about [friendly-waffle](#) ?

Description (optional)

☒  **Public**

Anyone on the Internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template:None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

```
apnacollege-demo / README.md in main
Edit Preview
1 # apnacollege-demo
2 This is my first Git Repository.
3
4 Author - Shradha Khapra
5
```

Setting up Git

Visual Studio Code

Windows (Git Bash)

Mac (Terminal)

Download:

```
git --version
```

Configuring Git

```
git config --global user.name "My Name"
```

```
git config --global user.email "someone@email.com"
```

Setting:

```
git config --list
```

```
shradhakhapra@Shradhas-MacBook-Air ~ % git config --global user.name "Student ApnaCollege"
shradhakhapra@Shradhas-MacBook-Air ~ % git config --global user.email "student@apnacollege.in"
shradhakhapra@Shradhas-MacBook-Air ~ % git config --list
credential.helper=osxkeychain
user.name=Student ApnaCollege
user.email=student@apnacollege.in
```

Clone & status :

Clone & Status

Clone - Cloning a repository on our local machine

```
git clone <- some link ->
```

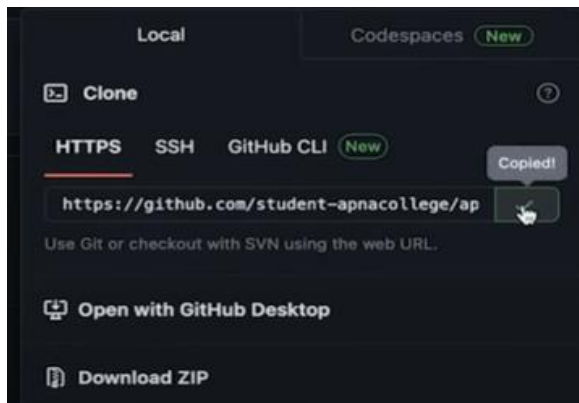
status - displays the state of the code

```
git status
```

remote [Github] local [laptop / PC]

cd → change directory





```
shradhakhapra@Shradhas-MacBook-Air GitDemo % git clone https://github.com/student-apnacollege/apnacollege-demo.git
Cloning into 'apnacollege-demo'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
```

untracked

new files that git doesn't yet track

modified

changed

staged

file is ready to be committed

unmodified

unchanged

change (modified) / new file (untracked)

↓

add (staged)

↓

commit (unchanged)

All file add:

```
shradhakhapra@Shradhas-MacBook-Air apnacollege-demo % git add .
shradhakhapra@Shradhas-MacBook-Air apnacollege-demo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   index.html

shradhakhapra@Shradhas-MacBook-Air apnacollege-demo %
```

Add & Commit

add - adds new or changed files in your working directory to the Git staging area.

```
git add <- file name ->
```

commit - it is the record of change

```
git commit -m "some message"
```



Push Command

push - upload local repo content to remote repo

```
git push origin main
```



Init Command

init - used to create a new git repo

```
git init
```

```
git remote add origin <- link ->
```

```
git remote -v (to verify remote)
```

```
git branch (to check branch)
```

```
git branch -M main (to rename branch)
```

```
git push origin main
```

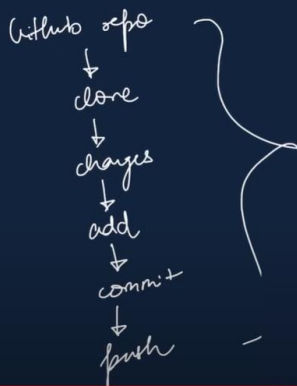
git push -u origin main
↓
set upstream

Final command:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE zsh - LocalRe
● shradhakhapra@Shradhas-MacBook-Air LocalRepo % git add .
● shradhakhapra@Shradhas-MacBook-Air LocalRepo % git commit -m "Fixed readme"
[main 3cd785d] Fixed readme
  1 file changed, 1 insertion(+), 1 deletion(-)
○ shradhakhapra@Shradhas-MacBook-Air LocalRepo % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 340 bytes | 340.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
```

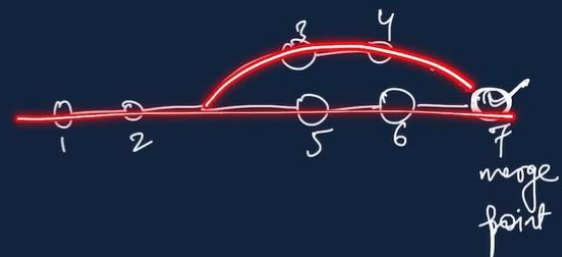
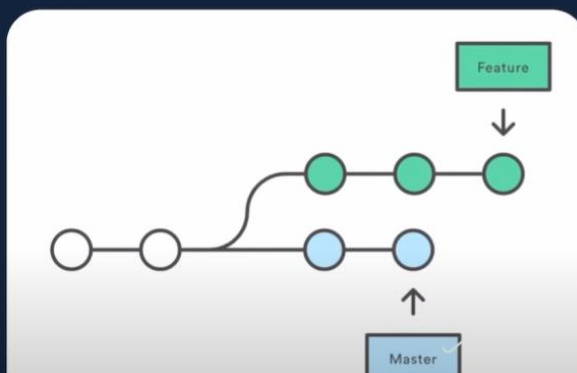
WorkFlow

Local Git



Branch:

Git Branches



Branch Commands

`git branch` (to check branch)

`git branch -M main` (to rename branch)

`git checkout <- branch name ->` (to navigate)

`git checkout -b <- new branch name ->` (to create new branch)

`git branch -d <- branch name ->` (to delete branch)

Merging Code



Way 1

`git diff <- branch name->` (to compare commits, branches, files & more)

`git merge <- branch name->` (to merge 2 branches)

Way 2

Create a PR



Pull Command

`git pull origin main`

used to fetch and download content from a remote repo and immediately update the local repo to match that content.

Git to remote :copy

Pull Request

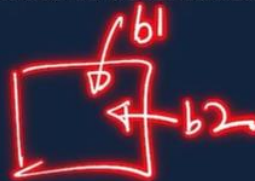
It lets you tell others about changes you've pushed to a branch in a repository on GitHub.



Resolving Merge Conflicts

An event that takes place when Git is unable to automatically resolve differences in code between two commits.

- 1) PR ✓
- 2) Git merge ✓



Fork

A fork is a new repository that shares code and visibility settings with the original “upstream” repository.

Fork is a rough copy.