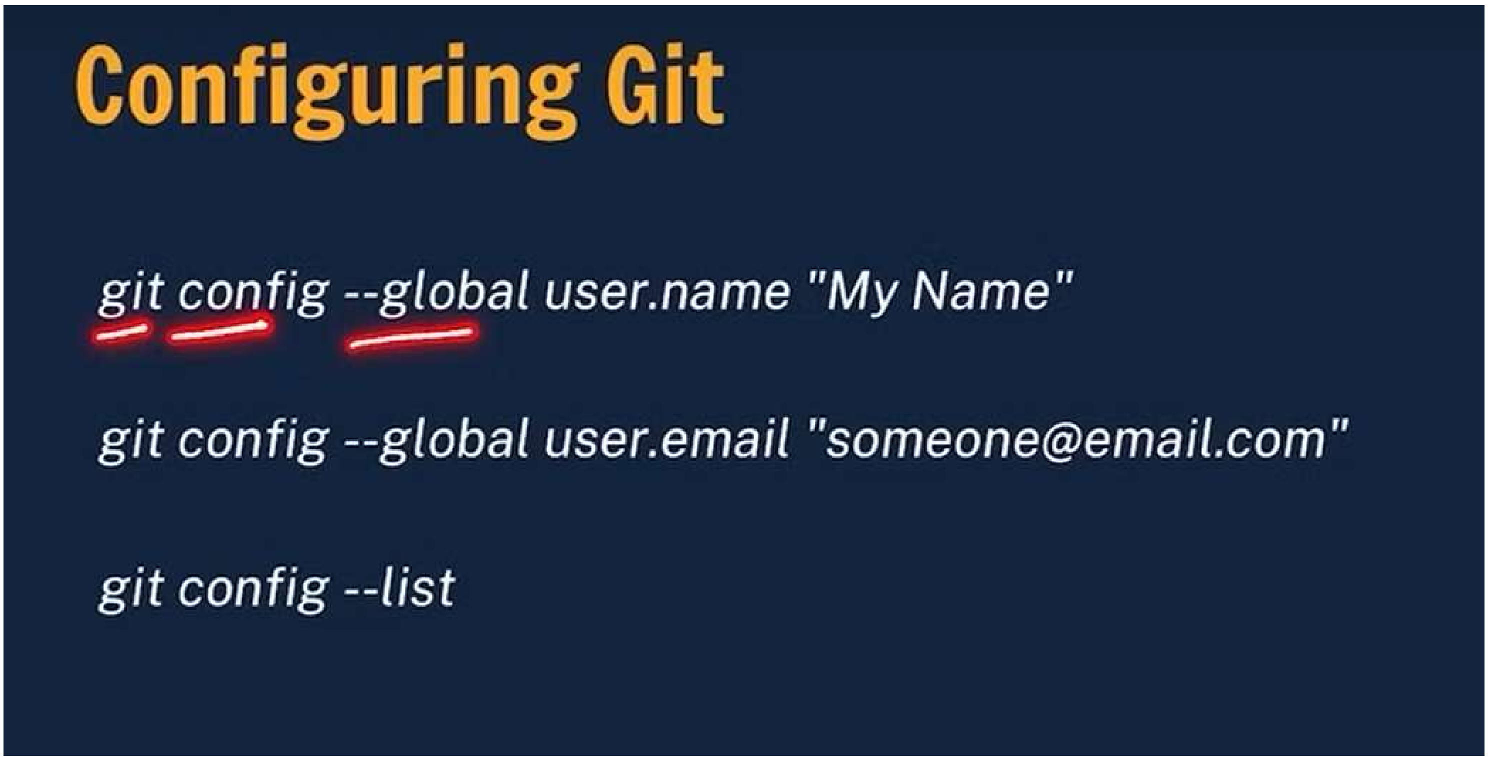
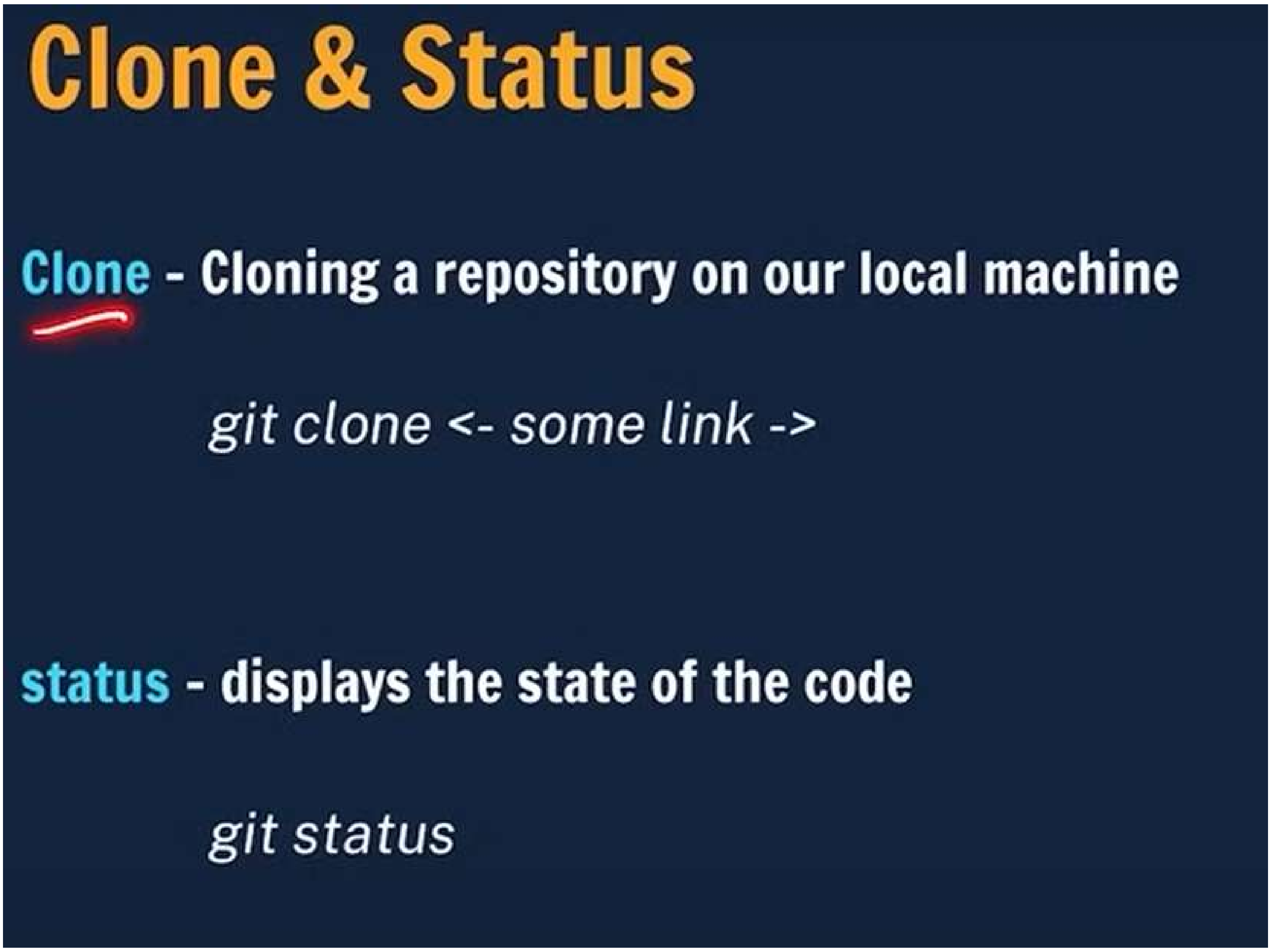


Configuring Git

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

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

git config --list
```



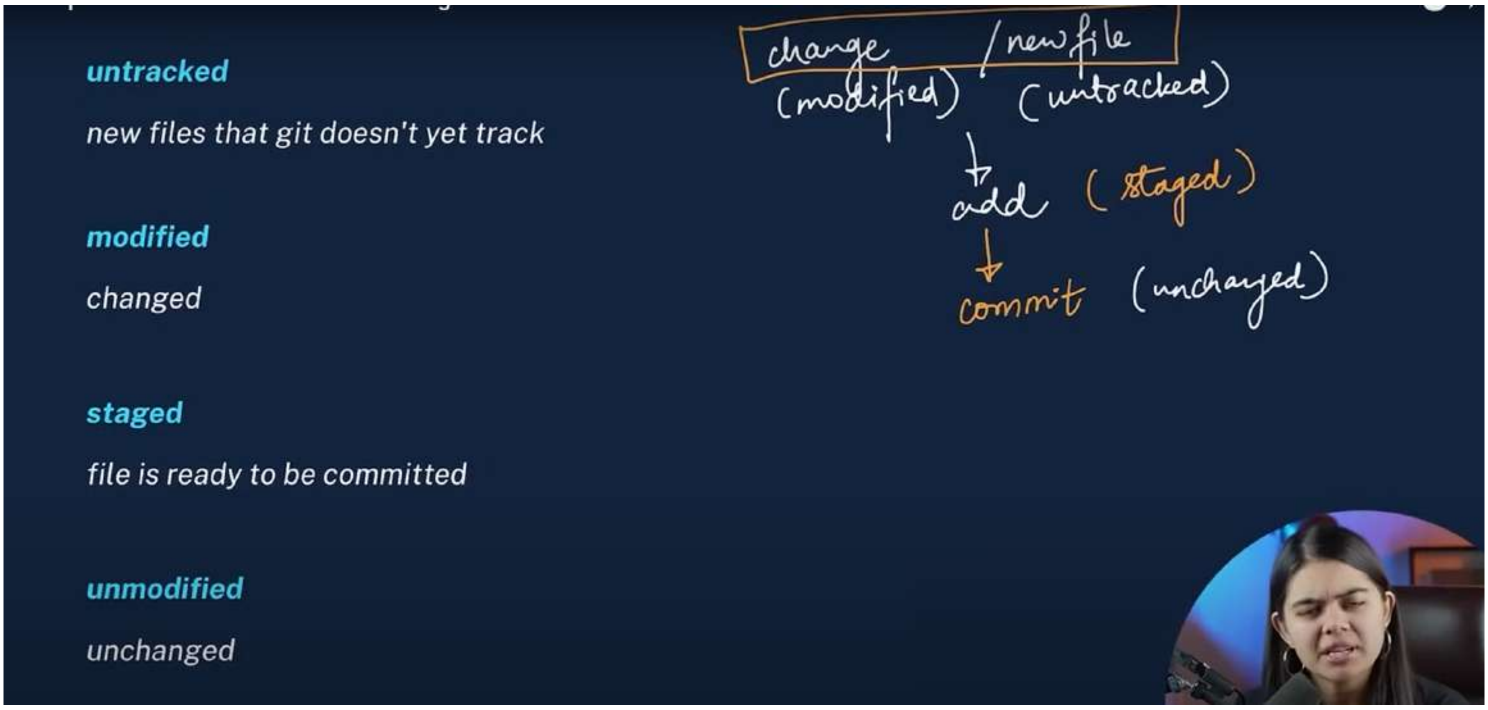
Clone & Status

Clone - Cloning a repository on our local machine

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

status - displays the state of the code

```
git status
```



untracked

new files that git doesn't yet track

modified

changed

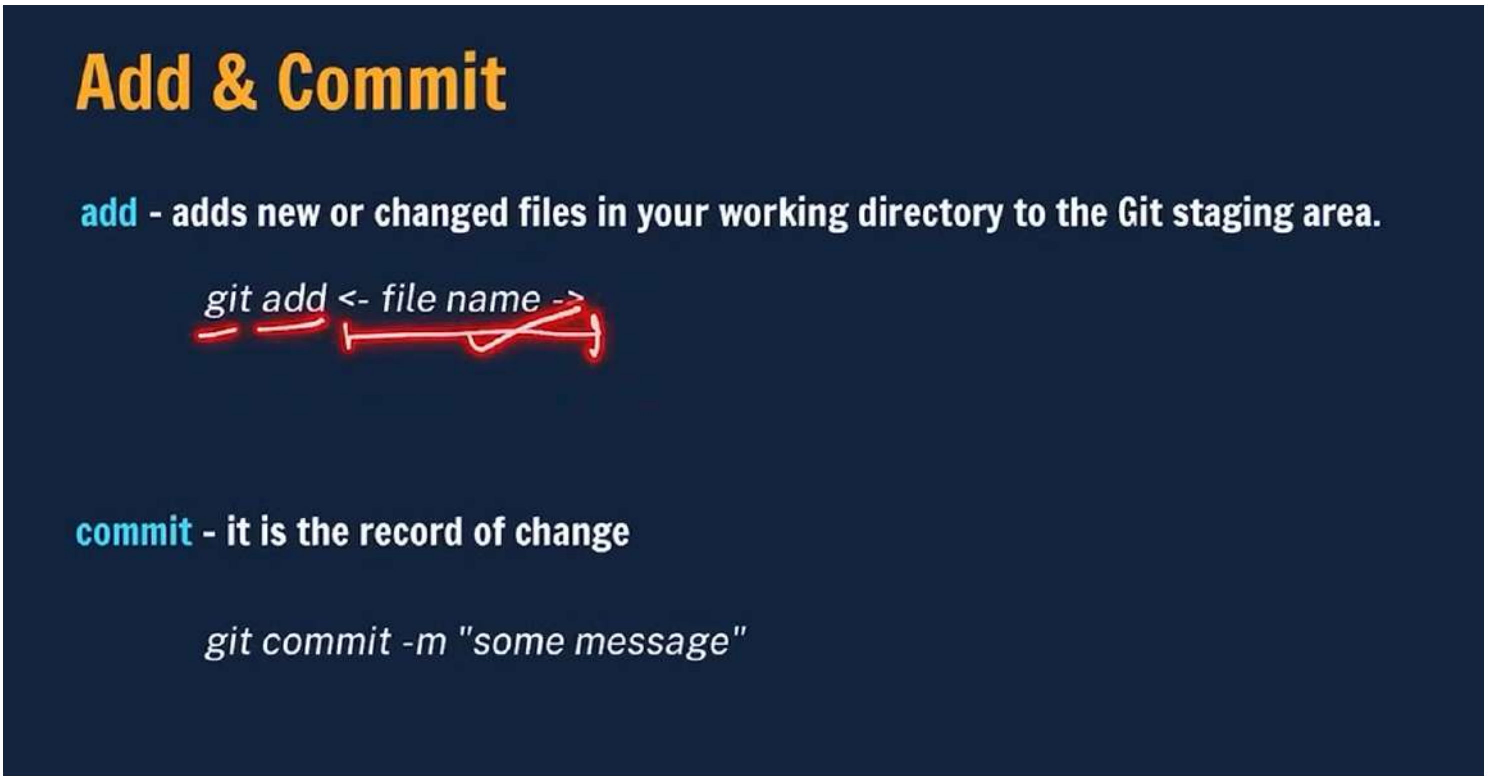
staged

file is ready to be committed

unmodified

unchanged

$$\begin{tikzcd}
\boxed{\text{change (modified)} \; / \; \text{new file (untracked)}} \arrow[d] \\
\text{add (staged)} \arrow[d] \\
\text{commit (unchanged)}
\end{tikzcd}$$



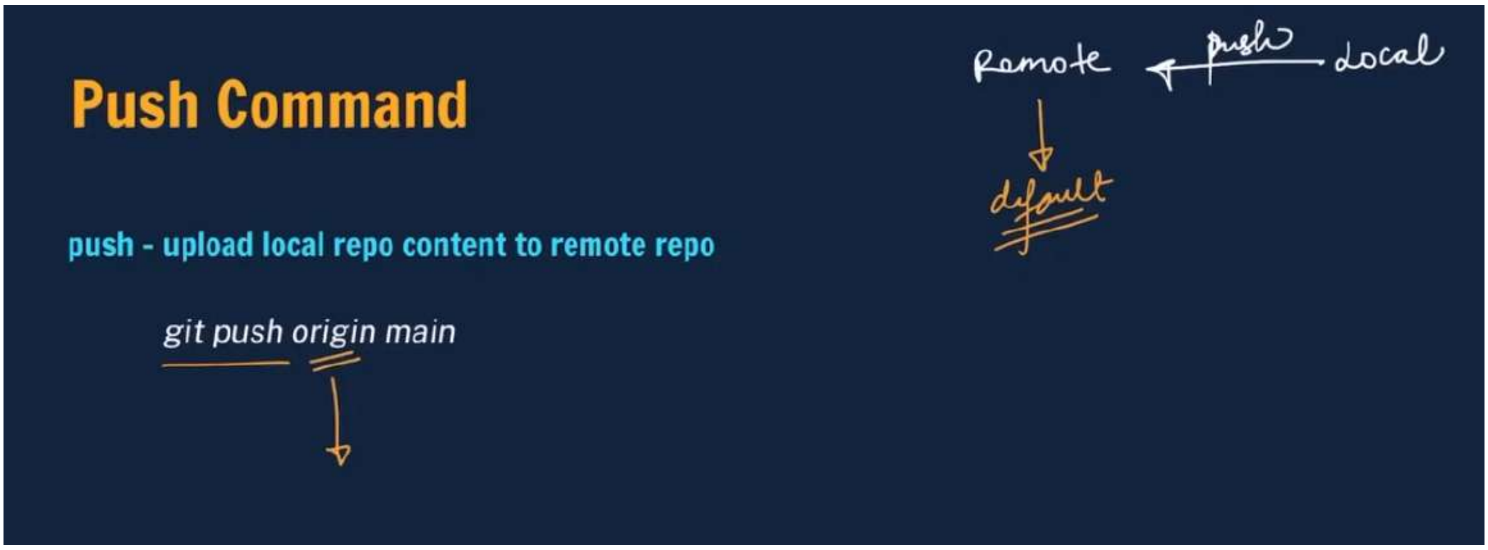
Add & Commit

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"
```

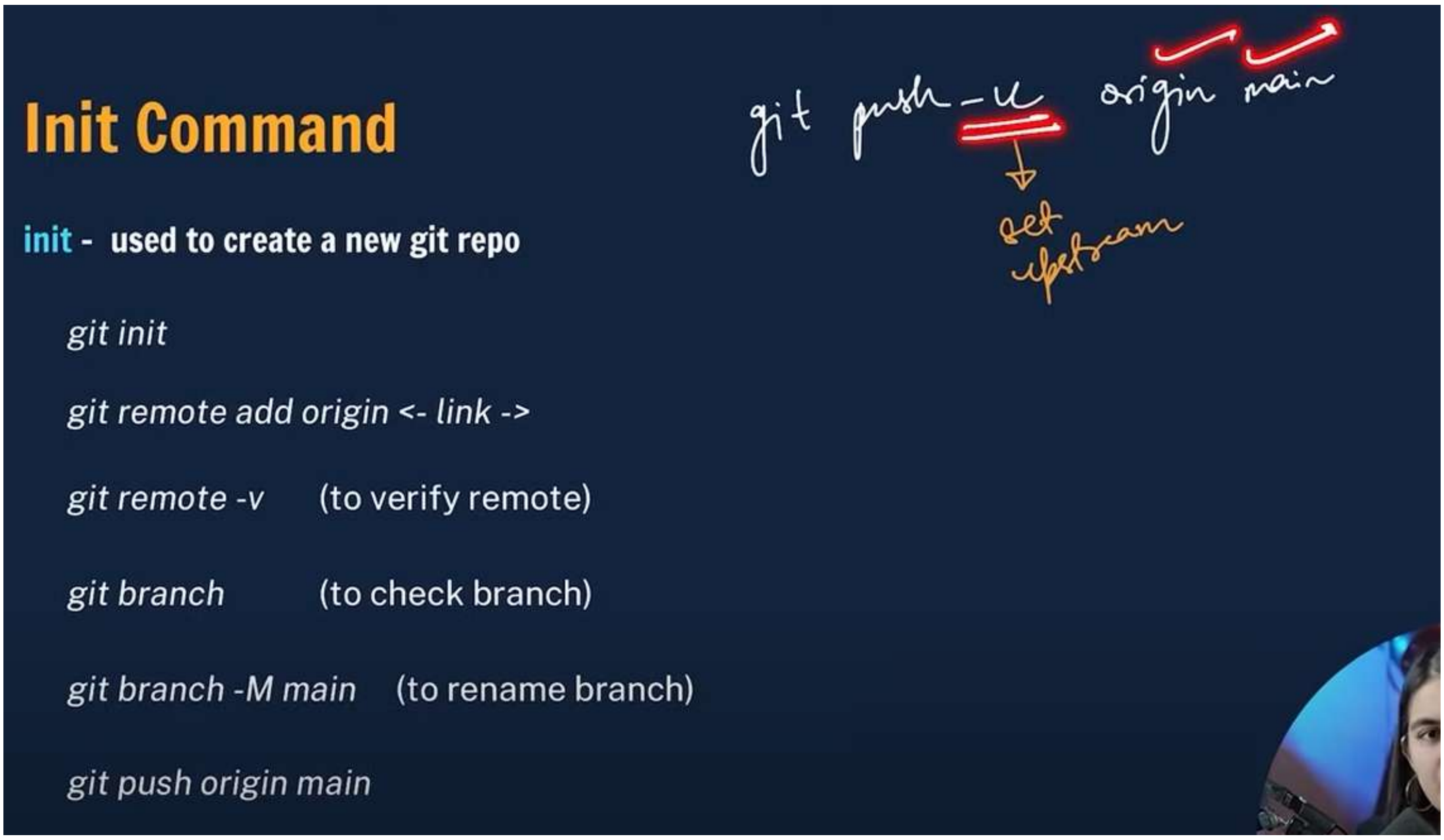


Push Command

push - upload local repo content to remote repo

```
git push origin main
```

$$\begin{tikzcd}
\text{Remote} \arrow[d] & \text{local} \arrow[l, "\text{push}"'] \\
\text{default} &
\end{tikzcd}$$



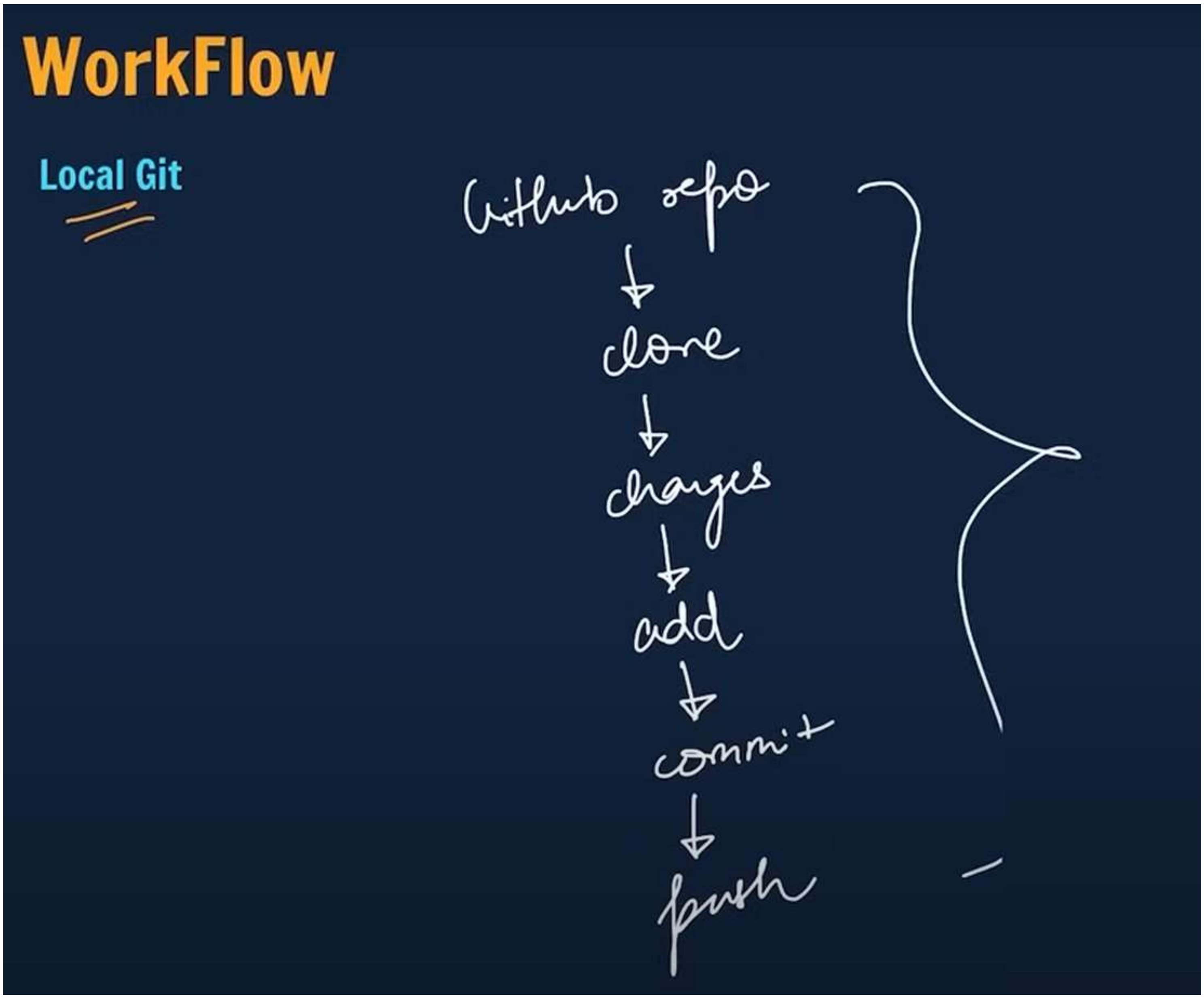
Init Command

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 $\underset{\downarrow}{\underline{-u}}$ origin main

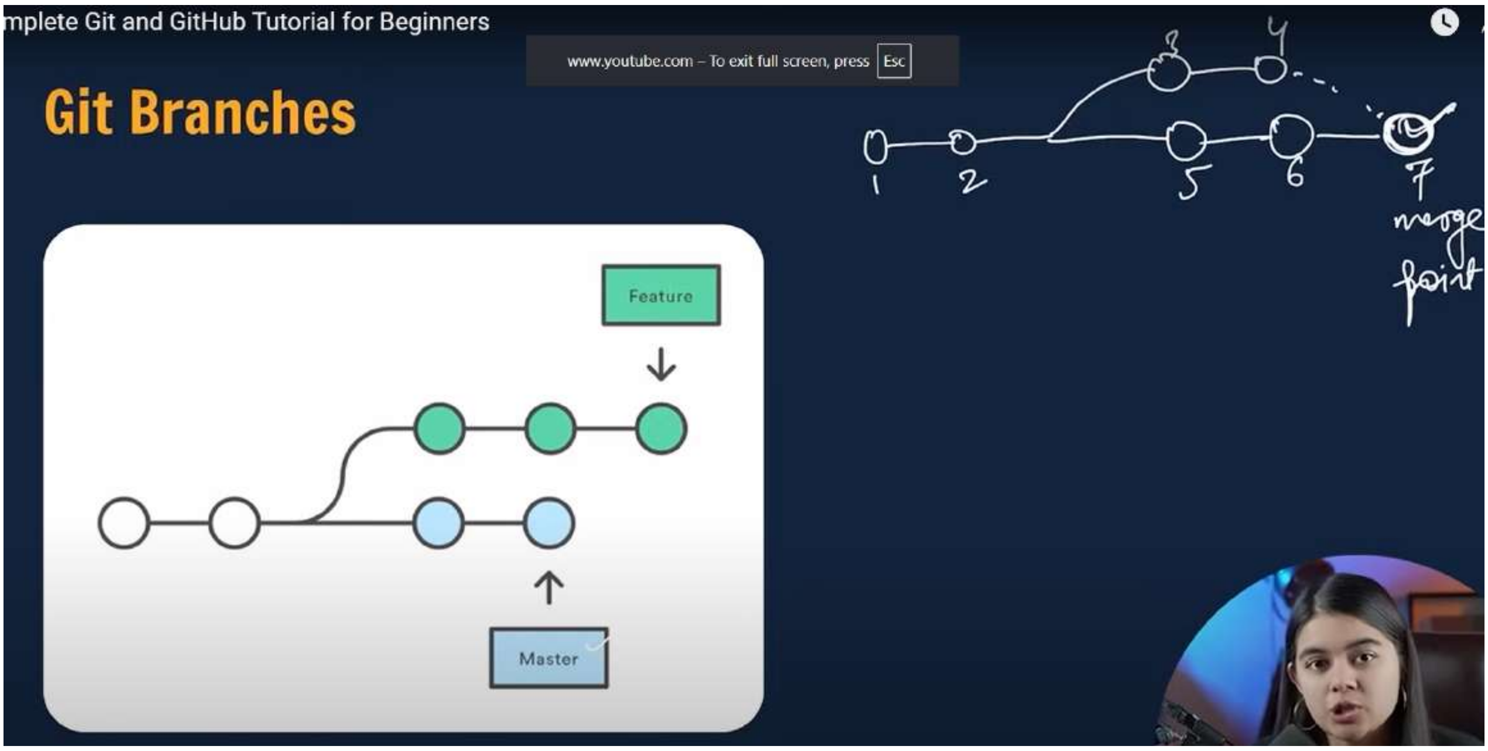
set upstream



WorkFlow

Local Git

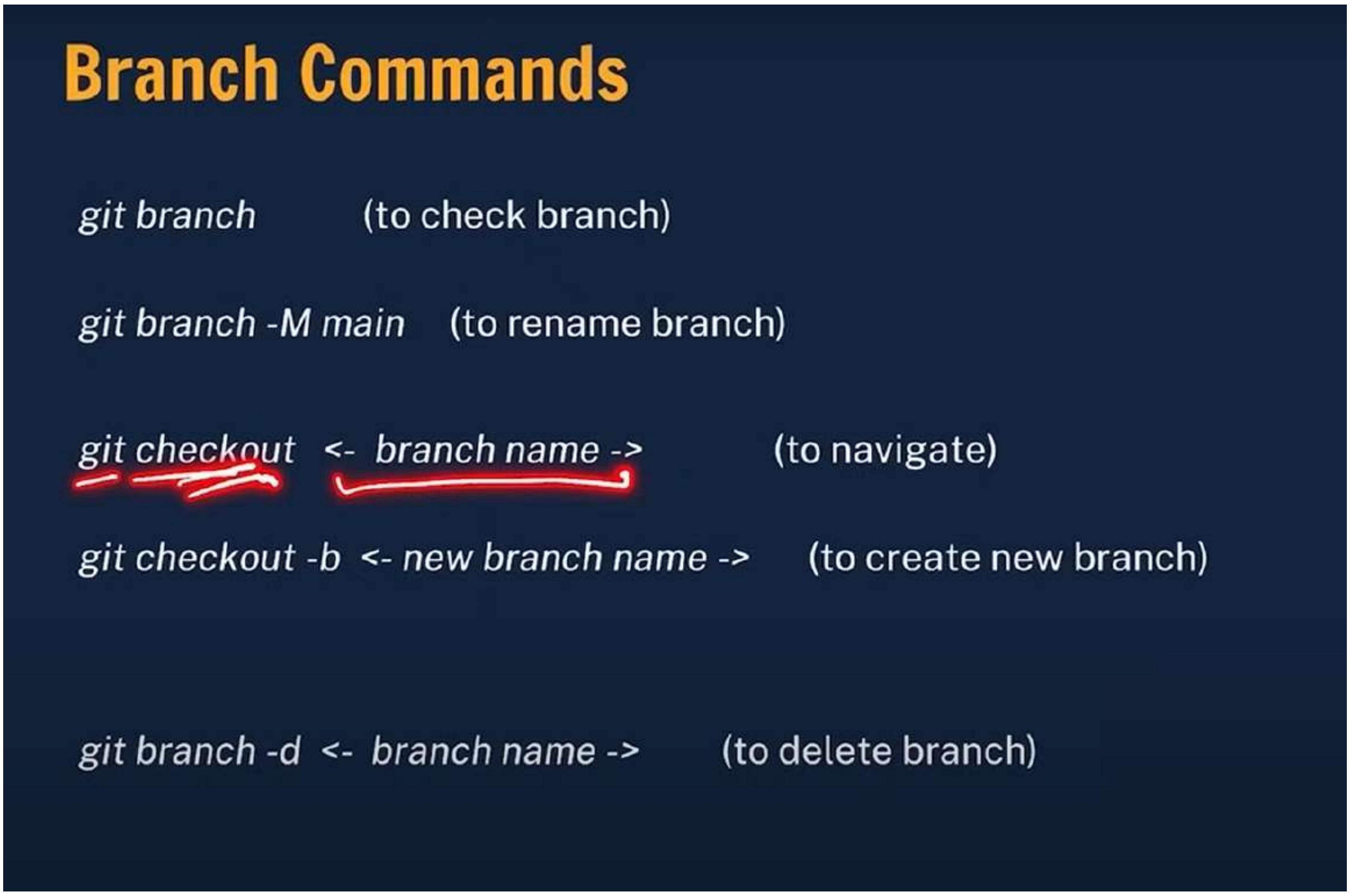
$$\begin{tikzcd}
\text{Github repo} \arrow[d] \\
\text{clone} \arrow[d] \\
\text{changes} \arrow[d] \\
\text{add} \arrow[d] \\
\text{commit} \arrow[d] \\
\text{push}
\end{tikzcd}$$



mplete Git and GitHub Tutorial for Beginners

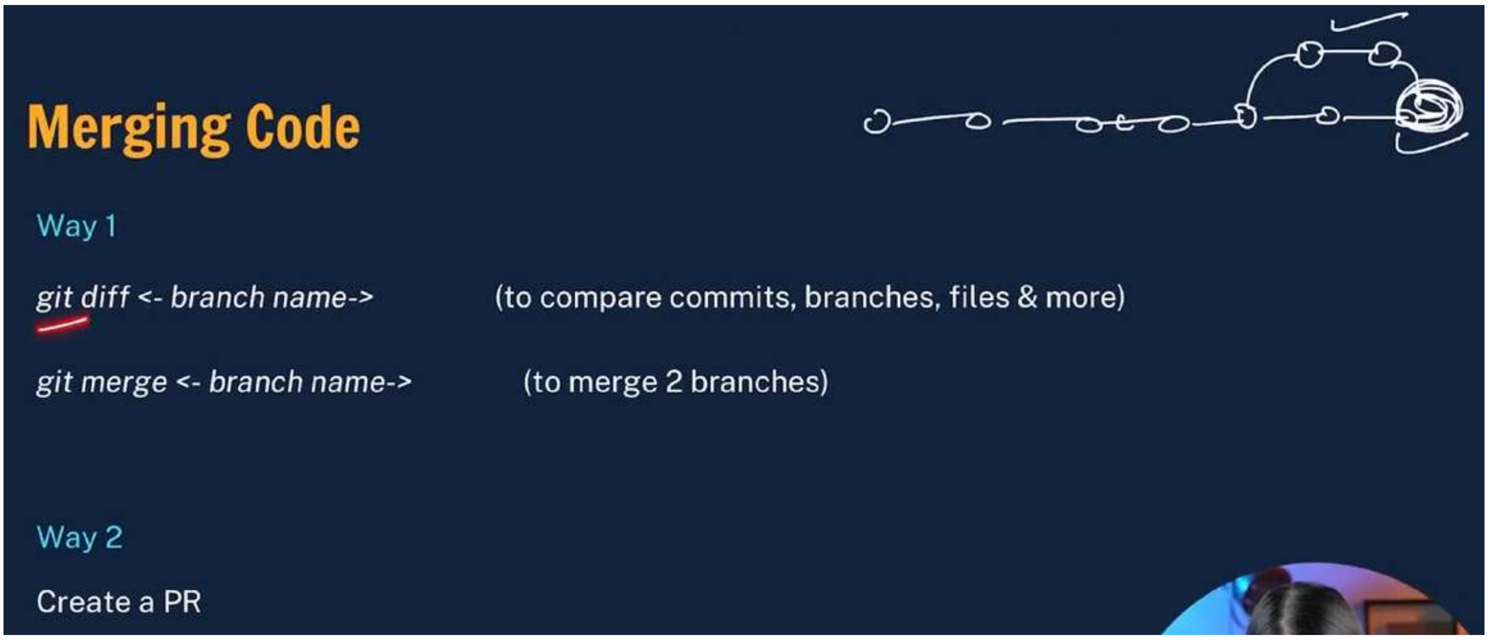
www.youtube.com – To exit full screen, press Esc

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)

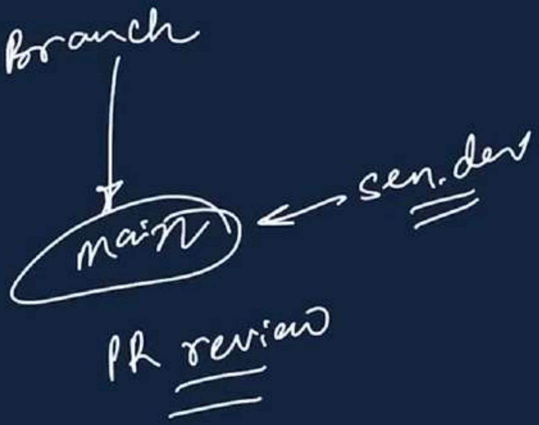
git merge <- branch name-> (to merge 2 branches)

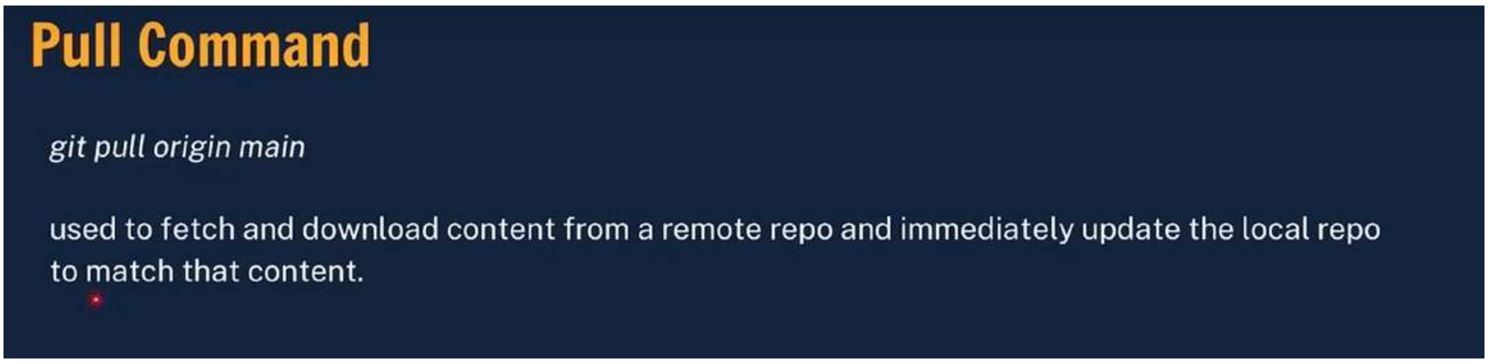
Way 2

Create a PR

Pull Request

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



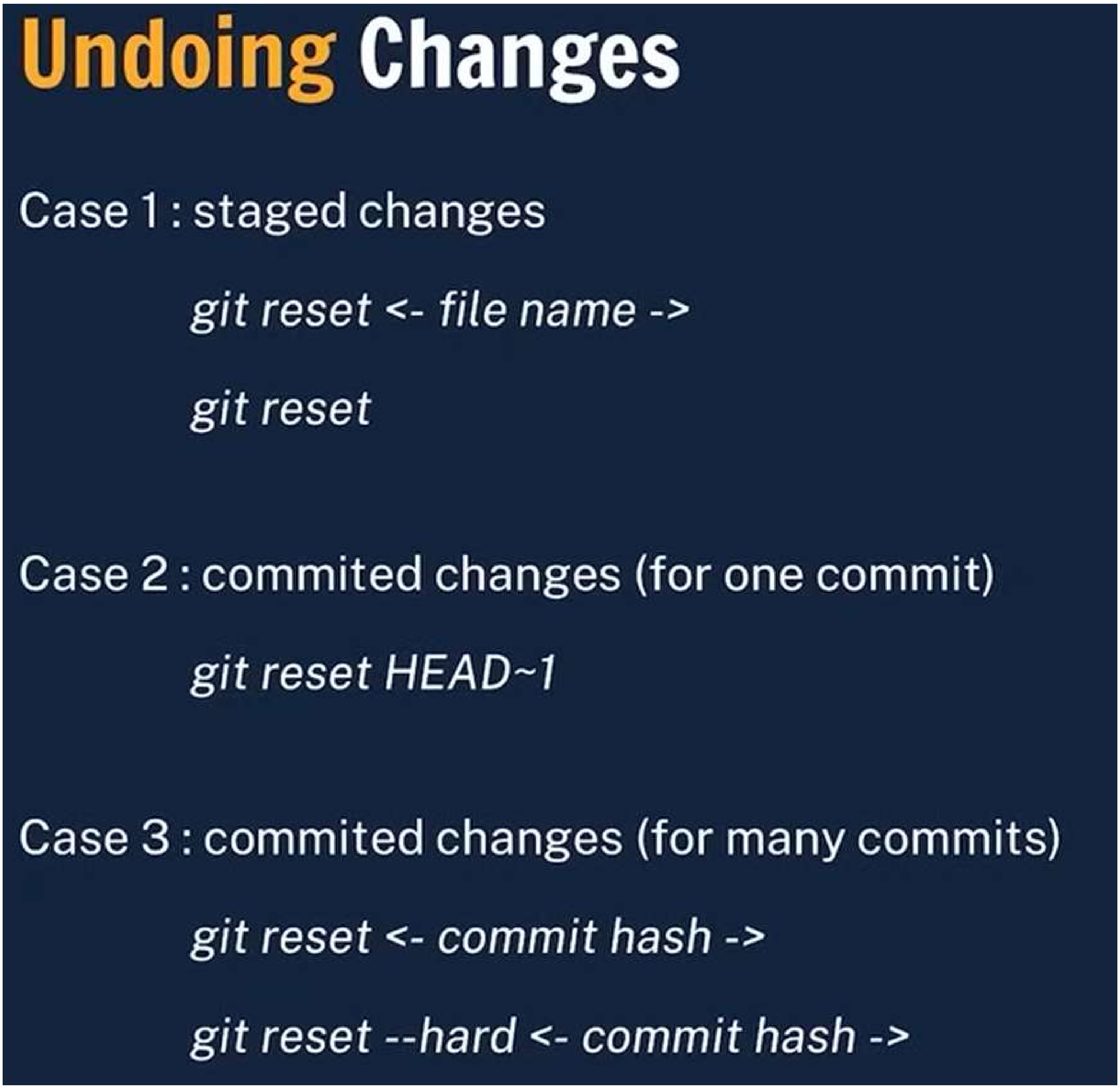

Pull Command

git pull origin main

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

Resolving Merge Conflicts

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



Undoing Changes

Case 1 : staged changes

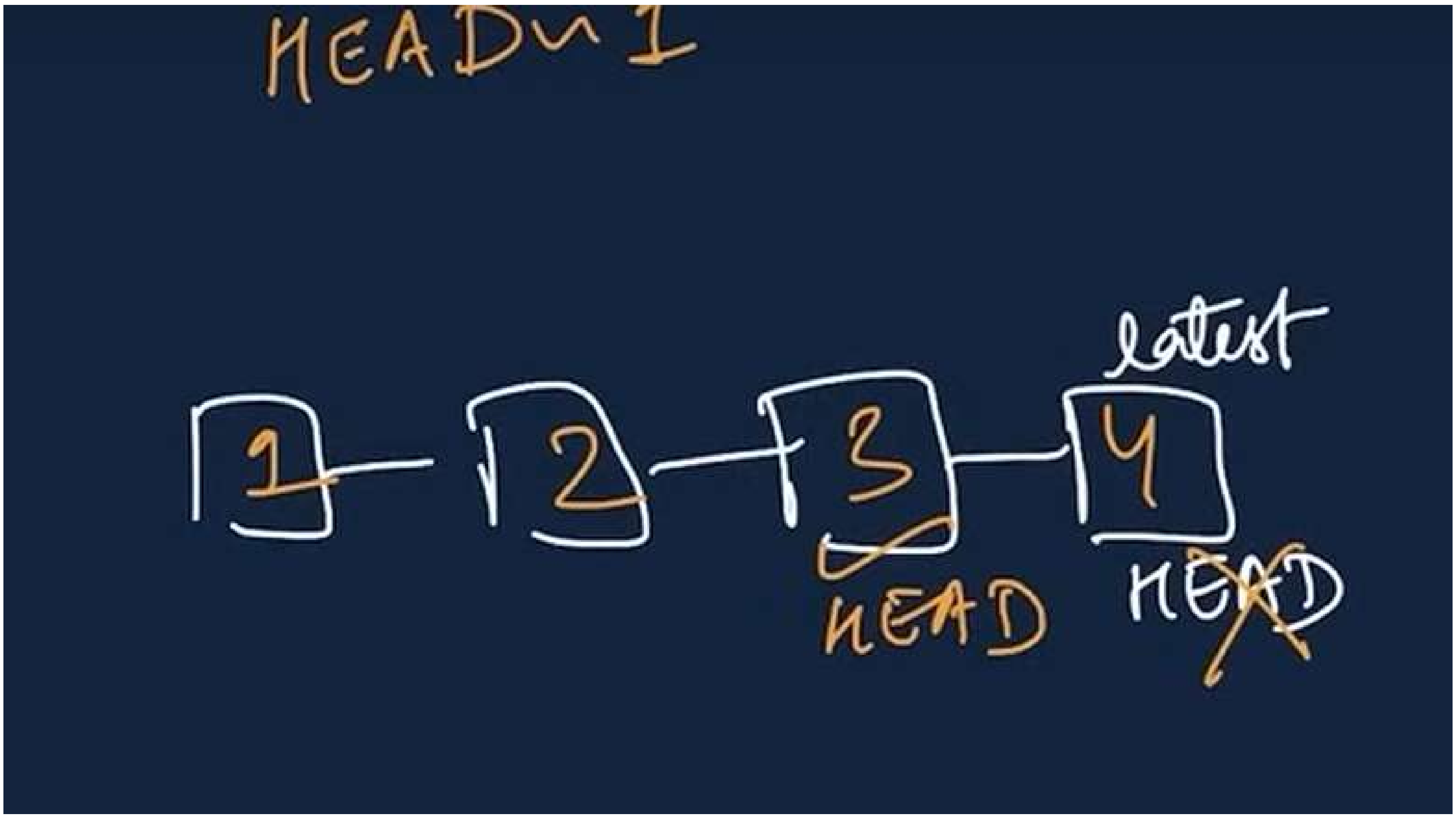
```
git reset <- file name ->
git reset
```

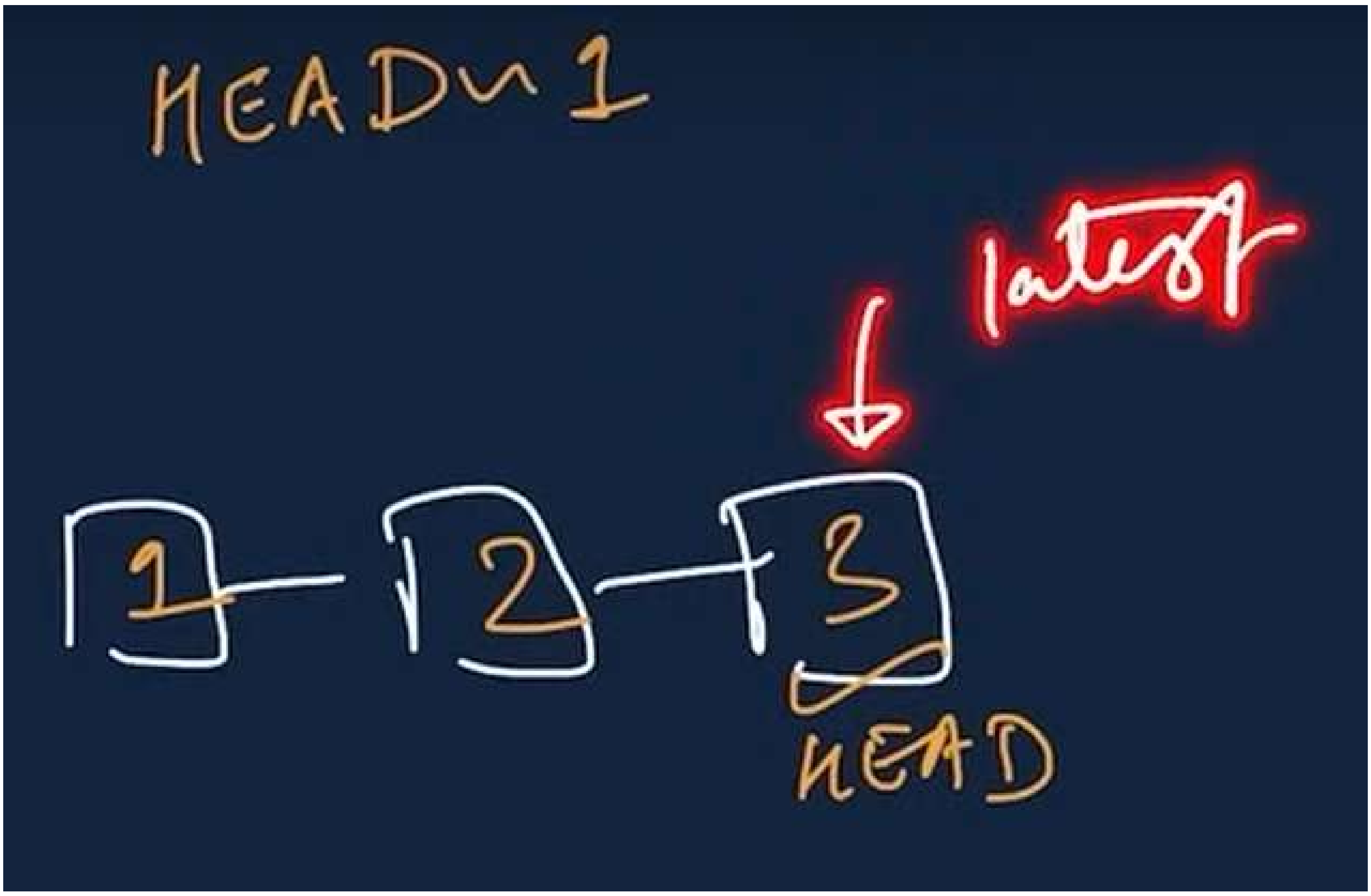
Case 2 : commited changes (for one commit)

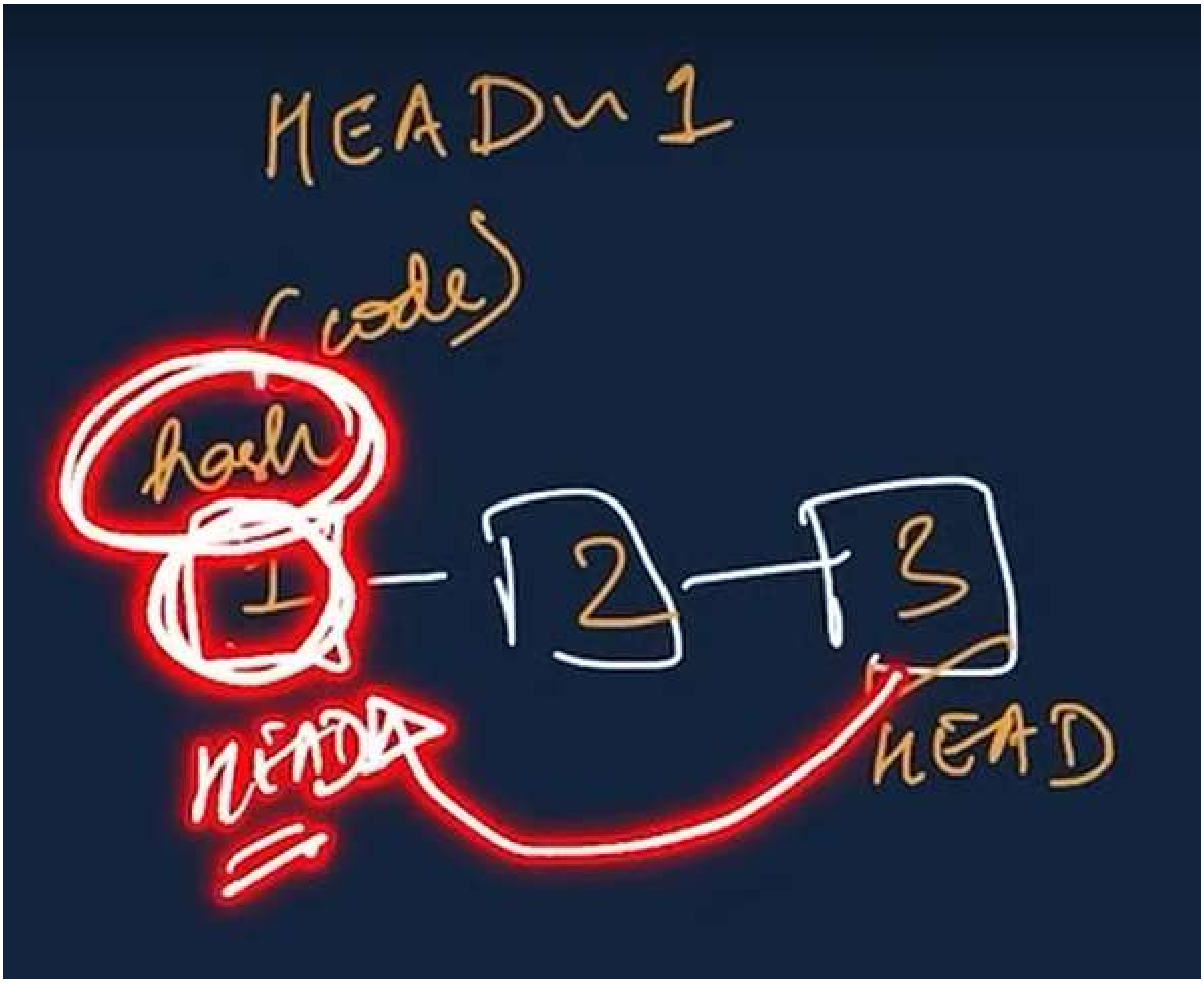
```
git reset HEAD~1
```

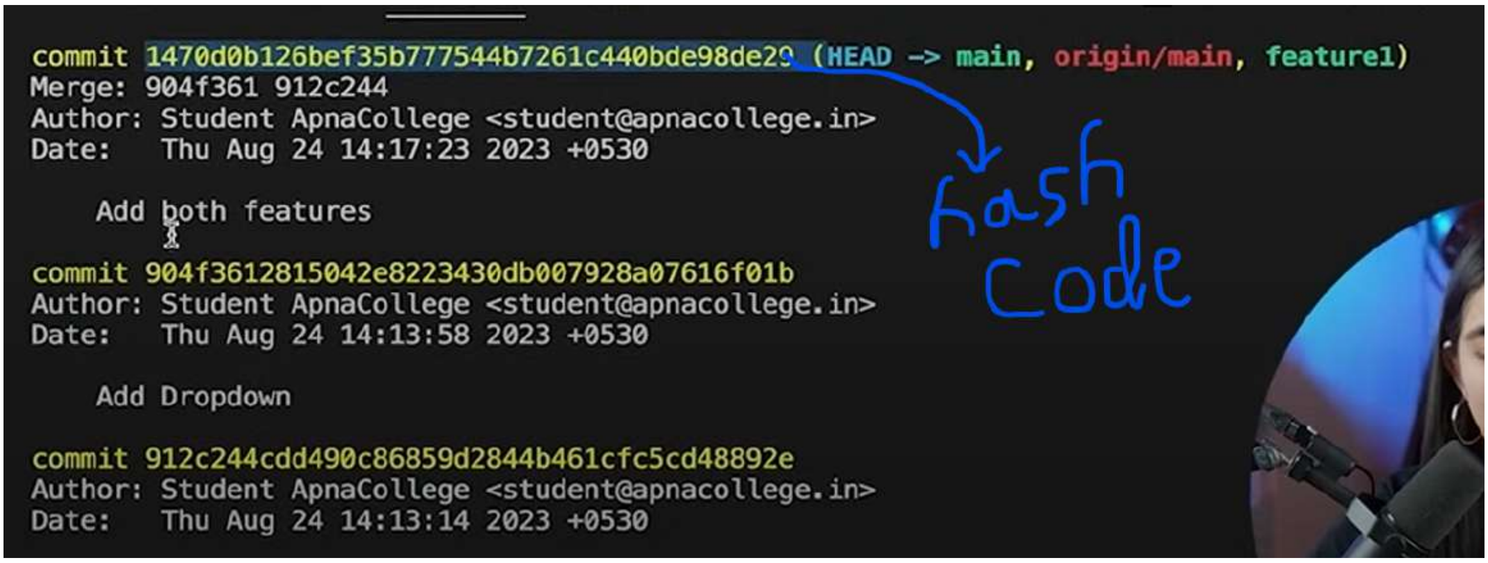
Case 3 : commited changes (for many commits)

```
git reset <- commit hash ->
git reset --hard <- commit hash ->
```

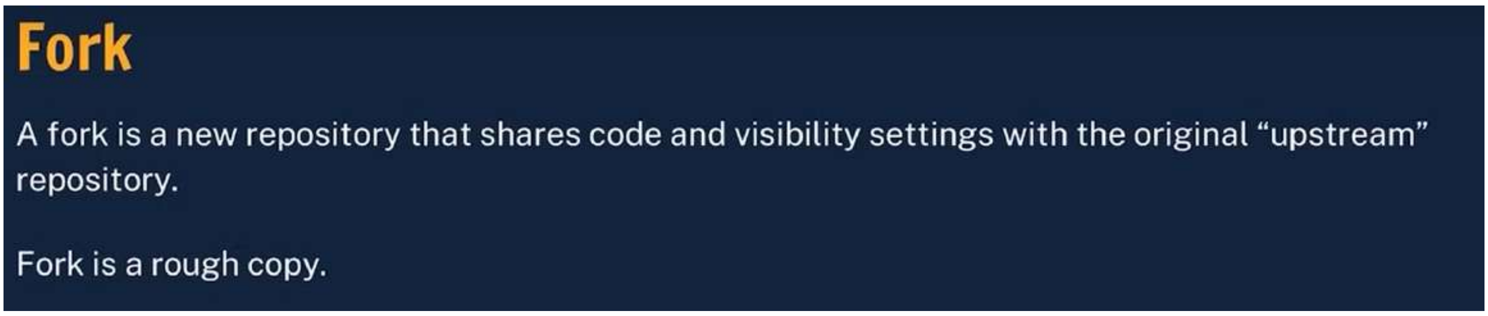

HEAD~1
latest
1
2
3
4
HEAD
HEAD


HEAD~1
latest
1
2
3
HEAD


HEAD = 1
(code)
head
1
2
3
HEAD
HEAD



```
commit 1470d0b126bef35b777544b7261c440bde98de29 (HEAD -> main, origin/main, feature1)
Merge: 904f361 912c244
Author: Student ApnaCollege <student@apnacollege.in>
Date:   Thu Aug 24 14:17:23 2023 +0530

    Add both features

commit 904f3612815042e8223430db007928a07616f01b
Author: Student ApnaCollege <student@apnacollege.in>
Date:   Thu Aug 24 14:13:58 2023 +0530

    Add Dropdown

commit 912c244cdd490c86859d2844b461cfc5cd48892e
Author: Student ApnaCollege <student@apnacollege.in>
Date:   Thu Aug 24 14:13:14 2023 +0530
```



Fork

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

Fork is a rough copy.