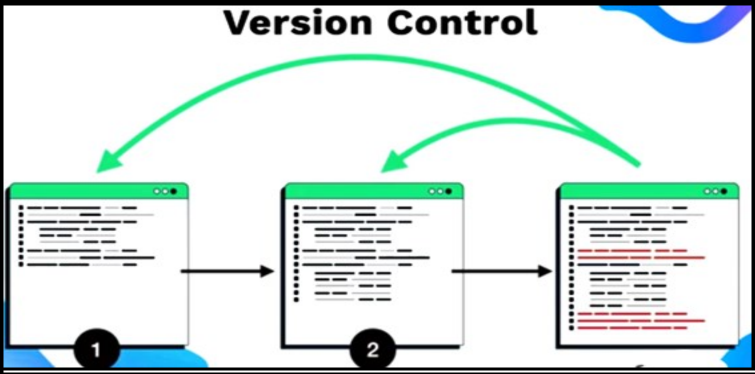
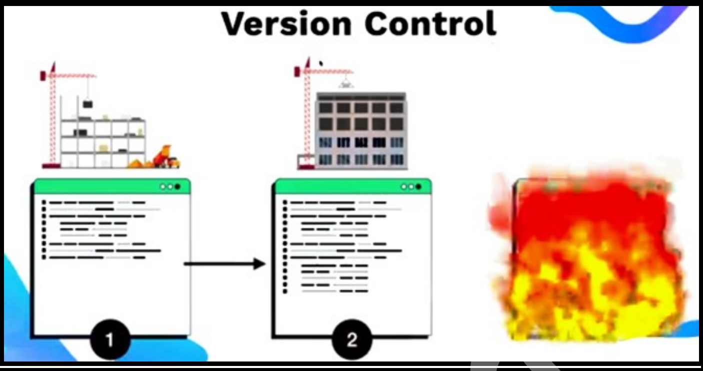
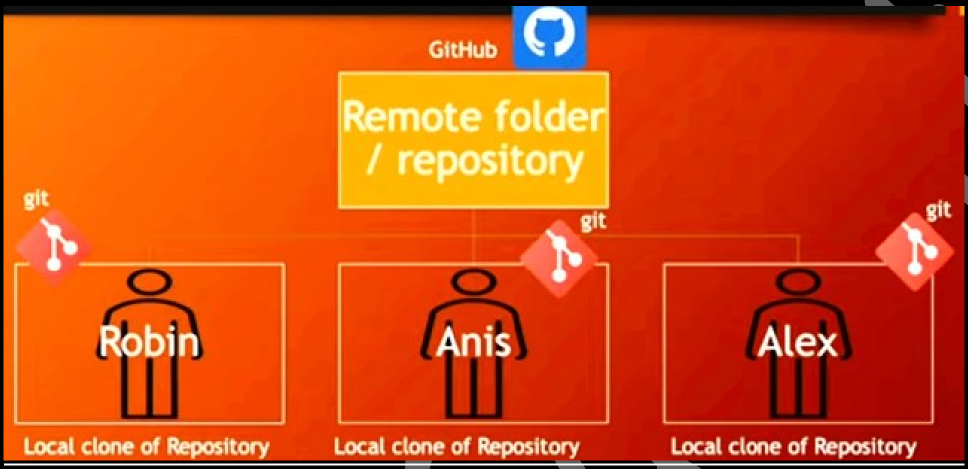
Git & GitHub:






GitHub
Remote folder / repository
git
git
git
Robin
Anis
Alex
Local clone of Repository
Local clone of Repository
Local clone of Repository

- **Git** tracks changes to files on your local machine. It records historical versions or snapshots of files as they exist at specific points in time.
- Git is a **version control system** that helps keep track of changes and supports collaboration in a project.

➢ **GitHub** is a **hosting platform** where you can upload your project directory, so that you can share with anyone from anywhere.

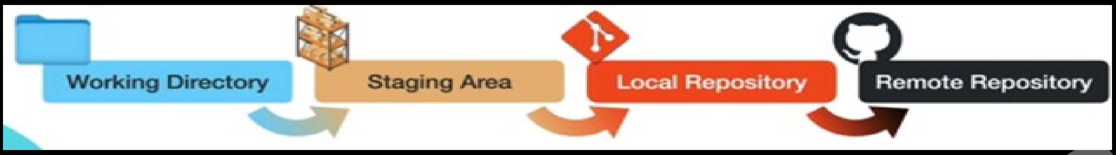
➢ When you and your colleagues work on the project from different local machines and make individual changes, GitHub acts as a **centralized platform** where everyone can **push their changes**. This allows all contributions to be merged and managed effectively in one place.

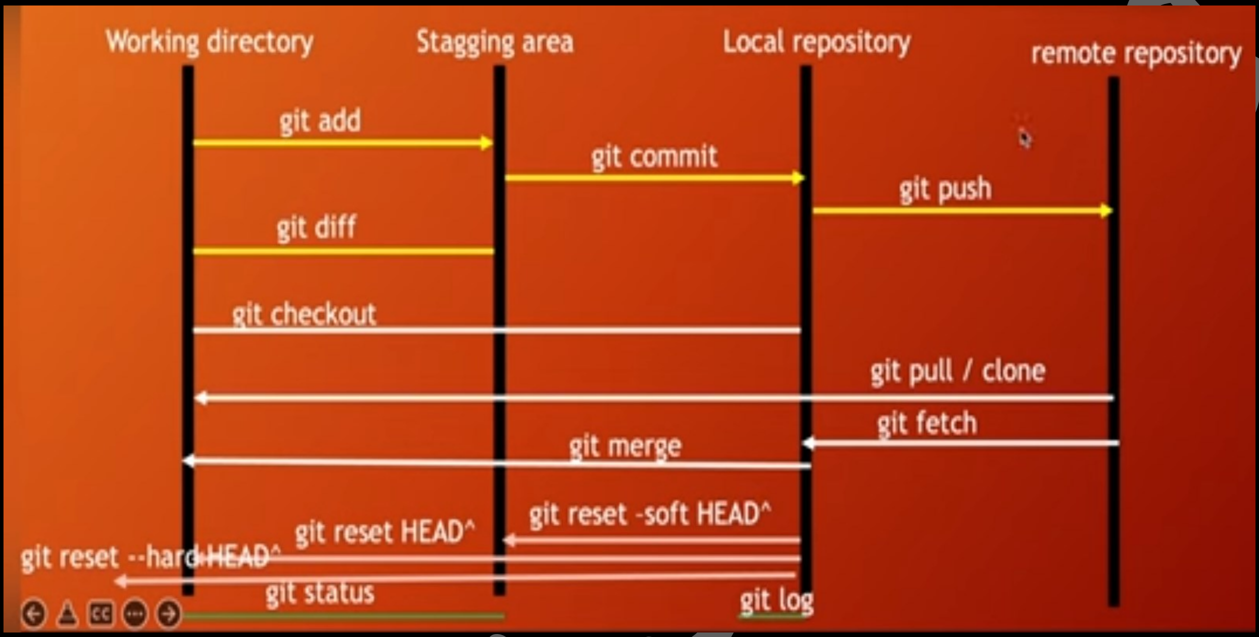
Git is working locally and GitHub is working globally.

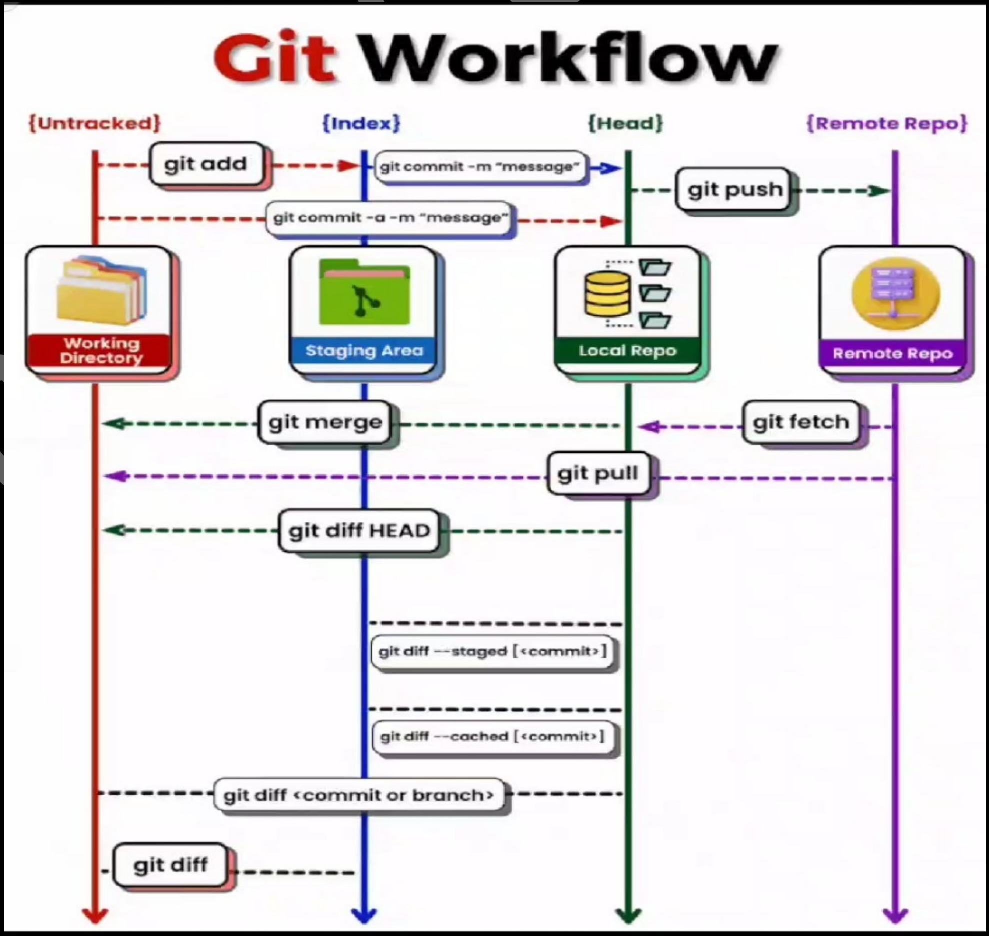
Tanmay Sakpal

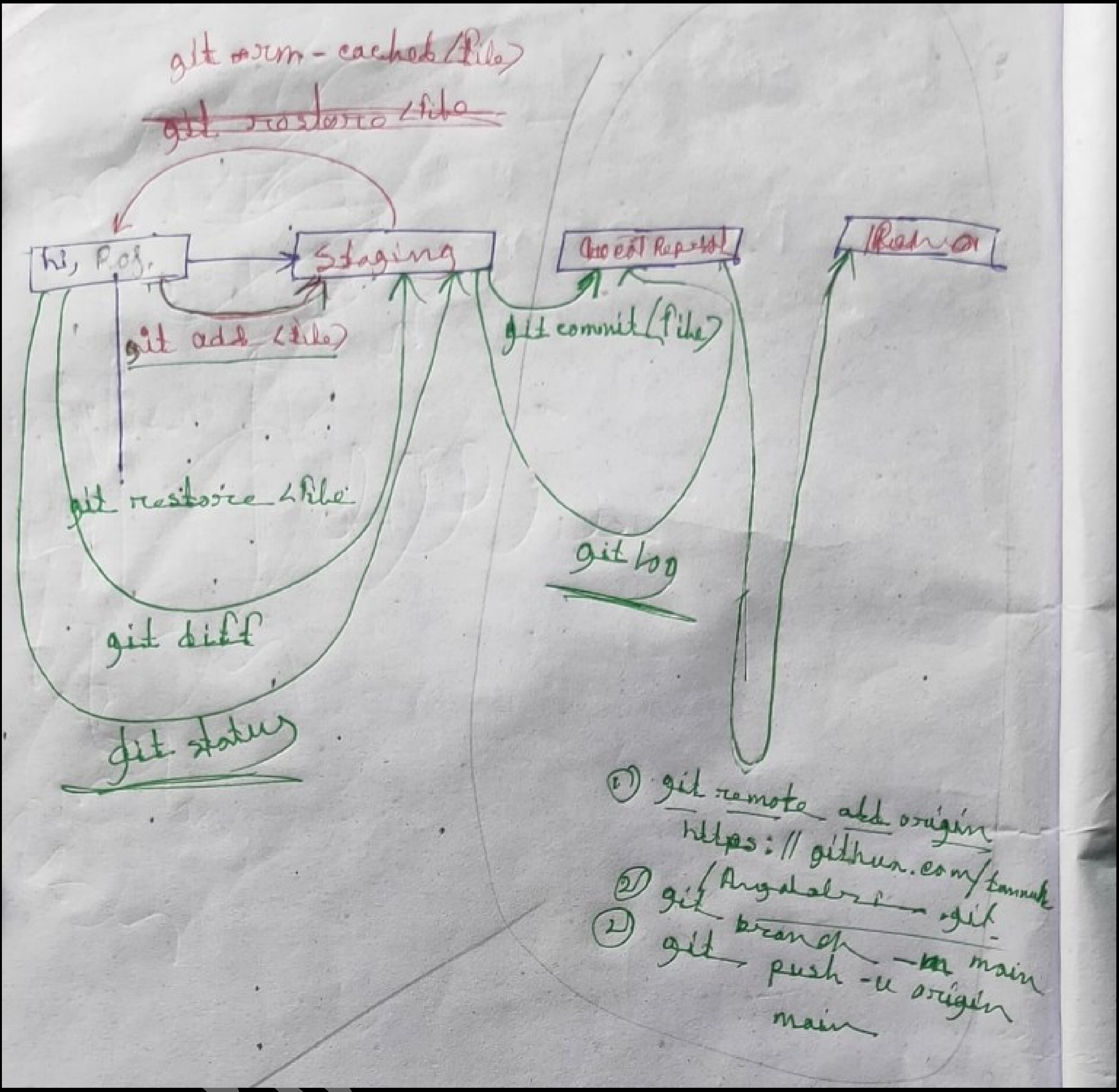
Improve your own version
v1.0
v2.0

Improve your own version
v1.0
v2.0


Working Directory
Staging Area
Local Repository
Remote Repository

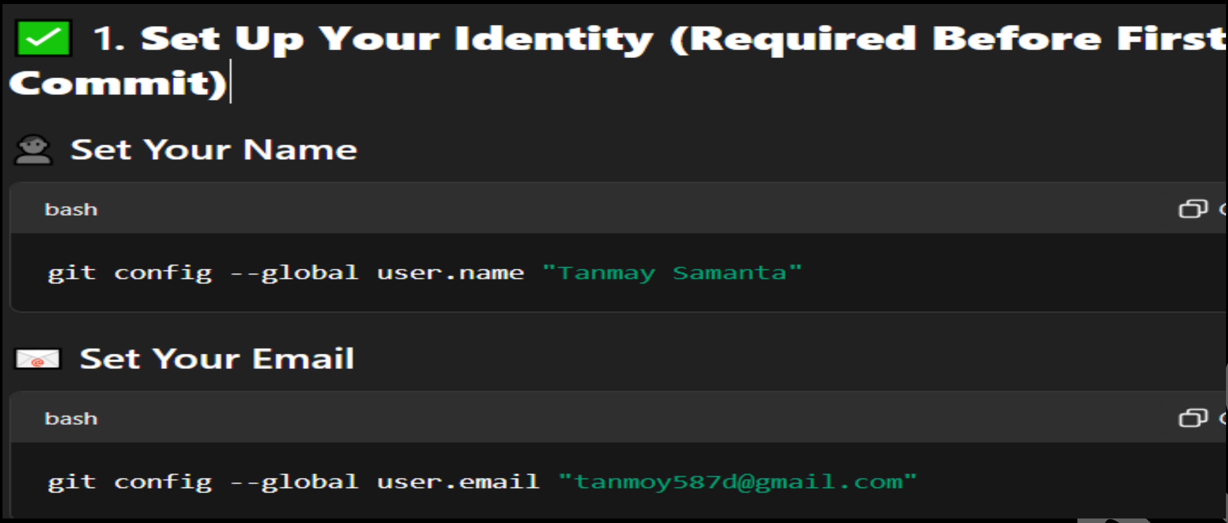

Working directory
Stagging area
Local repository
remote repository
git add
git commit
git push
git diff
git checkout
git pull / clone
git fetch
git merge
git reset -soft HEAD^
git reset HEAD^
git reset --hard HEAD^
git status
git log


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


git rm --cached (file)
~~git restore (file~~
hi, P.of.
Staging
Current Repository
Remote
git add (file)
git commit (file)
git restore (file
git log
git diff
git status
1) git remote add origin https://github.com/tannuk/Angadalri.git
2) git branch -M main
2) git push -u origin main

Tan

Git:



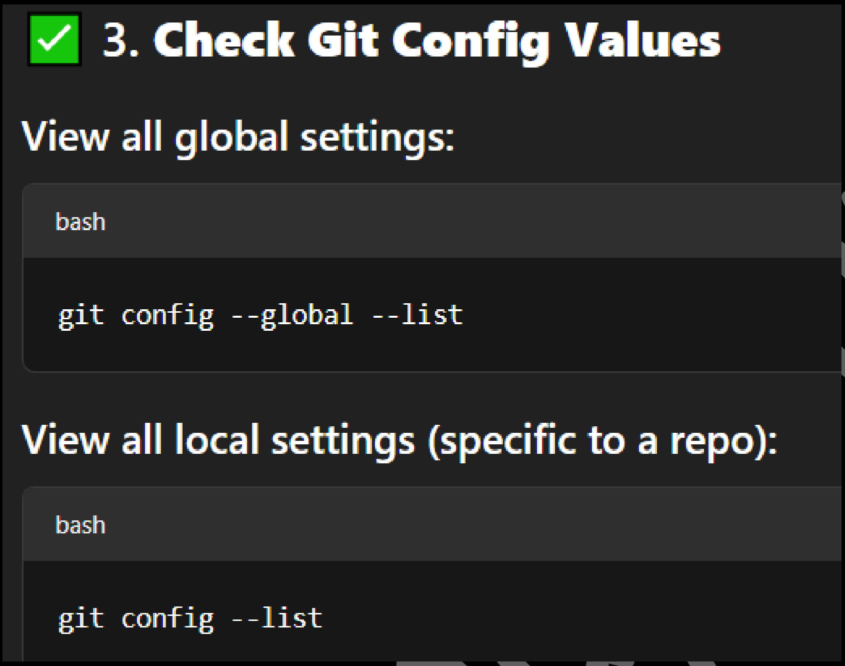
✅ 1. Set Up Your Identity (Required Before First Commit)

Set Your Name

```bash
git config --global user.name "Tanmay Samanta"
```

Set Your Email

```bash
git config --global user.email "tanmoy587d@gmail.com"
```



✅ 3. Check Git Config Values

View all global settings:

```bash
git config --global --list
```

View all local settings (specific to a repo):

```bash
git config --list
```

```
$ git init myproject
$ cd myproject
```

This creates a directory that can contain the project files as will as control files that store the historical elements, the history snapshots of yours documents, images, source code if you're working with program.

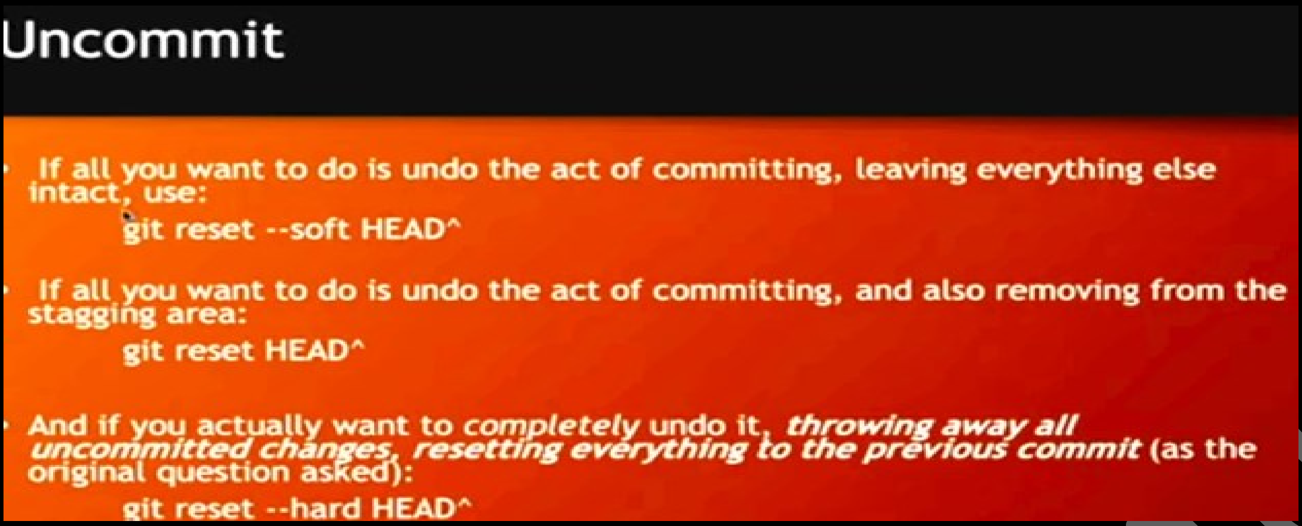
```
$ git add
```

This commend actually notice the files and puts them into a kind of holding zone, ready to committed.

```
$ git commit -m"Importing all the code"
```

Using this we permanently records a historical version or snapshot of the files as they exist at a given point in time.

Tanmay Samanta



Uncommit

- If all you want to do is undo the act of committing, leaving everything else intact, use:

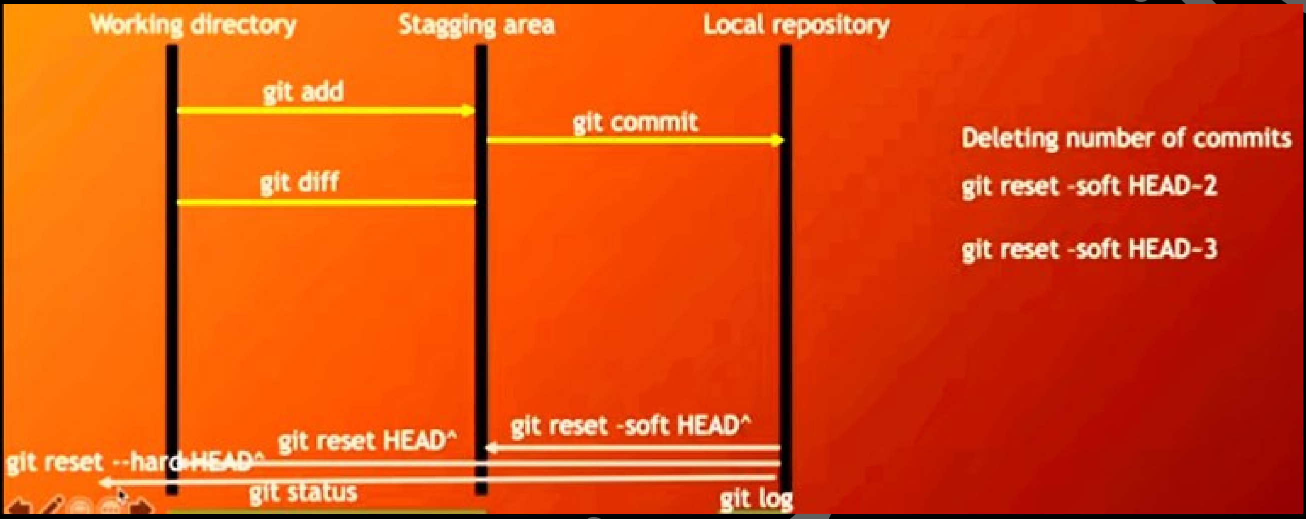
```
git reset --soft HEAD^
```

- If all you want to do is undo the act of committing, and also removing from the stagging area:

```
git reset HEAD^
```

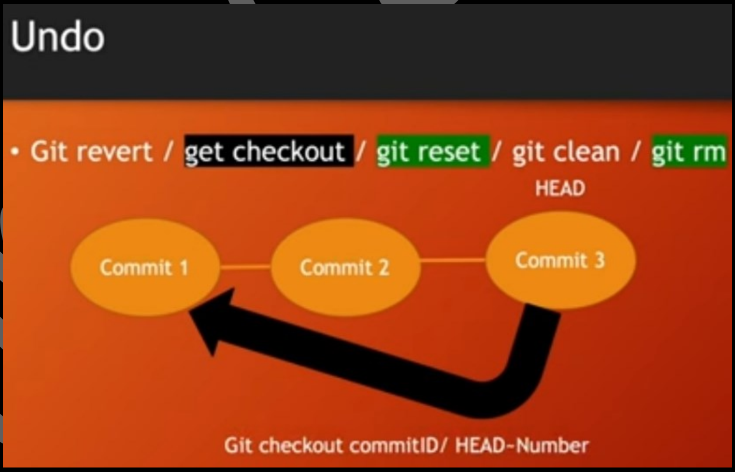
- And if you actually want to *completely* undo it, ***throwing away all uncommitted changes, resetting everything to the previous commit*** (as the original question asked):

```
git reset --hard HEAD^
```



Deleting number of commits

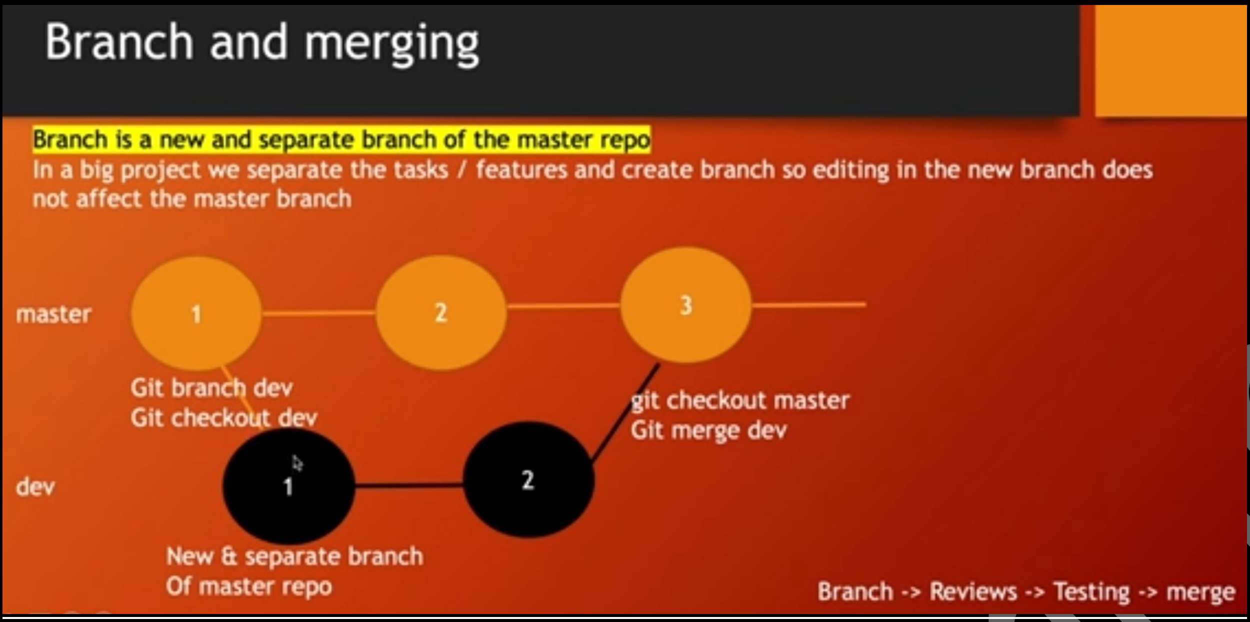
```
git reset -soft HEAD-2

git reset -soft HEAD-3
```



Undo

- Git revert / get checkout / git reset / git clean / git rm

Merging:



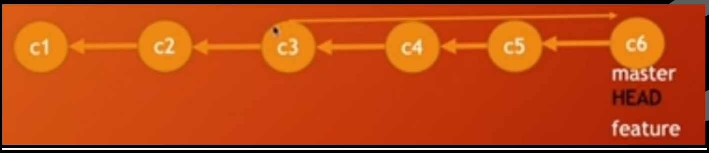
Branch and merging

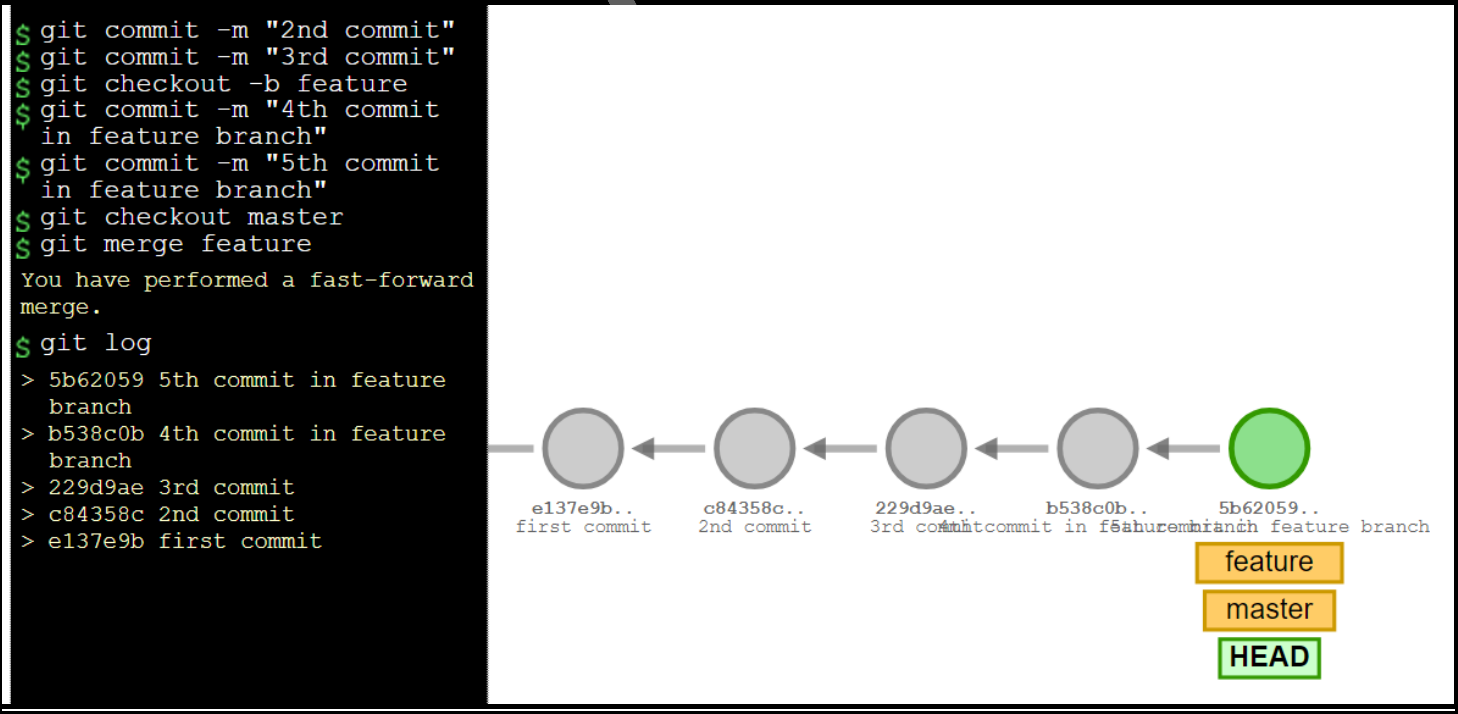
Branch is a new and separate branch of the master repo
In a big project we separate the tasks / features and create branch so editing in the new branch does not affect the master branch

Branch -> Reviews -> Testing -> merge

[Visualizing Git]()

2-way Merging Forward Merging:

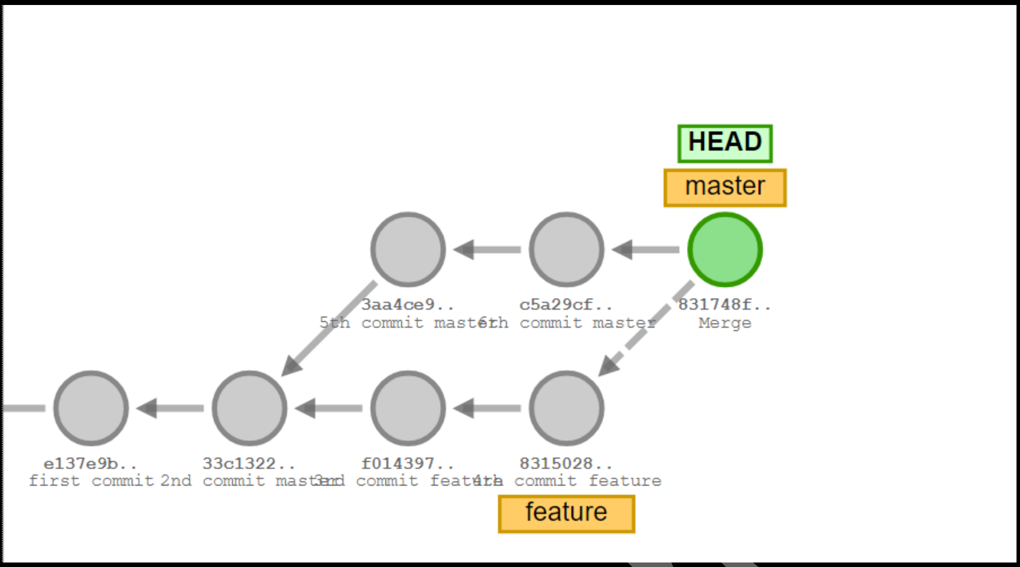




```
$ git commit -m "2nd commit"
$ git commit -m "3rd commit"
$ git checkout -b feature
$ git commit -m "4th commit
  in feature branch"
$ git commit -m "5th commit
  in feature branch"
$ git checkout master
$ git merge feature
You have performed a fast-forward
merge.
$ git log
> 5b62059 5th commit in feature
  branch
> b538c0b 4th commit in feature
  branch
> 229d9ae 3rd commit
> c84358c 2nd commit
> e137e9b first commit
```

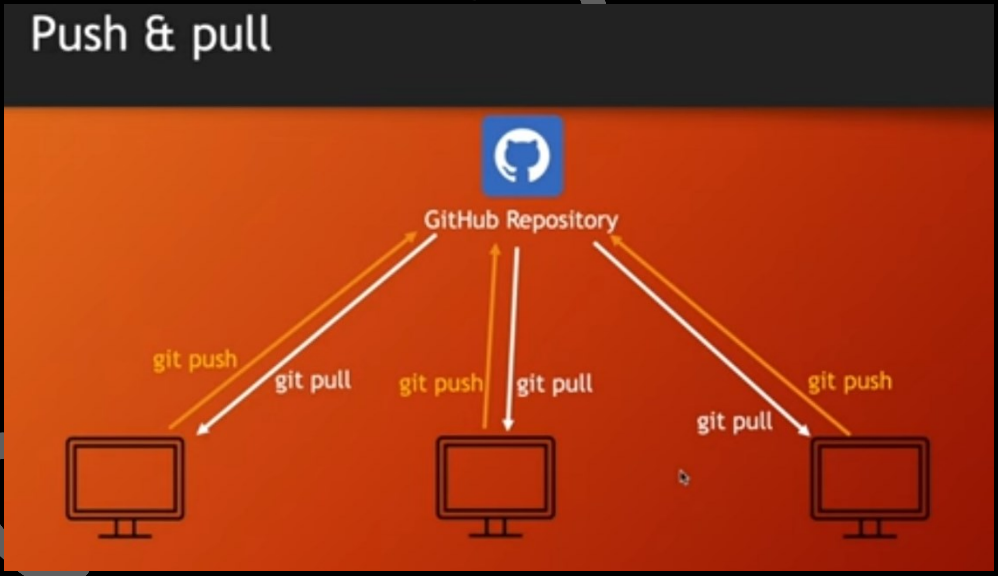
3-way Merging:

```
$ git commit -m "2nd commit
  master"
$ git checkout -b feature
$ git commit -m "3rd commit
  feature"
$ git commit -m "4th commit
  feature"
$ git checkout master
$ git commit -m "5th commit
  master"
$ git commit -m "6th commit
  master"
$ git merge feature
$ git log
 > 831748f Merge
 > c5a29cf 6th commit master
 > 3aa4ce9 5th commit master
 > 8315028 4th commit feature
 > f014397 3rd commit feature
 > 33c1322 2nd commit master
 > e137e9b first commit
```

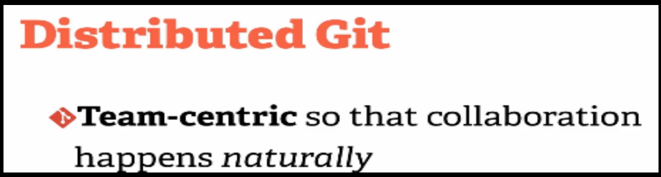


Te Saman

Pull & Push:


Push & pull

GitHub:

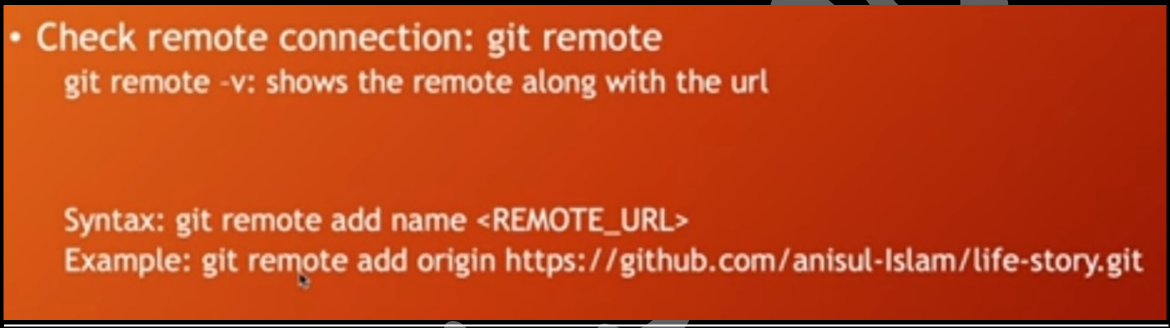


Distributed Git

- **Team-centric** so that collaboration happens *naturally*

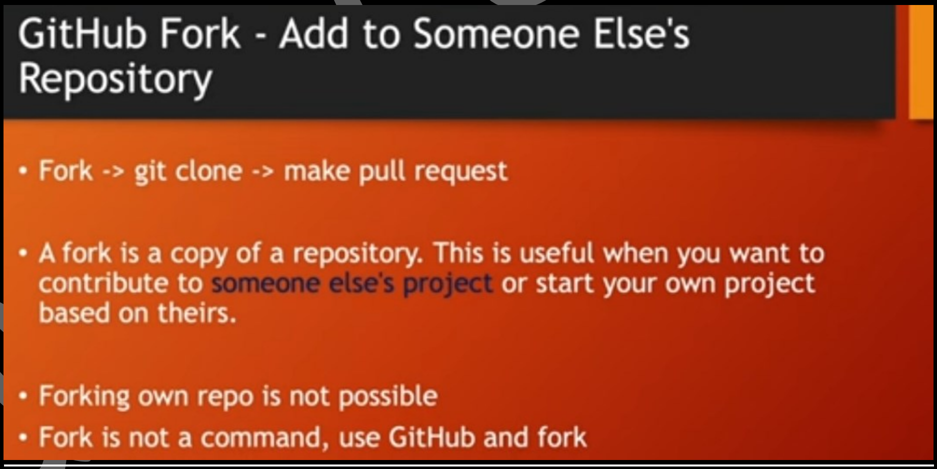
Collaborative History Tracking

Time and Versions



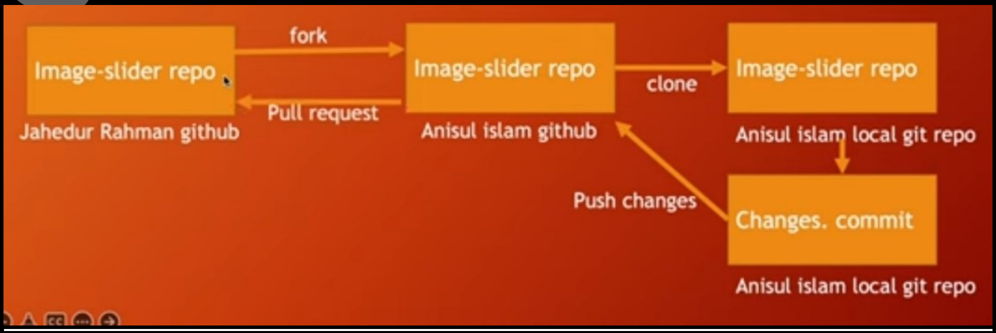
- Check remote connection: git remote
  git remote -v: shows the remote along with the url

  Syntax: git remote add name <REMOTE_URL>
  Example: git remote add origin https://github.com/anisul-Islam/life-story.git



GitHub Fork - Add to Someone Else's Repository

- Fork -> git clone -> make pull request
- A fork is a copy of a repository. This is useful when you want to contribute to someone else's project or start your own project based on theirs.
- Forking own repo is not possible
- Fork is not a command, use GitHub and fork



Tanjanta

Tanmay Samanta

Local repository & a Remote repository

The important thing to note is that you can have a local repository completely in parallel with a remote repository check the differences between them, but you can also sync them or push things from your local repository to your remote local repository completely in parallel with a remote repository.

when we performed the command git push, then that effectively pushed all of those commits, all of those various versions and changes and code pieces to our remote repository on GitHub. So that's what "git push" does.