git

+

GitHub

AVNEET KAUR

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

Github

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

https://github.com

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

Add & Commit

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

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

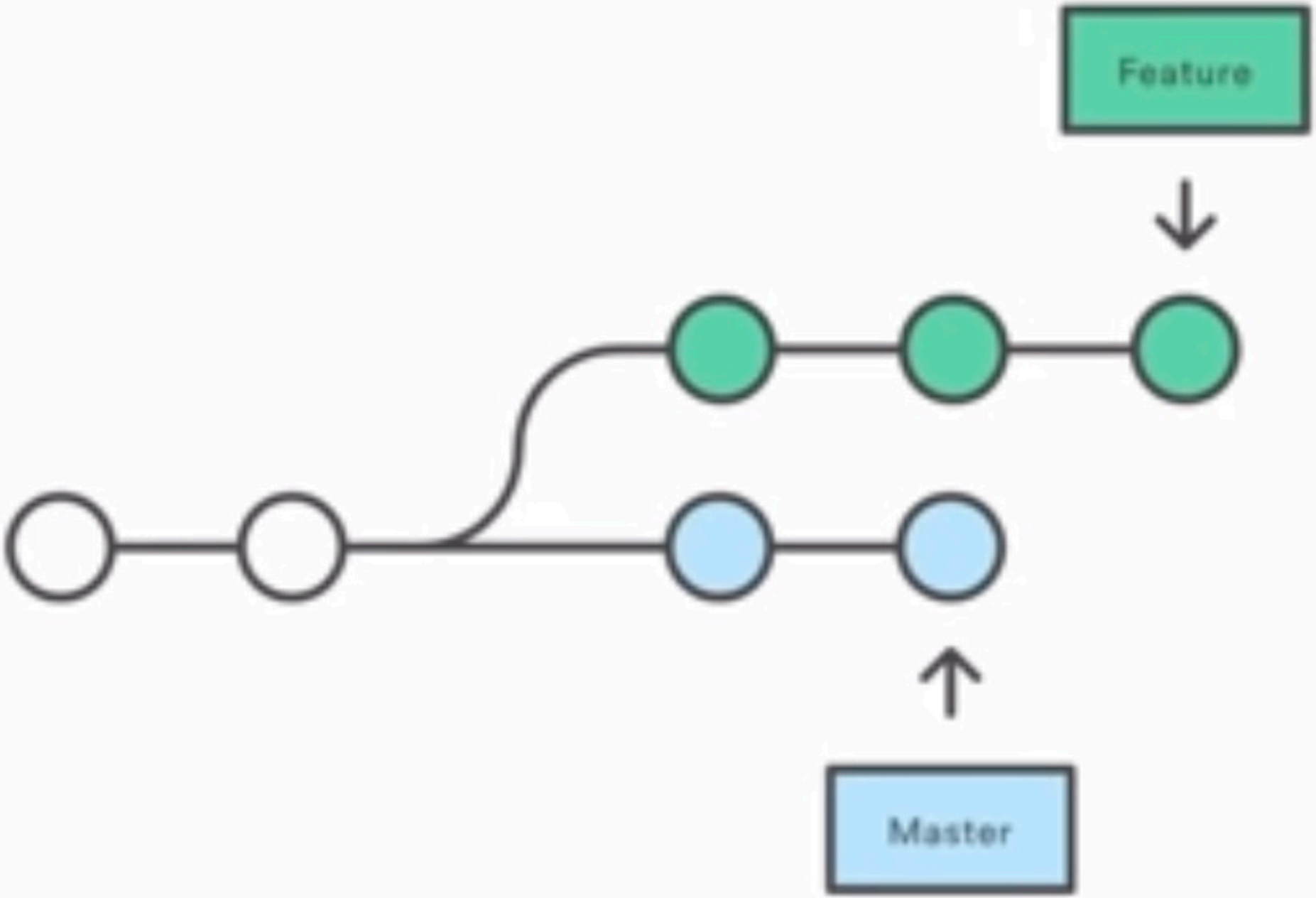
git remote add origin <- link ->

git remote -v (to verify remote)

git branch (to check branch)

git branch -M main (to rename branch)

git push origin main


Feature
Master

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->

git merge <- branch name->

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.

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 ->

git checkout	**git reset**	**git revert**
Discards the changes in the working repository.	Unstages a file and bring our changes back to the working directory.	Removes the commits from the remote repository.
Used in the local repository.	Used in local repository.	Used in the remote repository.

Does not make any changes to the commit history.	Alters the existing commit history,	Adds a new commit to the existing commit history.

```
s added to commit (use "git add" and/or "git commit -a")
loud:demo kammana$ git checkout one.sh
loud:demo kammana$ git status
 master
ch is up to date with 'origin/master'.

ot staged for commit:
it add <file>..." to update what will be committed)
it checkout -- <file>..." to discard changes in working directory)

odified:   two.sh

s added to commit (use "git add" and/or "git commit -a")
loud:demo kammana$ git checkout .
loud:demo kammana$ git status
 master
ch is up to date with 'origin/master'.
```

```
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   one.sh
        modified:   two.sh

javahomecloud:demo kammana$ git reset HEAD *
Unstaged changes after reset:
M       one.sh
M       two.sh
javahomecloud:demo kammana$ git reset HEAD one.sh
```

THANK YOU