1. nano file/folder name: Edit directly in terminal

2. git - configuration levels:
    _ system : All users + All their repositories
    _ Global : All repositories on the computer for current user
    _ Local : Only current repository

Git steps : create the file/folder -> add to git <=> (modified) -> commit -> push origin/master (to remote directory)

3. git config --global user.name "name" : create name for github global

4. git config --global user.email "email": create email for github global

5. git config user.name: check out name of github

6. git config --list : list all info of the folder

7. git config --global core.editor "link you want to set text editor in github"

8. To create and add git folder/file

    create folder in terminal: mkdir <folderName>

    create file in terminal: touch <fileName>

create github of the file: git init (in the file directory)

add the file to git: git add <fileName>

add all file : git add .

To see the hidden item in mac: command shift .

9.
git status : check the status of the directoty you are in

git status -s: check status quickly
A : already add to the directory
?? : untracked
M : modified

10. git add *.java : add all the files have .java in the name

11. git commit -m "What you did/ message" : commit the changes after you add the files to staging area

12. git status --help: get help from git

13. git commit -a -m 'what you did/ message' : Auto commit and track changes to modified file - (commit without add steps)

14. git rm --cached <fileName>: untracked the fileName

15: git rm -r --cached <folderName> : Untracked the folder

16. Create .gitignore
    touch .gitignore
    git add .gitignore

17. git log: see the history using git

    git log - <number> of history you want to see

    git log --all: list all commit history

    git log <branch name> : list all commit history of specific branch

18. git commit --amend: to change the comments/ messages of the commit

19. git restore --stage <fileName> : to unstage the file

20. git restore <fileName> : to discard changes in working directory

21. git remote -v : check weather we have any remote repository

22. git remote add origin <link Https>: add the link of remote repository to local repository.

23. git push origin -u master: push and remember all local repository to remote repository on GitHub

24. git clone <link Https> : clone all folders/files from github repository to local computer

USING BRANCH

25. git branch <name> : create new branch

26. git branch : check which branch we are using

27. git branch -r : check which branch we are using fro remote repository

28. git branch -a : check all information of branch including remote and local

29. git check out -b (branch) (name of branch): create new branch and move directly to that branch

    git switch -c (create) <branch name> : create new branch and move directly to that branch

30. git check out <branch name> : switch between branches

    git switch <branch name> : switch between branches

    git switch - : switch to master

31. git push origin -u <branch name>: push and remember

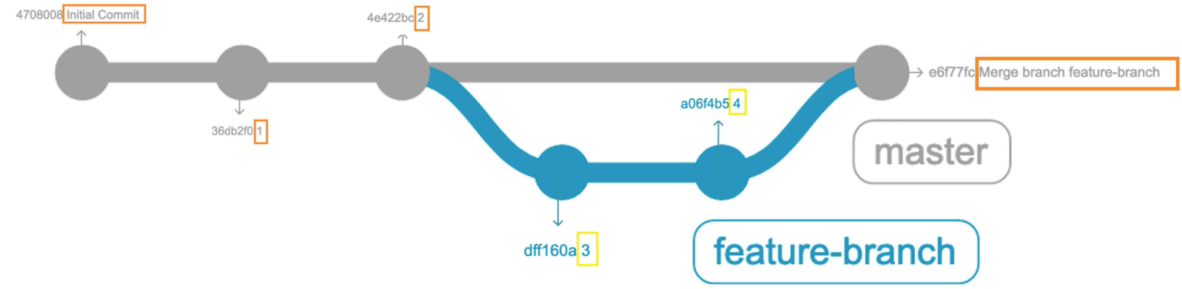
all local repository in specifics branch to remote repository on GitHub

32. git branch -d (delete) <branch name> : delete specific branch locally

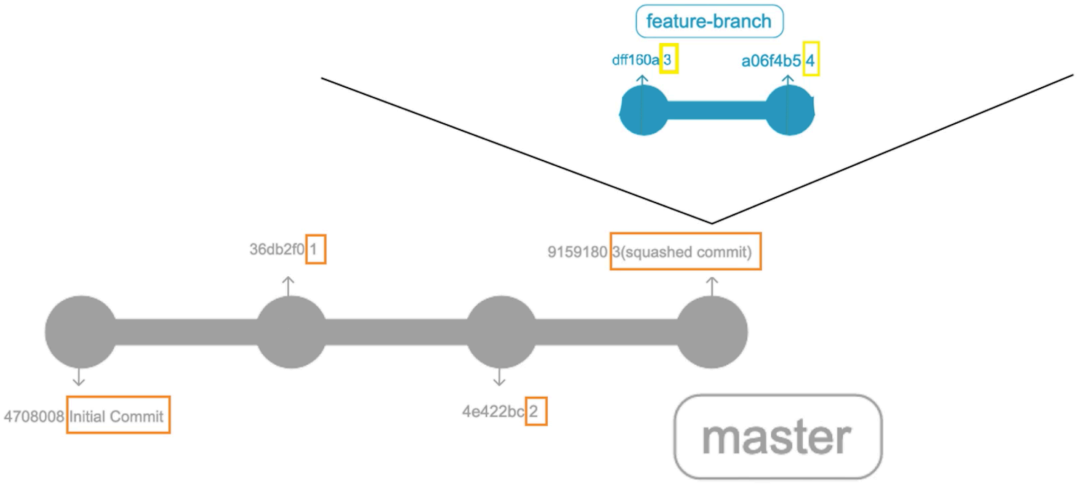
33. git push origin --delete <branch name> : delete specific remote branch

PULL OPTIONS From another branch to master

Merge commit



Squash and Merge



Rebase and Merge

Rebase and Merge

. LifeCycle of git:

Untracked -> (add the file) -> Unmodified

<- (Remove the file) <-

Unmodified -> (Edit the file) -> Modified

Modified -> (Stage the file) -> Staged
Staged -> (Commit) Modified / Unmodified

.vi editor :
press i key for insert/editing
press esc key for get out of insert mode
write :wq for write and quit