

## configuration

```
git config --global user.name "Hossam"
```

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

to check the user name/email

```
git config user.name
```

```
git config user.email
```

## Terminal crash Course

ls → list in the folder      ls -a → hidden files

pwd → by default e/path ely enta Lh now

start . → by default el folder ely enta Lh

rm -rf Foldername

git log --abbrev-commit → by default el commit SAH

git log --oneline → shakl mustasr w view Kwafy

law nset haga w mazmlth hash add w gyz thothe Lh

Commit ely lab Zavatel

git add elhaga

git commit --amend

law 3ayz files mo3ayna mayt3mlhash track  
Zay maslan files extension betwo .log

- ① hat3ml file esmo .gitignore
- ② ~~\*~~.log → kol el files dg extension .log msh  
hayt3mlhom track

## Branching

viewing branches

git branch

creating branch

git branch name

to switch branch to another one

git switch branchname

abbreviation to make add & commit together in one command

git commit -a -m "message"


Another way to switching

git checkout branchname

create & switch

git switch -c branchname

## Deleting

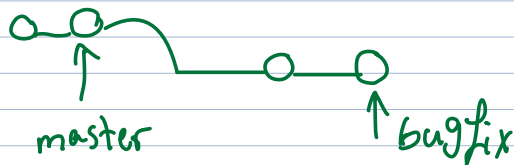
git branch  branchname  
force

## Renaming

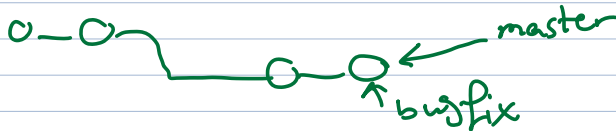
git branch -m newname

## Merging

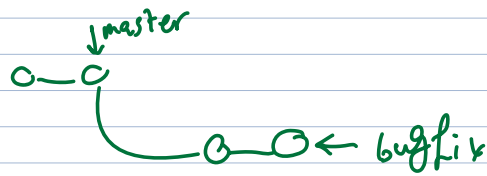
Last forward: linear path form



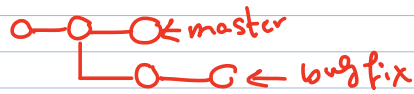
başlı lama tuznel merge (combine)



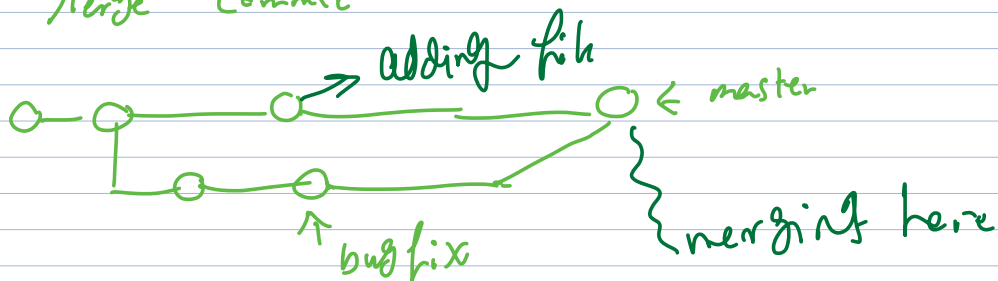
Not all merges are fast forwards!



what if we add commit to master branch? 🤔



Merge Commit



Resolve conflict merge

Some conflicts you should resolve it manually

Git Diff → to see changes

Viewing unstaged changes

mat3mlsh add, ba3d kda git diff  
haywarelak e/ differences

staged & unstaged changes

git diff Head

staged only

git diff --staged

git diff --cached

for specific file

git diff --staged Filename

Comparing branches

git diff branch1..branch2

Comparing Commits

git diff Commit1..Commit2

## stashing

git stash → to stash files  
/ git stash save

git stash pop → Zasham tera93 Kcl el hajat  
el9 Le1 stash

git stash apply → law hatlas el stash Ly aktar  
mn branch

to see your stashes

git stash list

- - - - -

git stash apply stash@{n}

git stash drop stash@{ } → delete stash

to delete all stashes

git stash clear

to go back for specific commit

git checkout hash

git checkout HEAD~1,2,3,4,---

git restore --source HEAD~1 filename

to discard changes

git checkout -- filename

git checkout head filename

} Same

To restore changes

git restore filename

git switch - → hayragsk del branch efk abl  
ma te3mel detached head 3da bel



Unstage files with restore

git restore --staged file name

undoing commits

git reset commit hash → regular reset

To delete commit

5d965A

Third commit

ABCD69

second commit

ACHIB9

first commit

if you want to delete third & second commit  
you will be back at the first commit

git reset --hard ACHIB9



law enter zamel share lcl commits dy  
ma3 nas tanya la mn el afdal enk testasdem

git revert hash commit



git remote -v → view our remotes

git remote rename old new

git remote remove name

git remote add name url

## Pushing

git push remote branch

to view remote branch

git branch -r

git fetch remote  
git fetch remote branch } download updates from  
remote (Github) to your  
local repo not in the  
working directory  
you can see it in  
origin/branch  
name

if you want to download updates into your  
working directory

git Pull remote branch

~~~~~

git rebase -i head~no → da zashan yetla?/k el  
commits wtshef hatsim leha  
ehh!

reword → edit commit message

fixup → combine commit msg commit & keep contents of both

to restore rebase

git reflog

## viewing Tags

### git Tag

git tag -l "\*beta\*" → search tags that include beta  
↓  
list  
fy hoge zabutaka  
fy hoge bezdaka

git show tagname → to see metadata of this tag

git tag -a tagname → annotated

git tag tagname → lightweight

git tag tagname commit  
↓

git tag tagname commit (-f) → to update tag  
↓  
force

to delete tag

git tag -d tagname

to push tags

git push remote (origin) tagname → specific tag  
git push origin --tags → all tags

Head Reflog → Records everything have done  
in local repo

about 90 days and will be removed

git reflog show branchname / head

git reflog → to see logs

Timed References

git reflog master@{one.week.ago}

git checkout bugfix@{2.days.ago}

unroute Plax

git diff main@{0} main@{yesterday}

Reflogs Rescue :-

if u delete a commit by mistake (git reset --hard hash)

git reflog show master

& get the hash for<sup>th</sup> commit that have been deleted

be3d Kda

git reset --hard commit / master@{ }