git blame <filename> → میاد نشون میده کی روی این فایل تغییرات ایجاد کرده

Touch <fileName> → creates a new file if the file didn't exist.
حالا اگر فایل موجود نبوده یه فایل رو اضافه میکنه براش.

vi <fileName> →

اگر توی یک directory که git init رو بزنیم، git میاد کل اون دیرکتوری مربوطه رو تا پایین میبینه.

git status → اون فایل هایی که تازه اومدن و تغییر کردن، status نشون میده

git commit → a new window would open. in order to write comments.

git log

git config --global user.name "___"
user.email "___"

git commit -m "comment"

git add → adds a file to your staging area.

git add . → adds all files in the directory to our git directory.

git checkout <the commit hash> → goes to the checkedout commit.

git checkout master → comes to the current situation.

git checkout HEAD^ → goes one step back from current commit.

gitg → a graphic that shows our git and the changes.

git branch <branch name> → creates a new branch

git checkout <branch name> → goes to another branch.

✓ git merge <branch name> → merges the <branch_name> branch with current branch.

✓ when we are done with one branch we delete it:

git branch -d <branch name>.

✓ we can not change our commits. → Exception: we can only change our last commit → some times creating a new commit for a little change is not logical so we change the last commit.

✓ git commit --amend → modifies the last commit & updates the changes on it.

✓ what's the difference between git & github? (gitlab?)

- git → local
- github → a version control on a server. ↑ Public
- gitlab → Private + Public

✓ gitIgnore.io → it comes and ignores what you don't need to update every time (like SDKs and APKs)

✓ git add .gitignore

✓ git remote add origin <SSH>

✓ git remote -v → Lists the remotes.

✓ we can apply labels for even new comers → pull issue label.

✓ git push origin master → go to origin and push on master branch.

✓ cloning → our own project → SSH
not our project → HTTPS

✓ git clone (HTTPS)

✓ we can fork sb's repository → all changes until you are going to fork would be included into your repository → you make a copy of the repository with all the changes till now & then you start making your own changes.

✓ git remote show origin.

✓ if we click on "review changes" then we can set comment for each line that has changed in new commit.

✓ Revert : Branch : master

we can revert to the condition before pull request & merging and we can revert the changes on a new branch.

✓ Squash & merge → we have 4 commits on 1 branch. → Squash 4 commits to 1.
→ (merges all 4 commits to 1) → creates & combines to 1 new commit → viewer does this.

✓ Rebase & merge → search yourself (up to you)

✓ poll = ~~fetch~~ fetch + merge

✓ deploy = Test the project + merge