Git config

Before start using git you have to config your git, your name, your email, your default editor and your endline config like this:

Git config –global user.name “vahidrobati”

Git config –global user.email “vahid.robati@gmail.com”

git config --global core.editor "code --wait"

git config --global core.autocrlf true

when you need any help about git config, type: git config --help

Git status: this command track the status of git stage and changes

Git add: add one or many files to stage(index)

Git commit -m “message”: send stage version of any file to repository and update it and tag it with a specific command

If you 100% certain about your changes, you can skip staging step and commit your changes directly with this command: git commit -a -m “message”

Remove file from repository

First delete it with this command: del file2.txt

Then add it to stage like this: git add file2.txt

And finally send stage to repository.

To see the list of staged state file, use this command: git ls-files

Rename the file from repository

First rename the file with this command: git mv file1.txt index.txt

And commit the result.

Git Ignore

To protect some files from git changes and stages, they should be placed in gitignore file. To do that, first need to create a file without name and with gitignore extension in the root of the project.

After that with this command, define the file or directory git change protected:

Echo logs/ > gitignore

After this command, all the files which are in /log folder, keep protected from git change.

Important notice: if one time we add and commit some file or folder, we can not keep that file protected unless delete this manually from git monitor like this:

Git rm –cached -r log/

Git diff

With diff command can see changes in content of files like this:

git diff –staged : show the staged differences which wait to commit

git diff : show the non-staged changes which wait to add

but we can use visually the changes. There are different tools to show changes graphically. But we can use vscode with these two commands:

git config --global diff.tool vscode

git config --global difftool.vscode.cmd "code --wait --diff $LOCAL $REMOTE"

After that we can view workspace and stage changes with these commands:

git difftool

git difftool –staged