Github - un loc in cloud unde ne putem urca codul public sau privat - ajuta angajatorii sa vada ce am lucrat

Git - tool de versionare, care ne ajuta sa ne organizam si sa uploadam codul pe github

Repository (repo) - Spatiu local sau in cloud unde ne tinem proiectul

Commit local - cand efectuam modificari/adaugari de fisiere => local, vom avea o versiune mai nou a codului fata de cea din cloud. Noile modificari le impachetam intr-un asa numit ‘commit’

Branch - o ramura de cod in care lucram care contine acel commit. Pentru inceput vom lucra doar in ramura ‘main’

(in programare poti lucra pe 2 taskuri deodata. Branch1 -> commit1, 2; Branch2 ->commit 1,2)

Push - urcam codul in repository-ul nostru public din github

Clone - cand clonam pt. Prima data un proiect din github la noi local

Pull - cand aducem versiunea mai noua din github la noi pe pc local

(ex: un coleg poate sa lucreze in acelasi repo, astfel versiunea noastra ramane in urma. Facem un git pull sa o aducem la zi)

1. Create github account: https://github.com/

2. Download git: https://git-scm.com/downloads

3. From github -> Repositories -> New (public, NU BIFAM NIMIC!)

4. Pycharm: Click dr pe numele proiectului -> open in explorer

5. Explorer in folderul root(principal) din proiect -> click dr -> open git bash here

6. touch README.txt (ne cream un readme file)

7. git init (initializam git)

Daca va incurcati pe parcurs si vreti sa incepeti de la inceput, stergeti folderul .git si reluati de la 7 in jos.

8. git add . (adaugam fisierele la commit)

9. git status (verificam ca fisierele s-au adaugat cu verde)

10. git commit -m "first commit" (generam primul commit)

Atentie! La acest pas s-ar putea sa va ceara sa va setati email si user name

Folositi pe rand aceste 2 comenzi:

git config --global user.email "email\_tau"

git config --global user.name "user\_preferat"

Mai folosim o data comanda:

git commit -m "first commit"

11. git status (verificam pe ce branch suntem)

12. git branch -M main (ne mutam pe branchul main)

13. git branch (verificam ca suntem pe main)

14. git remote add origin https://github.com/Andy-ITFactory/Test1.git

facem legatura cu github

Voi veti pune url vostru in loc de ‘https://github.com/Andy-ITFactory/Test1.git’

Aceasta legatura se face o singura data

15. git push -u origin main (urcam codul pe net)

Doar prima data s-ar putea sa va ceara git-ul sa va autentificati

Selectati Autentificare din browser

Puneti parola dvs si dupa submit veti vedea in gitbash un mesaj de succes

Voila! Puteti da refresh in github si veti vedea codul vostru

De aici in jos, repetati aceste comenzi de fiecare data cand adaugati/editati fisiere si doriti sa le incarcati pe net

16. editam noi fisiere pe care vrem sa le adaugam

17. git status (ne apar cu rosu)

18. git add . (se adauga)

19. git status (apar cu verde)

20. git commit -m "second commit"

21. git push (se urca modificarile)

How to create gitignore and add files/folder to ignore

1. mergeti in folderu de proiecte

2. click dr -> open git bash here

3. touch .gitignore (se genereaza fisierul)

4. go to gitignore.io and generate a python specific gitignore

5. copy paste content

6. git rm --cached filetoignore.txt (daca era deja added si vrei sa il ignori)

7. in gitignore adaugi fisiere/foldere pe care vrei sa le ignori (ca in video)

Pentru a importa un proiect la tine local

1. mergeti in folderu de proiecte

2. click dr -> open git bash here

3. git clone <https://github.com/Andy-ITFactory/Demo.git>

Voi veti pune linkul vostru sau orice doriti sa importati de pe net

4. Din editorul de cod puteti da open Project

5. deschidem git bash in project folder

6. dupa ce modificati ceva: git add .

7. git commit -m "ce mesaj vreti voi"

8. git push