

# KeyTraining

## Git commands

### Initializing a repository

`git init` //Initializeaza un git repository

### Staging files

`git add file1.txt` // adauga din working directory in staging area

`git add file1.txt file2.txt` // adauga din working directory in staging area mai multe fisiere

`git add *.txt` // adauga din working directory in staging area toate fisierele care au extensia respectiva

`git add .` // adauga din working directory in staging area tot

### View the status

`git status` //verifica full status

`git status -s` //verifica short status

### Committing the staged files

`git commit -m "Mesaj"` //muta toate fisierele din staging area in repository

### Removing files

`git rm file1.txt` //sterge din working directory si staging area

`git rm --cached file1.txt` //sterge doar din staging area dar il pastreaza in working directory

### Renaming/moving files

`git mv file1.java file1.txt`

### Viewing the staged/unstaged changes

`git diff` // arata modificarile care sunt in working directory (unstaged)

`git diff --staged` // arata modificarile care sunt in staging area (staged)

`git diff --cached` // arata modificarile care sunt in staging area

### Viewing the history logs

`git log` // full

`git log --oneline` // summary scurt

`git log --reverse` // listeaza comiturile de la cel mai vechi la cel mai nou

### **Viewing a commit**

git show <commit\_nr> // arata detaliile commitului  
git show HEAD // arata ultimul commit  
git show HEAD~2 // arata 2 pasi in spatele ultimului commit

### **Viewing the history of a file**

git log file.txt // arata comiturile care au afectat file.txt  
git log --stat file.txt // arata numarul de modificari pentru file.txt  
git log --patch file.txt // arata modificarile facute pentru file.txt

### **Finding the author of lines**

git blame file.txt // arata cine este autorul fiecărei linii din file.txt

### **Tagging**

git tag v1.0 // pune tag ultimului commit ca si v1.0  
git tag v1.0 <commit nr> // pune tag unui commit anterior  
git tag // listeaza toate tagurile  
git tag -d v1.0 // sterge tagul dat

### **Managing branches**

git branch exemplu // creaza un branch nou care se cheama "exemplu"  
git checkout exemplu // trece pe branch-ul exemplu  
git switch exemplu // la fel ca cel de sus  
git switch -C exemplu // creaza si trece pe branchul exemplu  
git branch -d exemplu // sterge branchul exemplu

### **Comparing branches**

git log master..exemplu // listeaza comiturile din exemplu care nu sunt in master  
git diff master..exemplu // arata diferentele intre ele

**Stashing** (*temporarily shelves (or stashes) changes you've made to your working copy so you can work on something else, and then come back and re-apply*)

git stash push -m "exemplu de mesaj" // creaza un stash nou  
git stash list // listeaza toate stashurile  
git stash apply 1 // aplica stashul respectiv in working directory  
git stash drop 1 // sterge respectivul stash  
git stash clear // sterge toate stashurile

### **Merging**

git merge exemplu // adauga schimbarile din branchul exemplu in master  
git merge --abort // abandoneaza adaugarea schimbarilor

### **Viewing the merged branches**

git branch --merged // arata branchurile care au fost merged  
git branch --no-merged // arata branchurile care nu au fost inca merged

### **Cherry picking**

git cherry-pick commit\_nr // aplica respectivul commit pe branchul curent

## **Cloning a repository**

`git clone url`

## **Syncing with remotes**

`git fetch origin master` //aduce modificarile din master

`git fetch origin` //aduce toate modificarile din origin

`git fetch` // la fe ca mai sus

`git pull` // aduce modificarile si face merge

`git push origin master` //uploadeaza masterul in origin

`git push` //la fel ca mai sus

## **Sharing tags**

`git push origin v1.0` //face push tagului in origin

`git push origin --delete v1.0` //sterge tagul din origin

## **Sharing branches**

`git branch -r` //arata branchurile tracked pe remote repo

`git branch -vv` //arata branchurile tracked pe local si remote

`git push -u origin bugfix` //uploadeaza exemplu in origin

`git push -d origin bugfix` //sterge exemplu din origin

## **Managing remotes**

`git remote` //arata care sunt repositurile din remote

`git remote add origin url` //adauga un nou repo remote

`git remote rm origin`

## **Undoing commits**

`git reset --soft HEAD^` //sterge ultimul commit , si tine schimbarile din staging

`git reset --mixed HEAD^` //scoate si schimbarile din staging

`git reset --hard HEAD^` //sterge schimbarile locale