

Git ściągą

Etap 0 ustawienie gita w domu

```
# sets up Git with your name
git config --global user.name "<Your-Full-Name>"

# sets up Git with your email
git config --global user.email "<your-email-address>"

# makes sure that Git output is colored
git config --global color.ui auto

# displays the original state in a conflict
git config --global merge.conflictstyle diff3

git config --list

#Konfiguracja pod edytor Visual Studio Code
git config --global core.editor "code --wait"
```

Etap 1

- **git init** – w danym folderze tworzy repozytorium gita
- **git clone link_lub_sciezka_do_pliku** – tworzy kopie innego repozytorium
- **git status** – wyświetla jakie pliki zostały zmodyfikowane, jakie są śledzone, które zostały dodane do „working space” lub „staging index”
- **git log** – wyświetla informacje o dotychczasowych comitach
- **git log --stat**
- **git log -p lub -patch** – pokazuje zmiany w plikach
- **git log -w**
- **git log --oneline** –
- **git show numer_comita** – pokazuje informacje o danym commicie

Etap 2 – dodawanie commitów

git add **nazwa_pliku** – dodaje pliki do staging index

git commit – zapis z staging index do repo

git diff – pokazuje różnice między plikiem w repo a naszym aktualnie

Etap 3 – tagi, branche i merge

git tag – wyświetli wszystkie tagi w repo

git tag -a nazwa_taga – tworzy tag o podanej nazwie

git tag -d nazwa_taga – usuwa tag o danej nazwie

git tag -a nazwa_taga sha_comita – przypisuje tag do commita o podanym kluczu

git branch – wyświetla wszystkie branche w repo

git branch nazwa_brancha – tworzy branch o podanej nazwie

git branch -d nazwa_brancha – usuwa branch o danej nazwie (nie można usunąć brancha na którym się aktualnie znajdujemy, lub jeżeli branch ma unikatowe commity)

git branch -D nazwa_brancha – usuwa brancha mimo unikatowych commitów

git checkout nazwa_brancha – zmieniamy aktualny branch na branch o podanej nazwie

git merge **nazwa_brancha** – łączy ze sobą 2 branche do tego na którym jesteśmy dodaje drugi o podanej nazwie

Etap 4 – cofanie zmian i usuwanie commitów

git commit --amend – zamiast utworzyć nowego commita, dodaje zmiany do już istniejącego

git revert numer_comita – tworzy nowego comita który cofa się w zmianach do podanego wcześniej numeru

git reset [flaga] [numer_komitu lub HEAD^ lub HEAD~3]

flagi:

--mixed – cofa do wybranego comita a późniejsze dodaje do working directory

--soft – cofa do wybranego comita, późniejsze są dodawane do staging index

--hard – cofa wybranego comita i usuwa późniejsze

hint : przed usuwaniem najlepiej sobie zrobić brancha