

**github & docker**

# kim jestem?

Wiceprezes **NAJLEPSZEGO KOŁA NA POLITECHNICE** ❤️

R&D Lead - Asseco Business Solutions S.A.

Student III roku IAD

# **trudna ankiet**



**zeskanuj kod qr lub  
wejdź na stronę  
wooclap.com i wpisz  
poniższy kod**

**ZJDRLT**

***“projekt\_final\_v5\_poprawiony\_ostateczny\_kopia\_po\_korekcie\_dla\_wlazia.zip”***

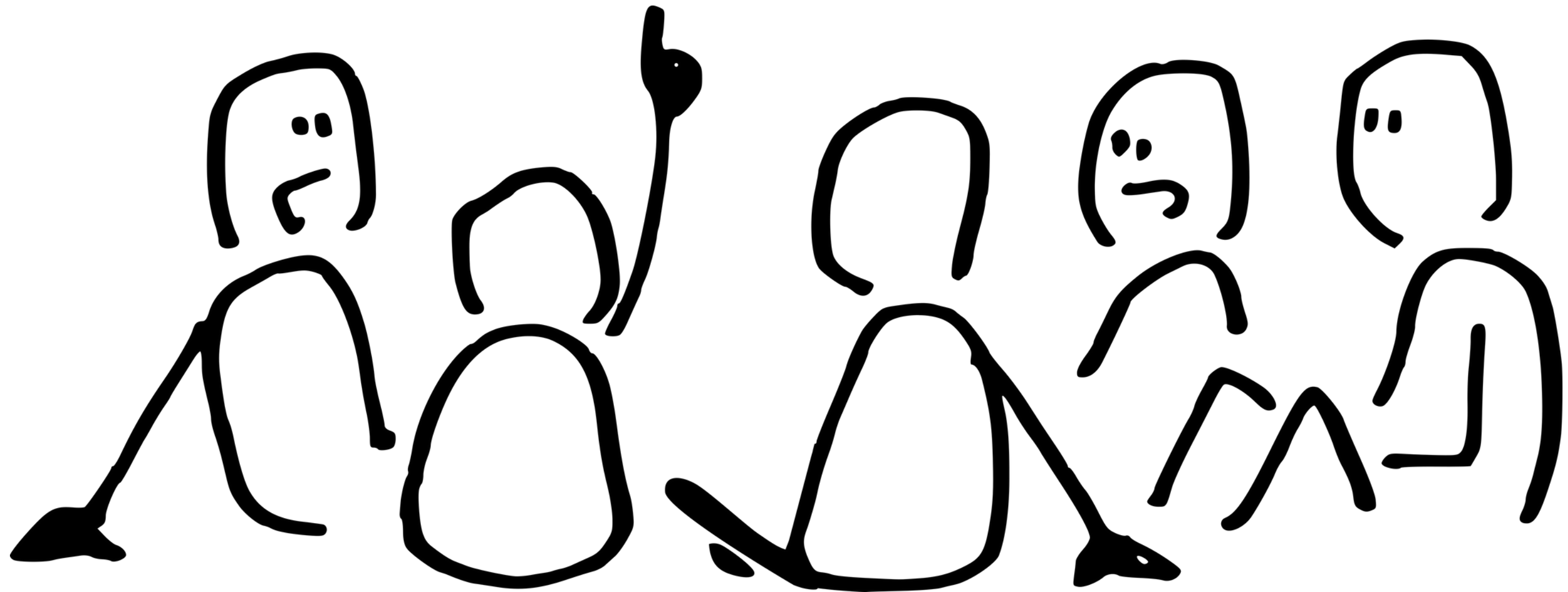
***“Co Ty zrobiłeś w tym projekcie tydzień temu?”***

***“Hey GPT how to undo deleting my entire project?”***

***“Kiedyś działało, teraz nie, jak to cofnąć do tego co działało?”***

**rozwiązanie!**

# kontrola wersji git



**git ≠ github**

# instalacja

windowsiarze: **git-scm.com**

linuxiarze (z cywilizowaną dystrybucją: ***sudo apt install git***

*macosiarze: **xcode-select --install** lub **brew install git***



# setup

***git config*** --global user.name "imie\_nazwisko"

***git config*** --global user.email "fajny.mail@wp.pl"

***git config*** --list

**zakładanie konta na githubie**

***<http://github.com/>***

**<https://github.com/settings/tokens>**

# repozytorium lokalne vs zdalne

repozytorium lokalne - *to co masz na komputerze*

repozytorium zdalne - *to co masz na githubie*

*klonujesz repo z githuba → wprowadzasz zmiany*

*lokalnie → robisz commita → robisz push*

# podstawowy cykl

## (prymitywny)

**git pull** (pobierasz to co zrobili inni)

**edytujesz kod** (robisz to co masz do zrobienia)

**git add** (dodajesz pliki który edytowałeś)

**git comit** (tworzysz commita z twoimi zmianami)

**git push** (wypychasz je do repo zdalnego)

# **komercyjny cykl**

(zaawansowany i poprawny)

**git pull** (pobierasz to co zrobili inni)

**git switch** (tworzysz własną gałąź)

**edytujesz kod** (robisz to co masz do zrobienia)

**git add** (dodajesz pliki który edytowałeś)

**git comit** (tworzysz commita z twoimi zmianami)

**git push** (wypychasz je **tylko na swój branch**)

**pull request** (to robi senior po sprawdzeniu twojego kodu)

**merge request** (to też senior)

**branche branche branche...**

**main** (nie musisz to nie ruszaj 😊 )

**dev\_0** (twoje zmiany  
w jednym sprincie )

**dev\_1** (twoje zmiany w  
kolejnym sprincie )

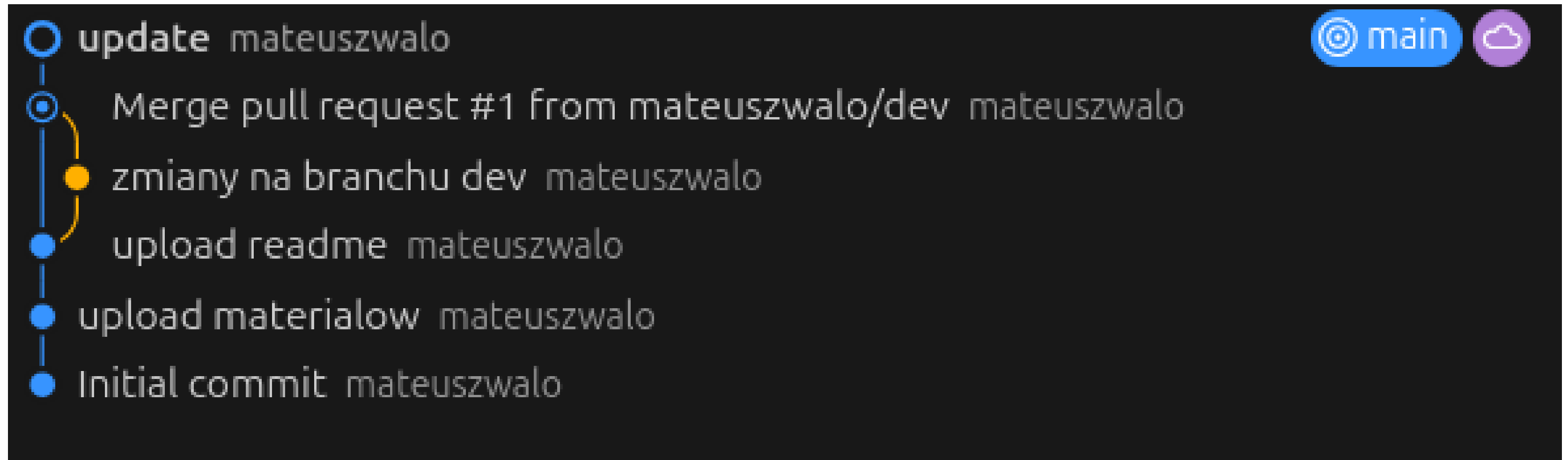
**dev\_2** (twoje zmiany w  
kolejnym sprincie )

**test** (branch gdzie  
testujesz release  
zanim dasz na prod)

**nie\_wiesz\_co\_to\_z\_a\_  
branch**



# branche branche branche...

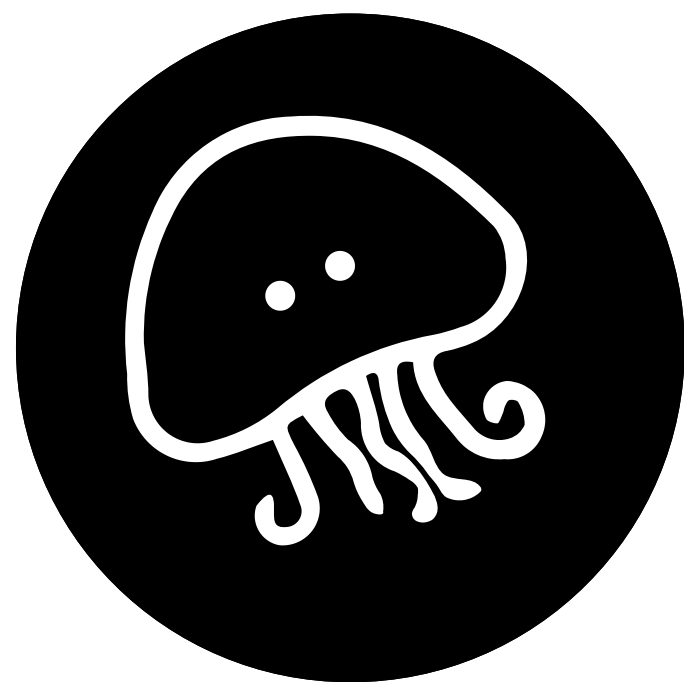


***“U mnie działa”***

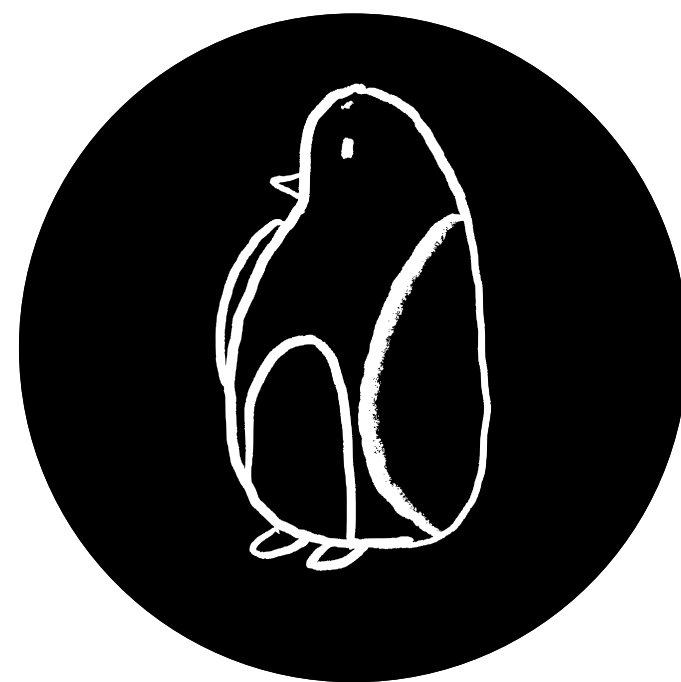
***“Ja to nie wiem jak to będzie na tym windowsie działać”***

***“zaktualizuj pythona to zadziała ?!”***

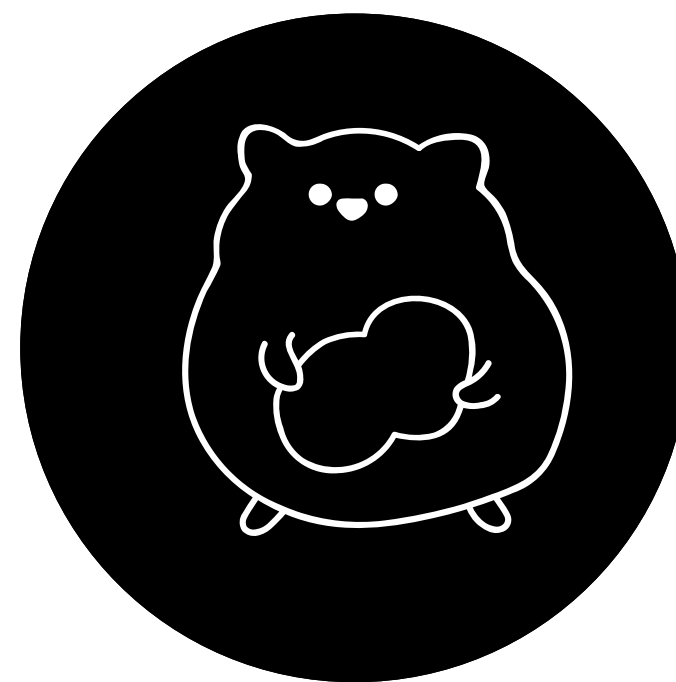
***“dlaczego na tablicy prowadzącemu działa, a mi nie?”***



windowsiarz



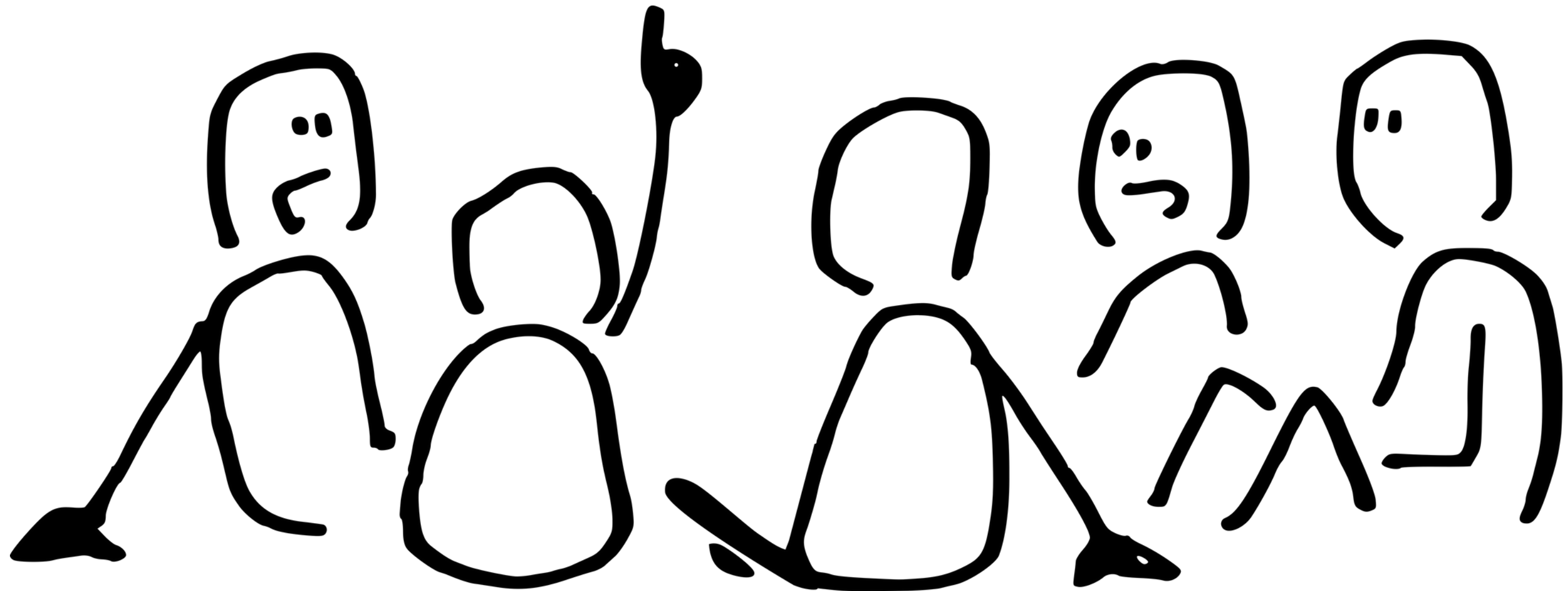
linuxiarz



macosiarz

**rozwiązanie!**

# kontenery dockerowe



# instalacja

windowsiarze: **pobieracie i setupujecie Docker Desktop i WSL**

linuxiarze: ***sudo apt-get install docker.io***

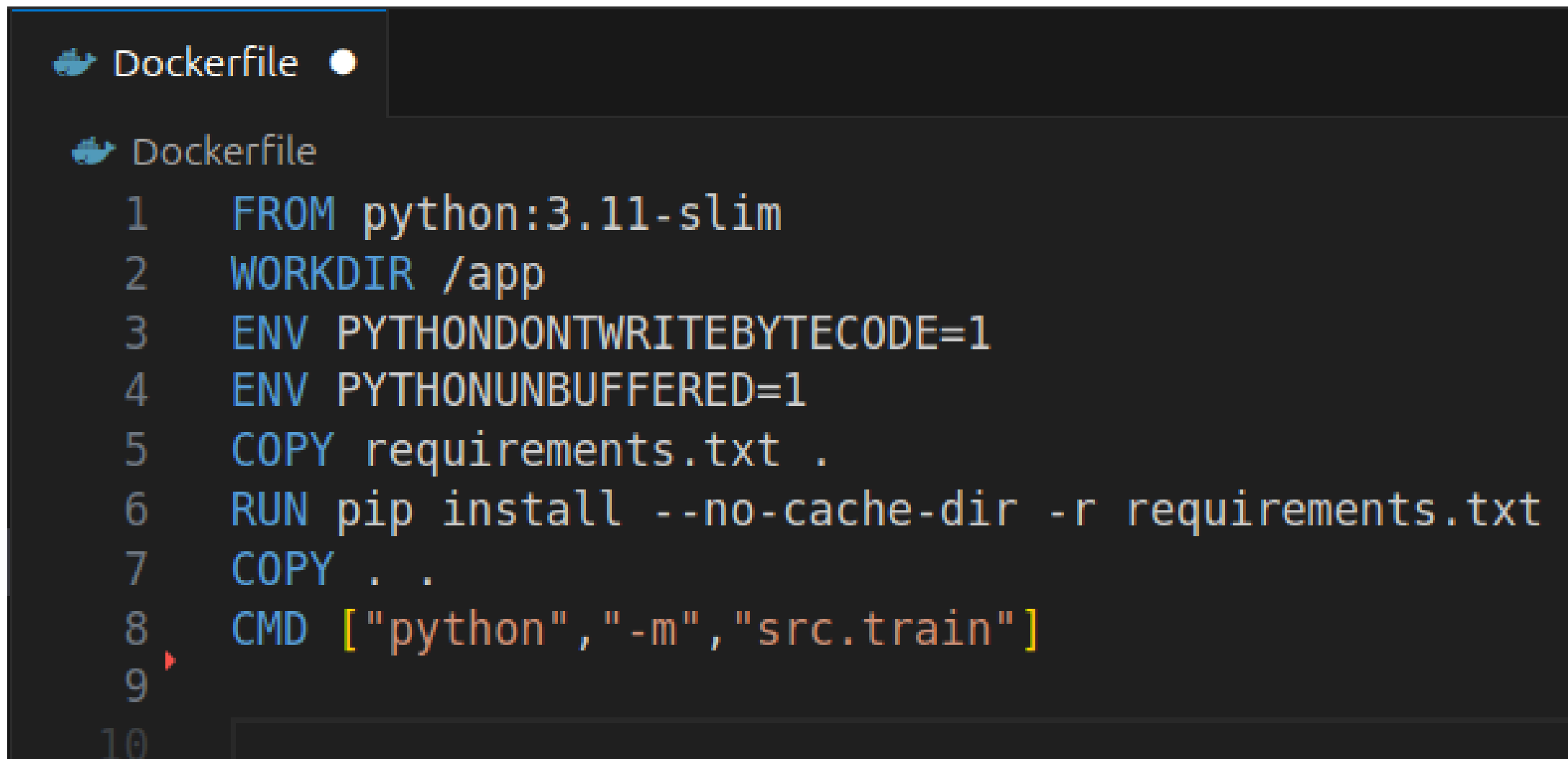
macosiarze: ***<https://docs.docker.com/desktop/setup/install/mac-install/>***

**test**

**docker --version**

**(najlepiej w wcześniej zsetupowanym terminalu bashowym)**

# pierwsze podejścia do dockera



The image shows a code editor window with a tab labeled 'Dockerfile'. The editor contains a Dockerfile with the following content:

```
Dockerfile
1  FROM python:3.11-slim
2  WORKDIR /app
3  ENV PYTHONDONTWRITEBYTECODE=1
4  ENV PYTHONUNBUFFERED=1
5  COPY requirements.txt .
6  RUN pip install --no-cache-dir -r requirements.txt
7  COPY . .
8  CMD ["python", "-m", "src.train"]
9
10
```



# drugie podejścia do dockera

 *docker-compose.yml* X

 *docker-compose.yml*

```
1  services:
2    train:
3      build: .
4      command: python -m src.train
5      volumes:
6        - ./artifacts:/app/artifacts
7      tty: true
8      stdin_open: true
9
```

# podstawowe komendy dockerowe

***docker compose up*** (*uruchamia kontener z compose*)

***docker compose down*** (*zamyka kontener z compose*)

***docker ps*** (*lista aktywnych kontenerów*)

***docker rm <id\_kontenera>*** (*usuwa kontener*)

**praktyczna praca z repozytorium**

**[https://github.com/mateuszwalobootcamp\\_github\\_docker](https://github.com/mateuszwalobootcamp_github_docker)**

**dzięki za uwagę**