

## GIT

**Git** jest rozproszonym systemem kontroli wersji **ułatwiającym pracę** nad tworzeniem aplikacji, stron internetowych i innych narzędzi w grupach. System śledzi wszystkie wykonywane zmiany w plikach, a także umożliwia przywrócenie ich dowolnej, wcześniejszej wersji. Za pomocą Gita możesz nanosić zmiany w kodzie przy jednoczesnym zachowaniu niezależności.

- 1) Instalacja GIT na komputerze: <https://github.com/git-guides/install-git>
- 2) Zakładanie konta na <https://github.com/>
- 3) Tworzymy w serwisie **github** nowe repozytorium – patrz niżej

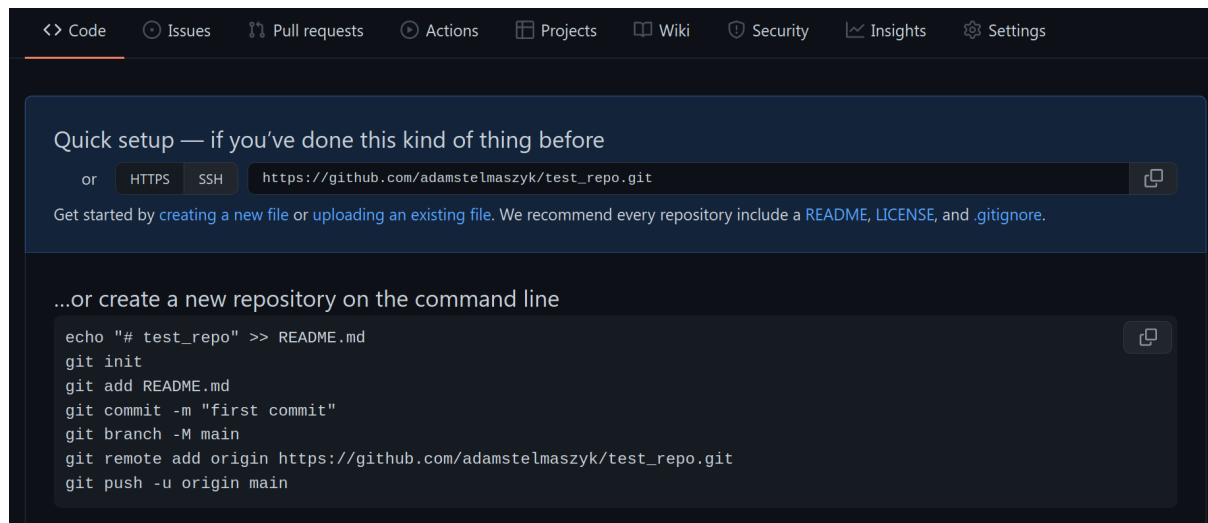
### Jak używać GIT?

Na komputerze zrób sobie folder w którym będziesz umieszczał swoje programy.

Proponuję: **PROGRAMOWANIE**

W tym folderze stworzymy za pomocą **GIT** repozytoria które jednocześnie będą znajdować się na zdalnym serwisie **github**.

**Teraz utworzymy w serwisie github nowe repozytorium! (wpisz tylko nazwę i zaznacz publiczne) – pojawi się jak poniżej:**



Po utworzeniu w serwisie **github** przechodzimy do folderu na komputerze gdzie chcemy mieć nasze repozytoria.

**Tworzymy folder którego nazwa ma być taka sama jak naszego repozytorium!**

Na rysunku jest **test\_repo** – taki folder tworzymy! **Wchodzimy do niego!!!**

Teraz będziemy wykonywać wszystkie operacje jak wyżej opisano z małą zmianą!!!

**echo "# test\_repo" >> README.md**

**//utwórz ręcznie plik .gitignore (uwaga z kropką na początku)**

**//Do pliku wpisz/wklej następujący tekst:**

**DLA języka programowania JAVA:**

W google wpisz **.gitignore java** – wchodzisz w pierwszy link i kopiujesz zawartość pliku z przeglądarki a następnie wklejasz do pliku **.gitignore**

Jeśli używasz IDE do tworzenia programów to należy w przeglądarce wpisać:

**.gitignore IntelliJ** //jeśli używasz intellij – zapobiegnie to przesyłaniu zbędnych plików na GITHUB

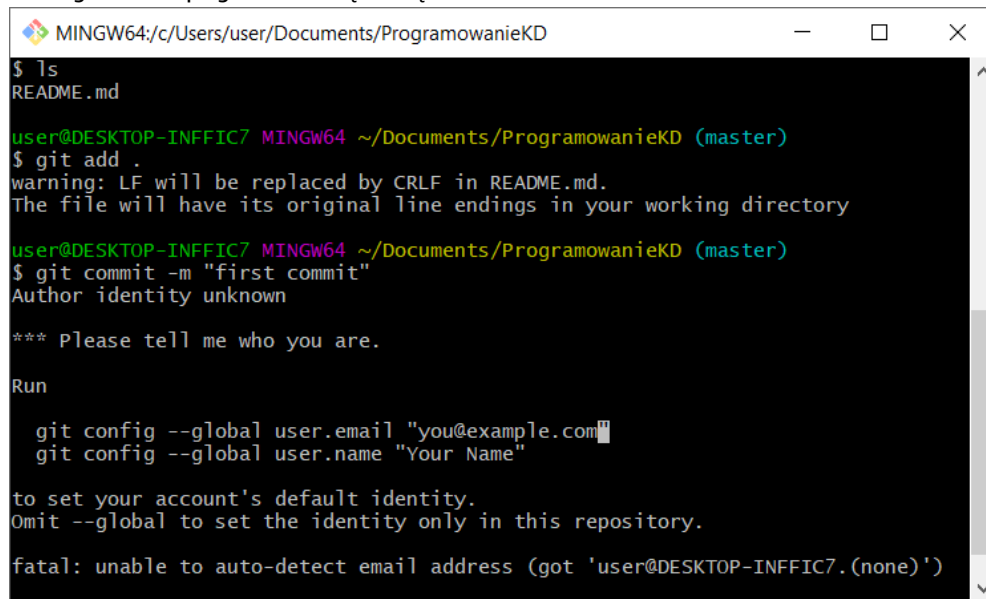
**git init**

**git add .**

**git commit -m "first commit"**

### UWAGA:

Tutaj może pojawić się błąd!



```
MINGW64:/c:/Users/user/Documents/ProgramowanieKD
$ ls
README.md

user@DESKTOP-INFFIC7 MINGW64 ~/Documents/ProgramowanieKD (master)
$ git add .
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

user@DESKTOP-INFFIC7 MINGW64 ~/Documents/ProgramowanieKD (master)
$ git commit -m "first commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'user@DESKTOP-INFFIC7.(none)')
```

Teraz musisz wpisać

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

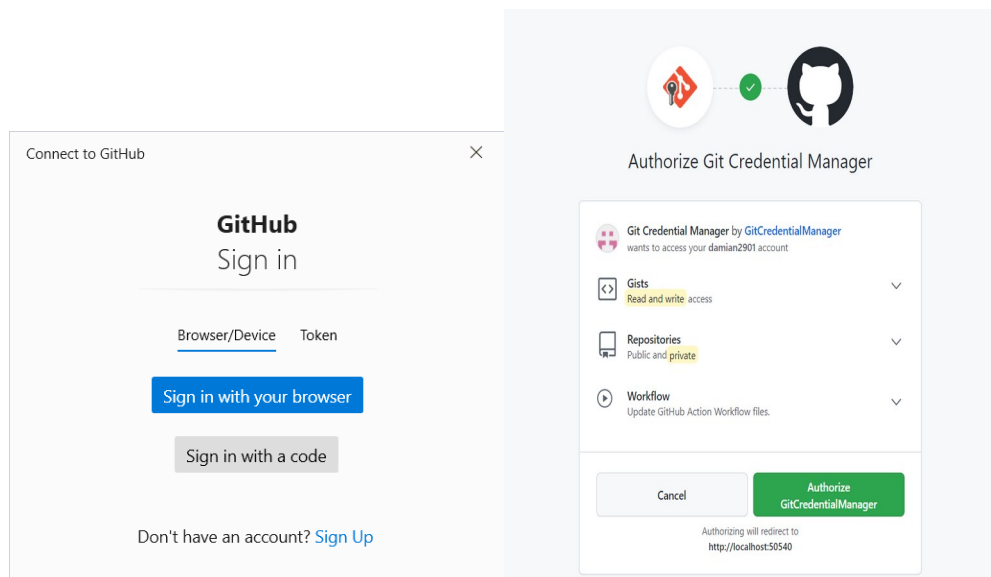
**Uzupełniając swoimi danymi logowania do serwisu Github**

**git branch -M main**

**git remote add origin https://github.com/twój\_login\_do\_githuba/twoja\_nazwa\_repo.git**

**git push -u origin main**

Po użyciu tej komendy (git push -u origin main) będziesz musiał się zalogować do serwisu github – wyskoczy Ci okienko logowania a następnie okienko autoryzacji. PATRZ NASTĘPNA STRONA.



#### UŻYWANIE:

***jakakolwiek zmiana w folderze, dodanie pliku/plików, zmiana zawartości jakiegokolwiek pliku - aby wysłać na serwis github należy wykonać następujące polecenia:***

***git add . //dodaj wszystkie pliki do repozytorium***

***git commit -m "Komentarz co zrobiłeś" //ustawienie punktu zmiany***

***git push //wysłanie do serwisu***

***Jak pracujesz w szkole to w domu musisz zrobić następującą operację jako pierwszą:***

***git pull***

***po niej możesz pracować normalnie.***

***Jeśli stworzyłeś repo w szkole a nie masz go w domu to wystraszy, że je sklonujesz (stworzy się folder o nazwie repozytorium). Przykładowo:***

***git clone https://github.com/adamstelmaszyk/demogit.git***

***Na komputerze w folderze w którym wykonałeś powyższą komendę utworzy się folder **demogit*****