

Intro GIT 1

Wojciech Barczyński

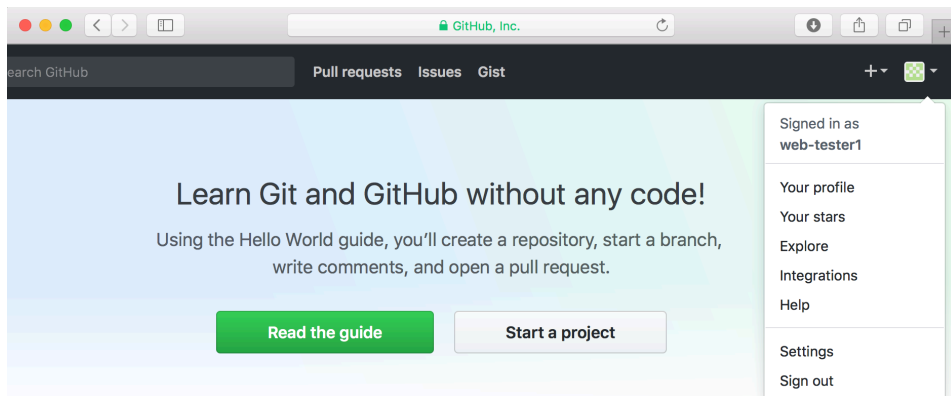
Spis treści

Spis treści	2
1. Utworzenie konta GitHub	3
2. Podstawy Gita.....	4

1. Utworzenie konta GitHub

System kontroli wersji (SCM – Source Control Management), jest podstawą każdego projektu. Możesz używać gita do zachowywania aplikacji i skryptów rozwijanych w czasie zajęć.

- Utwórz konto na githubie: <https://github.com/join>



2. Podstawy Gita

1. Utwórz folder na swoim komputerze o nazwie takiej jak Twój użytkownik git-a (którego właśnie utworzyłeś):

```
$ mkdir GIT_USERNAME  
$ cd GIT_USERNAME
```

2. Utwórz folder:

```
$ mkdir nauka_gita  
$ cd nauka_gita
```

3. Zainicjuj repozytorium git-a:

```
# sprawdź, czy jesteś we właściwej ścieżce  
$ pwd  
$ git init
```

4. Skonfiguruj git-a (zastąp *wojciech11* swoim użytkownikiem git-a):

```
# sprawdź konfigurację (małe L):  
$ git config -l  
$ git config --global user.name "wojciech11"  
  
# nie lubimy spamatorów:  
$ git config --global user.email "wojciech11@users.noreply.github.com"  
  
# zauważ:  
$ cat ~/.gitconfig
```

5. Podążając za wskazówkami wykładowcy, utwórz README.md z pomocą *atom* (packages -> Markdown Preview -> Toggle).

Dobre wprowadzenie do Markdown znajdziesz na:

<https://guides.github.com/features/mastering-markdown/>

6. Podążaj za wskazówkami wykładowcy, aby nauczyć się o komendach git działających na **lokalnym repozytorium**:

```
# jak ls w bashu  
$ git status  
$ git add README.md  
$ git status  
$ git commit -m "Moj pierwszy commit"  
$ git status  
$ git log
```

1. Zmodyfikuj README.md i umieść zmiany w repozytorium. Powtórz 3 razy.

2. Przyglądnij się, jak wygląda historia z pomocą graficznych programów gitk i gitg:

```
$ gitk  
$ gitg
```

7. Utwórz repozytorium w githubie – nauka_gita, aby umieścić pliki z naszego repozytorium na github.com:

```
$ git remote add origin https://... # tak jak na github.com  
$ git push origin master
```

Powinieneś zobaczyć swój README.md na github.

8. Zmodyfikuj README.md i umieść zmiany na github, wykorzystując:

- `git status`
- `git add`
- `git commit`
- `git log`
- `git push`

Powtórz 3 razy powyższe ćwiczenie.

Jaka jest równica, który ze sposobów, kiedy używać:

```
$ git commit  
$ git commit -m "msg"
```

9. Pobieranie zmian z github.com:

1. Upewnij się, że wszystkie lokalne zmiany są na github.com:

```
$ git status  
$ git push -u origin master
```

2. Zmodyfikuj README.md w przeglądarce.

3. Ściągnij zmiany z github.com:

```
$ cat README.md  
$ git pull  
$ cat README.md  
# korzystając ze środowiska graficznego, które bardziej lubisz:  
$ gitk  
$ gitg
```

10. Klonowanie:

```
$ cd ~  
$ mkdir GIT_USER2  
  
# url z github.com  
$ git clone https://github.com/TWOJ_GITHUB_USER/nauka_git
```

11. [Dodatkowe] Podążaj za wskazówkami wykładowcy, aby nauczyć się o komendach git działających na gałęziach (branch):

```
# jak git status dla branchy  
$ git branch  
  
$ git checkout -b ekperyment  
$ git branch  
  
# dorzuć 2-3 commity ze zmianami  
# uruchom gitg i gitk.  
  
$ git push origin ekperyment  
# wejdź na github.com  
  
# wracamy na branch master  
$ git checkout master  
$ cat README.md  
$ git checkout ekperyment  
$ cat README.md  
  
# nanieśmy zmiany z ekperyment na master  
$ git checkout master  
git merge ekperyment  
  
# uruchom gitg i gitk  
git push                                # teraz sprawdź na github.com
```

12. [Dodatkowe] Naszkicuj, jak wygląda proces wrzucania plików do git-a (zaznacz *staging* oraz *index* i *remote*)