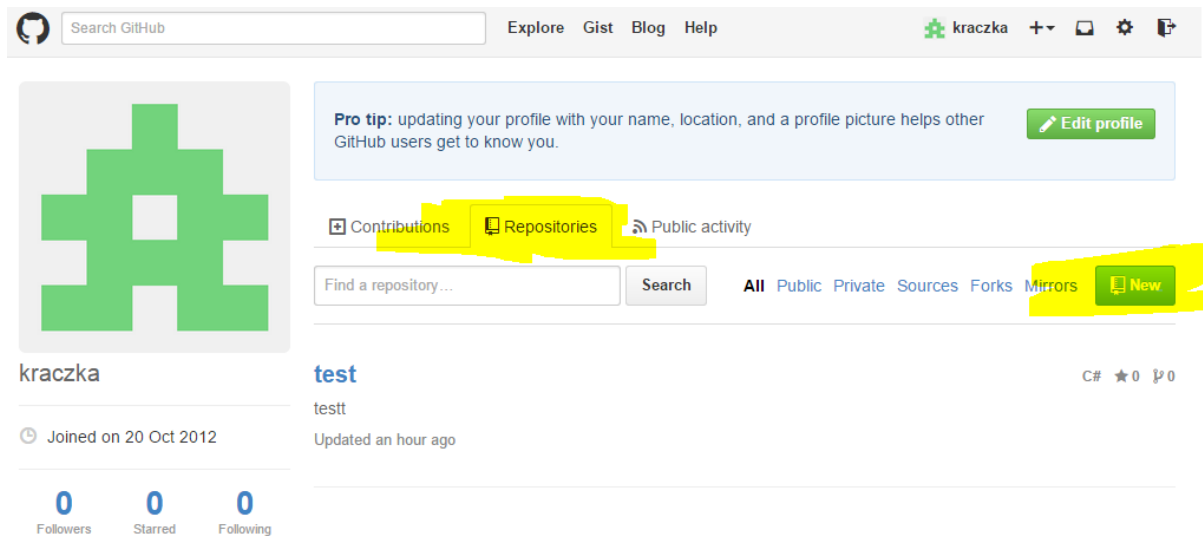


# GIT - GitHub

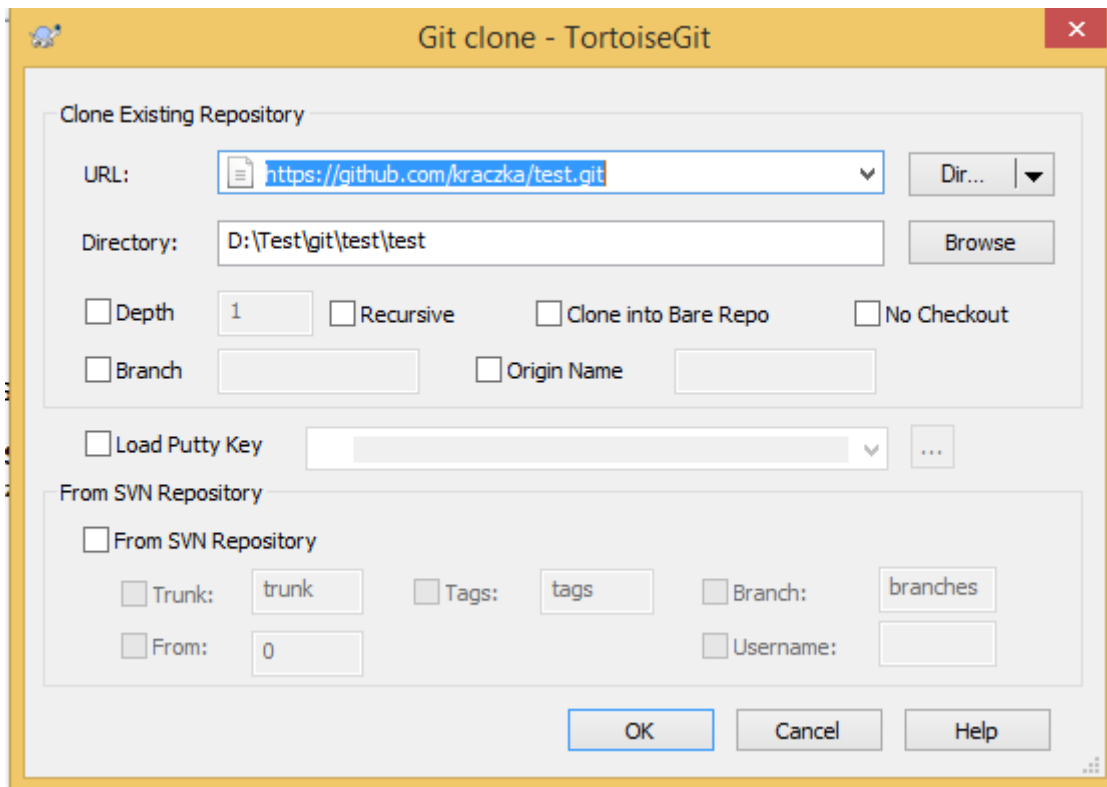
## 1. Tworzymy konto na platformie [www.github.com](https://www.github.com)

## 2. Repozytorium GIT

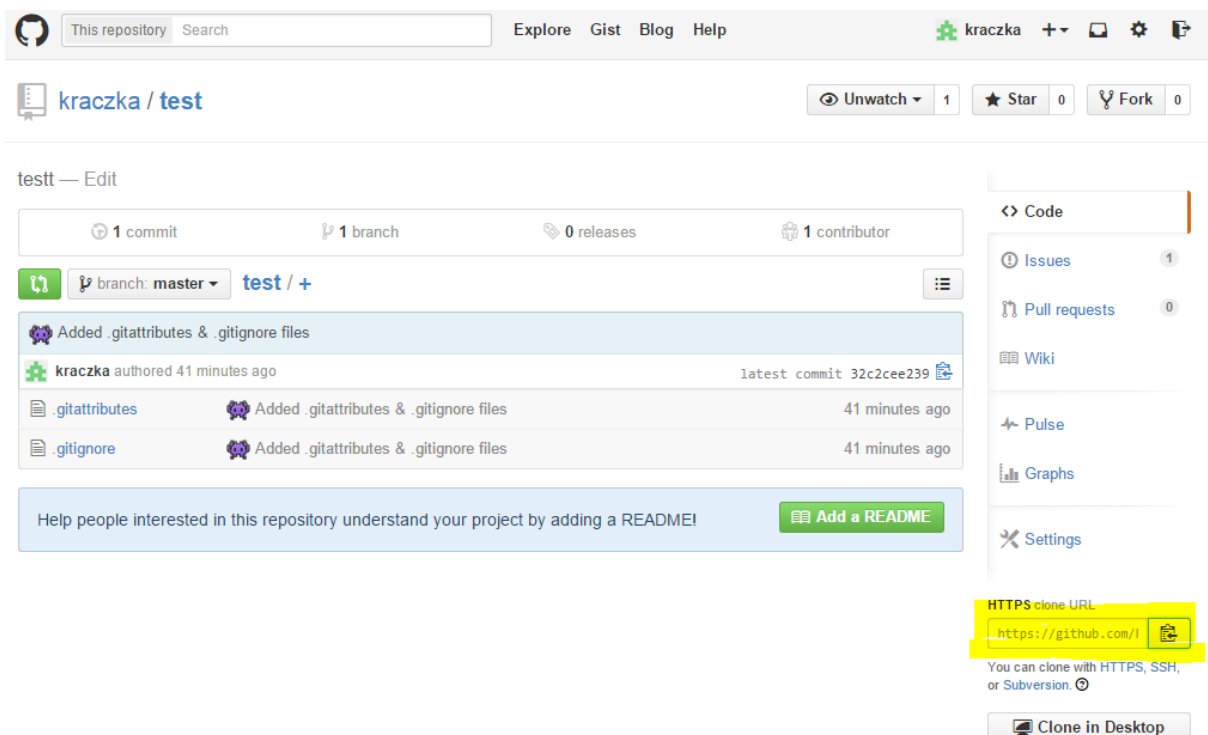
1. Tworzymy nowe repozytorium na GitHub'ie (nazwa dowolna).



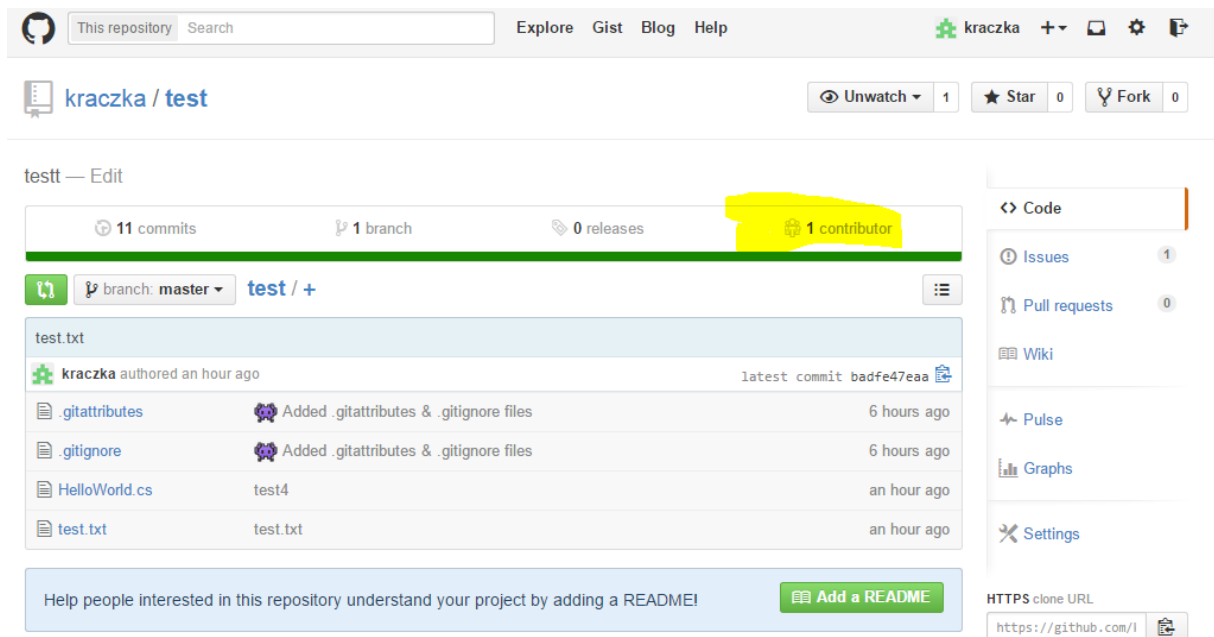
2. Tworzymy folder na dysku w którym będą się znajdować pliki z repozytorium GIT.
3. Klikamy prawy przyciskiem myszy na folder i wybieramy 'Git Clone'.



Url do repozytorium znajduje się we właściwościach utworzonego repozytorium na GitHub (zakreślony na żółto).



## Dodanie współpracownika do projektu



The screenshot displays the GitHub interface for a repository named 'test' by user 'kraczka'. At the top, navigation links include 'This repository', 'Search', 'Explore', 'Gist', 'Blog', and 'Help'. The repository header shows 'kraczka / test' with 'Unwatch', 'Star' (0), and 'Fork' (0) buttons. Below this, the repository statistics are shown: '11 commits', '1 branch', '0 releases', and '1 contributor' (highlighted in yellow). The commit history table lists recent changes to files like .gitattributes, .gitignore, HelloWorld.cs, and test.txt. On the right, a sidebar offers links to 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. At the bottom, a blue banner encourages adding a README, and the HTTPS clone URL is provided.

Aby dodać nowego współpracownika, należy kliknąć w link zakreślony na żółto (z powyższego ekranu), następnie wprowadzić dane współpracownika i zatwierdzić.

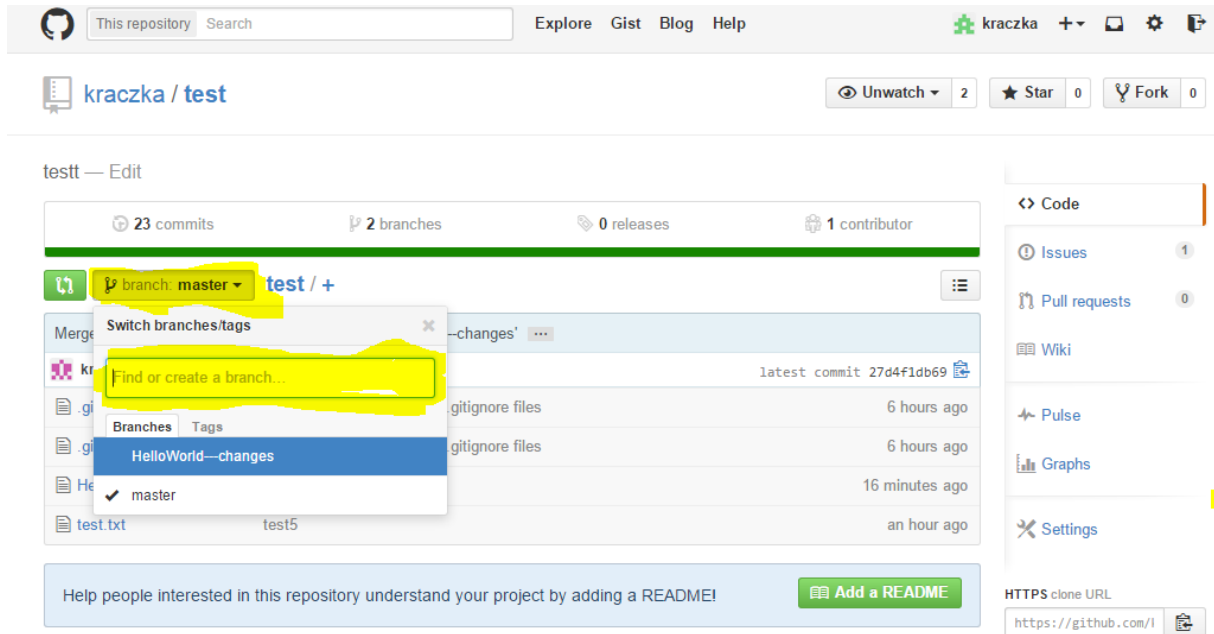
### Zadania

1. Należy wykonać commit 10 plików (jako 10 odrębnych commitów) do repozytorium z minimum 30 liniami kodu źródłowego napisanego w dowolnym języku programowania (wyszukać w internecie).
2. Należy dodać jedną osobę jako współpracownika w repozytorium.
3. Dwie osoby jednocześnie wprowadzają zmiany w tym samym pliku. Następnie obie osoby zatwierdzają (commit) zmiany na serwer.
  - a. Pierwsza osoba w pierwszej linijce pliku. Druga osoba w drugiej linijce pliku.
  - b. Obie osoby w tych samych linijkach

Jeśli wystąpi konflikt i nie zostanie on rozwiązany automatycznie, należy rozwiązać go ręcznie (za pomocą edytora).

## Gałęzie (Branch)

Aby stworzyć nową gałąź (branch) należy kliknąć opcję 'branch' w oknie projektu i wpisać nazwę nowej gałęzi (branch) oraz nacisnąć Enter.



### Łączenie gałęzi (branch) do gałęzi głównej (master branch)

1. Kliknij prawym przyciskiem myszy na folder (na dysku) w którym znajduje się zawartość głównej gałęzi (master branch).
2. Wybierz z menu kontekstowego TortoiseGit następnie z submenu opcję Merge.
3. Wybierz gałąź (branch) z jaką chcesz dokonać złączenia (merge).

#### Zadania:

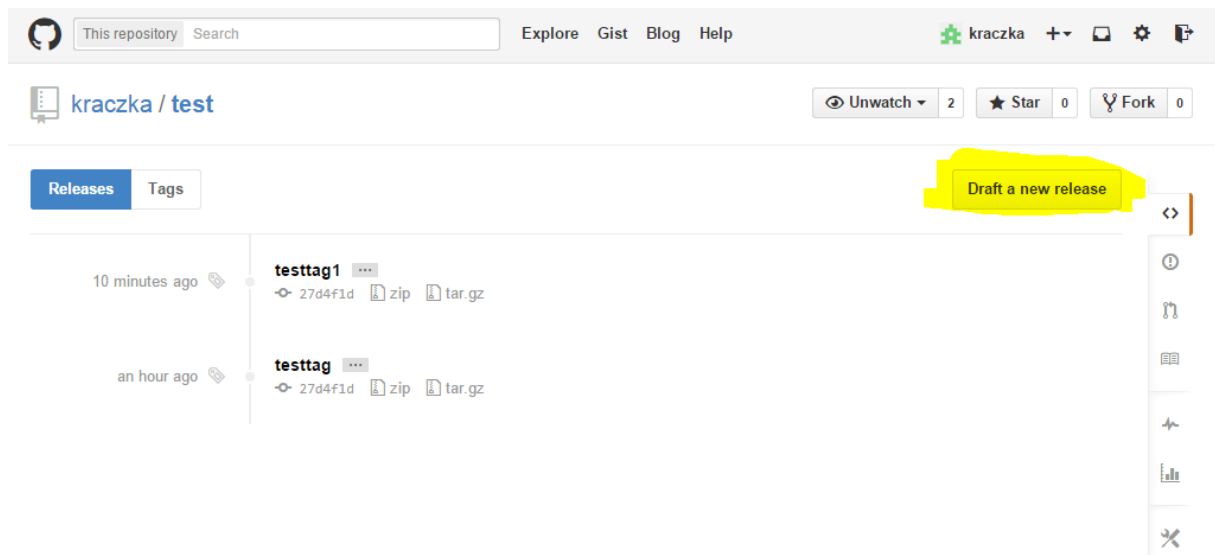
1. Stwórz nowego branch'a w projekcie.
2. Dokonaj 10 zmian w utworzonej gałęzi (zmiany zatwierdzaj oddzielnymi commit'ami).
3. Dokonaj 10 zmian w głównej gałęzi (master branch).
4. Wykonaj złączenia utworzonej gałęzi do gałęzi głównej.

# Tag wersji

Aby stworzyć Tag należy:

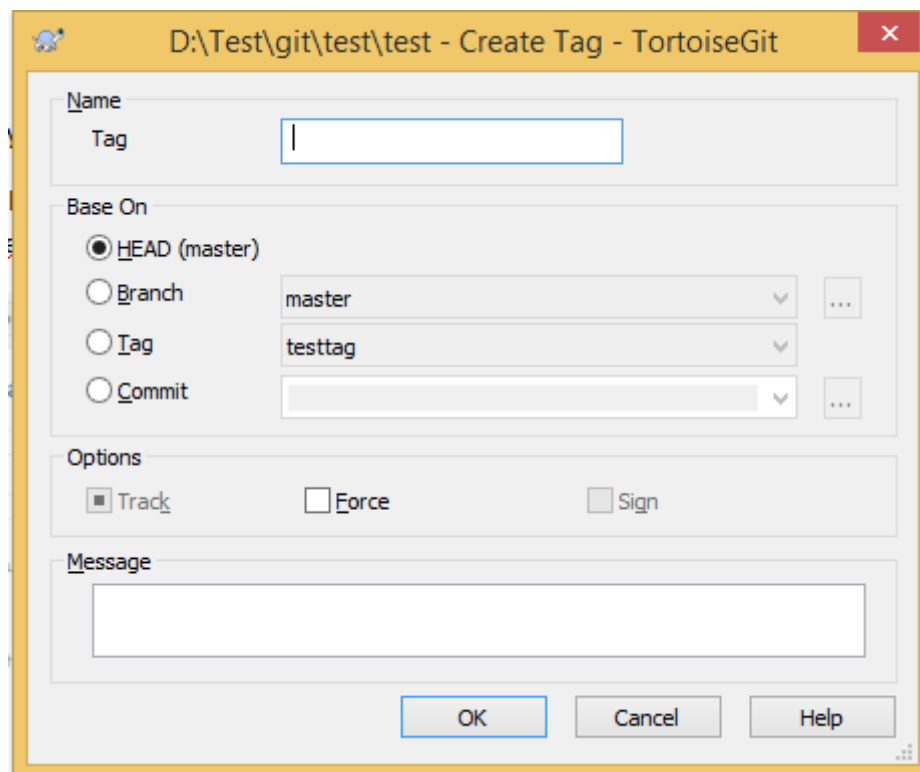
## 1. Za pomocą klienta web'owego

Wybrać **Releases** z okna głównego projektu. Następnie **Draft a new release**.



## 2. Za pomocą TortoiseGit

Kliknij prawym przyciskiem myszy na folder zawierający dane gałęzi głównej i wybierz TortoiseGit, następnie Create Tag.



Klikamy OK. Zatwierdzenie utworzenia Tag'a następuje po wybraniu opcji Push z submenu dla opcji TortiseGit z menu kontekstowego na danym folderze.

## Zadania

1. Stwórz dwa tag'i (nazwa dowolna):
  - a. Za pomocą klienta web'owego.
  - b. Za pomocą Tortoise Git.

## Blame

### Za pomocą klienta webowego

Należy wybrać repozytorium, wybrać odpowiedni plik i kliknąć opcję Blame.

### Za pomocą TortoiseGIT

Należy kliknąć prawy przyciskiem myszy na wybrany plik i wybrać opcję Blame.

## Ignore List

Ignore List są to wyrażenia regularne, które powodują pominięcie plików spełniających odpowiedni warunek podczas zatwierdzania (commit) zmian.

Wyrażenia regularne umieszczamy w pliku .gitignore w katalogu projektu na dysku lokalnym.

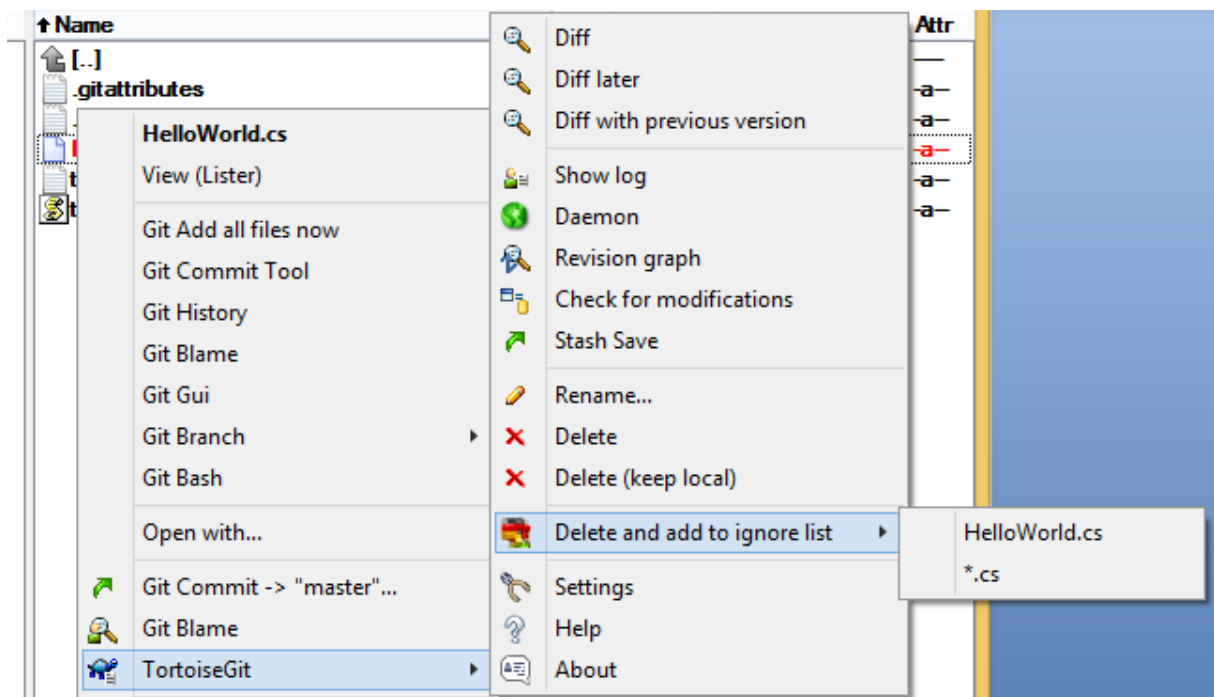
Dodanie nowego wyrażenia regularnego wymaga edycji pliku .gitignore lub wybrania pliku, który ma się znaleźć na liście oraz dodania go do Ignore List.

1. Edycja pliku:



Name	Ext	Size	Date
[.]		<DIR>	2015
.gitattributes		395	2015
.gitignore		623	2015
HelloWorld	cs	384	2015
test	txt	50	2015
test22	js	50	2015

## 2. Dodanie pliku do ignore list



### Zadania:

1. Utwórz nowy plik o nazwie test.txt i dodaj go do ignore list, przetestuj czy jest pomijany podczas zatwierdzania zmian (commit).
2. Utwórz kilka plików o rozszerzeniu \*.js i dodaj wszystkie pliki z takim rozszerzeniem do ignore list, przetestuj czy są one pomijane podczas zatwierdzania zmian (commit).
3. Utwórz wyrażenie regularne uniemożliwiające commit pliku posiadającego w nazwie 'tmp' . Utwórz kilka tego typu plików i przetestuj możliwość commit'u.

## Code review

Należy wykonać "code review" dla plików znajdujących się w repozytorium.

Wymagane 12 komentarzy przez każdego użytkownika (2 dla danego commit'u, 10 dla danej linijki), oraz przeprowadzenie dyskusji z drugim użytkownikiem pod min. jedną linijką.

## 1. Wybieramy 'Commits' z okna głównego repozytorium

testt — Edit

6 commits 1 branch 0 releases 1 contributor

branch: master test / +

helloworld

kraczka authored 6 minutes ago latest commit 6621f996ef

.gitattributes	Added .gitattributes & .gitignore files	an hour ago
.gitignore	Added .gitattributes & .gitignore files	an hour ago
HelloWorld.cs	helloworld	6 minutes ago
test.txt	3	23 minutes ago

Help people interested in this repository understand your project by adding a README! [Add a README!](#)

HTTPS clone URL  
<https://github.com/I>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Zakreślone na żółto.

## 2. Wybieramy dany commit.

3. Klikając '+' przy danej linijce możemy dodać komentarz do linijki, natomiast dodając komentarz na dole strony dodajemy go do całego Commit'u.

## Raportowanie błędów

Dodaj 20 błędów w module Raportowanie błędów (Issues) oraz przypisz je do 3 różnych kamieni milowych (milestones).