

# Jak przygotować konto na githubie

# Plan

- Repozytorium –jak je przygotować?
- Readme
- .gitignore
- Zainspiruj się

# Repozytorium – jak je przygotować?

# Repozytorium –jak je przygotować?

## Wstęp

Dobrze przygotowane repozytorium to nie tylko kod. Oczywiście jest on bardzo ważny, ale dobrze udokumentowany i opisany będzie się lepiej prezentował.

W tej prezentacji poznasz kilka zasad jak dobrze przygotować repozytorium, co jest ważne oraz na co zwracają uwagę przeglądający. Na końcu znajdziesz kilka linków do najpopularniejszych repozytoriów.

# Repozytorium –jak je przygotować?

## Nazwa repozytorium i opis

Dobrze przygotowane repozytorium to nie tylko kod. Oczywiście jest on bardzo ważny, ale dobrze udokumentowany i opisany będzie się lepiej prezentował.



## Stwórz plik **Readme**

Więcej o w kolejnym rozdziale - **Readme**



## Pilnuj porządku w projekcie

Usuń zbędne rzeczy typu pliki psd, pdf, pliki tworzone przez edytory np. .idea, a także foldery typu node\_modules. Jeśli chodzi o pliki, które są potrzebne do działania projektu, a których nie chcesz wrzucać do zdalnego repozytorium stwórz plik .gitignore. Więcej o tworzeniu tego pliku w rozdziale **.gitignore**

# Repozytorium –jak je przygotować?

Popular repositories	
	<b>atom-shortcuts-cheat-sheet</b> A shortcuts cheat sheet for Atom.
	<b>SJSJ</b>

Brak opisu – nazwa nic nam nie mówi

Popular repositories	
	<b>atom-shortcuts-cheat-sheet</b> A shortcuts cheat sheet for Atom.
	<b>SJSJ</b> Simplified JavaScript Jargon

Krótki opis od razu mówi przeglądającemu czego może się spodziewać w środku



# Repozytorium – nazwa i opis

## Zmiana nazwy i opisu repozytorium

Nazwę i opis repozytorium możesz nadać na samym początku tworzenia repozytorium lub zmienić później. Aby zmienić nazwę repozytorium musisz wejść w ustawienia:

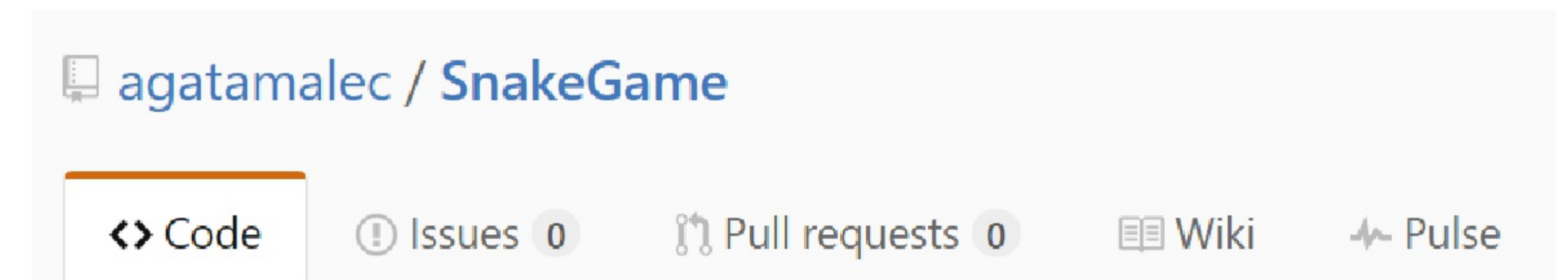
### Zmiana nazwy



Tu zmieniasz nazwę

Aby dodać lub zmienić opis repozytorium wystarczy, że klikniesz przycisk Edit.

### Dodanie/Zmiana opisu



*No description or website provided.* — [Edit](#)

Dodanie/Zmiana opisu

# Readme



# Readme.md – co to jest?

## Czym jest Readme?

Bardzo ważną częścią repozytorium jest plik Readme z rozszerzeniem md:

### Readme.md

Jest to plik tekstowy, który zawiera zazwyczaj opis repozytorium, licencję, informacje techniczne itp. Jest dokumentacją Twojego repozytorium.

## Przykład – aplikacja Dragula

### Inspiration

---

Have you ever wanted a drag and drop library that just works? That doesn't just depend on browser great support? That actually understands where to place the elements when they are dropped? A zillion things to get it to work? Well, so did I!

### Features

---

- Super easy to set up
- No bloated dependencies
- **Figures out sort order** on its own
- A shadow where the item would be dropped offers **visual feedback**
- Touch events!
- Seamlessly handles clicks *without any configuration*

### Install

---

You can get it on npm.

```
npm install dragula --save
```

# Readme.md – po co ?

## W jakim celu mam tworzyć plik Readme?

Jeśli chcesz, żeby ludzie używali Twojego projektu muszą wiedzieć do czego służy.

Jeśli jest to aplikacja jak ją uruchomić, jak z niej korzystać. Jeśli strona internetowa, którą chcesz się pochwalić, to co ją wyróżnia spośród innych?

## Chcesz, aby Twój kod był coraz lepszy.

Readme pozwala zebrać w jednym miejscu całą wiedzę na temat tego jak działa, na przykład Twoja aplikacja. Pozwala na usystematyzowanie informacji, które dadzą szerszy pogląd na to co już jest, a czego jeszcze brakuje.

## ...i jeszcze lepszy.

Domyślnie repozytoria na githubie są publiczne, tzn., że każdy może wejść i przejrzeć Twój kod. Jest to bardzo przydatne ponieważ ludzie mogą pomóc Ci znaleźć błędy, pomóc Ci stworzyć nowe funkcjonalności, doradzić itp. Bez odpowiedniej dokumentacji bardzo ciężko jest odnaleźć się w kogoś kodzie. Dlatego pamiętaj, aby zawsze dołączać Readme do swoich projektów.

# Readme.md – jak stworzyć?

## Jak stworzyć plik Readme?

Do stworzenia czytelnego pliku Readme służy bardzo prosty język znaczników Markdown.

Markdown jest językiem, który jest zarówno prosty do tworzenia jak i do czytania, dodatkowo jest konwertowany do HTML-a.

Markdown jest na tyle prosty, że nie wymaga nauki, można go napisać w każdym edytorze tekstowym.

Więcej:

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

## Jak wygląda Markdown?

### Headers

```
# This is an <h1> tag
## This is an <h2> tag
##### This is an <h6> tag
```

### Emphasis

```
*This text will be italic*
_This will also be italic_

**This text will be bold**
__This will also be bold__

_You **can** combine them_
```

### Lists

#### Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```



# Readme.md – co w nim umieścić?

## O czym można napisać w Readme?

Kilka przydatnych wskazówek co możesz zawrzeć w Readme. Wybierz te podpunkty, które najbardziej będą przydatne do Twojego projektu.

### Tytuł i krótki opis

W jakim celu twój projekt został stworzony z jakich technologii korzystasz.

### Link do demo

Jeśli nie masz swojej domeny, zajrzyj na Github pages: <https://pages.github.com/>

### Instrukcja instalacji

Np., jeśli używasz gulp'a to co trzeba zrobić (krok po kroku), żeby móc go uruchomić.

### Przykład kodu

Możesz w pliku Readme również umieścić kod, pokazujący np. jak korzystać z Twojej biblioteki, przykład:

You can also set the `containers` from the `options` object.

```
var drake = dragula({ containers: containers });
```

### Licencja i autor

Jeśli nie wiesz jaką licencję wybrać zajrzyj tutaj: <http://choosealicense.com/>

### Inspiracja

Skąd pomysł na taką stronę, aplikację



**.gitignore**

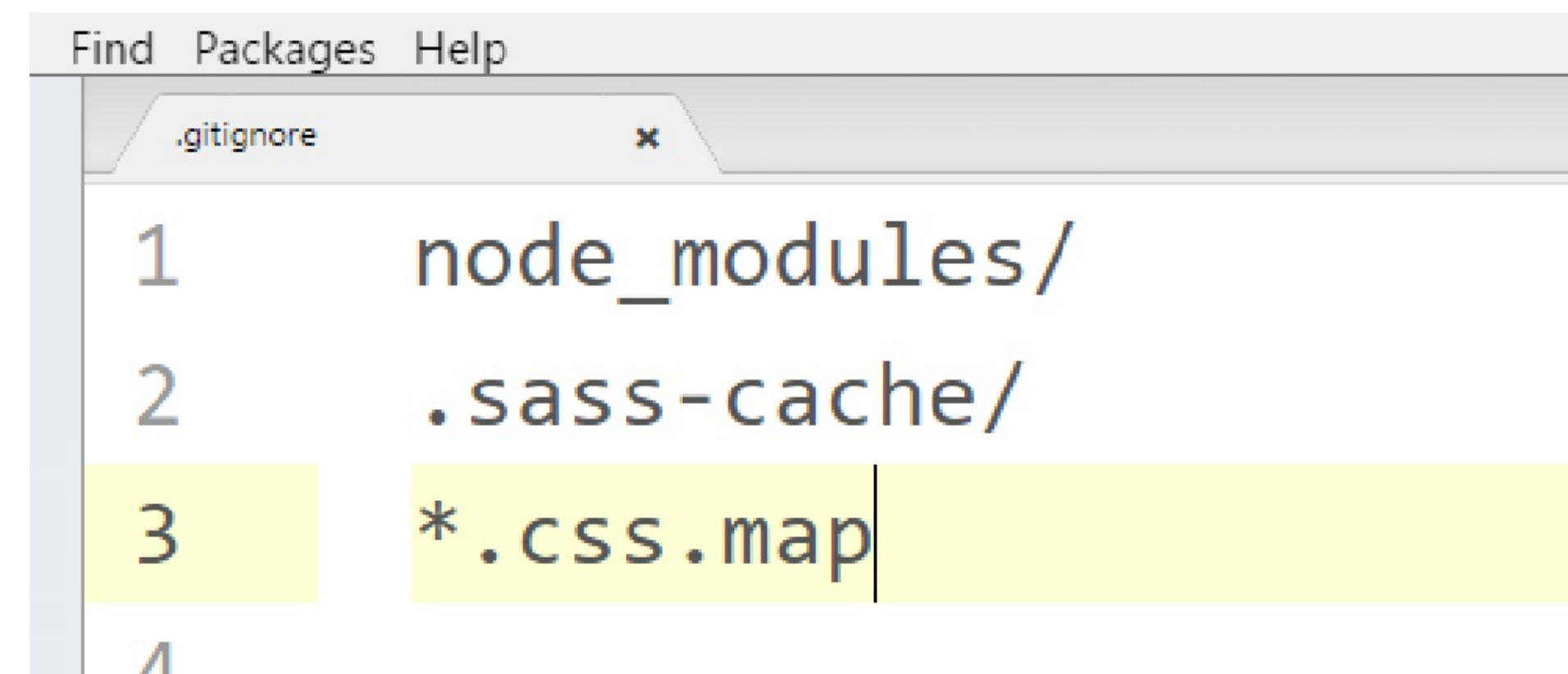
# .gitignore – co to jest?

## Ignorownie plików i folderów

Gdyby git zbierał wszystko co mamy w naszym lokalnym projekcie i wysyłał do zdalnego repozytorium, bardzo ciężko by się nam pracowało. Aby uniknąć wrzucania do zdalnego repozytorium wszystkiego co popadnie, możemy stworzyć specjalny plik tekstowy o nazwie - .gitignore

Każda linijka w .gitignore jest ścieżką do plików lub folderów, których nie chcemy umieszczać w zdalnym repozytorium. Plik .gitignore jest normalnym plikiem wersjonowanym i przechowywanym jak wszystkie inne.

## Przykład pliku .gitignore



```
Find Packages Help
.gitignore x
1 node_modules/
2 .sass-cache/
3 *.css.map
4
```

W powyższym przykładzie ignorujemy folder node\_modules oraz folder .sass-cache/ i wszystkie pliki z rozszerzeniem \*.css.map

## Przykładowe szablony .gitignore

<https://github.com/github/gitignore>

**Zainspiruj się!**

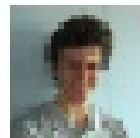





# Przykładowe konta na GitHub

## Aktywni użytkownicy

Zajrzyj na poniższe konta najbardziej aktywnych użytkowników Githuba.  
Podejrzyj ich opisy i pliki Readme, zainspiruj się!

<https://gist.github.com/paulmillr/2657075/>

#	User	Contribs	Location	Picture
#1	GrahamCampbell (Graham Campbell)	11088	The United Kingdom	
#2	fabpot (Fabien Potencier)	10000	Paris, France	
#3	weierophinney (Matthew Weier O'Phinney)	9946	Sioux Falls, SD, USA	
#4	rkh (Konstantin Haase)	8413	Zurich	
#5	josh (Joshua Peek)	8369	San Francisco, CA	