

**Kurs** Front End Developer  
GIT

# WPROWADZENIE DO GIT

**GIT** jest systemem kontroli wersji, czyli narzędziem, które pozwala na śledzenie historii zmian w plikach projektu.



Dzięki posiadaniu historii mamy dostęp do wersji projektu z każdego etapu jego powstawania. Jest to niezwykle istotne przy zespołowej pracy, ponieważ GIT pozwala na równoległe pisanie kodu projektu i zapobiega wzajemnemu nadpisywaniu plików.

Dzięki przechowywaniu plików na zewnętrznym serwerze mamy pewność, że nie utracimy naszego projektu w przypadku awarii naszego komputera. Mamy też dostęp do niego z każdego urządzenia, które ma skonfigurowane środowisko GIT.

# WPROWADZENIE DO GIT

Pracując nad projektem, który zlokalizowany jest w folderze na naszym komputerze możemy stworzyć w nim tak zwane repozytorium GITa, które będzie śledziło zmiany i pozwalało je zapisać.

Tworząc takie repozytorium będziemy mówić o tym, że pliki naszego projektu znajdują się w trzech stanach:

- Śledzony – dodanie pliku do śledzenia oznacza, że GIT będzie monitorował na bieżąco, czy w pliku są dokonywane zmiany (komenda `git add`)
- Zmodyfikowany – ten stan ma plik, który jest śledzony i zostały w nim dokonane zmiany, ale zmiany nie zostały zatwierdzone i zapisane do bazy danych GIT – taki plik można porównać z zatwierdzoną wersją pliku i zobaczyć jakie są między nimi różnice
- Zatwierdzony - plik i dane w nim zawarte zostały zapisane w repozytorium GIT (komenda `git commit -m "komentarz opisujący zmiany"`)

# WPROWADZENIE DO GIT

Mówiąc inaczej, gdy damy polecenie GITowi aby śledził plik, to na bieżąco sprawdza on, czy plik jest modyfikowany (stan śledzony).

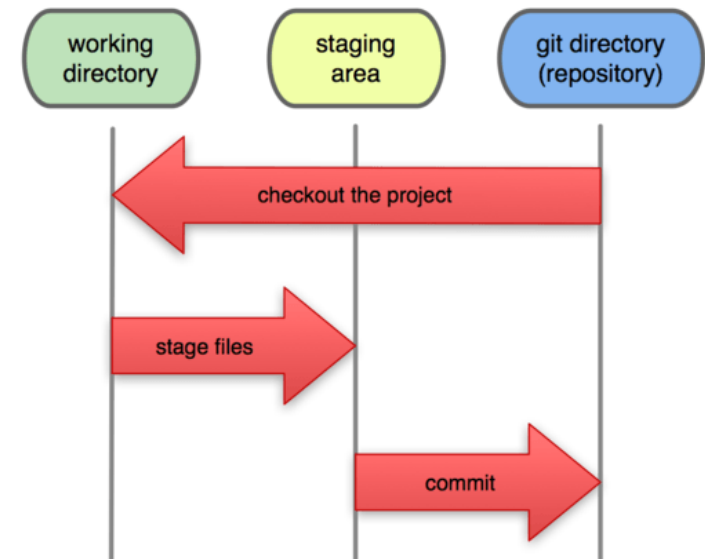
W momencie dokonania zmian w pliku przechodzi on w stan zmodyfikowany i trafia do przechowalni, w której możemy zobaczyć zmiany jakie zostały dokonane w stosunku do wcześniejszej wersji.

Gdy chcemy zatwierdzić wszystkie zmiany, to używamy komendy

```
git commit -m "komentarz opisujący zmiany"
```

zapisując je w lokalnym repozytorium GIT – czyli przechodzimy do stanu zatwierdzonego 😊

## Local Operations



# INSTALACJA GIT

Pobierz i zainstaluj GIT na swoim systemie operacyjnym

Windows, OSX, Linux: <https://git-scm.com/downloads>

Po instalacji uruchom Git Bash:

Windows (domyślnie): Menu Start -> Git -> Git Bash

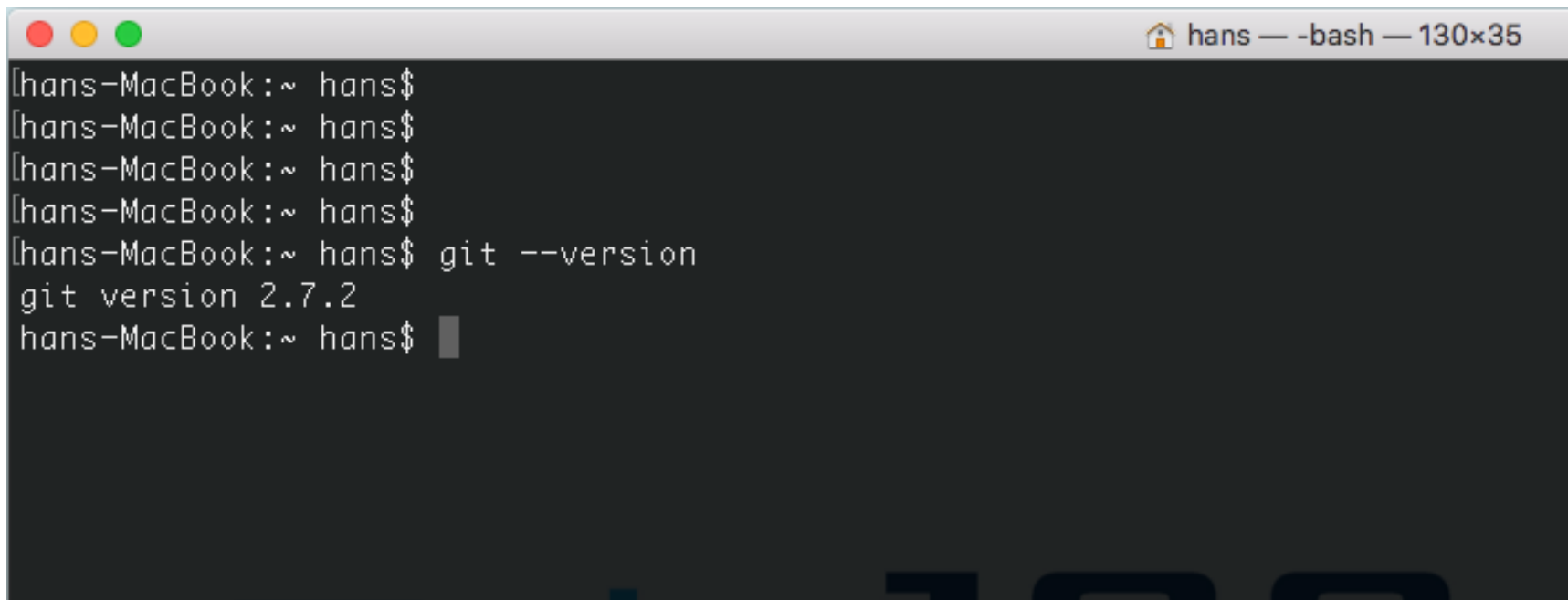
OSX, Linux: Uruchom Terminal

A nast.ępnie sprawdź zainstalowaną wersję GIT wydając komendę:

```
git --version
```

# INSTALACJA GIT

Wynikiem tej operacji powinno być wyświetlenie obecnie zainstalowanej wersji GIT



```
hans — -bash — 130x35
[hans-MacBook:~ hans$
[hans-MacBook:~ hans$
[hans-MacBook:~ hans$
[hans-MacBook:~ hans$
[hans-MacBook:~ hans$ git --version
git version 2.7.2
hans-MacBook:~ hans$
```

# KONFIGURACJA GITA

W celu korzystania z repozytorium GIT niezbędna jest jego podstawowa konfiguracja, która pozwala na identyfikację użytkownika wprowadzającego zmiany

Wszystkie działania konfiguracyjne prowadzimy w `Git Bash/Terminal`

Na początek musimy skonfigurować użytkownika GITa. Robimy to wpisując w terminalu następujące komendy. **Każdą komendę zatwierdzamy przyciskiem Enter.**

```
git config --global user.name "Jan Nowak"  
git config --global user.email jannowak@example.com
```

**\*\* w miejsce imienia i nazwiska oraz adresu wpisujemy własne dane.**

Podstawowa konfiguracja Git zakończona. Nie zamykaj Terminala, ponieważ w dalszej części wprowadzenia będziesz z niej korzystać.

# ZDALNE REPOZYTORIUM GITA

Pliki naszego projektu możemy przechowywać w zdalnym repozytorium. Na potrzeby kursu będziemy korzystać z repozytorium oferowanego przez serwis GitHub

<https://github.com/>

Pierwszym zadaniem będzie stworzenie konta na serwisie.

Więcej informacji na temat działania serwisu GitHub można znaleźć pod tym linkiem <https://guides.github.com/activities/hello-world/> - tworząc nazwę użytkownika używaj liter od a do z i cyfr od 0 do 9

**NIE UŻYWAJ polskich liter, dużych liter, znaków specjalnych i spacji!**

Na potrzeby wprowadzenia pokażemy w jaki sposób stworzyć nowy projekt w serwisie GitHub oraz jak stworzyć lokalne repozytorium i połączyć je ze zdalnym repozytorium na GitHub.

A na końcu wyślemy stworzone przez nas pliki do zdalnego repozytorium na GitHub.



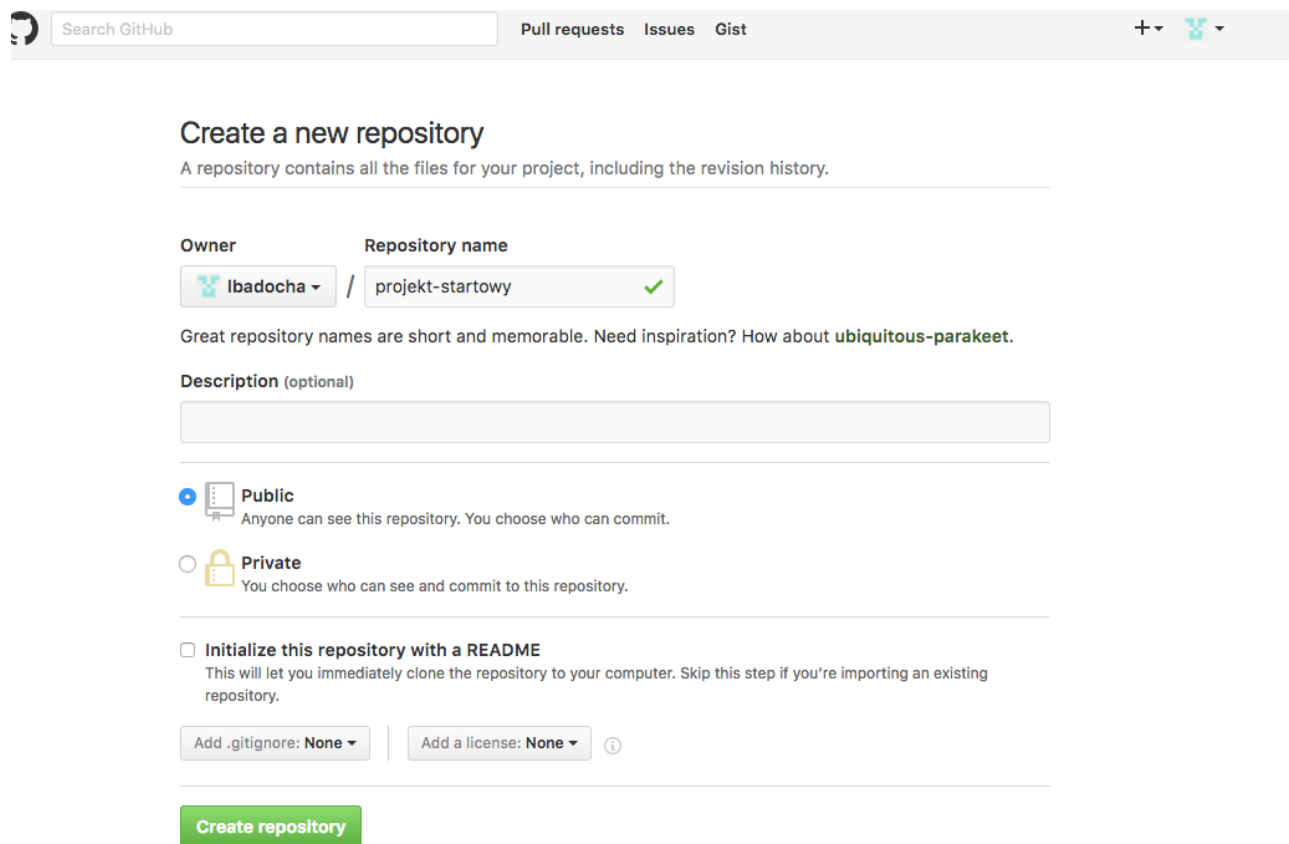
# NOWY PROJEKT GIT

W serwisie GitHub tworzymy nowy projekt, klikając przycisk Start a project



# NOWY PROJEKT GIT

Nadajemy nazwę naszemu projektowi `projekt-startowy` i klikamy przycisk `Create repository`




Search GitHub Pull requests Issues Gist + -

## Create a new repository


A repository contains all the files for your project, including the revision history.


Owner Repository name

 lbadocha / projekt-startowy ✓

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-parakeet**.

Description (optional)

☒  **Public**  
Anyone can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

**Create repository**

# NOWY PROJEKT GIT

Tworzymy folder na Pulpicie o nazwie `projekt-startowy`.

Następnie przechodzimy do niego wydając komendę w terminalu:

```
cd ~/Desktop/projekt-startowy
```

(Aby lepiej zrozumieć działanie podstawowych komend w terminalu zapoznaj się z tym materiałem [https://tutorial.djangogirls.org/pl/intro\\_to\\_command\\_line/](https://tutorial.djangogirls.org/pl/intro_to_command_line/) )

Folder `projekt-startowy` będzie folderem naszego repozytorium GIT

W folderze tworzymy plik `index.html` (komendą `touch index.html`)

Plik można także utworzyć w zwykłym windowsowym Exploratorze, lub w Finder na OSX.

# NOWY PROJEKT GIT

Następnie w terminalu wykonujemy następujące polecenia.

`git init` - inicjuje lokalne repozytorium

`git add *` - dodaje do śledzenia pliki i foldery z naszego katalogu (w naszym przypadku będzie to plik `index.html`)

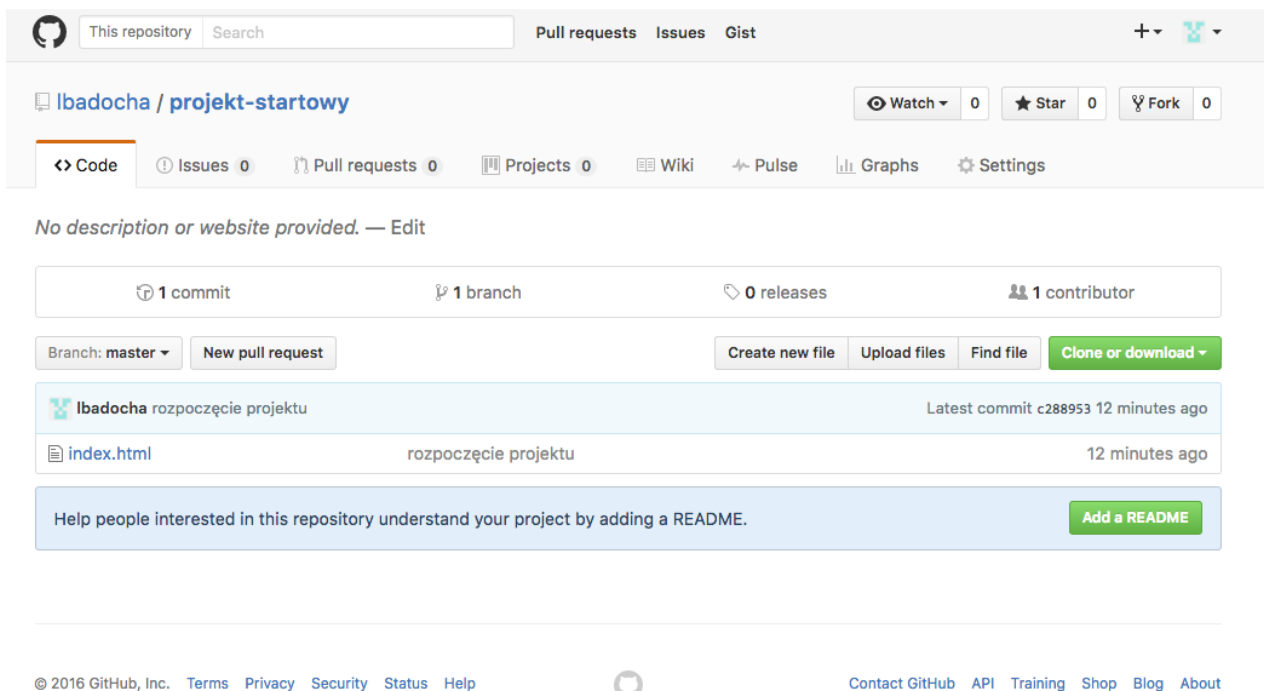
`git commit -m 'rozpoczęcie projektu'` - zatwierdzamy wersję plików w lokalnym repozytorium - w nawiasach

`git remote add origin https://github.com/twoja_nazwa_uzytkownika/projekt-startowy.git`  
- czyli tworzymy powiązanie lokalnego repozytorium z repozytorium zdalnym

`git push -u origin master` - wysyłamy stan lokalnego repozytorium do repozytorium zdalnego

# NOWY PROJEKT GIT

Wejdź na stronę serwisu GitHub i wybraniu naszego repozytorium (projekt-startowy). Powinieneś zobaczyć swój plik `index.html`, który został wysłany do zdalnego repozytorium.



# NOWY PROJEKT GIT

Edytuj plik `index.html`, dodaj do niego tekst Projekt GIT i zapisz go.

Dodaj pliki do śledzenia:

```
git add *
```

Zatwierdź zmiany w lokalnym repozytorium z komentarzem:

```
git commit -m 'Pierwsza zmiana'
```

Wyślij zmiany do repozytorium zdalnego:

```
git push
```

Aktualną wersję pliku oraz historię zmian możesz prześledzić na swoim profilu na GitHub, wybierając repozytorium `projekt-startowy`.

# PODSTAWY UŻYWANIA GITA

Najczęściej używane komendy GIT:

`git config --list` - pokazuje konfigurację wszystkich opcji danego repozytorium

`git init` - tworzy nowe repozytorium plików

`git status` - wyświetla informacje o aktualnym stanie repozytorium

`git log` - pokazuje wszystkie zatwierdzone zmiany w lokalnym repozytorium

# PODSTAWY UŻYWANIA GITA

Zapoznaj się z informacjami dostępnymi w oficjalnym poradniku GITA

<https://git-scm.com/book/pl/v1/Pierwsze-kroki>

Na początek zapoznaj się szczególnie z rozdziałami 1, 2, 3 oraz 5.

Pełna dokumentacja GITA dostępna jest pod linkiem

<https://git-scm.com/documentation>





Akademia 108

<https://akademia108.pl>