

# Technologie Informacyjne

## Systemy kontroli wersji

Opracował: Maciej Penar

## Spis treści

1. Systemy kontroli wersji .....	3
2. GIT .....	3
3. Koncept forków i pull requestów .....	6
4. Kilka komend gita .....	8
5. Zadanie .....	9
Zadanie 1 .....	9
Zadanie 2 .....	9
Zadanie 3 .....	9

## 1. Systemy kontroli wersji

Systemy kontroli wersji (Version Control System) to systemy służące do archiwizowania dowolnej treści np. kodów źródłowych, dokumentów ([wiki](#)). Czasem spotyka się skrót SCM – Source Code Management – gdy akcentujemy, że archiwizowaniu podlega kod aplikacji.

Systemy te mają możliwość:

- przeglądania stanu treści z dowolnego momentu w czasie
- śledzenia alternatywnych wersji treści
- łączenia kilku różnych wersji treści

Najczęściej spotykanym VCS-em jest **git**, który działa jako sieć P2P. Alternatywnymi SCM-ami są SVN oraz Mercurial.

My zajmiemy się **git-em**. Ze względu na swoją popularność istnieje kilka serwisów które umożliwiają hostowanie tzw. repozytoriów (projektów) treści za darmo. W zależności od wybranego serwisu mamy różne ograniczenia:

- GitHub - <https://github.com/> - umożliwia tworzenie darmowych publicznych repozytoriów. Jeśli repozytorium jest stroną webową, to od razu mamy możliwość hostowania takiej strony (bez serwera aplikacji) – na pewnym ustalonym hoście.
- BitBucket - <https://bitbucket.org/> - umożliwia tworzenie darmowych prywatnych repozytoriów (drużyna do 5 osób).
- Gitlab - <https://gitlab.com/> - płatny dostawca

Repozytorium Git to nic innego jak aplikacja internetowa, hostowana jak zwykła strona internetowa. Jeśli ktoś chce to może repozytorium hostować na swoim sprzęcie np. z użyciem implementacji Bonobo ([link](#)).

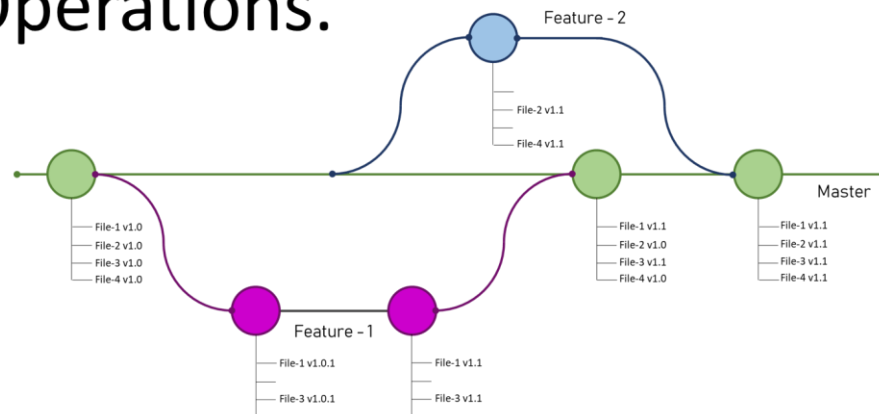
## 2. GIT

Do pobrania tutaj: <https://git-scm.com/>

W git-cie główną ścieżkę kodu (treści) przechowujemy na tzw. gałęzi (branch) o nazwie main (kiedyś master). W folderze zawierającym pliki repozytorium możemy w dowolnym momencie przełączyć się lub otworzyć inną dowolnie nazwaną gałąź (operacja 'checkout'). Zaciągnięcie dotychczasowych zmian na gałęzi na której się znajdujemy wykonujemy za pomocą operacji 'pull'.

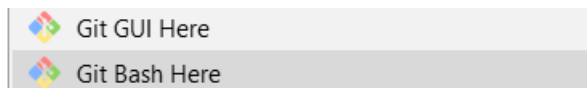
Graficznie archiwum treści reprezentowane jest często jako graf skierowany którego gałęzie najczęściej rozbiegają się i zbiegają do gałęzi main/master. Jak na obrazku pod spodem:

# GIT Branch and its Operations.



<https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fwp-content%2Fdigitalvarys.com%2Fwp-content%2Fuploads%2F2019%2F06%2FGIT-Branchand-its-Operations.png%3Ffit%3D1921%252C1057%26ssl%3D1&f=1&nofb=1>

Po instalacji git-a pod prawym klawiszem myszki powinniśmy zobaczyć dwie opcje:



Opcja:

- Git GUI Here – otwiera w podanej lokalizacji graficzny interfejs użytkownika GIT-a
- Git Bash Here – otwiera w podanej lokalizacji terminal

Wybierając opcje: Git Bash Here możemy wpisać komendy programu git dzięki którym możemy kontrolować repozytoria.

Na przykład komenda:

**git clone [adres repozytorium]**

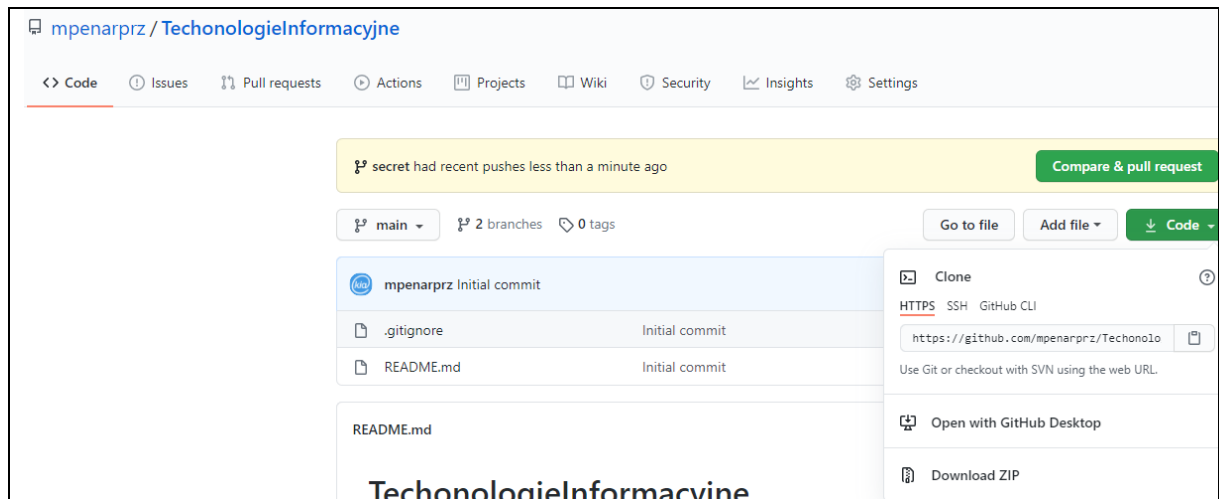
Pobierze aktualny stan podanego w adresie repozytorium – używając gałęzi main. Na przykład:

**git clone <https://github.com/mpenarprz/TechonologieInformacyjne.git>**

Pobierze repozytorium dla tego kursu

## UWAGA

Na githubie naciskając przycisk Code można pobrać adres repozytorium do komendy git clone



Jeśli repozytorium ulegnie zmianie, to zmiany możemy pobrać wykonując polecenie

**git pull**

#### UWAGA

Pamiętajcie, że katalog z projektem spiętym z gita respektuje gałąź. Terminal pokazuje na jakiej gałęzi jesteście:



Jeśli posiadacie uprawnienia do repozytorium, to możecie dodać pliki/zmiany za pomocą komendy:

**git add [plik]**

Lub dodać wszystkie nowe pliki/zmiany

**git add -A**

Zatwierdzenie zmian lokalnie odbywa się przez komendę:

**git commit -m „Komentarz”**

Z kolei wypchanie zmian do innego repozytorium poprzez komendę

**git push origin [gałąź]**

I w tym miejscu dygresja – najczęściej korzystając z zewnętrznych repozytoriów chcemy by kopia zapasowa naszej treści była przechowana właśnie na jakimś bezpiecznym sprzęcie – a nie lokalnie na komputerze. Dlatego korzystając z git-a (i inny SCM-ów) szablonowo używamy 3 komend do dodania/zatwierdzenia/wypchnięcia zmian:

**git add [plik]**

**git commit -m „Komentarz”**

**git push origin [gałąź]**

Dla leniwych: dodanie wszystkich zmian, zatwierdzenie i wypchnięcie do zewnętrznego repozytorium to te trzy komendy:

**git add -A**

**git commit -m „Komentarz”**

**git push origin master**

W komendzie **git push** pojawia się słowo origin – w tym miejscu możemy podać alias dla zewnętrznego repozytorium – możemy mieć ich kilka (jeśli jesteśmy paranoikami).

Utworzenie nowej gałęzi:

**git branch [gałąź]**

Ostatnia rzecz o której chciałbym wspomnieć, to komenda:

**git checkout [gałąź]**

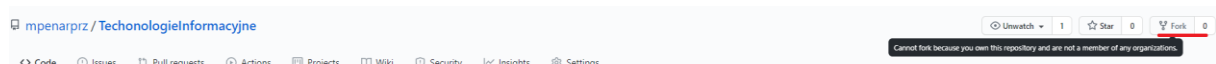
Które powoduje zmianę obecnej gałęzi kodu. Najczęstszym wzorcem korzystania jest:

**git checkout [gałąź]**

**git pull**

### 3. Koncept forków i pull requestów

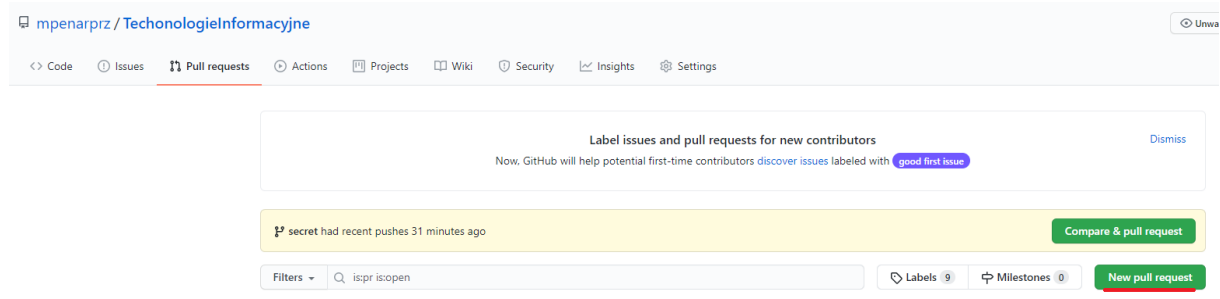
Git zdobył popularność ze względu na popularność rozwiązań open-source których kod jest wystawiany publicznie na tego typu repozytoriach. W swojej filozofii Git umożliwia „fork” (kopię) repozytorium. Na stronie githuba fork repozytorium wykonujemy za pomocą guzika „fork”:



Na naszym koncie zostanie zapisana kopia repozytorium z **danego momentu**. Tak naprawę jest to branch 😊.

Na swoim koncie macie uprawnienia do zapisu, dlatego publicznie dostępny projekt możecie dowolnie modyfikować na swoim lokalnym „forku”.

Sytuacja staje się ciekawa w momencie gdy chcemy podzielić się ze światem naszymi zmianami i chcemy by źródłowe repozytorium zaakceptowało zmiany z naszego „forka”. W tym celu musimy uruchomić proces zwany „pull request”



Pull request może zostać wykonany:

- w obrębie repozytorium – czyli łączymy 2 gałęzie. Najczęściej jakąś gałąź do gałęzi master/main.
- Pomędzy forkami – czyli łączymy zmiany pomiędzy 2 dowolnymi gałęziami forków.

## 4. Kilka komend gita

Komenda	Co robi
git clone [adres]	Pobiera adres repozytorium
git add -A	Dodaje do tranży (commit) wszystkie zmiany
git add [plik]	Dodaje do tranży (commit) zmiany podanego pliku
git commit -m [komenatrz]	Zatwierdza tranżę zmian
git push origin [gałąź]	Wysyła zatwierdzenia pod adres origin (zdalny) do danej gałęzi
git pull	Pobiera zmiany na aktualnej gałęzi
git checkout [gałąź]	Zmienia gałąź
git reset --hard	Twardy reset repozytorium
git status	Stan zatwierdzenia
git branch	Tworzy lokalnie gałąź



## 5. Zadanie

Sporządźcie sprawozdanie w dowolnym edytorze tekstowym – zamieście zrzuty ekranu + krótki opis tego co się dzieje na ekranie.

### ZADANIE 1

1. Zainstalujcie gita: <https://git-scm.com/>
2. Zarejestrujcie się na githubie: (<https://github.com/>)

### ZADANIE 2

1. Utwórzcie repozytorium na githubie:
  - a. Klik na konto
  - b. Your repositories
  - c. New
2. Utwórzcie przykładowe publiczne repozytorium
3. Zaciągnijcie świeżo utworzone repozytorium
4. Dodajcie przykładowy plik do gałęzi main (git add)
5. Utwórzcie gałąź o dowolnej nazwie (git checkout)
6. Dodajcie inny przykładowy plik do nowo utworzonej gałęzi

### ZADANIE 3

1. Sforkujcie moje repozytorium:  
<https://github.com/mpenarprz/TechonologieInformacyjne>
2. Zaciągnijcie świeżo sforkowane repozytorium
3. Dodajcie sprawozdanie do swojego repozytorium