Zawartość

[1. Przygotowanie do rozpoczęcia pracy z repozytorium na serwerze github. 1](#_Toc528700834)

[1.1. Ogólny opis pracy z repozytorium. 1](#_Toc528700835)

[1.2. Przygotowanie i konfiguracja TortoiseGit. 2](#_Toc528700836)

[1.3. Podstawowe pojęcia przy pracy z repozytorium. 2](#_Toc528700837)

[2. Sklonowanie pierwszego repozytorium 3](#_Toc528700838)

# Przygotowanie do rozpoczęcia pracy z repozytorium na serwerze github.

Na potrzeby informacyjne przyjmujemy, że repozytorium znajduje się na publicznym serwerze gihthub dostępnym pod adresem <https://github.com>

Do pracy z repozytorium na serwerze gita potrzebne jest konto dostępowe. Github oferuje zarówno repozytoria odpłatne jak i darmowe, które są publiczne i dostępne dla wszystkich . Publiczne repozytoria dostępne są dla wszystkich tylko do odczytu. Uprawnienia do modyfikacji nadaje właściciel zakładanego repozytorium.

Po utworzeniu konta serwis github oferuje interfejs WWW, przez który można zarządzać swoimi repozytoriami lub udostępnionymi przez innych użytkowników do edycji.

Lokalnie do pracy z repozytorium można wykorzystać konsolę lub interfejs graficzny np. [TortoiseGit](https://tortoisegit.org/). Poniższa instrukcja będzie opisywała obsługę z wykorzystaniem tortoisegit. Wiele narzędzi programistycznych np. Visual Studio posiada wbudowany mechanizm obsługi repozytorium.

## Ogólny opis pracy z repozytorium.

Praca z repozytorium polega tworzeniu kod a następnie na wersjonowaniu kolejno dodawany zmian. Mechanizm repozytorium przechowuje kolejno tworzone wersje w sposób sterowalny przez użytkownika. Git pozwala na cofanie zmian w obydwu kierunkach, łączenie pracy wielu członków zespołu, oraz zapewnia dostęp do dokumentów z różnych miejsc w sieci.

W przypadku gita tworzone jest lokalne repozytorium(na dysku lokalnym) oraz drugie na serwerze. Taki mechanizm pozwala na dowolne modyfikacje oraz szybki powrót do pierwotnej wersji.

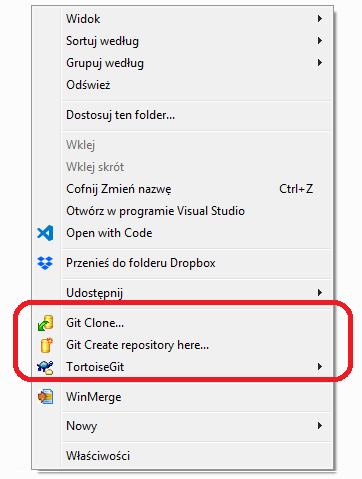
Zaimplementowane funkcje zwalniają użytkownika z opracowywania systemów przechowywania kopii a w konsekwencji zagubienia się w kolejnych wersjach.

Przechowywanie kodu na serwerze jest również idealnym sposobem na jednoczesną pracę wielu osób na tych samych dokumentach. Dostępne narzędzia w znacznym stopniu ułatwiają łączenie powstałych fragmentów dokumentu w wersję ostateczną.

Do repozytorium na serwerze dodawane są zmiany w konkretnych liniach plików a nie całe pliki. Takie podejście znacznie zmniejsza ilość przechowywanych danych, co również znacznie przyspiesza poruszanie się po kolejnych wersjach dokumentu.

Repozytorium pozwala na tworzenie gałęzi (branch), które stanowią alternatywną wersję dla głównej. Funkcjonalność ta pozwala na tworzenie kodu tylko na potrzeby testów lub sprawdzenia innego rozwiązania jakiegoś problemu. Przy pracy grupowej można stworzyć osobną gałąź dla każdego a następnie łączenia sprawdzonego kodu. Pozwoli to uniknąć przenoszenia błędów poza lokalne środowisko jednego użytkownika. j

## Przygotowanie i konfiguracja TortoiseGit.



Po pobraniu i zainstalowaniu tortoisegit(zarówno samego gita jak i interfejsu graficznego) w menu kontekstowym systemu powinny być widoczne dodatkowe opcje związane z obsługą repozytorium. Dostępne menu to:

* Git clone – tworzy repozytorium lokalnie na dysku na podstawie już istniejącego na serwerze
* Git create repository here – tworzy lokalne repozytorium
* TortoiseGit – pozostałe opcje

Katalogi zawierające repozytoria będą miały zmienione ikony. Nowe ikony będą informowały o stanie repozytorium, czy zostało zmienione czy nie.

## Podstawowe pojęcia przy pracy z repozytorium.

Aby dogłębnie poznać różne możliwości gita polecam profesjonalne tutoriale dostępne w sieci. Na potrzeby tej instrukcji zostaną wymienione i opisane podstawowe pojęcia, najczęściej wykorzystywane w pracy z repozytorium

* Commit – inaczej wczytanie lub „zapchanie” kodu do lokalnego repozytorium
* Push – „zapchanie” zmian do repozytorium na serwerze na podstawie wcześniejszego commita
* Pull – pobranie zmian z serwera (jednocześnie dokunując merga)
* Merge – połączenie dwóch gałęzi tego samego repozytorium lub dołączenie zmian będących na serwerze do lokalnego repozytorium
* Branch – gałąź repozytorium będąca alternatywą dla właściwej wersji dokumentu
* Create branch – tworzy nową gałąź
* Switch/Checkout – zmiana brancha
* Stashe save – zapisanie w celu tymczasowym dodanych zmian poza mechanizmem wersji
* Rebase – polecenie pozwalające połączyć kilka commitów w jeden
* Reverte change by this commit – cofa zmiany dokonane przez wskazany commit
* Reset „master” to this – cofa stan całego repozytorium lokalnego do wskazanego commita
* Show log – pokazuje okno z historią kolejnych zmian

# Sklonowanie pierwszego repozytorium