

Git Commands:

Opis:	Komenda Gita:
Konfiguruje nazwę ->	git config --global user.name "name"
Tworzy nowe lokalne repozytorium ->	git init
Tworzy kopię roboczą lokalnego repozytorium ->	git clone /path/to/repo
Klonuje repozytorium ->	git clone username@host:/path/to/repo
Polecenie dodania jednego pliku ->	git add <particular file>
Dodanie wszystkich plików ->	git add .
Wyślij zmiany z treścią ->	git commit -m " message"
Wyślij pliki ->	git commit -a
Wypchnij zmiany do głównej gałęzi zdalnego repozytorium ->	git push origin master
Aktualny status plików do zmiany/dodania/zatwierdzenia ->	git status
Połącz się ze zdalnym repozytorium – dodaj serwer, aby móc do niego wypchnąć ->	git remote add origin <server>
Lista wszystkich aktualnie skonfigurowanych repozytoriów zdalnych ->	git remote -v
Utwórz nową gałąź i przejdź do niej ->	git checkout -b <branchname>
Przełącz się z jednej gałęzi do drugiej ->	git checkout <branchname>
Wypisz wszystkie gałęzie w swoim repozytorium ->	git branch
Usuń gałąź funkcji ->	git branch -d <branchname>
Wypchnij gałąź do zdalnego repozytorium, aby inni mogli z niego korzystać ->	git push origin <branchname>
Wypchnij wszystkie gałęzie do zdalnego repozytorium ->	git push --all origin
Usuń gałąź w swoim zdalnym repozytorium ->	git push origin :<branchname>
Pobierz i scal zmiany na serwerze zdalnym do katalogu roboczego ->	git pull
Scal gałąź z aktywną/bieżącą gałęzią ->	git merge <branchname>
Wyświetl wszystkie konflikty scalania ->	git diff

Zobacz konflikty z plikiem podstawowym ->

`git diff --base <filename>`

Wyświetl podgląd zmian przed scaleniem ->

`git diff <sourcebranch> <targetbranch>`

Możesz użyć tagowania, aby oznaczyć znaczący zestaw zmian, taki jak wydanie –

CommitId to identyfikator zestawu zmian, max 10; musi być unikalne ->

`git tag 1.0.0 <commitID>`

Wyświetla zatwierdzenia dokonane w tym repozytorium w odwrotnej kolejności chronologicznej; to znaczy, najnowsze zatwierdzenia pojawiają się jako pierwszą ->

`git log`

Wypcha wszystkie tagi do zdalnego repozytorium ->

`git push --tags origin`

Jeśli się pomylisz, możesz zastąpić zmiany w drzewie roboczym ostatnią zawartością w nagłówku ->

`git checkout -- <filename>`

Zmiany już dodane do indeksu oraz nowe pliki zostaną zachowane. Zamiast tego, aby porzucić wszystkie lokalne zmiany i zatwierdzenia, pobrać najnowszą historię z serwera i wskazać na nią lokalną gałąź master, wykonaj następujące czynności ->

`git fetch origin`

`git reset --hard origin/master`

Przeszukaj katalog roboczy foo() ->

`git grep "foo()"`