

Podstawowe komendy git

Komendy git możesz wykonać między innymi:

- w konsoli cmd (widnows start -> cmd. Otworzy się aplikacja Command Prompt. Aby przejść do katalogu głównego wpisz cd \ . Następnie wpisz cd + ścieżka do katalogu z twoim projektem, np: cd Users\Joanna\Desktop\moj-projekt
- w konsoli git bash, którą możesz pobrać ze strony <https://git-scm.com/downloads> . Po zainstalowaniu konsoli wejdź do folderu ze swoim projektem, kliknij prawy przycisk myszki i wybierz: Git Bash Here (otworzy się wtedy konsola).

Jeśli masz już konsolę z zaznaczoną ścieżką do katalogu, w którym chcesz umieścić swój projekt, możesz wpisać do niej pierwszą komendę (punkt pierwszy).

Podstawowe komendy git:

I - Repozytorium lokalne:

1. Utworzenie nowego repozytorium lokalnego:

-> `git init`

W katalogu z Twoim projektem powinien utworzyć się ukryty folder `.git`

2. Jeśli w swoim katalogu masz również pliki, których nie chcesz dodawać do repozytorium (bo nie będziesz chciał/a wysyłać ich później np. na githuba), możesz utworzyć plik `.gitignore`, w którym wpiszesz ścieżki do tych plików. Git będzie je pomijał przy dodawaniu plików do repozytorium.

-> `touch .gitignore`

3. Pokazuje zmiany, jakie zaszły w projekcie, np. nazwy plików, które zmieniły zawartość, ale nie zostały jeszcze dodane do stage wyświetlane są na czerwono, a takie, które dodano do stage, ale nie zrobiono jeszcze commit-a wyświetlane są na zielono.

-> `git status`

Podstawowe komendy git

4. Dodanie pliku do stage (jeśli nie wiesz, co to stage, pomyśl o nim jak o półce, na którą tymczasowo odkładasz swoje pliki. Stage przechowuje stan Twoich plików, dzięki czemu możesz zamknąć edytor kodu i wyłączyć komputer, a Twoje dane zostaną zachowane, abyś później mógł/mogła zapisać je na stałe w projekcie, robiąc commit).

-> `git add nazwa-pliku`, np: `git add index.html`

-> `git add .` - gdzie kropka oznacza, że dodajemy do stage wszystkie pliki z folderu projektu.

5. Usunięcie pliku ze stage:

-> `git rm --cached nazwa-pliku`, np: `git rm --cached index.html`

6. Dodanie commit-a:

-> `git commit -m "opis commit-a"` (gdzie -m oznacza message - można wtedy podać opis commita).

7. Wyświetlenie historii commitów (jeśli chcesz cofnąć się do któregoś w wcześniejszych commitów, potrzebujesz jego numer. Wyświetlając historię zobaczysz wszystkie zrobione commity z przypisanymi do nich numerami):

-> `git log` <- pokazuje więcej szczegółów

-> `git log --oneline` <- mniej szczegółów, co jest bardziej czytelne

8. Cofanie commitów (lub wyświetlanie kodu z wcześniejszych commitów)

-> `git checkout numer_commit`, np. `git checkout af6b84c`

Pokazuje w edytorze kod z wybranego commita. Nie kasuje historii, nie kasuje żadnych commitów.

Jeśli chcesz powrócić do swojego kodu wpisz: `git checkout master` (jeśli pracujesz na masterze, czyli głównej linii gita).

Podstawowe komendy git

-> `git revert numer_commit`, np. `git revert af6b84c`

Kasuje zmiany z wybranego commita (np af6b84c) poprzez dodanie nowego commita (np bg6b92d), w którym usunięty został kod wpisany po zrobieniu commita o numerze af6b84c. Nie kasuje historii.

-> `git reset numer_commit`, np. `git reset af6b84c` - kasuje commity z repozytorium, usuwa historię (nie jest zalecany). Jeśli wskażesz do skasowania trzeci commit od końca, to usunie ostatnie 3 commity. Nie skasuje jednak kodu w Twoim edytorze - jeśli jest aktualnie otwarty. Aby skasować również kod w edytorze użyj komendy:

-> `git reset numer_commit --hard`

II - Repozytorium zdalne:

Jako przykład posłużmy nam github (github.com)

Załącz konto na githubie. Trzeba będzie utworzyć nowe repozytorium zdalne - podać jego nazwę (opcjonalnie opis).

Opcja 1 - Masz już projekt w repozytorium lokalnym i chcesz dodać go do repozytorium zdalnego:

9. Github -> utwórz nowy projekt -> podaj nazwę i opis -> NIE zaznaczaj opcji "Initialize this repository with a README" -> create repository.

10. Dodanie ścieżki do repozytorium zdalnego w repozytorium lokalnym. W konsoli wpisz:

-> `git remote add origin + ścieżka repozytorium na githubie`, przykład:

`https://github.com/NazwaKontaGitHub/NazwaProjektu.git`

Razem:

-> `git remote add origin https://github.com/NazwaKontaGitHub/NazwaProjektu.git`

Ścieżkę do repozytorium znajdziesz na githubie.

11. Sprawdzanie ścieżki, na którą wysyłane są pliki do repozytorium zdalnego:

-> `git remote -v`

Podstawowe komendy git

Opcja 2 - NIE masz jeszcze projektu w żadnym repozytorium:

9. Github -> utwórz nowy projekt -> podaj nazwę i opis -> ZAZNACZ opcję "Initialize this repository with a README" -> create repository.

10. Kolonowanie repozytorium z githuba na swój komputer (zielony przycisk "clone or download" na githubie). W konsoli, w folderze, w którym chcesz umieścić swój projekt, wpisz:

-> `git clone + link do repozytorium`, np:

`git clone https://github.com/NazwaKontaGitHub/NazwaProjekt.git`

We wskazanej w konsoli lokalizacji powinien pojawić się folder z projektem

11. Zaciągnięcie zmian z githuba (z repozytorium zdalnego do lokalnego). Przełącz się na odpowiedni branch (gałąź), na który chcesz ściągnąć zmiany. Jeśli korzystasz z głównego branch-a i nie masz innych, nie musisz się przełączać.

-> `git pull origin nazwa_branch-a`, z którego ściągasz pliki. np: `git pull origin master`

Do poczki 1 i 2:

12. Dodanie plików z projektu lokalnego do repozytorium zdalnego (z komputera na github):

-> `git push -u origin nazwa_branch`, na który wysyłasz pliki. Jeśli masz tylko jeden główny branch, to komenda brzmi:

-> `git push -u origin master`

Flagę -u ustawić trzeba tylko kiedy wysyła się pliki do repozytorium zdalnego pierwszy raz, albo pierwszy raz po zrobieniu merge. Ułatwia to późniejsze pobieranie plików z repozytorium zdalnego na komputer (do repozytorium lokalnego). Każde kolejne wysłanie plików z repozytorium lokalnego na githuba to komenda:

-> `git push origin master`

13. Stworzenie osobnego brancha (osobnego odgałęzienia) - lokalnie:

-> `git branch nazwa-brach`, np: `git branch features/searchBar`

Podstawowe komendy git

14. Pokaż wszystkie branche:

-> `git branch -a`

15. Przełącz się na branch:

-> `git checkout nazwa-branch`, np: `git checkout features/searchBar` albo: `git checkout master`

16. Stworzenie brancha z jednoczesnym przełączeniem się na niego (zamiast punktów 13 i 15):

-> `git checkout -b nazwa-branch`, np: `git checkout -b features/searchBar`

17. Commit wykonany na branch:

-> `git checkout features/searchBar`

-> `git add .`

-> `git commit -m "opis commita"`

Dodanie commita z repozytorium lokalnego do zdalnego, na inną gałąź niż master:

-> `git push origin features/searchBar`

18. Usunięcie brancha. Najpierw przełącz się na branch master:

-> `git checkout master`

Następnie wpisz:

-> `git branch -d nazwa-branch`, np: `git branch -d features/searchBar` <- to usunie branch tylko jeśli jest zrobiony merge z masterem

-> `git branch -D nazwa-branch`, np: `git branch -D features/searchBar` <- to usunie branch nawet jak nie ma merge'a z masterem



Podstawowe komendy git

19. Merge branch. Najpierw przełącz się na branch master:

-> `git checkout master`

Następnie wpisz:

-> `git merge nazwa-branch`, np: `git merge features/menu`