

System Kontroli Wersji GIT

K.S

A.K

R.A

O czym będzie?

- Czym jest git, a czym jest github
- W jakim celu używamy w.w oprogramowanie
- Budowa gita
- Jak poruszać się po gicie w konsoli oraz podstawowe polecenia

Czym jest git

- Jest to rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2.
- System kontroli wersji to oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych w plikach przez wiele osób w różnym czasie.

Czym jest github

- GitHub to hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang.

Po co nam git

- Przez gita, możemy każdy krok zapisywać do repozytorium. Daje nam to możliwość cofnięcia się do poprzedniej wersji pliku lub w wyniku utraty danych ich odzyskanie
- Umożliwia również pracę wielu ludzi nad jednym projektem



nieśledzony
(untracked)



niezmodyfikowany
(unmodified)



zmodyfikowany
(modified)



wystawiony
(staged)

git add plik
git commit -m "komentarz"
lub
git commit -a -m "komentarz"



modyfikacja pliku



dodanie pliku do poczekalni
git add plik



zatwierdzenie zmian
git commit -m "komentarz"



usunięcie pliku z kontroli
wersji
git rm --cached plik



Git Clone oraz git init

- git clone – komenda która klonuje repozytorium znajdujące się na serwerze github
- git init - tworzy repozytorium lokalne, aby zrobić push na serwery github trzeba wykonać kilka poleceń
- echo "# prezentacja" >> README.md
- git init
- git add README.md
- git commit -m "first commit"
- git remote add origin <https://github.com/hausdorffspace/prezentacja.git>
- git push -u origin master

Git status

- Git status informuje nas o stanie danego repozytorium
- `git status`
- `# On branch master`
- `# Untracked files:`
- `# (use "git add <file>..." to include in what will be committed)`
- `#`
- `# README`
- nothing added to commit but untracked files present (use "git add" to track)

Git add

- Tą komendą pliki nieśledzone zaczynają być śledzone przez repozytorium.
- Istnieje również plik `.gitignore` który pozwala ignorować wybrane przez nas pliki

Git commit

- Git commit zatwierdza zmiany w repozytorium
- Do każdego commita trzeba dodać -m
- m oznacza tutaj Message czyli ,że do commita dodajemy wiadomość

Git diff oraz git log

- Komenda która z kolei pokazuje zmiany jakie zaszły w pliku
- Git log wyświetla wszystkie commity w danym repozytorium. Fajną opcją jest -p, która oprócz commitów wyświetla również zmiany jakie zaszły w plikach

Git push

Git push wypycha nam na server github zatwierdzone zmiany

- Oczywiście najpierw należy ustawić źródło (origin)

Dziękujemy za uwagę

