# **GIT** - Spis podstawowych komend

|  |
| --- |
| **git init** - Stwórz puste repozytorium w aktualnym katalogu. **git add** **<ścieżka>** - Dodaje wszystkie zmiany ze *<ścieżka>* do *stagging area*. **git commit** – Tworzy commita ze zmianami aktualnie w *staging area*. |
| **git status** – Pokazuje, które pliki się zmieniły od ostatniego commita, a które zostały dodane lub usunięte i czy są lokalnie jakieś commity do wypushowania lub pobrania. |
| **git branch <name>** - Tworzy nowy branch o nazwie *<name>*.**git checkout <name>** - *Przełącza się* na brancha *<name>* (zmieniajac pliki na dysku).**git checkout -b <name>** *- W*ykonuje *git branch <name>*, a następnie *git checkout <name>.* |
| **git reset --hard <commit>** - *Resetuje* repozytorium do wersji *<commit>,* oraz czyści *working directory* i *staging area* (czasami bezpowrotnie - zależnie od scenariusza). **git reset --mixed <commit>** - *Resetuje* repozytorium do wersji *<commit>*, pozostawia zmiany w *working directory* (jest to domyślny sposób resetowania), nie wymaga przełącznika. **git reset --soft <commit>** - Resetuje repozytorium do wersji *<commit>*, pozostawia zmiany w staging area. **git checkout <file>** - Usuwa zmiany naniesione na *śledzony* plik *<file>.* **git clean** **<file>** **-f** - Usuwa *nieśledzony* plik *<file>. A*by usunąć również katalog przełącznik *-d*, aby tylko podejrzeć co ma zostać usunięte, przełącznik *-n*. Zazwyczaj: *-df*. |
| **git rebase <target-commit>** - Szuka commita, który był wspólnym przodkiem zarówno aktualnego brancha jak i *<target-commit>*, nakłada wszystkie commity, których nie ma w *<target-commit>* na górę *<target-commit>*. **git merge <target-branch>** - Jeżeli *<target-branch>* ma niezależną historię, tworzy commita łączącego *<target-branch>* do aktualnego brancha. Jeżeli *<target-branch>* nie ma niezależnej historii, następuje połączenie *Fast Forward*, czyli dodanie commitów, bez utworzenia commita mergującego. **git rebase --interactive <commit>** - Pozwala zmieniać kolejność commitów, łączyć je ze sobą, usuwać, modyfikować, zaczynając od *<commit>* do aktualnego commita *(HEAD).* **git revert <commit>** - Tworzy nowego commita, który odwraca zmiany z *<commit>.* |
| **git push** – Wysyła lokalne commity do zdalnego repozytorium. **git fetch** - Pobiera commity do lokalnego repozytorium (do branchy *remotes/nazwa\_remota/nazwa\_brancha*). **git pull** – Wykonuje *git fetch*, następnie *git merge.* **git pull** **--rebase** – Wykonuje *git fetch*, następnie rebasuje lokalne niewypushowane commity, nakładając je na wierzch commitów pobranych ze zdalnego repozytorium. |

www.gitwarsztaty.pl