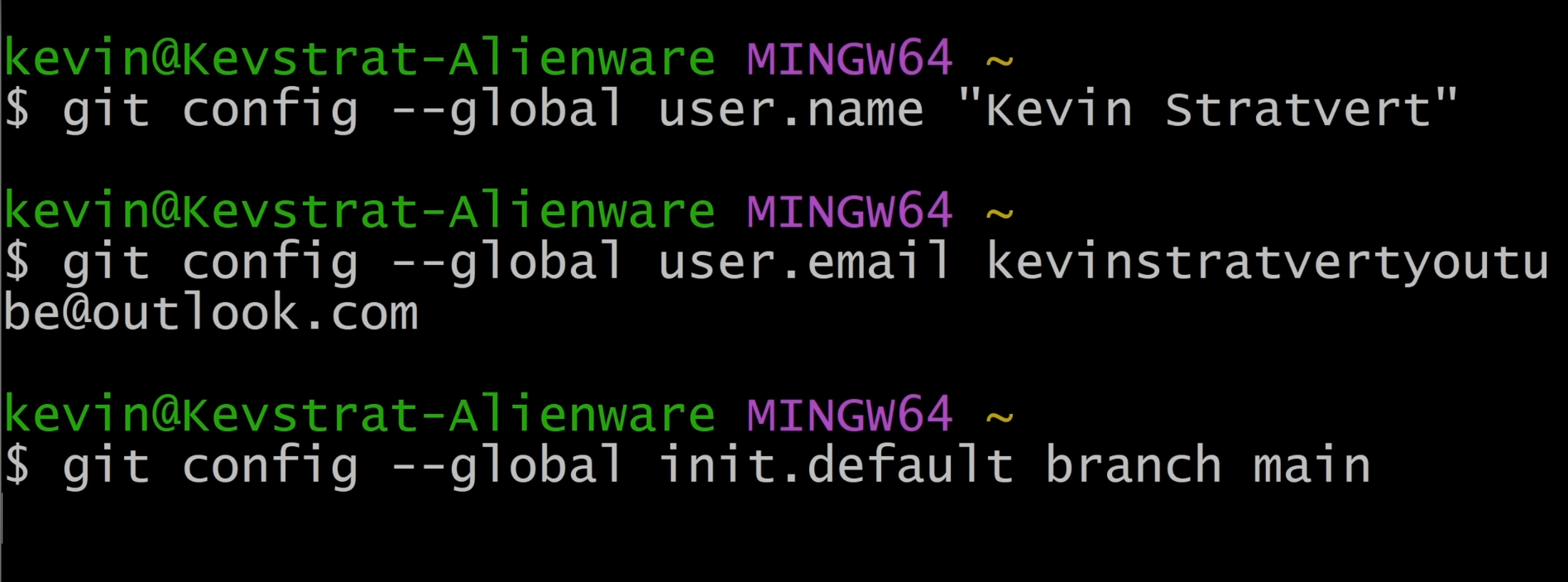
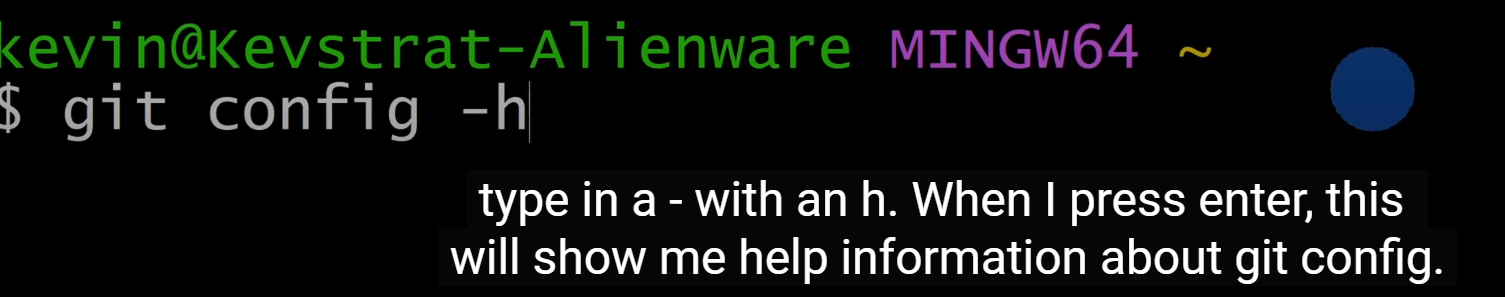
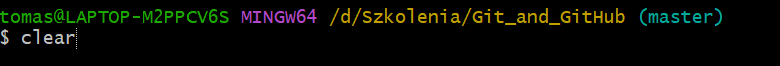
początkowa konfiguracja



pomoc o komendzie



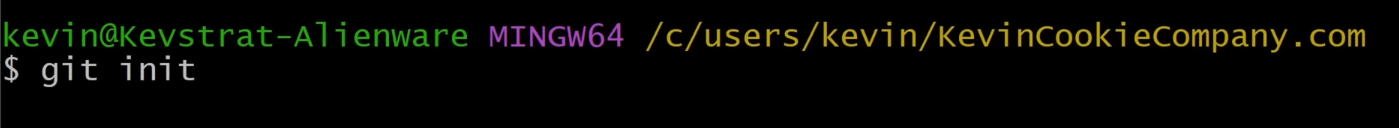
wyczyszczenie terminala git bash



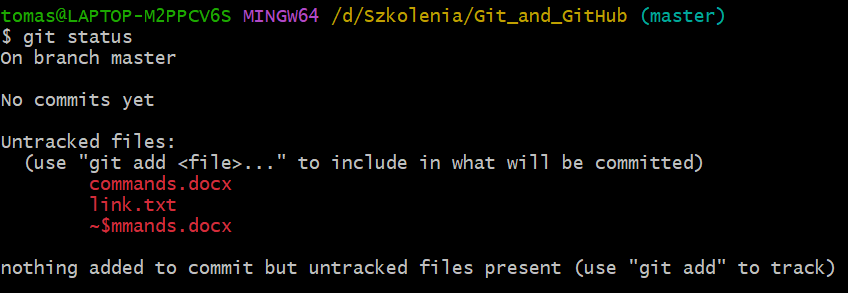
zmiana lokalizacji



to turn this into git repository (tworzenie repozytorium – tworzy też ukryty folder .git w lokalizacji z plikami repozytorium)



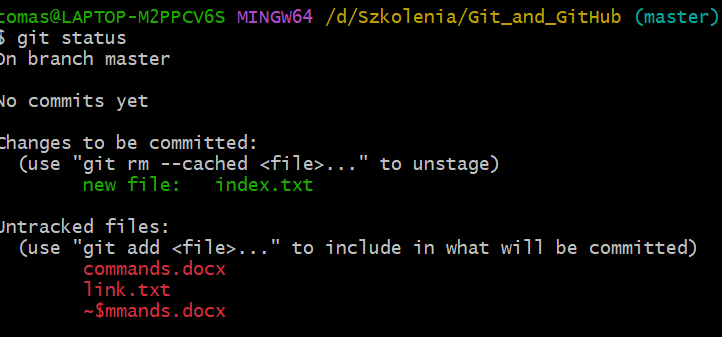
status repozytorium



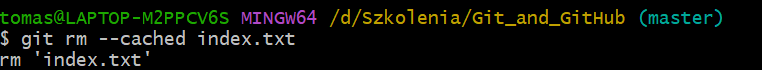
- untracked files nie są śledzone przy zmianach

rozpoczynanie śledzenia pliku





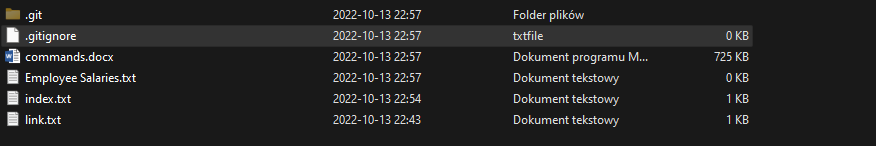
kończenie śledzenia pliku

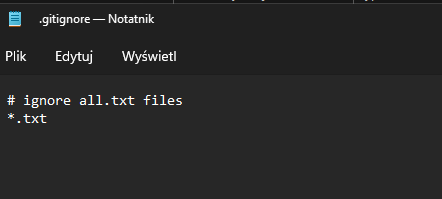


ignorowanie plików przez git

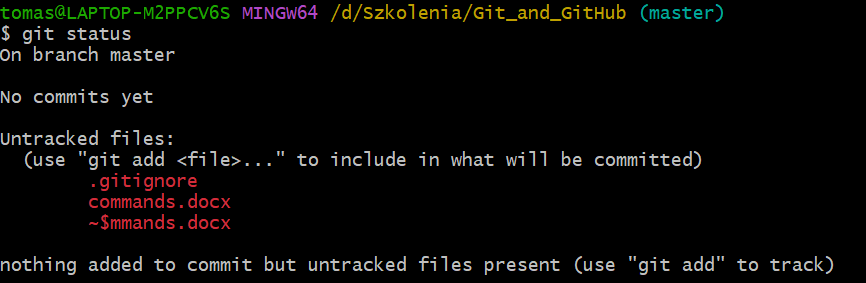
- trzeba utworzyć plik o rozszerzeniu .gitignore i wpisać tam pliki które mają nie być trackowane

- info o ignorowaniu plików <https://github.com/github/gitignore>

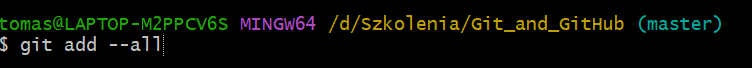


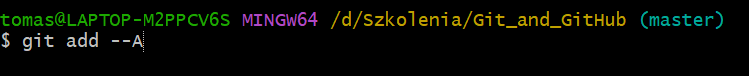


wywaliło .txt co widać w status



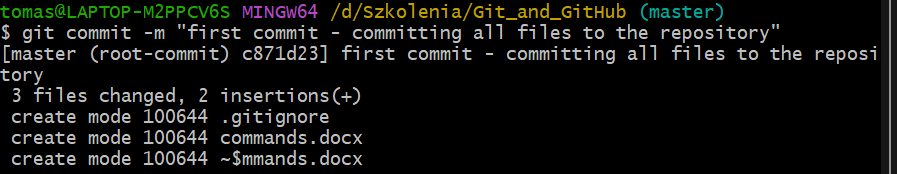
trackowanie wszystkich innych plików poza tymi ignorowanymi







COMMIT – snapshot of repository in this moment in time



Change one of the files – po sprawdzeniu statusu git powie jaki plik został zmieniony

git diff – powie jakie są dokładnie zmiany w pliku który uległ zmianie

git add „filename” – przechodzi do fazy staging, czyli czekania aż zrobię commit (working files > staging > commit)

git restore --staged „filename” – usuwanie pliku z fazy staging

**git commit -a - m „updated text to free range”** - pomija faze staging i od razu rovi commit

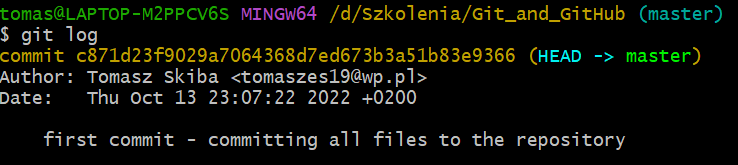
**git rm „filename”** - (trzeba z cudzysłowem) usuwa plik, i jeżeli był trackowany to będzie w statusie

**git restore „filename”** - (trzeba z cudzysłowem) przywraca plik

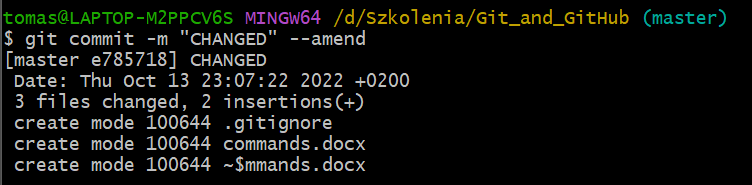
**git mv „filename.roz” „new\_filename.roz” -** zmienia nazwę pliku - mv to move

**git commit -m „changed the file name”** - od razu robi commit z komentarzem

**git log** - pokazuje wszystkie commity



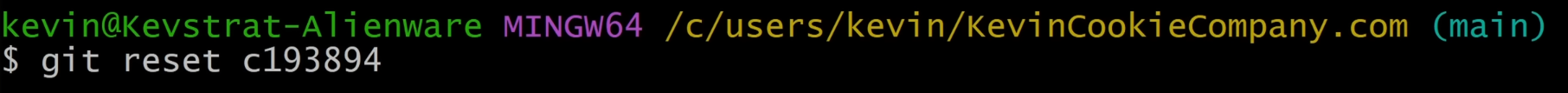
**git commit -m „komentarz” --amend** - poprawia jeden z commitów



**git log -p** - pokazuje wszystkie zmiany dokonane przez commit

**git log --online** - online log

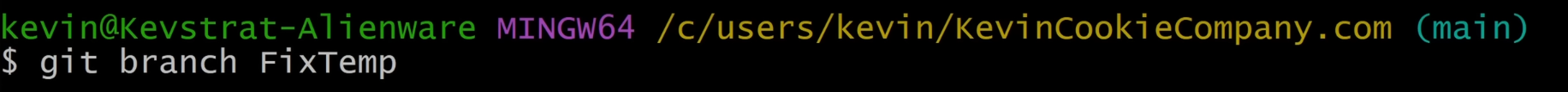
**git reset „commit number”** - resetuje commit o tym numerze



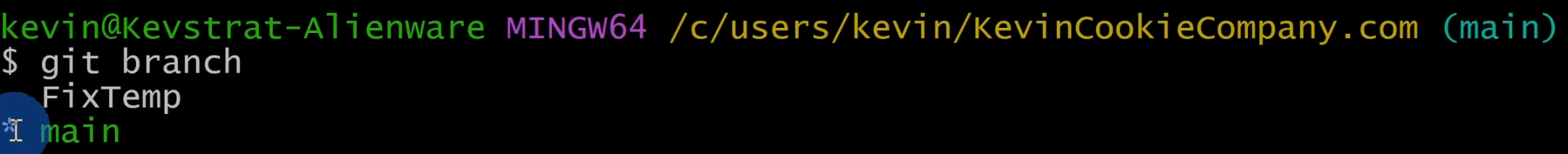
**git rebase -i --root** - modyfikacja czegoś z history book

żeby wyjść z view trzeba nacisnąć klawisz dwukropek (:)

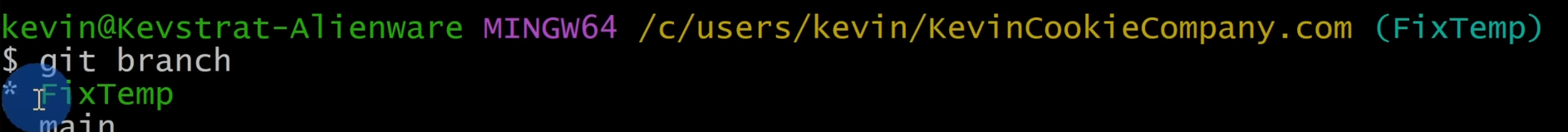
**git branch** Nazwa - Tworzenie nowej „branch” - czyli kopii głównej branch, żeby na niej pracować (np. naprawiać buga)



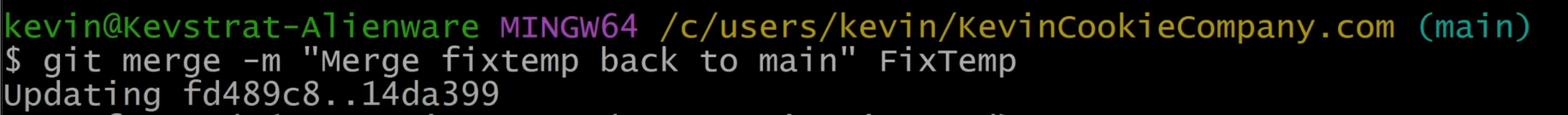
**git branch** - bez nazwy to po prostu sprawdzenie ile jest branch



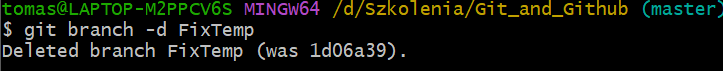
**git switch Nazwa** - zmiana na podaną branch



**git merge -m „komentarz” -** integrowanie zmian jednej branch z inną



**git branch -d NazwaBranch** - kasowanie branch



Co jeżeli w czasie w którym zmieniasz coś w dodatkowej branch, a coś się zmieni też w głównej (np. ktoś pracował na niej)

**git swithc -c NazwaBranch** - jednoczesne utworzenie i przejście do nowej branch

