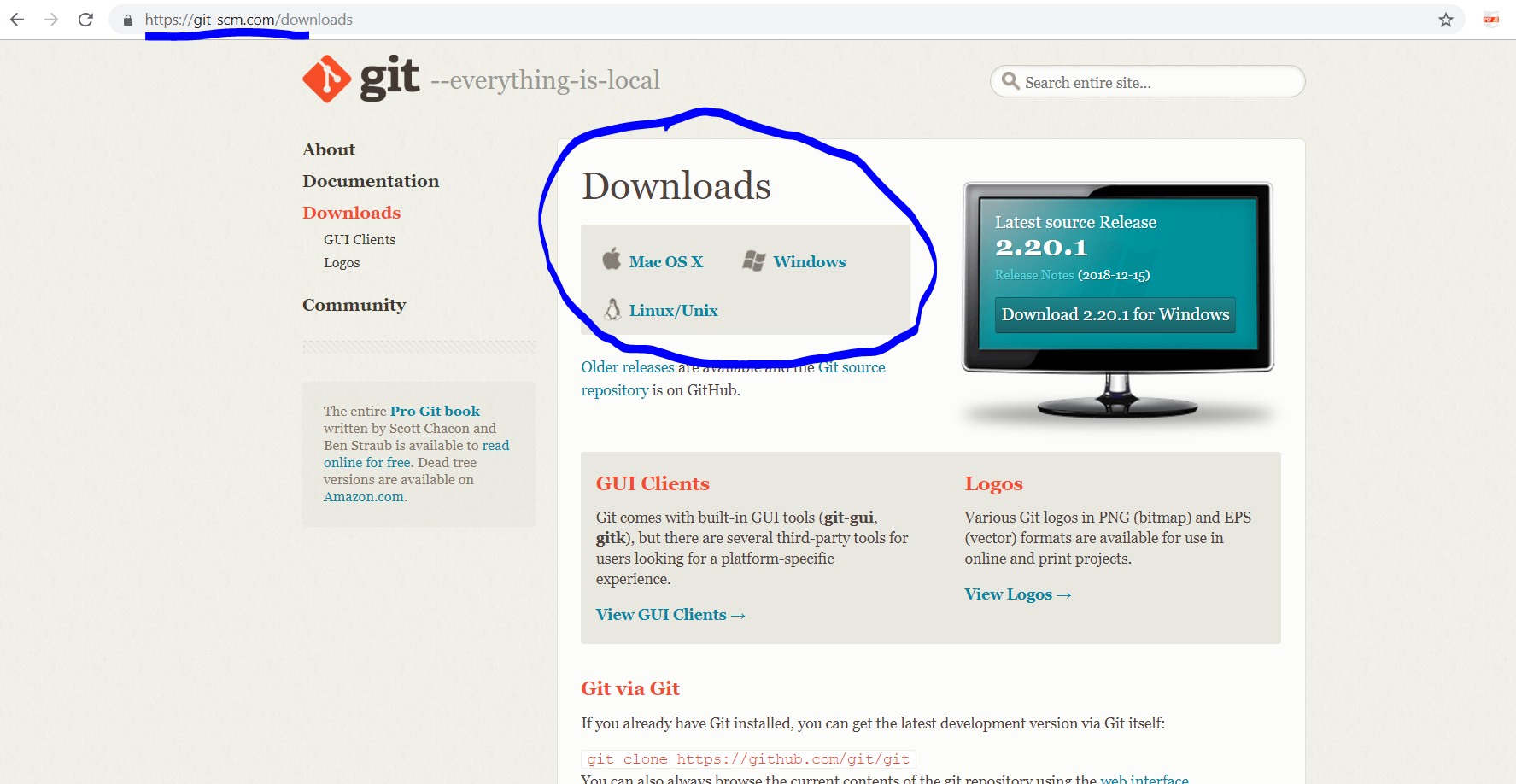
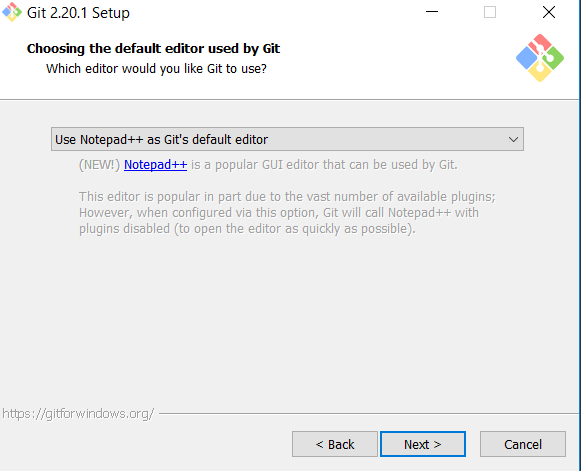
**GIT – HOW TO**

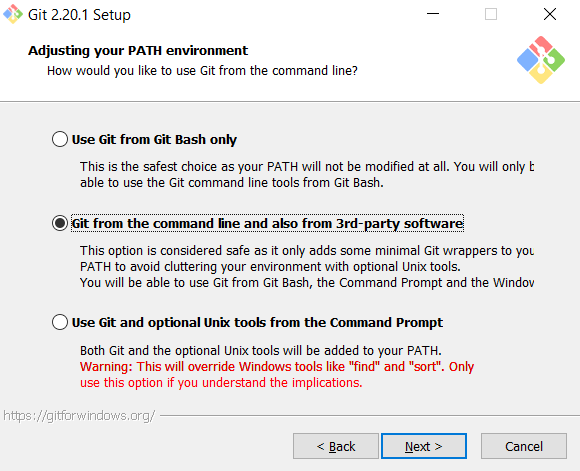
1. Download und Installation
2. Download



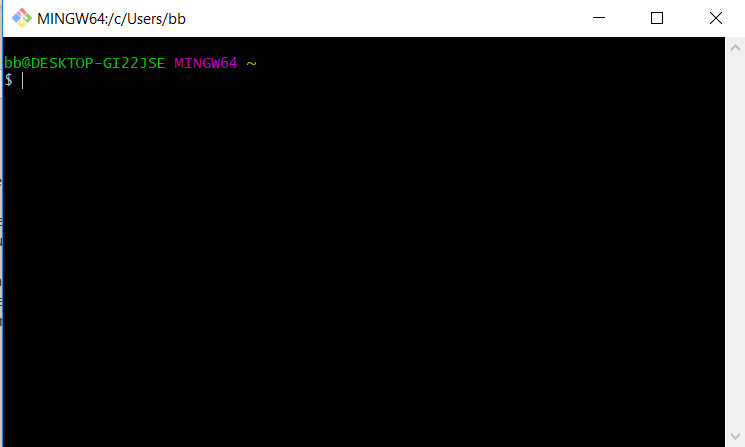
1. Hier Lieblingstexteditor auswählen, der öffnet sich dann, wenn es einen Merge Konflikt oder Ähnliches gibt und man den beheben muss. Notepad++ ist einfacher zu bedienen als Vim.



1. Bei Windows: Bei allen dreien installiert man die Git Bash, aber die Entscheidung beeinflusst, ob man Git Befehle in der Command Prompt (Cmd, „Windows Terminal“) eingeben kann und ob man die Befehle in der Cmd abändert.



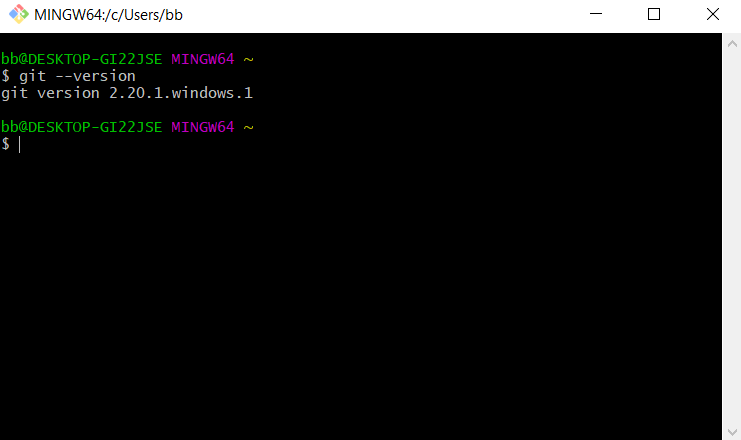
1. Die weiteren Fenster kann man auch einfach auf default lassen.
2. Nun hat man einen Terminal, in dem man etliche Unix-Befehle verwenden kann und in dem man die Git-Befehle eingibt.



1. Git anwenden

1. Schauen, ob man Git korrekt installiert hat und verwenden kann:

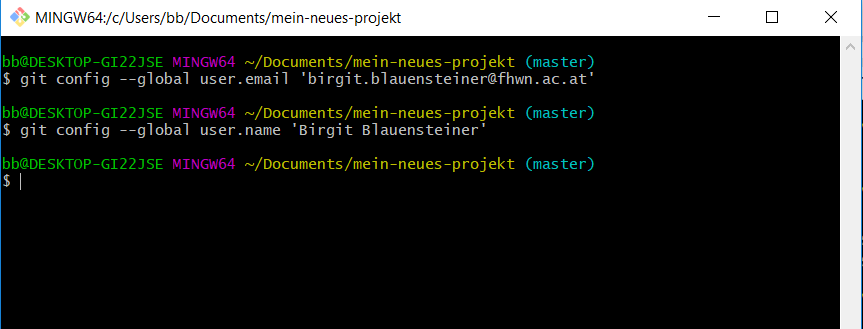
$ git --version



1. Konfiguration von Git

$ git config --global user.email ‘xyz@fhwn.ac.at’

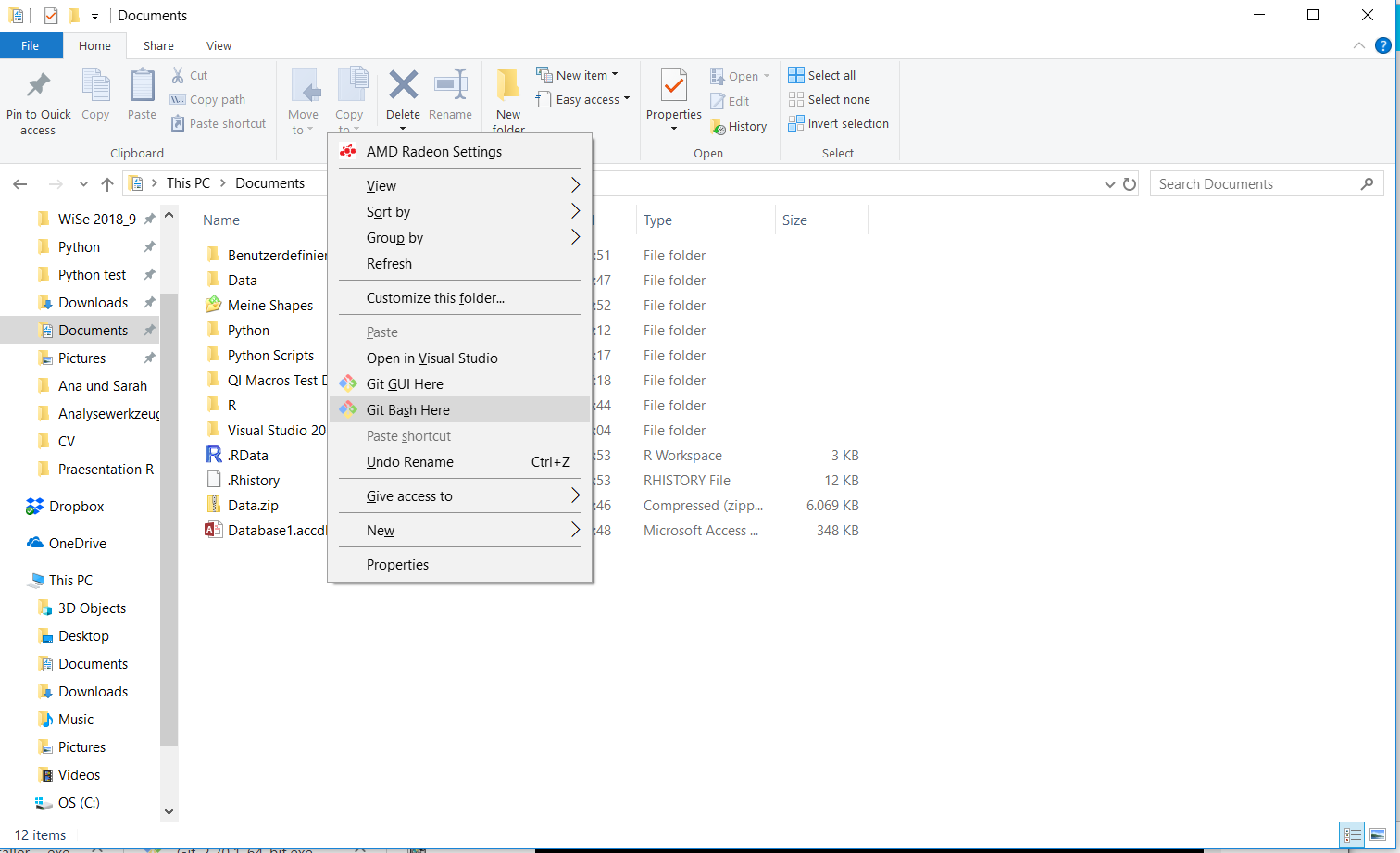
$ git config --global user.name ‘Birgit Blauensteiner’



1. print working directory – sagt mir, wo ich bin:

$ pwd

1. In das Verzeichnis wechseln, in dem man das neue Projekt anlegen will. Einfachste Lösung ist, in den Ordner zu gehen und Git-Bash dort erst öffnen:



1. Neues Verzeichnis erstellen mit dem Programm mkdir (steht für make directory):

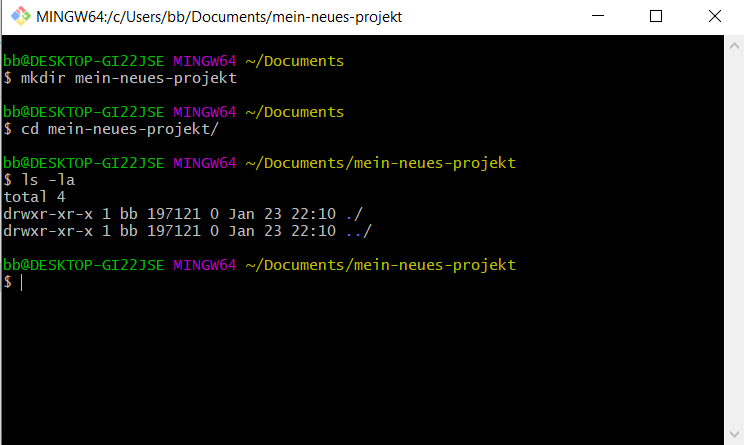
$ mkdir mein-neues-projekt

1. In das neue Verzeichnis wechseln mit dem Programm cd (steht für change directory):

$ cd mein-neues-projekt

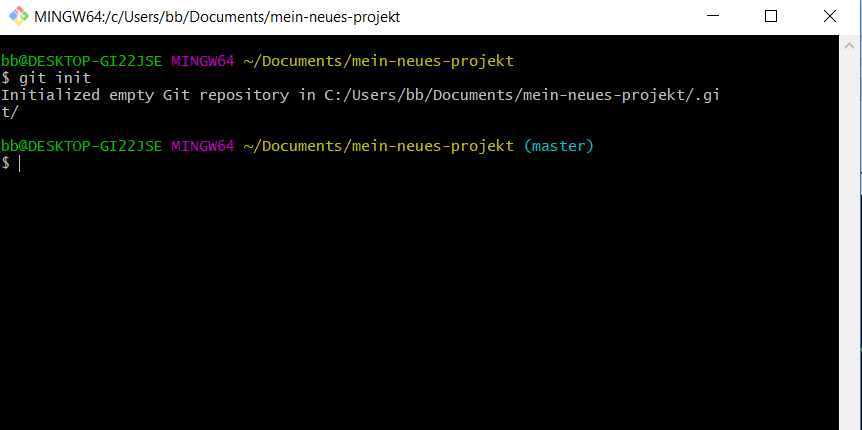
1. Verzeichnisinhalt ansehen mit dem Programm ls (steht für list). Die Optionen l und a geben zusätzliche Informationen aus (Dateiattribute und versteckte Dateien):

$ ls -la



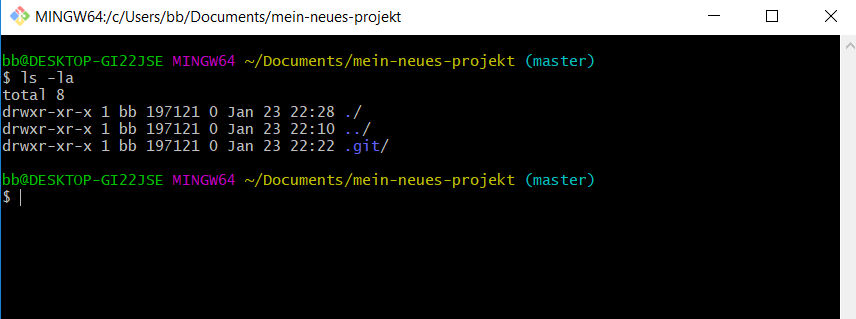
1. Git Repository erstellen

$ git init



1. Schauen, ob die Git Dateien, die von Git erstellt werden, wenn man ein Git Repository erstellt, da sind:

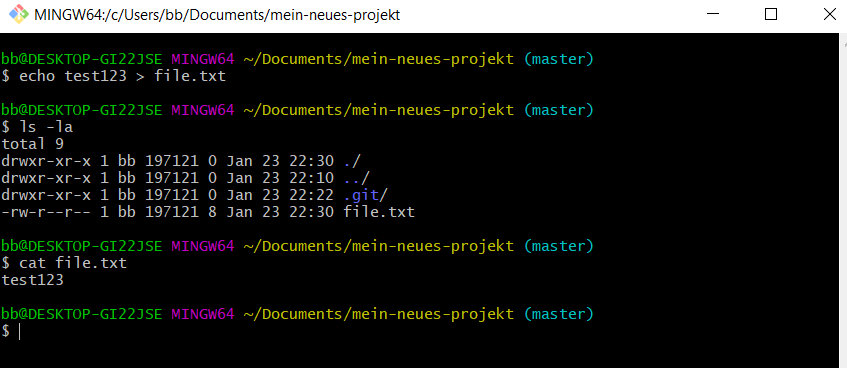
$ ls -la



1. Eine Textdatei erstellen namens file.txt mit dem Inhalt „test123“, optional kann man nachsehen, ob die Datei da ist mit dem Programm ls und was drinnen steht mit dem Programm cat. Wenn man eine Datei laden will (keine .txt, zB .pdf oder .ipynb), dann hier 1 und 2 nachsehen und weitermachen.

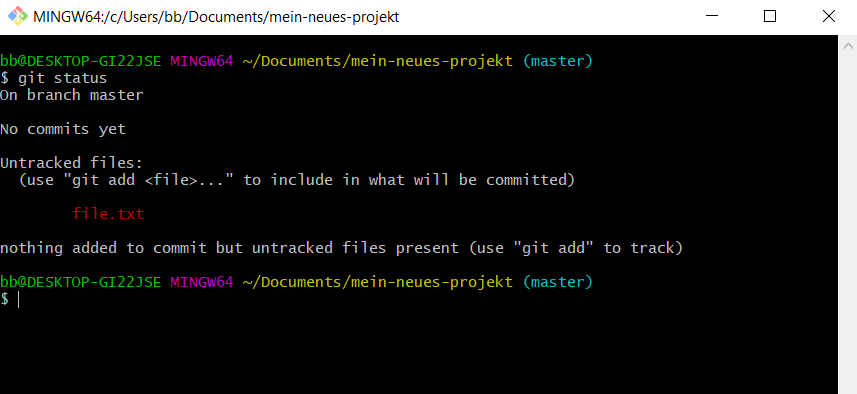
$ echo test123 > file.txt

$ cat file.txt



1. Status des Git Repositories anschauen mit dem Programm git

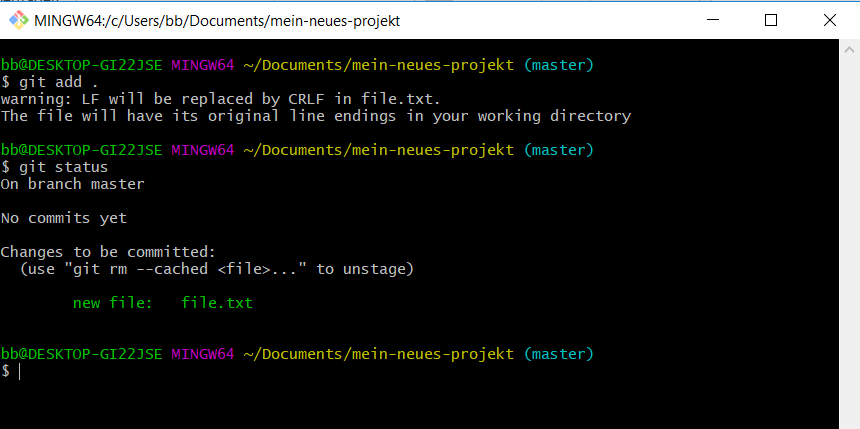
$ git status



1. Alle unvorgemerkten Dateien zum Commit vormerken. Der . steht für alle Dateien im aktuellen Verzeichnis und darunter. Es werden alle Dateien und alle Dateien in allen Subverzeichnissen vorgemerkt. Mit git status sieht man, dass die Datei jetzt vorgemerkt ist, aber noch nicht committed.

$ git add .

$ git status



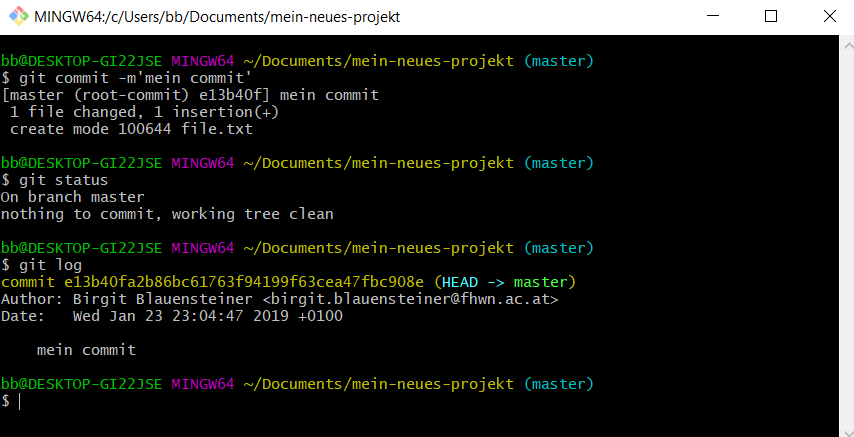
1. Vorgemerkte Änderungen ins Git Repository committen. Optional: Mit git status überprüfen, ob alles committed wurde. Mit git log kann man sich die Zusammenfassung aller commits anschauen. Um log zu schließen, q drücken.

$ git commit -m ’mein commit’

$ git status

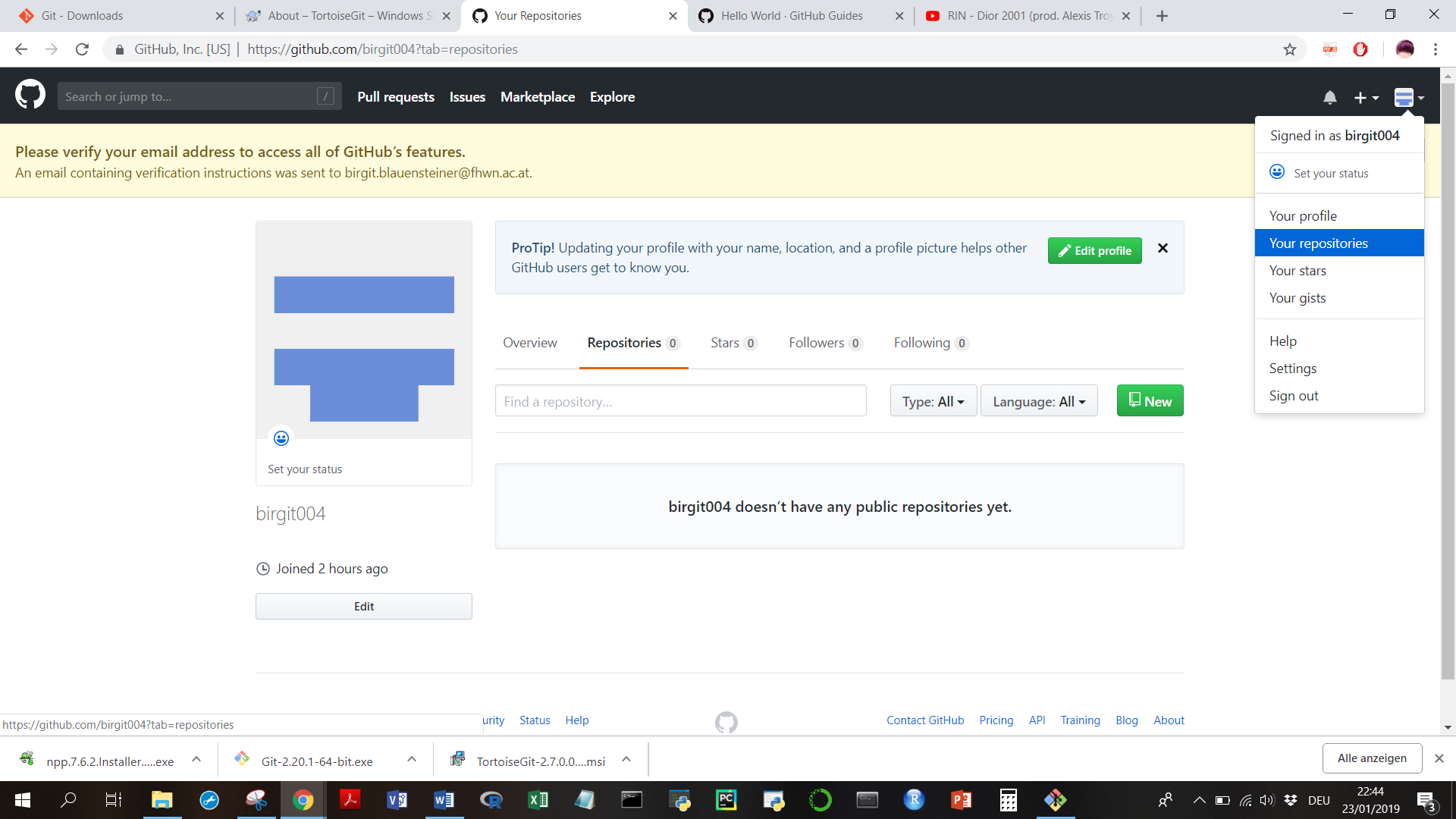
$ git log

q

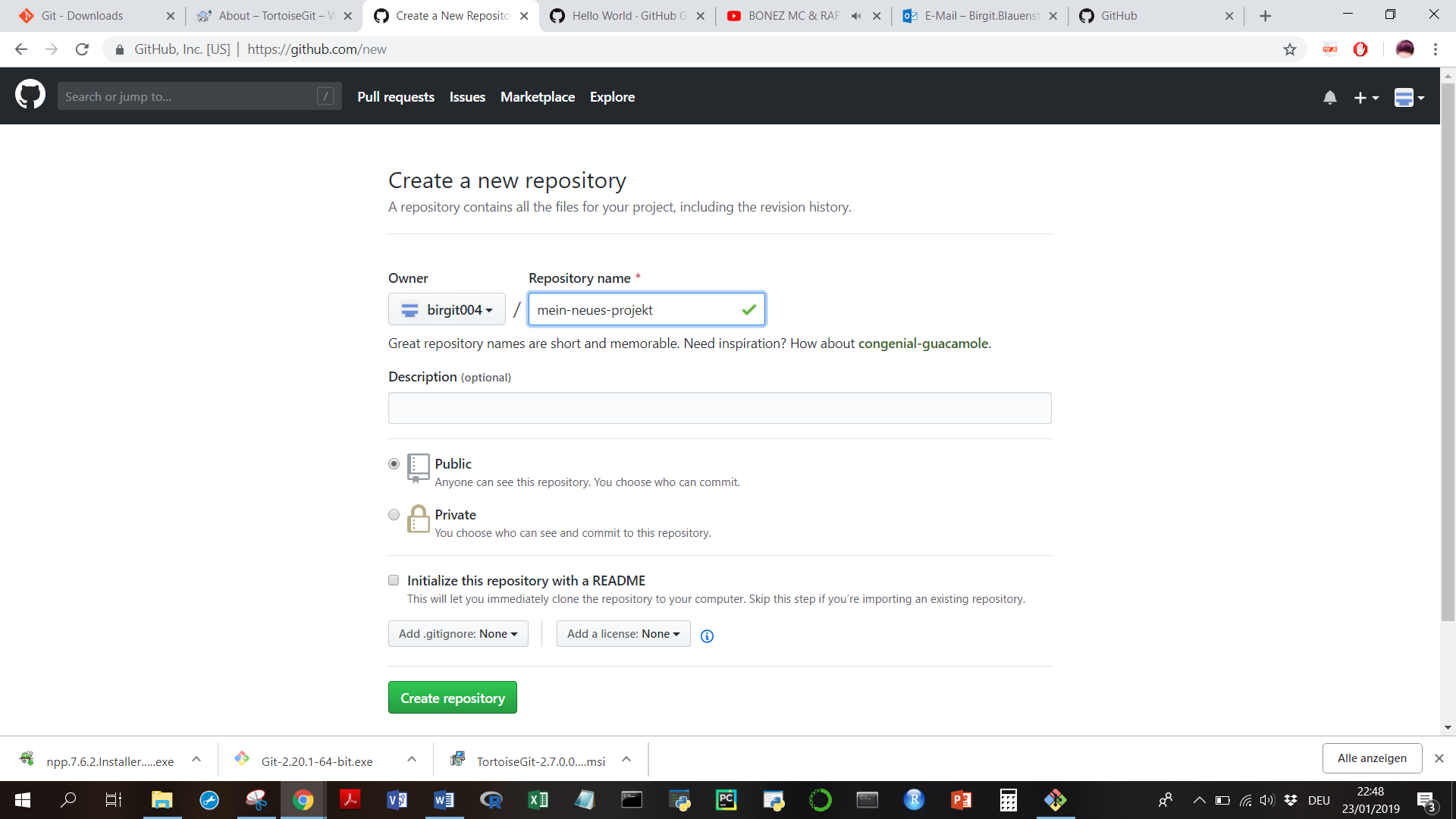


1. Bei Github anmelden und leeres Repository mit Default Einstellungen erstellen

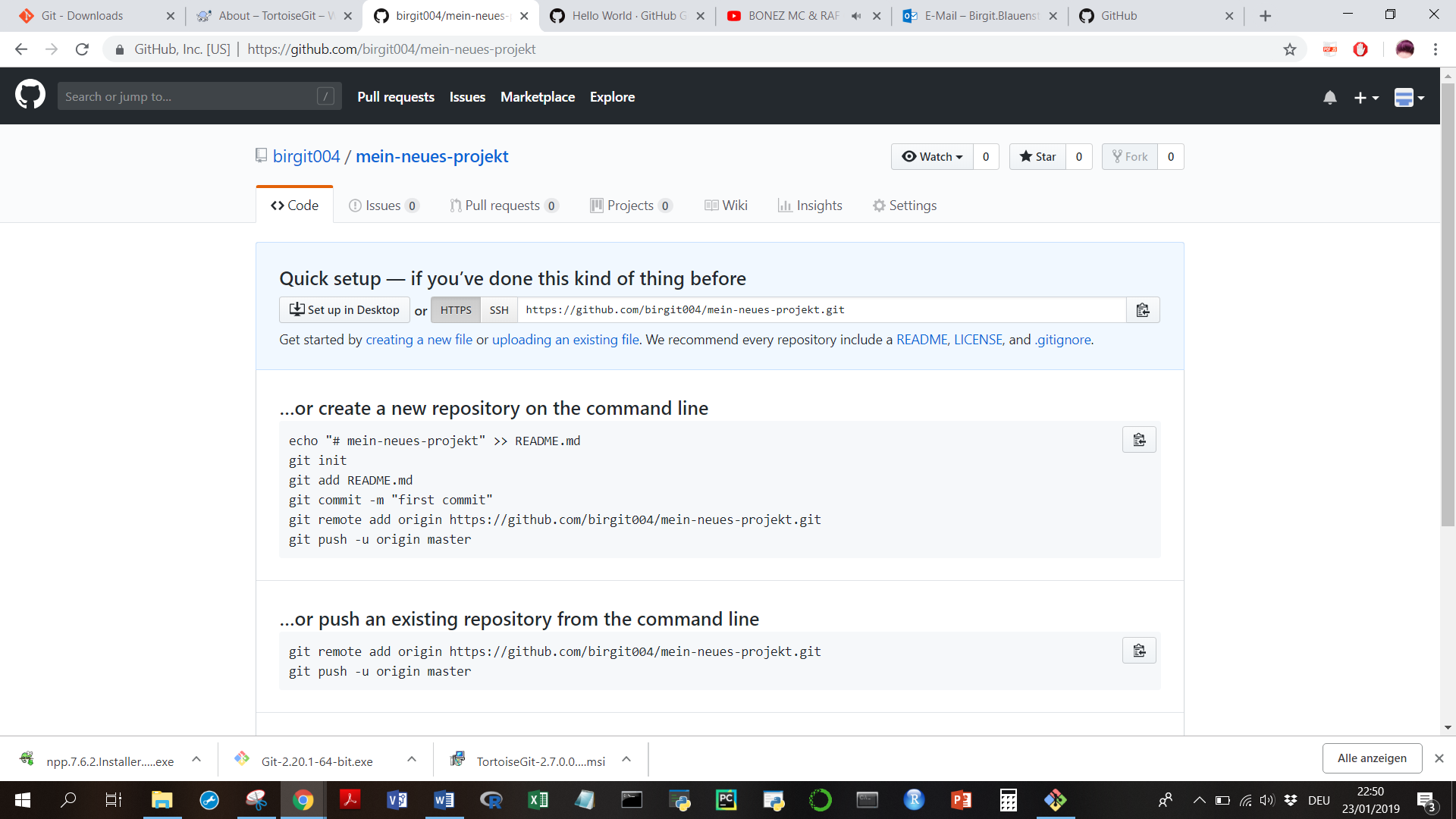
1. Account erstellen, oben rechts auf Your repositories klicken und dann in der Mitte auf New



1. Repository erstellen mit den default Einstellungen und einem selbst gewählten Namen:



1. Git Repo erfolgreich erstellt:



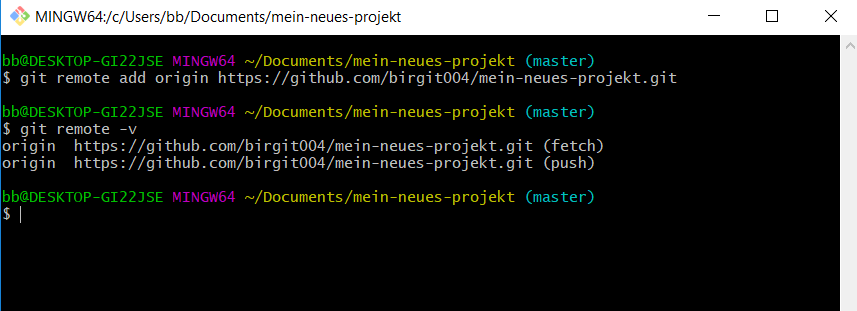
1. Lokales Git Repository mit dem remote Repository (=Github) verbinden:

1. Remote hinzufügen

$ git remote add origin https://github.com/birgit004/mein-neues-projekt.git

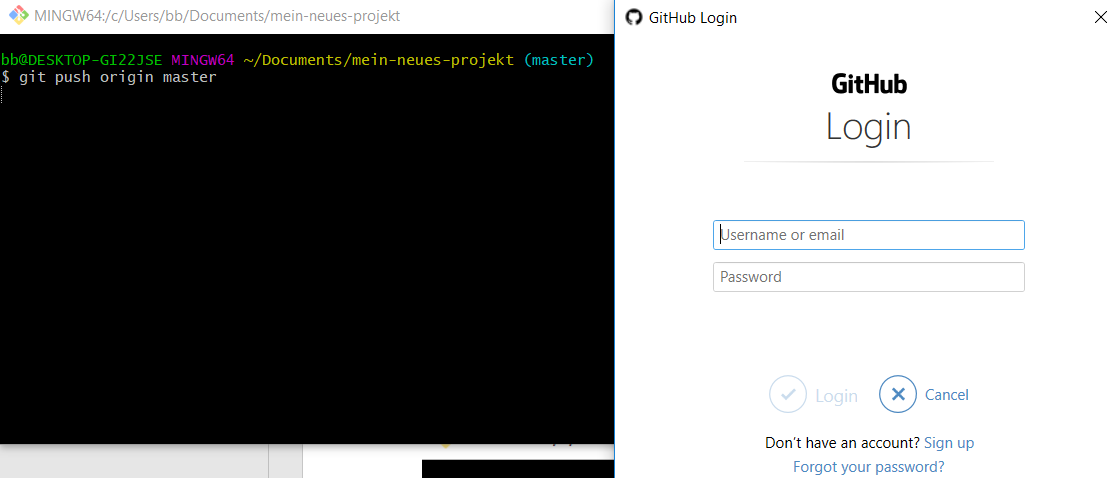
1. Überprüfen, ob erfolgreich hinzugefügt:

$ git remote -v

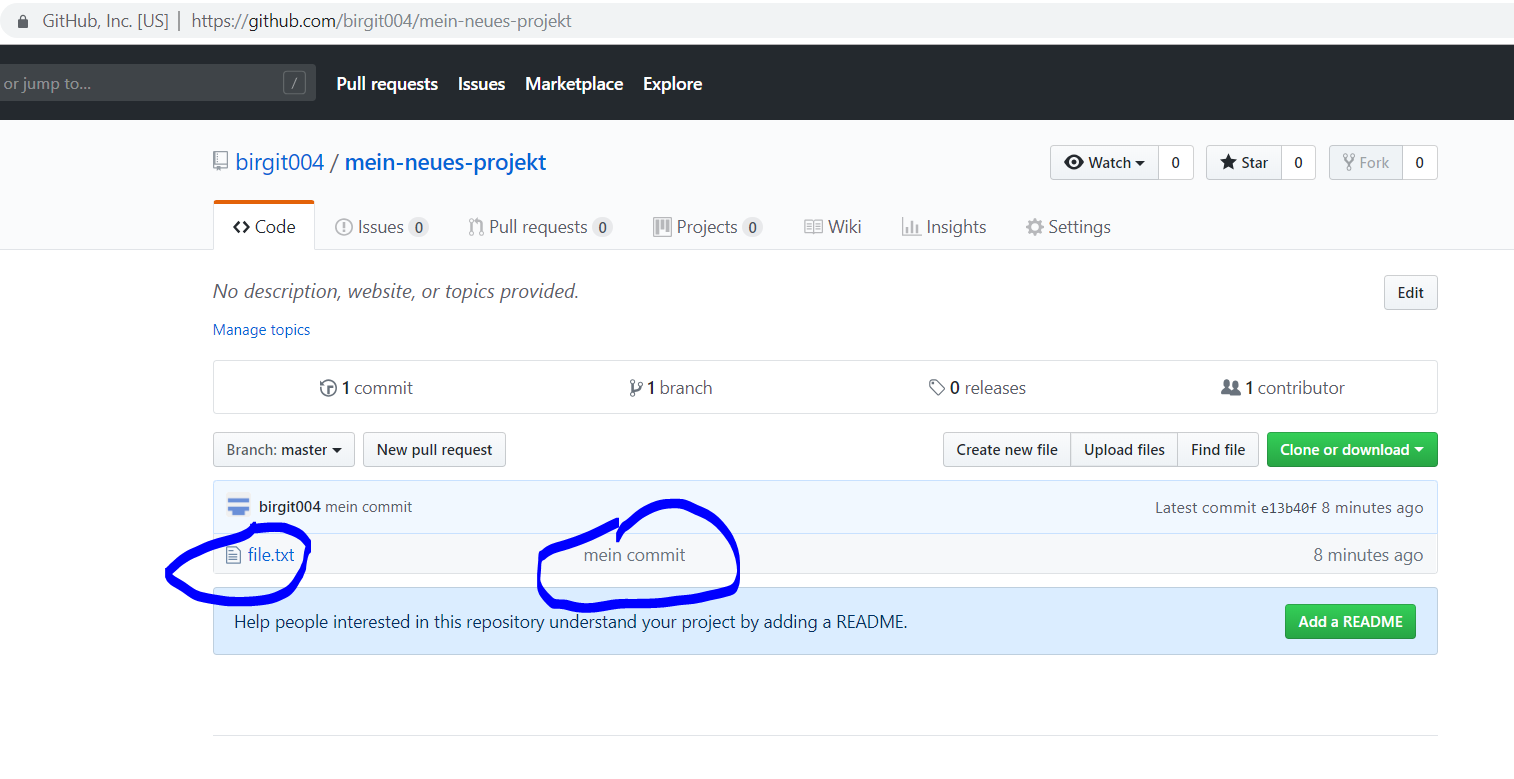


1. Änderungen vom lokalen Repository in das remote Repository auf Github pushen. Es dürfen nur Berechtigte auf das remote Repository pushen, deshalb werden die Zugangsdaten abgefragt. Zugangsdaten eingeben

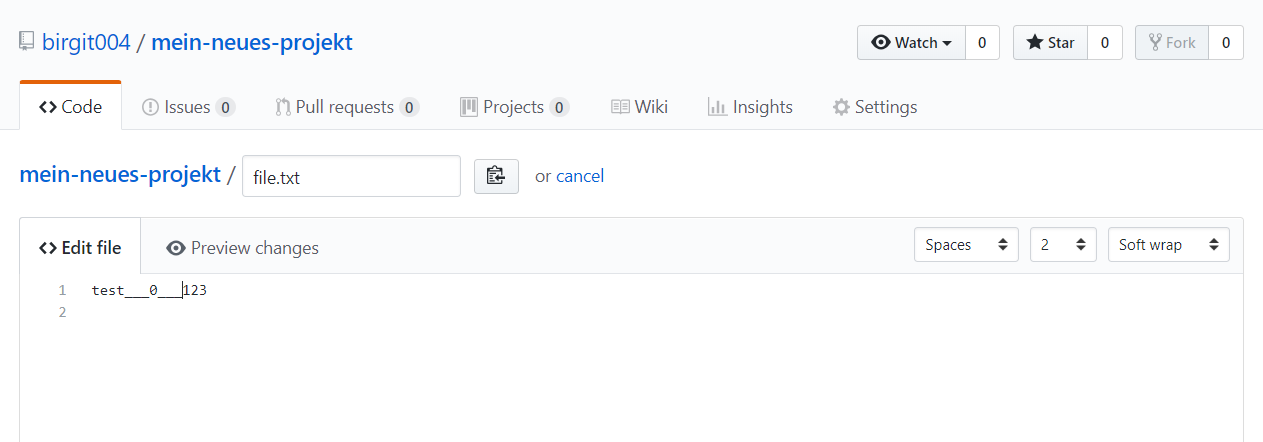
$ git push origin master



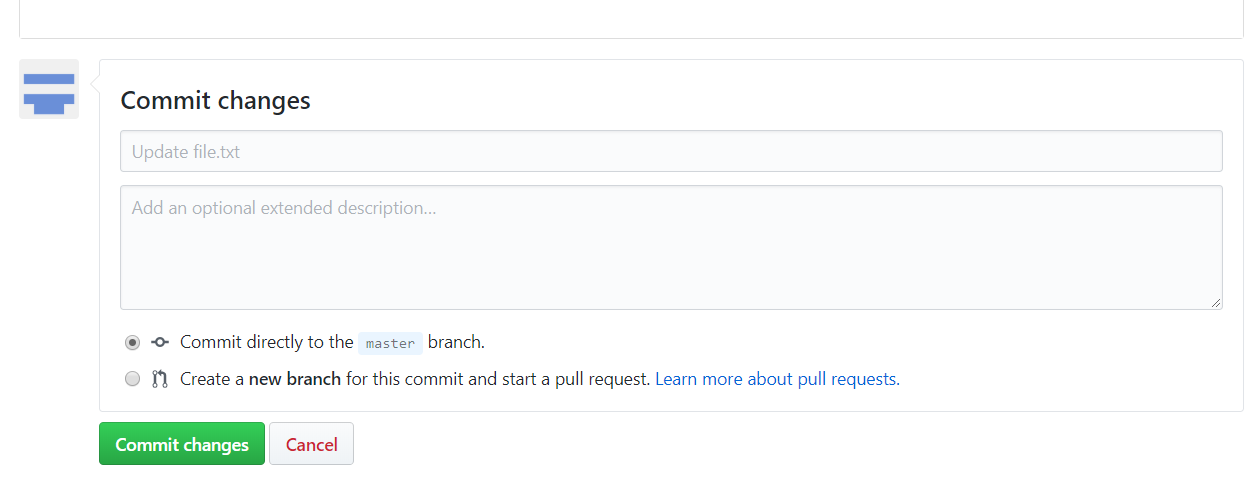
1. Nach dem Push kann auf dem Github Repository die gepushte Datei und den gepushten Commit sehen



1. Die Datei bearbeiten und mittendrin was einfügen. Wir imitieren den Fall, dass ein\_e andere Developer\_in jetzt unsere Datei unsere Datei bearbeitet und mittendrin irgendwas einfügt oder unseren Code ausbessert.

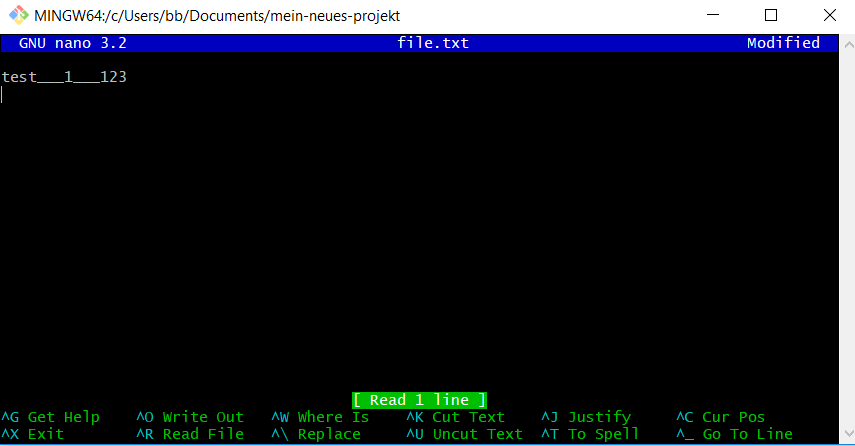


1. Runterscrollen und Changes committen:

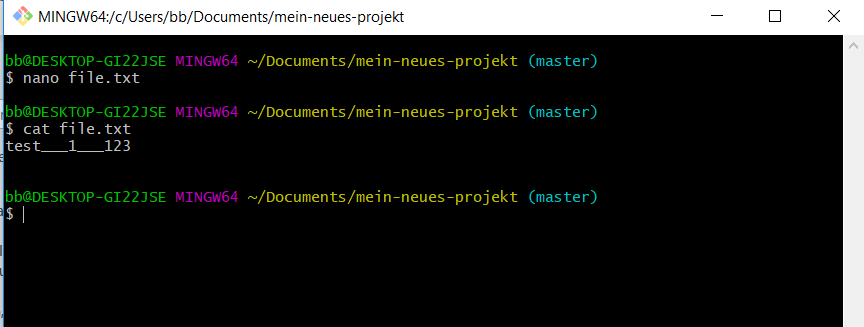


1. Mit dem Programm nano oder irgendeinem anderen Texteditor den Text in unserer lokalen Datei file.txt ändern von test123 auf test\_\_1\_\_123. Wir imitieren damit den Fall, dass wir einen Fehler in unserem Code bemerkt haben und diesen ausgebessert haben.

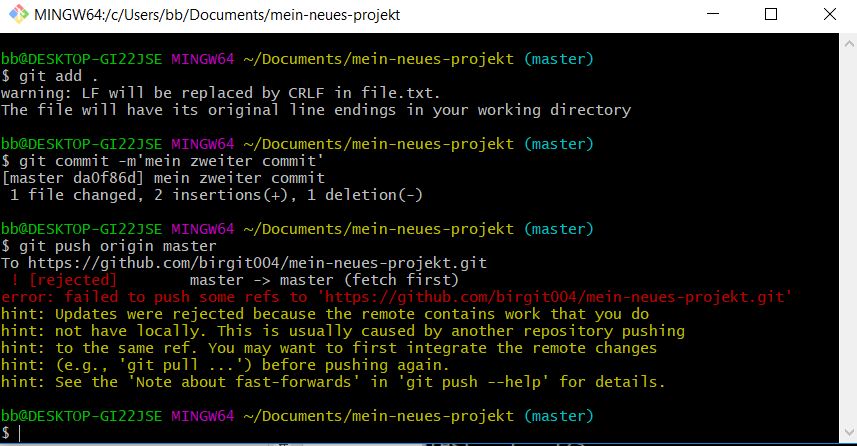
$ nano file.txt



1. Strg+X, dann Y, dann Enter, um zu speichern und Programm zu schließen.

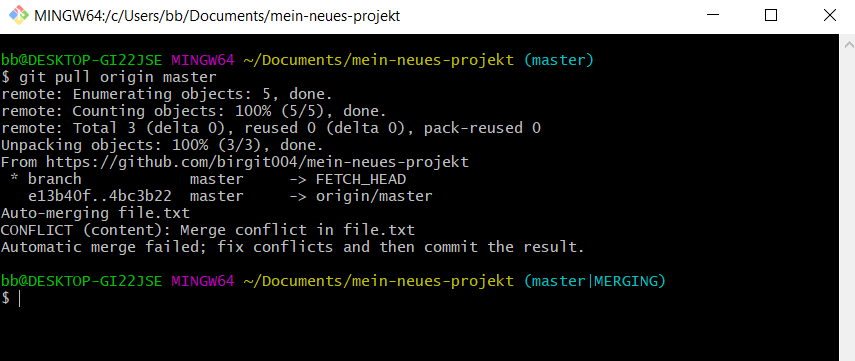


1. Adden, committen und pushen. Der Push failed, weil am remote repository was geändert wurde. Wir müssen daher zuerst die Änderungen vom remote repository in unser lokales Repository pullen.

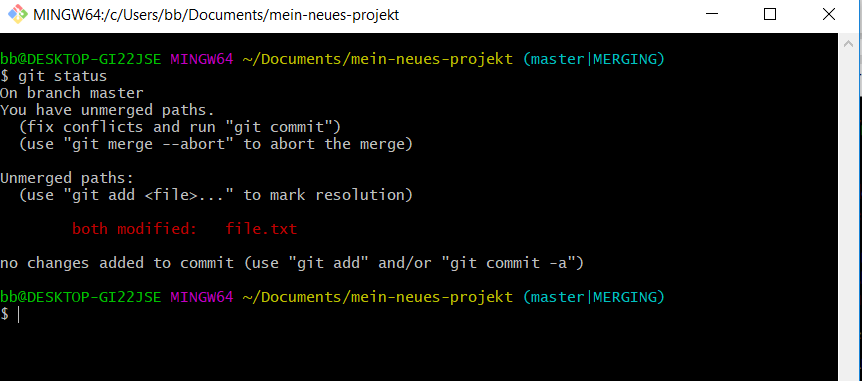


1. Die Änderungen vom remote repository ins lokale repository pullen.

$ git pull origin master

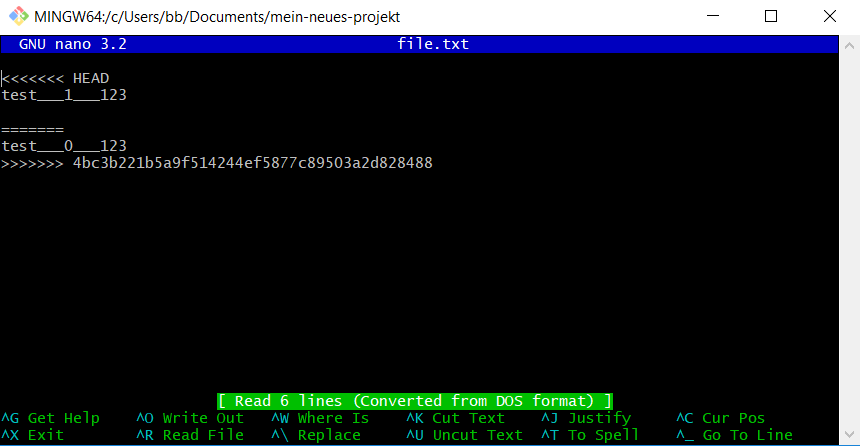


1. Den Status vom lokalen Git Repository anzeigen mit git status:

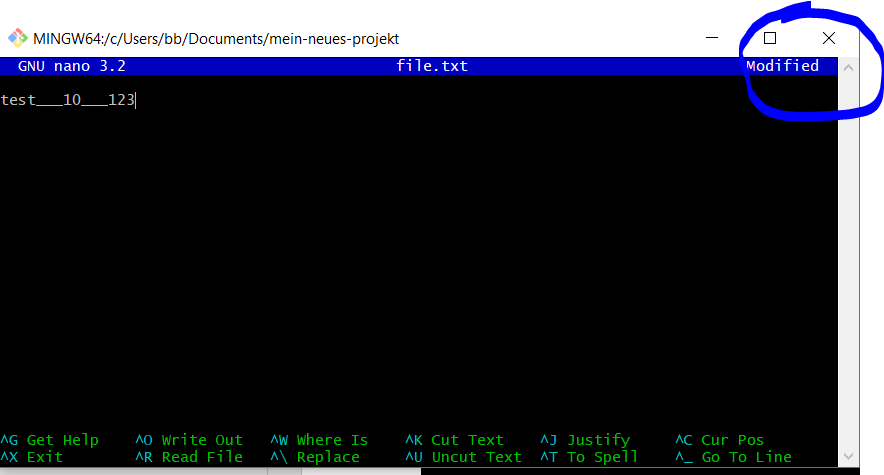


1. File.txt mit irgendeinem Texteditor öffnen, zB mit dem Programm nano. Die zwei widersprüchlichen Versionen tauchen auf.

$ nano file.txt



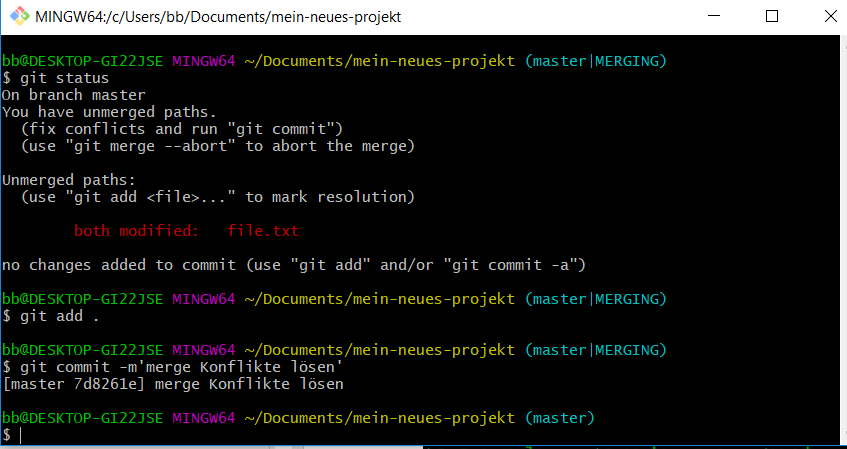
1. Den Inhalt so ändern, wie man ihn haben möchte, das heißt, Teile rauslöschen oder bearbeiten. Dann Strg+X, mit Y speichern und Enter.



1. Änderungen adden und committen:

$ git add .

$ git commit -m ‘merge Konflikte lösen‘

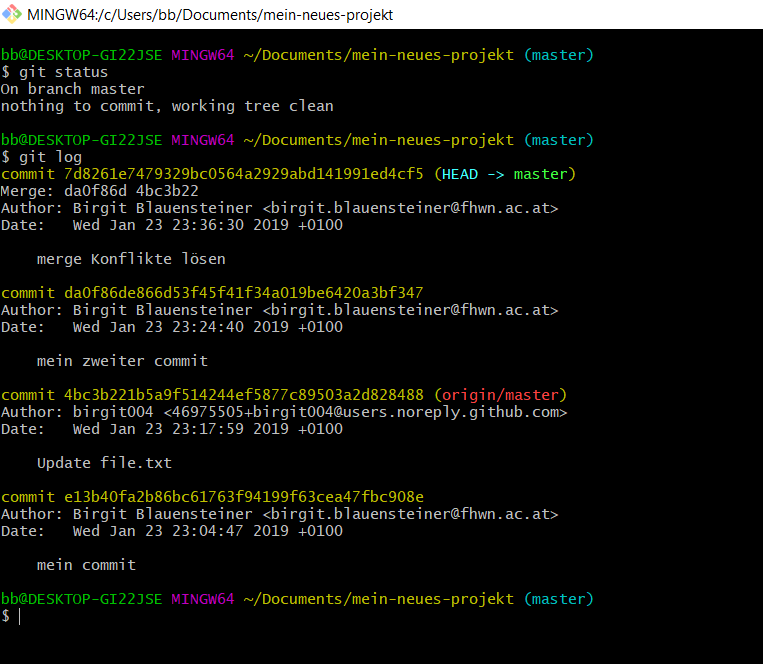


1. Den Status ausgeben und die Historie. Um log zu schließen, q drücken.

$ git status

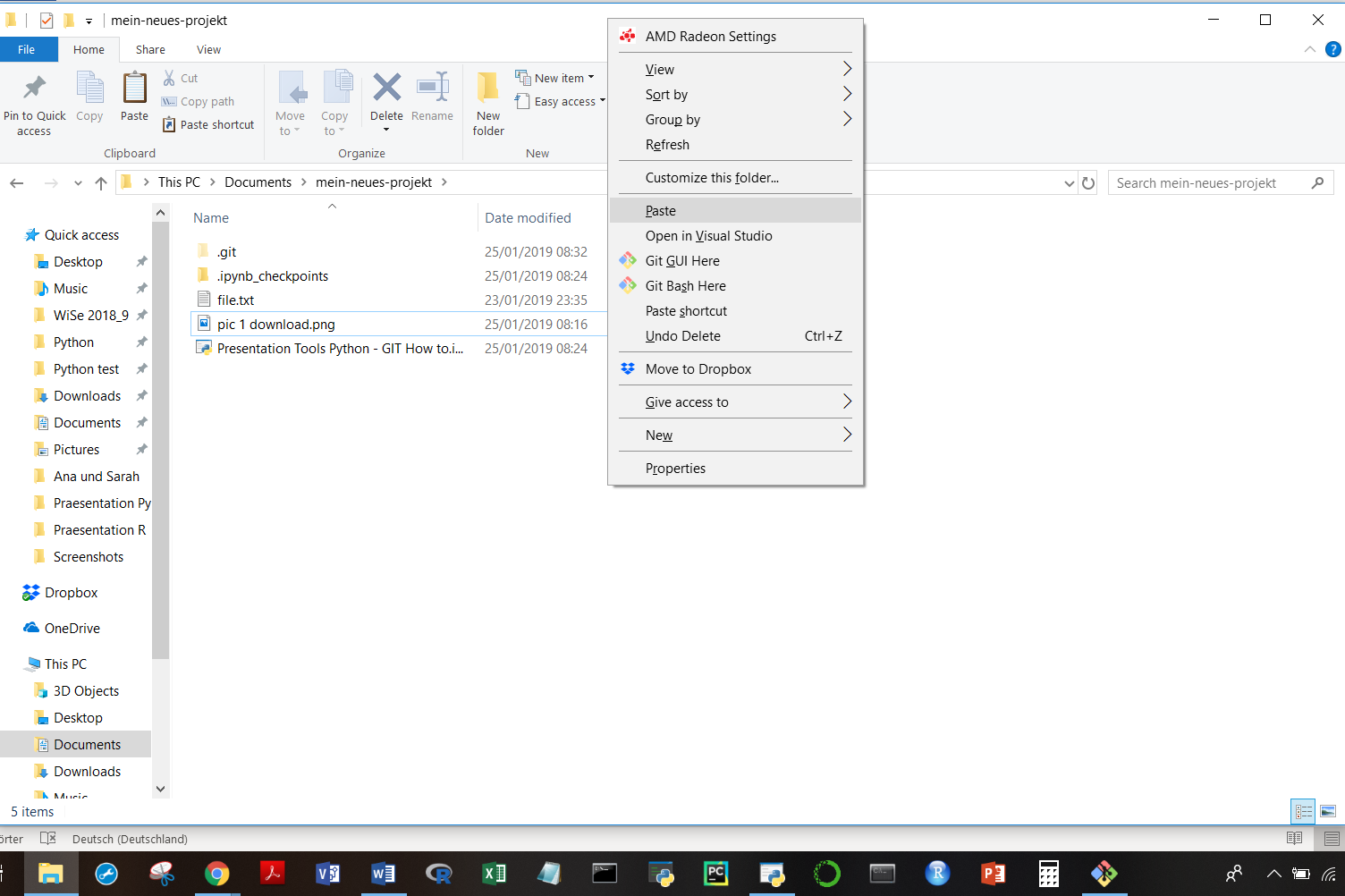
$ git log

q

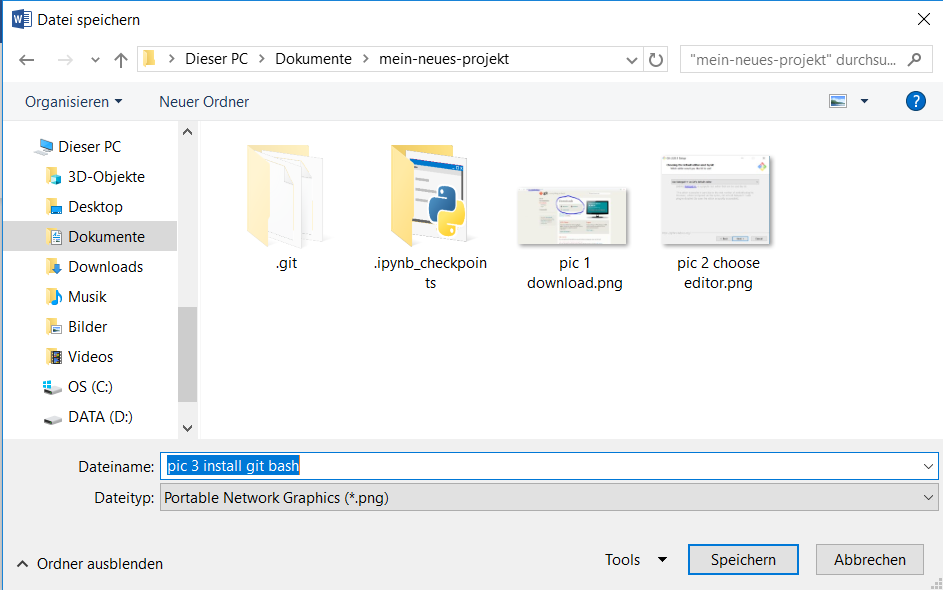


1. Upload von Dateien von einem lokalen Git Repository
2. Von einem lokalen Git Repository: Nachdem man ein lokales Repository erstellt hat (siehe Abschnitte II. 5, 6, 7 und 8), in den lokalen Git Ordner gehen und Datei dort einfügen bzw. Datei, die man erstellt hat, dort abspeichern.

* einfügen:



* bzw. abspeichern:



1. Dann macht man bei II. 11, 12, 13 und III. 3 weiter, kurz zusammengefasst:

$ git status

$ git add .

$ git status

$ git commit -m ’mein commit’

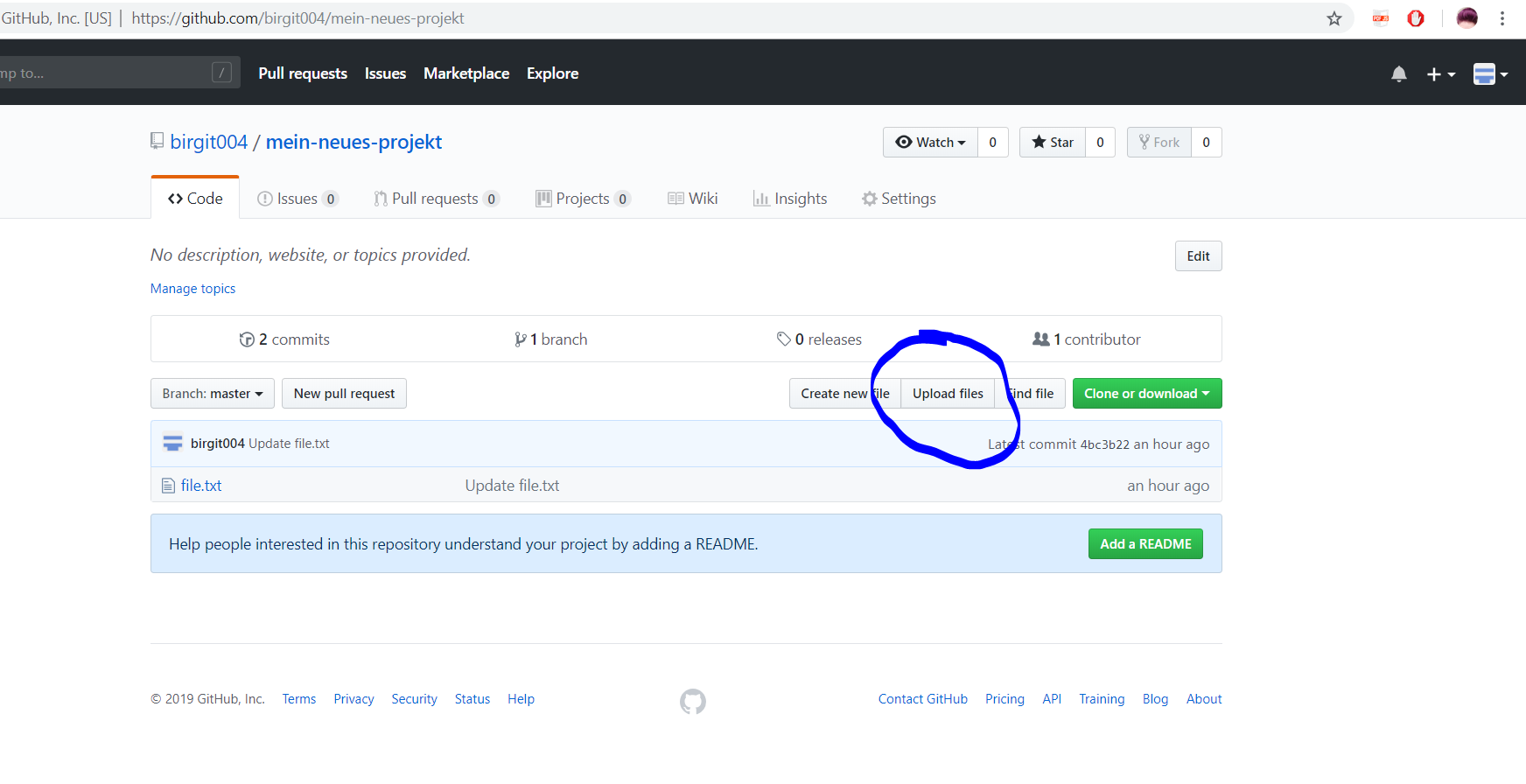
$ git status

$ git log

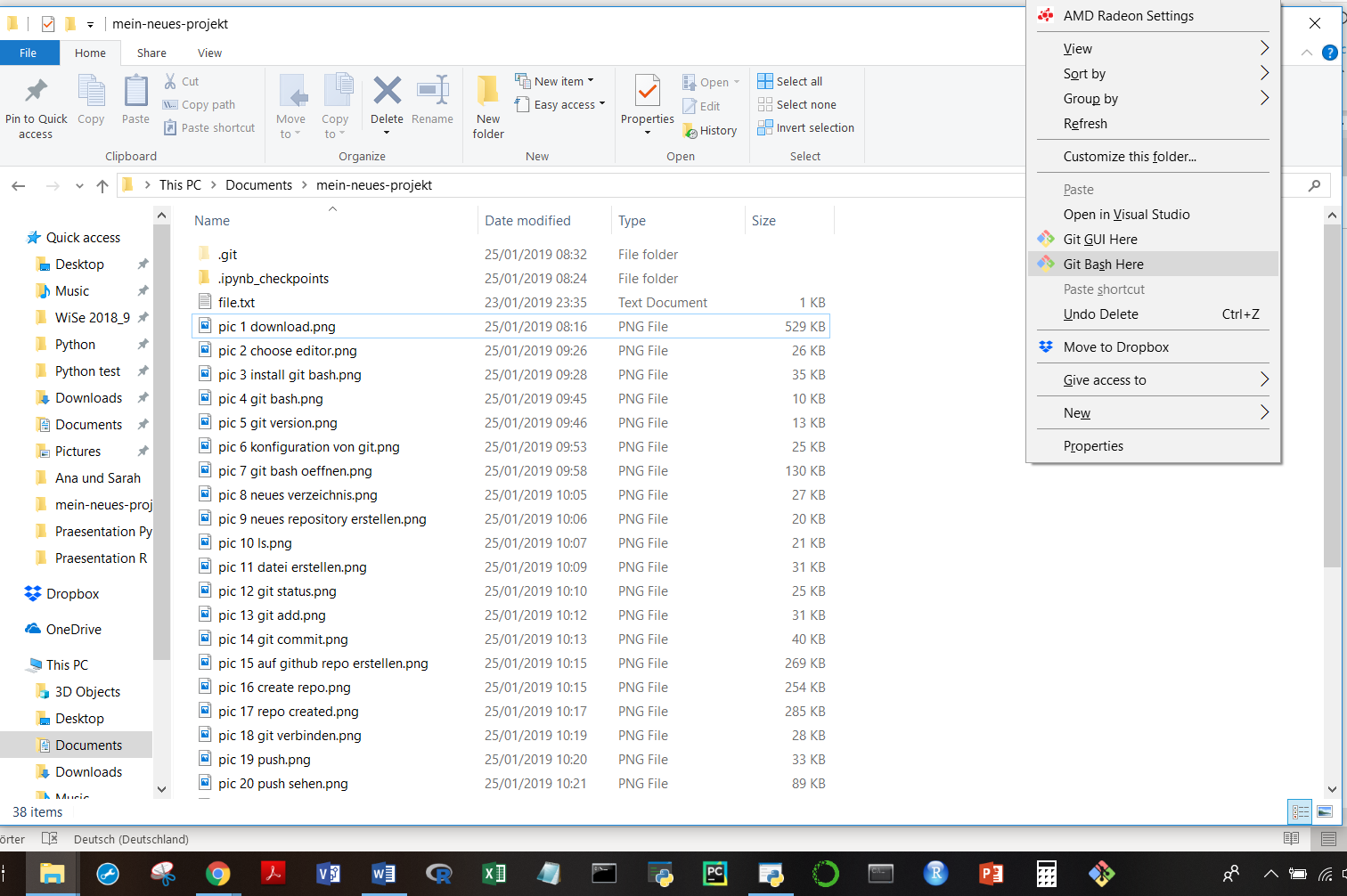
q

$ git push origin master

1. Remote Upload auf Github Repository und Pullen ins lokale Git
2. Auf Github direkt uploaden

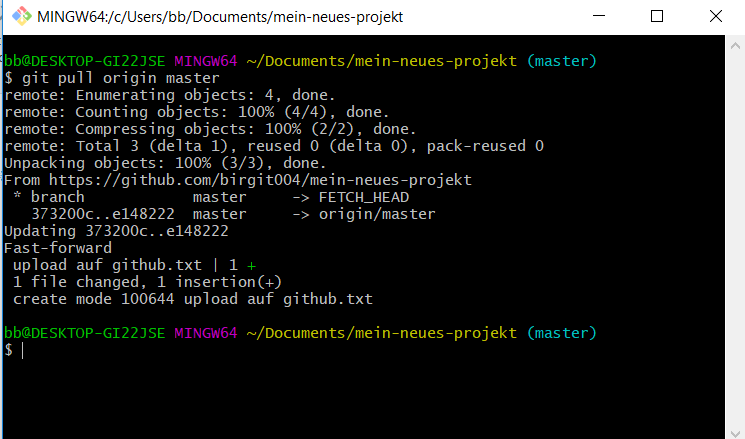


1. Ins lokale Git pullen: Git Bash im eigenen Verzeichnis öffnen



1. Pull Befehl eingeben:

$ git pull origin master



1. Verzeichnisinhalt ansehen

$ ls -la