

Für Hilfe:

→ `git --help`

Anmelden in Git:

`git config --global user.name NAME`

`git config --global user.email EMAIL`

Changing directory:

`cd` *Wo man hin will*

`cd ..` → für zurück

`ls` → um alles anzeigen zu lassen

`rm FILENAME` → um ein file zu löschen

`mkdir NAME` → für neuen Ordner

`rmdir NAME` → um einen Ordner zu löschen

Creating a repository (hat der Sebi schon gemacht, braucht man nicht mehr machen):

1. In einen Ordner gehen in dem man ein repository haben will
2. `git init`
3. jetzt sollte nach dem directory (master) stehen

Commits:

`git log` → um ein log von commits zu öffnen

`git log --oneline` → um ein log zu bekommen (ein commit pro zeile)

`git status` → um den status zu prüfen

`git add FILENAME` → um das file zum commit hinzuzufügen

`git add .` → um alle file zum commit hinzuzufügen

`git rm --cached FILENAME` → um das file vom commit zu entfernen

`git commit -m „Kommentar“` → um zu commiten

Undoing things/commits löschen (werden wir nicht wirklich brauchen):

https://www.youtube.com/watch?v=RIYrfkZjWmA&list=PL4cUxeGkcC9goXbgTDQ0n_4TBzOO0ocPR&index=7

Risiko zu groß, dass wir alles verkacken!!!

Branches:

- `git branch BRANCHNAME` → erstellt einen neuen Branch
- `git branch -a` → um alle Branches anzuzeigen und auf welchem man ist
- `git checkout BRANCHNAME` → um auf einen anderen Branch zu wechseln
- `git branch -d BRANCHNAME` → um einen Branch zu löschen
- `git branch -D BRANCHNAME` → um einen Branch zu löschen ohne dass der Code auf dem Masterbranch ist → Code ist dann weg
- `git checkout -b BRANCHNAME` → kombi aus checkout und Branch erstellen

Merging Branches:

- `git checkout master` → Goes to master branch
- `git merge BRANCHNAME` → Verbindet den genannten Branch mit dem Master branch

GitHub:

- `git clone URL` → um ein Repository zu clonen

Wenn man clont wird automatisch ein Alias namens origin gesetzt.

- `git remote add ALIAS URL` → um einen Alias zu erstellen (Beispiel: `git remote add origin https://github.com/SebiGoebel/G1-TravelingSalesMan_PAD2-Projekt`)
- `git remote -v` → um alle Aliase und URLs anzeigen zu lassen

(Vor dem Pushen muss man commiten)

- `git push URL BRANCHNAME` → lädt den Code auf GitHub

Nach dem erstellen eines Alias kann man den Alias statt der URL verwenden:

- `git push ALIAS BRANCHNAME` → lädt den Code auf GitHub (Beispiel: `git push origin master`)