My little helper

* Git init 🡪 um im Pfad eine Repository anzulegen
* Git status 🡪 Status des Repository abfragen
* Git add 🡪 um Änderung am File hinzuzufügen (speichern) datei muss aber im Pfad liegen, mit Leertaste zwischen einzelnen Files können auch mehrere Files gleichzeitig hinzugefügt werden
* Git add . 🡪 alle geänderten Files im Stammverzeichnis hinzufügen
* Git add –all/-A 🡪 alle Files aus allen Verzeichnissen werden hinzugefügt
* Git rm –cached namederDatei / git redet nameDerDatei 🡪 löscht Datei aus verzeichniss/Computer (vor 1. Commit nicht mehr wiederherstellbar)
* Git commit -m „message“ 🡪 file aus Staging in Repository zu verschieben
* Git commit -a -m „“ fügt Datei in Staging-Bereich und auch gleich ins Repository
* Git reset – soft HEAD ^ 🡪 Festschreiben (commit) rückgängig machen
* Geut commit --amend -m „Vergessene Datei hinzugefügt“ 🡪fügt Datei zu letzem commit hinzu
  + Git remote add origin adressevongithubrepository 🡪 stellt verbindung zwischen lokalem Repository u Githup Repository her
  + Git push -u origin master 🡪 pusht beim ersten mal alles von lokal auf github
  + Git push -force origin master 🡪 knn psh erzwingen
  + Git push 🡪 kann bei weitern pushes veurwendet werden statt -u origin master
* -Insert-: ESC 🡪 :pw + enter
* Git remot -v 🡪 zeigt die verwendeten remote Repositories an
* Git clone adressevongithub 🡪 in entprechend pfad wo ich es haben will, wird ein Repository von github erstellt
* Git pull 🡪 holt änderunge nvon github repo
* Git branch 🡪 zeigt alle branches an
* Git branch nameDesNeuenBranches 🡪 Bildung eines neuen Branches
* Git checkout nameDesBranchesWoIchHinWill 🡪 wechselt in einen bestimmten branch
* Git merge anderesProfil 🡪 chekcen sie in den branch ein (checkout master), in den die Neuerungen gemerged werden sollen und dann wird der Befehl aufgeführt.
* git push --set-upstream origin manuel 🡪 verbindet neuen lokalen branch mit github u legt dort neuen branch an
* git push origin --delete manuel 🡪 lösch branch auf github
* git checkout -b namedesNeuenBranches 🡪 erst neuen branch u checkt ihn ein
* *git rebase master* 🡪 nimmt Änderungen vom master branch in den neuen branch auf u legt die Änderungen des neuen branches oben drüber
* *wenn ich von einem bestehend branch auf github etwas holen will:*
  + *git checkout nameDesBranchesDerAufGitHubIst*

Wenn Sie eine Pull-Anforderung für eine Verzweigung ausführen, können Sie in einer anderen Verzweigung weiterarbeiten und eine weitere Pull-Anforderung für diese andere Verzweigung ausführen.

Ziehen Sie die Änderungen vom Upstream, bevor Sie einen neuen Zweig erstellen. Ihr Meister muss auf dem neuesten Stand sein.

$ git pull

Erstellen Sie den Zweig auf Ihrem lokalen Computer und wechseln Sie in diesen Zweig:

$ git checkout -b [Name\_Ihres\_Neubranch]

Schieben Sie den Zweig auf Github:

$ git push origin [name\_of\_your\_new\_branch]

Stellen Sie sicher, dass Sie sich in Ihrer Filiale befinden, wenn Sie etwas in Ihrer Filiale festlegen möchten. Fügen Sie den Parameter -u hinzu, um den Upstream festzulegen.

Sie können alle erstellten Zweige anzeigen, indem Sie Folgendes verwenden:

$ git branch -a

Welches wird zeigen:

\* approved\_messages

Meister

master\_clean

Fügen Sie eine neue Fernbedienung für Ihre Niederlassung hinzu:

$ git remote add [name\_of\_your\_remote] [name\_of\_your\_new\_branch]

Änderungen aus Ihrem Commit in Ihre Filiale übertragen:

$ git push [name\_of\_your\_new\_remote] [url]

Aktualisieren Sie Ihren Zweig, wenn der ursprüngliche Zweig aus dem offiziellen Repository aktualisiert wurde:

$ git fetch [name\_of\_your\_remote]

Dann müssen Sie sich bewerben, um Änderungen zusammenzuführen. Wenn Ihr Zweig von "Develop" abgeleitet ist, müssen Sie Folgendes tun:

$ git merge [name\_of\_your\_remote] / develop

Löschen Sie einen Zweig in Ihrem lokalen Dateisystem:

$ git branch -d [name\_of\_your\_new\_branch]

So erzwingen Sie das Löschen des lokalen Zweigs in Ihrem Dateisystem:

$ git branch -D [name\_of\_your\_new\_branch]

Löschen Sie den Zweig auf Github:

$ git push origin: [name\_of\_your\_new\_branch]

Der einzige Unterschied ist: zu sagen, löschen, Sie können es auch tun, indem Sie die Github-Schnittstelle verwenden, um den Zweig zu entfernen: <https://help.github.com/articles/deleting-unused-branches> .

Wenn Sie den Standardzweig ändern möchten, ist es mit Github so einfach. Wechseln Sie in Ihrem Zweig in Admin und wählen Sie in der Dropdown-Liste den gewünschten Zweig aus.

Wenn Sie einen neuen Zweig erstellen möchten:

$ git branch < name\_of\_your\_new\_branch >

Git flow für Projekt:

* Git checkout develop
* Git fetch
* Git pull
* Git checkout UI-Tests
* Git merge develop
  + Git add/commit -a -m
  + Git push

Vim pfad = ofnet mir den ordner im Pfad

Git checkout --theirs