

## My Git Workflow - Mini-Leitfaden

=====

Dieser Leitfaden hilft dir, wie du NACH dem ersten Commit deine Arbeit im Projekt korrekt mit Git verwaltest.

-----

### # 1. Projekt öffnen

```
cd C:/Users/Ben/Documents/my-dream-of-jeannie
```

ODER (wenn du es schon geklont hast)

```
cd my-dream-of-jeannie
```

-----

### # 2. Status prüfen (was hat sich verändert?)

```
git status
```

Zeigt an:

- Geänderte Dateien
- Neue Dateien
- Dateien, die noch nicht zum Commit vorgemerkt sind

-----

### # 3. Änderungen zum Commit vorbereiten

```
git add dateiname.py
```

ODER für alle Änderungen im Projekt:

```
git add .
```

-----

### # 4. Commit erstellen

```
git commit -m "Kurze Beschreibung deiner Änderung"
```

Beispiel:

```
git commit -m "Added greeting feature to Jeannie"
```

-----

### # 5. Änderungen auf GitHub hochladen

```
git push
```

ODER beim ersten Push:

```
git push -u origin main
```

Danach reicht immer nur:

git push

---

# 6. Zusammenfassung - typischer Git-Alltag

# Neuen Code geschrieben -> dann:

git status

git add .

git commit -m "Was habe ich geändert?"

git push

---

# 7. Tipp: Status IMMER vorher prüfen

git status

-> Sehr wichtig! Damit siehst du, ob du alles korrekt zum Commit hinzugefügt hast.

---

# 8. Kontrolle der globalen Git-Einstellungen (nur 1x nötig)

git config --global --list

-> Kontrolliert deinen User + E-Mail

-----

Viel Spaß beim Entwickeln mit Git & GitHub!

Sinnvolles Ziel: JEDES Commit sollte eine kleine abgeschlossene Änderung haben.