# GIT Tutorial

| Guide zum erstellen der perfekten GIT-Message. |
| --- |
| **Hilfstext Befehlsform für GIT Messages:**<br><br>If applied, this commit will [commit message].<br><br>[commit message] bezieht sich auf die Überschrift der git message. Diese soll so formuliert sein das sie zum Hilfstext passt (Befehlsform etc).<br><br>**Die 7 Goldenen Regeln für eine perfekte Commit Nachricht:**<br><br>1) Header in Befehlsform<br>2) Großschreibung am Headeranfang<br>3) Header ohne Satzpunkt<br>4) Header maximal 50 Zeichen<br>5) Zwischen Header und Body gehört eine Leerzeile<br>6) Body hat nach 72 Zeichen Zeilensprung<br>7) Body erklärt Was? und Warum? anstatt Wie? |

**Init:** Erstellt unsichtbaren GIT Ordner, der GIT Einstellungen speichert:

git init

**Clone:** Kopiert das Repository an den Ort wo die GIT-Bash aktiv ist:

git clone https://github.com/BenPapple/ai-GOALdigger

**status:** Zeigt Informationen über git an (zB Branch: Master, ..):

git status

**Master/Main:** Master in Main in git umbenennen(die Community möchte den Begriff Master loswerden, Github hat schon Main als Default, das Programm git aber noch nicht, deswegen muss man das noch hiermit ändern damit Beide zusammenarbeiten):

git branch -m master main

**Commits:** erstellen:

1. Vorher mit **pull** automatisch neue Dateien/Änderungen runterladen:

git pull origin <branch-name>

2. Mit **add** (**rm** um Dateien zu entfernen) Dateien zum Commit hinzufügen, diese werden ab dann von git auf Änderungen überwacht:

git add DATEI1 DATEI2 DATEI3

3. **Commit message** als Direkteingabe (nur Überschrift):

git commit -m "Add Readme"

3.1 Oder **Commit message** aus Textdatei (Überschrift plus Textblock):

git commit -F message.txt

4. Dann mit **push** Daten auf den Server schieben:

git push origin <branch-name>

**Branch** erstellen (-b) und dorthin wechseln (checkout):

git checkout -b <branch-name>

In **Branch** wechseln:

git checkout <branch-name>

In .**gitignore Datei** einfach Dateinamen eintragen, die nicht von git getrackt werden sollen, zum Beispiel:

.project
goaldiggersrc/.project
message.txt

| **Begriffe:** |
| --- |
| • **Merge Branches**<br>  ◦ **Rebase**: Branches werden zu einer linearen Commithistory, alle Commits zählen<br>  ◦ **Squash**: wie Rebase, aber Commits eines Branch werden zu einem Commit zusammengefasst und eingefügt |