

Cheat sheet



Wichtige Befehle:

git init	Lokales git Repository erstellen (.git wird angelegt)
git status	Aktueller Status von Working-Directory, Staging-Area und Repository
git add <Datei>	Genannte Datei in die Stage einfügen
git add .	Fügt alle Dateien der Stage hinzu
git stage .	Analog zu add
git commit -m“...”	Stage ins Repository verschieben (aussagekräftige Nachricht!)
git log	Zeigt den zeitlichen Verlauf der Aktionen
git checkout -b <name>	Erstellt einen neuen Branch
git checkout <name>	Holt einen Branch ins Working-Directory
git merge <branch>	Mergt den genannten Branch in das aktuelle Working-Directory

Lokales Repository hochladen

1.	Leeres „remote Repository“ erstellen (z.B. bei GitHub)	
2.	Git remote add origin <pfad>	Repository lokal hinzufügen
3.	Git push	Lokales git Repository hochladen
!!!	Um zum Beispiel auf ein privates Repository zugreifen zu können, muss der GitHub Benutzer im lokalen Repository hinterlegt sein! git config --global user.name "name" git config --global user.email max@mustermann.de README.md und .gitignore nicht vergessen	

Bestehendes online Repository herunterladen

git init	Lokales git Repository erstellen
git clone <pfad>	Online Repository in lokales Repository herunterladen

Ebenfalls hilfreich: <https://rogerdudler.github.io/git-guide/index.de.html>

Cheat sheet



Um das gleiche auszuführen wie in der Präsentation:

Java Projekt erstellen	
Code implementieren (zB Chatbot)	
git init	Lokales git Repository erstellen
git add <Datei>	Testweise eine Datei der Stage hinzufügen
git add .	Alle Dateien der Stage hinzufügen
git status	Überprüfen ob alles geklappt hat
git commit -m“...”	Stage ins Repository verschieben (aussagekräftige Nachricht!)
git log	Zeigt den zeitlichen Verlauf der Aktionen
Weitere use Cases implementieren	
git checkout -b <name>	Branch erstellen für die entwickelten, neuen use cases
git add .	Alle Dateien der Stage hinzufügen
git commit -m“...”	Stage ins Repository verschieben (aussagekräftige Nachricht!)
git checkout master	Master ins Working-Directory holen
git merge <branch>	Feature Branch mit neuen use cases in den master mergen
git log	Zeigt den zeitlichen Verlauf der Aktionen → Kontrolle
Online GitHub Repository erstellen	
git remote add origin <euer repo>	Online GitHub Repo dem lokalen git hinzufügen
git push	Das lokale Repo hochladen
Weitere use Cases implementieren	
git checkout -b <name>	Branch erstellen für die entwickelten, neuen use cases
git add .	Alle Dateien der Stage hinzufügen

Cheat sheet



git commit -m“...”	Stage ins Repository verschieben (aussagekräftige Nachricht!)
git push	Branch ins online Repository hochladen
In GitHub einen Pull request für den Branch erstellen	
In GitHub den Branch in den Master mergen	
git checkout master	Master ins Working-Directory holen
git pull	Online zusammengeführten master ins lokale Repository laden