

# Wichtige Git-Befehle

---

## git init

---

Initialisiert ein neues Git-Repository in einem Verzeichnis, indem ein `.git`-Verzeichnis erstellt wird, das alle Git-bezogenen Daten für das Repository enthält.

## git clone

---

Klont ein bestehendes Git-Repository von einer Remote-URL in ein lokales Verzeichnis und erstellt dabei eine Kopie aller Dateien und Historie des Repositories.

## git add

---

Fügt eine oder mehrere Dateien zum Staging-Bereich hinzu, damit sie für den nächsten Commit vorbereitet sind. In einfachen Worten: `git add .` fügt alle Änderungen hinzu, während `git add <file>` nur die angegebene Datei hinzufügt.

## git commit -m ""

---

Erstellt einen Commit mit den Änderungen im Staging-Bereich und fügt eine kurze Commit-Nachricht hinzu, um die gemachten Änderungen zu beschreiben.

## git status

---

Zeigt den aktuellen Status des Arbeitsverzeichnisses und des Staging-Bereichs an. Zeigt unter anderem unversionierte Dateien, veränderte Dateien und Dateien im Staging-Bereich an.

## git log

---

Zeigt eine chronologische Liste aller bisherigen Commits im Repository an, einschließlich Autor, Datum und Commit-Nachricht.

## git branch

---

Zeigt eine Liste aller vorhandenen Branches im Repository an. Der aktuelle Branch wird durch ein Sternchen (\*) markiert.

## git checkout

---

Wechselt zu einem anderen Branch im Repository. Du kannst auch mit `git checkout -b <new-branch-name>` einen neuen Branch erstellen und zu diesem wechseln.

## git merge

---

Führt die Änderungen aus einem anderen Branch in den aktuellen Branch zusammen (merge). Dies wird typischerweise verwendet, um Feature-Banches in den Hauptentwicklungs-Branch zu integrieren.

## git pull

---

Holt Änderungen aus einem Remote-Repository und führt automatisch ein `git merge` durch, um die Änderungen in deinem lokalen Repository zu fusionieren.

## git push

---

Aktualisiert das Remote-Repository mit den lokalen Commits. Damit werden die Commits von deinem lokalen Repository auf das Remote-Repository hochgeladen.

## git remote -v

---

Zeigt alle Remote-Repositorys an, die mit dem lokalen Repository verknüpft sind, sowie die dazugehörigen URLs.

## git reset

---

Entfernt eine Datei oder Änderungen an einer Datei aus dem Staging-Bereich, ohne die Datei selbst zu ändern.

## git rm

---

Entfernt eine Datei aus dem Arbeitsverzeichnis und dem Staging-Bereich und markiert sie für den nächsten Commit als gelöscht.

## git fetch

---

Holt alle Daten von einem Remote-Repository, ohne die lokalen Dateien zu ändern. Dies ist nützlich, um zu überprüfen, ob es neue Änderungen im Remote-Repository gibt, ohne die eigenen lokalen Dateien zu aktualisieren.