

BEGRIFFSWELT:

Repository

Der Ort/Ordner, an welchem die Historie der Dateien gespeichert wird.

Staging Area

Dateien, die für den nächsten **Commit** vorgesehen werden. **git add** fügt Dateien zur Staging Area hinzu.

Commit

Ein Set von Änderungen (Dateien + Metadaten wie Zeit, Autor und eine Beschreibung - die Commitnachricht) zu einem bestimmten Zeitpunkt. Ein Commit enthält jeweils den Verweis auf den vorherigen Commit = Eltern/Parentcommit (Oder zwei Elterncommits, wenn es ein **Merge** war).

Commithash

Eine Prüfsumme über die Datei-Änderungen und die Metadaten um einen Commit identifizieren zu können.

Branch

Ein Branch ist ein simpler Zeiger auf einen Commit. Durch einen Branch können Änderungen abseits der Hauptlinie (meistens master-Branch) entwickelt und später wieder mit einem **Merge** zusammengeführt werden.

HEAD

Ein Zeiger auf den aktuell aktiven Branch.

Remote

Branches gibt es lokal, aber auch bei sogenannten Remotes (z.B. bereitgestellt durch github, gitlab, usw.). Über gemeinsame Remote-Banches kann der Stand mit anderen Entwicklern geteilt und organisiert werden.

Merge

Führt zwei Branches zusammen. Dabei kann es zu Mergekonflikten kommen.

Rebase

Löst eine bestimmte Anzahl von Commits von ihrer aktueller Commitbasis los und versetzt sie auf eine andere Basis. Die Elterncommits und damit die Commithashes ändern sich.

.gitignore

Eine Datei, um bestimmte Ordner oder Dateitypen aus der Repositoryverwaltung durch Git auszuschliessen.

BEFEHLE:

git init

Initialisiert das Repository vor erstmaliger Verwendung.

git add DATEINAME

Fügt die Datei(en) passend zu **DATEINAME** zur **Staging Area** hinzu

git commit -m COMMIT-NACHRICHT

Erstellt einen Commit mit den aktuellen Änderungen der Staging Area und der benutzten **Commit-Nachricht**.

git log

Zeigt die Historie der letzten Commits an.

git branch

Zeigt alle lokalen Branches an.

git branch BRANCH-NAME

Erstellt einen neuen Branch namens **BRANCH-NAME**, der erstmal auf den gleichen Commit verweist, wie der aktuell aktive Branch (**HEAD**)

git checkout BRANCH-NAME

Wechselt zum Branch **BRANCH-NAME**.

git branch -d BRANCH-NAME

Lösche den Branch **BRANCH-NAME** (Nur wenn es keine ungemergten Änderungen gibt).

git merge BRANCH-NAME

Bringt die Änderungen von Branch **BRANCH-NAME** in den aktuellen Branch.

git rebase BRANCH-NAME

Verändert die Basis der Commits des Branches mit dem Branch **BRANCH-NAME** als neue Basis.

git push -u origin **BRANCH-NAME**

Speichert für den aktuellen lokalen Branch einen Verweis auf einen Remotebranch **BRANCH-NAME** (Erzeugt diesen, falls nicht vorhanden) und bringt die lokalen Änderungen dorthin.

git push

Bringt die lokalen Änderungen des aktuellen Branches zum entsprechenden Remotebranch).

git fetch

Holt die Änderungen der Remotebranches, wendet sie aber bei keinem lokalen Branch direkt an.

git pull

Holt erst wie git fetch die Änderungen der Remotebranches und führt dann ein **Merge** vom passenden Remotebranch in den aktuellen lokalen Branch aus.