

# Git Notizen

---

[Arbeitsverzeichnis] <- - - -> [Index] <- - - -> [HEAD]

- <Arbeitsverzeichnis> ... lokaler Arbeitsstand
- Index ... gespeicherter Zwischenstand (staging area), welcher später in das Repository übertragen wird
- HEAD ... zeigt auf den aktuellen Branch im Repository. (Sonderfall detached HEAD ... zeigt auf einen älteren commit)
- -- ... behandelt **Parameter** nicht als **Option** (bspw. *-file* als Datei)

## neues Repository erstellen

```
$ mkdir repository && cd $_  
$ git init
```

## Änderungen und Änderungslog anschauen

```
$ git status // Arbeitsverzeichnis gegenüber Index  
  
$ git log  
$ git log --oneline  
  
$ git diff <file> // Arbeitsverzeichnis gegenüber Index  
$ git diff --cached <file> // Index gegenüber HEAD  
$ git diff HEAD <file> // Arbeitsverzeichnis gegenüber HEAD  
  
$ git show <id>:<file> // <id> ... commit id oder relativ HEAD~1  
$ cat <file> // Version im Arbeitsverzeichnis  
$ git show :<file> // Version im Index  
$ git show HEAD:<file> // Version im HEAD
```

## Arbeitsverzeichnis bearbeiten

```
$ git checkout <file> // reset <file> zu Index  
$ git checkout HEAD <file> // reset <file> und Index zu HEAD  
  
$ git clean -dfn // entfernt Dateien und Verzeichnisse, welche nicht unter Versionskontrolle bzw. im Index stehen. (Option `n` zeigt nur an, was gelöscht werden würde, `x` lösche auch von Git ignorierte Dateien und Verzeichnisse)
```

## Index bearbeiten

```
$ git add <file>
$ git add .

$ git rm --cached <file> // entfernt <file> vom Index
$ git rm <file> // entfernt <file> vom Index und Arbeitsverzeichnis

$ git reset --mixed HEAD <file> // entfernt <file> vom Index (Index = Head)
$ git reset <file> // Kurzform (Standardwerte: --mixed HEAD)
$ git reset // entfernt alle Änderungen vom Index (Index = Head)
$ git reset --hard HEAD // entfernt alle Änderungen vom Index und
Arbeitsverzeichnis (Arbeitsverzeichnis = Index = HEAD)
```

## Änderungen im Index in das Repository speichern

```
$ git commit -m "Commit-Nachricht"
```

## Änderungen im Repository zurücknehmen

private branch, undoing uncommitted changes:

```
$ git reset <id> // <id> ... commit id oder relativ HEAD~1
```

public branch, undoing committed changes:

```
$ git revert <id> // <id> ... commit id oder relativ HEAD~1
```

## Mit Remote-Repository arbeiten

```
$ git clone user@host:/path/to/repository
$ git clone /path/to/repository
$ git remote add origin git@iglu.informatik.htw-dresden.de:
<username>/<repository>.git

$ git pull // Änderungen herunterladen
$ git push -u origin master // Änderungen hochladen
```