

Wichtige GIT Befehle

Git-Repository anlegen:

```
cd Pfad
git init
git clone https://website/git-repository (falls Git
Repository existiert)
```

Repository Status und neue Dateien hinzufügen:

```
git status
git add Test.txt
```

Bestätigungen der Änderungen (Snapshot)

```
git commit
git commit -m „nachricht“
git commit -a
git commit -am „nachricht“
```

Commits ändern

```
git commit --amend
Andere Befehle(nicht so wichtig):
git reset --soft HEAD~1
git reset --hard HEAD~1
```

Erzeugten Commits anzeigen

```
git log
Andere Befehle(nicht so wichtig):
git log -P
git log -stat
git log -pretty
git log -
```

Branches erstellen, verwalten und löschen

```
git branch <new_branch_name>
git checkout <new_branch_name>

(git checkout master)
git merge test_branch

git branch -d <branch_name_to_be_deleted>
```

**Das lokale Repository zu einem remote Repository hinzufügen/
Verbindungen mit anderen Repositories erstellen, löschen**

```
git remote add origin/<name> <url_name>  
Git remote rm <name>
```

Commits/lokale Änderungen in einem zentralen Repository hochzuladen

```
git push origin/<remote> <branch>
```

Commits aus einem remote Repository in einem lokales Repository herunterladen

```
Git fetch <remote> <branch>  
Git fetch --all
```

**Inhalte aus einem remote Repository herunterladen und das lokale Repository zu
aktualisieren um die Inhalte beider übereinzustimmen**

```
git pull <remote>
```

Merging

```
git checkout <feature-Branch>  
Git merge <main-Branch>  
Oder:  
Git merge <feature-Branch> <main-Branch>
```

**Rebasing (Integration von Änderungen von einem Branch in einen anderen. Hier
aber in Unterschied zu Merging werden alle Commits neu erstellt werden.)**

```
git checkout <feature-Branch>  
git rebase <main-Branch>  
Oder:  
git rebase <feature-Branch> <main-Branch>
```

**Beliebige Git-Commits anhand einer Referenz auswählen und an den aktuellen HEAD
anhängen**

```
git checkout <main-Branch>  
git cherry-pick <Commit-Referenz>
```