

# GIT Tutorial

- Hilfreiche Links vorab:.
- Grundlegende Befehle:.
  - Branching.
  - Tags.
  - Reset lokal Repository.
- Software (Windows oder MultiPlattform).
- Häufiger Workflow.

## Hilfreiche Links vorab:

[Tutorial - kein SchnickSchnack](#)

[Git für Einsteiger](#)

[Github Playground](#)

[Think like \(a\) GIT](#)

[Dieses Repo](#)

---

## Grundlegende Befehle:

`git init`

`git add . || git add`

`git commit -m "grund" || git commit -a -m "grund"`

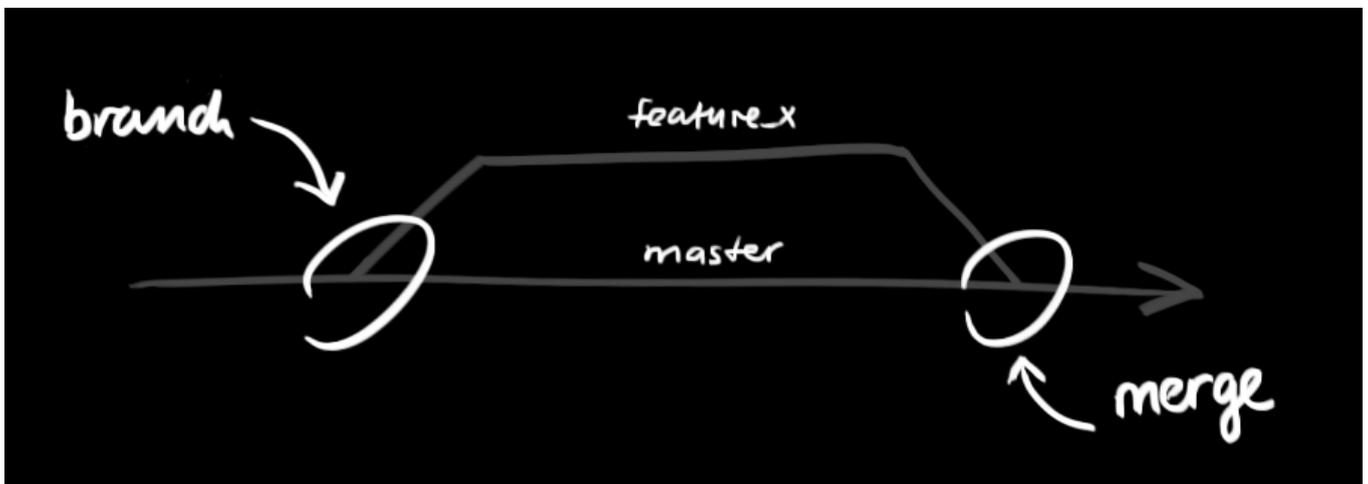
`git remote add origin http://pfad.zum/repo.git`

`git push -u origin master` (einmalig, dann langt git push für push ins master)

`git clone url`

`git clone --recursive`

## Branching



git branch "branchname" -> Branch erstellen

git checkout "branchname" -> Zum Branch wechseln

git push --set-upstream origin "branchname" -> Branch hochladen

git checkout master -> Zurück zum master Branch

git merge "branchname" -> branch einchecken

git branch -d "branchname" -> branch löschen

git push -> hochladen

## Tags

git tag -> Tags anzeigen

git tag -a v1.2.3 -m "Version 1.2.3" -> Kommentierter Tag erstellen

git push origin --tags -> Tags in Repo pushen

## Reset lokal Repository

git fetch origin

git reset --hard origin/master

## Software (Windows oder MultiPlattform)

[GitBash & GitGUI](#)

[SourceTree](#)

Github Desktop

Git Kraken

Fork

CodeReview

Git Blade

---

## Häufiger Workflow

Während der Entwicklungsarbeit wird der Workflow meistens in etwa so aussehen:

- Aktuellen master-Branch vom Server downloaden (vorher mit `git status` sicherstellen, dass man sich auf dem master-Branch befindet. Wenn nicht: `git checkout master`):

```
#> git pull
```

- Einen neuen Branch für das kommende Feature anlegen:

```
#> git checkout -b meinfeature master
```

- [Änderungen am Code durchführen]
- Geänderte und neue Dateien stagen:

```
#> git add .
```

- Änderungen committen:

```
#> git commit -m "Changes xyz"
```

- Jetzt den Branch auf den Server laden, wenn gewünscht:

```
#> git push -u origin meinfeature
```

- ... oder direkt in dem master wechseln:

```
#> git checkout master
```

- (nochmals aktuellen Code ziehen - zur Sicherheit)

```
#> git pull
```

- Eigenen Code mit master zusammenführen:

```
#> git merge meinfeature
```

- Nicht mehr benötigten Branch löschen:

```
#> git branch -d meinfeature
```

- Aktualisierten master auf den Server pushen:

```
#> git push
```

- Fertig