

## Downloads

### Git Core

Windows, Mac OS X:  
<https://git-scm.com>

Linux: Über Paketmanager:

```
apt-get install git-core
yum install git-core
```

### GUIs für Windows & Mac OS X

- SourceTree (kostenlos):  
<https://www.sourcetreeapp.com/>
- GitKraken (kostenlos, beta):  
<http://www.gitkraken.com/>
- SmartGit (kostenlos bei nichtkommerzieller Nutzung):  
<http://www.syntevo.com/smartgit/>
- GitHub Desktop (kostenlos):  
<https://desktop.github.com/>

### GUIs für Linux

- GitKraken (kostenlos, beta):  
<http://www.gitkraken.com/>
- SmartGit (kostenlos bei nichtkommerzieller Nutzung):  
<http://www.syntevo.com/smartgit/>
- gitg (kostenlos):  
<https://wiki.gnome.org/Apps/Gitg/>

## Ersteinrichtung

```
git config --global user.name "Mein Name"
git config --global user.email email@example.com
```

## Befehle

### Repository erstellen

Existierendes Repository:

```
git clone ssh://user@domain.com/repo.git
```

Neues Repository:

```
git init
```

### Änderungen

Veränderte Dateien:

```
git status
```

Änderungen an versionierten Dateien:

```
git diff
```

Alle Änderungen zum nächsten Commit hinzufügen:

```
git add .
```

Vorher hinzugefügte Änderungen committen:

```
git commit
```

Alle Änderungen direkt committen:

```
git commit -a
```

Letzten Commit verändern (**nur ungepushte Commits!**):

```
git commit --amend
```

### Commit-Historie

Alle Commits:

```
git log
```

Commits für eine bestimmte Datei:

```
git log <Datei>
```

### Branches:

Alle Branches anzeigen:

```
git branch -av
```

Anderen Branch auschecken:

```
git checkout <Branch>
```

Neuen Branch auf HEAD erstellen:

```
git branch <Neuer Branch>
```

Branch löschen:

```
git branch -d <Branch>
```

Aktuellen Commit taggen:

```
git tag <Tag-Name>
```

## Arbeiten mit Remotes

Alle eingestellten Remotes anzeigen:

```
git remote -v
```

Infos zu einem Remote:

```
git remote show <Remote>
```

Remote hinzufügen:

```
git remote add <Name> <URL>
```

Änderungen von Remote herunterladen:

```
git fetch <Remote>
```

Änderungen herunterladen und in HEAD mergen:

```
git pull <Remote> <Branch>
```

Lokale Änderungen auf Remote veröffentlichen:

```
git push <Remote> <Branch>
```

Branch auf Remote löschen:

```
git branch -dr <Remote>/<Branch>
```

Tags veröffentlichen

```
git push -tags
```

## Merge, Rebase

Anderen Branch in HEAD mergen:

```
git merge <Branch>
```

HEAD auf einen anderen Branch rebasen:  
**(Keine gepushten Commits rebasen!)**

```
git rebase <Branch>
```

Rebase abbrechen:

```
git rebase -abort
```

Rebase nach gelösten Konflikten fortsetzen:

```
git rebase -continue
```

Konflikt als gelöst markieren:

```
git add <Datei>
```

oder:

```
git rm <Datei>
```

## Zurücksetzen

Alle Änderungen im Arbeitsverzeichnis zurücksetzen:

```
git reset -hard HEAD
```

Änderungen an bestimmten Dateien zurücksetzen:

```
git checkout HEAD <Datei>
```

Commit rückgängig machen (erstellt neuen Commit):

```
git revert <Commit-Hash>
```

HEAD auf anderen Commit setzen...  
... und alle Änderungen verwerfen:

```
git reset -hard <Commit-Hash>
```

... und alle Änderungen erhalten:

```
git reset <Commit-Hash>
```

... und zusätzlich uncommittete Änderungen erhalten:

```
git reset -keep <Commit-Hash>
```