

# GIT

Version Control System

<https://git-scm.com/book/en/v2>

... Freebook

<https://marklodato.github.io/visual-git-guide/index-en.html>

... super Tutorial

<http://onlywei.github.io/explain-git-with-d3/#fetch>

... interaktives Tutorial +++

<https://git-school.github.io/visualizing-git/>

```
git config --global user.name "kner"  
git config --global user.email "kner@kner.at"
```

Verzeichnis anlegen, dann

```
git init  
git status
```

```
git add filename.ext  
git add *.c  
git add .  
git add *  
git commit -m "Short Message"  
git commit --amend ... auf den letzten commit zurücksetzen
```

```
git log
```

```
git diff  
git diff --staged
```

```
git rm x.txt ... removes file from working and stage
```

```
git branch test_branch ... neuer Branch  
git checkout test_branch
```

kurz:

```
git checkout -b test_branch2 ... neuer Branch
```

```
git checkout s1 ... copy from stage to work (discard changes)  
git checkout 99aaf s1 ... copy from repo to work  
git reset HEAD s1 ... from repo to staging area; Attention: not in work yet  
HEAD is the last commit  
git reset 99aaf s1 ... from the 99aaf commit
```

```
git push origin master ... den aktuellen Zustand auf den Server überspielen
```

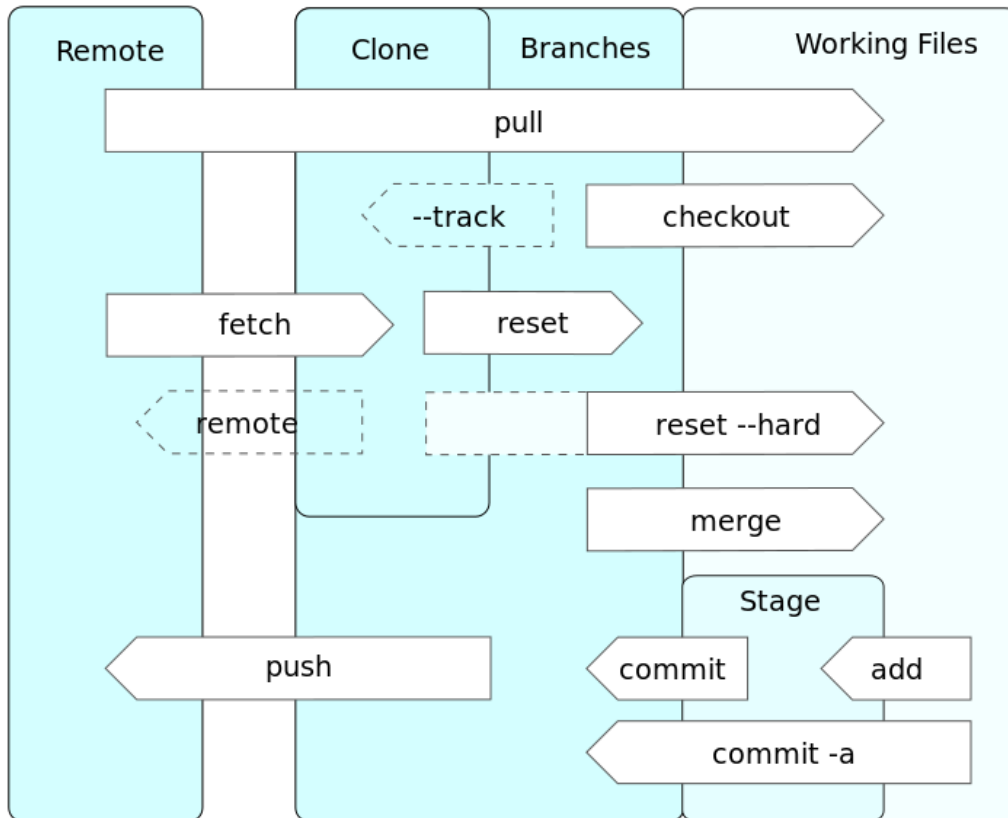
## Clonen über https

```
git clone https://github.com/ex/ex1
git clone https://github.com/ex/ex1 ex_neuename
```

## Clone/Push über ssh (Nötig wenn github Projekt)

```
git clone git@github.com:kner/MainGame.git
```

wenn man über https klont, funktioniert die Authentifizierung bei push nicht.



by Daniel Kinzler CC BY 3.0

## Gist

Code Snippets bereitstellen

# Giggle

einfache GUI für git

# Beispiel

Programmierer Anton und Otto arbeiten am selben Projekt. Anton initiiert das Repository, Otto klonet es, dann arbeiten beide zusammen weiter am Projekt. Es gibt lokale Variable L1,L2,... und es gibt globale Variable G1,G2,â€¦

#Anton		#Otto	
Alles neu			
rm -rf /tmp/Anton rm -rf /tmp/Otto		rm -rf /tmp/Otto	
Neues Projekt anlegen			
mkdir /tmp/Anton cd /tmp/Anton echo L1=1>>locales.txt echo L2=2>>locales.txt echo G1=1>>globales.txt echo G2=2>>globales.txt			
Git Init			
git init git add --all git commit -m "CA1"	CA1		
echo L3=3>>locales.txt git add locales.txt git commit -m "CA2" git log	CA2 CA1		
Otto klonet den Anton			
		cd /tmp git clone /tmp/Anton /tmp/Otto cd /tmp/Otto git log	CA2 CA1
echo "L3=3A" >>locales.txt git status	locales.txt geändert, nur im Bereich Working		
git add --all git status	locales.txt staged		
git pull kommt zu früh, noch kein commit von Anton			
echo L3=3B>>locales.txt		git pull	CA2 CA1
git commit -m "C3A"	C3A (3B nicht erfasst) CA2 CA1	git pull	add locales.txt vergessen
git status	zeigt, dass locales geändert wurde		
git add --all git commit --amend -m "C3B"	C3B CA2 CA1	git fetch git show git merge	anstelle von git pull
Änderungen auf beiden Seiten gleichzeitig ==> Konflikt			
echo G1=1A>>globales.txt echo commit -a -m "CG1A"	CG1A C3B CA2 CA1	echo G1=1B>>globales.txt echo commit -a -m "CG1O"	CG1O C3B CA2 CA1

		git pull	Konflikt muss aufgelöst werden
git pull /tmp/Otto			

kner 25.01.22 17:01:26 - Seite neu 25.01.2022