

Installation (Version checken)

```
[manuelpuchner CPR_SW >> git --version  
git version 2.30.1 (Apple Git-130)  
manuelpuchner CPR_SW >>
```

Einstellungen

```
[manuelpuchner CPR_SW >> git --version  
git version 2.30.1 (Apple Git-130)  
[manuelpuchner CPR_SW >> git config --list  
credential.helper=osxkeychain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
user.name=ManuelPuchner  
user.email=75803474+ManuelPuchner@users.noreply.github.com  
core.autocrlf=input  
manuelpuchner CPR_SW >> |
```

GIT - Hilfe

```

[manuelpuchner CPR_SW >> git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
  clone                Clone a repository into a new directory
  init                 Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
  add                  Add file contents to the index
  mv                   Move or rename a file, a directory, or a symlink
  restore              Restore working tree files
  rm                   Remove files from the working tree and from the index
  sparse-checkout      Initialize and modify the sparse-checkout


examine the history and state (see also: git help revisions)
  bisect               Use binary search to find the commit that introduced a bug
  diff                 Show changes between commits, commit and working tree, etc
  grep                 Print lines matching a pattern
  log                  Show commit logs
  show                 Show various types of objects
  status               Show the working tree status


grow, mark and tweak your common history
  branch               List, create, or delete branches
  commit               Record changes to the repository
  merge                Join two or more development histories together
  rebase               Reapply commits on top of another base tip
  reset                Reset current HEAD to the specified state
  switch               Switch branches
  tag                  Create, list, delete or verify a tag object signed with GPG


collaborate (see also: git help workflows)
  fetch                Download objects and refs from another repository
  pull                 Fetch from and integrate with another repository or a local branch
  push                 Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
[manuelpuchner CPR_SW >> |

```

Kommandozeileninterpreter

```

[manuelpuchner CPR_SW >> cd /
[manuelpuchner / >> ls
Applications Users      cores      home       sbin        var
Library      Volumes      dev        opt         tmp
System       bin          etc        private    usr
[manuelpuchner / >> cd Users/manuelpuchner/Private\ Projects/test
[manuelpuchner test >> cd /
manuelpuchner / >> cd Users/manuelpuchner/Private\ Projects/test|

```

Lokales Repo erzeugen

```
[manuelpuchner CPR_SW >> mkdir firstRepo
manuelpuchner CPR_SW >> cd firstRepo
manuelpuchner firstRepo >> git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/manuelpuchner/Schule/2Klasse/CPR_SW/firstRepo/.git/
manuelpuchner firstRepo >> ls -a
.  ..  .git
manuelpuchner firstRepo >>
```

Status prüfen

```
[manuelpuchner firstRepo >> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
manuelpuchner firstRepo >> |
```

Inhalt adden

```
[manuelpuchner firstRepo >> touch firstFile.txt
manuelpuchner firstRepo >> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    firstFile.txt

nothing added to commit but untracked files present (use "git add" to track)
manuelpuchner firstRepo >> git add firstFile.txt
manuelpuchner firstRepo >> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   firstFile.txt

manuelpuchner firstRepo >>
```

Inhalt Commiten

```
[manuelpuchner firstRepo >> git commit -m "added first file"
[master (root-commit) 7102ac0] added first file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 firstFile.txt
[manuelpuchner firstRepo >> git status
On branch master
nothing to commit, working tree clean
manuelpuchner firstRepo >>
```

Inhalt bearbeiten:

```
[manuelpuchner firstRepo >> nano firstFile.txt
[manuelpuchner firstRepo >> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   firstFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[manuelpuchner firstRepo >> git add firstFile.txt
[manuelpuchner firstRepo >> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   firstFile.txt

[manuelpuchner firstRepo >> git commit -m "added fist line in the file"
[master ebba549] added fist line in the file
 1 file changed, 1 insertion(+)
[manuelpuchner firstRepo >> git status
On branch master
nothing to commit, working tree clean
manuelpuchner firstRepo >>
```

GIT - Historie

```
[manuelpuchner firstRepo >> git log
commit ebba54920011e4d89b973c5711231c7ebaf1e006 (HEAD -> master)
Author: ManuelPuchner <75803474+ManuelPuchner@users.noreply.github.com>
Date:   Tue Nov 16 09:10:40 2021 +0100

    added first line in the file

commit 7102ac0f16629070de831c28244693109242abdd
Author: ManuelPuchner <75803474+ManuelPuchner@users.noreply.github.com>
Date:   Tue Nov 16 09:09:44 2021 +0100

    added first file
manuelpuchner firstRepo >> |
```

Dateien löschen (Vollständig)

```
[manuelpuchner firstRepo >> touch delFile.txt delFile2.txt
[manuelpuchner firstRepo >> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    delFile.txt
    delFile2.txt

nothing added to commit but untracked files present (use "git add" to track)
[manuelpuchner firstRepo >> git add delFile.txt delFile2.txt
[manuelpuchner firstRepo >> git rm delFile.txt
error: the following file has changes staged in the index:
    delFile.txt
(use --cached to keep the file, or -f to force removal)
[manuelpuchner firstRepo >> git rm -f delFile.txt
rm 'delFile.txt'
[manuelpuchner firstRepo >> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   delFile2.txt

[manuelpuchner firstRepo >> git commit -m "added another file"
[master 5543c5a] added another file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 delFile2.txt
[manuelpuchner firstRepo >> git rm -f delFile2.txt
rm 'delFile2.txt'
[manuelpuchner firstRepo >> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    delFile2.txt

[manuelpuchner firstRepo >> git commit -m "removed the message"
[master a21787d] removed the message
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 delFile2.txt
manuelpuchner firstRepo >> |
```

Dateien löschen (nur aus der Versionsverwaltung)

```
[manuelpuchner firstRepo >> touch doNotTrack.txt
[manuelpuchner firstRepo >> touch doNotTrack2.txt
[manuelpuchner firstRepo >> ls
doNotTrack.txt doNotTrack2.txt firstFile.txt
[manuelpuchner firstRepo >> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    doNotTrack.txt
    doNotTrack2.txt

nothing added to commit but untracked files present (use "git add" to track)
[manuelpuchner firstRepo >> git add doNotTrack.txt doNotTrack2.txt
[manuelpuchner firstRepo >> git rm --cached doNotTrack.txt
rm 'doNotTrack.txt'
[manuelpuchner firstRepo >> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   doNotTrack2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    doNotTrack.txt

[manuelpuchner firstRepo >> git commit -m "added a file"
[master f5d347d] added a file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 doNotTrack2.txt
[manuelpuchner firstRepo >> git rm --cached doNotTrack2.txt
rm 'doNotTrack2.txt'
[manuelpuchner firstRepo >> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    doNotTrack2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    doNotTrack.txt
    doNotTrack2.txt

[manuelpuchner firstRepo >> git commit -m "untracked the files"
[master 9eb5b16] untracked the files
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 doNotTrack2.txt
[manuelpuchner firstRepo >> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    doNotTrack.txt
    doNotTrack2.txt

nothing added to commit but untracked files present (use "git add" to track)
manuelpuchner firstRepo >> |
```

Dateien umbenennen

```
[manuelpuchner firstRepo >> git mv doNotTrack.txt doTrack.txt
fatal: not under version control, source=doNotTrack.txt, destination=doTrack.txt
[manuelpuchner firstRepo >> git add doNotTrack.txt doNotTrack2.txt
[manuelpuchner firstRepo >> git mv doNotTrack.txt doTrack.txt
[manuelpuchner firstRepo >> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   doNotTrack2.txt
    new file:   doTrack.txt

[manuelpuchner firstRepo >> git commit -m "changed file name"
[master b777652] changed file name
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 doNotTrack2.txt
 create mode 100644 doTrack.txt
[manuelpuchner firstRepo >> git status
On branch master
nothing to commit, working tree clean
manuelpuchner firstRepo >>
```

Änderungen einer Datei restaurieren

```
[manuelpuchner firstRepo >> git status
On branch master
nothing to commit, working tree clean
[manuelpuchner firstRepo >> nano firstFile.txt
[manuelpuchner firstRepo >> cat firstFile.txt
this is the first line in this file
this is a new line
[manuelpuchner firstRepo >> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   firstFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[manuelpuchner firstRepo >> git restore firstFile.txt
[manuelpuchner firstRepo >> git status
On branch master
nothing to commit, working tree clean
[manuelpuchner firstRepo >> cat firstFile.txt
this is the first line in this file
manuelpuchner firstRepo >> |
```

Bestehendes GitHub – Repo klonen

```
[manuelpuchner CPR_SW >> git clone git@github.com:MartinaReisHTL/CPRSW_StudentsShare.git
Cloning into 'CPRSW_StudentsShare'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 13 (delta 3), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), 1.14 MiB | 2.20 MiB/s, done.
Resolving deltas: 100% (3/3), done.
[manuelpuchner CPR_SW >> cd CPRSW_StudentsShare
[manuelpuchner CPRSW_StudentsShare (main) >> ls -a
.                .git
..               CPRSW_WS_2_Git-Praxis.pdf
manuelpuchner CPRSW_StudentsShare (main) >>
```

Cheat Sheet – Git Commands.pdf