

Terminal Funktionen

Contents

1	Teriminal Notes	1
2	Shortcuts	1
2.1	Git-Funktionen	2
2.2	Coursera course	3
2.3	Git	3

1 Teriminal Notes

Hier kommen die Notizen zu meinen Terminal Eingaben, die ich bislang kann.

- `pwd` – to see what the current working directory is.
 - `ls` – um die Unterverzeichnisse bzw. Dateien anzuzeigen.
 - `cd foo` – to change to the foo subdirectory of your working directory.
 - `cd ..` – to move up to the parent of the working directory.
 - `mkdir foo` – to create a subdirectory called foo in the working directory.
 - `pandoc --version` – um zu erfahren welche Pandoc-Version installiert ist
-

um eine Datei durch pandoc in eine html-Datei zuschreiben

```
pandoc test1.md -f markdown -t html -s -o test1.html
```

um eine Datei in eine pdf datei umzuwandeln

```
pandoc text.md -s -o test1.pdf
```

falls man eine Option/Funktion vergessen hat – hilft folgendes:

```
pandoc --help
-- oder --
man pandoc # ruft das Manual auf
```

2 Shortcuts

- up-arrow – to go back through your command history.
- mit dem Tab werden Komandos vervollständigt.
- mit `cmd d` lässt sich das Terminal splitten
- mit `Ctrl-D` kommt man aus einem Programm raus
- `ctrl-c` – zum beenden von laufenden Programmen

2.1 Git-Funktionen

```
q :q Q :Q ZZ      Exit.
```

```
cd my-project
git init
```

```
git add project.tex
```

```
git add figures/      # Adds the figures directory
git add *.tex         # Adds all .tex files in the current directory
git add .             # Adds all of the files (. is the current directory)
```

```
git commit -m "My first commit"
```

You need to run `git add` whenever you change a file, not just at the beginning. This tells git that it should include these changes in the next snapshot. Alternatively you can pass the `-a` option to `git commit` to tell it to commit all changes to files that it is monitoring:

```
git commit -a -m "My second commit"
```

```
git log
```

```
git show
```

```
git checkout f69606d7e24ad45b31bb6eb4b38192bd07f274fc # Frühere version mit dieser Nummer wird aufgerufen
```

Another common situation is you decide that you want to undo only one set of changes from a while ago. Perhaps you've edited the document in three steps: 1) Adding an abstract, 2) Updating your acknowledgements, 3) Adding in a figure. At each step you've created a new version in git, but now you decide that you didn't really want to update your acknowledgements. Unfortunately this is sandwiched between other changes so a simple rollback like before won't do. Fear not, because git is clever enough to do what you want, with the `revert` command:

```
git revert f69606d7e24ad45b31bb6eb4b38192bd07f274fc
```

```
git push origin master
```

The other possibility is you want to download changes made by someone else. This is a process which git calls 'pulling'. To do this, simply run

```
git pull origin master
```

```
git status
```

2.2 Coursera course

Struktur eines Terminal Kommandos * `command flags argument`—`cp -r Documents More_docs` ein flag wird von `-` eingeleitet.

- `/` – Root directorie
- `~` – home directorie
- `clear` – macht alles wieder leer
- `ls -a` – zeigt auch die versteckten Dateien
- `ls -al` – zeigt die details von versteckten und nicht versteckten dateien
- `cd` – direktory ändern, oder bring einem zum home direktor
- `mkdir - + name` erstellt einen neuen ordner
- `touch test_file` – erzeugt ein neue Datei `test_file`
- `cp test_file Documents` – macht eine kopie vom `test_file` in den Ordner `Documents`
- `cp -r Documents More_docs` – kopiert ein Ordner `Documents` in einen andern Ordner `More_docs`
- `rm test_file` – löschen datei
- `rm More_docs` – löscht ordner und alle Dateien darin
- `mv new_file Documents` – bewegt `new_file` zu `Documents`
- `mv new_file renamed_file` – ändert den namen von `new_file` zu `renamed_file`
- `echo Hallo welt` – zeigt hallo welt
- `date` – zeigt datum

2.3 Git

[Ein link](#)

- `git add` – alle neuen Dateien werden getrackt von version control