

# Cheatsheet Shell Commands

Seminar KED2022

Alex Flückiger

21 Oktober 2022

## Contents

1 Basic Shell Commands	1
2 NLP-related Shell Commands	2
3 Regular Expressions	2

## 1 Basic Shell Commands

---

Shell Command	Explanation
<code>cd filepath</code>	change <b>d</b> irectory aka move into a different folder
<code>ls -lh folder</code>	<b>l</b> ist the files and folders in your current <b>d</b> irectory
<code>pwd</code>	show <b>p</b> ath of <b>w</b> orking <b>d</b> irectory aka the folder that you're in right now
<code>touch fname</code>	make a new file
<code>mkdir dirname</code>	<b>m</b> ake a new <b>d</b> irectory aka a folder
<code>rm fname</code>	<b>r</b> emove aka delete a file or directory
<code>cp original-fname copied-fname</code>	<b>c</b> opy a file or directory
<code>mv original-fname new-fname</code>	<b>m</b> ove or rename a file or directory
<code>cat fname</code>	show all the contents of a file
<code>more fname</code>	show snippet of a file that allows you to scroll through the entire thing
<code>head fname</code>	show the first 10 lines of a file (change number of lines by adding a flag, e.g. <code>head -100</code> )
<code>tail fname</code>	show the last 10 lines of a file (change number of lines by adding a flag, e.g. <code>tail -100</code> )
<code>wc -w -l fname</code>	show how many <b>w</b> ords or lines in a file
<code>man command</code>	show the <b>m</b> anual aka the documentation that tells you what a particular command does
<code>echo</code>	print text to the command line
<code>grep "search pattern" fname or dirname</code>	search for lines that include search term in file. See below for the arguments of <code>grep</code> .
<code>wget url</code>	<b>g</b> et a file from the <b>w</b> eb

---

This cheatsheet is based on [this resource](#). Please also refer to this resource for a more in-dept explanation in prose. You should follow the guide for macOS and Unix even as a Windows user as we have installed a Unix environment.

## 1.1 egrep

The most common arguments of `egrep`:

- `-i` search case insensitive
- `-r` search recursively in folder
- `-o` show exact matches only instead of entire lines with matches
- `-h` suppress the file path where the match occurred

## 1.2 Operators

- `|`: A pipe takes the output of one command and passes it as the input to another.

```
echo "pass this text to next command" | cat
```

- `>`: This operator redirects the output to a file (overwrites if it already exists). Example:

```
echo "first line of file1" > file1
```

- `>>`: This operator redirects and appends the output to an *existing* file: Example:

```
echo "line following existing content of file1" >> file1
```

# 2 NLP-related Shell Commands

coming soon!

## 3 Regular Expressions

### 3.1 Example Patterns

```
# alle Kleinbuchstaben
```

```
echo "Das ist ein Satz mit der Zahl 1000" | egrep --colour "[a-z]"
```

```
# alle Grossbuchstaben
```

```
echo "Das ist ein Satz mit der Zahl 1000" | egrep --colour "[A-Z]"
```

```
# das Wort "ist" und das nächste Wort
```

```
echo "Das ist ein Satz mit der Zahl 1000" | egrep --colour "ist [a-z]*"
```

```
# das Wort "Zahl" gefolgt von einer Ziffer
```

```
echo "Das ist ein Satz mit der Zahl 1000" | egrep --colour "Zahl [0-9]"
```

```
# das Wort "Zahl" gefolgt von beliebig vielen Ziffern
```

```
echo "Das ist ein Satz mit der Zahl 1000" | egrep --colour "Zahl [0-9]*"
```

### 3.2 Pattern Equivalence

```
a+ == aa*           # "a" once or more than once
a? == (a|_)         # "a" once or nothing
a{3} == aaa         # three "a"
a{2,3} == (aa|aaa)  # two or three "a"
[ab] == (a|b)       # "a" or "b"
[0-9] == (0|1|2|3|4|5|6|7|8|9) #any digit
```