

Cheatsheet Shell Commands

Alex Flückiger

01 April 2021

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 d irectory aka move into a different folder
<code>ls -lh folder</code>	l ist the files and folders in your current d irectory
<code>pwd</code>	show p ath of w orking d irectory aka the folder that you're in right now
<code>touch fname</code>	make a new file
<code>mkdir dirname</code>	m ake a new d irectory aka a folder
<code>rm fname</code>	r emove aka delete a file or directory
<code>cp original-fname copied-fname</code>	c opy a file or directory
<code>mv original-fname new-fname</code>	m 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 w ords or lines in a file
<code>man command</code>	show the m anual aka the documentation that tells you what a particular command does
<code>echo</code>	print text to the command line
<code>grep "search term" fname or dirname</code>	search for lines that include search term in file
<code>wget url</code>	g et a file from the w 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 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. Example:

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

- >>: This operator redirects and append 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
```