Arbeiten in der Shell (Bash)

Warum? https://xkcd.com/1205/

verschiedene Shells

- bash
- /bin/sh
- zsh
- ksh
- etc.

In einer Desktop Umgebung lässt man die Shell normalerweise in einem Terminal Programm laufen. Dieses heisst z.B. Konsole/Console/Terminal etc.

cd

- cd DIR
- cd ..

ls

- ls
- ls -l
- man ls # siehe Paragraph "man"

Pfad-Vervollständigung ("globing")

-
  -

man

- man man
- man -w -a man

Docu

- /usr/share/doc

Dateien anzeigen

- cat
- less (more)

Varianten von Anzeigen

- tail
- tail -f /var/log
- head

Logs

/var/log - syslog - kern

Was geht?

- ps faux - was sieht man da? - [
- ps mit eigenen Feldern

Daemons, Kernel Threads

- /etc/init.d
- /etc/systemd/system
- systemctl list-units

Speicher und Prozesse

- smem
- top (htop, atop)

Dateien finden

- find
- find -exec
- man find
- find -newer
- find -type

Sachen in Dateien finden

- grep (ack, rg)
- man 7 regex

Paketverwaltung

- http://packages.debian.org

Debian/Ubuntu...	RedHat/Fedora/SuSE/...

dpkg -i	rpm -i
dpkg -P	
dpkg -r	rpm -e
dpkg -s	rpm -qi
dpkg -S	rpm -qf
dpkg -L	rpm -ql
apt install	yum install / yum update
apt remove	yum remove

- aptitude
- rpm/yum -> dnf

Tab Completion

- bash-completion
- CTRL-r
- TAB-TAB
- $PATH

Command Options

- short options
    - dpkg -i
- long options
    - dpkg --install
- sub-commands
    - apt-get install

Umleiten

- >
- <
- 2>
-

Iterieren

- ls | while read x; do irgend "$x"; was; done # Achtung...
- for i in 1 2 3; do was $i; anderes $i; done

Variablen

- A=7
- a=7
- a="a b c"

Quoting

- for i in *seq 1 10*
- for i in $(seq 1 10)
- "$foo" - foo="a b"
- '$foo'
- '"'
- Space als Separator

Scripte Schreiben

- history

Editoren

- nano
- vim - i - Esc - :w - :q!

Hashbang

- #!

Filesystem Layout

- tree -L 1 /

  /etc /bin, /usr, /lib, /boot /var /mnt /media /dev /sys /proc /proc/id /home ~/.dotfiles ~/.config ~/.cache ~/.local -> daten

Skript anschauen

- /etc/init.d/*

SSH

- ssh
- sshfs

sed

awk, perl

Othogonalität

- ssh + shell