

# Shell Basics

<http://www.gnu.org/software/bash/manual/bashref.html>

<https://www.shellscript.sh/>

<http://www.linux-services.org/shell/>

## GB keyboard

~	!	"	£	\$	%	^	&	*	(	)	-	+	Backspace
`	1	2	3	4 €	5	6	7	8	9	0	-	=	Enter
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}	Enter
Caps Lock	A	S	D	F	G	H	J	K	L	:	@	~	Enter
Shift		Z	X	C	V	B	N	M	<	>	?	Shift	
Ctrl	Win Key	Alt							Alt Gr	Win Key	Menu	Ctrl	

## Screen

Siehe auch <https://www.mattcutts.com/blog/a-quick-tutorial-on-screen/>

### Screen Arbeitsablauf

Normalerweise im User, nicht mit sudo

`screen -S [name]`

alternativ `screen -R` (erzeugt neue Session wenn noch keine da oder re-attached an vorhandene)

Weitere Screen Session wird mit `Ctrl + a` eröffnet.

Wechseln zwischen den Sessions mit `Control-a [Nummer]`, wobei 0 die erste ist

Hilfe: `Ctrl A ?`

Liste aller Fenster in einer Session `Ctrl A w`

Von Session abmelden

`Ctrl-A d`

Danach aus ssh abmelden, die screen session läuft weiter. Von einem anderen Ort wieder per ssh auf

die Arbeitsmaschine und dann wieder an die Session binden

`screen -d -R sessionname` will resume a session that you started in a different location.

### Alle Screen Sessions auflisten

`screen -ls`

### Session killen

`$ screen -X -S [session # you want to kill] quit`

## Shadow

Einfügen eines Passwort hashes

```
thommie@odysseus3:~> python -c "import crypt, getpass, pwd; \>
print crypt.crypt('[klartext]', '\$6\$SALTsalt\$')"
```

`\$6\$SALTsalt\$0[hashwert]`

## update-alternatives

<https://manpages.ubuntu.com/manpages/trusty/de/man8/update-alternatives.8.html>

## Random string erzeugen

```
tr -dc A-Za-z0-9 </dev/urandom | head -c 25 ; echo ''
```

## Links

"In -s /Zieldatei /Referenz

## Find

Alle Dateien älter als 365 Tage finden

```
find /mnt/pve/sb_u203461/dump/ -mtime +365 -type f
```

Mit löschen:

```
find /mnt/pve/sb_u203461/dump/ -mtime +365 -type f -delete
```

Luxus Variante mit Backup

```
#!/bin/bash
path="/data/backuplog/"
timestamp=$(date +%Y%m%d_%H%M%S)
filename=log_${timestamp}.txt
log=$path$filename
days=7

START_TIME=$(date +%s)

find $path -maxdepth 1 -name "*.txt" -type f -mtime +$days -print -delete
>> $log

echo "Backup:: Script Start -- $(date +%Y%m%d_%H%M)" >> $log

... code for backup ...or any other operation .... >> $log

END_TIME=$(date +%s)

ELAPSED_TIME=$(( $END_TIME - $START_TIME ))

echo "Backup :: Script End -- $(date +%Y%m%d_%H%M)" >> $log
echo "Elapsed Time :: $(date -d 00:00:$ELAPSED_TIME +%Hh:%Mm:%Ss) " >>
$log
```

## sed

replace "-d" with "-d "

```
<font inherit/monospace;;inherit;;#000000background-color:#ffffff;>sed s/-
d/'-d '/g domains3.txt> domains4.txt</font>
```

trim string and remove newlines

```
<font inherit/monospace;;inherit;;#000000background-color:#ffffff;>tr --
delete '\n' <domains2.csv> domains3.txt</font>
```

From:

<https://wiki.netzwissen.de/> - **netzwissen.de Wiki**

Permanent link:

<https://wiki.netzwissen.de/doku.php?id=shell&rev=1728801997>

Last update: **13/10/2024 - 06:46**

