

Scripting 1

TSBE Frühlingssemester 2016

<http://smlz.github.io/tsbe-2016fs/scri/>

Marco Schmalz

marco.schmalz@gibb.ch



1 / 15

Kursübersicht

1. **Bash und UNIX-Tools**
2. Einführung in Python
3. Python: Datei und Prozessoperationen
4. (Auffahrt)
5. Repetition

2 / 15

Heute

1. Administratives
2. Gruppenarbeit zu Bash
3. Übung Bash
4. Einführung in Python

3 / 15

Gruppenpuzzle (30')

- If und test ([])
- while loop
- find
- for loops
- sed/grep/cut

4 / 15

Einstiegsübung (5')

Schreiben sie ein Skript, welches den Benutzer nach dem Jahrgang fragt, diesen in einer Variabel abspeichert, und danach das Alter ausgibt.

Tipps:

- `read name`

speichert die Benutzereingabe unter `name` ab.

- `man echo`

Gibt Auskunft, wie man Zeilenumbrüche bei der Ausgabe verhindern kann.

5 / 15

Verzweigung - IF

```
if [ "$a" -gt 0 ]
then
  if [ "$a" -lt 5 ]
  then
    echo "Der Wert von \"$a\" liegt zwischen 0 und 5."
  fi
fi
```

Gleiches Resultat mit

```
if [ "$a" -gt 0 ] && [ "$a" -lt 5 ]
then
  echo "Der Wert von \"$a\" liegt zwischen 0 und 5."
fi
```

6 / 15

Werte testen mit [

Achtung: [ist ein Programm. Darum muss immer ein Leerschlag nach [folgen.

Siehe auch:

```
man [
```

Achtung:

- [ist ein Programm!
- Ebenso wie `true` und `false`

7 / 15

Strings vergleichen:

```
if [ "$a" = "$b" ]  
then  
    echo "a und b sind gleich"  
fi
```

- `=`: Gleich
- `!=`: Ungleich
- `[-n "$a"]`: `$a` ist **nicht** ein leerer String
- `[-z "$a"]`: `$a` **ist** ein leerer String

8 / 15

Zahlen vergleichen:

```
if [ "$a" -gt "$b" ]
then
  echo "a ist grösser als b"
elif [ "$a" -eq "$b" ]
then
  echo "a ist gleich gross wie b"
else
  echo "a ist kleiner als b"
fi
```

- -eq (equal): Gleich
- -ne (not equal): Ungleich
- -gt (greater than): Grösser
- -ge (greater or equal): Grösser oder gleich
- -lt (lesser than): Kleiner
- -le (lesser or equal): Kleiner oder gleich

9 / 15

Tests mit Dateien

```
if [ -e "$file" ]
then
  echo "Datei $file existiert"
fi
```

- -e (exists): Datei existiert
- -f (file exists): Datei existiert und ist eine normale Datei
- -d (directory exists): Datei existiert und ist ein Verzeichnis
- -w (writable): Datei existiert darf geschrieben werden
- -x (eXecute): Datei existiert und darf ausgeführt werden

10 / 15

Übung (15')

Erweitern sie ihr Script von vorher:

- Entscheiden sie anhand des Jahrgangs ob der Benutzer volljährig ist
- Falls ja:
 - Überprüfen sie ob eine ausführbare Datei mit dem Namen `autofahren` existiert.
 - Erstellen sie die Datei automatisch, fall sie nicht existiert.
 - Machen sie falls nötig die Datei ausführbar.
 - Führen sie die Datei `autofahren` aus.
 - `autofahren` soll "WROOOM" auf dem Bildschirm ausgeben
- Falls nein:
 - Überprüfen sie ob eine ausführbare Datei mit dem Namen `velofahren` existiert.

11 / 15

Schleifen - for

Beispiel:

```
for file in *.JPEG
do
  echo mv "$file" "${file/.JPEG/.jpg}"
done
```

- Dateinamen-Komplementierung mit * (Stern)

12 / 15

Übung

Verschieben sie alle PNG-Bilder im aktuellen Verzeichnis in ein Unterverzeichnis mit den Name `bilder`.

13 / 15

Schleifen - while

Beispiel:

```
while true
do
  echo "Endlosschleife"
done
```

- Bedingung gleich wie bei `if`

14 / 15

Übung

Geben sie mit einem Skript die endlose Fibonacci-Reihe aus:

- 1
- 1
- 2
- 3
- 5
- 8
- 13
- 21
- ...