

Aufgabe 3 - Eigene Tools entwickeln

Szenario

Um schneller auf verschiedene Cheat Sheets zugreifen zu koennen, macht es Sinn, sich ein eigenes Tool zu entwickeln.

Grundlagen

- Mit dem HTTP Endpunkt cheat.sh kann man schnell einfach verstaendliche Informationen finden. Das geht zum Beispiel mit dem `curl` command

```
curl cht.sh/curl
```

- Jetzt solltest du verschiedene Beispiele zum Umgang mit `curl` gezeigt bekommen

1. Erstellung des Scripts

- Erstelle einen neuen Ordner und darin eine neue Datei mit dem Namen `cheat.sh`

```
#!/bin/bash  
  
curl cht.sh/curl
```

- Führe das Script aus (vergesse nicht, die entsprechenden Berechtigungen zu setzen)

2. Commandline Argument verwenden

- Ändere das Script so, dass nicht mehr curl verwendet wird sondern das Commandline Argument was übergeben wird

```
# So soll die Datei aufgerufen werden  
./cheat.sh cat
```

3. Output in CLI Pager (less)

- Der Output des curl commands, soll ab jetzt mit einem command line pager (zB. `less`) angezeigt werden

4. Funktion

- Schreibe das Script so um, dass die Logik innerhalb einer Funktion aufgerufen wird
- Nenne die Funktion `hilfe`
- Falls du eine Wiederholung brauchst: <https://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO-8.html>

```
function test {  
    echo $1  
}  
  
test "Hallo"  
# Hallo
```

5. Bashrc

- Um die Funktion jetzt in unserer Shell einfach einsetzen zu können, müssen wir die Datei erst zum PATH hinzufügen
- Lese hier mehr über die PATH Environment Variable: [Link](#)

```
source cheat.sh
```

- Du kannst diese Zeile auch zu deiner `.bashrc` hinzufügen, falls du diese Änderung persistieren magst (Achte aber darauf dann den absoluten Pfad anzugeben)
- Jetzt kannst du einfach über das `hilfe` command mehr infos zu allen core-utils bekommen

```
hilfe xargs  
hilfe grep
```

6. Freiwillig - Fuzzy Finding

- Mit einem Fuzzy Finder kann man aus einer Liste schnell etwas auswählen
- Ein einfacher Fuzzy Finder ist `fzf` [Link](#)
- Man kann einfach eine Liste in `fzf` hinein pipen und `fzf` gibt die Auswahl des Users wieder zurück

```
selected=$(echo "Ja\nNein" | fzf)  
# Nutzer kann jetzt Ja oder Nein auswählen, was dann in der selected Variable gespeichert wird
```

- Baue das Script so um, dass der User ein command (von allen, die in `/bin` gespeichert sind) mittels `fzf` auswählen kann