

Aufgabe 1

```
1 #!/bin/bash
2
3 hobbies=("Code" "Eat" "Sleep" "Repeat")
4
5 echo "Mein Tagesablauf:"
6 for i in ${hobbies[@]}; do
7     echo "$i"
8 done
9
```

julian@julian-VirtualBox:~/Scripts\$./hobbies.sh

Mein Tagesablauf:
Code
Eat
Sleep
Repeat

Zusatzaufgabe

```
1 #!/bin/bash
2
3 hobbies=()
4
5 read -p "Was ist dein liebstes Hobby?" wert1
6 read -p "Was machst du zur Mittagszeit?" wert2
7 read -p "Was machst du, wenn du müde wirst?" wert3
8 read -p "Wie startest du in den nächsten Tag?" wert4
9
10 hobbies+=("$wert1" "$wert2" "$wert3" "$wert4")
11
12 echo "Dein aktueller Tagesablauf:"
13 for i in ${hobbies[@]}; do
14     echo "$i"
15 done
16
17
```

julian@julian-VirtualBox:~/Scripts\$./hobbies.sh

Was ist dein liebstes Hobby?Code
Was machst du zur Mittagszeit?Eat
Was machst du, wenn du müde wirst?Sleep
Wie startest du in den nächsten Tag?Repeat
Dein aktueller Tagesablauf:
Code
Eat
Sleep
Repeat

Aufgabe 2

Den Großteil meiner Zeit habe ich damit verbracht, mein Script für Datensicherungen zu verfeinern.

```
1 #!/bin/bash
2
3 config_file="/home/$USER/Project/config.txt"
4
5 create_project_folder() {
6     if [ ! -d "/home/$USER/Project" ]; then
7         mkdir -p /home/$USER/Project
8     fi
9 }
10
11 create_config_file() {
12     if [ ! -f "$config_file" ]; then
13         touch "$config_file"
14     fi
15 }
16
17 create_directory_path() {
18     echo "Gib bitte den absoluten Pfad für die zu sichernden Dateien an:"
19     read directory_path
20     if [ -d "$directory_path" ]; then
21         echo "$directory_path" > "$config_file"
22     else
23         echo "Das von dir angegebene Verzeichnis existiert nicht."
24         create_directory_path
25     fi
26 }
27
28 create_backup_path() {
29     echo "Gib bitte den absoluten Pfad für den Speicherort des Backups an:"
30     read backup_path
31     if [ ! -d "$backup_path" ]; then
32         echo "Der angegebene Pfad existiert nicht und wird erstellt."
33         mkdir -p "$backup_path"
34     fi
35     echo "$backup_path" >> "$config_file"
36 }
37
38 a=true
39 while $a; do
40     if [ -f "$config_file" ]; then
41         paths=()
42         while IFS= read -r line; do
43             paths+=("$line")
44         done < "$config_file"
45
46         directory_path="${paths[0]}"
47         backup_path="${paths[1]}"
48         base_name=$(basename "$directory_path")
49         current_date=$(date +%Y-%m-%d)
50         backup_name="${base_name}_Backup"
51         if [ -n "$directory_path" ] && [ -n "$backup_path" ]; then
52             mkdir -p "$backup_path/$backup_name" && tar -cJvf "$backup_path/$backup_name/${backup_name}_${current_date}.tar.xz" "$directory_path"
53         fi
54     else
55         echo "Es besteht leider keine Konfigurationsdatei, Dateien werden erstellt."
56         create_project_folder
57         echo "Ordner wurde unter /home/$USER/Project erstellt"
58         create_config_file
59         echo "Konfigurationsdatei wurde in /home/$USER/Project erstellt"
60         create_directory_path
61         echo "Die zu sichernden Dateien befinden sich unter: $directory_path"
62         create_backup_path
63         echo "Die Dateien werden unter $backup_path archiviert"
64     fi
65     a=false
66 done
```

Der Plan ist, dass dieses Script später automatisiert nutzbar gemacht werden kann. Weitergehend möchte ich mit if-Bedingungen Argumente nutzbar machen, damit der Backup-Pfad und der Pfad für das zu sichernde Verzeichnis geändert werden können. Grundsätzlich habe ich mich etwas mit Funktionen beschäftigt und gestern schon herausgefunden, dass diese durch Bedingungen getriggert werden müssen. Die angegebenen Funktionen werden dann getriggert, sobald keine Config-Datei gegeben ist. Mit den Funktionen wird quasi ein Installationsverzeichnis erstellt, worin dann eine Config-Datei abgelegt wird. Die while-Loop ist jetzt erstmal sporadisch gelöst worden, mit der Variabel a=true, die auf false wechselt, sobald das Archiv erstellt wurde. Problem vorher war, dass beim Ausführen eine Dauerschleife stattgefunden hat, weil die Bedingung noch

nicht ganz ausgeklügelt ist.

In der if-Bedingung kam es auch zu Herausforderungen, weil die Config nicht richtig ausgelesen wurde. Ich bin der Meinung, dass die config auch leichter ausgelesen werden könnte, habe aber noch nicht genauer nachgeforscht, da es erstmal funktioniert (lachend). Jedenfalls wird die config in ein Array eingeordnet und weist dem Inhalt einen Index zu. Im nächsten Schritt bekommt dann `directory_Path` den Index 0 und `backup_path` den Index 1. Dann habe ich `basename` benutzt, um mir vom `directory_path` lediglich das letzte Verzeichnis darstellen zu lassen, um den Namen so kurz wie möglich zu halten.

Natürlich habe ich das nicht alles aus dem Kopf hinbekommen, ich habe ChatGPT benutzt, um Code erklärt zu bekommen oder um den Text auf Zeichenfehler zu setzen. Häufig habe ich das `$`-Zeichen in eine Klammer gesetzt oder irgendwo ein Leerzeichen gehabt. Hubertus hat mir die Empfehlung gegeben, das Installationsverzeichnis erstmal zu fixieren, anstatt dem Nutzer die Möglichkeit geben, es selbst zu bestimmen. Suheib hat mir am Ende noch den Tipp mit der Variable `a=true` gegeben.

Am Ende soll es mit cron automatisiert zu festen Zeiten abgerufen werden. z.B. 1x täglich usw. Oder wenn das Verständnis da ist, auch auf einem Cloudspeicher gebracht werden. Lange Rede kurzer Sinn, es ist noch lange nicht fertig.