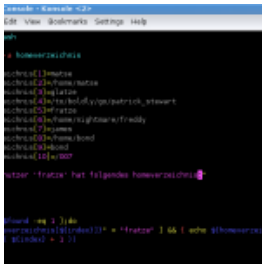


Assoziative Arrays in Bash



```
fratze@kali:~$ declare -A homeverzeichnis
fratze@kali:~$ homeverzeichnis[1]=/home/1
fratze@kali:~$ homeverzeichnis[2]=/home/2
fratze@kali:~$ homeverzeichnis[3]=/home/3
fratze@kali:~$ homeverzeichnis[4]=/home/4
fratze@kali:~$ homeverzeichnis[5]=/home/5
fratze@kali:~$ homeverzeichnis[fratze]=/home/fratze
fratze@kali:~$ echo ${homeverzeichnis[fratze]}
/home/fratze
```

Mit bash 4.0 und höher hat man die Möglichkeit Assoziative Arrays verwenden zu können. In Python werden diese Art von Datenstrukturen Dictionaries genannt, in Perl heissen diese Hashes. Im Prinzip sind assoziative Arrays eine Erweiterung von normalen (indizierten) Arrays. Diese werden in der Regel dann eingesetzt, wenn man Arrayeinträge über vorher definierte Zeichenketten wiederfinden möchte.

Der Unterschied zu den indizierten Arrays besteht darin, dass bei den assoziativen Arrays nicht fortlaufende Nummern als Indizes, sondern sogenannte Keys (engl. Schlüssel) benutzt werden, die neben Zahlen auch Zeichenketten beinhalten können. Die einzelnen Arrayelemente werden demnach nicht einfach durchnummeriert, sondern jedes Element erhält eine Zeichenkette als Index.

Wollte man z.B. die Homeverzeichnisse von fünf Benutzern in einem Array abspeichern, musste man ein wenig tricksen um diese Information pro Benutzer auch wieder auslesen zu können. Im folgenden Beispiel wird einfach definiert, dass in den ungeraden Indices die Benutzernamen stehen und in den geraden Indices die Pfade zu den Homeverzeichnissen.

```
#!/bin/bash
```

```
declare -a homeverzeichnis
```

```
homeverzeichnis[1]=matse
homeverzeichnis[2]=/home/matse
homeverzeichnis[3]=glatze
homeverzeichnis[4]=/to/boldly/go/patrick_stewart
homeverzeichnis[5]=fratze
homeverzeichnis[6]=/home/nightmare/freddy
homeverzeichnis[7]=james
homeverzeichnis[8]=/home/bond
homeverzeichnis[9]=bond
homeverzeichnis[10]=/007
```

```
echo "Benutzer 'fratze' hat folgendes homeverzeichnis:"
```

```
found=0
index=1
```

```
until [ $found -eq 1 ];do
if [ "${homeverzeichnis[${index}]}" = "fratze" ];then
echo ${homeverzeichnis[`${index} + 1`]}
found=1
fi
index=$(( ${index} + 1 ))
done
```

Wie man sieht, ist das relativ umständlich. Einfacher geht dies seit Bash 4.0. Hierbei deklariert man zuerst ein Assoziatives Array mittels:

```
declare -A homeverzeichnis
```

Danach kann man einfach das Array füllen indem man als Schlüssel den Benutzernamen und als Wert den Pfad zum Homeverzeichnis definiert.

```
homeverzeichnis[matse]=/home/matse
homeverzeichnis[glatze]=/to/boldly/go/patrick_stewart
[etc.]
```

Das fertige Script sieht dann folgendermassen aus:

```
#!/bin/bash
```

```
declare -A homeverzeichnis
```

```
homeverzeichnis[matse]=/home/matse
```

```
homeverzeichnis[glatze]=/to/boldly/go/patrick_stewart
```

```
homeverzeichnis[fratze]=/home/nightmare/freddy
```

```
homeverzeichnis[james]=/home/bond
```

```
homeverzeichnis[bond]=/007
```

```
echo "Benutzer 'fratze' hat folgendes homeverzeichnis:"
```

```
echo "${homeverzeichnis[fratze]}"
```

Zum Schluss noch ein kleiner Tipp. Will man jeweils die Liste aller Schlüssel oder Werte erhalten, kann man folgendermassen vorgehen:

Eine Liste aller Keys eines Arrays erhält man mittels:

```
echo ${!homeverzeichnis[@]}
```

Eine Liste aller Werte eines Arrays erhält man mittels:

```
echo ${homeverzeichnis[@]}
```