

- Zur Verarbeitung von Listen werden Schleifen benötigt. Beispiele:

```
zahlen = [5, 7, 6, -10, 0, 77, 21, 23, 78, -1]
```

```
summe = 0
```

```
for i in zahlen:
```

```
    summe = summe + i
```

```
print(summe)
```

```
zahlen_abs = [ ]
```

```
for i in zahlen:
```

```
    zahlen_abs.append(abs(i))
```

```
print(zahlen_abs)
```

- Im ersten Beispiel wird aus der Liste ein Ergebniswert berechnet, im zweiten Beispiel wird die Liste in eine andere Liste umgewandelt.
- Für beide Anwendungsfälle bietet Python einige eingebaute Funktionen, z.B. sum, min, max, sorted.

- Freilich kann man auch eigene Funktionen definieren. Beispiel:

```
def abs_sum(z):  
    summe = 0  
    for i in z:  
        summe = summe + abs(i)  
    return summe  
print(abs_sum(zahlen))
```

- Für eine Umwandlung einer Liste in eine andere Liste gibt es in Python die Funktion map:

```
map_objekt = map(funktion, sequenz)  
  
liste_neu = list(map(funktion, sequenz))    # Umwandlung des map_objekts in eine Liste
```

- Beispiel:

```
zahlen_abs = list(map(abs, zahlen))  
print(zahlen_abs)
```

- Als Funktion sind natürlich auch lambda-Funktionen erlaubt. Beispiel:

```
zahlen_abs = list(map(lambda x: abs(x - 5), zahlen))  
print(zahlen_abs)           # Ausgabe: [0, 2, 1, 15, 5, 72, 16, 18, 73, 6]
```

- Listen können auch in Listen eines anderen Datentyps umgewandelt werden. Beispiel:

```
zahlen_abs = list(map(lambda x: x > 5, zahlen))  
print(zahlen_abs)           # Ausgabe: [False, True, True, False, False, True, True, True, True, False]
```

-  Kapitel 30.2 in (Klein 2018)



1. Schreiben Sie ein Skript ohne Schleifen, sondern unter Verwendung der map-Funktion, das in einer Liste von Strings alle Listeneinträge durch die Zahlen 0 oder 1 ersetzt, je nachdem, ob der Listeneintrag gleich '_' ist oder nicht. D.h. '_' wird in 0 umgewandelt, alle anderen Strings werden zu 1.

Beispiel:

Vorher: `zeile = ['_', 'A', '_', 'A', 'B', 'A', '_']`

Nach Ausführung des Skripts:

```
print(zeile)           # Ausgabe: [0, 1, 0, 1, 1, 1, 0]
```

Die Lösung hat nur eine Programmzeile!



2. Im Anwendungsbeispiel Vier Gewinnt war das Spielfeld eine zweidimensionale Liste mit den möglichen Einträgen 'A', 'B' oder '_'.

Schreiben Sie ein Skript ohne Schleifen, sondern unter Verwendung der map-Funktion, das alle Einträge durch 0 oder 1 ersetzt, je nachdem, ob der Eintrag gleich '_' ist oder nicht.

Beispiel:

Vorher:

```
sf = [ ['_', '_', '_', '_', '_', '_', '_'],  
        ['_', '_', '_', '_', '_', '_', '_'],  
        ['_', '_', '_', '_', '_', '_', '_'],  
        ['_', '_', '_', '_', 'A', 'B', '_'],  
        ['_', 'A', '_', 'A', 'B', 'A', '_'],  
        ['B', 'B', 'A', 'B', 'B', 'A', '_'] ]
```

Nach Ausführung des Skripts:

```
print(sf)  
# Ausgabe: [ [0,0,0,0,0,0,0],  
              [0,0,0,0,0,0,0],  
              [0,0,0,0,0,0,0],  
              [0,0,0,0,1,1,0],  
              [0,1,0,1,1,1,0],  
              [1,1,1,1,1,1,0] ]
```

Auch diese Lösung hat nur eine Programmzeile!