

Loops in Python

For und while

Einführung

Schleifen sind eine Kontrollstruktur in Programmiersprachen, um Blöcke wiederholt auszuführen.

Python kennt zwei Arten von Schleifen:

Sammlungskontrollierte Schleifen (for) und **bedingungskontrollierte** Schleifen (while).

Der For-Loop

For-Schleifen in Python sind sogenannte **Sammlungskontrollierte Schleifen**. Solange ein Element in der Sammlung (Liste, Dict..) vorhanden ist, wird die Schleife ausgeführt.

Dabei wird pro Iteration ein Element aus der Sammlung entfernt und dem Schleifenkörper zur Verfügung gestellt.

Das allgemeine Schema:

for Element in Sammlung:

Schleifenkörper, mit verfügbarem **Element** aus der **Sammlung**

for fruit in ['Banane', 'Birne', 'Kirsche']:

print(fruit)

Enumeration

Um den Index bzw. die Zahl der Iteration benötigt, können wir mit der `enumerate`-Funktion arbeiten. Diese gibt uns als erste Variable pro Iteration den aktuellen Schleifendurchlauf zurück

```
For index, card in enumerate(["K","Q","A","J"]):  
    print(index) # ergibt 0, 1, 2, 3
```

Das Keyword break

Um eine Schleife vorzeitig zu verlassen, kann das **break** - Keyword verwendet werden. Beim Auftreffen des Programmzeigers auf das Keyword break, wird die Schleife abgebrochen und das Programm weiter ausgeführt.

```
for p in [1, 2 ,3, 4]:  
    if p == 2:  
        break  
    print("Hier geht es weiter...")
```

das Keyword continue

Um einen Schleifendurchlauf vorzeitig zu verlassen, kann das **continue** - Keyword verwendet werden.

```
x = 0
```

```
for p in [1, 2 ,3]:
```

```
    if p == 2:
```

```
        continue;
```

```
        x += p # kann auch als x = x + p geschrieben werden
```

x ist 4, da bei 2 continue ausgeführt wurde und die Berechnung nicht stattfand.

While

Der While-Loop ist eine sogenannte **bedingungskontrollierte Schleife**.
Solange die definierte Bedingung wahr ist, wird der Schleifenkörper ausgeführt.

Das Schema:

While **BEDINGUNG WAHR:**

führe aus

Beispiel für eine While-Schleife

`x = 0`

`y = 10`

`while x < y:`

`x += 1`

`print(x)`