



Python – Programmieren – For-Schleifen



Eine *Schleife* (englisch *loop*) ist eine *Kontrollstruktur* in Programmiersprachen. Sie wiederholt einen Anweisungs-Block – den sogenannten *Schleifenrumpf/-körper* –, solange die *Schleifenbedingung* gültig bleibt.

Eine *for-each-Schleife* (in Python einfach nur *for-Schleife* genannt) ist ein Sonderfall: Auch hier gibt es zwar einen Anweisungs-Block (*Schleifenrumpf/-körper*), dieser wird allerdings für jedes Element einer angegebenen Liste einmal ausgeführt.

Mögliche Hilfsquellen und Tutorials finden Sie unten im Dokument.

Schwierigkeitsgrad: Einfach

Aufgabe 1 – Wiederholte Codeausführung

Führen Sie mithilfe einer For-Schleife einen Codeabschnitt Ihrer Wahl fünfmal aus.

Aufgabe 2 – Wiederholte Begrüßung

Das Programm soll den Benutzer auffordern einzugeben, wie oft er begrüßt werden möchte. Das Programm soll ihn anschließend entsprechend oft mit „Hallo“ begrüßen.

Aufgabe 3 – Zahlenfolgen ausgeben

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben. Anschließend soll das Programm alle natürlichen Zahlen von 1 bis n nacheinander ausgeben.

Aufgabe 4 – Zahlenfolgen rückwärts ausgeben

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben. Anschließend soll das Programm alle natürlichen Zahlen von n bis 1 nacheinander ausgeben.

Aufgabe 5 – Zahlenfolgen aufsummieren

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben. Anschließend soll das Programm alle natürlichen Zahlen von 1 bis n aufsummieren und die Summe dann auf der Konsole ausgeben.

Beispiel: Wenn der Benutzer 6 eingibt, soll das Programm als Ergebnis 21 ausgeben (weil $1+2+3+4+5+6=21$).

Schwierigkeitsgrad: Mittel

Aufgabe 6 – Zahlenfolgen filtern und ausgeben

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben.

Anschließend soll das Programm alle natürlichen Zahlen von 1 bis n ausgeben, die durch 3 teilbar sind.

Beispiel: Wenn der Benutzer 6 eingibt, soll das Programm „3“ und „6“ ausgeben.

Aufgabe 7 – Zahlenfolgen filtern und aufsummieren

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben.

Anschließend soll das Programm alle geraden natürlichen Zahlen von 1 bis n aufsummieren und die Summe dann auf der Konsole ausgeben.

Beispiel: Wenn der Benutzer 6 eingibt, soll das Programm als Ergebnis 12 ausgeben (weil $2+4+6=12$).

Aufgabe 8 – Zahlenfolgen filtern und aufsummieren

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben.

Anschließend soll das Programm alle ungeraden natürlichen Zahlen von 1 bis n aufsummieren und die Summe dann auf der Konsole ausgeben.

Beispiel: Wenn der Benutzer 6 eingibt, soll das Programm als Ergebnis 9 ausgeben (weil $1+3+5=9$).

Aufgabe 9 – Zahlen multiplizieren

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben.

Anschließend soll das Programm alle natürlichen Zahlen von 1 bis n miteinander multiplizieren (das nennt man auch „die Fakultät bilden“) und das Produkt dann auf der Konsole ausgeben.

Beispiel: Wenn der Benutzer 6 eingibt, soll das Programm als Ergebnis 720 ausgeben (weil $1*2*3*4*5*6=720$).

Schwierigkeitsgrad: Fortgeschritten

Aufgabe 10 – Primzahlermittler

Das Programm soll den Benutzer auffordern eine Ganzzahl „n“ einzugeben.

Das Programm soll ermitteln, ob es sich bei der eingegebenen Zahl um eine Primzahl handelt.

Das Programm soll in einem vollständigen Satz antworten, was der Primzahltest ergeben hat.

Beispiel: Wenn n=17 ist, soll das Programm ausgeben „die eingegebene Zahl ist eine Primzahl“.

Aufgabe 11 – Verschachtelte Schleifen

Lösen Sie die beiden Aufgaben mit dem Namen „Verschachtelte Schleifen“ auf dem Übungsblatt zu while-Schleifen mit for-Schleifen.

Aufgabe 12 – Verschachtelte Schleifen

Gegeben sind zwei Listen:

adjektive=["rote", "gelbe", "grüne", "blaue", "große", "kleine"]

fruechte=["Äpfel", "Bananen", "Birnen", "Grapefruits", "Kaktusfeige"]

Geben Sie mithilfe verschachtelter Schleifen jede Kombinationsmöglichkeit von Elementen der ersten mit Elementen der zweiten Liste aus.

Es sollen jedoch nur solche Kombinationen ausgegeben werden, bei denen das Adjektiv weniger Buchstaben hat als die Frucht, und bei denen das Adjektiv einen anderen Anfangsbuchstaben hat als die Frucht.

Beispiel:

Ausgegeben werden sollen Kombinationen wie „Rote Äpfel“, „grüne Kaktusfeigen“, usw.

Nicht ausgegeben werden sollen Kombinationen wie „Gelbe Äpfel“, „Blaue Bananen“, usw.

Mögliche Quellen:

- https://www.w3schools.com/python/python_for_loops.asp (englisch)
- <https://www.programiz.com/python-programming/for-loop> (englisch)
- <https://www.ionos.de/digitalguide/websites/web-entwicklung/for-loop-in-python/> (deutsch)
- <https://www.youtube.com/watch?v=ISo1uqLcVw8> (und Folgevideo 17/24, deutsch)