

## Aufgaben

1. Summe aller Zahlen von 1 bis  $n$

Schreiben Sie eine Funktion, welche alle Zahlen bis zu einer Zahl  $n$  aufaddiert, welche Sie der Funktion als Parameter übergeben.

Hinweis: In Übung 6 haben wir diese Aufgabe mit der `for`-Schleife realisiert. Nutzen Sie diesmal die `while`-Schleife.

2. Nutzereingaben prüfen und Schleifen anwenden

Schreiben Sie eine Funktion, bei welcher eine ganze Zahl vom Nutzer eingelesen wird. Prüfen Sie die Eingabe des Nutzers auf Gültigkeit und wiederholen Sie die Nutzereingabe so lange, bis eine gültige Zahl eingegeben wurde. Geben Sie in diesem Fall einen Hinweis an den Nutzer, dass eine Zahl eingegeben werden muss.

Hinweis: Sie können mit der Funktion `isnumeric()` einen String prüfen, ob dieser numerisch ist.

Beispiel für einen String `zeichenkette`: `zeichenkette.isnumeric()`

3. Vergleich von `while`- und `for`-Schleifen (1)

Geben Sie die Zahlen von 1 bis 10 mit einer Schleife aus. Schreiben Sie hierzu zwei Funktionen: eine, die eine `for`-Schleife und eine, die eine `while`-Schleife verwendet. Welche Schleife finden Sie einfacher?

4. Vergleich von `while`- und `for`-Schleifen (2)

Schreiben Sie eine Funktion, welche vom Nutzer wiederholt Zahlen einliest, bis die Summe der eingegebenen Zahlen 100 überschreitet. Nutzen Sie hierzu zunächst eine `for`-Schleife und anschließend eine `while`-Schleife.

Hinweis: Um eine `for`-Schleife abubrechen, können Sie den Befehl `break` verwenden.

Welche Schleife finden Sie einfacher?

5. Die Fibonacci-Folge berechnen 0,1,1,2,3,5,8,13, ...

Schreiben Sie eine Funktion, welche  $n$  Elemente der Fibonacci-Folge berechnet und auf dem Bildschirm ausgibt. Die Fibonacci-Folge beginnt mit den Zahlen 0 und 1 und jedes weitere Element  $f(i)$  der Reihe ist die Summe der beiden vorherigen Elemente  $f(i-1)$  und  $f(i-2)$ , d.h.  $f(i) = f(i-1) + f(i-2)$ .

Hinweis: Sie können die letzten beiden Zahlen der Folge einfach in zwei Variablen speichern und dann jedes Mal bei der Berechnung des nächsten Elementes die Werte aktualisieren.

6. Optional: Eine Steuerung für die Turtle programmieren

Schreiben Sie ein Programm, mit welchem Sie die Turtle steuern können. Hierzu soll über Tastatureingaben vom Nutzer die Turtle bewegt werden. Folgende Befehle sollen über Nutzereingaben umgesetzt werden:

f	Vorwärtsbewegung um 100
b	Rückwärtsbewegung um 100
l	Drehung um 45° nach links
r	Drehung um 45° nach rechts
q	Ende des Programms

7. Optional: Die Steuerung der Turtle erweitern

Erweitern Sie den Befehlssatz der Turtle so, dass neben den einfachen Befehlen auch Längen- bzw. Gradangaben eingelesen werden können. Die Syntax eines Befehls soll dabei wie folgt aussehen:

"Befehl.Wert"

D.h. der Wert wird durch einen Punkt getrennt an den Befehl eingegeben und soll ausgewertet werden.

Beispiele:

"b.23"	bewege die Turtle um 23 Punkte rückwärts
"r.90"	drehe Turtle um 90° nach rechts

8. Schon fertig? Dann können Sie die Turtle-Steuerung noch um weitere sinnvolle und weniger sinnvolle Befehle erweitern wie z.B. setzen der Farbe und Stiftstärke, zeichnen von geometrischen Figuren wie Kreise und Rechtecke, Strichmännchen, HSHL-Logos etc.  
Viel Spaß!