

LS 2 – Programmierertechnik

Das Erstellen erster kleiner Programme klappt bereits ganz gut. Sie wollen sich noch tiefer in die möglichen Kontroll- und Datenstrukturen von Python einarbeiten. Nebenbei wollen Sie Ihren Programmierstil noch etwas verfeinern.

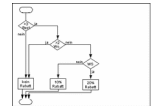


Aufgabe 1 - Programmierhilfsmittel



- Welche Vorteile sollen durch Programmierrichtlinien erzielt werden?
- Zusammen mit Python wurden Python Programmierrichtlinien spezifiziert. Wo sind diese definiert?
- Welche Konventionen zur Namensgebung sind in den Python Programmierrichtlinien festgelegt?
- Welche Konventionen zur Einrückung sind in den Python Programmierrichtlinien festgelegt?
- Was versteht man in Python unter einem DocString?
- Welche Aufgabe besitzen Code-Formatierer? Welche Formater für Python kennen Sie?
- Was versteht man unter einem (Static-)Code-Analyzer? Welche Code-Analyzer für Python kennen Sie? Mit welchem anderen Namen werden Code-Analyzer auch bezeichnet?
- Viele Hilfsmittel stehen in Python in Form von Paketen zur Verfügung. Wie heißt der standardmäßig mit Python ausgelieferte Paket-Manager? Wie lauten die Befehle, um sich die verfügbaren Pakete anzusehen? Wie kann man ein zusätzliches Paket installieren?

Aufgabe 2 - Fingerübung Kontrollstrukturen



Im folgenden finden Sie einige kleine Aufgaben unterschiedlichen Schwierigkeitsgrads zum Thema Schleifen und Fallunterscheidungen.

Bei allen Aufgaben sollen die auszugebenden Zahlen jeweils untereinander stehen.

Lösen Sie die Aufgaben zuerst auf Papier. Kontrollieren Sie Ihre Lösung am Rechner.

Dokumentieren Sie die Aufgaben beispielhaft mit Struktogrammen und/oder Projektablaufplänen.

a)	Mit einer WHILE-Schleife von -100 bis +100 zählen in 10-er Schritten	aufgabe_ls2_01.py
b)	Mit einer FOR-Schleife von 100 rückwärts in 10-er Schritten bis -100 zählen	
c)	Mit einer FOR-Schleife von 0 bis 10 zählen und bei allen Zahlen, die größer als 5 sind neben der Zahl zusätzlich „Hurra“ ausgeben.	
d)	Mit einer WHILE-Schleife rückwärts von 10 bis -10 zählen; bei allen Zahlen, die größer gleich als 5 sind neben der Zahl zusätzlich „gross“ ausgeben; bei allen Zahlen, die kleiner gleich -5 sind neben der Zahl zusätzlich „klein“ ausgeben;	

e)	Mit einer WHILE-Schleife von 0 bis 10 zählen; bei 3 soll neben der Zahl „drei“ stehen, bei 4 soll neben der Zahl „vier“ stehen, bei 5 soll neben der Zahl „fünf“ stehen, ansonsten soll nur die Zahl ausgegeben werden.	
f)	Mit einer FOR-Schleife rückwärts in 10-er Schritten von 100 bis 0 zählen. Wenn die Zahl durch 30 teilbar ist, soll neben der Zahl „Bingo“ ausgegeben werden.	aufgabe_ls2_02.py
g)	Mit einer WHILE-Schleife von 0 bis 100 zählen. Wenn die Zahl durch 5 teilbar ist, soll neben der Zahl „Bingo“ ausgegeben werden. Wenn die Zahl durch 8 teilbar ist, soll neben der Zahl „Ringo“ ausgegeben werden. Wenn die Zahl durch 5 und 8 teilbar ist, soll neben der Zahl „Bingo Ringo“ ausgegeben werden.	
h)	Es soll eine ganze Zahl zwischen 0 und 5 eingelesen werden. Anschließend soll der Name der Zahl ausgegeben werden, z. B. Eingabe: „3“ Ausgabe „drei“.	aufgabe_ls2_03.py
i)	Es soll eine ganze Zahl eingelesen werden. Anschließend soll ausgegeben werden, ob die Zahl positiv oder negativ ist. Zusätzlich soll ausgegeben werden, ob die Zahl gerade oder ungerade ist, wie ihr Vorzeichen und wie ihr Betrag ist.	

Aufgabe 3 - Reflexion Kontrollstrukturen



- Welche drei Kontrollstrukturen kennt die strukturierte Programmierung?
- Was ist der Unterschied zwischen einer kopfgesteuerten und einer fußgesteuerten Schleife? Wie werden sie in Struktogrammen dargestellt?
- Wie oft wird eine kopfgesteuerte Schleife mindestens durchlaufen?
- Sie wollen die Schleifenvariable `i` in jedem Durchlauf um 10 erhöhen. Wie sieht die entsprechende Python Anweisung aus?
- Wenn der Inhalt der Variablen `i` größer als 50 ist, soll „groß“ ausgegeben werden, ansonsten soll nichts gemacht werden. Wie sieht der entsprechende Python Code aus?
- Bei einer Fallunterscheidung müssen Sie mehrere Bedingungen verknüpfen. Welche Möglichkeiten (Schlüsselwörter) bietet Python? Geben Sie jeweils ein kleines Beispiel an.
- Was ist der Unterschied zwischen `=` und `==`?
- Zum Umgang mit Schleifen gibt es in Python die Schlüsselwörter `continue` und `break`. Was bewirken diese Anweisungen?

Aufgabe 4 - Fingerübung Listen



a)	Speichere die Elemente „Schwalbe“, „Kokosnuss“, 13, „Spam“ und 3.14 in einer Liste mit dem Namen <code>liste</code> ab und versuche herauszufinden, was die folgenden Methoden machen. Probieren Sie es anschließend aus und	aufgabe_ls2_04.py
----	--	--------------------------

	<p>geben Sie jeweils die Ergebnisse aus.</p> <pre>>>> liste[2] = 666 >>> len(liste) >>> liste.append("Ni") >>> liste.extend([4, 5, 3.14]) >>> liste.insert(2, "Tauben") >>> liste.count(3.14) >>> liste.index(3.14) >>> liste.remove(3.14) >>> liste.pop() >>> liste.reverse()</pre>	
b)	Legen Sie eine Liste mit den Elementen 1, 2, 3, 9, 8, 7, 6, 4, 5, 0, 10 an. Anschließend durchlaufen Sie die Liste mit einer FOR-Schleife und geben alle durch 5 teilbaren Zahlen aus.	aufgabe_ls2_05.py
c)	Erstellen Sie eine leere Liste. Schreiben Sie mit einer WHILE-Schleife die Zahlen von 0 bis 10 (jeweils einschließlich) in eine Liste. Durchlaufen Sie anschließend die Liste mit einer WHILE-Schleife und verdoppeln Sie alle Werte in der Liste. Geben Sie die Liste jeweils aus.	aufgabe_ls2_06.py

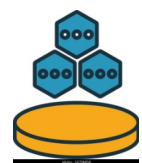
Aufgabe 5 - Fingerübung Dictionary



a)	Erstellen Sie ein Dictionary mit den Schlüsseln Vorname, Nachname, Strasse, Hausnummer, PLZ, Ort. Benutzen Sie Werte Ihrer Wahl.	aufgabe_ls2_07.py
b)	Durchlaufen Sie das Dictionary mit einer FOR-Schleife und geben Sie es zeilenweise aus.	
c)	Geben Sie alle Schlüssel aus.	
d)	Ersetzen Sie die Strasse und Hausnummer durch neue Werte.	
e)	Geben Sie die Anzahl der Schlüssel aus.	
f)	Ergänzen Sie einen Schlüssel Land mit einem Wert Ihrer Wahl.	

Aufgabe 6 - Reflexion Datenstrukturen

- Python kennt standardmäßig 4 Datenstrukturen. Welche?
- Geben Sie ein Beispiel für eine Liste an, welche aus 5 Vornamen besteht.
- Mit welchem Index wird das erste Listenelement indiziert?
- Mit welcher Python Funktion kann man bestimmen, wie lang eine Liste ist?



- e) Wodurch unterscheidet sich ein Tuple von einer Liste? Wie würde das Beispiel aus (b) als Tuple aussehen?
- f) Python bietet noch weitere Datenstrukturen: Tuple und Set. Erarbeiten Sie jeweils ein Beispiel, wofür diese Datenstrukturen gut sind und wie man sie nutzen kann.

Aufgabe 7 - Gemischte Aufgaben

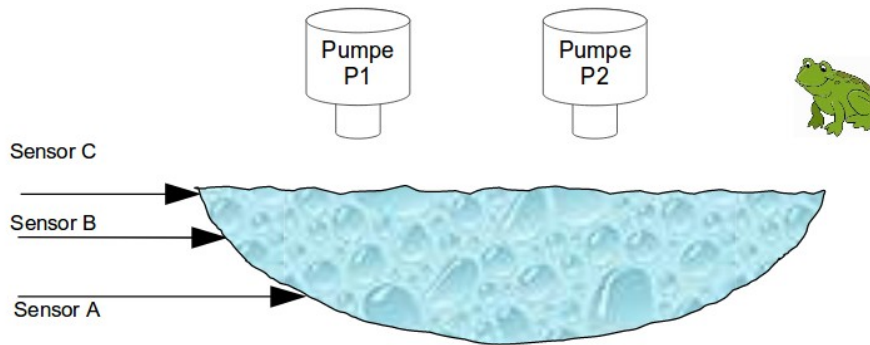


- a) Zur Größenangabe gibt es neben Byte verschiedene Einheiten: KB, MB, GB, TB, KiB, MiB, GiB, TiB. Schreiben Sie ein Python-Programm, welches eine beliebige Größe einliest und diese in alle anderen umrechnet. Speichern Sie das Programm unter dem Namen **aufgabe_ls2_10.py** ab.
- b) Erstellen Sie ein Programm zur Messwertauswertung. Zuerst gibt der Benutzer den Namen einer Datei an. Danach liest das Programm die Messwerte (Gleitkommazahlen) aus der Datei ein. Nach dem Einlesen der Messwerte gibt das Programm folgende Daten aus:
 - Anzahl der Messwerte
 - Summe der Messwerte
 - Mittelwert der Messwerte
 - Maximum der Messwerte
 - Minimum der MesswerteSpeichern Sie das Programm unter dem Namen **aufgabe_ls2_11.py** ab.
- c) Schreiben Sie ein weiteres Programm, welches eine Datei mit Zufallszahlen als Messwertdatei erzeugt. Speichere das Programm unter dem Namen **aufgabe_ls2_12.py** ab.
- d) Schreiben Sie ein weiteres Programm, welches prüft, ob ein Jahr ein Schaltjahr ist. Speichere das Programm unter dem Namen **aufgabe_ls2_13.py** ab.

Aufgabe 8 - „Teichsteuerung“



- Als Amphibienfreunde wollen wir das Überleben der Frösche im Gartenteich sichern und entwickeln eine Steuerung zur Befüllung eines Froschteiches mit zwei Pumpen. Die Sensoren liefern 1 Signal wenn das Wasser sie erreicht bzw. 0 Signal, wenn kein Wasser am Sensor ist. Die Eingänge sollen jede Minute abgefragt werden (zum Testen mit einem kürzeren Rhythmus arbeiten).

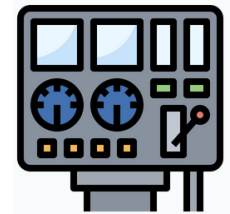


- Befindet sich der Wasserstand unterhalb des Sensors A, dann müssen beide Pumpen laufen.
- Befindet sich der Wasserstand zwischen den Sensoren A und B darf nur Pumpe 1 laufen.
- Befindet sich der Wasserstand zwischen den Sensoren B und C darf nur Pumpe 2 laufen.
- Erreicht der Wasserstand den Sensor C oder höher darf keine der Pumpen in Betrieb sein.
- Wenn die Sensoren einen Zustand melden, der nicht möglich ist ("schwebendes Wasser"), gehen beide Pumpen aus und ein Warnsignal ertönt.
- Eingänge: C, B, A
- Ausgänge: P1, P2, W
- Speichere das Programm unter dem Namen **aufgabe_ls2_14.py** ab.

Aufgabe 9 – „Schaltwarte“

Schreiben Sie ein Python-Programm zur Lösung folgender Aufgabenstellung:

In einer Schaltwarte kommen vier Störungsmeldungen a, b, c, d an, deren Dringlichkeit (Priorität) von a nach d abnimmt. Die Störungen sollen mit vier Lampen A, B, C, D so signalisiert werden, dass jeweils nur die beiden Störungsmeldungen mit der höchsten Priorität angezeigt werden. Die Eingänge sollen jede Sekunde abgefragt werden.



Hinweise:

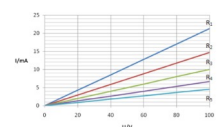
- Die ankommenden Störmeldungen sollen durch entsprechende Eingaben simuliert werden.
- Die vier Lampen sollen durch entsprechende Ausgaben simuliert werden.
- Speichere das Programm unter dem Namen **aufgabe_ls2_15.py** ab.

Aufgabe 10 – „I-U-Kennlinien von linearen Widerständen“

Die I-U-Kennlinien von linearen Widerständen lassen sich als lineare Funktionen darstellen.

Erstellen Sie ein Python-Programm, welches folgende Funktionalität liefert:

1. Der Benutzer soll folgende Daten eingeben:
 - Widerstandswert R in Ohm

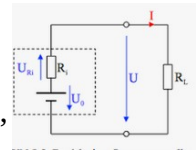


- Max. Spannung U_{\max} in Volt, die dargestellt werden soll
2. Basierend auf den obigen Eingabedaten wird automatisch eine Wertetabelle erstellt, die aus den folgenden Spalten besteht:
- Spannung U am Widerstand
 - Stromstärke I
 - Leistung am Widerstand P

Die Werte für die Spannung U sollen in 11 Schritten von 0 bis U_{\max} laufen. Die anderen Werte werden jeweils mit Hilfe geeigneter Formeln berechnet.

Speichere das Programm unter dem Namen **aufgabe_ls2_16.py** ab.

Aufgabe 11 - „Innenwiderstand“



Beim Thema „Innenwiderstand einer realen Spannungsquelle“ geht man davon aus, dass eine reale Spannungsquelle aus einer idealen Spannungsquelle und einem in Reihe geschalteten Innenwiderstand besteht. Mit dieser realen Spannungsquelle wird ein variabler Lastwiderstand betrieben.

Erstellen Sie ein Python-Programm, welches folgende Funktionalität liefert:

1. Der Benutzer soll folgende Daten eingeben können:
 - Spannung U_0 der idealen Spannungsquelle in Volt
 - Innenwiderstand R_i der realen Spannungsquelle in Ohm
 - Max. Widerstandswert des Lastwiderstands $R_{L\max}$ in Ohm
2. Basierend auf den obigen Eingabedaten wird automatisch eine Wertetabelle gefüllt, die aus den folgenden Spalten besteht:
 - Lastwiderstand R_L
 - Leistung am Lastwiderstand P_L
 - Gesamtwiderstand R_{ges}
 - Gesamtleistung P_{ges}
 - Stromstärke I
 - Wirkungsgrad μ

Die Werte für den Lastwiderstand R_L sollen in 21 Schritten von 0 bis $R_{L\max}$ laufen. Die anderen Werte werden jeweils mit Hilfe geeigneter Formeln berechnet.

Speichere das Programm unter dem Namen **aufgabe_ls2_17.py** ab.