

# LS 2 – Programmiertechnik

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.



## Teil A: Programmierhilfsmittel

### Aufgabe 1 - Überwachungsprogramm aufarbeiten

Als Sie das Überwachungsprogramm (`aufgabe_ls1_12.py`) in LS1 erstellt haben, haben Sie sich noch keine Gedanken über Programmierrichtlinien und Code-Formatierung gemacht. Arbeiten Sie das Programm auf:



1. Das Überwachungsprogramm soll den BK-GuT-Programmierrichtlinien entsprechen.
  2. Das Überwachungsprogramm soll mit einem der Formatierungswerkzeuge `black` oder `autopep8` bearbeitet worden sein.
  3. Das Programm soll mit den Code-Analysen `pylint` und `flake8` analysiert werden.
- Führen Sie die vorgeschlagenen Verbesserungen durch.

Speichern Sie das Programm unter dem Namen **`aufgabe_ls2_01.py`** ab.

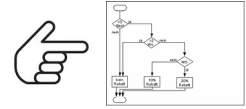
### Aufgabe 2 - Reflexion Programmierhilfsmittel

- a) Welche Vorteile sollen durch Programmierrichtlinien erzielt werden?
- b) Zusammen mit Python wurden Python Programmierrichtlinien spezifiziert. Wo sind diese definiert?
- c) Welche Konventionen zur Namensgebung sind in den Python Programmierrichtlinien festgelegt?
- d) Welche Konventionen zur Einrückung sind in den Python Programmierrichtlinien festgelegt?
- e) Was versteht man in Python unter einem DocString?
- f) Welche Aufgabe besitzen Code-Formatierer? Welche Formater für Python kennen Sie?
- g) Was versteht man unter einem (Static-)Code-Analyzer? Welche Code-Analyzer für Python kennen Sie? Mit welchem anderen Namen werden Code-Analyzer auch bezeichnet?
- h) 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?



# Teil B: Kontrollstrukturen

## Aufgabe 3 - 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. Speichern Sie das Programm unter dem Namen **aufgabe\_ls2\_02.py** ab.

- Mit einer WHILE-Schleife von -100 bis +100 zählen in 10-er Schritten
- Mit einer WHILE-Schleife rückwärts von 10 bis -10 zählen; bei allen Zahlen, die größer gleich 5 sind neben der Zahl zusätzlich „groß“ ausgeben; bei allen Zahlen, die kleiner gleich -5 sind neben der Zahl zusätzlich „klein“ ausgeben;
- 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.
- 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.
- 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“.
- 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 4 - 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?

# Teil C: Datenstrukturen



## Aufgabe 5 - Fingerübung Listen

a)	<p>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 geben Sie jeweils die Ergebnisse aus.</p> <pre>&gt;&gt;&gt; liste[2] = 666 &gt;&gt;&gt; len(liste) &gt;&gt;&gt; liste.append("Ni") &gt;&gt;&gt; liste.extend([4, 5, 3.14]) &gt;&gt;&gt; liste.insert(2, "Taubе") &gt;&gt;&gt; liste.count(3.14) &gt;&gt;&gt; liste.index(3.14) &gt;&gt;&gt; liste.remove(3.14) &gt;&gt;&gt; liste.pop() &gt;&gt;&gt; liste.reverse()</pre>	<b>aufgabe_ls2_04.py</b>
b)	<p>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.</p>	<b>aufgabe_ls2_05.py</b>
c)	<p>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.</p>	<b>aufgabe_ls2_06.py</b>
d)	<p>Auf der W3Schools-Web-Site ist der Umgang mit Listen gut erklärt (<a href="https://www.w3schools.com/python/python_lists.asp">https://www.w3schools.com/python/python_lists.asp</a>). Lesen Sie sich den entsprechenden Abschnitt durch und experimentieren Sie mit den dort genutzten Beispielen.</p>	

## Aufgabe 6 - Dictionary für einen Drucker



- Erstellen Sie ein Dictionary mit den Schlüsseln ID, Hersteller, Modell, Beschaffungsjahr, Farbe und Standort. Benutzen Sie Werte Ihrer Wahl, wobei Farbe ein boolescher Wert sein soll.
- Durchlaufen Sie das Dictionary mit einer FOR-Schleife und geben Sie es zeilenweise aus.
- Geben Sie alle Schlüssel aus.
- Geben Sie die Anzahl der Schlüssel aus.
- Ergänzen Sie weitere Schlüssel mit Werten Ihrer Wahl und geben Sie das Dictionary noch

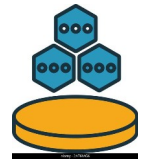
einmal vollständig aus.

Speichere das Programm unter dem Namen **aufgabe\_ls2\_07.py** ab.

### Aufgabe 7 - Vertiefung Dictionaries

Auf der W3Schools-Web-Site ist der Umgang mit Dictionaries gut erklärt ([https://www.w3schools.com/python/python\\_dictionaries.asp](https://www.w3schools.com/python/python_dictionaries.asp)). Lesen Sie sich den entsprechenden Abschnitt durch und experimentieren Sie mit den dort genutzten Beispielen.

### Aufgabe 8 - 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?
- Wodurch unterscheidet sich ein Tuple von einer Liste? Wie würde das Beispiel aus (b) als Tuple aussehen?
- 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 9 - Druckerverwaltung



Schreiben Sie ein Programm zur Druckerverwaltung. Pro Drucker sollen die folgenden Informationen abgespeichert werden: `ID`, `Hersteller` und `Modell`. Das Programm soll folgende Funktionalität bereitstellen:

1. Neue Drucker anlegen
2. Bei vorhandenen Druckern Daten ändern
3. Drucker zu löschen
4. Alle Drucker anzuzeigen

Intern soll das Programm die Informationen als eine Liste von Dictionaries darstellen. Wenn das Programm beendet wird, sollen die Informationen in der CSV-Datei `druckerverwaltung.csv` gespeichert werden, bei Programmstart soll die CSV-Datei wieder eingelesen werden.

Zur Vereinfachung gehen Sie davon aus, dass die Datei `druckerverwaltung.csv` immer vorhanden ist und die `Id` eindeutig ist.

Speichere das Programm unter dem Namen **aufgabe\_ls2\_08.py** ab.

#### Erweiterungsmöglichkeiten:

- Erfassen Sie noch zusätzliche Informationen für die Drucker.
- Sorgen Sie dafür, dass die `ID` eindeutig ist.
- Dokumentieren Sie das Überwachungsprogramm mit geeigneten Diagrammen.
- Geben Sie die Druckerinformationen als PDF-Datei aus.

# Teil D: Zusatzaufgaben



## Aufgabe 10 - Gemischte Aufgaben

- a) Zur Größenangabe gibt es neben Byte verschiedene Einheiten: KB, MB, GB, TB, KiB, MiB, GiB, TiB. Schreiben Sie ein Python-Programm **Größen-Rechner**, 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 einen **Netzwerk-Rechner**. Nach dem Einlesen einer Host-IP und eines CIDR-Suffixes gibt das Programm folgende Daten aus:

- Netzadresse
- CIDR-Suffix
- Netzmaske
- Hostmaske
- Anzahl der adressierbaren Hosts
- Broadcastadresse
- Kleinste und größte Host-IP-Adresse



Als Hilfsmittel können Sie das Modul `ipaddress` nutzen. Speichern Sie das Programm unter dem Namen **aufgabe\_ls2\_11.py** ab.