Python – Listen

Hintergrund

Bisher hatten wir Daten (z.B. Zahlen, Texte, Booleans, …) einzeln in Variablen gespeichert.

Python bietet jedoch auch die Möglichkeit, Ansammlungen von Daten in Listen zu speichern.

Beispiel: Erstellung einer Liste

```
schuelernamen = ["Guiseppe", "Liam", "Okan", "Mulue"]
```

Hier haben wir nun eine Liste erstellt. Diese hat den Namen „schuelernamen“ und enthält nicht nur einen Wert (egal ob Zahl, Text, Boolean, …), wie wir das bisher von Variablen gewohnt waren, sondern mehrere Werte – die schuelernamen-Liste oben enthält bspw. mehrere Texte (Strings), und zwar die Namen von einigen Schülern.

Mit Listen können wir nun einige Dinge anstellen, die ohne die Verwendung von Listen viel umständlicher wären. Ein paar Beispiele:

Beispiel: Ausgabe einer Liste als Ganzes

```
print(schuelernamen)
```

Wir können den Inhalt einer Liste als Ganzes mit dem print-Befehl ausgeben. In unserem Beispiel würde das zu folgender Ausgabe in der Konsole führen:

```
['Guiseppe', 'Liam', 'Okan', 'Mulue']
```

Beispiel: Ausgabe der Liste „Element für Element“ bzw. „Wert für Wert“

```
for name in schuelernamen:
    print(name)
```

Wir können den Inhalt einer Liste auch Element für Element bzw. Wert für Wert ausgeben, indem wir eine For-Schleife verwenden. Die For-Schleife geht nun jedes Element unserer Liste („schuelernamen“) einzeln durch. Im obigen Beispiel nennen wir das Element des aktuellen Schleifendurchlaufs „name“. Mit „print(name)“ schreiben wir dementsprechend nun Name für Name in die Konsole. Dort erhalten wir also folgende Ausgabe:

```
Guiseppe
Liam
Okan
Mulue
```

Beispiel: Hinzufügen von weiteren Elementen zur Liste

```
schuelernamen.append("Kebba")
schuelernamen.append("Ronald")
```

Zu einer bereits existierenden Liste können wir zudem neue Elemente hinzufügen. Mit der „append“-Methode können wir bspw. neue Schülernamen zu unserer „schuelernamen“-Liste hinzufügen.

```
schuelernamen.extend(["Marco", "Omar"])
```

Mit der „extend“-Methode können wir bspw. gleich ganze Listen hinten an unsere „schuelernamen“-Liste anhängen, also mehrere Schülernamen gleichzeitig!

Wenn wir nach den obigen Beispielen nochmal unsere Liste ausgeben, sehen wir, dass die neuen Schülernamen erfolgreich zu unserer „schuelernamen“-Liste hinzugefügt wurden:

```
for name in schuelernamen:
    print(name)
```

Dies führt nun zu folgender Ausgabe auf der Konsole:

```
Guiseppe
Liam
Okan
Mulue
Kebba
Ronald
Marco
Omar
```

Beispiel: Hinzufügen eines weiteren Elementes an einer bestimmten Stelle der Liste

In dem vorigen Beispiel haben wir neue Elemente nur hinten an die Liste angehängt. Mit der „insert“-Methode können wir Elemente allerdings auch an einer beliebigen Stelle der Liste hinzufügen.

```
schuelernamen.insert(1, "Jonas")
```

In dem obigen Beispiel fügen wir den Namen „Jonas“ in unsere „schuelernamen“-Liste ein – diesmal allerdings nicht ans Ende, sondern, wie man am ersten Parameter sieht, an der Position „1“.

Die Position „1“ ist die zweite Position in der Liste, weil man in Programmiersprachen immer bei „0“ anfängt zu zählen – Position 0 ist also die erste Position in der Liste, Position 1 ist die zweite Position in der Liste, Position 2 ist die dritte Position in der Liste usw.

Wenn wir uns nun die Liste erneut ausgeben lassen, sollte „Jonas“ also nun an zweiter Stelle in die Liste eingefügt worden sein. Schauen wir nach:

```
print(schuelernamen)
```

Dies erzeugt auf der Konsole nun folgende Ausgabe:

```
['Guiseppe', 'Jonas', 'Liam', 'Okan', 'Mulue', 'Kebba', 'Ronald', 'Marco', 'Omar']
```

Wie wir sehen hat das Einfügen von „Jonas“ an der zweiten Stelle der Liste funktioniert.

Weitere Listenfunktionen (auch „Listenmethoden“ genannt)

Diese sind unter „Arbeitsmaterialien“ zu finden: 01P-5.2.1_ListenUebersichtFunktionen.pdf

Arbeitsaufträge

Aufgabe 1

Verwenden Sie jede der in dem obigen Dokument vorgestellten Listenfunktionen in mindestens einem selbst gewählten Beispiel (verwenden Sie bspw. die „sort“-Methode, um die „schuelernamen“-Liste oder eine andere Liste alphabetisch zu sortieren, die „delete“-Methode, um bestimmte Elemente aus einer Liste zu löschen usw.).

Kommentieren Sie Ihre Beispiele jeweils kurz im Quellcode!

Aufgabe 2

Schreiben Sie ein Programm, welches mit einer leeren Liste namens „zahlenliste“ beginnt. Das Programm soll den Benutzer nun solange zu Eingaben auffordern, bis der Benutzer „Ende“ eingibt.

Wenn der Benutzer „Einfügen“ eingibt, so soll Ihr Programm den Benutzer nach einer Zahl fragen und diese ans Ende der Liste einfügen.

Wenn der Benutzer „Ausgeben“ eingibt, so soll Ihr Programm die gesamte Liste ausgeben (egal ob als Ganzes oder Element für Element).

Wenn der Benutzer „Länge“ eingibt, so soll Ihr Programm die Länge Ihrer Liste ausgeben (also wie viele Zahlen Ihre Liste enthält).

Wenn der Benutzer „LöscheZahlAnPosition“ eingibt, so soll Ihr Programm den Benutzer nach einer Position fragen und in der zahlenliste die Zahl an der entsprechenden Position löschen. Falls es diese Position in Ihrer Liste gar nicht gibt, soll Ihr Programm eine Fehlermeldung ausgeben.

Wenn der Benutzer „LöscheAlleZahlen“ eingibt, so soll Ihr Programm alle Zahlen in der Liste löschen. Die Liste ist danach also wieder leer, wie zu Beginn.

Wenn der Benutzer „Ende“ eingibt, soll sich Ihr Programm, wie oben erwähnt, beenden.