Woche 03: Programmierung – Erstellen von Listen in Python

Skript

Erarbeitet von
Ludmila Himmelspach

Lernziele

- Eine Liste in Python erstellen können
- Neue Elemente einer Liste hinzufügen können

Inhalt

Einstieg

In Python werden Listen oft dafür benutzt, um komplexe Daten zu strukturieren und übersichtlich abzulegen. Dafür steht eine Reihe von Methoden zur Verfügung, mit deren Hilfe Elemente einer Liste angefügt, geändert oder gelöscht werden können. In diesem

Projekt „KI für alle"
HeiCAD – Heine Center for Artificial Intelligence and Data Science

Video lernst du, wie du in Python Listen erstellen und diesen neue Elemente hinzufügen kannst.

Erzeugung von Listen

In Python werden Listen durch eckige Klammern gekennzeichnet. Die Elemente innerhalb einer Liste werden durch Kommas voneinander getrennt. Es gibt mehrere Möglichkeiten eine Liste zu erzeugen. Wenn die Einträge bzw. die Elemente einer Liste bereits bekannt sind, können diese direkt beim Erzeugen angegeben werden.

Quelle [1]

Unsere erste Liste soll *kinderbuecher* heißen, in der wir Titel einiger Kinderbücher speichern wollen. Zuerst geben wir den Listennamen an, also *kinderbuecher*. Danach kommt das Gleichheitszeichen als Zuweisungsoperator. Nun geben wir mit den eckigen Klammern an, dass es sich um eine Liste handelt.

Jetzt können wir innerhalb der Klammer unsere Elemente angeben, die wir voneinander durch Kommas trennen. Bei Buchtiteln handelt es sich um Texte, die wir als Zeichenketten speichern, also müssen wir diese in Anführungszeichen angeben. Also schreiben wir „Peter Pan“, „Pippi Langstrumpf“ und „Alice im Wunderland“.

```
kinderbuecher = ["Peter Pan", "Pippi Langstrumpf", "Alice im Wunderland"]
```

Wenn du wissen möchtest, ob du einen Fehler bei der Erzeugung der Liste gemacht hast, kannst du einfach deinen Code ausführen. Das machen wir jetzt. Es gibt keine syntaktischen Fehler, also wurde unsere Liste erzeugt.

Wir können die gesamte Liste auch mit dem *print()*-Befehl ausgeben. Dazu schreiben wir den Listennamen in die Klammer der *print()*-Anweisung. Die Listen werden in Python genauso ausgegeben, wie sie definiert wurden, vielleicht bis auf die Art der Anführungszeichen.

```
print(kinderbuecher)
```

```
['Peter Pan', 'Pippi Langstrumpf', 'Alice im Wunderland']
```

Wir haben gerade eine Liste mit drei String-Elementen erzeugt. Wir können aber auch eine weitere Liste mit Zahlen z. B. Erscheinungsjahren der Bücher erstellen. Wir nennen diese Liste *erscheinungsjahr*. Laut Google ist „Peter Pan“ im Jahr 1911, „Pippi Langstrumpf“ 1945 und „Alice im Wunderland“ 1865 zum ersten Mal erschienen.

```
erscheinungsjahr = [1911, 1945, 1865]
```

Wir können auch diese Liste mit der *print()*-Anweisung ausgeben lassen.

```
print(erscheinungsjahr)
```

```
[1911, 1945, 1865]
```

Im Unterschied zu vielen anderen Programmiersprachen kann man in Python in einer Liste Elemente unterschiedlicher Datentypen speichern. Wir können zum Beispiel eine Liste erstellen, in der wir die Basisinformationen zu einem Buch speichern. Wir nennen diese Liste *kinderbuch_1*. In dieser Liste speichern wir den Titel, den Autor und das Erscheinungsjahr eines Buchs. Also geben wir als Elemente der Liste *kinderbuch_1* Folgendes ein: *„Pippi Langstrumpf“* in Anführungszeichen, *„Astrid Lindgren“* in Anführungszeichen und *1945* ohne Anführungszeichen, weil wir das Erscheinungsjahr als eine Zahl speichern.

```
kinderbuch_1 = ["Pippi Langstrumpf", "Astrid Lindgren", 1945]
```

Auch diese Liste können wir mit der *print()*-Anweisung ausgeben lassen.

```
print(kinderbuch_1)
```

```
['Pippi Langstrumpf', 'Astrid Lindgren', 1945]
```

Diese Art von Listen wird oft genutzt, um tabellenartige Strukturen zu erzeugen. Man kann kleine Listen erzeugen, die einzelne Tabellenzeilen darstellen und später diese zu einer verschachtelten Liste zusammenfügen. So erstellen wir eine weitere Liste mit Informationen zu einem Buch und nennen diese *kinderbuch_2*.

```
kinderbuch_2 = ["Alice im Wunderland", "Lewis Carroll", 1865]

print(kinderbuch_2)
```

```
['Alice im Wunderland', 'Lewis Carroll', 1865]
```

Jetzt können wir eine übergeordnete Liste erstellen, die diese beiden Listen als Unterlisten enthält. Da wir schon eine Liste namens *kinderbuecher* haben, nennen wir diese Liste einfach *buchliste*.

```
buchliste = [kinderbuch_1, kinderbuch_2]
```

Wir geben auch diese Liste mit der *print()*-Anweisung aus.

```
print(buchliste)
```

```
[['Pippi Langstrumpf', 'Astrid Lindgren', 1945], ['Alice im Wunderland',
'Lewis Carroll', 1865]]
```

In der Ausgabe sieht man, dass *buchliste* zwei Listen als Elemente enthält.

Hinzufügen neuer Listenelemente

Bis jetzt haben wir Listen mit vorher festgelegten Elementen erstellt. Wenn die Elemente einer Liste am Anfang noch nicht bekannt sind und erst im Laufe des Programms generiert werden, kann zuerst eine leere Liste erstellt werden und dieser nach und nach Elemente hinzugefügt werden.

Wir erstellen eine leere Liste *krimis*, indem wir nach dem Zuweisungsoperator eckige Klammer auf und Klammer zu eingeben.

```
krimis = []
```

Wenn wir diese Liste ausgeben lassen, sehen wir, dass sie keine Elemente enthält.

```
print(krimis)
```

```
[]
```

Um ein Element der Liste *krimis* hinzuzufügen, benutzt man die Funktion *append()* und zwar so, dass man nach dem Listennamen einen Punkt schreibt, danach kommt die Funktion *append()* und in einer runden Klammer das Element, welches man einfügen möchte, also „Mord im Orient-Express“.

```
krimis.append("Mord im Orient-Express")
```

Wenn wir unsere Liste jetzt ausgeben, sehen wir, dass sie ein Element enthält.

```
print(krimis)
```

```
['Mord im Orient-Express']
```

Wir fügen noch den Krimi „Der Pate“ unserer Liste hinzu. Dafür nutzen wir wieder die Funktion *append()*.

```
krimis.append("Der Pate")
```

Wir geben unsere Liste *krimis* mit der *print()*-Anweisung aus und sehen, dass „Der Pate“ am Ende der Liste eingefügt wurde.

```
print(krimis)
```

```
['Mord im Orient-Express', 'Der Pate']
```

Um ein Element an eine bestimmte Stelle in die Liste einzufügen, benutzt man die Funktion *insert().* Wie bei der *append()*-Funktion schreibt man zuerst den Listennamen, danach einen Punkt, dann die Funktion *insert()*. In der runden Klammer der Funktion gibt man zuerst die Position an, an die das neue Element eingefügt werden soll. Wir wollen den Krimi „Alibi“ an

die zweite Position einfügen, also geben wir den Index 1 an. Nach dem Komma geben wir das Element selbst, also „Alibi“ an.

```
krimis.insert(1, "Alibi")
```

Wenn wir jetzt die Liste ausgeben, sehen wir, dass „Alibi“ an der zweiten Stelle in der Liste steht und „Der Pate“ an die dritte Stelle verschoben wurde.

```
print(krimis)
```

```
['Mord im Orient-Express', 'Alibi', 'Der Pate']
```

Take-Home Message

In diesem Video hast du gelernt, wie man in Python einfache und verschachtelte Listen erstellen kann. Außerdem hast du erfahren, wie man neue Elemente ans Ende oder an eine beliebige Stelle in der Liste hinzufügen kann.

Quellen

Quelle [1] Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Weiterführendes Material

Schmitt, S. (2021). *Python Kompendium: Professionell Python Programmieren lernen.* BMU Media Verlag

Barry, P. (2017). *Python von Kopf bis Fuß.* O‘Reilly

Disclaimer

Transkript zu dem Video „Woche 03: Programmierung – Erstellen von Listen in Python“, Ludmila Himmelspach.