

Übungsblatt 03

Aufgabe 1

Gegeben ist eine Liste:

```
punkte = [[40,60],[50,60],[60,60],[70,60],[80,60],[90,60],[100,60],  
[110,60],[120,60]]
```

Die Unterlisten enthalten zwei Zahlen, welche die Koordinaten von Punkten darstellen. Es soll ein Programm erstellt werden, das entscheidet, ob diese Punkte innerhalb eines Kreises liegen. Der Kreis hat die Mittelpunktkoordinate x_k und y_k . Der Durchmesser des Kreises wird in dk gespeichert. Schreiben Sie das Programm in allgemeingültiger Form, sodass beliebige Kreise betrachtet werden können.

Beispielausgabe:

Geben Sie x_k an: 80

Geben Sie y_k an: 50

Geben Sie den Kreisdurchmesser dk an: 60

```
[[40, 60, False], [50, 60, False], [60, 60, True], [70, 60, True], [80,  
60, True], [90, 60, True], [100, 60, True], [110, 60, False], [120, 60,  
False]]
```

```
In [14]: # TODO Aufgabe 1)  
punkte = [[40,60],[50,60],[60,60],[70,60],[80,60],[90,60],[100,60],[110,60],[120,60]]
```

Aufgabe 2.1

Schreiben Sie ein Programm, dass die Wahrscheinlichkeit dafür berechnet, dass beim Wurf zweier Würfel die Summe der Augenzahlen 7 entspricht. Lösen Sie das Problem mit *list comprehension*. Tipp: nutzen Sie die *built-in function* `sum(iterable)` und zur Ausgabe *f-Strings*

Ausgabe: Die Wahrscheinlichkeit, die Augensumme 7 zu würfeln beträgt:
16.67 %

```
In [15]: # TODO Aufgabe 2.1)
```

Aufgabe 2.2

Erweitern Sie Ihr Programm um eine Funktion, die obige Wahrscheinlichkeit für eine **beliebige** Summe berechnet und ausgibt. Geben Sie anschließend alle Summen mit ihren Wahrscheinlichkeiten aus.

```
In [16]: # TODO Aufgabe 2.2)
```

Aufgabe 3

In einer Buchhandlung findet sich in einem Abrechnungsprogramm in Python eine Liste mit Unterlisten mit folgendem Aufbau:

Bestellnummer	Buchtitel und Autor	Anzahl	Einzelpreis
34587	Learning Python, Mark Lutz	4	40,95 €
98762	Programming Python, Mark Lutz	5	56,80 €
77226	Head First Python, Paul Barry	3	32,95 €

Schreiben Sie ein Programm unter Benutzung von `lambda` und `map`, das als Ergebnis eine Liste mit Zweier-Tupeln liefert. Jedes Tupel besteht aus der Bestellnummer und dem Produkt aus der Anzahl und dem Einzelpreis. Das Produkt soll jedoch um 10,- € erhöht werden, wenn der Bestellwert unter 100,00 € liegt.

Ausgabe: `[('34587', 163.8), ('98762', 284.0), ('77226', 108.85)]`

```
In [17]: bestellungen = [ ["34587", "Learning Python, Mark Lutz", 4, 40.95],  
                        ["98762", "Programming Python, Mark Lutz", 5, 56.80],  
                        ["77226", "Head First Python, Paul Barry", 3, 32.95]]  
  
# TODO Aufgabe 3)
```

Aufgabe 4

Erzeugen Sie für einen vorgegebenen Bereich von 1 bis n alle *pythagoreischen Triple* in der Form `p = (a, b, c)`. Für diese muss nach dem Satz des Pythagoras gelten: $a^2 + b^2 = c^2$. Geben Sie die Zahlen sowie die Anzahl der Zahlenpaare im vorgegebenen Bereich aus. Achten Sie darauf, dass keine Doppelten vorkommen, z.B. `(3, 4, 5)` ist das gleiche wie `(4, 3, 5)`

Beispielausgabe:

Geben Sie eine Obergrenze n ein: 20
`[(3, 4, 5), (5, 12, 23), (6, 8, 10), (8, 15, 17), (9, 12, 15)]`
Zwischen 1 und 20 gibt es 5 pythagoreische Tripel

```
In [18]: # TODO Aufgabe 4a) mittels Iterationen und Listen
```

```
In [19]: # TODO Aufgabe 4b) mittels List Comprehension
```

```
In [20]: # TODO Aufgabe 4c) mittels Lambda-Formen
```

Aufgabe 5

In dieser Aufgabe beschäftigen Sie sich mit dem Datentyp `dictionary`. Zur Übung sollen Sie Daten, die in einer Liste abgelegt sind, in ein Dictionary umwandeln.

Aufgabe 5.1

Die gegebene Liste `actors` besteht aus jeweils fünf Unterlisten mit jeweils fünf Tupeln unterschiedlicher Länge. Jedes dieser Tupel enthält Einträge vom Typ `String`.

Wandeln Sie die Liste in ein Dictionary folgender Form um:

```
{0: {'salutation': 'Mr.',
    'givenName': 'Peter',
    'familyName': 'Dinklage',
    'birthday': '06/11/1969',
    'GoT': ['Tyrion', 'Lannister']}},
1: {'salutation': 'Mrs.',
    'givenName': 'Michelle',
    ... }
...}
```

```
In [21]: actors = [(["salutation", "Mr."), ("givenName", "Peter"),
                    ("familyName", "Dinklage"), ("birthday", "06/11/1969"),
                    ("GoT", "Lannister", "Tyrion")],
                  [(["salutation", "Mrs."), ("givenName", "Michelle"),
                    ("familyName", "Fairley"), ("birthday", "01/17/1964"),
                    ("GoT", "Stark", "Catelyn")],
                  [(["salutation", "Mr."), ("givenName", "Nicolaj"),
                    ("familyName", "Coster-Waldau"), ("birthday", "07/27/1970"),
                    ("GoT", "Lannister", "Jamie")],
                  [(["salutation", "Ms."), ("givenName", "Maisie"),
                    ("familyName", "Williams"), ("birthday", "04/17/1997"),
                    ("GoT", "Stark", "Arya")],
                  [(["salutation", "Mrs."), ("givenName", "Emilia"),
                    ("familyName", "Clarke"), ("birthday", "05/01/1987"),
                    ("GoT", "Targaryen", "Daenerys")]]

# TODO Aufgabe 5.1)
```

Aufgabe 5.2

Erstellen Sie nun durch geschickte Umformung ein neues Dictionary. Bei diesem dient der Familienname des gespielten Charakters als Schlüssel des äußeren Dictionary. Der Wert zu jedem Schlüssel ist wiederum ein Dictionary. In diesem (inneren) Dictionary werden nun die Vornamen der Charaktere als Schlüssel verwendet. Die Werte in den inneren Dictionaries sind dann die Daten des Schauspielers des jeweiligen Charakters. Das resultierende Dictionary hat also die folgende beispielhafte Struktur:

```
{'Lannister': { 'Tyrion': {'salutation': 'Mr.',
                          'givenName': 'Peter',
                          'familyName': 'Dinklage',
                          'birthday': '06/11/1969'},
                'Jamie': {'salutation': 'Mr.',
                          'givenName': 'Nicolaj',
                          'familyName': 'Coster-Waldau',
                          'birthday': '07/27/1970'}}},
'Stark': {      'Catelyn': {'salutation': 'Mrs.',
```

```
...}}
```

```
...}
```

Geben Sie außerdem aus, wie viele Mitglieder die fiktive Familie "Stark" nach Ihrem Dictionary hat:

```
Die Familie Stark hat laut diesem Datensatz 2 Mitglieder
```

```
In [22]: # TODO Aufgabe 5.2)
```