

Prescriptive Analytics, Hausaufgabe 1

HENRY HAUSTEIN

Aufgabe 1: Listen & Sortierungen

(a) mittels List Comprehension

```
1  ### Ergänzen Sie hier Ihren Code, um die gewünschte Ausgabe zu
    erzielen. ###
2  thisList2 = [element[::-1] for element in thisList]
3
4  ### Führen Sie anschließend nachfolgenden Programmcode aus, um
    Ihre Lösung zu überprüfen ###
5  print(thisList2)
```

(b) noch mehr List Comprehension

```
1  ### Ergänzen Sie hier Ihren Code, um die gewünschte Ausgabe zu
    erzielen. ###
2  thisList3 = [element for element in thisList if type(element)
    == str]
3
4  ### Führen Sie anschließend nachfolgenden Programmcode aus, um
    Ihre Lösung zu überprüfen ###
5  print(thisList3)
```

(c) mittels lambda-Funktion

```
1  import random
2  evaluations = [('Otto', 3.779, 1.238, 0.49), ('Level', 3.961,
    5.725, 0.233), ('Leseesel', 3.935, 1.482, 2.41), ('Kajak',
    1.989, 2.66, 0.656), ('Mueller', 2.024, 1.427, 3.013), ('
    Retter', 3.297, 2.36, 3.179), ('Rotor', 1.733, 4.218,
    4.972), ('Effe', 3.311, 3.197, 3.991), ('Meier', 5.956,
    2.554, 4.622)]
3
4  ### Bitte ergänzen Sie hier Ihren Code ###
5  sortiert = sorted(evaluations, key = lambda x: x[1] + x[2] + x
    [3], reverse = True)
6  print(sortiert[0])
```

(d) für sowas gibt auch Pakete... aber die sollen wir nicht nutzen

```
1  thisList3Corrupted = ['Otto', ['Level', 'Leseesel', 'Kajak'],
    ['Mueller', 'Retter', 'Rotor', ['Effe', 'Meier']]
```

```

2
3  ### Bitte ergänzen Sie hier Ihren Code ###
4  def flatten(x):
5      liste = []
6      for element in x:
7          if type(element) is list:
8              liste.extend(flatten(element))
9          else:
10             liste.append(element)
11     return liste
12
13  ### Führen Sie anschließend nachfolgenden Programmcode aus, um
    Ihre Lösung zu überprüfen ###
14  thisList3Repaired = flatten(thisList3Corrupted)
15  print(thisList3Repaired)

```

Aufgabe 2: Einfache Funktionen

- (a) Wir benutzen wieder den Index-Operator `::-1`. Eigentlich schreibt man den ersten Buchstaben einer Funktion klein (lowerCamelCase) und den ersten Buchstaben einer Klasse groß (UpperCamelCase).

```

1  thisList = ['Otto', 'Level', 'Leseesel', [1, 2, 3], 'Kajak', '
    Mueller', 'Retter', 'Rotor', [5, 6, 7], 'Effe', 'Meier']
2
3  ### Bitte ergänzen Sie hier Ihren Programmcode ###
4  def CheckForPalindroms(x):
5      liste = []
6      for idx, value in enumerate(x):
7          if type(value) is str:
8              if value.lower() == value[::-1].lower():
9                  liste.append(idx)
10     return liste
11
12  ### Führen Sie anschließend nachfolgenden Programmcode aus, um
    Ihre Lösung zu überprüfen ###
13  CheckForPalindroms(thisList)

```

- (b) Dass der folgende Quelltext beim Ausführen einen Fehler gibt, ist beabsichtigt und soll auch so sein:

```

1  ### Bitte ergänzen Sie hier Ihren Programmcode ###
2  def splitNumber(x):
3      try:
4          ganzeZahl = int(x)
5          rest = x - ganzeZahl
6          return ganzeZahl, rest
7      except:
8          print("Bitte nur Zahlen eingeben")
9          raise TypeError("Input has wrong type!")
10
11  ### Führen Sie anschließend nachfolgenden Programmcode aus, um
    Ihre Lösung zu überprüfen ###

```

```

12 print(splitNumber(3.56))
13 print(splitNumber("Hallo"))

```

Aufgabe 3: Klassendefinition & weitere Funktionen

(a) Klasse Rechteck

```

1  ### Bitte ergänzen Sie hier Ihren Programmcode ###
2  import json
3
4  class Rechteck:
5      id = None
6      width = None
7      height = None
8      area = None
9
10     def __init__(self, id, w, h):
11         self.id = id
12         self.width = w
13         self.height = h
14         self.area = self.DetermineArea()
15
16     def DetermineArea(self):
17         return self.height * self.width
18
19     def __str__(self):
20         return f"Rechteck {self.id} mit Breite {self.width}, Höhe {self.height}, Fläche {self.area}"
21
22     def Load(filename):
23         liste = []
24         with open(filename) as file:
25             data = json.load(file)
26             for id in data.keys():
27                 r = Rechteck(id, data[id]["width"], data[id]["height"])
28                 liste.append(r)
29         return liste
30
31     ### Führen Sie anschließend nachfolgenden Programmcode aus, um
32         Ihre Lösung zu überprüfen ###
33
34     rectangulars = Load('Rectangulars.json')
35     for rec in rectangulars:
36         print(rec)

```

(b) Cäsar-Chiffre

```

1  verschluesselteNachricht = "Ylmny Buomuozaavy aymwbuzzn!
    Scjjcy :)"
2  schluessel = 20

```

```

3
4  ### Bitte ergänzen Sie hier Ihren Programmcode ###
5  def verschieben(x, anzahl):
6      if (x.isupper()):
7          return chr((ord(x) + anzahl - 65) % 26 + 65)
8      else:
9          return chr((ord(x) + anzahl - 97) % 26 + 97)
10
11 def Decrypt(x, anzahl):
12     result = ""
13     for letter in x:
14         if letter not in [" ", ":", ")", "!", "]:
15             result += verschieben(letter, -anzahl)
16         else:
17             result += letter
18     return result
19
20  ### Führen Sie anschließend nachfolgenden Programmcode aus, um
    Ihre Lösung zu überprüfen ###
21  Decrypt(verschlüsselteNachricht, schluessel)

```