

# Prescriptive Analytics, Seminar 1

HENRY HAUSTEIN

## Aufgabe 1: Podcast-Sendezeit

(a) Lösung mittels f-String (<https://zetcode.com/python/fstring/>):

```
1 sendezeit = 70
2 result = 2 * (70 - 3*3)
3 print(f'Bei einer Podcastlänge von {sendezeit} Minuten ergeben
      sich {result/60:.2f} Stunden Sendezeit pro Woche')
```

(b) mittel .format()

```
1 print("Bei einer Podcastlänge von {} Minuten ergeben sich {}
      Stunden Sendezeit pro Woche".format(sendezeit, round(
      result/60,2)))
```

## Aufgabe 2: Chaos in der Mailingliste

(a) mittels List Comprehension ([https://www.w3schools.com/python/python\\_lists\\_comprehension.asp](https://www.w3schools.com/python/python_lists_comprehension.asp))

```
1 mail = "service@klara_Klarna.com"
2 [name.split(".")[0] for name in mail.split("@")[1].split("_")]
```

(b) mittels zip()

```
1 namen = ["Klara", "Sebastian", "Britta", "Klaus"]
2 mails = ["service@klara_Klarna.com", "
      Sebastian_Duesentrieb@gmail.com", "bwiebritta@yahoo.com",
      "Klaus@peter.de"]
3 result = {}
4 for name, mail in zip(namen, mails):
5     result[name] = mail
6 result
```

## Aufgabe 3: Anpassung der Werbeeinnahmen

(a) mittels Definition einer Funktion

```
1 input_price = 333
2 def preisanpassung(oldPrice):
3     if oldPrice <= 400:
```

```

4     return oldPrice * 1.1
5     if 400 < oldPrice and oldPrice <= 600:
6         return oldPrice * 1.2
7     if 600 < oldPrice:
8         return oldPrice * 1.25
9
10    preisanpassung(333)

```

(b) wieder mittels List Comprehension

```

1    oldPrices = [350, 300, 600, 800]
2    [preisanpassung(oldPrice) for oldPrice in oldPrices]

```

(c) mittels `numpy.dot()` welches unabhängig von der Länge der Inputvektoren ist. Die Musterlösung des Lehrstuhls wird falsch, wenn es nicht mehr 4, sondern 5 verschiedene Werbepakete gibt

```

1    import numpy as np
2
3    sales = [3, 5, 7, 4]
4    oldPrice = [350, 300, 600, 800]
5    newPrice = [385, 330, 720, 1000]
6
7    def umsatz(newprices, oldprices, sales):
8        diff = [new - old for new, old in zip(newprices, oldprices)]
9        return np.dot(diff, sales)
10
11    umsatz(newPrice, oldPrice, sales)

```

(d) mittels `lambda`-Funktion

```

1    umsatz = lambda preis, vkz: sum([preis[i]*vkz[i] for i in
2                                     range(len(sales))])
3
3    umsatz(sales, newPrice) - umsatz(sales, oldPrice)

```

## Aufgabe 4: Expansionswahnsinn

(a) mittels Pandas

```

1    import pandas as pd
2    births = pd.read_csv("births.csv")
3    births["age"] = 2022 - births["Year"]
4    zielgruppe = births.loc[births["age"].between(25, 35)]
5    zielgruppe["Count"].sum()

```

(b) mit Pandas geht das besser als mit einem Dictionary

```

1    zielgruppe.groupby(["age"]).sum().sort_values("Count",
2                                                    ascending = False)

```