

07_logical_functions_exercise_solution

October 20, 2020

1 Python Einführung - Übungen

1.1 Aufgaben

Beantworte die einzelnen Fragen. Hilfe bietet dir das Handout, wenn Du nicht weiter weißt

**** Rechne 7^4 (7 hoch 4)?****

```
[1]: 7**5
```

```
[1]: 16807
```

Zerlege (split) den String:

```
s = "Hallo wie geht es dir heute?"
```

in eine Liste

```
[3]: s = "Hallo wie geht es dir heute?"
```

```
[4]: s.split()
```

```
[4]: ['Hallo', 'wie', 'geht', 'es', 'dir', 'heute?']
```

Ausgabe: ['Hallo', 'wie', 'geht', 'es', 'dir', 'heute?']

**** Gegeben sind folgende Variablen:****

```
stadt = "Heidelberg"  
einwohner = 155000  
flaeche_in_km2 = 109
```

**** Benutze .format() um den folgenden String auszudrucken: ****

Die Stadt Heidelberg hat 155000 Einwohner auf einer Flaeche von 109 Quadratkilometern.

```
[6]: stadt = "Heidelberg"  
einwohner = 155000  
flaeche_in_km2 = 109
```

```
[8]: "Die Stadt {} hat {} Einwohner auf einer Flaeche von {} Quadratkilometern."  
     ↪format(stadt, einwohner, flaeche_in_km2)
```

```
[8]: 'Die Stadt Heidelberg hat 155000 Einwohner auf einer Flaeche von 109  
     Quadratkilometern.'
```

Ausgabe: Die Stadt Heidelberg hat 155000 Einwohner auf einer Flaeche von 109 Quadratkilometern.

**** Lass dir von dieser mehrdimensionalen Liste das Element 'hello' ausgeben****

```
[10]: lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
```

```
[13]: lst[3][1][2][0]
```

```
[13]: 'hello'
```

Ausgabe: 'hello'

**** Lass dir von diesem mehrdimensionalen Dictionary das Element 'hello' ausgeben ****

```
[15]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':  
     ↪[1,2,3,'hello']}]}}]
```

```
[24]: d["k1"][3]["tricky"][3]["target"][3]
```

```
[24]: 'hello'
```

Ausgabe: 'hello'

**** Was unterscheidet hauptsächlich ein Tuple von einer Liste? ****

**** Mache hierzu ein Markdown Feld in Jupyter, damit du es als Text beantworten kannst ****

- *tuples* sind **immutable**, d.h. können nicht verändert werden
- *lists* sind **mutable**, d.h. können verändert werden

Erstelle eine Funktion mit dem Namen domainGet die die E-Mail-Domain aus dem folgenden String extrahiert:

user@domain.com

Hier "user@domain.com" gibt zurück: domain.com

```
[28]: # hier die funktion definieren  
def domainGet(x):  
     return x.split("@")[-1]
```

```
[29]: domainGet('user@domain.com')
```

```
[29]: 'domain.com'
```

****** Erstelle eine Funktion mit dem Namen `look_for_money`, welche `True` zurück gibt, wenn das Wort ‘money’ in dem input-String enthalten ist. Großschreibung soll berücksichtigt werden, aber wenn ein Zeichen an dem Wort hängt nicht (Bsp: ‘money,’ wird nicht berücksichtigt). ******

```
[32]: # hier die funktion definieren
def look_for_money(x):
    return "money" in x.lower().split()
```

```
[33]: look_for_money('Ich habe soviel Money in meinen Taschen, digga!')
```

```
[33]: True
```

****** Erstelle eine Funktion `count_money`, welche die Anzahl des Wortes `money` in dem input-String zählt. Großschreibung soll berücksichtigt werden, aber wenn ein Zeichen an dem Wort hängt nicht (Bsp: ‘money,’ wird nicht berücksichtigt). ******

```
[42]: def count_money(x):
    return sum([word in "money" for word in x.lower().split()])
```

```
[43]: count_money('Money all I need is money Money, dude!')
```

```
[43]: 2
```

Benutze `lambda` Ausdrücke und die `filter()` Funktion um Wörter aus einer Liste zu filtern, welche nicht mit dem Buchstaben ‘s’ beginnen. Beispiel:

```
seq = ['schatz', 'freund', 'sind', 'immer', 'wichtig']
```

soll zu folgendem gefiltert werden:

```
['schatz', 'sind']
```

```
[51]: seq = ['schatz', 'freund', 'sind', 'immer', 'wichtig']
```

```
[54]: list(filter(lambda x: x[0]!="s", seq))
```

```
[54]: ['schatz', 'sind']
```

1.1.1 Letzte Aufgabe!

Du fährst ein wenig zu schnell, und ein Polizist hält dich auf. Schreibe eine Funktion, um eines von 3 möglichen Ergebnissen zurückzugeben: “Kein Ticket”, “Kleines Ticket” oder “Grosses Ticket”. Wenn deine Geschwindigkeit 100 oder weniger beträgt, ist das Ergebnis “Kein Ticket”. Wenn die Geschwindigkeit zwischen 101 und einschließlich 120 liegt, ist das Ergebnis “Kleines Ticket”. Wenn die Geschwindigkeit 121 oder mehr beträgt, ist das Ergebnis “Grosses Ticket”. Es sei denn, es ist dein Geburtstag (kodierte als boolescher Wert in den Parametern der Funktion) – an deinem Geburtstag kann deine Geschwindigkeit in allen Fällen um 10 höher sein.

```
[56]: def caught_speeding(speed, is_birthday):  
      if is_birthday:  
          speed -= 10  
      if speed <= 100:  
          return "Kein Ticket"  
      elif speed <= 120:  
          return "Kleines Ticket"  
      else:  
          return "Grosses Ticket"
```

```
[57]: caught_speeding(101, True)
```

```
[57]: 'Kein Ticket'
```

```
[58]: caught_speeding(101, False)
```

```
[58]: 'Kleines Ticket'
```

2 Stay tuned!