

05_logical_functions_exercise

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]:

[1]: 2401

Zerlege (split) den String:

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

in eine Liste

[]:

[]:

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.

[5]:

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

[]:

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

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

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

```
[ ]:
```

Ausgabe: 'hello'

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

```
[16]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'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 ****

```
[9]: # wandel diese Code-Zelle in eine Markdown-Zelle um und beantworte die Frage.
     ↪ Lösche diesen Kommentar dann!
```

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

```
[24]: # hier die funktion definieren
```

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

```
[26]: '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). ****

```
[27]: # hier die funktion definieren
```

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

```
[28]: 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). ****

[30]:

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

[31]: 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']
```

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

[11]:

```
[11]: ['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.

```
[12]: def caught_speeding(speed, is_birthday):  
      # hier dein Code!
```

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

[13]: 'Kein Ticket'

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

[14]: 'Kleines Ticket'

2 Stay tuned!