Tutorium Python

October 23, 2021

Basics

1. Schau dir die wichtigsten Datentypen an und mach dich mit ihnen vertrauter (e.g. str, int, float, list, dict, tuple, etc).
   (a) Wie kann ich herausfinden was für ein Datentyp meine Variable hat?
   (b) Was passiert wenn ich zwei **int** dividiere, welche keine ganze Zahl ergeben?
   (c) Was ist *casten* in Python? Wie funktioniert das für die unterschiedlichen Datentypen?
2. Was genau mach ich mit **import**?
   (a) Erstelle ein Python-File mit Konstanten und greif auf diese Konstanten über ein anderes Python-File zu.
3. Write a python script that takes 3 numbers as command line arguments, adds them up and prints the result. Access to the command line arguments is provided by the module sys, where sys.argv[n] represents the argument at the n position.
4. Write a Python script that takes a string as a command-line argument and reverses the order of the letters.
5. Assume that these two lists are 3D coordinates:

Algorithm 1: Vektoren

```
point_1 = [2.8, -4.7, 0.4]
point_2 = [-8.1, 3.0, -10.6]
```

Write a script that computes the distance between those points (by looping over the individual components)

Mittel

6. Was sind Funktionen?

   (a) Schreib nun Funktionen, welche das Skalarprodukt und das Kreuzprodukt für die Vektoren von oben ausrechnet!

7. Was ist der unterschied zwischen **numpy.array()** und einer normalen Liste?

   (a) Schau dir folgenden Code an und versuche zu verstehen was er macht. Verwende **time.time()** um die Zeiten zu messen (siehe Kommentare im Code) und erkläre warum die Unterschiede so hoch sind!

Algorithm 2: Was passiert hier?

```
N = 10**8
# Method 1
# Time from here
daten=[]
for i in range(0,N) :
    daten.append(i**0.5)
# to here
# Method 2
# and from here
daten = np.sqrt(np.arange(0,N))
# to here
```

8. Plotten ist in der Astronomie *extrem* wichtig (vor allem für zukünftige Protokolle die ihr schreiben müsst :))!

   (a) Deshalb: Verpacke die zwei unterschiedlichen Methoden (siehe Code in Nr. 7) nun in Funktionen, so dass du das N variieren kannst.

   (b) Plotte nun (mit matplotlib.pyplot) das N gegen die gebrauchte Zeit zum Ausrechnen der Wurzel für die beiden Methoden (Liste vs. Array)!

9. File handling: Wie öffne ich ein .txt File?

   (a) Erkläre den Unterschied:

   Algorithm 3: Was passiert hier?

```
#Methode 1
file = open("file.txt")

for line in file:
    print(line)

#Methode 2
with open("file.txt") as file:
    for line in file:
        print(line)
```

   (b) Schreibe nun in ein File "Hello World"!

10. Wie funktioniert *slicing* in Python? Was ist das?

   (a) Erstelle eine liste mit 10 Einträgen deiner Wahl: Gib das 5. bis 8. Element aus. Gib jedes Zweite Element aus. Drehe die Liste um!

   (b) Nehmen wir an wir wissen nicht wie lang eine Liste ist: Wie kann ich trotzdem auf das letzte Element zugreifen?

Und schon kannst du gut programmieren

1. Write a python function that takes a one-dimensional integer array of arbitrary length and a float that represents a percentage (it should have a default value of 10%). The function should remove all duplicates from the array and return the number of removed entries and the resulting array. If the percentage of removed numbers is bellow the given percentage, an error shall be raised. The input array can be generated with e.g np.random.randint(value, size=N), where value is the upper limit of the possible values and N the number of random values to be generated.

2. The prime 41, can be written as the sum of six consecutive primes:

$$41 = 2 + 3 + 5 + 7 + 11 + 13 \tag{1}$$

   This is the longest sum of consecutive primes that adds to a prime below one-hundred.

   What is the longest sum of consecutive primes below one-thousand that adds to a prime?

3. (*Spicy*) Was genau passiert hier? Wie würdest du die Funktion am besten benennen? (Tipp: Schau dir Rekursion in Bezug zu Python an)

Algorithm 4: Was passiert hier?

```
def geheim(n):
    if n > 1:
        return geheim(n-1) + geheim(n-2)

    return n

print(geheim(8))
```