

Informatics for Astronomers - WS2021

Roland Ottensamer, Marina Dütsch, Miguel Verdugo, Andreas Schanz

Exercise sheet 3 - Python basics and Data types

The following will be also part of the assessment:

(1) Try to present exercises in a way that everyone can understand (even those who didn't do the exercises), so please explain the vital parts of your solution in a clear way.

(2) Try to also include some background information where applicable, and/or explain the possible context/motivation for the given exercise.

1. The function `sys.getsizeof(object)` returns the size (in bytes) of a `python` object in memory. According to that function, How much memory a `float` uses **in python**?

Please explain the difference with respect to the the system requirements for a `float`.

2. Write a Python script that takes a string as a command-line argument, reverses the order of the letters, calculates its entropy and print it on the screen.
3. Write a simple python script that takes a distance in lightyears and returns that distance in both parsec and kilometers.

In the case of parsec, the script should return the proper SI notation (e.g. Gpc, Mpc, kpc depending on the distance). The distance in kilometers shall be written in proper scientific notation.

4. Assume that these two lists are 3D coordinates (vectors):

```
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)

5. `numpy` is likely the most important library in `python` and it is heavily used in scientific analysis.
 - How do `numpy` arrays differ from `python` lists?
 - Show the creation of `numpy` arrays with different properties (e.g. converting from a `list`, different step size and dimensions)
 - Apply some mathematical functions to arrays and comment the differences with using `lists`
6. `Python` provides a standard module (`timeit`) for timing the execution of scripts and pieces of code. Please time the execution of these two equivalents blocks

```

import numpy as np

N = 10**8
# Time from here
daten=[]
for i in range(0,N) :
    daten.append(i**0.5)
# to here

# and from here
daten = np.sqrt(np.arange(0,N))
# to here

```

- Which one is faster?
- For what values of N the effect is really noticeable. Please try a few wildly different values to have an idea.