Python Tech Tag

Hier gibt es Materialien für das Python Workshop am Freitag, dem 13. Dezember 2019, Tech-Tag des jetzigen adveisor Jahrgangs. Das Workshop wird zweisprachig gehalten; das bedeutet ihr dürft sowohl Deutsch als auch Englisch in der Kommunikation mit den Tutoren benutzen. Der Einfachheit halber werden wir ab jetzt in diesen Unterlagen ausschließlich Englisch benutzen.

English part starts here, German can be used when asking tutors.

Just before you get to working on the challenge, I will state my 3 main goals with this workshop:

> 1. Teach you the very basics of Python and the Linux Shell, in a manner that is approachable for total beginners;
> 2. Make it such that you can complete the tutorial independently at your work station in 90 minutes, only asking us questions when you need help;
> 3. Whet your apetite for more reading on your own and offer possible hints at how Python could be useful for your project work at adveisor.

Why Python? Why Python for adveisor?

Python is a *very* high level language, which offers you an often faster and more human-experience centric development process than traditional, compiled languages. Python is also the language of choice in many, many different fields in the industry and is *used by a lot of people with no STEM background*. Python is a fun and quite versatile place to start your journey into programming and eventually unlocking the **secrets of the Universe**.

adveisor, on the other hand, is a place where you explore and get to know **yourself** (and the world -- heck, there are people from like *25+ countries* taking part in it). And since Python enables you to use the full, raw power of **any computer anywhere**, I think it is a perfect tool to have on your hands when going out with such high goals in mind.

Just so you also get to know a bit about me and the motivation behind me ending up holding this workshop:

I'm in my fifth semester of Electrical Engineering here at TUM and was involved with adveisor for the past 2 years back-to-back and saw basically every facet of it first-hand, first as an *adveisee* (what you are now), and then as an *adveisor* (tutor with a group of *adveisees*). I want to give back to this program, so I figured an English-Language programming treasure hunt game with a Christmas Theme is just the way to do it.

So without further ado, let's get going.

How this workshop works

> 1. This is a programming treasure hunt game with a Christmas Theme.
>
> 2. There will be rewards.

> 3. Everything you need is in this repo. Pop us questions when having a hard time figuring something out on your own.

There will be 6 problems in total. You cannot progress to the next one unless you finished the previous problem. You should be able to finish in 90 minutes and you will have to submit your solution at the end. **Standard submissions should be by email, more details at the end.**

ALTERNATIVELY, if you know some git, you can fork the repo at https://github.com/munober/python-tutorial and submit your work as a pull request at the end. If you have no idea what I just said, ignore this part about git.

The workshop has been designed as individual work, but you can work with your peers if you feel like it. I have provided you with several cheat sheets by email, which you can also find in these documents. Google and the Internet in general are your best pals right now.

Challenges

Each challenge has its own separate folder, each folder contains Python (*.py*) files. Navigate using the command line.

Quick refresher for the Linux command line:

1. *ls* - lists contents of current folder.
2. *cd* "*folderName*" - switches into folder called *folderName*.
3. Use the tab key to auto-complete.
4. Use *python* "*program name.py*" to run a python script.

First challenge.

Basic arithmetics

1. Swap 2 numbers without using a third variable.

As this instruction states, you have to write an algorithm that switches two numbers between them, each originally stored in one variable, without introducing any other new variable.

> *For example:*
>
> a, b = 2,5 should become: a, b = 5,2 Write your code in *numberSwitch.py*.

2. Santa rocketship fuel problem.

Santa has become stranded at the edge of the Solar System while delivering presents to other planets! To accurately calculate his position in space, safely align his warp drive, and return to Earth in time to save Christmas, he needs you and his Elves to bring him measurements from fifty stars.

The Elves quickly load you into a spacecraft and prepare to launch, but they first need to determine the amount of fuel required.

Fuel required to launch a given module is based on the module's mass. Specifically, to find the fuel required for a module, take its mass, divide by three, round down, and subtract 2.

> *For example:*
>
> For a mass of 12, divide by 3 and round down to get 4, then subtract 2 to get 2. For a mass of 14, dividing by 3 and rounding down still yields 4, so the fuel required is also 2. For a mass of 1969, the fuel required is 654. For a mass of 100756, the fuel required is 33583. The Fuel Counter-Upper needs to know the total fuel requirement. To find it, individually calculate the fuel needed for the mass of each modul (your puzzle input), then add together all the fuel values.

What is the sum of the fuel requirements for all of the modules on your spacecraft?

Write your code in *rocketship.py* and use *rocketshipInput.txt* as your program input to compute the final answer.

Thanks to https://adventofcode.com/2019/day/1 for this problem!

Second challenge.

Sorting algorithm

Write a Python script which sorts a list using Insertion Sort.

See: https://en.wikipedia.org/wiki/Insertion_sort.

Write your code in *insertionSort.py* and use *sortInput.txt* as your program input to compute the final answer.

Third challenge.

String processing

Write a Python script to check for anagrams.

See: https://en.wikipedia.org/wiki/Anagram

Your script will have to print a '1' for every found anagram pair and a '0' for every given pair that isn't an anagram, such that you get a string of 1's and 0's. Write your code in *anagram.py* and use *anagramList.txt* as your program input to compute the final answer.

Fourth challenge.

Data Structures

1. Check your work!

To make sure the results you have provided are correct and your space travels really do take you back to Earth in time for Christmas, you need to double-check your math. For that, the Elves have written a handy script. Run *verify.py* to check your answers!

2. Create a class *Elf*

Do that in the same file as above: *dataStructures.py*. The class should contain: name, color, status. Instantiate an Elf called Oscar.

Fifth challenge.

Keys

In their rush to get everything ready, the Elves have misplaced the two keys needed to start the rocket engines. Help them find these HASH keys.

1. The Elves have lost the first key needed for liftoff in the html body of https://fratiloiu.com/python.

Find the key in the html body of the page, then add it in *dataStructures.py* under websiteKey.

2. Run *python dataStructures.py*

Do this in the terminal to progress to the next leg of your journey. If you did everything right, you will find the final, second key, needed to start the rocket engines in the file *keys.txt*.

Sixth challenge.

Almost there!

1. Submit your work.

Save your files in a *.zip* archive with your name as its file name and send it attached to teo@fratiloiu.com *Alternatively*, if you know git, send a pull request to this repo: https://github.com/munober/python-tutorial.

2. You're done.

Come get your well-deserved Lebkuchen.

Useful links:

Online Interpreters

1. https://www.onlinegdb.com/online_python_interpreter
2. https://ide.geeksforgeeks.org/
3. https://www.python.org/shell/
4. https://www.pythonanywhere.com/

Cheats sheets and documentation

1. https://docs.python.org/3/tutorial/
2. https://blog.finxter.com/python-cheat-sheet/