Pytuto (Python Tutorial) 3rd assignment.

Hyungwon Yang
Hyung8758@gmail.com
EMCS Lab

This task was designed for those who just started python. I wish them try to complete the task and improve their python skill. Please search as many solutions as you can through the internet and ask questions to me and your teammates to tackle down the tasks. However, DO NOT share codes. Good luck.

Introduction

Functions

Let's say you want to bake a cake. So you decide to drive to a market and buy some fresh ingredients such as flour, milk, eggs, and so on. In the kitchen, you put all the ingredients in front of you and start thinking how to bake a cake. You can search the cake recipe from the Internet and follow the given directions. In a bowl, you sift the dry ingredients such as flour, sugar, and salt with sifter and then pour eggs, water, and vinegar into the bowl and stir all the ingredients well with a spatula. In the end, you pour the mixture into a cake pan and bake it in an oven.

Attaining a good quality of ingredients might be important for making delicious cakes. However, **using good tools** is more important because it saves time (increase efficiency) and improves the quality of cakes. As you can see above, a bowl, sifter, spatula, and oven, all of them together, increase the efficiency of making a cake. Without those tools, the procedure might be very slow or the result (cake) might not be satisfactory.

In python, all modules are full of functions. These functions help programmers to write a script in an efficient way. If functions were not available in computer languages, writing a programming script must have taken much more time and energy. Let's look at a very simple example.

There are numbers 1 to 10. You want to square each of the numbers and sum them up.

```
>>> A = [1,2,3,4,5,6,7,8,9,10]
>>> B = []
>>> for add in range(1,len(A)):
>>>     B.append(add**add)
>>> C = sum(B)
```

If you run the script above, you will get ‘C’ as follows: 10405071317. The example might not be demanding, but it would be much simpler and easier to users if you make the script above as a function. Let’s make a function named ‘squareSum’.

```
>>> def squareSum(A):
>>>     B = []
>>>         for add in range(1,len(A)):
>>>         B.append(add**add)
>>>     C = sum(B)
>>>     return C
```

Python reads ‘def’ as a function. ‘squareSum’ is the name that you have made. (You can change the name as you desire) ‘A’ is an input. (You can assign the number of inputs as you wish) From now on, you can get the squared and summed numbers easily by utilizing this ‘squareSum’ function.

```
>>> A = [1,2,3,4,5,6,7,8,9,10]
>>> squareSum(A)
```

Writing a script with proper functions has a lot of advantages.

First, it saves time. If you have to do many things such as mathematical calculation or image processing, well-made functions amazingly reduce the time for writing scripts. Instead of using plus, minus, or multiplication signs to get mean and standard deviation, you can simply attain those values using ‘std’ or ‘mean’ functions. When it comes to image processing, difficult and complicated procedures such as Fourier transform and edge detection can be managed by ‘numpy.fft’ and ‘cv2.Canny’.

Second, it provides accurate and diverse results. When you search for a function that downsamples sound files, you would be astonished from the number of methods that modules support. Depending on your tasks, you can choose a specific downsampling function then you will get a highly accurate result.

There are still a number of reasons why functions are useful but I would not discuss it further because I want you to experience the importance of functions.

Task

A) Simple functions

A-1. **Make a ‘getmvs’ function.**

- Make a ‘getmvs’ function that print mean, variance and standard deviation.
- By using ‘getmvs’ function, any number arrays (vectors or matrices) have to be calculated and return the values of mean, variance and standard deviation. (You SHOULD NOT use ‘mean’,’variance’,’std’ or any other functions but

have to make formulas yourself)
look) numpy.random

A-2. **Make an 'imageMaker' function.**
- Generate an 'imageMaker function.
- Set two input arguments: 'image file' and 'filter name'. (use PIL module)
- The function imports an 'image file' and filters it based on 'filter name'.
- Set 3 filter options: BLUR, DETAIL, and FIND_EDGES in the function.
- Test the function with the 'peppers.png' image file whether the function works fine.
- The function should show the filtered image.

look) PIL.Image, PIL.ImageFilter

B) Game functions

B-1. **Make a 'fortuneCookie' function.**
- Generate a 'fortuneCookie' function.
- Set one input argument: 'sentence'. (only need a sentence)
- Separate each character in the input string and put it into any random variable name.
- Turn each character to number. (use ord)
- Make a syntax that chooses one number from them.

Ex) 'day' > 'd', 'a', 'y' > 100, 97, 121 > 121 (random choice)
- Write fortune telling sentences for 10 cases (in case that last digit is 0 to 9)

Ex) 0: Your life is wonderful, 1: BEWARE train is coming.
- Print the fortune telling sentence as an output.

If you finish your assignment, please send the code to 'hyung8758@gmail.com'. The script name should be '?th Assignment_YourName.py'. In the code script, the assignment number (e.g, 1st assignment), your name and email address should be written in the first line. Ask questions and give comments to me. It is always welcome.

Weekly Tips

- os : os 모듈은 전반적인 컴퓨터 시스템과 관련된 작업을 수행할수 있게 도와준다. >>설명 : https://docs.python.org/2/library/os.html 몇가지 중요한 메서드를 알아보자.

  path : os.path.abspath(path) : 현재 경로를 prefix 로 하여 입력받은 경로를 절대경로로 바꿔서 반환합니다.
  os.path.exists(path) : 입력받은 경로가 존재하면 True 를 존재하지 않을경우

False 를 반환합니다.

os.path.isdir(path) : 입력받은 경로라 디렉터리이면 True 를 아닐경우 False 를 반환합니다.

os.path.join(path1,path2) : 입력받은 각각의 path 를 연결합니다.

os.getcwd() : 현재 디렉터리 위치를 알려준다.

os.chdir() : 입력받은 위치로 이동한다. 터미널의 cd 기능이다.

os.mkdir() : 디렉터리를 생성한다.

os.remove() : 파일을 삭제한다. os.rmdir()

os.rename(src,dst) : src 를 dst 로 이름을 변경하거나 이동한다. 파일, 디렉터리 모두 적용된다.

os.listdir() : 해당경로에 존재하는 파일과 디렉터리 리스트를 반환한다.

os.environ : 현재 설정된 path 값을 모두 반환한다.