

Pytuto (Python Tutorial) 5th 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

Class II

Python class is very useful but difficult to use at the first time. In this chapter, we will go into more depth on the class.

- Instance

When you finish writing a class and start using it, you have to assign the class as a variable first. (you cannot use a class itself.) This assigned variable called instance. Let's look at the example of making an instance.

```
>>> class Shout(object):
>>>     def human(self):
>>>         print 'YES!'
>>>
>>> S = Shout() # You make an instance of the class named Shout.
>>> S.human() # You are able to use class function.
'YES!'
```

- Self

If you have read scripts written in python, you might have seen 'self' in class often. You may wonder why self needs to be used in a class. However, it is very useful when you need to keep variables in the class. Let's say you have 3 functions in your class and those functions take the same input variables.

```
>>> class Walk():
>>>     def setValue(self, personName, oneStep):
>>>         self.name = personName
>>>         self.step = oneStep
>>>     def shortwalk(self):
>>>         print 'my name is %s' % self.name
>>>         return self.step * 3
>>>     def longwalk(self):
```

```
>>>         print 'my name is %s' % self.name
>>>         return self.step * 7
```

After you made the 'Walk' class, type as below.

```
>>> W = Walk()
>>> W.setValue('Sean', 5)
>>> W.longwalk()
my name is Sean
35
```

You don't need to assign the same variables in different functions every time.

- `__init__`

You can think of it as an initiator. If your class has this function at the beginning then this '`__init__`' function activates itself whenever the class is called.

```
>>> class Greeting(object):
>>>     def __init__(self):
>>>         print 'welcome to python'
>>>         self.good = 'good afternoon'
>>>     def writing(self):
>>>         print self.good
```

```
>>> G = Greeting()
welcome to python
```

Task

A) Restaurant class

- Make a class named 'yourname_restaurant'.
Ex) 'HyungWon_restaurant'
- Print 'welcome to our restaurant' and the menu. Set 5 lists of main dishes and their prices (write your favorite dishes). It needs to be displayed when the class is instanced.
- Make 3 functions named 'order', 'leave', and 'tip' functions
- Order function prints 'May I take your order?' first. Use input function to get the order from the 5 lists. Make a pause for 5 seconds and print '-ordered dish name- is here, please enjoy'.
look) time.sleep
ex) 'Spaghetti is here, please enjoy'
- Leave function prints 'Thank you. Please visit us next time again.'

- Tip function takes number input. If the input is smaller than 30 print 'Thank you very much', if the input is between 31 to 100 print 'Wow I'm really appreciate it', and if the input is over 100 print 'Oh my lord!'

B) Dice game class

- Make a class named 'DiceGame'
 - Make a dice rolling game in your own way. It needs at least 3 functions and 'self' needs to be used in every function. Use '__init__' function as well. Refer to the main rules below (As long as you follow the main rules bellows, you can change and add minor rules in your own way).
 - A user has to choose dice from 3 different shapes. (for example, hexagon, dodecagon dices. Please search the internet and check the dice shapes and number location)
 - When the game starts (roll the dice) random number result should print. (you have to decide what to do with that results. For example, make the user bet the money, give only 1 trial, based on the result give him a reward or nothing.)
- DO NOT import dice module.**
- The game does not need to be complicated but should be **fun** and **interesting**.

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

* 파일 입출력: pickle 모듈이 도움이 된다. 참고하자.

- rjust or ljust 를 통해 통일성을 갖춰서 프린트 할 수 있다.

```
word = 'right now'
word.rjust(30)
>>> '          right now'
for x in range(1,6):
    print(x, '*', x, '=', str(x*x).rjust(5) )
```

- 포매팅

```
print("{0} is {1}".format("apple", "red"))
dic = {"item": "apple", "color": "red"}
print("{0[item]} is {0[color]}".format(dic)) or
print("{item} is {color}".format(**dic))
```

- 글 쓰기, 열기

`f = open('test.txt','r')` # r: 읽기모드, w = 쓰기모드, a = 쓰기+이어쓰기 모드, b= 바이너리 모드(mp3 같은 파일은 바이너리로 해야함 ex. 'rb')

`f.read()`

`f.close`

`f.closed` # 제대로 닫혔는지 확인, 제대로 닫혔다면 True 값 반환

`f.seek(0)` # 처음으로 돌아간다

`f.readline()` # 처음부터 줄 단위로 읽는다.

`f.readlines()` # 줄단위로 모두 리스트로 읽어온다.