Pytuto (Python Tutorial) 4th 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

Once you get used to making functions and using them, you are ready to write your own script. However, what if you make more than just one function or two or more. In this case, you want to put them together in one group. You can put similar functions into one class and use that class whenever you need to use functions.

Let's say you want to make functions for game. First, you think you have to make a class named character and then build movement and speaking functions.

```
>>> class Character(object):
>>>     def forward:
>>>             if right_arrow == 1:
>>>                     self.movement += 1
>>>     def backward:
>>>             if back_arrow == 1:
>>>                     self.movement -= 1
>>>     def needHelp
>>>             if character_HP < 30:
>>>                     print help me!
```

(This is an example so it won't work in python.)

As you can see above, you have to write 'class' at the beginning and type the class name. Mostly first character in the class name is capitalized. (You might wonder what object is but you will get it sooner or later.) If you generate class name and put functions below it then you just made a class. AWESOME.

You kept writing functions and scripts for making a game and realized that there are some functions need to be used twice or more. It is very inefficient write same functions in many classes. In order to solve this problem python suggests you with inheritance that a class can inherit functions from other classes.

Let's say you want to make a 'Monster' class. You want the monster moves forward and backward as character does. In this case you can inherit a 'Character' class to a 'Monster' class so as to use movement functions.

```
>>> class Monster(Character):
>>>     def kick:
>>>             if character_distance < 3:
>>>                     self.kick()
>>>                     print wawawa uga wawa
```

By inheriting 'Character' class, 'Monster' class doesn't have to add movement functions. If you want to inherit more than one class then you can add more.

```
>>> class Monster(Character, Ghost, Magic):
```

Why don't you search for 'self' and 'instance'? These are also important concepts when you generate class.

Task

A) Simple Class

Sean: Hello Python programmer. This is Sean. I just heard that you are really good at computer languages. So I want you to do something for me. I want a simple module that can be used for simple calculation. I will jot down detail information below. Just let me know if you are available. Thanks!

Request: Make a class name **'Calculator'** that contains **'summation'**, **'subtraction'**, **'division'**, **'multiplication'** functions. Therefore, You need to make a **'Calculator'** class first and then put 4 functions below it. All functions should accept two or more numbers as inputs and run without errors.

B) Class Inheritance

Tailor: Hi there. I heard that you made a calculator for Sean. I think that's really smart because I don't get it how to make a calculator with python. Anyway, I wonder whether you can make a more interesting module for me. All I want to do is importing and manipulating image and sound. I will look forward to it. Thanks. p.s. I will attach two image and sound files.

Request:

1. Make a class named **'ImageControl'** and generate 5 functions**: 'imgImport'** that imports image, **'blurring'** that blurs the imported image, '**sharpening'** that sharpens the image, **'resizing'** that resizes the image and **'imgSave'** that saves the image.
2. Make a class named **'SoundControl'** and generate 5 functions: **'soundImport'** that imports the sound, **'resampling'** that resamples the sound, **'noising'** that add noise in the sound, **'soundPlot'** that plot the sound wave, **'soundSave'** that saves the sound.
3. Make a class named **'SoundNImage'** that inherits **'ImageControl'** and **'SoundControl'** classes. You have to check whether Tailor can use all functions from **'ImageControl'** and **'SoundControl'** classes only with the **'soundNImage'** class.
   look) matplotlib.pyplot, pyglet, wave, PIL.Image, PIL.ImageFilter

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

* 클래스에 대해 알아보자.

- 클래스(class)의 이름공간에 속한 멤버 함수를 메서드라고 한다.
- 클래스를 필요할때마다 인스턴스 객체를 생성하여 사용하면 된다. 인스턴스 객체는 생성이 완료된 후, 원본 클래스와 동일한 데이터와 함수를 가지고 있다.
- 클래스 내 정보를 함부로 변경할 수 없도록 하기 위해 정보은닉(information hidding)을 사용한다.
- 공통적으로 사용되는 데이터를 클레스들로부터 추출하여 기본 클래스를 작성하는 작업을 추상화(abstraction)라고 한다.

  추상화 결과로 작성된 클래스를 부모클래스(super class), 이를 받아서 각 특성을 추가한 클래스를 자식클래스(sub class)라고 한다.

  이들은 각기 상속관계를 형성하고 있다. 부모클래스가 2 이상일 경우 다중상속이라고 한다.
- 인스턴스 객체의 데이터가 변경되면 클래스 객체의 데이터와 구분하기 위하여 인스턴스 객체 이름공간에 변경된 데이터를 저정한다. 반면에 아직 변경되지 않은

데이터와 메서드는 여전히 클래스 객체와 공유하고 있다.

- 기본적으로 클래스의 메서드는 클래스 객체의 이름공간에 선언된다. 이러한 이유로 인하여 인스턴스 객체가 클래스의 메서드를 호출하면, 자기 이름공간에 대한 정보를 호출하는 메서드에게 넘겨줘야 한다. 메서드 호출시 암묵적으로 첫 인자로 인스턴스 객체를 넘기는 호출 방식을 바운드 메서드(bound method)호출이라 한다. 이때에는 메서드 정의시 첫 인자가 인스턴스 객체임을 선언하나, 호출시에는 자동으로 반영되기에 명시적으로 입력하지 않는다. 반면에 메서드 호출시 명시적으로 첫 인자로 인스턴스 객체를 넘기는 호출방식을 언바운드 메서드 호출이라고 한다. 이때에는 클래스 객체를 통하여 메서드를 호출하며, 첫 인자로 인스턴스 객체를 입력하여야한다

ex. p1.Print() <바운드 호출 : pi 는 Person 이라는 클래스의 인스턴스, 그 안에는 print 메서드가 있다.

Person.Print(pi) < 언바운드 호출

- 클래스 객체와 인스턴스 객체의 이름공간이 다르고 인스턴스 객체를 통해 변수나 함수 이름을 찾는 경우 순서는 다음과 같다.

인스턴스 객체 영역 > 클래스 객체 영역 > 전역 영역