

Python implementation of OOP I (Class, Instance, Method)

ESM2014-41

객체지향프로그래밍 및 실습

SKKU 시스템경영공학과

조영일

Python : Pure Object-oriented Language

- Python을 구성하는 것은 모두 객체이다.
- Python의 모든 자료형(type)은 Class로 표현됨
- Python의 모든 변수는 Object로 표현됨

```
>>> type(10)
<class 'int'>
>>> type("A string")
<class 'str'>
>>> type([1, 2, 3])
<class 'list'>
>>> type({1, 2, 3})
<class 'set'>
```

Class 정의

```
class Movie:
    # Class Attribute
    name = ""
    genre = ""
    director = ""
    rating = 0 # Initial Value

    # Class Method
    def show_name(self):
        print("Movie name is " + self.name)

    def show_rating(self):
        print("Rating : " + str(self.rating))
```

Object from Class

```
new_movie = Movie()
new_movie.name = "Lion King"
new_movie.genre = "Animation"
new_movie.director = "Disney Pictures"

new_movie.show_name()
>>> "Lion King"
new_movie.show_rating()
>>> 0
```

def __init__ : Class Initializer

- def __init__ : Magic Method – Class Initializer(초기화자), 객체의 초기화를 담당하는 method. 객체가 최초로 생성될 때 호출되어 객체 내의 Attribute 값을 지정하고, 객체 생성 시점에 실행되어야 하는 로직들을 포함 시킬 수 있음. (주의 생성자(__new__)와는 다름)

```
class Movie:
    def __init__(self, name, genre, director, rating=0):
        self.name = name
        self.genre = genre
        self.director = director
        self.rating = rating

    # Class Method
    def show_name(self):
        print("Movie name is " + self.name)

    def show_rating(self):
        print("Rating : " + str(self.rating))

new_movie = Movie(name="Lion King", genre="animation", director="Disney Pictures")
new_movie.show_name()
```

self keyword & Class Method

- self : class로 부터 생성된 객체 자기 자신을 의미하며, class method의 경우 반드시 method 정의 시점에 첫번째 인수로 self를 명시적으로 기입하여야 함
- class method는 class 코드 블록 내에 정의 되며, 해당 class로 부터 생성되는 모든 객체는 해당 method를 가지게 된다.

```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def change_radius(self, radius):
        self.radius = radius

    def get_area(self):
        return self.radius * self.radius * 3.141592

c = Circle(radius=5)
c.change_radius(6)
c.get_area()
>>> 113.097312
```

Magic Method in Class

- Magic Method : 일반적인 Method 들과는 달리 특정한 기능을 하기 위해 Python Syntax 상 지정되어 있는 Method 들(__xx__)
- `__new__` method : 생성자, 객체 생성단계 최초로 호출됨(거의 사용되지 않는다.)
- `__init__` method : 초기화자, 객체 생성 시 `__new__` 다음으로 호출되어 객체의 내용을 초기화(initialize) 하는데 사용됨
- `__del__` method : 소멸자, 객체가 소멸될 때의 동작을 정의
- `__repr__` method : 객체를 프린팅(print()) 했을 때 표시될 값을 정의

Practice

- 다음과 같은 Attribute / Method 를 가지는 Class를 작성하고, Class 로 부터 Object를 생성해보세요.(__init__ 포함)
- Student Class
 - Attribute
 - 이름 <string>
 - 학번 <string>
 - 전공 <string>
 - 수강교과목 <list>
 - 수강교과목 중간고사 점수 <dict>
 - 중간고사 평균 점수 <float>
 - Method
 - __init__ : 초기화 시 이름, 학번, 전공, 수강교과목, 수강교과목 중간고사 점수를 arguments로 받아 객체의 attribute에 할당하고, 중간고사 평균 점수는 해당 자료를 토대로 계산하여 넣는다.
 - get_lecture_count : 수강교과목의 개수를 반환하는 함수
 - get_summary : 해당 학생의 attribute 를 요약하여 print()하는 함수
 - change_major : 해당 학생의 전공 attribute 를 변경하는 함수

Summary

- Class Definition & Object
- Class_INITIALIZER
- Class Method
- Class Magic Method