

[클래스 멤버와 인스턴스 멤버]

멤버에는 클래스 멤버와 인스턴스 멤버 두 가지가 있다.

클래스 멤버는 Class name space에 생성되지만, 인스턴스 멤버는 Instance name space에 생성된다.

클래스 멤버는 모든 객체에서 공유되며, 인스턴스 멤버는 각각의 객체에서만 참조할 수 있다.

```
class foo:
    class_name_space_var = 20                # 클래스 멤버

    def set_instance_name_space_var(self, x):
        self.instance_name_space_var = x    # 인스턴스 멤버
    def get_instance_name_space_var(self):
        return self.instance_name_space_var

def test_member():
    p = foo()
    p.set_instance_name_space_var(10)

    # Instance name space에 class_name_space_var가 없으므로
    # class name space에서 class_name_space_var 변수를 찾는다.

    print('{0}, {1}'.format(p.instance_name_space_var,
                             p.class_name_space_var))

test_member()
```

[인스턴스 메서드, 클래스 메서드, 정적 메서드]

클래스 메서드는 클래스를 인스턴스화한 객체들이 공통적으로 사용할 수 있는 메서드이고,

인스턴스 메서드는 인스턴스가 사용할 수 있는 메서드이다.

따라서 같은 클래스를 인스턴스화 해도 사용할 수 있는 메서드는 다를 수 있다.

Python에서 인스턴스 메서드와 클래스 메서드의 차이는 인스턴스 멤버에 접근할 수 있는지 없는지에 대한 차이이기 때문에,

따라서 메서드를 정의할 때 첫 번째 파라미터에 `self`를 작성하면 인스턴스 메서드, 그렇지 않으면 클래스 메서드이다.

정적 메서드도 클래스 메서드와 같이 인스턴스 멤버에 접근할 수 없는 메서드이다.

더불어서 클래스 메서드가 접근할 수 있는 클래스 멤버에도 접근 할 수 없는 메서드가 정적 메서드이다.

정적 메서드와 클래스 메서드는 장식자(자바의 어노테이션 같은 것)를 통해 구분하며,

클래스 메서드를 정의할 때 첫 번째 매개변수로 클래스 멤버에 접근할 수 있는 `cls`를 작성한다.

(`self`와 마찬가지로 관례상 `cls` 라는 이름을 사용합니다.)

```
class foo:
    a = 0                                # class member

    def instance_method(self):
        self.b = 10                     # instance member
        print("instance method called")

    @classmethod
    def class_method(cls):
        print("class method called")
        return cls.a                    # class member에 접근 가능

    @staticmethod
```

```
def static_method():
    print("static method called")

f = foo()
f.instance_method()
f.class_method()
f.static_method()
```

[생성자와 소멸자]

생성자는 객체가 생성될 때 실행되고, 소멸자는 객체가 소멸할 때 실행된다.

생성자는 `__init__` 속성으로, 소멸자는 `__del__` 속성으로 정의하면 된다.

```
class foo:
    count_of_instance = 0

    def __init__(self):
        foo.count_of_instance += 1

    def __del__(self):
        foo.count_of_instance -= 1

f1 = foo()
f2 = foo()
print(foo.count_of_instance)    # 2
del f2
print(foo.count_of_instance)    # 1
```