

Unit 35.

☀ Status	완료
👤 담당자	
📅 마감일	
📅 완료일	@2022년 11월 30일

Unit 35. 클래스 속성과 정적, 클래스 메서드 사용하기

35.1 클래스 속성과 인스턴스 속성 알아보기

- 속성에는 클래스 속성과 인스턴스 속성 두 가지 종류가 있습니다. `__init__` 메서드에서 만들었던 속성은 인스턴스 속성이다.

35.1.1 클래스 속성 사용하기

- 클래스 속성은 다음과 같이 클래스에 바로 속성을 만든다.

```
class Person:
    bag = []

    def put_bag(self, stuff):
        self.bag.append(stuff)

james = Person()
james.put_bag('책')

maria = Person()
maria.put_bag('열쇠')

print(james.bag)
print(maria.bag)
```

james와 maria 인스턴스를 만들고 각자 `put_bag` 메서드로 물건을 넣었는데, `james.bag`과 `maria.bag`을 출력해보면 넣었던 물건이 합쳐져서 나온다 그 이유는 클래스 속성은 클래스에 속해 있으며 모든 인스턴스에서 공유한다.

- 클래스 속성에 접근할 때는 클래스 이름으로 접근하면 좀 더 코드가 명확해진다.

35.1.2 인스턴스 속성 사용하기

- 물건들을 공유하지 않으려면 `bag`을 클래스 속성이 아닌 인스턴스 속성으로 만들면 된다.

```
class Person:
    def __init__(self):
        self.bag = []

    def put_bag(self, stuff):
        self.bag.append(stuff)

james = Person()
james.put_bag('책')

maria = Person()
maria.put_bag('열쇠')

print(james.bag)
print(maria.bag)
```

인스턴스 속성은 인스턴스별로 독립되어 있으며 서로 영향을 주지 않는다.

- 클래스 속성: 모든 인스턴스가 공유. 인스턴스 전체가 사용해야 하는 값을 저장할 때 사용
- 인스턴스 속성: 인스턴스별로 독립되어 있음. 각 인스턴스가 값을 따로 저장해야 할 때 사용

35.2 정적 메서드 사용하기

인스턴스를 통하지 않고 클래스에서 바로 호출할 수 있는 정적 메서드와 클래스 메서드

정적 메서드는 메서드 위에 `@staticmethod`를 붙인다. 이때 정적 메서드는 매개변수에 `self`를 지정하지 않습니다.

```
class 클래스이름:
    @staticmethod
    def 메서드(매개변수1, 매개변수2):
        코드
```

정적 메서드를 호출할 때는 다음과 같이 클래스에서 바로 메서드를 호출하면 됩니다.

```
Calc.add(10, 20)    # 클래스에서 바로 메서드 호출
Calc.mul(10, 20)    # 클래스에서 바로 메서드 호출
```

정적 메서드는 self를 받지 않으므로 인스턴스 속성에는 접근할 수 없다.

정적 메서드는 인스턴스의 상태를 변화시키지 않는 메서드를 만들 때 사용한다.

35.3 클래스 메서드 사용하기

클래스 메서드는 다음과 같이 메서드 위에 @classmethod를 붙인다 이때 클래스 메서드는 첫번째 매개변수에 cls를 지정해야 한다(cls는 class에서 따온것)

```
class 클래스이름:
    @classmethod
    def 메서드(cls, 매개변수1, 매개변수2):
        코드
```

```
class Person:
    count = 0    # 클래스 속성

    def __init__(self):
        Person.count += 1    # 인스턴스가 만들어질 때
                             # 클래스 속성 count에 1을 더함

    @classmethod
    def print_count(cls):
        print( '{0}명 생성되었습니다.'.format(cls.count))    # cls로 클래스 속성에 접근

james = Person()
maria = Person()

Person.print_count()    # 2명 생성되었습니다.
```

@classmethod를 붙여서 클래스 메서드를 만든다. 클래스 메서드는 첫 번째 매개변수가 cls인데 여기에는 현재 클래스가 들어오므로 cls.count처럼 cls로 클래스 속성 count에 접근할 수 있습니다.

클래스 메서드는 정적 메서드처럼 인스턴스 없이 호출할 수 있다는 점은 같다. 하지만 클래스 메서드는 메서드 안에서 클래스 속성, 클래스 메서드에 접근해야 할 때 사용해야 한다.