Unit 34.

Status	완료
담당자	
마감일	
완료일	@2022년 11월 29일

Unit 34. 클래스 사용하기

- 클래스는 객체를 표현하기 위한 문법

34.1 클래스와 메서드 만들기

- 클래스는 class에 클래스 이름을 지정하고 :(콜론)을 붙인 뒤 다음 줄부터 def로 메서드를 작성하면 된다.

```
>>> class Person:
...     def greeting(self):
...         print('Hello')
...
```

- 클래스 이름을 짓는 방법은 변수와 같습니다. 보통 파이썬에서는 클래스의 이름은 대문자로 시작
- 메서드 작성 방법은 함수와 같으며 코드는 반드시 들여쓰기를 해야 하고 메서드의 첫 번째 매개변수는 반드시 self를 지정

1. **인스턴스 = 클래스()**

```
>>> james = Person()
```

클래스는 특정 개념을 표현만 할뿐 사용을 하려면 인스턴스를 생성해야 한다.

34.1.1 메서드 호출하기

- 메서드는 클래스가 아니라 인스턴스를 통해 호출한다.
- 인스턴스를 통해 호출하는 메서드를 인스턴스 메서드라고 한다
- **인스턴스.메서드()**

```
>>> james.greeting()
Hello
```

34.1.2 파이썬에서 흔히 볼 수 있는 클래스

- 지금까지 사용한 int, list, dict 등도 사실 클래스

```
>>> a = int(10)
>>> a
10
>>> b = list(range(10))
>>> b
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> c = dict(x=10, y=20)
>>> c
{'x': 10, 'y': 20}
```

34.2 속성 사용하기

- 클래스에서 속성을 만들고 사용하기
- 속성을 만들 때는 _init_ 메서드 안에서 self.속성에 값을 할당
- _ init _ 메서드는 james = Person()처럼 클래스에 ( )(괄호)를 붙여서 **인스턴스를 만들 때 호출되는 메서드이고 인스턴스(객체)를 초기화한다.**

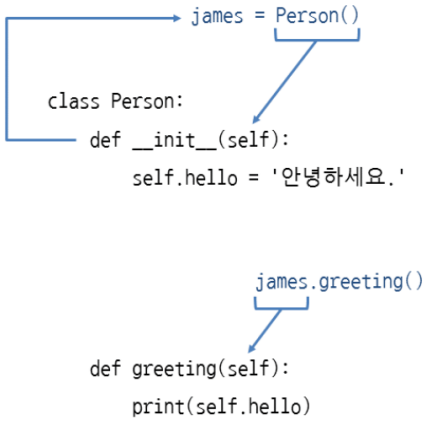
```
class Person:
    def __init__(self):
        self.hello = '안녕하세요.'

    def greeting(self):
        print(self.hello)

james = Person()
james.greeting()    # 안녕하세요.
```

34.2.1 self의 의미

- self는 인스턴스 자기 자신을 의미한다
- 인스턴스가 생성될 때 self.hello = '안녕하세요.'처럼 자기 자신에 속성을 추가



34.2.2 인스턴스를 만들 때 값 받기

- __init__ 메서드에서 self 다음에 값을 받을 매개변수를 지정하고 매개변수를 **self.속성**에 넣어준다.

```
class 클래스이름:
    def __init__(self, 매개변수1, 매개변수2):
        self.속성1 = 매개변수1
        self.속성2 = 매개변수2
```

```
class Person:
    def __init__(self, name, age, address):
        self.hello = '안녕하세요.'
        self.name = name
        self.age = age
```

```
        self.address = address

    def greeting(self):
        print('{0} 저는 {1}입니다.'.format(self.hello, self.name))

maria = Person('마리아', 20, '서울시 서초구 반포동')
maria.greeting()    # 안녕하세요. 저는 마리아입니다.

print('이름:', maria.name)       # 마리아
print('나이:', maria.age)        # 20
print('주소:', maria.address)    # 서울시 서초구 반포동
```

1. **init** 메서드를 보면 self 다음에 name, age, address를 지정했다. 그리고 메서드 안에서는 self.name = name처럼 매개변수를 그대로 self에 넣어서 속성으로 만들었다.

```
   def __init__(self, name, age, address):
        self.hello = '안녕하세요.'
        self.name = name
        self.age = age
        self.address = address
```

2. greeting 메서드는 인사를 하고 이름을 출력하도록 수정

```
    def greeting(self):
        print('{0} 저는 {1}입니다.'.format(self.hello, self.name))
```

3. Person의 ( )(괄호) 안에 이름, 나이, 주소를 콤마로 구분해서 넣은 뒤에 변수에 할당한다

`maria = Person('마리아', 20, '서울시 서초구 반포동')`

34.3 비공개 속성 사용하기

- 클래스 바깥에서는 접근할 수 없고 클래스 안에서만 사용할 수 있는 비공개 속성
- 비공개 속성은 __**속성**과 같이 이름이 __(밑줄 두 개)로 시작

```
class 클래스이름:
    def __init__(self, 매개변수)
        self.__속성 = 값
```

```
class Person:
    def __init__(self, name, age, address, wallet):
        self.name = name
        self.age = age
        self.address = address
        self.__wallet = wallet    # 변수 앞에 __를 붙여서 비공개 속성으로 만듦

maria = Person('마리아', 20, '서울시 서초구 반포동', 10000)
maria.__wallet -= 10000    # 클래스 바깥에서 비공개 속성에 접근하면 에러가 발생함
```

클래스 바깥에서 maria.__wallet으로는 접근할 수 없다.