

# 31장

## 객체의 타입을 표현하는 클래스 만들기

# 31장 객체의 타입을 표현하는 클래스 만들기

---

31.1 클래스를 사용해 새로운 객체 타입 구현하기

31.2 데이터 속성을 나타내는 객체 프로퍼티

31.3 객체의 동작과 연산인 메서드

31.4 직접 정의한 객체 타입 사용하기

31.5 매개변수를 받는 `__init__`을 사용해 클래스 만들기

31.6 객체가 아닌 클래스 이름에 마침표 표기를 사용하는 경우

31.7 요약



## 31 객체의 타입을 표현하는 클래스 만들기

- » 새로운 객체 타입을 구현하려면 그 타입에 속하는 객체를 이루는 프로퍼티를 정의하고, 그 타입의 객체가 수행할 수 있는 동작(메서드)을 정의해야 한다.
- » 보통 직접 객체 타입을 정의하는 이유는 프로퍼티와 동작을 원하는 대로 만들고, 나중에 재활용하기 위함이다.
- » 클래스(class)를 사용해 객체 타입을 정의하기 전에, 다음 두 질문을 던져 클래스를 어떻게 구현할 지 구상해야 한다.
  - 객체가 어떤 것(특성 또는 프로퍼티)들로 구성되는가?
  - 객체가 어떤 일(동작 또는 연산)을 하기를 원하는가?

## 31.1 클래스를 사용해 새로운 객체 타입 구현하기

---



# 31.1 클래스를 사용해 새로운 객체 타입 구현하기

» 객체 타입을 만들 때 우선 해야 할 일은 클래스를 정의하는 것이다.

- class라는 키워드를 사용하면 클래스를 정의할 수 있다.
- 원을 표현하는 객체를 만들고 싶다고 가정하자

```
class Circle(object):
```

## 31.2 데이터 속성을 나타내는 객체 프로퍼티

---



## 31.2.1 객체를 `__init__`으로 초기화하기

» 객체를 초기화하기 위해서는 `__init__`라는 특별한 연산을 구현해야 한다(init 앞뒤에 밑줄이 두개씩 오는 것에 유의하라).

```
class Circle(object):
    def __init__(self):
        # 초기화 코드
```

» `__init__` 메서드 안에 있는 코드는 보통 객체를 정의하는 데이터 속성을 초기화한다.

- circle 클래스에서 반지름을 나타내는 radius라는 속성을 다음과 같이 0으로 초기화할 수 있다.
- self라는 변수는 새로 만들어진 Circle타입의 객체를 가리킨다. 여러분이 생성한 모든 원 객체는 내부에 self.radius로 접근할 수 있는 자기 자신만의 radius를 가진다.

```
class Circle(object):
    def __init__(self):
        self.radius = 0
```

» 셀프 체크 31.1

» 셀프 체크 31.2

## 31.3 객체의 동작과 연산인 메서드

---





## 31.3 객체의 동작과 연산인 메서드

» 객체에는 그 객체에 대해 수행하거나, 다른 객체와 함께 사용할 수 있는 여러 동작을 정의할 수 있다.

- 동작은 메서드를 통해 구현된다. 원의 경우 다른 메서드를 작성해 반지름을 변경할 수 있다.

```
class Circle(object):
    def __init__(self):
        self.radius = 0

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

- 메서드는 함수와 비슷하다. `__init__` 메서드와 마찬가지로 메서드의 첫 번째 매개변수는 `self`여야 한다.
- 메서드 정의에서 메서드 이름이 `change_radius`이고 `radius`라는 인자를 하나 받는다는 사실을 알 수 있다.
- 이 메서드의 본문은 한 줄뿐이다. 클래스의 데이터 속성을 변경하고 싶으므로 메서드 안에서 `self`를 사용해서 값을 바꿔야 한다



## 31.3 객체의 동작과 연산인 메서드

» 원의 다른 동작으로 자신의 반지름을 알려주는 것을 들 수 있다.

```
class Circle(object):
    def __init__(self):
        self.radius = 0

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

    def get_radius(self):
        return self.radius
```

- 이번에도 get\_radius는 메서드지만, self 이외의 다른 매개변수는 받지 않는다.
- 이 메서드가 하는 일은 radius라는 객체의 프로퍼티를 반환하는 것뿐이다.
- 이전과 마찬가지로 self.를 radius 앞에 붙여야 데이터 속성을 반환할 수 있다

» 셀프 체크 31.3

## 31.4 직접 정의한 객체 타입 사용하기

---



## 31.4 직접 정의한 객체 타입 사용하기

» 직접 정의해서 만든 객체 타입은 Circle 클래스 정의를 이용해서 새로운 Circle 타입의 객체를 만들어야 한다.

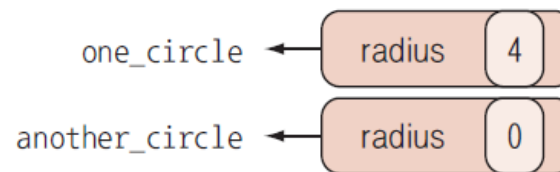
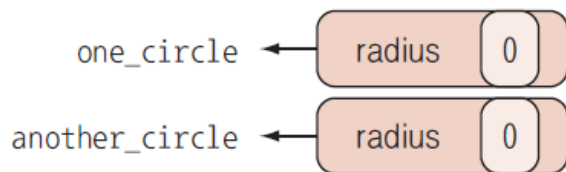
- one\_circle이라는 변수가 Circle 클래스의 인스턴스(instance)인 객체와 연결됐다고 말한다.
- 클래스를 호출하면 객체가 생기고, 이 객체를 변수에 대입하면 원하는 만큼 객체를 생성할 수 있다.

```
one_circle = Circle()
another_circle = Circle()
```

```
one_circle.change_radius(4)
```

```
print(one_circle.get_radius())
print(another_circle.get_radius())
```

4  
0



## 31.4 직접 정의한 객체 타입 사용하기



» 셀프 체크 31.4

## 31.5 매개변수를 받는 `__init__` 을 사용해 클래스 만들기

---



## 31.5 매개변수를 받는 `__init__`을 사용해 클래스 만들기

» 직사각형을 표현하는 클래스

```
class Rectangle(object):
    """ length(길이)와 width(너비)를 가지는 직사각형 객체 """
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def set_length(self, length):
        self.length = length

    def set_width(self, width):
        self.width = width
```



## 31.5 매개변수를 받는 `__init__`을 사용해 클래스 만들기

» `__init__` 메서드에 `self`를 제외하고도 두 가지 매개변수가 더 있다. 새 `Rectangle` 객체를 만들려면 반드시 두 값을 인자로 넘겨야 한다. 각각은 길이와 너비에 해당한다.

```
a_rectangle = Rectangle(2,4)
```

» 만약 다음 코드와 같이 인자를 하나만 넣으면 어떻게 될까?

```
bad_rectangle = Rectangle(2)
```

» 파이썬은 `Rectangle` 타입의 객체를 초기화할 때 인자가 두 개 필요한데, 여러분이 인자를 하나만 넘겼다는 사실을 알려준다.

```
TypeError: __init__() missing 1 required positional argument: 'width'
```

» `__init__`에서 데이터 속성과 매개변수 이름이 같다.

- 둘이 꼭 같을 필요는 없다. 하지만 보통 매개변수 이름과 데이터 속성의 이름을 일치시키는 경우가 많다.



## 31.6 객체가 아닌 클래스 이름에 마침표 표기를 사용하는 경우

---



## 31.6 객체가 아닌 클래스 이름에 마침표 표기를 사용하는 경우

```
c = Circle()
c.change_radius(2)
r = c.get_radius()
print(r)
```



```
c = Circle()
Circle.change_radius(c, 2)
r = Circle.get_radius(c)
print(r)
```

객체를 초기화한 다음, 클래스 이름을 사용하면서  
self가 될 객체를 직접 지정하는 더 명시적인 방법을  
사용해서 같은 일을 할 수도 있다.

## 31.7 요약

---



## 31.7 요약

- » 클래스는 객체의 타입을 정의한다.
- » 클래스는 데이터 속성(프로퍼티)과 메서드(연산)을 정의한다.
- » `self`는 (메서드에서) 관례적으로 사용하는 매개변수 이름으로 그 클래스에 속하는 객체의 인스턴스를 가리킨다.
- » `__init__` 메서드는 객체를 초기화하는 방법을 지정하는 특별한 연산이다. 객체가 생성될 때 `__init__`이 호출된다.
- » 다른 메서드(클래스 안에 정의된 `self`를 첫 번째 매개변수로 받는 함수)를 정의하여 객체가 다른 연산을 수행하게 할 수 있다.
- » 클래스를 사용하려면 연산을 수행하거나 데이터 속성에 접근하고자 하는 대상 객체에 대해 마침표 표기법을 사용한다.



## 31.7 요약

» (Q31.1) Circle 클래스에 `get_area`라는 메서드를 추가하라. 이 메서드는 원의 넓이를 반환한다. 공식은  $3.14 * \text{반지름} * \text{반지름}$ 이다. 객체를 만들고 `get_area` 메서드를 호출해 얻은 넓이를 화면에 출력해서 `get_area` 메서드를 테스트하라.



## 31.7 요약

» (Q31.2) Rectangle 클래스에 `get_area`와 `get_perimeter`라는 메서드를 추가하라. 객체를 만들고 각각의 메서드를 호출해 얻은 결과를 화면에 출력해서 두 메서드를 테스트하라.

- `get_area`는 직사각형의 넓이를 반환한다. 공식은 길이 \* 너비다.
- `get_perimeter`는 직사각형의 둘레를 반환한다. 공식은  $2 * (\text{길이} + \text{너비})$ 이다.