인공지능프로그래밍

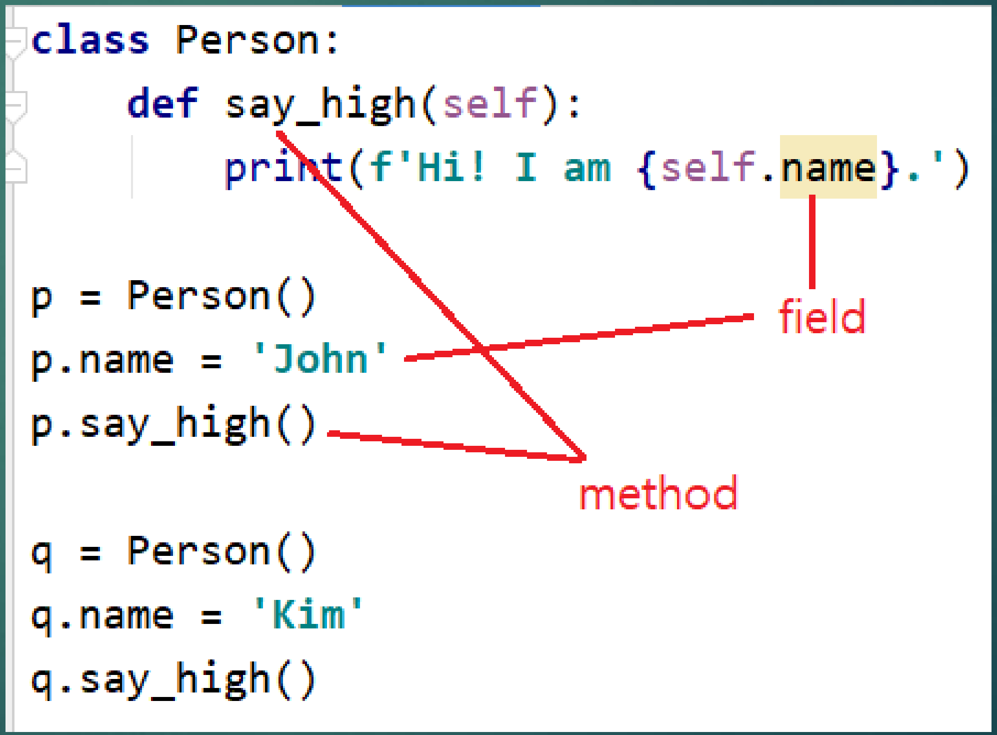
게임콘텐츠학과 박경수

https://github.com/ggorr/Machine-Learning/tree/master/Python

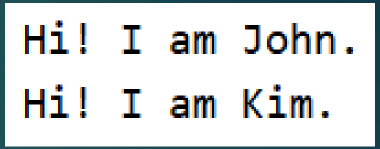
12. 객체 지향 프로그래밍

12. 객체 지향 프로그래밍

- 대상을 표현하는 방법
  - class – 설계도
  - field – 구성
  - method - 행동
- example:
  - Person이란 무엇인가?
  - field - name
  - method - say_high()



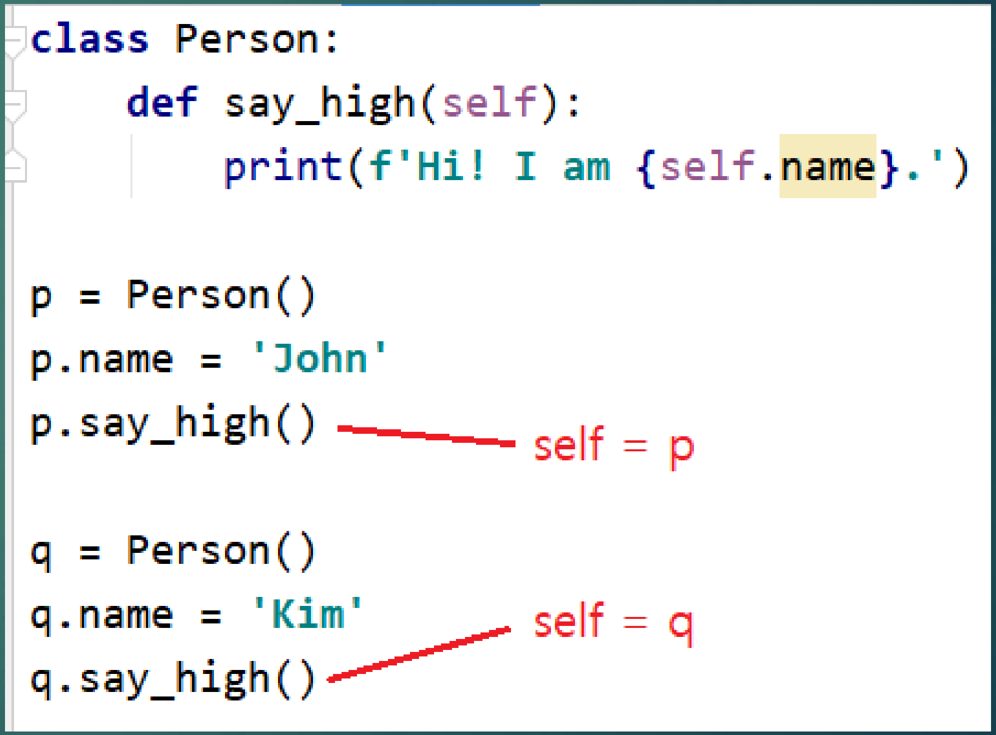
```
class Person:
    def say_high(self):
        print(f'Hi! I am {self.name}.')

p = Person()
p.name = 'John'
p.say_high()

q = Person()
q.name = 'Kim'
q.say_high()
```



```
Hi! I am John.
Hi! I am Kim.
```

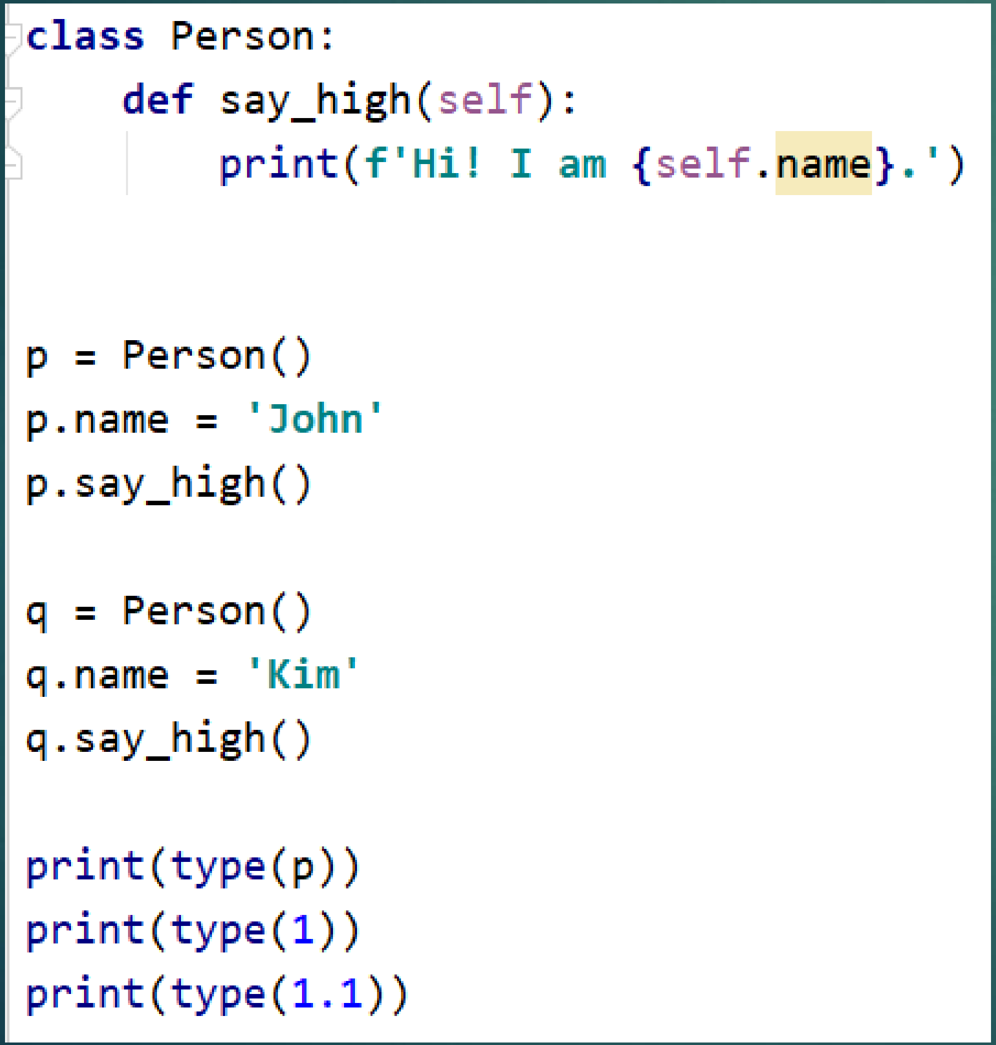
12.1. self 에 대하여

- self = 객체 자신
  - this for Java, C++
- method의 첫 parameter는 self

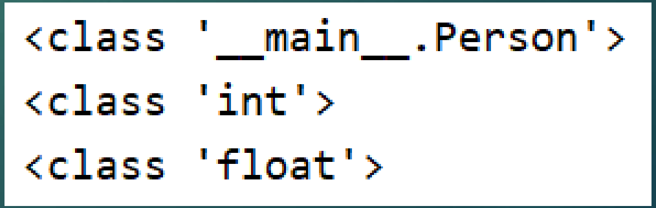


```
class Person:
    def say_high(self):
        print(f'Hi! I am {self.name}.')

p = Person()
p.name = 'John'
p.say_high()

q = Person()
q.name = 'Kim'
q.say_high()
```

12.2. 클래스



```
class Person:
    def say_high(self):
        print(f'Hi! I am {self.name}.')


p = Person()
p.name = 'John'
p.say_high()

q = Person()
q.name = 'Kim'
q.say_high()

print(type(p))
print(type(1))
print(type(1.1))
```



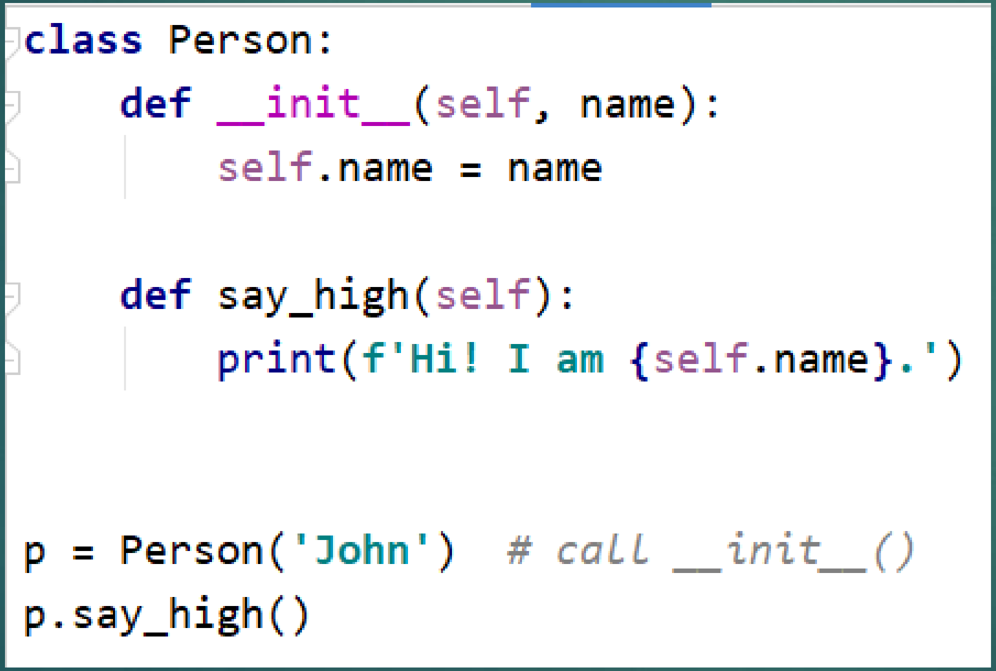
```
<class '__main__.Person'>
<class 'int'>
<class 'float'>
```

- 연습문제 12.1. 직사각형(rectangle)을 클래스로 표현하시오.
  field – width, height
  method – area()

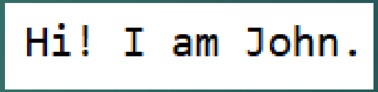
- 연습문제 12.2. String을 클래스로 표현하시오.
  field – content
  method – length(), last_char()

12.4. init 메소드

- Constructor



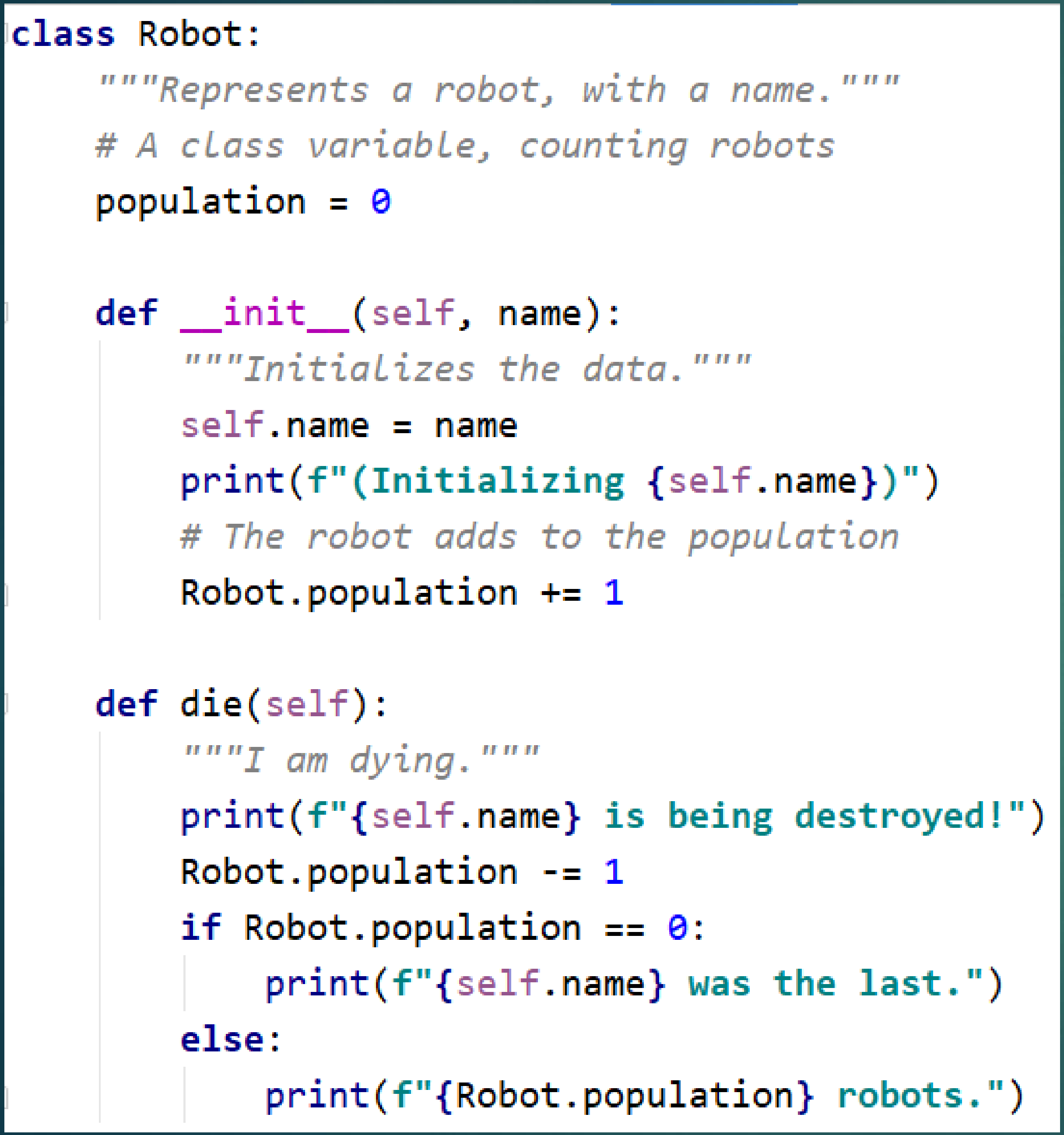
```python
class Person:
    def __init__(self, name):
        self.name = name

    def say_high(self):
        print(f'Hi! I am {self.name}.')


p = Person('John')  # call __init__()
p.say_high()
```



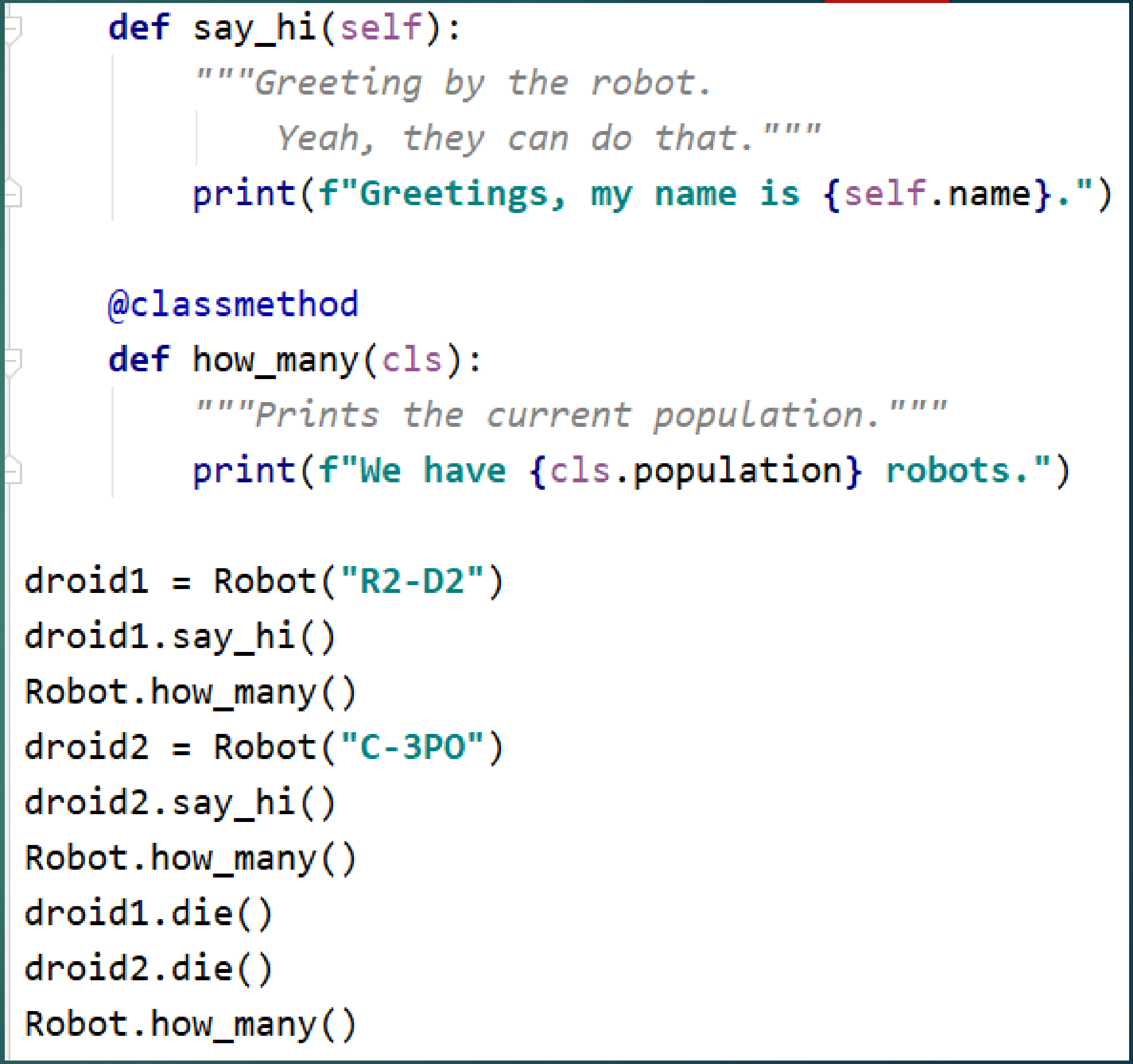
```
Hi! I am John.
```

12.5. 클래스 변수와 객체 변수

- 클래스 변수
  - 클래스에 속한 변수
  - 한 개만 존재
- 객체 변수
  - 객체에 속한 변수
  - 객체마다 존재



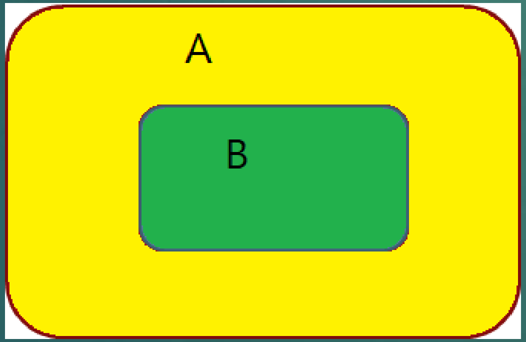
```python
class Robot:
    """Represents a robot, with a name."""
    # A class variable, counting robots
    population = 0

    def __init__(self, name):
        """Initializes the data."""
        self.name = name
        print(f"(Initializing {self.name})")
        # The robot adds to the population
        Robot.population += 1

    def die(self):
        """I am dying."""
        print(f"{self.name} is being destroyed!")
        Robot.population -= 1
        if Robot.population == 0:
            print(f"{self.name} was the last.")
        else:
            print(f"{Robot.population} robots.")
```

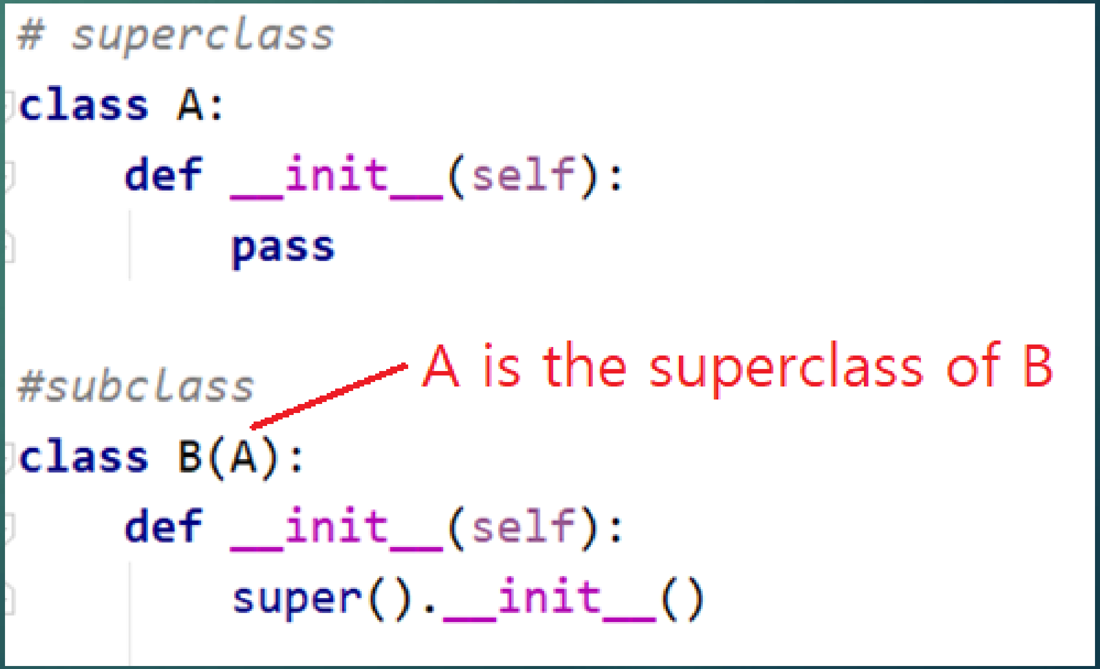


```python
    def say_hi(self):
        """Greeting by the robot.
           Yeah, they can do that."""
        print(f"Greetings, my name is {self.name}.")

    @classmethod
    def how_many(cls):
        """Prints the current population."""
        print(f"We have {cls.population} robots.")

droid1 = Robot("R2-D2")
droid1.say_hi()
Robot.how_many()
droid2 = Robot("C-3PO")
droid2.say_hi()
Robot.how_many()
droid1.die()
droid2.die()
Robot.how_many()
```

12.6. 상속

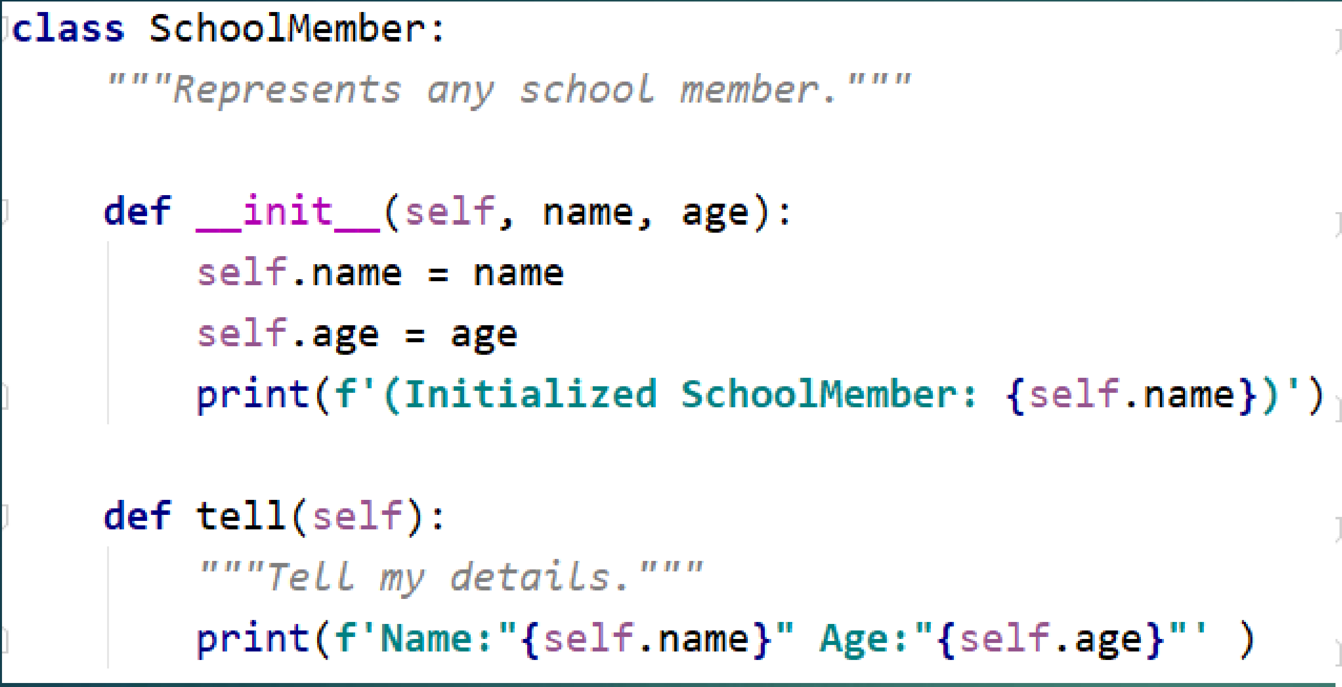
- superclass
  aka base class, parent class
- subclass
  aka derived class, child class



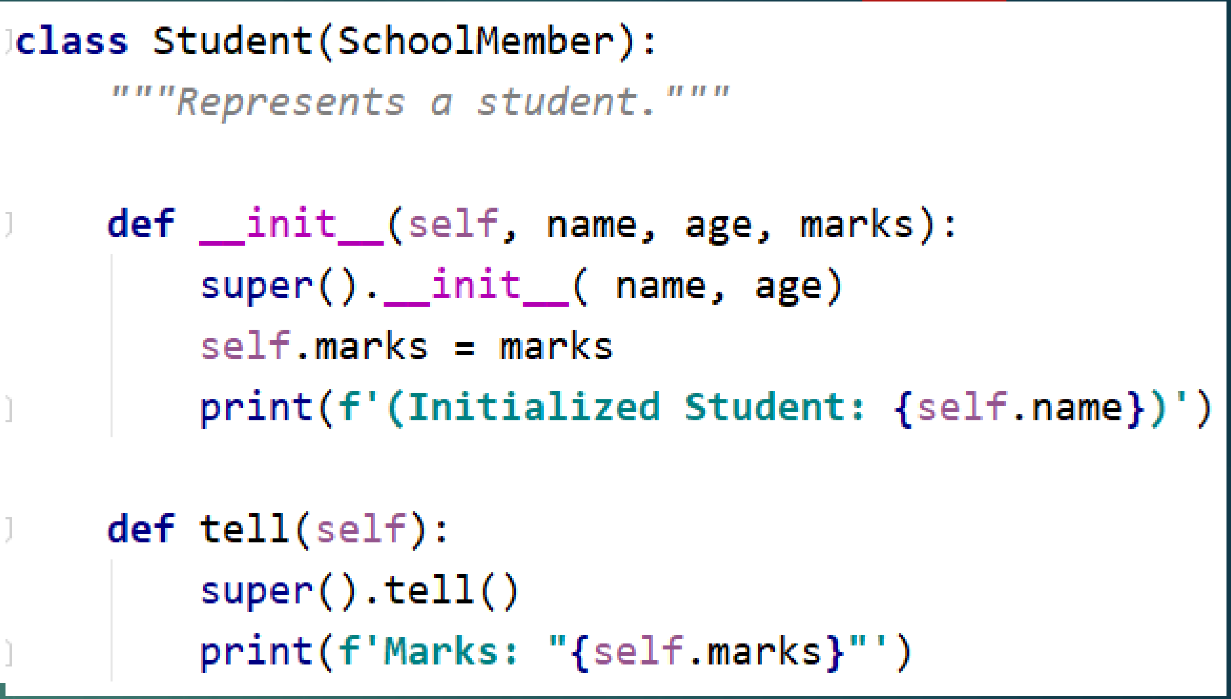


```
# superclass
class A:
    def __init__(self):
        pass

#subclass
class B(A):
    def __init__(self):
        super().__init__()
```

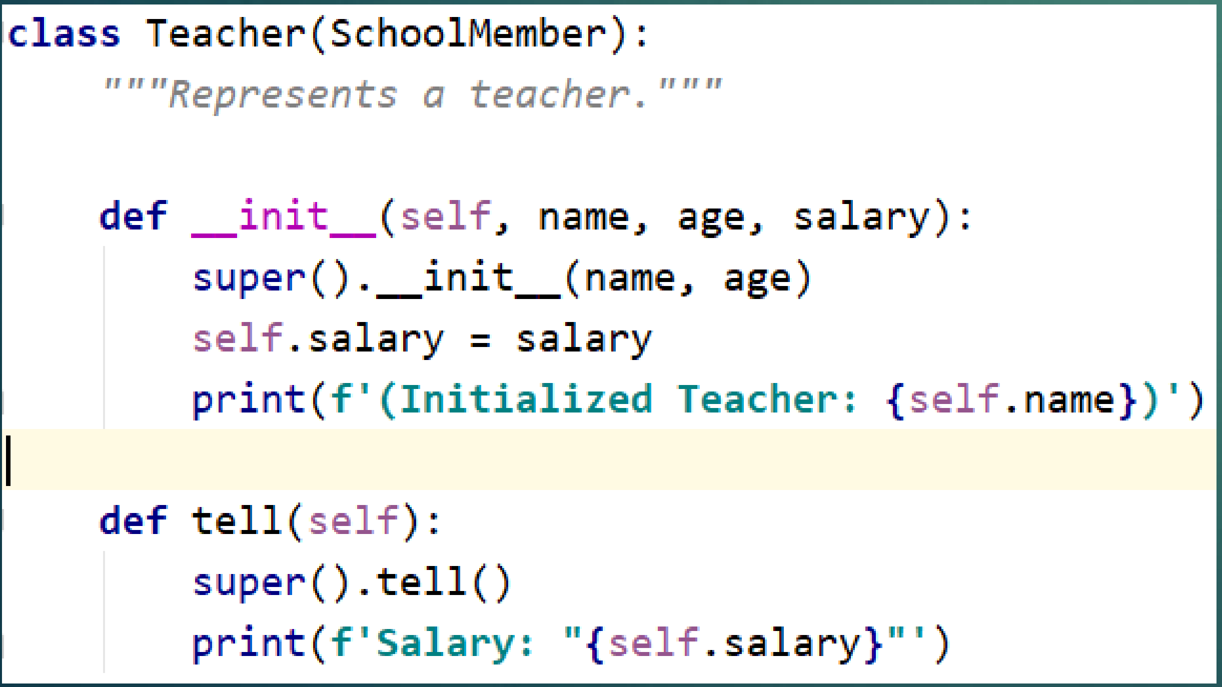
A is the superclass of B



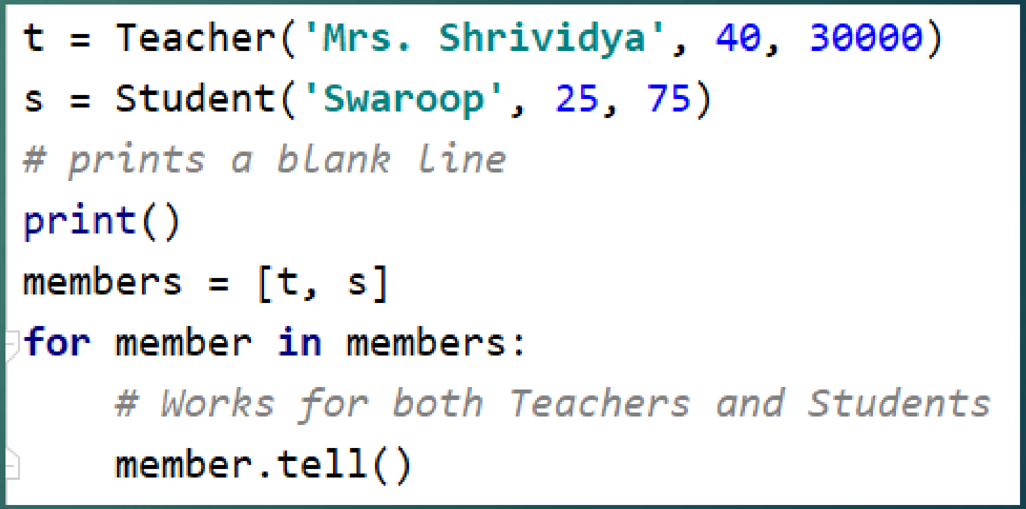
```python
class SchoolMember:
    """Represents any school member."""

    def __init__(self, name, age):
        self.name = name
        self.age = age
        print(f'(Initialized SchoolMember: {self.name})')

    def tell(self):
        """Tell my details."""
        print(f'Name:"{self.name}" Age:"{self.age}"' )
```



```python
class Student(SchoolMember):
    """Represents a student."""

    def __init__(self, name, age, marks):
        super().__init__( name, age)
        self.marks = marks
        print(f'(Initialized Student: {self.name})')

    def tell(self):
        super().tell()
        print(f'Marks: "{self.marks}"')
```



```python
class Teacher(SchoolMember):
    """Represents a teacher."""

    def __init__(self, name, age, salary):
        super().__init__(name, age)
        self.salary = salary
        print(f'(Initialized Teacher: {self.name})')

    def tell(self):
        super().tell()
        print(f'Salary: "{self.salary}"')
```



```python
t = Teacher('Mrs. Shrividya', 40, 30000)
s = Student('Swaroop', 25, 75)
# prints a blank line
print()
members = [t, s]
for member in members:
    # Works for both Teachers and Students
    member.tell()
```

- 연습문제 12.3. Vehicle, bicycle, car를 클래스로 표현하시오.
  superclass와 subclass를 구분하여 나타내시오.

- 연습문제 12.4. 주변에서 적당한 소재를 찾아 superclass와 subclass로 표현하시오