

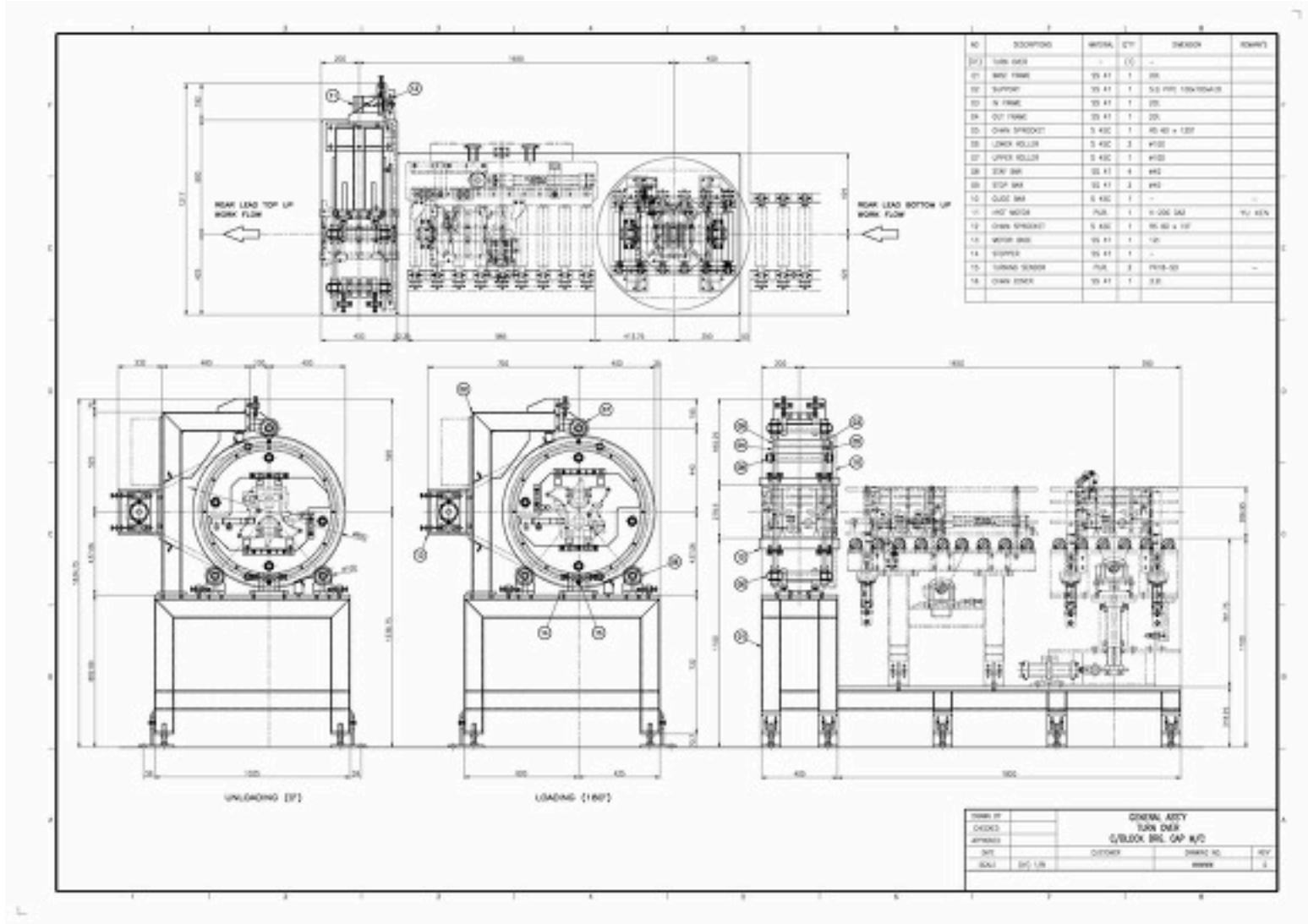
Object Oriented Programming

모든 것을 객체로 바라볼 것.

Object? 객체?

객체는 속성과 행동을 가진다.

파이썬에서 객체는 클래스로 표현.



클래스 == 설계도



순둥이

꿀 중독

포동포동



인기 많음

눈의 여왕

마법




```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(self, name, trait):
        self.name = name
        self.trait = trait

    def eat(self):
        if self.name == '곰돌이 푸':
            print('꿀만 먹어용 🍯')
        elif self.name == '엘사':
            print('엘사는 초콜릿! 🍫')

    def magic(self):
        if self.name == '엘사':
            print('얼음 마법 압압 🧊')
```

```
pooh = DisneyCharacter('곰돌이 푸', '순둥이')  
elsa = DisneyCharacter('엘사', '눈의 여왕')
```

```
pooh.name    # '곰돌이 푸'  
pooh.trait   # '순둥이'
```

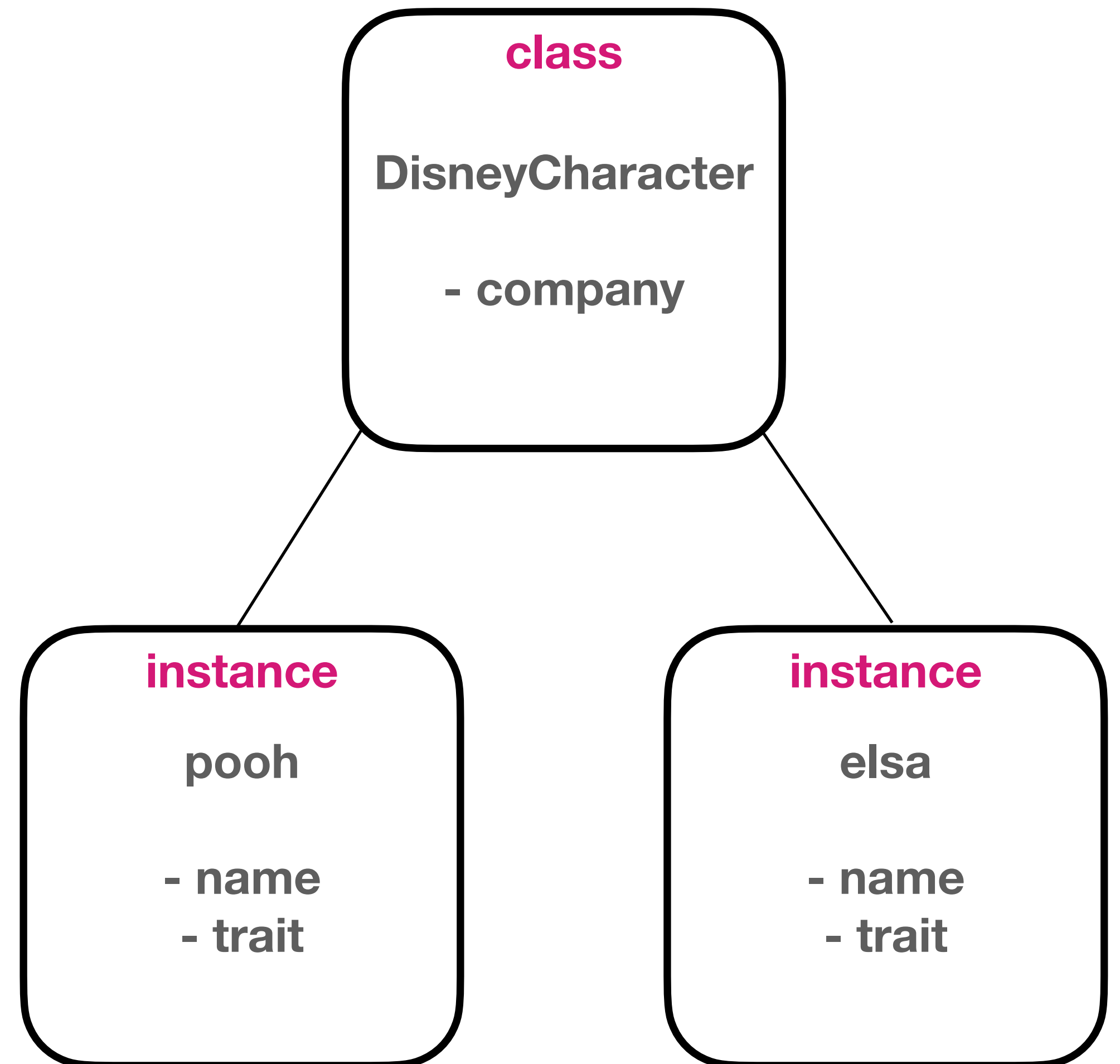
```
elsa.name    # '엘사'  
elsa.trait   # '눈의 여왕'
```

```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(self, name, trait):
        self.name = name
        self.trait = trait

    def eat(self):
        if self.name == '곰돌이 푸':
            print('꿀만 먹어용 🍯')
        elif self.name == '엘사':
            print('엘사는 초콜릿! 🍫')

    def magic(self):
        if self.name == '엘사':
            print('얼음 마법 압압 ❄️')
```



```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(self, name, trait):
        self.name = name
        self.trait = trait

    def eat(self):
        if self.name == '곰돌이 푸':
            print('곰돌이 푸는 꿀맛이야 🍯')
        elif self.name == '엘사':
            print('엘사는 초콜릿! 🍫')

    def magic(self):
        if self.name == '엘사':
            print('얼음 마법 압압 ❄️')
```

**Q. 만약에 곰돌이 푸의
회사 정보(company)가 궁금하다면?**

class

DisneyCharacter

instance

곰돌이 푸

name
trait

instance

엘사

name
trait

pooh.company

pooh.company

1. pooh 인스턴스에 company가 있는지 찾아본다.

3. 클래스에서 발견. 가져다 쓴다.

class

DisneyCharacter

- company

instance

pooh

- name

- trait

instance

elsa

name

trait

2. 없다. 그러면 클래스로!

이름을 찾는 순서!

인스턴스 => 클래스 => 상위 클래스

(상속)

```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(self, name, trait):
        self.name = name
        self.trait = trait

    def eat(self):
        if self.name == '곰돌이 푸':
            print('꿀을 먹어 🍯')
        elif self.name == '엘사':
            print('엘사는 초콜릿! 🍫')

    def magic(self):
        if self.name == '엘사':
            print('얼음 마법 압압 ❄️')
```

Q2. 만약 엘사에게 나이 정보(age)를 추가하고 싶다면?

class

DisneyCharacter

instance

곰돌이 푸

name
trait

instance

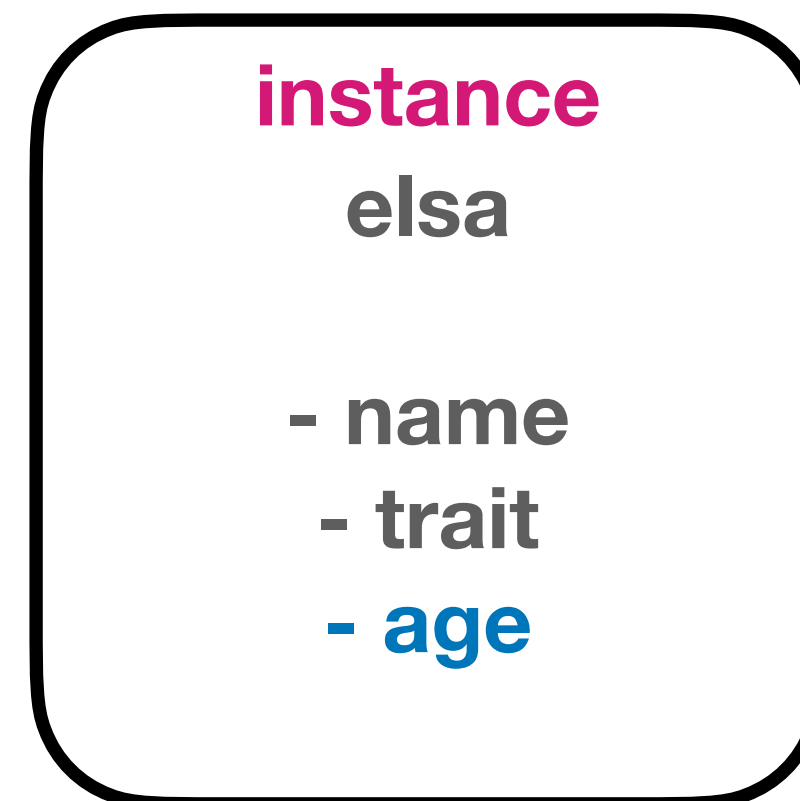
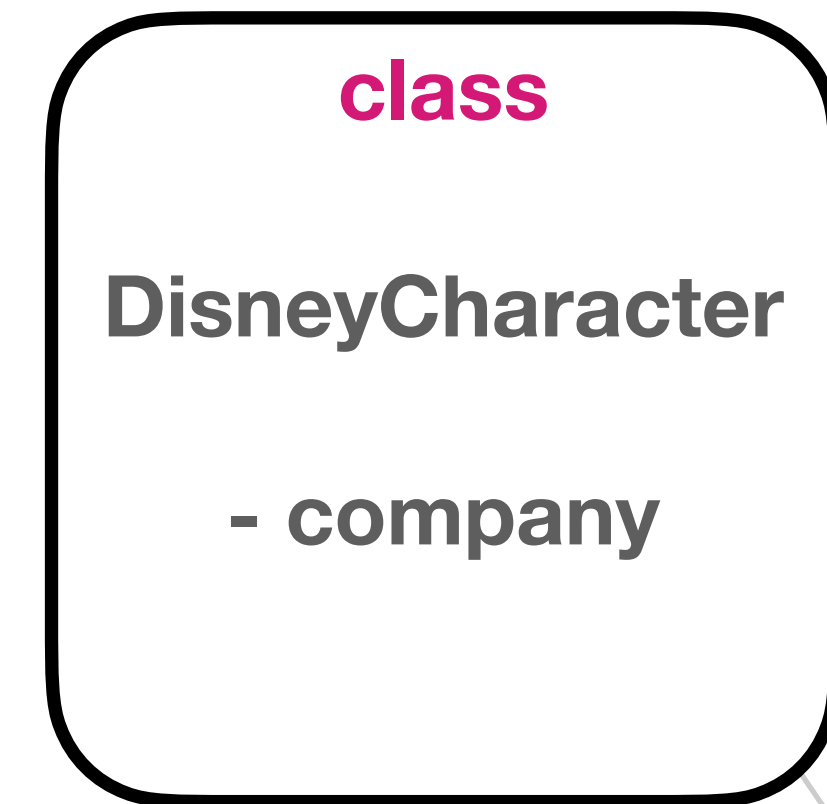
엘사

name
trait

elsa.age = 100

elsa.age = 100

1. elsa 인스턴스에 age가 이미 있든 없든 추가한다.



Quiz Time.

```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(first, name, trait):
        first.name = name
        first.trait = trait

pooh = DisneyCharacter('곰돌이 푸', '순둥이')
print(pooh.name)
```

print문의 결과로 옳은 것은?

1. '곰돌이 푸'
2. 에러
3. 순둥이


```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(first, name, trait):
        first.name = name
        first.trait = trait

    @classmethod
    def greeting(self):
        print(f'안녕하세요, {self.name}입니다. ')

pooh = DisneyCharacter('곰돌이 푸', '순둥이')
print(pooh.greeting())
```

print문의 결과는?

```
class DisneyCharacter:
    company = 'The Walt Disney Company'

    def __init__(self, name, trait):
        self.name = name
        self.trait = trait

pooh = DisneyCharacter('곰돌이 푸', '순둥이')
elsa = DisneyCharacter('엘사', '얼음 여왕')

pooh.company = '삼성전자'
print(pooh.company)
print(DisneyCharacter.company)
```

print문의 결과는?