

## Python의 Class와 상속(inheritance)의 개념

- Python 문법에서의 상속(inheritance)란, 부모 클래스(Super Class)의 속성(property)과 함수(method)를 그대로 물려 받는 개념입니다.
- Super Class의 내용을 자식 클래스(Sub Class)가 물려 받게 되면, Super Class의 속성과 함수를 자식 클래스에서 사용할 수 있습니다.
- class 에 속한 함수 (method)는 첫 번째 인자로 `self` 를 입력합니다. (사실 self가 아니어도 상관없지만, 일반적으로 self를 많이 사용합니다.)
- 자식클래스의 함수를 재정의하게 되면, **재정의된 함수로 실행**되게 됩니다.

```
class Person:
    """Super Class"""
    # 클래스 변수
    total_count = 0

    # 생성자 메서드(method)
    def __init__(self):
        self.name = '홍길동'
        self.age = 1
        Person.total_count+=1

    # class내 정의된 메서드(method)
    def introduce(self):
        print(f'제 이름은 {self.name} 이고, 나이는 {self.age}살 입니다.')
```

# 클래스 상속 (inheritance) 받기

```
class Student(Person):  
    """Sub Class"""  
  
    def __init__(self):  
        super().__init__()  
  
    def print_name(self):  
        print(f'제 이름은 {self.name} 입니다.')  
  
    def print_age(self):  
        print(f'제 나이는 {self.age} 입니다.')
```

```
s1 = Student()
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-9-906bfe3cbabc> in <module>()  
----> 1 s1 = Student()  
  
<ipython-input-8-ed9ff6a99777> in __init__(self)  
      3  
      4     def __init__(self):  
----> 5         super().__init__()  
      6  
      7     def print_name(self):  
  
TypeError: __init__() missing 2 required positional arguments: 'name' and 'age'
```

오류를 어떻게 해결할 수 있을까?

## Sub Class 에서의 오류

Student 클래스는 Person 클래스를 상속받아 구현하게 되면서, 생성자 (**init**) 함수에서 **name**, **age** argument를 요구합니다. 이 Rule을 지켜주지 못하면 에러가 발생합니다.

## 첫 번째 해결책

super().\_\_init\_\_() 호출시 name과 age argument를 넘겨줍니다.

```
class Student(Person):  
    """Sub Class"""  
  
    def __init__(self):  
        super().__init__('테디', 30)
```

```
student = Student()
```



## 두 번째 해결책

Student 클래스의 `__init__()` 을 `.__init__(name, age)` 인자를 받는 생성자 메서드로 **재정의**할 수 있습니다.

```
class Student(Person):  
    """Sub Class"""  
  
    def __init__(self, name, age):  
        super().__init__(name, age)
```

```
student = Student('테디', 30)  
student.introduce()
```

제 이름은 테디 이고, 나이는 30살 입니다.