

인공지능을 위한 파이썬 코딩 기초



파이썬 함수, 클래스

김루진 강사

소속

인공지능을 위한 파이썬 코딩 기초

학습 내용

- 1 파이썬이란
- 2 기본구조(시작은 MAIN에서)
- 3 입출력
- 4 산술 연산자



함수

+++

요구 사항에 따라 명시적으로
프로그래머가 만든 함수를 사용자 정의 함수

» 함수 이름과 함께 키워드를 사용 def하여 함수를 정의

```
def function_name(parameter1, parameter2):
```

```
    # function body
```

```
    # write some action
```

```
return value
```

Python Functions

In Python, the **function** is a block of code defined with a name

- A Function is a block of code that only runs when it is called.
- You can pass data, known as parameters, into a function.
- Functions are used to perform specific actions, and they are also known as methods.
- **Why use Functions?** To reuse code: define the code once and use it many times.

```
def add(num1, num2):  
    print("Number 1:", num1)  
    print("Number 2:", num1)  
    addition = num1 + num2  
  
    return addition  
  
res = add(2, 4)  
print(res)
```

Function Name Parameters

Function Body

Return Value

Function call

PYnative

+++



클래스 이해

+++

정보를 집약해서 만든 추상화 구조체
(정보들을 담는 그릇 정의)

» 사람을 클래스 만들어 main함수에서 사용

```
class Person: # 클래스
    def __init__(self, name, age, address):
        self.name = name
        self.age = age
        self.address = address

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

▶ person.py

+++

+++



클래스 이해

+++

정보를 집약해서 만든 추상화 구조체
(정보들을 담는 그릇 정의)

» 사람을 클래스 만들어 main함수에서 사용

```
import person
```

```
p1 = person.Person("홍길동",22,"충남")
```

```
p1.greeting()
```

▶ main.py

+++

+++

◆ 클래스 이해

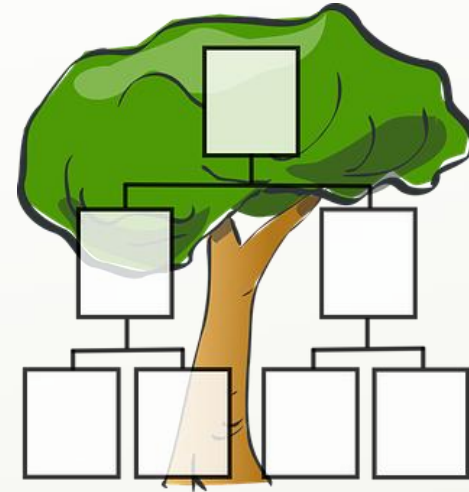
⚙ 객체지향 프로그래밍의 특징

» 추상화(Abstraction)

- ▶ 실제 세상을 프로그래밍으로 옮길 수 있게 필요한 기능들만 가져와서 단순화

» 캡슐화(Encapsulation)

- ▶ 보안적인 기능에 도움을 줌
- ▶ 규칙 정의, 사용 용도 명확히
- ▶ 안에 있는 값을 노출하지 않음





◆ 클래스 이해

+++

⚙ 객체지향 프로그래밍의 특징

» 상속성(Inheritance)

- ▶ 부모를 지정하고, 부모의 코드를 가져다가 쓸 수 있는 것
- ▶ 코드의 재사용성

» 다형성(Polymorphism)

- ▶ 상수, 변수, 식, 객체, 메소드 등이 다양한 자료형(type)에 속하는 것이 허가

+++

» 오버라이드(Override)

- ▶ 기존의 클래스에 새로운 기능을 추가하거나 변경

+++





클래스 이해

+++

인스턴스 생성하기 (Instantiation)

```
class robot:  
    name = "robot"  
    def info(self):  
        print('나의 이름은', self.name, '입니다!')  
  
robot_instance = robot()
```

+++

+++



클래스 이해

+++

⚙️ 인스턴스 생성하기 (Instantiation)

» 파이썬 생성자, 소멸자

```
class robot:  
    name = "robot"  
    age = 0  
    def __init__(self, name, age):  
        print('생성자 호출!')  
        self.name = name  
        self.age = age
```

⋮

+++

+++



클래스 이해

+++

⚙ 인스턴스 생성하기 (Instantiation)

» 파이썬 생성자, 소멸자

⋮

```
def __del__(self):  
    print('소멸자 호출!')  
  
def info(self):  
    print('나의 이름은', self.name, '입니다!')  
    print('나이는', self.age, '입니다!')
```

+++

+++



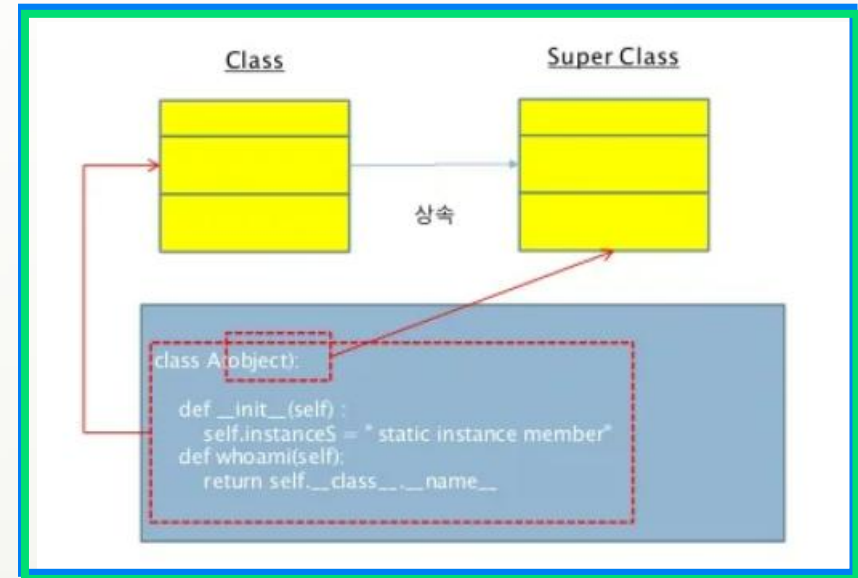
클래스 이해

+++

파이썬 클래스 상속

```
class robot:  
    name = 'robot'  
    age = 0  
    def __init__(self, name, age):  
        print('robot 생성자 호출!')  
        self.name = name  
        self.age = age
```

⋮



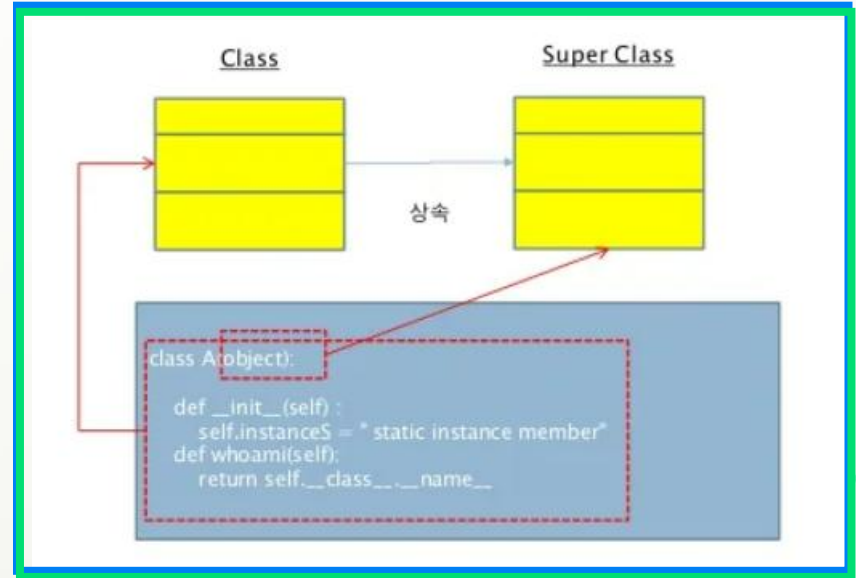
++

+++

클래스 이해

파이썬 클래스 상속

```
def __del__(self):  
    print('robot 소멸자 호출!')  
  
def info(self):  
    print('나의 이름은', self.name, '입니다!')  
    print('나이는', self.age, '입니다!')
```



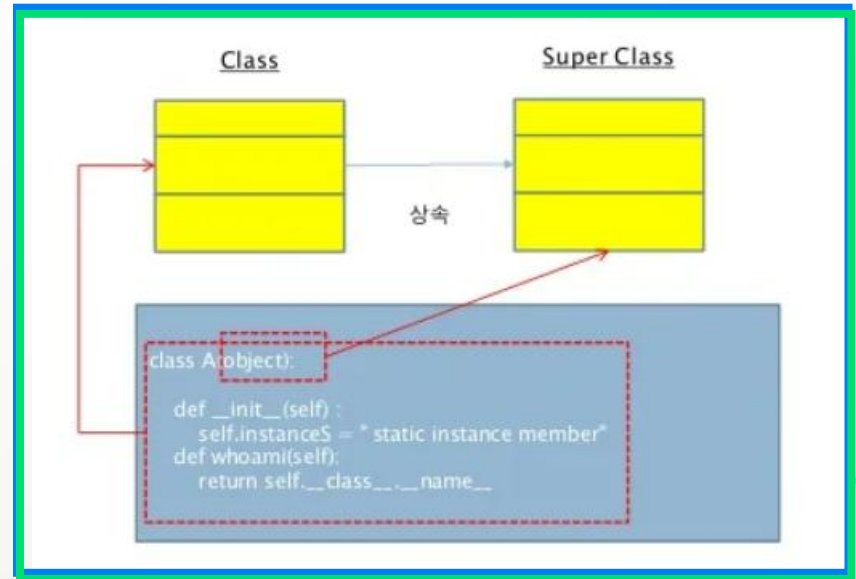


클래스 이해

+++

파이썬 클래스 상속

```
class strong_robot(robot):  
    weapon = 'gun'  
    def __init__(self, name, age, weapon):  
        print('strong_robot 생성자 호출!')  
        super().__init__(name, age)  
        self.weapon = weapon  
    def info(self): #오버라이딩  
        super().info()  
        print(self.weapon, '로 싸웁니다!')
```



++

+++



클래스 이해

+++

파이썬 get, set 메소드 함수 사용

```
class test:  
    __data = 10  
    def getData(self):  
        return self.__data  
    def setData(self, data):  
        self.__data = data
```

+++

+++

클래스 이해

파이썬 클래스 메소드 VS 인스턴스 메소드

```
class test:
    data = 10 # 클래스 속성
    def __init__(self, data):
        self.data = data # 인스턴스 속성
    @classmethod
    def printClass(cls): # 클래스 메소드
        print(cls.data)

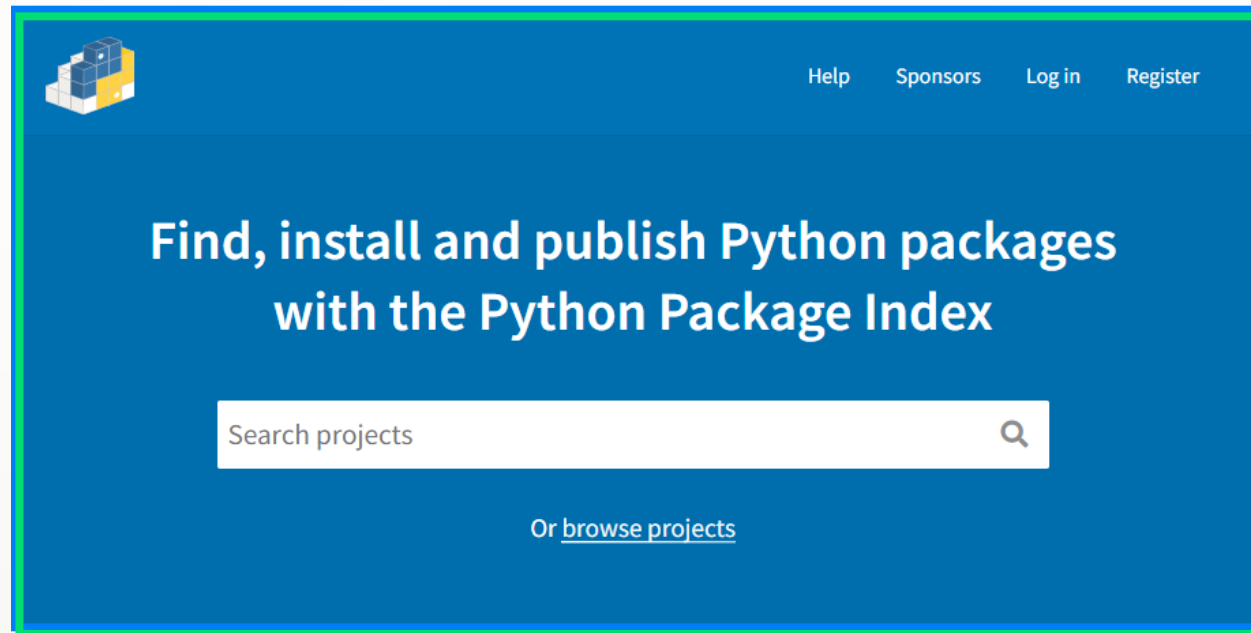
    def printInstance(self): # 인스턴스 메소드
        print(self.data)
```

- ▶ 메소드는 기본적으로 self를 전달하는 인스턴스 메소드
- ▶ @classmethod, @staticmethod를 사용하여 클래스 메소드를 만듦
- ▶ 클래스 메소드는 클래스 이름으로 함수 호출 가능
예 test.printClass()
- ▶ 클래스 속성, 클래스 메소드는 모든 객체(인스턴스)가 공유



PyPI(Python Package Index)

Python 패키지 색인을 사용하여
Python 패키지 찾기,
설치 및 게시



» <https://pypi.org/>

» 패키지 설치하기: `pip install pandas`



PyPI(Python Package Index)



+++

+++



감사합니다.