

데이터처리프로그래밍

Object Oriented programming



강원대학교 교육혁신원 송혜정

<hjsong@kangwon.ac.kr>



Object Oriented Programming

✓ 학습목표

- Object Oriented Programming을 이해한다.

✓ 학습내용

- 객체지향프로그래밍
- Class 선언
- 객체 초기화 함수 (__init__())
- 객체(Instance) 생성
- Car class 예
- 객체 문자열 함수 (__str__())
- Class 변수 & 객체(instance)변수
- NoteBook 프로그램



강의에 앞서서..

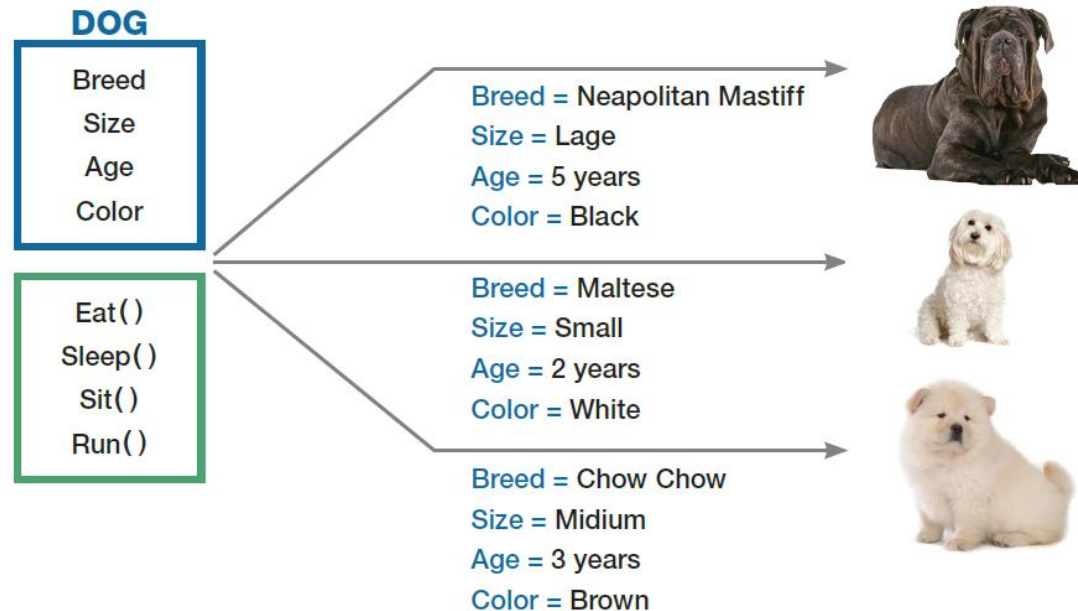
- 본 강의자료는 아래의 자료들을 참고하여 만들어 졌음을 알립니다
 1. 데이터과학을 위한 파이썬 프로그래밍, 최성철, 한빛아카데미,2019
 2. Python (<https://docs.python.org>)
 3. 실용 파이썬 프로그래밍(Practical Python Programming)
(<https://wikidocs.net/84360>)
 4. 점프 투 파이썬 (<https://wikidocs.net/book/1>)

객체지향프로그래밍

- 객체 지향 프로그래밍(Object Oriented Programming, OOP)
 - 프로그램의 단위를 객체로 정의하여 처리하는 방식
 - 함수(method)와 데이터(variable)를 묶어서 객체를 정의
 - 객체사이의 메시지 전달로 문제를 해결하는 프로그래밍 기법
- 객체(Object)란?
 - 현실 세계의 물리적으로 존재하는 사물(자동차, 책, 사람)이나 추상적인 개념(회사, 날짜)을 프로그램의 구성요소로 표현한 소프트웨어 단위
- 객체는 속성(필드)과 행위(메소드)로 구성.
 - 속성, 필드(field, attributes, properties, state, data, variable)
 - 객체를 구별시키는 상태 값을 나타내는 데이터
 - 행위, 메서드(behaviors, messages, methods, function)
 - 객체가 할 수 있는 기능
 - 속성값을 변경하거나 조작하는 기능

객체지향프로그래밍

- DOG 객체 예
 - 속성
 - 품종(Breed), 크기(Size), 나이(Age), 색(Color)
 - 행위
 - 먹다(Et()), 잠자다(Sleep()), 앉다(Sit()), 뛰다(Run())



Class

- 객체의 타입을 정의
- 클래스 정의 형식

class SoccerPlayer(object):

↑ ↑ ↑

클래스 예약어 클래스 이름 상속받는 객체명

- 객체 속성 선언
 - `__init__()` 함수로 속성 변수를 초기화
 - `self` 뒤의 매개변수로 객체의 속성 초기값 할당
 - `self`: 클래스로부터 생성된 객체 인스턴스(instance)

```
class SoccerPlayer(object):
    def __init__(self, name, position, back_number):
        self.name = name
        self.position = position
        self.back_number = back_number
```



Class

- 객체 함수 선언
 - 객체의 다양한 동작을 정의
 - 객체 인스턴스 함수 정의를 위해 매개변수로 self 를 포함
 - self로 속성을 접근하여 처리

```
class SoccerPlayer(object):  
    def change_back_number(self, new_number):  
        print("선수의 등번호를 변경한다: From %d to %d" % (self.back_number, new_number))  
        self.back_number = new_number
```

Class

- SoccerPlayer class 정의 예

```
#SoccerPlayer class 선언
class SoccerPlayer(object):
    #속성 초기화 함수
    def __init__(self, name, position, back_number):
        self.name = name
        self.position = position
        self.back_number = back_number
    #함수 선언
    def change_back_number(self, new_number):
        print("선수의 등번호를 변경한다: From %d to %d" % (self.back_number, new_number))
        self.back_number = new_number
```


Class

- 객체 인스턴스 생성

- 클래스명으로 객체 생성
- 매개변수로 `_init_()` 함수의 매개변수를 설정

jinyun = SoccerPlayer("Jinyun", "MF", 10):

객체명

클래스 이름

`_init_` 함수 Interface, 초깃값

`def _init_(self, name, position, back_number);`

- 객체 사용

- 객체명.필드명, 객체명.함수명()으로 접근하여 사용

```
13 # SoccerPlayer를 사용하는 instance 코드
14 jinyun = SoccerPlayer("Jinyun", "MF", 10)
15
16 print("현재 선수의 등번호는:", jinyun.back_number)
17 jinyun.change_back_number(5)
18 print("현재 선수의 등번호는:", jinyun.back_number)
```

Class

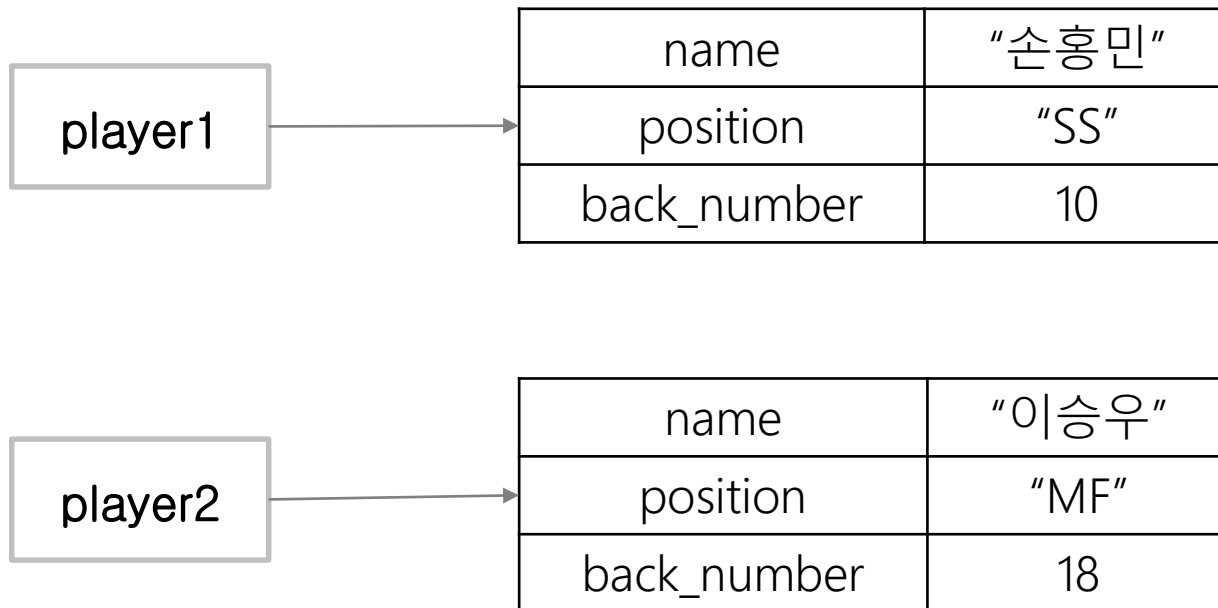
- SoccerPlayer 객체 사용 예

```
# SoccerPlayer 클래스의 속성 초기화 함수로 객체 생성 및 속성 초기값 설정
player1 = SoccerPlayer("손흥민", "SS", 7)
print("{0} 선수의 등번호는: {1}".format(player1.name, player1.back_number))
player2 = SoccerPlayer("이승우", "MF", 10)
print("{0} 선수의 등번호는: {1}".format(player2.name, player2.back_number))
#객체(instance) 함수호출
player2.change_back_number(18)
print("{0} 선수의 등번호는: {1}".format(player2.name, player2.back_number))
```

손흥민 선수의 등번호는: 7
 이승우 선수의 등번호는: 10
 선수의 등번호를 변경한다: From 10 to 18
 이승우 선수의 등번호는: 18

Class

- SoccerPlayer 클래스의 객체(instance)



Class

- Car class 예
 - 자동차 클래스 선언
 - 속성: 이름(name), 색상(color), 속도(speed)
 - 함수 : 속성 초기화 (__init__()), 속성값 출력(printCar()), 속도올리기(upSpeed()), 속도내리기(downSpeed())

```
#Car class 선언
class Car(object):
    #속성 초기화 함수
    def __init__(self, name, color, speed):
        self.name = name
        self.color = color
        self.speed = speed
    #함수 선언
    def printCar(self):
        print ("Car name : ", self.name)
        print ("Car color : ", self.color)
        print ("Car speed : ", self.speed)
    def upSpeed(self, value):
        self.speed += value
    def downSpeed(self, value):
        self.speed -= value
```

Class

- Car class 예
 - 객체 생성 및 속성확인
 - 속도 올리기, 속도 내리기 함수 호출로 처리 결과 확인

```
#Car 객체 생성
car1 = Car("소나타", "빨강", 0)
car2 = Car("티코", "노랑", 0)

#Car 객체의 속성확인, printCar() 함수 호출
print(car1.name)
car1.printCar()
car2.printCar()

#Car 속도 변경
car1.upSpeed(50)
print("car1.upSpeed(50) ==> ")
car1.printCar()
print("car1.downSpeed(20) ==> ")
car1.downSpeed(20)
car1.printCar()
```

```
소나타
Car name : 소나타
Car color : 빨강
Car speed : 0
티코
Car name : 티코
Car color : 노랑
Car speed : 0
car1.upSpeed(50) ==>
Car name : 소나타
Car color : 빨강
Car speed : 50
car1.downSpeed(20) ==>
Car name : 소나타
Car color : 빨강
Car speed : 30
```