

## 공학적 문제해결 기법

11주차 1차시

# 객체와 클래스의 유용한 기능들





# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## I. 개요

상속, self 매개변수, 객체의 초기화

## II. 학습 개요

### 1) 학습 목표

부모의 함수를 자식 클래스가 어떻게 상속받는지, 같은 클래스로 생성된 두 개의 객체가 반드시 동일하지 않을 수 있음을 배운다. 또한 다른 함수와 변수를 참조하기 위해 함수에 있는 self 매개변수의 이용법을 알아본다.

### 2) 학습 목차(세부 목차)

- 상속된 함수
- 다른 함수들을 호출하는 함수
- 객체의 초기화



# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 1 - 상속된 함수

### 상속

- ✅ 클래스와 객체는 함수들을 쉽게 그룹화해준다.
- ✅ 프로그램을 작은 분량으로 나눠서 생각하고자 할 때 유용
- ✅ 대규모의 프로그램을 개발할 경우에는 다른 프로그래머들과 함께 작업할 수 있도록 작은 조각으로 나눈다.
- ✅ 윈도우와 같은 복잡한 프로그램들은 많은 전 세계에 있는 개발팀이 동시에 서로 다른 부분을 함께 개발한 것
- ✅ 앞서 생성했던 클래스 (Animals, Mammals, Giraffes)를 확장해 봅니다.
- ✅ 만약 해야 할 일이 너무 많을 경우 코드를 작성하는 작업을 나눠서
- ✅ 한 사람은 Animals 클래스, 또 한 사람은 Mammals 클래스를 그리고 또 다른 사람은 Giraffes 클래스를 작업하도록 분업한다.



# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 1 - 상속된 함수

### 상속

- ✓ Giraffes 클래스를 가지고 작업하는 사람들은 운이 좋다.
- ✓ Giraffes 클래스에서 사용될 수 있는 함수들을
- ✓ Animals 클래스와 Mammals 클래스로 작업하는 사람들이 생성해주기 때문
- ✓ Giraffes 클래스는 Mammals 클래스로부터 상속받음
- ✓ Mammals 클래스는 Animals 클래스로부터 상속받음
- ✓ 즉, 기린 객체를 생성하면 Giraffes 클래스에 정의된 함수들을 사용할 수 있으며
- ✓ Mammals 클래스와 Animals 클래스에 있는 함수들도 사용할 수 있다.
- ✓ 즉, Animals 클래스는 Mammals 클래스의 부모이며
- ✓ Mammals 클래스는 Giraffes 클래스의 부모이다.

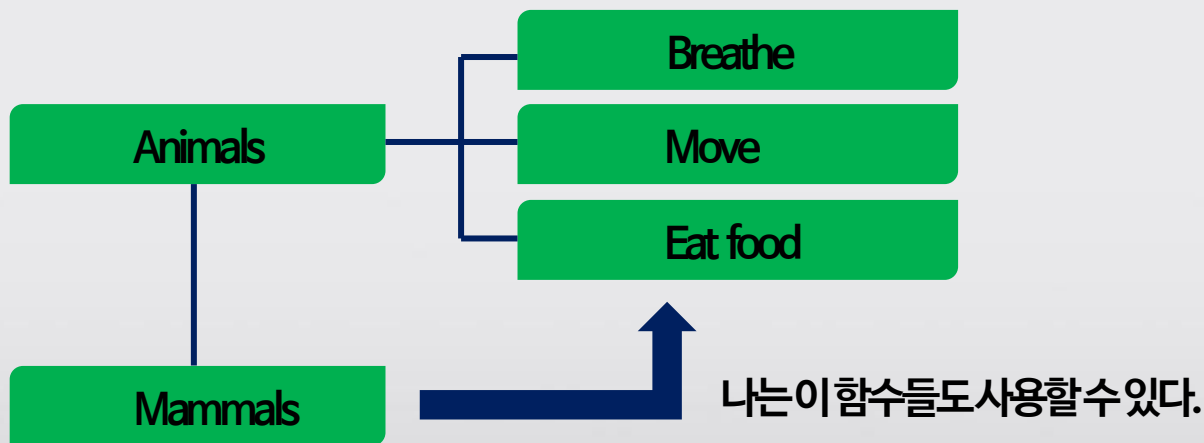


# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 1 - 상속된 함수

### 상속

☒ Animals와 Mammals, 그리고 Giraffes 클래스 사이의 관계





# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 1 - 상속된 함수

### 상속이 중요한 이유

- ✓ 상속은 클래스의 중요한 특징
- ✓ 재사용이 가능
- ✓ 상속받은 클래스는 상속해준 클래스의 모든 기능을 사용할 수 있다.
- ✓ 추가로 필요한 기능만을 정의하거나
- ✓ 기존의 기능을 변경해서 새로운 클래스를 만들 수 있다.
- ✓ 앞의 예에서 reginald는 Giraffes 클래스의 객체
- ✓ 부모 클래스에 정의된 함수들은 자식 클래스에서 사용할 수 있기 때문에
- ✓ Animals 클래스에 정의된 move 함수 호출 가능



```
>>> reginald = Giraffes()
```

```
>>> reginald.move()
```

```
moving
```





# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 1 - 상속된 함수

### 상속이 중요한 이유

- ☒ Animals 클래스와 Mammals 클래스에 정의된 모든 함수들은 자식 클래스에 되므로 reginald 객체에서 호출될 수 있다.

➡  
>>> reginald = Giraffes()  
>>> reginald.breathe()  
breathing  
>>> reginald.eat\_food()  
eating food  
>>> reginald.feed\_young\_with\_milk()  
feeding young



# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 2 - 다른 함수들을 호출하는 함수

### self 매개변수

☒ 객체의 함수를 호출할 때는 객체의 변수명을 이용

☒ 기린인 reginald의 move 함수를 호출할 때

➡ >>> reginald.move()

☒ Giraffes 클래스에서 move 함수를 호출하려면 self 매개변수를 이용

☒ self 매개변수: 클래스에 있는 함수가 그 클래스에 있는 다른 함수를 호출하는 방법

☒ Giraffes 클래스에 find\_food라는 함수를 다음과 같이 추가해 보자.

➡ >>> class Giraffes(Mammals) :  
    def find\_food(self) :  
        self.move()  
        print("I've found food!")  
        self.eat\_food()





# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 2 - 다른 함수들을 호출하는 함수

### self 매개변수

- ☒ 이 코드에서 두 개의 다른 함수들을 결합한 함수를 생성했다.
- ☒ self를 이용하여 Giraffes 클래스에 몇몇 함수들을 추가해 보자.

➡ 

```
>>> class Giraffes(Mammals):  
    def find_food(self):  
        self.move()  
        print("I've found food!")  
        self.eat_food()  
    def eat_leaves_from_trees(self):  
        self.eat_food()  
    def dance_a_jig(self):  
        self.move()  
        self.move()  
        self.move()
```



# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 2 - 다른 함수들을 호출하는 함수

### self 매개변수

- ✓ Giraffes 클래스의 eat\_leaves\_from\_trees 함수와 dance\_a\_jig 함수를 정의
- ✓ 부모인 Animals 클래스의 eat\_food 함수와 move 함수를 이용
- ✓ 이렇게 할 수 있는 이유는 그것들이 상속된 함수들이기 때문
- ✓ 다른 함수들을 호출하는 함수를 추가하면 클래스의 객체를 생성하여 단 하나의 함수만 호출해도 여러 작업을 할 수 있게 됨

➡ >>> reginald.Giraffes()  
>>> reginald.dance\_a\_jig()  
moving  
moving  
moving



# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 3 - 객체의 초기화

### **init 함수**

- ✓ 객체를 생성할 때마다 나중에 사용하기 위한 몇몇 값들(속성(property))이라고 함
- ✓ 객체 초기화(initialize): 초기값 설정
- ✓ 기린 객체를 생성할 때(초기화할 때) 기린의 반점 개수를 설정하고 싶다면?
- ✓ `_init_` 함수를 만들어야 함
- ✓ 파이썬 클래스에 있는 특별한 함수이며 반드시 `__init__` 이 이름이어야 함
- ✓ 즉 `init` 함수는 객체가 처음 생성될 때 객체의 속성들을 설정하는 방법
- ✓ `init` 함수 역시 첫 번째 매개변수로 `self` 를 받아야 함
- ✓ 파이썬은 새로운 객체가 생성될 때 자동으로 이 함수를 호출



```
>>> class Giraffes :
```

```
    def _init_(self, spots) :
```

```
        self.giraffe_spots = spots
```



# 11주차 1차시 : 객체와 클래스의 유용한 기능들

## III. 학습 3 - 객체의 초기화

### **init 함수**

- ✓ self 매개변수를 이용하여 다른 함수를 호출할 수 있는 클래스 함수처럼, 클래스에 있는 변수들도 self를 이용하여 접근할 수 있다.



```
>>> ozwald = Giraffes(100)
>>> gertrude = Giraffes(150)
>>> print(ozwald.giraffe_spots)
100
>>> print(gertrude.giraffe_spots)
150
```

- ✓ 매개변수 값이 100으로 Giraffes 클래스의 인스턴스를 생성하면서, spots 매개변수의 값으로 100을 넣어 \_init\_ 함수를 호출
- ✓ ozwald처럼 클래스의 객체(인스턴스)를 생성하면 이용하고자 하는 변수 또는 함수의 이름과 도트(dot) 연산자를 이용하여 그 객체의 변수와 함수를 참조



『 이 콘텐츠는 2014학년도 학부교육 선도대학 육성사업(ACE)에 의하여 개발한 것임 』

