

# Python 문법 with Pygame

정규 수업 4차시



다시 되새기며..



```
def 함수_이름():
```

```
    함수가 실행할 코드
```

```
    return 반환값
```

다시 되새기며..

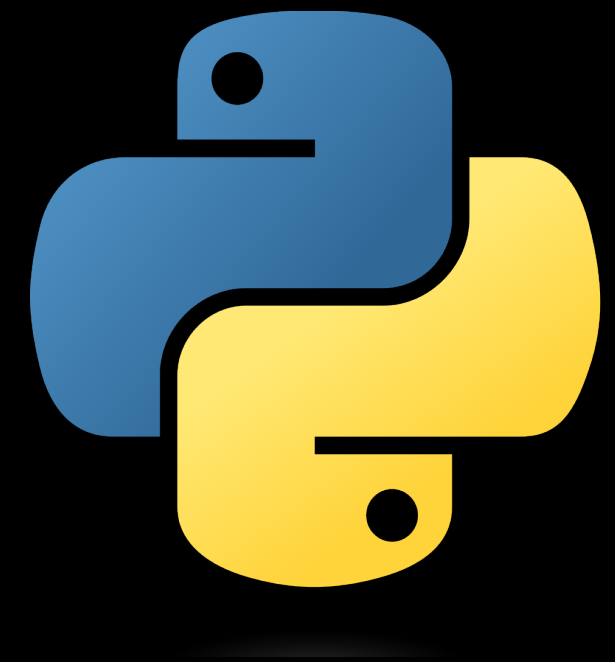


```
def 더하기(더하는_수1, 더하는_수2):  
    더하는_수1+=1  
    return 더하는_수 +더하는_수2  
print(더하기(10, 11))
```

들어가며

# 본격적인 코딩의 시작 OOP

정규 수업 4차시



OOP

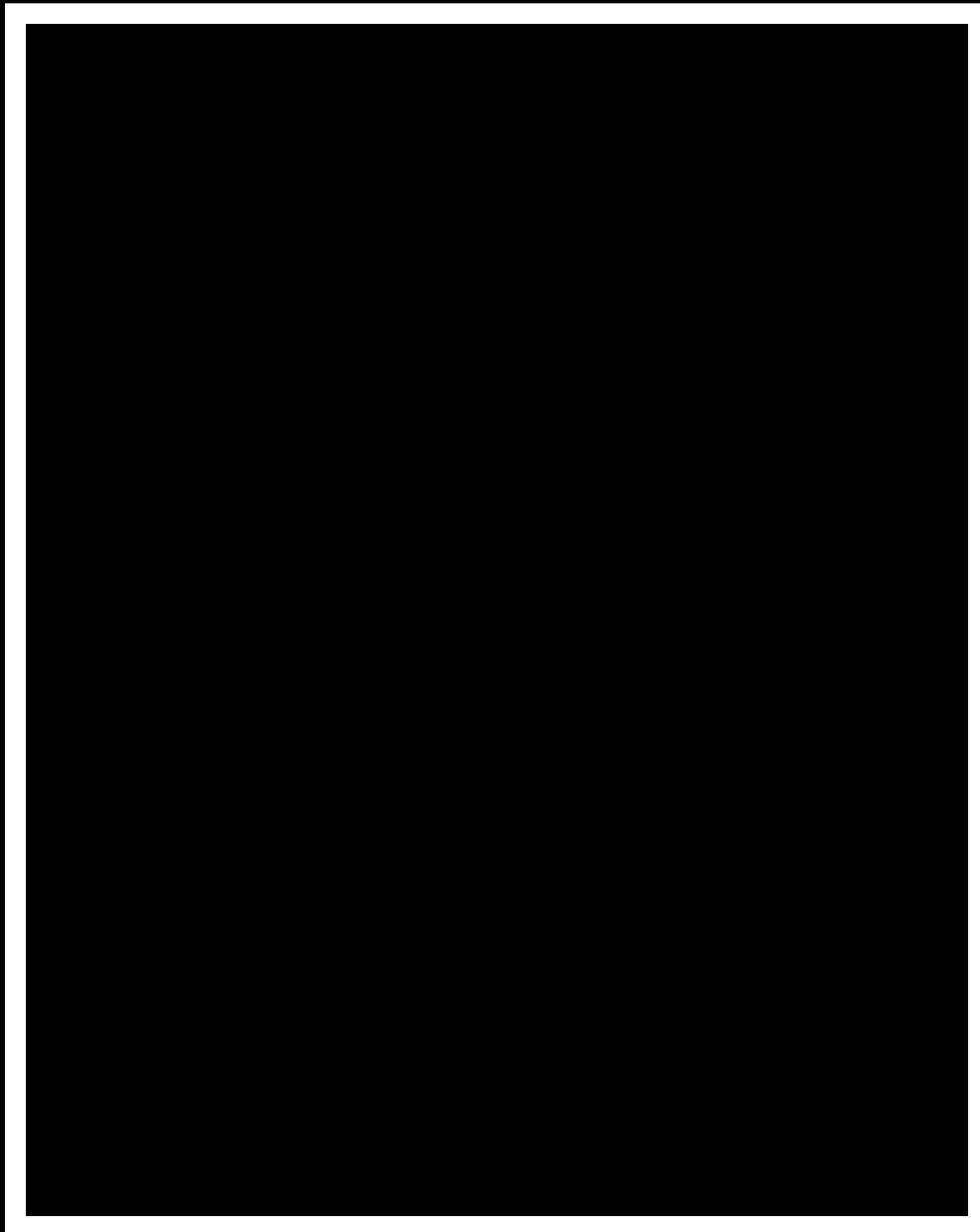
OBJECT-ORIENTED PROGRAMMING

OOP

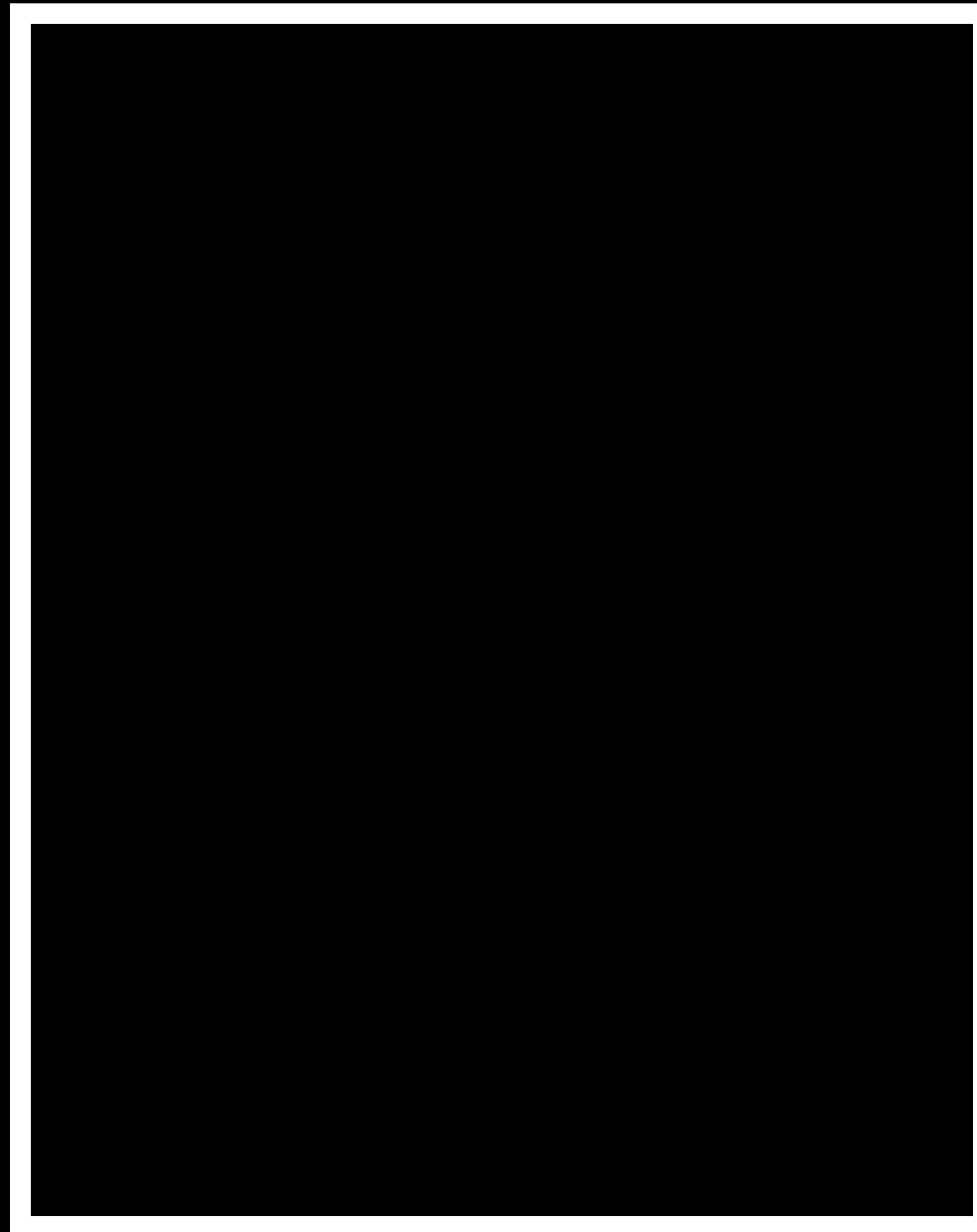
# Object (객체)란?

## Object

함수



변수



클래스(객체)



# Object (객체)란?

## Object

함수

print  
input  
range .....  
list  
dict

변수

name = "정훈"  
age = 123  
height = 185.5  
.....

클래스(객체)

# OOP 그 첫번째

## 캡슐화





무엇인가....  
딱 하나만 기억하자

`print()` 가 어떻게  
돌아가는지 알고서 사용하는  
개발자가 과연 몇 명이나 될까?

동작 구현이 어떻게 이뤄져 있는지 감추고,  
외부에서 접근하지 못하도록 함으로서 다른  
객체와의 결합도를 낮게 유지한다.  
근데 이렇게 설명하면 모르잖아

## 캡슐화의 핵심

안에서 어떤 식으로 구동되는지  
몰라도 그냥 가져다가 사용한다!

```
__sus => _{class_name}__sus
```

다른 사용자가 내부적으로 실행되는 것에서  
바꾸면 안되는 값들은  
\_\_를 붙여서 변수를 선언한다.

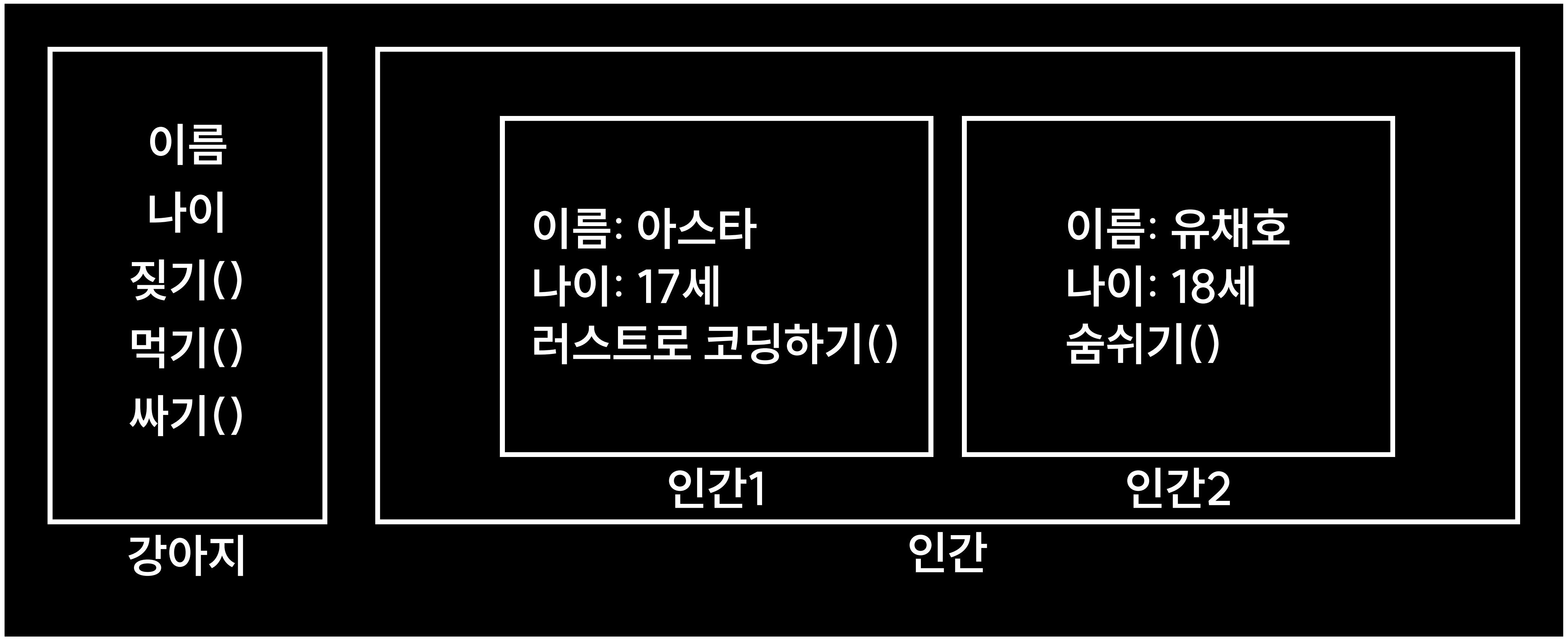


OOP 그 두번째

★상속★



상속



생명체

상속

생명체-> 동물 -> 하위요소  
을 구현해보자



상속

자 잘 따라오세요