

## Chapter 09

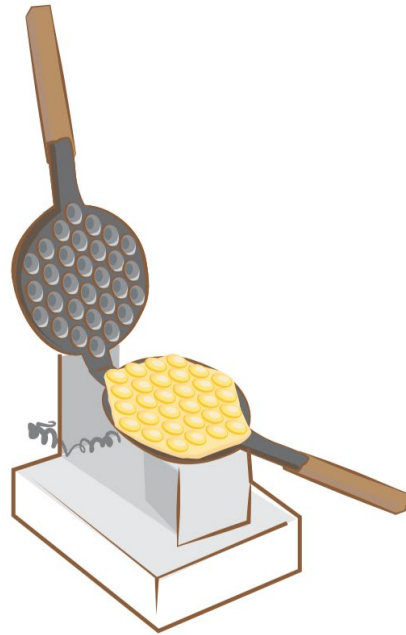
# 객체지향 프로그래밍

# 객체 지향 프로그래밍

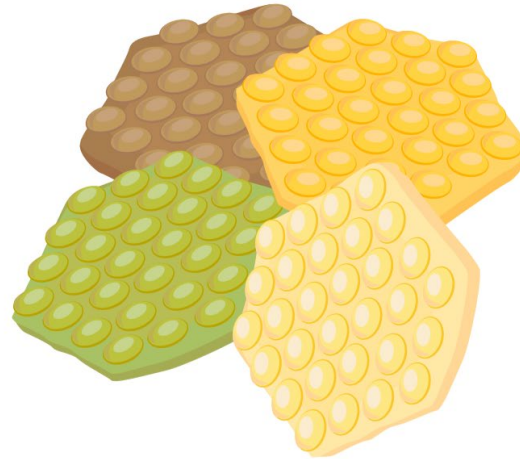
---

- 객체 : 클래스로부터 생성되어 클래스의 속성과 메소드를 가짐
- 클래스 : 객체에서 사용되는 속성과 메소드를 정의한 틀
- 속성 : 클래스와 객체에서 사용되는 변수
- 메소드 : 클래스와 객체에서 사용되는 함수

# 클래스와 객체의 관계



클래스(와플기계)



객체(와플빵)

# 객체지향의 예

ex9-1.py

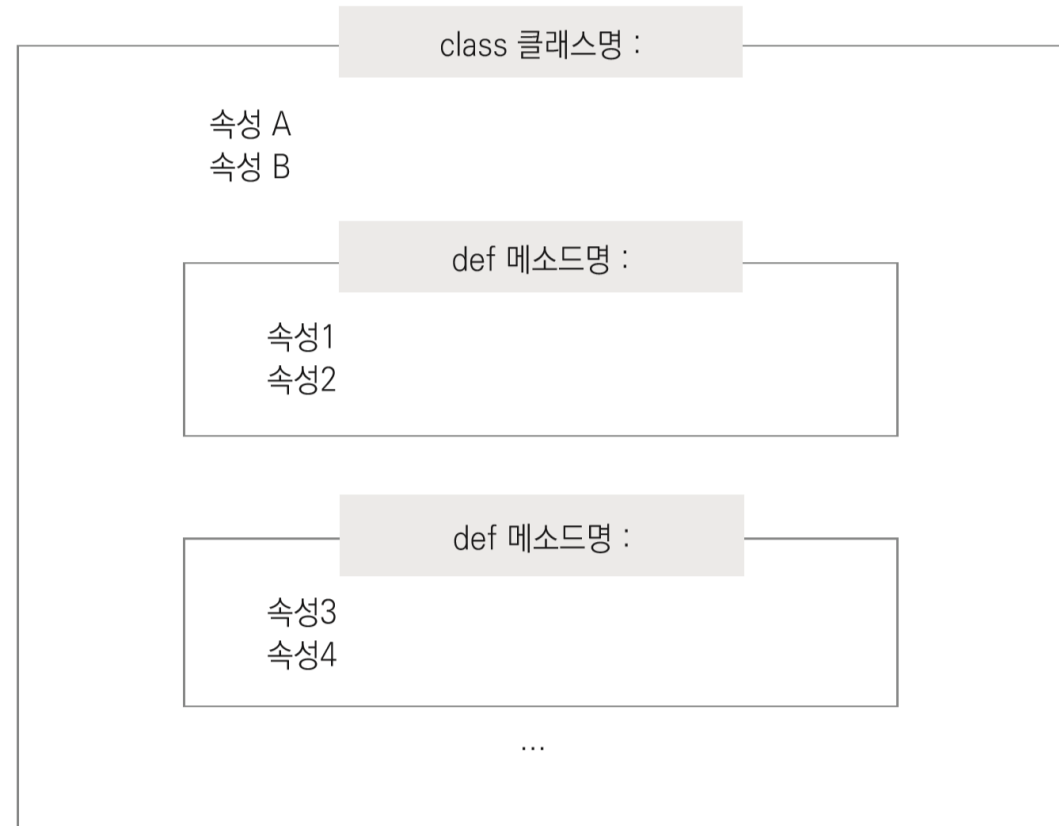
```
class Calculator :  
    def set(self, x, y) :  
        self.first = x  
        self.second = y  
  
    def add(self) :  
        result = self.first + self.second  
        return result  
  
cal1 = Calculator()
```

```
print('%d + %d = %d' % (cal1.first, cal1.second,  
cal1.add()))  
cal1.set(100, 200)  
print('%d + %d = %d' % (cal1.first, cal1.second,  
cal1.add()))
```

:: 실행 결과

```
10 + 20 = 30  
100 + 200 = 300
```

# 클래스의 속성과 메소드



# 생성자

ex9-2.py

```
class Member :  
    def __init__(self, name, age) :  
        self.name = name  
        self.age = age  
  
    def showMember(self) :  
        print('이름 : %s' % self.name)  
        print('나이 : %d' % self.age)
```

```
mem1 = Member('홍지수', 24)  
mem1.showMember()  
mem2 = Member('안지영', 20)  
mem2.showMember()
```

:: 실행 결과

```
이름 : 홍지수  
나이 : 24  
이름 : 안지영  
나이 : 20
```

# 클래스 속성

ex9-3.py

```
class MyClass :  
    number = 100  # 클래스 속성  
    def inc_10(self):  
        MyClass.number += 10  
  
    def inc_20(self):  
        MyClass.number += 20  
  
obj1 = MyClass()  
obj1.inc_10()  
print(obj1.number)
```

```
obj2 = MyClass()  
obj2.inc_20()  
print(obj2.number)
```

:: 실행 결과

```
110  
130
```

# 인스턴스 속성

ex9-4.py

```
class MyClass :  
    def __init__(self, number) :  
        self.number = number    # 인스턴스 속성  
    def inc_10(self):  
        self.number += 10  
    def inc_20(self):  
        self.number += 20  
  
obj1 = MyClass(100)  
obj1.inc_10()
```

```
obj1.inc_20()  
print(obj1.number)
```

```
obj2 = MyClass(200)  
obj2.inc_10()  
obj2.inc_20()  
print(obj2.number)
```

:: 실행 결과

```
110  
230
```



# Q9-1 객체지향 원의 면적과 원주

:: 실행 결과

반지름: 10  
원의 면적 : 314.16  
원주의 길이 : 62.83

```
import math
```

```
class Circle :
```

```
    def __init__(self, radius) :
```

```
        ❶ _____ = radius
```

```
    def getArea(self) :
```

```
        ❷ _____ = math.pi * self.radius * self.radius
```

```
    return area
```

```
def getCircum(self) :
```

```
    circum = 2 * 3.141592 * ❸ _____
```

```
    return circum
```

```
cir = Circle(10)
```

```
print('반지름: %d' % cir.radius)
```

```
print('원의 면적 : %.2f' % ❹ _____)
```

```
print('원주의 길이 : %.2f' % ❺ _____)
```

# Q9-2 객체지향 합계/평균

:: 실행 결과

- 3과목 합계와 평균  
이름 : 김성윤  
국어 : 85, 영어 : 90, 수학 : 83  
합계 : 258, 평균 : 86.0

```
class SumAvg :  
    title = '- 3과목 합계와 평균'  
    def ❶_____(self, name, kor, eng, math) :  
        self.name = name  
        self.kor = kor  
        self.eng = eng  
        self.math = math
```

```
def getSum(self) :  
    sum = self.kor + self.eng + self.math  
    return ❷_____
```

```
s1 = SumAvg('김성윤', 85, 90, 83)  
print(SumAvg.❸_____)  
print('이름 : %s' % s1.name)  
print('국어 : %d, 영어 : %d, 수학 : %d' % (s1.kor,  
s1.eng, s1.math))  
print('합계 : %d, 평균 : %.1f' % (❹_____,  
❺_____))
```

# Q9-3 객체지향 사칙연산 계산

:: 실행 결과

첫번째 수를 입력하세요 : 10  
두번째 수를 입력하세요 : 20  
10 - 20 = -10  
10 / 20 = 0.5

```
class Calculator :
```

```
    def __init__(self, num1, num2) :
```

```
        ❶ _____ = num1
```

```
        ❷ _____ = num2
```

```
    def add(self) :
```

```
        return self.num1 + self.num2
```

```
    def sub(self) :
```

```
        return self.num1 - self.num2
```

```
    def mul(self) :
```

```
        return self.num1 * self.num2
```

```
    def div(self) :
```

```
        return self.num1 / self.num2
```

```
a = int(input('첫번째 수를 입력하세요 : '))
```

```
b = int(input('두번째 수를 입력하세요 : '))
```

```
cal1 = Calculator(a, b)
```

```
print('%d - %d = %d' % (a, b, ❸ _____ ))
```

```
print('%d / %d = %.1f' % (a, b, ❹ _____ ))
```

# Q9-4 생성자 매개변수 리스트

:: 실행 결과

성명 : 김지혜  
이메일 : rubato@hanmail.net  
전화번호 : 010-1234-4567

```
class Person :  
    def __init__(self, ❶_____):  
        ❷_____ = info  
        def getName(self) :  
            return self.❸_____  
        def getEmail(self) :  
            return self.❹_____
```

```
def getPhoneNum(self) :
```

```
    return self.❺_____
```

```
info = ['김지혜', 'rubato@hanmail.net', '010-1234-4567']
```

```
person = Person(info)
```

```
print('성명 : %s' % person.getName())
```

```
print('이메일 : %s' % person.getEmail())
```

```
print('전화번호 : %s' % person.getPhoneNum())
```

# Q9-5 객체지향 문자열 다루기

:: 실행 결과

역순 : !snoipmahc eht era eW  
하이픈(-) 삽입 : We-are-the-champions!

```
class EngSentence :  
    def __init__(self, sentence) :  
        self.sentence = ❶_____ ❷_____  
    = len(self.sentence)  
    def reverse(self) :  
        tmp = ''  
        for i in range(self.length) :  
            tmp += (self.sentence[❸_____])  
        return tmp
```

```
def insertHypen(self) :  
    tmp = ''  
    for i in range(self.length) :  
        if self.sentence[i] == ❹_____ :  
            tmp += '-'  
        else :  
            tmp += self.sentence[i]  
    return tmp  
  
a = 'We are the champions!'  
eng1 = EngSentence(a)  
print('역순 : %s' % ❺_____)  
print('하이픈(-) 삽입 : %s' % eng1.insertHypen())
```

# 클래스 상속

ex9-5.py

```
class Animal:
    def __init__(self, name):
        self.name = name
    def printName(self):
        print(self.name)

class Dog(Animal):
    def __init__(self, name, sound):
        super().__init__(name)
        self.sound = sound
```

```
def printSound(self):
    print(self.sound)
```

```
dog1 = Dog('행복이', '멍멍~~~')
dog1.printName()
dog1.printSound()
```

:: 실행 결과

```
행복이
멍멍~~~
```

# super() 함수

ex9-6.py

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

    def printInfo(self):
        print('이름:%s, 나이:%d' % (self.name, self.age))

    def getInfo(self) :
        return self.name + ', ' + str(self.age)
```

```
class Student(Person):
    def __init__(self, name, age, department, id):
        super().__init__(name, age)
        self.department = department
        self.id = id

    def printStudentInfo(self):
        name_age = super().getInfo()
        print(name_age)
        print('%s님의 학과:%s, 학번:%s' % (self.name,
self.department, self.id))
```

# super() 함수(계속)

```
x = Student('홍지수', 20, '소프트웨어공학과', '20215550001')  
x.printInfo()  
x.printStudentInfo()
```

:: 실행 결과

```
이름:홍지수, 나이:20  
홍지수, 20  
홍지수님의 학과:소프트웨어공학과, 학번:20215550001
```