

# 코드로 한 번에 이해되는 파이썬 (10)

실습 코드 : [https://github.com/hansikyung/Python\\_OneTime](https://github.com/hansikyung/Python_OneTime)

# Today's point!



- ✓ 클래스란?
- ✓ 클래스의 정의
- ✓ 객체 생성
- ✓ 상속과 오버라이딩

# 클래스란?

## 객체지향성(OOP, Object-Oriented Programming)

프로그램을 객체들의 상호작용으로 구성하는 것

- 코드의 재사용성, 확장성, 유지보수성을 높여 줌
- 복잡한 시스템을 모듈화 하여 다형성/캡슐화를 통해 유연하게 개발 가능함
- 단점! 1. 잘못 쓰면 오히려 복잡하게 구성될 수 있음
- 2. 작은 프로그램에서는 부담이 될 수 있으며, 성능이나 메모리의 오버헤드가 일어날 수 있음

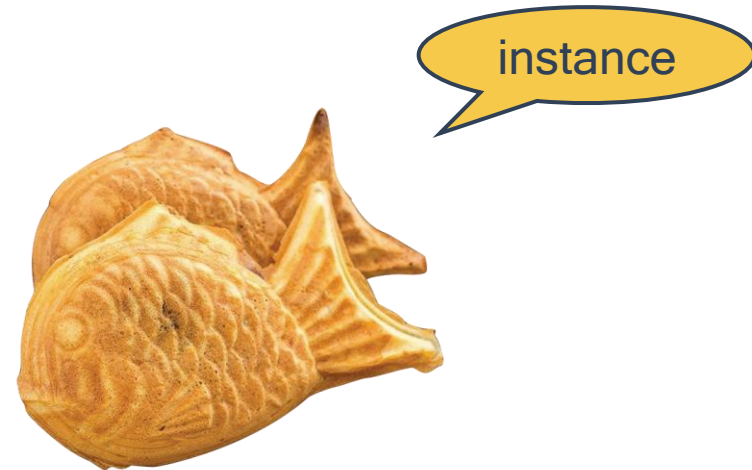


# 클래스란?

## 클래스

속성과 메서드로 이루어져 있으며, 인스턴스로 생산되어 활용하는 것

- 속성 : 클래스가 가지고 있는 변수
- 메서드 : 클래스가 가지고 있는 함수
- 인스턴스 : 클래스를 통해 새로 만들어진 1개의 객체



# 클래스의 정의

- 클래스의 인스턴스 생성은 아래와 같이 적용해볼 수 있습니다.
- 클래스는 `__init__` 이라는 함수를 반드시 가지고 있어야 합니다.
- `self.` 가 붙은 것은 이 클래스에 속해 있는 속성과 메서드를 의미합니다.

```
class fish_bread():  
    def __init__(self):  
        self.flour = None  
        self.filing = None  
  
    def whatisfiling(self):  
        print(self.filing, '맛 붕어빵이다!')
```

```
class fish_bread():  
    def __init__(self, flour, filing):  
        self.flour = flour  
        self.filing = filing
```

# 클래스의 객체 생성

- 클래스를 통해 인스턴스를 생성한 후 객체는 다음과 같이 활용할 수 있습니다.

```
class fish_bread():  
    def __init__(self):  
        self.flour = None  
        self.filing = None  
  
    def whatisfiling(self):  
        print(self.filing, '맛 붕어빵이다!')
```

```
bread = fish_bread()  
bread.whatisfiling()
```

```
'None맛 붕어빵이다!'
```

# 클래스의 객체 생성

- 클래스를 통해 인스턴스를 생성한 후 객체는 다음과 같이 활용할 수 있습니다.

```
class fish_bread():  
    def __init__(self, flour, filing):  
        self.flour = flour  
        self.filing = filing  
  
    def whatisfiling(self):  
        print(self.filing, '맛 붕어빵이다!')
```

```
bread = fish_bread(50, '팥')  
bread.whatisfiling()
```

```
'팥맛 붕어빵이다!'
```

# 클래스의 상속과 오버라이딩

- 클래스는 '상속'을 통해 자신이 가지고 있는 속성과 메서드를 다른 클래스에 물려줄 수 있습니다.
- 상속은 아래와 같습니다. 이 때 물려준 클래스는 '부모클래스' , 물려받은 클래스는 '자식클래스' 라고 합니다.

```
class fish_bread():  
    def __init__(self):  
        self.flour = None  
        self.filing = None  
  
    def whatisfiling(self):  
        print(self.filing, '맛 붕어빵이다!')
```

```
class peanut_bread(fish_bread):  
    def __init__(self):  
        super().__init__()
```

```
new_bread = peanut_bread()  
new_bread.whatisfiling()
```



# 클래스의 상속과 오버라이딩

- 오버라이딩은 상속받은 자식 클래스와 부모와 다른 메서드를 가지고 싶을 때,
- 부모가 가졌던 함수 이름을 똑같이 사용하면서
- 자식 클래스에서 동작을 다르게 하는 것을 의미합니다.

```
class fish_bread():  
    def __init__(self):  
        self.flour = None  
        self.filing = None  
  
    def whatisfiling(self):  
        print(self.filing, '맛 붕어빵이다!')
```

```
class peanut_bread(fish_bread):  
    def __init__(self):  
        super().__init__()  
  
    def whatisfiling(self):  
        print(f'땅콩이 포함된 {self.filing}맛 빵입니다.')
```