

# 프로그래밍2 튜터링

(3회차)

조세연

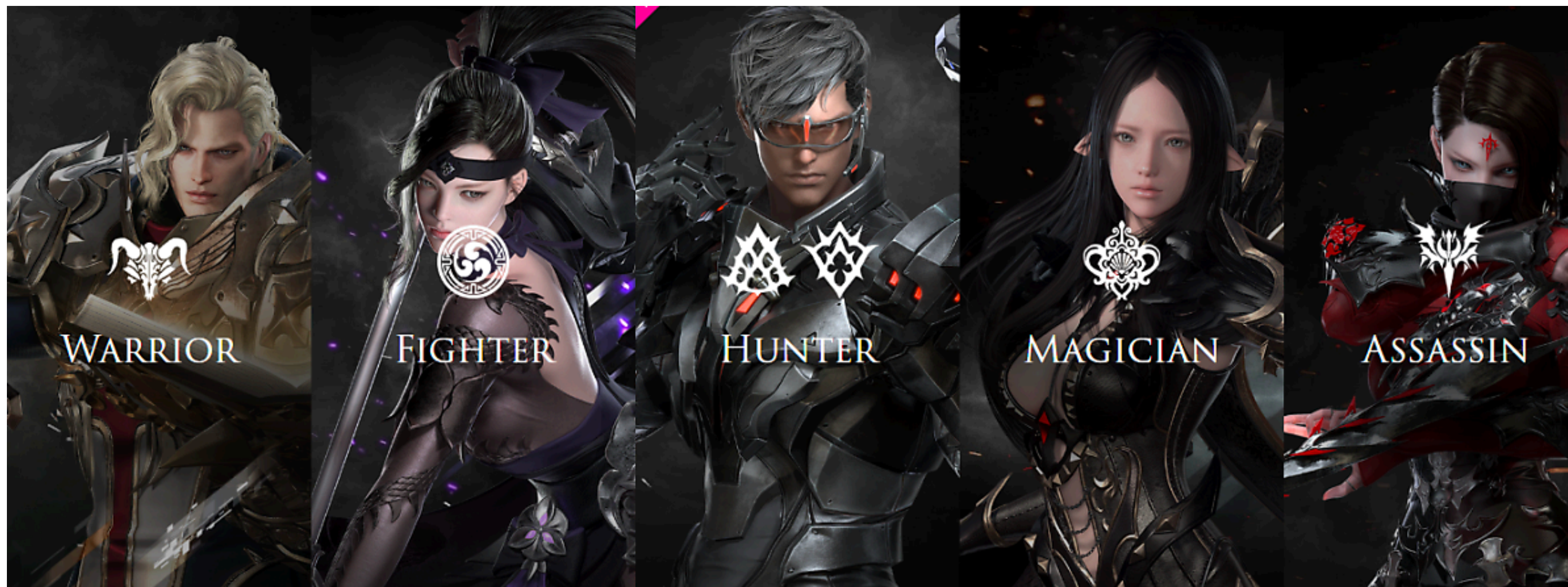
# 목표

- class 개념 익히기
- 9/26에 진행했던 파이썬 시험 문제에 클래스 적용해보기



게임을 만들어보자!!





로스트아크

워리어, 파이터, 헌터 등등...

이들의 공통점은??

## 사람 (직업)

=> 공격력 / 방어력

=> 체력

=> 제스처

=> 마나

=> 무기

=> 스킬

마법사

마법 데미지, 빛 섬광을 날림, 마법 지팡이

헌터

물리 데미지, 주먹으로 내리침, 무기없음

암살자

물리 데미지, 그림자로 접근해 표창을 날림, 표창

힐러

마법 데미지, 주변의 적들을 회복 및 공속 증가, 우산

## 사람 (직업)

=> 공격력 / 방어력

=> 체력

=> 제스처

=> 마나

=> 무기

=> 스킬

이를 기반으로 class를 정의한다면, 어떤 것이 속성이고 어떤 것이 함수일까?

HINT. 속성은 어떠한 상태, 고유한 특성

함수는 어떠한 행동, 동작

속성

공격력/방어력, 체력, 마나, 무기

함수

제스처, 스킬

# class 없이 설계하면?

- 마법사
  - 공격력/방어력, 제스처, 마나, 체력, 스킬...
- 헌터
  - 공격력/방어력, 제스처, 마나, 체력, 스킬...
- 하나하나...

- 마법사
  - 공격력 10, 방어력 10
  - 마나 100
  - 체력 1000
  - 스킬(): 빛 섬광 날림, 체력 - 공격력(10), 마나-10
- 헌터
  - 공격력 15, 방어력 40
  - 마나 0
  - 체력 3000
  - 스킬(): 주먹으로 내려치, 체력 - 공격력(15)

# class가 있으면!!

- 직업
  - 공격력/방어력, 제스처, 마나, 체력, 스킬...

끝



# class == 틀

- 직업

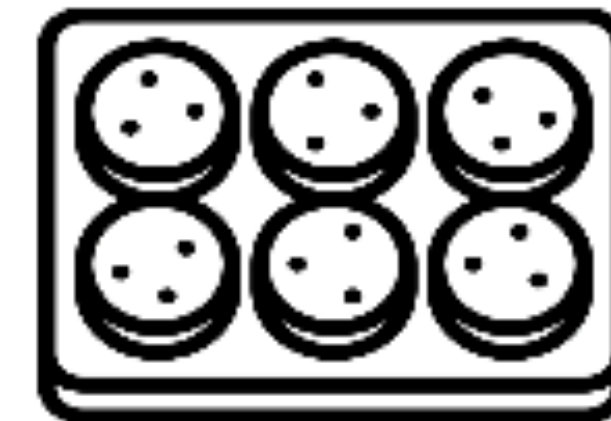
- 공격력/방어력, 제스처, 마나, 체력, 스킬...

마법사    헌터    암살자    힐러

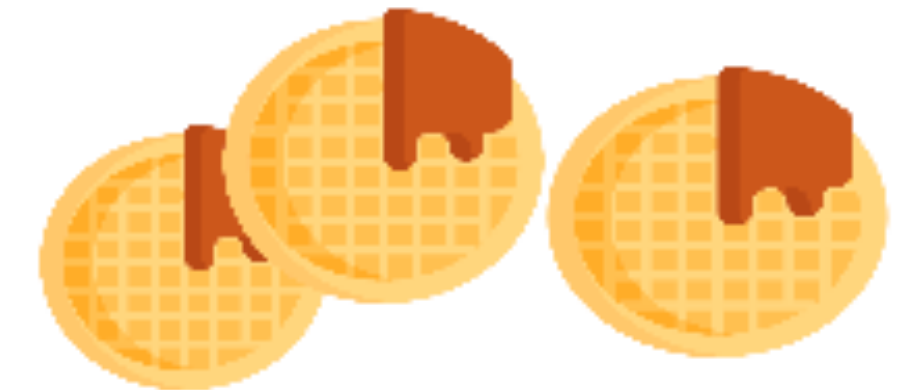
각각에 맞는 특성을 넣어줌!

그렇게 만들어진 것이 바로 "객체"

## 클래스



## 객체



## class 정의

- 객체를 생성하기 위한 설계도 혹은 **틀**
  - 서로 관련되어 있는 **메서드의 집합**
  - 객체들의 **속성**을 묶어 놓은 **집합체**

## 객체(object) 정의

- 어떠한 **속성값**과 **행동**을 갖고 있는 데이터
- 클래스로부터 만들어진 객체를 '클래스의 인스턴스(instance)'라고 부름

## class 만드는 법

- 클래스는 class에 클래스 이름을 지정하고 :(콜론)을 붙임
  - class 클래스명:

## 객체 생성하는 법

- 객체명 = 클래스명(속성)

## \_\_init\_\_??

- 생성자(Initializer)
  - 클래스로부터 새 객체를 생성할 때마다 실행되는 메서드로, \_\_init\_\_() 으로 작성
  - 객체를 생성할 때, 인스턴스 변수를 초기화하거나 객체의 초기상태를 만들기 위한 문장을 실행하는 곳

## self ??

- 자기 자신을 의미

## 예제 살펴보기

```
class Person:
    def __init__(self, name, age, address):
        self.hello = '안녕하세요.'
        self.name = name
        self.age = age
        self.address = address

    def greeting(self):
        print('{0} 저는 {1}입니다.'.format(self.hello, self.name))
```

## 예제 살펴보기

```
maria = Person('마리아', 20, '서울시 서초구 반포동')
maria.greeting()      # 안녕하세요. 저는 마리아입니다.
```

```
print('이름:', maria.name)      # 마리아
print('나이:', maria.age)       # 20
print('주소:', maria.address)    # 서울시 서초구 반포동
```

```
maria = Person('마리아', 20, '서울시 서초구 반포동')

class Person:
    def __init__(self, name, age, address):
        self.hello = '안녕하세요.'
        self.name = name
        self.age = age
        self.address = address
```

The diagram illustrates the relationship between the class constructor and the object instance. Three blue brackets are placed above the arguments '마리아', 20, and '서울시 서초구 반포동' in the line `maria = Person('마리아', 20, '서울시 서초구 반포동')`. Three blue arrows point from these brackets down to the parameters `name`, `age`, and `address` respectively in the `def __init__(self, name, age, address):` line of the `Person` class definition.



# 문제풀이 - class로 변경

아래와 같은 하나의 string을 **리스트로 변환**해서 출력하세요. 문자열에는 정수, 콤마, 빈칸만 사용할 수 있습니다. string 내부에는 **동일한 숫자**를 사용할 수 없습니다.

아래의 문제에서는 2가지 문자열을 테스트하세요. 이 문제는 아래를 입력받아서 **2번째 가장 큰 값**을 출력하는 프로그램을 작성하는 문제입니다. 이를 순서대로 구현하시오.

- 위의 내용을 함수로 구현하세요. 위의 문제의 입력과 같이 **하나의 string**을 파라미터로 받아서, **정수 리스트를 반환**하는 함수를 작성하시오.

- 이 함수의 이름은 **string\_2\_list()**로 하세요.

- 이 함수는 **list**를 반환합니다.

- 위에서 만든 **list**를 **파라미터**로 받아서 **최대값을 반환**하는 **함수**를 작성하시오.

- 함수의 이름은 **max\_of\_list()**로 하세요.

- 파이썬의 built-in 함수인 max() 함수를 사용하지 마세요.

- 이 함수는 **int**를 반환합니다.

- 이 함수를 이용하여, 위의 2개의 string 샘플을 테스트, 출력하시오.

- 위의 **max\_of\_list()** 함수를 변경해서 **second\_max\_of\_list()** 함수를 작성하시오.

- max() 함수를 사용하지 마세요.

- 이 함수는 **int**를 반환합니다.

- 이 함수를 이용하여, 위의 2개의 string 샘플을 테스트, 출력하시오.

# 문제풀이 - class로 변경

```
class Calc:
    def __init__(self, strInputNum):
        self.listNum = list(map(int, strInputNum.split(",")))

    def max_of_list(self):
        nMax = self.listNum[0]
        for i in self.listNum:
            if i > nMax:
                nMax = i
        return nMax

    def second_max_of_list(self, bDuplicate = False):
        arrNew = []
        if bDuplicate:
            arrNew = list(set(self.listNum))
        else:
            arrNew = self.listNum
```

```
nFirstMax = nSecondMax = -float("inf")
for i in arrNew:
    if i > nFirstMax:
        nSecondMax = nFirstMax
        nFirstMax = i
    elif nSecondMax < i < nFirstMax:
        nSecondMax = i
return nSecondMax

def second_max_of_list_with_same_numbers(self):
    return self.second_max_of_list(True)

inputNum = input("문자열을 입력하세요: ")
listObj = Calc(inputNum)
print("first max number: ", listObj.max_of_list())
print("second max number: ", listObj.second_max_of_list())
print("second max number(중복제거): ",
listObj.second_max_of_list_with_same_numbers())

# 10, 20, 40, 30, 20, 50, 10, 70, 30, 10
```

감사합니다:)