

# 2020-2학기 PYTHON 기초 튜터링

---

정보사회학과 김재훈

## 6차시



## 오늘 할 내용

- ▶ 클래스 (Class)
- ▶ 직접 알고리즘 문제 풀어보기
  - ▶ 1번 문제 (추후 공개)
  - ▶ 2번 문제 (추후 공개)



# 클래스(CLASS)

클래스(class)는 [객체 지향 프로그래밍](#)(OOP)에서 특정 객체를 생성하기 위해 [변수](#)와 메소드를 정의하는 일종의 틀이다. 객체를 정의 하기 위한 상태(멤버변수)와 메서드(함수)로 구성된다.<sup>[1][2]</sup>

[템플릿](#)을 사용하면 객체를 클래스로 정의할 때 멤버의 자료형을 미리 정하지 않고 객체를 사용할 때 결정할 수 있다. 이를 통해 클래스나 멤버의 중복 정의를 하지 않아도 되므로 효율적으로 코딩이 가능하다.

객체는 클래스로 규정된 인스턴스로서, 변수 대신 실제값을 가진다.

클래스는 OOP를 정의하는 개념 중 하나인데, 클래스에 대한 중요한 몇가지의 개념들은 다음과 같다.

클래스는 전부 혹은 일부를 그 클래스 특성으로부터 상속받는 [서브클래스](#)를 가질 수 있으며, 클래스는 각 서브클래스에 대해 [수퍼클래스](#)가 된다.

서브클래스는 자신만의 메소드와 변수를 정의할 수도 있다.

이러한 클래스와 그 서브클래스 간의 구조를 "클래스 계층(hierarchy)"이라 한다.



# 클래스(CLASS)

- ▶ 클래스의 두 가지 개념
  - ▶ 클래스 = 설계도 (틀)
  - ▶ 객체 (Object) = 클래스를 통해 찍어낸 도구



# 클래스(CLASS)

## ▶ 왜 필요한가?



# 클래스(CLASS)

- ▶ 왜 필요한가?

- ▶ 지난 시간 우리가 만들었던 함수 Adder (덧셈기)



# 클래스(CLASS)

- ▶ 왜 필요한가?
  - ▶ 지난 시간 우리가 만들었던 함수 Adder (덧셈기)
  - ▶ 만약 덧셈기가 직전 값을 계속 저장하고 있어야 한다면?



# 클래스(CLASS)

- ▶ 왜 필요한가?
  - ▶ 지난 시간 우리가 만들었던 함수 Adder (덧셈기)
  - ▶ 만약 덧셈기가 직전 값을 계속 저장하고 있어야 한다면?
    - ▶ Ex. 장부 정리 업무중인 사원 5명이, 각자 덧셈기를 한 대씩 써야한다면?





# 클래스(CLASS)

## ▶ 왜 필요한가?

- ▶ 지난 시간 우리가 만들었던 함수 Adder (덧셈기)
- ▶ 만약 덧셈기가 직전 값을 계속 저장하고 있어야 한다면?
  - ▶ Ex. 장부 정리 업무중인 사원 5명이, 각자 덧셈기를 한 대씩 써야한다면?
    - ▶ Adder1, Adder2, Adder3, Adder4, Adder5 ... ?



# 클래스(CLASS)

## ▶ 왜 필요한가?

- ▶ 지난 시간 우리가 만들었던 함수 Adder (덧셈기)
- ▶ 만약 덧셈기가 직전 값을 계속 저장하고 있어야 한다면?
  - ▶ Ex. 장부 정리 업무중인 사원 5명이, 각자 덧셈기를 한 대씩 써야한다면?
    - ▶ Adder1, Adder2, Adder3, Adder4, Adder5 ... ?

```
def Adder1(num1, num2): def Adder2(num1, num2): def Adder3(num1, num2): def Adder4(num1, num2): def Adder5(num1, num2):
```

```
    result = num1 + num2
```

```
    result = num1 + num2
```

```
    result = num1 + num2
```

```
    result = num1 + num2
```

```
    result = num1 + num2
```

```
    return result
```

```
    return result
```

```
    return result
```

```
    return result
```

```
    return result
```



## 클래스(CLASS)

- ▶ 덧셈기를 찍어 낼 수 있는 틀(Class)을 활용
  - ▶ 원하는 대로 찍어낸 덧셈기들 (Objects)
    - ▶ 각자 독립적인 역할 (각자 다른 최신 값을 저장 가능)
    - ▶ 각자 다른 기능으로 수정 가능 (Adder2는 결과에서 매번 -1 해준다던지...)



# 클래스(CLASS)

- ▶ 조금 후 Jupyter Notebook에서 실습으로 마저 진행하겠습니다!



## 직접 알고리즘 문제 풀어보기

- ▶ 5차시까지 배운 내용 중 일부를 활용해서 풀 수 있는 아주 간단한 문제 2개
  - ▶ 1. 별찍기
  - ▶ 2. 숫자 뒤집어 비교하기



# 직접 알고리즘 문제 풀어보기

▶ 5차시까지 배운 내용 중 일부를 활용해서 풀 수 있는 아주 간단한 문제 2개

▶ 1. 별찍기

## 문제

첫째 줄에는 별 1개, 둘째 줄에는 별 2개, N번째 줄에는 별 N개를 찍는 문제

## 입력

첫째 줄에  $N(1 \leq N \leq 100)$ 이 주어진다.

## 출력

첫째 줄부터 N번째 줄까지 차례대로 별을 출력한다.

## 예제 입력 1 복사

5

## 예제 출력 1 복사

```
*  
**  
***  
****  
*****
```

출처 : <https://www.acmicpc.net/problem/2438>



## ▶ 5차시까지 배운 내용 중 일부를 활용해서 풀 수 있는 아주 간단한 문제 2개

### ▶ 2. 숫자 뒤집어 비교하기

#### 문제

상근이의 동생 상수는 수학을 정말 못한다. 상수는 숫자를 읽는데 문제가 있다. 이렇게 수학을 못하는 상수를 위해서 상근이는 수의 크기를 비교하는 문제를 내주었다. 상근이는 세 자리 수 두 개를 칠판에 써주었다. 그 다음에 크기가 큰 수를 말해보라고 했다.

상수는 수를 다른 사람과 다르게 거꾸로 읽는다. 예를 들어, 734와 893을 칠판에 적었다면, 상수는 이 수를 437과 398로 읽는다. 따라서, 상수는 두 수중 큰 수인 437을 큰 수라고 말할 것이다.

두 수가 주어졌을 때, 상수의 대답을 출력하는 프로그램을 작성하시오.

#### 입력

첫째 줄에 상근이가 칠판에 적은 두 수 A와 B가 주어진다. 두 수는 같지 않은 세 자리 수이며, 0이 포함되어 있지 않다.

#### 출력

첫째 줄에 상수의 대답을 출력한다.

#### 예제 입력 1 복사

734 893

#### 예제 출력 1 복사

437

출처 : <https://www.acmicpc.net/problem/2908>



## 6주차

### 문제

상근이의 동생 상수는 수학을 정말 못한다. 상수는 숫자를 읽는데 문제가 있다. 이렇게 수학을 못하는 상수를 위해서 상근이는 수의 크기를 비교하는 문제를 내주었다. 상근이는 세 자리 수 두 개를 칠판에 써주었다. 그 다음에 크기가 큰 수를 말해보라고 했다.

상수는 수를 다른 사람과 다르게 거꾸로 읽는다. 예를 들어, 734와 893을 칠판에 적었다면, 상수는 이 수를 437과 398로 읽는다. 따라서, 상수는 두 수중 큰 수인 437을 큰 수라고 말할 것이다.

두 수가 주어졌을 때, 상수의 답을 출력하는 프로그램을 작성하시오.

### 입력

첫째 줄에 상근이가 칠판에 적은 두 수 A와 B가 주어진다. 두 수는 같지 않은 세 자리 수이며, 0이 포함되어 있지 않다.

### 출력

첫째 줄에 상수의 답을 출력한다.

### 예제 입력 1 복사

734 893

### 예제 출력 1 복사

437

출처 : <https://www.acmicpc.net/problem/2908>

