

Week 03 • 데이터 저널리즘

Python Crash Course 2

Joonhwan Lee

human-computer interaction + design lab.

오늘 다룰 내용

- Flow Control
- File I/O
- Object Oriented Programming

1. Object Oriented Programming

클래스와 오브젝트

- ◆ 파이썬에서 데이터의 기본 단위를 “오브젝트(객체)”라고 부름.
- ◆ 변수, 리스트 등, 모든 것이 오브젝트.
- ◆ 파이썬에서 주로 사용되는 오브젝트의 종류.

오브젝트	오브젝트가 하는 일	클래스 이름
수치 오브젝트	수치를 표현	numbers
문자열 오브젝트	문자열을 표현	string
정규 표현 오브젝트	문자열 매칭을 위한 패턴 표현	re
시간 오브젝트	시간을 표현	datetime
파일 오브젝트	파일 읽기/쓰기에 사용	io
해시 오브젝트	해시를 표현	dict
리스트 오브젝트	배열을 표현	list

클래스

- ◆ 오브젝트의 성질이나 기능을 나타내는 일종의 설계도.
- ◆ 클래스로 부터 오브젝트를 생성해 낼 수 있음.
 - ◆ `datetime` 클래스는 날짜와 시간이라는 데이터를 표현하기 위해 만들어 짐.
 - ◆ `datetime`이라는 클래스로 부터 현재 시간, 특정 시간의 오브젝트를 생성할 수 있음.
 - ◆ `now = datetime.now()` # 현재의 시간을 표현하는 `now` 라는 오브젝트 생성

Object Oriented Programming (OOP)

◆ 객체지향 프로그래밍 (OOP)

- ◆ 객체지향 프로그래밍은 여러 단위의 객체 간의 협업을 통해 프로그램이 수행하고자 하는 목적을 이루는 방법.
- ◆ 각각의 클래스는 서로 다른 역할을 담당하고 있고, 여러 클래스로 만들어진 객체(object)가 모여서 프로그램을 완성.
- ◆ 각각의 객체는 서로 메시지를 주고 받음.
- ◆ 예: 자동차
 - ◆ 엔진 객체: 엑셀로 부터 메시지를 받아 작동, 바퀴에 메시지 전달
 - ◆ 바퀴 객체: 엔진으로 부터 메시지 받아 회전, 브레이크에서 메시지 받으면 회전을 멈춤. 운전대로부터 좌/우 이동 메시지 받음.
 - ◆ 운전대 객체: 좌/우로 움직이며 바퀴에게 메시지 전달.
 - ◆ 페달 객체 → 악셀 객체 / 브레이크 객체: 밟았다가 떼 수 있으며 메시지를 바퀴와 엔진에 전달.

Object Oriented Programming (OOP)

- ◆ 클래스는 오브젝트를 생성하기 위한 청사진.
 - ◆ 오브젝트의 상태를 저장하는데 사용될 속성을 정의
 - ◆ 운전대 클래스: 좌/우로 움직임. 버튼이 하나 있고 누르면 소리가 남.
→ 차종에 관계없이 비슷한 기능을 제공, 그러나 차종에 따라 추가되는 속성이 있을 수 있음 (예: 오디오 컨트롤 버튼)
 - ◆ 엔진 클래스: 회전을 하여 동력을 바퀴로 전달. 멈춤 속성과 시동 속성이 있음.
 - ◆ 오브젝트가 이해할 수 있는 메시지와 메시지에 응답하는 과정을 정의 (메소드의 역할)
 - ◆ 엔진 클래스: 엑셀 오브젝트로 부터 메시지가 전달되면 엔진 시동 메소드를 실행하여 엔진을 움직임. 브레이크 오브젝트로부터 메시지가 전달되면 엔진 멈춤 메소드를 실행.
 - ◆ 이 과정에서 클래스가 내부적으로 어떻게 움직이는지 클래스 외부에서 알 필요가 없음 → 예: list에서 sort 메소드.

클래스 만들기

- ◆ 클래스를 만들기 위해서는 `class` 라는 키워드를 사용하며 다음과 같은 구조를 따른다.

```
import random
class Die:
    def roll(self):          ← class method
        return 1 + random.randrange(6)
```

- ◆ 클래스는 다음과 같이 오브젝트를 생성하여 사용한다.
 - ◆ `die1 = Die()` # 주사위 하나 생성
 - ◆ `die1.roll` # 주사위의 메소드인 roll 호출

클래스의 인스턴스 변수

- ◆ 인스턴스 변수는 클래스의 변수. **self.변수이름** 으로 지정
- ◆ 클래스 내에서는 글로벌하게 사용 (어디서나 참조 가능)

```
class Die:  
    def roll(self):  
        self.number = 1 + random.rand...  
  
    def show(self):  
        return self.number
```

```
die.Die()  
die.show()
```

클래스의 인스턴스 변수

- ◆ 인스턴스 변수는 클래스의 변수. **self.변수이름** 으로 지정
- ◆ 클래스 내에서는 글로벌하게 사용 (어디서나 참조 가능)

```
class Die:  
    def roll(self):  
        self.number = 1 + random.rand...  
  
    def show(self):  
        return self.number
```

```
die.Die()  
die.show()    => error !!!
```

클래스의 인스턴스 변수

- ◆ 인스턴스 변수는 클래스의 변수. **self.변수이름** 으로 지정
- ◆ 클래스 내에서는 글로벌하게 사용 (어디서나 참조 가능)

```
class Die:  
    def roll(self):  
        self.number = 1 + random.rand...
```

```
    def show(self):  
        return self.number
```

```
die = Die()  
die.show()    => error !!!
```

이 경우, 반드시 다음과 같이 써야 함.

```
die = Die()  
die.roll()  
die.show()
```

클래스의 인스턴스 변수 초기화

- ♦ 오브젝트가 생성될 때, 인스턴스 변수를 초기화 해주어야.

```
class Die:
    def initialize(self):
        self.roll()

    def roll(self):
        self.number = 1 + random.rand...

    def show(self):
        return self.number
```

클래스의 인스턴스 변수 초기화

- ♦ 오브젝트가 생성될 때, 인스턴스 변수를 초기화 해주어야.

```
class Die:
    def initialize(self):
        self.roll()

    def roll(self):
        self.number = 1 + random.rand...

    def show(self):
        return self.number
```

die = Die() 를 했을 때
__init__ 메소드가 자동 호출되면서
roll 함수가 역시 호출된다.

OOP 연습: Dragon 키우기

- ◆ 다마고치와 같은 Dragon 키우기 프로그램을 만들어 보자.
 - ◆ Dragon은 이름을 가진다.
 - ◆ 밥을 주면 배가 가득 찬다.
 - ◆ 소화관이 차면 산책을 가서 용변을 봐야 한다.
 - ◆ 재우면 3번 코를 골며 자다가 곧 일어난다.
 - ◆ 하늘높이 던지면(toss) 좋아하고, 안아서 흔들어주면(rock) 잠이 든다. 곧 잠에서 깨지만.
 - ◆ 시간이 지날 수록 (한번 뭔가 액션을 취할 수록) 배는 고파지고 소화관은 차오른다.
 - ◆ 배가 너무 고프면 게임은 끝. 소화관이 너무 차면 실례를...

Dragon 게임을 보다 interactive 하게...

- ◆ Dragon 키우기 게임을 interactive 하게 바꾸어 보자.
 - ◆ `input()` 메소드를 참고하여 다음과 같이 명령어의 보기를 주고 선택하게 하자.
 - ◆ Dragon의 이름을 입력해 주세요.
G-Dragon
'G-Dragon'이 태어났습니다.
 - ◆ `commands: feed, toss, walk, rock, put to bed, exit`
feed
'G-Dragon'에게 밥을 주고 있어요.
 - ◆ `commands: feed, toss, walk, rock, put to bed, exit`
toss
'G-Dragon'을 하늘 높이 던졌어요!
즐거워 하네요

Assignment #2-2: Python Crash Course 2

Assignment #2-2: Python Crash Course 2

- ◆ Exercise 문제

- ◆ 다음과 같은 포맷으로 제출

- ◆ 제출 방법: GitHub (다음 주 일요일 9/29 자정까지)
 - ◆ 파일 이름: A2-2-학번-이름(영어로).iphynb 형식으로
(예: A2-2-13403-999-jiyoon.iphynb)

Assignment #2-2: Python Crash Course 2

- ◆ #1. 다음을 출력하는 프로그램을 작성하시오.
 - ◆ 100 little monkeys jumping on the bed.
One fell off and broke his(or her) head.
Mama(or Daddy) called the doctor and the doctor said,
"No more monkeys jumping on the bed!"
...
...
1 little monkey jumping on the bed.
One fell off and broke his(or her) head.
Mama(or Daddy) called the doctor and the doctor said,
"No more monkeys jumping on the bed!"
 - ◆ Mama 혹은 Daddy를 랜덤하게 선택하게 하자.

Assignment #2-2: Python Crash Course 2

- ◆ #2. 시작하는 연도와 끝나는 연도를 물어보는 프로그램을 작성 하고, 그 사이에 있는 모든 윤년을 출력하라.

(시작하는 연도와 끝나는 연도가 윤년이면 그 해도 출력)

- ◆ 윤년은 4로 나누어짐 (예: 1984, 2012)
- ◆ 100으로 나누어지면 윤년이 아님 (예: 1800, 1900)
- ◆ 윤년은 400으로 나누어짐 (예: 1600, 2000)

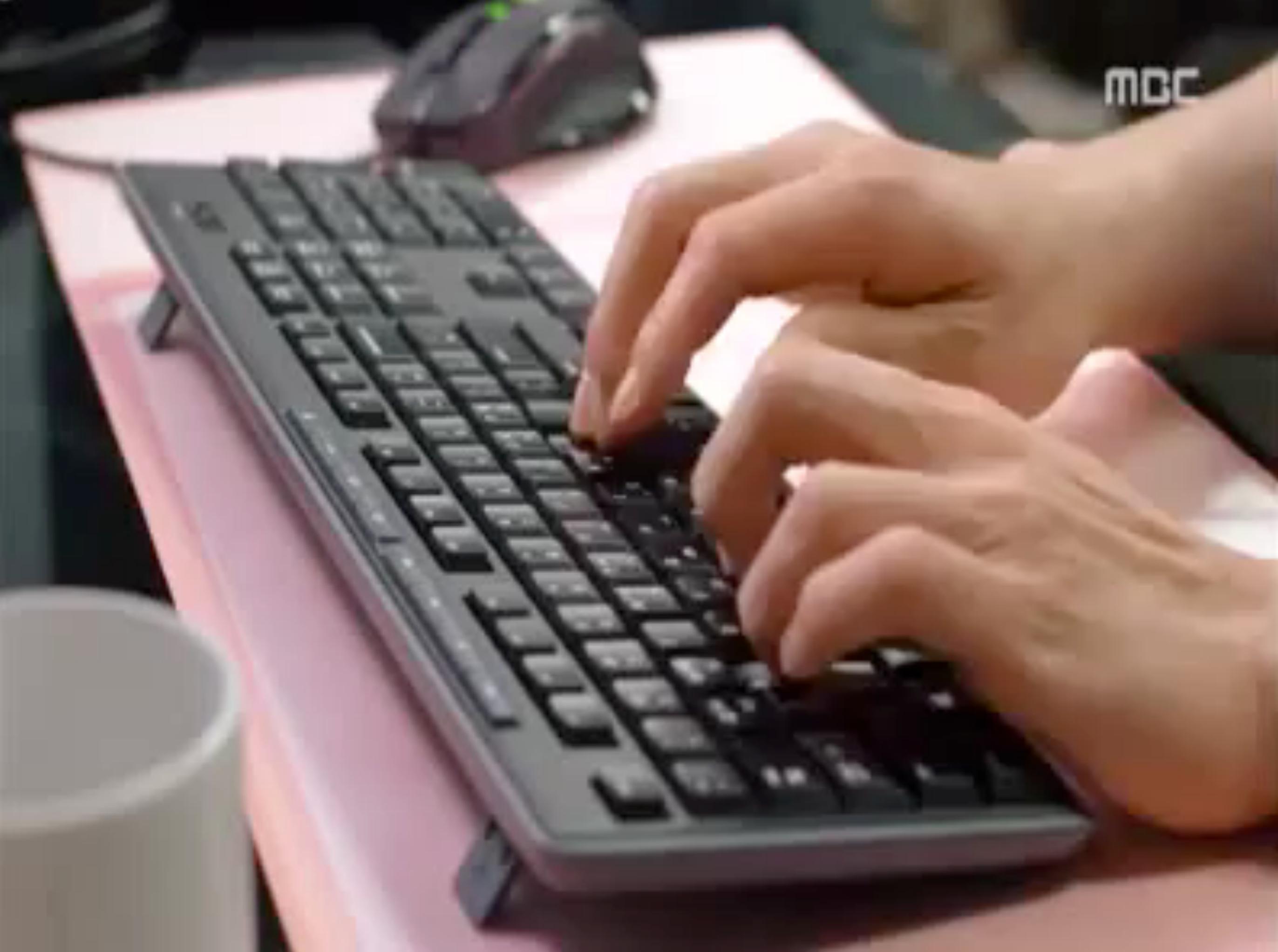
> Pick a starting year (like 1973 or something): => 1973

> Now pick an ending year:

=> 1977

> Check it out... these years are leap years: => 1976

- ◆ (참고) 윤년 리스트: <http://mwultong.blogspot.com/2005/12/18002300-leap-year-list.html>



3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

// 게임 로딩 부분

```
public class GuguClass {  
    public static void main(String[] args) {  
        System.out.format("%n < gugu>%n");  
        for (int j = 1; j <= 9; j++) {  
            System.out.println();  
            for (int i = 2; i <= 5; i++) { // 2~ 5  
                System.out.format("%d X %d = %2d ", i, j, i * j);  
            }  
        }  
        System.out.println(); // 종료  
        for (int j = 1; j <= 9; j++) {  
            System.out.println();  
        }  
    }  
}
```

Assignment #2-2: Python Crash Course 2

- ◆ #3. 구구단을 출력하는 프로그램을 작성하고 결과를 txt 파일로 저장하시오.

Questions?
