

Report #5

Introduction to Bigdata Programming STS2011-01

20180594 이정훈
경제학부

이번 과제는 날짜를 입력받아 윤년 여부를 판단하고, 요일을 출력하는 프로젝트였습니다. Library를 사용할 수 없었기 때문에 조금 까다로운 문제였으나, 조금만 생각하고 차근차근 풀면 쉽게 접근이 가능했습니다.

Problem 1.

- 년 월 일을 입력 받아서 요일을 구한다

```
In [5]: %%writefile date.py
def input_date() :
    year = int(input('__년도를 입력하시오 : '))
    month = int(input('__월을 입력하시오 : '))
    day = int(input('__일을 입력하시오 : '))
    return year, month, day

def is_leap(year) :
    if (year % 4 == 0) and (year % 100 != 0) or (year % 400 == 0) : return True
    else : return False

def get_day_name(year, month, day) :
    week = ['일요일', '월요일', '화요일', '수요일', '목요일', '금요일', '토요일']
    month_days = {1 : 31, 2 : 28, 3 : 31, 4 : 30, 5 : 31, 6 : 30,
                  7 : 31, 8 : 31, 9 : 30, 10 : 31, 11 : 30, 12 : 31}
    total_days = 365 * (year - 1) + sum([1 for i in range(1, year) if is_leap(i) == True])

    if is_leap(year) == True : month_days[2] = 29
    else :
        if (month, day) == (2, 29) : return False

    total_days += sum([month_days[i] for i in range(1, month)]) + sum([1 for i in range(day)
                                                                    if day <= month_days[month]])

    return week[total_days % 7]

if __name__ == '__main__' :
    year, month, day = input_date()
    day_name = get_day_name(year, month, day)

    if day_name == False : print('입력하신 날짜는 존재하지 않습니다.')
    else :
        print(day_name)
        if is_leap(year) == True : print('입력하신 %s은 윤년입니다' % year)

Overwriting date.py
```

```
In [2]: %run date.py

__년도를 입력하시오 : 1988
__월을 입력하시오 : 12
__일을 입력하시오 : 22
목요일
입력하신 1988은 윤년입니다
```

```
In [3]: %run date.py

__년도를 입력하시오 : 2019
__월을 입력하시오 : 2
__일을 입력하시오 : 29
입력하신 날짜는 존재하지 않습니다.
```

```
In [4]: %run date.py

__년도를 입력하시오 : 2021
__월을 입력하시오 : 3
__일을 입력하시오 : 23
화요일
```

함수 `input_date()`는 날짜를 입력하는 함수입니다. 함수를 실행시키면 `input`을 받는 지역변수가 존재하며, 입력을 마친 후에는 연,월,일 값을 반환합니다.

함수 `is_leap()`은 윤년 여부를 판단하는 함수입니다. 조건식을 이용해 4년과 400년의 배수가 되는 해는 윤년임을, 100년이 배수가 되는 해외 이외에는 윤년이 아닌것으로 분류를 가능하게 합니다.

함수 `get_day_name()`은 연, 월, 일을 입력받아 요일을 출력하는 함수입니다. 우선, `week`에는 요일을 미리 입력해 두었습니다. 변수 `month_dats`는 사전형 데이터로 월마다 몇일까지 있는지 기록해놓은 데이터입니다. 이를 이용해서, 입력한 년도 이전 년도까지의 총 일수를 `total_days`로 계산하게 했습니다.

다음은 입력한 년도가 윤년인지 아닌지 판단해야 합니다. 만약 윤년이라면 2월을 29일까지로 변경하고, 윤년이 아닌데 2월 29일이 입력되었다면 `False`를 반환하게 합니다.

다음으로는, 입력한 년도의 월, 일까지의 일수를 계산합니다. `sum()`과 `list comprehension`을 적절히 이용하여 한 줄에 담아내도록 노력하였습니다. 연산이 끝난 뒤에는, `total_days`를 7로 나눈 나머지만큼 `list indexing`을 시행하여 요일을 반환하게 합니다.

After work

• List Comprehension

- 한 줄 코드는 메모리를 효율적이게 하고, 코드가 깔끔해지는 효과를 기대할 수 있습니다. 이번 과제를 통해 이를 연습할 수 있게 되었습니다.

• 사전형 데이터의 처리

- 사전형은 변경 가능한 데이터이기에, 인덱스를 지정해 값을 변경하는 것 만으로 쉽게 이용을 할 수 있습니다. 다만, 원본 데이터가 변경되는 것이기 때문에 다룰 때에는 헛갈리지 않게 조심이 다뤄야 한다는 것을 깨닫게 되었습니다.