

Lab #8- Factorial 계산기 (factorial_calculator)

Copyright 2015 © document created by TeamLab.Gachon@gmail.com

Introduction

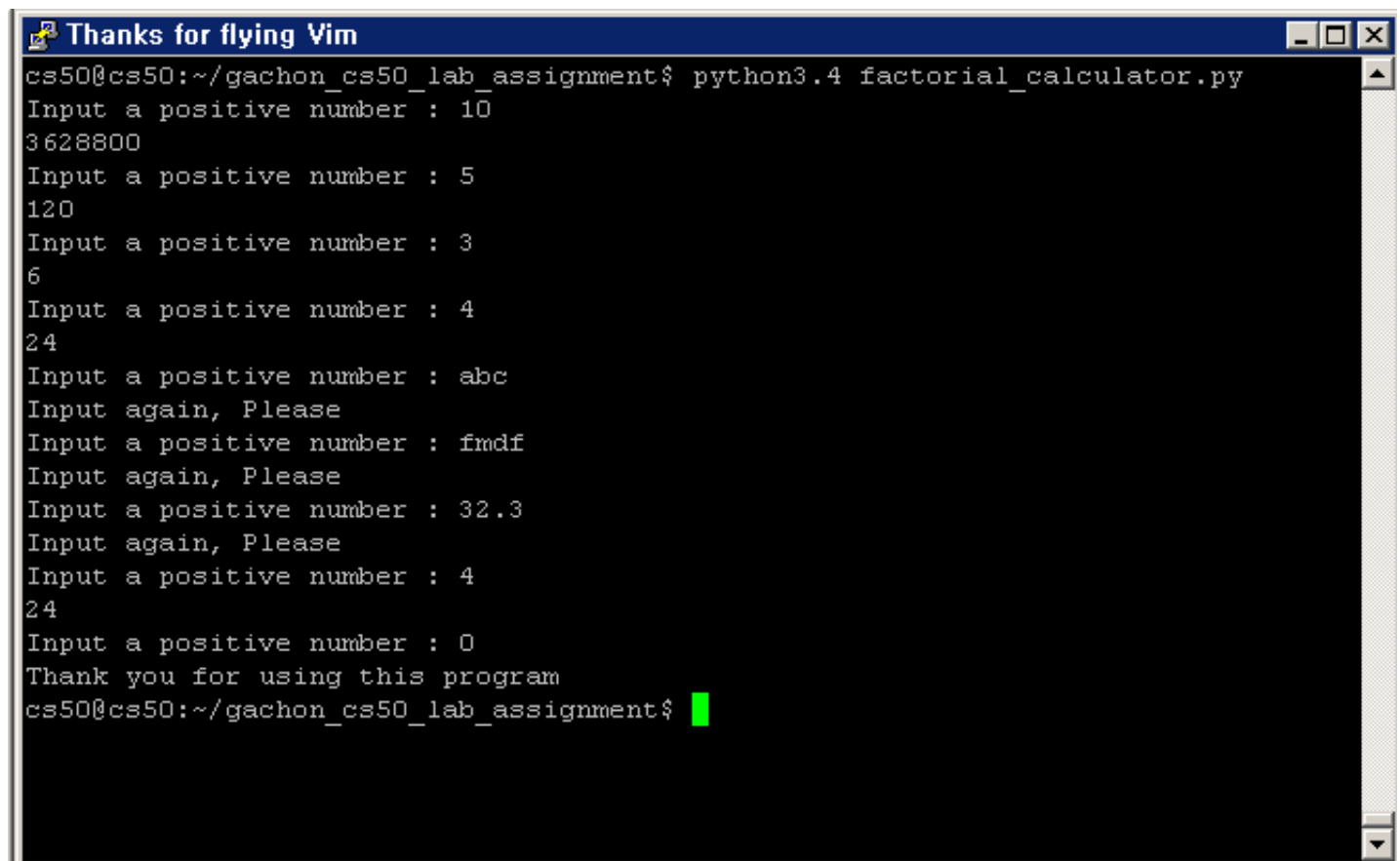
이번 Lab은 처음으로 main 함수에서 control을 다룬다. 이때까지 모든 Lab은 단순히 단위 함수를 수정하거나, 약간의 main 함수 수정으로 1회성으로 진행되는 프로그램만 작성해 보았다. 이번 Lab은 사용자가 특정한 입력을 하기 전까지 프로그램이 계속 수행되기 위하여 Loop구문과 if문을 활용한 main함수를 작성해 본다. 처음이라 상당히 어렵게 느껴질 수도 있는데, 이 역시 시간이 지나가서 보면 쉬운 Lab 중 하나 일거라는 생각이 들 것이다. 즐거운 마음으로 시작해보자.

Factorial Calculator Overview

먼저 이번 숙제의 목적인 Factorial 계산기에 대하여 알아보자. 한국어로는 "계승"으로도 번역되는 Factorial은 1 부터 n까지의 모든 자연수를 곱한 결과를 의미한다. 즉 n factorial은 $1 \times 2 \times 3 \times \dots \times n$ 의 의미이며, 숫자기호로는 n!로 표시한다. 만약 5! 인 경우는 $1 \times 2 \times 3 \times 4 \times 5$ 즉 120을 의미한다. 수학적으로 좀 유식하게 쓰면 아래처럼 표현하기도 한다(From wikipedia). 자세한 내용은 [wikipedia의 계승 페이지 \(https://ko.wikipedia.org/wiki/%EA%B3%84%EC%8A%B9\)](https://ko.wikipedia.org/wiki/%EA%B3%84%EC%8A%B9)를 참고하기 바란다.

$$n! = \prod_{k=1}^n k \quad \text{for all } n \geq 0$$

실제 우리가 구현해야할 프로그램은 아래 그림과 같다.



```
cs50@cs50:~/gachon_cs50_lab_assignment$ python3.4 factorial_calculator.py
Input a positive number : 10
3628800
Input a positive number : 5
120
Input a positive number : 3
6
Input a positive number : 4
24
Input a positive number : abc
Input again, Please
Input a positive number : fmdf
Input again, Please
Input a positive number : 32.3
Input again, Please
Input a positive number : 4
24
Input a positive number : 0
Thank you for using this program
cs50@cs50:~/gachon_cs50_lab_assignment$
```

본 프로그램은 다음과 같이 세 가지 규칙에 의해 실행된다.

1. 프로그램이 실행되어 사용자가 자연수를 입력하면 입력된 수의 Factorial 값을 계산하여 출력한다. 즉 사용자가 5라고 입력하면 120이 다음줄에 출력되어야 한다.
2. 사용자가 자연수와 0이외 다른 문자를 Input again, Please라는 글자가 출력되면서 다시 입력할 수 있도록 한다.
3. 사용자가 0을 입력하면 Thank you for using this program이라는 메시지를 출력하면서 종료한다.

로직 자체는 상당히 간단하고, 이미 배운 내용들로 충분히 구현할 수 있다. 하지만, 처음 해보는 Control 숙제이기 때문에 상당히 어렵게 느껴질 것이다. 이번 Lab은 조금 자세히 설명하니 꼭 문서를 정독하길 바란다.

숙제 template 파일 다운로드

먼저 숙제 template 파일을 cs50 서버로 부터 다운로드 받는다. 로그인 후 나타나는 bash shell에서 다음과 같은 명령을 입력하자.

```
python3.4 submit_assignment.py -get factorial_calculator
```

입력되면 다운로드 안내 메시지와 함께 factorial_calculator.py 파일이 다운로드 된다.

factorial_calculator.py 파일 Overview

vim editor로 factorial_calculator.py을 열어 전체적인 개요를 보자. vi fahrenheit_converter.py명령으로 파일을 열어보면 main 함수와 is_positive_number함수가 존재할 것이다. 본 Lab에서 수정 및 추가해야할 함수는 아래 세 가지이다.

함수	설명
is_positive_number	문자열로 된 값을 입력받아, 해당 값이 자연수로 변환이 될 경우에는 True를 반환하고 변환이 안될 경우에는 False를 반환한다.
get_factorial_value	자연수를 입력받아 해당 자연수의 Factorial 값을 계산하여 반환한다. 즉 5를 입력받으면 120을 반환한다.
main	사용자가 값을 입력받아 Factorial 값을 화면에 출력하도록 한다. 앞장 "Factorial Calculator Overview" 적힌 규칙에 따라 0을 입력하면 프로그램을 종료하고, 자연수가 아닌 값을 입력하면 다시 입력하도록 요청한다.

각 함수별로 작성하는 방법을 살펴보자.

is_positive_number 함수 수정하기

첫 번째 함수는 is_positive_number 함수이다. 이미 template이 아래와 같이 작성되어 있다. 실제 코드에는 주석이 달려있지만, 설명을 위해 아래에는 생략했다.

```
def is_positive_number(integer_str_value):
    # 주석생략
    try:
        # ===Modify codes below=====
        # 시작전 반드시 'pass'를 지울 것
        pass

        # =====
    except ValueError:
        return False
```

여러분이 수정해야할 영역은 pass라고 적힌 부분이다. 보면 try ~ except ValueError 라는 부분이 보이는데 현재는 알 필요가 없는 "Exception Handling" 구문이다. Lab 작성을 위해 기본 template을 제시했으니 신경쓰지 말길 바란다. 본 함수에서의 역할은 형변환이 불가능할 경우 무조건 False를 반환하게 하는 것이다. 예를 들어 "abc" 라고 입력된 문자열 값을 int("abc") 라고 시도할 경우 무조건 본 함수에서 False를 return 하게 한다. 본 함수의 기본 목적은 사용자가 입력한 값이 자연수인지 아닌지 확인 하는 것이다. 본 함수는 integer_str_value로 입력된다. pass 부분을 지우고 indentation에 맞춰서 integer_str_value가 자연수인지 아닌지 확인하는 조건문을 if를 사용하여 작성해보자. 물론 사용자가 입력하는 값은 문자열이기 때문에 해당 값을 먼저 integer 값으로 변환해주어야 한다. 이미 알고 있겠지만 자연수라 함은 "1이상인 정수"를 의미한다. 다시 정리하면 본 함수에서는

1. 입력된 integer_str_value를 intger type값으로 변환한다.
2. 변환된 값이 자연수 인지 아닌지 확인한다.
3. 자연수 일 경우 True를, 아닐 경우 False를 반환한다.

integer로 변환이 불가능할 경우, try ~ catch 문에 의해 자동으로 False를 반환하니 신경쓰지 말자. 위의 목적에 맞게 코드를 작성해 보자.

get_factorial_value 함수 작성하기

두 번째 함수는 get_factorial_value 이다. 본 함수는 integer_value 라는 자연수 값을 입력받아, 해당 값의 factorial 값을 반환한다. 이미 자연수로 변환된 값만 입력받기 때문에 위 is_positive_number 함수처럼 입력된 값에 대한 확인을 할 필요가 없다. Factorial 값을 구하기 위해서는 많은 방법이 있지만, for문을 활용하는 것이 가장 좋을 것으로 생각된다. for문을 사용하면 1부터 integer_value 까지를 모두 곱하는 것이 가장 쉽다. 한가지 조심해야 하는 것은 integer_value을 사용하여 for문을 작성할 경우, 흔히 활용하는 range(integer_value)구문을 쓰면 0부터 값이 시작되기 때문에 range(1,integer_value+1)로 작성해줘야 한다. 물론 range 구문을 쓰는 것 말고도 factorial 값을 구할 수 있는 방법은 무궁무진 하다. 수강자가 원하는 어떤 방법을 사용하든 입력된 integer_value의 Facotrial 값만 반환해주면 된다.

main 함수 수정하기

마지막 함수는 main함수이다. main함수의 template는 아래와 같다.

```
def main():
    user_input = 999
    # ===Modify codes below=====

    # =====
```

참고로 본 Lab에서 설명하는 방법은 많은 구현 방법중 하나일 뿐이다. 스스로 구현하는 방법이 있다면 그 방법대로 하면 된다. 아래 설명은 구현에 대한 대략적인 설명이다. loop문이나 if문에 대해 이해가 없이 시작하면 이해가 불가능하다. 설명을 이해한 후 스스로 코드를 작성해 보기 바란다. main함수의 시작은 user_input = 999이다. user_input은 사용자가 입력한 값을 할당받는 변수이다. 만약 user_input이 필요없다고 생각되면 지워도 전혀 문제가 없다. user_input = 999이 이유는 while문에 진입하기 위해서다. 아래 설명에도 나오지만 본 lab에서는 while문에 종료조건은 user_input이 0인 경우이다. 제일 처음 시작을 위해 user_input에 999를 할당하였다. 이미 설명이 된 부분이지만, Loop 구문에서 종료를 해야하는 횟수가 정해져 있지 않다면 while 문을 쓰는 것이 좋다. 여기에선 "사용자가 0을 입력하면 종료"라는 조건이 있기 때문에 while(user_input is not 0): 이라는 구문으로 시작하면 좋을 것이다. while문을 실행한 후 처음 할 일은 사용자에게 입력을 받는 것이다. 입력을 받을 때는 input("Input a positive number : ") 문을 사용하면 되고, 입력된 값은 user_input 변수에 할당한다. 다음으로 입력된 값이 factorial 값을 계산할 수 있는지 확인하기 위해 if문과 is_positive_number 함수를 사용한다. if is_positive_number(user_input): 와 같이 쓰면 입력된 값이 자연수일 때는 True를, 아닐 경우는 False를 반환하여 분기를 수행한다.

- 반환된 값이 True일 경우 user_input 값을 integer 값으로 변환한 후, get_factorial_value 함수를 사용하여 user_input의 factorial 값을 반환받고 화면에 출력한다.
- 반환된 값이 False일 경우, 해당 값이 '0' 인지 확인하여 0일 경우, 프로그램을 종료한다. 이를 위해서는 elif user_input is '0': 과 같은 구문을 쓰고, 본 조건에서 user_input = 0 으로 할당해주면 while 문의 종료 조건에 의해 프로그램이 종료될 것이다. 물론 종료되기 전에 Thank you for using this program라는 문자열을 반드시 화면에 출력해주자.
- 반환된 값이 False이면서도, user_input이 0이 아니면 화면에 Input again, Please라는 문자열을 화면에 출력해주면 된다. 이때 위의 모든 조건이 아닐 경우라는 의미로 else:문을 사용하면 된다. user_input이 '0'만 아니라면 프로그램은 계속 동작한다.

위의 설명은 상당히 복잡하다. 여러분들이 if, while 문에 대해서 기본적인 이해가 부족하다면 실제로 구현하기 어려울 것이다. 반드시 강의 자료를 복습하고 실제 구현을 해보길 바란다.

결과 출력하기

실제 코드가 다 작성되어 python3.4 factorial_calculator.py 아래와 같이 결과를 볼 수 있을 것이다. 당연히 입력 부분은 수강자가 직접 입력을 해주어야 프로그램 진행된다.

```
Input a positive number : 10
3628800
Input a positive number : 3
6
Input a positive number : 5
120
Input a positive number : abc
Input again, Please
Input a positive number : ls
Input again, Please
Input a positive number : 32.3
Input again, Please
Input a positive number : 0
Thank you for using this program
```

숙제 제출하기

모든 함수를 다 수정했다면, 아래와 같이 제출하자

```
python3.4 submit_assignment.py -submit factorial_calculator.py
```

제대로 작성했다면 아래와 같은 메시지가 뜰 것이다.

```
----- | ----- | -----
```

Function Name	Passed?	Feedback
get_factorial_value	PASS	Good Job
main	PASS	Good Job
is_positive_number	PASS	Good Job

Next Work

축하한다. 여러분은 처음으로 프로그램처럼 돌아가는 프로그램을 만들었다. 여러분이 사용하는 웹 브라우저, 엑셀, 파워포인트 등 모든 프로그램에는 본 Lab에서 나오는 while, if, for문들이 사용된다. 어렵게 느껴졌을 지도 모르겠지만, 정말 쉬운 숙제였다. 믿어라. 다음 주는 지옥을 보게 될 것이다.

Human knowledge belongs to the world - from movie 'Password' -

Footnotes