

Lab - Factorial 계산기 (factorial_calculator)

Copyright 2017 © document created by TeamLab.Gachon@gmail.com

Introduction

이번 Lab은 처음으로 main 함수에서 control을 다룹니다. 이때까지 모든 Lab은 단순히 단위 함수를 수정하거나, 약간의 main 함수 수정으로 1회성으로 진행하는 프로그램만 작성했습니다. 이번 Lab은 사용자가 특정한 입력을 하기 전까지 프로그램이 계속 수행되기 위하여 Loop구문과 if문을 활용한 main함수를 작성합니다. 처음이라 상당히 어렵게 느껴질 수도 있는데, 이 역시 시간이 지나가서 보면 쉬운 Lab 중 하나 일거라는 생각이 들 것 입니다. 즐거운 마음으로 시작해 봅시다.

[PDF 파일 다운로드](#)

숙제 파일(lab_6.zip) 다운로드

먼저 해야 할 일은 숙제 파일을 다운로드 받는 것 입니다. 이미 해보았기 때문에 어렵지 않을 것입니다. Chrome 또는 익스플로러와 같은 웹 브라우저 주소창에 아래 주소를 입력합니다.

https://github.com/TeamLab/introduction_to_python_TEAMLAB_MOOC/blob/master/lab_assignment/lab_6/lab_6.zip

다운로드를 위해 `View Raw` 또는 `Download` 버튼을 클릭합니다. 또는 아래 다운로드 링크를 클릭하면 자동으로 다운로드가 됩니다. [Lab 6 - 다운로드](#)

다운로드 된 `lab_6.zip` 파일을 작업 폴더로 이동한 후 압축해제 후 작업하길 바랍니다.

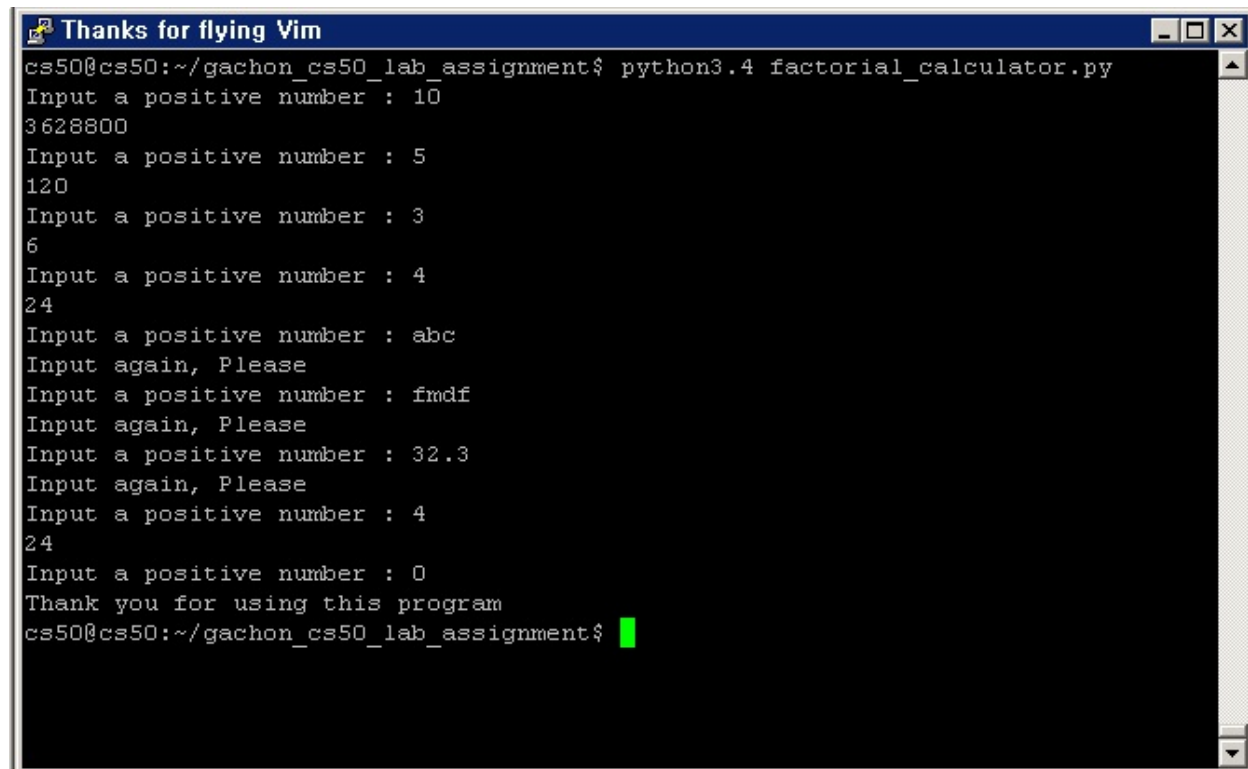
압축해제 하면 폴더가 `linux_mac` 과 `windows` 로 나뉘져 있습니다. 자신의 OS에 맞는 폴더로 이동해서 코드를 수정해 주시기 바랍니다.

Factorial Calculator Overview

먼저 이번 숙제의 목적인 Factorial 계산기에 대하여 알아보면, 한국어로는 "계승"으로도 번역되는 Factorial은 1 부터 n 까지의 모든 자연수를 곱한 결과를 의미합니다. 즉 n factorial은 $1 \times 2 \times 3 \times \dots \times n$ 의 의미이며, 숫자기호로는 $n!$ 로 표시합니다. 만약 $5!$ 인 경우는 $1 \times 2 \times 3 \times 4 \times 5$ 즉 120 을 의미합니다. 수학적으로 좀 유식하게 쓰면 아래처럼 표현하기도 합니다(From wikipedia). 자세한 내용은 [wikipedia의 계승 페이지](#)를 참고하시기 바랍니다.

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

실제 수강생이 구현해야할 프로그램은 아래 그림과 같습니다.



```

Thanks for flying Vim
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 실행됩니다.

1. 프로그램이 실행되어 사용자가 자연수를 입력하면 입력된 수의 Factorial 값을 계산하여 출력합니다. 즉 사용자가 5 라고 입력하면 120 이 다음줄에 출력되어야 합니다.
2. 사용자가 자연수와 0이외 다른 문자를 Input again, Please 라는 글자가 출력되면서 다시 입력할 수 있도록 합니다.
3. 사용자가 0을 입력하면 Thank you for using this program 이라는 메시지를 출력하면서 종료합니다.

로직 자체는 상당히 간단하고, 이미 배운 내용들로 충분히 구현할 수 있습니다. 하지만, 처음 해보는 Control 숙제이기 때문에 상당히 어렵게 느껴질 것입니다. 이번 Lab은 조금 자세히 설명하니 꼭 문서를 정독하시길 바랍니다.

factorial_calculator.py 파일 Overview

atom 으로 factorial_calculator.py 을 열어 전체적인 개요를 봅시다.

파일을 열어보면 main 함수와 is_positive_number , get_factorial_value 함수가 존재할 것입니다.

본 Lab에서 수정해야할 함수 목록은 아래와 같습니다.

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

각 함수별로 작성하는 방법을 살펴봅시다.

is_positive_number 함수 수정하기

첫 번째 함수는 `is_positive_number` 함수입니다. 이미 template이 아래와 같이 작성되어 있습니다. 실제 코드에는 주석이 달려있지만, 설명을 위해 아래에는 생략했습니다.

```
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 `pass` 라고 적힌 부분입니다. 보면 `try ~ excpet ValueError` 라는 부분이 보이는데 현재는 알 필요가 없는 "Exception Handling" 구문입니다. Lab 작성을 위해 기본 template을 제시했으니 신경쓰지 마시길 바랍니다. 본 함수에서의 역할은 형변환이 불가능할 경우 무조건 False를 반환하게 하는 것입니다. 예를 들어 `"abc"` 라고 입력된 문자열 값을 `int("abc")` 라고 시도할 경우 무조건 본 함수에서 False를 return 하게 합니다.

본 함수의 기본 목적은 사용자가 입력한 값이 자연수인지 아닌지 확인 하는 것입니다. 본 함수는

`integer_str_value` 로 입력됩니다. `pass` 부분을 지우고 indentation에 맞춰서

`integer_str_value` 가 자연수인지 아닌지 확인하는 조건문을 `if` 를 사용하여 작성해보기 바랍니다. 물론 사용자가 입력하는 값은 문자열이기 때문에 해당 값을 먼저 integer 값으로 변환해주어야 합니다. 이미 알고

있겠지만 자연수라 함은 "1이상인 정수"를 의미합니다.
다시 정리하면 본 함수에서는

1. 입력된 `integer_str_value`를 `intger` type값으로 변환합니다.
2. 변환된 값이 자연수 인지 아닌지 확인합니다.
3. 자연수 일 경우 `True`를, 아닐 경우 `False`를 반환합니다.

`integer`로 변환이 불가능할 경우, `try ~ catch` 문에 의해 자동으로 `False` 를 반환하니 신경쓰지 마시길 바랍니다. 위의 목적에 맞게 코드를 작성해 보시다.

get_factorial_value 함수 작성하기

두 번째 함수는 `get_factorial_value` 입니다. 본 함수는 `integer_value` 라는 자연수 값을 입력받아, 해당 값의 factorial 값을 반환합니다. 이미 자연수로 변환된 값만 입력받기 때문에 위

`is_positive_number` 함수처럼 입력된 값에 대한 확인을 할 필요가 없습니다.

Factorial 값을 구하기 위해서는 많은 방법이 있지만, `for`문을 활용하는 것이 가장 좋을 것으로 생각됩니다.

`for`문을 사용하면 1부터 `integer_value` 까지를 모두 곱하는 것이 가장 쉽습니다. 한가지 조심해야 하는 것은 `integer_value`을 사용하여 `for`문을 작성할 경우, 흔히 활용하는 `range(integer_value)` 구문을 쓰면 0 부터 값이 시작되기 때문에 `range(1, integer_value+1)` 로 작성해줘야 합니다. 물론 `range` 구문을 쓰는 것 말고도 factorial 값을 구할 수 있는 방법은 무궁무진 합니다. 수강생이 원하는 어떤 방법을 사용하든 입력 된 `integer_value` 의 Facotrial 값만 반환해주면 됩니다.

main 함수 수정하기

마지막 함수는 `main` 함수입니다. `main` 함수의 template는 아래와 같습니다.

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

    # =====
```

참고로 본 Lab에서 설명하는 방법은 많은 구현 방법중 하나일 뿐입니다. 스스로 구현하는 방법이 있다면 그 방법대로 하셔도 됩니다. 아래 설명은 구현에 대한 대략적인 설명입니다. `loop`문이나 `if`문에 대해 이해가 없이 시작하면 이해가 불가능합니다. 설명을 이해한 후 스스로 코드를 작성해 보시기 바랍니다.

`main` 함수의 시작은 `user_input = 999` 입니다. `user_input` 은 사용자가 입력한 값을 할당받는 변수입니다. 만약 `user_input` 이 필요없다고 생각되면 지워도 전혀 문제가 없습니다. `user_input = 999` 이 이유는 `while`문에 진입하기 위해서 입니다. 아래 설명에도 나오지만 본 lab에서는 `while`문에 종료조건은 `user_input` 이 0인 경우입니다. 제일 처음 시작을 위해 `user_input` 에 999 를 할당하였습니다.

이미 설명이 된 부분이지만, `Loop` 구문에서 종료를 해야하는 횟수가 정해져 있지 않다면 `while` 문을 쓰는

것이 좋습니다. 여기에선 "사용자가 0을 입력하면 종료" 라는 조건이 있기 때문에 `while(user_input is not 0):` 이라는 구문으로 시작하면 좋을 것입니다.

`while` 문을 실행한 후 처음 할 일은 사용자에게 입력을 받는 것입니다. 입력을 받을 때는 `input("Input a positive number : ")` 문을 사용하면 되고, 입력된 값은 `user_input` 변수에 할당합니다.

다음으로 입력된 값이 factorial 값을 계산할 수 있는지 확인하기 위해 `if`문과 `is_positive_number` 함수를 사용합니다. `if is_positive_number(user_input):` 와 같이 쓰면 입력된 값이 자연수일 때는 `True` 를, 아닐 경우는 `False` 를 반환합니다.

- 반환된 값이 `True` 일 경우 `user_input` 값을 `integer` 값으로 변환한 후, `get_factorial_value` 함수를 사용하여 `user_input`의 factorial 값을 반환받고 화면에 출력합니다.
- 반환된 값이 `False` 일 경우, 해당 값이 `'0'` 인지 확인하여 0일 경우, 프로그램을 종료합니다. 이를 위해서는 `elif user_input is '0':` 과 같은 구문을 쓰고, 본 조건에서 `user_input = 0` 으로 할당 해주면 `while` 문의 종료 조건에 의해 프로그램이 종료될 것입니다. 물론 종료되기 전에 `Thank you for using this program` 라는 문자열을 반드시 화면에 출력해줘야 합니다.
- 반환된 값이 `False` 이면서도, `user_input` 이 `0` 이 아니면 화면에 `Input again, Please` 라는 문자열을 화면에 출력해주면 됩니다. 이때 위의 모든 조건이 아닐 경우라는 의미로 `else:` 문을 사용하면 됩니다. `user_input` 이 `'0'`만 아니라면 프로그램은 계속 동작합니다.

위의 설명은 상당히 복잡합니다. 여러분들이 `if`, `while` 문에 대해서 기본적인 이해가 부족하다면 실제로 구현하기 어려울 것입니다. 반드시 강의자료를 복습하고 실제 구현을 해보길 바랍니다.

결과 출력하기

실제 코드가 다 작성되어 `python factorial_calculator.py` 아래와 같이 결과를 볼 수 있을 것입니다. 당연히 입력 부분은 수강생이 직접 입력을 해주어야 프로그램이 진행됩니다.

```
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 lab assignment가 종료되었습니다. 이제 숙제를 제출합니다.

###숙제 template 파일 제출하기 (윈도우의 경우)

1. `windows` + `r` 를 누르고 cmd 입력 후 확인을 클릭합니다.
2. 작업을 수행한 폴더로 이동 합니다.
3. 밑에 명령어를 cmd창에 입력합니다.

```
submit.bat [YOUR_HASH_KEY]
```

숙제 template 파일 제출하기 (Mac or Linux)

1. 터미널을 구동합니다.
2. 작업을 수행한 디렉토리로 이동 합니다.
3. 밑에 bash창을 입력합니다.

```
./submit.sh [YOUR_HASH_KEY]
```

참고로 lab assignment 제출은 진행 중간중간해도 문제가 없습니다. 제대로 작성되었다면 아래와 같은 메시지를 확인할 수 있을 것입니다.

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

Next Work

축하합니다. 여러분은 처음으로 프로그램처럼 돌아가는 프로그램을 만들었습니다. 여러분이 사용하는 웹 브라우저, 엑셀, 파워포인트 등 모든 프로그램에는 본 Lab에서 나오는 `while` , `if` , `for` 문들이 사용됩니다. 어렵게 느껴졌을 지도 모르겠지만, 정말 쉬운 숙제였습니다. 믿으셔도 됩니다. 다음 숙제는 지옥을 보게 될지도 모릅니다.

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

Footnotes
