

Lab #7 - list, control, loop 연습 (gowithflow)

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

Introduction

3주차임에도 불구하고 16주차처럼 느껴진다면 기분 탓이다. 기분 좋게 Lab 6를 시작하는 여러분들을 환영한다. 굉장히 어렵고 힘들게 느껴지겠지만, 실은 기초중에 기초를 하고 있다는 것에 좌절하지 않길 바란다. 혹시 인스타그램을 쓰는가? 참고로 인스타그램은 파이썬으로 개발된 대표적인 서비스이다. 언젠가 그런 서비스를 개발할 날을 꿈꾸며, 오늘의 Lab을 시작하자. 이번 차시의 Lab은 이미 배운 list, if문, for문과 while문 등을 연습한다. 기본적인 형태는 이미 경험해본 Lab #4 - basic_operations와 같다. 다양한 함수들이 존재하고 각 함수에 목적에 맞게 수정하면 된다. 난이도가 점점 올라간다. 그럼에도 불구하고 아직 기초라는 사실을 잊지 말고, Lab을 즐기길 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 gowithflow
```

정상적으로 다운했다면 현재 디렉토리에 gowithflow.py 파일 생성되었을 것이다. ls 명령어로 확인하자.

수정 해야할 함수 종류들

숙제 파일을 다운로드 후 vi gowithflow.py 명령어를 입력하여 숙제 파일을 살펴보자. 참고로 본 강의의 Lab 구성은 Lab #5나 Lab #6처럼 main() 함수 자체를 수정해서 전체 프로그램 목적에 맞게 프로그램을 작성하는 Lab도 있고, 본 Lab #6나 Lab #5처럼 단위 함수를 수정하여 단순히 기능들을 연습하는 Lab도 있다. 단순히 기능 연습만으로 프로그램을 짤 수 없고, 로직에 대한 이해가 어렵기 때문에 보통 한 차수의 Lab은 두개이상이다. 애정이 있어서 Lab이 많은 거니 너무 화를 내지 말자. 본 Lab에서 수정해야할 함수 목록은 아래와 같다.

함수	설명
sum_of_list	숫자 값으로 구성된 list를 입력받아 list element들의 합을 반환한다.
merge_and_sort	숫자 또는 문자로만 구성된 두 개의 list를 입력받아, 두 list를 합치고 정렬한 후 반환한다.
delete_a_list_element	list와 기본 데이터 Type 값을 입력받아, 입력받은 기본 데이터 값이 list에 포함되어 있다면 해당 값이 제거된 list를, 포함되어 있지 않다면 Integer Type의 0 을 반환한다.
comparison_list_size	두 개의 list를 입력받아, 두 list의 길이(크기)를 비교하고 둘 중 큰 리스트를 반환한다.
odd_even_check	두 개의 Integer Type 값을 입력받아, 두 값의 합이 짝수면 'Even', 홀수이면 'Odd'의 문자열을 반환한다.
discount_price	물건의 가격에 해당하는 price를 숫자형 값을 입력받아, 해당 값이 100,000 미만이면 10% 할인된 값을, 100,000 이상이면 20% 할인된 값을 반환한다.
find_smallest_value	숫자형 값으로 이루어진 list를 입력받아, 가장 작은 element를 반환한다.
binary_converter	양수 Integer Type의 값을 입력받아, 해당 값을 2진수로 바꾼 문자열 Type의 값을 반환한다.
number_of_cases	숫자 또는 문자 값으로 이루어진 list를 입력받아, 해당 list의 element 값들 조합의 모든 경우의 수를 반환한다. 단 이때 중복되는 조합은 제거한다.

수정후 테스트 하기

제시된 각 함수를 위의 목적에 맞게 수정한 후 테스트를 실시해야 한다. 이번 부터는 특별한 테스트 코드를 제공해주지 않는다. 수강자가 직접 python shell이나 main함수에 수정 코드를 작성하여 실행해보기 바란다. 단, 테스트 코드예제는 각 함수의 주석(comment) 형태로 달려 있으니, 확인해 보기 바란다.

수정전 알아야 하는 것들

막상 해보면 어렵진 않지만, 하기전에는 어려운 것들이 있다. 예를 들면, 파이썬에서 list간의 merge를 위해서는 아래와 같은 코드를 사용할 수 있다.

```
a = [1, 2, 3, 4, 5]
b = ['a', 'b', 'c', 'd', 'e']
c = a + b
```

저 방법외에 a.extend(b)를 하면 a 변수에 b list가 합쳐진다. 특정한 값이 해당 리스트에 존재하는지 확인하기 위해서는 if value in list_data 같은 표현을 쓸 수 있다. value라는 값이 list_data라는 list type에 들어가있는지 확인하는 구문이다. 당연히 if문 형식이기 때문

에 indentation을 사용하여 실행 명령을 아래 적어줘야 한다. 아래와 같이 코드를 작성할 수 있다.

```
result = []
if not(value in list_data):
    result.append(value)
```

위의 코드는 value값이 list_data list에 들어가 있지 않으면 result list에 value의 값을 추가하는 명령이다. 중요한 예시이니 꼭 이해하고 넘어가자. list에서 특정한 값을 지우기 위해서는 list_data.remove(특정한값) 을 쓸 수 있다. 혹시나 노파심에 하는 말인데, 저 명령문에서 list_data는 list type의 변수명을 의미하면 특정한값은 지우고자 하는 값을 얘기한다. 예를 들어 0을 지우고 싶으면, 저기 특정한값에 0을 적어주면 된다. list에 새로운 값을 추가하는 방법은 list_data.append(추가하는값)이다. 물론 문자열일때는 '추가하는값'을 붙여야함을 잊지말자 제일 작은 값을 찾는 방법은 min(list_data) 이고, 특정한 값이 list에 존재하는 갯수를 찾을 때는 list_data.count(특정한값)이다. 지금 설명하는 내용은 number_of_cases에 필요한 내용들이므로 숙제를 하기 바란다.

숙제 제출하기

모든 함수를 다 수정했다면, 아래와 같이 제출하자

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

제대로 작성했다면 아래와 같은 메시지가 뜰 것이다.

Function Name	Passed?	Feedback
find_smallest_value	PASS	Good Job
sum_of_list	PASS	Good Job
comparison_list_size	PASS	Good Job
binary_converter	PASS	Good Job
merge_and_sort	PASS	Good Job
discount_price	PASS	Good Job
odd_even_check	PASS	Good Job
number_of_cases	PASS	Good Job
delete_a_list_element	PASS	Good Job

아직도 몇몇 분들은 제출하기 전에 해당 코드를 테스트해보지 않고 제출한다. unsupported error가 가끔 나는데, 코드 자체가 분석 불가능할 경우나 에러로 얼마나 수강자가 숙제에 무심한지 티를 내는 것이다. 반드시 bash shell에서 python3.4 gowithflow.py 명령으로 실행을 해보고 문제가 없을 경우에만 제출하도록 하자. 물론, 위의 명령을 실행하기 위해서는 자기 나름대로 테스트 코드를 만들어 봐야 한다.

Next Work

이번 숙제는 사람에 따라 굉장히 어렵게 느끼기도 했을 것이다. 그 이유는 처음으로 로직이 숙제에 들어갔기 때문이다. 거의 모든 컴퓨터 프로그램은 로직에 의해 움직인다. 조금 난해하더라도 반드시 스스로의 힘으로 작성해야 머리에 남을 것이다. 이번 차시에는 Lab은 두개밖에 없다. 한 개만 더해보자.

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

Footnotes