

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

---

Copyright 2017 © document created by [TeamLab.Gachon@gmail.com](mailto:TeamLab.Gachon@gmail.com)

## Introduction

---

4주차 임에도 불구하고 16주차처럼 느껴진다면 기분 탓일 겁니다. 기분 좋게 Lab을 시작하는 여러분들을 환영합니다. 굉장히 어렵고 힘들게 느껴지겠지만, 실은 기초중에 기초를 하고 있다는 것에 좌절하지 않길 바랍니다.

혹시 인스타그램을 쓰십니까? 참고로 인스타그램은 파이썬으로 개발된 대표적인 서비스입니다. 언젠가 그런 서비스를 개발할 날을 꿈꾸며, 오늘의 Lab을 시작합니다.

이번 차시의 Lab은 이미 배운 list, if문, for문과 while문 등을 연습합니다. 기본적인 형태는 이미 경험해본 Lab #2 - basic\_operations 와 같습니다. 다양한 함수들이 존재하고 각 함수에 목적에 맞게 수정하면 됩니다.

난이도가 점점 올라갑니다. 그럼에도 불구하고 아직 기초라는 사실을 잊지 말고, Lab을 즐기길 바랍니다.

[PDF 파일 다운로드](#)

## 숙제 파일(lab\_5.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\\_5/lab\\_5.zip](https://github.com/TeamLab/introduction_to_python_TEAMLAB_MOOC/blob/master/lab_assignment/lab_5/lab_5.zip)

다운로드를 위해 `View Raw` 또는 `Download` 버튼을 클릭합니다. 또는 아래 다운로드 링크를 클릭하면 자동으로 다운로드가 됩니다. [Lab 5 - 다운로드](#)

다운로드 된 lab\_5.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.

## 수정 해야할 함수 종류들

---

숙제 파일을 다운로드 후 `atom` 으로 숙제 파일을 살펴 보시다.

참고로 본 강의의 Lab 구성은 Lab #3나 Lab #4처럼 `main()` 함수 자체를 수정해서 전체 프로그램 목적에 맞게 프로그램을 작성하는 Lab도 있고, 본 Lab #5나 Lab #2 처럼 단위 함수를 수정하여 단순히 기능들을 연습하는 Lab도 있습니다. 단순히 기능 연습만으로 프로그램을 짤 수 없고, 로직에 대한 이해가 어렵기 때문에 보통 Lab이 난해해지기 시작합니다. 애정이 있어서 Lab이 많은 거니 너무 화를 내지는 맙시다.

본 Lab에서 수정해야할 함수 목록은 아래와 같습니다.

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

## 수정후 테스트 하기

제시된 각 함수를 위의 목적에 맞게 수정한 후 테스트를 실시해야 합니다. 이번 부터는 특별한 테스트 코드를 제공하지 않습니다. 그러므로 수강자가 직접 `python shell` 이나 `main` 함수에 수정 코드를 작성하여 실행

해보시길 바랍니다. 단, 테스트 코드예제는 각 함수의 주석(comment) 형태로 달려 있으니, 확인해 보시길 바랍니다.

## 수정전 알아야 하는 것들

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

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

저 방법외에 `a.extend(b)` 를 하면 a list변수에 b list가 합쳐집니다.

특정한 값이 해당 리스트에 존재하는지 확인하기 위해서는 `if value in list_data` 같은 표현을 쓸 수 있습니다. `value` 라는 값이 `list_data` 라는 list type에 들어가있는지 확인하는 구문입니다. 당연히 if문 형식이기 때문에 `indentation` 을 사용하여 실행 명령을 아래 적어줘야 합니다. 아래와 같이 코드를 작성할 수 있습니다.

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

위의 코드는 `value` 값이 `list_data` list에 들어가 있지 않으면 `result` list에 `value` 의 값을 추가하는 명령입니다. 중요한 예시이니 꼭 이해하고 넘어갑시다.

list에서 특정한 값을 지우기 위해서는 `list_data.remove(특정한값)` 을 쓸 수 있습니다. 이 명령문에서 `list_data` 는 list type의 변수명을 의미하며 `특정한값` 은 지우고자 하는 값을 얘기합니다. 예를 들어 0 을 지우고 싶으면, `특정한값` 에 0 을 적어주면 됩니다.

list에 새로운 값을 추가하는 방법은 `list_data.append(추가하는값)` 입니다. 물론 문자열일때는 '추가하는 값' 을 붙여야함을 잊지 맙시다.

제일 작은 값을 찾는 방법은 `min(list_data)` 이고, 특정한 값이 list에 존재하는 갯수를 찾을 때는 `list_data.count(특정한값)` 입니다. 지금 설명하는 내용은 `number_of_cases` 에 필요한 내용들이므로 숙제를 하기 바랍니다.

###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]
```

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 제출합니다. `unsupported error` 가 가끔 나는데, 코드 자체가 분석 불가능할 경우나 에러로 얼마나 수강자가 숙제에 무심한지 티를 내는 것입니다. 반드시 `terminal` 또는 `cmd` 에서 `python gowithflow.py` 명령으로 실행을 해보고 문제가 없을 경우에만 제출하도록 합시다. 물론, 위의 명령을 실행하기 위해서는 자기 나름대로 테스트 코드를 만들어 봐야 합니다.

## Next Work

이번 숙제는 사람에 따라 굉장히 어렵게 느끼기도 했을 것입니다. 그 이유는 처음으로 로직이 숙제에 들어갔기 때문입니다. 거의 모든 컴퓨터 프로그램은 로직에 의해 움직입니다. 조금 난해하더라도 반드시 스스로의 힘으로 작성해야 머리에 남을 것입니다. 아직 갈길이 멀습니다. 최선을 다해봅시다!

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

## Footnotes

