



Branch: master

Gachon_CS50_Python_KMOOC / lab_assignment / lab_7 / README.md

Find file

Copy path



blissray Add lab 7

6576299 7 minutes ago

1 contributor

135 lines (102 sloc) 12.3 KB

Raw

Blame

History



Lab Assignment #7- Baseball Game (baseball_game)

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

Introduction

수많은 난간. 자그마치 8개의 Lab을 끝내고 여기까지 온 여러분들을 환영합니다. 이 Lab Assignment은 사상최초로 어렵습니다. 놀랍게도 말이죠. 이때까지 어려운 Lab은 사실 존재하지 않았습니다. 단지 조금 시간이 오래 걸리거나 약간 복잡하게 느껴졌을 뿐입니다. 어렵게 느껴졌다면 일종의 착각일 것 입니다.

근데 이번엔 어렵습니다. 진짜로요. 하지만 어려운 만큼 가장 여러분에게 도움이 많이 되는 숙제라고 믿습니다. 참고로 가천대학교 수강생 설문조사에서도 가장 어렵지만 가장 도움이 많이된 숙제로 늘 꼽히고 있습니다. 프로그래밍을 해도 될지 말지 판별하는 숙제라고나 할까요?

본 Baseball Game Lab은 **간단한** 숫자 맞추기 게임입니다. 컴퓨터는 3자리의 Random Number를 생성하고 사용자는 숫자를 입력하면, 그 세자리 숫자가 Random Number와 얼마나 비슷한지에 따라 Strike과 Ball로 알려주는 게임입니다. 간단한 규칙은 다음과 같습니다.

- 사용자가 입력한 숫자와 컴퓨터가 생성한 숫자의 한 숫자와 자릿수(숫자의 위치)가 모두 일치하면 1 Strike
- 자릿수는 다르나 입력한 한 숫자가 존재하면 1 Ball
- 세자리 숫자를 정확히 입력하면 3 Strike로 게임 종료

이 때 컴퓨터가 생성하는 세 자리 숫자는 반드시 같은 숫자를 한 개만 가지고 있어야 합니다. 예를 들면 332, 474, 555 처럼 같은 숫자가 2개 이상있는 숫자는 게임에서 제외되어야 합니다. 실제 볼 스트라이크 판정 예제는 다음과 같습니다.

컴퓨터 생성 숫자	사용자 입력	판정 결과
472	123	0 Strike, 1 Ball
472	547	0 Strike, 2 Ball
472	247	0 Strike, 3 Ball
472	742	1 Strike, 2 Ball
472	472	3 Strike

어떻게 할지 걱정될 수도 있지만, 꽤 많은 힌트와 함께 여러분들을 바른 길로 인도해주는 함수를 제시할 것 입니다. 너무 걱정 말고 시작해 봅시다.

Baseball Game Overview

단순히 Strike와 볼판정만 해주면 좋겠지만 본 Lab에서는 처음으로 사용자의 입력에 대한 오류 처리를 실시하여야 합니다. 여러분들도 가끔 웹 사이트에 접속할 때 잘못된 입력을 바로 잡아주는 프로그램을 본적이 있을 것 입니다. 예를 들면, 이메일 주소를 입력할 때 "@"를 쓰지 않았던가, 비밀번호 입력시 숫자를 같이 쓰지 않았던가 할 경우, 자동으로 프로그램이 오류 입력임을 지정하고 재입력을 하게해 줄 것입니다. 본 프로그램도 마찬가지로 사용자가 아래와 같이 잘못 입력했을 경우, "Wrong Input" 이라는 메시지를 남기며 오류 처리를 해 줍니다.

-사용자가 숫자 대신 문자가 포함된 입력을 했을 경우 (ex. ab3)

- 사용자가 실수를 입력했을 경우 (ex. 23.5)
- 사용자가 입력한 세 자리 정수에 중복되는 숫자가 있을 경우 (ex. 333)

또한 사용자가 숫자를 정확히 입력하여 "3 Strikes" 나왔을 경우, 사용자에게 한번 더 게임을 할 의향이 있는지 물어보고 의향이 있을 경우 다시 게임을 실행 시킵니다. 이 때 아래와 같은 조건으로 게임을 재시작을 결정합니다.

- 사용자가 "Y" 또는 "Yes"라고 입력했을 경우 대소문자에 상관없이 게임을 재시작한다. 즉 "y", "yEs", "yes" 등의 경우에도 게임은 다시 시작된다.
- 사용자가 "N" 또는 "No"라고 입력했을 경우 대소문자에 상관없이 게임을 종료한다. 즉 "n", "nO", "NO" 등의 경우에도 게임은 종료된다.
- 사용자가 이외에 입력을 하였을 경우 "Wrong Input" 이라고 출력되면서 종료된다.

또 다른 종료 조건으로 사용자가 게임 중 0 을 입력하게 되면 종료해야 합니다. 이는 Lab #8의 Factorial Calculator의 종료 조건과 같습니다.

또한 새로운 게임이 시작할 때는 반드시 Random Number is : xxx 라는 문자가 출력되면서 xxx 부분에는 컴퓨터가 random하게 생성한 것을 표시해줍니다. 이는 자동 채점 프로그램을 동작하게 하기 위한 것으로 표시 되지 않을 경우 컴퓨터가 채점을 하지 못합니다. 실제 돌아가는 프로그램은 아래와 같습니다.

```
estella.gachon.ac.kr - PuTTY
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9
baseball_game$ python3.4 answer_baseball_game.py
Play Baseball
Random Number is : 694
Input guess number : abc
Wrong Input, Input again
Input guess number : 123.2
Wrong Input, Input again
Input guess number : 999
Wrong Input, Input again
Input guess number : 143
Strikes : 0 , Balls : 1
Input guess number : 142
Strikes : 0 , Balls : 1
Input guess number : 691
Strikes : 2 , Balls : 0
Input guess number : 694
Strikes : 3 , Balls : 0
You win, one more(Y/N)?l
Wrong Input, Input again
You win, one more(Y/N)?dw
Wrong Input, Input again
You win, one more(Y/N)?YES
Random Number is : 468
Input guess number : 484
Wrong Input, Input again
Input guess number : 486
Strikes : 1 , Balls : 2
Input guess number : 468
Strikes : 3 , Balls : 0
You win, one more(Y/N)?N
Thank you for using this program
End of the Game
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9
baseball_game$
```

```
estella.gachon.ac.kr - PuTTY
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9
baseball_game$ python3.4 answer_baseball_game.py
Play Baseball
Random Number is : 572
Input guess number : 58
Wrong Input, Input again
Input guess number : 582
Strikes : 2 , Balls : 0
Input guess number : 571
Strikes : 2 , Balls : 0
Input guess number : 573
Strikes : 2 , Balls : 0
Input guess number : 0
Thank you for using this program
End of the Game
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9
baseball_game$
```

숙제 template 파일 다운로드

먼저 숙제 template 파일을 다운받아야 합니다. Chrome 또는 익스플로러와 같은 웹 브라우저 주소창에 아래 주소를 입력합니다.

https://github.com/TeamLab/Gachon_CS50_Python_KMOOC/blob/master/lab_assignment/lab_7/lab_7.zip

다운로드를 위해 View Raw 또는 Download 버튼을 클릭합니다. 또는 [Lab 7 - 다운로드 링크](#) 를 클릭하면 자동으로 다운로드가 됩니다. 다운로드 된 lab_7.zip 파일을 작업 폴더로 이동한 후 압축해제 후 작업하길 바랍니다.

baseball_game.py 파일 Overview

atom 로 baseball_game.py 을 열어 전체적인 개요를 봅시다. atom baseball_game.py 명령으로 파일을 열어보면 main 함수와 여러개의 함수들이 존재할 것 입니다. 각 함수들은 여러분께서 직접 작성해서 제출해야 하는 함수이고, main 함수는 실제 baseball game을 실행하는 함수입니다. 각 함수의 구현 내용은 아래와 같습니다.

함수	설명
is_digit	문자열 값을 입력받아, 정수로 변환 가능할 경우는 True, 그렇지 않을 경우는 False로 변환해주는 함수
is_between_100_and_999	문자열 값을 입력받아, 정수로 변환하여 100이상 1000미만일 경우 True, 그렇지 않을 경우는 False를 반환해주는 함수. 이 때 입력되는 문자열 값은 정수형태의 문자열 값 임이 보장됨 ex) 324, 1103
is_duplicated_number	문자열로 된 세자리 양의 정수 값을 입력 받아 정수로 변환하였을 경우 중복되는 수가 있으면 True, 그렇지 않을 경우는 False를 반환해주는 함수. 이 때 문자열로 된 세자리 양의 정수값의 입력이 보장됨 ex) 117 - True, 123 - False, 103 - False, 113 - True
is_validated_number	문자열 값을 입력받아 아래 세가지 조건에 맞으면 True, 그렇지 않으면 False를 반환함 1) 숫자형 문자열이며, 2) 100이상 1000미만이며, 3) 중복되는 숫자가 없을 경우. 위의 세 가지 함수를 사용하여 구현이 가능함
get_not_duplicated_three_digit_number	입력 값은 없으며, 중복되는 숫자가 없는 세자리 정수값이 반환됨, 문자열이 아닌 정수값으로 반환됨. 이때 get_random_number() 함수를 사용하여 random number를 생성하고 생성된 random number에 중복되는 숫자가 없을 경우에만 해당 숫자를 반환해줌
get_strikes_or_ball	사용자가 입력한 세 자리 정수 문자열과, 컴퓨터가 자동으로 생성한 세 자리 정수 문자열을 입력받아, baseball game 규칙에 맞춰 [strikes, balls]를 반환해줌. 리스트 형태로 반환하여 strikes와 balls는 Integer Type 값
is_yes	문자열값을 입력받아 해당 입력값이 대소문자에 관계없이 "Y" 또는 "YES"일 경우 True, 그렇지 않을 경우 False를 반환함

함수	설명
is_no	문자열값을 입력받아 해당 입력값이 대소문자에 관계없이 "N" 또는 "NO"일 경우 True, 그렇지 않을 경우 False를 반환함

상당히 많고 복잡하고, 게다가 어렵기까지 합니다. 일단 하나하나 해결해보시길 바랍니다. 참고로 `get_random_number()` 함수는 임의의 3자리 자연수를 반환하는 함수로 반드시 써야 되는 함수니 `get_not_duplicated_three_digit_number` 함수에서 사용하시기 바랍니다.

main 함수 수정하기

위의 함수도 상당히 어렵지만 본 Lab의 가장 어려운 숙제는 바로 `main` 함수를 수정하는 일입니다. `main` 함수의 기본 template은 아래와 같습니다.

```
def main():
    print("Play Baseball")
    user_input = 999
    random_number = str(get_not_duplicated_three_digit_number())
    print("Random Number is : ", random_number)
    # ===Modify codes below=====
    # 위의 코드를 포함하여 자유로운 수정이 가능함

    # =====
    print("Thank you for using this program")
    print("End of the Game")
```

template은 lab 7 factorial calculator와 흡사하면서도 조금 다릅니다. 일단 프로그램이 시작되면 "Play Baseball" 한줄과 "Random Number is : " 라는 한 줄이 화면에 찍히고 시작합니다. 앞에서 설명했듯이 반드시 새로운 baseball game이 시작되면 `print("Random Number is : ", random_number)` 가 실행되어야 합니다. `random_number` 변수는 수강생이 임의로 수정가능하겠지만 `get_not_duplicated_three_digit_number` 함수를 잘 작성하여 `random_number` 에 할당해 주는게 이 숙제를 해결하는데 있어 가장 쉬운 것 입니다. 이 숙제가 까다로운 것은 숙제 자체도 그렇지만, 숙제를 검사할 때 검사 프로그램의 한계로 인해 위쪽 screen shot 처럼 작동되지 않으면 올바르게 코딩했음에도 불구하고 pass에 실패할 수도 있습니다. 한 가지 유의할 점은 사용자의 입력을 받을 때 `print` 문을 쓴 다음 `input` 문을 쓰지 말고 `input` 문만 사용하여 입력을 받아야 합니다. 예를 들면 `user_input = input('Input guess number : ')` 형태로 작성되어야 합니다. 이 또한 강의자의 귀차니즘과 개발 능력 한계니 이해해주면 좋겠습니다. 먼저 각 개별 함수를 목적에 맞게 잘 수정한 후, 개별 함수들을 잘 활용하여 `main` 함수를 작성하기 바랍니다. 실패할 가능성이 높은 함수지만, 굉장히 즐거운 작업이 될 것 입니다.

숙제 제출하기

모든 lab assignment가 종료되었습니다. 이제 숙제를 제출합니다.

- windows + r 를 누르고 cmd 입력 후 확인을 클릭합니다.
- 작업폴더로 경로를 이동합니다.
- cmd 창에서 아래의 코드를 입력합니다.

```
python submit.py
```

제대로 작성했다면 아래와 같은 메시지가 뜰 것 입니다.

Function Name	Passed?	Feedback
is_digit	PASS	Good Job
is_yes	PASS	Good Job
is_duplicated_number	PASS	Good Job
is_no	PASS	Good Job
get_strikes_or_ball	PASS	Good Job
is_validated_number	PASS	Good Job
is_between_100_and_999	PASS	Good Job
main	PASS	Good Job
get_not_duplicated_three_digit_number	PASS	Good Job

아마도 정말로 감격스러운 순간일 것 입니다. 고생하셨습니다.

Next Work

이제 여러분은 프로그래밍을 위한 거대한 한발을 내딛었다고 생각합니다. 이때까지는 영어로치면 A, B, C 알파벳 노래를 부르거나 기껏해야 "How are you? Fine thank you, and you" 정도를 읊었다면 이제 프로그래밍으로 쉬운 어린이 "팝송" 한곡 부른 정도라고나 할까요? 여러분들은 충분히 잘하셨습니다. 오늘 자기전에 맥주한잔을 마시면서 자신이 무한한 가능성이 있는 존재임을 깨닫기 바랍니다. 여기까지 와주셔서 감사합니다.

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

Footnotes

