

고급통계프로그래밍 #10

2017580034 통계학과 이유민

#1

random 모듈을 이용하여 숫자 맞추기 게임을 구현하시오. 컴퓨터가 랜덤하게 1과 100사이의 정수를 고르고, 사용자가 1과 100사이의 추측값을 주면 정답과 비교하여 적절한 메시지(가령 크다, 작다, 잘했어요 등)를 주시오.

In [1]:

```
import random
def guess(a): #추측값
    answer=random.randrange(1,101) # 1~100 사이의 추측값
    print('answer:', answer)
    if answer == a:
        print('You are right!')
    elif answer < a:
        print("Lower number")
    else :
        print('Upper number')
```

In [2]:

```
guess(10)
```

```
answer: 33
Upper number
```

#2

Medals.csv, Athlete_Country_Map.csv, Athlete_Sports_MAP.csv 파일을 이용하여 대한민국의 연도별로 메달을 집계해 보시오. 또한 대한민국의 선수별로 총메달 갯수를 구하고 금메달, 은메달, 동메달 갯수 순으로 정렬하시오.

In [3]:

```
#-*- coding:utf-8 -*-
import pandas as pd
import numpy as np

# data
medal=pd.read_csv('./Medals.csv')
c_map=pd.read_csv('./Athlete_Country_Map.csv')
s_map=pd.read_csv('./Athlete_Sports_Map.csv')
```

In [4]:

```
df=pd.merge(left=medal, right=c_map, left_on='Athlete', right_on='Athlete')
df.head()
```

Out[4]:

	Athlete	Age	Year	Closing Ceremony Date	Gold Medals	Silver Medals	Bronze Medals	Total Medals	Country
0	Michael Phelps	23.0	2008	08/24/2008	8	0	0	8.0	United States
1	Michael Phelps	19.0	2004	08/29/2004	6	0	2	8.0	United States
2	Michael Phelps	27.0	2012	08/12/2012	4	2	0	6.0	United States
3	Natalie Coughlin	25.0	2008	08/24/2008	1	2	3	6.0	United States
4	Natalie Coughlin	21.0	2004	08/29/2004	2	2	1	5.0	United States

In [5]:

```
c_map_dp=c_map.drop_duplicates(subset='Athlete')
merged_dp=pd.merge(left=medal, right=c_map_dp, left_on='Athlete', right_on='Athlete')
s_map_dp=s_map.drop_duplicates(subset='Athlete')
m_final=pd.merge(left=merged_dp, right=s_map_dp, left_on='Athlete', right_on='Athlete' )

m_final
```

Out[5]:

	Athlete	Age	Year	Closing Ceremony Date	Gold Medals	Silver Medals	Bronze Medals	Total Medals	Country	Sp
0	Michael Phelps	23.0	2008	08/24/2008	8	0	0	8.0	United States	Swimmer
1	Michael Phelps	19.0	2004	08/29/2004	6	0	2	8.0	United States	Swimmer
2	Michael Phelps	27.0	2012	08/12/2012	4	2	0	6.0	United States	Swimmer
3	Natalie Coughlin	25.0	2008	08/24/2008	1	2	3	6.0	United States	Swimmer
4	Natalie Coughlin	21.0	2004	08/29/2004	2	2	1	5.0	United States	Swimmer
...
8526	Olena Sadovnycha	32.0	2000	10/01/2000	0	1	0	1.0	Ukraine	Archery
8527	Kateryna Serdiuk	17.0	2000	10/01/2000	0	1	0	1.0	Ukraine	Archery
8528	Wietse van Alten	21.0	2000	10/01/2000	0	0	1	1.0	Netherlands	Archery
8529	Sandra Wagner-Sachse	31.0	2000	10/01/2000	0	0	1	1.0	Germany	Archery
8530	Rod White	23.0	2000	10/01/2000	0	0	1	1.0	United States	Archery

8531 rows × 10 columns



1. 대한민국의 연도별 메달

In [6]:

```
# 대한민국 추출
s_med = m_final[m_final['Country']=='South Korea']

# 연도별 메달 집계
year_sorted=s_med.groupby('Year') #연도로 groupby
year_sorted.sum()
```

Out[6]:

	Age	Gold Medals	Silver Medals	Bronze Medals	Total Medals
Year					
2000	1730.0	12	26	35	73.0
2002	87.0	5	2	0	7.0
2004	1223.0	14	28	10	52.0
2006	222.0	14	3	2	19.0
2008	1861.0	41	11	26	78.0
2010	220.0	6	10	2	18.0
2012	1453.0	18	13	30	61.0

2. 선수별 총 메달 갯수 & 금/은/동 갯수 순 정렬

In [7]:

```
# 금/은/동 sort
med_sort=s_med.sort_values(['Gold Medals', 'Silver Medals', 'Bronze Medals'], ascending=False) #금>
med_sort
```

Out[7]:

	Athlete	Age	Year	Closing Ceremony Date	Gold Medals	Silver Medals	Bronze Medals	Total Medals	Country	Sport
43	An Hyeon-Su	20.0	2006	02/26/2006	3	0	1	4.0	South Korea	Short-Track Speed Skating
139	Jin Seon-Yu	17.0	2006	02/26/2006	3	0	0	3.0	South Korea	Short-Track Speed Skating
133	Lee Jeong-Su	20.0	2010	02/28/2010	2	1	0	3.0	South Korea	Short-Track Speed Skating
469	Jin Jong-O	32.0	2012	08/12/2012	2	0	0	2.0	South Korea	Shooting
920	Ki Bo-Bae	24.0	2012	08/12/2012	2	0	0	2.0	South Korea	Archery
...
7799	Ha Tae-Gwon	25.0	2000	10/01/2000	0	0	1	1.0	South Korea	Badminton
7803	Kim Dong-Mun	24.0	2000	10/01/2000	0	0	1	1.0	South Korea	Badminton
7809	Na Gyeong-Min	27.0	2004	08/29/2004	0	0	1	1.0	South Korea	Badminton
8478	Im Dong-Hyeon	27.0	2012	08/12/2012	0	0	1	1.0	South Korea	Archery
8484	Kim Bub-Min	21.0	2012	08/12/2012	0	0	1	1.0	South Korea	Archery

274 rows × 10 columns

#3

계산기 프로그램의 모든 기능이 잘 작동하도록 완성하시오.

In [8]:

```
from tkinter import *
from decimal import *
from math import factorial
```

In [9]:

```
# 상수/진수 변환/계승 기능

## 팩토리얼 함수
def factorial(n):
    try:
        n = int(n)
    except:
        return "--> Error!"

    # 0! = 1
    if n == 0:
        return 1

    #400이상의 수는 받지 X
    if n > 40:
        return "--> The Answer is too large!"

    #음수는 error
    if n < 0:
        return "--> Error!"

    ans = n
    while n > 1:
        ans = ans*(n-1)
        n = n-1
    return ans

## 10진수 > 2진수
def to_binary(n):
    try:
        n = int(n)
        return bin(n)[2:]
    except:
        return "-->Error!"

## 2진수 > 10진수
def from_binary(n):
    try:
        return int(n,2)
    except:
        return "-->Error!"
```

In [10]:

```
def click(key):

    # = 버튼 누르면 계산 수행
    if key == "=":
        try:
            result = str(eval(display.get()))[0:10]
            display.insert(END, "=" + result)
        except:
            display.insert(END, "--> Error!")

    # c 버튼 - display 엔트리 위젯 내용 비움
    elif key == "C":
        display.delete(0, END)

    # 상수 버튼 작업
    elif key == constants_list[0]:
        display.insert(END, "3.141592654")

    # 함수 버튼 작업
    elif key == functions_list[0]:
        n = display.get() #현재 디스플레이 엔트리 위젯 값 수집
        display.delete(0,END) #현재 디스플레이 엔트리 위젯 내용 비우기
        display.insert(END, factorial(n))

    elif key == functions_list[1]:
        n = display.get() # display 엔트리 위젯 내용 수집
        display.delete(0,END) # display 엔트리 위젯 내용 비움
        display.insert(END, to_binary(n))

    elif key == functions_list[2]:
        n = display.get() # display 엔트리 위젯 내용 수집
        display.delete(0,END) # display 엔트리 위젯 내용 비움
        display.insert(END, from_binary(n))

    # 이외의 다른 키 누르는 경우 실행시킬 기본 동작
    else:
        display.insert(END, key)
```

In [11]:

```
window = Tk()
window.title("Mycalculator")

top_row = Frame(window)
top_row.grid(row=0, column=0, columnspan=2, sticky=N)

# 내용 수정이 가능한 엔트리 위젯을 사용해 결과 디스플레이 사용
display = Entry(top_row, width=45, bg="light gray")
display.grid()

# 숫자 버튼 프레임 생성
num_pad = Frame(window)
num_pad.grid(row=1, column=0, sticky=W)

# 숫자 버튼에 제공되는 숫자
num_pad_list = [
    '7', '8', '9',
    '4', '5', '6',
    '1', '2', '3',
    '0', '.', '=' ]

# 숫자 버튼 생성
r = 0
c = 0

for btn_text in num_pad_list:
    def cmd(x=btn_text):
        click(x)

    Button(num_pad, text=btn_text, width=7, command=cmd).grid(row=r, column=c)
    c = c+1
    if c > 2:
        c = 0
        r = r + 1

# 연산자 프레임 생성
operator = Frame(window)
operator.grid(row=1, column=1, sticky=E)

operator_list = [
    '*', '/',
    '+', '-',
    '(', ')',
    'C' ]

# 연산자 버튼 생성
for btn_text in operator_list:
    def cmd(x=btn_text):
        click(x)

    Button(operator, text=btn_text, width=5, command=cmd).grid(row=r, column=c)
    c = c+1
    if c > 1:
        c = 0
        r = r + 1

# 상수 프레임 생성
```



```
constants = Frame(window)
constants.grid(row=3, column=0, sticky=W)

constants_list = [
    'pi']

# 상수 버튼 생성
r = 0
c = 0
for btn_text in constants_list:
    def cmd(x=btn_text):
        click(x)
    Button(constants, text=btn_text, width=24, command=cmd).grid(row=r, column=c)
    r = r+1

# 함수 프레임 생성
functions = Frame(window)
functions.grid(row=3, column=1, sticky=E)

functions_list = [
    'factorial (!)',
    '-> binary',
    'binary -> 10']

# 함수 버튼 생성
r = 0
c = 0

for btn_text in functions_list:
    def cmd(x=btn_text):
        click(x)
    Button(functions, text=btn_text, width=24, command=cmd).grid(row=r, column=c)
    r = r+1

window.mainloop()
```