

파이썬 초초초보

****UNIST Mathematical Sciences****
kyunghoon@unist.ac.kr

PYTHON 3

변수 이름 만들기

변수 이름은 영어알파벳 또는 언더바로 시작해야 합니다.

In [1]:

```
a = 1
my_name = 'Hoon'
MYNAME = 'Kim'
YOUR_Name = 'Lee'
_apple = 500.
```

다음의 이름들은 변수 이름으로 사용할 수 없습니다.

In [2]:

```
import keyword
```

In [53]:

```
keyword.kwlist
```

Out[53]:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'cl
ass', 'continue', 'def', 'del', 'elif', 'else', 'except', 'fi
nally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'retur
n', 'try', 'while', 'with', 'yield']
```

변수에 값 대입하기

숫자

In [4]:

```
number = 1                # number에 숫자 1을 대입
number = number + 2       # number에 원래 number에 있던 값에 2를 더하여 대입
print(number)             # number 값 출력
```

3

문자

In [5]:

```
alphabet = 'apple'        # 변수명 alphabet에 string 'apple'을 대입
alphabet = alphabet + 'pie' # 문자열에 문자열을 더하여 alphabet에 대입
print(alphabet)           # 변수명 alphabet 값 출력
```

applepie

주석 사용하기

In [6]:

```
# 만든이 : Royden
# 만든 날짜 : 2017.08.10. 금요일
```

In [7]:

```
"""
    길다란 주석
"""
a = 5
```

자료형

In [8]:

```
int_data = 10                # 정수
float_dat = 3.1415           # 실수
complex_data = 1+2j          # 복소수
string_data_1 = 'I LOVE MATH' # 문자열
string_data_2 = '수학이 제일 좋아' # 문자열
list_data = [0, 1, 0, 1, 2, 3, 4, 0, 0, 0, 0] # 리스트
tuple_data = (9, 87, 6, 543, 2, 1)           # 튜플
dict_data = {'apple': 'red', 'banana': 'yellow'} # 사전
```

- 리스트는 순서 있게 객체를 나열한 자료형. 각 요소는 콤마로 구분함.
- 튜플은 리스트와 비슷하지만 요소 값을 바꿀 수 없음.
- 사전은 {} 안에 키:값 으로 된 요소가 들어 있는 순서가 없는 자료형.

파이썬 사용하기

제어문, 함수이름, 클래스이름 뒤에 콜론(:)으로 끝을 표시함.

콜론 다음 줄부터 실행 코드를 작성함.

1. 가장 바깥쪽(위쪽)의 실행 코드는 들여쓰기 없이 실행해야 함.
2. 콜론 다음 라인부터 시작하는 실행 코드는 들여쓰기 간격이 모두 동일해야 함.

조건문

조건이 하나일 때

if 조건:

실행코드

조건을 만족하지 않는 경우 else를 쓴다

if 조건:

실행코드

else:

실행코드

조건이 여러 개 일 때

if 조건:

실행코드

elif 조건:

실행코드

In [9]:

```
data = ['a', 'b', 'c']
if 'a' in data:
    print('a가 data에 있습니다.')
    print('b, c가 있는지는 모르겠네요.')
else:
    print('a가 data에 없습니다.')
```

a가 data에 있습니다.

b, c가 있는지는 모르겠네요.

반복문 1

for 변수 in 범위:

반복으로 실행할 코드

In [10]:

```
data = [2, 3, 5, 7, 9]
for x in data:
    print(x)
```

2
3
5
7
9

In [11]:

```
string = 'abcde'
for s in string:
    print(s)
```

a
b
c
d
e

반복문과 조건문 섞어보기

In [12]:

```
data = [2, 3, 4, 5, 6]
for x in data:
    print(x)
    if x < 3:
        continue    # 반복문 계속 수행
    else:
        break        # 반복문 중지
```

2
3

for과 else

In [13]:

```
data1 = [1, 2, 3]
data2 = ['a', 'b']
for x in data1:
    for y in data2:
        print(x, y)
else: # for 반복이 모두 끝났을 때 실행할 코드
    print("끝!")
```

1 a
1 b
2 a
2 b
3 a
3 b
끝!

반복문 2

In [14]:

```
x = 0
while x < 5:
    x = x + 1
    print(x)
```

1
2
3
4
5

In [15]:

```
x = 0
while x < 5:
    x = x + 1
    if x == 3:
        print(x, 'is 3')
```

3 is 3

문제 1

직각 삼각형을 출력하시오.

In [16]:

```
for i in range(10):
    print(i*'*')
```

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

오른쪽으로 정렬된 직각 삼각형을 출력하시오.

In [17]:

```
for i in range(10):
    print(' '* (10-i), i*'*')
```

```

    *
  **
 ***
****
*****
*****
*****
*****
*****
*****
```

역직각 삼각형을 출력하시오.

In [18]:

```
list(range(10))[::-1]
```

Out[18]:

```
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

In [19]:

```
data = list(range(10))[::-1]
for i in data:
    print(i*'*')
```

```
*****
*****
*****
*****
*****
****
***
**
*
```

In [20]:

```
data = range(10)
for i in data:
    print('*'*(10-i))
```

```
*****
*****
*****
*****
*****
*****
****
***
**
*
```

오른쪽 정렬된 역직각 삼각형을 출력하시오.

In [21]:

```
data = range(10)
for i in data:
    print(' '*i, '*'*(10-i))
```

```
*****
*****
*****
*****
*****
*****
****
***
**
*
```

문제 2

1부터 100까지 더한 값을 구하시오.

In [22]:

```
value = 0
for i in range(1, 101):
    value += i
```

In [23]:

```
print(value)
```

5050

$$\frac{n(n+1)}{2}$$

을 이용해서 합 구하기

In [24]:

```
def f(n):
    return n*(n+1)/2
```

In [25]:

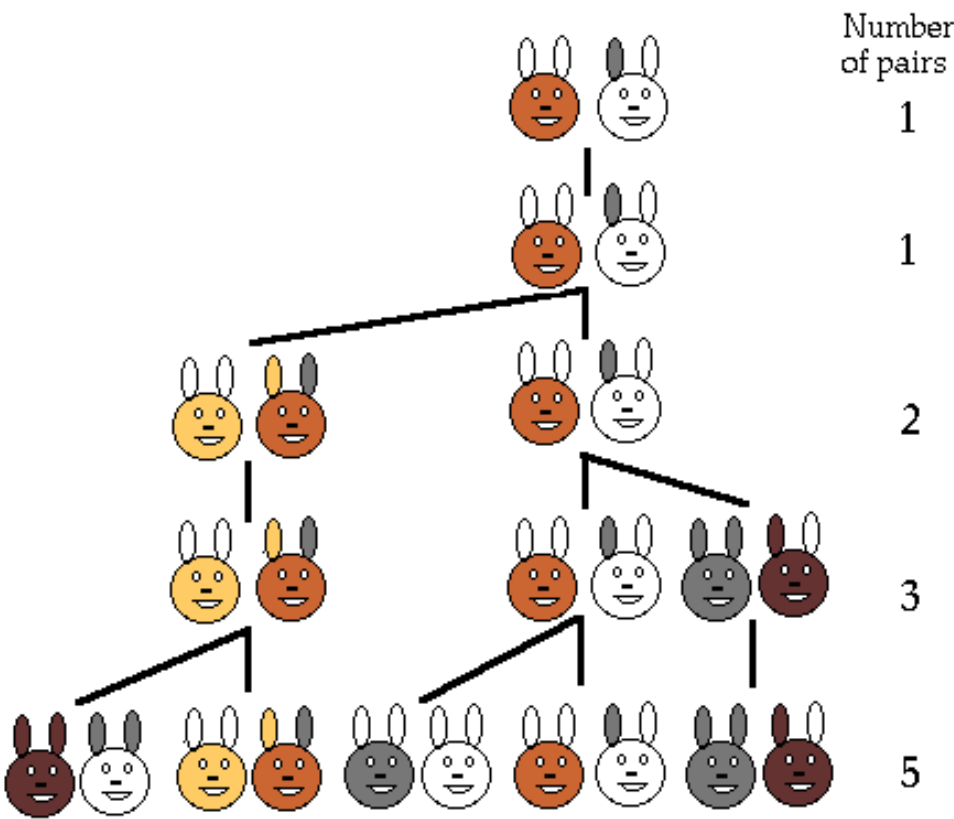
```
f(100)
```

Out[25]:

5050.0

문제 3

피보나치 수열을 구하는 코드를 짜보자



1 1 2 3 5 8 13 21 ...

반복문으로 짜보자

In [26]:

```
a = 1
b = 1
print(a)
print(b)
for i in range(10):
    c = a + b
    print(c)
    a = b
    b = c
```

```
1
1
2
3
5
8
13
21
34
55
89
144
```

In [27]:

```
a = 1
b = 1
print(a)
print(b)
for i in range(10):
    c = a + b
    print(c)
    a, b = b, c
```

```
1
1
2
3
5
8
13
21
34
55
89
144
```

함수로 짜보자

In [28]:

```
def fibo(n):  
    if (n<=2):  
        return 1  
    else:  
        return fibo(n-1) + fibo(n-2)
```

In [29]:

```
fibo(5)
```

Out[29]:

5

In [30]:

```
fibo(20)
```

Out[30]:

6765

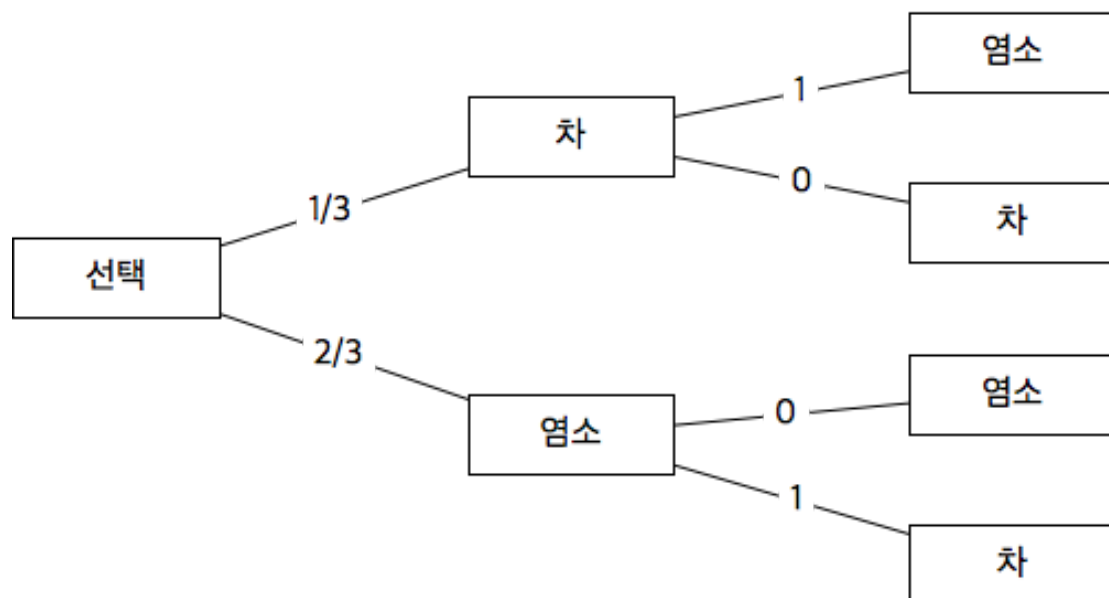
In [31]:

```
for i in range(1, 11):  
    print(fibo(i))
```

1
1
2
3
5
8
13
21
34
55

문제 4

몬티홀 딜레마를 짜보자



In [32]:

```
import random # random 라이브러리를 불러오자
```

In [33]:

```
doors = ["A", "B", "C"] # 문 세 개를 만들자
prize = random.choice(doors) # 문 세 개 중에 하나를 랜덤하게 선택하자
```

In [34]:

```
doors[0] # 내가 선택한 문을 첫 번째 문이라고 하자
```

Out[34]:

'A'

In [35]:

```
def MC(myanswer, prize):  
    """  
    진행자 함수.  
    내가 선택한 답과 정답인 문을 알려주면  
    내가 선택한 답과 정답이 아닌 문을 하나 알려준다.  
    """  
  
    candidates = ["A", "B", "C"] # 후보 리스트를 하나 만들자  
    if myanswer == prize: # 만약 내 답과 정답이 같으면  
        candidates.remove(prize) # 정답을 후보리스트에서 지우자  
    else: # 내 답이 정답과 같지 않으면  
        candidates.remove(prize) # 후보리스트에서 정답을 지우고  
        candidates.remove(myanswer) # 후보리스트에서 내 답을 지우자  
    return random.choice(candidates) # 남아 있는 후보 중에 랜덤하게 선택해서 return 하자.
```

In [36]:

```
MC(doors[0], prize) # MC에게 내가 선택한 문과 정답을 알려주면, 내가 선택하지 않은 문 중에서 정답이 아닌 문을 하나 보여준다.
```

Out[36]:

'C'

내 답을 유지하는 실험

In [37]:

```
doors = ["A", "B", "C"]
for i in range(10):
    print(":: 실험", i)

    # 상품이 있는 곳을 결정하자.
    prize = random.choice(doors)

    # 문을 하나 선택하자.
    myanswer = doors[0]

    # 진행자가 문을 하나 열어 보여준다.
    opened_door = MC(myanswer, prize)

    # 내 답을 유지한다.
    # 아무것도 안함.

    if myanswer == prize:
        print('당첨!')
    else:
        print('실패')
```

```
:: 실험 0
실패
:: 실험 1
당첨!
:: 실험 2
실패
:: 실험 3
실패
:: 실험 4
당첨!
:: 실험 5
실패
:: 실험 6
실패
:: 실험 7
당첨!
:: 실험 8
실패
:: 실험 9
당첨!
```

내 답을 바꾸는 실험

In [38]:

```
for i in range(10):
    doors = ["A", "B", "C"]

    print("실험", i)

    # 상품이 있는 곳을 결정하자.
    prize = random.choice(doors)

    # 문을 하나 선택하자.
    myanswer = doors[0]

    # 진행자가 문을 하나 열어 보여준다.
    opened_door = MC(myanswer, prize)

    # 내 답을 바꾼다.
    doors.remove(myanswer)
    doors.remove(opened_door)

    myanswer = doors[0]

    if myanswer == prize:
        print('당첨!')
    else:
        print('실패')
```

실험 0
실패
실험 1
당첨!
실험 2
실패
실험 3
실패
실험 4
실패
실험 5
당첨!
실험 6
당첨!
실험 7
당첨!
실험 8
당첨!
실험 9
실패

함수로 만들어 보자

In [39]:

```
def nochange():
    doors = ["A", "B", "C"]

    # 상품이 있는 곳을 결정하자.
    prize = random.choice(doors)

    # 문을 하나 선택하자.
    myanswer = doors[0]

    # 진행자가 문을 하나 열어 보여준다.
    opened_door = MC(myanswer, prize)

    # 내 답을 유지한다.
    # 아무것도 안함.

    if myanswer == prize:
        return 1 # 내 답이 정답이면 1을 반환
    else:
        return 0 # 내 답이 틀리면 0을 반환
```

In [40]:

```
value = 0
for i in range(100):
    value += nochange() # 내 답이 정답이면 1을 더하고, 내 답이 틀리면 0을 더한다.
print(value, '/ 100')
```

29 / 100

In [41]:

```
result = [] # 최종 결과값을 넣을 리스트를 만들자.
for i in range(100): # 100번 시행하는 실험을 100번 실행하자.
    value = 0
    for i in range(100): # 100번 시행하자
        value += nochange() # 당첨된 값을 더하자
    result.append(value) # 100번 중 당첨된 시행 결과를 result에 새로운 값으로 넣어
주자.
```

In [51]:

```
%pylab inline
```

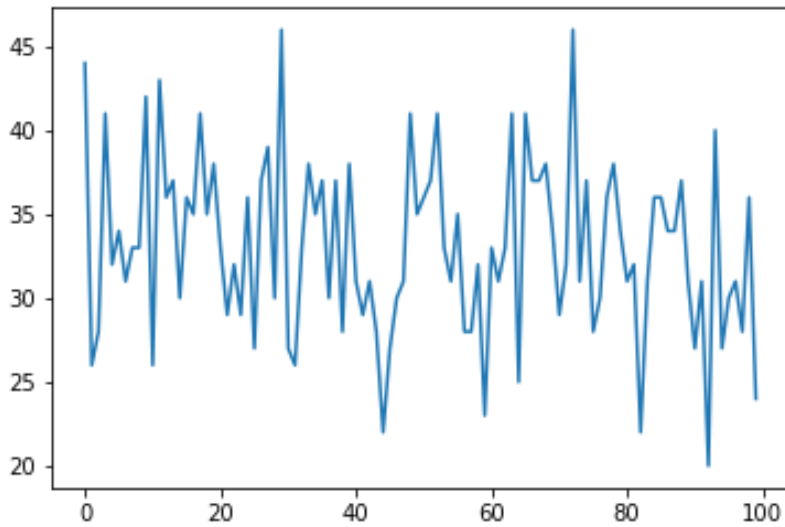
Populating the interactive namespace from numpy and matplotlib
b

In [43]:

```
plt.plot(result) # 그림으로 그리자
```

Out[43]:

[<matplotlib.lines.Line2D at 0x7fe528d36630>]



In [44]:

```
def change():
    doors = ["A", "B", "C"]
    # 상품이 있는 곳을 결정하자.
    prize = random.choice(doors)

    # 문을 하나 선택하자.
    myanswer = doors[0]

    # 진행자가 문을 하나 열어 보여준다.
    opened_door = MC(myanswer, prize)

    # 내 답을 바꾼다.
    doors.remove(myanswer)
    doors.remove(opened_door)

    myanswer = doors[0]

    if myanswer == prize:
        return 1
    else:
        return 0
```

In [45]:

```
# 100번 시행해서 당첨된 값을 구해보자.  
value = 0  
for i in range(100):  
    value += change()  
print(value)
```

72

In [46]:

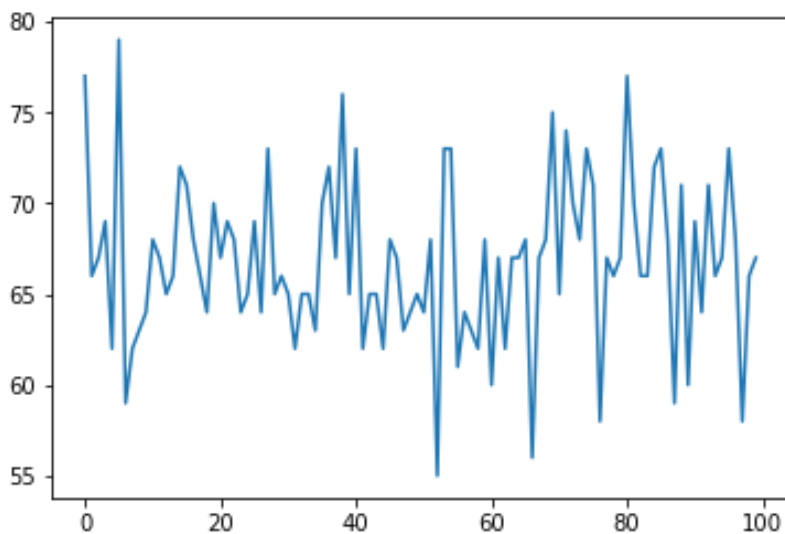
```
result = []  
for i in range(100):  
    value = 0  
    for i in range(100):  
        value += change()  
    result.append(value)
```

In [47]:

```
plt.plot(result)
```

Out[47]:

[<matplotlib.lines.Line2D at 0x7fe528ca4470>]



최종 함수를 만들어 보자.

In [48]:

```
result_change = []
result_nochange = []
for i in range(100):
    value = 0
    for i in range(100):
        value += change()
    result_change.append(value)

for i in range(100):
    value = 0
    for i in range(100):
        value += nochange()
    result_nochange.append(value)
```

In [49]:

```
plt.plot(result_change, label='Change')
plt.plot(result_nochange, label='No Change')
plt.legend()
plt.title('Monty Hall Dilemma')
plt.xlabel('Trial')
plt.ylabel('? / 100')
```

Out[49]:

<matplotlib.text.Text at 0x7fe528cd52e8>

