

2주. Python Basics			
학번	32152339	이름	송준영

Q1. 화면에서 구구단의 단(2~9)를 입력받아 해당 단을 출력하는 프로그램을 작성하시오

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
def gugudan():
    dan=int(input('단(2~9)을 입력해주세요 : '))
    for i in range(2,10):
        print( ('{}*{}={}'.format(dan,i,dan*i))
    gugudan()
```

실행화면 캡처:

```
In [61]: def gugudan():
...:     dan=int(input('단(2~9)을 입력해주세요 : '))
...:     for i in range(2,10):
...:         print( ('{}*{}={}'.format(dan,i,dan*i))
...:
...:     gugudan()
```

단(2~9)을 입력해주세요 : 7

7*2=14

7*3=21

7*4=28

7*5=35

7*6=42

7*7=49

7*8=56

7*9=63

Q2. 비만도 체질량지수 (BMI)는 체중(kg단위) ÷ (키(m단위))² 로 계산을 한다.

(예: 키170cm, 몸무게 85kg 이면 BMI = 85 ÷ (1.7)² 이다.

BMI 값에 따라 다음과 같이 비만도를 판단한다.

18.5 이하: 저체중, 18.5~23: 정상, 23~25: 과체중, 25~30: 비만, 30이상: 고도비만

화면에서 키(kg)와 몸무게(cm)를 입력받아 BMI 값과 비만도를 출력하는 프로그램을 작성 하시오.

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
def bmi():
    h=int(input('키(cm)를 입력하십시오 :'))
    w=int(input('몸무게(kg)를 입력하십시오 :'))
    bmi=w/((h/100)**2)
    print('체질량지수(BMI)={}'.format(bmi))
    if bmi>=30 : print('비만도 : 고도비만')
    elif bmi >=25 : print('비만도 : 비만')
    elif bmi >=23 : print('비만도 : 과체중')
    elif bmi >=18.5 : print('비만도 : 정상')
    else : print('비만도 : 저체중')
bmi()
```

실행화면 캡처:

```
In [63]: def bmi():
...:     h=int(input('키(cm)를 입력하십시오 :'))
...:     w=int(input('몸무게(kg)를 입력하십시오 :'))
...:     bmi=w/((h/100)**2)
...:     print('체질량지수(BMI)={}'.format(bmi))
...:     if bmi>=30 : print('비만도 : 고도비만')
...:     elif bmi >=25 : print('비만도 : 비만')
...:     elif bmi >=23 : print('비만도 : 과체중')
...:     elif bmi >=18.5 : print('비만도 : 정상')
...:     else : print('비만도 : 저체중')
...:
...:     bmi()
```

키(cm)를 입력하십시오 :170

몸무게(kg)를 입력하십시오 :85

체질량지수(BMI)=29.411764705882355

비만도 : 비만

Q3. 다음과 같이 높이를 입력하면 트리를 그리는 프로그램을 작성하시오

높이를 입력하시오: 5

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

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오  
def tree():  
    x=int(input('높이를 입력하시오:'))  
    for i in range(x):  
        print('*'*(i+1))  
tree()
```

실행하면 캡처:

```
In [59]: def tree():  
...:     x=int(input('높이를 입력하시오:'))  
...:     for i in range(x):  
...:         print('*'*(i+1))  
...:     tree()
```

높이를 입력하시오:5

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

Q4. 다음과 같이 높이를 입력하면 트리를 그리는 프로그램을 작성하시오. 단, 트리를 그리는 부분을 함수로 작성하여 이용하시오.

높이를 입력하시오: 5

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

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
def tree_reverse():
    x=int(input('높이를 입력하시오:'))
    for i in range(x):
        print((x-i-1)*' ','*'*i)
    tree_reverse()
```

실행하면 캡처:

```
In [58]: def tree_reverse():
...:     x=int(input('높이를 입력하시오:'))
...:     for i in range(x):
...:         print((x-i-1)*' ','*'*i)
...:
...: tree_reverse()
```

높이를 입력하시오:5

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

Q5. 배열 before_arr 에는 다음과 같은 값이 저장되어 있다.

{ 7, 1, 10, 4, 6, 9, 2, 8, 15, 12, 17, 19, 18 }

before_arr 에 있는 값들을 오름차순으로 정렬하여 배열 after_arr 에 저장하는 프로그램을 작성하시오. 단, 알고리즘은 다음과 같다.

- 1) before_arr에서 가장 작은 값을 찾는다.
- 2) 그 값을 after_arr 에 앞쪽에 넣는다.
- 3) before_arr에서 찾은 가장 작은 값은 999 로 변경한다.
- 4) 1)~3) 과정을 배열의 길이만큼 반복한다.

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
import numpy as np
before_arr=[ 7, 1, 10, 4, 6, 9, 2, 8, 15, 12, 17, 19, 18 ]
after_arr=[]
def sorting(lst):
    for i in range(len(lst)):
        global after_arr
        after_arr.append(min(lst))
        lst[lst.index(min(lst))]=999
    return after_arr
print(sorting(before_arr))
print(after_arr)
```

실행화면 캡처:

```
In [250]: import numpy as np
...: before_arr=[ 7, 1, 10, 4, 6, 9, 2, 8, 15, 12, 17, 19, 18 ]
...: after_arr=[]
...: def sorting(lst):
...:     for i in range(len(lst)):
...:         global after_arr
...:         after_arr.append(min(lst))
...:         lst[lst.index(min(lst))]=999
...:     return after_arr
...:
...: print(sorting(before_arr))
...: print(after_arr)
[1, 2, 4, 6, 7, 8, 9, 10, 12, 15, 17, 18, 19]
[1, 2, 4, 6, 7, 8, 9, 10, 12, 15, 17, 18, 19]
```

Q6. 길이가 같은 두 정수 배열에서 동일 인덱스 위치의 값을 곱하여 새로운 배열에 저장하여 반환하는 함수 `mul_arr()`을 작성하고 제대로 작동하는지 테스트 하시오.

예) $A = \{ 1, 2, 3 \}$, $B = \{ 4, 5, 6 \}$ 일 때 `mul_arr(A,B)` 는 $\{ 4, 10, 18 \}$ 를 return 한다.

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
def mul_arr(a,b):
    c=[]
    for i in range(len(a)):
        c.append(a[i]*b[i])
    return c

A,B=[1,2,3],[4,5,6]
mul_arr(A,B)
```

실행화면 캡처:

```
In [240]: def mul_arr(a,b):
...:     c=[]
...:     for i in range(len(a)):
...:         c.append(a[i]*b[i])
...:     return c
...:
...:
...: A,B=[1,2,3],[4,5,6]
...: mul_arr(A,B)
Out[240]: [4, 10, 18]
```

Q7. 다음과 같은 구조의 numpy array 인 **my_arr** 를 생성하여 내용을 보이시오.

- 10행 5열의 구조
- array의 내용은 1~50 사이의 정수.
- 1~50의 정수로 1차원 array를 만든 뒤 reshape 으로 2차원 배열로 변형

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
my_arr=np.random.randint(1,51,50)
my_arr.ndim
my_arr=my_arr.reshape(10,5)
my_arr.ndim
my_arr
```

실행화면 캡처:

In [9]: my_arr=np.random.randint(1,51,50)

In [10]: my_arr.ndim

Out[10]: 1

In [11]: my_arr=my_arr.reshape(10,5)

In [12]: my_arr.ndim

Out[12]: 2

In [13]: my_arr

Out[13]:

```
array([[38, 20, 13, 17,  5],
       [50, 39, 35, 46, 21],
       [19,  2,  4, 34, 40],
       [19,  8, 46, 22,  2],
       [25, 37, 47, 45, 19],
       [38,  1, 11, 25,  9],
       [32, 44, 28, 34, 11],
       [21,  6, 48, 21, 17],
       [25, 15, 36, 22, 47],
       [ 1, 31, 16, 14,  6]])
```

Q8. Q7에서 생성한 **my_arr** 에 대해 다음의 문제를 해결하는 코드를 작성하고 실행 결과를 보이시오

- (1) **my_arr** 의 값들에 각각 2를 곱한 결과를 보이시오
- (2) **my_arr** 의 값들중 20 이하의 값들에 대해서만 100을 더한 후에 **my_arr** 에 저장하시오. **my_arr** 의 내용을 보이시오
- (3) **my_arr** 에서 2,3열의 데이터만 잘라서 보이시오.
- (4) **my_arr** 에서 5~8행의 데이터만 잘라서 보이시오.

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
(1)
my_arr*2
(2)
my_arr=my_arr[my_arr<=20]+100
(3)
my_arr[:,(2,3)]
(4)
my_arr[4:7]
```

실행화면 캡처:

(1)

In [30]: my_arr*2

Out[30]:

```
array([[ 76,  40,  26,  34,  10],
       [100,  78,  70,  92,  42],
       [ 38,   4,   8,  68,  80],
       [ 38,  16,  92,  44,   4],
       [ 50,  74,  94,  90,  38],
       [ 76,   2,  22,  50,  18],
       [ 64,  88,  56,  68,  22],
       [ 42,  12,  96,  42,  34],
       [ 50,  30,  72,  44,  94],
       [  2,  62,  32,  28,  12]])
```

(2)

In [33]: my_arr=my_arr[my_arr<=20]+100

In [34]: my_arr

Out[34]:

```
array([120, 113, 117, 105, 119, 102, 104, 119, 108, 102, 119, 101, 111,
       109, 111, 106, 117, 115, 101, 116, 114, 106])
```


(3)

```
In [32]: my_arr[4:7]
```

```
Out[32]:
```

```
array([[25, 37, 47, 45, 19],  
       [38,  1, 11, 25,  9],  
       [32, 44, 28, 34, 11]])
```

(4)

```
In [31]: my_arr[:,(2,3)]
```

```
Out[31]:
```

```
array([[13, 17],  
       [35, 46],  
       [ 4, 34],  
       [46, 22],  
       [47, 45],  
       [11, 25],  
       [28, 34],  
       [48, 21],  
       [36, 22],  
       [16, 14]])
```

Q9. slide 46의 그래프 작성 코드를 참조하여 다음의 월별 강수량을 그래프로 나타내시오

1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
20	22	37	79	90	109	288	277	140	50	48	19

source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
import matplotlib.pyplot as plt
%matplotlib inline
rain=np.array([20,22,37,79,90,109,288,277,140,50,48,19])
month=np.arange(1,13)
plt.plot(month,rain,color='black',label='$w=2$')
plt.title('$monthly precipitation$')
plt.xlabel('$month$')
plt.ylabel('$precipitation$')
plt.grid(True)
plt.show()
```

실행화면 캡처:

```
In [46]: rain=np.array([20,22,37,79,90,109,288,277,140,50,48,19])
...: month=np.arange(1,13)
...: plt.plot(month,rain,color='black',label='$w=2$')
...: plt.title('$monthly precipitation$')
...: plt.xlabel('$month$')
...: plt.ylabel('$precipitation$')
...: plt.grid(True)
...: plt.show()
```

