

สร้าง list ตัวเลขแบบอัตโนมัติ range()

```
In [ ]: range5_output = range(5)
        print(list(range5_output))

[0, 1, 2, 3, 4]
```

```
In [ ]: for i in range(100):
        print('งง',end=' ')

งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง
งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง
งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง งง
งง งง งง งง งง งง งง งง งง
```

```
In [ ]: list(range(1,11)) #เริ่มต้นเป็นวงทึบ จบเป็นวงกลมโปร่ง เหมือน List slicing
```

```
Out[ ]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [ ]: list(range(-3,20,4))
```

```
Out[ ]: [-3, 1, 5, 9, 13, 17]
```

loop in loop

สามารถใช้ for loop ซ้อนกันสองรอบได้

```
In [ ]: for name1 in range(2,5): #(2, 3, 4)
        print(f'now name1 = {name1}')
        for name2 in range(1,13):
            print(name1, ' x ',name2, ' = ',name1*name2)
        print('the inner loop is end')
```

```

now name1 = 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24
the inner loop is end
now name1 = 3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
3 x 11 = 33
3 x 12 = 36
the inner loop is end
now name1 = 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
4 x 11 = 44
4 x 12 = 48
the inner loop is end

```

loop in function

การวนลูปในฟังก์ชัน

```

In [ ]: def print_feelings(I,friendS,feeling = 'คิดถึง'):
        for friend in friendS:
            print(f'{I} {feeling} {friend}')

```

```

In [ ]: list_friends = ['พุ่ม', 'ดายน์', 'แดงโม', 'ฝน', 'พีเจ็ท', 'น้องแคมป์']

```

```

In [ ]: print_feelings('เจมส์',list_friends)

```

```

เจมส์ คิดถึง พุ่ม
เจมส์ คิดถึง ดายน์
เจมส์ คิดถึง แแดงโม
เจมส์ คิดถึง ฝน
เจมส์ คิดถึง พีเจ็ท
เจมส์ คิดถึง น้องแคมป์

```

```
In [ ]: print_feelings('เจมส์',list_friends,feeling='รัก')
```

```
เจมส์ รัก ฟินุม  
เจมส์ รัก ดายนน์  
เจมส์ รัก แดงโม  
เจมส์ รัก ฝน  
เจมส์ รัก พี่เจ้ท  
เจมส์ รัก น้องแคมป์
```

```
In [ ]: print_feelings('เจมส์',list_friends,'รัก')
```

```
เจมส์ รัก ฟินุม  
เจมส์ รัก ดายนน์  
เจมส์ รัก แดงโม  
เจมส์ รัก ฝน  
เจมส์ รัก พี่เจ้ท  
เจมส์ รัก น้องแคมป์
```

looping in dict

ทำการวนลูปในค่า dict โดยที่เรากำหนดตัว key และvalueขึ้นมา ได้ดังนี้

```
In [ ]: name_grade = {'พันทิพา' : 'B','ศุภนิดา' : 'C','ปณิตดา' : 'A','ทรงกลด' : 'F'}  
name_grade
```

```
Out[ ]: {'พันทิพา': 'B', 'ศุภนิดา': 'C', 'ปณิตดา': 'A', 'ทรงกลด': 'F'}
```

```
In [ ]: name_grade['ปณิตดา']
```

```
Out[ ]: 'A'
```

```
In [ ]: for i in name_grade.keys():  
        print(i)
```

```
พันทิพา  
ศุภนิดา  
ปณิตดา  
ทรงกลด
```

```
In [ ]: name_grade.keys()
```

```
Out[ ]: dict_keys(['พันทิพา', 'ศุภนิดา', 'ปณิตดา', 'ทรงกลด'])
```

```
In [ ]: for i in name_grade.keys():  
        print(f'{i} ได้เกรด {name_grade[i]}')
```

```
พันทิพา ได้เกรด B  
ศุภนิดา ได้เกรด C  
ปณิตดา ได้เกรด A  
ทรงกลด ได้เกรด F
```

conditional คือ เงื่อนไข /ประโยคเงื่อนไข

True / False

การวนลูปโดยมีเงื่อนไขต่าง ๆ มาเกี่ยวข้อง โดยมี pattern ดังนี้

```
if condition1 :  
    do something  
elif condition2 : #elif คือ else if
```

```

        do another thing
    else:
        do ...

```

```
In [ ]: print(list_friends)
```

```
['พินุม', 'ดายน์', 'แดงโม', 'ฝน', 'พีเจ็ท', 'น้องแคมป์']
```

```
In [ ]: for name1 in list_friends :
        for name2 in list_friends :
            print(name1, 'รัก', name2)
```

```

พินุม รัก พินุม
พินุม รัก ดายน์
พินุม รัก แแดงโม
พินุม รัก ฝน
พินุม รัก พีเจ็ท
พินุม รัก น้องแคมป์
ดายน์ รัก พินุม
ดายน์ รัก ดายน์
ดายน์ รัก แแดงโม
ดายน์ รัก ฝน
ดายน์ รัก พีเจ็ท
ดายน์ รัก น้องแคมป์
แดงโม รัก พินุม
แดงโม รัก ดายน์
แดงโม รัก แแดงโม
แดงโม รัก ฝน
แดงโม รัก พีเจ็ท
แดงโม รัก น้องแคมป์
ฝน รัก พินุม
ฝน รัก ดายน์
ฝน รัก แแดงโม
ฝน รัก ฝน
ฝน รัก พีเจ็ท
ฝน รัก น้องแคมป์
พีเจ็ท รัก พินุม
พีเจ็ท รัก ดายน์
พีเจ็ท รัก แแดงโม
พีเจ็ท รัก ฝน
พีเจ็ท รัก พีเจ็ท
พีเจ็ท รัก น้องแคมป์
น้องแคมป์ รัก พินุม
น้องแคมป์ รัก ดายน์
น้องแคมป์ รัก แแดงโม
น้องแคมป์ รัก ฝน
น้องแคมป์ รัก พีเจ็ท
น้องแคมป์ รัก น้องแคมป์

```

```
In [ ]: for name1 in list_friends:
        for name2 in list_friends:
            if name1 != name2: #ไม่ปรั้นชื่อคนเดียวกันซ้ำ
                print(name1, 'รัก', name2)
```

พี่นุม รัก ดายน์
 พี่นุม รัก แดงโม
 พี่นุม รัก ฝน
 พี่นุม รัก พี่เจี๊ยะ
 พี่นุม รัก น้องแคมป์
 ดายน์ รัก พี่นุม
 ดายน์ รัก แดงโม
 ดายน์ รัก ฝน
 ดายน์ รัก พี่เจี๊ยะ
 ดายน์ รัก น้องแคมป์
 แดงโม รัก พี่นุม
 แดงโม รัก ดายน์
 แดงโม รัก ฝน
 แดงโม รัก พี่เจี๊ยะ
 แดงโม รัก น้องแคมป์
 ฝน รัก พี่นุม
 ฝน รัก ดายน์
 ฝน รัก แดงโม
 ฝน รัก พี่เจี๊ยะ
 ฝน รัก น้องแคมป์
 พี่เจี๊ยะ รัก พี่นุม
 พี่เจี๊ยะ รัก ดายน์
 พี่เจี๊ยะ รัก แดงโม
 พี่เจี๊ยะ รัก ฝน
 พี่เจี๊ยะ รัก น้องแคมป์
 น้องแคมป์ รัก พี่นุม
 น้องแคมป์ รัก ดายน์
 น้องแคมป์ รัก แดงโม
 น้องแคมป์ รัก ฝน
 น้องแคมป์ รัก พี่เจี๊ยะ

```
In [ ]: for name1 in list_friends:
        for name2 in list_friends:
            if name1 == name2: #ปรี้นชื่อคนเดียวกันซ้ำ
                print(name1,'รัก',name2)
```

พี่นุม รัก พี่นุม
 ดายน์ รัก ดายน์
 แดงโม รัก แดงโม
 ฝน รัก ฝน
 พี่เจี๊ยะ รัก พี่เจี๊ยะ
 น้องแคมป์ รัก น้องแคมป์

```
In [ ]: for name1 in list_friends:
        for name2 in list_friends:
            if name1 == name2:
                'do nothing'
            else:
                print(name1,'รัก',name2)
```

พี่นุ้ม รัก ดายน์
 พี่นุ้ม รัก แดงโม
 พี่นุ้ม รัก ฝน
 พี่นุ้ม รัก พี่เจี๊ท
 พี่นุ้ม รัก น้องแคมป์
 ดายน์ รัก พี่นุ้ม
 ดายน์ รัก แดงโม
 ดายน์ รัก ฝน
 ดายน์ รัก พี่เจี๊ท
 ดายน์ รัก น้องแคมป์
 แดงโม รัก พี่นุ้ม
 แดงโม รัก ดายน์
 แดงโม รัก ฝน
 แดงโม รัก พี่เจี๊ท
 แดงโม รัก น้องแคมป์
 ฝน รัก พี่นุ้ม
 ฝน รัก ดายน์
 ฝน รัก แดงโม
 ฝน รัก พี่เจี๊ท
 ฝน รัก น้องแคมป์
 พี่เจี๊ท รัก พี่นุ้ม
 พี่เจี๊ท รัก ดายน์
 พี่เจี๊ท รัก แดงโม
 พี่เจี๊ท รัก ฝน
 พี่เจี๊ท รัก น้องแคมป์
 น้องแคมป์ รัก พี่นุ้ม
 น้องแคมป์ รัก ดายน์
 น้องแคมป์ รัก แดงโม
 น้องแคมป์ รัก ฝน
 น้องแคมป์ รัก พี่เจี๊ท

==, !=, >=, <=, <, >

```
In [ ]: 'พี่นุ้ม' == 'พี่เจี๊ท'
```

```
Out[ ]: False
```

```
In [ ]: 'พี่นุ้ม' != 'พี่เจี๊ท'
```

```
Out[ ]: True
```

หากหลังเงื่อนไข True ถูกต้องจะทำการ print ข้อความออกมาว่า Yes

```
In [ ]: if True:
        print('Yes')
```

Yes

```
In [ ]: if False:
        print('No')
```

หากหลังเงื่อนไข False ไม่ถูกต้องจะไม่ปรั้นข้อความใดๆออกมา

ซึ่งจะต้องใช้เงื่อนไขนี้ จึงจะได้ปรั้นค่าเงื่อนไขนั้นออกมา ดังนี้

```
In [ ]: if False:
        print('Yes')
        else:
        print('No')
```

No

HW

เขียน function ดัดเกรด โดยที่ input เป็นคะแนน(0-100) , output เป็นเกรด (F-A)

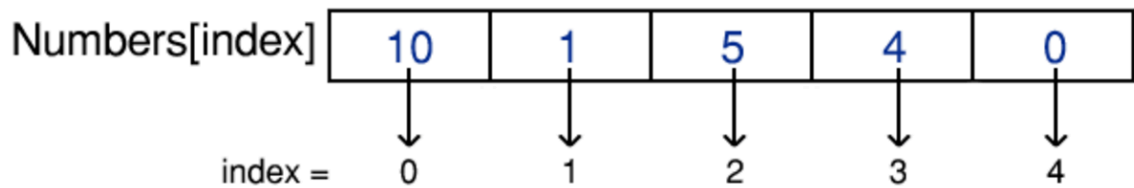
วนลูปดัดเกรด input = [1,50,65,90,101,-5,49]

กำหนด

ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 0 แต่น้อยกว่า 50 จะได้เกรด F
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 50 แต่น้อยกว่า 55 จะได้เกรด D
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 55 แต่น้อยกว่า 60 จะได้เกรด D+
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 60 แต่น้อยกว่า 65 จะได้เกรด C
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 65 แต่น้อยกว่า 70 จะได้เกรด C+
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 70 แต่น้อยกว่า 75 จะได้เกรด B
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 75 แต่น้อยกว่า 80 จะได้เกรด B+
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 80 จะได้เกรด A
แต่ว่าค่าคะแนนจะต้องไม่ต่ำกว่า 0 และมากกว่า 100 ไม่งั้นจะเกิด error

Finish

ที่เรียนไปแล้วมี 1 ตัวคือ list หน้าตาประมาณนี้



Dictionary

สามารถ index ด้วยอะไรก็ได้ แต่ลำดับจะหายไป

```
In [ ]: ex_dict = {'a':10, '1':1, 'stat':5} #curly brackets ในการบอกว่าเป็น dict
```

```
In [ ]: ex_dict['stat']
```

```
Out[ ]: 5
```

```
In [ ]: ex_dict[10]
```

```
-----
KeyError                                Traceback (most recent call last)
<ipython-input-8-008be89af7ca> in <module>()
----> 1 ex_dict[10]

KeyError: 10
```

```
In [ ]: ex2_dict = {1:'one', 2:'two', 3:'three'}
```

```
In [ ]: ex2_dict[2]
```

```
Out[ ]: 'two'
```

การเพิ่มสมาชิกใน dict

```
In [ ]: ex2_dict[0] = 'zero'
```

```
In [ ]: ex2_dict
```

```
Out[ ]: {0: 'zero', 1: 'one', 2: 'two', 3: 'three'}
```

คำสั่ง

```
In [ ]: ex2_dict.keys()
```

```
Out[ ]: dict_keys([1, 2, 3, 0])
```

```
In [ ]: for index in ex2_dict.keys():
        print(index)
```


1
2
3
0

```
In [ ]: ex2_dict.values()
```

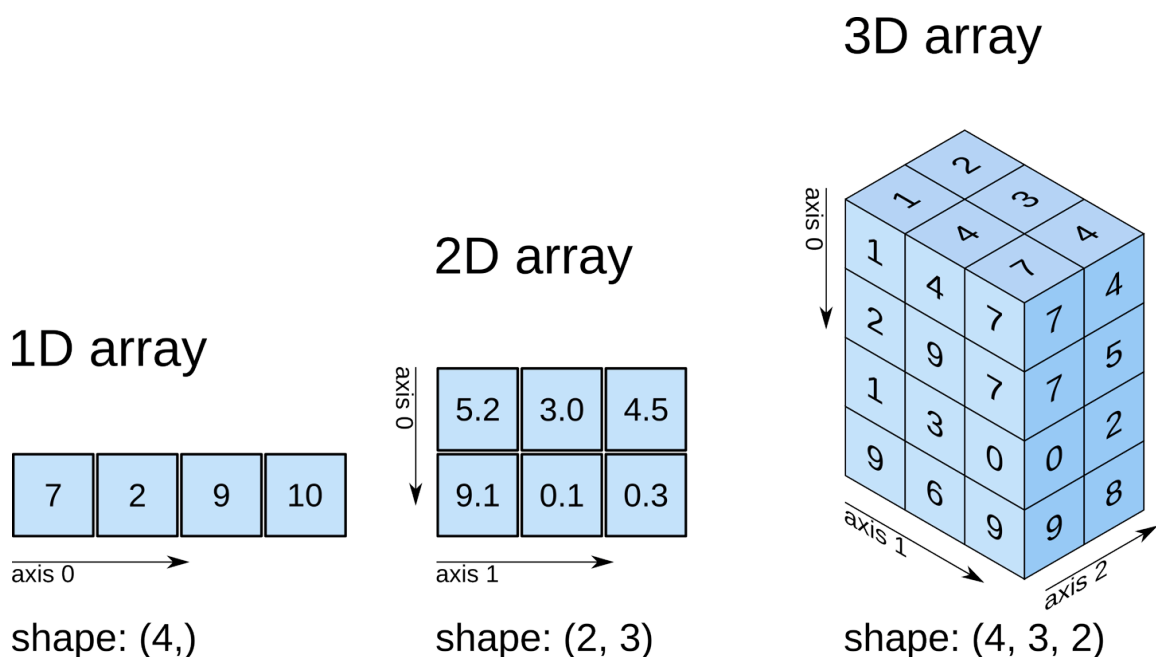
```
Out[ ]: dict_values(['one', 'two', 'three', 'zero'])
```

```
In [ ]: index
```

```
Out[ ]: 0
```

dict เอาไปใช้ร่วมกับ Dataframe ของ pandas

Numpy Array (array n มิติ)



numpy คือ package ที่ทำงานเกี่ยวกับ array ของตัวเลข (ที่มีคนเขียนขึ้นมาและนิยมใช้)

numpy -> number python

```
In [ ]: import numpy #วิธีเรียกใช้งาน package
```

สร้าง numpy array

```
In [ ]: ex_2d_array = numpy.array([[5.2, 3.0, 4.5], [9.1, 0.1, 0.3]])  
print(ex_2d_array)
```

```
[[5.2 3.  4.5]  
 [9.1 0.1 0.3]]
```

การชี้ค่าใน numpy array

```
In [ ]: ex_2d_array[1, 2]
```

Out[]: 0.3

```
In [ ]: list_x = [[5.2,3.0,4.5],[9.1,0.1,0.3]] #การชี้ list
print(len(list_x))
print(list_x[1])
print(list_x[1][2])

2
[9.1, 0.1, 0.3]
0.3
```

Operations

```
In [ ]: ex2_2d_array = numpy.array([[1,0,0],[0,0,1]])
print(ex2_2d_array)
print(ex2_2d_array)

[[1 0 0]
 [0 0 1]]
[[5.2 3.  4.5]
 [9.1 0.1 0.3]]
```

```
In [ ]: ex2_2d_array.shape #ตรวจสอบขนาดของ matrix
```

Out[]: (2, 3)

```
In [ ]: ex2_2d_array + ex2_2d_array
```

Out[]: array([[6.2, 3. , 4.5],
[9.1, 0.1, 1.3]])

```
In [ ]: list_x = [[5.2,3.0,4.5],[9.1,0.1,0.3]]
list_x2 = [[1,0,0],[0,0,1]]

list_x + list_x2
```

Out[]: [[5.2, 3.0, 4.5], [9.1, 0.1, 0.3], [1, 0, 0], [0, 0, 1]]

```
In [ ]: ex2_2d_array - ex2_2d_array
```

Out[]: array([[4.2, 3. , 4.5],
[9.1, 0.1, -0.7]])

```
In [ ]: ex2_2d_array * ex2_2d_array #การคูณแบบ array เอาตำแหน่งเดียวกันมาคูณกัน
```

Out[]: array([[5.2, 0. , 0.],
[0. , 0. , 0.3]])

matix multiplication (dot product)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \end{bmatrix}$$

```
In [ ]: numpy.dot(ex_2d_array,ex2_2d_array)
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-36-d923e9f0df4f> in <module>()  
----> 1 numpy.dot(ex_2d_array,ex2_2d_array)  
  
<__array_function__ internals> in dot(*args, **kwargs)  
ValueError: shapes (2,3) and (2,3) not aligned: 3 (dim 1) != 2 (dim 0)
```

```
In [ ]: print(ex_2d_array)  
        print(ex_2d_array.T)
```

```
[[5.2 3.  4.5]  
 [9.1 0.1 0.3]]  
[[5.2 9.1]  
 [3.  0.1]  
 [4.5 0.3]]
```

```
In [ ]: dot_mat = numpy.dot(ex_2d_array,ex2_2d_array.T)  
        print(dot_mat)
```

```
[[5.2 4.5]  
 [9.1 0.3]]
```

```
In [ ]: numpy.linalg.det(dot_mat)
```

```
Out[ ]: -39.389999999999986
```

search google -> stackoverflow หรือ web ของ package สำหรับ operations อื่นๆ

matrix slicing

```
In [ ]: print(ex_2d_array)
```

```
[[5.2 3.  4.5]  
 [9.1 0.1 0.3]]
```

```
In [ ]: ex_2d_array[1,1:]
```

```
Out[ ]: array([0.1, 0.3])
```

```
In [ ]: ex_2d_array[:, :2]
```

```
Out[ ]: array([[5.2, 3. ],  
               [9.1, 0.1]])
```

จบวันที่ 14 Jan 2021

HW 5 เขียน function คูณ matrix

แล้ว test กับ matrix ขนาด

- (2,3)*(3,2)
- (4,4)*(4,1)
- (2,2)*(2,2)

zeros, ones, random

```
In [ ]: import numpy as np #import numpy แล้วตั้งชื่อเป็น np
```

zeros

```
In [ ]: np.zeros(2)
```

```
Out[ ]: array([0., 0.])
```

```
In [ ]: np.zeros((2,3))
```

```
Out[ ]: array([[0., 0., 0.],
               [0., 0., 0.]])
```

ones

```
In [ ]: np.ones((2,3))
```

```
Out[ ]: array([[1., 1., 1.],
               [1., 1., 1.]])
```

Matrix Operation (scalar multiplication)

```
In [ ]: M_one = np.ones((2,3))
        2*M_one
```

```
Out[ ]: array([[2., 2., 2.],
               [2., 2., 2.]])
```

Random

```
In [ ]: np.random.rand(3,2) # uniform random [0,1)
```

```
Out[ ]: array([[0.12461684, 0.63204405],
               [0.240901   , 0.34341953],
               [0.22536518, 0.86663463]])
```

```
In [ ]: np.random.randn(3,2) # sample from normal distribution mean=0 std=1
```

```
Out[ ]: array([[ 2.06762285,  0.91239845],
               [-2.08011942, -0.46261935],
               [ 0.66804796,  1.19419422]])
```

```
In [ ]: np.random.choice([1,2,3,'a','b','c'])
```

```
Out[ ]: 'a'
```

```
In [ ]: np.random.choice([1,2,3,28,11,100],size = 2)
```

```
Out[ ]: array([28,  3])
```

```
In [ ]: np.random.choice([1,2,3,28,11,100],size = (2,3))
```

```
Out[ ]: array([[ 3,  3, 11],
               [28,  1, 11]])
```

```
In [ ]: np.random.choice([1,2,3,28,11,100], size = (2,3), replace=False)
```

```
Out[ ]: array([[ 3, 11,  2],
               [100, 28,  1]])
```

```
In [ ]: np.random.choice(['A','B','C','D','F'],size=35, p = [0.1,0.2,0.4,0.29,0.01])
```

```
Out[ ]: array(['D', 'C', 'C', 'C', 'C', 'C', 'B', 'C', 'B', 'D', 'A', 'C', 'C',
               'D', 'A', 'C', 'C', 'D', 'D', 'B', 'C', 'A', 'C', 'C', 'C', 'D',
               'A', 'D', 'C', 'D', 'B', 'D', 'C', 'D', 'C'], dtype='<U1')
```

เลข HW Matrix Multiplication

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

```
In [ ]: def mat_mul(A,B):
        C = np.zeros((A.shape[0],B.shape[1]))#เตรียมขนาดของ matrix ผลลัพธ์
        for r_a in range(A.shape[0]):
            for c_b in range(B.shape[1]):
                for every_element in range(A.shape[1]):
                    C[r_a,c_b] = C[r_a,c_b] + (A[r_a,every_element] * B[every_element,c_b])
        return C
```

```
In [ ]: mat1 = np.array([[1,2,3],[4,5,6]])
        mat2 = np.array([[7,8],[9,10],[11,12]])
```

```
In [ ]: mat_mul(mat1,mat2)
```

```
Out[ ]: array([[ 58.,  64.],
               [139., 154.]])
```

```
In [ ]:
```