



# GBDi

Government Big Data Institute

สถาบันส่งเสริมการวิเคราะห์และบริหารข้อมูลขนาดใหญ่ภาครัฐ (สวช.)



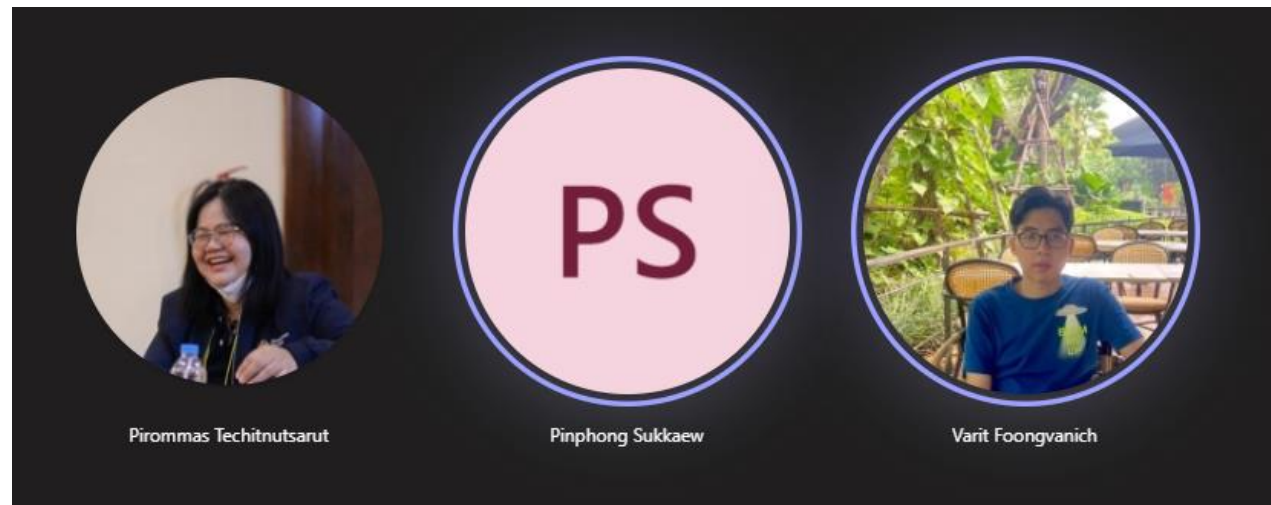
# Python Basics (2)

May 25, 2022

Chainarong Tumapha

# Agreements

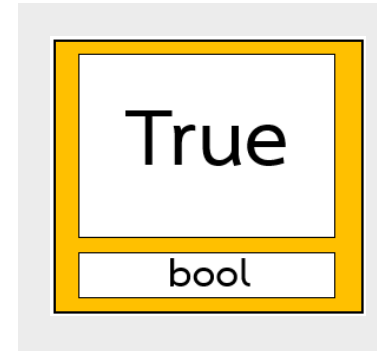
1. ท้ายคาบจะให้ส่งลิงค์ colab ผ่าน Form โดยจะต้องมีการทำ lab ในทุก lab (ไม่นับ extra)
2. code เฉลยจะถูกส่งให้หลังจากเลยกำหนดส่ง colab แล้ว
3. สุ่มแจกของรางวัลกับผู้ส่ง colab ในข้อแรกตามเวลาที่กำหนด
4. กรณีที่มีปัญหาให้ทักถาม TA หรือถามในแชทได้เลย



# Basic Data Types



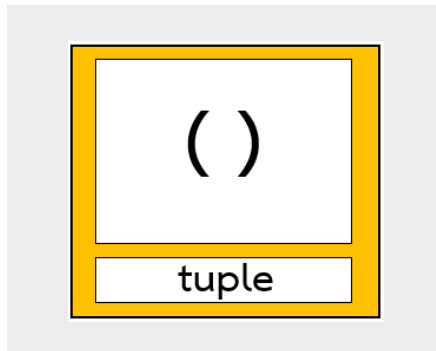
Numbers



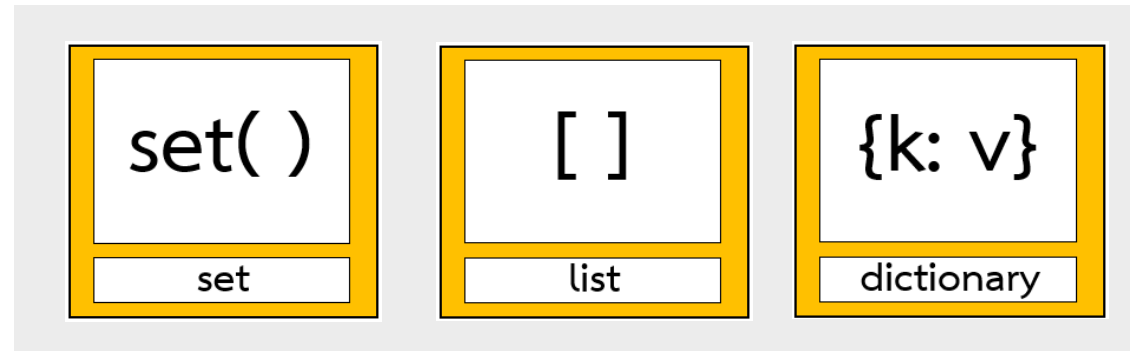
Logical values



Texts

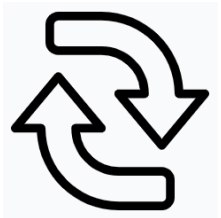
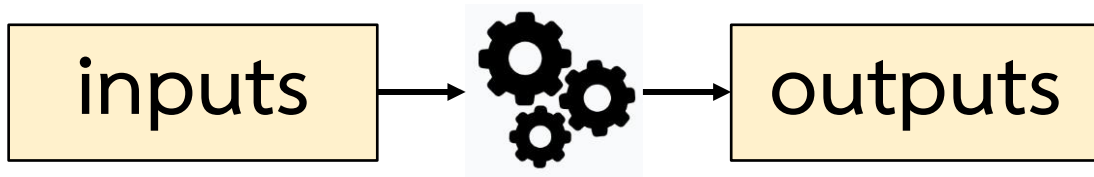


Immutable Containers  
(เปลี่ยนค่าข้างในไม่ได้)



Mutable Containers  
(เปลี่ยนค่าข้างในได้)

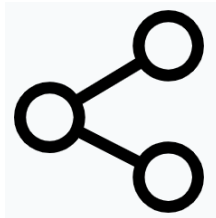
# Functions



Reuse



Shorten



Share

Please visit [this link](#) for more information.

Function declaration

Parameters

```
def function_name(input1, input2, ...):  
    '''  
    docstring: description of the function  
    '''  
    statement1  
    statement2  
    ...  
    return output1, output2, ...
```

Function called

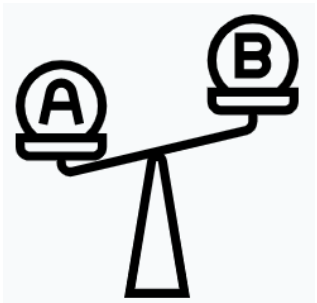
Arguments

```
x = function_name(input1, input2, ...)
```

# Logical Values: bool (Boolean)



Can be either  
True or False



Obtained from  
comparison

Operation	Operator	Example
is greater than	<code>&gt;</code>	<code>x &gt; y</code>
is less than	<code>&lt;</code>	<code>x &lt; y</code>
is greater than or equal to	<code>&gt;=</code>	<code>x &gt;= y</code>
is less than or equal to	<code>&lt;=</code>	<code>x &lt;= y</code>
is equal to	<code>==</code>	<code>x == y</code>
is different from	<code>!=</code>	<code>x != y</code>

The equals sign here does not refer to the assignment operator.

# For-Loop

```
for var in container:  
    statement1  
    statement2  
    ...
```

Count-controlled loop

```
for char in 'Python!':  
    print(char)
```

P  
y  
t  
h  
o  
n  
!

String

```
for item in ('This', True, 3.14, -2):  
    print(item)
```

This  
True  
3.14  
-2

Tuple

```
for num in [1, 2, 3, 4, 5]:  
    print(num)
```

range(1, 6)

1  
2  
3  
4  
5

List

```
for elm in {'A', 'B'}:  
    print(elm)
```

B  
A

Set

The exact syntax of the range function is range(start, stop, step).

# Examples: While-Loop

```
while condition:
    statement1
    statement2
    ...
```

Event-controlled loop

```
idx = 0
text = 'Python!'
while idx < len(text):
    print(text[idx])
    idx += 1
```

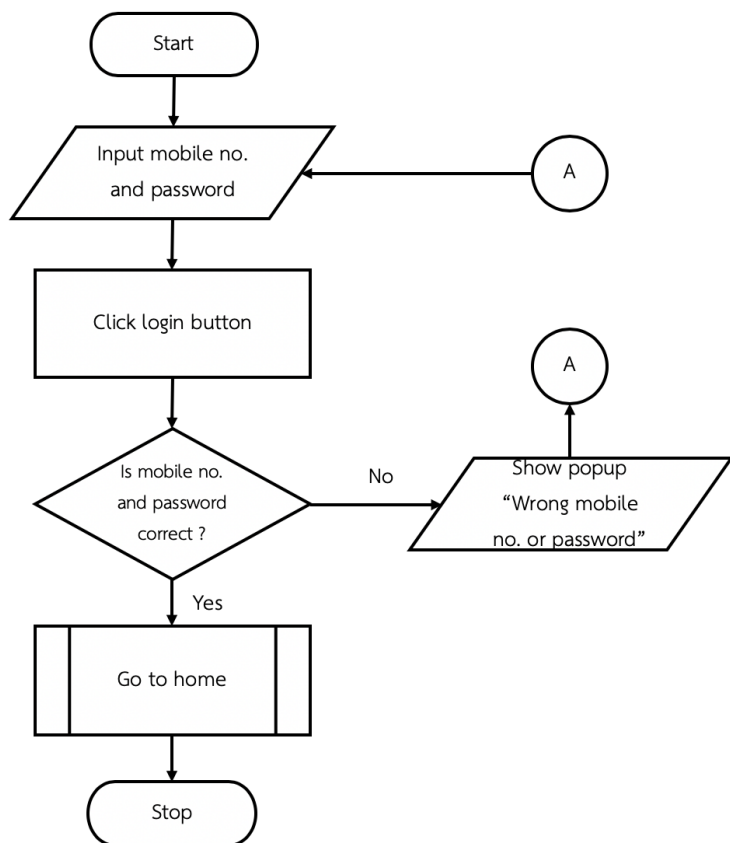
P  
y  
t  
h  
o  
n  
!

```
num = 1
while num < 6:
    print(num)
    num += 1
```

1  
2  
3  
4  
5

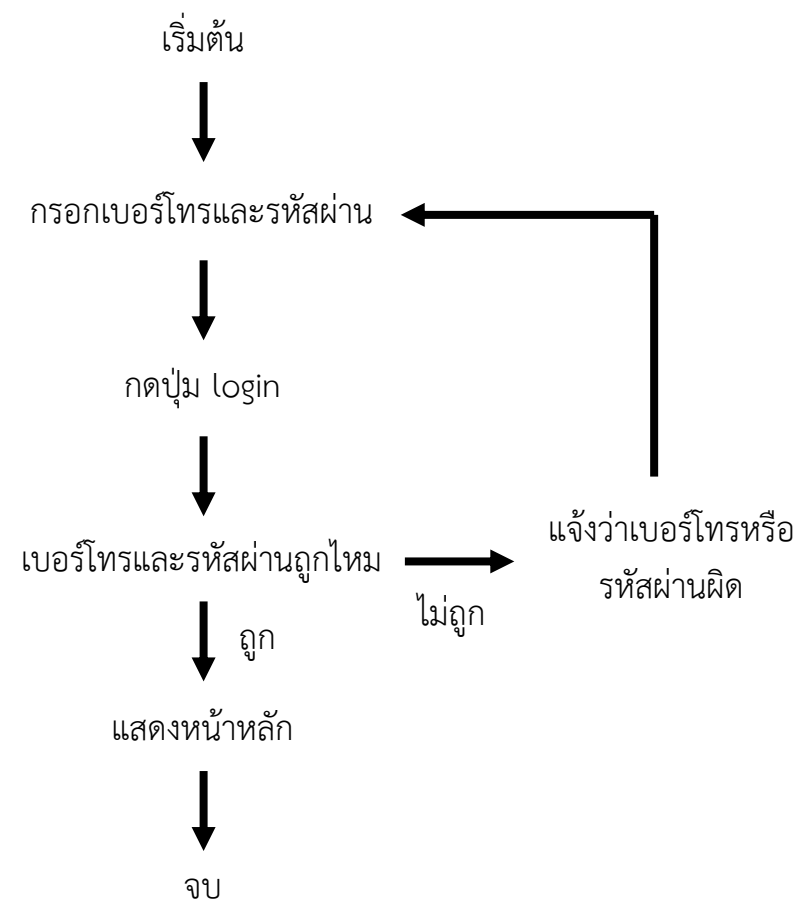
# Programming design

## ออกแบบฟังก์ชันการลงชื่อเข้าใช้งาน



<https://blog.nextzy.me/flowchart-101-f3ef25c3a92a>

เขียนให้ง่ายแต่  
ไม่เป็นทางการ





# Previous Quiz

# Quiz แจกซอง (25 min.)

# LAB13 : num2word งานกลุ่ม (20 Min.)

ออกแบบฟังก์ชัน num2word ในการแปลงจำนวนเต็มที่ไม่เกิน 1 ล้านเป็นข้อความ  
ตัวแทนกลุ่ม capture ภาพ/ขั้นตอนการทำงานของฟังก์ชัน โพสต์ในแชทห้องรวม

Extra: เขียน code ตามแบบที่คิดไว้

```
num2word(123451)
```

'หนึ่งแสนสองหมื่นสามพันสี่ร้อยห้าสิบเอ็ด'

```
num2word(2021)
```

'สองพันยี่สิบเอ็ด'

```
num2word(1000001)
```

กรุณาใส่จำนวนเต็มที่มีค่าไม่เกิน 1 ล้าน

```
num2word('Hello')
```

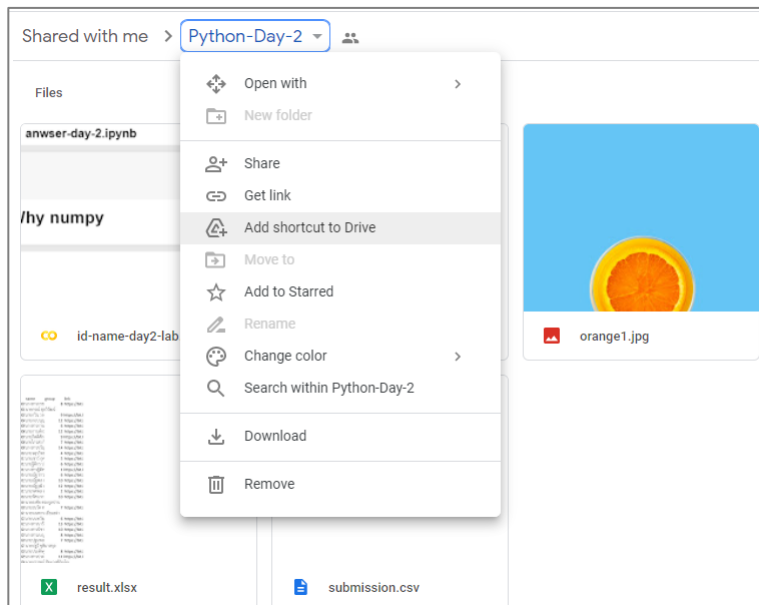
กรุณาใส่จำนวนเต็มที่มีค่าไม่เกิน 1 ล้าน

# LAB13 Sharing (5-10 min.)

Break (15 min.)

# Examples: Pandas Attributes (10 min.)

1. เข้าไปที่ drive
2. กด add shortcut to drive
3. กดอีกที
4. ไปที่ drive
5. เลือก
6. เปิดไฟล์ .ipynb
7. แก้ PATH\_FOLDER
8. รันโค้ดกล่องแรก
9. กด Run anyway
10. Connect to Google Drive
11. เลือก Account
12. กด Allow
13. รันกล่อง Why numpy ต้องเห็นรูปส้ม



```
from google.colab import drive
drive.mount('/content/drive')
```

```
PATH_FOLDER = '/content/drive/MyDrive/Python-Day-2/'
```

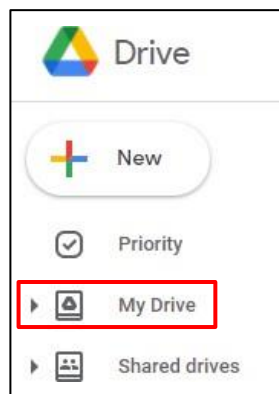
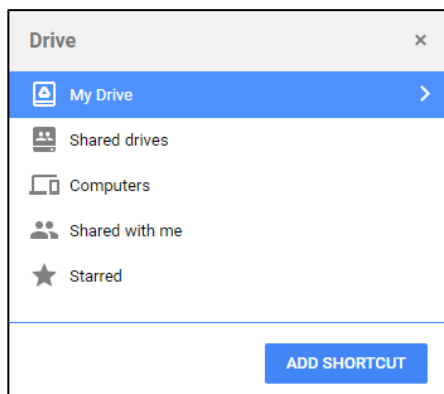
Warning: This notebook was not authored by Google.

This notebook was authored by **cbright130@gmail.com**. It may request access to your data stored with Google, or read data and credentials from other sessions. Please review the source code before executing this notebook. Please contact the creator of this notebook at cbright130@gmail.com with any additional questions.

Cancel Run anyway

## Why numpy

2 cells hidden



Permit this notebook to access your Google Drive files?

This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

No thanks Connect to Google Drive



Choose an account from depa.or.th

to continue to Google Drive for desktop

Chainarong Tumapha  
chainarong.tu@depa.or.th

Make sure you trust Google Drive for desktop

You may be sharing sensitive info with this site or app. You can always see or remove access in your [Google Account](#).

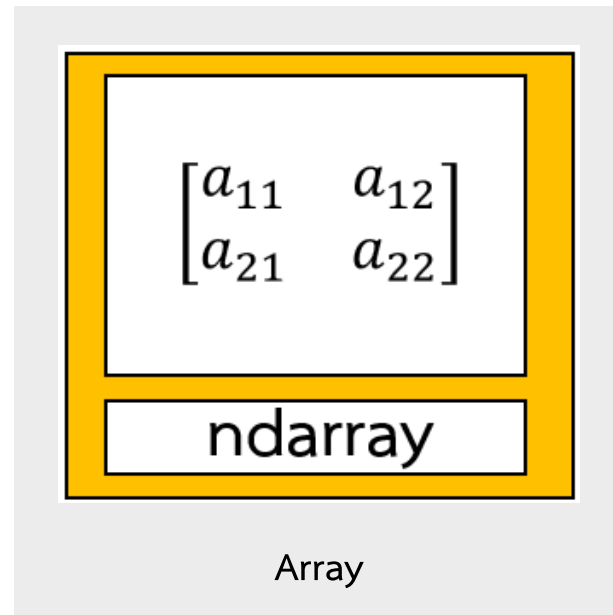
Learn how Google helps you [share data safely](#).

See Google Drive for desktop's [Privacy Policy](#) and [Terms of Service](#).

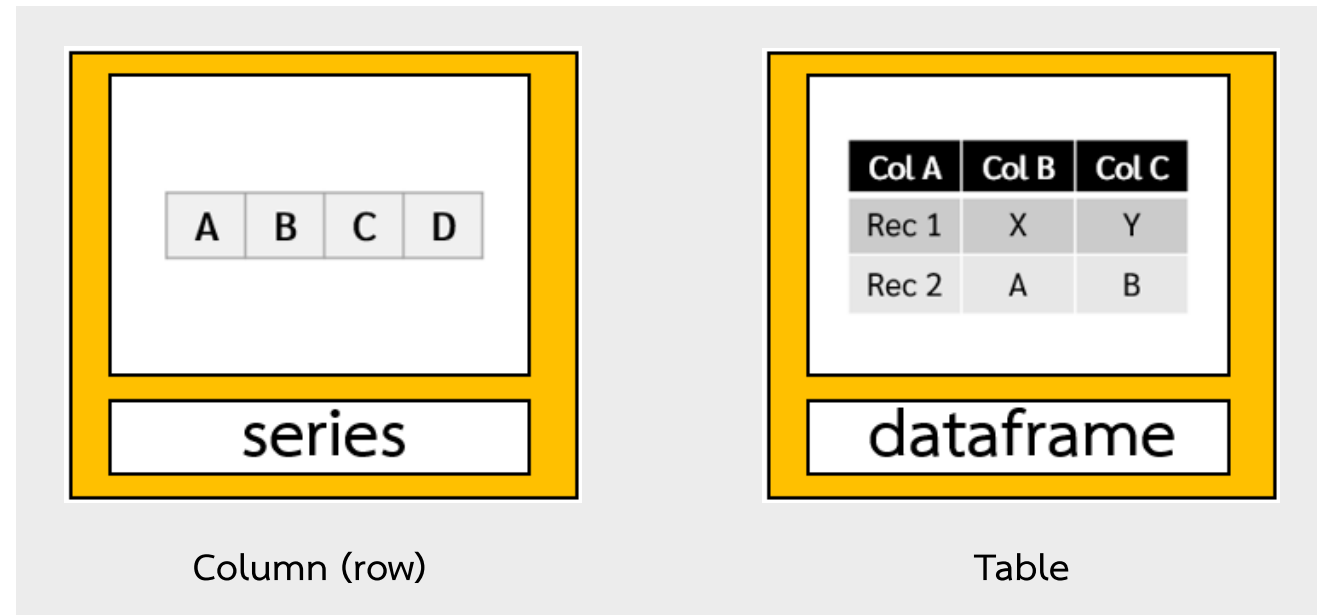
Cancel

Allow

# New Data Types



NumPy



Pandas

# Why NumPy?

```
# การคูณค่าใน list 2 ชั้น
x = [[1,0,0],
      [0,1,2]]

y = 2

for row_idx in range(len(x)) :
    for col_idx in range(len(x[row_idx])):
        x[row_idx][col_idx] *= 2

x

[[2, 0, 0], [0, 2, 4]]
```



```
# ใช้ Numpy เพื่อชีวิตที่ดีกว่า
import numpy as np
x = [[1,0,0],
      [0,1,2]]
np.array(x) * 2
```

```
array([[2, 0, 0],
       [0, 2, 4]])
```

$$x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \end{bmatrix} \longrightarrow \begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 2. \end{bmatrix}$$

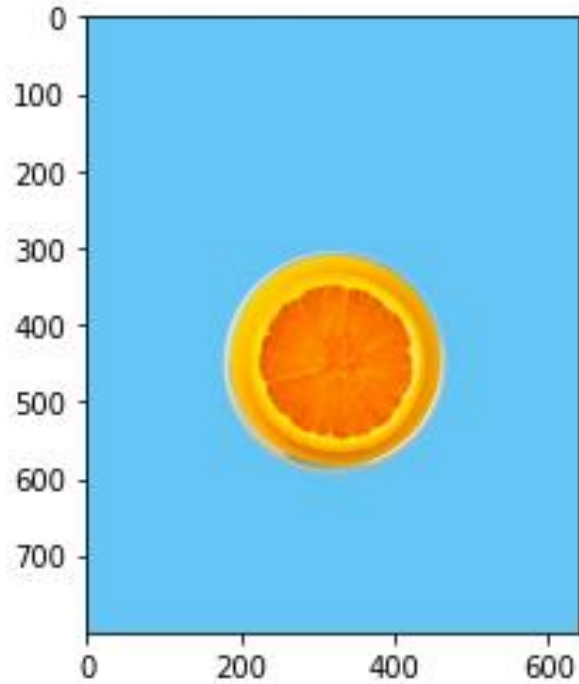
Operation	Syntax	Output	Description
Size	x.size	6	Return number of elements in the array.
Shape	x.shape	(2, 3)	Return the dimension of the array.
Type	x.dtype	dtype('int64')	Return the data type of elements in the array.

Please visit [this link](#) for more information.



# Why NumPy?

(800, 640, 3)



```
array([[101, 197, 245],  
       [101, 197, 245],  
       [101, 197, 245],  
       ...,  
       [101, 197, 245],  
       [101, 197, 245],  
       [101, 197, 245]],  
  
       [[101, 197, 245],  
        [101, 197, 245],  
        [101, 197, 245],  
        ...,  
        [101, 197, 245],  
        [101, 197, 245],  
        [101, 197, 245]],  
  
       [[101, 197, 245],  
        [101, 197, 245],  
        [101, 197, 245],  
        ...,  
        [101, 197, 245],  
        [101, 197, 245],  
        [101, 197, 245]],  
       ...,  
       ...,  
       ...]
```

`type(image)`

`numpy.ndarray`

ค่าสี Red Green Blue มีช่วงตั้งแต่ 0 (มืด) ถึง 255 (สว่าง)

# Array Creation: Type Casting

```
import numpy as np
```

```
np.array( Array_like )
```

```
a = np.array([1, 2, 3, 4, 5])
```

```
b = np.array((1, 2, 3, 4, 5))
```

```
c = np.array(['H', 'e', 'l', 'l', 'o'])
```

ผลลัพธ์

```
[1 2 3 4 5]
```

```
[1 2 3 4 5]
```

```
['H' 'e' 'l' 'l' 'o']
```

# Array Creation: Type Casting

```
x = np.array([[1, 2], [3, 4], [5, 6]])
```

ผลลัพธ์

```
[[1 2]
 [3 4]
 [5 6]]
```

```
y = np.array([[1, 2, 3], [4, 5, 6]])
```

```
[[1 2 3]
 [4 5 6]]
```

```
z = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
```

```
[[[1 2]
   [3 4]]
```

```
 [[5 6]
  [7 8]]]
```

# Array Creation: Other Built-In Functions

Syntax	Example	Description
<code>np.arange( start , stop , step )</code>	<code>np.arange(10, 0, -1)</code>	Generate value within the interval
<code>np.linspace( start , stop , num )</code>	<code>np.linspace(1, 10, 10)</code>	Generate value equally spaced samples in the closed interval
<code>np.zeros( shape )</code>	<code>np.zeros((3, 4))</code>	Generate a new array of given shape, filled with zeros
<code>np.ones( shape )</code>	<code>np.ones((3, 8))</code>	Generate a new array of given shape, filled with ones

```
np.arange(10,0,-1)
```

```
array([10,  9,  8,  7,  6,  5,  4,  3,  2,  1])
```

```
np.linspace(1,10,10)
```

```
array([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
np.zeros((3,4))
```

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

```
np.ones((3,8))
```

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

# Array Creation: Other Built-In Functions

Syntax	Example	Description	
<code>np.stake( Array_like , axis )</code>	<code>np.stake([[1,2,3,4]], axis=1)</code>	Join a sequence of arrays along a new axis	
<code>np.eye( n )</code>	<code>np.eye(3)</code>	Generate 2-D array with ones on the diagonal and zeros elsewhere	
<code>np.random.randint( start , stop , size )</code>	<code>np.random.randint(1, 10, size=10)</code>	Random integers from low (inclusive) to high (exclusive)	
<code>np.stack([1,2,3,4])</code> <code>array([1, 2, 3, 4])</code>	<code>np.stack([[1,2,3,4]],axis=1)</code> <code>array([[1], [2], [3], [4]])</code>	<code>np.eye(3)</code> <code>array([[1., 0., 0.], [0., 1., 0.], [0., 0., 1.]])</code>	<code>np.random.randint(1,10,size=10)</code> <code>array([1, 8, 5, 2, 1, 7, 9, 7, 9, 3])</code>

# Array Methods

Operation	Syntax	Description
Reshape	<code>x.reshape( shape )</code>	Change dimension of the array.
Transpose	<code>x.T</code>	Transpose the array.
Change data type	<code>x.astype( type )</code>	Convert elements of the array.

```
x = np.arange(1,5,1)
x = x.reshape((2,2))
x
array([[1, 2],
       [3, 4]])
```

```
x.T
array([[1, 3],
       [2, 4]])
```

```
x.astype(float)
array([[1., 2.],
       [3., 4.]])
```

# LAB14: Array Creation (15 min.)

ให้บอกคำสั่งที่ใช้ในการสร้าง array ดังต่อไปนี้ โดยไม่ใช่ np.array()

1. 

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

2. 

```
[[0 0 0]
 [0 0 0]
 [0 0 0]
 [0 0 0]]
```

3. 

```
[[1 0]
 [0 1]]
```

4. 

```
[[0.1 0.2 0.3 0.4 0.5]
 [0.6 0.7 0.8 0.9 1. ]]
```

# Array Slicing

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

```
x[0:9:2]
```

```
array([1, 3, 5, 7, 9])
```

```
x[x>7]
```

```
array([ 8,  9, 10, 11, 12])
```

```
x[[1, 3, 5, 7]]
```

```
array([2, 4, 6, 8])
```

```
x[x%2==0]
```

```
array([ 2,  4,  6,  8, 10, 12])
```

```
x[[True, False, True, False,  
   True, False, True, False,  
   True, False, True, False]]
```

```
x[(x>5) & (x%2==0)]
```

```
array([ 6,  8, 10, 12])
```

```
x[(x<3) | (x>10)]
```

```
array([ 1,  2, 11, 12])
```

```
array([ 1,  3,  5,  7,  9, 11])
```

```
x[x>15]
```

```
array([], dtype=int64)
```



# Array Indexing

```
x = np.array([[10,6,5],  
              [14,8,12],  
              [5,2,6],  
              [13,2,8],  
              [12,15,15]  
            ])
```

```
x[1,2]
```

```
12
```

```
x[0:2,2]
```

```
array([ 5, 12])
```

```
x[:,1:2]
```

```
array([[ 6],  
       [ 8],  
       [ 2],  
       [ 2],  
       [15]])
```

# Array Method: Statistics

`x.sum()`

`x.mean()`

`x.max()`

`x.std()`

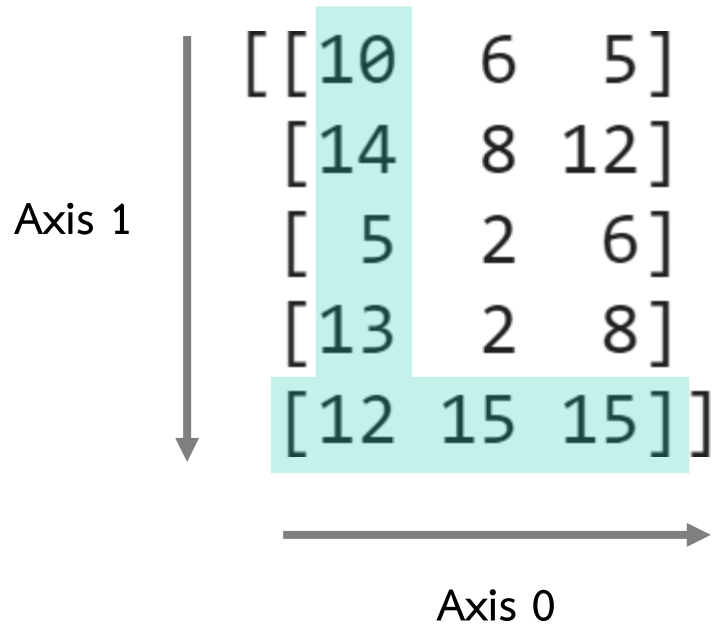
`x.min()`

```
x = np.array([[10,6,5],
              [14,8,12],
              [5,2,6],
              [13,2,8],
              [12,15,15]
              ])
```

```
print(x.sum())
print(x.max())
print(x.min())
print(x.mean())
print(x.std())
```

```
133
15
2
8.866666666666667
4.333846123505335
```

# Array Method: Axis



```
x.max()
```

15

```
x.max(axis=0)
```

array([14, 15, 15])

```
x.max(axis=1)
```

array([10, 14, 6, 13, 15])

# Array Operations

```
x = np.array([-5, -2, -5, 1, -3])
y = np.array([3, -2, 2, 0, 3])
```

$x + 1$

array([-4, -1, -4, 2, -2])

`np.square(x)`

array([25, 4, 25, 1, 9])

$2 * x$

array([-10, -4, -10, 2, -6])

`np.dot(x, y)`

-30

$x + y$

array([-2, -4, -3, 1, 0])

`np.log(np.abs(x))`

array([1.60943791, 0.69314718, 1.60943791, 0. , 1.09861229])

# LAB15: Array Method: Statistics (15 min.)

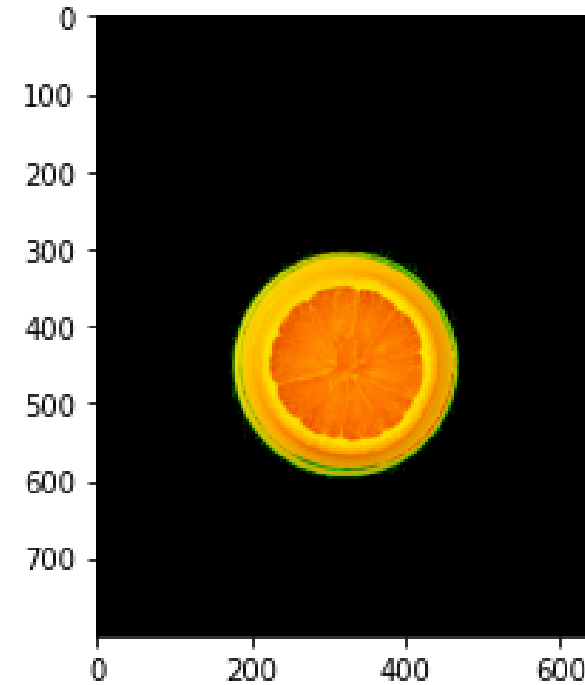
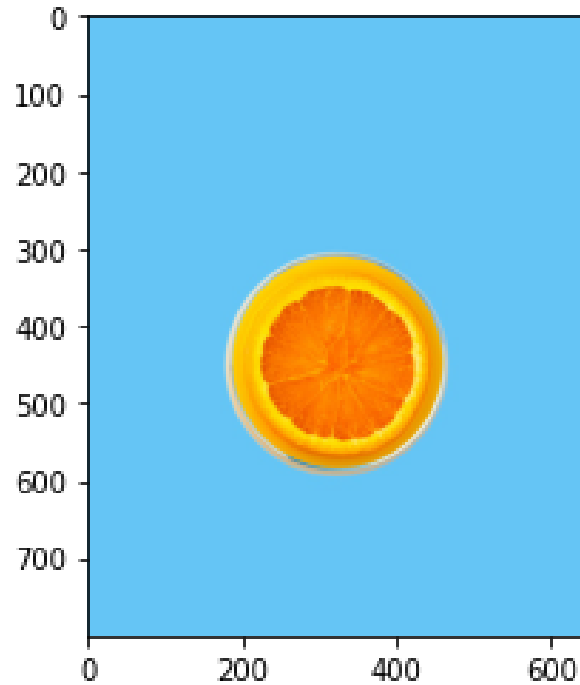
1. หาผลรวมของจำนวนเต็มตั้งแต่ 1 ถึง 500 โดยใช้ NumPy

2. หาผลรวมของจำนวนคู่ตั้งแต่ 1 ถึง 500 โดยใช้ NumPy

3. หาผลรวมของ x ที่มากกว่า 10 โดยใช้ NumPy

```
x = np.array([[10,6,5],  
              [14,8,12],  
              [5,2,6],  
              [13,2,8],  
              [12,15,15]  
              ])
```

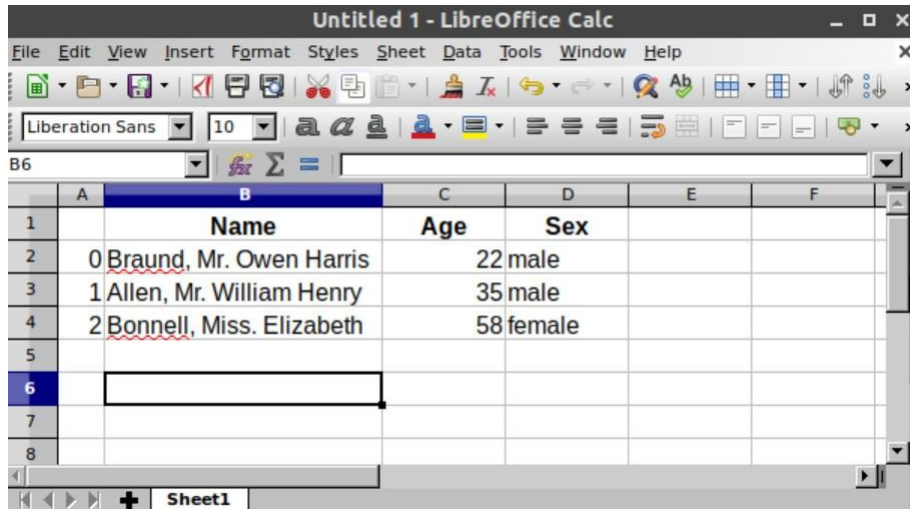
# Extra : A orange should be orange. (before end of class)



ให้ลบ background ของรูปด้านซ้ายออก

Hint: - พื้นหลังจะดำก็ต่อเมื่อ r,g,b เป็น 0 ทั้งหมด  
- สีส้มเกิดจากส่วนผสมของสีแดงและสีเหลือง

# Introduction to Pandas

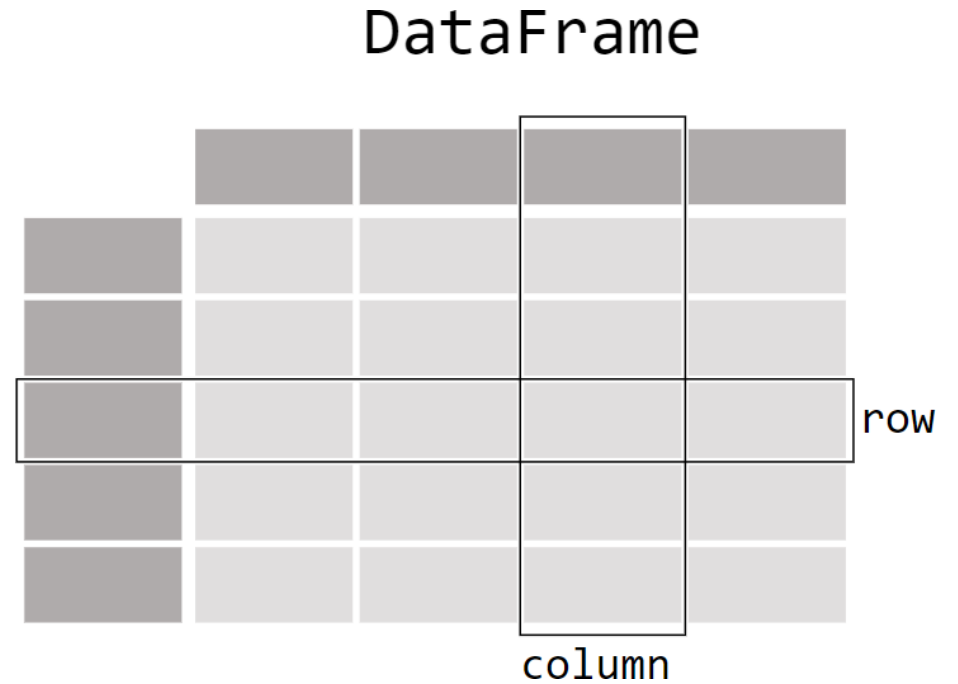


A screenshot of a LibreOffice Calc spreadsheet titled 'Untitled 1 - LibreOffice Calc'. The spreadsheet has columns A through F and rows 1 through 8. The data is as follows:

	A	B	C	D	E	F
1		Name	Age	Sex		
2	0	Braund, Mr. Owen Harris	22	male		
3	1	Allen, Mr. William Henry	35	male		
4	2	Bonnell, Miss. Elizabeth	58	female		
5						
6						
7						
8						



DataFrame



A diagram showing a 6x4 grid of cells, representing a DataFrame. The grid is composed of light gray cells. A black border highlights the third row and fourth column. The word 'row' is written to the right of the highlighted row, and the word 'column' is written below the highlighted column.


```
import pandas as pd
```

# Pandas: Series

```
pd.Series( container, index )
```

```
x = pd.Series(np.arange(4))
```

0	0	ant	0
1	1	bird	1
2	2	cat	2
3	3	dog	3
dtype: int64		dtype: int64	

```
x = pd.Series(np.arange(4),  
              index=['ant', 'bird', 'cat', 'dog'])
```

```
x = pd.Series({'ant': 0, 'bird': 1, 'cat': 2, 'dog': 4})
```



# Series Attributes

```
x.loc[['ant', 'dog']]
```

```
ant    0
dog    4
dtype: int64
```

```
x.unique()
```

```
array([0, 1, 2, 4])
```

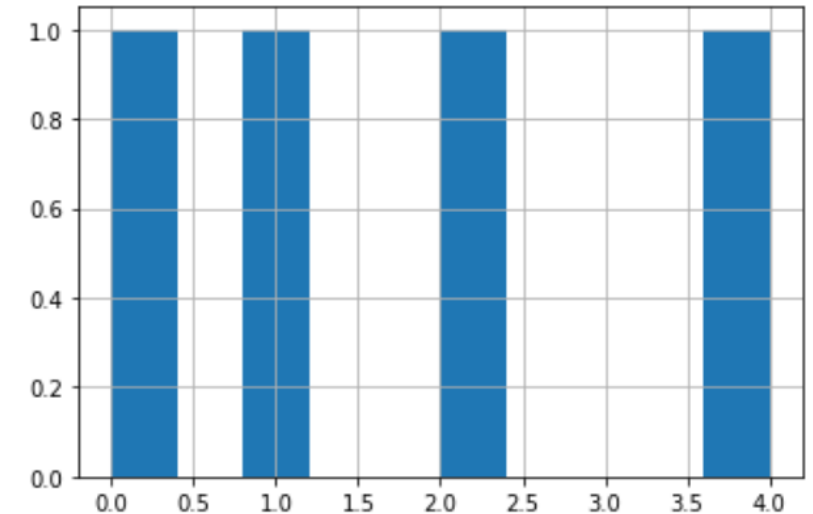
```
x.describe()
```

```
count    4.000000
mean     1.750000
std      1.707825
min      0.000000
25%      0.750000
50%      1.500000
75%      2.500000
max      4.000000
dtype: float64
```

```
x.value_counts()
```

```
4    1
2    1
1    1
0    1
dtype: int64
```

```
x.hist()
```



# LAB16: apply (5 min.)

```
x.apply(lambda x: x ** 2)
```

```
ant      0
bird     1
cat      4
dog     16
dtype: int64
```

```
def squire(x):
    '''
    ยกกำลังสอง
    input: x - int
    output: y - int
    '''
    y = x ** 2
    return y
```

```
x.apply(squire)
```

```
def power(x, power):
    y = x**power
    return y

x.apply(power, args=(3,))
```

```
ant      0
bird     1
cat      8
dog     64
dtype: int64
```

ใช้คำสั่ง apply ในการคำนวณเศษจากการหาค่าของ x ด้วย 3 ด้วยการเขียนแบบ lambda function และการประกาศ function แยก

# Pandas: DataFrame

```
x = pd.DataFrame(np.arange(15).reshape((5, 3)),
                  columns=['Col1', 'Col2', 'Col3'])
```

	Col1	Col2	Col3
0	0	1	2
1	3	4	5
2	6	7	8
3	9	10	11
4	12	13	14

```
x = pd.DataFrame({'Name': ['Alice', 'Bob'], 'Wallet': [250, 350]})
```

	Name	Wallet
0	Alice	250
1	Bob	350

```
x['Expense'] = [40, 225]
```

	Name	Wallet	Expense
0	Alice	250	40
1	Bob	350	225

```
x['Total'] = x['Wallet'] - x['Expense']
```

	Name	Wallet	Expense	Total
0	Alice	250	40	210
1	Bob	350	225	125

```
pd.read_excel( file )
```

```
pd.read_csv( file )
```

# LAB17: DataFrame (10 min.)

1. สร้าง DataFrame ที่มีลักษณะดังต่อไปนี้ และเก็บอยู่ในตัวแปรชื่อ x

2. สร้างคอลัมน์ C เพื่อเก็บค่าผลบวกของค่าในคอลัมน์ A และ B ในแต่ละแถว

3. สร้างคอลัมน์ D เพื่อเก็บค่าผลคูณของค่าในคอลัมน์ A, B และ C

	A	B
0	1	-5
1	2	-4
2	3	-3
3	4	-2
4	5	-1

# DataFrame Slicing

	A	B
0	1	-10
1	2	-9
2	3	-8
3	4	-7
4	5	-6
5	6	-5
6	7	-4
7	8	-3
8	9	-2
9	10	-1

```
x.loc[[1, 3, 7]]
```

```
x.loc[[1, 3, 7], ['B']]
```

```
x.loc[[1, 3, 7], 'B']
```

```
type(x.loc[[1, 3, 7], 'B'])  
pandas.core.series.Series
```

```
x['A']
```

```
x[x['A'] > 8]
```

	A	B
8	9	-2
9	10	-1

	A	B
1	2	-9
3	4	-7
7	8	-3

	B
1	-9
3	-7
7	-3

```
1    -9  
3    -7  
7    -3  
Name: B, dtype: int64
```

```
0     1  
1     2  
2     3  
3     4  
4     5  
5     6  
6     7  
7     8  
8     9  
9    10  
Name: A, dtype: int64
```

# Pandas Attributes: drop

```
x.drop(index=5)
```

```
x.drop(5,axis=0)
```

	A	B
0	1	-10
1	2	-9
2	3	-8
3	4	-7
4	5	-6
6	7	-4
7	8	-3
8	9	-2
9	10	-1



```
x.drop(columns='B')
```

```
x.drop('B',axis=1)
```

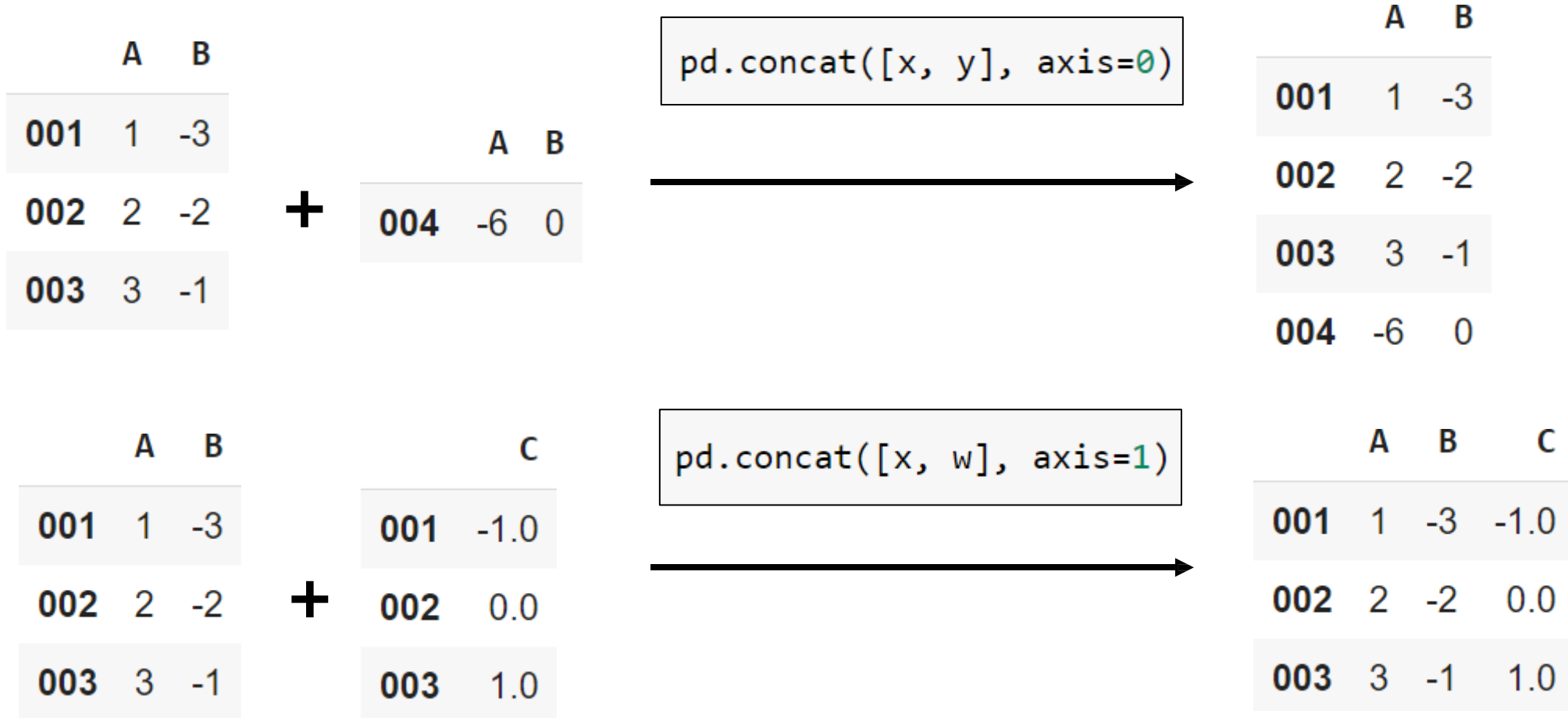
	A
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10



```
x.drop(5, axis=0, inplace=True)
```

```
x.drop('B', axis=1, inplace=True)
```

# Pandas Attributes: concat



# Pandas Attributes: merge

ID   Value A				ID   Value B		
0	PY101	-35	+	0	PY132	List
1	PY243	23		1	PY155	Tuple
2	PY132	36		2	PY101	Array

```
x.merge(y, on='ID', how='')
```

	ID	Value A	Value B
0	PY101	-35	Array
1	PY132	36	List

inner

	ID	Value A	Value B
0	PY101	-35.0	Array
1	PY243	23.0	NaN
2	PY132	36.0	List
3	PY155	NaN	Tuple

outer

	ID	Value A	Value B
0	PY101	-35	Array
1	PY243	23	NaN
2	PY132	36	List

left

	ID	Value A	Value B
0	PY132	36.0	List
1	PY155	NaN	Tuple
2	PY101	-35.0	Array

right



# LAB18: Grouping (75 min.)

ให้สร้างตารางกลุ่มและลิงค์สำหรับเข้าห้องสนทนาของผู้เข้าร่วมอบรมโดยมีไฟล์ที่เกี่ยวข้องดังนี้

1. participants.csv เก็บรายชื่อผู้เข้าร่วมอบรม
2. submission.csv เก็บรายชื่อกลุ่มของผู้เข้าอบรมที่กรอกแบบตอบรับเมื่อ class ที่แล้ว (หมายความว่า มีข้อมูลแค่บางคน) โดยต้องแก้ไขปัญหาข้อมูลซ้ำ, ข้อมูล pattern ไม่ตรงด้วย
3. link.csv เก็บลิงค์ ms team ของแต่ละกลุ่มไว้

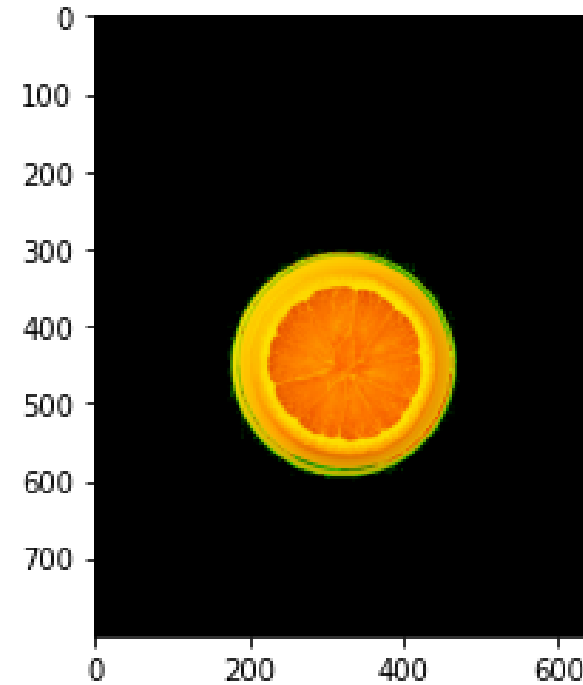
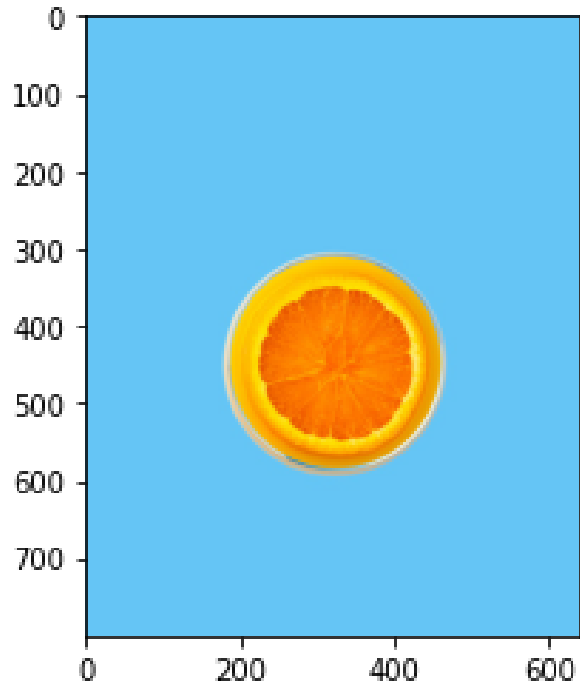
## Step:

1. เข้าใจโจทย์ ดูข้อมูลที่ได้รับมา วางแผน ออกแผนผัง/ขั้นตอนของการแก้โจทย์ (15 min.)
2. นำเข้าข้อมูลและแก้ไขข้อมูลที่ผิดพลาด (30 min.)
3. ทำตารางผลลัพธ์และแสดงผลออกมา โดยข้อมูลที่ไม่ต้องเป็นค่าว่าง ‘ ’ (25 min.)

## Extra:

1. Plot กราฟผู้เข้าอบรมในแต่ละกลุ่ม
2. เมื่อได้ตารางดังกล่าวแล้วให้ export เป็นไฟล์ excel (ทดลองหาคำสั่ง excel และแก้ไขปัญหาเรื่องภาษาเอเลียนด้วยตนเองผ่าน internet)

Extra : A orange should be orange. (before afternoon session) 



ให้ลบ background ของรูปด้านซ้ายออก

Hint: - พื้นหลังจะดำก็ต่อเมื่อ r,g,b เป็น 0 ทั้งหมด  
- สีส้มเกิดจากส่วนผสมของสีแดงและสีเขียว



# GBDi

Government Big Data Institute

สถาบันส่งเสริมการวิเคราะห์และบริหารข้อมูลขนาดใหญ่ภาครัฐ (สวช.)

Follow us on



# GBDi

gbdi.depa.or.th

Facebook



Twitter



govbigdata

Blockdit



YouTube

Government Big Data Institute  
(GBDi)



Line  
Official

@gbdi