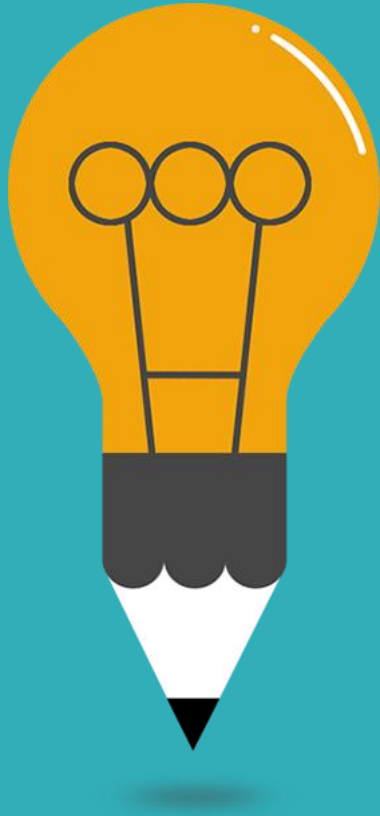
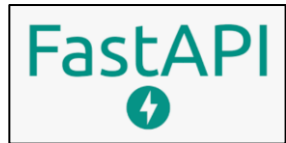


# การเขียนโปรแกรมภาษาไพธอน (Python) เบื้องต้น





# Python and Google's colab



จัดการข้อมูลที่เป็นตาราง  
(เช่น ไฟล์ excel)  
กรองข้อมูล (data filtering)

สร้างเว็บไซต์



การประมวลผลภาพและวิดีโอ  
(Image and Video Processing)

Deep Learning



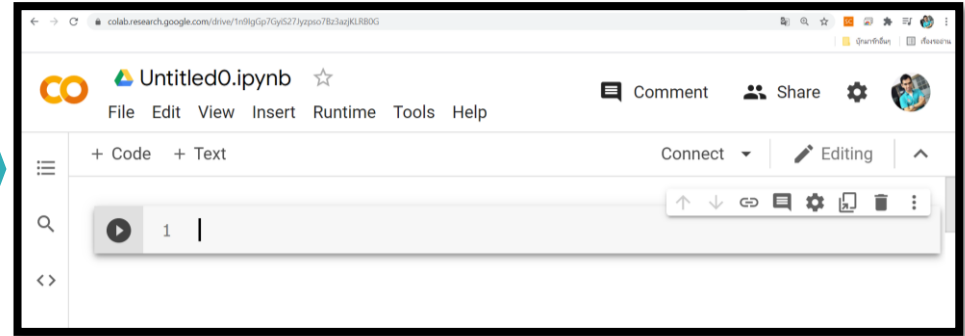
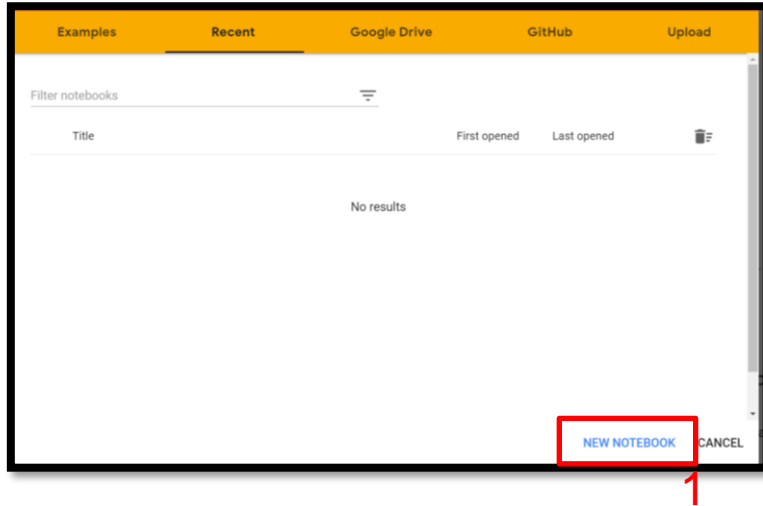
การคำนวณทาง  
คณิตศาสตร์, matrix



Machine Learning

# Google's colab

<https://colab.research.google.com> (ต้อง login Google Account)



The image shows the Google Colab web interface. At the top, the file name is 'Untitled.ipynb' with a star icon. Below it is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. To the right are buttons for 'Comment', 'Share', and a settings icon, along with a user profile picture. Below the menu bar, there are buttons for '+ Code' and '+ Test'. On the right side of this bar, there are indicators for 'RAM' and 'Disk' usage, and a status 'Editing'. The main area contains a code cell with a play button icon on the left. The code inside the cell is:

```
x=5  
y=3  
print(x+y)
```

Below the code cell, there is an output box showing the number '8'. The interface is annotated with several red callouts in Thai: 'ชื่อไฟล์' (File name) points to the 'Untitled.ipynb' text; 'เพิ่ม cell' (Add cell) points to the '+ Code' button; 'ปุ่ม Run' (Run button) points to the play button icon; 'พิมพ์ code' (Type code) points to the code text; 'output' points to the output box; and 'cell' points to the code cell itself.

CO Untitled.ipynb ☆ ชื่อไฟล์

File Edit View Insert Runtime Tools Help

+ Code + Test เพิ่ม cell

RAM Disk Editing

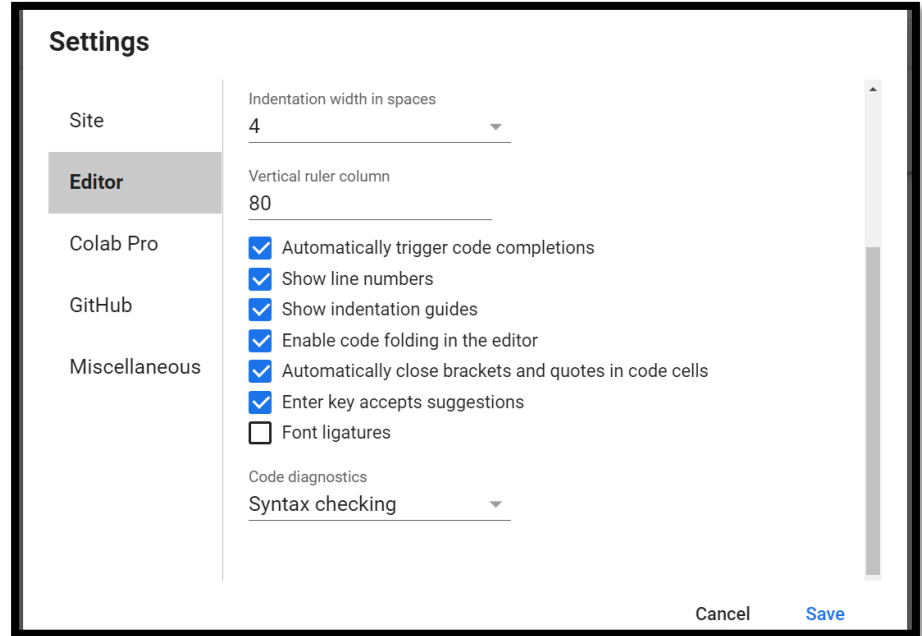
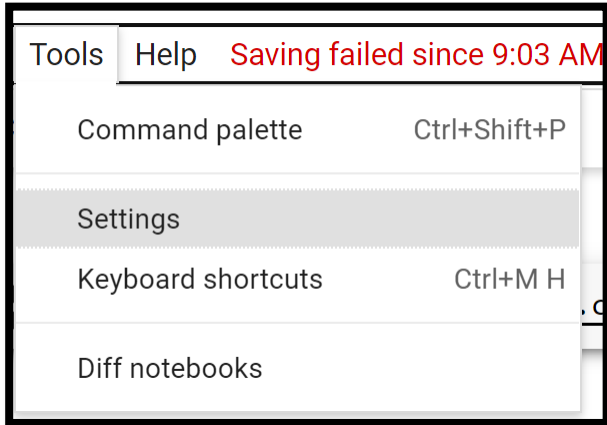
ปุ่ม Run

พิมพ์ code

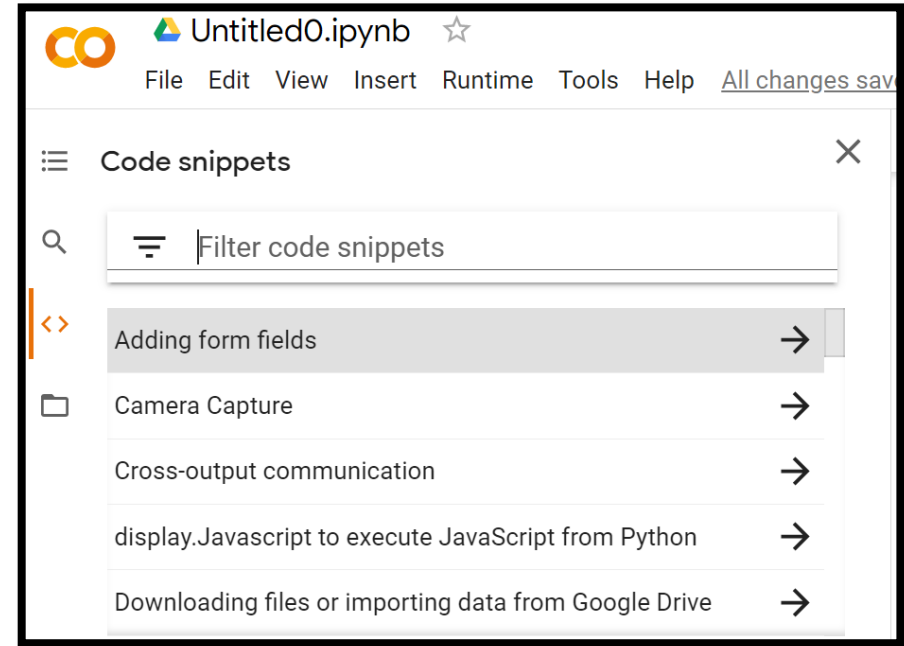
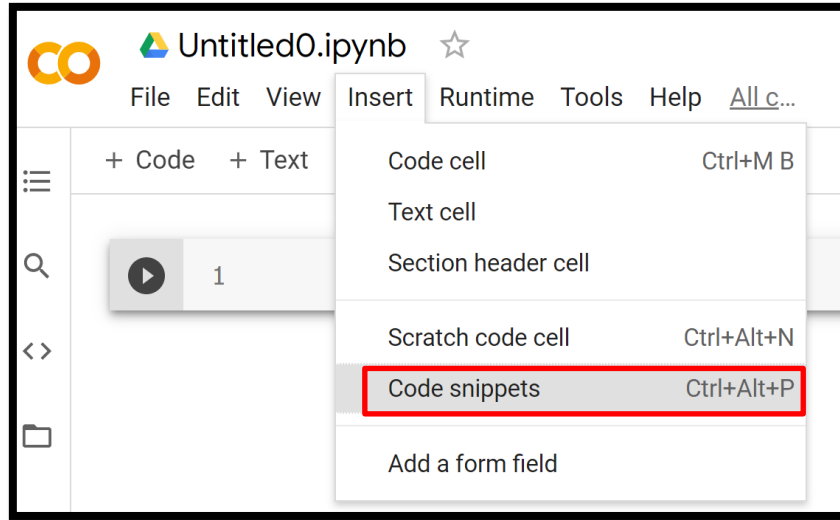
output

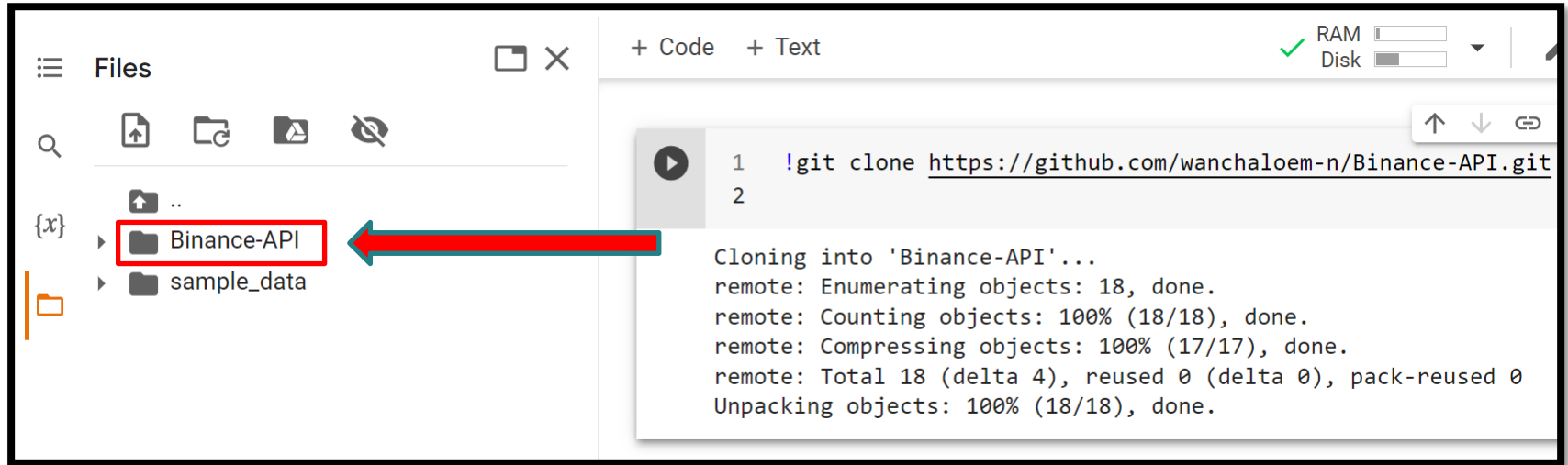
cell

# Setting



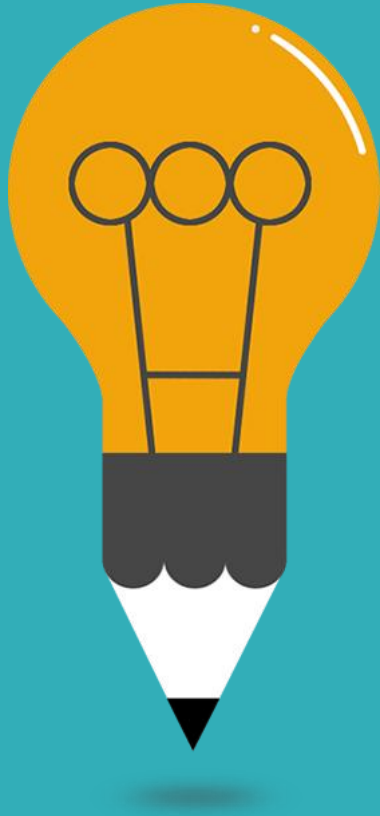
# ตัวอย่าง code ใน Google's colab





```
!git clone https://github.com/wanchaloem-n/Binance-API.git
```





# Python - Basic Syntax

# ตัวแปร (Variables)

```
a = 3.21  
A = 1.23  
num_1 = 5
```

```
# python code  
a=3.21  
A=1.23  
Num=5  
print(a)  
print(A)  
print(a+A)
```

## Reserved keywords

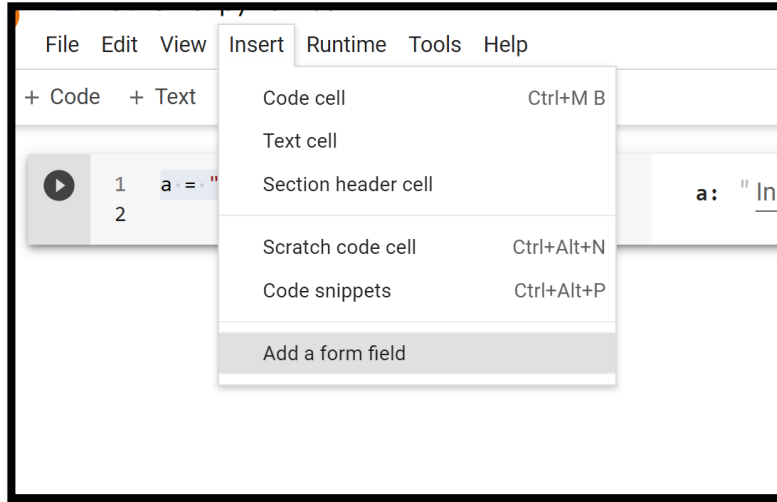
False	None	True	and
as	assert	break	class
continue	def	del	elif
else	except	finally	for
from	global	if	import
in	is	lambda	nonlocal
not	or	pass	raise
return	try	while	with
yield			

# Variable Types

Type	Example
integer	age = 12
float	height = 103.5
complex	C = 5+2j
string	gender = "male"
boolean	single = False

```
# python code  
age=12  
height=103.5  
C=5+2j  
gender='male'  
single=True  
single=False  
print(age,type(age))
```

# Form Field



A screenshot of a Jupyter Notebook cell. The code cell contains the following code:

```
1 # =====
2 in1 = "test" #@param {type:"string"}
3 in2 = "b" #@param ["a", "b", "c"]
4 in3 = 14 #@param {type:"slider", min:0, max:100,
5 print(in1,in2,in3)
```

The output of the code is 'test b 14'. To the right of the code cell, there are three form fields corresponding to the parameters in the code:

- in1:** A text input field containing 'test'.
- in2:** A dropdown menu with 'b' selected.
- in3:** A slider control with a blue dot at 14.

# ตัวดำเนินการพื้นฐานทางคณิตศาสตร์

$+$ , $-$ , $*$ , $/$	บวก ลบ คูณ หาร
$a ** b$ หรือ $\text{pow}(a, b)$	$a^b$
$\text{pow}(a, b, m)$	$a^b \bmod m$
$//$	หารตัดเศษ
$\%$	เศษจากการหาร

# การเปรียบเทียบ

==	เท่ากับ
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ
!=	ไม่เท่ากับ

# สัญลักษณ์ทางตรรกศาสตร์

Logical operation	Bitwise operation
and	&
or	
not	~

# 0.1 + 0.2 = \_\_\_\_?

```
1  0.1+0.2
2
```

```
0.30000000000000004
```

```
1  (0.1+0.2)==0.3
```

```
False
```

$10^3$	$10^2$	$10^1$	$10^0$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$		
2	2	0	.	5					
(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)	(0.5)	(0.25)
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$
1	1	0	1	1	1	0	0.	1	0

$$128 + 64 + 16 + 8 + 4 + 0.5 = 220.5$$

$$0.1_{10} = 0.0001100_2$$

$$0.2_{10} = 0.001100_2$$

Ref: <https://stackpython.medium.com/0-1-0-2-%E0%B8%88%E0%B8%B6%E0%B8%87%E0%B9%84%E0%B8%A1%E0%B9%88%E0%B9%80%E0%B8%97%E0%B9%88%E0%B8%B2%E0%B8%81%E0%B8%B1%E0%B8%9A-0-3-%E0%B9%81%E0%B8%A5%E0%B8%B0-ieee-754-4908787e1d1c>



# การลบตัวแปร

del var1, var2, var3

# การติดตั้ง packages

**!pip install** (ชื่อ package)

```
# python code  
!pip install pythainlp
```

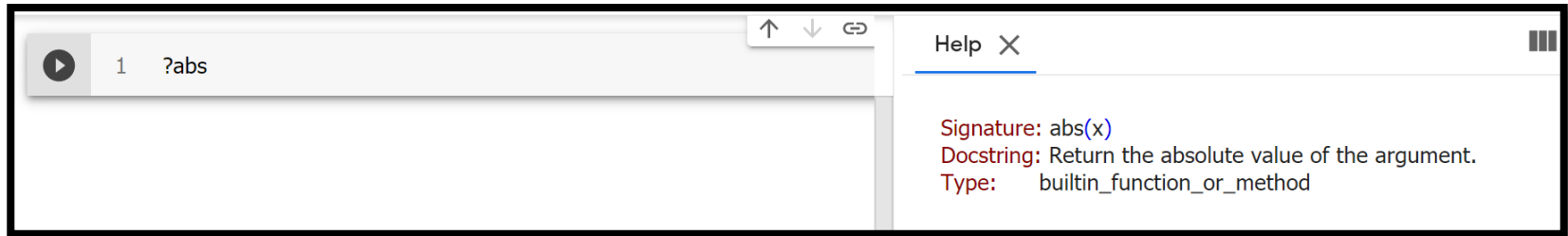
# การ import packages

```
# python code  
import math  
from math import *  
from math import sqrt  
import math.sqrt as sqt
```

# การใช้ help และ dir ใน python

## ?function name

ตัวอย่าง



```
# python code  
print(abs(-5))  
?abs
```

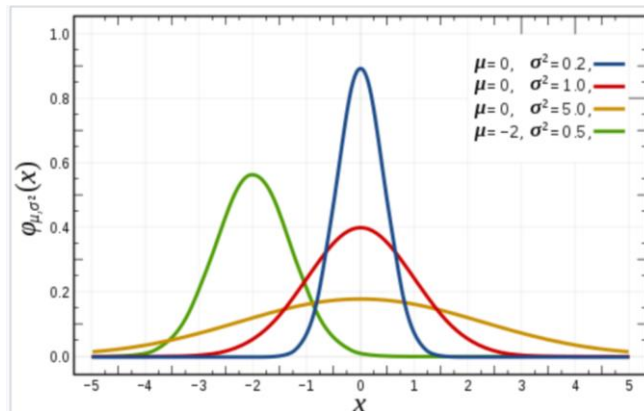
```
# python code  
dir(math)
```

# การใช้ฟังก์ชันทางคณิตศาสตร์

abs(a)	$ a $
math.sin(a)	$\sin(a)$
math.floor(a)	$\lfloor a \rfloor$
math.ceil(a)	$\lceil a \rceil$
round(a)	$[a]$
math.exp(a)	$e^a$
math.e	$e$
math.pi	$\pi$
math.sqrt(a)	$\sqrt{a}$
math.log(a)	$\ln(a)$
math.log(a,b)	$\log_b(a)$

# แบบฝึกหัด จงเขียน code เพื่อหาค่า Y

$$Y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



```
# python code
x=1
mu=0.5
sigma=1.5
...
y=_____?
```

# List, Tuple, Set, และ Dictionary

## List

```
X = ["red" , "blue" , "red"]
```

```
Y= [1,3,2,5]
```

code	output
Z=X+Y	["red" , "blue" , "red",1,3,2,5]
Y.append(1)	[1,3,2,5,1]
len(X)	3
X.insert(2,'a')	["red" , "blue" , "a","red"]
Y.sort()	[1,2,3,5]
Y.reverse()	[5,2,3,1]

# List

Index of elements

```
X = [1, 'red', 'blue', 'red', 'r', 'b']
```

code	output
X[2]	'blue'
X[-2]	'r'
X[1:4]	['red', 'blue', 'red']
X[4:]	['r', 'b']
X[:3]	[1, 'red', 'blue']



# Nested List

```
X = [ 1,[2,3],[4,5,6],[[7,8]]]
```

code	output
X[1]	[2,3]
X[1][0]	2
X[3][0][1]	8

# Set

`X = {"red" , "blue" , "red"}`

`Y= {"red" , "green"}`

code	output
<code>X == {"red" , "blue"}</code>	True
<code>"red" in X</code>	True
<code>len(X)</code>	3
<code>X.union(Y)</code>	<code>{"red" , "green", "blue"}</code>
<code>X.intersection(Y)</code>	<code>{"red"}</code>
<code>X-Y</code>	<code>{"blue"}</code>

# Tuple

X = (30, 20000, "male")

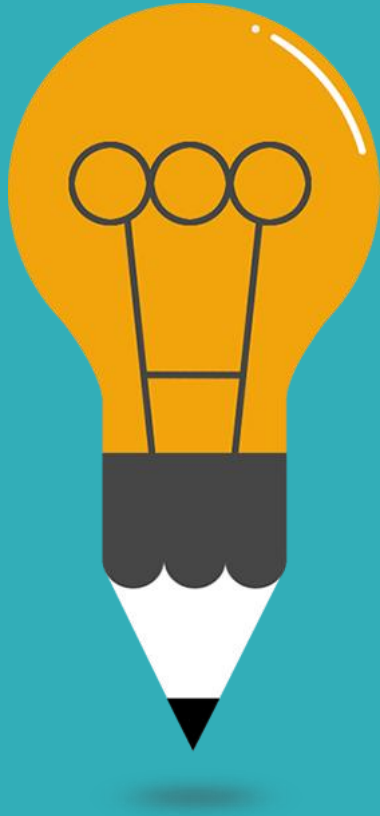
Y = (25, 16000, "male")

code	output
Z=X+Y	(30, 20000, 'male', 25, 16000, 'male')
Z=(X,Y)	((30, 20000, 'male'), (25, 16000, 'male'))
Z=X*2	(30, 20000, 'male', 30, 20000, 'male')
len(X)	3
tuple('male')	('m', 'a', 'l', 'e')

# Dictionary

```
X = {'Name': 'A', 'Income': 12000, 'Gender': 'Male'}
```

code	output
<code>X["name"]</code>	'A'
<code>X.keys()</code>	{'Name', 'Income', 'Gender'}

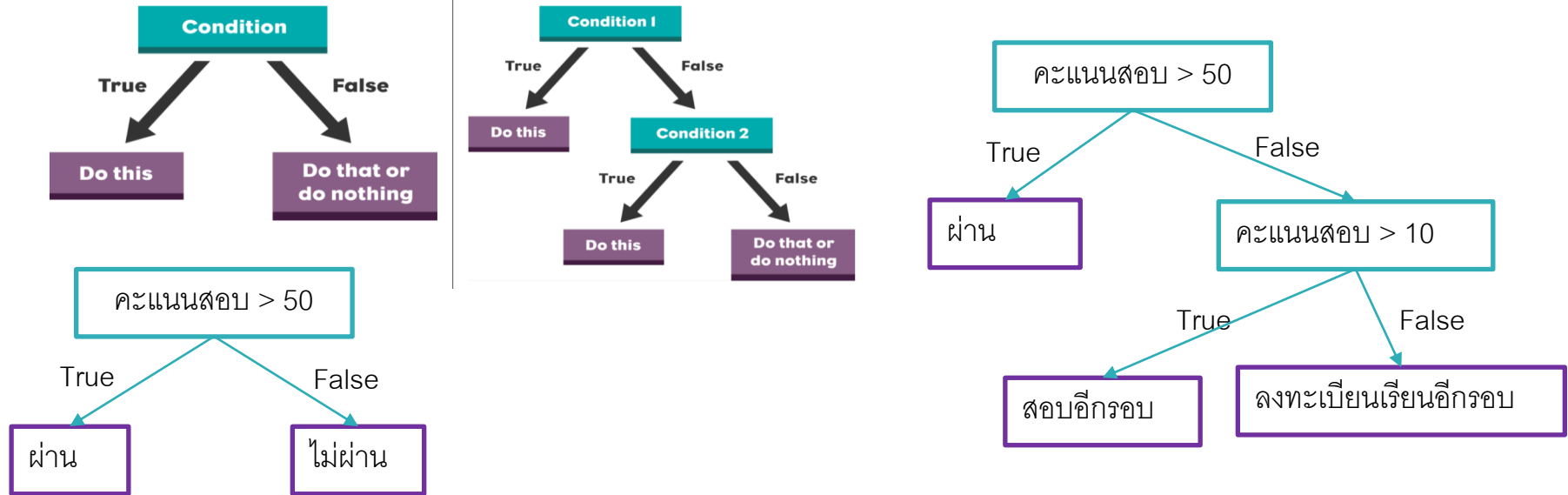


# Condition and Loop

if, else, for, while

# Condition and loop

## Condition (if ... else ...)



```
# python code
score=47
if score>50:
    print("ผ่าน")
else:
    print("ไม่ผ่าน")
```

```
# python code
score=2
if score>50:
    print("ผ่าน")
elif score>10:
    print("สอบอีกรอบ")
else:
    print("ไม่ผ่าน")
```

# Condition and loop

## Loop (while loop)

**while condition:**  
**do something**

**while True:**  
**do something**  
**if condition:**  
**break**

```
# python code
scores=[2,47,0,52]
i=0
while scores[i]>0:
    print(scores[i])
    i=i+1
```

```
# python code
scores=[2,47,0,52]
i=0
while True:
    print(scores[i])
    i=i+1
    if scores[i]<=0:
        break
```



# Condition and loop

## Loop (For loop)

```
for i in [1,2,4]:  
    do something
```

```
for i in range(2,5):  
    do something
```

```
# python code  
for score in [2,47,50,52]:  
    if score>50:  
        print("ผ่าน")  
    else:  
        print("ไม่ผ่าน")
```

```
# python code  
scores=[2,47,50,52]  
for i in range(len(scores)):  
    if scores[i]>50:  
        print(i,"ผ่าน")  
    else:  
        print(i,"ไม่ผ่าน")
```

แบบฝึกหัด

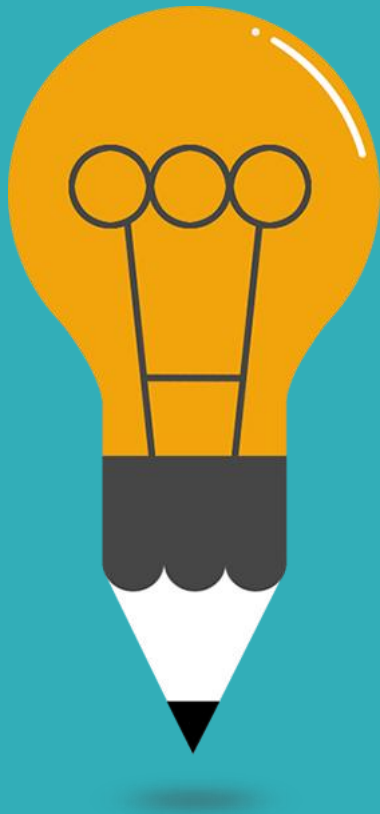
จงเขียน code เพื่อหาตัวประกอบเฉพาะทั้งหมด

ของ  $N1, N2$

เมื่อกำหนด

$N1=2475115831,$

$N2=166153499473114472511703242428645381$



# การสร้าง custom function

# custom function

คณิตศาสตร์

$$z = f(x, y)$$

python

```
def function_name(x, y):  
    do something  
    return z
```

```
function_name = lambda x, y : z
```

$$f(x) = 2x^2 + 1$$

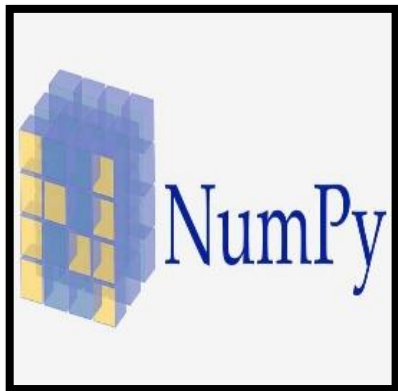
```
# python code
def f1(x):
    return 2*x**2+1
def f2(x):
    y=2*x**2+1
    return y
f3 = lambda x: 2*x**2+1
print(f1(2))
print(f2(2))
print(f3(2))
```

$$\text{gaussian}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

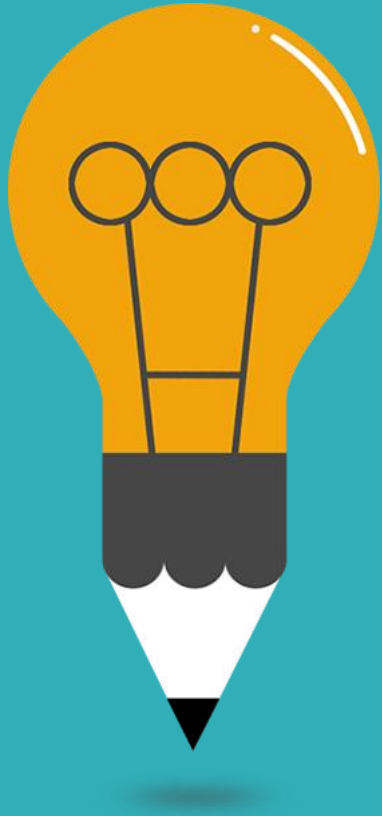
```
# python code
import math
def gaussian(x,mu,sigma):
    coef=1/(math.sqrt(2*math.pi*(sigma**2)))
    expo=-((x-mu)**2)/(2*sigma**2)
    return coef*math.exp(expo)
print(gaussian(1,0,1))
```

# Numpy array

```
import numpy as np
```



code	
<code>V=np.array([1,2,3,5])</code>	สร้าง vector
<code>M=np.array([[1,2],[3,5]])</code>	สร้าง matrix



Numpy array

# ตัวดำเนินการพื้นฐาน

```
M1=np.array([[1,2],[3,5]])  
M2=np.array([[5,-4],[3,5]])
```

M1*M2	การคูณทีละสมาชิก
np.dot(M1,M2)	การคูณเมทริกซ์
M1+M2	
M1-M2	
np.log(M1)	
np.diag([1,2,3])	สร้าง diagonal matrix
np.diag(M1)	diagonal matrix ของ M1
np.eye(4)	
np.random.rand(4,5)	random matrix
np.zeros((2,3))	
np.ones((2,3))	
M1.transpose()	
M1.shape	dimension ของ M1



# Numpy.linalg

from numpy import linalg as LA

## Matrix eigenvalues

<code>linalg.eig(a)</code>	Compute the eigenvalues and right eigenvectors of a square array.
<code>linalg.eigh(a[, UPLO])</code>	Return the eigenvalues and eigenvectors of a complex Hermitian (conjugate symmetric) or a real symmetric matrix.
<code>linalg.eigvals(a)</code>	Compute the eigenvalues of a general matrix.
<code>linalg.eigvalsh(a[, UPLO])</code>	Compute the eigenvalues of a complex Hermitian or real symmetric matrix.

## Norms and other numbers

<code>linalg.norm(x[, ord, axis, keepdims])</code>	Matrix or vector norm.
<code>linalg.cond(x[, p])</code>	Compute the condition number of a matrix.
<code>linalg.det(a)</code>	Compute the determinant of an array.
<code>linalg.matrix_rank(M[, tol, hermitian])</code>	Return matrix rank of array using SVD method
<code>linalg.slogdet(a)</code>	Compute the sign and (natural) logarithm of the

## Solving equations and inverting matrices

<code>linalg.solve(a, b)</code>	Solve a linear matrix equation, or system of linear scalar equations.
<code>linalg.tensorsolve(a, b[, axes])</code>	Solve the tensor equation $a \cdot x = b$ for $x$ .
<code>linalg.lstsq(a, b[, rcond])</code>	Return the least-squares solution to a linear matrix equation.
<code>linalg.inv(a)</code>	Compute the (multiplicative) inverse of a matrix.
<code>linalg.pinv(a[, rcond, hermitian])</code>	Compute the (Moore-Penrose) pseudo-inverse of a matrix.
<code>linalg.tensorinv(a[, ind])</code>	Compute the 'inverse' of an N-dimensional array.

## Exceptions

<code>linalg.LinAlgError</code>	Generic Python-exception-derived object raised by linalg functions.
---------------------------------	---

<https://numpy.org/doc/stable/reference/routines.linalg.html>

# Example code

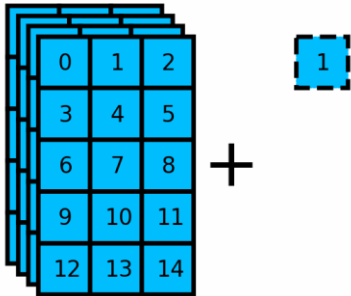
```
# python code
import numpy as np
import numpy.linalg as LA

V=np.array([1,2])
print("2-norm: ", LA.norm(V,ord=2))

M=np.array([[1,2],[3,4]])
print("inv: ", LA.inv(M))
print("eig values & vectors: ", LA.eig(M))
print("det: ", LA.det(M))
```

# Numpy array: broadcasting

$$\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} + [0 \quad 1 \quad 2 \quad 3] = \_\_\_?$$



© Matt Eiding



\*



© Matt Eiding

# แบบฝึกหัด จงเขียน python code เพื่อหา $A^+$

Moore–Penrose inverse (ref: [https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose\\_inverse](https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose_inverse))

$$AX = B \Rightarrow X \approx A^+ B$$

$$A^+ = (A^T A)^{-1} A^T$$

```
# python code
import numpy as np
A=np.array([[1,2],[3,4],[5,6]])
B=np.array([[1],[2],[3]])
...
X=___?
```

# Slicing and Filtering

```
# python code
import numpy as np
M1=np.array([[1,2],[3,4],[5,6],[7,8]])
M2=np.array([[1,2,3,4]])

print("1st: ",M1)
print("2nd: ",M1[1:,:])
print("3rd: ",M1[1:3,:])
print("4th: ",M1[:3,:])
print("5th: ",M1[2,1])
print("6th: ",M1[[1,3],:])
print("7th: ",M1[0:4:2,:])
```

```
# python code
import numpy as np
M1=np.array([[1,2],[3,4],[5,6],
],[7,8]])
M2=np.array([[1,2,3,4]])
M3=np.array([[1,2,3],[3,4,5],
[5,6,7],[7,8,9]])

print("1st: ",M1[M1[:,1]>5])
print("2nd: ",M2[M2>=3])
print("3rd: ",M3[M1[:,1]>5])
```

# Scipy stats

T-test, F-test,  
p-value



# PANDAS

ใช้สำหรับจัดการข้อมูลในตาราง

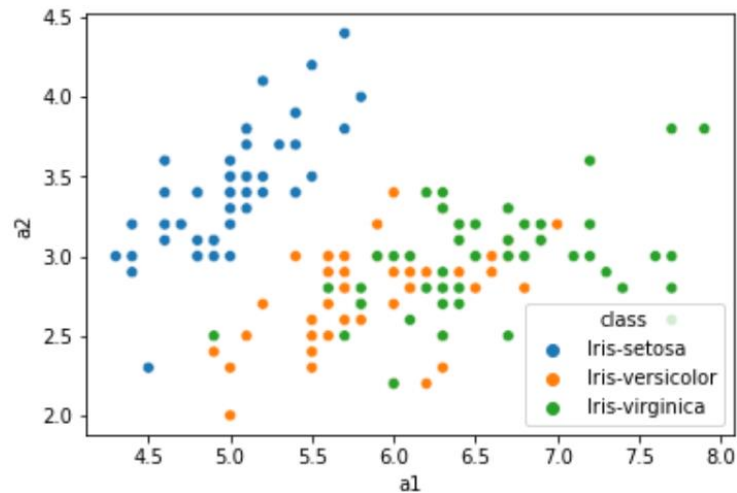
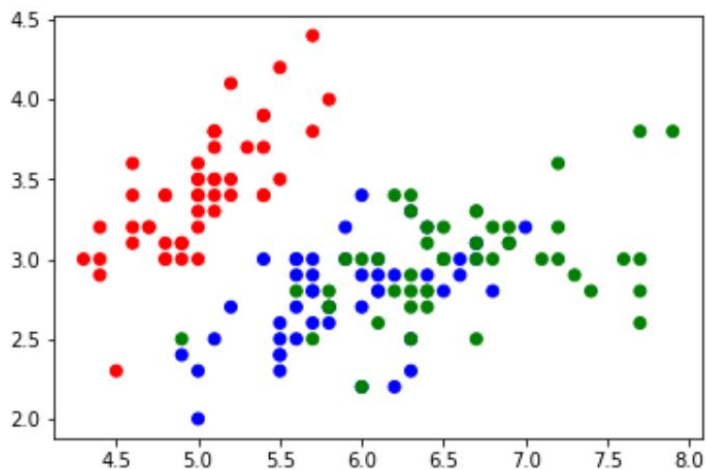
```
import pandas as pd
```

```
import pandas as pd
df=pd.DataFrame({
    "ID": [6, 7, 8, 9],
    "AGE": [14, 16, 18, 20],
    "GENDER": ["M", "F", "F", "M"]
})
```

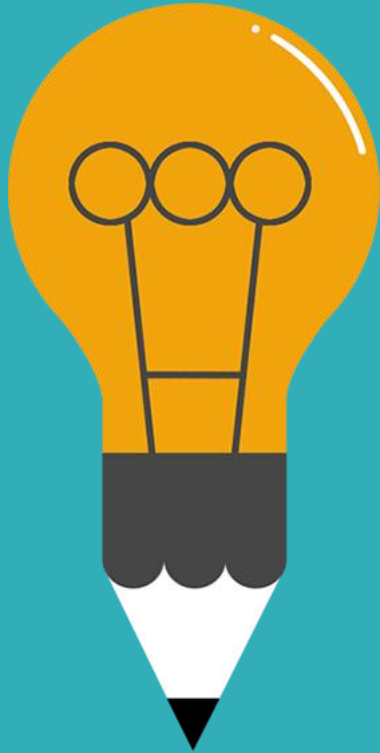
	ID	AGE	GENDER
0	6	14	M
1	7	16	F
2	8	18	F
3	9	20	M

# Matplotlib and Seaborn

การพอร์ตกราฟ



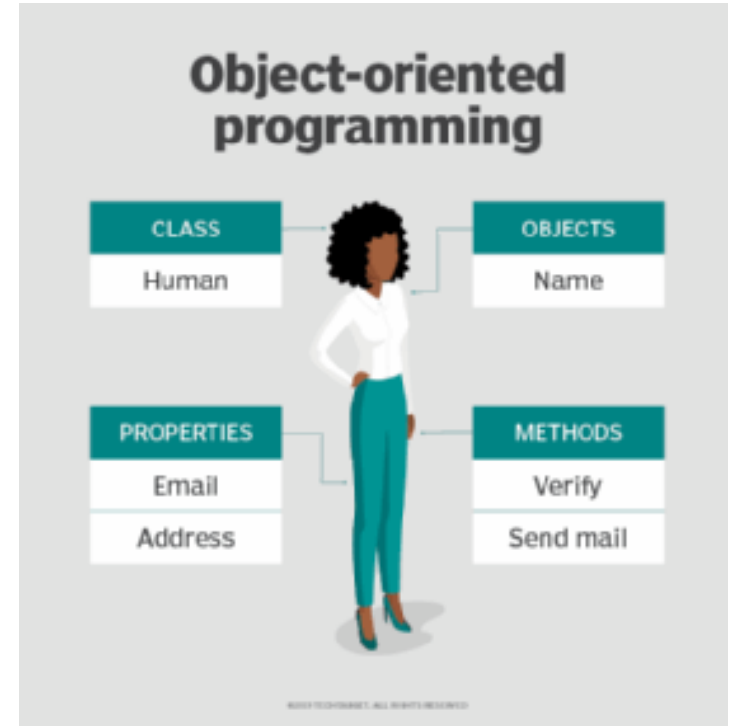
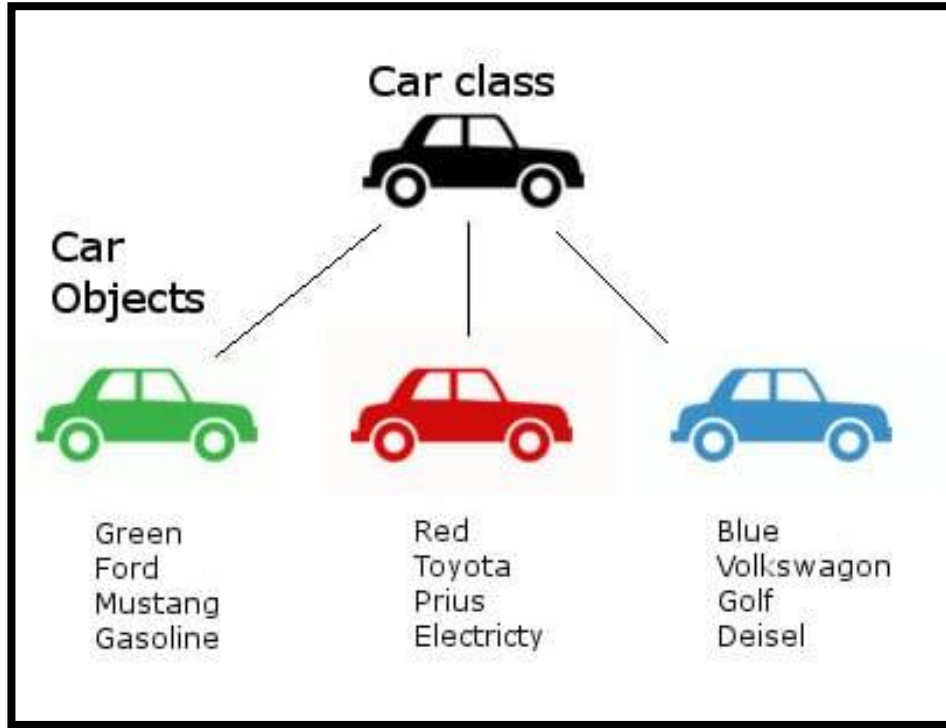




# OOP

## Object Oriented Programming

# Object Oriented Programming (OOP)



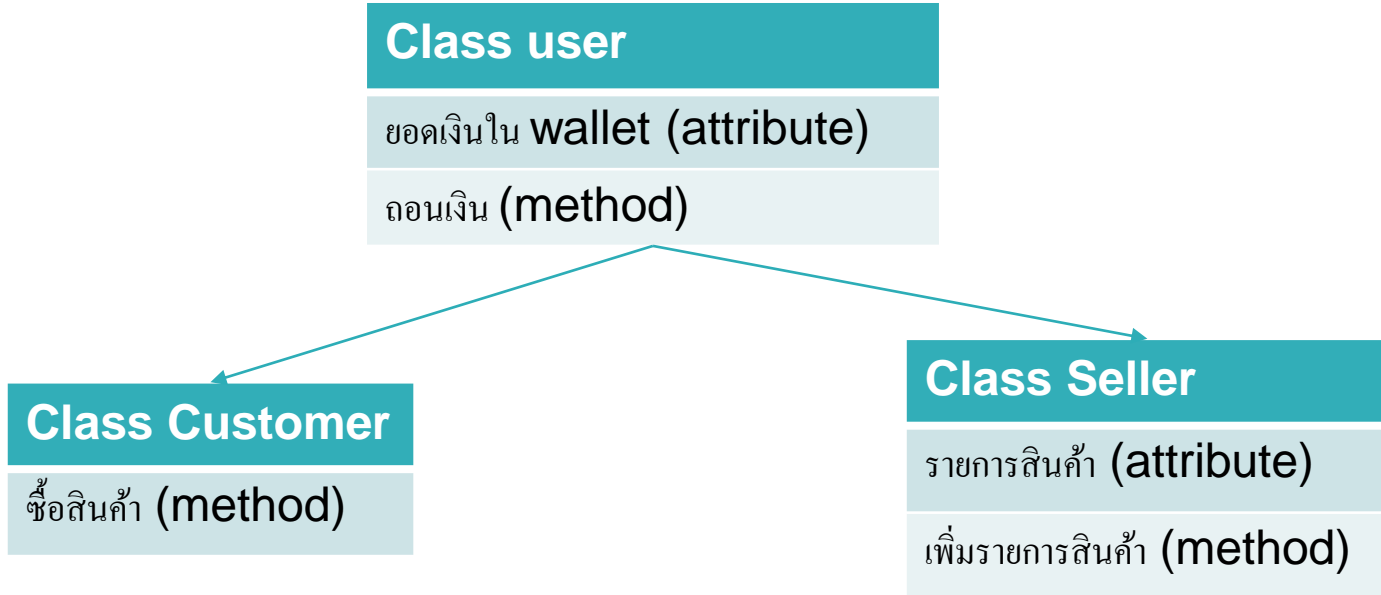
# โครงสร้าง class

```
class ClassName:
    # Attributes
    var1="some values"

    def __init__(self, var1):
        self.var1 = var1

    # Methods
    def function1(self, val2):
        self.var1 = val2
Obj1 = ClassName(val1)
```

# Inheritance Class



```
# python code
class User:
    def __init__(self,wallet):
        self.wallet=wallet
    def withdraw(self,amount):
        self.wallet=self.wallet-amount

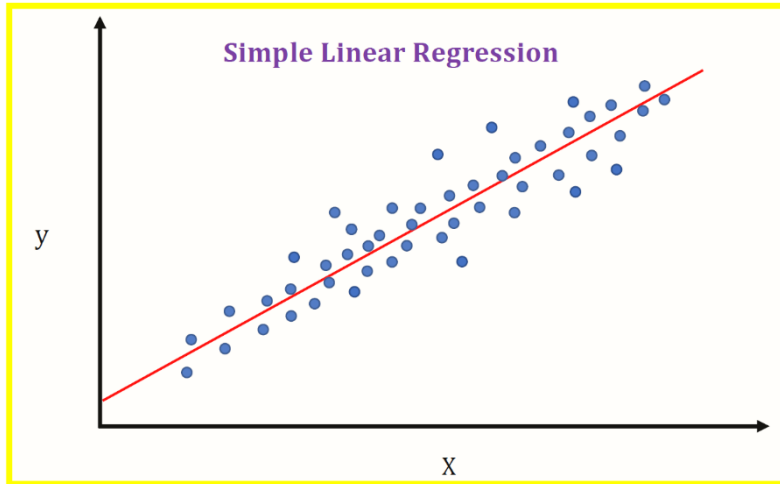
class Customer(User):
    def __init__(self,wallet):
        super().__init__(wallet)
        self.cart=[]
    def buy(self,product,price):
        self.cart.append(product)
        self.wallet=self.wallet-price
```

```
U1=User(500)
print(U1.wallet)
U1.withdraw(30)
print(U1.wallet)
C1=Customer(5000)
print(C1.wallet)
C1.buy("cat",10)
C1.buy("dog",10)
print(C1.cart)
print(C1.wallet)
print("\n==== withdraw====")
C1.withdraw(100)
print(C1.cart)
print(C1.wallet)
```

# Simple Linear Regression

ref: [https://home.kku.ac.th/nikom/regsimp\\_nk2559.pdf](https://home.kku.ac.th/nikom/regsimp_nk2559.pdf)

$$\hat{y}_i = a + bx_i$$



$$a = \bar{y} - b\bar{x}$$

$$b = \left[ \frac{\sum x_i y_i - \frac{(\sum x_i)(\sum y_i)}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}} \right]$$



Thank you