```
x = 100
y = 200
print(x+y)
```

```
300
```

```
print("Hello World")
```

```
Hello World
```

Python Programming 101

1. Variable
2. Data Type
3. Data structure
4. control flow
5. function

```
# calculation
1+1
```

```
2
```

```
print(5*2)
print(2/4)
print(2+2)
print(3-1)
print(2**2) # Power
```

```
10
0.5
4
2
4
```

```
10/3
```

```
3.3333333333333335
```

```
# modulo (mod)
10 % 3
```

```
1
```

```
10 % 4
```

```
2
```

```
# variables
x = 100
y = 200
print(x+y)
```

```
300
```

```
# naming variable -> sanke case
# data types int , float , string ,boolean
my_name = "louis" # string
my_age = 25  # int
be_gpa = 3.23 #float
netflix = True ## Flase / boolean
```

```
print(my_age,my_name, be_gpa)
```

```
25 louis 3.23
```

```
print("Hello!", "my name is ", "louis")
```

```
Hello! my name is  louis
```

```
## creat_multiple variables
## tuple unpacking
x, y, z = 100, 200 , 330
print(x, y, z)
```

```
100 200 330
```

```
## replace value / รันจากบนลงล่างดูตัวแปรดีๆ
my_name = "toy"
my_name = "Louis"
print(my_name)
```

```
Louis
```

```
## remove variable
del my_name
```

```
## data types conversion
## int(), float(), str(), bool()
bool(0)
```

```
False
```

```
str(100)
```

```
'100'
```

```
int("555")
```

```
555
```

```
float("3.41")
```

```
3.41
```

```
## homework 01 : pao ying chub
## get input from user Pao ying chub
```

```
age = int(input("What's your age :"))
```

```
What's your age :25
```

```
print(age, type(age))
```

```
25 <class 'int'>
```

```
"I love" + "Python!"
```

```
'I lovePython!'
```

```
## fstirng template
my_name = "Louis"
my_age = 25
text = f"I love Python! My age is {age} and I am {my_name}"
print(text)
```

```
I love Python! My age is 25 and I am Louis
```

```
## function
def  double(x):
  return x*2
```

```
double(100)
```

```
200
```

```
# default argument
def greeting(name , food = "ramen") : # สามารถกำหนดตัวแปรข้างในได้
  text = f" {name} likes to eat {food}"
  print("hi! ") # เมื่อก่อนใช้ debug code
  return text
```

```
result = greeting("louis") # หากไม่ได้กำหนดตัวแปรจะดึง Default ที่กำหนดไว้แทน
```

```
hi!
```

```
result
```

```
' louis likes to eat ramen'
```

```
## return more than one input
def greeting2(x) :
  return (x**2, x+2 , x + 5)
```

```
x, y, _ = greeting2(5) # x +  5 ไม่ได้อยากใช้ ใช้ _ รับค่าแทน
print(x, y)
```

```
25 7
```

```
## modularity (modular programming) ## เป็น Concept ของการเขียน code เป็นขั้นเป็นตอน
def f1():
  print("hi")

def f2():
  print("Hello")

def f3():
  print("Ni hao")

def f4():
  f1()  # load data
  f2() # clean data
  f3() # prep data
  print("done!!!")
```

```
f4()
```

```
hi
Hello
Ni hao
done!!!
```

```
## contorl flow
## if-else, for ,while
```

```
def grade(score):
  if score >= 80:
    return "Pass"
  else:
    return "Please retake the exam"
```

```
grade(70)
```

```
'Please retake the exam'
```

```
## f elif else
def grade_adv(score):
  if score >= 90:
    return "A"
  elif score >= 80:
    return "B"
  elif score >= 50:
    return "C"
  else:
    return "Please retake the exam!"
```

```
grade_adv(49)
```

```
'Please retake the exam!'
```

```
## multiple conditions
## and, or (reserve key word)
def testing():
  if (1+1 == 2 and 2*2 == 4):
    print("correct")
  else:
    print("incorrect")
```

```
testing()
```

```
correct
```

```
## Python basic data structures
## list , tuple , dict , set
```

```
## 1 . list (similar to vector in R)
my_shopping = ["egg", "milk","bread"]

## index starts at zero
print(my_shopping[0])
print(my_shopping[1])
print(my_shopping[2])

## check number of items
print(len(my_shopping))
```

```
egg
milk
bread
3
```

```
## update value in list
## mutable data structure สามารถ Update ค่าข้างในได้ (can be update)
my_shopping[0] = 'orange'
my_shopping
```

```
['orange', 'milk', 'bread']
```

```
type(my_shopping)
```

```
list
```

```
## list methods
## method is a function designed for a specific data structures / types
my_shopping.append("honey") ## .append คือ method

my_shopping
```

```
['orange', 'milk', 'bread', 'honey']
```

```
my_shopping.append("Coke")
len(my_shopping) ## generic function จะอยู่ข้างหน้า
my_shopping
```

```
['orange', 'milk', 'bread', 'honey', 'Coke']
```

```
my_shopping.pop() # ลบขวาสุด
my_shopping
```

```
['orange', 'milk', 'bread', 'honey']
```

```
## remove item
my_shopping.remove('milk')
```

```
my_shopping
```

```
['orange', 'bread', 'honey']
```

```
## insert item
my_shopping.insert(2,'rice')
my_shopping
```

```
['orange', 'bread', 'rice', 'honey']
```

```
## sort data
my_shopping.sort(reverse= True) ## from z to a (Desc)
```

```
my_shopping
```

```
['rice', 'orange', 'honey', 'bread']
```

```
## combine two lists
full_list =['egg', 'chicken'] +  ['milk', 'cereal']
full_list
```

```
['egg', 'chicken', 'milk', 'cereal']
```

```
fruits = ['banana', 'orange', 'starwberry']

for fruit in fruits:
  if fruit == "banana":
    print("monkey")
  else:
    print(fruit.title())
```

```
monkey
Orange
Starwberry
```

```
## mutable vs. immutable
## string is immutable
text = "a duck walks into a bar"
text.replace("duck","lion")
```

```
'a lion walks into a bar'
```

```
text
```

```
'a duck walks into a bar'
```

```
text =text.replace("duck","lion")
text
```

```
'a lion walks into a bar'
```

```
## or Immutable
my_name = "Vachiravit"
"W" + my_name[1: ]
```

```
'Wachiravit'
```

```
## reference

x = [1,2 ,3]
y = x
z = x

x[0] = 5
print(x, y, z)
```

```
[5, 2, 3] [5, 2, 3] [5, 2, 3]
```

```
## if- else
## for loop

for i in range(10): ## 1:10
  print(i+1 , "hello")
```

```
1 hello
2 hello
3 hello
4 hello
5 hello
6 hello
7 hello
8 hello
9 hello
10 hello
```

```
list(range(5))
```

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

```
## while loop
## careful infinite loop

count = 0
```

```
while (count < 5):
  print("hi")
  count +=1
```

```
hi
hi
hi
hi
hi
```

```
play = True

while play :
  user_input = input("What do you want ? ")
  print(user_input)
  if user_input == "Nothing":
     print("Bye!")
     play = False
```

```
What do you want ? Wa
Wa
What do you want ? Wa
Wa
What do you want ? Nothing
Nothing
Bye!
```

```
## data structure
## list , tuple , dict , set

laptops = ['dell , lenovo', 'apple']

result = [ ] ## empty list

for laptop in laptops:
  tmp = laptop.upper()
  result.append(tmp)

print(result)
```

```
['DELL , LENOVO', 'APPLE']
```

```
## list comprehension
laptops = ['dell , lenovo', 'apple']
laptops_upper = [laptop.upper() for laptop in laptops]

print(laptops_upper)
```

```
['DELL , LENOVO', 'APPLE']
```

```
## tuple is IMMUTABLE
x = (1,2,3)
print(x, type(x))
```

```
(1, 2, 3) <class 'tuple'>
```

```
## tuple and list
## multiple data types

["louis", 25, ["R", "Python","SQL"], ("Economics","Engineer"), True]
```

```
['louis', 25, ['R', 'Python', 'SQL'], ('Economics', 'Engineer'), True]
```

```
## set (no duplicates) Unique value
food = {"orange",'orange','lemon','lemon','ramen'}
```

```
food
```

```
{'lemon', 'orange', 'ramen'}
```

```
## set operation
a = {"orange", "banana"}
b = {"orange", "grape","pineapple"}
```

```
## intersect (inner join)
a & b
```

```
{'orange'}
```

```
## union
a | b
```

```
{'banana', 'grape', 'orange', 'pineapple'}
```

```
## difference
a - b
```

```
{'banana'}
```

```
## dictionary
## key-value pair (JSON)
user = {
    "name" : "louis",
    "age" : 25,
    "location" : "BKK",
    "fav_movies" : ['Superman',"Thunderbolt","Avenger"],
    "steaming" : {"netflix" : True,
                  "amazon" : False},
}
```

```
user["age"] = 26
user["name"] = "Por"
```

```
# หากไม่มี key อยู่จะเป็นการเพิ่มเข้าไป
user["dog name"] = "Box"
user
```

```
{'name': 'Por',
 'age': 26,
 'location': 'BKK',
 'fav_movies': ['Superman', 'Thunderbolt', 'Avenger'],
 'steaming': {'netflix': True, 'amazon': False},
 'dog name': 'Box'}
```

```
del user["dog name"]
```

```
user
```

```
{'name': 'Por',
 'age': 26,
 'location': 'BKK',
 'fav_movies': ['Superman', 'Thunderbolt', 'Avenger'],
 'steaming': {'netflix': True, 'amazon': False}}
```

```
user["fav_movies"][0]
```

```
'Superman'
```

```
user["fav_movies"][1]
```

```
'Thunderbolt'
```

```
user["fav_movies"][-1]
```

```
'Avenger'
```

```
user["fav_movies"][-2]
```

```
'Thunderbolt'
```

```
user["fav_movies"][0:2] # ตัวสุดท้ายจะไม่โดนดึงมา [0:2] จะดึงมาหมดแต่ไม่ดึงตัวที่ 2 มาด้วย
```

```
['Superman', 'Thunderbolt']
```

```
## import module
import math
```

```
math.pi
```

```
3.141592653589793
```

```
from math import pi , log , exp
```

```
pi
```

```
3.141592653589793
```

```
log(5)
```

```
1.6094379124341003
```

```
## homework 01: pao ying chub
## homework 02 : bot pizza calculate cost
## 02 : bot ordering pizza
```

```
from random import choice
```

```
choice(["ค้อน","กรรไกร","กระดาษ"])
```

```
'ค้อน'
```

```
## 02 : bot ordering pizza
```