

03 ตัวแปร (Variable), ประเภทข้อมูล (Data Type)

ตัวแปร (Variable)

ตัวแปร (variable) คือ ชื่อหรือสัญลักษณ์ที่กำหนดขึ้นเพื่ออ้างอิงถึงตำแหน่งหน่วยความจำที่ใช้จัดเก็บข้อมูล

กฎการตั้งชื่อตัวแปร

- ชื่อของตัวแปรต้องขึ้นต้นด้วย a-z, A-Z หรือเครื่องหมาย _
- ภายในชื่อของตัวแปรประกอบด้วย a-z, A-Z, 0-9 หรือ เครื่องหมาย _
- ห้ามเป็นช่องว่างหรือเครื่องหมายอื่น
- Case Sensitive
- ห้ามใช้ reserved words ในการตั้งชื่อ

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	

In [1]:

```
#ตัวอย่างการตั้งชื่อตัวแปร
#Legal variable names:
myvar = "John"
my_var = "John"
_my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"

#Illegal variable names:
2myvar = "John"
my-var = "John"
my var = "John"
```

File "<ipython-input-1-3fea0c4bac54>", line 11

2myvar = "John"

^

SyntaxError: invalid syntax

In [9]:

```
#ตัวอย่างการสร้างตัวแปร
name = "My name is Python" #ตัวแปร name มีค่าเท่ากับ My name is Python เป็นชนิดข้อความ
income = 15000 #ตัวแปร income มีค่าเท่ากับ 15000 เป็นชนิดจำนวนเต็ม
grade = 4.0 #ตัวแปร grade มีค่าเท่ากับ 4.0 เป็นจำนวนทศนิยม

x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)

x = y = z = "Orange"
print(x)
print(y)
print(z)
```

Orange
Banana
Cherry
Orange
Orange
Orange

ชนิดข้อมูล (Data Types)

1. ข้อมูลชนิดตัวเลข (Number)

- int ตัวเลขจำนวนเต็ม เช่น 1, 2, 3, 4, 5
- float ตัวเลขมีทศนิยม เช่น 1.2, 2.5, 5.43, 9.345
- complex ตัวเลขจำนวนเชิงซ้อน โดยเขียนอยู่ในรูปแบบ $x+yi$ โดย
เรียก x ว่า ส่วนจริง
เรียก y ว่า ส่วนจินตภาพ
ส่วน i ในภาษาไพทอน (Python) สามารถใช้ 'j' หรือ 'J' หลังตัวเลข สร้างส่วนจินตภาพเพื่อสร้างจำนวนเชิงซ้อน

In [10]:

```
#ตัวอย่างข้อมูลชนิดตัวเลข
a = 10 #ตัวแปร a มีค่าเท่ากับ 10 เป็นชนิดจำนวนเต็ม
b = 0b1010 #ตัวแปร b เก็บข้อมูลเลขฐาน 2 มีค่าเท่ากับ 10 เป็นชนิดจำนวนเต็ม
c = 0o12 #ตัวแปร c เก็บข้อมูลเลขฐาน 8 มีค่าเท่ากับ 10 เป็นชนิดจำนวนเต็ม
d = 0xA #ตัวแปร d เก็บข้อมูลเลขฐาน 16 มีค่าเท่ากับ 10 เป็นชนิดจำนวนเต็ม

e = 10.0 #ตัวแปร e มีค่าเท่ากับ 10.0 เป็นชนิดจำนวนทศนิยม
f = 10+2j #ตัวแปร f มีค่าเท่ากับ 10+2j เป็นชนิดจำนวนเชิงซ้อน
```

2. ข้อมูลชนิดค่าความจริง (Boolean)

เป็นข้อมูลที่มีค่าเป็นจริงหรือเท็จเท่านั้น โดยจะแทนด้วย True และ False

In [11]:

```
#ตัวอย่างข้อมูลชนิด Boolean
a = True
b = False
c = 1>2
d = 5 == 3
```

3. ข้อมูลชนิด None

เป็นข้อมูลที่ไม่มีค่า หรือยังไม่ได้กำหนดค่า

In [12]:

```
#ตัวอย่างข้อมูล None
a = None
```

4. ข้อมูลแบบเรียงลำดับ (Sequence)

เป็นข้อมูลที่จัดเก็บเรียงลำดับต่อกัน ได้แก่ String, List, Tuple

4.1 ข้อมูลชนิดข้อความ (String)

เป็นข้อมูลที่ประกอบไปด้วยตัวอักษร ตัวเลข สัญลักษณ์และอักขระต่างๆ เช่น a-z, A-Z, ก-ฮ, 0-9, + - * @ * .
ป็นต้น ซึ่งจะเรียงต่อกันภายในเครื่องหมาย
double quote (" ") หรือ single quote (' ')
โดยข้อมูลต้องอยู่ต้องอยู่บรรทัดเดียวกัน และสามารถใส่เครื่องหมาย backslash (\) แทรกหากต้องการแยกบรรทัดใหม่

In [13]:

```
#ตัวอย่างข้อมูล string
str_a = "hello python"
str_b = "สวัสดี python"
str_c = "20"
str_d = '20'
str_e = "ยินดีต้อนรับสู่" \
        "ประเทศไทย"
```

4.2 ข้อมูลชนิดลิสต์ (List)

เป็นข้อมูลที่เรียงต่อกันภายในเครื่องหมายวงเล็บก้ามปู [] คั่นด้วยเครื่องหมายคอมมา (,)

In [5]:

```
#ตัวอย่างข้อมูลประเภท List
list_a = [1, 2, 3, 4, 5]
list_b = ['a', 'b', 'c', 'd', 'e']
list_c = ['python', True, 10, 4, 1.5]
list_d = [] #empty list
```

In [6]:

```
list_c # ตัวแปร list_c มีค่าเท่ากับ ['python', True, 10, 4, 1.5]
list_c[0] # ตัวแปร list_c[0] มีค่าเท่ากับ 'python'
list_c[1] # ตัวแปร list_c[1] มีค่าเท่ากับ True
list_c[2] # ตัวแปร list_c[2] มีค่าเท่ากับ 10
list_c[3] # ตัวแปร list_c[3] มีค่าเท่ากับ 4
list_c[4] # ตัวแปร list_c[4] มีค่าเท่ากับ 1.5
list_c[-1] # ตัวแปร list_c[-1] มีค่าเท่ากับ 1.5
```

Out[6]:

1.5

4.3 ข้อมูลชนิดทูเปิล (Tuple)

เป็นข้อมูลที่เรียงต่อกันภายใต้เครื่องหมายวงเล็บ หรือ ไม่ต้องมีวงเล็บครอบก็ได้ และ คำนวณด้วยเครื่องหมายคอมมา (,) ลักษณะจะเหมือนกับ List แต่จะไม่สามารถ เพิ่มเปลี่ยนแปลงแก้ไข หรือลบข้อมูลใน Tuple ได้

In [15]:

```
#ตัวอย่างข้อมูลประเภท Tuple
tuple_a = (1, 2, 3, 4, 5)
tuple_b = 'a', 'b', 'c', 'd', 'e'
tuple_c = ('python', True, 10, 4, 1.5)
tuple_d = () #empty tuple
```

In [22]:

```
tuple_c[0]
```

Out[22]:

'python'

In [23]:

```
tuple_c # ตัวแปร tuple_c มีค่าเท่ากับ ('python', True, 10, 4, 1.5)
tuple_c[0] # ตัวแปร tuple_c[0] มีค่าเท่ากับ 'python'
tuple_c[1] # ตัวแปร tuple_c[1] มีค่าเท่ากับ True
tuple_c[2] # ตัวแปร tuple_c[2] มีค่าเท่ากับ 10
tuple_c[3] # ตัวแปร tuple_c[3] มีค่าเท่ากับ 4
tuple_c[4] # ตัวแปร tuple_c[4] มีค่าเท่ากับ 1.5
tuple_c[-1] # ตัวแปร tuple_c[-1] มีค่าเท่ากับ 1.5
```

Out[23]:

1.5

5. ข้อมูลชนิดเซต (Set)

เป็นข้อมูลที่เรียงต่อกันด้วยเครื่องหมายคอมมา (,) ภายในเครื่องหมายวงเล็บปีกกา { } สามารถเพิ่ม แก้ไข ลบข้อมูลและเป็นค่าว่างได้ แต่การจัดเก็บข้อมูลจะไม่มีลำดับ และจะไม่เก็บข้อมูลที่ซ้ำซ้อนกัน ดังนั้นจะไม่สามารถเข้าถึงข้อมูลด้วยตำแหน่งข้อมูล (index)

In [24]:

```
## ตัวอย่างข้อมูลประเภท Set
set_a = {1,2.5,3,'Thailand'}
set_b = {1,2.5,3,True,'Thailand'}
set_c = {1,2.5,3,'Thailand','Thailand'}
set_d = {}
```

6. ข้อมูลชนิดดิกชันนารี (Dictionary)

เป็นข้อมูลที่เรียงต่อกันด้วยเครื่องหมายคอมมา (,) ภายในเครื่องหมายวงเล็บปีกกา { } และอยู่ในรูปแบบของ key:value

In [25]:

```
## ตัวอย่างข้อมูลประเภท Dictionary

dict_salary = {'Peter': 150000, 'Stark':200000, 'Thor':0}

dict_salary['Peter']
dict_salary['Stark']
dict_salary['Thor']
```

Out[25]:

0

การตรวจสอบชนิดของข้อมูล

การตรวจสอบชนิดข้อมูลโดยใช้ฟังก์ชัน type()

In [26]:

```
a = 20
type(a)
```

Out[26]:

int

In [27]:

```
b = 20.5
type(b)
```

Out[27]:

float

In [28]:

```
c = '20'
type(c)
```

Out[28]:

str

In [29]:

```
d = (1,2,3)
type(d)
```

Out[29]:

tuple

In [30]:

```
e = [1,2,3]
type(e)
```

Out[30]:

list

In [31]:

```
set_a = {1,2.5,3,'Thailand'}
type(set_a)
```

Out[31]:

set

In [32]:

```
dict_salary = {'Peter': 150000, 'Stark':200000, 'Thor':0}
```

In [33]:

```
type(dict_salary)
```

Out[33]:

dict

การแปลงชนิดข้อมูล

- `int(x[,base])` แปลงข้อมูล x เป็นจำนวนเต็ม จากฐานที่กำหนด
- `float(x)` แปลง x ให้เป็นทศนิยม
- `complex(real[,im])` สร้างจำนวนเชิงซ้อนจากค่า real และ imagine
- `str(x)` แปลง x ให้เป็นตัวอักษร
- `tuple(s)` แปลงข้อมูลแบบ sequence s เป็น tuple
- `list(s)` แปลงข้อมูลแบบ sequence s เป็น list
- `set(s)` แปลงข้อมูลแบบ sequence s เป็น set
- `dict(d)` แปลงข้อมูล d เป็นข้อมูล dictionary
- `chr(x)` แปลงข้อมูล x ให้เป็นอักขระ

In [34]:

```
#ตัวอย่างการแปลงเป็นตัวเลขจำนวนเต็ม
```

```
x = int(1)
y = int(2.8)
z = int("3")
```

In [35]:

```
#ตัวอย่างการแปลงตัวเลขทศนิยม
```

```
x = float(1)
y = float(2.8)
z = float("3")
w = float("4.2")
```

In [36]:

```
#ตัวอย่างการแปลงเป็นตัวอักษร
```

```
x = str("s1")
y = str(2)
z = str(3.0)
```

In [37]:

```
#Convert list to tuple
```

```
listx = [5, 10, 7, 4, 15, 3]
print(listx)
tuplex = tuple(listx)
print(tuplex)
```

```
[5, 10, 7, 4, 15, 3]
(5, 10, 7, 4, 15, 3)
```

In [38]:

```
listy = list(tuplex)
print(listy)
```

```
[5, 10, 7, 4, 15, 3]
```

In [39]:

```
# initializing string
s = 'geeks'

# printing string converting to tuple
c = tuple(s)
print (c)

# printing string converting to set
c = set(s)
print (c)

# printing string converting to list
c = list(s)
print (c)
```

```
('g', 'e', 'e', 'k', 's')
{'k', 'g', 's', 'e'}
['g', 'e', 'e', 'k', 's']
```

In [40]:

```
# Convert ASCII value to characters
a = chr(76)
b = chr(77)

print(a)
print(b)
```

L
M

Summary

สร้างตัวแปรใน Python ด้วยเครื่องหมาย = เวลาตั้งชื่อก็ตั้งให้อ่านง่ายๆ สื่อความหมายตามข้อมูลที่จัดเก็บ

data type ที่ได้แก่

- int ข้อมูลตัวเลขจำนวนเต็ม
- str ข้อมูลตัวอักษร
- float ข้อมูลตัวเลขทศนิยม
- complex ข้อมูลเลขเชิงซ้อน
- List [] ใช้กับข้อมูลทุกประเภท "แก้ไขได้"
- Tuple () ใช้กับข้อมูลทุกประเภท "แก้ไขไม่ได้"
- Set {} ข้อมูลไม่มีลำดับ ไม่ซ้ำ
- Dictionary ข้อมูลแบบ key:value

ตรวจสอบ data type โดยใช้ ฟังก์ชัน type(variable)

สามารถเปลี่ยนประเภทของข้อมูลได้ เช่น จากตัวอักษรเป็นตัวเลข, จากทศนิยมเป็นตัวอักษร

In [41]:

```
### ตัวอย่างการดึงค่าจากตัวแปร Range of Indexes###  
print(list_c[0:1])  #slice index 0 ไปจนถึง 1 (แต่ไม่รวม 1)  
print(list_c[0:3])  #slice index 0 ไปจนถึง 3 (แต่ไม่รวม 3)  
print(list_c[1:3])  #slice index 1 ไปจนถึง 3 (แต่ไม่รวม 3)  
  
print(list_c[:4])   #slice index 0 ไปจนถึง 4 (แต่ไม่รวม 4)  
print(list_c[2:])   #slice index 0 ไปจนถึงตำแหน่งสุดท้าย
```

```
['python']  
['python', True, 10]  
[True, 10]  
['python', True, 10, 4]  
[10, 4, 1.5]
```