



มหาวิทยาลัยราชภัฏนครปฐม  
Nakhon Pathom Rajabhat University

# บทที่ 4 การเขียนโปรแกรม python ประเภท ตัวแปร และ ข้อความ

รายวิชา 3602801 การเขียนโปรแกรมคอมพิวเตอร์ทางธุรกิจ

ผศ.ดร.เดช ธรรมศิริ



## การ comment ในภาษา python ใช้เครื่องหมาย # ดังตัวอย่าง

- `#This is a comment`  
`print("Hello, World!")`
- หรือ `print("Hello, World!") #This is a comment`
- หรือ
- `#print("Hello, World!")`  
`print("Cheers, Mate!")`
- **\*\*** หมายเหตุ ใน **editor** ส่วนใหญ่หากข้อมูลตรงไหนเป็นการ **comment** จะแสดงผลข้อความ เป็นสีเขียว แต่ทั้งนี้ขึ้นอยู่กับที่การตั้งค่าด้วย



# ตัวแปรในภาษา python (VARIABLES)

# สร้างตัวแปรง่ายๆด้วย =

```
variable_name = value
```

# ตัวอย่าง

```
x = 100
```

```
y = 200
```

```
hello_world = 'hello world!'
```

```
my_name = 'dech'
```

```
your_name = 'dechThailand'
```

วิธีการตั้งชื่อที่ดี

1. ใช้ตัวพิมพ์เล็กทั้งหมด
2. ห้ามขึ้นต้นด้วยตัวเลข
3. ถ้าชื่อยาวให้เชื่อมคำด้วย \_ (underscore)
4. ชื่อห้ามซ้ำกับคำสั่ง



# คำสั่งวนในภาษาไพธอน

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



# ประเภทตัวแปรของภาษา python

- # เบื้องต้นมีแค่ 4 แบบ ที่พบบ่อย ๆ
  - int: 1 2 3 4 5 6 7 8 9
  - float: 2.4 3.5 6.9 10.2
  - string: 'hello world' 'food' 'dog'
  - bool: True False

เราสามารถเช็ค data type ด้วย ฟังก์ชัน type()



## สามารถกำหนดค่าให้กับตัวแปรพร้อมกัน

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

```
Orange  
Banana  
Cherry
```



## หรือกำหนดค่าหนึ่งค่าใด ให้กับตัวแปรหลายตัว

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

```
Orange  
Orange  
Orange
```



# Unpack a Collection คือการเก็บค่าลงใน list, tuple และภาษา python อนุญาตให้เอาค่ากลับมาใส่ตัวแปร เรียกว่า unpacking

```
• fruits = ["apple", "banana", "cherry"]  
x, y, z = fruits  
print(x)  
print(y)  
print(z)
```

```
apple  
banana  
cherry
```





## การแสดงผลของตัวแปร สามารถใช้คำสั่ง print

- `x = "awesome"`  
`print("Python is " + x)`

```
Python is awesome
```

- `#`เครื่องหมายบวกร เป็นการเชื่อมข้อความกับตัวแปร ทั้งนี้ตัวแปรที่เชื่อมได้ต้องเป็นประเภท **string** เท่านั้น

- `x = "Python is "`  
`y = "awesome"`  
`z = x + y`  
`print(z)`

```
Python is awesome
```



# Global Variables คือตัวแปรที่สามารถสร้างและทำงานได้นอก function

- `x = "awesome"`

```
def myfunc():  
    print("Python is " + x)
```

```
myfunc()
```

```
Python is awesome
```



## สร้างตัวแปรไว้ภายนอก function

- `x = "awesome"`

```
def myfunc():  
    x = "fantastic"  
    print("Python is " + x)
```

```
myfunc()
```

```
print("Python is " + x)
```

```
Python is fantastic  
Python is awesome
```



# ประเภทของตัวแปร python แบบละเอียด

- Built-in Data Types

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview



## ตัวอย่างข้อมูลแบบต่าง ๆ

Example	Data Type
<code>x = "Hello World"</code>	str
<code>x = 20</code>	int
<code>x = 20.5</code>	float
<code>x = 1j</code>	complex
<code>x = ["apple", "banana", "cherry"]</code>	list
<code>x = ("apple", "banana", "cherry")</code>	tuple

Example	Data Type
<code>x = range(6)</code>	range
<code>x = {"name": "John", "age": 36}</code>	dict
<code>x = {"apple", "banana", "cherry"}</code>	set
<code>x = frozenset({"apple", "banana", "cherry"})</code>	frozenset
<code>x = True</code>	bool
<code>x = b"Hello"</code>	bytes
<code>x = bytearray(5)</code>	bytearray
<code>x = memoryview(bytes(5))</code>	memoryview



# การรับข้อมูลเก็บเข้าสู่ตัวแปร จากผู้ใช้งาน ผ่าน keyboard ด้วย input()

- #ตัวอย่าง

```
name=input("กรรณากรอกชื่อ")  
print("สวัสดีครับคุณ"+name)
```

```
>name=input("กรรณากรอกชื่อ")  
print("สวัสดีครับคุณ"+name)
```

กรรณากรอกชื่อ เดช ธรรมศิริ

```
กรรณากรอกชื่อเดช ธรรมศิริ  
สวัสดีครับคุณเดช ธรรมศิริ
```



# ข้อมูลประเภทตัวเลข ของ python

- ประกอบด้วย 3 ประเภท

int

float

Complex ที่แสดงในแบบ  $a + bj$

- ตัวอย่าง

```
x = 1      # int      print(type(x))
y = 2.8    # float    print(type(y))
z = 1j     # complex  print(type(z))
```



# การแปลงชนิดข้อมูลใน python

- `int()` ใช้แปลงข้อมูลใด ๆ เป็นเลขจำนวนเต็ม
- `float()` ใช้แปลงข้อมูลใด ๆ เป็นเลขทศนิยม
- `str()` ใช้แปลงข้อมูลใด ๆ เป็นข้อความ





## ตัวอย่างการแปลงข้อมูลเป็น integer โดยใช้ int()

- `x = int(1) # x will be 1`
- `y = int(2.8) # y will be 2`
- `z = int("3") # z will be 3`



## ตัวอย่างการแปลงเป็นจำนวนทศนิยม โดยใช้ float()

- `x = float(1)`      # x will be 1.0
- `y = float(2.8)`    # y will be 2.8
- `z = float("3")`    # z will be 3.0
- `w = float("4.2")` # w will be 4.2



## ตัวอย่างการแปลงข้อมูลเป็นข้อความด้วย str()

- `x = str("s1")` # x will be 's1'
- `y = str(2)` # y will be '2'
- `z = str(3.0)` # z will be '3.0'



# ข้อมูลประเภทข้อความ String

- `print("Hello")`  
`print('Hello')`

- #สามารถใช้เป็น ฟนทอง(single quote) ' ' หรือ ฟนหนุ(double quote) " " ปิดหัวท้ายก็ได้



## การประกาศตัวแปรเพื่อรับค่าข้อความ

- `a = "Hello"`  
`print(a)`



# ข้อความที่มีหลายบรรทัด สามารถใช้ ฟันหนู หรือฝนทอง สามอัน รวมกันได้

- `a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua."""`  
`print(a)`
- `a = '''Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.'''`  
`print(a)`

```
a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua."""  
print(a)
```

```
Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua.
```



## การนับจำนวนข้อความ โดยใช้ฟังก์ชัน len()

- `a = "Hello, World!"`  
`print(len(a))`

13



## สามารถเช็คข้อความที่อยู่ในประโยคได้ ด้วย keyword in

- `txt = "The best things in life are free!"`  
`print("free" in txt)`

True

- หรืออาจให้แสดงรายละเอียดโดยใช้เงื่อนไข `if` เข้าช่วย

- `txt = "The best things in life are free!"`  
`if "free" in txt:`  
`print("Yes, 'free' is present.")`

Yes, 'free' is present.





# การแบ่งแยกข้อความ ด้วย Slicing

- หากต้องการแบ่งข้อความ ตั้งแต่ตำแหน่งที่ 2 ถึง 5 (โดยตำแหน่งที่ 5 จะไม่ถูกรวมในข้อความที่แบ่งแยก)
- `b = "Hello, World!"`  
`print(b[2:5])`

```
llo
```



# การใช้ Slice จากจุดเริ่มต้น

- เป็นการตัดข้อมูลออกมาเฉพาะตัวหน้าสุดจนถึงตำแหน่งที่ 5 (ตำแหน่งที่ 5 จะไม่ถูกนำไปรวม)
- `b = "Hello, World!"`  
`print(b[:5])`

```
Hello
```



## การใช้ Slice จนถึงตำแหน่งสุดท้าย

- เริ่มทำการ `slice` จากตำแหน่งที่ 2 จนถึงตำแหน่งสุดท้าย
- `b = "Hello, World!"`  
`print(b[2:])`

```
llo, World!
```



# การปรับข้อความเป็นตัวอักษรตัวใหญ่ ด้วย upper() method

- `a = "Hello, World!"`  
`print(a.upper())`

```
HELLO, WORLD!
```



## การปรับข้อความเป็นตัวอักษรตัวเล็ก ด้วย lower() method

- `a = "Hello, World!"`  
`print(a.lower())`

```
hello, world!
```



## ลบช่องว่าง ก่อนและท้ายข้อความ ด้วย strip() method

- `a = " Hello, World! "`  
`print(a.strip()) # returns "Hello, World!"`



## การแทนที่ข้อความด้วย replace() method

- `a = "Hello, World!"`  
`print(a.replace("H", "J"))`

```
Jello, World!
```



## การตัดแบ่งข้อความ ด้วย split() method

- `a = "Hello, World!"`  
`print(a.split(","))` # returns ['Hello', ' World!']

```
['Hello', ' World!']
```





## การเชื่อมข้อความ โดยใช้เครื่องหมาย +

- `a = "Hello"`  
`b = "World"`  
`c = a + b`  
`print(c)`

```
HelloWorld
```



## การเชื่อมข้อความ โดยเพิ่มช่องว่าง

- `a = "Dech"`  
`b = "Thammasiri"`  
`c = a + " " + b`  
`print(c)`

```
Dech Thammasiri
```



# การเชื่อมข้อความกับตัวเลข ด้วย String Format

- # ตัวอย่างนี้จะเกิด Error
- `age = 36`  
`txt = "My name is John, I am "+age`  
`print(txt)`

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-52-08a9fe7c94d8> in <module>()  
      1 age =36  
----> 2 txt ="My name is John, I am "+age  
      3 print(txt)  
  
TypeError: can only concatenate str (not "int") to str
```

- #สามารถแก้ไขได้โดย
- `age = 36`  
`txt = "My name is John, and I am {}"`  
`print(txt.format(age))`

```
My name is John, and I am 36
```



## การใช้ String Format กับตัวแปรหลายตัว

- `quantity = 3`  
`itemno = 567`  
`price = 49.95`  
`myorder = "I want {} pieces of item {} for {}`  
`dollars."`  
`print(myorder.format(quantity, itemno, price))`

```
I want 3 pieces of item 567 for 49.95 dollars.
```



# การเชื่อมข้อความกับตัวเลข

- # หรืออาจแก้ไขโดย แปลงข้อมูลประเภทตัวเลขให้เป็นข้อความ
- `age = 36`  
`txt = "My name is John, I am " + str(age)`  
`print(txt)`

```
My name is John, I am 36
```



# Escape Character

- ตัวอย่างนี้โปรแกรมจะเกิด **Error** เนื่องจาก มี **double quote** ภายใน **string**

- `txt = "We are the north."`

- สามารถแก้ปัญหาได้ดังนี้

- `txt = "We are the so-called \"Vikings\" from the north."`

```
txt = "We are the so-called "Vikings" from the north."

File "<ipython-input-57-0fa5c340b616>", line 1
    txt = "We are the so-called "Vikings" from the north."
                                   ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
We are the so-called "Vikings" from the north.
```



# Escape Characters

Code	Result
\'	Single Quote
\\	Backslash
\n	New Line
\r	Carriage Return
\t	Tab
\b	Backspace
\f	Form Feed
\ooo	Octal value
\xhh	Hex value



# ตัวแปรประเภท บูลีน Python Booleans

- ค่าที่ได้จะมีแค่ จริง (True) หรือ เท็จ (False)
- ตัวอย่าง

```
print(10 > 9)
print(10 == 9)
print(10 < 9)
```

```
True
False
False
```

- ผลลัพธ์ที่ได้จะแสดงออกมาเป็นค่า **True or False**





## แต่ถ้าอยากแสดงเป็นข้อความอาจใช้ if เข้ามาช่วยเช็คค่า

- a = 200  
b = 33

```
if b > a:  
    print("b is greater than a")  
else:  
    print("b is not greater than a")
```

```
b is not greater than a
```



ขอบคุณครับ