

Python Basic

ฉัตรชัย เกษมทวีโชค

Chatchai.kase@ku.th

PYTHON PROGRAMS

- เป็นโปรแกรมที่มีลำดับของข้อกำหนดและคำสั่ง (definitions and commands)
 - ข้อกำหนดถูกตรวจสอบประเมินก่อน (evaluated)
 - คำสั่งที่เขียนถูกประมวลผล (executed) โดย Python interpreter
- Commands (Statement) เป็นคำสั่งที่ต้องการให้ตัวแปลภาษาสั่งการให้คอมพิวเตอร์ทำงานบางอย่าง
- เราสามารถพิมพ์คำสั่งได้โดยตรงใน Shell หรือเก็บไว้ในไฟล์แล้วอ่านคำสั่งใน Shell และประเมินผลการทำงานอีกครั้ง

Python Syntax

- Python Indentation เป็นช่องว่างหน้าคำสั่งในบรรทัด โดยใช้ TAB เป็นคีย์กำหนด Indent
- การใช้ Indentation ทำให้คำสั่งอ่านได้ง่ายขึ้น
- เมื่อเขียนภาษา Python ต้องระวังการใช้ indentation เพราะ python ใช้ indent เป็นตัวกำหนด block of code

ภาษา Python

```
if 10 > 5 :  
    print("10 is more than 5. ")  
    print("5 x 2 = 10.")  
  
print("end program...")
```

ภาษา C, JAVA

```
if (10 > 5)  
{  
    print("10 is more than 5. ")  
    print("5 x 2 = 10.")  
}  
  
print("end program...")
```

Python Comments

- Comment เป็นคำอธิบายโปรแกรม ทำให้ผู้สนใจโปรแกรม อ่านการทำงานของโปรแกรมได้ง่ายขึ้น
- ในบางกรณี Comment ก็ถูกใช้งานเพื่อยุติการรันบางคำสั่ง

```
#This is a comment  
print("Hello, World!")
```

```
print("Hello, World!") #This is a comment
```

```
#print("Hello, World!")  
print("Cheers, Mate!")
```

Multi Line Comments (คอมเมนต์หลายบรรทัด)

```
"""  
This is a comment  
written in  
more than just one line  
"""  
  
print("Hello, World!")
```

https://www.w3schools.com/python/python_comments.asp

Python Variables (ตัวแปร)

- ตัวแปรไม่ต้องประกาศ ไม่ต้องกำหนดประเภทข้อมูล (Data Type)
- อยากรู้ตัวแปร ก็กำหนดค่าเลย และเปลี่ยนประเภทข้อมูลได้เลย
- เน้นสะดวกใช้งาน

```
x = 5
y = "John"
print(x)
print(y)
```

```
x = 4          # x is of type int
x = "Sally"    # x is now of type str
print(x)
```

Single or Double Quotes?

กำหนดตัวแปรข้อความ ใช้ " หรือ ' ก็ได้เหมือนกัน

```
x = "John"
# is the same as
x = 'John'
```

```
a = 4
A = "Sally"
#A will not overwrite a
```

Case-Sensitive

ชื่อตัวแปรเดียวกัน แต่มีตัวอักษรใหญ่และเล็กต่างกัน ถือว่าเป็นตัวแปรคนละตัวกัน

Variable Names

- กฎการตั้งชื่อตัวแปร มีดังนี้
 - ขึ้นต้นด้วยตัวอักษรหรือ underscore (_)
 - ห้ามขึ้นต้นด้วยตัวเลข
 - ชื่อตัวแปรจะได้ตัวอักษร หรือ ตัวเลข หรือ underscore (_)
 - ชื่อตัวแปรเป็น case sensitive เช่น ตัวแปร Myname กับ MYNAME เป็นตัวแปรคนละตัวกัน

ชื่อตัวแปรถูกต้อง ใช้งานได้

```
myvar = "John"  
my_var = "John"  
_my_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"
```

ชื่อตัวแปรไม่ถูกต้อง

```
2myvar = "John"  
my-var = "John"  
my var = "John"  
  
My$var = "John"
```

แบบฝึกหัดการตั้งชื่อตัวแปร

M3 a	A4_b	1First_Name
\$LstName	_3BB	Tel-Num
aN24	_Address	

- จงเขียนรายชื่อตัวแปรลงในช่องว่าง

ตัวแปรที่ตั้งชื่อถูกต้อง	
ตัวแปรที่ตั้งชื่อไม่ถูกต้อง	

Python Data Types

Data Characteristics	Data Types
Text (ข้อความ)	str
Numeric (ตัวเลข)	int, float, complex
Sequence (รายการค่า)	list, tuple, range
Mapping (การเทียบชื่อกับค่า)	dict
Set (เซต)	Set, frozenset
Boolean (ค่าความจริง)	Bool
Binary (เลขฐานสอง)	bytearray, memoryview

https://www.w3schools.com/python/python_variables.asp

Python Data Types

Example	Data Type
<code>x = "Hello World"</code>	<code>str</code>
<code>x = 20</code>	<code>int</code>
<code>x = 20.5</code>	<code>float</code>
<code>x = 1j</code>	<code>complex</code>
<code>x = ["apple", "banana", "cherry"]</code>	<code>list</code>
<code>x = ("apple", "banana", "cherry")</code>	<code>tuple</code>
<code>x = range(6)</code>	<code>range</code>
<code>x = {"name" : "John", "age" : 36}</code>	<code>dict</code>
<code>x = {"apple", "banana", "cherry"}</code>	<code>set</code>
<code>x = frozenset({"apple", "banana", "cherry"})</code>	<code>frozenset</code>
<code>x = True</code>	<code>bool</code>
<code>x = b"Hello"</code>	<code>bytes</code>
<code>x = bytearray(5)</code>	<code>bytearray</code>
<code>x = memoryview(bytes(5))</code>	<code>memoryview</code>

Assign Multiple Values

- Many Values to Multiple Variables (หนึ่งคำสั่งกำหนดค่าให้ตัวแปรหลายตัว หนึ่งค่าต่อหนึ่งตัวแปร)

```
x, y, z = "Orange", "Banana", "Cherry"
```

- One Value to Multiple Variables (หนึ่งคำสั่งกำหนดค่าให้ตัวแปรหลายตัว หนึ่งค่ากำหนดค่าให้กับตัวแปรทุกตัว)

```
x = y = z = "Orange"
```

- Unpack a Collection (กำหนดรายการค่าให้กับตัวแปรหลายตัว)

```
fruits = ["apple", "banana", "cherry"] # fruits is a list  
x, y, z = fruits
```

แบบฝึกหัดกำหนดค่าตัวแปร

- จงบอกว่า การกำหนดค่าตัวแปรถูกต้องตามหลักการเขียนภาษา Python หรือไม่ และกรณีที่ผิด ให้นิสิตแก้ไขในถูกต้อง

ความต้องการ	คำสั่ง
สร้างตัวแปรและบันทึกชื่อลูกค้า FirstName เท่ากับ John	
สร้างตัวแปรและบันทึกจำนวนอายุ Age เท่ากับ 38 ปี	
สร้างตัวแปรและบันทึกจำนวนเงิน Money เท่ากับ 4500.50	
สร้างตัวแปรและบันทึกสถานะของการโอนเงิน Transfer ที่เป็นจริง	
สร้างตัวแปรและบันทึกรายการประเภทสินค้าในคลัง Category ได้แก่ CPU, Mainboard, Storage, Graphic Card, Case	

นิพจน์ (EXPRESSIONS)

- นิพจน์เป็นการรวมระหว่าง Objects (Operands) และ Operators
- นิพจน์จะให้ผลลัพธ์เป็นค่าที่มีประเภทข้อมูล
- การเขียนนิพจน์ จะมี Operator คั่นกลางระหว่าง Object 2 ตัว
 <object> <operator> <object>
- ตัวอย่างเช่น

	Object1	Operator	Object2	=
3 * 5	3	*	5	
9 / 2	9	/	2	
"Hello" + " World"	"Hello"	+	" World"	
SEX == "MALE"	SEX	==	"MALE"	
AGE > 25	AGE	>	25	

ตัวดำเนินการ (Python Operators)

- ประเภทของตัวดำเนินการในภาษา Python
 - Arithmetic operators
 - Assignment operators
 - Comparison operators
 - Logical operators
 - Identity operators
 - Membership operators
 - Bitwise operators

Arithmetic Operators

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Assignment Operators (เครื่องหมายลดรูป)

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

BINDING VARIABLES AND VALUES

- การใช้เครื่องหมายเท่ากับ (= equal sign) เป็นการกำหนดค่าให้กับค่าตัวแปร
 - $\text{Pi} = 3.14159$
 - $\text{Pi_approx.} = 22/7$
- ค่าตัวแปรจะจัดเก็บไว้ในหน่วยความจำของคอมพิวเตอร์
- ชื่อตัวแปรจะผูกกับค่าตัวแปร
- สามารถเรียกใช้ค่าตัวแปรผ่านชื่อตัวแปรได้

ABSTRACTING EXPRESSIONS

- ประโยชน์ของการกำหนดชื่อค่าตัวแปรคือ
 - การที่เราสามารถนำค่าข้อมูลกลับมาใช้ซ้ำได้ (reuse names)
 - ง่ายต่อเปลี่ยนแปลงคำสั่งในภายหลัง

pi = 3.14159

radius = 2.2

area = pi*(radius**2)

circumference = 2 * pi * radius

print function

- เป็นฟังก์ชันที่ใช้ในการแสดงผลลัพธ์ออกทางหน้าจอ โดยจะรับรายการค่าที่ต้องการแสดงผล

`print(format_string)`

- `format_string` คือรูปแบบข้อความหรือรายการค่าที่ต้องการแสดงผล

print function

```
a=1
```

```
b='ball'
```

#Method 1

```
print('I have %d %s' %(a,b))
```

I have 1 ball

#Method 2

```
print('I have',a,b)
```

#Method 3

```
print('I have {}'.format(a,b))
```

#Method 4

```
print('I have ' + str(a) + ' ' + b)
```

แบบฝึกหัดการแสดงผลด้วยคำสั่ง print

- กำหนดให้ตัวแปร Nisit = "Sombat", Age = 19, GPA = 3.25, Major = "CS", Year = "second", University = "Kasetsart"

ตัวอย่างผลลัพธ์ที่ต้องการ	คำสั่ง print
I am Sombat.	
I am 19 years old.	
I have a grade of 3.25 in CS major.	
I am a second-year student in Kasetsart University.	

input function

- เป็นฟังก์ชันที่ใช้ในการรับค่าจากคีย์บอร์ด โดยจะส่งค่าดังกล่าวเป็นข้อความ (string) เท่านั้น

`Variable = input(Prompt)`

- Variable คือตัวแปรที่ต้องการให้เก็บค่าจากคีย์บอร์ด
 - Prompt คือข้อความที่ต้องการแสดงในตอนรับค่า (ถ้ามี)
-
- ถ้าต้องการให้ค่าที่รับมาเป็นประเภทข้อมูลอื่น ๆ เช่น เลขจำนวนเต็ม (int) หรือ เลขจำนวนทศนิยม (float)

TYPE CONVERSIONS (CAST)

- แปลงค่าตัวแปรจากประเภทหนึ่งเป็นอีกประเภทหนึ่ง
 - `float(3)` แปลงตัวเลขจำนวนเต็ม (integer) 3 เป็นตัวเลขจำนวนทศนิยม (float) 3.0
 - `int(3.9)` แปลงตัวเลขจำนวนทศนิยม (float) 3.9 เป็นเลขจำนวนเต็ม (integer) 3
 - `str(3.4)` แปลงตัวเลข (float) 3.4 เป็นข้อความ (string) "3.4"

input function

```
print('Enter your name:')  
x = input()  
print('Hello, ' + x)
```

```
x = input('Enter your name:')  
print('Hello, ' + x)
```

```
age = input("What is your age?")  
age = int(age)  
print (age)
```

```
age = int(input("What is your age?"))  
print (age)
```

```
try:  
    value=int(input("Type a number:"))  
except ValueError:  
    print("กรุณากรอกเป็นตัวเลขเท่านั้น")
```