

Wk02



Flowgorithm

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...")
```

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

Python Comments

- Comment เป็นคำอธิบายโปรแกรม ทำให้ผู้สนใจโปรแกรม อ่านการทำงานของโปรแกรมได้ง่ายขึ้น
- ในบางกรณี Comment ก็ถูกใช้งานเพื่อหยุดการรันบางคำสั่ง

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

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

ไม่ต้องรันคำสั่ง `print("Hello, World!")`

```
#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)
- อยากใช้ตัวแปร 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'
```

Case-Sensitive

ชื่อตัวแปรเดียวกัน แต่มีตัวอักษรใหญ่และเล็กต่างกัน ถือว่าเป็นตัวแปรคนละตัวกัน

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

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

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"
```

Assign Multiple Values

- Many Values to Multiple Variables

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

- One Value to Multiple Variables

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

- Unpack a Collection

```
fruits = ["apple", "banana", "cherry"]  
x, y, z = fruits
```

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

Python Data Types

Data Charecteristics	Data Types
Text	str
Numeric	int, float, complex
Sequence	list, tuple, range
Mapping	dict
Set	Set, frozenset
Boolean	Bool
Binary	bytearray, memoryview

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

Python Data Types

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
<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

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

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"

OBJECTS

programs จัดการข้อมูลต่าง ๆ ในรูปแบบ Data Object

- `int` – represent **integers**, ex. 5
- `float` – represent **real numbers**, ex. 3.27
- `bool` – represent **Boolean** values `True` and `False`
- `NoneType` – **special** and has one value, `None`
- can use `type()` to see the type of an object

```
>>> type(5)
```

```
int
```

```
>>> type(3.0)
```

```
float
```

*what you write into
the Python shell*

*what shows after
hitting enter*

นิพจน์ (**EXPRESSIONS**)

- นิพจน์เป็นการรวมระหว่าง Objects 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	

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	$x // y$

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

Assignment Operators

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

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

Comparison Operators

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

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

Other Operators

Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>

Identity Operators

Operator	Description	Example
is	Returns True if both variables are the same object	<code>x is y</code>
is not	Returns True if both variables are not the same object	<code>x is not y</code>

Membership Operators

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	<code>x in y</code>
not in	Returns True if a sequence with the specified value is not present in the object	<code>x not in y</code>

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

SIMPLE OPERATIONS

- ทำงานในวงเล็บก่อน (parentheses)
- หลังจากการทำงานในวงเล็บ Python จะทำงานตามลำดับต่อไปนี้
 - **
 - *
 - /
 - + และ - จากคำสั่งซ้ายไปขวา
- ตัวอย่างเช่น

$$(8+2)**2-4*2/2 =$$

$$8+2**2-4*2/2 =$$

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

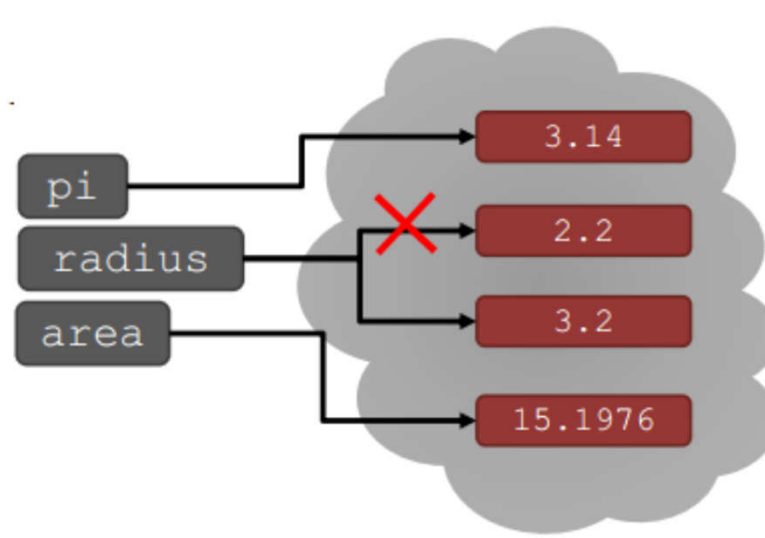
area = pi*(radius**2)

circumference = 2 * pi * radius

CHANGING BINDINGS

- เราสามารถกำหนดค่าใหม่ (re-bind) ให้กับตัวแปร
 - ค่าเดิมที่อาจจะอยู่ในตำแหน่งที่เดิมในหน่วยความจำ แต่ไม่มีการผูกหรือใช้งานแล้ว
 - ค่าของพื้นที่จะไม่เปลี่ยนแปลงจนกระทั่งคุณจะบอกให้คอมพิวเตอร์ทำการคำนวณค่าใหม่

```
pi = 3.14  
radius = 2.2  
area = pi*(radius**2)  
radius = radius+1
```



input function

- เป็นฟังก์ชันที่ใช้ในการรับค่าจากคีย์บอร์ด โดยจะส่งค่าดังกล่าวเป็นข้อความ (string)

Variable = input(Prompt)

- Variable คือตัวแปรที่ต้องการให้เก็บค่าจากคีย์บอร์ด
- Prompt คือข้อความที่ต้องการแสดงในตอนรับค่า (ถ้ามี)

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

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

- ถ้าต้องการให้ค่าที่รับมาเป็นประเภทข้อมูลอื่น ๆ เช่น เลขจำนวนเต็ม (int) หรือ เลขจำนวนทศนิยม (float)

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

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