

Constants, Variables & Basic Functions

(ค่าคงที่ , ตัวแปร และ ฟังก์ชันพื้นฐาน)



Outline

- ค่าคงที่
 - จำนวนเต็ม
 - จำนวนจริง
 - สายอักขระ หรือ สตริง
 - ค่าความจริง หรือ บูลีน
- ตัวแปร
- ฟังก์ชันพื้นฐาน
 - ฟังก์ชันรับและแสดงข้อมูล
 - ฟังก์ชันเกี่ยวกับชนิดข้อมูล
 - ฟังก์ชันเกี่ยวกับสตริง
 - ฟังก์ชันทางคณิตศาสตร์

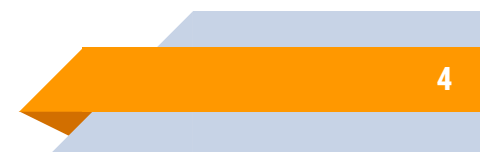
1

ค่าคงที่



ค่าคงที่

- จำนวนเต็ม (Integers)
- จำนวนจริง (Floating-point numbers)
- สายอักขระ หรือ สตริง (Strings)
- ค่าความจริง หรือ บูลีน (Boolean)



▪ จำนวนเต็ม (Integers)

ถูก

5

2021

-8800

0

ผิด

9.0 (เป็นจำนวนจริง)

"9" (เป็นสตริง)

'976' (เป็นสตริง)

▪ จำนวนจริง (Floating-point numbers)

ถูก

0.0

314159.

-0.1234

3e8

314.159E-2

0.145e7

1e9

ผิด

2e0.5

3E-1.67

3 e 1

■ สายอักขระ หรือ สตริง (Strings)

ถูก

`"Comp Prog"`

`'_ _ _ _ _'`

`" It's an exam. "`

`' I said "NO" '`

`''' I said "It's an exam" '''`

`""" I said "It's an exam" """`

`"3.14159"`

`'-0.012 e -4'`

ผิด

`'Comp Prog"`

`"_ _ _ _ _'`

`""" It's an exam. """`

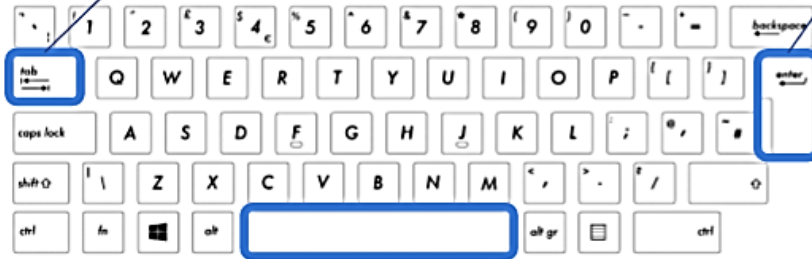
`'' I said "NO" ''`

`''' I said "It's an exam" '''`

Tab and Newline Characters

'\t' เป็นสตริงที่มีความยาว 1
และแทนตัวอักษร tab

'\n' เป็นสตริงที่มีความยาว 1
และแทนตัวอักษร newline



' ' (blank) เป็นสตริงที่มีความยาว 1

เช่น 'Anna\tLee\n 0123456\n12' พิมพ์ออกมาจะได้

Anna Lee

0123456

n12

- ค่าความจริง หรือ บูลีน (Boolean)

True

False

Comments

```
""" Comment เป็นส่วนของโปรแกรมที่ไม่ใช่คำสั่งที่ทำงาน  
แต่เป็นคำอธิบายสำหรับคนที่อ่านโปรแกรม เช่น  
This program reads temperature in celsius  
and calculates temperature in Fahrenheit  
"""
```

```
c=int(input('enter temp. in c')) # read input  
f=c*9/5+32 # calculate  
print(c,'celsius =',f,'degree f') #print output
```

“

Quiz

Quiz

ค่าคงที่ต่อไปนี้เป็นชนิดใด (จำนวนเต็ม จำนวนจริง สตริง ค่าความจริง) หรือ ทำให้เกิด error

41500

41500.0

41500.

4.15e2

'41500'

-0

-0.0

"0"

'0 . 0'

0 . 0

1dayfree

'1day free'

'1day"

"a\nb"

'1+2=3'

2

ตัวแปร



ตัวแปร

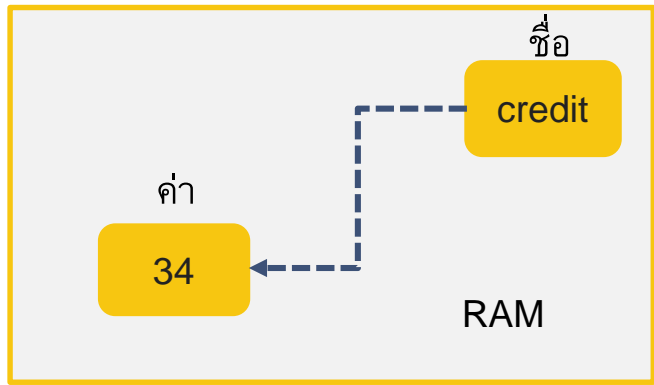
- ชื่อ และชนิดของตัวแปร
- การกำหนดค่าตัวแปร (Variable assignment)



▪ ตัวแปร (Variables)

มีที่เก็บตัวแปรใน memory (RAM)

มีชื่อที่ใช้ เชื่อมกับที่ในหน่วยความจำใช้ เก็บค่าในตัวแปร





■ การตั้งชื่อตัวแปร

ขึ้นต้นด้วยตัวหนังสือ (A-Z, a-z) หรือ _ (underscore)

ตามด้วยตัวหนังสือ ตัวเลข หรือ _ ก็ตัวก็ได้ ชื่อตัวแปรควรจะสื่อความหมาย

Example

✓ x crs1 credit_1 c_181 day_month tempDate

✗ 181 c-181 day.month 181crs

ตัวหนังสือ : ตัวใหญ่ต่างจากตัวเล็ก Name กับ name ไม่ใช่ตัวแปรเดียวกัน

X กับ x ไม่ใช่ตัวแปรเดียวกัน

▪ ข้อควรระวัง

ห้าม ใช้ blank ' ' หรือสัญลักษณ์อื่น (+ - * / & \$ @ ! \$ ^ ...) ในชื่อ ตัวแปร

ห้าม ตั้งชื่อตัวแปรซ้ำกับ **keywords**

ตัวอย่างของ **Keywords**

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

▪ ชนิดของตัวแปร

ชนิดของตัวแปรเป็นชนิดของค่าที่เก็บในตัวแปร อาจเป็น

- จำนวนเต็ม (int)
- จำนวนจริง (float)
- สายอักขระ (string)
- บูลีน (boolean)

ตัวแปรในภาษาไพธอน เปลี่ยนชนิดตามค่าที่เก็บ

▪ การกำหนดค่าตัวแปร Variable assignment

ชื่อตัวแปร = นิพจน์

นิพจน์สร้างจากตัวแปร ค่าคงที่ ตัวกระทำ

ชื่อตัวแปร, ..., ชื่อตัวแปร = นิพจน์, ..., นิพจน์

จำนวนตัวแปรทางซ้าย เท่ากับ จำนวนนิพจน์ ข้างขวา

■ ตัวอย่างการกำหนดค่าตัวแปร

```
radius = 2.0
```

- เก็บจำนวนจริง 2.0 ในตัวแปร radius เป็นตัวแปรชนิด float

```
rad = 2
```

- เก็บจำนวนเต็ม 2 ในตัวแปร rad เป็นตัวแปรชนิด int

```
area = 3.14159*radius**2
```

```
area = 'undefined'
```

- หาค่าจากนิพจน์ $3.14159*radius**2$ แล้วไปเก็บในตัวแปร area ที่เป็นตัวแปรชนิด float ในขณะนั้น
- เก็บสตริง 'undefined' ในตัวแปร area ทำให้ตัวแปร**เปลี่ยน**เป็นชนิดสตริง

▪ ตัวอย่างการกำหนดค่าตัวแปร

radius=2

Radius='r'

- เก็บจำนวนเต็ม 2 ในตัวแปร radius เป็นตัวแปรชนิด int
- Radius เป็นอีกตัวแปรที่เป็นชนิดสตริง

name , age = 'Tom' , 22

- เก็บ 'Tom' ในตัวแปร name, 22 ในตัวแปร age

x , y, z = 0, 1, 2

- เก็บ 0, 1, 2 ในตัวแปร x, y, z ตามลำดับ

“

Quiz

Quiz

ข้อความต่อไปนี้ชื่อตัวแปรเป็นในโปรแกรมภาษาไพธอนได้หรือไม่

distance day fee 'day fee' grade3.5 grade3

fee_2day fee-2day _2dayfee 2dayfee fee2Days

3

ฟังก์ชันพื้นฐาน



ฟังก์ชันพื้นฐาน (Some Basic Functions)

- print**
- input**
- type**
- int, float, str**
- len**
- strip**
- split**

▪ ฟังก์ชัน (Function)

ฟังก์ชัน (function) เป็นโปรแกรมย่อยที่สร้างขึ้นเพื่อ

- ให้ทำงานเฉพาะอย่าง
- สามารถนำมาใช้ได้ใหม่ภายหลัง
- ทำให้เกิดความสะดวกในการจัดทำโปรแกรมขนาดใหญ่
- ช่วยในการตรวจสอบความถูกต้องของโปรแกรมได้ง่ายขึ้น

ในภาษาไพธอนมีฟังก์ชันให้ใช้งาน (Built-in function) อยู่มากมาย

■ Import

ฟังก์ชันของภาษาไพธอนถูกแบ่งจัดเก็บอยู่ในโมดูล (module)

- ก่อนที่เราจะใช้ฟังก์ชัน ต้อง “นำเข้า” (import) โมดูลที่เก็บฟังก์ชันนั้นไว้
- คำสั่ง import เป็นคำสั่งที่ใช้ในการนำเข้าโมดูลหรือฟังก์ชันที่อยู่ในโมดูลต่าง ๆ
- `import <<ชื่อโมดูล>>` เช่น `import math`
- `import <<ชื่อโมดูล>> as <<ชื่อใหม่>>` เช่น `import math as m`
- นำเข้าโมดูล `math` แต่เปลี่ยนชื่อการอ้างถึงผ่านตัวแปร `m` ซึ่งสามารถเรียกใช้ งานค่าคงที่และฟังก์ชันภายในได้ตามปกติ เช่น `m.exp(3,2)` หรือ `math.log(2)`

▪ ตัวอย่าง

```
import math  
area = math.pi*math.pow(radius,2)
```

```
import math as m  
area = m.pi*m.pow(radius,2)
```

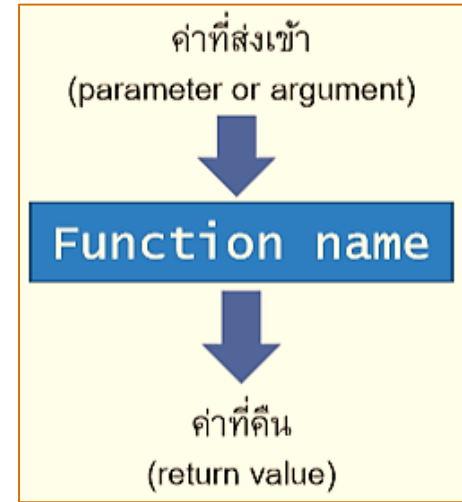
■ การใช้ฟังก์ชัน

รูปแบบ : <<ชื่อฟังก์ชัน>>(รายการของค่าที่ส่งเข้าในฟังก์ชัน)

ตัวอย่าง : `input("Enter n:")`

การใช้งานฟังก์ชัน ทำได้โดย

- เรียกใช้ฟังก์ชัน (call function)
- ส่งค่าเข้า (pass) ตามจำนวนที่ฟังก์ชันกำหนด โดย
 - ระบุอยู่ภายในเครื่องหมาย ()
 - คั่นด้วยเครื่องหมาย , หากข้อมูลที่ส่งไปให้กับฟังก์ชันมีมากกว่าหนึ่งตัว
- เราสามารถใช้บ้างฟังก์ชันในไพธอน เช่น `print`, `int`, `type` โดยไม่ต้อง `import`
- สำหรับฟังก์ชันในโมดูลที่นำเข้า ต้องอ้างชื่อโมดูล



■ ตัวอย่าง

- `print('Is', c, 'degree celsius comfort?', c == 25)`



เรียกใช้ฟังก์ชัน `print()` โดยส่งค่า 4 ค่าเข้าไป แต่ละค่าคั่นด้วย ,

- `type(c != 30)`

เรียกใช้ฟังก์ชัน `type()` เพื่อหาชนิดของค่าข้อมูลที่ระบุ

- `import math`

`area = math.pi*math.pow(radius,2)`

เรียกใช้ฟังก์ชัน `pow()` เพื่อหาค่ายกกำลัง ($radius^2$)

▪ ค่าที่คืนจากฟังก์ชัน

สมมติให้ $c=2$, $x=3$

Examples:

◦ `print('Type:', c, 'sq=', x*2)`

ไม่คืนค่า

◦ `type(c != 30)`

คืนค่า bool

◦ `int('30')`

คืนค่า 30

◦ `abs(8*2-56/7)`

คืนค่า 8.0

Examples:

◦ `math.ceil(3.2)`

คืนค่า 4

◦ `math.floor(3.2)`

คืนค่า 3

◦ `math.pow(2, 0.5)`

คืนค่า 1.4142135623730951

◦ `input("Name :")`

คืนค่าสตริงที่ผู้ใช้พิมพ์เข้ามา

▪ ฟังก์ชัน print

พารามิเตอร์ของฟังก์ชัน print มีกี่ค่าก็ได้ ค่าที่ส่งเข้าไปให้ฟังก์ชัน print อาจเป็น integer, float, string, ... คื้นค่า None หรืออาจเรียกว่าไม่คื้นค่า เพราะเราจะไม่นำค่า None ไปใช้ต่อ

```
fn = 'John'  
ln = 'Fox'  
print(fn, ln)  
print(fn, 2, ln, '.')
```

ผลลัพธ์ที่แสดงบนหน้าจอ
John Fox
John 2 Fox .

▪ ฟังก์ชัน type

พารามิเตอร์ของฟังก์ชัน type มี 1 ค่า
ค่าที่ส่งเข้าไปให้ฟังก์ชัน type เป็นชนิดใดก็ได้
ฟังก์ชัน type คืนค่าเป็นชนิดของค่าที่ส่งไป

```
type(3.14)
```

```
<class 'float'>
```

```
type(3)
```

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

```
type("3.14")
```

```
<class 'str'>
```

Type Error

```
"test"+1
```

```
32/'4'
```

```
x=3.14
```

```
type(x)
```

```
<class 'float'>
```

```
x="3.14"
```

```
type(x)
```

```
<class 'str'>
```

▪ ฟังก์ชันเปลี่ยนชนิดของค่า int, float, str

ฟังก์ชัน int, float, str รับพารามิเตอร์ 1 ค่า

ฟังก์ชัน int นำพารามิเตอร์ที่รับเข้าไปแปลงเป็นจำนวนเต็มและส่งคืน

ฟังก์ชัน float นำพารามิเตอร์ที่รับเข้าไปแปลงเป็นจำนวนจริงและส่งคืน

ฟังก์ชัน str นำพารามิเตอร์ที่รับมาไปแปลงเป็นสตริงและส่งคืน

- ✓ แปลง int \Leftrightarrow float ได้
- ✓ แปลง int, float \Rightarrow string ได้
- ✓ แปลง string \Rightarrow int , float ได้ถ้าสตริงอยู่ในรูปแบบของ int, float

▪ ฟังก์ชันเปลี่ยนชนิดของค่า

`int(3.1)`

ได้ 3 ที่เป็นชนิดจำนวนเต็ม

`int('3.1')`

เกิด value error

`str(3)`

ได้ '3' ที่เป็นชนิดสายอักขระ

`float(3)`

ได้ 3.0 ที่เป็นชนิดจำนวนจริง

`float('3')`

ได้ 3.0 ที่เป็นชนิดจำนวนจริง

`int('hijk')`

เกิด value error

▪ ฟังก์ชันรับข้อมูล input

พารามิเตอร์ของฟังก์ชัน input มี 1 ค่า คือ สตริงที่แสดงที่หน้าจอเมื่อรอรับค่าจากผู้ใช้

ฟังก์ชัน input รับสตริงที่ผู้ใช้พิมพ์เข้ามาจนเคาะ enter และ คืนค่าเป็นสตริงนั้น

เช่น

```
n = input('type data:')
```

- แสดงข้อความ 'type data:'
- รับสายอักขระทางแป้นพิมพ์มาเก็บในตัวแปร n

```
type data : 23↵
```

```
ได้ n เป็นตัวแปรชนิด string เก็บค่า '23'
```

▪ ฟังก์ชันรับข้อมูล input (ต่อ)

```
n = int(input('Enter integer:'))
```

- แสดงข้อความ 'Enter integer:'

- รับสายอักขระ แล้วเปลี่ยนเป็น int เก็บในตัวแปร n

```
Enter integer:23↵
```

ได้ n เป็นตัวแปรชนิด int เก็บค่า 23

```
Enter integer:2.3↵
```

เกิด error เพราะไม่สามารถแปลงสตริง 2.3 ที่รับเข้า เป็นจำนวนเต็มได้

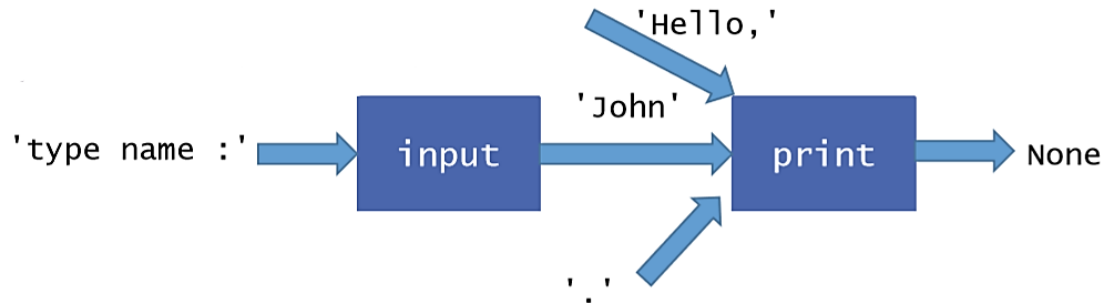
▪ การส่งค่าจากฟังก์ชันไปให้อีกฟังก์ชัน

```
name = input('type name :')  
print('Hello,',name, '.')
```

```
type name :John↵  
Hello, John .
```

```
type name : John ↵  
Hello, John .
```

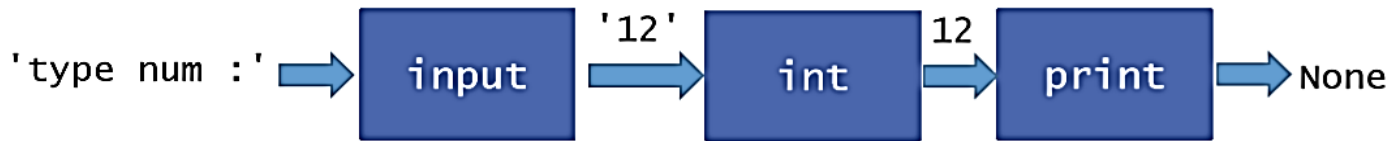
เหมือนกับ `print('Hello,',input('type name :'), '.')`



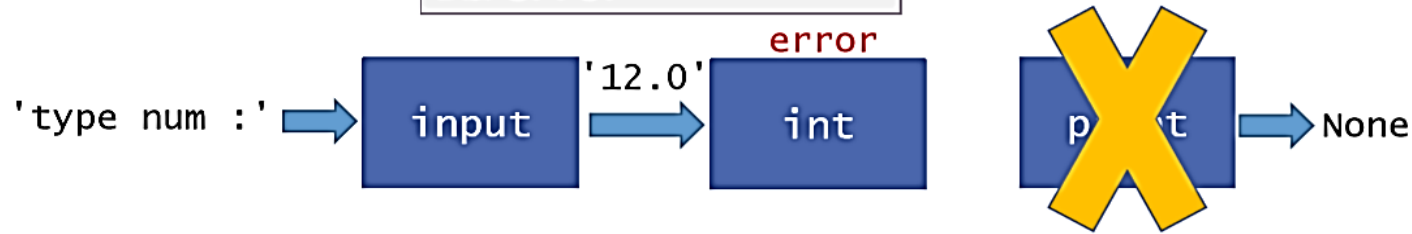
- การส่งค่าจากฟังก์ชันไปให้อีกฟังก์ชัน (ต่อ)

```
print(int(input('type num : ')))
```

```
type num : 12↵  
12
```



```
type num : 12.0↵  
เกิด error
```



▪ ฟังก์ชัน len

พารามิเตอร์ของฟังก์ชัน len มี 1 ค่า

ค่าที่ส่งเข้าไปให้ฟังก์ชัน len เป็นสตริง (ลิสต์ ทูเปิล ดิคชันนารี)

ฟังก์ชัน len ส่งคืนจำนวนเต็มที่เป็นความยาวของสตริง

เช่น

```
len('my name') คืนค่า 7
```

len(input('Enter name:')) คืนจำนวนเต็มที่เป็นความยาวของสตริงที่รับเข้ามา เช่น จะคืนค่า 9 ถ้าผู้ใช้พิมพ์ค่าดังตัวอย่างข้างล่าง

```
Enter name:John Fox ↵
```


▪ ฟังก์ชัน strip

รูปแบบการเรียกใช้ฟังก์ชัน

```
strip <สตริง1>.strip(<สตริง2>)
```

การทำงานของฟังก์ชัน strip

ลบตัวอักษรที่อยู่ใน <สตริง2> ออกจากด้านซ้ายสุดและขวาสุดของ <สตริง1>

เช่น `mthst = '+-+x+2-7-+-'.strip('+-')`

```
print(mthst)
```

แสดงผลลัพธ์เป็น `x+2-7`

▪ ฟังก์ชัน strip (ต่อ)

จากการเรียกใช้ฟังก์ชันนี้

```
<สตริง1>.strip(<สตริง2>)
```

เราสามารถละ <สตริง2> เมื่อต้องการให้ลบ whitespace
whitespace ได้แก่ blank tab newline

เช่น

```
nm = ' \t John Smith \n'.strip()  
print(nm)
```

แสดงผลลัพธ์เป็น John Smith

“

Quiz

Quiz

```
inp = '00080'  
age = inp.strip('0')
```

ค่าของตัวแปร age คืออะไร

```
inp = ' \t 23 Bat \tMan \t'.strip()
```

ค่าของตัวแปร inp คืออะไร

```
x = input('name:')  
y = x.strip('0123456789()')
```

ถ้าโปรแกรมทำงานแล้วผู้ใช้ใส่ค่า
(103) Bob Cane 0881234574<enter>
ค่าของตัวแปร y คืออะไร

▪ ฟังก์ชัน split

รูปแบบการเรียกใช้ฟังก์ชัน split

```
<สตริง1>.split(<สตริง2>)
```

การทำงานของฟังก์ชัน split

แบ่ง <สตริง1> เป็นสตริงย่อย โดยใช้ <สตริง2> เป็นตัวแบ่ง เช่น

```
mthst = '1+-+x+2-7-+-'.split('+-')
```

```
print(mthst)
```

แสดงผลลัพธ์เป็น ['1', '+x+2-7-', '']

ซึ่งเป็นลิสต์ เนื่องจากเรายังไม่ได้เรียนเรื่องลิสต์ เราจะใช้ฟังก์ชันนี้
อย่างง่ายไปก่อน

▪ ฟังก์ชัน split (ต่อ)

```
a0, a1, a2 = '1+-+x+2-7-+-'.split('+-')
```

จะได้ a0='1', a1='+x+2-7-', a2=''

```
sname = ' John\tSmith '
```

```
fn, ln = sname.split('\t')
```

จะได้ fn=' John', ln='Smith '

▪ ฟังก์ชัน split (ต่อ)

ถ้าเราละสตริงที่ใช้เป็นตัวแบ่ง หมายความว่า ใช้ whitespace เป็นตัวแบ่ง เช่น

```
sname = ' John \t Smith '
```

```
fn, ln = sname.split()
```

จะได้ fn='John', ln='Smith'

▪ ฟังก์ชัน split (ต่อ)

ตัวอย่างการใช้งาน

รับชื่อและนามสกุล (เช่น Tom Smith) จากผู้ใช้ แล้วไปเก็บในตัวแปร 2 ตัว

```
name=input('Enter first and last name')  
fn,ln=name.split()
```

เหมือนกับ

```
fn,ln=input('Enter first and last name').split()
```


“

Quiz

Quiz

```
x,y,z = '36 \t 23\t 58\t'.split()
```

ค่าของตัวแปร x, y และ z คืออะไร

```
x,y,z = '36 \t 23\t 58\t'.split('\t')
```

ค่าของตัวแปร x, y และ z คืออะไร

```
inp = input('x,y:')  
x,y = inp.split()
```

ถ้าโปรแกรมทำงานแล้วผู้ใช้ใส่ค่า 20 30<enter>
ชนิดของตัวแปร x และ y คืออะไร

```
x,y = input('x,y:').split()  
x = int(x)
```

ถ้าโปรแกรมทำงานแล้วผู้ใช้ใส่ค่า 20 30<enter>
ชนิดของตัวแปร x และ y คืออะไร

▪ ฟังก์ชันทางคณิตศาสตร์

module math มีฟังก์ชันทางคณิตศาสตร์ เช่น sin, ceil, floor, log2, pow, sqrt และ ค่าคงที่ เช่น pi, e (จาก <https://docs.python.org/3/library/math.html>)

```
import math
print(math.pi, math.e)
r = 4.1
areaC = math.pi*r**2 # area of circle
a = 3
b = 2.1
c = 0.4
areaTr = a*b*math.sin(c)/2 # area of triangle
```



End

