

ตัวแปร และ ชนิดข้อมูล (Variables & Data Types)

การกำหนดค่าให้ตัวแปร

```
>>> a = 10
แล้วเรียกใช้หละ ? แคลพิมพ์ชื่อตัวแปรลงไปไหนหละ !
>>> a
10
```

การนำข้อมูลในตัวแปรมาคำนวณ

ทำได้ในรูปแบบ (ตัวแปร ตัวดำเนินการ ตัวเลขหรือตัวแปร)

```
>>> a + 10
20
>>> a/2
5
```

โดยตัวดำเนินการก็คือ + - * / และ ** (ยกกำลัง)

การแปลงประเภทข้อมูล

โดยในฟังก์ชันเหล่านี้จะมีการรับค่า เราสามารถใส่ข้อมูล หรือ ตัวแปรลงไปไหนเพื่อแปลงได้เลย

`str()` สำหรับการแปลงเป็น String เช่น `str(a) >> "10"`

`int()` สำหรับการแปลงเป็น Integer เช่น `int(3.45) >> 3`

`float()` สำหรับการแปลงเป็น Float เช่น `float(1) >> 1.0`

`bool()` สำหรับการแปลงเป็น Boolean เช่น `bool(1) >> true`

เรื่องของสตริงที่ไม่ใช่แนวเพลง ! (Strings)

การกำหนดค่าให้ตัวแปร และ ใช้งานสตริง

```
>>> b = 'Hello'
>>> b
'Hello'
```

การใช้งานตัวดำเนินการกับสตริง

```
>> b[3]
'l'
>> b[0:2]
'He'
```

การใช้งานตัวดำเนินการกับสตริง

```
>>> b * 3
'HelloHelloHello'
>>> b + 'FromMars'
'HelloFromMars'
>>> 'e' in b
True
```

การใช้เมธอดกับสตริง

```
>> b.lower() แปลงข้อความเป็นตัวพิมพ์เล็ก
>> b.upper() แปลงข้อความเป็นตัวพิมพ์ใหญ่
>> b.count('l') นับสมาชิกในสตริง
>> b.replace('e','x') ทำการแทนที่ตัวอักษร
>> len(b) แสดงขนาดของข้อความ
```

ทำยังไงดีไม่รู้ว่าคำสั่งนี้ใช้ยังไง ? (Help me please !)

ใช้คำสั่งง่ายๆ เพื่อดูว่าไว้ใช้ทำอะไร

```
>>> help(int)
ใส่ประเภทข้อมูล ชื่อฟังก์ชัน หรือ คำสั่งต่างๆ ที่ส่งสยลงไป
ในวงเล็บ ระบบจะแสดงให้เราเลยว่าสั่งดังกล่าวมีไว้ใช้ทำอะไร !
```

การใช้งานข้อมูลประเภทลิสต์ (Lists)

```
>>> x = 'so'
>>> y = 'beautiful'
>>> list1 = ['you','are',x,y]
>>> list2 = [[10,11,12,13],[0,1,2,3]]
```

การดึงข้อมูลจากลิสต์

```
>>> list1[0]
>>> list1[-2]
>>> list1[1:3]
>>> list1[1:]
>>> list[:3]
```

เลือกข้อมูลตัวแรก
เลือกข้อมูลตัวที่ 2 นับจากสุดท้าย
เลือกข้อมูลลำดับที่ 1 ถึง 2
เลือกข้อมูลหลังจากลำดับที่ 0
เลือกข้อมูลทั้งหมดก่อนลำดับที่ 3

```
>>> list2[1][0]
>>> list[1][:2]
```

list2[ลำดับลิสต์][ลำดับข้อมูลในลิสต์]

ตัวดำเนินการของลิสต์สามารถใช้งานได้ลักษณะคล้ายกับ String สามารถ + เพื่อต่อ * เพื่อทำซ้ำ และ เปรียบเทียบ < > =

การใช้งานข้อมูลประเภทลิสต์ (Lists) ต่อ

การใช้เมธอดกับลิสต์

```
>>> list1.index(a)
>>> list1.count(a)
>>> list1.append('!!')
>>> list1.remove('!!')
>>> del(list1[0:1])
>>> list1.reverse()
>>> list1.pop()
>>> list1.insert(0, '!!')
>>> list1.sort()
```

ดึงลำดับของข้อมูลที่ใส่ไปออกมา
นับว่าข้อมูลดังกล่าวในลิสต์
เพิ่มข้อมูลลงลิสต์
ลบข้อมูลในลิสต์
ลบข้อมูลในลิสต์
เรียงลำดับข้อมูลแบบย้อนกลับ
ลบข้อมูลในลิสต์จากลำดับ
แทรกข้อมูลลงไปลิสต์จากลำดับ
เรียงลำดับข้อมูลในลิสต์

เครื่องหมายทางตรรกะ

การเปรียบเทียบจะใช้เครื่องหมาย < > <= >= (มากกว่าเท่ากับ) >= (น้อยกว่าเท่ากับ) == (เปรียบเทียบข้อมูลซ้าย-ขวามือ) != (ไม่เท่ากับ)

a and b ใช้เชื่อมค่าทางตรรกะ: **ซ้ายและขวาเป็นจริงเท่านั้นถึงจะเป็นจริง** นอกนั้นเป็นเท็จ
a or b ใช้เชื่อมค่าทางตรรกะ: **ซ้ายและขวาเป็นเท็จเท่านั้นถึงจะเป็นเท็จ** นอกนั้นเป็นจริง
not a เป็นการใส่ปฏิเสธให้กับตรรกะ

โดยข้อมูลทางตรรกะจะมีค่าเป็น True หรือ False ใดๆอย่างหนึ่ง

การตรวจสอบเงื่อนไข (Conditional Statement)

ตรวจสอบว่าเงื่อนไขดังกล่าวเป็นจริงหรือไม่ ถ้าเป็นจริงจะทำงานภายใต้ Block ของ if โดยมีโครงสร้างดังขวามือ

ถ้าหากเงื่อนไขดังกล่าวไม่เป็นจริงจะทำงานในส่วน of elif เพื่อตรวจสอบต่อไป และ สุดท้ายจะเข้าเงื่อนไข else หากไม่เป็นจริงทั้งหมด

if เงื่อนไขทางตรรกะ:
คำสั่งต่างๆ

elif เงื่อนไขที่ 2:
คำสั่งต่างๆ

else::
คำสั่งต่างๆ

การวนซ้ำจากเงื่อนไข (Conditional Loop Statement)

เหมือนกับการใช้ตรวจสอบเงื่อนไขด้านบนเลย ! ถ้าหากเป็นจริงจะทำงานภายใน Block ของ while

while เงื่อนไขทางตรรกะ:
คำสั่งต่างๆ

การวนซ้ำ (Iterative Loop Statement)

เป็นการดึงข้อมูลออกมาทีละตัวจากการวนซ้ำโดย 1 ครั้ง จะทำการดึงมา 1 ไอเทม ใส่ลงในตัวแปรตำแหน่ง var

for var in sequence:
คำสั่งต่างๆ

การรับค่าและแสดงผล

การแสดงผล
`print("Hello", 14)` สามารถใช้ลูกน้ำเชื่อมข้อความได้

การรับค่า
`a = input("ข้อความแสดงผล")` รับเสร็จเก็บไว้ในตัวแปรด้านหน้า

Version 1.0
วันที่ 8 มกราคม 2561