

ทำตามคำสั่ง

ทำตามคำสั่ง

- รับ **string** ที่เป็นตัวเลขคั่นด้วย **,** แล้วแยกใส่ **list** ชื่อ **slist**
- แปลง **slist** เป็น **ilist** คือ **list** ของตัวเลข
 ilist = [] # จะใช้ **list** ให้ประกาศว่างๆ ไว้ก่อนเสมอ
 for item in slist :
 ilist.append(int(item))
- ถ้า **ilist** มีขนาดมากกว่า **2** ให้ลบตำแหน่งที่ **2**
- ต่อท้าย **ilist** ด้วยความยาวของ **ilist**
- ทำการ **pop 5** ครั้ง โดยตรวจสอบก่อน **pop** ว่า **ilist** ไม่ว่างถึงค่อย **pop** ทุกครั้ง
- ต่อท้าย **ilist** ด้วย **Boolean** จากการตรวจสอบว่า **ilist** ว่างหรือไม่
 - ว่างต่อด้วย **True**
 - ไม่ว่างต่อด้วย **False**

ทำตามคำสั่ง (ต่อ)

- ถ้า `ilist` ไม่ว่างและตัวแรกของ `ilist` เป็นชนิด `int` ให้ทำวงวนจากค่า `1` ถึงค่า `ilist[0]` (รวมค่า `ilist[0]` ด้วย) เพื่อเพิ่มค่าที่ตำแหน่ง `0` ด้วย `1`, ตำแหน่ง `1` ด้วย `2` จนจบวงวน

ใบ้ 1 : `type(x) is int` จะให้ค่าเป็น `True` เมื่อ `x` เป็นชนิด `int` และเป็น `False` เมื่อ `x` ไม่เป็นชนิด `int`

ใบ้ 2 :

for `i` in `range(x,y)` :

`ilist.insert(i-1,z)` # `x,y,z` คือสิ่งที่ต้องแก้

เช่น `ilist = [3,4,5,False]` ตัวแรกของ `ilist` คือ `3` ในส่วนนี้จะทำการเพิ่ม `1,2,3` ไปข้างหน้า ได้ผลเป็น `ilist = [1,2,3,3,4,5,False]`

ทำตามคำสั่ง (ต่อ)

- ต่อท้าย **ilist** ด้วยการนับจำนวน **0** ใน **ilist**
- ต่อท้าย **ilist** ด้วย **0**
- ต่อท้าย **ilist** ด้วยตำแหน่ง **0** แรกที่หาเจอ
- เอา **0** ออกจาก **ilist**
- พิมพ์ **ilist**

ทำตามคำสั่ง (ต่อ)

```
str = input()
slist = str.split(',')
ilist = []

for item in slist:
    ilist.append(int(item))

if ??? > 2 :
    ??? ilist[???]

ilist.append(len(???))

for i in range(???):
    if (len(ilist) > 0) :
        ilist.???()
```

```
if len(ilist) == ??? :
    ilist.append(???)
else :
    ilist.append(???)

if len(ilist) > 0 and type(ilist[0]) is int :
    for i in range(???, ???) :
        ilist.insert(i-1, ???)

ilist.append(ilist.???())
ilist.append(0)
ilist.append(ilist.???())
ilist.???()
print(ilist)
```

List

- สร้างด้วย [] คั่นด้วย ,
- ไม่จำเป็นต้องเป็น **data type** แบบเดียวกัน
- [] คือ **list** ว่าง

```
>>> [1,3,True,6.5]
[1, 3, True, 6.5]
>>> myList = [1,3,True,6.5]
>>> myList
[1, 3, True, 6.5]
```

- เริ่มต้น **list** ด้วยค่าเดียวกัน

```
>>> myList = [0] * 6
>>> myList
[0, 0, 0, 0, 0, 0]
```

List's operators

Operation Name	Operator	Explanation
indexing	[]	ดึงค่าจากตำแหน่งที่กำหนดใน []
concatenation	+	รวมลำดับ
repetition	*	รวมลำดับแบบทำซ้ำเป็นจำนวนครั้ง
membership	in	ถามว่ามี item ใน list หรือไม่
length	len	จำนวน item ใน list
slicing	[:]	ดึงลำดับมาบางส่วน

Run

Show CodeLens

```
1 myList = [1,2,3,4]
2 A = [myList]*3
3 print(A)
4 myList[2]=45
5 print(A)
6
7
```

[[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
[[1, 2, 45, 4], [1, 2, 45, 4], [1, 2, 45, 4]]

List's methods

Method Name	Use	Explanation
append	<code>alist.append(item)</code>	เพิ่ม item ท้าย list
insert	<code>alist.insert(i,item)</code>	เพิ่ม item ที่ตำแหน่ง i
pop	<code>alist.pop()</code>	คืนค่า item ที่ตำแหน่งสุดท้ายแล้วเอาออกด้วย
pop	<code>alist.pop(i)</code>	คืนค่า item ที่ตำแหน่ง i แล้วเอาออกด้วย
sort	<code>alist.sort()</code>	เรียงลำดับใน list
reverse	<code>alist.reverse()</code>	ย้อนลำดับใน list
del	<code>del alist[i]</code>	ลบ item ที่ตำแหน่ง i
index	<code>alist.index(item)</code>	ค้นหา item แล้วคืนค่าตำแหน่งที่เจอมาให้
count	<code>alist.count(item)</code>	นับจำนวนที่มีค่าเดียวกับ item ใน list
remove	<code>alist.remove(item)</code>	ลบตัวที่มีค่าเดียวกับ item ตัวแรกที่เจอ

ตัวอย่าง

Run

Show CodeLens

```
1 myList = [1024, 3, True, 6.5]
2 myList.append(False)
3 print(myList)
4 myList.insert(2, 4.5)
5 print(myList)
6 print(myList.pop())
7 print(myList)
8 print(myList.pop(1))
9 print(myList)
10 myList.pop(2)
11 print(myList)
12 myList.sort()
13 print(myList)
14 myList.reverse()
15 print(myList)
16 print(myList.count(6.5))
17 print(myList.index(4.5))
18 myList.remove(6.5)
19 print(myList)
20 del myList[0]
21 print(myList)
22
```

```
[1024, 3, True, 6.5, False]
[1024, 3, 4.5, True, 6.5, False]
False
[1024, 3, 4.5, True, 6.5]
3
[1024, 4.5, True, 6.5]
[1024, 4.5, 6.5]
[4.5, 6.5, 1024]
[1024, 6.5, 4.5]
1
2
[1024, 4.5]
[4.5]
```

range

- เป็น **function** ที่ใช้สร้างลำดับของตัวเลข
- ใช้บ่อยใน **for loop**
- การทำงานต่างกันตามจำนวน **parameter**
 - **range(n)** สร้าง 0 ถึง n-1
 - **range(n, m)** สร้าง n ถึง m-1
 - **range(n, m, o)** สร้าง n ด้วย step o และไม่เกิน m

```
>>> range(10)
range(0, 10)
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(5,10)
range(5, 10)
>>> list(range(5,10))
[5, 6, 7, 8, 9]
>>> list(range(5,10,2))
[5, 7, 9]
>>> list(range(10,1,-1))
[10, 9, 8, 7, 6, 5, 4, 3, 2]
>>>
```