

List

Outline

- List
- Elements in lists
- for loop and lists
- Working with lists
- List slicing
- List comprehension
- String

List

ลักษณะของ List

- List หรือ รายการ เป็นชนิดของข้อมูลชนิดหนึ่งซึ่งประกอบด้วยสมาชิกเรียงกันเป็นลำดับ
- สมาชิกแต่ละตัวใน List เป็นชนิดใดก็ได้และไม่จำเป็นต้องเป็นชนิดเดียวกัน
- ขนาดของ List เพิ่มลดได้อัตโนมัติตามจำนวนสมาชิก

การเขียน List ใน Python

- เขียนสมาชิกภายใน วงเล็บสี่เหลี่ยม ([]) สมาชิกแต่ละตัวคั่นด้วยจุลภาค (,)
- [1, 'a', "comp prog", [True, False], 2.56]
- [] หมายถึงลิสต์ว่าง

ขนาดของ List

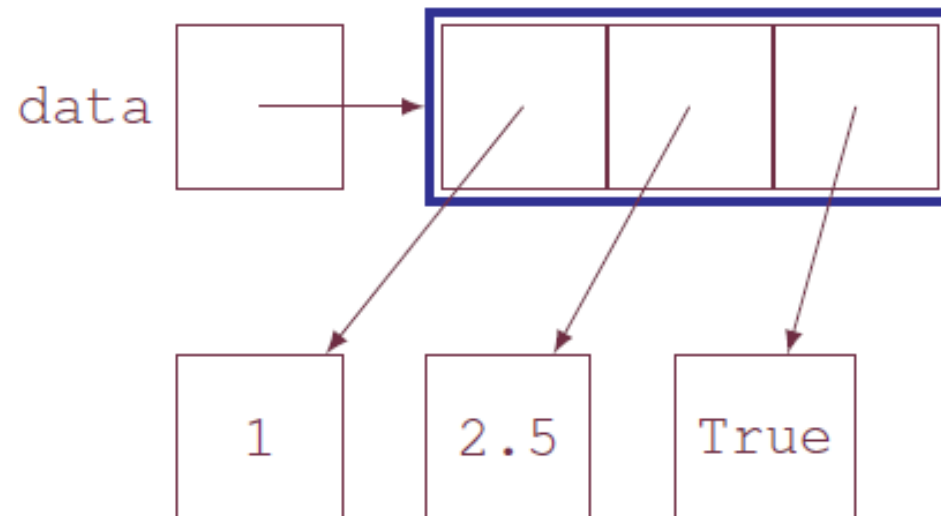
- ขนาดของ List คือจำนวนสมาชิกภายใน List
- ใช้ฟังก์ชัน **len** ในการหาขนาด
- **len**([1, 'a', "comp prog", [True, False]])
 - มีค่า 4
- กำหนด

`data = [2, 'a', 'True', -5, False]`

- **len**(data) จะมีค่า 5

การสร้าง List

```
data = [1, 2.5, True]
```



รูป: การสร้าง List

Elements in lists

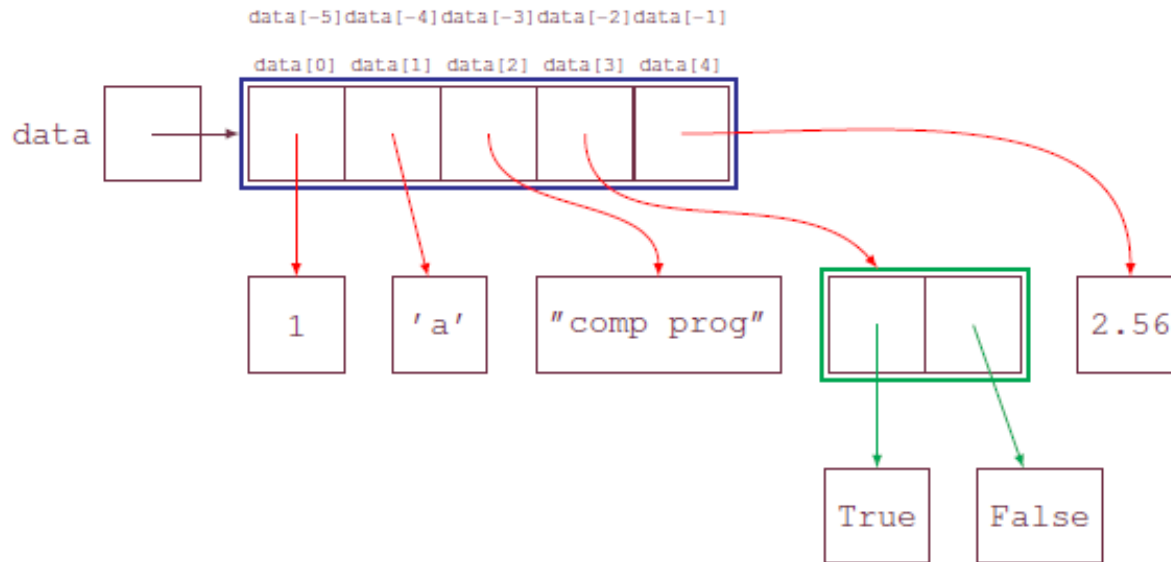
การเก็บข้อมูลใน List และดัชนี

- ลักษณะการเก็บข้อมูลใน List จะเป็นช่อง ๆ มีดัชนีเป็นเลขจำนวนเต็มอ้างอิงช่องเหล่านั้น
- สำหรับ Python
 - ค่าดัชนีช่องแรกเป็น 0
 - นับดัชนีย้อนจากท้าย List ได้โดยช่องสุดท้ายมีดัชนีเป็น -1

การเก็บข้อมูลใน List และดัชนี

กำหนด

```
data = [1, 'a', "comp prog", [True, False], 2.56]
```



รูป: ตัวอย่างการเก็บข้อมูลของ List

การเข้าถึงสมาชิกใน List โดยใช้ดัชนี

กำหนด

```
data = [1, 'a', "comp prog", [True, False], 2.56]
```

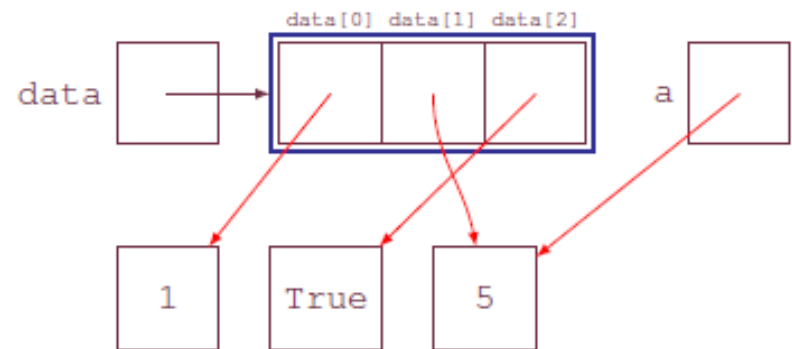
การเข้าถึงสมาชิกใน List ใช้ชื่อของ List (ชื่อตัวแปร) ร่วมกับเลขดัชนีในวงเล็บสี่เหลี่ยม

- data[0] ได้ค่าจำนวนเต็ม 1
- data[1] ได้ค่าสตริง 'a'
- data[2] ได้ค่าสตริง "comp prog"
- data[3] ได้ค่า List ซึ่งมีสมาชิกเป็น True และ False
- data[4] ได้ค่าจำนวนจริง 2.56
- data[-1] ได้ค่าจำนวนจริง 2.56 เช่นเดียวกับ data[4]

การเก็บข้อมูลใน List และการกำหนดค่าโดยใช้ดัชนี

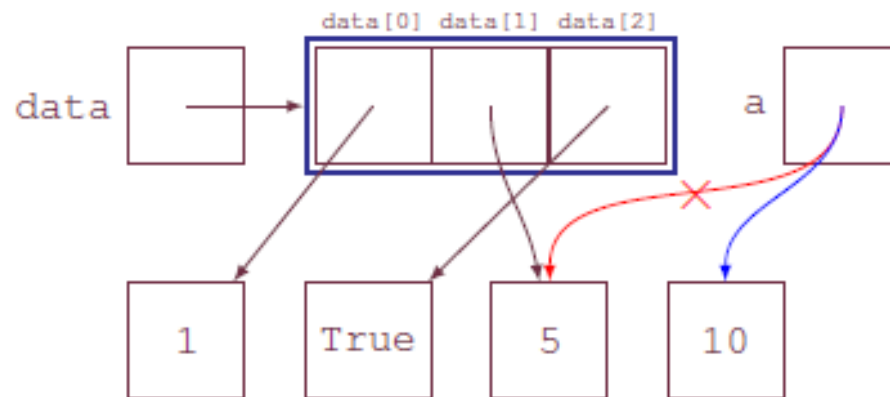
- สมาชิกแต่ละตัวใน List เป็นเสมือนตัวแปรหนึ่งซึ่งเชื่อม (bind) กับค่าจริง ๆ ที่อยู่ภายนอก
- การกำหนดค่าให้สมาชิกใน List จึงเป็นการเชื่อมสมาชิกใน List กับค่าใหม่
 - ไม่มีการเปลี่ยนแปลงค่าเดิมที่อยู่ภายนอก
เช่น กำหนดค่าเริ่มต้น

```
1 a = 5
2 data = [1, a, True]
```



การเปลี่ยนแปลงค่าในตัวแปร

```
1 a = 5
2 data = [1, a, True]
3 a = 10 #กำหนดค่าของตัวแปร a ใหม่ให้เชื่อมกับ 1
```

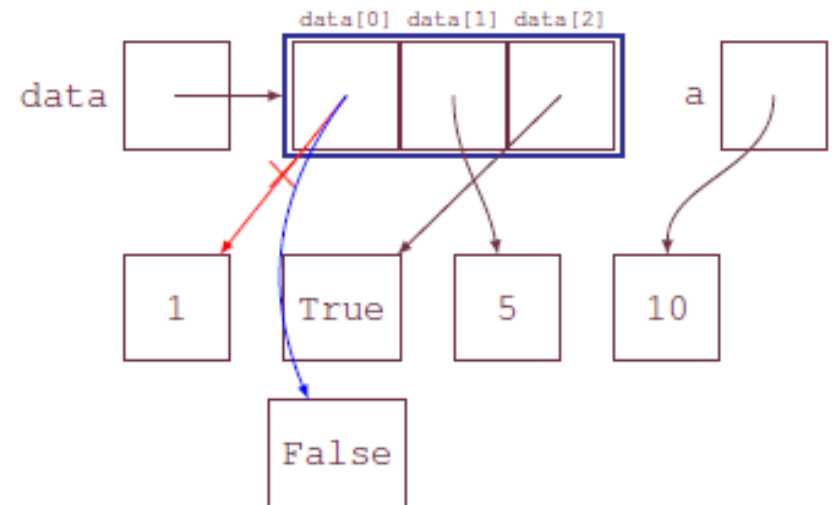


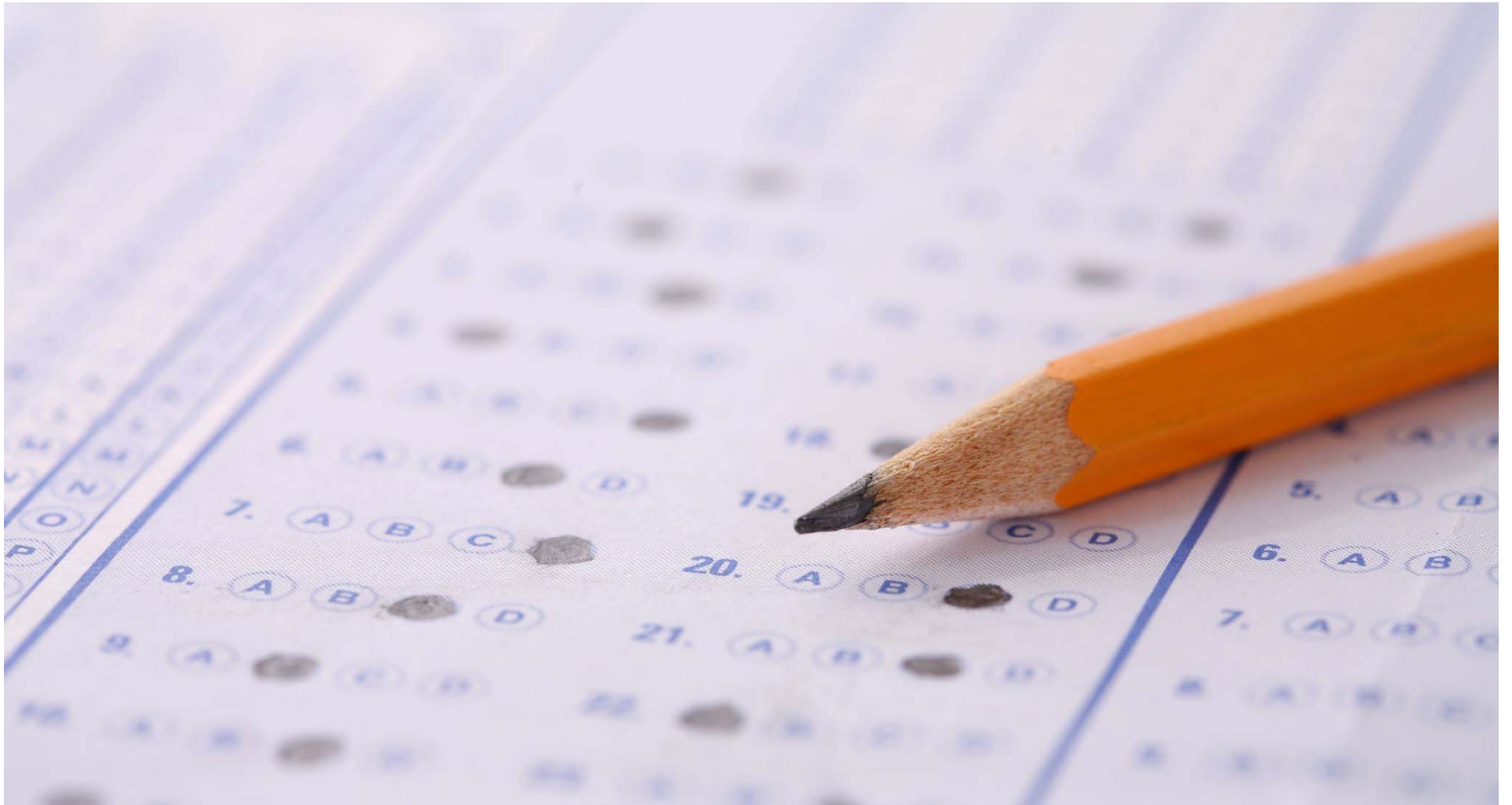
ค่าของ `data[1]` เชื่อมกับ 5 ไม่เปลี่ยนแปลง

การเปลี่ยนแปลงค่าใน List

```
1 a = 5
2 data = [1, a, True]
3 a = 10
4 data[0] = not data[2]
```

- หาผลลัพธ์ของนิพจน์ **not data[2]**
 - เนื่องจาก data[2] เชื่อมกับค่า True
 - ผลลัพธ์ของนิพจน์จึงเป็น False
- กำหนดให้ data[0] เชื่อมกับผลลัพธ์(False)





Quiz

Quiz

1. โค้ดด้านล่างให้ผลลัพธ์อะไร

```
numbers = [2,4,6,8]
```

```
numbers[2] = numbers[0]*2
```

```
numbers[-1] = numbers[-1]/2
```

```
print(numbers)
```

2. จงเขียนโค้ดเพื่อแสดง element ตัวแรกสุด (2 แบบ) และตัวท้ายสุด (2 แบบ) ใน list ชื่อ x

for loop and lists

- for in list
- for in range

การเข้าถึงข้อมูลใน List โดยใช้ iterator

- iterator คือตัวแปรที่อ้างถึงสมาชิกใน List ไปทีละตัว
- ใน Python ใช้ **for** variable **in** [...] ในการแจงนับสมาชิกใน List

เช่น

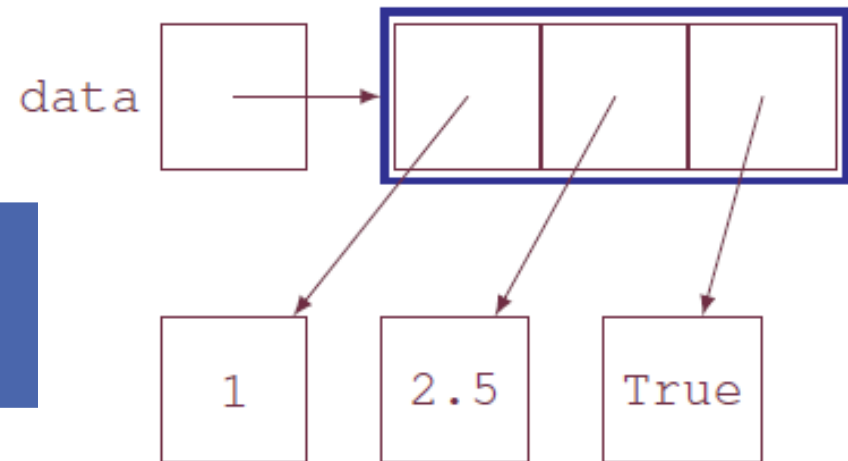
```
1 data = [1, 2.5, True]
2 for item in data:
3     ...
```

- อ่านได้ว่า สำหรับ item แต่ละตัวซึ่งอยู่ใน data
- ในแต่ละรอบของการทำงาน item จะเชื่อมกับค่าของสมาชิกแต่ละตัวใน data ซึ่งได้แก่
 - 1
 - 2.5
 - True

ตัวแปรของ List กับการกำหนดค่า

กำหนด

```
1 data = [1, 2.5, True]
2 for item in data:
3     item = 0
```



รูป: ค่าเริ่มต้นใน data

- อ่านได้ว่า สำหรับ item แต่ละตัวซึ่งเชื่อมกับค่าสมาชิกใน data ให้เชื่อมค่า item กับ 0
- เนื่องจาก item นั้นเป็นตัวแปรหนึ่งซึ่งรับค่าจากตัวแปร
◦ การเชื่อมค่าให้ item ใหม่จะ**ไม่**เป็นการแก้ไขค่าสมาชิกใน List

ตัวแปรของ List กับการกำหนดค่า

```
1 data = [1, 2.5, True]
2 for item in data:
3     item = 0
```

รอบที่ 1

1. item เชื่อมค่าเดียวกันกับ

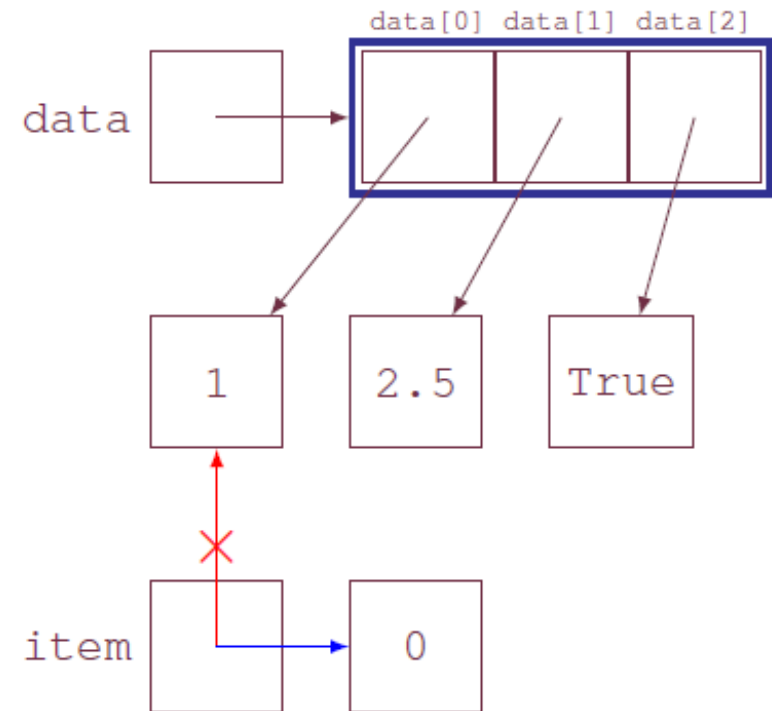
data[0]

2. กำหนดใหม่ให้ item เชื่อมกับ 0

ซึ่งไม่เกี่ยวกับการเชื่อมค่าของ data[0]

นั่นคือการเชื่อมค่าของ data[0]

ไม่มีการเปลี่ยนแปลง



รูป: รอบที่ 1

ตัวแปรของ List กับการกำหนดค่า

```
1 data = [1, 2.5, True]
2 for item in data:
3     item = 0
```

รอบที่ 2

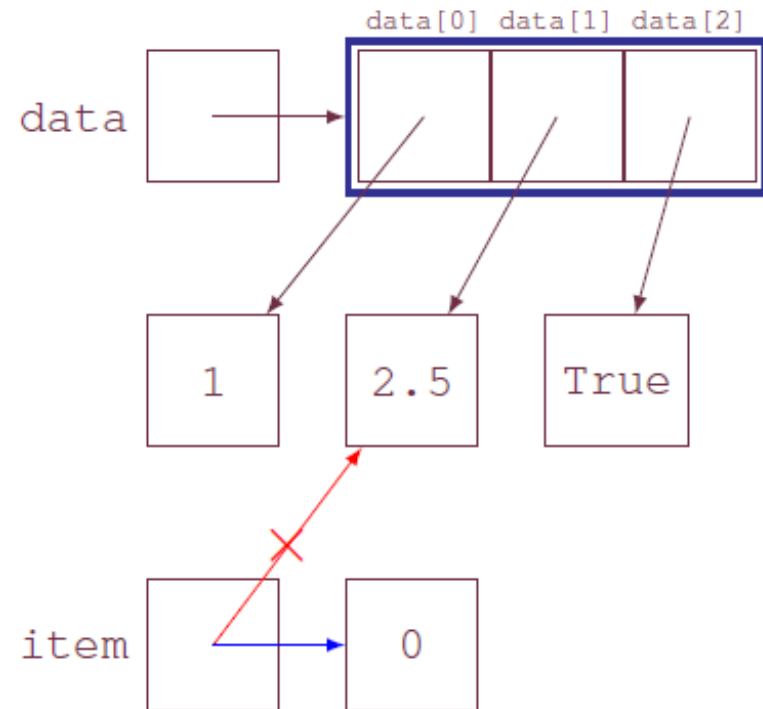
1. item เชื่อมค่าเดียวกันกับ

data[1]

2. กำหนดใหม่ให้ item เชื่อมกับ 0

การเชื่อมค่าของ data[1]

ไม่มีการเปลี่ยนแปลง



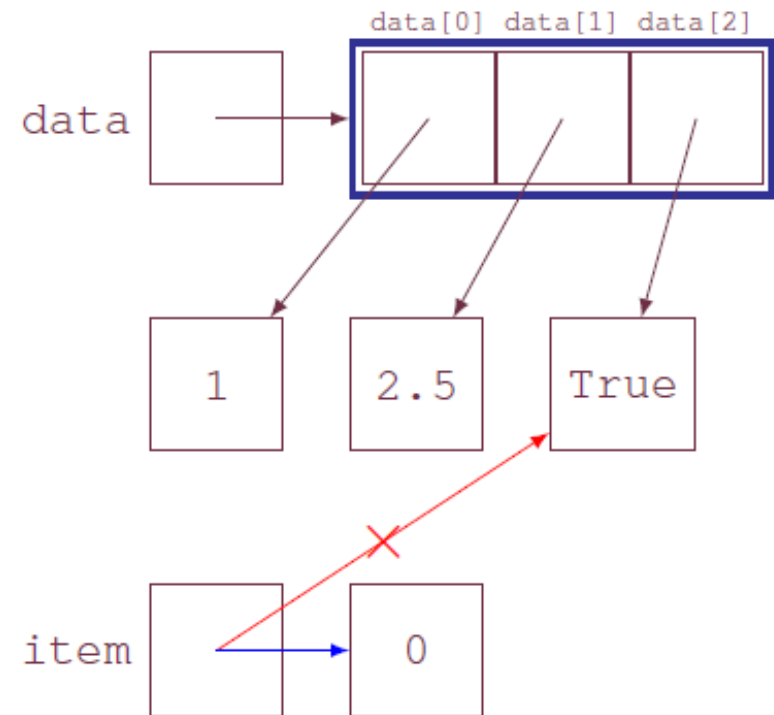
รูป: รอบที่ 2

ตัวแปรของ List กับการกำหนดค่า

```
1 data = [1, 2.5, True]
2 for item in data:
3     item = 0
```

รอบที่ 3

1. item เชื่อมค่าเดียวกันกับ data[2]
 2. กำหนดใหม่ให้ item เชื่อมกับ 0
- การเชื่อมค่าของ data[2]
ไม่มีการเปลี่ยนแปลง



รูป: รอบที่ 3

List และ range

- เนื่องจากฟังก์ชัน `range` เป็นการแจกแจงนับจำนวนเต็มทีละตัว
- เราอาจใช้ `range` ในการแจกแจงนับดัชนีของ List ได้
 - ไม่กำหนดค่าเริ่มต้นของ `range` จะได้ค่าเริ่มต้นเป็น 0 โดยปริยาย
 - กำหนดขอบปลายเป็นขนาดของ List

```
data = [2, 'a', 'True', -5, False]
for index in range(len(data)):
    print(data[index])
```

- `index` จะมีค่าตั้งแต่ 0 ถึง `len(data)-1` ซึ่งเป็นดัชนีแต่ละตัวของ สมาชิกใน List
- ส่วนของโปรแกรมนี้จะพิมพ์ค่าสมาชิกแต่ละตัวใน List จนนครบ

เปรียบเทียบ for in list กับ for in range

หากต้องการเข้าถึงข้อมูลในลิสต์
เราสามารถใช่ for ได้ทั้งสองรูปแบบ

```
1 data = [1, 2.5, True]
2 for item in data:
3     print(type(item))
```

```
1 data = [1, 2.5, True]
2 for index in
3     range(len(data)):
4         print(type(data[index]))
```

ผลลัพธ์การรันของทั้งสองโปรแกรม

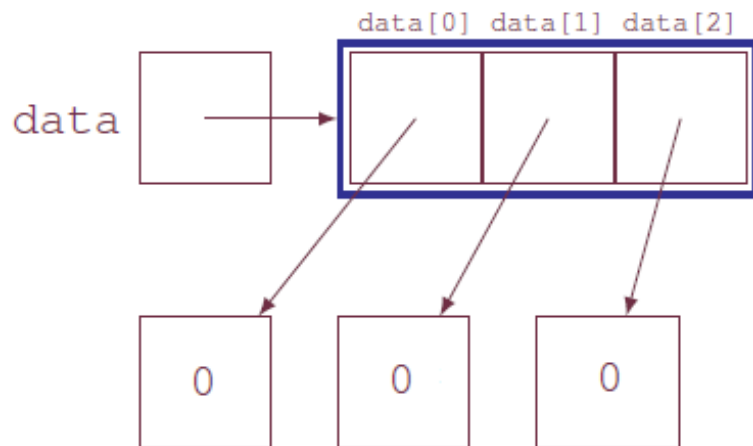
```
<class 'int' >
<class 'float' >
<class 'bool' >
```

- โปรแกรมด้านบนใช้รูปแบบ for in list แล้วแสดงชนิดของแต่ละตัวที่เก็บอยู่ในลิสต์ ดังนั้นสามารถเรียก type(item) ได้เลย
- โปรแกรมด้านล่างใช้รูปแบบ for in range และให้ผลลัพธ์เหมือนโปรแกรมด้านบน แต่ต้องเรียก type(data[index])

การแก้ไขค่าในลิสต์

หากเราต้องการแก้ไขค่าแต่ละตัวในลิสต์ เราสามารถใช้รูปแบบ for in range เพื่อแก้ไขค่าของแต่ละตัวแปรในลิสต์ได้

```
1 data = [1, 2.5, True]
2 for index in
3 range(len(data)):
   data[index] = 0
```



รูป: หลังลูปเสร็จสิ้น

- ในรอบที่ 1 index มีค่า 0
ดังนั้น $\text{data}[0] = 0$ จะ
เปลี่ยนการเชื่อมค่าของ
 $\text{data}[0]$ ให้ไปเชื่อมกับ 0
- ในรอบที่ 2 index มีค่า 1
ดังนั้น $\text{data}[1] = 0$ จะ
เปลี่ยนการเชื่อมค่าของ
 $\text{data}[1]$ ให้ไปเชื่อมกับ 0
- ในรอบที่ 3 index มีค่า 2
ดังนั้น $\text{data}[2] = 0$ จะ
เปลี่ยนการเชื่อมค่าของ
 $\text{data}[2]$ ให้ไปเชื่อมกับ 0

การกำหนดค่าลงในลิสต์

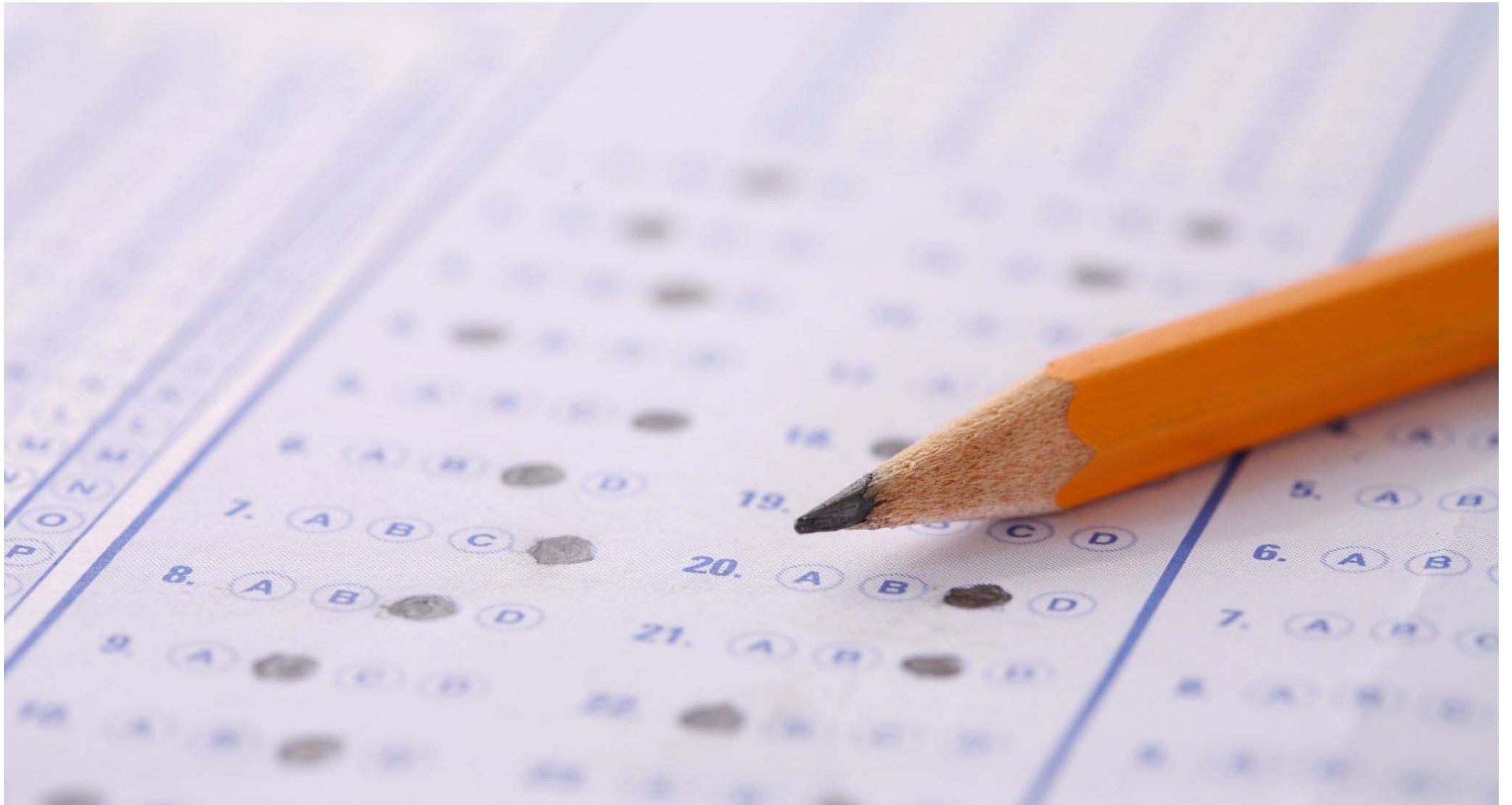
เราสามารถรับค่าจากผู้ใช้แล้วเก็บลงในลิสต์ได้

ในตัวอย่างนี้มีนิสิต 5 คน และวนรับคะแนนสอบ 5 ครั้งเพื่อเก็บลงในลิสต์

```
1 score = [0, 0, 0, 0, 0]
2 for index in range(len(score)):
3     score[index] = int(input("Enter a score: "))
4 print(score)
```

ตัวอย่างผลลัพธ์การรัน

```
Enter a score: 9
Enter a score: 7
Enter a score: 8
Enter a score: 10
Enter a score: 8
[9, 7, 8, 10, 8]
```



Quiz

Quiz

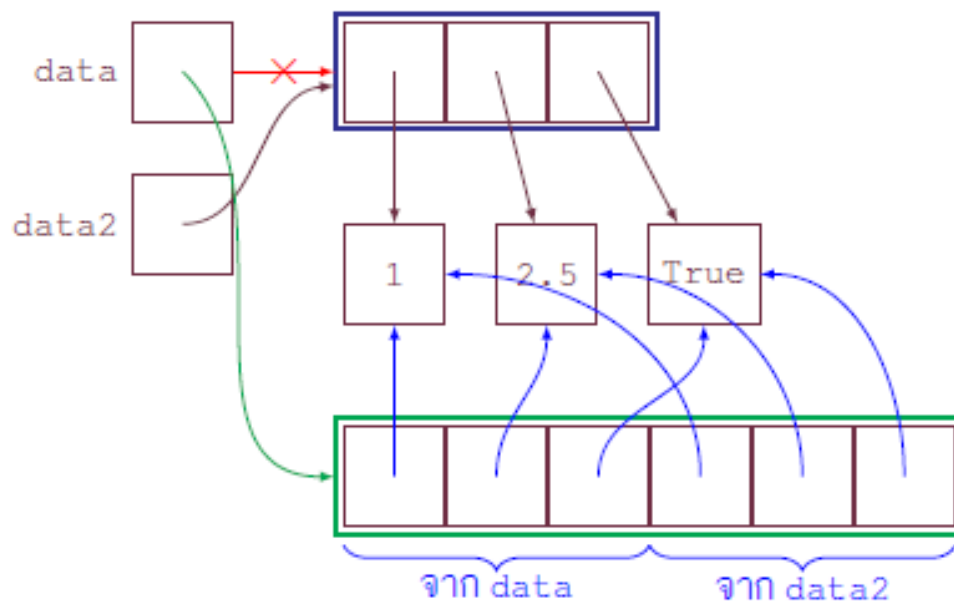
1. จงเขียนฟังก์ชันที่รับลิสต์ของตัวเลขแล้วคืนค่าเฉลี่ยโดยให้เขียนสองแบบใช้ `for in list` และ `for in range`
2. จงเขียนฟังก์ชันที่รับ list ของ integer **ซึ่งถูกเรียงลำดับจากน้อยไปมากแล้ว** แล้วคืนค่า mode ใน list กำหนดให้ list นั้นมี mode เพียงค่าเดียว

Working with lists

- Combining two lists
- The repetition operator
- Adding an element at the end of a list (append)
- Inserting an element at the specified position (insert)
- Copying a list
- Checking if a value is in a list
- Locate the position of a value in a list (index)
- Delete an element in a list
- Sorting a list
- Reversing a list

การรวมสองลิสต์เข้าด้วยกัน

```
data = data + data2
```



- สร้าง List ใหม่ซึ่งเกิดจากการนำสมาชิกใน List เดิมมาเรียงต่อกัน

รูป: ตัวดำเนินการ + กับ List

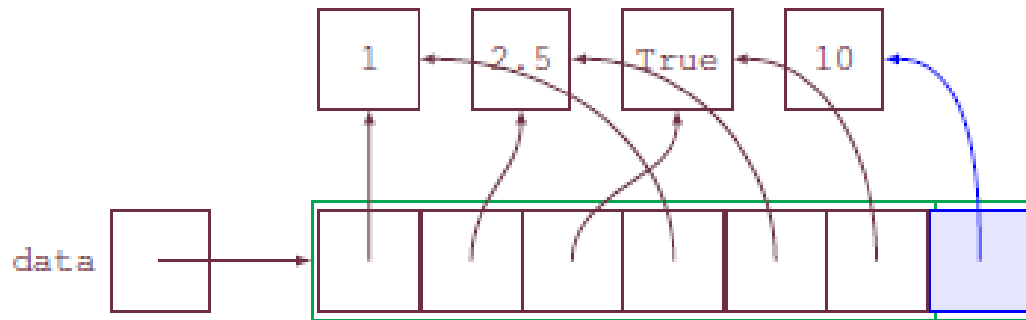
เครื่องหมาย * กับ List

- รูปแบบ: list * n โดย n คือ integer
- * จะสร้าง copies ของ list ขึ้นมาตามจำนวน n แล้วนำมาต่อกัน
- เช่น [0] * 3 จะได้ [0,0,0] โดยแรกสุดจะได้ copies ของ [0] 3 อัน => [0] [0] [0] จากนั้น [0] ทั้ง 3 list จะถูกเชื่อมกันได้เป็น [0,0,0]

```
>>numbers = [1, 3, 5]
>>numbers*3
[1, 3, 5, 1, 3, 5, 1, 3, 5]
```

การเพิ่มสมาชิกลงในลิสต์ที่ตำแหน่งสุดท้าย

```
data = [1, 2.5, True, 1, 2.5, True]
data.append(10)
```



รูป: ฟังก์ชันหลัง `append` เสร็จแล้ว

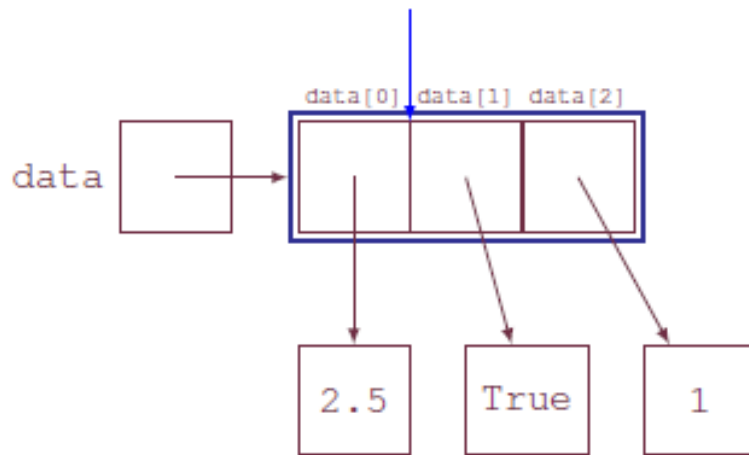
- เพิ่มสมาชิกลงใน List เดิม
- ข้อควรระวัง: `data.append([1,2,3])` จะได้
`[1,2.5,True,1,2.5,True,[1,2,3]]`
 - มีสมาชิกเพิ่มมา 1 ตัว เป็น List

การเพิ่มสมาชิกลงในลิสต์ตามตำแหน่งที่ระบุ

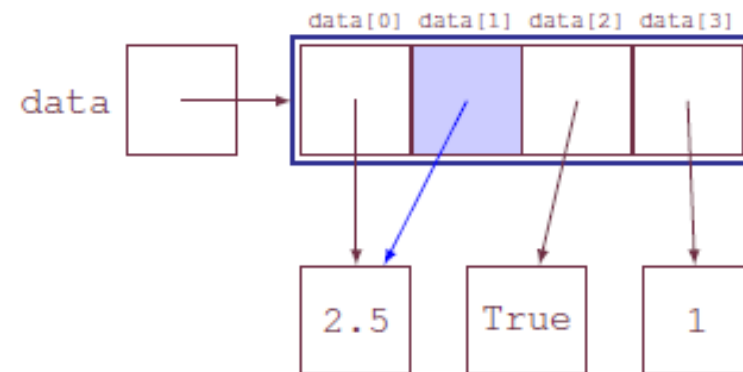
เพิ่มสมาชิกใหม่ ณ ตำแหน่งที่กำหนด

ฟังก์ชัน insert: insert(<index>, <item>)

```
data = [2.5, True, 1]  
data.insert(1, 2.5)
```



รูป: List ก่อนแทรกสมาชิก

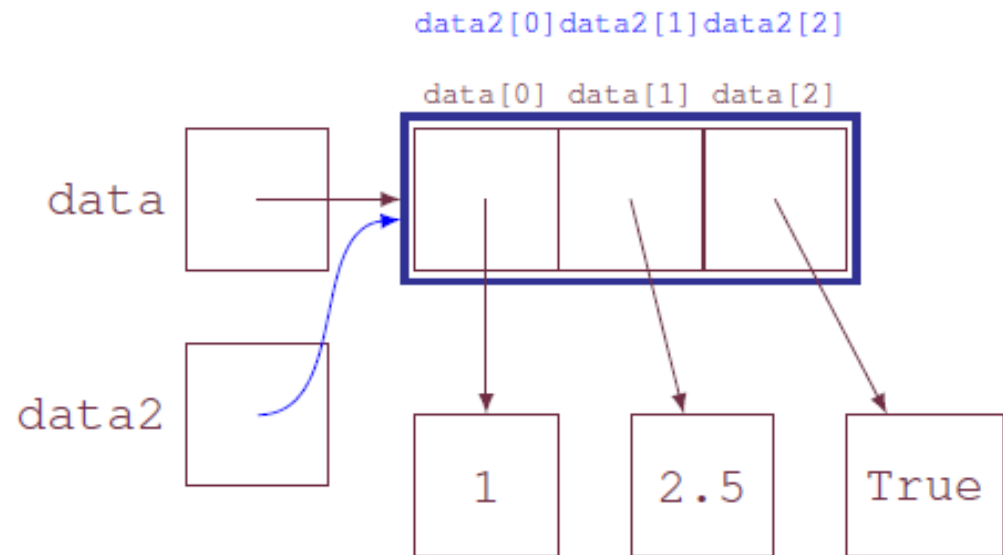


รูป: List หลังแทรกสมาชิก

Shallow Copy

```
data2 = data
```

- เป็นการเชื่อม (bind) ค่าให้ตัวแปร List ทางซ้ายของเครื่องหมาย = เชื่อมไปยังตำแหน่งเดียวกับตัวแปร List ทางขวาของเครื่องหมาย =
 - เป็น shallow copy
 - หากเปลี่ยนค่าผ่าน data2 จะส่งผลต่อ data และเปลี่ยน data จะส่งผลกับ data2
- เช่น `data2[0] = 0` เมื่อสั่ง `print(data[0])` จะได้ 0 ไม่ใช่ 1



Copying a list

หากต้องการ copy list ให้เป็นสองลิสต์ที่แยกจากกันแต่มีค่าเหมือนกัน สามารถทำได้หลายวิธี

วิธีที่ 1 ใช้การ combine กับ empty list

```
data2 = [] + data
```

วิธีที่ 2 ใช้การวนลูป copy ค่าทีละตัว

```
data2 = []  
for item in data:  
    data2.append(item)
```

วิธีที่ 3 ใช้ slicing ซึ่งจะเรียนในหัวข้อถัดไป

```
data2 = data[:]
```

วิธีที่ 4 ใช้ฟังก์ชัน list

```
data2 = list(data)
```

การตรวจสอบว่าค่าอยู่ในลิสต์

- สามารถตรวจสอบว่าค่าปรากฏอยู่ในลิสต์หรือไม่โดยรูปแบบ
value in list
- และตรวจสอบว่าค่าไม่ปรากฏอยู่ในลิสต์โดยรูปแบบ
value not in list

```
>>names = ["Manee", "Mana", "Pi ti "]  
>>"Mana" in names  
True  
>>"Chooj ai " in names  
False  
>>"Manoj " not in names  
True  
>>"Pi ti " not in names  
False
```

การหาตำแหน่งของ element ในลิสต์

- หากต้องการหาตำแหน่งของ element ในลิสต์ สามารถใช้ index ในรูปแบบ `list.index(item)`
- จะคืนตำแหน่งของ element แรกที่เจอในลิสต์
- หากไม่เจอ element นั้นจะเกิด `ValueError` exception

```
>>names = ["Manee", "Mana", "Pi ti ", "Manee"]  
>>names.index("Manee")  
0  
>>names.index("Mana")  
1  
>>names.index("Manatee")  
ValueError: 'Manatee' is not in list
```

การลบสมาชิกในลิสต์โดยตำแหน่ง

ใช้คำสั่ง **del** เหมือนกับการลบตัวแปรปกติ

การลบเป็นการลบชื่อตัวแปรเท่านั้น ตัวแปรอื่นที่ผูกอยู่กับค่าเดียวกัน จะไม่มีการเปลี่ยนแปลง เช่น

```
>>x = 10
>>print(x)
10
>>y = x
>>del x
>>print(x)
NameError: name 'x' is not defined
>>print(y)
10
```

การลบสมาชิกในลิสต์โดยตำแหน่ง

หากใช้คำสั่ง **del** กับตัวแปรชนิด List และระบุดัชนีจะเป็นการลบเฉพาะดัชนีที่กำหนด

```
>> data = [2.5, True, 1]
>> data2 = data
>> del data[1]
>> data
[2.5, 1]
>> data2
[2.5, 1]
>> del data
>> data
NameError: name 'data' is not defined
>> data2
[2.5, 1]
```

การลบสมาชิกแบบระบุค่า

- สามารถลบ item ออกจากลิสต์โดยรูปแบบ `list.remove(item)`
- Element แรกที่มีค่าเท่ากับ item จะถูกลบออก
- ขนาดของลิสต์หลังถูกลบจะลดลงไปหนึ่ง และ ดัชนีของ element หลังตำแหน่งที่ถูกลบจะลดลงไปหนึ่ง
- หาก item ที่ระบุไม่มีอยู่ในลิสต์จะเกิด `ValueError` exception

```
>>names
['Manee' , 'Mana' , 'Pi ti' , 'Manee' ]
>>names.remove('Manee' )
>>names
['Mana' , 'Pi ti' , 'Manee' ]
>>names.remove("Chooj ai ")
ValueError: list.remove(x): x not in list
```

การเรียงลำดับข้อมูลใน List

- ใช้ sort
- ผลลัพธ์ที่ได้อยู่ใน List เดิม
- ไม่มีการคืนค่า
- สำหรับ Python 3 ขึ้นไป
 - เรียงข้อมูลซึ่งมีชนิดเดียวกันเท่านั้น
 - จำนวนเต็มและจำนวนจริงสามารถเปรียบเทียบกันได้
 - กรณีที่ต้องการเปรียบเทียบข้อมูลข้ามชนิดกัน ต้องกำหนดฟังก์ชัน key ซึ่งเป็นตัวแปลงลำดับเพื่อให้เปรียบเทียบลำดับกันได้
 - ค่าโดยปริยายคือเรียงลำดับจากน้อยไปมาก หากต้องการเรียงจากมากไปน้อย ให้กำหนด option reverse=True

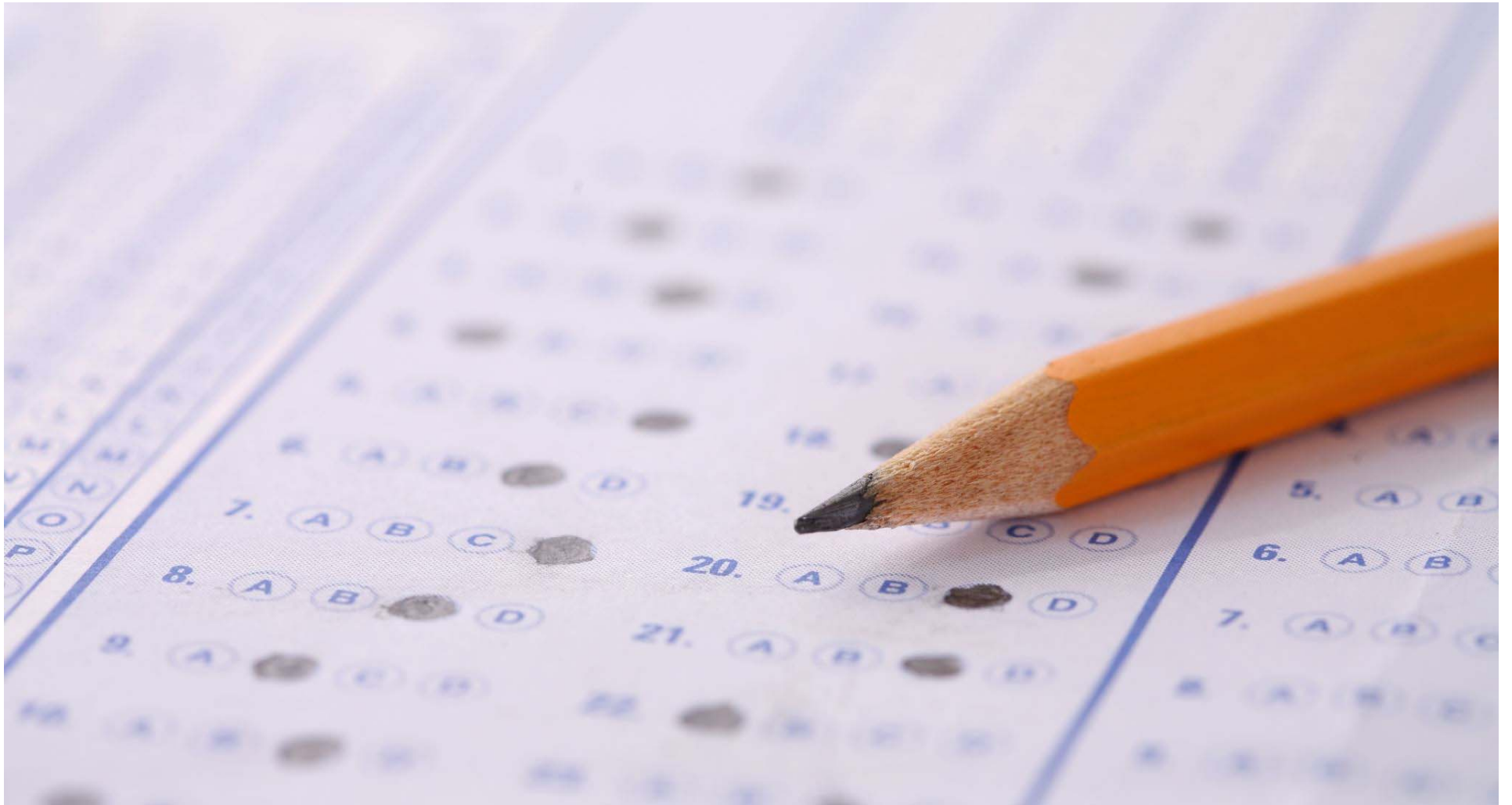
การเรียงลำดับข้อมูลใน List

```
>> data1 = [2, 10, 0.1, -5.3, 4]
>> data1.sort()
>> data1
[-5.3, 0.1, 2, 4, 10]
>> data1.sort(reverse=True)
>> data1
[10, 4, 2, 0.1, -5.3]
>> data2 = ['a', 'A', '01234', '2301170', 'Comp
Prog']
>> data2.sort()
>> data2
['01234', '2301170', 'A', 'Comp Prog', 'a']
```

การกลับลำดับการเรียงข้อมูลใน List

- ใช้ reverse
- ผลลัพธ์ที่ได้อยู่ใน List เดิม
- ไม่มีการคืนค่า

```
>> data1 = [2, 10, 0.1, -5.3, 4]
>> data1.reverse()
>> data1
[4, -5.3, 0.1, 10, 2]
>> data2 =
['a', 'A', '01234', '2301170', 'Comp Prog']
>> data2.reverse()
>> data2
['Comp Prog', '2301170', '01234', 'A', 'a']
```



Quiz

Quiz

1. โค้ดด้านล่างให้ผลลัพธ์อะไร

```
mylist = []  
mylist.append(5)  
mylist.insert(7,10)  
mylist.insert(0,3)  
mylist=mylist*2  
mylist = mylist+[2,4]  
mylist.remove(10)  
del mylist[0]  
mylist.sort()  
print(mylist)
```

2. เขียนโปรแกรมที่อ่านค่าจากไฟล์เก็บลง list โดยไฟล์เก็บชื่อนิสิต บรรทัดละหนึ่งคน เช่น

Mana

Piti

Manee

จากนั้นรับชื่อจากผู้ใช้แล้วแสดงข้อความว่า Student is in class หากชื่อที่ผู้ใช้พิมพ์ อยู่ใน list และแสดงข้อความว่า Student is NOT in class หากชื่อที่ผู้ใช้พิมพ์ไม่อยู่ใน list

List Slicing

slicing

- Slicing คือการสร้าง List ขึ้นมาใหม่เป็น List ย่อยจาก List เดิม โดยกำหนดช่วงของดัชนีที่ต้องการ
- `data[start:end]`
 - ดัชนีที่ได้คือจำนวนเต็มทุกตัวในช่วง `[start,end)`
- `data[start:end:step]`
 - ช่วงของดัชนีที่ได้คือ `[start,end)` ระยะห่างระหว่างดัชนีที่ได้คือ `step`
 - ทำนองเดียวกับฟังก์ชัน **range**
- ถ้าต้องการเริ่มจากดัชนี 0 สามารถเว้นไม่ใส่ค่า `start` ได้
- ถ้าต้องการช่วงถึงดัชนีสุดท้าย สามารถเว้นไม่ใส่ค่า `end` ได้

```
>>data = [2, 'a', 'True', -5, False]
```

```
>>data[0:2]
```

```
[2, 'a']
```

```
>>data[1:]
```

```
['a', 'True', -5, False]
```

```
>>data[:-1]
```

```
[2, 'a', 'True', -5]
```

```
>>data[-3:-1]
```

```
['True', -5]
```

```
>>data[1:-1]
```

```
['a', 'True', -5]
```

```
>>data[:]
```

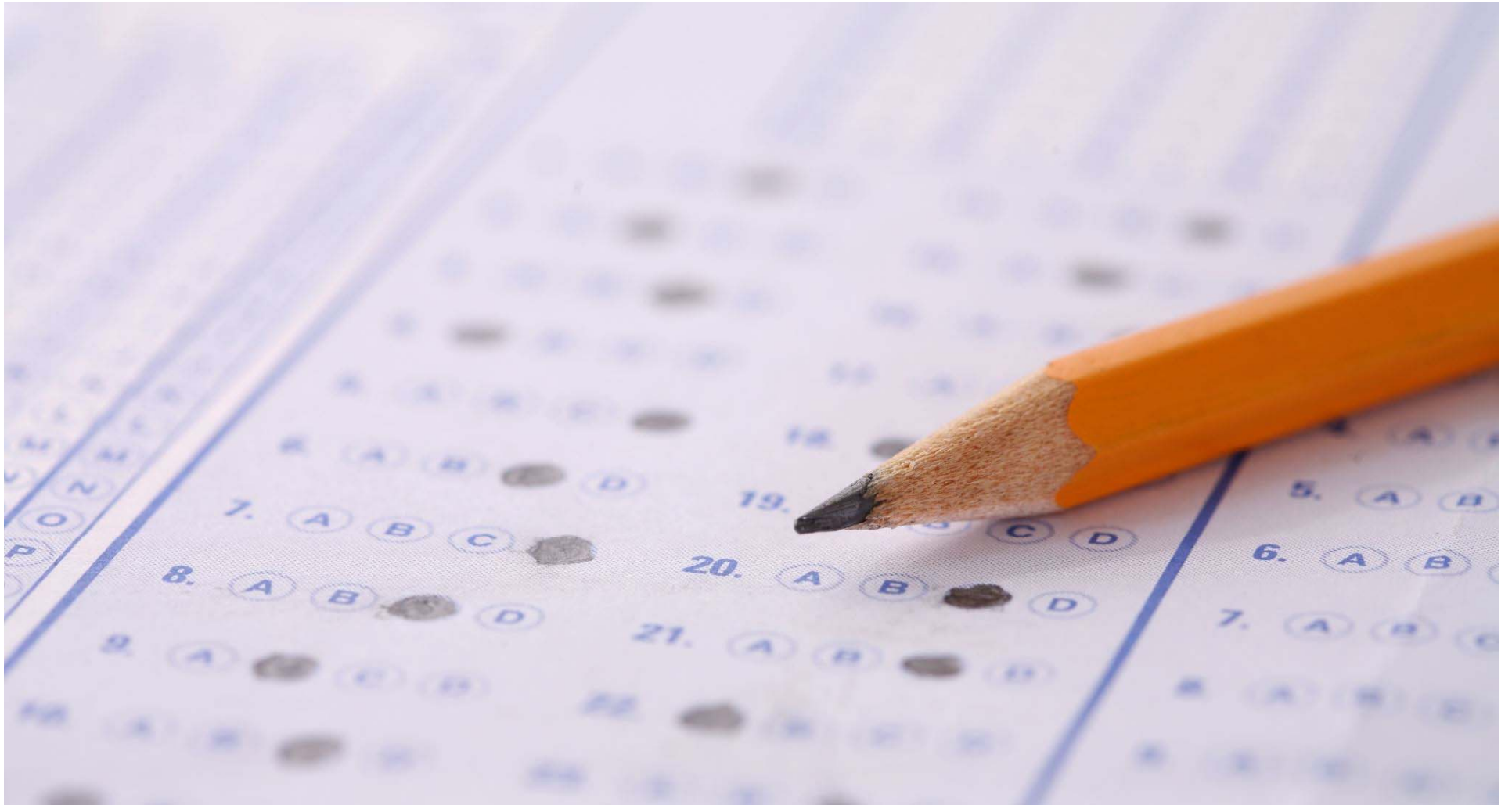
```
[2, 'a', 'True', -5, False]
```

```
>>data[::2]
```

```
[2, 'True', False]
```

```
>>data[-1:3]
```

```
[]
```



Quiz

Quiz

กำหนด list

numbers = [1,2,3,4,5,6]

จงเขียน list slicing เพื่อให้ได้ผลลัพธ์ดังนี้

[2, 3, 4] __numbers[1:4]__

[1, 3, 5] _____

[2, 4, 6] _____

[1, 4] _____

[5, 3, 1] _____

[4, 3, 2] _____

List comprehension

List comprehension

- การสร้าง List โดยอาศัยนิยามทางคณิตศาสตร์
- ใช้ร่วมกับ for ซึ่งสร้างตัวแปรนับเพื่อกำหนดสมาชิกภายใน List
- สามารถใช้ if ร่วมด้วยเพื่อกำหนดเงื่อนไขของสมาชิกใน List ได้

เช่น List ของ x โดยที่ x เป็นจำนวนเต็มในช่วง [1,100)

```
data = [x for x in range(1, 100)]
```

หากไม่ใช่ list comprehension สามารถเขียนได้ดังนี้

```
data = []  
for x in range(1, 100):  
    data.append(x)
```

ตัวอย่างเพิ่มเติม

List ของสมาชิกใน data และชนิดข้อมูลของสมาชิกนั้นเป็นจำนวนเต็ม

```
data = [2, 'a', 'True', -5, False]
data_int = [item for item in data
            if type(item) == int]
```

หากไม่ใช่ list comprehension สามารถเขียนได้ดังนี้

```
data_int = []
for item in data:
    if type(item) == int:
        data_int.append(item)
```

ตัวอย่างเพิ่มเติม

List ของชนิดข้อมูลของสมาชิกแต่ละตัวใน data

```
data_type = [type(item) for item in data]
```

หากไม่ใช่ list comprehension สามารถเขียนได้ดังนี้

```
data_type = []  
for item in data:  
    data_type.append(type(item))
```

ตัวอย่างเพิ่มเติม

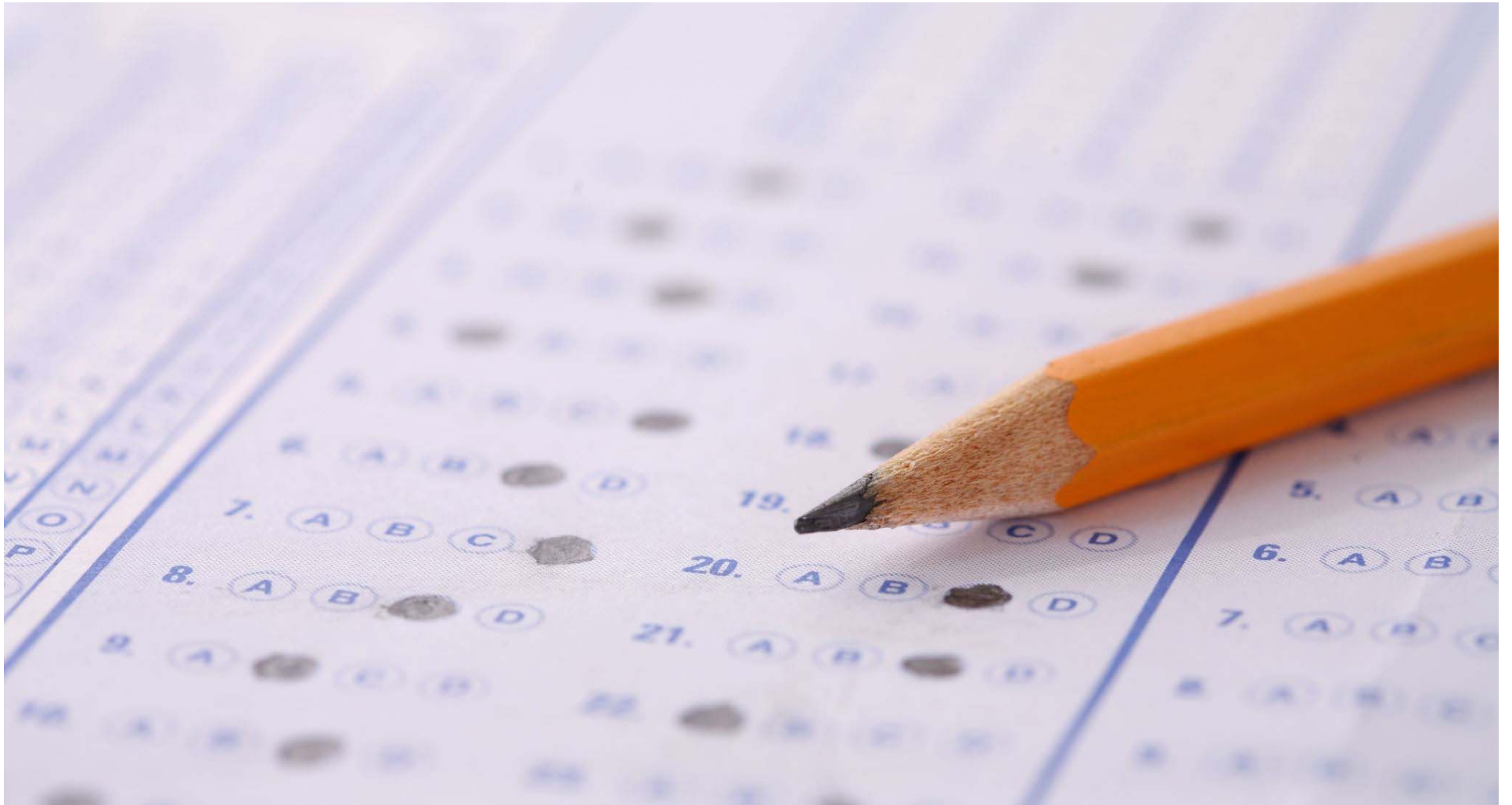
กำหนดให้ x เป็นจำนวนเต็มบวก ต้องการสร้าง factor เป็น list ของตัวประกอบของ x

```
factor = [num for num in range(1, x+1)
          if x % num == 0]
```

- List ของ num โดยที่ num เป็นจำนวนเต็มในช่วง $[1, x]$ และ x หารด้วย num เหลือเศษ 0 (หารลงตัว)

หากไม่ใช่ list comprehension สามารถเขียนได้ดังนี้

```
factor = []
for num in range(1, x+1):
    if x % num == 0:
        factor.append(num)
```



Quiz

Quiz

จงใช้ list comprehension สำหรับข้อต่อไปนี้

1. จงสร้าง list ของเลขสุ่ม 10 ตัว โดยใช้ฟังก์ชัน random ซึ่งคืน float ระหว่าง 0 ถึง 1
2. จากลิสต์ที่สร้างขึ้นในข้อแรก จงสร้าง list ที่สมาชิกมีค่าเกิน 0.5
3. กำหนด list ชื่อ account_no ซึ่งเก็บข้อมูลเลขที่บัญชีซึ่งเป็นเลขแปดหลักที่ไม่ขึ้นต้นด้วย 0 จงสร้าง list ที่ตัดมาเฉพาะสี่หลักแรก

String

- For in string
- The term "Immutable"
- String slicing
- in and not in string
- Methods for string

String

- สตริง หรือสายอักขระ มีคลาสเป็น **str** ในไพธอน
- ลักษณะของสตริงจะคล้าย List ของอักขระ
- สามารถอ่านค่าสมาชิกแต่ละตัวในสตริงได้โดยใช้ดัชนีเช่นเดียวกับ List แต่ไม่สามารถแก้ไขค่าของสมาชิกได้
- กำหนด `a = '123Abc'`
จะได้
 - `a[0]` ผูกกับอักขระ 1
 - `a[-1]` ผูกกับอักขระ c
- ขนาดของสตริงคือจำนวนอักขระภายในสตริง ใช้ฟังก์ชัน **len** ในการหา เช่นเดียวกับ List เช่น **len(a)** จะมีค่า 6

for in string

สามารถใช้ for เพื่อวนทีละตำแหน่งใน string ได้

เช่น

```
name = "Han"  
for ch in name:  
    print(ch)
```

ในรอบแรก ch มีค่าเป็น 'H'

ในรอบสอง ch มีค่าเป็น 'a'

ในรอบสาม ch มีค่าเป็น 'n'

ผลลัพธ์จากการรันคือ

H

a

n

Strings are immutable

- ในภาษาไพธอน `string` หากถูกสร้างขึ้นแล้วจะไม่สามารถถูกแก้ไขได้ เรียกว่า `string` มัน `immutable` เช่น

```
>>name = 'Ginger'  
>>name[1] = 'e'  
TypeError: 'str' object does not support item  
assignment
```

Strings are immutable

- Operation บางอย่างเช่นการนำ string มาต่อกัน อาจดูเหมือนมีการแก้ไข string แต่จริง ๆ แล้ว string ใหม่จะถูกสร้างขึ้นเช่นรูปด้านล่าง

```
name = 'Ginger'
```

name → Gi nger

```
name = name + ' Summers'
```

name

Gi nger

Gi nger Summers

String slicing

- สามารถทำ string slicing ได้โดยหลักการเดียวกับ list slicing นั่นคือใช้รูปแบบ `string[start:end:step]`
 - start คือตำแหน่งของตัวอักษรแรกที่ต้องการหั่น
 - end คือตำแหน่งของตัวอักษรสุดท้ายที่ต้องการหั่น (string ที่หั่นออกมาจะเริ่มที่ start จนถึง end แต่ไม่รวมตำแหน่ง end)
 - Step คือระยะห่างระหว่างดัชนี

```
>>name = "Ginger Summers"
>>name[: 6]
'Ginger'
>>name[7: ]
'Summers'
>>name[1: 4]
'ing'
```

in และ not in กับ String

- สามารถใช้ in เพื่อตรวจสอบว่า string เป็นส่วนหนึ่งของอีก string หรือไม่ โดยใช้รูปแบบ

string1 in string2

- ถ้า string1 เป็นส่วนหนึ่งของ string2 จะคืนค่า True

- สามารถใช้ not in เพื่อตรวจสอบว่า string ไม่เป็นส่วนหนึ่งของอีก string โดยใช้รูปแบบ

string1 not in string2

- ถ้า string1 ไม่เป็นส่วนหนึ่งของ string2 จะคืนค่า True

```
>>text = "I woke up in  
the morning"  
>>'woke' in text  
True  
>>'wake' in text  
False  
>>'woke' not in text  
False  
>>'wake' not in text  
True
```

เมธอดอื่น ๆ สำหรับตรวจสอบstring

เรียกเมธอดเหล่านี้ในรูปแบบ `string.method(arguments)`

<code>isalnum()</code>	คืนค่า True หาก string มีเฉพาะตัวเลขหรือตัวอักษรและความยาวอย่างน้อยหนึ่งตัวอักษร
------------------------	----------------------------------------------------------------------------------

<code>isalpha()</code>	คืนค่า True หาก string มีเฉพาะตัวอักษรและความยาวอย่างน้อยหนึ่งตัวอักษร
------------------------	------------------------------------------------------------------------

<code>isdigit()</code>	คืนค่า True หาก string มีเฉพาะตัวเลขและความยาวอย่างน้อยหนึ่งตัวอักษร
------------------------	----------------------------------------------------------------------

<code>islower()</code>	คืนค่า True หากทุกตัวอักษรใน string เป็นตัวพิมพ์เล็ก
------------------------	------------------------------------------------------

<code>isspace()</code>	คืนค่า True หาก string มีแต่ whitespace และความยาวอย่างน้อยหนึ่งตัวอักษร
------------------------	--------------------------------------------------------------------------

<code>isupper()</code>	คืนค่า True หากทุกตัวอักษรใน string เป็นตัวพิมพ์ใหญ่
------------------------	------------------------------------------------------


```
>>text = 'hello'
>>text.isalpha()
True
>>text.isalnum()
True
>>text.isdigit()
False
>>text.islower()
True
>>text.isupper()
False
```

```
>>text = '022185141'
>>text.isdigit()
True
>>text.isalpha()
False
>>text = ' '
>>text.isspace()
True
>>text = ' hello'
>>text.isalpha()
False
```

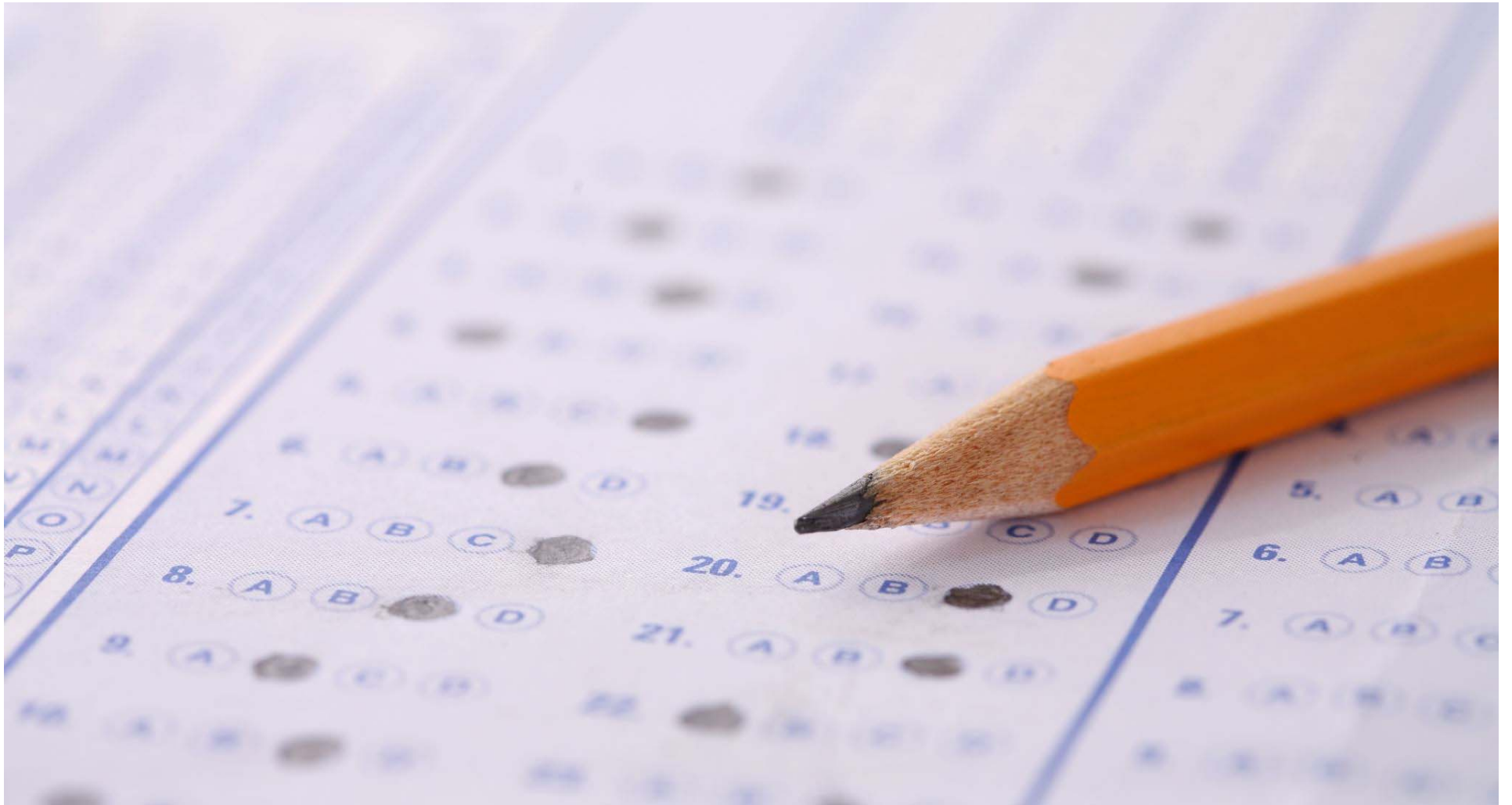
เมธอดที่กระทำกับ string

`upper()` คัดลอก copy ของ string โดยตัวอักษรทุกตัวจะเป็นตัวพิมพ์ใหญ่

`lower()` คัดลอก copy ของ string โดยตัวอักษรทุกตัวจะเป็นตัวพิมพ์เล็ก

`split()` ตัด string โดยหากไม่ระบุตัวตัดจะตัดโดย whitespace และคืนค่ากลับมาเป็น list ของ string ที่ถูกตัด

```
>>text ="UPPER lower"  
>>text.upper()  
'UPPER LOWER'  
>>text.lower()  
'upper lower'  
>>text  
'UPPER lower'
```

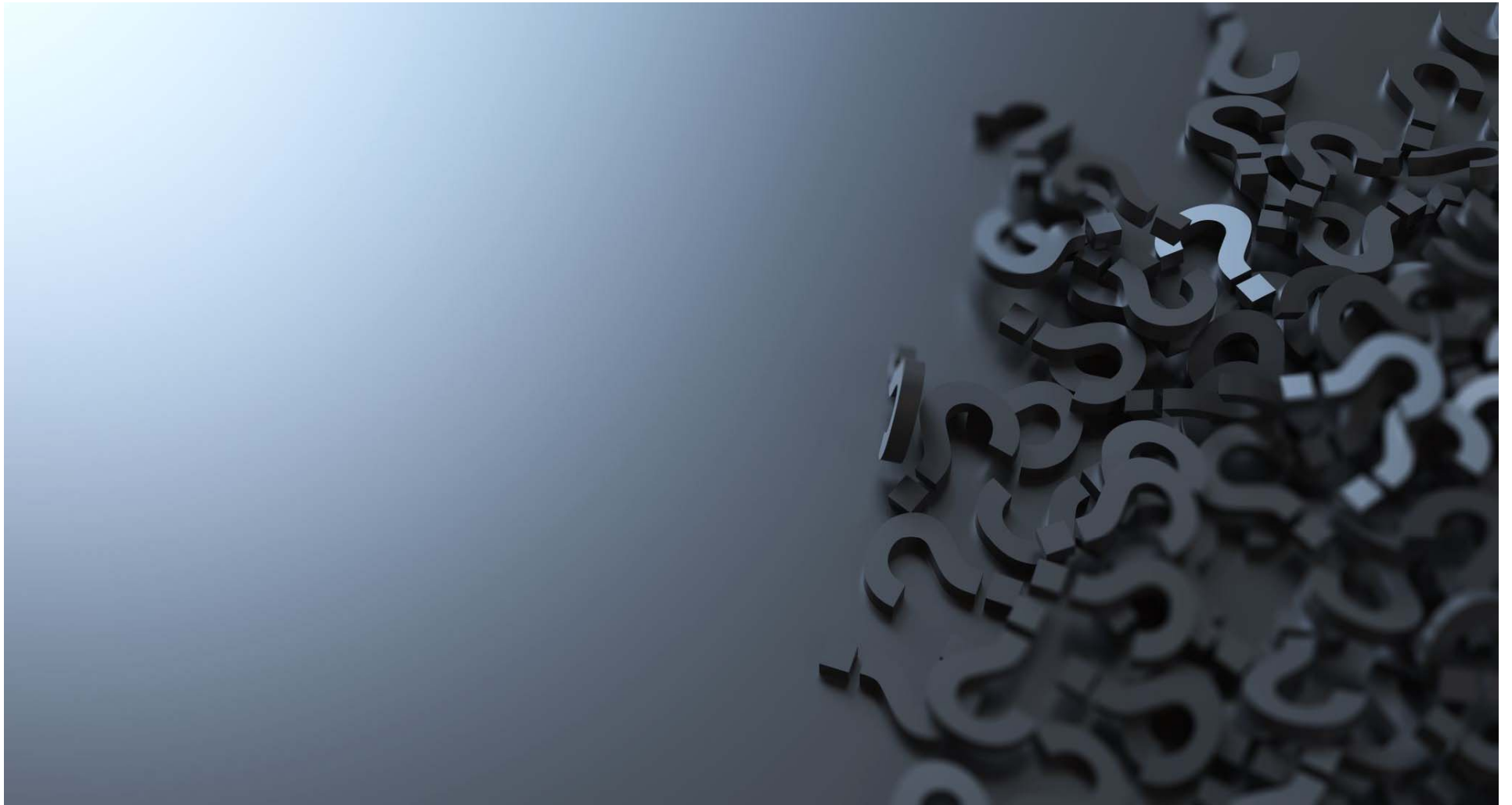


Quiz

Quiz

กำหนด mystring เก็บข้อมูลชนิด string

1. จงเขียนโปรแกรมเพื่อนับจำนวนตัวพิมพ์ใหญ่ใน mystring โดยใช้ list comprehension
2. จงเขียนโปรแกรมเพื่อแสดงจำนวนตัวเลขและตัวอักษร
3. จงคืนค่า True หาก mystring มีคำว่า 'is'
4. จงทำ string slicing เพื่อเอาเฉพาะตัวอักษรในตำแหน่งคู่ออกมา



Exercises

Question 1

รับ list ของ integer 2 list และตรวจสอบว่าสอง list นี้เหมือนกัน
อย่าง circular หรือไม่ ปรี้น yes หากเหมือนกัน ปรี้น no หากไม่
เหมือนกัน

ตัวอย่างเช่น

please input values in list1:10 10 0 0 10

please input values in list2:10 10 10 0 0

yes

Question 2

หุ่นยนต์เดินในระนาบโดยเริ่มจากจุดเริ่มต้น (0,0) หุ่นยนต์สามารถเดินขึ้น (U) เดินลง (D) เดินซ้าย (L) เดินขวา (R) โดยเดินตาม input ที่รับเข้ามาเช่น U 5 D 3 L 3 R 2

ตัวเลขหลังตัวอักษรที่บอกทิศการเดิน (U, D, L, R) คือจำนวนก้าวที่เดิน จงคำนวณการกระจัดจากจุดเริ่มต้นกับจุดสุดท้ายที่หุ่นยนต์อยู่หลังจากเดินเสร็จ

ตัวอย่างเช่น

Enter a sequence of move:U 5 D 3 L 3 R 2

2.23606797749979

Question 3

จงเขียนโปรแกรมเพื่อคำนวณยอดเงินสุดท้ายจาก transaction log ซึ่งรับมาจากทางหน้าจอ โดย transaction log จะอยู่ในรูปแบบ [W|D number]* ความยาวหนึ่งบรรทัด เช่น D 300 D 300 W 200 D 300

W หมายถึงถอน

D หมายถึงฝาก

ผลลัพธ์ของตัวอย่างข้างต้นคือ $300 + 300 - 200 + 300 = 700$

ตัวอย่างการรัน

Specify deposit and withdrawal amounts:

D 300 D 300 W 200 D 300

Total amount is 700

Question 4

จงเขียนโปรแกรมที่ตรวจสอบความถูกต้องของ password โดยจะถือว่าถูกต้องหากมีตัว c อย่างน้อยหนึ่งตัวและมีเลข 1 เพียงตัวเดียว เช่น ABcd0123bca\$ ถูกต้อง ส่วน c11 ไม่ถูกต้อง

ตัวอย่างการรัน

Please enter the password you want to check for validity:ABcd0123bca\$

The password is valid