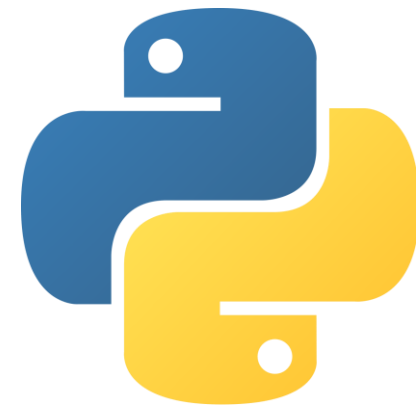


อัปเดตล่าสุดวันที่ 09.09.2566



# รายวิชา การเขียนโปรแกรมเชิงวัตถุ รหัสวิชา 4122309

บทที่\_6\_ลิสต์และการทำงานกับลิสต์



โดย ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่องเนียม  
สาขาวิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏพระนคร

# Agenda

ในบทนี้จะอธิบายเกี่ยวกับโครงสร้างข้อมูลแบบลิสต์ (List) ในภาษาไพธอน โดยจะอธิบายถึงการสร้างและใช้งานลิสต์ในเบื้องต้น การใช้งานเมธอดและฟังก์ชันเพื่อจัดการข้อมูลภายในลิสต์และการทำงานคำสั่ง for loop กับ List รวมถึงการ slicing

6.1 ลิสต์ในภาษาไพธอน

6.2 คำสั่งสำหรับทำงานวนซ้ำในลิสต์

6.3 ฟังก์ชันสำหรับการทำงานในลิสต์



## 6.1 ความหมายของลิสต์

ลิสต์(List) คือโครงสร้างข้อมูลชนิดหนึ่งในภาษาไพธอน ที่ใช้เก็บข้อมูลแบบลำดับ (Sequence) โดยมี Index เป็นตัวระบุตำแหน่งในการเข้าถึงข้อมูล เราสามารถใช้ลิสต์เพื่อเก็บข้อมูลจำนวนมากและหลากหลายประเภทในเวลาเดียวกันลิสต์เป็นประเภทข้อมูลที่ใช้อย่างหลากหลายในการเขียนโปรแกรม นอกจากนี้ ในภาษาไพธอน ยังมี built-in function ที่สามารถทำงานกับลิสต์ และในลิสต์ออบเจกต์เองก็มีเมธอดต่าง ๆ เป็นจำนวนมากที่ช่วยอำนวยความสะดวกในการเขียนโปรแกรม



Python Lists

```
[1, 2, 10.5, 'Python', 'List']
```

## 6.2 การประกาศตัวแปร ลิสต์

ลิสต์นั้นเป็นตัวแปรประเภทหนึ่ง การใช้งานของมันจะเหมือนกับอาเรย์ในภาษาอื่นๆ ในการประกาศลิสต์นั้นข้อมูลของลิสต์จะอยู่ภายในเครื่องหมาย [ ] และกั้นสมาชิกแต่ละตัวด้วยเครื่องหมายคอมมา (,) , มาดูตัวอย่างการประกาศลิสต์ในภาษาไพธอน

```
VS_Code_Projects > PL1 > Python_7_01_List.py > ...
```

```
1 numbers = [-1, 2, 5, 8, 10, 13]
2 names = ['Asst. Dr. Nattapong', 'Somchai', 'Somying', 'Somsri', 'Somruk']
3 mixed_type = [-2, 5, 84.2, "Programming", "Python", "Language"]
4 |
```

ในตัวอย่าง เราได้สร้างตัวแปรลิสต์สามตัวแปร numbers เป็นตัวแปรลิสต์ ที่มีสมาชิกเป็นตัวเลขจำนวนเต็ม 6 ตัว names เป็นลิสต์ ของสตริง ที่สำหรับเก็บชื่อและมี 5 รายชื่อ และสุดท้ายตัวแปร mixed\_type เป็น ลิสต์ ที่เก็บข้อมูลประเภทต่าง ๆ แบบรวมกันในตัวแปรเดียวซึ่งมีสมาชิกทั้งหมด 5 ตัว ซึ่งทั้งหมดนี้เป็นการกำหนดสมาชิกให้กับลิสต์ พร้อมกับการประกาศตัวแปร

## 6.3 การเพิ่มรายการในลิสต์

### ให้สร้างไฟล์ใหม่ ชื่อ List2.py

ในภาษาไพธอนสามารถกำหนดค่าให้กับลิสต์ หลังจากประกาศตัวแปรแล้วได้ มาดูตัวอย่าง

VS\_Code\_Projects > PL1 > Python\_7\_02\_List\_Append.py > ...

```

1 numbers = [] # ประกาศ List ว่างเปล่าไม่มีสมาชิก
2 numbers.append(-1) #การเพิ่มสมาชิก
3 numbers.append(2)
4 numbers.append(5)
5 numbers.append(8)
6 numbers.append(10)
7 numbers.append(13)
8
9 names = ['Asst. Dr. Nattapong', 'Somchai']
10 names.append('Somying')
11 names.append('Somsak')
12 names.append('Somsri')
13 print(numbers)
14 print(names)
15 print('numbers count = ', len(numbers)) # นับจำนวนสมาชิก
16 print('names count = ', len(names)) # นับจำนวนสมาชิก
17

```

```

PS D:\VS_Code_Projects> & C:/Users/ASUS/anaconda3/python.exe d:/VS_Cod
PY
[-1, 2, 5, 8, 10, 13]
['Asst. Dr. Nattapong', 'Somchai', 'Somying', 'Somsak', 'Somsri']
numbers count = 6
names count = 5
PS D:\VS_Code_Projects>

```

ในตัวอย่าง เป็นการกำหนดค่าให้กับลิสต์ หลังจากที่มันถูกสร้างแล้ว เราใช้เมธอด `append()` เพื่อเพิ่มข้อมูลใหม่เข้าไปในลิสต์ ซึ่งข้อมูลที่เพิ่มเข้าจะอยู่ท้ายสุดและเรียง Index เพิ่มขึ้นเรื่อยๆ โดยเริ่มจาก 0 ในตัวแปร `numbers` เราได้เพิ่ม 5 จำนวนเต็มเข้าไปใน List และในตัวแปร `names` ในตอนแรกได้ประกาศและกำหนดสองชื่อให้กับตัวแปร และเพิ่มเข้าไปภายหลังอีก 3 ชื่อ และฟังก์ชัน `len()` ใช้เพื่อบันทึกจำนวนสมาชิกภายใน List

## 6.3 การเพิ่มรายการในลิสต์

1) การเพิ่มรายการใน List  
`List.append()`

2) นับจำนวนสมาชิก  
`Len(List)`

ตัวอย่าง

```
Numbers = [1,2,3]  
Numbers.append(4)
```

ตัวอย่าง

```
Numbers = [1,2,3]  
print('Number count : ' , Len(Numbers))
```

## 6.4 การเข้าถึงข้อมูลภายในลิสต์

ลิสต์นั้นใช้ **Index** สำหรับการเข้าถึงข้อมูล โดย **Index** ของลิสต์จะเป็นจำนวนเต็ม que เริ่มจาก 0 และเพิ่มขึ้นทีละ 1 ไปเรื่อย ๆ ดังนั้น เราจึงสามารถเข้าถึงข้อมูลภายในลิสต์ เพื่ออ่านหรืออัปเดตค่าได้โดยตรงผ่าน **Index** ของมัน นี่เป็นโค้ดการเข้าถึงข้อมูลภายในลิสต์ ในภาษาไพธอน

```
names = ['Asst. Dr. Nattapong Songneam', 'Somchai', 'Somying', 'Somsak', 'Somsri']  
print('names[0] = ', names[0])  
print('names[3] = ', names[3])  
print('names[-1] = ', names[-1])  
  
# update value  
names[0] = 'Bob'  
print('names[0] = ', names[0])
```

ให้สร้างไฟล์ใหม่ ชื่อ List3.py

## 6.4 การเข้าถึงข้อมูลภายในลิสต์

ในตัวอย่างจะมีตัวแปรลิสต์ที่ชื่อว่า `names` ดังนั้น เพื่อเข้าถึงสมาชิกตัวแรกภายในลิสต์ ซึ่งก็คือ "Asst. Dr. Nattapong Songneam" นั้นจะใช้คำสั่ง `names[0]` และสมาชิกที่มีค่าเป็น "Somsak" ซึ่งอยู่ตำแหน่งที่ 4 จะใช้คำสั่ง `names[3]` สังเกตว่า Index จะลดลงหนึ่งเพราะ Index ของลิสต์นั้นเริ่มต้นจาก 0

```
print('names[-1] = ', names[-1])
```

นอกจากนี้ เราสามารถเข้าถึงข้อมูลภายในลิสต์ โดยการใช้ Index เป็นจำนวนลบได้ โดยเริ่มจาก -1 ซึ่งเป็นสมาชิกตัวสุดท้ายของลิสต์ และ -2 สมาชิกตัวถัดมาและลดลงไปทีละ 1

```
names = ['Asst. Dr. Nattapong Songneam', 'Somchai', 'Somying', 'Somsak', 'Somsri']
print('names[0] = ', names[0])
print('names[3] = ', names[3])
print('names[-1] = ', names[-1])

# update value
names[0] = 'Bob'
print('names[0] = ', names[0])
```



## ตัวอย่างโจทย์

1. จงเขียนโปรแกรมเพื่อรับตัวเลขมา 10 จำนวนและ หาค่าผลรวมของตัวเลขทั้งหมด
2. จงเขียนโปรแกรมเพื่อรับตัวเลขมา 10 จำนวนและ หาค่าเฉลี่ยของตัวเลขทั้งหมด
3. จงเขียนโปรแกรมเพื่อรับตัวเลขมา 10 จำนวนและ หาค่ามากที่สุดของตัวเลขทั้งหมด
4. จงเขียนโปรแกรมเพื่อรับตัวเลขมา 10 จำนวนและ หาค่าน้อยที่สุดของตัวเลขทั้งหมด

ให้สร้างไฟล์ใหม่ ชื่อ List3\_exp01.py  
List3\_exp02.py  
List3\_exp03.py  
List3\_exp04.py

# จงเขียนโปรแกรมเพื่อรับตัวเลขมา 10 จำนวนและ หาค่าผลรวมของ ตัวเลขทั้งหมด

```
List3_exp01.py > ...
1  n=[]
2
3  for i in range(10):
4      number = int(input('enter number :'))
5      n.append(number)
6
7  print()
8  sum=0
9  for i in n:
10     print(i)
11     sum+=i
12
13  print("SUM : ", sum)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
enter number :3
enter number :4
enter number :5
enter number :6
enter number :7
enter number :8
enter number :9
enter number :10
```

```
1
2
3
4
5
6
7
8
9
10
```

```
SUM : 55
PS D:\OOP Python>
```

```
4 # Last Modify :: 14.01.2566
5
6 number = []
7 MAX_INPUT = 10 #กำหนดจำนวนที่จะให้กรอก
8 print('Enter %d numbers to the list' % MAX_INPUT)
9 i = 1
10 while i <= MAX_INPUT:
11     print('Number %d: ' % i, end = '')
12     n = int(input())
13     number.append(n) #การเพิ่มค่าที่กรอกลงไปเ็นลิสต์
14     i += 1
15
16 sum = 0
17 for b in number:
18     print(b, end = ',')
19     sum += b
20
21 print('sum = ',sum)
```

```
PS C:\Desktop\ปริศนาตรี\ปี3_ทอม2\01_การเขียนโปรแกรมเชิงวัตถุ_4122309_ดร.นัฐพงศ์ ส่งเนียม>
รี/ปี3_ทอม2/01_การเขียนโปรแกรมเชิงวัตถุ_4122309_ดร.นัฐพงศ์ ส่งเนียม/OOP_Projects/List03_exp01.p
Enter 10 numbers to the list
Number 3: 1
Number 4: 1
Number 5: 1
Number 6: 1
Number 7: 1
Number 8: 1
Number 9: 1
Number 10: 1
1,1,1,1,1,1,1,1,1,1,sum = 10
PS C:\Desktop\ปริศนาตรี\ปี3_ทอม2\01_การเขียนโปรแกรมเชิงวัตถุ_4122309_ดร.นัฐพงศ์ ส่งเนียม>
```

**จงเขียนโปรแกรมเพื่อรับตัวเลขมา 10 จำนวนและ  
หาค่ามากที่สุดของตัวเลขทั้งหมด**

## 6.5 การอ่านค่าในลิสต์ด้วยคำสั่ง For loop

เนื่องจากลิสต์นั้นเก็บข้อมูลเป็นแบบลำดับและใช้ Index ในการเข้าถึงข้อมูล ดังนั้น เราจึงมักจะใช้คำสั่งวนซ้ำสำหรับการเขียนโปรแกรมที่ทำงานกับลิสต์ เพราะทำให้การทำงานรวดเร็วและง่ายขึ้น เช่น การใช้คำสั่งวนซ้ำอ่านค่าในลิสต์ที่มีข้อมูลเป็นจำนวนมาก เป็นต้น ต่อไป มาดูตัวอย่างการใช้งานคำสั่ง For loop กับ List ในภาษา ไพธอน

```
numbers = [10, 20, 30, 40, 50, 60, 70]
```

```
sum = 0
```

```
for n in numbers:
```

```
    print(n, end = ' ')
```

```
    sum += n
```

```
print('sum = ', sum)
```

```
names = ['Asst. Dr. Nattapong', 'Somchai', 'Somying', 'Somsak', 'Somsri']
```

```
for i in range(0, len(names)):
```

```
    print(names[i].upper(), end = ' ')
```

## 6.5 การอ่านค่าในลิสต์ด้วยคำสั่ง For loop

ในตัวอย่าง เป็นการวนอ่านค่าภายในลิสต์ด้วยการใช้คำสั่ง For loop โดยเราได้แยกการทำงานออกเป็นสองลูป ในลูปแรกเป็นการใช้งานคำสั่ง For loop เพื่อวนอ่านค่าภายใน List numbers โดยตรง โปรแกรมจะวนอ่านค่าไปที่ละค่าและนำค่าในแต่ละรอบที่ได้มาใส่ในตัวแปร n เราได้ทำการแสดงผลตัวเลขภายในลิสต์และหาผลรวมของตัวเลขภายในลิสต์โดยเก็บไว้ในตัวแปร sum ในลูปที่สอง เป็นการใช้คำสั่ง For loop เช่นกัน แต่ในตอนนี้เราจะสร้าง Index ขึ้นมาโดยการใช้ตัวแปร i เป็นตัวรับ Index จาก 0 ถึงขนาดตัวสุดท้ายของลิสต์ที่สร้างจากฟังก์ชัน range() และเราแสดงผลชื่อในตัวพิมพ์ใหญ่ด้วยเมธอด upper() ของ String

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```
10 20 30 40 50 60 70 sum = 280
```

```
ASST. DR. NATTAPONG SOMCHAI SOMYING SOMSAK SOMSRI
```

```
#MATEO DANNY JAMES THOMAS LUKE
```

## 6.6 การแบ่งส่วนของลิสต์ (List slicing)

การแบ่งส่วนของลิสต์ (List slicing) ในภาษาไพธอนเราสามารถตัดข้อมูลจากลิสต์หนึ่งแล้วนำไปสร้างเป็นลิสต์ใหม่ได้ โดยวิธีดังกล่าว นั้นเรียกว่า slicing ในการตัดข้อมูลในลิสต์นั้นจะทำในรูปแบบ [a:b] เมื่อ a เป็น Index เริ่มต้นและ b เป็น Index ก่อนสมาชิกตัวสุดท้ายที่ต้องการตัด มาดูตัวอย่างของ list slicing

```
ch = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'] # ลิสต์ของตัวอักษร a - h

a = ch[0:4] # a - d
b = ch[4:9] # e - h
c = ch[:3] # a - c
d = ch[3:] # c - h
e = ch[:] # copy all list, or equivalent to e = ch
f = ch[0:2] + ch[6:8] # a - b and g - h

print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

## 6.6 การแบ่งส่วนของลิสต์ (List slicing)

ในตัวอย่างมีตัวแปร `ch` ซึ่งมีสมาชิกเป็นตัวอักษรในภาษาอังกฤษจาก `a` ถึง `h` หลังจากนั้นเราได้ทำการ `slice` ข้อมูลจาก List ดังกล่าว ในตัวแปร `a` นั้นเป็นการตัดเอาข้อมูลจากตำแหน่งที่ 0 ถึง 3 มา ในตัวแปร `b` นั้นตัดเอาตำแหน่งที่ 4 ถึง 8 ถัดมาเป็นตัวแปร `c` `d` และ `e` เป็นการเว้นว่างตำแหน่งข้างหน้าและข้างหลัง ซึ่งถ้าตำแหน่งข้างหน้าถูกเว้นว่างไว้ เป็นการตัดเอาสมาชิกตั้งแต่ตำแหน่งเริ่มต้นของลิสต์ และถ้าตำแหน่งสิ้นสุดถูกเว้นว่างไว้ เป็นการตัดเอาจนถึงสมาชิกตัวสุดท้ายของลิสต์ และในตัวแปร `f` เป็นการตัดเอาสองส่วนของ List มาต่อกัน ในการนำสองลิสต์มาต่อกันนั้นเราจะใช้ตัวดำเนินการ `+`

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```
['a', 'b', 'c', 'd']  
['e', 'f', 'g', 'h']  
['a', 'b', 'c']  
['d', 'e', 'f', 'g', 'h']  
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']  
['a', 'b', 'g', 'h']
```



## 6.7 การใช้งานคำสั่ง del เพื่อลบข้อมูลในลิสต์

คำสั่ง del เป็นคำสั่งที่ใช้สำหรับลบตัวแปรใด ๆ ออกจากหน่วยความจำหรือใช้ยกเลิกตัวแปรที่เคยประกาศไปแล้ว เราสามารถใช้คำสั่ง del เพื่อลบสมาชิกภายในลิสต์ ได้เช่นเดียวกัน มาดูตัวอย่างการใช้งานคำสั่ง del ในภาษาไพธอน

```
ch = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
```

```
del ch[0] # delete a  
print(ch)
```

```
del ch[0] # delete b  
print(ch)
```

```
del ch[2:4] # delete e, f  
print(ch)
```

```
del ch[:] # delete all  
print(ch)
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```
['b', 'c', 'd', 'e', 'f', 'g', 'h']  
['c', 'd', 'e', 'f', 'g', 'h']  
['c', 'd', 'g', 'h']  
[]
```

## 6.8 เมธอดสำหรับลิสต์ List methods

เมธอดสำหรับลิสต์ (List methods) ในหัวข้อนี้ เราจะพูดถึงการใช้งานลิสต์เมธอด ในภาษาไพธอนเนื่องจากลิสต์ เป็นออบเจกต์ ดังนั้นไพธอนจึงได้ให้เมธอดต่าง ๆ สำหรับทำงานพื้นฐานกับลิสต์ ที่ช่วยอำนวยความสะดวกในการทำงานเป็นอย่างมาก นอกจากนี้ เรายังจะพูดถึงการใช้งาน built-in function กับลิสต์ ถึงแม้ในภาษาไพธอนนั้นมีเมธอดและฟังก์ชันเป็นจำนวนมาก แต่เราจะมาดูตัวอย่างการใช้งานเฉพาะที่สำคัญ

### 6.7.1 การใช้งานเมธอดของลิสต์

ตัวอย่างแรกของการใช้เมธอดของลิสต์นั้นจะเป็นการใช้งานเมธอดเพื่อทำงานพื้นฐานกับลิสต์คือการเพิ่มและลบข้อมูลออกจากลิสต์ โปรแกรมของเราเป็นการใช้ลิสต์เพื่อเก็บรายชื่อ หลังจากนั้นเราใช้เมธอดต่าง ๆ กับเพื่อจัดการกับข้อมูลภายในลิสต์นี้เป็นได้การทำงานของโปรแกรม

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```
['b', 'c', 'd', 'e', 'f', 'g', 'h']  
['c', 'd', 'e', 'f', 'g', 'h']  
['c', 'd', 'g', 'h']  
[]
```

## 6.8 เมธอดสำหรับลิส List methods

เมธอดสำหรับลิส (List methods) ในหัวข้อนี้ เราจะพูดถึงการใช้งานลิสต์เมธอด ในภาษาไพธอนเนื่องจากลิสต์เป็นออบเจกต์ ดังนั้นไพธอนจึงได้ให้เมธอดต่าง ๆ สำหรับทำงานพื้นฐานกับลิสต์ ที่ช่วยอำนวยความสะดวกในการทำงานเป็นอย่างมาก นอกจากนี้ เรายังจะพูดถึงการใช้งาน built-in function กับลิสต์ ถึงแม้ในภาษาไพธอนนั้นมีเมธอดและฟังก์ชันเป็นจำนวนมาก แต่เราจะมาดูตัวอย่างการใช้งานเฉพาะที่สำคัญ

```
names = ['David', 'Johnny', 'Dennis', 'Jason']

# add 2 names to the end of the list
names.append('Andy')
names.append('Rene')
print(names)

# add a name at position 3
names.insert(2, 'Mike')
print(names)

# remove 2 specific names
names.remove('Somchai')
names.remove('Somying')
print(names)

# pop the last item fro the list
print('Popped item =', names.pop())
print(names)

print('Popped item =', names.pop())
print(names)

# clear all items
names.clear()
print(names)
```

## 6.9 เมธอดทั้งหมดของลิสต์ ในภาษาไพธอน

ในสองตัวอย่างก่อนหน้านี้เป็นการใช้งานเมธอดต่าง ๆ ของลิสต์ อย่างไรก็ตามยังมีเมธอดอื่น ๆ อีกที่เราไม่ได้กล่าวถึง นี่เป็นรายการเมธอดทั้งหมดของลิสต์ ในภาษาไพธอน

Methods	Description
<code>append(x)</code>	เพิ่มออบเจ็กต์ x เข้ามายัง List
<code>extend(iterable)</code>	เพิ่มข้อมูลจาก iterable เข้ามายัง List
<code>insert(i, x)</code>	ใส่ออบเจ็กต์ x เข้ามายัง List ที่ตำแหน่ง i
<code>remove(x)</code>	นำออบเจ็กต์ x อันแรกที่พบใน List ออก
<code>pop([i])</code>	นำข้อมูลอันสุดท้ายออกจาก List
<code>clear()</code>	ลบข้อมูลทั้งหมดภายใน List
<code>index(x[, start[, end]])</code>	ค้นหาตำแหน่งของออบเจ็กต์ x
<code>count(x)</code>	นับออบเจ็กต์ x ใน List
<code>sort(key=None, reverse=False)</code>	เรียงข้อมูลภายใน List จากน้อยไปมาก
<code>reverse()</code>	ย้อนกลับลำดับของ List
<code>copy()</code>	คัดลอก List ทั้งหมด

## 6.10 การใช้งานฟังก์ชันกับลิสต์

ในภาษาไพธอน นั้นมี built-in function ที่สามารถใช้งานกับโครงสร้างข้อมูลทั่วไปได้ เช่น List Tuple หรือ Dictionary ฟังก์ชันเหล่านี้อำนวยความสะดวกเป็นอย่างมาก เช่น การนับจำนวนสมาชิก การหาค่าน้อยสุดหรือค่ามากที่สุด เป็นต้น ต่อไปมาดูตัวอย่างการใช้งานฟังก์ชันที่สามารถใช้ได้กับลิสต์

```
numbers = [12, 3, 1, -5, 8, 3, -2, 10]
```

```
print('%d numbers in list' % len(numbers))  
print('Min number = %d' % min(numbers))  
print('Max number = %d' % max(numbers))
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

ในตัวอย่าง เป็นการใช้งานฟังก์ชันกับลิสต์ ของตัวเลข โดยฟังก์ชัน len() ที่เรารู้จักกันดีนั้นใช้สำหรับนับจำนวนของออบเจกต์ภายในลิสต์ อย่างไรก็ตามฟังก์ชันนี้สามารถใช้งานได้กับข้อมูลประเภทอื่น ๆ เหมือนกัน เช่น ใช้กับ Tuple Dictionary หรือการหาความยาวของ String ฟังก์ชัน min() สำหรับหาค่าน้อยที่สุดในลิสต์ และฟังก์ชัน max() นั้นสำหรับหาค่าที่มากที่สุด

## 6.10 การใช้งานฟังก์ชันกับลิสต์

ในภาษาไพธอน นั้นมี built-in function ที่สามารถใช้งานกับโครงสร้างข้อมูลทั่วไปได้ เช่น List Tuple หรือ Dictionary ฟังก์ชันเหล่านี้อำนวยความสะดวกเป็นอย่างมาก เช่น การนับจำนวนสมาชิก การหาค่าน้อยสุดหรือค่ามากที่สุด เป็นต้น ต่อไปมาดูตัวอย่างการใช้งานฟังก์ชันที่สามารถใช้ได้กับลิสต์

```
print(list(range(0, 20, 2)))
```

```
tuple1 = (1, 2, 3, 4, 5)
```

```
print(list(tuple1))
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

8 numbers in list

Min number = -5

Max number = 12

[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

[1, 2, 3, 4, 5]

ต่อไปเป็นการใช้งานฟังก์ชัน list() ในการแปลงข้อมูลประเภทต่าง ๆ ไปเป็นลิสต์ ในคำสั่งแรกเป็นการแปลงข้อมูลจากออบเจกต์ลำดับของตัวเลขที่สร้างจากฟังก์ชัน range() เป็นและในคำสั่งที่สองเป็นการแปลง Tuple ให้เป็นลิสต์

## 6.10 การใช้งานฟังก์ชันกับลิสต์

เนื้อหาในบทนี้ครอบคลุมการใช้งานเมธอดของลิสต์ ในภาษาไพธอน จะได้เห็นตัวอย่างในการใช้งานเมธอดกับลิสต์ ในการเขียนโปรแกรมแบบต่าง ๆ นอกจากนี้ ยังรวมถึงการใช้งาน built-in function ที่สามารถใช้งานกับลิสต์ ได้

- เนื้อหาในบทนี้ครอบคลุมการใช้งานเมธอดของลิสต์ ในภาษาไพธอน จะได้เห็นตัวอย่างในการใช้งานเมธอดกับลิสต์ ในการเขียนโปรแกรมแบบต่าง ๆ นอกจากนี้ ยังรวมถึงการใช้งาน built-in function ที่สามารถใช้งานกับลิสต์ ได้



# แบบฝึกหัด บทที่ 6

1. สร้างลิสต์ว่าง (empty list) ชื่อ `my_list`
2. เพิ่มสมาชิกในลิสต์ `my_list` โดยเพิ่มตัวเลข 1, 2, 3
3. พิมพ์ลิสต์ `my_list` ออกมา
4. เขียนโปรแกรมเพื่อหาความยาว (จำนวนสมาชิก) ของลิสต์ `my_list` แล้วพิมพ์ค่านั้นออกมา
5. สร้างลิสต์ชื่อ `fruits` ที่มีสมาชิกเป็นชื่อผลไม้ "แอปเปิ้ล", "กล้วย", "ส้ม"
6. เขียนโปรแกรมเพื่อวนลูปผ่านลิสต์ `fruits` แล้วพิมพ์ชื่อของผลไม้แต่ละตัวออกมา
7. ให้ทำการลบสมาชิก "กล้วย" ออกจากลิสต์ `fruits` และพิมพ์ลิสต์ `fruits` ใหม่
9. สร้างลิสต์ชื่อ `numbers` ที่มีสมาชิกเป็นตัวเลข 1 ถึง 5
10. สร้างลิสต์ชื่อ `colors` ที่มีสมาชิกเป็นสี "แดง", "เขียว", "น้ำเงิน"
11. สร้างลิสต์ชื่อ `mixed_data` ที่มีสมาชิกเป็นตัวเลข, สตริง, และบูลีน
12. ใช้ลิสต์ `numbers` และ `colors` ในการสร้างลิสต์ใหม่ชื่อ `combined_list` โดยรวมสมาชิกจากทั้งสองลิสต์
13. ใช้ลิสต์ `numbers` ในการสร้างลิสต์ใหม่ชื่อ `squared_numbers` โดยเปลี่ยนแต่ละตัวเลขใน `numbers` เป็นค่ากำลังสองของตัวเลขนั้น
14. เขียนโปรแกรมเพื่อหาค่ามากที่สุดและค่าน้อยสุดในลิสต์ `numbers` และพิมพ์ค่าเหล่านั้นออกมา
15. เขียนโปรแกรมเพื่อหาค่ามากที่สุดและค่าน้อยสุดในลิสต์ `numbers` และพิมพ์ค่าเหล่านั้นออกมา

# ตัวอย่าง แบบฝึกหัด

1. สร้างลิสต์ชื่อ `numbers` ที่มีสมาชิกเป็นตัวเลข 1 ถึง 5

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

2. สร้างลิสต์ชื่อ `colors` ที่มีสมาชิกเป็นสี "แดง", "เขียว", "น้ำเงิน":

```
colors = ["แดง", "เขียว", "น้ำเงิน"]
```

3. เพิ่มสี "ชมพู" ลงในลิสต์ `colors`:

```
colors.append("ชมพู")
```

# ตัวอย่าง แบบฝึกหัด

4. สร้างลิสต์ชื่อ fruits ที่มีสมาชิกเป็นชื่อผลไม้ "แอปเปิ้ล", "กล้วย", "ส้ม"

```
fruits = ["แอปเปิ้ล", "กล้วย", "ส้ม"]
```

5. ให้คุณลบสมาชิก "กล้วย" ออกจากลิสต์ fruits และพิมพ์ลิสต์ fruits ใหม่

```
fruits.remove("กล้วย")  
print(fruits)
```

6. สร้างลิสต์ชื่อ mixed\_data ที่มีสมาชิกเป็นตัวเลข, สตริง, และบูลีน

```
mixed_data = [42, "Hello", True]
```

## อ้างอิง

ภาษาไพธอน. online : <http://marcuscode.com/lang/python>, สืบค้นเมื่อ 2 ก.ค. 2565

<https://www.9experttraining.com/articles/python-คืออะไร>

<https://medium.com/@moungsiri/โครงสร้างภาษา-python-53cc38a51462>

<http://marcuscode.com/lang/python/variables-and-types>

<http://cms576.bps.in.th/group11/introduction-to-computer-programming>

<https://www.mindphp.com/นเรียนออนไลน์/83-python/2398-python-operator.html>