**List methods**

April 12, 2017 · 1,108 views

ในบทนี้ เราจะพูดถึงการใช้งาน List เมธอด ในภาษา Python เนื่องจาก List เป็นออบเจ็ค ดังนั้น Python จึงได้ให้เมธอดต่างๆ สำหรับทำงานพื้นฐานกับ List ที่ช่วยอำนวยความสะดวกในการทำงานเป็นอย่างมาก นอกจากนี้ เรายังจะพูดถึงการใช้งาน built-in function กับ List ถึงแม้ในภาษา Python นั้นมีเมธอดและฟังก์ชันเป็นจำนวนมาก แต่เราจะมาดูตัวอย่างการใช้งานเฉพาะที่สำคัญ

**การใช้งานเมธอดของ List**

ตัวอย่างแรกของการใช้เมธอดของ List นั้นจะเป็นการใช้งานเมธอดเพื่อทำงานพื้นฐานกับ List คือการเพิ่มและลบข้อมูลออกจาก List โปรแกรมของเราเป็นการใช้ List เพื่อเก็บรายชื่อ หลังจากนั้นเราใช้เมธอดต่างๆ กับเพื่อจัดการกับข้อมูลภาย List นี่เป็นโค้ดการทำงานของโปรแกรม

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 specify names
names.remove('David')
names.remove('Johnny')
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)

ในตัวอย่าง เราได้สร้างตัวแปร List ขึ้นมาและกำหนดสมาชิกให้กับตัวแปร โดยมีสมาชิกที่เป็น String จำนวน 4 ตัว หลังจากนั้นเราใช้เมธอด append() ในการเพิ่มอีกสองชื่อเข้ามา โดยเมธอดนี้จะทำการเพิ่มข้อมูลใหม่เข้าไปยังท้ายของ List ต่อมาเราใช้เมธอด insert() เพื่อเพิ่มสมาชิกไปยังตำแหน่งที่ต้องการ เราได้เพิ่ม 'Mike' ไปยังตำแหน่งที่ 3 ของ List ทำให้ในตอนนี้ List ของเรามีสมาชิกทั้งหมด 7 ตัว

names.remove('David')
names.remove('Johnny')

หลังจากนั้นเป็นการใช้งานเมธอด remove() เพื่อนำออกข้อมูลใน List เราได้ทำการลบสองรายชื่อ คือ 'David' และ 'Johnny' ออกไปจาก List โดยเมธอดจะทำการลบข้อมูลแรกที่เจอใน List ออกไปในกรณีที่ข้อมูลนั้นมีเหมือนกันมากกว่าหนึ่งใน List แต่ในการใช้งานเมธอดคุณต้องตรวจสอบให้แน่ใจว่าข้อมูลนั้นมีอยู่จริง ไม่เช่นนั้นจะทำให้เกิดข้อผิดพลาดขึ้น

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

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

ต่อมาเป็นการนำข้อมูลจาก List ด้วยการใช้เมธอด pop() การทำงานของเมธอดนั้นจะนำข้อมูลที่อยู่ท้ายสุดของ List และส่งค่านั้นกลับมา และเนื่องจาก List มีเมธอด append() และ pop() ดังนั้น เราจึงสามารใช้ List เหมือนกับการทำงานของ Stack ได้ และเมธอด clear()นั้นสำหรับลบข้อมูลออกจาก List ทั้งหมด

['David', 'Johnny', 'Dennis', 'Jason', 'Andy', 'Rene']
['David', 'Johnny', 'Mike', 'Dennis', 'Jason', 'Andy', 'Rene']
['Mike', 'Dennis', 'Jason', 'Andy', 'Rene']
Popped item = Rene
['Mike', 'Dennis', 'Jason', 'Andy']
Popped item = Andy
['Mike', 'Dennis', 'Jason']
[]

นี่เป็นผลการทำงานของโปรแกรมในการใช้งานเมธอดเพื่อเพิ่มและลบข้อมูลกับ List ในรูปแบบต่างๆ

มาดูตัวอย่างในการใช้งาน List กับเมธอดเพิ่มเติมในภาษา Python ที่เป็นโปรแกรมในการเก็บชื่อของผลไม้ เราได้ทำการประกาศ List fruits และกำหนดสมาชิก 6 ตัวให้กับ List แต่มี 'apple' ที่มี 2 ตัวใน List นี่เป็นโค้ดการทำงานของโปรแกรม

fruits = ['banana', 'orange', 'apple', 'grape', 'apple', 'peach']

print(fruits)

print('%d apples in list' % fruits.count('apple'))
print('%d banana in list' % fruits.count('banana'))

print('Index of grape is %d' % fruits.index('grape'))

fruits.sort()
print('Sorted', fruits)

fruits.reverse()
print('Reversed', fruits)

ในตัวอย่าง เรามี List ในการเก็บรายชื่อของผลไม้ เมธอด count() ใช้สำหรับนับจำนวนออบเจ็คที่ตรงกันกับที่ระบุภายใน List เราได้ทำการนับ 'apple' และ 'banana' ใน List และแสดงจำนวนที่ได้ออกทางหน้าจอ เมธอด index() นั้นใช้สำหรับหาตำแหน่ง Index ของออบเจ็คตัวแรกที่พบใน List เราได้ใช้เมธอดเพื่อหา Index ของ 'grape' และแสดงผลตำแหน่งที่ได้

เมธอด sort() ใช้สำหรับเรียงข้อมูลภายใน List ตามตัวอักษรจากก่อนไปทีหลัง ในกรณีที่ข้อมูลใน List เป็นตัวเลขจะเป็นการเรียงจากน้อยไปหามากแทน คุณสามารถใช้ Keyword argument ที่ชื่อว่า reverse โดยกำหนดให้เป็น True หากต้องการเรียงจากมากไปหาน้อย ซึ่งโดยปกติแล้วค่านี้จะมีค่าเป็น False และเมธอด reverse() ใช้สำหรับย้อนกลับข้อมูลภายใน List จากตำแหน่งแรกไปตำแหน่งสุดท้ายและในทางกลับกัน

['banana', 'orange', 'apple', 'grape', 'apple', 'peach']
2 apples in list
1 banana in list
Index of grape is 3
Sorted ['apple', 'apple', 'banana', 'grape', 'orange', 'peach']
Reversed ['peach', 'orange', 'grape', 'banana', 'apple', 'apple']

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

ในสองตัวอย่างก่อนหน้า เป็นการใช้งานเมธอดต่างๆ ของ List อย่างไรก็ตามยังมีเมธอดอื่นๆ อีกที่เราไม่ได้กล่าวถึง นี่เป็นรายการเมธอดทั้งหมดของ List ในภาษา Python

|  |  |
| --- | --- |
| **Methods** | **Description** |
| append(x) | เพิ่มออบเจ็ค x เข้ามายัง List |
| extend(iterable) | เพิ่มข้อมูลจาก iterable เข้ามายัง List |
| insert(i, x) | ใส่ออบเจ็ค x เข้ามายัง List ที่ตำแหน่ง i |
| remove(x) | นำออบเจ็ค x อันแรกที่พบใน List ออก |
| pop([i]) | นำข้อมูลอันสุดท้ายออกจาก List |
| clear() | ลบข้อมูลทั้งหมดภายใน List |
| index(x[, start[, end]]) | ค้นหาตำแหน่งของออบเจ็ค x |
| count(x) | นับออบเจ็ค x ใน List |
| sort(key=None, reverse=False) | เรียงข้อมูลภายใน List จากน้อยไปมาก |
| reverse() | ย้อนกลับลำดับของ List |
| copy() | คัดลอก List ทั้งหมด |

**การใช้งานฟังก์ชันกับ List**

ในภาษา Python นั้นมี built-in function ที่สามารถใช้งานกับโครงสร้างข้อมูลทั่วไปได้ เช่น List Tupple หรือ Dictionary ฟังก์ชันเหล่านี้อำนวยความสะดวกเป็นอย่างมาก เช่น การนับจำนวนสมาชิก การหาค่าน้อยสุดหรือค่ามากสุด เป็นต้น ต่อไปมาดูตัวอย่างการใช้งานฟังก์ชันที่สามารถใช้ได้กับ List

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))

ในตัวอย่าง เป็นการใช้งานฟังก์ชันกับ List ของตัวเลข โดยฟังก์ชัน len() ที่เรารู้จักกันดีนั้นใช้สำหรับนับจำนวนของออบเจ็คภายใน List อย่างไรก็ตามฟังก์ชันนี้สามารถใช้งานได้กับข้อมูลประเภทอื่นๆ เหมือนกัน เช่น ใช้กับ Tuple Dictionary หรือการหาความยาวของ String ฟังก์ชัน min() สำหรับหาค่าที่น้อยที่สุดใน List และฟังก์ชัน max() นั้นสำหรับหาค่าที่มากที่สุด

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

tuple1 = (1, 2, 3, 4, 5)
print(list(tuple1))

ต่อไปเป็นการใช้งานฟังก์ชัน list() ในการแปลงข้อมูลประเภทต่างๆ ไปเป็น List ในคำสั่งแรกเป็นการแปลงข้อมูลจากออบเจ็คลำดับของตัวเลขที่สร้างจากฟังก์ชัน range() เป็น List และในคำสั่งที่สองเป็นการแปลง Tuple ให้เป็น List

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]

นี่เป็นผลลัพธ์การทำงานของโปรแกรม ในการใช้ฟังก์ชันกับ List

ในบทนี้ เราได้ครอบคลุมการใช้งานเมธอดของ List ในภาษา Python คุณได้เห็นตัวอย่างในการใช้งานเมธอดกับ List ในการเขียนโปรแกรมแบบต่างๆ นอกจากนี้ ยังทราบถึงการใช้งาน built-in function ที่สามารถใช้งานกับ List ได้