

List, Tuple



<https://www.w3schools.com/python/>

Lists

- เป็นรูปแบบการจัดเก็บค่าหลายค่าไว้ในตัวแปรเดียวคล้ายกับอาร์เรย์ (arrays)
- สามารถเก็บข้อมูลประเภทใดก็ได้ หรือเก็บค่าตัวแปรหรือ List ด้วยกันก็ได้
- สามารถใช้การวนลูปคำสั่ง for ในการเรียกดูรายการข้อมูลใน List ได้

```
mylist = [5717400144, "chatchai"]
```

เพิ่มข้อความหรือค่าต่อท้ายรายการ	<code>mylist.append("Computer Science")</code>
แทรกข้อความไว้ในตำแหน่งที่ 2 (รายการที่ 3) ของ mylist	<code>mylist.insert(2, "Faculty of Science")</code>
ดึงรายการที่ 2 ออกจาก mylist	<code>item = mylist.pop(2)</code>
ลบรายการออกจาก mylist	<code>mylist.remove("chatchai")</code>
คัดลอกรายการใน mylist ไปเก็บไว้ใน newlist	<code>newlist = mylist.copy()</code>
สลับรายการใน newlist จากหลังมาหน้า	<code>newlist.reverse()</code>
ลบรายการทั้งหมดออกจาก mylist	<code>mylist.clear()</code>

Lists

```
mylist = [1, "chatchai", 3.5]
print(mylist[0])
print(mylist[1])
print(mylist[2])
```

```
mylist = []

#append 1,2,3 into mylist
mylist.append(1)
mylist.append(2)
mylist.append(3)

# prints out 1,2,3
for x in mylist:
    print(x)
```

```
mylist = [1, "chatchai", 3.5]

mylist.pop(2) # remove third element from list
mylist.remove("chatchai") # remove chatchai from list
print(mylist)
```

List

แก้ไขรายการตำแหน่งแรก และแสดงแต่ละรายการใน list

```
Primes = [2, 3, 5, 7, 11, 13]
Rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']

print(Rainbow[0])
Rainbow[0] = 'red'    #Update value in list

print('Print the rainbow')
for i in range(len(Rainbow)):
    print(Rainbow[i])
```

เพิ่มรายการใหม่ใน list ตามจำนวนที่ต้องการ

```
a = [ ] # start an empty list
n = input("Number of Elements: ")
for i in range(n):
    newelement = int(input("Enter Element :"))
    a.append(newelement)

print(a)
```

Split and join methods

- เมธอด split เป็นคำสั่งในการแยกข้อความออกเป็นคำ ๆ แล้วเก็บไว้ใน List โดยผู้ใช้งานสามารถกำหนดตัวอักขระที่ใช้ในการแยกข้อความได้ (ไม่ระบุตัวอักขระจะใช้ช่องว่าง space)

```
s = input("Enter your name: ")  
a = s.split()  
print(a)
```

```
a = input("Enter your id: ").split()  
for i in range(len(a)):  
    a[i] = int(a[i])  
print(a)
```

```
a = '192.168.0.1'.split('.')  
print(a)
```

- เมธอด join เป็นคำสั่งที่ใช้ในการรวมข้อความหรือค่าต่าง ๆ ภายใน List ให้กลายเป็นข้อความเดียวกัน

```
a = ['red', 'green', 'blue']  
print(' '.join(a))  
  
b = [1, 2, 3]  
print(' '.join([str(i) for i in b]))
```

Lists

- เราสามารถใช้คำสั่ง for ในการวนสร้างค่าภายใน List ได้

```
a = [0 for i in range(5)]  
print(a)
```

```
a = [1 for i in range(5)]  
print(a)
```

```
n = 5  
a = [i ** 2 for i in range(1, n + 1)]  
print(a)
```

```
# random ten numbers from 0 to 10
```

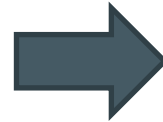
```
from random import randrange  
n = 10  
a = [randrange(1, 10) for i in range(n)]  
print(a)
```

```
# input three elements into list
```

```
a = [input() for i in range(3)]  
print(a)
```

Python For Loops

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```



```
apple  
banana  
cherry
```

- ตัวดำเนินการ "in"

- ตัวดำเนินการ "in" ใช้สำหรับตรวจสอบว่าค่าในตัวแปรปรากฏอยู่ในรายการภายใน list หรือไม่

```
name = input("Enter your search name : ")  
  
if name in ["John", "Rick"]:  
    print("Your name is either John or Rick.")
```

Python Tuples

- Tuple เป็นการจัดเก็บรายการข้อมูลหลายค่าไว้ในตัวแปรเดียวเช่นเดียวกับ List, Dictionary, Set
- Tuple เป็นชุดรายการที่มีการจัดเรียงตามข้อมูลและไม่เปลี่ยนแปลงข้อมูลได้ (ordered and unchangeable)
- Tuple อนุญาตให้มีรายการค่าซ้ำกันได้ (Allow duplicate)

```
fruits = ("apple", "banana", "cherry")  
print(fruits)
```

```
fruits = ("apple", "banana", "cherry", "apple", "cherry")  
print(fruits)
```


Python Tuples

- Tuple เป็นการจัดเก็บรายการข้อมูลหลายค่าไว้ในตัวแปรเดียวเช่นเดียวกับ List, Dictionary, Set
- Tuple เป็นชุดรายการที่มีการจัดเรียงตามข้อมูลและไม่เปลี่ยนแปลงข้อมูลได้ (ordered and unchangeable)
- Tuple อนุญาตให้มีรายการค่าซ้ำกันได้ (Allow duplicate)
- คำสั่ง len แสดงจำนวนรายการใน tuple
- คำสั่ง type แสดงประเภทของข้อมูลของ tuple

```
fruits = ("apple", "banana", "cherry")  
print(fruits)
```

```
fruits = ("apple", "banana", "cherry", "apple", "cherry")  
print(fruits)
```

```
print(len(fruits))  
print(type(fruits))
```

Python Tuples

- การเข้าถึงรายการใน tuple สามารถระบุเป็นตัวเลขตำแหน่ง โดยเริ่มต้นตำแหน่งที่ 0 ถึง n-1
 - ค่าตำแหน่งติดลบ เป็นการอ้างตำแหน่งจากขวามาซ้าย (ด้านหลังมาด้านหน้า)
 - ช่วงตำแหน่ง เป็นตำแหน่งเริ่มต้น จนถึงตำแหน่งสุดท้าย
 - เฉพาะตำแหน่งแรก เป็นตำแหน่งเริ่มต้น จนถึงตำแหน่งสุดท้าย
- การค้นหาค่าข้อมูลใน tuple สามารถทำได้โดยใช้คำสั่ง in

```
fruits = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
```

```
print(fruits[1])
```

```
print(fruits[-1])
```

```
print(fruits[2:5])
```

```
print(fruits[2:])
```

```
print(fruits[-4:-1])
```

```
if "orange" in fruits:
```

```
    print ("Yes, 'orange' is in the fruits tuple")
```

Python Tuples

- Unpack tuple เป็นการแยกรายการไปเก็บไว้ในตัวแปรต่าง ๆ

```
fruits = ("apple", "banana", "cherry")
```

```
(green, yellow, red) = fruits
```

```
print(green)
```

```
print(yellow)
```

```
print(red)
```

Python Tuples

● Iterating over tuples

```
#แสดงรายการผลไม้ใน fruits
fruits = ("apple", "banana", "cherry")

for f in fruits:
    print (f)
```

```
#แสดงรายการผลไม้ใน fruits
fruits = ("apple", "banana", "cherry")

for i in range(len(fruits)):
    print (fruits[i])
```

```
#รวมรายการ tuple1 และ tuple2 เข้าด้วยกัน

tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2
print(tuple3)
```

```
#แสดงรายการผลไม้ใน fruits เพิ่มขึ้น 2 เท่า

fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2

print(mytuple)
```

Python Tuple Methods

Method	Description
<u>count()</u>	นับจำนวนค่าตามชื่อที่กำหนดไว้
<u>index()</u>	แสดงค่าตำแหน่งที่พบตามชื่อที่กำหนดไว้

```
fruits = ("apple", "banana", "cherry", "apple", "cherry")
```

```
print(fruits.count("apple"))
```

```
print(fruits.index("banana"))
```