Ngôn Ngữ Lập Trình Python
LIST

Nội Dung

- Cách tạo List
- Các hàm và toán tử trên List

Tạo List

- **List Rỗng:** **[]** hoặc dùng **list()**

```
print("Two standard ways to create an empty list:")
a = [ ]
b = list()
print(type(a), len(a), a)
print(type(b), len(b), b) print(a == b)
```

```
Two standard ways to create an empty list:
<class 'list'> 0 []
<class 'list'> 0 []
True
```

Tạo List

- **List có một phần tử**

```
a = [ "hello" ]
b = [ 42 ]
print(type(a), len(a), a)
print(type(b), len(b), b)
print(a == b)
```

```
<class 'list'> 1 ['hello']
<class 'list'> 1 [42]
False
```

Tạo List

- **List có nhiều phần tử:** các phần tử cách nhau bởi dấu phẩy

```
a = [2, 3, 5, 7]
b = list(range(5))
c = ["mixed types", True, 42]
print(type(a), len(a), a)
print(type(b), len(b), b)
print(type(c), len(c), c)
```

```
<class 'list'> 4 [2, 3, 5, 7]
<class 'list'> 5 [0, 1, 2, 3, 4]
<class 'list'> 3 ['mixed types', True, 42]
```

Tạo List

```
n = 10
a = [0] * n
b = list(range(n))
print(type(a), len(a), a)
print(type(b), len(b), b)
```

```
<class 'list'> 10 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
<class 'list'> 10 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Các Hàm và Toán Tử Thông Dụng

- **len(), min(), max(), sum()**

```
a = [ 2, 3, 5, 2 ]
print("a = ", a)
print("len =", len(a))
print("min =", min(a))
print("max =", max(a))
print("sum =", sum(a))
```

```
a =  [2, 3, 5, 2]
len = 4
min = 2
max = 5
sum = 12
```

Các Hàm và Toán Tử Thông Dụng

- **Chỉ số trong List** : dùng toán tử []

```
a = [2, 3, 5, 7, 11, 13]
print("a =", a)
print("a[0] =", a[0])
print("a[2] =", a[2])
# Chi so am
print("a[-1] =", a[-1])
print("a[-3] =", a[-3])
# mot khoang nam trong list
print("a[0:2] =", a[0:2])
print("a[1:4] =", a[1:4])
print("a[1:6:2] =", a[1:6:2])
```

```
a          = [2, 3, 5, 7, 11, 13]
a[0]       = 2
a[2]       = 5
a[-1]      = 13
a[-3]      = 7
a[0:2]     = [2, 3]
a[1:4]     = [3, 5, 7]
a[1:6:2]   = [3, 7, 13]
```

Các Hàm và Toán Tử Thông Dụng

- **List Aliases: Alias** là khả năng mà tại 1 ô nhớ có nhiều đối tượng cùng trỏ tới

```
# Tao mot list a
a = [ 2, 3, 5, 7 ]
# Tao mot bi danh den list a
b = a
# Co hai tham chieu cung mot list
a[0] = 42
b[1] = 99
print(a)
print(b)
```

```
[42, 99, 5, 7]
[42, 99, 5, 7]
```

```
a = [ 2, 3, 5, 7 ]
b = a
c = [ 2, 3, 5, 7 ]
print("initially:")
print(" a==b :", a==b)
print(" a==c :", a==c)
print(" a is b:", a is b)
print(" a is c:", a is c)
a[0] = 42
print("After changing a[0] to 42")
print(" a=",a)
print(" b=",b)
print(" c=",c)
print(" a==b :", a==b)
print(" a==c :", a==c)
print(" a is b:", a is b)
print(" a is c:", a is c)
```

Thông Dụng

```
  a==b  : True
  a==c  : True
  a is b: True
  a is c: False
After changing a[0] to 42
  a= [42, 3, 5, 7]
  b= [42, 3, 5, 7]
  c= [2, 3, 5, 7]
  a==b  : True
  a==c  : False
  a is b: True
  a is c: False
```

Các Hàm và Toán Tử Thông Dụng

- **Tìm phần tử trong list: in và not in**

```
a = [ 2, 3, 5, 2, 6, 2, 2, 7 ]
print("a =", a)
print("2 in a =", (2 in a))
print("4 in a =", (4 in a))
```

```
a = [ 2, 3, 5, 2, 6, 2, 2, 7 ]
print("a =", a)
print("2 not in a =", (2 not in a))
print("4 not in a =", (4 not in a))
```

```
a      = [2, 3, 5, 2, 6, 2, 2, 7]
2 in a = True
4 in a = False
```

```
a          = [2, 3, 5, 2, 6, 2, 2, 7]
2 not in a = False
4 not in a = True
```

Các Hàm và Toán Tử Thông Dụng

- **Đếm số lần xuất hiện: list.count(item)**

```
a = [ 2, 3, 5, 2, 6, 2, 2, 7 ]
print("a =", a)
print("a.count(1) =", a.count(1))
print("a.count(2) =", a.count(2))
print("a.count(3) =", a.count(3))
```

```
a          = [2, 3, 5, 2, 6, 2, 2, 7]
a.count(1) = 0
a.count(2) = 4
a.count(3) = 1
```

Các Hàm và Toán Tử Thông Dụng

- **Tìm chỉ số của một phần tử:**

list.index(item) và **list.index(item, start)**

```
a = [ 2, 3, 5, 2, 6, 2, 2, 7 ]
print("a =", a)
print("a.index(6) =", a.index(6))
print("a.index(2) =", a.index(2))
print("a.index(2,1) =", a.index(2,1))
print("a.index(2,4) =", a.index(2,4))
print("a.index(2,7) =", a.index(2,7))
```

```
a             = [2, 3, 5, 2, 6, 2, 2, 7]
a.index(6)    = 4
a.index(2)    = 0
a.index(2,1) = 3
a.index(2,4) = 5
Traceback (most recent call last):
ValueError: 2 is not in list
```

Các Hàm và Toán Tử Thông Dụng

- **Thêm phần tử hoặc một list vào list**: khi thêm sẽ thay đổi list hoặc tạo list mới.
- Thêm **một phần tử** dùng: **list.append(item)**

```
a = [ 2, 3 ]
a.append(7)
print(a)
```

```
[2, 3, 7]
```

- Thêm **một list vào một list**: **list += list2**

```
a = [ 2, 3 ]
a += [ 11, 13 ]
a += [0]
print(a)
```

```
[2, 3, 11, 13, 0]
```

Các Hàm và Toán Tử Thông Dụng

- Thêm **một list** dùng **list.extend(list2)**

```
a = [ 2, 3 ]
a.extend([ 17, 19 ])
print(a)
```

```
[2, 3, 17, 19]
```

- Thêm **một phần tử tại vị trí cho trước**: dùng insert()

```
a = [ 2, 3, 5, 7, 11 ]
a.insert(2, 42) # at index 2, insert 42
print(a)
```

```
[2, 3, 42, 5, 7, 11]
```

Các Hàm và Toán Tử Thông Dụng

- **Thêm phần tử hoặc list bằng cách tạo list mới.**

```
a = [ 2, 3 ]
b = a + [ 13, 17 ]
print(a)
print(b)
```

```
[2, 3]
[2, 3, 13, 17]
```

```
a = [ 2, 3 ]
b = a[:2] + [5] + a[2:]
print(a)
print(b)
```

```
[2, 3]
[2, 3, 5]
```

```
print("Destructive:")
a = [ 2, 3 ]
b = a
a += [ 4 ]
print(a)
print(b)
print("Non-Destructive:")
a = [ 2, 3 ]
b = a
a = a + [ 4 ]
print(a)
print(b)
```

```
Destructive:
[2, 3, 4]
[2, 3, 4]
Non-Destructive:
[2, 3, 4]
[2, 3]
```

Các Hàm và Toán Tử Thông Dụng

- **Thay đổi các phần tử trong list**

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print(letters)
# thay the
letters[2:5] = ['C', 'D', 'E']
print(letters)
# xoa
letters[2:5] = []
print(letters)
# xóa list bằng cách thay tất cả các phần tử bằng một list rỗng
letters[:] = []
print(letters)
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g']
['a', 'b', 'C', 'D', 'E', 'f', 'g']
['a', 'b', 'f', 'g']
[]
```

Các Hàm và Toán Tử Thông Dụng

- Xóa phần tử.
- Xóa **một phần tử** dùng **list.remove(item)**

```
a = [ 2, 3, 5, 3, 7, 6, 5, 11, 13 ]
print("a =", a)
a.remove(5)
print("After a.remove(5), a=", a)
a.remove(5)
print("After another a.remove(5), a=", a)
```

```
a = [2, 3, 5, 3, 7, 6, 5, 11, 13]
After a.remove(5), a= [2, 3, 3, 7, 6, 5, 11, 13]
After another a.remove(5), a= [2, 3, 3, 7, 6, 11, 13]
```

Các Hàm và Toán Tử Thông Dụng

- Xóa **một phần tử dùng chỉ số : list.pop(index)**

```
a = [ 2, 3, 4, 5, 6, 7, 8 ]
print("a =", a)
item = a.pop(3)
print("After item = a.pop(3)")
print(" item =", item)
print(" a =", a)
item = a.pop(3)
print("After another item = a.pop(3)")
print(" item =", item)
print(" a =", a)
# Xoa phan tu cuoi cung cua lít dung list.pop()
item = a.pop()
print("After item = a.pop()")
print(" item =", item)
print(" a =", a)
```

```
a = [2, 3, 4, 5, 6, 7, 8]
After item = a.pop(3)
    item = 5
    a = [2, 3, 4, 6, 7, 8]
After another item = a.pop(3)
    item = 6
    a = [2, 3, 4, 7, 8]
After item = a.pop()
    item = 8
    a = [2, 3, 4, 7]
```

Các Hàm và Toán Tử Thông Dụng

- Xóa **phần từ dùng toán tử del**

```
a = [ 2, 3, 4, 5, 6, 7, 8 ]
del a[2:4]
print("a =", a)
```

```
a = [2, 3, 6, 7, 8]
```

- **Phương thức clear() cũng được dùng để làm rỗng một list**

```
a = [ 2, 3, 4, 5, 6, 7, 8 ]
a.clear()
print("a =", a)
```

```
a = []
```

Các Hàm và Toán Tử Thông Dụng

- **Xóa phần từ dựa vào chỉ số bằng cách tạo list mới**

```
a = [ 2, 3, 5, 3, 7, 5, 11, 13 ]
print("a =", a)
b = a[:2] + a[3:]
print("After b = a[:2] + a[3:]")
print(" a =", a)
print(" b =", b)
```

```
a = [2, 3, 5, 3, 7, 5, 11, 13]
After b = a[:2] + a[3:]
    a = [2, 3, 5, 3, 7, 5, 11, 13]
    b = [2, 3, 3, 7, 5, 11, 13]
```

Các Hàm và Toán Tử Thông Dụng

- Hoán đổi các phần tử (swapping)
- **Failed swap**

```
a = [ 2, 3, 5, 7 ]
print("a =", a)
a[0] = a[1]
a[1] = a[0]
print("After failed swap of a[0] and a[1]:")
print(" a=",a)
```

```
a = [2, 3, 5, 7]
After failed swap of a[0] and a[1]:
   a= [3, 3, 5, 7]
```

Các Hàm và Toán Tử Thông Dụng

- **Swap dùng biến temp**

```
a = [ 2, 3, 5, 7 ]
print("a =", a)
temp = a[0]
a[0] = a[1]
a[1] = temp
print("After swapping a[0] and a[1]:")
print(" a=",a)
```

```
a = [2, 3, 5, 7]
After swapping a[0] and a[1]:
    a= [3, 2, 5, 7]
```

Các Hàm và Toán Tử Thông Dụng

- Swap dùng **parallel assignment**

```
a = [ 2, 3, 5, 7 ]
print("a =", a)
a[0],a[1] = a[1],a[0]
print("After swapping a[0] and a[1]:")
print(" a=",a)
```

```
a = [2, 3, 5, 7]
After swapping a[0] and a[1]:
    a= [3, 2, 5, 7]
```

Các Hàm và Toán Tử Thông Dụng

- Vòng lặp for trong list
- Vòng lặp **dùng phần tử** trong list: **for item in list**

```
a = [ 2, 3, 5, 7 ]
print("Here are the items in a:")
for pt in a:
      print(pt)
```

```
Here are the items in a:
2
3
5
7
```

Các Hàm và Toán Tử Thông Dụng

- Vòng lặp **dùng chỉ số** trong list : **for index in range(len(list))**

```
a = [ 2, 3, 5, 7 ]
print("Here are the items in a with their indexes:")
for i in range(len(a)):
      print("a[", i, "] =", a[i])
```

```
Here are the items in a with their indexes:
a[ 0 ] = 2
a[ 1 ] = 3
a[ 2 ] = 5
a[ 3 ] = 7
```

Các Hàm và Toán Tử Thông Dụng

- **Duyệt ngược** list **dùng chỉ số**

```
a = [ 2, 3, 5, 7 ]
print("And here are the items in reverse:")
for index in range(len(a)):
        revIndex = len(a)-1-index
        print("a[", revIndex, "] =", a[revIndex])
```

```
And here are the items in reverse:
a[ 3 ] = 7
a[ 2 ] = 5
a[ 1 ] = 3
a[ 0 ] = 2
```

- **Duyệt ngược** dùng hàm **reversed()**

```
a = [ 2, 3, 5, 7 ]
print("And here are the items in reverse:")
for item in reversed(a):
        print(item)
print(a)
```

```
And here are the items in reverse:
7
5
3
2
[2, 3, 5, 7]
```

Các Hàm và Toán Tử Thông Dụng

- So sánh lists :

```
# Create some lists
a = [ 2, 3, 5, 3, 7 ]
b = [ 2, 3, 5, 3, 7 ] # same as a
c = [ 2, 3, 5, 3, 8 ] # differs in last elem
d = [ 2, 3, 5 ] # prefix of a
print("a =", a)
print("b =", b)
print("c =", c)
print("d =", d)
print("------------------")
print("a == b", (a == b))
print("a == c", (a == c))
print("a != b", (a != b))
print("a != c", (a != c))
print("------------------")
print("a < c", (a < c))
print("a < d", (a < d))
```

```
a = [2, 3, 5, 3, 7]
b = [2, 3, 5, 3, 7]
c = [2, 3, 5, 3, 8]
d = [2, 3, 5]
------------------
a == b True
a == c False
a != b False
a != c True
------------------
a < c True
a < d False
```

```python
# Create some lists
a = [ 2, 3]
b = [ 2, 3]
c = [ 2, 4]
d = [ 2]
e = [ 3]
f = [ 2 , 3 ,5]
print("--------------------")
print("a < c", (a < c))
print("a < d", (a < d))
print("a <= b", (a <= b))
print("a < b", (a < b))
print("a < d", (a < d))
print("a < e", (a < e))
print("a < f", (a < f))
```

```
--------------------
a < c True
a < d False
a <= b True
a < b False
a < d False
a < e True
a < f True
```

Các Hàm và Toán Tử Thông Dụng

- **Copying list vs. List Aliasses**

```
import copy
# Create a list
a = [ 2, 3 ]
# Try to copy it
b = a # Error! Not a copy, but an alias
c = copy.copy(a) # Ok
print("At first...")
print(" a =", a)
print(" b =", b)
print(" c =", c)
# Now modify a[0]
a[0] = 42
print("But after a[0] = 42")
print(" a =", a)
print(" b =", b)
print(" c =", c)
```

```
At first...
    a = [2, 3]
    b = [2, 3]
    c = [2, 3]
But after a[0] = 42
    a = [42, 3]
    b = [42, 3]
    c = [2, 3]
```

Các Hàm và Toán Tử Thông Dụng

- **Các cách sao chép list**

```
import copy
a = [2, 3]
b = copy.copy(a)
c = a[:]
d = a + [ ]
e = list(a)
f = sorted(a)
a[0] = 42
print(a, b, c, d, e, f)
```

```
[42, 3] [2, 3] [2, 3] [2, 3] [2, 3] [2, 3]
```

Các Hàm và Toán Tử Thông Dụng

- Sắp xếp trên list
- **Sắp xếp và thay đổi list** dùng **list.sort() :**

iterable.sort(key=None, reverse=False)

```
a = [ 7, 2, 5, 3, 5, 11, 7 ]
b = list(a) # copy of a
print("At first, a =", a)
a.sort()
print("After a.sort(), a =",a)
############################
print("At first, b =", b)
b.sort(reverse=True)
print("After b.sort(reverse=True), b =",b)
```

```
At first, a = [7, 2, 5, 3, 5, 11, 7]
After a.sort(), a = [2, 3, 5, 5, 7, 7, 11]
At first, b = [7, 2, 5, 3, 5, 11, 7]
After b.sort(reverse=True), b = [11, 7, 7, 5, 5, 3, 2]
```

Các Hàm và Toán Tử Thông Dụng

- **Sắp xếp nhưng không thay đổi list** (tạo ra list mới) dùng **sorted(list)**

sorted(iterable, key=None, reverse=False)

```
a = [ 7, 2, 5, 3, 5, 11, 7 ]
print("At first")
print(" a =", a)
b = sorted(a)
c = sorted(a,reverse=True)
print("After b = sorted(a)")
print(" a =", a)
print(" b =", b)
print(" c =", c)
```

```
At first
   a = [7, 2, 5, 3, 5, 11, 7]
After b = sorted(a)
   a = [7, 2, 5, 3, 5, 11, 7]
   b = [2, 3, 5, 5, 7, 7, 11]
   c = [11, 7, 7, 5, 5, 3, 2]
```

Các Hàm và Toán Tử Thông Dụng

- Sắp xếp với key function
- **Dùng abs()**

```
a = [ 10, 2, -5, 8, -3, 7, 1 ]
print(sorted(a))

print(sorted(a, key=abs))
```

```
[-5, -3, 1, 2, 7, 8, 10]
[1, 2, -3, -5, 7, 8, 10]
```

Các Hàm và Toán Tử Thông Dụng

- **Sort dựa vào chiều dài của chuỗi**

```
a = [ 'a', 'ab', 'aab', 'ac', 'abccc']
print(sorted(a))
```

```
['a', 'aab', 'ab', 'abccc', 'ac']
```

```
####################
def mylensort(a):
        return len(a)

print(sorted(a, key=mylensort))
```

```
['a', 'ac', 'ab', 'aab', 'abccc']
```

Các Hàm và Toán Tử Thông Dụng

- So sanh thời gian và không gian bộ nhớ sử dụng giữa sort() và sorted()

https://viblo.asia/p/so-sanh-listsort-voi-sortedlist-trong-python-gDVK22MrKLj

Các Hàm và Toán Tử Thông Dụng

- **List và Function:** list được dùng như input của một hàm

```
def countOdds(a):
        count = 0
        for item in a:
                if (item % 2 == 1):
                        count += 1
        return count


print(countOdds([2, 3, 7, 8, 21, 23, 24]))
```

```
4
```

Các Hàm và Toán Tử Thông Dụng

- **Dùng hàm để thay đổi các giá trị của list**

```
def fill(a, value):
      for i in range(len(a)):
            a[i] = value
a = [1, 2, 3, 4, 5]
print("At first, a =", a)
fill(a, 42)
print("After fill(a, 42), a =", a)
```

```
At first, a = [1, 2, 3, 4, 5]
After fill(a, 42), a = [42, 42, 42, 42, 42]
```

Các Hàm và Toán Tử Thông Dụng

- **List comprehension (Cách tạo list mới ngắn gọn):** là một biểu thức đi kèm với lệnh for được đặt trong cặp dấu ngoặc vuông [].

```
cub3 = [3 ** x for x in range(9)]
# Output: [1, 3, 9, 27, 81, 243, 729, 2187, 6561]
print(cub3)
```

Code trên tương đương với:

```
cub3 = []
for x in range (9):
        cub3.append(3**x)
print(cub3)
```

Các Hàm và Toán Tử Thông Dụng

```
cub3 = [3 ** x for x in range(9) if x > 4]
 # Output: [243, 729, 2187, 6561]
print(cub3)

so_le = [x for x in range (18) if x % 2 == 1]
# Output: [1, 3, 5, 7, 9, 11, 13, 15, 17]
print(so_le)

noi_list = [x+y for x in ['Ngôn ngữ ','Lập trình '] for y in ['Python','C++']]
# Output: ['Ngôn ngữ Python', 'Ngôn ngữ C++', 'Lập trình Python', 'Lập trình C++']
print(noi_list)
```

Các Hàm và Toán Tử Thông Dụng

- **Converting Between Lists and Strings**

```
# use list(s) to convert a string to a list of characters
a = list("wahoo!")
print(a) # prints: ['w', 'a', 'h', 'o', 'o', '!']

a = "How are you doing today?".split(" ")
print(a) # prints ['How', 'are', 'you', 'doing', 'today?']

a = ["parsley", " ", "is", " ", "gharsley"]
s = "".join(a)
print(s) # prints: parsley is gharsley
```