# Bài 18: Các hàm xử lý list trong Python

**1, list().**

Hàm này có tác dụng chuyển đổi kiểu dữ liệu của một biến sang dạng list.

**Cú pháp**: list(data)

**Trong đó**, data là biến chứa tuple bạn cần chuyển đổi.

## 2, len().

## Hàm này trả về số lượng phần tử có trong list.

**Cú pháp**: len(list)

**Trong đó,**list là list mà các bạn cần đếm.

## 3, max().

Hàm này sẽ trả về phần tử có giá trị lớn nhất trong list. Nếu là chuỗi thì nó sẽ trả về phần tử có độ dài chuỗi dài nhất, nếu là số thì nó sẽ trả về phần tử có số lớn nhất.

**Cú pháp**: max(list)

## 4, min().

Hàm này sẽ trả về phần tử có giá trị nhỏ nhất trong list. Nếu là chuỗi thì nó sẽ trả về phần tử có độ dài chuỗi ngắn nhất, nếu là số thì nó sẽ trả về phần tử có số nhỏ nhất.

**Cú pháp**: min(list)

## 5, append().

Phương thức này có tác dụng thêm phần vào cuối của một list.

**Cú pháp**: mylist.append(obj)

**Trong đó**:

* mylist là list mà các bạn cần thêm phần tử.
* obj là phần tử mà bạn muốn thêm vào mylist.

## 6, extend().

## **Cú pháp**: list1.extend(list2)

**Trong đó**:

* list1 là list mà bạn muốn kế thừa từ một list khác (ở đây là list2).
* list2 là list được sử dụng để cho list khác kết thừa (ở đây là list1).
* VD:
* List=[‘A’,’B’]
* List.extend(‘D’)
* Print(List)
* # [‘A’,’B’,’D’]

## 7, count().

Phương thức này có tác dụng đếm số lần xuất hiện của một thành phần trong list!

**Cú pháp**: mylist.count(val)

**Trong đó**:

* mylist là list mà các bạn cần kiểm tra.

## 8, index().

Phương thức này có tác dụng trả về index xuất hiện đầu tiên của phần tử mà bạn muốn tìm và nếu như không tìm thấy thì nó sẽ gọi [exception](https://toidicode.com/exception-trong-python-356.html).

**Cú Pháp**: mylist.index(val)

* val là phần tử mà bạn muốn tìm và đếm trong list mylist.

## 9, insert().

## Phương thức có tác dụng thêm phần tử vào vị trí index của list, và các phần tử sau index đó sẽ được đẩy về phía sau.

**Trong đó**:

**Cú pháp**: mylist.insert(index, val)

* mylist là list mà các bạn cần thêm.
* index là vị trí mà bạn muốn thêm phần tử val vào.
* val là phần tử mà bạn muốn thêm vào trong list mylist.

## 10, reverse().

Phương thức này có tác dụng đảo ngược vị trí của các phần tử trong list.

**Cú pháp**: mylist.reverse()

## 11, remove().

Phương thức này có tác dụng xóa phần tử khỏi list.

**Cú Pháp**: mylist.remove(val)

**Trong đó**:

* mylist là list mà các bạn cần xóa phần tử.
* val là phần tử mà bạn muốn muốn xóa trong list mylist.

## 12, pop().

Phương thức này có tác dụng xóa bỏ phần tử trong list dựa trên index của nó.

**Cú pháp**: mylist.pop(index)

**Trong đó**:

* mylist là list mà các bạn cần xóa phần tử.
* index là index của phần tử mà bạn muốn muốn xóa trong list mylist. Mặc định thì index = mylist[-1] (phần tử cuối cùng trong list).

## 13, sort().

Phương thức này có tác dụng sắp xếp lại các phần tử trong list theo một thứ tự xác định.

**Cú pháp**: mylist.sort(reverse, key)

* mylist là list mà các bạn muốn sắp xếp.
* reverse là một boolean cấu hình kiểu sắp xếp. Nếu reverse = True thì list sẽ được sắp xếp từ lớn đến bé, nếu reverse = False thì list sẽ được sắp xếp theo thứ tự từ bé đến lớn. Mặc định thì reverse = False.

list = ['A', 'C', 'B', 'E', 'D']

list.sort()

print(list)

# Kết quả: ['A', 'B', 'C', 'D', 'E']

list.sort(reverse=True)

print(list)

# Kết quả: ['E', 'D', 'C', 'B', 'A']

## 14, clear().

Phương thức này có tác dụng xóa bỏ hết tất cả các phần tử trong list.

**Cú pháp**: mylist.clear()

15. cmp

**cmp(list1, list2)**: so sánh các phần tử của 2 list

**16. sum()**

**sum()**: Trả về tổng giá trị của các phần tử trong list. Hàm này chỉ làm việc với kiểu number

**CHUYỂN ĐỔI LIST THÀNH STRING**

## 1. Chuyển list chỉ chứa string thành string trong Python | phương thức join() Python

mylist = ["a", "b", "c"]  
mystring = "".join(mylist)  
print(mystring)  
#>> abc

## 2. Chuyển list chứa số thành string trong Python | Phương thức str() Python

## Th1:

mylist = ["a", "b", "c", 1]  
**mystring = "".join([str(char) for char in mylist])**print(mystring)  
#>> abc1

**th2:**

mylist = [1, 2, 3, 4]  
**mystring = "".join([str(char) for char in mylist])**print(mystring)  
#>> 1234

**3. Để loại bỏ ký tự đặc biệt trong chuỗi Python , chúng ta sẽ sử dụng tới RegEx trong Python với hàm re.sub() như sau:**

import re  
mylist = [1, ['a', 'b'], 3, 4, (5,6)]  
mystring = "".join([str(char) for char in mylist])  
print(mystring)  
#>> 1['a', 'b']34(5, 6)  
  
new\_string = re.sub(r"[^a-zA-Z0-9]","",mystring)  
print(new\_string)  
#>> 1ab3456

## 4. Chuyển list thành string trong Python | Hàm map() Python

Chúng ta cũng có thể sử dụng hàm map() để **chuyển list sang trong Python**. Hàm map() trong Python sẽ chuyển các phần tử trong list chỉ định sang dạng chuỗi và tạo ra một trình lặp. Sau đó chúng ta cũng sử dụng phương thức join() để lấy ra và nối các chuỗi là phần tử của trình lặp này để tạo ra chuỗi string kết quả.

mylist = ["a", "b", "c"]  
mystring = "".join(map(str, mylist))  
print(mystring)  
#>> 1234

# Bài 17: Các hàm xử lý chuỗi trong Python

**1, Capitalize().**

Hàm này có tác dụng in hoa chữ cái đầu tiên của chuỗi.

**2, Center().**

Hàm này có tác dụng trả về chuỗi được hiển thị ở giữa một chuỗi.

**Cú Pháp**: string.center(len, char)

VD:

ststring = "toidicode.com"

print(string.center(20));

# Kết quả: Toidicode.com

print(string.center(20, '\*'));

# Kết quả:\*\*\*toidicode.com\*\*\*\* ring.center(len, char)

**3, Count().**

**Cú Pháp**:

string.count(sub, start, end)

**Trong đó**:

* sub là chuỗi các bạn cần tìm kiếm và đếm.
* start là index bắt của chuỗi cần tìm. Mặc định thì strart = 0.
* end là index kết thúc của chuỗi cần tìm.  Mặc định thì end = len() của chuỗi.

**4, encode().**

Hàm này có tác dụng encode (mã hóa) một chuỗi.

* **Cú pháp**: string.encode(type, mode)

**5, decode().**

* Hàm này có tác dụng decode (gải mã) chuỗi trông Python.
* **Cú Pháp**: string.decode(type, mode)

**6, endswith().**

* Hàm này có tác dụng kiểm tra xem chuỗi hoặc khoảng chuỗi có được kết thúc bằng ký tự nào đó hay không. Nó sẽ trả về True nếu đúng và False nếu sai.
* **Cú pháp**:

string.endswith(str, start, end)

**Trong đó**:

* str là chuỗi các bạn cần xác thực xem có phải chuỗi kết thúc không.
* strart là index bắt đầu chuỗi cần so sánh. Mặc định thì start = 0.
* end là index kết thúc chuỗi cần so sánh.  Mặc định thì end = len().

**7, expandtabs().**

* Hàm này có tác dụng tìm kiếm thay thế \t bằng các ký tự khoảng trắng.
* **Cú Pháp**: string.expandtabs(len)

**Trong đó**: len là số lượng khoảng trắng mà bạn muốn thay thế cho một \t. Mặc định thì len = 8.

**8, find().**

Hàm này có tác dụng tìm kiếm một chuỗi trong một chuỗi hoặc khoảng chuỗi. Nó sẽ trả về là vị trí bắt đầu của chuỗi tìm được trong chuỗi nếu tìm thấy và nếu không tìm thấy nó sẽ trả về  -1.

**Cú pháp**: string.find(str, start, end)

**Trong đó**:

* str là chuỗi các bạn cần xác thực xem có phải chuỗi kết thúc không.
* strart là index bắt đầu chuỗi cần so sánh. Mặc định thì start = 0.
* end là index kết thúc chuỗi cần so sánh.  Mặc định thì end = len().

**9, index().**

* Hàm này tương tự như hàm find() chỉ khác duy nhất là nếu như không tìm thấy thì hàm này sẽ gọi exception.

**10, isalnum().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải là chứa duy nhất các ký tự chữ hoặc chuỗi hay không? Nó sẽ trả về True nếu chuỗi chỉ chứa các ký tự chữ hoặc số. Và ngược lại nó sẽ trả về False nếu chuỗi chứa ký tự khác chuỗi và số.

**11, isalpha().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải là chứa duy nhất các ký tự chữ hay không? Nó sẽ trả về True nếu chuỗi này chỉ chứa duy các ký tự chữ trong bảng chữ cái, và sẽ trả về False nếu nó chứa số hoặc ký tự đặc biệt khác.

**12, isdigit().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải là chứa duy nhất các chữ số hay không? Nó sẽ trả về True nếu đúng và False nếu sai.

**13, islower().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải là in thường hay không? Nó sẽ trả về True nếu đúng và False nếu sai.

**14, isupper().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải là in Hoa hay không? Nó sẽ trả về True nếu đúng và False nếu sai.

**15, isnumeric()**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải chỉ chứa duy nhất các ký tự số hay không? Nó sẽ trả về True nếu đúng và False nếu sai.

**16, isspace().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải chỉ chứa duy nhất các ký tự khoảng trắng không? Nó sẽ trả về True nếu đúng và False nếu sai.

**17, istitle().**

* Hàm này có tác dụng kiểm tra xem một chuỗi có phải là title hay không, chuỗi title là chuỗi có các chữ cái đầu đều được in hoa. Nó sẽ trả về True nếu đúng và ngược lại False nếu sai.

**18, join().**

* Hàm này có tác dụng join squence bởi string.
* **Cú pháp**: string.join(squence)

**Trong đó**: squence là string, list,... mà bạn cần join lại với nhau bởi chuỗi string.

**19, len().**

Hàm này có tác dụng trả về độ dài của chuỗi.

**20, ljust().**

Hàm này có tác dụng trả về một chuỗi với độ dài length được xác định, nếu chuỗi được chọn nhỏ hơn width thì nó sẽ sử dụng char để bù chỗ thiếu đó về phía bên phải của chuỗi.

string.ljust(length, char)

**Trong đó**:

* length là độ dài của chuỗi mới cần in ra.
* char là ký tự sẽ bù vào chuỗi mới nếu chuỗi cũ không đủ length. Mặc định thì char = khoảng trắng.
* VD:
* string = "Vu Thanh Tai"
* print(string.ljust(17, "-"))
* # Kết quả: Vu Thanh Tai-----

**21, rjust().**

* Tương tự hàm ljust() nhưng chỉ có điều là nó sẽ bù về phía bên trái của chuỗi.

**22, lower().**

* Hàm này có tác dụng chuyển đổi chuỗi về dạng in thường.

**23, upper().**

* Hàm này có tác dụng chuyển đổi chuỗi sang dạng in hoa.

**24, lstrip().**

* Hàm này có tác dụng loại bỏ đi các ký tự char ở phía đầu của chuỗi

**Cú Pháp**: string.lstrip(char)

**Trong đó**: char là ký tự bạn muốn loại bỏ. Mặc định thì char sẽ bằng khoảng trắng (white space).

**string = " Vu Thanh Tai"**

**print(string.lstrip())**

**# Kết quả: Vu Thanh Tai**

**string = "----Vu Thanh Tai"**

**print(string.lstrip('-'))**

**# Kết quả: Vu Thanh Tai**

**25, rstrip().**

Tương tự như lstrip(), chỉ khác là rstrip nó sẽ loại bỏ char ở phần cuối của chuỗi.

string = "Vu Thanh Tai "

**print(string.rstrip())**

# Kết quả: Vu Thanh Tai

string = "Vu Thanh Tai----"

**print(string.rstrip('-'))**

# Kết quả: Vu Thanh Tai

**26, strip().**

Hàm này là sự kết hợp của lstrip() và rstrip(). Nó sẽ lại bỏ các ký tự char ở cả hai đầu của chuỗi.

string = " Vu Thanh Tai "

**print(string.strip())**

# Kết quả: Vu Thanh Tai

string = "----Vu Thanh Tai----"

**print(string.strip('-'))**

# Kết quả: Vu Thanh Tai

**27, rfind().**

Tương tự như hàm find(), nhưng hàm này nó sẽ trả về index của chuỗi cuối cùng tìm được trong chuỗi. Cú pháp sử dụng tương tự hàm find().

string = "Vu Thanh Tai"

**print(string.rfind('T'))**

# Kết quả: 9

**28, rindex().**

Tương tự như hàm index(),nhưng hàm này nó sẽ trả về index của chuỗi cuối cùng tìm được trong chuỗi. Cú pháp sử dụng tương tự hàm index().

string = "Vu Thanh Tai"

**print(string.rindex('T'))**

## # Kết quả: 9

**29, replace().**

Hàm này có tác dụng tìm kiếm và thay thế chuỗi tìm được bằng chuỗi mới.

**Cú Pháp**: string.replace(old,new,max)

**Trong đó**:

* old là chuỗi mà bạn cần tìm kiếm trong string.
* new là chuỗi mà bạn cần thay thế cho chuỗi old tìm được.
* max là số lượng từ có thể thay thế tối đa.

**VD**:

**30, max().**

Hàm này trả về chữ cái có độ sắp xếp cuối cùng theo bảng chữ cái alphabet nằm trong chuỗi.

string = "Vu Thanh Tai"

**print(max(string))**

# Kết quả: u

**31, min().**

Hàm này trả về chữ cái có độ sắp xếp đầu tiên theo bảng chữ cái alphabet nằm trong chuỗi.

string = "Vu Thanh Tai"

**print(min(string))**

# Kết quả: a

**33, swapcase().**

Hàm này có tác dụng chuyển đổi chuỗi sang dạng nghịch đảo của nó (nghịch đảo ở đây là hoa - thường).

string = "vu thanh tai"

**print(string.swapcase())**

# Kết quả: VU THANH TAI

string = "Vu Thanh Tai"

**print(string.swapcase())**

# Kết quả: vU tHANH tAI

**34, zfill().**

Hàm này có tác dụng như hàm ljust() , nhưng nó sẽ chỉ thêm được các ký tự zero (số 0) và trước chuỗi thôi.

**VD**:

string = "vu thanh tai"

**print(string.zfill(17))**

# Kết quả: 00000vu thanh tai

**35, isdecimal().**

Hàm này có tác dụng **gần như** hàm isdigit(), nó sẽ trả về True nếu chuỗi cần kiểm tra chỉ chứa các số thập phân, và ngược lại....

**36, split().**

Hàm này có tác dụng tác chuỗi thành mảng bởi các char.

**Cú Pháp**: string.split(char, max)

string = "Vu Thanh Tai"

**print(string.split())**

# Kết quả: ['Vu', 'Thanh', 'Tai']

string = "Vu Thanh Tai"

**print(string.split('a'))**

# Kết quả: ['Vu Th', 'nh T', 'i']

string = "Vu Thanh Tai"

**37, splitlines().**

Hàm này sẽ tách chuỗi bởi các ký tự \n.

**Cú pháp**: string.splitlines(max)

Vd:

string = "Vu\nThanh\nTai"

print(string.splitlines())

# Kết quả: ['Vu', 'Thanh', 'Tai']

**38, startswith().**

Hàm này có tác dụng kiểm tra xem chuỗi hoặc khoảng chuỗi có được bắt đầu bằng ký tự nào đó hay không. Nó sẽ trả về True nếu đúng và False nếu sai.

**Cú pháp**:

## string.startswith(str, start, end)

**rong đó**:

* str là chuỗi các bạn cần xác thực xem có phải chuỗi bắt đầu không.
* strart là index bắt đầu chuỗi cần so sánh. Mặc định thì start = 0.
* end là index kết thúc chuỗi cần so sánh.  Mặc định thì end = len().

string = 'toidicode.com'

print(string.startswith('t'));

# Kết quả: True

print(string.startswith('m', 3, 10));

# Kết quả: False

**39, maketrans().**

Hàm này có tác dụng tạo ra các translation cho chuỗi. Dùng kết hợp với phương thức translate().

**Cú Pháp**:

string.maketrans(in, out)

* in là chuỗi các ký tự các bạn cần tìm.
* out là chuỗi chứa các ký tự các bạn cần thay thế.

**40, translate().**

* Hàm này có tác dụng thực thi việc dịch chuỗi. Dùng kết hợp với phương thức **makestrans()**.
* VD:

inputs = "abcdefghijklmnopqrstuxyz";

outputs = "ABCDEFGHIJKLMNOPQRSTUXYZ";

string = "Vu Thanh Tai";

trans = string.maketrans(inputs, outputs)

print(string.translate(trans))

# Kết quả: VU THANH TAI