## **1, Capitalize().**

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

VD:

python

*copy*

string = "toidicode.com"

print(string.capitalize());

# Kết quả: Toidicode.com

## **2, Center().**

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

Cú Pháp:

python

*copy*

string.center(len, char)

Trong đó:

* string là chuỗi các bạn cần chuyển đổi.
* len là số lượng ký tự của chuỗi mới.
* char là ký tự sẽ hiển thị ở 2 bên chuỗi cũ. Mặc định nó sẽ là khoảng trắng.

VD: Nếu khó hiểu thì mời mọi người cùng xem qua ví dụ sau:

python

*copy*

string = "toidicode.com"

print(string.center(20));

# Kết quả: Toidicode.com

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

# Kết quả:\*\*\*toidicode.com\*\*\*\*

Chú ý: Nếu len nhỏ hơn độ dài chuỗi cần xử lý thì, hàm này sẽ trả về chuỗi ban đầu.

## **3, Count().**

Hàm này có tác dụng đếm xem trong chuỗi có bao nhiêu ký tự cần tìm.

Cú Pháp:

python

*copy*

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.

VD:

python

*copy*

string = "toidicode.com"

print(string.count('i'));

# Kết quả: 2

print(string.count('i', 3));

# Kết quả: 1

## **4, encode().**

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

Cú pháp:

python

*copy*

string.encode(type, mode)

Trong đó:

* type là kiểu encode của string. Mặc định sẽ là utf-8
* mode là chế độ báo lỗi nếu có khi encode. Python hỗ trợ 6 dạng mode như sau:
  + strict - Chế độ nghiêm ngặt, nó sẽ hiển thị lỗi dưới UnicodeDecodeError exception. Đây là chế độ mặc định.
  + ignore - bỏ qua tất cả các lỗi nếu có.
  + replace - nó sẽ thay thế lỗi bằng dấu ?.
  + xmlcharrefreplace - chèn tham chiếu XML.
  + backslashreplace - Chèn chuỗi \uNNNN.
  + namereplace - Chèn chuỗi \N{...}.

VD:

python

*copy*

string = "toidicode.com"

print(string.encode());

# Kết quả: b'toidicode.com'

## **5, decode().**

Hàm này có tác dụng decode (gải mã) chuỗi trông Python.

Cú Pháp:

python

*copy*

string.decode(type, mode)

Về phần type và mode thì hoàn toàn giống ở encode nhé mọi người!

VD:

python

*copy*

string = b'toidicode.com'

print(string.decode());

# Kết quả: toidicode.com

## **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:

python

*copy*

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().

VD:

python

*copy*

string = 'toidicode.com'

print(string.endswith('m'));

# Kết quả: True

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

# Kết quả: False

## **7, expandtabs().**

Hàm này có tác dụng tìm kiếm thay thế \t bằng các ký tự khoảng trắng.

Cú Pháp:

python

*copy*

string.expandtabs(len)

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

VD:

python

*copy*

string = 'toidicode.com\thoc lap trinh'

print(string.expandtabs());

# Kết quả: toidicode.com hoc lap trinh

## **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:

sql

*copy*

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().

VD:

python

*copy*

string = 'toidicode.com'

print(string.find('di'));

# Kết quả: 3

## **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.

VD:

python

*copy*

string = 'toidicode.com'

print(string.index('vuthanhtai'));

# Kết quả: ValueError: substring not found

## **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.

VD:

python

*copy*

string = 'toidicode'

print(string.isalnum());

# Kết quả: True

string = 'toidicode.com'

print(string.isalnum());

# Kết quả: False

## **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.

VD:

python

*copy*

string = 'toidicode96'

print(string.isalpha());

# Kết quả: False

string = 'toidicodecom'

print(string.isalpha());

# Kết quả: True

## **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.

VD:

python

*copy*

string = 'toidicode96'

print(string.isdigit());

# Kết quả: False

string = '12051996'

print(string.isdigit());

# Kết quả: True

## **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.

VD:

python

*copy*

string = 'toidicode.com'

print(string.islower());

# Kết quả: True

string = '12051996'

print(string.islower());

# Kết quả: False

string = '9toidicode.com6'

print(string.islower());

# Kết quả: True

string = '9Toidicode.com6'

print(string.islower());

# Kết quả: False

## **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.

VD:

python

*copy*

string = 'TOIDICODE.COM'

print(string.isupper());

# Kết quả: True

string = '12051996'

print(string.isupper());

# Kết quả: False

string = '9TOIDICODE.COM6'

print(string.isupper());

# Kết quả: True

string = '9Toidicode.com6'

print(string.isupper());

# Kết quả: False

## **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.

VD:

python

*copy*

string = 'TOIDICODE.COM'

print(string.isnumeric());

# Kết quả: False

string = '12051996'

print(string.isnumeric());

# Kết quả: True

string = '9TOIDICODE.COM6'

print(string.isnumeric());

# Kết quả: False

string = '9Toidicode.com6'

print(string.isnumeric());

# Kết quả: False

## **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.

VD:

python

*copy*

string = ' '

print (string.isspace());

# Kết quả: True

string = 'Vu Thanh Tai'

print (string.isspace());

# Kết quả: False

## **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.

VD:

python

*copy*

string = 'vu thanh Tai'

print(string.istitle())

# Kết quả: False

string = 'Vu Thanh Tai'

print(string.istitle())

# Kết quả: True

## **18, join().**

Hàm này có tác dụng join squence bởi string.

Cú pháp:

python

*copy*

string.join(squence)

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

VD:

python

*copy*

string\_one = ' '

string\_two = 'TAI'

print(string\_one.join(string\_two))

# Kết quả: T A I

string\_one = '-'

string\_two = ['T','D','C',]

print(string\_one.join(string\_two))

# Kết quả: T-D-C

## **19, len().**

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

VD:

python

*copy*

string = "Vu Thanh Tai"

print(len(string))

# Kết quả: 12

## **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.

python

*copy*

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:

python

*copy*

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.

VD:

python

*copy*

string = "Vu Thanh Tai"

print(string.rjust(17, "-"))

# Kết quả: -----Vu Thanh Tai

## **22, lower().**

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

VD:

python

*copy*

string = "Vu Thanh Tai"

print(string.lower())

# Kết quả: vu thanh tai

## **23, upper().**

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

VD:

python

*copy*

string = "Vu Thanh Tai"

print(string.upper())

# Kết quả: VU THANH TAI

## **24, lstrip().**

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

Cú Pháp:

python

*copy*

string.lstrip(char)

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

VD:

python

*copy*

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.

VD:

python

*copy*

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.

VD:

python

*copy*

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().

VD:

python

*copy*

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().

VD:

csharp

*copy*

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:

python

*copy*

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:

python

*copy*

string = "Chao \*!"

print(string.replace('\*', 'Tai'))

# Kết quả: Chao Tai!

string = "A A A"

print(string.replace('A', 'Tai', 2))

# Kết quả: Tai Tai A

## **30, max().**

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

VD:

python

*copy*

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.

VD:

python

*copy*

string = "vuthanhtai"

print(min(string))

# Kết quả: a

## **32, title().**

Hàm này có tác dụng chuyển đổi chuỗi sang kiểu title (xem ở trên).

VD:

python

*copy*

string = "vu thanh tai"

print(string.title())

# Kết quả: Vu Thanh Tai

## **33, swapcase().**

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

VD:

python

*copy*

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:

python

*copy*

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....

VD:

python

*copy*

string = "Vu Thanh Tai 96"

print(string.isdecimal())

# Kết quả: False

string = "12051996"

print(string.isdecimal())

# Kết quả: True

## **36, split().**

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

Cú Pháp:

python

*copy*

string.split(char, max)

Trong đó:

* char là ký tự các bạn tìm và tách chuỗi bởi nó. Mặc định thì char = khoảng trắng.
* max là số lượng chuỗi tách tối đa.

VD:

python

*copy*

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"

print(string.split(' ', 1))

# Kết quả: ['Vu', 'Thanh Tai']

## **37, splitlines().**

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

Cú pháp:

python

*copy*

string.splitlines(max)

Trong đó: max là số lần có thể cắt tối đa.

VD:

python

*copy*

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:

python

*copy*

string.startswith(str, start, end)

Trong 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().

VD:

python

*copy*

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:

python

*copy*

string.maketrans(in, out)

Trong đó:

* 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:

python

*copy*

inputs = "abcdefghijklmnopqrstuxyz";

outputs = "ABCDEFGHIJKLMNOPQRSTUXYZ";

string = "Vu Thanh Tai";

trans = string.maketrans(inputs, outputs)

print(string.translate(trans))

# Kết quả: VU THANH TAI