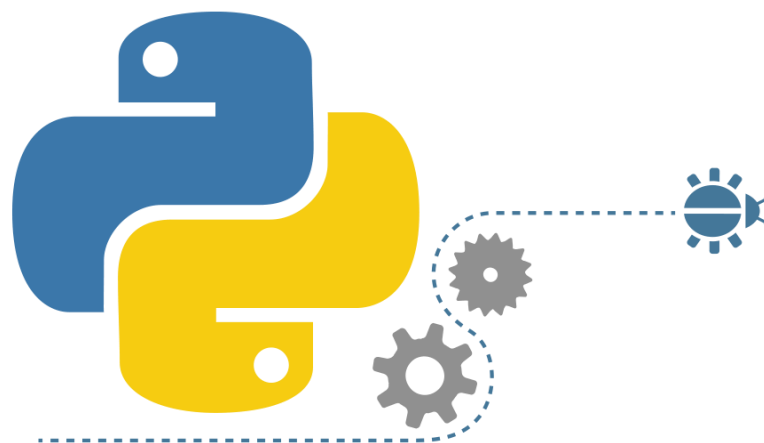


Khái niệm và cấu trúc của chuỗi



Nội dung bài học

Chuỗi là tập các ký tự nằm trong nháy đơn hoặc nháy đôi, hoặc 3 nháy đơn hoặc 3 nháy đôi. Chuỗi rất quan trọng trong mọi ngôn ngữ, hầu hết ta đều gặp xử lý chuỗi

```
1 s1='Hello Simon'
2 s2="Hello David"
3 s3=""
4 Quanh năm buôn bán ở mom sông
5 nuôi đủ năm con với một chồng
6 lặn lội thân cò khi quãng vắng
7 eo sèo mặt nước buổi đò đông
8 ""
9 s4='''Cha mẹ thói đời ăn ở bạc
10 có chồng hờ hững cũng như không'''
11 print(s1)
12 print(s2)
13 print(s3)
14 print(s4)
```



```
Hello Simon
Hello David

Quanh năm buôn bán ở mom sông
nuôi đủ năm con với một chồng
lặn lội thân cò khi quãng vắng
eo sèo mặt nước buổi đò đông

Cha mẹ thói đời ăn ở bạc
có chồng hờ hững cũng như không
```

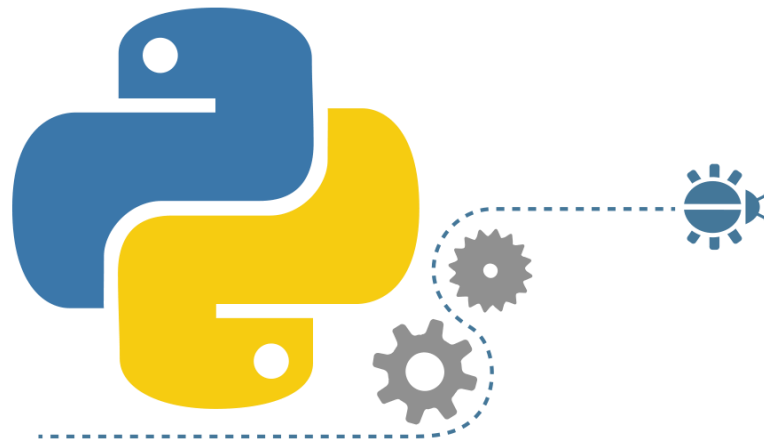
Nội dung bài học

Chuỗi trong Python cũng là đối tượng, nó cung cấp một số hàm rất quan trọng:

object . *method name* (*parameter list*)

Tên hàm	Mô tả
upper, lower	Xử lý in Hoa, in thường
rjust	Căn lề phải
ljust	Căn lề trái
center	Căn giữa
strip	Xóa khoảng trắng dư thừa
startswith	Kiểm tra Chuỗi có phải bắt đầu là ký tự ?
endswith	Kiểm tra Chuỗi có phải kết thúc là ký tự ?
count	Đếm số lần xuất hiện trong Chuỗi
find	Tìm kiếm Chuỗi con
format	Định dạng Chuỗi
__len__()	Trả về số lượng ký tự trong chuỗi, dùng index để lấy ký tự ra: str[index]

Hàm upper, lower -in HOA-thường



Nội dung bài học

Hàm upper → đưa Chuỗi về In HOA

Hàm lower → đưa Chuỗi về In thường

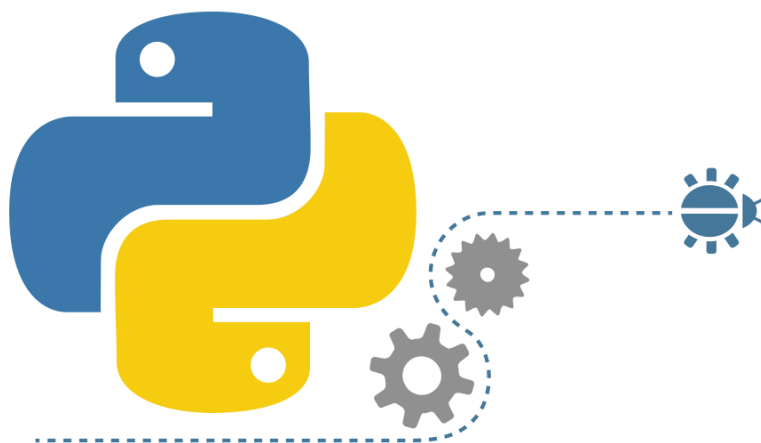
```
In [19]: name = "Nguyễn Tuấn Anh"  
         name.lower()
```

```
Out[19]: 'nguyễn tuấn anh'
```

```
In [20]: name = "Nguyễn Tuấn Anh"  
         name.upper()
```

```
Out[20]: 'NGUYỄN TUẤN ANH'
```

Hàm căn lề rjust, ljust, center



Nội dung bài học

Hàm rjust → căn lề phải

Hàm ljust → căn lề trái

Hàm center → căn giữa

Rjust

Hàm rjust sẽ căn phải Chuỗi, nếu truyền 1 đối số Python sẽ chèn khoảng trắng, nếu có đối số thứ 2 thì chèn nó vào trước.

```
word = "ABCD"  
print(word.rjust(10, "*"))  
print(word.rjust(3, "*"))  
print(word.rjust(15, ">"))  
print(word.rjust(10))
```



```
*****ABCD  
ABCD  
>>>>>>>>>>ABCD  
          ABCD
```

Lưu ý nếu số ký tự chèn nhỏ hơn chuỗi gốc thì không có gì thay đổi (trường hợp `rjust(3, "*")`)

ljust

Hàm ljust sẽ căn trái Chuỗi, nếu truyền 1 đối số Python sẽ chèn khoảng trắng đằng sau, nếu có đối số thứ 2 thì chèn nó vào sau.

```
word="OBAMA"  
print(word.ljust(1))  
print(word.ljust(2))  
print(word.ljust(3))  
print(word.ljust(4))  
print(word.ljust(5))  
print(word.ljust(10))  
print(word.ljust(10, '*'))
```



```
OBAMA  
OBAMA  
OBAMA  
OBAMA  
OBAMA  
OBAMA  
OBAMA*****
```

Lưu ý nếu số ký tự muốn chèn nhỏ hơn Chuỗi gốc thì không có gì thay đổi

center

Hàm center căn giữa Chuỗi, nó tự đẩy khoảng trắng 2 bên sao cho tổng ký tự bằng giá trị muốn truyền vào. Nếu có đối số thứ 2 thì thay khoảng trắng bằng ký tự mới này

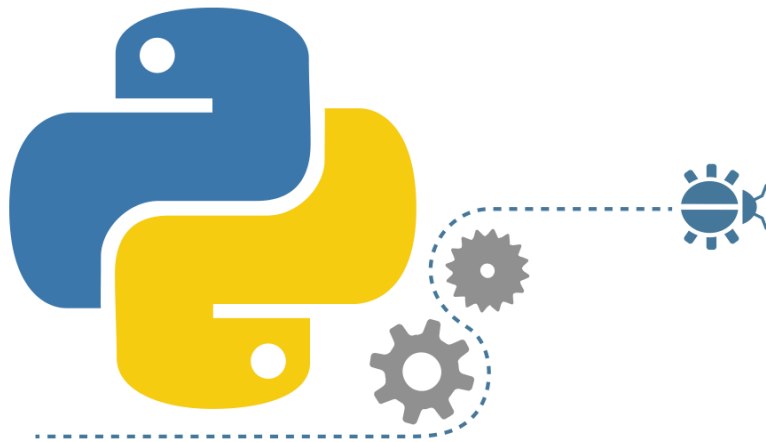
```
word="TRUMP"  
print(word.center(10))  
print(word.center(10, '*'))  
print(word.center(21, '*'))
```



```
TRUMP  
**TRUMP**  
*****TRUMP*****
```

Lưu ý: Nếu số lượng căn giữa mà nhỏ hơn số ký tự gốc thì không có gì thay đổi.

Hàm xóa khoảng trắng dư thừa `strip`



Nội dung bài học

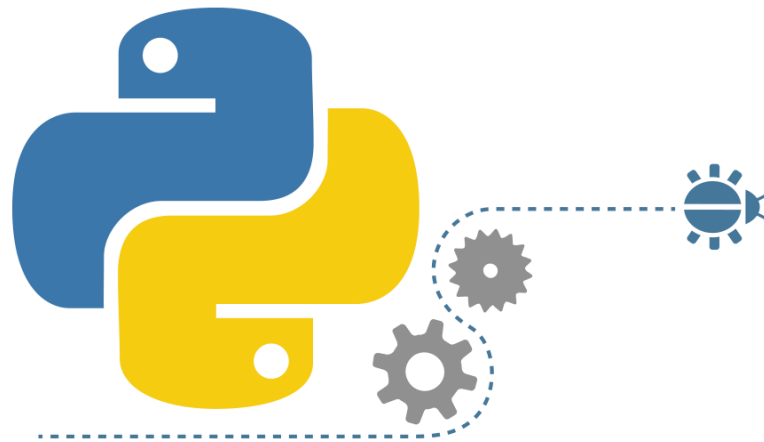
Để xóa khoảng trắng dư thừa, Python hỗ trợ hàm `strip`

```
s = " ABCDEFGHBCDIJKLMNOPQRSBCDTUVWXYZ "  
print(s)  
print(s.__len__())  
s=s.strip()  
print(s)  
print(s.__len__())
```



```
ABCDEFGHIHBCDIJKLMNOPQRSBCDTUVWXYZ  
34  
ABCDEFGHIHBCDIJKLMNOPQRSBCDTUVWXYZ  
32
```

Hàm `startsWith`, `endsWith`



Nội dung bài học

startswith để kiểm tra Chuỗi có bắt đầu bằng 1 chuỗi con nào đó hay không

endswith để kiểm tra Chuỗi có kết thúc bằng 1 chuỗi con nào đó hay không

```
s="#hello Python*"
```

```
print(s.startswith("#"))
```

```
print(s.startswith("*"))
```

```
print(s.endswith("#"))
```

```
print(s.endswith("*"))
```



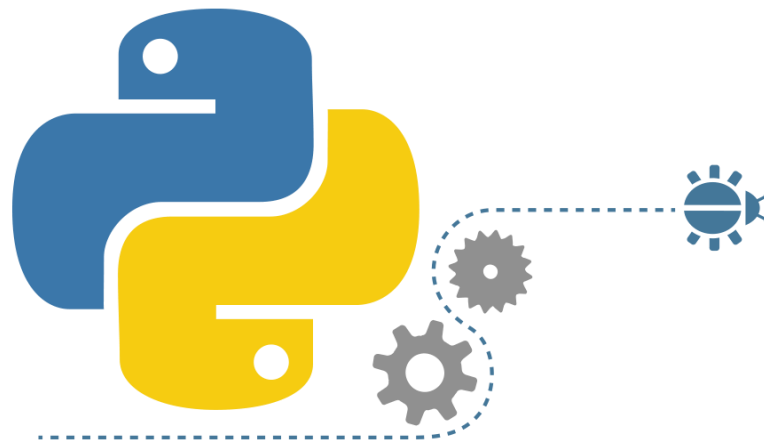
```
True
```

```
False
```

```
False
```

```
True
```

Hàm find, count



Nội dung bài học

Hàm **find** trả về vị trí đầu tiên tìm thấy, hàm **rfind** trả về vị trí cuối cùng tìm thấy. Nếu không thấy sẽ trả về -1

```
s="hello hello hello"  
x1=s.find('o')  
print(x1)  
x2=s.rfind('o')  
print(x2)  
x3=s.find('x')  
print(x3)
```




```
4  
16  
-1
```


Nội dung bài học

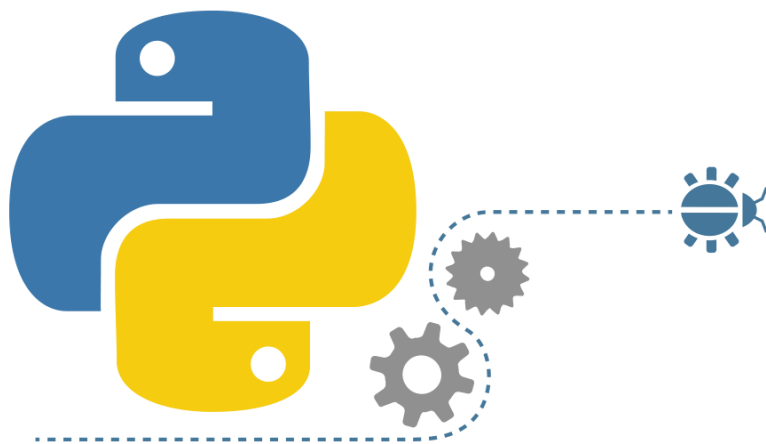
Hàm **count** trả về số lần xuất hiện của Chuỗi con trong Chuỗi gốc, không tồn tại trả về 0

```
s="Obama likes Putin, Putin likes Kim Jong Un"  
c1=s.count("Putin")  
print(c1)  
c2=s.count("Trump")  
print(c2)
```



3	2
4	0

Hàm format, substring



Nội dung bài học

Hàm format sử dụng {} để dành chỗ xuất dữ liệu

```
a=5
b=9
c=a/b
s="{0}/{1}={2}".format(a,b,c)
print(s)
```



```
5/9=0.5555555555555556
```

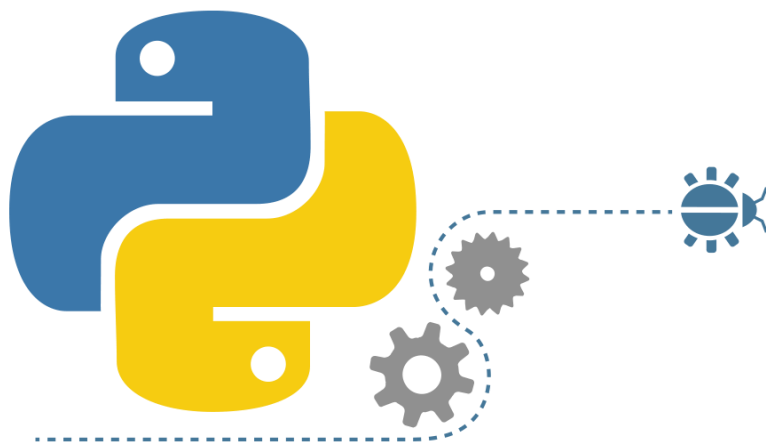


Nội dung bài học

substring

```
x = "Hello World!"  
print(x[2:]) # "llo World!"  
print(x[:2]) # "He"  
print(x[:-2]) # "Hello Worl"  
print(x[-2:]) # "d!"  
print(x[2:-2]) # "llo Worl"  
print(x[6:11]) # "World"
```

Hàm tách chuỗi



Nội dung bài học

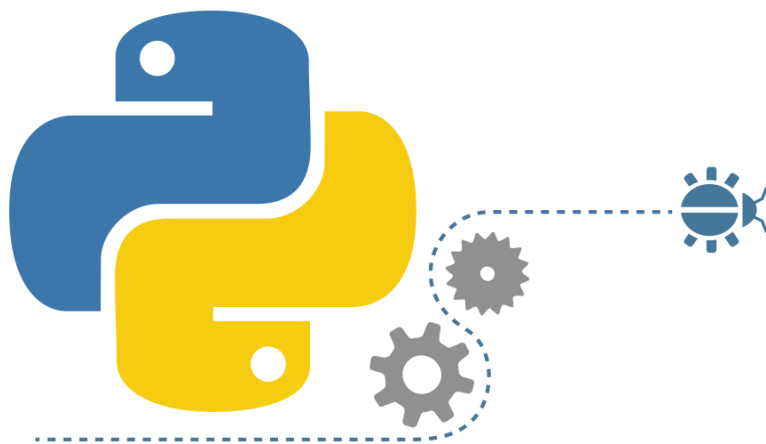
Hàm split dùng để tách chuỗi thành mảng các chuỗi con

```
s="sv007;Nguyễn Thị Tệt;1/1/1999"  
arr=s.split(';')  
for x in arr:  
    print(x)
```



```
sv007  
Nguyễn Thị Tệt  
1/1/1999
```


Hàm nối chuỗi



Nội dung bài học

Hàm **join** dùng để nối Chuỗi:

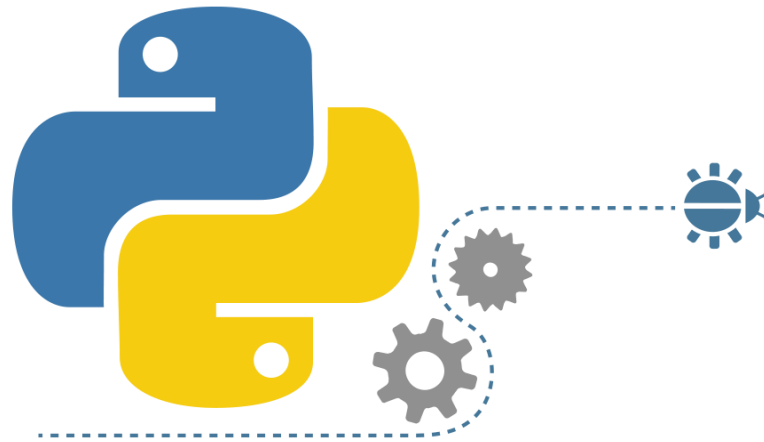
```
s="sv007;Nguyễn Thị Tệt;1/1/1999"  
arr=s.split(';')  
for x in arr:  
    print(x)  
s2=","  
s2=s2.join(arr)  
print(s2)
```



```
sv007  
Nguyễn Thị Tệt  
1/1/1999  
sv007,Nguyễn Thị Tệt,1/1/1999
```


Bài tập rèn luyện

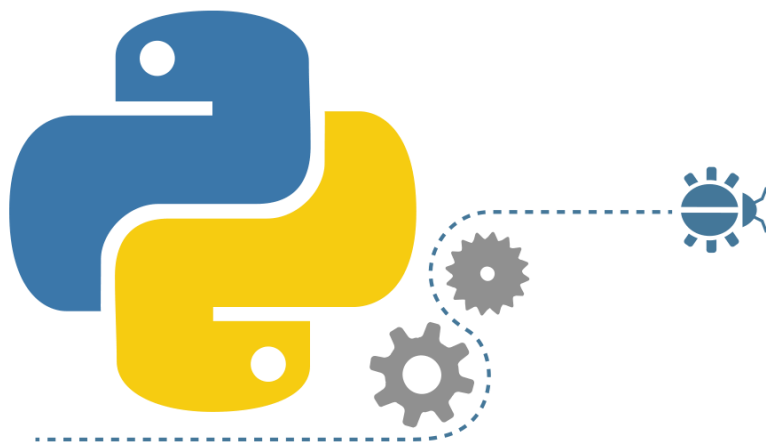
-Kiểm tra chuỗi đối xứng



Nội dung bài học

Dùng vòng lặp while vĩnh cửu, cho phép Nhập vào một Chuỗi → Xuất Chuỗi này có phải đối xứng hay không? Hỏi người sử dụng có tiếp tục phần mềm. Nếu tiếp tục thì nhập Chuỗi mới, còn không thì thoát và thông báo cảm ơn

Viết chương trình tối ưu chuỗi

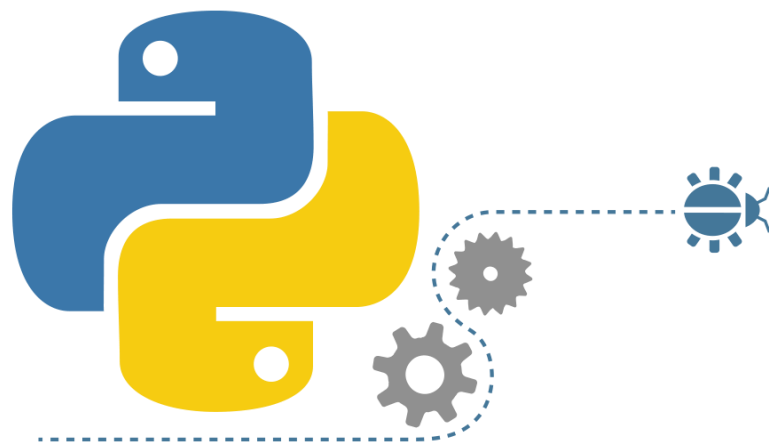




Nội dung bài học

Một Chuỗi được gọi là tối ưu khi: Không chứa các khoảng trắng dư thừa, các từ cách nhau bởi một khoảng trắng

Tách xử lý chuỗi



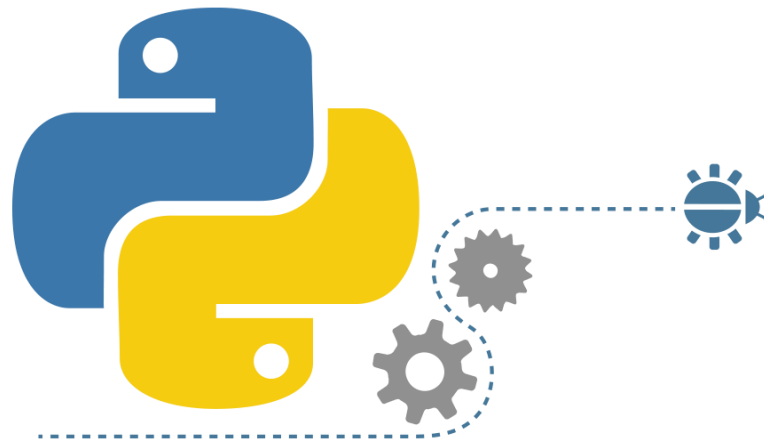


Nội dung bài học

Cho 1 Chuỗi như sau “5;7;8;-2;8;11;13;9;10”

- xuất các chữ số trên các dòng riêng biệt
- Xuất có bao nhiêu chữ số chẵn
- Xuất có bao nhiêu số âm
- Xuất có bao nhiêu chữ số nguyên tố
- Tính giá trị trung bình

Các bài tập tự rèn luyện



Nội dung bài học

Câu 1: Trình bày một số hàm quan trọng trong xử lý Chuỗi của Python

Câu 2: Viết chương trình cho phép nhập vào 1 chuỗi. Yêu cầu xuất ra:

- Bao nhiêu chữ IN HOA
- Bao nhiêu chữ in thường
- Bao nhiêu chữ là chữ số
- Bao nhiêu chữ là ký tự đặc biệt
- Bao nhiêu chữ là khoảng trắng
- Bao nhiêu chữ là Nguyên Âm
- Bao nhiêu chữ là Phụ âm

Nội dung bài học

Câu 3: Viết một hàm đặt tên là **NegativeNumberInStrings(str)**. Hàm này có đối số truyền vào là một chuỗi bất kỳ, Hãy viết lệnh để xuất ra các số nguyên âm trong chuỗi.

- **Ví dụ:** Nếu nhập vào chuỗi “**abc-5xyz-12k9l--p**” thì hàm phải xuất ra được 2 số nguyên âm đó là -5 và -12

Nội dung bài học

Câu 4: Viết chương trình tối ưu Chuỗi danh từ

Một Chuỗi được gọi là tối ưu khi: Không chứa các khoảng trắng dư thừa, các từ cách nhau bởi một khoảng trắng, Ký tự đầu tiên của các từ Viết Hoa

Ví dụ:

Input “ TRần duY thAnH ”

Output “Trần Duy Thanh”