

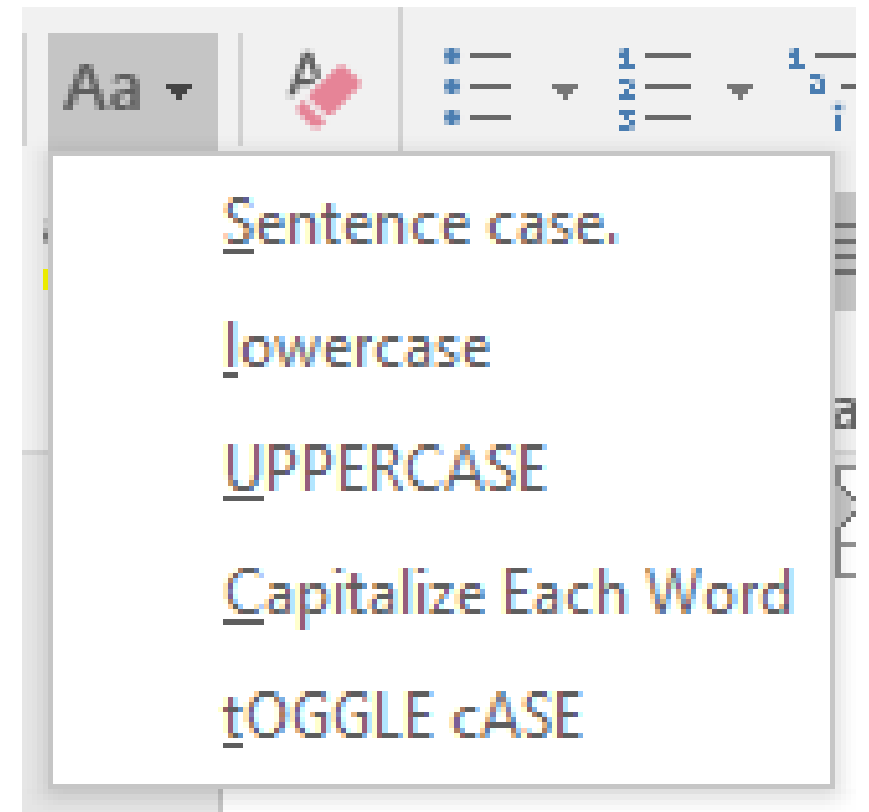
Kiểu dữ liệu Xâu kí tự

Hà Nội, 2025



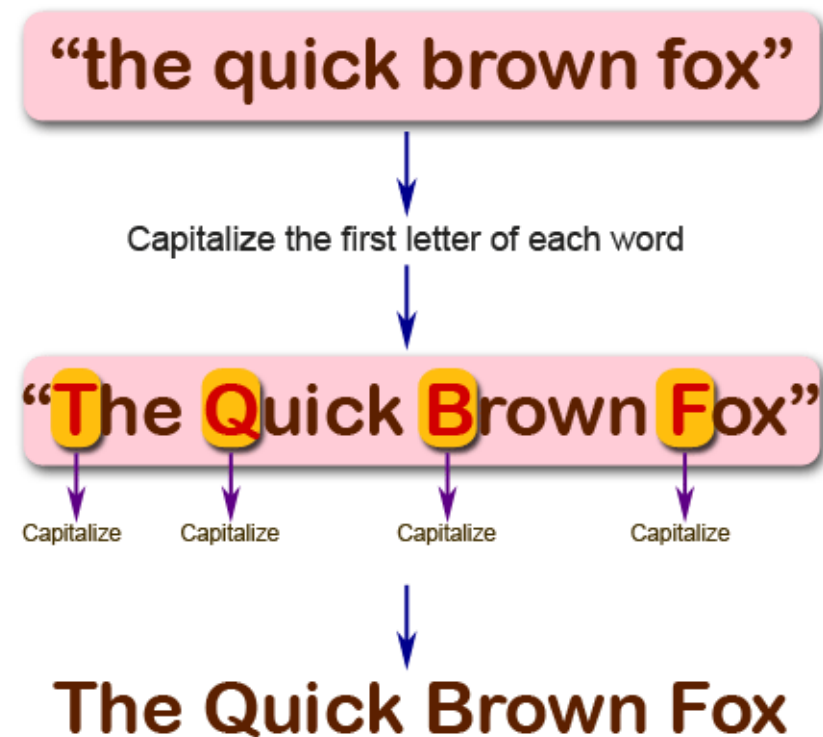
XỬ LÝ VĂN BẢN

Có nhiều bài toán cần phải xử lý chuỗi văn bản. Microsoft Word là một trong những phần mềm xử lý văn bản phổ biến nhất trên thế giới. Chức năng chính của phần mềm này là xử lý và định dạng văn bản.



NỘI DUNG

- Cấu trúc dữ liệu chuỗi kí tự, **string**
 - Định nghĩa và khai báo
 - Truy cập các kí tự trong chuỗi
- Các phép toán
- Các thao tác với chuỗi
- Các hàm thư viện xử lý chuỗi
- Định dạng chuỗi



© w3resource.com

KIỂU DỮ LIỆU STRING

XÂU KÍ TỰ STRING

- Là một kiểu dữ liệu cơ bản trong Python, dùng để lưu trữ một danh sách các ký tự liên tiếp nhau.
- Viết các ký tự trong cặp dấu nháy đơn (') hoặc cặp dấu nháy kép (") hoặc cặp ba dấu nháy đơn ('''') hoặc cặp ba dấu nháy kép ("""') để biểu thị một hằng xâu ký tự.
 - Mang đến sự thuận tiện cho người dùng
- Xâu có thể rỗng (gồm 0 ký tự)
- Python không có kiểu ký tự, thay vào đó ký tự được coi là xâu có độ dài 1, chỉ bao gồm một ký tự

```
str1 = "Hello"
str2 = 'there'
str = str1 + str2
print(str)
str3 = ""
print(str3)
str4 = """A"""
print(str4)
```

XÂU KÍ TỰ STRING

```
>>>str = '123'
>>>str = str + 1
Traceback (most recent call last): File
"<input>", line 1, in <module> TypeError:
can only concatenate str (not "int") to
str
>>>x = int(str) + 1
>>>print(x)
124
>>>y = float("1234.5")
>>>print(y)
1234.5
```

- Xâu chứa các chữ số khác với một số
 - "123" != 123
- Có thể chuyển một xâu kí tự số thành một số tương ứng và ngược lại.
- Tất cả dữ liệu đọc vào từ bàn phím đều được lưu dưới dạng xâu kí tự
- Cần chuyển đổi kiểu phù hợp trước khi sử dụng

MỘT SỐ KÍ TỰ ĐẶC BIỆT

Kí tự	Nội dung
\n	Xuống dòng
\r	Về đầu dòng (Enter)
\t	Tab
\v	Tab dọc
\b	Lùi trái (Backspace)
\f	Hết form
\'	Nháy đơn '
\"	Nháy kép "
\\	Dấu sổ ngược \

- Ngoài các kí tự thông thường (1 kí tự), python còn cung cấp các kí tự đặc biệt.
- Kí tự đặc biệt là dãy các kí tự bao gồm dấu gạch phải (\) và theo sau là một kí tự hoặc dãy số.
- Hãy cho biết kết quả hiển thị đoạn code dưới đây:

```
str = """Me di lam\n\tTu sang som\n\t\bDay  
thoi com\n\tMua thit ca"""  
print(str)
```

TRUY CẬP CÁC PHẦN TỬ

- Các kí tự trong xâu được tham chiếu đến bằng cách sử dụng chỉ số (*index*) đặt bên trong dấu ngoặc vuông([]).
- Chỉ số của xâu kí tự được bắt đầu từ 0, tính từ trái sang phải hoặc -1 tính từ phải sang trái

Chỉ số âm	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
str	l	t		i	s		a		c	a	r	d
Chỉ số dương	0	1	2	3	4	5	6	7	8	9	10	11

TRUY CẬP CÁC PHẦN TỬ

- Nếu chỉ số nằm ngoài phạm vi cho phép, lỗi sẽ xảy ra
- Nếu xâu có độ dài n, chỉ số nằm trong khoảng [0, n)

```
>>> st = "Day la cai ban"
```

```
>>> st[1]
```

```
???
```

```
>>> st[16]
```

```
Traceback (most recent call last):
```

```
  File "<pyshell>", line 1, in <module>
```

```
IndexError: string index out of range
```

```
>>> st[-3]
```

```
???
```

```
>>> st[-20]
```

```
Traceback (most recent call last):
```

```
  File "<pyshell>", line 1, in <module>
```

```
IndexError: string index out of range
```

ĐẶC ĐIỂM CỦA STRING

- Xâu kí tự là bất biến
 - Không thể thay đổi, thêm hay bớt kí tự.
 - Các kí tự cũng không thể thay đổi
- Khi thực hiện các phép toán trên xâu, ví dụ phép nối xâu '+', một xâu mới sẽ được tạo ra để gán vào biến hiện tại. Ví dụ
 - `s = "Hello"`
 - `s = s + "Everybody"`

```
s = "nham Dan"
```

Lệnh	Kết quả
<code>s[0] = "N"</code>	Đ/S
<code>s = "Nham Ty"</code>	Đ/S
<code>s = "2022 la" + s</code>	Đ/S
<code>s = "123" + 1</code>	Đ/S
<code>print(s[20])</code>	Đ/S

ĐẶC ĐIỂM CỦA STRING

- Xâu kí tự là bất biến
 - Không thể thay đổi, thêm hay bớt kí tự.
 - Các kí tự cũng không thể thay đổi
- Khi thực hiện các phép toán trên xâu, ví dụ phép nối xâu '+', một xâu mới sẽ được tạo ra để gán vào biến hiện tại. Ví dụ
 - `s = "Hello"`
 - `s = s + "Everybody"`

```
s = "nham Dan"
```

Lệnh	Kết quả
<code>s[0] = "N"</code>	S
<code>s = "Nham Ty"</code>	Đ
<code>s = "2022 la" + s</code>	Đ
<code>s = "123" + 1</code>	S
<code>print(s[20])</code>	Đ

DUYỆT XÂU KÍ TỰ

- Có 2 cách để duyệt chuỗi:
 - Duyệt trực tiếp qua từng ký tự với vòng lặp **for**
 - Duyệt dựa trên các chỉ số với vòng lặp **for** hoặc **while**
 - Cần chú ý độ dài chuỗi để không truy cập vào các vị trí nằm ngoài phạm vi cho phép.
 - Nên sử dụng hàm **len()**

```
st = "Ten toi la Hoa"
for ch in st:
    print(ch, end=' ')
for i in range(len(st)):
    print(st[i], end=' ')
```

Kết quả

```
Ten toi la Hoa
Ten toi la Hoa
```

Bài tập *thực hành* 6.1. Giới thiệu về kiểu xâu kí tự trên hệ thống LMS

Thời gian: 20 phút

Từ : 20h15 đến: 20h20

CÁC PHÉP TOÁN VÀ HÀM

CÁC HÀM CƠ BẢN

Phương thức	Mô tả
<i>len(st)</i>	Trả về độ dài của xâu <i>st</i>
<i>max(st)</i>	Trả về kí tự lớn nhất (theo mã ASCII) trong xâu <i>st</i>
<i>min(st)</i>	Trả về kí tự nhỏ nhất (theo mã ASCII) trong xâu <i>st</i>

CÁC PHÉP TOÁN CƠ BẢN

Toán tử	Cú pháp	Mô tả
+	str1 + str2	Nối 2 chuỗi kí tự <i>str1</i> và <i>str2</i> để tạo thành một chuỗi mới chứa cả hai chuỗi
*	str * <so_lan>	Tạo một chuỗi kí tự bằng cách nối nhiều bản sao của chuỗi <i>str</i> lại
[:]	str[start:stop:step]	Trả về một chuỗi con chứa các kí tự trong str bắt đầu từ <start> đến <stop>-1. Dấu ":" bắt buộc phải có, các thành phần start, stop và step có thể không có. Nếu không có, mặc định start=0; stop=len(str); step=1.
in	x in str	Trả về True nếu chuỗi x có xuất hiện trong chuỗi <i>str</i>
not in	x not in str	Trả về True nếu chuỗi x không có trong chuỗi <i>str</i>

PHÉP CẮT XÂU :

0 1 2 3 4 5 6 7 8 9 10

S =

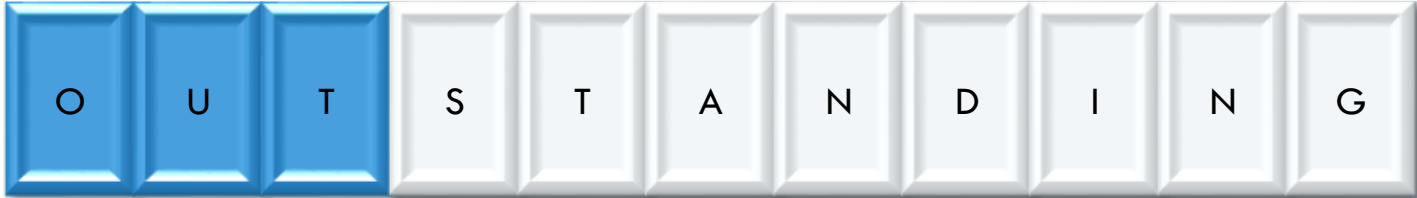
O	U	T	S	T	A	N	D	I	N	G
---	---	---	---	---	---	---	---	---	---	---

$s1 = s[3:8]$

$s1 =$

PHÉP CẮT XÂU :

0 1 2 3 4 5 6 7 8 9 10

$s =$ 

$s1 = s[3: 100]$

$s1 =$

PHÉP CẮT XÂU :

$s =$

0	1	2	3	4	5	6	7	8	9	10
O	U	T	S	T	A	N	D	I	N	G

$s1 = s[:]$

$s1 =$

CÁC PHÉP TOÁN CƠ BẢN

s1 = "Chao"

STT	Phép toán	Kết quả
1	s = s1 + s2	???
2	s = s1*3	???
3	s = s1[1:2]	???
4	s = s1[1:]	???
5	s = s1[-1:-3]	???
6	s = s1[-1:-3:-1]	???
7	"Un" in s1	

s2 = "ban"

STT	Phép toán	Kết quả
1	s = s1 + s2	Chaoban
2	s = s1*3	ChaoChaoChao
3	s = s1[1:2]	h
4	s = s1[1:]	hao
5	s = s1[-1:-3]	
6	s = s1[-1:-3:-1]	oa
	"Un" in s1	False

CÁC PHÉP TOÁN QUAN HỆ

Toán tử	Cú pháp	Mô tả
==	str1 == str2	Trả về <i>True</i> nếu hai chuỗi <i>str1</i> và <i>str2</i> giống nhau
!=	str1 != str2	Trả về <i>True</i> nếu hai chuỗi <i>str1</i> và <i>str2</i> không giống nhau
>	str1 > str2	Trả về <i>True</i> nếu chuỗi <i>str1</i> lớn hơn <i>str2</i>
>=	str1 >= str2	Trả về <i>True</i> nếu chuỗi <i>str1</i> lớn hơn hoặc bằng <i>str2</i>
<	str1 < str2	Trả về <i>True</i> nếu chuỗi <i>str1</i> nhỏ hơn <i>str2</i>
<=	str1 <= str2	Trả về <i>True</i> nếu chuỗi <i>str1</i> nhỏ hơn hoặc bằng <i>str2</i>

Python so sánh hai chuỗi dựa trên mã ASCII của ký tự, bắt đầu từ trái sang phải.
Và mọi chuỗi đều lớn hơn chuỗi rỗng

THỨ TỰ TỪ ĐIỂN

- Nguyên lý so sánh hai xâu:

$$A = a_0 a_1 \dots a_m \text{ và } B = b_0 b_1 \dots b_n$$

- So sánh từ đầu theo từng cặp ký tự (a_i, b_i) , $i = 1, 2, 3, \dots$
 - Nếu gặp $a_i < b_i$, kết luận ngay $a < b$, stop
 - Nếu gặp $a_i > b_i$, kết luận ngay $a > b$, stop
- Nếu không thấy cặp ký tự tương ứng khác nhau, xâu ngắn hơn được coi là nhỏ hơn

"computer" > "calculator"

"COMPUTER" < "calculator"

"professional" < "professor"

"10" < "9"

"with" < "without"

Tốc độ phép so sánh phụ thuộc vào vị trí i đầu tiên mà $a_i \neq b_i$

CÁC PHÉP TOÁN QUAN HỆ

s1 = "Hanh"

s2 = "Hanna"

STT	Phép toán	Kết quả
1	s1 == s2	???
2	s1 != s2	???
3	s1 > s2	???
4	s1 >= s2	???
5	s1 < s2	???
6	s1 <= s2	???
7	s1 >= "Hanhit"	

STT	Phép toán	Kết quả
1	s1 == s2	False
2	s1 != s2	True
3	s1 > s2	False
4	s1 >= s2	False
5	s1 < s2	True
6	s1 <= s2	True
7	s1 >= "Hanhit"	False

Bài tập *Thực hành 6.2. Các hàm với chuỗi ký tự* trên hệ thống LMS

Thời gian: 20 phút

Từ : 20h40 đến: 21h

CÁC THAO TÁC XÂU

CÁC THAO TÁC VỚI XÂU

- Thực hiện dựa trên các phương thức của kiểu dữ liệu **string**
- Được chia làm 4 nhóm chính:
 - Chuyển đổi
 - Kiểm tra
 - Tìm kiếm, thay thế
 - Phân tích



THAO TÁC CHUYỂN ĐỔI

Phương thức	Mô tả
<i>st.capitalize()</i>	Viết hoa chữ cái đầu tiên của xâu
<i>st.lower()</i>	Trả về một bản sao của xâu <i>st</i> trong đó chữ hoa được chuyển sang chữ thường
<i>st.swapcase()</i>	Trả về một bản sao của xâu <i>st</i> trong đó kiểu chữ viết của tất cả các kí tự được đảo ngược, chữ hoa thành chữ thường và ngược lại
<i>st.title()</i>	Trả về một bản sao của xâu <i>st</i> trong đó tất cả kí tự đầu tiên của tất cả các từ là chữ hoa
<i>st.upper()</i>	Trả về một bản sao của xâu trong đó tất cả kí tự đều là chữ hoa

VÍ DỤ 6.4

```
s = "Cong Hoa xa hoi chu nghia"  
st = s.lower()  
print(st)  
sh = s.upper()  
print(sh)  
ss = s.capitalize()  
print(ss)  
sw = s.title()  
print(sw)  
so = sw.swapcase()  
print(so)
```

```
cong hoa xa hoi chu nghia  
CONG HOA XA HOI CHU NGHIA  
Cong hoa xa hoi chu nghia  
Cong Hoa Xa Hoi Chu Nghia  
cONG hOA xA hOI cHU nGHIA
```

THAO TÁC KIỂM TRA

Phương thức	Mô tả
<i>st.isalpha()</i>	Trả về <i>True</i> nếu xâu <i>st</i> có ít nhất 1 kí tự và tất cả các kí tự là chữ cái, ngược lại trả về <i>False</i> .
<i>st.isdigit()</i>	Trả về <i>True</i> nếu xâu <i>st</i> có ít nhất một kí tự và tất cả các kí tự trong xâu đều là chữ số, ngược lại, trả về <i>False</i> .
<i>st.islower()</i>	Trả về <i>True</i> nếu tất cả các kí tự trong xâu <i>st</i> là chữ thường, ngược lại trả về <i>False</i> .
<i>st.istitle()</i>	Trả về <i>True</i> nếu xâu <i>st</i> có ít nhất một kí tự và tất cả kí tự phải ở dạng titlecase (kí tự đầu mỗi từ là chữ hoa, các kí tự khác là chữ thường), ngược lại trả về <i>False</i> .
<i>st.isupper()</i>	Trả về <i>True</i> nếu <i>st</i> có ít nhất một kí tự và tất cả kí tự trong xâu <i>st</i> là chữ hoa, ngược lại trả về <i>False</i> .

TÌM KIẾM VÀ THAY THẾ

Phương thức	Mô tả
<code>st.find(substr [,start] [,stop])</code>	Trả về vị trí xuất hiện đầu tiên của chuỗi <i>substr</i> trong chuỗi <i>st</i> hoặc chuỗi con từ vị trí <i>start</i> đến vị trí <i>stop</i> của chuỗi <i>st</i> . Nếu không tìm thấy, trả về -1.
<code>st.replace(old, new [, max])</code>	Trả về một bản sao của <i>st</i> trong đó thay thế tất cả các chuỗi <i>old</i> bằng chuỗi <i>new</i> với số lần tối đa là <i>max</i> . Tham số <i>max</i> có thể có hoặc không, nếu không mặc định là thay thế tất cả các chuỗi <i>old</i> tìm thấy.
<code>st.rfind(substr [,start] [,stop])</code>	Tương tự <i>find()</i> nhưng tìm kiếm từ phải sang trái, do đó trả về vị trí xuất hiện cuối cùng của chuỗi <i>substr</i> trong chuỗi <i>st</i> .

VÍ DỤ 6.6

- Họ tên của người Việt bao gồm ba thành phần: họ, tên đệm và tên. Hãy viết chương trình cho phép thực hiện các công việc sau:
 - Nhập vào họ tên đầy đủ của một người
 - Hiển thị họ của người đó ra màn hình
 - Hiển thị tên của người đó ra màn hình
 - Hiển thị họ đệm của người đó ra màn hình

Đặt tên con trai họ Nguyễn

NGUYỄN BÁ AN

NGUYỄN MINH HUY

NGUYỄN VŨ ĐĂNG KHOA

NGUYỄN LÊ KIM LUÂN

NGUYỄN TRẦN THANH DƯƠNG

NGUYỄN PHẠM GIA HÙNG



VÍ DỤ 6.6

```
s = input("Nhap ho ten: ")
i = s.find(" ")
ho = s[:i]
print(f"Ho: {ho}")
j = s.rfind(" ")
ten = s[(j+1):]
print(f"Ten: {ten}")
dem = s[(i+1):j]
print(f"Dem: {dem}")
```

```
Nhap ho ten: dang thanh trung
Ho: dang
Ten: trung
Dem: thanh
```

```
Nhap ho ten:   dang thanh   trung
Ho:
Ten:
Dem:  dang thanh   trung
```

PHÂN TÍCH

Phương thức	Mô tả
<i>sep.join(lst)</i>	Trả về một chuỗi bằng cách nối các chuỗi biểu diễn các phần tử trong dãy <i>lst</i> thành một chuỗi trong đó sử dụng <i>sep</i> làm dấu phân tách giữa chuỗi con.
<i>st.split(sep)</i>	Trả về danh sách các chuỗi con của <i>st</i> được chia theo <i>sep</i> .
<i>st.strip()</i>	Trả về một chuỗi bằng cách loại bỏ các khoảng trắng ở bên trái và bên phải chuỗi
<i>st.rstrip()</i>	Trả về một chuỗi bằng cách loại bỏ các khoảng trắng ở bên phải chuỗi
<i>st.lstrip()</i>	Trả về một chuỗi bằng cách loại bỏ các khoảng trắng ở bên trái chuỗi

VÍ DỤ 6.6

```
s = input("Nhap ho ten: ")
l = s.split()
print(l)
print(f"Ho: {l[0]}")
print(f"Ten: {l[2]}")
print(f"Dem: {l[1]}")
```

```
Nhap ho ten: dang thanh trung
Ho: dang
Ten: trung
Dem: thanh
```

```
Nhap ho ten:  dang  thanh
trung
['dang', 'thanh', 'trung']
Ho: dang
Ten: trung
Dem: thanh
```

Bài tập *thực hành* 6.3. Các thao tác với xâu kí tự trên hệ thống LMS

Thời gian: 20 phút

Từ : 21h15 đến: 21h35

CÂU HỎI

