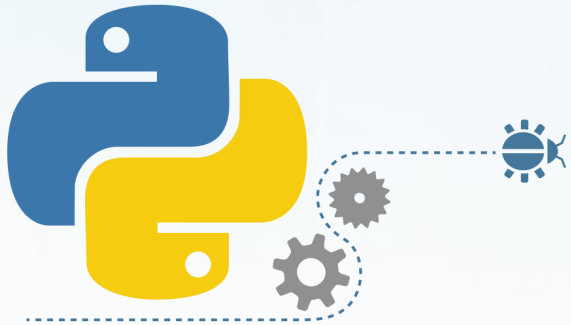
TRƯỜNG ĐẠI HỌC
SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
HCMC University of Technology and Education

KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN

NHẬP MÔN LẬP TRÌNH PYTHON (IPPA233277)

XỬ LÝ CHUỖI



GV. Trần Quang Khải

1. Hiểu được khái niệm và cấu trúc của chuỗi
2. Nắm và vận dụng được cách thức định dạng chuỗi
3. Nắm được chuỗi định dạng Unicode, ký tự thoát
4. Thực hiện được các hàm cơ bản trên chuỗi

1. Khái niệm và cấu trúc của chuỗi
2. Truy cập giá trị trong chuỗi
3. Định dạng chuỗi
4. Các ký tự thoát
5. Chuỗi dạng Unicode
6. Các hàm cơ bản trên chuỗi

- Chuỗi là một tập các ký tự nằm trong nháy đơn hoặc nháy kép hoặc 3 nháy đơn hoặc 3 nháy kép (Python coi nháy đơn hoặc nháy kép là như nhau)
- Khi sử dụng nháy đơn hoặc nháy kép, toàn bộ chuỗi phải nằm trong một dòng.
- Khi nằm trên nhiều dòng, thì chuỗi phải nằm trong 3 nháy đơn hoặc 3 nháy kép

Ví dụ:

```
str1 = 'Hello'
str2 = "everyone!"
str3 = '''\nWelcome to: '''
str4 = """\nPython class
Math class
... """
print(str1, str2, str3, str4)
```

- Các chuỗi trong Python là mảng các byte đại diện cho các ký tự Unicode
- Python không tồn tại kiểu dữ liệu ký tự, mà hiểu một chuỗi với độ dài là 1
- Sử dụng dấu ngoặc [ ] để truy xuất từng phần tử của chuỗi với ký tự đầu tiên có index = 0
- Việc truy xuất chuỗi giống truy xuất mảng

- Sử dụng hàm len() tính chiều dài chuỗi trong Python

```
a = "Hello World!"
print(len(a)) # 12
```

- Sử dụng hàm strip() loại bỏ khoảng trắng ở đầu hoặc cuối chuỗi

```
a = "    Hello World!    "
print(a.strip()) # returns "Hello World!"
```

- Sử dụng hàm lower() / upper() / capitalize() định dạng chữ thường / chữ hoa / in hoa chữ cái đầu cho chuỗi

```
a = "hello World!"
print(a.lower()) # returns "hello world!"
print(a.upper()) # returns "HELLO WORLD!"
print(a.capitalize()) # returns "Hello World!"
```


- Sử dụng hàm replace() để thay thế một chuỗi xác định bằng một chuỗi khác

```
a = "Hello World!"
print(a.replace("World", "friends")) # Hello friends!
```

- Sử dụng hàm split() để tách chuỗi thành các chuỗi con

```
a = "Bao, Khang, Huyen"
print(a.split(",")) # ['Bao', ' Khang', ' Huyen']
```

- Sử dụng từ khóa in hoặc not in để kiểm tra chuỗi có tồn tại trong một chuỗi khác

```
str1 = "Python class"
x = "Python" in str1
print(x) # True
```

- Sử dụng hàm startswith() / endswith() để kiểm tra chuỗi có bắt đầu hoặc kết thúc bằng một chuỗi con nào không

```
str = 'Hello World!'
print(str.startswith('hello'))   # False
print(str.endswith('!'))         # True
```

- Sử dụng hàm find() / rfind() để trả về vị trí đầu tiên / cuối cùng tìm thấy trong chuỗi hoặc -1 nếu không tìm thấy

```
str = 'Hello World!’
print(str.find('hello'))         # -1
print(str.find('Hello'))         # 0
print(str.rfind('World'))        # 6
print(str.rfind('!'))            # 11
```


- Sử dụng hàm count() để đếm số lần xuất hiện của chuỗi con trong chuỗi gốc, không tồn tại trả về 0

```
str = 'Hello World!'
print(str.count('o’)) # 2
print(str.count('l’)) # 3
print(str.count('a’)) # 0
```


- Sử dụng hàm join() để nối chuỗi

```
str = '0912218,Tran Quang A,1991,CNTT'
arrS = str.split(',')
for s in arrS:
    print(s)
str1 = ','
str2 = str1.join(arrS)
print(str2)
```

- Sử dụng hàm isalnum() để kiểm tra chuỗi chứa các ký tự chữ. True nếu chứa ký tự chữ và số

```
str = 'FIT-UTE'
print(str.isalnum()); # False (vì chứa '-')
```

- Sử dụng hàm isalpha() để kiểm tra chuỗi chứa các ký tự chữ hay không. False nếu chứa số hoặc ký tự đặc biệt

```
str = 'FITUTE'
print(str.isalpha()); # True
```

- Sử dụng hàm isdigit() để kiểm tra chuỗi chứa các ký tự số hay không.

```
str = '0912218'
print(str.isdigit()); # True
```

- Sử dụng các hàm căn lề cho chuỗi (trong đó len – số lượng ký tự chuỗi mới, char – ký tự hiện thị hai bên chuỗi cũ)

Căn lề giữa: center(len, char)

Căn lề phải: rjust(len, char)

Căn lề trái : lright(len, char)

```
str = 'FIT-UTE'
print(str.center(20))     #       FIT-UTE
print(str.center(20, '*'))# ******FIT-UTE*******
print(str.rjust(20, '*')) # ************FIT-UTE
print(str.ljust(20, '*')) # FIT-UTE************
```

- Sử dụng hàm format() để định dạng cách thức xuất chuỗi

```
str1 = 'Khang'; str2 = 'Bao'
print('Hello {0}, my name is {1}'.format(str1, str2)) # Hello Khang, my name is Bao
```


- Toán tử định dạng chuỗi (%) được sử dụng với hàm print()

Ví dụ:
```
print ("My name is %s and my weight is %d kg!" % ('Khang', 71) )
```

CHUỖI	CHUYỂN ĐỔI	CHUỖI	CHUYỂN ĐỔI
%c	Ký tự	%x	Số nguyên hệ thập lục phân (các chữ cái thường)
%s	Chuyển đổi thành chuỗi thông qua hàm str() trước khi định dạng	%X	Số nguyên hệ thập lục phân (các chữ cái hoa)
%i	Số nguyên thập phân có dấu	%e	Ký hiệu số mũ (với chữ thường 'e')
%d	Số nguyên thập phân có dấu	%E	Ký hiệu số mũ (với chữ hoa 'E')
%u	Số nguyên thập phân không dấu	%f	Số thực dấu chấm động
%o	Số nguyên hệ bát phân	%g	Viết gọn của %f và %e
		%G	Viết gọn của %f và %E

KÝ TỰ	BIỂU DIỄN (HỆ 16)	MIÊU TẢ	KÝ TỰ	BIỂU DIỄN (HỆ 16)	MIÊU TẢ
\a	0x07	Bell hoặc alert	\n	0x0a	Newline
\b	0x08	Backspace	\nnn		Notation trong hệ cơ số 8 với n là trong thuộc [0 .. 7]
\cx		Control-x	\r	0x0d	Carriage return
\C-x		Control-x	\s	0x20	Space
\e	0x1b	Escape	\t	0x09	Tab
\f	0x0c	Formfeed	\v	0x0b	Tab dọc
\M-\C-x		Meta-Control-x	\x		Ký tự x
			\xnn		Notation trong hệ 16 với n là trong dãy từ 0.9, a.f, hoặc A.F

- Chuỗi thông thường trong Python được lưu trữ dưới dạng ASCII 8bit, trong khi các chuỗi Unicode được lưu trữ dưới dạng Unicode 16bit. Điều này giúp chuỗi đa dạng hơn, bao gồm các ký tự đặc biệt

Ví dụ:

```
str = u'\u265E \u265F'
print(str); # ♞ ♟
```

- ✓ Họ tên : Trần Quang Khải
- ✓ Email : khaitq@hcmute.edu.vn
- ✓ Zalo (mã Qr)