

Bài 7: KIỂU DỮ LIỆU CHUỖI TRONG PYTHON (Phần 1)

Xem bài học trên website để ủng hộ Kteam: [Kiểu dữ liệu chuỗi trong Python – Phần 1](#).

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Trong các bài trước, Kteam đã giới thiệu đến bạn [KIỂU DỮ LIỆU SỐ](#) trong Python

Ở bài này chúng ta sẽ bắt đầu tìm hiểu đến các **KIỂU DỮ LIỆU CHUỖI** trong Python. Một trong những kiểu dữ liệu cực kì quan trọng trong Python.

Nội dung chính

Để đọc hiểu bài này tốt nhất bạn cần:

- Cài đặt sẵn [MÔI TRƯỜNG PHÁT TRIỂN CỦA PYTHON](#).
- Xem qua bài [CÁCH CHẠY CHƯƠNG TRÌNH PYTHON](#).
- Nắm [CÁCH GHI CHÚ](#) và [BIẾN TRONG PYTHON](#).

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Chuỗi là gì?
- Sự khác nhau giữa ' ' và " ".
- Chuỗi nhiều dòng với ' ' và " ". Khái niệm Docstring.
- Escape Sequence là gì?
- Câu hỏi củng cố

Chuỗi là gì?

Trong Python, chuỗi là những thứ được đặt trong cặp dấu ' ', hoặc " ", có thể cũng là trong cặp ' ' ' ', ' ' ' ' ' '. Nhưng cơ bản và thường dùng nhất là cặp ' ' và " ".

Ví dụ:

```
>>> 'How Kteam'
'How Kteam'
>>> s = 'Free Education' # gán cho biến s với giá trị là một chuỗi
>>> s
'Free Education'
>>> player = "Kteam" # gán cho biến player với một chuỗi có giá trị là 'Kteam'
>>> player
'Kteam'
>>> '# day la comment' # không đâu, nó là chuỗi. Đây mới là comment
'# day la comment'
>>> s = "String"
>>> type(s) # và kiểu dữ liệu chuỗi sẽ thuộc lớp 'str'
<class 'str'>
```

Sự khác nhau giữa " " và ' '

Nói về công dụng, thì hai cặp dấu nháy trên là tương đương. Những thứ nằm bên trong nó là một chuỗi.

Nhưng không có thứ gì sinh ra là để cho có. Hãy đặt vấn đề bạn muốn có chuỗi với nội dung sau đây và bạn muốn Python hiểu đó là một chuỗi

I'm Beginner

```
>>> 'I'm Beginner' # đặt nội dung vào trong cặp dấu ''
File "<stdin>", line 1
'I'm Beginner'
  ^
SyntaxError: invalid syntax
```

Hãy nhìn lại và phân tích tại sao lại có lỗi xảy ra???

Khi bạn gõ `'I'm Beginner'`. Python sẽ đọc từ trái qua phải và lấy từng kí tự trong chuỗi của bạn và việc này sẽ dừng lại khi nó gặp được dấu `'` còn lại.

Có nghĩa là nó sẽ đọc được chuỗi `'I'` sau đó kết thúc. Nhưng ta lại còn có một đoạn ở phía sau `m Beginner`. Thứ này với Python hoàn toàn vô nghĩa. Nó không hiểu được ý của bạn. Do đó một **SyntaxError** được thông báo lên.

Vậy, làm cách nào để ta có thể có được chuỗi với nội dung I'm Beginner???

Ta có 3 cách cơ bản để giải quyết vấn đề này. Và ngay sau đây, mình sẽ giới thiệu cách đơn giản nhất.

Ta sẽ lựa chọn cặp dấu ngoặc `" "` khi nội dung chuỗi của chúng ta có những kí tự `'`, và sẽ chọn cặp dấu ngoặc `" "` nếu nội dung chuỗi của chúng ta có những kí tự `"`.

```
>>> "I'm Beginner" # nội dung có kí tự ', chọn cặp dấu ""
"I'm Beginner"
>>> s = "It's good"
>>> s
"It's good"
>>> s2 = 'this is "special" word' # nội dung có kí tự ", chọn cặp ''
```

```
>>> s  
'this is "special" word'
```

Bạn có thể có câu hỏi rằng: "Nếu nội dung trong chuỗi vừa có kí tự ", lại vừa có kí tự ', thì ta phải chọn cặp dấu ngoặc nào?".

Kteam sẽ nợ bạn câu hỏi này vào phần sau. Theo dõi phần tiếp theo sẽ có câu trả lời từ Kteam nhé!

Chuỗi nhiều dòng với ''' và """". Khái niệm Docstring

Chuỗi nhiều dòng với ''' và """

Thường khi nhắc đến chuỗi, ta hay nghĩ tới một dòng. Và khi đó, ta sử dụng cặp dấu '' hoặc ". Nếu là nhiều dòng chuỗi kết nối với nhau, như những câu chữ bạn hay viết trong những cuốn vở thì đó cũng là một chuỗi, nhưng chuỗi đó sẽ được đặt trong cặp dấu ''' và """.

```
>>> s = '''dong 1  
...     dong 2  
...     dong 3'''  
>>> s  
'dong 1\ndong2\ndong3'  
>>> print(s) # kết quả mong muốn sẽ xuất hiện khi bạn sử dụng hàm print  
dong 1  
dong 2  
dong 3
```

Hãy khoan nói về việc tại sao kết quả chúng ta mong muốn phải qua tay hàm **print**. Nếu để ý, những lần chúng ta nhấn phím **enter** để xuống dòng, nhập tiếp dòng tiếp theo. Ở đó sẽ được thêm vào 2 kí tự \ và n.

Sự thật, `\n` được coi là một kí tự. Và đây chính là một **escape sequence**. Để hiểu rõ nó ra sao, **Kteam** sẽ giới thiệu với các bạn ở phần tiếp theo.

Quay trở lại, chúng ta đã biết muốn có nhiều dòng chuỗi kết hợp với nhau, ta sử dụng cặp dấu `"""` hoặc `'''`. Và đương nhiên, những thứ đặt trong cặp dấu `'''` hoặc `"""` cũng là một chuỗi. Do đó, ta cũng có thể tạo ra chuỗi chỉ một dòng và chứa những kí tự `'` và `"` khác.

```
>>> """chuoi vua co ki tu ' va ki tu ", that vi dai"""  
'chuoi vua co ki tu \' va ki tu ", that vi dai'
```

Có thể, một trong số các bạn sẽ bất ngờ với kết quả. Vì sao lại vậy nhỉ? Lại một lần nữa, vấn đề này liên quan tới escape sequence, thứ mà chúng ta sẽ tìm hiểu tiếp theo sau.

Khái niệm Docstring

Thêm một nội dung liên quan đến cặp dấu `'''` và `"""` nữa Kteam muốn giới thiệu với các bạn. Hai cặp dấu này thường không được sử dụng để tạo ra một chuỗi nhiều dòng, mà dùng để làm **DOCSTRING**

Docstring là :

- Một dạng chú thích nhiều dòng.
- Hay xuất hiện ở đầu một file Python, sau một dòng định nghĩa lớp, hàm.
- Và đây cũng là một trong những chuẩn quy ước về định dạng, trình bày code Python.

```
"""  
Đây là những dòng chú thích đầu file  
Về việc import các thư viện, module  
"""  
import lungtung  
import taolao  
  
def ham_vo_dung():  
    """  
    Còn đây là docstring
```

```
Cho một function
Đó là ham_vo_dung
'''

pass

class vo_van:
    '''
    Class khong co gi dau
    That day
    '''
    pass
```

Escape Sequence là gì?

Escape Sequence là một chuỗi (chính xác là kí tự) đặc biệt trong Python. Bắt đầu với một dấu ****.

Python có rất nhiều các escape sequence. Tuy nhiên, Kteam sẽ giới thiệu một số escape sequence chúng ta hay sử dụng nhất đối với mức độ cơ bản.

Tên	Kí hiệu	Giải thích
Alert	\a	Phát ra một tiếng bíp
Backspace	\b	Đưa con trỏ về lại một khoảng trắng
Newline	\n	Đưa con trỏ xuống dòng tiếp theo
Horizontal tab	\t	In một horizontal tab
Single quote	\'	In ra kí tự '
Double quote	\"	In ra kí tự "
Backslash	\\	In ra kí tự \

Kteam sẽ giúp bạn hiểu hơn về những escape sequence này. Nhớ là, nó phải được qua tay hàm print và hàm này sẽ được **Kteam** đề cập ở bài Nhập xuất trong Python.

```
>>> print('\a') # bạn sẽ nghe thấy một tiếng bíp

>>> print('\a\a') # nhớ bật âm lượng cho máy tính của mình

>>> print('abcd\be') # lùi con trỏ về trước 1 space, đề lên chữ d
abce

>>> print('dong 1\ndong 2')
dong 1
dong 2
>>> print('xuong dong\ndong moi\nthem mot dong moi')
xuong dong
dong moi
them mot dong moi
>>> print('abc\txyz')
abc   xyz
>>> print('\t\thello')
```

```
hello
>>> print('I\'m Beginner')
I'm Beginner
>>> print("one thing \"special\", that's it")
one thing "special", that's it
>>> print('Muon in dau \nay') # bạn cũng nên cẩn thận, kết quả sẽ không như
đợi đôi khi
Muon in dau
ay
>>> print('Muon in dau \\nay')
Muon in dau \nay
```

Củng cố bài học

Đáp án bài trước

Bạn có thể tìm thấy câu hỏi của phần này tại [CÂU HỎI Củng Cố](#) trong bài [KIỂU DỮ LIỆU SỐ TRONG PYTHON](#).

1. Kiểu dữ liệu số thuộc lớp `int`
2. Biến a là số nguyên thuộc lớp `int`, còn biến b là số thực thuộc lớp `float`
3. Vì kết quả của hàm `trunc` sẽ trả về một số nguyên là phần nguyên của phép chia, còn toán tử `//` thì kết quả sẽ cũng là số nguyên nhưng luôn luôn nhỏ hơn hoặc bằng kết quả.

- Ở trường hợp đầu tiên là

15 / - 4

Thì kết quả ta được

-3.75

Ta lấy phần nguyên bằng hàm `trunc` thì sẽ có kết quả là `-3`. Riêng với toán tử `//` sẽ làm tròn. Một là -3, hai là -4. Vì $-4 < -3.75$ do đó kết quả sẽ được là -4. Hai kết quả khác nhau

- Ở trường hợp thứ hai

15 / 4 thì kết quả sẽ là

3.75

Hàm `trunc` sẽ lấy phần nguyên là `3`. Toán tử `//` sẽ làm tròn. 3 hoặc là 4, mà $3 < 3.75$, do đó kết quả là 3. Hai kết quả giống nhau.

Câu hỏi củng cố

1. Những chuỗi nào sau đây là hợp lệ?

```
'nasdfiuqwnerp', "234a'adadf", ""asd34'asdfjoaisdfadf""", "\", ""\""", "
```

2. Sự khác nhau giữa hai biến a và b dưới đây là gì?

```
>>> a = 69
>>> b = '69'
```

3. Chỉ ra các Escape Sequence trong những giá trị sau đây

```
Chuỗi 1: '35\53ni34'
Chuỗi 2: '\n'
Chuỗi 3: "\\\V\\V\\V"
```

Đáp án của phần này sẽ được trình bày ở bài tiếp theo. Tuy nhiên, Kteam khuyến khích bạn tự trả lời các câu hỏi để củng cố kiến thức cũng như thực hành một cách tốt nhất!

Kết luận

Bài viết này đã giới thiệu sơ cho các bạn KIỂU DỮ LIỆU CHUỖI TRONG PYTHON – Phần 1.

Ở bài sau, Kteam sẽ tiếp tục nói về [KIỂU DỮ LIỆU CHUỖI TRONG PYTHON \(Phần 2\)](#)

Cảm ơn bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên “**Luyện tập – Thử thách – Không ngại khó**”.