Nhập xuất dữ liệu

Xuất dữ liệu

- Trong Python, hàm print() được sử dụng để xuất dữ liệu lên thiết bị xuất chuẩn (Màn hình Console)
- Cú pháp đầy đủ:

```
print(*object, sep=' ', end='\n', file=sys.stdout, flush=False)
```

Trong đó:

- objects – là các giá trị sẽ được xuất ra màn hình
- sep – phân cách giữa các giá trị. Giá trị mặc định của sep là một khoảng trắng.
- end – là kí tự được in ra sau khi các giá trị của objects đã được in. Mặc định của end là '\n' – xuống dòng mới.
- file – là đối tượng nơi mà các giá trị của objects sẽ được xuất ra. Giá trị mặc định là sys.stdout – màn hình console của Python.
- flush – tham số quy định cách thức xuất dữ liệu theo khối hay theo tuần tự các ký tự. Giá trị mặc định là False, nghĩa là dữ liệu được xuất ra theo khối.

Ví dụ 1 – Xuất ra một thông báo

```
print('This sentence is an output to the screen')
```

Kết quả của dòng lệnh trên sẽ là:

'This sentence is an output to the screen'

Ví dụ 2 – Xuất thông báo cùng với giá trị của biến

```
a = 5
print('The value of a is ' + str(a))
```

Kết quả của khối lệnh trên sẽ là:

```
The value of a is 5
```

- Xuất dữ liệu có định dạng (formatting): Xuất dữ liệu theo định dạng giúp cho dữ liệu xuất ra tự nhiên hơn, đẹp hơn bằng cách sử dụng phương thức str.format().
- Ví dụ 3: Xuất dữ liệu theo định dạng

```
x = 5
```

```
y = 10
print('The value of x is {} and y is {}'.format(x, y))
```

Hai dấu ngoặc nhọn {} là thứ tự vị trí hai biến x và y sẽ được xuất ra. Thứ tự xuất ra theo thứ tự của các biến trong hàm format(). Trong dấu ngoặc nhọn chúng ta cũng có thể để các giá trị index để quy định thứ tự xuất ra màn hình như ví dụ dưới:

```
print('I love {0} and {1}'.format('bread', 'butter'))
print('I love {1} and {0}'.format('bread', 'butter'))
```

Kết quả:

```
I love bread and butter
```

```
I love butter and bread
```

Ngoài ra trong dấu ngoặc nhọn, chúng ta có thể để các keyword thay thể cho các giá trị số index như ví dụ dưới:

```
print('Hello {name}, {greeting}'.format(greeting='Good morning',
name='John'))
```

** Ngoài cách thức xuất dữ liệu ra màn hình như đã trình bày ở trên, còn có các cách thức xuất dữ liệu với file, cơ sở dữ liệu…sẽ được trình bày ở các phần tiếp theo của khóa học.*

Nhập dữ liệu

Trong Python, hàm input() được sử dụng để nhập dữ liệu từ người dùng vào.

- Cú pháp:

```
input([prompt])
```

Trong đó prompt là một chuỗi kí tự xuất hiện khi nhập dữ liệu, giá trị prompt không bắt buộc phải có.

- Ví dụ

```
num = input('Enter a number: ')
print(num)
```

Import

Import trong Python là một cơ chế cho phép người lập trình có thể chia nhỏ code của một chương trình lớn thành các mô đun. Thông thường các mô đun này đảm nhận một vài chức năng nào đó và nó thường được lưu trong một file .py riêng biệt. Khi sử dụng, các file mô đun này có thể được liên kết bằng từ khóa import.

Ví dụ 1:

```
import math
print(math.pi)
```

Ở ví dụ trên, mô đun math được import vào chương trình, khi đó các hàm, các thuộc tính trong math được sử dụng bình thường. Trong ví dụ trên, dòng code thứ 2 chúng ta in ra giá trị số pi như hình dưới:

```
3.141592653589793
```

Khi chúng ta chỉ muốn sử dụng cụ thể một mô đun con của một mô đun lớn hơn, thao tác được thực hiện bằng từ khóa from.

Ví dụ 2:

```
>>> from math import pi
>>> pi
3.141592653589793
```

String

1. String trong Python là gì?

String trong Python là một chuỗi gồm các kí tự Unicode

2. Cách khởi tạo một string trong Python

String trong Python có thể được tạo bằng cách khai báo các kí tự được đặt trong dấu nháy đơn hoặc nháy kép như ví dụ bên dưới:

```
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)

my_string = '''Hello'''
```

```
print(my_string)

my_string = """Hello, welcome to
           the world of Python"""
print(my_string)
```

Kết quả:

Hello

Hello

Hello

Hello, welcome to the world of Python

Ta thấy rằng, ở ví dụ trên có bốn cách khởi tạo string, strong đó cách khai báo 1, 2 dùng để khai báo các chuỗi string trên cùng một dòng. Trong khi đó, cách thứ 3 và 4 dùng để khởi tạo một chuỗi string trên nhiều dòng.

1. Cách lấy các kí tự trong một string

Để truy xuất đến các kí tự riêng lẻ trong một chuỗi string chúng ta sử dụng chỉ số của các kí tự hoặc dùng slicing. Chỉ số trong chuỗi string bắt đầu từ 0 và là kiểu số nguyên. Khi truy xuất đến một kí tự mà có chỉ số vượt ra giới hạn chỉ số của chuỗi (chỉ số không tồn tại) chương trình sẽ phát sinh lỗi *IndexError.*

Chỉ số bằng -1 là một cách khác trong Python dùng để truy xuất đến kí tự cuối cùng của chuỗi, -2 là kí tự liền trước kí tự cuối cùng….

Ví dụ truy xuất đến các kí tự của chuỗi:

```
# Khai báo chuỗi string
my_string = "Python programming language"

#In chuỗi string
print('Chuỗi string = ', my_string)

# Kí tự đầu tiên của chuỗi
print('Kí tự đầu tiên của chuỗi = ', my_string[0])

# Kí tự cuối cùng của chuỗi
print('Kí tự cuối cùng của chuỗi = ', my_string[-1])

# Slicing từ kí tự 5 đến kí tự cuối, không bao gồm kí tự cuối
print('Chuỗi con = ', my_string[4:-1])
```

Kết quả khối lệnh:

Chuỗi string = Python programming language

Kí tự đầu tiên của chuỗi = P

Kí tự cuối cùng của chuỗi = e

Chuỗi con = on programming languag

1. Các thao tác với string
    1. Nối các string

Trong Python, toán tử + được dùng để nối 2 hoặc nhiều string với nhau.

Toán tử * được dùng để lặp chuỗi string với số lần cho trước.

Ví dụ:

```
# Khai báo chuỗi string
my_string01 = "Python "

my_string02 = "programming language"

# Dùng toán tử + để ghép hai chuỗi string
my_string03 = my_string01 + my_string02
print(my_string03)

# Dùng toán tử * để lặp chuỗi string
my_string04 = my_string01 * 4
print(my_string04)
```

Kết quả đoạn mã:

Python programming language

Python Python Python Python

1. Kiểm tra các kí tự, chuỗi con có trong chuỗi

Để kiểm tra sự tồn tại của các chuỗi con trong một chuỗi dùng *in, not in*

Ví dụ:

```
# Khai báo chuỗi string
my_string = "Python programming language"

# dùng in, not in để kiểm tra sự tồn tại của chuỗi con trong một chuỗi
print("Python" in my_string)
# Kết quả True
```

```
print("Python" not in my_string)
# Kết quả False
```

1. Thao tác string với các hàm có sẵn

Python cung cấp sẵn nhiều hàm để tăng hiệu quả khi làm việc với chuỗi. Sau đây là một số hàm thông dụng:

- len – trả về độ dài của chuỗi
- count – hàm trả về số lần chuỗi con có mặt trong chuỗi
- lower, upper – hàm chuyển đổi chuỗi về dạng in thường, in hoa
- lstrip – hàm loại bỏ các kí tự trắng ở đầu chuỗi
- rstrip – hàm loại bỏ các kí tự trắng ở cuối chuỗi
- strip – hàm loại bỏ các kí tự trắng ở đầu và cuối chuỗi.
- split – hàm chia chuỗi thành các chuỗi nhỏ hơn

Ví dụ các sử dụng các hàm giới thiệu ở trên:

```
# Sử dụng hàm len để lấy độ dài của chuỗi
my_string = "Python programming language"
print("Độ dài của chuỗi = ", len(my_string))

# Hàm count - trả về số lần chuỗi con có mặt trong chuỗi
my_string = "Python programming language and PHP programming language"
print("Số lần từ 'language' xuất hiện trong chuỗi = ",
my_string.count('language'))

# Hàm lower() - chuyển chuỗi về dạng in thường
my_string = "Python programming language"
print("Chuỗi in thường = ", my_string.lower())

# Hàm upper() - chuyển chuỗi về dạng in hoa
my_string = "Python programming language"
print("Chuỗi in hoa = ", my_string.upper())

# Hàm lstrip - loại bỏ các kí tự trắng ở đầu chuỗi
my_string = "   Python programming language   "
print("Chuỗi sau khi loại bỏ kí tự trắng ở đầu
'{}'".format(my_string.lstrip()))

# Hàm rstrip - loại bỏ các kí tự trắng ở cuối chuỗi
my_string = "   Python programming language   "
print("Chuỗi sau khi loại bỏ kí tự trắng ở cuối
'{}'".format(my_string.rstrip()))

# Hàm rstrip - loại bỏ các kí tự trắng ở đầu và cuối chuỗi
my_string = "   Python programming language   "
print("Chuỗi sau khi loại bỏ kí tự trắng ở đầu và cuối chuỗi
'{}'".format(my_string.strip()))

# Hàm split - chia chuỗi
```

```
my_string = "Python programming language"
print("Chuỗi sau khi chia = ",my_string.split(' '))
```

Kết quả của đoạn mã:

Độ dài của chuỗi = 27

Số lần từ ‘language’ xuất hiện trong chuỗi = 2

Chuỗi in thường = python programming language

Chuỗi in hoa = PYTHON PROGRAMMING LANGUAGE

Chuỗi sau khi loại bỏ kí tự trắng ở đầu ‘Python programming language   ‘

Chuỗi sau khi loại bỏ kí tự trắng ở cuối ‘   Python programming language’

Chuỗi sau khi loại bỏ kí tự trắng ở đầu và cuối chuỗi ‘Python programming language’

Chuỗi sau khi chia = [‘Python’, ‘programming’, ‘language’]

1. Định dạng String
    1. Escape Sequence

Xem ví dụ dưới:

```
# Khai báo chuỗi string
print('What's string in Python')
```

Khi chạy đoạn mã sẽ xuất hiện lỗi sau:

File “C:/caocao/Python Project/Python Fundamental/hello10.py”, line 2

print(‘What’s string in Python’)

^

SyntaxError: invalid syntax

Lỗi trên xuất hiện khi chuỗi string chứa các ký tự đặc biệt như nháy đơn, nháy kép, xuống dòng…

Để sửa lỗi, chuỗi phải được khai báo trong 3 nháy đơn hoặc 3 nháy kép như hình dưới:

```
print('''What's string in Python''')

print("""What's string in Python""")
```

Ngoài ra chúng ta có thể khắc phục lỗi trên bằng cách sử dụng escape sequence – kí tự '\'

Khi khởi tạo chuỗi bằng dấu nháy đơn thì tất cả các kí tự nháy đơn trong chuỗi phải được escaped. Tương tự khi khởi tạo chuỗi bằng dấu nháy kép thì phải escaped các kí tự nháy kép trong chuỗi đó.

Để khởi tạo chuỗi có chứa các kí tự đặc biệt như nháy đơn hoặc nháy kép…

```
print('What\'s string in Python')

print("What's \"string\" in Python""")
```

Các escape sequence hỗ trợ trong Python

[Bài tập] Ứng dụng chuyển đổi tiền tệ

Mục tiêu

Luyện tập cách thức nhập xuất dữ liệu, ép kiểu, tính toán cơ bản

Mô tả

Ứng dụng chuyển đổi tiền tệ cho phép tính giá trị tiền tệ giữa các đồng tiền khác nhau dựa vào một tỉ giá cho trước.

Hướng dẫn

Bước 1: Nhập số tiền USD cần đổi

Bước 2: Nhập tỉ giá USD/VND

Bước 3: Tính toán ra số tiền VND theo tỉ giá và số tiền USD nhập vào bên trên

Bước 4: In ra kết quả

[Thực hành] Tính chu vi và diện tích hình tròn

Mục tiêu

Luyện tập với cách thức nhập xuất dữ liệu

Mô tả bài toán

Nhập vào bán kính của hình tròn. Tính chu vi và diện tích của hình tròn có bán kính vừa nhập. Hiển thị kết quả ra màn hình.

Hướng dẫn thực hiện

Bước 1: Import các thư viện math

```
import math
```

Bước 2: Nhập vào bán kính

```
r = int(input("Enter the radius: "))
```

Giá trị nhận vào từ người dùng thông qua lệnh input là một chuỗi (string). Bán kính hình tròn có kiểu dữ liệu là số, phải dùng hàm int() để chuyển đổi từ string sang số.

Thử thay int với các kiểu dữ liệu khác như float. Xem sự khác biệt.

Bước 3: Tính chu vi, diện tích

```
c = 2 * math.pi * r
s = math.pi * r * r
```

Bước 4: Hiển thị kết quả

```
print("Chu vi của hình tròn có bán kính = {r} là {c}".format(r=r, c=c))
print("Diện tích của hình tròn có bán kính = {r} là {s}".format(r=r, s=s))
```