



Bài 06 - Nhập xuất dữ liệu

Khóa học: Python căn bản

Tóm tắt



Qua bài học này, chúng ta đã tìm hiểu những kiến thức sau:

1. Cách thức nhập xuất dữ liệu
2. Import file, thư viện, modules
3. String

Xuất dữ liệu



- Cú pháp

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

- 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.

Demo



Xuất dữ liệu có định dạng



- 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()`.
- Định dạng với index
- Định dạng với keyword

Demo



- Demo định dạng với index
- Demo định dạng với keyword

Nhập dữ liệu



- Hàm `input()` được sử dụng để nhập dữ liệu từ người dùng vào chương trình
- 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

Demo



- Demo chương trình thực hiện các phép toán giữa hai số được nhập vào từ bàn phím, kết quả hiển thị ra màn hình

Import



- Import là gì?
- Ưu điểm của cơ chế import:
 - Chia nhỏ chương trình, làm cho chương trình có cấu trúc
 - Dễ quan sát, theo dõi code
 - Dễ chia sẻ code

Demo



- Demo cách import 1 file .py (tự tạo) vào chương trình
- Demo cách import thư viện math và cách sử dụng các hàm trong math

String



- String trong Python là một chuỗi gồm các kí tự Unicode
- Cách khởi tạo:

```
# defining strings in Python
# all of the following are equivalent
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)

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

# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
                the world of Python"""
print(my_string)
```

String



- Cách truy xuất các kí tự trong string
 - Dùng index
 - Dùng slicing
- Lỗi *IndexError* khi làm việc với string
- Thao tác với string dùng các toán tử +, *
- Kiểm tra kí tự, chuỗi con trong chuỗi với **in**, **not in**

Demo



- Cách truy xuất các kí tự trong string với index, slicing
- Kiểm tra tồn tại của kí tự, chuỗi con với in, not in

String



- Thao tác string với các hàm có sẵn
 - len
 - count
 - lower, upper
 - lstrip
 - rstrip
 - strip
 - split

Demo



- Demo cách sử dụng các hàm thao tác với string có sẵn
 - len
 - count
 - lower, upper
 - lstrip
 - rstrip
 - strip
 - split



Escape Sequence

- Tình huống:
 - Khi string chứa các kí tự đặt biệt
- Vấn đề:
 - Không phân biệt được đó là nội dung của string hay là các kí tự chức năng
- Giải pháp
 - Phải escape để chương trình hiểu đó là nội dung của string

Escape Sequence



Escape Sequence	Description
<code>\newline</code>	Backslash and newline ignored
<code>\\</code>	Backslash
<code>\'</code>	Single quote
<code>\"</code>	Double quote
<code>\a</code>	ASCII Bell
<code>\b</code>	ASCII Backspace
<code>\f</code>	ASCII Formfeed
<code>\n</code>	ASCII Linefeed
<code>\r</code>	ASCII Carriage Return
<code>\t</code>	ASCII Horizontal Tab
<code>\v</code>	ASCII Vertical Tab
<code>\ooo</code>	Character with octal value ooo
<code>\xHH</code>	Character with hexadecimal value HH

Demo



- Demo một số trường hợp phải escape
 - Chuỗi có chứa kí tự '
 - Chuỗi là đường dẫn

Raw String



- Để kí tự r hoặc R trước string để bỏ qua tất cả các escape sequence trong string
- Demo

Tổng kết



- Cách nhập xuất dữ liệu
- Import
- String và một số thao tác trên String