

Học Python cơ bản và nâng cao - Học ngôn ngữ lập trình Python hay nhất

Python là gì

Python là một ngôn ngữ lập trình bậc cao, thông dịch, hướng đối tượng, đa mục đích và cũng là một ngôn ngữ lập trình động.

Cú pháp của Python là khá dễ dàng để học và ngôn ngữ này cũng mạnh mẽ và linh hoạt không kém các ngôn ngữ khác trong việc phát triển các ứng dụng. Python hỗ trợ mẫu đa lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và mệnh lệnh hoặc là các phong cách lập trình theo thủ tục.

Python không chỉ làm việc trên lĩnh vực đặc biệt như lập trình web, và đó là tại sao ngôn ngữ này là đa mục đích bởi vì nó có thể được sử dụng với web, enterprise, 3D CAD, ...

Bạn không cần sử dụng các kiểu dữ liệu để khai báo biến bởi vì kiểu của nó là động, vì thế bạn có thể viết `a=15` để khai báo một giá trị nguyên trong một biến.

Với Python, việc phát triển ứng dụng và debug trở nên nhanh hơn bởi vì không cần đến bước biên dịch và chu trình edit-test-debug của Python là rất nhanh.

Các đặc điểm của Python

Dưới đây là một số đặc điểm chính của Python:

- **Dễ dàng để sử dụng:** Python là một ngôn ngữ bậc cao rất dễ dàng để sử dụng. Python có một số lượng từ khóa ít hơn, cấu trúc của Python đơn giản hơn và cú pháp của Python được định nghĩa khá rõ ràng, ... Tất cả các điều này là Python thực sự trở thành một ngôn ngữ thân thiện với lập trình viên.
- Bạn có thể **đọc code của Python khá dễ dàng**. Phần code của Python được định nghĩa khá rõ ràng và rành mạch.
- Python có một **thư viện chuẩn khá rộng lớn**. Thư viện này dễ dàng tương thích và tích hợp với UNIX, Windows, và Macintosh.
- Python là một **ngôn ngữ thông dịch**. Trình thông dịch thực thi code theo từng dòng (và bạn không cần phải biên dịch ra file chạy), điều này giúp cho

quá trình debug trở nên dễ dàng hơn và đây cũng là yếu tố khá quan trọng giúp Python thu hút được nhiều người học và trở nên khá phổ biến.

- Python cũng là một ngôn ngữ lập trình **hướng đối tượng**. Ngoài ra, Python còn hỗ trợ các phương thức lập trình theo hàm và theo cấu trúc. Ngoài các đặc điểm trên, Python còn khá nhiều đặc điểm khác như hỗ trợ lập trình GUI, mã nguồn mở, có thể tích hợp với các ngôn ngữ lập trình khác, ...

Lịch sử của Python

Python được phát triển bởi Guido Van Rossum vào cuối những năm 80 và đầu những năm 90 tại Viện toán-tin ở Hà Lan. Python kế thừa từ nhiều ngôn ngữ như ABC, Module-3, C, C++, Unix Shell, ...

Ngôn ngữ Python được cập nhật khá thường xuyên để thêm các tính năng và hỗ trợ mới. Phiên bản mới nhất hiện nay của Python là **Python 3.3** được công bố vào 29/9/2012 với nguyên tắc chủ đạo là "bỏ cách làm việc cũ nhằm hạn chế trùng lặp về mặt chức năng của Python".

Điều khiển luồng trong Python

Các bạn cũng đã khá quen thuộc với các lệnh điều khiển luồng trong C, C++ như if, if else, ... Chương này sẽ trình bày về các lệnh điều khiển luồng trong Python.

Ngôn ngữ lập trình Python coi các giá trị **khác null và khác 0** là **true**, và coi các giá trị là **null hoặc 0** là **false**.

Sau đây chúng ta cùng tìm hiểu các lệnh điều khiển luồng trong Python:

Lệnh if trong Python

Lệnh if trong Python là giống như trong ngôn ngữ C. Lệnh này được sử dụng để kiểm tra một điều kiện, nếu điều kiện là true thì lệnh của khối if sẽ được thực thi, nếu không nó sẽ bị bỏ qua.

Bạn theo link sau để tìm hiểu chi tiết về **Lệnh if trong Python**.

Lệnh if...else trong Python

Một lệnh else có thể được sử dụng kết hợp với lệnh if. Một lệnh else chứa khối code mà thực thi nếu biểu thức điều kiện trong lệnh if được ước lượng là 0 hoặc một giá trị false. Lệnh else là lệnh tùy ý và chỉ có duy nhất một lệnh else sau lệnh if.

Bạn theo link sau để tìm hiểu chi tiết về **Lệnh if...else trong Python**.

Lồng các lệnh if trong Python

Đôi khi có một tình huống là khi bạn muốn kiểm tra thêm một điều kiện khác sau khi một điều kiện đã được ước lượng là true. Trong tình huống như vậy, bạn có thể sử dụng các lệnh if lồng nhau trong Python.

Trong cấu trúc các lệnh if lồng nhau, bạn có thể có cấu trúc if...elif...else bên trong cấu trúc if...elif...else khác.

Bạn theo link sau để tìm hiểu chi tiết về **Lồng các lệnh if trong Python**.

Trường hợp các Suite lệnh đơn

Bạn đã tìm hiểu về suite trong chương Cú pháp cơ bản. Nếu suite của một mệnh đề **if** chỉ bao gồm một dòng lệnh đơn, thì nó có thể ở trên cùng một dòng như là header của lệnh.

Dưới đây là ví dụ đơn giản về một **mệnh đề if một dòng**:

```
var = 6677028

if ( var == 6677028 ) : print "Gia tri cua bieu thuc la 6677028"

print "Beautiful!"
```

Khi code trên được thực thi sẽ cho kết quả:

```
Gia tri cua bieu thuc la 6677028
Beautiful!
```

Lệnh if trong Python

Lệnh `if` trong Python là giống như trong ngôn ngữ C. Lệnh này được sử dụng để kiểm tra một điều kiện, nếu điều kiện là `true` thì lệnh của khối `if` sẽ được thực thi, nếu không nó sẽ bị bỏ qua.

Cú pháp của lệnh `if` là:

```
if biểu_thức:  
    các_lệnh
```

Ví dụ của lệnh `if` trong Python:

```
var1 = 100  
  
if var1:  
    print "1 - Nhan mot gia tri true"  
    print var1  
  
var2 = 0  
  
if var2:  
    print "2 - Nhan mot gia tri true"  
    print var2  
  
print "Good bye!"
```

Khi code trên được thực thi sẽ cho kết quả:

```
1 - Nhan mot gia tri true  
100  
Good bye!
```

Lệnh `if...elif...else` trong Python

Một lệnh **`else`** có thể được sử dụng kết hợp với lệnh **`if`**. Một lệnh `else` chứa khối code mà thực thi nếu biểu thức điều kiện trong lệnh `if` được ước lượng là 0 hoặc một giá trị `false`. Lệnh `else` là lệnh tùy ý và chỉ có **duy nhất** một lệnh `else` sau lệnh `if`.

Cú pháp của lệnh `if...else` là:

```
if biểu_thức:
```

```
    cac_lenh
else:
    cac_lenh
```

Dưới đây là ví dụ minh họa lệnh if...else trong Python:

```
var1 = 100
if var1:
    print "1 - Nhan mot gia tri true"
    print var1
else:
    print "1 - Nhan mot gia tri false"
    print var1

var2 = 0
if var2:
    print "2 - Nhan mot gia tri true"
    print var2
else:
    print "2 - Nhan mot gia tri false"
    print var2

print "Good bye!"
```

Khi code trên được thực thi sẽ cho kết quả:

```
1 - Nhan mot gia tri true
100
2 - Nhan mot gia tri false
0
Good bye!
```

Lệnh *elif* trong Python

Lệnh *elif* cho phép bạn kiểm tra nhiều điều kiện và thực thi khối code ngay khi một trong các điều kiện được ước lượng là true. Cũng giống như lệnh *else*, lệnh *elif* là tùy ý. Tuy nhiên, không giống *else* mà chỉ có một lệnh được theo sau *if*, thì bạn có thể sử dụng **nhiều *elif* theo sau *if***.