

TÁC GIẢ: 8 SYNC

CỘNG ĐỒNG



[Kevin Nguyễn](#)



[Nhóm Chia Sẻ Công Nghệ](#)



[Nhóm BlockChain](#)



[Tiktok: 8 Sync](#)



[Youtube: 8 Sync Dev](#)



[Zalo](#)

KHÓA HỌC:



[Fullstack Python](#)



[Fullstack Nextjs](#)



[Fullstack Android-IOs](#)

Tài liệu sẽ được cập nhật định kì và thông báo trong group nên các bạn chú ý nhen .



PYTHON - HƯỚNG DẪN VỀ CÚ PHÁP

Trong hướng dẫn này, chúng ta sẽ tìm hiểu về cú pháp trong Python và cách tạo và kích hoạt một môi trường ảo để xây dựng ứng dụng Python.

MÔI TRƯỜNG ẢO PYTHON

Môi trường ảo Python tạo ra một cài đặt ảo của Python trong một thư mục dự án. Người dùng sau đó có thể cài đặt và quản lý các gói Python cho mỗi dự án. Điều này cho phép người dùng cài đặt gói và sửa đổi môi trường Python mà không lo sợ làm hỏng các gói được cài đặt trong các môi trường khác.

KHÁI NIỆM VỀ MÔI TRƯỜNG ẢO TRONG PYTHON

Một môi trường ảo Python là:

- Được coi là có thể bỏ đi.
- Được sử dụng để chứa một trình thông dịch Python cụ thể và các thư viện và tập lệnh cần thiết để hỗ trợ một dự án.
- Được chứa trong một thư mục, thường là có tên là venv hoặc .venv trong thư mục dự án.
- Không được coi là có thể di chuyển hoặc sao chép.

TẠO MÔI TRƯỜNG ẢO TRONG PYTHON SỬ DỤNG VENV

Chức năng này được hỗ trợ bởi module venv trong bản phân phối Python tiêu chuẩn. Sử dụng các lệnh sau để tạo một môi trường ảo mới.



< 8 Sync Dev />

```
mkdir pythonapp  
cd pythonapp  
python -m venv myvenv
```

Ở đây, myvenv là thư mục mà một môi trường ảo Python mới sẽ được tạo ra với cấu trúc thư mục như sau:



< 8 Sync Dev />


```
Directory of C:\pythonapp\myvenv
```

```
22-02-2023 09:53 <DIR> .
```

```
22-02-2023 09:53 <DIR> ..
```

```
22-02-2023 09:53 <DIR> Include
```

```
22-02-2023 09:53 <DIR> Lib
```

```
22-02-2023 09:53 77 pyvenv.cfg
```

```
22-02-2023 09:53 <DIR> Scripts
```

Các tiện ích để kích hoạt và vô hiệu hóa môi trường ảo cũng như bản sao cục bộ của trình thông dịch Python sẽ được đặt trong thư mục Scripts.

KÍCH HOẠT MÔI TRƯỜNG ẢO

Để kích hoạt môi trường ảo mới này, thực thi activate.bat trong thư mục Scripts.



< 8 Sync Dev />

```
myvenv\scripts\activate
```

Lưu ý tên của môi trường ảo trong dấu ngoặc đơn. Thư mục Scripts chứa một bản sao cục bộ của trình thông dịch Python. Bạn có thể bắt đầu một phiên Python trong môi trường ảo này.

KIỂM TRA PYTHON ĐANG CHẠY TRONG MỘT MÔI TRƯỜNG ẢO HAY KHÔNG?

Để xác nhận liệu phiên Python này có trong môi trường ảo hay không, kiểm tra sys.path.



< 8 Sync Dev />

```
python
```



< 8 Sync Dev />

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC  
v.1929
```

```
64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>> import sys
```

```
>>> sys.path
```

```
['', 'C:\\Python310\\python310.zip', 'C:\\Python310\\DLLs',  
'C:\\Python310\\lib', 'C:\\Python310', 'C:\\pythonapp\\myvenv',  
'C:\\pythonapp\\myvenv\\lib\\site-packages']
```

Thư mục Scripts của môi trường ảo này cũng chứa các tiện ích pip. Nếu bạn cài đặt một gói từ PyPI, gói đó sẽ chỉ hoạt động trong môi trường ảo hiện tại.

PYTHON - CÚ PHÁP

CHƯƠNG TRÌNH PYTHON ĐẦU TIÊN

Chúng ta hãy thực thi một chương trình Python để in ra "Hello, World!" ở hai chế độ lập trình Python khác nhau. (a) Lập trình Chế Độ Tương Tác (b) Lập Trình Chế Độ Kịch Bản.

PYTHON - LẬP TRÌNH CHẾ ĐỘ TƯƠNG TÁC

Chúng ta có thể gọi trình thông dịch Python từ dòng lệnh bằng cách nhập python tại dấu nhắc lệnh như sau:



< 8 Sync Dev />


```
$ python3
```



< 8 Sync Dev />

```
Python 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] on  
linux  
Type "help", "copyright", "credits" or "license" for more  
information.  
>>>
```

Ở đây >>> cho biết Dấu Nhắc Lệnh Python nơi bạn có thể nhập các lệnh của mình. Hãy nhập văn bản sau tại dấu nhắc Python và nhấn Enter:



< 8 Sync Dev />

```
>>> print ("Hello, World!")
```

Nếu bạn đang chạy phiên bản cũ hơn của Python, như Python 2.4.x, thì bạn cần sử dụng câu lệnh print mà không có dấu ngoặc đơn như in print "Hello, World!". Tuy nhiên, trong phiên bản Python 3.x, điều này tạo ra kết quả sau:



< 8 Sync Dev />

```
Hello, World!
```

PYTHON - LẬP TRÌNH CHẾ ĐỘ KỊCH BẢN

Chúng ta có thể gọi trình thông dịch Python với tham số script để bắt đầu thực thi kịch bản và tiếp tục cho đến khi kịch bản kết thúc. Khi kịch bản kết thúc, trình thông dịch không còn hoạt động nữa.

Hãy viết một chương trình Python đơn giản trong một tập tin script, là một tệp văn bản đơn giản. Các tệp Python có phần mở rộng .py. Nhập mã nguồn sau vào một tệp test.py –



< 8 Sync Dev />

```
print ("Hello, World!")
```

Chúng ta giả sử rằng bạn đã cài đặt đường dẫn trình thông dịch Python trong biến PATH. Bây giờ, hãy thử chạy chương trình này như sau:



< 8 Sync Dev />

```
$ python3 test.py
```

Kết quả sẽ là:



< 8 Sync Dev />

```
Hello, World!
```

Hãy thử một cách khác để thực thi một kịch bản Python. Đây là tệp test.py được sửa đổi –



< 8 Sync Dev />

```
#!/usr/bin/python3
```

```
print ("Hello, World!")
```

Chúng ta giả sử rằng bạn đã cài đặt trình thông dịch Python có sẵn trong thư mục /usr/bin. Bây giờ, hãy thử chạy chương trình này như sau:

< 8 Sync Dev />

```
$ chmod +x test.py      # Điều này làm cho tệp có thể thực thi  
được  
$ ./test.py
```

Kết quả sẽ là:

< 8 Sync Dev />

Hello, World!

NHẬN DIỆN BIẾN TRONG PYTHON

Một nhận dạng Python là một tên được sử dụng để xác định một biến, hàm, lớp, module hoặc đối tượng khác. Một nhận dạng bắt đầu bằng một chữ cái A đến Z hoặc a đến z hoặc một gạch dưới (_) theo sau là không hoặc nhiều chữ cái, gạch dưới và chữ số (0 đến 9).

Python không cho phép các ký tự dấu chấm câu như @, \$ và % trong nhận dạng.

Python là một ngôn ngữ lập trình phân biệt chữ hoa chữ thường. Do đó, Manpower và manpower là hai nhận dạng khác nhau trong Python.

Dưới đây là các quy tắc đặt tên cho các nhận dạng Python:

- Tên lớp Python bắt đầu bằng một chữ cái in hoa. Tất cả các nhận dạng khác bắt đầu bằng một chữ cái thường.
- Bắt đầu một nhận dạng với một gạch dưới đơn làm cho nhận dạng là một nhận dạng riêng tư.
- Bắt đầu một nhận dạng với hai gạch dưới đầu tiên chỉ ra một nhận dạng mạnh mẽ riêng tư.
- Nếu nhận dạng cũng kết thúc với hai gạch dưới, nhận dạng là một tên đặc biệt được định nghĩa bởi ngôn ngữ.

TỪ KHÓA ĐÃ ĐƯỢC DÀNH RIÊNG TRONG PYTHON

Danh sách sau đây hiển thị các từ khóa Python. Đây là các từ khóa được dành riêng và bạn không thể sử dụng chúng như tên hằng số hoặc biến hoặc bất kỳ nhận dạng nào khác. Tất cả các từ khóa Python chứa các chữ cái thường.



< 8 Sync Dev />

and	as	assert
break	class	continue
def	del	elif
else	except	False
finally	for	from
global	if	import
in	is	lambda
None	nonlocal	not
or	pass	