

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 .



BIẾN TRONG PYTHON

Biến Python là các vị trí bộ nhớ được dành riêng để lưu trữ các giá trị trong chương trình Python. Điều này có nghĩa là khi bạn tạo một biến, bạn đặt dành một số không gian trong bộ nhớ.

Dựa trên kiểu dữ liệu của một biến, trình thông dịch Python phân bổ bộ nhớ và quyết định những gì có thể được lưu trữ trong bộ nhớ dành riêng đó. Do đó, bằng cách gán các kiểu dữ liệu khác nhau cho các biến Python, bạn có thể lưu trữ số nguyên, số thập phân hoặc ký tự trong các biến này.

ĐỊA CHỈ BỘ NHỚ

Các mục dữ liệu thuộc các loại dữ liệu khác nhau được lưu trữ trong bộ nhớ của máy tính. Các vị trí bộ nhớ của máy tính có một số hoặc địa chỉ, được biểu diễn bên trong dưới dạng nhị phân. Dữ liệu cũng được lưu trữ dưới dạng nhị phân vì máy tính hoạt động theo nguyên lý biểu diễn nhị phân. Trong biểu đồ dưới đây, một chuỗi May và một số 18 được hiển thị là được lưu trữ tại các vị trí bộ nhớ.

[Memory Diagram]

Nếu bạn biết ngôn ngữ hợp ngữ, bạn sẽ chuyển đổi các mục dữ liệu này và địa chỉ bộ nhớ, và đưa ra một hướng dẫn ngôn ngữ máy. Tuy nhiên, điều này không dễ dàng đối với mọi người. Trình dịch ngôn ngữ như trình thông dịch Python thực hiện loại chuyển đổi này. Nó lưu trữ đối tượng trong một vị trí bộ nhớ được chọn ngẫu nhiên. Hàm `id()` được tích hợp sẵn trong Python trả về địa chỉ nơi đối tượng được lưu trữ.



< 8 Sync Dev />

```
>>> "May"  
May  
>>> id("May")  
2167264641264
```

```
>>> 18  
18  
>>> id(18)  
140714055169352
```

Khi dữ liệu được lưu trữ trong bộ nhớ, nó nên được truy cập lặp đi lặp lại để thực hiện một quy trình nhất định. Rõ ràng, việc trích xuất dữ liệu từ ID của nó là rườm

rà. Ngôn ngữ cao cấp như Python giúp cho việc đặt một biệt danh hoặc nhãn phù hợp để tham chiếu đến vị trí bộ nhớ.

Trong ví dụ trên, hãy đặt nhãn cho vị trí của May là month và vị trí mà 18 được lưu trữ là age. Python sử dụng toán tử gán (=) để liên kết một đối tượng với nhãn.



< 8 Sync Dev />

```
>>> month = "May"
>>> age = 18
```

Đối tượng dữ liệu (May) và tên của nó (month) có cùng id(). id() của 18 và age cũng giống nhau.



< 8 Sync Dev />

```
>>> id(month)
2167264641264
>>> id(age)
140714055169352
```

Nhãn là một nhận dạng. Thông thường, nó được gọi là một biến. Một biến Python là một tên biểu tượng là một tham chiếu hoặc con trỏ đến một đối tượng.

TẠO BIẾN PYTHON

Biến Python không cần khai báo rõ ràng để đặt dành không gian bộ nhớ hoặc bạn có thể nói để tạo ra một biến. Một biến Python được tạo ra tự động khi bạn gán một giá trị cho nó. Dấu bằng (=) được sử dụng để gán giá trị cho biến.

Toán hạng bên trái của toán tử = là tên của biến và toán hạng bên phải của toán tử = là giá trị được lưu trữ trong biến. Ví dụ –

VÍ DỤ TẠO BIẾN PYTHON

Ví dụ này tạo ra các loại biến khác nhau (một số nguyên, một số thực và một chuỗi).



< 8 Sync Dev />

```
counter = 100          # Tạo ra một biến số nguyên
miles = 1000.0         # Tạo ra một biến số thực
name = "8 Sync Dev"    # Tạo ra một biến chuỗi
```

IN CÁC BIẾN PYTHON

Sau khi chúng ta tạo một biến Python và gán một giá trị cho nó, chúng ta có thể in nó bằng cách sử dụng hàm print(). Dưới đây là phần mở rộng của ví dụ trước và chỉ ra cách in các biến khác nhau trong Python:

VÍ DỤ IN CÁC BIẾN PYTHON

Ví dụ này in các biến.



< 8 Sync Dev />

```
counter = 100          # Tạo ra một biến số nguyên
miles = 1000.0         # Tạo ra một biến số thực
name = "8 Sync Dev"    # Tạo ra một biến chuỗi

print(counter)
print(miles)
print(name)
```

Ở đây, 100, 1000.0 và "8 Sync Dev" là các giá trị được gán cho các biến counter, miles và name, tương ứng. Khi chạy chương trình Python trên, điều này tạo ra kết