Bài: Biến trong Python

Xem bài học trên website để ủng hộ Kteam: Biến trong Python

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Trong các bài trước, **Kteam** đã giới thiệu đến bạn cách CHẠY MỘT CHƯƠNG TRÌNH PYTHON. Bên cạnh đó, chúng ta cũng đã biết CÁCH GHI CHÚ TRONG PYTHON.

Ở bài này, Kteam sẽ giới thiệu với các bạn về **BIẾN TRONG PYTHON**. Một phần tuy cơ bản nhưng rất quan trọng trong lập trình và nó sẽ theo bạn mãi cho tới khi bạn còn code.

Nội dung chính

Để đọc hiểu bài này tốt nhất bạn cần:

- Cài đặt sẵn MÔI TRƯỜNG PHÁT TRIỂN CỦA PYTHON.
- Xem qua bài CÁCH CHẠY CHƯƠNG TRÌNH PYTHON.
- Nắm CÁCH GHI CHÚ TRONG PYTHON.

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Biến là gì?
- Tại sao lại cần biến?
- Khởi tạo biến trong Python.
- Cách kiểm tra kiểu dữ liệu của biến.

Biến là gì?

Nếu bạn từng làm các bài toán đại số thì các bạn luôn phải chạm mặt các biến như là biến x, biến y, biến a, biến b,... Và như bạn thấy nó chả có giá trị cụ thể.

Trong lập trình, biến (variable) là tên của một vùng trong bộ nhớ RAM, được sử dụng để lưu trữ thông tin. Bạn có thể gán thông tin cho một biến, và có thể lấy thông tin đó ra để sử dụng. Khi một biến được khai báo, một vùng trong bộ nhớ sẽ dành cho các biến.

Biến là một thứ cực kì quan trọng trong lập trình mà không thể thiếu trong bất cứ chương trình lớn, nhỏ nào.

Tại sao lại cần biến?

Biến giúp chúng ta lưu trữ các dữ liệu và cho phép chúng ta lấy các dữ liệu của chúng để tính toán được thuận tiện và chính xác hơn.

Hãy tưởng tượng như sau, bạn có một số dữ liệu là những con số với nhiều chữ số và các thao tác tính toán

Python:

```
# cộng hai số
>>> 52348252408 + 523482034
52871734442
# tiếp tục thực hiện việc tính toán
>>> 52871734442 + 412312323
53284046765
```

Một điều mà các bạn dễ dàng nhận ra đó là những con số với nhiều chữ số gây khó khăn trong việc sử dụng vì chúng có quá nhiều chữ số, đôi lúc chúng ta cũng có thể vô tình gây sai lệnh giá trị.

Ta hãy giải quyết bài toán trên khi nhờ tới sự giúp đỡ của các biến

Python:

```
# lưu giá trị 52348252408 cho biến a
>>> a = 52348252408
# lưu giá trị 523482034 cho biến b
>>> b = 523482034
# cộng giá trị hai biến a và b, sau đó lưu vào biến c
>>> c = a + b
# lưu giá trị 412312323 cho biến d
>>> d = 412312323
# cộng giá trị biến c với giá trị biến d
>>> c + d
53284046765
```

Dễ thấy, ta cũng được kết quả tương tự, nhưng lại dễ dàng tính toán, giảm thiểu tỉ lệ sai lệnh giá trị hơn khi không sử dụng tới biến.

Khởi tạo biến trong Python

Những thứ cần biết về tên của biến

- Tên của biến không được bắt đầu bằng số
- Tên biến không được trùng với các từ khóa của Python

Một số từ khóa của Python

```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try
```

- Tên của biến chỉ chứa các chữ cái, số và '_'
- Tên biến trong Python có phân biệt chữ in hoa và in thường. Ví dụ: PI, Pi, pI, pi là 4 tên biến khác nhau.

Khởi tạo một biến trong Python

Cú pháp:

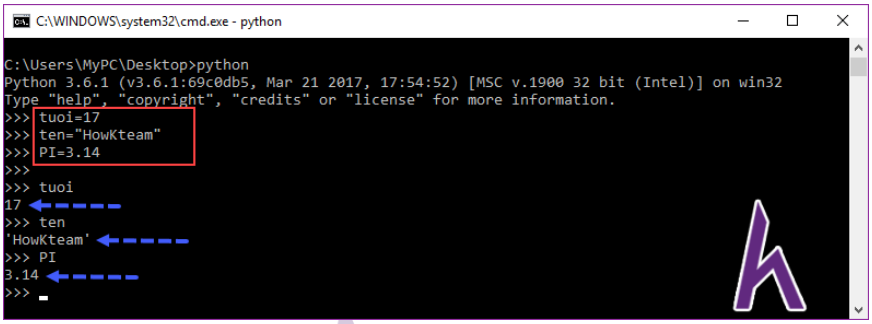
```
<tên biến> = <giá trị của biến>
```

Ví dụ:

Python:

```
# Đoạn code sau đây khai báo 3 biến tuoi, ten và PI và giá trị của chúng
>>>tuoi = 17
>>>ten = “How Kteam”
>>>PI = 3.14
```

Kết quả:



Giải thích ví dụ:

Những dòng có **khung đỏ** đó chính là dòng lệnh dùng để **khai báo**. Còn dòng **mũi tên màu xanh** chính là **kết quả** của biến.

"ten", "tuoi", "PI" chính là những tên biến còn những thứ sau dấu bằng như là "17", "How Kteam", "3.14" đó chính là giá trị của biến.

Khởi tạo nhiều biến

Cú pháp:

> <tên biến thứ nhất>, <tên biến thứ hai>, ..,<tên tên biến thứ n> = <giá trị biến thứ nhất>, <giá trị biến thứ hai>, .., <giá trị biến thứ n>

Ví dụ:

Python:

```
# khai báo 3 biến tuoi, ten và PI và giá trị của chúng trên cùng một dòng
>>> tuoi, ten, PI = 17, "How Kteam", 3.14
```

Kết quả:

Giải thích ví dụ:

Ở ví dụ trên '**tuoi**' là biến thứ nhất, '**ten**' là biến thứ hai và '**PI**' là biến thứ ba. Và sau đó '**17**' là giá trị biến thứ nhất, "**How Kteam**" là giá trị biến thứ hai và "**3.14**" là giá trị biến thứ ba. Do đó dòng code trên cũng tương tự như:

Python:

```
tuoi = 17 # biến thứ nhất lấy giá trị biến thứ nhất
ten = “How Kteam” # biến thứ hai lấy giá trị biến thứ hai
PI = 3.14 # biến thứ ba lấy giá trị biến thứ ba
```

Kiểm tra kiểu dữ liệu của biến

Không như đa số các ngôn ngữ lập trình khác, khi khai báo biến phải đi kèm với kiểu dữ liệu. Trong Python việc khai báo kiểu dữ liệu cho biến không cần thiết mà Python sẽ tự biết kiểu dữ liệu của giá trị gán cho biến.

Vậy để kiểm tra kiểu dữ liệu giá trị của một biến đã khởi tạo, ta sử dụng hàm **type()**

Cú pháp:

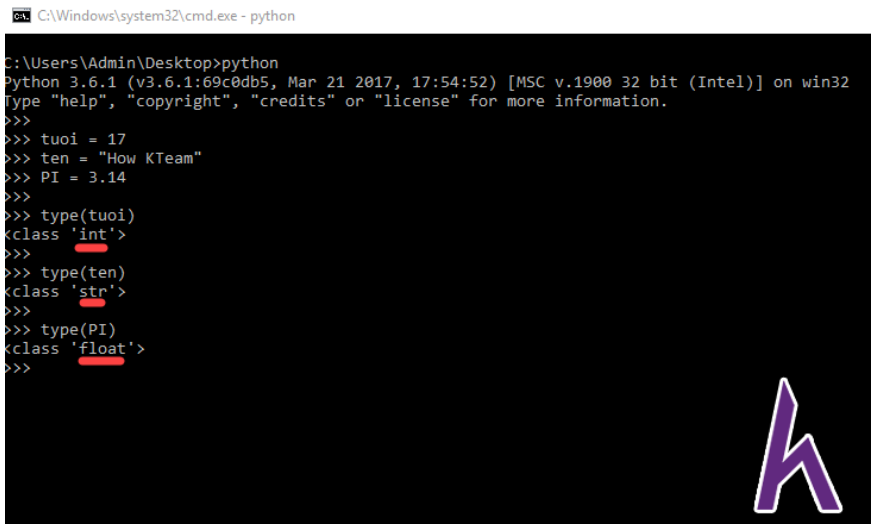
> **type**(<tên biến>)

Ví dụ:

Python:

```
tuoi = 17

ten = “How KTeam”

PI = 3.14

type(tuoi) # kiểm tra kiểu dữ liệu giá trị của biến tuoi

type(ten) # kiểm tra kiểu dữ liệu giá trị của biến ten

type(PI) # kiểm tra kiểu dữ liệu giá trị của biến PI
```

Kết quả:



Như bạn thấy ở ví dụ trên kết quả tra ra một số kiểu dữ liệu đó là ‘int’, ‘str’, ‘float’. Đó là các kiểu dữ liệu phổ biến trong các ngôn ngữ lập trình hiện nay. Kteam sẽ giới thiệu những kiểu dữ liệu này ở những bài sau.

Kết luận

Qua bài viết này, bạn đã nắm được về những thứ cơ bản về BIẾN TRONG PYTHON (variable).

Các bài tiếp theo mình sẽ giới thiệu về KIỂU DỮ LIỆU SỐ TRONG PYTHON.

Cảm ơn bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".