Biến trong Python là gì?

Một biến Python là một khu vực bộ nhớ được dành riêng để lưu trữ các giá trị. Nói cách khác, biến trong python cung cấp dữ liệu cho quá trình xử lý của máy tính.

Biến đóng vai trò rất quan trọng, nếu không có biến thì bạn không thể tạo ra một chương trình hoàn hảo, nói cách khác bạn không thể giải quyết bất kì một bài toán nào cả, kể cả một ứng dụng đơn giản.

Khái niệm về biến không phải chỉ tồn tại ở Python, mà ngay cả những ngôn ngữ khác cũng có, vì vậy đây là bài đầu tiên mà bạn phải học khi tìm hiểu một ngôn ngữ lập trình bất kì.

Mỗi giá trị trong Python có một kiểu dữ liệu. Các kiểu dữ liệu khác nhau trong Python là kiểu số, kiểu danh sách, kiểu bộ (tuple), kiểu chuỗi (string), kiểu từ điển (dictionary), v.v.

Các biến có thể được khai báo bằng bất kỳ tên nào hoặc thậm chí cả bằng chữ cái như a, aa, abc. Tuy nhiên chúng ta nên đặt tên biến có ý nghĩa để code được tốt hơn.

Ví dụ:

```
# Không nên
A = “BkaCad Online”
# Nên
name = “BkaCad Online”;
# Không nên
abc = 25;
# Nên
age = 25
```

Khai báo biến

Trong một số ngôn ngữ khác biến sẽ cần được *khai báo*. Việc khai báo biến bao gồm đặt tên cho biến và xác định kiểu dữ liệu của biến đó.

Với Python chúng ta sẽ bỏ qua bước *khai báo* thay vào đó sẽ thực hiện việc *khởi tạo* biến. Khởi tạo biến sẽ bao gồm đồng thời việc khai báo biến và gán giá trị cho biến.

Để bắt đầu nhanh khai báo biến thì ta sử dụng cú pháp như sau:

```
tenbien = giatri
```

Ví dụ khai báo biến name

```
name = "bkacad-online"
```

Trong Python khi đặt tên biến bạn cần chú ý:

- Tên biến *có thể* được bắt đầu bằng ký tự _, $ hoặc bất cứ chữ cái nào (in hoa hoặc in thường).
- Tên biến *không* được bắt đầu bằng chữ số. Ví dụ tên biến 100invalid là không hợp lệ.
- Tên biến *không* được bao gồm ký tự trống trong đó.
- *Không* sử dụng các từ được bảo lưu trong Python như print, if, else… để đặt tên cho biến. Ví dụ tên biến print là không hợp lệ, ngược lại tên biến print1, print_2, printHello… là hợp lệ.

Ví dụ về đặt tên biến đúng

```
myclass = "BKACAD_PYF"
my_class = "BKACAD_PYF"
myClass = "BKACAD_PYF"
MYCLASS = "BKACAD_PYF"
MYVAR = "BKACAD_PYF"
myclass2 = "BKACAD_PYF"
```

Ví dụ về đặt tên biến đặt sai

```
2myclass = "BKACAD_PYF"
my-class = "BKACAD_PYF"
my class = "BKACAD_PYF"
```

Lưu ý: Tên biến trong python có phân biệt hoa thường.

Kiểu dữ liệu

Dữ liệu xuất hiện trong quá trình chạy chương trình được Python lưu trữ trong bộ nhớ tạm. Dữ liệu được phân biệt theo các kiểu khác nhau, mỗi kiểu sẽ có đặc thù riêng và sẽ cần một dung lượng bộ nhớ khác nhau để lưu trữ. Các kiểu dữ liệu cơ bản của Python bao gồm:

- Kiểu Number
- Kiểu String
- Kiểu Boolean
- Kiểu Tuple
- Kiểu List
- Kiểu Dictionary

Trong bài học này chúng ta sẽ tìm hiểu về hai kiểu dữ liệu à kiểu Number và String các kiểu dữ liệu còn lại chúng ta sẽ tìm hiểu chi tiết ở các bài học tiếp theo.

Kiểu Number (Số)

Kiểu Number được sử dụng để lưu trữ các số nguyên (nguyên âm, nguyên dương, số 0), số thập phân, số thực.

Ví dụ:

```
year = 2017 // số nguyên dương
dollarExchangeRate = 22727.50 # số thập phân
```

Kiểu String (Chuỗi)

Kiểu chuỗi bao gồm một tập hợp các ký tự và thường được đặt bên trong cặp dấu nháy kép hoặc nháy đơn. Ví dụ:

```
year = "2017"
```

Chuỗi cũng có thể không có ký tự nào:

```
myString = ""
```

Để nối chuỗi chúng ta sử dụng toán tử +

```
year = "2017"
print("Năm nay là " + year)
```

Khác với nhiều ngôn ngữ lập trình khác khi chúng ta sử dụng toán tử + với một chuỗi và một số thì Python sẽ *không* tự động chuyển đổi dữ liệu kiểu số sang kiểu chuỗi sau đó thực hiện việc nối 2 chuỗi mới, thay vào đó Python sẽ báo lỗi:

```
year = 2017
print("Năm nay là " + year)
#Error: cannot concatenate str and int objects
```

Thay vào đó để thực hiện việc nối chuỗi trong trường hợp trên bạn cần sử dụng hàm chuyển đổi, ví dụ dùng hàm str() để đổi kiểu số sang chuỗi như sau:

```
year = 2017
print("Năm nay là " + str(year))
```

Kiểu Boolean và Kiểu Number

Kiểu dữ liệu boolean chỉ bao gồm hai giá trị true hoặc false và thường được dùng trong các phép toán về logic.

```
myBool = True
print(myBool) # True
```

Hoặc lấy một ví dụ khác như sau:

```
myBool = (5 < 2)
print(myBool) # False
```

Trong Python kiểu dữ liệu boolean là một kiểu con của kiểu Number, Điều này có nghĩa bạn có thể thực hiện các phép toán số học giữa hai giá trị thuộc hai kiểu dữ liệu này:

```
myBool = (5 >= 5)
print(myBool) # True
print(10 + myBool) # 11

myBool = (5 > 5)
print(myBool) # False
print(10 + myBool) # 10
```

Kiểu List

Trong Python thì list có thể được xem là một mảng danh sách các phần tử được sắp xếp với nhau theo một trật tự. Phần tử đầu tiên được đánh dấu là 0, phần tử cuối cùng được đánh dấu là tổng số phần tử trừ đi một.

Để khai báo kiểu List thì ta sử dụng cặp ngoặc vuông [], và mỗi phần tử sẽ được ngăn cách nhau bởi dấu phẩy.

Ví dụ

```
domains = ["bkcad.vn", "bkcad-online.vn", "blog.bkcad.vn"]
```

Kiểu dữ liệu của List không nhất thiết là phải giống nhau. Như ví dụ dưới đây mình vừa khai báo kiểm number vừa kiểu string cho hai phần tử.

```
infor = ["bkcad-online", 2000]
```

Quay trở lại ví dụ ở phần 2 kiểu chuỗi, thực tế thì chuỗi trong Python được lưu trữ thành dạng List.

Kiểu Tuple

Tuple cũng là một kiểu dữ liệu dạng danh sách giống như kiểu List, điểm khác biệt là Tuple sử dụng **dấu ngoặc đơn**, còn List sử dụng **dấu ngoặc vuông**.

Ví dụ dưới đây khai báo một kiểu Tuple.

```
domains = ("bkcad.vn", "bkcad-online.vn", "blog.bkcad.vn")
```

Điểm khác biệt thứ hai nữa đó là dữ liệu trong Tuple không thể thay đổi, nó giống như một hằng số vậy. Tuy nhiên nếu một phần tử của Tuple là kiểu List thì ta có thể thay đổi dữ liệu cho phần tử đó.

Kiểu dữ liệu Dictionary trong Python

Kiểu từ điển (*dictionary*) là một loại bảng băm (*hash table*), nó sẽ lưu trữ dữ liệu ở dạng key => value nên việc truy xuất cực kì dễ dàng.

Ví dụ: Khai báo thông tin bkcad online:

```
infor = {'name': 'bkcad-online','domain': 'bkcad-online.vn', 'students': 2000}
```

Vì các key được đánh theo số hoặc chuỗi nên việc sắp xếp theo đúng thứ tự trong dictionary là không có. Nói cách khác là các phần tử trong dictionary là không có thứ tự.

Toán tử

Trong lập trình một toán tử là một ký tự được sử dụng để thực hiện một phép toán số hoặc logic. Python hỗ trợ nhiều loại toán tử khác nhau. Trong phần này chúng ta sẽ tìm hiểu các toán tử phổ biến sau đây:

- Toán tử số học
- Toán tử so sánh
- Toán tử gán giá trị (cho biến)
- Toán tử logic

Toán Tử Số Học

Python hỗ trợ tất cả các toán tử số học phổ biến dùng để thực hiện cộng, trừ, nhân, chia, luỹ thừa và tính phần dư.

#	Mô Tả	Ví Dụ
+	Toán tử cộng các giá trị lại với nhau	a + b = 12
–	Toán tử trừ các giá trị lại với nhau	a – b = -2
*	Toán tử nhân các giá trị lại với nhau	a * b = 42

/	Toán tử chia các giá trị cho nhau	a / b = 0.7142857142857143
%	Toán tử chia lấy phần dư	a % b = 5
**	Toán tử mũ. a**b = ab	a ** b = 78125
//	Toán tử chia làm tròn xuống.VD:0,57 => 00.9 => 0-07 => -1-0.1 => -1	a // b = 0

Toán Tử So Sánh

Toán tử so sánh dùng để so sánh các giá trị với nhau. Ví dụ:

Ở trên bạn chú ý để so sánh ngang bằng chúng ta sử dụng hai dấu bằng == thay vì chỉ 1 dấu =.

#	Mô tả	Ví Dụ
==	So sánh giá trị của các đối số xem có bằng nhau hay không.Nếu bằng nhau thì kết quả trả về sẽ là True và ngược lại sẽ là False.	a == b # False
!=	So sánh giá trị của các đối số xem có khác nhau hay không.Nếu khác nhau thì kết quả trả về sẽ là True và ngược lại sẽ là False.	a != b # True
<	Dấu < đại diện cho phép toán nhỏ hơn, nếu đối số 1 nhỏ hơn đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a < b # True

>	Dấu > đại diện cho phép toán lớn hơn, nếu đối số 1 lớn hơn đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a > b # False
<=	Dấu > đại diện cho phép toán nhỏ hơn hoặc bằng, nếu đối số 1 nhỏ hơn hoặc bằng đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a <= b # True
>=	Dấu > đại diện cho phép toán lớn hơn hoặc bằng, nếu đối số 1 lớn hơn hoặc bằng đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a>= b # False

Gán Giá Trị

Toán tử gán giá trị dùng để thiết lập giá trị ban đầu hoặc thay đổi giá trị của biến. Một trong những toán tử gán giá trị mà chúng ta hay sử dụng ở các ví dụ trước đây là toán tử =.

#	Mô tả	Ví Dụ
=	Toán tử này dùng để gán giá trị của một đối tượng cho một giá trị	c = a (lúc này c sẽ có giá trị = 5)
+=	Toán tử này cộng rồi gắn giá trị cho đối tượng	c += a (tương đương với c = c + a)
-=	Toán tử này trừ rồi gắn giá trị cho đối tượng	c -= a (tương đương với c = c – a)
*=	Toán tử này trừ rồi gắn giá trị cho đối tượng	c *= a (tương đương với c = c * a)

/=	Toán tử này chia rồi gắn giá trị cho đối tượng	c /= a (tương đương với c = c / a)
%	Toán tử này chia hết rồi gắn giá trị cho đối tượng	c %= a (tương đương với c = c % a)
**=	Toán tử này lũy thừa rồi gắn giá trị cho đối tượng	c **= a (tương đương với c = c ** a)
//=	Toán tử này chia làm tròn rồi gắn giá trị cho đối tượng	c //= a (tương đương với c = c // a)

Toán Tử Logic

Toán tử logic dùng để thực hiện các phép toán logic gồm có:

#	Mô tả	Ví dụ
and	Nếu 2 vế của toán tử này đều là True thì kết quả sẽ là True và ngược lại nếu 1 trong 2 vế là False thì kết quả trả về sẽ là False.	A = TrueB = False print (A and B) # False
or	Nếu 1 trong 2 vế là True thì kết quả trả về sẽ là True và ngược lại nếu cả 2 vế là False thì kết quả trả về sẽ là False.	print (A or B) # True
not	Đây là dạng phủ định, nếu biểu thức là True thì nó sẽ trả về là False và ngược lại.	print (not A) # False

Tính diện tích hình tam giác

Mục tiêu

Luyện tập sử dụng biến và toán tử

Mô tả

Trong phần này chúng ta sẽ:

- Luyện tập nhập xuất dữ liệu
- Luyện tập sử dụng biến và toán tử

Hướng dẫn

Bước 1: Tạo dự án mới

Trên thanh công cụ, chọn New, chọn Python 3, sau đó đổi tên dự án AreaOfTriangle và import thư viện Math

```
import math
```

Bước 2: Nhập dữ liệu từ bàn phím và lưu vào 3 biến a, b, c

```
a = float(input("Enter the length of side a: "))
b = float(input("Enter the length of side b: "))
c = float(input("Enter the length of side c: "))
```

Bước 3: Tính diện tích tam giác bằng công thức: area = math.sqrt(s*(s-a)*(s-b)*(s-c))

Trong đó:

- s là chu vi của tam giác, được tính với công thức: “s = (a+b+c)/2”
- math.sqrt là hàm tính căn bậc hai

```
s = (a+b+c)/2
area = math.sqrt(s*(s-a)*(s-b)*(s-c))
```

Bước 4: In kết quả ra màn hình bằng lệnh print(” Area of the triangle is: “, area)

Ta có đoạn mã hoàn thiện như sau:

```
import math
a = float(input("Enter the length of side a: "))
b = float(input("Enter the length of side b: "))
c = float(input("Enter the length of side c: "))
s = (a+b+c)/2
area = math.sqrt(s*(s-a)*(s-b)*(s-c))
print(" Area of the triangle is: ", area)
```

Để hoàn thành bài tập, học viên cần:

- Đưa mã nguồn lên GitHub
- Dán link của repository lên phần nộp bài

[Bài tập] Chuyển đổi nhiệt độ

Mô tả

Xây dựng chương trình chuyển đổi nhiệt độ từ độ C sang độ F. Biết rằng độ F được nhập từ bàn phím bởi người dùng

Gợi ý

Bước 1: Dùng lệnh input để nhập dữ liệu từ bàn phím

Bước 2: Sử dụng công thức f = (9*(int(c))/5)+32 để chuyển đổi từ độ C sang độ F

Bước 3: In ra kết quả bằng lệnh print

Code tham khảo

```
c = input(" Enter temperature in Centigrade: ")

f = (9*(int(c))/5)+32

print(" Temperature in Fahrenheit is: ", f)
```

1. Để hoàn thành bài tập, học viên cần:
    - Đưa mã nguồn lên GitHub
    - Dán link của repository lên phần nộp bài