



 master ▼

...

PYF2210R1 / buoi 5 / Buoi 5 - Biến và kiểu dữ liệu - Finished.ipynb

 tam-dien buoi 6

History

 1 contributor

1496 lines (1496 sloc) | 437 KB

...

# BIẾN

## Biến là gì???

Biến là một khái niệm trừu tượng nhưng vô cùng quan trọng trong lập trình

Biến là nơi để lưu trữ những giá trị tạm thời

Mỗi lần ta khai báo một biến, chương trình sẽ sử dụng một ít vùng nhớ trong RAM

Mỗi một biến sẽ mang một loại kiểu dữ liệu riêng

Có rất nhiều kiểu dữ liệu

## Khai báo biến trong Python

Khai báo biến trong python rất đơn giản theo cách sau:

```
tên biến = giá trị biến
```

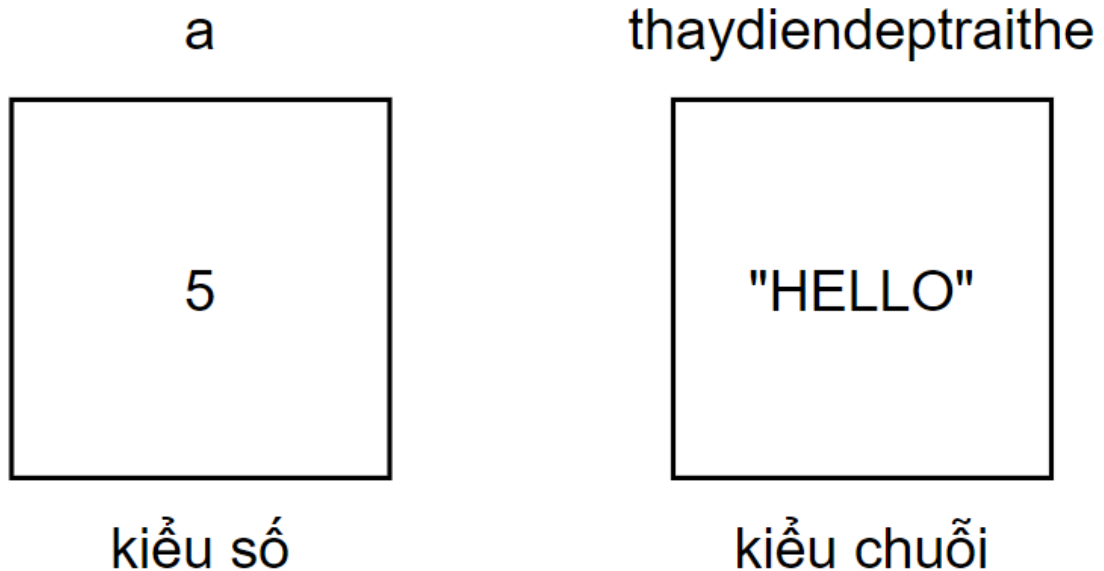
Ví dụ:

```
a = 2
```

hoặc:

```
thaydiendeptraithe = "HAHA"
```

Kiểu dữ liệu của mỗi biến, sẽ tự được định dạng khi khai báo



Nguyên tắc đặt tên biến:

1. Không bắt đầu từ số
2. Biến trong python CÓ phân biệt chữ hoa và thường
3. Không dùng ký tự đặc biệt
4. Tên biến trong được trùng với những keyword có sẵn (if, else,...)

Một số kiểu dữ liệu trong python:

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

## Kiểu số (Number)

Khai báo kiểu số:

a = 5 (Kiểu số nguyên)

a = 5.4 (Kiểu số thực)

Thực hiện một phép tính với kiểu số:

In [1]:

```
## số nguyên: interger ~~> int
## số thực: float
a = 5
b = 5.0
c = a + b
print(c)
```

10.0

## Kiểu chuỗi (String)

Kiểu chuỗi là kiểu dùng để lưu trữ các ký tự (các chữ cái, con số, ký tự đặc biệt,...)

Khai báo kiểu chuỗi:

a = "Hello mọi người"

Thực hiện phép tính cộng với kiểu chuỗi:

In [10]:

```
a = "Hello mọi người "
b = "xin chào"
c = a + b
print(c)
```

Hello mọi người xin chào

## Kiểu boolean

Kiểu boolean là kiểu chỉ lưu trữ 2 trạng thái True (đúng) và False (sai)

Khai báo kiểu boolean:

a = True

## Chuyển đổi kiểu dữ liệu

Để chuyển đổi một kiểu dữ liệu này sang một kiểu dữ liệu khác, ta chỉ cần dùng lệnh:

<kiểu dữ liệu>(<giá trị cần đổi>)

- Kiểu dữ liệu số nguyên: int

- Kiểu dữ liệu số thực: float
- Kiểu dữ liệu String: str
- Kiểu dữ liệu list: list
- Kiểu dữ liệu dictionary: dict

Ví dụ:

```
a = 2 b = str(a)
```

Lúc này, biến a là kiểu số, b là kiểu chuỗi có giá trị là "2"

Hãy thử chuyển từ kiểu chuỗi sang kiểu số

## Chuyển đổi không thành công:

In [11]:

```
a = 2
b = str(a)
### sau khi thực hiện lệnh này thì b là kiểu chuỗi còn a là kiểu số nguyên
print(a)
print(b)
```

```
2
2
```

In [13]:

```
a = 2
b = str(a)

c = a/4
d = b/4
print(c)
print(d)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In [13], line 5
      2 b = str(a)
      4 c = a/4
----> 5 d = b/4
      6 print(c)
      7 print(d)
```

**TypeError:** unsupported operand type(s) for /: 'str' and 'int'

In [14]:

```
a = 2
b = str(a)

c = a + 4
d = b + 4
print(c)
print(d)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In [14], line 5
      2 b = str(a)
      4 c = a + 4
----> 5 d = b + 4
      6 print(c)
      7 print(d)
```

**TypeError:** can only concatenate str (not "int") to str

In [15]:

```
a = 2
```

```
a = 2
b = str(a)

c = a + 4
d = b + "4"
print(c)
print(d)
```

6  
24

In [17]:

```
a = 12948.54254
b = str(a)

print(a)
print(b)

## kiểu số chuyển sang kiểu chuỗi thì luôn thành công
```

12948.54254  
12948.54254

In [18]:

```
a = "7"
b = int(a)

### a là kiểu chuỗi, b là kiểu số nguyên

print(a)
print(b)
```

7  
7

In [19]:

```
a = "7b"
b = int(a)

### không thể chuyển từ kiểu chuỗi sang kiểu số
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In [19], line 2
      1 a = "7b"
----> 2 b = int(a)
```

**ValueError:** invalid literal for int() with base 10: '7b'

In [20]:

```
a = "7.2"
b = int(a)

print(a)
print(b)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In [20], line 2
      1 a = "7.2"
----> 2 b = int(a)
      4 print(a)
      5 print(b)
```

**ValueError:** invalid literal for int() with base 10: '7.2'

In [21]:

```
a = "7.0"
b = int(a)
```

ValueError

Traceback (most recent call last)

Cell In [21], line 2

```
1 a = "7.0"  
----> 2 b = int(a)
```

ValueError: invalid literal for int() with base 10: '7.0'

In [22]:

```
a = "7.2"  
b = float(a)  
  
print(a)  
print(b)
```

7.2

7.2

In [23]:

```
a = "7"  
b = float(a)  
  
print(a)  
print(b)
```

7

7.0

In [24]:

```
a = "7.2"  
b = float(a)  
c = int(b)  
  
print(a)  
print(b)  
print(c)
```

7.2

7.2

7

In [25]:

```
a = "7.9"  
b = float(a)  
c = int(b)  
  
print(a)  
print(b)  
print(c)
```

7.9

7.9

7

In [26]:

```
a = "7.9"  
b = float(a)  
c = round(b)  
  
print(a)  
print(b)  
print(c)
```

7.9

7.9

8

In [29]:

```
a = 10 ## a đang là kiểu số nguyên  
a = "abc" ## a sẽ chuyển sang kiểu chuỗi  
print(a)
```

abc

In [30]:

```
a = "10"
a = int(a)
b = a + 15
print(b)
```

25

In [31]:

```
a = "10"
print(type(a))
```

<class 'str'>

In [32]:

```
a = 10
print(type(a))
```

<class 'int'>

In [33]:

```
a = 10.5
print(type(a))
```

<class 'float'>

In [38]:

```
### nhấ'n tổ hợp phím Ctrl + / để comment 1 hoặc nhiều dòng lệnh

a = "7.9"
b = float(a)
c = round(b)

# print(a)
# print(b)
# print(c)
```

In [45]:

```
a = 12000000
b = 0.1*a
c = f"{a:,}".replace(",", ".") + " đ"
print(c)
```

12.000.000 đ

## Chuyển đổi thành công:

In [ ]:

# 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

## Các toán tử phổ biến

- 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, lũy 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. $a ** b = a^b$	$a ** b = 78125$
//	Toán tử chia làm tròn xuống.VD: $0.57 \Rightarrow 0.9 \Rightarrow 0$ - $07 \Rightarrow -1$ - $0.1 \Rightarrow -1$	$a // b = 0$

Thực hiện một phép chia lấy phần dư:

In [48]:

```
a = 10
b = a % 7
print(b)
```

3

In [49]:

```
a = 17
b = a % 15
print(b)
```

2

In [ ]:

```
# a = ???
# b = ???
### thầy không biết a và b bằng bao nhiêu nhưng thầy biết a chia hết cho b
### thì a % b bằng bao nhiêu?
```

In [ ]:

```
### nếu a % b == 0 thì a chia hết cho b
### nếu a % b != 0 thì a không chia hết cho b
```

Thực hiện một phép tính bất kỳ rồi lưu vào biến:

In [50]:

```
a = 2**3
print(a)
```

8



In [51]:

```
a = 17
b = a // 15
print(b)
```

1

In [ ]:

```
### thâ'y cho b = a % 5
### thâ'y muố'n b bằ'ng 5 thì a phả'i bằ'ng bao nhiêu?
### khô'ng tìm đượ'c a lý do vì chia lấ'y dư cho 5 thì a sẽ nhỏ hơn 5
```

In [ ]:

```
### a = ?
### nếu a khô'ng phả'i là số' nguyên thì a % 1 khô'ng bằ'ng 0
### nếu a là số' nguyên thì a % 1 bằ'ng 0
```

## Toán tử so sánh

Khi thực hiện một toán tử so sánh, kết quả sẽ trả về là True (đúng) hoặc False (sai)

#	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

Thực hiện thử một phép so sánh:

In [54]:

```
## toán tử so sánh sẽ cho ra kết quả mang kiểu dữ liệu là boolean (kết quả chỉ có 1 tr
a = 7 == 8
print(a)
```

False

In [55]:

```
a = 15
b = a > 7
```

```
print(b)
```

True

In [56]:

```
a = 19
b = a >= 19

print(b)
```

True

## Toán tử gán giá trị

Toán tử gán giá trị dùng để thiết lập giá trị ban đầu hoặc thay đổi giá trị của biến

#	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 gán giá trị cho đối tượng	c += a (tương đương với c = c + a)
-=	Toán tử này trừ rồi gán giá trị cho đối tượng	c -= a (tương đương với c = c - a)
*=	Toán tử này nhân rồi gán giá trị cho đối tượng	c *= a (tương đương với c = c * a)
/=	Toán tử này chia rồi gán giá trị cho đối tượng	c /= a (tương đương với c = c / a)
%	Toán tử này chia hết rồi gán giá trị cho đối tượng	c %= a (tương đương với c = c % a)
**=	Toán tử này lũy thừa rồi gán giá trị cho đối tượng	c **= a (tương đương với c = c ** a)
//=	Toán tử này chia làm tròn rồi gán giá trị cho đối tượng	c //= a (tương đương với c = c // a)

Lệnh `a += 3` là viết tắt của lệnh `a = a + 3`, đây được gọi là phép cộng dồn

Thực hiện thử một toán tử gán giá trị:

In [58]:

```
a = 2
a = a + 3

print(a)
```

5

In [59]:

```
o = o + 10

print(o)
```

**NameError**

Traceback (most recent call last)

Cell In [59], line 1

```
----> 1 o = o + 10
      3 print(o)
```

**NameError**: name 'o' is not defined

In [60]:

```
a = 2
```

```
a = a + 3 ## phép cộng dồn, có thể viết tắt thành a += 3
```

2

In [61]:

```
a = 2
a += 3

print(a)
```

5

In [ ]:

```
a = 5
a += 7 ### a = 12
a *= 3 ### a = 36

print(a)
```

In [62]:

```
a = 5

for i in range(3):
    a *= a

### a *= a ~~> a = a * a ~~> a = 5 * 5 = 25
### a *= a ~~> a = a * a ~~> a = 25 * 25 = 625
### a *= a ~~> a = a * a ~~> a = 625 * 625 = 390625

print(a)
```

390625

In [63]:

```
a = 5
S = 0
b = 1

for i in range(a):
    S += b
    b += 1

## lầ`n 1
# S += b ~~> S = S + b = 0 + 1 = 1
# b += 1 ~~> b = b + 1 = 1 + 1 = 2
## lầ`n 2
# S += b ~~> S = S + b = 1 + 2 = 3
# b += 1 ~~> b = b + 1 = 2 + 1 = 3
## lầ`n 3
# S += b ~~> S = S + b = 3 + 3 = 6
# b += 1 ~~> b = b + 1 = 3 + 1 = 4
## lầ`n 4
# S += b ~~> S = S + b = 6 + 4 = 10
# b += 1 ~~> b = b + 1 = 4 + 1 = 5
## lầ`n 5
# S += b ~~> S = S + b = 10 + 5 = 15
# b += 1 ~~> b = b + 1 = 5 + 1 = 6

# S += 1
# S += 2
# S += 3
# S += 4
# S += 5

# S = 1 + 2 + 3 + 4 + 5

print(b)
print(S)
```

## Toán tử logic

Toán tự logic dùng để kết hợp các điều kiện

Toán tử logic trả về giá trị là True (đúng) hoặc False (sai)

#	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 = True B = 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

Thực hiện thử một phép toán tử logic:

In [ ]:

```
### toán tử logic sẽ cho ra kết quả dạng boolean (True hoặc là False)
### toán tử logic dùng để kết hợp 2 giá trị boolean

a = 17 > 5 ### True
b = 10 < 5 ### False
c = a and b ### False
print(c)
```

In [64]:

```
a = 17 > 5 ### True
b = False
c = 5 == 5 ### True
d = 17 > 19 ### False

e = a and b or c and d

### False or c and d
### True and d
### False
print(e)
```

False

In [65]:

```
a = 7
b = 10
c = 5

for i in range(10):
    c *= c

for i in range(15):
    a *= a

for i in range(8):
    b *= b

d = b > a
e = a > b or d
```

##### toán tử số học ~~> toán tử so sánh ~~> toán tử logic

```
f = e or c % 4 == 0
print(f)
```

True

## Bài tập

Viết chương trình chuyển đổi nhiệt độ từ độ C của biến t sang độ F rồi in lên màn hình

In [66]:

```
t = 37
f = t*1.8 + 32
print(f)
```

98.60000000000001

Sử dụng thư viện Turtle để vẽ một tam giác đều với độ dài các cạnh là biến a

Sau đó tính chu vi và diện tích của hình tam giác đó

In [1]:

```
a = 50

import turtle

t = turtle.Turtle()

## red, green, blue

t.color("red", "blue")

t.begin_fill()

for i in range(3):
    t.fd(a)
    t.left(120)

P = 3*a
S = a**2*(3**0.5)/4

print("Chu vi:", P)
print("Diện tích:", S)

t.end_fill()
```

Chu vi: 150

Diện tích: 1082.5317547305483

Sử dụng thư viện Turtle để vẽ một hình tròn với bán kính là biến r

Sau đó tính chu vi và diện tích của hình tròn đó

In [ ]:

```
r = 50

import turtle
t = turtle.Turtle()
t.circle(r)

P = r*2*3.14
S = r**2*3.14

print("chu vi:", P)
```

```
print("Chu vi: ", P)  
print("diện tích: ", S)
```

Cho hai biến a và b, hãy giải phương trình  $ax + b = 0$  (cho a luôn khác 0)

In [1]:

```
a = 10  
b = 5  
  
x = -b/a  
  
print(x)
```

-0.5

Cho hai biến t và h là tổng và hiệu của 2 số, hãy tính và in ra màn hình 2 số đó

In [2]:

```
t = 20  
h = 10  
  
a = (t + h)/2  
b = (t - h)/2  
  
print(a, b)
```

15.0 5.0

Cho hệ phương trình sau:

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

Cho các biến a1, a2, b1, b2, c1, c2, tính x và y (cho các biến sao cho phương trình luôn có cặp nghiệm x,y)

In [3]:

```
a1 = 1  
b1 = 1  
c1 = 2  
a2 = 2  
b2 = 3  
c2 = 5  
  
y = (c1*a2-c2*a1)/(b1*a2-b2*a1)  
x = (c1 - b1*y) / a1  
  
print(x, y)
```

1.0 1.0

Cho biến a, hãy vẽ a hình vuông

In [1]:

```
### mình cho a = 10 ~~> vẽ 10 hình vuông  
  
a = 10  
  
import turtle  
  
t = turtle.Turtle()  
  
for i in range(a):  
    for i in range(4):  
        t.fd(30)  
        t.lt(90)
```

```
t. penup()  
t. fd(40)  
t. pendown()
```