



Bài 4: Các toán tử cơ bản trong Python

TS. Trịnh Tuấn Đạt
Viện CNTT-TT, ĐHBK Hà Nội



Nội dung

1. Khái niệm
2. Các toán tử trong Python
3. Thứ tự ưu tiên toán tử



Nội dung

1. **Khái niệm**
2. Các toán tử trong Python
3. Thứ tự ưu tiên toán tử



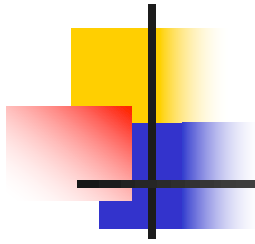
1. Khái niệm cơ bản

- **Toán tử** (operator) giống như một hàm thao tác trên các giá trị và tạo ra một giá trị mới
- VD với biểu thức $2 + 3 = 5$:
 - 2 **toán hạng** (operand) là 2 và 3
 - **Toán tử** (operator) +
- Các toán tử trong Python
 - Toán tử số học
 - Toán tử quan hệ (còn gọi là toán tử so sánh)
 - Toán tử gán
 - Toán tử logic
 - Toán tử membership
 - Toán tử identity
 - Toán tử thao tác bit



Nội dung

1. Khái niệm
2. **Các toán tử trong Python**
3. Thứ tự ưu tiên toán tử



2.1. Toán tử số học

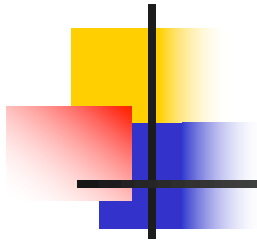
Toán tử	Miêu tả
//	Thực hiện phép chia, trong đó kết quả là thương số sau khi đã xóa các chữ số sau dấu phẩy
+	Phép cộng
-	Phép trừ
*	Phép nhân
/	Phép chia
%	Phép chia lấy phần dư
**	Phép lấy số mũ (ví dụ $2^{**}3$ cho kết quả là 8)



2.1. Toán tử số học

```
a = 10 + 3
b = 10/3
c = 10.0/3
d = 10//3
e = 10.0//3
f = 10%3
g = 10.0%3
h = 10*3
k = 10**3
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
print(g)
print(k)
```

```
13
3.3333333333333335
3.3333333333333335
3
3.0
1
1.0
30
1000
```



2.2. Toán tử quan hệ trong Python

Toán tử	Miêu tả
<	Nhỏ hơn. Nếu giá trị của toán hạng trái là nhỏ hơn giá trị của toán hạng phải, thì điều kiện trở thành true
>	Lớn hơn
<=	Nhỏ hơn hoặc bằng
>=	Lớn hơn hoặc bằng
==	Bằng
!=	Không bằng

```
>>> 10<20
True
>>> 10>20
False
>>> 10<=10
True
>>> 20>=15
True
>>> 5==6
False
>>> 5!=6
True
```


2.3. Toán tử gán

Toán tử	Miêu tả
=	Phép gán
/=	Chia toán hạng trái cho toán hạng phải, và gán kết quả cho toán hạng trái
+=	Cộng và gán
-=	Trừ và gán
*=	Nhân và gán
%=	Chia lấy phần dư và gán
**=	Lấy số mũ và gán
//=	Thực hiện phép chia // và gán

```
>>> C=10
>>> C
10
>>> C+=5
>>> C
15
>>> C-=5
>>> C
10
>>> C*=2
>>> C
20
>>> C/=2
>>> C
10
>>> C%=3
>>> C
1
>>> C=5
>>> C**=2
>>> C
25
>>> C//=2
>>> C
12
```

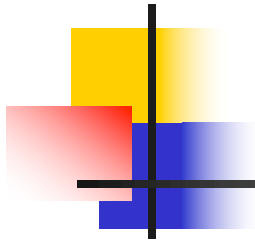


2.4. Toán tử logic

Toán tử	Miêu tả
and	Phép Và. Nếu cả hai điều kiện là true thì kết quả sẽ là true
or	Phép Hoặc. Nếu một trong hai điều kiện là true thì kết quả là true
not	Phép phủ định. Được sử dụng để đảo ngược trạng thái logic của toán hạng

```
a=5>4 and 3>2
print(a)
b=5>4 or 3<2
print(b)
c=not(5>4)
print(c)
```

```
True
True
False
```



2.5. Toán tử thao tác bit trong Python

- Toán tử thao tác bit làm việc trên các bit và thực hiện các hoạt động theo từng bit. Với $a = 60$ và $b = 13$ ($a = 0011\ 1100$ và $b = 0000\ 1101$)

Toán tử	Miêu tả	Ví dụ
& AND	Bit kết quả là 1 nếu 1 cả 2 bit toán hạng là 1. Ngược lại, là 0	$(a \& b)$ cho kết quả $0000\ 1100$
OR	Bit kết quả là 1 nếu 1 trong 2 bit toán hạng là 1	$(a b) = 61$ (tức là $0011\ 1101$)
^ XOR	Bit kết quả là 1 nếu đúng 1 trong 2 bit toán hạng là 1	$(a ^ b) = 49$ (tức là $0011\ 0001$)
~	Đây là toán tử một ngôi, được sử dụng để đảo bit	$(\sim a) = -61$ (tức là $1100\ 0011$)
<<	Toán tử dịch trái nhị phân. Giá trị của toán hạng trái được dịch chuyển sang trái một số lượng bit đã được xác định bởi toán hạng phải	$a << = 240$ (tức là $1111\ 0000$)
>>	Toán tử dịch phải nhị phân. Giá trị của toán hạng trái được dịch chuyển sang phải một số lượng bit đã được xác định bởi toán hạng phải	$a >> = 15$ (tức là $0000\ 1111$)

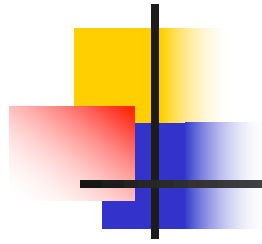
```
print (bin(60)) 0b111100
```



2.6. Toán tử Membership (thành viên)

Toán tử	Miêu tả
in	Trả về true nếu một biến là nằm trong dãy các biến, nếu không là false
not in	Trả về true nếu một biến là không nằm trong dãy các biến, nếu không là false

```
a=10
b=20
list=[10,20,30,40,50];
print (a in list)           True
print (b not in list)       False
```



2.7. Toán tử identity (định danh)

- Toán tử identity so sánh các vị trí ô nhớ của hai đối tượng

Toán tử	Miêu tả
is	Trả về true nếu các biến ở hai bên toán tử cùng trỏ tới một đối tượng, nếu không là false
is not	Trả về false nếu các biến ở hai bên toán tử cùng trỏ tới một đối tượng, nếu không là true



2.7. Toán tử identity

```
a = 20; b = 20
print ('Line 1', 'a=', a, ': ', id(a), ' b=', b, ': ', id(b))

if ( a is b ):
    print ("Line 2 - a and b have same identity")
else:
    print ("Line 2 - a and b do not have same identity")

if ( id(a) == id(b) ):
    print ("Line 3 - a and b have same identity")
else:
    print ("Line 3 - a and b do not have same identity")

b = 30
print ('Line 4', 'a=', a, ': ', id(a), ' b=', b, ': ', id(b))

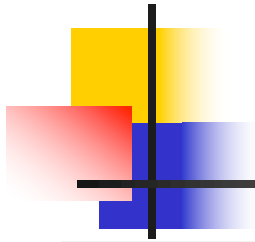
if ( a is not b ):
    print ("Line 5 - a and b do not have same identity")
else:
    print ("Line 5 - a and b have same identity")
```

```
Line 1 a= 20 : 4442510064 b= 20 : 4442510064
Line 2 - a and b have same identity
Line 3 - a and b have same identity
Line 4 a= 20 : 4442510064 b= 30 : 4442510384
Line 5 - a and b do not have same identity
```



Nội dung

1. Khái niệm
2. Các toán tử trong Python
3. **Thứ tự ưu tiên toán tử**



3. Thứ tự ưu tiên toán tử

Toán tử	Miêu tả
**	Toán tử mũ
~ + -	Phần bù; phép cộng và trừ một ngôi (với tên phương thức lần lượt là +@ và -@)
* / % //	Phép nhân, chia, lấy phần dư và phép chia //
+ -	Phép cộng và phép trừ
>> <<	Dịch bit phải và dịch bit trái
&	Phép Và Bit
^	Phép XOR và OR
<= < > >=	Các toán tử so sánh
<> == !=	Các toán tử so sánh bằng
= %= /= //= -= += *= **=	Các toán tử gán
is is not	Các toán tử Identity
in not in	Các toán tử Membership
not or and	Các toán tử logic

3. Thứ tự ưu tiên toán tử

$x = 7 + 3 * 2$

```
a = 20
b = 10
c = 15
d = 5
```

```
e = (a + b) * c / d      #( 30 * 15 ) / 5
print ("Value of (a + b) * c / d is ", e)
```

```
e = ((a + b) * c) / d    # (30 * 15) / 5
print ("Value of ((a + b) * c) / d is ", e)
```

```
e = (a + b) * (c / d)    # (30) * (15/5)
print ("Value of (a + b) * (c / d) is ", e)
```

```
e = a + (b * c) / d      # 20 + (150/5)
print ("Value of a + (b * c) / d is ", e)
```

```
Value of (a + b) * c / d is 90.0
Value of ((a + b) * c) / d is 90.0
Value of (a + b) * (c / d) is 90.0
Value of a + (b * c) / d is 50.0
```