TOÁN TỬ TRONG PYTHON

Toán tử được dùng để thực hiện các phép toán trên các biến và giá trị.

Python chia các toán tử thành các nhóm sau:

- Toán tử toán học (Arithmetic operators)
- Toán tử gán (Assignment operators)
- Toán tử so sánh (Comparison operators)
- Toán tử logic (Logical operators)
- Toán tử xác định (Identity operators)
- Toán tử quan hệ (Membership operators)
- Toán tử thao tác với bit (Bitwise operators)

1.Toán tử toán học

Các toán tử toán học được dùng với các giá trị là số để thực hiện các phép toán toán học:

Ký hiệu	Tên toán tử	Biểu thức	Ví dụ
+	Cộng	x + y	3 + 4 = 7
-	Hiệu	x - y	5 - 3 = 2
*	Tích	x * y	3 * 4 = 12
/	Chia	x / y	5 / 2 = 2.5
%	Chia lấy dư	x % y	5 % 2 = 1
**	Mũ	x ** y	5 ** 2 = 25
//	Chia lấy nguyên	x // y	5 // 2 = 2

2. Toán tử gán

Được dùng để giá trị vào biến

Toán tử	Ví dụ	Biểu thức tương đương
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5

&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

In [ ]:
```
x = 3
x += 5 #Tương tự x = x + 5 = 3 + 5 = 8
x **= 5 #Tương tự x = x**5 = 3**5 = 243
```

3. Toán tử so sánh

Được dùng để so sánh hai giá trị:

Ký hiệu	Tên toán tử	Biểu thức	Ví dụ	Kết quả
==	So sánh bằng	x == y	3 == 4	False
!=	So sánh không bằng	x != y	2 != 3	True
>	So sánh lớn hơn	x > y	3 > 4	False
<	So sánh bé hơn	x < y	5 < 2	False
>=	So sánh lớn hơn hoặc bằng	x >= y	5 >= 2	True
<=	So sánh bé hơn hoặc bằng	x <= y	5 <= 2	False

4. Toán tử logic

Dùng để kết hợp nhiều câu lệnh điều kiện lại với nhau: (x = 4)

Toán tử	Mô tả	Ví dụ	Kết quả
and	Trả về True nếu tất cả các câu lệnh điều kiện đều là True	x < 5 and x < 10	True
or	Trả về True nếu một trong các câu lệnh điều kiện là True	x < 5 or x < 4	True
not	Đảo ngược kết quả, trả về	not(x < 5 and x < 10)	False

False nếu kết quả là True

5. Toán tử xác định

Dùng để so sánh các object, không phải so sánh các object bằng nhau, mà là so sánh xem cac object có cùng là một object hay cùng vị trí bộ nhớ hay không:

Toán tử	Mô tả	Ví dụ
is	Trả về True nếu cả 2 biến **cùng** một object	x is y
is not	Trả về True nếu cả 2 biến **không** cùng một object	x is not y

In [ ]:
```
x = ["apple", "banana", "orange"]
y = ["apple", "banana", "orange"]
z = x
print(x is z)
print(x is y)
print(x == y)
print(x is not y)
```

```
True
False
True
True
```

6. Toán tử quan hệ

Dùng để kiểm tra xem một object **a** có trong một object **b** hay không:

Toán tử	Mô tả	Ví dụ
ns	Trả về True nếu object **a có trong b**	a is b
not in	Trả về True nếu object **a không tồn tại** trong object **b**	a is not b

In [ ]:
```
x = ["apple", "banana"]

print("banana" in x)
print(2 in x)
print(True not in x)
```

```
True
False
True
```

7. Toán tử thao tác với bit

Dùng để so sánh các số ở dạng nhị phân

Số thập phân	Số nhị phân
0	0000 0000
1	0000 0001
2	0000 0010

3	0000 0011	9	0000 1001
4	0000 0100	10	0000 1010
5	0000 0101	11	0000 1011
6	0000 0110	12	0000 1100
7	0000 0111	13	0000 1101
		14	0000 1110
		15	0000 1111

Số thập phân	Số nhị phân
8	0000 1000

Toán tử	Tên toán tử	Mô tả	Ví dụ
&	AND	Nếu cả 2 bit đều là 1 thì giá trị là 1	x & y
|	OR	Nếu 1 trong 2 bit là 1 thì giá trị là 1	x | y
^	XOR	Nếu chỉ có 1 trong 2 bit là 1 thì giá trị là 1	x ^ y
~	NOT	Nếu bit là 1 thì giá trị là 0, và ngược lại	x ~ y
<<	Dịch bit sang trái	Dịch các bit sang trái bằng cách đặt bit 0 vào bên phải của dãy bit và loại bỏ bit ngoài cùng bên trái	x << y
>>	Dịch bit sang phải	Dịch các bit sang phải bằng cách đặt bản sao của bit ngoài cùng bên trái vào bên trái của dãy bit và loại bỏ bit ngoài cùng bên phải	x >> y

Cách chuyển đổi giá trị thập phân sang nhị phân theo bảng sau:

Index	7	6	5	4	3	2	1	0
Tổng bit ở thập phân	128	64	32	16	8	4	2	1
Giá trị bit								

8. Độ ưu tiên của các toán tử

Dùng để mô tả thứ tự thực hiện các phép toán

Ví dụ 1: Dấu ngoặc đơn () có thứ tự ưu tiên cao nhất, điều này có nghĩa là biểu thức trong dấu ngoặc () phải được đánh giá đầu tiên:

In [ ]:
```
print((6 + 3) - (6 + 3))
```

```
0
```

Ví dụ 2: Phép nhân * có độ ưu tiên cao hơn phép cộng +, do đó phép nhân sẽ được đánh giá trước phép cộng:

In [ ]:
```
print(100 + 5*3)
```

```
115
```

Thứ tự ưu tiên được mô tả trong bảng sau, bắt đầu với độ ưu tiên cao nhất (từ trên xuống):

Toán tử	Mô tả	Ví dụ	Kết quả
()	Dấu ngoặc đơn	(6 + 3) - 2	7
**	Toán tử mũ	3 ** 3 - 2	25
+x -x ~x	Toán tử cộng 1 ngôi, trừ 1 ngôi, toán tử NOT	100 + ~3	96
* / // %	Nhân, chia, chia lấy nguyên, chia lấy dư	100 + 5 * 3	115
+ -	Cộng, trừ	100 - 5 * 3	85
<< >>	Dịch bit trái, dịch bit phải	8 >> 4 - 2	2
&	Bitwise AND	6 & 2 + 1	2
^	Bitwise XOR	6 ^ 2 + 1	5
|	Bitwise OR	6 | 2 + 1	7
== != > >= < <= is is not in not in	Toán tử so sánh, xác định, quan hệ	5 == 4 + 1	True
not	Logical NOT	not 5 == 5	False

and	Logical AND	1 or 2 and 3	1
or	Logical OR	4 or 5 + 10 or 8	4

Nếu hai toán tử có cùng độ ưu tiên thì biểu thức được đánh giá từ trái sang phải

In [ ]:
```
print(5 + 4 - 7 + 3)
```

```
5
```