

Bài 6: Các toán tử trong Python

Mục lục:

1. Toán tử số học
2. Toán tử gán
3. Toán tử so sánh
4. Toán tử logic
5. Toán tử định danh
6. Toán tử thành viên
7. Toán tử thao tác trên bit

Nội dung:

1. Toán tử số học

- Dùng để thực hiện các phép toán.

Toán tử	Tên	Ví dụ minh họa
+	Phép cộng	$3 + 5 = 8$
-	Phép trừ	$5 - 8 = -3$
*	Phép nhân	$9 * 8 = 72$
/	Phép chia	$9 / 4 = 2$
%	Phép chia lấy dư	$9 \% 5 = 4$
**	Phép lũy thừa	$5 ** 2 = 25$
//	Phép chia làm tròn xuống	$5 // 2 = 2$

2. Toán tử gán

- Gán giá trị phía phải phép gán cho giá trị phía bên trái.

Toán tử	Ví dụ	Tương đương
=	$X = 100$	$X = 100$
+=	$X += 100$	$X = X + 100$
-=	$X -= 12$	$X = X - 12$
*=	$X *= 20$	$X = X * 20$
/=	$X /= 35$	$X = X / 35$
%=	$X \% = 6$	$X = X \% 6$
//=	$X //= 3$	$X = X // 3$
**=	$X ** = 3$	$X = X ** 3$

&=	X &= 2	X = X & 2
=	X = 2	X = X 2
^=	X ^= 2	X = X ^ 2
>>=	X >>= 5	X = X >> 5
<<=	X <<= 6	X = X << 6

3. Toán tử so sánh

- So sánh hai giá trị ở hai vế toán tử. Trả về kết quả true/false.

Toán tử	Tên	Ví dụ minh họa
==	So sánh bằng	X == Y
<	So sánh nhỏ hơn	X < Y
>	So sánh lớn hơn	X > Y
<=	So sánh nhỏ hơn hoặc bằng	X <= Y
>=	So sánh lớn hơn hoặc bằng	X >= Y
!=	So sánh không tương đương	X != Y

4. Toán tử logic

- Dùng để kết hợp các câu lệnh điều kiện.

Toán tử	Mô tả	Ví dụ
and	Trả về true nếu cả hai vế đều true	3 < 5 and 9 > 4
or	Trả về true nếu một trong hai vế là true	3 > 5 or 5 > 1
not	Trả về kết quả phủ định	not(3 > 5) : true

5. Toán tử định danh

- Kiểm tra hai đối tượng có cùng địa chỉ bộ nhớ hay không

Toán tử	Mô tả	Ví dụ
is	Trả về true nếu cả hai vế là cùng một đối tượng	X is Y
is not	Trả về true nếu hai vế là hai đối tượng khác nhau	X is not Y

6. Toán tử thành viên

Toán tử	Mô tả	Ví dụ
In	Trả về true nếu đối tượng là thành phần của danh sách	X = [1, 2, 3, 4] A = 1

		A in X: true
Not in	Trả về true nếu đối tượng không là thành phần của danh sách	X = [1, 2, 3, 4] A = 9 A not in X: true

7. Toán tử thao tác trên bit

Toán tử	Tên	Mô tả
&	AND	Kết quả là bit 1 nếu cả hai là bit 1
	OR	Kết quả là bit 1 nếu 1 trong hai bit là 1
^	XOR	Kết quả là bit 1 nếu chỉ 1 trong 2 bit là 1
~	NOT	Đảo ngược chuỗi bit
<<	Dịch chuỗi bit sang trái	Dịch chuỗi bit sang trái và thêm bit 0 vào bit phải cùng
>>	Dịch chuỗi bit sang phải	Dịch chuỗi bit sang phải và cắt bỏ phần bit phải cùng bị dịch.