

# Bài 3: Operators trong Python

Tuần 26 - Thứ 3 - Buổi sáng (09:00-12:00)

Mục tiêu học tập:

- Hiểu và sử dụng các toán tử số học trong Python
- Nắm vững các toán tử so sánh
- Thành thạo các toán tử logic
- Thực hành kết hợp các loại toán tử

## 1. Toán tử số học (Arithmetic Operators)

Python cung cấp các toán tử số học cơ bản để thực hiện các phép tính toán học.

### 1.1 Các toán tử cơ bản

Toán tử	Ý nghĩa	Ví dụ	Kết quả
+	Cộng	5 + 3	8
-	Trừ	5 - 3	2
*	Nhân	5 * 3	15
/	Chia	5 / 3	1.6666666666666667
//	Chia lấy phần nguyên	5 // 3	1
%	Chia lấy phần dư	5 % 3	2
**	Lũy thừa	5 ** 3	125

```
In [ ]: # Ví dụ về các toán tử số học
a = 10
b = 3

print(f"a = {a}, b = {b}")
print(f"a + b = {a + b}")
print(f"a - b = {a - b}")
print(f"a * b = {a * b}")
print(f"a / b = {a / b}")
print(f"a // b = {a // b}")
print(f"a % b = {a % b}")
print(f"a ** b = {a ** b}")
```

### 1.2 Thứ tự ưu tiên của toán tử

Python tuân theo thứ tự ưu tiên toán học:

1. `()` - Ngoặc đơn (cao nhất)
2. `**` - Lũy thừa
3. `*`, `/`, `//`, `%` - Nhân, chia
4. `+`, `-` - Cộng, trừ (thấp nhất)

**Lưu ý:** Các toán tử cùng mức ưu tiên được tính từ trái sang phải.

```
In [ ]: # Ví dụ về thứ tự ưu tiên
result1 = 10 + 5 * 2
result2 = (10 + 5) * 2
result3 = 2 ** 3 ** 2
result4 = (2 ** 3) ** 2

print(f"10 + 5 * 2 = {result1}") # 20
print(f"(10 + 5) * 2 = {result2}") # 30
print(f"2 ** 3 ** 2 = {result3}") # 512 (2**(3**2))
print(f"(2 ** 3) ** 2 = {result4}") # 64
```

## 2. Toán tử so sánh (Comparison Operators)

Toán tử so sánh được sử dụng để so sánh hai giá trị và trả về kết quả `True` hoặc `False`.

### 2.1 Các toán tử so sánh

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<code>==</code>	Bằng	<code>5 == 5</code>	<code>True</code>
<code>!=</code>	Khác	<code>5 != 3</code>	<code>True</code>
<code>&gt;</code>	Lớn hơn	<code>5 &gt; 3</code>	<code>True</code>
<code>&lt;</code>	Nhỏ hơn	<code>5 &lt; 3</code>	<code>False</code>
<code>&gt;=</code>	Lớn hơn hoặc bằng	<code>5 &gt;= 5</code>	<code>True</code>
<code>&lt;=</code>	Nhỏ hơn hoặc bằng	<code>3 &lt;= 5</code>	<code>True</code>

```
In [ ]: # Ví dụ về toán tử so sánh
x = 10
y = 5

print(f"x = {x}, y = {y}")
print(f"x == y: {x == y}")
print(f"x != y: {x != y}")
print(f"x > y: {x > y}")
print(f"x < y: {x < y}")
print(f"x >= y: {x >= y}")
print(f"x <= y: {x <= y}")
```

## 2.2 So sánh chuỗi (String)

Python có thể so sánh chuỗi dựa trên thứ tự từ điển (lexicographic order).

```
In [ ]: # So sánh chuỗi
name1 = "Alice"
name2 = "Bob"
name3 = "alice"

print(f'"{name1}" == "{name2}": {name1 == name2}')
print(f'"{name1}" < "{name2}": {name1 < name2}')
print(f'"{name1}" == "{name3}": {name1 == name3}')
print(f'"{name1}" < "{name3}": {name1 < name3}')
```

## 3. Toán tử logic (Logical Operators)

Toán tử logic được sử dụng để kết hợp các biểu thức điều kiện.

### 3.1 Các toán tử logic

Toán tử	Ý nghĩa	Ví dụ	Kết quả
and	Và (cả hai đều đúng)	True and False	False
or	Hoặc (một trong hai đúng)	True or False	True
not	Phủ định	not True	False

```
In [ ]: # Ví dụ về toán tử logic
a = True
b = False

print(f"a = {a}, b = {b}")
print(f"a and b: {a and b}")
print(f"a or b: {a or b}")
print(f"not a: {not a}")
print(f"not b: {not b}")
```

### 3.2 Bảng chân lý (Truth Table)

Toán tử AND

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

## Toán tử OR

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

## Toán tử NOT

A	not A
True	False
False	True

```
In [ ]: # Minh họa bảng chân lý
print("Bảng chân lý AND:")
print(f"True and True = {True and True}")
print(f"True and False = {True and False}")
print(f"False and True = {False and True}")
print(f"False and False = {False and False}")

print("\nBảng chân lý OR:")
print(f"True or True = {True or True}")
print(f"True or False = {True or False}")
print(f"False or True = {False or True}")
print(f"False or False = {False or False}")
```

## 4. Kết hợp các loại toán tử

### 4.1 Sử dụng toán tử trong biểu thức phức tạp

```
In [ ]: # Ví dụ kết hợp các toán tử
age = 25
salary = 50000
experience = 3

# Kiểm tra điều kiện phức tạp
is_eligible = (age >= 18 and age <= 65) and (salary > 30000 or experience >= 2)

print(f"Tuổi: {age}")
print(f"Lương: {salary}")
print(f"Kinh nghiệm: {experience} năm")
print(f"Đủ điều kiện: {is_eligible}")

# Phân tích từng phần
age_check = age >= 18 and age <= 65
```

```

salary_or_exp_check = salary > 30000 or experience >= 2

print(f"\nPhân tích:")
print(f"Tuổi hợp lệ (18-65): {age_check}")
print(f"Lương > 30000 HOẶC kinh nghiệm >= 2: {salary_or_exp_check}")
print(f"Kết quả cuối cùng: {age_check and salary_or_exp_check}")

```

## 4.2 Thứ tự ưu tiên toán tử tổng hợp

1. `()` - Ngoặc đơn
2. `**` - Lũy thừa
3. `*`, `/`, `//`, `%` - Nhân, chia
4. `+`, `-` - Cộng, trừ
5. `==`, `!=`, `>`, `<`, `>=`, `<=` - So sánh
6. `not` - Phủ định logic
7. `and` - Và logic
8. `or` - Hoặc logic

```

In [ ]: # Ví dụ về thứ tự ưu tiên
x = 5
y = 10
z = 15

# Biểu thức phức tạp
result = x + y * 2 > z and not y < x or z % 5 == 0

print(f"x = {x}, y = {y}, z = {z}")
print(f"Biểu thức: x + y * 2 > z and not y < x or z % 5 == 0")
print(f"Kết quả: {result}")

# Phân tích từng bước
print("\nPhân tích từng bước:")
step1 = y * 2 # 20
print(f"1. y * 2 = {step1}")

step2 = x + step1 # 25
print(f"2. x + (y * 2) = {step2}")

step3 = step2 > z # 25 > 15 = True
print(f"3. (x + y * 2) > z = {step3}")

step4 = y < x # 10 < 5 = False
print(f"4. y < x = {step4}")

step5 = not step4 # not False = True
print(f"5. not (y < x) = {step5}")

step6 = step3 and step5 # True and True = True
print(f"6. ((x + y * 2) > z) and (not (y < x)) = {step6}")

step7 = z % 5 # 15 % 5 = 0
print(f"7. z % 5 = {step7}")

```

```
step8 = step7 == 0 # 0 == 0 = True
print(f"8. (z % 5) == 0 = {step8}")

final_result = step6 or step8 # True or True = True
print(f"9. Kết quả cuối cùng: {final_result}")
```

## 5. Bài tập thực hành

### Bài tập 1: Tính toán cơ bản

Cho hai số  $a = 15$ ,  $b = 4$ . Tính và in ra:

- Tổng, hiệu, tích, thương
- Chia lấy phần nguyên và phần dư
- $a$  mũ  $b$

```
In [ ]: # Bài tập 1: Điền code vào đây
a = 15
b = 4

# Viết code tính toán ở đây
```

### Bài tập 2: So sánh và logic

Viết chương trình kiểm tra một số có phải là số chẵn và lớn hơn 10 không.

```
In [ ]: # Bài tập 2: Điền code vào đây
number = 24

# Viết code kiểm tra ở đây
```

### Bài tập 3: Biểu thức phức tạp

Cho điểm toán = 8, điểm văn = 7, điểm anh = 9. Kiểm tra học sinh có đạt học lực khá không? (Điều kiện: điểm trung bình  $\geq 7$  và không có môn nào dưới 5)

```
In [ ]: # Bài tập 3: Điền code vào đây
diem_toan = 8
diem_van = 7
diem_anh = 9

# Viết code kiểm tra ở đây
```

## 6. Tóm tắt

Kiến thức đã học:

1. **Toán tử số học:** `+`, `-`, `*`, `/`, `//`, `%`, `**`
2. **Toán tử so sánh:** `==`, `!=`, `>`, `<`, `>=`, `<=`
3. **Toán tử logic:** `and`, `or`, `not`
4. **Thứ tự ưu tiên** của các toán tử
5. **Kết hợp các toán tử** trong biểu thức phức tạp

## Lưu ý quan trọng:

- Sử dụng ngoặc đơn để làm rõ thứ tự tính toán
- Toán tử so sánh luôn trả về `True` hoặc `False`
- Toán tử logic hoạt động với giá trị boolean
- Chú ý sự khác biệt giữa `=` (gán) và `==` (so sánh)

## Chuẩn bị cho buổi sau:

- Ôn tập các toán tử đã học
- Thực hành thêm các bài tập về operators
- Sẵn sàng cho bài học về **If/else statements**