

# TÁC GIẢ: 8 SYNC

---

## CỘNG ĐỒNG



[Kevin Nguyễn](#)



[Nhóm Chia Sẻ Công Nghệ](#)



[Nhóm BlockChain](#)



[Tiktok: 8 Sync](#)



[Youtube: 8 Sync Dev](#)



[Zalo](#)

## KHÓA HỌC:



[Fullstack Python](#)



[Fullstack Nextjs](#)



[Fullstack Android-IOs](#)

Tài liệu sẽ được cập nhật định kì và thông báo trong group nên các bạn chú ý nhen .



## PYTHON - TOÁN TỬ SO SÁNH

---

Toán tử so sánh trong Python rất quan trọng trong các câu lệnh điều kiện của Python (if, else và elif) và các câu lệnh lặp (while và for loops). Các toán tử so sánh cũng được gọi là các toán tử quan hệ. Một số toán tử được biết đến là "<" đại diện cho toán tử nhỏ hơn, và ">" đại diện cho toán tử lớn hơn.

Python sử dụng thêm hai toán tử nữa, kết hợp "=" với hai toán tử này. Toán tử "<=" đại diện cho toán tử nhỏ hơn hoặc bằng và toán tử ">=" đại diện cho toán tử lớn hơn hoặc bằng.

### Các Toán tử So sánh Khác nhau trong Python

Python có thêm hai toán tử so sánh dưới dạng "==" và "!=". Chúng được sử dụng cho các toán tử bằng và khác. Do đó, có sáu toán tử so sánh trong Python và chúng được liệt kê dưới đây trong bảng này:

- < Ít hơn a<b
- > Lớn hơn a>b
- <= Nhỏ hơn hoặc bằng a<=b
- >= Lớn hơn hoặc bằng a>=b
- == Bằng a==b
- != Không bằng a!=b

Các toán tử so sánh là nhị phân theo tính chất, yêu cầu hai toán hạng. Một biểu thức liên quan đến một toán tử so sánh được gọi là biểu thức Boolean, và luôn trả về True hoặc False.

### Ví dụ



< 8 Sync Dev />

```
a = 5
b = 7
print(a > b)
print(a < b)
```

Kết quả sẽ là:



< 8 Sync Dev />

```
False
True
```



Cả hai toán hạng có thể là các biến, biểu thức hoặc các biểu thức Python. Vì Python hỗ trợ toán hạng kết hợp, bạn có thể có bất kỳ loại số học nào.

## Ví dụ

```
< 8 Sync Dev />

# Cả hai toán hạng đều là số nguyên
a = 5
b = 7
print("Cả hai toán hạng đều là số nguyên")
print("a =", a, "b =", b, "a > b là", a > b)
print("a =", a, "b =", b, "a < b là", a < b)
```

## Python - Các Toán tử So sánh

### Toán tử So sánh trong Python

Toán tử so sánh trong Python rất quan trọng trong các câu lệnh điều kiện của Python (if, else và elif) và các câu lệnh lặp (vòng lặp while và for). Các toán tử so sánh còn được gọi là các toán tử quan hệ. Một số toán tử nổi tiếng là "<" đại diện cho toán tử nhỏ hơn, và ">" đại diện cho toán tử lớn hơn.

Python sử dụng thêm hai toán tử nữa, kết hợp với dấu "=" với hai toán tử này. Dấu "<=" đại diện cho toán tử nhỏ hơn hoặc bằng và dấu ">=" đại diện cho toán tử lớn hơn hoặc bằng.

### Các Toán tử So sánh Khác nhau trong Python

Python có thêm hai toán tử so sánh dưới dạng "==" và "!=". Chúng được sử dụng cho các toán tử bằng và không bằng. Do đó, có sáu toán tử so sánh trong Python và chúng được liệt kê dưới đây trong bảng:

- < So sánh nhỏ hơn  $a < b$

- < 8 Sync Dev />

So sánh lớn hơn  $a > b$

- <= Nhỏ hơn hoặc bằng  $a \leq b$

- = Lớn hơn hoặc bằng  $a \geq b$

- == Bằng a == b
- != Không bằng a != b

Các toán tử so sánh là nhị phân, yêu cầu hai toán hạng. Một biểu thức liên quan đến một toán tử so sánh được gọi là một biểu thức Boolean và luôn trả về True hoặc False.

### Ví dụ

```
< 8 Sync Dev />

a = 5
b = 7
print(a > b)
print(a < b)
```

Kết quả sẽ là:

```
< 8 Sync Dev />

False
True
```

Cả hai toán hạng có thể là các hằng số Python, biến hoặc biểu thức. Vì Python hỗ trợ phép tính kết hợp, bạn có thể có bất kỳ kiểu số nào.

### Ví dụ

Dưới đây là mã minh họa về việc sử dụng các toán tử so sánh của Python với các số nguyên:

```
< 8 Sync Dev />

print("Cả hai toán hạng đều là số nguyên")
a = 5
b = 7
print("a=", a, "b=", b, "a > b là", a > b)
```

```
print("a=", a, "b=", b, "a < b là", a < b)
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:



< 8 Sync Dev />

Cả hai toán hạng đều là số nguyên

a= 5 b= 7 a > b là False

a= 5 b= 7 a < b là True

a= 5 b= 7 a == b là False

a= 5 b= 7 a != b là True

## So sánh của Số thực

Trong ví dụ dưới đây, một toán hạng số nguyên và một toán hạng số thực được so sánh.

Ví dụ



< 8 Sync Dev />

```
print("So sánh của số nguyên và số thực")
a = 10
b = 10.0
print("a=", a, "b=", b, "a > b là", a > b)
print("a=", a, "b=", b, "a < b là", a < b)
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:



< 8 Sync Dev />



So sánh của số nguyên và số thực

`a= 10 b= 10.0 a > b` là False

`a= 10 b= 10.0 a < b` là False

`a= 10 b= 10.0 a == b` là True

`a= 10 b= 10.0 a != b` là False

## So sánh của Số phức

Mặc dù đối tượng phức là một loại dữ liệu số trong Python, hành vi của nó khác biệt so với các loại dữ liệu khác. Python không hỗ trợ các toán tử `<` và `>`, tuy nhiên, nó hỗ trợ các toán tử bằng (`==`) và toán tử không bằng (`!=`).

Ví dụ



< 8 Sync Dev />

```
print("So sánh của số phức")
a = 10 + 1j
b = 10 - 1j
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:



< 8 Sync Dev />

```
So sánh của số phức
a= (10+1j) b= (10-1j) a == b là False
a= (10+1j) b= (10-1j) a != b là True
```

Bạn sẽ nhận được một `TypeError` khi sử dụng các toán tử nhỏ hơn hoặc lớn hơn.

Ví dụ



< 8 Sync Dev />

```
print("So sánh của số phức")
a = 10 + 1j
b = 10 - 1j
print("a=", a, "b=", b, "a < b là", a < b)
print("a=", a, "b=", b, "a > b là", a > b)
```

Kết quả sẽ là:



< 8 Sync Dev />

So sánh của số phức

Traceback (most recent call last):

File "C:\Users\mlath\examples\example.py", line 5, in <module>

print("a=", a, "b=", b, "a < b là", a < b)

TypeError: '<' not supported between instances of 'complex' and 'complex'

## So sánh của Kiểu Boolean

Các đối tượng Boolean trong Python thực sự là số nguyên: True là 1 và False là 0. Trong thực tế, Python coi bất kỳ số khác không nào là True. Trong Python, việc so sánh các đối tượng Boolean là có thể. "False < True" là True!

Ví dụ



< 8 Sync Dev />

```
print("So sánh của Kiểu Boolean")
a = True
b = False
print("a=", a, "b=", b, "a < b là", a < b)
print("a=", a, "b=", b, "a > b là", a > b)
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:



< 8 Sync Dev />

So sánh của Kiểu Boolean

```
a= True b= False a < b là False
```

```
a= True b= False a > b là True
```

```
a= True b= False a == b là False
```

```
a= True b= False a != b là True
```

## So sánh của Các loại Dãy

Trong Python, chỉ có thể thực hiện so sánh của các đối tượng dãy tương tự. Một đối tượng chuỗi có thể so sánh với một chuỗi khác chỉ. Một danh sách không thể được so sánh với một tuple, ngay cả khi cả hai đều có các phần tử giống nhau.

Ví dụ



< 8 Sync Dev />

```
print("So sánh của các loại dãy khác nhau")
```

```
a = (1, 2, 3)
```

```
b = [1, 2, 3]
```

```
print("a=", a, "b=", b, "a < b là", a < b)
```

Kết quả sẽ là:



< 8 Sync Dev />

So sánh của các loại dãy khác nhau

Traceback (most recent call last):

```
File "C:\Users\mlath\examples\example.py", line 5, in <module>
```

```
    print("a=", a, "b=", b, "a < b là", a < b)
```

```
TypeError: '<' not supported between instances of 'tuple' and 'list'
```



Các đối tượng dãy được so sánh bằng cơ chế sắp xếp từ điển. So sánh bắt đầu từ mục tại chỉ mục 0. Nếu chúng bằng nhau, so sánh di chuyển đến chỉ mục tiếp theo cho đến khi các mục tại một chỉ mục nhất định không bằng nhau hoặc một trong hai dãy đã hết. Nếu một dãy là một phần dãy con ban đầu của dãy kia, dãy ngắn hơn sẽ là dãy nhỏ hơn.

Sự khác biệt giữa hai toán hạng phụ thuộc vào sự khác biệt giữa các giá trị của các mục tại chỉ mục mà chúng không bằng nhau. Ví dụ, 'BAT' > 'BAR' là True, vì T đến sau R trong thứ tự Unicode.

Nếu tất cả các mục của hai dãy so sánh bằng nhau, các dãy được coi là bằng nhau.

### Ví dụ

```
< 8 Sync Dev />

print("So sánh của chuỗi")
a = 'BAT'
b = 'BALL'
print("a=", a, "b=", b, "a < b là", a < b)
print("a=", a, "b=", b, "a > b là", a > b)
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:

```
< 8 Sync Dev />

So sánh của chuỗi
a= BAT b= BALL a < b là False
a= BAT b= BALL a > b là True
a= BAT b= BALL a == b là False
a= BAT b= BALL a != b là True
```

Trong ví dụ dưới đây, hai đối tượng tuple được so sánh:

### Ví dụ



< 8 Sync Dev />

```
print("So sánh của các tuples")
a = (1, 2, 4)
b = (1, 2, 3)
print("a=", a, "b=", b, "a < b là", a < b)
print("a=", a, "b=", b, "a > b là", a > b)
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:



< 8 Sync Dev />

```
a= (1, 2, 4) b= (1, 2, 3) a < b là False
a= (1, 2, 4) b= (1, 2, 3) a > b là True
a= (1, 2, 4) b= (1, 2, 3) a==b là False
a= (1, 2, 4) b= (1, 2, 3) a != b là True
```

## So sánh của Các Đối tượng Từ điển

Việc sử dụng các toán tử "<" và ">" cho từ điển của Python không được định nghĩa. Trong trường hợp của các toán hạng này, TypeError: '<' not supported between instances of 'dict' and 'dict' được báo cáo.

So sánh bằng kiểm tra xem độ dài của cả hai mục từ điển có giống nhau không. Độ dài của từ điển là số cặp khóa-giá trị trong đó.

Các từ điển Python đơn giản chỉ được so sánh bằng độ dài. Từ điển có ít phần tử hơn được coi là nhỏ hơn một từ điển có nhiều phần tử.

### Ví dụ



< 8 Sync Dev />

```
print("So sánh của các đối tượng từ điển")
a = {1: 1, 2: 2}
```

```
b = {2: 2, 1: 1, 3: 3}
print("a=", a, "b=", b, "a == b là", a == b)
print("a=", a, "b=", b, "a != b là", a != b)
```

Kết quả sẽ là:



< 8 Sync Dev />

So sánh của các đối tượng từ điển

a= {1: 1, 2: 2} b= {2: 2, 1: 1, 3: 3} a==b là False

a= {1: 1, 2: 2} b= {2: 2, 1: 1, 3: 3} a!=b là True

Đó là một cái nhìn tổng quan về các toán tử so sánh trong Python cùng với các ví dụ minh họa. Các toán tử so sánh này là cực kỳ hữu ích trong việc so sánh các giá trị và điều khiển luồng của chương trình Python.