

# 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 .



## BÀI 23. PYTHON - ƯU TIÊN TOÁN TỬ

---

Trong Python, một biểu thức bao gồm một hoặc nhiều biến, hằng số và toán tử (toán tử số học, logic, bitwise, v.v.). Trình thông dịch Python đánh giá biểu thức và kết quả được gán cho một biến hoặc được sử dụng trong một câu lệnh khác. Trình

thông dịch thực hiện các hoạt động khác nhau tùy thuộc vào ưu tiên của các toán tử.

## ƯU TIÊN TOÁN TỬ PYTHON

Một biểu thức có thể chứa nhiều toán tử được đánh giá. Ưu tiên toán tử xác định thứ tự mà các toán tử được đánh giá. Nói cách khác, thứ tự đánh giá toán tử được xác định bởi ưu tiên toán tử.

Nếu một biểu thức cụ thể chứa nhiều toán tử, thứ tự đánh giá của chúng được xác định bởi thứ tự ưu tiên. Ví dụ, xem xét biểu thức sau:



< 8 Sync Dev />

```
>>> a = 2+3*5
```

Ở đây, giá trị của  $a$  sẽ là gì? - có thể là 17 (nhân 3 với 5 trước và sau đó cộng 2) hoặc 25 (cộng 2 và 3 và sau đó nhân với 5)? Nguyên tắc ưu tiên toán tử của Python xuất hiện ở đây.

Nếu chúng ta chỉ xem xét các toán tử số học trong Python, nguyên tắc BODMAS truyền thống cũng được sử dụng bởi trình thông dịch Python, trong đó dấu ngoặc được đánh giá trước, các toán tử chia và nhân tiếp theo, tiếp theo là cộng và trừ. Do đó,  $a$  sẽ trở thành 17 trong biểu thức trên.

Ngoài ưu tiên toán tử, tính kết hợp của các toán tử cũng quan trọng. Nếu một biểu thức bao gồm các toán tử có cùng mức ưu tiên, tính kết hợp xác định thứ tự. Hầu hết các toán tử có tính kết hợp từ trái sang phải. Điều đó có nghĩa là toán tử ở bên trái được đánh giá trước toán tử ở bên phải.

Hãy xem xét một biểu thức khác:



< 8 Sync Dev />

```
>>> b = 10/5*4
```



Trong trường hợp này, cả \* (nhân) và / (chia) đều có cùng mức ưu tiên. Tuy nhiên, nguyên tắc tính kết hợp từ trái sang phải thực hiện phép chia trước ( $10/5 = 2$ ) và sau đó phép nhân ( $2*4 = 8$ ).

# BẢNG ƯU TIÊN TOÁN TỬ PYTHON

Bảng sau liệt kê tất cả các toán tử trong Python theo thứ tự giảm dần của ưu tiên. Các toán tử trong cùng một ô dưới cột Toán Tử có cùng ưu tiên.

Sr.No.	Toán Tử & Mô Tả
1	<code>() , [] , {}</code>
2	<code>[index] , [index:index]</code>
3	<code>await x</code>
4	<code>**</code>
5	<code>+x , -x , ~x</code>
6	<code>*</code> , <code>@</code> , <code>/</code> , <code>//</code> , <code>%</code>
7	<code>+</code> , <code>-</code>
8	<code>&lt;&lt;</code> , <code>&gt;&gt;</code>
9	<code>&amp;</code>
10	<code>^</code>
11	<code> </code>
12	<code>in</code> , <code>not in</code> , <code>is</code> , <code>is not</code> , <code>&lt;</code> , <code>&lt;=</code> , <code>&gt;</code> , <code>&gt;=</code> , <code>!=</code> , <code>==</code>
13	<code>not x</code>
14	<code>and</code>
15	<code>or</code>
16	<code>if – else</code>
17	<code>lambda</code>
18	<code>:=</code>

# VÍ DỤ VỀ ƯU TIÊN TOÁN TỬ PYTHON

## < 8 Sync Dev />

```
a = 20
b = 10
c = 15
d = 5
e = 0
```

```
e = (a + b) * c / d      #( 30 * 15 ) / 5
print ("Giá trị của (a + b) * c / d là ", e)
```

```
e = ((a + b) * c) / d    # (30 * 15) / 5
print ("Giá trị của ((a + b) * c) / d là ", e)
```

```
e = (a + b) * (c / d);   # (30) * (15/5)
print ("Giá trị của (a + b) * (c / d) là ", e)
```

```
e = a + (b * c) / d;     # 20 + (150/5)
print ("Giá trị của a + (b * c) / d là ", e)
```

Khi bạn thực thi chương trình trên, nó sẽ cho ra kết quả sau:

## < 8 Sync Dev />

```
Giá trị của (a + b) * c / d là  90.0
Giá trị của ((a + b) * c) / d là  90.0
Giá trị của (a + b) * (c / d) là  90.0
Giá trị của a + (b * c) / d là  50.0
```

Trong các ví dụ này:

- Biểu thức thứ nhất  $e = (a + b) * c / d$  tính toán phép nhân trước  $(a + b)$  và sau đó chia cho  $d$ .
- Biểu thức thứ hai  $e = ((a + b) * c) / d$  sử dụng ngoặc để chỉ định thứ tự đánh giá, cũng cho ra kết quả như biểu thức đầu tiên.

- Biểu thức thứ ba  $e = (a + b) * (c / d)$  sử dụng ngoặc để chỉ định thứ tự đánh giá, và kết quả cũng giống như hai biểu thức trước đó.
- Biểu thức cuối cùng  $e = a + (b * c) / d$  thực hiện phép nhân trước  $(b * c)$  và sau đó chia cho  $d$ .

Như bạn có thể thấy, việc hiểu và sử dụng đúng ưu tiên toán tử là rất quan trọng để đảm bảo tính chính xác của các biểu thức trong Python.