

# Precedência do Operador Python

Em Python, uma expressão contém uma ou **mais variáveis**, **literais** e **operadores** (**aritméticos**, **lógicos**, bit a bit, etc.). O **intérprete Python** avalia a expressão e o resultado é atribuído a uma variável ou usado em outra instrução. O intérprete realiza diversas operações de acordo com a precedência dos operadores.

## Precedência do Operador Python

Uma expressão pode ter vários operadores a serem avaliados. A precedência do operador define a ordem em que os operadores são avaliados. Em outras palavras, a ordem de avaliação do operador é determinada pela precedência do operador.

Se uma determinada expressão contém vários operadores, a sua ordem de avaliação é determinada pela ordem de precedência. Por exemplo, considere a seguinte expressão

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

Aqui, qual será o valor de **a**? - sim, será 17 (multiplicar 3 por 5 primeiro e depois somar 2) ou 25 (somar 2 e 3 e depois multiplicar por 5)? A regra de precedência do operador do Python entra em cena aqui.

Se considerarmos apenas os operadores aritméticos em Python, a regra **BODMAS** tradicional também é empregada pelo interpretador Python, onde os **colchetes** são avaliados primeiro, os operadores **de divisão** e **multiplicação** em seguida, seguidos pelos operadores **de adição** e **subtração**. Portanto, a se tornará 17 na expressão acima.

Além da precedência dos operadores, a associatividade dos operadores também é importante. Se uma expressão consiste em operadores com o mesmo nível de precedência, a associatividade determina a ordem. A maioria dos operadores tem associatividade da esquerda para a direita. Isso significa que o operador da esquerda é avaliado antes do da direita.

Consideremos outra expressão:

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

Neste caso, os operadores **\*** (multiplicação) e **/** (divisão) têm o mesmo nível de precedência. Porém, a regra de associatividade da esquerda para a direita realiza

primeiro a divisão ( $10/5 = 2$ ) e depois a multiplicação ( $2*4 = 8$ ).

## Tabela de precedência de operadores Python

A tabela a seguir lista todos os operadores em Python em ordem decrescente de precedência. Os operadores na mesma célula da coluna Operadores têm a mesma precedência.

Sr. Não.	Operador e Descrição
1	<b>() , [] , {}</b> Parênteses e colchetes
2	<b>[índice] , [índice:índice]</b> Assinatura, fatiamento,
3	<b>espere x</b> Aguarde expressão
4	<b>**</b> Exponenciação
5	<b>+x , -x , ~x</b> Positivo, negativo, bit a bit NÃO
6	<b>*, @, /, //, %</b> Multiplicação, multiplicação de matrizes, divisão, divisão de piso, resto
7	<b>+, -</b> Adição e subtração
8	<b>&lt;&lt;, &gt;&gt;</b> Mudanças à esquerda, mudanças à direita
9	<b>&amp;</b> E bit a bit
10	<b>^</b> XOR bit a bit
11	<b> </b> OU bit a bit
12	<b>dentro, não dentro, é, não é, &lt;, &lt;=, &gt;, &gt;=, !=, ==</b> Comparações, incluindo testes de adesão e testes de identidade

13	<b>não x</b> Booleano NÃO
14	<b>e</b> Booleano E
15	<b>ou</b> Booleano OU
16	<b>se - senão</b> Expressão condicional
17	<b>lambda</b> Expressão lambda
18	<b>:=</b> Operador de morsa

## Exemplo de precedência de operador Python

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

e = (a + b) * c / d      #( 30 * 15 ) / 5
print ("Value of (a + b) * c / d is ", e)

e = ((a + b) * c) / d    # (30 * 15 ) / 5
print ("Value of ((a + b) * c) / d is ", e)

e = (a + b) * (c / d);    # (30) * (15/5)
print ("Value of (a + b) * (c / d) is ", e)

e = a + (b * c) / d;      # 20 + (150/5)
print ("Value of a + (b * c) / d is ", e)
```

Quando você executa o programa acima, ele produz o seguinte resultado -

```
Value of (a + b) * c / d is 90.0
Value of ((a + b) * c) / d is 90.0
```

Value of  $(a + b) * (c / d)$  is 90.0

Value of  $a + (b * c) / d$  is 50.0