

Aula 06 - Operadores

Uma das partes mais importantes da programação, independente da linguagem, é aprender sobre os **operadores**. Como a Informática funciona como uma extensão da matemática, em programação, precisaremos realizar algumas operações matemáticas. Há diversos operadores para aprender, mas vamos começar do início. Nessa aula, iremos aprender os seguintes tipos de operadores: operadores aritméticos; operadores relacionais; operadores lógicos e/ou booleanos; operadores de incremento/decremento; e operadores de atribuição.

Operadores aritméticos

São usados nas operações matemáticas, como equações ou as quatro operações básicas da matemática, por exemplo.

- `+` : soma.
- `-` : subtração.
- `*` : multiplicação.
- `/` : divisão.
- `%` : módulo. Utilizado para retornar o resto de uma divisão.
- `**` : potênciação.

Exemplos:

Soma

```
In [ ]: print(2+2)
```

4

Subtração

```
In [ ]: print(3-2)
```

1

Multiplicação

```
In [ ]: print(4*2)
```

8

Divisão

```
In [ ]: print(6/3)
```

2.0

Obs: o resultado de uma divisão **SEMPRE** será um ponto flutuante.

Módulo

```
In [ ]: print(7%2)
```

1

Obs: o módulo de uma divisão **SEMPRE** será um número inteiro.

Potência

```
In [ ]: print(4**2)
```

16

Operadores relacionais

São usados para comparar valores.

- `>` : maior que.
- `<` : menor que.
- `>=` : maior ou igual a.
- `<=` : menor ou igual a.
- `==` : igualdade (não confundir com `=`).
- `!=` : diferente.

Exemplos

Maior que

```
In [ ]: 7 > 3
```

```
Out[ ]: True
```

Menor que

```
In [ ]: 4 < 10
```

```
Out[ ]: True
```

Maior ou igual a

```
In [ ]: 11 >= 11
```

```
Out[ ]: True
```

```
In [ ]: 10 >= 5
```

```
Out[ ]: True
```

Menor ou igual a

```
In [ ]: 2 <= 2
```

```
Out[ ]: True
```

```
In [ ]: 12 <= 20
```

```
Out[ ]: True
```

Igualdade

```
In [ ]: 10 == 10
```

```
Out[ ]: True
```

Diferente

```
In [ ]: 15 != 5
```

```
Out[ ]: True
```

Operadores lógicos ou booleanos

Geralmente são utilizados em conjunto com estruturas de decisão ou de repetição.

- `and` : operador E. Utilizado para verificar se duas afirmações são verdadeiras. Se uma for falsa, então a sentença inteira retorna falso.
- `or` : operador OU. Utilizado para verificar se uma ou outra afirmação é verdadeira. Basta que uma delas seja verdadeiro para retornar verdadeiro. As duas precisam ser falsas para retornar falso.
- `not` : operador NÃO. É um operador de negação. A afirmação precisa ser falsa para retornar verdadeiro.

Exemplos

and

```
In [ ]: 7 > 5 and 5 > 3
```

```
Out[ ]: True
```

```
In [ ]: 7 > 2 and 10 < 5
```

```
Out[ ]: False
```

or

```
In [ ]: 10 > 5 or 5 == 5
```

```
Out[ ]: True
```

```
In [ ]: 10 == 10 or 1 > 10
```

```
Out[ ]: True
```

```
In [ ]: 10 != 10 or 2 == 20
```

```
Out[ ]: False
```

not

```
In [ ]: not 10 > 5
```

```
Out[ ]: False
```

```
In [ ]: not 1 > 8
```

```
Out[ ]: True
```

Operadores de atribuição

São operadores empregados para realizar a atribuição de valores entre os operandos.

- `=` : atribui, recebe um valor.
- `+=` : soma a um valor e atribui o resultado.
- `-=` : subtrai a um valor e atribui o resultado.
- `*=` : multiplica a um valor e atribui o resultado.
- `/=` : divide a um valor e atribui o resultado.
- `%=` : obtém o resto de uma divisão e atribui o resultado.

Obs: diferentemente de outras linguagens de programação, o Python não suporta incrementos (`++`) ou decrementos (`--`).

Exemplos

Atribuição

```
In [ ]: valor = 1  
print(valor)
```

1

Atribuição de soma

```
In [ ]: valor = 1  
valor += 2  
print(valor)
```

3

Atribuição de subtração

```
In [ ]: valor = 10  
        valor -= 1  
        print(valor)
```

9

Atribuição de multiplicação

```
In [ ]: valor = 5  
        valor *= 2  
        print(valor)
```

10

Atribuição de divisão

```
In [ ]: valor = 6  
        valor /= 2  
        print(valor)
```

3.0

Atribuição de módulo

```
In [ ]: valor = 5  
        valor %= 3  
        print(valor)
```

2

Esses foram os principais operadores, ou pelo menos os mais utilizados. Aprenda-os, e você estará com o Python na palma da sua mão!