

OPERADORES EM PYTHON

IFBA – Campus Ilhéus
Curso: Back End com Python

Objetivo da Aula

- Apresentar os operadores básicos para manipulação de dados e tomada de decisão em Python.

Operadores Aritméticos

- São usados para realizar cálculos matemáticos entre valores numéricos.

Operador	Descrição	Exemplo em Python	Resultado
+	Adição	2 + 3	5
-	Subtração	5 - 2	3
*	Multiplicação	4 * 3	12
/	Divisão (float)	10 / 3	3.3333333333333335
//	Divisão inteira	10 // 3	3
%	Módulo (resto)	10 % 3	1
**	Exponenciação	2 ** 3	8

Operadores Aritméticos

- Uso diferente para os operadores **+** e *****

Operador	Descrição	Exemplo em Python	Resultado
+	Concatenação	'A' + 'B'	AB
*	Repetição	'A' * 4	AAAA

Operadores Aritméticos

- Exemplo prático

`a = 10`

`b = 3`

`print(a + b) # 13`

`print(a - b) # 7`

`print(a * b) # 30`

`print(a / b) # 3.333...`

`print(a // b) # 3`

`print(a % b) # 1`

`print(a ** b) # 1000 (10^3)`

Operadores Relacionais

- Usados para comparar valores. Retornam True ou False.

Operador	Descrição	Exemplo	Resultado
==	Igual a	5 == 5	True
!=	Diferente de	5 != 3	True
>	Maior que	5 > 3	True
<	Menor que	3 < 2	False
>=	Maior ou igual a	5 >= 5	True
<=	Menor ou igual a	4 <= 3	False

Operadores Relacionais

- Exemplo prático

idade = 18

```
print(idade == 18)  # True
```

```
print(idade != 21)  # True
```

```
print(idade > 16)   # True
```

```
print(idade < 18)   # False
```

```
print(idade >= 18)  # True
```

```
print(idade <= 17)  # False
```

Operadores Lógicos

- Combinam expressões lógicas. Retornam True ou False.

Operador	Descrição	Exemplo	Resultado
and	E (conjunção)	(5 > 3) and (2 < 4)	True
or	OU (disjunção)	(5 > 3) or (2 > 4)	True
not	Negação lógica	not (5 > 3)	False

Operadores Lógicos

- Exemplo prático

```
a = True
```

```
b = False
```

```
print(a and b) # False
```

```
print(a or b)  # True
```

```
print(not a)   # False
```

```
# Com condições
```

```
idade = 20
```

```
tem_carteira = True
```

```
pode_dirigir = idade >= 18 and tem_carteira
```

```
print(pode_dirigir) # True
```

Valores falsy

- São aqueles que, quando avaliados em um contexto booleano (como em uma condição *if*), são considerados como False.
- Ou seja, mesmo que esses valores não sejam o booleano False, eles se comportam como se fossem.

Valores falsy

- ⦿ False
- ⦿ Zero de qualquer tipo numérico (0, 0.0)
- ⦿ sequências e coleções vazias
 - " (string vazia)
 - [] (lista vazia)
 - {} (dicionário vazio)
 - () (tupla vazia)
 - set() (conjunto vazio)
 - range(0) (intervalo vazio)

Precedência e Associação dos Operadores

- Em expressões mais complexas, Python avalia os operadores seguindo uma ordem de precedência.
- Ordem de precedência:
 - Parênteses: `()`
 - Expoente: `**`
 - Sinais unários: `+x`, `-x`, `~x`
 - Multiplicação, Divisão, Módulo, Divisão inteira: `*`, `/`, `//`, `%`
 - Adição e Subtração: `+`, `-`
 - Operadores relacionais: `<`, `<=`, `>`, `>=`, `==`, `!=`
 - Operadores lógicos: `not`, `and`, `or`

Precedência e Associação dos Operadores

- Exemplo prático

Sem parênteses

resultado1 = 2 + 3 * 4

Com parênteses

resultado2 = (2 + 3) * 4

print(resultado1) # 14

print(resultado2) # 20

Observação: É boa prática usar parênteses para tornar a leitura mais clara, mesmo quando conhece a precedência.

Avaliação curto circuito

- O Python para de avaliar uma expressão lógic assim que o resultado final já pode ser determinado.

- Exemplo:

```
x=0
```

```
resultado = x and print("Isso não será  
impresso!")
```

```
print(resultado)
```

Avaliação curto circuito

⦿ Vantagens:

- Melhorar performance
- Escrita concisa de códigos, pois permite usar valores padrão de forma elegante, como:

```
nome = input("Digite seu nome") or "Anônimo"  
print(nome)
```

Resumo

Operadores	Exemplos
Aritméticos	+, -, *, /, //, %, **
Relacionais	==, !=, >, <, >=, <=
Lógicos	and, or, not
Precedência	Use parênteses para clareza

Exercício 1

- ⦿ Crie um programa que leia dois números inteiros e exiba:
 - A soma dos dois números.
 - A diferença entre eles.
 - O produto entre eles.
 - O quociente da divisão inteira.
 - O resto da divisão.
 - O resultado do primeiro número elevado ao segundo.

Exercício 2

- ⦿ Peça ao usuário para digitar sua idade. Verifique e imprima se a pessoa:
 - É maior de idade ($\text{idade} > 18$).
 - É menor de idade ($\text{idade} < 18$).
 - Tem exatamente 18 anos.

Exercício 3

- ⦿ Resolva as seguintes expressões, calcule e mostre o resultado:
- ⦿ $2 + 3 * 4$
- ⦿ $(2 + 3) * 4$
- ⦿ $2 ** 3 * 2$
- ⦿ $2 ** (3 * 2)$
- ⦿ Explique cada resultado, destacando o uso de parênteses.