



# Aula 4 - Operadores

Rodrigo Maciel

# O que são operadores ?

Os operadores podem realizar operações diferentes, dependendo do tipo de dado usado. Os operadores em Python são:

- Aritméticos (usados em contas);
- Relacionais (usados em comparações numéricas);
- Lógicos (usados em comparações lógicas);
- De atribuição (armazenamento de valores em variáveis).

# Operadores de atribuição

Os operadores de atribuição realizam a ação de alocar valores para variáveis.

- Variável do lado esquerdo, valor ou expressão do lado direito

```
>>> x = 0
```

- Pode-se atribuir valor a várias variáveis ao mesmo tempo

```
>>> x = y = 2
```

# Operadores de atribuição

- Pode-se também atribuir valores diferentes para variáveis diferentes ao mesmo tempo

```
>>> x, y = 3, 5
>>> x
3
>>> y
5
```

# Operadores aritméticos

Os operadores aritméticos são responsáveis por realizar as operações matemáticas básicas.

Operador	Conceito
+ (Adição ou sinal positivo)	- Realiza a soma entre operandos - Adiciona o sinal de positivo ao número
- (Subtração ou sinal negativo)	- Realiza a subtração entre operandos - Adiciona o sinal de negativo ao número
* (Multiplicação)	Realiza a multiplicação entre operandos
/ (Divisão)	Realiza a divisão entre operandos
// (Divisão inteira)	Realiza a divisão entre operandos e a parte decimal do resultado
% (Módulo)	Retorna o resto da divisão entre operandos
** (Exponenciação)	Retorna um número elevado a potência de outro

# Prioridade dos operadores aritméticos

Operador	Exemplo	Prioridade
(x)	$(1 + 2) * 3 \rightarrow 9$	1
**	$2 ** 3 \rightarrow 8$	2
+x	+15	3
-x	$-(5+3) \rightarrow -8$	3
*	$5 * 3 \rightarrow 15$	4
/	$5 / 3 \rightarrow 1.66$	4
//	$5 // 3 \rightarrow 1$	4
%	$5 \% 3 \rightarrow 2$	4
+	$5 + 3 \rightarrow 8$	5
-	$5 - 3 \rightarrow 2$	5

# Operadores aritméticos

```
numero_1 = 5
numero_2 = 2

soma = numero_1 + numero_2
subtracao = numero_1 - numero_2
multiplicacao = numero_1 * numero_2
divisao = numero_1 / numero_2
divisao_inteira = numero_1 // numero_2
modulo = numero_1 % numero_2
exponenciacao = numero_1 ** numero_2

print(soma) # 7
print(subtracao) # 3
print(multiplicacao) # 10
print(divisao) # 2.5
print(divisao_inteira) # 2
print(modulo) # 1
print(exponenciacao) # 25
```



# Operadores de atribuição

Operador	Exemplo	Equivalente a
=	$x = 1$	$x = 1$
+=	$x += 1$	$x = x + 1$
-=	$x -= 1$	$x = x - 1$
*=	$x *= 1$	$x = x * 1$
/=	$x /= 1$	$x = x / 1$
%=	$x \% = 1$	$x = x \% 1$



# Operadores Lógicos

Os operadores lógicos são muito utilizados em estruturas condicionais e estruturas de repetição. Esses operadores avaliam expressões lógicas e retornam verdadeiro ou falso.

Operador	Conceito
and	Retorna True se todas as condições forem verdadeiras, caso contrário retorna False
or	Retorna True se uma das condições for verdadeiras, caso contrário retorna False
not	Inverte o resultado: se o resultado da expressão for True, o operador retorna false

# Operadores Lógicos

```
>>> a = 1
>>> b = 1
>>> c = 3
>>> a == b
```

```
True
```

```
>>> b == c
```

```
False
```

```
>>> (a==b) and (b==c)
```

```
False
```

```
>>> (a==b) or (b==c)
```

```
True
```

```
>>> nome1 = "Maria"
```

```
>>> nome2 = "Joao"
```

```
>>> (nome1 == "Maria") and (nome1 != nome2)
```

```
True
```

a	b	not a	a and b	a or b
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False

# Operadores relacionais/comparação

Os operadores relacionais avaliam expressões e retornam verdadeiro ou falso, assim como os lógicos. No entanto, os relacionais são responsáveis por operações de comparação de magnitude.

Operador	Conceito
>(Maior que)	Verifica se um valor é maior que outro
<(Menor que)	Verifica se um valor é menor que outro
==(Igual a)	Verifica se um valor é igual a outro
!=(Diferente de)	Verifica se um valor é diferente de outro
>=(Maior ou igual a)	Verifica se um valor é maior ou igual a outro
<=(Menor ou igual a)	Verifica se um valor é menor ou igual a outro

# Operadores relacionais

```
>>> a = 1
```

```
>>> b = 1
```

```
>>> a == b
```

```
True
```

```
>>> a <= b
```

```
True
```

```
>>> a != b
```

```
False
```

```
>>> 2 >= -1
```

```
True
```

# Operador de Filiação/Associação

Operador	Conceito
in	Retorna True caso o valor seja encontrado na sequência
not in	Retorna True caso o valor não seja encontrado na sequência

```
frutas = ["banana", "laranja", "uva", "ameixa"]
```

```
fruta_1 = "ameixa"
```

```
fruta_2 = "melancia"
```

```
print(fruta_1 in frutas) # True
```

```
print(fruta_2 in frutas) # False
```

# Bibliotecas úteis - Math

Este módulo fornece acesso às funções matemáticas.

Método	Descrição	Exemplo
<code>math.ceil(x)</code>	Arredonda para cima	<code>math.ceil(5.3) → 6</code>
<code>math.copysign(x, y)</code>	Obtém um float com o valor absoluto de x, mas com o sinal de y	<code>math.copysign(-5.3, 1) → 5.3</code>
<code>math.fabs(x)</code>	Valor absoluto de x	<code>math.fabs(-5.3) → 5.3</code>
<code>math.floor(expr)</code>	Arredonda para baixo	<code>math.floor(5.3) → 5</code>
<code>math.fmod(x, y)</code>	Resto da divisão de x por y (usar quando x ou y forem float, caso contrário usar %)	<code>math.fmod(5.4, 2) → 1.4</code>
<code>math.trunc(x)</code>	Parte inteira de x	<code>math.trunc(5.6) → 5</code>

```
import math
```

# Bibliotecas úteis - Math

Este módulo fornece acesso às funções matemáticas.

Método	Descrição	Exemplo
<code>math.exp(x)</code>	$e^{**}x$	<code>math.exp(2)</code> → 7.38905609893065
<code>math.log(x)</code>	Logaritmo natural de x (base e)	<code>math.log(2)</code> → 0.6931471805599453
<code>math.log(x, y)</code>	Logaritmo de x na base y	<code>math.log(2, 10)</code> → 0.30102999566398114
<code>math.pow(x, y)</code>	$x^{**}y$	<code>math.pow(2, 3)</code> → 8.0
<code>math.sqrt(x)</code>	Raiz quadrada de x	<code>math.sqrt(16)</code> → 4.0

```
import math
```



# Bibliotecas úteis - Math

Este módulo fornece acesso às funções matemáticas.

Função	Descrição	Exemplo
<code>math.sin(x)</code>	Seno	<code>math.sin(0) → 0.0</code>
<code>math.asin(x)</code>	Arco seno	<code>math.asin(1) → 1.5707963267948966</code>
<code>math.cos(x)</code>	Cosseno	<code>math.cos(0) → 1.0</code>
<code>math.acos(x)</code>	Arco cosseno	<code>math.acos(-1) → 3.141592653589793</code>
<code>math.tan(x)</code>	Tangente	<code>math.tan(1) → 1.5574077246549023</code>
<code>math.atan(x)</code>	Arco tangente	<code>math.atan(1) → 0.7853981633974483</code>
<code>math.degrees(x)</code>	Converte radianos para graus	<code>math.degrees(math.pi) → 180.0</code>
<code>math.radians(x)</code>	Converte graus para radianos	<code>math.radians(180) → 3.141592653589793</code>

```
import math
```

# Bibliotecas úteis - Random

Este módulo implementa geradores de números pseudoaleatórios para várias distribuições.

- Número entre 0 e 1  
`random.random()`
- Número entre 5 e 6  
`5 + random.random()`
- Número entre 0 e 10  
`random.random() * 10`
- Número entre 50 e 70  
`50 + random.random() * 20`

```
import random
```

# Bibliotecas úteis - Random

Este módulo implementa geradores de números pseudoaleatórios para várias distribuições.

- Geração de número aleatórios inteiros

```
random.randint(intervaloInicial, intervaloFinal)
```

```
import random
```

# Praticando Python...

- Faça um programa que leia dois valores numéricos e realize a soma, subtração, multiplicação e divisão deles.
- Um triângulo equilátero é um triângulo que possui medidas iguais ( $AB = BC = AC$ ). Faça um programa onde o usuário fornece as medidas A, B e C e seu programa diga se é um triângulo equilátero (True ou False)