

# ▼ C# Fundamentals

## Operadores aritméticos

# Operadores aritméticos:

- Os operadores a seguir executam operações aritméticas com operandos de tipos numéricos:
- Operadores unários  $+$ ,  $--$ ,  $+$  e  $-$
- Operadores binário  $*$ ,  $/$ ,  $\%$ ,  $+$  e  $-$


# Operadores aritméticos:

## Operador de incremento pós-fixado

O resultado de `x++` é o valor de `x` *antes* da operação, como mostra o exemplo a seguir:

C#

 Copiar

 Executar


```
int i = 3;
Console.WriteLine(i); // output: 3
Console.WriteLine(i++); // output: 3
Console.WriteLine(i); // output: 4
```

## Operador de incremento de prefixo

O resultado de `++x` é o valor de `x` *após* a operação, como mostra o exemplo a seguir:

C#

 Copiar

 Executar

```
double a = 1.5;
Console.WriteLine(a); // output: 1.5
Console.WriteLine(++a); // output: 2.5
Console.WriteLine(a); // output: 2.5
```

# Operadores aritméticos:

## Operador de decremento pós-fixado

O resultado de `x--` é o valor de `x` *antes* da operação, como mostra o exemplo a seguir:

C#

Copiar

Executar

```
int i = 3;  
Console.WriteLine(i); // output: 3  
Console.WriteLine(i--); // output: 3  
Console.WriteLine(i); // output: 2
```

## Operador de decremento de prefixo

O resultado de `--x` é o valor de `x` *após* a operação, como mostra o exemplo a seguir:

C#

Copiar

Executar

```
double a = 1.5;  
Console.WriteLine(a); // output: 1.5  
Console.WriteLine(--a); // output: 0.5  
Console.WriteLine(a); // output: 0.5
```



# Operadores aritméticos:

## Operadores unários de adição e subtração

O operador unário `+` retorna o valor do operando. O operador unário `-` calcula a negação numérica do operando.

C#

 Copiar

 Executar

```
Console.WriteLine(+4);    // output: 4

Console.WriteLine(-4);    // output: -4
Console.WriteLine(-(-4)); // output: 4

uint a = 5;
var b = -a;
Console.WriteLine(b);      // output: -5
Console.WriteLine(b.GetType()); // output: System.Int64

Console.WriteLine(-double.NaN); // output: NaN
```

# Operadores aritméticos:

## Operador de adição +

O operador de adição `+` calcula a soma dos operandos:

C#

 Copiar

 Executar

```
Console.WriteLine(5 + 4);           // output: 9  
Console.WriteLine(5 + 4.3);        // output: 9.3  
Console.WriteLine(5.1m + 4.2m);    // output: 9.3
```

# Operadores aritméticos:

## Operador de subtração -

O operador de subtração `-` subtrai o operando à direita do operando à esquerda:

C#

 Copiar

 Executar

```
Console.WriteLine(47 - 3);      // output: 44  
Console.WriteLine(5 - 4.3);    // output: 0.7  
Console.WriteLine(7.5m - 2.3m); // output: 5.2
```

# Operadores aritméticos:

## Operador de multiplicação \*

O operador de multiplicação \* calcula o produto dos operandos:

C#

 Copiar

 Executar

```
Console.WriteLine(5 * 2);           // output: 10  
Console.WriteLine(0.5 * 2.5);       // output: 1.25  
Console.WriteLine(0.1m * 23.4m);    // output: 2.34
```



# Operadores aritméticos:

## Operador de divisão /

O operador de divisão `/` divide o operando à esquerda pelo operando à direita.

### Divisão de inteiros

Para os operandos de tipos inteiros, o resultado do operador `/` é de um tipo inteiro e igual ao quociente dos dois operandos arredondados para zero:

C#

Copiar

Executar

```
Console.WriteLine(13 / 5);    // output: 2
Console.WriteLine(-13 / 5);   // output: -2
Console.WriteLine(13 / -5);   // output: -2
Console.WriteLine(-13 / -5);  // output: 2
```

Para obter o quociente dos dois operandos como um número de ponto flutuante, use o tipo `float`, `double` ou `decimal`:

C#

Copiar

Executar

```
Console.WriteLine(13 / 5.0);    // output: 2.6

int a = 13;
int b = 5;
Console.WriteLine((double)a / b); // output: 2.6
```

# Operadores aritméticos:

## Divisão de ponto flutuante

Para os tipos `float`, `double` e `decimal`, o resultado do operador `/` é o quociente dos dois operandos:

C#

 Copiar

 Executar

```
Console.WriteLine(16.8f / 4.1f);    // output: 4.097561
Console.WriteLine(16.8d / 4.1d);    // output: 4.09756097560976
Console.WriteLine(16.8m / 4.1m);    // output: 4.0975609756097560975609756098
```

# Operadores aritméticos:


## Operador de resto %


O operador de resto % calcula o resto após dividir o operando à esquerda pelo à direita.

## Resto inteiro

Para os operandos de tipos inteiros, o resultado de `a % b` é o valor produzido por `a - (a / b) * b`. O sinal do resto diferente de zero é o mesmo que o do operando à esquerda, conforme mostra o seguinte exemplo:

C#

 Copiar

 Executar

```
Console.WriteLine(5 % 4);    // output: 1  
Console.WriteLine(5 % -4);   // output: 1  
Console.WriteLine(-5 % 4);   // output: -1  
Console.WriteLine(-5 % -4);  // output: -1
```

# Operadores aritméticos:

## Atribuição composta

C#

 Copiar Executar

```
int a = 5;  
a += 9;  
Console.WriteLine(a); // output: 14  
  
a -= 4;  
Console.WriteLine(a); // output: 10  
  
a *= 2;  
Console.WriteLine(a); // output: 20  
  
a /= 4;  
Console.WriteLine(a); // output: 5  
  
a %= 3;  
Console.WriteLine(a); // output: 2
```

# Operadores aritméticos:

## Precedência e associatividade do operador


A seguinte lista ordena os operadores aritméticos da precedência mais alta para a mais baixa:


- Incluir um pós-fixos a operadores de incremento `x++` e decremento `x--`
- Incluir um prefixos a operadores de incremento `++x` e de decremento `--x` e operadores unários `+` e `-`
- Operadores de multiplicação `*`, `/` e `%`
- Operadores de adição `+` e `-`

Operadores aritméticos binários são associativos à esquerda. Ou seja, os operadores com o mesmo nível de precedência são avaliados da esquerda para a direita.

Use parênteses, `()`, para alterar a ordem de avaliação imposta pela precedência e pela capacidade de associação do operador.

C#

 Copiar

 Executar

```
Console.WriteLine(2 + 2 * 2);    // output: 6
Console.WriteLine((2 + 2) * 2); // output: 8

Console.WriteLine(9 / 5 / 2);    // output: 0
Console.WriteLine(9 / (5 / 2)); // output: 4
```



# Operadores aritméticos:

- Links de documentação oficial:
- **Documentação:**
- <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/arithmetic-operators>





# Dúvidas?