

# Operadores



Prof. Wanderson Timóteo  
[www.wandersontimoteo.com.br](http://www.wandersontimoteo.com.br)



# O que vamos aprender

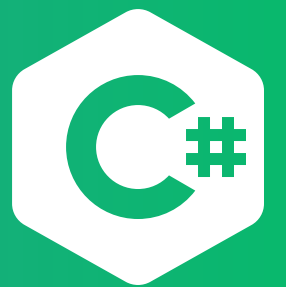
- Operador de Atribuição;
- Operadores Aritméticos;
- Operadores de comparação;
- Operadores Lógicos;





# O que vamos aprender

- **Operador de Atribuição;**
- ~~Operadores Aritméticos;~~
- ~~Operadores de comparação;~~
- ~~Operadores Lógicos;~~





# Operador de Atribuição

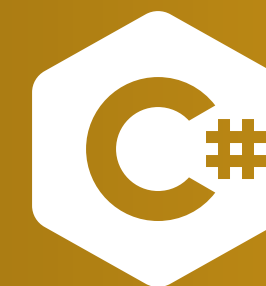
O operador de atribuição “=” atribui o valor do operando do lado direito a uma variável, uma **propriedade** ou um elemento **indexador** fornecido pelo operando do lado esquerdo. O resultado de uma expressão de atribuição é o valor atribuído a um operando do lado esquerdo. O tipo do operandos do lado direito deve ser do mesmo tipo ou implicitamente conversível para o operando do lado esquerdo.

```
operandoEsquerdo = operandoDireito;
```

Exemplo:

```
int x = 5;
```

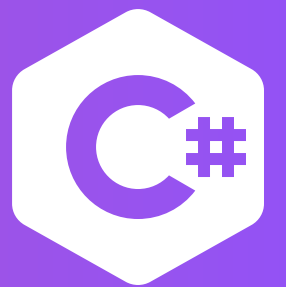
<https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/assignment-operator>





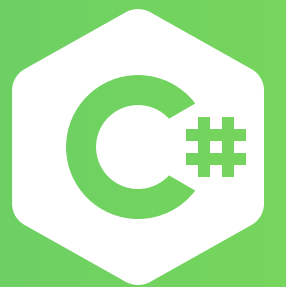
# O que vamos aprender

- ~~Operador de Atribuição;~~
- **Operadores Aritméticos;**
- ~~Operadores de comparação;~~
- ~~Operadores Lógicos;~~



# Operadores Aritméticos

Os operadores aritméticos são símbolos ou palavras-chave que permitem realizar cálculos matemáticos em valores numéricos. Eles são essenciais para executar operações como adição, subtração, multiplicação, divisão e outras.





# Operadores Aritméticos

Os principais operadores aritméticos são:

- **Adição (+):** Realiza a soma de dois valores.
  - Exemplo:  $3 + 4$  resulta em 7.
- **Subtração (-):** Realiza a diferença entre dois valores.
  - Exemplo:  $5 - 2$  resulta em 3.
- **Multiplicação (\*):** Realiza o produto de dois valores.
  - Exemplo:  $2 * 6$  resulta em 12.
- **Divisão (/):** Divide um valor pelo outro.
  - Exemplo:  $10 / 2$  resulta em 5.



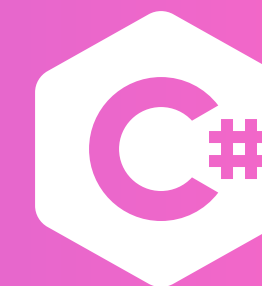


continuação...

## Operadores Aritméticos

Os principais operadores aritméticos são:

- **Módulo ou mod (%):** Retorna o resto da divisão entre dois valores.
  - Exemplo:  $7 \% 3$  resulta em 1 (resto da divisão de 7 por 3).
- **Exponenciação (\*\*):** Eleva um valor à potência de outro.
  - Exemplo:  $2 ** 3$  resulta em 8 (2 elevado à potência de 3).
- **Incremento (+ +):** Aumenta o valor da variável ou número em 1.
  - Exemplo:  $2++$  resulta em 3.
- **Decremento(- -):** Diminuí o valor da variável ou número em 1.
  - Exemplo:  $2--$  resulta em 1.



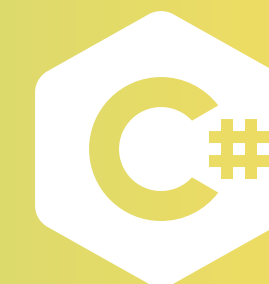




# Operadores Aritméticos

Tabela de operadores aritméticos:

Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Modulo	%
Exponenciação	**
Incremento	++
Decremento	--

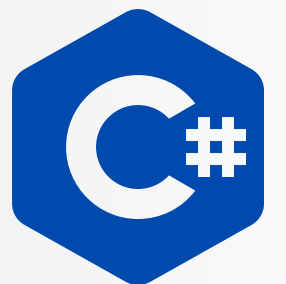


# Operadores Aritméticos

## Precedência de operadores aritméticos

Qual é o valor da variável **resultado**?

```
double resultado = 10 + 8 + 12 / 3;  
Console.WriteLine(resultado);
```



[https://www.w3schools.com/cs/trycs.php?filename=demo\\_compiler](https://www.w3schools.com/cs/trycs.php?filename=demo_compiler)

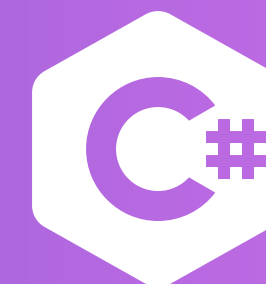




# Operadores Aritméticos

## Resultado da Precedência de operadores aritméticos

```
editor
1 10 + 8 + 12 / 3
2 10 + 8 + 4
3 22
4
```



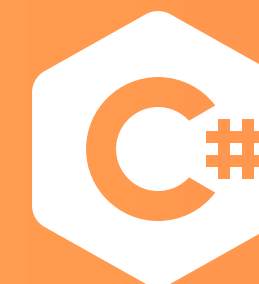


# Operadores Aritméticos

## Precedência de operadores aritméticos

Qual é o valor da variável **resultado**?

```
double resultado = (10 + 8 + 12) / 3;  
Console.WriteLine(resultado);
```

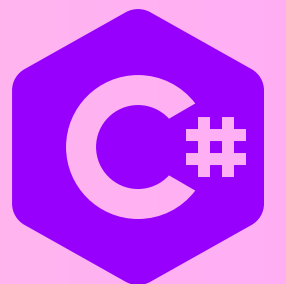




# Operadores Aritméticos

## Resultado da Precedência de operadores aritméticos

```
editor
1 (10 + 8 + 12) / 3
2 30 / 3
3 10
4
```

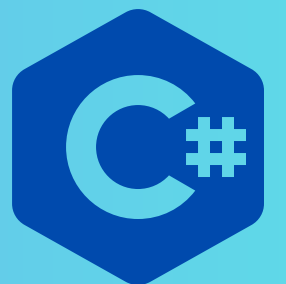


# Operadores Aritméticos

## Precedência de operadores aritméticos

Qual é o valor da variável **resultado**?

```
double resultado = (2 + 8) * 2 / 4;  
Console.WriteLine(resultado);
```



[https://www.w3schools.com/cs/trycs.php?filename=demo\\_compiler](https://www.w3schools.com/cs/trycs.php?filename=demo_compiler)





# Operadores Aritméticos

## Resultado da Precedência de operadores aritméticos

```
editor
1 (2 + 8) * 2 / 4
2 10 * 2 / 4
3 20 / 4
4 5
```

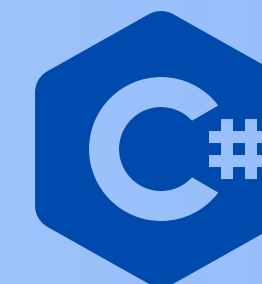




# Operadores Aritméticos

Tabela Precedência de operadores aritméticos

Parênteses	()
Exponenciação	**
Multiplicação, Divisão e Módulo	*, /, %
Adição, Subtração	+, -







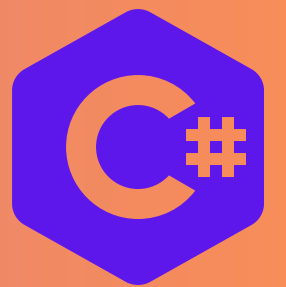
# O que vamos aprender

- ~~Operador de Atribuição;~~
- ~~Operadores Aritméticos;~~
- **Operadores de comparação;**
- ~~Operadores Lógicos;~~



# Operadores de comparação

Os operadores de comparação são utilizados para comparar dois valores e retornam um valor booleano, verdadeiro ou falso, com base no resultado da comparação. Operadores são fundamentais para criar expressões condicionais. Eles nos permitem tomar decisões com base nas comparações entre valores.





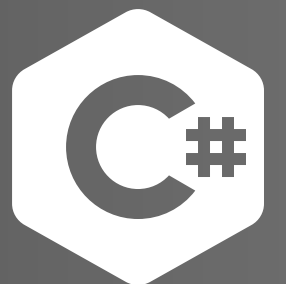
# Operadores de comparação

**Valor**

**Operador**

**Valor**

Retorna **true** ou **false**

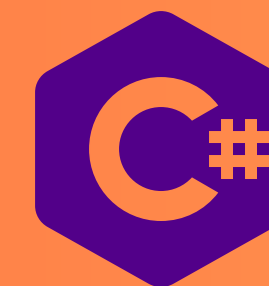




# Operadores de comparação

Vamos analisar alguns **operadores de comparação** mais comuns:

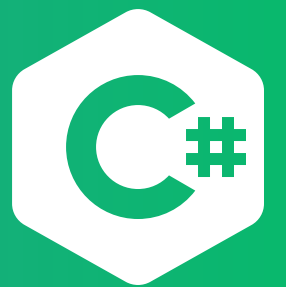
- **Igual (==)**: Retorna verdadeiro caso os operandos sejam iguais.
  - Exemplo: `3 == 3` retorna verdadeiro.
- **Não igual ou diferente (!=)**: Retorna verdadeiro caso os operandos não sejam iguais.
  - Exemplo: `3 != 4` retorna verdadeiro.
- **Maior que (>)**: Retorna verdadeiro se o operando à esquerda for maior que o operando à direita.
  - Exemplo: `5 > 3` retorna verdadeiro.





# Operadores de comparação

- **Menor que (<):** Retorna verdadeiro se o operando à esquerda for menor que o operando à direita.
  - Exemplo:  $2 < 7$  retorna verdadeiro.
- **Maior ou igual (>=):** Retorna verdadeiro se o operando à esquerda for maior ou igual ao operando à direita.
  - Exemplo:  $5 \geq 5$  retorna verdadeiro.
- **Menor ou igual (<=):** Retorna verdadeiro se o operando à esquerda for menor ou igual ao operando à direita.
  - Exemplo:  $4 \leq 6$  retorna verdadeiro.



# Operadores de comparação

Tabela de Operadores de comparação:

Igual	==
Não igual ou diferente	!=
Maior que	>
Menor que	<
Maior ou igual	>=
Menor ou igual	<=





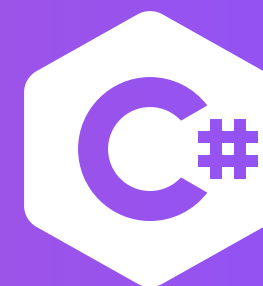
# Operadores de comparação

10

<

20

Qual é a Resposta?





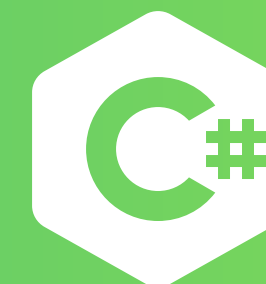
# Operadores de comparação

10

==

20

Qual é a Resposta?







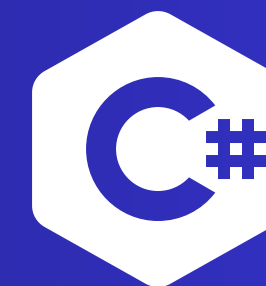
# Operadores de comparação

10

!=

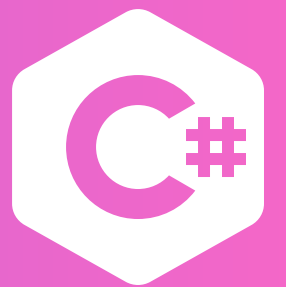
20

Qual é a Resposta?



# O que vamos aprender

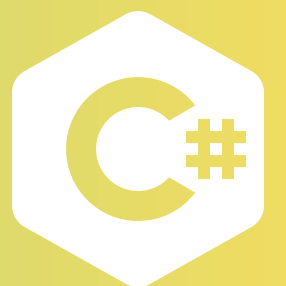
- ~~Operador de Atribuição;~~
- ~~Operadores Aritméticos;~~
- ~~Operadores de comparação;~~
- **Operadores Lógicos;**





# Operadores Lógicos

Os operadores lógicos são utilizados para combinar expressões booleanas e avaliar a lógica entre elas. Esses operadores são fundamentais para criar condições e tomar decisões. Eles nos permitem controlar o fluxo do programa com base em avaliações lógicas que resultam em true ou false e são especialmente úteis em instruções condicionais (if, while, etc.).

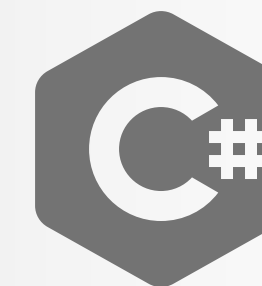




# Operadores Lógicos

## Operadores Lógicos em C#

- AND Lógico && (E)
- OR Lógico || (OU)
- NOT Lógico ! (NÃO)
- XOR Lógico ^ (OU Exclusivo)



# Operadores Lógicos

## AND Lógico && (E)

- Retorna true somente se ambas as expressões forem verdadeiras. Caso uma delas seja false, o resultado será false.
- Se a primeira expressão é false, a segunda expressão não é avaliada, pois o resultado final já será false.

Qual é a resposta?

```
bool a = true;  
bool b = false;  
Console.WriteLine(a && b);
```



# Operadores Lógicos

## OR Lógico || (OU)

- Retorna true se pelo menos uma das expressões for verdadeira. Retorna false somente se ambas as expressões forem false.
- Se a primeira expressão é true, a segunda expressão não é avaliada, pois o resultado final já será true.

Qual é a resposta?

```
bool a = true;  
bool b = false;  
Console.WriteLine(a || b);
```





# Operadores Lógicos

## NOT Lógico ! (NÃO)

- Inverte o valor lógico de uma expressão. Se a expressão é true, ! a torna false e vice-versa.

Qual é a resposta?

```
bool a = true;  
Console.WriteLine( !a );
```





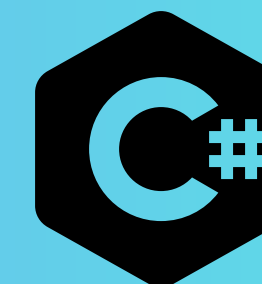
# Operadores Lógicos

## XOR Lógico ^ (OU Exclusivo)

- Retorna true somente se uma das expressões for true e a outra for false. Se ambas as expressões forem true ou ambas forem false, o resultado é false.

Qual é a resposta?

```
bool a = true;  
bool b = false;  
Console.WriteLine(a ^ b);
```







# Operadores Lógicos

## Tabela Verdade

### E Lógico (&&)

False	&&	False	=	False
True	&&	False	=	False
False	&&	True	=	False
True	&&	True	=	True

### OU Lógico (||)

False		False	=	False
True		False	=	True
False		True	=	True
True		True	=	True



# Operadores Lógicos

## Tabela Verdade

### Negação Lógica (!)

!False	=	True
!True	=	False

### OU Exclusivo (^)

False	^	False	=	False
True	^	False	=	True
False	^	True	=	True
True	^	True	=	False

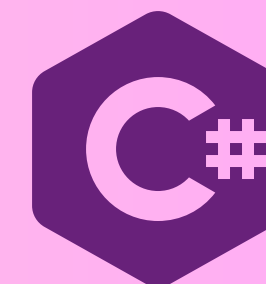




# Operadores Lógicos

Tabela de Precedência dos Operadores Lógicos

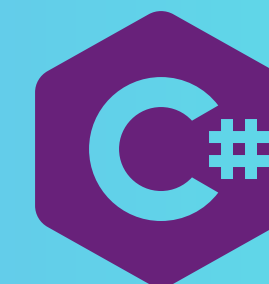
Negação lógica	!
E lógico	&&
OU lógico	
OU Exclusivo	^





# Operadores Lógicos

Operadores lógicos têm menor precedência que operadores aritméticos e de comparação. Isso significa que expressões aritméticas e comparações são avaliadas primeiro. Parênteses () também podem ser usados para alterar a ordem de precedência.





# Operadores Lógicos

## Exemplo de Precedência

Qual é a resposta?

```
bool a = true;  
bool b = false;  
bool c = false;  
bool resultado = !a || (b && c);  
Console.WriteLine(resultado);
```





# Operadores Lógicos

## Exemplo de Precedência

- Neste exemplo os parênteses tem a precedência mais alta e será avaliada primeiro, seguida pela Negação Lógica ! e por último o OU Lógico ||:

```
bool a = true;  
bool b = false;  
bool c = false;  
bool resultado = !a || (b && c);  
Console.WriteLine(resultado);
```

Resposta: False



# Operadores Lógicos

## Exemplo de Precedência

Qual é a resposta?

```
bool a = false;  
bool b = false;  
bool c = false;  
bool resultado = !a || b && c;  
Console.WriteLine(resultado);
```



# Operadores Lógicos

## Exemplo de Precedência

- Neste exemplo a Negação Lógica ! tem a precedência mais alta e será avaliada primeiro, seguida pelo E Lógico && e por último o OU Lógico ||:

```
bool a = false;  
bool b = false;  
bool c = false;  
bool resultado = !a || b && c;  
Console.WriteLine(resultado);
```

Resposta: True





# Exercícios em formato de jogo

**Acesse <https://kahoot.it/>**

Após acessar com o PIN do jogo adicione um nome para sua identificação, atenção pois existe um tempo predefinido para cada pergunta, você precisa responder antes de acabar o tempo, caso contrário, você não marca ponto no jogo. Você também pode alterar o icone do seu perfil antes de iniciar o jogo.





# FIM!

