

# C# Essencial



Exercícios - Fase 2

Fundamentos

## Exercícios fase 2

1- Declare as variáveis **nome**, **idade** e **nota** atribuindo os valores “**Paulo**”, **17** e **7.5** e exiba a saída no formato : **Aluno <nome> tem <idade> anos e nota <nota>** usando a *concatenação e a interpolação de strings*.

```
string nome = "Paulo";
int idade = 17;
double nota = 7.5;

Console.WriteLine(nome + " tem " + idade + " anos e a nota " + nota);
Console.WriteLine($"{nome} tem {idade} anos e a nota {nota}");

Console.ReadKey();
```

2- Para o exercício anterior exiba **o nome a idade e a nota** em linhas separadas usando as sequências de escapes.

```
string nome = "Paulo";
int idade = 17;
double nota = 7.5;

Console.WriteLine(nome + "\ntem " + idade + " anos \ne a nota " + nota);

Console.ReadKey();
```

## Exercícios fase 2

3- Para qual tipo de dados você pode converter um **float** *implicitamente* ?

- ☐ int
- ☒ double
- ☐ long
- ☒ decimal

byte - 1 byte
short – 2 bytes
int – 4 bytes
long – 8 bytes
float – 4 bytes
double – 8 bytes
decimal - 16 bytes

4- Em qual conversão numérica você precisaria realizar o casting (*convesão forçada*) ?

- ☐ int para long
- ☒ double para long
- ☒ double para float
- ☒ decimal para float
- ☒ long para int
- ☒ double para decimal

## Exercícios fase 2

5- Escreva um programa que recebe **3 letras via teclado** e as exiba na **ordem reversa** usando *a concatenação e também a interpolação de strings*

```
char letra1, letra2, letra3;

Console.Write("Informe a letra : ");
letra1 = Convert.ToChar(Console.ReadLine());

Console.Write("Informe a letra : ");
letra2 = Convert.ToChar(Console.ReadLine());

Console.Write("Informe a letra : ");
letra3 = Convert.ToChar(Console.ReadLine());

Console.WriteLine(letra3 + " " + letra2 + " " + letra1);
Console.WriteLine($"{letra3} {letra2} {letra1}");

Console.ReadKey();
```

Classe **Convert**

Métodos:

**Convert.ToInt32()**  
**Convert.ToDouble()**  
**Convert.ToFloat()**  
**Convert.ToString()**  
**Convert.ToChar()**  
**Convert.ToBool()**

## Exercícios fase 2

6- Marque verdadeiro(V) ou falso(F) para os códigos abaixo:

- ( **F** ) long resultado = 1.32;
- ( **V** ) var nome = “Maria”;
- ( **V** ) string resultado = 100.ToString();
- ( **V** ) A sequência de escape `\n` inclui uma nova linha
- ( **F** ) float f = 5.45;
- ( **V** ) decimal valor = (decimal) 10.99f;
- ( **F** ) var status = null;
- ( **V** ) object o = 12.45m;
- ( **V** ) string titulo = true.ToString();
- ( **F** ) A sequencia `\t` inclui uma tabulação vertical

<b>byte - 1 byte</b>
<b>short – 2 bytes</b>
<b>int – 4 bytes</b>
<b>long – 8 bytes</b>
<b>float – 4 bytes</b>
<b>double – 8 bytes</b>
<b>decimal - 16 bytes</b>

## Exercícios fase 2

7- Escreva um programa para receber **dois valores** via teclado do tipo **double** e a seguir realize as operações *de soma, subtração, multiplicação, exponenciação, divisão e módulo* exibindo o resultado:

```
Console.WriteLine("Informe o valor de x :");  
double x = Convert.ToDouble(Console.ReadLine());  
  
Console.WriteLine("Informe o valor de y :");  
double y = Convert.ToDouble(Console.ReadLine());  
  
Console.WriteLine($"Soma = {x+y}");  
Console.WriteLine($"Subtração = {x-y}");  
Console.WriteLine($"Multiplicação = {x*y}");  
Console.WriteLine($"Divisão = {x/y}");  
Console.WriteLine($"Módulo = {x%y}");  
Console.WriteLine($"Exponenciação = {Math.Pow(x,y)}");  
  
Console.ReadKey();
```

## Exercícios fase 2

8 - Faça um programa para calcular o resultado da **fórmula de baskara** dados os valores de **a b e c**.

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

Valores : **a = 1 , b = 12 e c = -13**

```
int a = 1, b = 12, c = -13;  
double delta, x1, x2;
```

```
delta = Math.Pow(b,2) - 4 * a * c;
```

```
x1 = ((-b) + Math.Sqrt(delta)) / 2 * a;
```

```
x2 = ((-b) - Math.Sqrt(delta)) / 2 * a;
```

```
Console.WriteLine(" x1 = " + x1);
```

```
Console.WriteLine(" x2 = " + x2);
```

```
Console.ReadKey();
```

Fórmula do delta

delta =>  $\Delta = b^2 - 4ac$

$$x1 = \frac{-b + \sqrt{\Delta}}{2.a}$$

$$x2 = \frac{-b - \sqrt{\Delta}}{2.a}$$

## Exercícios fase 2

9- Escreva um programa que receba *um nome e uma senha via teclado*. Nome é uma string e Senha é um inteiro. Se o nome for igual a **'admin'** ou **'maria'** e a senha for igual a **'123'** então exiba a mensagem *'Login feito com sucesso'* caso contrário exiba a mensagem *'Login inválido'*:

```
Console.Write("Informe o nome \t");
string nome = Console.ReadLine();

Console.Write("\nInforme a senha \t");
int senha = Convert.ToInt32(Console.ReadLine());

var resultado = (nome == "admin" || nome == "maria") && senha == 123 ? "Login Legal" : "Falhou";

Console.WriteLine(resultado);

Console.ReadKey();
```



## Exercícios fase 2

10- Indique verdadeiro(**V**) ou falso (**F**) para as seguintes declarações de variáveis considerando a nomenclatura usada:

**&&** -> retorna **false** se apenas uma das condições for **false**  
**||** -> retorna **true** se apenas uma das condições for **true**

- ( **V** ) **string? nome;** é um exemplo de *nullable reference type*;
- ( **F** ) para x igual a 0 , **Console.WriteLine(x++);** imprime o valor 1
- ( **F** ) A ordem de precedência dos operadores lógicos é : **!** , **||** e **&&** (**NOT**, **OR** e **AND**)
- ( **V** ) para y igual a 0 , **Console.WriteLine(++y);** imprime o valor 1
- ( **F** ) **(10 % 2 == 0) ? "Par" : "Impar";** Vai retornar "**Impar**"
- ( **F** ) Para **x=25** e **y=5** , a expressão **( y >= x ) && ( y <= x );** retorna **true**;
- ( **V** ) Os **nullable reference types** emitem um alerta em tempo de compilação para uso do **null**
- ( **F** ) Para **x = 10** , o código **Console.WriteLine(x+=x);** imprime o valor 10;
- ( **F** ) Para **x= 5** , a expressão **!(9 >= x) && x <= 7 || x == 5);** retorna **false**;
- ( **V** ) O operador **?.** permite verificar se um objeto é **null** e evitar o **NullReferenceException**
- ( **V** ) Os tipos de valor podem ser declarados como **anuláveis** com a ajuda do operador **"?"**

## Exercícios fase 2

11- Escreva um programa que recebe via teclado **dois números inteiros x e y** e imprima no console se x é par ou não e se y é par ou não. *Use o operador condicional ternário (? :)*

**Condicao ? expressao1\_true : expressao2\_false ;**

```
Console.Write("\nInforme o valor de x : \t");  
int x = Convert.ToInt32(Console.ReadLine());  
  
Console.Write("Informe o valor de y : \t");  
int y = Convert.ToInt32(Console.ReadLine());  
  
Console.WriteLine(x % 2 == 0 ? "x é par " : "x não é par");  
  
Console.WriteLine(y % 2 == 0 ? "y é par " : "y não é par");  
  
Console.ReadKey();
```

## Exercícios fase 2

12- Crie um programa que receba um numero inteiro x via teclado e calcule e imprima no console o resultado das seguintes operações : **(x^2 -> x ao quadrado) (pi = 3.1415)**

**-6 + x \* 5**

**(13-2) \* x**

**(x + -2) \* (20 / x)**

**(12 + x) / (x - 4)**

**3 x^2 + x + 10**

**pi \* x^2**

**(), \*, /, %, + e -**

```
Console.Write("Informe o valor de x : \t");
int x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(-6 + x * 5);
Console.WriteLine((13 - 2) * x);
Console.WriteLine((x + -2) * (20 / x));
Console.WriteLine((12 + x) / (x - 4));

double resultado = 3 * Math.Pow(x, 2) + x + 10 ;
double area = Math.PI * Math.Pow(x,2);

Console.WriteLine(resultado);
Console.WriteLine(area);

Console.ReadKey();
```

## Exercícios fase 2

13- Considere o seguinte trecho de código:

```
int y = 5 ;
```

```
y = (y++)+y+(++y);      ou      y = y++ + y + ++y;
```

```
Console.WriteLine(y);
```

O valor de y será 18.

**y++** vai retornar 5 e então incrementará para 6 -> **primeiro resolve e depois incrementa**  
**++y** vai incrementar 6 para 7 -> **primeiro incrementa depois resolve**

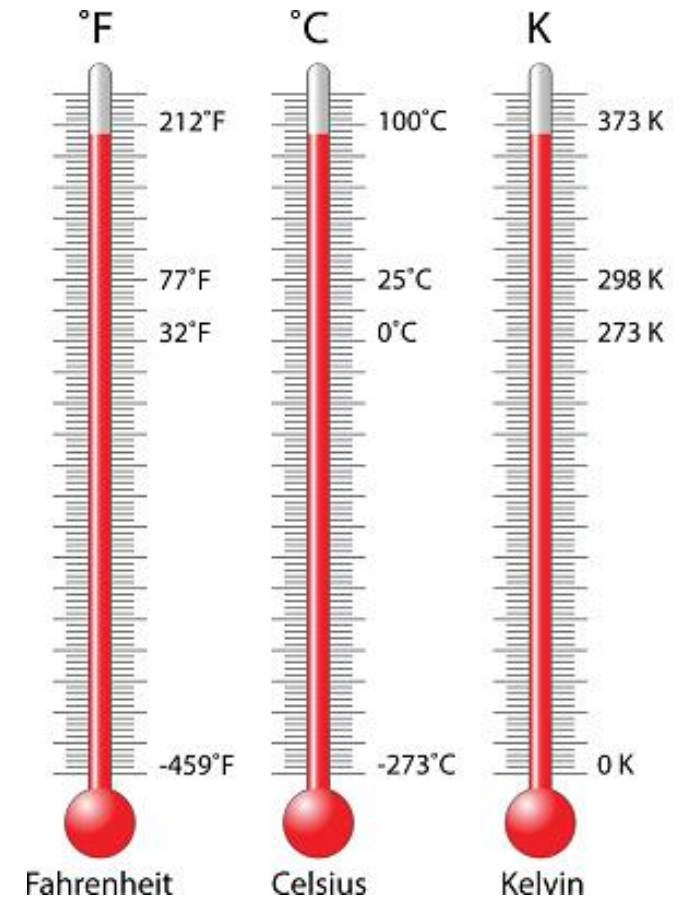
logo: **y = (5+6+7) = 18**

## Exercícios fase 2

14- Escreva um programa que solicite a temperatura em graus **Celsius** e converta para **Kelvin** e **Farhenheit** usando as fórmulas a seguir:

- Converter para Kelvin =>  **$K = C + 273$**  ;
- Converter para Farhenheit =>  **$F = (C * 9) / 5 + 32$**  ;

```
Console.WriteLine("Informe o valor em graus Celsius: ");  
double celsius = Convert.ToDouble(Console.ReadLine());  
  
Console.WriteLine($"Graus Kelvin = {celsius + 273}");  
  
double farhenheit = (celsius * 9) / 5 + 32;  
  
Console.WriteLine($"Graus Fahrenheit = {farhenheit}");  
  
Console.ReadKey();
```



## Exercícios fase 2

15 - Escolha a opção que representa a exibição do resultado para o código usando os *operadores de decremento e incremento (pré e pós)* :

```
var numero = 5;  
Console.WriteLine(numero++);  
numero = 1;  
Console.WriteLine(++numero);  
numero = 2;  
Console.WriteLine(numero--);  
numero = 3;  
Console.WriteLine(--numero);  
  
Console.ReadKey();
```

( x )	( )	( )	( )
5	6	5	6
2	3	2	2
2	2	1	1
2	2	2	3