

Hello World

Um pouco de Teoria não faz mal a ninguém.

Vamos seguir as Boas Práticas de Código

Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Errado:

```
int 5minutos;  
int salário;  
int salario do funcionario;
```

Correto:

```
int _5minutos;  
int salario;  
int salarioDoFuncionario;
```

Vamos seguir as Boas Práticas de Código

Convenções

- Camel Case: `lastName` (parâmetros de métodos, variáveis dentro de métodos)
- Pascal Case: `LastName` (namespaces, classe, properties e métodos)
- Padrão `_lastName` (atributos "internos" da classe)

Vamos seguir as Boas Práticas de Código

- ◆ **CONCATENAR:** ligar(-se), juntar(-se) numa cadeia ou sequência lógica ou orgânica.

Placeholders, concatenação e interpolação

Demo:

```
int idade = 32;
double saldo = 10.35784;
String nome = "Maria";

Console.WriteLine("{0} tem {1} anos e tem saldo igual a {2:F2} reais", nome, idade, saldo);

Console.WriteLine($"{nome} tem {idade} anos e tem saldo igual a {saldo:F2} reais");

Console.WriteLine(nome + " tem " + idade + " anos e tem saldo igual a "
    + saldo.ToString("F2", CultureInfo.InvariantCulture) + " reais");
```


Vamos **PRATICAR** as Boas Práticas de Código

◈ EXERCÍCIO DE FIXAÇÃO

Em um novo programa, inicie as seguintes variáveis:

```
string produto1 = "Computador";  
string produto2 = "Mesa de escritório";  
  
byte idade = 30;  
int codigo = 5290;  
char genero = 'M';  
  
double preco1 = 2100.0;  
double preco2 = 650.50;  
double medida = 53.234567;
```

Em seguida, usando os valores das variáveis, produza a seguinte saída na tela do console:

```
Produtos:  
Computador, cujo preço é $ 2100,00  
Mesa de escritório, cujo preco é $ 650,50  
  
Registro: 30 anos de idade, código 5290 e gênero: M  
  
Medida com oito casas decimais: 53,23456700  
Arredondado (três casas decimais): 53,235  
Separador decimal invariant culture: 53.235
```

```
qonpj6 ueq7q9 = 23'532032  
qonpj6 bleco5 = 020'20'  
qonpj6 bleco7 = 3700'0'
```

```
26b9L9q0L q6C7w9J 7uV9L79uF cN7f0L6: 23'532  
yLLeqouq9q0 (fL62 c9292 q6C7w972): 23'532  
ueq7q9 com 0710 c9292 q6C7w972: 23'53203200
```

Operadores Aritméticos

- ◆ Esses são os sinais pra fazer conta e operações lógicas

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

Operadores de Atribuição

- ◊ O que são operadores de atribuição?
- ◊ Na Programação, em algum momento você vai se deparar com a seguinte situação:

```
string mensagem = "Esta é uma mensagem";
```

- ◊ Perceba que temos o sinal "=" logo após declararmos o tipo e o nome da variável.

- ◊ Então quando a gente diz que
String mensagem = "Esta é uma mensagem"
- ◊ O sinal "=" não significa igual, como na matemática.
- ◊ O sinal "=" significa recebe. Ou seja, quando temos essa mensagem, falamos que:
- ◊ A variável do tipo string **mensagem** recebe a string(ou o texto) "essa é uma mensagem"

Operadores de Atribuição

Operador	Exemplo	Significado
=	a = 10;	a RECEBE 10
+=	a += 2;	a RECEBE a + 2;
-=	a -= 2;	a RECEBE a - 2;
*=	a *= 2;	a RECEBE a * 2;
/=	a /= 2;	a RECEBE a / 2;
%=	a %= 3;	a RECEBE a % 3;

Vamos usar dois exemplos daqui.

Temos `a=10;`

E temos `a += 2;`

Se a gente escrever o código

```
Int a=10;  
Console.WriteLine(a);
```

O console vai nos mostrar:

10

Se a gente escrever o código

```
Int a=10;  
a+=2;  
Console.WriteLine(a)
```

O console vai nos mostrar:

12

Incremento e Decremento

Como programador, em algum momento da sua vida você vai se deparar com isso:

```
for (int i = 0; i < 10; i++)  
{  
    //Código a ser executado  
}
```

O for é uma estrutura de repetição. Mas não nos interessa por agora.

O nosso objetivo aqui é o `i++`.

Funciona mais ou menos assim.

`(int i = 0; i < 10; i++)` significa que temos uma variável `i` que recebe 0. uma condição no meio `i < 10` (enquanto for menor que 10), e no final o incremento `i++`

Operador	Exemplo	Significado
++	<code>a++;</code> ou <code>++a;</code>	<code>a = a + 1;</code>
--	<code>a--;</code> ou <code>--a;</code>	<code>a = a - 1;</code>

```
int a = 10;  
a++;  
Console.WriteLine(a);  
  
SAÍDA:  
11
```

```
int a = 10;  
int b = a++;  
Console.WriteLine(a);  
Console.WriteLine(b);  
  
SAÍDA:  
11  
10
```

```
int a = 10;  
int b = ++a;  
Console.WriteLine(a);  
Console.WriteLine(b);  
  
SAÍDA:  
11  
11
```

Se escrevermos o seguinte código:

```
for (int i = 0; i < 10; i++)  
{  
    Console.WriteLine(i);  
};
```

O Console vai nos mostrar:

0
1
2
3
4

Até chegar em 10 valores.

Exercício

Chega de lenga lenga e vamos praticar.

Fazer um programa para executar a seguinte interação com o usuário, lendo os valores destacados em vermelho, e depois mostrar os dados na tela:

Entre com seu nome completo:

Alex Green

Quantos quartos tem na sua casa?

3

Entre com o preço de um produto:

500.50

Entre seu último nome, idade e altura (mesma linha):

Green 21 1.73

SAÍDA ESPERADA (NÚMEROS REAIS COM DUAS CASAS DECIMAIS):

Alex Green

3

500.50

Green 21 1.73

Final

Se você chegou até aqui

Nos vemos no próximo PDF

