

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

## Tipos de dados

**Prof.**

**Samuel Ribeiro (Samuka)**



<http://www.processware.com.br>

# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Métodos

- **Rotina:**

Executa alguma ação e não retorna nada.

- **Função:**

Executa alguma ação e retorna algum resultado sempre

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Métodos

<modificador de acesso> **void** <nome da rotina>()

```
{  
    ...  
}
```

Exemplo:

```
public void MinhaRotina()  
{  
    x = 1;  
}
```

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Métodos

```
<modificador de acesso> <tipo de dado>  
<nome da função>()
```

```
{
```

```
...
```

```
}
```

Exemplo:

```
public string MinhaFuncao()  
{  
    return "Isto é uma função pois retorna algo.";  
}
```

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Tipos de dados

#### Variáveis

Podemos dizer que uma variável é um espaço alocado na memória RAM para ser utilizada por um ou mais processos que necessitam de armazenar ou manipular alguma informação. Ou ainda, variáveis representam locais de armazenamento.

Para C#: **TIPO** *NomeDaVariavel*

Exemplo:

```
string Recebe_Indice;
```

Se estas variáveis forem declaradas dentro de um procedimento ou bloco, poderão ser acessíveis apenas dentro deste procedimento ou bloco.

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Tipos de dados

Podemos armazenar muitos tipos de informações diferentes dentro de uma variável, como números, datas, e strings.

Nesta linguagem, seus tipos podem ter dois tipos:

- Por Valor: os valores são gerenciados diretamente na memória.
- Por Referencia: os valores são passados por meio de um ponteiro.

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Tipos de dados

Tipo C#	Tipo .NET(CTS)	Faixa de dados	Descrição
bool	System.Boolean	<i>true</i> ou <i>false</i>	Booleano
byte	System.Byte	-127 a 128	Inteiro de 8-bit com sinal
char	System.Char	U+0000 a U+ffff	Caracter Unicode de 16-bit
decimal	System.Decimal	$1,0 \times 10^{-28}$ a $7,9 \times 10^{28}$	Inteiro de 96-bit com sinal com 28-29 dígitos significativos
double	System.Double	$\pm 5,0 \times 10^{-324}$ a $\pm 1,7 \times 10^{308}$	Flutuante IEEE 64-bit com 15-16 dígitos significativos
float	System.Single	$\pm 1,5 \times 10^{-45}$ a $\pm 3,4 \times 10^{38}$	Flutuante IEEE 32-bit com 7 dígitos significativos
int	System.Int32	-2.147.483.648 a 2.147.483.647	Inteiro de 32-bit com sinal
long	System.Int64	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807	Inteiro de 64-bit com sinal
Object	System.Object		Classe base
Sbyte	System.Sbyte	0 a 255	Inteiro de 8-bit sem sinal
Short	System.Int16	-32,768 a 32,767	Inteiro de 16-bit com sinal
String	System.String		String de caracteres Unicode
UInt	System.UInt32	0 a 4,294,967,295	Inteiro de 32-bit sem sinal
Ulong	System.UInt64	0 a 18,446,744,073,709,551,615	Inteiro de 64-bit sem sinal
Ushort	System.UInt16	0 a 65,535	Inteiro de 16-bit sem sinal

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Objeto SESSION

Na primeira solicitação de um determinado cliente ao aplicativo ASP.Net, é criado um objeto do tipo Dictionary chamado Session. Nele podemos guardar qualquer valor com strings e números que poderão ser lidos em qualquer página da aplicação.

### Session["Nome"]

variáveis de sessão podem ser lidas em todas as páginas. Este recurso é muito utilizado para contar o numero de usuários conectados no site por exemplo.

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Operadores

- \* Operadores aritméticos
- \* \* Operadores de atribuição
- \* \* Operadores relacionais
- \* Operadores lógicos

#### Operadores aritméticos

+	(Adição)
-	(Subtração)
*	(Multiplicação)
/	(Divisão)
%	(Resto/Módulo)

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Operadores

- \* Operadores aritméticos
- \* \* Operadores de atribuição
- \* \* Operadores relacionais
- \* Operadores lógicos

#### Operadores de atribuição

=	(Atribuição simples)
+=	(Atribuição aditiva)
-=	(Atribuição Subtrativa)
*=	(Atribuição Multiplicativa)
/=	(Atribuição de divisão)
%=	(Atribuição de módulo)

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Operadores

- \* Operadores aritméticos
- \* \* Operadores de atribuição
- \* \* Operadores relacionais
- \* Operadores lógicos

#### Operadores relacionais

<b>==</b>	<b>(Igualdade)</b>
<b>&gt;</b>	<b>(Maior)</b>
<b>&lt;</b>	<b>(Menor)</b>
<b>&lt;=</b>	<b>(Menor igual)</b>
<b>&gt;=</b>	<b>(Maior igual)</b>
<b>!=</b>	<b>(Diferente)</b>

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Operadores

- \* Operadores aritméticos
- \* \* Operadores de atribuição
- \* \* Operadores relacionais
- \* Operadores lógicos

#### Operadores lógicos

**&&** (E)

**||** (OU)

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### Operadores

Incrementar um número ou variável significa adicionar 1 ao valor atual, da mesma forma que subtrair 1 de um número ou variável se dá o nome de Decremento.

#### Incremento e Decremento

++	(Incremento)
--	(Decremento)

Exemplo:

```
int X = 1;  
X++;
```