

## Tutorial 2 – Básico do VBA do Excel

Este tutorial contém a 2ª lição sobre a série de Tutoriais do VBA Básico do Excel. Ele cobre tópicos dos Objetos e Coleções mais usados do Excel. Iniciantes na programação de VBA serão encorajados a percorrerem de cabo a rabo as 1ª lições desta série se eles ainda não fizeram isto. Este documento contém informação sobre os seguintes tópicos.

[Objetos e Coleções](#)

[Objeto Workbook e Worksheet](#)

[Objeto Range e Propriedade Cells](#)

[Métodos e Propriedades](#)

[Atribuindo Variáveis Objeto e Usando Argumento Nomeado](#)

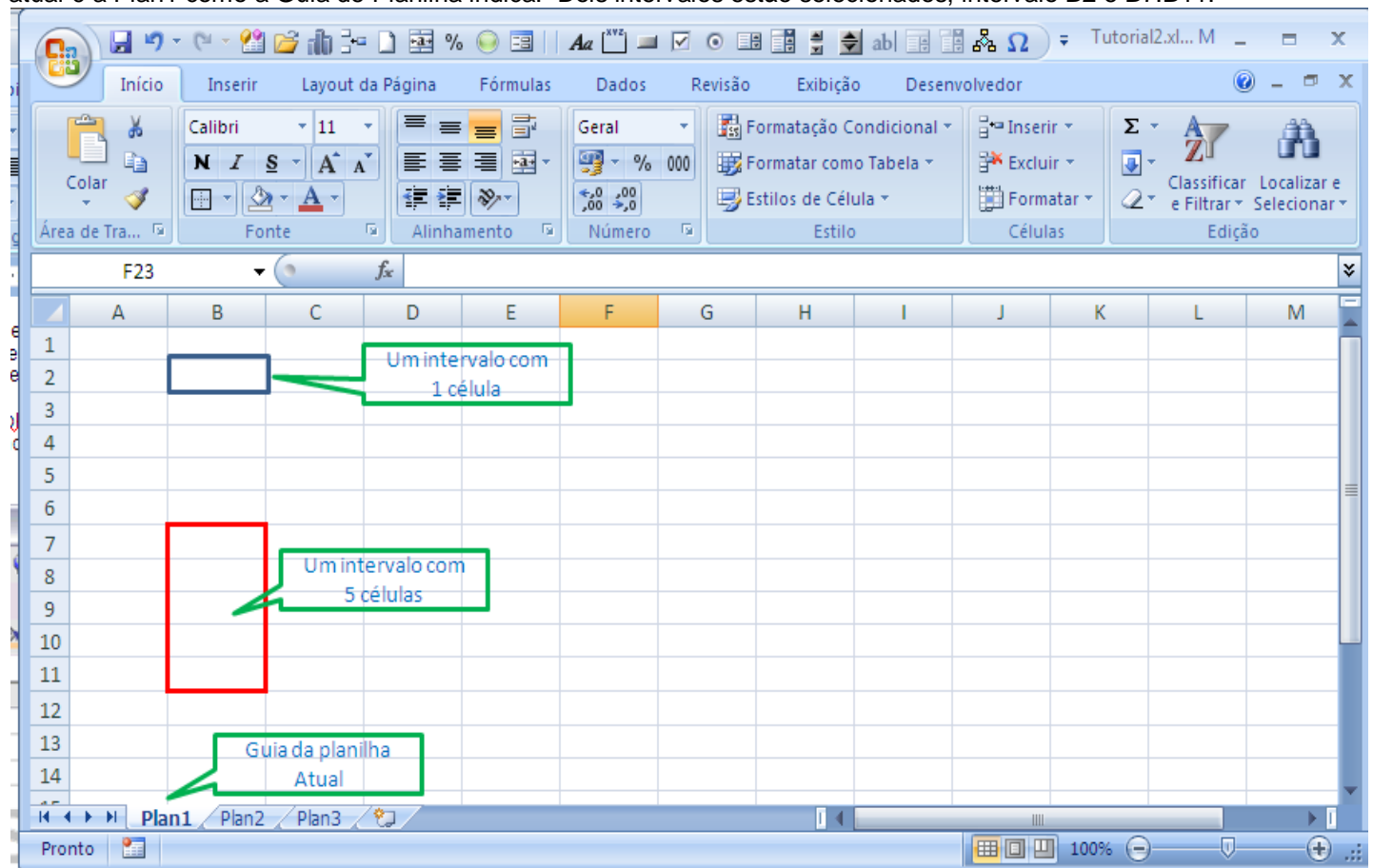
Microsoft Support site ou a seção Ajuda (Help) do VBA do Excel no seu computador contém exemplos compreensivos da maioria das coisas cobertas neste tutorial. Para mais informação, por favor refira-se a elas.

### Objetos e Coleções [Microsoft Support](#)

Objetos são os blocos de construção fundamentais do Visual Basic. Um **objeto** é um tipo especial de variável que contém ambos, dados e códigos. A **coleção** é um grupo de objetos da mesma classe. Os objetos mais usados do Excel na programação VBA são Workbook, Worksheet, Sheet, e Range.

Workbooks é uma coleção de todos os objetos Workbook. Worksheets é uma coleção de objetos Worksheet. O objeto Workbook representa uma workbook, o objeto Worksheet representa uma planilha, o objeto Sheet representa uma planilha ou gráfico, e o objeto Range representa um intervalo de células.

A figura seguinte mostra todos os objetos mencionados. A workbook (arquivo Excel) atual é a Book3.xls. A planilha atual é a Plan1 como a Guia de Planilha indica. Dois intervalos estão selecionados, intervalo B2 e B7:B11.



## Objetos Workbook e Worksheet

Um objeto **workbook** é o mesmo que um arquivo Excel. A coleção Workbook contém todas as workbooks que estão atualmente abertas. Dentro de uma workbook contém no mínimo uma **worksheet**. Em VBA, uma worksheet pode ser referenciada como segue:

```
Worksheets("Plan1")
```

Worksheets("Plan1") é a worksheet chamada "Plan1."

Outro modo de se referir a uma planilha é usar números índices como o que segue:

```
Worksheets(1)
```

Acima se refere à primeira planilha na coleção.

\* Note que a Worksheets(1) não é necessariamente a mesma planilha que Worksheets("Plan1").

**Sheets** é uma coleção de planilhas e gráficos (se presente). Uma planilha pode ser indexada exatamente como uma planilha. **Sheets(1)** é a primeira planilha na workbook.

Para se referir as sheets (ou outros objetos) com o mesmo nome, você tem de qualificar o objeto. Por exemplo:

```
Workbooks("Book1").Worksheets("Plan1")  
Workbooks("Book2").Worksheets("Plan1")
```

se o objeto não for qualificado, o objeto ativo ou atual (por exemplo workbook ou worksheet) é usado.

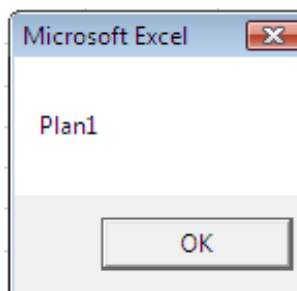
A guia planilha no fundo da planilha (worksheet) mostra qual planilha está ativa. Como mostra a figura abaixo, a planilha ativa é "Plan1" (mostrada em fonte negrito e fundo branco).



\* Você pode mudar a cor das guias de planilha clicando com o botão direito do mouse na guia, escolhendo a Guia Cor, e selecionando a cor para a guia.

A sub-rotina abaixo mostra o nome de cada planilha na workbook atualmente aberta. Você pode usar o laço **For Each...Next** para circular pela coleção Worksheets.

```
Sub MostrarPlanilhas()  
    Dim minhaPlanilha As Worksheet  
  
    For Each minhaPlanilha In Worksheets  
        MsgBox minhaPlanilha.Name  
    Next minhaPlanilha  
  
End Sub
```



Clicando Ok vai aparecendo os outros nomes

## Objeto Range e Propriedade Cells

**Range** representa uma célula, uma linha, uma coluna, uma seleção de células contendo um ou mais blocos contíguos de células, ou um intervalo 3-D. Mostraremos a você alguns exemplos sobre como o objeto Range pode ser usado.

O exemplo seguinte coloca o texto "AB" no intervalo A1:B5, na Plan2.

```
Worksheets("Plan2").Range("A1:B5") = "AB"
```

	A	B	C	D	E
1	AB	AB			
2	AB	AB			
3	AB	AB			
4	AB	AB			
5	AB	AB			
6					

Note que, `Worksheets.Range("A1", "B5") = "AB"` levará ao mesmo resultado que o exemplo acima.

O que segue coloca "AAA" na célula A1, A3, e A5 na Plan2.

```
Worksheets("Plan2").Range("A1, A3, A5") = "AAA"
```

	A	B	C	D
1	AAA			
2				
3	AAA			
4				
5	AAA			
6				

O objeto **Range** tem uma propriedade **Cells**. Esta propriedade é usada em todos os projetos VBA deste tutorial (**muito importante**). A propriedade **Cells** carrega um ou dois índices como seus parâmetros.

Por exemplo,

```
Cells(índice) ou Cells(linha, coluna)
```

onde *linha* é o índice linha e *coluna* é o índice coluna.

As três declarações seguintes podem ser trocadas:

```
ActiveSheet.Range.Cells(1,1)  
Range.Cells(1,1)  
Cells(1,1)
```

O que segue retorna a mesma saída:

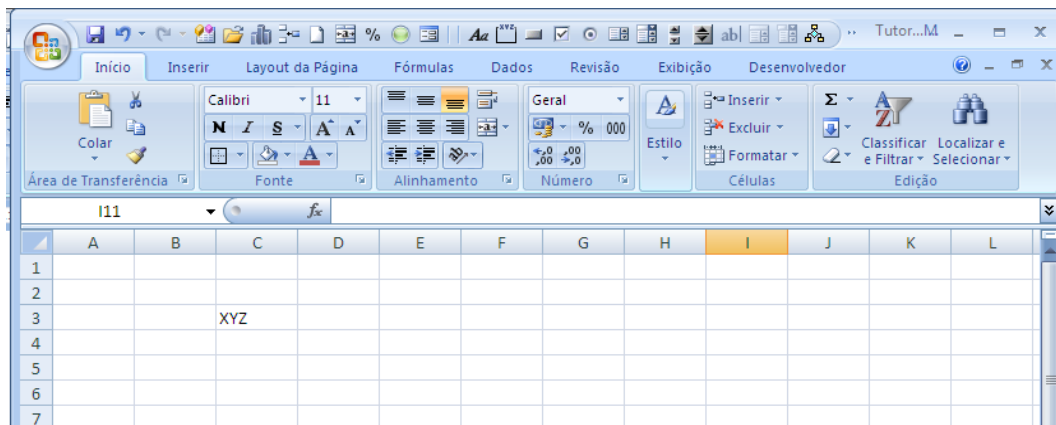
```
Range("A1") = 123 e Cells(1,1) = 123
```

O que segue coloca "XYZ" na Célula(1,12) ou Range("L1") assumindo que a célula A1 é a célula atual:

```
Cells(12) = "XYZ"
```

O que segue coloca "XYZ" na célula C3:

```
Range("B1:F5").cells(12) = "XYZ"
```



\* O pequeno número cinza em cada uma das células é apenas para propósitos de referência somente. Eles são usados para mostrar como as células estão indexadas dentro do intervalo.

Aqui está uma sub rotina que imprime a os correspondentes índices de linha e de coluna correspondente de A1 a E5.

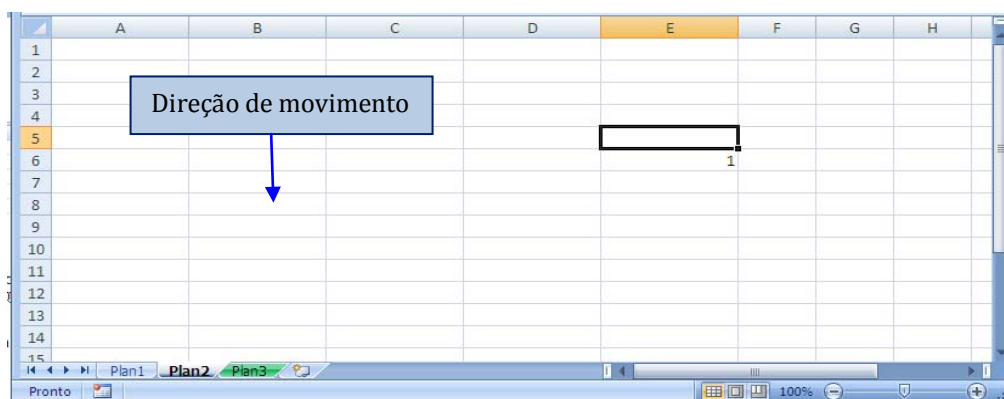
```
Sub ExemploCélulas()
    For i = 1 To 5
        For j = 1 To 5
            Cells(i, j) = "Row " & i & " Col " & j
        Next j
    Next i
End Sub
```

	A	B	C	D	E	F	G	H
1	Linha 1 Coluna 1	Linha 1 Coluna 2	Linha 1 Coluna 3	Linha 1 Coluna 4	Linha 1 Coluna 5			
2	Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3	Linha 2 Coluna 4	Linha 2 Coluna 5			
3	Linha 3 Coluna 1	Linha 3 Coluna 2	Linha 3 Coluna 3	Linha 3 Coluna 4	Linha 3 Coluna 5			
4	Linha 4 Coluna 1	Linha 4 Coluna 2	Linha 4 Coluna 3	Linha 4 Coluna 4	Linha 4 Coluna 5			
5	Linha 5 Coluna 1	Linha 5 Coluna 2	Linha 5 Coluna 3	Linha 5 Coluna 4	Linha 5 Coluna 5			

O objeto Range tem uma propriedade **Offset** que pode ser muito útil quando se quer mover ao redor da célula ativa. Os exemplos seguintes demonstram como a propriedade Offset pode ser implementada (assuma a célula atual antes de se mover seja a E5):

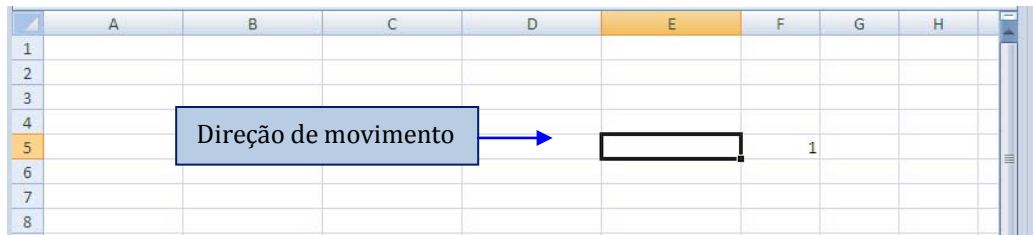
```
ActiveCell.Offset(1,0) = 1
```

Coloca um "1" uma linha abaixo de E5 (em E6)



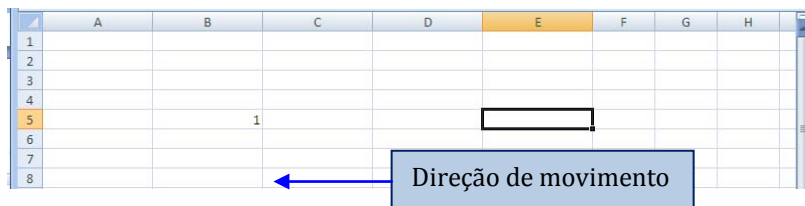
```
ActiveCell.Offset(0,1) = 1
```

Coloca um "1" uma coluna à direita de E5 (em F5)



```
ActiveCell.Offset(0,-3) = 1
```

Coloca um "1" três colunas à esquerda de E5 (em B5)



## Métodos e Propriedades

Cada objeto contém seus métodos e propriedades próprios.

A **Propriedade** representa uma característica embutida ou definida pelo usuário do objeto. Um **método** é uma ação que você faz num objeto. Abaixo estão exemplos de um método e uma propriedade para o Objeto Workbook:

```
Workbooks.Close
```

O método **Close** fecha a workbook ativa

```
Workbooks.Count
```

A propriedade **Count** retorna o número de workbooks que estão abertas atualmente

Alguns objetos têm propriedades default. Por exemplo, a propriedade default do **Range** é o **Value**. O que segue leva ao mesmo resultado.

```
Range("A1") = 1 e Range("A1").Value = 1
```

Aqui estão exemplos de como configurar e obter o valor da propriedade **Range**:

O que segue configura o valor do range A1 ou Cells(1,1) como "2005". Ele realmente imprime "2005" em A1.

```
Range("A1").Value = 2005
```

O que segue dá o valor do range A1 ou Cells(1,1).

```
X = Range("A1").Value
```

Métodos podem ser usados com ou sem argumento(s). Os dois exemplos seguintes demonstram seu comportamento.

Métodos Que Não Levam Nenhum Argumento:

```
Worksheets("Plan").Column("A:B").AutoFit
```

Métodos Que Levam Argumentos:

```
Worksheets("Plan1").Range("A1:A10").Sort _  
Worksheets("Plan1").Range("A1")
```

Worksheets("Plan1").Range("A1") é a **Key** (ou coluna) para classificar por ela.

### Atribuindo Variáveis a Objetos e Usando Argumentos Nomeados

Às vezes um método leva mais do que um argumento. Por exemplo, o método **Open** para o objeto **Workbook**, leva 12 argumentos. Para abrir uma workbook com proteção de password, você precisará escrever o código seguinte:

```
Workbooks.Open "Book1.xls", , , , "pswd"
```

Como este método leva tantos argumentos, é fácil colocar o argumento password no lugar errado. Para superar este problema potencial, podem-se usar argumentos nomeados como o exemplo seguinte:

```
Workbook.Open fileName:="Book1.xls", password:="pswd"
```

Você pode também atribuir uma variável a um objeto Declaração **Set**.

Por exemplo:

```
Dim meuIntervalo as Range  
Set meuIntervalo = Range("A1:A10")
```

[Continuar como Tutorial 3 do Básico do VBA do Excel](#)