Visual Basic For Applications

SINTAXE VBA

Conceitos Básicos

- Worksheet Code Name: Nome técnico de referência da planilha no arquivo.

  Ex.: Sheet1
- Worksheet Name: Nome visível de referência da planilha no arquivo.

  Ex.: Contas Contábeis
- Módulo: Projeto de criação/edição de macros.
- Subrotina (Sub): Conjunto de comandos que formam uma macro.
- Objeto: Aquilo que será manipulado (Worksheet, Range etc.).
- Método: O tipo de ação que será executada sobre o objeto (Offset, Resize etc.).
- Propriedade: Comandos configuráveis sobre um objeto/método (Select, Value etc.).

Tipos de dados

- Byte: Valores de 0 a 255.
- String: valores de texto.
- Boolean: valores True ou False.
- Object: objeto do VBA.
- Integer: números inteiros até 32768.
- Long: armazena e mostra números inteiros até 2.147.483.647.
- Single: números com até 6 decimais.
- Double: números com até 14 decimais.
- Decimal: números com até 28 decimais.
- Currency: valores monetários.
- Date: datas.
- Variant: qualquer tipo de dado sob um grande consumo de memória.

Objetos e Propriedades

- Object é o alvo da programação, aquilo se será afetado por uma operação.
- Property é o componente do Object, o qual será diretamente acionado ou alterado para executar uma operação. Podem ser do tipo Read-Only e/ou Write.
- Method é o que um Object faz. Podem ser usados para alterar propriedades.

Nível das Variáveis

- Local: Definida dentro de um procedimento e válida somente nele.
- Modular: Definida antes de todos os procedimentos de dado módulo (projeto VBA).
- Global/Pública: Definida antes de todos os procedimentos de dado módulo com o termo ‘Public’.
- Static: Definida com este termo ao invés de *Dim*, salvando o valor a cada execução.

Definição de Constante e Variável

- Contante:

  Const DiasMes = 30

- Variável:

  Dim message as String
  message = “Hello”

Regras para nomenclatura:

- Não pode iniciar com números;
- Não pode haver espaços ou pontos;
- Não pode utilizar caracteres especiais: !, %, ?, #, $.

OBS: Quando não houver a declaração da variável, somente a atribuição de valor, o Excel irá determinar o tipo de dado automaticamente para *Variant*. Para mitigar estas ocorrências, inicie o modulo com a instrução *Option Explicit*, que gerará um erro em caso de não declaração de variável.

Definição de Arrays

```
Dim MyArray(4) As Integer                          | Iniciado com primeiro índice em 0
Dim MyArray(1 To 5) As Integer                     | Iniciado conforme range definido
Dim MyArray(5, 4) As Integer                       | Duas dimensões (6 linhas e 5 colunas)
Dim MyArray(1 To 5, 1 To 4, 1 To 6) As Integer     | Tridimensional
```

Definição de Macro

```
Sub nomedamacro()
        *CÓDIGO*
End Sub
```

- Aspectos Adicionais:

Para executar uma Sub dentro de outra, basta apenas adicionar a palavra-chave *Call* e o nome da Sub pretendida.

As Subs por padrão são públicas, portanto visíveis na listagem de macros do Excel. Para ocultar uma Sub, deve-se adicionar a palavra-chave *Private* ao início da declaração da mesma – Ex.: Private MinhaSub().

Definição de Função

Function nomedafuncao() As TipodeDado

CÓDIGO

End Function

OBS: Para a acessar a relação de funções pré-criadas, deve-se utilizar a notação “*WorksheetFunction.*”.

Inserção de Comentário

- ‘ Comentário
- ‘ Comentário _

Continuação na linha seguinte

Seleção de Células

- Célula única:

```
Range("C2").Select
Cells(1, 1).Select
```

- Intervalo:

```
Range("A1:C2").Select
```

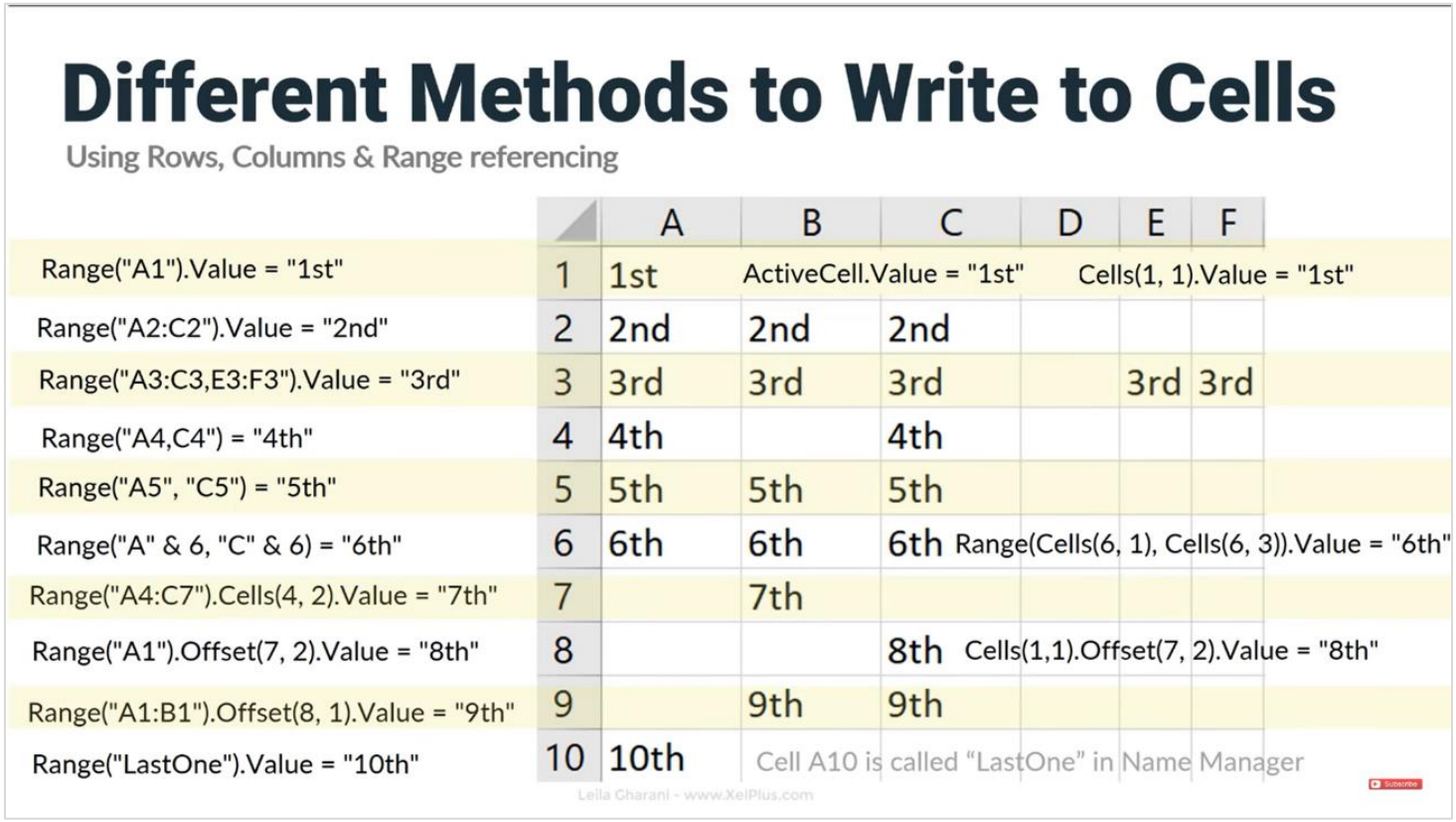
- Conjunto:

```
Set Minha_Lista = Range(“A1:A4”)
Minha_Lista.Select
```

Navegação

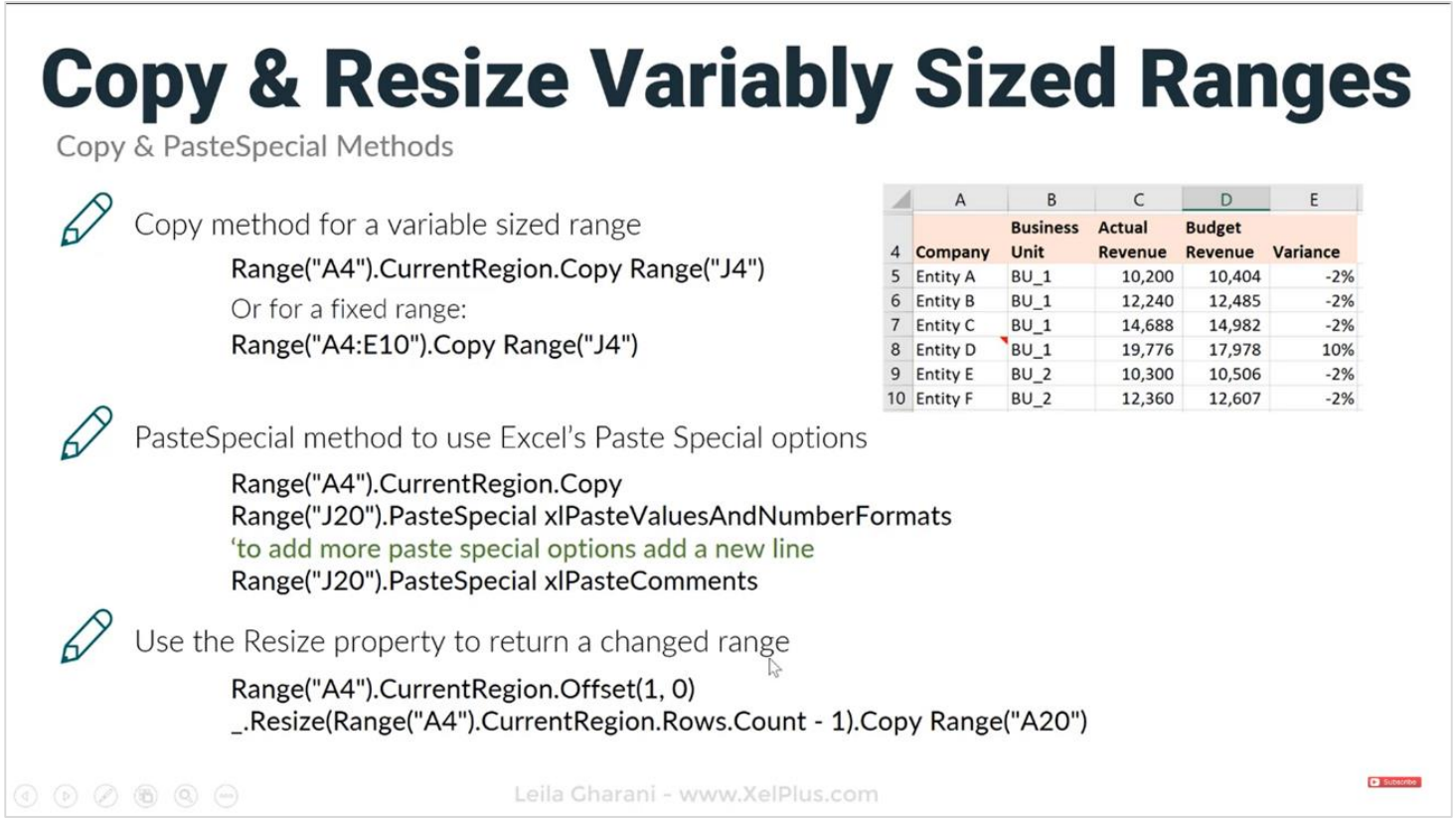
- ActiveCell.Offset(1, 2).Select | Deslocamento por linhas e colunas
- ActiveCell.Offset(-1, -2).Select | Deslocamento negativo por linhas e colunas
- ActiveCell.Resize(1, 2).Select | Seleção de range de linhas e colunas
- Sheets("Sheet2").Activate | Mudança de planilha ativa

Alteração de Valores



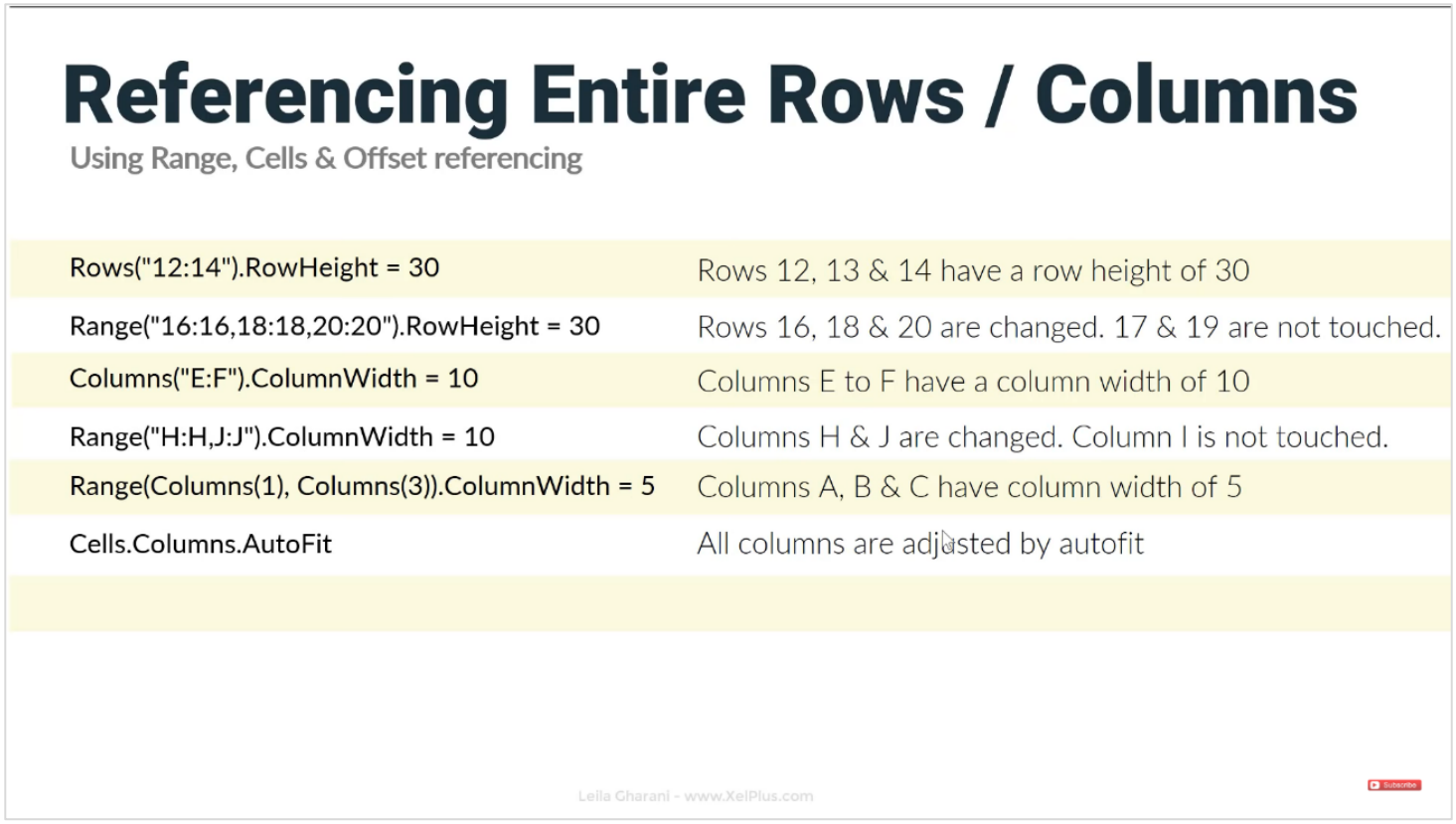
Cópia de Valores

Range("B2").Value = Range("A2").Value



	A	B	C	D	E
4	Company	Business Unit	Actual Revenue	Budget Revenue	Variance
5	Entity A	BU_1	10,200	10,404	-2%
6	Entity B	BU_1	12,240	12,485	-2%
7	Entity C	BU_1	14,688	14,982	-2%
8	Entity D	BU_1	19,776	17,978	10%
9	Entity E	BU_2	10,300	10,506	-2%
10	Entity F	BU_2	12,360	12,607	-2%

Formatação


Referencing Entire Rows / Columns
Using Range, Cells & Offset referencing
Rows("12:14").RowHeight = 30
Rows 12, 13 & 14 have a row height of 30
Range("16:16,18:18,20:20").RowHeight = 30
Rows 16, 18 & 20 are changed. 17 & 19 are not touched.
Columns("E:F").ColumnWidth = 10
Columns E to F have a column width of 10
Range("H:H,J:J").ColumnWidth = 10
Columns H & J are changed. Column I is not touched.
Range(Columns(1), Columns(3)).ColumnWidth = 5
Columns A, B & C have column width of 5
Cells.Columns.AutoFit
All columns are adjusted by autofit
Leila Gharani - www.XelPlus.com

Formatação

- Fonte:
  ActiveCell.Font.Name = "Times New Roman"
- Negrito:
  ActiveCell.Font.Bold = True
- Itálico:
  Range(“A2”).Font.Italic = True
- Sublinhado:
  Range(“A2”).Font.Underline = xlUnderlineStyleSingle
- Cor de fundo:
  ActiveCell().Interior.Color = RGB(0, 255, 0) | Escala Red, Green e Blue
- Alinhamento:
  ActiveCell.HorizontalAlignment = xlCenter | Right, Center, Left
  ActiveCell.VerticalAlignment = xlBottom | Top, Center, Bottom

Condicionais

If *Lógica* Then *Retorno1*

ElseIf *Lógica* Then *Retorno2*

Else *Retorno3*

End If

- Operadores Condicionais e Lógicos:

= | <> | < | > | <= | >=

Not | And | Or | Xor

Condicionais

```
Dim nota As Integer
nota = ActiveCell.Value
Select Case nota
    Case 9 To 10
    ActiveCell(1, 2).Value = “Ótimo”
    Case 7 To 8
    ActiveCell(1, 2).Value = “Bom”
    Case 5 To 6
    ActiveCell(1, 2).Value = “Mediano”
    Case Else
    ActiveCell(1, 2).Value = “Insuficiente”
End Select
```

Loopings

For Next: Execução de código n vezes.

```
Sub AddNumeros()
    Dim Total As Integer
    Dim n As Integer
    Total = 0
    For n = 1 To 10
    Total = Total + n
    Next n
    MsgBox Total
End Sub
```

Loopings

For Each: Execução de código para cada item de uma dada coleção.

```
Sub ProtegerPlanilhas()
    Dim ws As Worksheet
    For Each ws In ActiveWorkbook.Worksheets
    ws.Protect
    Next ws
End Sub
```

Loopings

Do While: Execução de código enquanto condição ser verdadeira.

```
Sub AddPrimeiros10NumerosPositivos()
    Dim i As Integer
    i = 1
    Do While i <= 10
    Resultado = Resultado + i
    i = i + 1
    Loop
    MsgBox Resultado
End Sub
```

Loopings

Do Until: Execução de código até condição ser verdadeira.

```
Sub AddPrimeiros10NumerosPositivos()
    Dim i As Integer
    i = 1
    Do Until i > 10
    Resultado = Resultado + i
    i = i + 1
    Loop
    MsgBox Resultado
End Sub
```

Loopings

While: Execução de código enquanto condição ser verdadeira.

```
Sub AddPrimeiros10NumerosPositivos()
    Dim i As Integer
    i = 1
    While i <= 10
    Resultado = Resultado + i
    i = i + 1
    Wend
    MsgBox Resultado
End Sub
```

Funções Textuais

Range("A2").Value = LCase(meu_texto)	Tudo minúsculo
Range("A2").Value = UCase(meu_texto)	Tudo maiúsculo
Range("A2").Value = Application.WorksheetFunction.Proper(meu_texto)	Capitalizar
LengthFName = Len(meu_texto)	Extensão do texto
FName = Trim(meu_texto)	Remoção de espaços iniciais e finais
FName = Space(5) & Fname	Adicionar espaços
TextoCorreto = Replace(meu_texto, “antigo”, “novo”)	Substituir caracteres
Índice_caracter = InStr(meu_texto, “caracter")	Índice do caracter na string pelo início
Índice_caracter = InStrRev(meu_texto, “caracter")	Índice do caracter na string pelo fim
TextoReverso = StrReverse(meu_texto)	Inverter ordem dos caracteres
Texto_Filtrado = Left(meu_texto, 6)	Seleção de caracteres pela esquerda
Texto_Filtrado = Right(meu_texto, 9)	Seleção de caracteres pela direita
Texto_Filtrado = Mid(meu_texto, início, num_chars)	Seleção de caracteres conforme critério
Texto_Separado = Split(meu_texto, “separador”)	Separação de palavras
Texto_Unido = Join(meu_texto, “separador”)	Agregação de palavras

Exemplo: Alteração múltipla de objeto

```
With ActiveCell.Font
    .Bold = True
    .Color = vbBlue
    .Name = "Arial"
    .Size = 22
    .Italic = True
End With
```

Exemplo: Ativar pop-up

- Simples:

  MsgBox "Hi!" & vbNewLine & "This is a message box"

- Composta por objeto:

  MsgBox “Mensagem” & Excel.Application.UserName, , “Título da caixa”

- Contendo botões para seleção:

  Dim Answer As VbMsgBoxResult

  Answer = MsgBox(“Você está certo disto? ”, vbYesNoCancel + vbQuestion + vbDefaultButton2, “Clear cells”)

  If Answer = vbYes Then

  Range(*RANGE*).Clear

  End If

Exemplo: Editar arquivo

```
Sub EditFile()
    Dim FilePath As String
    FilePath = "C:\Users\User\VBA\myfile.txt“
    Open FilePath For Output As #1
            *CÓDIGO*
    Close #1
End Sub
```

- Modos existentes:
  - Append: adicionar dados a um arquivo já existente
  - Input: ler arquivo
  - Output: escrever dados em arquivo
  - Binary: ler ou escrever dados em arquivo no formato de bytes

Exemplo: Editar arquivo 2

```
Sub EditFile()
    Dim OpenBook As Workbook
    Dim FilePath As String
    FilePath = "C:\Users\User\VBA\myfile.txt“
    Set OpenBook = Application.Workbooks.Open(FilePath)
            *CÓDIGO*
    OpenBook.Close False
End Sub
```

Exemplo: Identificar limites de planilha

```
Sub EndSheet()
    Dim FilePath As String
    Dim CellData As String
    Dim LastCol As Long
    Dim LastRow As Long
    ÚltimaLinhaUsada = ActiveSheet.UsedRange.SpecialCells(xlCellTypeLastCell).Row
    ÚltimaColunaUsada = ActiveSheet.UsedRange.SpecialCells(xlCellTypeLastCell).Column
    DadosCélula = “”
    FilePath = "C:\Users\User\VBA\myfile.csv“
    Open FilePath For Output As #1
        For i=1 To ÚltimaLinhaUsada
            For j=1 To ÚltimaColunaUsada
                    If j= ÚltimaColunaUsada Then
                            DadosCélula = DadosCélula + Trim(ActiveCell(I, j).Value)
                    Else
                            DadosCélula = DadosCélula + Trim(ActiveCell(I, j).Value) + “, ”
                    End If
            Next j
            Write #1, DadosCélula
            DadosCélula = “”
        Next i
    Close #1
    MsgBox(“Finalizado”)

End Sub
```

Exemplo:
Validar existência de pasta

```
Sub Path_Exists()
        Dim Path As String
        Dim Folder As String
Path = “C:\Users\User\Pasta01”
Folder = Dir(Path, vbDirectory)
        If Folder <> “” Then
                MsgBox “Pasta existente”
        Else
                MsgBox “Pasta inexistente”
```

Exemplo:
Validar existência de arquivo

```
Sub File_Exists()
        Dim FileName As String
        FileName = Dir("C:\Users\User\Pasta01\arquivo01.txt")
        If FileName = vbNullString Then
                MsgBox "File doesn't exist"
        Else
                MsgBox "File exists"
```

Exemplo:
Retornar Informações do Arquivo

- Nome do arquivo: Debug.Print ThisWorkbook.Name
- Caminho do arquivo: Debug.Print ThisWorkbook.FullName
- Nome da sheet: Debug.Print ThisWorkbook.Worksheets(1).Name

Exemplo: Agilizar macros

```
Sub YourMacro()
    Application.ScreenUpdating = False
    Application.EnableEvents = False
    Application.Calculation = xlCalculationManual
    *CÓDIGO*
    Application.ScreenUpdating = True
    Application.EnableEvents = True
    Application.Calculation = xlCalculationAutomatic
End Sub
```