Public Function GetMatchingCells(ByRef DataRange As Range, ByVal SearchValue As Variant, \_

Optional PatternMatching As Boolean = False) As Range

'---------------------------------------------------------------------------------------

' Procedure : GetMatchingCells

' Author : Adiv Abramson

' Date : 04/26/2023

' Purpose : Finds all cells in specified range whose value matches SearchValue.

' : Returns a range object containing multiple cell references.

' : Must iterate through each Area of DataRange.

' : Search is case sensitive.

' : (1.1) Now supporting simple patter matching

' :

' Versions : 1.0 - 04/26/2023 - Adiv Abramson

' : 1.1 - 09/09/2024 - Adiv Abramson

' :

'---------------------------------------------------------------------------------------

'Numerics:

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim i As Integer

Dim intArea As Integer

Dim intAreas As Integer

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Ranges:

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim rngCell As Range

Dim rngArea As Range

Dim rngMatchingCells As Range

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Booleans:

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dim blnIsMatchingCell As Boolean

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10 On Error GoTo ErrProc

20 Set GetMatchingCells = Nothing

'========================================================

'Validate inputs: DataRange must consist of two or more

'cells and SearchValue cannot be Null or an empty string.

'========================================================

30 If DataRange Is Nothing Then Exit Function

40 If DataRange.Cells.Count < 2 Then Exit Function

50 If IsNull(SearchValue) Then Exit Function

60 If SearchValue = "" Then Exit Function

'========================================================

'Get count of areas in DataRange in order to look for

'matching cells in each one.

'========================================================

70 intAreas = DataRange.Areas.Count

80 For intArea = 1 To intAreas

90 Set rngArea = DataRange.Areas(intArea)

100 For Each rngCell In rngArea.Cells

110 If PatternMatching Then

120 blnIsMatchingCell = (rngCell.Value Like SearchValue)

130 Else

140 blnIsMatchingCell = (rngCell.Value = SearchValue)

150 End If

160 If blnIsMatchingCell Then

170 If rngMatchingCells Is Nothing Then

180 Set rngMatchingCells = rngCell

190 Else

200 Set rngMatchingCells = Application.Union(rngMatchingCells, rngCell)

210 End If 'rngMatchingCells Is Nothing

220 End If 'blnIsMatchingCell

230 Next rngCell

240 Next intArea

250 Set GetMatchingCells = rngMatchingCells

260 Exit Function

ErrProc:

270 Set GetMatchingCells = Nothing

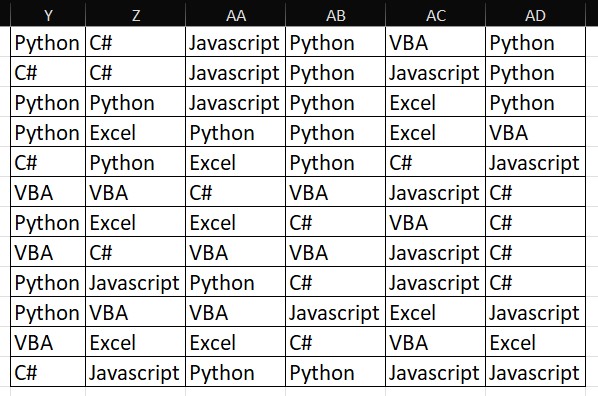
280 MsgBox "Error " & Err.Number & " (" & Err.Description & ") at line " \_

& Erl & " in procedure GetMatchingCells of Module " & MODULE\_NAME

End Function

Usage Examples:

Source Range



Set rngMatchingCells = GetMatchingCells(DataRange:=rngSource, SearchValue:="Python")

'Address: $Y$1,$Y$3:$Z$3,$AB$1:$AB$3,$AD$1:$AD$3,$Y$4,$AA$4:$AB$4,$Z$5,$AB$5,

$Y$7,$AA$9,$Y$9:$Y$10,$AA$12:$AB$12

'#Cells: 20

A table with text on it

Description automatically generated

Set rngMatchingCells = GetMatchingCells(DataRange:=rngSource, SearchValue:="\*[l-t]", PatternMatching:=True)

'Address: $Y$1,$AD$1,$AA$1:$AB$2,$AC$2:$AD$2,$Y$3:$AD$3,$Y$4:$AC$4,

$Z$5:$AB$5,$AD$5,$AC$6,$Y$7:$AA$7,$Y$9:$AA$9,$AC$8:$AC$9,$Y$10,$AB$10:$AD$10,$AD$11,

$Z$11:$AA$12,$AB$12:$AD$12

'#Cells: 44

Set rngMatchingCells = GetMatchingCells(DataRange:=rngSource, SearchValue:="")

'No matching cells

Set rngMatchingCells = GetMatchingCells(DataRange:=rngSource, SearchValue:="\*[a-u][b-r]", PatternMatching:=True)

'Address: $Y$1,$AD$1:$AD$2,$AB$1:$AB$3,$AC$3:$AD$3,$Y$3:$Z$4,$AA$4:$AC$4,$Z$5:$AB$5,$Y$7:$AA$7,

'$AA$9,$Y$9:$Y$10,$AC$10,$Z$11:$AA$11,$AD$11,$AA$12:$AB$12

'#Cells: 30

A table with text on it

Description automatically generated

Here's an example of updating referenced cells, where we add " is really great!" to the cells referenced by GetMatchingCells():

Set rngMatchingCells = GetMatchingCells(DataRange:=rngSource, SearchValue:="Python")

'Add string " is really great!" to matching cells

'Something you can't do through the UI, I think.

For Each rngCell In rngMatchingCells.Cells

rngCell.Value = rngCell.Value & " is really great!"

Next rngCell

rngMatchingCells.Columns.EntireColumn.AutoFit

A screenshot of a computer

Description automatically generated

The function works with a non-contiguous source range as well.

A screen shot of a computer

Description automatically generated

Set rngMatchingCells = GetMatchingCells(DataRange:=Selection, SearchValue:="Python")

'Address: $AO$1,$AY$1,$AO$3:$AO$4,$AT$6,$AY$6,$AQ$7,$AP$8:$AP$9,$AN$11,$AT$13,$BA$13,$AR$17

'#Cells: 13

A screenshot of a computer

Description automatically generated

The function can match numeric values as well.

A table of numbers in a row

Description automatically generated

Set rngMatchingCells = GetMatchingCells(DataRange:=rngSource, SearchValue:=26)

'Address: $BI$3,$BG$5,$BI$5,$BJ$9,$BG$11,$BJ$13

'#Cells: 6

A table with numbers and letters

Description automatically generated