Place the following enum in standard module.

Public Enum SwapOp

Always

VarA\_GT\_VarB

VarA\_GTE\_VarB

VarA\_LT\_VarB

VarA\_LTE\_VarB

VarA\_E\_VarB

VarA\_NE\_VarB

End Enum

Public Sub SwapIf(ByRef VarA As Variant, ByRef VarB As Variant, Optional ByVal SwapCondition As SwapOp = SwapOp.Always)

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

' Procedure : SwapIf

' Author : Adiv Abramson

' Date : 09/28/2024

' Purpose : Mimics Python's variable swapping feature, i.e. given a = 1 and b -2,

' : a,b = b,a results in a = 2 and b = 1

' : Special sauce is optional conditional test.

' : By incororating the conditional test in the procedure, the calling code

' : doesn't have to, thus making it more succinct.

' : Variables are swapped only swap condition is met.

' : For dictionaries, ranges, worksheets, and workbooks always swap.

' : Both variables must have the same TypeName. Neither can be "Nothing"

' : Supported types:

' : \*String

' : \*Integer

' : \*Long

' : \*Single

' : \*Double

' : \*Currency

' : \*Date

' : \*Dictionary

' : \*Range

' : \*Worksheet

' : \*Workbook

' : Conditions are

' : \*Always (default)

' : \*A > B

' : \*A >= B

' : \*A < B

' : \*A <= B

' : \*A <> B

' :

' :

' :

' Versions : 1.0 - 09/28/2024 - Adiv Abramson

' :

' :

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

'Strings:

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

Dim strVarATypeName As String

Dim strVarBTypeName As String

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

'Arrays:

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

Dim arVars As Variant

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

'Variants:

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

Dim vntTypeName As Variant

Dim vntVar As Variant

Dim vntTempVar As Variant

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

'Booleans:

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

Dim blnOKToSwap As Boolean

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

10 On Error GoTo ErrProc

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

'Validate inputs.

'Both must be of the same type.

'Neither can be "Nothing" or "Error"

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

20 arVars = Array(VarA, VarB)

30 vntTypeName = ""

40 For Each vntVar In arVars

50 If vntTypeName = "" Then

60 vntTypeName = TypeName(vntVar)

70 ElseIf TypeName(vntVar) <> vntTypeName Then

80 Debug.Print "'Incompatible types: " & "VarA is " & TypeName(VarA) & ", VarB is " & TypeName(VarB)

90 Exit Sub

100 End If

110 Next vntVar

120 If Not IsInList(vntTypeName, "String", "Integer", "Long", "Single", "Double", "Currency", \_

"Date", "Dictionary", "Range", "Worksheet", "Workbook") Then

130 Debug.Print "'Invalid variable type: " & vntTypeName

140 Exit Sub

150 End If

160 If vntTypeName = "Nothing" Then

170 Debug.Print "'VarA and VarB are Nothing. Cannot swap."

180 Exit Sub

190 End If

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

'For object type variables, ignore SwapType.

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

200 If IsInList(vntTypeName, "Dictionary", "Range", "Worksheet", "Workbook") Then

210 Set vntTempVar = VarA

220 Set VarA = VarB

230 Set VarB = vntTempVar

240 Debug.Print "'Object variables have been swapped."

250 Exit Sub

260 End If 'IsInList()

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

'For all other variable types, check specified condition,

'if any.

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

270 Select Case SwapCondition

Case SwapOp.Always

280 blnOKToSwap = True

290 Case SwapOp.VarA\_E\_VarB

300 blnOKToSwap = (VarA = VarB)

310 Case SwapOp.VarA\_NE\_VarB

320 blnOKToSwap = (VarA <> VarB)

330 Case SwapOp.VarA\_LT\_VarB

340 blnOKToSwap = (VarA < VarB)

350 Case SwapOp.VarA\_LTE\_VarB

360 blnOKToSwap = (VarA <= VarB)

370 Case SwapOp.VarA\_GT\_VarB

380 blnOKToSwap = (VarA > VarB)

390 Case SwapOp.VarA\_GTE\_VarB

400 blnOKToSwap = (VarA >= VarB)

410 End Select

420 If Not blnOKToSwap Then

430 Debug.Print "'VarA and VarB not swapped because swap condition was not satisfied."

440 Exit Sub

450 End If

460 vntTempVar = VarA

470 VarA = VarB

480 VarB = vntTempVar

490 Debug.Print "'VarA is now " & VarA & " and VarB is now " & VarB

500 Exit Sub

ErrProc:

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

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

End Sub

Here is the testing procedure I used to put SwapIf() through its paces.

Public Sub TestSwapIf()

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

' Procedure : TestSwapIf

' Author : Adiv Abramson

' Date : 09/28/2024

' Purpose : Test SwapIf()

' :

' :

' :

' Versions : 1.0 - 09/28/2024 - Adiv Abramson

' :

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

'Strings:

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

Dim strA As String

Dim strB As String

Dim strSwapConditionType As String

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

'Numerics:

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

Dim intA As Integer

Dim intB As Integer

Dim lngA As Long

Dim lngB As Long

Dim sngA As Single

Dim sngB As Single

Dim dblA As Double

Dim dblB As Double

Dim curA As Currency

Dim curB As Currency

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

'Worksheets:

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

Dim wsA As Worksheet

Dim wsB As Worksheet

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

'Workbooks:

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

Dim wbA As Workbook

Dim wbB As Workbook

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

'Ranges:

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

Dim rngA As Range

Dim rngB As Range

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

'Arrays:

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

Dim arA As Variant

Dim arB As Variant

Dim arDynA() As String

Dim arDynB() As Integer

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

'Objects:

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

Dim dictA As Dictionary

Dim dictB As Dictionary

Dim objTableA As ListObject

Dim objTableB As ListObject

Dim dictTestCases As Dictionary

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

'Variants:

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

Dim dteA As Date

Dim dteB As Date

Dim vntTestCaseKey As Variant

Dim vntTestCases As Variant

Dim vntTestCase As Variant

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

'Booleans:

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

Dim blnA As Boolean

Dim blnB As Boolean

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

'Constants

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

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

10 On Error GoTo ErrProc

'SwapIf VarA:=strA, VarB:=strB

'SwapIf VarA:=dictA, VarB:=dictB

'Set dictA = New Dictionary: Set dictB = New Dictionary

'SwapIf VarA:=dictA, VarB:=dictB

'SwapIf VarA:=intA, VarB:=dblB

'SwapIf VarA:=dteA, VarB:=dteB

'Set rngA = ActiveSheet.Range("A57")

'Set rngB = ActiveSheet.Range("A58")

'SwapIf VarA:=rngA, VarB:=rngB

'Set objTableA = GetTable(TableName:="tblPTInfo", DataWorksheet:="Lists")

'Set objTableB = GetTable(TableName:="tblVisibleSheets")

'SwapIf VarA:=objTableA, VarB:=objTableB

'intA = 1: intB = 1

'SwapIf VarA:=intA, VarB:=intB, SwapCondition:=Always

'VarA is now 1 and VarB is now 1

'intA = 10: intB = 25

'SwapIf VarA:=intA, VarB:=intB, SwapCondition:=VarA\_E\_VarB

'VarA and VarB not swapped because swap condition was not satisfied.

'intA = 10: intB = 25

'SwapIf VarA:=intA, VarB:=intB, SwapCondition:=VarA\_GT\_VarB

'VarA and VarB not swapped because swap condition was not satisfied.

'intA = 10: intB = 5

'SwapIf VarA:=intA, VarB:=intB, SwapCondition:=VarA\_GT\_VarB

'VarA is now 5 and VarB is now 10

'intA = 10: intB = 5

'SwapIf VarA:=intA, VarB:=intB, SwapCondition:=VarA\_GTE\_VarB

'VarA is now 5 and VarB is now 10

'intA = 10: intB = 5

'SwapIf VarA:=intA, VarB:=intB, SwapCondition:=VarA\_LT\_VarB

20 Set dictTestCases = New Dictionary

30 With dictTestCases

40 .Item(SwapOp.Always) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

50 .Item(SwapOp.VarA\_E\_VarB) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

60 .Item(SwapOp.VarA\_GT\_VarB) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

70 .Item(SwapOp.VarA\_GTE\_VarB) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

80 .Item(SwapOp.VarA\_LT\_VarB) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

90 .Item(SwapOp.VarA\_LTE\_VarB) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

100 .Item(SwapOp.VarA\_NE\_VarB) = Array(Array(12, 25), \_

Array(25, 12), \_

Array(30, 30), \_

Array(78.9, -0.03), \_

Array(-0.03, 78.9), \_

Array(-0.03, -0.03), \_

Array(#1/1/2023#, #3/2/2023#), \_

Array(#3/2/2023#, #1/1/2023#), \_

Array(#1/1/2023#, #1/1/2023#), \_

Array("VBA", "C#"), \_

Array("C#", "VBA"), \_

Array("VBA", "VBA"), \_

Array("excel", "EXCEL"), \_

Array(CDbl(350), CDbl(897.4)), \_

Array(CDbl(897.4), CDbl(350)), \_

Array("text", 456.8))

110 End With

120 For Each vntTestCaseKey In dictTestCases.Keys

130 vntTestCases = dictTestCases.Item(vntTestCaseKey)

140 If IsEmpty(vntTestCases) Then GoTo NextTestCase

150 If Not IsArray(vntTestCases) Then GoTo NextTestCase

160 strSwapConditionType = Switch(vntTestCaseKey = SwapOp.Always, "Always", \_

vntTestCaseKey = SwapOp.VarA\_E\_VarB, "VarA = VarB", \_

vntTestCaseKey = SwapOp.VarA\_GT\_VarB, "Var A > VarB", \_

vntTestCaseKey = SwapOp.VarA\_GTE\_VarB, "Var A >= VarB", \_

vntTestCaseKey = SwapOp.VarA\_LT\_VarB, "Var A < VarB", \_

vntTestCaseKey = SwapOp.VarA\_LTE\_VarB, "Var A <= VarB", \_

vntTestCaseKey = SwapOp.VarA\_NE\_VarB, "Var A <> VarB")

170 Debug.Print "'Testing swap condition " & strSwapConditionType & ":"

180 For Each vntTestCase In vntTestCases

190 Debug.Print "'VarA = " & vntTestCase(0) & ", VarB = " & vntTestCase(1)

200 SwapIf VarA:=vntTestCase(0), VarB:=vntTestCase(1), SwapCondition:=vntTestCaseKey

210 Debug.Print "'" & String(40, "\*")

220 Next vntTestCase

NextTestCase:

230 Next vntTestCaseKey

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

'Set wbA = Application.Workbooks(3)

'Set wbB = Application.Workbooks(2)

'SwapIf VarA:=wbA, VarB:=wbB, SwapCondition:=Always

'Object variables have been swapped.

'wbA is AdventureWorks Analysis.xlsxand wbB is Application Shell.xlsm

'Object variables have been swapped.

'wbA is Application Shell.xlsmand wbB is AdventureWorks Analysis.xlsx

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

'Set rngA = Application.Workbooks(2).ActiveSheet.UsedRange

'Set rngB = Application.Workbooks(3).ActiveSheet.UsedRange

'SwapIf VarA:=rngA, VarB:=rngB, SwapCondition:=VarA\_GTE\_VarB

'Object variables have been swapped.

'rngA.Address is $D$7:$H$30

'rngB.Address is $A$1:$AT$65

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

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

'What happens if you pass an expression?

'Within the procedure, the local variables are swapped.

'Obiviously, there are no variables here in the calling

'code to which the values in the local variables can be

'passed.

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

'SwapIf VarA:=2 + 3, VarB:=9 \* 3

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

'Set wsA = ActiveWorkbook.Worksheets(1)

'Set wsB = ThisWorkbook.Worksheets(4)

'SwapIf VarA:=wsA, VarB:=wsB, SwapCondition:=VarA\_NE\_VarB

'Object variables have been swapped.

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

'Set dictA = New Dictionary

'Set dictB = New Dictionary

'

'With dictA

' .Item(1) = ""

' .Item(2) = ""

' .Item(3) = ""

'End With

'

'With dictB

' .Item(1) = ""

' .Item(2) = ""

' .Item(3) = ""

' .Item(4) = ""

' .Item(5) = ""

'End With

'

'SwapIf VarA:=dictA, VarB:=dictB, SwapCondition:=VarA\_LT\_VarB

'Object variables have been swapped.

'dictA.Count = 5 and dictB.Count = 3

'Debug.Print "'dictA.Count = " & dictA.Count & " and dictB.Count = " & dictB.Count

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

240 Exit Sub

ErrProc:

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

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

End Sub