**# WRITE FIRST MACRO**

Sub write\_VB\_Macro()

End Sub

**#INSERT MSGBOX**

MsgBox "Hello i m programming knowledge", vbInformation, "greetings"

MsgBox "i am a yotube channel", vbQuestion, "question"

MsgBox "sentence", vbOKCancel, "hello"it will give a msg box whenever we run the macro

Here second parameter i.e vb information, vb question, vb okcancel is a hyperparameter, and do diff things, check individually,

The **Buttons** parameter can take any of the following values −

* 0 vbOKOnly - Displays OK button only.
* 1 vbOKCancel - Displays OK and Cancel buttons.
* 2 vbAbortRetryIgnore - Displays Abort, Retry, and Ignore buttons.
* 3 vbYesNoCancel - Displays Yes, No, and Cancel buttons.
* 4 vbYesNo - Displays Yes and No buttons.
* 5 vbRetryCancel - Displays Retry and Cancel buttons.
* 16 vbCritical - Displays Critical Message icon.
* 32 vbQuestion - Displays Warning Query icon.
* 48 vbExclamation - Displays Warning Message icon.
* 64 vbInformation - Displays Information Message icon.
* 0 vbDefaultButton1 - First button is default.
* 256 vbDefaultButton2 - Second button is default.
* 512 vbDefaultButton3 - Third button is default.
* 768 vbDefaultButton4 - Fourth button is default.
* 0 vbApplicationModal Application modal - The current application will not work until the user responds to the message box.
* 4096 vbSystemModal System modal - All applications will not work until the user responds to the message box.

The above values are logically divided into four groups: The **first group** (0 to 5) indicates the buttons to be displayed in the message box. The **second group** (16, 32, 48, 64) describes the style of the icon to be displayed, the **third group** (0, 256, 512, 768) indicates which button must be the default, and the **fourth group** (0, 4096) determines the modality of the message box.

Return Values

The MsgBox function can return one of the following values which can be used to identify the button the user has clicked in the message box.

* 1 - vbOK - OK was clicked
* 2 - vbCancel - Cancel was clicked
* 3 - vbAbort - Abort was clicked
* 4 - vbRetry - Retry was clicked
* 5 - vbIgnore - Ignore was clicked
* 6 - vbYes - Yes was clicked
* 7 - vbNo - No was clicked

**#PUT VALUE IN ACTIBE CELL OR ANY SPECIFIC CELL**

ActiveCell.Value = "prgramming knowledge*" #### will give programming knowledge to the active cell*

[F6].Value = 25 *### will give 25 to F6 cell specigfic everytime we run macro*

**# PUT VALUE IN RANGE OF CELLS**

[A4:E4].Value = "test"

[A5:A15].Value = "test2"

Range("e5").Value = "sample"

Range("h1:h15").Value = "trekdhbxn"

Cells(6, 5).Value = 500 *### cells([row index],[column index]).value*

**# CONCATINATE VALUE FROM EXISITING CELL**

[b5].Value = "hello"

[b5].Value = [b5].Value + " world! "

**# COPY**

[b5].Copy

[g10].PasteSpecial

**# TO GET RID OF DANCING ARROWS WHICH APPER WHEN WE COPY PASTE**

Application.CutCopyMode = False

**#COPY RANGE OF CELLS**

[a5:a10].Value = "hello"

[e7:e12].Value = [a5:a10].Value

[a5:a10].Copy

[e7:e12].PasteSpecial

**#CHANGE FOnts**

Range("e7:e12").font.Name = "Agency FB"

Range("e7:e12").font.Bold = True

Range("e7:e12").font.Size = 25

Range("e7:e12").font.Underline = True

Range("e7:e12").font.Strikethrough = True

Range("e7:e12").font.Subscript = True

Range("e7:e12").font.TintAndShade = False

Range("e7:e12").font.Superscript = True

Range("e7:e12").font.Color = vbRed *### everycolor we need to prefix vb, vbgreen,vbblue etc*

**# CHANGE COLOUR OF FONT AND BACKGROUND**

Range("e7:e12").font.ColorIndex = 22 ## (1 to 55) any number for different colors

Range("e7:e12").font.Color = vbBlack

**#CHANGE COLOUR OF BCKGORUND**

Range("e7:e12").Interior.color = vbblue

**#GRADIENT**

**R**ange("e7:e12").Interior.Pattern = xlPatternLinearGradient ### check beautiful function

**# WITH BLOCK**

With Range("e7:e12").font

.Name = "ariel"

.Bold = True

.Size = 50

.Color = vbGreen

End With

**#PASTESPECIAL**

Range("a2:a10").Value = "programming"

Range("a2:a10").Font.Color = vbBlue

Range("a2:a10").Font.Size = 25

Range("a2:a10").Font.Bold = True

Range("a2:a10").Copy

Range("e2:e10").PasteSpecial xlPasteValues

Range("e2:e10").PasteSpecial xlPasteFormats

Range("j2:j10").PasteSpecial xlPasteColumnWidths

**#FORMATTING ORIENTATION**

Range("a12:e25").Value = "excel"

[a17].Orientation = 90

[a18].Orientation = -90

[a18].WrapText = True

Range(**“**a12:e25"). Orientation = 90

**#MERGE,UNMERGE**

‘row wise merging

Range("a12:e12").Cells.Merge

Range("a13:e13").Cells.MergeCells = True

Range("a13:e13").Cells.UnMerge

'column wise merging

Range("a14:a25").MergeCells = True

Range("b14:b25").Merge

Range("a14:a25").UnMerge

**# ALIGNMENT**

Range("a12:e25").Value = "excel"

Range("a12:e25").HorizontalAlignment = xlCenter

Range("a12:e25").HorizontalAlignment = xlRight

Range("a12:e25").HorizontalAlignment = xlLeft

‘vertical alignment

Range("a12:e25").VerticalAlignment = xlTop

Range("a12:e25").VerticalAlignment = xlCenter

Range("a12:e25").VerticalAlignment = xlBottom

**#BORDERS**

[h17].Value = "new"

Range("h17").Borders.Color = vbBlack

Range("h17").Borders.LineStyle = xlContinuous

Range("h17").Borders.LineStyle = xlDot

Range("h17").Borders.LineStyle = xlDash

Range("h17").Borders.LineStyle = xlNone

Range("h17").Borders.LineStyle = xlDouble

Range("h17").Borders.Weight = 1

Range("h17").Borders.Weight = 2

Range("h17").Borders.Weight = 3

Range("h17").Borders.Weight = 4

**#CLEAR CELLS FEATURE**

Range("a12:e12").Clear

Range("a13:e13").ClearComments

Range("a13:e13").ClearContents

Range("a13:e13").ClearFormats

Range("a14:e14").ClearNotes

Range("a15:e15").ClearOutline

**#DELETING CONCENTS**

'diff in delete and clear content is that after deleting, every data remaing is moved upwards column wise and left row wise

'delete a cell content

Range("g16").Delete

'delete entire column

Range("g16").EntireColumn.Delete

'delete entrie row

Range("g16").EntireRow.Delete

**#DELETE WORKSHEET**

Worksheets("Sheet2").Delete

**#INSERT ROW, COLUMN, CELL**

Range("g16").EntireColumn.Insert

Range("g16").Insert

Range("g16").EntireRow.Insert

**#MANIPULATE COLUMN WIDTH**

'range is 0 to 255

Range("g16").ColumnWidth = 1

Range("g16").Columns.ColumnWidth = 12

**#AUTOFIT WIDTH**

Range("g16").Columns.AutoFit

**#SELECT CELL OR RANGE OF CELLS**

Range("g16:g70").Select

Range("g16").Activate

**#HISE AND UNHIDE ROW AND COLUMNS**

Range("d16").Rows.Hidden = True

Range("d16").Rows.Hidden = False

Range("d16").Columns.Hidden = True

Range("d16").Columns.Hidden = False

#**INPUT VALUES IN DIFF SHEETS WHILE IN OTHER SHEETS**

Range("a1:a10").Value = "hello"

Worksheets(3).Range("a1:a10").Value = "hello"

**#ADD NEW WORKSHEET**

Worksheets.Add

**#ADD SHEET AFTER ANY SPECIFIC SHEET**

Worksheets.Add after: = Worksheets("sheet3")

**#ADD SHEET BEFORE ANU SHEET**

Worksheets.Add before: = Worksheets("sheet3")

**#RENAME A SHEETS**

worksheets(3).name = "rename sheet"

**#COPY A SHEET BEFORE OR AFTER ANY SHEET**

Worksheets("sheet3").Copy before:=Worksheets("sheet10")

Worksheets("sheet3").Copy after:=Worksheets("sheet10")

**#MOVE SHEETS BEFORE OR AFTER**

Worksheets(1).Move after:=Worksheets(3)

Worksheets(1).Move before:=Worksheets(3)

**#CHANGE COLOUR OF SHEETS TAB**

Worksheets(2).Tab.Color = vbBlue

Worksheets(2).Tab.Colorindex = 20

**#HIDE AND UNHIDE SHEETS**

Worksheets(2).Visible = False

Worksheets(2).Visible = True

**#ACTIBATE ANY SPECIFIC SHEETS**

Worksheets(3).Activate

**#PROTECT AND UNPROTECT SHEETS USIBG PASSWORD**

Worksheets("Sheet10").Protect Password:="maths293"

Worksheets("sheet10").Unprotect Password:="maths**293"**

**#ADD A NEW WORKBOOK AND SAVEAS TO SPECIFIC LOCATION**

Workbooks.Add.SaveAs Filename:="C:\Users\Admin\OneDrive\Documents\R\newworkbook.xlsx"

**#OPEN A WORKBOOK**

Workbooks.Open Filename:="C:\Users\Admin\OneDrive\Documents\R\newworkbook.xlsx"

**#CREATE A FOLDER USING EXCERL VBA**

MkDir "C:\Users\Admin\OneDrive\Documents\R\xlcreated"

**#MAKING VARIABLE AND USING IT**

v = "programming"

[a1].Value = v

[f3].Value = v

v = 5

[a1].Value = v \* 2 + (v \* v)

a = "youtube"

b = "channel"

c = " "

[a1].Value = a & c & b

**#DIM FUNCTION**

'dim is used to declare variables before it is used,it just gives glimse to reader of code that what n all variables are used

Dim v

v = 5

[c3] = v

**#SIMPLE DIM FUNCTION WILL NOT GIVE AN ERROR IF OTHER VARIABLES ARE ALSO DECLARED IN DIM, BUT WE WANT AN ERROR IF EXTRA VARIABLES ARE USED**

Option Explicit

Sub worksbook()

'dim is used to declare variables before it is used,it just gives glimse to reader of code that what n all variables are used

Dim v

v = 5

[c3] = v

End Sub

**#STORE A VARIABLE AS SPECIFIC TYPE**

Dim v As Integer

v = 5

[c6] = v

**‘DIFF FROM ABOVE IS IN THIS A WILL BE STRING BY DEFAULT AND V WILL BE INTEGER,**

**‘ONLY LAST VARIABLE IS ASSIGNED AS INTEGER**

Dim a, v As Integer

v = 5

[c6] = v

a = "hello"

[g6] = a

**#SIMPLE FOR LOOP**

Option Explicit

Sub worksbook()

Dim i As Integer

For i = 1 To 5

Cells(i, 1) = i

Next

End Sub

**#FOR LOOP FOR SHOWING ALL COLORINDEX UPTO 56**

Option Explicit

Sub worksbook()

Dim i As Integer

For i = 1 To 56 Step 2

Cells(i, 4) = i

Cells(i, 5).Interior.ColorIndex = i

Next

End Sub

**#FOR I = 1 TO LAST SHEET WE HAD**

Option Explicit

Sub worksbook()

Dim x As Integer

For x = 1 To **ThisWorkbook.Sheets.Count**

Cells(x, x) = "love"

Next

End Sub

**#FOR EACH LOOP**

‘for each loop is same as for loop, it just don’t need starting and ending value

For Each x In ThisWorkbook.Sheets

MsgBox x.Name

**#DO WHILE LOOP, JUST ADD 50 TO ALL VALUES IN A COLUMN UNTILL THERE ARE VALUES IN A**

i = 1

Do While Cells(i, 1).Value **<> "" ## (<>””) is translated as untill value of cells(I,1) is null**

Cells(i, 2).Value = Cells(i, 1) + 50

i = i + 1

Loop

**#DO UNTIL LOOP**

i = 1

Do Until **i > 5 ### here condition must be termination, means it must not be > = 5**

Cells(i, 1).Value = i

i = i + 1

Loop

**#ERRORS**

1. Runtime error
2. Compilation error
3. Syntax error

**#ERROR HANDLING**

On Error Resume Next

[a10] = 5 + 5

[a11] = 5 + 8

[a12] = 5 / 0

[a13] = 5 + 9

On Error GoTo lbl:

Range("b1").Interior.ColorIndex = 50

Range("b2").Interior.ColorIndex = 59  **### there there will be error and code will go to lbl that is a msgbox that error occured**

Range("b3").Interior.ColorIndex = 45

lbl:

MsgBox ("error occured")

**#NOW IF WE DON’T HAVE ANY ERROR, LBL WILL STILL RUN AS IT IS APART TO CODE, BUT WE DON’T WANT IT NOW,**

On Error GoTo lbl:

Range("b1").Interior.ColorIndex = 50

Range("b2").Interior.ColorIndex = 52 ## here value is valid, so no error will pop up

Range("b3").Interior.ColorIndex = 45

done:

Exit Sub

lbl:

MsgBox ("error occured")

**#IF ELSE LOOP WITH FOR**

For i = 2 To 11

If Cells(i, 1).Value < 30 Then

Cells(i, 2).Value = "fail"

Else

If Cells(i, 1).Value < 50 Then

Cells(i, 2).Value = "3 div"

Else

If Cells(i, 1).Value < 70 Then

Cells(i, 2).Value = "2 div"

Else

If Cells(i, 1).Value < 90 Then

Cells(i, 2).Value = "1 div"

Else

If Cells(i, 1).Value > 90 Then

Cells(i, 2).Value = "incredible"

Else

Cells(i, 2).Value = 0

End If

End If

End If

End If

End If

Next

a = 0

For i = 2 To 11

If Cells(i, 2).Value = "3 div" Then

a = a + 1

End If

Next

[b18].Value = a

**#SAME CODE WITH AND OPERATOR**

For i = 2 To 11

If Cells(i, 1).Value >= 0 And Cells(i, 1).Value < 30 Then

Cells(i, 2).Value = "fail"

Else

If Cells(i, 1).Value >= 30 And Cells(i, 1).Value < 50 Then

Cells(i, 2).Value = "3 div"

Else

If Cells(i, 1).Value >= 50 And Cells(i, 1).Value < 70 Then

Cells(i, 2).Value = "2 div"

Else

If Cells(i, 1).Value >= 70 And Cells(i, 1).Value < 90 Then

Cells(i, 2).Value = "1 div"

Else

If Cells(i, 1).Value >= 90 Then

Cells(i, 2).Value = "incredible"

Else

If Cells(i, 1).Value > 100 Then

MsgBox ("invalid input")

End If

End If

End If

End If

End If

End If

Next

**#SELECT CASE USING A PROGRAM OF DISPLAYING MONTH NAME AFTER INPUT BY USER**

Dim v

[b3].Value = "month"

[c3].Value = "month name"

v = Range("b4").Value

Select Case v

Case 1: Range("c4").Value = "jan"

Case 2: Range("c4").Value = "feb"

Case 3: Range("c4").Value = "mar"

Case 4: Range("c4").Value = "apr"

Case 5: Range("c4").Value = "may"

Case 6: Range("c4").Value = "jun"

Case 7: Range("c4").Value = "jul"

Case 8: Range("c4").Value = "aug"

Case 9: Range("c4").Value = "sep"

Case 10: Range("c4").Value = "oct"

Case 11: Range("c4").Value = "nov"

Case 12: Range("c4").Value = "dec"

Case Else: Range("c4").Value = "invalid"

End Select

**#INPUT VALUE FROM USER USING INPUT BOX**

Dim v

[b3].Value = "month"

[c3].Value = "month name"

v = InputBox("enter number", "month", 1, 1500, 1500) ### [(input),[title],[deafult value],[xcord],[ycord])

[b4] = v

Select Case v

Case 1: Range("c4").Value = "jan"

Case 2: Range("c4").Value = "feb"

Case 3: Range("c4").Value = "mar"

Case 4: Range("c4").Value = "apr"

Case 5: Range("c4").Value = "may"

Case 6: Range("c4").Value = "jun"

Case 7: Range("c4").Value = "jul"

Case 8: Range("c4").Value = "aug"

Case 9: Range("c4").Value = "sep"

Case 10: Range("c4").Value = "oct"

Case 11: Range("c4").Value = "nov"

Case 12: Range("c4").Value = "dec"

Case Else: Range("c4").Value = "invalid"

End Select

**#UCASE,LCASE,LEFT,RIGHT FUNCTIONS**

Dim x As Integer

For x = 3 To 9

Cells(x, 3).Value = UCase(Cells(x, 2).Value)

Cells(x, 4).Value = LCase(Cells(x, 2).Value)

Cells(x, 5).Value = Left(Cells(x, 2).Value, 3)

Cells(x, 6).Value = Right(Cells(x, 2).Value, 3)

Next

**#CURRENT DATE, SPECIFIC DATE, ADD DATE, ADD MONTH, ADD DAY, ADD HOURS, ADD MONUTES, ASS SECONDS**

b = Date

MsgBox b

c = CDate("31 dec 2020")

MsgBox c

MsgBox DateAdd("yyyy", 5, c)

MsgBox DateAdd("m", 3, c)

MsgBox DateAdd("d", 2, c)

MsgBox DateAdd("h", 5, "31 dec 2020 12:00:00")

MsgBox DateAdd("n", 5, "31 dec 2020 12:00:00")

MsgBox DateAdd("s", 5, "31 dec 2020 12:00:00")

**#DATE PART FUNCTION**

c = #12/31/2020#

MsgBox (DatePart("yyyy", c))

MsgBox (DatePart("m", c))

MsgBox (DatePart("d", c))

MsgBox (DatePart("q", c))

**#DAY, MONTH , YEAR FUNCTION**

MsgBox (Day(c))

MsgBox (Month(c))

MsgBox (Year(c))

**#DISPLAY CURRENT DATE AND TIME**

MsgBox Now()

MsgBox Time()

**#DEFINE MY OWN FUNCTION**

Sub func()

Function add(x As Integer, y As Integer) As Integer

add = x + y

End Function

**#CREATING AN ARRAY**

Dim a(3) As Integer

For i = 0 To 3

a(i) = InputBox("enter value")

Next

For i = 0 To 3

MsgBox a(i)

Next

**#2D ARRRAY**

Dim a(2, 2) As Integer

For i = 0 To 2

For j = 0 To 2

a(i, j) = (j + i) \* 10

Next

Next

For i = 0 To 2

For j = 0 To 2

MsgBox (a(i, j))

Next

Next

**# AUTOFILL**

Range("e5").Select

ActiveCell.FormulaR1C1 = "=rc[-2]\*rc[-1]"

Range("e5").Select

Selection.autofill Destination:=Range("e5:e8"), Type:=xlFillDefault

‘R c [-2] = 2 columns left

‘rc[-1] = 1 column left

‘Rc[1] = 1 colum right

‘R[-1]c = one row up

‘R[1]c[1] = one row down, one column right