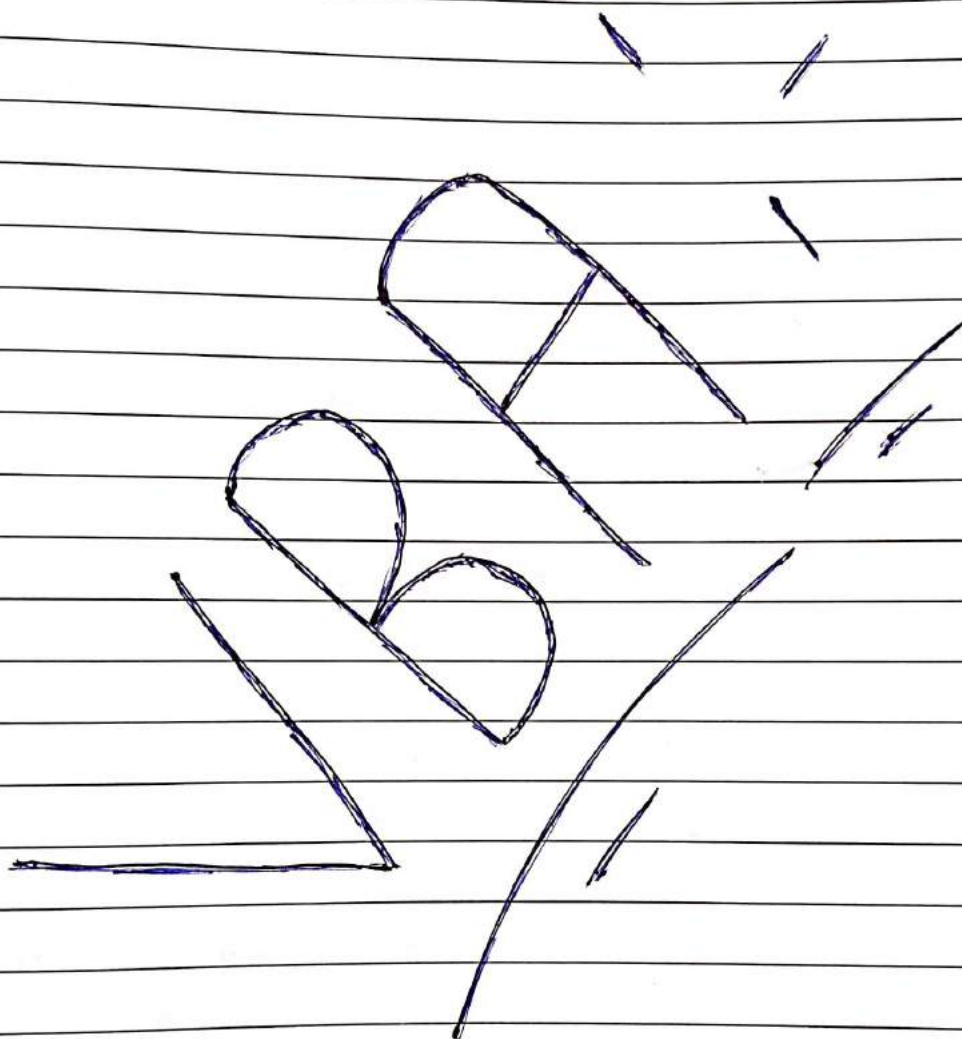


Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page _____

Sub: _____ Lesson No. _____ Date _____



VBA - Visual basic for Applications

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 1-

Sub: _____ Lesson No. _____ Date _____

Introduction to VBA :-

VBA (Visual Basic for Applications) is a language related to Visual basic that can only run through a host application (Excel, in our case).

Using VBA, we can do almost anything imaginable with Excel...

But before we get started, let's begin by making sure that the tools we need are visible.

If you are using Excel version 2007 (or a higher version)

Click on file → options → Customize the Ribbon and then check "Developer".

To work with VBA code, we will need an Editor, which is installed by default. You can open it by pressing the shortcut key combination "ALT F11".

VBA Variables! Variables are specific values that are stored in a computer memory or storage system. Later you can use that value in code and execute.

To name the variables in VBA, you need to follow the following rules.

- It must be less than 255 characters
- No spacing is allowed
- It must not begin with a number
- Period is not permitted

Syntax-

```
Sub Exercise ()  
    Dim < Name >  
End Sub
```

Variables are some types -

- Integer
- String
- Double
- Boolean

- Integer :- Integer variables are used to store whole numbers.

Create a Commandbutton.

```
Dim x As Integer  
x = 6  
Range ("A1"). Value = x
```

we write the value of x to cell A1.

NATIONAL SKILL TRAINING INSTITUTE

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 2.

Sub: _____ Lesson No. _____ Date _____

- String :- String variables are used to store text.

Create a Commandbutton.

```
Dim book As String
```

```
book = "bible"
```

```
Range("A1").value = book
```

finally we write the text of the variables book to cell A1.

- Double :- A variable of type double is more accurate than a variable of type integer and can also store numbers after the comma.

```
Dim x As Integer
```

```
x = 5.5
```

```
MsgBox "value is " & x
```

O/P-

Microsoft Excel
value is 6
OK

But that is not the right value! we initialized the variable with the value 5.5 and we get the value 6.

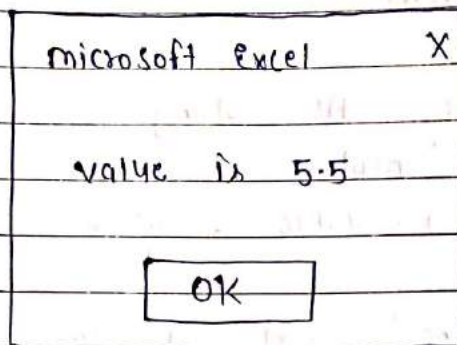
what we need is a variable of type double.


```
Dim x As Double
```

```
x = 5.5
```

```
MsgBox "value is " & x
```

o/p-



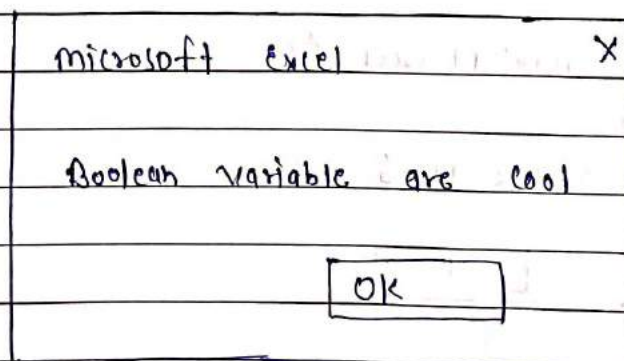
- Boolean :- use a Boolean variables to hold the value true and false.

```
Dim Continue As Boolean
```

```
Continue = True
```

```
if Continue = True then MsgBox "Boolean variable are cool"
```

o/p-



we initialize Continue with the value true. finally then Result

NATIONAL SKILL TRAINING INSTITUTE

Sion, Mumbai - 400 022.

Trade: _____

Unit No. _____

Page 3

Sub: _____

Lesson No. _____

Date _____

Constant in VBA :- Constant is like a variable, but you cannot modify it. To declare VBA constant, you can use keyword Const.

There are two types of constant.

- Built-in or Intrinsic provided by the Application
- Symbolic or user defined

Summary:-

- Variables are specific values that are stored in a computer memory or storage system.
- you can use VBA dim types keyword in syntax to declare variable explicitly
- VBA datatypes can be segregated into two types.
 - Numeric data type
 - Non-Numeric data type
- In VBA, if the datatype is not specified. It will automatically declare the variable as a variant.
- Constant is like a variable, but you cannot modify it. To declare a constant in VBA you can use keyword Const.

Ex-

Private Sub CommandButton1_Click()

Dim yourname As String,

Joiningday As Date,

Income As Currency

your Name = "kuldeep kumar"

Joiningday = "1 April 2016"

Income = 10000

Range ("A1") = yourname

Range ("A2") = Joiningday

Range ("A3") = Income

End Sub

o/p-

	A	B
1	kuldeep kumar	
2	4/1/2016	
3	\$10,000.00	
4		

NATIONAL SKILL TRAINING INSTITUTE

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page _____

Sub: _____ Lesson No. _____ Date _____

Logical Operators :- The three most used logical operators in excel VBA are -

- Logical operator And
- Logical operator or
- Logical operator Not

• Logical operator And :- Place a Commandbutton on your worksheet and add the following code.

```
Dim Score1 As Integer, Score2 As Integer, Result As String
```

```
Score1 = Range ("A1").value
```

```
Score2 = Range ("B1").value
```

```
if Score1 >= 60 And Score2 > 1 then
```

```
Result = "Pass"
```

```
Else
```

```
Result = "fail"
```

```
End if
```

```
Range ("C1").value = Result
```

• Logical operator or :- Place a Commandbutton on your worksheet and add the following code like-

```
Dim Score1 As Integer, Score2 As Integer, Result As String
```

```
Score1 = Range ("A1").Value
```



```
Score2 = Range ("B1").Value
```

```
If Score1 >= 60 or Score2 > 1 then
```

```
    Result = "Pass"
```

```
Else
```

```
    Result = "fail"
```

```
End if
```

```
Range ("C1").Value = Result
```

• Logical operator Not :- Place a Commandbutton on your worksheet and add the following code lines.

```
Dim Score1 As Integer, Score2 As Integer, Result As String
```

```
Score1 = Range ("A1").Value
```

```
Score2 = Range ("B1").Value
```

```
If Score1 >= 60 And Not Score2 = 1 then
```

```
    Result = "Pass"
```

```
Else
```

```
    Result = "fail"
```

```
End if
```

```
Range ("C1").Value = Result
```

Explanation - If Score1 is greater than or equal to 60 and Score2 is not equal to 1, Excel VBA returns Pass, Else Excel VBA returns fail.

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 5

Sub: _____ Lesson No. _____ Date _____

Arithmetic Expression in VBA:-

Following Arithmetic operators are supported by VBA.

Assume variables A holds 5 and variable B hold 10, then

Operator	Description	Example
+	Adds the two operands	$A+B$ will give 15
-	Subtracts the second operand from the first	$A-B$ will give -5
*	multiplies both the operands	$A*B$ will give 50
/	Divides the Numerator by the denominator	B/A will give 2
%	modules operator and the Remainder After an integer division	$B\%A$ will give 0
^	Exponentiation Operator	B^A will give 100000

Ex- Private Sub Constant_demo_Click()

Dim a As Integer

a = 5

Dim b As Integer

b = 10

Dim c As Double

c = a + b

MsgBox (" Addition Result is " & c)

c = a - b

MsgBox (" Substraction Result is " & c)

c = a * b

MsgBox (" Multiplication Result is " & c)

c = b / a

MsgBox (" Division Result is " & c)

c = b mod a

MsgBox (" modules Result is " & c)

c = b ^ a

Msg (" Exponentiation Result is " & c)

End Sub.

O/P - Addition Result is 15

Substraction Result is -5

Multiplication Result is 50

Division Result is 2

modules Result is 0

Exponentiation Result is 100000

Trade: _____ Sion, Mumbai - 400 022.
Unit No. _____ Page 6
Sub: _____ Lesson No. _____ Date _____

Introduction to String in VBA :-

Strings are a sequence of characters, which can consist of either Alphabets, Numbers, special characters, or all them.

A variable is said to be a string if it is enclosed within double quotes " ".

Syntax -

VariableName = "String"

String function :-

Instr () :- Returns the first occurrence of the specified substring. Search happens from the left to the right.

InstrRev () :- Returns the first occurrence of the specified substring. Search happens from the right to the left.

Lcase () :- Returns the lower case of the specified string.

Ucase () :- Returns the upper case of the specified string.

Left () :- Returns a specific number of characters from the left side of the string.

Right () :- Returns a specific Number of characters from the Right Side of the String.

End () :- Returns a specific Numbers of character from a String based of the Specified Parameters.

Ltrim () :- Returns a string After Removing the spaces on the left side of the Specified strings.

Rtrim () :- Return a string After Removing the spaces on the Right side of Specified the Specified string.

Trim () :- Returns a string value After Removing both the leading and the trailing blank spaces.

Len () :- Returns the length of the given String.

Replace () :- Returns a string After Replacing a string with another string.

space () :- fill a string with the specified Number of spaces.

strcmp () :- Returns an Integer value After comparing the two Specified strings.

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 7

Sub: _____ Lesson No. _____ Date _____

String () :- Returns a string with a specified character for specified number of times.

StrReverse () :- Returns a string after reversing the sequence of the characters of the given string.

Introduction to Arrays :-

An Array is a group of variables. In Excel VBA, you can refer to a specific variables (element) of an array by using the array name and the index number.

One - dimensional Array :-

To create a one - dimensional Array, execute the following steps.

Place a Command button on your worksheet. and Add the following code.

```
Dim films (1 To 5) As String  
films (1) = "Lord of the Rings"  
films (2) = "Speed"  
films (3) = "Star wars"  
films (4) = "The Godfather"  
films (5) = "Pulp fiction"
```


Result whe you click the Command button on the Sheet.

o/p -	Microsoft Excel *
	The Godfather
	OK

Two-dimensional Array - !

This time we are going to read the names from the sheet and create a Command button on your worksheet.

	A	B	C
1	Lord of the Rings	Adventure	Command button!
2	Speed	Action	
3	Star wars	Sci-fi	
4	The God father	Crime	

```

Dim films (1 To 5, 1 To 2) As String
Dim i As Integer, j As Integer
for i = 1 To 5
    for j = 1 To 2
        films (i, j) = cells (i, j) .. value
    next j
next i

```

msgBox films (4, 2)	o/p -	Microsoft Excel X
		Crime
		OK

Trade: _____

Sion, Mumbai - 400 022.

Unit No. _____

Page _____

8

Sub: _____

Lesson No. _____

Date _____

If then statement :-

Place a Commandbutton on your worksheet and add the following code lines.

```
Dim Score As Integer, result As String
```

```
Score = Range ("A1").value
if Score >= 60 Then result = "Pass"
```

```
Range ("B1").value = result
```

Result when you click the Command button on the sheet.

o/p-		A	B	C	D
	1	80	Pass		
	2			Commandbutton1	
	3				
	4				

Explanation - if Score is greater than or Equal to 60, Excel VBA Return Pass.

Note - if Score is less than 60, Excel VBA Places the value of the Empty Variable Result into Cell B1.

Else Statement :-

Place a Command button on your worksheet and Add the following Code lines.

Dim Score As Integer, Result As String

Score = Range ("A1"). Value

if Score >= 60 Then

result = "Pass"

Else

result = "fail"

End if

Range ("B1"). Value = result

Result when you click the Command button on the Sheet.

o/p -	A	B	C
1	40	fail	
2			Commandbutton1
3			
4			
5			

Note - Only if you have one code line after Then and NO Else Statement, it is Allowed to Place a code line directly After Then and to Omit (leave out) End if (first example).

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 9

Sub: _____ Lesson No. _____ Date _____

Select Case in VBA :- instead of multiple if then statements in excel VBA.

Place a Command button on your worksheet and Add the following code lines.

Dim Score As Integer, result As string

Score = Range ("A1"). Value

Select Case score

Case Is >= 80

Result = "very good"

Case Is >= 70

Result = "good"

Case Is >= 60

Result = "sufficient"

Case Else

Result = "Insufficient"

End Select.

Range ("B1"). Value = Result.

O/P -		A	B	C	D
1		70	good		Command button1
2					
3					
4					

Note - Excel VBA executes the code under the Select Case.

Mod Operator :-

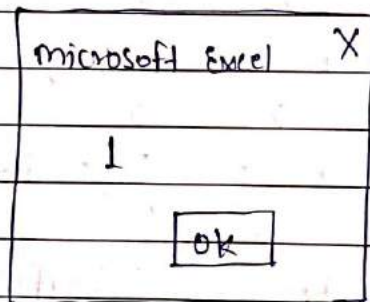
The mod operator in Excel VBA gives the Remainder of a division.

Place a Command button on your worksheet and Add the following Code lines.

Code line - `MsgBox 7 Mod 2`

Result when you click the Command button on the sheet.

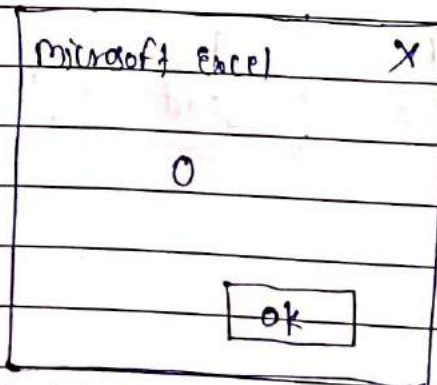
O/P -



Code line -

`MsgBox 8 Mod 2`

O/P -



Expla - 8 divided by 2 Equals 4 with a Remainder of 0.

NATIONAL SKILL TRAINING INSTITUTE

Trade: _____ Sion, Mumbai - 400 022.

Unit No. _____ Page 10

Sub: _____ Lesson No. _____ Date _____

Introduction of Loop in VBA :-

Looping is one of the most powerful programming techniques. A loop in Excel VBA enables you to loop through a range of cells with just a few code lines.

Single loop :- you can use a single loop to loop through a one-dimensional range of cells.

Place a command button on your worksheet and add the following code lines.

```
Dim i As Integer
    For i = 1 To 6
        Cells(i, 1).Value = 100
    Next i
```

Result when you click the command button on the sheet.

O/P -	A	B	C	D	E
1	100				
2	100		Commandbutton1		
3	100				
4	100				
5	100				
6	100				

Note. - - - - -

Double loop :- you can use a double loop through a two - dimensional range of cells.

Place a Command button on your worksheet and add the following code.

```
Dim i As Integer, j As Integer
for i = 1 To 5
    for j = 1 To 2
        Cells (i, j) . Value = 100
    Next j
Next i
```

Result when you click the Command button on the sheet.

o/p-

	A	B	C	D
1	100	100		
2	100	100		
3	100	100	Commandbutton1	
4	100	100		
5	100	100		

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 11

Sub: _____ Lesson No. _____ Date _____

Triple loop :- you can use a triple loop to loop through two-dimensional ranges on multiple Excel worksheet.

Place a Command button on your worksheet and add the following codes.

```
Dim c As Integer, i As Integer, j As Integer
```

```
    For c = 1 To 3
```

```
        For i = 1 To 6
```

```
            For j = 1 To 2
```

```
                Worksheets(c).Cells(i,j).Value = 100
```

```
            Next j
```

```
        Next i
```

```
    Next c
```

Do while loop :- Besides the for Next loop, there are other loops in Excel VBA. For example the do while loop. Code placed between do while and loop between will be repeated as long as the part after do while is true.

Place a Command button on your worksheet and add the following code lines.

```
Dim i As Integer
```

```
i = 1
```

```
do while
```

```
https://t.me/UF3XBQkuncyHu09C
```


Cells (i, 1). Value = 20

i = i + 1

Loop

O/P-

	A	B	C	D
1	20			
2	20			
3	20		Commandbutton1	
4	20			
5	20			

Enter Some numbers in Column A. (6 Value)

Place a Command button on your worksheets and

Add the following Code lines.

Dim i As Integer

i = 1

Do While Cells (i, 1). Value <> ""

Cells (i, 2). Value = Cells (i, 1). Value + 10

i = i + 1

Loop

Result when you click the Command button on your sheet.

O/P-	A	B	C	D
1	27	37		
2	35	45		
3	44	54	Commandbutton1	
4	66	76		
5	70	80		
6	81	91		

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 12

Sub: _____ Lesson No. _____ Date _____

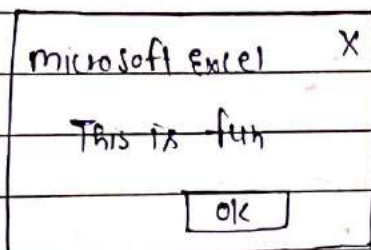
MsgBox in VBA :- The MsgBox is a dialog box in Excel VBA you can use to inform the user of your programme.

Place a Command button on your worksheet and Add the following code.

1. A Simple message-

MsgBox " This is fun "

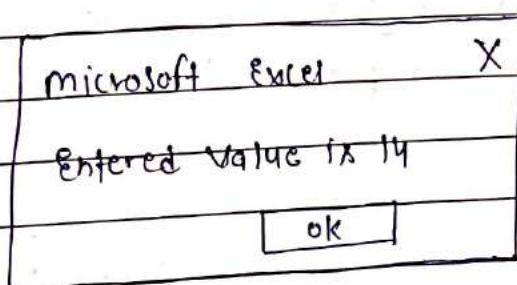
O/P -



2- A little more Advanced . message. first Enter a Number into Cell A1. (Suppose - 14)

MsgBox " Entered value is " & Range ("A1") .value

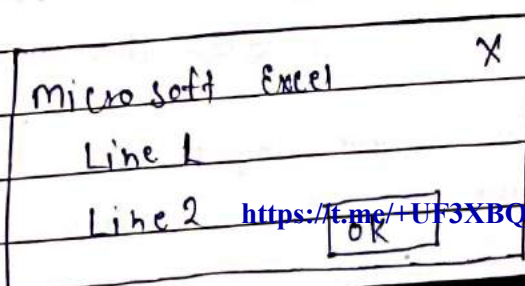
O/P -



3- To start a new line in a message, use vbNewLine.

MsgBox " Line 1 " & vbNewline & " Line 2 "

O/P -



MsgBox function :-

Place a Command button on your worksheet and add the following code.

```
Dim Answer As Integer
```

```
Answer = MsgBox ("Are you Sure you want  
to Empty the sheet?", vbYesNo + vbQuestion, "Empty sheet")
```

```
if Answer = vbYes Then
```

```
Cells.ClearContents
```

```
Else
```

```
do nothing
```

```
End if
```

O/P-

Empty sheet

X



Are you sure you want to
Empty the sheet?

OK

InputBox function :-

```
Dim myValue As Variant
```

```
myValue = InputBox ("Give me some input type")
```

O/P-

Microsoft Excel

X

Give me some input type

OK

Cancel

1

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

```
myValue = InputBox ("Give me some input", "Hi", 1)
```

O/P-

Hi

X

Give me some input

OK

Cancel

1