



A Formula Version with VBA

EXCEL VBA FORMULAS FOR SPREADSHEET

© sharath2630@gmail.com, 2016. All rights reserved.

Edition 1

Contents

Introduction

Exercise 1: IIF Function	2
Exercise 2: Custom Address Function.....	3
Exercise 3: Standard IF Function.....	5
Exercise 4: Specific IF Function.....	7
Exercise 5: IF,AND Function	9
Exercise 6: Multi OR Function.....	11
Exercise 7: Simple Multiplication.....	13
Exercise 8: Condensing HLOOKUP function.....	14
Exercise 9: Simplifying excel sheet Match function	15
Exercise 10: Rlookup (Right lookup) Function	16
Exercise 11: Simplified Vlookup formula version.....	18
Exercise 12: Choose Function.....	19
Exercise 13: Excel Run Time Version Function.....	21
Exercise 14: IsDate Function.....	22
Exercise 15: Auto recalculate.....	23
Exercise 16: Convert VBA Function to Excel Function.....	24
Exercise 17: National festive days Function.....	25
Exercise 18: Static Commission Function.....	28

Exercise 19: A Custom Function.....	29
Exercise 20: View the Formula of cell in another Cell.....	30
Exercise 21: Function with Case Insensitive	31
Exercise 22: Columnwidth function.....	32
Exercise 23: Rowheight function.....	33
Exercise 24: Monthname Function.....	34
Exercise 25: Space Function.....	36
Conclusion	

1.Introduction

This book is written to provide the basics of the Excel VBA Formula for excel spread sheet ,this book VBA Formulas offer ease and convenience at your understanding

The Book Intends for Excel VBA User's & beginners.

The Book Covers Way to work with VBA Formulas for spreadsheet

Influence the Basic Understanding of Formulas & VBA Formulas to Strike Similarity in spreadsheet management.

The book is recommended to all the potential Learners who look for help in understanding the Excel VBA Formula's fundamentals and will get insight of an appropriate Way to do so. All functions are 2016 Excel IDE Designed

Exercise 1: IIF Function

1. Create a UDF (User Defined Function)

To create this Function, execute the following steps

1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

```
Function imedife(key)  
imedife = If(key <= 18, "Minor", "Major")  
End Function
```

This Will Create A new UDF (User Defined Function) with name imedife, we Can use this As we Required. now type in Excel the function = imedife ,the UDF shows in small letter to differentiate itself from Standard Excel Formula

This Code lines will create function imedife ()

In Cell B1 Type

= imedife (A1)



The screenshot shows a portion of an Excel spreadsheet. The columns are labeled 'A' and 'B'. Row 1 contains the value '31' in column A and the text 'Major' in column B. Row 2 is empty. This demonstrates that the function 'imedife' correctly returns 'Major' for the input value 31.

	A	B
1	31	Major
2		

2. Use Excel Formulas (Lookup, Vlookup)

Syntax

IF(logical_test, value_if_true, [value_if_false])

=IF(A1<18,"Minor","Major")

Exercise 2: Custom Address Function

1. Create a UDF (User Defined Function)

To create this Function, execute the following steps

1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

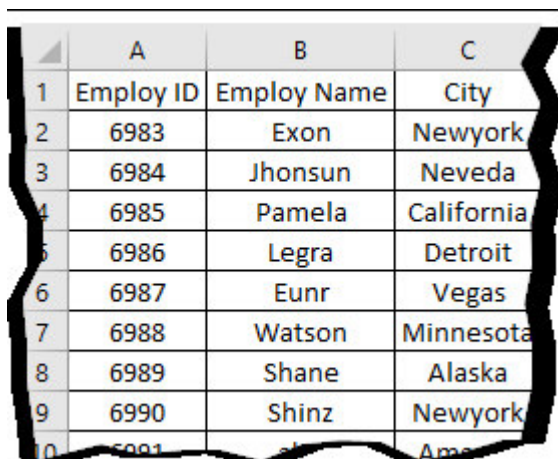
```
Function Empycity(ID As String)  
If ID = "6983" Then Empycity = "Newyork"  
If ID = "6984" Then Empycity = "Neveda"  
If ID = "6985" Then Empycity = "California"  
End Function
```

This Will Create A new UDF (User Defined Function) with name Empycity, we Can use this As we Required. now type in Excel the function =Empycity ,the UDF shows in small letter to differentiate itself from Standard Excel Formula

This Code lines will create function Empycity ()

In Cell C1 Type

= Empycity (A1)



	A	B	C
1	Employ ID	Employ Name	City
2	6983	Exon	Newyork
3	6984	Jhonsun	Neveda
4	6985	Pamela	California
5	6986	Legra	Detroit
6	6987	Eunr	Vegas
7	6988	Watson	Minnesota
8	6989	Shane	Alaska
9	6990	Shinz	Newyork
10	6991	sh	Ame

2. Use Excel Formulas (Lookup, Vlookup)

Syntax

LOOKUP(lookup_value, array)

=LOOKUP(6983,A1:C10) ,

Excel will perform search to find out the city of the employeeID 6983

Syntax

VLOOKUP (lookup_value, table_array, col_index_num, [range_lookup])

=VLOOKUP(6983,A1:C10,3,TRUE)

Excel will perform search to find out the city of the employeeID 6983

Exercise 3: Standard IF Function

1. Create a UDF (User Defined Function)

Use the If Then statement in Excel VBA code lines function to act if a specific condition is met. To create this Function, execute the following steps

1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

```
Function grade(S)
    If S < 19 Then
        grade = "Worst"
    Else
        If S < 49 Then
            grade = "Average"
        Else
            If S < 79 Then
                grade = "Good"
            Else
                If S < 100 Then
                    grade = "Excellent"
                Else
                    grade = "n/a"
                End If
            End If
        End If
    End If
End Function
```

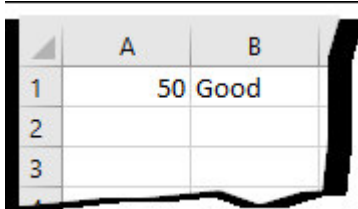
This Will Create A new UDF (User Defined Function) with name grade(S), we Can use this As we Required. type in Excel the function = grade(A1) ,the UDF Comes up in small letter's to differentiate itself from Standard Excel Formula

Excel VBA Formulas for Spreadsheet

This Code lines will create function grade ()

In Cell B1 Type

```
= grade(A1)
```



	A	B
1	50	Good
2		
3		

2. Use Excel Formulas (IF)

Syntax

```
IF(logical_test, value_if_true, [value_if_false])
```

```
=IF(B1>79,"Excellent",IF(B1>49,"Good",IF(B1>19,"Average",IF(B1>1,"Worst","N/a"))))
```

Exercise 4: Specific IF Function

1. Create a UDF (User Defined Function)

To create this Function, execute the following steps

1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

Use the If Then statement in Excel VBA code lines function to act if any of 2 specific condition is met. Insert the below code in a New Module

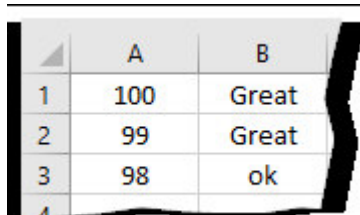
```
Function Lans(Lval)  
If Lval = "100" Or Lval = "99" Then  
Lans = "Great"  
Else  
Lans = "ok"  
End If  
End Function
```

This Will Create A new UDF (User Defined Function) with name Lans(Lval) we Can use this As we Required. type in Excel the function = Lans(Lval) ,the UDF Comesup in small letter's to differentiate itself from Standard Excel Formula

This Code lines will create function Lans()

In Cell B1 Type

= Lans(A1)



	A	B
1	100	Great
2	99	Great
3	98	ok

2. Use Excel Formulas (IF)

Syntax

Excel VBA Formulas for Spreadsheet

IF(logical_test, value_if_true, [value_if_false])

=IF(A1=100,"Great", IF(A1=99,"Great","ok"))

OR(logical1, [logical2], ..

=IF(OR(A1=100,A1=99),"Great","ok")

Exercise 5: IF,AND Function

1.Create a UDF (User Defined Function)

To create this Function, execute the following steps

- 1.Open Excel VBA (Alt + F11)
- 2.Insert a New Module
- 3.In the Project Explorer, double click on the Module
- 4.Add the following code line:

Use the If Then statement in Excel VBA code lines function to act if the both 2 specific condition is met. Insert the below code in a New Module.

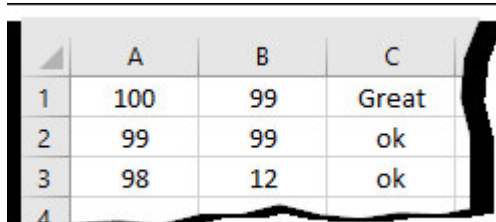
```
Function Lhans(Lval, Lvxc)
If Lval = "100" And Lvxc = "99"
Then
Lhans = "Great"
Else
Lhans = "ok"
End If
End Function
```

This Will Create A new UDF (User Defined Function) with name Lhans(Lval, Lvxc) we Can use this As we Required. type in Excel the function As =Lhans(Lval, Lvxc), the UDF Comes up in small letter's to differentiate itself from Standard Excel Formula

This Code lines will create function Lhans()

In Cell C1 Type

= Lhans(A1,B1)



	A	B	C
1	100	99	Great
2	99	99	ok
3	98	12	ok
4			

2.Use Excel Formulas (IF,AND)

Syntax

IF(logical_test, value_if_true, [value_if_false])

AND(logical1, [logical2], ...)

=IF(AND(A1=100,B1=99),"Great","ok")

Exercise 6: Multi OR Function

1.Create a UDF (User Defined Function)

To create this Function, execute the following steps

- 1.Open Excel VBA (Alt + F11)
- 2.Insert a New Module
- 3.In the Project Explorer, double click on the Module
- 4.Add the following code line:

Use the If Then statement in Excel VBA code lines function to act if the both 2 specific condition is met. Insert the below code in a New Module.

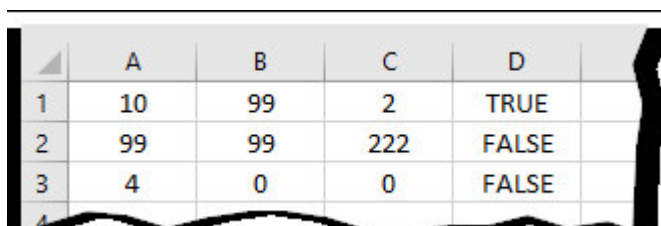
```
Function Lhazs(Lval, Lvxc, Lvzl)
If Lval = "100" Or Lvxc = "99" Or
Lvzl = "99" Then
Lhazs = "TRUE"
Else
Lhazs = "FALSE"
End If
End Function
```

This Will Create A new UDF (User Defined Function) with name Lhazs(Lval, Lvxc, Lvzl) we Can use this As we Required. type in Excel the function As = Lhazs(Lval, Lvxc, Lvzl)), the UDF Comes up in small letter's to differentiate itself from Standard Excel Formula,if any one condition met it turns TRUE.

This Code lines will create function Lhazs()

In Cell D1 Type

=Lhazs(A1,B1,C1)



	A	B	C	D
1	10	99	2	TRUE
2	99	99	222	FALSE
3	4	0	0	FALSE

2.Use Excel Formulas (XOR)

Syntax

XOR(logical1, [logical2],...)

=XOR(A1>30,B1>50,C1=20)

Exercise 7: Simple Multiplication

To create this Function, execute the following steps

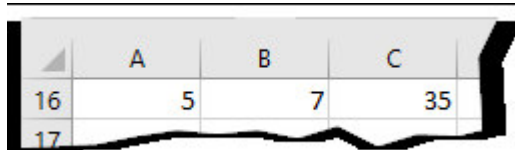
1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

```
Function Multiply(x As Double, y As Double) As Double  
Multiply = x * y  
End Function
```

This Code lines will create function Multiply()

In Cell C1 Type

= Multiply(A1,B1)



A screenshot of an Excel spreadsheet showing a grid with columns A, B, and C, and rows 16 and 17. Cell A16 contains the number 5, cell B16 contains the number 7, and cell C16 contains the result 35. Row 17 is partially visible below row 16.

	A	B	C
16	5	7	35
17			

2. Use Excel Formulas

=A1*B1

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

