



A Formula-Friendly with VBA

EXCEL VBA FORMULAS FOR SPREADSHEET

Excel VBA formulas for Spreadsheet

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

Edition 1

Contents

[Introduction](#)

[Exercise 1: IIF Function](#)

[Exercise 2: Custom Address Function](#)

[Exercise 3: Standard IF Function](#)

[Exercise 4: Specific IF Function](#)

[Exercise 5: IF,AND Function](#)

[Exercise 6: Multi OR Function](#)

[Exercise 7: Simple Multiplication](#)

[Exercise 8: Condensing HLOOKUP function](#)

[Exercise 9: Simplifying excel sheet Match function](#)

[Exercise 10: Rlookup \(Right lookup\) Function](#)

[Exercise 11: Simplified Vlookup formula version](#)

[Exercise 12: Choose Function](#)

[Exercise 13: Excel Run Time Version Function](#)

[Exercise 14: IsDate Function](#)

[Exercise 15: Auto recalculate](#)

[Exercise 16: Convert VBA Function to Excel Function](#)

[Exercise 17: National festive days Function](#)

[Exercise 18: Static Commission Function](#)

[Exercise 19: A Custom Function](#)

[Exercise 20: View the Formula of cell in another Cell](#)

[Exercise 21: Function with Case Insensitive](#)

[Exercise 22: Columnwidth function](#)

[Exercise 23: Rowheight function](#)

[Exercise 24: Monthname Function](#)

[Exercise 25: Space Function](#)

[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:

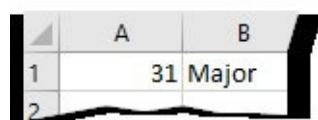
```
Function imedife(key)  
imedife = IIf(key <= 18, "Minor", "Major")  
End Function
```

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'. In row 1, column A contains the number '31' and column B contains the text 'Major'. Row 2 is empty.

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

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

This Code lines will create function Emplycity ()

In Cell C1 Type

= Emplycity (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	Am

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

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 "tandard 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

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 "tandard 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 "tandard 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

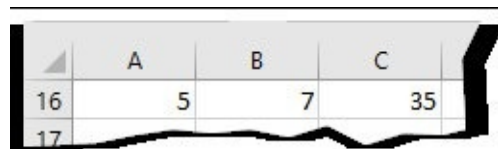
1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

```
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 value 5, cell B16 contains the value 7, and cell C16 contains the value 35. Row 17 is empty.

	A	B	C
16	5	7	35
17			

2. Use Excel Formulas

=A1*B1

Exercise 8: Condensing HLOOKUP 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 hlkup(product)  
    hlkup = Application.WorksheetFunction.HLookup(product,Range("produ"), 3, 0)  
End Function
```

This Code lines will create function hlkup() In Cell C2 Type

= hlkup(product,Range("produ"), 3, 0)

	A	B	C
1	product	value	Qty
2	apple	10	2
3	oranze	14	5
4	pine	21	5
5	neem	9	8
6	donut	19	7
7	Pizza	75	8
8	Samosa	12	3
9			

Syntax

HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])

=HLOOKUP(C25,B17:D22,3,FALSE)

Exercise 9. Simplifying excel sheet Match 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 Matched(product)  
Matched = Application.WorksheetFunction.Match(product, Range("stk"), 0)  
End Function
```

This Code lines will create function Matched ()

In Cell E2 Type

= Matched (product, Range("stk"), 0)

	A	B	C	D	E
1	product	value	Qty		
2	apple	10	2	Pizza	6
3	oranze	14	5	neem	4
4	pine	21	5	apple	1
5	neem	9	8	oranze	2
6	donut	19	7	oranze	2
7	Pizza	75	8	oranze	2
8	Samosa	12	3	pine	3

Syntax

2. Use Excel Formulas (XOR)

MATCH(lookup_value, lookup_array, [match_type])

=MATCH(E17,\$C\$17:\$C\$22,0)

Exercise 10: Create Rlookup (Right lookup) Function

1. Create a UDF Rlookupname (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:

Define Name With Below Range

score :A2:C8

screval :C2:C8

Scorval:A2:A8

Open Excel VBA (Alt + F11)

Insert the below code in a New Module

```
Function Rlookupname(val)
Rlookupname = Application.WorksheetFunction.Index(Range("scorval"),
WorksheetFunction.Match(val, Range("screval"), 0))
End Function
```

This Code lines will create function Rlookupname ()

In Cell D1 Type

= Rlookupname(88)

It will show result As 4

	A	B	C
1	Number	Name	Screval
2	1	John	22
3	2	Jenny	55
4	3	Benny	77
5	4	Tim	88
6	8	Justin	99
7	5	Messi	74
8	6	Francis	23

Use Excel Formulas (Index,Match)

Syntax

INDEX(array, row_num, [column_num])

MATCH(lookup_value, lookup_array, [match_type])

=INDEX(A2:C8,(MATCH(G2,C2:C8,0)),2)

Exercise 11 : Create simplified Vlookup version

1. Create a UDF Vlkup(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 Vlkup(val)  
Vlkup = Application.WorksheetFunction.VLookup(val,Range("score"), 3, 0)  
End Function
```

Define Name With Below Range score :A2:C8

	A	B	C
1	Number	Name	Screval
2	1	John	22
3	2	Jenny	55
4	3	Benny	77
5	4	Tim	88
6	8	Justin	99
7	5	Messi	74
8	6	Francis	23

Syntax

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

=VLOOKUP(8,score,3,FALSE)

Exercise 12: Choose Function

1. Create a UDF DayName((User Defined Function)

To create this Function, execute the following steps

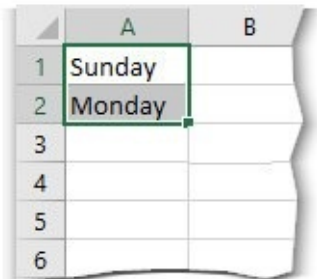
1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

```
Function DayName(jour As Date) As String
    DayName = Choose(Weekday(jour), "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday")
End Function
```

The Code Will Create A new UDF (User Defined Function) with name DayName, With this now type in Excel the function = DayName ,the UDF shows in small letter to differentiate itself from Standard Excel Formula

=DayName(11/4) returns day name in current year

=DayName(11/5/2015) returns day name in the year 2015



	A	B
1	Sunday	
2	Monday	
3		
4		
5		
6		

2. Use Excel Formulas (Choose)

Syntax

CHOOSE(index_num, value1, [value2], ...)

= CHOOSE(3,"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

Excel will perform search for 3rd And the Day Results As Tuesday

In File – Options –Advanced – General - Enter Custom list As Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

Upon Entering A day Drag the Cell bottom for Next day to follow

Exercise 13: Excel RT Version 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:

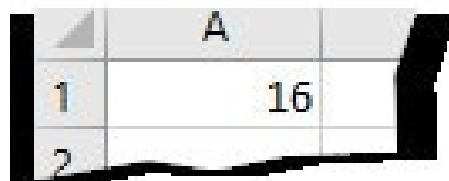
```
Function xlVers() As Integer  
xlVers = Application.Version  
End Function
```

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

In Cell A1 Type

= xlVers()

It will show result As 16



A screenshot of an Excel spreadsheet showing a single cell in column A, row 1, containing the value 16. The cell is highlighted with a black border. The spreadsheet grid shows columns A and B, and rows 1 and 2.

	A	B
1	16	
2		

Exercise 14: IsDate 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 isDat(S)  
isDat = isDate(S)  
End Function
```

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

In Cell B1 Type

= isDat (A1)

It will show result As TRUE

Exercise 15: Auto recalculate

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 Application.Volatile True statement in Excel VBA function code lines to force the formula to recalculate, Insert the below code in a New Module.

```
Function Qtravg()  
  
Application.Volatile True  
  
Qtravg = (Worksheets("Sheet1").Range("B2") + Worksheets("Sheet1").Range("C2") +  
Worksheets("Sheet1").Range("D2") + Worksheets("Sheet1").Range("E2")) / 4  
  
End Function
```

In This the code line Application Volatile True forces the function to recalculate every time the cell or range value changes, the new UDF (User Defined Function) with name Qtravg recalculate's every time.

	A	B	C	D	E	F
1		Q1	Q2	Q3	Q4	Qtravg
2	Sales	52	47	62	52	53.25

In Cell F2 Type

= Qtravg()

2. Use Excel sheet options

1. Press Ctrl + ALT + F9 in sheet
2. Press F9 to recalculate entire spreadsheet

Exercise 16: Convert VBA Function to Excel Function

1. Create a UDF (User Defined Function)
2. Use Defined Name

To create this Function, execute the following steps

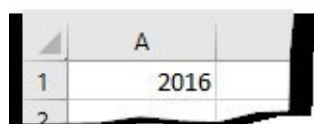
1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

Use the FIX Function of Excel VBA code lines function to use it in Excel sheet ,Insert the below code in a New Module.

```
Function Stdyr()  
Stdyr = Fix(2016)  
End Function
```

This Will Create A new UDF (User Defined Function) with name Stdyr() we Can use this As we Required. type in Excel the function As = Stdyr(), the UDF Comes up in the year as 2016 which is constant figure

Keep the Cell Format as General In Spreadsheet



	A
1	2016
2	

This Code lines will create function Stdyr()

In Cell A1 Type

= Stdyr(A1)

2. Defined Name

Click on Formulas → Defined Formulas → Define Name

Define Name for Any cell With Value “2016 “, Type in any cell As =stdyr

Exercise 17: National festive days 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 FestivDay(FestivalDay As Date)
```

```
Dim FD As String
```

```
Select Case FestivalDay
```

```
Case Is = #1/1/2016#
```

```
FD = "New Year's Day"
```

```
Case Is = #1/5/2016#
```

```
FD = "Labor Day"
```

```
Case Is = #8/5/2016#
```

```
FD = "WWII Victory Day"
```

```
Case Is = #7/14/2016#
```

```
FD = "Bastille Day"
```

```
Case Is = #8/15/1990#
```

```
FD = "Assumption Day"
```

```
Case Is = #11/1/2016#
```

```
FD = "La Toussaint"
```

```
Case Is = #11/11/2016#
```

```
FD = "Armistice Day"
```

```
Case Is = #12/25/2016#
```

```
FD = "Noel Day"
```

```
Case Is = #12/26/2016#
```

```
FD = "Cristmas Day(Alsace)"
```

```
Case #1/1/2016# To #12/31/2016#
```

```
FD = "Non-Festival Day"
```

```
End Select
```

```
FestivDay = FD
```

= FestivDay((A1)

This Will Create A new UDF (User Defined Function) with name FestivDay((A1)we Can use this As we Required. type in Excel the function As = FestivDay(A1), the UDF Comes up in small letter's to differentiate itself from "tandard Excel Formula,if the day fall's to National

This Code lines will create function FestivDay()

In Cell B1 Type

= FestivDay(A1)

	A	B	
1	01-01-2016	New Year's Day	
2	02-01-2016	Labor Day	
3	03-01-2016	Non-Festival Day	
4	04-01-2016	Non-Festival Day	
5	05-01-2016	Labor Day	
6			

2. Use Excel Formulas (NOT,OR)

Syntax

=NOT(OR(A1>10,B1>10,C1>10))

Exercise 18: Static Commission 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 Commission(ByRef C As Integer) As Integer  
    C = C * 0.15  
    Commission = C  
End Function
```

This Code lines will create function Commission ()

Use Excel Spreadsheet

Click on sheet and insert the function

	A	B	C
1	Amt	Commission	
2	12555	=Commission(A2)	

Exercise 19: A Custom Function

1. Create a UDF (User Defined Function)

To create this Function, execute the following steps

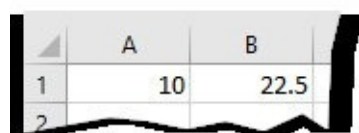
1. Open Excel VBA (Alt + F11)
2. Insert a New Module
3. In the Project Explorer, double click on the Module
4. Add the following code line:

```
Function Brickwall(Count)
Brickwall = (Count * 0.75) + (Count * 1.5)
End Function
```

This Code lines will create function Brickwall()

In Cell B1 Type

= Brickwall(A1)



	A	B
1	10	22.5
2		

Syntax

2. Use Excel Formulas (Bricks *2.25)

Exercise 20: View the Formula of cell in another Cell

1. Create a UDF Formuladisplay (User Defined Function)

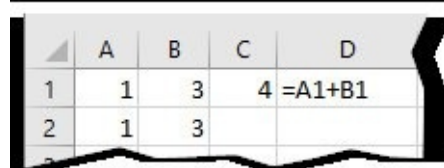
To create this Function, execute the following steps

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:

Insert the below code in a New Module

```
Function Formuladisplay(X)  
Formuladisplay = X.Formula  
End Function
```

This Code lines will create function Formuladisplay () , In Cell D1 Type
=Hedqtrs((C1)



	A	B	C	D
1	1	3	4	=A1+B1
2	1	3		

Syntax FORMULATEXT(reference)

2. Use Excel Formulas (FORMULATEXT)

=FORMULATEXT(C1)

Exercise 21: Function with Case Insensitive

1. Create a UDF Hedqtrs(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 Hedqtrs(C)  
If LCase(C) = "city" Then Hedqtrs = "Newyork"  
End Function
```

This Code lines will create function Hedqtrs() , In Cell B1 Type
= Hedqtrs(A1)

	A	B
1	city	Newyork
2	CITY	Newyork
3		

2. Use Excel Formulas (IF,PROPER)

Syntax

IF(logical_test, value_if_true, [value_if_false])

PROPER(text)

Use (IF,PROPER) function together, In Cell B1 Type

=IF(PROPER(A1)="City","NEWYORK"," ")

Exercise 22: Column Width function


To create this Function, execute the following steps

1. Create a UDF Columnwidth() (User Defined Function)
 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 Columnwidth(Z)  
Columnwidth = Z.Columnwidth  
End Function
```

This Code lines will create function Columnwidth() In Cell B1 Type

=Columnwidth(A1)



The screenshot shows a portion of an Excel spreadsheet with two columns, A and B, and two rows, 1 and 2. Cell A1 is highlighted with a red border. Cell B1 contains the value 8.43. The formula bar is not visible, but the text above indicates that the formula in B1 is =Columnwidth(A1).

	A	B
1		8.43
2		

In Excel Sheet

click on → Home → Cells → Format → Column width to know the size of the Column

Exercise 23: Rowheight() function

To create this Function, execute the following steps

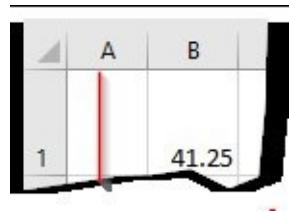
1. Create a UDF Rowheight() (User Defined Function)

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:

This Code lines will create function Rowheight()

In Cell B1 Type

= Rowheight(A1)



In Excel Sheet

click on → Home → Cells → Format → Row height to know the size of the Column

Exercise 24: Monthname Function

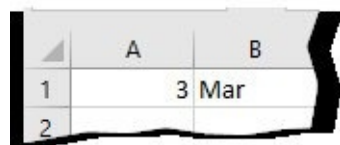
To create this Function, execute the following steps

1. Create a UDF Monthname() (User Defined Function)

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 Mnthname(Mn)  
Mnthname = MonthName(Mn, True)  
End Function
```

This Code lines will create function Monthname() In Cell B1 Type
=Monthname(A1)



	A	B
1		3 Mar
2		

2. Use Excel Formulas (CHOOSE)

Syntax

CHOOSE(index_num, value1, [value2], ...)

=CHOOSE(A1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13)



	A	B	C
1		3 Mar	
2			Jan
3			Feb
4			Mar
5			Apr
6			May
7			Jun
8			Jul
9			Aug
10			Sep
11			Oct
12			Nov
13			Dec
14			

Exercise 25: Space 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 Spce(W)  
Spce = Space(W)  
End Function
```

This Code lines will create function Spce() In Cell B1 Type

= Spce(3)

In D1 As =A1&B1&C1

	A	B	C	D
1	AAA		BBB	AAA BBB
2				

2. Use Keyboard space button

Conclusion

A User Defined function that is located in a code module in a workbook will go along with the Workbook As long as you share. Likewise the function remains with the workbook and will always be available to share.

If your workbook refers UDF you can save the Macro code –Module in Basic file (.bas) format Mail the. bas file to someone with whom file you want the UDF to be available else you can share the file also.

This book was distributed courtesy of:



For your own Unlimited Reading and FREE eBooks today, visit:

<http://www.free-ebooks.net/>

Share this eBook with anyone and everyone automatically by selecting any of the options below:



[Share on Facebook](#)



[Share on Twitter](#)



[Share on LinkedIn](#)

To show your appreciation to the author and help others have wonderful reading experiences and find helpful information too, we'd be very grateful if you'd kindly

[post your comments for this book here.](#)



COPYRIGHT INFORMATION

Free-eBooks.net respects the intellectual property of others. When a book's copyright owner submits their work to Free-eBooks.net, they are granting us permission to distribute such material. Unless otherwise stated in this book, this permission is not passed onto others. As such, redistributing this book without the copyright owner's permission can constitute copyright infringement. If you believe that your work has been used in a manner that constitutes copyright infringement, please follow our Notice and Procedure for Making Claims of Copyright Infringement as seen in our Terms of Service here:

<http://www.free-ebooks.net/tos.html>