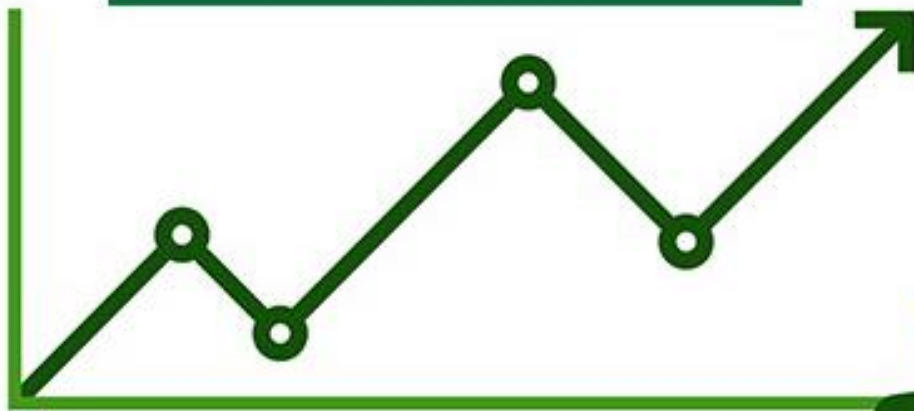


EXCEL VBA PROGRAMMING

Avinash Verma



Excel



MATLABSOLUTIONS

Contents

About	1
Chapter 1: Getting started with Excel VBA	2
Section 1.1: Opening the Visual Basic Editor (VBE)	3
Section 1.2: Declaring Variables	5
Section 1.3: Adding a new Object Library Reference	6
Section 1.4: Hello World	10
Section 1.5: Getting Started with the Excel Object Model	12
Chapter 2: Arrays	16
Section 2.1: Dynamic Arrays (Array Resizing and Dynamic Handling)	16
Section 2.2: Populating arrays (adding values)	16
Section 2.3: Jagged Arrays (Arrays of Arrays)	17
Section 2.4: Check if Array is Initialized (If it contains elements or not)	17
Section 2.5: Dynamic Arrays [Array Declaration, Resizing]	17
Chapter 3: Conditional statements	19
Section 3.1: The If statement	19
Chapter 4: Ranges and Cells	21
Section 4.1: Ways to refer to a single cell	21
Section 4.2: Creating a Range	21
Section 4.3: Offset Property	23
Section 4.4: Saving a reference to a cell in a variable	23
Section 4.5: How to Transpose Ranges (Horizontal to Vertical & vice versa)	23
Chapter 5: Named Ranges	25
Section 5.1: Define A Named Range	25
Section 5.2: Using Named Ranges in VBA	25
Section 5.3: Manage Named Range(s) using Name Manager	26
Section 5.4: Named Range Arrays	28
Chapter 6: Merged Cells / Ranges	29
Section 6.1: Think twice before using Merged Cells/Ranges	29
Chapter 7: Locating duplicate values in a range	30
Section 7.1: Find duplicates in a range	30
Chapter 8: User Defined Functions (UDFs)	32
Section 8.1: Allow full column references without penalty	32
Section 8.2: Count Unique values in Range	33
Section 8.3: UDF - Hello World	33
Chapter 9: Conditional formatting using VBA	36
Section 9.1: FormatConditions.Add	36
Section 9.2: Remove conditional format	37
Section 9.3: FormatConditions.AddUniqueValues	37
Section 9.4: FormatConditions.AddTop10	38
Section 9.5: FormatConditions.AddAboveAverage	38
Section 9.6: FormatConditions.AddIconSetCondition	38
Chapter 10: Workbooks	41
Section 10.1: When To Use ActiveWorkbook and ThisWorkbook	41
Section 10.2: Changing The Default Number of Worksheets In A New Workbook	41
Section 10.3: Application Workbooks	41
Section 10.4: Opening A (New) Workbook, Even If It's Already Open	42

Section 10.5: Saving A Workbook Without Asking The User	43
Chapter 11: Working with Excel Tables in VBA	44
Section 11.1: Instantiating a ListObject	44
Section 11.2: Working with ListRows / ListColumns	44
Section 11.3: Converting an Excel Table to a normal range	44
Chapter 12: Loop through all Sheets in Active Workbook	45
Section 12.1: Retrieve all Worksheets Names in Active Workbook	45
Section 12.2: Loop Through all Sheets in all Files in a Folder	45
Chapter 13: Use Worksheet object and not Sheet object	47
Section 13.1: Print the name of the first object	47
Chapter 14: Methods for Finding the Last Used Row or Column in a Worksheet	48
Section 14.1: Find the Last Non-Empty Cell in a Column	48
Section 14.2: Find the Last Non-Empty Row in Worksheet	48
Section 14.3: Find the Last Non-Empty Column in Worksheet	49
Section 14.4: Find the Last Non-Empty Cell in a Row	50
Section 14.5: Get the row of the last cell in a range	50
Section 14.6: Find Last Row Using Named Range	50
Section 14.7: Last cell in Range.CurrentRegion	51
Section 14.8: Find the Last Non-Empty Cell in Worksheet - Performance (Array)	51
Chapter 15: Creating a drop-down menu in the Active Worksheet with a Combo Box	54
Section 15.1: Example 2: Options Not Included	54
Section 15.2: Jimi Hendrix Menu	55
Chapter 16: File System Object	57
Section 16.1: File, folder, drive exists	57
Section 16.2: Basic file operations	57
Section 16.3: Basic folder operations	58
Section 16.4: Other operations	58
Chapter 17: Pivot Tables	60
Section 17.1: Adding Fields to a Pivot Table	60
Section 17.2: Creating a Pivot Table	60
Section 17.3: Pivot Table Ranges	63
Section 17.4: Formatting the Pivot Table Data	63
Chapter 18: Binding	64
Section 18.1: Early Binding vs Late Binding	64
Chapter 19: autofilter ; Uses and best practices	66
Section 19.1: Smartfilter!	66
Chapter 20: Application object	70
Section 20.1: Simple Application Object example: Display Excel and VBE Version	70
Section 20.2: Simple Application Object example: Minimize the Excel window	70
Chapter 21: Charts and Charting	71
Section 21.1: Creating a Chart with Ranges and a Fixed Name	71
Section 21.2: Creating an empty Chart	72
Section 21.3: Create a Chart by Modifying the SERIES formula	73
Section 21.4: Arranging Charts into a Grid	75
Chapter 22: CustomDocumentProperties in practice	79
Section 22.1: Organizing new invoice numbers	79
Chapter 23: PowerPoint Integration Through VBA	82
Section 23.1: The Basics: Launching PowerPoint from VBA	82

Chapter 24: How to record a Macro	83
Section 24.1: How to record a Macro	83
Chapter 25: SQL in Excel VBA - Best Practices	85
Section 25.1: How to use ADODB.Connection in VBA?	85
Chapter 26: Excel-VBA Optimization	87
Section 26.1: Optimizing Error Search by Extended Debugging	87
Section 26.2: Disabling Worksheet Updating	88
Section 26.3: Row Deletion - Performance	88
Section 26.4: Disabling All Excel Functionality Before executing large macros	89
Section 26.5: Checking time of execution	90
Section 26.6: Using With blocks	91
Chapter 27: VBA Security	93
Section 27.1: Password Protect your VBA	93
Chapter 28: Debugging and Troubleshooting	94
Section 28.1: Immediate Window	94
Section 28.2: Use Timer to Find Bottlenecks in Performance	95
Section 28.3: Debugger Locals Window	95
Section 28.4: Debug.Print	96
Section 28.5: Stop	97
Section 28.6: Adding a Breakpoint to your code	97
Chapter 29: VBA Best Practices	98
Section 29.1: ALWAYS Use "Option Explicit"	98
Section 29.2: Work with Arrays, Not With Ranges	100
Section 29.3: Switch off properties during macro execution	101
Section 29.4: Use VB constants when available	102
Section 29.5: Avoid using SELECT or ACTIVATE	103
Section 29.6: Always define and set references to all Workbooks and Sheets	105
Section 29.7: Use descriptive variable naming	105
Section 29.8: Document Your Work	106
Section 29.9: Error Handling	107
Section 29.10: Never Assume The Worksheet	109
Section 29.11: Avoid re-purposing the names of Properties or Methods as your variables	109
Section 29.12: Avoid using ActiveCell or ActiveSheet in Excel	110
Section 29.13: WorksheetFunction object executes faster than a UDF equivalent	111
Chapter 30: Excel VBA Tips and Tricks	113
Section 30.1: Using xlVeryHidden Sheets	113
Section 30.2: Using Strings with Delimiters in Place of Dynamic Arrays	114
Section 30.3: Worksheet .Name, .Index or .CodeName	114
Section 30.4: Double Click Event for Excel Shapes	116
Section 30.5: Open File Dialog - Multiple Files	117
Chapter 31: Common Mistakes	118
Section 31.1: Qualifying References	118
Section 31.2: Deleting rows or columns in a loop	119
Section 31.3: ActiveWorkbook vs. ThisWorkbook	119
Section 31.4: Single Document Interface Versus Multiple Document Interfaces	120

Chapter 1: Getting started with Excel VBA

Microsoft Excel includes a comprehensive macro programming language called VBA. This programming language provides you with at least three additional resources:

1. Automatically drive Excel from code using Macros. For the most part, anything that the user can do by manipulating Excel from the user interface can be done by writing code in Excel VBA.
2. Create new, custom worksheet functions.
3. Interact Excel with other applications such as Microsoft Word, PowerPoint, Internet Explorer, Notepad, etc.

VBA stands for Visual Basic for Applications. It is a custom version of the venerable Visual Basic programming language that has powered Microsoft Excel's macros since the mid-1990s.

IMPORTANT

Please ensure any examples or topics created within the excel-vba tag are **specific** and **relevant** to the use of VBA with Microsoft Excel. Any suggested topics or examples provided that are generic to the VBA language should be declined in order to prevent duplication of efforts.

- on-topic examples:

- ✓ *Creating and interacting with worksheet objects*
- ✓ *The WorksheetFunction class and respective methods*
- ✓ *Using the xlDirection enumeration to navigate a range*

- off-topic examples:

- ✗ *How to create a 'for each' loop*
- ✗ *MsgBox class and how to display a message*
- ✗ *Using WinAPI in VBA*

VB

Version Release Date

VB6	1998-10-01
VB7	2001-06-06
WIN32	1998-10-01
WIN64	2001-06-06
MAC	1998-10-01

Excel

Version Release Date

16	2016-01-01
15	2013-01-01
14	2010-01-01
12	2007-01-01
11	2003-01-01
10	2001-01-01
9	1999-01-01