

Share on your Social Media



Featured Articles

VBA Macros Tutorial for Beginners

Published On: October 10, 2024

VBA Macros Tutorial for Beginners

VBA macros are programs that automate repetitive operations in Microsoft Excel and other applications using the Visual Basic for Applications (VBA) programming language. Learn the fundamentals in this VBA Macros tutorial designed for beginners.

[Download VBA Macros Tutorial PDF](#)

Introduction to VBA Macros

VBA Macros automates manual chores by developing custom user-generated routines and using Excel's Visual Basic Application to create automated procedures. We cover the following in this VBA Macros tutorial:

- Overview of VBA Macros
- Applications of VBA Macros
- Using VBA Editor in MS Excel
- VBA Macros Terminologies
- Message Box in VBA Macros
- Error Handling in VBA Macros
- Sample Program in VBA Macros in Excel
- Advantages of VBA Macros

Overview of VBA Macros

Want to know more about becoming an expert in IT?



Click Here to Get Started >>

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER

IBI

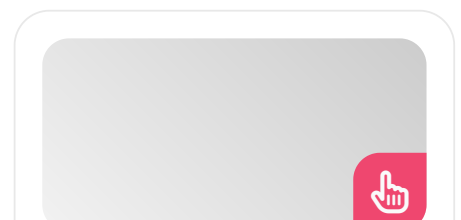


Quick Enquiry

Related Courses at SLA

- ➔ VBA Macros Online Training
- ➔ VBA Macros Training in OMR
- ➔ VBA Macros Training in Chennai

Related Posts



VBA increases the capabilities of some programs. It is frequently used to construct custom forms, automate processes, develop macros, and carry out repetitive operations that may require little or no human participation.

VBA macros are useful for the following purposes:

- **Increase the speed of manual tasks:** VBA macros can automate processes to minimize human intervention and expedite manual labor.
- **Provide individualized remedies:** Custom functions, toolbars, menus, dialog boxes, and forms can all be made with VBA macros.
- **Use the Windows API:** The Windows Application Programming Interface (API) can be accessed via VBA.

In Excel, to record a macro, you can:

- Verify that the ribbon's Developer tab is displayed.
- From the Developer tab, choose the Record Macro button.
- Execute the tasks you wish to have automated.

By default, the Record Macro button creates VBA code to replicate your actions. Reshape your career with our [advanced Excel course in Chennai](#).

[VBA Macros Interview Questions](#)

Key Features of VBA Macros

Macros written in Visual Basic for Applications (VBA) contain numerous important features, such as:

- **Automation:** Repeated operations like data formatting, report generation, and record updating can be automated with VBA macros. Time can be saved, and there is a lower chance of human error.

VBA Developer Salary in Chennai

Published On: October 12, 2024

Introduction A VBA Developer focuses on using Visual Basic for Applications to automate tasks and...

Tableau Developer Salary in Chennai

Published On: October 12, 2024

Introduction A Tableau Developer designs, develops, and maintains dashboards and visualizations using Tableau software. Key...

VMware Tutorial for Cloud Computing Aspirants

Published On: October 12, 2024

VMware Tutorial for Cloud Computing Aspirants VMware software allows you to run a virtual machine...

VBA Macros Tutorial for Beginners

Published On: October 10, 2024

VBA Macros Tutorial for Beginners VBA macros are programs that automate repetitive operations in Microsoft...

- **Customization:** To improve the functionality of Office apps, users can write their custom add-ins and functions.
- **Collaboration:** VBA enables communication between various Office apps. One way that users can interact with Outlook is by writing code in Excel.
- **User Interface Integration:** VBA's user interface integration feature enables users to design unique user forms and interfaces for gathering and organizing user input.
- **Event-driven programming:** When a user clicks a button, opens a workbook, or modifies data in a cell, VBA can react to those events.
- **Debugging:** Debugging is integrated into the Excel VBA editor, enabling users to step through the code and observe what occurs at each level.
- **String manipulation:** VBA has functions for working with strings and numbers, which are helpful when building both basic and sophisticated programs.
- **Error Handling:** VBA's error handling feature alerts users right away when they write incorrect code.

Enroll in our [MS Office training in Chennai](#) to learn the fundamentals.

Applications of VBA Macros

Office applications can be expanded using the programming language VBA. All it takes to write a macro is one click. Macro applications for Visual Basic for Applications (VBA) can be used for many purposes, such as:

- **Automating repetitive tasks:** Formatting, data cleansing, and report production are just a few of the things that VBA macros can automate.
- **Custom forms and user interfaces** can be made with VBA in order to direct users through procedures and guarantee that data is submitted accurately.

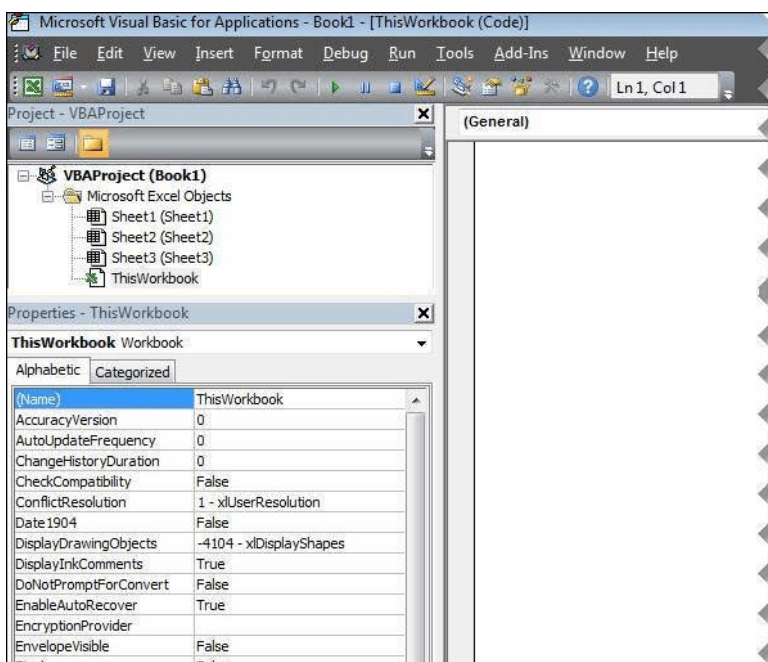
- **Managing databases:** Data entry, query execution, and report creation can all be automated with VBA.
- **Creating custom reports and dashboards:** With just a single click, VBA may be used to build custom reports and dashboards.
- **Email automation:** Email sending and receiving can be automated with VBA.
- **Document management:** The production and formatting of documents can be automated with VBA.
- **Analyzing vast volumes of data:** VBA is capable of handling large-scale data analysis.
- **Model creation and upkeep:** Models can be created and maintained using VBA.
- **Developing investment scenarios:** Investment scenarios can be developed with VBA.

Explore our wide range of [software training courses](#) to kickstart your career in IT.

Using VBA Editor in MS Excel

Although Visual Basic is a feature-rich programming language, you may start creating apps far more quickly by learning Microsoft Office's VBA programming language instead.

Step 1: To open the Excel window, hit "ALT+F11." As seen in the screenshot below, a VBA window opens.

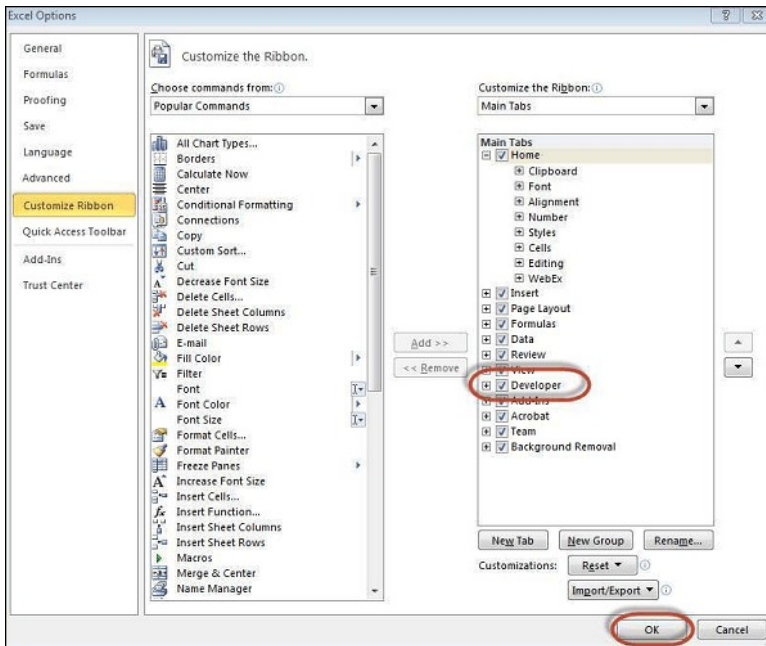


| | |
|---------------------------|-------|
| ForceFullCalculation | False |
| HighlightChangesOnScreen | False |
| InactiveListBorderVisible | True |
| IsAddin | False |
| KeepChangeHistory | True |
| ListChangesOnNewSheet | False |

VBA Macros Tutorial 1

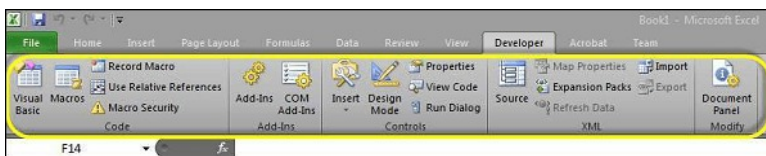
Step 2: Open Excel 2023 and select the “Developer” menu. Similarly, select File → Options.

Step 3: Select the ‘Customize the Ribbon’ tab and enable ‘Developer’. Press “OK.”



VBA Macros Tutorial 2

Step 4: The menu bar displays the ‘Developer’ ribbon.



VBA Macros Tutorial 3

Step 5: Open the VBA Editor by clicking the ‘Visual Basic’ button.

VBA Macros Tutorial 4

Step 6: Add a button to begin scripting. Press Insert
→ Choose the desired button.

VBA Macros Tutorial 5

Step 7: Use the right-click menu to select
“properties.”

VBA Macros Tutorial 6

Step 8: As seen in the screenshot below, edit the
name and caption.

VBA Macros Tutorial 7

Step 9: Double-clicking the button now will cause the sub-procedure outline to appear, as seen in the screenshot below.

VBA Macros Tutorial 8

Step 10: To begin coding, just add a message.

```
Private Sub say_helloworld_Click()
```

```
    MsgBox "Hi"
```

```
End Sub
```

Step 11: Press the button to start the subprocess. The following screenshot displays the sub-procedure's output. Verify that the design mode is activated. If it isn't on, just click it to make it so.

People with zero coding expertise can enter into IT through our [web design courses in Chennai](#).

VBA Macros Course Syllabus

VBA Macros Terminologies

Familiarize yourself with the Excel VBA terminologies that are frequently used. Since each of these terms will be utilized in later sections, it is crucial to learn them all.

Modules: The place where the code is written is in modules.

VBA Macros Tutorial 10

- Navigate to Insert → Module to insert a module. 'module1' is formed once a module is inserted.

VBA Macros Tutorial 11

We can write VBA code inside the modules, and that code is written inside a procedure. A sequence of VBA statements that specify actions to take make up a procedure or subprocedure.

Procedure: Procedures are collections of statements that are run collectively to give Excel

instructions on how to carry out a certain task.

- Either a very basic or a very hard activity can be completed. Nonetheless, it's a good idea to divide up complex processes into simpler ones.
- Sub and Function Procedures are the two primary categories.

VBA Macros Tutorial 12

- **Function:** Anywhere in your application, you can invoke a function, which is a collection of reusable code. As a result, writing the same code repeatedly is no longer necessary.
- This makes it easier for programmers to break up a large program into several smaller, more doable tasks.
- VBA supports writing user-defined functions in addition to the built-in functions, and statements can be inserted between a function and its end.

Syntax

Function Functionname(parameter-list)

statement 1

statement 2

statement 3

.....

statement n

End Function

Example

Function findArea(Length As Double, Optional Width As Variant)

If IsMissing(Width) Then

*findArea = Length * Length*

Else

*findArea = Length * Width*

End If

End Function

- **Sub-Procedures:** Sub-procedures function in a manner akin to functions.
 - Functions might or might not return a value, but subprocedures DO NOT.
 - Subprocedures can be invoked without requiring a call keyword.
 - Sub and End Sub statements surround subprocedures at all times.

While functions might or might not return a value, subroutines DO NOT. It is possible to call subroutines without a call keyword. Sub and End Sub statements surround subroutines at all times.

Example

Sub Area(x As Double, y As Double)

*MsgBox x * y*

End Sub

Comments: To facilitate future programming collaboration on the same code, comments are used to record user information and program logic.

It contains details created by and altered, and it

may also contain logic that has been integrated.
The interpreter ignores comments during execution.

There are two ways in VBA to indicate comments.

- A single quote (') at the beginning of any statement indicates that it is a comment.

Example:

'This script is launched following a successful login.

- Everything that begins with the word "REM"

REM This script was created to verify the input that was entered.

Are you interested in learning analytics? Check out our [digital marketing courses in Chennai](#) for your exponential career growth.

Message Box in VBA Macros

A message box is displayed by the *MsgBox* function, which then waits for the user to press a button before acting accordingly.

Syntax

```
MsgBox(prompt[,buttons][,title][,helpfile,context])
```

Example

```
Function MessageBox_Demo()
```

```
    'Message Box with just prompt message
```

```
    MsgBox("Welcome")
```

```
    'Message Box with title, yes no and cancel
```

```
Buttons
```

```
    int a = MsgBox("Do you like blue color?",3,"Choose options")
```

```
    ' Assume that you press No Button
```

```
    msgbox ("The Value of a is " & a)
```

End Function

Step 1: The previously mentioned function can be called from an Excel worksheet, as seen in the following screenshot, or it can be run by pressing the “Run” button on the VBA window.

VBA Macros Tutorial 13

Step 2: A “Welcome” message and an “OK” button appear in a simple message box.

VBA Macros Tutorial 14

Step 3: After selecting OK, another dialog box containing a message and buttons labeled “yes,” “no,” and “cancel” appears.

VBA Macros Tutorial 15

Step 4: When the “No” button is clicked, its value (7) is recorded as an integer and presented to the user as a message box, as seen in the screenshot below.

It is possible to determine which button the user has clicked by using this value.

VBA Macros Tutorial

16

Do you want to learn a simple programming language that brings a promising career? Enroll in our [Python training in Chennai](#).

[VBA Developer Salary in Chennai](#)

Error Handling in VBA Macros

Programming errors can be classified into three categories: syntax errors, runtime errors, and logical errors.

Syntax Error: Syntax errors, also known as parsing errors, happen while VBScript is being interpreted.

For example, the lack of a closing parenthesis in the line that follows results in a syntactic fault.

Example

```
Function ErrorHandling_Demo()
```

```
    dim x,y
```

```
    x = "SLA"
```

```
    y = Ucase(x
```

```
End Function
```

Runtime Errors: Errors that arise during execution, following interpretation, are referred to as runtime errors or exceptions.

For example, the following line attempts to call `fnmultiply`, a function that does not exist, even though the syntax is fine. This results in a runtime error.

```
Function ErrorHandler_Demo1()
```

```
    Dim x,y
```

```
    x = 10
```

```
    y = 20
```

```
    z = fnadd(x,y)
```

```
    a = fnmultiply(x,y)
```

```
End Function
```

```
Function fnadd(x,y)
```

```
    fnadd = x + y
```

```
End Function
```

Logical Errors: Finding logical faults can be the hardest kind of error to find.

- Runtime or syntax issues are not the cause of these errors.
- Rather, they arise from errors in the logic underlying your script, which prevent you from getting the desired outcome.

Because the kind of logic you wish to include in your software depends on your business requirements, you are unable to detect those problems.

For example, dividing a value by zero or writing a script that runs endlessly.

Error Handling

Both the ability to enable and disable error-

handling routines is provided by VBA.

Any run-time error that transpires without an On Error statement is fatal; it results in the display of an error message and an abrupt termination of the program.

On Error { GoTo [line | 0 | -1] | Resume Next }

- **GoTo line:** Enables the error-handling routine that starts at the line specified in the required line argument.
- **GoTo 0:** Resets the current procedure's enabled error handler to Nothing and disables it.
- **GoTo -1:** Resets the current procedure's enabled exception to Nothing and disables it.
- **Resume Next:** This specifies that in the event of a run-time error, the execution proceeds from the statement that comes just after the error-causing statement.

Example

```
Public Sub OnErrorDemo()
```

```
    On Error GoTo ErrorHandler ' Enable error-  
handling routine.
```

```
    Dim x, y, z As Integer
```

```
    x = 50
```

```
    y = 0
```

```
    z = x / y ' Divide by ZERO Error Raises
```

```
ErrorHandler: ' Error-handling routine.
```

```
    Select Case Err.Number ' Evaluate error number.
```

```
        Case 10 ' Divide by zero error
```

```
            MsgBox ("You attempted to divide by zero!")
```

```
        Case Else
```

```
            MsgBox "UNKNOWN ERROR - Error# " &
```

Err.Number & ": " & Err.Description

End Select

Resume Next

End Sub

Learn the fundamentals of computer programming with our [C and C++ training in Chennai](#).

Sample Program in VBA Macros in Excel

Charts based on specific criteria can be created with VBA. Let's examine it with the help of an example.

Step 1: Input the data that needs to be used as the basis for the graph.

VBA Macros Tutorial 17

Step 2: Make three buttons: one for creating a pie chart, one for creating a bar graph, and one for creating a column chart.

VBA Macros Tutorial 18

Step 3: Create a macro to produce all of these kinds of charts.

' Procedure to Generate Pie Chart

Private Sub fn_generate_pie_graph_Click()

Dim cht As ChartObject


```
For Each cht In Worksheets(1).ChartObjects
```

```
    cht.Chart.Type = xlPie
```

```
Next cht
```

```
End Sub
```

```
' Procedure to Generate Bar Graph
```

```
Private Sub fn_Generate_Bar_Graph_Click()
```

```
    Dim cht As ChartObject
```

```
    For Each cht In Worksheets(1).ChartObjects
```

```
        cht.Chart.Type = xlBar
```

```
    Next cht
```

```
End Sub
```

```
' Procedure to Generate Column Graph
```

```
Private Sub fn_generate_column_graph_Click()
```

```
    Dim cht As ChartObject
```

```
    For Each cht In Worksheets(1).ChartObjects
```

```
        cht.Chart.Type = xlColumn
```

```
    Next cht
```

```
End Sub
```

Step 4: The chart is generated after selecting the relevant button. Click the "generate pie chart" button to view the output that follows.

Enhance your automation skills by learning [RPA courses in Chennai](#).

Advantages of VBA Macros

There are numerous benefits to using VBA (Visual Basic for Applications) macros, such as:

- **Task automation:** By automating repeated operations, VBA macros can reduce errors and save time. You can create a macro, for instance, to open several files with a single click.
- **Creating user-defined functions:** Microsoft Office products can have user-defined functions built into them due to VBA.
- **Error tracking:** VBA can monitor errors and assist users in timely correction.
- **Simple to learn:** Learning VBA macro programming is thought to be enjoyable and simple.
- **Functionality extension:** Documents' functionality can be expanded via VBA.
- **Enhancing user experience:** By including instructions, custom commands, and prompts, VBA macros can lend a more user-friendly feel to documents. When several individuals are working on a document together, this can be really helpful.

[VBA Macros online Training](#)

Conclusion

We hope this VBA Macros tutorial will be helpful for you to understand the fundamentals. By enrolling in our hands-on [VBA Macros course in Chennai](#), you may automate and expedite a variety of operations.

Share on your Social Media



Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai – 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

Courses

Python

Software Testing

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Navigation

[About Us](#)

[Blog Posts](#)

[Careers](#)

[Contact](#)

[Placement Training](#)

[Corporate Training](#)

[Hire With Us](#)

[Job Seekers](#)

[SLA's Recently Placed Students](#)

[Reviews](#)

[Sitemap](#)

Important Links

[Disclaimer](#)

[Privacy Policy](#)

[Terms and Conditions](#)

Social Media Links



Review Sources

[Google](#)

[Trustpilot](#)

[Glassdoor](#)

[Mouthshut](#)

Cloud Computing

Sulekha

Hardware and Networking

Justdial

VBA Macros

Ambitionbox

Mobile App Development

Indeed

DevOps

Software Suggest

Sitejabber

Copyright © 2024 - Softlogic
Systems. All Rights Reserved

SLA™ is a trademark of Softlogic Systems, Chennai.
Unauthorised use prohibited.