

# EXCEL CHAMPS

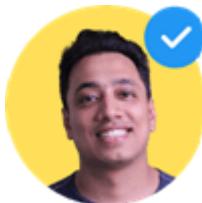
≡ Navigation



≡ Tutorials

[Home](#) > [VBA](#) > How to RENAME a Sheet using VBA in Excel

## How to RENAME a Sheet using VBA in Excel



- by Puneet

### Contents [ hide ]

[Steps to Rename a Sheet using a VBA Code](#)

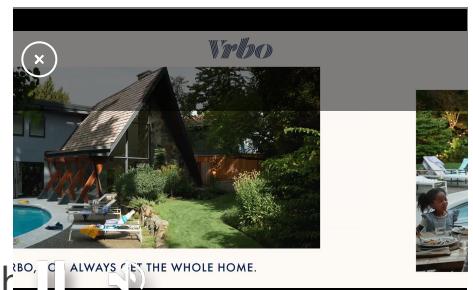
[Rename the Activesheet](#)

[Rename a Sheet using Sheet Number](#)

[Check if the Sheet Exists before Renaming](#)

[Rename a Sheet using Value from a Cell or a Range](#)

### Related Tutorials



When you add a new sheet in a workbook, you have the

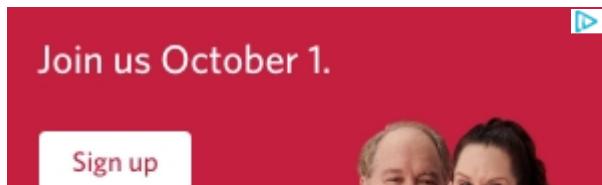
But you can also ren



Property of the

worksheet. In this tutorial, we will look at different ways to rename a sheet or multiple sheets using a VBA code.

## Steps to Rename a Sheet using a VBA Code



1. First, define the sheet that you want to rename using the worksheet object.
2. After that, you need to use (.Name) to access the name property that you want to change.
3. Next, type an equal sign to that to tell VBA the value you want to use to rename the sheet.
4. In the end, type the name of the sheet that you want to use.

A screenshot of a video player interface. At the top, there's a dark grey header with a play button icon (a white triangle) and a circular close button icon. Below the header, the main video area is mostly black, indicating the video has ended. At the bottom, there's a light pink advertisement for Adobe. The ad features the Adobe logo (a stylized 'A') and the text 'Onboard customers. Fast.' in bold. To the right of the text is a 'Try free' button with a white background and black text. To the right of the button is a small screenshot of a web page with various buttons and links, with the URL 'https://na2.documents.adobe.com' visible. There are also icons for 'Copy' and 'Form'. The overall layout is clean and professional.

The code renames the first sheet to 'mySheet'. Four green circles numbered 1 through 4 are overlaid on the code: 1 is under 'Sub', 2 is under 'Sheets("Sheet1").Name', 3 is under 'mySheet', and 4 is under 'End Sub'." data-bbox="220 110 770 318"/>

```
(General) vba_rename_sheet()

Sub vba_rename_sheet()
    Sheets("Sheet1").Name = "mySheet"
End Sub
```



Helpful Links: [Run a Macro](#) – [Macro Recorder](#) – [Visual Basic Editor](#) – [Personal Macro Workbook](#)

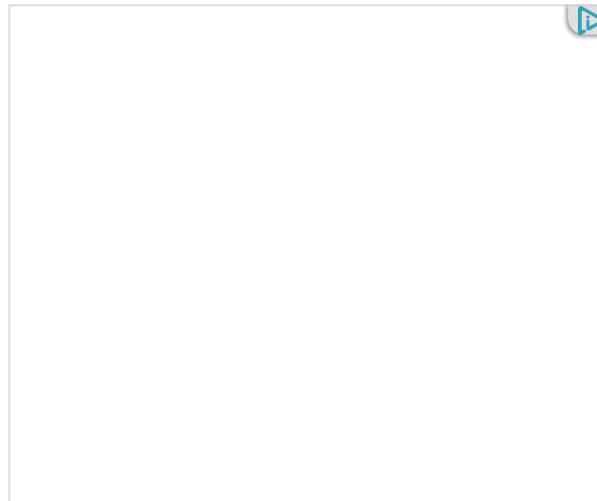
## Rename the ActiveSheet



If you want to rename the active sheet, in that case, you don't need to define the sheet name, instead, you need to use the ActiveSheet object that tells



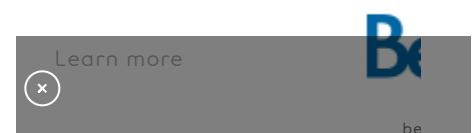
VBA to refer to the sheet that is active right now. Here's the code.



```
Activesheet.Name = "mySheet"
```



Get on Canada's best 5G network.

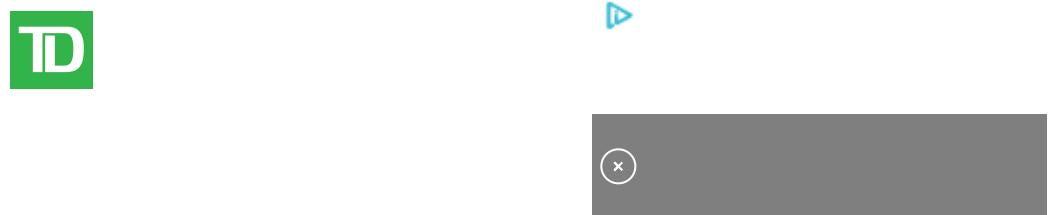


**Note:** To rename a sheet you don't need to activate it.

# Rename a Sheet using Sheet Number

As you know every sheet has a number based on its position in the workbook. Let's say you want to rename the sheet that you have on the fifth number, the code would be.

```
Sheets(5).Name = "mySheet5"
```



When you [run the above macro](#), it renames the sheet that is on the fifth number.

# Check if the Sheet Exists before Renaming

If you try to rename a worksheet that doesn't exist, VBA will show you an error, just like below.



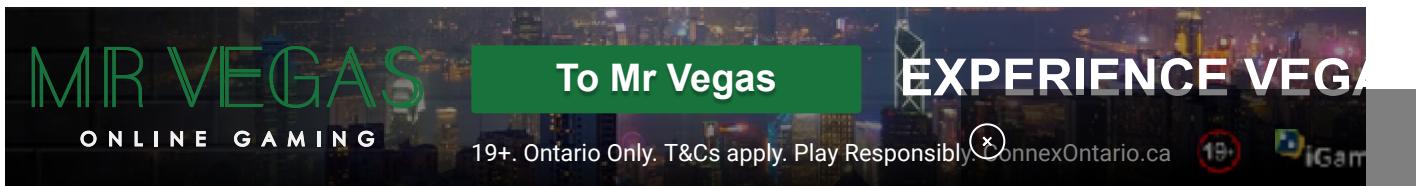
The solution to this problem is the following code that uses **FOR EACH**, which can **loop through all the worksheets** to find the sheet that you have defined and then rename that sheet.

```
Sub check_sheet_rename()
    Dim ws As Worksheet
    Dim mySheet As String
    Dim SheetName As String

    mySheet = InputBox("enter the name of the sheet that you want to rename.")
    SheetName = InputBox("Enter new name for the sheet.")

    For Each ws In ThisWorkbook.Worksheets
        If mySheet = ws.Name Then
            ws.Name = SheetName
        End If
    Next ws

End Sub
```



Rename a Sheet using Value from a Cell or a Range

You can also rename a sheet by taking value from a cell. Let's say the value is in cell A1.

```
Sheets("Sheet1").name = Range("A1").Value
```

But let's say you want to name multiple sheets based on the values in a range of cells. In that case, you need to have code like the following.

```
Sub vba_sheet_rename_multiple()
Dim wsCount As Long
Dim rCount As Long
Dim ws As Worksheet
Dim name As Range
Dim i As Long

wsCount = ThisWorkbook.Worksheets.Count
rCount = Range("A1:A10").Rows.Count

'Checks if the count of the names provided is less _
or more than the sheets in the workbook
If wsCount <> rCount Then
    MsgBox "There's some problem with the names provided."
    Exit Sub
Else
    'Check if any of the cells in the name range is empty.
    For Each name In Range("A1:A10")
        If IsEmpty(name) = True Then
            i = i + 1
        End If
    Next name
    If i > 0 Then
        MsgBox "There's a blank cell in the names range."
        Exit Sub
    End If
End If
```

```
End If  
End If  
  
'rename each sheet using the value from the range cell by cell.  
i = 1  
  
For Each ws In ThisWorkbook.Worksheets  
    ws.name = Range("A1:A10").Cells(i, 1).Value  
    i = 1 + i  
Next ws  
  
End Sub
```



Plus save \$20/mo. for each additional member you add to your account.

B

When you run this VBA code, first it will check if the cells in the range are equal to the number of sheets that you have in the workbook. After that, it will check if all the cells in the range that you have specified have values or not. And in the end, rename all the sheets using those names.

It will verify two conditions using **IF THEN ELSE** and then  rename all the sheets.

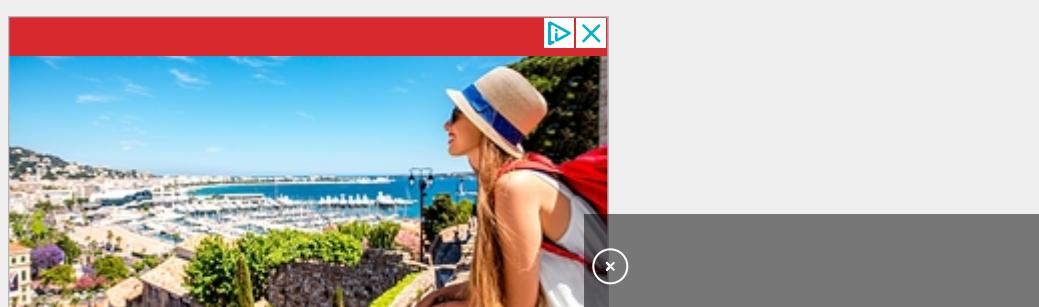
What is VBA

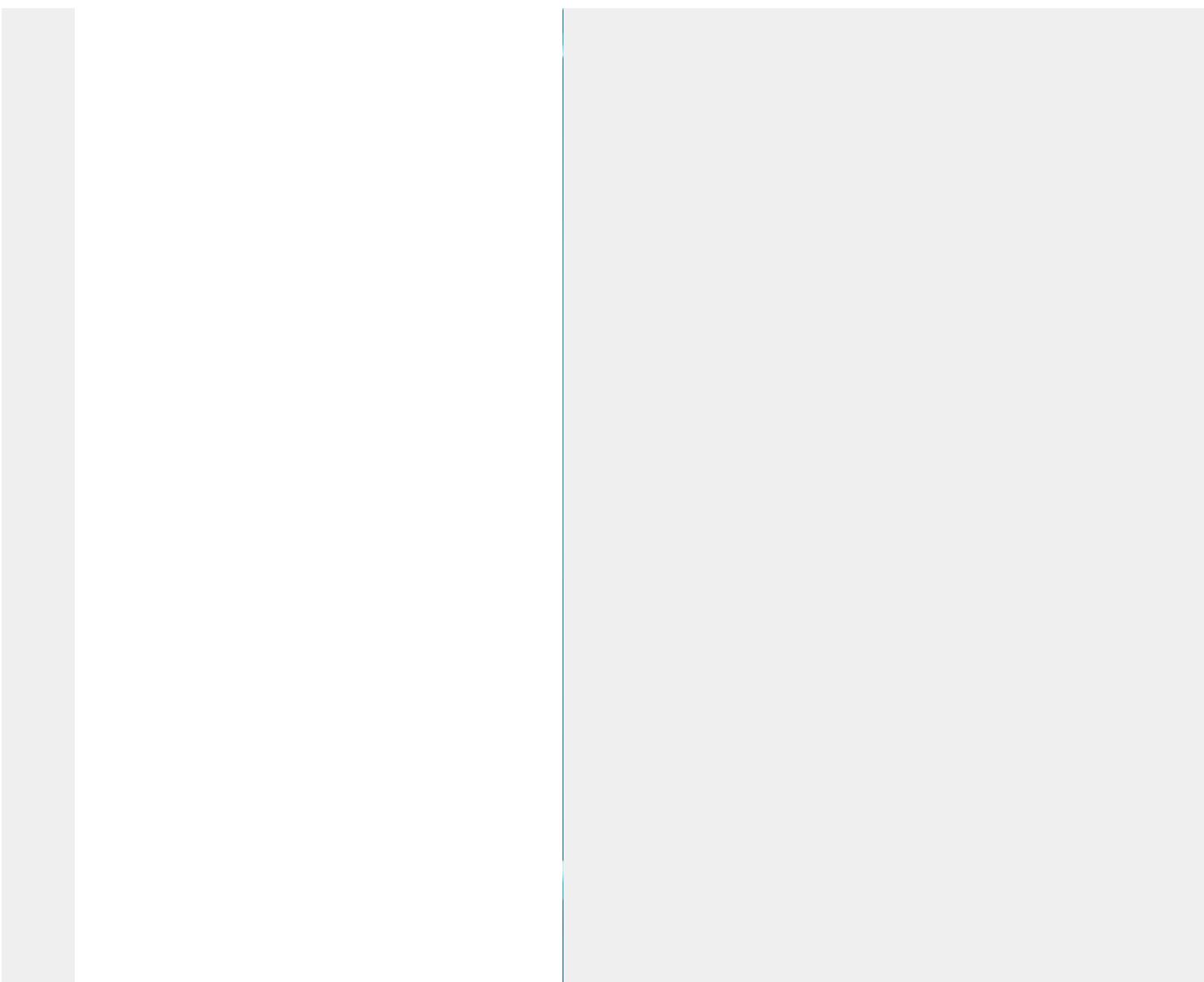
## Related Tutorials

- [CLEAR an Entire Sheet using VBA in Excel](#)
- [Copy and Move a Sheet in Excel using VBA](#)
- [COUNT Sheets using VBA in Excel](#)
- [DELETE a SHEET using VBA in Excel](#)
- [Hide & Unhide a Sheet using VBA in Excel](#)
- [PROTECT and UNPROTECT a Sheet using VBA in Excel](#)
  
- [Write a VBA Code to Create a New Sheet in Excel](#)
- [VBA Worksheet Object](#)
- [Activate a Sheet using VBA](#)
- [VBA Check IF a Sheet Exists](#)



The screenshot shows the Adobe Acrobat Pro landing page. At the top, there's a large orange banner with the Adobe logo and the text "Acrobat Pro" followed by "Acrobat's got it.". Below the banner is a "Try free" button. The main area features a smartphone displaying a document titled "PROPOSAL". The document contains a table with three rows and a total row, followed by a signature field labeled "SIGNATURE".





---

Free Online Excel Courses

Excel Blog



Terms

Policy



---

Describe Your Excel Skills in CV

Excel for Accountants

About - Contact

© 2023 Excel Champs - All Rights Reserved.

A RAPTIVE PARTNER SITE

