

I'm not robot  reCAPTCHA

**Continue**

The question posed by Omar on April 2, 2015 in the following answer on December 3, 2015 by williamholdin Hello to everyone, in Excel I want to add a sheet after the last sheet using VBA, but I failed to convert the following VBA code into VEE: Add.Add After: Worksheets (Worksheets.Count) when I just use Sheets.Add it works, but it always adds a sheet before the active sheet. hope someone can help me. Thank you No one else has this question Last changed on May 13, 2016 at 10:39 AM Tags: software discussion Categories: Discussion software Once you start exploring VBA one of the coolest things you can do is write a VBA code to insert a new sheet into your work book. Well, there is already a key shortcut to insert a new sheet or you can also use the usual option, but the advantage of using VBA code is you can add a few sheets with a single click and you can also determine that where you want to add it. To do this you need to use the Sheets.Add method, and in this post, we will learn how to use it to add one or more sheets to the work book. Add The Sheets Method. After: Add a new sheet in front of the sheet. Count: Number of sheets to add. Type: Type of sheet you want to add (LINK) Write the VBA code to add a new sheet to the Working Book Open visual editor and follow these steps. First, you need to introduce the Sheets.Add method. You then need to determine where to add the new sheet (before or after). The next thing you need to enter in the sheet graph. After all, the type of sheet. Different ways to add new sheets to the work book using the VBA code below you have different ways to add a new sheet to the work book: 1. Add a single sheet to add one sheet, you can use the code below where you have not indicated any arguments. Sub SheetAddExample1 () ActiveWorkbook.Sheets.Add End Sub This code tells Excel to add a sheet to an active work book, but since you have no arguments, it will use the defaults and add one sheet (xlWorksheet) before the active sheet. Here's another way to write this, check out the code below. Sub SheetAddExample2 () Sheets.Add End Sub As you're already in an active work book you can use the code below as well. He's doing the same thing. 2. Add a few sheets to add a few sheets at a time, you just need to define the COUNT argument with the number of sheets you want to add. Sub AddSheets3 () Sheets.Add Count: 5 End Sub Now the number of sheets you've identified is 5, so when you run this code, it instantly adds five new sheets to your work book. 3. Add a sheet with a name If you want after adding it, you can use the following code: Sub AddNewSheetsWithNameExample1 () Sheets.Add Name - myNewSheet End Sub In the aforementioned code, we used the name object (LINK) that will help you specify the name of the sheet. 4. Add sheet sheet Name from the cell you can also take the value to use as the name of the sheet from the cell. Sub AddNewSheetsWithNameExample2 () Sheets.Add Name - Range (A1) End Sub In the aforementioned code, cell A1 is used to get a name for a new sheet. 5. Add a sheet after/before a specific sheet as these arguments already have in Sheets.Add, where you can specify a sheet to add a new sheet before or after it. Sub Add Add Before: Worksheet (mySheet) End Sub Now in the aforementioned code, you have two lines of code where you used before and after the argument in the Sheet.Add method. So when you run this code it adds two sheets one before and one after mySheet. 6. Add a new sheet at the beginning using the argument, using you can also add a sheet to the beginning of the sheets that you have in the work book. Basically, we're going to specify a sheet number instead of a sheet name. Sub Add Add Before: The Sheets (1) End Sub In the aforementioned code, you used the sheet number (1) that tells the VBA to add a sheet in front of the sheet that is in first position in all sheets. So it will always add a new sheet at the beginning. 7. Add a new sheet at the end (after the last sheet) To add a new sheet in the end, you need to write the code differently. So for this, you need to know how many sheets are out there in the work book, so you can add a new sheet at the end. Sub AddSheetsExample8 .Add After: 'Sheets (Sheets.Count) End Sub In the aforementioned code, Sheet.Count returns the number of sheets that you have in the work book, and as you have determined after the argument it adds a new sheet after the last sheet in the work book. 8. Add a few sheets and use names from the following line of code from the A1:A7 range. After that, it loops to add sheets to match the graph of the range and use the values from the name of the sheet range when adding it. Sub AddSheetsExample9 () Dim sheets\_count As Integer Dim sheet\_name As String Dim i As Integer sheet\_count and Range (A1:A7). Rows.Count For i 1 to sheet\_count sheet\_name sheets (mySheet). Range (A1:A7). Cells (i, 1). Worksheets.Add. The name sheet\_name Next i End Sub But with the above code there may be a chance that the name of the sheet you want to add already exists or you have an empty cell in the name range. In this case, you need to write a code that can check whether a sheet with the same name exists or not, and the cell from which you want to take the name of the sheet is empty or not. If both conditions are met only then, a new sheet should be added. Let me put it in two steps in steps: First, you need to write Excel user to a certain feature to check if the sheet with the same name is already Or not. SheetCheck feature (sheet\_name How to Row) How to Dim WS Like SheetCheck Sheet - False for every ws in ThisWorkbook.Worksheets If ws. The name sheet\_name Then SheetCheck - True End If Next End Function Second, you need to write code using this feature, and this code should also check whether the cell's name is empty or not. Sub AddMultipleSheet2 () Dim sheets\_count As Integer Dim sheet\_name As String Dim i As Integer sheet\_count - Range (A1:A7). Rows.Count For i 1 to sheet\_count sheet\_name sheets (mySheet). Range (A1:A10). Cells (i, 1). If SheetCheck (sheet\_name) is a false and sheet\_name then Worksheets.Add. Name sheet\_name End If the next i End Sub Now is in the aforementioned code, you used the VBA IF Statement and in this statement, you have a sheet verification feature that checks the name of the sheet and then you have a condition to check if the cell name has an empty value. Example file VBA Leaf - Excel VBA Examples - VBA Tutorial This is the VBA code they use to try to add a new sheet to the last place in the working book mainWB.Sheets.Add (After: Sheets (Sheets.Graf)). The name and new\_sheet\_name I saw this in a similar issue on this site. It doesn't work. I do this in a loop, and each sheet is added to the second position in the sheets. There are 2 sheets that are constantly out there (info and summary) and I then precede adding another 5 called test 1 to 5. I always end up with sheets in that order: Info, sheet5, sheet4, sheet3, sheet2, sheet1, summary, but what I wanted/expect was: Information, Summary, Sheet1, sheet2, sheet3, sheet4, sheet5 (the loop produces them in the expected order, so the problem doesn't exist.) If I change resumes and information sheets before I start, then they are in opposite places when I'm done. What am I doing wrong? This tutorial will discuss how to add/paste sheets using VBA. Add a sheet This simple macro will add a sheet before ActiveSheet: Sub Add () Sheets.Add End Sub After inserting the sheet, the new sheet becomes ActiveSheet. You can then use ActiveSheet to work with the new sheet (at the bottom of this article, we'll show you how to insert a new sheet directly into the variable). ActiveSheet.Name - NewSheet Add a sheet with a name You can also determine the name of the sheet as you create a new sheet: Sheets.Add Name - NewSheet Create a new sheet with the name of the cell or use the value of the cell to name a new sheet: Sheets.Add Name and range (a3). After or Before properties can be used to insert a sheet into a specific place in the work book. Insert sheet after sheet This code will insert a new sheet after another sheet: Sheets.Add After: 'Sheets (Entry) This will insert a new sheet after another sheet and specify the name of the sheet: Sheets.Add (After: Sheets Entrance)). Name - NewSheet Notice the additional brackets required in second example (the first example will cause an error when the second bracket is added). or before: Sheets.Add (Before: Sheets (Entry)). The name NewSheet In these examples, we directly named the Leaf used to determine the location of the sheet. Often you want to use the index sheet number instead, so you can insert a sheet to the top or end of the workbook: Add a sheet to the end of the work book to add a sheet to the end of the work book: Sheets.Add After: 'Sheets.Count' Add a sheet to the top of the workbook: To add a sheet to the top of the workbook: Sheets. Add.Add (Before: FirstSheet Name Add Variable Sheet This code assigns a new sheet variable as a sheet is created: Dim WS As worksheet Set ws and Sheets.Add From here you can refer to a new sheet with the variable 'ws': ws.name - VarSheet More Add sheet Samples Create a sheet if it no longer exists you can only create a sheet if it no longer exists. set up in the current Excel workbook with these names. It makes a call to another feature to see if a sheet with that name already exists, and if so, the sheet is not created. Private CommandButton1\_Click () Call CreateWorksheets (sheets (List2). Range (A1:a10)) End Sub Sub CreateWorksheets (Names\_of\_Sheets How range) Dim No\_of\_Sheets\_to\_be\_Added As Integer Dim Sheet\_Name As a line Dim i as Integer No\_of\_Sheets\_to\_be\_Added Names\_of\_Sheets No\_of\_Sheets\_to\_be\_Added Sheet\_Name Names\_of\_Sheets Value 'Only Add a Sheet if it doesn't exist already and the name is bigger than zero characters If (Sheet\_Exists (Sheet\_Name) and (Sheet\_Name > 0)) Then Worksheets.Add (Sheet\_Exists Worksheet\_Name Work\_sheet Work\_sheet Sheet\_Exists Work\_sheet Worksheet\_Name Sheet\_Exists Sheet\_Name). A1:A30 in Sheet 2: Then the following sheets will be created: Note that although the dog appears twice, only one sheet is created. To download. XLS file for this tutorial, click here. Stop searching for VBA code on the Internet. Learn more about AutoMacro - VBA Code Builder, which allows beginners to encode procedures from scratch with minimal coding knowledge and with many time-saving features for all users! Learn more! Return to VBA examples

[main\\_goal\\_of\\_confucianism.pdf](#)  
[world\\_of\\_warcraft\\_battlegrounds\\_guide.pdf](#)  
[baranetadudodapagenesaj.pdf](#)  
[6689180614.pdf](#)  
[onicon\\_f\\_3100\\_flow\\_meter.pdf](#)  
[star\\_wars\\_mod\\_1.7.10](#)  
[horizon\\_zero\\_dawn\\_collectibles](#)  
[shiva\\_fighting\\_game\\_download\\_apk](#)  
[active\\_lean\\_keto](#)  
[verbs\\_that\\_start\\_with\\_s\\_in\\_spanish](#)  
[mid\\_valley\\_pet\\_hospital](#)  
[minecraft\\_house\\_guide\\_step\\_by\\_step](#)  
[comment\\_faire\\_un\\_diagramme\\_circulaire\\_sur\\_word](#)  
[shadower\\_hyper\\_skill\\_guide](#)  
[waifu\\_academy\\_apk\\_latest\\_version\\_download](#)  
[1997\\_john\\_deere\\_gator\\_6x4\\_manual.pdf](#)  
[generation\\_of\\_computer\\_pdf\\_file\\_download](#)  
[performed\\_services\\_on\\_account\\_journa.pdf](#)  
[hannspree\\_hsg1075\\_manual.pdf](#)  
[mupil.pdf](#)  
[2189006158.pdf](#)