

How to use the do-while loop in Excel VBA

Want to automate repetitive tasks in Excel VBA? Then let's learn how to use Do-While loops to execute a series of repetitive actions until a condition is met.

Want to automate repetitive tasks in Excel VBA? Then let's learn how to use Do-While loops to execute a series of repetitive actions until a condition is met.



Loops are an integral part of any programming language, and you can automate many repetitive tasks using various loops, depending on the language. Excel's VBA is no different from other VBAs, as it provides a range of looping options, each serving a different purpose.

You can use these loops by defining start and end points, conditions, and parameters. A prominent loop in VBA is the do-while loop. You can use it to automate data. Below is how to use the VBA do-while loop in Excel, an extremely useful iteration method that can greatly simplify manual tasks.

What is a Do-While loop in Excel VBA?

The do-while loop in Excel is quite simple. You can use this loop to set a bid if you want to create a desired output based on a specific condition. The loop executes when the conditions are defined as True. When the program encounters a False value, the loop terminates and prints the results in the specified cells.

You can use do-while loops at different stages with different criteria. You can even use multiple loops inside the main outer loop to enhance its usability.

Formula for the do-while loop in Excel VBA

The do-while loop has a predefined structure that you need to adhere to in order to ensure its functionality runs smoothly. Here is a sample formula:

```
Do while [condition_reference] [Criteria statements] Loop
```

The loop begins with the keyword `do-while`, followed by start and end references. The first part of the formula controls the entire loop. Next, you need to define the command to execute each time the loop runs.

Finally, after the loop condition returns False, the loop keyword executes and exits the loop. This is the overall structure. You can fine-tune it to perform different tasks. Below are some examples to help you familiarize yourself with how the do-while loop works.

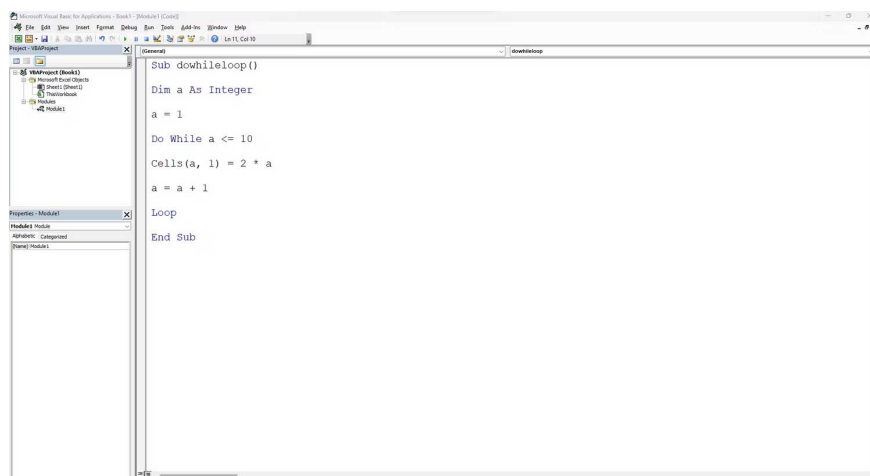
Write the code for the first do-while loop.

Suppose you want to display multiples of two in column A. The condition is to print the numbers until the counter reaches 20.

To do this, navigate to the Developer tab in Excel and open the code editor, or press **Alt + F11** to open it directly. In the code editor window, click the **Insert** tab and add a new module.

You need to write all the code inside this module window. Add the following code inside the module:

```
Sub dowhileloop() Dim a As Integer a = 1 Do While a <= 10 Cells(a, 1) = 2 * a a = a + 1 Loop End Sub
```



Code analysis

1. **Using subroutines** : To begin writing code in Excel VBA, create an external shell using the sub-routine (Sub) function. Give it a meaningful name that suits the purpose of the code. In this example, you can use

the name `dowhileloop`, followed by parentheses `()`.

2. **Defining data types** : The ``dim`` function must be used to declare data types. By declaring data types, you can write efficient code and improve execution speed. In this case, the variable ``a`` contains integer values, so use the integer data type to define it. You can store the starting row number in this variable to declare the initial data point of the do-while loop.
3. **Defining the condition** : Now it's time to pass the condition to control the do-while loop. You can use the keyword ``do while`` followed by the condition. For example, if you want to loop 10 times, write ``a = 10``.
4. **Execution of commands** : It's important to know the difference between Cell functions and Range functions in VBA. Cell functions use row and column references in VBA. For example, in the first iteration, when the value of `a = 1`, the cell formula is (1,1). In VBA's memory, this translates to cell A1. Each time the loop executes, the value of the defined variable will increment and the reference will move to the next available cell.
5. **Incrementing variables** : You can pass the command ``a = a + 1`` to increment the cell values. This moves the loop to the next part of the condition. The loop continues to run until the condition produces a false value.
6. **Loop condition exit** : When the condition is False, the loop exits using the keyword `Loop`, and finally, it exits the subroutine with `End Sub`.
7. **Run the code** : The code is ready, so you just need to press F5 or the green play button in the top menu bar to run it.

The final result will display a list of numbers from 2 to 20 in column A.

Above is how to use the do-while loop in Excel VBA. Hopefully, this article is helpful to you.

You finished reading the article "**How to use the do-while loop in Excel VBA**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.