## Example

## While loop

Private Sub Constant\_demo\_Click()

Dim Counter : Counter = 10

While Counter < 15 ' Test value of Counter.

Counter = Counter + 1 ' Increment Counter.

msgbox "The Current Value of the Counter is : " & Counter

Wend ' While loop exits if Counter Value becomes 15.

End Sub

Example

Add a button and add the following function.

Private Sub Constant\_demo\_Click()

Dim a As Integer

a = 10

For i = 0 To a Step 2

MsgBox "The value is i is : " & i

Next

End Sub

## Example

Private Sub Constant\_demo\_Click()

'fruits is an array

fruits = Array("apple", "orange", "cherries")

Dim fruitnames As Variant

'iterating using For each loop.

For Each Item In fruits

fruitnames = fruitnames & Item & Chr(10)

Next

MsgBox fruitnames

End Sub

## Example

The following example uses **Do…while** loop to check the condition at the beginning of the loop. The statements inside the loop are executed, only if the condition becomes True.

Private Sub Constant\_demo\_Click()

Do While i < 5

i = i + 1

msgbox "The value of i is : " & i

Loop

End Sub

## Example

The following example uses **Do…Until** loop to check the condition at the beginning of the loop. The statements inside the loop are executed only if the condition is false. It exits out of the loop, when the condition becomes true.

Private Sub Constant\_demo\_Click()

i = 10

Do Until i>15 'Condition is False.Hence loop will be executed

i = i + 1

msgbox ("The value of i is : " & i)

Loop

End Sub

Debug:

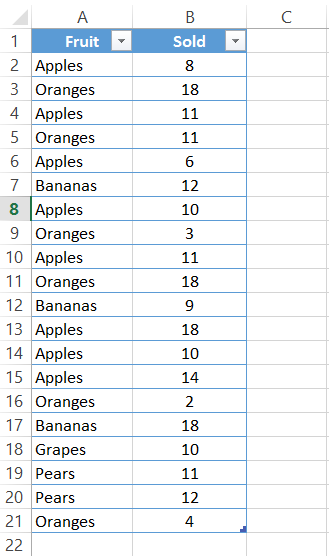
However, using a loop we only need to write Debug.Print once.

**For** i = 1 **To** 20

**Debug.Print** i

**Next** i

The following screenshot shows an example of this list

[](https://excelmacromastery.com/wp-content/uploads/2015/02/Fruit-Loops.png)

**Sample Data of Fruit Sales**

   
We can use the code to count the oranges

**Sub** CountFruit()

' Get the last row with text

**Dim** LastRow **As** **Long**

LastRow = Sheet1.Cells(Sheet1.Rows.Count, 1).End(xlUp).Row

**Dim** i **As** Long, Total **As** **Long**

' Use LastRow in loop

**For** i = 2 **To** LastRow

' Check if cell has text "Orange"

**If** Sheet1.Cells(i, 1).Value = "Oranges" **Then**

' Add value in column B to total

Total = Total + Sheet1.Cells(i, 2).Value

**End** **If**

**Next** i

' Print total

**Debug.Print** "Total oranges sold was:"; Total

**End** **Sub**

## Using the For Loop with a Collection

The For loop can also be used to read items in a [Collection](https://excelmacromastery.com/excel-vba-collections/).

   
In the following example, we display the name of all the open workbooks

**Dim** i **As** **Long**

**For** i = 1 **To** Workbooks.Count

**Debug.Print** Workbooks(i).FullName

**Next** i

## Using Nested For Loops

Sometimes you may want to use a loop within a loop. An example of this would be where you want to print the names of the [worksheets](https://excelmacromastery.com/excel-vba-worksheet/)of each open [workbook](https://excelmacromastery.com/excel-vba-workbook/).

The first loop would go through each workbook. Each time this loop runs it would use a second loop to go through all the worksheets of that workbook. It is actually much easier to do than it sounds.

   
The following code shows how

**Sub** ListWorksheets()

**Dim** i **As** Long, j **As** **Long**

' First Loop goes through all workbooks

**For** i = 1 **To** Workbooks.Count

' Second loop goes through all the worksheets of workbook(i)

**For** j = 1 **To** Workbooks(i).Worksheets.Count

**Debug.Print** Workbooks(i).Name + ":" + Worksheets(j).Name

**Next** j

**Next** i

**End** **Sub**

## Order of Items

For Each goes through items in one way only.

For example, if you go through all the worksheets in a workbook it will always go through from left to right. If you go through a range it will start at the lowest cell e.g. Range(“A1:A10”) will return A1,A2,A3 etc.

This means if you want any other order then you need to use the **For** loop.

   
Both loops in the following example will read the worksheets from left to right.

' Both loops read the worksheets from left to right

**Dim** wk **As** Worksheet

**For** **Each** wk **In** ThisWorkbook.Worksheets

**Debug.Print** wk.Name

**Next**

**Dim** i **As** **Long**

**For** i = 1 **To** ThisWorkbook.Worksheets.Count

**Debug.Print** ThisWorkbook.Worksheets(i).Name

**Next**

   
As you can see the For Each loop is neater to write. However if you want to read the sheets in any other order e.g. right to left then you have to use the for loop.

' Reading the worksheets from right to left

**Dim** i **As** **Long**

**For** i = ThisWorkbook.Worksheets.Count **To** 1 **Step** -1

**Debug.Print** ThisWorkbook.Worksheets(i).Name

**Next**

## For Each With Arrays

One thing to keep in my is that the **For Each** loop is that it is read-only when you use it with arrays.

   
The following example demonstrates this

**Sub** UseForEach()

' Create array and add three values

**Dim** arr() **As** **Variant**

arr = Array("A", "B", "C")

**Dim** s **As** **Variant**

**For** **Each** s **In** arr

' Changes what s is referring to - not value of array item

s = "Z"

**Next**

' Print items to show the array has remained unchanged

**For** **Each** s **In** arr

**Debug.Print** s

**Next**

**End** **Sub**

When we use the For Loop we can change the array item. If we change the previous code to use the **For** Loop you it will change all the array values to “Z”

**Sub** UsingForWithArray()

' Create array and add three values

**Dim** arr() **As** **Variant**

arr = Array("A", "B", "C")

**Dim** i **As** **Long**

**For** i = LBound(arr) **To** UBound(arr)

' Changes value at position to Z

arr(i) = "Z"

**Next**

' Print items to show the array values have change

**For** i = LBound(arr) **To** UBound(arr)

**Debug.Print** arr(i)

**Next**

**End** **Sub**

## Nested For Each Loop

We saw already that you can have a loop inside other loops. Here is the example from above

**Sub** ListWorksheets()

**Dim** i **As** Long, j **As** **Long**

' First Loop goes through all workbooks

**For** i = 1 **To** Workbooks.Count

' Second loop goes through all the worksheets of workbook(i)

**For** j = 1 **To** Workbooks(i).Worksheets.Count

**Debug.Print** Workbooks(i).Name + ":" + Worksheets(j).Name

**Next** j

**Next** i

**End** **Sub**

This time we will use the **For Each** loop to perform the same task

**Sub** ReadAllWorksheets()

**Dim** wk **As** Workbook, sh **As** Worksheet

' Read each workbook

**For** **Each** wk **In** Workbooks

' Read each worksheet in the wk workbook

**For** **Each** sh **In** wk.Worksheets

' Print workbook name and worksheet name

**Debug.Print** wk.Name + ": " + sh.Name

**Next** sh

**Next** wk

**End** **Sub**