

User Defined Functions in VBA Cheat Sheet

Introduction to User Defined Functions (UDFs)

User Defined Functions (UDFs) are custom functions that you can create in VBA to perform specific tasks or calculations that are not readily available through built-in Excel functions. UDFs allow you to extend the capabilities of Excel by writing your own functions.

Creating a UDF

To create a User Defined Function in VBA, follow these steps:

1. Open the Visual Basic for Applications (VBA) editor by pressing Alt + F11.
2. Click Insert > Module to insert a new module.
3. Write your UDF using the following syntax:

Function functionName(parameter1 As Type, parameter2 As Type, ...) As ReturnType

 ' Function body: Write your code here

 functionName = resultValue ' Assign the value you want to return

End Function

Parameters

functionName: The name of your UDF.

parameter1, parameter2, ...: Input parameters that the function takes.

Type: Data type of the parameters.

ReturnType: Data type of the value the function will return.

resultValue: The value that the UDF will return.

Example UDF

Function AddNumbers(num1 As Double, num2 As Double) As Double

 AddNumbers = num1 + num2

End Function

Using UDFs in Excel

After creating the UDF, save the module and return to the Excel workbook. You can now use the UDF like any other Excel function. In a cell, type `=functionName(parameter1, parameter2)`.

Tips for Creating Effective UDFs

Use Descriptive Names: Choose meaningful names for your UDFs that convey their purpose.

Error Handling: Implement proper error handling in your UDFs to handle unexpected situations.

Optimize Performance: Ensure your UDFs are efficient and don't slow down calculations.

Document Your UDFs: Add comments explaining how to use your UDFs and what they do.

Advanced UDF Techniques

Variable Number of Arguments: Use the `ParamArray` keyword to allow a varying number of arguments.

Worksheet Functions: UDFs can call built-in worksheet functions using `WorksheetFunction` object.

Arrays and Ranges: UDFs can accept and return arrays or ranges for batch processing.

Conclusion

User Defined Functions in VBA provide a powerful way to extend Excel's capabilities and perform custom calculations. By following the steps above and considering best practices, you can create effective and efficient UDFs that enhance your Excel experience.