

Using logical functions

Logical functions in Excel are incredibly useful for making decisions based on specific conditions. Here are some of the key logical functions:

IF Function:

This is the most commonly used logical function. It checks whether a condition is met and returns one value if true and another value if false.

Syntax: =IF(logical_test, value_if_true, value_if_false)

Example: =IF(A1>10, "Yes", "No") checks if the value in cell A1 is greater than 10. If true, it returns "Yes"; otherwise, it returns "No".

AND Function:

This function returns TRUE if all the conditions are true.

Syntax: =AND(logical1, logical2, ...)

Example: =AND(A1>10, B1<5) returns TRUE if both conditions are met.

OR Function:

This function returns TRUE if any of the conditions are true.

Syntax: =OR(logical1, logical2, ...)

Example: =OR(A1>10, B1<5) returns TRUE if either condition is met.

[The AND function](#)



IFS Function

The IFS function in Excel is a powerful tool that allows you to test multiple conditions and return a value corresponding to the first TRUE condition. It's a more efficient and readable alternative to using multiple nested IF statements. Here's a breakdown of how it works:

Syntax: =IFS(logical_test1, value_if_true1, [logical_test2, value_if_true2], ...)

Parameters

logical_test1, logical_test2, ...: These are the conditions you want to test.

value_if_true1, value_if_true2, ...: These are the values to return if the corresponding condition is TRUE.

Example

Suppose you want to assign grades based on scores:

=IFS(A1 >= 90, "A", A1 >= 80, "B", A1 >= 70, "C", A1 >= 60, "D", TRUE, "F")

In this example:

If the score in cell A1 is 90 or above, it returns "A".

If the score is between 80 and 89, it returns "B".

If the score is between 70 and 79, it returns "C".

If the score is between 60 and 69, it returns "D".

For any other score, it returns "F".

Key Points

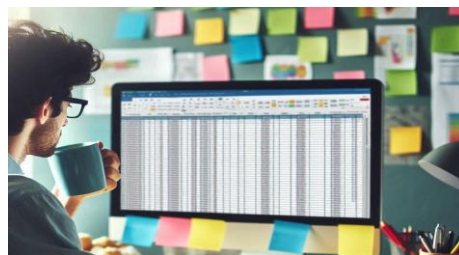
Order Matters: The IFS function evaluates conditions in the order they appear. Once a condition is TRUE, it stops evaluating further conditions.

Default Value: To specify a default value, use TRUE as the final logical test. This ensures that if none of the other conditions are met, the default value is returned.

Benefits

Readability: IFS functions are easier to read and manage compared to multiple nested IF statements.

Efficiency: It simplifies complex logical tests into a single function.





Nested IF with AND

You can nest the AND function inside an IF function to test multiple conditions. For example:

```
=IF(AND(A1>10, B1<5), "Yes", "No")
```

This formula checks if the value in cell A1 is greater than 10 and the value in cell B1 is less than 5. If both conditions are true, it returns "Yes"; otherwise, it returns "No".

Nested IF with OR

Similarly, you can nest the OR function inside an IF function to test multiple conditions. For example:

```
=IF(OR(A1>10, B1<5), "Yes", "No")
```

This formula checks if the value in cell A1 is greater than 10 or the value in cell B1 is less than 5. If either condition is true, it returns "Yes"; otherwise, it returns "No".

Combining IF, AND, and OR

You can also combine all three functions for more complex logic. For example:

```
=IF(AND(A1>10, OR(B1<5, C1=20)), "Yes", "No")
```

This formula checks if the value in cell A1 is greater than 10 and either the value in cell B1 is less than 5 or the value in cell C1 is equal to 20. If these conditions are met, it returns "Yes"; otherwise, it returns "No".

Working with Lookups

VLOOKUP

VLOOKUP (Vertical Lookup) searches for a value in the first column of a table and returns a value in the same row from a specified column. It's useful for finding data in a large table.

Syntax:

```
VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])
```

lookup_value: The value you want to search for.

table_array: The range of cells that contains the data.

col_index_num: The column number in the table from which to retrieve the value.

range_lookup: Optional. TRUE for an approximate match, FALSE for an exact match.

HLOOKUP

HLOOKUP (Horizontal Lookup) works similarly to VLOOKUP but searches for a value in the first row of a table and returns a value in the same column from a specified row.

Syntax:

```
HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])
```

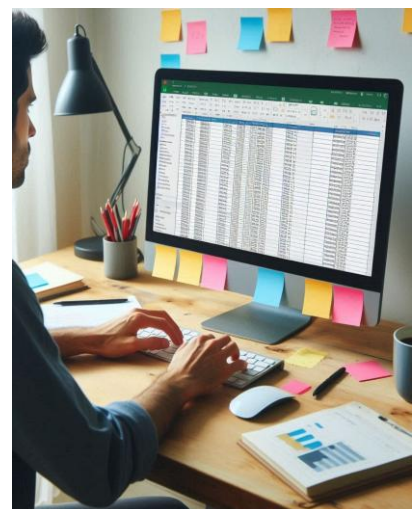
lookup_value: The value you want to search for.

table_array: The range of cells that contains the data.

row_index_num: The row number in the table from which to retrieve the value.

range_lookup: Optional. TRUE for an approximate match, FALSE for an exact match.

[excel-vlookup-hlookup-functions](#)



XLOOKUP

XLOOKUP is a more flexible and powerful function that can replace both VLOOKUP and HLOOKUP. It searches a range or an array and returns an item corresponding to the first match it finds. It can search both vertically and horizontally.

Syntax:

XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])

lookup_value: The value you want to search for.

lookup_array: The range or array to search.

return_array: The range or array to return.

if_not_found: Optional. The value to return if no match is found.

match_mode: Optional. Specifies the type of match (exact, next smaller, next larger).

search_mode: Optional. Specifies the search mode (first-to-last, last-to-first).

[excel-the-xlookup-function](#)

INDEX MATCH

INDEX and MATCH are often used together to perform more complex lookups. INDEX returns the value of a cell in a table based on the row and column number, while MATCH returns the relative position of a value in a range.

INDEX Syntax:

INDEX(array, row_num, [column_num])

array: The range of cells.

row_num: The row number in the array.

column_num: Optional. The column number in the array.

MATCH Syntax:

MATCH(lookup_value, lookup_array, [match_type])

lookup_value: The value you want to search for.

lookup_array: The range of cells to search.

match_type: Optional. 0 for an exact match, 1 for the largest value less than or equal to the lookup value, -1 for the smallest value greater than or equal to the lookup value.

When combined, INDEX MATCH can look up values in a table more flexibly than VLOOKUP or HLOOKUP



Data Validation

Data validation in MS Excel is a feature that allows you to control the type of data or values that users can enter into a cell. This helps maintain data accuracy and consistency. Here's a step-by-step guide on how to use it:

Select the Cell(s): Choose the cell or range of cells where you want to apply data validation.

Open Data Validation: Go to the Data tab on the Ribbon, then click on Data Validation.

Set Validation Criteria:

Settings Tab: Under the Allow dropdown, choose the type of data you want to allow (e.g., Whole Number, Decimal, List, Date, Time, Text Length, or Custom).

Data: Depending on your choice in the Allow dropdown, set the specific criteria (e.g., between, greater than, less than).

Input Message (Optional): On the Input Message tab, you can create a message that will appear when the cell is selected, guiding users on what data to enter.

Error Alert: On the Error Alert tab, you can customize the error message that appears if invalid data is entered. You can choose the style of the alert (Stop, Warning, or Information).

Example: Creating a Drop-Down List

Select the Cell(s): Choose the cell where you want the drop-down list.

Open Data Validation: Go to Data > Data Validation.

Settings Tab: Under Allow, select List.

Source: Enter the items for the drop-down list, separated by commas (e.g., Apple, Banana, Cherry).

Now, when users click on the cell, they will see a drop-down arrow and can select from the list you provided.

Data validation is a powerful tool to ensure data integrity and guide users in entering the correct information

[Data validation list](#)

[Create Dynamic Drop Down lists in Excel](#)

PivotTables and PivotCharts



PivotTables

A PivotTable is a powerful tool in Excel that allows you to summarize, analyze, explore, and present large amounts of data.

Here are the key features:

Summarization: You can quickly summarize data by categories and subcategories, using functions like sum, average, count, etc.

Data Analysis: PivotTables help in analyzing numerical data in detail and answering unanticipated questions about your data.

Flexibility: You can pivot (rotate) rows and columns to see different summaries of the source data.

Filtering and Sorting: Easily filter and sort data to focus on the most relevant information.

Drill Down: Expand and collapse data levels to see more or less detail as needed.

How to Create a PivotTable

Select Your Data: Ensure your data is in a tabular format with no blank rows or columns.

Insert PivotTable: Go to the Insert tab and click on PivotTable. Choose where you want the PivotTable to be placed.

Arrange Fields: Drag fields to the Rows, Columns, Values, and Filters areas to organize your data.

PivotCharts

A PivotChart is a graphical representation of a PivotTable. It provides a visual way to see comparisons, patterns, and trends in your data.

Dynamic Visualization: PivotCharts update automatically when the associated PivotTable is changed.

Interactivity: You can filter and sort data directly within the chart.

Customization: Customize the chart type, layout, and style to best represent your data.

How to Create a PivotChart

Create a PivotTable: Follow the steps above to create a PivotTable.

Insert PivotChart: With the PivotTable selected, go to the Insert tab and choose a chart type from the Charts group.

Customize: Use the PivotChart tools to customize the chart's appearance and functionality.

Example

Imagine you have sales data for different products across various regions. A PivotTable can help you quickly summarize total sales by region and product. A PivotChart can then visually display this summary, making it easier to spot trends and patterns.

Using analytical options

What-If Analysis in MS Excel is a powerful feature that allows you to explore different scenarios and their potential outcomes by changing the values in your formulas. This can be incredibly useful for decision-making and forecasting. There are three main tools within What-If Analysis:

Scenario Manager:

Purpose: Allows you to create and save different groups of values, or "scenarios," and switch between them to see how they affect your results.

Example: You could create scenarios for best-case, worst-case, and most-likely sales forecasts and compare the outcomes.

Goal Seek:

Purpose: Helps you find the necessary input value to achieve a specific goal or result.

Example: If you know the desired profit and need to find out the required sales volume to achieve it, Goal Seek can calculate that for you.

Data Table:

Purpose: Allows you to see the effects of changing one or two variables on a range of outcomes.

Example: You can create a table that shows how different interest rates and loan amounts affect monthly payments.

[what-if-analysis-in-excel](#)

Data Tables		Projected Projects ->									
		20	40	60	80	100	120	140	160	180	200
A R E V E N U E	€ 25,000.00	€ 1,050,000.00	€ 600,000.00	€ 150,000.00	€ 300,000.00	€ 750,000.00	€ 1,200,000.00	€ 1,650,000.00	€ 2,100,000.00	€ 2,550,000.00	€ 3,000,000.00
	€ 30,000.00	€ 950,000.00	€ 400,000.00	€ 150,000.00	€ 700,000.00	€ 1,250,000.00	€ 1,800,000.00	€ 2,350,000.00	€ 2,900,000.00	€ 3,450,000.00	€ 4,000,000.00
	€ 35,000.00	€ 850,000.00	€ 200,000.00	€ 450,000.00	€ 1,100,000.00	€ 1,750,000.00	€ 2,400,000.00	€ 3,050,000.00	€ 3,700,000.00	€ 4,350,000.00	€ 5,000,000.00
	€ 40,000.00	€ 750,000.00	€ -	€ 750,000.00	€ 1,500,000.00	€ 2,250,000.00	€ 3,000,000.00	€ 3,750,000.00	€ 4,500,000.00	€ 5,250,000.00	€ 6,000,000.00
	€ 45,000.00	€ 650,000.00	€ 200,000.00	€ 1,050,000.00	€ 1,900,000.00	€ 2,750,000.00	€ 3,600,000.00	€ 4,450,000.00	€ 5,300,000.00	€ 6,150,000.00	€ 7,000,000.00
	€ 50,000.00	€ 550,000.00	€ 400,000.00	€ 1,350,000.00	€ 2,300,000.00	€ 3,250,000.00	€ 4,200,000.00	€ 5,150,000.00	€ 6,100,000.00	€ 7,050,000.00	€ 8,000,000.00
	€ 55,000.00	€ 450,000.00	€ 600,000.00	€ 1,650,000.00	€ 2,700,000.00	€ 3,750,000.00	€ 4,800,000.00	€ 5,850,000.00	€ 6,900,000.00	€ 7,950,000.00	€ 9,000,000.00
	€ 60,000.00	€ 350,000.00	€ 800,000.00	€ 1,950,000.00	€ 3,100,000.00	€ 4,250,000.00	€ 5,400,000.00	€ 6,550,000.00	€ 7,700,000.00	€ 8,850,000.00	€ 10,000,000.00

Projected 2021

Projected Projects	Average Revenue	Total Average Revenue
100	€ 25,000.00	€ 2,500,000.00

Project Staff

Average Salary	Total Salary	
20	€ 50,000.00	€ 1,000,000.00

Fixed Project Costs

€ 500,000.00

Variable Cost Per Project

	First Class	Economy
Average Travel Cost	€ 2,400.00	€ 1,100.00
Hotel	€ 2,000.00	€ 900.00
Food	€ 500.00	€ 500.00
Total	€ 4,900,000.00	€ 2,500,000.00

Profit First Class

€ 510,000.00

Profit Economy

€ 750,000.00

Scenario Manager

Scenarios:
Best Case
Worst Case

add...
Delete
Edit...
Merge...
Summary...

Changing cells: \$A\$4:\$B\$4,\$A\$7:\$B\$7,\$B\$13:\$C\$15
Comment: Created by Jens Bonde on 06/08/2024

ShowClose

Macros

Excel Macros are a powerful feature that allows you to automate repetitive tasks in Excel. Here's a breakdown of what they are and how you can use them:

What are Excel Macros?

Macros are a series of recorded actions or commands that you can play back to automate tasks. Think of them as small programs that can perform tasks for you.

They are particularly useful for tasks you perform frequently, such as formatting data, creating reports, or performing calculations.

How to Create a Macro

Enable the Developer Tab:

Go to File > Options > Customize Ribbon and check the Developer option.

Record a Macro:

Click on the Developer tab, then click Record Macro.

Give your macro a name, assign a shortcut key if desired, and add a description.

Perform the actions you want to automate. Excel will record your mouse clicks and keystrokes.

Click Stop Recording when you're done.

Run the Macro:

You can run the macro by pressing the assigned shortcut key or by going to the Developer tab and selecting Macros, then choosing the macro you want to run.

Editing Macros

Macros are written in Visual Basic for Applications (VBA). You can edit them by going to the Developer tab, clicking Macros, selecting the macro, and clicking Edit. This will open the VBA editor where you can modify the code¹².

Benefits of Using Macros

Time-Saving: Automate repetitive tasks to save time.

Consistency: Ensure tasks are performed the same way every time.

Efficiency: Reduce the chance of human error.

[saving-time-with-macros](#)

**Ask questions on our
post course learning
support forum**

Log in using your email
and your post course
email when you
completed the feedback

