Logical Functions

The logical functions involve a test, to which the outcome can only be true or false. You can then specify different results from the function depending on the outcome of the test.

The IF Function

The IF function uses the arguments:

=IF(Logical_test, Value_if_true, Value_if_false)

In the example below, the aim is to give staff whose actual is greater than their target, a commission of 10% of their actual. If they haven’t hit their target they will get 0.

The logical test would, therefore, have to compare the actual against the target and evaluate whether it was larger. If TRUE, the formula must return 10% of actual, but if it is FALSE it must return 0. The resulting function is:

=IF(D2>A2,C2*10%,0)

The result of the function is: 70.00

AND Function

The AND function lets you verify whether all of a number of tests are true. It is often used in the Logical_test argument of an IF function to evaluate multiple conditions.

In the example below, the aim is only to give commission to staff who have sold (C2) more than 25000 and have a status of A (D2). In all other cases they get 0.

The resulting function is:

=IF(AND(C2>25000,D2="A"),C2*10%,0)

The result of the function is: 0

OR Function

The OR function lets you verify whether any of a number of tests are true. It is often used in the Logical_test argument of an IF function to evaluate multiple conditions.

In the example above, the aim is to give staff who have sold (C2) more than 25000 or have a status of A (D2) a bonus of 5% of their actual. If they haven’t reached their target or their status is not A then they get 0.

The resulting function is:

=IF(OR(C2>25000,D2="A"),C2*5%,0)

The result of the function is: 0

Nested IF Functions

If you wish to have a number of tests performed consecutively on your data you can nest IF functions. In the example below, if the status (D2) is A then the bonus rate is 10%, if the status is B then the rate is 5% and if the status has any other value, the rate is 2%.

The resulting function is:

=IF(D2="A",C2*5%,IF(D2="B",C2*10%,C2*2%))

TIP: You can use up to 64 levels of nesting. In the example above, the second IF function is a second level function nested inside the false part of the initial IF function.

Lookup Functions

When you need to cross refer to a data table, you can make use of the VLOOKUP or the HLOOKUP functions. Their purpose is the same – to match data from the first column (VLOOKUP) or row (HLOOKUP) of a data table and then return the value for an adjoining column/row in the table.

In the example above, the aim is to find the price for the ProductID shown in A3 (lookup_value). The price list is in A14:C120. Because the data is presented in rows, VLOOKUP will be used. The prices of these products are listed in A14:C120 so the reference table is A14:C120 (table_array), with the ProductID shown in A3 (lookup_value). The function will return the words True or False.

TIP: Do not use thousand separators (,) in large numbers as commas are used to divide up the function’s arguments.

TIP: If you wish to test if a cell contains a certain piece of text, or wish to put text into a cell as an outcome of the function, place the text inside quotation marks, eg. “Yes”

TIP: If you don’t give a Value_If_True or Value_If_False argument the function will return the words True or False.
The COUNTIFS Function

Sometimes you may want to count items in a table or list where there is a match in two or more equivalent columns. This is where the COUNTIFS function can be used. The syntax of the COUNTIFS function is:

=COUNTIFS(criteria_range1, criteria1, criteria_range2, criteria2)

- criteria_range1 and criteria_range2 are the ranges where you want to do the counting.
- criteria1 and criteria2 is what you want to count in each range.

In the example above, the aim is to count the number of rows where London is in the Area column and M is in the Status column.

The resulting function is:

=COUNTIFS(B4:B10,”London”,C4:C10,”M”)