

How to Display “Yes” or “No” Based on Text Values Using the IF Function in Excel

Authored by
stats writer

November 21, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Display “Yes” or “No” Based on Text Values Using the IF Function in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99080>

The IF function is one of the most fundamental and powerful tools in Excel, allowing users to perform conditional logic based on specified criteria. When working with textual data, the IF function evaluates whether a cell meets a certain text condition. For instance, a common application involves setting up the formula to check if a condition is met (the logical test argument). If this condition evaluates to true, the function returns a designated text string, such as "Yes" (the `value_if_true` argument); if the condition is false, it returns an alternative string, such as "No" (the `value_if_false` argument). This binary outcome is essential for data classification and reporting, providing clear feedback on whether the target cell contains the required text value.

Leveraging the IF Function for Text Evaluation in Excel

The core utility of the IF function lies in its ability to execute logical tests and return different values based on the result. When dealing specifically with text values, proper syntax management, including the use of quotation marks around text strings, is paramount. We can structure conditional checks in several sophisticated ways, depending on whether we need an exact match, a partial text search, or a check against multiple possible text values simultaneously. The following methods outline the three primary ways to utilize the IF function effectively when processing textual data within your spreadsheets.

Method 1: Checking for Exact Text Equality

The simplest application of the IF function with text involves checking if the content of a specific cell is exactly equal to a predefined string. This requires setting the logical test to directly compare the cell reference against the desired text value. It is crucial to remember that this comparison is case-insensitive in standard Excel comparison operations, but for absolute accuracy, ensuring consistent data entry is always recommended. This method is highly effective for categorizing data based on precise labels.

The formula structure below demonstrates this exact match methodology. If the cell reference, in this case, **A2**, contains the precise text "Starting Center", the function proceeds to the `value_if_true` argument and returns "Yes". Should the cell contain any other value, including variations in spelling or extra spaces, the function executes the `value_if_false` argument, resulting in "No".

=IF(A2="Starting Center", "Yes", "No")

This formula provides a straightforward Boolean outcome: it returns "Yes" only if the value in cell **A2** is identically "Starting Center"--otherwise, it returns "No." This forms the basis for more complex text analyses.

Method 2: Checking if a Cell Contains Specific Substring Text

Often, the requirement is not to check for an exact match, but rather to determine if a cell contains a specific substring within a longer string of text. For this complex evaluation, we must incorporate helper functions like SEARCH and ISNUMBER within the IF function's logical test. The SEARCH function attempts to locate the specified substring and returns the starting numerical position of that text if found; otherwise, it returns a #VALUE! error.

To convert the numerical output or error state into a simple TRUE or FALSE logical test, we wrap the SEARCH function within the ISNUMBER function. ISNUMBER returns TRUE if the result of the SEARCH operation is a number (meaning the text was found), or FALSE if an error occurred (meaning the text was not found). This logical result then feeds directly into the IF function.

=IF(ISNUMBER(SEARCH("Guard", A2)), "Yes", "No")

This sophisticated formula returns "Yes" if the text in cell **A2** contains the substring "Guard" anywhere within its content, regardless of its position. Conversely, if the substring is absent, the formula returns "No." This method is extremely useful for parsing descriptive fields or complex job titles.

Method 3: Checking if a Cell Contains One of Several Specific Texts

When the analysis requires checking a cell against multiple possible text values--an 'OR' condition--the complexity increases, necessitating the use of array logic combined with the SEARCH function. To handle the array of results generated by searching for multiple values, the SUMPRODUCT function is employed to aggregate the results and determine if any of the specified texts are present.

In this formula, we pass an array constant (e.g., {"Backup","Guard"}) to the SEARCH function. This generates an array of results for cell A2. The ISNUMBER function then converts these results into an array of TRUE/FALSE values. The double negative (--) coerces these logical TRUE/FALSE values into numerical 1s and 0s. Finally, SUMPRODUCT sums these 1s and 0s. If the sum is greater than 0, it means at least one text item was found, fulfilling the 'OR' condition.

=IF(SUMPRODUCT(--ISNUMBER(SEARCH({"Backup","Guard"},A2)))>0, "Yes", "No")

This powerful array formula returns "Yes" if the text in cell **A2** contains either "Backup" or "Guard" anywhere within the cell content. If neither substring is found, the result is "No." This technique is

invaluable for broad categorization based on multiple criteria.

Practical Demonstration: Applying IF Formulas to a Dataset

To illustrate the efficacy of these three methods, we will apply them to a sample dataset in Excel. Imagine we have a list of basketball players and their positions, and we need to categorize them based on specific textual requirements. The following dataset, displayed in Columns A and B, will serve as the foundation for our examples. We will use Column C to display the resulting conditional output.

	A	B	C	D	E
1	Position	Points			
2	Backup Shooting Guard	12			
3	Starting Point Guard	15			
4	Starting Center	8			
5	Backup Center	4			
6	Starting Power Forward	23			
7	Backup Point Guard	9			
8	Starting Shooting Guard	34			
9	Backup Power Forward	14			
10	Starting Small Forward	20			
11	Backup Small Forward	7			
12					
13					
14					
15					
16					
17					
18					
19					
20					

Example 1: Implementing Exact Match Comparison

In our first scenario, we want to identify only the players whose position is exactly designated as "Starting Center." This requires a strict comparison of the cell content against the target string. We initiate this process by entering the required formula into cell **C2**, targeting the position data in cell **A2**.

The objective is simple: return "Yes" if the value in cell **A2** is precisely "Starting Center," and return "No" otherwise. This ensures that variations like "Starting Center (Captain)" or "Backup Center" are correctly excluded, maintaining data integrity based on strict textual equality.

We input the following formula into cell **C2**:

=IF(A2="Starting Center", "Yes", "No")

Following the insertion of the formula, we then use the fill handle feature in Excel to click and drag this calculation down, applying it dynamically to every remaining cell throughout Column C. This automatically adjusts the cell references (A2 becomes A3, A4, and so on) for the entire range.

	A	B	C	D
1	Position	Points	Position = "Starting Center"	
2	Backup Shooting Guard	12	No	
3	Starting Point Guard	15	No	
4	Starting Center	8	Yes	
5	Backup Center	4	No	
6	Starting Power Forward	23	No	
7	Backup Point Guard	9	No	
8	Starting Shooting Guard	34	No	
9	Backup Power Forward	14	No	
10	Starting Small Forward	20	No	
11	Backup Small Forward	7	No	
12				
13				
14				
15				
16				
17				
18				
19				
20				

Example 2: Implementing Substring Containment Check

Next, we want to broaden our scope and identify any player whose position description includes the term "Guard." Since this requires checking for partial text presence, we must utilize the

combination of the ISNUMBER function nested around the SEARCH function within the IF statement, as previously outlined in Method 2. This structure allows the IF function to execute a logical test based on whether a numerical position (indicating success) is returned by the SEARCH operation.

We apply this method by typing the required formula into cell **C2**. The formula evaluates the text in cell **A2**, returning "Yes" if the substring "Guard" is found anywhere within the text, and returning "No" if it is not present. This is crucial for isolating specific role types that might be combined with other descriptors.

The specific formula entered into cell **C2** is:

=IF(ISNUMBER(SEARCH("Guard", A2)), "Yes", "No")

After entering the formula, we again use the fill handle feature to propagate this formula down Column C. This powerful technique efficiently applies the partial text check across the entire dataset, identifying all instances where "Guard" is part of the position description.

	A	B	C	D	E
1	Position	Points	Position Contains "Guard"		
2	Backup Shooting Guard	12	Yes		
3	Starting Point Guard	15	Yes		
4	Starting Center	8	No		
5	Backup Center	4	No		
6	Starting Power Forward	23	No		
7	Backup Point Guard	9	Yes		
8	Starting Shooting Guard	34	Yes		
9	Backup Power Forward	14	No		
10	Starting Small Forward	20	No		
11	Backup Small Forward	7	No		
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

Upon reviewing the results in Column C, we observe that the formula successfully returns "Yes" for every row where the position in column A contains the substring "Guard" (including "Starting Guard" and "Backup Guard"), and accurately returns "No" for all other positions, such as Center or Forward.

Example 3: Implementing Multiple Substring Containment (OR Logic)

For the final example, we seek to identify players holding positions that include either the term "Backup" or the term "Guard." This complex requirement mandates the use of array processing and aggregation, which is handled efficiently through the SUMPRODUCT function, as detailed in Method 3. This formula allows us to test a cell against an array of text strings simultaneously, implementing an implicit OR logic.

We begin by inputting the array-based formula into cell **C2**. This specific formula checks if cell **A2** contains "Backup" or "Guard" anywhere in the position description. If either (or both) of these terms are present, the SUMPRODUCT calculation results in a number greater than zero, triggering the IF

function to return "Yes."

The precise formula structure required is:

=IF(SUMPRODUCT(--ISNUMBER(SEARCH({"Backup","Guard"},A2)))>0, "Yes", "No")

After entering this advanced formula, we apply it to the entire dataset by clicking and dragging the formula down to the remaining cells in column C. This completes the conditional analysis across all records, providing a quick assessment of players matching either of the specified criteria.

	A	B	C	D	E	F
1	Position	Points	Positions Contains "Backup" OR "Guard"			
2	Backup Shooting Guard	12	Yes			
3	Starting Point Guard	15	Yes			
4	Starting Center	8	No			
5	Backup Center	4	Yes			
6	Starting Power Forward	23	No			
7	Backup Point Guard	9	Yes			
8	Starting Shooting Guard	34	Yes			
9	Backup Power Forward	14	Yes			
10	Starting Small Forward	20	No			
11	Backup Small Forward	7	Yes			
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						

The final results clearly show that the formula successfully returns "Yes" for each row that contains either "Backup" or "Guard" in column A, demonstrating robust conditional filtering. All other rows where neither term is present are correctly designated with "No."

Advanced Customization and Flexibility

It is important to understand the flexibility offered by Method 3. The use of curly brackets allows for easy expansion of the criteria. You are encouraged to include as many text values as required within the curly brackets (the array constant) in the SEARCH function. This ability to search for numerous specific text values simultaneously makes this formula an essential tool for creating comprehensive categorical filters within large datasets in Excel. Whether you need to check for three, five, or ten different keywords, the structural logic remains sound and efficient, returning "Yes" if any one of the specified criteria is met.

Summary of Key Functions Used

IF Function: Used for performing a logical test and returning one value for a TRUE result and another for a FALSE result.

SEARCH Function: Essential for finding the starting position of a substring within a text string; case-insensitive.

ISNUMBER Function: Converts the output of the SEARCH function (either a number or an error) into a Boolean TRUE/FALSE value suitable for the logical test of the IF function.

SUMPRODUCT Function: Used in advanced array formulas (Method 3) to count the number of TRUE results (converted to 1s by the double negative), ensuring that the IF function executes if at least one criterion is met.