

# How to Check if a Number is an Integer in Excel

Authored by  
**stats writer**

February 2, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Check if a Number is an Integer in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129142>

Determining whether a value within a spreadsheet is a true integer--a whole number without any fractional component--is a common requirement when working with data validation or numerical analysis in Excel. We explore several robust techniques to accurately perform this check.

One of the most straightforward and reliable methods involves comparing the original number to its truncated version using the INT function. If the original value matches the integer portion extracted by the INT function, then the number is indeed a whole number or integer. Alternatively, specialized functions like the MOD function (checking for a remainder of zero) can be employed, often in conjunction with the ISNUMBER function to ensure the cell content is numerical. These proven strategies allow users to quickly and precisely validate data types within their worksheets.

## Check if Number is an Integer in Excel

### The Primary Method: Leveraging the INT Function

The most elegant and widely accepted technique for confirming if a value is an integer in Excel relies on the mathematical principle that an integer is equal to itself after being rounded down to the nearest whole number. The INT function performs this rounding operation, returning the integer part of a number by truncating the decimal portion.

If a number, say 14.5, is processed by INT, the result is 14. Since 14 does not equal 14.5, the number is not an integer. Conversely, if the number is 15, the INT function returns 15, confirming that the value is an integer. This comparison forms the basis of the powerful Boolean formula used for data validation.

You can use the following formula structure to perform this robust check in Excel:

**=INT(A2)=A2**

This particular formula checks if the number residing in cell **A2** is equivalent to the result of applying the INT function to A2. If they match precisely, the formula returns the Boolean value **TRUE**; otherwise, it returns **FALSE**. This approach is highly efficient for data screening and type verification within large datasets.

### Step-by-Step Example: Implementing the INT Comparison Formula

To illustrate the practical application of this method, let us consider a common scenario where you have a list of numerical entries in a column and need to verify which ones are pure integers. This process involves setting up your data and then applying the formula systematically across the entire range.

Suppose we have the following list of numbers in Excel, organized in Column A:

	A	B	C	D	E
1	<b>Numbers</b>				
2	12				
3	14.2				
4	15.00001				
5	19				
6	35				
7	-3				
8	6				
9	7				
10	17.234				
11	22.9				
12					
13					
14					
15					
16					

Our objective is to check if each corresponding value in column A is an integer. We will use Column B to display the results of our validation test, starting with the first data point in row 2.

To begin, we type the following formula into cell **B2**:

**=INT(A2)=A2**

After successfully entering the formula in **B2**, we can swiftly apply this logic to the rest of the dataset. We accomplish this by dragging the fill handle (the small green square at the bottom-right corner of the selected cell) down to each remaining cell in column B. This action automatically adjusts the cell reference (A2 becomes A3, A4, and so forth) for accurate calculation across the whole list.

	A	B	C	D	E
1	<b>Numbers</b>	<b>Integer?</b>			
2	12	TRUE			
3	14.2	FALSE			
4	15.00001	FALSE			
5	19	TRUE			
6	35	TRUE			
7	-3	TRUE			
8	6	TRUE			
9	7	TRUE			
10	17.234	FALSE			
11	22.9	FALSE			
12					
13					
14					
15					

## Understanding the Output: TRUE/FALSE Boolean Logic

Upon completing the drag-and-fill operation, Column B is populated entirely with Boolean values--either **TRUE** or **FALSE**--which distinctly indicate the nature of the corresponding value in Column A. This output provides an immediate visual confirmation of data type validity for quick auditing.

When the formula returns **TRUE**, it confirms that the value in the adjacent cell is mathematically identical to its integer portion, meaning it has no decimal remainder. Conversely, a **FALSE** result signifies that the original number possesses a decimal component, no matter how small, and is therefore not an integer.

Consider the specific examples demonstrated in the output above, paying close attention to the precision of the check:

For the value **12**, applying the INT function yields 12, which matches the original number. The formula returns **TRUE**.

For the value **14.2**, the INT function returns 14. Since 14 is not equal to 14.2, the formula returns **FALSE**.

Crucially, for the number **15.00001**, even a minute decimal value prevents it from being an integer. The INT function truncates this to 15, resulting in a mismatch with the original value, and thus the formula returns **FALSE**.

## Conditional Formatting: Returning Custom Text (Yes/No) with IF

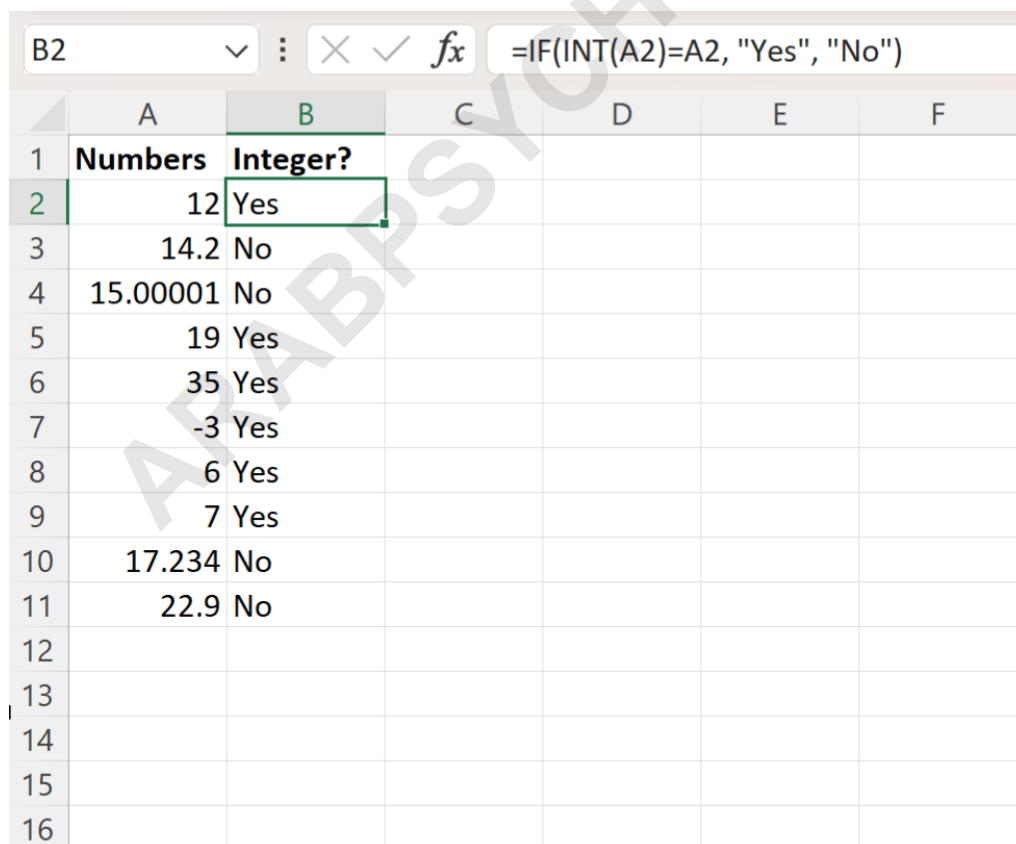
While **TRUE** and **FALSE** are mathematically precise outputs, data analysis often requires more user-friendly or customized textual responses for reporting purposes. To convert the Boolean result into specific text strings, such as "Yes" or "No," we must nest the INT comparison formula inside the powerful IF function.

The IF function tests a specified condition and returns one defined value if the condition is met (TRUE) and another defined value if the condition is not met (FALSE). By using our integer check as the logical test within the IF function, we gain complete control over the displayed output.

For example, you can use the following formula to return the confirmation "Yes" if the number is an integer and the denial "No" if it is not:

**=IF(INT(A2)=A2, "Yes", "No")**

This composite formula first evaluates whether the value in A2 is equal to its truncated integer part. If this returns **TRUE**, the IF function outputs the text "Yes"; otherwise, it outputs "No." The following screenshot demonstrates how to use this refined formula in practice:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	<b>Numbers</b>	<b>Integer?</b>				
2	12	Yes				
3	14.2	No				
4	15.00001	No				
5	19	Yes				
6	35	Yes				
7	-3	Yes				
8	6	Yes				
9	7	Yes				
10	17.234	No				
11	22.9	No				
12						
13						
14						
15						
16						

As illustrated, Column B now displays the textual confirmation of "Yes" or "No" to indicate whether or not the corresponding value in column A satisfies the strict condition of being an integer, providing a clearer, non-technical result set suitable for immediate user consumption.

## Alternative Approaches: Using the MOD and ISNUMBER Functions

While the INT comparison method is the most commonly taught technique, Excel offers alternative function combinations that achieve the same result, often preferred by analysts for specific computational needs or stylistic preferences. A powerful alternative involves the MOD function, which is specifically designed to calculate the remainder after a division operation.

The underlying logic is mathematically elegant: if you divide a number by 1, and the number is an integer, the remainder must be exactly 0. If the number has any fractional part, the remainder will be non-zero. Therefore, we can check if the remainder of the number divided by 1 is equal to zero:

**=MOD(A2, 1)=0**

This formula checks the result of the MOD function applied to the value in A2 using a divisor of 1. If the result is 0, the number is confirmed as an integer, and the formula returns **TRUE**. This method is highly reliable and is often used interchangeably with the INT method.

Furthermore, if your data set contains mixed entries (numbers, text, or blank cells) and you need to strictly confirm that a cell contains a valid number AND that it is a whole number, you must incorporate the ISNUMBER function alongside the MOD function within the AND logical test. The ISNUMBER function verifies that the input is a recognized numerical format (excluding text, errors, or logical values). A comprehensive check would look like this:

**=AND(ISNUMBER(A2), MOD(A2, 1)=0)**

## Troubleshooting and Advanced Considerations

When performing integer checks in Excel, it is vital to be aware of how floating-point arithmetic can occasionally lead to unexpected results, particularly when dealing with complex calculations. Excel uses standard floating-point precision, which sometimes creates minute rounding errors (e.g., extremely small positive or negative remainders) that can cause a number that should visually be an integer to fail the INT or MOD comparison test.

This issue typically occurs not with manually typed numbers but with numbers resulting from intermediate formulas (e.g., financial calculations or complex algebra). If a calculation results in 5.000000000000001, the INT comparison will incorrectly return **FALSE**.

To mitigate potential floating-point precision issues, especially when working with calculated values, it is often safer to use the ROUND function. By rounding the number to a sufficient number of decimal places (e.g., 10 or 12, which is beyond standard display precision) before comparing it to its integer self, you effectively eliminate irrelevant computational noise:

**=ROUND(A2, 10) = INT(ROUND(A2, 10))**

Using this rounded approach ensures that minor discrepancies caused by computer arithmetic do not corrupt the integrity of your integer check. However, for most standard datasets involving manually entered data, the simpler formula **=INT(A2)=A2** remains the most efficient and readable solution for confirming if a value is an integer.

## Summary of Integer Check Formulas

We have covered three distinct yet related methods for identifying integers in your data. Choosing the right formula depends on the specific requirements of your analysis, such as whether you need absolute mathematical precision, a simple textual output, or protection against floating-point errors.

**Primary INT Method (Standard Check):** The simplest formula, returning **TRUE** or **FALSE**.

**=INT(A2)=A2**

**Conditional IF Method (Textual Output):** Nesting the check within the IF function to return custom strings like "Yes" or "No."

**=IF(INT(A2)=A2, "Yes", "No")**

**MOD Remainder Method (Alternative Check):** Returns **TRUE** or **FALSE** based on zero remainder after division by 1.

**=MOD(A2, 1)=0**

Mastering these techniques ensures high data integrity and accuracy when processing numerical data within spreadsheets.

The following tutorials explain how to perform other common operations in Excel: