How to Determine if a Date Falls on a Weekend in Excel

Authored by

stats writer

January 1, 2026

Understanding the Goal: Identifying Non-Working Days in Excel

The ability to programmatically determine if a specific date falls on a weekend is a common requirement in data analysis and scheduling within Excel. This functionality is crucial for various operational tasks, including calculating accurate working days, adjusting project timelines that exclude non-business hours, or automating payroll processes where non-working days must be factored out. While Excel offers a rich library of date and time functions, the most reliable and straightforward approach for weekend identification involves leveraging the powerful WEEKDAY function in combination with logical operators to create a definitive classification.

Understanding how these function components interact is the key to creating robust and efficient spreadsheet solutions that scale effectively across large datasets. By converting a standard date into a corresponding numeric value representing the day of the week, we can easily isolate Saturday and Sunday. This structured approach moves beyond manual checks, providing a foundation for complex conditional calculations based on whether a date is a workday or a weekend day.

This article provides a comprehensive, step-by-step guide detailing the exact formula required and illustrating its practical application through a detailed example. We will explore the underlying logic of the WEEKDAY function and how to embed it within an IF function statement to produce clear, actionable results, whether you need to return simple text indicators or perform subsequent calculations.

The Core Tool: Introducing the WEEKDAY Function

The foundation of our weekend detection solution rests entirely upon the WEEKDAY function, which is specifically designed to convert a standard date serial number into a corresponding integer representing the day of the week. This function is essential because logical comparison operations rely on numerical inputs rather than date formatting or text names. The function returns an integer ranging from 1 to 7, where the interpretation of these numbers depends crucially on the optional return type argument supplied within the function.

For most standard uses, particularly in North America, the WEEKDAY function defaults to a system where Sunday is represented by the numeric value 1 and Saturday is represented by the numeric value 7. This default numbering system (Return Type 1) simplifies the weekend identification process significantly because the two weekend days occupy the two extreme ends of the 1-7 spectrum. Consequently, any date that yields a value of 2, 3, 4, 5, or 6 signifies a standard weekday, while the presence of 1 or 7 definitively confirms that the date falls on a weekend.

The general syntax for the WEEKDAY function is structured as: `WEEKDAY(serial_number, )`. The `serial_number` is the required date input, typically provided via a cell reference (e.g., A2), while

the optional `return_type` allows users to adjust the numbering standard based on regional preferences. By either omitting the `return_type` argument or explicitly setting it to 1, we ensure that Sunday corresponds to 1 and Saturday corresponds to 7, which are the two target values we must identify in our logical test.

Constructing the Logical Test using IF and OR

To effectively translate the WEEKDAY function's numeric output into a boolean (TRUE or FALSE) or descriptive text result, we must integrate powerful logical functions. Since a date is considered a weekend if it is *either* Sunday (1) *or* Saturday (7), we require a mechanism to check multiple conditions simultaneously. This requirement is perfectly met by the OR function, which is designed to evaluate several independent logical tests and return TRUE if at least one of those tests proves to be true. The OR function takes the structure `OR(logical1, logical2, ...)`, enabling us to efficiently bundle the two necessary weekend checks into a single, cohesive argument.

The output of the nested OR function--either TRUE (it is a weekend) or FALSE (it is a weekday)--is then passed to the outer IF function. The IF function provides the final layer of control, structuring the output into user-defined results. Structured as `IF(logical_test, value_if_true, value_if_false)`, the IF function allows us to output descriptive text strings such as "Weekend" or "Not on Weekend," or alternatively, to execute different calculations based on the nature of the day. This layered nesting ensures a clear, highly readable, and robust formula capable of handling the entire decision-making process within a single cell.

The Complete Weekend Detection Formula Breakdown

The definitive formula for checking if a date in a specified cell, such as A2, is a weekend integrates all the functions discussed above into a powerful and concise expression. Utilizing array constants (the `{1,7}` structure) within the OR function makes the formula exceptionally clean and easy to maintain, as it explicitly lists all the numerical outcomes that define a weekend.

You can use the following formula in Excel to check if a given date is on a weekend:

=IF(OR(WEEKDAY(A2)={1,7}), "Weekend", "Not on Weekend")

This formula operates through a sequential evaluation process. First, the `WEEKDAY(A2)` component evaluates the date in cell A2, returning a single numerical integer (1 through 7). Next, the result of the WEEKDAY function is compared against the array constant `{1,7}` inside the OR function. If the day number matches either 1 (Sunday) or 7 (Saturday), the OR function yields a TRUE result, indicating that the date is indeed a weekend. Finally, the outer IF function interprets this TRUE result and outputs the specified text string "Weekend." Conversely, if the WEEKDAY result falls

between 2 and 6, the OR function returns FALSE, prompting the IF function to output the value defined in the `value_if_false` argument, which is "Not on Weekend."

This particular formula checks if the date in cell **A2** is on a weekend or not and returns one of the following descriptive results:

"Weekend"
"Not on Weekend"

The following section demonstrates the practical application of this powerful formula using a typical dataset to illustrate the workflow in a real-world scenario.

Setting Up the Data for Practical Analysis

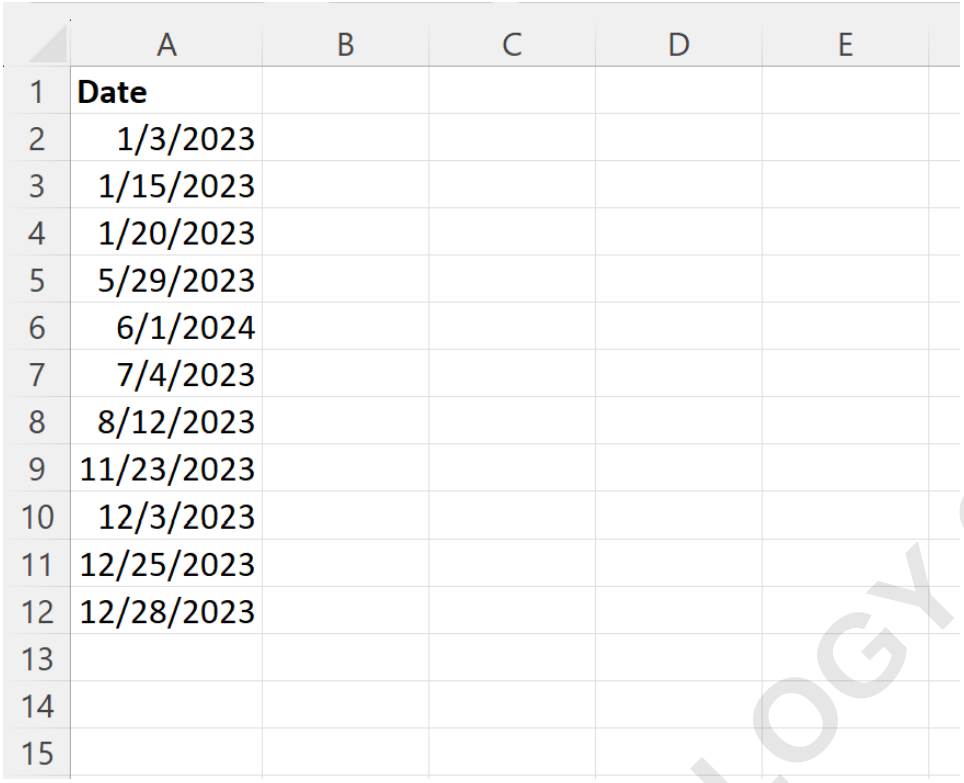
Prior to deploying the weekend detection formula, meticulous data setup is required to ensure accurate results. We initiate the process by populating a dedicated column--in our scenario, column A--with the complete list of dates that require classification. These dates must be entered in a standard, recognized date format that Excel can correctly interpret as date serial numbers, which are the underlying numeric values necessary for the date functions to calculate properly. Consistency in date formatting across the entire column is vital to prevent formula errors or unintended misinterpretations of the input data.

The objective is to analyze each date entry in column A and precisely determine whether it corresponds to a Saturday or Sunday, classifying it as a weekend, or if it falls on any of the five standard weekdays. To clearly display and manage these classifications, we allocate an adjacent column, Column B, specifically for outputting the calculated results generated by our compound logical formula. This separation ensures that the source data remains pristine while the analytical results are easily identifiable.

It is structurally important that the date in the first row of the dataset, cell **A2**, is targeted by the initial formula deployment. This cell serves as the relative reference point upon which the subsequent rows will base their calculations when the formula is efficiently copied down the column.

Example: How to Check if Date is on a Weekend in Excel

Suppose we have the following list of dates in Excel that we need to classify by day type, spanning several weeks:



	A	B	C	D	E
1	**Date**				
2	1/3/2023				
3	1/15/2023				
4	1/20/2023				
5	5/29/2023				
6	6/1/2024				
7	7/4/2023				
8	8/12/2023				
9	11/23/2023				
10	12/3/2023				
11	12/25/2023				
12	12/28/2023				
13					
14					
15					

The goal is to populate Column B with the status ("Weekend" or "Not on Weekend") corresponding to the dates listed in Column A.

Applying and Extending the Formula Across the Dataset

The practical implementation phase begins by placing the complete weekend detection formula into the first cell of the designated results column. According to our example structure, this means carefully entering the formula into cell **B2**. This initial entry tests the logical criteria against the date contained in A2, providing an immediate classification for that specific entry and confirming the formula's correct syntax and operation.

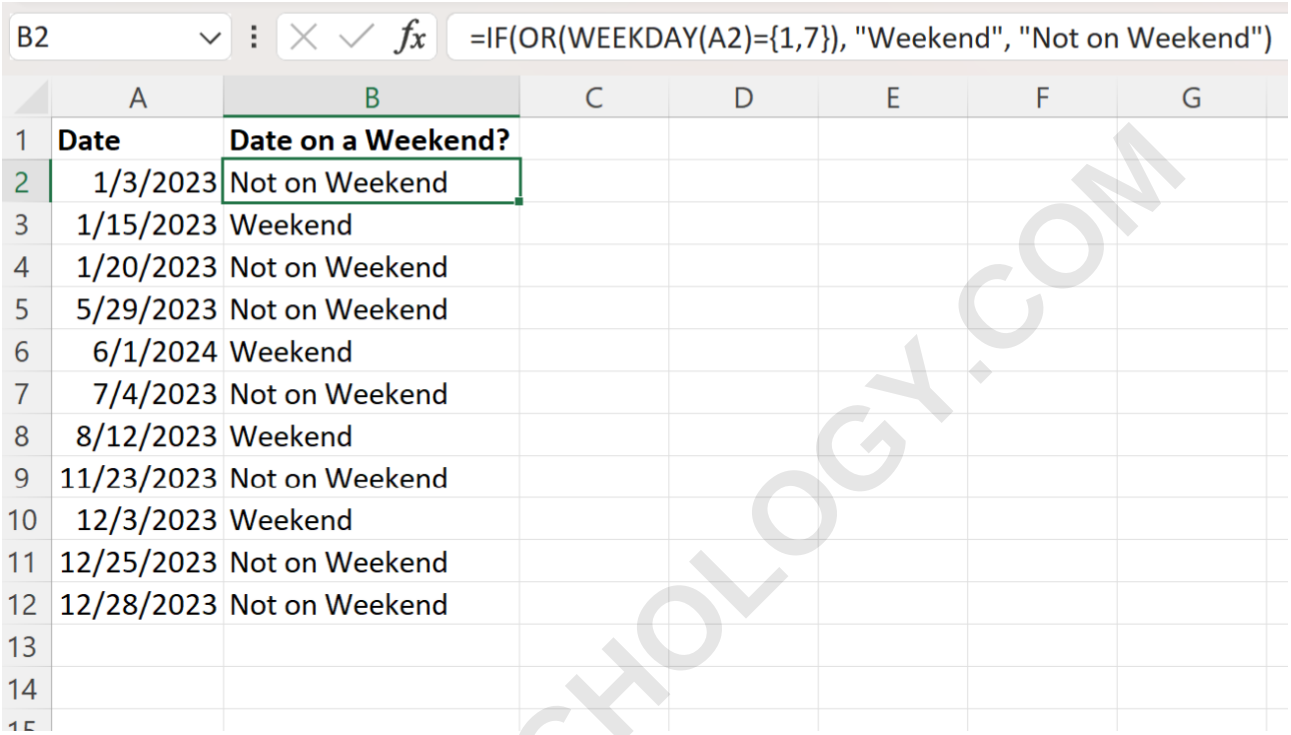
We must type the following formula precisely into cell **B2** to classify the date in cell A2:

=IF(OR(WEEKDAY(A2)={1,7}), "Weekend", "Not on Weekend")

Following the successful entry in **B2**, we leverage Excel's powerful relative referencing capability to analyze the rest of the dataset. This is achieved by selecting the fill handle--the small green square located at the bottom-right corner of cell B2--and dragging it downwards across all remaining rows that correspond to the dates in column A. As the formula is dragged, it automatically adjusts its cell reference (A2 becomes A3, A4, and so on), ensuring that the weekend detection logic is applied accurately to every date in the list without the need for repetitive manual input. This mass

application efficiently completes the classification of the entire date list.

We can then click and drag this formula down to each remaining cell in column B to complete the analysis:



	A	B	C	D	E	F	G
1	**Date**	**Date on a Weekend?**					
2	1/3/2023	Not on Weekend					
3	1/15/2023	Weekend					
4	1/20/2023	Not on Weekend					
5	5/29/2023	Not on Weekend					
6	6/1/2024	Weekend					
7	7/4/2023	Not on Weekend					
8	8/12/2023	Weekend					
9	11/23/2023	Not on Weekend					
10	12/3/2023	Weekend					
11	12/25/2023	Not on Weekend					
12	12/28/2023	Not on Weekend					
13							
14							

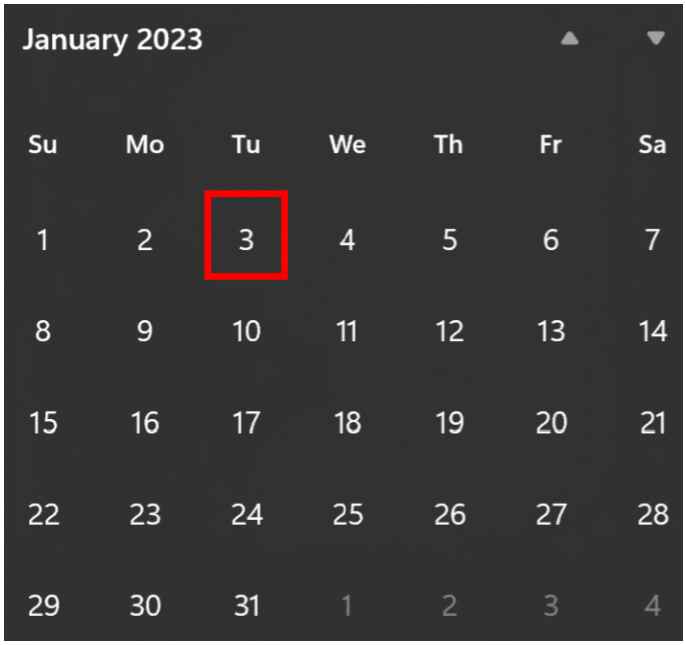
Column B is now fully populated, providing the required status and clearly indicating if each corresponding date in column A is on a weekend or not, based on the logical check performed by the nested OR function.

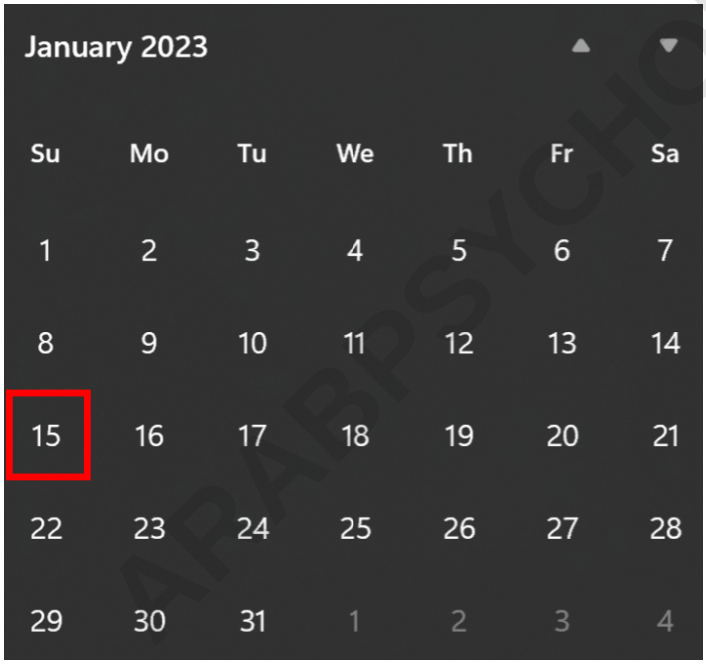
Validating Results and Customizing Output Values

After the formula has been applied to the entire dataset, a crucial step involves verifying the results to ensure they are logically accurate and align with external calendar data. A simple spot check, perhaps involving manually consulting a calendar for a few specific dates, confirms the integrity of the WEEKDAY and IF function logic. This verification confirms that the implicit default numbering system (Sunday=1, Saturday=7) was applied as intended and that the resulting output strings accurately reflect the day type.

We can verify that the results are correct by manually checking a calendar for selected entries.

For example, we can see that the date **1/3/2023** is correctly identified as "Not on Weekend," aligning with the Tuesday classification for that date:





The text strings used within the IF function are highly customizable, offering significant flexibility beyond the simple "Weekend" and "Not on Weekend" labels. Users can replace these output values with any desired text, specific numeric value, or even another nested formula. For instance, if the goal is to perform subsequent calculations, returning 0 for a weekend and 1 for a weekday might be more beneficial, allowing the column to be easily summed or used in conditional formatting rules. The adaptability of the IF function ensures this core structure can be tailored to meet a wide range of analytical and reporting requirements.

Note #1: We chose to return either "Weekend" or "Not on Weekend" as descriptive results from the **IF** function, but you have the flexibility to choose and return whatever values, calculations, or cell references you would prefer to use in place of these text strings.

Advanced Considerations: Handling Regional Return Types

While the default system (Return Type 1, where Sunday=1 and Saturday=7) is standard and forms the basis of our formula, it is important to note that the WEEKDAY function supports several return types. These alternative types accommodate diverse regional standards, especially those where the work week formally begins on a Monday, such as in ISO 8601 standards. For example, if the optional `return_type` argument is set to 2, Monday returns 1 and Sunday returns 7. In this configuration, the weekend days are Saturday (6) and Sunday (7).

If a user opts to utilize Return Type 2 or any other numbering system, the formula requires a necessary modification to maintain accuracy. Specifically, the array constant within the OR function must be adjusted to match the numerical representation of the weekend days under the new system. For Return Type 2, you would change the array from `{1,7}` to `{6,7}`. Although our primary formula implicitly relies on the default setting, awareness of these alternatives is crucial for handling global data and ensuring that the logical test remains valid irrespective of specific regional configurations or customized working week definitions.

Note #2: The **WEEKDAY** function returns a number between 1 and 7 to indicate the day of the week as a numeric value. Our formula checks if the function returns either a 1 or 7, which represent the weekend numeric values under the default (Return Type 1) configuration where Sunday is the first day of the week.

Summary and Further Analytical Applications

The strategic combination of the WEEKDAY function, the OR function, and the powerful IF function provides a robust, reliable, and highly efficient solution for accurately classifying dates in Excel. This method dramatically improves efficiency compared to manual checks, especially when dealing with large volumes of date entries, and significantly enhances the automation capabilities of your spreadsheets for scheduling and reporting purposes.

Mastering this formula is a crucial step towards implementing more advanced date-based calculations. It allows for complex tasks such as filtering transaction reports to explicitly exclude weekend sales, calculating precise net working days for project planning when standard functions like `NETWORKDAYS` might not suffice due to unique schedule requirements, or applying conditional formatting only to weekend rows for visual clarity. The reliability of this weekend detection structure serves as a fundamental building block for sophisticated time-series analysis.

By implementing and consistently applying the formula `=IF(OR(WEEKDAY(A2)={1,7}), "Weekend", "Not on Weekend")`, users gain immediate and accurate insight into the work status of any given date. This foundational technique is indispensable for professional Excel users who manage complex timelines and require automated, precise segregation of business days from non-working days to maintain data integrity and streamline operational reporting.