

Learning to Identify Weekend Dates in Excel Using the WEEKDAY Function

Authored by
Mohammed loot

November 11, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Identify Weekend Dates in Excel Using the WEEKDAY Function*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16616>

Introduction to Weekend Detection in Excel

The ability to automatically categorize dates is a fundamental requirement in data analysis, particularly when managing schedules, calculating working hours, or performing financial modeling. Determining whether a specific date falls on a weekend--Saturday or Sunday--can be efficiently accomplished in [Excel](#) using a concise combination of logical and date functions. This automation replaces tedious manual calendar checks, saving significant time and reducing the risk of human error in large datasets.

To perform this check, we integrate three powerful functions: the [WEEKDAY function](#), the [OR function](#), and the [IF function](#). By nesting these functions, we create a robust conditional statement that evaluates the date and returns a definitive text output.

You can utilize the following powerful formula in [Excel](#) to check if a given date is designated as a weekend day based on the standard calendar definition where Sunday is the first day of the week (1) and Saturday is the last (7):

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

This particular formula is designed to scrutinize the date contained in cell **A2**. It performs a logical test to determine if the day of the week corresponds to the weekend criteria. Consequently, it yields one of two clear, predefined results depending on the outcome of the conditional evaluation, which is governed by the [IF function](#).

"Weekend"

"Not on Weekend"

Deconstructing the Core Formula: IF, OR, and WEEKDAY

Understanding how the core formula operates requires a deep dive into the individual roles of the functions involved. The architecture of this solution relies on the seamless integration of date conversion and [Boolean logic](#).

The innermost function, [WEEKDAY function](#), is the critical component that translates a standard date serial number into a numeric value representing the day of the week. By default, and specifically in this formula, we are relying on the return type where Sunday equals 1 and Saturday equals 7. The argument A2 tells the function which date to evaluate. If the date in A2 were a Sunday, [WEEKDAY](#)(A2) would return 1. If it were a Saturday, it would return 7.

The next layer is the [OR function](#), which is essential for handling the dual condition of the weekend. A date is a weekend if it is EITHER Saturday OR Sunday. The array notation **={1,7}**

provides this check efficiently. The OR condition checks if the result of the [WEEKDAY function](#) is equal to 1 or equal to 7. If either condition is true, the [OR function](#) returns a value of TRUE. If the result is any number between 2 and 6 (Monday through Friday), the [OR function](#) returns FALSE.

Finally, the outermost function is the [IF function](#), which uses the TRUE/FALSE result generated by the nested [OR function](#) as its primary logical test. If the test is TRUE (meaning the date is a weekend day), the [Excel](#) formula returns the first value specified: "Weekend". Conversely, if the test is FALSE (meaning the date is a weekday), the formula returns the second value specified: "Not on Weekend". This structure ensures that every date input receives a clear, conditional classification.

Practical Application: Step-by-Step Example

To demonstrate the practical utility of this formula, consider a scenario where we have a column containing numerous dates, and we require a corresponding column indicating whether each date necessitates special scheduling or processing, such as weekend overtime calculation. The following example illustrates how to integrate this formula into a standard [Excel](#) sheet.

Suppose we begin with the following list of dates in Column A of our [Excel](#) spreadsheet, representing various points throughout the calendar year:

	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					

Our objective is to systematically check if each date listed in column A falls on a weekend or not.

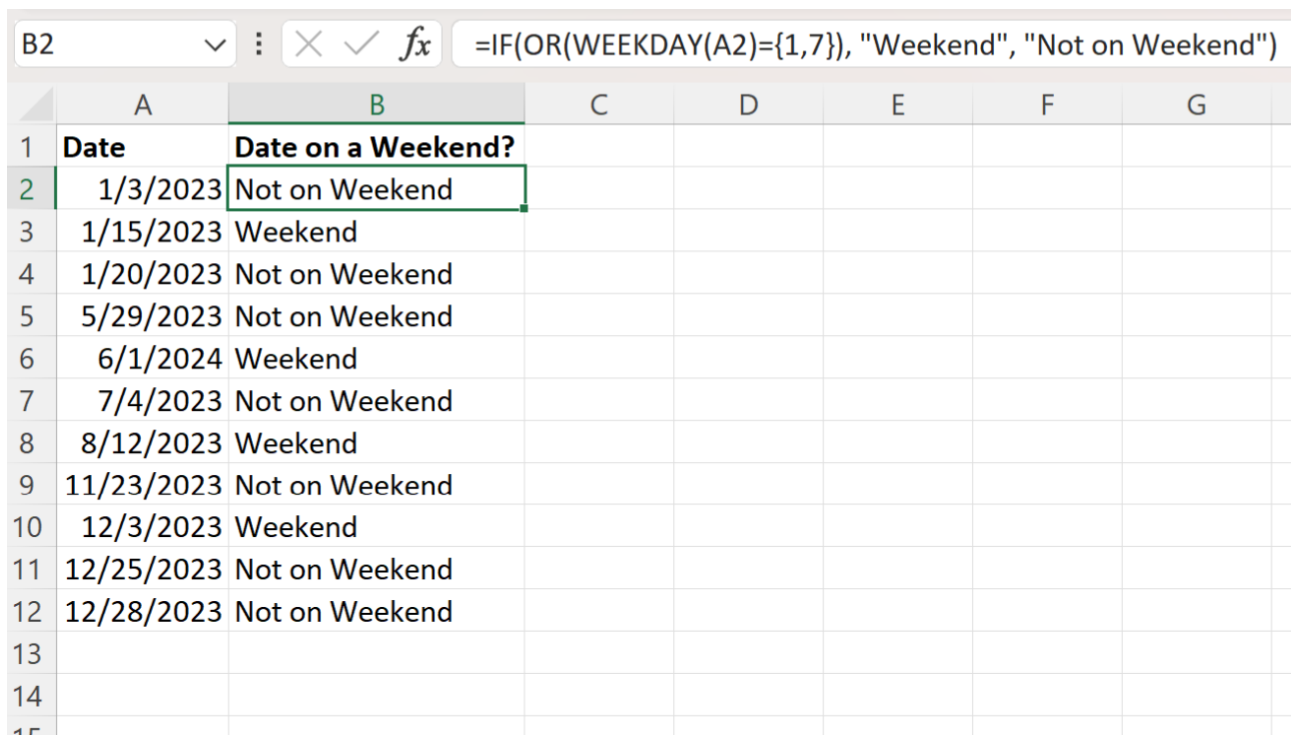
This process will involve inserting the appropriate formula once and then efficiently applying it to the entire dataset.

We initiate the process by typing the critical formula into cell **B2**, which corresponds directly to the first date entry in cell A2. The formula references A2, ensuring that the calculation is specific to that row's date value:

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

Once the formula is entered into **B2**, we can leverage the powerful fill handle feature in [Excel](#). By clicking and dragging the corner of cell B2 down to each remaining cell in Column B, we automatically apply the formula to the corresponding dates in Column A, adjusting the cell reference (A2 changes to A3, A4, and so on) thanks to relative referencing.

The result is a fully populated Column B, which provides immediate and accurate classification for every date in the list:



	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							
15							

As clearly illustrated, Column B now serves as a reliable indicator, telling us definitively whether each corresponding date in Column A is designated as a weekend or a standard weekday. This output is ready for use in further analysis, sorting, or filtering operations.

Interpreting the WEEKDAY Function Logic

A thorough understanding of the [WEEKDAY function](#)'s return types is paramount to ensuring the accuracy of the weekend check. The function's utility lies in its ability to convert complex date serial data into a simple integer, but the meaning of that integer depends on the optional second argument, **return_type**.

In the formula presented, we implicitly use the default return type (Type 1) because no second argument is specified. Under this standard setting, the numbering system is based on the North American convention where the week begins on Sunday. The numerical representations are as follows:

Sunday

Monday

Tuesday

Wednesday

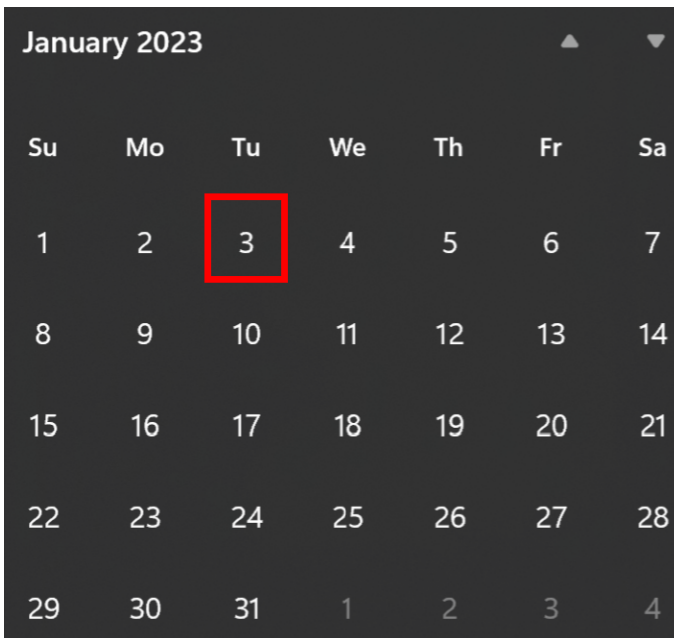
Thursday

Friday

Saturday

Therefore, our formula explicitly checks for the values 1 (Sunday) and 7 (Saturday) using the **={1,7}** array. If the date in question, such as **1/3/2023** in our example, returns a value of 3 (Tuesday), the [OR function](#) evaluates to FALSE, and the [IF function](#) returns "Not on Weekend."

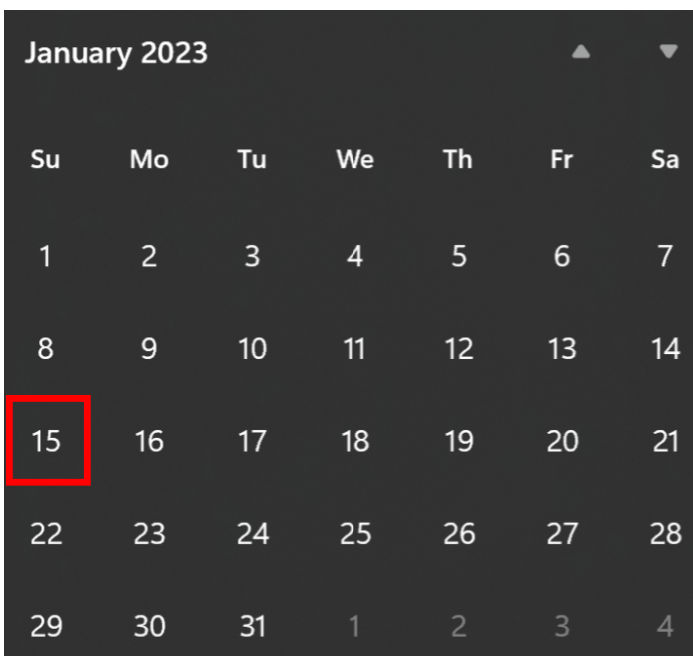
We can confidently verify that the results generated by the formula are accurate by manually cross-referencing a calendar. For instance, based on the calculation, we can confirm that the date **1/3/2023** is indeed a weekday:



A calendar for January 2023 with a dark background. The days of the week are abbreviated as Su, Mo, Tu, We, Th, Fr, Sa. The date 3 is highlighted with a red square.

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Similarly, if the date were 1/1/2023, the [WEEKDAY function](#) would return 1, satisfying the OR condition and yielding the result "Weekend." This reliance on consistent numeric mapping is what makes the formula reliable across different versions of [Excel](#).



A calendar for January 2023 with a dark background. The days of the week are abbreviated as Su, Mo, Tu, We, Th, Fr, Sa. The date 15 is highlighted with a red square.

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Customizing Outputs and Alternative Methods

The true power of the [IF function](#) lies in its flexibility regarding output. While we chose to return

either "Weekend" or "Not on Weekend" as the resulting values, these text strings can be easily customized to suit specific operational needs.

Note #1: Users often need to return numerical values, such as a monetary bonus for weekend work (e.g., \$100) or a simple Boolean numerical indicator (1 for weekend, 0 for weekday). To achieve this, you simply replace the text strings in the formula's `value_if_true` and `value_if_false` arguments with the desired numeric values or cell references. For example, to return a 1 for a weekend and 0 otherwise, the formula would be written as: `=IF(OR(WEEKDAY(A2)={1,7}), 1, 0)`.

This classification technique is highly valuable for integrating with other calculation structures. For instance, the resulting column can be used as a conditional flag for complex payroll formulas or as the basis for powerful data filtering. Furthermore, this classification can be achieved without generating a separate column by applying the same logical test directly within [Excel's](#) Conditional Formatting rules. This allows for immediate visual identification (e.g., highlighting weekend dates in red) based on the same underlying [WEEKDAY](#) logic.

Troubleshooting and Key Considerations

When implementing date-based formulas, attention to detail regarding data format and function parameters is crucial. While the formula provided is highly effective, users must be aware of potential issues that could lead to incorrect results.

The most common error encountered relates to the input date itself. [Excel](#) must recognize the value in cell A2 as a valid date serial number. If the cell is formatted as text, or if the date format (e.g., MM/DD/YYYY vs. DD/MM/YYYY) conflicts with the regional settings of the spreadsheet, the [WEEKDAY function](#) may return a **#VALUE!** error or an incorrect numeric code, thus skewing the final result. Always ensure the input column is correctly formatted as a Date.

Note #2: As discussed, the [WEEKDAY function](#) returns a number between 1 and 7, which indicates the day of the week as a numeric value. Crucially, our formula relies on the default configuration where 1 represents Sunday and 7 represents Saturday. If your region defines the start of the week differently (e.g., Monday as the start), you might need to manually specify a different **return_type** argument for the [WEEKDAY function](#) (such as type 2, where Monday = 1 and Sunday = 7). If using Type 2, the weekend check would need to be `= {6,7}` (Friday and Saturday) or `= {7,1}` (Sunday and Monday) depending on the definition used.

Additional Resources

Mastering date and time calculations is fundamental for advanced [Excel](#) users. The following tutorials explain how to perform other common operations in [Excel](#) that complement date checking:

Calculating the number of working days between two dates.

Extracting the month or year from a date value.

Using conditional formatting to highlight dates based on specific criteria.