

# Learn How to Calculate the First Sunday of Each Month Using Excel Formulas

Authored by  
**Mohammed loot**

November 11, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learn How to Calculate the First Sunday of Each Month Using Excel Formulas*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16634>

## Understanding the Need for Dynamic Date Calculations in Excel

[Excel](#) is an indispensable tool for managing large datasets, particularly those involving time-sensitive scheduling, financial reporting, and project management. A common and often challenging requirement is performing complex [date calculation](#), such as determining the exact date of a recurring event like the first Sunday of any given month. While calendars make this task simple visually, translating that logic into a reliable, scalable spreadsheet [formula](#) requires a deep understanding of how Excel handles dates internally.

In Excel, dates are stored as serial numbers, where January 1, 1900, is serial number 1. This system allows for mathematical operations on dates, but extracting specific day-of-week information necessitates the use of specialized functions. The method presented here provides a robust and dynamic solution that automatically adjusts regardless of the month or year provided, making it highly valuable for creating automated schedules or dynamic monthly reports.

To successfully identify the first Sunday, we must first establish the beginning of the relevant month and then calculate the offset needed to reach the first occurrence of the desired weekday (Sunday, in this case). This process involves combining several core date functions to manipulate the serial number effectively.

### The Core Formula Explained

To efficiently locate the first Sunday of a month corresponding to a specific input date, you can employ the following powerful combination of native [Excel](#) functions. This approach leverages basic arithmetic operations on Excel's date serial numbers to pinpoint the precise date.

The generalized [formula](#) used to find the first Sunday of the month based on a date in cell **A2** is presented below. Understanding the components of this formula is crucial for effective implementation and troubleshooting.

**=A2-DAY(A2)+8-WEEKDAY(A2-DAY(A2))**

This particular formula structure is designed to first determine the starting point of the month and then calculate the necessary offset. The segment [DAY\(A2\)](#) returns the day number of the month (e.g., 15 if the date is January 15th). Subtracting this value from the original date in A2 effectively yields the last day of the preceding month. For instance, if A2 is January 15th, subtracting 15 results in December 31st of the previous year (the serial number equivalent).

The subsequent steps manipulate this result to find the starting point of the week within the target month. The expression [A2-DAY\(A2\)](#) is equivalent to the day before the first day of the month. By applying the [WEEKDAY](#) function to this value, we determine which day of the week that precursor

date falls on (where 1 is Sunday and 7 is Saturday, by default). We then add 8 and subtract the determined weekday offset. The addition of 8 ensures we move past the first day of the month and account for the necessary shift to land precisely on the first Sunday, irrespective of whether the first day of the month is a Sunday or a Monday. This elegant mathematical [formula](#) reliably returns the date corresponding to the first Sunday for the month specified in cell **A2**.

## Step-by-Step Implementation: Finding the First Sunday

To demonstrate the practical application of this dynamic date extraction technique, let us consider a scenario where we have a list of various dates spanning different months and years. Our objective is to generate a parallel column displaying the date of the first Sunday for the corresponding month of each date listed.

Suppose we have the following list of dates entered into Column A of our [Excel](#) worksheet, starting in cell **A2**:

	A	B	C	D	E	F
1	<b>Date</b>					
2	1/1/2023					
3	2/1/2023					
4	3/1/2023					
5	4/1/2023					
6	5/1/2023					
7	6/1/2023					
8	7/1/2023					
9	8/1/2023					
10	9/1/2023					
11	10/1/2023					
12	11/1/2023					
13	12/1/2023					
14						
15						
16						
17						
18						

We initiate the process by selecting cell **B2**, which will serve as the output location for the first Sunday corresponding to the date in **A2**. The formula presented earlier is then input directly into this cell. This formula is carefully constructed to handle the inherent complexity of identifying a specific weekday occurrence within a monthly cycle, relying on Excel's internal serial number

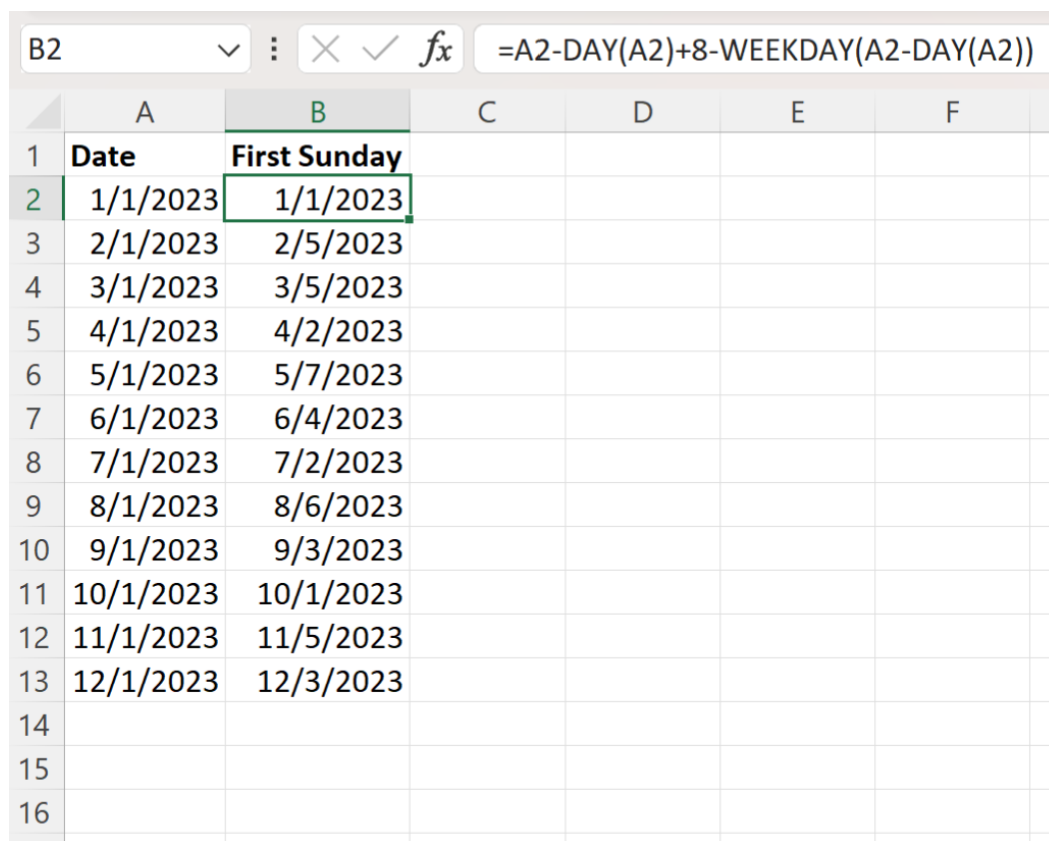
system for accurate [date calculation](#).

Type the following exact [formula](#) into cell **B2**:

**=A2-DAY(A2)+8-WEEKDAY(A2-DAY(A2))**

After entering the formula into **B2**, the result will display the first Sunday of the month represented by the date in **A2**. The true efficiency of this method is realized when we apply it to the entire dataset. Utilizing the fill handle--the small square at the bottom-right corner of cell **B2**--we can click and drag this formula down to populate the remaining cells in Column B. This action automatically adjusts the cell references (e.g., changing **A2** to **A3**, **A4**, and so on) for each subsequent row, ensuring that the first Sunday is calculated correctly for every date in Column A.

The resulting table demonstrates the successful application of the formula across all input dates:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	<b>Date</b>	<b>First Sunday</b>				
2	1/1/2023	1/1/2023				
3	2/1/2023	2/5/2023				
4	3/1/2023	3/5/2023				
5	4/1/2023	4/2/2023				
6	5/1/2023	5/7/2023				
7	6/1/2023	6/4/2023				
8	7/1/2023	7/2/2023				
9	8/1/2023	8/6/2023				
10	9/1/2023	9/3/2023				
11	10/1/2023	10/1/2023				
12	11/1/2023	11/5/2023				
13	12/1/2023	12/3/2023				
14						
15						
16						

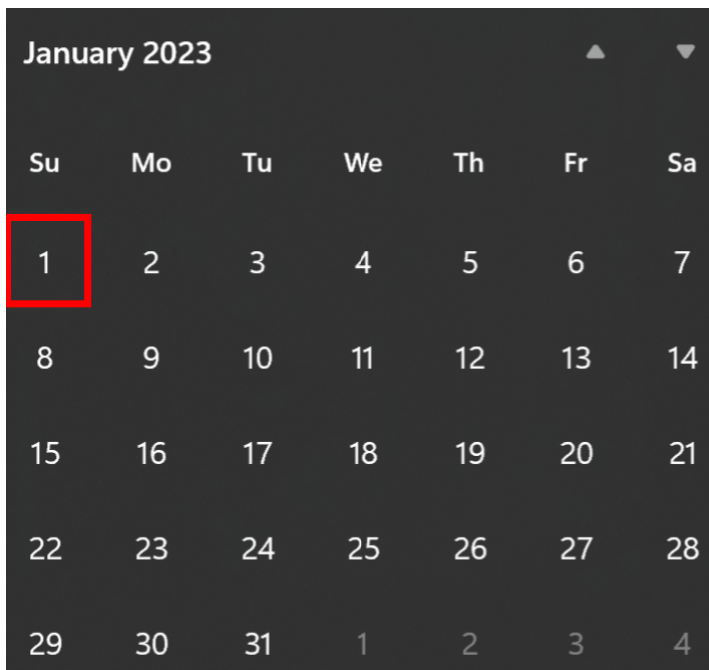
## Analyzing the Results and Verification

Upon completing the drag-and-fill operation, Column B now contains the date of the first Sunday for the month associated with each corresponding date in Column A. It is always good practice to verify the results of complex date calculations using an external reference, such as a traditional

calendar, to confirm the accuracy of the spreadsheet [formula](#).

Consider the first input date, **1/1/2023**, which corresponds to the month of January 2023. The formula returns the date **1/1/2023** as the first Sunday. We can verify this result by consulting a calendar for January 2023. This visual confirmation ensures that our calculation method is sound and that the combination of the [WEEKDAY](#) and [DAY](#) functions correctly identified the required date offset.

The calendar view below visually confirms that January 1, 2023, was indeed the first Sunday of that month:

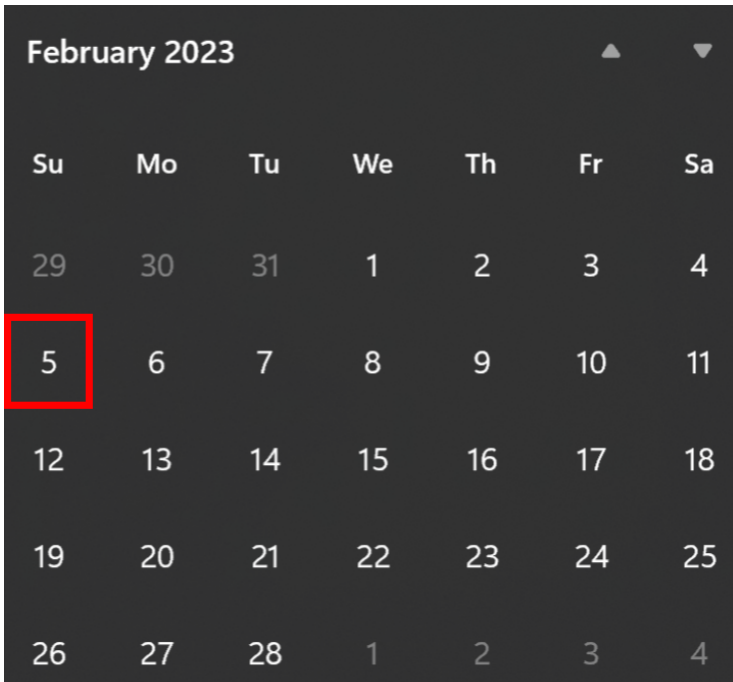


A calendar for January 2023. The days of the week are listed at the top: Su, Mo, Tu, We, Th, Fr, Sa. The dates are arranged in a grid. The date 1 is highlighted with a red square, indicating it is a Sunday.

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

Let us examine the second input date, **2/1/2023**, which represents the month of February 2023. The calculated result in Column B is **2/5/2023**. Since February 1st was a Wednesday, the first Sunday naturally falls several days into the month. Consulting the calendar for February 2023 allows us to confirm that February 5th is the correct date for the first Sunday.

This step-by-step verification process validates the reliability of the complex date [calculation](#) implemented in the [Excel](#) sheet:



February 2023

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	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	1	2	3	4

By verifying these representative examples, we can be confident that every date displayed in Column B accurately represents the first Sunday of the month corresponding to its input date in Column A. This successful application demonstrates the power of combining simple arithmetic with sophisticated date functions for precise logistical scheduling.

## Advanced Considerations and Date Function Components

While the formula provided is highly effective, it is beneficial to understand the specific role of the two primary functions utilized: [DAY](#) and [WEEKDAY](#). The efficiency of the overall date calculation hinges on the proper interaction between these two components and the underlying math that manages the serial dates.

The [DAY](#) function is straightforward; it extracts the day number (1 through 31) from a provided date serial number. In our formula, `A2-DAY(A2)` is the cornerstone for resetting the date to the last day of the preceding month, essentially giving us a known starting point from which to calculate the offset into the new month.

The [WEEKDAY](#) function is more complex as it returns a number (1 to 7) representing the day of the week. Crucially, the function allows for different return types (which day is considered 1). By default, and in the context of this formula, Sunday is 1 and Saturday is 7. The application of `WEEKDAY(A2-DAY(A2))` tells us what day of the week the last day of the previous month was. This result is then used as the necessary subtraction offset to shift the date forward precisely to the next Sunday. The final addition of 8 ensures that the calculation lands on the first day of the new

month that is a Sunday.

## **Additional Resources for Date Manipulation**

Mastering dynamic date manipulation in [Excel](#) opens up numerous possibilities for advanced reporting and scheduling tasks. The calculation of the first Sunday is just one specialized application. Below is a list of related tutorials and functions that explain how to perform other common and complex operations using date and time serial numbers:

Calculating the last day of a month using the **EOMONTH** function.

Determining the Nth occurrence of a specific weekday (e.g., the third Tuesday) using array formulas or combinations of **DATE** and **WEEKDAY**.

Working with financial periods and calculating workdays using the **NETWORKDAYS** function.

Extracting components of a date using **YEAR**, **MONTH**, and [DAY](#) functions for detailed reporting.

Using conditional formatting to highlight specific weekdays or dates within a schedule.

These resources provide further insight into creating robust and automated date systems within your spreadsheets, moving beyond basic data entry to sophisticated [date calculation](#) methodology.