

Learn How to Calculate the First Day of the Month in Excel

Authored by
Mohammed looti

March 11, 2026

RECOMMENDED CITATION

Mohammed looti (2026). *Learn How to Calculate the First Day of the Month in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=3226>

Mastering date calculations in [Excel](#) is an essential skill, critical for professionals engaged in [data analysis](#), [financial reporting](#), and [project management](#) across various industries. A very frequent requirement in these fields is the ability to accurately determine the **first day of a given month** based on any specific date within that period. This functionality is vital for tasks such as standardizing reporting periods, generating precise monthly summaries, and ensuring efficient organization of time-sensitive data within complex spreadsheets. Fortunately, Excel provides users with robust and powerful functions designed specifically to achieve this date manipulation with impressive simplicity and accuracy.

This expert guide is dedicated to exploring two distinct yet equally effective formula methodologies for extracting the first day of the month from any date input in Excel. We will systematically examine the underlying mechanics of each technique, providing crystal-clear explanations and step-by-step practical examples. Our goal is to ensure you can confidently implement these solutions in your professional workflow. Understanding both approaches--one based on fundamental date arithmetic and the other utilizing a specialized date function--will equip you with versatile tools for tackling a wide array of date manipulation challenges.

We will begin by contrasting the two primary formulas available for retrieving the monthly starting date in Excel:

Formula 1: Utilizing Fundamental Date Arithmetic with DAY()

=A2-DAY(A2)+1

Formula 2: Employing the Specialized EOMONTH() Function

=EOMONTH(A2,-1)+1

Both of these formulas are meticulously designed to return the **first day of the month** for the arbitrary date housed in [cell A2](#). The following detailed sections will guide you through the application of each formula using a consistent sample dataset.

For demonstration purposes, please refer to the sample dataset below, which will serve as the input for both calculation methods:

	A	B	C	D	E	F
1	Date	Sales				
2	1/4/2022	23				
3	2/15/2022	29				
4	3/23/2022	40				
5	4/15/2022	45				
6	4/18/2022	34				
7	5/1/2022	35				
8	8/7/2022	20				
9	10/31/2022	20				
10	11/4/2022	23				
11	12/19/2022	29				
12						
13						
14						
15						
16						
17						
18						
19						
20						

Understanding the DAY() Function Approach

The [DAY\(\) function](#) in Excel is a fundamental component of the date and time function library. Its purpose is singular and straightforward: to extract the numerical day component (ranging from 1 to 31) from a provided date value. For example, if a cell contains the date "March 25, 2024," applying the **DAY() function** to that cell will yield the integer 25. This basic extraction capability is absolutely crucial to the success of our first method for programmatically identifying the beginning of the month.

The underlying logic of the formula $=A2-DAY(A2)+1$ is rooted in how [Excel](#) handles date storage: as sequential [date serial numbers](#). In this system, every date is represented by an integer count of days passed since January 1, 1900. When you perform basic [date arithmetic](#) (subtraction or addition) on a date cell, you are effectively shifting the serial number backward or forward by that many days. This mathematical foundation allows us to precisely manipulate the date to land on the first day of the month.

To fully grasp this technique, let's dissect the formula into its three operational steps, using an example date of "July 18, 2023" stored in **cell A2**:

DAY(A2): This initial calculation isolates the day component of the date. For our example, **DAY(A2)** returns the number 18.

A2 - DAY(A2): By subtracting the extracted day number (18) from the full date ("July 18, 2023"), we effectively move backward in the serial number sequence. This maneuver always results in the date corresponding to the day *before* the current month started--that is, the **last day of the previous month**. In our case, "July 18, 2023" minus 18 days yields "June 30, 2023."

+1: The final step involves adding the number 1 to the result obtained in the previous step. Adding one day to the last day of the preceding month ("June 30, 2023") advances the date to "July 1, 2023," which is the sought-after **first day of the month**. This method is highly favored by those who appreciate a direct, mathematically transparent solution utilizing Excel's core date structure.

Practical Application: Implementing the DAY() Function

To demonstrate the practical efficiency of the **DAY() function** method, we will now apply the formula to our defined example dataset. Our objective is to generate a new column, **Column C**, which will be populated with the **first day of the month** corresponding to every date entry found in **Column A**.

Follow the steps below precisely to implement and deploy this formula across your data range:

Click and select **cell C2**, which is designated as the location for our initial calculated result.

Carefully type or paste the following formula into **cell C2**:

```
=A2-DAY(A2)+1
```

Once the formula is entered, press the **Enter** key. Excel will execute the calculation and display the **first day of the month** derived from the date in **cell A2**. For instance, if the input date in A2 is "7/18/2023", the output in C2 will correctly show "7/1/2023".

To quickly and efficiently apply this calculation to the entire dataset, utilize Excel's powerful auto-fill feature. Locate the fill handle--the small, dark square positioned at the bottom-right corner of **cell C2**. Click on this handle and drag it downward until you reach the last row containing data in **Column A**. This action systematically copies the formula down the column, automatically adjusting the [cell references](#) (A2 becomes A3, A4, and so forth) for each subsequent calculation. The result is a fully populated column where every cell accurately displays the monthly start date corresponding to its adjacent entry in **Column A**.

C2						
=A2-DAY(A2)+1						
	A	B	C	D	E	F
1	Date	Sales	First Day of Month			
2	1/4/2022	23	1/1/2022			
3	2/15/2022	29	2/1/2022			
4	3/23/2022	40	3/1/2022			
5	4/15/2022	45	4/1/2022			
6	4/18/2022	34	4/1/2022			
7	5/1/2022	35	5/1/2022			
8	8/7/2022	20	8/1/2022			
9	10/31/2022	20	10/1/2022			
10	11/4/2022	23	11/1/2022			
11	12/19/2022	29	12/1/2022			
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

As clearly visualized in the image above, **Column C** now accurately reflects the **first day of the month** for every date listed in **Column A**. This methodology proves to be highly effective and scalable for rapidly calculating monthly start dates across extensive datasets.

Understanding the EOMONTH() Function Approach

The [EOMONTH\(\) function](#), which stands for "End Of MONTH," provides an elegant and specialized alternative for date calculations, particularly useful when requirements center around month-end or month-start periods. Unlike the **DAY() function**, which only extracts a numerical part of a date, **EOMONTH()** calculates and returns a complete date value--specifically, the [date serial number](#) representing the last day of the month, shifted by a user-specified number of months before or after the initial date.

The standard syntax for the **EOMONTH() function** is straightforward: `EOMONTH(start_date, months)`. Understanding its arguments is key to successful implementation:

start_date: This is the mandatory initial date used as the reference point for the calculation. It can be a hardcoded date value, the result returned by another date function, or, most commonly, a [cell](#)

[reference](#) containing the date you wish to analyze.

months: This is an integer value that determines how many months to shift forward or backward from the **start_date**.

A **positive value** (e.g., 1) moves the date forward, returning the end of a future month.

A **negative value** (e.g., -1) moves the date backward, returning the end of a preceding month.

A **zero value** (0) returns the end of the current month.

To efficiently find the **first day of the current month** using this dedicated function, we construct the formula as `=EOMONTH(A2, -1)+1`. The formula works by strategically calculating the last day of the previous month, then simply adding one day. If **A2** contains "September 15, 2024," the calculation proceeds as follows: **EOMONTH(A2, -1)** immediately yields "August 31, 2024" (the last day of the previous month). By then adding **+1** to this result, we advance the date to "September 1, 2024," successfully isolating the **first day of the month**. This high-level, semantic approach is often preferred for its clean readability and directness in complex month-shifting scenarios.

Practical Application: Implementing the EOMONTH() Function

We will now apply the **EOMONTH() function** to the exact same dataset utilized previously. This exercise clearly demonstrates that both primary methods reliably yield identical results when calculating the **first day of the month** for each corresponding entry.

Follow these implementation steps to populate **Column C** using the **EOMONTH()** method:

Select the target starting cell, **cell C2**, where the first calculated date will be displayed.

Enter the specialized formula into **cell C2**:

=EOMONTH(A2,-1)+1

After pressing **Enter**, the **first day of the month** derived from the date in **cell A2** will populate **C2**. Consistent with our previous examples, an input of "7/18/2023" in A2 will correctly result in "7/1/2023" in C2.

To finish the task, leverage the fill handle functionality. Click on **cell C2**, then click and drag the small square at the bottom-right corner of the selection down to the end of your data range in **Column A**. This action automatically copies the **EOMONTH() function** formula to all subsequent rows, ensuring dynamic adjustment of the [cell reference](#) for accurate calculations across the entire dataset. Upon completion of the auto-fill, **Column C** will be fully populated with the required start dates, matching the output achieved using the **DAY() function** method.

	A	B	C	D	E	F
1	Date	Sales	First Day of Month			
2	1/4/2022	23	1/1/2022			
3	2/15/2022	29	2/1/2022			
4	3/23/2022	40	3/1/2022			
5	4/15/2022	45	4/1/2022			
6	4/18/2022	34	4/1/2022			
7	5/1/2022	35	5/1/2022			
8	8/7/2022	20	8/1/2022			
9	10/31/2022	20	10/1/2022			
10	11/4/2022	23	11/1/2022			
11	12/19/2022	29	12/1/2022			
12						
13						
14						
15						
16						
17						
18						
19						
20						

The resulting visual confirms that **Column C** displays the correct **first day of the month** for every entry in **Column A**. The **EOMONTH()** function provides a robust, standardized, and highly direct approach to date manipulation, especially beneficial when calculations involve shifting across complex month or year boundaries.

Comparing the Methods and Best Practices

While both the **DAY()** function approach ($=A2-DAY(A2)+1$) and the **EOMONTH()** function approach ($=EOMONTH(A2,-1)+1$) successfully calculate the **first day of the month** in Excel, the decision of which to use often hinges on context, personal familiarity, and the demands of the overall spreadsheet structure. Both are highly optimized, but they operate using different logical frameworks that appeal to different user preferences.

The **DAY()** function methodology is generally perceived as more foundational and transparent. It explicitly leverages Excel's native **date serial number** system and relies purely on basic **date arithmetic**. Users who prefer to maintain a granular, step-by-step understanding of how the calculation processes--subtracting the current day count to reach the previous month's end, then adding one--often find this method more satisfying and intuitive. It stands as an excellent

introductory example of how dates are mathematically managed within the spreadsheet environment.

In contrast, the **EOMONTH() function** method is highly specialized and generally more concise for complex date shifting. Its primary advantage is its dedicated purpose: calculating month boundaries. If your future requirements frequently involve finding the last day of the current month, the last day of the previous quarter, or the start date of a period several months away, **EOMONTH()** offers a superior, more semantic solution. It successfully abstracts the complexities of date serial numbers, focusing instead on logical month shifts, making the formula easier to read and maintain for complex financial or reporting models.

To ensure maximum reliability and professionalism in your spreadsheets, regardless of the formula chosen, adhere to these best practices:

Source Data Validation: Always confirm that your input cell (e.g., **cell A2**) contains a valid date format. If **A2** holds non-date text or an unrecognized value, both formulas will typically fail, resulting in a #VALUE! error.

Output Formatting: It is critical to format the resulting cells (e.g., **Column C**) as a **Date**. If the default General format is retained, Excel will display the underlying [date serial number](#) (a large integer) instead of a readable date format like "7/1/2023."

Readability: While the performance difference is negligible for standard data sizes, many advanced Excel users favor **EOMONTH()** for its clear semantic meaning (End Of MONTH), which enhances formula auditability and maintainability over time.

Conclusion and Next Steps for Date Mastery

The effective management and manipulation of dates within [Excel](#) remains a cornerstone skill for anyone dealing with time-sensitive data, particularly in fields requiring rigorous monthly or quarterly reporting. By mastering the two powerful methodologies presented here--the fundamental approach using the [DAY\(\) function](#) and the specialized technique employing the [EOMONTH\(\) function](#)--you are equipped to accurately and efficiently determine the **first day of any given month** from a specified reference date. These skills are invaluable for streamlining report generation and ensuring precise timeline tracking.

Although both formulas achieve the identical outcome, a professional Excel user gains significant advantage by understanding the distinct underlying logic and appropriate use cases for each. The **DAY() function** provides a transparent, arithmetic-based solution rooted in Excel's serial number system, while the **EOMONTH() function** offers a specialized, concise, and highly readable alternative for calculations focused on month boundaries. We highly recommend practicing both methods with varied datasets to determine which approach integrates best into your personal workflow and meets the complexity of your data models.

To further enhance your data [manipulation](#) capabilities, we encourage deep exploration of Excel's extensive suite of date and time functions. Functions such as **YEAR()**, **MONTH()**, **WEEKDAY()**, **TODAY()**, **NOW()**, and **DATE()** can be combined creatively to solve even the most intricate date-related challenges in your spreadsheets, moving you toward true mastery of time-based data management.

For additional learning and to further refine your Excel skills, the following tutorials explain how to perform other common operations in Excel: