

Learn to Calculate the First Friday of Any Month in Excel

Authored by
Mohammed looti

November 11, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn to Calculate the First Friday of Any Month in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16629>

The Challenge of Temporal Calculation in [Excel](#)

The ability to manipulate dates effectively is a cornerstone of advanced spreadsheet analysis, yet identifying specific temporal markers, such as the first occurrence of a particular weekday within a given month, often requires complex nested formulas. Unlike simple arithmetic, date calculations in [Excel](#) rely on the concept of a **Date Serial Number**, where each day is represented sequentially from a specific starting point (typically January 1, 1900). Understanding how this numerical representation interacts with functions designed to extract day-of-week information is essential for constructing robust and reliable date formulas. We aim to provide a detailed, single-cell solution that efficiently determines the first Friday of any month referenced by a source date. This technique moves beyond basic date subtraction, leveraging several core functions to normalize the calendar calculation and pinpoint the desired day accurately, regardless of the year or month provided.

This sophisticated formula streamlines what would otherwise require multiple helper columns or complex conditional logic. When dealing with large datasets or recurring scheduling tasks, efficiency is paramount. By encapsulating the entire logic into one concise [Excel](#) formula, we minimize potential errors and maximize calculation speed. The formula presented here is universally applicable for finding the first Friday, assuming the source date is located in cell **A2**. Mastering this approach provides a valuable tool for anyone managing project schedules, payroll, or recurring financial deadlines within a spreadsheet environment, ensuring precision in identifying those critical first-Friday dates.

The specific [Excel](#) formula used to accomplish this task is presented below. This formula is designed to perform a series of numerical manipulations on the initial date, effectively rewinding the date to the first day of the month and then calculating the offset required to jump forward to the first Friday. Pay close attention to the structure, as the constants within the formula are critical for ensuring that the calculation aligns precisely with the **WEEKDAY** system used by Excel.

=A2-DAY(A2)+8-WEEKDAY(A2-DAY(A2)+2)

This particular construction returns the first Friday of the month corresponding to the date found in cell **A2**. The remainder of this guide will meticulously dissect the components of this powerful formula, illustrating how each element contributes to the final, accurate result.

Deconstructing the Formula: Understanding Date Offsets

To truly understand the elegance of this solution, we must break down the formula into its constituent parts, focusing on how it leverages date arithmetic to isolate the target day. The calculation begins with the core objective of finding the date of the first day of the month, regardless of the day referenced in **A2**. This is achieved by the expression: $A2 - \text{DAY}(A2)$. The [DAY](#)

[function](#) extracts the day number (1 to 31) from the date in **A2**. Subtracting this number from the full serial date in **A2** results in the serial number for the day immediately preceding the first day of the current month. For instance, if **A2** contains January 15th, subtracting 15 yields December 31st of the previous year. Therefore, to land precisely on the first day of the month, we must add 1 to this result.

The next critical piece of the puzzle involves determining the weekday of the first day of the month and calculating the necessary offset to reach the first Friday. The full expression that handles this offset calculation is $+8 - \text{WEEKDAY}(A2 - \text{DAY}(A2) + 2)$. The term $A2 - \text{DAY}(A2) + 1$ successfully identifies the first day of the month. The number 2 within the **WEEKDAY** function adjusts the standard calculation to ensure that Monday is recognized as day 1 and Sunday as day 7. However, the formula utilizes $+2$ inside the **WEEKDAY** function because the calculation is subtly geared toward finding the offset from the end of the previous month. The **WEEKDAY** function returns a number between 1 and 7 representing the day of the week.

The final addition of 8, coupled with the subtraction of the calculated **WEEKDAY** value, is the heart of the solution. The constant 8 is used because it combines the necessary offset to get to the first day of the month (which is handled slightly differently in this specific formulation compared to simply adding 1) and ensures that the result lands on a Friday. Friday is represented by the number 6 in Excel's standard **WEEKDAY** system (where Sunday=1). By performing $8 - \text{WEEKDAY}(\dots)$, we effectively calculate how many days must be added to the start-of-month reference point to land precisely on the sixth day (Friday) of the first week of that month. This intricate combination of subtraction and addition, leveraging the serial nature of [Excel Date Serial Numbers](#), ensures the result is always accurate, regardless of which day the month begins on.

Step-by-Step Implementation: Setting Up the Data Range

To demonstrate the practical application of this powerful formula, let us consider a typical scenario involving a list of various dates spanning several months or years. This exercise requires a structured setup within your spreadsheet environment, typically involving a column dedicated to the source dates and an adjacent column where the results--the first Friday of the corresponding month--will be displayed. For this example, we assume our list of source dates begins in cell **A2** and extends downward. Establishing a clear data structure is the first essential step in any advanced spreadsheet calculation.

Suppose we have the following list of varied input dates in column A of our [Excel](#) workbook. These dates serve as the reference points for our calculation; the formula must interpret the month and year of each date individually to determine the correct target date.

	A	B	C	D	E	F
1	Date					
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 will utilize column B to house the results. It is important to ensure that column B is formatted correctly as a **Date** field, as the output of the formula is a **Date Serial Number**. If the column is left in the default 'General' format, the output will appear as a large number (e.g., 45000), which represents the calculated serial date. Applying the appropriate date formatting (e.g., mm/dd/yyyy or custom format) is crucial for the results to be displayed in a human-readable calendar format.

Applying and Interpreting the First Friday Formula

Once the data is set up in column A, we can proceed to implement the calculation. We initiate the process by typing the following complete formula into cell **B2**. This cell corresponds directly to the first reference date in **A2**. The formula's structure ensures that it correctly identifies the month associated with the date in **A2** and calculates the precise serial number for the first Friday of that month.

=A2-DAY(A2)+8-WEEKDAY(A2-DAY(A2)+2)

Upon entering the formula and pressing Enter, cell B2 will immediately display the date corresponding to the first Friday of the month referenced in A2. The power of relative referencing in [Excel](#) allows for rapid application across the entire dataset. By clicking on the fill handle (the small

square in the bottom-right corner of cell B2) and dragging the formula downward, we can populate column B with the calculated first Friday dates for every corresponding date in column A. This efficiency is why single-cell formulas are highly valued in spreadsheet automation.

The resulting spreadsheet, after applying the drag-and-fill operation, clearly displays the calculated results. Column B now contains the date of the first Friday for each month represented in column A. This demonstrates the formula's effectiveness across various dates, proving that it accurately handles month boundaries and varying starting days.

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

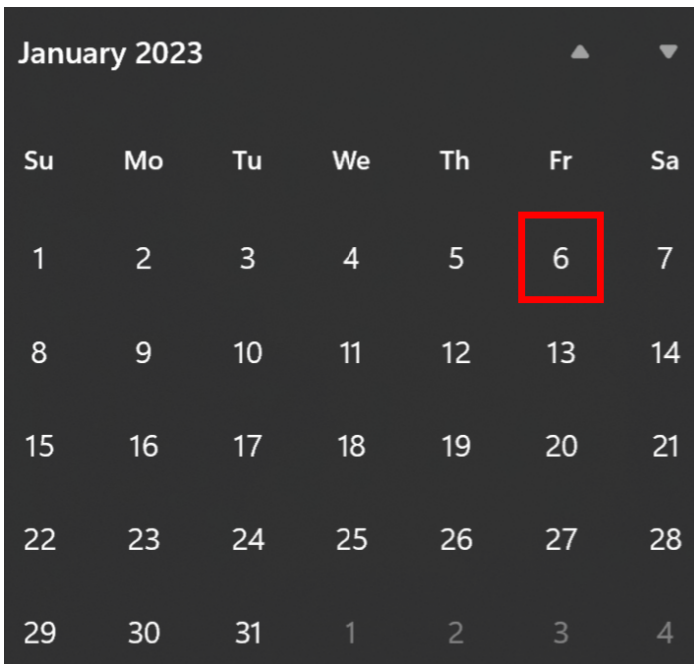
For instance, if the date in **A2** is **1/1/2023** (January 2023), the formula correctly calculates and returns **1/6/2023** in cell **B2**. Similarly, if **A3** contains a date from February 2023, the calculation will yield the first Friday of that subsequent month. The consistency and speed of this calculation are paramount for data analysts and scheduler managers who require reliable, automatically generated dates.

Verifying the Results Against a Calendar

While the mathematical logic of the [Excel](#) formula is sound, it is always best practice to verify the output, especially when dealing with complex date [formulas](#). This verification step confirms that the calculated serial number correctly translates into the desired calendar day. Let's take the first date, **1/1/2023**, which corresponds to the month of January 2023. Our formula determined that the first

Friday of this month is **1/6/2023**.

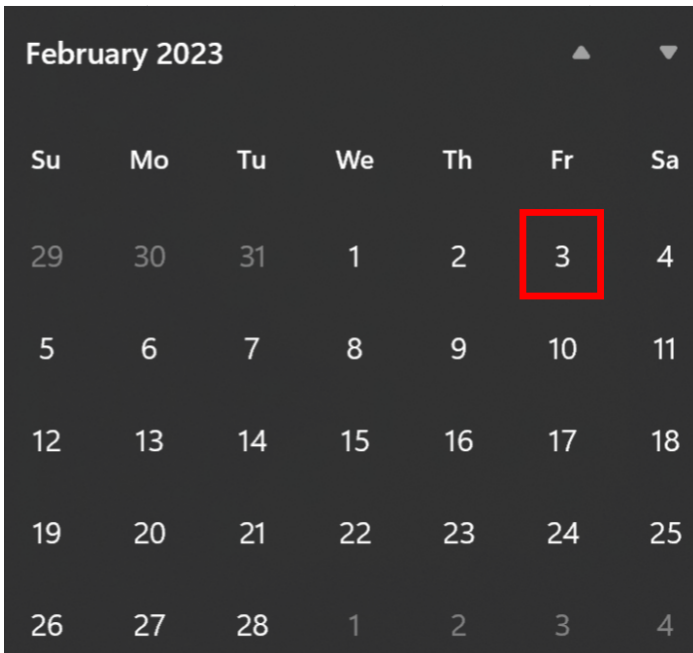
By referring to a standard Gregorian calendar for January 2023, we can visually confirm the calculation. January 1st fell on a Sunday, meaning the first full week of the month began on Monday, January 2nd. Counting forward, Friday naturally falls on January 6th, confirming the accuracy of the formula's output.



The image shows 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 6 is highlighted with a red box, indicating it is the first Friday of the month.

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

We can perform a similar verification for the next entry. Suppose the date in **A3** is **2/1/2023**, corresponding to February 2023. The formula yields **2/3/2023** as the first Friday. Consulting the calendar for February 2023 reveals that February 1st was a Wednesday. Counting forward, the first Friday is indeed February 3rd. This consistent accuracy highlights the robust nature of the formula's design, which correctly adjusts for the varying starting days of different months.



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

This verification process assures us that every date generated in column B accurately represents the first Friday of the month for its corresponding date in column A. This rigorous confirmation is essential when deploying automated scheduling or reporting systems that rely on these precise date calculations.

Extending the Logic: Finding Other Specific Weekdays

The foundational logic used to find the first Friday can be easily adapted to find the first occurrence of any other weekday within a month. The key adjustments lie in the constants used within the **WEEKDAY** calculation, specifically the target day of the week, which is intrinsically linked to the **WEEKDAY** function's return type. While the specific formula presented here is optimized for Friday, understanding the underlying mechanism allows for modification.

The standard [WEEKDAY function](#) typically assigns 1 to Sunday and 7 to Saturday. Since Friday corresponds to the value 6, the constants in our original formula were carefully chosen to target this specific numerical offset. If, for instance, you needed to find the first Monday of the month (which corresponds to 2 in the default system or 1 in the system where Monday=1), the constant values within the overall formula would need subtle manipulation to shift the offset accordingly. Resources detailing generalized "Nth weekday" formulas often provide charts for these constant adjustments, enabling users to find the first, second, or even last instance of a specific weekday.

Furthermore, this technique is not limited to just finding the first occurrence. Advanced date [formulas](#) can be built upon this foundation using functions like **EOMONTH** (End of Month) to calculate the last day of the month and then work backward to find the last specific weekday. For

example, finding the last working day of the month often involves similar serial date manipulation combined with the **WORKDAY** function. Mastering the components of the first Friday formula is therefore an excellent gateway to solving a wide range of complex temporal calculation problems within [Excel](#).

Additional Resources for Date and Time Manipulation

For those seeking to further enhance their proficiency in handling date and time data within spreadsheets, several tutorials and functions provide deeper insight into Excel's temporal capabilities. Understanding how the [Excel Date Serial Number](#) system works is fundamental to mastering complex calculations beyond simple addition or subtraction.

Mastery of the following functions is highly recommended for any user frequently dealing with scheduling or financial reporting:

EOMONTH: Calculates the last day of the month, often used as a starting point for finding the last weekday of a month.

EDATE: Returns the serial number of the date that is a specified number of months before or after the start date.

NETWORKDAYS: Calculates the number of whole working days between two dates, excluding weekends and specified holidays.

The following tutorials explain how to perform other common operations in [Excel](#):

How to Calculate the Number of Working Days in a Month.

Using Conditional Formatting with Dates to Highlight Weekends.

Creating Dynamic Calendars with Date Functions.