

Finding the First Monday of Each Month Using Excel Formulas

Authored by
Mohammed looti

November 10, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Finding the First Monday of Each Month Using Excel Formulas*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15442>

Mastering complex [date calculation](#) in [Excel](#) is a common requirement for financial reporting, scheduling, and project management. One frequent challenge is accurately determining the first specific weekday of any given month--for instance, the first Monday. While this task might seem complicated, [Excel](#) provides a powerful and elegant [formula](#) that combines several essential date functions to achieve this result dynamically.

This tutorial provides a complete guide to using and understanding this specific [formula](#), ensuring you can apply this technique across various datasets. This method relies on establishing a reliable anchor date within the month and then using the **WEEKDAY** function to precisely calculate the offset required to arrive at the desired Monday.

The Essential Formula for Date Calculation

The following combined function is the core solution for locating the first Monday of the month referenced by a specific date. This particular structure is designed to be robust, handling months of varying lengths and starting days without error.

```
=DATE(YEAR(A2),MONTH(A2),7)-WEEKDAY(DATE(YEAR(A2),MONTH(A2),7),3)
```

In this specific illustration, the [formula](#) is configured to reference the date contained within cell **A2**. It will extract the year and month from **A2** and then calculate the exact date of the first Monday within that period. This structure ensures that no matter what day of the month the source date (in **A2**) falls on, the calculation remains accurate.

Understanding this [formula](#) is crucial for efficient data management. It leverages the power of nested functions: the inner **DATE** functions establish a common anchor point (the 7th day of the month), and the outer calculation adjusts this anchor based on the day of the week, effectively "rolling back" to the first Monday.

Step-by-Step Example Implementation in Excel

To demonstrate the practical application of this powerful date calculation [formula](#), let us consider a scenario where we have a column containing a list of various dates spanning different months and years. Our objective is to generate a corresponding column that displays the first Monday for the month associated with each date listed.

Suppose we begin with the following dataset in [Excel](#), with the source dates located in Column A:

	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						

To initiate the process, we must enter the core calculation into cell **B2**. This cell will correspond directly to the date listed in **A2**. We type the complete [formula](#), ensuring the cell references (specifically **A2**) are correctly maintained:

```
=DATE(YEAR(A2),MONTH(A2),7)-WEEKDAY(DATE(YEAR(A2),MONTH(A2),7),3)
```

Upon pressing Enter, the result in **B2** will immediately display the first Monday of January 2023. Given the use of relative referencing in [Excel](#), we can efficiently apply this calculation to the rest of the dataset. The next step involves clicking on cell **B2** and dragging the formula down using the fill handle to populate the remaining cells in column B. This action automatically adjusts the cell reference (e.g., A2 becomes A3, A4, etc.) for each subsequent row.

Analyzing the Results and Verification

Once the [formula](#) has been successfully applied to all relevant rows, Column B will be populated with the date of the first Monday corresponding to the month and year derived from the date in Column A. This resulting output demonstrates the efficiency of using calculated fields for scheduling tasks that depend on specific weekdays.

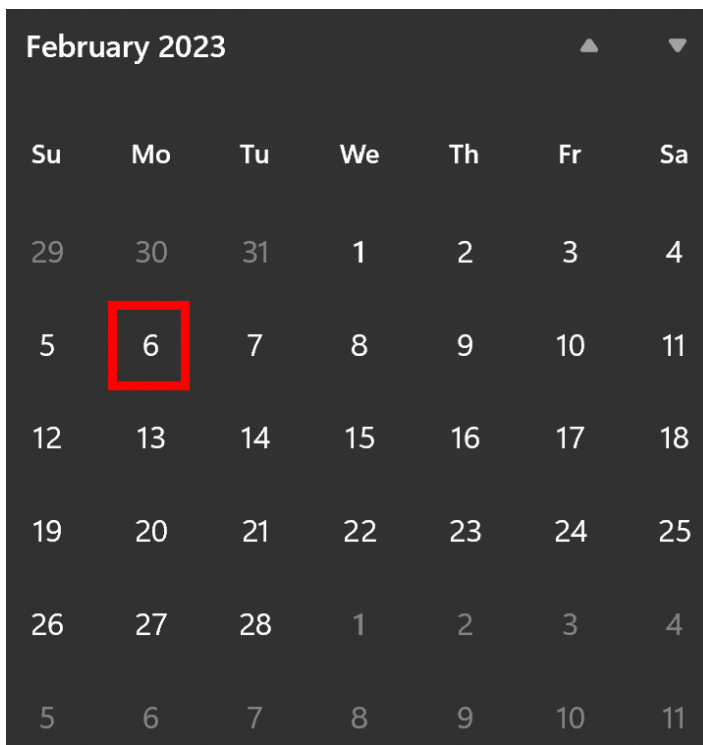
B2		=DATE(YEAR(A2),MONTH(A2),7)-WEEKDAY(DATE(YEAR(A2),MONTH(A2),7),3)									
	A	B	C	D	E	F	G	H	I		
1	Date	First Monday									
2	1/1/2023	1/2/2023									
3	2/1/2023	2/6/2023									
4	3/1/2023	3/6/2023									
5	4/1/2023	4/3/2023									
6	5/1/2023	5/1/2023									
7	6/1/2023	6/5/2023									
8	7/1/2023	7/3/2023									
9	8/1/2023	8/7/2023									
10	9/1/2023	9/4/2023									
11	10/1/2023	10/2/2023									
12	11/1/2023	11/6/2023									
13	12/1/2023	12/4/2023									
14											
15											
16											

It is essential to verify the accuracy of the results, especially when dealing with complex date logic. Let us examine the first entry: the original date of **1/1/2023** refers to January 2023. The calculated result in column B is **1/2/2023**. By referencing a standard calendar for January 2023, we can confirm that January 2nd was indeed the first Monday of that 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

Moving to the second entry, the source date **2/1/2023** refers to February 2023. The calculated result is **2/6/2023**. A quick calendar verification confirms this result as well, proving the reliability

and accuracy of the underlying date logic embedded within the [formula](#). This verification process is critical for maintaining data integrity in large-scale spreadsheet applications.



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
5	6	7	8	9	10	11

Deconstructing the Formula: A Detailed Explanation

To truly appreciate the elegance of this solution, we must break down the components of the core calculation. Recall the structure used to determine the first Monday of the month anchored by the date 1/1/2023 in [cell A2](#):

=DATE(YEAR(A2),MONTH(A2),7)-WEEKDAY(DATE(YEAR(A2),MONTH(A2),7),3)

The entire process is predicated on two fundamental steps: first, establishing a guaranteed anchor date within the first seven days of the month, and second, determining the necessary offset to rewind from that anchor date back to the first Monday. This method avoids the complexity of trying to determine the starting day of the month directly, which can lead to more convoluted conditional statements.

The first part of the [formula](#), `DATE(YEAR(A2),MONTH(A2),7)`, utilizes the [DATE function](#) combined with the **YEAR** and **MONTH** functions. This construction efficiently extracts the year and month from the source date in **A2**, but crucially, it forces the day argument to **7**. Why the 7th? Because the first Monday of any month must fall on a date between the 1st and the 7th. By setting the anchor at the 7th, we ensure that the anchor date is always in the same week or later than the first

Monday we are trying to find. For 1/1/2023, this entire expression returns **1/7/2023** (January 7, 2023).

The second, subtracted portion, `WEEKDAY(DATE(YEAR(A2), MONTH(A2), 7), 3)`, is responsible for calculating the necessary rollback. It takes our anchor date (the 7th) and determines its day number using the [WEEKDAY function](#). The key here is the optional argument, **3**. The argument 3 specifies a return type where the week numbering starts at 0 for Monday and ends at 6 for Sunday. For our anchor date of 1/7/2023 (which is a Saturday), the [WEEKDAY function](#) with argument 3 returns **5**. This means the 7th of January is the 5th day in the Monday-based numbering system (Monday=0, Saturday=5).

Finally, we perform the subtraction: **1/7/2023 - 5**. Since Excel stores dates as serial numbers, subtracting 5 days from the date 1/7/2023 effectively rolls the date back five days, landing us precisely on **1/2/2023**--the first Monday of the month. This robust logic ensures that regardless of whether the 7th is a Monday (returning 0) or a Sunday (returning 6), the correct number of days is subtracted to arrive at the preceding or concurrent Monday. This same logical structure is applied consistently to find the first Monday for every date listed in the dataset.

Understanding Key Date Functions: DATE, YEAR, MONTH, and WEEKDAY

A sophisticated [formula](#) like this one relies heavily on the correct interaction of fundamental [Excel](#) date and time functions. Mastering these individual components provides the foundation necessary for developing more complex chronological analyses.

The YEAR and MONTH Functions: These functions are straightforward extractors. The `YEAR(serial_number)` function takes an [Excel](#) serial date number and returns the year as a four-digit integer. Similarly, `MONTH(serial_number)` returns the month as an integer from 1 (January) to 12 (December). In our formula, these ensure that we isolate the correct chronological context from the source date in [cell A2](#), regardless of the day of the month it happens to be.

The DATE Function: The [DATE function](#) constructs a valid date from three separate numerical arguments: `DATE(year, month, day)`. By embedding the **YEAR** and **MONTH** functions inside the **DATE** function and manually specifying the day as 7, we generate a new date object that serves as our reliable anchor point within the target month. This function is essential because it allows us to manipulate date components independently before combining them back into a valid [Excel](#) serial date.

The WEEKDAY Function: This function is arguably the most critical component for determining specific weekdays. `WEEKDAY(serial_number,)` returns a number from 1 to 7 identifying the day of the week of a date. The optional **return_type** argument dictates how the week is numbered. We specifically use **3** for the return type, which assigns 0 to Monday and 6 to Sunday. This numbering

convention is ideal for date arithmetic because subtracting the resulting number (0 through 6) from the anchor date guarantees a rollback directly to the preceding or current Monday. For example, if the anchor date is already a Monday, WEEKDAY returns 0, and subtracting 0 leaves the date unchanged.

Further Resources for Advanced Excel Date Manipulation

The technique demonstrated here provides a robust framework for solving scheduling and reporting requirements that rely on specific weekdays. By understanding how the **DATE** and [WEEKDAY function](#) arguments interact, users can adapt this methodology to find the first (or last) specific weekday (e.g., the last Friday, the third Tuesday) of any month.

For those seeking to expand their proficiency in handling date and time data within spreadsheets, consulting the official documentation for the primary functions is highly recommended. The official Microsoft documentation provides comprehensive details on all return types and edge cases, ensuring optimal application in complex scenarios.

The following tutorials explain how to perform other common tasks in [Excel](#), building upon the foundational knowledge of date arithmetic:

Tutorial on Finding the Last Day of the Month in Excel.

Guide to Calculating Business Days Between Two Dates Using NETWORKDAYS.

Advanced Techniques for Conditional Formatting Based on Dates.