

Understanding the Google Sheets Formula for Finding the First Monday of a Month

Authored by
Mohammed loot

November 11, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Understanding the Google Sheets Formula for Finding the First Monday of a Month*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16639>

One common challenge in [Google Sheets](#) data management involves calculating specific dates, such as the [formula](#) needed to determine the very first Monday occurring in any given month. This operation is vital for scheduling reports, automated payroll systems, or planning monthly events based on a fixed weekday schedule. While seemingly complex, a combination of core date functions provides an elegant and highly reliable solution. The structure of this calculation relies on finding a date guaranteed to fall within the first week (the 7th day) and then using the [WEEKDAY function](#) to adjust back to the target Monday.

The following powerful formula is designed to dynamically identify the first Monday of the month based on any reference date provided in a specified cell. This method is superior to manually checking calendars as it scales immediately across large datasets, ensuring accuracy regardless of the year or month referenced. It leverages the inherent numerical representation of dates within spreadsheet applications to perform precise date arithmetic, which is the foundation of efficient spreadsheet automation.

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

In this specific instance, the calculation is parameterized to analyze the date contained within cell **A2**. The result returned by this formula will be the standardized serial date corresponding to the first Monday of the month associated with the date in **A2**. Understanding the mechanics of how this combination of [DATE](#), [YEAR](#), [MONTH](#), and [WEEKDAY](#) functions work together is crucial for anyone looking to master advanced date management techniques in Google Sheets.

Step-by-Step Example: Implementing the Formula

To illustrate the practical application of this complex date formula, let us assume we are working with a dataset that contains various dates spread across different months and years. Our objective is to generate a parallel column where each corresponding entry represents the first Monday of the month from the original date. This task requires careful placement and subsequent replication of the formula across the range.

Suppose we are presented with the following list of random dates within our Google Sheets interface. These dates serve as the input for our calculation, residing in column A.

	A	B	C	D
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				

We initiate the process by typing the complete formula into cell **B2**. This cell will correspond to the date found in **A2**. The absolute precision of the formula ensures that even if the date in **A2** (e.g., 1/1/2023) is a Sunday, the formula will correctly identify the next day (1/2/2023) as the first Monday, or if the date were near the end of the month, it would still anchor to the correct Monday within that same month.

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

Once the formula is correctly entered in **B2**, we utilize the fill handle feature--the small square at the bottom-right corner of the selected cell--to drag the formula down. This action automatically applies the necessary relative referencing adjustments (e.g., changing A2 to A3, A4, and so on) to calculate the first Monday for every subsequent date listed in column A, thereby automating the extraction process for the entire dataset. This efficiency is paramount when dealing with extensive records.

B2 `=DATE(YEAR(A2), MONTH(A2), 7) - WEEKDAY(DATE(YEAR(A2), MONTH(A2), 7), 3)`

	A	B	C	D	E	F
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						

Upon completing the fill operation, column B now furnishes us with the exact date of the first Monday for every month referenced in column A. For example, considering the initial date of **1/1/2023**, which is the start of January 2023, the calculated output in column B should be **1/2/2023**. We can verify this result against a standard calendar to confirm that the calculation is accurate and robust across different monthly starting conditions.

Validating the Results Through Calendar Reference

Verification is a critical step in any complex spreadsheet operation, ensuring that the derived dates align with real-world calendar logic. The formula's output must be rigorously checked, especially since date arithmetic can be prone to error if the underlying logic is flawed. We can manually confirm the results generated by the spreadsheet against static calendar images for the relevant months, providing a visual confirmation of the calculated dates.

Taking the input date **1/1/2023**, the formula yields **1/2/2023**. As 1/1/2023 was a Sunday, 1/2/2023 is indeed the first Monday of January. This confirms that the formula correctly identifies the starting day of the week and adjusts accordingly.

January 2023

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 next entry, **2/1/2023**, which anchors the month of February 2023, the calculated result is **2/6/2023**. A quick check of the February calendar confirms that February 1st was a Wednesday, and therefore, the sixth day of the month corresponds precisely to the first Monday. This iterative confirmation process demonstrates the reliability of the formula across various starting days within the month.

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 Logic: How the Formula Pinpoints Monday

To fully appreciate the elegance of this solution, we must dissect the two main components of the calculation. The overall formula is structured as a subtraction: (A) a fixed date guaranteed to fall within the first week of the target month, minus (B) an offset value that aligns that fixed date back to the desired weekday (Monday). Recall the full [formula](#) used for the date 1/1/2023:

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

The first major component is responsible for establishing a fixed anchor point. This is achieved using **DATE(YEAR(A2), MONTH(A2), 7)**. The nested [YEAR function](#) and [MONTH function](#) extract the year and month from the reference date in A2. By setting the day argument of the [DATE function](#) to '7', we guarantee that the resulting date--in the case of 1/1/2023, this is **1/7/2023**--will fall within the first seven days of the target month. This date is crucial because it ensures we are working within the correct monthly boundary.

The second component, which is subtracted from the anchor date, is the offset calculated by the [WEEKDAY function](#). This function determines how many days the anchor date (1/7/2023) is past the desired starting day (Monday). If 1/7/2023 is a Saturday, the function needs to calculate an offset that, when subtracted, results in the preceding Monday (1/2/2023). This calculation is only possible because dates in spreadsheets are stored as serial numbers, allowing for simple arithmetic manipulation.

Mastering the WEEKDAY Function and Return Types

The key to this entire operation lies in the precise configuration of the [WEEKDAY function](#). The standard WEEKDAY function returns a number corresponding to the day of the week, typically starting with 1 for Sunday or 2 for Monday, depending on the optional argument provided. However, our formula utilizes a specific return type argument, **3**, which is fundamentally important for achieving the correct subtraction offset.

When the optional argument is set to **3**, the WEEKDAY function specifies that the days are numbered starting from 0 (Monday) through 6 (Sunday). This is known as the ISO 8601 standard numerical system for days of the week. Therefore:

Monday = 0

Tuesday = 1

...

Saturday = 5

Sunday = 6

Applying this to our example, the date **1/7/2023** is a Saturday. When the WEEKDAY function evaluates 1/7/2023 using the return type 3, it returns the value **5**. This value of 5 represents the number of days we must subtract from the anchor date (1/7/2023) to arrive at the preceding Monday (the day coded as 0).

The final step in the [formula](#) is the subtraction: Anchor Date (1/7/2023) minus Offset (5 days). Since 1/7/2023 is day 7 of the month, subtracting 5 days results in day 2 of the month, which is **1/2/2023**--the first Monday of January. This process is systematically repeated for every date in column A, ensuring that regardless of where the 7th day falls, the correct offset is calculated to land precisely on the first Monday.

Practical Applications and Efficiency Considerations

The ability to reliably calculate the first instance of a specific weekday within a month opens up numerous possibilities for advanced spreadsheet users and data analysts. This technique is not limited to finding just the first Monday; by adjusting the return type argument in the WEEKDAY function, or by altering the day subtraction logic slightly, one could find the first Tuesday, the third Friday, or even the last Sunday of a month. This customization capability transforms a single formula into a versatile scheduling tool.

For organizations that rely on monthly reporting cycles, such as financial closes, board meeting schedules, or regulatory submission deadlines that are often tied to the first weekday of the month, this formula provides an unparalleled level of automation. Instead of manually updating critical dates each month, a dashboard linked to this calculation remains perpetually accurate. Furthermore, integrating this calculation into larger [Google Sheets](#) workflows, perhaps combined with conditional formatting or automated alerts, significantly enhances operational efficiency.

It is important to note that while this formula is robust, users should always ensure their input data (column A) is correctly formatted as a date serial number. If the input data is stored as text, the [DATE function](#) components (YEAR and MONTH) will fail, resulting in an error. Always verify data types when troubleshooting complex date arithmetic.

Further Resources for Spreadsheet Mastery

Mastering date and time functions is a cornerstone of advanced spreadsheet proficiency. The techniques demonstrated here can be adapted to solve a wide variety of time-based challenges. For those seeking to deepen their knowledge of date manipulation and other complex calculations within Google Sheets, consulting the official documentation is highly recommended.

The following tutorials explain how to perform other common tasks in Google Sheets, often utilizing similar principles of sequential function nesting and date serial number arithmetic:

Finding the last day of the month using EOMONTH.

Calculating the number of working days between two dates using NETWORKDAYS.

Advanced techniques for filtering data based on specific weekdays.