

Calculating the Last Day of the Week in Excel: A Step-by-Step Guide

Authored by
Mohammed loot

November 10, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Calculating the Last Day of the Week in Excel: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16069>

Introduction to Advanced Date Manipulation in Excel

Mastering date manipulation is a fundamental skill for effective data analysis and comprehensive financial reporting within [Microsoft Excel](#). Data professionals frequently encounter scenarios requiring the consolidation or grouping of information based on precise time boundaries, such as weeks, fiscal quarters, or reporting months. Among the most critical requirements is the ability to accurately determine the exact **last day of the week** corresponding to any given input date. This capability is indispensable for tasks ranging from calculating weekly performance summaries and defining billing cycle cutoffs to aligning complex project milestones. Since Excel does not include a single, dedicated function for this specific boundary calculation, we must skillfully combine core date functions, primarily the [WEEKDAY function](#), with foundational arithmetic logic to derive this crucial calendar value with high precision.

The primary complexity in this operation stems from the varying definitions of a week's structure across different organizational and geographical contexts. In certain international standards or specific business environments, the week might officially conclude on Saturday, while in others--particularly in North American data formats--Sunday is designated as the endpoint. Fortunately, the robust nature of [Excel](#) allows for flexible accommodation of both conventions. The formulas we will explore below provide precise, customizable methods for calculating the last day of the week, irrespective of whether the organizational standard mandates a Saturday or a Sunday conclusion. This accuracy is rooted in the way Excel internally manages time: all dates are stored as sequential [serial numbers](#), which makes date arithmetic exceptionally reliable and simple to implement across any worksheet.

Understanding the Core Tool: The WEEKDAY Function

The foundation for nearly all sophisticated date boundary calculations in Excel is the [WEEKDAY function](#). This powerful function serves to translate a specific date into a corresponding numerical value, ranging from 1 to 7, which represents its position within the week. The exact numbering convention--that is, which day is assigned the number 1 and which is assigned 7--is entirely dictated by the function's optional second argument, which is known as the **return_type**. A clear understanding of how the return type calibrates the week is absolutely essential for tailoring the formula to identify the correct week-end day, whether that target is Saturday or Sunday.

The standard syntax for the function is `WEEKDAY(serial_number,)`. The `serial_number` is the reference to the cell containing the [date value](#), and the optional `return_type` defines the week's starting point and, consequently, how the subsequent days are numerically labeled. For example, if the `return_type` is omitted or set to 1 (the default), Sunday is assigned the value 1, and Saturday is assigned 7. Conversely, if the `return_type` is set to 2, Monday receives the value 1, and Sunday is labeled 7. This seemingly minor difference in numbering scheme is precisely what

enables us to adapt a single arithmetic structure to fit different weekly definitions effortlessly.

To successfully calculate the last day of the week, the process involves two logical steps. First, we determine the input date's current position using `WEEKDAY()`. Second, we use this numerical position to calculate precisely how many days must be added to or subtracted from the original date to arrive at the specified end-of-week marker. The subsequent formulas illustrate how we leverage this numerical representation of the day (the difference from the start of the week) to jump accurately to the Saturday or Sunday that concludes the current seven-day period containing the input date.

Method 1: Calculating the Last Day (Assuming Week Ends on Saturday)

When your data governance or reporting standards dictate that the week must conclude on a **Saturday**, we utilize the standard, default behavior of the [WEEKDAY function](#). In this default mode (where the return type is 1 or omitted), the week begins on Sunday (1) and concludes on Saturday (7). If the source date for which the boundary needs to be found is located in cell **A2**, the formula below operates by subtracting the numerical value of the current day and then adding exactly 7 days. This sequence ensures a precise alignment with the Saturday concluding the week.

The required formula, designated as Formula 1, is structured as follows, assuming the date of interest is contained within the cell **A2**:

=A2-WEEKDAY(A2)+7

A detailed breakdown of this calculation reveals its elegant logic. The expression `WEEKDAY(A2)` yields a number from 1 (Sunday) to 7 (Saturday). By performing the subtraction `A2 - WEEKDAY(A2)`, you effectively step back to the date corresponding to the previous Saturday. Specifically, you arrive at the Saturday immediately preceding the Sunday start of the current week. Therefore, adding 7 days (+7) reliably pushes the date forward by exactly one full week, landing precisely on the Saturday that marks the end of the current week boundary. This methodology is the most reliable approach for systems where the conventional week traditionally begins on Sunday and ends on Saturday.

Method 2: Calculating the Last Day (Assuming Week Ends on Sunday)

For reporting systems, financial models, or data sets where the official week concludes on a **Sunday**, we must strategically modify the calculation. This adjustment is achieved by employing the crucial `return_type` argument within the [WEEKDAY function](#). By explicitly setting the return type to 2, we redefine the numbering scheme so that Monday is assigned 1 and Sunday is assigned 7. This recalibration is vital because it allows us to apply the same core mathematical

subtraction and addition logic used in Method 1 while guaranteeing that Sunday is correctly identified as the seventh and final day of the cycle.

The formula below, designated as Formula 2, incorporates the return type 2, ensuring that the calculation accurately targets Sunday as the conclusion of the weekly cycle. Like the previous example, this formula also assumes the source date is located in cell **A2**:

=A2-WEEKDAY(A2,2)+7

In this specific context, the function `WEEKDAY(A2, 2)` returns a value where Monday corresponds to 1 and Sunday corresponds to 7. When we subtract this value from the original [date](#), the result lands exactly on the date of the previous Monday minus one day--which is the date of the previous Sunday. By subsequently adding 7, we jump forward precisely one complete week, thereby landing on the Sunday that concludes the current week containing the original date in A2. Both formulas are engineered for maximum efficiency, requiring only the original date reference in cell **A2** to determine the appropriate end-of-week boundary.

Practical Implementation Example in Excel

To fully grasp the operational efficiency of these specialized formulas, let us walk through a concrete practical example using a typical column of sample dates. Imagine we have a list of transaction dates beginning in cell **A2**, and our objective is to populate two adjacent columns (B and C) with the corresponding last day of the week, calculated according to both the Saturday-end and Sunday-end conventions. The initial data set includes various dates that require weekly boundary determination:

The starting point for our analysis, showing the dates to be analyzed in Column A, is depicted below:

	A	B	C	D	E
1	Date				
2	1/10/2023				
3	1/15/2023				
4	2/15/2023				
5	3/5/2023				
6	4/1/2023				
7	4/29/2023				
8	6/7/2023				
9	7/15/2023				
10	9/12/2023				
11	10/30/2023				
12	12/25/2023				
13					
14					
15					
16					
17					

We initiate the calculation process by carefully entering the required formulas into the header row of our dedicated calculation columns. We place the Saturday-end formula into cell **B2** and the Sunday-end formula into cell **C2**, ensuring that both correctly reference the starting date located in **A2**:

B2: =A2-WEEKDAY(A2)+7

This formula utilizes the default return type (1) to accurately identify the last day, based on the convention that Saturday concludes the week.

C2: =A2-WEEKDAY(A2,2)+7

This formula employs return type 2, which is necessary to correctly determine the last day under the assumption that Sunday is the designated week end.

Once the initial formulas have been correctly entered into cells B2 and C2, the subsequent step is to efficiently apply these calculations across the entirety of the dataset. This is achieved using Excel's powerful [fill handle](#) functionality--simply clicking and dragging the formulas down from the initial cells to cover all corresponding rows in columns B and C. This action automatically adjusts the relative cell references (e.g., A2 transitions to A3, A4, and so forth) for each row, thereby calculating the appropriate last day for every date listed in column A, without manual intervention.

The resulting spreadsheet, after the drag-and-fill operation has been completed, clearly displays the calculated last day for each date based on both defined weekly assumptions:

C2 ✕ ✓ <i>fx</i> =A2-WEEKDAY(A2,2)+7				
	A	B	C	D
1	Date	Last Day of Week (Saturday)	Last Day of Week (Sunday)	
2	1/10/2023	1/14/2023	1/15/2023	
3	1/15/2023	1/21/2023	1/15/2023	
4	2/15/2023	2/18/2023	2/19/2023	
5	3/5/2023	3/11/2023	3/5/2023	
6	4/1/2023	4/1/2023	4/2/2023	
7	4/29/2023	4/29/2023	4/30/2023	
8	6/7/2023	6/10/2023	6/11/2023	
9	7/15/2023	7/15/2023	7/16/2023	
10	9/12/2023	9/16/2023	9/17/2023	
11	10/30/2023	11/4/2023	11/5/2023	
12	12/25/2023	12/30/2023	12/31/2023	
13				
14				
15				
16				
17				
18				

Formatting Date Results and Conclusion

A final, yet critical, step in any Excel date manipulation workflow is guaranteeing that the output is formatted in a way that is immediately understandable and readable. If the results displayed in columns B and C initially appear as large, unintelligible **numeric values** (such as 44574), this is not an error but an expected behavior. As previously noted, Excel stores all dates as sequential **serial numbers**, counting the days elapsed since January 1, 1900. When the cell formatting defaults to "General," the underlying serial number is displayed instead of the human-readable date format.

To rectify this visual issue, we must apply the appropriate **Number Format**. Start by highlighting the entire calculated range, which in this demonstration is **B2:C11**. Navigate to the **Home** tab on the Excel ribbon, locate the **Number Format** dropdown menu (usually showing "General"), and select either **Short Date** or **Long Date** from the available options. This conversion step is absolutely essential for accurate interpretation and presentation of the analysis.

Once the formatting has been successfully applied, every numeric value in columns B and C will

be correctly converted into a standard date format. Column B now shows the derived last day of the week based on a Saturday endpoint, while Column C displays the last day based on a Sunday endpoint. For instance, considering the first date in the list (1/10/2022):



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
5	6	7	8	9	10	11

If we adhere to **Saturday** as the last day of the week, the correctly derived date is **1/15/2022**. Conversely, if we define **Sunday** as the end of the week, the derived date is **1/16/2022**. These consistent and verifiable results confirm that the formulas successfully identify the appropriate end-of-week boundary based on the specified organizational definition, establishing them as highly flexible and reliable tools for any analyst managing complex time-series data.

Additional Resources for Advanced Chronological Operations

Proficiency in date and time calculations, as demonstrated here, is a foundational element of advanced [Excel](#) functionality. The techniques discussed--particularly the precise manipulation of serial dates and the expert use of specialized functions like [WEEKDAY](#)--form the bedrock for executing far more complex chronological analyses. These analyses can include calculating specific working days, accurately determining the elapsed time between project milestones, or forecasting future dates based on established business cycles.

For users committed to expanding their proficiency in date manipulation and other common data operations crucial for effective reporting, the following tutorials provide valuable guidance on related advanced Excel capabilities: