

Combining Date and Time in Google Sheets: A Step-by-Step Guide

Authored by
Mohammed looti

November 10, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Combining Date and Time in Google Sheets: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16164>

When handling extensive datasets or preparing critical information for advanced [time-series analysis](#), data integrity often demands the merging of separate date and time entries, typically housed in two distinct spreadsheet columns, into a single, unified cell. This process of consolidation is not merely cosmetic; it is fundamentally crucial for accurate calculation of durations, seamless chronological sorting, and ensuring robust data compatibility with external processing systems. Within the powerful environment of [Google Sheets](#), spreadsheet experts utilize several highly efficient and reliable methods to achieve this combination cleanly and efficiently. Understanding the underlying logic of how Sheets handles temporal data is the first step toward mastering these techniques. The following advanced formulas and demonstrations provide clear guidance on how to combine a date and time from separate cells into a single, functional value.

Choosing the correct method--whether prioritizing mathematical functionality or precise presentation format--depends entirely on the end goal of the data. If the combined value needs to be used for complex calculations and data manipulation, a numerical approach is necessary. Conversely, if the data is intended solely for reporting or visual presentation, a text-based concatenation method offers superior formatting flexibility. We will explore both foundational methods in detail, detailing their strengths, limitations, and practical application steps to ensure you can manage your temporal data effectively.

Method 1: The Simple Addition Approach for Numerical Integrity

The most intuitive and mathematically sound technique for merging date and time relies on the core architecture of how [Google Sheets](#) handles and stores temporal data. Unlike simple text strings, both dates and times are stored internally as a single, continuous [Serial Value](#). This unique numerical representation is the key to performing accurate date arithmetic. Specifically, the date component is represented by the integer part of the serial value, quantifying the number of days elapsed since the epoch date (December 30, 1899). The time component, conversely, is represented by the fractional part, which precisely measures the proportion of a 24-hour day that has passed.

Because time is inherently a decimal fraction of a full day (e.g., noon is 0.5, 6 AM is 0.25), the simple act of adding the cell containing the date to the cell containing the time results in a new, singular serial number. This resulting number perfectly encapsulates both the date and the exact time stamp. This method is highly favored in environments where subsequent calculations are necessary, such as determining precise time differences between events, calculating lead times, or sorting extensive logs chronologically. It ensures that the combined value retains its essential numerical integrity, which is vital for any automated data processing or complex analytical modeling.

To implement this powerful yet remarkably simple technique, the user merely needs to reference the two source cells--the date cell and the time cell--and connect them using the standard addition operator (+). This simplicity belies the mathematical sophistication occurring in the background. The resulting formula is highly efficient and easily scalable across vast ranges of data, making it the preferred method for most analytical tasks. The core formula structure remains consistent regardless of the size of the dataset, as demonstrated below:

=A2+B2

It is crucial to remember that while this method maintains numerical accuracy, the displayed format of the resulting cell (C2 in the example above) will default to the standard date and time format settings of your spreadsheet, often displaying time in a 24-hour clock format. If a specific display format (like AM/PM) is required, manual formatting adjustment of the destination cell is necessary after the calculation is complete, typically using the Format menu.

Method 2: Achieving Presentation Precision with CONCATENATE and TEXT

While the simple addition method excels in numerical operations, it often lacks the control required for sophisticated reporting where precise output formatting is mandatory. For scenarios demanding granular control over the output structure--including specific date separators, forced AM/PM indicators, or adherence to ISO standard year formats--the combined use of the [CONCATENATE function](#) (or its shorthand, the ampersand operator, &) and the powerful [TEXT function](#) offers superior flexibility and stylistic control.

The central advantage of employing the **TEXT** function lies in its ability to convert numerical date and time serial values into highly customized text strings. It achieves this by applying specific, user-defined [Date and Time Format](#) codes (e.g., "m/d/yyyy" or "h:mm AM/PM"). Because the output of **TEXT** is, by definition, a text string, we then employ **CONCATENATE** to join these two newly formatted strings--the date string and the time string--together, usually separated by a space or a specific delimiter, ensuring maximum readability and presentation quality.

It is essential to understand the trade-off inherent in this technique: this approach sacrifices the numerical integrity of the combined value. The resulting cell contains a descriptive text string rather than a calculable [Serial Value](#). Consequently, the output cell cannot be reliably used for direct date arithmetic, such as subtracting one time stamp from another to find a duration. However, this is precisely the method needed when presentation is the absolute priority, making it indispensable for generating standardized reports, labels, or data exports where strict stylistic consistency is required.

=CONCATENATE(TEXT(A2,"m/d/yyyy")," ",TEXT(B2,"h:mm AM/PM"))

As a critical prerequisite for both methodologies, it must be emphasized that the examples provided assume the source date data resides in cell **A2** and the time data in cell **B2**. Users must diligently adjust these cell references (e.g., A2, B2) to accurately reflect the actual columnar layout of their respective spreadsheets. The subsequent practical demonstrations will illustrate the powerful application of both techniques using a typical date and time columnar setup in [Google Sheets](#):

	A	B	C
1	Date	Time	
2	1/3/2023	6:17 AM	
3	1/15/2023	3:23 PM	
4	2/23/2023	9:15 PM	
5	3/14/2023	11:59 AM	
6	3/25/2023	4:15 AM	
7	4/1/2023	10:45 PM	
8	5/16/2023	4:37 AM	
9	5/17/2023	5:18 AM	
10			
11			
12			
13			

Practical Demonstration: Combining Date & Time Using Simple Addition

For our initial practical exercise, we will leverage the efficiency and mathematical robustness of the numerical addition method. This approach is highly recommended when the resulting timestamp is intended to serve as a functional data point, maintaining its core mathematical properties for future complex temporal calculations or seamless chronological sorting within large data arrays. The goal is to produce a combined [Serial Value](#) that Google Sheets can immediately recognize and process as a single point in time.

The process begins by identifying an empty target cell, typically cell **C2**, where the combined value will first appear. In this cell, we input the simplest possible formula, directing Google Sheets to calculate the sum of the date value from **A2** and the time value from **B2**. Upon execution, this formula instantly generates a unified timestamp. This instant consolidation is a testament to the efficient handling of serial values within the application, ensuring the mathematical integrity of the combined data remains intact.

=A2+B2

Once the core formula is successfully entered into **C2**, the calculation must be swiftly propagated across the entire dataset. This is most efficiently accomplished by utilizing the spreadsheet's built-in **fill handle**. By clicking on the small square at the bottom-right corner of cell C2 and dragging the formula down through the remaining rows in column C, the system automatically adjusts the relative cell references (e.g., A2+B2 correctly becomes A3+B3, then A4+B4, and so on). This automation quickly populates the entire target column with the consolidated date and time for every record.

Upon completion of the fill process, column C now displays the complete date and time in one cell for each row. A critical observation involves the default formatting: when using simple addition, Google Sheets frequently defaults to omitting AM or PM indicators, displaying the hours using a [24-hour clock](#) format. While mathematically precise, users who specifically require the 12-hour AM/PM formatting must manually apply custom number formatting settings to the cells in column C to meet their reporting needs.

C2 ▾ *fx* =A2+B2

	A	B	C	D
1	Date	Time	Date & Time	
2	1/3/2023	6:17 AM	1/3/2023 6:17:00	
3	1/15/2023	3:23 PM	1/15/2023 15:23:00	
4	2/23/2023	9:15 PM	2/23/2023 21:15:00	
5	3/14/2023	11:59 AM	3/14/2023 11:59:00	
6	3/25/2023	4:15 AM	3/25/2023 4:15:00	
7	4/1/2023	10:45 PM	4/1/2023 22:45:00	
8	5/16/2023	4:37 AM	5/16/2023 4:37:00	
9	5/17/2023	5:18 AM	5/17/2023 5:18:00	
10				
11				
12				
13				

Practical Demonstration: Controlling Output with CONCATENATE and TEXT

In stark contrast to the addition method, this second demonstration focuses exclusively on

achieving perfect presentation control. This technique is indispensable if the final output must strictly adhere to a predetermined display standard, such as mandating a specific `mm/dd/yyyy` date format combined with an explicit `h:mm AM/PM` time format. For reports intended for external stakeholders or systems that rigorously parse only text strings, the combination of [CONCATENATE](#) and the **TEXT** function is the essential tool.

The structure of this formula meticulously formats each component independently before joining them. We use the [TEXT function](#) first on cell **A2** (the date), specifying the desired format string, `"m/d/yyyy"`. This converts the date serial number into a visually appealing text string. Next, the **CONCATENATE** function inserts a defined separator (in this case, " ", a simple space). Finally, we apply **TEXT** again to cell **B2** (the time), using the format string `"h:mm AM/PM"`. This step is crucial as it guarantees the inclusion of the AM or PM notation, a feature often missing by default in the addition method.

=CONCATENATE(TEXT(A2,"m/d/yyyy")," ",TEXT(B2,"h:mm AM/PM"))

Once the formula is correctly constructed in the starting cell (e.g., C2), the user simply needs to click and drag the formula down the column using the fill handle. This action immediately applies the precise, text-based formatting to the entirety of the dataset. The core benefit of this method is the absolute assurance that the output will match the specified style guide, providing the user with unparalleled control over the final visual appearance of the combined timestamp.

C2	=CONCATENATE(TEXT(A2,"m/d/yyyy")," ",TEXT(B2,"h:mm AM/PM"))				
	A	B	C	D	E
1	Date	Time	Date & Time		
2	1/3/2023	6:17 AM	1/3/2023 6:17 AM		
3	1/15/2023	3:23 PM	1/15/2023 3:23 PM		
4	2/23/2023	9:15 PM	2/23/2023 9:15 PM		
5	3/14/2023	11:59 AM	3/14/2023 11:59 AM		
6	3/25/2023	4:15 AM	3/25/2023 4:15 AM		
7	4/1/2023	10:45 PM	4/1/2023 10:45 PM		
8	5/16/2023	4:37 AM	5/16/2023 4:37 AM		
9	5/17/2023	5:18 AM	5/17/2023 5:18 AM		
10					
11					
12					
13					
14					

The resulting column C displays the date and time in a single cell for each row, adhering meticulously to the specified format strings: `m/d/yyyy` for the date and `h:mm AM/PM` for the time. This successful execution underscores the power of defining explicit [Date and Time Format](#) codes. Mastering these format codes is the key to advanced text-based manipulation of temporal data in [Google Sheets](#).

Mastering Formatting Codes for Advanced Customization

For spreadsheet professionals looking to move beyond basic concatenation, a comprehensive understanding of the full range of formatting codes available within the [TEXT function](#) is absolutely essential. These codes allow for precise control when displaying combined date and time stamps, ensuring compliance with both internal organizational standards and external international data requirements.

The formatting strings allow users to specify exactly how years, months, days, hours, and minutes are represented, including whether leading zeros should be present or if full names should be used instead of numbers. Utilizing these codes correctly is the bridge between a functional spreadsheet and a highly polished report. Some of the most commonly utilized and critical format codes include:

"yyyy": Displays the year using four digits (e.g., **2024**). Using **"yy"** displays only the last two digits.

"m": Displays the month number (1 through 12). Using **"mm"** ensures a leading zero for single-digit months (01 through 12). Using **"mmm"** displays the month abbreviation (Jan, Feb).

"h": Displays the hour using the 12-hour format (1 through 12). Using **"hh"** ensures a leading zero.

"H": Displays the hour using the 24-hour format (0 through 23). Using **"HH"** ensures a leading zero.

"AM/PM": This crucial code automatically applies the correct meridian indicator based on the time value derived from the fractional part of the [Serial Value](#).

Mastering the combination and application of these formatting codes allows users to move significantly beyond simple concatenation, enabling the creation of highly sophisticated, readable, and internationally compliant timestamps. For instance, to produce a timestamp that reads "24 May 2024 14:30," the format string would be `"dd mmm yyyy HH:mm"`.

Summary of Techniques and Further Learning

In summary, the choice between the simple addition method and the [CONCATENATE](#) and [TEXT function](#) combination hinges entirely on whether the resulting output must be numerically calculable or strictly presentation-formatted. For maximum analytical power, addition is the superior choice, maintaining the numerical integrity of the [Serial Value](#). For maximum visual control, the text functions are necessary, allowing for granular control over the final string output.

For complete and official documentation on all available formatting strings and their nuanced uses, consult the comprehensive guide for the **TEXT** function in Google Sheets online. Additionally, the following tutorials explain how to perform other common and advanced data manipulation tasks within Google Sheets: