

# How to Easily Convert Dates to Text Strings in Google Sheets

Authored by  
**stats writer**

November 30, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Convert Dates to Text Strings in Google Sheets*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102862>

Working with data in Google Sheets often requires specific data type management, especially when dealing with temporal values. The short answer is unequivocally yes: you can seamlessly convert numerical date formats into text-based strings using powerful built-in functions. This transformation is not merely cosmetic; it is often essential for complex data manipulation, concatenation, reporting, and integration with external systems that require standardized text inputs.

The primary tool for achieving this conversion is the robust **TEXT function**. Unlike standard cell formatting applied via the menu, which only changes the appearance of a cell while maintaining its underlying numerical date value, the TEXT function explicitly forces the output into a literal text string. This distinction is crucial for advanced spreadsheet tasks where data integrity depends on the data type. Furthermore, understanding the reverse process--converting a text date back into a numerical date--is equally important, which is where the complementary **DATEVALUE function** proves indispensable.

This comprehensive guide will detail the mechanics of date-to-string conversion, provide practical, step-by-step examples for both simple dates and complex datetime stamps, and thoroughly explain why this specific manipulation technique is vital for maintaining data integrity and improving the flexibility of your Google Sheets workflows. Mastering these conversion techniques will significantly enhance your ability to handle diverse datasets containing both temporal and categorical information, making data processing tasks much more streamlined and reliable.

## Understanding Dates as Serial Numbers in Google Sheets

Before diving into the conversion process, it is essential to understand how Google Sheets, like most spreadsheet applications, internally manages dates and times. Dates are not stored as text; rather, they are stored as sequential serial numbers. This numerical representation allows for easy calculation, such as determining the number of days between two dates. Specifically, the number 1 represents January 1, 1900. Every subsequent day increases this serial number by one.

When you input a date like 11/15/2023, Google Sheets automatically converts it into its corresponding serial number (e.g., 45244). The cell formatting merely applies a visual mask over this number to display it in a user-friendly way. If you change the cell format to a plain number, you will see the underlying serial value. This context is critical because functions that rely on numerical calculations, like **AVERAGE** or **SUM**, must operate on the serial number, not the formatted text.

The need for date-to-string conversion arises when you need to use the date value in a non-mathematical context. For instance, if you are attempting to concatenate a date with a piece of descriptive text (e.g., "Report generated on:") or if you need to output data into a specific flat file format (like CSV) where dates must strictly adhere to a text pattern (e.g., YYYYMMDD), relying on

the cell's default display format is insufficient. The **TEXT function** is designed precisely to extract the numerical date and format it into a permanent, predictable text string.

## Introducing the TEXT Function for Conversion

The TEXT function is the dedicated tool within Google Sheets for formatting numerical values--including dates and times--and returning them as literal text strings. It requires two main arguments: the value you wish to format (the date or number) and the format pattern you want to apply. This second argument is provided as a format code enclosed in quotation marks, offering exceptional control over the final output appearance.

The standard syntax for the function is `=TEXT(value, format_pattern)`. For dates, the `value` is usually a cell reference containing a date (e.g., A1), and the `format_pattern` uses specific codes like `"YYYY"` for the year, `"mm"` for the month, and `"dd"` for the day. The flexibility of this function means you can create virtually any date format required for integration or reporting purposes, transforming the serial number into a static text representation that will not be affected by changes in cell formatting.

Understanding the common date and time format codes is paramount to utilizing this function effectively. By using codes like `"mmm"` for abbreviated month names or `"d"` for a day number without a leading zero, you can tailor the output precisely. When the function executes, it reads the serial number in the input cell, applies your specified format template, and outputs the result as non-numerical text. This ensures that the date is preserved exactly as formatted, regardless of how other functions or formats might interpret the original date value.

## Converting Simple Dates to Strings (Method 1)

For straightforward date conversion, where the goal is simply to represent the date in a specific textual format (such as MM-DD-YYYY), the TEXT function provides a reliable and simple solution. This method is frequently used when preparing data for external databases or reports that demand a standardized text layout for dates, preventing any potential ambiguity or parsing errors caused by localized date formats.

The following formula is used to convert a standard date entry into a string representation using the Month-Day-Year convention. Note that the format code itself is encapsulated within quotation marks, signifying that it is a string argument describing the desired output pattern:

You can use the following formula to convert a date to a string in Google Sheets:

### Method 1: Convert Date to String

**`=TEXT(A1, "mm-dd-yyyy")`**

In this example, `A1` refers to the cell containing the original date value. The format pattern `"mm-dd-yyyy"` specifies that the output string should display the two-digit month, followed by a hyphen, the two-digit day, followed by another hyphen, and finally the four-digit year. This approach guarantees that even if the original cell `A1`'s display format were changed (e.g., to show full month names), the output in the target cell would consistently maintain the defined text structure.

## Step-by-Step Example 1: Date to String Conversion

To illustrate the practical application of Method 1, consider a scenario where you have a column of dates and need to generate a corresponding column of text strings for exporting to a legacy system. Maintaining the data type integrity during this process is paramount. We begin with a simple list of numerical dates in Column A, which Google Sheets automatically displays based on the default date formatting setting.

Suppose we have the following list of dates in Google Sheets:

	A	B	C	D
1	<b>Date</b>			
2	01-01-2022			
3	01-15-2022			
4	01-17-2022			
5	02-01-2022			
6	02-04-2022			
7	02-09-2022			
8	02-13-2022			
9				
10				
11				
12				
13				
14				
15				
16				
17				

We apply the `TEXT` function in the adjacent column, referencing the first date in the list. By inputting the formula `=TEXT(A1, "mm-dd-yyyy")` into cell `B1`, we instruct the spreadsheet to interpret the serial number in `A1` and output a text string formatted specifically as month-day-year. The output in `B1` will now be a text value, clearly distinct from the numerical date input.

We can use the following formula to convert the date in cell A1 to a string:

**=TEXT(A1, "mm-dd-yyyy")**

The power of spreadsheet formulas lies in their replicability. Once the formula is correctly entered in B1, we can use the fill handle--or simply copy and paste--to apply this exact logic down the entire Column B, ensuring that every corresponding date in Column A is converted into a consistent text string format. This saves significant manual effort and eliminates the risk of human error associated with re-typing or manual formatting.

We can then copy and paste this formula down to the remaining cells in column B:

	A	B	C	D
B2		<code>=TEXT(A2, "mm-dd-yyyy")</code>		
1	<b>Date</b>	<b>String</b>		
2	01-01-2022	01-01-2022		
3	01-15-2022	01-15-2022		
4	01-17-2022	01-17-2022		
5	02-01-2022	02-01-2022		
6	02-04-2022	02-04-2022		
7	02-09-2022	02-09-2022		
8	02-13-2022	02-13-2022		
9				
10				
11				
12				
13				
14				
15				
16				

All of the cells in column B are now a string, ready for any operation that requires a text data type.

## Handling Complex Datetime Values (Method 2)

Often, data sets contain not just a date but a full datetime stamp, incorporating hours, minutes, and seconds. In Google Sheets, a datetime value is still represented by a single serial number, where the integer portion represents the date and the decimal portion represents the time elapsed throughout that day. For example, 45244.5 represents noon on November 15, 2023. Converting

these complex numerical values into a specific, readable text format requires extending the format pattern used in the TEXT function.

To accurately capture and format both the date and time components, we introduce time-specific format codes such as "hh" for hours, "mm" for minutes, and "ss" for seconds. It is important to remember that the format for minutes ("mm") is the same as the format for months, meaning their context within the format string (e.g., before or after the date components) dictates their interpretation.

The formula below provides a common structure for converting a full datetime stamp into a single ISO-style text string, which is highly valuable for logging, auditing, and structured data exchange:

### Method 2: Convert Datetime to String

**=TEXT(A1,"yyyy-mm-dd hh:mm:ss")**

Using this precise format pattern ensures that the final output is a clearly defined, concatenated text string (e.g., "2023-11-15 14:30:00") that captures the temporal complexity of the original numerical value. The ability to standardize datetime outputs in this way is a cornerstone of reliable data integration and analysis in modern spreadsheet practice.

### Step-by-Step Example 2: Datetime to String Conversion

Imagine we have a transactional log where Column A records the exact moment an event occurred, displayed as a combination of datetime. We need to convert these into a standardized text format for archival purposes. The initial data will appear as full datetime stamps, leveraging the decimal portions of the serial number.

Suppose we have the following list of datetimes in Google Sheets:

	A	B	C	D
1	<b>Datetime</b>			
2	2022-01-01 8:30:15			
3	2022-01-15 3:30:00			
4	2022-01-17 10:40:10			
5	2022-02-01 12:15:00			
6	2022-02-04 1:10:12			
7	2022-02-09 4:15:25			
8	2022-02-13 7:45:50			
9				
10				
11				
12				
13				
14				
15				
16				
17				

By applying Method 2's formula into cell B1, `=TEXT(A1, "yyyy-mm-dd hh:mm:ss")`, we extract the full date and time components from the numerical value in A1 and output them as a single, unified text string. This preserves the granularity of the time data while ensuring it is treated purely as text by any subsequent function or application.

We can use the following [formula](#) to convert the datetime in cell A1 to a string:

**`=TEXT(A1,"yyyy-mm-dd hh:mm:ss")`**

As before, the resulting formula can be efficiently copied down to the rest of Column B. This action rapidly processes the entire dataset, converting all numerical datetime values into consistent text strings. The resulting text strings are visually identical to the original display, but their underlying data type has been fundamentally changed from number/date to text/string, which is the desired outcome for many data manipulation tasks.

We can then copy and paste this [formula](#) down to the remaining cells in column B:

B2 fx =TEXT(A2, "yyyy-mm-dd hh:mm:ss")

	A	B	C	D
1	<b>Datetime</b>	<b>String</b>		
2	2022-01-01 8:30:15	2022-01-01 08:30:15		
3	2022-01-15 3:30:00	2022-01-15 03:30:00		
4	2022-01-17 10:40:10	2022-01-17 10:40:10		
5	2022-02-01 12:15:00	2022-02-01 12:15:00		
6	2022-02-04 1:10:12	2022-02-04 01:10:12		
7	2022-02-09 4:15:25	2022-02-09 04:15:25		
8	2022-02-13 7:45:50	2022-02-13 07:45:50		
9				
10				
11				
12				
13				
14				
15				
16				
17				

All of the cells in column B are now confirmed to be a string, suitable for non-calculation-based operations.

## Verification and Data Integrity Checks

After performing a data type conversion, especially one as fundamental as numerical to text, it is crucial to verify the results to ensure data integrity. Google Sheets provides a dedicated function, **ISTEXT()**, which serves as a Boolean check, returning **TRUE** if the referenced cell contains a text string, and **FALSE** otherwise.

We can use **ISTEXT()** to compare the original date cell (A2) with the newly converted string cell (B2) in our dataset. This verification step confirms that the original date value is indeed numerical (returning **FALSE**) and the converted value is text (returning **TRUE**). This clear distinction proves the efficacy of the **TEXT function** in changing the underlying data type.

To verify this, we can type **=ISTEXT(A2)** and **=ISTEXT(B2)** to check if the values in cells A2 and B2 are strings, respectively:

	A	B	C	D
1	<b>Date</b>	<b>String</b>	=ISTEXT(A2)	=ISTEXT(B2)
2	01-01-2022	01-01-2022	FALSE	TRUE
3	01-15-2022	01-15-2022		
4	01-17-2022	01-17-2022		
5	02-01-2022	02-01-2022		
6	02-04-2022	02-04-2022		
7	02-09-2022	02-09-2022		
8	02-13-2022	02-13-2022		
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

In the verification column for Example 1, we observe that the test on cell A2 returns **FALSE**, confirming that the value in A2 remains a numerical date serial number. Conversely, the test on cell B2 returns **TRUE**, confirming the successful conversion to a permanent text string.

For the datetime example (Example 2), the verification process yields the same results, confirming the change in data type across both the input (A2) and the output (B2) cells, demonstrating successful conversion even for time components:

	A	B	C	D
1	<b>Datetime</b>	<b>String</b>	=ISTEXT(A2)	=ISTEXT(B2)
2	2022-01-01 8:30:15	2022-01-01 08:30:15	FALSE	TRUE
3	2022-01-15 3:30:00	2022-01-15 03:30:00		
4	2022-01-17 10:40:10	2022-01-17 10:40:10		
5	2022-02-01 12:15:00	2022-02-01 12:15:00		
6	2022-02-04 1:10:12	2022-02-04 01:10:12		
7	2022-02-09 4:15:25	2022-02-09 04:15:25		
8	2022-02-13 7:45:50	2022-02-13 07:45:50		
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

We can see that the value in cell A2 is not a string since the **ISTEXT()** function returned **FALSE**. Conversely, we can see that the value in cell B2 is clearly a string since the **ISTEXT()** function returned **TRUE**.

## The Reverse Process: Using DATEVALUE and TIMEVALUE

While the TEXT function converts dates to strings, data manipulation often requires the opposite: converting a text string that looks like a date back into a numerical date serial number so that it can be used in mathematical calculations. This reverse process is handled efficiently by the **DATEVALUE** function in Google Sheets.

The **DATEVALUE** function takes a date string (e.g., "11/15/2023") and converts it into the corresponding date serial number (e.g., 45244). This is particularly useful when importing data from external sources, like CSV files, where dates may have been imported as text and therefore cannot be used in functions like **DATEDIF** or be correctly sorted chronologically. The syntax is simply =DATEVALUE("date\_string") or =DATEVALUE(A1) if A1 contains the date string.

If the imported text string includes both date and time (a datetime string), you would typically use

**DATEVALUE** to extract the date serial number and **TIMEVALUE** to extract the decimal representation of the time. You can then combine these two outputs using addition (`=DATEVALUE(A1) + TIMEVALUE(A1)`) to reconstruct the full numerical datetime serial number, restoring the ability to perform accurate temporal calculations on the data.

ARABPSYCHOLOGY.COM