

How to Easily Convert Month Names and Numbers in Google Sheets

Authored by
stats writer

December 3, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Convert Month Names and Numbers in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=104176>

The ability to seamlessly convert between month names (e.g., "January") and their corresponding numerical values (e.g., 1) is a fundamental requirement for advanced data manipulation and reporting within Google Sheets. While this conversion might seem simple, it necessitates an understanding of how Google Sheets handles date and time data internally.

This comprehensive guide details the precise formulas and methodologies required to perform these conversions efficiently. We will utilize two primary functions--the powerful TEXT function for converting numbers to names, and the robust MONTH function for extracting numerical values from text-based month entries.

Understanding Date Handling in Google Sheets

Before diving into the conversion formulas, it is crucial to recognize that Google Sheets, like many spreadsheet applications, stores dates not as displayable text but as sequential serial numbers. This internal structure is essential for performing calculations accurately.

The serial number system begins counting from Day 1, which corresponds to December 30, 1899. For example, January 1, 1900, is represented internally as the number 2. When you input a date, Sheets automatically converts it to this serial value, and then applies date formatting to display it in a user-friendly format (like 01/01/1900).

Understanding the difference between the underlying numerical value and the displayed text format is the key to successfully using the `TEXT()` and `MONTH()` functions for conversions.

Method 1: Converting Month Number to Month Name using TEXT()

To convert a month number (1 through 12) into a descriptive month name (January through December), we utilize the **TEXT()** function. The purpose of **TEXT()** is to apply specific formatting rules to a numerical value, thereby changing its appearance without altering the underlying data type significantly.

The challenge when using **TEXT()** with a month number is that **TEXT()** expects a valid date serial number as its first argument, not just a standalone number between 1 and 12. Therefore, we must manipulate the month number input to simulate a recognizable date value.

A common technique, shown in the examples below, is multiplying the month number by 29 (or any number greater than 28 that is less than 31). While this may seem arbitrary, it effectively creates a large serial number that represents a specific date far enough into the calendar system for the **TEXT()** function to correctly interpret the month component based on the format code provided.

The following formulas illustrate how to perform this conversion, providing options for both full and abbreviated month names:

Method 1: Convert Month Number to Month Name

#convert month number in cell A1 to full month name

=TEXT(A1*29, "mmmm")

#convert month number in cell A1 to abbreviated month name

=TEXT(A1*29, "mmm")

Deep Dive into TEXT() Formatting Codes

The second argument of the **TEXT()** function, which specifies the format, is critical for controlling the output quality. When dealing with dates, specific codes instruct Sheets on how to present the date components. For month names, we primarily use two formats:

"mmmm" (Four 'm's): This format instructs Google Sheets to display the **full month name**. For instance, if the input number corresponds to the third month, the output will be **"March"**. This is highly useful for formal documents and reports where full clarity is required.

"mmm" (Three 'm's): This format provides the **abbreviated month name**. Using the same example, the output for the third month would be **"Mar"**. Abbreviated names are excellent for compact dashboards, visualizations, and scenarios where space is limited.

Note that these codes are locale-dependent. If your spreadsheet is set to a different language, the output names (e.g., 'Janvier' instead of 'January') will automatically adjust, ensuring consistency across international data sets.

Practical Demonstration: Number to Name Conversion (Example 1)

Let's look at how the **TEXT()** function transforms raw numerical data into recognizable, formatted month names. This process enhances the readability of data sets significantly, moving from cryptic numbers to intuitive labels.

The following screenshot shows how to use the **TEXT()** function to convert month numbers (from 1 to 12) to their full month names (from January to December):

	A	B	C	D	E
B1		<code>=TEXT(A1*29, "mmm")</code>			
1		1 January			
2		2 February			
3		3 March			
4		4 April			
5		5 May			
6		6 June			
7		7 July			
8		8 August			
9		9 September			
10		10 October			
11		11 November			
12		12 December			
13					
14					
15					
16					
17					
18					
19					
20					
21					

As illustrated, by applying the formula `=TEXT(A1*29, "mmm")` to cell A1 (which contains the number 1), the result in the output column is **"January"**. This demonstrates the power of utilizing a date format against an artificially created serial number.

We can also use the **"mmm"** format to convert the month number to the abbreviated month name, which is useful for data visualization efficiency:

	A	B	C	D	E
1		1 January	Jan		
2		2 February	Feb		
3		3 March	Mar		
4		4 April	Apr		
5		5 May	May		
6		6 June	Jun		
7		7 July	Jul		
8		8 August	Aug		
9		9 September	Sep		
10		10 October	Oct		
11		11 November	Nov		
12		12 December	Dec		
13					
14					
15					
16					
17					
18					
19					

Switching the format code to **"mmm"** instantly adjusts the output to the three-letter standard (e.g., "Jan", "Feb", "Mar"). This flexible approach ensures that the conversion method can be tailored precisely to the data presentation requirements of any project.

Method 2: Converting Month Name to Month Number using MONTH()

When the input data is a text string representing a month name (e.g., "February"), and the desired output is the corresponding numerical month (2), we employ the **MONTH()** function. The **MONTH()** function is designed specifically to extract the month component from a valid date value.

However, like the **TEXT()** function, **MONTH()** requires its input to be recognized as a date. A simple text string like "January" is not inherently recognized as a date object by Sheets unless it is explicitly concatenated with a day and year component.

To overcome this, we use a clever concatenation trick: appending a day value (typically '1') to the month name string. For example, if cell A1 contains "March," the expression `A1&1` evaluates internally to "March1." Google Sheets is intelligent enough to parse this concatenated string as a date (specifically, the first day of the specified month in the current year, or a default year if the

current year is ambiguous), allowing the **MONTH()** function to successfully extract the numerical index.

Method 2: Convert Month Name to Month Number

#convert month name in cell A1 to month number

=MONTH(A1&1)

The following examples show how to use these formulas in practice, emphasizing the necessity of creating a valid date string via concatenation.

Handling Month Name Input Variations and Concatenation

The concatenation method (`MONTH(A1&1)`) is remarkably robust and handles both full and abbreviated month names seamlessly, provided the input string is correctly spelled and recognized by the spreadsheet's locale settings.

Considerations for Input Data:

Locale Matching: The input month name must match the language settings of your Google Sheet. If your sheet is set to English, "Janvier" will likely result in an error or an incorrect conversion, whereas "January" will succeed.

Case Insensitivity: Fortunately, Google Sheets' date parsing is generally case-insensitive, meaning "january," "January," and "JANUARY" will all be correctly interpreted as the first month.

Explicit Date Creation (Alternative): For scenarios requiring explicit year handling, the **DATE()** function provides a more structured alternative. While the simple concatenation method is faster for month-only conversion, if you need to perform calculations based on a specific year, you would typically convert the month name to a number first, and then build the date using `=DATE(Year, MONTH(A1&1), Day)`.

Practical Demonstration: Name to Number Conversion (Example 2)

The final examples showcase the efficiency of the **MONTH()** function combined with the concatenation operator (&) to perform the reverse conversion--from textual month names back to numerical indices.

The following screenshot shows how to use the **MONTH()** function to convert full month names (from January to December) to month numbers (from 1 to 12):

	A	B	C	D	E
1	January	1			
2	February	2			
3	March	3			
4	April	4			
5	May	5			
6	June	6			
7	July	7			
8	August	8			
9	September	9			
10	October	10			
11	November	11			
12	December	12			
13					
14					
15					
16					
17					
18					
19					
20					

Here, the formula `=MONTH(A1&1)` takes the string "January" in A1, turns it into a recognizable date (January 1st), and extracts the number 1. This method provides the required numerical index needed for sorting, indexing, or feeding into other complex spreadsheet formulas.

It is important to note that this functionality is robust enough to handle abbreviated month names as well, demonstrating its versatility in various data structures:

	A	B	C	D	E
B1		<code>=MONTH(A1&1)</code>			
1	Jan	1			
2	Feb	2			
3	Mar	3			
4	Apr	4			
5	May	5			
6	Jun	6			
7	Jul	7			
8	Aug	8			
9	Sep	9			
10	Oct	10			
11	Nov	11			
12	Dec	12			
13					
14					
15					
16					
17					
18					
19					
20					

Whether the input is "Jan" or "January," the result remains the same numerical value (1). This resilience against formatting variations makes the `MONTH(A1&1)` technique a highly reliable tool for cleaning and standardizing date data imported from external sources.

Advanced Scenarios and Error Prevention

While the `TEXT()` and `MONTH()` methods are highly effective, advanced users should be aware of potential pitfalls and alternative solutions for more complex data handling.

Handling Invalid Input: If the cell containing the month number is empty (blank) or contains text that cannot be parsed as a date (e.g., "Tues"), the `TEXT()` or `MONTH()` functions may return a `#VALUE!` error. It is good practice to wrap these conversions in an `IFERROR()` or `ISBLANK()` function to handle exceptions gracefully, resulting in a cleaner output (e.g., displaying a blank cell or a default value instead of an error message).

Using DATEVALUE(): If you encounter issues where the simple concatenation (`A1&1`) fails to produce a date object due to regional settings or non-standard input, the `DATEVALUE()` function can be used to force Sheets to interpret a text string as a serial date number before passing it to

MONTH(). For instance, `=MONTH(DATEVALUE(A1 & " 1, 2000"))` explicitly constructs a full date string that is less prone to ambiguity.

Ensuring Year Context: Remember that when converting a month name to a number using the simple `MONTH(A1&1)` method, Sheets assigns the current year by default. If your conversion is part of a larger process where the year matters (e.g., calculating the number of days between two dates), always ensure the full date is constructed using the **DATE()** function after the month number has been successfully extracted.

Mastering these simple yet critical conversion techniques allows users to move beyond static data presentation and unlock the full analytical potential of [Google Sheets](#) for time-based data.

ARABPSYCHOLOGY.COM