

Convert Between Month Name & Number in Google Sheets

Authored by
Mohammed looti

November 2, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Convert Between Month Name & Number in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=8298>

Introduction to Essential Date Conversion Techniques in Google Sheets

Effective data management and high-quality reporting fundamentally rely on the ability to seamlessly manipulate and convert date formats. Within the environment of [Google Sheets](#), analysts frequently encounter datasets where chronological information, specifically months, is represented inconsistently. Months may appear as numerical identifiers (1 through 12) or as full or abbreviated textual strings (e.g., "January" or "Jan"). Achieving interoperability between these two formats is not merely convenient; it is essential for critical tasks such as generating user-friendly reports, ensuring accurate chronological sorting, and executing specialized temporal calculations that require numeric inputs.

This expert guide provides a precise breakdown of the formulas required to execute these conversions in both directions: transforming numeric month identifiers into their corresponding names and parsing textual month names back into numerical indices. We will delve into the powerful application of the **TEXT()** function, which governs numeric formatting, and the inventive combination of the **MONTH()** function with [concatenation](#) to accurately interpret text strings as date components. Mastery of these techniques will significantly streamline and enhance your data preparation and analysis workflows within [Google Sheets](#).

The subsequent sections detail the core methodologies, providing step-by-step explanations of the underlying logic and practical examples that demonstrate their implementation in a live spreadsheet environment. Understanding the nuances of these functions, especially how they interact with Google Sheets' serial date system, is the key to successful date conversion.

Core Methodology 1: Converting Month Number to Textual Name

When the objective is to translate a simple month number (such as 5, representing May) into its full or abbreviated textual equivalent, the primary tool is the robust [TEXT function](#). This function is designed to apply specific display formatting rules to numeric values. However, a specific and crucial adjustment is required in [Google Sheets](#): the **TEXT()** function fundamentally expects a date or time value, which is internally stored as a serial number (the count of days since January 1, 1900). Therefore, a raw month number cannot be used directly; it must first be converted into a recognizable date serial number.

The standard, reliable approach involves multiplying the input month number by a factor that ensures the resulting serial number falls within the target month. By multiplying the month number (assumed to be in cell A1) by 29, we generate a serial date number that Google Sheets can interpret as belonging to the desired month. For instance, if A1 holds 1 (January), $A1*29$ results in a serial number corresponding to January 29th. If A1 holds 12 (December), the resulting large serial number is correctly parsed by the **TEXT()** function, which only extracts the appropriate month component based on the specific format defined.

The second argument of the [TEXT function](#) is a critical formatting mask. Using the mask "mmmm" mandates the return of the full month name (e.g., "January"), while "mmm" yields the abbreviated month name (e.g., "Jan"). This formatting flexibility allows users to tailor the output precisely to meet various reporting and dashboard requirements, ensuring data is displayed clearly and concisely.

#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 the TEXT() Function Syntax and Date Formatting

The [TEXT function](#) is a cornerstone of advanced data display in Google Sheets, enabling users to move beyond standard numeric presentation to handle complex display requirements. Its standardized syntax is `TEXT(value, format_pattern)`. When performing date conversions, achieving success hinges entirely upon a thorough understanding of the `format_pattern` argument, particularly the specific sequences used for month representation.

For month names, the format patterns rely exclusively on specific repetitions of the lowercase letter 'm'. The number of times 'm' is repeated dictates the level of detail and abbreviation in the resulting output:

"m" or "mm": These patterns return the month as a number (1 or 01, respectively). While functional, they are not used for text conversion.

"mmm": This pattern returns the universally accepted three-letter abbreviated month name (e.g., "Feb"). This concise format is invaluable for space-constrained dashboards or summary reports where screen real estate is limited.

"mmmm": This pattern yields the full, unabbreviated month name (e.g., "February"). This is the standard choice for formal documentation, clear communication, and non-abbreviated data presentation.

The necessity of the `*29` adjustment within the formula cannot be overstated. Since the [TEXT function](#) is engineered to process a date serial number--where the first day is January 1, 1900--simply inputting the number '1' would correctly return January, but inputting '12' (for December) would only return January 12th, not December. By multiplying the month number by 29, we guarantee that the calculated serial number is sufficiently advanced into the year that the date parsing mechanism correctly identifies the associated month component. This creative application

of [date formatting codes](#) and serial number manipulation is the defining characteristic of this reliable conversion method.

Core Methodology 2: Converting Month Name Back to Number

The inverse operation--translating a month name string (e.g., "August") back into its corresponding numeric index (8)--is equally vital for data processing, enabling accurate filtering, chronological sorting, and calculation steps that require integer inputs. This conversion process is efficiently handled by leveraging the specialized [MONTH function](#).

The **MONTH()** function is inherently simple: it accepts a valid date input and extracts the month number (1 to 12). However, if the input is only a solitary text string, such as "March," the function will fail because it lacks the context to parse a full, recognizable date. To circumvent this constraint, we must programmatically construct a valid date string that the parsing engine can successfully recognize.

We achieve this by employing the [concatenation](#) operator (&) to append a dummy day value, typically '1', to the month name housed in cell A1. This action transforms the ambiguous text ("March") into a fully recognizable date string (e.g., "March 1"). The [MONTH function](#) can then successfully process this composite string, extract the month component, and return the correct integer index. This technique proves reliable regardless of whether the input is a full month name or a standard three-letter abbreviation.

#convert month name in cell A1 to month number
=MONTH(A1&1)

Understanding the MONTH() Function and Date Parsing Robustness

The reliance on [concatenation](#) and the inclusion of a dummy day value represent an essential trick for handling incomplete date strings. [Google Sheets](#) utilizes advanced date parsing logic capable of interpreting ambiguous text inputs, provided they include the minimum necessary components--typically a month and a day--to resolve into a valid date. By appending &1, we satisfy this minimal structural requirement, ensuring the function executes successfully.

It is crucial to emphasize that the [MONTH function](#) consistently returns a standardized integer value between 1 (for January) and 12 (for December). This standardized numeric output is critically important, making the results immediately suitable for use in advanced calculations, filtering operations, pivot table groupings, or sorting algorithms where text strings would otherwise lead to chronological errors or require complex lookup tables.

Furthermore, this conversion methodology exhibits significant robustness against minor variations

in the input text. Whether the source cell contains "January," the abbreviated "JAN," or even the lowercase "january," the sophisticated parsing engine of [Google Sheets](#) typically recognizes the month name correctly. This inherent reliability ensures consistent results across varied datasets, although maintaining consistency in spelling and language settings remains the best practice for optimal performance.

Practical Demonstration: Full Examples and Visual Aids

To solidify the mastery of these conversion formulas, the following practical examples provide clear visual confirmation of their application within a typical spreadsheet context. These illustrations demonstrate the necessary input format, the precise formula application, and the resulting converted output, confirming the efficacy of both the **TEXT()** and **MONTH()** methods in real-world scenarios.

The following screenshot illustrates the process of utilizing the **TEXT()** function to convert numeric month identifiers (ranging from 1 to 12) into their corresponding full month names (January through December). Observe the mandatory inclusion of the *29 factor, which is vital for correct date serial interpretation.

| | A | B | C | D | E |
|----|---|-------------|---|---|---|
| 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 | | | | | |

For scenarios requiring a more compact presentation, the format pattern within the [TEXT function](#) can be easily modified to produce abbreviated month names. By simply substituting "mmmm" with the "mmm" format code, the output remains highly readable yet concise, making it ideal for compact reports and summaries.

| | 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 | | | | | |

Conversely, the next visual aid demonstrates the critical reverse conversion: translating month names back into their corresponding numerical indices using the **MONTH()** function paired with [concatenation](#) (&1). This formula successfully parses the text "January" and returns the integer 1, "February" returns 2, and so forth, providing the necessary numeric data for calculations.

| | A | B | C | D | E |
|----|-----------|------------------------|---|---|---|
| B1 | | <i>fx</i> =MONTH(A1&1) | | | |
| 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 | | | | | |

A key advantage of this method is its ability to handle abbreviated month names efficiently. As illustrated in the final example, the formula successfully parses the abbreviated input "Aug" and accurately returns the number 8, ensuring consistency and high reliability regardless of the length of the textual input string.

| | 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 | | | | | |

Advanced Date Functionality and Integrating Conversions

While the **TEXT()** and **MONTH()** functions offer precise, direct solutions for month name and number conversions, a comprehensive understanding of related date and time functions in [Google Sheets](#) can unlock significantly greater analytical power. These complementary functions are particularly valuable when the analysis moves into complex date arithmetic, duration calculations, or the generation of chronological reports spanning multiple years.

For users routinely managing extensive chronological data, exploring functions such as **DATE()**, **EOMONTH()**, and **DATEDIF()** is highly recommended. Specifically, the **DATE()** function allows the construction of a valid, full date from separate year, month, and day components. This is often the necessary next step after using the **MONTH()** function to derive a standardized numerical month index from a raw text input. This capability is vital for standardizing chronological inputs originating from varied external spreadsheet sources.

Furthermore, for scenarios involving international date formats or complex time zone adjustments, maintaining awareness of the differences between Google Sheets' internal date handling (which

begins on January 1, 1900) and other systems is paramount. Implementing these specialized conversion techniques ensures that your monthly data remains structurally sound, accurate, and consistently presentable, irrespective of the initial format of the month information received.