

How to Easily Convert Text to Numbers in Google Sheets: A Step-by-Step Guide

Authored by
stats writer

November 30, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Convert Text to Numbers in Google Sheets: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102583>

Data integrity relies heavily on accurate data types. In environments like [Google Sheets](#), numerical values must be stored as numbers, not as text strings, to ensure mathematical calculations function correctly. Often, data imported from external sources (such as web forms, databases, or legacy systems) can mistakenly classify figures--including prices, scores, or measurements--as text. This article serves as an expert guide, detailing three powerful and distinct methods available within [Google Sheets](#) to consistently and accurately convert text representations into usable numerical values.

We will examine the operational differences between three core functions: the fundamental [VALUE function](#), which standardizes text that already appears numeric; the specialized [TO PURE NUMBER function](#), designed to strip away non-numeric formatting like currency symbols or percentage signs; and finally, a more advanced formula combination utilizing the [SPLIT](#) and [CONCATENATE](#) functions to surgically extract numbers embedded within complex, mixed-character text strings. Understanding these distinctions is critical for high-efficiency data processing and analysis.

Overview of Key Conversion Methods

Effective data manipulation in spreadsheet software requires selecting the appropriate tool for the specific data format you are handling. The three primary methods outlined below offer solutions for the most common text-to-number conversion challenges encountered in [Google Sheets](#). Each method targets a specific type of textual data representation, ranging from straightforward numerical text to highly complex alphanumeric strings.

The first method focuses on simple conversion, ideal when the text data simply needs to be re-interpreted by [Google Sheets](#) as a number, often resolving minor formatting issues or trailing spaces. The second method is tailored for financial or quantitative data where symbols must be disregarded to isolate the pure numerical magnitude. The third and most complex method is reserved for data extraction tasks where the desired numerical values are interspersed with arbitrary characters and punctuation, requiring sophisticated formula logic to isolate the digits.

You can use the following fundamental formulas to convert text to numbers in [Google Sheets](#), depending on your source data complexity:

Method 1: Convert Text to Number (Simple Numerical Text)

=VALUE(A1)

Method 2: Convert Currency or Formatted Text to Pure Number

=TO_PURE_NUMBER(A1)

Method 3: Extract Numerical Digits from Complex Alphanumeric Text

=SPLIT(A1,CONCATENATE(SPLIT(A1,".0123456789")))

The subsequent sections provide detailed explanations and practical examples illustrating how to implement and verify each formula in a live spreadsheet environment.

Method 1: Converting Simple Text-Formatted Numbers using VALUE

The VALUE function is the most direct approach for coercing a text string into a true numerical format, provided the string itself consists solely of characters that represent a number (e.g., "123.45" or "-50"). This function attempts to mathematically evaluate the contents of the cell. If the input is a valid numerical representation, even if stored as text by the spreadsheet, VALUE function returns the numeric equivalent. This is essential for cleaning up columns where numbers may have been entered with a leading apostrophe or through data import processes that default to text formatting.

Syntax for the VALUE function is straightforward: `=VALUE(cell_reference)`. It takes a single argument, which is the cell or value containing the text you wish to convert. A common scenario where this function proves invaluable is when numbers are center-aligned or left-aligned within a cell (a typical indicator of text formatting) but are intended for calculations. Applying VALUE function instantly resolves this issue, making the data computationally accessible.

It is important to note the limitations of the VALUE function. It cannot handle extraneous characters such as currency signs, commas used as thousands separators (depending on locale settings), or descriptive text mixed with numbers. If the input string contains non-numeric elements that are not part of a standard numerical format, the function will return an error (usually #VALUE!), necessitating the use of the more advanced techniques described in the following sections.

Practical Demonstration: Using the VALUE Function (Example 1)

Consider a column (Column A) where numerical data, such as product quantities or scores, has been incorrectly formatted as text. To initiate the conversion process, we use the VALUE function in an adjacent column (Column B). Specifically, we type the following formula into cell **B2** to convert the textual content residing in cell **A2** into a recognized numerical format:

=VALUE(A2)

Once this formula is entered in **B2**, we then copy and paste or drag the formula down to apply it to all corresponding cells in column B. This action efficiently converts the entire range of text-formatted numbers in Column A into functional numerical values in Column B. The visible effect is often a change in alignment (numbers typically align to the right by default) and the ability to sum or average the new column data without error.

We can copy and paste this formula down to the remaining cells in column B, demonstrating the successful batch conversion:

	A	B	C	D
1	Text	Number		
2	4	4		
3	13	13		
4	19	19		
5	22	22		
6	25	25		
7	30	30		
8	35	35		
9	39	39		
10	40	40		
11	41	41		
12	44	44		
13				
14				
15				
16				
17				
18				

All of the cells in column B are now registered internally as true numbers. To confirm that the conversion was successful and that the resulting data type is indeed numeric, a validation step is crucial. This verification utilizes the powerful ISNUMBER function, a fundamental diagnostic tool in spreadsheet analysis.

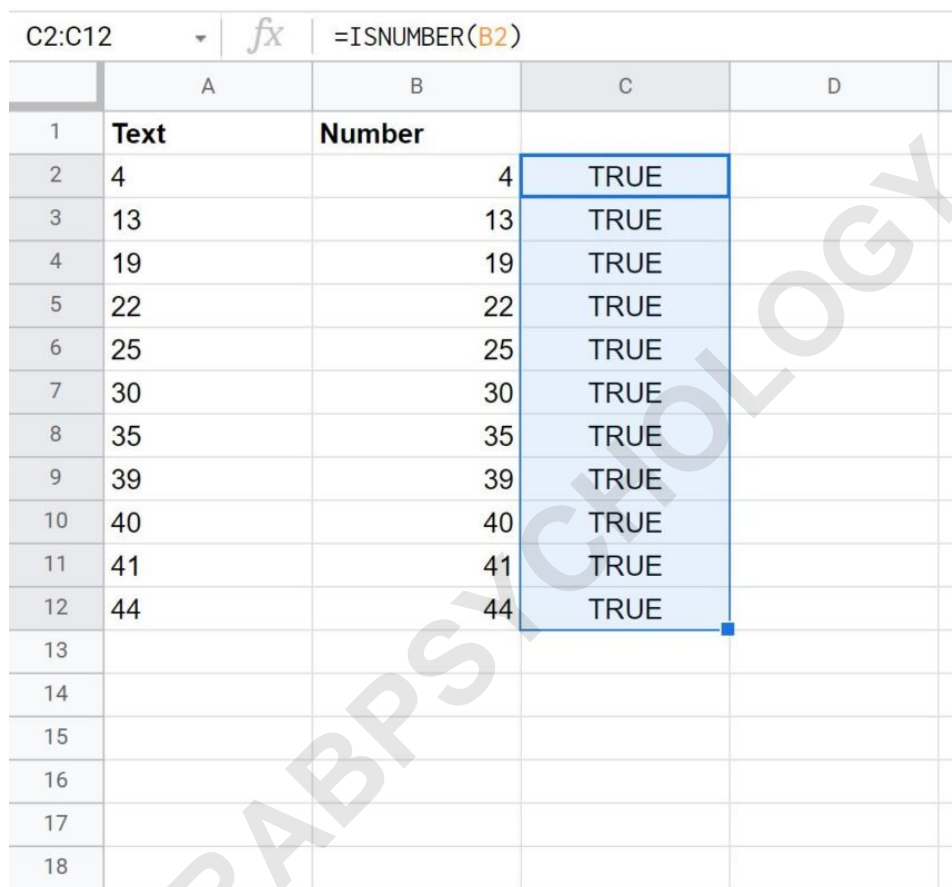
Verifying Data Types: Using the ISNUMBER Function

After performing any data type conversion, it is best practice to verify the output using the ISNUMBER function. This function returns a simple Boolean value--TRUE if the cell contains a number, and FALSE otherwise. To verify the success of the conversion demonstrated in Example 1, we can input the following verification formula into cell **C2**:

=ISNUMBER(B2)

Upon execution, the expected result in cell **C2** should be TRUE, confirming that the VALUE function successfully transformed the text in A2 into a valid numerical data type in B2. This diagnostic step ensures that subsequent calculations referencing column B will execute without type mismatch errors.

Observing the output, we confirm the validity of our conversion:



	A	B	C	D
1	Text	Number		
2	4	4	TRUE	
3	13	13	TRUE	
4	19	19	TRUE	
5	22	22	TRUE	
6	25	25	TRUE	
7	30	30	TRUE	
8	35	35	TRUE	
9	39	39	TRUE	
10	40	40	TRUE	
11	41	41	TRUE	
12	44	44	TRUE	
13				
14				
15				
16				
17				
18				

We can clearly see that the value in cell **B2** is recognized as a number since the **ISNUMBER()** function returned **TRUE**. This methodology provides a transparent and verifiable means of data validation throughout your Google Sheets workflow.

Method 2: Handling Currency and Special Formatting with TO_PURE_NUMBER

When dealing with financial data, imported figures often include currency symbols (like '\$' or '€'), percentage signs ('%'), or other non-numeric characters that the standard VALUE function cannot

parse. The TO PURE NUMBER function is specifically designed to address this challenge by stripping away all non-numeric elements from a cell and returning only the raw numerical component. This is critical for performing aggregated calculations on financial columns without manually cleaning each cell.

The TO PURE NUMBER function, like VALUE function, takes a single argument: the cell reference containing the formatted data. For instance, if cell A1 contains "\$1,500.00", `=TO_PURE_NUMBER(A1)` will return the number 1500. Similarly, if A1 contains "15%", it will return 0.15 (as 15 percent mathematically equates to 0.15). It efficiently handles standard currency and percentage notations across various locales supported by Google Sheets.

The primary advantage of using TO PURE NUMBER function over manual text manipulation techniques (like repeated use of the SUBSTITUTE function) is its robustness and simplicity. It automatically identifies and removes the symbols associated with common formatting, ensuring that the resulting number is immediately ready for aggregation, charting, or complex modeling tasks, saving significant time during the data cleaning phase.

Practical Demonstration: Converting Currency Data (Example 2)

Assume Column A contains sales figures formatted as currency strings, such as "\$450" or "€12.99". To convert these currency representations into clean, calculable numerical values, we employ the TO PURE NUMBER function. We input the following formula into cell **B2** to convert the currency value in cell **A2**:

```
=TO_PURE_NUMBER(A2)
```

By copying this formula down the column, we transform the entire dataset. The function intelligently recognizes the currency symbol and other non-digit characters and discards them, preserving the underlying magnitude of the monetary value. This step is critical before any budget analysis or calculation of totals.

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

	A	B	C	D
B2		<code>=TO_PURE_NUMBER(A2)</code>		
1	Currency	Number		
2	\$4.00	4		
3	\$34.43	34.43		
4	\$50.00	50		
5	\$100.35	100.35		
6	\$150.00	150		
7	\$190.00	190		
8	\$200.00	200		
9	\$20.99	20.99		
10	\$20.77	20.77		
11	\$35.00	35		
12				
13				
14				
15				

Once applied, all of the cells in column B hold the pure numerical quantity, ready for computation. Just as in Example 1, we must confirm the success of the conversion using the ISNUMBER function. Entering `=ISNUMBER(B2)` in cell **C2** provides the necessary confirmation.

After performing the validation step:

	A	B	C	D
C2			=ISNUMBER(B2)	
1	Currency	Number		
2	\$4.00	4	TRUE	
3	\$34.43	34.43		
4	\$50.00	50		
5	\$100.35	100.35		
6	\$150.00	150		
7	\$190.00	190		
8	\$200.00	200		
9	\$20.99	20.99		
10	\$20.77	20.77		
11	\$35.00	35		
12				
13				
14				
15				
16				

We confirm that the value in cell **B2** is unequivocally a number, as the **ISNUMBER()** function successfully returned **TRUE**. This confirms that the TO PURE NUMBER function is the optimal choice for standardized currency conversion.

Method 3: Extracting Pure Numerical Data from Complex Text Strings

In highly unstructured data, numerical information is often embedded within descriptive text strings (e.g., "The result was 42 units and 1 failure."). Neither VALUE function nor TO PURE NUMBER function can handle this scenario, as they fail when encountering arbitrary text. To solve this, we must employ a sophisticated formula combination involving **SPLIT** and **CONCATENATE** to isolate the digits from the surrounding characters.

The complex formula works by first creating a list of all non-numeric characters present in the cell using a combination of **SPLIT** and **CONCATENATE**. It then uses this list of non-numeric characters as the delimiter for a final **SPLIT** operation on the original string. This effectively breaks the string into pieces, separating the numbers from the text. Although complex, this is a robust method for mimicking regular expression extraction techniques within standard Google Sheets functions.

The full formula used is highly efficient for this purpose, although it looks intimidating. It ensures

that every character that is not a digit (0-9) or a decimal point (.) is used as a delimiter, leaving only the required numerical segments. This is an essential technique for analysts working with log files, sensor outputs, or poorly structured qualitative data where numeric metrics must be salvaged.

Practical Demonstration: Isolating Numbers (Example 3)

Suppose cell **A2** contains a mixed string like "Batch_001_17.5_kg_total". Our objective is to extract the numerical value 17.5. We achieve this by typing the following intricate, yet powerful, formula into cell **B2** to extract the numbers from the text in cell **A2**:

=SPLIT(A2,CONCATENATE(SPLIT(A2,".0123456789")))

This formula intelligently identifies and removes all non-numeric separators, leaving the numerical data isolated. The result in **B2** will be the pure number (17.5 in our example). Note that if the original text contains multiple separate numbers (e.g., "17.5 and 5.2"), the **SPLIT** function will output these into multiple adjacent cells.

We proceed by copying and pasting this formula down to the remaining cells in column B to process the entire dataset:

	A	B	C	D	E
1	Currency	Number			
2	400 dollars	400			
3	\$355.44	355.44			
4	50##	50			
5	\$650.00	650			
6	\$900.00	900			
7	\$\$800	800			
8	430dollars	430			
9	600 Dollars	600			
10	\$20.77	20.77			
11	\$50	50			
12					
13					
14					
15					
16					

Following successful extraction, all of the cells in column B now contain the purified numerical data. This method provides the highest degree of flexibility for cleaning difficult, unstructured text strings. As a final step, verification using ISNUMBER function is performed to ensure data integrity.

To verify this, we type **=ISNUMBER(B2)** in cell **C2**, confirming the extracted value is a proper number:

	A	B	C	D
1	Currency	Number		
2	400 dollars	400	TRUE	
3	\$355.44	355.44		
4	50##	50		
5	\$650.00	650		
6	\$900.00	900		
7	\$\$800	800		
8	430dollars	430		
9	600 Dollars	600		
10	\$20.77	20.77		
11	\$50	50		
12				
13				
14				
15				
16				

We can see that the value in cell **B2** is a number since the **ISNUMBER()** function returned **TRUE**. This extraction technique is invaluable for deep data cleaning.

Summary and Best Practices for Data Conversion

Selecting the correct conversion function is paramount to efficient data preparation in Google Sheets. For data that is visually numeric but stored incorrectly, the VALUE function provides a fast, clean solution. When dealing with standardized formatting like currency or percentages, the TO PURE NUMBER function is the superior choice, as it automatically accounts for common symbols and scales percentages appropriately.

For more challenging scenarios involving heterogeneous data where numbers are buried within descriptive text, the advanced **SPLIT/CONCATENATE** formula combination offers a powerful,

albeit complex, extraction methodology. Regardless of the method used, always employ the ISNUMBER function as a final diagnostic step to ensure the integrity of your converted numerical values before commencing any analytical work. Mastery of these techniques ensures robust and reliable data processing across all of your spreadsheet projects.

ARABPSYCHOLOGY.COM