How do I get the column letter in Google Sheets (with example)?

Authored by

stats writer

November 18, 2025

The ubiquity of Google Sheets makes it an indispensable tool for data analysis, reporting, and complex financial modeling across countless organizations. When working with large datasets or developing dynamic spreadsheet architecture, a common requirement arises: converting a numerical column number (e.g., 8) into its corresponding alphabetical column letter (e.g., H). This conversion is crucial when building formulas that rely on dynamic cell reference generation or when interfacing with external scripts.

While this might seem like a complex programming task, Google Sheets offers an elegant, combined function solution that achieves this goal with remarkable efficiency. Understanding how to execute this transformation allows users to create significantly more flexible and robust spreadsheets, moving beyond static references to fully automated data processing. This comprehensive tutorial will guide you through the precise formula required, provide detailed, practical examples, and thoroughly explain the underlying mechanics of the functions utilized.

By the end of this expert guide, you will possess a clear, authoritative method for obtaining the column letter for any given column index in your spreadsheet environment, enabling superior automation capabilities. We focus specifically on using the ADDRESS function in tandem with the SUBSTITUTE function to reliably achieve the desired output.

The Combined Formula for Column Letter Retrieval

To dynamically retrieve the column letter corresponding to a specific numeric column index in Google Sheets, we employ a sophisticated concatenation of two primary functions: ADDRESS and SUBSTITUTE. The key challenge is that the ADDRESS function, while excellent for creating cell coordinates, always includes the row number, which must be subsequently stripped away.

The following syntax represents the authoritative method for obtaining the column letter that correlates to a specified column number. In the example provided below, we are targeting column number 8. Note the specific arguments used in the ADDRESS function to ensure we generate a result that is easily parsable.

=SUBSTITUTE(ADDRESS(1,8,4),"1","")

This meticulously constructed formula is designed to first generate a complete cell reference and then isolate only the alphabetical component. Specifically, this arrangement will return the column letter that corresponds to column 8 in the spreadsheet, which is typically the letter “H”. The structure is highly adaptable; by merely changing the numeric argument “8” to any other positive integer, you can retrieve the corresponding column letter for any column index.

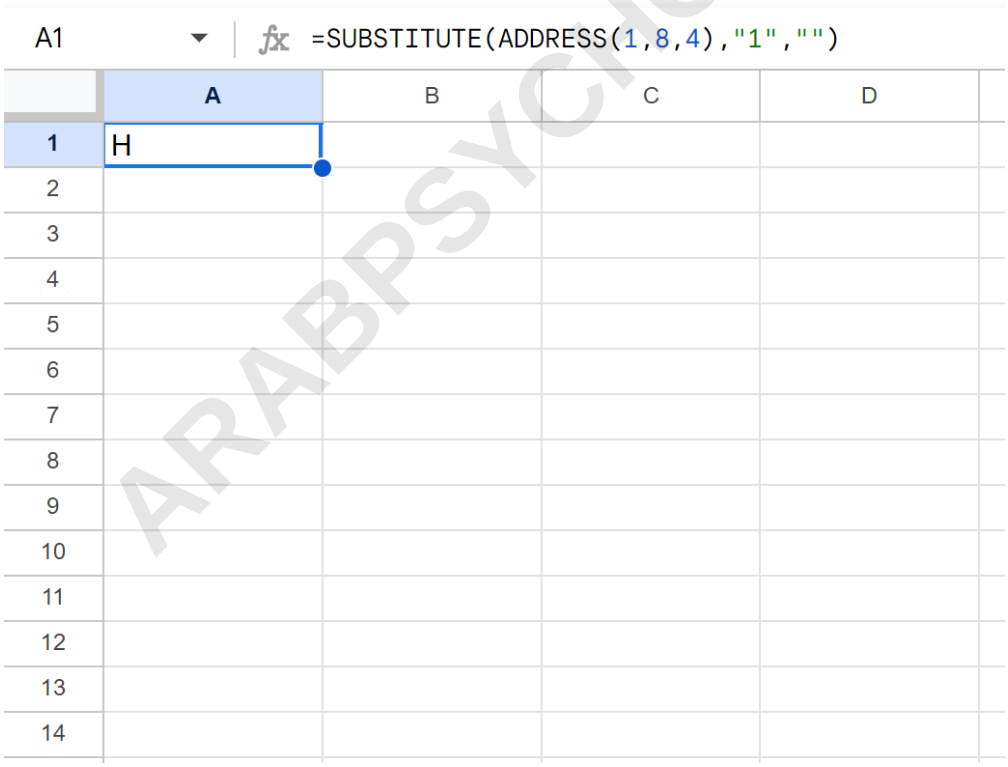
Step-by-Step Example: Retrieving Column 'H'

To illustrate the practical implementation of this technique, let us consider a scenario where we need to return the column letter for the eighth column of our Google Sheets spreadsheet. This is often required when constructing lookup tables or when the column index is derived from another calculation within the sheet.

We begin by inputting the formula directly into a target cell, such as A1, where we wish the result to appear. The inclusion of the value “8” as the column argument within the ADDRESS function tells the formula precisely which column number we are attempting to convert. Observe the formula structure used for this specific requirement:

=SUBSTITUTE(ADDRESS(1,8,4),"1","")

Upon execution, the formula successfully isolates the column letter “H”, confirming that the eighth column is correctly identified. The process is visualized in the subsequent screenshot, demonstrating the formula's input and its immediate, desired output within the spreadsheet interface.



The result clearly shows “H”, since this is the alphabetical designation that corresponds perfectly to the eighth column number in the standard spreadsheet layout. This validates the effectiveness of

the combined ADDRESS and SUBSTITUTE approach for precise column letter retrieval.

Adapting the Formula for Diverse Column Indices

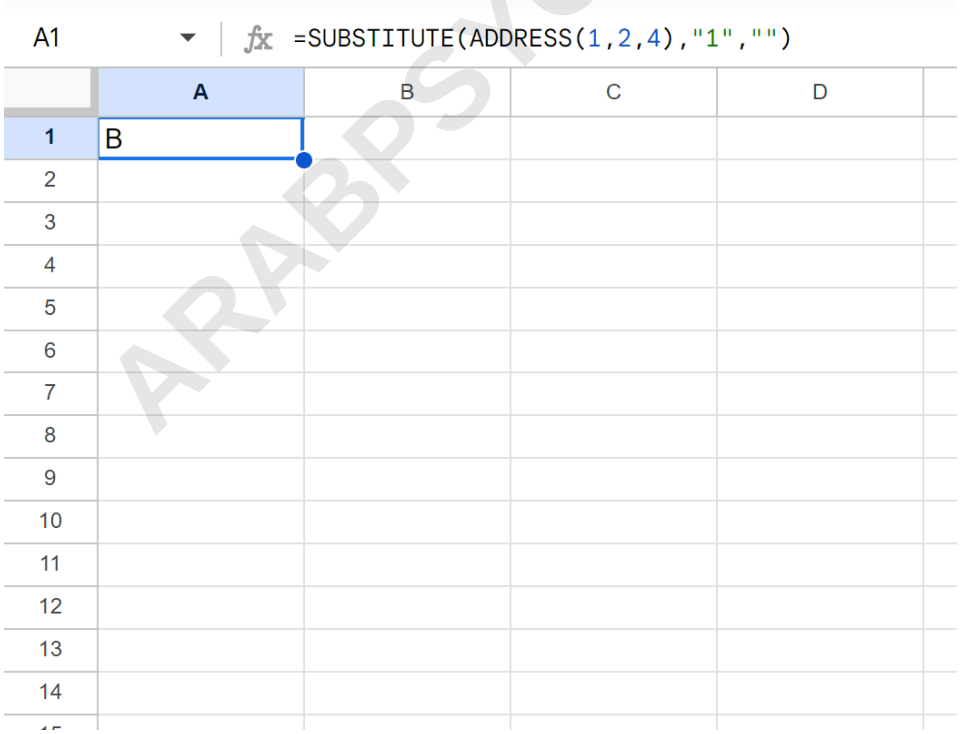
The true power of this formula lies in its flexibility. To retrieve the column letter for any other column, the user simply needs to modify the numeric argument currently set to 8 within the ADDRESS function. This allows for seamless transitions between referencing early columns, middle columns, or even those far off to the right (e.g., column 50, which corresponds to “AX”).

For instance, imagine we need to retrieve the letter corresponding to the second column, which is “B”. We only need to change the column index parameter from 8 to 2, maintaining all other parameters of the formula exactly as they are. This preserves the isolation logic built into the SUBSTITUTE function.

The revised formula targeting the second column is presented below:

=SUBSTITUTE(ADDRESS(1,2,4),"1","")

After implementing this adjustment, the formula returns “B”. This demonstrates the simple and intuitive nature of adapting this solution. The subsequent visualization confirms the successful execution of the formula for the second column:



Deconstructing the Formula: Understanding ADDRESS and SUBSTITUTE

To fully appreciate the robustness and efficiency of this method, it is essential to understand the individual roles played by the ADDRESS function and the SUBSTITUTE function. Recall the core formula used to obtain the letter corresponding to the eighth column:

=SUBSTITUTE(ADDRESS(1,8,4),"1","")

The calculation is executed sequentially, starting with the innermost function, ADDRESS. The ADDRESS function takes the structure ADDRESS(row, column, , ,). In our usage, we provided three critical parameters: 1 for the row, 8 for the column number, and 4 for the reference type. By specifying row 1, we ensure that the resulting cell reference is the shortest possible string containing the column letter. The final argument of 4 is crucial; it specifies that the returned reference must be relative (not absolute), adhering to the A1 notation standard, yielding “H1”.

Once the ADDRESS function returns “H1”, the result is passed immediately to the outer SUBSTITUTE function. The SUBSTITUTE function operates as SUBSTITUTE(text_to_search, search_for, replace_with,). We instruct it to search the resulting text (“H1”) for the specific string “1” and replace it with a blank string (“”). This perfectly isolates the remaining component, which is the desired column letter, “H”.

Why This Method is Preferred Over Alternatives

While various complex methods exist, including using `INDEX` combined with `COLUMN` within an `ARRAYFORMULA`, or creating custom functions using Google Apps Script, the combined ADDRESS/SUBSTITUTE method stands out due to its simplicity and native spreadsheet efficiency. It relies solely on built-in functions, making it portable and easily understood by analysts familiar with core spreadsheet logic.

Other methods often require array manipulation or dealing with complex numerical patterns (since column letters function as a base-26 numbering system), which can be computationally intensive or difficult to debug. This approach, however, leverages the ADDRESS function's inherent ability to translate numbers into column letters, thereby simplifying the task significantly. It relies on a straightforward textual replacement (SUBSTITUTE) rather than requiring complex mathematical or iterative processes.

Furthermore, using the ADDRESS/SUBSTITUTE combination ensures compatibility across different versions of Google Sheets and minimizes potential errors associated with formula overhead. For generating a single, reliable column letter from a numeric input, this combination remains the industry-standard, most practical technique.

Conclusion: Mastering Dynamic Cell Referencing

The ability to convert a numeric column number into its corresponding column letter is a fundamental skill for advanced spreadsheet manipulation. This technique, utilizing the powerful nesting of the ADDRESS and SUBSTITUTE functions, allows users to overcome the limitations of static cell definitions, opening up possibilities for dynamic report generation and automated scripting within Google Sheets.

By using ADDRESS(1, N, 4) to generate a cell string and then applying SUBSTITUTE(..., “1”, “”) to clean up the row component, you achieve a clean, reliable, and easily adaptable solution. Whether you are dealing with column 8 (“H”) or column 200 (“GR”), the logic remains universally sound, ensuring accuracy regardless of the magnitude of the column index.

Mastering this technique is a significant step toward becoming an expert user of data platforms, enabling the construction of spreadsheets that are not only functional but also highly responsive to changing data structures and reference needs.