

Autofill Letters of the Alphabet in Google Sheets

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Autofill Letters of the Alphabet in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=94904>

Generating sequential lists of the alphabet is a common requirement in data management, whether for labeling categories, creating indices, or structuring complex spreadsheets. Manually typing out the entire alphabet or long sequences of letters in a program like Google Sheets is time-consuming and prone to errors. Fortunately, Google Sheets offers a highly efficient method to automate this process. By leveraging specific text manipulation functions, users can rapidly autofill alphabetical sequences with precision, dramatically reducing the preparation time required for data entry and organization tasks. This capability makes advanced use of Google Sheets an invaluable skill for anyone dealing with structured data.

This tutorial details the robust technique utilizing the fundamental CODE and CHAR functions. These functions are based on the standardized ASCII code system, which assigns a unique numerical value to every character, including both uppercase and lowercase letters. Understanding how to manipulate these codes is the key to successfully creating dynamic alphabetical series that seamlessly extend across any required range in your spreadsheet.

The Power of ASCII: Understanding CODE and CHAR Functions

When the need arises to quickly populate a column with the entire alphabet, from A to Z, traditional drag-and-drop autofill methods often fail, as spreadsheet programs typically do not recognize the alphabet as a standard numerical or date series. To bypass this limitation, we rely on the internal character encoding used by computers--specifically, the ASCII code system. The CODE function converts a character into its corresponding ASCII code number, and the CHAR function performs the reverse operation, transforming a number back into its character equivalent. This duality allows us to treat letters numerically, enabling the automatic generation of subsequent characters simply by adding 1 to the numerical code.

Step-by-Step Guide: Generating the Alphabet Sequence

Implementing this solution is straightforward and requires only two primary steps: establishing a starting point and then creating a dynamic formula that can be efficiently copied down the column. This method ensures maximum flexibility, allowing you to start your sequence at any desired letter and control the case (uppercase or lowercase) of the resulting sequence.

Initial Setup: Inputting the Starting Letter

The first critical step involves setting up the anchor point for your sequence. Unlike numerical series where you might input two numbers (e.g., 1 and 2) to establish a pattern, the alphabetical series only requires the first character. This character dictates the starting point and the case (uppercase or lowercase) for the entire sequence.

For this comprehensive example, we will begin with the capital letter "A." Navigate to cell **A1** in

your [Google Sheets](#) document and manually type the letter **A**. This cell reference (**A1**) will be the input argument for our primary generating formula.

	A	B	C	D
1	A			
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				

Applying the Core Formula

The next step is to introduce the calculation logic. This formula must be entered into the cell immediately following the starting character, which, in our case, is cell **A2**. This single formula is engineered to calculate the character that sequentially follows the character in the previous cell.

Input the following expression precisely into cell **A2**:

=CHAR(CODE(A1)+1)

Upon execution, the system will look at the character in **A1** ("A"), convert it to its numerical [ASCII code](#) (65), increment that value by one (66), and then use the [CHAR](#) function to translate the new code back into its corresponding letter, which is **B**. This output confirms the formula is correctly generating the immediate successor character, as shown below:

	A	B	C	D
1	A			
2	B			
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Utilizing Autofill for Complete Sequences

With the core formula established in **A2**, the final step involves applying the spreadsheet program's inherent autofill functionality to replicate the pattern downwards. Because the cell reference **A1** is relative within the formula **=CHAR(CODE(A1)+1)**, dragging the formula down causes the reference to shift automatically (e.g., the formula in **A3** will reference **A2**, **A4** will reference **A3**, and so on). This creates a continuous chain reaction of incrementing ASCII code values.

To execute the autofill: hover your cursor over the bottom right corner of cell **A2**. When the pointer changes to a small cross icon (+), click and drag the formula down the column to generate the desired number of sequential letters. For a full alphabet run, you will typically drag it down 25 cells (A2 through A26).

A2 fx =CHAR(CODE(A1)+1)

	A	B	C	D
1	A			
2	B			
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				

The successful execution of the drag operation results in a perfectly formed, complete column of alphabetical characters, sequentially ordered from A through Z, ensuring high data quality and consistency.

	A	B	C	D
1	A			
2	B			
3	C			
4	D			
5	E			
6	F			
7	G			
8	H			
9	I			
10	J			
11	K			
12	L			
13	M			
14	N			
15	O			
16	P			
17	Q			
18	R			
19	S			
20	T			
21	U			
22	V			
23	W			
24	X			
25	Y			
26	Z			
27				

Case Sensitivity: Handling Uppercase and Lowercase Letters

A crucial detail when using the CODE and CHAR method is recognizing the separation of character cases within the ASCII code set. Uppercase letters (A-Z) have numerical values ranging from 65 to 90, while lowercase letters (a-z) range from 97 to 122. Because these two ranges are distinct and do not overlap, the case of the starting character in cell **A1** entirely determines the case of the entire resulting sequence.

If you initiate the series using a lowercase 'a' in cell **A1**, the formula will automatically generate the corresponding lowercase sequence (a, b, c, ...). Conversely, starting with an uppercase 'A' generates the uppercase sequence. This allows for simple control over the output format without

needing to modify the underlying formula structure.

Deconstructing the Formula: CODE and CHAR Explained

To fully appreciate the efficiency of this technique, it is beneficial to analyze the nested structure of the generating formula: **=CHAR(CODE(A1)+1)**. This sequence of operations is fundamental to character manipulation in [Google Sheets](#).

The process follows these steps sequentially:

The inner function, **CODE(A1)**, is executed first. Since cell **A1** contains "A," the **CODE** function returns its corresponding numerical value: 65.

The structure then takes this output (65) and adds 1: **65 + 1 = 66**. This increment is what drives the sequential nature of the list.

Finally, the outer function, **CHAR(66)**, takes the resulting integer (66) and converts it back into its textual character representation. The character associated with ASCII code 66 is "B."

When this formula is dragged down to cell **A3**, the relative reference updates to **=CHAR(CODE(A2)+1)**. Cell **A2** now holds "B" (ASCII 66), leading to a final result of $66 + 1 = 67$, which the **CHAR** function converts to "C." This iterative process ensures that every subsequent cell correctly generates the next letter in the sequence until the column is completely populated.

Efficiency Through Automation

Mastering methods like the combined use of the **CODE** and **CHAR** functions moves users beyond basic spreadsheet tasks and into powerful data automation. This technique is not limited solely to generating the primary alphabet; it can be adapted to create sequential patterns of almost any character that adheres to the **ASCII code** standard. Such automated processes minimize the risk of human error in transcription and significantly accelerate list generation, proving to be an indispensable tool for efficient data preparation and analysis in [Google Sheets](#) environments.