

How to Remove the First Two Digits from a Cell in Google Sheets

Authored by
stats writer

January 17, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Remove the First Two Digits from a Cell in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126417>

Data cleaning and manipulation are essential tasks in modern spreadsheet management, particularly when dealing with large datasets imported from various sources. A frequent requirement is the need to standardize data formats by removing extraneous prefixes, such as status codes, two-digit identifiers, or old numerical sequences attached to records. While various functions exist within the spreadsheet environment, achieving reliable removal of the first N characters requires a sophisticated combination of tools designed for text handling.

Although some users initially consider utilizing the LEFT function to achieve this result--by simply extracting everything except the unwanted prefix--this approach lacks flexibility and is often prone to errors when dealing with variable-length input strings. A much more robust and universally applicable technique involves nesting two distinct functions: the RIGHT function and the LEN function. This powerful combination dynamically calculates the exact length of the remaining data, ensuring that regardless of the total length of the original cell content, precisely the first two characters are excluded from the output.

This comprehensive guide details how to leverage the combined power of these textual manipulation tools within Google Sheets to efficiently and reliably strip the first two digits or characters from any target cell. We will explore the underlying logic, provide step-by-step application instructions, and discuss potential pitfalls and advanced considerations for maintaining data integrity during these critical transformation processes. Mastery of this technique is fundamental for anyone performing complex data processing in the cloud-based spreadsheet environment.

In many scenarios involving imported data or generated identification codes, it is often necessary to systematically remove the initial two digits or characters from a cell in Google Sheets.

To accomplish this task with maximum reliability and dynamic capability, you must use the **RIGHT function** combined strategically with the **LEN function**. This combination creates a formula that adapts to the length of the input data:

=RIGHT(A2,LEN(A2)-2)

This specific formula is engineered to remove the first 2 characters from the content housed within cell **A2**, regardless of how long the remaining string is.

For instance, if cell **A2** contains the value **AA2806**, the execution of this formula would return the desired substring, which is simply **2806**. The formula effectively calculates the total length (6), subtracts the two unwanted characters, and then instructs the RIGHT function to extract the remaining 4 characters from the right side of the string.

The following detailed example illustrates the process of deploying this powerful formula in a real-

world data context.

Understanding the Core Functions: RIGHT and LEN

To truly appreciate the elegance and efficiency of the `=RIGHT(A2, LEN(A2)-2)` structure, it is imperative to understand the individual roles played by the two key functions involved. The RIGHT function is fundamentally designed to extract a specified number of characters from the right side of a given text string. Its basic syntax is `=RIGHT(string,)`. If you knew the string was always exactly six characters long, you could hardcode 4 as the number of characters to extract (6 minus 2). However, data seldom maintains such uniformity, necessitating a dynamic measurement mechanism.

This is where the LEN function becomes crucial. LEN stands for 'Length,' and its sole purpose is to return the total number of characters in a text string. Its syntax is simply `=LEN(string)`. By embedding the LEN function within the RIGHT function, we achieve the dynamic calculation required for universal application across varying data lengths. The LEN function first measures the total length of the content in the reference cell (e.g., A2). This total length then acts as the primary input for the subtraction operation.

The calculation `LEN(A2) - 2` determines the exact length of the resultant substring we wish to retain. If A2 contains 10 characters, the result is 8. If A2 contains 5 characters, the result is 3. This calculated number (N-2) is then passed as the second argument to the RIGHT function. Thus, the RIGHT function extracts exactly that calculated number of characters, starting from the rightmost position. This nested approach ensures that irrespective of the original string's size--whether it is five digits or twenty characters long--only the first two characters are reliably excluded, providing a robust solution for data standardization.

Step-by-Step Implementation Example

To solidify the theoretical understanding, let us walk through a practical scenario involving a dataset of employee identifiers. Suppose a human resources department has compiled a list of employee IDs where the first two characters denote the region or department code, which must be removed before integrating the data into a centralized system that only uses the unique sequential ID. This type of data transformation is highly common and easily managed using the nested formula structure we have identified.

Suppose we have the following list of employee IDs in Google Sheets, located in Column A:

	A	B	C
1	Employee ID		
2	AA2806		
3	AB9450		
4	AC3405		
5	BB8935		
6	AA3902		
7	AC7855		
8	AB0021		
9	AA2340		
10	AA8405		
11	HB9984		
12	AB3004		
13			
14			
15			

Our objective is to create a new, standardized column (Column B) that displays these employee IDs without the initial two-character prefix. We must apply the dynamic truncation formula starting in the cell adjacent to the first data point, which is cell **B2**, corresponding to cell A2.

To achieve the desired result, we carefully construct the formula in cell **B2**. We instruct the spreadsheet to calculate the length of A2, subtract 2, and then extract that resulting number of characters from the right side of A2. The precise formula entered into cell **B2** is as follows:

=RIGHT(A2,LEN(A2)-2)

After entering the formula into B2, the spreadsheet immediately calculates the output, displaying the truncated value. The key advantage of this setup is its immediate scalability. Instead of manually repeating the process for every cell, spreadsheet applications like Google Sheets allow for quick application across an entire range. We simply use the fill handle (the small square at the bottom-right corner of cell B2) and drag this formula down to apply it to every remaining cell in Column B. This action automatically adjusts the cell reference (A2 becomes A3, A4, and so on) for each row, processing the entire dataset efficiently.

Analyzing the Results and Data Integrity

Upon successfully dragging the formula down column B, we observe the transformation of the

employee ID data. Column B now presents a clean, standardized list derived directly from Column A, with the first two digits effectively removed from every entry. This demonstrates the formula's ability to handle diverse data entries simultaneously, confirming the dynamic nature of the LEN subtraction component. The visual result confirms the operation:

B2 ▾ | **fx** =RIGHT(A2,LEN(A2)-2)

	A	B	C
1	Employee ID	First 2 Digits Removed	
2	AA2806	2806	
3	AB9450	9450	
4	AC3405	3405	
5	BB8935	8935	
6	AA3902	3902	
7	AC7855	7855	
8	AB0021	0021	
9	AA2340	2340	
10	AA8405	8405	
11	HB9984	9984	
12	AB3004	3004	
13			
14			
15			

Column B now displays the employee ID's originally found in column A, but with the first two digits definitively truncated from each employee ID. This process is non-destructive to the original data in Column A, which is a significant advantage in data processing. By maintaining the raw data, you ensure that you can always revert to the original state or perform alternative analyses if required, adhering to best practices in data governance.

It is crucial to understand that the output in Column B is initially a calculated value, meaning it relies entirely on the formula referencing Column A. If the original data in Column A is modified, the results in Column B will update instantly. While this responsiveness is usually desirable, there are scenarios where you might need the results in Column B to be static values, disconnected from the source. To achieve this, you must copy the calculated cells in Column B and then paste them back into the same location (or a new location) using the 'Paste values only' option. This converts the dynamic formula output into static text or numerical data.

Maintaining data integrity during such transformations involves careful validation. After applying the formula, one should always spot-check several entries, especially those with unusual lengths, to

ensure the two-character removal worked as intended. This process verifies that the source data did not contain unexpected leading spaces or non-printable characters that could distort the LEN function's count. For instance, if an ID was accidentally entered as ' 123456' (with two leading spaces), the LEN function would count those spaces, and the RIGHT function would return '123456', failing to remove the intended numerical prefix. Mitigation strategies for leading spaces are discussed in a later section.

Detailed Breakdown of the Formula Logic

The efficiency of the `=RIGHT(A2, LEN(A2)-2)` structure stems from its precise execution order and the reliance on dynamic input generated by the nested function. Understanding this sequence is key to adapting the formula for different needs, such as removing three characters instead of two. The processing unfolds in a strict, predictable manner, beginning with the innermost component.

Inner Function Execution (LEN): The LEN function, `LEN(A2)`, is executed first. It reads the content of cell A2 and returns an integer representing the total count of characters, including letters, numbers, symbols, and spaces. For example, if A2 contains "TR98765", `LEN(A2)` returns 7.

Subtraction Operation: The total length (7, in our example) is immediately subjected to the arithmetic operation: `7 - 2`. The value '2' represents the fixed number of initial characters we intend to remove. The result of this operation is 5. This resulting number dictates how many characters must be extracted for the final result.

Outer Function Execution (RIGHT): The calculated number (5) is passed as the second argument to the RIGHT function. The function then looks at the original string "TR98765" and extracts five characters starting from the right end. The characters extracted are "98765".

Final Output: The cell displays "98765", achieving the goal of removing the first two characters ("TR"). This dynamic approach guarantees accurate removal even if the next cell, A3, contained a longer string like "TR1000000", which would result in `LEN(A3)-2 = 10-2 = 8`, and the RIGHT function would extract the last 8 characters.

This nested function architecture provides significantly greater accuracy than attempting to use text-to-columns tools, which can often fail when delimiters are not consistent or when dealing with mixed data types. By keeping the manipulation strictly within the realm of text functions, we leverage the spreadsheet's built-in handling for strings, ensuring that leading zeros, for example, are treated as characters and not dropped prematurely as they might be if the data were converted to a numeric format too early in the process.

Handling Edge Cases: Variable Lengths and Spaces

While the RIGHT/LEN combination is highly robust, real-world data is rarely perfectly clean. Two primary edge cases often require further consideration: ensuring the data is truly text-based, and eliminating extraneous whitespace that might skew the LEN count. Since the formula relies on the total character count, any hidden or visible leading spaces will be counted, potentially causing the removal operation to fail on the intended data prefix.

For example, if cell A2 contains ' 123456' (note the leading space), `LEN(A2)` will return 7. The calculation `7 - 2` yields 5. The RIGHT function then extracts the last 5 characters, resulting in '123456'. In this scenario, the intended prefix '12' was not removed because the formula removed the space and the digit '1'. To correct this, we must first clean the string before measuring its length.

The standard solution for dealing with unwanted whitespace is to introduce the TRIM function. The TRIM function removes all leading, trailing, and excessive internal spaces, leaving only single spaces between words. We integrate TRIM into our existing formula structure by wrapping it around the reference cell A2. The enhanced, more resilient formula becomes:

```
=RIGHT(TRIM(A2), LEN(TRIM(A2))-2)
```

By applying TRIM to A2, we ensure that the LEN function measures the length of the cleaned string and the RIGHT function operates on the cleaned string. This critical addition dramatically increases the formula's reliability across datasets where inconsistencies in user entry or data import procedures might have introduced leading or trailing whitespace. It is best practice to always incorporate TRIM when working with user-generated text data to prevent these common character-counting errors.

Alternative Text Manipulation Methods

While the RIGHT/LEN approach is optimal for consistently removing a fixed number of leading characters regardless of the total string length, Google Sheets offers other powerful functions that may be suitable for related tasks or specific niche scenarios. Understanding these alternatives allows users to select the most efficient tool based on the exact requirements of the data cleaning task.

One alternative involves the **MID function**, which extracts a substring starting from a specified position. To remove the first two characters, you would instruct MID to start at the third character and extract the rest of the string. The required formula would be `=MID(A2, 3, LEN(A2) - 2)`. This formula achieves the exact same result as the RIGHT/LEN combination. The '3' indicates the

starting position (the third character), and `LEN(A2) - 2` dynamically calculates how many characters remain to be extracted. For many users, this MID approach is conceptually simpler, as it directly addresses the desired starting point.

Another, more complex alternative involves using the **REPLACE function**, which replaces a specified number of characters in a string starting from a given position. To remove the first two characters, you would replace them with an empty string (""). The formula structure would be `=REPLACE(A2, 1, 2, "")`. Here, '1' indicates the starting position (the first character), '2' indicates the number of characters to replace, and the empty quotation marks ("") serve as the replacement text. This method is exceptionally clean and often the shortest formula to achieve the goal of prefix removal, though it operates differently by modifying the string rather than extracting a substring.

Finally, for cases where the prefix being removed is consistent (e.g., always "AB"), the **SUBSTITUTE function** is highly effective. You would use `=SUBSTITUTE(A2, "AB", "", 1)`. This instructs the function to substitute the text "AB" with nothing (""), but critically, the '1' specifies that only the first occurrence should be replaced. This prevents accidental removal of "AB" if it appears later in the string. While this is not ideal for general prefix removal (where the prefix might be variable, like "01", "02", "AA", etc.), it is the definitive method for removing a known, fixed leading substring.

Considerations for Data Type Conversion

When the first two characters are removed, the resulting output in Column B remains a text string, even if the remaining characters are purely numerical (e.g., "3456"). This is the standard behavior of text manipulation functions like RIGHT, MID, and REPLACE in Google Sheets. For most analytical purposes, such as sorting or concatenation, leaving the data as text poses no issue. However, if the result is intended for mathematical operations (calculations, summing, averaging), it must be explicitly converted into a numerical data type.

Attempting to use mathematical functions on text-formatted numbers can lead to unexpected errors or zero results. To ensure the resulting data can be used arithmetically, we must wrap our existing formula within a function that forces the conversion. The most common and reliable method is using the **VALUE function**. The VALUE function takes a text string that looks like a number and converts it into an actual numerical value, ready for computation.

If we apply the VALUE function to our original successful formula, the final version ready for mathematical analysis becomes:

`=VALUE(RIGHT(A2,LEN(A2)-2))`

It is important to note that converting text to numbers may drop leading zeros if they exist in the

remaining portion of the string (e.g., "0078" becomes 78). If leading zeros are structurally important (e.g., in postal codes or standardized serial numbers), the data should remain as a text string. The decision to implement the VALUE function should be based strictly on whether subsequent calculations require true numerical interpretation of the remaining data.

Adapting the Formula for Different Prefix Lengths

The beauty of the RIGHT/LEN approach is its easy adaptability to scenarios requiring the removal of more or fewer than two characters. The only element in the core formula that needs modification is the subtraction factor. If the requirement shifts from removing 2 characters to removing N characters, the formula is adjusted simply by replacing the '2' with 'N'.

Removing 3 Characters: If the prefix consists of three digits (e.g., "123"), the formula is adapted to subtract 3 from the total length: `=RIGHT(A2, LEN(A2) - 3)`.

Removing 5 Characters: If a standardized code prefix is five characters long (e.g., a zip code appended to a serial number), the formula becomes: `=RIGHT(A2, LEN(A2) - 5)`.

Removing 1 Character: For a simple single-character identifier: `=RIGHT(A2, LEN(A2) - 1)`.

This flexibility underscores why the nested RIGHT and LEN function combination is the preferred method for text truncation based on prefix length. It provides a generalized template that can be applied across numerous data cleaning tasks simply by adjusting a single integer value. When combined with the TRIM function for robust cleaning, this formula is highly reliable for large-scale data standardization projects within Google Sheets.

In conclusion, mastering text manipulation functions is non-negotiable for advanced data analysts working in spreadsheet environments. While various methods exist, the structure `=RIGHT(A2, LEN(A2) - N)` offers a dynamic, scalable, and highly adaptable solution for the common task of removing a fixed number of leading characters. By paying attention to whitespace (via TRIM) and subsequent data type needs (via VALUE), users can ensure accurate and functional data transformations.