Learning to Remove the First Two Digits from Cells in Google Sheets

Authored by

Mohammed looti

November 11, 2025

Efficient Data Cleansing: Removing Fixed Prefixes in Google Sheets

When managing extensive datasets, data integrity frequently depends on robust sanitation procedures. It is a common requirement to standardize information by removing extraneous prefixes, such as fixed-length codes or non-essential leading digits, from core identifiers. In the environment of Google Sheets, this often translates to the precise removal of a specific number of characters--for instance, the initial two digits--from the beginning of a cell's content. This cleanup is essential when preparing data for crucial tasks like normalization, database merging, or advanced analytical reporting where only the pure identifying sequence is needed. Developing expertise in basic text manipulation formulas is foundational for effective spreadsheet management and maintaining high data quality.

Fortunately, Google Sheets offers powerful, built-in functions designed to manage and dynamically manipulate text strings. While manual deletion might work for a small handful of entries, automating this critical process across thousands of rows demands a highly reliable, programmatic solution. The most universally applicable and dependable method for truncating a fixed prefix involves the strategic combination of two primary functions: the **RIGHT** function and the **LEN** function. This pairing enables the spreadsheet to intelligently calculate the exact remaining length of the data after the prefix has been discarded, guaranteeing accuracy regardless of the original total length of the string.

This comprehensive tutorial will detail the construction and deployment of this highly effective formula. We will demonstrate how to instruct the spreadsheet to first ascertain the total length of the cell content and subsequently extract the necessary subset of characters, starting from the right side, thereby effectively ignoring the first two leading characters. This precise technique is particularly valuable when processing standardized input formats, such as specific product codes or regional identifiers where the initial two positions act merely as categorical markers that are not required for core analysis.

The Dynamic Solution: Leveraging RIGHT and LEN Functions

To successfully implement the removal of the first two characters (whether they are digits or letters) from a designated cell in Google Sheets, we must construct a nested formula. This methodological approach relies on the capability of the **LEN function** to accurately determine the total length of the data string. This length information is then passed to the **RIGHT function**, which uses it to calculate precisely how many characters it needs to extract from the trailing end of the string. This methodology ensures that the solution is not only precise but also dynamic, adapting seamlessly to cell contents of varying lengths, provided they consistently share the defining characteristic of having a two-character prefix intended for removal.

The core formula required to execute this specific text transformation is structured as follows.

Assuming the source data that requires cleansing is located in cell **A2**, the formula works by first determining the total character length of the content in **A2**, then subtracting the two characters we wish to remove, and finally utilizing that calculated remainder to pull the relevant data from the right side of the string.

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

This specific implementation of the formula is designed to truncate the first two leading characters found within cell **A2**. To provide a concrete illustration, consider a scenario where cell **A2** holds the text **AA2806**. Initially, the **LEN function** processes this input and returns 6, representing the total number of characters. The calculation then executes 6 minus 2, yielding the result 4. Subsequently, the **RIGHT function** is executed, extracting the rightmost 4 characters, which successfully produces the desired result: **2806**. This powerful integration of the **RIGHT** and **LEN** functions allows for highly precise and reliable text manipulation tasks.

Practical Application: A Step-by-Step Guide

To clearly illustrate the practical application of this text manipulation technique, let us examine a typical real-world scenario involving a set of standardized identification numbers. Imagine you are working with a dataset of employee identifiers where every ID is prefixed with a two-digit department code that is now obsolete or unnecessary for your current analytical goals. Your objective is to efficiently isolate the unique five-digit core ID from the complete seven-character identifier.

Our starting point is the following list of employee IDs, situated in column A of our Google Sheets document. It is important to note that every entry uniformly adheres to the pattern of two prefix characters followed by five core identifier digits.

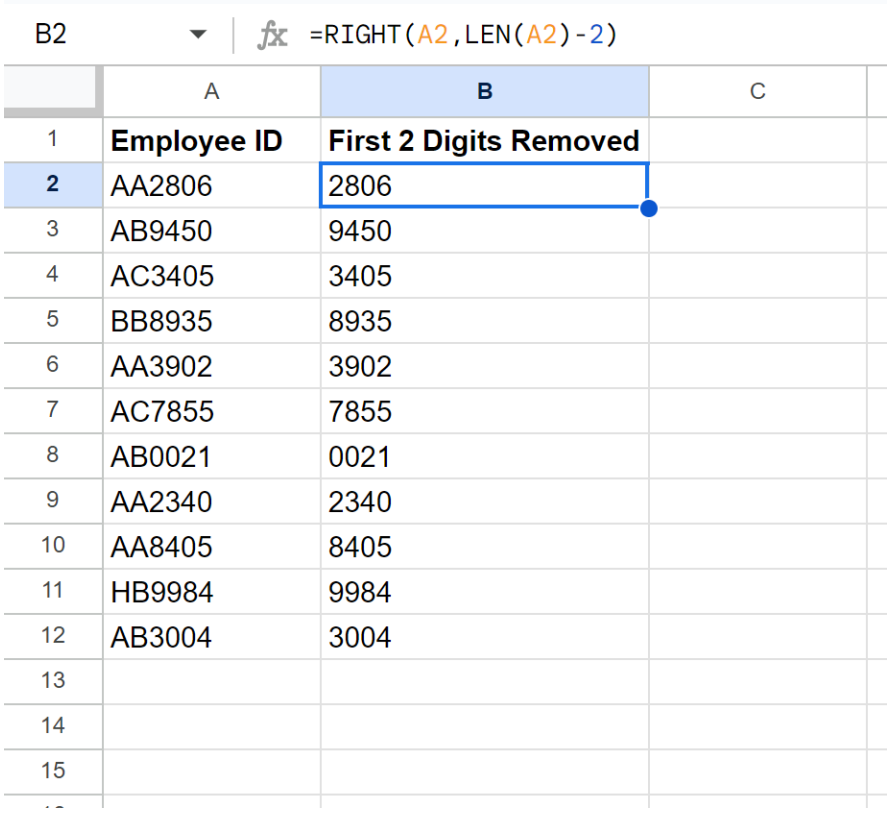
	A	B	C
1	**Employee ID**		
2	AA2806		
3	AB9450		
4	AC3405		
5	BB8935		
6	AA3902		
7	AC7855		
8	AB0021		
9	AA2340		
10	AA8405		
11	HB9984		
12	AB3004		
13			
14			
15			

The core objective here is to streamline this data by systematically removing the initial two characters from every employee ID listed in column A, and then placing the resulting clean, truncated ID into column B. This crucial process ensures that subsequent data operations, such as linking data via VLOOKUP or performing unique counts, operate exclusively on the core numerical identifiers, thereby eliminating potential inconsistencies or errors introduced by non-essential, leading prefixes.

To achieve this transformation, we input the established formula into cell **B2**, specifically targeting the source data residing in cell **A2**:

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

Upon entering the formula into **B2**, the cell immediately calculates the total length of **A2** (which is 7), subtracts 2, and subsequently extracts the rightmost 5 characters, resulting in the desired, clean output. The next essential step involves extending this formula dynamically to all subsequent rows in column B. We accomplish this efficiently by utilizing the fill handle--the small square located at the bottom-right corner of cell B2--and dragging it down until the last row containing data in column A is covered.



	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			

As clearly demonstrated by the resulting output displayed in column B, the employee IDs originally housed in column A have been successfully processed, with the first two digits uniformly and accurately removed from every entry. This method proves exceptionally efficient for large-scale data cleansing operations, particularly when uniformity of the input string length is maintained, ensuring rapid and error-free transformations across extensive datasets.

Formula Mechanics: Dissecting RIGHT and LEN

To confidently apply this technique to diverse data scenarios and adapt it for future requirements (e.g., removing the first 3 or 4 characters), a profound understanding of how the **RIGHT** function and the **LEN function** operate, both independently and in combination, is paramount.

The function governing the actual extraction is the **RIGHT** function. Its syntax is straightforward: `RIGHT(text, )`. This function is designed to extract a specified count of characters, initiating the count from the right-hand side of the input text. Crucially, if the second argument, `number_of_characters`, is omitted, the function defaults to 1, returning only the final character. However, for our specific use case, we require it to return every character *except* the fixed two-character prefix.

This is precisely the point at which the **LEN function** fulfills its vital supporting role. The **LEN function** requires only one argument: `LEN(text)`. Its output is the total numerical length of the

input text string, meticulously counting every element, including spaces, numerals, and specialized symbols. By nesting the **LEN function** inside the **RIGHT function**, we achieve the dynamic calculation of the required extraction length.

The pivotal calculation step is `LEN(A2) - 2`. This subtraction determines the exact number of characters that must be preserved when the two-character prefix is removed from the total length. The resulting number is then supplied as the crucial second argument to the **RIGHT function**. Consequently, the complete formula instructs Google Sheets to extract a subset of the string whose length is precisely equal to the original string's length minus the two characters slated for removal. This guarantees that whether the original identifier is 7 characters long (e.g., AA12345) or 10 characters long (e.g., AB12345678), the first two prefix characters (AA or AB) are consistently truncated, leaving the remainder of the identifier perfectly intact.

Ensuring Robustness: Addressing Data Anomalies

Although the **RIGHT** and **LEN** combination is exceptionally reliable for structured data, real-world data often contains inconsistencies that can compromise formula output. Identifying and addressing these edge cases is critical for establishing robust data cleaning procedures.

One of the most common issues encountered is the inadvertent inclusion of extra spaces, especially leading spaces, within the cell contents. The **LEN function** treats every space as a valid, countable character. If, for example, cell **A2** contains a leading space followed by 'AB12345' (total length of 8), executing `=RIGHT(A2, LEN(A2)-2)` would result in 'B12345'. This occurs because the space is counted as the first character removed, and 'A' is counted as the second character removed. This unintended truncation of valid data severely compromises data consistency and integrity.

To effectively counteract the negative impact of rogue whitespace, the recommended best practice is to wrap the input cell reference (**A2**) within the **TRIM function** before it is processed by **LEN** or **RIGHT**. The **TRIM function** automatically removes any leading, trailing, and repeated internal spaces, thereby normalizing the data string. The enhanced, significantly more resilient formula for data cleansing is structured as follows:

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

Another critical consideration is the uniformity of the data length. This specific method fundamentally assumes that the prefix to be removed is consistently exactly two characters long and that the total length of the string is at least three characters. If the formula encounters a cell containing only one or two characters (e.g., 'A' or 'AB'), the result of the calculation `LEN(A2)-2` will be 0 or a negative number. While the **RIGHT function** is designed to gracefully handle non-

positive lengths by returning an empty string, it is prudent to implement error checking using functions like **IF** or **IFERROR**. A robust conditional check ensures that the truncation only proceeds if the string length is sufficient, otherwise returning the original value or a designated error message, thus preventing unexpected output in subsequent analytical steps.

Advanced Techniques: MID and REGEXEXTRACT Alternatives

Although the **RIGHT/LEN** pairing represents the optimal solution for fixed-length prefix removal, Google Sheets provides alternative functions that may be preferred, depending on the user's familiarity with advanced text processing or the complexity of the underlying data structure.

Using the MID Function

The **MID function** offers a direct way to extract a substring by specifying both a starting point and a desired extraction length. Its syntax is defined as: `MID(string, starting_at, extract_length)`. To effectively remove the first two characters, we must initiate the extraction process at the third character position and continue extracting until the absolute end of the string is reached.

The `string` being manipulated is **A2**.
The `starting_at` position is 3 (as we intend to skip the initial two characters).
The `extract_length` must be calculated dynamically as `LEN(A2) - 2`.

Consequently, the **MID** formula is structurally very similar to the **RIGHT/LEN** method previously discussed:

=MID(A2, 3, LEN(A2)-2)

Functionally, this formula achieves an identical result to the **RIGHT/LEN** method when performing fixed prefix removal. Some spreadsheet users find the **MID** function more transparent because they explicitly define the start position (3), which clearly signifies that the first two positions are being intentionally bypassed.

Using the REGEXEXTRACT Function

For advanced users who require extraction based on complex patterns, or who deal with prefixes of variable length conditional on specific criteria, the **REGEXEXTRACT** function offers the most powerful and flexible solution. This function harnesses the power of regular expressions to define a precise search pattern and extract only the corresponding matching part of the string.

To remove the first two characters using this method, we define a pattern that matches the initial

two characters (represented by `^..`) and simultaneously captures everything that immediately follows (represented by `(.*)`). The caret symbol (`^`) anchors the pattern search to the absolute start of the string, and the dot (`.`) matches any single character. The parentheses define a capturing group, which is the exact portion that **REGEXEXTRACT** will return.

=REGEXEXTRACT(A2, "^..(.+)$")

This formula instructs the spreadsheet to recognize any two characters at the beginning of the string (`^..`) and then capture the entire remaining portion of the string (`(.+)$`). While this expression is initially more challenging to construct, **REGEXEXTRACT** is invaluable if the prefix itself needs to meet specific criteria--for example, if you only want to remove the first two characters if they are strictly letters, but not if they are numbers.

Conclusion and Further Text Manipulation Resources

The objective of removing the first two characters from a cell in Google Sheets is most reliably and effectively achieved by deploying the combined functionality of the **RIGHT function** and the **LEN function**. This nested formula guarantees that the extraction process is entirely dynamic, adapting correctly to data inputs of various lengths, provided the unwanted prefix remains fixed at two characters. For optimal data quality and robust cleansing, especially when processing user-inputted data, the integration of the **TRIM** function is strongly recommended to neutralize the disruptive impact of unexpected whitespace.

The underlying principle demonstrated here--calculating the total length of a string and systematically subtracting the length of the unwanted prefix--is a fundamental and highly adaptable concept in spreadsheet text manipulation. This technique can be readily adapted to remove any fixed number of leading characters simply by modifying the subtraction value (e.g., using `-3` to remove the first three characters).

The following resources detail how to perform other common text manipulation and essential data cleaning operations within Google Sheets:

How to accurately remove leading zeros from a numerical string.
Effective use of the **LEFT** function for extracting fixed prefixes.
Comprehensive methods for splitting text data based on defined delimiters.
Advanced applications of **REGEXREPLACE** for cleaning complex data patterns.