

Excel: Remove First and Last Character from String

Authored by
Mohammed loot

November 10, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Excel: Remove First and Last Character from String*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15432>

The Core Challenge: Manipulating Text Strings in Excel

It is a remarkably common task in data management to require the precise extraction of information from within a larger text block. Often, datasets contain identifier codes, product serials, or employee keys where the first and last characters serve as administrative markers, delimiters, or simply noise that must be stripped away to reveal the true core value. Handling these text-based entries, known formally as a [string](#), requires sophisticated use of Microsoft [Excel](#)'s built-in [Text Functions](#).

While a simple find-and-replace operation might suffice for static, recurring characters, the challenge intensifies when the length of the string varies dynamically across your dataset. In such cases, a solution that removes exactly one character from the start and exactly one character from the end, regardless of the overall length, becomes essential for maintaining data integrity and consistency during cleaning or preparation phases.

Fortunately, [Excel](#) provides a powerful and elegant way to achieve this precise truncation by combining two key functions: the **MID function**, which handles extraction, and the **LEN function**, which provides the necessary dynamic length calculation. This combination allows us to create a single, efficient formula capable of processing thousands of rows of data simultaneously.

The Essential Formula: Using MID and LEN Functions

To successfully remove both the first character and the last character from any given text [string](#), we rely on the [MID function](#). The [MID function](#) is designed to extract a specific number of characters from the middle of a text string, starting at a designated position. Its syntax is structured as `MID(text, start_num, num_chars)`.

For our specific goal of removing the first character, the `start_num` argument must be set to **2** (since 1 is the first character we want to skip). The more challenging part is defining the `num_chars` argument--the total number of characters we need to extract. Since the string length is variable, we cannot use a fixed number. This is where the [LEN function](#) comes into play, which simply returns the total character count of the string.

By combining these functions, we instruct [Excel](#) to calculate the total length (N) and then subtract 2 from that length (N-2). This calculated value becomes the precise number of characters to extract. Since we already started at position 2, this extraction count ensures that the very last character of the original [string](#) is also excluded from the final output.

The complete formula to remove the first and last character from the string contained in cell **A2** is:

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

For example, if cell **A2** contains the string `AA4506`, the [LEN function](#) determines the length is 6. The formula then becomes `MID(A2, 2, 6-2)`, or `MID(A2, 2, 4)`. Starting at the second position (A), it extracts four characters, returning the desired result: `A450`.

Step-by-Step Example: Truncating Employee IDs

To illustrate the practical application of this powerful formula, let us consider a common business scenario involving a list of employee IDs. Suppose we have a dataset in Column A where every ID is prefixed by a department code and suffixed by a checksum character, both of which we need to eliminate to isolate the unique internal identifier.

We begin with the following raw data residing in Column A of our [Excel](#) sheet:

	A	B	C	D	E
1	Employee ID				
2	AA4506				
3	BV9002				
4	GH3484				
5	FH3400				
6	AA2003				
7	AB3044				
8	AC9003				
9	DV4599				
10	AR3040				
11					
12					
13					
14					
15					
16					
17					

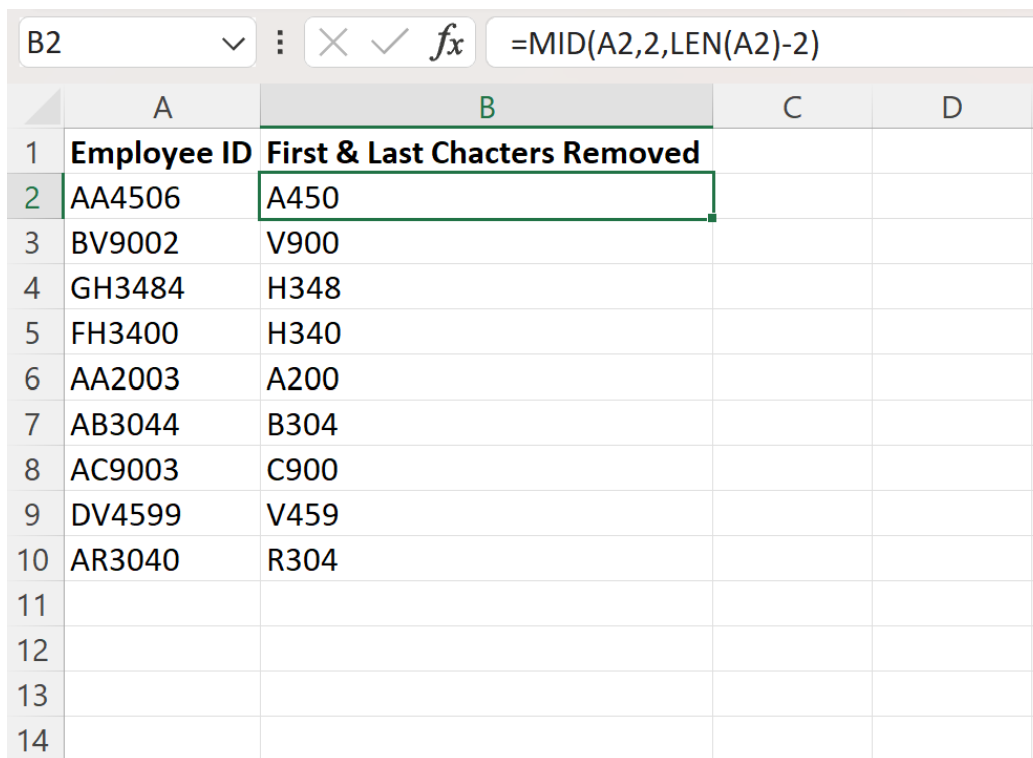
Our objective is to populate Column B with the cleaned employee IDs, having removed the first and last characters from each entry in Column A. We start by entering the formula into cell **B2**, ensuring we reference the corresponding data cell **A2**:

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

Once the formula is correctly entered in **B2**, we can swiftly apply this logic to the remainder of the dataset. By clicking and dragging the fill handle (the small square at the bottom right corner of the

selected cell) down column B, [Excel](#) automatically adjusts the cell references (A3, A4, etc.) for each subsequent row. This process instantly cleanses the entire list, irrespective of the individual length of the employee IDs.

The resulting transformation is immediate, providing a clean, standardized list of core identifiers in Column B, demonstrating the efficiency of using dynamic [LEN function](#) calculations for large-scale data cleansing operations:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Employee ID	First & Last Chacters Removed		
2	AA4506	A450		
3	BV9002	V900		
4	GH3484	H348		
5	FH3400	H340		
6	AA2003	A200		
7	AB3044	B304		
8	AC9003	C900		
9	DV4599	V459		
10	AR3040	R304		
11				
12				
13				
14				

The formula bar at the top shows the formula: `=MID(A2,2,LEN(A2)-2)`

Deconstructing the Logic: How MID and LEN Collaborate

A deeper understanding of how the components of the formula interact is essential for troubleshooting and adapting this technique for future text manipulation needs. Let us revisit the core expression:

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

The first argument, `A2`, simply specifies the target [string](#) we are working with. The second argument, the starting position `2`, is the mechanism for removing the first character. Because text positions in [Excel](#) are 1-indexed (the first character is position 1), starting the extraction at position 2 immediately skips the initial character, achieving the first part of our goal.

The third argument, `LEN(A2)-2`, is the key to dynamic length management and the removal of the

final character. The **LEN function** calculates the total length of the [string](#). By subtracting 2, we are reducing the required extraction length by two characters total. Since we already skipped one character by starting at position 2, reducing the extraction count by one more ensures that the final character is also excluded.

In essence, the formula executes a simultaneous two-part truncation: the `start_num = 2` handles the left side truncation, and the `LEN(A2)-2` handles the right side truncation, making the **MID function** ideal for this symmetric removal task. This avoids the need for complex nesting that might be required if attempting to solve this using separate **LEFT function** and **RIGHT function** calls.

Advanced Considerations: Handling Errors and Whitespace

When dealing with real-world data, especially that which has been manually entered or imported from external systems, two issues frequently arise that can disrupt the clean execution of the MID/LEN formula: whitespace and insufficient string length.

The first critical factor to remember is that **blank spaces count as characters** within [Excel's Text Functions](#). If your data contains leading or trailing spaces (e.g.,), the MID/LEN formula will treat these spaces as the "first" or "last" character to be removed. If you intend to remove non-space characters, you must first clean the string using the **TRIM function**. The robust, clean formula would then be nested as follows:

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

The second major consideration is handling short strings. If the original string in cell A2 has a length of 2 or less (e.g., AB or C), the formula `LEN(A2)-2` will result in a length of 0 or a negative number. When the **MID function** is asked to extract a non-positive length of characters, it can return an empty [string](#) or, in some versions or contexts, an error.

To prevent errors or unwanted blank results when dealing with strings too short to be truncated (i.e., those with a length less than 3), it is best practice to wrap the entire operation in an **IF statement** combined with the **LEN function** as a gatekeeper. This ensures the truncation only occurs if the string is long enough to support the removal of two characters:

```
=IF(LEN(A2)<3, "", MID(A2, 2, LEN(A2)-2))
```

Alternative Methods for String Manipulation

While the MID/LEN combination is the most straightforward method for this specific task, it is helpful to understand alternative approaches, particularly those involving the **LEFT function** and

the **RIGHT function**. These methods demonstrate the flexibility of [Excel](#)'s text processing capabilities, although they often result in more complex, nested formulas.

One common alternative involves using the **RIGHT function** to handle the removal of the first character, and then using the **LEFT function** to handle the removal of the last character from the result. This requires two length calculations. First, we use `RIGHT(A2, LEN(A2)-1)` to extract everything except the first character. This resulting string is now one character shorter.

Next, we must apply the **LEFT function** to this shortened result, asking it to extract everything except its last character. Since the total length of the internal string is now `LEN(A2)-1`, we must tell the **LEFT function** to extract `(LEN(A2)-1)-1`, which simplifies to `LEN(A2)-2`.

The fully nested alternative formula is:

```
=LEFT(RIGHT(A2, LEN(A2)-1), LEN(A2)-2)
```

Although mathematically identical in its outcome, this alternative is less intuitive because it requires the user to track the changing length of the string through multiple steps. For simplicity, speed, and readability, the primary [MID function](#) approach is overwhelmingly recommended for symmetric string truncation.

Summary of Key Takeaways and Resources

Mastering the combination of the [MID function](#) and the [LEN function](#) is fundamental to effective data manipulation in [Excel](#). This technique provides a robust, dynamic solution for trimming character strings where the starting and ending characters must be removed regardless of the total length.

Remember these key points when implementing the solution:

Start Position: Always use `2` to skip the first character.

Extraction Length: Always use `LEN(Cell)-2` to skip the last character dynamically.

Data Cleaning: Pre-clean your data using the **TRIM function** if whitespace might be present.

Error Prevention: Use an **IF statement** to check if the string length is 3 or greater before attempting truncation.

The following tutorials explain how to perform other common operations in [Excel](#), further enhancing your data cleaning toolkit:

How to Extract the Nth Word from a String in Excel

How to Use the SEARCH Function in Excel

How to Use the SUBSTITUTE Function in Excel