

Learn How to Use Wildcards with Excel's Find and Replace Function

Authored by
Mohammed looti

November 10, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Use Wildcards with Excel's Find and Replace Function*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15543>

The Power of Wildcard Characters in Excel

The [Excel Find and Replace](#) functionality is a robust tool designed to quickly locate specific text strings and substitute them with desired characters or leave them blank. While standard searches are useful, the true efficiency of this feature is unlocked through the strategic use of [wildcard characters](#). These special symbols act as placeholders, allowing users to match patterns rather than just exact sequences.

Understanding how to deploy these characters is essential for anyone dealing with large datasets requiring sophisticated data cleaning or restructuring operations. By implementing wildcards, users can perform complex tasks, such as removing variable text segments or isolating key data points within inconsistent text formats, all in a single action across thousands of cells. This capability significantly streamlines data management workflows, saving considerable time and reducing the risk of manual errors.

Specifically, [Excel](#) supports several key wildcard applications. For the purpose of replacing content, the asterisk (*) is the most critical symbol, representing any sequence of characters. We will explore three primary methods for leveraging this powerful tool to target and replace specific segments of text relative to fixed strings.

Core Wildcard Replacement Methods

The flexibility offered by the asterisk wildcard enables three distinct and highly practical replacement operations. Mastering these techniques allows for precise manipulation of text strings based on their relative position within a cell.

Replacing All Characters Between Two Specific Strings: This technique uses the wildcard to define a variable block of characters bounded by two fixed strings (e.g., parentheses or specific words). The pattern targets the entire sequence, including the boundaries and all content between them, for replacement.

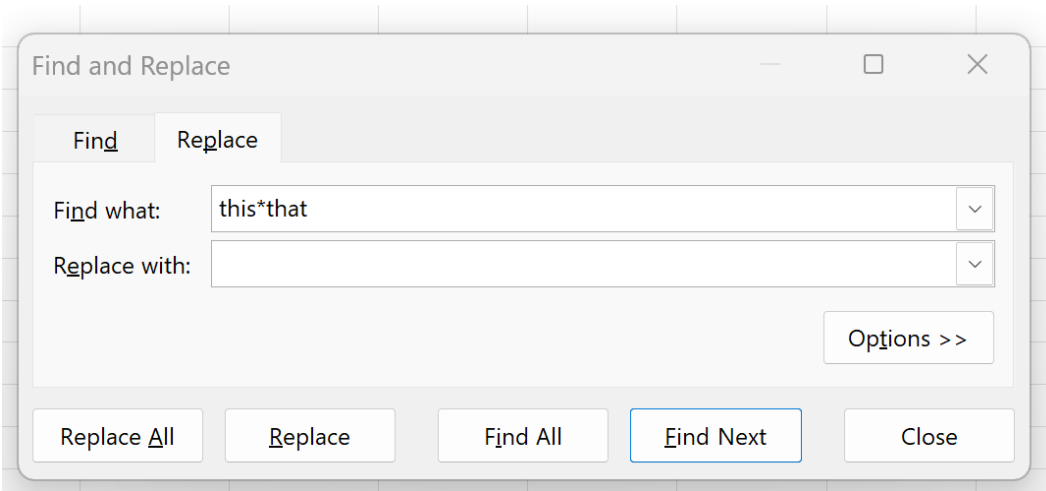
Replacing All Characters Before a Specific String: Here, the wildcard is positioned before a fixed target string. This configuration instructs [Excel](#) to find the target string and consider all preceding characters, regardless of their length or content, as part of the match to be replaced.

Replacing All Characters After a Specific String: Conversely, placing the wildcard after a fixed target string allows the user to remove the target string itself along with every subsequent character in the cell. This is useful for truncating text at a known point.

Visualizing these patterns is crucial for successful implementation. The following images demonstrate the syntax required in the [Find and Replace](#) dialog box for each scenario:

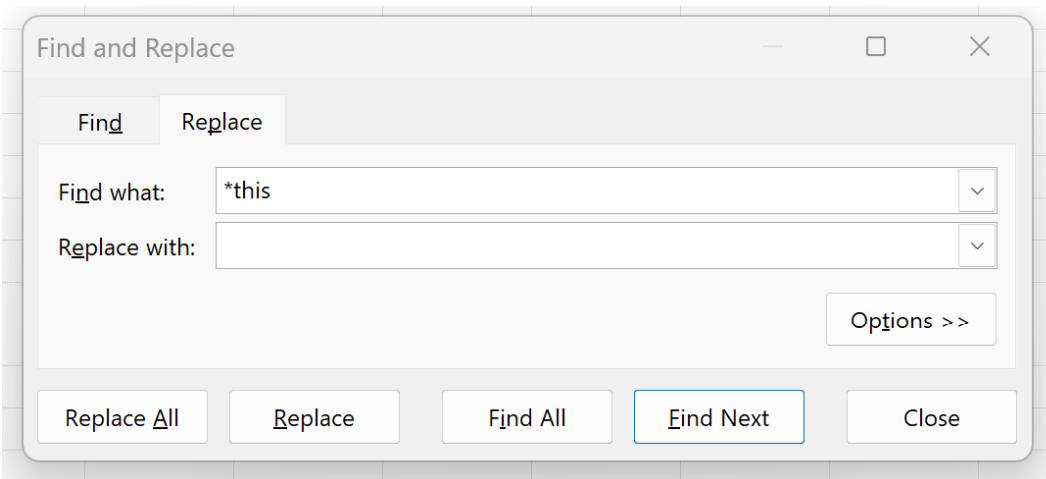
1. Replacing Content Between Specific Strings

This syntax, such as "this*that," targets the string "this," the string "that," and all characters **between** them.



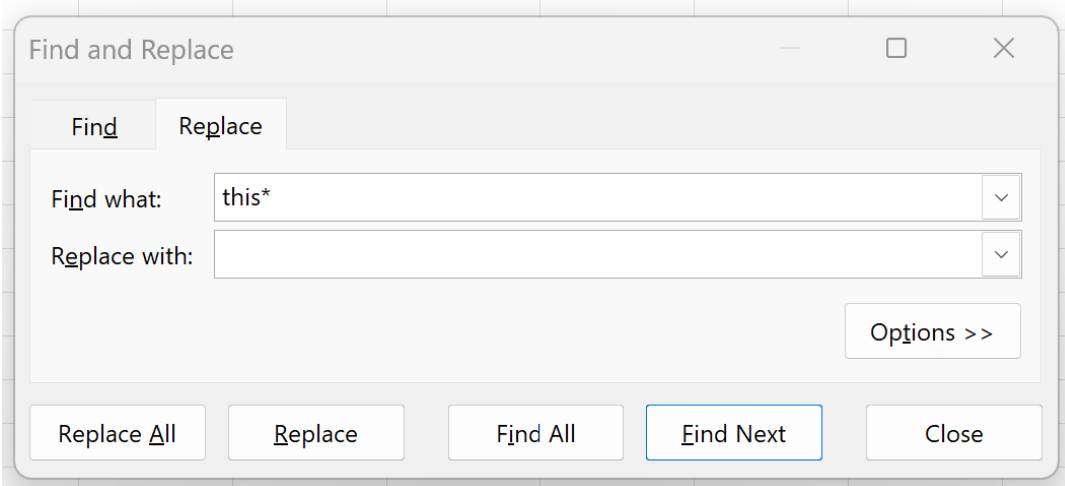
2. Replacing Content Before a Specific String

By placing the asterisk before a target string, such as "*this," we replace the string "this" along with all preceding characters in the cell.



3. Replacing Content After a Specific String

When the asterisk follows a target string, such as "this*," the replacement includes the string "this" and all characters that come **after** this string in the cell.



The following practical examples illustrate how to apply these wildcard replacement techniques to real-world data scenarios.

Example 1: Removing All Characters Between Two Specific Characters

Consider a scenario where we are managing a [dataset](#) containing information on basketball players. The positions are currently listed with extra descriptive text enclosed in parentheses, which we need to remove to standardize the data format.

Below is the initial [dataset](#) showing player names and their positions:

	A	B	C	D
1	Name	Position		
2	Andy	(Starter) Guard		
3	Bob	(Backup) Guard		
4	Chad	(Backup) Guard		
5	Doug	(Starter) Forward		
6	Eric	(Backup) Forward		
7	Frank	(Backup) Forward		
8	Greg	(Starter) Center		
9	Henry	(Backup) Center		
10	Isaac	(Backup) Center		
11				
12				
13				
14				
15				

Our objective is to completely eliminate the parentheses and all variable characters enclosed within them across the entire **Position** column. This task requires utilizing the wildcard to span the content between the fixed characters (the opening and closing parentheses).

To execute this operation effectively, follow these steps:

Highlight the relevant [cell range](#), which in this case is **B2:B13**.

Open the [Find and Replace](#) dialog box (using **Ctrl + H** or navigating via the Home tab).

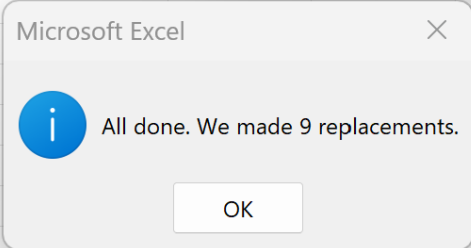
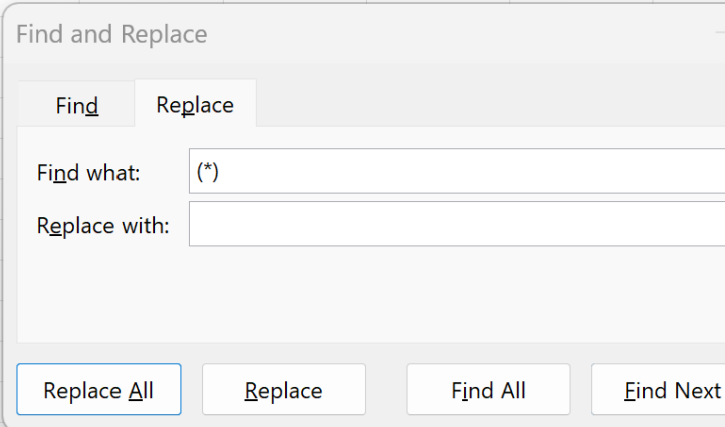
In the **Find what** field, enter the pattern (*). The asterisk acts as the [wildcard character](#) that captures everything between the parentheses.

Ensure the **Replace with** box is left completely empty, as our goal is deletion.

Click **Replace All** to apply the changes across the highlighted range.

Upon execution, the parentheses and all characters between them are successfully removed, resulting in a clean, standardized data column:

	A	B	C	D	E	F	G	H
1	Name	Position						
2	Andy	Guard						
3	Bob	Guard						
4	Chad	Guard						
5	Doug	Forward						
6	Eric	Forward						
7	Frank	Forward						
8	Greg	Center						
9	Henry	Center						
10	Isaac	Center						
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								



Example 2: Replacing All Characters Before Specific Characters

In this second example, we focus on isolating the end portion of a text string by removing everything that precedes a specific character sequence. We will use the same structure of player data, but the goal is to target a specific substring within the position names.

Here is the initial [dataset](#) we will be modifying:

	A	B	C	D	E
1	Name	Position			
2	Andy	Guard			
3	Bob	Guard			
4	Chad	Guard			
5	Doug	Forward			
6	Eric	Forward			
7	Frank	Forward			
8	Greg	Center			
9	Henry	Center			
10	Isaac	Center			
11					
12					
13					
14					
15					

Suppose we wish to remove the string "ar" and every character that appears before it in each cell within the **Position** column. This technique is useful for extracting suffixes or the latter part of descriptive fields.

To achieve this precise removal, follow the procedure outlined below:

Select the target [cell range](#), **B2:B13**.

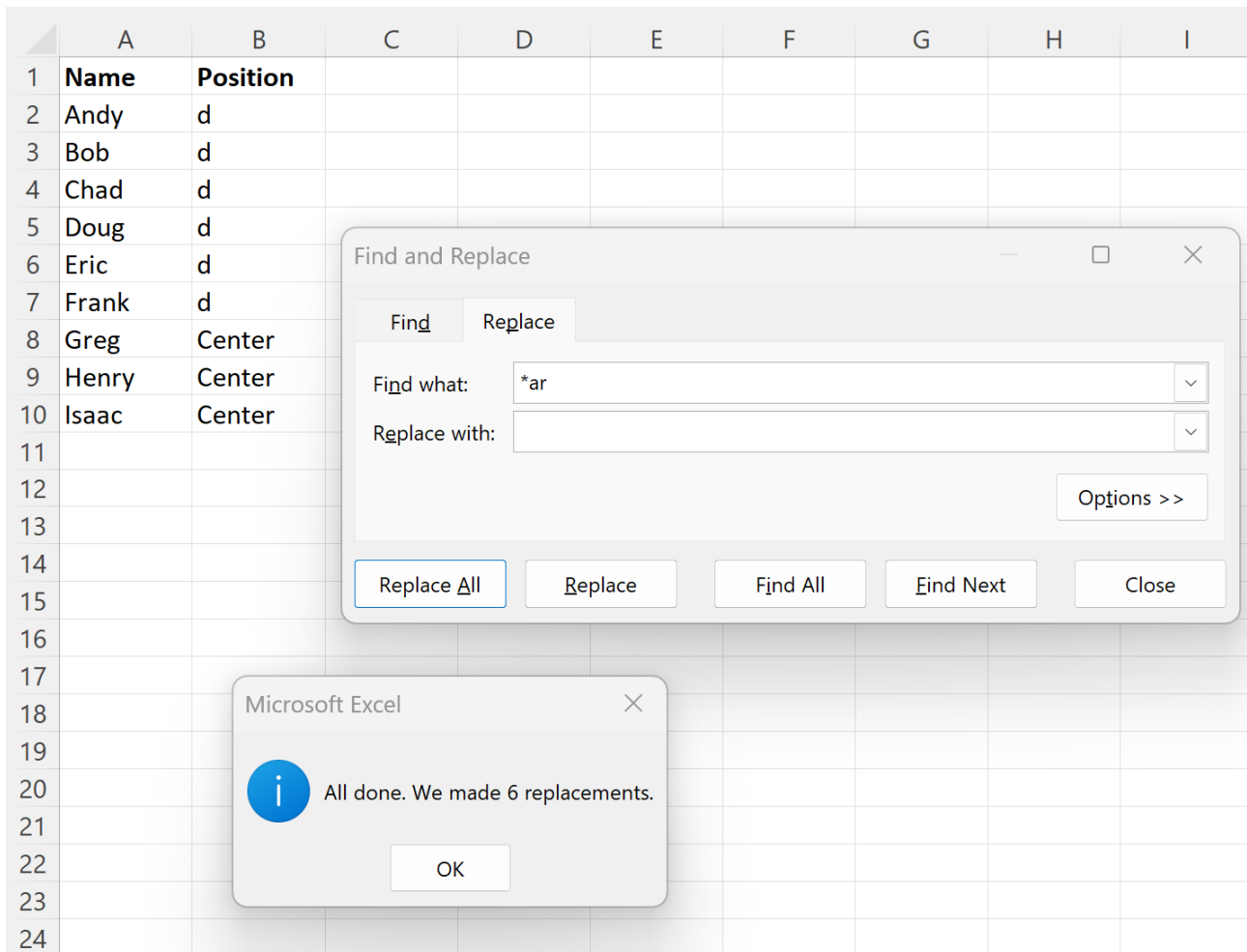
Use the keyboard shortcut **Ctrl + H** to open the [Find and Replace](#) window.

In the **Find what** box, input the pattern ***ar**. This tells [Excel](#) to find "ar" and include all preceding characters (represented by the asterisk) in the match.

Leave the **Replace with** field blank.

Click **Replace All**.

The result shows that "ar" and all preceding characters have been removed, leaving only the characters that followed that sequence. For instance, "Power Forward" becomes "d".



The screenshot shows an Excel spreadsheet with columns A through I and rows 1 through 24. The data is as follows:

	A	B	C	D	E	F	G	H	I
1	Name	Position							
2	Andy	d							
3	Bob	d							
4	Chad	d							
5	Doug	d							
6	Eric	d							
7	Frank	d							
8	Greg	Center							
9	Henry	Center							
10	Isaac	Center							
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									

The "Find and Replace" dialog box is open, showing the "Replace" tab. The "Find what:" field contains the wildcard pattern "*ar". The "Replace with:" field is empty. The "Options >>" button is visible. Below the dialog box, a "Microsoft Excel" information message box displays the text "All done. We made 6 replacements." with an "OK" button.

It is important to **Note** that any cell containing "Center" remained entirely unchanged. This is because the *ar pattern requires the string "ar" to be present within the cell for the replacement operation to be executed. If the target string is absent, the wildcard match fails, and the cell remains as it was.

Example 3: Replacing All Characters After Specific Characters

The final method demonstrates how to use the [wildcard character](#) to truncate strings by removing the target characters and everything that follows them. We will use the same positional [dataset](#) once more.

The [dataset](#) remains as follows:

	A	B	C	D	E
1	Name	Position			
2	Andy	Guard			
3	Bob	Guard			
4	Chad	Guard			
5	Doug	Forward			
6	Eric	Forward			
7	Frank	Forward			
8	Greg	Center			
9	Henry	Center			
10	Isaac	Center			
11					
12					
13					
14					
15					

Our goal now is to identify the string "ar" and replace it, along with every character that appears **after** it, across the cells in the **Position** column. This action effectively shortens the strings, keeping only the initial part of the position name up to the point just before "ar".

To perform this reverse truncation, utilize the following steps:

Select the necessary [cell range](#), **B2:B13**.

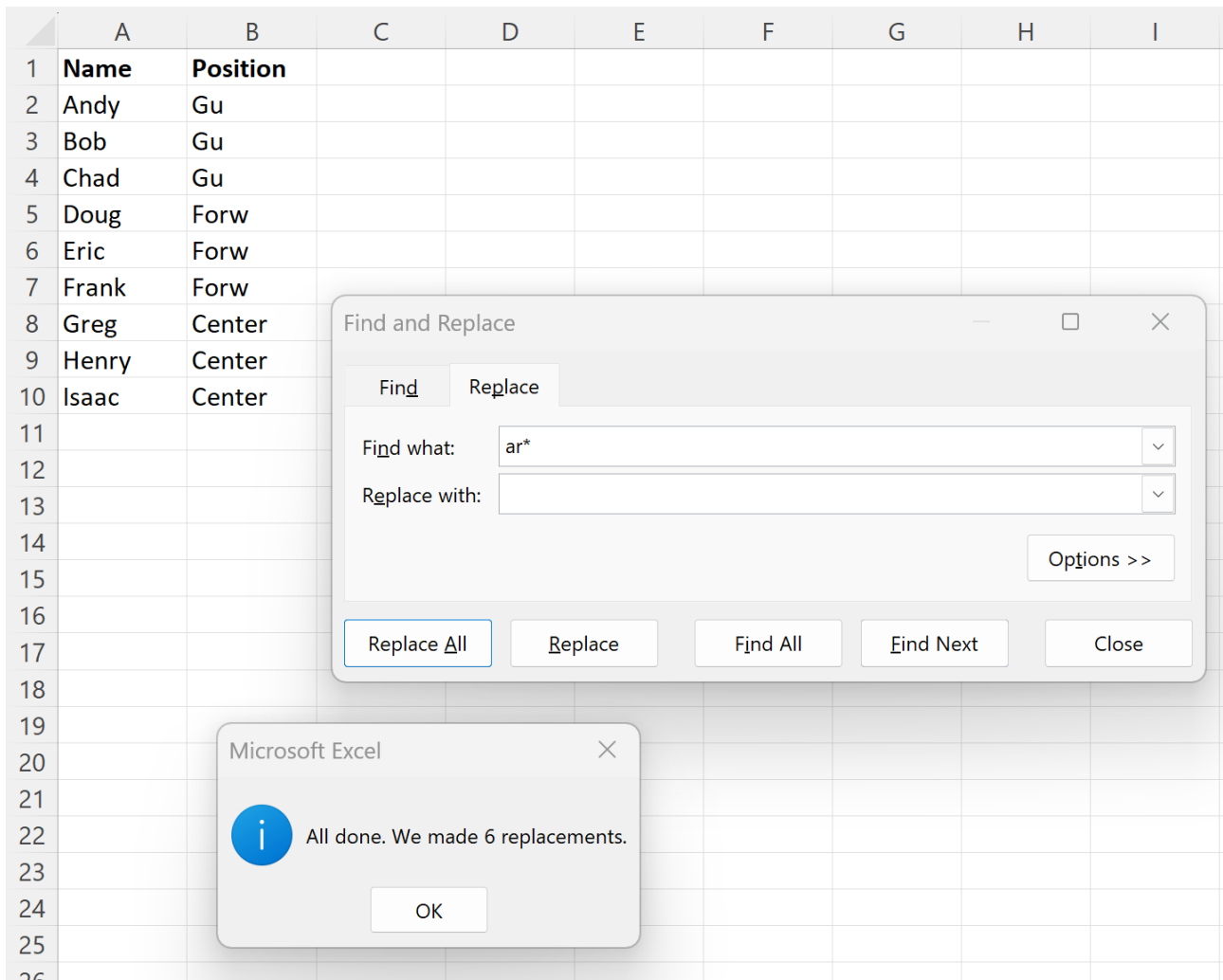
Access the [Find and Replace](#) utility (**Ctrl + H**).

In the **Find what** box, enter the pattern **ar***. This setup captures the string "ar" and all subsequent characters represented by the asterisk.

Ensure the **Replace with** field is left empty.

Click **Replace All**.

After clicking **Replace All**, the "ar" string and all trailing characters are removed from the cells where the pattern was found, resulting in strings that have been truncated at that specific point:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	Name	Position							
2	Andy	Gu							
3	Bob	Gu							
4	Chad	Gu							
5	Doug	Forw							
6	Eric	Forw							
7	Frank	Forw							
8	Greg	Center							
9	Henry	Center							
10	Isaac	Center							

The 'Find and Replace' dialog box is open, showing the 'Find' tab. The 'Find what' field contains 'ar*' and the 'Replace with' field is empty. The 'Options >>' button is visible. Below the dialog box, a 'Microsoft Excel' message box displays: 'All done. We made 6 replacements.' with an 'OK' button.

As observed in the previous example, **Note** that cells containing the position "Center" were unaffected by the replacement operation, as they did not contain the target string "ar". This confirms that wildcard searches are highly conditional upon the presence of the fixed characters defined in the pattern.

Additional Resources for Excel Proficiency

Mastering the use of [wildcard characters](#) within [Excel](#) is a fundamental step toward advanced data manipulation. To further enhance your proficiency in data processing and analysis, we recommend exploring tutorials on related operations:

How to use advanced filtering techniques in spreadsheets.

Methods for combining text from multiple cells using formulas.

Guides on cleaning and standardizing data formats efficiently.