How to Extract Text Between Commas in Excel: A Simple Guide

Authored by

stats writer

February 23, 2026

Excel: Extract Text Between Two Commas

In the contemporary landscape of **data management**, the ability to isolate specific segments of information within a larger text string is a fundamental skill for any professional using **Microsoft Excel**. Often, datasets imported from external sources like **CSV** files or database exports contain concatenated values where individual attributes are separated by a **delimiter**, most commonly a comma. Extracting a specific element located between these markers--such as a state name situated between a city and a country--requires a strategic approach to formula construction to ensure **data integrity** and precision across thousands of rows in a **spreadsheet**.

Historically, performing this task required complex nested formulas involving the **FIND**, **LEN**, and **MID** functions, which could be difficult to audit and prone to errors if the text length varied significantly. However, with the introduction of dynamic array functions in **Microsoft 365**, the process has been dramatically simplified. By utilizing the **TEXTBEFORE** and **TEXTAFTER** functions, users can now create more readable and robust logic to parse strings. This guide provides a comprehensive overview of how to leverage these modern tools to efficiently extract text between two commas, ensuring your **data analysis** workflows remain both efficient and scalable.

By mastering these techniques, you can transform messy, unorganized strings into structured data that is ready for **pivot tables**, **visualization**, or further computational processing. The logic remains consistent whether you are dealing with a handful of records or an enterprise-level dataset. Understanding the underlying mechanics of how **Microsoft Excel** processes text will empower you to handle even the most irregular data formats with confidence and technical rigor.

The Foundational Mechanics of TEXTBEFORE and TEXTAFTER

To effectively extract text between delimiters, one must first understand the individual utility of the **TEXTBEFORE** and **TEXTAFTER** functions. The **TEXTAFTER** function is designed to return all the text that occurs after a specified character or string. Its primary strength lies in its ability to ignore the leading portion of a string, which is essential when your target data is buried deep within a cell. In our specific context, **TEXTAFTER** serves as the first filter, stripping away the initial segment of the string up to the first **delimiter**.

Complementing this is the **TEXTBEFORE** function, which operates in the opposite direction by capturing only the text that precedes a chosen character. When these two functions are nested, they create a "sandwich" effect that isolates the middle content. The **TEXTBEFORE** function acts as the final boundary, ensuring that once the leading text is removed, the trailing text after the second **delimiter** is also discarded. This dual-layered approach is highly effective for parsing standardized strings such as addresses, product codes, or full names.

Furthermore, both functions offer optional arguments that allow for advanced customization. For instance, you can specify which instance of a **delimiter** to target if a cell contains more than two commas. You can also define case sensitivity or provide a custom value to return if the delimiter is not found. This level of control makes these tools significantly more powerful than the legacy **Text to Columns** feature, as formulas are dynamic and update automatically whenever the source data changes in the **spreadsheet**.

Constructing the Nested Formula Logic

The core strategy for isolating a middle value involves nesting the **TEXTAFTER** function inside the **TEXTBEFORE** function. This creates a sequential operation where **Microsoft Excel** first identifies the portion of the string following the first comma and then identifies the portion of that resulting string that precedes the next comma. This mathematical logic ensures that regardless of the length of the city or the country name, the middle element--the state--is consistently captured.

You can use the following syntax to do so:

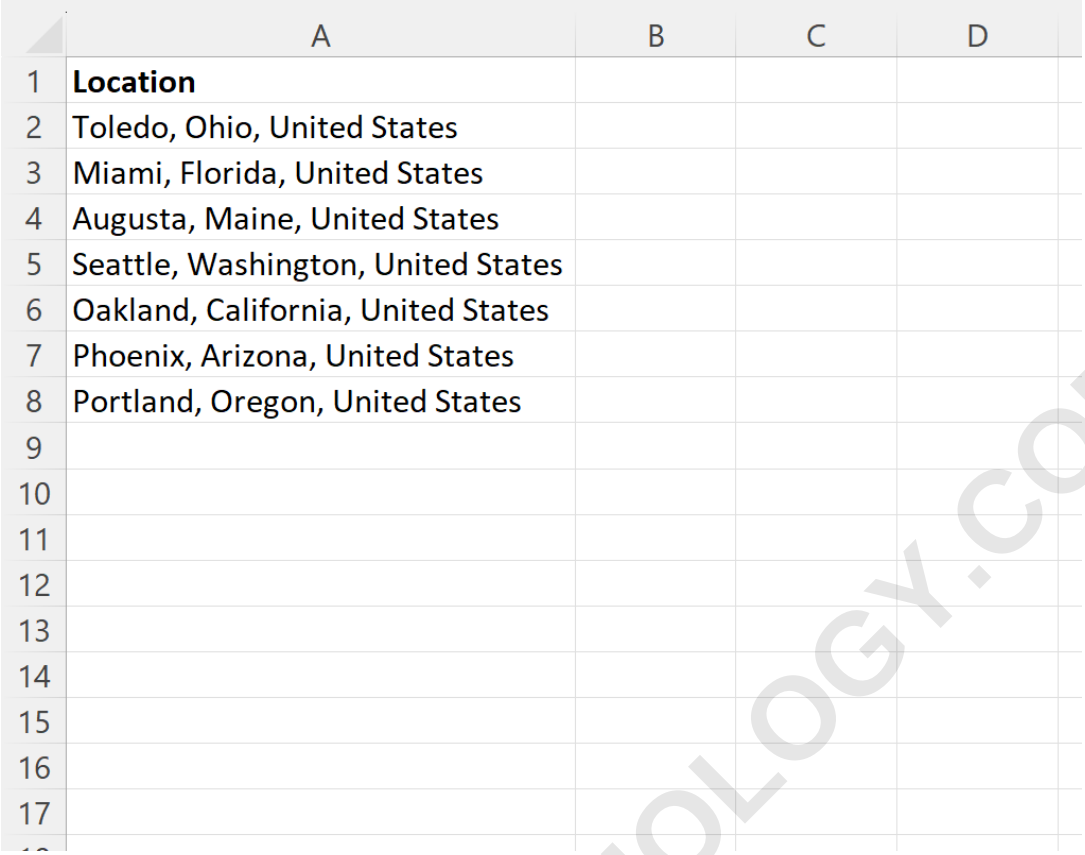
=TEXTBEFORE(TEXTAFTER(A2, ","), ",")

This particular example extracts all of the text between the two commas in cell **A2**. The inner function, `TEXTAFTER(A2, ",")`, looks at the content of cell A2 and finds the first occurrence of a comma. It then outputs everything to the right of that comma. This intermediate result is then passed to the outer function, `TEXTBEFORE(..., ",")`, which looks for the first comma in that new string and outputs everything to the left of it, effectively isolating the text that was originally between the two markers.

It is important to note that the order of operations is critical. If you were to reverse these functions, the logic would fail because the first comma would be removed before the second could be properly identified as a boundary. By nesting the functions in this specific way, you are creating a logical pipeline that refines the data in stages. This modular approach to **formula construction** is a hallmark of advanced **data analysis** within **Microsoft 365** environments, allowing for complex transformations with relatively simple syntax.

Practical Application: Extracting State Data

The following example shows how to use this syntax in practice. Suppose we have the following list of locations in Excel, where each entry consists of a city, a state, and a country, all separated by commas. This is a very common format for geographic datasets, yet it presents a challenge when you only need to perform a **data analysis** on the state level without the clutter of the other identifiers.



	A	B	C	D
1	**Location**			
2	Toledo, Ohio, United States			
3	Miami, Florida, United States			
4	Augusta, Maine, United States			
5	Seattle, Washington, United States			
6	Oakland, California, United States			
7	Phoenix, Arizona, United States			
8	Portland, Oregon, United States			
9				
10				
11				
12				
13				
14				
15				
16				
17				

Now suppose we would like to extract the state name between the two commas in each cell. This task is essential for creating regional reports or sorting your data by administrative divisions. Without an automated formula, a user would be forced to manually type these names, which is not only time-consuming but also introduces a high risk of typographical errors that could compromise the entire **spreadsheet**.

For example, we would like to extract:

Ohio from the first cell.
Florida from the second cell.
Maine from the third cell.

And so on. Each of these values is currently trapped between two commas. By applying our nested formula, we can automate this extraction across the entire column. This ensures that even if you add thousands of new rows to your dataset in the future, the extraction process remains consistent and requires zero manual intervention.

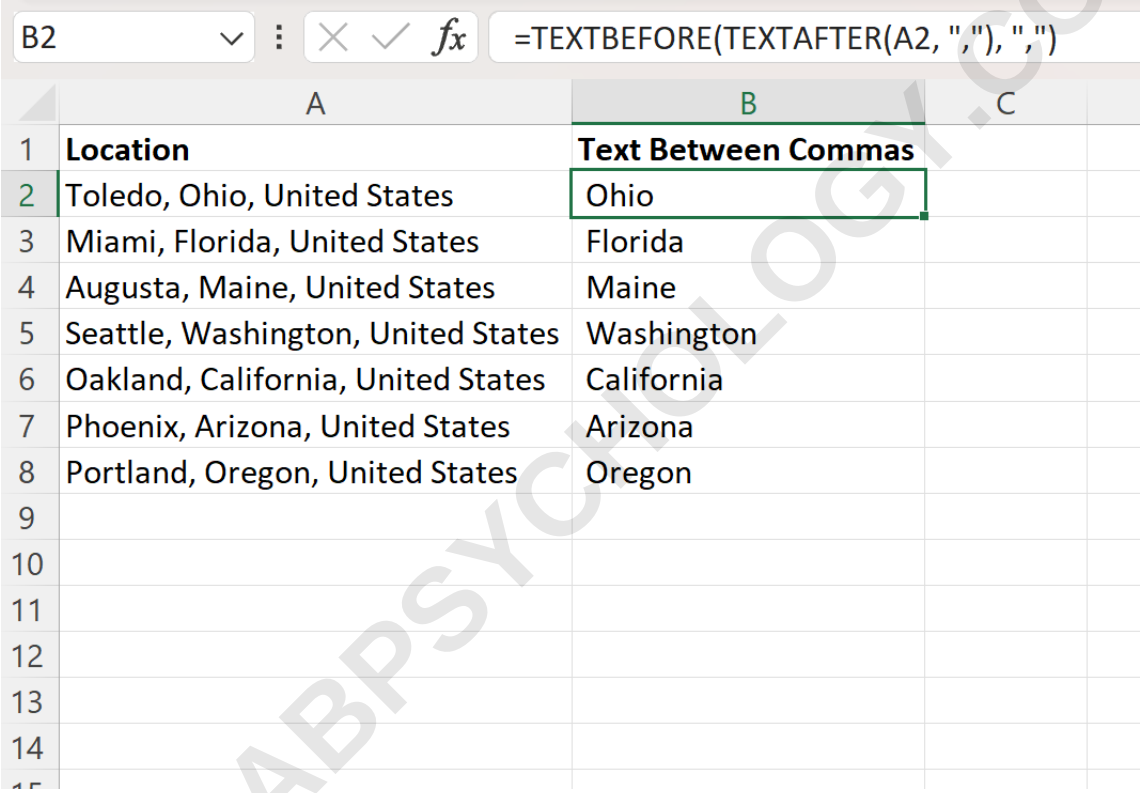
Step-by-Step Execution in the Spreadsheet

To implement this solution, begin by identifying the target cell containing your concatenated string.

We can type the following formula into cell **B2** to extract the text between the commas in cell **A2**:

=TEXTBEFORE(TEXTAFTER(A2, ","), ",")

Once the formula is entered, **Microsoft Excel** will immediately process the request and display the extracted text. To apply this logic to the rest of your dataset, we can then click and drag this formula down to each remaining cell in column B. This action utilizes Excel's **relative referencing** capability, automatically updating the cell coordinates (e.g., A3, A4, A5) as the formula is copied downward, ensuring that each row is processed individually and accurately.



As shown in the updated illustration, Column B now contains the text between the commas for each corresponding cell in column A. This transformation allows you to see the state names clearly separated from the rest of the geographic data. The result is a clean, structured dataset that can be used for more sophisticated **data analysis** tasks, such as counting occurrences of each state or mapping the data using Excel's built-in 3D Maps feature.

In-Depth Breakdown of Formula Logic

To truly master **Microsoft Excel**, one must understand exactly why a formula works the way it does. Recall the formula that we used to extract the text between the commas in cell **A2**: `=TEXTBEFORE(TEXTAFTER(A2, ","), ",")`. The logic follows a linear path of reduction. Initially,

the **TEXTAFTER** function is triggered. It scans the source text for the first **delimiter**. Once found, it discards everything to the left, including the comma itself, resulting in a string that starts with the targeted information.

In our example, this first step returns **Ohio, United States**. Note that the leading city name and the first comma have been successfully removed. However, the string still contains the trailing country name and the second comma, which are not desired. This is where the **TEXTBEFORE** function becomes necessary. It takes the output of the first function as its input and performs its own search for the next comma.

When **TEXTBEFORE** identifies the comma following the word "Ohio," it isolates and returns everything to the left of that point. This effectively clips off the "United States" portion and the remaining **delimiter**. This returns **Ohio** as the final value. This elegant combination of functions eliminates the need for calculating character positions manually, making your workbook more resilient to changes in data formatting.

Improving Accuracy with the TRIM Function

A common issue when working with comma-separated data is the presence of unnecessary spaces. Often, a comma is followed by a space for readability (e.g., "City, State, Country"). If you use the extraction formula as is, the resulting text may contain a **leading space**. While this might not be immediately visible in the cell, it can cause significant issues when using **VLOOKUP**, **MATCH**, or filters, as " Ohio" is not the same as "Ohio" in a computational context.

To resolve this, you can wrap the **TRIM** function around your formula. The **TRIM** function is specifically designed to remove all leading and trailing spaces from a text string, while leaving single spaces between words intact. By incorporating this into your workflow, you ensure that the extracted data is clean and ready for immediate use in other functions or external software applications.

The modified formula would look like this: `=TRIM(TEXTBEFORE(TEXTAFTER(A2, ","), ","))`. Using this enhanced version is considered a **best practice** in professional **spreadsheet** design. It prevents "ghost" errors that are difficult to track down and ensures that your data remains standardized. Whether you are preparing a mailing list or a financial report, maintaining high data quality through functions like **TRIM** is essential for reliable results.

Alternative Methods for Legacy Excel Versions

While **TEXTBEFORE** and **TEXTAFTER** are available in **Microsoft 365** and Excel 2021, users on older versions must rely on a different set of tools. The traditional method involves using the **MID** function combined with **FIND**. This approach requires you to find the position of the first comma,

add 1 (to skip the comma itself), and then calculate the number of characters between the first and second comma. Although more verbose, it is a universal solution that works in almost any version of the software.

Another powerful alternative is **Power Query**, a data transformation engine built into modern versions of **Microsoft Excel**. In Power Query, you can use the "Split Column by Delimiter" feature. By selecting "Comma" as the delimiter and choosing to split into three columns, you can easily isolate the middle value without writing a single formula. This is particularly useful for very large datasets where formula performance might become an issue.

Lastly, for one-time tasks, **Flash Fill** is an incredibly intuitive tool. By typing the desired result for the first two or three rows, Excel can often recognize the pattern and fill the remaining cells automatically. However, unlike formulas, Flash Fill is not dynamic; if the source text changes, the extracted text will not update. Therefore, for ongoing projects, the formula-based approach using **TEXTBEFORE** and **TEXTAFTER** remains the most professional and reliable choice.

Conclusion and Advanced Considerations

Mastering the extraction of text between delimiters is a gateway to more advanced **data analysis** capabilities within **Microsoft Excel**. As you become comfortable with these functions, you can begin to tackle more complex scenarios, such as strings with varying numbers of commas or inconsistent spacing. The principles of nesting functions to create a logical sequence of operations are applicable across a wide range of Excel challenges, from financial modeling to scientific research.

Always remember to validate your data after performing an extraction. Check for unexpected results, such as cells that may only contain one comma or none at all. In such cases, the formula might return an error. You can handle these exceptions gracefully by wrapping your formula in an **IFERROR** function, providing a fallback value like "Check Data" or leaving the cell blank. This attention to detail is what separates basic users from true Excel experts.

The following tutorials explain how to perform other common tasks in Excel, helping you further expand your technical toolkit and improve your efficiency in any data-driven role: