

Excel: Split Text and Get First Item

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Excel: Split Text and Get First Item*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=93377>

Introduction to Advanced Text Manipulation in Excel

The ability to efficiently manipulate strings of text is fundamental for data cleaning and analysis within spreadsheet software. Modern versions of Excel have introduced powerful dynamic array functions that simplify complex tasks, such as parsing concatenated data. Specifically, the **TEXTSPLIT** function is invaluable for segmenting text strings based on a specified character or sequence--known as a delimiter. This function transforms a single cell value into an array of results, which is then dynamically spilled across adjacent cells.

While **TEXTSPLIT** is excellent for creating a full split, often we only require a specific component of the resulting array--such as the very first item. Achieving this specific extraction requires nesting **TEXTSPLIT** within another specialized function designed to select specific columns from an array. This combination allows for precise data extraction without the need for older, more complex formulas involving **FIND**, **LEFT**, and **MID** functions.

To retrieve only the first component resulting from the split operation, we combine **TEXTSPLIT** with the **CHOOSECOLS** function. The following section introduces the essential syntax required to execute this targeted data parsing efficiently within your spreadsheet workflows.

The Essential Syntax for Extracting the First Item

The goal of this operation is to isolate the data that appears before the first instance of a specified delimiter. By using the nested formula structure, we ensure that the entire array created by the splitting process is handled internally, and only the required column (the first one) is ultimately displayed in the cell where the formula is entered.

The general structure involves telling Excel to choose a specific column from the array that the splitting function returns. The syntax below is designed to split the text found in cell **A2**, using a space (" ") as the splitting criterion, and then extract the first column of the resulting array:

=CHOOSECOLS(TEXTSPLIT(A2, " "), 1)

In this specific instance, the formula first processes the string in cell **A2**. If **A2** contains a name like **Andy Bernard**, the **TEXTSPLIT** component internally generates two items: "Andy" (Column 1) and "Bernard" (Column 2). Subsequently, the **CHOOSECOLS** function immediately selects only the first column, resulting in the desired output: **Andy**. This powerful method is extremely useful when managing large datasets containing concatenated information, such as full names or product codes separated by specific characters.

Step-by-Step Example: Isolating First Names

To demonstrate the practical application of this function, consider a scenario where you have a dataset containing a column of full names. Your objective is to efficiently extract only the first name of each individual into a new, separate column. This often occurs during data normalization or preparation for database import.

Suppose we begin with the following column of names in our Excel spreadsheet, located in Column A:

	A	B	C	D	E
1	Name				
2	Andy Bernard				
3	Chad Douglas				
4	Eric Ferdinand				
5	Josh Mann				
6	Arnold Smith				
7	Jake Johnson				
8	Tyson Reed				
9	Brett Handzel				
10	Mike Scott				
11					
12					
13					
14					

Our primary requirement is to split these full names wherever a space occurs, and then, from the resulting split components, retrieve only the very first item--which corresponds to the first name. We will apply the combined **CHOOSECOLS** and **TEXTSPLIT** formula directly into cell **B2** to begin the extraction process for the first entry.

To split the text in cell **A2** and retrieve the first element, we input the formula precisely as follows into cell **B2**:

=CHOOSECOLS(TEXTSPLIT(A2, " "), 1)

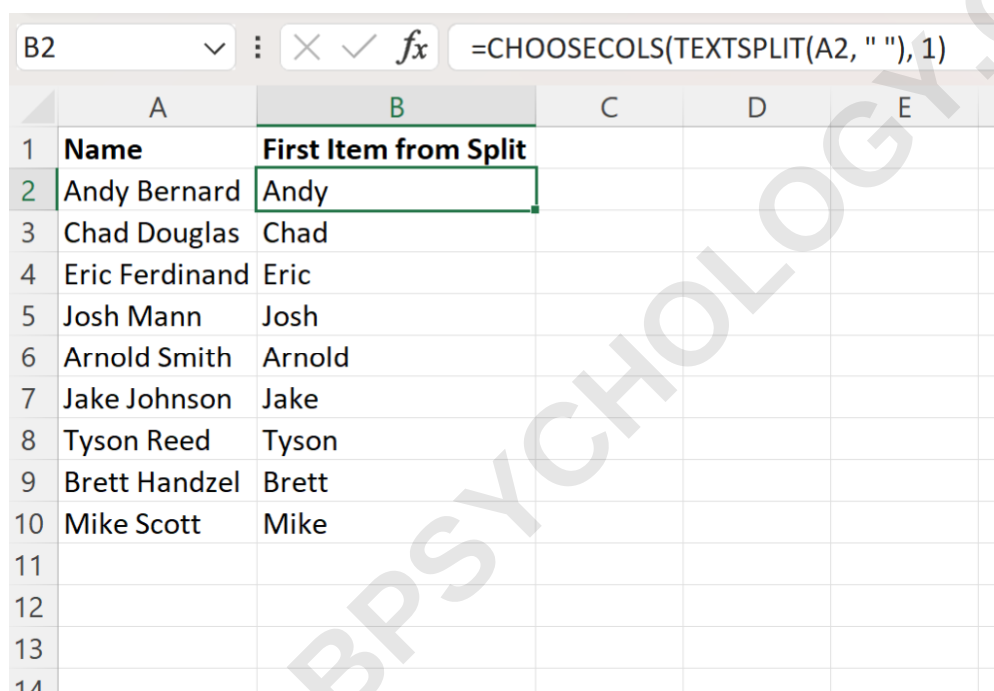
Executing the Formula and Reviewing Results

Once the formula is entered into cell **B2**, it immediately calculates the first name from the adjacent

cell. To apply this logic across the entire dataset, we simply utilize the fill handle functionality in [Excel](#). We click and drag this formula down to each remaining cell in Column B corresponding to the data rows in Column A. The dynamic nature of the relative cell reference (A2) ensures that the formula correctly adjusts to reference A3, A4, and so on, for each subsequent row.

Observing the results, we can confirm the successful implementation of the extraction logic. The formula effectively splits the names in Column A based on the space [delimiter](#) and returns only the first item from that internal split array.

The following visual representation confirms the outcome after applying the formula to all relevant cells:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E
1	Name	First Item from Split			
2	Andy Bernard	Andy			
3	Chad Douglas	Chad			
4	Eric Ferdinand	Eric			
5	Josh Mann	Josh			
6	Arnold Smith	Arnold			
7	Jake Johnson	Jake			
8	Tyson Reed	Tyson			
9	Brett Handzel	Brett			
10	Mike Scott	Mike			
11					
12					
13					
14					

As intended, for this practical example, the output in Column B exclusively consists of the first name associated with each entry in Column A. This technique provides a clean and robust solution for separating data fields that are combined using a consistent [delimiter](#).

A Detailed Breakdown of the Formula Logic

Understanding the internal mechanism of this nested formula is essential for mastering its use and adapting it to different data splitting needs. Let's revisit the core formula used to achieve our extraction:

=CHOOSECOLS(TEXTSPLIT(A2, " "), 1)

The execution of this formula follows a two-step process, working from the inside out:

The inner function, **TEXTSPLIT(A2, " ")**, executes first. It takes the text string in cell **A2** and breaks it apart wherever it encounters the specified delimiter, which is a space (" "). This results in a temporary, internal array (a spill range that is not immediately visible). For instance, if **A2** contains "Andy Bernard," the array produced is {"Andy", "Bernard"}.

The outer function, **CHOOSECOLS(array, 1)**, then processes this internal array. The second argument, **1**, instructs the function to select only the first column (or element) from the array generated by the split.

The result of this sequence is the isolated first item, **Andy**, which is returned directly to the formula cell. This mechanism ensures that only the required element is displayed, preventing the generation of a sprawling spill range and keeping the spreadsheet tidy and efficient.

Modifying the Formula to Extract Subsequent Items

One of the major benefits of using the **CHOOSECOLS** function is the ease with which you can adapt the extraction to target any specific item in the split sequence. If your data structure requires isolating the second or third item instead of the first, a simple modification to the function's column index argument is all that is necessary.

For example, if you wish to return the second item--the last name in our current scenario--you simply replace the number **1** with the number **2** in the formula's final argument. This tells **CHOOSECOLS** to select the second column from the array produced by the split.

The modified formula to extract the second element looks like this:

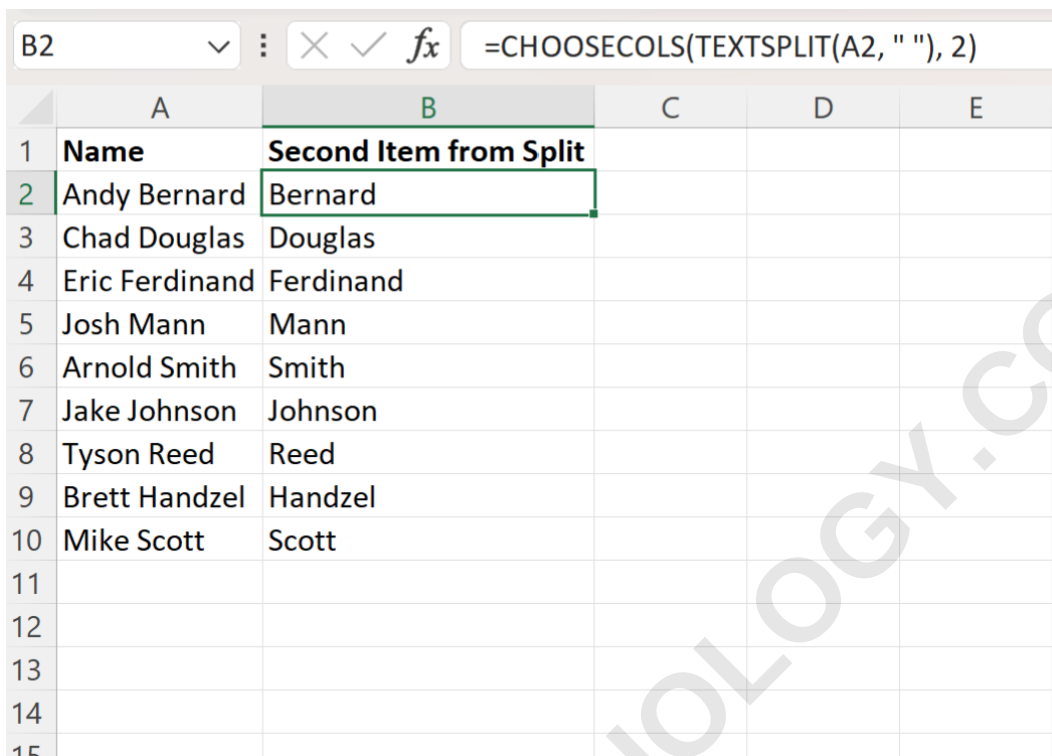
=CHOOSECOLS(TEXTSPLIT(A2, " "), 2)

Implementing this change allows analysts to quickly flip between extracting first names, last names, middle initials, or any other segment of text that is separated by the consistent delimiter. This flexibility highlights why dynamic array formulas are the preferred method for modern text manipulation in Excel.

Visualizing Extraction of the Second Item

To confirm the effectiveness of changing the column index, observe the result when the modified formula is applied to our original dataset. When the index is set to **2**, the formula successfully isolates the last name (or second item) from the input string in Column A.

The following screenshot illustrates the output in Column B when using the formula targeting the second column index:



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E
1	Name	Second Item from Split			
2	Andy Bernard	Bernard			
3	Chad Douglas	Douglas			
4	Eric Ferdinand	Ferdinand			
5	Josh Mann	Mann			
6	Arnold Smith	Smith			
7	Jake Johnson	Johnson			
8	Tyson Reed	Reed			
9	Brett Handzel	Handzel			
10	Mike Scott	Scott			
11					
12					
13					
14					
15					

The formula bar at the top shows the formula: `=CHOOSECOLS(TEXTSPLIT(A2, " "), 2)`.

As clearly demonstrated, this time the formula accurately returns the second item from the split text, confirming the versatile control offered by combining **TEXTSPLIT** and **CHOOSECOLS** for specific text extraction requirements.