

# Learn How to Combine Text from Two Excel Columns Using Formulas

Authored by  
**Mohammed looti**

November 10, 2025

## RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Combine Text from Two Excel Columns Using Formulas*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15473>

## Introduction: Essential Techniques for Data Combination in Excel

Microsoft [Excel](#) is a powerful tool for data management, but often raw datasets require manipulation before analysis can begin. One of the most common requirements is the need to combine text or numerical values stored in separate columns into a single, cohesive cell. This process, known as [concatenation](#), is essential when merging first and last names, creating unique identifiers, or simply formatting data for presentation. While various methods exist for joining data, adding a crucial space or separator between the values ensures readability and usability.

This article details three primary, highly effective formulas available in [Excel](#) for achieving text combination with a space delimiter: the traditional [Ampersand \(&\) operator](#), the modern [CONCAT function](#), and the versatile [TEXTJOIN function](#). Understanding the strengths and weaknesses of each method allows users to select the most efficient solution based on the complexity and volume of the data task at hand. We will demonstrate each technique using a practical example involving basketball player data, combining first names and last names into a full name column.

You can use the following formulas to combine values from two columns in [Excel](#) with a space in between them:

### Three Core Formulas for Concatenation with a Space Delimiter

Before diving into the practical examples, it is helpful to review the foundational syntax for each of the three methods. All three approaches share the goal of merging the content of cells **A2** and **B2**, inserting a literal space character (" ") as the separator. These formulas represent increasing levels of sophistication and flexibility, catering to different versions of [Excel](#) and varying data structures.

The first and most universal method relies on the concatenation operator, while the function-based approaches offer cleaner syntax, particularly for combining ranges of data. Mastering these formulas is fundamental to efficient data preparation within any spreadsheet environment.

#### Formula 1: Use & Symbol (Concatenation Operator)

```
=A2&" "&B2
```

#### Formula 2: Use CONCAT Function (Modern Approach)

```
=CONCAT(A2, " ",B2)
```

#### Formula 3: Use TEXTJOIN Function (Advanced Range Handling)

```
=TEXTJOIN(" ", TRUE, A2:B2)
```

Each of these formulas combines the values from cells **A2** and **B2** into a single cell with a space in between the values. The following examples illustrate how to implement each formula in practice using a dataset containing information about various basketball players, where Column A holds the first name and Column B holds the last name.

	A	B	C	D	E
1	<b>Role</b>	<b>Position</b>			
2	Starting	Point Guard			
3	Backup	Point Guard			
4	Starting	Shooting Guard			
5	Backup	Shooting Guard			
6	Starting	Small Forward			
7	Backup	Small Forward			
8	Starting	Power Forward			
9	Backup	Power Forward			
10	Starting	Center			
11	Backup	Center			
12					
13					
14					
15					
16					

## Method 1: Utilizing the Ampersand (&) Operator for Simple Concatenation

The [Ampersand \(&\) operator](#) is arguably the most recognized and backward-compatible method for joining strings in [Excel](#). It works by linking cell references or text strings together sequentially. To insert a space, or any custom [delimiter](#), you must explicitly include it as a quoted text string within the formula structure. This method offers granular control over the combination process, making it ideal for joining a fixed number of cells.

When constructing a formula using the ampersand, every element--whether it is a cell reference (like **A2**), a literal string (like " "), or another function's output--must be separated by the **& symbol**. This ensures that [Excel](#) processes the elements in the correct order, transforming disparate data points into a continuous, merged string. Although slightly verbose compared to newer functions, its simplicity and ubiquitous availability across all [Excel](#) versions make it a reliable choice for fundamental data combination tasks.

In our specific example, we aim to combine the first name found in cell **A2** with the last name in cell **B2**, guaranteeing a single space between them. We achieve this by linking the first cell, then the space delimiter, and finally the second cell. We can type the following formula into cell **C2** to initiate this combination:

**=A2&" "&B2**

Once the formula is entered into the first row of the output column (C2), the power of [Excel](#) autofill can be utilized. We can then click and drag the fill handle (the small square at the bottom right corner of the selected cell) down to each remaining cell in column C. This action automatically adjusts the [cell references](#) (e.g., A2 becomes A3, A4, and so on) for each subsequent row, efficiently applying the concatenation logic across the entire dataset.

The result of applying this method is a new column where the data from columns A and B is elegantly merged. Column C successfully combines the values from columns A and B into a single cell, separated by a space, providing a clean, presentable list of full names.

	A	B	C	D	E
1	<b>Role</b>	<b>Position</b>	<b>Role &amp; Position Combined</b>		
2	Starting	Point Guard	Starting Point Guard		
3	Backup	Point Guard	Backup Point Guard		
4	Starting	Shooting Guard	Starting Shooting Guard		
5	Backup	Shooting Guard	Backup Shooting Guard		
6	Starting	Small Forward	Starting Small Forward		
7	Backup	Small Forward	Backup Small Forward		
8	Starting	Power Forward	Starting Power Forward		
9	Backup	Power Forward	Backup Power Forward		
10	Starting	Center	Starting Center		
11	Backup	Center	Backup Center		
12					
13					
14					
15					
16					

## Method 2: Leveraging the CONCAT Function for Modern Combining

Introduced in [Excel](#) 2016 and succeeding the older `CONCATENATE` function, the [CONCAT](#)

[function](#) provides a streamlined way to perform [concatenation](#). Unlike the ampersand operator, which requires the & symbol between every single element, [CONCAT](#) accepts multiple text arguments or even entire ranges as input, separating them with commas. This significantly improves formula readability, especially when combining many different fields.

A key advantage of the [CONCAT function](#) is its ability to handle cell ranges dynamically. While the ampersand method demands explicit linking of every cell, [CONCAT](#) can process a contiguous block of cells, though it still requires the user to insert the [delimiter](#) (the space " ") as a separate argument if it needs to appear between every item. For combining just two cells with a space, the syntax is clean and efficient, acting as a functional alternative to the Ampersand Operator.

To implement this method, we input the following formula into cell **C2**. Notice how the structure is slightly cleaner than the ampersand approach, grouping all elements within the function parentheses, separated by commas. The space character, enclosed in quotation marks, is passed as the second argument, ensuring separation between the cell contents:

```
=CONCAT(A2, " ",B2)
```

After entering the formula in cell **C2**, the process of applying it to the entire column remains consistent with other [Excel](#) formulas. We click and drag this formula down to the remaining cells in column C. This action replicates the formula, automatically updating the row number for the [cell references](#) (A2 and B2), ensuring every player's first and last name is correctly merged into the corresponding row in Column C.

As demonstrated by the resulting table, Column C effectively combines the values from columns A and B. This method is highly recommended for users operating on modern versions of [Excel](#) due to its improved syntax and support for ranges.

	A	B	C	D	E
1	<b>Role</b>	<b>Position</b>	<b>Role &amp; Position Combined</b>		
2	Starting	Point Guard	Starting Point Guard		
3	Backup	Point Guard	Backup Point Guard		
4	Starting	Shooting Guard	Starting Shooting Guard		
5	Backup	Shooting Guard	Backup Shooting Guard		
6	Starting	Small Forward	Starting Small Forward		
7	Backup	Small Forward	Backup Small Forward		
8	Starting	Power Forward	Starting Power Forward		
9	Backup	Power Forward	Backup Power Forward		
10	Starting	Center	Starting Center		
11	Backup	Center	Backup Center		
12					
13					
14					
15					
16					

**Note:** You can find the complete documentation for the [CONCAT function](#) in [Excel's](#) official support resources, which detail its behavior when handling mixed data types and ranges.

### Method 3: Mastering the TEXTJOIN Function for Advanced Delimitation

The [TEXTJOIN function](#), available in Excel 2019 and newer, represents the pinnacle of text combination tools. Its primary advantage is the ability to specify a single [delimiter](#) that automatically separates all items within a specified range, eliminating the need to repeatedly insert the delimiter manually as required by the Ampersand or [CONCAT function](#). This is exceptionally efficient when combining data from three or more columns.

The syntax of [TEXTJOIN](#) is structured around three core arguments: 1) the [delimiter](#) (the character or string used to separate the combined text, e.g., " "), 2) a boolean value (TRUE or FALSE) indicating whether to ignore empty cells, and 3) the text arguments or ranges to be joined. By setting the second argument to **TRUE**, empty cells are automatically skipped, preventing unwanted extra spaces or delimiters in the final output--a common challenge when dealing with incomplete datasets.

For our scenario, combining first and last names, the [TEXTJOIN function](#) provides the most compact and elegant solution. We enter the formula into cell **C2**, specifying the space character ("

) as the delimiter, setting the ignore\_empty argument to TRUE (though less critical for just two cells, it is good practice), and defining the range **A2:B2** as the text to be joined:

**=TEXTJOIN(" ", TRUE, A2:B2)**

Once the formula is correctly input into **C2**, we proceed with the standard procedure: clicking and dragging the formula down through the remaining cells in column C. This action populates the column, utilizing the defined space delimiter to separate the values sourced from the specified range in columns A and B for every corresponding row. This dramatically simplifies the process when dealing with wide tables or complex [concatenation](#) requirements.

The final output, shown in the image below, confirms that Column C successfully combines the first and last names from columns A and B into a single, space-separated string. The robustness of [TEXTJOIN function](#) ensures a clean result, even if some rows were missing a first or last name.

	A	B	C	D	E
1	<b>Role</b>	<b>Position</b>	<b>Role &amp; Position Combined</b>		
2	Starting	Point Guard	Starting Point Guard		
3	Backup	Point Guard	Backup Point Guard		
4	Starting	Shooting Guard	Starting Shooting Guard		
5	Backup	Shooting Guard	Backup Shooting Guard		
6	Starting	Small Forward	Starting Small Forward		
7	Backup	Small Forward	Backup Small Forward		
8	Starting	Power Forward	Starting Power Forward		
9	Backup	Power Forward	Backup Power Forward		
10	Starting	Center	Starting Center		
11	Backup	Center	Backup Center		
12					
13					
14					
15					
16					

**Note:** The [TEXTJOIN function](#) is highly versatile. It can utilize complex delimiters (like hyphens, commas, or entire phrases) and is particularly efficient for consolidating data from a large range of cells where manual insertion of delimiters would be tedious.

## Practical Application: Choosing the Right Concatenation Method

While all three formulas--the [Ampersand \(& operator\)](#), [CONCAT](#), and [TEXTJOIN](#)--achieve the goal of combining two cells with a space, the choice of method should be guided by specific project requirements, the version of [Excel](#) being used, and the desire for formula simplicity or robustness.

The **Ampersand Operator** is the safest choice for maximum compatibility. If you are sharing spreadsheets with users running older versions of [Excel](#) or if you are only combining two or three cells, the ampersand formula is quick to write and universally recognized. Its main drawback is the length and complexity it introduces when attempting to combine data from a large number of columns, as every cell and every [delimiter](#) must be explicitly listed and separated by the & symbol.

The [CONCAT function](#) offers a modern replacement for the older `CONCATENATE` function and is ideal for users on post-2016 versions of [Excel](#). While its syntax is cleaner than the ampersand, it still requires manual insertion of the space delimiter argument (" ") between the [cell references](#) if you are combining distinct cells. However, its ability to handle contiguous ranges makes it marginally more powerful than the simple operator for some tasks, bridging the gap between basic [concatenation](#) and advanced text joining.

Finally, the [TEXTJOIN function](#) is highly recommended for any environment where robustness and efficiency are critical. If you are joining data from many columns (e.g., combining street address, city, state, and zip code), or if you anticipate missing data (empty cells) that should not result in extra separators, [TEXTJOIN](#) is the superior choice. Its structure--defining the delimiter once--simplifies formula creation and maintenance immensely.

## Summary and Additional Resources for Excel Data Manipulation

Combining values from multiple columns is a foundational skill in data processing within [Excel](#). Whether you choose the reliable **Ampersand Operator**, the efficient [CONCAT function](#), or the advanced [TEXTJOIN function](#), the ability to properly merge and format data ensures that your spreadsheets are clean, readable, and ready for further analysis or reporting. Always remember that when inserting a space or any other separator, it must be treated as a literal text string enclosed in double quotation marks (" ").

We have demonstrated that each method successfully transforms two separate columns of data (first name and last name) into a single, consolidated column (full name) using a space as the necessary separator. Selecting the appropriate tool depends on your specific version of [Excel](#) and the complexity of the data range you are manipulating. For modern data professionals, understanding the capabilities of [TEXTJOIN](#) can be a significant time-saver.

To further enhance your skills in data preparation and manipulation within spreadsheets, consider

exploring related techniques such as splitting text columns (the inverse of concatenation), trimming excess whitespace, or using conditional formatting based on the combined cell values.

## **Additional Resources**

The following tutorials explain how to perform other common operations in [Excel](#), building upon the foundational knowledge of text manipulation:

Understanding the difference between absolute and relative [cell references](#) when copying formulas.

Using the `TRIM` function to clean up unwanted spaces before [concatenation](#).

Exploring advanced uses of the `IF` function in conjunction with text manipulation.