

How to Use Exact and Approximate Matches in VLOOKUP with Google Sheets

Authored by
stats writer

January 15, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Use Exact and Approximate Matches in VLOOKUP with Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126267>

The VLOOKUP function is a foundational tool within Google Sheets, designed to facilitate vertical lookups within large datasets. Its primary purpose is to efficiently search for a specific criterion, known as the search key, within the first column of a designated range, and then retrieve a corresponding value from a column further to the right. Mastering this function is essential for data analysis, reporting, and automated information retrieval across various business and academic applications. However, the true power and potential pitfalls of VLOOKUP often lie in the manipulation of its final, optional argument: the parameter that dictates the required match type.

This critical parameter, typically represented by the Boolean values of **TRUE** or **FALSE**, determines whether the function should seek an approximate match or an exact match, respectively. Understanding the fundamental distinction between these two modes is paramount for ensuring the integrity and accuracy of your data lookups. Setting this argument to **TRUE** instructs VLOOKUP to find the closest value less than or equal to the search key, a behavior that is beneficial for tasks like calculating tax brackets or assigning letter grades based on numerical scores. Conversely, setting it to **FALSE** demands perfect correspondence, ensuring that the function only returns a result if an identical value is found, which is the preferred method for locating specific items or unique identifiers.

Understanding the VLOOKUP Syntax and Arguments

To utilize the power of **VLOOKUP** effectively within Google Sheets, a thorough understanding of its required structure and arguments is necessary. The function demands four components, three of which are mandatory for successful execution, defining precisely what data is sought, where it resides, and what corresponding information should be retrieved.

The standard syntax for this function is as follows:

VLOOKUP(search_key, range, index,)

Each element plays a distinct and crucial role:

search_key: This is the specific piece of data--a number, text string, or cell reference--that you are attempting to locate. It is mandatory that this key exists in the very first column of the specified range.

range: This defines the entire block of cells where the data lookup will occur. It must encompass both the column containing the **search_key** (always the first column) and the column containing the desired return value.

index: Represented by a numerical value, this argument specifies which column within the defined **range** holds the result you wish to return. The counting begins at 1 (the **search_key** column) and moves rightward.

is_sorted: This is the optional, yet most critical, Boolean argument that dictates the matching behavior. Using **TRUE** triggers an approximate match search, while using **FALSE** mandates an exact match.

The **is_sorted** argument is the focus of precision in data retrieval. If omitted, the system defaults to **TRUE**, which assumes your data is sorted and attempts an approximate match. While this default setting might seem convenient, it is often misleading for standard data retrieval tasks where specific unique IDs or names must be located precisely. Consequently, in the vast majority of practical applications involving specific lookups, explicitly setting this argument to **FALSE** is considered a best practice to prevent unexpected or inaccurate results.

When to Use TRUE: The Approximate Match Scenario

Setting the optional fourth argument in VLOOKUP to **TRUE** activates the approximate matching mode. This functionality is distinctly different from standard lookups, as it does not require a perfect correspondence between the **search_key** and the values in the first column of the range. Instead, VLOOKUP scans the lookup column and stops at the largest value that is less than or equal to the **search_key**. This capability makes **TRUE** invaluable when working with tiered structures, such as commission rates, grading scales, or numerical thresholds.

A classic application of the approximate match involves translating numerical scores into categorical labels, such as converting raw test scores into letter grades (A, B, C, etc.). In this scenario, you define the minimum score required for each grade level in your lookup table. If a student scores 87, VLOOKUP looks down the table and finds the highest boundary that 87 surpasses (e.g., the boundary for a 'B' grade), returning the corresponding letter. This tiered approach avoids the necessity of listing every single possible score, making the lookup table significantly more manageable and efficient for interpolation tasks.

However, for **VLOOKUP** to function correctly in approximate match mode, there is a strict, non-negotiable requirement: the first column of your lookup **range** must be sorted in **ascending order**. If the data is not meticulously sorted from smallest to largest, the function will fail to traverse the data correctly and will likely return completely erroneous values, or sometimes an error, because the binary search algorithm it employs cannot locate the appropriate threshold efficiently. Neglecting this sorting requirement is the most common reason users experience unexpected results when utilizing the **TRUE** argument.

When to Use FALSE: Ensuring Exact Data Retrieval

When the goal is to find a unique, unambiguous piece of information--such as an employee ID, product SKU, or specific name--the **FALSE** argument must be used for the **is_sorted** parameter. Setting this value mandates an exact match, meaning the VLOOKUP function will only return a

value if it finds the **search_key** replicated perfectly within the first column of the **range**. If the exact value is not present, the function returns the **#N/A** error, indicating that the value is not available in the dataset.

The primary advantage of using **FALSE** is the elimination of ambiguity. Since it does not rely on proximity or data sorting, it guarantees that the returned data corresponds precisely to the item being sought. This is the preferred method for transactional data lookups, inventory management, and linking data across multiple tables based on unique keys. Due to this reliability and lack of sorting requirement, **FALSE** is overwhelmingly the most frequently used setting in professional and analytical applications of VLOOKUP.

Unlike the approximate match mode, the exact match mode (**FALSE**) does not require the source data to be sorted. This flexibility significantly simplifies data preparation and maintenance. However, it is crucial to remember that the search remains case-insensitive for text strings in Google Sheets, meaning "Apple" and "apple" are treated as the same match. If an identical **search_key** appears multiple times in the first column, **VLOOKUP** will always return the value associated with the **first instance** it encounters, highlighting the importance of using unique identifiers whenever possible.

Practical Demonstration: Setting up the Dataset

To clearly illustrate the fundamental difference in operation between using **TRUE** (approximate match) and **FALSE** (exact match), we will analyze a simple dataset within [Google Sheets](#). This dataset represents fictional sports team scores, where we have a master list of teams and their respective points, and a separate list of teams we wish to look up.

Our source data is located in columns A and B, defining the lookup **range**. Column A contains the Team Name, and Column B contains the Points. Columns D and E will be used for our lookup exercise. Column D contains the **search_keys** (the teams we are looking for), and Column E will display the results generated by the **VLOOKUP** formulas.

Observe the initial configuration of the data below. Note that the master list of Team Names (Column A) is not sorted alphabetically, which is a critical factor when testing the behavior of the **TRUE** argument:

	A	B	C	D
1	Team	Points		
2	Mavs	22		
3	Warriors	29		
4	Cavs	35		
5	Heat	13		
6	Thunder	18		
7	Rockets	29		
8	Spurs	24		
9	Lakers	10		
10	Nuggets	14		
11				
12				
13				
14				
15				

The objective across both subsequent examples is to use the team names in Column D to retrieve their corresponding points from Column B. The success of the retrieval hinges entirely on how we define the final **is_sorted** argument.

Example 1: Analyzing Results with Approximate Match (TRUE)

In this first demonstration, we intentionally use the **TRUE** argument to observe the behavior of **VLOOKUP** when attempting an approximate match. We are looking up the team names in column D within the defined range **\$A\$2:\$B\$10**, targeting the second column for the return value. Critically, we set the final argument to **TRUE**, indicating that the system should find the closest match.

The formula applied to the first cell (D2) and copied down the column is:

=VLOOKUP(D2, \$A\$2:\$B\$10, 2, TRUE)

The resulting calculation, shown in the screenshot below, immediately highlights the danger of using **TRUE** when seeking specific, textual data. Because our lookup table (Column A) is not sorted alphabetically, the binary search utilized by the approximate match function fails spectacularly, resulting in illogical and incorrect associations. For instance, looking up "Team 9" returns 55 points, which is the point value for "Team 4" in the dataset, demonstrating a complete

breakdown of data integrity.

E2 ∇ | fx =VLOOKUP(D2, \$A\$2:\$B\$10, 2, TRUE)

	A	B	C	D	E
1	Team	Points		Team Lookup	Points
2	Mavs	22		Spurs	13
3	Warriors	29		Mavs	13
4	Cavs	35		Lakers	13
5	Heat	13			
6	Thunder	18			
7	Rockets	29			
8	Spurs	24			
9	Lakers	10			
10	Nuggets	14			
11					
12					
13					
14					

This example serves as a powerful reminder: unless you are dealing with numerical thresholds (like grading scales) and your data is meticulously sorted in ascending order, the use of **TRUE** for data lookup in Google Sheets will almost certainly yield inaccurate results. For simple retrieval of text or identifiers, this setting should be strictly avoided.

Example 2: Achieving Precision with Exact Match (FALSE)

To contrast the chaotic results of the previous example, we now modify only the final argument of the VLOOKUP function, switching it from **TRUE** to **FALSE**. This change forces the function into exact match mode, requiring a perfect character-for-character match of the **search_key** before any value is returned.

The revised formula, applied to D2 and copied down, now strictly demands precision:

=VLOOKUP(D2, \$A\$2:\$B\$10, 2, FALSE)

As clearly demonstrated in the resulting spreadsheet view below, the outcomes are now accurate and reliable. The function successfully located each specific team name in Column A and returned its corresponding points value from Column B, regardless of the unsorted nature of the dataset. For example, "Team 5" correctly returns 90 points, and "Team 1" correctly returns 78 points.

E2 fx =VLOOKUP(D2, \$A\$2:\$B\$10, 2, FALSE)

	A	B	C	D	E
1	Team	Points		Team Lookup	Points
2	Mavs	22		Spurs	24
3	Warriors	29		Mavs	22
4	Cavs	35		Lakers	10
5	Heat	13			
6	Thunder	18			
7	Rockets	29			
8	Spurs	24			
9	Lakers	10			
10	Nuggets	14			
11					
12					
13					
14					

This consistency confirms why **FALSE** is the default choice for lookups involving non-numeric or unsorted data. By insisting on an exact correspondence, the analyst ensures data integrity and avoids the unpredictable behaviors associated with approximate matching on unsuitable datasets. If a team name were misspelled in Column D, the function would simply return **#N/A**, clearly signaling a data mismatch rather than returning an incorrect, approximate value.

Conclusion and Best Practices for VLOOKUP

The distinction between using **TRUE** and **FALSE** in the final argument of the VLOOKUP function is arguably the most important concept to master when working with data retrieval in Google Sheets. While **TRUE** offers a specialized method for handling tiered numerical data, its strict requirement for ascending sorting and propensity to return incorrect matches on unsorted text data makes it risky for general use. Conversely, **FALSE** provides the robust guarantee of an exact match, making it the industry standard for specific data lookups.

As a general rule of thumb, always default to using **FALSE** unless you are explicitly creating a tiered grading or commission schedule based on ascending numerical criteria. Even when using **TRUE**, it is imperative to double-check that the lookup column is sorted correctly, or the function will inevitably fail its purpose. Choosing the correct Boolean value ensures that your spreadsheet analyses remain both efficient and trustworthy.

For further technical details and advanced usage scenarios, you can consult the complete

documentation for the Google Sheets **VLOOKUP** function provided by Google.

ARABPSYCHOLOGY.COM