Understanding VLOOKUP with TRUE/FALSE in Google Sheets

Authored by

Mohammed looti

November 12, 2025

Introduction to the VLOOKUP Function and the Range Lookup Parameter

The **VLOOKUP** function stands as one of the most foundational and powerful utilities available within **Google Sheets** and other spreadsheet applications. Its primary purpose is to execute a vertical lookup: searching for a specified value within the leftmost column of a designated data range and then returning a corresponding value from a column located to the right. This mechanism is indispensable for tasks requiring the efficient cross-referencing of large datasets, the merging of information across multiple tables, and the overall automation of data retrieval processes. However, mastering this function requires a deep understanding of all its components, particularly the final, often misunderstood argument, which critically determines the function's search behavior and, consequently, the accuracy of your results.

While the core concept of **VLOOKUP** seems straightforward, many users encounter frustrating errors or retrieve misleading data because they fail to properly define this final argument, known as the range lookup parameter. This parameter dictates whether the function seeks a perfect correspondence (an **exact match**) or accepts the closest possible value (an **approximate match**). For analysts and data professionals, the distinction is paramount; relying on the function's default behavior without fully appreciating the implications of seeking an approximate result can introduce significant errors into complex workbooks. This comprehensive guide will dissect the role of the final, optional **Boolean** value--**TRUE** or **FALSE**--and demonstrate through practical, real-world examples how this setting drastically impacts the reliability and precision of your data analysis.

Deconstructing the VLOOKUP Syntax and Its Four Critical Arguments

The functional structure of the **VLOOKUP** command is standardized, requiring four distinct pieces of information to successfully locate and return data. A clear grasp of each argument's role is essential for proficient use of this utility, ensuring that the function retrieves precisely the data intended. The syntax, as structured in **Google Sheets**, is presented below, with the final argument enclosed in brackets to denote its optional nature--though its importance is anything but optional:

VLOOKUP(search_key, range, index,)

Each element plays a specific, non-negotiable role in the vertical lookup process:

search_key: This is the identifier--a text string, unique code, or numeric value--that you are trying to locate. It is the core input that drives the search.
range: This defines the entire block of cells encompassing the source data. It is crucial to remember that the **search_key** must always be present in the leftmost column of this defined **range**, as the **VLOOKUP** function is fundamentally unable to search columns to the left of the lookup column.
index: This numerical argument specifies which column within the defined **range** contains the

desired return value. The count begins at 1 for the leftmost column. For example, if your **range** covers columns B through E, and you wish to retrieve data from column D, the index would be 3.
is_sorted: This is the critical, optional **Boolean** argument. Setting it to **TRUE** initiates an **approximate match** search, while setting it to **FALSE** demands an **exact match**.

The parameter **is_sorted** is the primary focus of this analysis. The choice between **TRUE** and **FALSE** determines the reliability of the retrieved data. When users omit this argument, the function defaults to **TRUE**, assuming the data is pre-sorted and proceeding with an approximate lookup. This default behavior is frequently the source of cryptic formula errors and data mismatches for users who expect a perfect correspondence between their search term and the dataset.

Understanding the Default Behavior: Approximate Matching with TRUE

When the final argument, **is_sorted**, is set to **TRUE** (or is simply omitted, leveraging the default setting), the **Google Sheets** function engages in an **approximate match** search. In this mode, the function is not necessarily seeking an identical counterpart for the **search_key**. Instead, it aims to locate the largest value in the first column of the **range** that is less than or equal to the specified **search_key**. This functionality is highly valuable for specialized applications, such as calculating numerical thresholds--think grading systems, tax brackets, or discount tiers--where input values fall within defined ranges.

However, relying on **TRUE** carries a critical technical prerequisite: the first column of the search **range** must be strictly sorted in **ascending order**. If the source data is unsorted, the **VLOOKUP** function will fail its search logic and frequently return completely random and inaccurate results. The function assumes the sorted structure allows it to efficiently stop searching once it passes the necessary threshold. If the data order is randomized, this assumption causes an immediate failure, often retrieving a value far removed from the intended search criteria without signaling a clear error like **#N/A**.

Due to the strict requirement for ascending order and the inherent risk of returning a value that is merely "close enough," using **TRUE** is generally discouraged unless the user explicitly needs numerical range lookup functionality. For the vast majority of common data analysis tasks involving unique identifiers--such as product SKUs, client names, or distinct IDs--the potential ambiguity introduced by an **approximate match** makes **TRUE** unreliable. This default setting is a frequent cause of wasted time, as users troubleshoot formula issues only to find that the function was returning misleading, yet technically non-error, data points.

Achieving Guaranteed Precision: Exact Matching with FALSE

In sharp contrast to the potential ambiguity of the default setting, explicitly setting **FALSE** for the **is_sorted** argument mandates that the **VLOOKUP** function search exclusively for an **exact match**

of the **search_key**. When **FALSE** is utilized, the function executes a rigorous, sequential scan through the entire first column of the **range**, looking only for an entry that is textually or numerically identical to the requested value. This process ensures absolute correlation between the lookup value and the data point retrieved.

Employing **FALSE** offers several significant advantages crucial for maintaining data integrity. First, it maximizes the reliability of the results; if a value is returned, it is guaranteed to correspond precisely to the searched item. Second, the use of **FALSE** eliminates the prerequisite for the source data to be sorted. The function will successfully search the column regardless of the order, providing crucial flexibility when handling dynamic or ad-hoc datasets. Crucially, if the function fails to locate an **exact match**, it provides the standard, unambiguous error value **#N/A**, clearly signaling that the search key does not exist within the specified **range**.

For most practical spreadsheet applications involving non-numeric, categorical, or identifier data, setting the final argument to **FALSE** is the undisputed industry best practice. While an **exact match** search may theoretically be marginally slower than an approximate search on exceptionally large, perfectly sorted datasets, this negligible performance difference is overwhelmingly eclipsed by the increased accuracy and reduced risk of critical data errors. Consequently, unless you are deliberately configuring a lookup table for numerical tiers, the use of **FALSE** is mandatory to enforce strict matching criteria and ensure reliable data extraction.

Practical Demonstration: The Pitfalls of Using TRUE (Example 1)

To clearly illustrate the critical operational differences between these two search methodologies, let us examine a common scenario involving a simple dataset tracking team scores. The following table displays team names and their corresponding points in the source columns (A and B). Our objective is to retrieve the points for the specific team names listed in column D:

	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				

In this initial example, we intentionally set the final argument to **TRUE**, thereby requesting an **approximate match**. We aim to look up the team names (starting with cell D2) within the range A2:B10 and return the corresponding points from the second column (index 2). Note that, for this demonstration, the primary search column (A) is sorted alphabetically, meeting the requirement for a **TRUE** lookup.

The formula applied in cell E2 and copied down the column is structured as follows, with **TRUE** explicitly defining the search behavior:

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

The immediate consequence of utilizing **TRUE**, even with a sorted list, is the profound potential for error when the list of teams being searched (Column D) does not perfectly match the source data (Column A). The following results reveal the outcome of this approximate search:

E2 =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					

As evident in column E, the function sought the largest value in Column A that was less than or equal to the lookup term in Column D based on alphabetical order. This approximation resulted in significant data corruption. For instance, "Team A" incorrectly returned 30 points, and "Team F" returned 55 points, neither of which aligns with the actual scores in the original table. This clearly illustrates that when dealing with unique, non-numeric identifiers, the **approximate match** feature fundamentally undermines data accuracy, demonstrating the severe limitations of the default **TRUE** setting.

Practical Demonstration: Ensuring Accuracy with FALSE (Example 2)

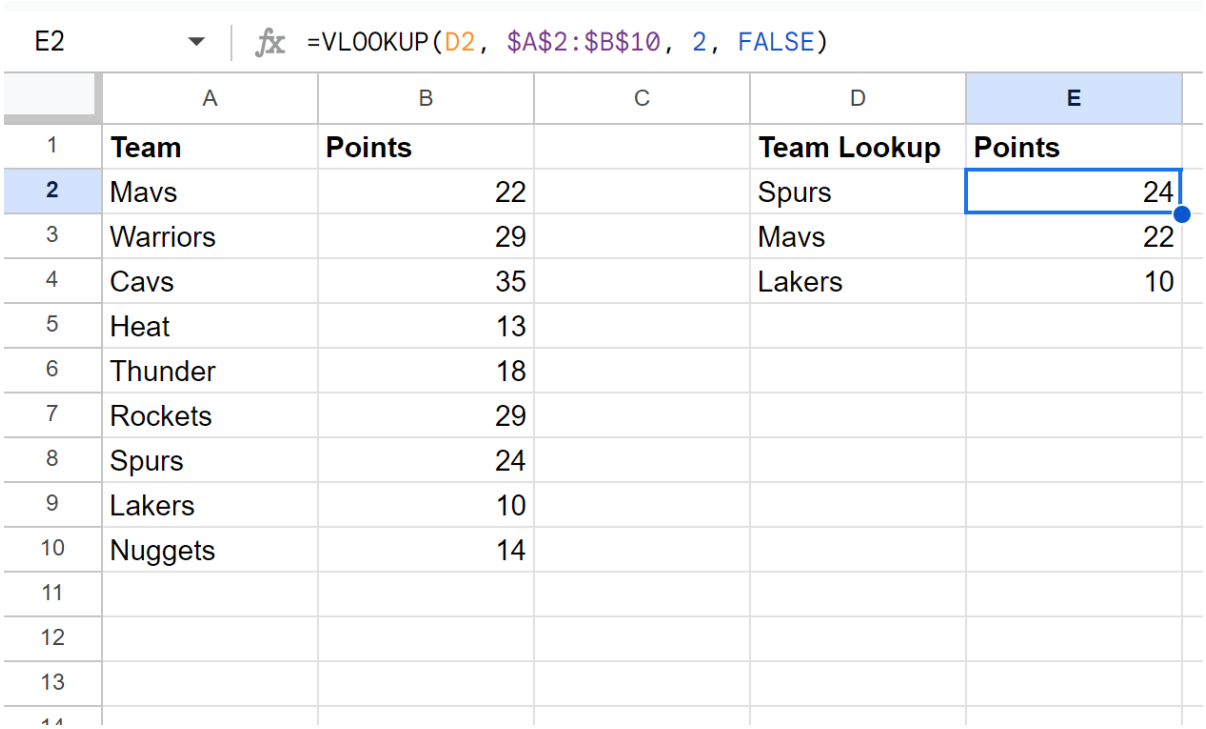
Now, we adjust the final argument to mandate an **exact match**, preserving the integrity of the data. We utilize the identical dataset and the same search objective: locating team names in column D within the range A2:B10 and returning the points value from column 2. By changing the **Boolean** value from **TRUE** to **FALSE**, we explicitly instruct **Google Sheets** to accept only results that are a perfect, character-for-character textual match for the team names.

The revised formula applied to cell E2, demanding absolute precision, is now:

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

Observe the immediate and dramatic transformation in the results when this formula is deployed

across column E, prioritizing definitive accuracy over approximation:



	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					

Because we specified **FALSE** for the **is_sorted** argument in **VLOOKUP**, the function successfully located the **exact match** for every team name present in the source data. The points values returned in column E now perfectly correspond to the original table (A2:B10). For example, "Team C" correctly returns 55 points, and "Team H" correctly returns 75 points. By insisting upon **FALSE**, we eliminate the inherent ambiguity of approximate searches and guarantee that the returned data is a precise, accurate representation of the source information, which is a fundamental requirement for reliable data integration and reporting.

Summary of Best Practices for Reliable Lookups

The choice between **TRUE** and **FALSE** in the final argument of the **VLOOKUP** function is a pivotal decision that dictates the functional integrity of your spreadsheet model. While **TRUE** is the default setting and is appropriate exclusively for numerical range lookups where the source data is meticulously sorted in ascending order, **FALSE** is the compulsory setting for virtually all lookups involving text strings, unique identifiers, product codes, or any categorical data where an **exact match** must be guaranteed. Failing to explicitly mandate **FALSE** when seeking precision will almost certainly lead to difficult-to-detect data mismatches and ultimately compromise the accuracy of your analysis.

To maintain robust and trustworthy spreadsheet models, adopting the following best practice is

essential: always explicitly specify the final argument. Never rely on the default **TRUE** setting unless you have intentionally verified that your lookup column is sorted ascendingly and you specifically require an **approximate match** for tiered data. For standard data retrieval and integration tasks--which constitute the overwhelming majority of everyday spreadsheet operations--setting the final argument to **FALSE** is the simplest, most effective, and most reliable way to ensure absolute data accuracy.

Additional Resources

For those seeking deeper technical information or exploring advanced uses of the vertical lookup function, the complete documentation for the **VLOOKUP** function is provided officially by **Google Sheets**:

The following tutorials explain how to perform other common operations in **Google Sheets**: