

# Learning Google Sheets: Conditional Formatting Based on Text in Another Cell

Authored by  
**Mohammed loot**

October 27, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning Google Sheets: Conditional Formatting Based on Text in Another Cell*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=3925>

Harnessing the full power of [Google Sheets](#) requires moving beyond basic data entry and embracing sophisticated visualization techniques. One of the most impactful tools for data analysis is [conditional formatting](#), particularly when combined with the versatile [custom formula](#) feature. This synergy allows spreadsheet users to apply dynamic styles--such as color changes or font modifications--to [cells](#) based on complex, user-defined criteria. Crucially, this advanced capability extends beyond simple numeric comparisons; it enables formatting based on the textual content found in entirely different [cells](#). This method is foundational for creating highly responsive and informative dashboards, ensuring that critical data points are immediately visible and actionable.

The primary objective of this comprehensive guide is to demystify the process of implementing text-based conditional formatting using the powerful [custom formula](#) function. We will meticulously break down the necessary steps, starting from selecting the target data and progressing through the creation and refinement of the logical formulas. Our step-by-step approach will utilize a practical, real-world example--color-coding player performance ratings--to ensure maximum clarity and precision in understanding the underlying mechanics. This detailed walkthrough will solidify your ability to link formatting triggers across disparate columns within your spreadsheet architecture.

Upon concluding this tutorial, you will possess the specialized knowledge required to construct sophisticated formatting rules that automatically respond to changes in your underlying data. This skill set dramatically enhances the readability and interpretability of complex spreadsheets, transforming raw [dataset](#) information into intuitive visual intelligence. For professionals managing extensive records, tracking project statuses, or analyzing performance metrics, this technique is absolutely invaluable for quickly identifying critical patterns, anomalies, or entries requiring immediate attention, thereby streamlining decision-making processes.

## The Foundational Role of Custom Formulas in Conditional Formatting

To fully appreciate the utility of this technique, it is essential to first establish a strong conceptual understanding of [conditional formatting](#) itself. At its core, this feature in [Google Sheets](#) operates as an automated styling system: it applies specific aesthetic treatments--such as updating background colors, modifying text colors, or changing font weights--to [cells](#) only when predefined logical criteria are satisfied. This mechanism eliminates the need for manual data inspection, automating the visual identification process and significantly reducing the risk of human oversight when dealing with voluminous data. It shifts the focus from passively viewing data to actively interpreting visually categorized information.

While Google Sheets provides a comprehensive suite of built-in conditional formatting options--handling simple rules like "value is greater than X" or "date is before Y"--the maximum flexibility and analytical depth are unlocked through the use of the [custom formula](#) option. This powerful

feature enables users to define virtually any condition using standard spreadsheet formula syntax, extending capabilities far beyond basic comparisons. Custom formulas allow for intricate logical evaluations that can reference multiple cells, perform calculations, or, most relevant to our discussion, trigger formatting based on text strings contained within another column or [range](#).

The ability to reference external cells for formatting conditions is a paradigm shift in data visualization. Instead of formatting cell A based only on the value in cell A, we can format cell A based on the status reported in cell C, or the category assigned in cell D. This non-adjacent dependency is critical for developing dynamic status trackers, project management dashboards, or any [dataset](#) where visual cues need to reflect categorical data points. For instance, if you track inventory, you might highlight the product name (Column A) if the stock level (Column E) reads "Low." Our tutorial specifically addresses the challenge of making the visual cue (e.g., color) dependent on a specific text value (e.g., "Pending," "Complete," or, in our example, "Great").

## Preparing the Spreadsheet Environment and Data Structure

To demonstrate the practical mechanics of cross-cell conditional formatting, we will work within a clearly defined scenario. We are compiling and managing a [dataset](#) containing performance information related to various basketball players. This example requires a structured table that includes key identifiers and evaluative data points. The resulting visual categorization must allow stakeholders to instantly identify player performance tiers without needing to read the specific rating text for every entry.

The foundational structure of our sample dataset is designed to facilitate our formatting exercise. It comprises three distinct columns: **Player** (Column A), **Team** (Column B), and **Rating** (Column C). The **Rating** column contains the categorical text values ("Great," "Good," or "Bad") that will serve as the logical triggers. The data below shows the initial, unformatted state of the spreadsheet, establishing the baseline upon which we will build our visualization rules.

	A	B	C	D
1	<b>Team</b>	<b>Points</b>	<b>Rating</b>	
2	Mavs	34	Great	
3	Nets	29	Good	
4	Warriors	25	Good	
5	Heat	25	Good	
6	Kings	22	Good	
7	Lakers	17	Bad	
8	Hornets	29	Good	
9	Pacers	18	Bad	
10	Cavs	38	Great	
11	Suns	22	Bad	
12				
13				
14				
15				
16				
17				
18				
19				
20				

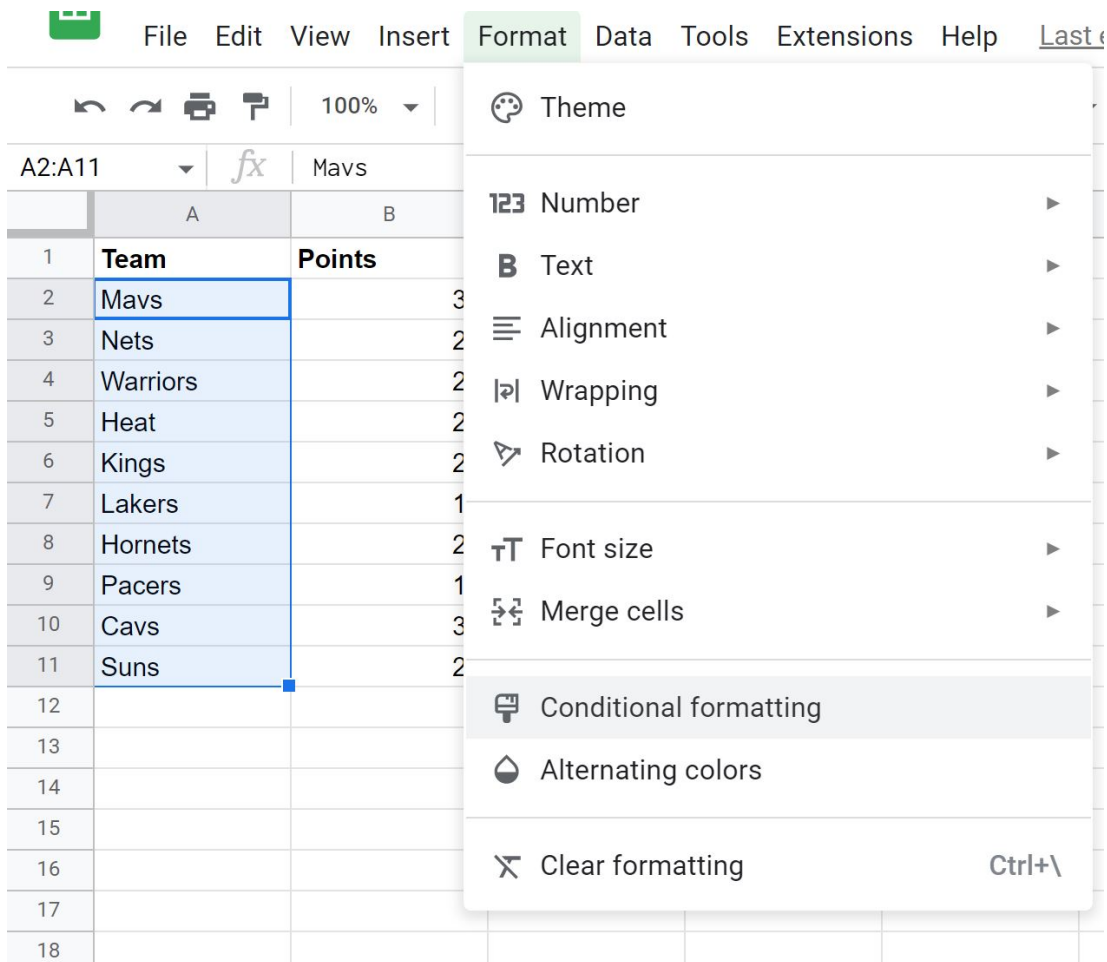
Our specific goal is to apply visual highlighting to the entries within the **Team** column (Column B). The applied color or style will be entirely contingent upon, or dynamically linked to, the textual content present in the corresponding row of the **Rating** column (Column C). This dynamic linkage--formatting B based on C--is the precise capability that the [custom formula](#) enables. This approach allows for an intelligent data presentation where the visual importance of one piece of data is dictated by the qualitative assessment contained within another.

## Defining the Target Range and Accessing the Rules Panel

The first crucial step in setting up any conditional formatting rule is accurately specifying the area of the spreadsheet that should receive the formatting changes--the target [range](#). In our basketball performance example, we aim to highlight the team names, which reside in Column B. Therefore, we must select the data portion of that column, excluding the header row. The correct selection for our sample data is the [range B2:B11](#). Selecting this precise [range](#) ensures that the formatting rules are applied exclusively to the relevant team data.

Once the target cells (**B2:B11**) are highlighted, the next step is to access the dedicated conditional formatting interface. Navigate to the top menu bar in [Google Sheets](#) and click on the **Format** tab.

This action will reveal a comprehensive dropdown menu containing various options for cell and text styling. Among these options, select **Conditional formatting**. This selection immediately opens the **Conditional format rules** panel, typically situated on the right side of your screen, which serves as the control center for defining, managing, and prioritizing all visual rules within the sheet.



This panel is where the logic is injected into the visualization process. Before proceeding, verify that the 'Apply to range' field correctly displays **B2:B11** (or the relevant range for your project). If you had selected the range prior to opening the panel, it should populate automatically. If not, you can manually input or adjust the range here. This preparatory step is vital, as an incorrect range definition will either fail to format the intended [cells](#) or apply rules where they are not desired.

## Implementing the Custom Formula Logic and Relative Referencing

Within the **Conditional format rules** panel, locate the **Format rules** section. We must change the default rule type to enable advanced formula input. Click on the **Format cells if** dropdown menu. Scroll through the predefined options (such as 'Text contains' or 'Date is') until you reach and select the final option: **Custom formula is**. Selecting this option immediately reveals a dedicated

text input field, granting you the ability to write a logical expression that precisely controls when the formatting is triggered. This is the core mechanism that connects the rating text to the team name's appearance.

For our immediate goal--highlighting teams associated with a "Great" rating--we must define a formula that checks the corresponding cell in the Rating column (Column C). The formula to achieve this specific condition is entered as follows:

**=C2="Great"**

Understanding the concept of **relative referencing** in this context is paramount. Although the formula is written using the explicit reference **C2**, [Google Sheets](#) does not restrict the evaluation solely to cell C2. When the conditional formatting rule is applied across the specified [range](#) (B2:B11), the formula is evaluated row-by-row, relative to the top-left cell of the defined range. For the cell **B2**, the rule checks the condition in **C2**. For the cell **B3**, it automatically adjusts and checks **C3**, and so forth, down to **B11** checking **C11**. The formula [=C2="Great"](#) is a Boolean expression that returns **TRUE** if the content of the corresponding cell in column C matches the text string "Great" exactly, and **FALSE** otherwise. Formatting is only applied when the evaluation yields **TRUE**.

Conditional format rules

**Single color** Color scale

Apply to range

A2:A11

Format rules

Format cells if...

Custom formula is

=C2="Great"

Formatting style

Default

**B** *I* U ~~S~~ A ▾ | ▾

Cancel Done

+ Add another rule

After successfully inputting the custom formula, the final step for this rule is to define the visual consequence. Utilizing the formatting options provided directly beneath the formula input field, select a distinct style--such as a vibrant green fill color--to represent "Great" performance. This color selection should be easily distinguishable from the default sheet background. Once the color and style are finalized, click the **Done** button to save and apply the rule instantly. The spreadsheet should reflect the change immediately, highlighting all team names corresponding to the "Great" rating.

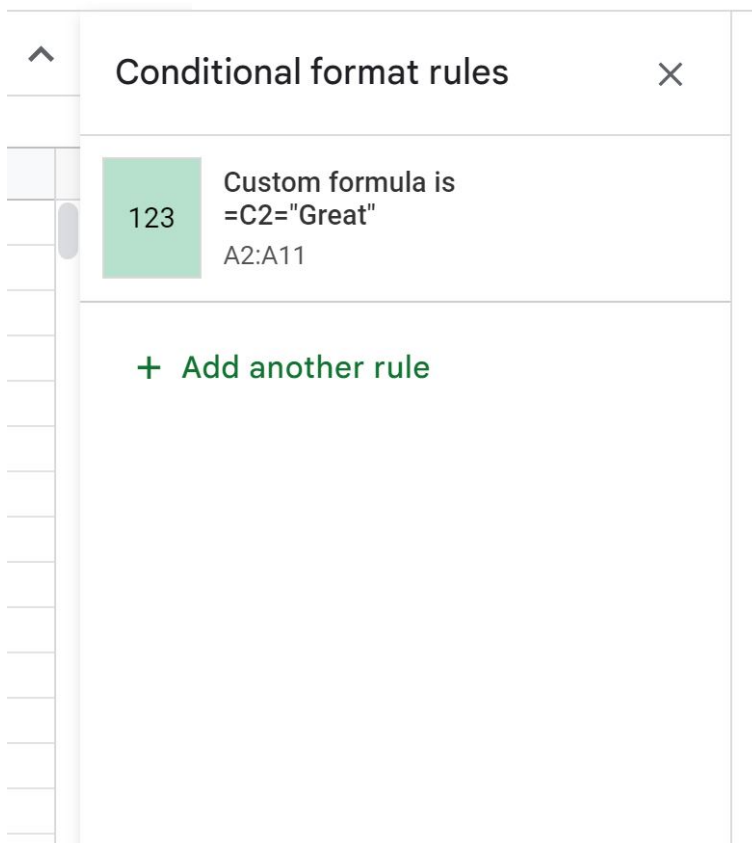
## Validating the Rule Application and Expanding Categorical Visualization

The moment you click **Done**, the power of dynamic formatting becomes evident. The spreadsheet instantly updates, displaying the chosen green background color for every entry in the **Team**

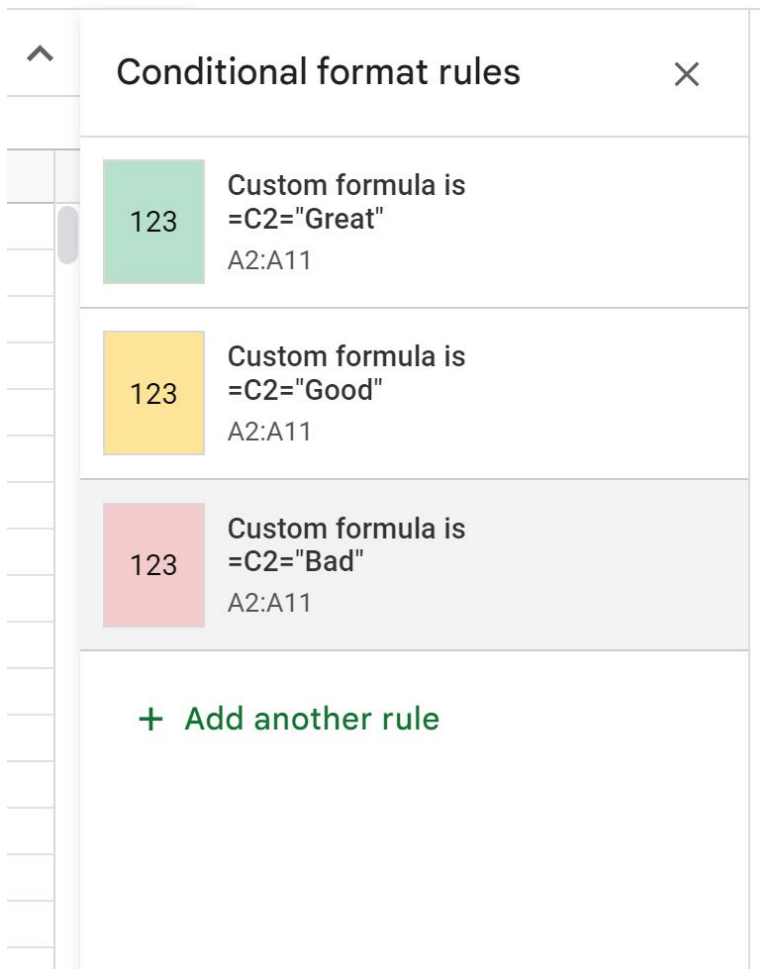
column (B2:B11) that corresponds to a "Great" rating in the adjacent **Rating** column (C). This immediate visual feedback not only confirms the successful implementation of the first rule but also underscores the efficiency of using cross-cell references for rapid data interpretation and analysis. The visual hierarchy begins to take shape, making high-performing teams instantly noticeable.

	A	B	C	D
1	<b>Team</b>	<b>Points</b>	<b>Rating</b>	
2	Mavs	34	Great	
3	Nets	29	Good	
4	Warriors	25	Good	
5	Heat	25	Good	
6	Kings	22	Good	
7	Lakers	17	Bad	
8	Hornets	29	Good	
9	Pacers	18	Bad	
10	Cavs	38	Great	
11	Suns	22	Bad	
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				

To achieve a comprehensive visual categorization, it is necessary to establish rules for all possible text conditions present in the [dataset](#). The strength of [conditional formatting](#) lies in its ability to manage multiple, distinct rules simultaneously. To address the other rating categories ("Good" and "Bad"), simply return to the **Conditional format rules** panel and click the **Add another rule** button. This action generates a new, blank rule template, pre-set to apply to the same target [range](#) (B2:B11).



This iterative process requires repeating the steps for selecting **Custom formula is** and defining the specific logic and styling for each remaining category. For instance, to highlight teams with a "Good" rating, the new rule would use the custom formula `=C2="Good"` and be assigned a yellow background color, signaling moderate performance. Subsequently, a third rule for "Bad" ratings would use the custom formula `=C2="Bad"` and be styled with a red background, drawing attention to areas requiring improvement. This layering of rules creates a robust, multi-tiered visualization system that adapts fluidly to the underlying data changes, providing immediate and intuitive status updates.



## Analyzing the Multi-Layered Visual Data Presentation

Once all necessary conditional formatting rules--covering "Great," "Good," and "Bad" performance ratings--have been successfully defined and applied, the transformation of your spreadsheet is complete. What started as static text data is now a dynamic, color-coded resource that provides immediate, actionable insights. The final output serves as a sophisticated visual representation, clearly distinguishing performance tiers across the entire spectrum of the basketball player data. This visual organization drastically cuts down the time required for data scanning and interpretation.

	A	B	C	D
1	<b>Team</b>	<b>Points</b>	<b>Rating</b>	
2	Mavs	34	Great	
3	Nets	29	Good	
4	Warriors	25	Good	
5	Heat	25	Good	
6	Kings	22	Good	
7	Lakers	17	Bad	
8	Hornets	29	Good	
9	Pacers	18	Bad	
10	Cavs	38	Great	
11	Suns	22	Bad	
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				

A detailed analysis of the finalized, formatted output confirms the accuracy and utility of our custom formulas based on text references:

The **Team** entries corresponding to a **Rating** of "Great" are highlighted in green, instantly flagging top performers and successes.

Teams categorized with a **Rating** of "Good" are assigned a yellow background, indicating satisfactory but not exceptional status.

Conversely, teams associated with a **Rating** of "Bad" are prominently marked in red, effectively signaling potential problem areas or data points requiring immediate managerial attention or further investigation.

It is important to acknowledge how the system handles exceptions. Should a new data entry introduce a **Rating** value that falls outside the defined criteria--such as "Average" or if the cell is left blank--that specific team name would simply retain its default background color (typically white or unformatted). Since no specific rule was created to match that text string, the formatting is ignored. This behavior ensures that only the relevant, predefined categories receive visual emphasis, maintaining the integrity and focus of the data visualization. Furthermore, should the

data in Column C change (e.g., a "Good" rating is updated to "Great"), the formatting in Column B will dynamically update in real-time without any manual intervention.

## Conclusion: Maximizing Data Insights Through Dynamic Formatting

The mastery of [conditional formatting](#) coupled with [custom formulas](#) represents a significant elevation in data management skills within Google Sheets. This technique transforms standard, static spreadsheets into highly dynamic, interactive analytical instruments that provide immediate and intuitive visual cues. By leveraging rules that are triggered by text content in adjacent or distant [cells](#), users gain the power to create bespoke visualizations perfectly suited to diverse operational needs--from tracking project milestones and managing inventory levels to evaluating performance metrics in large-scale [dataset](#). This versatility makes the method indispensable for professionals across various industries seeking to streamline analysis and enhance data-driven decision-making.

We strongly encourage users to apply these learned concepts and actively experiment with different formula configurations and complex logical operators (such as [AND](#), OR, and NOT) to unlock the full potential of this feature within their specific projects. The ability to automatically highlight critical data points based on textual categories ensures that your spreadsheets function not merely as data repositories, but as powerful, responsive analytical tools that prioritize attention and focus on the most important information. Continuously refining your custom formatting rules will ensure your data visualization remains effective and aligned with evolving analytical requirements.

For further exploration and to enhance your proficiency in [Google Sheets](#), consider delving into the following related tutorials: