

# Learning to Sum Time Durations Accurately in Google Sheets

Authored by  
**Mohammed loot**

October 27, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Sum Time Durations Accurately in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=4296>

## Introduction: Mastering Time Duration Summation in Google Sheets

Working with **time durations** in [Google Sheets](#) can often present unique challenges, particularly when you need to accurately sum a series of intervals that might exceed a standard 24-hour cycle. Standard [SUM functions](#) tend to reset at the 24-hour mark, leading to incorrect calculations and misleading [data analysis](#). This article will guide you through a robust method to overcome this common hurdle.

To correctly aggregate a range of [time durations](#) in your [spreadsheet](#), you need a specialized formula that accounts for values exceeding 24 hours. The solution involves combining several powerful functions to first convert time values into a numerical format that can be summed indefinitely, and then reformatting the result back into a readable time string.

The following formula provides an effective and reliable way to sum time durations in [Google Sheets](#), ensuring accuracy regardless of the total accumulated time:

```
=ARRAYFORMULA(TEXT(SUM(IFERROR(TIMEVALUE(C2:C8))), ":mm:ss"))
```

This powerful formula is designed to sum all time durations within a specified range, such as **C2:C8** in our example, and present the cumulative result in a clear, unambiguous format of hours, minutes, and seconds. It ensures that the total time correctly accumulates even when it spans multiple days.

### Why Standard SUM() Functions Fall Short for Time Durations

Before diving into the correct solution, it's essential to understand why conventional methods, like simply using the [SUM function](#), often fail when dealing with time durations that might exceed 24 hours. [Google Sheets](#), by default, treats time values as fractions of a 24-hour day. This means that a time like 6:00 AM is numerically represented as 0.25 (1/4 of a day), and 12:00 PM as 0.5 (half a day).

When you sum these fractional values and the total exceeds 1 (representing 24 hours), [Google Sheets](#) automatically converts the integer part of the sum into whole days, displaying only the remaining fractional part as the time. Consequently, if the sum of durations is, for instance, 26 hours, a standard [SUM function](#) would display 02:00:00 (2 hours), effectively resetting the count after every 24-hour period. This behavior is suitable for displaying a time of day, but not for calculating an accumulated duration.

Consider a scenario where you have a dataset detailing the duration of various events. If we were to naively apply the [SUM function](#) to these values, the outcome would be misleading. Suppose we have the following sample data:

	A	B	C	D
1	<b>Start</b>	<b>End</b>	<b>Duration</b>	
2	4:15:22	20:44:21	16:28:59	
3	4:40:01	18:22:32	13:42:31	
4	5:17:22	10:57:43	5:40:21	
5	6:16:00	12:14:01	5:58:01	
6	12:15:23	14:16:22	2:00:59	
7	16:19:28	16:39:43	0:20:15	
8	20:40:45	21:22:45	0:42:00	
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

If we attempt to sum the time durations in column C using the simple [SUM function](#):

**=SUM(C2:C8)**

The result, as illustrated below, will incorrectly reset the sum to 0 each time the accumulated duration surpasses 24 hours. This behavior is a critical limitation for anyone needing to track total elapsed time over longer periods.

	A	B	C	D	E
E2					<i>fx</i> =SUM(C2:C8)
1	<b>Start</b>	<b>End</b>	<b>Duration</b>		<b>Sum of Duration</b>
2	4:15:22	20:44:21	16:28:59		20:53:06
3	4:40:01	18:22:32	13:42:31		
4	5:17:22	10:57:43	5:40:21		
5	6:16:00	12:14:01	5:58:01		
6	12:15:23	14:16:22	2:00:59		
7	16:19:28	16:39:43	0:20:15		
8	20:40:45	21:22:45	0:42:00		
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					

## Step-by-Step Example: Accurately Summing Event Durations

To demonstrate the correct approach, let's use the same dataset as before, which outlines the [time duration](#) for various events. Our objective is to calculate the precise total duration without any resets or data loss.

Instead of the problematic standard [SUM function](#), we must employ the comprehensive formula introduced earlier. This formula meticulously converts each time entry into a numerical value that can be summed, and then formats the final aggregate into a human-readable time string that correctly represents cumulative hours.

Here's the formula we will use:

**=ARRAYFORMULA(TEXT(SUM(IFERROR(TIMEVALUE(C2:C8))), ":mm:ss"))**

Upon applying this formula to our duration column, [Google Sheets](#) will now correctly calculate and

display the sum of all durations, regardless of whether they exceed 24 hours. The result will accurately reflect the total accumulated time, as shown below:

E2 <span style="float: right;">fx</span> =ARRAYFORMULA(TEXT(SUM(IFERROR(TIMEVALUE(C2:C8))), "[h]:mm:ss"))						
	A	B	C	D	E	F
1	<b>Start</b>	<b>End</b>	<b>Duration</b>		<b>Sum of Duration</b>	
2	4:15:22	20:44:21	16:28:59		44:53:06	
3	4:40:01	18:22:32	13:42:31			
4	5:17:22	10:57:43	5:40:21			
5	6:16:00	12:14:01	5:58:01			
6	12:15:23	14:16:22	2:00:59			
7	16:19:28	16:39:43	0:20:15			
8	20:40:45	21:22:45	0:42:00			
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

As clearly demonstrated, the sum of values in the duration column is precisely calculated as **44 hours, 53 minutes, and 6 seconds**. This outcome verifies that our specialized formula effectively bypasses the 24-hour reset limitation, providing an accurate and reliable total for extended [time durations](#).

## Deconstructing the Formula: Key Components Explained

To fully appreciate the power and functionality of this solution, let's break down each component of the formula: `=ARRAYFORMULA(TEXT(SUM(IFERROR(TIMEVALUE(C2:C8))), ":mm:ss"))`

**ARRAYFORMULA:** This function enables the formula to work with an array of cells (e.g., C2:C8) rather than just a single cell. It allows the inner functions like [TIMEVALUE](#) to process multiple cells at once, making the overall formula more concise and efficient. Without [ARRAYFORMULA](#), you would typically need to drag the formula down for each row, or explicitly apply it to each cell.

**TIMEVALUE(C2:C8):** The [TIMEVALUE function](#) converts a time string into a numerical fraction of a 24-hour day. For example, "06:00:00" becomes 0.25. When applied to a range with

[ARRAYFORMULA](#), it transforms each time duration in the specified range (C2:C8) into its corresponding decimal value. This conversion is crucial because standard time formats are text strings, which cannot be reliably summed to exceed 24 hours.

**IFERROR(...):** The [IFERROR function](#) acts as an error handler. It checks if the [TIMEVALUE](#) function returns an error (e.g., if a cell in the range contains non-time text or is empty). If an error occurs, [IFERROR](#) returns a blank value by default, allowing the [SUM function](#) to ignore invalid entries without breaking the calculation. This makes the formula more robust against imperfect data.

**SUM(...):** After [TIMEVALUE](#) converts all valid time durations into numerical fractions, the [SUM function](#) then aggregates these numerical values. Because these are now pure numbers (fractions of days), the [SUM function](#) correctly adds them up, even if the total represents several days' worth of time. For example, if the sum is 1.5, it means 1.5 days or 36 hours.

**TEXT(..., ":mm:ss"):** The final and crucial step is to format the summed numerical value back into a readable time string. The [TEXT function](#) converts a number into a formatted text string. The format code "mm:ss" is particularly important:

The square brackets around explicitly tell [Google Sheets](#) to display the total number of hours, even if it exceeds 23 (i.e., multiple days). Without the brackets, hours would reset after 23, showing only the remainder within a 24-hour cycle.

**mm** represents minutes.

**ss** represents seconds.

This [custom number format](#) ensures that the accumulated time is displayed as a continuous duration rather than a time of day.

## Troubleshooting and Best Practices for Time Durations

While the provided formula is robust, ensuring smooth operation often comes down to maintaining good practices in your [spreadsheet](#). Here are some tips to avoid common pitfalls:

**Consistent Time Entry:** Always enter your time durations in a consistent and recognizable format, such as `HH:MM:SS` or `HH:MM`. Inconsistent formats (e.g., "5h 30m" mixed with "05:30") can cause the [TIMEVALUE function](#) to fail, even with [IFERROR](#).

**Avoid Text in Time Columns:** Ensure that the cells in your duration range (e.g., C2:C8) contain only valid time entries or are empty. Any descriptive text within these cells will be treated as an error, even if [IFERROR](#) prevents the formula from crashing. While [IFERROR](#) handles errors gracefully, it's best to keep your data clean.

**Verify Cell Formatting:** Although the formula outputs a text string, the source cells should ideally be formatted as "Duration" or "Time" to ensure [Google Sheets](#) correctly interprets your entries as time values from the start. This can be done via Format > Number > Duration or Time.

**Understanding the Output:** Remember that the result of this formula is a text string. While it looks like a time value, it cannot be directly used in further time-based calculations (e.g., multiplying by a rate per hour) without first converting it back to a numerical format. If you need to perform further calculations, you might sum the [TIMEVALUE](#) results without the final [TEXT function](#), and then apply custom number formatting directly to the cell where the sum resides.

## Conclusion and Additional Resources

Accurately summing [time durations](#) in [Google Sheets](#) is a common requirement for many users, from project managers tracking task times to analysts monitoring event lengths. By leveraging the combined power of [ARRAYFORMULA](#), [TIMEVALUE](#), [IFERROR](#), [SUM](#), and the [TEXT function](#) with its special " :mm:ss " [custom number format](#), you can ensure your calculations are always precise and reliable, regardless of how many hours, minutes, or seconds your total duration encompasses.

This method empowers you to handle complex time data with confidence, transforming what could be a source of frustration into a streamlined process for accurate [data analysis](#) and reporting within your [spreadsheet](#).

## Additional Resources

For further exploration and to deepen your understanding of [Google Sheets](#) functionalities, consider reviewing the official documentation for the functions discussed:

Google Sheets Help: [ARRAYFORMULA function](#)

Google Sheets Help: [TEXT function](#)

Google Sheets Help: [SUM function](#)

Google Sheets Help: [IFERROR function](#)

Google Sheets Help: [TIMEVALUE function](#)

Google Sheets Help: [Custom number formats](#)