

How to Extract the Hour from a Date & Time in Google Sheets

Authored by
stats writer

January 14, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Extract the Hour from a Date & Time in Google Sheets*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126146>

Extracting specific components from a datetime stamp is a fundamental operation in analyzing data, especially when dealing with timestamped events such as sales records, log files, or sensor readings. In the Google Sheets environment, the system stores dates and times as serial numbers, where the integer portion represents the date and the decimal portion represents the time. To isolate the time element, specifically the hour, Google Sheets provides a straightforward and powerful tool: the **HOUR** function. This function is designed to bypass the complexity of serial number arithmetic, allowing users to quickly pull the hour component from any valid datetime input.

The primary purpose of the **HOUR** function is to take a given datetime value--whether it is a direct input, a cell reference, or the result of another function--and return the hour as an integer ranging from 0 (12:00 AM) to 23 (11:00 PM). This 24-hour format ensures consistency regardless of the display formatting (AM/PM) used in the sheet. For instance, if a cell contains the datetime value 9/25/2021 3:30 PM, applying the **HOUR** function to it will yield the numerical result 15, corresponding to the 15th hour of the day. Utilizing this feature is crucial for tasks like grouping events by time of day or calculating peak operational hours, making it an indispensable tool for time-series aggregation and organizing data based on specific time periods.

Understanding the HOUR Function Syntax and Mechanics

The syntax for the HOUR function is remarkably simple, requiring only one argument. The structure is `=HOUR(datetime)`, where `datetime` must be a valid representation of time, either as a reference to a cell containing a datetime value or a time serial number itself. It is paramount that the input is recognized by Google Sheets as a numerical time value; if the input is plain text that Sheets cannot interpret as a date or time, the function may return an error or an unexpected result. Successful execution of this function relies entirely on the underlying serial representation of the time component.

When you reference a cell, for example, cell **A2**, which contains a complete datetime stamp like **1/1/2023 10:15**, the function automatically isolates the fractional part of the serial number (which represents the time of day) and calculates the corresponding hour based on a 24-hour clock. The function then returns the extracted hour as an integer. This mechanism is powerful because it handles all standard time formats defined within Google Sheets, ensuring that the extraction process is robust and reliable across different regional settings.

You can easily employ the **HOUR** function in your formulas to quickly isolate the hourly component from any datetime entry stored in your spreadsheet.

To demonstrate, if you wish to extract the hour from the datetime housed in cell **A2**, you would use the following precise formula:

=HOUR(A2)

If cell **A2** contains the specific datetime entry **1/1/2023 10:15**, this formula will correctly analyze the serial time value and return the integer **10**, representing the tenth hour of the day. This fundamental application forms the basis for all hour-based data manipulation within the spreadsheet.

Practical Example: Extracting the Hour from a Dataset

Consider a business scenario where a company maintains a detailed record of sales, capturing the exact date and time of each transaction. To determine peak performance times or to schedule staffing needs efficiently, it becomes necessary to analyze the frequency of sales strictly based on the hour of the day, disregarding the specific date or minutes. Suppose we have the following dataset organized in Google Sheets, illustrating sales activity timestamps:

	A	B	C	D
1	Datetime	Sales		
2	1/1/2023 10:15	22		
3	1/4/2023 11:59	13		
4	1/19/2023 22:40	19		
5	2/4/2023 15:45	40		
6	2/17/2023 5:16	23		
7	3/1/2023 12:34	28		
8	3/5/2023 17:01	24		
9	3/7/2023 18:20	17		
10	4/1/2023 2:19	10		
11	4/5/2023 20:04	19		
12				
13				
14				
15				
16				
17				

Our objective is to create a new column that explicitly isolates the hour from the full datetime stamps listed in Column A. This isolation allows for subsequent pivot table analysis or frequency counts, which are critical for time-based operational reporting and business intelligence.

To achieve this separation, we initiate the process by typing the required formula into cell **C2**, targeting the first datetime entry in cell **A2**. The formula remains consistent with the standard usage

of the HOUR function:

=HOUR(A2)

After inputting the formula into **C2**, the resulting value will display the hour of the first sale. To apply this extraction logic across the entire dataset, we simply utilize the fill handle feature. We must click and drag this formula down through the remaining cells in column C. This process automatically adjusts the cell references (A3, A4, etc.) ensuring that every transaction's hour is accurately extracted:

	A	B	C	D
1	Datetime	Sales	Hour	
2	1/1/2023 10:15	22	10	
3	1/4/2023 11:59	13	11	
4	1/19/2023 22:40	19	22	
5	2/4/2023 15:45	40	15	
6	2/17/2023 5:16	23	5	
7	3/1/2023 12:34	28	12	
8	3/5/2023 17:01	24	17	
9	3/7/2023 18:20	17	18	
10	4/1/2023 2:19	10	2	
11	4/5/2023 20:04	19	20	
12				
13				
14				
15				

Following this operation, Column C successfully displays only the integer hour value derived from each corresponding calculation, effectively transforming raw timestamps into discrete hourly categories ready for deeper analysis. This numerical output is essential for quantitative analyzing data methodologies.

Advanced Extraction: Combining Hours and Minutes

While extracting just the hour is often sufficient for broad analysis, situations frequently arise where greater granularity is required. If the analysis demands the display of both the hour and the minute components together--formatted in a readable time string (e.g., "15:30")--a combination of

functions must be employed. This involves using the **HOUR** function alongside the **MINUTE** function, and then linking these two numerical results using the concatenation operator (the ampersand, **&**).

The **MINUTE** function operates similarly to the **HOUR** function, accepting a datetime serial value and returning the minute component as an integer between 0 and 59. By combining these two numerical results and inserting a colon separator as a text string, we construct the desired time format. This method is particularly useful when the goal is to present the time stamp in a standard time format without relying solely on cell number formatting, or when the combined time string needs to be used as a key in another operation, such as VLOOKUP or data pivot tables.

To achieve the combined Hour:Minute output, the following formula should be used, assuming the datetime is still located in cell **A2**:

```
=HOUR(A2)&":"&MINUTE(A2)
```

It is important to understand the role of the **&** symbol in this context: it serves as the concatenate operator, joining the output of the **HOUR(A2)** calculation, the literal text string **":"**, and the output of the **MINUTE(A2)** calculation into a single cohesive text string. This process converts the numerical components into a displayable time representation.

Visualizing Combined Time Extraction

Applying the combined formula `=HOUR(A2)&":"&MINUTE(A2)` to our existing dataset demonstrates its utility in creating precise, time-specific labels. While the output is technically a text string, it provides immediate readability that standard number formats sometimes obscure or fail to handle properly when combined with other data manipulation steps.

The screenshot below illustrates the result of applying this formula across Column C:

C2 fx =HOUR(A2)&":"&MINUTE(A2)

	A	B	C	D
1	Datetime	Sales	Hour & Minute	
2	1/1/2023 10:15	22	10:15	
3	1/4/2023 11:59	13	11:59	
4	1/19/2023 22:40	19	22:40	
5	2/4/2023 15:45	40	15:45	
6	2/17/2023 5:16	23	5:16	
7	3/1/2023 12:34	28	12:34	
8	3/5/2023 17:01	24	17:1	
9	3/7/2023 18:20	17	18:20	
10	4/1/2023 2:19	10	2:19	
11	4/5/2023 20:04	19	20:4	
12				
13				
14				
15				

As is evident, Column C now contains the derived time values, showing both the hours and minutes for each datetime entry originally found in Column A. This composite string is vital for analyses requiring hour-level precision while maintaining minute-level detail. Furthermore, this technique allows for the creation of standardized time labels regardless of the source cell formatting, simplifying cross-sheet data referencing.

A crucial element of this approach is the use of the **MINUTE** function, which is responsible for isolating the minute component, and the **&** symbol, which is used to logically concatenate the hours, the colon separator, and the minutes. This structure allows for precise control over the output format and ensures a human-readable result.

Alternative Time Extraction Methods and Formatting

While the **HOUR** function is the most direct way to get the integer hour, there are alternative methods, particularly if you need to extract the time in a specific text format or handle time zone differences. One common alternative is using the **TEXT** function, which allows you to format a datetime value according to a custom pattern. For example, `=TEXT(A2, "hh")` would extract the hour in a two-digit format (e.g., "03" or "15"), and `=TEXT(A2, "hh:mm")` would produce a standard time string.

However, a key difference between **HOUR(A2)** and **TEXT(A2, "h")** is that the former returns a

numeric value (an integer from 0 to 23) suitable for mathematical operations and grouping, whereas the latter returns a text string. When performing statistical analysis, the numerical output of **HOUR** is almost always preferred, as text results cannot be directly summed or averaged without further conversion using functions like **VALUE**. The simplicity and numerical output of the HOUR function make it superior for analytical grouping.

Furthermore, for users dealing with time stamps that might span multiple days or involve time zone shifts, functions like **QUERY** or more complex combinations involving **INT** and mathematical manipulation of the serial number might be necessary. Nonetheless, for simple hourly extraction within a standard dataset, the dedicated **HOUR** function remains the fastest and most reliable option, minimizing calculation overhead and potential formatting errors.

Importance of Hourly Data for Business Analysis

The ability to accurately extract the hour from timestamps is not merely a technical exercise; it drives significant business insights. By aggregating data based on the hour of the event, analysts can identify crucial trends, such as peak server load times, optimal customer service availability windows, or periods of highest sales volume. This level of temporal granularity is often essential for operational efficiency and resource allocation, moving beyond generalized daily or weekly reporting.

For instance, a retail manager using this technique could determine that 3 PM (hour 15) consistently yields the highest volume of transactions. This insight allows them to ensure maximum staffing levels during this specific period, thereby minimizing customer wait times and maximizing conversion rates. Conversely, identifying slow hours allows for strategic resource reduction or scheduling of non-customer-facing tasks during those off-peak times. This targeted approach to time-series analysis enhances decision-making across various departments, including logistics and marketing.

In technical environments, hourly extraction can be used for monitoring system health. Analyzing log entry timestamps by hour can reveal hourly patterns in error rates or system crashes, guiding maintenance schedules or identifying potential bottlenecks that only manifest under peak hourly load. Therefore, mastering the **HOUR** and related time functions is a prerequisite for effective time-based operational analyzing data in Google Sheets.

Summary of Essential Time Extraction Functions

While this guide focuses heavily on the **HOUR** function, it is beneficial to recognize the family of functions available in Google Sheets that handle time components. These functions operate on the same underlying serial number principle and are used in conjunction to provide a comprehensive view of time data, offering flexibility for both extraction and construction of time values.

HOUR(datetime): Returns the hour component as an integer (0-23).

MINUTE(datetime): Returns the minute component as an integer (0-59).

SECOND(datetime): Returns the second component as an integer (0-59).

TIME(hour, minute, second): Creates a time serial number from provided hour, minute, and second integers, useful for manually constructing time values for comparison or input.

By mastering the integration of these functions, users can manipulate and present time data exactly as needed for sophisticated reporting and visualization. Note that complete documentation for the **HOUR** function and other related time functions can be found in the official Google Sheets support center.

ARABPSYCHOLOGY.COM