

# How to Extract the Hour from a Datetime in Google Sheets

Authored by  
**stats writer**

January 30, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Extract the Hour from a Datetime in Google Sheets*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128700>

The ability to manipulate and analyze datetime values is fundamental when working with large datasets, especially those involving transactional or log data. In Google Sheets, isolating specific components of a timestamp, such as the hour, is a common requirement for tasks like performance metric tracking, shift analysis, or identifying peak usage times. The primary tool for this extraction is the built-in HOUR function, a simple yet powerful feature that ensures data is organized and ready for deep analysis.

This function streamlines the process of converting complex timestamp strings into a manageable numerical format, specifically the hour component, presented in a standard 24-hour clock structure. Understanding how to apply the HOUR function effectively allows analysts to transform raw data into actionable insights, enabling better data visualization and reporting across various business intelligence applications. We will explore the precise mechanics of this function and demonstrate its practical application using real-world examples in a spreadsheet environment.

## Google Sheets: Extract Hour from Datetime

### The Importance of Time-Based Data Extraction

Extracting time-based components from a full datetime stamp is a crucial step in preparing data for meaningful statistical analysis. Raw timestamps often contain extraneous information (year, month, day, minutes, seconds) that can obscure patterns when focusing solely on hourly trends. By isolating the hour, data analysts can quickly categorize events, sales, or activities into discrete hourly bins, which is essential for identifying cyclical patterns or peak operational periods within the collected information.

Consider a scenario where a company needs to optimize staffing levels based on website traffic or call volume. Without extracting the hour, comparing traffic patterns across different days becomes complicated and cumbersome. The ability to generate a column dedicated solely to the hour simplifies pivot table creation and allows for accurate aggregation of metrics like average sales per hour or total users logged in during specific time frames. This process moves the data from descriptive storage to analytical readiness, maximizing its utility for strategic decision-making.

Furthermore, the 24-hour format returned by the HOUR function (0 to 23) standardizes the output, making subsequent calculations straightforward. This standardization avoids the ambiguity inherent in 12-hour formats (AM/PM), which can lead to errors when performing arithmetic operations or comparing data points generated at midnight versus noon. Proper data hygiene starts with clear, standardized components, and the HOUR function provides exactly that foundation for any time-series analysis in Google Sheets.

## Introduction to the HOUR Function in Google Sheets

The technical mechanics of the HOUR function are designed for simplicity and efficiency. Its sole purpose is to accept a valid numerical representation of a datetime--which is how Google Sheets and other spreadsheet software store dates and times internally--and return the integer representing the hour portion. This function is an integral part of the extensive suite of date and time manipulation tools available within the platform, offering immediate utility for data preprocessing.

The argument provided to the function must resolve to a valid datetime serial number. If the input cell contains a text string that Google Sheets recognizes as a date (e.g., "1/1/2023 10:15"), the system automatically converts this textual representation into its corresponding numerical value before the HOUR function operates on it. If the input is empty, non-existent, or invalidly formatted, the function will typically return an error value, such as #VALUE!, underscoring the necessity of ensuring the source data is clean and properly recognized by the spreadsheet engine.

Using the HOUR function is far superior to attempting manual text parsing or substring extraction, especially when dealing with inconsistent datetime formats across a dataset. It is reliable, fast, and handles all underlying complexities of timezone differences and format conventions internally, providing a robust, non-volatile solution for extracting the hour component quickly and accurately, regardless of the complexity of the original timestamp format.

## Syntax and Basic Application of the HOUR Function

The syntax for the HOUR function is remarkably straightforward, requiring only one argument: the datetime value itself. This mandatory argument must point to the cell containing the timestamp or be a hard-coded date value that Google Sheets can interpret correctly.

The standard syntax is:

```
=HOUR(datetime)
```

Where `datetime` is the cell reference (e.g., A2) or the hard-coded serial date value containing the timestamp you wish to analyze. This simplicity belies the function's power, allowing users to rapidly apply it across thousands of rows of data without complex nesting or error-prone conditional logic.

You can use the **HOUR** function in Google Sheets to extract the hour from a datetime.

To demonstrate this core functionality, consider a single cell containing a timestamp. If cell **A2** holds the specific date and time, the formula below extracts the hour component:

For example, you can use the following formula to extract the hour from the datetime in cell **A2**:

**=HOUR(A2)**

If cell **A2** contains **1/1/2023 10:15** then this formula would return **10**.

The resulting output is always an integer between 0 (representing 12:00 AM) and 23 (representing 11:00 PM). This structured numerical output is immediately usable for sorting, filtering, and performing advanced statistical calculations within the spreadsheet environment, establishing a crucial link between raw source data and derived analytical metrics.

### Practical Example: Isolating Hours from a Sales Dataset

To understand the practical utility of the HOUR function, let us apply it to a typical business scenario involving a sales log. Suppose a retail company tracks sales transactions, recording both the amount sold and the precise timestamp of the transaction. The primary goal is to determine which hours of the day generate the highest sales volume for optimizing employee scheduling and resource allocation planning.

Suppose we have the following dataset in Google Sheets that shows the number of sales made during various dates and times for some company:

	A	B	C	D
1	<b>Datetime</b>	<b>Sales</b>		
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				

In this structure, Column A contains the full datetime stamp, and Column B contains the numerical sales data. To effectively analyze the distribution of sales over time, we must first isolate the hour component from the verbose timestamps in Column A. This extraction is a prerequisite for creating meaningful pivot tables or frequency distributions based solely on hourly activity, allowing for quick identification of peak hours.

By dedicating a new column--Column C, for instance--to hold the extracted hour, we create a standardized key for aggregation. This derived column transforms complex time data into simple categorical data (hours 0-23). The visual representation provided in the image clearly illustrates the initial dataset setup before the application of the HOUR function, highlighting the necessity of this data preparation step.

## Step-by-Step Implementation of the Extraction Formula

Implementing the HOUR function across the sales dataset is a simple, two-step process that ensures all datetime entries are processed efficiently and accurately. This approach utilizes the relative referencing capabilities inherent in spreadsheet software.

Suppose we would like to extract the hour from each datetime in column A.

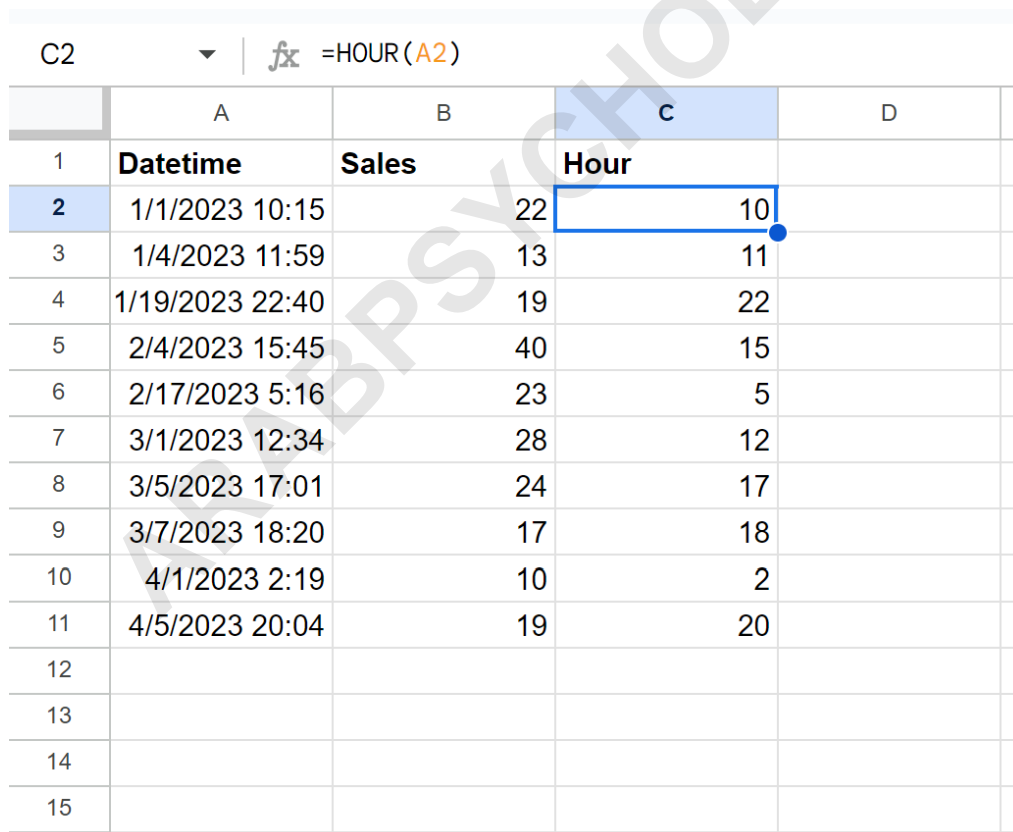
To do so, we can type the following formula into cell **C2**:

**=HOUR(A2)**

This initial action processes the very first datetime entry in the dataset (A2) and returns its corresponding hour. It is vital to confirm that the input cell (A2) is correctly formatted as a datetime or date/time structure recognized by Google Sheets to avoid common formatting errors. The function automatically handles the conversion from the underlying numerical serial number into the visible integer hour.

The major efficiency gain of this operation comes from the ability to quickly apply the formula to the rest of the column. By utilizing the click-and-drag feature (or double-clicking the fill handle in the corner of cell C2), the relative cell references adjust automatically down the column (e.g., C3 references A3, C4 references A4, and so on). This allows for instantaneous processing of the entire dataset, regardless of its size, providing immediate results.

We can then click and drag this formula down to each remaining cell in column C:



	A	B	C	D
1	<b>Datetime</b>	<b>Sales</b>	<b>Hour</b>	
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				

Column C now displays only the hour value from each datetime in column A.

## Advanced Extraction: Combining Hours and Minutes

While extracting just the hour is often sufficient for high-level aggregation, there are analytical scenarios where a finer granularity is required, necessitating the extraction of both the hours and the minutes components simultaneously. This combined output is particularly useful for measuring elapsed time between events or for displaying a more human-readable time stamp derived directly from the original full datetime field.

To achieve this combination, we must utilize the MINUTE function, which operates identically to the HOUR function but is designed to extract the minutes (an integer from 0 to 59). These two separate function outputs must then be joined together using a specialized text manipulation technique known as concatenation, inserting a colon separator to accurately format the time display.

The resultant formula leverages both functions along with the ampersand operator (&) to string the pieces together:

If you would also like to extract the minutes along with the hour, you can use the following formula instead:

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

This formula is a robust example of how multiple date/time functions can be integrated with text operators to produce customized output formats. The outcome is a text string representing the time (e.g., "10:15"), which maintains the accuracy derived from the original datetime stamp while presenting it in a standardized, easily digestible time format.

## Understanding the Concatenation Process (&)

The key to successfully combining the hours and minutes into a single cell lies in understanding the concatenation operator, represented by the ampersand symbol (&) in Google Sheets formulas. This operator is fundamentally a way to join multiple values--whether they are numbers resulting from functions, static text strings, or cell references--into one single output string, making complex display formats possible.

In the formula `=HOUR(A2)&":"&MINUTE(A2)`, the processing sequence is strictly ordered:

The `HOUR(A2)` function executes first, extracting the numerical hour (e.g., 10).

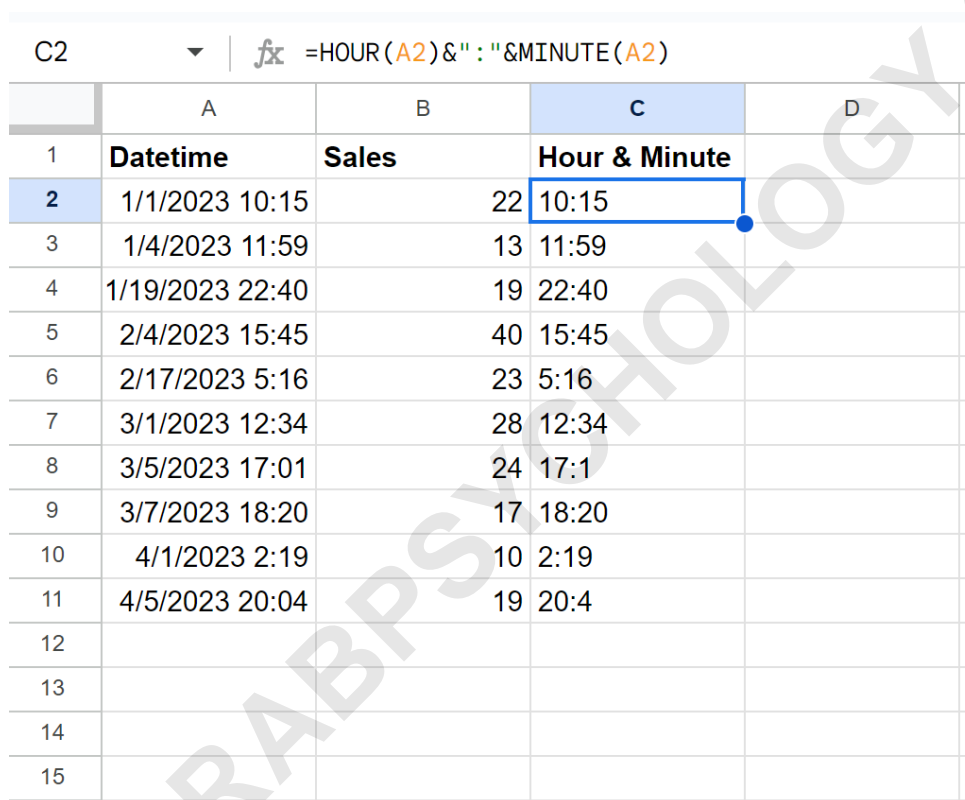
This number is immediately joined (concatenated) to the static text string `":"`, resulting in `"10:"`.

The resulting string is then joined (concatenated) to the output of the `MINUTE(A2)` function, which extracts the numerical minutes (e.g., 15).

The final output presented in the cell is the complete text string "10:15".

It is important to note a critical distinction: when hours and minutes are combined using concatenation, the resulting cell content is stored as a **text string**, not a numerical time value. While this format is highly readable and suitable for display, reporting, and sorting, it may require further conversion functions (like `TIMEVALUE`) if subsequent direct mathematical calculations involving elapsed time or time difference are needed. However, for simple data logging and display purposes, this method, utilizing the `HOUR` and `MINUTE` functions, is the most effective way to combine and format the time components.

The following screenshot shows how to use this formula in practice:



	A	B	C	D
1	<b>Datetime</b>	<b>Sales</b>	<b>Hour &amp; Minute</b>	
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				

Column C now contains both the hours and minutes for each datetime in column A.

Note that in this formula we used the `MINUTE` function to extract the minutes from the datetime and the `&` symbol to concatenate the hours and minutes.

**Note:** You can find the complete documentation for the `HOUR` function in Google Sheets .

## Further Resources and Time Analysis Techniques

The successful extraction and combination of both time components demonstrates advanced data

manipulation capabilities within Google Sheets, moving beyond basic function application toward customized data output tailored to specific analytical needs. Mastering these techniques is fundamental for anyone working with time-series data.

For those looking to expand their knowledge of working with complex datetime fields, there are several related functions and tutorials that cover common advanced tasks in Google Sheets, allowing for comprehensive time analysis:

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

Methods for calculating the precise time difference between two timestamps, utilizing functions like `DATEDIF` or simple subtraction on serial numbers.

Techniques for converting text strings that represent time (like the concatenated output) back into numerical time values for mathematical use.

Advanced applications of conditional formatting based on time criteria, such as highlighting sales made during peak hours.

Utilizing the `ARRAYFORMULA` wrapper to apply the `HOUR` function efficiently across an entire column without needing to drag the formula down manually.

Understanding these time manipulation methods ensures that your data preparation process is both robust and scalable for datasets of any size.