

Excel: Extract Hour from Datetime

Authored by
Mohammed loot

November 14, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Excel: Extract Hour from Datetime*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=845>

Introduction to Datetime Handling in Microsoft Excel

Microsoft Excel is an indispensable tool for data management and analysis, frequently requiring users to process complex time-series data. When dealing with records that contain full [datetime](#) stamps--which combine both the date and the precise time of an event--it is often necessary to isolate specific components for deeper analysis.

For instance, an analyst might need to determine how many transactions occur during specific hours of the day, regardless of the date. To accomplish this powerful data segmentation, [Microsoft Excel](#) provides a dedicated built-in function: the **HOUR** function. This function is designed to efficiently parse a standard [datetime](#) serial number and return only the integer value representing the hour of the day.

The ability to quickly extract the hour is fundamental when performing tasks like operational efficiency studies, calculating peak service times, or setting up complex scheduling systems based on temporal patterns.

While a [datetime](#) entry in Excel might appear straightforward (e.g., 2023-10-26 14:30:00), the underlying value is stored as a serial number, where the integer portion represents the date and the decimal portion represents the time. The **HOUR** function performs the necessary mathematical conversion to translate that decimal time component into a recognized hour value between 0 (midnight) and 23 (11 PM).

This detailed guide will explore the syntax, practical application, and advanced usage of the **HOUR** function, ensuring you can confidently manipulate temporal data within your spreadsheets. For example, if you have a full [datetime](#) value located in cell **A2**, you can use the formula detailed below to immediately extract the hour component. This simple yet effective tool streamlines data preparation, making subsequent analysis much cleaner and more focused.

=HOUR(A2)

Understanding the Core HOUR Function Syntax

The [HOUR function](#) in Excel is categorized as a Date/Time function and adheres to a very simple syntax, requiring only one argument. This simplicity contributes to its widespread utility, as it drastically reduces the complexity typically associated with parsing time strings manually. The primary purpose is to return the hour component of a time value as an integer, ranging from 0 (12:00 A.M.) to 23 (11:00 P.M.). Understanding its structure is the first step toward effective implementation in large datasets.

The syntax is defined as **=HOUR(serial_number)**. The only required argument, **serial_number**, refers to the date and time that contains the hour you want to find. This argument can be supplied in several ways, making the function highly flexible. It can be a direct cell reference containing the [datetime](#) stamp (as demonstrated in our main example), a text string representing a time (e.g., "10:45 AM"), or the result of another formula that returns a valid time value (such as the **NOW** or **TIMEVALUE** functions). It is crucial that the input be recognizable by Excel as a valid date or time; otherwise, the function will likely return the **#VALUE!** error.

When the function successfully processes the input, the output is always a whole number. This integer output is highly advantageous because it allows the extracted hour to be used immediately in mathematical calculations, logical tests (e.g., **IF** statements), or as criteria in advanced filtering operations and

[Pivot Tables](#). For instance, if the input cell **A2** contains the time 16:25:30 (4:25 PM), the formula **=HOUR(A2)** will return the integer **16**. This standardized output ensures consistency when analyzing large batches of time data, allowing analysts to quickly group and summarize events by the hour of occurrence.

Step-by-Step Example: Isolating the Hour Value

To illustrate the practical application of the

[HOUR function](#), let us consider a common business scenario involving sales tracking. Suppose a company has compiled a comprehensive dataset that records the exact moment each sale was executed, shown here in column A. Analyzing the hourly distribution of these sales can provide critical insights into customer traffic patterns and staffing needs.

Our hypothetical dataset, displayed below, contains various

[datetime](#) stamps in column A and the corresponding sales figures in column B. Our objective is to generate a new column, column C, which contains only the hour component derived from the timestamp in column A. This transformation allows us to pivot the data later to identify peak selling hours.

	A	B	C	D	E
1	Datetime	Sales			
2	1/1/2023 5:15	13			
3	1/4/2023 8:14	8			
4	1/19/2023 16:34	10			
5	2/4/2023 20:22	4			
6	2/5/2023 21:14	9			
7	2/19/2023 23:30	16			
8	2/24/2023 4:13	8			
9	3/1/2023 17:55	4			
10	3/3/2023 5:15	13			
11	5/5/2023 9:18	7			
12					
13					
14					
15					
16					
17					
18					

To perform this extraction, we initiate the process by typing the following simple formula into cell **C2**, targeting the first [datetime](#) entry in the dataset. This formula references cell **A2**, where the initial timestamp is located, and instructs Excel to return the hour value as a number.

=HOUR(A2)

After entering the formula into **C2**, we can efficiently apply this logic to the entire dataset. By utilizing the fill handle--the small square located in the bottom-right corner of cell **C2**--and dragging the formula down through the remaining cells in column C, Excel automatically adjusts the cell references (e.g., C3 references A3, C4 references A4, and so on). The result, as shown in the subsequent image, is a perfectly cleaned column of data containing only the integer hour extracted from each original [datetime](#) stamp. Column C now displays values like 15, 12, 18, etc., representing 3 PM, 12 PM, and 6 PM respectively, ready for subsequent data aggregation or visualization steps.

	A	B	C	D	E
1	Datetime	Sales	Hour		
2	1/1/2023 5:15	13	5		
3	1/4/2023 8:14	8	8		
4	1/19/2023 16:34	10	16		
5	2/4/2023 20:22	4	20		
6	2/5/2023 21:14	9	21		
7	2/19/2023 23:30	16	23		
8	2/24/2023 4:13	8	4		
9	3/1/2023 17:55	4	17		
10	3/3/2023 5:15	13	5		
11	5/5/2023 9:18	7	9		
12					
13					
14					
15					
16					
17					

Handling Time Formatting and Data Types

While the

[HOUR function](#) is designed to return an integer between 0 and 23, users occasionally encounter unexpected results related to number formatting. When working with temporal data in [Microsoft Excel](#), it is critical to remember that date and time values are fundamentally numerical serial numbers. If the source cell (e.g., A2) was previously formatted as a date or time, Excel might sometimes automatically apply that same custom format to the result cell (e.g., C2), even though the calculated result is just a simple number (the hour).

If, upon using the formula, the output in column C appears as a seemingly random date, a decimal number, or a time value rather than the expected integer (e.g., 15 for 3 PM), the issue is almost always the cell formatting. For example, if the result cell is formatted as "Time," the number 15 might be displayed incorrectly, or the number 14 (representing 2 PM) might be displayed as 12:00:00 AM if the cell is formatted as a date. To rectify this, simply select the output column (Column C in our example) and navigate to the Home tab. In the Number group, change the format from Date, Time, or Custom to **General** or **Number**. This forces Excel to display the calculated integer value clearly.

Furthermore, understanding Excel's handling of time is essential for troubleshooting. Excel calculates time as a fraction of a 24-hour day. Midday (12:00 PM) is represented internally by the decimal 0.5, and 6:00 PM is 0.75. The **HOUR** function performs the reverse calculation, taking that decimal fraction and converting it back into the integer hour. If the input data is stored merely as a text string that looks like time (and not recognized as a valid time serial number), the [HOUR function](#) may fail. Using the **TIMEVALUE** function to convert the text to a serial number before passing it to the **HOUR** function is a robust solution for ensuring data validity and preventing formatting errors.

Advanced Extraction: Combining Hour and Minute

While extracting only the hour is sufficient for many analyses, situations often arise where greater granularity is required, necessitating the extraction of both the hour and the minute components, often presented in a standard HH:MM format. Achieving this requires combining the [HOUR function](#) with its counterpart, the **MINUTE** function, and using a technique called [text concatenation](#).

The **MINUTE** function operates identically to the **HOUR** function, taking a serial number as its argument and returning the minute component as an integer between 0 and 59. To combine the hour and minute into a single, readable text string, we utilize the ampersand symbol (&), which serves as the [concatenation](#) operator in Excel. This operator joins separate text strings or values into one continuous string. We also need to insert a colon (":") as a literal text string between the two numerical components to correctly format the time.

The complete formula to extract the hour and minute from cell **A2**, formatted as HH:MM, is as follows:

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

In this formula, **HOUR(A2)** retrieves the hour, and **MINUTE(A2)** retrieves the minutes. The **&":"&** inserts the colon separator, effectively joining the three elements. The result is a text string (e.g., "15:30" instead of the numerical time value). The subsequent screenshot demonstrates how this advanced formula is applied across the dataset, yielding the concise time string necessary for more detailed time-based grouping or reporting.

	A	B	C	D	E
1	Datetime	Sales	Hour & Minute		
2	1/1/2023 5:15	13	5:15		
3	1/4/2023 8:14	8	8:14		
4	1/19/2023 16:34	10	16:34		
5	2/4/2023 20:22	4	20:22		
6	2/5/2023 21:14	9	21:14		
7	2/19/2023 23:30	16	23:30		
8	2/24/2023 4:13	8	4:13		
9	3/1/2023 17:55	4	17:55		
10	3/3/2023 5:15	13	5:15		
11	5/5/2023 9:18	7	9:18		
12					
13					
14					
15					
16					

It is important to note that because the **&** symbol is used, the resulting output is a **text string**, not a numerical time value. While this is perfect for display and grouping purposes, if you intend to perform further mathematical calculations on the resulting time (e.g., calculating the duration between two extracted times), you might need to use the **TIMEVALUE** function on the concatenated output to convert it back into a valid numerical time serial number.

Practical Applications and Use Cases

The extraction of hourly and minute components using the [HOUR function](#) and related time functions opens up a wide array of practical analytical possibilities within

[Microsoft Excel](#). These techniques move beyond simple data cleaning and enable sophisticated business intelligence tasks. The primary use case is transforming raw, timestamped data into actionable time-based categories.

One significant application is **Temporal Trend Analysis**. By extracting the hour, analysts can easily create frequency distributions showing how many events (sales, website visits, customer service calls) occur during each of the 24 hours of the day. This data is invaluable for optimizing resource allocation. For example, a retail manager can use this information to schedule maximum staff coverage during the three busiest hours identified by the analysis. This requires linking the extracted hour column to a

[Pivot Table](#) and counting the occurrences.

Another powerful use is in **Conditional Formatting and Logic**. The numerical output of the **HOUR** function can be directly fed into **IF** statements or conditional formatting rules.

Example 1: Identify rush hour events: `=IF(HOUR(A2)>=16, "Rush Hour", "Normal")`.

Example 2: Apply a specific color highlight to data entries that occurred between 9 AM and 5 PM using a conditional formatting rule based on the extracted hour.

Finally, time extraction is fundamental for **Creating Time Bins**. Often, simply knowing the exact hour is too granular. Analysts prefer to group data into broader time bins, such as "Morning" (6-11), "Afternoon" (12-16), and "Evening" (17-23). The extracted numerical hour makes it trivial to create these custom bins using nested **IF** statements or the **CHOOSE** function, significantly simplifying the interpretation of complex time-series data.

Summary and Additional Resources

Mastering the extraction of temporal components from [datetime](#) stamps is essential for any advanced [Microsoft Excel](#) user performing quantitative analysis. The [HOUR function](#) provides a robust and simple mechanism for isolating the hour, returning a clean integer value ready for filtering, grouping, or complex logical testing. Furthermore, combining the **HOUR** function with the [MINUTE function](#) and the [concatenation](#) operator allows for the creation of customized time strings (HH:MM) that offer greater detail while remaining highly readable.

To ensure seamless execution, always verify that your source data is recognized as a valid date or time serial number, and remember to adjust the cell formatting of your output column to **General** or **Number** if the results appear corrupted by residual date formats.

For those requiring similar extractions for other components of a [datetime](#) stamp, Excel provides a comprehensive set of corresponding functions. These tools allow for full deconstruction of timestamps into their constituent parts, facilitating virtually any temporal analysis requirement.

Additional Resources

The following functions and tutorials explain how to perform other common operations in Excel related to temporal data manipulation:

DAY Function: Extracts the day of the month (1-31) from a date serial number.

MONTH Function: Extracts the month (1-12) from a date serial number.

YEAR Function: Extracts the four-digit year from a date serial number.

SECOND Function: Extracts the seconds (0-59) from a time serial number.

TIMEVALUE Function: Converts a time represented by a text string to an Excel time serial number.

By leveraging these powerful, dedicated functions, users can transform overwhelming timestamp datasets into structured, meaningful information, driving better decision-making based on when events truly occur.