

# Learning to Use the IF Function with WEEKDAY in Excel

Authored by  
**Mohammed loot**

November 10, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Use the IF Function with WEEKDAY in Excel*.  
PSYCHOLOGICAL STATISTICS. Retrieved from  
<https://statistics.arabpsychology.com/?p=15438>

Excel is an essential tool for data analysis, and frequently, analytical tasks require conditional logic based on specific dates. This guide demonstrates how to combine the power of the [IF function](#) with the robust date classification provided by the [WEEKDAY function](#) to create formulas that automatically check if a date falls on a specific day or within a broader category like "Weekday" or "Weekend."

## Introduction to Conditional Date Logic in Excel

Often, analysis requires performing actions or generating specific outputs based on whether a date falls on a specific day, such as a Sunday, or whether it is part of a standard workweek or a weekend. Achieving this requires the sophisticated integration of conditional statements, provided primarily by the [IF function](#), with specialized date functions like the **WEEKDAY** function. This combination unlocks powerful [Boolean logic](#) capabilities directly applicable to temporal data sets, essential for everything from payroll calculation to project timeline management.

Understanding how [Excel](#) handles dates is foundational to mastering these formulas. Internally, Excel stores dates not as traditional calendar entries but as sequential [serial numbers](#), starting from January 1, 1900, which is represented as the number 1. The conditional execution of tasks--such as calculating overtime pay only for weekend shifts or scheduling events based on specific days--relies on converting these serial date values into meaningful identifiers (like the day of the week) and then testing those identifiers against predefined criteria.

The following discussion details two primary methods for combining the conditional power of the **IF** function with the robust date classification of the **WEEKDAY** function. These integrated formulas allow users to create automated checks that return specific text outputs based on the temporal properties of any given date cell. We will examine the syntax, explore the logic behind each component, and apply these concepts through practical examples using a sample data set of dates. These examples demonstrate how to categorize dates efficiently, offering clear and immediate feedback on whether a date meets the criteria.

## Understanding the Excel WEEKDAY Function

Before implementing complex conditional statements, it is imperative to fully grasp the operation of the [WEEKDAY function](#). This function is designed to return an integer representing the day of the week for a specified date. Crucially, the return value mapping is dependent on the optional second argument, the *return\_type*, which dictates where the week begins. By default, and in the absence of a specified return type (or using return type 1), the standard convention is that Sunday is represented by the integer 1, and the sequence proceeds through Saturday, which is represented by the integer 7.

The standard syntax for the **WEEKDAY** function is simply `=WEEKDAY(serial_number, )`. If we

refer to a date in cell **A2**, the formula `WEEKDAY(A2)` will return a number between 1 and 7. For instance, if the date in **A2** is a Monday, **WEEKDAY** (using the default type 1) will return 2. This integer output is the key element that we feed into the logical test argument of the **IF** function. When constructing your formulas, always confirm the return type being used, as selecting return type 2, for example, assigns Monday the value 1 and Sunday the value 7, a common source of error for beginners.

The note below summarizes the default return convention, which is used throughout these examples. This default setup is standard for many calendar applications and is often the simplest approach when checking for specific weekend days. It is this predictable numeric output that allows for precise conditional testing. Once the **WEEKDAY** function successfully translates a complex date into a simple integer, the logical testing provided by **IF** can begin its evaluation.

#### **Formula 1: Create IF Function to Check if Date is on Specific Day of Week**

**=IF(WEEKDAY(A2)=1, "Sunday", "Not Sunday")**

This particular formula checks if the day of the week for the date in cell **A2** is on Sunday and returns either "Sunday" or "Not Sunday" as a result. This relies entirely on the default **WEEKDAY** return type, where 1 signifies Sunday.

#### **Formula 2: Create IF Function to Check if Date is on Weekday or Weekend**

**=IF(AND(WEEKDAY(A2)>1, WEEKDAY(A2)<7), "Weekday", "Weekend")**

This particular formula checks if the day of the week for the date in cell **A2** is on a weekday (Monday-Friday) or weekend (Saturday-Sunday) and returns "Weekday" or "Weekend" as a result. This structure utilizes the [AND function](#) to ensure the date is neither 1 nor 7.

The following examples show how to use each formula in practice with the following column of dates in [Excel](#):

	A	B	C	D	E
1	<b>Date</b>				
2	1/1/2023				
3	2/1/2023				
4	3/1/2023				
5	4/1/2023				
6	5/1/2023				
7	6/1/2023				
8	7/1/2023				
9	8/1/2023				
10	9/1/2023				
11	10/1/2023				
12	11/1/2023				
13	12/1/2023				
14					
15					
16					
17					

### Example 1: Identifying a Specific Day of Week

The first common application is determining if a given date falls on one specific day of the week, such as Sunday. This requires a direct comparison between the integer output of the **WEEKDAY** function and the corresponding numeric identifier for the target day. If the date in question is located in cell **A2**, and we are searching for Sunday, which corresponds to the integer 1, our **IF** function must test for equality with 1. This method provides a clear, binary output for rapid data categorization, returning "Sunday" if true and "Not Sunday" if false.

We can type the following formula into cell **B2** to check if the date in cell **A2** is on a Sunday, replicating Formula 1 precisely. This formula immediately evaluates the date's serial number, translates it into the day index, and performs the logical comparison against the target value of 1.

### Example 1: Create IF Function to Check if Date is on Specific Day of Week

**=IF(WEEKDAY(A2)=1, "Sunday", "Not Sunday")**

We can then click and drag this formula down to each remaining cell in column B. This action utilizes Excel's relative referencing capability, ensuring that as the formula moves down the

column, the cell reference updates from **A2** to **A3**, **A4**, and so on, performing the correct calculation for every date in the dataset without manual intervention.

	A	B	C	D	E	F	G
1	<b>Date</b>	<b>Sunday?</b>					
2	1/1/2023	Sunday					
3	2/1/2023	Not Sunday					
4	3/1/2023	Not Sunday					
5	4/1/2023	Not Sunday					
6	5/1/2023	Not Sunday					
7	6/1/2023	Not Sunday					
8	7/1/2023	Not Sunday					
9	8/1/2023	Not Sunday					
10	9/1/2023	Not Sunday					
11	10/1/2023	Sunday					
12	11/1/2023	Not Sunday					
13	12/1/2023	Not Sunday					
14							
15							
16							
17							
18							

The formula returns either "Sunday" or "Not Sunday" to indicate if each date in column A is on a Sunday or not. This validates the effectiveness of the simple IF and [WEEKDAY function](#) combination for single-day identification. If you require classification for another day, such as identifying all Mondays, you simply substitute the integer 1 for the corresponding number (2 for Monday, 3 for Tuesday, etc.).

**Note:** The **WEEKDAY** function in Excel returns an integer value between 1 (Sunday) and 7 (Saturday) to indicate the day of the week, using the default return type (1).

If you would instead like to check if a date is on a different day of the week, simply change the 1 to a different value. For instance, testing for Monday (value 2) requires the following modification:

**=IF(WEEKDAY(A2)=2, "Monday", "Not Monday")**

## Example 2: Distinguishing Weekdays from Weekends

A more advanced, and highly useful, application involves classifying dates into one of two broader

categories: Weekday (Monday through Friday) or Weekend (Saturday and Sunday). This requires checking for a range of numeric values, necessitating the introduction of compound logical operators, specifically the [AND function](#), nested within the primary **IF** statement.

Using the default **WEEKDAY** return type (where Sunday=1 and Saturday=7), a date qualifies as a Weekday if its corresponding integer is greater than 1 AND less than 7. This defines the inclusive set {2, 3, 4, 5, 6}. The [AND function](#) is perfectly suited to verify that both these conditions are simultaneously true. If both conditions within the **AND** are met, the **AND** function returns **TRUE**, which triggers the "Weekday" output of the outer [IF function](#).

We can type the following formula into cell **B2** to check if the date in cell **A2** is on a weekday or a weekend. The formula ensures that only values between 2 and 6 (inclusive) result in a "Weekday" classification, while the endpoints (1 and 7) are categorized as "Weekend."

## Example 2: Create IF Function to Check if Date is on Weekday or Weekend

```
=IF(AND(WEEKDAY(A2)>1, WEEKDAY(A2)<7), "Weekday", "Weekend")
```

We can then click and drag this formula down to each remaining cell in column B. This action instantly classifies the entire range of dates based on the complex [Boolean logic](#) established by the nested functions. The resulting spreadsheet clearly delineates which dates fall into the "Weekday" category and which fall into the "Weekend" category, providing immediate analytical insight.

	A	B	C	D	E	F	G	H	I
1	<b>Date</b>								
2	1/1/2023	Weekend							
3	2/1/2023	Weekday							
4	3/1/2023	Weekday							
5	4/1/2023	Weekend							
6	5/1/2023	Weekday							
7	6/1/2023	Weekday							
8	7/1/2023	Weekend							
9	8/1/2023	Weekday							
10	9/1/2023	Weekday							
11	10/1/2023	Weekend							
12	11/1/2023	Weekday							
13	12/1/2023	Weekday							
14									
15									
16									
17									

The formula returns either "Weekend" or "Weekday" to indicate if each date in column A is on a weekday (Monday-Friday) or weekend (Saturday-Sunday). This method is far superior to attempting multiple nested **IF** functions to check for specific day numbers, offering a clean, scalable solution using range comparison.

**Note:** We used the [AND function](#) with the **WEEKDAY** function in [Excel](#) to check if a cell had a weekday integer value equal to 2, 3, 4, 5 or 6 to indicate that it was a weekday.

## Advanced Considerations and Further Resources

While the examples focused on returning simple text strings, the power of combining **IF** and **WEEKDAY** extends far beyond basic classification. The "value if true" and "value if false" arguments of the **IF** function can be replaced by complex calculations, formulas, or cell references, allowing for dynamic data manipulation based on the day of the week. For example, you might adjust a price by 10% if the date is a weekend or return a specific project code if the day is Monday.

Furthermore, flexibility is achieved through the optional *return\_type* argument in the **WEEKDAY** function. If your corporate standard or regional calendar defines the start of the week differently--say, Monday as the first day (return type 2, where Monday=1 and Sunday=7)--you must adjust your comparisons accordingly. For instance, to check for a weekend (Saturday=6, Sunday=7) using return type 2, the logical test would need to use the **OR** function: `OR(WEEKDAY(A2, 2)=6,`

`WEEKDAY(A2, 2)=7`). Understanding these return types is essential for global data consistency.

Another powerful customization involves utilizing the **OR** function instead of **AND** when checking for multiple specific days. For instance, if you need to identify both Mondays (2) and Fridays (6), you would use: `=IF(OR(WEEKDAY(A2)=2, WEEKDAY(A2)=6), "M/F", "Other")`. This demonstrates how conditional date logic can be precisely tailored to filter and categorize data based on highly specific business or analytical requirements.

## Additional Resources

Mastering conditional functions and date arithmetic is crucial for advanced data handling in Excel. The following tutorials explain how to perform other common tasks in [Excel](#), building upon the foundational knowledge of date and conditional processing:

Tutorial on using the **EOMONTH** function for month-end calculations.

Guide to nesting multiple **IF** statements for complex, multi-criteria evaluations.

Explanation of the **NETWORKDAYS** function for calculating business days between two dates.