

Excel: A Formula for MID From Right

Authored by
Mohammed loot

October 27, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Excel: A Formula for MID From Right*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=3987>

The **MID function** in **Excel** is an essential tool utilized for precise text extraction. Its primary purpose is to isolate a specific number of **characters** from a given **string** of text, operating based on a designated starting position that is always counted from the **left side**. This functionality is ideal when dealing with structured data where the desired information consistently begins at a known offset from the start of the cell content.

However, many real-world datasets present complexities that the standard **MID function** cannot directly resolve. A common requirement is extracting information whose starting point must be determined relative to the **right side** of the string. For instance, you might need to extract a unique identifier that is always located a fixed number of characters away from the end of the text, irrespective of the overall length of the string.

To effectively address this limitation and enable dynamic text extraction that begins from the **right side**, we must leverage a sophisticated combination of standard **Excel functions**. The formula below is a powerful construct that replicates the core behavior of the **MID function** while providing the crucial flexibility of specifying a starting position counted from the **right side** of the text string:

```
=RIGHT(REPLACE(A1, LEN(A1)-start_from_right+2, LEN(A1),""), num_characters)
```

Understanding the Standard MID Function in Excel

The core functionality of **Excel's MID function** is centered on extracting a substring based on a left-to-right count. This function is fundamental to text manipulation tasks and requires three specific arguments to operate correctly.

The syntax is defined as `MID(text, start_num, num_chars)`, where each parameter serves a distinct role:

text is the original **string** from which characters will be extracted.

start_num specifies the position of the very first character intended for extraction, counting exclusively from the leftmost character of the string.

num_chars dictates the total count of characters to be extracted, starting from the specified **start_num** position.

For illustration, consider the formula `MID("Development", 6, 4)`. Since the count starts from the left, the sixth character is 'p', and extracting four characters results in the output "pmen". This direct, left-based counting mechanism makes the **MID function** highly valuable for data that adheres to a consistent format where the target segment always occupies a fixed position relative to the beginning of the string.

The Necessity for Right-Based Text Extraction

Although the standard [MID function](#) is robust, its limitation lies in the lack of a built-in parameter to define the starting point from the right end of the string. This becomes a significant obstacle when processing variable-length text strings where the critical data element consistently appears a specific number of characters *before the end*.

Attempting to use the standard function in these scenarios would necessitate calculating the equivalent left-based starting position for every single entry using the [LEN function](#), which is highly inefficient, tedious, and susceptible to errors when dealing with large datasets.

Common business and technical examples requiring right-based extraction include extracting version numbers that are appended to the end of product names, isolating specific date codes embedded just before a file extension, or pulling out trailing numerical sequences in log identifiers. In all these cases, a dynamic solution that can efficiently count backward from the end of the string offers superior efficiency and ensures the formula remains accurate even if the total string length changes.

Introducing the Advanced Formula for Dynamic Extraction

To successfully implement the "MID from right" functionality, we must combine three core [Excel functions](#) into a single powerful expression: [REPLACE](#), [LEN](#), and [RIGHT](#). This combination is designed to first isolate the desired segment by truncating the text and then accurately extract the final characters.

This composite formula requires two critical custom parameters tailored to your extraction needs:

start_from_right: This determines the precise starting point of your desired segment, counted backward from the final character of the string. For example, if this value is 5, the extraction will begin at the fifth character from the end.

num_characters: This specifies the exact length of the resulting substring you wish to retrieve, starting from the calculated position.

Let us consider a practical application: extracting four middle characters from the text located in [cell A1](#), with the extraction mandated to start three positions from the right end of the text. The required dynamic formula is constructed as follows:

```
=RIGHT(REPLACE(A1,LEN(A1)-3+2,LEN(A1),""), 4)
```

Step-by-Step Breakdown of the Formula's Logic

To gain mastery over this function, it is necessary to comprehend the internal mechanism and the purpose of each nested function. The formula operates in a sequential manner, moving from the innermost calculations outward.

`LEN(A1)`: This initial step calculates the total length of the **string** contained within **cell A1**. Knowing the total length is paramount, as it provides the basis for all positional calculations relative to the end of the string.

`LEN(A1)-start_from_right+2`: This complex expression calculates the precise **left-based starting position** required by the **REPLACE function**.

`LEN(A1) - start_from_right` determines the position of the character immediately preceding the desired starting point, counted from the left.

The addition of `+2` is crucial. Since the **REPLACE function** must begin its operation *after* the characters we intend to retain (including the segment to be extracted), the addition of `+2` correctly offsets the position. This action ensures that the replacement process isolates the string segment we need.

`REPLACE(A1, LEN(A1)-start_from_right+2, LEN(A1), "")`: The **REPLACE function** performs the main manipulation of the original string:

It starts replacing characters from the calculated position determined in the previous step.

It instructs Excel to replace `LEN(A1)` characters--which guarantees replacement through to the end of the string--with an empty string (`" "`).

The output of this step is a shortened string that contains only the characters from the beginning up to, and including, the specific segment required for extraction. All irrelevant characters trailing the target segment are efficiently removed.

`RIGHT(..., num_characters)`: In the final stage, the **RIGHT function** processes the truncated string generated by **REPLACE**. Since the string has been precisely cut, the **RIGHT function** simply extracts the last `num_characters` from this modified text. This extracted segment corresponds exactly to the characters that started at the `start_from_right` position in the original string.

This systematic flow of execution ensures that, despite Excel's native text functions primarily operating left-to-right, we achieve a reliable right-to-left extraction for complex **data manipulation** requirements.

Example: Implementing MID From Right in Excel

To demonstrate the practical effectiveness of this advanced formula, we will apply it to a common scenario involving text data in an [Excel spreadsheet](#). Suppose we are working with a list of team names and need to extract a specific segment of each name based on a count from the right side.

We will utilize the following sample dataset, which contains a list of team names in Column A:

	A	B	C	D	E	F
1	Team					
2	Mavericks					
3	Rockets					
4	Hornets					
5	Pacers					
6	Raptors					
7	Thunder					
8	Pelicans					
9	Nuggets					
10	Timberwolves					
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Practical Application: Extracting Substrings from Team Names

Our objective is clear: for every team name, we must extract a segment consisting of exactly 4 characters, with the starting point for this extraction being the 3rd position counted from the right end of the name. If we take "Mavericks" as an example, the third character from the right is 'c'. Starting at 'c' and extracting four characters backward should yield the result "eric".

To accomplish this dynamic extraction across the dataset, enter the comprehensive formula into [cell B2](#), which corresponds to the first team name entry in [cell A2](#):

=RIGHT(REPLACE(A2,LEN(A2)-3+2,LEN(A2),""), 4)

Once the formula is correctly entered in [cell B2](#), the next step is to efficiently apply it to the remainder of your list. Select [cell B2](#) and use the fill handle--the small square located at the bottom-right corner--to click and drag the formula down Column B. This action will automatically populate the remaining cells, adjusting the cell references (e.g., from A2 to A3, A4, and so on) as needed, ensuring consistent and accurate application across the entire range.

	A	B	C	D	E
1	Team	Middle 4 characters starting 3 from the right			
2	Mavericks	eric			
3	Rockets	ocke			
4	Hornets	orne			
5	Pacers	Pace			
6	Raptors	apto			
7	Thunder	hund			
8	Pelicans	lica			
9	Nuggets	ugge			
10	Timberwolves	wolv			
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

Interpreting the Results and Verifying Accuracy

After successfully dragging the formula down, Column B will display the extracted 4-character segments for every team name. The output confirms the formula's ability to maintain a consistent extraction logic (starting 3 positions from the right and extracting 4 characters) even when applied to strings of varying lengths.

To confirm the precision of the logic, let us re-examine the string "Mavericks":

Original Team Name: **Mavericks**

The 3rd position when counting backward from the right is the character '**c**' (Maver**ics**).

Starting from 'c' and extracting 4 characters to the left yields: M**averics**.

The resulting extracted string, "**eric**", precisely matches the output generated by the formula in the spreadsheet. This detailed verification underscores the utility of this combined function for complex and dynamic [data manipulation](#) tasks within [Excel](#). The formula systematically repeats this robust process for every entry in the team column, guaranteeing accurate and consistent results throughout your dataset.

Additional Resources and Further Exploration

For continued professional development and to further enhance your mastery of [Excel's](#) advanced capabilities, exploring other functions related to text analysis and [data manipulation](#) is highly recommended. Understanding how to combine functions effectively is the key to solving complex data challenges.