

 [Collapse All](#)  [Code: All](#)

SEARCH Function (DAX)

[See Also](#) [Send Feedback](#)

Returns the number of the character at which a specific character or text string is first found, reading left to right. Search is case-sensitive.

Syntax

```
SEARCH(<find_text>, <within_text>, [start_num])
```

Parameters

Term	Definition
find_text	The text that you want to find.
within_text	The text in which you want to search for find_text , or a column containing text.
start_num	(optional) The character position in within_text at which you want to start searching. If omitted, 1.

Property Value/Return Value

The number of the starting position of the first text string from the first character of the second text string.

Remarks

By using this function, you can locate one text string within a second text string, and return the position where the first string starts.

You can use the SEARCH function to determine the location of a character or text string within another text string, and then use the MID function to return the text, or use the REPLACE function to change the text.

If the **find_text** cannot be found in **within_text**, the formula returns an error. This behavior is like Excel, which returns #VALUE if the substring is not found. Nulls in **within_text** will be interpreted as an empty string in this context.

Example: Search within a String

Description

The following formula finds the position of the letter "n" in the word "printer".

Code

 [Copy Code](#)

```
=SEARCH("n", "printer")
```

Comments

The formula returns 4 because "n" is the fourth character in the word "printer."

Example: Search within a Column

Description

You can use a column reference as an argument to SEARCH. The following formula finds the position of the character "-" (hyphen) in the column, [PostalCode].

Code

 [Copy Code](#)

```
=SEARCH("-", [PostalCode])
```

Comments

The return result is a column of numbers, indicating the index position of the hyphen.

Example: Error-Handling with SEARCH

Description

The formula in the preceding example will fail if the search string is not found in every row of the source column. Therefore, the next example demonstrates how to use IFERROR with the SEARCH function, to ensure that a valid result is returned for every row.

The following formula finds the position of the character "-" within the column, and returns -1 if the string is not found.

Code

 [Copy Code](#)

```
= IFERROR(SEARCH("-",[PostalCode]),-1)
```

Comments

Note that the data type of the value that you use as an error output must match the data type of the non-error output type. In this case, you provide a numeric value to be output in case of an error because SEARCH returns an integer value.

However, you could also return a blank (empty string) by using [BLANK\(\)](#) as the second argument to IFERROR.

See Also

Reference

[MID Function \(DAX\)](#)

[REPLACE Function \(DAX\)](#)

Other Resources

[Text Functions \(DAX\)](#)

Stay Up to Date with PowerPivot for Excel

For the latest information about PowerPivot for Excel, visit the [PowerPivot for Excel Online Help](#)
