

# RxFind

## %RxFind function

---

### Function

The function searches for the first substring that matches the specified regular expression and returns the index where the substring starts. Search is case sensitive.

### Declaration

```
INT %RxFind(  
    TEXT in Text,  
    TEXT in regExp,  
    INT      in from := 1,  
    INT      out endIndex  
)
```

### Parameters

<b>text</b>	Text, which is to be searched.
<b>regExp</b>	Regular expression that will be searched in the input parameter <i>text</i> .
<b>from</b>	Index, from where the input parameter <i>text</i> will be searched (optional input parameter).
<b>endIndex</b>	Index in <i>text</i> , where the substring matching the specified regular expression ends (optional output parameter).

### Example

The function returns the index, where the substring matching the specified regular expression starts. Optional parameter is *endIndex*. If this parameter is entered, after the function call there is stored ending index of the substring that matches the specified regular expression.

If the searching is unsuccessful, or if the *from* parameter exceeds the total length of input *text*, the function returns 0.

If the parameter *regExp* is not valid regular expression, the function returns -1.

If some of the input parameters is invalid, the function returns undefined value.

```

;looking for an expression "some"
%RxFind("text some text", "some") ;returns 6

;looking for an expression "some" at the beginning
%RxFind("text some text", "^some") ;returns
0 - expression "some" is not at the beginning

;looking for an expression "text" at the beginning
%RxFind("text some text", "^text") ;returns
1 - expression "text" is at the beginning

;looking for an expression "text" at the end
%RxFind("text some text", "text$") ;returns
13 - expression "text" is at the end (starts at the position of 13)

;looking for an expression "xt" or an expression "so"
%RxFind("text some text", "xt | so") ;returns 3 -
expression "xt" was found first and starts at the position of 3

;looking for an expression "some" at the beginning or the expression
"text" at the end
%RxFind("text some text", "^some| text$") ;returns 12 - the
entered string ends with an expression "text" and starts at the position
of 12

;looking for a character in range "a-d"
%RxFind("text some text", "[a-d]") ;returns
9 - character 'a' is located on that position

;looking for a character out of range "a-d"
%RxFind("text some text", "[^a-d]") ;
returns 1 - character 'a' is located on that position

;looking for 'o' character sequence 2 times in a row
%RxFind("text soome text", "e{2}") ;returns
7

;looking for an expression "oper.tor", ( "." means one occurrence of any
character)
%RxFind("operator operaaaator", "oper.tor") ;returns 1 -
matching expression starts at this position

;looking for an expression "oper.*tor", ( ".*" means 0 or more occurrences
of any character), while the entered string is searched from the position
of 10
%RxFind("operator operaaaator", "oper.*tor", 10, _last) ;
returns 10 - matching expression starts at this position, parameter _last
= 20

;looking for an expression "oper.+tor", ( ".+" means 1 or more occurrences
of any character), while the entered string is searched from the position
of 10
%RxFind("operator operaaaator", "oper.+tor", 10, _last) ;
returns 10 - matching expression starts at this position, parameter _last
= 20

```

### Examples of regular expressions

Regular expression	Meaning
expression	Searches for the specified expression in a string.
^expression	Searches for the specified expression at the beginning of the string.
expression\$	Searches for the specified expression at the end of the string.
expression1   expression2	Searches for one of the entered expressions.

<code>^expression   expression \$</code>	Searches for an expression at the beginning or end of the string.
<code>[a-d]</code>	Searches for characters in the specified range (in this case, characters a,b,c,d).
<code>[^a-d]</code>	Searches for all characters except the characters in the specified range (in this case, all characters except a,b,c,d).
<code>a{5}</code>	Searches for 'a' character sequence 5 times in a row.
<code>oper.tor</code>	Searches for "oper.tor", where . is replaced with occurrence of any single character (finds expressions like operator, operxtor, ...).
<code>oper.*tor</code>	Searches for "oper.*tor", where .* is replaced with 0 or more occurrences of any character (finds expressions like opertor, operator, operaaaaator, ...).
<code>oper.+tor</code>	Searches for "oper.+tor", where .+ is replaced with 1 or more occurrences of any character ( <b>does not find expression opertor</b> ).



**Related pages:**

- [Implemented functions](#)
- [Function arguments - types](#)