How to Use Excel's Regex Function to Power Up Your Searches

Regex functions change that. Now you can specify exactly what you need—complex patterns, partial matches, or extracting structured data—without any effort.

Filtering and searching in Excel is no easy task. Regex functions change that. Now you can specify exactly what you need—complex patterns, partial matches, or extract structured data—without the effort.

What is Regex?

Regex is a type of pattern used to search a string of text or characters for a match. Have you ever wondered how websites can tell you that the email pattern you entered on the login page is invalid? That's an example of a regex pattern using an email signature in action.

Regular expressions aren't unique to Excel—they're available in many text editors, programming languages, command-line tools, IDEs, and even Excel's competitor, Google Sheets .

Regex can seem complicated, and it can be if you want to use it to its full potential, but you don't need to be a programmer to use it effectively. In some cases, you can get away with just knowing how to use a few basic symbols and patterns. This guide will keep it as simple as possible so you can get started.

The following are the symbols that will be used in this guide:

Symbol	Describe
-	Specify the character range in parentheses.
^	Matches the beginning of a string.
$	Matches the end of a string.
.	Matches any character except a newline character.

Symbol	Describe
*	Matches zero or more preceding characters.
+	Matches one or more previous characters.
()	Group matching characters into one.
[]	Matches any character inside the brackets.
[^]	Matches any character not within the brackets.
{n}	Matches exactly n instances of the previous character.
{n,}	Matches n or more occurrences of the previous character

Simple regular expression patterns that you can build using these symbols include:

Regex Pattern	Describe
[0-9]	Matches a digit from 0 to 9
[a-zA-z0-9]	This is a range of combinations that matches a single character from lowercase a to z, uppercase A to Z, and 0 to 9.
^pro	Matches any string starting with **pro** .
[^$]	Matches any character other than **$** .
(child)	**Subgroup** sample .

Regex Pattern	Describe
a{3,}	Matches 3 or more occurrences of the part following **a** (for example, **a** , **aa** , or **aaa**).

Regex functions are predefined Excel formulas that can be used to define a pattern for searching and manipulating text strings. There are currently three regex functions. We will see how to use them individually and with other functions.

Search for patterns

The first function we will look at is REGEXTEST. This function takes a text string that you want to search against and a regex pattern, then uses the latter pattern to find a match in the former. The function will return True or False.

The syntax of the REGEXTEST function is as follows:

```
REGEXTEST(string_to_search, regex_pattern_to_use, [case_senstivity])
```

The first two parameters, **string_to_search** and **regex_pattern_to_use,** are self-explanatory. The **[case_sensitivity]** parameter is optional—anything in brackets when talking about Excel syntax is optional—and indicates whether you want the search to be case-sensitive (0) or case-insensitive (1). The default is case-sensitive.

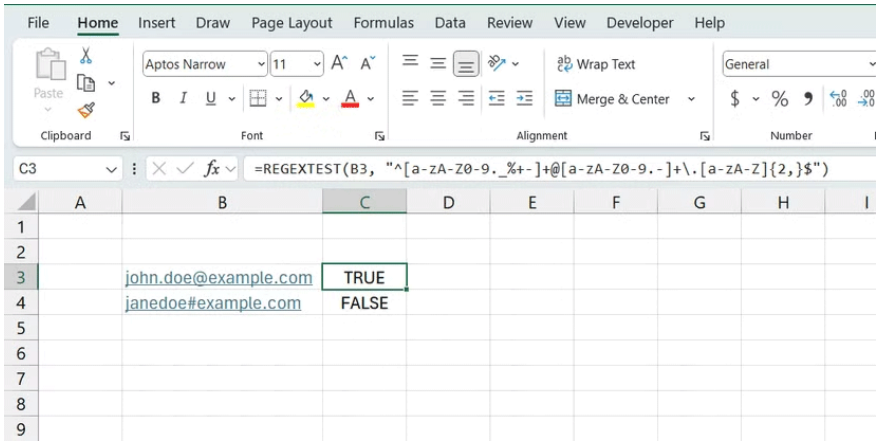
The example will use REGEXTEST to see if the user entered a valid email address using the following formula:

```
REGEXTEST(B3, "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,}$")
```

Here, we are searching in cell B3 to see if it contains an email address using the regular expression pattern below:

```
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,}$
```

If you put the formula in cell C3 and enter **john.doe@example.com** in cell B3, the formula will return True because it matches the email signature.



Additional data using Regex

Next, let's look at the REXEXEXTRACT function. This function returns a substring (a portion of a string) that matches the provided regex pattern.

The syntax of the REXEXEXTRACT function is as follows:

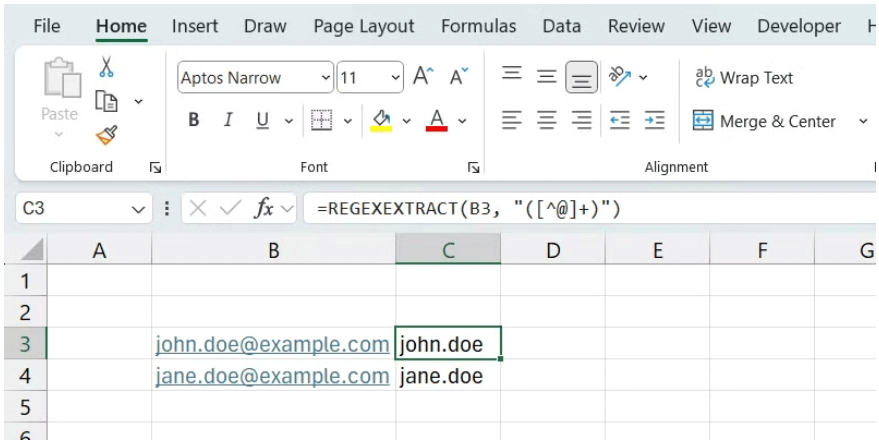
```
REGEXEXTRACT(string_to_search, regex_pattern_to_use, [return_mode], [case_sensti
```

Continuing with the email example, let's add a formula to cell B4 to extract the username of the email part.

The formula would look like this:

```
=REGEXEXTRACT(B3, "([^@]+)")
```

In this formula, we extract everything before the **@** symbol in the email address entered in B3.



Find and replace with Regex

The last regex function we will look at is REGEXREPLACE. This function is similar to Excel's REPLACE function, but also supports RegEx. It takes the text string you want to modify and checks to see if any substrings

match the specified regex pattern. If found, it replaces that string with the provided replacement string.

The syntax of the REGEXREPLACE function is as follows:

```
REGEXREPLACE(string_to_modify, regex_pattern_to_use, replacement_string, [number_
```

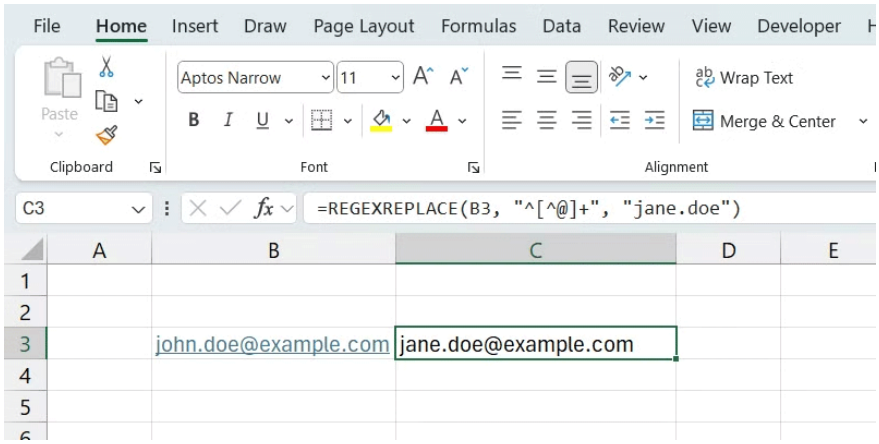
Here are the important parameters to note in this function:

1. **string_to_modify** : The text string you want to modify.
2. **replacement_string** : String to replace the substring with.
3. **number_of_occurrences** : The exact instances you want to replace.

Here's an example of using the function to replace the username portion of an email with another text string:

```
=REGEXREPLACE(B3, "^[^@]+", "jane.doe")
```

The value of B3 is **john.doe@example.com** and after we enter the above formula in cell C3, it will return **jane.doe@example.com.**



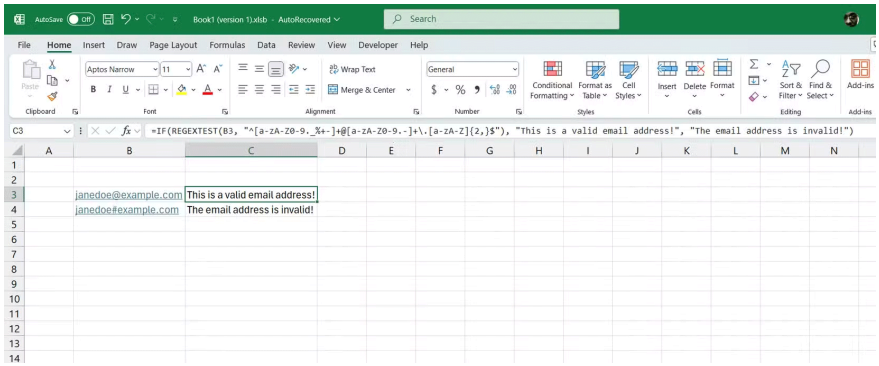
Combining Regex with other functions

You can also combine regex functions with other Excel functions. For example, you can combine the REGEXTEST function with Excel's IF statement and display appropriate messages based on the results.

Here is an example formula:

```
=IF(REGEXTEST(B3, "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,}$"), "This is a
```

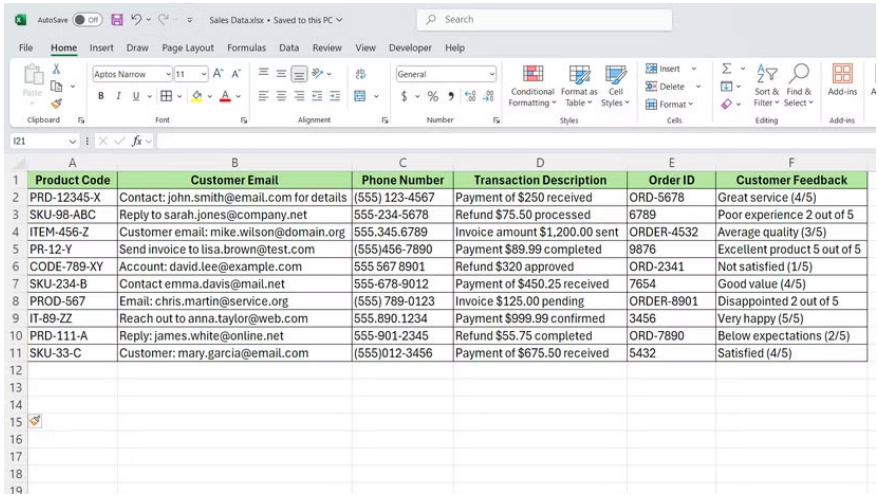
This formula uses an IF statement to check if the email address entered in cell B6 is valid and then displays **This is a valid email address!** if **TRUE** or **The email address is invalid!** if **FALSE** . Additionally, you can pair this formula with the FIND function to quickly find data in Excel.



This is a great way to get started using RegEx in Excel. The use cases and possibilities are only limited by your imagination.

Excel has 3 REGEX functions you can use

They handle different tasks well.



	A	B	C	D	E	F
1	Product Code	Customer Email	Phone Number	Transaction Description	Order ID	Customer Feedback
2	PRD-12345-X	Contact: john.smith@email.com for details	(555) 123-4567	Payment of $250 received	ORD-5678	Great service (4/5)
3	SKU-98-ABC	Reply to sarah.jones@company.net	555-234-5678	Refund $75.50 processed	6789	Poor experience 2 out of 5
4	ITEM-456-Z	Customer email: mike.wilson@domain.org	555.345.6789	Invoice amount $1,200.00 sent	ORDER-4532	Average quality (3/5)
5	PR-12-Y	Send invoice to lisa.brown@test.com	(555)456-7890	Payment $89.99 completed	9876	Excellent product 5 out of 5
6	CODE-789-XY	Account: david.lee@example.com	555 567 8901	Refund $320 approved	ORD-2341	Not satisfied (1/5)
7	SKU-234-B	Contact emma.davis@mail.net	555-678-9012	Payment of $450.25 received	7654	Good value (4/5)
8	PROD-567	Email: chris.martin@service.org	(555) 789-0123	Invoice $125.00 pending	ORDER-8901	Disappointed 2 out of 5
9	IT-89-ZZ	Reach out to anna.taylor@web.com	555.890.1234	Payment $999.99 confirmed	3456	Very happy (5/5)
10	PRD-111-A	Reply: james.white@online.net	555-901-2345	Refund $55.75 completed	ORD-7890	Below expectations (2/5)
11	SKU-33-C	Customer: mary.garcia@email.com	(555)012-3456	Payment of $675.50 received	5432	Satisfied (4/5)

Excel implemented three REGEX functions in 2024: REGEXTEST, REGEXEXTRACT, and REGEXREPLACE. Each function performs a different task, and understanding when to use which one makes all the difference.

REGEXTEST

This function checks if a pattern exists in the text and returns TRUE or FALSE. The syntax is:

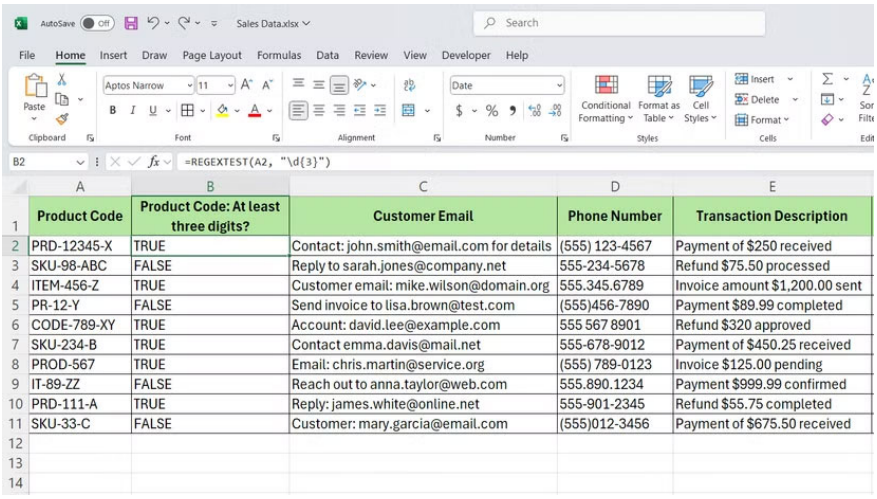
```
=REGEXTEST(text, pattern, [mode])
```

1. **text** : The cell or string you want to check.
2. **pattern** : The REGEX pattern to search for.
3. **mode** (optional): Controls case sensitivity. Use "i" for case-insensitive matching.

Let's say you have a column of product codes in a sales data spreadsheet and you want to highlight items that contain at least 3 consecutive digits. You could use:

```
=REGEXTEST(A2, "d{3}")
```

If cell A2 contains "PRD-12345-X", the function will return TRUE in column B because it found 3 consecutive digits.



	A	B	C	D	E
1	**Product Code**	**Product Code: At least three digits?**	**Customer Email**	**Phone Number**	**Transaction Description**
2	PRD-12345-X	TRUE	Contact: john.smith@email.com for details	(555) 123-4567	Payment of $250 received
3	SKU-98-ABC	FALSE	Reply to sarah.jones@company.net	555-234-5678	Refund $75.50 processed
4	ITEM-456-Z	TRUE	Customer email: mike.wilson@domain.org	555.345.6789	Invoice amount $1,200.00 sent
5	PR-12-Y	FALSE	Send invoice to lisa.brown@test.com	(555)456-7890	Payment $89.99 completed
6	CODE-789-XY	TRUE	Account: david.lee@example.com	555 567 8901	Refund $320 approved
7	SKU-234-B	TRUE	Contact emma.davis@mail.net	555-678-9012	Payment of $450.25 received
8	PROD-567	TRUE	Email: chris.martin@service.org	(555) 789-0123	Invoice $125.00 pending
9	IT-89-ZZ	FALSE	Reach out to anna.taylor@web.com	555.890.1234	Payment $999.99 confirmed
10	PRD-111-A	TRUE	Reply: james.white@online.net	555-901-2345	Refund $55.75 completed
11	SKU-33-C	FALSE	Customer: mary.garcia@email.com	(555)012-3456	Payment of $675.50 received
12					
13					
14					

REGEXTRACT

This function extracts specific text from a string based on a pattern. The syntax is:

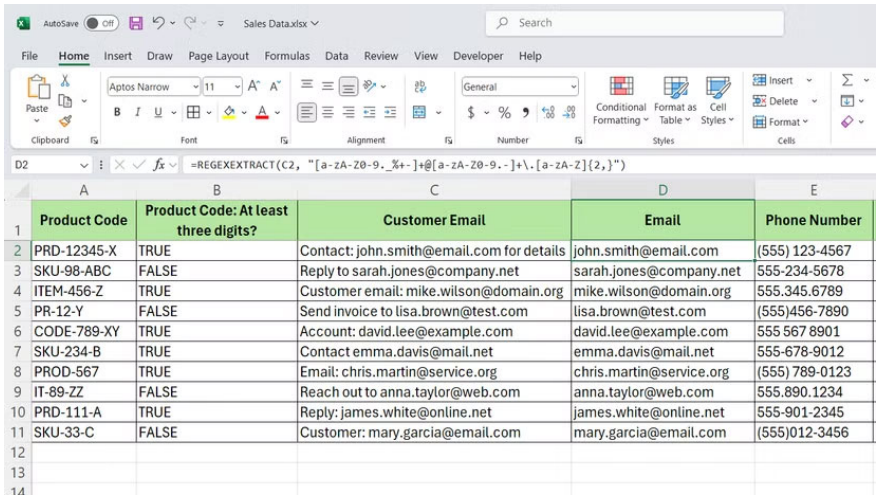
```
=REGEXEXTRACT(text, pattern, [mode], [instance])
```

1. **text** : Source text.
2. **pattern** : The REGEX pattern that defines the content to extract.
3. **mode** (optional): Control case sensitivity ("i" for case insensitive).
4. **instance** (optional): Which match to return if there are multiple results (1 for the first letter, 2 for the second letter, etc.).

In the sales data, column C contains customer emails mixed with other text. To extract just the emails, we would use:

```
=REGEXEXTRACT(B2, "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,}")
```

This template defines standard email formats and extracts them explicitly in column D.



D2 =REGEXEXTRACT(C2, "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}")

	A	B	C	D	E
1	Product Code	Product Code: At least three digits?	Customer Email	Email	Phone Number
2	PRD-12345-X	TRUE	Contact: john.smith@email.com for details	john.smith@email.com	(555) 123-4567
3	SKU-98-ABC	FALSE	Reply to sarah.jones@company.net	sarah.jones@company.net	555-234-5678
4	ITEM-456-Z	TRUE	Customer email: mike.wilson@domain.org	mike.wilson@domain.org	555.345.6789
5	PR-12-Y	FALSE	Send invoice to lisa.brown@test.com	lisa.brown@test.com	(555)456-7890
6	CODE-789-XY	TRUE	Account: david.lee@example.com	david.lee@example.com	555 567 8901
7	SKU-234-B	TRUE	Contact emma.davis@mail.net	emma.davis@mail.net	555-678-9012
8	PROD-567	TRUE	Email: chris.martin@service.org	chris.martin@service.org	(555) 789-0123
9	IT-89-ZZ	FALSE	Reach out to anna.taylor@web.com	anna.taylor@web.com	555.890.1234
10	PRD-111-A	TRUE	Reply: james.white@online.net	james.white@online.net	555-901-2345
11	SKU-33-C	FALSE	Customer: mary.garcia@email.com	mary.garcia@email.com	(555)012-3456
12					
13					
14					

REGEX REPLACE

It swaps the text matching a pattern with another text. The syntax is:

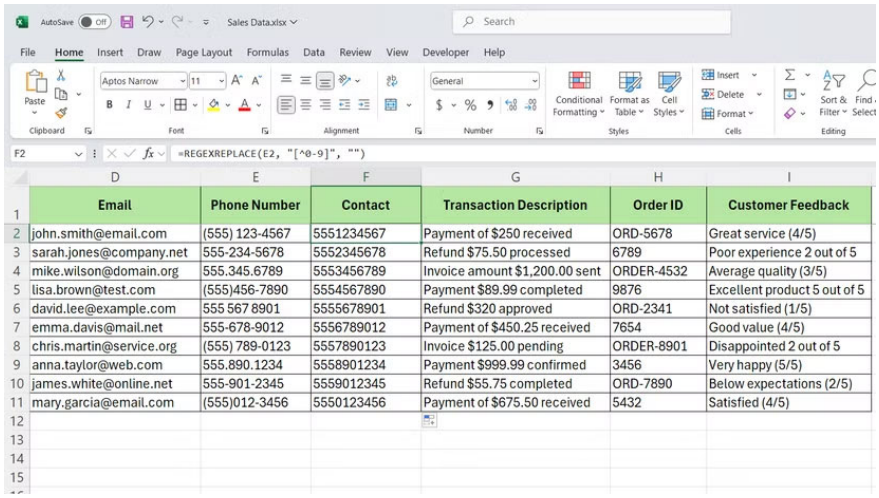
```
=REGEXREPLACE(text, pattern, replacement, [mode], [instance])
```

1. **text** : Original text.
2. **pattern** : The content to search.
3. **replacement** : Content to be replaced.
4. **mode** (optional): Case sensitive.
5. **instance** (optional): Which instance to replace (leave blank to replace all).

If column E has phone numbers in different formats—some with dashes, some with parentheses—you can normalize them by using the following formula to remove everything but the digits:

```
=REGEXREPLACE(C2, "[^0-9]", "")
```

The pattern [^0-9] means "anything that is not a number", and replacing it with an empty string only keeps the digits.



F2 =REGEXREPLACE(E2, "[^0-9]", "")

	D	E	F	G	H	I
1	Email	Phone Number	Contact	Transaction Description	Order ID	Customer Feedback
2	john.smith@email.com	(555) 123-4567	5551234567	Payment of $250 received	ORD-5678	Great service (4/5)
3	sarah.jones@company.net	555-234-5678	5552345678	Refund $75.50 processed	6789	Poor experience 2 out of 5
4	mike.wilson@domain.org	555.345.6789	5553456789	Invoice amount $1,200.00 sent	ORDER-4532	Average quality (3/5)
5	lisa.brown@test.com	(555)456-7890	5554567890	Payment $89.99 completed	9876	Excellent product 5 out of 5
6	david.lee@example.com	555 567 8901	5555678901	Refund $320 approved	ORD-2341	Not satisfied (1/5)
7	emma.davis@mail.net	555-678-9012	5556789012	Payment of $450.25 received	7654	Good value (4/5)
8	chris.martin@service.org	(555) 789-0123	5557890123	Invoice $125.00 pending	ORDER-8901	Disappointed 2 out of 5
9	anna.taylor@web.com	555.890.1234	5558901234	Payment $999.99 confirmed	3456	Very happy (5/5)
10	james.white@online.net	555-901-2345	5559012345	Refund $55.75 completed	ORD-7890	Below expectations (2/5)
11	mary.garcia@email.com	(555)012-3456	5550123456	Payment of $675.50 received	5432	Satisfied (4/5)
12						
13						
14						
15						

These three functions cover most text processing needs. However, you can combine them with the SCAN function for even more flexibility, especially when processing data across multiple rows or extracting repeating patterns.