

Spreadsheet functions

Regular expressions

Overview

01. Introduction

02. Regex special characters

03. Regex functions

04. Data overview

05. The REGEXEXTRACT function

06. The REGEXREPLACE function

07. The REGEXMATCH function

Introduction

A **regular expression** (or **regex** in short) is a sequence of characters used to specify a search pattern.



Why learn regex?

Regexes are used to **solve problems with text strings**. They work by **matching patterns**.

They **simplify complex tasks** involving text transformation.

They also **reduce the chances of human error**.

Regex syntax in Sheets



To create a regular expression, you must use a **specific syntax** – that is, **special characters** and **functions**.

The sequence of special regex characters uses ASCII such as **digits, letters, punctuation, and other symbols**.

Regex functions **match, remove, or replace data** based on their content, e.g. **removing sensitive data** like addresses.

Regex has the reputation for being difficult, but a small time investment will yield recurring benefits.

Regex special characters: anchor

Anchors are special characters that define the position of a pattern in a string. They do not match any characters themselves but rather indicate a position relative to the beginning or end of a line, word, or string. Some commonly used anchors are:

^ (caret)

Matches the start of the line or string of text that the regular expression is searching.

For example, the sequence below captures any text that **begins** with the letters **abc**.

^abc

\$ (dollar)

Matches the end of the line or string of text that the regular expression is searching.

For example, the sequence below captures any text that **ends** with the letters **xyz**.

xyz\$

Regex special characters: metacharacters

Metacharacters are special characters that are used to create patterns that can match a wide range of text inputs. Some commonly used metacharacters are:

. (dot)

Matches any single character, except a new line.

For example, the sequence below captures any string that **starts with "b", ends with "t"**, and has any single character in between.

b . t

| (pipe)

Indicates alternation – that is, an **OR**.

For example, the sequence below matches the word **cat OR dog**.

cat | dog

\ (backslash)

Indicates that the next character is a literal rather than a special character.

For example, the sequence below matches a **literal period**, rather than any character (dot character).

\ .

Regex special characters: character classes

Character classes are sets of characters enclosed in square brackets ([]) that matches any single character within the set. They provide a convenient way to specify a range of characters that a pattern should match. Some commonly used character classes are:

[...]

Matches any character from a set of characters. We separate the first and last character in a set with a dash.

For example, the sequence below matches any letter from **a to f**.

[a-f]

[^...]

Matches any character not in the set of characters.

For example, the sequence below matches any character that's **not** a letter from **a to f**.

[^a-f]

[:alnum:]

Matches alphanumeric characters (letters or digits).

It will match any character in the set of **a-z, A-Z, or 0-9**.

Regex special characters: character classes

[alpha:]

Matches alphabetic characters (letters).

[punct:]

Matches punctuation characters and symbols.

[digit:]

Matches digits.

[print:]

Matches visible characters and spaces.

[graph:]

Matches visible characters only – that is, any characters except spaces, control characters, and so on.

[space:]

Matches all whitespace characters, including spaces, tabs, and line breaks.

Regex special characters: character classes

`[:word:]`

Matches any word character – that is, any letter, digit, or underscore.



Character classes with words in them must be surrounded with another set of square brackets when used in a regular expression. This is because they are not valid shorthand character classes on their own and will not be interpreted correctly without the enclosing brackets. For example:

`[[alpha:]]`, `[[digit:]]`, `[[alnum:]]`, `[[space:]]`, `[[word:]]`, `[[punct:]]`, `[[graph:]]`, and `[[print:]]`.

Regex special characters: shorthand character classes

Shorthand character classes are abbreviated character classes that match common sets of characters. They provide a convenient way to write shorter regular expressions. Some commonly used shorthand character classes are:

\w

Matches any word character – that is, any letter, digit, or underscore.

It will match any character in the set **a-z, A-Z, 0-9, or _**

Equivalent to **[:word:]**

\W

Matches any non-word character – that is, any character that's not a letter, digit, or underscore.

It will match any character **not** in the set **a-z, A-Z, 0-9, or _**

Equivalent to **[^[:word:]]**

\s

Matches a whitespace character.

For example, the sequence **stock\stips** matches the phrase “stock tips”.

Equivalent to **[:space:]**

Regex special characters: shorthand character classes

\s

Matches any character that's not a whitespace.

Equivalent to `[^[:space:]]`

\d

Matches any digit from 0-9.

Equivalent to `[:digit:]`

\D

Matches any character that's not a digit from 0-9.

Equivalent to `[^[:digit:]]`

Regex special characters: quantifiers

Quantifiers are special characters that are used to specify the number of times a character or group of characters should be matched in a regular expression. Some commonly used quantifiers are:

*** (asterix)**

Matches zero or more occurrences of the preceding character or group.

For example, the sequence below matches **a**, **ab**, **abb**, **abbb**, and so on.

ab*

+ (plus sign)

Matches one or more occurrences of the preceding character or group.

For example, the sequence below matches **ab**, **abb**, **abbb**, and so on, but **not a**.

ab+

? (question mark)

Matches zero or one occurrence of the preceding character or group.

For example, the sequence below matches **a** and **ab**, but **not abb**.

ab?

Regex special characters: quantifiers

{n}

Matches the preceding expression exactly **n** times.

For example, the sequence below matches any letter from **a** to **c** only if two letters occur in a row. Thus, it would match **ab** and **bc** but not **abc** or **aabbc**.

[a-c]{2}**{n,}**

Matches **n** or more occurrences of the preceding character or group.

For example, the sequence below matches **aa**, **aaa**, **aaaa**, and so on.

a{2, }**{n,m}**

Matches between **n** and **m** occurrences of the preceding character or group.

For example, the sequence below matches **aa**, **aaa**, and **aaaa**, but **not a** or **aaaaa**.

a{2,4}

Regex special characters: group

A **group** is a portion of a regular expression enclosed in parentheses (). It captures a portion of the matched text and extracts it as a separate substring.

(...)

Groups parts of an expression. Grouping is used to apply an anchor or quantifier to a group or to match a character class before or after a group.

For example, the sequence below matches "" (zero occurrence), **ab**, **abab**, **ababab**, and so on.

(ab)*

Regex functions

Regex functions in spreadsheets refer to functions that can use regular expressions to manipulate text in a spreadsheet. Regex is not utilised by every spreadsheet function because:

01. Regexes have a specialised syntax and require specific processing to be applied to text.
02. Many functions perform tasks on non-textual data that don't require pattern matching or text manipulation.
03. Regex functions are provided as specialised tools that are easy to understand and can be used as needed.
04. Regex can be complex and difficult to master and including them in every function could lead to user errors.

Data overview


To investigate the use of regular expressions in spreadsheets, we will use a **Tweets on climate change** dataset that has 100 rows and the following columns:

1. ID

A numeric string that is associated with and uniquely identifies a single Tweet within the dataset. It makes it possible to access and interact with a specific Tweet.

2. Text

An aggregated Tweet pertaining to climate change.



Dataset		
	A	B
1	ID	Text
2	1028954403129184256	Gotta love the facts. https://t.co/bZ2G8AZuo9
3	1028954356572250112	RT @ToolangiForest: A great day of action for our message of "Dear Dan"! Toolangi community & friends joined together to respectfully ask @...
4	1028954497341480960	@jonkudelka Harvey Norman reckons climate change is bunkum because his mates who own coal companies need people to buy polluting stuff
5	1028954494133043200	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...
6	1028954811511844864	RT @FranceinIreland: On 5th November we call all creative citizens w/ practical solutions to fight against #climatechange to join us for a...
7	1028954782457909250	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...
+ ☰ Tweets on climate change ▾		

The REGEXEXTRACT function

The **REGEXEXTRACT** function is used to **extract the first matching substrings** according to a regular expression.

=REGEXEXTRACT(text, regular_expression)

- **text** – The input text.
- **regular_expression** – The first part of **text** that matches this expression will be returned.

Some applications include extracting:

- The first/last characters from a string.
- Numbers from a string.
- Whole words based on a partial match.
- One of a list of words.
- Contents between certain characters.
- Different parts of a URL.

The REGEXEXTRACT function

Example use:

Extract at least one of a chosen set of climate change related keywords (**global warming OR climate change**) from the Tweets.

- Our function will only extract the first, if any, of the two phrases it comes across in a Tweet. We will therefore use **OR logic**.
- Tweets that **do not mention** any of the key words will be tagged with the label **No mention**.



Note that **REGEXEXTRACT** returns an error when the **text** does not have a match for the **regular_expression**. We will therefore use **IFERROR** to detect these errors and replace them with the label **No mention**.

Recall that the **pipe symbol (|)** represents the **OR** operator in **regular expressions**.

The REGEXEXTRACT function

Example use:

01. Enter `=IFERROR(REGEXEXTRACT(B2, "global warming|climate change"), "No mention")` on cell C2.

02. Replicate the formula to the other rows by dragging the fill handle down.

Try on your own

How many occurrences of **global warming** versus **climate change** did you discover in the dataset?

01.

`=IFERROR(REGEXEXTRACT(B2, "global warming|climate change"), "No mention")`

	A	B	C
1	ID	Text	
2	1028954403129184256	Gotta love the facts. https://t.co/bZ2G8AZuo9	No mention
3	1028954356572250112	RT @ToolangiForest: A great day of action for our message of "Dear Dan"! Toolangi community & friends joined together to respectfully ask @...	
4	1028954497341480960	@jonkudelka Harvey Norman reckons climate change is bunkum because his mates who own coal companies need people to buy polluting stuff	
5	1028954494133043200	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...	
6	1028954811511844864	RT @FranceinIreland: On 5th November we call all creative citizens w/ practical solutions to fight against #climatechange to join us for a...	
7	1028954782457909250	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...	
8	1028954810781814784	You send me crap It's 5 minutes to midnight for a mute https://t.co/FFhYHCitKb #Alaska's #NorthSlope hit by strongest #quake noted in region	

02.

+ ≡ Tweets on climate change ▾

The REGEXREPLACE function

The **REGEXREPLACE** function is used to **replace part of a text string** with a different text string using regular expressions.

=REGEXREPLACE(text, regular_expression, replacement)

- **text** – The text, a part of which will be replaced.
- **regular_expression** – The regular expression.
All matching instances in **text** will be replaced.
- **replacement** – The text which will be inserted into the original text.

Some applications include removing or replacing:

- All spaces from a text.
- All numerical values.
- All URLs from a string.
- HTML tags from a string.

The REGEXREPLACE function

Example use:

Remove all **retweet** and **hashtag** symbols from all the Tweets.

- In this case, we want to remove all instances of **RT** and **#** from our texts.
- We will use the **OR logic** here as well since a Tweet can have both, either of the two, or neither of the symbols.



- A **retweet** is a feature on Twitter that allows users to **share someone else's Tweet** with their own followers. It is a way to amplify content and share interesting or informative Tweets with a wider audience. Retweets are prepended with the symbol "**RT**".
- A **hashtag** is a word or phrase that is used to **identify** and **categorise Tweets on a particular topic**. It makes it easier for users to search for and find Tweets on specific topics or themes, as well as to participate in broader conversations and communities on Twitter. Hashtags are prepended with the symbol "**#**".

The REGEXREPLACE function

Example use:

- To remove a **regular_expression** from a **text** in our **REGEXREPLACE** function, we will make the **replacement** a blank string ("").

01. Enter **=REGEXREPLACE(B2,"#|RT", "")** on cell **C2**.

02. Replicate the formula to the other rows by dragging the fill handle down.

Try on your own

It is common practice to remove **mentions** (@soandso) and **URLs**. How would the **regular_expression** look if these were included in our function?

C2			01.
=REGEXREPLACE(B2,"# RT", "")			
	A	B	C
1	ID	Text	Symbols
2	1028954403129	Gotta love the facts. https://t.co/bZ2G8AZuo9	Gotta love the facts. https://t.co/bZ2G8AZuo9
3	1028954356572	RT @ToolangiForest: A great day of action for our message of "Dear Dan"! Toolangi community & friends joined together to respectfully ask @...	
4	1028954497341	@jonkudelka Harvey Norman reckons climate change is bunkum because his mates who own coal companies need people to buy polluting stuff	
5	1028954494133	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...	
6	1028954811511	RT @FranceinIreland: On 5th November we call all creative citizens w/ practical solutions to fight against #climatechange to join us for a...	
7	1028954782457	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...	
8	1028954810781	You send me crap It's 5 minutes to midnight for a mute https://t.co/FFhYHCitKb #Alaska's #NorthSlope hit by strongest #quake noted in region	

02.

Tweets on climate change

The REGEXMATCH function

The **REGEXMATCH** function is used to check whether **a piece of text matches** a regular expression.

=REGEXMATCH(text, regular_expression)

- **text** – The text to be tested against the regular expression.
- **regular_expression** – The regular expression to test the **text** against.

Some applications include:

- Identifying cells that contain a particular letter, word, or phrase.
- Finding the exact match of a string.
- Identifying texts that contain hashtags.
- Finding or validating phone numbers, email addresses, IDs, credit card numbers.

The REGEXMATCH function

Example use:

Identify all Tweets that are **not retweets**.

- All Tweets that are not retweets will ideally not start with the symbol **RT**. We will therefore identify all Tweets that **start with RT** and tag them as **Retweet** and the rest as **Not retweet**.



Consider the following scenario:

Twitter **doesn't** allow its users to post empty Tweets or Tweets that only contain regular space characters. Would we therefore assume that a text with the symbol **RT** only is an empty retweet?



Note that **REGEXEXTRACT** returns a boolean.

We will therefore use the regex expression **^RT** in a **REGEXEXTRACT** function to identify if a text is a retweet. When the **IF** function is **TRUE**, the Tweet will be tagged as a **Retweet**. Otherwise, the Tweet will be tagged as **Not retweet**.

Recall that the caret symbol (^), when **not** enclosed in and at the beginning of square brackets (**[^...]**), **matches the start** of the line or string of text that the regular expression is searching.

The REGEXMATCH function

Example use:

01. Enter `=IF(REGEXMATCH(B2, "^RT[a-zA-Z0-9]+"), "Retweet", "Not retweet")` on cell C2.
02. Replicate the formula to the other rows by dragging the fill handle down.

Try on your own

What if a user tweeted the symbol **RT** only which, in this case, would not be a retweet? Delete everything after the **RT** symbol on cell B3 to see why we included the regex `[a-zA-Z0-9]+`.

C2 =IF(REGEXMATCH(B2, "^RT[a-zA-Z0-9]+"), "Retweet", "Not retweet")		
ID	Text	
1028954403129184256	Gotta love the facts. https://t.co/bZ2G8AZuo9	Not retweet
1028954356572250112	RT @ToolangiForest: A great day of action for our message of "Dear Dan"! Toolangi community & friends joined together to respectfully ask @...	
1028954497341480960	@jonkudelka Harvey Norman reckons climate change is bunkum because his mates who own coal companies need people to buy polluting stuff	
1028954494133043200	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...	
1028954811511844864	RT @FranceinIreland: On 5th November we call all creative citizens w/ practical solutions to fight against #climatechange to join us for a...	
1028954782457909250	RT @jayrosen_nyu: Why does skepticism about immigration walk hand in hand with skepticism about the science of climate change? I know we're...	
1028954810781814784	You send me crap It's 5 minutes to midnight for a mute https://t.co/FFhYHCitKb #Alaska's #NorthSlope hit by strongest #quake noted in region	