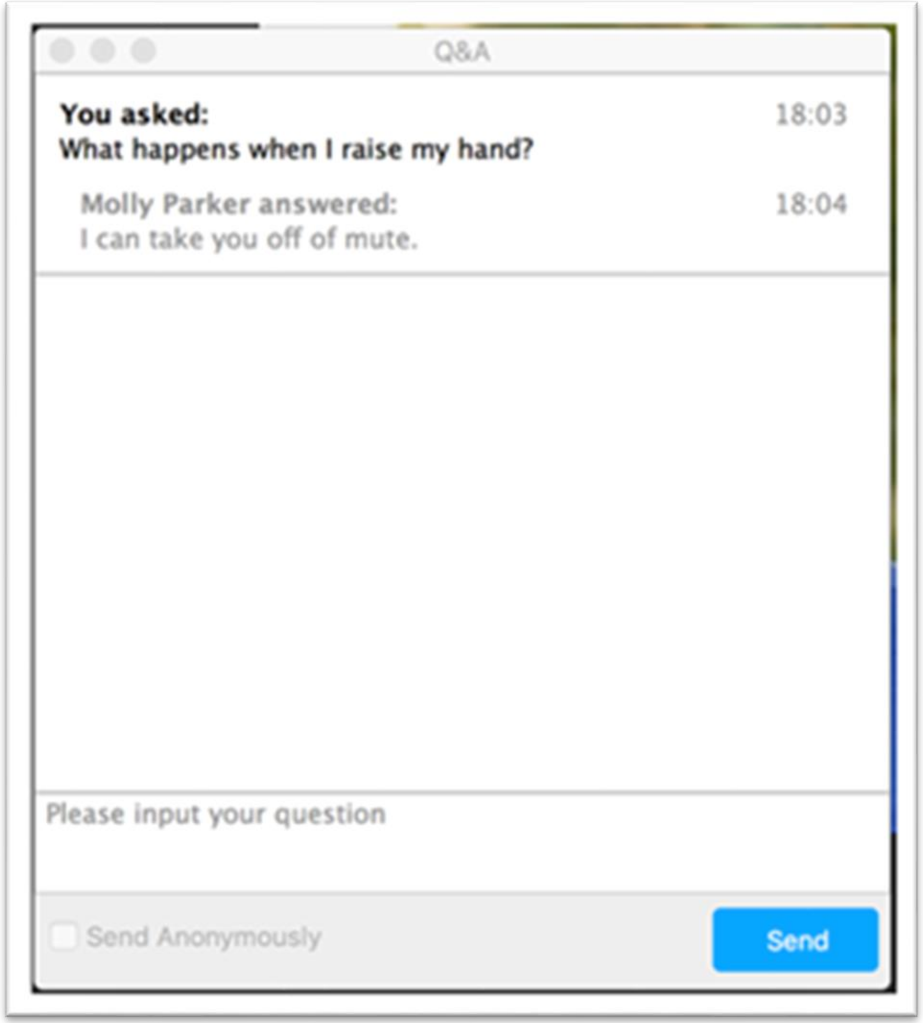
Introducing Regex

(in Excel)

Date: 16 January 2026

Presenter: Ian Pay, ICAEW

Ask a question



You asked: 18:03
What happens when I raise my hand?

Molly Parker answered: 18:04
I can take you off of mute.

Please input your question

□ Send Anonymously

Click on the Q&A button in the bottom toolbar to open the submit question prompt.

Type in your question and click send.

Note. If you wish to ask anonymously tick the send anonymously box shown on the illustration to the left.

Agenda

- What is regex?
- Regex syntax 101
- Application of regex in Excel
- Beyond regex in Excel
- Q&A

About today's presenter

Ian Pay
Head of Data Analytics & Tech ICAEW

- Joined ICAEW in 2022 after over 10 years in data analytics at PwC
- User of Excel, SQL and Alteryx, amongst other tools
- Passionate about data skills
- Former teacher

1

What is regex?

(and how to pronounce it!)

What is a Regular Expression (regex)?

- Syntax for matching text patterns
- Not executable in its own right
- Available in many programming languages and function-driven software
- The basis for advanced search syntax in search engines and even ‘Find and Replace’ in Microsoft Word!
- Is it a ‘hard’ g (as in regular) or ‘soft’ g (as in rejoice)? No-one can agree!

2

Regex syntax 101

Regex syntax 101

- At a simple level, a regex pattern is used to look for ‘a’ in ‘b’.
- Most characters are ‘literal’ – they match exactly as they are written
- For example:

The cat sat on the mat next to the catalogue and was placated

- The regex “**`cat`**” will identify every time that text appears
- Regex is by default *case sensitive:*

Searching for ‘**`cat`**’ in *The cat called Cat sat on the mat next to the catalogue* will **not** return ‘Cat’, unless changes are made to the syntax

Metacharacters

- Special characters that open up a wider range of matching options
- Many metacharacters are contextual and their behaviour is based on what precedes/follows them
- A range of metacharacters defined in the syntax:
  - Classes and groups – sets or categories of characters to be matched
  - Quantifiers – to specify zero, one or more characters
  - Modifiers and anchors – to specify how regex patterns are matched against a text string
- Regex reads character by character – this can make it difficult to parse, especially with complex combinations of metacharacters
- There are far more permutations than can be covered in this webinar, and usually more than one ‘right’ answer!

Classes

Syntax:

- **`[ ]`** – a list or range of characters. **`^`** after [inverts the list
- **`\d \w \s`** – matching any digit (0-9), alphanumeric (a-z,A-Z,0-9 and _) or space
- **`\D \W \S`** – any non-digit, non-alphanumeric or non-space
- **`.`** – any one character

Examples:

`[A-Z]` matches any upper case letter

`[^A-Z]` matches any non-upper case character

`[aeiou]` matches any vowels

`[bchm]at` matches 'bat, 'cat', 'hat' or 'mat', but not 'sat'

`[A-Z]\d` matches a single upper case letter followed by a single digit e.g. A1, B5, G9

`c.t` matches cat, cot, cut (or indeed any character between c and t, including spaces!

Quantifiers

Syntax:

- **`?`** – zero or one of the previous character, class or group
- **`*`** – zero or more of the previous character, class or group
- **`+`** – one or more of the previous character, class or group
- **`{x,y}`** – specific minimum/maximum quantifiers (both are optional)
- **`|`** (pipe) – equivalent to ‘OR’

Example:

`colou?r` matches colour or color

`to+t` matches tot, toot, tooot etc. but not toast, tout etc.

`\d{4}` matches a number which is 4 digits long. **`\d{4,6}`** matches 4-6 digits, **`\d{,4}`** matches 0-4 digits, and **`\d{4,}`** matches *at least* 4 digits

`specializ|se` matches specialise or specialize

Groups

- Use of ( ) defines groups of characters, which can be used in different ways depending on the context or other quantifiers or modifiers:
  - `(abc)` matches the group ‘abc’ every time it appears
    - `(cat|dog)s` will match ‘cats’ and ‘dogs’
  - `(?=abc)` matches anything immediately before the parentheses, where it is followed by ‘abc’ – known as ‘lookahead’. `(?<=abc)` matches anything immediately after the parentheses – ‘lookbehind’.
    - `\d+(?=th)` matches any one or more digits that are followed by ‘th’ e.g. “3rd, **4**th, **5**th, **10**th”
  - Use of `\n` (where n is the group number) allows matching to a previous group
    - `(\d+)=\1` matches where the digits before ‘=‘ are the same as the digits after e.g. 5=5, 12=12 but not 10=100

Modifiers, anchors and escapes

Syntax:

- **`^`** matches the start of a string
- **`$`** matches the end of a string
- **`\b`** defines a boundary between alphanumeric and non-alphanumeric character (i.e. the start or end of a word)
- **`(?i)`** forces case insensitivity for the remainder of the string. **`(?-i)`** reverts this
- **`\`** can be used to 'escape' a metacharacter for a literal match

Example:

`^\w+` matches the first word in a string

`\w+$` matches the last word in a string

`\b(t|T)\w+` matches any word that starts with 't' or 'T'

`\w+(t|T)\b` matches any word that ends with 't' or 'T'

`te(?i)st` matches tesT, teSt, teST but not tESt

`\.` matches a '.' (instead of treating it as a wildcard)

Regex Example – UK Postcodes

The rules:

- Should be alphanumeric, upper case with a space before the last 3 characters
- In one of the following formats:
  - A9 9AA
  - A99 9AA
  - AA9 9AA
  - AA99 9AA
  - A9A 9AA
  - AA9A 9AA
- C, I, K, M, O, V are never used in the second part of the postcode

The regex:

- `^[A-Z]{1,2}\d[A-Z\d]?\s\d[ABD-HJLNP-UW-Z]{2}$`
- This application enforces upper case and the space, however it may be necessary to set as case insensitive using **`(?i)`**, or make the space optional with **`\s?`**

Regex Example – UK Telephone Numbers

The rules:

- If international code provided, should start with “44” or “0044”, then “0” is optional
- If no international code then must start with “0”
- For standard telephone numbers, next digit should not be a 4, 6 or 9
- After this, number should be 8 or 9 digits in total
- Assumed spaces or ‘-’, parentheses and ‘+’ cleansed before parsing!

The regex:

- `((0044|44)0?|0)[123578]\d{8,9}`
- Regex doesn’t have a simple “if…then…else” concept – just have to nest ‘|’ in groups

Regex Question

Question – what is the correct regex syntax to parse an e-mail address?

1. ^.+@.+\..+$
2. ^\w+\.?\w+@\w+\.[A-Za-z]+$
3. ^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$
4. ^[A-Za-z0-9._%+-]*@[A-Za-z0-9.-]*\.[A-Za-z]{2,}$

Answer…

Unfortunately with regex, the complex answer is usually the (most) right one!

Always pay attention to quantifiers!

3

Applying regex in Excel

Regex in Excel

- Regex is great for input validation, cleansing and extracting substrings
- Three Excel functions introduced in 2024:
  - `REGEXTEST` – a true/false flag to match text against a regex pattern
  - `REGEXEXTRACT` – pull out a section of text from a string, according to a regex pattern
  - `REGEXREPLACE` – replacing text in a string, where the regex pattern occurs
- Can also use in combination with other functions:
  - `REGEXTEST` can be used with any function requiring a true/false input e.g. IF type functions, `FILTER` etc.
  - `REGEXEXTRACT` and `REGEXREPLACE` return texts/arrays which can be used as inputs for other functions
- Regex can also be used as the ‘lookup value’ in `XLOOKUP` and `XMATCH`, to enable partial, pattern-based, *case sensitive* text matching in these lookup functions.

Regex in Excel

- Let’s see it in action…

4

Beyond regex in Excel

Beyond regex in Excel

- https://regexr.com/ and https://regex101.com/ - useful resources to test and explain regex syntax
- AI tools to help write regex – including Copilot in Excel!
- Regex enables more *programmatic*, *repeatable* solutions – but AI can be used to support
- Regex unlocks a lot of potential in:
  - Python, by importing the built-in ‘re’ module, or many other general-purpose programming languages such as Java/Javascript, C(#/++), VBA and R, as well as many flavours of SQL
  - Low-code software such as Alteryx and KNIME
  - Regex also supported in Google Docs/Sheets
- There is much more to learn with regex – go forth and regexify!

5

Questions?

Next Excel Community webinar

26 Feb | Excel tips and tricks live: your questions answered

icaew.com/excelcommunityevents

ICAEW

icaew.com