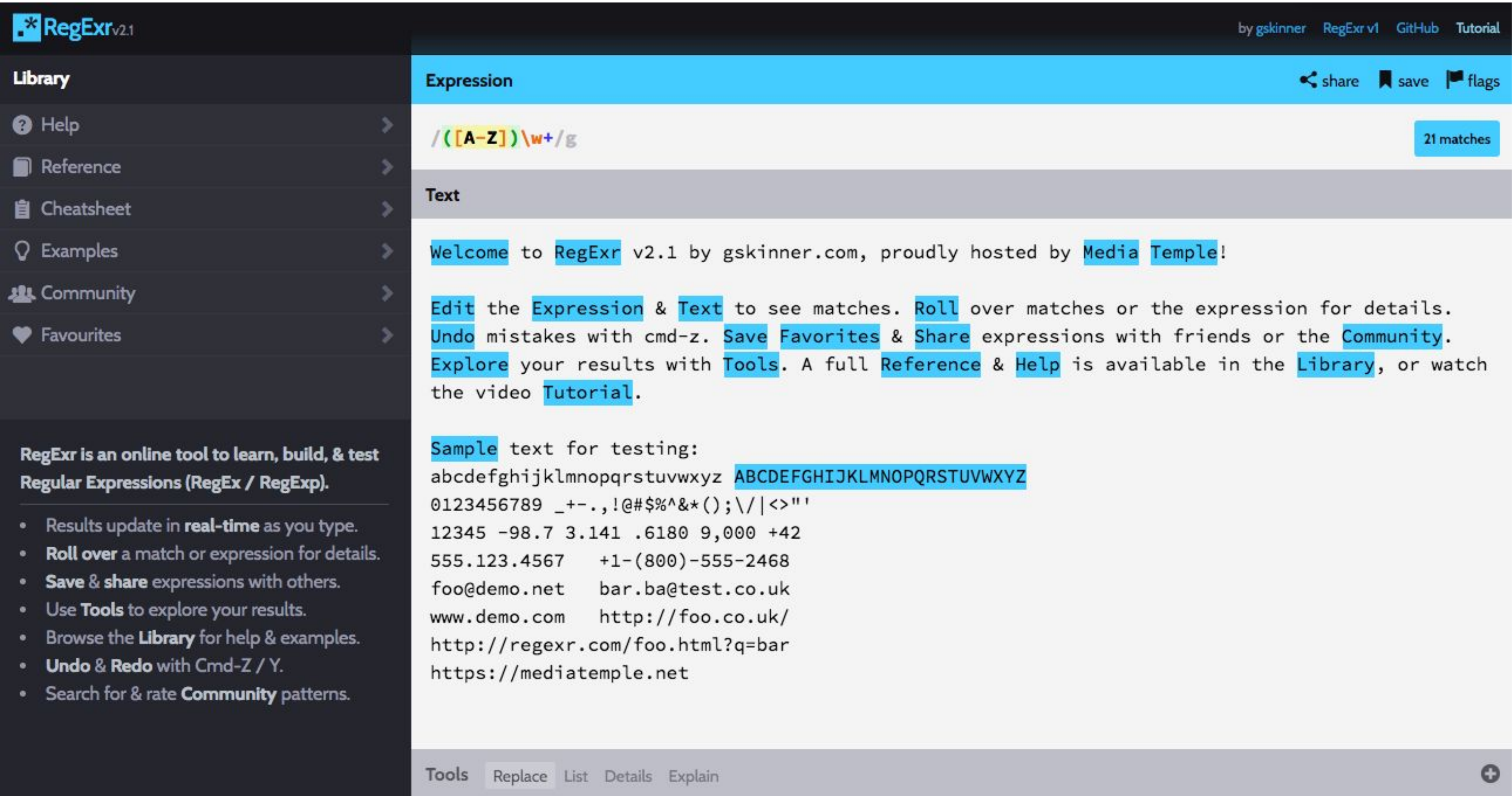
Regex



```
/([A-Z])\w+/g
```

RegExr is an online tool to learn, build, & test Regular Expressions (RegEx / RegExp).

- Results update in **real-time** as you type.
- **Roll over** a match or expression for details.
- **Save** & **share** expressions with others.
- Use **Tools** to explore your results.
- Browse the **Library** for help & examples.
- **Undo** & **Redo** with Cmd-Z / Y.
- Search for & rate **Community** patterns.

```
Welcome to RegExr v2.1 by gskinner.com, proudly hosted by Media Temple!

Edit the Expression & Text to see matches. Roll over matches or the expression for details.
Undo mistakes with cmd-z. Save Favorites & Share expressions with friends or the Community.
Explore your results with Tools. A full Reference & Help is available in the Library, or watch
the video Tutorial.

Sample text for testing:
abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789 _+-.,!@#$%^&*();\/|<>"'
12345 -98.7 3.141 .6180 9,000 +42
555.123.4567   +1-(800)-555-2468
foo@demo.net   bar.ba@test.co.uk
www.demo.com   http://foo.co.uk/
http://regexr.com/foo.html?q=bar
https://mediatemple.net
```

Playground! https://regexr.com/

Regular expressions (regex)

- A way of describing a pattern
- Useful for identifying information you want to extract
- E.g. dates: 11th September 2017
- E.g. emails: name@site.org
- E.g. surrounding HTML tags etc.
- Combination of specific characters and symbols to describe rest

The regex

- **Literal**: any character that means what it says, e.g. `a`, `3`
- **Metacharacter**: special character that means something else, e.g. `[\^$.|?*+()`
- **Escape sequence**: when we want a special character to be treated literally! E.g. `\?`

Metacharacters

- `.` Any character
- `\d` Any single digit
- `\D` Any non-digit
- `\w` Any alphanumeric character
- `\W` Any symbol (@, #, £, %)
- `\s` Any space character
- `\r` Line break

Ranges: use [ ]

- `[0-9]` Any number
- `[a-z]` Any lower case letter
- `[aeiou]` Any vowel
- `[0-9][a-zA-Z]` number, then letter
- `[A-Z]` Any upper case letter
- `[a-zA-Z]` Any letter
- `[Rr]` Upper or lower case R

```
[ ] p = re.compile(' we \w+')
    p
```

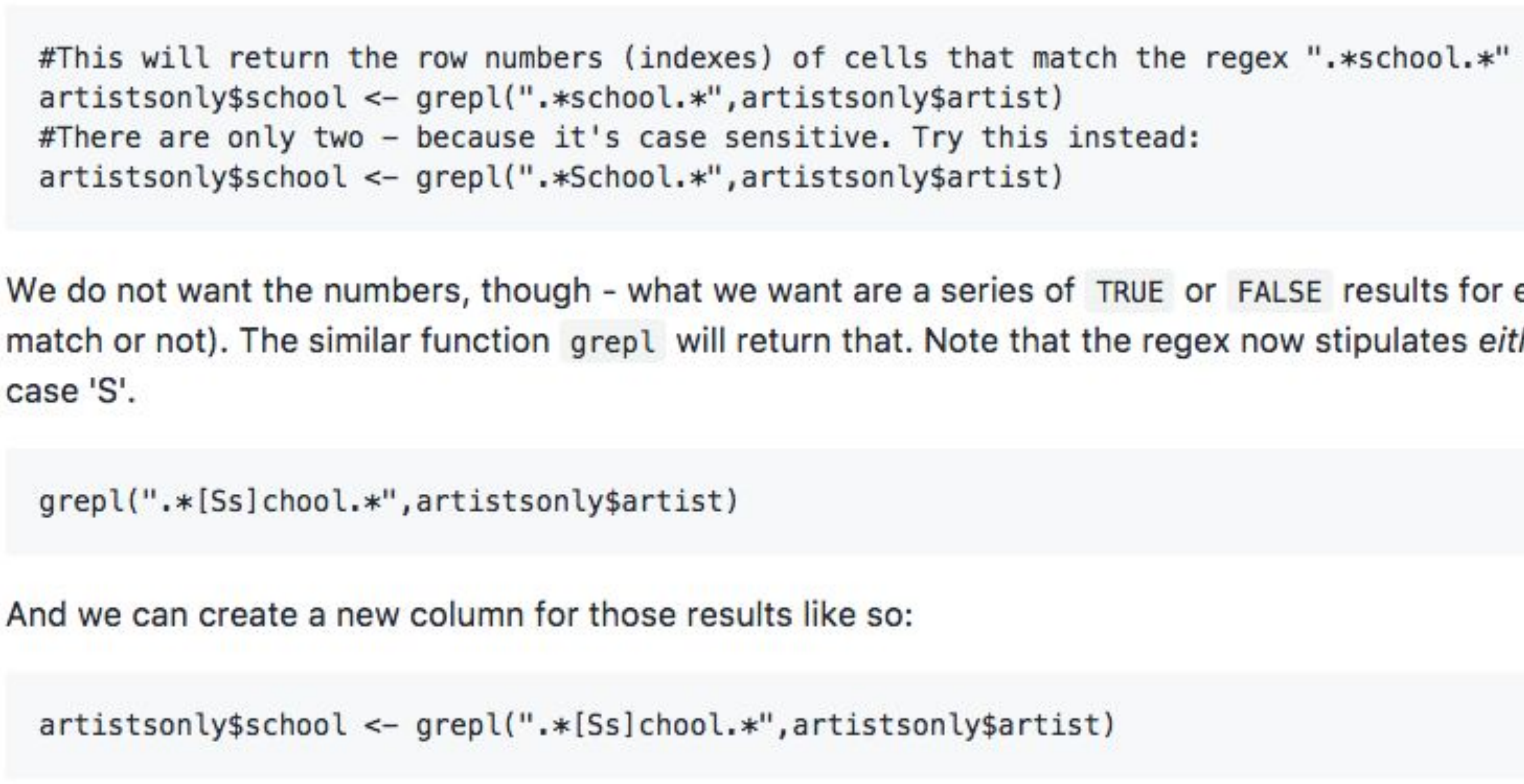
```
re.compile(r' we \w+', re.UNICODE)
```

Finding all matches using `.findall()`

We then use that with `.findall()` to find all matches within a specified string,

```
print(p.findall(" and we will build"))
```

```
[' we will']
```



```
#This will return the row numbers (indexes) of cells that match the regex ".*school.*"
artistsonly$school <- grepl(".*school.*",artistsonly$artist)
#There are only two - because it's case sensitive. Try this instead:
artistsonly$school <- grepl(".*School.*",artistsonly$artist)
```

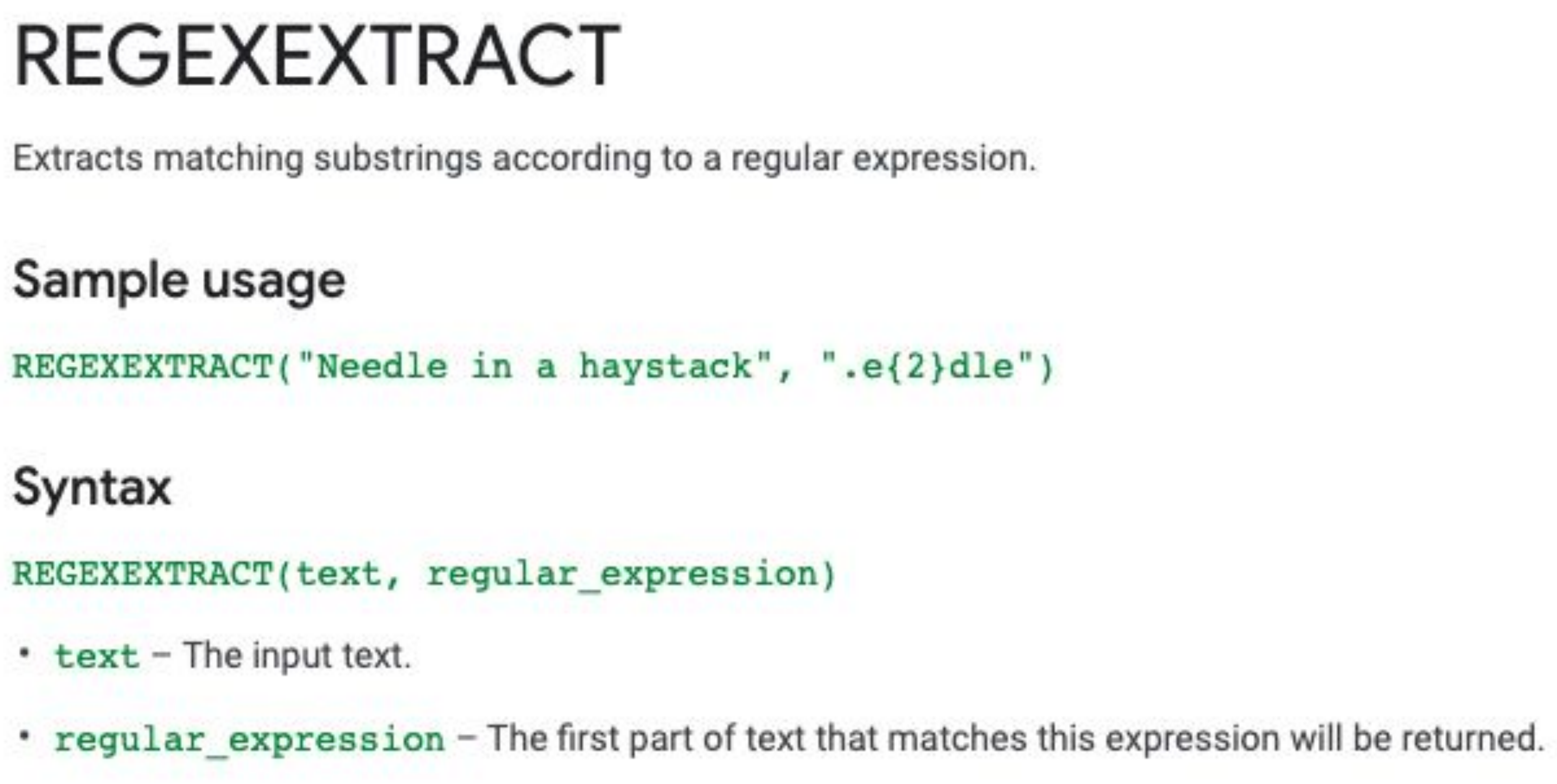
We do not want the numbers, though - what we want are a series of `TRUE` or `FALSE` results for e
match or not). The similar function `grepl` will return that. Note that the regex now stipulates *eit*
case 'S'.

```
grepl(".*[Ss]chool.*",artistsonly$artist)
```

And we can create a new column for those results like so:

```
artistsonly$school <- grepl(".*[Ss]chool.*",artistsonly$artist)
```

[GitHub repo]()
More material in [Rintro repo, text folder]()



REGEXEXTRACT

Extracts matching substrings according to a regular expression.

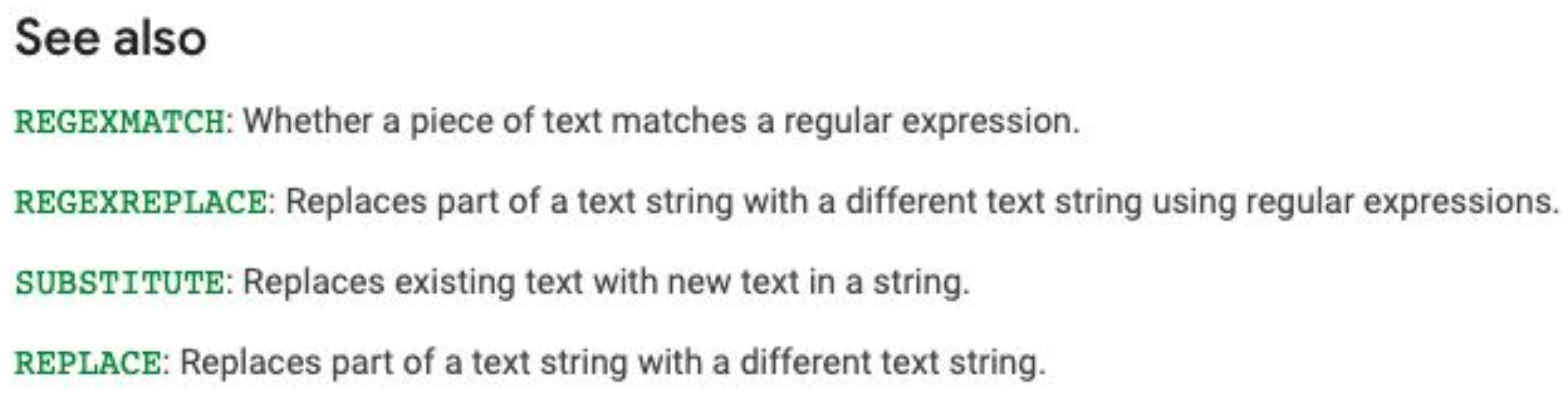
Sample usage

`REGEXEXTRACT("Needle in a haystack", ".e{2}dle")`

Syntax

`REGEXEXTRACT(text, regular_expression)`

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



See also

`REGEXMATCH`: Whether a piece of text matches a regular expression.

`REGEXREPLACE`: Replaces part of a text string with a different text string using regular expressions.

`SUBSTITUTE`: Replaces existing text with new text in a string.

`REPLACE`: Replaces part of a text string with a different text string.

https://support.google.com/docs/answer/3098244?hl=en-GB

What are regular expressions — and how to use them in Google Sheets to get data from text

Leave a reply

FINDING STORIES WITH SPREADSHEETS
By Paul Bradshaw
Get the book:
Leanpub.com/spreadsheetstories

This is an extract from Finding Stories in Spreadsheets. Get the book here.

*In an extract from a new chapter in the ebook Finding Stories in Spreadsheets, I explain what **regular expressions** are — and how they can be used to extract information from spreadsheets. The ebook version of this tutorial includes a dataset and exercise to employ these techniques.*

The story was an unusual one: the BBC Data Unit had been given access to a dataset on more than 200,000 works of art in galleries across the UK. What patterns could we find in the data that would allow us to tell a story about the nature of the nation's paintings?

https://onlinejournalismblog.com/2021/06/22/what-are-regular-expressions-and-how-to-use-them-in-google-sheets-to-get-data-from-text/

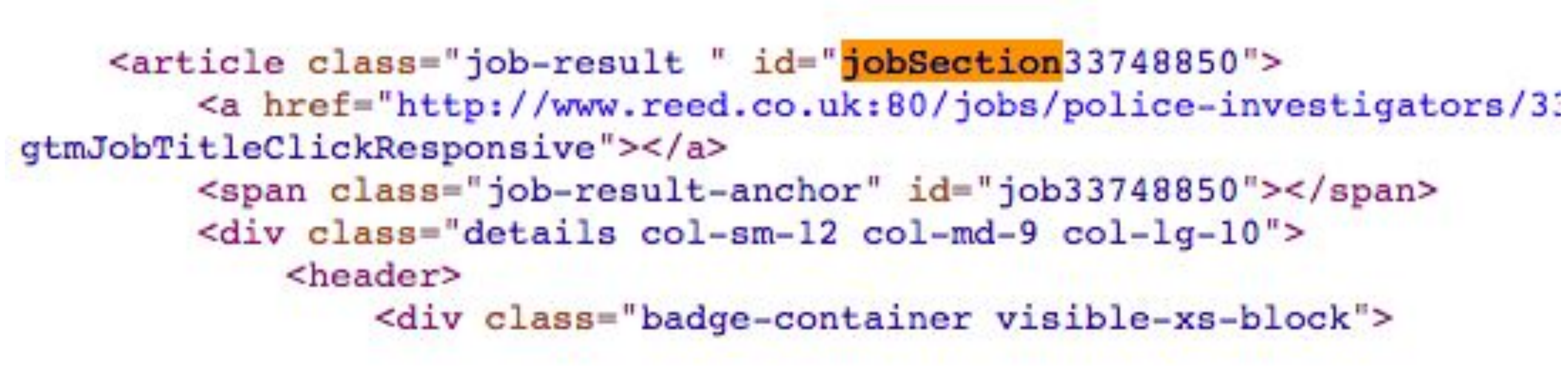
Quantifiers: * + ? { }

- `.*` Any character – **none** or more
- `1+` Number 1 – **one** or more
- `\d+` single digit – one or more
- `u?` A letter u – **one** or **none**
- `colou?r` matches “colour” (**one** u) or “color” (**none** u)

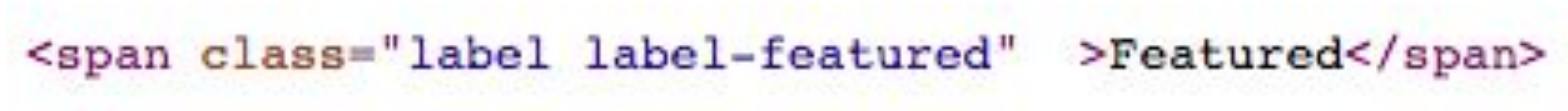
Quantifiers: * + ? { }

- `\d{3}` single digit – three together
- `\d{7,9}` single digit – **between** 7 and 9 together
- `\d{7,}` single digit – at **least** 7 together

How would you grab these codes?



```
    <article class="job-result " id="jobSection33748850">
        <a href="http://www.reed.co.uk:80/jobs/police-investigators/3
gtmJobTitleClickResponsive"></a>
        <span class="job-result-anchor" id="job33748850"></span>
        <div class="details col-sm-12 col-md-9 col-lg-10">
            <header>
                <div class="badge-container visible-xs-block">
```



```
<span class="label label-featured"  >Featured</span>
```

Position: ^ $ \b \B

- `^Hello` that word at the **start** of a line
- `Hello$` that word at the **end** of a line
- `\bHe` at **beginning** of a **word**
- `\BHe` at **end** of a **word**
- When inside brackets ^ means NOT: `[^aeiou]` = NOT a vowel

OR: the pipe |

- Put between 2 or more expressions
- `st|th|nd|rd` Either “st” or “th” or “nd” or “rd” (no spaces)
- Brackets can also be used to group characters: `(st)|(th)|(nd)|(rd)`

Escaping special characters: \

`\.`	Match an actual full stop
`\+`	A plus sign
`\?`	A question mark
`\\`	A backslash
`\*`	An asterisk
`\$`	A dollar sign
`\[`	A square bracket
`\(`	A parenthesis

Why would we look for this pattern?

- `[tsnr][htd]`

t or s or n or r
Followed by
h or t or d

Clue…

- `[0-9]+[tsnr][htd]`

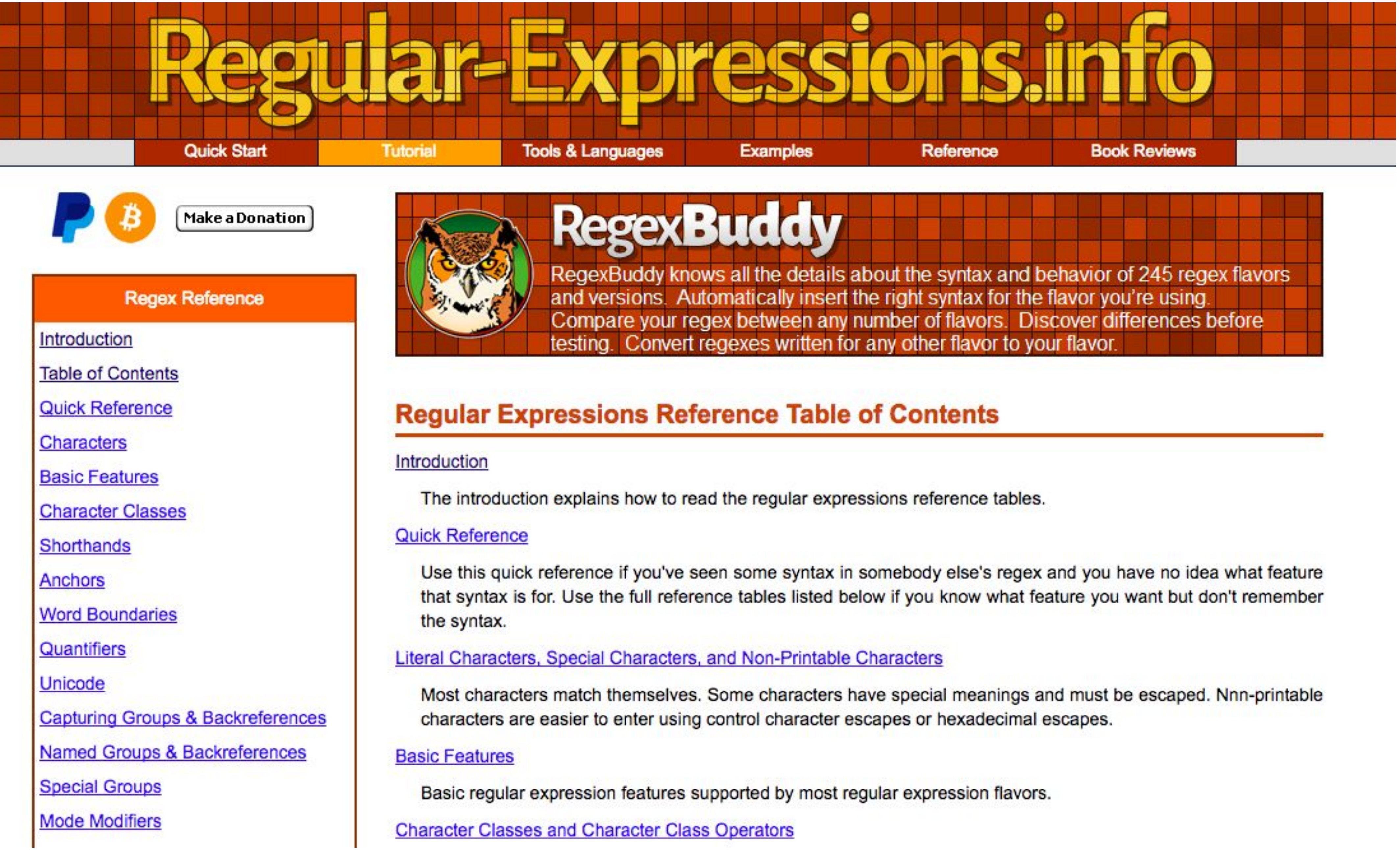
A number (one or more)
Followed by
t or s or n or r
Followed by
h or t or d

Examples

- `"hello"` Look for “hello”
- `"[0-9][0-9]th September"` Look for any 2 digits followed by “th September”

Used in R, Python etc.

- `regexpr` and `gregexpr` functions in R
- `re` module in Python



RegexBuddy knows all the details about the syntax and behavior of 245 regex flavors and versions. Automatically insert the right syntax for the flavor you're using. Compare your regex between any number of flavors. Discover differences before testing. Convert regexes written for any other flavor to your flavor.

Regular Expressions Reference Table of Contents

Introduction

The introduction explains how to read the regular expressions reference tables.

Quick Reference

Use this quick reference if you've seen some syntax in somebody else's regex and you have no idea what feature that syntax is for. Use the full reference tables listed below if you know what feature you want but don't remember the syntax.

Literal Characters, Special Characters, and Non-Printable Characters

Most characters match themselves. Some characters have special meanings and must be escaped. Nnn-printable characters are easier to enter using control character escapes or hexadecimal escapes.

Basic Features

Basic regular expression features supported by most regular expression flavors.

Character Classes and Character Class Operators

Reference:

https://www.regular-expressions.info/refflavors.html

Summary:

- Regex is hugely useful for identifying patterns you want to look for and grab
- Trial and error: try modifiers
- Can always clean parts later