

LIGHTNING TALK

RegEx

Expresiones

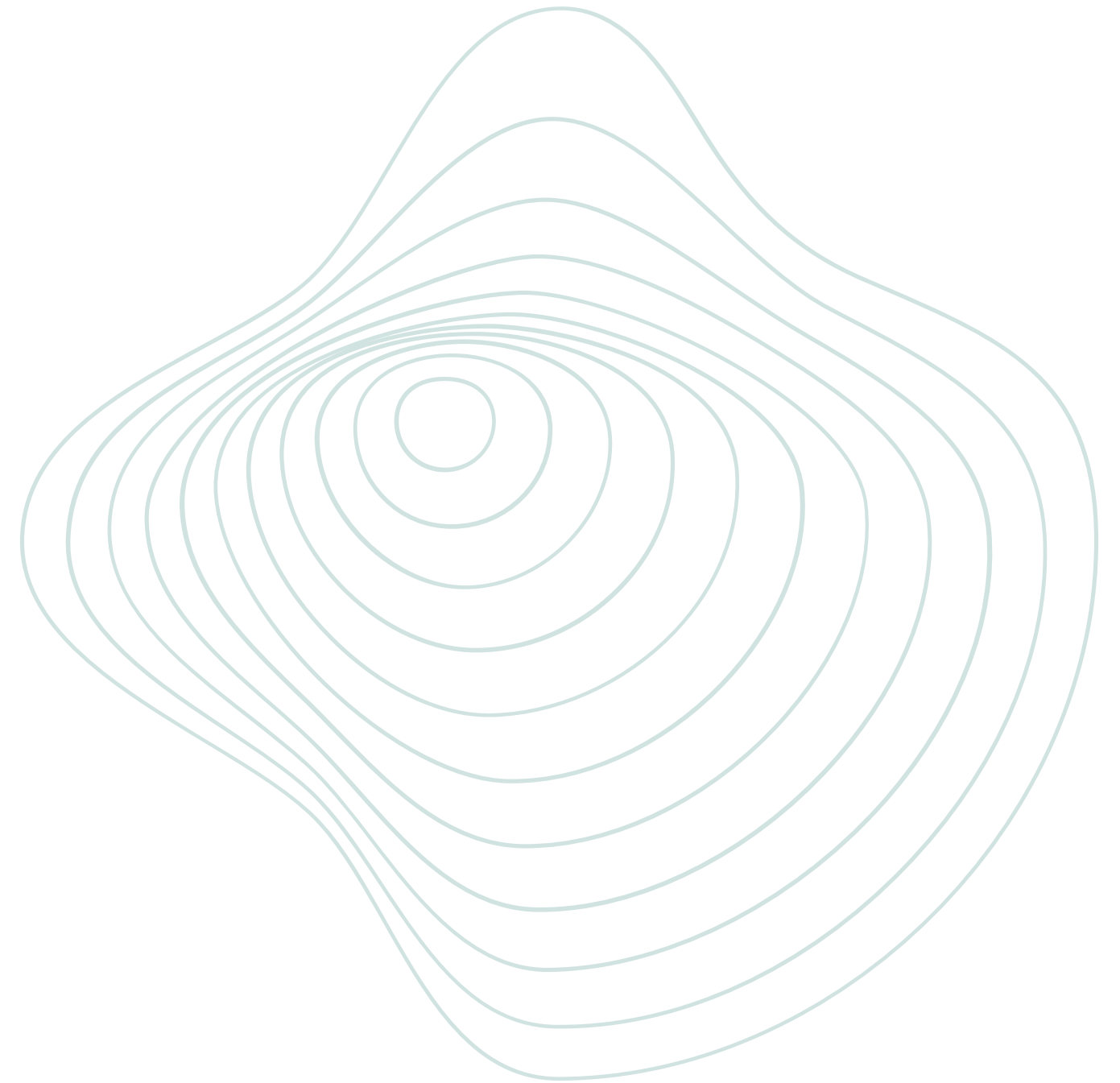
Regulares

lunes 11, abril 2022 - Sonia Reyes

¿Qué son?

Una manera muy sencilla de describir secuencias de caracteres en nuestro código.

Por lo general, el objetivo es modificar o sustituir fragmentos de texto o líneas de código, cuya complejidad aumenta en función de las veces que la secuencia de caracteres aparece en el documento.





mmm, igual suena un
poco enrredado!

Descubriendo RegEx

| checking si una string contiene solo números + javascript



stackoverflow About Products For Teams Search...

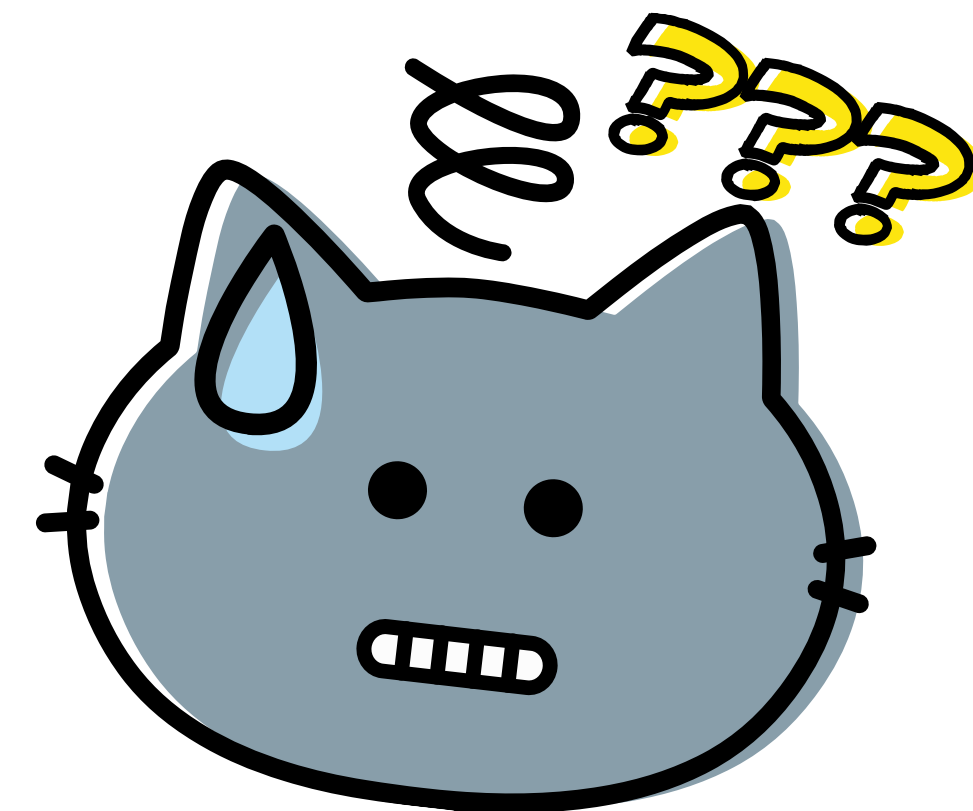
Home

PUBLIC

Questions

Check if string contains only digits

Asked 12 years, 4 months ago Modified 1 year ago Viewed 487k times



```
var isNumber = /^\d+\.\d+$/ .test(value);
```

```
const reg = new RegExp('^[0-9]+$');
```

or

```
const reg = new RegExp('^\\d+$')
```

check if string only contains integer digits numbers javascript

```
1 let val = '1234567890'
2 let isnum = /^\d+$/ .test(val);
3 if (isnum) {
4   // val is a number
5 }
```

Explanation of the regex:

```
/^-?(\d+\.\d*|(\d*\.\d+)\d*)$/
```

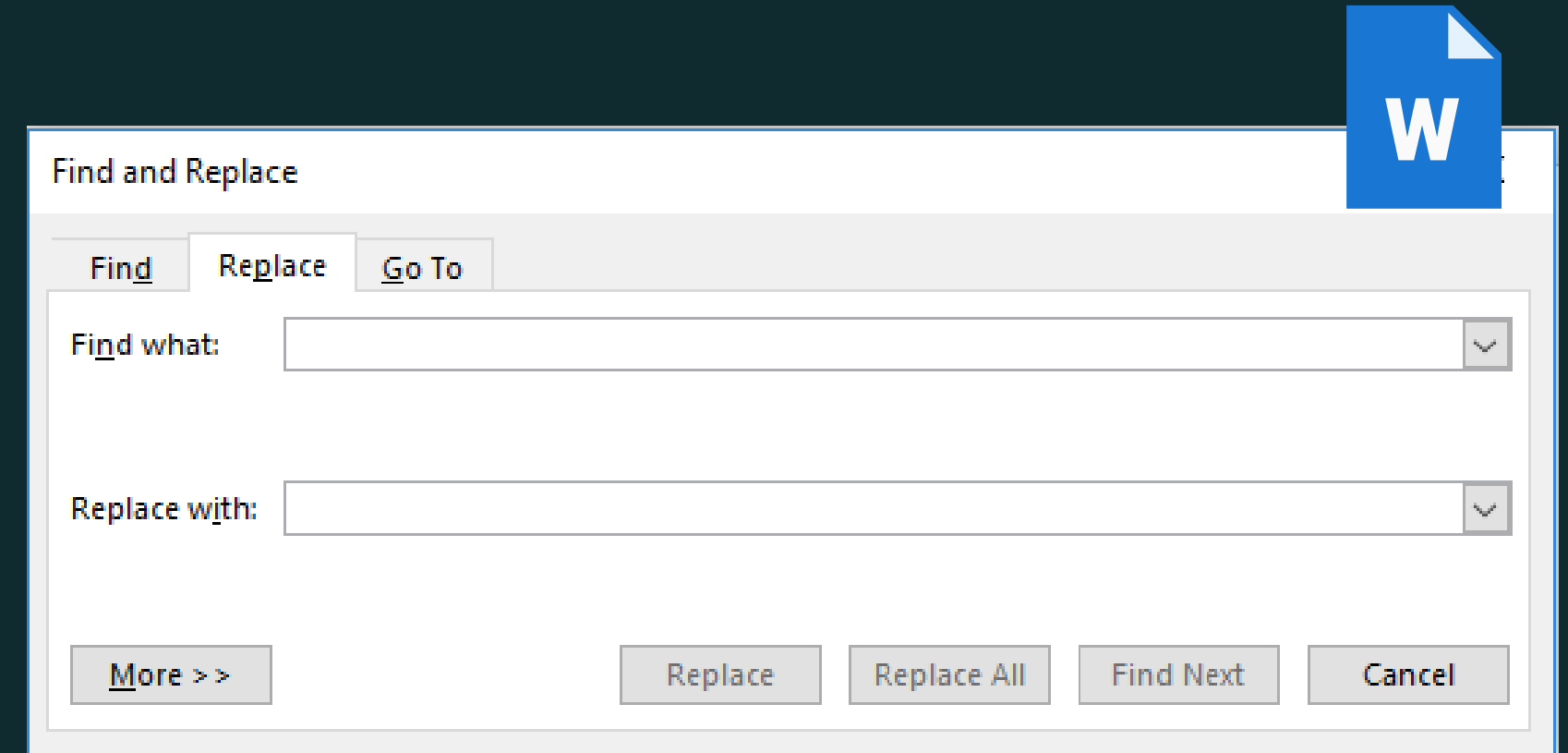
Entonces...

RegEx son patrones que se utilizan para hacer coincidir combinaciones de caracteres en un texto.

El uso de RegEx no es solo en lenguajes de programación como JS, Python, PHP, sino también en bases de datos como MySQL, Oracle, PostgreSQL.

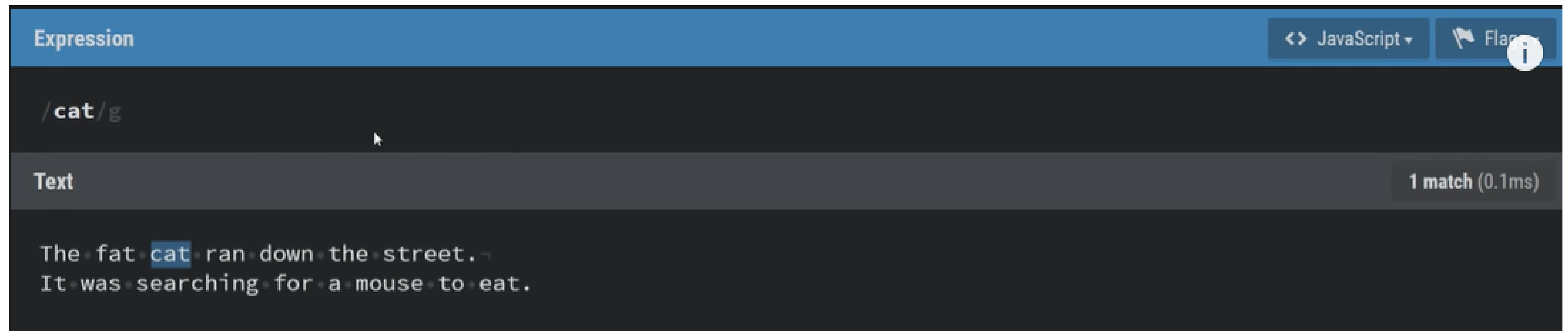
En JavaScript, se utilizan con los métodos `exec()` y `test()` de `RegExp`, y con `match()`, `matchAll()`, `replace()`, `replaceAll()`, `search()` y `split()` métodos de `String`.

Fuente: Developer Mozilla



Nos ayudan a **buscar, hacer coincidir, reemplazar, subcadenar, analizar y dividir una string, por ejemplo.**

Pensemos en un ejemplo sencillo en un texto:



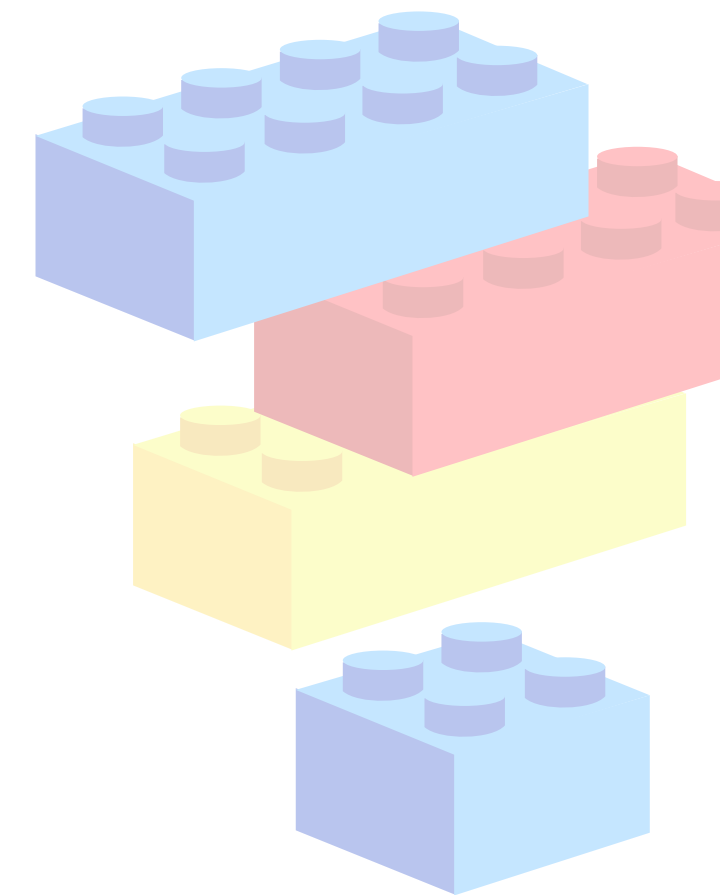
cat sería mi RegEx, escrito dentro de `/ /`
g --> flag, indicador-->búsqueda global--> todas las coincidencias de cat dentro de mi texto.

¿Qué contiene un RegEx?

Caracteres

| | |
|---|--|
| . | Coincide con cualquier carácter único excepto terminadores de línea: <code>\n</code> , <code>\r</code> , <code>\u2028</code> o <code>\u2029</code> . |
| <code>\d</code> , o <code>[0-9]</code> | Busca coincidencias para cualquier dígito |
| <code>\D</code> | Busca cualquier caracter que no sea un dígito |
| <code>\w</code> o <code>[A-Za-z0-9_]</code> | Busca cualquier caracter alfanumérico del alfabeto latino básico. |
| <code>\s</code> | Busca un espacio en blanco |

Fuente: Developer Mozilla



anchors/Límites

Busca posiciones, en lugar de caracteres.

| | |
|----|---|
| ^ | Coincide con el comienzo de una entrada, por ejemplo, una string. |
| \$ | Coincide con el final de una entrada, por ejemplo, una string. |
| \b | Marca el límite de una palabra |

Flags/Indicadores

Las expresiones regulares tienen seis indicadores opcionales que permiten funciones como la búsqueda global y que no distinga entre mayúsculas y minúsculas.

| | |
|---|--|
| g | Búsqueda global. |
| i | Búsqueda que no distingue entre mayúsculas y minúsculas. |
| m | Búsqueda multilínea. |

Moraleja:

No temerles, nos ayudan a ahorrar tiempo!

Regex nos servirá para buscar, analizar y validar un input.

Match an Email Address

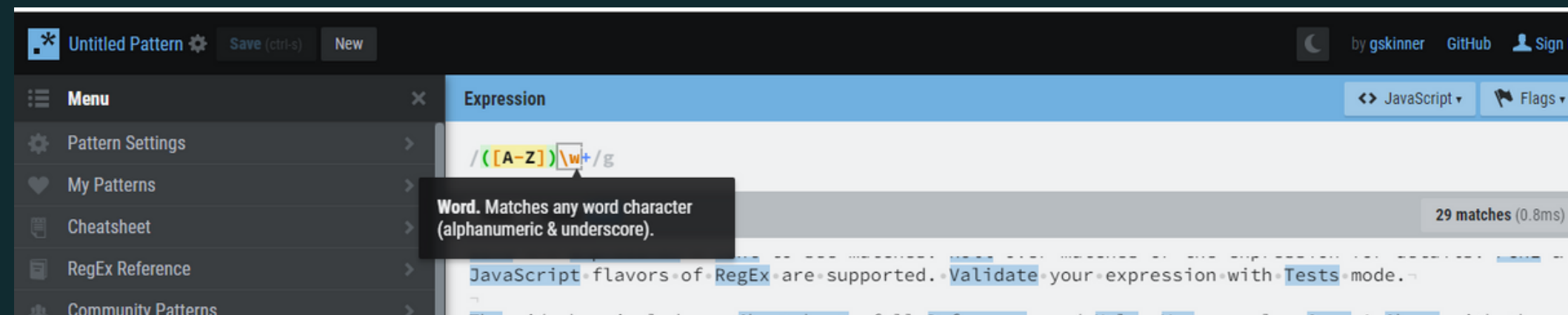
```
\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b
```

Match a Valid Date

MM/DD/YYYY

```
^(0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])[- /.](19|20)\d\d$
```

Recursos



<https://regexr.com/>

learnbyexample Books Mini Tips Tags About

JavaScript regular expressions cheatsheet and examples

2020-07-20

RegExp: `/\b(\w+)(\1)+\b/`

The diagram illustrates the components of the regular expression `/\b(\w+)(\1)+\b/`. It starts with a green dot representing the start of the string, followed by a purple box labeled 'WordBoundary!'. This is followed by 'Group #1', which is a green box labeled 'Word' with a red circle around it and the text '1 or more times' below it. This is followed by 'Group #2', which is a blue box labeled 'Backref #1' with a red circle around it and the text '1 or more times' below it. Finally, there is another purple box labeled 'WordBoundary!' followed by a black dot representing the end of the string.

Above diagram created using [Regex](#)

<https://learnbyexample.github.io/javascript-regexp-cheatsheet/>