

This is your **last** free member-only story this month. [Sign up for Medium and get an extra one](#)

# Regex cookbook — Top 10 Most wanted regex

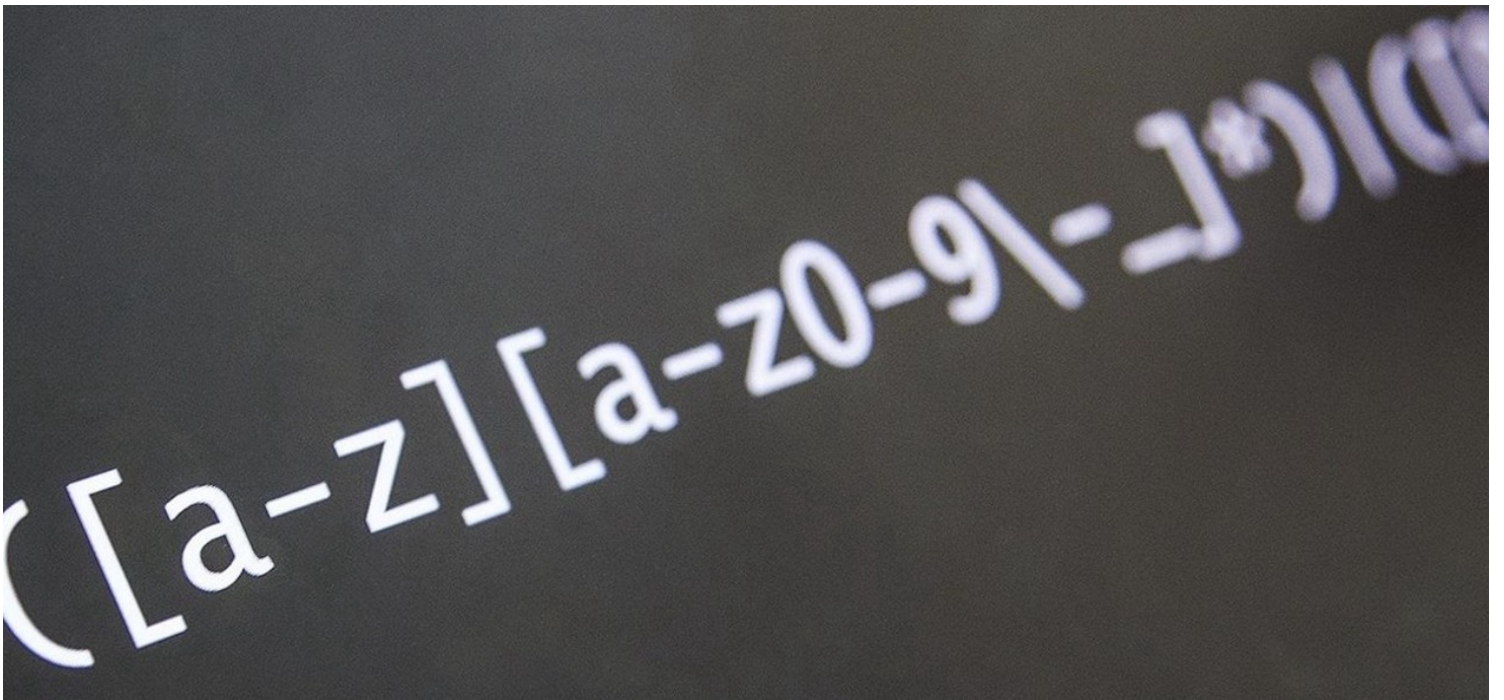


Jonny Fox

Follow

Mar 19, 2019 · 4 min read ★

The most commonly used (and most wanted) regexes



After some time I thought about publishing this article ([see my previous tutorial](#)) with the regex I used most in the projects on which I was involved.

Write in the comments any regex that you would like to add and I will try (if I can) to implement them 🙏

Happy coding!!!

. . .

## Trim spaces — try it!

Matches text avoiding additional spaces

```
:/ ^[\s]*(.*?)[\s]*$
```

/gm

test

avoid starting spaces

avoid ending spaces

avoid starting and ending spaces

```
^[\s]*(.*?)[\s]*$
```

. . .

## HTML Tag — try it!

Matches any valid HTML tag and the **corresponding closing tag**

```
:/ <([a-z]+)([^\<]+)*(?:>|.*<\/\1>|\/>)
```

/gm

<a>match only <b>the corresponding</b> tag</a> even if there is other <b> text </b>

```
<([a-z]+)([^\<]+)*(?:>|.*<\/\1>|\/>)
```

. . .

## Hexadecimal value — try it!

Matches any valid **hex color** inside text

```
:/ \B#([a-fA-F0-9]{6}|[a-fA-F0-9]{3})\b
```

/g

```
#fff
#fefefe
#242424
#24ff33
#24ff3o //o is not a valid hex value
```

```
\B#(?:[a-fA-F0-9]{6}|[a-fA-F0-9]{3})\b
```

. . .

## Valid email (RFC5322) — try it!

Matches any valid **email** inside text

```
:/ \b[\w.!\#$%&'*+\/=?^`{|}~-]+@[ \w-]+(?:\.[ \w-]+)*\b
```

/mg

```
Matches an email address like john.doe@my-domain.com inside text
valid-email@email.com
not!valid@email.com
```

```
\b[\w.!\#$%&'*+\/=?^`{|}~-]+@[ \w-]+(?:\.[ \w-]+)*\b
```

. . .

## Username (simple) — try it!

Minimum length of 3, maximum length of 16, composed by letters, numbers or dashes.

```
:/ ^[a-z0-9_-]{3,16}$
```

/m

```
not-valid user
valid-user
```

```
/^[a-z0-9_-]{3,16}$/
```

. . .

## Strong password — try it!

Minimum length of 6, at least one uppercase letter, at least one lowercase letter, at least one number, at least one special character

```
:/^(?=^.{6,}$)((?=.*\w)(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[!'"$%&\/\(\)\ \^\'\\\\+\\-\\*]))^.*
```

```
my valid 2passwoRd+
my missing number passwoRd+
MY MISSING LOWERCASE LETTER 2PASSWORD+
my missing uppercase letter 2passwod+
my missing special character 2passwoRd
not valid password
```

```
(?=^.{6,}$)((?=.*\w)(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[!'"$%&\/\(\)\ \^\'\\\\+\\-\\*]))^.*
```

. . .

## 2 of a kind — try it!

At least 2 letters (uppercase or lowercase) at any index, minimum length of 8, maximum length of 32

```
:/^(?=[0-9]*[a-z]){2,})([a-zA-Z0-9]{8,32})$
```

```
1a2b3cd2312321
12a321312312314
```

[illegible]

```
^(?=[0-9]*[a-z]){2,}([a-zA-Z0-9]{8,32})$
```

If you want to use capturing groups to get scheme, path, etc. (or add user-info, host, port...) feel free to ask it in comments!

```
^((https?|ftp):\/\/)?([\w\-\.])([\w]){2,4}([\w\/+=%&_\.\~?|-]*)*$
```

```
http://www.example.com/index.html
www.example.com/index.html
www.example.com
example.com
example not valid
http://xn--fsqu00a.xn--3lr804guic/
http://example.com/%E5%BC%95%E3%81%8D%E5%89%B2%E3%82%8A.html
```

```
^(((https?|ftp):\/\/)?([\w\-\.])(\.)([\w]){2,4}([\w\/+=%&_\.\~?
\-*]))*$
```

## IPv4 address — try it!

Matches any valid IPv4 address inside text

```
:/\b(?:?:25[0-5])|2[0-4]\d|([01]?[d\d?])\.){3}(?:25[0-5])|2[0-4]\d|([01]?[d\d?])\b /gm
```

```
192.168.0.1
255.255.255.0
0.0.0.0
127.0.0.1
256.0.0.1
there is my IP address (89.35.35.0) inside this text
```

```
\b(?: (?:25[0-5]|2[0-4]\d|[01]?\d\d?)\.){3}(?:25[0-5]|2[0-4]\d|[01]?\d\d?)\b
```

. . .

## “Defanged” URL or IPv4 address — try it!

```
^/^(?!(h..ps?|f.p):\\\/)?(?:([w\-\.\-])+(\[?\.\\]?)([w]){2,4}|(?:25[0-5]|2[0-4]\d|[01]?\d\d?)(\[?\.\\]?){3}(?:25[0-5]|2[0-4]\d|[01]?\d\d?)))(\[?w\+=%&_\.\~?\\-]*$
```

```
www[.]domain[.]com
www.domain.com
hxxp://www[.]domain[.]com
192.168.8.1/folder/file.html
192.168.1[.]1
192.168.1[.]1/folder/file.html
192[.]168[.]1[.]1/folder/file.html
hxxp://www[.]domain[.]com/folder/file.html
```

```
^(((h..ps?|f.p):\\\/)?(?:([w\-\.\-])+(\[?\.\\]?)([w]){2,4}|(?:25[0-5]|2[0-4]\d|[01]?\d\d?)(\[?\.\\]?){3}(?:25[0-5]|2[0-4]\d|[01]?\d\d?)))(\[?w\+=%&_\.\~?\\-]*$
```

. . .

## SSN — Social Security Number (simple) — try it!

If you want to check the validity of an SSN feel free to ask in comments!

```
^((?<area>[d]{3})[-][\d]{2}[-][\d]{4})$
```

721-07-4426  
000-00-0000

```
^((?<area>[\d]{3})[-][\d]{2}[-][\d]{4})$
```

. . .

Alpha-numeric, literals, digits, lowercase, uppercase chars only

```
\w          //alpha-numeric only  
[a-zA-Z]    //literals only  
\d          //digits only  
[a-z]       //lowercase literal only  
[A-Z]       //uppercase literal only
```

Have fun and do not forget to recommend the article if you liked it ❤️

Do not forget to check out my previous article about [regex tutorial!](#)

Programming

Regex

Java

Python

JavaScript



[About](#) [Help](#) [Legal](#)

Get the Medium app

