



Published in Factory Mind

This is your **last** free member-only story this month.

[Sign up for Medium and get an extra one](#)



Jonny Fox

Follow

Mar 20, 2019 · 4 min read · ✨ · 🎧 Listen

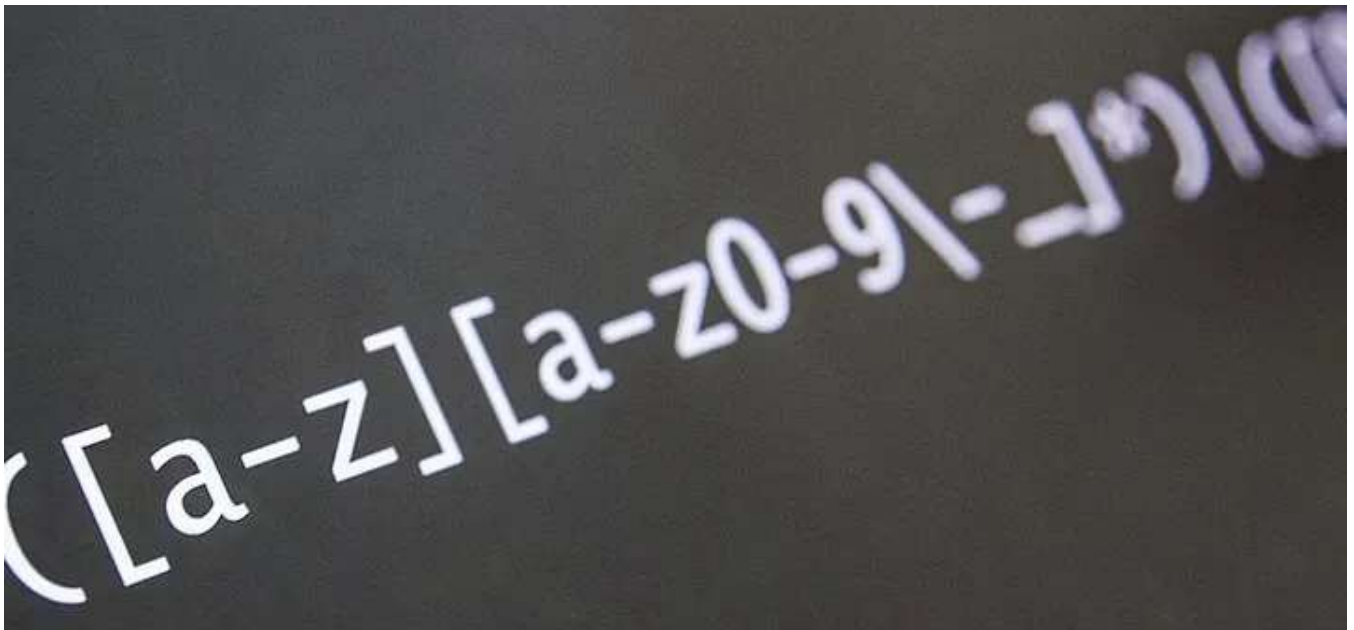


Save



# Regex cookbook — Top 15 Most common regex

The most commonly used (and most wanted) regexes



**UPDATE 10/2022:** See further explanations/answers in story responses!

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!!!

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

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

### Simple numbers — try it!

Matches simple numbers only (no decimal nor fractions)

```
^(\d+)\$.      15/12  8.5  12
```

### Decimal numbers — try it!

```
^(\d*)[.,](\d+)\$  15/12  8.5  12  8,7
```

### Fractions — try it!

```
^(\d+)[\ /](\d+)\$  15/12  8.5  12
```

### Alphanumeric without spaces — try it!

```
^(\w*)$          hello123
                  but not this text
```

### Alphanumeric with spaces — try it!

Open in app ↗

Sign up

Sign In



## Email (simple — see advanced section for more)

Classic email — [try it!](#)

```
^([a-zA-Z0-9._%~]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})*$
```

jonny.fox@factorymind.com  
hello@sdasdad.hello  
but not this!

Email tokens — [try it!](#)

```
^([a-z0-9_\.\\+~]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$
```

jonny.fox@factorymind.com

## Advanced Regex

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]*$
```



2.1K



15

## HTML Tag — try it!

Matches any valid HTML tag and the **corresponding closing tag** (thanks to [Danilo Gomez Gomez](#) for the suggestions!)

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

```
<a>match only <b>the corresponding</b> tag</a> even if there is other <b>text</b>
<a>123</a> <a>456</a> // Thanks to Danilo Gomez Gomez
```

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

## Hexadecimal value — try it!

Matches any valid hex color inside text

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

```
#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}$
```

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

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

**Strong password — try it!**

Minimum length of 6, at least 1 uppercase letter, at least 1 lowercase letter, at least 1 number, at least 1 special character

```
#!/usr/bin/perl -e 'print "1" if (/^(?=[^6$])(?=[^\w])(?=[^A-Z])(?=[^a-z])(?=[^0-9])(?=[^!"$%&\'\/\(\)\?\\\^*\\\\+|-|*]))^$/);'
```

```
my valid 2password+
my missing number password+
MY MISSING LOWERCASE LETTER 2PASSWORD+
my missing uppercase letter 2password+
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})$
```

[illegible]

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

## URL tokenization—try it!

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
```

## URL or IPv4 address — try it!

```
/ ^(((https?|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\/+=%&_\.\~?\-\])* $ /gm
```

```
www[.]domain[.]com
www.domain.com
http://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
http://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\/+=%&_\.\~?\-\])* $
```

```
\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})$
```

/gm

```
721-07-4426
000-00-0000
```

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

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](#)
[Terms](#)
[Privacy](#)

Get the Medium app

