

Top 15 Commonly Used Regex

May 30, 2018 / Javascript, Tips & Tricks / Niket Pathak / 3 min read

Regex

Regular Expressions aka Regex are expressions that define a search pattern. They are widely used for validation purposes, like email validation, url validation, phone number validation and so on.

Forming Regex:

A regex is written between two **forward slashes** (/) delimiters. These delimiters are essential only for certain Regex engines (JS,PHP..). It is then followed by 2 flags:

- **i** : Ignore case Flag, ignores the case(uppercase/lowercase) of the input string
- **g** : Global Flag, searches for multiple matches instead of stopping at the first match

```
1 /* Regex Example */
2 /hello/ig
3 /* will match the word "hello" irrespective of its case and
4 will return all found matches. */
```

A quick cheat sheet for those of you who do not remember all the rules of Regex



Cheat Sheet – Regex Rules

Character classes		Quantifiers & Alternation	
.	any character except newline	a* a+ a?	0 or more, 1 or more, 0 or 1
\w \d \s	word, digit, whitespace	a{5} a{2,}	exactly five, two or more
\W \D \S	not word, digit, whitespace	a{1,3}	between one & three
[abc]	any of a, b, or c	a+? a{2,}?	match as few as possible
[^abc]	not a, b, or c	ab cd	match ab or cd
[a-g]	character between a & g		
Anchors		Groups & Lookaround	
^abc\$	start / end of the string	(abc)	capture group
\b	word boundary	\1	backreference to group #1
Escaped characters		(?:abc)	
\.*\\\	\ is used to escape special chars. * matches *	(?=abc)	positive lookahead
\t\n\r	tab, linefeed, carriage return	(?!abc)	negative lookahead

Commonly used Regex

1. Digits

- Whole Numbers – `/^\d+$/`
- Decimal Numbers – `/^\d*\.\d+$/`
- Whole + Decimal Numbers – `/^\d*(\.\d+)?$/`
- Negative, Positive Whole + Decimal Numbers – `/^-?\d*(\.\d+)?$/`
- Whole + Decimal + Fractions – `/[-]?[0-9]+[.,]?[0-9]*([/\][0-9]+[.,]?[0-9]*)*/`

2. Alphanumeric Characters

- Alphanumeric without space – `/^[a-zA-Z0-9]*$/`
- Alphanumeric with space – `/^[a-zA-Z0-9]*$/`

3. Email

- Common email Ids – `/^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})*$/`
- Uncommon email ids – `/^([a-z0-9_\.+-]+)@([\da-zA-Z\.-]+)\.([a-zA-Z]{2,6})$/`

Meet Return
Shipping
on Us.*

Learn More

*Activation required.
Limitations apply.



Recent Posts

- Smooth Scroll in native Javascript/css
- Lint on precommit Git hook made easy
- Can I use font-weight: 1000 ?
- Duplicate a Mysql Database - Tutorial
- How to use Webpack Analyzer Bundle

Tags



4. Password Strength

- **Complex:** Should have 1 lowercase letter, 1 uppercase letter, 1 number, 1 special character and be at least 8 characters long
`/^(?=.*[0-9])(?=.*[!@#$%^&*)(\\[\\]{})\\-_+=~`|:;''<>,./?])^(?=.*[a-z])(?=.*[A-Z]))^(?=.*(.)).{8,}/`
- **Moderate:** Should have 1 lowercase letter, 1 uppercase letter, 1 number, and be at least 8 characters long
`/^(?=.*[0-9])(?=.*[A-Za-z0-9])(?=.*[A-Z])(?=.*[a-z]))^.{8,}$/`

5. Username

- **Alphanumeric string** that may include _ and - having a length of 3 to 16 characters –
`/^[_a-zA-Z0-9_-]{3,16}$/`

6. URL

- Include **http(s)** Protocol

```
/https?:\/\/((www\.)?[-a-zA-Z0-9@:%._\+~#=]{2,256}\.?[a-zA-Z]{2,6}\b([-a-zA-Z0-9@:%_\+.~#()?&/=]*)/
```

- **Protocol Optional**

```
/(https?:\/\/)?(www\.)?[-a-zA-Z0-9@:%._\+~#=]{2,256}\.?[a-zA-Z]{2,6}\b([-a-zA-Z0-9@:%_\+.~#?&/=]*)/
```

7. IP Address

- **IPv4 address**
- **IPv6 address**
- Both **IPv4, IPv6 addresses**

```
1 /* Match IPv4 address */
2 /*((0-9|[1-9][0-9]|1[0-9][2]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|1-9)[0-9]1[0-9]{2}|2[0-4][0-9]|25[0-5])
3 /* Match IPv6 address */
4 /(([0-9a-fA-F]{1,4}):{7,7}[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,5}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,4}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,3}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,2}:[0-9a-fA-F]{1,4}|[0-9a-fA-F]{1,4}:){1,1}:[0-9a-fA-F]{1,4})
5 /* Match both IPv4, IPv6 addresses */
6 /((^\s*((([0-9|[1-9][0-9]|1[0-9][2]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|1-9)[0-9]1[0-9]{2}|2[0-4][0-9]|25[0-5]))|([0-9a-fA-F]{1,4}:){7,7}[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,5}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,4}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,3}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,2}:[0-9a-fA-F]{1,4}|[0-9a-fA-F]{1,4}:){1,1}:[0-9a-fA-F]{1,4})
```

8. Dates

- Date Format **YYYY-MM-dd** using separator `-`
- Date Format **dd-MM-YYYY** using separators `-` or `.` or `/`
- Date Format **dd-mmm-YYYY** using separators `-` or `.` or `/`

```
1 /* Date Format YYYY-MM-dd */
2 //([12]\d{3}-0[1-9]|1[0-2])-(0[1-9]|1[0-2]\d|3[01]))/
3
4 /* Date Format dd-MM-YYYY or
5 dd.MM.YYYY or
6 dd/MM/YYYY
7 with check for leap year */
8 /(^:(?:31(\//|-|\.)?:0?:[13578]|1[02]))\1(?::(?:29|30)(\//|-|\.)?:0?:[1,3-9]|1[0-2])\2)(?::(?:1[6-9]
9
10 /* Date Format dd-mmm-YYYY or
11 dd/mmm/YYYY or
12 dd.mmm.YYYY */
13 /(^:(?:31(\//|-|\.)?:0?:[13578]|1[02])|(?:(Jan|Mar|May|Jul|Aug|Oct|Dec)))\1(?::(?:29|30)(\//|-|\.)?:0?:
```

9. Time

- Time Format **HH:MM 12-hour**, optional leading 0
`/^(0?:[1-9]|1[0-2]):[0-5][0-9]$/`
- Time Format **HH:MM 12-hour, optional leading 0, Meridiems (AM/PM)**
`/((1[0-2]|0?:[1-9]):([0-5][0-9])?:([AaPp][Mm]))/`
- Time Format **HH:MM 24-hour** with leading 0
`/^(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]$/`
- Time Format **HH:MM 24-hour, optional leading 0**
`/^(0-9|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]$/`
- Time Format **HH:MM:SS 24-hour**
`/(:[01]\d|2[0123]):(:[012345]\d):(:[012345]\d)/`

10. HTML Tags

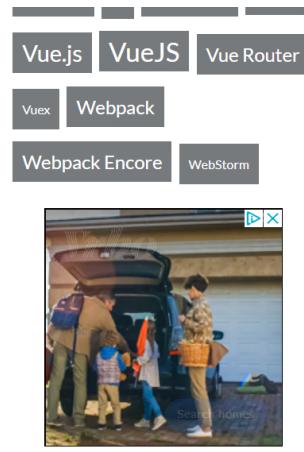
- **Elements with Attributes** `/</?[\w\s]*>|<.+[\W]>/`

11. Javascript Handlers

- **Inline JS handler** `/\bon\w+=\S+(?=,*)/`
- **Inline JS handler with element** `/(?:<[^>]+>)(on\s+)=["]?(?:.(?!["?])?\s+(?:\S+)=|[""])+.)([""])?/`

12. Slug

- **Slug** `/^[-a-zA-Z0-9]+(-[-a-zA-Z0-9]+)*$/`



A screenshot of a Best Buy product page for "Fast & Furious Crossroads" on Xbox One. The page shows the game cover, price (\$54.99), original price (\$79.99), and a "Shop Now" button. The background features a blue banner with the text "Score a savings touchdown." and the Best Buy logo.

13. Match Duplicates in a String

- [Search Duplicates](#) `/(\b\w+\b)(?=.*\b\1\b)/`

14. Phone Numbers

- [International Phone Numbers](#) – with optional country code/extension

```
1 /* International Phone Numbers */
2 /^(?:\(?:(?:\(?:(?:00|+)([1-4]\d\d|[1-9]\d?)\))?)?[-.\ \\\/]?)?((?:\(?:\d{1,}\)\)?[-.\ \\\/]?)\{0,\})(?:[\\"
```

Note: Use regex for validating phone numbers **only** if you don't have the choice to use a library. There are [several libraries](#) that handle phone numbers more accurately and should be used instead.

15. File Path

- [File Path with Filename and extension](#)

```
/(^|\\|\\/|https?:\\\/\/[a-zA-Z-_@-^!#%&+={}.\\\\[\]]+)+.[a-zA-Z]+$/
```

- [File Path with optional Filename, extension](#)

```
/^(.+)/(^\[^\]++$)/
```

- [File Name with extension having 3 chars](#)

```
/^[\w,\s-]+\.[A-Za-z]{3}$/
```

Additional Regexes

1. Zip codes

There is **NO** single Regex that can handle all zip codes given that zip codes around the world do not follow a common pattern. Here is a [list](#) that contains Regex specific to each country.

2. Payment Validation

Here is a [link](#) that contains regex for validating leading Credit cards like Visa, Mastercard and so on.

3. Identity Documents

- [Social Security Number – Ref](#)

```
/* can use either hyphen(-) or space( ) character as separator */
/^((?!219-09-9999|078-05-1120)(?!666|000|9\d{2})\d{3}-(?!\d{2})\d{2}-(?!\d{4})\d{4})|((?!219 09 9999|078 05 112
```

- [Passport](#) – `/^A-PR-WY[1-9]\d\s?\d{4}[1-9]$/`

References

- <https://www.codeproject.com/validate-passport-number>
- <https://gist.github.com/>
- <https://code.tutsplus.com/regular-expressions>
- <https://www.regular-expressions.info>
- <http://www.regexlib.com/>
- <https://projects.lukehaas.me/regexhub/>
- <http://wiki.zoolz.com/commonly-used-regular-expressions/>
- <https://www.smashingmagazine.com/advanced-regular-expressions/>

P.s. Need a **graphical explanation** of any of the above Regex ? Try [Regular Expression Visualization](#) and copy-paste any Regex in the input box to get a graphical representation of it. Hope that helps!

Share this:



Like this:



Be the first to like this.

ALSO ON DIGITAL FORTRESS



in the world or Javascript, aren't Promises just regular Callbacks? What are the ...

Step by step tutorial to configure JWT Authentication with ...

want to start using async generator function, spread operator and all the latest ...

Create click to call links easily with the use of the tel URI scheme. Supported ...

4 Comments Digital Fortress [🔒 Disqus' Privacy Policy](#)

[1 Login](#) ▾

[Heart Recommend 1](#)

[Tweet](#)

[Share](#)

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS [?](#)



Name



Greta Chesnokova • 2 years ago • edited

Thanks for this comprehensive list! My team has been developing a crawler, it has been quite helpful.

1 ⤵ | ⤴ • Reply • Share >



Niket Pathak [Mod](#) ➔ Greta Chesnokova • 2 years ago

you're welcome. :)

^ | ⤵ • Reply • Share >



Vladan • a year ago • edited

HTML Tags doesn't work with this input:

<https://regex101.com/r/tj42i6/1>

^ | ⤵ • Reply • Share >



Niket Pathak [Mod](#) ➔ Vladan • a year ago • edited

yes, you're right, reason being that you have concatenated your `span` tags on the same line. However, if the html is well formatted, it works as expected. Here's your updated link with the regex working -

<https://regex101.com/r/tj42...> (Unfortunately, concatenating html as you did will definitely break the regex.

The is one of the downsides of using a general regex. But for specific cases as yours, you could modify the regex to target `span` tags as needed.)

^ | ⤵ • Reply • Share >

[✉ Subscribe](#)

[>Add Disqus to your site](#)

[⚠ Do Not Sell My Data](#)

DISQUS



Pages

- [About](#)
- [Contact](#)
- [Technical Writers / Volunteers Needed](#)

Recent Posts

- [Smooth Scroll in native Javascript/css](#)
- [Lint on precommit Git hook made easy](#)
- [Can I use font-weight: 1000 ?](#)

Categories

- [Bugfix](#)
- [CSS](#)
- [Debug](#)
- [Javascript](#)

Meta

- [Register](#)
- [Log in](#)
- [Entries feed](#)
- [Comments feed](#)