Ultimate Regex Cheatsheet - KeyCDN Support

Updated on October 4, 2018

What Is Regex?

Regex, also commonly called Regular Expression, is a combination of characters that **define a particular search pattern**. These expressions can be used for matching a string of text, find & replace operations, data validation, etc. For example, with regex you can easily check a user's input for common misspellings of a particular word (e.g. h[ei][ei]g[th][th]).

This guide provides a regex cheatsheet that you can use as a reference when creating regex expressions. We will also go over a couple of popular regex examples and mention a few tools you can use to validate / create your regex expressions.

Regex Cheatsheet

Consult the following regex cheatsheet to get a quick overview of what each regex token does within an expression.

Regex Basics	Description
^	The start of a string
$	The end of a string
.	Wildcard which matches any character, except newline (\n).
|	Matches a specific character or group of characters on either side (e.g. a|b corresponds to a or b)
\	Used to escape a special character
a	The character "a"
ab	The string "ab"

Quantifiers	Description
*	Used to match 0 or more of the previous (e.g. xy*z could correspond to “xz”, “xyz”, “xyyz”, etc.
?	Matches 0 or 1 of the previous
+	Matches 1 or more of the previous
{5}	Matches exactly 5
{5, 10}	Matches everything between 5-10

Character Classes	Description
\s	Matches a whitespace character
\S	Matches a non-whitespace character
\w	Matches a word character
\W	Matches a non-word character
\d	Matches one digit
\D	Matches one non-digit
[\b]	A backspace character
\c	A control character

Special Characters	Description
\n	Matches a newline
\t	Matches a tab
\r	Matches a carriage return
\ZZZ	Matches octal character ZZZ
\xZZ	Matches hex character ZZ
\0	A null character
\v	A vertical tab

Groups	Description
(xyz)	Grouping of characters
(?:xyz)	Non-capturing group of characters

[xyz]	Matches a range of characters (e.g. x or y or z)
[^xyz]	Matches a character other than x or y or z
[a-q]	Matches a character from within a specified range
[0-7]	Matches a digit from within a specified range

String Replacements	Description
$`	Insert before matched string
$’	Insert after matched string
$+	Insert last matched
$&	Insert entire match
$n	Insert nth captured group

Assertions	Description
(?=xyz)	Positive lookahead
(?!xyz)	Negative lookahead
?!= or ?<!	Negative lookbehind
\b	Word Boundary (usually a position between /w and /W)
?#	Comment

Regex Examples

With the regex cheatsheet above, you can dissect and verify **what each token within a regex expression actually does**. However, you may still be a little confused as to how to put these tokens together to create an expression for a particular purpose. The following section contains a couple of examples that show how you can use regex to match a given string.

1. Matching an Email Address

To match a particular email address with regex we need to utilize various tokens. The following regex snippet will match a commonly formatted email address.

```
/^([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,5})$/
```

The first part of the above regex expression uses an `^` to start the string. Then the expression is broken into 3 separate groups.

- **Group 1** `([a-z0-9_\.-]+)` - In this section of the expression, we match one or more lowercase letters between a-z, numbers between 0-9, underscores, periods, and hyphens. The expression is then followed by an `@` sign.
- **Group 2** `([\da-z\.-]+)` - Next, the domain name must be matched which can use one or more digits, letters between a-z, periods, and hyphens. The domain name is then followed by a period `\.`.
- **Group 3** `([a-z\.]{2,5})` - Lastly, the third group matches the top level domain. This section looks for any group of letters or dots that are 2-5 characters long. This can also account for region-specific top level domains.

Therefore, with the regex expression above you can match many of the commonly used emails such as `firstname.lastname@domain.com` for example.

2. Matching a Phone Number

To use regex in order to search for a particular phone number we can use the following expression.

```
/^\b\d{3}[-.]?\d{3}[-.]?\d{4}\b$/
```

This expression is somewhat similar to the email example above as it is broken into 3 separate sections. Once again, to start off the expression, we begin with `^.`

- **Section 1** `\b\d{3}` - This section begins with a word boundary to tell regex to match the alpha-numeric characters. It then matches 3 of any digit between 0-9 followed by either a hyphen, a period, or nothing `[-.]?`.
- **Section 2** `\d{3}` - The second section is quite similar to the first section, it matches 3 digits between 0-9 followed by another hyphen, period, or nothing `[-.]?`.
- **Section 3** `\d{4}\b` - Lastly, this section is slightly different in that it matches 4 digits instead of three. The word boundary assertion is also used at the end of the expression. Finally, the end of the string is defined by the `$`.

Therefore, with the above regex expression for finding phone numbers, it would identify a number in the format of 123-123-1234, 123.123.1234, or 1231231234.

Regex Tools

There are a few tools available to users who want to verify and test their regex syntax. The following list provides tools you can use to **verify, create, and test regex expressions**. All tools more or less perform the same functionality, however you may find one that you prefer over another.

- Regexr - This tool allows you to input your regex expression as well as your text string and it will tell you if any matches are returned. You can also hover over each section of your regex expression to get an explanation of what each part is doing. Additionally, this tool also comes with a small regex cheatsheet and a few examples you can consult from the sidebar.

- Regex101 - Regex101 is quite comprehensive in terms of explaining what the regex expression is doing as

well as identifying which parts of the test string correspond to which sections of the regex expression. You can consult the regex cheatsheet at the bottom of the page to verify which regex tokens you can use to achieve your desired outcome.

- RegexPal - This tool also allows you to input your regex expression along with a test string to see if you receive the results you expect. RegexPal also provides you with a larger list of regex examples as well as a regex cheatsheet for reference.

If you're using regex in a web project and would like a **quick reference to the regex tokens available**, use the regex cheatsheet above as well the tools mentioned to help simplify the regex expression building process.