

Discussion 2.1 – Regular Expressions

Eric McCool

Bellevue University - College of Science and Technology

Web330-339A Enterprise JavaScript II

Professor Krasso

January 12, 2023

This week we look at regular expressions, or RegExp (W3S, 2023), or regex (Chodnicki, 2019). What is a regular expression? “A regular expression is a pattern of characters. The pattern is used to do pattern-matching ‘search-and-replace’ functions on text” (W3S, 2023)

What is the point of a regex? What are its advantages? With a regex, one can (Chodnicki, 2019):

- verify the structure of strings
- extract sub-strings from structured strings
- search/replace/rearrange parts of a string
- split a string into tokens

Well, that sounds pretty handy. What are some disadvantages to using a regex?

Marchete from Codingame (Marchete, n.d.) suggests:

- “Chaotic, evil syntax. Depending on the situation, the same metacharacter has many different meanings, which makes reading a regex a complicated task. For example, the ‘?’ . First it's a metacharacter for 0 or 1 repetitions, but suddenly it's also used as a lazy quantifier. But wait! As if two different meanings aren't enough, inside a parenthesis (? has more than 10 different meanings!: Non-capturing groups, named groups, look-ahead and look-behind, conditionals, recursion.... And this same thing happens with many other metacharacters.”
- “Regex expressions could have bad performance in some instances. Unbounded repetitions can match a string in many different ways, and regex engines usually need to do many steps and backtracking to find all of these matches.”

- “Regular Expressions are not suited for very complex, recursive data formats, like XML or HTML. In these cases, it's better to use an XML parser.”
- “There are many different regex engines, and each one has a different syntax. Therefore depending on the language, you need to learn some particular flags and metacharacters.”

Brackets are used when creating a regex to “find a range of characters” (W3S, 2023). For instance, to find any digit between 0 and 9, one would write [0-9] in brackets. The expression [abc] means find any character between the brackets (W3S, 2023).

Modifiers can also be used to further fine-tune a regex. Using ‘g’ for global, ‘i’ for case-insensitive, and ‘m’ for multi-line matching, they are used to perform case-insensitive and global searches. Global searches find all matches and do not stop after the first match (W3S, 2023).

Here is an example of a regular expression in JavaScript that validates a string value that contains alphanumeric characters only (W3S.blog, 2023):

<code>/^[0-9a-zA-Z]+\$</code>	>> the ‘/’ represent the regex body
<code>^[0-9a-zA-Z]+\$</code>	>> the ‘^’ is asking if it’s ‘not’ and the ‘\$’ runs this at the end of each string, I think.
<code>[0-9a-zA-Z]</code>	>> ‘[]’ makeup the subset(s) we are looking for
<code>0-9a-zA-Z</code>	>> the tests: 0 thru 9,
	a thru z (lower-case),
	and A thru Z (upper-case)

Translation: Run this regex at the end of each string and point out any non-alphanumerics.

References

- Chodnicki, S., 2019. *"Everything You Need to Know About Regular Expressions."* Medium, 2023. <https://towardsdatascience.com/everything-you-need-to-know-about-regular-expressions-8f622fe10b03>
- Marchete, (n.d.). *"Regular Expressions Basics – Final Words."* Codingame.com, 2023. <https://www.codingame.com/playgrounds/218/regular-expressions-basics/next-steps>
- W3S, 2023. *"JavaScript RegExp Reference."* JavaScript Reference – RegExp. W3Schools.com. W3Schools, Larsamyraa 18, 4313 SANDNES, Norway. https://www.w3schools.com/jsref/jsref_obj_regexp.asp
- W3S.blog, 2023. *"alphanumeric validation JavaScript JS."* W3Schools.blog, 2023. <https://www.w3schools.blog/alphanumeric-validation-javascript-js>