In this thread I want to discuss regular expressions in JavaScript. I will discuss the advantages and disadvantages of regular expressions. I will also describe how brackets are used in regular expressions and define what [0-9] means as well as what [abc] means. In this thread I will also describe what modifiers are in expressions and I will also write a regular expression in JavaScript that validates a string value that contains alphanumeric characters only. So, what is a regular expression in JavaScript?

“Regular expressions are patterns used to match character combinations in strings. In JavaScript, regular expressions are also objects.” (*Regular Expressions - JavaScript | MDN*, 2022). Regular expressions are used to verify the structure of strings, extract substrings from structured strings, (search/replace/ rearrange) parts of the string, and to split a string into tokens. (Chodnicki, 2021). Some of the advantages and disadvantages of regular expressions are:

|  |  |
| --- | --- |
| **Regular Expressions** | |
| **Advantages** | **Disadvantages** |
| Wide range of use possibility, you can create one regular expression to validate any kind of input. | Inefficient if poor expressions are written. |
| Supported by not only JavaScript, but almost any other programming language as well. | Cannot parse HTML. |
| Able to do more with less, which allows developers to write cleaner code. | Complex regular expressions can be hard to understand, debug, and maintain. |
| Faster validations, instead of having many IF and Else operators you may validate only once with a regular expression. | Can lead to catastrophic backtracking. |

“Brackets indicate a set of characters to match. Any individual character between the brackets will match. This means that [0-9] the expression is used to find any character between the brackets. The digits inside the brackets can be any numbers or span of numbers from 0 to 9.” (*JavaScript RegExp Group [0-9]*, 2022) The regular expression [abc] will search for the characters of (a,b, or c) in a string. (*JavaScript RegExp Group [Abc]*, 2022). Modifiers in regular expressions (also called flags) redefine regular expression behavior. Regular expression modifiers can regular or inline. The most common modifiers are global, case sensitive, multiline and dotall modifiers. (*Regular Expressions Tutorial - Regex Modifiers (Flags)*, n.d.) A regular expression in written form to validate a string value that contains alphanumeric characters only can be written as [a-zA-Z0-9] .

In conclusion we can see what regular expressions are in JavaScript. We can also see how to write a simple regular expression. We can see how the regular expression can be modified by using case sensitive lettering. Regular expression can be simple and easy to use in JavaScript.

References:

*Regular expressions - JavaScript | MDN*. (2022, August 9). Mdn Web Docs. Retrieved August 17, 2022, from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions>

Chodnicki, S. (2021, December 12). *Everything you need to know about Regular Expressions*. Medium. Retrieved August 17, 2022, from <https://towardsdatascience.com/everything-you-need-to-know-about-regular-expressions-8f622fe10b03>

*JavaScript RegExp Group [0-9]*. (2022). W3 Schools. Retrieved August 17, 2022, from <https://www.w3schools.com/jsref/jsref_regexp_0-9.asp>

*JavaScript RegExp Group [abc]*. (2022). W3 Schools. Retrieved August 17, 2022, from <https://www.w3schools.com/jsref/jsref_regexp_charset.asp>

*Regular Expressions Tutorial - Regex modifiers (flags)*. (n.d.). Sodocumentation.Net. Retrieved August 17, 2022, from <https://sodocumentation.net/regex/topic/5138/regex-modifiers--flags->