Zahava Gopin

Professor Krasso

Web-330: discussion 2.1

1/9/2023

Regular Expression

Regular expressions are a search pattern for text search or text replacement. There are a few ways to use a regular expression. The first regular expression is search(), which will return the position of a match, for example text.search(“hello”); will return what position on the string does hello appear. The second expression is replace() which will change the string where the pattern is placed, for example, text.replace(“hello”, “goodbye”); will change hello to goodbye. Another regular expression is test(). This will return true or false based on the test run, for example, /e/.test(“this is a test”); will return true because there is an e in the above string. The last regular expression is exec(). This will search the text for a pattern and return the text as an object, for example, /e/.exec(“this is a test”); will return what position the letter e is with the text string attached to the results.

In every regular expression there are two parts- the pattern used in the search and the modifier. There are three modifiers i is case-insensitive, g is global matching, and m is multiline matching. Brackets give a range of characters to be used in the expression, for example [abc] means find any of the bracketed characters or [0-9] means to include any number between 0 and 9 in the search pattern. Regular expressions allow for more powerful searching, however, using regular expressions make the code more difficult to read.

Code sample:

<script>

let text = “This is discussion board 2.1”;

let result = text.replace(/discussion board/i, “Forum”);

</script>

Resources:

*JavaScript regular expressions*. JavaScript RegExp Object. (n.d.). Retrieved January 9, 2023, from https://www.w3schools.com/js/js\_regexp.asp