JavaScript Regular Expressions - Basic Functions | JavaScript Tutorial In Hindi #46

When we first see the Regular Expressions, they may seem like a random string. While they might look awkward because they have confusing syntax, they are also extremely useful. This tutorial acts as an introduction to using regular expressions in JavaScript. In this tutorial, we will cover some useful ways to use them. As we know, regular expressions are notably hard to read as they gain in complexity, but it is necessary for the developer to have some knowledge of regular expressions to know what is being tested.

What are Regular Expressions ?

Regular expressions are the patterns that are used to match character combinations in strings. Regular expressions are a powerful way of doing search and replace in strings. It is a small language which is a part of many programming languages like JavaScript, Python, and Java. Regular expression allows us to check a string of characters like a password for patterns, to see if the set password matches with the pattern defined by that regular expression. In this tutorial, we will create a regular expression by using forward slashes (/) to enclose the pattern.

Syntax:-

/pattern/modifiers;

Example:-

```
let str = /Code with Harry/i;
```

Here **/**Code with Harry**/i** is a regular expression. **"**Code with Harry" is a pattern and "**i**" is a modifier that modifies the search to be case-insensitive. If we write **/**Code with Harry**/g, here "g"** performs a global match that will find all matches rather than stopping after the first match.

Regular Expressions Methods:-

Regular expressions are used with the RegExp methods like test() and exec() and with the string methods match() , replace() , search() , and split() .

These methods are explained in detail below with examples.

exec():-

This method will executes a search for a match in a string. It returns an array of information or null on a mismatch. Here is an example:

```
let obj = /h/.exec("Code with harry");
```

test():-

This method tests for a match in a string. It returns true or false. Here is an example:

```
let str = /Code/;
str.test("Code with harry!");
```

match():-

This method will return an array containing all of the matches, including capturing groups, or null if no match is found. Here is an example:

```
let str = "JavaScipt tutorial from begineer to advance level";
let result = str.match(/ial/);
```

search():-

This method will tests for a match in a string. It returns the index of the match, or -1 if the search fails. Here is an example:

```
let str = 'Code with harry';
let reg = /od/;
// search if the pattern is in string variable
let result = str.search(reg);
console.log(result);
```

replace():-

This method will executes a search for a match in a string, and replaces the matched substring with a replacement substring. Here is an example:

```
let str = "Code with abc!";
let result = str.replace("abc", "Harry");
```

split():-

This method uses a regular expression or a fixed string to break a string into an array of substrings. Here is an example:

```
// splitting strings into array elements
const test = /[\s,]+/;
let res = 'Hello world! '.split(re);
console.log(result); // ["I", "am", "learning", "JavaScript", "RegEx"]
```

Code website.html as described/written in the video

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <title>Document</title>
</head>

<body>
    <div class="container">
        <h1 id="heading" class='yourhead rhia is'> Welcome to Code With Harry</h1>
        <div id="myfirst" class="child red good" id="first">child 1

            <ul class="this" id='myul'>
                <li class="childul" id='fui'>this</li>
                <li class="childul">is</li>
                <li class="childul">a</li>
                <li class="childul">list </li>
                <li class="childul" id='lui'>of my dreams</li>
        </div>
        <div class="child">child 2</div>
        <div class="child red">child 3</div>
        <div class="child">child 4</div>
        <form action="none.html" method="post">
            <a href="//codewithharry.com">Go to Code With Harry</a>
```

```
        <br>
        <br>
        Search this website: <input type="text" name="Hello" id="">
        <button id="btn">Submit form</button>
        <!-- <input type="button" id='btn' value="submit"> -->
    </form>
</div>
<!-- <br>
<div class="no">this is a dummy div1</div>
<div class="no">this is a dummy div2</div>
<div class="no">this is a dummy div3</div> -->
<div class="container">
    <h1>Student list</h1>
    <ul id="students"></ul>

    <button id="myBtn" class="btn btn-primary">Your Button</button>
    <button class="btn btn-primary">Fetch Data</button>
    <div id="content"></div>
</div>

<div class="container my-3">
    <h1>Pull the results by clicking the button below:</h1>

    <button class="btn btn-primary" id="meanings">Get meanings</button>
    <div class="my-2">
        <ul id="defs"></ul>
    </div>
</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
    integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
    crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
    integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
    integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
    crossorigin="anonymous"></script>
```

```
</body>
<!-- <script src="js/tut12.js"></script> -->
<!-- <script src="js/tut14.js"></script> -->
<!-- <script src="js/tut15.js"></script> -->
<!-- <script src="js/tut16.js"></script> -->
<!-- <script src="js/tut17.js"></script> -->
<!-- <script src="js/tut18.js"></script> -->
<!-- <script src="js/tut20.js"></script> -->
<!-- <script src="js/tut21.js"></script> -->
<!-- <script src="js/tut23.js"></script> -->
<!-- <script src="js/tut24.js"></script> -->
<!-- <script src="js/tut25.js"></script> -->
<!-- <script src="js/tut27.js"></script> -->
<!-- <script src="js/tut28.js"></script> -->
<!-- <script src="js/tut30.js"></script> -->
<!-- <script src="js/tut31.js"></script> -->
<!-- <script src="js/tut32.js"></script> -->
<!-- <script src="js/tut34.js"></script> -->
<!-- <script src="js/tut37.js"></script> -->
<!-- <script src="js/tut39.js"></script> -->
<!-- <script src="js/tut39b.js"></script> -->
<!-- <script src="js/tut41.js"></script> -->
<!-- <script src="js/tut43.js"></script> -->
<!-- <script src="js/tut44.js"></script> -->
<!-- <script src="js/tut 45.js"></script> -->
<script src="js/tut46.js"></script>

</html>
```

Code tut46.js as described/written in the video

Copy

```
console.log("This is tutorial 46");
let reg = /harry/; // This is a regular expression literal in js
reg = /harry/g; // g means global
// reg = /harry/i; // i means case insensitive

// console.log(reg);
```

```
// console.log(reg.source);

let s = "This is great code with harry and also harry bhai";
// Functions to match expressions
// 1. exec() - This function will return an array for match or null for no match
let result = reg.exec(s);
// result = reg.exec(s);
// console.log(result);
// result = reg.exec(s);
// console.log(result); ---> We can use multiple exec with global flag

// if (result) {
//     console.log(result);
//     console.log(result.input);
//     console.log(result.index);
// }

// 2. test() - Returns true or false
let result2 = reg.test(s);
// console.log(result2); --> This will only print true if the "reg" is there in the string "s"

// 3. match() - It will return an array of results or null
// let result3 = reg.match(s) ---> This is wrong!!
let result3 = s.match(reg) // ---> This is right
// console.log(result3);

// 4. search() - Returns index of first match else -1
// let result4 = reg.search(s) ---> This is wrong!!
let result4 = s.search(reg) // ---> This is right
// console.log(result4);

// 5. replace() - Returns new replaced string with all the replacements (if no flag is given, first match will be replaced)

let result5 = s.replace(reg, 'SHUBHAM');
console.log(result5);
```

Previous

Next

CodeWithHarry