Regular Expressions - Character sets | JavaScript Tutorial In Hindi #48

A **regular expression** is a sequence of characters that define a *search pattern*. Usually, such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings or input validation. In the previous tutorial, we have studied the methods and metacharacters of the regular expression. If you have not watched the tutorial yet, then check ***tutorial#46*** and ***tutorial#47***. In today's tutorial, we will learn how to use regular expressions to work with sets of characters. Unlike the dot(.), which matches any single character, character sets enable us to match specific characters and character ranges.

Several characters or character classes inside square brackets […] mean "search for any character among given." Brackets and sets are used to find a range of characters.

Sets:-

For instance, [aeio] means any of the 4 characters: 'a', 'e' , 'i', or 'o'. That is called a *set*. Sets can be used in a regular expressions along with regular characters:

Ranges:-

Square brackets may also contain *character ranges*. For instance, [a-z] is a character in range from a to z, and [0-9] is a digit from 0 to 9.

Excluding ranges:-

Besides normal ranges, there are "excluding" ranges that look like [^…]. They are denoted by a caret character ^ at the start and match any character *except the given ones*.

For Example:

- [^aeiou] – any character except 'a', 'e', 'i' 'o' or'u'.
- [^0-5] – any character except a digit, the same as \D.
- [^\s] – any non-space character, same as \S.

The following are the some common character set example.

[abcdef]:-

This expression will find any character between the brackets. Here is an example:
Do a global search for the characters "D" and "o" in a string:

```
let reg = /[Do]/gi;
let str = "Do I Know You?";
```

[^abcdef]:-
This expression will find any character NOT between the brackets. Here is an example:
Do a global search for characters that are NOT inside the brackets [abc]:

```
let str = "Is this all there is?";
let reg = /[^abc]/g;
```

[0-9]:-
This expression will find any character between the brackets (any digit). Here is an example:
Do a global search for the numbers 1, 2, 3,4 and 5 in a string:

```
let str = "123456789";
let reg = /[1-5]/g;
```

[^0-9]:-
This expression will find any character NOT between the brackets (any non-digit).Here is an example:
Do a global search for the numbers that are NOT 1 to 5 in a string.

```
let str = "123456789";
let reg = /[^1-5]/g;
```

(a|b):-
This expression will find any of the alternatives specified. Here is an example:
Do a global search to find any of the specified alternatives (0|3|2):

```
let str = "01234567890123456789";
let reg = /(0|3|2)/g;
```

Quantifier:-

Quantifier matches a number of instances of a character, group, or character class in a string. It is appended to a character and specifies how many characters we need. Following is the table of quantifiers, along with its description.

Quantifier	Description
*	It matches zero or more times.
+	It matches one or more times.
?	It matches zero or one time.
{ x }	It matches exactly x times.
{x ,}	It matches at least x times.
{ x, y }	It matches from x to y times.

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> -->
<!-- <script src="js/tut47.js"></script> -->
<!-- <script src="js/tut48.js"></script> -->
<script src="js/tut49.js"></script>

</html>
```

Code as described/written in the video

Copy

```
console.log("This is tutorial 48");
// Regular Expressions:
    // Basic functions
    // Metacharacter Symbols

// const regex = /^h/i;

// Character Sets - We use []
let regex = /h[a-z]rry/; // can be any character from a to z
regex = /h[aty]rry/; // can be an a, t or y
regex = /h[^aty]rry/; // cannot be any of a, t or y
regex = /h[^aty]rr[yYu]/; // cannot be any of a, t or y + can be a u or y
regex = /h[a-zA-Z]rr[yu0-9][0-9]/; // we can have as many character sets as we want

// Quantifiers - We use {}
regex = /har{2}y/; // r can occur exactly 2 times
regex = /har{0,2}y/; // r can occur exactly 0 to 2 (0 or 1 or 2) times

// Groupings  - We use paranthesis for groupings ()
regex = /(har){2}([0-9]r){3}/

// const str = "hirry9 bhai";
str = "harrry bhai"
str = "harhar1r4r5r bhai";


let result = regex.exec(str);
console.log("The result from exec is ", result);


if(regex.test(str)){
    console.log(`The string ${str} matches the expression ${regex.source}`);
}
else{
    console.log(`The string ${str} does not match the expression ${regex.source}`);
}
```

Previous

Next

CodeWithHarry

Copyright © 2022 CodeWithHarry.com