**Regular Expressions:**

A Regular Expression is a Sequence of characters that forms a search pattern.

(chars could be digits,characters)

It can be any number of characters, be it alphabet, digits or special characters,

These Expressions are more commonly used for text search and text replacement.

**Flags, patterns, Quantifiers:**

There are several flags secify to alter the behavior of a regular expression.

Flags may be appended to the end of the regular expression literal or they may be specified as the second argument to the Regular Expression.

General Syntax: started and ended with / /

^ - used to start pattern

$ - used to end the pattern.

Example:

/a/ - check the give char or string contain ‘a’.

This is an Example for JavaScript regular Expressions. -

**Flag Details**

G Global. Find all matches instead of stopping after the first.

I Ignore case/[a-z]/is equivalent to /[a-z][A-Z]/

M Multiline. ^ and $ match the beginning and end of each line respectively, treating \n and \r as delimeters.

U Unicode. If this flag is not supported you must match specific Unicode characters with \uxxxx where xxxx is the character value in Hexadecimal.

Y Find all the consecutive/adjacent matches.

**Patterns:**

Brackets are used to find the range of Characters.

**Pattern:**

[A-Z] Find all the characters from a to z. (lowercase only)

[0-9] find any digits between [0-9].

[a-z|0-9] find any character of digits separated by “|”(alphabets or digits)

**Quantifiers:**

N+ +indicates one or more occurrences of the character n.

N\* \* indicates zero or more occurences of the character n.

N? indicates zero or more Occurences of the character n.

Demo:

Ex: welcome to Javascript. Welcome back!

1. /a/ - Find and select all the a’s in the above sentence.
2. /a\*/ - zero or more occurances of a is selected.

Ex: aaaabbb (It selects aaaa.)

1. /a\*b\*/ - It selects a either followed by b or not followed by b.

**Some other Patterns:**

\w - Matches any word character (alphanumeric, underscore)

\w+ - Matches two or more word character(alphanumeric, underscore)

\s - Matches any whitespace character. (spaces, tabs,linebreaks)

/ \S / - Matches any character that is not a whitespace character.

/\d/ - Matches any digits.

/\d{3}/ - Match 3 of the preceding tokens.

/\w{3}/ - All the characters that are grouped in 3.

/\w{3,4}/ - All the characters that are grouped either in 3 or 4.

**Demo Code: To Retrieve a numbers from a given string (used match() function)**

<!DocType html>

<html>

    <head>

    </head>

    <body>

        <input type ="text" id="txtBox"/>

        <input type="button" value="Process String" onclick="processstring()" style="width=250px"/>

        <br/><br/>

        <textarea id="txtarea" rows="4" cols="30"></textarea>

        <script type="text/javascript">

            function processstring() {

                document.getElementById("txtarea").value="";

                var inputstr = document.getElementById("txtBox").value;

                var result = inputstr.match(/\d+/g);

                if(result!=null)

                {

                    for(var i=0;i<result.length;i++)

                    {

                      document.getElementById("txtarea").value+=result[i]+"\r\n";

                    }

                }

            }

            </script>

    </body>

</html>

Email Validation: (used Test() function)

<!DocType html>

<html>

    <head>

    </head>

    <body>

        <script type="text/javascript">

            document.write(/^\w$/.test("A"));

            document.write(/^\w$/.test("a"));

            document.write(/^\w$/.test("9"));

            document.write(/^\w$/.test("\_"));

        //add one or more char by w+  for username.

            document.write(/^\w+$/.test("abcd"));

            document.write(/^\w+$/.test("Abcd"));

            document.write(/^\w+$/.test("abcd123"));

            document.write(/^\w+$/.test("abcd123\_efg"));

            document.write(/^\w+$/.test("abcd-fg")); //false

            document.write(/^\w+([\.-]?w+)\*$/.test("abcd-efg"));

            document.write(/^\w+ ([.-]?w+)[\*@$/.test("abcd@"))](mailto:*@$/.test(%22abcd@%22)));

//add domain name.

            document.write(/^\w+([\.-]?w+)\*@\w+([-]?w+)\*$/.test("abcd@gmail"));

            document.write(/^\w+([\.-]?w+)[\*@\w+([-]?w+)\*$/.test("abcd@any-word"))](mailto:*@\w+(%5b-%5d?w+)*$/.test(%22abcd@any-word%22)));

//Finally include the top level domain name.

            document.write(/^\w+([\.-]?\w+)\*@\w+([-]?\w+)\*\.\w{2,3}([\.]?\w{2,3})\*$/.test("abcd-df@cool-world.co.in"));

            </script>

    </body>

</html>

All the Methods used in the Regular Expressions are:

1. Exec
2. Match
3. Test
4. Replace

**Samplecode:**

<!DocType html>

<html>

    <body>

        <script type="text/javascript">

            function regexmethod()

            {

                var regex = /welcome/;

            var str = "Hello to js! welcome back!";

            var output = regex.exec(str);

            alert(output);

            var output1 = str.match(regex); //;

            alert(output1);

            var output2 = str.replace(regex,"Hi"); /hello/g

            alert(output2);

            var output3 = regex.test(str);

            alert("Hi This is"+output3);

            //document.write(output3);

            }

        </script>

        <input type="button" value="Click me" onclick="regexmethod()" />

    </body>

    </html>