Alternation

Alternation allows you to specify alternative version of a pattern.

Alternation in a regular expression works just like the OR operator in an if-else conditional statement.

You can specify alternation using a vertical bar (|).

For example, the regexp /fox|dog|cat/ matches the string "fox", or the string "dog", or the string "cat".

Implementation……..

Grouping

Regular expressions use parentheses to group subexpressions, just like mathematical expressions.

Parentheses allow a repetition quantifier to be applied to an entire subexpression.

For example, in regexp /go+/ the quantifier + is applied only to the last character o and it matches the strings "go", "goo", and so on.

Whereas, in regexp /(go)+/ the quantifier + is applied to the group of characters g and o and it matches the strings "go", "gogo", and so on.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript Grouping Sub-expressions in a Regular Expression</title>
</head>
<body>
   <script>
   var str = "One day Gogo will go to school.";
   var regex1 = /(go)+/i;
   var regex2 = /(go)+/gi;

   var matches = str.match(regex1); // case-insensitive match
   console.log(matches);
   // expected output: ["Gogo", "go", index: 8, ...]

   matches = str.match(regex2); // global, case-insensitive match
   console.log(matches);
   // expected output: ["Gogo", "go"]
   </script>
                <p><strong>Note:</strong> Please check out the browser console by pressing the f12 key on the keyboard.</p>
</body>
</html>
```

Word Boundaries

A word boundary character (\b) helps you search for the words that begins and/or ends with a pattern.

For example, the regexp /\bcar/ matches the words beginning with the pattern car, and would match cart, carrot, or cartoon, but would not match oscar.

Similarly, the regexp /car\b/ matches the words ending with the pattern car, and would match oscar or supercar, but would not match cart.

Likewise, the /\bcar\b/ matches the words beginning and ending with the pattern car, and would match only the word car.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript Match Words Starts or Ends with a Pattern Using Regular Expression</title>
</head>
<body>
   <script>
   var regex = /(\bcar\w*)/g;
   var str = "Words begining with car: cart, carrot, cartoon. Words ending with car: oscar, supercar.";
   var replacement = '<b>$1</b>';

   // Highlights the words beginning with car in bold
   var result = str.replace(regex, replacement);
   document.write(result);
   </script>
</body>
</html>
```

JavaScript Form Validation

Client-Side Validation

Web forms have become an essential part of web applications. It is often used to collect user's information such as name, email address, location, age, and so on. But it is quite possible that some user might not enter the data what you've expected. So to save bandwidth and avoid unnecessary strain on your server resources you can validate the form data on client-side (i.e. user's system) using JavaScript before passing it onto the web server for further processing.

Client-side validation is also helpful in creating better user experience, since it is faster because validation occurs within the user's web browser, whereas server-side validation occurs on the server, which require user's input to be first submitted and sent to the server before validation occurs, also user has to wait for server response to know what exactly went wrong.

Note:

Client-side validation is not a substitute or alternative for server-side validation.

You should always validate form data on the server-side even if they are already validated on the client-side, because user can disable JavaScript in their browser.

Form Validation with JavaScript

The form validation process typically consists of two parts—

1. the required fields validation which is performed to make sure that all the mandatory fields are filled in, and

2. the data format validation which is performed to ensure that the type and format of the data entered in the form is valid.

Creating the HTML Form

Building the Form Validation Script

Adding Style Sheet to Beautify the Form