Dialog Boxes in JavaScript

JavaScript provides the ability to pick up user input or display small amounts of text to the user by using dialog boxes. These dialog boxes appear as separate windows and their content depends on the information provided by the user. JavaScript provides three types of pop-up dialog boxes for giving feedback and retrieving user-entered data using methods of window object for use in your applications.

- An alert box
- A confirm box
- A user-input prompt box

Alert dialog box

It is used to display message or information to user. For example, you want to tell the user to enter his name in a text field before submitting a form. It is most commonly used window method. Following figure shows an example of alert dialog. It has one OK button. User can acknowledge the message by pressing OK button.

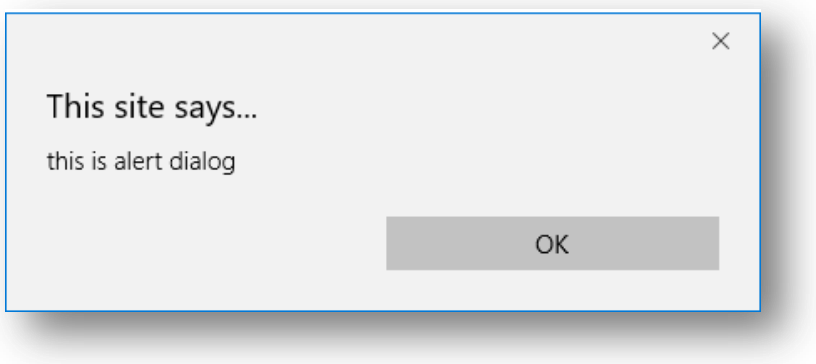


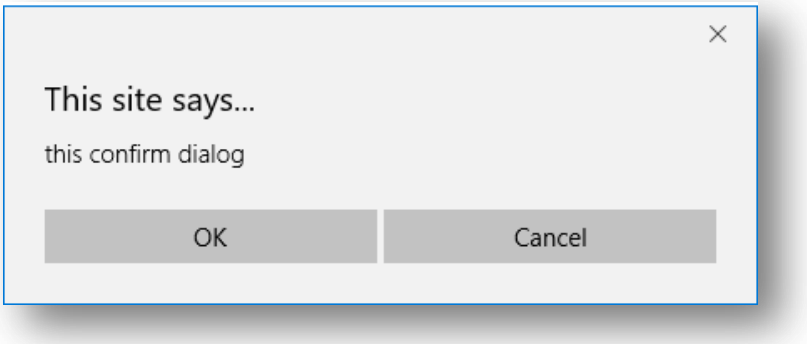
Figure: Alert Dialog Box

Alerts look different from browser to browser and operating system to operating system. On Microsoft Edge browser on Windows 10 operating system, the alert looks like what you see in above Figure.

Alerts do not return any information to the script. It simply gives a message to the user and stops any further script execution till the OK button is pressed.

Confirm Dialog box

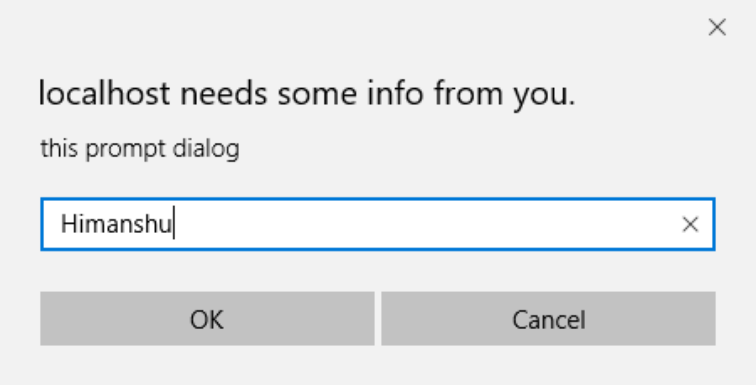
It is used to confirm action by user. For example, you want to confirm the user before deleting record. Following figure shows an example of confirm dialog. It has two buttons OK and Cancel. User can confirm the message by pressing OK button or deny the action by pressing cancel button. confirm() is a method that returns a Boolean value (true or false) depending on the user press the OK or Cancel button respectively



Confirm dialog is an easy way to stop user from taking unwanted steps in web applications.

Prompt Dialog Box

Alert () and confirm () provides information to the user, but to receive the user input we can use the prompt () method. It takes two parameters, the first parameter is a string displayed above the entry field as a message and the second parameter is a default value for the entry field. It has two buttons labeled OK and Cancel as shown in following figure.



var user=prompt(this prompt dialog','Himanshu');

When the user presses the OK button, the value of the variable user will be either Himanshu or whatever is entered. When user presses the Cancel button, the value will be null.

Example

Consider following web page.

We want to give following functionality:

1. When user clicks 'Show Alert' button it should display alert box with some message.
2. When user double clicks 'Show Confirm' button it should display confirm dialog
3. When user moves his mouse over 'Show Prompt' button it should take some input and input should be displayed as alert dialog.

The code for the above functionality is shown below

```
<html>
    <head>
        <script>
        function show1()  {
            alert('this is alert dialog');
        }
        function show2() {
            confirm('this confirm dialog');
        }
        function show3() {
            var name = prompt('this prompt dialog');
            alert('You have entered '+name);
        }
        </script>
    </head>
    <body>
        <h1>Dialog Box Demo</h1>
        <button onClick="show1()"> Show Alert </button>
        <button onDblClick="show2()"> Show Confirm </button>
        <button onMouseOver="show3()"> Show Prompt</button>
    </body>
</html>
```

Window Open Function

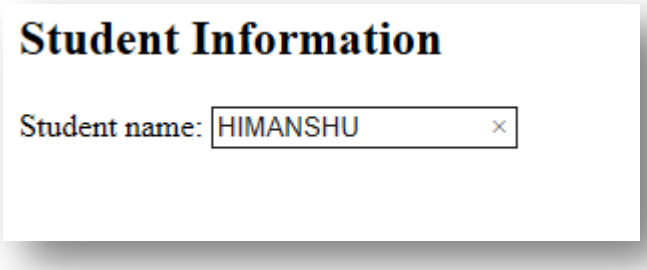
The window object open method is used to open any web page with given URL. Consider following web page where we want to open e PG Pathshala home page when user clicks button with caption 'Open e PG Pathshala'.

The code for the above functionality is shown below

```
<html>
    <head>
        <script>
            function openepgp() {
               window.open("http://epgp.inflibnet.ac.in/");
            }
        </script>
    </head>
    <body>
        <H1> Type URL to open Website... </H1>
        <form>
           <input type="button" value="Open e PG Pathshala" onclick="openepgp()">
        </form>
    </body>
</html>
```

Handling Keyboard Event

Consider following web page where we want to write down JavaScript code which converts username entered into uppercase as user type it.



The code for above functionality is given below.

```
<html>
    <head>
        <script>
            function toUpper() {
               var sname = document.getElementById("sname");
               sname.value = sname.value.toUpperCase();
            }
        </script>
        </head>
    <body>
        <H2>Student Information</H2>
        Student name: <input type="text" id="sname" onkeyup="toUpper()">
    </body>
</html>
```

Handling Mouse Event

Now let us learn how to handle mouse event. Consider following web page where we want to change image whenever user moves mouse over or out of an image.

Initially web page display following

Whenever user moves mouse over image the image changes as shown below.

The code to achieve above mentioned functionality is given below.

```
<html>
    <head>
        <title>Test Document</title>
        <script type="text/javascript">
            function MouseOverimg() {
                var img=document.getElementById("img");
                img.src = "1.jpg";
            }
            function MouseOutimg() {
                var img=document.getElementById("img");
                img.src = "2.jpg";
            }
        </script>
    </head>
    <body>
        <h1>Move Mouse Over Image to Change Image </h1>
        <img src="2.jpg" id="img" name="img" height="200px" width="200px"
        onMouseOver="MouseOverimg();" onMouseOut="MouseOutimg();" />
    </body>
</html>
```

Data Validation

One of the important uses of JavaScript is for data validation. Data validation is the process of checking that user input is proper information. Followings are the examples of validations:

- All required fields are entered by user or not
- Is there any data type mismatch
- Check whether numeric data in range or not
- Comparing two values
- Checking data for certain combination of characters

Regular Expression

Regular expressions help you match a string against a character pattern and we can use regular expression for validating user data. A regular expression is an object that describes a pattern of characters. Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text. The syntax for declaring regular expression is shown below:

Syntax: */pattern/modifiers*;

We can use modifier i to perform case-insensitive matching, g to perform a global match and m to perform multiline matching

Regular Expressions Brackets

Regular expression use square brackets to limit the scope of the choices only to those you want to offer. Following are example of square bracket with its meaning

Expression	Description
[abc]	Find any character between the brackets
[^abc]	Find any character NOT between the brackets
[0-9]	Find any character between the brackets
[^0-9]	Find any character NOT between the brackets
(x|y)	Find any of the alternatives specified

Regular expression Meta characters

There are some shortcut notations available for represent regular expression which is known as Meta characters. Following table list regular expression meta characters with meaning they represents.

Char	Description
.	Find a single character, except newline or line terminator
\w	Find a word character
\W	Find a non-word character
\d	Find a digit
\D	Find a non-digit character
\s	Find a whitespace character

Char	Description
\S	Find a non-whitespace character
\b	Find a match at the beginning/end of a word
\B	Find a match not at the beginning/end of a word
\0	Find a NUL character
\n	Find a new line character
\f	Find a form feed character
\r	Find a carriage return character
\t	Find a tab character
\v	Find a vertical tab character
\xxx	Find the character specified by an octal number xxx
\xdd	Find the character specified by a hexadecimal number dd
\uxxxx	Find the Unicode character specified by a hexadecimal number xxxx

Regular Expressions Quantifiers

Sometime you want to allow range of characters, like a to z, but you want to restrict their number. For this you can use quantifiers in regular expressions. The quantifier and its meaning is given in Table.

Quantifier	Description
n+	Matches any string that contains at least one *n*
n*	Matches any string that contains zero or more occurrences of *n*
n?	Matches any string that contains zero or one occurrences of *n*
n{X}	Matches any string that contains a sequence of *X n*'s
n{X,Y}	Matches any string that contains a sequence of X to Y *n*'s
n{X,}	Matches any string that contains a sequence of at least X *n*'s
n$	Matches any string with *n* at the end of it
^n	Matches any string with *n* at the beginning of it
?=n	Matches any string that is followed by a specific string *n*

Methods Using Regular Expressions

There are several methods that take regular expressions as parameters. Following are regular expression methods

Method	Description
pattern.test(string)	Tests whether the string matches the pattern and returns true or false.
pattern.exec(string)	Matches the string and the pattern one time and returns an array of matches or null.
string.match(pattern)	Matches the string and the pattern and returns the resulting matches as an array of strings or null.
string.search(pattern)	Matches the string and the pattern and returns the positions of the positive matches. If the string does not match any of the pattern, the search returns -1.

string.replace(pattern, replaceString)	Matches the string against the pattern and replaces every positive match with replaceString
string.split(pattern, limit)	Matches the string against the pattern and splits it into an array with the substrings surrounding the pattern matches as array items. The optional limit parameter cuts down the number of array elements

Example: consider above syntax for regular expression. The regular expression to check email and mobile number is given below.

- Regular Expression for Email: /^[a-zA-Z0-9_]+@[a-zA-Z]+\.[a-zA-Z]{2,4}$/
- Regular Expression for Mobile Number: /^[0-9]{10,11}$/

Try to analyze and understand meaning of above regular expression yourself.

Validation Example

Consider webpage to capture details for user registration with following interface.

Registration Page	
User Fist Name	
User Middle Name	
User Last Name	
User Name	
Password	
Confirm Password	
Email Address	
Phone No	
Submit Query	

We want to perform following validation in user input.

- All field are compulsory
- Password and confirm password should match
- Email address should be valid in format
- Mobile number should be only digit and 10-11 digits

The code for above mention requirement is given below.

```
<html>
	<head>
		<title>Validation Demo</title>
	<script>
		function validate()  {
			var ufn=document.forms["myform"]["ufn"].value;
			if(ufn=="" || ufn==null)  {
				alert('Please enter first Name');
				return false;
			}
			var umn=document.forms["myform"]["umn"].value;
			if(umn=="" || umn==null) {
				alert('Please enter middle Name');
				return false;
			}
			var uln=document.forms["myform"]["uln"].value;
			if(uln=="" || uln==null) {
				alert('Please enter last Name');
				return false;
			}
			var un=document.forms["myform"]["un"].value;
			if(un=="" || un==null) {
				alert('Please enter user name');
				return false;
			}
			var pass=document.forms["myform"]["pass"].value;
			if(pass=="" || pass==null) {
				alert('Please fill out the Password');
				return false;
			}
			var cpass=document.forms["myform"]["cpass"].value;
			if(cpass=="" || cpass==null) {
				alert('Please fill out the Confirm Password');
				return false;
			}
			if(pass==cpass) {
			}
			else {
				alert('Password Does not Match');
				return false;
			}
			var email=document.forms["myform"]["email"].value;
			if(email=="" || email==null) {
				alert('Please fill out the Email Address');
				return false;
			}
			var filter=/^[a-zA-Z0-9_]+@[a-zA-Z]+\.[a-zA-Z]{2,4}$/;
			if(!filter.test(email)) {
				alert('Please,provide proper email address');
				return false;
```

```
                        }
                        var mo=document.forms["myform"]["pno"].value;
                        if(mo=="" || mo==null) {
                                alert('Please fill out the Mobile Number');
                                return false;
                        }
                        var chk=/^[0-9]{10,11}$/;
                        if(!chk.test(mo)) {
                                alert('please enter valid mobile number');
                                return false;
                        }
                }
        </script>
</head>
<body>
        <form name="myform" action="" method="" onsubmit="return validate()">
        <table border="2" align="center">
                <tr>
                        <th colspan=2> Registration Page </th>
                </tr>
                <tr>
                        <th>User Fist Name </th>
                        <td><input type="text" name="ufn" ></td>
                </tr>
                <tr>
                        <th>User Middle Name </th>
                        <td><input type="text" name="umn" ></td>
                </tr>
                <tr>
                        <th>User Last Name </th>
                        <td><input type="text" name="uln" ></td>
                </tr>
                <tr>
                        <th>User Name </th>
                        <td><input type="text" name="un" ></td>
                </tr>
                <tr>
                        <th>Password </th>
                        <td><input type="password" name="pass" ></td>
                </tr>
                <tr>
                        <th>Confirm Password </th>
                        <td><input type="password" name="cpass" ></td>
                </tr>
                <tr>
                        <th>Email Address </th>
                        <td><input type="text" name="email" ></td>
                </tr>
                <tr>
                        <th>Phone No </th>
```

```
                <td><input type="text" name="pno" ></td>
            </tr>
            <tr>
                <th colspan=2><input type="submit" name="submit"/></th>
            </tr>
        </table>
        </form>
</body>
</html>
```