JavaScript Message Boxes: alert(), confirm(), prompt()

JavaScript provides built-in global functions to display popup message boxes for different purposes.

- alert(message): Display a popup box with the specified message with the OK button.
- confirm(message): Display a popup box with the specified message with OK and Cancel buttons.
- prompt(message, defaultValue): Display a popup box to take the user's input with the OK and Cancel buttons.

In JavaScript, global functions can be accessed using the window object like window.alert(), window.confirm(), window.prompt().

Example: alert()

```
<html>

<body>

<h1>Demo: alert()</h1>

<button onclick="displayInfo()">Display Messages</button>

<script>

function displayInfo(){

alert("This is an alert message box.");  // display string message

alert('This is a numer: ' + 100); // display result of a concatenation

alert(100); // display number

alert(Date()); // display current date

}

</script>

</body> </html>
```

Example: confirm()

```
<!DOCTYPE html>

<html>

<body>

        <h1>Demo: confirm() </h1>
        <button onclick="save()">Save Data</button>
        <p id="msg"></p>

        <script>
                function save(){
                  var userPreference;

                        if (confirm("Do you want to save changes?") == true) {
                                userPreference = "Data saved successfully!";
                        } else {
                                userPreference = "Save Canceled!";
                        }

                        document.getElementById("msg").innerHTML = userPreference;
                }
    </script>
</body>
</html>
```

Example: Take User's Input using prompt()

```
<!DOCTYPE html>
<html>
<body>
        <h1>Demo: prompt()</h1>
        <button onclick="myinput()">Click to enter your name</button>
        <p id="msg"></p>
        <script>

                function myinput(){

                        var name = prompt("Enter Your Name:");


                        if (name == null || name == "") {

                                document.getElementById("msg").innerHTML = "You did not entert anything. Please enter your name again";

                        }

                        else

                        {

                                document.getElementById("msg").innerHTML =

                        }

                }

    </script>

</body>

</html>
```

JavaScript Date

JavaScript provides Date object to work with date & time, including days, months, years, hours, minutes, seconds, and milliseconds.

Use the Date() function to get the string representation of the current date and time in JavaScript. Use the new keyword in JavaScript to get the Date object.

Example: Date In JavaScript

```
<html>

<body>
```

```
        <h1>Demo: Current Date</h1>

        <p id="p1"></p>

        <p id="p2"></p>

        <script>

                document.getElementById("p1").innerHTML = Date();

                var currentDate = new Date();

                document.getElementById("p2").innerHTML = currentDate;

    </script>

</body>

</html>
```

Date Methods Reference

Sr.No.	Method & Description
1	Date() Returns today's date and time
2	getDate() Returns the day of the month for the specified date according to local time.
3	getDay() Returns the day of the week for the specified date according to local time.
4	getFullYear() Returns the year of the specified date according to local time.
5	getHours() Returns the hour in the specified date according to local time.
6	getMilliseconds() Returns the milliseconds in the specified date according to local time.

7	getMinutes() Returns the minutes in the specified date according to local time.
8	getMonth() Returns the month in the specified date according to local time.
9	getSeconds() Returns the seconds in the specified date according to local time.
10	getTime() Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC.
11	setDate() Sets the day of the month for a specified date according to local time.
12	setFullYear() Sets the full year for a specified date according to local time.
13	setHours() Sets the hours for a specified date according to local time.

14	setMilliseconds() Sets the milliseconds for a specified date according to local time.
15	setMinutes() Sets the minutes for a specified date according to local time.
16	setMonth() Sets the month for a specified date according to local time.
17	setSeconds() Sets the seconds for a specified date according to local time.
18	setTime() Sets the Date object to the time represented by a number of milliseconds sinceJanuary 1, 1970, 00:00:00 UTC.

Example

```
<html>
<body>
<script type="text/javascript"
    language="javascript">var d1 = new Date();
    document.write(d1+"<br>");
    document.write(d1.getDay()+"<br>");
    document.write(d1.getMonth()+"<br>");
    document.write(d1.getDate()+"<br>");
    document.write(d1.getFullYear()+"<br>");
    document.write(d1.getHours()+"<br>");
    document.write(d1.getMinutes()+"<br>");
    document.write(d1.getSeconds()+"<br>");
</script>
</body>
</html>
```

JavaScript HTML DOM Document

Changing HTML Content

The easiest way to modify the content of an HTML element is by using the innerHTML property.

To change the content of an HTML element, use this syntax:

document.getElementById(*id*).innerHTML = *new HTML*

This example changes the content of a <p> element:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript can Change HTML</h2>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>
<p>The paragraph above was changed by a script.</p>
</body>
</html>
```

The document object represents your web page.

If you want to access any element in an HTML page, you always start with accessing the document object.

Finding HTML Elements

Method	**Description**
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

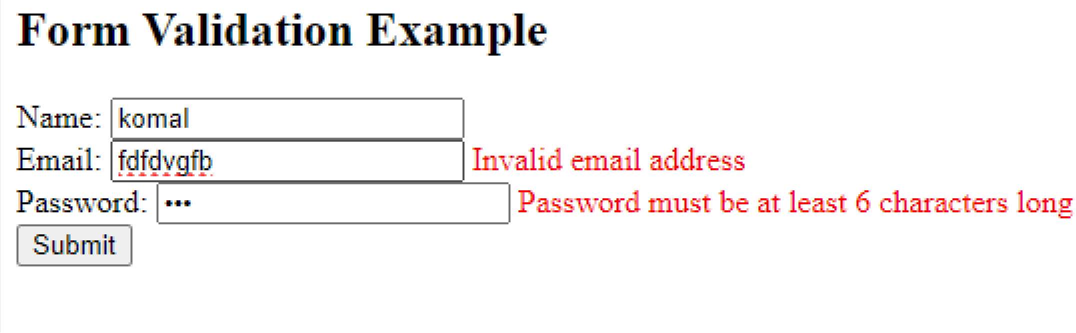
Changing HTML Elements

Property	**Description**
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = *new value*	Change the attribute value of an HTML element
element.style.*property* = *new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

Form Validation Using JavaScript

```html
<!DOCTYPE html>
<html>
<head>
    <title>Form Validation Example</title>
    <style>
        .error {
            color: red;
        }
    </style>
</head>
<body>
    <h2>Form Validation Example</h2>
    <form id="myForm" onsubmit="return validateForm()" method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name">
        <span id="nameError" class="error"></span><br>

        <label for="email">Email:</label>
        <input type="text" id="email" name="email">
        <span id="emailError" class="error"></span><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password">
        <span id="passwordError" class="error"></span><br>

        <input type="submit" value="Submit">
    </form>

    <script>
        function validateForm() {
            var name = document.getElementById("name").value;
            var email = document.getElementById("email").value;
            var password = document.getElementById("password").value;

            var nameError = document.getElementById("nameError");
            var emailError = document.getElementById("emailError");
            var passwordError =
document.getElementById("passwordError");

            // Reset error messages
            nameError.innerHTML = "";
            emailError.innerHTML = "";
            passwordError.innerHTML = "";

            var isValid = true;
            if (name === "") {
                nameError.innerHTML = "Name is required";
                isValid = false;
            }

            if (email === "") {
                emailError.innerHTML = "Email is required";
                isValid = false;
            } else if (!isValidEmail(email)) {
                emailError.innerHTML = "Invalid email address";
                isValid = false;
            }

            if (password === "") {
                passwordError.innerHTML = "Password is required";
                isValid = false;
            } else if (password.length < 6) {
                passwordError.innerHTML = "Password must be at least 6 characters long";
                isValid = false;
            }
```

```
            return isValid;
        }

        function isValidEmail(email) {
            var emailRegex = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
            return emailRegex.test(email);
        }
    </script>
</body>
</html>
```



Form Validation Example

Name: komal

Email: fdfdvgfb Invalid email address

Password: ••• Password must be at least 6 characters long