# 

|  |  |
| --- | --- |
| **Ex. No: 3** | LA-3:- Form Making and Validation using JavaScript |
| **31.07.2023** |

**Aim:**

To create a form using usual form elements in JavaScript including the Alert(), Confirm(),

and Response() functions and validate the form elements.

**Algorithm**:

There are 3 ways of Javascript implementation in HTML1. Inline JavaScript: Inline JavaScript is code that is written directly within the HTML

document, usually in the <script> element within the HTML file. It's placed within the

<head> or <body> of the HTML document.

Ex-

<!DOCTYPE html>

<html>

<head>

<script>

function sayHello() {

alert("Hello, World!");

}

</script>

</head>

<body>

<button onclick="sayHello()">Click me</button>

</body>

</html>

2. **Internal JavaScript**: Internal JavaScript is written within the HTML document but

separated into a <script> element, either in the <head> or <body>. It's often used for

scripts specific to a single page.

Ex-

<!DOCTYPE html>

<html>

<head>

<script>

function sayHello() {

alert("Hello, World!");

}

</script>

</head>

<body>

<button onclick="sayHello()">Click me</button>

</body>

</html>

3. **External JavaScript (Linked)**: External JavaScript involves creating a separate .js

file with JavaScript code and linking it to one or more HTML pages. This approach is

used for reusability and better organization.

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Form Validation Example</title>

<style>

.error {

color: red;

}

</style>

</head>

<body>

<h1>Form Validation Example</h1>

<form id="myForm">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<span class="error" id="nameError"></span>

<br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<span class="error" id="emailError"></span>

<br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<span class="error" id="passwordError"></span>

<br>

<button onclick="onpress()">Submit</button>

<p id="demo"></p>

</form>

<script src="script.js"></script>

</body>

</html>

CSS

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

line-height: 1.6;

background-color: #f5f5f5;

padding: 20px;

}

h1 {

text-align: center;

margin-bottom: 20px;

}

form {

max-width: 400px;

margin: 0 auto;

background-color: #ffffff;

padding: 20px;

border: 1px solid #ddd;

border-radius: 5px;

}

label {

display: block;

margin-bottom: 8px;

}

input[type="text"],

input[type="email"],

input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

margin-bottom: 15px;

}

input[type="submit"] {

background-color: #4CAF50;

color: #fff;

padding: 10px 15px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

input[type="submit"]:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 14px;

}

/\* Responsive styles \*/

@media (max-width: 480px) {

form {

max-width: 100%;

}

}

JAVA SCRIPT

function validateForm() {

let x = document.forms["myForm"]["name"].value;

if (x == "") {

alert("Name must be filled out");

return false;

}

let y = document.forms["myForm"]["email"].value;

if (y == "") {

alert("Email must be filled out");

return false;

}

let z = document.forms["myForm"]["password"].value;

if (z == "") {

alert("password must be filled out");

return false;

}

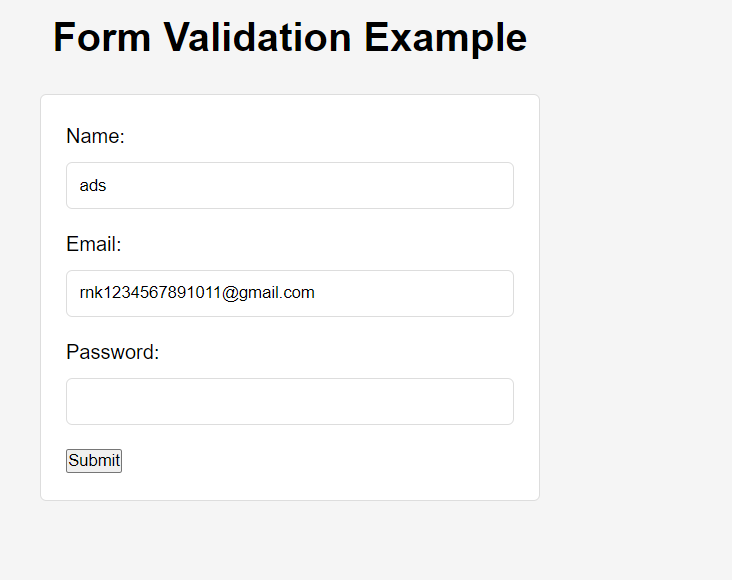
}

let test

function onpress(){

confirm("YOU have submitted")

}



Result: Therefore, we've successfully created a form using HTML and CSS and we’ve validated it using Javascript.