

**Course
&
Test Series**

Javascript Forms and Validations

Document Object Model DOM and Validations

Module 5 - Javascript Forms and Validations

 **CBSE**

Forms in JavaScript

A form in HTML is used to collect user input. Example: login form, registration form, feedback form, etc.

 **ICSE**

A simple HTML form:

 **NTSE**

```

<!DOCTYPE html>
<html>
<head>
<title>Form with Validation</title>
<script>
// Run when the page loads
window.onload = function() {
// Add event listener to form
document.getElementById("myForm").addEventListener("submit",
function(event) {
// Get values
let name = document.getElementById("name").value.trim();
let email = document.getElementById("email").value.trim();

// Email pattern (basic regex)
let emailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;

// Validation checks
if (name === "") {
alert("Name cannot be empty!");
event.preventDefault(); // stop form submission
return;
}

if (!email.match(emailPattern)) {
alert("Please enter a valid email address!");
event.preventDefault();
return;
}

// If everything is fine
alert("Form submitted successfully!");

```

 **Banking & Insurance**

 **Central Govt. Service**

 **State Govt. Services**

 **LAW Entrance**

 **MBA Entrance**

 **Railways & Metro Services**

...many more

abhyasonline.in



Course
&
Test Series

Javascript Forms and Validations

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

```
});
}
</script>
</head>
<body>
<h2>Simple Form with Validation</h2>

<form id="myForm">
<label>Name:</label>
<input type="text" id="name"><br><br>

<label>Email:</label>
<input type="email" id="email"><br><br>

<button type="submit">Submit</button>
</form>
</body>
</html>
```

OUTPUT:

Simple Form with Validation

Name:

Email:

In JavaScript, we can:

- Access form elements
let nameValue = document.getElementById("name").value;
- Handle form submission
document.getElementById("myForm").addEventListener("submit",
function(event){
event.preventDefault(); // stops default form submission
alert("Form submitted!");
});



Course & Test Series

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking & Insurance**

 **Central Govt. Service**

 **State Govt. Services**

 **LAW Entrance**

 **MBA Entrance**

 **Railways & Metro Services**

...many more

abhyasonline.in

Javascript Forms and Validations

Form Validation

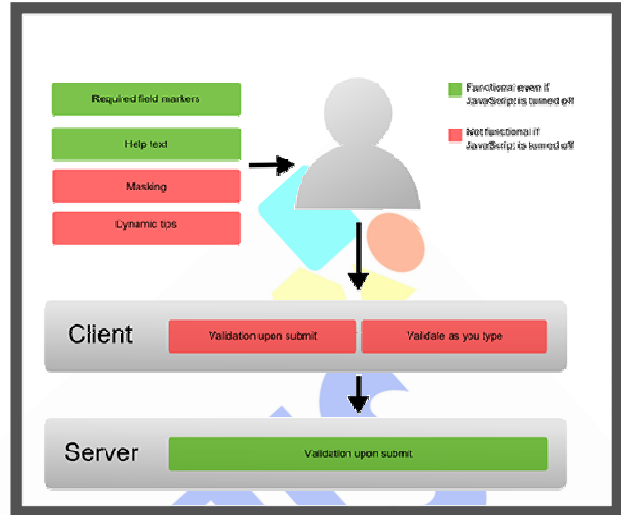
Form validation in JavaScript is the process of checking whether the data entered into a web form by a user is correct, complete, and meets certain criteria before it's submitted to a server.

Types of Form Validation

- Leaving a field empty
- Entering wrong email format
- Entering very short passwords

Why Form Validation is Important?

- Prevents incorrect or incomplete data from being sent to the server.
- Improves user experience by providing immediate feedback.
- Helps reduce server load by catching errors early.
- Enhances security by filtering potentially malicious input.



CASE 1: Required Field Validation

```
<!DOCTYPE html>
<html>
<head>
<title>Username Form Validation</title>
<script>
// Wait until the page loads
window.onload = function() {
document.getElementById("form1").addEventListener("submit",
function(event) {
let user = document.getElementById("username").value.trim();

if (user === "") {
alert("Username cannot be empty!");
event.preventDefault(); // stop form submission
} else {
alert("Form submitted successfully! ✅");
}
}
});
};
</script>
```

Course
&
Test Series

Javascript Forms and Validations

CBSE

```
</head>
<body>
  <h2>Simple Username Form</h2>
```

ICSE

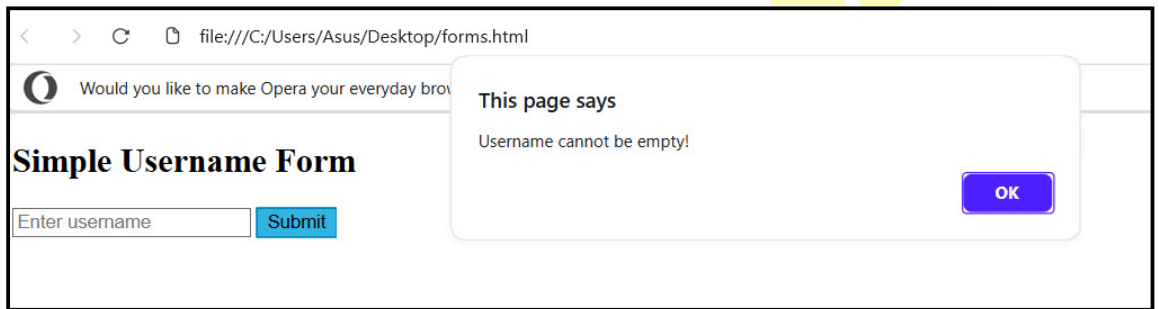
```
<form id="form1">
  <input type="text" id="username" placeholder="Enter username">
  <button type="submit">Submit</button>
</form>
```

NTSE

```
</body>
</html>
```

Banking & Insurance

OUTPUT:



Central Govt. Service

How it works:

- If you leave the input empty and click Submit → Alert: "Username cannot be empty!"
- If you enter a username → Alert: "Form submitted successfully!"

State Govt. Services

CASE 2: Email Validation

LAW Entrance

```
<!DOCTYPE html>
<html>
<head>
  <title>Email Validation Form</title>
  <script>
    // Run the script after the page loads
    window.onload = function() {
      document.getElementById("form2").addEventListener("submit",
function(event) {
  let email = document.getElementById("email").value.trim();
  let pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/; // simple regex for email

  if (!email.match(pattern)) {
    alert("Please enter a valid email address!");
```

MBA Entrance

Railways & Metro Services

...many more

abhyasonline.in

Course
&
Test Series

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Javascript Forms and Validations

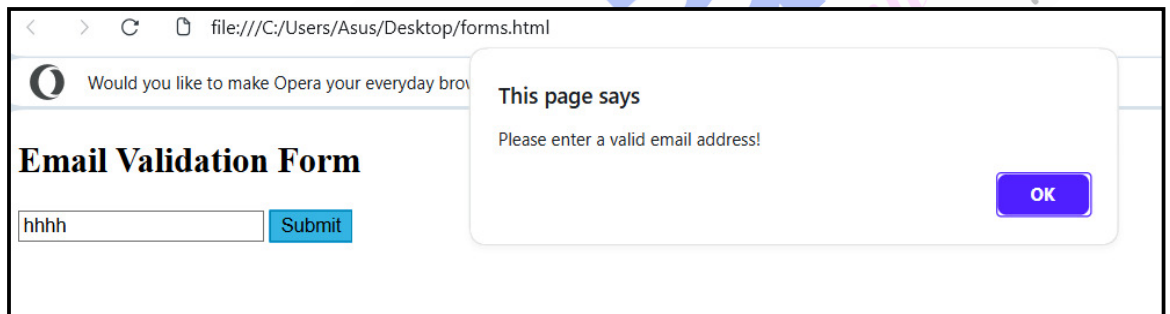
```

event.preventDefault(); // stop form submission
} else {
    alert("Email submitted successfully! ✅");
}
});
};
</script>
</head>
<body>
<h2>Email Validation Form</h2>

<form id="form2">
    <input type="text" id="email" placeholder="Enter email">
    <button type="submit">Submit</button>
</form>

</body>
</html>
    
```

OUTPUT:



How it works:

- If you type wrong format like abc@ → Alert: "Please enter a valid email address!"
- If you type correct email like abc@gmail.com → Alert: "Email submitted successfully! "

CASE 3: Password Validation

```

<!DOCTYPE html>
<html>
<head>
    <title>Password Validation Form</title>
<script>
    
```

**Course
&
Test Series**

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking &
Insurance**

 **Central Govt.
Service**

 **State Govt.
Services**

 **LAW
Entrance**

 **MBA
Entrance**

 **Railways & Metro
Services**

...many more

abhyasonline.in

Javascript Forms and Validations

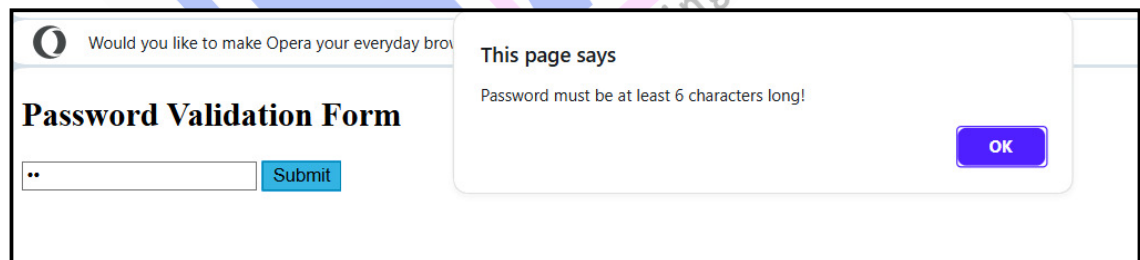
```
// Run script after page loads
window.onload = function() {
    document.getElementById("form3").addEventListener("submit",
function(event) {
    let password = document.getElementById("pass").value.trim();

    // Check password length
    if (password.length < 6) {
        alert("Password must be at least 6 characters long!");
        event.preventDefault(); // Stop form submission
    } else {
        alert("Password accepted! ✅");
    }
    });
};
</script>
</head>
<body>
<h2>Password Validation Form</h2>

<form id="form3">
    <input type="password" id="pass" placeholder="Enter password">
    <button type="submit">Submit</button>
</form>

</body>
</html>
```

OUTPUT:



How it works:

- If the password is less than 6 characters → Alert: "Password must be at least 6 characters long!"
- If the password is 6 or more characters → Alert: "Password accepted! "

**Course
&
Test Series**

Javascript Forms and Validations

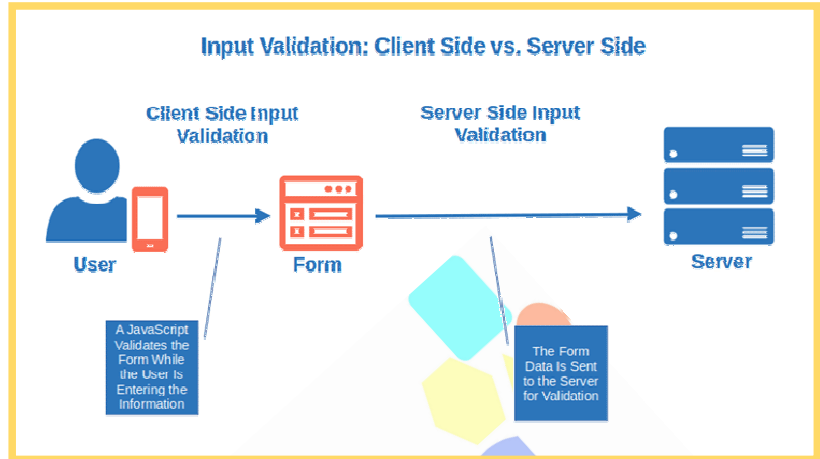
Types of Form Validation:

Client-side Validation (JavaScript):

- Happens in the browser.
- Uses JavaScript to validate fields before sending to the server.

Server-side Validation:

- Happens after form submission on the server.
- More secure and cannot be bypassed.
- Client-side validation is faster and provides instant feedback, but it should always be backed up by server-side validation for security.



Advantages of Form Validation

- Improves user experience with immediate feedback
- Prevents submission of incomplete or incorrect data
- Reduces unnecessary server requests
- Enhances data accuracy and consistency
- Helps protect against malicious input (e.g. XSS, SQL injection)
- Saves server processing time and resources
- Minimizes back-and-forth communication with users
- Supports dynamic error handling and messages

CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

**Course
&
Test Series**

Javascript Forms and Validations

Difference Between required HTML Attribute and JavaScript Validation:

Feature	HTML required Attribute	JavaScript Validation
Where it's defined	In the HTML markup (e.g. <input required>)	In a JavaScript function or script
Execution	Automatically handled by the browser	Manually triggered via JS functions
Customization	Limited (standard checks only)	Highly customizable (complex rules)
Browser Dependency	Depends on browser support	Works across all browsers with JS enabled
Error Message	Default browser messages	Custom messages possible
Flexibility	Simple and quick for basic checks	Ideal for complex, multi-field validation
Can be bypassed?	Can be bypassed by disabling HTML5 validation	Can also be bypassed if JS is disabled
When it triggers	On form submission	You control when it runs (e.g. on input, submit, blur)

Sample Code: Using All Form Validations Together

```

<!DOCTYPE html>
<html>
<head>
  <title>Form Validation Example</title>
</head>
<body>
<form onsubmit="return validateForm()">
  Name: <input type="text" id="name"><br><br>
  <input type="submit" value="Submit">
</form>
<script>
function validateForm() {
  var name = document.getElementById("name").value;
  if (name === "") {
    alert("Name is required!");
    return false;
  }
}

```

CBSE

ICSE

NTSE

Banking & Insurance

Central Govt. Service

State Govt. Services

LAW Entrance

MBA Entrance

Railways & Metro Services

...many more

abhyasonline.in



Course & Test Series
 CBSE
 ICSE
 NTSE
 Banking & Insurance
 Central Govt. Service
 State Govt. Services
 LAW Entrance
 MBA Entrance
 Railways & Metro Services
...many more
abhyasonline.in

Javascript Forms and Validations

```
document.write("Form submitted successfully!<br>Name: " + name);
return false; // prevent actual form submission (for demo)
}
</script>
</body>
</html>
```

Explanation

- Validates that the name field is not empty.
- Uses alert() for errors.
- If valid, it uses document.write() to show a message.
- return false; prevents the form from reloading the page.

Assignment

Create a simple HTML form that asks the user for the following information:

- Full Name
- Email Address
- Password
- Confirm Password

Write JavaScript validation that checks the following before allowing form submission:

1. All fields must be filled (no empty input).
2. The email address must contain @ and . in the correct position.
3. The password must be at least 6 characters long.
4. The "Confirm Password" field must match the "Password" field.

If all validations pass, display a message:
Form submitted successfully.

If any validation fails, show an appropriate alert message and prevent the form from submitting.

User Registration Form

Full Name:

Email Address:

Password:

Confirm Password: