

```
// php validation example 1
```

```
//html
```

```
<!DOCTYPE html>
```

```
<html>
```

&lt;head&gt;

&lt;title&gt;Test JavaScript Form Validation&lt;/title&gt;

```
<link rel="stylesheet" href="validatcss.css">
```

```
<script type="text/javascript" src="validatefrm.js" ></script>
```

&lt;/head&gt;

<body>

## Test JavaScript Form Validation

```
<form method="post" action="validation.php" name="empfrm" id="eform" onsubmit="return validateForm(this)">
```

|
 Name\* |  |

```
<input type="text" id="name" name="name" tabindex="1" placeholder="Enter" value="Default Name"
```

/&gt;

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

|
 Email\* |  |

```
<input type="text" id="email" name="email" />
```

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

|
  |

```

        <td>

        <input type="submit" value="SEND" id="submit"/>&nbsp;  

        <input type="reset" value="CLEAR" id="reset"/>&nbsp;  &nbsp;  

        </td></tr>

    </table>

</form>

</body>

</html>

```

### **//validatefrm.js**

```

window.onload = init;

function init() {

    // Attach "onclick" handler to "reset" button
    document.getElementById("reset").onclick = clearDisplay;

    // Set initial focus
    document.getElementById("name").focus();
}

//document.forms (Collection of all forms in the document)
//document.forms[0] (Refers to the first form in the document)

//1.[index]    Returns the element in <form> with the specified index (starts at 0).
//document.forms[0].elements[0].value;
//document.forms.item(0).elements[0].value;
// document.getElementById("eform").elements[0].value;
//document.forms.namedItem("eform").elements[0].value;

//2.item(index) Returns the element in <form> with the specified index (starts at 0).
//document.getElementById("eform").elements.item(0).value;

//3.namedItem(id)    Returns the element in <form> with the specified id.
//document.getElementById("eform").elements.namedItem("name").value;

```

```

function validateForm(thisForm) {
    alert("hello");
    with(thisForm) {
        //The with statement extends the scope chain for a statement
        return (isEmpty(name.value, name, "Please enter your name!", nameError)
            && isValidEmail(email.value, email, "Enter a valid email!", emailError));
    }
}

```

```

function showMessageAndFocus(isValid, focusInputElm, errMsg, errElm) {

```

```

    if (!isValid) {
        // Show errMsg on errElm, if provided.
        if (errElm !== undefined && errElm !== null
            && errMsg !== undefined && errMsg !== null) {
            errElm.innerHTML = errMsg;
        }
        // Set focus on Input Element for correcting error, if provided.
        if (focusInputElm !== undefined && focusInputElm !== null) {
            focusInputElm.className = "error";
            focusInputElm.focus();
        }
    } else {
        // Clear previous error message on errElm, if provided.
        if (errElm !== undefined && errElm !== null) {
            errElm.innerHTML = "";
        }
        if (focusInputElm !== undefined && focusInputElm !== null) {
            focusInputElm.className = "";
        }
    }
}

```

```
}  
}
```

```
/* Validate the inputValue is not empty (and not null). */
```

```
function isEmpty(inputValue, focusInputElm, errMsg, errElm) {
```

```
    var isValid = (inputValue !== null)
```

```
        && (inputValue.trim() !== "");
```

```
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
```

```
    return isValid;
```

```
}
```

```
/* Return true if the input value contains only digits (at least one) */
```

```
function isNumeric(inputValue, focusInputElm, errMsg, errElm) {
```

```
    var isValid = (inputValue !== null
```

```
        && inputValue.trim().match(/^\d+$/ ) !== null);
```

```
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
```

```
    return isValid;
```

```
}
```

```
/* Return true if the input value contains only letters (at least one) */
```

```
function isAlphabetic(inputValue, focusInputElm, errMsg, errElm) {
```

```
    var isValid = (inputValue !== null
```

```
        && inputValue.trim().match(/^[a-zA-Z]+$/ ) !== null) ;
```

```
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
```

```
    return isValid;
```

```
}
```

```
/* Return true if the input value contains only digits and letters (at least one) */
```

```
function isAlphanumeric(inputValue, focusInputElm, errMsg, errElm) {
```

```
    var isValid = (inputValue !== null
```

```

        && inputValue.trim().match(/^([0-9a-zA-Z]+)$/i) !== null);
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
    return isValid;
}

/* Return true if the input length is between minLength and maxLength */
function isLengthMinMax(inputValue, minLength, maxLength, focusInputElm, errMsg, errElm) {
    var inputValue = inputValue.trim();
    var isValid = (inputValue.length >= minLength) && (inputValue.length <= maxLength);
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
    return isValid;
}

// Return true if the input value is a valid email address
function isValidEmail(inputValue, focusInputElm, errMsg, errElm) {
    var isValid = (inputValue !== null)
        && (inputValue.trim().match(/^[-$]+\w+[-$]+@manipal\.(edu|in)$/i) !== null);

    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
    return isValid;
}

/* Return true if selection is made (not default of "") in <select> input */
function isSelected(inputValue, focusInputElm, errMsg, errElm) {
    // You need to set the default value of <select>'s <option> to "".
    var isValid = (inputValue !== "");
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
    return isValid;
}

/* Return true if the one of the checkboxes or radio buttons is checked

```

```

* Need to check all elements of the "names" */
function isChecked(inputName, focusInputElm, errMsg, errElm) {
    var inputElements = document.getElementsByName(inputName);
    var isChecked = false;
    for (var i = 0; i < inputElements.length; i++) {
        if (inputElements[i].checked) {
            isChecked = true; // found one element checked
            break;
        }
    }
    showMessageAndFocus(isChecked, focusInputElm, errMsg, errElm);
    return isChecked;
}

```

```

// Validate password, 6-8 characters of [a-zA-Z0-9_]
function isValidPassword(inputValue, focusInputElm, errMsg, errElm) {
    var isValid = (inputValue !== null)
        && (inputValue.trim().match(/^\w{6,8}$/) !== null);
    showMessageAndFocus(isValid, focusInputElm, errMsg, errElm);
    return isValid;
}

```

```

// Verify password.
function verifyPassword(pw, verifiedpw, focusInputElm, errMsg, errElm) {
    var isTheSame = (pw === verifiedpw);
    showMessageAndFocus(isTheSame, focusInputElm, errMsg, errElm);
    return isTheSame;
}

```

```
// The "onclick" handler for the "reset" button to clear the display
function clearDisplay() {
    var elms = document.getElementsByTagName("*"); // all tags
    for (var i = 0; i < elms.length; i++) {
        if ((elms[i].id).match(/Error$/)) {
            elms[i].innerHTML = "";
        }
        if (elms[i].className === "error") { // assume only one class
            elms[i].className = "";
        }
    }
    // Set initial focus
    document.getElementById("name").focus();
}
```

#### **//validatcss.css**

```
/* for error messages */
```

```
.red {
    color: red;
}
```

```
/* for the error input text fields */
```

```
input.error {
    border: 1px red inset;
    padding: 2px;
}
```

```
table {
    border: 0;
}
```

```
td {  
    margin: 0;  
    padding: 3px 10px 3px 3px;  
}
```

### **//validation.php**

```
<?php  
// define variables and set to empty values  
$nameErr = $emailErr = $genderErr = "";  
$name = $email = $gender = "";  
function test_input($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    if (empty($_POST["name"])) {  
        $nameErr = "Name is required";  
    } else {  
        $name = test_input($_POST["name"]);  
        if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {  
            $nameErr = "Only letters and white space allowed";  
        }  
    }  
  
    if (empty($_POST["email"])) {  
        $emailErr = "Email is required";  
    } else {  
        $email = test_input($_POST["email"]);  
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
```



```
$emailErr = "Invalid email format";
}
}

}
?>

<!DOCTYPE html>

<html>

<head>

    <title>Form Validation</title>

</head>

<body>

    <h2>Server-side Form Validation</h2>

    <form method="post" name="empfrm" id="eform" action="" >

Name: <input type="text" id="name" name="name" value="<?php echo $name;?>">

<span class="error">* <?php echo $nameErr;?></span>

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

    <br><br>

E-mail:

    <input type="text" id="email" name="email" value="<?php echo $email;?>">

    <span class="error">* <?php echo $emailErr;?></span>

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

    <br><br>

    <input type="submit" id="submit" name="submit" value="Submit">

    <input type="reset" value="CLEAR" id="reset"/>

    </form>

</body>

</html>
```

**// php validation example 2**

**//validation.php**

**//Refer to above eg JS and CSS**

```
<?php
```

```
// define variables and set to empty values
```

```
$nameErr = $emailErr = $genderErr = "";
```

```
$name = $email = $gender = "";
```

```
function test_input($data) {
```

```
    $data = trim($data);
```

```
    $data = stripslashes($data);
```

```
    $data = htmlspecialchars($data);
```

```
    return $data;
```

```
}
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    if (empty($_POST["name"])) {
```

```
        $nameErr = "Name is required";
```

```
    } else {
```

```
        $name = test_input($_POST["name"]);
```

```
        if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
```

```
            $nameErr = "Only letters and white space allowed (Server-side)";
```

```
}
```

```
}
```

```
if (empty($_POST["email"])) {
```

```
    $emailErr = "Email is required";
```

```
} else {
```

```
    $email = test_input($_POST["email"]);
```

```
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
```

```
        $emailErr = "Invalid email format (Server-side)";
```

```
}
```

```
}
```

```

}
?>

<!DOCTYPE html>

<html>

<head>

    <title>Test JavaScript Form Validation</title>

    <link rel="stylesheet" href="validatcss.css">

    <script type="text/javascript" src="validatefrm.js" ></script>

</head>


<body>

    <h2>Test JavaScript Form Validataion</h2>


    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
name="empfrm" id="efrm" onsubmit="return validateForm(this)">

        <table>

            <tr>

                <td>Name<span class="red">*</span></td>

                <td>

                    <input type="text" id="name" name="name" tabindex="1" placeholder="Enter" value="<?php
echo $name;?>"

                        />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</td>

                <td>

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

                    <span class="error">* <?php echo $nameErr;?></span>

                </td></tr>

            <tr>

                <td>Email<span class="red">*</span></td>

                <td>

                    <input type="text" id="email" name="email" value="<?php echo $email;?>" />

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

                    <span class="error">* <?php echo $emailErr;?></span>

                </td></tr>

        </table>

    </form>


```

```
<tr>
  <td>&nbsp;</td>
  <td>
    <input type="submit" value="SEND" id="submit"/>&nbsp;
    <input type="reset" value="CLEAR" id="reset"/>&nbsp;&nbsp;
  </td></tr>
</table>
</form>
</body>
</html>
```