

Input validations in Forms

Whenever we are working with Forms on webpages to submit some data, the Form validations are usually occurring at the server end, i.e. as the user entered all the necessary data and press the Submit button, all the filled data are submitted to server and subsequently validated. If the data entered by the user is incorrect or simply missing, the server sends all the data back to the client and requests for resubmitting of data through form with correct information. This is really a lengthy process which used to put a lot of burden on the server.

Using JavaScript, there are ways to validate most of the form's data on the user system itself before sending it to the web server. This is called Form validation and generally performs two functions:

- **Basic Form Validation** – In this, initially form must be checked to make sure all the mandatory fields are filled in. This is simple and requires a loop through each field in the form to check for data. For Example, Name, Mobile no. and email is mandatory.
- **Data Format Validation** – In this type of validation, the data entered can be checked for correct format and values (may be some value ranges). It is required that the JS code written must include appropriate logic to test correctness of data. For example, mobile no must be of 10 digits, email must be in correct format or PinCode shall be of 6 digits not starting with zero or Marks obtained shall be between 0 and 100.

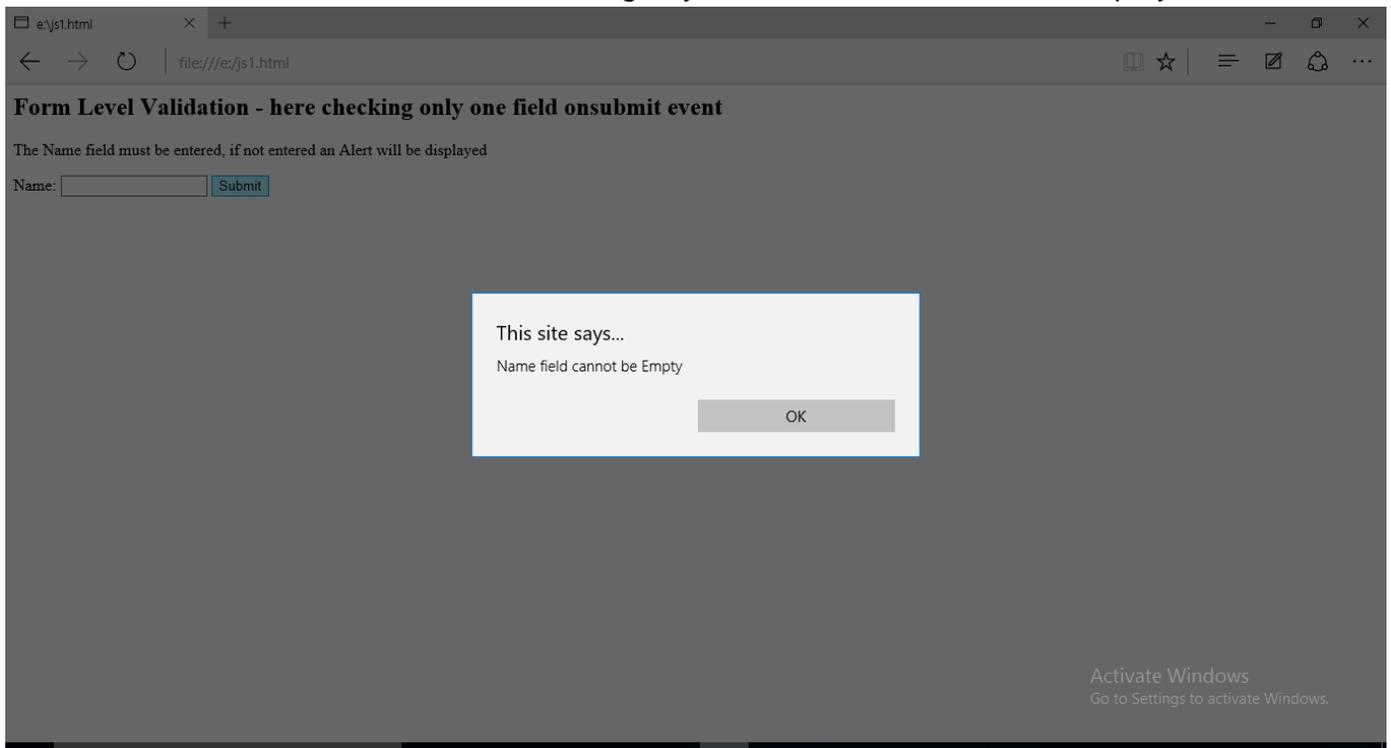
Basic Form Validation

Here, we are taking a simple form with only one Name Filed and enforcing user to fill the name and if name data is not filled and submit button is clicked it will display an alert to the user.

In the example below, a function **validateForm()** has been created and called to validate data when **onsubmit** event is occurring.

```
<html>
<head>
<h2> Form Level Validation - here checking only one field onsubmit event </h2>
<script>
function validateForm() {
  var x = document.forms["myForm"]["name"].value;
  if (x == "") {
    alert("Name field cannot be Empty");
    return false;
  }
}
</script>
</head>
<body>
<p> The Name field must be entered, if not entered an Alert will be displayed</p>
<form name="myForm" onsubmit="return validateForm()" method="post">
  Name: <input type="text" name="name">
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

If we click, the submit button without entering any value in Name field, it will display like



Data Format Validation

Now we will see how we can validate our entered form data before submitting it to the web server.

Lets consider a format, which require age must be entered and it shall be 18 or above. However maximum permitted age is 60 years.

In the example below, we have created a JavaScript Function myAge() to check the valid age entry. The function **myAge()** is called **onclick** event of Submit button. It will display the message as well as Alert about the Valid or Invalid age entry.

```
<html>
<body>
<h2>Data Validation to verify the Input</h2>

<p> Age cannot be empty and must be 18 & above or 60 & below ..</p>

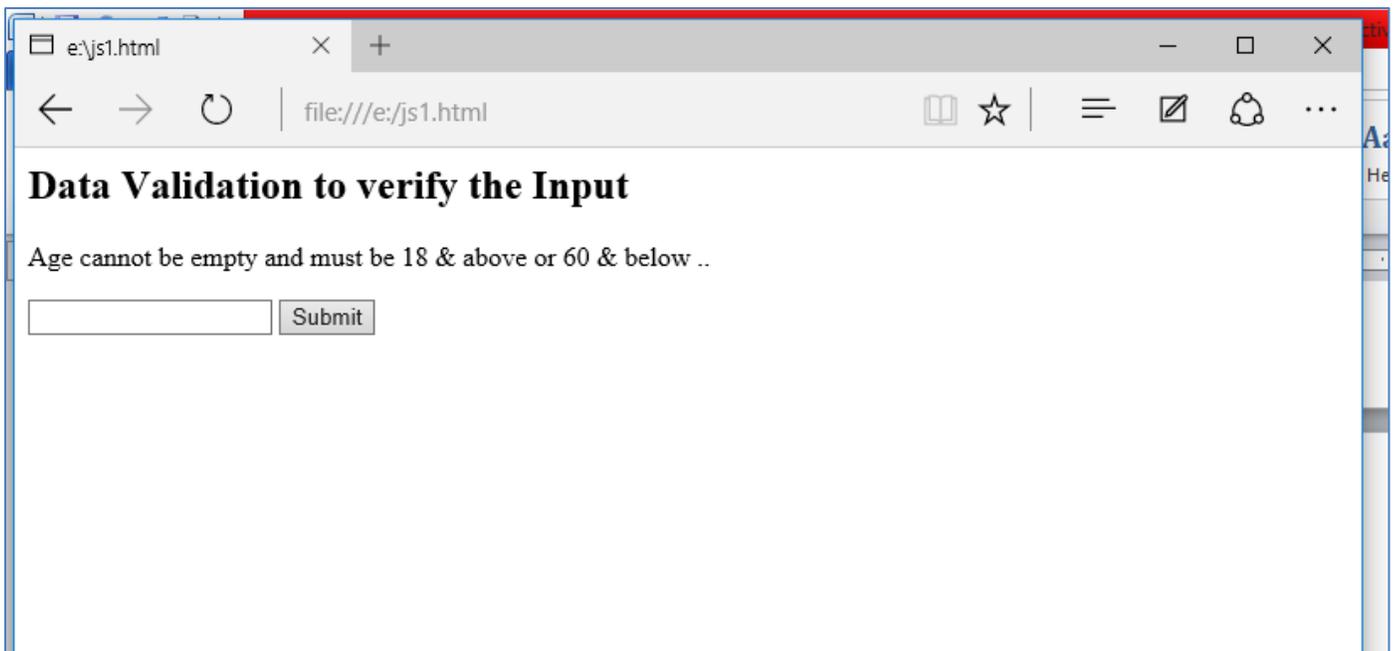
<input id="age">
<button type="button" onclick="myAge()">Submit</button>

<p id="testage"></p>

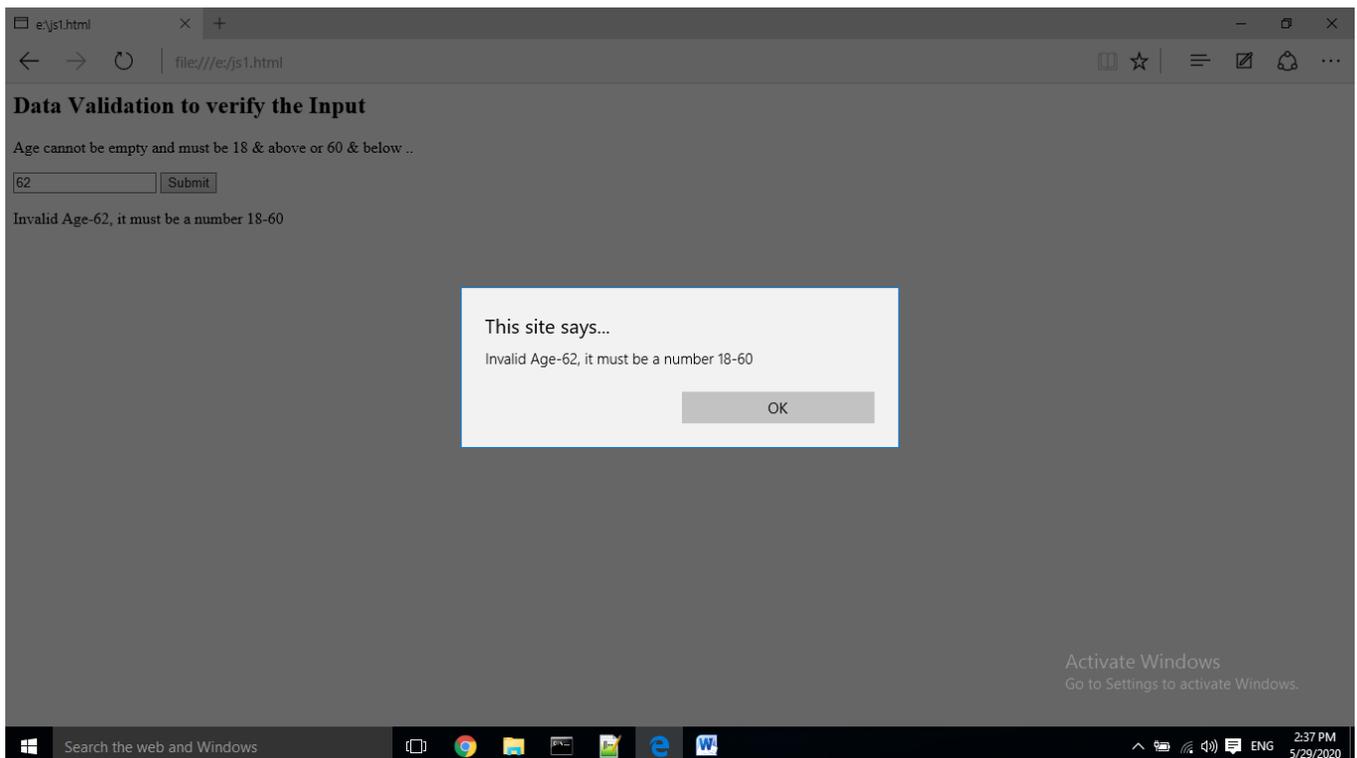
<script>
function myAge() {
  var n, msg;

  n = document.getElementById("age").value;

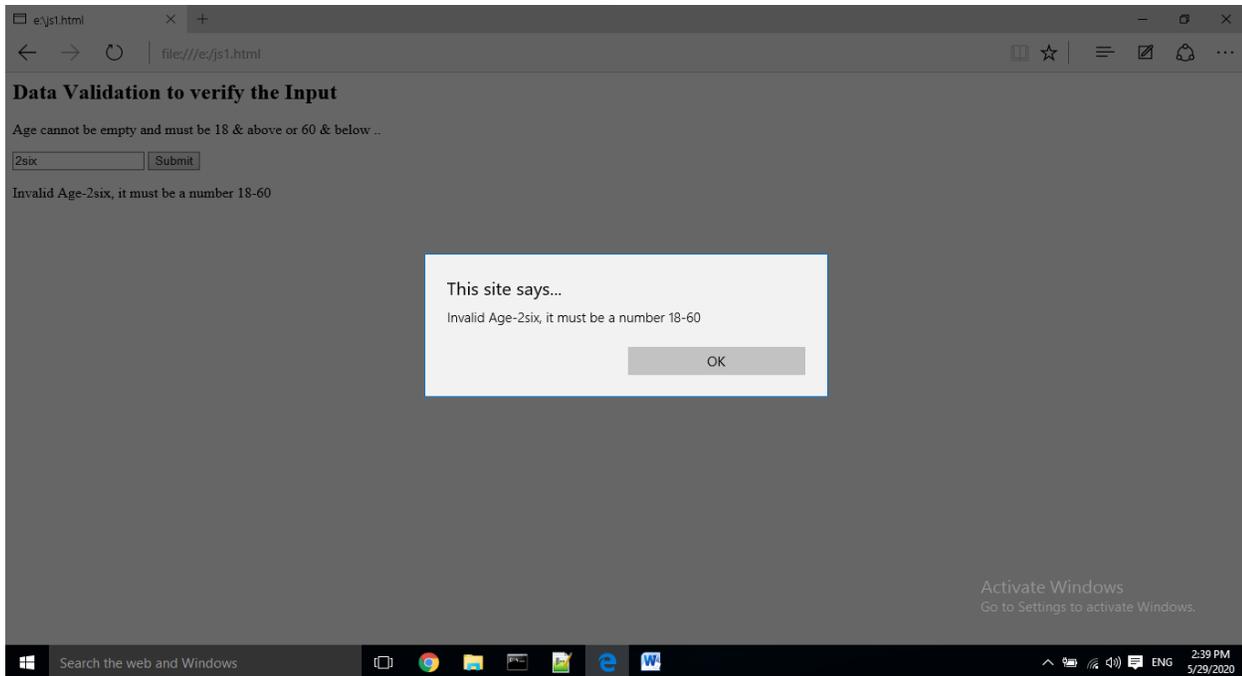
  // To check that the age is Not a Number or less than 18 or greater than 60
  if (isNaN(n) || n<18 || n>60) {
    msg = "Invalid Age-" + n + ", it must be a number 18-60";
  } else {
    msg = "Valid Age-" +n +", entered"; document.getElementById("testage").innerHTML
= msg;
    alert(msg);
  }
}
</script>
</body>
</html>
```



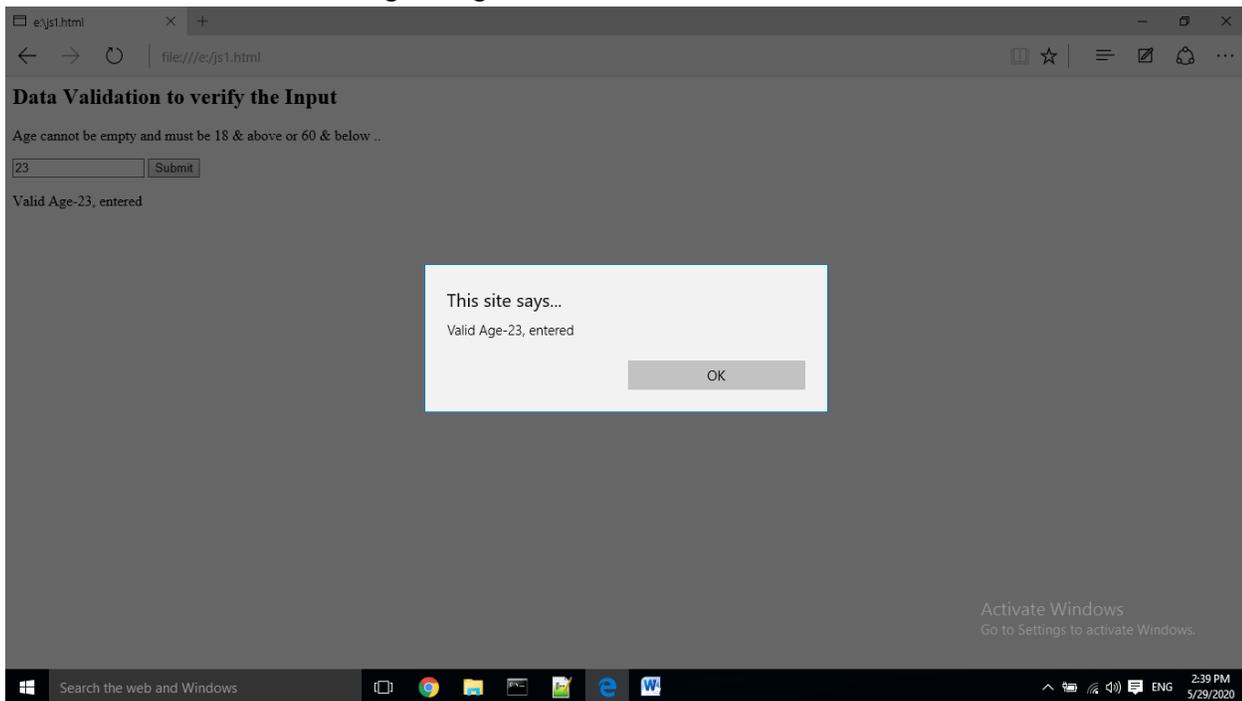
- Now make and invalid Entry of age i.e. 62



- Now, enter a non-numeric value, e.g. 2six



- Now, enter a valid age , e.g. 23



Assignment

1. What is form validation in JavaScript? How client side validation is different from server level validation.
2. Explain Types of Form Validation with example.