

# JavaScript Form validation

# form validation

When you enter data, the browser and/or the web server will check to see that the data is in the correct format and within the constraints set by the application.

Validation done in the browser is called **client-side** validation, while validation done on the server is called **server-side** validation. In this course, we are focusing on client-side validation.

- JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.
- **Basic Validation** – First of all, the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for data.
- **Data Format Validation** – Secondly, the data that is entered must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

# Automatic HTML Form Validation (by using **required** attribute)

**Ques.** WAP using JavaScript to create an input field(Fname) If this form field is empty, the **required** attribute should prevent this form from being submitted.

**Soln**

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<form action="/action_page.php" method="post">
```

```
<input type="text" name="fname" required>
```

```
<input type="text" name="lname" required>
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

# Question

Create a form with user Id and password.

If the user enters “user ID=123” and  
“password=abc”, only then the homepage  
should open.

Otherwise, an alert box must be displayed as  
“incorrect userid and password”

# SOLUTION

```
<html>
<head>
<title>Login page</title></head>

<body>
<h1>Simple Login Page</h1>

<form name="login">

  User id<input type="text" name="userid">
<br>

  Password<input type="password" name="pswrd">
<br>

  <input type="button" onclick="check(this.form)" value="Login">
  <br>
  <input type="reset" value="Reset">
</form>

<script language="javascript">
function check(form)
{

  if(form.userid.value == "123" && form.pswrd.value == "abc")

  {
    window.open("page1.html")
  }

  else

  {
    alert("Incorrect Password or Username")
  }

}

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

# To validate numeric input

Ques: WAP to create an input field that accepts only the numbers between 1 and 10.

Soln

```
<html><body>
<input type= text id="numb">
<button type="button" onclick="myFunction()">Submit</button>
<p id="demo"></p>
<script>
function myFunction() {
    var x, text;

    // Get the value of the input field with id="numb"
    x = document.getElementById("numb").value;

    // If x is Not a Number or less than one or greater than 10
    if (isNaN(x) || x < 1 || x > 10) {
        text = "Input not valid";
    } else {
        text = "Input OK";
    }
    document.getElementById("demo").innerHTML = text;
}
</script></body></html>
```

**Question: write code to create a text box. If the value of text box is blank, then alert box must be displayed “username cannot be null”, else open a web page named “page1.html”**

```
<html> <head> <title>Login page</title></head>
  <body>
    <form name="login">
      Username<input type="text" name="userid"> <br>
      Contact No.<input type="text" name="Contact"> <br>
      <input type="button" onclick="check(this.form)" value="enter details"><br>
      <input type="reset" value="Reset">
    </form>
    <script language="javascript">
      function check(form)
      {
        if(form.userid.value == "" || form.userid.value == null)
        {
          alert("user name cannot be null")
        }
        else
        {
          window.open("page1.html")
        }
      }
    </script></body></html>
```



# Data Validation

- Data validation is the process of ensuring that user input is correct

## **EXAMPLE OF DATA VALIDATION:**

- has the user filled in all required fields?
- has the user entered a valid date?
- has the user entered text in a numeric field?

Validation can be defined by many different methods, and deployed in many different ways.

- **Server side validation** is performed by a web server, after input has been sent to the server.
- **Client side validation** is performed by a web browser, before input is sent to a web server.

# HTML Constraint Validation

## Constraint Validation HTML Input Attributes

Attribute	Description
disabled	Specifies that the input element should be disabled
max	Specifies the maximum value of an input element
min	Specifies the minimum value of an input element
pattern	Specifies the value pattern of an input element
required	Specifies that the input field requires an element
type	Specifies the type of an input element

# The disabled Attribute

- The disabled attribute specifies that the input field is disabled.
- A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form:

# example

```
<html>
<body>
<form action="">
First name:<br>
<input type="text" name="firstname" value ="John" disabled>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>

</body>
</html>
```

First name:

John

Last name:

# min and max

- min and max attributes specify the minimum and maximum values for an <input> element.
- The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

Example:

```
<input type="number" name="quantity" min="1"
max="5"><br>
```

# example

```
<html>
```

```
<body>
```

```
<form >
```

```
<input type="number" id="i1" name="quantity" min="1" max="5"><br><br>
```

```
<input type="submit" value="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```