

In the above example, we have an input box and it must accept only number which is between 1 to 20 including 1 and 20, and if you enter any text value then it give message like "Input not valid", and if you enter value according to the condition given then it give message like "Input Ok". If by default you have not entered any value to it and hit on submit button then also it gives an error message in alert box like "Please enter any value".

उपरोक्त उदाहरण में, हमारे पास एक इनपुट बॉक्स है और ये केवल उसी संख्या को स्वीकार करेगा जो 1 से 20 के बीच तक का है, जिसमें 1 और 20 भी शामिल है, और यदि आप कोई टेक्स्ट वैल्यू दर्ज करते हैं, तो यह "इनपुट मान्य नहीं" जैसे संदेश देगा, और यदि आप मान दी गई कंडीशन के अनुरूप दर्ज करते हैं तो यह "इनपुट ओके" जैसा संदेश देगा। यदि गलती से आपने इसे बिना कोई वैल्यू दर्ज नहीं किये सबमिट बटन पर हिट करते हैं, तो यह "कृपया अपना मूल्य दर्ज करें" जैसे अलर्ट बॉक्स में एक त्रुटि संदेश देता है।

Example to use JavaScript's type of:
जावास्क्रिप्ट टाइपऑफ का एक उदाहरण:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Example of JavaScript : typeof</h1>
6
7 <script type="text/javascript">
8
9   var x = 200;
10  if(typeof x == 'number')
11  {
12    document.write(x + " is a number <br/>");
13  }
14  else
15  {
16    document.write(x + " is not a number <br/>");
17  }
18 </script>
19
20 </body>
21 </html>
        
```

Automatic form validation in JavaScript: HTML form validation can be performed automatically by the browser. If a form field (eName) is empty, the **required attribute** prevents this form from being submitted.

जावास्क्रिप्ट में आटोमैटिक रूप से फॉर्म वैलिडेशन: HTML फार्म वैलीडेट ब्राउजर द्वारा स्वचालित रूप से किया जाता है। यदि कोई फार्म फील्ड (eName) खाली होता है, तो रिक्वायर्ड ऐट्रिब्यूट (आवश्यक विशेषता) इस फॉर्म को सबमिट होने से रोक देती है।

Note1: If you click submit, without filling out the text field, your browser will display an error message. यदि आप टेक्स्ट फील्ड को बिना भरे सबमिट पर क्लिक करते हैं, तो आपका ब्राउजर एक एरर मैसेज डिस्प्ले करेगा।

Note2: Automatic HTML form validation does not work in Internet Explorer 9 or earlier version. Internet Explorer 9 या पहले के वर्जन में आटोमैटिक (स्वचालित) HTML फॉर्म वैलिडेशन काम नहीं करता है।


```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h4> Fill the Form </h4>
5 <form action="/action_page.php" method="post">
6   Enter your Name : <input type="text" name="eName" required>
7   <br><br>
8   <input type="submit" value="Submit">
9 </form>
10 </body>
11 </html>

```

PREVIEW

Fill the Form

Enter your Name :

Submit

Please fill out this field.

Open in your browser and check the result. अपने ब्राउजर में ओपेन करें और परिणाम जांचें।

See the example given below: Here we are going to validate the name and password field. Note that the name can't be empty and password can't be less than 6 characters long.

नीचे दिए गए उदाहरण पर एक नजर डालें, हम नाम और पासवर्ड फील्ड को वैलिडेट करने जा रहे हैं। इसमें नाम खाली न हो और पासवर्ड 6 कैरेक्टर से कम लंबा इनपुट ना ले।

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h2>Example to validate the name and password field in the form:</h2>
5 <p>Please input name and password which can't be empty and the password can't be less than 8 characters long:</p>
6 <h3>Login Form</h3>
7 <form name="myform" method="post" action="abc.jsp" onsubmit="validateform()">
8   Name:<br>
9   <input type="text" name="name"><br>
10  Password:<br>
11  <input type="password" name="password"><br> <br>
12  <input type="submit" value="Submit">
13 </form>
14 <script>
15 function validateform()
16 {
17   var name = document.myform.name.value;
18   var password=document.myform.password.value;
19
20   if(name == null|| name=="")
21   {
22     alert("Name can't be blank");
23     return false;
24   }
25   else if(password == null|| password=="")
26   {
27     alert("Password can't be blank");
28     return false;
29   }
30   else if(password.length<8)
31   {
32     alert("Password must be at least 8 character long.");
33     return false;
34   }
35   document.write("Data successfully submitted.");
36 }</script>
</body></html>

```

Generate preview file

Example to validate the name and password field in the form:

Please input name and password which can't be empty and the password can't be less than 8 characters long.

Login Form

Name:

Password:

Submit

Open the above code in your browser and check the result.

अपने ब्राउजर में ऊपर दिए गए कोड को ओपेन करें और परिणाम जांचें।

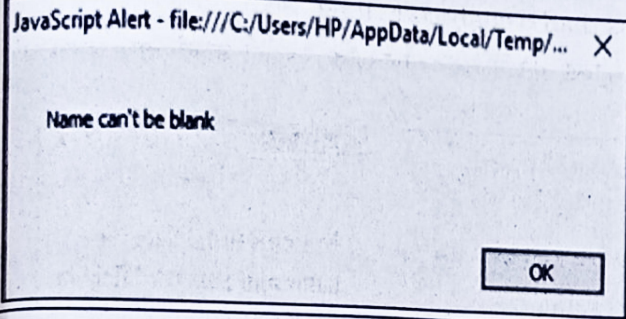
Output of the above code is given below: ऊपर दिए गए कोड का आउटपुट नीचे दिया गया है-

If you click submit without entering the name and password then the following message is shown:

Login Form

Name:

Password:

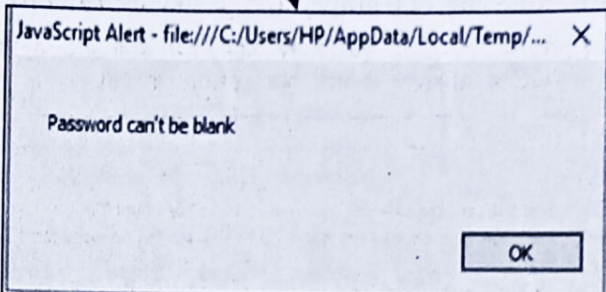


If you click submit button by just entering the name, but password is empty then the following message is shown:

Login Form

Name:

Password:



By entering the name and password which must be at least 8 characters long then your data successfully submitted.

Login Form

Name:

Password:

Data successfully submitted.

Let us take an example in the real life web development scenario; that how **the basic validation and data format validation** is being implemented. To understand it see the following code step by step and use or try it in your editor.

आइए हम अब वास्तविक जीवन में वेब डेवलपमेंट सिनैरिओ का एक उदाहरण लेते हैं कि कैसे बेसिक वैलिडेशन और डेटा फॉर्मेट वैलिडेशन कार्यान्वित (इम्प्लीमेंट) किया जाता है। इसे समझने के लिए नीचे लिखे गए कोड को स्टेप बाई स्टेप देखें एवं इसे अपने एडिटर में यूज या ट्राई करें।

Step 1: Design your form using HTML and CSS style. The form having the following details:
HTML और CSS स्टाइल का उपयोग करके अपना फॉर्म डिजाइन करें। जिसमें नीचे दिए गए डीटेल होंगे-

INTRESTED TOPIC FIELD [not required], FULL NAME FIELD [required, no error text], AGE FIELD [required, error text], EMAIL FIELD, SELECT FIELD [choose your qualification].


```
1 |<!DOCTYPE html>
2 |<html>
3 |<head>
4 |
5 |   <meta charset="UTF-8">
6 |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 |   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8 |   <title>Form Validation</title>
9 |
10 |  <style>
11 |    body {
12 |      margin: 0;
13 |      font-family: "Open Sans", sans-serif;
14 |      font-size: 16px;
15 |      line-height: 22px;
16 |      font-weight: 400;
17 |      color: #242424;
18 |      -webkit-font-smoothing: antialiased;
19 |      background: #f6f6f6;
20 |    }
21 |
22 |    div, span, form, label, input, p {
23 |      box-sizing: border-box; color:blue;
24 |    }
25 |
26 |    h1 {
27 |      margin: 0 0 12px;
28 |      font-size: 28px;
29 |      line-height: 40px;
30 |    }
31 |
32 |    p {
33 |      margin: 0 0 36px;
34 |    }
35 |
36 |    a {
37 |      color: #3498db;
38 |    }
39 |
40 |    .centered {
41 |      text-align: center;
42 |    }
43 |
44 |    .wrapper {
45 |      margin: 40px auto;
46 |      width: 300px;
47 |    }
48 |
49 |    label {
50 |      margin: 0 0 4px;
```



```
51     font-size: 11px;
52     line-height: 17px;
53     text-transform: uppercase;
54     letter-spacing: 1px;
55     font-weight: 700;
56     color: #797979;
57     display: block;
58 }
59
60 .form-input-wrap {
61     position: relative;
62 }
63
64 .form-input-wrap--select:before {
65     position: absolute;
66     top: 42px;
67     right: 14px;
68     z-index: 1;
69     content: '';
70     border: 5px solid transparent;
71     border-top-color: #666666;
72     pointer-events: none;
73     display: block;
74 }
75
76 .select-wrap {
77     position: relative;
78 }
79
80 .input {
81     margin: 0 0 22px;
82     padding: 11px 16px 11px;
83     width: 100%;
84     font-size: 16px;
85     line-height: 22px;
86     font-weight: 400;
87     border: 1px solid #cccccc;
88     border-radius: 0;
89     outline: none;
90     font-weight: 400;
91     color: #242424;
92     -webkit-appearance: none;
93     -moz-appearance: none;
94     display: block;
95     background: #ffffff;
96 }
97
98 .errored {
99     border-color: #e74c3c;
100     background: #fff0ed;
101 }
```



```
102
103     .input:focus {
104         border-color: #aaaaaa;
105         outline: none;
106     }
107
108     .input-error {
109         position: relative;
110         top: -18px;
111         font-size: 12px;
112         line-height: 18px;
113         text-align: right;
114         font-weight: 600;
115         color: #e74c3c;
116         display: block;
117     }
118
119     .button {
120         margin-top: 8px;
121         padding: 10px 26px;
122         font-size: 12px;
123         line-height: 18px;
124         font-weight: 700;
125         text-transform: uppercase;
126         color: #ffffff;
127         border: none;
128         border-radius: 3px;
129         outline: none;
130         cursor: pointer;
131         display: inline-block;
132         background: #3498db;
133     }
134
135     .button:hover {
136         background: #2b87c5;
137     }
138 </style>
139 </head>
```



```
140 <body>
141   <div class="wrapper">
142     <h1>Fill Your Details</h1>
143     <form name="myForm" autocomplete="off" novalidate>
144
145       <div class="form-input-wrap">
146         <label for="myInput1">Interested Topic Field (not required)</label>
147         <input type="text" name="myInput1" id="myInput" class="input" />
148       </div>
149
150       <div class="form-input-wrap">
151         <label for="myInput2">Full Name Field(required, no error text)</label>
152         <input type="text" name="myInput2" id="myInput2" class="input" required />
153       </div>
154
155       <div class="form-input-wrap">
156         <label for="myInput3">Age Field(required, error text)</label>
157         <input type="text" name="myInput3" id="myInput3" class="input"
158         data-error-text="Error text custom to this field" required />
159       </div>
160
161       <div class="form-input-wrap">
162         <label for="myInput4">Email Field</label>
163         <input type="email" name="myInput4" id="myInput4" class="input"
164         data-error-text="Please enter a valid email address" required />
165       </div>
166
167       <div class="form-input-wrap form-input-wrap--select">
168         <label for="myInput5">Select Field</label>
169         <select name="myInput5" id="myInput5" class="input"
170         data-error-text="Please enter your Qualification" required>
171           <option value="">[Choose your Qualification]</option>
172           <option value="Option 1">Below 10th</option>
173           <option value="Option 2">Matric</option>
174           <option value="Option 3">Inter</option>
175           <option value="Option 4">Degree</option>
176           <option value="Option 5">PG</option>
177         </select>
178       </div>
179       <div class="centered">
180         <input type="submit" value="Submit" class="button" />
181       </div>
182     </form>
```



```

183
184     <script src="validator-standalone.js"></script>
185
186     <script>
187         const myForm = document.querySelector("form[name=myForm]");
188         myForm.addEventListener("submit", function(event) {
189             if(validator.checkForm(event.target)) {
190                 alert("Continue processing the form!");
191             }
192             event.preventDefault();
193         });
194     </script>
195
196 </div>
197 </body>
198 </html>

```

Now, Save the above code with the file name **FormValidation.html**. ऊपर दिए गए कोड को **FormValidation.html** नाम से सेव करें।

Step 2: Since Designing part is over, Now it's time to develop Data validation using JavaScript and save this file with the file name **validator-standalone.js** in the same html file folder were the previous html file is saved.

डिजाइनिंग करने के बाद अब इसमें जावास्क्रिप्ट वैलिडेशन लगाने के लिए नीचे दिए गए कोड को **validator-standalone.js** नाम में सेव करें जिसका लोकेशन या फोल्डर वही हो जहाँ आपने पहले फाइल (अर्थात **FormValidation.html** वाली फाइल) को सेव किया हों।

Since, the following code is used to add the validation in the form design which is also called form validation. After putting this code you are restricted to fill all the required fields in the forms. Without javascript validation the forms gets submitted without entering any data in the field which is not good in case of database. So, to restrict this you must need to validate the fields according to the requirements.

चूँकि, निम्नलिखित कोड का उपयोग फॉर्म डिजाइन में वैलिडेशन को जोड़ने के लिए किया जाता है जिसे फॉर्म वैलिडेशन भी कहा जाता है। इस कोड को डालने के बाद आप सभी आवश्यक फील्ड को भरने के लिए प्रतिबंधित होते हैं। जावास्क्रिप्ट वैलिडेशन के बिना, फील्ड्स में किसी भी डेटा को दर्ज किए बिना, आप इसे सबमिट कर सकते हैं जो डेटाबेस के मामले में अच्छा नहीं होता है, तो आपको इसे प्रतिबंधित करने के लिए आवश्यकताओं के अनुसार फील्ड को वैलिडेट करना होगा।

Let us see how do we validate the from with the following code: यहाँ हम देखते हैं कि नीचे लिखे गए कोड द्वारा कैसे फॉर्म को वाइलिडेट किया जाता है:


```
1 (function(window) {
2   function validator()
3   {
4     var _validator = {},
5         settings = {
6       errored_input_classname: "errored",
7       error_text_classname: "input-error",
8       error_text_data_attribute: "data-error-text"
9     };
10    _validator.checkForm = function(form, settings_override) {
11      if(settings_override) {
12        settings = Object.assign(settings, settings_override);
13      }
14      this.resetForm(form);
15      var errorsFound = false,
16          requiredInputs = form.querySelectorAll("input[required], textarea[required], select[required]"),
17          re = email_regex = /^(^<>()|[\]\[\]\/\.,;:\s@"]+\.[^\<>()|[\]\[\]\/\.,;:\s@"]+)|(\.+)|(\.[0-9]{1,3}\.){0,3}\.?[0-9]{1,3}\.?[0-9]{1,3}\.?[0-9]{1,3})|((\[[a-zA-Z\-\0-9]+\.\.]+\.[a-zA-Z]{2,}))$/;
18      for (var i = 0; i < requiredInputs.length; i++) {
19        var input = requiredInputs[i],
20            val = input.value;
21        if (!val) {
22          this.createError(input);
23          errorsFound = true;
24        } else if (input.type === "email" && re.test(val) === false) {
25          this.createError(input);
26          errorsFound = true;
27        }
28      }
29      if (errorsFound) {
30        return false;
31      }
32      return true;
33    };
34    _validator.createError = function(input) {
35      var errorMessage = input.getAttribute(settings.error_text_data_attribute),
36          errorMessageNode = document.createElement("span");
37
38      input.classList.add(settings.errored_input_classname);
39      errorMessageNode.classList.add(settings.error_text_classname);
40      errorMessageNode.innerText = errorMessage;
41
42      // Insert error message directly after the input element
43      input.parentNode.insertBefore(errorMessageNode, input.nextSibling);
44    };
45    _validator.resetForm = function(form) {
46      var erroredInputs = form.querySelectorAll("." + settings.errored_input_classname),
47          errorMessageNodes = form.querySelectorAll("." + settings.error_text_classname);
48
49      for (var j = 0; j < erroredInputs.length; j++) {
50        var input = erroredInputs[j];
51        input.classList.remove(settings.errored_input_classname);
52      }
53      for (var k = 0; k < errorMessageNodes.length; k++) {
54        var errorMessageNode = errorMessageNodes[k];
55        errorMessageNode.parentNode.removeChild(errorMessageNode);
56      }
57    }
58  }
59 }
60 )
```



```

57     };
58     return _validator;
59 }
60 if (typeof (window.validator) === 'undefined') {
61     window.validator = validator();
62 }
63 })(window);
    
```

Now, Save the above code at the previous file location/folder with the file name **validator-standalone.js**. अब ऊपर लिखे गए कोड को पुराने फाइल के लोकेशन या फोल्डर में **validator-standalone.js** नाम से सेव करें।

Step 3: Open your **FormValidation.html** file in browser for the output.

ब्राउजर में अपनी **FormValidation.html** फाइल को आउटपुट के लिए खोलें।

Step 4: Fill the form and check your fields whether your validations are working properly or not.

आउटपुट में आए हुए फॉर्म को भरें और अपने फील्ड्स की जांच करें कि आपका वैलिडेशन ठीक से काम कर रहा है या नहीं।

Output of the above code without javascript validation:	Output of the above code with javascript validation:
<p>Before javascript validation when you click submit button without entering any data then the output of the above code is given below:</p>	<p>After javascript validation when you click submit button without entering any data then the output of the above code is given below:</p>
<div style="border: 1px solid black; padding: 10px;"> <p>Fill Your Details</p> <p>INTERESTED TOPIC FIELD (NOT REQUIRED)</p> <input type="text"/> <p>FULL NAME FIELD(REQUIRED, NO ERROR TEXT)</p> <input type="text"/> <p>AGE FIELD(REQUIRED, ERROR TEXT)</p> <input type="text"/> <p>EMAIL FIELD</p> <input type="text"/> <p>SELECT FIELD</p> <p>[Choose your Qualification] ▾</p> <p style="text-align: center;">SUBMIT</p> </div>	<div style="border: 1px solid black; padding: 10px;"> <p>Fill Your Details</p> <p>INTERESTED TOPIC FIELD (NOT REQUIRED)</p> <input type="text"/> <p>FULL NAME FIELD(REQUIRED, NO ERROR TEXT)</p> <input type="text"/> <p>AGE FIELD(REQUIRED, ERROR TEXT)</p> <input type="text"/> <p style="text-align: right;">Error text custom to this field ←</p> <p>EMAIL FIELD</p> <input type="text"/> <p style="text-align: right;">Please enter a valid email address ←</p> <p>SELECT FIELD</p> <p>[Choose your Qualification] ▾</p> <p style="text-align: right;">Please enter your Qualification ←</p> <p style="text-align: center;">SUBMIT</p> </div>