

<pre> 1 <!DOCTYPE html> 2 <html> 3 <body> 4 <h2>Prompt Dialogue Box</h2> 5 <button onclick="myP()">Try it</button> 6 <p id="demo"></p> 7 <script> 8 function myP() 9 { 10 var eName; 11 var person = prompt("Your Sweetest Name Please ?:", "Balaji"); 12 if (person == null person == "") 13 eName = "User cancelled the prompt."; 14 else 15 eName = "Hello " + person + "! How are you Buddy?"; 16 document.getElementById("demo").innerHTML = eName; 17 } 18 </script> 19 </body> 20 </html> </pre>	<p>PREVIEW</p> <p>Prompt Dialogue Box</p> <p>Try it → Hello Balaji ! How are you Buddy ?</p> <p>An embedded page on this page says Your Sweetest Name Please ?:</p> <p>OK Cancel</p> <p>↓ User cancelled the prompt.</p>
--	---

6.6 Java Script Events - जावा स्क्रिप्ट इवेंट्स

JavaScript event is an action which is performed on any HTML elements. It is used to process based on Events. An HTML event is performed by either browser or by user's action.

जावास्क्रिप्ट इवेंट एक क्रिया है जो किसी भी HTML एलीमेंट्स पर की जाती है। यह इवेंट के बेस्ड पर प्रोसेस करता है। एक HTML इवेंट ब्राउजर या उपयोगकर्ता के एक्शन के द्वारा परफॉर्म की जाती है।

Below are some examples of HTML events: नीचे HTML घटनाओं के कुछ उदाहरण दिए गए हैं:

- When a user clicks the mouse. - जब कोई उपयोगकर्ता माउस पर क्लिक करता है।
- When a web page has loaded. - जब कोई वेब पेज लोड हुआ हो।
- When an image has been loaded. - जब कोई चित्र लोड किया गया हो।
- When the mouse moves over an element. - जब माउस किसी एलीमेंट पर चलता है।
- When an input field is changed. - जब एक इनपुट फील्ड बदली जाती है।
- When an HTML form is submitted. - जब HTML फॉर्म सबमिट किया जाता है।
- When a user strokes a key. - जब एक उपयोगकर्ता एक कुंजी स्ट्रोक करता है।

JavaScript events can be placed within single quotes or double quotes.
जावास्क्रिप्ट इवेंट को एकल उद्धरण या दोहरे उद्धरण के भीतर रखा जा सकता है।

With Single Quote: <HTML-element event-name='JavaScript code'><HTML-element>

With Double Quote: <HTML-element event-name="JavaScript code"><HTML-element>

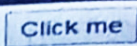
In this below example we will try to understand a relation between Event and JavaScript. In example the content of the <div> element is changed when a user clicks on it.

यहां इस उदाहरण में हम इवेंट और जावास्क्रिप्ट के बीच एक संबंध को समझने का प्रयास करेंगे। इस उदाहरण में, उपयोगकर्ता द्वारा उस पर क्लिक करने पर <div> एलीमेंट की कंटेंट बदल जाती है।

<pre> 1 <!DOCTYPE html> 2 <html> 3 <body> 4 <h2> 5 <div onclick="this.innerHTML='Wow'"> Hit Me... </div> 6 </h2> 7 </body> 8 </html> </pre>	<p>PREVIEW</p> <p>Hit Me...</p>
---	--

In below example, an **onclick** attribute is added for button input element.

नीचे उदाहरण में, बटन इनपुट एलीमेंट के लिए एक ऑनक्लिक विशेषता ऐट्रिब्यूट जोड़ी जाती है।

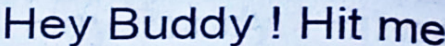
	h1 h2 h3 Tp b i u span link img ul •	PREVIEW
1	<!DOCTYPE html>	
2	<html>	
3	<body>	
4	<input type="button" name="clickme" value="Click me"	
5	onclick="alert('You just clicked on a button');"/>	
6	<p id="demo"></p>	
7	</body></html>	

Take a look at below example, a function is called from the event handler.

नीचे दिए गए उदाहरण पर एक नजर डालें, एक फंक्शन को इवेंट हैंडलर से काल किए किया जाता है।

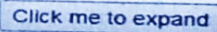
Note: 'this' keyword can be used to refer current element within JavaScript event.

नोट: 'this' कीवर्ड का उपयोग जावास्क्रिप्ट इवेंट के भीतर वर्तमान एलीमेंट को संदर्भित करने के लिए किया जा सकता है।

	h1 h2 h3 Tp b i u span link img ul •	PREVIEW
1	<!DOCTYPE html>	
2	<html>	
3	<body>	
4	<h1 onclick="changeText(this)"> Hey Buddy ! Hit me</h1>	
5	<script>	
6	function changeText(id)	
7	{	
8	id.innerHTML = "Awesome Dude, you did it";	
9	}	
10	</script>	
11	</body>	
12	</html>	

In below example, onclick event code will change the size of button.

नीचे दिए गए उदाहरण में, ऑनक्लिक इवेंट कोड बटन के आकार को बदल देगा।

	h1 h2 h3 Tp b i u span link img ul •	PREVIEW
1	<!DOCTYPE html>	
2	<html>	
3	<body>	
4	<input type="button" name="clickme" value="Click me to expand"	
5	onclick="changeButtonSize(this);" />	
6	<script>	
7	function changeButtonSize(\$this) {	
8	\$this.style.padding="10px";	
9	}	
10	</script>	
11	<p id="demo"></p>	
12	</body>	
13	</html>	
14	</html>	

It is rare that JavaScript code is of single line. So it is more common code practice to define a function for event. **onClick()** is the most frequently used event type which occurs when a user clicks the button of his mouse. You can put your validation, warning etc., against this event type.

जावास्क्रिप्ट के कोड सिंगल लाइन के हैं। यह घटना के लिए एक समारोह को परिभाषित करने के लिए अधिक सामान्य कोड अभ्यास है। **onClick()** सबसे अधिक इस्तेमाल किया जाने वाला इवेंट प्रकार है जो तब होता है जब कोई उपयोगकर्ता अपने माउस के बटन पर क्लिक करता है। आप इस इवेंट के प्रकार के विपरीत वेलिडेशन, चेतावनी आदि रख सकते हैं।

```

1 <html>
2 <head>
3 <script type = "text/javascript">
4     function callingFun() {
5         alert("Yes... thankx for Calling");
6     }
7 </script>
8 </head>
9
10 <body>
11 <p>Click the following button and see result</p>
12 <form>
13 <button onclick = "callingFun()">Hit me</button>
14 </form>
15 </body>
16 </html?
    
```

PREVIEW

Click the following button and see result

Hit me

The onchange Event: This event is frequently used in combination with validation of input fields. How to use the onchange event can be understood with the help of the example given below. The **UpperCase()** function will be called when a user changes the content of an input field.

आनचेंज इवेंट: इस इवेंट को अक्सर इनपुट क्षेत्रों के सत्यापन के साथ संयोजन में उपयोग किया जाता है। onchange इवेंट का उपयोग कैसे करें नीचे दिए गए उदाहरण की मदद से समझा जा सकता है। जब कोई उपयोगकर्ता इनपुट फील्ड की सामग्री को बदलना चाहता है तो **UpperCase()** फंक्शन को काल किया जाता है।

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <script>
5     function myFun()
6     {
7         var x = document.getElementById("emName");
8         x.value = x.value.toUpperCase();
9     }
10 </script>
11 </head>
12 <body>
13 <h4>Enter Your Name:<input id="emName" onchange="myFun()">
14 <br><br>
15 <p>When you leave the input field, a function is triggered <br>
16 which transforms the input text to upper case.</p>
17 </h4>
18 </body>
19 </html>
    
```

Generate preview file

Enter Your Name: hy i am balaji

When you leave the input field, a function is triggered which transforms the input text to upper case.

The above text converted to upper case as given below:

Enter Your Name: HY I AM BALAJI

When you leave the input field, a function is triggered which transforms the input text to upper case.

onmouseover and onmouseout Event:

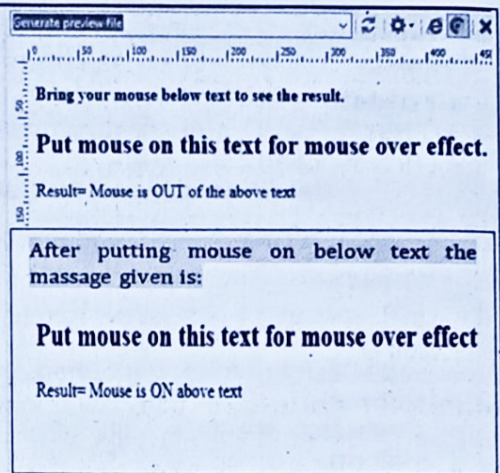
These two types of event helps in creating nice effects with images or even with text as well. The **onmouseover** event triggers when the mouse is brought over any element and the **onmouseout** activates when the mouse is moved out from that element. The following example could be tried.

ये दो इवेंट आपको इमेज के साथ या टेक्स्ट के साथ भी अच्छा प्रभाव बनाने में मदद करते हैं। जब माउस किसी एलिमेंट पर लाया जाता है तो ऑनमाउसओवर इवेंट चालू हो जाता है और जब माउस उस एलिमेंट से हटाते है तो ऑनमाउसआउट एक्टिवेट हो जाता है। निम्नलिखित उदाहरण को ट्राई किया जा सकता है।

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <script type="text/javascript">
5     function over()
6     {
7       document.getElementById("overMe").innerHTML="Result= Mouse is ON above text";
8     }
9     function out()
10    {
11      document.getElementById("overMe").innerHTML="Result= Mouse is OUT of the above text";
12    }
13  </script>
14 </head>
15 <body>
16   <h4>Bring your mouse below text to see the result.</h4>
17   <div onmouseover="over()" onmouseout="out()">
18     <h2>Put mouse on this text for mouse over effect.</h2>
19   </div>
20   <span id="overMe"></span>
21 </body>
22 </html>

```



onsubmit Event: onsubmit event occurs when one tries to submit a form. Form validation can be put against this event type.

ऑनसबमिट इवेंट: जब कोई फॉर्म सबमिट करता है, तो ऑनसबमिट इवेंट अक्कर्स (यूज) होती है। इस इवेंट के प्रकार के में फॉर्म वैलिडेशन लगाया जा सकता है।

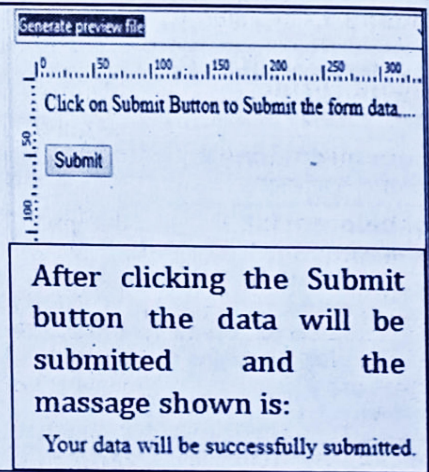
The example given here shows how to use **onsubmit**. Here we are calling a **validate()** function before submitting a form data to the webserver. If **validate()** function returns true, the form will be submitted, otherwise it will not submit the data.

नीचे दिए गए उदाहरण से पता चलता है कि **onsubmit** का उपयोग कैसे करें। यहां हम वेबसर्वर पर एक फॉर्म डेटा सबमिट करने से पहले एक **validate()** फंक्शन को काल करते हैं। यदि **validate()** फंक्शन ट्रू रिटर्न करता है, तो फॉर्म सबमिट होगा, अन्यथा यह डेटा सबमिट नहीं करेगा।

```

1 <html>
2   <head>
3     <script type = "text/javascript">
4       function callFun()
5       {
6         document.write("Your data will be successfully submitted.");
7       }
8     </script>
9   </head>
10  <body>
11    <form method = "POST" action = "t.cgi" onSubmit = "callFun()">
12      Click on Submit Button to Submit the form data....<br><br>
13      <input type = "submit" value = "Submit" />
14    </form>
15  </body>
16 </html>

```



Common HTML events list - आम HTML इवेंट की सूची-

Event घटना	Description विवरण
onchange ऑनचेंज	An HTML element value has been changed. एक HTML एलीमेंट के मान को बदल देता है।
onclick ऑनक्लिक	When an HTML element has been clicked. जब कोई HTML एलीमेंट क्लिक किया गया हो।
onmouseover ऑनमाउसओवर	When user moves the mouse over an HTML element. जब उपयोगकर्ता HTML एलीमेंट पर माउस ले जाता है।

onmouseout ऑनमाउसआउट	When user moves the mouse out of an HTML element. जब उपयोगकर्ता माउस को HTML एलीमेंट से बाहर निकालता है।
onkeydown ऑनकीडाउन	When user presses a keyboard key. जब उपयोगकर्ता एक कीबोर्ड कुंजी दबाता है।
onload ऑनलोड	When the browser completes loading the page. जब ब्राउजर पेज को लोड करना पूरा कर लेता है।

The onload and onunload Events: When the user enters or leaves the page the onload and onunload events are activated. The onload event is used for checking the visitor's browser type and browser version, and loads the proper version of the web page based on the information. These events are also used to deal with cookies.

onload और onunload इवेंट: जब उपयोगकर्ता किसी पेज पर प्रवेश करता है या पेज छोड़ता है, तो onload और onunload इवेंट ट्रिगर होते हैं। ऑनलोड इवेंट का उपयोग आगंतुक (विजिटर) के ब्राउजर प्रकार और ब्राउजर संस्करण की जांच करने के लिए किया जा सकता है, और जानकारी के आधार पर वेब पेज के उचित संस्करण को लोड किया जा सकता है। onload और onunload इवेंट्स का उपयोग कुकीज से निपटने के लिए किया जा सकता है।

There are many more events available in HTML and the list is much longer. HTML में कई और इवेंट उपलब्ध हैं और इनकी सूची बहुत लंबी है।

HTML5 Standard Events: The standard HTML5 events are listed here for your reference. Here script indicates a Javascript function to be executed against that event.

HTML5 मानक इवेंट: मानक HTML 5 इवेंट आपके संदर्भ के लिए यहां सूचीबद्ध हैं। यहां स्क्रिप्ट उस घटना के विपरीत क्रियान्वित होने के लिए एक जावास्क्रिप्ट फंक्शन को इंगित करती है।

Attribute गुण	Value	Description विवरण
offline ऑफलाइन	script	Triggers when the document goes offline. दस्तावेज ऑफलाइन होने पर ट्रिगर।
onabort ऑनअबॉर्ट	script	Triggers on an abort event. एक अबॉर्ट की घटना पर ट्रिगर।
onafterprint ऑनऑफ्टरप्रिंट	script	Triggers after the document is printed. डॉक्यूमेंट प्रिंटेड होने के बाद ट्रिगर।
onbeforeunload ऑनविफोरऑनलोड	script	Triggers before the document loads. दस्तावेज लोड होने से पहले ट्रिगर।
onbeforeprint ऑनविफोरप्रिंट	script	Triggers before the document is printed. दस्तावेज प्रिंट होने से पहले ट्रिगर।
onblur ऑनब्लर	script	Triggers when the window loses focus. जब विंडो फोकस करता है तो ट्रिगर
oncanplay ऑनकैनप्ले	script	Triggers when media can start play, but might has to stop for buffering. ट्रिगर जब मीडिया प्ले शुरू कर सकता है, लेकिन बफरिंग के लिए रुकना पड़ सकता है।
oncanplaythrough ऑनकैनप्लेथ्रू	script	Triggers when media can be played to the end, without stopping for buffering. ट्रिगर जब मीडिया अंत तक प्ले किया जा सकता है, बफरिंग के लिए बिना रुके।
onchange ऑनचेंज	script	Triggers when an element changes. जब कोई एलीमेंट बदलता है तो ट्रिगर।
onclick ऑनक्लिक	script	Triggers on a mouse click. एक माउस क्लिक पर ट्रिगर।
oncontextmenu ऑनकंटेक्स्टमेन्यू	script	Triggers when a context menu is triggered. एक कंटेक्स्ट मेनू ट्रिगर होने पर ट्रिगर।
ondblclick ऑनडबलक्लिक	script	Triggers on a mouse double-click माउस डबल-क्लिक पर ट्रिगर।
ondrag	script	Triggers when an element is dragged.

ऑनड्रैग		किसी तत्व को खींचने पर ट्रिगर।
ondragend ऑनड्रैगइंड	script	Triggers at the end of a drag operation. एक ड्रैग ऑपरेशन के अंत में ट्रिगर।
ondragenter ऑनड्रैगइंटर	script	Triggers when an element has been dragged to a valid drop target. जब कोई तत्व किसी वैध ड्रॉप लक्ष्य पर खींचा जाता है तो ट्रिगर।
ondragleave ऑनड्रैगलीव	script	Triggers when an element is being dragged over a valid drop target. जब कोई तत्व वैध ड्रॉप लक्ष्य पर खींचा जा रहा है तो ट्रिगर।
ondragover ऑनड्रैगओवर	script	Triggers at the start of a drag operation. ड्रैग ऑपरेशन की शुरुआत में ट्रिगर।
ondragstart ऑनड्रैगस्टार्ट	script	Triggers at the start of a drag operation. ड्रैग ऑपरेशन की शुरुआत में ट्रिगर।
ondrop ऑनड्रॉप	script	Triggers when dragged element is being dropped. ड्रैग एलिमेंट को जब ड्रॉप करेंगे तो ट्रिगर।
ondurationchange ऑनड्यूरेशनचेंज	script	Triggers when the length of the media is changed. मीडिया की लंबाई बदलने पर ट्रिगर।
onemptied ऑनइम्पटीड	script	Triggers when a media resource element suddenly becomes empty. ट्रिगर जब एक मीडिया रिसोर्स एलिमेंट अचानक खाली हो जाता है।
onended ऑनइंडेड	script	Triggers when media has reach the end. ट्रिगर जब मीडिया अंत तक पहुँच गया है।
onerror ऑनएरर	script	Triggers when an error occur. एरर होने पर ट्रिगर।
onfocus परफोकस	script	Triggers when the window gets focus. विंडो फोकस होने पर ट्रिगर।
onformchange ऑनफॉर्मचेंज	script	Triggers when a form changes जब कोई लेता बदलता है तो ट्रिगर।
onforminput ऑनफॉर्मइनपुट	script	Triggers when a form gets user input. ट्रिगर जब एक फार्म उपयोगकर्ता इनपुट लेता है।
onhaschange ऑनहैसचेंज	script	Triggers when the document has change. जब दस्तावेज बदल गया है तो ट्रिगर।
oninput ऑनइनपुट	script	Triggers when an element gets user input. जब किसी एलिमेंट को उपयोगकर्ता का इनपुट मिलता है तो ट्रिगर।
oninvalid ऑनइनवैलिड	script	Triggers when an element is invalid. किसी एलिमेंट के अमान्य होने पर ट्रिगर।
onkeydown ऑनकीडाउन	script	Triggers when a key is pressed. ट्रिगर जब एक की दबाया जाता है।
onkeypress ऑनकीप्रेस	script	Triggers when a key is pressed and released. ट्रिगर जब एक कुंजी दबाया और जारी किया जाता है।
onkeyup ऑनकीअप	script	Triggers when a key is released. एक कुंजी जारी होने पर ट्रिगर।
onload ऑनलोड	script	Triggers when the document loads. दस्तावेज लोड होने पर ट्रिगर।
onloadeddata ऑनलोडेडडेटा	script	Triggers when media data is loaded. मीडिया डेटा लोड होने पर ट्रिगर।
onloadedmetadata ऑनलोडेडमेटाडेटा	script	Triggers when the duration and other media data of a media element is loaded. ट्रिगर जब एक मीडिया एलिमेंट की अवधि और अन्य मीडिया डेटा लोड किया जाता है।
onloadstart	script	Triggers when the browser starts to load the media data.

ऑनलोडस्टार्ट		ट्रिगर जब ब्राउजर मीडिया डेटा लोड करना शुरू करता है।
onmessage ऑनमैसेज	script	Triggers when the message is triggered. मैसेज ट्रिगर होने पर ट्रिगर।
onmousedown ऑनमाउसडाउन	script	Triggers when a mouse button is pressed. ट्रिगर जब एक माउस बटन दबाया जाता है।
onmousemove ऑनमाउसमूव	script	Triggers when the mouse pointer moves. जब माउस पॉइंटर चलता है तो ट्रिगर।
onmouseout ऑनमाउसआउट	script	Triggers when the mouse pointer moves out of an element. ट्रिगर जब माउस पॉइंटर एक एलीमेंट से बाहर निकलता है।
onmouseover ऑनमाउसओवर	script	Triggers when the mouse pointer moves over an element. जब माउस पॉइंटर किसी एलीमेंट पर चलता है तो ट्रिगर।
onmouseup ऑनमाउसअप	script	Triggers when a mouse button is released. माउस बटन छोड़े जाने पर ट्रिगर।
onmousewheel ऑनमाउसव्हील	script	Triggers when the mouse wheel is being rotated. ट्रिगर जब माउस व्हील घुमाया जाता है।
onoffline ऑनऑफलाइन	script	Triggers when the document goes offline. दस्तावेज ऑफलाइन होने पर ट्रिगर।
ononline ऑनऑनलाइन	script	Triggers when the document comes online. दस्तावेज ऑनलाइन आने पर ट्रिगर।
onpagehide ऑनपेजहाईड	script	Triggers when the window is hidden. जब विन्डो हिडेन हो तो ट्रिगर।
onpageshow ऑनपेजशो	script	Triggers when the window becomes visible. विन्डो दिखाई देने पर ट्रिगर।
onpause ऑनपॉज	script	Triggers when media data is paused. मीडिया डेटा रोक दिए जाने पर ट्रिगर।
onplay ऑनप्ले	script	Triggers when media data is going to start playing. ट्रिगर जब मीडिया डेटा प्ले करना शुरू कर रहा है।
onplaying ऑनप्लेइंग	script	Triggers when media data has start playing. ट्रिगर जब मीडिया डेटा प्ले शुरू हो गया है।
onpopstate ऑनपॉपस्टेट	script	Triggers when the window's history changes. जब विन्डो का हिस्ट्री बदलता है तो ट्रिगर।
onprogress ऑनप्रोग्रेस	script	Triggers when the browser is fetching the media data. ट्रिगर जब ब्राउजर मीडिया डेटा प्राप्त कर रहा है।
onratechange ऑनरेटचेंज	script	Triggers when the media data's playing rate has changed. ट्रिगर जब मीडिया डेटा की प्लेइंग रेट बदल गई हो।
onreadystatechange ऑनरेडीस्टेटचेंज	script	Triggers when the ready-state changes. जब तैयार स्थिति बदल जाती है तो ट्रिगर।
onredo ऑनरीडू	script	Triggers when the document performs a redo. जब डॉक्यूमेंट एक नया प्रदर्शन करता है तो ट्रिगर।
onresize ऑनरीसाइज	script	Triggers when the window is resized. विन्डो के आकार बदलने पर ट्रिगर।
onscroll ऑनस्कॉल	script	Triggers when an element's scrollbar is being scrolled. जब किसी तत्व के स्कॉलबार को स्कॉल किया जा रहा हो तो ट्रिगर।
onseeked ऑनसीकड	script	Triggers when a media element's seeking attribute is no longer true, and the seeking has ended. ट्रिगर जब एक मीडिया एलीमेंट की मांग विशेषता अब सच नहीं है, और मांग समाप्त हो गई है।
onseeking ऑनसीकिंग	script	Triggers when a media element's seeking attribute is true, and the seeking has begun. ट्रिगर जब एक मीडिया एलीमेंट की मांग विशेषता सच है, और मांग शुरू हो गई

		है।
onselect ऑनसेलेक्ट	script	Triggers when an element is selected. जब किसी एलीमेंट का चयन किया जाता है तो ट्रिगर।
onstalled ऑनस्टाल्ड	script	Triggers when there is an error in fetching media data. मीडिया डेटा प्राप्त करने में त्रुटि होने पर ट्रिगर।
onstorage ऑनस्टोरेज	script	Triggers when a document loads. डॉक्यूमेंट लोड होने पर ट्रिगर।
onsubmit ऑनसबमिट	script	Triggers when a form is submitted. फॉर्म सबमिट करने पर ट्रिगर।
onsuspend ऑनसस्पेंड	script	Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched. ट्रिगर जब ब्राउजर मीडिया डेटा प्राप्त कर रहा है, लेकिन पूरी मीडिया फाइल को लाने से पहले रोक दिया गया था।
ontimeupdate ऑनटाईपअपडेट	script	Triggers when media changes its playing position. जब मीडिया अपनी प्लेइंग पोजीशन बदलता है तो ट्रिगर करता है।
onundo ऑनअन्डू	script	Triggers when a document performs an undo. जब कोई दस्तावेज पूर्ववत करता है तो ट्रिगर।
onunload ऑनअनलोड	script	Triggers when the user leaves the document. उपयोगकर्ता द्वारा दस्तावेज छोड़ने पर ट्रिगर।
onvolumechange ऑनवाल्यूमचेंज	script	Triggers when media changes the volume, also when volume is set to "mute". जब मीडिया वॉल्यूम बदलता है, तब भी जब वॉल्यूम "म्यूट" पर सेट होता है।
onwaiting ऑनवेटिंग	script	Triggers when media has stopped playing, but is expected to resume. ट्रिगर जब मीडिया ने प्ले बंद कर दिया है, लेकिन फिर से शुरू होने की उम्मीद है।

6.7 Basic Form Validations in JavaScript- जावा स्क्रिप्ट में बेसिक फॉर्म वैलिडेशन

A form is also known as web form or HTML form and are used in web-pages. In form the user will enter his required details that are further sending it to the server for processing. JavaScript Form Validation is a method to validate form values enter by user.

एक फॉर्म को वेब फॉर्म या HTML फॉर्म के रूप में भी जाना जाता है। उपयोगकर्ता द्वारा आवश्यक विवरण दर्ज करने के लिए प्रपत्रों का उपयोग वेब-पेजों में किया जाता है जो आगे इसे प्रोसेसिंग के लिए सर्वर पर प्रेषित करते हैं। जावास्क्रिप्ट फॉर्म सत्यापन उपयोगकर्ता द्वारा दर्ज किए गए फॉर्म वैल्यू को मान्य करने की एक विधि है।

There are various ways to do validation - वैलिडेशन करने के लिए विभिन्न तरीके हैं-

- **Server side validation** is performing by web server, after input data is sent to server.
इनपुट डेटा सर्वर पर सेंड करने पर सर्वर साइड वैलिडेशन वेब सर्वर के माध्यम से परफार्म होता है।
- **Client side validation** is performing by web browser, before input data is sent to server.
इनपुट डेटा सर्वर पर सेंड करने से पहले क्लाइंट साइड वैलिडेशन वेब ब्राउजर के माध्यम से परफार्म होता है।

As you know that the form validation normally occurs at the server. In the form, after entering all the necessary data the client presses the submit button to validate it at the server. When the data entered by the client was incorrect or missing, then the server would have to send all the data back to the client and request that the form is to be resubmitted with correct information. JavaScript provides a way to validate the form data on the client's computer before sending it to the web server. There are two functions that perform in the form validation.

क्लाइंट पर सभी आवश्यक डेटा दर्ज करने और फिर सबमिट बटन दबाए जाने के बाद सामान्य रूप से सर्वर पर फॉर्म डेटा वापस भेज देता है। यदि क्लाइंट द्वारा दर्ज किया गया डेटा गलत होता है या बस गायब होता है, तो सर्वर क्लाइंट को सभी में एक लंबी प्रक्रिया होता है जो सर्वर पर बहुत अधिक बोझ डालती है। जावास्क्रिप्ट वेब सर्वर पर भेजने से पहले क्लाइंट के कंप्यूटर पर फॉर्म के डेटा को मान्य करने का एक तरीका प्रदान करता है। फॉर्म वैलिडेशन आम तौर पर दो कार्य करता है।

1. Basic Validation- बेसिक वैलिडेशन- You have to make sure that all the mandatory fields are filled in. It would require just a loop through each field in the form and check it for data. यह सुनिश्चित करें कि सभी अनिवार्य फ़िल्ड भरे गए हैं। इसके लिए फॉर्म में प्रत्येक फ़िल्ड के माध्यम से एक लूप की आवश्यकता होती है और डेटा की भी जाँच किया जाता है।

2. Data Format Validation - डेटा फॉर्मेट वैलिडेशन- In this process you have to make sure that user input is clean, safe and useful. The validation includes that the user fill up all the require fields, इस प्रक्रिया में उपयोगकर्ता यह सुनिश्चित कर लें कि इनपुट क्लीन, सुरक्षित और उपयोगी है एवं सभी आवश्यक फ़िल्ड को भर दिया गया हो, दिनांक वैल्यू प्रोपर फॉर्मेट में दर्ज करें एवं उचित प्रारूप में संख्यात्मक फ़िल्ड में उचित संख्यात्मक वैल्यू भरा गया हो एवं कुछ अन्य।

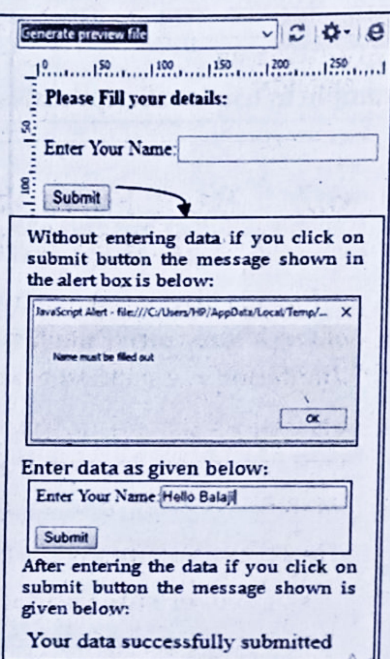
With the example given below you can understand the process of validation.

नीचे दिए गए उदाहरण द्वारा आप वैलिडेशन के प्रोसेस को आसानी से समझ सकते हैं-

```

1 <html>
2   <head>
3     <script type = "text/javascript">
4       function callFun()
5         {
6           var x = document.forms["myForm"]["fname"].value;
7           if(x=="")
8             {
9               alert("Name must be filled out");
10              return false;
11            }
12           else
13             document.write("Your data successfully submitted");
14         }
15     </script>
16   </head>
17   <body>
18     <h4>Please Fill your details:</h4>
19     <form name="myForm" action="/actionPage.php" onsubmit = "callFun()" method="post">
20       Enter Your Name:<input name="fname"><br><br>
21       <input type = "submit" value="Submit"/>
22     </form>
23   </body>
24 </html>

```



Without entering data if you click on submit button the message shown in the alert box is below:

Enter data as given below:

After entering the data if you click on submit button the message shown is given below:

Your data successfully submitted

Validating Numeric Input: संख्यात्मक इनपुट को वैलिडेट करना:

JavaScript is often used to validate numeric input. The client side validations become very important when submitting any data on web page. As we know that, JavaScript is a medium to validate the user to entered values, we need a function that will check for the entered values in a Numeric Only Field.

संख्यात्मक इनपुट को वैलिडेट करने के लिए जावास्क्रिप्ट का उपयोग किया जाता है। क्लाइंट साइड वैलिडेशन काफी इम्पोर्टेंट हो जाता है जब हम कोई डेटा वेब पर सबमिट करते हैं। जैसा की हम जानते हैं की जावास्क्रिप्ट एक मीडियम है जिससे हम यूजर को केवल वैल्यू एंटर करने के लिए रेस्ट्रिक्ट कर देते हैं ताकि यूजर द्वारा कोई टेक्स्ट एक्सपैक्ट न किया

जाए। इसके लिए हमें एक फंक्शन की जरूरत होती है जो यह चेक करे कि एंटेर्ड वैल्यू न्यूमेरिक है जो न्यूमेरिक ओन्ली फील्ड में है।

In JavaScript, there are two ways to check if a variable is a number: जावास्क्रिप्ट में, वैरियेबल को नंबर चेक करने के दो प्रकार हैं जो निम्नलिखित हैं—

1. **isNaN()**- इजनैन()- It stands for "is Not a Number", if a variable is not a number then, it return true, else return false. यह चेक करता है कि क्या वैरियेबल नंबर है या नहीं; यदि वैरियेबल नंबर नहीं होता है तो यह ट्रू रिटर्न करेगा वरना फाल्स रिटर्न करेगा।
2. **typeof**- टाइपऑफ - "typeof" is an operator in JavaScript and it is used to return the data type of a variable, value, expression i.e. operand. The **typeof** operator returns the string indicating a type of the unevaluated operand. The operand can be any object, function or variable. If variable is a number, it will return a string named "number".
टाइपऑफ एक जावास्क्रिप्ट ऑपरेटर है जो वैरियेबल, वैल्यू, एक्सप्रेशन, अर्थात् आपरेंड का डाटा टाइप रिटर्न करता है। टाइपऑफ ऑपरेटर स्ट्रिंग रिटर्न करता है जो एक अनइवैल्यूएटेड आपरेंड के टाइप को इंडिकेट करता है। आपरेंड कोई आब्जेक्ट, फंक्शन या कोई वैरियेबल हो सकता है। यदि वैरियेबल कोई नंबर होता है तो यह नंबर नाम का एक स्ट्रिंग रिटर्न करेगा।

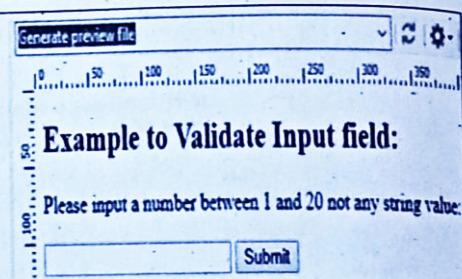
Note: Normally, people use isNaN() function to check number, but typeof is a nice try also. नार्मली, नंबर चेक करने के लिए यूजर इजनैन() फंक्शन का प्रयोग करते हैं, लेकिन टाइपऑफ भी अच्छा है यूज करने के लिए।

Example to use JavaScript's isNaN() function: जावास्क्रिप्ट इजनैन() फंक्शन का एक उदाहरण:

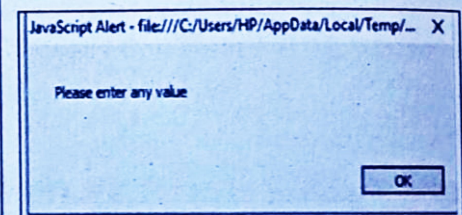
```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h2>Example to Validate Input field:</h2>
5 <p>Please input a number between 1 and 20 not any string value:</p>
6
7 <input id="txt">
8 <button type="button" onclick="myFun()">Submit</button>
9 </p id="text"></p>
10
11 <script>
12 function myFun() {
13     var x, msg;
14     // Get the value of the input field with id="txt"
15     x = document.getElementById("txt").value;
16     if(x=="")
17     {
18         alert("Please enter any value");
19         return false;
20     }
21     // If x is Not a Number or Less than one or greater than 20
22     else if (isNaN(x) || x < 1 || x > 20)
23         msg = "Input not valid";
24     else
25         msg = "Input OK";
26     document.getElementById("text").innerHTML = msg;
27 }</script>
28 </body></html>

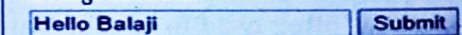
```



If you click submit button without entering any data the message shown is given below:

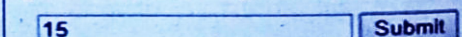


If you enter any text in the textbox the message shown is:



Input not valid

If you enter value according to the condition given; then, the message shown is below:



Input OK