

Events >>

Events are actions or occurrences that happen in the system you are programming — the system will fire a signal of some kind when an event occurs, and also provide a mechanism by which some kind of action can be automatically taken (e.g. some code running) when the event occurs.

Registering Event Handlers >>

- Event Handler Properties

<pre><button> Change color </button></pre>	<pre><script> var btn = document.querySelector('button'); function random(number) { return Math.floor(Math.random() * (number+1)); } btn.onclick = function() { var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')'; document.body.style.backgroundColor = rndCol; } </script></pre>
	<pre>var btn = document.querySelector('button'); function bgChange() { var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')'; document.body.style.backgroundColor = rndCol; } btn.onclick = bgChange;</pre>

- Inline Event Handler(**Bad Practice**):

<pre><button onclick="bgChange()"> Press me </button></pre>	<pre>function bgChange() { var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')'; document.body.style.backgroundColor = rndCol; }</pre>
---	---

- addEventListener()** and **removeEventListener()** methods:

<pre>var btn = document.querySelector('button'); function bgChange() { var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')'; document.body.style.backgroundColor = rndCol; } btn.addEventListener('click', bgChange);</pre>
--

advantages:

- You can remove event handler code if needed.
- You can add multiple listeners of the same type to elements if required.

Event Object >>

Sometimes inside an event handler function, you might see a parameter specified with a name such as **event**, **evt**, or simply **e**. This is called the **event object**, and it is automatically passed to event handlers to provide extra features and information.

Properties	Methods
type – returns the type/name of the event	preventDefault() – cancels the event(if it is cancelable) stopPropagation() – stops the propagation of events further along in the DOM
target – returns the element that triggers the event	Example <pre> var form = document.querySelector('form'); var fname = document.getElementById('fname'); var lname = document.getElementById('lname'); var submit = document.getElementById('submit'); var para = document.querySelector('p'); form.onsubmit = function(e) { if (fname.value === '' lname.value === '') { e.preventDefault(); para.textContent = 'You need to fill in both names!'; } }</pre>

Major Events List >>

Event Name	Description	Sample Code
load	A resource and its dependent resources have finished loading.	<pre> <html> <body onload="myFunction()" > <h1>Hello World!</h1> <script> function myFunction() { alert("Page is loaded"); } </script> </body> </html></pre>
focus	The focus event fires when an element has received focus.	<pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>JavaScript Events</title> </head> <body> <form id="form"> <input type="text" placeholder="text input"> <input type="password" placeholder="password"> </form> <script type="text/javascript"> const password = document.querySelector('input[type="password"]');</pre>
blur	An element has lost focus	

		<pre> password.addEventListener('focus', function (event) { event.target.style.background = 'pink'; }); password.addEventListener('blur', (event) => { event.target.style.background = ''; }); </script> </body> </html> </pre>
submit	The submit button is pressed	<pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>JavaScript Events</title> </head> <body> <form id="form"> <label>Test field: <input type="text"></label>

 <button type="submit">Submit form</button> <button type="reset">Reset form</button> </form> <p id="log"></p> <script type="text/javascript"> function logSubmit(event) { log.textContent = `Form Submitted! Time stamp: \${event.timeStamp}`; event.preventDefault(); } function logReset(event) { log.textContent = `Form reset! Time stamp: \${event.timeStamp}`; } const form = document.getElementById('form'); const log = document.getElementById('log'); form.addEventListener('submit', logSubmit); form.addEventListener('reset', logReset); </script> </body> </html> </pre>
reset	The reset button is pressed	
keydown	ANY key is pressed	<pre> <!DOCTYPE html> <html> <body> <p>A function is triggered when the user releases a key in the input field. The function transforms the character to upper case.</p> Enter your name: <input type="text" id="fname" onkeyup="myFunction()"> <script> function myFunction() { var x = document.getElementById("fname"); x.value = x.value.toUpperCase(); } </script> </pre>
keyup	ANY key is released	

		<pre> </body> </html> </pre>
click	A pointing device button (ANY button; soon to be primary button only) has been pressed and released on an element.	<pre> <!DOCTYPE html> <html> <body> <h1>The onclick Event</h1> </pre>
dblclick	A pointing device button is clicked twice on an element.	<pre> <p>The onclick event is used to trigger a function when an element is clicked on.</p> <p>Click the button to trigger a function that will output "Hello World" in a p element with id="demo".</p> <button onclick="myFunction()">Click me</button> <p id="demo"></p> <script> function myFunction() { document.getElementById("demo").innerHTML = "Hello World"; } </script> </body> </html> </pre>
mousedown	A pointing device button is pressed on an element.	<pre> <!DOCTYPE html> <html> <body> <div style="text-align:center;" id="notification"></div> <div style="margin:20px;border:1px solid blue;padding:20px;height:300px;width:auto;background- color:#ffe6e6;" onmousedown="mouseeventfn(event);" onmouseup="mouseeventfn(event);" onmouseover="mouseeventfn(event);" onmouseout="mouseeventfn(event);" onmouseenter="mouseeventfn(event);" onmouseleave="mouseeventfn(event);"> <div style="margin:50px;border:1px solid black;padding:50px;background-color:#e6ffee;"> Child </div> </div> <script> function mouseeventfn(event) { console.log(event); document.querySelectorAll("#notification")[0].innerHTML="Event Type: "+event.type+" Screen X and Y: "+event.screenX+" "+event.screenY+" Window X and Y: "+event.clientX+" "+event.clientY+" Document X and Y: </pre>
mouseup	A pointing device button is released over an element.	
mouseenter	A pointing device is moved onto the element that has the listener attached.	
mouseleave	A pointing device is moved off the element that has the listener attached.	
mousemove	A pointing device is moved over an element. (Fired continuously as the mouse moves.)	

		<pre> "+event.pageX+", "+event.pageY+" Offset X and Y: "+event.offsetX+", "+event.offsetY; } </script> </body> </html> </pre>
change	<p>The change event is fired for <input>, <select>, and <textarea> elements when an alteration to the element's value is committed by the user.</p>	<pre> <!DOCTYPE html> <html> <body> <p>Select a new car from the list.</p> <select id="mySelect" onchange="myFunction()" > <option value="Audi">Audi</option> <option value="BMW">BMW</option> <option value="Mercedes">Mercedes</option> <option value="Volvo">Volvo</option> </select> <p>When you select a new car, a function is triggered which outputs the value of the selected car.</p> <p id="demo"></p> <script> function myFunction() { var x = document.getElementById("mySelect").value; document.getElementById("demo").innerHTML = "You selected: " + x; } </script> </body> </html> </pre>

References:

1. <https://developer.mozilla.org/en-US/docs/Web/Events>
2. https://www.w3schools.com/jsref/dom_obj_event.asp