

Introduction to JavaScript

Internet Programming I: Chapter 4 – Part II



ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY
Department of Software Engineering

Main Source: <https://www.w3schools.com/js/default.asp>

JavaScript Events



- The change in the state of an object is known as an Event
- In html, there are various events which represents that some activity is performed by the user or by the browser
- When javascript(js) code is included in HTML, js react over these events and allow the execution
- This process of reacting over the events is called Event Handling
 - Thus, js handles the HTML events via Event Handlers.
- For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event

JavaScript Events mouse events



Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

JavaScript Events cont'd



Keyboard events

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

JavaScript Events

Window/Document events



Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

JavaScript Events example



Example Click Event

```
<html>
<head> Javascript Events </head>
<body>
<script language="Javascript" type="text/Javascript">
    function myFunc()
    {
        document.write("This is JavaTpoint");
    }
</script>
<form>
<input type="button" onclick="myFunc()"
value="Who's this?"/>
</form>
</body>
</html>
```

Example MouseOver Event

```
<html>
<head>
<h1> Javascript Events </h1>
</head>
<body>
<script language="Javascript" type="text/Javascript">
    function myFunc()
    {
        alert("This is JavaTpoint");
    }
</script>
<p onmouseover="myFunc()"> Keep cursor over me</p>
</body>
</html>
```

JavaScript Events example



Keydown Event

```
<html>
<head> Javascript Events</head>
<body>
<h2> Enter something here</h2>
<input type="text" id="input1"
onkeydown="keydownevent()"/>
<script>
  function keydownevent()
  {
    document.getElementById("input1");
    alert("Pressed a key");
  }
</script>
</body>
</html>
```

Load event

```
<html>
<head> Javascript Events</head>
<br>
<body onload="window.alert('Page suc
cessfully loaded');">
<script>
  document.write("The page is loaded su
ccessfully");
</script>
</body>
</html>
```

JavaScript addEventListener()



- The `addEventListener()` method is used to attach an event handler to a particular element
- An event listener is a JavaScript's procedure that waits for the occurrence of an event
- Syntax
 - `element.addEventListener(event, function, useCapture);`
- **event:** It is a required parameter. It can be defined as a string that specifies the event's name
 - Note: Do not use any prefix such as "on" with the parameter value. For example, Use "click" instead of using "onclick"
- **function:** It is also a required parameter. It is a JavaScript function which responds to the event occurred
- **useCapture:** It is an optional parameter. It is a Boolean type value that specifies whether the event is executed in the bubbling or capturing phase
 - When it is set to true, the event handler executes in the capturing phase
 - When it is set to false, the handler executes in the bubbling phase. Its default value is false
- **Event Capturing:** an event moves from the outermost element to the target
- **Event bubbling:** the event movement begins from the target to the outermost element in the file

JavaScript addEventListener() example

Simple Example

```
<!DOCTYPE html>
<html>
<body>
<p> Example of the addEventListener() method. </p>
<p> Click the following button to see the effect. </p>
<button id = "btn"> Click me </button>
<p id = "para"></p>
<script>
document.getElementById("btn").addEventListener("click", fun);
function fun() {
document.getElementById("para").innerHTML = "Hello World" + "<br>" + "Welcome to the javaTpoint.com";
}
</script>
</body>
</html>
```

Example: multiple events

```
...<body>
<p> This is an example of adding multiple events to the same element. </p>
<p> Click the following button to see the effect. </p>
<button id = "btn"> Click me </button>
<p id = "para"></p>
<p id = "para1"></p>
<script>
function fun() { alert("Welcome to the javaTpoint.com"); }
function fun1() {
    document.getElementById("para").innerHTML = "This is second function"; }
function fun2() {
    document.getElementById("para1").innerHTML = "This is third function"; }
var mybtn = document.getElementById("btn");
mybtn.addEventListener("click", fun);
mybtn.addEventListener("click", fun1);
mybtn.addEventListener("click", fun2);
</script>
</body>
</html>
```

Event Bubbling or Event Capturing example

```
<head>
<style>
div{ background-color: lightblue; border: 2px solid red; font-size: 25px; text-align: center; }
span{ border: 2px solid blue; }
</style>
</head>
<body>
<h1> Bubbling </h1>
<div id = "d1"> This is a div element.<br><br> <span id = "s1"> This is a span element. </span>
</div>
<h1> Capturing </h1>
<div id = "d2"> This is a div element. <br><br><span id = "s2"> This is a span element. </span>
</div>
<script>
document.getElementById("d1").addEventListener("dblclick", function() {alert('You have double clicked on div element')}, false);
document.getElementById("s1").addEventListener("dblclick", function() {alert('You have double clicked on span element')}, false);
document.getElementById("d2").addEventListener("dblclick", function() {alert('You have double clicked on div element')}, true);
document.getElementById("s2").addEventListener("dblclick", function() {alert('You have double clicked on span element')}, true);
</script>
</body>
</html>
```

Reading Assignment



- JavaScript HTML DOM Style Object
 - https://www.w3schools.com/jsref/dom_obj_style.asp

Exercise



- Form Validation
 - <https://www.javatpoint.com/javascript-form-validation>
- Email Validation
 - <https://www.javatpoint.com/javascript-form-validation#email>



