

Events

JavaScript



Table of Content

- JavaScript Events
- Event Flow
- Event Types



JavaScript Events

(JavaScript)

Events

- An event is some notable action to which a script can respond. It may be
 - Click
 - Mouseover
 - Keystroke etc.
- When a function is assigned to an event handler, that function is run when that event occurs.
- An Event handler is JavaScript code associated with a particular part of the document and a particular event.

Event Handler

- An Event handler is JavaScript code associated with a particular part of the document and a particular event.
- For example, an event handler associated with a button could open a new window when the button is clicked.
- A handler to the click event is called onclick.

Event Handler

- Events are not limited to basic user-actions associated with the document.
- Browser supports events such as resize, load, and unload.

Event Flow

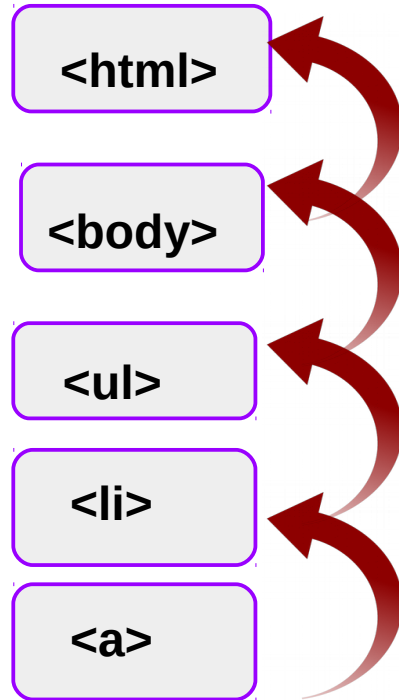
(JavaScript)

Event Flow

- The order in which events are received on the web page are described by event flow.
- An event has three phases :
 - Cycle
 - Target
 - Bubbling

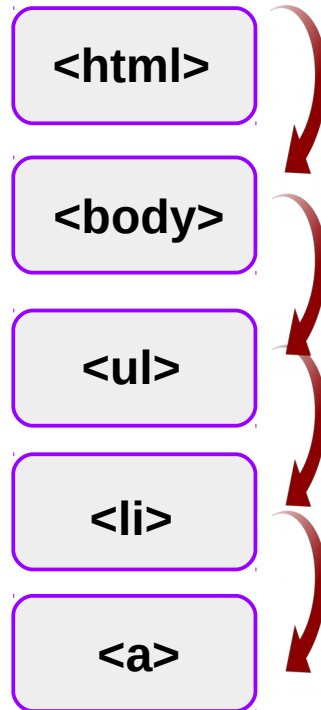
Event Bubbling

- Event Bubbling , the event is first captured and handled by innermost element and then propagated to outer element.



Event Capturing

- Event Capturing , the event is first captured by the outermost element and propagated to the inner elements.



Event Target

- Event Capturing, it provides an opportunity to intercept events if necessary.
- Then the actual target receives the event.
- Final phase is the bubbling , which allows a response to the event.

Event Listeners

The DOM 2 level 2 define two methods:

- `addEventListener`
- `removeEventListener`



Event Types

(JavaScript)

Mouse Events

Event Attribute	Description
onclick	Occurs when the mouse button is clicked
ondblclick	Occurs when the mouse button is double clicked
onmousedown	Occurs when the mouse button is pressed
onmouseup	Occurs when the mouse button is released
onmousemove	Occurs when mouse has moved while over an element.
onmouseover	Occurs when mouse has moved over an element.
onmouseout	Occurs when mouse has moved away from an element.



Keyboard Events

Event Attribute	Description
onkeypress	Occurs when a key pressed and released with focus on element
onkeydown	Occurs when a key pressed down
onkeyup	Releases a key



Loading Events

Event Attribute	Description
onload	Occurs when element has loaded
onunload	Indicates that browser is leaving the current document
onabort	Occurs when the user abort the loading of an image

Selection and Focus Event

Event Attribute	Description
onselect	Occurs after some text has been selected in an element
onchange	Occurs when text input has been changed
onfocus	Indicates that an element has received focus
onblur	Occurs when an element losses focus

Other Events

Event Attribute	Description
<code>onresize</code>	User resizes a window or a frame
<code>onsubmit</code>	Indicates form submission by clicking a submit button
<code>onreset</code>	Indicates that form is being reset by clicking reset button

Events Example

```
<body>

<button onclick="show()">Click Here</button>

<p id="ex"></p>

<script>

function show() {

    document.getElementById("ex").innerHTML = "Hello World";

}

</script>

</body>
```

Events Example

```
<!DOCTYPE html>

<html>

<body onLoad="alert('Welcome to my page!');"
      onUnload="alert('Goodbye! Sorry to see you go!');">



</body>

</html>
```

Events Example

```
<script>

    function OnMouseIn (elem) {
        elem.style.border = "4px solid green";
    }

    function OnMouseOut (elem) {
        elem.style.border = "";
    }

</script>
</head>
<body>
    <div style="background-color:#ddf0af; width:300px;color:#800000"
        onmouseover="OnMouseIn (this)" onmouseout="OnMouseOut (this)">
        Move your mouse pointer into and out of this element!
    </div>
</body>
```

Event Listeners

- The method `addEventListener()` is used to register a single event listener on the document.
- These methods exist on all DOM nodes. There is a slight change in event naming convention also, compared to how they are used with button elements (ex: `onclick` vs `click`)
- The event type to listen for (eg: `mouseout`, `click`, `error` etc)
- The event handler function to be executed when the event occurs.
- The third parameter is a boolean value specifying whether to use event bubbling or event capturing. This parameter is optional.
- The keyword `this` used with event handler represents a reference to the HTML element which fired the event handler.

Event Listeners

Syntax :

`element.addEventListener(event, function, useCapture);`

- We can add many event handlers to one element.
- We can add many event handlers of the same type to one element, i.e two "click" events.
- We can easily remove an event listener by using the `removeEventListener()` method.

Event Listeners Example

```
<html>
<body>

<p>EventListener works on the document. Click and check! </p>

<script>
document.addEventListener("click", function() {
    document.body.style.backgroundColor = "red";
});
</script>

</body>
</html>
```


Exercise

- Write a JavaScript program to create a paragraph and background color must change after some mouse events:
 - + onclick button → yellow
 - + odbleclick button → blue
 - + onmouseout → green
 - + onmouseover → red
- Write a JavaScript program to create a text field and show the effect of some events:
 - + onchange
 - + onfocus
 - + onblur



Web Stack Academy (P) Ltd

#83, Farah Towers,
1st floor, MG Road,
Bangalore - 560001

M: +91-80-4128 9576

T: +91-98862 69112

E: info@www.webstackacademy.com

*Thank
you*