Events & Event Object In JavaScript | JavaScript Tutorial In Hindi #17

In this tutorial, we discuss some important concepts about events and look how they work in browsers.

What are Events and Event Handlers ?

Events are actions that happen in the webpage like clicking a button or submit the form. If the user selects a button on a webpage, we might want to respond to that action by displaying an alert on the webpage. Each available event has an **event handler**, which is a block of code that runs when the event occurred. Event handlers are sometimes called **event listeners**. The listener listens out for the event happening, and the handler is the code that is run in response to it happening. There are many ways to assign a handler. Let us see them, starting from the simplest one.

HTML-attribute:-

We can set a handler in HTML with an attribute named on<event>. For example, to assign a click handler for an input, we can use onclick, here is an example:

```
<input value="Click here" onclick="alert('Click here!')" type="button">
```

The code inside onclick runs on mouse click. But note that the HTML-attribute is not a convenient place to write a lot of code, so it will better create a JavaScript function and call it there.

addEventListener():-

The JavaScript addEventListener() method allows the programmer to set up functions to be called when a specified event happens, such as when a user clicks a button. The purpose of using the addEventListener() method is to attach an event handler to the specified element. While using the addEventListener() method, remember that it attaches an event handler to an element without overwriting existing event handlers. We can add many event handlers to one element. Removing event is as simple as adding an event handler. This task can easily be done by using the removeEventListener() method. Following is the syntax to add an event listener.

Syntax:-

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

- The first parameter is the type of event like "click" or "mousedown".
- The second parameter is the function we want to call when the event occurs.
- The third parameter is optional. It is a boolean value specifying whether to use event bubbling or event capturing.

Example:-

Alert "Event Occurred" when the user clicks on an element:

```
document.addEventListener("click", function(){ alert("Event Occurred"); });
```

Similarly, we can also refer to an external "named" function:

```
document.addEventListener("click", myfunc);
 function myfunc () {
 document.getElementById("demo").innerHTML = "Hello World";
 }
```

Passing Event as a Parameter:-

Sometimes we may want to know more information about the event, such as what element was clicked. When an event happens, the browser creates an *event object*, puts details into it and passes it as an argument to the handler. Here is an example that shows event type, element and coordinates of the click.

```
document.addEventListener("click", myfunc);
function myfunc(event) {
alert(event.type + " at " + event.currentTarget);
alert("Coordinates: " + event.clientX + ":" + event.clientY);};
```

Some properties of the event object used in the above program are:

- **type:** This will tell the event type, here it is "click".
- **currentTarget:** It returns the element whose event listeners triggered the event.
- **clientX / event.clientY:** It returns the window-relative coordinates of the cursor, for pointer events.

There are many properties, and these depend on the event type. The keyboard events and pointer events have a different set of properties. Here are some properties of the event object

Event Properties and Methods:-

Property/Method	Description

altKey	It will return whether the "ALT" key was pressed when the mouse event was triggered.
button	It will return which mouse button was pressed when the mouse event was triggered.
charCode	It will return the Unicode character code of the key that triggered the onkeypress event.
clientX	It will return the horizontal coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered.
clientY	It will return the vertical coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered.
code	It will return the code of the key that triggered the event.
deltaX	It will return the horizontal scroll amount of a mouse wheel (x-axis).
deltaY	It will return the vertical scroll amount of a mouse wheel (y-axis).
deltaZ	It will return the scroll amount of a mouse wheel for the z-axis.
detail	It will return a number that indicates how many times the mouse was clicked.
keyCode	It will return the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event.
location	It will return the location of a key on the keyboard or device.
pageX	It will return the horizontal coordinate of the mouse pointer, relative to the document, when the mouse event was triggered.
pageY	It will return the vertical coordinate of the mouse pointer, relative to the document, when the mouse event was triggered.
screenX	It will return the horizontal coordinate of the mouse pointer, relative to the screen, when an event was triggered.
screenY	It will return the vertical coordinate of the mouse pointer, relative to the screen, when an event was triggered.
shiftKey	It will return whether the "SHIFT" key was pressed when an event was triggered.
type	It will return the name of the event.
which	It will return which mouse button was pressed when the mouse event was triggered.
view	It will return a reference to the Window object where the event occurred.

Website.html code as described/written in the video

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <div class="container">
        <h1 id="heading"> Welcome to Code With Harry</h1>
        <div id="myfirst" class="child red good" id="first">child 1

            <ul class="this">
                <li class="childul">this</li>
                <li class="childul">is</li>
                <li class="childul">a</li>
                <li class="childul">list </li>
                <li class="childul">of my dreams</li>
            </ul>
        </div>
        <div class="child">child 2</div>
        <div class="child red">child 3</div>
        <div class="child">child 4</div>
        <form action="none.html" method="post">
            <a href="//codewithharry.com">Go to Code With Harry</a>
            <br>
            <br>
            Search this website: <input type="text" name="Hello" id="">
            <input type="button" value="submit">
        </form>
    </div>
    <br>
    <div class="no">this is a dummy div1</div>
    <div class="no">this is a dummy div2</div>
    <div class="no">this is a dummy div3</div>
</body>
<!-- <script src="js/tut12.js"></script> -->
<!-- <script src="js/tut14.js"></script> -->
<script src="js/tut15.js"></script>
</html>
```

JavaScript code as described/written in the video

```
console.log("This is tutorial 17 on events");

document.getElementById("heading").addEventListener("click", function(e) {
  let variable;
  console.log("You have clicked the heading");
  variable = e.target;
  variable = e.target.className;
  variable = Array.from(e.target.classList);

  variable = e.target.id;
  variable = e.offsetX;
  variable = e.offsetY;
  variable = e.clientX;
  variable = e.clientY;
  console.log(variable);
  // location.href = '//codewithharry.com'
});
```

Previous Next

CodeWithHarry | Copyright © 2022 CodeWithHarry.com