

# JAVASCRIPT EVENTS

By: Sahil Bisht

# What is Event ?

An Event is a Signal Something has Happened.

The change in the state of an object is also known as an Event.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

# JavaScript addEventListener()

The **addEventListener()** method is used to attach an event handler to a particular element.

1. `<button id = "btn"> Click me </button>`
2. `<p id = "para"></p>`
3. `<script>`
4. `document.getElementById("btn").addEventListener("click", fun);`
5. `function fun() {`
6. `document.getElementById("para").innerHTML = "Hello World" ;`
7. `}`
8. `</script>`

**Syntax : `element.addEventListener(event, function, useCapture);`**

Although it has three parameters, the parameters ***event*** and ***function*** are widely used. The third parameter is optional to define.

**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. Its possible values are **true** and **false**. 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**

# Capturing and Bubbling

**Bubbling** and **Capturing** are the two ways of event propagation

The bubbling principle is simple.

**When an event happens on an element, it first runs the handlers on it, then on its parent, then all the way up on other ancestors.**

**While capturing is opposite of it .**

# LIST OF MOST USEFUL DOM EVENTS

**1: Mouse Events.**

**2: Keyboard Events.**

**3: Form Elements Events.**

**4: Css Events(when a css-animation finishes).**

# MOUSE EVENTS

- 1: click** – when the mouse clicks on an element (touchscreen devices generate it on a tap).
- 2: contextmenu** – when the mouse right-clicks on an element.
- 3: mouseover / mouseout** – when the mouse cursor comes over / leaves an element.
- 4: mousedown / mouseup** – when the mouse button is pressed / released over an element.
- 5: mousemove** – when the mouse is moved

# KEYBOARD EVENTS

**keydown**, **KeyPress** and **keyup** – when a keyboard key is pressed and released

**keypress** event generally doesn't work for the Backspace key (or Delete key), as they typically don't produce a character value.



# Form Elements Events

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements)

1.onblur

2.onreset

3.onselect

4.onsubmit

5.onfocus

6.oninput

# CSS Events

1: Transition Start / Transition End

2. Animations Start / Animations End