Javascript Events

Events are the techniques used by a user to interact with webpage, like **mouse click**, **mouse hover**, **key press**, **keyup**, **right click**, **drag touch** etc. Javascript can handle **Keyboard based events**, **mouse based events** and **Touch Based events**. A fully **interactive website** is not possible without **JS Events**.

Touch based events are also supported in javascript after ES5(2011).

---

click event in javascript using 3 methods.

Button

```
 <button onclick="alert(this.textContent)">Button </button>
    <!--Obtrusive JavaScript, not recommended-->

 <script>
 document.querySelector("button").onclick=function(){
 alert(this.textContent);
 }
 </script>
 <!--add event to element, can assign single function only-->

 <script>
 document.querySelector("button").addEventListener("click",function(){
alert(this.textContent) })

  </script>
<!--recommended way of dealing with Javascript Events-->
```

This first method id DOM level 0 event and works in all browsers, but isn't recommended, as we are writing javascript code in html. Second is DOM Level 2 events and is neat and clean, but is restricted to single function on same event. We recommended using **EventListener** in modern web.

List of Javascript Events

Here is a list of **mouse**, **keyboard** and **touch** based **events in javascript** with example.

← Swipe to view more →

Event	Interface		
	Keyboard	 Mouse	Touch
onclick			
oncontextmenu			
onfocus			
onblur			
oninput			
onchange			
onmouseover			
onmouseout			
onmousemove			
onmousedown			
onmouseup			
onkeypress			
onkeydown			
onkeyup			

Event	Interface		
	Keyboard	Mouse	Touch
onsubmit			
onreset			
onselect			
oncopy			
onpaste			
onscroll			
onload			
onreadystatechange			

onclick

Javascript click event is a global event for keyboard, mouse and touch interface. This event happen on mouse left click, tap on touch device, and keyboard click on focus (spacebar and enter key only). See example

Click

```
document.querySelector("button").onclick=function(){
    alert("clicked");
}
```

or

```
document.querySelector("button").addEventListener("click",function(){
    alert("clicked");
});
```

oncontextmenu

Javascript oncontextmenu is a global event for right click of mouse. This event can happen on pointing devices like mouse only. See example

Right Click

```
document.querySelector("button").addEventListener("contextmenu",function(){
    alert("right click");
})
```

To disable default right click menu, use `return false` in the end of function or e.preventDefault(). See example

Right Click

```
document.querySelector("button").oncontextmenu=function(e){
    alert("right click");
    return false;
}
```

Right Click

```
var btn=document.querySelector("button");

btn.addEventListener('contextmenu',function(e){
    e.preventDefault();
    alert("right click");
})
```

onfocus

Javascript onfocus is a global event for keyboard, mouse and touch interface. This event happen when user focus on form controls or hyperlink. See example

Name: Name

```
document.querySelector("input").addEventListener("focus",function(){
    this.value="";
})
```

onblur

Javascript onblur is a global event for keyboard, mouse and touch interface. This event happen when user loses focus from focusable element. See example

Name: Name

```
document.querySelector("input").addEventListener("focus",function(){
    this.value="";
```

```
});
document.querySelector("input").addEventListener("blur",function(){
    this.value="Name";
});
```

oninput

oninput event is global event of input, textarea, and select which invokes when user input data in following controls.

Enter No: 0

```
document.querySelector('input[type="range"]').addEventListener("input",functi
on(){
    document.querySelector('span').innerHTML=this.value;
})
```

Name: Enter Name

```
    document.querySelector("input").addEventListener("input",function(){
        document.querySelector('span').innerHTML=this.value;
    });
```

onchange

Javascript onchange is a global event for keyboard, mouse and touch interface. This event happen when user change value of select dropdown. See example

City: --Choose City--

```
document.querySelector("select").addEventListener("change",function(){
    console.log(this.value);
})
```

onmouseover

Javascript onmouseover is a global event for mouse. This event happen when user mouse over on an html element. See example

Mouse over event done

```
document.querySelector("div.box").addEventListener("mouseover",function(){
  this.innerHTML="Mouse over event done";
  this.style.backgroundColor="#ccc";
})
```

onmousemove

Javascript onmousemove is a global event for mouse and touch interface. This event happen when user move mouse over an element. This will happen every time user move his pointer. See example

Mouse over here

```
document.querySelector("div.box").addEventListener("mousemove",function(e){
   var x=e.clientX;
   var y=e.clientY;
console.log(`Cursor location is ${x} , ${y}`);
})
```

onmouseout

Javascript onmouseout is a global event for mouse and touch interface. This event happen when user remove mouse pointer from an element.. See example

mouse is out now

```
document.querySelector("div.box").addEventListener("mouseover",function(){
  this.innerHTML="mouse is in now";
});
document.querySelector("div.box").addEventListener("mouseout",function(){
  this.innerHTML="mouse is out now";
});
```

onmousedown

Javascript onmousedown is a global event for mouse. This event happen when user starts pressing mouse left click. Event will occur as soon as user initiate pressing mouse left key. See example

click here

```
document.querySelector("div.box").addEventListener("mousedown",function(){
  this.innerHTML="mouse left key is pressed";
});
```

onmouseup

Javascript onmouseup is a global event for mouse. This event happen when user remove mouse left key. It is happening every time user is releasing left key. See example

mouse is out now

```
document.querySelector("div.box").addEventListener("mouseup",function(){
  this.innerHTML="mouse left key is released";
});
```

onkeypress

Javascript onkeypress is a global event for keyboard and touch interface. This event happen when user press any keyboard key.

To check keycode, **e.which** is used where e is parameter in keypress function. To check which key is pressed, **String.fromCharCode(e.which)** is used. See example

Type:

```
document.querySelector("input").addEventListener("keypress",function(e){
     var x=e.which;    // keycode
     var y=String.fromCharCode(x);
  this.nextElementSibling.innerHTML=`you pressed ${y} key and keycode is
"${x}`;
})
```

Check Shift key

To check whether `Shift` key is press with keypress, use e.shiftKey.

Type:

```
document.querySelector("input").addEventListener("keypress",function(e){
     var x=e.which;    // keycode

     if( e.shiftKey){
        console.log("Shift key is pressed")
     }
     else{
        console.log("Shift key was not pressed")
     }
});
```

Check Alt key

To **check alt key** , use e.altKey.

Check Ctrl key

To **check Ctrl key** , use e.ctrlKey.

Check Cmd key in mac

To **check Cmd key** , use e.metaKey.

---

onkeydown

Javascript onkeydown is a global event for keyboard and touch interface. This event happen when user start pressing any keyboard key.

Javascript onkeydown event is used in gaming and canvas based applications. See example

Press any key

```
window.addEventListener("keydown","function(e){
    var x=e.which;
    document.querySelector('span').innerHTML=`keycode is ${x}.`;
})
```

---

onkeyup

Javascript onkeyup is a global event for keyboard and touch interface. This event happen when user release any keyboard key.

The **onkeyup event** is used to calculate no of characters in textbox or textarea. See example

Query: **160 characters left.**

```
document.querySelector("textarea").addEventListener("keyup","function(){
```

```
    var x=this.value.length;
    var y=160-x;
    this.nextElementSibling.innerHTML=`${y} characters left.`;
})
```

onsubmit

Javascript onsubmit is a global event for keyboard, mouse and touch interface. This event happen when user submit form data.

The **onsubmit event** act on click of submit button or enter key press. See example

Name: Submit

```
<form name="contact">
    <input type="text" name="username">
    <button>Submit</button>
</form>

<script>
document.contact.addEventListener("submit",function(e){
    var x=this.username.value;
   console.log(`thanks $x}`);

   e.preventDefault();
   // to avoid url redirection after submit
});
<script>
```

onreset

Javascript onreset is a global event for keyboard, mouse and touch interface. This event happen when user **reset form data** using reset button.

If **reset button** is pressed by mistake, all form data will be erased. To avoid this, use `window.confirm()` on **onreset event** . See example

Name: Reset Submit

```
<form name="contact1">
    <input type="text" name="username">
    <button type="reset">Submit</button>
    <button>Submit</button>
</form>

<script>
document.contact.addEventListener("reset",function(e){
    var x=window.confirm("would you like to reset?");

    if( x==false){e.preventDefault() }          // form reset disabled

})
<script>
```

onselect

Javascript onselect is a global event for keyboard, mouse and touch interface. This event happen when user select text in textbox or textarea.

The **onselect event** is used only when selection is done inside input and textarea element.

Select text:

```
document.querySelector("input").addEventListener("select","function(){
    this.nextElementSibling.innerHTML="You have selected text";
});
```

oncopy

Javascript oncopy is a global event for keyboard, mouse and touch interface. This event happen when user copy text from textbox or textarea.

The **oncopy event** is used only when selected text is copied.

Copy:

```
document.querySelector("input").addEventListener("copy",function(){
this.nextElementSibling.innerHTML="text copied";
})
```

onpaste

Javascript onpaste is a global event for keyboard, mouse and touch interface. This event happen when user paste text in textbox or textarea.

The **onpaste event** is used only when selected text is pasted after copying.

Paste:

```
document.querySelector("input").addEventListener("paste",function(){
this.nextElementSibling.innerHTML="text pasted";
});
```

onscroll

Javascript onscroll is a global event for keyboard, mouse and touch interface. This event happen when user scroll window using mouse, keyboard or touch interface.

The **onscroll event** is used in many scroll based effects like parallax scrolling, smooth scroll etc.

Scroll window you have scrolled 17959px from top

```
window.addEventListener("scroll",function(){
    console.log(this.scrollY);
})
```

onload

Javascript onload is a global event for keyboard, mouse and touch interface. This event happen when all requests in a webpage are loaded on DOM.

Onload event is helpful if we want toload DOM content first and then add javascript after DOM Elements are loaded .

```
    console.log("window loading");            // first

window.addEventListener("load",function(){
   console.log("window loaded");               // third
});

    console.log("window still loading");       // second
```

onreadystatechange

document.onreadystatechange events triggers when the readyState property of document changes. **readystatechange event** is not cancellable.

readyState

readyState or **document.readyState** has two values, **interactive** and **complete**. **interactive** means when page is ready for interactions, which happens when all html elements are parsed and complete means when even all media elements are loaded, like image, iframe, video etc.

Property	**Meaning**	**time** (in ms)
loading	document is loading	0
interactive	document is loaded and parses, but other resources like images, scripts, css , videos and frames are still awaited.	43
complete	document and other resources are loaded.	148

```
document.addEventListener("readystatechange",function(){
    if( this.readyState=="interactive"){
        // same as DOMContentLoaded event
    }
    else if(this.readyState=="complete"){
```

```
            // same as load event
        }
    });
```

Similar Events

1. **readystatechange interactive** event is same like **window.DOMContentLoaded** event
2. **readystatechange complete** event is same like **window.onload** event.