

# JavaScript - Window/Document Events

JavaScript window events are actions that occur when the user does something affecting the entire browser window, like loading, resizing, closing, or moving the window. The most common window event is simply loading the window by opening a particular web page. This event is handled by the onload event handler.

Developers can use JavaScript to create dynamic, interactive web pages that respond to user actions. The interactivity depends on two core aspects: window events and document events. Operating at the browser's panoramic level, window events bestow control over the overall state of the browser window; alternatively, document events interact with the HTML document, empowering developers to react specifically towards elements or actions within a page

## Window Events

At the browser level, window events happen and hold association with the window object; this global object represents the web browser's window. Frequently employed to oversee the overall state of a browser window or manage global interactions are these types of events.

Event Name	Description
load	Triggered when the entire web page, including all its resources, has finished loading.
unload	Fired when the user is leaving the page or closing the browser window or tab.
resize	Activated when the size of the browser window is changed.
scroll	Fired when the user scrolls the page.

## Example: Demonstrating Window Events

In this example, a script actively listens for the 'load,' 'resize,' and 'scroll' events on the window; it includes an initial page load alert to inform users that loading is complete. Should they subsequently resize their window, an alert will trigger thereby displaying the new size of their updated viewport. Moreover, when the user scrolls on the page, an alert is triggered to indicate their action.

[Open Compiler](#)

```
<!DOCTYPE html>
<html>
<head>
  <title>Window Events Example</title>
  <style>
    body {
      height: 2000px; /* Adding some content just to enable scrolling */
    }

    #resizeInfo {
      position: fixed;
      top: 10px;
      left: 10px;
      background-color: #fff;
      padding: 10px;
      border: 1px solid #ccc;
    }
  </style>
  <script>
    window.addEventListener('load', function() {
      var initialSizeInfo = 'Initial window size: ' + window.innerWidth + ' :
      document.getElementById('resizeInfo').innerText = initialSizeInfo;

      alert('The page has finished loading!');
    });

    window.addEventListener('resize', function() {
      var newSizeInfo = 'New window size: ' + window.innerWidth + ' x ' + wi
      document.getElementById('resizeInfo').innerText = newSizeInfo;
      alert("Page has been resized");
    });

    window.addEventListener('scroll', function() {
      alert('You have scrolled on this page.');
    }, {once: true});
  </script>
</head>
<body>
  <div id="resizeInfo">Initial window size: ${window.innerWidth} x ${window.i
</body>
</html>
```

## Document Events

Document events on the other hand occur at the level of the HTML document within the window and are associated with the document object which represents the HTML document thereby providing an interface to interact with the content of the page.

Event Name	Description
DOMContentLoaded	Triggered when the HTML document has been completely loaded and parsed, without waiting for external resources like images.
click	Fired when the user clicks on an element.
submit	Triggered when a form is submitted.
keydown/keyup/keypress	These events are triggered when a key is pressed, released, or both, respectively.
change	Fired when the value of an input element changes, such as with text inputs or dropdowns.

### Example: Demonstrating Document Events

In this example, we've included a script that listens for the 'DOMContentLoaded,' 'click,' 'submit,' and 'keydown' events on the document. Upon the 'DOMContentLoaded' event, it logs to the console that the DOM content has been fully loaded. Subsequently, clicking an element triggers an alert displaying the tag name of the clicked element. Submitting a form also alerts that the form has been submitted. Furthermore, pressing a key like entering the username with characters, alerts every keypress to the screen.

[Open Compiler](#)

```
<!DOCTYPE html>
<html>
<head>
  <title>Document Events Example</title>
  <script>
    document.addEventListener('DOMContentLoaded', function() {
      alert('DOM content has been fully loaded!');
    });

    document.addEventListener('click', function(event) {
```

```
        alert('Element clicked! Tag Name: ' + event.target.tagName);
    },{once:true});

    document.addEventListener('submit', function() {
        alert('Form submitted!');
    });

    document.addEventListener('keydown', function(event) {
        alert('Key pressed: ' + event.key);
    },{once:true});
</script>
</head>
<body>
    <form>
        <label for="username">Username:</label>
        <input type="text" id="username" name="username">
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```