



Eventi in JS

Programmazione

Gli eventi in JavaScript

In JavaScript, un evento è un'azione o una situazione che avviene all'interno del browser, generata dall'utente o dal sistema stesso.

Gli eventi rappresentano il cuore dell'interattività delle applicazioni web: cliccare su un bottone, digitare testo, scorrere una pagina, oppure caricare completamente un documento HTML, sono esempi comuni di eventi che si possono gestire tramite JavaScript.

Per gestire gli eventi, JavaScript utilizza appositi metodi chiamati gestori di eventi (event handlers) o listener. Questi gestori sono funzioni che reagiscono automaticamente quando si verifica uno specifico evento.

Collegando un evento a una funzione, è possibile definire il comportamento dell'applicazione web in risposta a interazioni specifiche dell'utente, rendendo dinamici e reattivi i contenuti delle pagine web.



Caratteristiche principali degli eventi JavaScript:

- **Interattività:** consentono agli utenti di interagire direttamente con la pagina.
- **Asincronia:** gli eventi avvengono in modo asincrono e non bloccano l'esecuzione principale del codice.
- **Propagazione:** gli eventi possono diffondersi attraverso gli elementi della pagina (bubbling e capturing).
- **Gestione flessibile:** un evento può essere gestito da più funzioni e rimosso dinamicamente.
- **Oggetto evento:** contiene informazioni specifiche sull'evento generato (es. posizione del mouse, tasto premuto).



Lista di esempi di eventi con codice:

Questi esempi costituiscono la base per comprendere come i vari tipi di eventi possano essere intercettati e gestiti con JavaScript, aumentando significativamente l'interattività delle applicazioni web.

- **click** : Azione al click sull'elemento.

```
1. elemento.onclick = function() {  
2.   alert("Hai cliccato!");  
3.};
```

- **mouseover** : Azione quando il mouse passa sopra l'elemento.

```
4. elemento.onmouseover = function() {  
5.   console.log("Mouse sopra l'elemento");  
6.};
```



- **mouseout** : Azione quando il mouse lascia l'elemento.

```
1. elemento.onmouseout = function() {  
2.   console.log("Mouse fuori dall'elemento");  
3.};
```

- **keydown** : Azione quando si preme un tasto.

```
4. document.onkeydown = function(event) {  
5.   console.log("Tasto premuto: " + event.key);  
6.};
```

- **load** : Azione al caricamento completo della pagina.

```
7. window.onload = function() {  
8.   console.log("Pagina caricata");  
9.};
```



Principali problemi e relative soluzioni:

- **Problema: Eventi che si sovrappongono o conflitti tra più listener.**
 - **Soluzione: Utilizzare `event.stopPropagation()` per evitare la propagazione indesiderata.**
- **Problema: Funzioni evento eseguite troppe volte o eventi multipli.**
 - **Soluzione: Rimuovere listener con `removeEventListener` oppure implementare un debounce.**



- **Problema: Gestori di eventi che non si attivano per elementi aggiunti dinamicamente.**
 - **Soluzione: Delegare gli eventi a un elemento genitore già presente nella pagina (event delegation).**
- **Problema: Gestori di eventi anonimi difficili da rimuovere o gestire.**
 - **Soluzione: Usare funzioni nominate per facilitare la rimozione dei listener.**
- **Problema: Errori di contesto con this.**
 - **Soluzione: Utilizzare le arrow functions `() => {}` per mantenere il contesto del `this` appropriato.**



Lista eventi comuni con spiegazione:

- **click**

Evento generato cliccando un elemento con il mouse o toccando su schermi touch.

- **mouseover**

Evento generato quando il cursore del mouse entra in un'area specifica di un elemento.

- **mouseout**

Evento generato quando il cursore del mouse lascia l'area specifica di un elemento.



- **keydown**

Evento generato quando viene premuto un tasto sulla tastiera.

- **keyup**

Evento generato quando si rilascia un tasto sulla tastiera dopo averlo premuto.

- **submit**

Evento generato quando un modulo (form) viene inviato.



- **input**

Evento generato quando il valore di un elemento input viene modificato (digitando, incollando, cancellando testo).

- **load**

Evento generato quando la pagina (o una risorsa specifica) viene completamente caricata.

- **resize**

Evento generato quando la finestra del browser viene ridimensionata.

- **scroll**

Evento generato quando si scorre la pagina con la rotella del mouse o con la barra di scorrimento.



```
1.<!DOCTYPE html>
2.<html lang="it">
3.<head>
4.  <meta charset="UTF-8">
5.  <title>Esempio eventi JavaScript</title>
6.</head>
7.<body>
8.
9.<button id="clicca">Cliccami</button>
10.<input type="text" id="inputTesto" placeholder="Scrivi qualcosa...">
11.<div id="areaMouse" style="width:200px;height:100px;background-color:lightblue;margin-top:10px;">
12.  Passa qui col mouse
13.</div>
14.
15.<script>
16.  // Evento click
17.  document.getElementById("clicca").addEventListener("click", function(){
18.    alert("Hai cliccato il pulsante!");
19.  });
20.
21.  // Evento input
22.  document.getElementById("inputTesto").addEventListener("input", function(event){
23.    console.log("Stai digitando: " + event.target.value);
24.  });
25.
26.  // Eventi mouseover e mouseout
27.  document.getElementById("areaMouse").addEventListener("mouseover", function(){
28.    this.style.backgroundColor = "yellow";
29.    console.log("Mouse sopra la zona");
30.  });
31.
32.  document.getElementById("areaMouse").addEventListener("mouseout", function(){
33.    this.style.backgroundColor = "lightblue";
34.    console.log("Mouse fuori dalla zona");
35.  });
36.
37.  // Evento keydown (pressione tasto)
38.  document.addEventListener("keydown", function(event){
39.    console.log("Hai premuto il tasto: " + event.key);
40.  });
41.
42.  // Evento al caricamento della pagina
43.  window.addEventListener("load", function(){
44.    console.log("La pagina è stata caricata completamente");
45.  });
46.</script>
47.
48.</body>
49.</html>
```

Spiegazione elementi della pagina:

Bottone (<button>):

- Ha l'id "clicca" e genera un evento di tipo "click" quando viene premuto dall'utente.

Campo di testo (<input>):

- Campo con id "inputTesto" che permette all'utente di digitare testo, generando eventi di tipo "input".

Area sensibile al mouse (<div>):

- Un rettangolo colorato con id "areaMouse", reagisce ai movimenti del mouse con due eventi: "mouseover" (entrata) e "mouseout" (uscita).



```
1.<!DOCTYPE html>
2.<html lang="it">
3.<head>
4.  <meta charset="UTF-8">
5.  <title>Secondo esempio eventi JavaScript</title>
6.</head>
7.<body>
8.
9.<form id="formEsempio">
10.  <input type="text" id="focusInput" placeholder="Clicca qui dentro">
11.  <button type="submit">Invia</button>
12.</form>
13.
14.<select id="selezione">
15.  <option>Opzione 1</option>
16.  <option>Opzione 2</option>
17.  <option>Opzione 3</option>
18.</select>
19.
20.<div style="height:800px;"></div>
21.<p id="scrollInfo">Scorri la pagina verso il basso...</p>
22.
23.<script>
24.  // Evento focus (campo input selezionato)
25.  document.getElementById("focusInput").addEventListener("focus", function(){
26.    console.log("Il campo input è selezionato");
27.  });
28.
29.  // Evento blur (campo input perde selezione)
30.  document.getElementById("focusInput").addEventListener("blur", function(){
31.    console.log("Il campo input non è più selezionato");
32.  });
33.
34.  // Evento submit (invio modulo)
35.  document.getElementById("formEsempio").addEventListener("submit", function(event){
36.    event.preventDefault(); // previene l'invio reale del modulo
37.    alert("Modulo inviato virtualmente!");
38.  });
39.
40.  // Evento change (selezione diversa nelle opzioni)
41.  document.getElementById("selezione").addEventListener("change", function(event){
42.    console.log("Hai scelto: " + event.target.value);
43.  });
44.
45.  // Evento scroll (scorrimento pagina)
46.  window.addEventListener("scroll", function(){
47.    document.getElementById("scrollInfo").textContent = "Stai scorrendo la pagina...";
48.  });
49.
50.  // Evento resize (ridimensionamento finestra browser)
51.  window.addEventListener("resize", function(){
52.    console.log("Finestra ridimensionata: " + window.innerWidth + "px");
53.  });
54.</script>
55.
56.</body>
57.</html>
58.
```



Spiegazione degli eventi utilizzati nell'esempio:

focus

- Generato quando un elemento (es. campo testo) viene selezionato per l'input.

blur

- Generato quando l'elemento perde il focus, ovvero non è più selezionato dall'utente.

submit

- Generato al tentativo di invio di un modulo (form). Spesso viene utilizzato preventDefault() per gestire manualmente l'invio.

change

- Generato quando il valore di un elemento (ad es. <select>) viene modificato dall'utente.

scroll

- Generato quando la pagina viene scorsa verticalmente o orizzontalmente.

resize

- Generato quando l'utente modifica le dimensioni della finestra del browser.

Buon Davante a tutti

