

Guida JavaScript per esempi

# Gli eventi

Cosa sono e come vengono organizzati gli eventi JavaScript



Wolfgang Cecchin  
Pubblicato il 17 mar 2006

Oltre essere organizzato a oggetti e metodi, JavaScript sfrutta moltissimo la presenza degli **eventi**.

Un evento è – molto semplicemente – qualcosa che accade nel documento.

Anche nella vita quotidiana un evento è qualcosa che accade: ad esempio una tazzina di caffè che cade e si rompe. Il documento dunque è l'ambiente entro cui vengono realizzate delle "azioni", in parte dovute alla volontà dell'utente (passaggio del mouse su un link, click su qualcosa...), altre volte dovute alla situazione contingente (la pagina è stata caricata).

Grazie agli eventi possiamo "impacchettare" il codice scritto attraverso JavaScript e farlo eseguire non appena l'utente esegue una data azione: quando clicca su un bottone di un form possiamo controllare che i dati siano nel formato giusto; quando passa su un determinato link possiamo scambiare due immagini, eccetera.

Gli eventi (le possibili cose che possono capitare all'interno della pagina) sono davvero tanti e ogni nuova versione del browser si arricchisce di nuove possibilità messe a disposizione degli sviluppatori. Vediamo quali sono quegli eventi che vi capiterà più spesso di incontrare.

Come si può vedere dagli esempi il modo corretto per sfruttare gli eventi è quello di inserire l'evento all'interno della sintassi dell'elemento HTML e racchiudere il codice JavaScript tra virgolette. Secondo questo modello:

```
<ELEMENTO nomeEvento="codice JavaScript">
```

Abbiamo già incontrato in precedenza l' "**alert**" è una finestra di dialogo che avverte di qualcosa.

Negli esempi qui sotto, quando viene eseguita una determinata azione, viene mostrato un **alert** con scritto "ciao".

| evento        | si applica a...               | descrizione  | esempio  |
|---------------|-------------------------------|--|--|
|               | (elementi maggiormente usati) |  |  |
| <b>onload</b> | <body><br><img>               | "sul caricamento..."<br><br>l'evento scatta quando l'oggetto risulta completamente caricato. | <body<br>onload="alert('ciao');"><br><br><img<br>src="miaimmagine.jpg" |

**onunload**

&lt;body&gt;

applicato al <body> si verifica quando il browser "scarica" il documento. cioè alla chiusura del documento.

```
<body
onunload="alert('ci
ao');">
```

**onmouseover**

&lt;a&gt;

&lt;area&gt;

<input> (submit, reset, button, checkbox, radio, option)

"al passaggio del mouse".

si verifica quando l'utente passa il mouse su un'area sensibile (link o mappa).

con ie 5+ questo evento può essere associato anche a td, tr, etc.

```
<a
onmouseover="ale
rt('ciao');"
href="pagina.html"
>
```

**onmouseout**

&lt;a&gt;

&lt;area&gt;

<input> (vedi sopra)

"quando il mouse esce".

si verifica quando l'utente con il mouse esce al di fuori dell'area sensibile (link o mappa).

con ie 5+ questo evento può essere associato anche a td, tr, etc.

```
<a
onmouseout="aler
t('ciao');"
href="pagina.html"
>
```

**onclick**

&lt;a&gt;

&lt;area&gt;

&lt;input&gt;

"sul click del mouse..."

avviene quando l'utente clicca sull'elemento

```
<a
onclick="alert('ciao
');"
href="pagina.html"
>
```

**onkeypress**

&lt;a&gt;

&lt;area&gt;

&lt;input&gt;

&lt;div&gt;

si verifica quando si è all'interno dell'elemento e viene premuto un tasto della tastiera.

```
<textarea
onkeypress="alert(
'ciao');">
</textarea>
```

**onchange**

&lt;input type="select"&gt;

si verifica quando l'elemento subisce un cambiamento a seguito dell'azione dell'utente.

```
<select
onchange="alert('ci
ao');">
<option>
uno
</option>
<option>
due
</option>
<option>
```



|                 |   |  |  |
|-----------------|---|--|--|
| <b>onsubmit</b> | <code>&lt;form&gt;</code>   | quando viene eseguita l'invio del form (tramite bottone o tramite invio da tastiera).                      | <code>&lt;form<br/>name="mioform"<br/>action="http://..."<br/>onsubmit="alert('cia<br/>ao');"&gt;</code>   |
| <b>onfocus</b>  | <code>&lt;a&gt;</code><br><code>&lt;input&gt;</code><br><code>&lt;body&gt;</code> | "quando l'elemento ha il focus..."<br><br>entra in azione non appena l'elemento è attivo.                  | <code>&lt;body<br/>onfocus="alert('cia<br/>o');"&gt;</code><br><br>(provate questo esempio con altre finestre del browser aperte, e provate a passare dalla vostra pagina alle altre finestre e viceversa) |
| <b>onblur</b>   | <code>&lt;a&gt;</code><br><code>&lt;input&gt;</code><br><code>&lt;body&gt;</code> | "quando l'elemento non è più attivo..."<br><br>viene richiamato quando l'elemento smette di essere attivo. | <code>&lt;body<br/>onblur="alert('ciao'<br/>);"&gt;</code>   |

Un discorso a parte merita l'evento **onFocus**, che spesso suscita delle perplessità. "onFocus" significa "quando l'elemento è focalizzato": significa che in quel momento è attivo quel determinato elemento della pagina HTML.

Nel caso dei link e delle mappe vi potete accorgere bene di quando l'elemento ha il focus: il link o la mappa ha infatti una piccola linea tratteggiata che lo circonda (nel caso del link, se l'elemento è focalizzato, si trova nello stato "ACTIVE").

Anche nel caso delle finestre del browser è facile accorgersi di quando un elemento ha il focus: con Windows se la finestra ha il focus (basta che si trovi in primo piano) la barra in alto è azzurra (o del colore definito dall'utente), in caso contrario è grigia.

**onBlur** semplicemente è il contrario di **onFocus**: quando l'elemento che aveva il focus lo perde (perché l'attenzione dell'utente si è concentrata su altro) viene azionato l'evento **onBlur**.

È da notare che a uno stesso elemento possono essere attribuiti più eventi. Ad esempio:

```
<a onmouseover="alert('passaggio sul link');" onmouseout="alert('uscita dal link');"
href="pagina.html">testo del link </a>
```



livello in Internet Explorer 6 è dotato di tutti i seguenti eventi (come si può vedere non c'è l'evento onLoad, né onSubmit, perché riferiti a un livello non avrebbero alcun senso):

onmouseup, oncontextmenu, onrowexit, onbeforepaste, onactivate, onmousemove, onmove, onselectstart, oncontrolselect, onkeypress, oncut, onrowenter, onmousedown, onpaste, onreadystatechange, onbeforedeactivate, onkeydown, onlosecapture, ondrag, ondragstart, oncellchange, onfilterchange, onrowsinserted, ondatasetcomplete, onmousewheel, ondragenter, onblur, onresizeend, onerrorupdate, onbeforecopy, ondblclick, onkeyup, onresize, onmouseover, onmouseleave, onmoveend, onresize, ondrop, onpage, onrowsdelete, onfocusout, ondatasetchanged, ondeactivate, onpropertychange, ondragover, onhelp, ondragend, onbeforeeditfocus, onfocus, onscroll, onbeforeactivate, onbeforecut, onclick, oncopy, onfocusin, onbeforeupdate, ondataavailable, onmovestart, onmouseout, onmouseenter, onlayoutcomplete, onafterupdate, ondragleave

## NOTA A MARGINE

Chi volesse sapere come ho ricavato tutti gli eventi menzionati sopra, sappia che per ricavarli ho utilizzato questo codice JavaScript. È sufficiente inserirlo nel BODY del documento:

```
<div id="mioLiv" style="position:absolute"></div>
```

```
<script type="text/javascript">
```

```
for (prop in document.all.mioLiv) {  
  if (prop.indexOf("on")==0)  
    document.write(prop+"<br/>");  
}
```

```
</script>
```

Se vuoi aggiornamenti su questo argomento inserisci la tua email nel box qui sotto:

Acconsento alla ricezione di comunicazioni promozionali da parte del Titolare, relative a prodotti e servizi di terzi. ☐ Sì ☐ No

Compilando il presente form acconsento a ricevere le informazioni relative ai servizi di cui alla presente pagina ai sensi dell'[informativa sulla privacy](#).

[Lezione precedente](#)[Lezione successiva](#)

### TI CONSIGLIAMO ANCHE



/ PHP

REST API e PHP 8

/ PHP



Configurare un ambiente di sviluppo con PHP 8 e Docker