

Eventi in JavaScript

Windows e tutti i moderni sistemi operativi sono in grado di segnalare alle applicazioni in esecuzione il verificarsi di un determinato *evento*. Ma cos'è un evento?

Gli eventi sono il mezzo utilizzato dagli oggetti per segnalare all'applicazione di cui fanno parte il verificarsi di una determinata situazione.

Un *evento* è un messaggio¹ che il sistema genera in risposta a un'azione effettuata dall'utente, quando interagisce con l'applicazione, o dal sistema stesso.

Può avvenire, pertanto, **a seguito di un'interazione con l'utente** (pressione di un tasto, compilazione di una casella di testo, ...), **o di un cambiamento di stato del componente che ha generato l'evento** (creazione, inizializzazione, completamento del caricamento, ...).

Praticamente ad ogni applicazione vengono segnalati, durante la sua esecuzione, centinaia di eventi.

I nomi degli eventi generalmente ne rispecchiano la natura.

Ad esempio, il nome dell'evento che gestisce la pressione del tasto del mouse (in realtà il *gestore dell'evento*) è *onClick*; l'evento (il *gestore dell'evento*) che segnala l'avvenuto caricamento di un oggetto prende il nome di *onLoad* e così via.

Le applicazioni JavaScript offrono la possibilità di gestire gli eventi che riguardano la finestra del browser ed il documento caricato.

Gestire un evento vuol dire collegare ad esso una funzione in modo che questa venga eseguita quando e se quel determinato evento dovesse verificarsi.

Gli *eventi*, in JavaScript, *vengono sempre generati da un oggetto*, quindi sarà sufficiente inserire un ulteriore parametro nel TAG di definizione dell'oggetto in questione.

Questo parametro conterrà il nome dell'evento che vogliamo gestire ed il nome della funzione che desideriamo collegarvi.

Ad esempio, si può gestire l'evento "*ingresso del puntatore del mouse nell'area dell'oggetto*" di un oggetto di tipo immagine.

```
<IMG SRC="img1.gif" onMouseOver="funzione1()">
```

In questo modo, quando l'utente muoverà il puntatore del mouse sull'area occupata dall'immagine *img1.gif* verrà richiamata la funzione *funzione1()*. Definiamo *funzione1()* nel seguente modo:

```
function funzione1()  
{  
    alert("Sei sull'immagine img1.gif");  
}
```

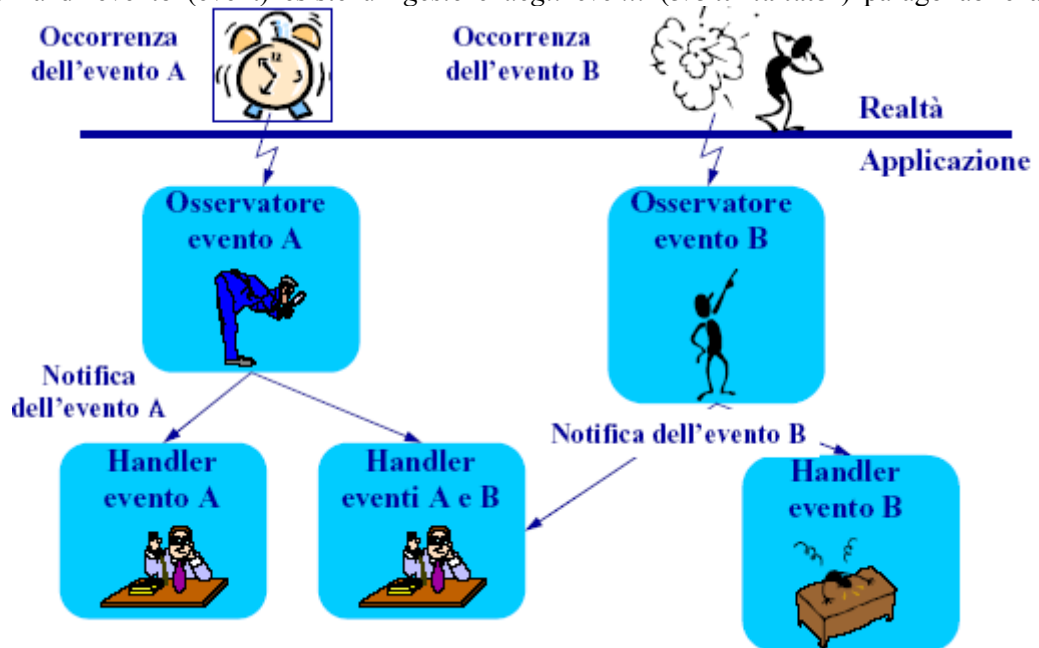
¹ In alcuni ambienti visuali (come Toolbook) viene usata una terminologia differente. Si dice infatti che l'evento ('azione' o occorrenza dell'evento) "click del mouse" su un oggetto genera i messaggi ('eventi') "buttonDown, buttonUp, buttonClick", nell'ordine.

Gestione di eventi in Javascript

L'**occorrenza di un evento** può essere definita come l'azione che si manifesta, di solito, successivamente ad operazioni dell'utente, come la pressione del mouse, il passaggio da una finestra a un'altra o il cambio di focus².

def: L'**occorrenza di un evento** è l'azione che l'utente esegue sull'oggetto proprietario del focus (un pulsante, un'immagine o un radio button).

Per ogni occorrenza di evento (event) esiste un *gestore degli eventi (event handler)* paragonabile a un



puntatore che descrive l'evento stesso. Utilizzando il gestore degli eventi è possibile scrivere gli script per eseguire le operazioni (**handler code**) quando l'utente scatena l'evento. Ad esempio, è possibile controllare la coerenza dei dati inseriti in un form quando l'utente preme il tasto di invio.

La parola chiave che rappresenta un gestore di eventi è costituita dal prefisso "**on**" e dal nome dell'evento stesso: all'evento Click corrisponde il gestore *onClick*.

Se un oggetto è in grado di generare un determinato evento, è possibile definire per esso un appropriato *gestore degli eventi*.

La sintassi generale per far ciò è la seguente: `<TAG gestoreEvento=\"Codice JavaScript\">`

dove TAG è un tag HTML e *gestoreEvento* è il nome composto dal prefisso **on** seguito dal nome che rispecchia la natura dell'occorrenza dell'evento da gestire.

In genere si possono inserire più comandi JavaScript per gestire gli eventi, separando gli stessi con punti e virgola ma conviene utilizzare delle funzioni collegate ai gestori degli eventi, in quanto ciò rende il codice più leggibile e modulare ed è *bene* porre le *definizioni di funzioni nella parte <HEAD>* di un documento in modo che vengano definite all'inizio del caricamento di una pagina, altrimenti l'utente potrebbe eseguire un'azione che porta all'esecuzione di un gestore di eventi che chiama una funzione indefinita, causando un errore.

² Il **focus** rappresenta l'oggetto che in quel preciso istante detiene l'attenzione (il controllo da parte del programma o del sistema operativo) cioè è in esecuzione rispetto a tutti gli altri.