

jQuery: Come creare delle finestre di dialogo personalizzate (1/2)

In questo articolo vedremo come sviluppare delle **finestre di dialogo personalizzate sia nell'estetica che nelle funzionalità**, andando così oltre alle limitatissime finestre messe a disposizione nativamente da javascript (alert, confirm, prompt).

Lo faremo grazie a [jQuery](#) ed in particolare a [dialog](#), un plugin appartenente alle [user interface](#). In questo articolo vedremo nel dettaglio questo plugin, le sue funzionalità e come utilizzarlo. Alla fine del tutorial non avrai difficoltà a passare da una [finestra di dialogo classica](#) ad una più elaborata ed accattivante come [questa](#).

Creare il pacchetto “custom”

Come ho già accennato in altri articoli, quando vogliamo utilizzare una *user interface* di jQuery, possiamo scaricare un pacchetto creato “on fly” contenente il **plugin e tutte le dipendenze necessarie**. Collegati allora all'[area di download](#) delle ui ed inizia a deselezionare tutto. Ora individua il widget *dialog* e selezionalo, in questo modo si selezioneranno automaticamente anche le dipendenze necessarie (ui.core, ui.widget e ui.position). Seleziona anche ui.button. È una dipendenza facoltativa; se installata sostituisce i normali pulsanti con quelli delle user interface. Ora che hai scaricato il pacchetto, possiamo procedere ad impostare la pagina.

Preparare la pagina

Crea ora la pagina *dialog.html*. Nell'header includi jQuery, il pacchetto custom ed il foglio di stile.

Creare la finestra di dialogo

Il primo esempio che realizzeremo è [questo](#).

Iniziamo a definire la finestra di dialogo nel body del nostro documento, in questo modo

Come puoi vedere, la finestra di dialogo consiste semplicemente in un div, che rappresenta il contenitore all'interno del quale possiamo scrivere il codice html che desideriamo. Nota che l'attributo *title* del div rappresenta il **contenuto evidenziato nella parte superiore della finestra**.

Dare vita alla finestra

Per rendere questa porzione di codice una finestra di dialogo, dovremo semplicemente passarla al metodo *dialog*, in questo modo

Ecco il documento completo

Ed ora divertiamoci

Poniamo di voler realizzare [questo](#) secondo esempio. Come vedi ho aggiunto il pulsante “*chiudi*” ed ho reso opaco lo sfondo. Questa seconda opzione la si può ottenere settando a TRUE il parametro *modal*.

I bottoni vanno invece definiti tramite l'accoppiamento *nome bottone -> funzione da eseguire alla pressione del bottone*. Vediamo

Come vedi faccio corrispondere al bottone “*chiudi*” una funzione nella quale viene applicato il metodo *close* alla finestra.

Possiamo definire tutti i bottoni che desideriamo ed attribuirgli le funzioni più bizzarre; vediamo questo [esempio](#).

Scopriamo come realizzarlo.

Innanzitutto modifica il body del documento in questo modo:

Come vedi ho modificato il contenuto della finestra ed ho aggiunto un paragrafo vuoto, nel quale andremo ad inserire la scritta “*ciao mondo*” quando sarà richiesto.

Il codice jQuery lo modificheremo invece in questo modo

La funzione relativa al bottone “*scrivi ciao mondo*”, prima di chiudere la finestra, va ad inserire nel paragrafo che abbiamo aggiunto la stringa “*ciao mondo*” tramite il metodo *html*.

Aprire la finestra in relazione ad un evento

Negli esempi che abbiamo visto fino ad ora la finestra si apre al caricamento della pagina. Questo in quanto il parametro *autoOpen* è impostato di default a TRUE. Se lo impostiamo su FALSE, **la finestra non si aprirà fin tanto che non eseguiamo il metodo *dialog* (“*open*”)**.

Dunque potremo legare l'apertura della finestra ad un evento, per esempio il click su di un link, come in [questa](#) pagina.

Per realizzare quanto visto, inizia ad aggiungere il link nel body del documento:

In seguito, nei parametri di *dialog*, aggiungi *autoOpen: false*.

Ed infine intercettiamo l'evento click sul link con id *open*, in questo modo:

Ed ecco la funzione completa:

Conclusioni

Come hai potuto vedere, il metodo *diag* é molto semplice e flessibile. Possiamo inoltre dare l'aspetto che vogliamo alla nostra finestra **senza nessun limite in quanto dipende interamente da html e css.**

Non vi é neppure **nessun limite per quello che riguarda i pulsanti e le funzionalità che andranno ad implementare.**

Ma continuiamo a divertirci!

Nel secondo articolo infatti, svilupperemo qualcosa di più complesso. Un modulo di login tramite finestra di dialogo con chiamata ajax per verificare i dati di autenticazione.