

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

Come promesso, eccoci a ritornare sul tema delle **finestre di dialogo** sviluppate in **jQuery**. In questo articolo vedremo **come realizzare un sistema di login basato su questo tipo di interfaccia**; per ingolosirti ti mostro subito il [risultato finale](#).

Il funzionamento, come puoi vedere, prevede che alla pressione del pulsante “login”, abbia luogo una chiamata asincrona che si occuperà di verificare i dati di autenticazione. Se questi non risultassero corretti, verrà mostrato un messaggio di errore.

Mi permetto di dare per scontato che tu abbia letto il precedente [articolo](#) che fornisce le basi per quello che andremo a vedere ora. Sei pronto?

### Prepariamo la pagina

Creiamo la pagina *login.php* in questo modo

Come vedi abbiamo incluso nell’header i file necessari: jQuery, il file custom contenente il plugin e le dipendenze, il foglio di stile di jQuery e *style.css* che invece é il nostro foglio di stile nel quale definiremo le proprietà css del modulo di login.

### Il file di supporto per l’autenticazione

La verifica dei dati di autenticazione avverrà tramite una chiamata ajax. Sarà dunque necessario avere **un file di supporto che verifichi tali dati e fornisca una risposta**.

Nella pratica la procedura sarà certamente più complessa. Noi ci limiteremo a verificare che *password* contenga “1234” ed *username* contenga “admin”, nel qual caso valorizzeremo la variabile di sessione *auth* con 1. Ogni volta che vorremo stabilire se l’utente é autenticato, verificheremo che la variabile *auth* sia settata con valore 1.

Scriviamo dunque il file *verify\_login.php* in questo modo:

Nota che stampiamo a video *1* in caso di procedura andata a buon fine o *0* in caso di verifica mancata. Questi valori verranno intercettati dalla chiamata ajax in modo da poter visualizzare il messaggio di errore se dovesse essere il caso.

## Riscontro dell'autenticazione

Come puoi vedere nella pagina di esempio, nel caso in cui l'utente non é autenticato, viene mostrata la scritta "utente sconosciuto" ed il link "login". Dopo l'autenticazione viene invece mostrata la scritta "utente autenticato" ed il link "logout".

Per poter fare questo (in modo da avere un effettivo riscontro dell'avvenuta autenticazione), aggiungiamo questo codice nel body di *login.php*:

Naturalmente dovremo anche inserire, nella prima riga di *login.php*, l'avvio delle sessione

Ora abbiamo inserito i due link. Quello per il *login* non punta a nessuna pagina in quanto andrà ad attivare la finestra di login. Il link per il *logout* punta invece alla pagina *logout.php*

## Implementare il logout

La pagina *logout.php* si occuperà semplicemente di "cancellare" la variabile di sessione *auth*. In questo modo

## Preparare la finestra di dialogo del login

Aggiungiamo ora nel body di *login.php* la parte di codice che diventerà poi la finestra di dialogo contenente il modulo di login e che, come vedremo, presenta diverse curiosità.

Nella seconda riga, inseriamo un paragrafo (classe *error*) che **contiene il messaggio di errore da mostrare in caso di mancata autenticazione**. Successivamente, nel foglio di stile, mediante un'apposita regola, renderemo **non visibile il messaggio di errore e faremo in modo che questo venga mostrato all'occorrenza mediante jQuery**.

Ai campi *username* e *password* assegniamo la classe *ui-corner-all*. Questa classe fa parte del foglio di stile di jQuery ed arrotonda i quattro angoli di un elemento.

Nota che jQuery ha già aggiornato questa classe con le nuove proprietà di IE9 che, dopo un'attesa di 8.000 anni, fornisce finalmente il supporto per l'arrotondamento degli angoli mediante CSS (!).

## Il foglio di stile

Creiamo ora il file *style.css* ed iniziamo a definire lo stile della finestra di dialogo:

Nulla di particolare fino ad ora: abbiamo definito il font, eliminato il bordo del fieldset e definito i campi di input.

Ora andiamo ad inserire le icone nei campi *password* e *username*:

Definiamo in seguito il paragrafo contenente il messaggio di errore come **non visibile** ed attribuiamogli un colore rosso ed un font grassetto:

Ed infine definiamo i link per la registrazione ed il recupero della password:

## Jquery e finestre di dialogo: dare vita alla finestra

Iniziamo finalmente a lavorare sul codice **jQuery**. La prima cosa che dovremo fare é sviluppare la funzione che apre la **finestra di dialogo** quando si clicca sul link *login*.

Ed ora iniziamo a definire la finestra di dialogo:

Fino a qui niente di nuovo, abbiamo definito la larghezza della finestra tramite parametro. Ora vediamo i bottoni. Su annulla c'è poco da dire, lo abbiamo già visto nell'articolo precedente.

Il pulsante "login" invece é un po' più complesso. Quando viene premuto bisognerà innanzitutto **leggere il contenuto dei campi**. Lo faremo con il metodo *serialize* sul quale ho scritto già un [articolo](#).

Ora la variabile *querystring* dovrà essere inviata al file *verify\_login.php* tramite una chiamata ajax.

La funzione di callback di questa chiamata, dovrà verificare la risposta. Se la risposta é 0, significa che i dati non erano corretti, **dunque mostreremo il paragrafo di errore** (magari tramite un'animazione).

Se la risposta é 1, il login é andato a buon fine e la variabile di sessione *auth* sarà valorizzata. Ma

per vederlo dovremo ricaricare la pagina.

A questo punto manca solo un piccolo particolare. Vai nella pagina di esempio ed inserisci dei dati sbagliati. Premendo su "login" verrà correttamente mostrato il messaggio di errore.

Ora, se provi a correggere i dati, il messaggio di errore sparisce.

Questa funzionalità trova ragione in una migliore usabilità. Infatti, se non dovessimo eliminare il messaggio di errore e l'utente sbagliasse una seconda volta i dati, si potrebbe avere l'impressione che non sia successo nulla.

Dunque quello che dovremo fare é intercettare l'ingresso del cursore nel campo (evento focus) e verificare se il messaggio di errore é visibile. Se lo é, lo nasconderemo utilizzando l'animazione opposta (slideUp) a quella che abbiamo utilizzato per mostrarlo.

E finalmente ecco la pagina *login.php* finita

## Download

Se lo desideri puoi scaricare tutti i file utilizzati per realizzare l'esempio descritto in questo tutorial.

## Conclusione

Come hai potuto vedere, dopo il primo articolo nel quale abbiamo trattato il tema delle finestre di dialogo in maniera introduttiva, siamo arrivati ora ad **un'applicazione pratica piuttosto avanzata** in grado di dare uno stile di sicuro effetto alla procedura di login.

E' ovviamente possibile lavorare sul foglio di stile in modo da modellare la finestra e ottenere qualsiasi risultato desiderato.

E tu, dopo questi due articoli, introdurrà le finestre di dialogo personalizzate nei tuoi siti?