

## L'autocompletamento di un campo, una nuova ui di jQuery (1/2)

Tra le novità introdotte recentemente da jQuery, troviamo finalmente (mancava proprio) **un'estensione per l'autocompletamento dei campi form**; sto parlando del metodo *autocomplete* appartenente alle *user interface*.

Questa ui permette, con la consueta disarmante semplicità, di implementare una *ciliegina* di usabilità nei nostri siti, come nell'[esempio](#) che puoi vedere.

Vuoi scoprire come?

### Prepariamo il documento (x)html

Iniziamo con il preparare un documento che chiameremo *index.html*.

Come vedi si tratta di un semplicissimo documento, nel quale ho inserito un campo di input con id *enter*.

### Richiamare le librerie necessarie

Vediamo come impostare l'header della pagina *index.html*

Dovremo innanzi tutto richiamare jQuery:

In seguito, visto che si tratta di un *user interface*, certamente ne dovremo richiamare il core.

Se poi guardiamo nella [documentazione](#) (bisognerebbe sempre farlo), scopriamo che **vi sono altre due dipendenze**: *ui.widget* ed *ui.position*

Andiamo dunque ad aggiungerle

Ed infine richiamiamo anche *ui.autocomplete*

*Una piccola nota: le dipendenze vanno sempre richiamate prima dell'estensione che intendiamo utilizzare.*

Ora, visto che si tratta di una *user interface*, **colleghiamo anche il foglio di stile** che, come ho già avuto modo di dire, è totalmente personalizzabile in maniera visuale nell'apposita [sezione](#) del sito di jQuery.

## Inserire il codice jQuery

A questo punto sviluppiamo la semplice funzione jQuery. Dovremo specificare su quale elemento, nel nostro caso *input#enter*, vogliamo applicare l'autocompletamento.

In questo modo però non funzionerà ancora. Dobbiamo ovviamente ancora dire **dove andare a prendere i dati che verranno poi visualizzati sotto il campo**. Lo faremo tramite il parametro *source* che chiaramente è **obbligatorio definire**.

Il caso più semplice è quello di passare a *source* un'array contenente i dati.

Ed ora basterà passare l'array *parole* come *source*.

Ora, come puoi anche vedere nell'[esempio](#), inserendo "b" compariranno come possibilità "bene" e "barba". Se continuiamo, e scriviamo "ba" rimarrà solo barba.

La pagina completa dovrebbe ora risultare così

## Vediamo alcuni parametri

Come sempre, è possibile configurare ulteriormente lo svolgimento del processo attraverso i parametri messi a disposizione dal plugin.

Pensiamo ad esempio se nella nostra lista vi fossero 100 parole che iniziano con la "b". Come potremmo fare per **evitare una finestra lunghissima sotto il nostro campo** scrivendo la lettera "b"?

Possiamo utilizzare il parametro *minLength* (di default settato ad 1), con il quale indichiamo **dopo quanti caratteri inizializzare l'autocompletamento**.

Dunque, riscrivendo il codice in questo modo:

la lista con i suggerimenti comparirà solo **a partire dal terzo carattere** che inseriremo.

Per fare un altro esempio, potremo vedere come **intercettare direttamente la scelta fatta dall'utente**. Il metodo *autocomplete* ci mette a disposizione il gestore dell'evento *select* con il quale indichiamo tramite una funzione cosa vogliamo fare quando appunto viene selezionato un elemento.

Facciamo dunque in modo che venga aperta una finestra che ci dica che elemento abbiamo scelto,

procedendo semplicemente in questo modo:

## Conclusion

Bisogna dire che nella pratica sarà piuttosto raro trovare una situazione del genere e cioè con un array già predefinito.

Più comunemente la situazione potrebbe essere **una lista contenuta in un database** con l'autocompletamento che avviene tramite una chiamata ajax che restituisce la lista dei suggerimenti.

**Questo è quello che vedremo nel prossimo articolo.**

Se credi che la cosa si complicherà all'inverosimile ti sbagli. Ti posso anticipare che basterà inserire come *source* un nome di file:

A questo punto jQuery provvederà automaticamente, ad ogni rilascio di tasto, ad inviare il contenuto del campo tramite una chiamata asincrona a *file.php*, il quale restituirà il risultato che verrà inviato alla nostra pagina.

Quello che vedremo nel dettaglio è **come dovrà essere sviluppato il codice php**, in quanto jQuery si aspetta il risultato in un formato preciso ed utilizza dei parametri standard sia per inviare che per ricevere i dati.