jQuery - Creazione di form con label animate

Anche in questo articolo, ed ancora una volta, integreremo nel nostro sito un effetto semplice e carino utilizzando poche righe di JavaScript, grazie alla libreria jQuery.
Questa volta, realizzeremo dei Label animati, che accompagneranno l'utente nella compilazione dei form; in sostanza si tratta della scritta vicino ad un campo input, che descrive il tipo di dato da inserire.

Puoi vedere un esempio di ciò che andremo a creare facendo click qui.
Per la realizzazione di questo esempio, utilizzeremo qualche riga di JavaScript, un pizzico di XHTML ed una spruzzatina di CSS.

Hai tutti gli ingredienti ?? Bene, allora iniziamo con la ricetta!

XHTML - Preparazione della struttura

La struttura può essere sia semplice che complessa, in questo esempio si tratta semplicemente di alcuni paragrafi che contengono il Label e l'Input; ecco il codice:

La struttura, come puoi vedere, è molto semplice; puoi comunque realizzarla come meglio credi, l'importante è che ogni contenitore (in questo caso i paragrafi

) contenga solamente l'input e il label.

Ora che abbiamo la struttura pronta, psssiamo alla creazione del foglio di stile; ecco che cosa faremo in poche parole:

1. Ad ogni elemento che contiene l'input ed il label verrà attribuita la propietà CSS position: relative; in modo che ci permetta di posizionare facilmente il contenuto all'interno di essa.
2. Ai label verrà invece attribuita la propietà CSS position: absolute; e verranno inoltre messi "dietro" agli input, questo verrà fatto attribuendo la propietà CSS z-index: -1;
3. Per far si che l'input non copra il testo del label, l'input dovrà avere la propietà background-color: transparent;

Ok, il lavoro da fare non è di certo tanto, inoltre, anche se per qualcuno potrebbe essere scontato, cerchiamo di capire meglio il perchè del posizionamento del label subito "dietro" l'input.

Supponiamo di aver attribuito al label un posizionamento assoluto ed uno z-index di 10, l'input sarebbe rimasto posizionato "dietro" al label, e quindi sarebbe impossibile per l'utente compilare i propri dati.
Inoltre così facendo, avremmo dovuto scrivere molte più righe di JavaScript per poter spostare il label al click sull'input (che sarebbe da simulare diversamente, ad esempio, al passaggio del mouse sul label).

Quindi, la soluzione secondo me migliore, è quella di posizionare il label "dietro" all'input, e di attribuire a quest'ultimo un backround-color: transparent, in modo che si veda il contenuto del label.

CSS - dalle parole ai fatti

Ora che hai capito come procedere, passiamo subito all'azione e andiamo a tradurre quanto detto in linguaggio CSS:

NB: è importante mettere il contenitore dei label+input in posizione relativa (position: relative).

I valori delle proprietà top e left sono calcolati in base al tipo di struttura del mio esempio, possono essere modificati a seconda della struttura creata.

In più di quanto detto, c'è la classe active associato al label, infatti quando su quest'ultimo verrà associata l'animazione, gli verrà anche associata la classe active, in modo che il testo risalti un po' di più.

jQuery - un pizzico di magia, per il funzionamento dell'esempio

Cerchiamo di capire, anche in questo caso, che cosa dobbiamo fare:

1. All'evento "click" sull'input il label verrà spostato verso destra, in modo da lasciare l'utente libero di compilare il campo.
2. Verrà memorizzato il valore del label, in modo da poter confrontare il valore scritto dall'utente con quello di default.
3. Al Focus Out (metodo blur() di jQuery), verrà controllato il valore inserito dell'utente: se quest'ultimo è vuoto, oppure è identico al valore di default (il contenuto del label), il valore dell'input viene svuotato, e il label viene riportato alla posizione originale.

I metodi di jQuery che utilizzeremo sono i seguenti: click(), blur(), prev(), addClass(), removeClass(), animate(), html() e val().

Ecco il codice che esegue quanto detto prima:

Vediamo ora passo passo, che cosa si occupa di fare il codice.

Prima di tutto, inseriamo all'interno della variabile $inputs, il risultato della query '#contact input', vale a dire tutti gli input contenuti all'interno dell'elemento avente per ID 'contact'.
Memorizzare l'oggetto jQuery ($('#contact input')) all'interno di una variabile ci permette di ottimizzare il nostro codice, infatti, così facendo, se in futuro avessimo bisogno de effettuare una nuova operazione sugli inputs, non abbiamo più bisogno di richiamare jQuery e fargli analizzare il DOM (operazione che richiede tempo, specialmente se utilizziamo selettori complessi) riducendo quindi, il tempo di esecuzione del nostro script: di seguito un esempio di quanto appena detto:

n.b.: nel secondo caso di codice ottimizzato, possiamo anche saltare un passaggio, poichè la maggior parte dei metodi di jQuery ritorna l'oggetto sul quale viene eseguita l'azione.

Utilizziamo il metodo click, per associare un listener agli input, infatti il contenuto della funzione verrà eseguito ad ogni click su qualsiasi input.

La keyword this, all'interno del metodo click, rappresenta l'oggeto del DOM che è stato cliccato. Visto che this è un'oggetto possiamo associarci ciò che vogliamo, l'importante e non utilizzare keyword riservate.

All'inizio della funzione controlliamo se la proprietà label è già stata definita, se non è così, assegnamo alla proprietà il risultato della query $(this).prev(), e quindi il label che si trova subito prima dell'input. (in questo caso, la proprietà label, è un oggetto jQuery)

Facciamo un'ulteriore controllo, questa volta per vedere se la proprietà labelValue è stata definita, se non è così, inseriamo al suo interno il contenuto html del label.

Infine, aggiungiamo al label la classe 'active' per evidenziare meglio il testo, e tramite il metodo animate() lo spostiamo fino a raggiungere un distanza di 175 px da sinistra (left: 175).

NB: in questo esempio abbiamo definito le variabili label e labelValue come propietà dell'oggetto this, in questo modo queste proprietà saranno ancora accessibili ogni volta che this punterà allo stesso oggetto;

A questo punto, utilizziamo il metodo blur(), per eseguire il nostro codice ogni volta che l'input perderà il focus, vale a dire, quando si cliccherà in un qualsiasi punto esterno all'input.

Nel codice di blocco appena visto abbiamo definito la variabile current che contiene il valore attuale dell'input; a quel punto, viene effettuato un controllo per capire se il valore è vuoto oppure è identico al contenuto del label: in caso positivo, il valore dell'input viene svuotato ed il label viene riportato alla sua posizione originale, in caso contrario, il label viene lasciato dov'è e pure il contenuto dell'input.

Rendiamo il nostro script accessibile

Visto che il nostro script viene applicato, in questo esempio, ad una form, è bene assicurarsi che quest'ultima sia perfettamente accessibile anche da chi ha JavaScript disabilitato.
Se provi infatti a disattivare JavaScript e quindi a visualizzare l'esempio ti accorgerai che, nonostante sia possibile scrivere all'interno di ogni input, le label rimangono visibili e costringono a chi compila i vari campi a scriverci sopra.

Per ovviare a questo problema, una possibile soluzione è la seguente: tramite CSS mostriamo le label nella loro posizione finale (left: 175px) ed aggiungiamo anche la classe active, in modo da evidenziarle maggiormente; così facendo, sia che l'utente abbia disabilitato JavaScript sia che l'utente l'abbia abilitato, le label vengono visualizzate alla destra degli input.

A questo punto, utilizziamo JavaScript per riportare le label alla posizione iniziale (left: 7px); ecco il codice che dovremo utilizzare:

esempio animato

esempio statico

Ovviamente, per il corretto funzionamento, il codice va inserito al caricamente del DOM, in questo modo:

Potete vedere questo esempio cliccando qui.

Qualche considerazione per finire

Anche in questo articolo, hai visto com'è facile creare dei semplici effetti da applicare ai nostri siti grazie alla libreria jQuery; infatti, tutto ciò che occorre è un po' di fantasia!

Nel nostro esempio abbiamo applicato il codice alle label di una form per puro effetto didattico, ma, con un po' di fantasia, questo effetto può essere implementato anche in tantissimi altri modi.

Tu cosa ne pensi di questo esempio ? In che modo lo implementeresti all'interno di un sito ?