

jQuery - Come Creare un News Ticker con poche righe di JavaScript?

Ormai lo sapete, amo reinventare la ruota (e jQuery me lo permette sempre), ed è per questo che, ancora una volta, vedremo come creare qualcosa che esiste già: un **news ticker**.

Solitamente il "news ticker" è un **visualizzatore di notizie** che **scorrono** in orizzontale, come quelle che **si possono vedere in alcuni notiziari nella parte bassa dello schermo**. Il nostro news ticker sarà leggermente diverso dato che le notizie non scorreranno in orizzontale ma in verticale, e tra il passaggio di una notizia all'altra ci sarà una pausa pre-stabilita (credo che questo metodo dia più risalto ad ogni singola notizia e allo stesso tempo conceda maggior tempo all'utente per la lettura dell'articolo in questione, ma questa ovviamente è soltanto una mia opinione).

Per creare il News Ticker utilizzeremo la libreria jQuery che ci permette con poche righe di codice (nel nostro caso undici) di ottenere effetti d'impatto, che consumano poche risorse e danno quel tocco in più al nostro sito.

Grazie a questo semplice articolo imparerai a **creare un News Ticker**, come quello visibile [qui](#). Tutto ciò di cui abbiamo bisogno è (X)HTML, CSS ed un pizzico di jQuery.

Procuriamo tutto l'occorrente

La prima cosa da fare è preparare un documento html su cui lavorare: ciò di cui abbiamo bisogno è davvero poco, poichè l'animazione di per se è molto semplice.

Vediamo di seguito il markup (X)HTML utilizzato per ottenere l'effetto desiderato:

Prima di tutto, dobbiamo ovviamente includere la libreria di jQuery, scaricabile [cliccando qui](#), ed un file esterno (nel nostro caso denominato "*common.js*") nel quale inseriremo lo script per il funzionamento del news ticker.

Quello che è essenziale per il nostro script è un'elenco non ordinato di elementi: la struttura **ul/li**. In sostanza si tratta di una serie di notizie, "racchiuse" all'interno di una "gabbia", la quale permette la visualizzazione di massimo "N" notizie alla volta.

Prendendo come esempio il codice proposto, vediamo quali sono le caratteristiche che il nostro elenco deve avere.

L'*ul*, che ha come id "*ticker*", **deve avere una altezza fissa** che corrisponde all'**altezza di ogni li moltiplicata per il numero di elementi da visualizzare**: se ad esempio, l'altezza di ogni *li* è di *100px* e voglio visualizzarne soltanto due, allora l'altezza dell'*ul* sarà di *200px (100x2)* e così via.

Nel nostro caso, l'altezza dell'*ul* corrisponde all'altezza di ogni *li*, poichè ne visualizzeremo al massimo uno, per dare più risalto ad ogni singola notizia.

Oltre a definire un'altezza, è molto importante che il nostro **ul** abbia la **proprietà overflow settata su hidden**, in modo che tutto il contenuto che eccede l'altezza definita non venga visualizzato.

Vediamo di seguito il codice CSS utilizzato per la formattazione della nostra lista di notizie.

Come puoi vedere, impostare la struttura del ticker è molto semplice, in questo esempio abbiamo applicato anche un piccolo bordo ed un padding a tutti i *li*, ma **ognuno è libero di personalizzare come meglio crede la lista di notizie**, perchè come abbiamo già detto, la parte più importante riguarda la formattazione dell'*ul*.

Una volta che la nostra struttura è pronta, possiamo passare alla parte più divertente: JavaScript.

Come realizzare lo script per far funzionare il News Ticker?

Il nostro script dovrà essere una semplice funzione che **ogni tot secondi rimuova la prima notizia** della lista, e la **re-inserisca in ultima posizione**; per rendere il tutto più simpatico, **sposteremo la prima notizia con margine superiore negativo**, in modo che si muova verso l'alto e quindi non venga più visualizzata (grazie alla proprietà `overflow:hidden` applicata all'*ul*), e di conseguenza **la seconda notizia andrà automaticamente a coprire la posizione della prima**.

Ecco di seguito il codice con cui metteremo in pratica quanto appena detto:

Iniziamo a capirne il significato.

Viene definita una **nuova funzione** e le viene assegnato il **nome "ticker"**; tutto il suo contenuto verrà eseguito dopo **X secondi** (in questo caso 4), per farlo, ci avvaliamo dell'utilizzo del **temporizzatore setTimeout()**, al quale come **primo argomento passiamo il codice da eseguire**, e come **secondo la quantità di ms (millisecondi) dopo i quali eseguirla**.

All'interno della funzione ticker **selezioniamo il primo li che si trova all'interno di "#ticker"**, ovvero dell'elemento che ha come id "*ticker*" (il nostro *ul*); per farlo, il selettore che utilizziamo è il seguente: `$('#ticker li:first')`.

Una volta ottenuto l'elemento, dobbiamo **spostarlo verso l'alto in modo che l'elemento seguente si posizioni al suo posto**, per farlo dobbiamo animarlo fino a che raggiunga, come **margine superiore il valore di -120px** (`margin-top: -120px`), vale a dire **l'altezza di ogni singolo li**.

Per ottenere l'effetto appena descritto, applichiamo all'elemento il metodo [animate\(\)](#), e **come parametro d'animazione**, utilizziamo il **margin superiore** (*marginTop*) che avrà come valore l'altezza di ogni *li* (nel nostro caso 120px) in negativo, in modo che si sposti verso l'alto (*marginTop: '-120px'*); **la durata dell'animazione** viene passata come **secondo parametro di animate()**, ed in questo caso è di **800ms**.

Il **terzo parametro da passare ad animate()** rappresenta il nostro **callback**, e cioè, la funzione che verrà chiamata quando l'animazione sarà finita.

Ecco che avviene la *magia*: tramite il metodo [detach\(\)](#) di jQuery, **rimuoviamo dal DOM** il *li* selezionato, ma **teniamo in memoria l'elemento**, in modo da poterlo riutilizzare.

A questo punto, utilizziamo il metodo [appendTo\(\)](#), per "**appendere**" l'elemento ad un nuovo **selettore (il nostro ul)**, ed infine applichiamo il metodo [removeAttr\(\)](#), che si occupa di **rimuovere dall'elemento, l'attributo passato come argomento (style, in questo caso)**, altrimenti l'elemento verrebbe reinserto con le ultime proprietà applicate dall'animazione, e cioè *margin-top: -120px*.

Subito **dopo l'applicazione del metodo animate()** al nostro elemento, la funzione "chiama" se stessa, in modo di **ripetere nuovamente l'operazione** dopo 4 secondi.

*NB: se il nostro file javascript viene chiamato all'interno del tag head, è **importante inserire il nostro script all'interno del metodo ready() di jQuery**, in modo che venga lanciata al caricamento del DOM, nel nostro caso lo facciamo tramite il seguente codice:*

quanto visto, è la versione compatta di:

Conclusioni

Ancora una volta, jQuery ci semplifica enormemente la vita; infatti grazie a sole 12 righe di JavaScript, siamo riusciti a creare un piccolo News Ticker molto comodo.

Questa volta reinventare la ruota non è stato difficile... tu cosa ne pensi ?