

Scrolling: usiamolo per le animazioni

Chi mi segue da qualche tempo forse saprà che amo particolarmente le animazioni. Questo perché, se implementate adeguatamente, sono un potentissimo strumento per migliorare l'esperienza dell'utente e aumentare il tasso di conversione. Tuttavia, perché limitarsi ad animare passivamente degli elementi a un determinato evento? Oggi ti farò vedere come manipolare un'animazione allo scrolling.

Dietro le quinte

Il ragionamento che si cela dietro questo meccanismo è piuttosto basilare, ma di grande effetto: misuriamo innanzitutto **la distanza da un punto A a un punto B**; quando quest'ultimo sarà uguale a 1 (usando un dividendo che vedremo più avanti) rappresenterà lo stato completo dell'animazione.

Immagina che il punto A sia il bordo superiore della finestra e il punto B sia quello inferiore: in questo caso la sintassi jQuery potrebbe essere la seguente:

A questo punto, basta semplicemente misurare l'attuale valore di *scrollTop()* (metodo che restituisce il numero di pixel scrollati) e dividerlo per la distanza sopracitata.

Ipotizziamo ora che il valore di *scrollTop()* sia di 500, mentre l'altezza della finestra sia di 1000 (px, ma JavaScript restituisce i valori in numeri semplici). Effettuando il calcolo, noterai che il valore risulta essere "0.5".

Ti ricorda qualcosa?

Proprio così: potremo controllare ogni valore dell'elemento alla perfezione. Infatti, qualora lo *scrollTop()* fosse uguale a 400, l'opacità del nostro elemento sarebbe **impostata di conseguenza** a 0.4.

Se fin qui il ragionamento ti sembra ostico, non temere, tutto si sarà più chiaro nell'esempio qui sotto.

L'Esempio

Immaginiamo di voler semplicemente ruotare un elemento allo scrolling decidendo lo stato della rotazione proprio in base a quest'ultimo: il codice qui sotto ci farà vedere un modo semplice e veloce per raggiungere il nostro obiettivo.

L'HTML

Il codice mark-up della pagina sarà davvero essenziale:

Come puoi notare, andremo a creare solo un div contenente una scritta, nulla più.

IL CSS

Anche qui non ci sarà nulla di complicato:

Imposteremo solo l'altezza del documento del 300% (rispetto l'altezza della finestra) al fine di testare il nostro piccolo script. Posizioneremo il nostro elemento, in questo caso **.scrooll**, perfettamente centrato nella finestra, inoltre, per evitare che durante lo scrolling esso salga nella pagina, useremo l'attributo *position* impostato a **fixed**.

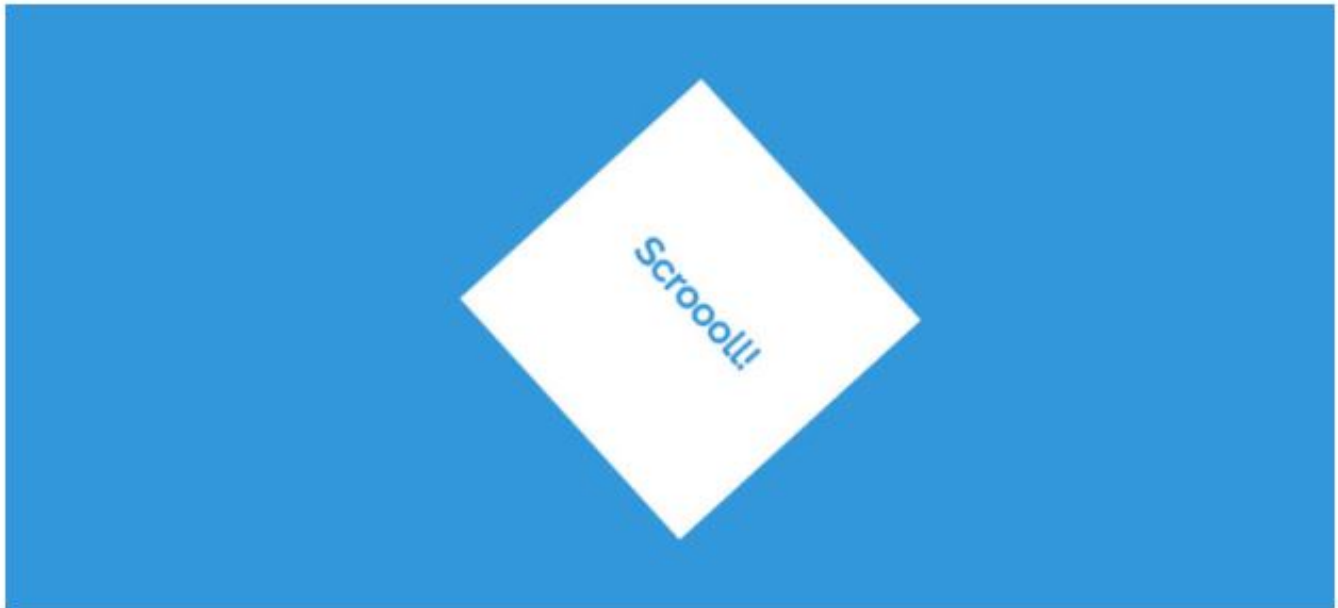
Il codice jQuery

Il codice continua ad essere semplice e pulito anche in questo piccolo script:

Innanzitutto, dichiariamo una variabile contenente l'altezza della finestra, nel nostro caso *winHeight*. Successivamente, andiamo a inizializzare l'evento scroll nella finestra, nel quale troveremo altre variabili molto interessanti, ovvero:

- **scrTop**: divide lo `scrollTop()` del documento effettuato dall'utente per l'altezza della finestra, in modo da poter calcolare lo stato corrente dell'animazione;
- **scrTopFixed**: serve per risolvere l'eccessiva gestione dei numeri decimali di JavaScript; grazie al metodo *toFixed()* potremmo approssimare infatti l'output a due numeri dopo la virgola;
- **scrTransform**: servirà per calcolare il valore da assegnare alla rotazione. Come detto sopra, il valore di `scrTop` avrà una forma simile a "0.9", "0.475", ecc. ma visto che gli step principali di una rotazione si possono misurare in multipli di 90 (es. 180, 270, 360), moltiplicheremo `scrTopFixed` proprio per quest'ultimo numero.

Ora non resta che applicare concretamente il valore alla proprietà *css transform*, al fine di raggiungere l'effetto desiderato, visibile nell'immagine appena sotto.



Tutto qui?

Ti sembra strano che sia così semplice, vero? Avrei potuto parlare per ore su come cambiare l'animazione, o quantomeno la direzione di questa durante lo scroll, ma non è questo lo scopo di questo breve tutorial. In verità, spesso utilizziamo molti script jQuery che non fanno altro che **appesantire** la pagina web, al fine di raggiungere risultati spesso facilmente ottenibili in modi parecchio più basilari.

Come sempre, puoi scaricare [il sorgente dell'esempio da qui](#) e visionare lo stesso [in live demo qui](#). Inoltre, se ti va puoi lasciarmi un feedback e se l'argomento ottimizzazione è di interesse comune, potremmo approfondirlo insieme in un prossimo futuro. Per ora, ti auguro una buona giornata e buon coding!