

# GSAP: creiamo un login animato

Le animazioni sono all'ordine del giorno nel web odierno. Purtroppo, però, sempre più spesso ci troviamo a combattere con la compatibilità cross-browser che, specialmente su browser come IE9 e precedenti, non è sempre assicurata.

Difatti per i suddetti, le trasformazioni CSS sono davvero complesse da realizzare, richiedendo spesso e volentieri l'utilizzo di lunghe stringhe di codice. Tuttavia, se le trasformazioni "statiche" sembrano difficili, le animazioni lo sono ancora di più, in quanto bisognerà creare funzioni macchinose con jQuery o JavaScript che sia.

Dì pure addio a tutti questi problemi: il framework che ti presenterò è semplice, performante e perfettamente cross-browser.

## Dietro le quinte

GSAP (questo è il nome del framework), utilizza un meccanismo molto diverso da quello usato in jQuery per gestire le animazioni. Infatti, questo framework utilizza un metodo JavaScript chiamato *requestAnimationFrame* che assicura una resa delle animazioni davvero fluida (60 FPS). Questo metodo è l'evoluzione di una vecchia tecnica che prevedeva l'utilizzo di un *setInterval* per impostare in modo "manuale" la resa dell'animazione.

## Un po' di sintassi

Ora che abbiamo visto all'incirca come funziona, buttiamoci a capofitto nella comprensione della sintassi che, per chi è abituato a quella di jQuery, potrebbe risultare un po' ostica, anche se estremamente semplice.

Nell'esempio sopra, **item** è un elemento qualsiasi — è possibile usare anche `$(item)`: il numero "1" rappresenta la durata dell'animazione espressa in secondi, mentre il contenuto tra le parentesi graffe rappresenta a destra la proprietà che vogliamo animare (in questo caso "left") e a sinistra il valore che vogliamo attribuire a quella proprietà (100px).

Come vedi, non è nulla di complesso, ma si può fare ancora di meglio!

## Il plugin jQuery

Hai capito benissimo. I ragazzi di [GreenSock](#) hanno pensato bene di rendere il tutto più fruibile andando a creare un plugin che modifica radicalmente il metodo `.animate()`, senza però alterarne la sintassi. Ciò comporta la possibilità di scrivere qualcosa del tipo:

Difatti **GSAP**, così come il suo plugin, supporta anche le trasformazioni 2D e 3D (come skew, rotate, scale ecc...).

Ma ora basta con la teoria e passiamo ad un caso pratico.

## Un login animato

Vediamo insieme come costruire un sistema di login animato per un eventuale sito web di qualsiasi genere. Ovviamente il codice preso in esempio sarà fittizio e adeguato ad un piccolo esempio; starà a te e/o al tuo sviluppatore integrarlo con PHP ed Ajax.

## L'HTML

Vediamo qui di seguito come andremo ad impostare la pagina:

Come puoi notare, il markup della pagina è davvero semplice. Il primo script è quello necessario per includere jQuery nel nostro progetto, il secondo è invece fornito dal **CDN** di Cloudflare e ci servirà per usare la versione (in questo caso quella massimizzata) di GSAP.

Il terzo script andrà ad inserire il plugin **jquery.gsap.min**, mentre l'ultimo sarà quello da noi creato per animare il tutto.

Passiamo adesso al contenuto vero e proprio.

L'elemento `.login` sarà il form attraverso il quale sarà possibile effettuare l'accesso, mentre il div `.wrapper` e il div `.log-into` conterranno rispettivamente gli elementi input e la foto dell'utente (quest'ultima usata come background-image del div `.john`).

Infine, lo span `.loader` servirà come progress bar per il processo di login.

## Il codice CSS

Per motivi di spazio, non includerò in questa descrizione il codice a fine decorativo, fruibile comunque attraverso la demo scaricabile, ma focalizzerò l'attenzione sui punti necessari per creare la base delle animazioni:

In questo esempio, il CSS verrà usato solo per definire lo stato iniziale dell'animazione. In particolare scaleremo a 0 la classe `.login`, assegnando anche l'attributo **perspective** per permetterci di ottenere l'effetto 3D in seguito a una rotazione che possiamo trovare in `input` e in `.log-into`. In quest'ultimo elemento possiamo anche trovare un'altra importante proprietà: **transform-origin**. Questa serve per spostare l'origine dell'animazione, normalmente

impostata al centro dell'elemento. Tuttavia la piena comprensione di questa proprietà è particolarmente ostica, ragion per cui ti consiglio di guardare la demo per una maggiore comprensione.

Ora invece passiamo al vero centro dell'azione: il nostro script jQuery.

## Lo script jQuery

Innanzitutto partiamo con l'assegnazione di alcune variabili che ci torneranno utili:

Le variabili da *input* a *submit* rappresentano dei semplici elementi, mentre ***animTime***, *inputDel* e *initSubDel* esprimono dei valori numerici che hanno a che fare con le animazioni. In particolare "animTime" esprime il tempo dell'animazione espresso in millisecondi, "inputDel" rappresenta il ritardo tra la comparsa di un input e l'altro, infine "initSubDel" esprime il tempo che intercorrerà tra tutte le altre animazioni.

Purtroppo le variabili da sole non fanno molto, quindi ci servirà del codice per far comparire il form iniziale:

Con l'ausilio di queste poche righe potremo mostrare il nostro form con un'elegantissima animazione. Innanzitutto animiamo il form vero e proprio grazie al metodo **.animate()** che, come già detto, è stato esteso e potenziato da GSAP.

Successivamente andiamo ad assegnare l'evento **.each()** a ogni elemento input. Grazie alla funzione contenuta all'interno, estraiamo l'indice di ogni input e assegniamogli un ritardo dell'animazione pari all'indice dell'elemento moltiplicato per il valore della variabile inputDel. In breve, questa funzione ci permetterà di ruotare gli input "uno alla volta", come mostrato nell'esempio.

Ora il codice inizia a farsi leggermente complesso:

Al click dell'input *.submit*, facciamo partire tutte le animazioni necessarie per completare il processo di login. Come prima cosa, animiamo il div *.wrap* per permettergli di lasciare spazio alla foto e alla progress bar. Fatto ciò, animiamo questi ultimi elementi, rispettivamente scalando il div *.photo* e allargando lo span *.loader*.

Ora nascondiamo la foto, portando il valore della trasformazione **scale** nuovamente a zero, e animiamo lo span portandolo ad avere un'altezza pari a quella dell'intero form, modificando il margin-top per riempire l'intera porzione di schermo. Contemporaneamente a ciò, inseriamo una scritta che ci avviserà della riuscita del tentativo.

Mi rendo benissimo conto che l'effetto del codice appena scritto possa risultare difficile, specialmente per i neofiti, motivo per cui invito comunque a provare la demo qui sotto, analizzandola in parallelo con lo script.

### Qualche piccola nota

Anche l'articolo di oggi è finito e ci terrei a farti notare alcune cose. Innanzitutto, il codice come al solito è **estramamente ottimizzabile**. Ad esempio, al posto dei ritardi assegnati via jQuery è possibile utilizzare il metodo nativo JavaScript `setInterval()`, e così per molte altre cose. Difatti questo articolo non vuole darti un sistema pronto e veloce da integrare nel tuo sito web, ma vuole essere piuttosto una semplice fonte di ispirazione. Come sempre, puoi scaricare il codice sul quale abbiamo lavorato [direttamente da qui!](#)

Alla prossima!