

## 4 jQuery tricks che non potete non conoscere

jQuery. Un framework javascript talmente potente e popolare che equivale praticamente a Zend per il PHP. Sempre più siti web lo utilizzano per la sua capacità di essere sintetico ma altamente funzionale, e non se ne può più fare a meno.

Ma andiamo a vedere cosa è davvero capace di fare questo framework in poche linee di codice con questa breve ma intensa guida ai "Best 4" tips di jQuery.

Prima di passare ai tutorial, ricordiamoci di implementare jQuery dentro i tag head del nostro documento nel seguente modo:

### 1. Creare div a tutto schermo

Non volete o non potete creare div della grandezza della finestra del browser assegnando width: 100%; e height: 100%; usando le semplici proprietà CSS?

Nessun problema, jQuery vi verrà incontro come nell'esempio che segue.

**HTML**

**CSS**

**jQuery**

Semplice vero?

Con **\$(window).height()** e **\$(window).width()**, grazie al metodo **.css**, modifichiamo dinamicamente le proprietà **width** e **height** dell'elemento desiderato e le rendiamo grandi quanto la finestra, rappresentata dall'elemento **window**, del browser.

### 2. Centrare un elemento

Dì la verità. Anche tu, *almeno una volta*, hai avuto difficoltà a centrare un elemento di cui non conoscevi a priori le dimensioni usando semplicemente CSS. Fortunatamente con jQuery è tutto più semplice. Segui l'esempio qui sotto.

### HTML

### CSS

### jQuery

Come vedi, anche in questo esempio il codice è piuttosto basilare. Consideriamo ad esempio la proprietà **left** : l'abbiamo impostata in modo da calcolare la metà della larghezza della finestra, sottraendo a quest'ultima la metà della larghezza del nostro elemento, in modo da spostarla a sinistra della quantità di spazio voluto, al fine di effettuare il centramento.

## 3. Cambiare attributi CSS allo scroll

Finora abbiamo trattato tutti argomenti che con CSS, avvalendosi di qualche piccolo trucco, era comunque possibile realizzare , anche se, come hai potuto notare, jQuery ci ha semplificato di molto la vita.

Ora invece, passiamo a parlare di qualcosa che non è assolutamente possibile con l'utilizzo di CSS, per via di regole più conformi a un vero linguaggio di programmazione.

L'avete intuito dal titolo del paragrafo, scommetto: stiamo parlando di cambiare gli attributi di un determinato elemento allo scrolling della pagina: e ora, capiamo come fare. Per l'esempio seguente utilizzeremo e modificheremo il codice dei tips 1 e 2.

### HTML

### CSS

### jQuery

Adesso il codice inizia a farsi un po' più complesso, ma neanche tanto.

La prima parte ci è ovviamente familiare, in quanto abbiamo utilizzato gli stessi codici di sopra, anche se con un piccolo cambiamento.

Come potete osservare, all'elemento **.full** abbiamo aggiunto all'**height** un valore di **100px**; per

appunto abilitare lo **scrolling**.

Passiamo ora al commento vero e proprio della nuova porzione di codice jQuery. Innanzitutto abbiamo definito una **variabile, scrolled**, che all'inizio ha un valore **booleano** impostato su **false**.

Fatto ciò, abbiamo aggiunto un nuovo evento, **.scroll**, con all'interno uno statuto **if/else** che ci permette di controllare se lo **scrolling (scrollTop)** è maggiore di **50px**. Nel caso si verifichi questa condizione, se lo **scrolling** avviene davvero, la variabile passerà a **true** e manterrà questo stato sin quando lo **scrolling** è maggiore di **50px**, permettendo al codice di procedere con l'animazione.

Al contrario invece, se lo scrolling non avverrà, la variabile `scrolled` tornerà ad avere come valore `false`, permettendo di eseguire l'animazione contraria che riporterà il nostro `div` alle **condizioni originali**.

Permettetemi una piccola nota. Spesso, nella praticità di tutti i giorni, si desiderano animare attributi come **background-color** ecc... Per **animare i valori non numerici**, bisognerà fare affidamento ad un progetto parallelo a jQuery chiamato **jQuery UI**, che si può inserire nel proprio progetto tramite inclusione nei tag head, esattamente come per jQuery, ovvero:

## 4. Animare lo scroll tra i div con i links anchor

Sei arrivato fin qui eh? I miei complimenti! Ora però, per concludere in bellezza, andremo a fare qualcosa di un po' più complesso, ma visivamente davvero accattivante. Prendiamo in considerazione i **links anchor**.

Come? Non sai che cosa sono? Ma dai, quei **link** sui quali i **click** portano a un determinato **punto della pagina**.

Se hai capito di cosa stiamo parlando, saprai anche che in realtà nello "**stacco**" tra il **punto di partenza** della pagina e quello di **destinazione** non è presente alcuna **animazione**.

Ebbene, andremo a vedere come **aggiungere** quest'ultima, con l'inserimento di appena **7 righe di codice jQuery**.

Anche in questo caso, riproporremo il codice già visto sopra.

### HTML

### CSS

### jQuery

Come possiamo notare, a parte l'HTML dove si sono aggiunti sia un **link** con **anchor**, sia un nuovo **div** di classe **.full**, associato alla classe **.red** per poterne modificare il colore di background, e il **CSS**, dove si è definita la classe suddetta, il codice **jQuery** ha subito una notevole **aggiunta**.

Innanzitutto, quando avviene l'**evento click sull'elemento a**, si instancia uno **scrollTop** per l'elemento che ha l'**ID** uguale **all'attributo href** dell'anchor (tutto questo avviene grazie a **scrollTop: \$( \$(this).attr('href') )** ).

Successivamente, si trova la posizione dell'elemento desiderato grazie al metodo **offset()**, e si mostra quest'ultimo facendo allineare il margine superiore dell'elemento al margine effettivo della finestra, grazie a **.top**.

Trovate l'esempio "in azione" di questo breve tutorial [questo link](#).

L'articolo termina qui. Come sempre, se avete dubbi o doveste riscontrare degli errori nel codice vi invito a farmelo sapere nei commenti.

Avete inoltre la possibilità di scaricare tutto il codice sorgente dal [questo file](#).

Alla prossima!