

Pillola CSS: come trasformare gli elementi con CSS3?

Fino a pochi anni fa l'unico modo per aggiungere delle piccole animazioni ai vari elementi di un sito web era quello di utilizzare JavaScript. Con l'introduzione di CSS3 le possibilità sono aumentate, permettendoci di applicare semplici effetti senza utilizzare un altro linguaggio di scripting. Vediamo come!

Ci sono quattro diverse specifiche CSS3 che consentono di applicare effetti animati o vere e proprie animazioni semplicemente utilizzando i CSS:

- CSS Transforms, 2D e 3D
- CSS Transitions
- CSS Animations

Il supporto a queste diverse specifiche è buono per i browser moderni (IE9+, Firefox, Chrome, Safari, Opera) ma, come deve essere sempre quando si parla di miglorie visuali, è sempre dovere del web designer accertarsi che l'utilizzo di queste tecniche avanzate non comprometta l'usabilità e la fruibilità del sito.

In questo articolo analizzeremo i CSS Transforms.

CSS transforms: due o tre dimensioni?

L'obiettivo di questa specifica è quello di permettere di modificare la forma e le dimensioni di un elemento. La specifiche relative a questa regola sono due: una per le trasformazioni bidimensionali e una per le trasformazioni tridimensionali. In questo articolo prenderemo in esame soltanto la prima specifica, mentre la seconda sarà trattata in un articolo successivo.

2D Transforms

Le trasformazioni bidimensionali applicabili ad un elemento sono quattro:

- Rotazione (**rotate**)
- Traslazione (**translate**)
- Ridimensionamento (**scale**)
- Inclinazione (**skewX**, **skewY**)

Tutte queste trasformazioni vengono effettuate tramite la regola CSS transform la cui sintassi è la seguente:

Quindi per applicare una rotazione avrei una regola del genere:

Purtroppo non tutti i browser supportano la regola transform standard, quindi è necessario aggiungere anche le regole con i prefissi proprietari, avendo cura di inserirle prima della regola standard. Quindi il codice completo sarà:

Ovviamente scrivere tutte queste regole ogni volta è un lavoro sfiancante: ecco perché esistono tool appositi come [CSS3please](#) o [Prefixr](#) che ci consentono di semplificare il processo. Un mio consiglio personale è quello di cercare anche plugin per il proprio editor preferito (ad esempio per SublimeText2 c'è un package che sfrutta Prefixr).

Nel codice dell'articolo io scriverò soltanto la regola standard, ma resta sottinteso che sono necessarie anche le regole con i vari prefissi per una buona visualizzazione su tutti i browser che supportano i CSS3.

Specificato questo, vediamo subito quali sono le sintassi per ogni tipo di trasformazione.

Rotazione

Come puoi intuire dal nome, la regola *rotate* permette di ruotare un elemento. L'unità di misura utilizzata per questa trasformazione è il grado (*deg*). Le rotazioni avvengono in senso orario, quindi una rotazione di 90 gradi inclinerà l'elemento a destra, una rotazione di 180 gradi lo capovolgerà, una rotazione di 270 gradi lo farà risultare capovolto a sinistra ed infine una rotazione di 360 gradi lo riporterà nella posizione originaria. Per avere delle rotazioni in senso antiorario basta utilizzare i numeri negativi, quindi -90 gradi inclinerà l'elemento a sinistra.

Facciamo subito un esempio: crea un documento HTML con un div con classe *box*. Dopodiché applica il seguente codice CSS:

Questa semplice regola crea un quadrato blu inclinato a destra di 45 gradi. Qui c'è un [esempio](#) in cui puoi sperimentare l'effetto della rotazione se un elemento più complesso.

Traslazione

Questa trasformazione permette di spostare un elemento rispetto alla sua posizione originaria sopra, sotto, a destra ed a sinistra. Ne esistono tre versioni:

- `translate(valoreX, valoreY)`
- `translateX(valore)`

- `translateY(valore)`

La prima permette di effettuare uno spostamento sia in orizzontale (X) che in verticale (Y), le altre due invece operano solo su una componente. Il parametro passato in ingresso può essere una qualsiasi delle unità di misura valide per i CSS (px, em, %, pt, ecc.). Ricorda che quando viene effettuata una traslazione, il flusso della pagina (e quindi la posizione degli altri elementi) non viene modificato.

Quindi la funzione `translate(x,y)` sposta un elemento del valore x a partire dal lato sinistro e del valore y a partire dall'alto:

Invece, per spostare un elemento solo orizzontalmente o verticalmente possiamo usare le funzioni `translateX(x)` o `translateY(y)`:

Qui c'è un [esempio in cui viene applicata la trasformazione `translateX`](#).

Ridimensionamento

La funzione `scale(x,y)` ci permette di ridimensionare un elemento della pagina rispetto alle sue attuali dimensioni. Prende in ingresso un fattore moltiplicativo: questo significa che se passiamo in input il valore 1, l'elemento non verrà ridimensionato; se passiamo il valore 2, le dimensioni verranno raddoppiate, mentre se passiamo 0.5 verranno dimezzate.

Questa regola CSS raddoppierà le dimensioni del box, non modificando la posizione degli elementi circostanti. Come per la traslazione esistono anche le versioni che influenzano solo la dimensione orizzontale o verticale: `scaleX(x)` e `scaleY(y)`.

Anche qui ho preparato un [esempio sul ridimensionamento](#).

Inclinazione

L'ultima funzione di trasformazione che esamineremo è quella che modifica l'inclinazione di un elemento: `skew(x, y)`, dove x e y sono i gradi dell'inclinazione desiderata. Ovviamente la x gestisce l'inclinazione orizzontale, mentre la y quella verticale. Ad esempio:

Come per le altre trasformazioni, esistono le versioni `skewX(x)` (di cui puoi visualizzare [un esempio](#) e `skewY(y)`).

A mio parere personale questa trasformazione è quella più *delicata* e deve essere utilizzata con criterio e gusto, dato che l'effetto che provoca è molto particolare e non adatto a tutte le situazioni.

Un esempio e conclusioni

In questo articolo abbiamo visto come modellare gli elementi con le trasformazioni introdotte in CSS3: come per tutti gli effetti visuali bisogna sempre bilanciare usabilità e bellezza estetica e ricordarsi che un buon web designer sa sempre quando fermarsi (e frenare il cliente: "Voglio l'ombra gialla, e poi lo voglio animato, che ruota, e poi fallo pulsare ogni tre secondi!").

Ecco un semplice esempio di come queste trasformazioni possono migliorare l'usabilità di una pagina: nella [pagina c'è un elenco di dolci](#) e, al passaggio del mouse, quello posizionato sotto al cursore viene messo in primo piano, sfruttando la trasformazione scale, un pizzico di box-shadow e le **CSS3 Transition**. Cosa sono le CSS3 Transition? Questo lo vedremo in un prossimo articolo!