CSS3: COME ANIMARE LE TUE IMMAGINI

Il tutorial di oggi è davvero semplice e nello stesso tempo molto efficace: vedremo insieme come creare una **semplice animazione** che permetta di rendere **dinamiche** le nostre **immagini,** inserendo **effetti zoom** e una o più **didascalie** su di esse.

è un ottimo metodo da utilizzare se hai integrato nel tuo progetto web una galleria di immagini o di prodotti ma anche se, più semplicemente, vuoi donare a un'immagine qualsiasi sul tuo sito un **effetto semplice** e **accattivante.**

Inoltre, come sempre, non utilizzeremo né il puro Javascript, né il Jquery, ma solo **CSS,** che si occuperà sia di dare lo stile, sia di gestire il comportamento dell'immagine/delle immagini.

Utilizzerò gli strumenti di cui mi sono servito anche nei tutorial precedenti e che, se hai letto gli articoli , non avrai difficoltà a comprendere. Per tutti gli altri, cercherò, ovviamente, di descrivere bene i diversi passaggi e di spiegare nel dettaglio gli strumenti usati.

Ringrazio la mia amica e collega Agnieszka Czerwinska che mi ha aiutato nella redazione del seguente tutorial.

Allora, iniziamo!

HTML&CSS

In questo esempio utilizzeremo una sola immagine, per cui il codice HTML si limiterà a contenere un'ancora e l'immagine stessa.

Fate attenzione con le immagini! Ricordate che su internet potete utilizzare solo immagini di cui siete proprietari o che hanno licenza free, quindi fate attenzione a cosa usate! Per questo esempio utilizzerò delle immagini free prese da Wikimedia.

L'unica cosa da notare nell'html, è l'attributo *"title"* che darò all'ancora. Questo attributo è presente praticamente da sempre, tuttavia è molto spesso tralasciato. Il compito dell'attributo *title* è creare un *tooltip* che viene visualizzato sull'elemento a cui questo è stato attribuito se si lascia il mouse sopra di esso.

Ne avremo bisogno, poiché il testo della didascalia animata che verrà visualizzato, sarà quello di questo attributo.

Quindi:

HTML:

Passiamo subito, quindi, al CSS. Come detto in precedenza, qui verrà svolta (come sempre! :D) la magia:

Qui niente di speciale, abbiamo assegnato qualche semplice proprietà all'elemento ancora per rimuovere le sottolineature e per coprire gli eventuali ***overflow.***

Andiamo avanti:

Qui utilizziamo gli **pseudoelementi** *:before* e *:after.*

Il loro scopo è creare degli elementi in maniera **"dinamica":** tramite questi pseudoelementi *(:before* e *:after)* è come se si scrivesse del contenuto (o dei nuovi tag html) prima e dopo l'elemento al quale vengono associati.

Per questo vengono chiamati "pseudoelementi": perché ci sono nuovi elementi nella pagina, ma in realtà non sono scritti nel codice HTML… vengono creati con il codice css, tramite la definizione di *:before* e *:after.*

Nel nostro caso lo scopo di *before* e *after* è creare sia uno sfondo per la nostra didascalia che una patina che scurisca leggermente la nostra immagine quando ci passiamo sopra. Per fare ciò, invece di utilizzare la proprietà **Opacity,** abbiamo usato l'rgba (che settiamo a 0.8).

Questo **escamotage** è necessario poiché se usassimo **"opacity"** questo cambierebbe l'opacità di tutti gli altri elementi collegati all'ancora. Da notare, inoltre, anche la proprietà ***"content":*** per farla breve, questa proprietà è essenziale per il corretto funzionamento dei due **pseudoelementi** che, senza di esso, non funzionerebbe.

A questo elemento può essere associato pressoché di tutto, dal nulla a degli attributi, i quali verrebbero quindi mostrati insieme al pseudoelemento a cui il **content** è associato.

Come nel nostro caso. Al ***"before"*** abbiamo associato solo le virgolette (in pratica abbiamo lasciato vuoto), questo perché ci servirà solo come elemento per "patinare" la nostra immagine; mentre nell***'after*** gli abbiamo assegnato ***"attr(title)",*** cioè gli abbiamo detto "mostrami l'elemento ***title*** dell'ancora".

Adesso passiamo alle ***transizioni*** e alle **trasformazioni:**

Completiamo, in questo modo, il nostro tutorial: agli pseudoelementi e all'immagine stessa assegniamo le proprietà di **transizione.**

Ci serviranno, in quanto, nel momento dello zoom dell'immagine, vogliamo che il tutto avvenga tramite **un'animazione liscia** e **pulita.** Senza di questi, semplicemente al passaggio del mouse otterremmo un'immagine zoomata senza alcun effetto.

Anche l'animazione della didascalia ne risulterebbe compromessa; Con le proprietà ***transform,*** invece, ci occupiamo dello zoom stesso, manipolando la proprietà “scale”. Semplicemente, senza di questo, non ci sarebbe nessuno zoom. Ricorda di mettere sempre i **prefissi proprietari,** per aumentare la **compatibilità** di ciò che crei nei vari browser.

Questo perché, essendo una tecnologia relativamente nuova, i browser non si sono ancora uniformati. In particolar modo, non dimenticare mai i prefissi per i browser ***“webkit”*** e **microsoft,** giacché le versioni precedenti di Internet Explorer la supportano solo in parte o addirittura non la supportano.

Infine, con gli **hover finali,** attiviamo le **transizioni** che abbiamo creato prima. And that's it!

Questo è ciò che dovresti aver ottenuto. Il codice completo è scaricabile da questo link.

Conclusioni

Di sicuro, come avrai notato, non è niente di difficile, sia nella comprensione che nella messa in pratica. Il CSS mette a disposizione davvero tanti elementi, proprietà e valori per personalizzare come più ci piace le nostre pagine, quindi perché non sfruttarle, anche per fare cose semplici ma di sicuro effetto?

Per questo motivo, nel file allegato al tutorial, abbiamo creato una griglia con lo stesso effetto, ma personalizzato in maniera differente, dategli una occhiata. ;)

Spero vivamente che questo tutorial sia stato di vostro gradimento, fatemi sapere nei commenti se l'avete usato per voi o se avete idee per arricchirlo. Alla prossima!