

## Creare un effetto hover animato sulle immagini con CSS3

*giovedì, 05 gennaio 2017*

Tra gli elementi che troviamo in praticamente tutti i siti, ci sono senza dubbio le immagini, utilizzate per dare una gradevolezza alle pagine e molto spesso utilizzate come aree su cui applicare link. Ma altrettanto spesso, le immagini che presentano dei link risultano statiche al passaggio del mouse su di loro, eccezion fatta per il cursore del mouse stesso che cambia quando si trova su di un link. Ma perché non aumentare l'impatto grafico delle nostre pagine con qualche animazione ad effetto?



### Sommario

Perché non utilizzare Javascript o jQuery?

Passo 1: HTML

Passo 2: Lo stile

## Perché non utilizzare Javascript o jQuery?

Quanto appena descritto, potrebbe essere implementato con poche righe di codice utilizzando qualche script Javascript o qualche plugin jQuery, navigando in rete se ne trovano a centinaia. Ma questa soluzione, appesantirebbe le tue pagine non solo per quanto riguarda il peso, visto che dovrebbero includere, caricare e richiamare gli script necessari, ma anche in termini di tempi di caricamento, rallentato dall'esecuzione degli script stessi.

Senza contare che in alcuni casi, l'esecuzione di Javascript potrebbe essere disattivata o bloccata da errori, impedendo di fatto di godersi gli effetti grafici.

Dovendo sempre prestare attenzione a velocità, performance, esperienza utente, e chi più ne ha più ne metta, è necessario trovare una soluzione che ti garantisca lo stesso risultato ma eliminando quanto più possibile gli aspetti negativi.

La risposta al problema è CSS3, vediamo come implementare l'animazione.

## Passo 1: HTML

La prima cosa da fare, è scrivere il markup. Niente di complicato, semplicemente inseriamo la nostra immagine, il nostro anchor text ed un elemento che li racchiuda:

```
<figure>  
  <img src="" alt="" title="" />  
  <a href="#"></a>  
</figure>
```

## Passo 2: Lo stile

Nel caso in esempio, abbiamo scelto come contenitore l'elemento `<figure>`, a cui andiamo ad applicare poche e semplici regole di CSS, di cui analizziamo nel dettaglio le parti più importanti:

```
figure {  
  background: #700877;  
  display: inline-block;  
  overflow: hidden;  
  position: relative;  
}
```

Applichiamo una transizione agli pseudo elementi collegati al contenitore (`::before` e `::after`) e all'immagine:

```
figure img,  
figure:before,  
figure:after {  
  transition: all 0.4s ease;  
  -webkit-transition: all 0.4s ease;  
}
```

Applichiamo due pseudo elementi al contenitore `<figure>`, i quali avranno opacità nulla (le rendiamo così inizialmente invisibili) e posizionamento assoluto:

```
figure:before,  
figure:after {  
  background-color: #fff;  
  content: '';  
  opacity: 0;  
  position: absolute;  
  z-index: 1;  
  transform: translate( -50%, -50% );  
  -webkit-transform: translate( -50%, -50% );  
}
```

Diamo una forma al primo pseudo elemento, che prenderà le sembianza di un piccolo tratto orizzontale:

```
figure:before {  
  left: 0;  
  height: 1px;  
  width: 60px;  
}
```

Mentre il secondo pseudo elemento, diventerà un piccolo tratto verticale di uguali dimensioni

rispetto al precedente:

```
figure:after {  
  height: 60px;  
  top: 100%;  
  width: 1px;  
}
```

Facciamo in modo che l'anchor text si espanda e occupi tutto lo spazio a disposizione fino a coprire il contenitore <figure>:

```
figure a {  
  bottom: 0;  
  left: 0;  
  position: absolute;  
  right: 0;  
  top: 0;  
  z-index: 1;  
}
```

All'evento :hover sull'elemento immagine (quando il mouse transita sull'immagine), applichiamo una trasformazione che aumenta le dimensioni dell'elemento del 20% e la rende trasparente al 50% (per lasciar intravedere il background precedentemente applicato al contenitore):

```
figure:hover img {  
  opacity: 0.5;  
  transform: scale( 1.2 );  
}
```

Muoviamo gli pseudo elementi, così che si posizionino esattamente al centro dell'immagine, creando una croce:

```
figure:hover:before,  
figure:hover:after {  
  left: 50%;  
  opacity: 1;  
  top: 50%;  
}
```

Ed ecco l'effetto finale, davvero molto molto bello ed elegante, guardalo su JSFiddle.

[Guarda la demo su JSFiddle](#)