3 Impresionantes Efectos De Animacion CSS Para Atraer Usuarios

Un hermoso diseño visual en nuestro sitio web ya no es suficiente. Las **animaciones en sus diseños** pueden proporcionar un enfoque, una atención directa, y crea una experiencia impactante a los visitantes. Para esto tenemos que aplicar **efectos de animación CSS** para lograr lo anteriormente mencionado.

Las **transiciones CSS** brindan la oportunidad a los diseñadores con conocimientos limitados de código mejorar de gran manera sus proyectos con **efectos de movimientos impresionantes,** esto ayudara a interactuar a los usuarios como nunca antes. Para esto vamos a empezar con algo sencillo: **pasar de una pantalla a otra …**

Técnica Efecto Deslizar

Para este ejemplo solo necesitamos de un editor de texto plano y un navegador web; con el primero incrustamos el código, el segundo probaremos el ejemplo :D Empecemos con el diseño del ejemplo:

[html]

<div class="screen">
<div class="slider">

</div>
</div>

[/html]

Usted necesitará un «**screen**» y luego un «**slider**» dentro de la pantalla. El control deslizante se extiende más allá del borde de la pantalla, y mantiene las imágenes en la maqueta.

Para lograr esto, es necesario asegurarse de que agrega **overflow:hidden** al div con la clase .screen. El **CSS** se verá justo de la siguiente manera:

[css]

```
.screen {
overflow:hidden;
width:320px;
height:568px;
}
.slider {
position:relative;
float:left;
height:568px;
width:700px;
left:0;
-webkit-transition:all 0.5s ease-in-out;
}
.slider img {
position:relative;
float:left;
height:568px;
width:320px;
}
.screen:hover .slider {
left:-320px;
}
```

[/css]

La ultima sentencia en del **CSS** es la que controla la posición del control deslizante, se mueve el div **.slider** hacia la

izquierda por 320px revelando la segunda imagen.

Con un poco de creatividad, puedes usar esta técnica simple y crear algunas animaciones ingeniosas. Las cosas empiezan a ponerse interesantes cuando se combinan los efectos. Por ejemplo: Se ha recreado el efecto de **Twitter** «**swipe-to-reveal profile**» de su aplicación móvil usando un enfoque de '**slider**' muy similar.

El siguiente será el código html:

[html]

```
&lt;div class=&quot;screen&quot;&gt;
&lt;div class=&quot;header&quot;&gt;
&lt;h4&gt;My Profile&lt;/h4&gt;
&lt;/div&gt;
&lt;!– positioned absolutely in the background –&gt;
&lt;img class=&quot;background&quot;
src=&quot;https://pbs.twimg.com/profile_banners/45394577/1384289888/web&quot;&gt;

&lt;!– This slider moves on hover –&gt;
&lt;div class=&quot;slider&quot;&gt;
&lt;!– Section 1 –&gt;
&lt;div class=&quot;header1&quot;&gt;
&lt;img class=&quot;profilepic&quot;
src=&quot;https://pbs.twimg.com/profile_images/378800000297360664/96d9e7224ec40be585466172a3785249.png&quot;&gt;
&lt;h3&gt; William Newton&lt;/h3&gt;

&lt;/div&gt;
&lt;!– Section 2 –&gt;
&lt;div class=&quot;header2&quot;&gt;
&lt;p&gt;Adventurer, Product Designer @DJZ, Building
```

@CrossfaderApp, Check out my work at http://williamnewton.co;
<p>San Francisco, CA · williamnewton.co/blog</p>
</div>
</div>
</div>

[/html]

Luego aplicamos el codigo CSS para dar el efecto y la apariencia:

[css]

```
.screen:hover .background {
opacity:0.5;
}
.slider:hover {
left:-320px;
-webkit-transition: all 0.5s ease-in-out;
}
body {
Helvetica Neue', Helvetica, Arial, sans-serif;
}
.header {
width:100%;
height:60px;
background:#56B8DE;
text-align:center;
float:left;
color:#fff;
}
.profilepic {
width:64px;
height:64px;
border-radius:50%;
margin-top:20px;
border:2px solid #fff;
box-shadow:0px 1px 2px rgba(0, 0, 0, 0.5);
```

```
}
h3 {
margin-top:0;
text-shadow:0px 1px 3px rgba(0, 0, 0, 0.75);
}
.screen {
margin:0 auto;
margin-top:24px;
width:320px;
height:200px;
box-shadow: 0px 2px 6px #999;
position:relative;
overflow:hidden;
background:#333;
}
.slider {
width:640px;
float:left;
left:0;
position:relative;
-webkit-transition: all 0.5s ease-in-out;
}
.background {
position:absolute;
width:320px;
height:auto;
z-index:0;
top:60px;
left:0;
-webkit-transition: all 0.5s ease-in-out;
}
.header1, .header2 {
text-align:center;
width:320px;
float:left;
height:140px;
color:#fff;
```

```
position:relative;
}
.header2 {
padding-top:24px;
}
p {
font-size:12px;
text-shadow:0px 1px 2px rgba(0, 0, 0, 0.5);
}
```

[/css]

El resultado sera el siguiente:

Transformaciones 3D

El uso de transformaciones **3D** proporcionan un efecto visual impresionante. Con el uso de la **propiedad -webkit-transform**, podemos tratar el navegador como un **espacio 3D** y hacer algunas animaciones manejando profundidad. En particular iOS7, hace uso de la metáfora «single space» en sus aplicaciones nativas. También, las transformaciones 3D son muy útiles para la creación de efectos «bouncing» o «popping» en las animaciones. El ejemplo se vera de la siguiente forma:

El codigo html seria el siguiente:

[html]

```
&lt;div class=&quot;screen&quot;&gt;
&lt;div class=&quot;header&quot;&gt;
Crosses | Library | New Packs
&lt;/div&gt;
&lt;div class=&quot;cube blue&quot;&gt;
&lt;div class=&quot;title&quot;&gt;
```

```
SOMEBODY THAT U USED TO KNO
&lt;span&gt;GYOTE&lt;/span&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class=&quot;cube red&quot;&gt;
&lt;div class=&quot;title&quot;&gt;
2 MILLION
&lt;span&gt;AVICII&lt;/span&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class=&quot;cube next&quot;&gt;
&lt;div class=&quot;title&quot;&gt;
SHAKE DAT
&lt;span&gt;SAVAGE&lt;/span&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class=&quot;cube nextNext&quot;&gt;
&lt;div class=&quot;title&quot;&gt;
NEXT SONG 0.9 OPACITY
&lt;span&gt;nextsongartist&lt;/span&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;

[/html]

El código CSS seria el siguiente:

[css]

h1, h3 {
color:#666;

}

.screen {
background:#000;
width:556px;
```

```
height:320px;
overflow:hidden;
position:relative;
-webkit-perspective:600px;
}

.cube {
width:200px;
height:200px;
position:absolute;
}

.cube .title {
color:#fff;

font-weight:lighter;
padding:16px;
}

.title span {
display:block;
font-size:12px;
padding-top:8px;
}

.red {
background:#ed4343;
top:50px;
left:278px;
-webkit-transform-origin: right center;
-webkit-transform-style: preserve-3d;
-webkit-transition: -webkit-transform 0.5s ease-in-out;
z-index:50;
}

.blue {
background:#43b7ed;
top:50px;
```

```
left:78px;
-webkit-transform-origin: left center;
-webkit-transform-style: preserve-3d;
-webkit-transition: -webkit-transform 0.5s ease-in-out;
z-index:50;
}

.header {
background:#222222;
position:absolute;
top:-40px;
height:40px;
line-height:40px;
color:#fff;
font-weight:bold;

text-align:center;
width:100%;
opacity:0.8;
-webkit-transition: -webkit-transform 0.5s ease;
}

.screen:hover .red {
-webkit-transform:   rotateY(-70deg)   translateX(150px)
translateY(24px) translateZ(50px);
}
.screen:hover .blue {
-webkit-transform:  rotateY(70deg)  translateX(  -150px  )
translateY(24px) translateZ(50px);
}

.screen:hover .header {
-webkit-transform: translateY(40px);
}

.screen:hover .next {
opacity:1;
margin-top:0px;
```

```
}

.next {
background:#ddd;
top:50px;
left:178px;
-webkit-transform-origin: left center;
-webkit-transform-style: preserve-3d;
-webkit-transition: all 0.6s ease-in-out;
z-index:1;
-webkit-transform: translate3d( -50px, 400px, -800px );
box-shadow:0px 0px 30px 10px #eee;
opacity:0.4;
}

.screen .next {
-webkit-transform: translate3d( 0px, 20px, 0px );
box-shadow:none;
opacity:0;
margin-top:250px;
}

.nextNext {
background:#0f0;
top:50px;
left:178px;
-webkit-transform-origin: left center;
-webkit-transform-style: preserve-3d;
-webkit-transition: all 0.6s ease-in-out;
z-index:100;
-webkit-transform: translate3d( 0px, 600px, 600px );
box-shadow:0px 0px 20px 10px #ed4343;
opacity:0.4;
}

.screen:hover .nextNext {
-webkit-transform: translate3d( 0px, 240px, 0px );
box-shadow:none;
```

```
opacity:0.9;
}

[/css]
```

Usando jQuery Y JavaScript

Hasta ahora sólo hemos visto el **evento hover de CSS para producir animaciones**. Con **jQuery**, podemos usar eventos de click para añadir ó eliminar clases en los elementos. Esto nos da una gran cantidad de flexibilidad para hacer todo tipo de animaciones que deseemos.

A continuación tenemos una función llamada **kaskade**, que se aplica a la clase abierta a cada uno de los 4 elementos de menú en un intervalo de 0.15s. La clase abierta da la opacidad con **opacity:1**; y **left:0**; cuando antes estaban en la **opacity:0**; y **left:-50px**. Esto crea un efecto de interacción en el menú. Tal como se aprecia en la siguiente animación:

El codigo html seria el siguiente:

```
[html]

&lt;div class=&quot;screen&quot;&gt;
&lt;div class=&quot;menu-btn&quot;&gt;&lt;/div&gt;
&lt;div class=&quot;menu-bar&quot;&gt;
&lt;div class=&quot;menu feed&quot;&gt;
&lt;img src=&quot;http://i.imgur.com/DBruDkk.png&quot; /&gt;
&lt;/div&gt;
&lt;div class=&quot;menu radio&quot;&gt;
&lt;img src=&quot;http://i.imgur.com/pIErM3w.png&quot; /&gt;
&lt;/div&gt;
&lt;div class=&quot;menu browse&quot;&gt;
&lt;img src=&quot;http://i.imgur.com/9PQemDg.png&quot; /&gt;
&lt;/div&gt;
&lt;div class=&quot;menu profile&quot;&gt;
```

```
&lt;img src=&quot;http://i.imgur.com/Tgp84Mm.png&quot; /&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;

[/html]
```

El código **CSS** para este ejemplo seria el siguiente:

```
[css]

/* image animating out one at a time */
.menu {height:60px; padding:0px; margin:0px;}

.menu img {
height:60px;
padding:0px;
margin:0px;
width:auto;
opacity:0;
position:relative;
left:-18px;
-webkit-transition: all .5s ease;
}
.menu.open img {
opacity:1;
left:0px;
}
.menu-bar {
position:absolute;
left:-50px;
top:0px;
width:50px;
height:320px;
overflow:hidden;
-webkit-transition: all .6s ease;
opacity:0;
pointer-events:none;
```

```
}
/* toggles outlines to slide in */
.menu-bar.active {
opacity:1;
left:0px;
background:#000;
pointer-events:auto;
}
.menu-bar.active .menu {
-webkit-transition: all .4s ease;
padding:0px;
outline:0px;
}
.menu-bar.active .menu.feed {
border-left:4px solid #e02726;
}
.menu-bar.active .menu.radio {
border-left:4px solid #58bce2;
}
.menu-bar.active .menu.browse {
margin-top:64px;
border-left:4px solid #efe700;
}
.menu-bar.active .menu.profile {
border-left:4px solid #ff91fe;
}
.menu-btn {
position:absolute;
top:0px;
left:0px;
width:50px;
height:50px;
background:url('http://i.imgur.com/MyRm6gv.png');
}
/* Core Background */
.container {
width:556px!important;
```

```
padding:136px 210px 176px 190px;
margin-left:auto;
margin-right:auto;
background:url(&quot;http://i.imgur.com/xtdiQFH.png&quot;);
background-size:100% 100%;
}
.screen {
background:#000;
width:150px;
height:320px;
overflow:hidden;
position:relative;
-webkit-perspective:600px;
margin: 24px auto;
}
```

[/css]

El código Js responsable de la interacción con el usuario es el siguiente:

[js]

```
var redOn = function () {
$(‘.feed’).addClass(‘open’);
}

var blueOn = function () {
$(‘.radio’).addClass(‘open’);
}

var yellowOn = function () {
$(‘.browse’).addClass(‘open’);
}

var pinkOn = function () {
$(‘.profile’).addClass(‘open’);
}
```

```
var kaskade = function () {
setTimeout(redOn, 150);
setTimeout(blueOn, 300);
setTimeout(yellowOn, 450);
setTimeout(pinkOn, 600);
};

var redOff = function () {
$('.feed').removeClass('open');
}

var blueOff = function () {
$('.radio').removeClass('open');
}

var yellowOff = function () {
$('.browse').removeClass('open');
}

var pinkOff = function () {
$('.profile').removeClass('open');
}

var unKaskade = function () {
setTimeout(redOff, 150);
setTimeout(blueOff, 300);
setTimeout(yellowOff, 450);
setTimeout(pinkOff, 600);
};
$(document).ready(function () {

$(document).on('click', &quot;.menu-btn&quot;, function () {
$('.menu-bar').addClass('active');
kaskade();

});

$(document).on('click', &quot;.menu&quot;, function () {
unKaskade();
```

```
setTimeout(function () {
$(‘.menu-bar’).removeClass(‘active’);
}, 1000);
});
});
```

[/js]

Esperamos que sean de tu agrados estos espectaculares efectos de animación para que puedas aplicarlos a tus sitios web.