

[Desplegar](#)

Crear ventanas modales usando solo css y su selector target (/blog/css/crear-ventanas-modales-usando-solo-css-y-su-selector-target)

Publicado el Martes, 18 Marzo 2014 22:27

7736

Twitter 24

in Compartir 2

Me gusta 47

Crear ventanas modales usando solo css y su selector target?

print-1&layout=default&page=1



Hace algún tiempo escribí sobre **como crear ventanas modales** con la ayuda de **Jquery UI** (<http://www.nosolocss.com/blog/jquery/ventanas-emergentes-con-jquery-ui>), y como técnica para salir del paso rápidamente no esta mal, pero no es la mejor manera de hacerlo, mas que nada porque estamos cargando una librería entera en nuestro proyecto para hacer algo que podríamos hacer **solo con CSS**. Ahora podrás pensar que tal vez sea demasiado pesado hacer esto **usando solo CSS**, o que los resultados no van a estar a la altura, pero lo cierto es que con las nuevas características de **CSS3**, se pueden obtener efectos sorprendentes con muy pocas lineas de código, y escasos o nulos conocimientos de programación.

DEMO

(/files/ventanasmodalescss/ventanas_modales_css.html)

| SOURCE

(/files/ventanasmodalescss/ventanas_modales_css.rar)

La parte HTML para crear esta demostración sera algo así:

```

1  <ul>
2    <li><a href="#modal1">DESLIZAR</a></li>
3    <li><a href="#modal2">ROTAR</a></li>
4    <li><a href="#modal3">REDIMENSIONAR</a></li>
5  </ul>
6  <div id="modal1" class="modalmask">
7    <div class="modalbox movedown">
8      <a href="#close" title="Close" class="close">X</a>
9      <h2>DESLIZAR</h2>
10     <p>La ventana modal aparece por arriba y se desliza hasta su p
11     <p>Aquí puedes incluir cualquier cosa como vídeos, mapas, form
12   </div>
13 </div>
14 <div id="modal2" class="modalmask">
15   <div class="modalbox rotate">
16     <a href="#close" title="Close" class="close">X</a>
17     <h2>ROTAR</h2>
18     <p>Usando la propiedad transform de CSS3, podemos hacer que la
19     <p>No hay nada de Javascript, solo unas pocas lineas de CSS.</
20   </div>
21 </div>
22 <div id="modal3" class="modalmask">
23   <div class="modalbox resize">
24     <a href="#close" title="Close" class="close">X</a>
25     <h2>REDIMENSIONAR</h2>
26     <p>También puedes redimensionar la ventana hasta hacerla desap
27     <p>Las posibilidades que ofrece CSS3 son múltiples, tan solo h
28   </div>
29 </div>

```

La clase modalmask sera la encargada de aplicar el fondo oscuro, característico de las ventanas modales, mediante la superposición de un div que se extienda sobre el contenido. En el interior de este div incluiremos la ventana, a la que se le aplican otras 2 clases, una para darle estilo y otra para animarla. Dentro incluiremos otro enlace para crear el botón de cerrar y el contenido que mostrara la ventana.

Lo realmente importante aquí es el **CSS**. Lo muestro por partes:

```

1  .modalmask {
2      position: fixed;
3      font-family: Arial, sans-serif;
4      top: 0;
5      bottom: 0;
6      left: 0;
7      background: rgba(0,0,0,0.8);
8      z-index: 99999;
9      opacity: 0;
10     -webkit-transition: opacity 400ms ease-in;
11     -moz-transition: opacity 400ms ease-in;
12     transition: opacity 400ms ease-in;
13     pointer-events: none;
14 }
15
16 .modalmask:target {
17     opacity: 1;
18     pointer-events: auto;
19 }

```

Arriba!

(http://creativecommons.org/licenses/by/3.0/deed.es_CO)

En primer lugar el **efecto modal**, creado mediante la superposición de un div semitransparente sobre todo el contenido, usando para ello la propiedad z-index. Para ocultarlo establezco la opacidad a 0 y desactivo los eventos del puntero (pointer-events: none) para evitar que interfiera con la pagina estando oculto.

Para iniciar la animación y mostrar las diferentes ventanas modales hago uso del **selector target de CSS3**. Lo que hace este selector es modificar los estilos del div al que hace referencia un enlace. En este caso restablece la opacidad y reactiva los eventos del puntero para activar el efecto.

Una vez que tenemos la capa modal hay que dar forma a la ventana:

```

1  .modalbox{
2      width: 400px;
3      position: relative;
4      padding: 5px 20px 13px 20px;
5      background: #fff;
6      border-radius: 3px;
7      -webkit-transition: all 500ms ease-in;
8      -moz-transition: all 500ms ease-in;
9      transition: all 500ms ease-in;
10 }
11

```

Y animar su aparición, aprovechando que el **selector target** también se puede usar con los elementos hijos:

```

1  .movedown {
2      margin: 0 auto;
3  }

```

```
4  .rotate {
5      margin: 10% auto;
6      -webkit-transform: scale(-5,-5);
7      transform: scale(-5,-5);
8  }
9  .resize {
10     margin: 10% auto;
11     width:0;
12     height:0;
13 }
14 }
15 .modalmask:target .movedown{
16     margin:10% auto;
17 }
18 .modalmask:target .rotate{
19     transform: rotate(360deg) scale(1,1);
20     -webkit-transform: rotate(360deg) scale(1,1);
21 }
22
23 .modalmask:target .resize{
24     width:400px;
25     height:200px;
26 }
```

Como se puede ver sobre estas líneas, establezco en cada animación el estado inicial en la clase correspondiente, y el estado final al **hacer target**, es decir, los estilos que se aplicaran al pulsar el enlace de cada **ventana modal**.

Ya solo quedaría el **CSS** correspondiente al botón o enlace encargado de cerrar las ventanas:

```
1  .close {
2      background: #606061;
3      color: #FFFFFF;
4      line-height: 25px;
5      position: absolute;
6      right: 1px;
7      text-align: center;
8      top: 1px;
9      width: 24px;
10     text-decoration: none;
11     font-weight: bold;
12     border-radius:3px;
13 }
14
15
16 .close:hover {
17     background: #FAAC58;
18     color:#222;
19 }
```

No tiene nada especial porque para hacer desaparecer las **ventanas modales** y sus estilos, bastara con cambiar el **target** a otro enlace, en este caso `#close`, que no aporta ningun **CSS**.

DEMO

(/files/ventanasmodalescss/ventanas_modales_css.html)

| SOURCE (/files/ventanasmodalescss/ventanas_modales_css.rar)

Conclusión

Como se puede ver, las nuevas características **CSS** cada vez están mas avanzadas y, combinadas con HTML5, nos permiten hacer cosas que antes solo podíamos implementar con Javascript y sus librerías, lo cual es una buena noticia para los desarrolladores, ya que nos va a suponer una reducción considerable en el tamaño de las web y una mejora en la accesibilidad de las mismas.

Por otro lado, ahora que lo pienso, con el **selector target** tenemos otra forma de simular un evento onclick, mejor incluso que **aquella** (<http://www.nosolocss.com/blog/css/como-hacer-evento-onclick-con-css-sin-necesidad-de-javascript>) que compartí hace algún tiempo, basada en el uso de inputs tipo radio. Sin duda con este selector vamos a poder crear efectos asombrosos, yo ya tengo algunos en mente. ¿Y tu?

No olvides compartir si te gustó lo que leiste.



También podría interesarte:

- **Efecto scroll deslizante a pantalla completa usando solo css**
(/blog/css/efecto-scroll-deslizante-a-pantalla-completa-usando-solo-css)
- **Tarjeta de navidad animada con sonido usando css3 keyframes y html5**