

# Cómo Utilizar El Clipping y Masking En CSS

El CSS es cada vez más sofisticado, con el tiempo llegan nuevas herramientas para nuestro uso. En este artículo hablaremos de una nueva característica: el [Clipping](#) con el clip-path y el [Masking](#).

Son dos propiedades que empujan los límites del modelo de caja un poco más lejos de lo que hemos visto antes.

## Compatibilidad De Navegadores

Felizmente, el soporte del navegador es bastante bueno para el clip-path. Para los navegadores de escritorio ya es soportado en IE9 (aunque algunos navegadores antiguos necesitan prefijos de proveedores, como veremos mas adelante). El único problema real es IE Mobile.

Para el Masking, la situación es muy diferente. En este momento, sólo está soportado por Chrome y Safari, y luego tienes que usar el -webkit- prefix.

# Clipping con SVG

Ya tenemos una propiedad de clip en CSS2, pero la propiedad sólo nos permite usar clip en formas rectangulares. La gran noticia con el clip-path, es que ahora podemos recortar una imagen mediante polígonos a través de [SVG](#).

Supongamos que tiene el siguiente pedazo de código SVG de esta manera:

```
[html]
```

```
<svg width="0" height="0">  
<clipPath id="clipPolygon">  
<polygon points="99 143,300 293,522 143,302 -5,300 -7">  
</polygon>  
</clipPath>  
</svg>
```

```
[/html]
```

Puede simplemente crear su figura personalizada mediante el uso de este [generador de recorrido CSS](#), le dará el código SVG para que lo use en su página o incluso se puede descargar el archivo SVG si quieres usarlo.

Como se puede ver el código SVG contiene una identificación simple, esto es lo que vamos a utilizar en el CSS para hacer referencia a ella. Para recortar una imagen simple que usando SVG usaríamos la propiedad clip-path y luego llamar a la función url():

```
[css]
```

```
img {
clip-path: url(#clipPolygon);
}
```

```
[/css]
```

En este caso, nuestro archivo SVG tiene un solo polígono por lo que también podemos utilizar la función `polygon()` y basta con insertar los puntos que tenemos en nuestro archivo SVG:

```
[css]
```

```
img {
clip-path: polygon(99px 143px,300px 293px,522px 143px,302px
-5px,300px -7px);
}
```

```
[/css]
```

Ambos de estos ejemplos funcionan exactamente de la misma manera, la única diferencia es que el segundo enfoque no requiere ningún formato HTML. Usando polígonos también puede utilizar las transiciones:

```
[css]
```

```
.clip{
-webkit-clip-path: polygon(195px 147px,295px 297px,395px
147px,295px -3px);
transition: 1s all ease;
}
```

```
.clip:hover{
-webkit-clip-path: polygon(82px 148px,295px 297px,496px
147px,295px -3px);
}
```

```
[/css]
```

El siguiente sería un ejemplo ;)

See the Pen [Clipping con CSS](#) by HostDime Colombia (@HostDimeColombia) on [CodePen](#).

# Masking En CSS

Esta propiedad es más experimental y, por tanto, un poco más difícil de probar, pero probablemente estará disponible para todos los navegadores, tan pronto como la gente comience a usarla en [Chrome](#) o [Safari](#).

El Masking funciona exactamente igual que lo hace en Photoshop: una imagen o un degradado se utiliza como filtro para algunas partes de la imagen.

Para utilizarlo, tenemos que usar la propiedad de Masking en la imagen y la referencia a un archivo SVG, de la siguiente manera:

```
[css]

img {
  -webkit-mask-image: url("mask.svg");
}
```

```
[/css]
```

Esto recuperará el archivo SVG y la usará como máscara para la imagen. Si lo prefiere, puede utilizar el ID de un elemento SVG:

```
[css]

img {
  -webkit-mask: url(#masking);
}
```

```
[/css]
```