

# Sombras CSS

Un efecto interesante que podemos utilizar en nuestros diseños web a partir de CSS3 es el de las **sombras**, ya que nos permiten añadir atractivas sombras que pueden suavizar y hacer más elegantes nuestros diseños.

Podemos utilizar dos tipos de propiedades, `text-shadow` y `box-shadow`. La primera de ellas para sombras en textos, y la segunda para sombras en otro tipo de elementos como cajas o contenedores. Ambas propiedades son prácticamente idénticas, solo con algunas diferencias:

Propiedad	Valor	Significado
<code>text-shadow</code>	<code>none</code>	Aplica una sombra a un contenido de texto.
<code>box-shadow</code>	<code>inset</code>   <code>none</code>	Aplica una sombra a una caja o contenedor.

- Los valores `x` y `y` nos permiten indicar las coordenadas de **X** e **Y** donde se situará la sombra, respecto a la posición del texto o contenedor original.
- El primer valor es el **blur**, que no es otra cosa que el radio de desenfoque de la sombra (*en px, em, etc...*). Cuanto más bajo sea este valor, menos difuminada estará la sombra, cuanto más alto sea, más borrosa.
- El segundo valor es el **factor de crecimiento** de la sombra. En algunos casos, en los que tenemos una sombra con muy poco desenfoque y las posiciones son **0**, puede que la sombra quede justo detrás del elemento, sin poderse apreciar. Aumentando este valor, haremos que la sombra crezca hacia los lados el tamaño que indiquemos.
- Por último, podemos indicar el `a` a utilizar (*recuerda que puedes utilizar canales alfa*) y la palabra clave `inset` si queremos que la sombra sea interna, en lugar de externa que es la que aparece por defecto.

```
div {  
  /* Sombra normal */  
  box-shadow: 2px 2px 10px #666;  
  /* Sombra superior sin desenfoque */  
  box-shadow: 5px -5px 0 2px #444;  
  /* Sombra interior */  
  box-shadow: 5px 5px 25px #222 inset;  
}
```

**Ojo:** Ten en cuenta que tanto el `a` de factor de crecimiento como la palabra clave `inset` sólo están disponibles para `box-shadow`, ya que `text-shadow` no posee esos parámetros.

## Sombras múltiples

Una característica interesante de la propiedad `background` en CSS es la posibilidad de indicar fondos múltiples separando por comas. En el caso de las sombras, también es posible hacer lo mismo, consiguiendo efectos muy vistosos:

```
p {  
    /* Se indican 3 sombras diferentes */  
    text-shadow:  
        2px 2px 0 #AAA,  
        6px 6px 2px #777,  
        12px 12px 8px #444;  
}
```

En el caso de la propiedad `box-shadow` también se pueden indicar sombras múltiples.

### Sombras flexibles

En el módulo de [filtros CSS](#) existe una función denominada `drop-shadow` que puede utilizarse en un filtro. Dicha función permite aplicar una sombra **idéntica** a un objeto determinado con una forma irregular.

Por ejemplo, podríamos disponer de una imagen PNG con fondo transparente, al que le queremos aplicar una sombra de modo que respete su contorno. Con la propiedad `box-shadow` la sombra que se aplicaría sería totalmente rectangular, sin embargo, con la propiedad `filter` y la función `drop-shadow` (*la cuál tiene los mismos parámetros que `text-shadow`*), aplicaría una sombra que respetará exactamente la forma de la imagen.

```
img {  
    filter: drop-shadow(5px 5px 10px #444);  
}
```

# Filtros CSS

[Lenguaje CSS](#)

[HTML5](#)

[CSS3](#)

[JavaScript](#)

[Terminal](#)

[CheatSheets](#)

[YouTube](#)

[Twitter](#)

[Sombras CSS](#) [Capítulo anterior](#) [Modos de fusión CSS](#) [Capítulo siguiente](#)

Los **filtros CSS** son una característica muy atractiva de CSS que permite aplicar ciertos efectos de imagen, propios de aplicaciones de retoque fotográfico, como **sepia**, variaciones de **brillo** o **contraste** (*u otros*) al vuelo en el propio navegador, sin hacer cambios permanentes sobre una imagen.

Dichos filtros funcionan a través de la propiedad `filter`, a la cuál hay que especificarle una función concreta de las existentes, como por ejemplo la función de **blanco y negro** (*grayscale*):

```
img {  
    filter: grayscale(75%);  
}
```

**Nota:** Aunque es la que se ha utilizado en el ejemplo, esta propiedad funciona con otros elementos, es decir, no debe aplicarse necesariamente sobre una imagen.

## Funciones de filtros

Los **filtros de CSS** tienen a nuestra disposición un amplio abanico de funciones, listas para utilizar mediante la propiedad `filter` y aplicarlas a los elementos que queramos:

Función	Significado	Valor	Mínimo	Máximo	>100%
<code>grayscale</code>	Escala de blanco y negro	0% <sup>1</sup>		100% = B&N	
<code>blur</code>	Grado de difuminado		Tamaño de radio (desenfoque)		
<code>sepia</code>	Grado de color sepia	0% <sup>1</sup>		100% = sepia	
<code>saturate</code>	Grado de saturación	0% = B&N		100% <sup>1</sup>	Sí
<code>opacity</code>	Grado de transparencia	0% = invisible		100% = visible <sup>1</sup>	
<code>brightness</code>	Brillo	0% = negro		100% <sup>1</sup>	Sí
<code>contrast</code>	Contraste	0% = gris		100% <sup>1</sup>	Sí
<code>hue-rotate</code>	Rotación de color (matiz)	0/360deg <sup>1</sup>			
<code>invert</code>	Invertir	0% <sup>1</sup>		100% = invertido	
<code>drop-shadow</code>	Sombra idéntica		(Ver apartado de <a href="#">sombras CSS</a> )		

(<sup>1</sup>) Es el valor considerado como original

**Nota:** En las cantidades de se pueden utilizar indistintamente porcentajes (*entre 0% y 100%*) o valores numéricos con decimales (*entre 0 y 1*). Por ejemplo, `filter: contrast(0.5)` es lo mismo que `filter: contrast(50%)`.



grayscale (0%)



grayscale (50%)



grayscale (100%)



blur (0)





blur

(5px)



blur (10px)





(0%)

sepia



(50%)

sepia



(100%)

sepia



(0%)

saturate



(50%)

saturate



(100%)

saturate





(200%)

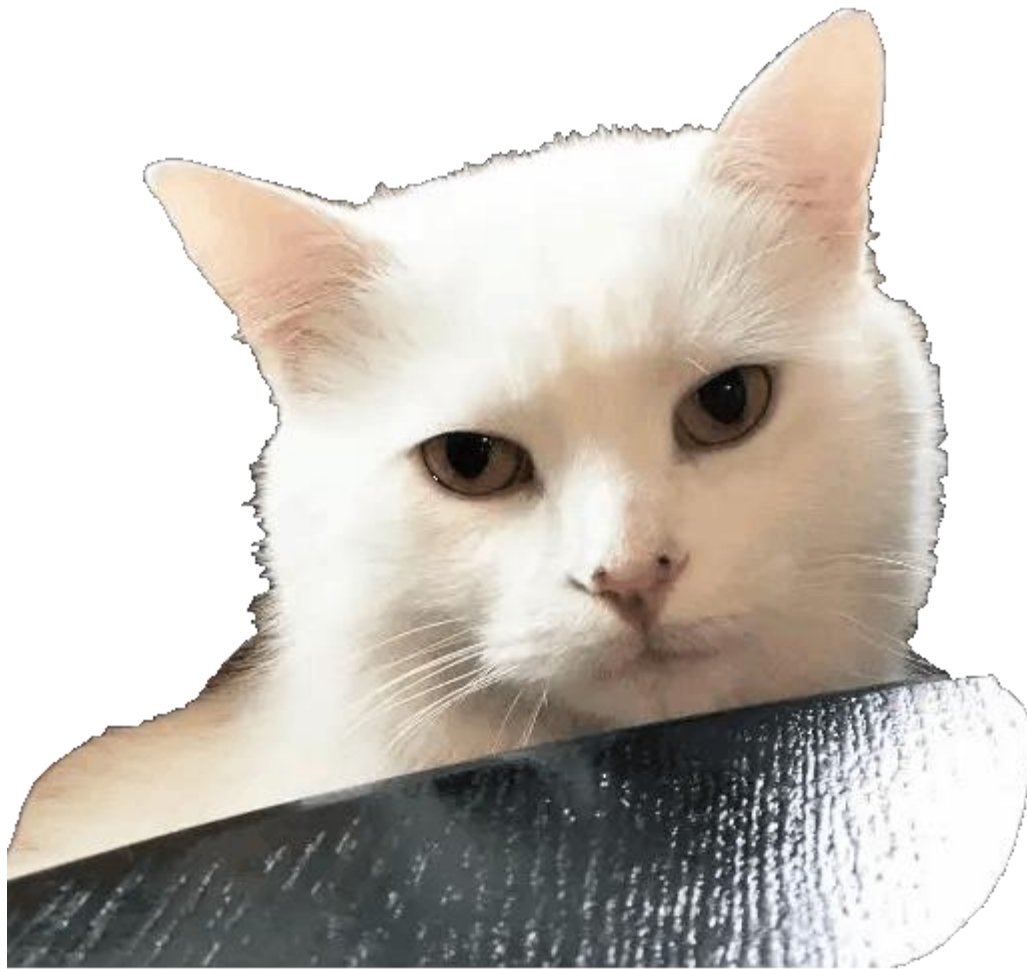
saturate





(0%)

opacity



(50%)

opacity



(100%)

opacity



brightness (0%)



brightness (50%)



brightness (100%)





brightness (200%)





(0%)

contrast



(50%)

contrast



(100%)

contrast



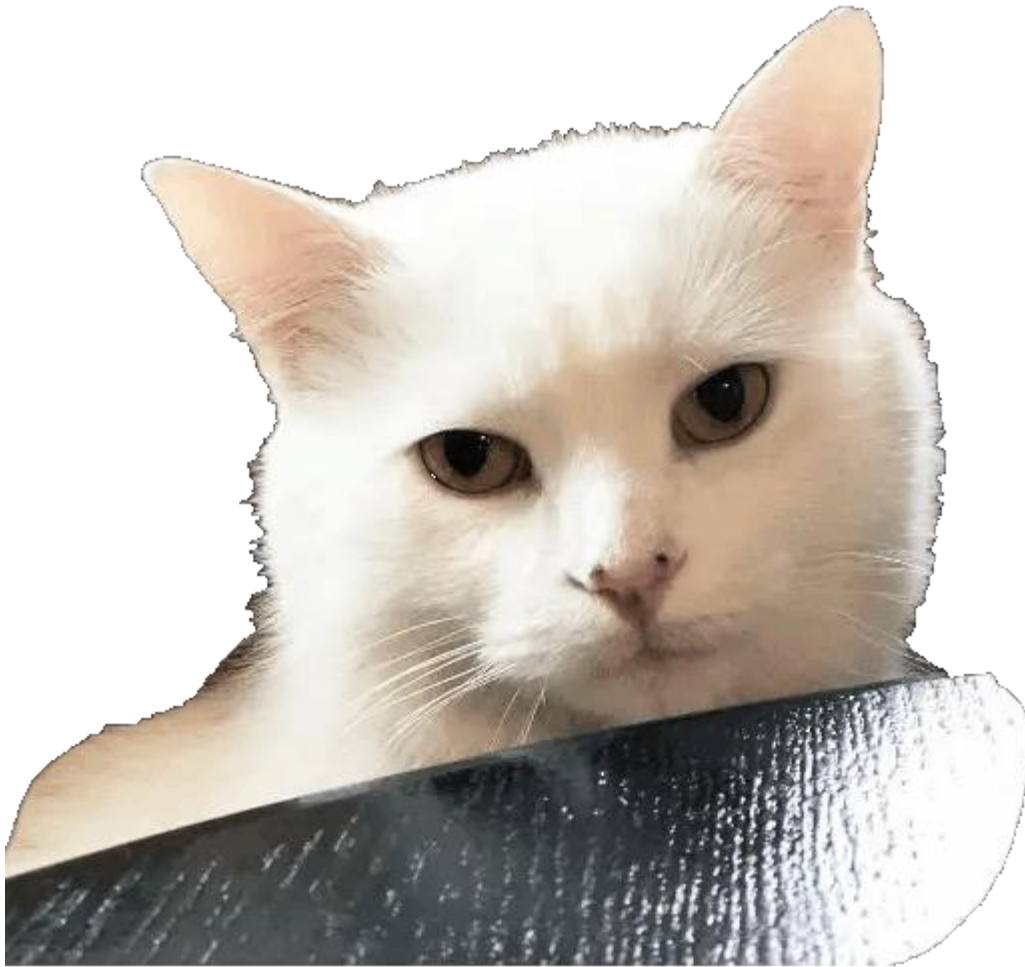
(200%)

contrast



rotate (0)

hue-



rotate (90deg)

hue-





rotate (180deg)

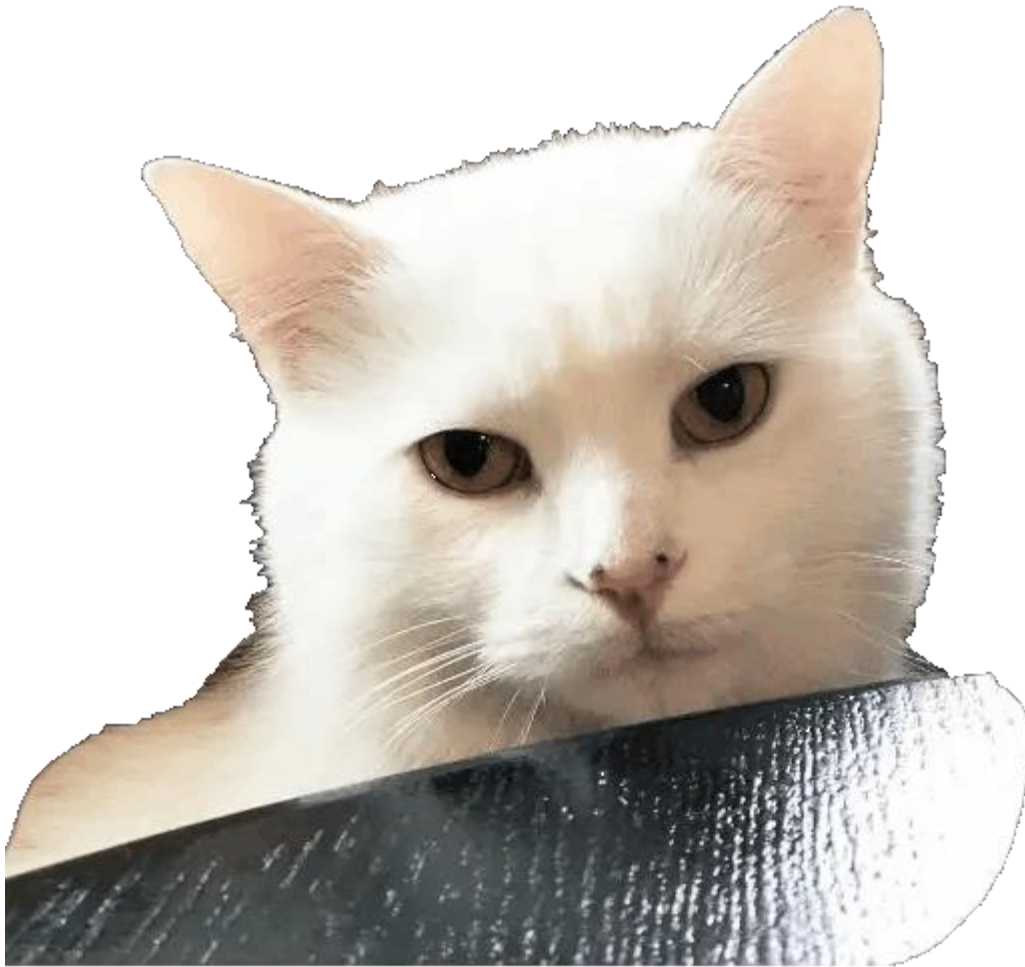
hue-





rotate (240deg)

hue-

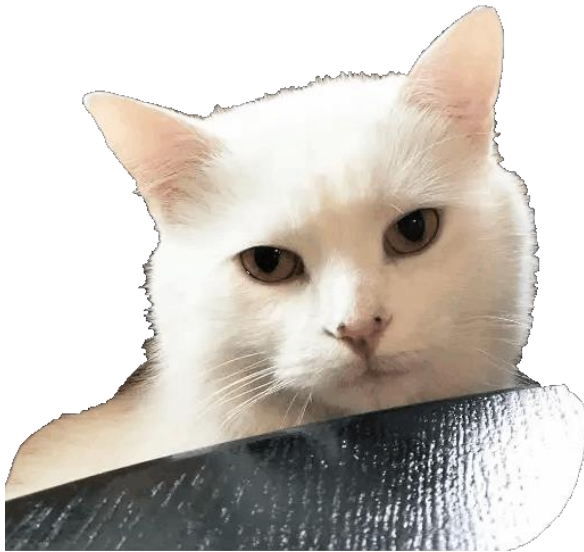


invert

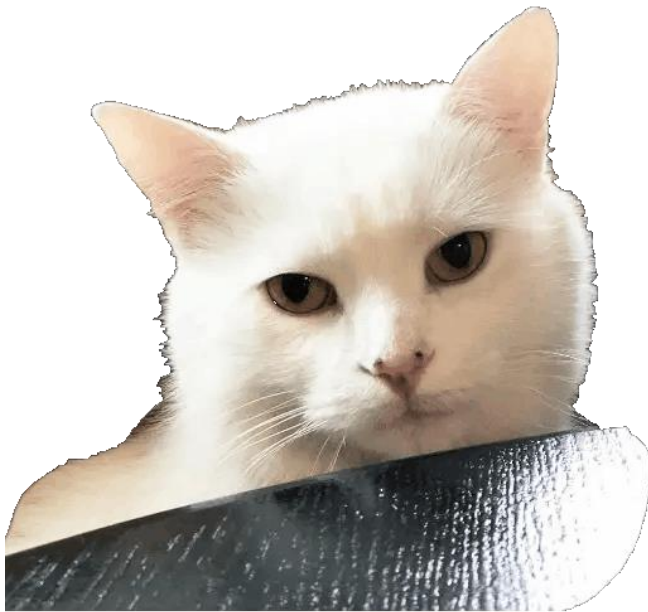
(0%)



invert (50%)



invert (100%)

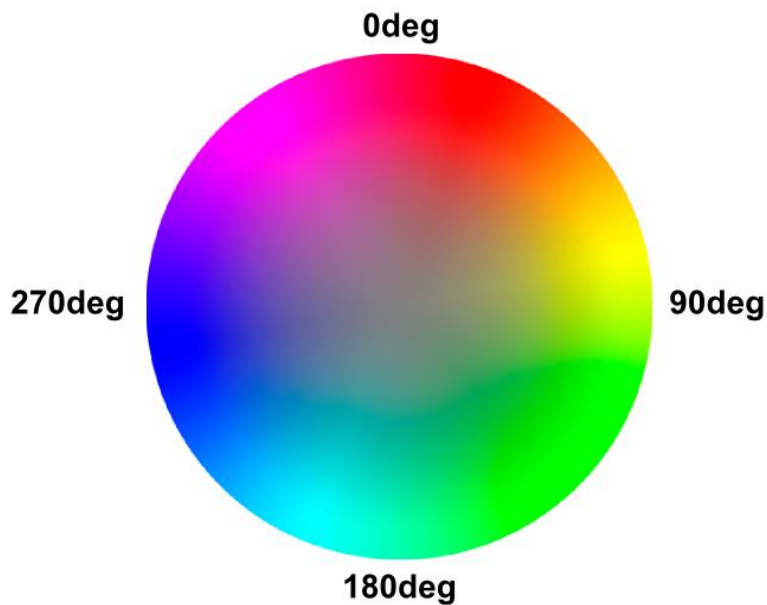


drop-shadow (2px 2px 5px gold)



drop-shadow (5px 5px 10px red)

Estas propiedades permiten alterar los colores, tonalidades o diferentes aspectos visuales, tal como se harían desde un programa de diseño gráfico. Por ejemplo, la rotación de colores (*hue-rotate*) se representa mediante una rueda de colores, donde el grado definido especifica el matiz que se le da a los colores, pudiendo alterarlo ligera o intensamente:



**Nota:** La función `opacity` ya existe en CSS fuera de los filtros CSS. Puede utilizarse de forma independientemente, con el siguiente formato: `opacity: 0.75`.

# Modos de fusión CSS

[Lenguaje CSS](#)

[HTML5](#)

[CSS3](#)

[JavaScript](#)

[Terminal](#)

[CheatSheets](#)

[YouTube](#)

[Twitter](#)

[Filtros CSS](#) [Capítulo anterior](#) [¿Qué es Responsive Design?](#) [Capítulo siguiente](#)

Los filtros CSS están muy bien en multitud de situaciones, pero hay casos en los que se nos pueden quedar cortos. Los **modos de fusión** son un interesante añadido que aparece en CSS para dotar de otra característica muy común en los programas de diseño y edición de imagen: la **fusión de dos elementos** sumando la información de color de los píxeles de cada uno de ellos.

Mediante las propiedades `mix-blend-mode` y `background-blend-mode` podemos utilizar modos de fusión en los diferentes elementos que seleccionemos en una página web, aplicando **composición** para mezclar los colores, superponiendo dos elementos, consiguiendo una **variación** como resultado final.

Dichas propiedades son las siguientes:

Propiedad	Valor	Significado
<code>mix-blend-mode</code>	<b>normal</b>	Aplica un modo de fusión específico a un elemento.
<code>background-blend-mode</code>	<b>normal</b>	Aplica un modo de fusión específico a un fondo.
<code>isolation</code>	<b>auto</b>   <code>isolate</code>	Establece si un elemento debe aislarse del resto.

La diferencia entre ambas propiedades es que la primera de ellas, `mix-blend-mode`, se utiliza para elementos, mientras que la segunda, `background-blend-mode`, se utiliza para aplicar modos de fusión en fondos. Por ejemplo, usándola con una imagen:

```
body {  
    background: green;  
}  
  
img {  
    mix-blend-mode: multiply;  
}
```

Entre los diferentes **modos de fusión** que podemos utilizar en dichas propiedades, se encuentran las siguientes:

- multiply (*multiplicar*)
- screen (*trama*)
- overlay (*superponer*)
- darken (*oscurecer*)
- lighten (*aclarar*)
- color-dodge (*sobreexponer color*)
- color-burn (*subexponer color*)
- hard-light (*luz fuerte*)
- soft-light (*luz suave*)
- difference (*diferencia*)
- exclusion (*exclusión*)
- hue (*tono*)
- saturation (*saturación*)
- color
- luminosity (*luminosidad*)

Si no tenemos ningún modo de fusión aplicado, el valor por defecto de la propiedad es **normal**. Como ejemplo, a continuación tenemos una imagen de **Smudge** aplicado con un modo de fusión sobre un fondo rojo. Puedes mover el ratón por encima de él para comparar la fotografía sin modo de fusión:



normal



multiply



screen





overlay



darken



lighten



color-dodge



color-burn



hard-light





soft-light



difference



exclusion



hue



saturation



color



luminosity

Podemos hacer uso de la propiedad `isolation` con valor **auto** o **isolate**, para que, en el último caso, se proteja un elemento de la mezcla de fusión y aislarlo sin que se aplique.

**Nota:** Lo mismo que hemos hecho en el ejemplo anterior con una imagen y un color de fondo, se podría hacer con 2 o varias imágenes.

El soporte de estas propiedades en navegadores modernos es relativamente bueno, como se ve a continuación en la siguiente tabla: