

# CSS3 Avanzado

## CAPÍTULO I Efectos visuales.

1

### Capítulo 1. Efectos visuales.

CSS avanzado permite crear efectos visuales modernos y dinámicos sin necesidad de JavaScript o imágenes externas.

En este tema veremos cómo aplicar sombras, filtros, recortes, máscaras y degradados para transformar la apariencia de una web.

Estas técnicas mejoran la profundidad, el contraste, la originalidad y la estética general del diseño.

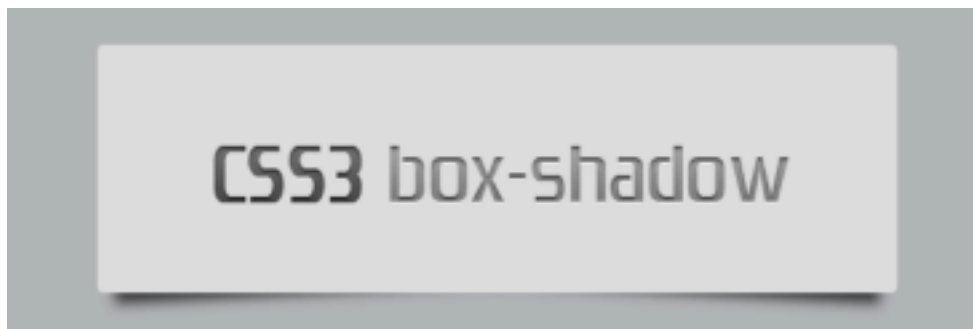
Además, permiten efectos visuales complejos directamente desde el navegador, optimizando el rendimiento.

Son recursos clave para diseñar interfaces atractivas y profesionales.

Dominar estos efectos lleva el desarrollo frontend a un nivel más creativo y

## actual. **1.1 Efectos con sombras.**

### Sombras



```
.elemento {  
    box-shadow: 2px 2px 5px 999;  
}
```

La sintaxis completa de la propiedad box-shadow es muy compleja y se define en el [borrador de trabajo del módulo de fondos y bordes de CSS3](http://dev.w3.org/csswg/css3-background/#the-box-shadow)

<http://dev.w3.org/csswg/css3-background/#the-box-shadow>).

A continuación, se muestra su sintaxis simplificada habitual:

**box-shadow:** <medida> <medida> <medida>? <medida>? <color>

2

- La **primera** medida es obligatoria e indica el desplazamiento horizontal de la sombra. Si el valor es positivo, la sombra se desplaza hacia la derecha y si es negativo, se desplaza hacia la izquierda.
- La **segunda** medida también es obligatoria e indica el desplazamiento vertical de la sombra. Si el valor es positivo, la sombra se desplaza hacia abajo y si es negativo, se desplaza hacia arriba.
- La **tercera** medida es opcional e indica el radio utilizado para difuminar la sombra. Cuanto más grande sea su valor, más borrosa aparece la sombra. Si se utiliza el valor 0, la sombra se muestra como un color sólido.
- La **cuarta** medida también es opcional e indica el radio con el que se expande la sombra. Si se establece un valor positivo, la sombra se expande en todas direcciones. Si se utiliza un valor negativo, la sombra se comprime.
- El **color** indicado es directamente el color de la sombra que se muestra.

La siguiente regla CSS muestra una sombra en los navegadores Firefox y Safari:

```
.elemento {  
    box-shadow: 2px 2px 5px #999;  
}
```

Si incluimos como último parámetro **inset**, las propiedades de las sombras se aplicarán dentro del elemento del “box-model”, eso quiere decir que los parámetros partirán del border hasta el centro:

```
box-shadow: 20px 20px 4px 5px #555 inset;
```

3

## Neumorfiesmo

Neumorfismo es un estilo de diseño que combina elementos minimalistas. Se caracteriza por el uso de sombras suaves y colores pastel para crear un aspecto tridimensional y atractivo.



```
border-radius: 50px;
```

```
background: linear-gradient(145deg, #cacaca, #f0f0f0);
box-shadow: 20px 20px 60px #bebebe,
-20px -20px 60px #ffffff;
```

Generador: <https://neumorphism.io/>

## Texto con sombra

La propiedad llamada text-shadow se utiliza para mostrar textos con sombra.

```
h1 {
    color: #000;
    text-shadow: 2px 2px 3px #555; }
```

La sintaxis de la propiedad text-shadow obliga a indicar dos medidas y permite establecer de forma opcional una tercera medida y un color. Las dos medidas obligatorias son respectivamente el

4 desplazamiento horizontal y vertical de la sombra respecto del texto. La tercera medida opcional indica lo nítido o borroso que se ve la sombra y el color establece directamente el color de la sombra.

## Efecto brillo

Si queremos crear un resplandor en vez de una sombra, tenemos que darle valor 0 a text-shadow en ambas distancias y aplicar un color claro.

```
text-shadow: 0px 0px 20px rgb(0,255,0);
```

Si queremos aumentar la intensidad del brillo, le añadimos más valores de sombra.

```
text-shadow: 0px 0px 20px rgb(0,255,0), 0px 0px 20px rgb(0,255,0), 0px 0px 20px rgb(0,255,0);
```



## Textos en 3D

Veamos una aplicación práctica de text-shadow que dará como resultado un efecto distinto a una simple sombra paralela.

Sabemos que podemos aplicar diferentes sombras a un texto, y que podemos decidir que el difuminado de la sombra sea 0. Partiendo de esta base, podemos crear un efecto de pseudo 3D a cualquier texto.

Primero tenemos que aplicar el efecto sombra a nuestro texto. Creamos un texto blanco y una sombra a una distancia x e y de un color gris claro y con difuminado 0:

```
text-shadow: 1px 1px 0px rgb(230,230,230);
```

5



Ahora mismo tiene poco de 3D. Probemos a ir añadiéndole unas cuantas sombras más, aumentando en cada una 1 pixel las distancias x e y haciendo el gris más oscuro:

```
text-shadow:  
1px 1px 0px rgb(230,230,230),  
2px 2px 0px rgb(200,200,200),  
3px 3px 0px rgb(180,180,180),  
4px 4px 0px rgb(160,160,160);
```



Eso ya tiene algo más de aspecto 3D, pero aún le podemos dar un aspecto más realista. Aplicaremos una nueva sombra siguiendo el método anterior, con la diferencia que esta vez le daremos color negro. Por último añadimos una sombra mayor que todas las demás, de color gris claro y con difuminado:

```
text-shadow:
1px 1px 0px rgb(230,230,230),
2px 2px 0px rgb(200,200,200),
3px 3px 0px rgb(180,180,180),
4px 4px 0px rgb(160,160,160),
/*Añadimos*/ 5px 5px 0px rgb(0,0,0), 8px 8px 20px rgb(0,0,0); 6
```



## Texto con relleno gradiente o degradado e imágenes

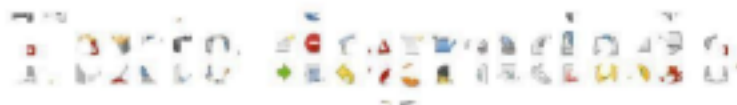
Combinando el texto con imágenes y degradados, se pueden lograr fácilmente efectos gráficos propios de los programas de diseño. A continuación, se detalla cómo crear un texto que muestra su color en forma de degradado o gradiente.

Para ello utilizaremos las siguientes propiedades:

- **background-clip: text;** Con esta propiedad adoptaremos el background a la forma del texto.
- **text-fill-color: transparent;** Con esta propiedad ponemos el color del texto transparente para poder visualizar su background-image.

Utilizando estas propiedades podemos declarar lo siguiente:

```
h1 {  
    background-image:url("imagenes/sprite.png"); /* Imagen fondo */  
    background-clip: text; /* adaptamos el degradado al texto */ color:  
transparent; /* Ocultamos el color del texto */ font-size:6em;  
}
```



Si utilizamos la propiedad **gradient** (que estudiaremos más adelante) podremos realizar un degradado en el fondo del texto:

```
h1 {  
    background: radial-gradient(circle,rgb(63, 94, 251) 0%, rgb(252,  
70, 107) 100%);
```

7

```
background-clip: text; /* adaptamos el degradado al texto */  
color: transparent; /* Ocultamos el color del texto */  
font-size:4em; }
```



## Sombreado de imágenes PNG y SVG

El filtro CSS drop-shadow() aplica un efecto de sombra a un elemento similar a la propiedad box-shadow, pero con algunas diferencias clave. Se usa principalmente para aplicar sombras a imágenes y gráficos (como íconos SVG), respetando la forma no rectangular de dichos elementos.

**filter: drop-shadow()**

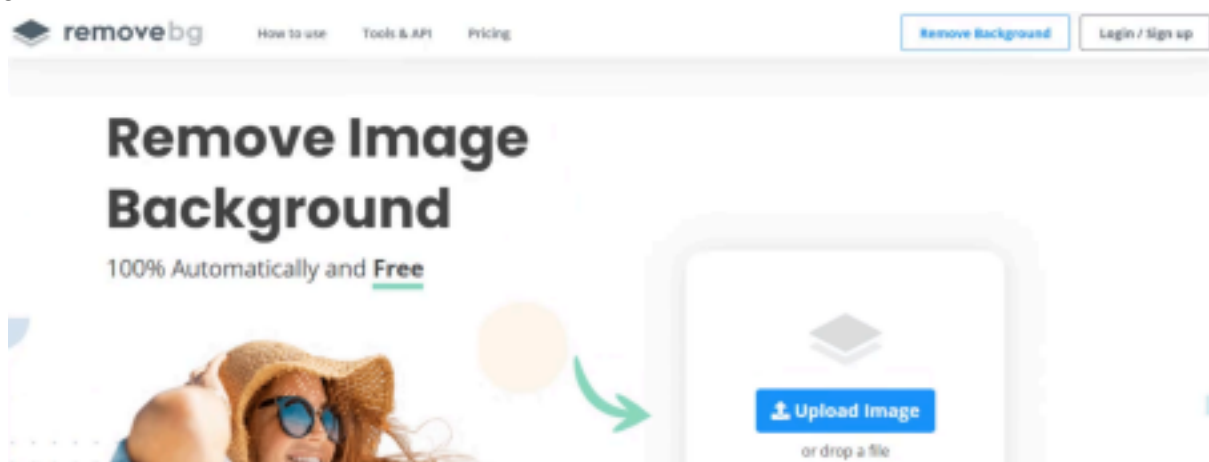


Ejemplo:

```
img {  
  filter: drop-shadow(10px 10px 5px rgba(0, 0, 0, 0.5));  
}
```

Si la imagen no tiene canal alfa, podemos borrar el background con la siguiente aplicación web, <https://www.remove.bg/es> :

8



## Otros filtros.

Los filtros CSS son una característica muy atractiva de CSS que permite aplicar ciertos efectos de imagen y elementos gráficos como fondos o videos 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.



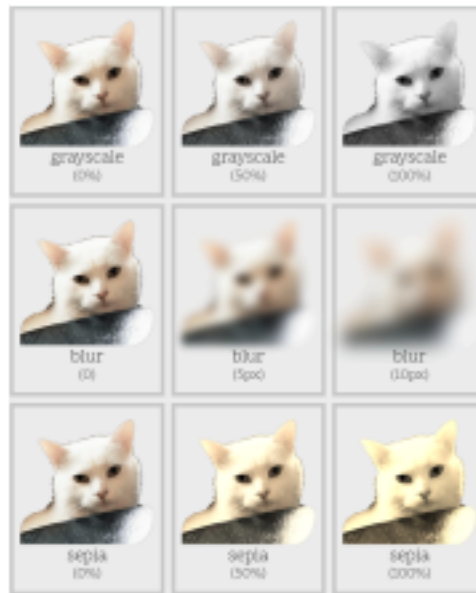
Dichos filtros funcionan a través de la propiedad filter, a la cuál hay que especificarle una función concreta de las existentes, como por ejemplo la función de blanco y negro (grayscale):

```
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.

grayscale	Escala de blanco y negro	0% <sup>1</sup>	100% = B&N	
blur	Grado de difuminado	Tamaño de radio (desenfoque) px		
sepia	Grado de color sepia	0% <sup>1</sup>	100% = sepia	
saturate	Grado de saturación	0% = B&N	100% <sup>1</sup>	Sí
opacity	Grado de transparencia	0% = invisible	100% = visible <sup>1</sup>	
brightness	Brillo	0% = negro	100% <sup>1</sup>	Sí
contrast	Contraste	0% = gris	100% <sup>1</sup>	Sí
hue-rotate	Rotación de color (matiz)	0/360deg <sup>1</sup>		

invert	Invertir	0% <sup>1</sup>	100% = invertido	
drop-shadow	Sombra idéntica	(2px 2px 5px gold)		



Ejemplo:

```
.galeria img:hover {
  filter: grayscale(0) brightness(1.1) saturate(1.3);
  width: 260px;
}
```



En la web <https://unsplash.com/es> se puede ver como aplican el efecto de filtro a las imágenes.

10

## Propiedad backdrop-filter

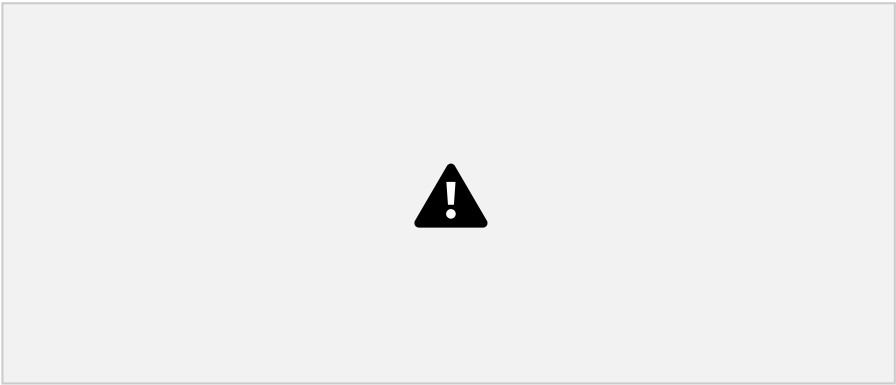
Permite aplicar efectos gráficos (como desenfoque, brillo, contraste o saturación) al fondo que hay detrás de un elemento, sin afectar directamente al contenido del propio elemento.

En otras palabras, funciona como un “filtro” aplicado al fondo visible a través del elemento, muy útil

para crear efectos de vidrio esmerilado o desenfoque translúcido

```
backdrop-filter: <función-de-filtro> [<función-de-filtro>]*;
```

Ejemplo de funciones de filtro: blur(), brightness(), contrast() , grayscale() ...



```
body { background: url('img.jpg') no-repeat center center/cover; }
.caja {
  background: rgba(255, 255, 255, 0.1);
  border-radius: 10px;
  /* filter: blur(2px); */
  backdrop-filter: blur(1px);
  color: #fbf8f8;
  text-align: center;
}
```

Propiedades de fusión

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.

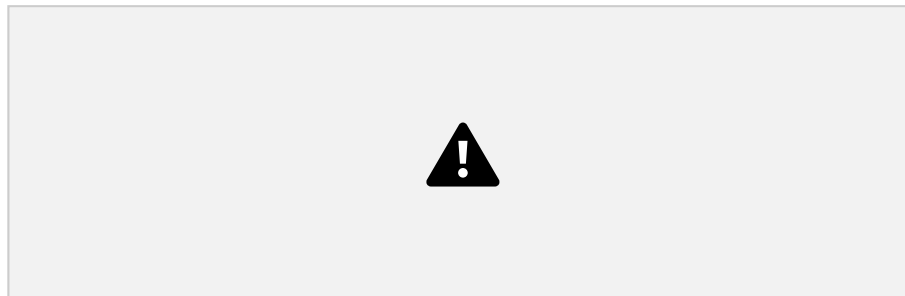
mix-blend-mode		Modos de fusión	Aplica un modo de fusión a un elemento con todo lo que hay debajo.	

11

background-ble nd mode		Modos de fusión		
isolation				

**Modos de fusión:** multiply, screen, overlay. darken , lighten color-dodge, color-burn, hard-light , soft-light, difference, exclusion, hue , saturation

```
body {  
margin: 0;  
height: 100vh;  
background: linear-gradient(45deg, #ff4b1f, #1dddf, #28a745);  
background-size: 300% 300%;  
.titulo {  
text-align: center;  
font-size: 120px;  
font-family: Arial, Helvetica, sans-serif;  
color: white;  
mix-blend-mode: color-burn;  
letter-spacing: 5px;  
}
```



Ejemplos reales: [Link](#)



## 1.2 DEGRADADOS

Los degradados implementan un gradiente de color, que pasa de un estado a otro a lo largo del fondo de los elementos HTML, ya sea capas, elementos de listas, botones, etc. Dichos degradados se obtendrán por medio de la especificación de una serie de características, como la posición inicial, la dirección hacia donde se realizará, si es circular o lineal, y los colores que se incorporarán en cada uno de los pasos del gradiente.



El navegador es el encargado de renderizar el degradado en función de las características descritas para definirlo. Podemos asignar degradados como fondo en cualquier elemento HTML donde se podía implementar una imagen de fondo.

Son muy comunes en el diseño **web moderno**, ya que permiten crear fondos dinámicos, suaves y ligeros sin necesidad de imágenes.

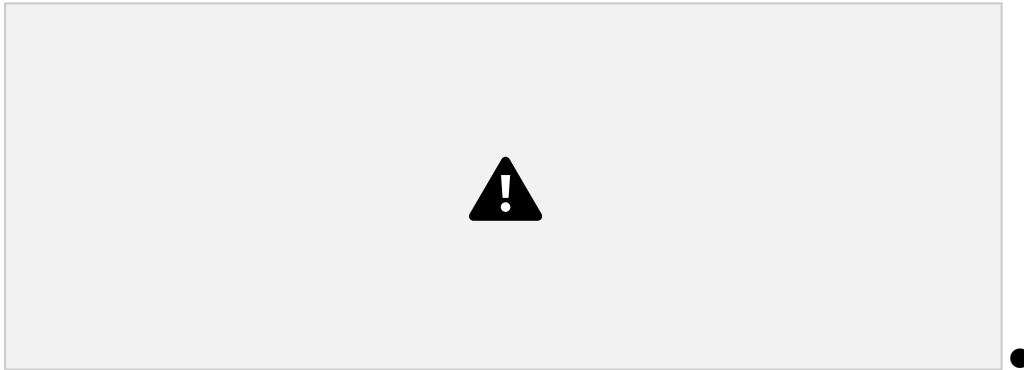


## ● Degradados lineales:

En los que se crea un gradiente que va de un color a otro de manera lineal.

Los degradados lineales se consiguen con el atributo background asignándole el gradiente con la propiedad "linear-gradient".

```
div{ height: 130px;  
width: 630px;  
background: linear-gradient(orange, pink);}
```

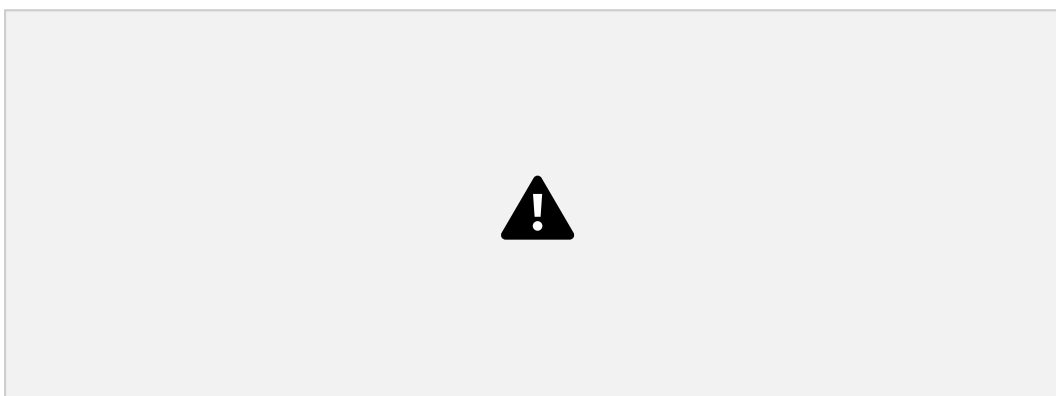


## Degradados circulares:

En ellos se implementa un gradiente que se distribuye radialmente, desde un punto del elemento hacia fuera, de manera circular, que puede tener el mismo valor de radio (para hacer degradados en círculos perfectos) o con valores de radio variables (lo que generaría elipses).

El valor que asignamos a background en este caso será por medio del atributo "radial-gradient", además de toda la serie de parámetros necesarios para definir el degradado según nuestras intenciones.

```
div.circular{ background: -webkit-radial-gradient(#0f0, #06f);
```



14

### 1.2.1 Explicación detallada de los degradados lineales.

El degradado de colores más sencillo que podemos crear es el degradado lineal. Con CSS3 se pueden especificar con tan solo definir una serie de parámetros en la propiedad linear-gradient, que nos permiten configurar todo tipo de gradientes de dos o más colores, sin la necesidad de usar imágenes.

```
background: linear-gradient(parámetros);
```

Como vemos, para asignar un degradado a un elemento, tenemos que utilizar la propiedad `linear-gradient` sobre un atributo `background`, o `background-image`. Todos los elementos que soportan imágenes de fondo permite también colocar degradados de fondo. Además, tendremos que indicar una serie de parámetros variables para la creación del degradado, que son los que realmente tienen alguna dificultad de entender. Estos parámetros son los siguientes:

**A) Origen-y/o-ángulo del degradado:**

El primer parámetro sería el origen desde donde comenzará el degradado y/o el ángulo de disposición del gradiente de color. Podemos decir que el degradado comience desde arriba, abajo o desde una esquina cualquiera. Por defecto los degradados serán distribuidos en un gradiente en línea recta, pero además podemos indicar un ángulo distinto con el que se vaya produciendo el gradiente de color.

- B) lista-de-colores y opcionalmente, el lugar hasta donde se debe mostrar cada uno:** Luego colocaremos los colores, todos los que queramos, que deben utilizarse en el degradado, separados por comas. Además, si lo deseamos, podemos definir las paradas de color "color stops", que consiste en declarar el lugar desde donde debe empezar el gradiente del color.

```
background: linear-gradient(orange, pink);
```

Esto hace un degradado desde el color naranja hacia el rosa. Todos los demás parámetros quedarían con sus valores predeterminados y el resultado sería que el degradado se realiza en toda la altura del elemento, de arriba a abajo, en un gradiente vertical, comenzando el naranja en la parte de arriba y acabando en rosa en la parte de abajo.

```
background: linear-gradient(top left, #fff, #f66);
```

Este degradado comienza en la esquina superior izquierda y se crea un gradiente que va hacia la esquina opuesta. Por tanto, el degradado formará un gradiente oblicuo, en diagonal desde la esquina superior izquierda, donde estaría el blanco (`#fff`), hasta la esquina inferior derecha, donde estaría el rosa (`#f66`).

```
background: linear-gradient(180deg, #fff, #f66);
```

Este degradado define su dirección por medio de un ángulo expresado en grados. 0 grados haría que el degradado comenzará en la parte de la izquierda y 180deg indica que el degradado empezaría justo por el lado contrario, es decir, por la derecha. De modo que en la parte de la derecha tendríamos el color morado y en la izquierda tendríamos el rosado.

15

```
background: linear-gradient(#00f 50%, #000);
```

Este degradado tiene lo que se llama un "color stop" es decir, una parada de color, que está asignada con el 50% en el primer color. Quiere decir que el primer color estaría homogéneo (sin degradado) hasta el 50% del tamaño del elemento y que luego comenzaría a degradarse hacia el

segundo color.

```
background: linear-gradient(45deg, #66f, #f80, #ffc);
```

Este degradado tiene una disposición en diagonal, por los 45 grados que se han definido. Además, podemos ver que hemos definido más de dos colores en el degradado. Podemos poner tantos como queramos, separados por comas. Como no hay "color stops" los tres colores se distribuyen de manera equitativa, desde la esquina inferior izquierda hasta la superior derecha.

```
background: linear-gradient(45deg, #66f 10%, #f80 30%, #ffc 60%);
```

Este degradado se hace también en diagonal, desde la esquina inferior izquierda, igual que el anterior, pero hemos definido una serie de paradas de color (color stops), con lo cual la distribución del gradiente no es homogénea. El primer color empezaría a degradarse hacia el segundo cuando se llega al 10% del tamaño del elemento. El degradado hacia el segundo color se completaría al llegar al 30% y a partir de ese punto empezaría a degradarse hacia el tercer color. El degradado entre el segundo y tercer color se realizaría desde el 30% al 60% del tamaño del elemento y se completaría cuando estamos en el 60%. A partir de ese último color stop (60%) tendríamos el último color de manera homogénea hasta el 100% del tamaño. Por tanto, el color predominante veremos que es el tercero, que tiene un 40% (100% del elemento - 60% del espacio donde veremos degradados) del espacio para mostrarse con su RGB tal cual fue definido.

```
background: linear-gradient(45deg, #66f 40px, #f80 180px, #ffc);
```

Este es el mismo degradado que el anterior, pero con paradas de color distintas. Además, estamos definiendo esos "color stops" con medidas en píxeles en lugar de porcentajes.

```
background: linear-gradient(left, #f00, #f80, #ff0, #0f0, #00f, #60f, #c0f);
```

Este degradado, que empieza en la izquierda y con un gradiente recto hacia la derecha, tiene varios colores, los del arcoiris.





### 1.2.2. Atributo repeating-linear-gradient, degradados lineales con múltiples repeticiones del mismo gradiente de color.

Los degradados con repetición se realizan de manera similar a los degradados lineales normales, la diferencia es que utilizaremos el nombre de atributo repeating-linear-gradient y que, para que se produzcan las repeticiones tendremos que utilizar paradas de color.

La sintaxis es exactamente la misma que ya conocemos de los degradados lineales:

```
repeating-linear-gradient(origen, colores)
```

Siendo que en origen podremos colocar, tanto la posición inicial donde comienza el degradado, como el ángulo que debe formar el gradiente. Luego, los colores, como también vimos, se indican separados por comas. Sin embargo, ahora, para que realice la repetición, estamos obligados a señalar una parada de color.

La parada de color la podremos hacer en cualquiera de los colores del degradado, pero claro, tendremos que asegurarnos que el último color tenga una parada de color menor que el tamaño del elemento, o si trabajamos con porcentajes, menor que el 100%. Así, en el espacio del elemento que sobre después de la última parada de color, comenzará la repetición del degradado.

Vamos ahora a mostrar varios ejemplos de degradados con repetición:

```
background: repeating-linear-gradient(#fff, #666 25%);
```

Este degradado comienza en blanco y termina en gris. Como no se indicó nada, irá de arriba a abajo, en un gradiente vertical en línea recta. Pero lo importante en este caso es que el segundo color tiene una parada al 25%. Eso quiere decir que se llegará al gris en el primer 25% del espacio del elemento y que a partir de ese punto comenzará de nuevo el degradado. El segundo degradado ocupará otro 25% de la imagen y entonces se volverá a repetir. Por

17

tanto, en la práctica veremos que este degradado de blanco a gris se repetirá por 4 veces en el fondo del elemento donde lo coloquemos.

```
background: repeating-linear-gradient(left, #ffc, #f96 30%, #963 40%, #630 51%);
```

En este segundo ejemplo tenemos otro degradado, esta vez con 4 colores. Como se puede ver, se le han asignado varias paradas de color, en lugar de solo una al último elemento. Como el último color stop está al 51% del elemento, el degradado se verá solo dos veces.

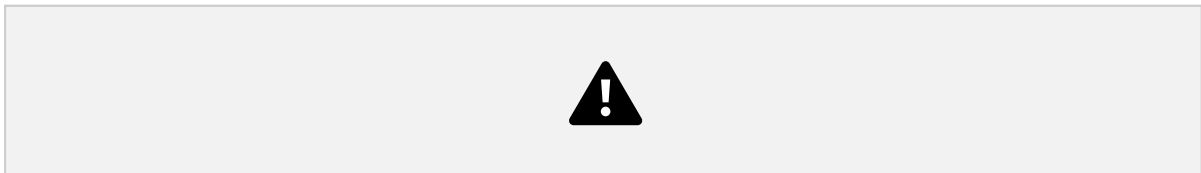
```
background: repeating-linear-gradient(45deg, red, blue, red 50px);
```

Esta es otra declaración de estilos, con un degradado en un gradiente oblicuo. Además, tiene la particularidad que se va de rojo a azul y luego de nuevo a rojo. Con ello conseguimos que las repeticiones del degradado siempre están suavizadas y no se note cuando comienza y acaba una repetición, como nos ocurría en los dos ejemplos anteriores. Además, como se puede ver, la parada de color la hemos colocado en 50px, lo que quiere decir que el degradado se repetirá cada 50 píxeles, de modo que, el número de repeticiones variará dependiendo del tamaño del contenedor donde se asigne este fondo.

```
background: repeating-linear-gradient(left, #f00, #f80, #ff0, #0f0, #00f, #60f, #c0f, #f00 100px);
```

Este último ejemplo tiene los colores del arcoiris repetidos cada 100 píxeles.

Se puede ver los distintos degradados realizados en la siguiente imagen.



18

### 1.2.3 Degradados en un gradiente de color de forma radial, círculos y elipses.

Los degradados radiales, que incluyen tanto los que tiene forma circular en general, como los que tienen forma de elipse, se consiguen a través del atributo radial-gradient. De modo que, mediante la aplicación de distintos parámetros, conseguiremos todas las posibilidades. Como veremos a continuación, tienen una forma de definirse muy parecida a la que vimos con los degradados lineales, aunque en este caso tendremos algunos otros parámetros a tener en cuenta, lo que puede dificultar un poquito más su entendimiento.

La sintaxis resumida será la siguiente:

```
background: radial-gradient(parámetros);
```

Los parámetros que podemos indicar en la declaración radial-gradient() es donde realmente

radica la dificultad y a la vez la potencia de este tipo de degradados. No obstante, la mayoría de los parámetros son opcionales, por lo que podemos hacer degradados radiales bastante simples, que tomarán parámetros por defecto. En realidad, como veremos, lo único que necesitaremos siempre es definir dos o más colores para realizar el gradiente de color.

El listado de parámetros que podremos indicar es el siguiente:

#### **A) Posición inicial del gradiente circular:**

Los degradados radiales comienzan en un punto cualquiera del fondo de un elemento y se extienden hacia fuera de ese punto con formas circulares o de elipse. Luego, para definirlos, necesitaremos una forma de especificar dicho punto de inicio del degradado. El punto se especifica con una o dos coordenadas, que pueden tener distintas unidades CSS. Si se omite, se entiende que el degradado tiene que comenzar en el punto central del fondo del elemento.

#### **B) Forma y/o dimensión:**

La forma puede ser circular o elipse, para lo cual especificamos las palabras "circle" o "ellipse". El tamaño lo expresamos con otra serie de palabras clave, que indican hasta donde debe crecer el círculo o elipse: closest-side | closest-corner | farthest-side | farthest-corner | contain | cover. Por ejemplo, closest-side indica que el círculo o elipse debe crecer hasta el lado más cercano. La palabra farthest-corner indicaría que debe crecer hasta la esquina más lejana. Contain sería lo mismo que decir closest-side y cover sinónimo de farthest-corner.

#### **Alternativa a B) Tamaño:**

19

De manera alternativa a especificar la forma y dimensión del degradado -punto B) anterior-, podemos indicar un par de medidas en cualquier unidad CSS o porcentajes. Esas medidas se utilizarían para generar un círculo o una elipse del tamaño deseado para nuestro gradiente. La primera medida sería para la anchura de la elipse y la segunda sería para la altura (si ambas son iguales se mostraría una forma circular en el degradado. Si son distintas, sería una elipse. El tamaño debe ser siempre positivo.

#### **C) Colores del degradado:**

Para acabar, se deben indicar cuantos colores se deseen, separados por comas, con la posibilidad de indicar las paradas de color que se deseen.

Ahora, veamos una serie de ejemplos que esperamos aclare las ideas con respecto a la declaración de fondos radiales.

```
background: radial-gradient(#0f0, #06f);
```

Esto hace un degradado desde el verde al azul turquesa, con todos los otros parámetros predeterminados. Haría un gradiente de forma circular, con su punto de inicio en el centro del elemento, en verde, haciendo que se llegase al azul turquesa en los bordes del elemento.

```
background: radial-gradient(top left, #fff, #f66);
```

En este caso hemos definido el punto de inicio del gradiente con "top left". Se trata de la esquina superior izquierda, donde aparecerá el blanco y el degradado tendría forma circular tendiendo hacia rosa, ocupando el 100% del elemento.

```
background: radial-gradient(200px 30px, #fff, #000);
```

Este degradado también declara la posición inicial del gradiente, pero lo hace mediante las coordenadas definidas con medidas en píxeles. Es circular y ocupa el 100% del espacio disponible en el elemento.

```
background: radial-gradient(center, #fff, #000 50%);
```

En este declaramos la posición inicial con center, el comportamiento predeterminado, que coloca el inicio del degradado en centro, tanto vertical como horizontal. El detalle es que el degradado se realiza desde el centro hasta el 50% del tamaño del elemento, ya que le hemos puesto una parada de color ("color stop") de 50% en el último color.

```
background: radial-gradient(circle, #66f, #f80, #ffc);
```

Este es el primero de los ejemplos en el que definimos la forma del degradado, aunque solo indicamos "circle". Por tanto, el degradado comenzará en el centro y ocupará el 100% del espacio disponible en el elemento, aunque siempre con el mismo radio.

```
background: radial-gradient(ellipse cover, #66f, #f80, #ffc);
```

Este degradado es exactamente igual que el anterior, pero en vez de circular es de elipse, cubriendo el 100% del espacio disponible, y comenzando en el centro. Este es el comportamiento predeterminado del estilo.

```
background: radial-gradient(10%, ellipse closest-side, #fff 60%, #f80 85%, #ffc);
```

20

Este ejemplo tiene definida la posición del degradado y la forma. Es el primero que especifica esos dos valores al mismo tiempo. En este caso, sobre la posición solo se declara 10%, así que aparecerá centrado en la vertical y en la horizontal aparecerá en el 10% del espacio del contenedor por la parte de la izquierda. Es de forma de elipse y closest-side significa que se expande en forma de elipse hasta completarse en el lado más cercano.

```
background: radial-gradient(10%, ellipse farthest-corner, #66f 60%, #f80 85%, #ffc);
```

Este degradado es igual que el anterior, en la misma posición y de forma de elipse, pero el tamaño se ha definido con farthest-corner, con lo que el degradado será mucho mayor, expandiéndose hasta la esquina más lejana.

```
background: radial-gradient(20px 100px, 30% 80%, #fff, #66f, #66f);
```

En este caso hemos definido la posición inicial con las coordenadas en píxeles y, lo que resulta novedad, hemos definido tanto la forma como el tamaño del degradado en porcentaje. La anchura será el 30% del elemento y la altura el 80%.

```
background: radial-gradient(top left, 150px 100px, #ffc, #f96, #963, #630);
```

Hemos definido la posición inicial por medio de los valores top y left y el tamaño por medio de

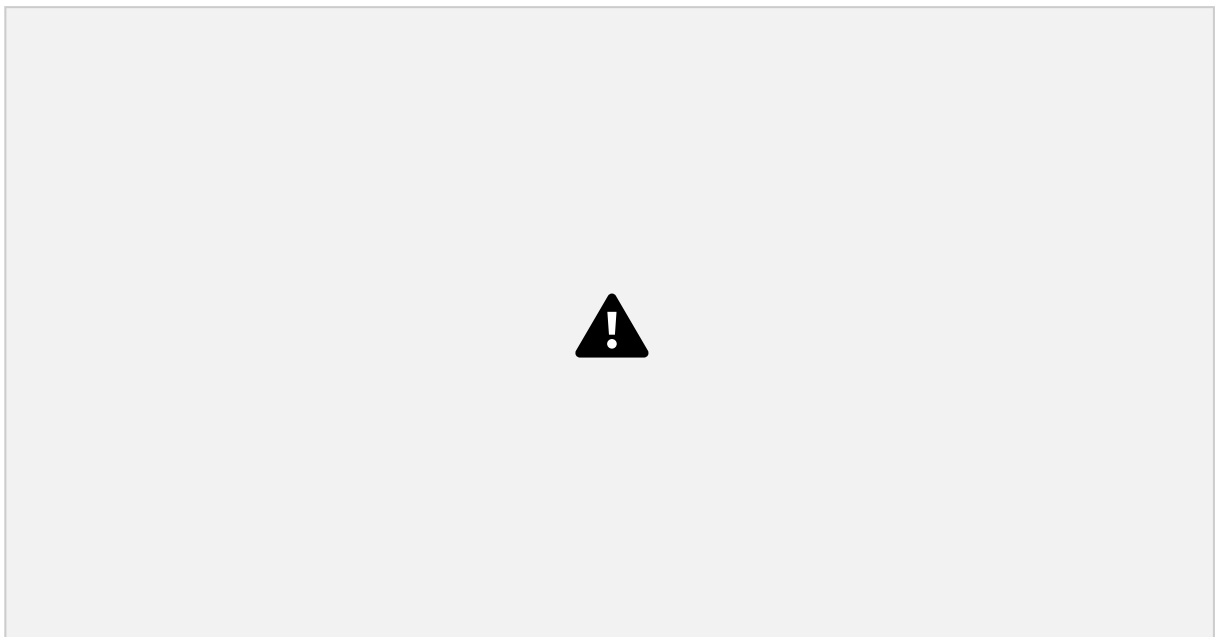
unidades en píxeles.

```
background: radial-gradient(20% 80%, 100% 50%, red, blue 50px, red);
```

La posición del centro del degradado está con porcentaje, así como el tamaño de la elipse, también con porcentajes.

```
background: radial-gradient(left, 200px 200px, #f00 20%, #f80, #ff0, #0f0,  
#00f, #60f, #c0f);
```

En este hemos definido el tamaño de la anchura y altura con dos valores en píxeles, pero como son iguales, en lugar de una elipse veríamos una forma circular, cuyo radio es siempre igual.



21

## 1.2.4 degradados radiales, con repeticiones, fondos con gradientes de color.

Como veníamos diciendo, una de las ventajas de estas nuevas características de las CSS3 es que nos permiten realizar fondos para páginas web con degradados, pero sin tener que usar imágenes. Eso consigue ahorrarnos tiempo de desarrollo y diseño, a la vez que aligera el peso de las páginas web. Pero como curiosidad, en el caso de los degradados radiales con repeticiones, además habría que añadir que serían casi imposibles de realizar usando imágenes de fondo, o por lo menos tendríamos que usar archivos gráficos muy grandes y pesados.

Los degradados radiales con repetición son casi idénticos a los degradados radiales normales,. La sintaxis básica de este tipo de degradados es la siguiente:

```
repeating-radial-gradient(parámetros)
```

Los parámetros, tales como centro del degradado radial, forma y tamaño del motivo y los colores son exactamente los mismos que para los degradados radiales. Las únicas diferencias es que tenemos que utilizar el atributo `repeating-radial-gradient`. Además, para que se produzca la repetición con el tamaño o intervalo que nosotros deseemos, tendremos que asignar alguna parada de color al último de los colores del degradado, que generalmente tendrá un valor menor del 100% del espacio del elemento.

A continuación podemos ver una serie de ejemplos:

```
background: repeating-radial-gradient(circle, #fff, #666 25%);
```

Este ejemplo hace un degradado entre dos colores, cuyo segundo elemento tiene una parada de color en el 25%. Por ello el resultado producirá el mismo degradado repetido 4 veces, una en cada 25% del espacio del elemento donde se coloque.

```
background: repeating-radial-gradient(left, circle, #ffc, #f96 30%, #963 40%, #630 51%);
```

Este degradado hace un gradiente entre varios colores y el último de ellos tiene una parada en el 51%. Esto quiere decir que el degradado se repetirá dos veces.

```
background: repeating-radial-gradient(20% 80%, ellipse closest-side, red, blue, red 50px);
```

En este ejemplo hacemos un degradado con forma de elipse y va desde rojo a azul y luego de nuevo a rojo. Al comenzar y acabar en rojo, se consigue que las repeticiones del degradado no tengan saltos bruscos de un color al otro. Como se puede ver, el último color tiene una parada de color en 50 píxeles, con lo que el degradado se repetirá de nuevo cada 50 puntos en la pantalla. El número de repeticiones dependerá del tamaño del elemento donde se coloque este degradado.

```
background: repeating-radial-gradient(left, ellipse farthest-side, #f00, #f80, #ff0, #0f0, #00f, #60f, #c0f, #f00);
```

Este otro caso produce una repetición del degradado, pero no hemos colocado ninguna parada de color en el último elemento. Esto quiere decir que el degradado se repetirá, pero el intervalo de repetición en esta ocasión dependerá del tamaño y forma radial que se haya configurado. En este caso en concreto tenemos una elipse que se expande, desde la izquierda hasta el lado opuesto, con lo que sólo en una pequeña porción del fondo se verá la repetición de los colores.

22



## 1.2.5 Generadores de gradientes CSS.

Afortunadamente contamos con herramientas online que nos facilitan la labor anterior de crear un degradado de colores. Una de estas es: <https://cssgradient.io/>



## 1.2.6 Patrones creados con CSS.

Un patrón no es más que un elemento, que se repite muchas veces. Si hablamos de patrones CSS, en este caso nos referimos a elementos que van a dar estilo a nuestra página web. Una forma bastante habitual de usar estos patrones, es creando fondos visualmente impactantes. Aunque se pueden utilizar en multitud de sitios, como banners, cabeceras, anuncios, etc.

Se pueden crear patrones usando los gradientes CSS, es decir en vez de usar una imagen (en el formato que sea) como background en las páginas, usan un gradiente de css.

Básicamente lo que hay que hacer es definir un color de base con la propiedad background-color y después definir por encima el patrón usando reglas como linear-gradient, radial-gradient o repeating-linear-gradient sobre background-image.

```
background-image:linear-gradient(#333 20%, transparent 25%),
                  linear-gradient(90deg, #333 20%, transparent 25%);

background-size:20px 20px;
```

23

### VELOCIDAD

El tener que importar un recurso a nuestra web siempre es más lento, incluso aunque se trate de un archivo pequeño como suelen ser los patrones, ya que el solo hecho de realizar una llamada es lo que nos penaliza.

Por otra parte puedes incrustar la imagen en base64 dentro de la propia hoja de estilos y evitar la llamada pero incluso de esta manera sería más lento y estaríamos subiendo el

peso del archivo notablemente.

```
background:url(
data:image/png;base64,
```



```
NgGwSgwVAFbmAgXQdISfAAAAAE1FTkSuQmCC
) repeat;
```



Patternify nos permite

crear patrones en base64 embebidos en el código:

24

Crear patrones con css requiere de un trabajo de diseño y programación avanzado, sobretodo cuando los patrones son ya bastante complejos, en contrapartida tiene grandes ventajas sobre el uso de imágenes de fondo.





**Patternizer**(generador de patrones) **Galería de Patrones:** (ejemplos de patrones ).

Si necesitamos patrones más complejos que no podemos desarrollar con CSS, nos quedará el último recurso de utilizar imágenes png o svg.

Recursos:





Compíte y aprende css a través de retos...