

Colores CSS

Uno de los primeros cambios de estilo que podemos pensar realizar en un documento HTML es hacer variaciones en los colores de primer plano y de fondo. Esto es posible con las primeras dos propiedades que veremos a continuación:

Propiedad	Valor	Significado
<code>color</code>		Cambia el color del texto que está en el interior de un elemento.
<code>background-color</code>		Cambia el color de fondo de un elemento.

La propiedad `color` establece el color del texto, mientras que la propiedad `background-color` establece el color de fondo del elemento.

Todas las propiedades CSS donde existen valores , establecen la posibilidad de indicar **4 formas alternativas** (*con algunos derivados*) para especificar el color deseado:

Nombre	Formato	Ejemplo
Palabra clave predefinida	<i>[palabra clave]</i>	<code>red</code>
Esquema RGB	<code>rgb(rojo, verde, azul)</code>	<code>rgb(255, 0, 0)</code>
Esquema RGB con canal alfa	<code>rgba(rojo, verde, azul, alfa)</code>	<code>rgba(255, 0, 0, 0.25)</code>
Esquema RGB hexadecimal	<code>#RRGGBB</code>	<code>#ff0000</code>
Esquema RGB hexadecimal con canal alfa	<code>#RGBBBA</code>	<code>#ff000040</code>
Esquema HSL	<code>hsl(color, saturación, brillo)</code>	<code>hsl(0, 100%, 100%)</code>
Esquema HSL con canal alfa	<code>hsla(color, saturación, brillo, alfa)</code>	<code>hsla(0, 100%, 100%, 0.25)</code>

A continuación iremos explicando cada uno de estos formatos para entender como se especifican los colores en CSS y utilizar el método que más se adapte a nuestras necesidades.

Consejo: Si lo que buscamos es un sistema para extraer colores (*eye dropper*) de una página web, podemos utilizar la extensión para Chrome de [ColorZilla](#) o el propio **Chrome Developer Tools**, que integra dicha funcionalidad.

Palabras clave de color

El primer caso (*y más limitado*) permite establecer el color utilizando palabras reservadas de colores, como **red**, **blue**, **orange**, **white**, **navy**, **yellow** u otras. Existen más de 140 palabras clave para indicar colores:

black
navy
darkblue
mediumblue
blue
darkgreen
green
teal
darkcyan
deepskyblue
darkturquoise
mediumspringgreen
lime
springgreen
aqua
cyan
midnightblue
dodgerblue
lightseagreen
forestgreen
seagreen
darkslategray
darkslategrey
limegreen
mediumseagreen
turquoise
royalblue
steelblue
darkslateblue
mediumturquoise
indigo
darkolivegreen
cadetblue
cornflowerblue
mediumaquamarine
dimgray
dimgrey
slateblue
olivedrab
slategray
slategrey
lightslategray
lightslategrey
mediumslateblue
lawngreen
chartreuse
aquamarine
maroon
purple
olive

gray	
grey	
skyblue	
lightskyblue	
blueviolet	
darkred	
darkmagenta	
saddlebrown	
darkseagreen	
lightgreen	
mediumpurple	
darkviolet	
rebeccapurple	
palegreen	
darkorchid	
yellowgreen	
sienna	
brown	
darkgray	
darkgrey	
lightblue	
greenyellow	
paleturquoise	
lightsteelblue	
powderblue	
firebrick	
darkgoldenrod	
mediumorchid	
rosybrown	
darkkhaki	
silver	
mediumvioletred	
indianred	
peru	
chocolate	
tan	
lightgray	
lightgrey	
thistle	
orchid	
goldenrod	
palevioletred	
crimson	
gainsboro	
plum	
burlywood	
lightcyan	
lavender	
darksalmon	
violet	

palegoldenrod
lightcoral
khaki
aliceblue
honeydew
azure
sandybrown
wheat
beige
whitesmoke
mintcream
ghostwhite
salmon
antiquewhite
linen
lightgoldenrodyellow
oldlace
red
fuchsia
magenta
deeppink
orangered
tomato
hotpink
coral
darkorange
lightsalmon
orange
lightpink
pink
gold
peachpuff
navajowhite
moccasin
bisque
mistyrose
blanchedalmond
papayawhip
lavenderblush
seashell
cornsilk
lemonchiffon
floralwhite
snow
yellow
lightyellow
ivory
white

Además, existen algunos valores especiales que puedes utilizar cuando quieras especificar un color, como colores **transparentes** o el **color actual del texto**, muy útil para SVG, por ejemplo:





Valor	Significado
Transparent	Establece un color completamente transparente (valor por defecto de <code>background-color</code>)
<code>currentColor</code>	Establece el mismo color que se está utilizando para el texto (CSS3 y SVG)

Veamos algunos ejemplos de palabras clave de color:

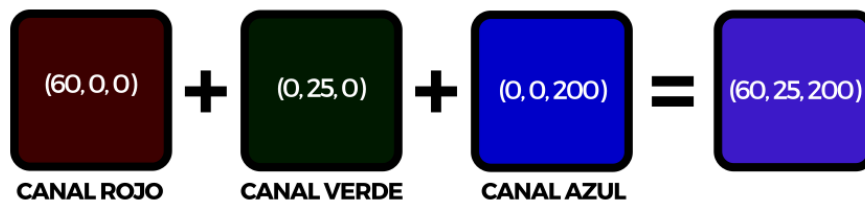
```
div {  
  background-color: blue;  
  background-color: transparent;  
  background-color: lightpink;  
  background-color: rebeccapurple; /* En honor a Rebeca, la hija de  
Eric Meyer */  
}
```

Formato RGB

Uno de los métodos más conocidos por los diseñadores gráficos es utilizar el **formato RGB**. Las siglas RGB significan **rojo**, **verde** y **azul**, por lo que cada cifra (*del 0 al 255*) representa la intensidad de cada componente de color. Como se puede ver en la siguiente imagen, si utilizamos una cantidad (0, 0, 0) de cada canal, obtenemos el **color negro**. En cambio, si utilizamos una cantidad (255, 0, 0), obtendremos el **color rojo**.

Rojo	0	255	255	0
Verde	0	255	0	0
Azul	0	255	0	255
				

De esta forma, mezclando las cantidades de cada canal, se puede obtener prácticamente cualquier color. Existen muchos esquemas de colores, pero en diseño web nos interesa particularmente el esquema RGB (*junto al HSL*).



La mayoría de los editores tienen los denominados **ColorPicker**, que no son más que un sistema cómodo y rápido para elegir un color a base de clics por una paleta o círculo visual. También podemos hacerlo directamente en buscadores como [Duck Duck Go](#) o [Google](#).

Veamos algunos ejemplos de colores en formato RGB:

```
div {
  background-color: rgb(125, 80, 10);
  background-color: rgb(0, 0, 34);
  color: rgb(255, 255, 0)
}
```

Formato hexadecimal

El **formato hexadecimal** es el más utilizado por los desarrolladores web, aunque en principio puede parecer algo extraño y complicado, sobre todo si no has oído hablar nunca del sistema hexadecimal (*sistema en base 16 en lugar del que utilizamos normalmente, en base 10*).

Cada par de letras simboliza el valor del RGB en el sistema de numeración hexadecimal, así pues, el color **#FF0000**, o sea HEX(FF,00,00), es equivalente al RGB(255,0,0), que es también equivalente al HSL(0, 100%, 100%). Veamos algunos ejemplos para clarificarlo:

Hexadecimal	Hex. abreviado	Color RGB	Palabra clave
#FF0000	#F00	255,0,0	red (rojo)
#000000	#000	0,0,0	black (negro)
#00FFFF	#0FF	0, 255, 255	cyan (azul claro)
#9370DB	#97D	147,112,219	mediumpurple (lila)

[HailPixel](#) nos proporciona una manera muy sencilla y rápida de seleccionar tonalidades de color en **formato hexadecimal** con sólo mover el ratón. Por otro lado, en [ColorSpire](#) puedes seleccionar el color deseado y observar como varían tanto los valores hexadecimales como los valores RGB o HSL (*ver a continuación*).

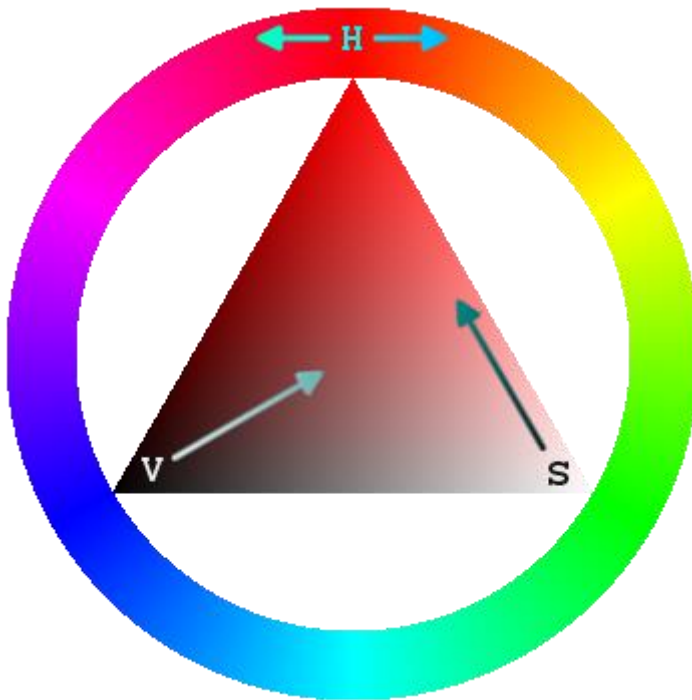
Truco: Como se puede ver en la segunda columna, para ahorrar espacio puedes utilizar el **formato hexadecimal abreviado**, especificando sólo las primeras tres cifras de cada par. Por ejemplo, #9933AA como #93A. El color abreviado sólo será fiel cuando los pares de cifras sean idénticos (o al menos, aproximados).

Veamos algunos ejemplos del formato hexadecimal (RGB abreviado):

```
div {
  background-color: #512592;
  background-color: #000000;
  background-color: #451;          /* Equivalente a #445511; */
}
```

Formato HSL

Las siglas HSL significan **color** (o *matiz*), **saturación** y **brillo**. La primera cifra selecciona el matiz de color (*una cifra de 0 a 360 grados*), seleccionando el color del círculo exterior de la imagen. Por su parte, las dos siguientes, son el porcentaje de saturación y el brillo del color, respectivamente (*ambos, porcentajes de 0% a 100%*).



Veamos algunos ejemplos del formato HSL:

```
div {
  background-color: hsl(35deg, 0%, 100%);
  background-color: hsl(120deg, 25%, 75%);
  background-color: hsl(5deg, 20%, 20%);
}
```

Canales Alfa

Es posible que deseemos indicar un color que tenga cierto grado de transparencia, y de esta forma, refleje el contenido, color o imágenes que se encuentren detrás. Hasta ahora solo conocemos la palabra clave `transparent`, que es un color de transparencia total (*totalmente transparente*).

Sin embargo, existe la posibilidad de utilizar los denominados **canales alfa**, que permiten establecer una **transparencia parcial** en determinados colores. Estos se pueden establecer en cualquier formato, salvo en los colores con palabras clave. Vamos a ver como hacerlo en cada caso:

- **Formato RGB:** En lugar de `rgb()` indicamos `rgba()` para establecer que usaremos un canal alfa. Posteriormente, en lugar de establecer 3 parámetros (*rojo, verde, azul*), añadiremos uno más, que será el canal alfa. Dicho canal alfa será un valor (*del 0 al 1 con decimales*) o un porcentaje (*del 0% al 100%*).
- **Formato HSL:** Prácticamente idéntico al anterior. En lugar de `hsl()` indicamos `hsla()`. Añadimos un nuevo valor como canal alfa (*valor o porcentaje*).
- **Formato Hexadecimal:** Es posible indicar (*al final*) un par adicional que indique el grado de transparencia. Por ejemplo, el color `#FF0000` reescrito como `#FF000077` se trataría de dicho color, con un grado de transparencia casi del 50% (00 es 0%, 80 es 50%, FF es 100%).

Veamos algunos ejemplos de cada caso:

```
div {
  background-color: rgba(0, 0, 0, 0.5);
  background-color: rgba(0, 0, 0, 50%);
  background-color: hsla(180deg, 50%, 25%, 0.75);
  background-color: hsla(180deg, 50%, 25%, 75%);
  background-color: #aa44ba80;
}
```

¡OJO! El formato de [transparencia en formato hexadecimal](#) se encuentra actualmente bien soportado, pero puede no ser compatible en versiones más antiguas u otros navegadores.

Consejo: Una herramienta genial para seleccionar varios colores en nuestros proyectos es [Adobe Color CC](#). Nos permite escoger entre colores análogos, monocromáticos, tríadas, colores complementarios, compuestos o tonos similares. Muy visual e intuitiva. Otra herramienta, más simple pero muy práctica es [HSL Picker](#), donde puedes elegir el color deseado utilizando el **formato de colores HSL** y pudiendo seleccionar incluso los canales alfa.

Fondos en CSS

Es posible que, buscando hacer un diseño más avanzado, en lugar de utilizar un color de fondo quieras utilizar imágenes. Para ello, CSS te proporciona la propiedad `background-image`, con la cuál puedes indicar imágenes de fondo o, como veremos más adelante, incluso degradados de varios colores.

Imágenes de fondo

En el caso de querer utilizar una imagen de fondo, como ya hemos dicho, utilizaremos la propiedad `background-image` y en el valor, el nombre de la imagen (*o la dirección donde está alojada*), siempre rodeada del texto `url()`.

Propiedad	Valor	Significado
<code>background-image</code>	none	No utiliza ninguna imagen de fondo.
<code>background-image</code>	<code>url(imagen.jpg)</code>	Usa la imagen de nombre imagen.jpg como fondo.

En el caso de que sólo se coloque el nombre de la imagen (*por ejemplo, `imagen.jpg`*), el navegador buscará la imagen en la misma carpeta donde está el archivo CSS. Esto es lo que se llama una **ruta relativa**. En el caso de que se coloque la ruta completa, por ejemplo `https://i.emezeta.com/img/logo.png`, se accederá a la imagen alojada en esa dirección web. Esto es lo que se llama **ruta absoluta**.

NOTA: En el caso de que no se encuentre la imagen o el valor de **`background-image`** se haya establecido a **none**, no se utilizará ninguna imagen de fondo, y en su lugar se mostrará el color establecido con **`background-color`**.

Una vez establecida una imagen de fondo con `background-image`, se puede personalizar la forma en la que se mostrará dicha imagen mediante propiedades como `background-repeat`, `background-attachment` o `background-position`, entre otras:

Propiedad	Valor	Significado
<code>background-repeat</code>	repeat	Repite la imagen de fondo horizontal y verticalmente.
	<code>repeat-x</code>	Repite la imagen de fondo sólo horizontalmente (eje x).
	<code>repeat-y</code>	Repite la imagen de fondo sólo verticalmente (eje y).
	<code>space</code>	Repite la imagen y rellena con espacio los huecos.
	<code>round</code>	Repite la imagen y amplía cada repetición para ajustar.
	<code>no-repeat</code>	La imagen de fondo no se repite.
<code>background-attachment</code>	scroll	Cuando hacemos scroll la imagen de fondo se

Propiedad	Valor	Significado
background-position		desplaza.
	fixed	Cuando hacemos scroll, la imagen de fondo permanece fija.
		1 parámetro. Desplaza la imagen de fondo al punto (x, 50%).
		2 parámetros. Desplaza la imagen de fondo al punto (x, y).

La propiedad `background-repeat` especifica si la imagen se repetirá horizontalmente (*repeat-x*), si se repetirá verticalmente (*repeat-y*), si lo hará en ambas direcciones (*repeat*) o en ninguna (*no-repeat*). Por defecto, si no se indica nada, esta propiedad está ajustada en **repeat**.

Existen también dos valores interesantes, **space** y **round**, los cuales asumen implícitamente que se repite el fondo. En el caso de que tengamos una imagen de fondo que se repita varias veces en mosaico, `space` evita que se corte la imagen, introduciendo un espacio entre las repeticiones individuales.

Por su parte, `round` lo que hace es ajustar la imagen individual, de modo que la expande o contrae para ajustarla al espacio disponible. En ambos casos la repetición de los fondos nunca se mostrará cortada.

background-repeat: repeat



background-repeat: space



background-repeat: round



Siempre se podrá combinar en cada eje, con valores mixtos, por ejemplo, utilizando `background-repeat: space round`, lo que aplicará `space` al eje X y `round` al eje Y. Si sólo se especifica uno, se aplicará a ambos ejes.

La propiedad `background-attachment` especificará si la imagen de fondo seguirá el desplazamiento del usuario (*scroll*, *el comportamiento por defecto*) o por el contrario, se quedará fijado y no se moverá (*fixed*), mientras el usuario se desplaza por la página.

Por último, `background-position` coloca la imagen en la zona especificada por `x` y `y`. Por defecto, esos valores son **0% 0%**, pero pueden especificarse con unidades

(porcentajes, píxels, etc...) o mediante palabras clave que representan zonas predefinidas (*top, left, right, bottom y center*).

Si sólo se especifica un valor, se tomará para el eje x, mientras que el valor del eje Y será automáticamente establecido a **center** (o 50%).

Atajo clásico: Fondos

Es posible establecer todas estas propiedades anteriores en una sola regla de CSS a modo de atajo, y así ahorrar mucho espacio en escribir las propiedades anteriores por separado. Aunque no es estrictamente obligatorio, se aconseja respetar el siguiente orden (*acostumbrarse a usar el mismo orden es una buena práctica*):

```
div {  
  /* background: <color> <image> <repeat> <attachment> <position> */  
  background: #FFF url(imagen.jpg) repeat-x scroll top left;  
}
```

Y con esto, ya conocemos las ventajas básicas de CSS a través de propiedades tan interesantes como `background-color` o `background-image`. Sin embargo, la llegada de CSS3 incorporó novedades muy interesantes en nuestros navegadores, que mediante versiones anteriores a CSS3 no era posible realizar (*o era algo bastante complejo*).

Opacidad

Es posible utilizar la propiedad CSS3 `opacity` para establecer una transparencia total sobre el elemento indicado. Cuando decimos «**transparencia total**» nos referimos a que la transparencia se aplicará al elemento en cuestión y a todos los elementos HTML que estén en su interior.

Propiedad	Valor	Significado
<code>opacity</code>		Establece una transparencia (0 = 100% transparente, a 1 = 100% opaco)

El valor a indicar es un número entre 0 (*completamente transparente*) y 1 (*completamente visible*), pudiendo establecer decimales para valores intermedios:

```
div {  
  background-color: #FF0000;  
  color: #FFFFFF;  
  opacity: 0.5;  
}
```



Como se puede ver en la imagen, esto hará que la capa `div` (*el recuadro rojo*) se muestre al **50% de opacidad**, con color de texto blanco (*en el caso de existir texto*) y fondo de color rojo. Si buscamos una transparencia parcial (*color de fondo transparente que no afecte al texto*) debemos utilizar los valores **RGBA** o **HSLA** (*canales alfa*) en la propiedad **background-color**:

```
div {  
  color: white;  
  background-color: RGBA(255, 0, 0, 0.5);  
}
```

De este modo, sólo aplicamos la transparencia al color de fondo, mientras que con **opacity** se aplica a toda la capa en sí, por lo que afecta a todos los elementos que están dentro de la capa.

Fondos múltiples

CSS3 ofrece nuevas características a la hora de utilizar imágenes de fondo, como por ejemplo la posibilidad de establecer **múltiples imágenes de fondo** de forma simultánea:

```
div {  
  background-image: url(manz.png), url(fondo2.jpg), url(fondo3.jpg);  
  background-repeat: no-repeat;  
}
```

De esta forma, se pueden utilizar varias imágenes y superponerlas una sobre la otra, algo especialmente interesante si la primera imagen de fondo está en formato PNG (*la cuál soporta transparencias*). Al establecer imágenes de fondo múltiples, las propiedades complementarias a los fondos como `background-repeat`, `background-position` y otras, pueden actuar de forma personalizada para cada fondo.

Mientras que en el fragmento de código anterior, el navegador le indica a cada una de las tres imágenes que no debe repetirse, en el siguiente fragmento de código veremos que es posible indicar individualmente el comportamiento de cada una, separando por comas:

```
div {  
  background-image: url(manz.png), url(fondo2.jpg), url(fondo3.jpg);  
  background-repeat: no-repeat, repeat-x, repeat;  
}
```

Además, **CSS3** también añade nuevas propiedades para especificar como cubrirá la imagen de fondo al elemento en cuestión:

Propiedad	Valor	Significado
<code>background-clip</code>	border-box <code>padding-box</code> <code>content-box</code>	Modo de relleno de la imagen de fondo
<code>background-origin</code>	<code>border-box</code> padding-box <code>content-box</code>	Origen del modo de relleno del fondo

La propiedad `background-clip` establece la forma en la que el color o la imagen de fondo cubrirá el elemento, mientras que la propiedad `background-origin` intenta posicionar el comienzo de la imagen de fondo, útil con imágenes. La primera utiliza **border-box** como valor por defecto, mientras que la segunda utiliza **padding-box**.

Ambas propiedades pueden tomar uno de los siguientes valores:

Valor	Significado
<code>padding-box</code>	La imagen o color de fondo cubrirá la zona del espaciado y contenido.
<code>border-box</code>	La imagen o color de fondo cubrirá la zona del borde, espaciado y contenido.
<code>content-box</code>	La imagen o color de fondo cubrirá sólo la zona del contenido.

Consejo: Una buena forma de darse cuenta del funcionamiento de estas propiedades es establecer un borde grueso punteado. Usando `border-box` la imagen de fondo se extenderá en todo el elemento, incluyendo borde, espaciado y contenido. El valor `padding-box` extenderá la imagen de fondo sólo mediante el padding y el contenido, y por último, la propiedad `content-box` extenderá la imagen de fondo sólo en la zona del contenido.

Tamaño de fondos

Una de las últimas incorporaciones a la familia de propiedades de fondos de imagen es la propiedad `background-size`, la cuál ajusta el tamaño (*ancho* y *alto*) de la imagen de fondo, por si deseamos escalarla a un tamaño diferente. Por defecto, una imagen de fondo toma automáticamente el tamaño de la imagen (*que podría ser demasiado grande, por ejemplo*). Para no tener que modificar la imagen original de forma manual con un editor de imágenes, podemos utilizar esta propiedad y ajustarla a nuestro agrado mediante CSS:

Propiedad	Valor	Significado
<code>background-size</code>	1 parámetro. Aplica un de (<i>ancho</i> × auto) a la imagen de fondo.	
	2 parámetros. Aplica un de (<i>ancho</i> × <i>alto</i>) a la imagen de fondo.	

Los valores de tamaño que podemos utilizar, son los siguientes:

Valor	Significado
auto	No escala la imagen. Utiliza el tamaño original. Es el valor por defecto.
<i>unidad</i>	Indicamos el tamaño específico que queremos usar (<i>píxels o porcentaje, por ej.</i>).
<code>cover</code>	Escala el ancho de la imagen de fondo al ancho del elemento.
<code>contain</code>	Escala el alto de la imagen de fondo al alto del elemento.

Los dos últimos valores (*cover* y *contain*) sólo pueden utilizarse en el caso de que se especifique un sólo parámetro como valor en la propiedad `background-size`.

Atajo moderno: Fondos

Los navegadores modernos, incluyen una nueva propiedad de atajo `background` que permite incluir los valores de propiedades CSS3 como `background-clip`, `background-origin` o `background-size`, que fueron incorporados más tarde que los demás. Es por ello, que el orden aconsejado para adquirir buenas prácticas es el siguiente y varía un poco respecto a la otra propiedad de atajo tradicional:

```
div {
  /* background: <color> <position> <size> <repeat> <origin> <clip>
  <attachment> <image> */
  background: #FFF top left cover repeat-x padding-box border-box
  scroll url(imagen.jpg);
}
```

Gradientes CSS

Otra funcionalidad que se echaba mucho de menos en versiones anteriores de CSS era la de utilizar fondos con **gradientes**, o lo que es lo mismo, un color inicial que se transforma poco a poco en uno o varios colores sucesivamente. Sin embargo, en versiones anteriores a CSS3 no era posible hacerlo desde código. Por suerte, hoy en día hemos superado esa limitación.

En CSS3 se introducen algunas expresiones relacionadas que harán esta tarea algo muy sencillo, pudiendo utilizarlas fácilmente mediante la propiedad `background-image` e indicando el tipo de gradiente deseado:

Propiedad	Función de gradiente	Significado
<code>background-image</code>	<code>linear-gradient(...)</code>	Define un gradiente lineal, en una dirección específica.
	<code>radial-gradient(...)</code>	Define un gradiente radial, en forma de círculo o elipse.
	<code>conic-gradient(...)</code>	Define un gradiente cónico, desde un punto de vista superior.

Veamos cada uno de estos gradientes por separado para entender como funcionan.

Gradientes lineales

El primer tipo de gradiente es el **gradiente lineal**, que permite crear fondos degradados que van en una cierta dirección y cambian de un color a otro, dependiendo de la lista de colores indicada por parámetro.

La sintaxis a utilizar debe ser una de las siguientes (*es la misma, pero con menor o mayor cantidad opcional de información*):

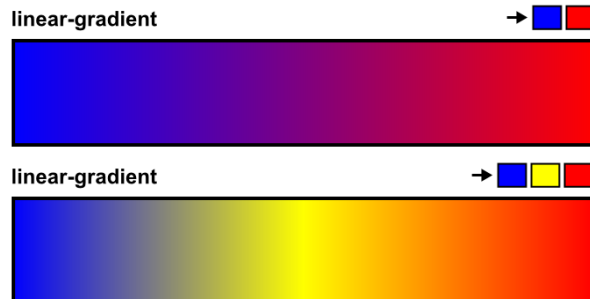
Valor	Significado
<code>linear-gradient(, , ...)</code>	Gradiente básico entre colores.
<code>linear-gradient(, , , ...)</code>	Gradiente con dirección o ángulo.
<code>linear-gradient(, , , ...)</code>	Gradiente indicando el tamaño del color.

El ejemplo más básico de los anteriores es el primero, donde simplemente indicaremos los colores del gradiente (*para que se aprecie el degradado, como mínimo deben ser 2 colores*):

```
div {  
  background-image: linear-gradient(blue, red);  
}
```

}

Esto dibujará un degradado con los colores indicados en dirección desde **arriba hacia abajo**, que es la dirección por defecto. Se pueden ir añadiendo más colores al gradiente si se desea, o cambiar la dirección del degradado, como veremos a continuación:



Pero los parámetros que se pueden utilizar son los siguientes:

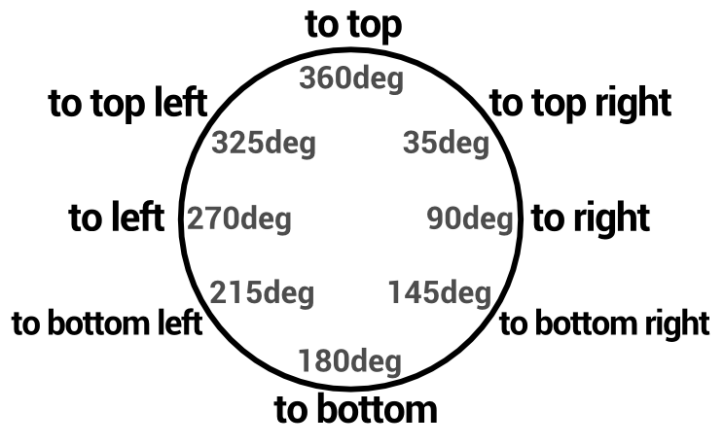
Parámetros	Valor
	Dirección del gradiente lineal (<i>ver valores posibles más adelante</i>). El valor por defecto es to bottom .
	Número de ángulos de la dirección del gradiente lineal. Alternativa a la opción anterior.
	El primer color del gradiente
	El siguiente color del gradiente
	De forma opcional, se puede indicar a que altura comienza a cambiar el gradiente.
...	(Se pueden seguir añadiendo colores)

Dirección del gradiente

En el ejemplo visual anterior, la dirección del gradiente es *horizontal* (hacia la derecha). El primer ejemplo tiene sólo 2 colores (azul y rojo), mientras que el segundo tiene 3 colores (azul, amarillo y rojo). Ambos tienen como dirección una orientación hacia la derecha:

```
div {  
  background-image: linear-gradient(to right, blue, red);          /*  
Primer ejemplo */  
  background-image: linear-gradient(to right, blue, yellow, red); /*  
Segundo ejemplo */  
}
```

Como vemos, el primer parámetro utilizado fue `to right`, que es lo que indica la orientación del gradiente. En este parámetro se puede indicar tanto una **palabra clave** como un número de grados que represente la dirección:



Para que quede aún más claro, veamos una tabla de equivalencias:

Dirección	Grados	Significado
to top	0 ó 360deg	Gradiente lineal hacia arriba.
to top right	35deg	Gradiente diagonal hacia derecha-arriba.
to right	90deg	Gradiente lineal hacia derecha.
to bottom right	145deg	Gradiente diagonal hacia abajo-derecha.
to bottom	180deg	Gradiente lineal hacia abajo.
to bottom left	215deg	Gradiente diagonal hacia abajo-izquierda.
to left	270deg	Gradiente lineal hacia izquierda.
to top left	325deg	Gradiente diagonal hacia izquierda-arriba.

Nota: Si utilizas los **grados** en lugar de las **palabras clave** puedes usar valores intermedios (*por ejemplo, 280deg o 58deg*) y ser más preciso con la dirección si lo necesitas.

Colores del gradiente

Por defecto, las distancias entre colores se ajustan automáticamente con proporciones equivalentes, salvo que especifiquemos un tamaño (*porcentaje o píxeles, por ejemplo*) justo después del color en cuestión, que se puede indicar de forma opcional:

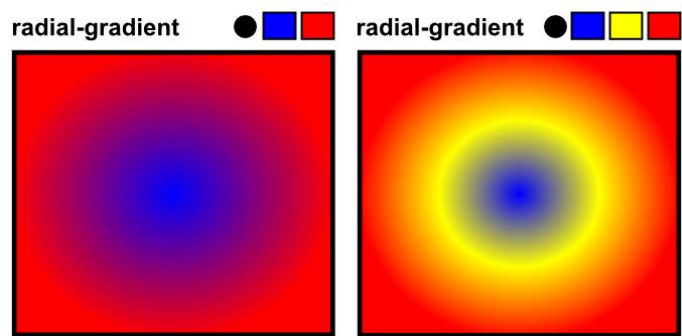
```
div {  
  background: linear-gradient(to right, blue 50%, red 55%, green 75%);  
}
```

Este tamaño representa el momento en el que el gradiente comienza a cambiar el color, respecto al tamaño del gradiente completo. En este fragmento de código, a diferencia del anterior, nos mostraría un gradiente similar al que vemos a continuación, donde se aprecia que los tamaños o distancias entre colores no son equivalentes, sino que corresponden **azul** (0-50%), **rojo** (50%-55%) y **verde** (55%-75%), el cuál permanece

hasta el final al no haber ningún color más

Gradiente radial

Como vimos al principio, a parte de gradientes lineales, también podemos crear otros tipos de gradientes, como por ejemplo, los **gradientes radiales**, que permite crear degradados con formas circulares. Para ello, sólo tenemos que utilizar la expresión `radial-gradient` en lugar de `linear-gradient`:



De la misma forma, también tenemos diferentes modalidades para crear **gradientes radiales**, desde una muy básica donde le indicamos sólo los colores, hasta otra más avanzada donde le damos forma, tamaño e incluso ubicación:

Valor	Significado
<code>radial-gradient(, , ...)</code>	Gradiente básico entre colores.
<code>radial-gradient(forma, , , ...)</code>	Gradiente con forma circular o elíptica.
<code>radial-gradient(forma , , , ...)</code>	Gradiente con tamaño o dimensión.
<code>radial-gradient(forma at ubicación, , , ...)</code>	Gradiente con colocación inicial.
<code>radial-gradient(forma at ubicación, , , ...)</code>	Gradiente con tamaños entre colores.

Algunos ejemplos aplicando estos parámetros:

```
div {
  background: radial-gradient(gold, red, black);
  background: radial-gradient(ellipse, gold 50%, red 55%, black 75%);
  background: radial-gradient(circle 200px, gold 50%, red 55%, black 75%);
  background: radial-gradient(circle 400px at left, gold 50%, red 55%, black 75%);
}
```

Donde:

Parámetros	Valor
[forma]	ellipse circle farthest-corner closest-corner farthest-side closest-side
[ubicación]	center top left right bottom top left top right bottom left bottom right El primer color del gradiente El siguiente color del gradiente De forma opcional, se puede indicar a que altura comienza a cambiar el gradiente. ...
	(Se pueden seguir añadiendo colores)

En el caso de la **forma** podemos elegir `circle` o `ellipse`, dependiendo de si queremos un gradiente con forma circular o uno con forma de elipse (*que se adapta a la forma del elemento*). En el caso del **tamaño** del elipse o círculo, tenemos varias opciones para indicar el tamaño que tendrá nuestro gradiente:

Valor	Significado
farthest-corner	Intenta cubrir hasta la esquina más lejana (<i>expande hacia ambos lados</i>).
farthest-side	Intenta cubrir hasta el lado más lejano (<i>expande hacia un sólo lado</i>).
closest-corner	Intenta cubrir hasta la esquina más cercana.
closest-side	Intenta cubrir hasta el lado más cercano. Le da un tamaño específico al gradiente (<i>píxeles, porcentajes, etc...</i>). Sólo usable en <code>circle</code> .

En el caso de la **ubicación**, simplemente especificamos la posición en la que queremos que esté colocado el centro del gradiente.

Gradiente cónico

Recientemente, se ha incorporado un nuevo tipo de gradiente muy interesante en CSS, visualmente muy similar al **gradiente radial**, pero tratándose de una visualización de un cono desde un plano superior, incluyendo sus sombras.

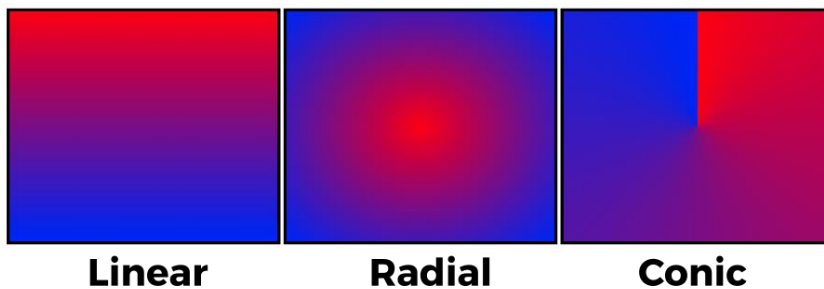
Valor	Significado
<code>conic-gradient(, , ...)</code>	Gradiente básico entre colores.

Valor	Significado
conic-gradient(from , , ...)	Gradiente indicando punto de inicio.
conic-gradient(from at <i>ubicación</i> , , , ...)	Gradiente indicando ubicación del inicio.

Donde:

Parámetros	Valor
from	Ángulo desde donde comienza el gradiente cónico. Por defecto from 0deg .
at [<i>ubicación</i>]	center top left right bottom top left top right bottom left bottom right
	El primer color del gradiente
	El siguiente color del gradiente
	De forma opcional, se puede indicar a que altura comienza a cambiar el gradiente.
...	(Se pueden seguir añadiendo colores)

A continuación podemos ver la diferencia de un gradiente `linear-gradient`, un `radial-gradient` y un `conic-gradient`:



Poco hay que añadir en este tipo de gradiente, ya que funciona muy similar a los anteriores. La única diferencia mencionable es el uso de la palabra clave **from** antes de indicar los ángulos de dirección, si se requiere.

Gradientes repetitivos

Por último, todos los gradientes que hemos visto permiten la posibilidad de añadir el prefijo `repeating-` para conseguir que el efecto del gradiente, en lugar de adaptarse a la región completa, realice una repetición constantemente. Comprueba los siguientes ejemplos de forma individual:

```
div {
  background: repeating-linear-gradient(blue 10%, yellow 20%, red 30%); /* Ejemplo 1 */
  background: repeating-radial-gradient(blue 10%, yellow 20%, red 30%); /* Ejemplo 2 */
  background: repeating-conic-gradient(blue 10%, yellow 20%, red 30%); /* Ejemplo 3 */
}
```

Es importante que al usar el prefijo `repeating`- se especifiquen tamaños a los colores, puesto que se necesita saber cuando comenzará a repetirse. En los ejemplos anteriores, los colores seguirían el patrón hasta el último color y volverían a repetirse sucesivamente, de forma que tendríamos:

- Primera repetición: **blue** 10%, **yellow** 20%, **red** 30%
- Segunda repetición: **blue** 40%, **yellow** 50%, **red** 60%
- Tercera repetición: **blue** 70%, **yellow** 80%, **red** 90%
- Cuarta repetición: **blue** 100%