



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

04-04.- Los colores y los fondos

Nombre		Curso	
Apellidos		Fecha	

CSS permite controlar el color y los fondos con unas posibilidades que están a años luz de los efectos que podemos alcanzar empleando sólo HTML.

Al utilizar estilos de forma habitual tendrás que establecer el color del los diferentes elementos,

por lo que debes conocer el modo de hacerlo.

Los colores en CSS se pueden indicar de cinco formas diferentes:

- **con palabras clave (nombres propios de los colores),**
- **empleando el sistema RGB con**
 - **numeración hexadecimal normal**
 - **numeración hexadecimal simplificada**
 - **decimal**
 - **porcentual.**
 - **Ich**
 - **Hls**
- **p {color: gray; }**
- **p { color: #0000FF; }**
- **p { color: #00F; }**
- **p { color: rgb(0,0,255); }**
- **p { color: rgb(0%, 0%, 100%); }**
- **p {color: Ich(29.2345% 44.2 27); }**
- **p { color: hsl(150deg 30% 60%); }**
- **p { color: lab(29,2345% 39,3 20,06); }**
- **p { color: oklab(29,2345% 39,3 20,06); }**
- **p { color: hwb(194 0% 0%); }**

4.1.- Color del primer plano y el fondo

Para establecer los colores de primer plano y de fondo existen dos propiedades distintas. La propiedad

color es la que debes utilizar para configurar el color del primer plano, es decir, el color del texto

y el color por defecto del borde de un elemento. Mientras que para configurar el color de fondo deberán

emplear la propiedad **background-color**.

Con respecto al color del primer plano deberás tener en cuenta que:

- Si añades color al primer plano de una imagen, ésta seguirá viéndose pero el color se aplicará al borde de la imagen.
- La propiedad **border-color** ignora la propiedad **color**.
- Para configurar el color de todo un documento debemos escribir una regla con esta propiedad color para el selector **body**.



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

- La aplicación de color a los elementos de los formularios no funciona bien en todos los navegadores.

Los ejemplos siguientes muestran diversas formas de configurar el color del texto de los párrafos.

color

CSS1 reconoce hasta 16 nombres de color válidos:

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.

aqua	black	blue	fuchsia	gray	green	lime	maroon	navy	olive	purple	red	silver	teal	white	yellow
aqua	black	blue	fuchsia	gray	green	lime	maroon	navy	olive	purple	red	silver	teal	white	yellow

CSS2 reconoce hasta 65 nombres de color válidos:

orange, aliceblue, antiquewhite, aquamarine, azure, beige, bisque, blanchedalmond, blueviolet, brown, burlywood, cadetblue, chartreuse, chocolate, coral, cornflowerblue, cornsilk, crimson, darkblue, darkcyan, darkgoldenrod, darkgray, darkgreen, darkgrey, darkkhaki, darkmagenta, darkolivegreen, darkorange, darkorchid, darkred, darksalmon, darkseagreen, darkslateblue, darkslategray, darkslategrey, darkturquoise, darkviolet, deeppink, deepskyblue, dimgray, dimgrey, dodgerblue, firebrick, floralwhite, forestgreen, gainsboro, ghostwhite, gold, goldenrod, greenyellow, grey, honeydew, hotpink, indianred, indigo, ivory, khaki, lavender, lavenderblush, lawngreen, lemonchiffon, lightblue, lightcoral, lightcyan,

Orange	Aliceblue	Antiquewhite	Aquamarine	Azure	Beige	Bisque	Blanchedalmond	Blueviolet	Brown
Burlywood	Cadetblue	Chartreuse	Chocolate	Coral	Cornflowerblue	Cornsilk	Crimson	Darkblue	Darkcyan
Darkgoldenrod	Darkgray	Darkgreen	Darkgrey	Darkkhaki	Darkmagenta	Darkolivegreen	Darkorange	Darkorchid	Darkred
Darksalmon	Darkseagreen	Darkslateblue	Darkslategray	Darkslategrey	Darkturquoise	Darkviolet	Deeppink	Deepskyblue	Dimgray
Dimgrey	Dodgerblue	Firebrick	Floralwhite	Forestgreen	Gainsboro	Ghostwhite	Gold	Goldenrod	Greenyellow
Grey	Honeydew	Hotpink	Indianred	Indigo	Ivory	Khaki	Lavender	Lavenderblush	Lawngreen
Lemonchiffon	Lightblue	Lightcoral	Lightcyan						

CSS3 reconoce hasta 65 nombres de color válidos:

lightgoldenrodyellow, lightgray, lightgreen, lightgrey, lightpink, lightsalmon, lightseagreen, lightskyblue, lightslategrey, lightslategray, lightsteelblue, lightyellow, limegreen, linen, mediumaquamarine, mediumblue, mediumorchid, mediumpurple, mediumseagreen, mediumslateblue, mediumspringgreen, mediumturquoise, mediumvioletred, midnightblue, mintcream, mistyrose, moccasin, navajowhite, oldlace, olivedrab, orangered, orchid, palegoldenrod, palegreen, paleturquoise, palevioletred, papayawhip, peachpuff, peru, pink, plum, powderblue, rosybrown, royalblue, saddlebrown, salmon, sandybrown, seagreen, seashell, sienna, skyblue, slateblue, slategrey, slategray, snow, springgreen, steelblue, tan, thistle, tomato, turquoise, violet, wheat, whitesmoke, yellowgreen,

Lightgoldenrodyellow	Lightgray	Lightgreen	Lightgrey	Lightpink	Lightsalmon	Lightseagreen	Lightskyblue	Lightslategrey	Lightslategray
Lightsteelblue	Lightyellow	Limegreen	Linen	Mediumaquamarine	Mediumblue	Mediumorchid	Mediumpurple	Mediumseagreen	Mediumslateblue
Mediumspringgreen	Mediumturquoise	Mediumvioletred	Midnightblue	Mintcream	Mistyrose	Moccasin	Navajowhite	Oldlace	Olivedrab
Orangered	Orchid	Palegoldenrod	Palegreen	Paleturquoise	Palevioletred	Papayawhip	Peachpuff	Peru	Pink
Plum	Powderblue	Rosybrown	Royalblue	Saddlebrown	Salmon	Sandybrown	Seagreen	Seashell	Sienna
Skyblue	Slateblue	Slategray	Slategrey	Snow	Springgreen	Steelblue	Tan	Thistle	Tomato
Turquoise	Violet	Wheat	Whitesmoke	Yellowgreen					



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

Modelos de color

Modelo hsl() Angulo de tono (hue) Saturacion% luminosidad%,

La notación funcional hsl() expresa un color sRGB según sus componentes de tono, saturación y luminosidad. Un componente alfa opcional representa la transparencia del color. Nota: La sintaxis heredada de hsla() es un alias para hsl(), acepta los mismos parámetros y se comporta de la misma manera.

La definición de colores complementarios con hsl() se puede realizar con una única fórmula, ya que se colocan en el mismo diámetro de la rueda de colores. Si θ es el ángulo de tono de un color, su complementario tendrá 180 grados - θ como ángulo de tono.

- `hsl(h s l)` Número de grados en , junto a porcentajes de saturación y luminosidad.
- `hsl(h s l / a)` Idem al anterior, añadiendo un canal alfa de transparencia a, separado por /.
Ver canales alfa

Ahora, expliquemos los parámetros de la función hsl():

- El valor `h` es el matiz de color, con un valor de 0deg a 360deg. En caso de omitir la unidad como , se asumen deg.
- La cifra `s` es el porcentaje de saturación, de 0% a 100%.
- La cifra `l` es el porcentaje de luminosidad, de 0% a 100%.

La sintaxis heredada de hsla() es un alias de hsl().

Sintaxis

`hsl(120 75% 25%)`

`hsl(120 75% 25%/0,6)`

La función también acepta una sintaxis heredada en la que todos los valores están separados por comas.

Valores

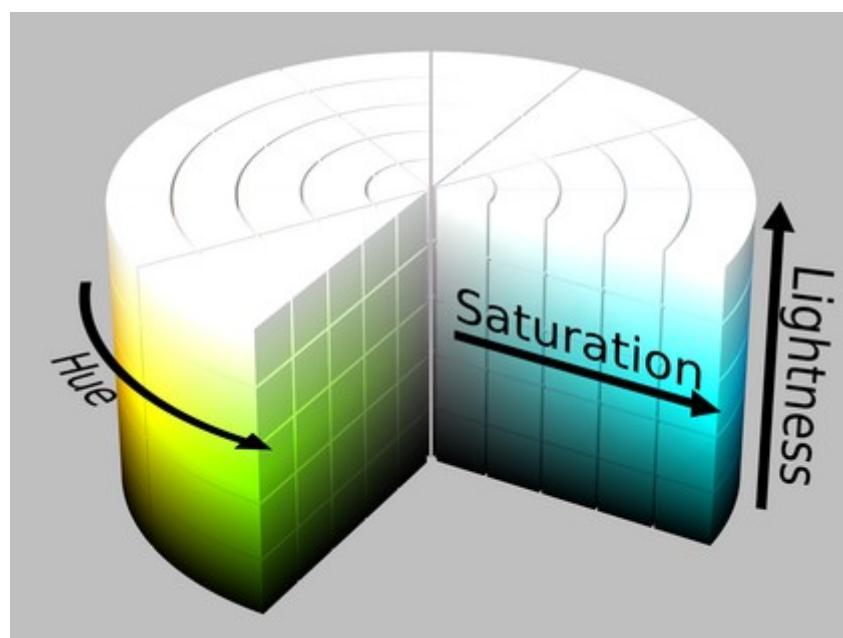
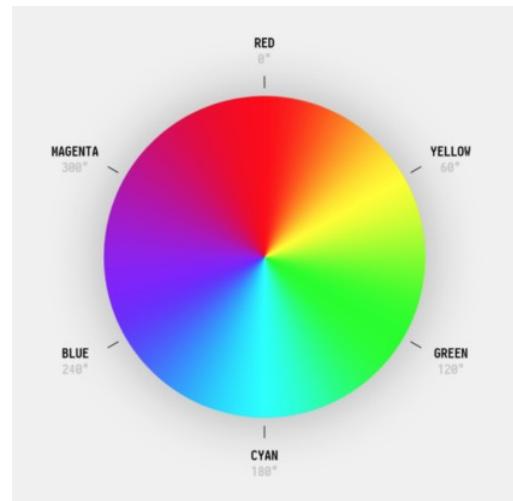
Notación funcional: `hsl(H S L[/ A])`

- `h`: Un <número>, un <ángulo> o la palabra clave `ninguno`, que representa el ángulo de tono. Se pueden encontrar más detalles sobre este tipo en la referencia de <hue>.
- `S`: Un <porcentaje> o la palabra clave `ninguno`, que representa saturación. Aquí el 100% está completamente saturado, mientras que el 0% está completamente insaturado (gris).

- L: Un <porcentaje> o la palabra clave `ninguno`, que representa ligereza. Aquí el 100% es blanco, el 0% es negro y el 50% es "normal".
- Un <valor alfa> o la palabra clave `none`, donde el número 1 corresponde al 100% (opacidad total).

Nota: Esta notación funcional se serializa a valores sRGB y los valores de los componentes rojo, verde y azul se pueden redondear en la serialización.

```
div {
    width: 100px;
    height: 100px;
    background: conic-gradient(
        hsl(360 100% 50%),
        hsl(315 100% 50%),
        hsl(270 100% 50%),
        hsl(225 100% 50%),
        hsl(180 100% 50%),
        hsl(135 100% 50%),
        hsl(90 100% 50%),
        hsl(45 100% 50%),
        hsl(0 100% 50%)
    );
    clip-path: circle(closest-side);
}
```



Modelo hwb() Angulo de tono (hue) Whiteness Blackness

La notación funcional hwb() expresa un color sRGB según su tono, blancura y negrura. Un componente alfa opcional representa la transparencia del color.

Ahora, expliquemos los parámetros de la función hwb():

El valor h es el matiz de color, con un valor de 0deg a 360deg. Se puede omitir la unidad como .

La cifra w es el porcentaje de claridad, de 0% a 100%.

La cifra b es el porcentaje de oscuridad, de 0% a 100%.

Sintaxis

```
hwb(194 0% 0%) /* #00c3ff */
hwb(194 0% 0% / .5) /* #00c3ff con 50% de opacidad */
```

Valores

Notación funcional: hwb(H W B[/ A])

- H Un <número>, un <ángulo> o la palabra clave ninguno, que representa el ángulo de tono. Se pueden encontrar más detalles sobre este tipo en la referencia de <hue>.
- W, B Cada uno como un <porcentaje> o la palabra clave ninguno, que representan blancura y negrura, respectivamente. Especifican la cantidad de blanco y negro que se mezclarán, desde 0% (sin blancura ni negrura) hasta 100% (blancura o negrura total).

Si W + B = 100%, define algún tono de gris. Si W + B > 100%, W y B se normalizan efectivamente como W/(W + B) y B/(W + B), respectivamente.

Un opcional

Un <valor alfa> o la palabra clave none, donde el número 1 corresponde al 100% (opacidad total).

Nota: Esta notación funcional se serializa a valores sRGB y los valores de los componentes rojo, verde y azul se pueden redondear en la serialización.

Nota: Consulte Componentes de color faltantes para ver el efecto de ninguno.





Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar



Pure red

```
.preview {  
background: hwb(0deg 0% 0%);  
}
```



Desaturated and
darkened red

```
.preview {  
background: hwb(0deg 12% 35%);  
}
```



Desaturated red

```
.preview {  
background: hwb(0deg 30% 30%);  
}
```



White

```
.preview {  
background: hwb(0deg 100% 0%);  
}
```



Achromatic gray

```
.preview {  
background: hwb(0deg 66% 66%);  
}
```

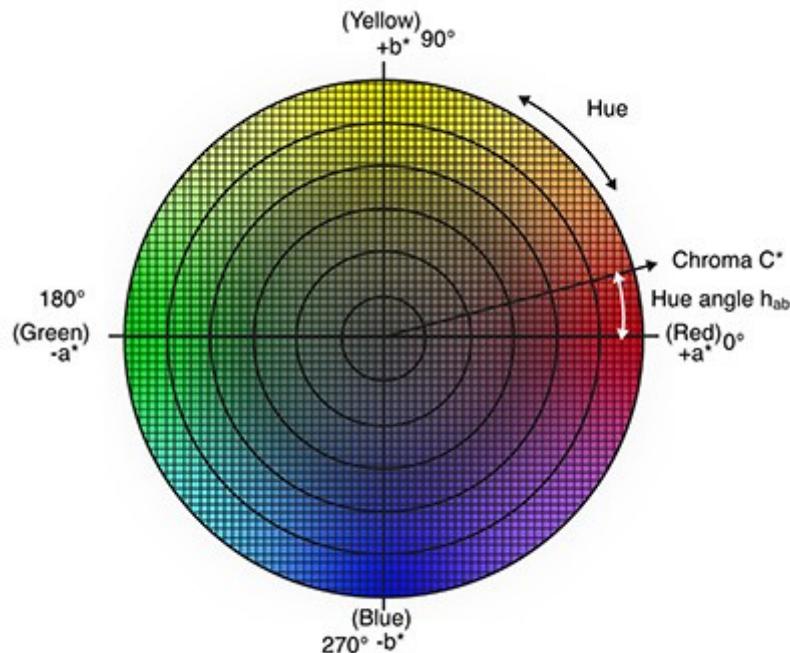


Black

```
.preview {  
background: hwb(0deg 0% 100%);  
}
```

Modelo Ich() luminosidad%, croma% Angulo de tono (hue). -

La notación funcional Ich() expresa un color determinado en el espacio de color LCH. Tiene el mismo eje L que lab(), pero usa las coordenadas polares C (Chroma) y H (Hue).



Sintaxis
CSS

```
Ich(29,2345% 44,2 27);
Ich(52,2345% 72,2 56,2);
Ich(52,2345 % 72,2 56,2 / 0,5);
```

Notación funcional: Ich(L C H[/ A])



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

- **L - luminosidad:** Un <número> entre 0 y 100, un <porcentaje> entre 0% y 100%, o la palabra clave ninguno, que especifica la luminosidad CIE. Aquí el número 0 corresponde al 0% (negro) y el número 100 corresponde al 100% (blanco).
- **C- croma :** Un <número>, un <porcentaje> o la palabra clave ninguno, donde 0% es 0 y 100% es el número 150. Es una medida del croma (que representa aproximadamente la "cantidad de color"). Su valor mínimo útil es 0, mientras que su máximo es teóricamente ilimitado (pero en la práctica no supera 230).
- **h tono-hue :** Un <número>, un <ángulo> o la palabra clave ninguno, que representa el ángulo de tono. Se pueden encontrar más detalles sobre este tipo en la referencia de <hue>.

Un optional

Un <alfa-valor> o la palabra clave none, donde el número 1 corresponde al 100% (opacidad total)

```
[data-color="blue"] { background-color: lch(0% 100 240); }
[data-color="blue-light"] { background-color: lch(100% 100 240); }
[data-color="red"] { background-color: lch(50% 130 20); }
[data-color="red-chroma"] { background-color: lch(100% 30 20); }
[data-color="green"] { background-color: lch(50% 132 130); }
[data-color="green-hue"] { background-color: lch(50% 132 180); }
```

La diferencia entre Ich y hsl aparte del orden es que el croma toma valores absolutos y la saturación es porcentual la l en un de ellos es luminosidad y en el otro es luminancia

**Ich(50% 230 360)
hsl(360 100% 50%)**

Modelo lab() luminosidad%, green/red% blue/yellow%

La notación funcional lab() expresa un color determinado en el espacio de color CIE L*a*b*. Lab representa toda la gama de colores que los humanos pueden ver.

Comparado con el RGB y CMYK, a menudo es más rápido hacer correcciones eficientes de color en Lab. El hecho de que la luminosidad es completamente ignorada en los canales A y B, hace que sea mucho menos sensible a errores.

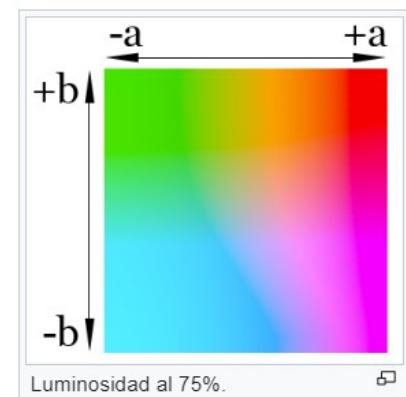
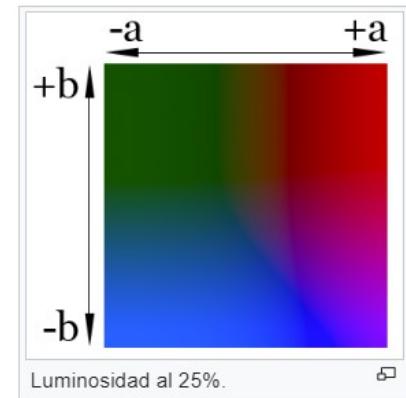
Sintaxis

- lab(29% 39 20);
- lab(52% 40 59);
- lab(52% 40 59/.5);

Valores

Notación funcional: lab(L a b[/ A])

- **L** - Un <número> entre 0 y 100, un <porcentaje> entre 0% y 100%, o la palabra clave `ninguno`, que especifica la luminosidad CIE. Aquí el número 0 corresponde al 0% (negro) y el número 100 corresponde al 100% (blanco).
- **a** - Un <número> entre -125 y 125, un <porcentaje> entre -100% y 100%, o la palabra clave `none`, que especifica la distancia a lo largo del eje a en el espacio de color CIELAB, que es qué tan verde/rojo es el color.
- **b** - Un <número> entre -125 y 125, un <porcentaje> entre -100% y 100%, o la palabra clave `none`, que especifica la distancia a lo largo del eje b en el espacio de color CIELAB, que es qué tan azul/amarillo es el color.
- opcional Un <valor alfa> o la palabra clave `none`, donde el 1 corresponde al 100% (opacidad total).



Nota: Normalmente, cuando los valores porcentuales tienen un equivalente numérico en CSS, 100% es igual al número 1. Este caso es notable cuando 100% es igual al número 100 para el valor L y 125 para los valores a y b.

modelo oklab()

La función oklab() es una mejora de la función lab(). Los parámetros de la función oklab() cambian ligeramente respecto a lab():

Función OKLAB

`oklab(l a b)` Tres parámetros, donde puedes indicar formato con o .



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

oklab(l a b / t) Idem al anterior, añadiendo un canal alfa de transparencia t, separado por /. alfa

Ahora, expliquemos los parámetros de la función oklab():

El valor l es la luminosidad negro-blanco: un valor de 0% a 100% o entre 0 y 1.

El valor a es la luminosidad verde-rojo: un valor de -100% a 100% o entre -0.4 y 0.4.

El valor b es la luminosidad azul-amarillo: un valor de -100% a 100% o entre -0.4 y 0.4.

Mientras que los parámetros A y B de lab() se mueven entre valores -125 y 125, los parámetros A y B de oklab() se mueven entre valores -0.4 y 0.4.

background-color

Con CSS no sólo se puede proporcionar un color de fondo a toda la página, también se puede configurar el color de fondo de cualquier elemento del documento, tanto si son elementos de bloque como de línea.

Con la aparición de CSS, se recomienda emplear "cajas de color" en sustitución de las tablas. Las propiedades relativas al fondo no se heredan pero, como el valor predeterminado de esta propiedad es **transparent**, salvo que se especifique un color concreto, el color de fondo del elemento padre aparecerá a través de sus elementos hijos.

```
p {padding: 5px; background-color: #ccc; }
```

4.2.- Imágenes de fondo.

Las imágenes las podrás configurar de forma muy completa ya que, además de poner una imagen como fondo de una página, podrás ajustar su punto de partida, su patrón de repetición, su posición dentro de un elemento cualquiera y lograr que permanezca fija en esa posición aunque se mueva el resto del documento.

Para realizar esta configuración detallada se emplean las propiedades:

background-image, **background-repeat**, **background-position**, **backgroundattachment** y **background**.

background-image

Esta propiedad sirve para configurar la imagen de fondo de cualquier elemento. La propiedad **background-image** prevalece sobre la propiedad **background-color** por lo que si con un elemento realizas declaraciones con estas dos propiedades ignorará la declaración de **backgroundcolor**.

`background-image: url(url_imagen);`

En el siguiente ejemplo se muestra la configuración de la imagen de fondo, usando un archivo de imagen de nombre “fondo.gif”, en todo el documento excepto para el párrafo que tiene un identificador “cabecera” que tendrá como imagen de fondo un archivo de imagen de nombre “fondocuerpo.gif”.



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

- body {background-image: url(fondo.gif);}
- p#cabecera {background-image: url(fondo-cuerpo.gif);}

background-repeat

Permite configurar a tu gusto la forma en la que se repetirá la imagen cuando su tamaño sea más pequeño que la ventana del navegador. También permite evitar que la imagen se repita.

background-repeat: repeat-x;repeat-y;no-repeat;

```
div#cuerpo {background-image: url(fondo.gif); background-repeat: no-repeat; }  
div.horiz {background-image: url(imagen.gif); background-repeat: repeat-x; }  
div.vert {background-image: url(imagen.gif); background-repeat: repeat-y; }
```

Con el primer ejemplo, consigues que la imagen aparezca sólo una vez, con el segundo ejemplo la imagen se repetirá a lo ancho (horizontalmente) tantas veces como necesite y, con el tercer ejemplo la imagen de fondo se repetirá a lo largo del documento (verticalmente).

background-position

Especifica la posición de la primera imagen que cubrirá el fondo del elemento en el que esté definida esta propiedad. Al posicionamiento podemos asignarles los valores: left, right, top, bottom y center

background-position: horizontal vertical

= 50% 50%;
= 150px 150px;
= (top, bottom) (left,center , right)

los cuales se usan, normalmente, de dos en dos sin importar el orden (uno indica su posición horizontal y el otro indica su posición vertical). Si sólo se indica una, se supone que la otra es center.

ejemplo 1 En el siguiente ejemplo, se configura el fondo del elemento `body` con una imagen llamada "fondo.gif" que aparecerá sólo una vez en la parte superior central del cuerpo del documento.

```
body {background-image: url(fondo.gif);  
background-position: top center;  
background-repeat: no-repeat; }
```

ejemplo 2 Para el posicionamiento se pueden emplear también las medidas de longitud vistas anteriormente. En este caso, las medidas son relativas al extremo superior izquierdo del elemento. En el siguiente **ejemplo**, se configura el mismo fondo de antes que aparecerá a 150 píxeles de la esquina superior izquierda del cuerpo del elemento (horizontal y verticalmente).



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

```
body {background-image: url(fondo.gif);  
background-position: 150px 150px;  
background-repeat: no-repeat; }
```

ejemplo 3 También puedes utilizar los valores porcentuales. Como seguro supones, si indicas sólo un valor se asume que el otro es un 50%. Debes tener en cuenta que el valor porcentual se aplica “al contenedor

y a la imagen en sí”. En el siguiente ejemplo se muestra la regla de estilo que logra que la imagen

quede colocada en el centro del elemento **body**.

```
body {background-image: url(imagen.gif);  
background-position: 50% 50%;  
background-repeat: no-repeat; }
```

background-attachment

Con esta propiedad puedes fijar la imagen en una posición concreta. Se le pueden asignar los valores:

scroll, **fixed** e **inherit**, siendo el valor por defecto. Se suele emplear el valor **fixed** para conseguir que la imagen no se desplace con el documento.

- **background-attachment : fixed** para conseguir que la imagen no se desplace con el documento.
- **background-attachment : scroll** para conseguir que la imagen no se desplace con el documento.

background.

Esta propiedad permite configurar todas las propiedades de fondo vistas anteriormente usando una única declaración, de forma similar a lo que ocurría con la propiedad **font**, pero a diferencia de ésta, no tiene ninguna propiedad obligatoria y sus valores pueden aparecer en cualquier orden.

Sólo debes tener una restricción: la posición se indica con dos valores que deben aparecer juntos, primero el horizontal seguido, inmediatamente después, por el vertical ya que si sólo aplicamos un valor el otro se configura por defecto a **center**.

```
body {background: url(fondo.gif) fixed top center no-repeat; }  
div.cabecera {background: repeat-x url(fondo.gif) red; }  
p {background: #ccc; }  
#contenido span.imagen {background: url("imagenes/imagen.gif") 0 50% no-repeat; }
```



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

4.3.- Opacidades.

Si te fijas en las ventanas, persianas y cortinas de una casa, verás que algunas ventanas como las de

los cuartos de baño son de cristal pero no son transparentes, dejar ver el exterior pero no de forma

nítida. Cuando se bajan las persianas, no se ve absolutamente nada del exterior, y las cortinas, dependiendo

de su tejido, pueden dejar pasar la claridad en mayor o menor medida.

La opacidad es una característica de los elementos que nos permite mostrar o no otros elementos

que tengan por debajo. Para conseguir efectos de transparencia en algunos elementos tienes las

siguientes propiedades: **opacity**, **moz-opacity** y **filter**.

opacity

Esta propiedad, que es compatible con todos los navegadores que soporten CSS3, permite asignar

valores comprendidos entre **0 (invisible o totalmente transparente)** y **1 (totalmente opaco)**.

-moz-opacity

Esta propiedad, permite asignar los mismos valores que la propiedad anterior. La diferencia está en

que sólo es compatible con versiones anteriores del Firefox 0.9.

filter

Esta propiedad, de IE (5.5 y siguientes), tiene varios efectos: degradaciones, desenfocados, sombras,

etcétera. Para lograr la transparencia hay que aplicar el filtro `alpha`, con valores entre 0 y 100.

En el ejemplo siguiente tienes el código donde se configura la opacidad de algunos elementos para

que funcione en todos los navegadores.



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

Responde

```
<html>
<head> <style type="text/css">
    body {background: #00f; color: black}
    #saludo { background: white; width: 200px; height: 200px; filter: alpha(opacity=50); -moz-opacity: 0.5; opacity: 0.5; }
    #frase { background: white; width: 200px; height: 200px; filter: alpha(opacity=100); -moz-opacity: 1; opacity: 1; }
    #despedida { background: white; width: 200px; height: 200px; filter: alpha(opacity=0); -moz-opacity: 0; opacity: 0; }
</style> </head>
<body>
    <div id="saludo"><h1>¡Hola!</h1></div>
    <div id="frase">Este es un ejemplo de opacidad</div>
    <div id="despedida"><h2>¡Adiós!</h2></div>
</body>
</html>
```

A la vista del ejemplo anterior ¿Cuáles de las siguientes afirmaciones son correctas?

El color del fondo de toda la página es blanco.

El color de fondo del saludo es blanco.

El color de fondo de la frase es blanco

El color de fondo de la despedida es blanco.

El saludo no se ve.

La frase no se ve.

La despedida no se ve.

El fondo del saludo es el mismo que el fondo de la página.

El fondo del texto es el mismo que el de la página.

El fondo de la despedida es el mismo que el de la página.

- *Si has fallado varias veces esta pregunta después de copiar el ejemplo y verlo funcionar en tu navegador, probablemente sea porque has dado como correcta la última respuesta.*
- *La pregunta tenía truco.*
- *Está claro que el fondo de la frase es de color blanco, porque está configurado con ese color y de forma totalmente opaca (opacidad=1 ó 100 en el caso de filter).*
- *También está claro que la despedida no se ve y es porque el elemento h2 hereda las propiedades de su elemento padre div, el cual está configurado como totalmente transparente.*
- *Esto último provoca que lo único que veamos, en lugar de la despedida, sea el fondo de la página, pero eso no significa que tenga el mismo color de fondo, su color de fondo es el blanco como en los otros dos casos.*



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

Practica

Para cada ejercicio debes aplicarlo en un elemento que se vea con formato

<h2> Ejercicio i</h2>

<p> ejercicio ... </p>

Ejercicio 1. Crear un javascript para generar todos los colores de esa misma con un bucle y estos arrays

```
var coloresCss1 = new array("aqua", "black", "blue", "fuchsia", "gray", "green", "lime", "maroon", "navy", "olive", "purple", "red", "silver", "teal", "white", "yellow");
```

```
var coloresCss2 = new array("orange", "aliceblue", "antiquewhite", "aquamarine", "azure", "beige", "bisque", "blanchedalmond", "blueviolet", "brown", "burlywood", "cadetblue", "chartreuse", "chocolate", "coral", "cornflowerblue", "cornsilk", "crimson", "darkblue", "darkcyan", "darkgoldenrod", "darkgray", "darkgreen", "darkgrey", "darkkhaki", "darkmagenta", "darkolivegreen", "darkorange", "darkorchid", "darkred", "darksalmon", "darkseagreen", "darkslateblue", "darkslategray", "darkslategrey", "darkturquoise", "darkviolet", "deeppink", "deepskyblue", "dimgray", "dimgrey", "dodgerblue", "firebrick", "floralwhite", "forestgreen", "gainsboro", "ghostwhite", "gold", "goldenrod", "greenyellow", "grey", "honeydew", "hotpink", "indianred", "indigo", "ivory", "khaki", "lavender", "lavenderblush", "lawngreen", "lemonchiffon", "lightblue", "lightcoral", "lightcyan", "lightgoldenrodyellow");
```

```
var coloresCss3 = new array("lightgray", "lightgreen", "lightgrey", "lightpink", "lightsalmon", "lightseagreen", "lightskyblue", "lightslategrey", "lightslategrey", "lightsteelblue", "lightyellow", "limegreen", "linen", "mediumaquamarine", "mediumblue", "mediumorchid", "mediumpurple", "mediumseagreen", "mediumslateblue", "mediumspringgreen", "mediumturquoise", "mediumvioletred", "midnightblue", "mintcream", "mistyrose", "moccasin", "navajowhite", "oldlace", "olivedrab", "orangered", "orchid", "palegoldenrod", "palegreen", "paleturquoise", "palevioletred", "papayawhip", "peachpuff", "peru", "pink", "plum", "powderblue", "rosybrown", "royalblue", "saddlebrown", "salmon", "sandybrown", "seagreen", "seashell", "sienna", "skyblue", "slateblue", "slategray", "slategrey", "snow", "springgreen", "steelblue", "tan", "thistle", "tomato", "turquoise", "violet", "wheat", "whitesmoke", "yellowgreen");
```

Ejercicio 2. Crea reglas CSS para mostrar divs con las diferentes maneras de expresar los colores con background-color (gray , #0000FF, #00F , rgb , rgba , lch, hsl, lab, oklab, hwb)

Ejercicio 3. Crea documento con una imagen de fondo con un gif animado

Ejercicio 4. Crea documento con imagen de fondo con un png con transparencias y un background-color que se repita hacia la izquierda

Ejercicio 5. Crea documento con una imagen de fondo con un jpg fijo que no se repita

Ejercicio 6. Crea documento con una imagen de fondo que no se repita con las configuraciones background-position

- a) left bottom
- b) left top
- c) right center
- d) right bottom
- e) bottom center
- f) 25% 10%
- g) 100px 500px

Ejercicio 7. Prueba el siguiente Código de ejemplo

```
<html> <head> <style type="text/css">  
    body {background: #00f; color: black}  
    #saludo { background: white; width: 200px; height: 200px; filter: alpha(opacity=50); -moz-opacity: 0.5; opacity: 0.5; }  
    #frase { background: white; width: 200px; height: 200px; filter: alpha(opacity=100); -moz-opacity: 1; opacity: 1; }  
    #despedida { background: white; width: 200px; height: 200px; filter: alpha(opacity=0); -moz-opacity: 0; opacity: 0; }  
</style> </head> <body>  
    <div id="saludo"><h1>¡Hola!</h1></div>  
    <div id="frase">Este es un ejemplo de opacidad</div>
```



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

```
<div id="despedida"><h2>¡Adiós!</h2></div>  
</body> </html>
```

Ejercicio 8. Inserta una imagen con estas opacidades

- a) filter: alpha(opacity=50); -moz-opacity: 0.5; opacity: 0.5;
- b) filter: alpha(opacity=25); -moz-opacity: 0.25; opacity: 0.25;
- c) filter: alpha(opacity=75); -moz-opacity: 0.75; opacity: 0.75;

Ejercicio 9. Inserta una imagen que se haga transparente al 50% cuando se pase el raton con :hover.