

TEMA 4. UNIDADES DE MEDIDA Y COLORES

1. UNIDADES DE MEDIDA	2
1.1. UNIDADES ABSOLUTAS	2
1.2. UNIDADES RELATIVAS	2
1.3. PORCENTAJES	4
1.4. RECOMENDACIONES	4
2. COLORES	5
2.1. PALABRAS CLAVE	5
2.2. RGB DECIMAL	5
2.3. RGB PORCENTUAL	6
2.4. RGB HEXADECIMAL	6
2.5. RGBA	7
2.6. COLORES DEL SISTEMA	8
2.7. HSL	8
2.8. HSLA	9
2.9. COLORES WEB	9

1. UNIDADES DE MEDIDA

Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas **se indican** como un valor numérico entero o decimal seguido de una unidad de medida (**sin ningún espacio en blanco entre el número y la unidad de medida**).

CSS divide las unidades de medida en dos grupos: **absolutas y relativas**. Las medidas relativas definen su valor en relación con otra medida, por lo que, para obtener su valor real, se debe realizar alguna operación con el valor indicado. Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

Si el valor es 0, la unidad de medida es opcional. Si el valor es distinto a 0 y no se indica ninguna unidad, la medida se ignora completamente, lo que suele ser uno de los errores más habituales de los diseñadores que empiezan con CSS. Algunas propiedades permiten indicar medidas negativas, aunque habitualmente sus valores son positivos. Si el valor decimal de una medida es inferior a 1, se puede omitir el 0 de la izquierda (0.5em es equivalente a .5em).

1.1. UNIDADES ABSOLUTAS

Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia. A continuación, se muestra la lista completa de unidades absolutas definidas por CSS y su significado:

- ✓ px, un punto en pantalla.
- ✓ .in, pulgadas ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.
- ✓ cm, centímetros.
- ✓ mm, milímetros.
- ✓ pt, puntos. Un punto equivale a 1/72 de pulgadas.
- ✓ pc, picas. Una pica equivale a 12 puntos.

A continuación, se muestran ejemplos de utilización de unidades absolutas:

```
/* El cuerpo de la página debe mostrar un margen de media pulgada */
body { margin: 0.5in; }

/* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */
h1 { line-height: 2cm; }

/* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre si */
p { word-spacing: 4mm; }

/* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */
a { font-size: 12pt }

/* Los elementos <span> deben tener un tamaño de letra de 1 pica */
span { font-size: 1pc }
```

La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. Su principal desventaja es que son muy poco flexibles y no se adaptan fácilmente a los diferentes medios.

1.1. UNIDADES RELATIVAS

Las unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son

las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.

A continuación, se muestran las tres unidades de medida relativas definidas por CSS y la referencia que toma cada una para determinar su valor real:

- ✓ **em**, (no confundir con la etiqueta `` de HTML) relativa respecto del tamaño de letra del elemento donde se utilizan. (se suele pensar que son relativas al elemento padre).
- ✓ **rem**, relativa respecto del tamaño de letra definida para el navegador. El tamaño del pixel al que se traducen depende del tamaño de la fuente del elemento raíz de la página, o sea el elemento html.
- ✓ **ex**, relativa respecto de la altura de la letra **x** ("*equis minúscula*") del tipo y tamaño de letra del elemento.
- ✓ **vw** y **vh**, ancho y alto de pantalla que estamos utilizando, viewport width y viewport height.

Las unidades em, rem y ex no han sido creadas por CSS, sino que llevan décadas utilizándose en el campo de la tipografía. Aunque no es una definición exacta, la unidad 1em equivale a la anchura de la letra M ("*eme mayúscula*") del tipo y tamaño de letra del elemento.

La unidad em hace referencia al tamaño en puntos de la letra que se está utilizando. Si se utiliza una tipografía de 12 puntos, 1em equivale a 12 puntos. El valor de 1ex se puede aproximar por 0.5 em.

Si se considera el siguiente ejemplo:

```
p { margin: 1em; }
```

La regla CSS anterior indica que los párrafos deben mostrar un margen de anchura igual a 1em. Como se trata de una unidad de medida relativa, es necesario realizar un cálculo matemático para determinar la anchura real de ese margen.

Las unidades cuando se anidan se heredan, es decir, cuando tenemos un p con una medida en ems y dentro tenemos un elemento hijo que tiene otra medida en ems, el elemento hijo toma tamaño en función del tamaño que se ha dado al elemento padre, en este caso el p.

A continuación, se modifica el ejemplo anterior para cambiar el tamaño de letra de los párrafos:

```
p { font-size: 32px; margin: 1em; }
```

El valor del margen sigue siendo el mismo en unidades relativas (1em) pero su valor real ha variado porque el tamaño de letra de los párrafos ha variado. En este caso, el margen tendrá una anchura de 32px, ya que 1em siempre equivale al tamaño de letra del elemento.

Si se quiere reducir la anchura del margen a 16px pero manteniendo el tamaño de letra de los párrafos en 32px, se debe utilizar la siguiente regla CSS:

```
p { font-size: 32px; margin: 0.5em; }
```

El valor 0.5em se interpreta como "*la mitad del tamaño de letra del elemento*", ya que se debe multiplicar por 0.5 su tamaño de letra (32px x 0.5 = 16px). De la misma forma, si se quiere mostrar un margen de 8px de anchura, se debería utilizar el valor 0.25em, ya que 32px x 0.25 = 8px.

La gran ventaja de las unidades relativas es que siempre mantienen las proporciones del diseño de la página. Establecer el margen de un elemento con el valor 1em equivale a indicar que "el

margen del elemento debe ser del mismo tamaño que su letra y debe cambiar proporcionalmente".

En efecto, si el tamaño de letra de un elemento aumenta hasta un valor enorme, su margen de 1em también será enorme. Si su tamaño de letra se reduce hasta un valor diminuto, el margen de 1em también será diminuto. El uso de unidades relativas permite mantener las proporciones del diseño cuando se modifica el tamaño de letra de la página.

El funcionamiento de la unidad ex es idéntico a em, salvo que, en este caso, la referencia es la altura de la letra x minúscula, por lo que su valor es aproximadamente la mitad que el de la unidad em.

Las unidades en rem son iguales que las unidades en em, pero con la diferencia de que no dependen del tamaño de su elemento padre. Las unidades en rem son dependientes del tamaño de la fuente definida para el navegador y no le afectan los tamaños de los elementos padres, es decir, no hereda.

Por último, las medidas indicadas en píxel también se consideran relativas, ya que el aspecto de los elementos dependerá de la resolución del dispositivo en el que se visualiza la página HTML. Si un elemento tiene una anchura de 400px, ocupará la mitad de una pantalla con una resolución de 800x600, pero ocupará menos de la tercera parte en una pantalla con resolución de 1440x900.

Las unidades de medida se pueden mezclar en los diferentes elementos de una misma página, como en el siguiente ejemplo:

```
body { font-size: 10px; }  
h1 { font-size: 2.5em; }
```

En primer lugar, se establece un tamaño de letra base de 10 píxel para toda la página. A continuación, se asigna un tamaño de 2.5em al elemento <h1>, por lo que su tamaño de letra real será de 2.5 x 10px = 25px.

1.1. PORCENTAJES

El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, rem, ex y px. Un porcentaje está formado por un valor numérico seguido del símbolo %. Cada una de las propiedades de CSS que permiten indicar como valor un porcentaje, define el valor al que hace referencia ese porcentaje.

Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de letra de los elementos:

```
body { font-size: 1em; }  
h1 { font-size: 200%; }  
h2 { font-size: 150%; }
```

Los tamaños establecidos para los elementos <h1> y <h2> mediante las reglas anteriores, son equivalentes a 2em y 1.5em respectivamente, por lo que es más habitual definirlos mediante em.

Los porcentajes también se utilizan para establecer la anchura de los elementos:

```
div#contenido { width: 600px; }
div.principal { width: 80%; }
```

```
<div id="contenido">
  <div class="principal">
    ...
  </div>
</div>
```

En el ejemplo anterior, la referencia del valor 80% es la anchura de su elemento padre. Por tanto, el elemento <div> cuyo atributo class vale principal tiene una anchura de 80% x 600px = 480px.

1.1. RECOMENDACIONES

En general, se recomienda el uso de unidades relativas siempre que sea posible, ya que mejora la accesibilidad de la página y permite que los documentos se adapten fácilmente a cualquier medio y dispositivo.

La W3C, recomienda el uso de la unidad em y rem para indicar el tamaño del texto y para todas las medidas que sean posibles.

Normalmente se utilizan píxel y porcentajes para definir el layout del documento (básicamente, la anchura de las columnas y de los elementos de las páginas) y em, rem y porcentajes para el tamaño de letra de los textos.

2. COLORES

Los colores en CSS se pueden indicar de diferentes forma: palabras clave, colores del sistema, RGB, RGBA, HSL, HSLA...

1.1. PALABRAS CLAVE

CSS palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

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

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

Figura 4.1 Colores definidos mediante las palabras clave de CSS

Aunque es una forma muy sencilla de referirse a los colores básicos, este método prácticamente no se utiliza en las hojas de estilos de los sitios web reales, ya que se trata de una gama de colores muy limitada.

Además de la lista básica, los navegadores modernos soportan muchos otros nombres de colores. La lista completa se puede ver en [nombre de colores](#).

[en.wikipedia.org/wiki/Websafe](http://en.wikipedia.org/wiki/Web_safe).

1.1. RGB DECIMAL

En el campo del diseño gráfico, se han definido varios modelos para hacer referencia a los colores. Los dos modelos más conocidos son RGB y CMYK. Simplificando su explicación, el modelo RGB consiste en definir un color indicando la cantidad de color rojo, verde y azul que se debe *mezclar* para obtener ese color. Técnicamente, el modelo RGB es un modelo de tipo "aditivo", ya que los colores se obtienen sumando sus componentes.

Por lo tanto, en el modelo RGB un color se define indicando sus tres componentes R (rojo), G (verde) y B (azul). Cada una de las componentes puede tomar un valor entre cero y un valor máximo. De esta forma, el color rojo puro en RGB se crea mediante el máximo valor de la componente R y un valor de 0 para las componentes G y B.

Si todas las componentes valen 0, el color creado es el negro y si todas las componentes toman su valor máximo, el color obtenido es el blanco. En CSS, las componentes de los colores definidos mediante RGB decimal pueden tomar valores entre 0 y 255. El siguiente ejemplo establece el color del texto de un párrafo:

```
p { color: rgb(71, 98, 176); }
```

La sintaxis que se utiliza para indicar los colores es `rgb()` y entre paréntesis se indican las tres componentes RGB, en ese mismo orden y separadas por comas. El color del ejemplo anterior se obtendría mezclando las componentes R=71, G=98, B=176, que se corresponde con un color azul claro.

Si se indica un valor menor que 0 para una componente, automáticamente se transforma su valor en 0. Igualmente, si se indica un valor mayor que 255, se transforma automáticamente su valor a 255.

1.1. RGB PORCENTUAL

Las componentes RGB de un color también se pueden indicar mediante un porcentaje. El funcionamiento y la sintaxis de este método es el mismo que el del RGB decimal. La única diferencia es que en este caso el valor de las componentes RGB puede tomar valores entre 0% y 100%. Por tanto, para transformar un valor RGB decimal en un valor RGB porcentual, es preciso realizar una regla de tres, considerando que 0 es igual a 0% y 255 es igual a 100%.

El mismo color del ejemplo anterior se puede representar de forma porcentual:

```
p { color: rgb(27%, 38%, 69%); }
```

Al igual que sucede con el RGB decimal, si se indica un valor inferior a 0%, se transforma automáticamente en 0% y si se indica un valor superior a 100%, se trunca su valor a 100%.

1.1. RGB HEXADECIMAL

Aunque es el método más complicado para indicar los colores, se trata del método más utilizado con mucha diferencia. De hecho, prácticamente todos los sitios web reales utilizan exclusivamente este método.

Para entender el modelo RGB hexadecimal, en primer lugar, es preciso introducir un concepto matemático llamado *sistema numérico hexadecimal*. Cuando realizamos operaciones

matemáticas, siempre utilizamos 10 símbolos para representar los números (del 0 al 9). Por este motivo, se dice que utilizamos un sistema numérico decimal.

No obstante, el sistema decimal es solamente uno de los muchos sistemas numéricos que se han definido. Entre los sistemas numéricos alternativos más utilizados se encuentra el sistema hexadecimal, que utiliza 16 símbolos para representar sus números.

Como sólo conocemos 10 símbolos numéricos, el sistema hexadecimal utiliza también seis letras (de la A a la F) para representar los números. De esta forma, en el sistema hexadecimal, después del 9 no va el 10, sino la A. La letra B equivale al número 11, la C al 12, la D al 13, la E al 14 y la F al número 15.

Definir un color en CSS con el método RGB hexadecimal requiere realizar los siguientes pasos: - Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176 - Transformar el valor decimal de cada componente al sistema numérico hexadecimal. Se trata de una operación exclusivamente matemática, por lo que puedes utilizar una calculadora. En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0 - Para obtener el color completo en formato RGB hexadecimal, se concatenan los valores hexadecimales de las componentes RGB en ese orden y se les añade el prefijo #. De esta forma, el color del ejemplo anterior #4762B0 en formato RGB hexadecimal.

Siguiendo el mismo ejemplo de las secciones anteriores, el color del párrafo se indica de la siguiente forma utilizando el formato RGB hexadecimal:

```
p { color: #4762B0; }
```

Recuerda que, aunque es el método más complicado para definir un color, se trata del método que utilizan la inmensa mayoría de sitios web, por lo que es imprescindible dominarlo. Afortunadamente, todos los programas de diseño gráfico convierten de forma automática los valores RGB decimales a sus valores RGB hexadecimales, por lo que no tienes que hacer ninguna operación matemática:

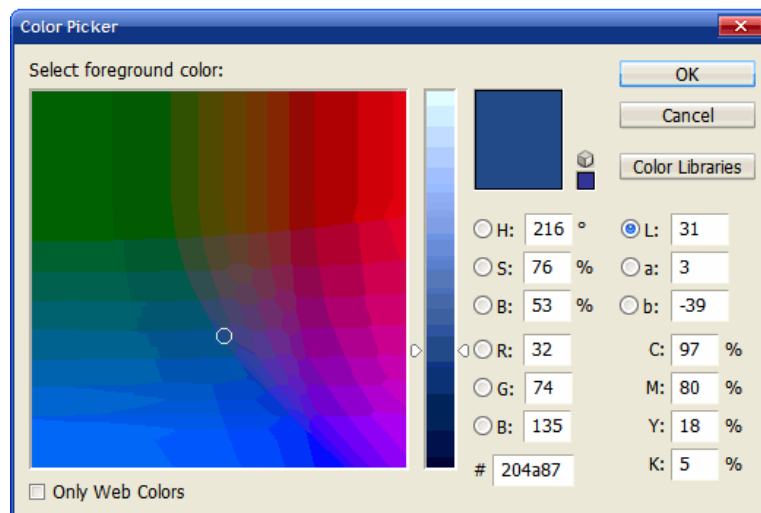


Figura 4.2 Herramienta de color de Photoshop para definir los colores según los modelos RGB, CMYK, Lab, HSB y RGB hexadecimal

El formato RGB hexadecimal es la forma más compacta de indicar un color, ya que incluso es posible comprimir sus valores cuando todas sus componentes son iguales dos a dos:

#AAA = #AAAAAA

#FFF = #FFFFFF

#A0F = #AA00FF

#369 = #336699

En el siguiente ejemplo se establece el color de fondo de la página a blanco, el color del texto a negro y el color de la letra de los titulares se define de color rojo:

```
body { background-color: #FFF; color: #000; }
```

```
h1, h2, h3, h4, h5, h6 { color: #C00; }
```

Las letras que forman parte del color en formato RGB hexadecimal se pueden escribir en mayúsculas o minúsculas indistintamente. No obstante, se recomienda escribirlas siempre en mayúsculas o siempre en minúsculas para que la hoja de estilos resultante sea más limpia y homogénea.

1.1. RGBA

La representación de colores con RGBA es idéntica a RGB, con la diferencia de que podemos añadir un cuarto parámetros A que es la transparencia, el canal Alpha. Este parámetro toma valores entre 0 y 1 donde 0 es transparente y 1 es opaco. Esta forma de aplicar transparencia es muy útil cuando no queremos que un elemento hijo herede la transparencia de un elemento padre.

```
div { color: rgba(71, 98, 176, 0.5); }
```

En este caso le estamos dando una transparencia del 50% a la capa, que no será heredada por los elementos hijos de la capa.

1.1. COLORES DEL SISTEMA

Los colores del sistema son similares a los colores indicados mediante su nombre, pero en este caso hacen referencia al color que muestran algunos elementos del sistema operativo del usuario.

Existen varios colores definidos, como por ejemplo `ActiveBorder`, que hace referencia al color del borde de las ventanas activas. Consulta la lista completa de colores del sistema.

Aunque es posible definir los colores en CSS utilizando estos nombres, se trata de un método que nunca se utiliza, por lo que se puede considerar prácticamente como una rareza de CSS.

1.1. HSL

Los **colores HSL** en nuestras hojas de estilo se representan utilizando el matiz, la saturación y la luminosidad de un color.

```
div{background-color:hsl(0,100%,50%)/* hsl( matiz, saturacion, luminosidad ) */;}
```

MATIZ

Pensemos un momento en la rueda de los colores. Cada grado de esta rueda es un color. De aquí que el matiz es un número de 0 a 359. A lo largo de los 360° el color cambia de rojo (0°) a verde (120°) a azul (240°) y de nuevo a rojo (360°).

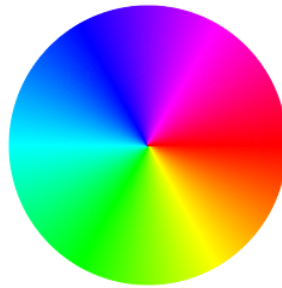


Figura 4.3 Matiz.

Conociendo el valor matiz de un color podemos calcular fácilmente, casi de manera intuitiva, el **color complementario** de este: o sea el color diametralmente opuesto.

El código hsl para rojo es `hsl(0,100%, 50%)`, donde $H = 0$;

El parámetro H del color complementario (diametralmente opuesto) será de 180, o sea `hsl(180,100%,50%)`, un color a medio camino entre verde y azul.

Por colores triádicos entendemos un grupo de tres colores situados a 120° uno del otro en el círculo cromático. De nuevo: calcular los colores triádicos es una tarea casi imposible si conocemos solo el código hex o rgb de nuestro color, pero vuelve a ser una tarea muy fácil si conocemos el código hsl de nuestro color.

En el caso del rojo `hsl(0,100%,50%)` los otros dos colores triádicos son: verde `hsl(120,100%,50%)` y azul `hsl(240,100%,50%)`

SATURACIÓN

La saturación de un color se representa en porcentaje. Por lo cual la **saturación** puede tomar valores de 100% a 0% (grises).



Figura 4.4 Saturación.

LUMINOSIDAD

Representamos la luminosidad de un color en porcentajes. Por lo cual la **luminosidad** puede tomar valores entre 100% (blanco) y 0% (falta de color o sea negro).



Figura 4.3 Luminosidad.

1.1. HSLA

Además de los códigos HSL podemos utilizar los códigos HSLA donde **a**, exactamente como en el caso de los colores RGBA, representa el valor alpha.

1.1. COLORES WEB

Como cada componente RGB de los colores puede tomar un valor entre 0 y 255, el número total de colores que se pueden representar con este formato es de $256 \times 256 \times 256 =$

16.777.216 colores. Sin embargo, en la década de los 90 los monitores de los usuarios no eran capaces de mostrar más de 256 colores diferentes.

A partir de todos los colores disponibles, se eligieron 216 colores que formaron la paleta de colores "web safe". Esta paleta de colores podía ser utilizada por los diseñadores con la seguridad de que se verían correctamente en cualquier navegador de cualquier sistema operativo de cualquier usuario.

Hoy en día, su importancia ha descendido notablemente, ya que prácticamente todos los usuarios utilizan dispositivos con una profundidad de color de 16 y 32 bits. No obstante, el auge en el uso de los dispositivos móviles hace que siga siendo un tema a considerar.

Consulta la [lista completa de colores web safe y sus valores hexadecimales](#). Podrás encontrar diferentes paletas de colores, con diferentes representaciones del color.