

# ÍNDICE

CSS: Unidades de medida y colores

Unidades de medida

4

Colores

14

# CSS: UNIDADES DE MEDIDA Y COLORES

CSS es tan flexible que permite indicar las medidas y colores de muchas formas diferentes. Por este motivo, se presentan a continuación todas las alternativas disponibles en CSS para indicar las medidas y los colores.

 Este manual no es una referencia de propiedades CSS. Para una consulta exhaustiva y actualizada de propiedades y valores CSS por favor consultar:

En inglés:

<https://www.w3schools.com/cssref/default.asp>

En castellano:

[https://developer.mozilla.org/es/docs/Web/CSS/Referencia\\_CSS](https://developer.mozilla.org/es/docs/Web/CSS/Referencia_CSS)

# 01

## UNIDADES DE MEDIDA

Establecen la altura, anchura y márgenes de los elementos y el tamaño de letra del texto

### CSS: Unidades de medida y colores

Las medidas en CSS 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).

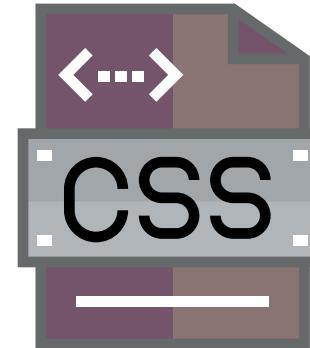
CSS divide todas las unidades de medida en dos grupos: **absolutas** y **relativas**. Las **medidas relativas** definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado. Las **unidades absolutas** establecen de forma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

El tipo de dato CSS **<length>** denota medidas de distancia. Es un valor **<number>** seguido por una unidad de longitud (**px**, **em**, **pc**, **in**, **mm**, ...). Al igual que en cualquier dimensión CSS, no debe haber espacio entre la unidad y el número. La unidad de longitud es opcional después del valor **<number> 0**.

**Muchas propiedades CSS (CSS properties) reciben valores <length>, como por ejemplo width, margin-top, y font-size.**

Para algunas propiedades, el uso de longitudes negativas es un error de sintaxis, mientras que para algunas propiedades está permitido. Nótese que aunque los valores **<percentage>** también son dimensiones CSS y son aceptadas por algunas propiedades CSS que aceptan valores **<length>**, no son valores **<length>** en sí.

**Los valores de tipo <length> pueden ser interpolados para permitir animaciones.** En este caso son interpolados como números reales, de punto flotante. La interpolación sucede en el valor calculado. La velocidad de la interpolación es definida por la función **<timing-function>** asociada a la animación.



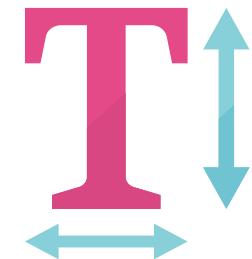
## UNIDADES DE LONGITUD RELATIVA

### Longitudes relativas a la fuente

Estas medidas son más flexibles que las absolutas porque se adaptan más fácilmente a los diferentes medios. A continuación se muestra la lista de unidades de longitud relativas y la referencia que se toma para determinar su valor real:

#### ■ em

Esta unidad **representa el tamaño calculado de fuente (font-size) del elemento**. Si se usa dentro de la propiedad **font-size**, representa el tamaño de fuente heredado por el elemento. Se usa, por lo general, para crear interfaces escalables que mantengan el ritmo vertical de la página, aun cuando el usuario cambie el tamaño de las fuentes. Las propiedades CSS **line-height**, **font-size**, **margin-bottom** y **margin-top** generalmente tienen valores expresados en **em**.



#### ■ ex

Esta unidad **representa la altura de la x de la fuente (font) del elemento**. En fuentes que incluyen la letra 'x', es generalmente la altura de letras minúsculas en la fuente; **1ex ≈ 0.5em** en muchas fuentes.

**■ ch**

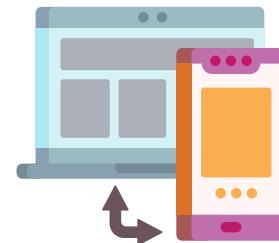
Esta unidad **representa la anchura, o más precisamente, la medida de avance, del glifo '0'** (cero, de carácter Unicode U+0030) en la fuente (**font**) del elemento.

**■ rem**

Esta unidad **representa el tamaño (font-size) del elemento raíz** (p.ej. el tamaño de fuente del elemento `<html>`). Cuando se aplica a **font-size** del elemento raíz, representa su valor inicial. Se trata de una unidad muy práctica para crear interfaces perfectamente escalables. Si no es soportada por los navegadores, se puede recurrir a unidades **em**, aunque éstas son algo más complejas.

## Longitudes de porcentaje del viewport

Las longitudes de porcentaje del viewport definen una longitud relativa al tamaño del viewport, que es la porción visible del documento. Solamente los navegadores basados en Gecko actualizan los valores del viewport dinámicamente, cuando el tamaño de éste es modificado (al cambiar el tamaño de la ventana en un ordenador, o al girar el dispositivo móvil o tablet).



En conjunto con `overflow:auto`, el espacio tomado por barras de desplazamiento no es restado al tamaño del viewport, mientras en el caso de `overflow:scroll`, sí lo es.

En un bloque de declaración de la **regla-at @page**, el uso de longitudes de viewport es inválido, y la declaración será desechada.

**■ vh**

1/100 de la altura del viewport.

**■ vw**

1/100 de la anchura del viewport.

**■ vmin**

1/100 del valor mínimo entre la altura y anchura del viewport.

**■ vmax**

1/100 del valor máximo entre la altura y anchura del viewport.

## UNIDADES DE LONGITUD ABSOLUTA

Las unidades de longitud absoluta representan una medida física, y cuando las propiedades físicas del medio de salida son conocidas, como en diseño para impresión. Esto se hace anclando una de las unidades a una unidad física, y definiendo el resto con relación a ésta.

La definición del ancla difiere entre dispositivos de baja resolución, como pantallas, y dispositivos de alta resolución, como impresoras.

**Para dispositivos de ppp bajo,** la unidad **px** representa el píxel de referencia física, y el resto son definidos con relación a éste. Así, **1in** es definido como 96px, que equivalen a 72pt. La consecuencia de esta definición es que en dichos dispositivos, las longitudes descritas en pulgadas (**in**), centímetros (**cm**), milímetros (**mm**) no necesariamente coincidirán con la longitud de la unidad física del mismo nombre.

**Para dispositivos de alto ppp,** las pulgadas (**in**), centímetros (**cm**), milímetros (**mm**) son definidos como su contraparte física. De esta forma, la unidad **px** es definida con relación a ellas (1/96 de 1 pulgada).

**Los usuarios pueden incrementar el tamaño de fuente por razones de accesibilidad.** Para permitir interfaces usables sin importar el tamaño de fuente, se usa únicamente unidades de longitud absolutas cuando las características físicas del medio de salida son conocidas, como imágenes de mapa de bits. Al establecer longitudes relacionadas al tamaño de fuente, es preferible usar unidades relativas, como **em** o **rem**.



#### ■ px

Medida relativa al dispositivo de visualización. Para pantallas, generalmente es el tamaño de un píxel (punto) de la pantalla del dispositivo. Para impresoras y pantallas de muy alta resolución, un píxel CSS implica múltiples píxeles del dispositivo, por lo que el número de píxeles por pulgada se mantenga al rededor de 96.

#### ■ mm

Un milímetro.

#### ■ q

Un cuarto de milímetro (1/40° de centímetro).

#### ■ cm

Un centímetro (10 milímetros).

#### ■ in

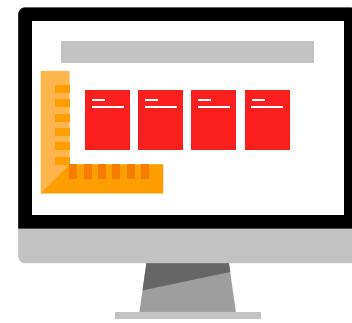
Una pulgada (2.54 centímetros).

#### ■ pt

Un punto (1/72° de pulgada).

#### ■ pc

Una pica (12 puntos).



**■ mozmm** 

Una unidad experimental que intenta generar exactamente un milímetro, sin importar el tamaño de resolución de la pantalla. Esto raramente será lo que se desea, pero podría ser útil para dispositivos móviles, en particular.

## UNIDADES CSS Y PUNTOS POR PULGADA

La unidad **in** no representa una pulgada física en pantalla, sino **96px**. Esto significa que sin importar la densidad de píxeles real en pantalla, se asume que serán **96ppp**. En dispositivos con mayor densidad de píxeles, **1in** será menor que una pulgada física. De forma similar, **mm**, **cm**, y **pt** no son longitudes absolutas.

Algunos ejemplos específicos:

- 1in siempre son 96px,**
- 3pt siempre son 4px,**
- 25.4mm siempre son 96px.**

## COMPATIBILIDAD DE NAVEGADORES

La tabla de compatibilidad dependiendo del navegador utilizado por cada usuario puedes consultarla en el siguiente enlace:

[https://developer.mozilla.org/es/docs/Web/CSS/length#Compatibilidad\\_de\\_navegadores#Compatibilidad\\_de\\_navegadores](https://developer.mozilla.org/es/docs/Web/CSS/length#Compatibilidad_de_navegadores#Compatibilidad_de_navegadores)

## ESPECIFICACIONES

Las especificaciones de las unidades de medición en CSS puedes consultarla en este enlace:

[https://developer.mozilla.org/es/docs/Web/CSS/length#Compatibilidad\\_de\\_navegadores#Especificaciones](https://developer.mozilla.org/es/docs/Web/CSS/length#Compatibilidad_de_navegadores#Especificaciones)

# 02

## COLORES

Las formas de aplicación de color con CSS en los elementos de nuestra página web

### CSS: Unidades de medida y colores

Los colores en CSS se pueden indicar de formas diferentes:

- Usando una **palabra clave**.
- Usando el **sistema RGB cubic-coordinate** (usando #hexadecimal o las notaciones funcionales rgb y rgba)
- Usando el **sistema HSL cylindrical-coordinate** (usando las notaciones funcionales hsl y hsla)

Aunque **el método más habitual es el del RGB hexadecimal**, a continuación se muestran todas las alternativas que ofrece CSS.

El tipo de datos CSS `<color>` denota un color en el sRGB color space. Hay que destacar que la lista de valores de color aceptados ha ido creciendo conforme la especificación evolucionaba, para acabar con la lista de colores de CSS3.

Asociado con el color en el espacio sRGB, un valor `<color>` también consiste en una coordenada alpha-channel o valor de transparencia, indicando cómo se debe mezclar el color con su color de fondo.

Aunque los valores de colores en CSS son definidos de manera precisa, existe la posibilidad de que parezcan distintos en dispositivos diferentes. La mayoría de ellos no están calibrados y algunos navegadores no soportan los color profile de algunos dispositivos de salidas. En esta situación el color puede variar bastante.

Los **valores del tipo de datos CSS `<color>` pueden ser interpolados para lograr animaciones o para crear valores `<gradient>`**. En este caso son interpolados en cada uno de sus componentes rojo, verde y azul, conteniendo, cada uno de ellos, un número decimal con coma flotante. Debe tenerse en cuenta que la interpolación del color sucede en el alpha-premultiplied sRGBA color space para evitar que aparezcan grises inesperados. En las animaciones, la velocidad de la interpolación viene determinada por la **timing function** asociada con la animación.

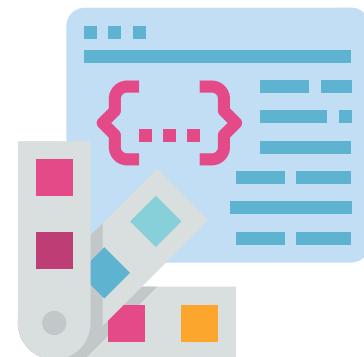
## PALABRAS CLAVE DE COLORES

Las palabras clave de colores **son identificadores en los que no hay distinción entre mayúsculas y minúsculas y que sirven para representar un color concreto**, por ejemplo `red`, `blue`, `brown`, `lightseagreen`. El nombre describe el color aunque es mayormente artificial. La lista de valores válidos cambia mucho de una especificación a otra:

■ **CSS Nivel 1** sólo acepta 16 colores básicos, los llamados colores VGA ya que son tomados de los colores que son capaces de mostrar las tarjetas gráficas VGA.

■ **CSS Nivel 2** añadió la palabra clave `orange`.

■ Desde el principio los navegadores aceptaron otros colores, sobre todo la lista de **colores X11** con unas pocas diferencias, ya que, los primeros navegadores eran en su mayoría aplicaciones X11. **SVG 1.0** fué el primer estándar que definió formalmente estas palabras clave. Con frecuencia nos referimos a ello como la lista extendida de colores, los colores X11 o los colores SVG.



## Advertencias a tener en cuenta

Hay algunas advertencias al usar las palabras clave relativas a colores:

- A excepción de los **16 colores básicos, que son comunes en HTML, los otros no pueden ser usados en HTML**. HTML convertirá estos valores desconocidos usando un algoritmo específico y se obtendrá colores totalmente diferentes. Estas palabras clave deberían sólo usarse en SGV y CSS.

- **Colores desconocidos hacen que la propiedad CSS sea declarada inválida.** Las propiedades inválidas son ignoradas y, por tanto, el color no tendrá efecto. Esto es un comportamiento diferente al del HTML

- **Ninguna palabra clave de color tiene transparencia en CSS,** son colores planos y sólidos.

- **Algunas mencionan el mismo color:**

- darkgray / darkgrey
- darkslategray / darkslategrey
- dimgray / dimgrey
- lightgray / lightgrey
- lightslategray / lightslategrey
- gray / grey
- slategray / slategrey

## CSS: Unidades de medida y colores

- Aunque las palabras clave de colores han sido tomadas de los nombres de colores usuales en X11, **el color puede ser ligeramente diferente al color en X11 ya que son adaptados por el fabricante para un hardware específico.**

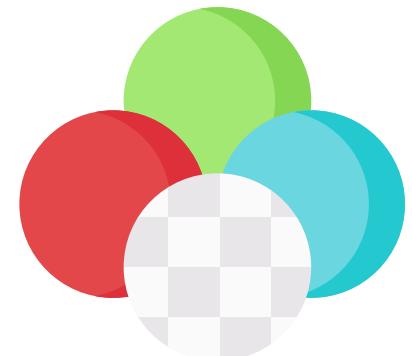
- \*La tabla de palabras clave de los diferentes colores puedes consultarla en el siguiente enlace:

[https://developer.mozilla.org/es/docs/Web/CSS/color\\_value](https://developer.mozilla.org/es/docs/Web/CSS/color_value)

### PALABRA CLAVE 'TRANSPARENT'

La palabra **transparent** representa un color totalmente transparente, es decir, el color que veremos será el color de fondo. Técnicamente es un color negro con un valor mínimo en el canal alfa y la manera de representarlo es **rgba(0,0,0,0)**.

**Transparent** fue redefinido como color verdadero hasta el CSS Nivel 3, permitiendo su uso en cualquier sitio donde se requiera un valor **<color>**.



## PALABRA CLAVE 'CURRENTCOLOR'

La palabra clave **currentColor** representa el valor calculado para la la propiedad del elemento color. **Permite hacer que las propiedades de color sean heredadas por los elementos hijo**, que no lo hacen por defecto. También puede ser usada en propiedades que heredan el valor calculado de la propiedad del elemento color, en este caso, será equivalente a la plabara clave **inherit** en estos elementos, si es que la hubiera.

## MODELO RGB()

Los colores puede ser definidos de dos maneras:

### ■ Notación Hexadecimal #RRGGBB y #RGB

"#", seguido de seis caracteres hexadecimales (0-9, A-F).

"#", seguido de tres caracteres hexadecimales (0-9, A-F).

La notación de tres dígitos (#RGB) y la de seis (#RRGGBB) son iguales. Por ejemplo, #f03 y #ff0033 representan el mismo color.

### ■ Notación funcional **rgb(R,G,B)**

"**rgb**", seguido de tres valores **<integer>** o tres **<percentage>** values.

El número entero 255 representa el 100%, y F o FF en notación hexadecimal.

## MODELO HSL()

Los colores tambien puede ser definidos usando el modelo tono (**hue**) - saturación (**saturation**) - brillo (**lightness**) (**HSL**) con la notación funcional **hsl()**.

**La ventaja del modelo HSL sobre el modelo RGB es que es mucho más intuitivo:** puedes adivinar los colores que tú quieres y luego modificarlos. Es más fácil crear conjuntos de colores manteniendo el tono (hue) igual y modificando el brillo y la saturación.

El tono (**Hue**) se representa como un ángulo del círculo de color (el arco iris representado en un círculo). El ángulo se proporciona como un **<number>** sin unidades. Por definición **rojo=0=360**, y los demás colores están dispersos por el círculo, **verde=120**, **azul=240**, etc.

La saturación y el brillo son representados por porcentajes: 100% es saturacion completa, y 0% es una sombra de gris. 100% brillo es blanco, 0% negro, y 50% es brillo "normal"

## MODELO RGBA()

Podemos definir los colores con el modelo Rojo (red) - Verde (green) - Azul (blue) - alpha (RGa) usando la notación funcional **rgba()**. **RGa amplia el modelo RGA añadiendo un canal alfa**, permitiendo de esta manera especificar la opacidad de un color:

**a** **representa opacidad: 0=transparente; 1=opaco.**

## MODELO HSLA()

Podemos definir los colores usando el modelo saturación tono (hue) - (saturation) - brillo (lightness) (HSLa) usando la notación funcional hsla(). **HSLa amplía el modelo HSL incluyendo el canal alfa, permitiendo de esta manera especificar la opacidad de un color:**

a representa opacidad: 0=transparente; 1=opaco.

## COLORES DE 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. Se trata de un método que nunca se utiliza.

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.

\*Puedes consultar las especificaciones de cada sistema operativo en el siguiente enlace:

[https://developer.mozilla.org/es/docs/Web/CSS/color\\_value#Colores\\_de\\_Sistema](https://developer.mozilla.org/es/docs/Web/CSS/color_value#Colores_de_Sistema)

The logo consists of the word "nowe" in a black sans-serif font. The letter "o" features a blue wavy line through it.

Espacio libre para tus ideas:

---