

## Capítulo 3. Unidades de medida y colores

Muchas de las propiedades de CSS que se verán en los próximos capítulos permiten indicar medidas y colores en sus valores. Por este motivo, se presentan a continuación todas las alternativas disponibles en CSS para indicar las medidas y los colores.

### 3.1. 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.

#### 3.1.1. Unidades relativas

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

- **em**, (no confundir con la etiqueta `<em>` de HTML) ***relativa respecto del tamaño de letra empleado***. Aunque no es una definición exacta, el valor de **1em** se puede aproximar por la anchura de la letra **M** ("eme mayúscula") del tipo y tamaño de letra que se esté utilizando.
- **ex**, ***relativa respecto de la altura de la letra x*** ("equis minúscula") del tipo y tamaño de letra que se esté utilizando (el valor de **1ex** se puede aproximar por **0.5 em**).
- **px**, (píxel) ***relativa respecto de la resolución de la pantalla del usuario***.

En el siguiente ejemplo, se indica que el tamaño de letra del texto de la página debe ser el **90%** del tamaño por defecto (que depende de cada navegador):

```
body { font-size: 0.9em; }
```

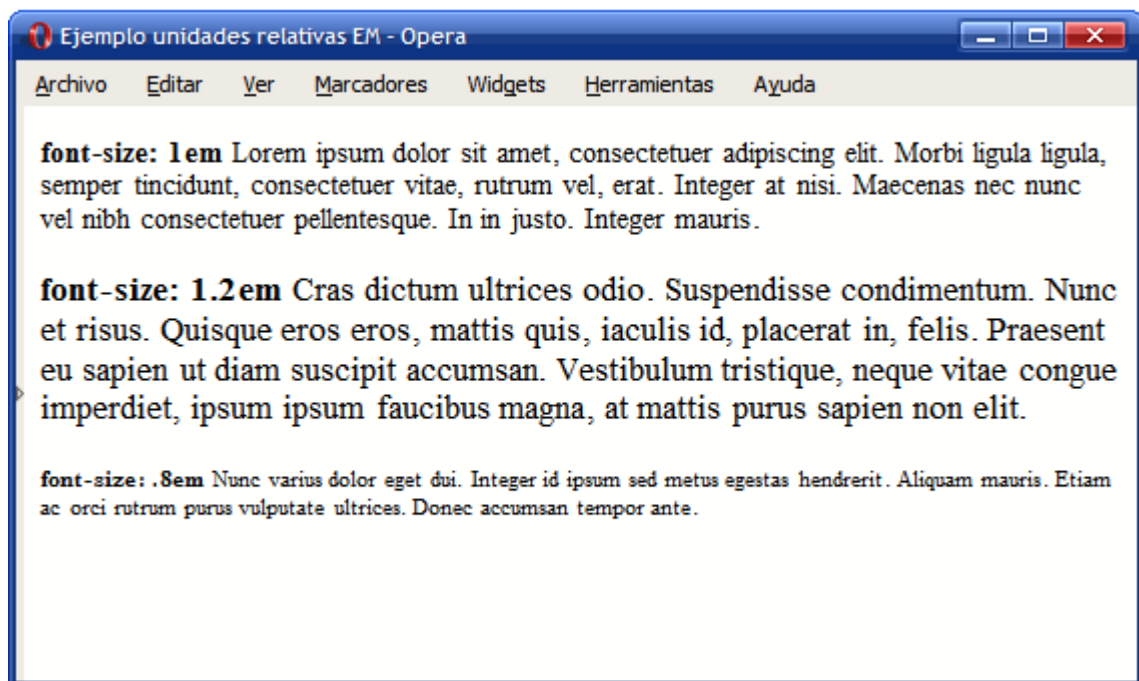
Como **em** es una unidad relativa, el valor **0.9** indicado sólo tiene sentido cuando se tiene en consideración su referencia. Para la unidad **em**, la referencia es el tamaño de letra por defecto del sistema (ordenador, dispositivo móvil, etc.) del usuario.

Por lo tanto, **0.9em** significa que se debe multiplicar **0.9** por el tamaño de letra por defecto, lo que en la práctica significa que la medida indicada es igual al **90%** del tamaño de letra por defecto. Si este tamaño por defecto es **12pt**, el valor **0.9em** sería igual a  $0.9 \times 12\text{pt} = 10.8\text{pt}$ .

Cuando el valor decimal de una medida es inferior a **1**, se puede omitir el **0** de la izquierda, por lo que el código anterior es equivalente al código siguiente:

```
body { font-size: .9em; }
```

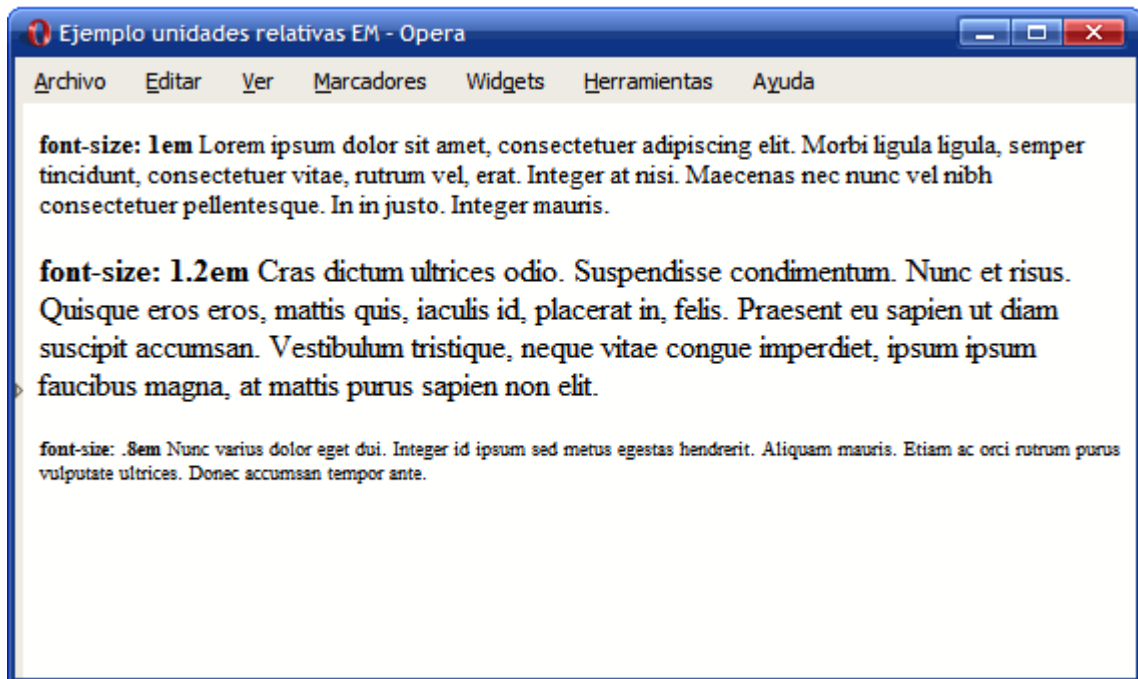
El siguiente ejemplo muestra el uso de la unidad **em** para establecer el tamaño de la letra de diferentes párrafos:



**Figura 3.1.** Ejemplo de tamaño de letra definido con la unidad relativa em

La **propiedad font-size** permite establecer el tamaño de letra del texto de un elemento. En este caso, la referencia para el valor de **font-size** de un elemento siempre es el tamaño de letra de su elemento padre. Si el elemento no se encuentra dentro de ningún otro elemento, la referencia es el tamaño de letra del *elemento <body>*. Si no se indica de forma explícita un valor para el tamaño de letra del elemento *<body>*, la referencia es el tamaño de letra por defecto del navegador.

Siguiendo esta norma, si en el ejemplo anterior **se modifica el tamaño de letra del elemento <body>** (que es el elemento padre de los tres párrafos) y se le asigna un valor de **0.8em**, el aspecto que muestran los párrafos en el navegador es el siguiente:



**Figura 3.2.** Ejemplo de tamaño de letra definido con la unidad relativa em

Al haber reducido el tamaño de letra que era la referencia del tamaño de letra de los párrafos, su texto se ve con una **letra más pequeña, aunque manteniendo las proporciones**.

**Las distintas unidades se pueden mezclar** entre 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**.

### 3.1.2. Unidades absolutas

Las unidades absolutas definen las **medidas de forma completa**, ya que ***sus valores reales no se calculan a partir de otro valor de referencia***, sino que son directamente los valores indicados. A continuación se muestra la lista completa de unidades absolutas definidas por CSS y su significado:

- **in**, del inglés "inches", **pulgadas** (1 pulgada son 2.54 centímetros).
- **cm**, **centímetros**.
- **mm**, **milímetros**.
- **pt**, **puntos** (1 punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros).
- **pc**, **picas** (1 pica equivale a 12 puntos, es decir, unos 4.23 milímetros).

A continuación se muestran ejemplos de utilización de unidades absolutas:

```
body { margin: 0.5in; }  
h1 { line-height: 2cm; }  
p { word-spacing: 4mm; }  
a { font-size: 12pt; }  
span { font-size: 1pc; }
```

Su uso es idéntico al de las unidades relativas, siendo su única diferencia que los valores indicados son directamente los valores que se utilizan, sin necesidad de calcular los valores reales en función de otras referencias.

***De todas las unidades absolutas, la única que se utiliza con cierta frecuencia es la de los puntos (pt).*** El motivo es que se trata de la unidad preferida para indicar el tamaño de letra del texto para los documentos que se van a imprimir, es decir, para el medio **print** de CSS.

### 3.1.3. Porcentajes

CSS define otra **unidad de medida relativa** basada en los **porcentajes**. Un porcentaje está formado por un valor numérico seguido del **símbolo %** y siempre está referenciado a otra medida. 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**.

### 3.1.4. 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.
- El organismo W3C recomienda el uso de la unidad **em** 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 elementos de las páginas) y **em** y porcentajes para el tamaño de letra de los textos.

## 3.2. Colores

Los colores en CSS se pueden indicar de *cinco formas diferentes*: **palabras clave**, **colores del sistema**, **RGB hexadecimal**, **RGB numérico** y **RGB porcentual**. Aunque el método más habitual es el del RGB hexadecimal, a continuación se muestran todas las alternativas que ofrece CSS.

### 3.2.1. Palabras clave

CSS define **17 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

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

**Figura 3.4.** 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 [http://en.wikipedia.org/wiki/Web\\_safe](http://en.wikipedia.org/wiki/Web_safe).

### 3.2.2. RGB decimal

En el campo del diseño gráfico, se han definido varios modelos para hacer referencia a los colores. Uno de los más conocidos es 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.

Cada una de las **componentes** (rojo, verde, azul) puede tomar un **valor entre cero y 255**. 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.

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.

### 3.2.3. RGB porcentual

Otra forma de indicar las **componentes RGB** de un color es **mediante un porcentaje**. El funcionamiento y la sintaxis de este método es el mismo que el del RGB decimal. La única diferencia en este caso es que el valor de las **componentes RGB** puede tomar **valores** entre **0%** y **100%**. El mismo color del ejemplo anterior se puede representar de forma porcentual:

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

### 3.2.4. 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.

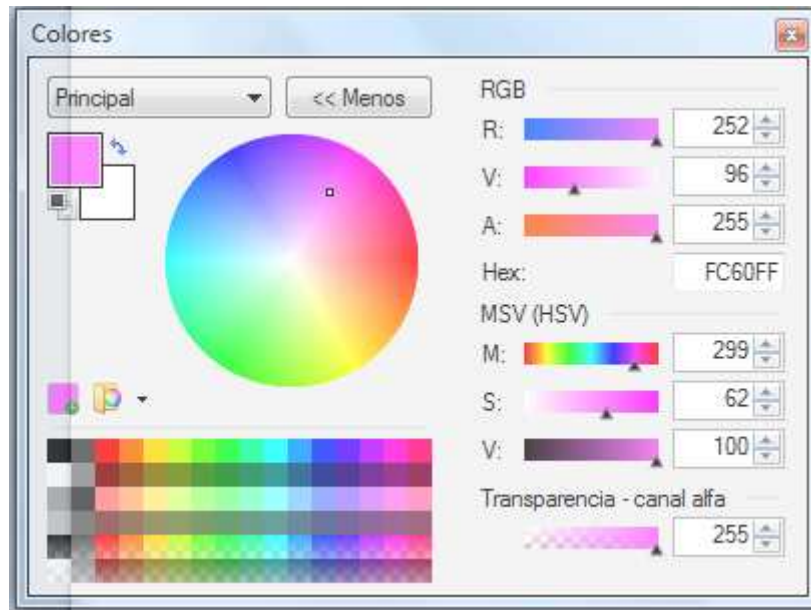
Para definir un color en CSS con el método RGB hexadecimal requiere realizar los **siguientes pasos**:

1. Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176
2. *Transformar el valor decimal de cada componente al sistema numérico hexadecimal.* En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0
3. 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 es #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; }
```

Afortunadamente, la mayoría de 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 3.5.** Herramienta de color de paint.NET

Una de las ventajas del formato **RGB hexadecimal** es que *se pueden 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 a rojo:

```
body { background-color: #FFF; color: #000; }
h1, h2, h3, h4, h5, h6 { color: #C00; }
```

### 3.2.5. 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. La lista completa de colores definidos se puede ver en <http://www.w3.org/TR/CSS21/ui.html#system-colors>.



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.

### 3.2.6. Colores web safe

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.

La lista completa de colores web safe y sus valores hexadecimales se pueden consultar en [http://en.wikipedia.org/wiki/Web\\_colors#Web-safe\\_colors](http://en.wikipedia.org/wiki/Web_colors#Web-safe_colors).