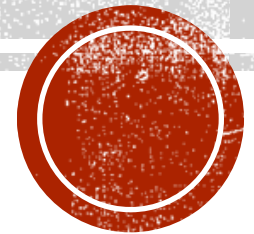


# UNIDADES DE MEDIDA



# 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).



# 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:

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



- 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 }`



# UNIDADES ABSOLUTAS

- 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.



# UNIDADES RELATIVAS

- La 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.





# UNIDADES RELATIVAS

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 <em> de HTML) relativa respecto del tamaño de letra del elemento.
- ex, relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.
- px, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.



# UNIDADES RELATIVAS

- Las unidades em y ex no han sido creadas por CSS, sino que llevan décadas utilizándose en el campo de la tipografía.
- 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.



- La unidad de medida em siempre hace referencia al tamaño de letra del elemento. Por otra parte, todos los navegadores muestran por defecto el texto de los párrafos con un tamaño de letra de 16 píxel. Por tanto, en este caso el margen de 1em equivale a un margen de anchura 16px.



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 ( $32\text{px} \times 0.5 = 16\text{px}$ ). De la misma forma, si se quiere mostrar un margen de 8px de anchura, se debería utilizar el valor 0.25em, ya que  $32\text{px} \times 0.25 = 8\text{px}$ .



- 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".



- 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.
- 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  $2.5 \times 10\text{px} = 25\text{px}$ .



# PORCENTAJES

- El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, ex y px. 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.



- 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  $80\% \times 600\text{px} = 480\text{px}$ .



# 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.



# 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		



# 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



# RGB DECIMAL

- 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); }
```





# RGB PORCENTUAL

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



# RGB HEXADECIMAL

Entre los sistemas numéricos alternativos más utilizados se encuentra el sistema hexadecimal, que utiliza 16 símbolos para representar sus números.

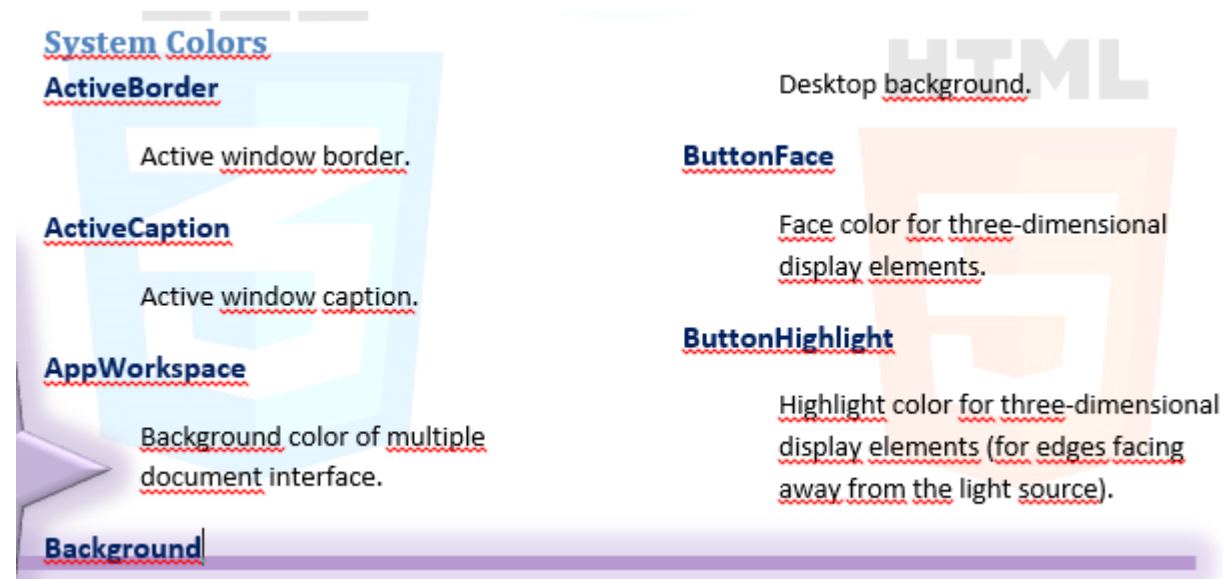
Como sólo conocemos 10 símbolos numéricos, el sistema hexadecimal utiliza también seis letras (de la A a la F) para representar los números. De esta forma, en el sistema hexadecimal, después del 9 no va el 10, sino la A. La letra B equivale al número 11, la C al 12, la D al 13, la E al 14 y la F al número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

p { color: #4762B0; }



# 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.



### ButtonShadow

Shadow color for three-dimensional display elements.

### ButtonText

Text on push buttons.

### CaptionText

Text in caption, size box, and scrollbar arrow box.

### GrayText

Grayed (disabled) text. This color is set to #000 if the current display

### InfoText

Text color for tooltip controls.

### Menu

Menu background.

### MenuText

Text in menus.

### Scrollbar

Scroll bar gray area.

### ThreeDDarkShadow

### GrayText

Grayed (disabled) text. This color is set to #000 if the current display driver does not support a solid gray color.

### ThreeDDarkShadow

Dark shadow for three-dimensional display elements.

### ThreeDFace

Face color for three-dimensional

### Highlight

## WindowFrame

### High

Window frame.

## WindowText

Text in

```
p { color: WindowText; background-color: Window }
```

### InfoBackground

display elements.

### Window

# 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. No obstante, el auge en el uso de los dispositivos móviles hace que siga siendo un tema a considerar, ya que las pantallas de muchos móviles sólo pueden representar un número reducido de colores.



# BACKGROUND Y COLORES DE FONDO

- En código HTML, el fondo se lo aplicamos dentro del <BODY> mediante los atributos bgcolor y background para asignar al fondo una imagen gif o jpg.
- Las especificaciones de las CSS introducen importantes novedades respecto a los colores a aplicar en las páginas web.



# ATRIBUTO COLOR

- El atributo color define el color del texto del documento (no se debe confundir con el color de background) tanto mediante los nombres de los colores en inglés: black, silver, gray, white, red, blue, etc) y con los códigos hexadecimales.



# ATRIBUTO BACKGROUND-COLOR , BACKGROUND-IMAGEN

Este atributo de background-color determina el color de fondo, el atributo background-image, tiene una función similar al background="imagen.gif" de HTML clasico, invocando una imagen en formato GIF o JPG cargada en la css.

El atributo background-repeat, permite repetir la imagen de fondo tal y como sucede con la marca BODY de HTML.

Puede asumir distintos valores:

- **repeat**, la imagen se replica en vertical y horizontal.
- **repeat-x**, replica la imagen sólo en horizontal.
- **repeat-y**, replica la imagen sólo en vertical.





# BORDES

Borde solid:

Este borde es de una línea sólida.

```
.bordesolido{  
  border-color: #aaaaaa;  
  border-width: 1px;  
  border-style: solid;  
}
```



# BORDE DOTTED (PUNTEADO):

Este borde está compuesto a base de una línea de puntos.

```
.bordepunteado{  
  border: 1px dotted #ff8833;  
}
```



# BORDE DOUBLE (DOBLE):

El borde double está compuesto por una doble línea sólida. Hay que saber que para que este borde se vea como una doble línea, tenemos que especificar un ancho de al menos 3 pixel. Así se vería un borde con dos líneas de un píxel más un espacio de separación del borde de otro pixel. A medida que especificamos una anchura mayor del borde doble, las líneas van aumentando de grosor, así como el espacio que las separa.

```
.bordedoble{  
  border: 9px double #990000;  
}
```



# BORDE DASHED (LÍNEA DISCONTINUADA):

El borde dashed es muy similar al borde punteado, pero en lugar de puntos, lo que tenemos es una línea partida.

```
.bordedashed{  
  border: 1px dashed #660000;  
}
```



# BORDE GROOVE (RANURA):

Es un borde especial, con una especie de relieve difícil de describir, creado con la combinación de colores claros y oscuros de borde.

```
.bordegroove{  
  border: 5px groove #66cc66;  
}
```



# BORDE RIDGE (CRESTA):

Este borde también es un poco raro. Parecido al groove, pero con los colores oscuros y claros invertidos.

```
div.borderidge{  
  border: 10px ridge #6666cc;  
}
```



# BORDE INSET (RELIEVE HACIA DENTRO):

Crea un borde con una imitación de biselado, como si estuviera hundido. El efecto de este borde se ve mejor si el elemento que está bordeando tiene un color de fondo parecido al color de borde.

```
div.bordeinset{  
  border: 10px inset #3333ff;  
  background-color: #0000ff;  
}
```



# BORDE OUTSET (CON RELIEVE HACIA FUERA):

Este último estilo de borde imita un biselado, como si fuera un botón que está levantado. Es igual que el estilo de borde inset, pero con los colores claros y oscuros invertidos. Veremos también este efecto mejor si el elemento tiene un color de fondo.

```
div.bordeoutset{  
  border: 10px outset #cccccc;  
  background-color: #cccccc;  
}
```





# BORDES REDONDEADOS CON CSS3.

- Tenemos la propiedad **border-radius**, que permite definir bordes redondeados en las esquinas, especificando las medidas del radio que deben darse a la curva de las esquinas. Su uso sería aproximado al que vemos a continuación:

`border-radius: 5px`



- Ejemplo

- `<span style="border-style:dotted; border-color:red; border-width:1px;">dotted</span>`
- `<span style="border-style:dashed; border-color:red; border-width:1px;">dashed</span>`
- `<span style="border-style:solid; border-color:red; border-width:1px;">solid</span>`
- `<span style="border-style:double; border-color:blue; border-width:thick;">double</span>`
- `<span style="border-style:groove; border-color:silver; border-width:thick;">groove</span>`
- `<span style="border-style:ridge; border-color:yellow; border-width:thick;">ridge</span>`
- `<span style="border-style:inset; border-color:aqua; border-width:thick;">inset</span>`
- `<span style="border-style:outset; border-color:lime; border-width:thick;">outset</span>`

