

# Lenguajes de marcas y sistemas de gestión de la información



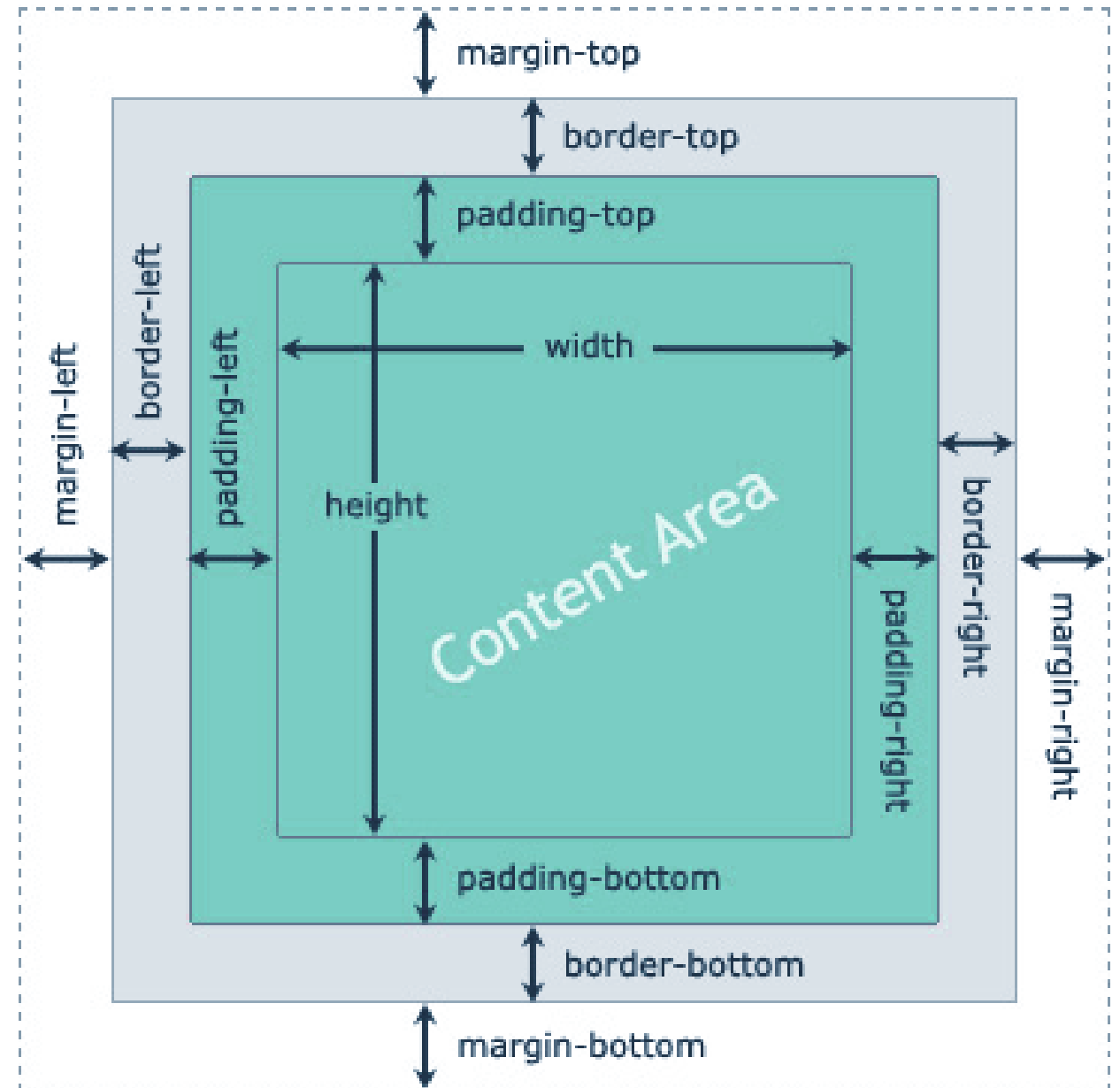
## UT03 – CSS

### 5 – Modelo de caja - Valores y unidades

# Modelo de caja en CSS – Partes de la caja

El modelo de caja interpreta los elementos como una caja, un rectángulo, formado por diferentes partes, que desde el interior al exterior de la caja son:

- Contenido (content).
- Relleno (padding).
- Borde (border).
- Margin (margen).



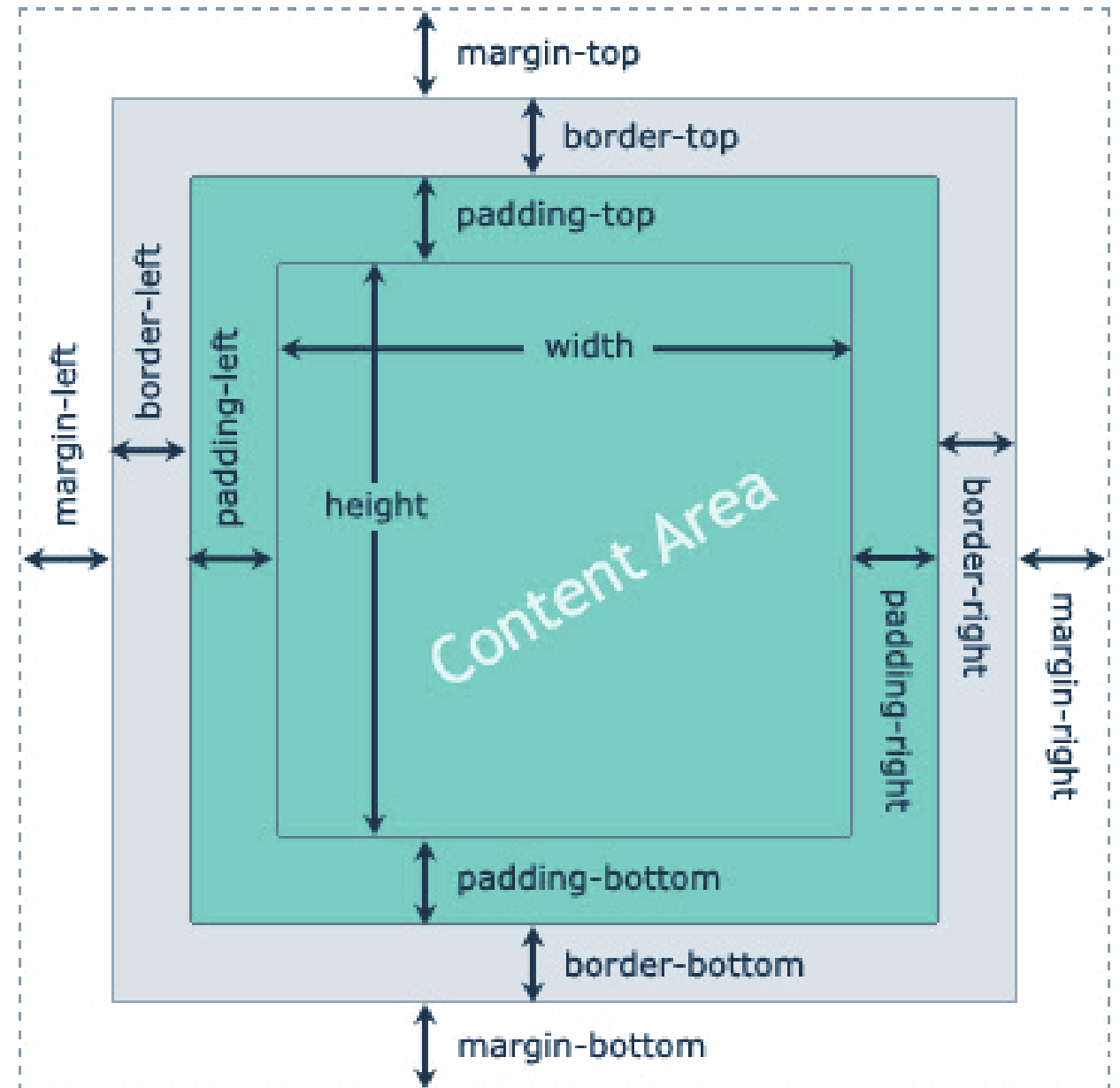
# Modelo de caja en CSS – Partes de la caja

## Contenido

- Definido por el contenido o los descendientes del elemento. Puede ser simplemente un texto o pueden ser múltiples elementos.

## Padding

- Espacio entre el contenido y el borde del elemento.
- Transparente. Si se ha definido un fondo para el elemento, este fondo se verá bajo el padding.



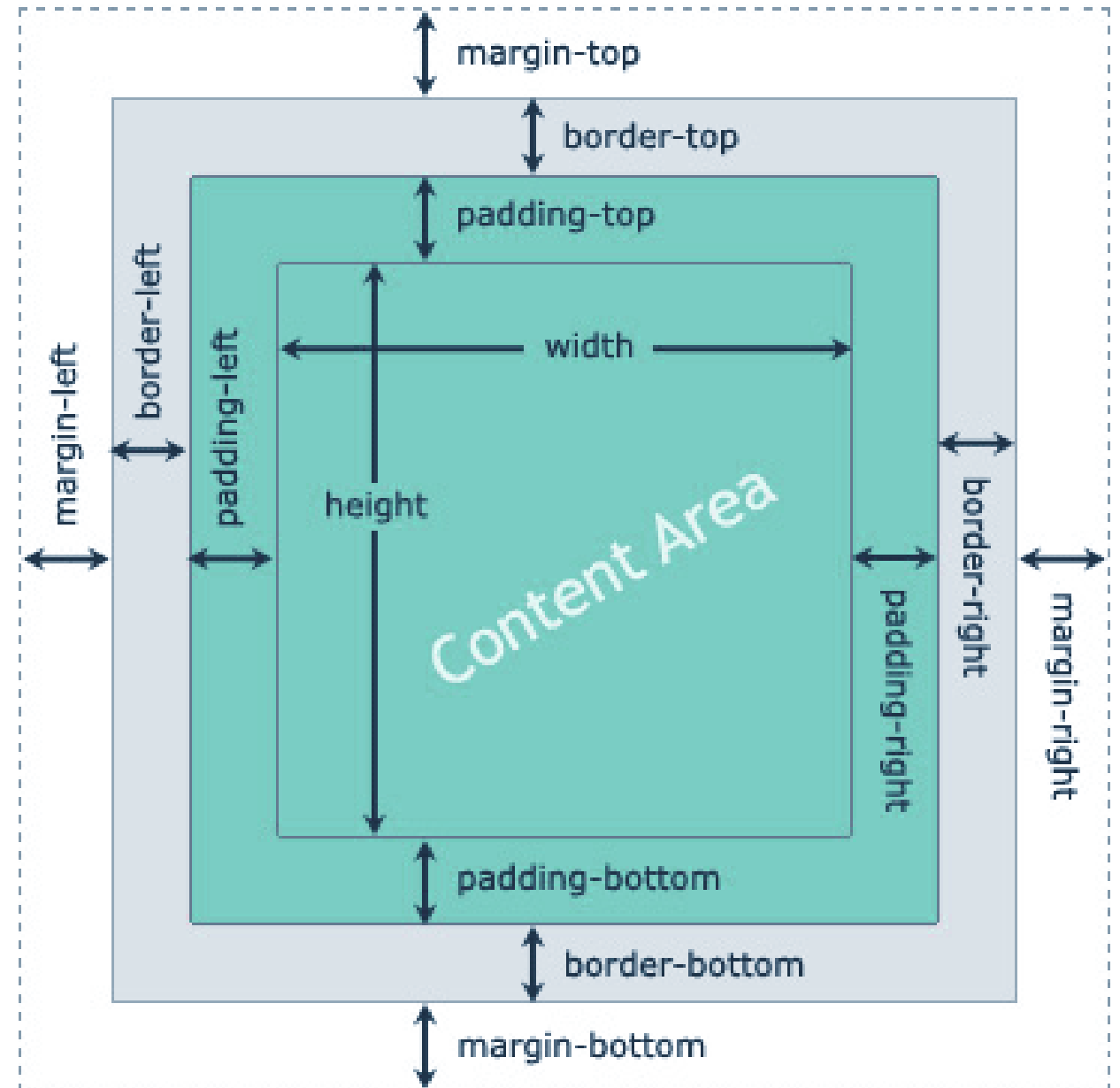
# Modelo de caja en CSS – Partes de la caja

Border:

- Espacio entre el límite exterior del padding y el margen del elemento.

Margin:

- Separa el elemento los elementos adyacentes.
- Es exterior y transparente.
- El fondo del elemento, si lo tiene, no se ve bajo el margen.



# Propiedades del modelo de caja

Para el contenido:

- width y height: ancho y alto. No se aplican a los elementos en línea. Es una dimensión, un tamaño.
- min-width y min-height: ancho y alto mínimo. Evitan que las propiedades width y height sean menores que cierto valor.
- max-width y max-height: ancho y alto máximo. Evitan que las propiedades width y height sean mayores que cierto valor.

# Propiedades del modelo de caja

Para el padding:

- padding-top, padding-right, padding-bottom, padding-left. También una dimensión, un tamaño.
- Atajo: padding. El atajo puede ser:
  - 4 valores: { padding arriba derecha abajo izquierda }
  - 1 valor: { padding todos } El mismo para los cuatro lados.
  - 2 valores: { padding arriba-abajo derecha-izquierda }
  - 3 valores: { padding arriba izquierda-derecha abajo }

# Propiedades del modelo de caja

Para el borde:

- Ancho del borde:
  - `border-top-width`, `border-right-width`, `border-bottom-width`, `border-left-width`. Es una dimensión, un tamaño.
  - Atajo `border-width`. Funciona con el mismo sistema de uno, dos, tres o cuatro valores.
- Estilo del borde:
  - `border-top-style`, `border-right-style`, `border-bottom-style`, `border-left-style`. Pueden ser `none`, `solid`, `dotted`, `dashed`, etc.
  - Atajo `border-style`. Funciona con el mismo sistema de uno, dos, tres o cuatro valores.

# Propiedades del modelo de caja

Para el borde:

- Color del borde:
  - border-top-color, border-right-color, border-bottom-color, border-left-color. Es un color (nombre, hexadecimal, función...)
  - Atajo border-color. Funciona con el mismo sistema de uno, dos, tres o cuatro valores.
- Atajo border.
  - Es un atajo para border-width, border-style y border-color cuando queremos aplicar el mismo valor a todos los bordes del elemento.
  - El orden de los valores no importa. Como son dimensiones diferentes (color, estilo y dimensiones) se identifica qué estamos indicando, aunque no estén ordenadas.



# Propiedades del modelo de caja

Para el margin:

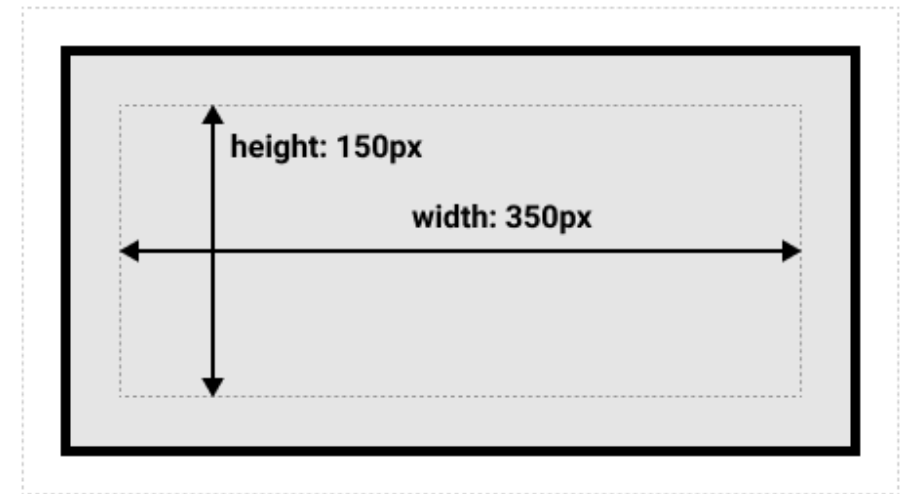
- margin-top, margin-right, margin-bottom, margin-left
- Atajo: margin. El atajo puede ser:
  - 4 valores: {margin arriba derecha abajo izquierda }
  - 1 valor: {margin todos } El mismo para los cuatro lados.
  - 2 valores: {margin arriba-abajo derecha-izquierda}
  - 3 valores: {margin arriba izquierda-derecha abajo }

# Modelo de caja clásico

En el modelo de caja clásico, cuando damos ancho o alto (propiedades width o height) estamos dando el tamaño al contenido.

El elemento ocupará, además del ancho y alto indicado, el espacio indicado para padding, border y margin.

```
.caja-clasica {  
  width: 350px;  
  height: 150px;  
  margin: 10px;  
  padding: 25px;  
  border: 5px solid black;  
}
```



Alto total =  $150 + (25 * 2) + (5 * 2)$ , y se separará 10 de su entorno.

Ancho total =  $350 + (25 * 2) + (5 * 2)$ , y se separará 10 de su entorno.

# Modelo de caja alternativo

El modelo de caja clásico puede ser problemático en algunos escenarios, porque es complicado saber cuánto ocupará realmente un elemento.

Para estas situaciones se puede activar el modelo de caja alternativo, con la propiedad `box-sizing`. Por defecto es `content-box` (clásico), pero se puede cambiar a alternativo con `border-box`.

En este modelo, el ancho y alto (`width` y `height`) se aplican a toda la caja, sin incluir el margen, que no se considera parte de la caja.

En este caso el ancho o alto del contenido se calcula restando de la caja las dimensiones de borde y `padding`.

# Modelo de caja alternativo

Ejemplo:

```
.caja-alternativa {  
    box-sizing: border-box;  
    width: 350px;  
    height: 150px;  
    margin: 10px;  
    padding: 25px;  
    border: 5px solid black;  
}
```



Alto del contenido =  $150 - (25 * 2) - (5 * 2)$ , y se separará 10 de su entorno.

Ancho del contenido =  $350 - (25 * 2) - (5 * 2)$ , y se separará 10 de su entorno.

# Caja en elementos en bloque

En los elementos en bloque en CSS:

- Su propiedad display es "block".
- La caja fuerza un salto de línea al final.
- La caja ocupa todo el ancho disponible, llegando al 100% si es posible.
- Se respetan las propiedades width y height
- El relleno, margen y borde mantienen alejados a otros elementos.

Ejemplos de estos elementos son <p>, <h1>, <div>, <ul>, ...

Cualquier elemento no en bloque se puede convertir a elemento en bloque cambiando su display a "block".

# Caja en elementos en línea

En los elementos en línea en CSS:

- Su propiedad display es "inline".
- La caja no fuerza ningún salto de línea. Los elementos en línea se suceden en la misma línea uno tras otro.
- La caja ocupa sólo el ancho necesario para su contenido.
- No se aplican las propiedades width y height.
- El relleno, margen y borde se respetan, pero no mantienen alejados a otros elementos. Se produce cierto solapamiento.

Ejemplos de estos elementos son <a>, <img>, <span>, <strong>, ...

Cualquier elemento no en línea se puede convertir a elemento en línea cambiando su display a "inline".

# A medio camino entre bloque y en línea

Hay una forma de hacer una mezcla entre en bloque y en línea.

- Su propiedad display es "inline-block".
- No fuerza salto de línea. Como un elemento en línea.
- Ocupa sólo el ancho necesario para su contenido, salvo si se indica width y/o height. En este caso se respeta, como en bloque.
- El relleno, margen y borde mantienen alejados a otros elementos, como en bloque

No hay elementos con este comportamiento. Hay que forzarlo.

Útil para dar márgenes a elementos en línea en el texto.

Para otras aplicaciones (menús en una sola línea, por ejemplo), se ha visto sobrepasado por flex y grid, que veremos más adelante.

# Colapso de márgenes

Cuando se trabaja con márgenes verticales (superior e inferior), hay que tener en cuenta que estos márgenes no se acumulan.

Es decir, si tengo dos elementos en bloque (dos p, por ejemplo) seguidos, el primero tiene un margen inferior de 20px y el segundo un margen superior de 50px, la separación no será de 70px.

En estos casos el margen aplicado será el mayor de los dos, así que serán sólo 50px.

Este comportamiento se denomina colapso de márgenes, y se puede producir en otras circunstancias, como que se colapse el margen de un elemento y el de su contenedor.

Más información en [Entendiendo el colapso de márgenes – MDN](#).



# Valores CSS

Todas las propiedades CSS tienen una serie de valores admitidos.

En la referencia CSS en MDN hay siempre una sección para cada propiedad en la que se muestran los valores admitidos.

Por ejemplo, si miramos la propiedad "padding-left" en MDN, encontramos una sección "Values", en la que podemos encontrar que esta propiedad admite "<length>" y "<percentage>".

Estos valores que aparecen entre símbolos <> indican que puede tomar un valor más o menos arbitrario, pero siguiendo normas. Una longitud (length) puede expresarse de varias formas (px, pt, em, etc.). Un porcentaje es un número seguido del símbolo %.

Otro ejemplo: si miramos text-align vemos que admite sólo valores concretos: start, end, left, right, center, justify, justify-all y match-parent.

# Valores CSS – Dimensiones

Hay ciertos valores CSS que identifican dimensiones. Estas dimensiones pueden ser (entre otras):

- Longitud. Por ejemplo, el alto o ancho de un elemento, el relleno (padding) o el margen, se establecen con longitud.
- Ángulos. En CSS los ángulos se miden en sentido de las agujas del reloj (a diferencia del estándar matemático, que mide en sentido contrario a las agujas del reloj)
- Tiempo. Usado en transiciones y animaciones.

Hay otros valores CSS que se refieren a color o posición, pero vamos a centrarnos de momento en dimensiones, porque es la mejor forma de entender el sistema de unidades de CSS.

# Valores CSS – Valores universales

Hay ciertos valores que se pueden aplicar a cualquier propiedad CSS:

- initial: devuelve la propiedad a su valor por defecto.
- inherit: calcula el valor de la propiedad en función del elemento padre. No hace nada si la propiedad no es heredable.
- unset: actúa como initial o inherit, dependiendo de la propiedad:
  - Si es heredable, será inherit
  - Si no lo es, será initial
- revert: similar a unset, pero si no es heredable volverá al valor establecido por los estilos de usuario o del navegador, si los hay.
- revert-layer: elimina un nivel de la cascada.

Los más utilizados son initial e inherit.

# Dimensiones en CSS y sus unidades

Una dimensión es un número con unas unidades asociadas. Por ejemplo: 30deg, 100ms, 50px, 2em, etc.

Las unidades son case-insensitive, podemos usar tanto 20px como 20PX.

Las unidades se escriben sin espacio entre el número y la unidad.

En CSS se usan dimensiones para especificar:

- Longitudes o distancias
- Ángulos
- Tiempo y frecuencia
- Resolución
- Flex. Similar a un porcentaje, pero es una fracción de cierta parte del elemento. Lo veremos cuando veamos flex y grid.

# Unidades CSS de distancia

Las unidades de longitud o distancia (o tamaño) pueden ser:

- Absolutas: definen un valor absoluto, que no depende de otros elementos en el documento.
- Relativas: definen un valor relativo al tamaño de otra cosa. Hay diferentes unidades que toman como referencia distintos elementos del documento u otros elementos, como la parte visible de la página.
- Relativas al contenedor de una media query. Utilizadas junto a las media queries para hacer diseños responsive.

# Unidades absolutas

Son unidades con un tamaño fijo, relativo al cm o a la pulgada.

- Píxeles (px): 1 píxel de la pantalla.  $1\text{px} = 1/96$  pulgadas.
- Puntos (pt):  $1\text{pt} = 1/72$  pulgadas.
- Picas (pc):  $1\text{pc} = 1/6$  pulgadas.
- Pulgadas (in):  $1\text{in} = 2.54\text{ cm} = 96\text{px}$ .
- Cuarto de milímetro (q):  $1\text{q} = 0.25\text{ mm} = 1/40\text{ cm}$
- Milímetro (mm):  $1\text{mm} = 1/10\text{ cm}$
- Centímetro (cm):  $1\text{cm} = 10\text{mm} = 96\text{px}/2.54$

Son útiles sobre todo para definir distancias en medios de tamaño fijo, como papel. Un A4 es siempre de las mismas dimensiones, pero una pantalla de móvil, tablet o PC puede tener tamaños variados.

# Unidades relativas

Son unidades con un tamaño relativo a otros elementos del documento.  
Pueden ser:

- Relativas al tamaño de fuente.
- Relativas a otra distancia.
- Relativas al viewport. El viewport es la porción de documento que se está visualizando.

# Unidades relativas a la fuente

Toman como referencia la fuente de un elemento del documento.

- em. Relativa al tamaño de la fuente (font-size) del elemento.
  - 1em es el 100% del tamaño de fuente.
  - 0.5em es el 50% y 2em es el doble del tamaño de fuente.
  - Muy usado en line-height, font-size, margin-top y margin-bottom.
- rem. Igual que em, pero relativa al tamaño de fuente de la raíz del documento. Es la mejor para crear interfaces escalables.
- ex. Relativa a la altura de la x minúscula de la fuente del elemento. En fuentes que tienen la x minúscula, es muy habitual que  $1\text{ex} = 0.5\text{em}$ .
- cap. Relativa al alto de las mayúsculas de la fuente del elemento.



# Unidades relativas al viewport

El viewport es la región del documento que está siendo visualizada en ese instante.

Es la parte del documento visible en la pantalla, es decir, se excluyen barras de herramientas del navegador, o el resto de la pantalla si el navegador no está maximizado.

Si el documento se está visualizando en pantalla completa, modo en que no se ven las herramientas del navegador ni otras aplicaciones, coincidirá con el tamaño de la pantalla.

En HTML 5 hay una directiva viewport que podemos poner en un tag meta en la cabecera para dar indicaciones al documento de cómo se tiene que visualizar. Suele funcionar sólo en dispositivos móviles. Los navegadores de escritorio.

# Unidades relativas al viewport

- vw. Relativa al ancho del viewport.  $1\text{vw} = 1\%$  del ancho del viewport.
- vh. Relativa al alto del viewport.  $1\text{vh} = 1\%$  del alto del viewport.
- vi. Relativa al eje "inline" del elemento raíz. El eje inline es el eje en sentido de escritura. Esta unidad es una unidad lógica que usa para dar soporte a escrituras verticales y horizontales. En nuestro estándar "left-to-right" vi es lo mismo que vw.
- vb. Relativa al eje "block" del elemento raíz. El eje block es el perpendicular al sentido de escritura. En nuestro estándar "left-to-right" vi es lo mismo que vh.
- vmin y vmax. Relativas a la menor dimensión (alto/ancho) del viewport. Igual que las otras,  $1\text{vmin}$  será el 1% del menor de los dos valores.

# Unidades relativas a la altura de línea

Toman como referencia la altura de línea (line-height) de un elemento del documento.

- lh. Relativa a la altura de línea del elemento.
- rlh. Relativa a la altura de línea del elemento raíz del documento.

# Otras - Ángulos y tiempo / frecuencia

## Ángulos:

- deg. Grados sexagesimales. Hay 360 grados en un círculo completo.
- grad. Gradianes. Hay 400 gradianes en un círculo completo.
- rad. Radianes. Hay  $2\pi$  radianes en un círculo completo.
- turn. Vueltas o revoluciones. Hay una vuelta en un círculo completo

## Tiempo

- s y ms. Segundo y milisegundo.  $1s = 1000ms$ .

## Frecuencia

- hz y khz. Hercio y Kilohercio.  $1Hz$  1 vez / segundo.  $1khz = 1000hz$ .

# Valores CSS – Color – Image

Un valor de color (<color>) puede ser:

- Palabras clave de colores: red, green, transparent, etc.
- Valores hexadecimales completos o abreviados, con o sin canal alfa.
- Valores de color (con o sin canal alfa) a partir de funciones, como rgb(), rgba(), hls(), hlsa(), etc.

Un valor de imagen (<image>) puede ser:

- Un archivo de imagen al que se accede a través de la función url()
  - URL absoluta: url(https://sitioweb.com/imagen.jpg).
  - URL relativa a la ubicación del CSS: url(imagen.png)
- Un gradiente (degradado)

# Valores CSS – Otros

Otros valores CSS que iremos viendo más adelante son:

- Posición (<position>): top, left, right, bottom, center. También puede ser una combinación de varios o combinarse con porcentajes u otras dimensiones.
- Cadenas de texto (<string>): usadas en ocasiones para indicar contenido antes o después de los elementos.