

Week 3_3(es)

Modelo de Caja(The Box Model)

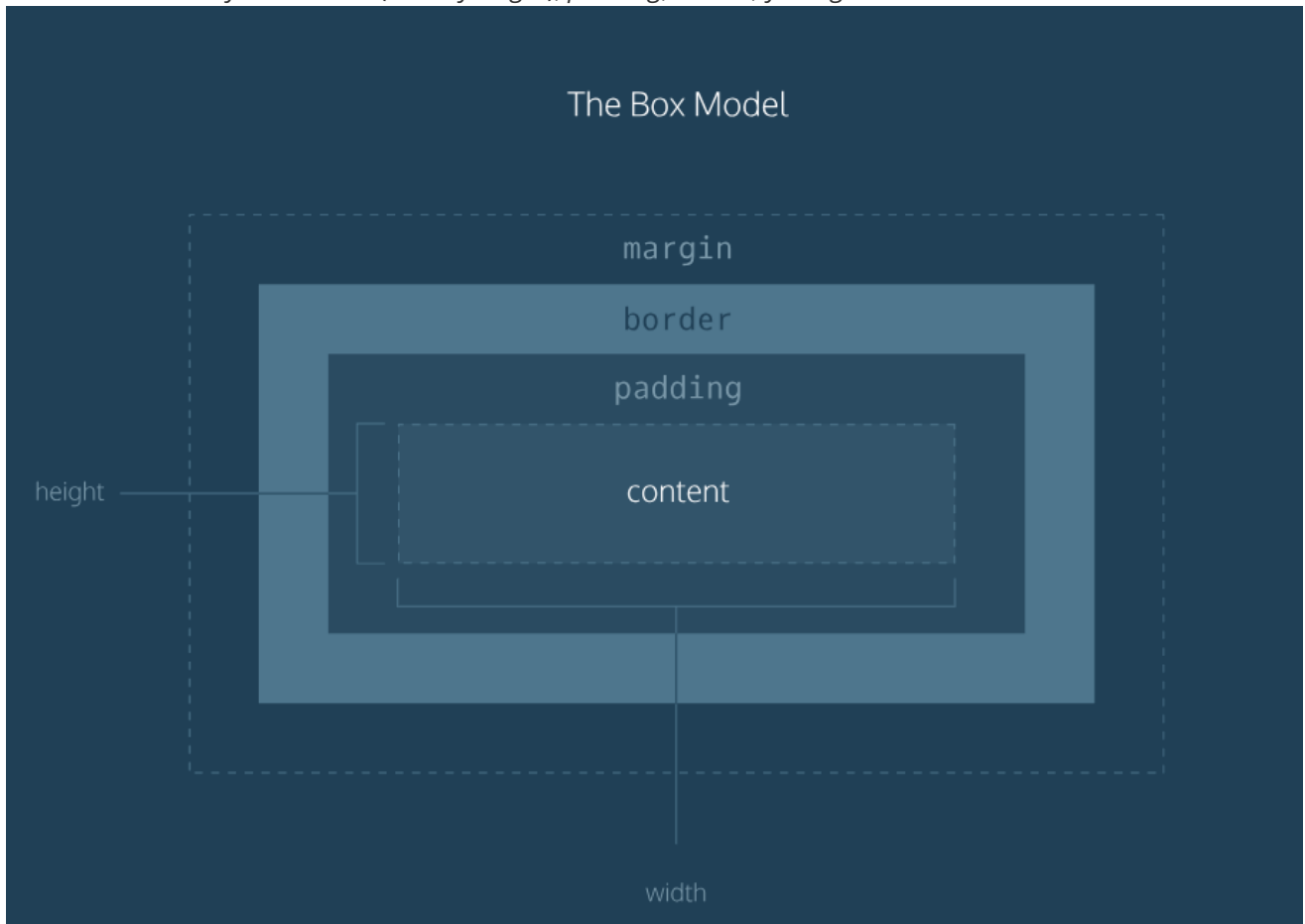
El *box model* es un concepto muy importante para entender como los elementos están situados y demostrados en la página web. Vamos a aprender sobre el box model!

El Concepto de Box Model

El concepto principal de box model es que **todos los elementos en una página web puede interpretarse por un navegador como una vida dentro de una caja.**

Básicamente, una página web es una colección de las cajas.

El box model incluye el tamaño(*width* y *height*), *padding*, *border*, y *margin*.



La imagen arriba es una representación visual del box model.

Cada propiedad indica el siguiente:

1. **Width and Height:** ancho y altura del área del contenido
2. **Padding:** espacio entre el área del contenido y el borde
3. **Border:** el espesor y el estilo del borde que envuelve el área del contenido y padding
4. **Margin:** espacio entre el border y el borde fuera del elemento

Height y Width

El contenido de un elemento tiene dos dimensiones: altura y ancho. Por defecto, las dimensiones de un box de HTML envuelve los contenidos crudos del box. Las propiedades `height` y `width` se pueden usar para modificar las dimensiones.

Vamos a poner `width` y `height` de la clase `.article-body` como se muestra abajo.

```
.article-body {  
  background-color: orange;  
  height: 650px;  
  width: 700px;  
}
```

La propiedad `background-color` está agregada para revisar el tamaño del box.

En el navegador, podemos ver que la altura y el ancho son 650 píxeles y 700 píxeles respectivamente.

Usamos píxeles para decidir el tamaño exacto del box de un elemento.

Nota: Cuando el ancho y la altura de un elemento están en píxeles, en todos los aparatos el box será el mismo tamaño. (Algunos elementos con el tamaño adecuado para la pantalla de computador portátil se demarraría en la pantalla de un celular.)

Border

Border se usa para dibujar la línea que envuelve un elemento.

Border puede ser específico con las siguientes propiedades:

1. `width`: el espesor de border (en píxeles o `thin` / `medium` / `thick`)
2. `style`: diseño de border (incluye `none`, `dotted`, `solid`)
[Todos los estilos de border](#)
3. `color`: el color de border ([built-in color keywords](#), [RGB cubic-coordinate system](#), o [HSL cylindrical-coordinate system](#))

Vamos a agregar la siguiente declaración a la regla de `.article-body`.

```
border: 3px solid darkorchid;
```

Como podemos ver en el código arriba, todas las tres propiedades están en una línea del código, separadas por un espacio. Por defecto, la propiedad de border tiene el valor de `medium none color`, en el que el `color` es el color actual del elemento.

Si una o dos de las tres propiedades no están especificadas en el archivo de CSS, el valor por defecto va a mostrarse.

Con la propiedad de `border-radius`, podemos crear un box con el borde redondeado o podemos crear un círculo.

Vamos a agregar la siguiente regla a `.article-body`.

```
border-radius: 5px;
```

Los cuatro esquinas del borde tienen radio de 5 píxeles. Eso significa que las esquinas tiene la misma curvatura de un círculo con el radio de 5 píxeles.

Si ponemos el valor de radio igual como la altura del box o como 100%, tendremos un círculo perfecto.

Padding

Padding es el espacio entre el contenido y el borde del box, como el espacio entre la pintura y el marco en la imagen abajo.



Vamos a agregar la siguiente declaración a la regla de `.article-body` y vamos a revisar la diferencia en el navegador.

```
padding: 10px;
```

Pueden ser más específico en padding con las siguientes propiedades.

1. `padding-top`
2. `padding-right`
3. `padding-bottom`
4. `padding-left`

Atajos para la propiedad `padding`

Hay atajos para especificar cada lado de padding.

En vez de escribir

```
padding-top: 6px;  
padding-right: 11px;  
padding-bottom: 4px;  
padding-left: 9px;
```

Pueden escribirlo de una vez como pueden ver abajo.

```
padding: 6px 11px 4px 9px;
```

Cada valor significa cada lado en el sentido de las agujas del reloj.

Cuando utilizan esa manera, **tienen que ponder valores para todos cuatro lados.**

Si los valores para top y bottom son iguales, y los valores de left y right son iguales como abajo,

```
padding-top: 5px;  
padding-right: 10px;  
padding-bottom: 5px;  
padding-left: 10px;
```

pueden utilizar el siguiente atajo.

```
padding: 5px 10px;
```

Margin

Margin significa el espacio afuera del box.

Esta propiedad se usa cuando quieren que varios boxes **tengan espacio entre ellos.**

Vamos a agregar la siguiente declaración a la regla de `.article-body` y vamos a revisarlo en el navegador.

```
margin: 20px;
```

Como la propiedad `padding`, pueden ser específicos sobre margin en cada lado del box con las siguientes propiedades:

1. `margin-top`

2. `margin-right`
3. `margin-bottom`
4. `margin-left`

Es común ver valores de margin usado para un lado específico de un elemento.

Atajos para la propiedad `margin`

Como la propiedad de `padding`, la propiedad de `margin` tiene el mismo atajo.

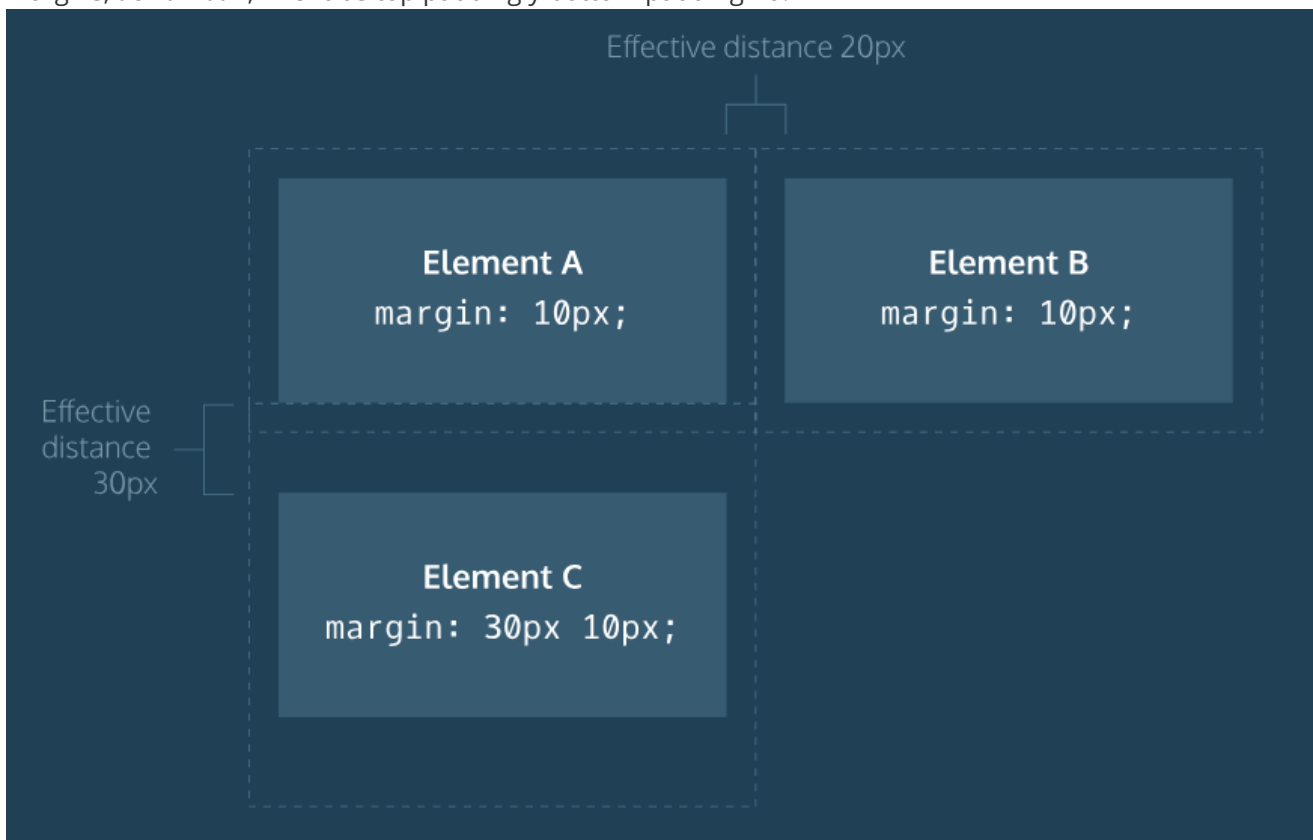
```
margin: 6px 10px 5px 12px;
```

```
margin: 5px 10px;
```

Ambos códigos son válidos en CSS.

Derrumbamiento de Margin(Margin Collapse)

Una diferencia entre `padding` y `margin` es que top margin y bottom margin, también se llaman vertical margins, *derrumban*, mientras top padding y bottom padding no.



Horizontal margins (left y right) siempre se muestran y se añaden juntos.

Pero, *vertical margins no se añaden*.

El valor más grande entre dos vertical margins va a ser la distancia entre elementos contiguos.

Overflow

Todos los componentes (width, height, padding, border, margin) constituyen el tamaño de un elemento. Pero a veces, estos componentes resultan en un elemento **más grande que el área designada**. Para ver todos los elementos, utilizamos la propiedad de `overflow`. La propiedad de `overflow` decide cómo se muestra el contenido que rebalsa fuera del box. Puede ser uno de los siguientes:

1. `hidden`: el contenido que rebalsa estará **escondido**
2. `scroll`: un **scrollbar** va a agregarse al box para que el resto del contenido puedan aparecerse por scrolling
3. `visible`: el contenido rebalsado estará mostrado **fuera** de contenido no rebalsado. (por defecto)

Por ejemplo, si la propiedad `overflow` de `div` es `scroll`, saldrá un scrollbar para mostrar el contenido rebalsado.

Vamos a cambiar `height` y añadir la propiedad `overflow` a la regla de `.article-body`.

```
height: 300px;
overflow: scroll;
```

Visibility

Los elementos pueden desaparecer con la propiedad de `visibility`.

La propiedad de `visibility` puede tener los siguientes valores:

1. `hidden`: oculta un elemento
2. `visible`: muestra un elemento (por defecto)

Vamos a agregar la siguiente declaración a la regla de `.article-body`.

```
visibility: hidden;
```

Pueden ver que el contenido del elemento está escondido *pero todavía existe un espacio vacío donde el elemento estaba antes*.

(Para quitar el elemento completamente desde la página web, necesitamos utilizar `display: none;`)