

# ALINEACIÓN VERTICAL DE ELEMENTOS

Salazar, Wilsón Enrique, COD: 092502  
ws092502@ingenieria.sanmartin.edu.co  
Fundación Universitaria San Martín

*Resumen*—CSS nivel 2 no tiene una propiedad para el centrado vertical de objetos. Con CSS2 es posible centrar bloques verticalmente, mediante la combinación de unas pocas propiedades. El truco consiste en indicar que el bloque exterior sea formateado como celda de tabla, porque los contenidos de celdas sí se pueden centrar verticalmente.

*Índice de Términos*—

## I. ALINEACIÓN VERTICAL DE ELEMENTOS

Una buena razón por lo que muchas personas defienden el uso de tablas para la diagramación de contenidos de un sitio web, es porque éstas permiten el correcto centrado horizontal y vertical de los elementos contenidos dentro de sus celdas. La propiedad vertical-align no es soportada por los browsers actuales, y va a ser lanzada con CSS3

Por motivos de estética, en ocasiones es necesario realizar alineación vertical de elementos. En los tiempos cuando las web se trabajaban con tablas esta propiedad era fácil de aplicar con la propiedad vertical-align, con la aparición de las capas Divs se tienen que idear trucos para poder alinear.

A continuación se explicaran unos trucos para realizar esta función de alineamiento.

## II. MÉTODO DEL ALTO DE LINEA (LINE-HEIGHT METHOD)

Es un método que solo funciona con una línea de texto, puede ser útil para mostrar un tweet etc.

Implementación:

```
<div id="padre">
  <div id="hijo">Texto alineado verticalmente.</div>
</div>

#hijo { line-height: 200px; //Damos 200px de alto para
notar el efecto }
```

## EL MÉTODO DE LA TABLA CSS (CSS TABLE METHOD)

Este método se basa en el posicionamiento vertical como anteriormente se utilizaba con las tablas y vertical-align. Lo que se realiza es mediante CSS una serie de capas como tablas HTML o celdas de tabla, con esto se logra que el navegador vea los taps de divs como tablas.

Cabe resaltar que las tablas de HTML no son iguales a las de CSS

```
<div id="padre">
  <div id="hijo">Contenido a centrar aquí.</div>
</div>
```

```
#padre {display: table;
        height:200px;}
#hijo {display: table-cell;
       vertical-align: middle;}
```

## POSICIÓN ABSOLUTA Y MARGEN NEGATIVO (ABSOLUTE POSITIONING AND NEGATIVE MARGIN)

Este método sólo funciona para elementos de tipo bloque (perfecto para las capas) y también funciona para todos los navegadores.

```
<div id="padre">
  <div id="hijo">Contenido alineado aquí.</div>
</div>
```

```
#padre {position: relative;}
#hijo {position: absolute;
      top: 50%;
      left: 50%;
      height: 30%;
      width: 50%;
      margin: -15% 0 0 -25%;}
```

Si nos fijamos en el CSS (El HTML no varía con los ejemplos anteriores) vemos que lo primero aplicamos un posicionamiento relativo a la capa padre y una posición absoluta al elemento hijo.

Después fijamos los valores top y left del elemento hijo al 50% (centro del elemento padre). Esto aún no es correcto.

Para que sea correcto necesitamos mover el elemento hijo hacia arriba (la mitad de su altura) y a la izquierda (la mitad de su ancho) para que lo que quede en el centro del elemento padre sea el centro del elemento hijo. Para esto necesitamos saber el tamaño de la capa (ancho y alto) y así poder restar el margen correspondiente.

Para hacer esto le damos un margen negativo al elemento hijo tanto superior como izquierdo igual a la mitad de su alto y su ancho respectivamente:

Tened en cuenta que habíamos dado un ancho del 50% y un alto del 30%, por eso el margen superior es del -15% (la mitad de 30) y el margen izquierdo es del -25% (mitad del ancho).

### POSICIÓN ABSOLUTA Y EXTENSIÓN (ABSOLUTE POSITIONING AND STRETCHING)

Tal y como hacíamos con el método anterior, vamos a aplicar una posición relativa al elemento padre y una absoluta al elemento hijo.

En el código que os muestro, utilizo el método para centrar tanto vertical como horizontalmente:

```
<div id="padre">
  <div id="hijo">Contenido aquí.</div>
</div>

#padre {
  position: relative;
}
#hijo {
  position: absolute;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  width: 50%;
  height: 30%;
  margin: auto;
}
```

La idea de este método es intentar que el elemento hijo se centre indicando un top, bottom, right y left con valor 0.

Como el elemento hijo es más pequeño que su elemento padre no podrá alinear las 4 esquinas. Indicando un margen automático a las 4 esquinas haremos que los márgenes opuestos (izquierda-derecha y arriba-abajo) sean iguales y mostrarán nuestra capa hijo en el centro del elemento padre.

### MISMO PADDING SUPERIOR E INFERIOR (EQUAL TOP AND BOTTOM PADDING)

En el anterior método le dejábamos al navegador la tarea de

calcular automáticamente los márgenes del elemento hijo para que sus opuestos fueran idénticos.

En este método haremos algo similar, pero lo indicaremos nosotros.

```
<div id="padre">
  <div id="hijo">Contenido aquí.</div>
</div>

#padre {padding: 5% 0;}
#hijo {padding: 10% 0;}
```

Con este CSS estamos especificando para ambos elementos (padre e hijo) el padding superior e inferior que deben tener.

Este método puede dar mucho juego si usamos medidas relativas para que se adapte automáticamente al alto de su contenido y demás, pero no es el tema de esta entrada. Realizaremos un ejemplo con medidas absolutas.

### DIV FLOTANTE (FLOATER DIV)

Este último método requiere un div vacío que flota y se utiliza para controlar dónde se encuentra el elemento hijo dentro del documento. El elemento flotante viene antes que el elemento hijo en el documento HTML.

```
<div id="padre">
  <div id="flotante"></div>
  <div id="hijo">Contenido aquí.</div>
</div>

#padre {height: 250px;}
#flotante {float: left;
  height: 50%;
  width: 100%;
  margin-bottom: -50px; }
#hijo {clear: both;
  height: 100px; }
```

Haremos que la capa flotante flote (valga la redundancia) a la izquierda (o derecha, da lo mismo) y le daremos un alto del 50% de su elemento padre. Esto hará que quede “ocupada” la mitad superior del elemento padre.

Como este elemento flotante es suprimido del flow normal del documento necesitamos limpiar el elemento hijo. De ahí que utilizemos clear:both, aunque realmente sólo necesitamos limpiar en el sentido que hicimos flotar la capa vacía.

La parte superior del elemento hijo quedará inmediatamente debajo de la parte inferior del elemento flotante. Necesitamos que el elemento hijo suba la mitad de su altura; para ello utilizamos un margen negativo como vimos anteriormente.

En esta ocasión el margen negativo lo aplicamos a la parte inferior de la capa flotante.

#### REFERENCIAS

- [1] <http://dariobf.com/centrar-verticalmente-con-css-6-formas-diferentes-de-lograrlo/>
- [2] <http://www.codedrinks.com/centrar-verticalmente-el-contenido-de-un-div-con-css/>
- [3]