[[1]](#footnote-1)

Centrar verticalmente con CSS

Lòpez, Juan.

jl091505@ingenieria.sanmartin.edu.co

Fundación Universitaria San Martin

*Resumen*— A veces, por motivos de estética, nos viene bien alinear verticalmente un sitio web.

En los tiempos en que las webs se hacían con tablas, esto era sencillo gracias a la propiedad vertical-align de las mismas; pero con la aparición de las capas (Divs) y desaparición de las tablas esta propiedad murió.

Hoy traigo 6 formas diferentes de alinear una web verticalmente con CSS sin morir en el intento, totalmente validadas por el W3C y que dará ese toque que le falta a tu web.

*Índice de Términos—Centrar,vertical,hijo, padre, align*

# Métodos Usados.

## Método del alto de linea (Line-height method)

Aunque es un método que sólo funciona con una línea de texto, puede ser útil para mostrar, por ejemplo, nuestro último tweet (muy de moda) en una única línea.

La forma de implementarlo sería la siguiente:

Resoluciones

<div id="padre">

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

</div>

#hijo {

line-height: 200px; //Damos 200px de alto para notar el efecto

Sólo a una línea de texto.

## El método de la tabla CSS (CSS Table method)

Lo que hacemos es tratar, mediante CSS, a una serie de capas como tablas HTML o celdas de tabla; con esto lograremos que el navegador vea dichas divs como tablas y podamos utilizar la propiedad vertical-align sobre ellas.

Debemos tener clara la idea de que las tablas CSS no son lo mismo que las tablas HTML.

La implementación sería así:

<div id="padre">

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

</div>

#padre {

display: table;

height:200px;

}

#hijo {

display: table-cell;

vertical-align: middle;

}

se otorga a la capa padre la propiedad display:table y a la capa hijo display:table-cell. Esto hará que la capa padre sea interpretada por el navegador como una tabla y la capa hijo como una celda de tabla.

A su vez, la capa hijo tendrá un vertical-align: middle para que se produzca el alineado vertical.

También tenemos en cuenta que en esta ocasión el alto de la capa va en el elemento padre.

Me gustaría advertir que este método no funciona por sí mismo en versiones antiguas de Internet Explorer, por lo que hay que “forzarlo” añadiendo un display:inline-block al elemento hijo.

## 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%;

}

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:

Para tener en cuenta que se habia 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).

Como vais a poder ver en el demo, lo que centramos es una capa y no el texto en sí. Para esto habría que hacer otras técnicas que nos permitiesen centrarlo verticalmente. Nos hemos interesado en centrar la capa.

## Posición absoluta y extensión (Absolute Positioning and Stretching)

Vamos a aplicar una posición relativa al elemento padre y una absoluta al elemento hijo.

<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)

<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.

## Doble centrado medidas desconocidas con transform

Cuando se desconocen las medidas del hijo es recurrir a las transform.

.centrado { position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%); }

## Doble centrado múltiples elementos en tamaños desconocidos o ventanas dinámicas.

Esta forma solventa el problema anterior, ya que no se necesita ningún margen lateral negativo ni superior tan grande. Además añade la ventaja de que no se necesita conocer de antemano los tamaños de los elementos a centrar

display: block;

font-family: Tenor Sans, sans-serif;

color: #444;

font-weight: bold;

text-decoration: none;

line-height: 30px; /\*margin-top of ul plus 2\*/

padding: 0 10px;

text-shadow: 1px 1px 0 rgba(255,255,255,0.1);

border-radius: 4px;}

a:hover {

background: #363636;

color: #5fa9c5;

border: 1px solid #5fa9c5;

}

## Doble centrado sin márgenes negativos con posición absoluta

Pese a que en la vertical el valor 'auto' para 'margin' no surte efecto, al actuar sobre una caja con tamaños declarado y estar posicionada a 0 en las cuatro propiedades (left/top/right/bottom)

#center {

position: absolute;

top:0;

left:0;

right:0;

bottom:0;

margin: auto;

background: #83C24A;

height: 100px;

width: 200px;

box-shadow: 0 0 4px rgba(0,0,0,.3);

}

referencias

1. <http://dariobf.com/centrar-verticalmente-con-css-6-formas-diferentes-de-lograrlo/>
2. <http://ksesocss.blogspot.com/2012/05/centrando-al-centro-con-css-16-maneras.html>

1. [↑](#footnote-ref-1)