ALINEACION VERTICAL DE ELEMENTOS BLOCK EN CSS (CENTRAR)

**William Steven Pedraza Suarez 111515**

*Ingeniería de Sistemas, Fundación Universitaria San Martín,*

*Bogotá D.C.*

wp111517@ingenieria.sanmartin.edu.co

**Abstract – (Line-height method) método del alto de línea, (CSS Table method) método de la tabla, (Absolute Positioning and Negative Margin) Posición absoluta y margen negativo, (Absolute Positioning and Stretching) Posición absoluta y extensión, (Equal Top and Bottom Padding) Mismo padding superior e inferior.**

I. INTRODUCCIÓN

Este artículo pretende definir diferentes métodos para alinear verticalmente elementos block y además ajustarlos al centro del elemento.

Debido a que en cuestión de diseño muchas veces es bastante práctico y estético realizarlo, por esto se investigó sobre el tema. Además la forma de ajustarlo es mediante Css para cambiar las propiedades de un modo más sencillo.

Anteriormente las páginas web se desarrollaban basándose en tablas que poseían la propiedad vertical-align, hoy en día con el concepto de los divs lo de las tablas quedo en el pasado junto con la propiedad.

II (LINE-HEIGHT METHOD)

Este método solo funciona con una línea de texto pero en él se evidencia fácilmente lo que queremos hacer con el elemento block, además funciona en todos los navegadores.

**Ej:**

<div id="padre">

<div id="hijo">

Texto alineado verticalmente

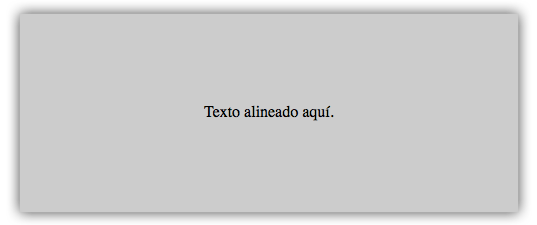
</div>

</div>

#hijo {

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

}



II (CSS TABLE METHOD)

Este método se basa en la forma que se hacía con las tablas pero sin hacerlo con tablas como tal, la idea es tratar con CSS los divs como tablas o celdas de tabla, con esto se logra que el navegador perciba dichos divs como tablas y así poder utilizar la propiedad vertical-align sobre dichos elementos.

**Ej:**

<div id="padre">

<div id="hijo">

Contenido a centrar aquí.

</div>

</div>

#padre {

**display**: table;

**height**:200px;

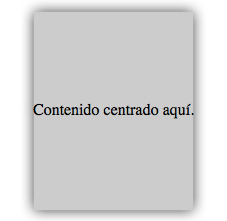
}

#hijo {

**display**: table-cell;

**vertical-align**: middle;

}



III (ABSOLUTE POSITIONING AND NEGATIVE MARGIN)

Este método solo funciona para elementos block, es perfecto para los divs además funciona en todos los navegadores.

Se fija un posicionamiento relativo al div padre y una posición absoluta el elemento hijo, después se fijan los valores top y left del elemento hijo al 50%, luego le damos un margen negativo al elemento hijo tanto superior como izquierdo igual a la mitad de su alto y su ancho respectivamente.

**Ej:**

<div id="padre">

<div id="hijo">

Contenido alineado aquí.

</div>

</div>

#padre {

**position**: relative;

}

#hijo {

**position**: absolute;

**top**: 50%;

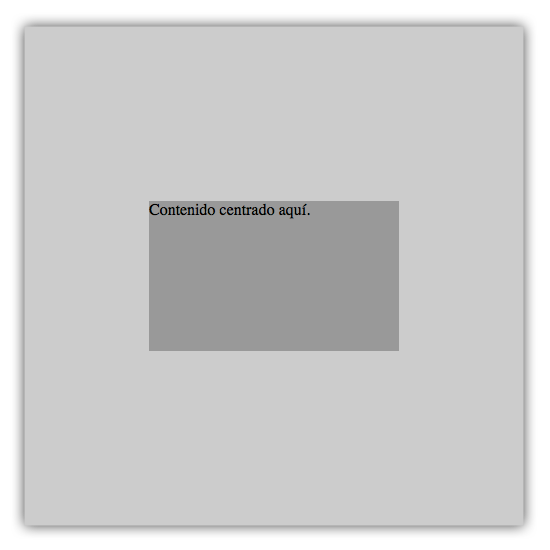
**left**: 50%;

**height**: 30%;

**width**: 50%;

**margin**: -15% 0 0 -25%;

}



IV (ABSOLUTE POSITIONING AND STRETCHING)

Como en el método anterior se aplica una posición relativa al elemento padre y una absoluta al elemento hijo, la idea es intentar que el elemento hijo se centre tanto vertical como horizontalmente indicando un top, bottom, right y left con valor 0.

Con esto el elemento hijo no podrá alinear las 4 esquinas debido a que es más pequeño que su padre, el único problema es que Internet Explorer 7 o inferior no muestre el CSS bien.

**Ej:**

<div id="padre">

<div id="hijo">

Contenido aquí.

</div>

</div>

#padre {

**position**: relative;

}

#hijo {

**position**: absolute;

**top**: 0;

**bottom**: 0;

**left**: 0;

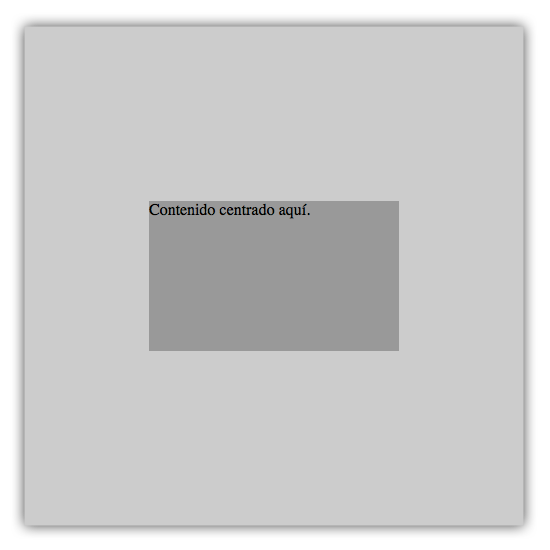
**right**: 0;

**width**: 50%;

**height**: 30%;

**margin**: auto;

}



V (EQUAL TOP AND BOTTOM PADDING)

En el método anterior el navegador hacia la tarea de calcular automáticamente los márgenes del elemento hijo para que sus opuestos fueran idénticos, en este método se hace algo similar pero indicado por nosotros.

Con el CSS especificamos para ambos elementos el padding superior e inferior que deben tener, el problema es que cuando las medidas son relativas es complicado que se adapte automáticamente.

**Ej:**

<div id="padre">

<div id="hijo">

Contenido aquí.

</div>

</div>

#padre {

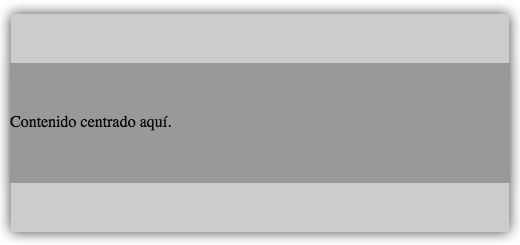
**padding**: 5% 0;

}

#hijo {

**padding**: 10% 0;

}



VI CONCLUSIONES

En conclusión existen diferentes métodos para lograr el objetivo que es alinear de forma vertical elementos block y centrarlos, la idea es escoger el mejor método que sea adaptable a cualquier pantalla o resolución, es decir que se auto ajuste o que sea responsivo.

REFERENCIAS

[1] <http://dariobf.com/centrar-verticalmente-con-css-6-formas-diferentes-de-lograrlo/>