



# Centrar horizontal y verticalmente en CSS un elemento

- 
- C.F.G.S. DAW
  - 6 horas semanales
  - Mes aprox. de impartición: Ene - Feb
  - iPasen - [cjaedia071@g.educaand.es](mailto:cjaedia071@g.educaand.es)

Carmelo José Jaén Díaz

# Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

# **OBJETIVO**



- **Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.**
- **Utilizar marcos y tablas para presentar la información de manera ordenada.**
- **Identificar nuevos elementos y etiquetas en HTML5.**
- **Reconocer las posibilidades de modificar etiquetas HTML.**
- **Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.**

# GLOSARIO

---



**Formularios.** Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

**Fuentes seguras.** Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

**Guías de estilo.** Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

**HTML.** Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

# GLOSARIO

---



**HTML5.** Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

**Legibilidad.** Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

**Marcos.** Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

**Tipografía.** Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa "modo o estilo en que está impreso un texto" o "clase de tipos de imprenta".

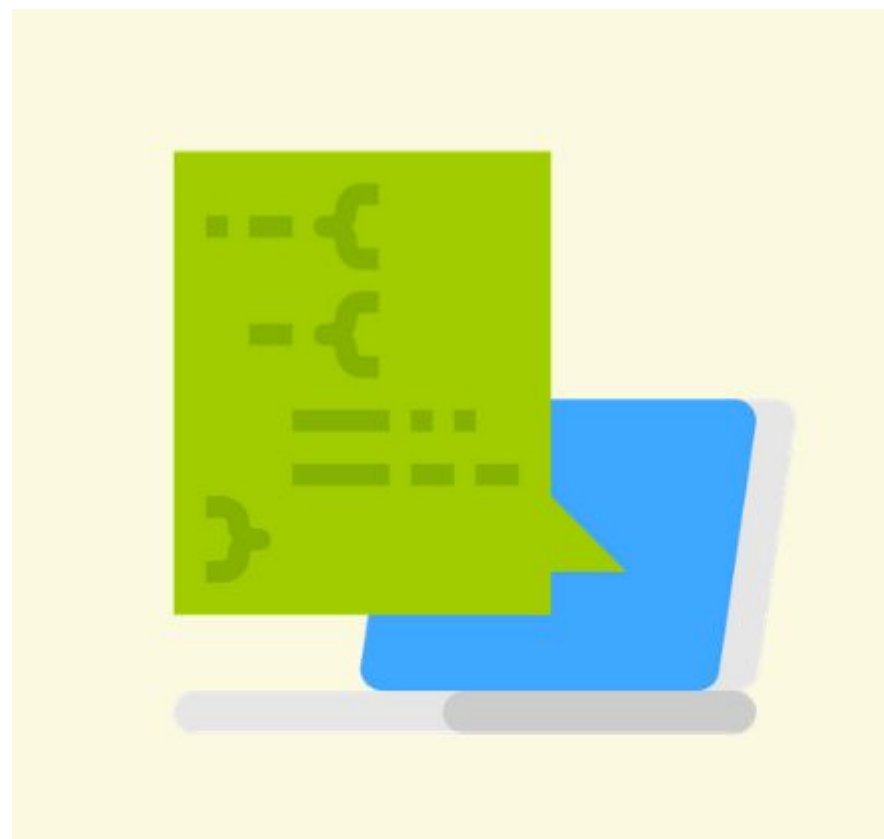
# INTRODUCCIÓN

---



Existen diversas formas de centrar horizontal y verticalmente un elemento en un contenedor mediante CSS. Cada método es adecuado para una situación concreta y, por ello, es necesario saber cómo funcionan todas las propiedades para reconocer fácilmente qué técnica utilizar en cada caso.

A continuación se describe cómo se puede realizar el centrado de algunos elementos mediante cajas flexibles, con la propiedad `transform`, con las propiedades `line-height` y `text-align` y con la propiedad `vertical-align`.



# CENTRAR VERTICAL Y HORIZONTALMENTE ELEMENTOS CON FLEXBOX

---



Este es el método más útil porque funciona para todos los elementos que se posicionen dentro de un contenedor. Por tanto, es válido para imágenes, textos, vídeos, etc. Sin embargo, para comprender cómo funciona es necesario tener conocimientos sobre el modelo de caja flexible o *flexbox*.

Para utilizar las propiedades de *flexbox* es necesario crear un contenedor y declararle la propiedad `display: flex`. Para centrar horizontal y verticalmente el contenido utilizaremos las propiedades `justify-content: center` y `align-items:center`.

# CENTRAR VERTICAL Y HORIZONTALMENTE ELEMENTOS CON FLEXBOX



Código HTML:

```
<div>
  <!-- Cualquier elemento incluido
dentro del div se centrará vertical y
horizontalmente con flexbox -->
  <h1>Centrado con flexbox</h1>
</div><br>
<div>
  
</div>
```

Código CSS:

```
div{
  height: 200px;
  background-color: #EEE;
  display: flex;
  justify-content: center;
  align-items: center;
}
```



# CENTRAR VERTICAL Y HORIZONTALMENTE ELEMENTOS CON FLEXBOX



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/KwPymaR>

**Centrado con flexbox**



# CENTRAR VERTICAL Y HORIZONTALMENTE UN TEXTO CON *line-height* Y *text-align*



Mediante las propiedades `line-height` y `text-align` podemos alinear únicamente textos. Así que este método sólo va a ser válido para el centrado de párrafos, encabezados, etc.

# CENTRAR VERTICAL Y HORIZONTALMENTE UN TEXTO CON *line-height* Y *text-align*



Código HTML:

```
<div>
  <h1>Centrar texto</h1>
</div>
```

Código CSS:

```
div {
  height: 200px;
  background-color: #EEE;
  text-align: center;
}
div h1 {
  line-height: 200px;}
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/EaYbmmz>



Centrar texto

# CENTRAR VERTICAL Y HORIZONTALMENTE UN ELEMENTO CON PROPIEDAD *transform*



Mediante este método se puede alinear tanto vertical como horizontalmente imágenes y contenedores. No es válido para la alineación de textos ya que no tiene en cuenta el tamaño de la letra. El centrado de elementos se realiza mediante posición absoluta y la propiedad `transform`.

# CENTRAR VERTICAL Y HORIZONTALMENTE UN ELEMENTO CON PROPIEDAD *transform*



Código HTML:

```
<div>
  
</div>
```

Código CSS:

```
div {
  height: 300px;
  background-color: #EEE;
  position: relative;
}
div img{
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  -webkit-transform: translate(-50%,
-50%);
}
```

# CENTRAR VERTICAL Y HORIZONTALMENTE UN ELEMENTO CON PROPIEDAD *transform*



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/JoPONrj>



# CENTRAR UN ELEMENTO UTILIZANDO POSICIÓN ABSOLUTA Y MARGEN NEGATIVO

---



Código HTML:

```
<div class="container">  
  <div class="centered-element">Contenido centrado</div>  
</div>
```

# CENTRAR UN ELEMENTO UTILIZANDO POSICIÓN ABSOLUTA Y MARGEN NEGATIVO

---



Código CSS:

```
.container {  
  position: relative;  
  width: 400px;  
  height: 300px;  
  background-color: lightgray;  
}
```

```
.centered-element {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  margin-top: -32px; /* Ajusta el  
valor según la mitad de la altura del  
elemento */  
  margin-left: -95px; /* Ajusta el  
valor según la mitad del ancho del  
elemento */  
  background-color: white;  
  padding: 20px;  
}
```



# CENTRAR UN ELEMENTO UTILIZANDO POSICIÓN ABSOLUTA Y MARGEN NEGATIVO

---



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/KwPymox>

A diagram illustrating the concept of absolute positioning and negative margins. It features a large light gray square representing a container. Centered within this container is a smaller white rectangle. The text "Contenido centrado" is centered inside the white rectangle, demonstrating how absolute positioning and negative margins can be used to center an element within its parent container.

Contenido centrado

# CENTRAR UN ELEMENTO CON GRID

---



Para centrar un elemento con grid se debe utilizar `display: grid` y `place-items: center` en el contenedor. De esta forma, el elemento se centrará tanto vertical como horizontalmente dentro del contenedor.

En el siguiente ejemplo, el contenedor tiene una altura fija de 300px y un borde para facilitar su visualización. El elemento a centrar está dentro del contenedor y tiene un fondo gris claro y un relleno de 20px.

# CENTRAR UN ELEMENTO CON GRID

---



Código HTML:

```
<div class="container">
  <div
class="centered-element">Elemento
centrado</div>
</div>
```

Código CSS:

```
.container {
  display: grid;
  place-items: center;
  height: 300px;
  border: 1px solid #ccc;
}

.centered-element {
  background-color: #f1f1f1;
  padding: 20px;
}
```

# CENTRAR UN ELEMENTO CON GRID

---



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/WbeXiqK>



# ALINEAR UN TEXTO VERTICALMENTE CON UNA IMAGEN CON *vertical-align*



Con esta técnica podemos alinear un texto a una imagen de forma vertical. Esta propiedad admite los valores *top*, *middle* y *bottom*, entre otros valores.

# ALINEAR UN TEXTO VERTICALMENTE CON UNA IMAGEN CON *vertical-align*



Código HTML:

```

```

Alinear texto

Código CSS:

```
img{  
    vertical-align: middle;  
}
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/emOeWxw>



Alinear texto

# ALINEAR VERTICALMENTE UN TEXTO EN UN *div* CON *vertical-align* Y EL MODELO DE TABLA



Vamos a alinear un texto en un *div* con la propiedad `vertical-align`. Hay que tener en cuenta que esta propiedad no se comporta de la misma manera con todos los elementos. En este caso, haremos uso de las reglas de estilo `display: inline-table` y `display: table-cell`.

# ALINEAR UN TEXTO VERTICALMENTE CON UNA IMAGEN CON *vertical-align*



Código HTML:

```
<div class="a">  
  <p>Texto dentro de un div</p>  
</div>
```

Código CSS:

```
div{  
  height: 200px; width: 200px;  
  background-color: #EEE;  
  display: inline-table;  
  width: 100%;  
}  
p{  
  text-align: center;  
  display: table-cell;  
  vertical-align: middle;  
}
```



# ALINEAR UN TEXTO VERTICALMENTE CON UNA IMAGEN CON *vertical-align*



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/KwPymjJ>



Texto dentro de un div

# ALINEAR HORIZONTALMENTE UNA IMAGEN CON *text-align*



Las imágenes se pueden alinear a la izquierda, derecha o centro insertándolas dentro de un *div* e indicando la propiedad `text-align` correspondiente.

# ALINEAR HORIZONTALMENTE UNA IMAGEN CON *text-align*



Código HTML:

```
<div class="a"></div>
<div class="b"></div>
<div class="c"></div>
```

Código CSS:

```
.a{text-align: left;}
.b{text-align: center;}
.c{text-align: right;}
```

En el ejemplo se utiliza la imagen:

<https://images.vexels.com/media/users/3/143166/isolated/lists/f76b5ed0f0de39d7e4c3bf59b8763c35-lo-go-de-la-flor-del-hipster.png>

# ALINEAR HORIZONTALMENTE UNA IMAGEN CON *text-align*



La salida del código anterior puede verse en la  
imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/ByBmZaV>



# ALINEAR VERTICALMENTE UN *checkbox* CON UN *label*



Cuando el tamaño del texto no coincide con el tamaño del *checkbox*, estos dos elementos no quedan alineados verticalmente. Para conseguir esta alineación utilizaremos `vertical-align:middle` y definiremos el tamaño del texto a las dos etiquetas, tal y como se muestra a continuación.

# ALINEAR VERTICALMENTE UN *checkbox* CON UN *label*



Código HTML:

```
<label  
for="texto">TEXT0</label><input  
type="checkbox" id="texto" />
```

Código CSS:

```
label, input{  
    font-size: 28px;  
    vertical-align: middle;  
}
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/RNbqPi>

TEXT0 ☐

TEXT0 ☒

# ALINEAR UN ELEMENTO SEGÚN EL ANCHO Y ALTO DE LA PANTALLA (*viewport*)



Para centrar elementos vertical y horizontalmente según el tamaño de la pantalla (viewport) en CSS, puedes utilizar las unidades de medida relativas **vh** y **vw** junto con las propiedades vistas anteriormente, por ejemplo con *Flexbox*. Esto garantiza que los elementos permanezcan centrados independientemente del tamaño de la pantalla del dispositivo.

# ALINEAR UN ELEMENTO SEGÚN EL ANCHO Y ALTO DE LA PANTALLA (*viewport*)



Código HTML:

```
<div>
  <h1>Centrado según tamaño viewport</h1>
</div>
```

Código CSS:

```
*{margin:0;padding:0;}
div{
  height: 100vh;
  background-color: #EEE;
  display:flex;
  justify-content: center;
  align-items: center;}
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/LEPOLNW>



Centrado según tamaño viewport