

# **HOJA DE ESTILOS: CSS**

## **Capítulo 12**

*Lenguajes de Marcas y Sistemas de Gestión de Información*

*Curso 2019/2020*

# + Capítulo 12: LAYOUT

1. Introducción
2. Centrar una Página Horizontalmente
3. Centrar una Página Verticalmente
4. Estructura o Layout
5. Alturas/Anchuras Máximas y Mínimas

## + 12.1. INTRODUCCIÓN

El diseño de las páginas web habituales se divide en **bloques**: cabecera, menú, contenidos y pie de página. Visualmente, los bloques se disponen en varias **filas y columnas**.

Por este motivo, hace varios años la estructura de las páginas HTML se definía mediante **tablas**.

El desarrollo de CSS ha permitido que se puedan realizar los mismos diseños sin utilizar tablas HTML.

# + 12.1. INTRODUCCIÓN

## VENTAJAS

- **Mantenimiento:** una página diseñada exclusivamente con CSS es mucho más fácil de mantener que una página diseñada con tablas. Cambiar el **aspecto** de una página creada con CSS es tan fácil como modificar unas pocas reglas en las hojas de estilos. Sin embargo, realizar la misma modificación en una página creada con tablas supone un **esfuerzo muy superior** y es más probable cometer **errores**.
- **Accesibilidad:** las páginas creadas con CSS son más accesibles que las páginas diseñadas con tablas. No obstante, diseñar una página web exclusivamente con CSS no garantiza que la página sea accesible.

# + 12.1. INTRODUCCIÓN

## VENTAJAS

- **Velocidad de Carga:** el código HTML de una página diseñada con tablas es mucho mayor que el código de la misma página diseñada exclusivamente con CSS, por lo que tarda más tiempo en descargarse.
- **Semántica:** aunque resulta obvio, las tablas HTML sólo se deben utilizar para mostrar datos cuya información sólo se entiende en forma de filas y columnas.



## 12.2. CENTRAR UNA PÁGINA HORIZONTALMENTE

A medida que aumenta el tamaño y la resolución de las pantallas de ordenador, se hace más difícil diseñar páginas que se adapten al tamaño de la ventana del navegador.

El principal reto se presenta con resoluciones **superiores a 1024 x 768 píxel**. → Se opta por diseños con una anchura fija limitada a un valor aceptable para mantener la legibilidad del texto.

Los navegadores alinean por defecto las páginas web **a la izquierda** de la ventana. Cuando la resolución de la pantalla es muy grande, la mayoría de páginas de anchura fija alineadas a la izquierda parecen muy estrechas y provocan una sensación de vacío.



## 12.2. CENTRAR UNA PÁGINA HORIZONTALMENTE

La solución más sencilla para evitar los grandes espacios en blanco consiste en crear **páginas con una anchura fija adecuada** y centrar la página horizontalmente respecto de la ventana del navegador.





## 12.2. CENTRAR UNA PÁGINA HORIZONTALMENTE

```
#contenedor {  
    width: 300px;  
    margin: 0 auto;  
}
```

```
<body>
```

```
    <div id="contenedor">
```

```
        <h1>Lorem ipsum dolor sit amet</h1>
```

```
        ...
```

```
    </div>
```

```
</body>
```



```
#contenedor {
```

```
    width: 70%;
```

```
    margin: 0 auto;
```

```
}
```





## 12.3. CENTRAR UNA PÁGINA VERTICALMENTE

El objetivo es que sus contenidos aparezcan en el centro de la ventana del navegador y por tanto, que sus márgenes verticales se adapten de **forma dinámica** en función del tamaño de la ventana del navegador.

Aunque centrar una página web horizontalmente es muy sencillo, centrarla verticalmente es mucho más **complicado**.

Afortunadamente, **no es muy común** que una página web aparezca centrada de forma vertical. El motivo es que la mayoría de páginas web son más altas que la ventana del navegador, por lo que no es posible centrarlas verticalmente.



## 12.3. CENTRAR UNA PÁGINA VERTICALMENTE

Mismo razonamiento que el planteado para centrar la página horizontalmente, se podrían utilizar las siguientes reglas CSS:

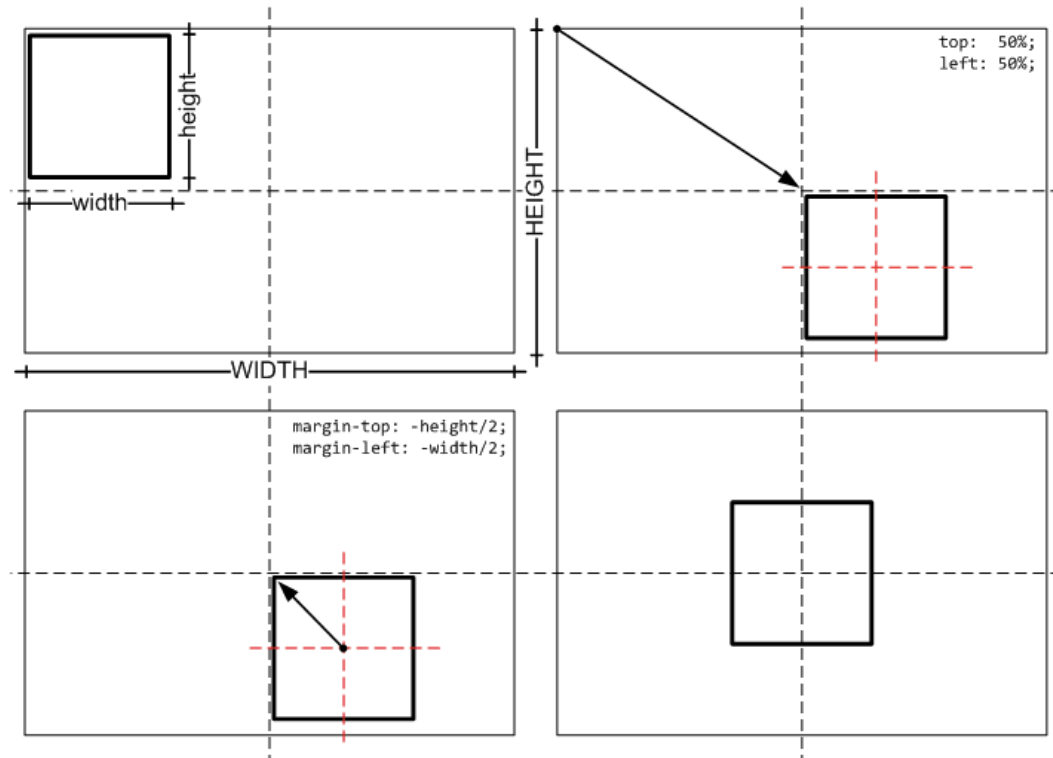
```
#contenedor {  
  width: 300px;  
  height: 250px;  
  margin: auto;  
}  
  
<body>  
  <div id="contenedor">  
    <h1>Lorem ipsum dolor sit amet</h1>  
    ...  
  </div>  
</body>
```

Valor `auto` se puede utilizar para que los márgenes laterales se adapten dinámicamente. Desafortunadamente, la propiedad `margin: auto` no funciona tal y como se espera para los márgenes verticales y la página no se muestra centrada.



## 12.3. CENTRAR UNA PÁGINA VERTICALMENTE

La solución correcta para centrar verticalmente una página web se basa en el posicionamiento absoluto e implica realizar un cálculo matemático sencillo.

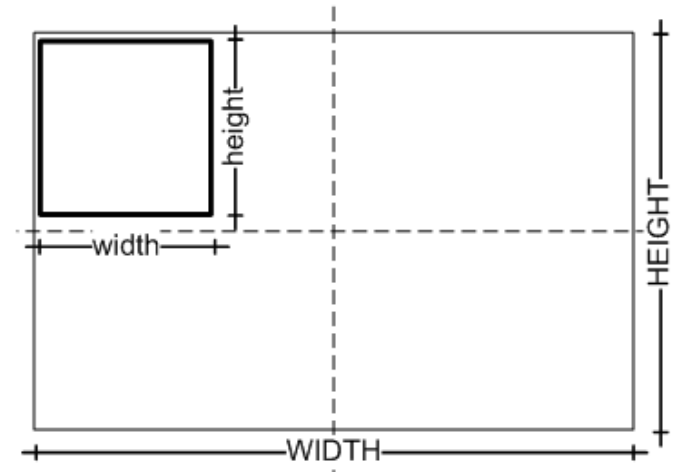




## 12.3. CENTRAR UNA PÁGINA VERTICALMENTE

1° se asigna una altura y una anchura al elemento que encierra todos los contenidos de la página. En el siguiente ejemplo, se supone que tanto la anchura como la altura de la página es igual a 500px:

```
#contenedor {  
    width: 500px;  
    height: 500px;  
}  
  
<body>  
    <div id="contenedor">  
        <h1>Lorem ipsum dolor sit amet</h1>  
        ...  
    </div>  
</body>
```

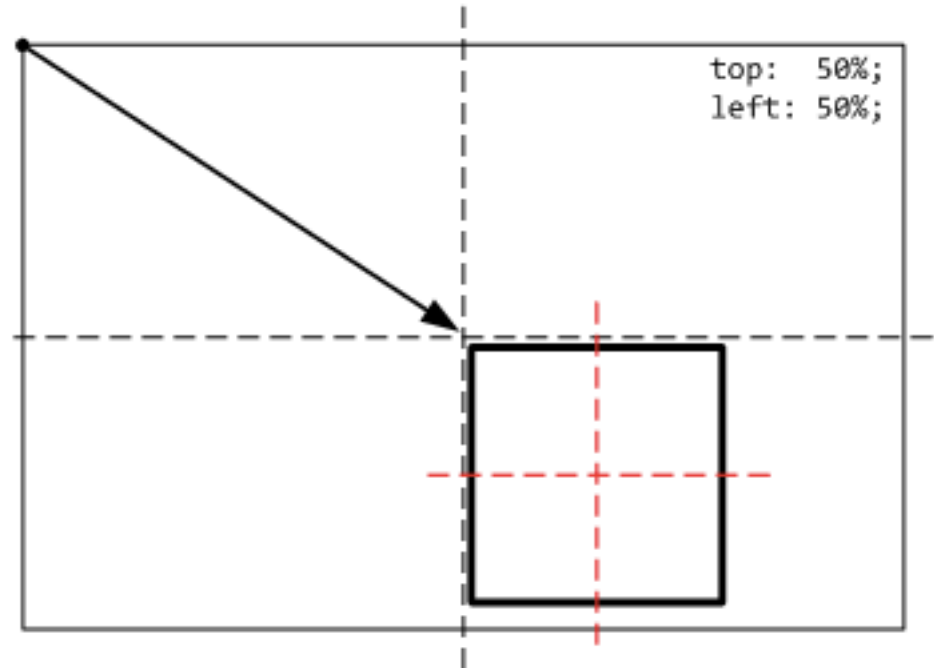




## 12.3. CENTRAR UNA PÁGINA VERTICALMENTE

2º Se posiciona de forma **absoluta** el elemento contenedor y se asigna un valor de 50% tanto a la propiedad `top` como a la propiedad `left`. El resultado es que la esquina superior izquierda del elemento contenedor se posiciona en el centro de la ventana del navegador:

```
#contenedor {  
    width: 500px;  
    height: 500px;  
  
    position: absolute;  
    top: 50%;  
    left: 50%;  
}
```





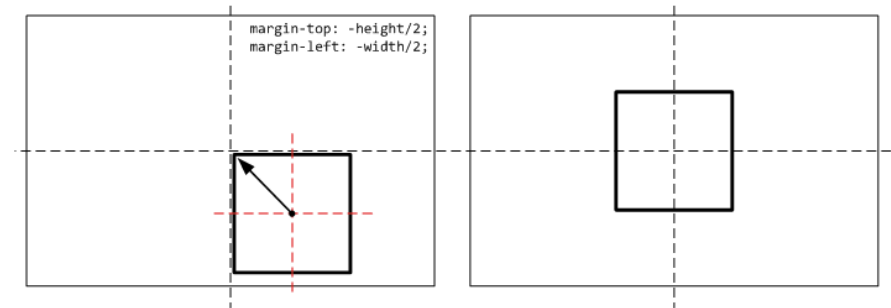
## 12.3. CENTRAR UNA PÁGINA VERTICALMENTE

3º La página se debe mostrar en el **centro** de la ventana del navegador. Para determinar el desplazamiento necesario, se realiza un cálculo matemático sencillo → El punto central de la página debe desplazarse hasta el centro de la ventana del navegador.

La página web debe moverse hacia arriba una cantidad igual a la mitad de su altura y debe desplazarse hacia la izquierda una cantidad equivalente a la mitad de su anchura. Utilizaremos las propiedades `margin-top` y `margin-left`:

```
#contenedor {
  width: 500px;
  height: 500px;

  position: absolute;
  top: 50%;
  left: 50%;
```



```
  margin-top: -250px; /* height/2 = 500px / 2 */
  margin-left: -250px; /* width/2 = 500px / 2 */
```

```
}
```

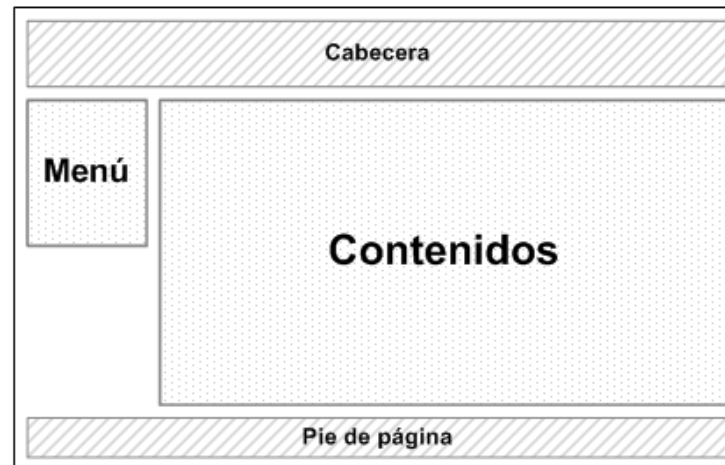


## 12.4. ESTRUCTURA O LAYOUT

### **DISEÑO A 2 COLUMNAS + CABECERA + PIE DE PÁGINA**

El objetivo es definir una estructura de página con cabecera y pie, un menú lateral de navegación y una zona de contenidos.

La solución CSS se basa en el uso de la propiedad `float` para los elementos posicionados como el menú y los contenidos y el uso de la propiedad `clear` en el pie de página para evitar los solapamientos ocasionados por los elementos posicionados con `float`.



## 12.4. ESTRUCTURA O LAYOUT

Los códigos HTML y CSS son los siguientes:

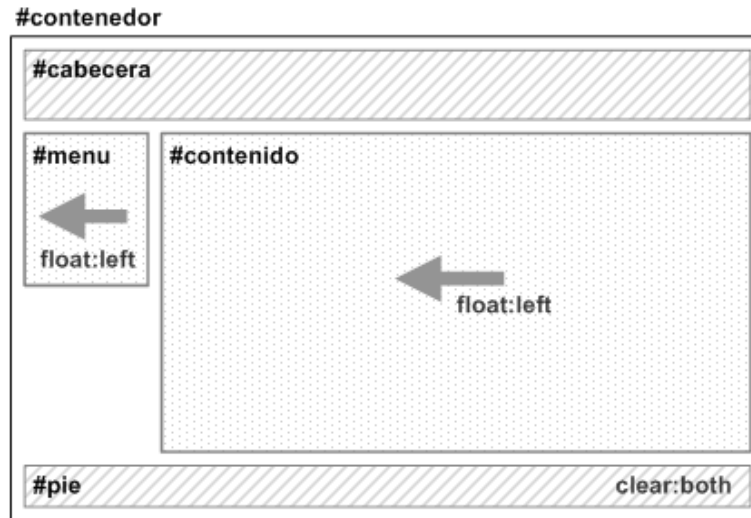
```
#contenedor {
    width: 700px;
}

#cabecera {
}

#menu {
    float: left;
    width: 150px;
}

#contenido {
    float: left;
    width: 550px;
}

#pie {
    clear: both;
}
```



```
<body>
<div id="contenedor">
    <div id="cabecera">
    </div>

    <div id="menu">
    </div>

    <div id="contenido">
    </div>

    <div id="pie">
    </div>
</div>
</body>
```





## 12.5. ANCHURAS/ALTURAS MÁXIMAS Y MÍNIMAS

Cuando se diseña la estructura de una página web, se debe tomar la decisión de optar por un diseño de **anchura** fija o un diseño cuya anchura se adapta a la anchura de la ventana del navegador.

Sin embargo, la mayoría de las veces sería conveniente una solución intermedia: que la anchura de la página sea variable y se adapte a la anchura de la ventana del navegador, pero respetando ciertos límites.

## 12.5. ANCHURAS/ALTURAS MÁXIMAS Y MÍNIMAS

CSS define cuatro propiedades que permiten **limitar** la anchura y altura mínima y máxima de cualquier elemento de la página.

<b>Propiedad</b>	<i>max-width</i>
<b>Valores</b>	unidad de medida   porcentaje   none   inherit
<b>Se aplica a</b>	Todos los elementos salvo filas y grupos de filas de tablas
<b>Valor Inicial</b>	none
<b>Descripción</b>	Permite definir la anchura máxima de un elemento



## 12.5. ANCHURAS/ALTURAS MÁXIMAS Y MÍNIMAS

CSS define cuatro propiedades que permiten **limitar** la anchura y altura mínima y máxima de cualquier elemento de la página.

<b>Propiedad</b>	<i>min-width</i>
<b>Valores</b>	unidad de medida   porcentaje   none   inherit
<b>Se aplica a</b>	Todos los elementos salvo filas y grupos de filas de tablas
<b>Valor Inicial</b>	none
<b>Descripción</b>	Permite definir la anchura mínima de un elemento

## 12.5. ANCHURAS/ALTURAS MÁXIMAS Y MÍNIMAS

CSS define cuatro propiedades que permiten **limitar** la anchura y altura mínima y máxima de cualquier elemento de la página.

<b>Propiedad</b>	<i>max-height</i>
<b>Valores</b>	unidad de medida   porcentaje   none   inherit
<b>Se aplica a</b>	Todos los elementos salvo columnas y grupos de columnas de tablas
<b>Valor Inicial</b>	none
<b>Descripción</b>	Permite definir la altura máxima de un elemento

## 12.5. ANCHURAS/ALTURAS MÁXIMAS Y MÍNIMAS

CSS define cuatro propiedades que permiten **limitar** la anchura y altura mínima y máxima de cualquier elemento de la página.

<b>Propiedad</b>	<i>min-height</i>
<b>Valores</b>	unidad de medida   porcentaje   none   inherit
<b>Se aplica a</b>	Todos los elementos salvo columnas y grupos de columnas de tablas
<b>Valor Inicial</b>	none
<b>Descripción</b>	Permite definir la altura mínima de un elemento



## 12.5. ANCHURAS/ALTURAS MÁXIMAS Y MÍNIMAS

Para conseguir un diseño de anchura variable pero controlada, se podrían utilizar reglas CSS como la siguiente:

```
#contenedor {  
    min-width: 500px;  
    max-width: 900px;  
}
```

Las propiedades que definen la altura y anchura máxima y mínima se pueden aplicar a cualquier elemento, aunque **solamente suelen utilizarse para estructurar la página**. En general, las propiedades más utilizadas son max-width y min-width, ya que no es muy habitual definir alturas máximas y mínimas.