

## E. CSS de posicionamiento

Los navegadores crean una caja para representar cada elemento de la página HTML y la posicionan de forma automática en ella, pero desde CSS se puede modificar dicha posición.

Existen tres formas de posicionar una caja:

- Posicionamiento normal o estático. Es el utilizado por defecto en los navegadores. Los elementos se muestran en la pantalla en el mismo orden en que se encuentran en el código fuente. Hay que conocer si el elemento es de bloque o en línea, sus propiedades `width`, `height` y su contenido.
- Posicionamiento flotante. Los elementos se muestran en la pantalla en el mismo orden en que se encuentran en el código fuente, pero pueden estar desplazados a derecha o izquierda de la posición en la que inicialmente estaba. Utiliza la propiedad `float`.
- Posicionamiento absoluto. Se utiliza para establecer de forma absoluta o exacta la posición en la que se muestra la caja de un elemento. Los elementos se pueden mostrar en pantalla en orden diferente al del código fuente. Utiliza la propiedad `position`.

### E.1 Posicionamiento absoluto

El posicionamiento absoluto de un elemento en la caja se controla con la propiedad `position`, que puede tomar los valores:

- **relative**: posicionamiento absoluto relativo. Es como el posicionamiento normal, pero luego la caja se desplaza y se controla con las propiedades `top`, `right`, `bottom` y `left`.
- **fixed**: el posicionamiento absoluto es fijo cuando la caja es posicionada de forma fija, es decir, inamovible dentro de la ventana del navegador. En ese caso, las cajas no modifican su posición, aunque el usuario mueva la página que muestra el navegador.
- **absolute**: el posicionamiento absoluto es absoluto cuando la caja se sitúa en la posición exacta que se le da. En ese caso, las cajas sí que pueden modificar su posición si el usuario mueve la página que muestra el navegador.

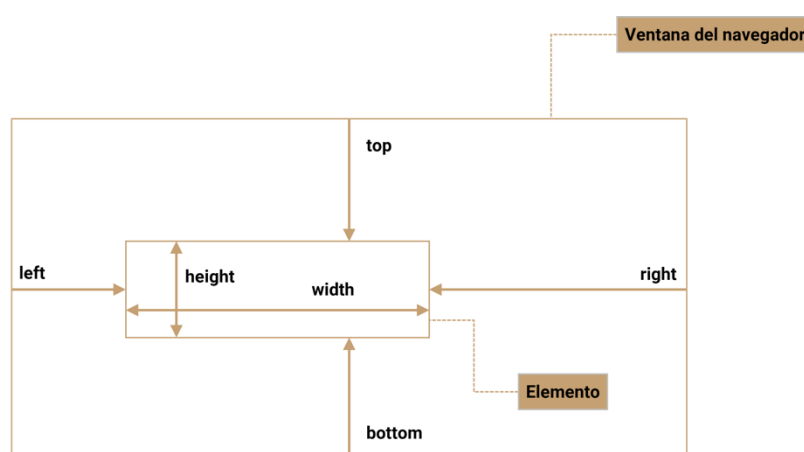


Fig. 2.36. Análisis de una caja.

En el posicionamiento absoluto, en cualquiera de las tres modalidades, las cajas se pueden superponer. Con la propiedad `z-index` se puede controlar qué caja va encima o debajo.

El valor de la propiedad `z-index` es un número entero ( $>0$ ,  $<0$ ,  $0$ ). Valor cero equivale a no tener `z-index`. Cajas con valores mayores de `z-index` se ponen encima de las que tengan valores menores.

## E.2 Otras propiedades

La siguiente tabla describe un conjunto de propiedades de las cajas.

Propiedad	Descripción y valores
<b>height</b>	Permite establecer la altura de un elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<b>width</b>	Permite establecer la anchura de un elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<b>visibility</b>	Indica si el elemento sobre el que actúa será visible o no. Valores posibles: <ul style="list-style-type: none"> <li>hidden (esconde un elemento, pero deja vacío el espacio donde debería aparecer)</li> <li>collapse (esconde filas o columnas de una tabla).</li> </ul>
<b>left</b>	Indica la posición del lado izquierdo del elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<b>top</b>	Indica la posición del lado superior del elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<b>overflow</b>	Indica si el elemento será visible o no en caso de superar los límites del contenedor. Los valores que pueden tomar son: <ul style="list-style-type: none"> <li>visible (se verá aunque sobrepase).</li> <li>hidden (se recorta si no cabe).</li> <li>scroll (se recorta pero muestra barras de desplazamiento).</li> <li>auto (depende del navegador que aparezcan barras o no).</li> </ul>

Tabla 2.19. Otras propiedades de las cajas.



### EJEMPLO 7

En el archivo `UN02_AP05_ejemplo_7.html` encontrarás un ejemplo de propiedades de posicionamiento utilizando CSS interno.

#### Ejemplo de propiedades CSS de posicionamiento



Fig. 2.17. Ejecución del ejemplo 7.

## F. Definición y uso de clases

Cuando se quiere aplicar estilos a un solo elemento de la página HTML se puede utilizar el atributo `class` de HTML sobre ese elemento y así enlazar directamente con la regla CSS que se le debe aplicar.

La clase define un estilo que, en principio, no está asociado a ninguna etiqueta HTML, pero que es susceptible de asociar a etiquetas concretas.

El atributo `class` se define mediante una lista de elementos separados por espacio, y uno de esos elementos debe coincidir exactamente con el nombre de clase dado en el selector.

En la declaración del selector CSS se incluye el nombre de clase precedido de un punto (.). Así, el selector de identificador aplica el estilo al elemento HTML cuyo identificador sea ese nombre.

Recordar que los selectores se utilizan para seleccionar los elementos a los que se van a aplicar las propiedades declaradas en el código CSS.

Por ejemplo, se define así una clase para el color rojo:

```
.clase_rojo{color:red}
```

Si se quiere que el título "h1" se muestre en rojo, se asocia en el archivo HTML el elemento `h1` a dicha clase de la forma:

```
<h1 class="clase_rojo">Cabecera h1 en rojo</h1>
```

Si solo se quiere utilizar esta clase con el elemento HTML de párrafo, en la declaración de la clase hay que añadir el selector `p`, y queda:

```
p.clase_rojo{color:red}
```



### IMPORTANTE

Las clases se pueden utilizar varias veces dentro del mismo documento. Si solo se utiliza una vez se puede emplear el atributo `id`.



### EJEMPLO 8

En el archivo `UN02_AP05_ejemplo_8.html` encontrarás un ejemplo del uso de clases utilizando CSS interno.

## G. Frameworks para CSS

Los frameworks CSS son herramientas de diseño web que facilitan la tarea de crear y editar hojas de estilo en cascada. Facilitan el trabajo del desarrollador/diseñador y le proporcionan una colección de hojas de estilo que se pueden adaptar a sus necesidades.

De ellas, las más importantes o conocidas son las siguientes:

- [Bootstrap](#).
- [Bulma](#).
- [Foundation](#).
- [Materialize CSS](#).

Muchas de ellas utilizan un diseño responsive, es decir, adaptan el diseño de la página al dispositivo que la visualiza.