

# Posicionar y flotar

El posicionamiento y el flotado son técnicas fundamentales en CSS que permiten controlar la ubicación y el diseño de elementos en una página web.

## 1. Posicionamiento

### 1.1. Posicionamiento Estático

El posicionamiento estático es el comportamiento predeterminado de un elemento. No se ve afectado por las propiedades de posicionamiento y se coloca según el flujo normal del documento.

```
.elemento {  
position: static;  
}
```

### 1.2. Posicionamiento Relativo

El posicionamiento relativo permite mover un elemento desde su posición normal. Se utiliza con las propiedades top, right, bottom y left para ajustar su posición.

```
.elemento {  
position: relative;  
top: 10px;  
left: 20px;  
}
```

Cuando aplicas position: relative; a un elemento en CSS, se creará un "contexto de posicionamiento" para ese elemento. Esto significa que cualquier elemento hijo que tenga una posición absoluta (position: absolute;) se posicionará en relación con el elemento padre que tenga position: relative;. Si no hay un elemento padre con posicionamiento relativo, el elemento se posicionará en relación con el cuerpo del documento (o el elemento <html>).

### Ejemplo:

```
.contenedor {  
position: relative;  
}  
.elemento-absoluto {  
position: absolute;  
top: 10px;  
left: 20px;  
}
```

En este ejemplo, tenemos un contenedor con la clase .contenedor al que se le aplica position: relative;. Dentro de este contenedor, hay un elemento con la clase .elemento-absoluto al que se le aplica position: absolute;.

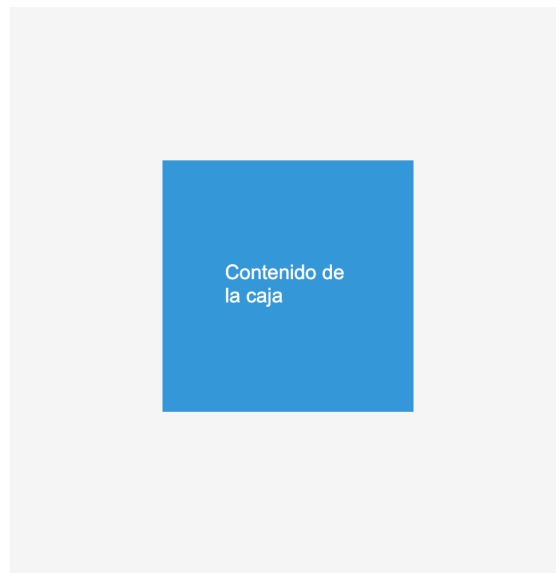
El elemento con la clase `.elemento-absoluto` se posiciona en las coordenadas especificadas (`top: 10px;` y `left: 20px;`), pero estas coordenadas se referirán al contenedor con `position: relative;` más cercano.

## Ejemplo:

Crear una caja que se desplace 20 píxeles hacia abajo desde su posición original. Utilizaremos posicionamiento relativo. El resultado será una caja con un párrafo dentro, donde el texto está posicionado relativamente dentro de la caja, y desplazado 20 píxeles hacia abajo desde su posición original.

El resultado será una caja con un párrafo dentro, donde el texto está posicionado relativamente dentro de la caja, y desplazado 20 píxeles hacia abajo desde su posición original.

```
<!DOCTYPE html> <html lang="es">
<head>
<meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Posicionamiento Relativo - Ejercicio</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="caja">
<p>Contenido de la caja</p>
</div>
</body>
</html>
```



```
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh; margin: 0;
}
```

```

.caja {
    width: 200px;
    height: 200px;
    background-color: #3498db;
    position: relative; /* Aplicando posicionamiento relativo */
}

p {
    position: absolute; /* Aplicando posicionamiento absoluto para el texto */
    top: 20px; /* Moviendo el texto 20px hacia abajo */
    left: 50%; /* Centrando horizontalmente */
    transform: translateX(-50%); /* Corrigiendo la posición del texto */
    color: #ffffff;
    margin: 0;
}

```

- Se crea un contenedor div con la clase caja que será nuestra caja. Le asignamos una anchura y altura, y le damos un color de fondo.
- Se aplica `position: relative;` a la caja. Esto establece el contexto de posicionamiento para los elementos hijos posicionados de manera absoluta dentro de ella.
- Se añade un párrafo (p) dentro de la caja con el texto "Contenido de la caja".
- Se aplica `position: absolute;` al párrafo para posicionarlo de manera absoluta con respecto a su elemento padre (la caja).
- Se utiliza `top: 20px;` para mover el párrafo 20 píxeles hacia abajo desde su posición original dentro de la caja.
- Para centrar horizontalmente el texto, se utiliza `left: 50%;` y `transform: translateX(-50%);`.

### 1.3. Posicionamiento Absoluto

El posicionamiento absoluto posiciona un elemento con respecto a su primer ancestro posicionado o al elemento de nivel superior (body).

```

.elemento {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}

```

## 1.4. Posicionamiento Fijo

El posicionamiento fijo mantiene un elemento fijo en la ventana del navegador, incluso cuando se desplaza la página.

```
.elemento {  
position: fixed;  
top: 0;  
right: 0;  
}
```

El posicionamiento fijo es útil cuando se pretende que un elemento se mantenga visible en la misma posición incluso cuando el usuario desplaza la página. Esto es comúnmente usado para elementos como barras de navegación, botones de acción rápida o notificaciones que deben permanecer accesibles en todo momento, o para el pie de página. Es especialmente útil en sitios web con contenido extenso donde los usuarios necesitan navegar o realizar acciones sin importar en qué parte de la página se encuentren.

En el ejemplo, el elemento con la clase `.fijo` se mantiene fijo en la esquina superior derecha de la ventana del navegador, independientemente de cómo se desplace el contenido en la página.

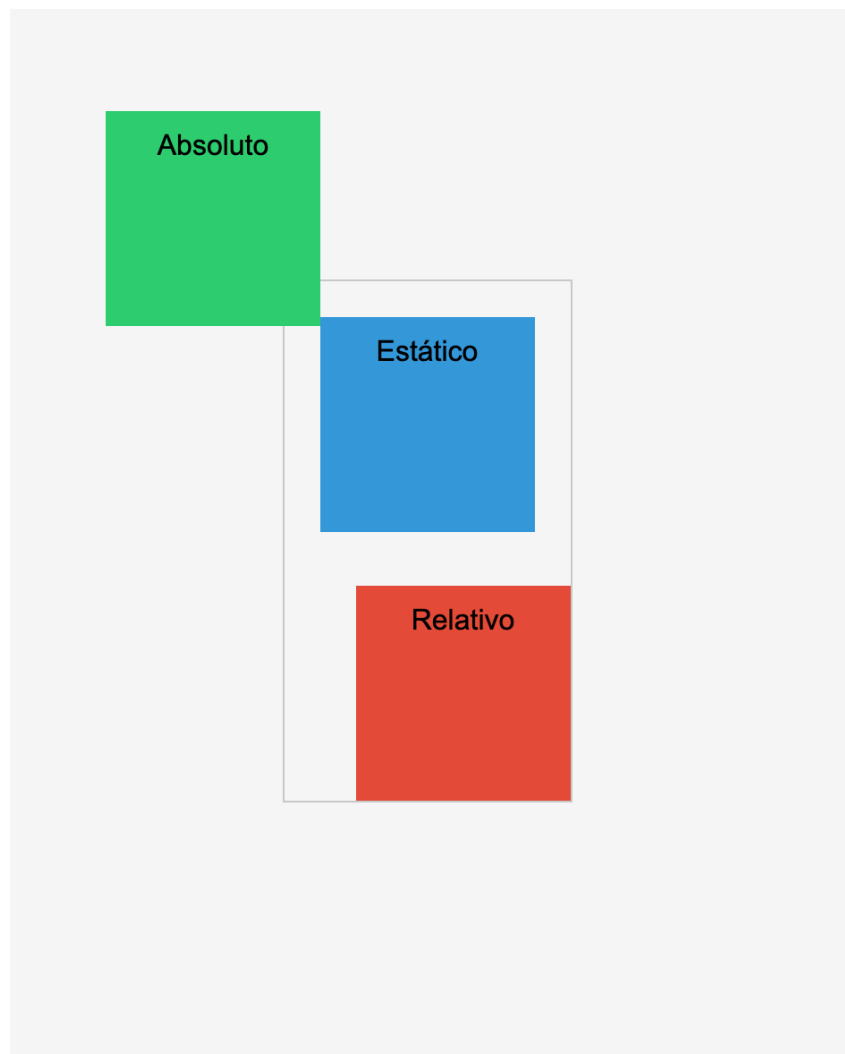
```
<!DOCTYPE html> <html lang="es">  
<head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title>Ejemplo de Posicionamiento Fijo en CSS</title>  
<link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  <div class="contenedor">  
    <div class="elemento fijo">Posicionamiento Fijo</div>  
    <div class="contenido">  
      <p>Contenido de la página.</p>  
      <p>Scroll para ver el efecto del posicionamiento fijo.</p>  
    </div>  
  </div>  
</body>  
</html>
```

```
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
}  
  
.contenedor {  
  padding: 20px;  
}
```

```
.elemento {  
    width: 200px;  
    padding: 10px;  
    text-align: center;  
    background-color: #3498db;  
    color: #ffffff;  
}  
  
.fijo {  
    position: fixed;  
    top: 20px;  
    right: 20px;  
}
```

Hemos utilizado `position: fixed;` para el elemento `.fijo`, lo que significa que se mantendrá en la misma posición incluso cuando se desplace la página. `top: 20px;` y `right: 20px;` se utilizan para fijar el elemento a 20 píxeles desde la parte superior y 20 píxeles desde la derecha del borde del viewport.

**EJERCICIO PARA HACER EN CLASE:** Utilizando los tipos de posicionamiento descritos reproduce la siguiente imagen.



## Descripción de la imagen:

- **Estático:** Posición estática (posición predeterminada). No se aplican las propiedades top, right, bottom o left.
- **Relativo:** Este elemento tiene una posición relativa, lo que significa que podemos moverlo respecto a su posición normal. Hemos aplicado top: 20px; y left: 20px;, lo que desplaza el elemento 20 píxeles hacia abajo y 20 píxeles hacia la derecha desde su posición normal.
- **Absoluto:** Este elemento tiene una posición absoluta, lo que significa que se posiciona con respecto al primer elemento padre que tenga una posición no estática. Hemos aplicado top: 50px; y left: 50px;, lo que lo coloca 50 píxeles desde la parte superior y 50 píxeles desde la izquierda del contenedor.

## 2. Flotar

### 2.1. Flotar a la Izquierda y Derecha

El flotado permite que un elemento se coloque a la izquierda o derecha de su contenedor.

```
.elemento-izquierda {  
float: left;  
}
```

```
.elemento-derecha {  
float: right;  
}
```

#### Ejemplo:



```
<!DOCTYPE html>  
<html lang="es">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Ejemplo de Flotado en CSS</title>  
<style>  
  .contenedor {  
    width: 600px;  
    margin: 0 auto;  
    border: 1px solid #ccc;  
    overflow: auto; /* Aplicando clearfix */
```

```
}
```

```
.elemento {
```

```
    width: 200px;
```

```
    height: 100px;
```

```
    margin: 10px;
```

```
    text-align: center;
```

```
    padding: 10px;
```

```
}
```

```
.izquierda {
```

```
    background-color: #3498db;
```

```
    float: left;
```

```
}
```

```
.derecha {
```

```
    background-color: #e74c3c;
```

```
    float: right;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <div class="contenedor">
```

```
        <div class="elemento izquierda">Izquierda</div>
```

```
        <div class="elemento derecha">Derecha</div>
```

```
    </div>
```

```
</body>
```

```
</html>
```