

## Posicionamiento en CSS

En CSS, la propiedad `position` se utiliza para especificar cómo un elemento debe posicionarse dentro del flujo del documento. Dependiendo del valor que se le asigne, cambia el comportamiento de los elementos en relación con sus contenedores o la ventana del navegador. Antes de comenzar tenemos que tratar unos conceptos previos:

- **Flujo de renderizado:** Por norma general los elementos se dibujan de izquierda a derecha y de arriba abajo. El punto 0,0 de los elementos, por norma general, es la esquina superior izquierda.
- **Espacio reservado:** Es el espacio que tiene un elemento asignado en el navegador.
- **Elemento posicionado:** Esto significa que el elemento tiene la propiedad `position` con un valor distinto de "static", que es el valor que tienen los elementos por defecto.
- **Stackin context:** Contexto de apilamiento. Es el orden en el que se apilarán las cajas que se superponen dentro del mismo contenedor.
- **Nota:** Al posicionar un elemento se habilitan 5 propiedades que podemos utilizar para mover los elementos en los 3 ejes. Estas son `top`, `left`, `bottom`, `right` y `z-index`. Teniendo `top` y `left` prioridad sobre `bottom` y `right`.

### 1. `position: static`

- **Descripción:** Este es el valor predeterminado para todos los elementos HTML. Los elementos con `position: static` se colocan en el flujo normal del documento, es decir, en el orden que les corresponde en el DOM. No se les puede aplicar propiedades como `top`, `right`, `bottom` o `left`.
- **Características:**
  - Los elementos no se pueden mover manualmente usando coordenadas.
  - No afecta a otros elementos del flujo del documento.
  - No responde al `z-index`.

```
.elemento {  
  position: static; /* Comportamiento por defecto */  
}
```

### 2. `position: relative`

- **Descripción:** Un elemento con `position: relative` se coloca en su posición normal en el flujo del documento, pero luego se puede ajustar mediante las propiedades `top`, `right`, `bottom`, `left` en relación con su posición original.
- **Características:**
  - Se mantiene en el flujo del documento (es decir, ocupa espacio), pero puedes moverlo con coordenadas relativas a su posición original.
  - Afecta al `z-index`, permitiendo que el elemento se superponga a otros elementos que también tengan un `position` diferente de `static`.

```
.elemento {  
  position: relative;
```

```
top: 10px; /* Se mueve 10px hacia abajo desde su posición original */
left: 20px; /* Se mueve 20px a la derecha desde su posición original */
}
```

### 3. position: absolute

- **Descripción:** Un elemento con `position: absolute` se saca del flujo normal del documento y se posiciona en relación con su contenedor posicionado más cercano (es decir, un ancestro con `position: relative, absolute` o `fixed`). Si no hay ningún ancestro posicionado, se posiciona en relación con el viewport (ventana del navegador).
- **Características:**
  - El elemento no ocupa espacio en el flujo del documento.
  - Las propiedades `top`, `right`, `bottom`, `left` se utilizan para posicionarlo respecto a su contenedor posicionado.
  - Afecta al `z-index` y puede superponerse a otros elementos.

```
.contenedor {
  position: relative; /* Necesario para que el elemento hijo absoluto se posicione
en relación a este */
}

.elemento-absoluto {
  position: absolute;
  top: 0; /* Se posiciona en la parte superior del contenedor */
  left: 0; /* Se posiciona a la izquierda del contenedor */
}
```

### 4. position: fixed

- **Descripción:** Un elemento con `position: fixed` se saca del flujo del documento y se posiciona en relación con el viewport (ventana del navegador). A diferencia de `absolute`, este valor no está relacionado con ningún otro contenedor, solo con la ventana del navegador.
- **Características:**
  - El elemento permanece fijo en la ventana del navegador incluso cuando la página se desplaza (scroll).
  - No ocupa espacio en el flujo del documento.
  - Afecta al `z-index`.

```
.elemento-fijo {
  position: fixed;
  top: 0; /* Se fija en la parte superior de la ventana */
  left: 0; /* Se fija en la parte izquierda de la ventana */
  width: 100%;
  height: 50px;
}
```

## 5. position: sticky

- **Descripción:** Un elemento con `position: sticky` es híbrido entre `relative` y `fixed`. Inicialmente, el elemento se comporta como `relative`, pero cuando se desplaza más allá de un umbral en la página (definido por `top`, `right`, `bottom`, `left`), cambia a `fixed`, quedándose fijo en esa posición mientras se sigue desplazando.
- **Características:**
  - El elemento ocupa espacio en el flujo del documento hasta que se activa el comportamiento "fijo".
  - El `z-index` puede aplicarse una vez que el elemento está fijo.

```
.elemento-sticky {  
  position: sticky;  
  top: 20px; /* El elemento se mantiene a 20px de la parte superior cuando el  
scroll lo alcanza */  
}
```

## 6. z-index

- **Descripción:** `z-index` controla el orden de apilamiento de los elementos posicionados. Un número mayor indica que el elemento debe aparecer por encima de los elementos con un valor más bajo de `z-index`. Sólo afecta a elementos con `position` distinta a `static` (es decir, `relative`, `absolute`, `fixed`, o `sticky`).
- **Características:**
  - Los valores pueden ser positivos, negativos o `auto`.
  - Un mayor valor de `z-index` coloca el elemento encima de otros.

```
.elemento-superior {  
  position: absolute;  
  z-index: 10; /* Se superpone a los elementos con z-index menores */  
}  
  
.elemento-inferior {  
  position: absolute;  
  z-index: 5; /* Se coloca debajo del elemento-superior */  
}
```

### Resumen de comportamiento:

- **Static:** flujo normal del documento, no se puede mover ni afecta al `z-index`.
- **Relative:** se posiciona respecto a su ubicación original.
- **Absolute:** se posiciona respecto a su contenedor posicionado más cercano.
- **Fixed:** se posiciona respecto al viewport.
- **Sticky:** se comporta como `relative` hasta que se cumple una condición de desplazamiento, luego se vuelve `fixed`.

- **Z-index:** controla la superposición de elementos posicionados.

Estas propiedades son fundamentales para controlar la disposición y el comportamiento visual de los elementos en una página web.

[Ir a mediaqueries](#)