

CSS POSITION

Position: Es la propiedad que nos va a permitir posicionar elementos en nuestra página web.

Para entender esta propiedad hay algunos conceptos importantes a tener en cuenta:

- Elemento de referencia: Cuando movemos un elemento SIEMPRE se moverá respecto a un punto de referencia.
- Elemento posicionado: Decimos que un elemento está posicionado cuando utilizemos un valor de position DISTINTO a static.
- Static es el valor por defecto de position y lo tendrán todos los elementos salvo que se lo cambiemos manualmente.
- Espacio reservado: Cuando el navegador pinta los elementos en pantalla reserva el espacio que ocupa cada uno, al utilizar position podemos hacer que los elementos posicionados pierdan ese espacio reservado.



CSS POSITION

Al posicionar cualquier elemento se nos “desbloquean” cinco propiedades nuevas:

- **Top:** Mueve el elemento hacia abajo si el valor es positivo y hacia arriba si el valor es negativo.
- **Right:** Mueve el elemento hacia la izquierda si el valor es positivo y hacia la derecha si el valor es negativo.
- **Bottom:** Mueve el elemento hacia arriba si el valor es positivo y hacia abajo si el valor es negativo.
- **Left:** Mueve el elemento hacia la derecha si el valor es positivo y hacia la izquierda si el valor es negativo.
- **Z-index:** Es la propiedad que utilizaremos para cambiar la posición de un elemento en el eje z, es decir modificar el orden en el que se apilan las cajas. Es **IMPORTANTE** recordar que esta propiedad sólo se nota cuando se superponen elementos.



CSS

POSITION - RELATIVE

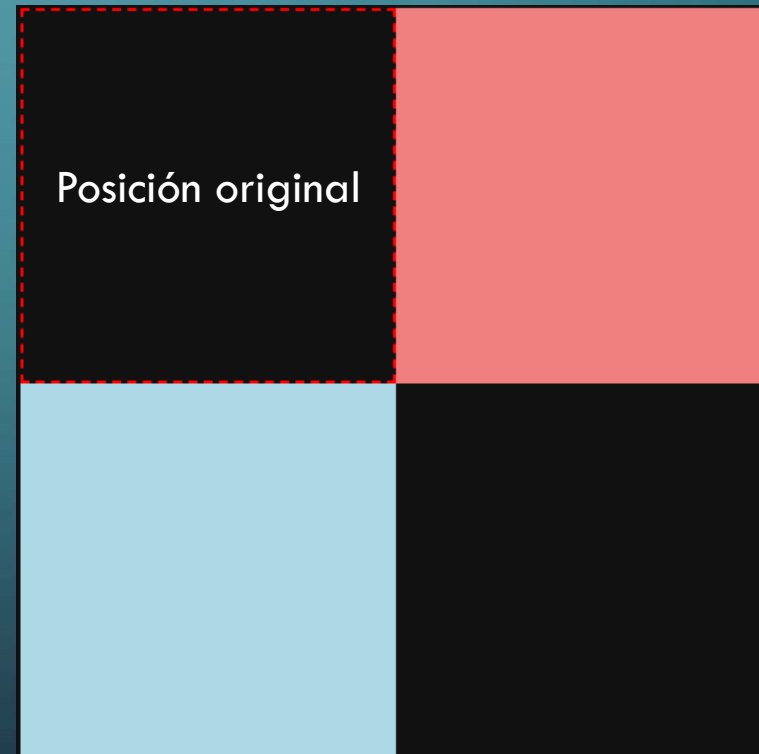
Cuando utilizamos el valor `relative`, el elemento mantendrá su posición y medidas en el flujo de renderizado y mantendrá su espacio reservado. Si lo movemos lo hará usando su posición en el HTML como punto de referencia.

```
<body>
  <div class="box box-1"></div>
  <div class="box box-2"></div>
</body>
```

```
.box {
  width: 200px;
  height: 200px;
}

.box-1 {
  position: relative;
  left: 200px;
  background-color: lightcoral;
}

.box-2 {
  background-color: lightblue;
}
```



CSS

POSITION - ABSOLUTE

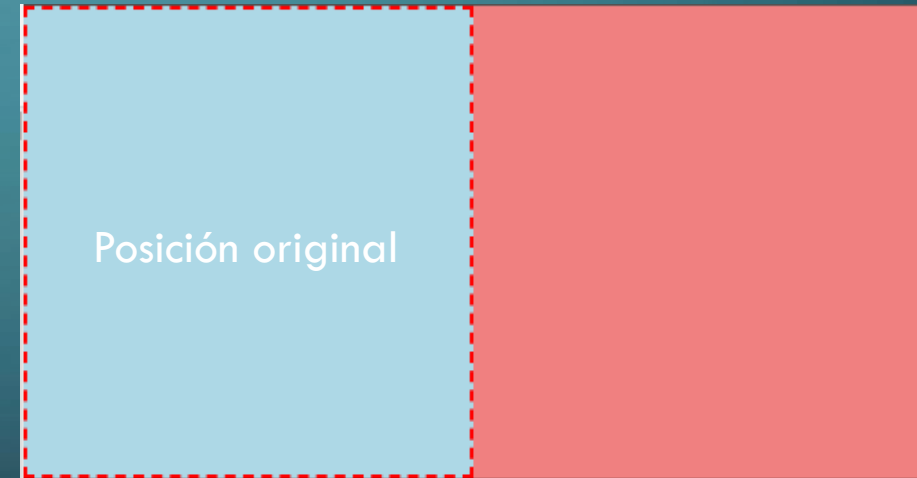
Cuando utilizamos el valor absolute, el elemento perderá su espacio reservado. Si lo movemos lo hará usando su contenedor con position (padre, abuelo...) más cercano como punto de referencia, si no encontrara ninguno lo hará respecto al elemento raíz (body)

```
<body>
  <div class="box box-1"></div>
  <div class="box box-2"></div>
</body>
```

```
.box {
  width: 200px;
  height: 200px;
}

.box-1 {
  position: absolute;
  left: 200px;
  background-color: lightcoral;
}

.box-2 {
  background-color: lightblue;
}
```



CSS

POSITION - ABSOLUTE

Es muy IMPORTANTE recordar que cuando usamos `position: absolute`, el elemento perderá sus medidas y sólo medirá lo que ocupe su contenido, al usar `absolute` es muy común tener que especificar un `width` o un `height` dependiendo de la situación.

```
<header class="header">
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
</header>
```

```
.header {
  height: 100px;
  background-color: lightcoral;
}
```

Lorem, ipsum dolor sit amet consectetur adipisicing elit.

```
.header {
  position: absolute;
  height: 100px;
  background-color: lightcoral;
}
```

Lorem, ipsum dolor sit amet consectetur adipisicing elit.



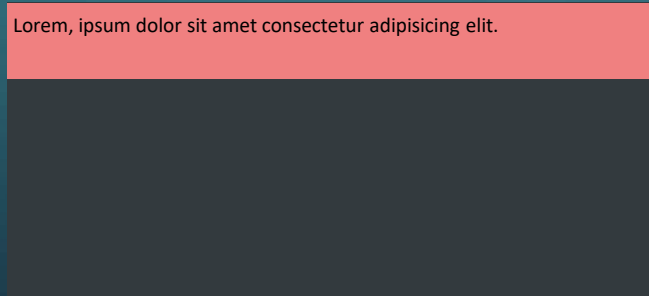
CSS

POSITION - FIXED

El elemento perderá sus medidas y su espacio reservado. Si lo movemos usará el viewport de referencia.

```
<header class="header">  
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>  
</header>
```

```
.header {  
  position: fixed;  
  height: 100px;  
  width: 100%;  
  background-color: lightcoral;  
}
```



Lorem, ipsum dolor sit amet consectetur adipisicing elit.

El elemento se quedará fijo en esa posición aunque hagamos scroll.



CSS

POSITION - STICKY

Con este tipo de posicionamiento los valores top, left, bottom y right no sirven para mover el elemento, si no para indicarle en qué punto pasará a tener un comportamiento de posicionamiento similar a fixed, hasta llegar a ese punto se comportará como si no tuviera posicionamiento, aunque sí contará como posicionado si necesitáramos colocar otro elemento respecto a él (absolute)

Ejemplo 1 - <https://codepen.io/DorianDesings/pen/abGRjwY>



CSS

POSITION – Z-INDEX

Es la propiedad que nos permite ordenar los elementos superpuestos para controlar cual se coloca por delante y cual por detrás. Este es el comportamiento por defecto.

```
<div class="box box-1"></div>  
<div class="box box-2"></div>  
<div class="box box-3"></div>
```

```
.box {  
  position: relative;  
  width: 150px;  
  height: 150px;  
}  
  
.box-1 {  
  background-color: lightcoral;  
}  
  
.box-2 {  
  top: -50px;  
  left: 100px;  
  background-color: lightgreen;  
}  
  
.box-3 {  
  top: -100px;  
  left: 200px;  
  background-color: lightseagreen;  
}
```



CSS

POSITION – Z-INDEX

Es recomendable trabajar con valores de z-index NO CONSECUTIVOS, 10, 20, 30... o 10, 100, 1000

```
.box {  
  position: relative;  
  width: 150px;  
  height: 150px;  
}  
  
.box-1 {  
  background-color: lightcoral;  
  z-index: 20;  
}  
  
.box-2 {  
  top: -50px;  
  left: 100px;  
  background-color: lightgreen;  
  z-index: 10;  
}  
  
.box-3 {  
  top: -100px;  
  left: 200px;  
  background-color: lightseagreen;  
}
```

```
<div class="box box-1"></div>  
<div class="box box-2"></div>  
<div class="box box-3"></div>
```

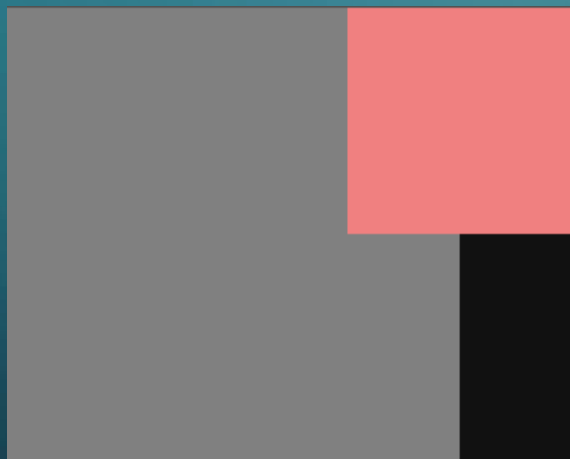


CSS

POSITION – Z-INDEX

Si queremos colocar un hijo por detrás de su padre es MUY IMPORTANTE que el padre no tenga un z-index declarado, y que el hijo tenga un z-index negativo.

```
<div class="container">  
  <div class="box"></div>  
</div>
```



```
.container {  
  width: 200px;  
  height: 200px;  
  background-color: gray;  
}  
  
.box {  
  position: relative;  
  left: 150px;  
  width: 100px;  
  height: 100px;  
  background-color: lightcoral;  
}
```

```
.box {  
  position: relative;  
  left: 150px;  
  width: 100px;  
  height: 100px;  
  background-color: lightcoral;  
  z-index: -1;  
}
```

