

Posicionamiento

Cuando vamos colocando elementos en una página web, estos fluyen a lo largo de la página; recordando que los elementos de tipo bloque ocupan por defecto todo el ancho de su contenedor y que realizan un salto de línea antes y después, y los de tipo en línea no.

Este posicionamiento podemos gestionarlo con la propiedad **position** que nos ofrece cuatro posibilidades (además de *inherit*): *relative*, *absolute*, *static* y *fixed*. Esta propiedad no se hereda y se puede aplicar a cualquier elemento.

Con ella podemos cambiar la posición de un elemento en la página y superponer unos elementos encima de otros. Estas coordenadas podremos indicarnos mediante las propiedades **top**, **bottom**, **right** y **left** (arriba, abajo, derecha e izquierda). Si se usan a la vez las propiedades top y bottom o right y left, top y left tienen prioridad.

Bloque contenedor

Como veremos posteriormente algunos posicionamientos se ven afectados por el contenedor del elemento. Para conocer este contenedor CSS 2.1 nos indica lo siguiente:

- si un elemento es raíz (solo está dentro de una etiqueta *body*) su contenedor es la etiqueta *html* o *body* según el navegador. A efectos prácticos será la ventana (la parte donde se visualiza la página)
- en un elemento no raíz con posicionamiento relativo o estático su contenedor será el antepasado más cercano de tipo bloque, tabla, celda de tabla, elemento de lista o bloque en línea
- en un elemento no raíz con posicionamiento absoluto el contenedor es el antepasado más cercano de cualquier tipo que tenga un posicionamiento absoluto, fijo o relativo

Posicionamiento absoluto

Podemos colocar el elemento de forma absoluta respecto a unas coordenadas fijas del elemento que lo contiene. En caso de que el elemento sea en línea, se comportará como bloque, con lo que se tendrán en cuenta todos sus márgenes y no solo los laterales. La posición arriba a la izquierda es la 0, 0.

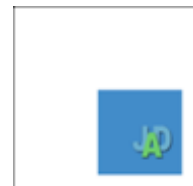
Con el siguiente ejemplo:

```
<body>
  
</body>
```

Colocamos la imagen en la posición 40, 40 de la ventana del navegador. Podemos colocar esta etiqueta en cualquier parte del código HTML ya que el posicionamiento del elemento no vendrá determinado por su posición en el flujo HTML.

En este código la imagen tiene las mismas coordenadas absolutas, pero observamos que pasa al estar contenida nuestra imagen dentro de otro elemento.

```
<body>
  <div style="position:absolute;left:100px;top:100px; width:100px;height:100px;
    background-color:rgb(66, 140, 201)">
    
  </div>
</body>
```



Como se observa se cambia el origen de las coordenadas y ahora se toma como referencia las coordenadas del elemento que lo contiene.

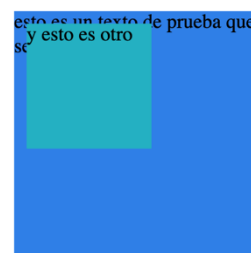
En el siguiente ejemplo el segundo div se coloca por encima del primero, pues al ser ambos hijos del body, las coordenadas se refieren respecto a este.

```
#uno {
  position:absolute;
  background-color:rgb(48, 127, 231);
  top:10px;
  left:10px;
  width:200px;
  height:200px;
}
#dos {
  position:absolute;
  background-color:rgb(34, 176, 194);
  top:10px;
  left:10px;
  width:100px;
  height:100px;
}
</style>
</head>
<body>
  <div id="uno">esto es un texto de prueba que se coloca en</div>
  <div id="dos">y esto es otro</div>
</body>
```

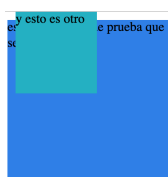


En este ejemplo, suponiendo los mismos estilos que en el ejemplo anterior, el segundo div se coloca con respecto al primero, pues este es el contenedor tal como vimos al comienzo (es el antepasado más cercano con posicionamiento no estático). Por ello ahora las coordenadas 0, 0 se refieren a este primer div.

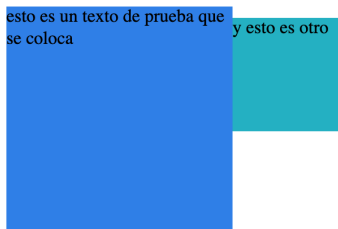
```
<div id="uno">esto es un texto de prueba que se coloca en
  <div id="dos">y esto es otro</div>
</div>
```



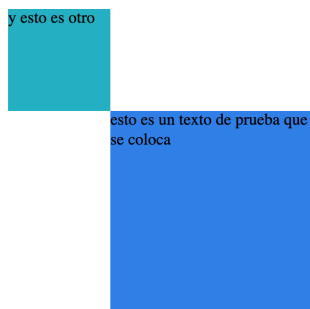
Ahora dándole un valor -10px a la propiedad top del segundo div:



También podemos usar porcentajes, tanto positivos como negativos en los ejemplos anteriores. Por ejemplo, si ponemos `left:100%` al segundo div, quedará fuera del mismo (a su derecha).



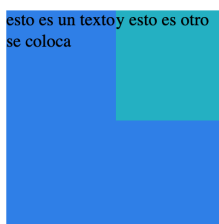
Ahora le damos un `top` y `left` a `-50%` al segundo div:



¿Por qué `-50` y no `-100%`? Porque le decimos un 50% del contenedor, es decir 100 píxeles pues su ancho es 200px; si usásemos `-100%`, serían 200 píxeles.

También podemos usar las propiedades `right` y `bottom` para colocar un elemento a la derecha o en la parte inferior.

Siguiendo con el ejemplo anterior de los div anidados, ahora colocamos la caja arriba a la derecha, usando `top: 0` y `right:0` en ese segundo div:



Posicionamiento fijo

Parecido el absoluto aunque siempre relativo a la ventana, sea cual sea su contenedor, y se mantiene siempre en esa posición aunque hagamos scroll (al modo del valor `fixed` de la propiedad `background-attachment`).

Se suele usar para tener una zona fija en la parte superior, inferior, derecha o izquierda, normalmente un encabezado.

Ejemplo:

```
<style>  
#uno {
```

```

position: absolute;
background-color: rgb(48, 127, 231);
top: 100px;
left: 200px;
width: 200px;
height: 200px;
}
#dos {
position: absolute;
background-color: rgb(34, 176, 194);
top: 0;
left: 0;
right: 0;
width: 100px;
height: 100px;
}
header {
position: fixed;
top: 0;
left: 0;
width: 100%;
height: 50px;
background-color: rgb(22, 114, 226);
font-size: x-large;
text-align: center;
}
main {
margin-top: 50px;
}
</style>
</head>
<body>
<header>Lorem Ipsum</header>
<main>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
  tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
  veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
  commodo consequat. Duis aute irure dolor in reprehenderit in voluptate
  velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit
  amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
  labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
  exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis
  aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
  fugiat nulla pariatur.
</main>

```

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Ved que hemos usado un margen superior en el estilo de la etiqueta para que no quede parte del texto debajo del encabezado y no se vea.

Posicionamiento relativo

En este caso recolocamos un elemento (desde su parte superior izquierda) con respecto a su posición original en el flujo natural del HTML. El espacio donde se debía situar en el flujo normal se conserva

vacío. Podemos usar las propiedades `top`, `left`, `bottom` y `right` para ello y valores negativos o positivos. Puede producirse solapamiento con otros elementos.

En el siguiente ejemplo colocamos el texto *ipsum* 20px por debajo de donde le tocaba. Fijaros que usamos un valor negativo para que se mueva hacia abajo; con un valor positivo sería 20 píxeles más arriba de la parte inferior. Podríamos haber conseguido lo mismo usando `top`: 20px.

```
span {
  position: relative;
  top:20px;
}
</style>
</head>
<body>
  <p>Lorem <span>ipsum</span> dolor sit, amet consectetur adipisicing elit. Sunt, exercitationem.</p>
```

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Sunt, exercitationem.
 ipsum

Ejemplos

Ejemplo1

Queremos que al pasar el ratón por encima de algunos términos en negrita, salga una explicación de estos:

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ullam, vel corrupti **numquam** quae eaque suscipit
 iste? Saepe excepturi distinctio minima repudiandae **exercitationem** animi incidunt esse, ut, co-
 repellat modi nemo praesentium id dicta perspiciatis placeat repellendus. Vero voluptate de
 praesentium, commodi enim libero dolor! Quisquam repellat impedit saepe libero atque s, cc

Esta es la
primera
explicación

Una posible solución es usar posicionamiento absoluto para esa caja azul y mostrarla cuando toque.

En el CSS:

```
.termino {
  font-weight: bold;
  /* Como el span de la definición va a usar posicionamiento absoluto, debemos
  hacer que su contenedor, el span con el término, tenga un posicionamiento
  diferente a estático (el valor por defecto). La solución fácil es darle
  posicionamiento relativo sin cambiar su top, left, bottom o right */
  position: relative;
}
/* Cuando pase el ratón sobre el término, cambiamos el display de la explicación
para que se muestre */
.termino:hover .explicacion {
  display: inline-block;
}
.explicacion {
  background-color: rgb(30, 125, 228);
  /* Con esto lo ocultamos y hacemos que no ocupe espacio.
  También tenemos visibility: hidden; pero esto mantiene
  el espacio vacío */
  display: none;
  /* Para poder moverla luego con respecto a su contenedor,
  que será el término a definir */
  position: absolute;
  /* La movemos un poco hacia abajo */
```

```

top: 10px;
/* Todo esto para hacerlo más bonito */
padding: 10px;
color: white;
border-radius: 5px;
border: 2px solid black;
}

```

En el HTML:

```

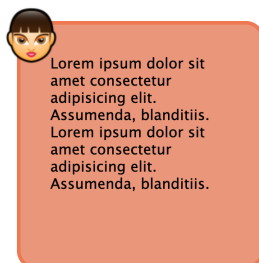
<p>
  Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ullam, vel
  corrupti
  <span class="termino">numquam<span class="explicacion">Esta es la primera explicación
    </span></span>
  quae eaque suscipit rem sequi nam animi officiis libero deleniti, debitis
  omnis fugit at id! Necessitatibus, maxime iste? Saepe excepturi distinctio
  minima repudiandae
  <span class="termino">exercitationem<span class="explicacion">
    Esta es la segunda explicación</span></span>
  animi incidunt esse? Sit cum illum, consectetur tenetur hic suscipit
  quidem
  ...
</p>

```

Hemos puesto un span dentro de otro, pues es técnicamente correcto, aunque podríamos ponerlo después y usar el selector + para seleccionar el span hermano al del término.

Ejemplo 2

Queremos mostrar una imagen en la parte superior izquierda de una caja y que, al pasar el ratón sobre la caja, se mueva un poco esta imagen hacia arriba.



Una posible solución es darle un posicionamiento absoluto a la imagen, con lo que la caja contenedora también deberá tener un posicionamiento no estático, y usar top y left para moverla.

En el CSS:

```

.ficha {
  /* Para que la caja no tenga posicionamiento estático y así la imagen la
  considere su contenedor */
  position: relative;
  /* Para que el texto no quede por debajo de la imagen, pues al moverla
  con posicionamiento absoluto, quedaría tapado */
  padding: 33px;
  /* Todo esto simple diseño */
  margin: 40px;
  width: 200px;
  height: 200px;
  border: 4px solid rgb(237, 120, 81);
  border-radius: 20px;
  background-color: darksalmon;
  font-family: 'Lucida Sans', 'Lucida Sans Regular', Geneva, Verdana, sans-serif;
}

```

```

}
.ficha img {
  /* Para colocarla con respecto a su contenedor; en nuestro caso la caja */
  position: absolute;
  /* Valores negativos para que quede parte de la imagen por fuera de la caja */
  top: -20px;
  left: -20px;
}
/* Al pasar el ratón por encima de la caja, movemos la imagen un poco más hacia arriba */
.ficha:hover img {
  top: -30px;
}

```

En el HTML:

```

<div class="ficha">
  
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda, blanditiis.
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda, blanditiis.
</div>

```

Ejemplo 3

Queremos poner el precio de la casa sobre la imagen, la cual no es un background, sino un img:



También con posicionamiento absoluto podemos colocar el párrafo del texto con respecto a la parte inferior de la caja contenedora.

En el CSS:

```

.casa {
  width: 300px;
  /* Para que la caja no tenga posicionamiento estático y así el párrafo la
  considere su contenedor */
  position: relative;
  border: 4px solid rgb(222, 7, 7);
}
.casa img {
  width: 100%;
  /* Para evitar un pequeño espacio que queda entre la image y el párrafo al
  ser un elemento en línea */
  display: block;
}
.casa p {
  position: absolute;
  /* Lo colocamos en la parte inferior dejando 10 píxeles entre el borde inferior
  y el párrafo */
  bottom: 10px;
  /* Todo esto para hacer el párrafo más bonito */
  margin: 0;
  font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', sans-serif;
  font-weight: bold;
  color: rgb(249, 251, 249);
  padding: 10px;
  font-size: 1.2em;
}

```



```
background-color: rgba(0,0,0,0.5); /* Medio transparente el color negro */
}
```

En el HTML:

```
<div class="casa">
  
  <p>420.000 €</p>
</div>
```