



HTML

Posicionamiento

Antes de explicar los valores de la propiedad `position`, es importante entender algunos conceptos clave como el flujo normal del navegador, que es la forma en que los elementos HTML se muestran en una página por defecto. Es decir, sin aplicar ningún tipo de posicionamiento.

En este flujo hay varios elementos, recordad que tenemos elementos en bloque o elementos en línea.

Un elemento en bloque sería un párrafo `<p>`:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Posicionamiento en CSS</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <p>Soy un elemento en bloque</p>
  </body>
</html>
```

Si le aplicamos un estilo podemos ver que ocupa todo el ancho del contenedor:

```
// style.css

p {
  border: 1px solid red;
}
```

A pesar de que su contenido no necesita todo el contenedor, va a ocupar el 100% de este.

Vamos a ver cómo se comporta un elemento en línea, en este caso sería un `span`:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

```

<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>Posicionamiento en CSS</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
    <p>Soy un elemento en bloque</p>
    <span>Soy un elemento en línea</span>
</body>
</html>

```

```

// style.css

p,
span {
    border: 1px solid red;
}

```

Y en este caso, si hacemos lo mismo y ponemos también este borde al span, el elemento en línea solo ocupa el ancho necesario para su contenido, es decir, no va a ocupar todo el ancho del 100% del contenedor. Si volvemos a poner un elemento span justo debajo, vamos a ver que se posiciona al lado hasta que ocupa el 100% y hay un salto de línea. En cambio el párrafo, a pesar de que no lo necesita, siempre va a ocupar el 100%.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <title>Posicionamiento en CSS</title>
        <link rel="stylesheet" href="style.css" />
    </head>
    <body>
        <p>Soy un elemento en bloque</p>
        <span>Soy un elemento en línea</span>
        <span>Soy un elemento en línea</span>
    </body>
</html>

```

Así que en HTML los elementos aparecen en pantalla en el orden en que se escriben en código, pero podemos hacer diseños más complejos y atractivos utilizando el posicionamiento en CSS.

Dos conceptos importantes son el contexto y el espacio reservado. El contexto es la zona que sirve como punto de referencia. Podemos pensar en él como un rectángulo imaginario a través del cual los elementos se posicionan, se mueven, se alinean o se distribuyen. Cuando hablamos de posicionamiento, el contexto es la posición inicial del elemento, es decir, donde se encuentra originalmente y desde la cual se calcula hacia

donde se moverá el elemento posicionado cuando colocamos la dirección con las propiedades `bottom`, `right`, `left` o `top`.

También es importante tener en cuenta que el navegador reservó un espacio para cada uno de los elementos para que ningún otro elemento ocupe ese espacio, es lo que se conoce como espacio reservado. Veremos que hay una casuística concreta donde ese espacio es inexistente ya que para el navegador ese elemento se convierte en invisible cuando utilizamos la posición absoluta.

Propiedad `position`

La propiedad `position`, como ya hemos dicho, nos permite mover un elemento a una posición diferente y además hace que este elemento pase a tener nuevas propiedades. Por defecto, los elementos se crean en la esquina superior izquierda de su contenedor, pero cuando utilizamos `position` podemos decidir desde dónde va a empezar a dibujarse ese elemento en nuestra página web. En este ejemplo, vamos a poder ver los diferentes tipos de posicionamiento que son: `Static`, `Relative`, `Absolute`, `Fixed` y `Sticky`.

`static`

El valor `static` es el valor por defecto de `position` y sigue el flujo normal de la página. Los elementos con `position: static` se posicionan acorde al flujo normal del documento HTML y no se ven afectados por las propiedades de posición (como `top`, `left`, `right`, `bottom` o `z-index`). Las propiedades de posición hacen posible mover el elemento en sus tres ejes:

- Eje X: `left` y `right`
- Eje Y: `top` y `bottom`
- Eje Z: `z-index` (profundidad) que nos sirve para que cuando dos o más elementos se solapan podamos decidir cuál aparece primero y cuál aparece por detrás.

Para que un elemento tenga activas estas propiedades tiene que tener un valor de `position` diferente a `static`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Posicionamiento en CSS</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>Posicionar elementos en CSS</h1>
    <div class="static">
      <h2>Static</h2>
      <p>
        Este elemento usa posición estática, la vista predeterminada para
        la
        posición de un elemento
      </p>
      <p>Un elemento estilizado con static 'no posicionado'</p>
    </div>
  </body>
</html>
```

```

    </div>
</body>
</html>

```

```

// style.css

div {
  padding: 10px;
  margin: 20px;
  margin-bottom: 50px;
  border: 3px solid indigo;
  background: #d5dbdb;
}

.static {
  position: static;
  top: 200px; // no funciona
}

```

relative

El valor **relative** permite mover un elemento respecto a su posición en el flujo normal de la página, es decir su posición por defecto. El desplazamiento se realiza utilizando las propiedades **top**, **left**, **right**, **bottom** y **z-index**.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Posicionamiento en CSS</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>Posicionar elementos en CSS</h1>
    <div class="relative1">
      <h2>Relative</h2>
      <p>
        El uso de la posición relativa da como resultado el mismo efecto
        que
        'static' a menos que utilices las propiedades top, left, bottom o
        right
      </p>
    </div>
    <div class="relative2">
      <p>
        Cuando utilizas estas propiedades, el elemento se ajusta fuera del
        flujo
        normal del navegador lejos de su posición habitual.
      </p>
    </div>
  </body>
</html>

```

```
        </p>
    </div>
</body>
</html>
```

```
// style.css

div {
    padding: 10px;
    margin: 20px;
    margin-bottom: 50px;
    border: 3px solid indigo;
    background: #d5dbdb;
}

.relative1 {
    position: relative;
    height: 400px;
}

.relative2 {
    position: relative;
    top: 20px;
    left: 200px;
    border: 3px solid red;
    width: 300px;
}
```

absolute

El elemento con `position: absolute` no estará dentro del flujo normal de la página y tomará como referencia la ventana del navegador o el elemento posicionado (que tenga cualquier valor de `position` excepto `static`) más cercano si es padre del elemento que queremos posicionar.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <title>Posicionamiento en CSS</title>
        <link rel="stylesheet" href="style.css" />
    </head>
    <body>
        <h1>Posicionar elementos en CSS</h1>
        <div class="relative1">
            <h2>Relative</h2>
            <p>
                Para que un elemento secundario tenga una posición absoluta, el
                elemento
                principal debe estar relativamente posicionado.
            </p>
        </div>
    </body>
</html>
```

```

</p>
<div class="absolute">
    <h2>Absolute</h2>
    <p>
        Un elemento absolutamente posicionado se posiciona en relación
        con su
        contenedor principal.
    </p>
</div>
</div>
</body>
</html>

```

```

// style.css

div {
    padding: 10px;
    margin: 20px;
    margin-bottom: 50px;
    border: 3px solid indigo;
    background: #d5dbdb;
}

.relative {
    position: relative;
    width: 600px;
    height: 400px;
}

.absolute {
    position: absolute;
    top: 120px;
    right: 0px;
    width: 300px;
    border: 3px solid red;
}

```

fixed

El valor **fixed** posiciona un elemento de forma fija respecto a la ventana del navegador. El elemento permanecerá en la misma posición incluso al hacer scroll en la página.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <title>Posicionamiento en CSS</title>
        <link rel="stylesheet" href="style.css" />
    </head>

```

```
<body>
    <h1>Posicionar elementos en CSS</h1>
    <div class="static">
        <h2>Static</h2>
        <p>
            Este elemento usa posición estática, la vista predeterminada para
la
            posición de un elemento
        </p>
        <p>Un elemento estilizado con static 'no posicionado'</p>
    </div>
    <div class="relative1">
        <h2>Relative</h2>
        <p>
            El uso de la posición relativa da como resultado el mismo efecto
que
            'static' a menos que utilices las propiedades top, left, bottom o
right
        </p>
    </div>
    <div class="relative2">
        <p>
            Cuando utilizas estas propiedades, el elemento se ajusta fuera del
flujo
            normal del navegador lejos de su posición habitual.
        </p>
    </div>
    <div class="relative">
        <h2>Relative</h2>
        <p>
            Para que un elemento secundario tenga una posición absoluta, el
elemento
            principal debe estar relativamente posicionado.
        </p>
        <div class="absolute">
            <h2>Absolute</h2>
            <p>
                Un elemento absolutamente posicionado se posiciona en relación
con su
                contenedor principal.
            </p>
        </div>
    </div>
    <div class="fixed">
        <h2>Fixed</h2>
        <p>Este elemento usa una posición fija.</p>
        <p>
            Se coloca en relación con la ventana del navegador, por lo que
siempre
            permanece en el mismo lugar. Como la posición relativa, también
utiliza
            las propiedades top, bottom, left y right para especificar su
ubicación.
        </p>
    </div>
```

```
</div>
</body>
</html>
```

```
// style.css

div {
    padding: 10px;
    margin: 20px;
    margin-bottom: 50px;
    border: 3px solid indigo;
    background: #d5dbdb;
}

.static {
    position: static;
}

.relative1 {
    position: relative;
    height: 400px;
}

.relative2 {
    position: relative;
    top: 20px;
    left: 200px;
    border: 3px solid red;
    width: 300px;
}

.relative {
    position: relative;
    width: 600px;
    height: 400px;
}

.absolute {
    position: absolute;
    top: 120px;
    right: 0px;
    width: 300px;
    border: 3px solid red;
}

.fixed {
    position: fixed;
    bottom: 0px;
    right: 0px;
    max-width: 25rem;
    border: 3px solid red;
}
```

sticky

El valor **sticky** hace que un elemento se comporte como si fuera **relative** hasta que se desplaza fuera de la ventana, momento en el que se comporta como **fixed**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Posicionamiento en CSS</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>Posicionar elementos en CSS</h1>
    <div>
      <h2>Sticky</h2>
      <p>Este elemento usa una posición sticky.</p>
      <p>
        Deserunt voluptate et commodo labore excepteur deserunt. Id ex
        nostrud
        sit eu. Duis irure magna voluptate reprehenderit pariatur nisi
        elit
        nulla do reprehenderit aute ex nisi. Amet Lorem excepteur laboris
        ullamco sint in sit excepteur reprehenderit sunt. Voluptate
        excepteur et
        enim voluptate laborum ad ex excepteur amet sint. Tempor eu eu
        excepteur
        sit do proident aliqua cupidatat est adipisicing incididunt. Nulla
        sint
        duis exercitation ad quis enim commodo nostrud sunt occaecat sit.
        Ipsum
        dolor laboris commodo est proident incididunt aute laborum dolore
        est
        quis velit adipisicing. Mollit excepteur minim commodo consectetur
        commodo incididunt elit et consequat cupidatat. Occaecat ullamco
        duis
        laborum nulla pariatur consequat amet et duis exercitation elit
        culpa.
      </p>
      <p class="sticky">
        Deserunt voluptate et commodo labore excepteur deserunt. Id ex
        nostrud
        sit eu. Duis irure magna voluptate reprehenderit pariatur nisi
        elit
        nulla do reprehenderit aute ex nisi. Cillum ex officia consectetur
        aliquip dolor sint incididunt esse magna minim consequat laboris
        do. Tempor laborum dolor voluptate minim amet reprehenderit velit
        duis. Enim aliquip quis magna deserunt ea voluptate est qui dolor
      </p>
    </div>
  </body>
</html>
```

consequat. Aute cupidatat aliquip labore duis veniam consectetur tempor
proident ad quis et tempor. Minim Lorem anim amet nisi excepteur
duis veniam enim irure sunt reprehenderit aliqua. Eu amet velit sit
officia voluptate officia velit. Minim ipsum est do culpa nulla aliquip id
culpa.

</p>

<p>

Deserunt voluptate et commodo labore excepteur deserunt. Id ex
nostrud sit eu. Duis irure magna voluptate reprehenderit pariatur nisi
elit nulla do reprehenderit aute ex nisi. Consectetur reprehenderit
laboris non veniam. Enim incididunt ex voluptate dolore aute eu.
Reprehenderit et nulla eu eiusmod quis adipisicing ut eiusmod.

</p>

<p>

Deserunt voluptate et commodo labore excepteur deserunt. Id ex
nostrud sit eu. Duis irure magna voluptate reprehenderit pariatur nisi
elit nulla do reprehenderit aute ex nisi. Pariatur exercitation ea
officia id sunt nisi eu sit. Sunt dolore adipisicing sunt ex quis qui qui
minim et fugiat Lorem tempor fugiat. Officia velit dolore velit deserunt
proident mollit Lorem esse minim culpa quis. Dolore id ad et veniam
pariatur Lorem consequat cillum laborum qui.

</p>

<p>

Deserunt voluptate et commodo labore excepteur deserunt. Id ex
nostrud sit eu. Duis irure magna voluptate reprehenderit pariatur nisi
elit nulla do reprehenderit aute ex nisi. Enim elit velit minim
excepteur magna dolor dolore dolore proident. Aute tempor minim irure nisi
quis mollit adipisicing aliquip dolor Lorem. Non nostrud sit nostrud
nisi commodo dolore mollit. Nostrud reprehenderit tempor tempor
laboris. Mollit et nisi qui veniam sit veniam non cillum enim qui sit velit
autem excepteur.

</p>

<p>

Deserunt voluptate et commodo labore excepteur deserunt. Id ex

nostrud sit eu. Duis irure magna voluptate reprehenderit pariatur nisi elit nulla do reprehenderit aute ex nisi. Non aute dolore occaecat proident ipsum nulla cillum ut occaecat aliqua amet aute. Ullamco tempor consectetur adipisicing duis labore voluptate proident pariatur ut fugiat et laboris elit anim. Commodo irure mollit do dolor voluptate. Proident adipisicing consectetur ipsum ex elit ipsum aliqua consequat officia pariatur duis. Qui elit mollit qui tempor id nisi id aliquip cupidatat incididunt quis.

</p>

<p>

Deserunt voluptate et commodo labore excepteur deserunt. Id ex nostrud sit eu. Duis irure magna voluptate reprehenderit pariatur nisi elit nulla do reprehenderit aute ex nisi. Exercitation deserunt ad dolor cupidatat ea duis ad sint eiusmod ut. Ea cillum et ullamco proident. Labore ex consequat enim occaecat qui officia officia aute ut anim consectetur exercitation. Eu esse et laborum et ipsum ex velit nostrud. Occaecat aliqua excepteur minim enim. Laborum cupidatat ad eu commodo ex anim aliqua aliquip dolore sit. Est sint in commodo tempor quis deserunt aliquip adipisicing laboris veniam velit adipisicing sint et.

</p>

<p>

Deserunt voluptate et commodo labore excepteur deserunt. Id ex nostrud sit eu. Duis irure magna voluptate reprehenderit pariatur nisi elit nulla do reprehenderit aute ex nisi. Tempor ad tempor cupidatat exercitation excepteur ea. Laboris consectetur ipsum veniam minim labore ex aute ipsum culpa anim fugiat excepteur. Incididunt aliqua quis exercitation excepteur voluptate tempor voluptate Lorem proident elit nisi minim eu nostrud. Magna incididunt mollit cupidatat fugiat. Labore id esse adipisicing esse consequat. Labore reprehenderit quis in aliqua ea.

</p>

</div>

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

```
// style.css

div {
    padding: 10px;
    margin: 20px;
    margin-bottom: 50px;
    border: 3px solid indigo;
    background: #d5dbdb;
}

.sticky {
    position: sticky;
    top: 0px;
    left: 0px;
    border: 3px solid red;
    background: #d5dbdb;
}
```

Estos son los principales valores de la propiedad `position` en CSS. Cada uno tiene sus usos y comportamientos particulares, lo que permite un mayor control sobre la disposición y el posicionamiento de los elementos en una página web.