

Esta página ha sido traducida del inglés por la comunidad. Aprende más y únete a la comunidad de MDN Web Docs.

# position

**Baseline** Widely available

La propiedad `position` de [CSS](#) especifica cómo un elemento es posicionado en el documento. Las propiedades `top`, `right`, `bottom`, y `left` determinan la ubicación final de los elementos posicionados.

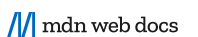
## Pruébalo

CSS Demo: position RESET

```
position: static;
```

```
position: relative;
top: 40px; left: 40px;
```

```
position: absolute;
top: 40px; left: 40px;
```



El código fuente de este ejemplo interactivo se encuentra almacenado en un repositorio de GitHub. Si quisiera contribuir al proyecto de ejemplos interactivos, por favor clone <https://github.com/mdn/interactive-examples> y envíenos sus contribuciones como un pull request.

## Tipos de posicionamiento

- Un **elemento posicionado** es un elemento cuyo valor [computado](#) de `position` es `relative`, `absolute`, `fixed`, o `sticky`. (En otras palabras, cualquiera excepto `static`).
- Un **elemento posicionado relativamente** es un elemento cuyo valor [computado](#) de `position` es `relative`. Las propiedades `top` y `bottom` especifican el desplazamiento vertical desde su posición original; las propiedades `left` y `right` especifican su desplazamiento horizontal.
- Un **elemento posicionado absolutamente** es un elemento cuyo valor [computado](#) de `position` es `absolute` o `fixed`. Las propiedades `top`, `right`, `bottom`, y `left` especifican el desplazamiento desde los bordes del [bloque contenedor](#) <sup>(inglés)</sup> del elemento. (El bloque contenedor es el ancestro relativo al cual el elemento está posicionado). Si el elemento tiene márgenes, se agregarán al desplazamiento. el elemento establece un nuevo contexto de formato de bloque para su contenido

- Un **elemento posicionado fijamente** es un elemento cuyo valor de [position computado](#) es `sticky`. Es tratado como un elemento posicionado relativamente hasta que su [bloque contenedor](#) <sup>(inglés)</sup> cruza un límite establecido (como por ejemplo dando a [top](#) cualquier valor distinto de `auto`), dentro de su flujo principal (o el contenedor dentro del cual se mueve), desde el cual es tratado como "fijo" hasta que alcance el borde opuesto de su [bloque contenedor](#) <sup>(inglés)</sup>.

La mayoría de las veces, los elementos absolutamente posicionados que tienen su [height](#) y [width](#) establecidos en `auto` son ajustados hasta acomodarse a su contenido. Sin embargo, elementos non-[replaced](#) y absolutamente posicionados se pueden crear para llenar el espacio vertical disponible, especificando tanto [top](#) como [bottom](#), y dejando [height](#) sin especificar (es decir, `auto`). De igual manera se pueden utilizar para llenar el espacio horizontal disponible especificando tanto [left](#) como [right](#), y dando a [width](#) el valor de `auto`.

A excepción del caso anteriormente descrito (de elementos posicionados absolutamente rellenoando el espacio disponible):

- Si ambos, `top` y `bottom` están especificados (técnicamente, no `auto`), `top` gana.
- Si ambos, `left` y `right`, están especificados, `left` gana cuando [direction](#) es `ltr` (Inglés, japonés horizontal, etc.) y `right` gana cuando [direction](#) es `rtl` (Persa, árabe, hebreo, etc.).

## Sintaxis

La propiedad `position` es especificada como una palabra única elegida de la siguiente lista de valores.

### Valores

#### `static`

El elemento es posicionado de acuerdo al flujo normal del documento. Las propiedades [top](#), [right](#), [bottom](#), [left](#), and [z-index](#) *no tienen efecto*. Este es el valor por defecto.

#### `relative`

El elemento es posicionado de acuerdo al flujo normal del documento, y luego es desplazado *con relación a sí mismo*, con base en los valores de `top`, `right`, `bottom`, and `left`. El desplazamiento no afecta la posición de ningún otro elemento; por lo que, el espacio que se le da al elemento en el esquema de la página es el mismo como si la posición fuera `static`. Este valor crea un nuevo [contexto de apilamiento](#), donde el valor de `z-index` no es `auto`. El efecto que tiene `relative` sobre los elementos `table-*group`, `table-row`, `table-column`, `table-cell`, y `table-caption` no está definido.

#### `absolute`

El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el elemento en el esquema de la página. Es posicionado relativo a su ancestro posicionado más cercano, si lo hay; de lo contrario, se ubica relativo al [bloque contenedor](#) <sup>(inglés)</sup> inicial. Su posición final está determinada por los valores de `top`, `right`, `bottom`, y `left`.

Este valor crea un nuevo [contexto de apilamiento](#) cuando el valor de `z-index` no es `auto`. Elementos absolutamente posicionados pueden tener margen, y no colapsan con ningún otro margen.

#### `fixed`

El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el elemento en el esquema de la página. Es posicionado con relación al [bloque contenedor](#) <sup>(inglés)</sup> inicial establecido por el [viewport](#), excepto cuando uno de sus ancestros tiene una propiedad `transform`, `perspective`, o `filter` establecida en algo que no sea `none` (ver [CSS Transforms Spec](#) ), en cuyo caso ese ancestro se comporta como el bloque contenedor. (Notar que hay inconsistencias del navegador con `perspective` y `filter` contribuyendo a la formación del bloque contenedor.) Su posición final es determinada por los valores de `top`, `right`, `bottom`, y `left`.

Estos valores siempre crean un nuevo [contexto de apilamiento](#). En documentos impresos, el elemento se coloca en la misma posición en *cada* página.

#### sticky

El elemento es posicionado de acuerdo al flujo normal del documento, y luego es desplazado *con relación a su ancestro que se desplace más cercano* y su [bloque contenedor](#) <sup>(inglés)</sup> (ancestro en nivel de bloque más cercano) incluyendo elementos relacionados a tablas, basados en los valores de `top`, `right`, `bottom`, y `left`. El desplazamiento no afecta la posición de ningún otro elemento.

Estos valores siempre crean un nuevo [contexto de apilamiento](#). Nótese que un elemento sticky se "adhiere" a su ancestro más cercano que tiene un "mecanismo de desplazamiento" (creado cuando el `overflow` es `hidden`, `scroll`, `auto`, o bien `overlay`), aún si ese ancestro no es el ancestro con desplazamiento más cercano. Esto inhibe efectivamente el comportamiento "sticky" (ver el [Github issue en W3C CSSWG](#) ).

## Sintaxis formal

```
position =
  static      .|.
  relative    .|.
  absolute    .|.
  sticky      .|.
  fixed       .|.
  <running()>

<running()> =
  running( <custom-ident> (inglés) )
```

## Examples

### Posicionamiento relativo

Elementos posicionados relativamente son desplazados una cantidad dada de su posición normal en el documento, pero sin que su desplazamiento afecte a otros elementos. En el ejemplo siguiente, nótese cómo los demás elementos se ubican como si "Two" estuviera ocupando el lugar de su ubicación normal.

#### HTML

```
HTML
<div class="box" id="one">One</div>
<div class="box" id="two">Two</div>
<div class="box" id="three">Three</div>
<div class="box" id="four">Four</div>
```

[Play](#)

## CSS

CSS

Play

```
.box {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  background: red;  
  color: white;  
}  
  
#two {  
  position: relative;  
  top: 20px;  
  left: 20px;  
  background: blue;  
}
```

Play

## Posicionamiento absoluto

Los elementos posicionados relativamente se mantienen en el flujo normal del documento. Por el contrario, un elemento posicionado absolutamente es removido del flujo de esta manera, los demás elementos se posicionan como si el mismo no existiera. El elemento posicionado absolutamente se posiciona relativamente a su *ancestro posicionado más cercano* (es decir, el ancestro más cercano que no es `static`). Si no hay ningún ancestro posicionado se ubica relativo al [bloque contenedor](#) <sup>(inglés)</sup> inicial. En el ejemplo siguiente, la caja "Two" no tiene un ancestro posicionado, por lo tanto se posiciona relativo al `<body>` del documento.

## HTML

HTML

Play

```
<div class="box" id="one">One</div>  
<div class="box" id="two">Two</div>  
<div class="box" id="three">Three</div>  
<div class="box" id="four">Four</div>
```

## CSS

CSS

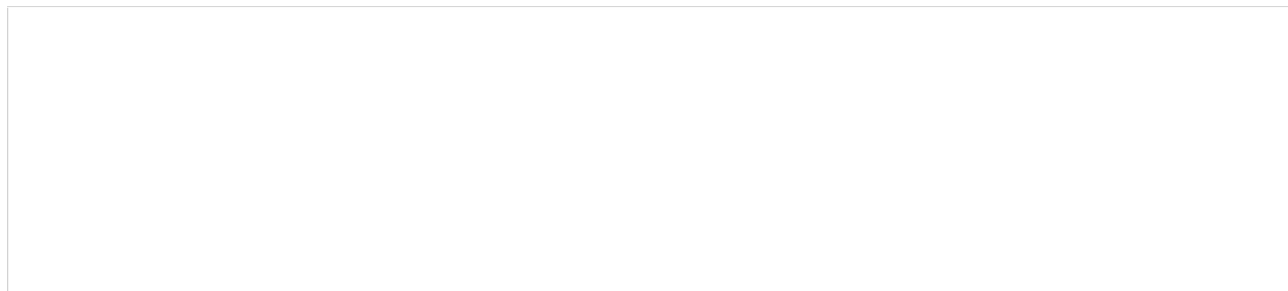
Play

```
.box {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  background: red;  
}
```

```
    color: white;
}

#two {
    position: absolute;
    top: 20px;
    left: 20px;
    background: blue;
}
```

Play



## Posicionamiento fijo

El posicionamiento fijo es similar al posicionamiento absoluto, con la excepción de que el bloque contenedor del elemento es el *viewport*. Esto puede usarse para crear un elemento flotante que se mantiene en la misma posición independientemente del desplazamiento. En el ejemplo siguiente, la caja "One" está fijada a 80 pixels del límite superior de la página y 10 pixels a la izquierda. Aún luego de desplazarse, se mantiene en el mismo lugar relativo al viewport.

HTML

Play

```
<div class="outer">
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam congue tortor
    eget pulvinar lobortis. Vestibulum ante ipsum primis in faucibus orci luctus
    et ultrices posuere cubilia Curae; Nam ac dolor augue. Pellentesque mi mi,
    laoreet et dolor sit amet, ultrices varius risus. Nam vitae iaculis elit.
    Aliquam mollis interdum libero. Sed sodales placerat egestas. Vestibulum ut
    arcu aliquam purus viverra dictum vel sit amet mi. Duis nisl mauris, aliquam
    sit amet luctus eget, dapibus in enim. Sed velit augue, pretium a sem
    aliquam, congue porttitor tortor. Sed tempor nisl a lorem consequat, id
    maximus erat aliquet. Sed sagittis porta libero sed condimentum. Aliquam
    finibus lectus nec ante congue rutrum. Curabitur quam quam, accumsan id
    ultrices ultrices, tempor et tellus.
  </p>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam congue tortor
    eget pulvinar lobortis. Vestibulum ante ipsum primis in faucibus orci luctus
    et ultrices posuere cubilia Curae; Nam ac dolor augue. Pellentesque mi mi,
    laoreet et dolor sit amet, ultrices varius risus. Nam vitae iaculis elit.
    Aliquam mollis interdum libero. Sed sodales placerat egestas. Vestibulum ut
    arcu aliquam purus viverra dictum vel sit amet mi. Duis nisl mauris, aliquam
    sit amet luctus eget, dapibus in enim. Sed velit augue, pretium a sem
    aliquam, congue porttitor tortor. Sed tempor nisl a lorem consequat, id
    maximus erat aliquet. Sed sagittis porta libero sed condimentum. Aliquam
    finibus lectus nec ante congue rutrum. Curabitur quam quam, accumsan id
    ultrices ultrices, tempor et tellus.
  </p>
</div>
```

```
<div class="box" id="one">One</div>
</div>
```

## CSS

CSS

Play

```
.box {
  width: 100px;
  height: 100px;
  background: red;
  color: white;
}

#one {
  position: fixed;
  top: 80px;
  left: 10px;
  background: blue;
}

.outer {
  width: 500px;
  height: 300px;
  overflow: scroll;
  padding-left: 150px;
}
```

Play

## Posicionamiento sticky

El posicionamiento sticky puede considerarse un híbrido de los posicionamientos relativo y fijo. Un elemento con posicionamiento sticky es tratado como un elemento posicionado relativamente hasta que cruza un umbral especificado, en cuyo punto se trata como fijo hasta que alcanza el límite de su padre. Por ejemplo...

CSS

Play

```
#one {
  position: sticky;
  top: 10px;
}
```

...posicionaría el elemento con id *uno* relativamente hasta que el viewport sea desplazado de manera tal que el elemento esté a menos de 10 píxeles del límite superior. Más allá de ese umbral, el elemento sería fijado a 10

píxeles del límite superior.

Un uso común para el posicionamiento sticky es para los encabezados en una lista alfabética. El encabezado "B" aparecerá justo por debajo de los ítems que comienzan con "A" hasta que se hayan desplazado más allá de la pantalla. En vez de deslizarse fuera de la pantalla como el resto del contenido, el encabezado "B" se mantendrá fijado al límite superior del viewport hasta que todos los ítems "B" se hayan desplazado fuera de la pantalla, en cuyo punto será cubierto por el encabezado "C", y así sucesivamente. Se debe especificar un umbral con al menos uno de `top`, `right`, `bottom`, or `left` para que el posicionamiento sticky se comporte de forma esperada. Caso contrario, será indistinguible del posicionamiento relativo.

## HTML

HTML	Play
<pre>&lt;dl&gt;   &lt;div&gt;     &lt;dt&gt;A&lt;/dt&gt;     &lt;dd&gt;Andrew W.K.&lt;/dd&gt;     &lt;dd&gt;Apparat&lt;/dd&gt;     &lt;dd&gt;Arcade Fire&lt;/dd&gt;     &lt;dd&gt;At The Drive-In&lt;/dd&gt;     &lt;dd&gt;Aziz Ansari&lt;/dd&gt;   &lt;/div&gt;   &lt;div&gt;     &lt;dt&gt;C&lt;/dt&gt;     &lt;dd&gt;Chromeo&lt;/dd&gt;     &lt;dd&gt;Common&lt;/dd&gt;     &lt;dd&gt;Converge&lt;/dd&gt;     &lt;dd&gt;Crystal Castles&lt;/dd&gt;     &lt;dd&gt;Cursive&lt;/dd&gt;   &lt;/div&gt;   &lt;div&gt;     &lt;dt&gt;E&lt;/dt&gt;     &lt;dd&gt;Explosions In The Sky&lt;/dd&gt;   &lt;/div&gt;   &lt;div&gt;     &lt;dt&gt;T&lt;/dt&gt;     &lt;dd&gt;Ted Leo &amp; The Pharmacists&lt;/dd&gt;     &lt;dd&gt;T-Pain&lt;/dd&gt;     &lt;dd&gt;Thrice&lt;/dd&gt;     &lt;dd&gt;TV On The Radio&lt;/dd&gt;     &lt;dd&gt;Two Gallants&lt;/dd&gt;   &lt;/div&gt; &lt;/dl&gt;</pre>	

## CSS

CSS	Play
<pre>* {   box-sizing: border-box; }  dl &gt; div {   background: #fff;   padding: 24px 0 0 0; }</pre>	

```
dt {  
  background: #b8c1c8;  
  border-bottom: 1px solid #989ea4;  
  border-top: 1px solid #717d85;  
  color: #fff;  
  font:  
    bold 18px/21px Helvetica,  
    Arial,  
    sans-serif;  
  margin: 0;  
  padding: 2px 0 0 12px;  
  position: -webkit-sticky;  
  position: sticky;  
  top: -1px;  
}  
  
dd {  
  font:  
    bold 20px/45px Helvetica,  
    Arial,  
    sans-serif;  
  margin: 0;  
  padding: 0 0 0 12px;  
  white-space: nowrap;  
}  
  
dd + dd {  
  border-top: 1px solid #ccc;  
}
```

Play

## Consideraciones de accesibilidad

Asegurarse de que los elementos posicionados con valor `absolute` o `fixed` no oscurezcan el resto del contenido cuando la página sea ampliada para aumentar el tamaño del texto.

- [MDN entendiendo el WCAG, explicaciones de los lineamientos 1.4.](#)
- [Presentación visual: Entendiendo SC 1.4.8 | Entendiendo WCAG 2.0](#)

## Performance y accesibilidad



Los elementos que se desplazan que contienen contenido `fixed` o `sticky` content pueden ocasionar problemas de performance y accesibilidad. Mientras un usuario desplaza la página, el navegador debe realizar repaint del contenido sticky o fijo en una nueva locación. Dependiendo del contenido que precisa ser repainted, la performance del navegador y la velocidad de procesamiento del dispositivo, el navegador podría no ser capaz de realizar el repaint a 60 fps, ocasionando problemas de accesibilidad para personas con sensibilidades y desprolijidad para todos. Una solución es agregar [will-change: transform](#) <sup>(inglés)</sup> a los elementos posicionados para renderizar el elemento en su propia capa, mejorando la velocidad del repaint y por lo tanto mejorando performance y accesibilidad.

## Especificaciones

Specification
<a href="#">CSS Positioned Layout Module Level 3</a> <a href="#"># position-property</a>

## Compatibilidad con navegadores

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
position	1	12	1	4	1	18	4	14	1	1.0	4.4	1
absolute	1	12	1	15	1	18	4	14	1	1.0	4.4	1
Absolutely-positioned flex children	52	12	52	39	11	52	52	41	11	6.0	52	11
fixed	1	12	1	4	1	18	4	14	1	1.0	4.4	1
Table elements as sticky positioning containers	56	16	59	43	8	56	59	43	8	6.0	56	8
relative	1	12	1	15	1	18	4	14	1	1.0	4.4	1
static	1	12	1	15	1	18	4	14	1	1.0	4.4	1
sticky	56	16	32	43	13	56	32	43	13	6.0	56	13

*Tip: you can click/tap on a cell for more information.*

Full support

See implementation notes.

Requires a vendor prefix or different name for use.

Has more compatibility info.

## Help improve MDN

Was this page helpful to you?

[Learn how to contribute.](#)

This page was last modified on 17 dic 2024 by [MDN contributors](#).