



Tipos de posicionamiento



Posicionamiento estático:

Es la posición predeterminada de cualquier elemento. Se sitúa en su lugar original en la página, como si no hubiera CSS aplicado.

El posicionamiento estático es el valor predeterminado para la propiedad position en CSS. Cuando un elemento tiene un posicionamiento estático, su posición en la página web se determina según el flujo normal del documento, es decir, según cómo se colocan los elementos uno debajo del otro.

En otras palabras, el posicionamiento estático no cambia la posición de un elemento en la página. Si no se especifica ninguna otra propiedad de posicionamiento para el elemento, como top, bottom, left o right, entonces el elemento se colocará en su posición original en la página.



Aquí te dejo algunos detalles adicionales sobre el posicionamiento estático:

- Los elementos con posicionamiento estático no se ven afectados por las propiedades de z-index, por lo que siempre se colocarán en el orden en que aparecen en el código HTML.
- Los elementos con posicionamiento estático no se pueden mover con las propiedades de top, bottom, left o right.
- Los elementos con posicionamiento estático se ajustarán automáticamente para adaptarse al tamaño de su contenido, a menos que se les especifique una anchura o altura fija.
- El posicionamiento estático se puede establecer explícitamente con la propiedad `position: static`, aunque como se mencionó anteriormente, este valor es el valor predeterminado y no es necesario establecerlo de manera explícita.

En resumen, el posicionamiento estático es la posición predeterminada de los elementos en CSS y es la forma en que se posicionan los elementos de manera predeterminada en una página web. Si no se especifica ninguna otra propiedad de posicionamiento, los elementos se colocarán según el flujo normal del documento.

Ejemplo:

```
div {  
  position: static;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Posicionamiento estático</title>  
  <style>  
    div {  
      background-color: yellow;  
      height: 100px;  
      width: 100px;  
    }  
  </style>  
</head>  
<body>  
  <div>Posicionamiento estático</div>  
</body>  
</html>
```



Posicionamiento relativo:

Se utiliza para mover un elemento con respecto a su posición original en la página. Permite ajustar la posición del elemento a través de los valores top, right, bottom, y left.

El posicionamiento relativo es una técnica de posicionamiento en CSS que permite mover un elemento en una dirección específica a partir de su posición original sin afectar a los otros elementos que se encuentran en la página.

Cuando se establece un posicionamiento relativo en un elemento, se pueden utilizar las propiedades top, bottom, left y right para mover el elemento en la dirección deseada, con respecto a su posición original. Por ejemplo, si se establece left: 20px, el elemento se moverá 20 píxeles a la derecha de su posición original.

Aquí te dejo algunos detalles adicionales sobre el posicionamiento relativo:

- El posicionamiento relativo no afecta la posición de los otros elementos en la página, es decir, los otros elementos se colocarán según el flujo normal del documento, sin verse afectados por el posicionamiento relativo.
- Los elementos con posicionamiento relativo aún ocupan su espacio original en la página, aunque se les haya movido con las propiedades de top, bottom, left o right.
- Los elementos con posicionamiento relativo se pueden solapar con otros elementos, pero el orden en que se solapan depende del valor de la propiedad z-index.
- La propiedad z-index sólo tiene efecto en elementos que tienen un posicionamiento relativo o absoluto.
- El posicionamiento relativo se puede establecer explícitamente con la propiedad position: relative.

En resumen, el posicionamiento relativo es una técnica de posicionamiento en CSS que permite mover un elemento en una dirección específica a partir de su posición original, sin afectar a los otros elementos en la página. Se utiliza la propiedad position: relative para establecer un posicionamiento relativo, y las propiedades top, bottom, left y right para mover el elemento en la dirección deseada.



Ejemplo:

```
div {  
  position: relative;  
  top: 10px;  
  left: 20px;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Posicionamiento relativo</title>  
  <style>  
    div {  
      background-color: yellow;  
      height: 100px;  
      width: 100px;  
      position: relative;  
      top: 10px;  
      left: 20px;  
    }  
  </style>  
</head>  
<body>  
  <div>Posicionamiento relativo</div>  
</body>  
</html>
```



Posicionamiento absoluto:

Este tipo de posición se utiliza para mover un elemento a una posición determinada en la página en función de su contenedor más cercano con posición relativa o absoluta. Permite ajustar la posición del elemento a través de los valores top, right, bottom, y left.

El posicionamiento absoluto en CSS es una técnica que permite posicionar un elemento en un lugar específico de la página, independientemente del flujo normal del documento y sin afectar a la posición de otros elementos.

Cuando se establece un posicionamiento absoluto en un elemento, se utiliza la propiedad top, bottom, left y right para definir la posición exacta del elemento, con respecto al elemento padre más cercano que tenga una posición relativa, absoluta o fija. Si no existe ningún elemento padre que tenga una posición establecida, el elemento se posicionará con respecto a la ventana del navegador.

Aquí te dejo algunos detalles adicionales sobre el posicionamiento absoluto:

- Los elementos con posicionamiento absoluto se colocan en la página en una posición específica, independientemente del flujo normal del documento.
- Los elementos con posicionamiento absoluto se pueden solapar con otros elementos, y el orden en que se solapan depende del valor de la propiedad z-index.
- Los elementos con posicionamiento absoluto no afectan la posición de los otros elementos en la página, a menos que se solapen con ellos.
- Los elementos con posicionamiento absoluto no ocupan ningún espacio en el flujo normal del documento, lo que significa que los otros elementos pueden ocupar el espacio que el elemento con posicionamiento absoluto "dejó".
- La propiedad z-index sólo tiene efecto en elementos que tienen un posicionamiento relativo o absoluto.
- El posicionamiento absoluto se puede establecer explícitamente con la propiedad position: absolute.

En resumen, el posicionamiento absoluto es una técnica de posicionamiento en CSS que permite posicionar un elemento en un lugar específico de la página, independientemente del flujo normal del documento y sin afectar a la posición de otros elementos. Se utiliza la propiedad position: absolute para establecer un posicionamiento absoluto, y las propiedades top, bottom, left y right para definir la posición exacta del elemento, con respecto al elemento padre más cercano que tenga una posición relativa, absoluta o fija.



Ejemplo:

```
div {  
  position: absolute;  
  top: 50px;  
  left: 50px;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Posicionamiento absoluto</title>  
  <style>  
    .container {  
      position: relative;  
      height: 300px;  
      width: 300px;  
      border: 1px solid black;  
    }  
  
    div {  
      background-color: yellow;  
      height: 100px;  
      width: 100px;  
      position: absolute;  
      top: 50px;  
      left: 50px;  
    }  
  </style>  
</head>  
<body>  
  <div class="container">  
    <div>Posicionamiento absoluto</div>  
  </div>  
</body>  
</html>
```



Posicionamiento fijo:

Este tipo de posición se utiliza para mover un elemento a una posición determinada en la página, independientemente de la posición del usuario. El elemento permanece fijo en su posición, incluso si se desplaza la página.

El posicionamiento fijo es una técnica de posicionamiento en CSS que permite posicionar un elemento en una posición fija en la ventana del navegador, independientemente de la posición del usuario en la página.

Cuando se establece un posicionamiento fijo en un elemento, se utiliza la propiedad `top`, `bottom`, `left` y `right` para definir la posición exacta del elemento con respecto a la ventana del navegador. El elemento con posicionamiento fijo permanecerá en la misma posición fija en la ventana del navegador, incluso si el usuario se desplaza por la página.

Aquí te dejo algunos detalles adicionales sobre el posicionamiento fijo:

- Los elementos con posicionamiento fijo permanecen en una posición fija en la ventana del navegador, independientemente de la posición del usuario en la página.
- Los elementos con posicionamiento fijo se pueden solapar con otros elementos, y el orden en que se solapan depende del valor de la propiedad `z-index`.
- Los elementos con posicionamiento fijo no afectan la posición de los otros elementos en la página.
- Los elementos con posicionamiento fijo no ocupan ningún espacio en el flujo normal del documento, lo que significa que los otros elementos pueden ocupar el espacio que el elemento con posicionamiento fijo "dejó".
- La propiedad `z-index` sólo tiene efecto en elementos que tienen un posicionamiento relativo o absoluto.
- El posicionamiento fijo se puede establecer explícitamente con la propiedad `position: fixed`.

En resumen, el posicionamiento fijo es una técnica de posicionamiento en CSS que permite posicionar un elemento en una posición fija en la ventana del navegador, independientemente de la posición del usuario en la página. Se utiliza la propiedad `position: fixed` para establecer un posicionamiento fijo, y las propiedades `top`, `bottom`, `left` y `right` para definir la posición exacta del elemento con respecto a la ventana del navegador.



Ejemplo

```
div {  
  position: fixed;  
  top: 0;  
  left: 0;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Posicionamiento fijo</title>  
  <style>  
    div {  
      background-color: yellow;  
      height: 100px;  
      width: 100px;  
      position: fixed;  
      top: 0;  
      left: 0;  
    }  
  </style>  
</head>  
<body>  
  <div>Posicionamiento fijo</div>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam dictum libero sit amet velit tempor  
finibus. Nam vitae mauris nec odio aliquet fermentum. Nunc id nibh vel nulla eleifend pharetra. In hac habitasse  
platea dictumst. Proin tristique purus quis nulla fermentum aliquet.</p>  
  <p>Sed vel nisi ac velit laoreet interdum a ac mi. Sed eget nulla sit amet quam sodales dignissim. In hac  
habitasse platea dictumst. Donec auctor libero id eros rutrum, eu tristique est cursus. Nulla facilisi.</p>  
</body>  
</html>
```



Posicionamiento (sticky):

Este tipo de posición se utiliza para mover un elemento a una posición determinada en la página, pero se comporta como un posicionamiento relativo hasta que se alcanza un punto determinado, donde se convierte en posicionamiento fijo.

El posicionamiento sticky es una técnica de posicionamiento en CSS que permite que un elemento se comporte como posición relativa hasta que se alcanza un límite específico y luego se convierte en fijo. En otras palabras, el elemento "se pega" a una posición específica en la ventana del navegador después de que se ha desplazado una cierta cantidad de píxeles.

Cuando se establece un posicionamiento sticky en un elemento, se utiliza la propiedad `top`, `bottom`, `left` y `right` para definir la posición exacta del elemento con respecto al elemento padre más cercano que tenga una posición relativa, absoluta o fija. Además, se debe establecer la propiedad `position: sticky` para que el elemento tenga un posicionamiento sticky.

Aquí te dejo algunos detalles adicionales sobre el posicionamiento sticky:

- Los elementos con posicionamiento sticky se comportan como posición relativa hasta que se alcanza un límite específico y luego se convierten en fijos.
- Los elementos con posicionamiento sticky se pueden solapar con otros elementos, y el orden en que se solapan depende del valor de la propiedad `z-index`.
- Los elementos con posicionamiento sticky no afectan la posición de los otros elementos en la página, pero pueden empujar otros elementos fuera de su camino a medida que se desplazan.
- Los elementos con posicionamiento sticky ocupan espacio en el flujo normal del documento.
- La propiedad `z-index` sólo tiene efecto en elementos que tienen un posicionamiento relativo o absoluto.
- El posicionamiento sticky se puede establecer explícitamente con la propiedad `position: sticky`.

En resumen, el posicionamiento sticky es una técnica de posicionamiento en CSS que permite que un elemento se comporte como posición relativa hasta que se alcanza un límite específico y luego se convierte en fijo. Se utiliza la propiedad `position: sticky` para establecer un posicionamiento sticky, y las propiedades `top`, `bottom`, `left` y `right` para definir la posición exacta del elemento con respecto al elemento padre más cercano que tenga una posición relativa, absoluta o fija.



Ejemplo:

```
div {  
  position: sticky;  
  top: 0;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Posicionamiento pegajoso</title>  
  <style>  
    div {  
      background-color: yellow;  
      height: 100px;  
      width: 100px;  
      position: sticky;  
      top: 0;  
    }  
  
    .container {  
      height: 2000px;  
    }  
  </style>  
</head>  
<body>  
  <div>Posicionamiento pegajoso</div>  
  <div class="container"></div>  
</body>  
</html>
```



Posicionamiento en cuadrícula (grid):

El posicionamiento en cuadrícula es una forma más avanzada de posicionar elementos en una página web. Se utiliza para crear diseños complejos con múltiples columnas y filas.

El posicionamiento en grid es una técnica de posicionamiento en CSS que permite diseñar diseños de página complejos y estructurados en una cuadrícula bidimensional. Con el posicionamiento en grid, se pueden definir filas y columnas para dividir la página en secciones y colocar elementos en cualquier celda de la cuadrícula.

Cuando se establece un posicionamiento en grid en un elemento contenedor, se utiliza la propiedad `display: grid` para convertir el elemento contenedor en una cuadrícula. A continuación, se pueden definir las filas y columnas de la cuadrícula utilizando las propiedades `grid-template-rows` y `grid-template-columns`. Además, se pueden establecer las áreas específicas de la cuadrícula en las que se colocarán los elementos utilizando la propiedad `grid-template-areas`.

Aquí te dejo algunos detalles adicionales sobre el posicionamiento en grid:

- El posicionamiento en grid permite dividir la página en una cuadrícula bidimensional de filas y columnas.
- Se puede especificar el tamaño de las filas y columnas utilizando unidades de medida como píxeles, porcentajes o fracciones de espacio disponible.
- Los elementos se pueden colocar en cualquier celda de la cuadrícula utilizando la propiedad `grid-row` y `grid-column`.
- Se puede especificar el espacio entre las filas y columnas de la cuadrícula utilizando la propiedad `grid-gap`.
- Se pueden agrupar las celdas de la cuadrícula en áreas utilizando la propiedad `grid-template-areas`, lo que facilita la organización y el diseño de la página.
- El posicionamiento en grid se puede utilizar en combinación con otras técnicas de posicionamiento, como el posicionamiento relativo y absoluto.

En resumen, el posicionamiento en grid es una técnica de posicionamiento en CSS que permite diseñar diseños de página complejos y estructurados en una cuadrícula bidimensional. Se utiliza la propiedad `display: grid` para convertir un elemento contenedor en una cuadrícula, y se pueden definir filas y columnas utilizando las propiedades `grid-template-rows` y `grid-template-columns`. Los elementos se pueden colocar en cualquier celda de la cuadrícula utilizando la propiedad `grid-row` y `grid-column`, y se pueden agrupar las celdas en áreas utilizando la propiedad `grid-template-areas`.



Ejemplo:

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 100px 200px;  
}
```

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <title>Posicionamiento en cuadrícula</title>  
  
  <style>  
  
    .container {  
  
      display: grid;  
  
      grid-template-columns: repeat(3, 1fr);  
      grid-template-rows: repeat(2, 150px);  
      grid-gap: 20px;  
  
      height: 500px;  
      width: 800px;  
      margin: 0 auto;  
  
    }  
  
    .box {  
  
      background-color: yellow;  
      height: 100px;  
  
      width: 100px;  
  
    }  
  
    .box-1 {  
  
      grid-column: 1 / 3;  
  
      grid-row: 1 / 3;  
  
    }  
  
    .box-2 {  
  
      grid-column: 3;  
  
      grid-row: 1 / 3;  
  
    }  
  
  </style>  
</head>  
  
<body>  
  
  <div class="container">  
  
    <div class="box"></div>  
  
    <div class="box"></div>  
  
    <div class="box"></div>  
  
    <div class="box"></div>  
  
  </div>  
  
</body>  
</html>
```



Posicionamiento en flexbox:

El posicionamiento en flexbox es otra forma avanzada de posicionar elementos en una página web. Se utiliza para alinear y distribuir elementos en una sola dirección.

El posicionamiento con flexbox es una técnica de posicionamiento en CSS que permite organizar y alinear elementos en una dimensión (horizontal o vertical) en función de un eje principal. Con el posicionamiento flexbox, se puede crear diseños fluidos y flexibles en una sola dimensión, lo que facilita la creación de diseños responsivos y adaptables.

Cuando se utiliza el posicionamiento con flexbox, se establece la propiedad `display: flex` en el contenedor que envuelve a los elementos que se desean alinear. Esto convierte al contenedor en un contenedor flex y permite que los elementos interiores se ordenen y se alineen en el eje principal del contenedor.

Aquí te dejo algunos detalles adicionales sobre el posicionamiento con flexbox:

- El posicionamiento con flexbox se utiliza para alinear elementos en una dimensión (horizontal o vertical) en función de un eje principal.
- El eje principal puede ser horizontal (`flex-direction: row`) o vertical (`flex-direction: column`).
- Los elementos dentro del contenedor flex se pueden ordenar y alinear utilizando las propiedades `justify-content` y `align-items`.
- La propiedad `justify-content` se utiliza para alinear los elementos a lo largo del eje principal, mientras que la propiedad `align-items` se utiliza para alinear los elementos en el eje perpendicular al eje principal.
- También se pueden utilizar otras propiedades flex para ajustar el tamaño de los elementos dentro del contenedor, como `flex-grow`, `flex-shrink`, y `flex-basis`.
- El posicionamiento con flexbox se puede utilizar en combinación con otras técnicas de posicionamiento, como el posicionamiento relativo y absoluto.

En resumen, el posicionamiento con flexbox es una técnica de posicionamiento en CSS que permite organizar y alinear elementos en una dimensión (horizontal o vertical) en función de un eje principal. Se utiliza la propiedad `display: flex` en el contenedor para convertirlo en un contenedor flex y permitir que los elementos interiores se ordenen y se alineen en el eje principal. Se pueden utilizar propiedades como `justify-content` y `align-items` para ajustar el alineamiento de los elementos, y otras propiedades flex para ajustar el tamaño de los elementos. El posicionamiento con flexbox es una técnica útil para crear diseños fluidos y responsivos.



Ejemplo:

```
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <title>Posicionamiento con Flexbox</title>  
  
  <style>  
  
    body {  
  
      margin: 0;  
  
      padding: 0;  
  
    }  
  
    .container {  
  
      display: flex;  
      flex-wrap: wrap;  
  
      justify-content: center;  
      align-items: center;  
      height: 100vh;  
  
    }  
  
    .box {  
  
      background-color: yellow;  
      height: 100px;  
  
      width: 100px;  
      margin: 10px;  
  
    }  
  
    .box-1 {  
  
      order: 2;  
  
    }  
  
    .box-2 {
```



En este ejemplo, hemos creado un contenedor con la clase `container` y le hemos aplicado las propiedades de Flexbox necesarias para posicionar los elementos dentro de él. Las diferentes clases de las cajas (`box-1`, `box-2`, etc.) se usan para aplicar diferentes propiedades de Flexbox a cada una de ellas.

Por ejemplo, la caja con la clase `box-1` tiene la propiedad `order` establecida en 2, lo que significa que aparecerá después de la caja `box-2` en el orden visual de la página.

La caja con la clase `box-2` tiene la propiedad `flex-grow` establecida en 1, lo que significa que se expandirá para llenar cualquier espacio disponible en el contenedor.

La caja con la clase `box-3` tiene la propiedad `flex-shrink` establecida en 2, lo que significa que se reducirá más que las otras cajas si el espacio en el contenedor se hace más pequeño.

La caja con la clase `box-4` tiene la propiedad `flex-basis` establecida en 200px, lo que significa que tendrá un tamaño base de 200 píxeles, antes de que se apliquen las propiedades de crecimiento o encogimiento.

La caja con la clase `box-5` tiene la propiedad `align-self` establecida en `flex-end`, lo que significa que se alineará en el extremo inferior del contenedor, mientras que las otras cajas se alinean en el centro.