**Introducción al posicionamiento de elementos**

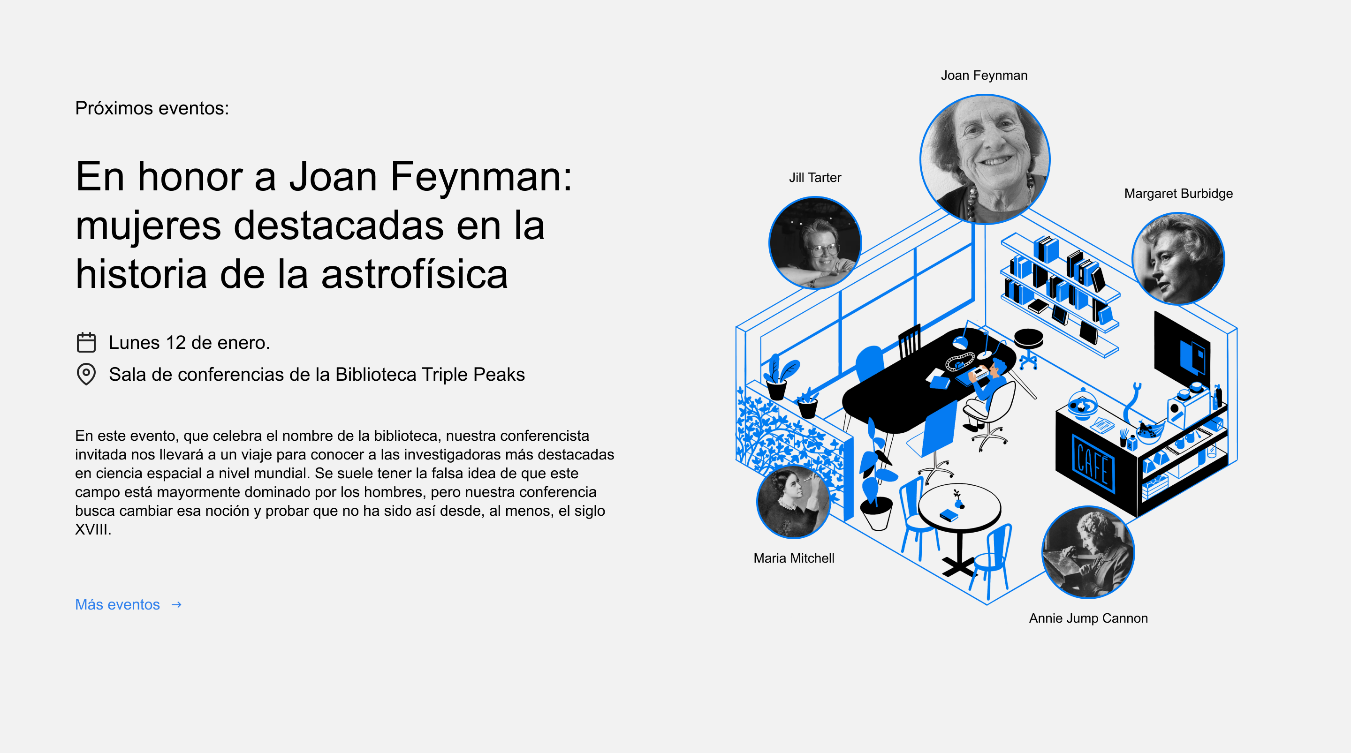
Al comienzo del capítulo sobre BEM, comparamos la forma en la que esta metodología organiza nuestro código con la forma en que se organizan los libros de una biblioteca mediante índices. Imagina que vas a una biblioteca y que te diriges hacia la ubicación de un libro que has localizado con la ayuda de un índice. Caminas hasta el pasillo indicado y encuentras el libro en la estantería esperada. Sin embargo, la escena con la que te encuentras no es la ideal. Si bien los libros están ordenados correctamente en la estantería, están colocados de una forma visualmente caótica. Algunos están con el lomo hacia atrás, otros están al revés y otros de lado.

Es decir que, aunque el sistema de índices funciona, no es suficiente para que nuestras estanterías se vean bien. De forma similar, mantener tu código bien organizado no es suficiente para hacer que los elementos de tu página aparezcan donde quieres. Para lograr eso, debemos conocer el concepto clave del posicionamiento de elementos, que es de lo que trata este capítulo.

Todos los elementos de una página están interconectados y reaccionan a los cambios que se producen en su entorno. Ese comportamiento a veces puede traer complicaciones, y usar flexbox para lidiar con ellas puede ser difícil. Lo que puede ayudarnos es controlar el posicionamiento de los elementos interconectados en el flujo. Existen varios tipos de posicionamiento, en las siguientes lecciones te explicaremos cómo funcionan.

En este capítulo conocerás la propiedad position y aprenderás a posicionar elementos en una página en el lugar que quieras, a crear menús con posiciones fijas, a centrar ventanas emergentes en la pantalla y a usar efectos hover que funcionan incluso con videos.

Después, tendrás la oportunidad de practicar lo aprendido completando la segunda etapa de tu proyecto para este sprint, donde crearás una composición de varias imágenes en diferentes posiciones, como se muestra a continuación.



# Flujo de documentos y posicionamiento estático

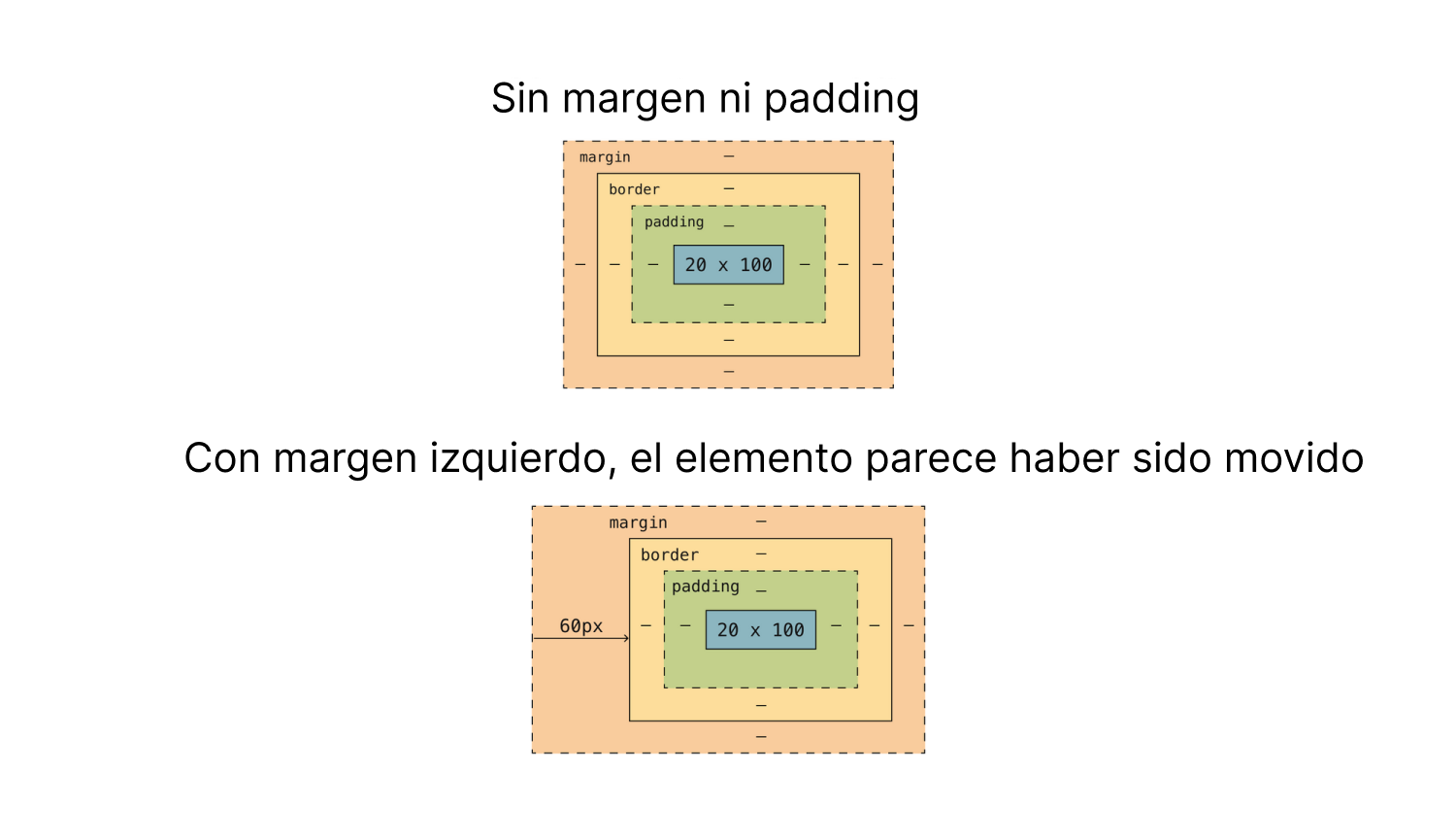
Los navegadores procesan y organizan el código HTML de acuerdo con una serie interna de reglas. Cada bloque se coloca en una nueva línea, mientras que los elementos de línea se agrupan juntos. Los objetos no se superponen entre sí, es decir que, si aplicas un margen a un objeto, los elementos que lo rodean serán alejados en consecuencia. Esto nos muestra que todos los elementos de nuestra página están interconectados. Es como una habitación llena de personas que se pueden ver y reaccionar ante las acciones de los demás.

Esta conexión entre elementos y su interdependencia es conocida como flujo normal (o flujo del documento).

## Flujo normal

Por defecto, todos los elementos en una página son parte del flujo, aunque esto puede cambiarse con la propiedad position de un elemento, como verás más adelante en este capítulo. Pero, antes de llegar a ello, vamos a describir a detalle el comportamiento de los elementos en el flujo normal. Estos son algunos puntos a tener en cuenta:

* Si un elemento es parte del flujo, entonces todos los demás elementos en el flujo sabrán dónde están y reaccionarán en consecuencia cuando el elemento se reposiciona.
* Los elementos en el flujo, en realidad, no pueden reposicionarse. Pero podemos hacer que *parezca* que se han reposicionado si aplicamos margin o padding.
* Al aplicar valores positivos de margin o padding a un elemento, de hecho estás aumentando el tamaño de la parte correspondiente de la caja contenedora del elemento. Esto hace *parecer* que el elemento se ha movido cuando en realidad sólo ha crecido.

Un elemento parece haberse movido al aplicársele el margen de posición.

* Aplicar valores negativos de margin crea el efecto de aparentemente cambiar el contenido del elemento en la dirección opuesta, causando que potencialmente los elementos contiguos se superpongan.

## Posición estática

Por defecto, a todos los elementos en un documento se les da position: static, así que decimos que están posicionados estáticamente. Un elemento posicionado estáticamente es, simplemente, una parte del flujo normal, tal como se describió anteriormente. En las siguientes lecciones, examinaremos los otros valores que la propiedad position puede tomar y cómo esta afecta el posicionamiento de un elemento dentro (o fuera) del flujo normal del documento.

**Posicionamiento relativo**

Cuando la posición de un elemento en el flujo se establece con respecto a su posición normal, se dice que tiene un posicionamiento relativo. Esto se consigue asignando la declaración position: relative; al elemento.

Estas son algunas cosas importantes que debes recordar con respecto a este tipo de posicionamiento:

* Dado que el elemento permanece en el flujo, los demás elementos seguirán reconociendo sus bordes
* El elemento no se mueve en relación a sus elementos vecinos como lo haría si tuviera un posicionamiento estático, sino que se mueve en relación a su propia posición inicial, de este modo:

top: desde el borde superior del bloque

right: desde el borde derecho

bottom: desde el borde inferior

left: desde el borde izquierdo

Estas propiedades no tienen la misma prioridad. La propiedad del borde superior top tiene una prioridad más alta que la del borde inferior; y la propiedad del borde izquierdo left tiene una prioridad más alta que la del borde derecho. No hace falta que uses 3 o las 4 de estas propiedades, puede ser confuso y no es necesario. Usaremos siempre dos atributos, uno vertical y otro horizontal:

div {

position: relative;

top: 15px;

left: 40px;

}

div {

position: relative;

bottom: 20px;

right: 30px;

}

# Posicionamiento fijo

Existe otro tipo de posicionamiento llamado posicionamiento fijo, que se aplica utilizando la declaración position: fixed.

Lo que diferencia a este tipo de posicionamiento de los demás es que saca a los elementos del flujo. Como resultado, el espacio en el que estaba será ocupado por sus vecinos, mientras que el elemento fijo cubrirá otros elementos colocándose por encima de ellos.

Este elemento es fijo en relación con la ventana del navegador:

div {

position: fixed;

top: 10px;

right: 40px;

}

Este código colocará el elemento en la esquina superior derecha de la ventana, a 10px por debajo del borde superior y 40px a la derecha del borde derecho. Cuando un usuario se desplaza hacia abajo, este elemento fijo conservará su posición en la parte superior de la pantalla. Podemos imaginarlo como un barco anclado en medio de un río. Aunque el río fluye a su alrededor, el barco permanece en su lugar.

Al igual que en el [posicionamiento relativo](https://tripleten.com/trainer/web/lesson/280231c1-fbb2-4327-a619-96ff0983568d), la esquina superior izquierda también tiene prioridad en el posicionamiento fijo: las propiedades top y left tienen prioridad sobre bottom y right.

Si no se especifican propiedades de posicionamiento para elementos con position: fixed, estos pueden quedar fuera de la pantalla. Por eso, debes asegurarte de establecer siempre la posición de los elementos que no están incluidos en el flujo.

# Posicionamiento absoluto

El último tipo de posicionamiento que aprenderemos se llama posicionamiento absoluto. Es uno de los más comunes, y se aplica mediante la declaración position: absolute.

Lo primero que debes saber es que los elementos con posicionamiento absoluto también son extraídos del flujo, dejando un espacio que ocupan los elementos contiguos.

Sin embargo, a diferencia del posicionamiento fijo, los elementos se comportan de forma diferente según la situación.

El primer caso se da cuando el elemento padre no tiene un posicionamiento específico asignado (es decir, tiene la declaración predeterminada position: static). En esos casos, nuestro bloque recurrirá al siguiente padre de la jerarquía e intentará posicionarse en relación con sus bordes.

Si todos los elementos padre tienen un posicionamiento predeterminado, el bloque se posicionará en relación con los bordes del documento («body»). En otras palabras, si nuestro elemento no encuentra algo a lo que adherirse, la siguiente pieza de código colocará el bloque 30px debajo del borde superior y 40px a la derecha del borde izquierdo del cuerpo:

div {

position: absolute;

top: 30px;

left: 40px;

}

A diferencia de lo que sucedía con el posicionamiento fijo, cuando te desplaces por la página, el elemento no permanecerá en la pantalla, ya que está posicionado en relación con el cuerpo de la página y no en relación con la ventana del navegador.

El segundo caso se da cuando el padre tiene una posición definida, es decir, cuando el valor de su propiedad position no es static.

En este caso, nuestro bloque con posicionamiento absoluto se ubicará en relación con los bordes del elemento padre, de acuerdo con los valores top, right, bottom y left especificados.

Asumiendo que los selectores del siguiente ejemplo corresponden a un elemento padre y su hijo, el código CSS colocará al bloque child en la esquina inferior derecha del bloque parent:

.parent {

position: relative;

}

.child {

position: absolute;

bottom: 0;

right: 0;

}

Aplica diferentes valores al siguiente widget y observa cómo se comporta el bloque amarillo dependiendo de si su padre (el bloque azul) tiene o no tiene un posicionamiento relativo. Al probar valores opuestos como top y bottom o left y right, recuerda que solo se aplica el valor con mayor prioridad cuando hay un conflicto. Por ejemplo, si se aplican top y bottom simultáneamente, top tendrá prioridad.

# Posicionamiento sticky

Ya sabes cómo utilizar cuatro de los cinco valores position: static, relative, fixed y absolute. Ahora nos enfocaremos en el último: position: sticky.

sticky es un híbrido entre el posicionamiento relativo y el fijo. Un elemento se comporta con posicionamiento relative hasta que cruza un umbral específico, en cuyo punto su posicionamiento se vuelve fijo.



Digamos que tenemos un encabezado que queremos "adherir" en la parte superior del viewport cuando se desplace el usuario. Para hacer que este encabezado "se adhiera", podemos utilizar el siguiente CSS:

.nav {

position: sticky;

top: 0;

}

top: 0 significa que el encabezado se va a adherir a la parte superior del viewport cuando te desplaces hacia abajo.

Cabe destacar que el elemento sólo se fijará si su padre también se desplaza. Si el padre no se desplaza (es decir, el elemento sticky está en la raíz del documento HTML), el elemento sticky no se va a adherir.

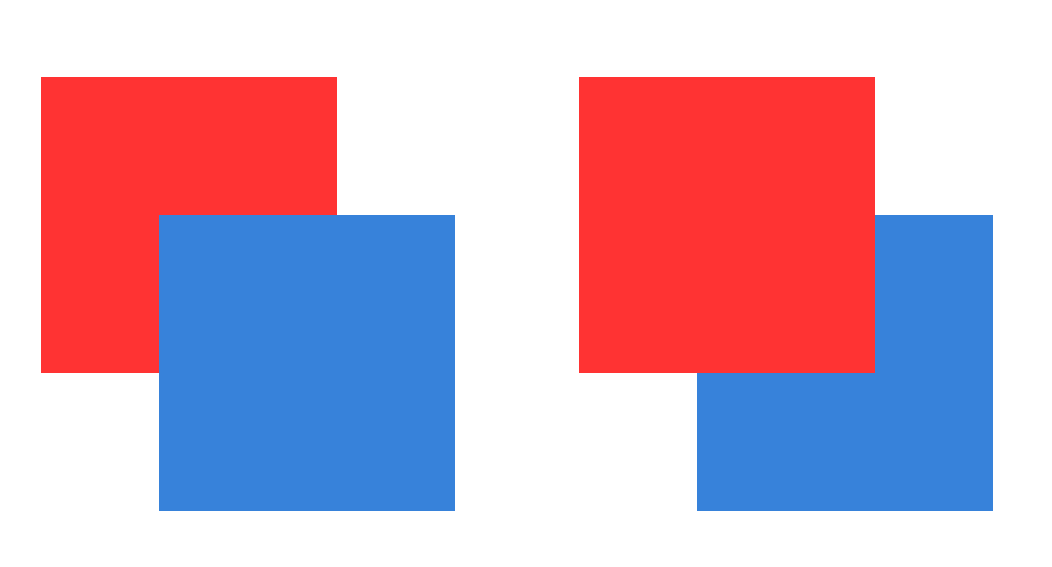
Además, el elemento sticky "se adhiere" a su ancestro más cercano que tenga un "mecanismo de desplazamiento" (creado cuando el desbordamiento es hidden, scroll, auto u overlay), incluso cuando dicho ancestro no sea su ancestro de desplazamiento más cercano.

position: sticky es una buena manera de mantener información importante (como una barra de navegación o una sección destacada de la página) visible todo el tiempo para el usuario. ¡Vamos a practicar!

# z-index

Cuando extraemos elementos del flujo normal para posicionarlos, a veces se superponen entre sí. Necesitamos algo que nos ayude a determinar el orden en el que lo harán.

Ejemplo:



Utilizamos la propiedad z-index para gestionar el orden de aparición de los elementos en la pantalla. En lugar de definir la posición de los elementos de forma horizontal (eje x) o vertical (eje y), la propiedad z-index lo hace en términos de profundidad, a lo largo del eje z. Como su nombre sugiere, estamos trabajando con una tercera dimensión. Los valores de esta propiedad se indican con números enteros y todos los elementos tienen una propiedad z-index con valor 0 de forma predeterminada.

Ejemplo:

.background {

z-index: -1;

}

.first-block {

z-index: 0;

}

.second-block {

z-index: 1;

}

.third-block {

z-index: 99;

}

.fourth-block {

z-index: 100;

}

En este código, el bloque background es el más lejano. El resto de los bloques aparecen de acuerdo con el valor de su z-index. El de valor más alto, en este caso el bloque .fourth-block, es el que aparece encima de toda la pila.

La propiedad z-index no se puede aplicar a elementos con posicionamiento estático ni a los que carecen de una propiedad position. Sin embargo, es importante entender que en el contexto de Flexbox (cuando el elemento padre tiene display: flex), la propiedad z-index aún se aplica solo a elementos que no tienen position: static. Es decir, un elemento con position: static dentro de un contenedor Flexbox no responderá a la propiedad z-index. Por lo tanto, para que z-index tenga efecto en un contenedor Flexbox, el elemento debe tener un valor de posición distinto de static, como relative, absolute, fixed, o sticky.

# Posicionamiento de elementos: conclusión

Ya sabes lo que es el flujo de documentos y cómo se pueden comportar los elementos. Puedes quitar elementos del flujo, posicionar elementos de tal modo que cubran otros elementos y moverlos en relación con la ventana del navegador u otras partes de la página. Has aprendido a colocar un elemento en cualquier lugar que desees, incluso encima o detrás de otro.

La mayoría de profesionales de desarrollo web usan el posicionamiento absolute para implementar algunos detalles en el diseño. Puedes usar este tipo de posicionamiento en tus propios proyectos, aunque no debes excederte. Es mejor atenerse al flujo de documentos. Si puedes colocar los elementos de la forma que deseas mediante un diseño bien pensado, prácticamente no necesitarás usar el posicionamiento absoluto o fijo.

Hemos analizado todos los valores de la propiedad position: static, relative, absolute y fixed. En realidad, existe uno más, llamado sticky, pero no hemos hablado de él aquí porque aún no está ampliamente admitido. Hemos recopilado para ti un poco de información de fuentes externas acerca del valor sticky, así como de las propiedades position y z-index:

* [Un manual sobre la propiedad posición](https://developer.mozilla.org/es/docs/Web/CSS/position)
* [Documentación oficial de la propiedad 'position'](https://www.w3.org/TR/css-position/) (artículo en inglés).
* [Un ejemplo del uso de la propiedad z-index](https://medium.com/hackernoon/my-approach-to-using-z-index-eca67feb079c) (artículo en inglés).

¡Felicidades, has completado un capítulo más! En los próximos capítulos aprenderás a escribir tu propio código. Al finalizar, comenzarás tu primer proyecto. ¡Darás un gran paso hacia el desarrollo web real y crearás una página de destino!

### Consejo para la planificación de tu carrera

Anteriormente, hablamos de los diferentes caminos que puedes tomar como profesional de desarrollo web según tus intereses y te invitamos a considerar qué tipo de trabajo te gustaría hacer en el futuro. Veamos ahora las cosas desde otro ángulo. ¿Cuál sería tu entorno de trabajo ideal? Para responder a esta pregunta, debes pensar en tus prioridades.

Prioridad A — Estabilidad: Cuando comienzas a trabajar para una gran empresa, es esperable avanzar gradualmente a un puesto de nivel medio y luego a un nivel superior. Esto llevará un tiempo, pero es una elección profesional confiable. Piénsalo como si fuera una caminata cuesta arriba por una suave pendiente a lo largo de un sendero amplio y agradable. ¡Lento pero constante se gana la carrera!

Prioridad B — Remuneración elevada: Si estás preparado(a) para negociar constantemente y crees que podrás proporcionar referencias excelentes por parte de empleadores y clientes, la opción de cambiar regularmente de lugar de trabajo para obtener el puesto mejor pagado podría ser el camino a seguir. Lo mismo se aplica a trabajar como consultor(a) o colaborador(a) independiente. Es una estrategia de alto riesgo y alta recompensa, que se parece más a escalar una montaña de pendiente pronunciada. Si te gustan las emociones fuertes, considera esta opción.

# Cuestionario: Posicionamiento de elementos

¡Felicidades, has llegado al final de otro capítulo! Hemos preparado un cuestionario para asegurarnos de que hayas comprendido todo el material. Consta de tres secciones:

* Flujo normal del documento (3 preguntas).
* La propiedad position (3 preguntas).
* z-index (2 preguntas).

Estos cuestionarios, al igual que las revisiones de código, buscan ayudarte a aprender y garantizar que realmente hayas comprendido todos los materiales.

Recuerda que este cuestionario es "a libro abierto". Puedes consultar las lecciones anteriores o utilizar el motor de búsqueda de tu elección. Como profesional de desarrollo web, no tienes que saber todas las respuestas de memoria. ¡Es más importante saber cómo hallar las respuestas a las preguntas cuando estas surjan!

Cuando quieras, haz clic en el botón de abajo. ¡Que comience la fiesta!

El cuestionario contiene 8 preguntas y deberías tardar alrededor de 15 minutos en completarlo.

## Flujo normal del documento (3 preguntas).

Pregunta

1. ¿Qué afirmaciones sobre el flujo normal del documento son verdaderas?

Elige tantas como quieras

El flujo normal del documento controla el posicionamiento de todos los elementos HTML en un documento.

El flujo normal del documento controla el posicionamiento de todos los elementos HTML posicionados estáticamente en un documento.

¡Correcto! Uno de los principales usos de la propiedad position es eliminar elementos del flujo normal.

En el flujo normal del documento, los elementos de bloque se colocan en líneas nuevas, mientras que los contenedores flexibles se agrupan en la misma línea.

En el flujo normal del documento, los elementos de bloque se colocan en líneas nuevas, mientras que los elementos en línea se agrupan en la misma línea.

¡Correcto! Puedes cambiar elementos de bloque a elementos en línea y viceversa, usando la propiedad display.

¡Bien hecho!

Pregunta

2. Verdadero o falso. Un elemento posicionado de manera estática puede verse como si se hubiese reposicionado usando las propiedades **top**, **right**, **bottom** y **left**.

Verdadero

Falso

¡Tienes razón! Esto es falso. Los elementos con position: static no se ven afectados por las propiedades top, right, bottom y left.

¡Bien hecho!

Pregunta

3. Verdadero o falso. Los elementos posicionados estáticamente nunca pueden superponerse uno arriba de otro.

Verdadero

Falso

Estás en lo correcto. Esta afirmación es falsa. Los elementos con position: static se pueden superponer si aplicamos márgenes negativos.

¡Excelente trabajo!

## La propiedad position (3 preguntas).

Pregunta

4. ¿Qué valor de **position** usarías para hacer que un elemento siempre aparezca en el mismo lugar dentro de la ventana del navegador?

static

relative

fixed

¡Correcto! Esta suele usarse con ventanas emergentes para asegurarnos de que sean visibles siempre que se abran.

absolute

¡Buen trabajo!

Pregunta

5. ¿Qué valor de **position** coloca inicialmente los elementos según el flujo normal del documento, pero nos permite reposicionar el elemento con respecto a sí mismo, empleando las propiedades **top**, **left**, **bottom** y **right**?

static

relative

Correcto. Un elemento posicionado relativamente, se coloca inicialmente según el flujo normal del documento, y aunque el elemento puede desplazarse desde su posición inicial, sus vecinos se comportarán como si no se hubiese movido.

fixed

absolute

¡Excelente trabajo!

Pregunta

6. Verdadero o falso. Los elementos posicionados absolutamente se posicionan con respecto a su contenedor padre inmediato.

Verdadero

Falso

¡Correcto! Esta afirmación es falsa. Un elemento posicionado de manera absoluta estará posicionado con respecto a su ancestro más cercano no posicionado estáticamente. Si todos sus ancestros están posicionados de manera estática, se posicionará con respecto a la ventana gráfica.

¡Lo has entendido bien!

## z-index (2 preguntas).

Pregunta

7. La propiedad **z-index** sólo funciona en algunos tipos de elementos. Selecciona los criterios que permitirán que **z-index** afecte al elemento.

Elige tantas como quieras

El elemento tiene position: static.

El elemento tiene una position diferente a static.

Correcto. Los elementos con position: static no se ven afectados por z-index

Este elemento tiene display: flex.

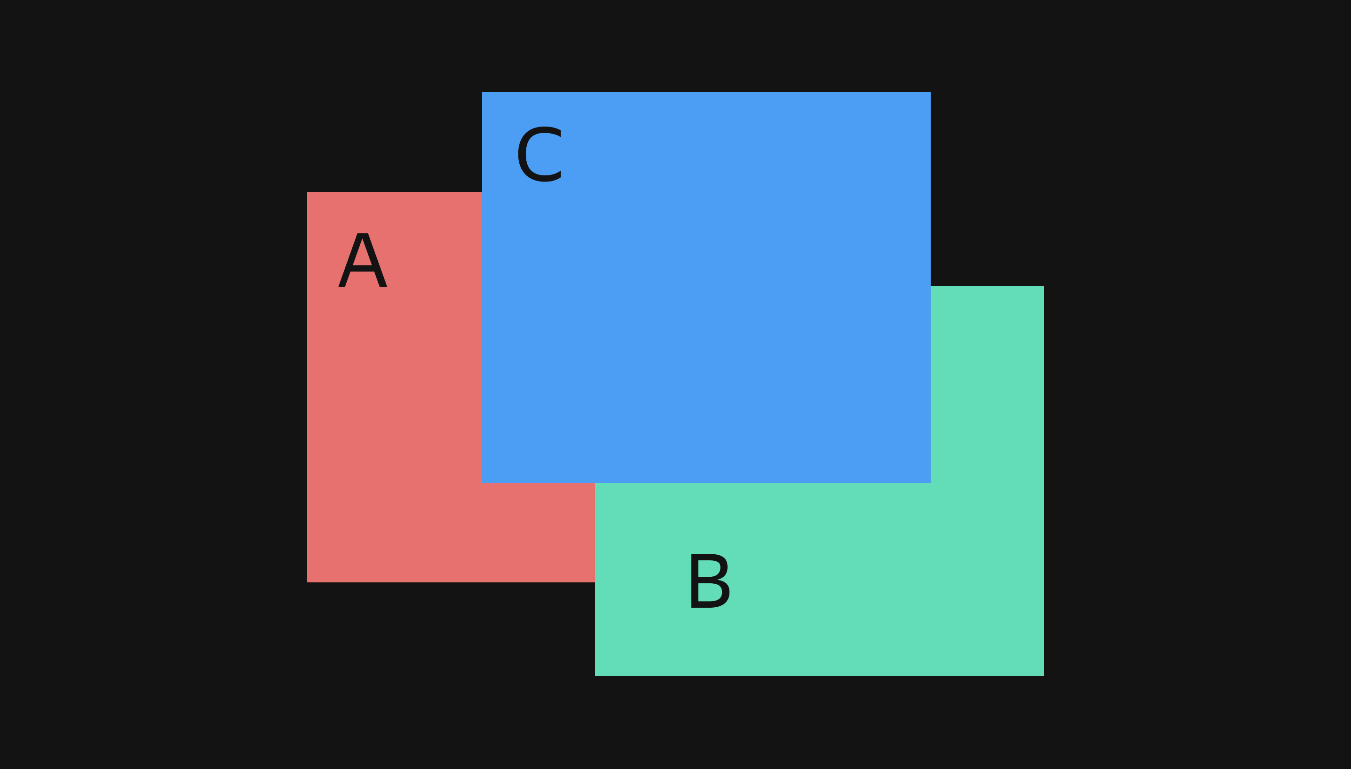
Este es un elemento flexible y tiene una position diferente a static.

Correcto. z-index afecta a los elementos posicionados y determina su orden de apilamiento en relación con otros elementos posicionados.

¡Bien hecho!

Pregunta

8. Imagina que tienes tres elementos dentro de un contenedor flexible, y quieres apilarlos como se muestra:



¿Cuál de los siguientes conjuntos de valores **z-index** *no* producirán este diseño?

A: z-index: -1, B: z-index: 0, C: z-index: 1

A: z-index: -1, B: default, C: z-index: 1

A: z-index: 1, B: default, C: z-index: 2

Estás en lo correcto. Este es el que no funcionará. Por defecto, los elementos apilables tienen un nivel de apilación de 0, así que la segunda caja se colocará en la parte inferior.

A: z-index: 1, B: z-index: 2, C: z-index: 3