



Ejes

Curso de Fundamentos

Introducción

Veamos cómo se puede controlar la orientación de los elementos dentro de un contenedor flexible usando la `flex-direction` propiedad.

Resumen de la lección

Esta sección contiene una descripción general de los temas que aprenderá en esta lección.

- Aprenderá sobre los 2 "ejes" de un contenedor flexible.
- Aprenderá cómo cambiar esos ejes para organizar su contenido en columnas en lugar de filas.

Ejes

Lo más confuso de flexbox es que puede funcionar tanto en horizontal como en vertical, y algunas reglas cambian un poco según la dirección en la que trabajes. La dirección predeterminada para un contenedor flexible es horizontal o `row`, pero puedes cambiar la dirección a vertical o `column`. La dirección se puede especificar en CSS de la siguiente manera:

```
1 | .flex-container {  
2 |   flex-direction: column;  
3 | }
```

```
> | }
```

Independientemente de la dirección que utilices, debes pensar en tus contenedores flexibles como si tuvieran 2 ejes: el eje principal y el eje transversal. Es la dirección de estos ejes la que cambia cuando se `flex-direction` cambia . En *la mayoría de los casos*, `flex-direction: row` coloca el eje principal horizontal (de izquierda a derecha) y `column` vertical (de arriba a abajo).

En otras palabras, en nuestro primer ejemplo, pusimos `display: flex` un div y este organizó sus elementos secundarios horizontalmente. Esta es una demostración de `flex-direction: row`, la configuración predeterminada. El siguiente ejemplo es muy similar. Si descomentamos la línea que dice `flex-direction: column`, esos divs se apilarán verticalmente.

HTML

CSS

Result

EDIT ON

LIVE

```
<div class="flex-container">
  <div class="one"></div>
  <div class="two"></div>
  <div class="three"></div>
</div>
```

Resources1x0.5x0.25xRerun

Una cosa a tener en cuenta es que en este ejemplo, `flex-direction: column` no funcionaría como se esperaba si usáramos la abreviatura `flex: 1`. Pruébalo ahora (es decir, vaya a cambiar el valor de flex en la `flex: 1 1 auto;` línea). ¿Puede averiguar por qué no funciona si `flex: 1` se usa ? Los divs colapsan, aunque *claramente* tienen un `height` definido allí.

El motivo es que la abreviatura flex se expande `flex-basis` a `0`, lo que significa que todos `flex-grow` los ing y `flex-shrink` ing comenzarían sus cálculos desde `0`. Los divs vacíos tienen una altura de 0 de forma predeterminada, por lo que para que

nuestros elementos flex llenen la altura de su contenedor, en realidad no necesitan tener ninguna altura.

El ejemplo anterior solucionó este problema especificando `flex: 1 1 auto`, indicando a los elementos flexibles que utilicen de forma predeterminada su valor dado `height`. También podríamos haberlo solucionado colocando una altura en el elemento principal `.flex-container` o utilizando `flex-grow: 1` en lugar de la abreviatura.

Otro detalle a tener en cuenta: cuando cambiamos la dirección flexible a `column`, `flex-basis` hace referencia a `height` en lugar de `width`. Dado el contexto, esto puede resultar obvio, pero es algo que hay que tener en cuenta.

Nos hemos desviado un poco del tema... Estábamos hablando de flex-direction y ejes. Para volver al punto principal, el comportamiento predeterminado es `flex-direction: row` que organiza las cosas horizontalmente. La razón por la que esto a menudo funciona bien sin cambiar otros detalles en el CSS es porque los elementos a nivel de bloque tienen por defecto el ancho completo de su elemento principal. Cambiar las cosas a vertical `flex-direction: column` agrega complejidad porque los elementos a nivel de bloque tienen por defecto la altura de su contenido y, en este caso, no hay *contenido*.

"Hay situaciones en las que el comportamiento de flex-direction podría cambiar si estás usando un idioma que se escribe de arriba a abajo o de derecha a izquierda, pero debes dejar de preocuparte por eso hasta que estés listo para comenzar a crear un sitio web en árabe o hebreo."

Comprobación de conocimientos

Las siguientes preguntas son una oportunidad para reflexionar sobre temas clave de esta lección. Si no puede responder una pregunta, haga clic en ella para revisar el material, pero tenga en cuenta que no se espera que memorice o domine este conocimiento.

- [¿Cómo hacer que los elementos flexibles se organicen verticalmente en lugar de horizontalmente?](#)
- [En un `column` contenedor flexible, ¿a qué `flex-basis` se refiere?](#)
- [En un `row` contenedor flexible, ¿a qué `flex-basis` se refiere?](#)
- [¿Por qué las dos preguntas anteriores tienen respuestas diferentes?](#)

Recursos adicionales

Esta sección contiene enlaces útiles a contenido relacionado. No es obligatoria, por lo que se la puede considerar complementaria.

- Esta [hoja de referencia visual de flexbox](#) tiene algunas referencias útiles sobre flex y sus propiedades.
- Para ver una demostración interactiva, consulte este [Scrim sobre ejes Flexbox](#) . Tenga en cuenta que para poder ver este Scrim es necesario iniciar sesión en Scrimba.

[!\[\]\(23d9fc146e83b5c3013cfa32c784f8d5_img.jpg\) Mejorar en GitHub](#)[!\[\]\(c694a3ff3b077d76910920a6a1593ab4_img.jpg\) Informar un problema](#)[Ver registro de cambios de la lección](#)[Ver curso](#)[Marcar como completado](#)[Próxima lección](#)

¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únase a nosotros para ayudar a estudiantes de todo el mundo apoyando el Proyecto Odin!

[Más información](#)[Dona ahora](#)



THE ODIN PROJECT

Educación en codificación de alta calidad mantenida por una comunidad de código abierto.



Sobre nosotros

[Acerca de](#)

[Equipo](#)

[Blog](#)

[Casos de éxito](#)

Apoyo

[Preguntas frecuentes](#)

[Contribuir](#)

[Contáctenos](#)

Guías

[Guías de la comunidad](#)

[Guías de instalación](#)

Legal

[Términos](#)

[Privacidad](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.