Alineación

Curso de Fundamentos

Introducción

Hasta ahora, todo lo que hemos hecho con flexbox ha utilizado la regla `flex: 1` de todos los elementos flexibles, lo que hace que los elementos crezcan o se encojan de manera uniforme para llenar todo el espacio disponible. Sin embargo, muy a menudo, este no es el efecto deseado. Flex también es muy útil para organizar elementos que tienen un tamaño específico.

Resumen de la lección

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

- Aprenderá a alinear elementos dentro de un contenedor flexible tanto vertical como horizontalmente.

Alineación

Veamos un ejemplo.

HTML CSS Result EDIT ON

LIVE

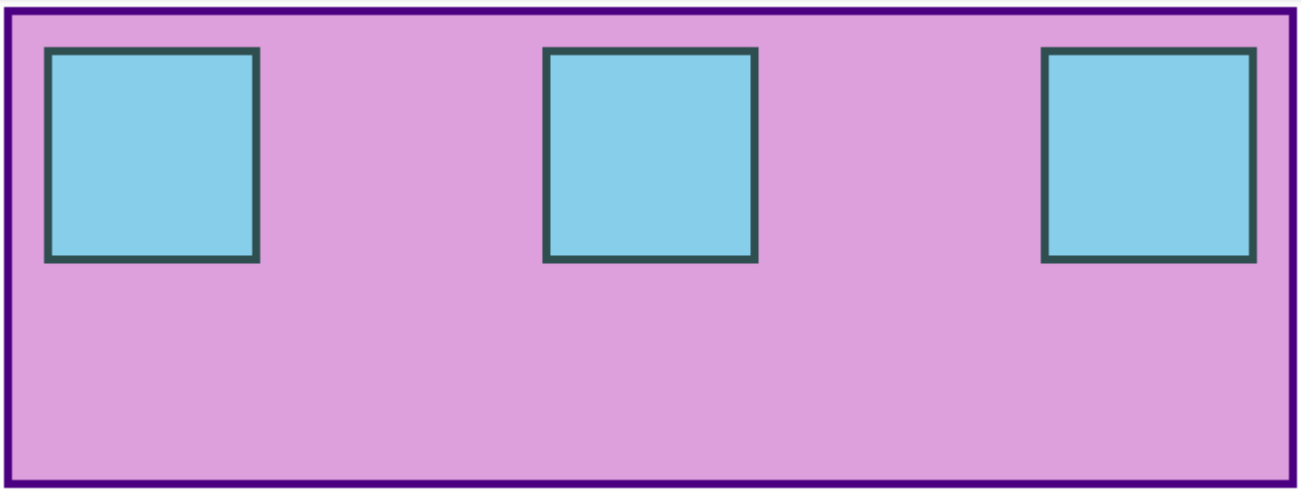
```
<div class="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

Resources 1× 0.5× 0.25× Rerun

Ya deberías poder predecir lo que sucederá si te pones `flex: 1` esto `.item`. ¡Pruébalo antes de continuar!

`flex: 1` Al sumar, `.item` cada uno de los elementos crece hasta llenar el espacio disponible, pero ¿qué pasa si queremos que mantengan el mismo ancho, pero que se distribuyan de forma diferente dentro del contenedor? ¡Podemos hacerlo!

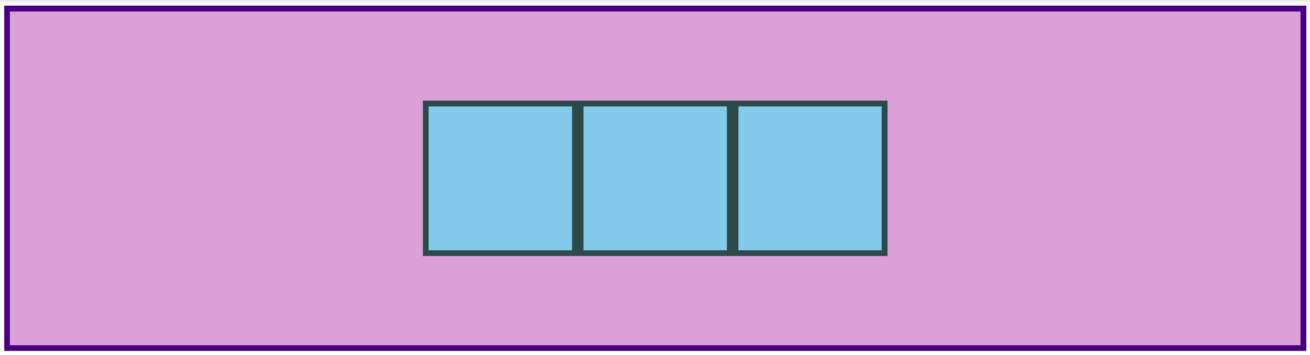
Quitar `flex: 1` de `.item` y agregar `justify-content: space-between` a `.container`. Al hacerlo, debería obtener algo como esto:



`justify-content` alinea los elementos a lo largo del **eje principal**. Hay algunos valores que puedes usar aquí. Aprenderás el resto en las tareas de lectura, pero por ahora

intenta cambiarlo a `center`, lo que debería centrar los cuadros a lo largo del eje principal.

Para cambiar la ubicación de los elementos a lo largo del **eje transversal**, utilice `align-items`. Intente colocar los cuadros en el centro del contenedor agregando `align-items: center` a `.container`. El resultado deseado se ve así:



Debido a que `justify-content` y `align-items` se basan en el eje principal y transversal de su contenedor, su comportamiento cambia cuando cambia la dirección flexible de un contenedor flexible. Por ejemplo, cuando cambia `flex-direction` a `column`, `justify-content` se alinea verticalmente y `align-items` se alinea horizontalmente. Sin embargo, el comportamiento más común es el predeterminado, es decir, `justify-content` alinea los elementos horizontalmente (porque el eje principal está predeterminado en horizontal) y `align-items` los alinea verticalmente. Uno de los mayores problemas que tienen los principiantes con flexbox es la confusión cuando este comportamiento cambia.

Brecha

Una característica muy útil de flex es la `gap` propiedad. La configuración `gap` en un contenedor flexible agrega un espacio específico entre los elementos flexibles, de manera similar a agregar un margen a los elementos mismos. `gap` es una propiedad *nueva*, por lo que aún no aparece en muchos recursos, pero funciona de manera confiable en todos los navegadores modernos, por lo que es segura de usar y muy útil. Al agregarla `gap: 8px` al ejemplo centrado anterior, se obtiene el resultado que se muestra a continuación.

HTML CSS Result EDIT ON

LIVE

```
<div class="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

Resources 1× 0.5× 0.25× Rerun

Hay más cosas que aprender en la siguiente lectura, pero en este punto seguramente podrá ver lo inmensamente útil que es flexbox. ¡Con solo las propiedades que ya cubrimos, ya podría armar algunos diseños impresionantes!

Tómate tu tiempo para leer la información. Habrá algunos repasos de los elementos que ya hemos tratado aquí, pero se profundiza más y se tocan algunas cosas que aún no se han mencionado. No te estreses demasiado por intentar memorizar cada pequeño detalle todavía; simplemente codifica junto con los ejemplos y haz lo mejor que puedas para internalizar todo lo que es *posible* con flexbox. Tendrás que recurrir a estos recursos nuevamente una vez que llegues a los ejercicios de práctica, pero eso es perfectamente aceptable. Cuanto más uses este material, mejor se te quedará en la mente... y lo usarás *constantemente* . ¡Diviértete!

Asignación

1. Esta hermosa guía interactiva de Flexbox cubre todo lo que necesitas saber. Te ayudará a reforzar los conceptos que ya hemos abordado con algunos ejemplos realmente divertidos y creativos. Dedica algo de tiempo a leerla, ya que en este punto deberías repasar parte de ella, pero los conceptos básicos son importantes.
2. Casos de uso típicos de Flexbox es un artículo de MDN que cubre algunos consejos más prácticos. ¡No te saltes las secciones interactivas! ¡Jugando con estas cosas es como aprenderás!

3. La "Guía de Flexbox" de CSS Tricks es un clásico. Las imágenes y los ejemplos son muy útiles. Sería una buena idea revisar las partes 1 a 3 y la parte 5 (no te preocupes por las partes de consulta de medios, las cubriremos más adelante en el curso) y luego guardarla como una excelente hoja de referencia para el futuro (tenla a mano para los ejercicios de práctica).
4. Realiza los ejercicios en el directorio de nuestro repositorio de ejercicios CSS `foundations/flex` (recuerda que las instrucciones están en el README) en el orden:
   - `01-flex-center`
   - `02-flex-header`
   - `03-flex-header-2`
   - `04-flex-information`
   - `05-flex-modal`
   - `06-flex-layout`
   - `07-flex-layout-2`

   Nota: Las soluciones de estos ejercicios se pueden encontrar en la `solution` carpeta de cada ejercicio.

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.

- ¿Cuál es la diferencia entre `justify-content` y `align-items` ?
- ¿Cómo se usa flexbox para centrar completamente un div dentro de un contenedor flexible?
- ¿Cuál es la diferencia entre `justify-content: space-between` y `justify-content: space-around` ?

Recursos adicionales

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

- Flexbox Froggy es un divertido juego para practicar cómo mover cosas con flexbox.
- Flexbox Zombies es otra versión gamificada de Flexbox. Es gratuito, pero requiere una cuenta.
- El artículo Conceptos básicos de Flexbox en MDN es otro buen punto de partida. Contiene ejemplos útiles y secciones interactivas.
- Alinear elementos en un contenedor flexible profundiza más en el tema de los ejes y `align-items` vs. `justify-content`
- Este tutorial de Flexbox de freecodecamp es otro recurso decente.
- Flexbox Crash Course es un buen recurso de Traversy Media.
- Para ver demostraciones más interactivas, prueba Scrim en la `justify-content` propiedad y Scrim en la `align-items` propiedad . Ten en cuenta que para ver estos Scrims es necesario iniciar sesión en Scrimba.

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