

# Entendiendo Flexbox

**Todo lo que necesitas para conocerlo**

# CONTENIDO

- Introducción
- Propiedades del Flex-container
- Propiedades del Flex-item
- Absolute and Relative Flex
- Alineación Auto-margin
- Cambiando el Flex-direction
- Flexbox resuelto
- Construyendo un layout de App musica
- Diseño responsive
- Conclusión

# Introduction

Aprender Flexbox puede ser un dolor de cabeza. Para la mayoría de las personas, no es particularmente divertido al principio. Te desafía a repensar cómo lo has afrontado. con diseños en CSS.

No te preocupes. Realmente te explicaré todo lo que necesitas saber. Ese es el objetivo de este libro.

# ¿Por qué utilizar FlexBox?

CSS ha evolucionado mucho en los últimos años. A los diseñadores les encantó el introducción de filtros, transiciones y transformaciones. Pero algo estaba desaparecido. Algo que todos anhelábamos.

La creación de diseños de páginas inteligentes con CSS parecía haber persistido durante demasiado tiempo, y esto hizo que muchos de nosotros escribiéramos CSS hacky.

Siempre tuvimos que lidiar con flotadores, trucos para exhibir mesas y el consecuencias que trajeron. Si ha escrito CSS durante algún tiempo, Probablemente pueda relacionarse con esto. Y si no, ¡bienvenido a un mundo mejor!

# ¿Por qué utilizar FlexBox?

Parece que nuestras oraciones como diseñadores y desarrolladores front-end han finalmente ha sido escuchado. Esta vez, a lo grande.

Ahora todos podemos deshacernos de esos trucos de CSS. No más uso incesante de flotadores, pantallas de celdas de tabla.

Es hora de adoptar una sintaxis moderna y más limpia para crear inteligencia diseños. Bienvenido al modelo CSS Flexbox.

# ¿Qué es Flexbox?

Según la especificación, el modelo Flexbox proporciona una manera eficiente de diseñar, alinear y distribuir el espacio entre elementos dentro de su documento, incluso cuando la ventana gráfica y el tamaño de su elementos es desconocido y/o dinámico.

Ya sea que escribas CSS en tus sueños o recién estás comenzando, te sentirás como en casa.

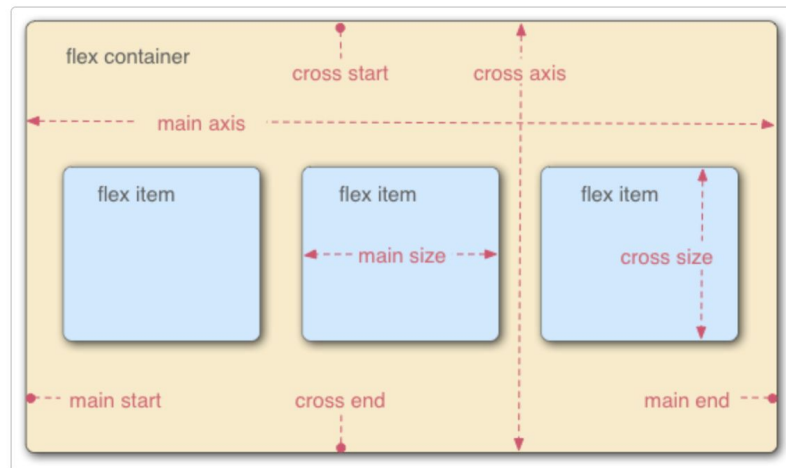
# Modelo de Flexbox

Técnicamente, "horizontal" y "vertical" no son las direcciones en el "mundo flexible".

Estos se describen como eje principal y eje transversal. Los valores predeterminados son se muestra a continuación.

Nuevamente en términos sencillos, la dirección predeterminada del eje principal parece "horizontal". De izquierda a derecha.

El eje transversal se siente como "vertical". De arriba a abajo.



# ¿Cómo empiezo a utilizar el modelo Flexbox?

Esta es la primera pregunta que todo el mundo se hace y la respuesta es mucho más sencilla de lo que esperabas.

Para comenzar a utilizar el modelo Flexbox, todo lo que necesita hacer es primero definir un contenedor flexible.

En HTML normal, diseñar una lista simple de elementos toma esta forma:

```
<ul> <!--elemento padre-->
  <li></li> <!--primer elemento hijo-->
  <li></li> <!--segundo elemento hijo-->
  <li></li> <!--tercer elemento hijo-->
</ul>
```



# ¿Cómo empiezo a utilizar el modelo Flexbox?

Lamarás a `ul` el elemento padre y a `li` el elemento hijo.

Para usar el modelo Flexbox, debe haber un elemento principal sea flexible contenedor (también conocido como contenedor flexible).

Para ello, configura `display: flex` o `display: inline-flex` para la variación en línea. Es así de simple y desde allí ya está todo listo para usar el modelo Flexbox.

Lo que realmente sucede es que un contexto de formato Flexbox es inmediatamente iniciado.

# ¿Cómo empiezo a utilizar el modelo Flexbox?

```
/*Hacer del elemento padre un contenedor flexible*/  
ul {  
    display: flex; /*o inline-flex*/  
}  
  
li {  
    width: 100px;  
    height: 100px;  
    background-color: #8cacea;  
    margin: 8px;  
}
```

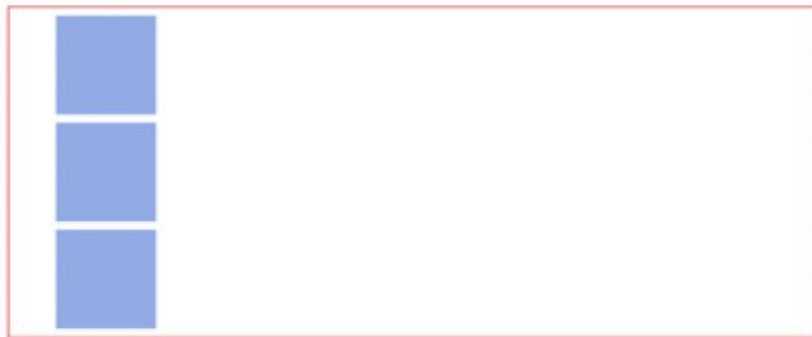
# ¿Cómo empiezo a utilizar el modelo Flexbox?



# ¿Cómo empiezo a utilizar el modelo Flexbox?

Quizás no lo hayas notado, pero algo ya pasó. El el contexto de formato Flexbox se ha iniciado.

Recuerda que, de forma predeterminada, los "divs" en CSS se apilan verticalmente, de arriba a abajo.



# Las propiedades del Flex container

Flex-direction

Flex-wrap

Flex-flow

Justify-content

Align-items

Align-content

---

# Flex-direction

La propiedad Flex-direction controla la dirección en la que los elementos flexibles se colocan a lo largo del eje principal.

```
/*donde ul representa un contenedor flexible*/  
ul {  
    flex-direction: row || column || row-reverse ||  
    Column-reverse;  
}
```

# Flex-wrap

La propiedad flex-wrap puede tomar cualquiera de estos tres valores:

*/\*donde ul representa un contenedor flexible\*/*

```
ul {  
    flex-wrap: wrap || nowrap || wrap-reverse;  
}
```

# Flex-wrap

Explicaré cómo funciona la propiedad flex-wrap guiándote a través de un ejemplo. Intentaré incluir muchos más elementos a la lista en la lista desordenada. ¿Qué piensas?

¿El contenedor flexible cambiará de tamaño para acomodar más o se dividirá?

¿Los elementos de la lista en otra línea?

```
/*agregando 3 elementos li más*/  
<ul> <!--elemento padre-»  
  <li></li> <!--primer elemento hijo-»  
  <li></li> <!--segundo elemento hijo-»  
  <li></li> <!--tercer elemento hijo-»  
  <li></li>  
  <li></li>  
  <li></li>  
</ul>
```



# Flex-wrap

Explicaré cómo funciona la propiedad flex-wrap guiándote a través de un ejemplo. Intentaré incluir muchos más elementos a la lista en la lista desordenada. ¿Qué piensas?

¿El contenedor flexible cambiará de tamaño para acomodar más o se dividirá?

¿Los elementos de la lista en otra línea?

```
/*agregando 3 elementos li más*/  
<ul> <!--elemento padre-»  
  <li></li> <!--primer elemento hijo-»  
  <li></li> <!--segundo elemento hijo-»  
  <li></li> <!--tercer elemento hijo-»  
  <li></li>  
  <li></li>  
  <li></li>  
</ul>
```


# Flex-flow

El flujo flexible es una propiedad abreviada que toma la dirección flexible y valores de envoltura flexible.

Múltiples valores declarados en una línea.

flex-direction

flex-wrap



```
ul {  
    flex-flow: row wrap; /*dirección 'fila' y sí, por favor  
    envuelva los artículos.*/  
}
```

# Justify-content

La vida es realmente buena con el modelo Flexbox. Si todavía dudas de eso, la propiedad justify-content puede convencerte.

La propiedad justify-content toma cualquiera de los 5 valores siguientes.

```
ul {  
    justify-content: flex-start || flex-end || center ||  
    space-between || space-around  
}
```

# Justify-content

La propiedad justificar contenido define cómo se distribuyen los elementos flexibles el eje principal.

Un ejemplo rápido.

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ul>
```

```
ul {
  border: 1px solid red;
  padding: 0;
  list-style: none;
  background-color: #e8e8e9;
}
li {
  background-color: #8cacea;
  width: 100px;
  height: 100px;
  margin: 8px;
  padding: 4px;
}
```

# Justify-content

Deberías tener esto.

## Flex-start

The default value is flex-start

```
ul {  
  justify-content: flex-start;  
}
```



# Justify-content

Deberías tener esto.

## Flex-start

The default value is flex-start

```
ul {  
  justify-content: flex-start;  
}
```

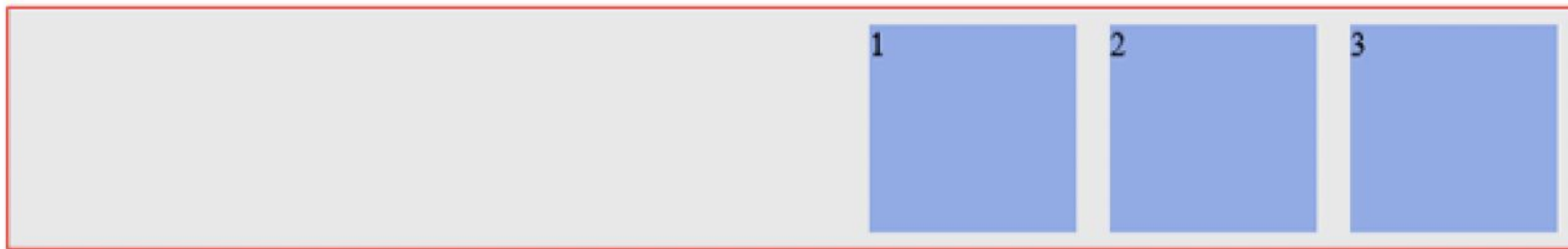


# Justify-content

## Flex-end

flex-end agrupa los elementos flexibles al final del eje principal.

```
ul {  
  justify-content: flex-end;  
}
```

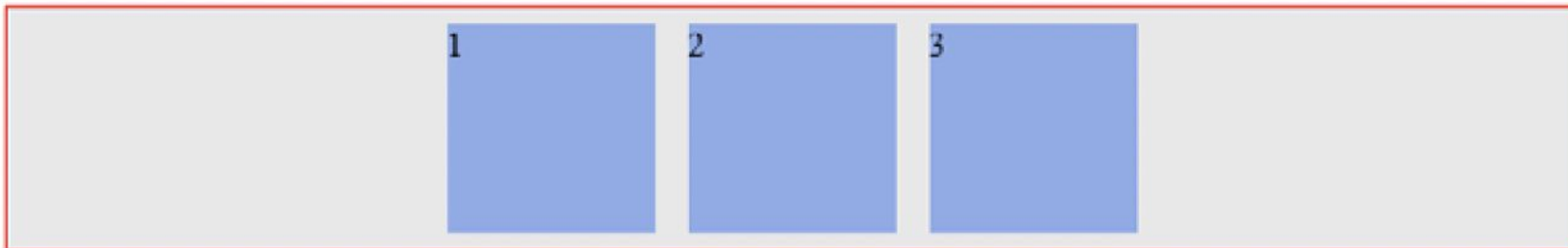


# Justify-content

## Center

Center hace exactamente lo que esperarías. Centra los elementos flexibles a lo largo del eje principal.

```
ul {  
  justify-content: center;  
}
```





# Justify-content

## Space-between

Space-between mantiene el mismo espacio entre cada elemento flexible.

```
ul {  
  justificar contenido: space-between;  
}
```



# Justify-content

## Space-around

Finally, space-around keeps the same spacing around flex items.

```
ul {  
  justify-content: space-around;  
}
```



# Justify-content

## Space-evenly

Los elementos se distribuyen uniformemente dentro del contenedor de alineación a lo largo del eje principal. El espacio entre cada par de elementos adyacentes, el borde inicial principal y el primer elemento, y el borde final principal y el último elemento, son todos exactamente iguales.

```
ul {  
  justify-content: space-evenly;  
}
```

