

Diplomatura en



FullStack Developer

CLASE – 11

Layouts con CSS: Flexbox y Grid



Introducción a los Layouts con CSS

Layouts con CSS son la base fundamental para diseñar la estructura visual de una página web. En esta presentación, exploraremos los conceptos clave de los sistemas de layout más populares: **Flexbox** y **Grid**. Aprenderemos sus objetivos, características y cómo aplicarlos en ejemplos prácticos.

Conceptos clave de Flexbox

1

Contenedor Flexible

El contenedor **"display: flex"** define un contexto flexible para organizar sus elementos hijos.

2

Dirección y Alineación

Flexbox permite controlar la dirección del layout (fila o columna) y la alineación de los elementos.

3

Flexibilidad

Los elementos hijos pueden crecer o encoger según las necesidades del layout.

Objetivos y características de Flexbox

Objetivos de Flexbox

Flexbox busca ofrecer un sistema de layout eficiente y adaptable para crear diseños complejos.

Características clave

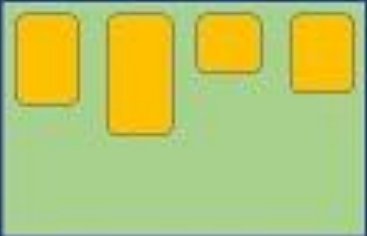
Control de dirección, alineación, orden y tamaño de los elementos flexibles.

Aplicaciones comunes

Barras de navegación, sistemas de grillas, diseños responsivos y layouts centrados.

align-items

flex-start



flex-end



center



stretch

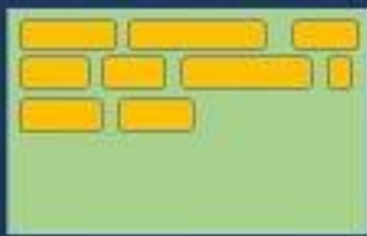


baseline

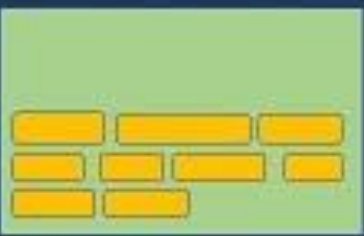


align-content

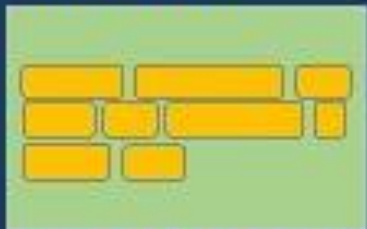
flex-start



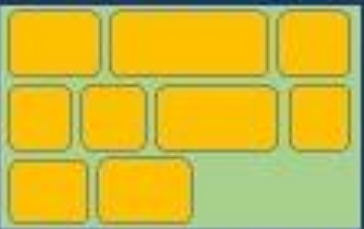
flex-end



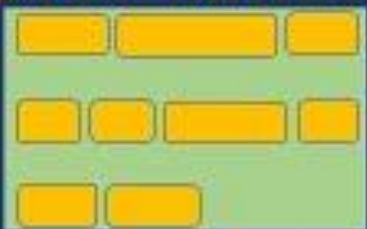
center



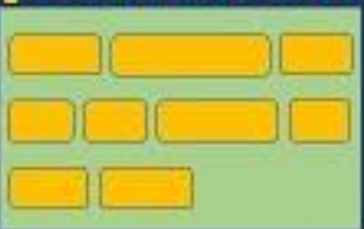
stretch



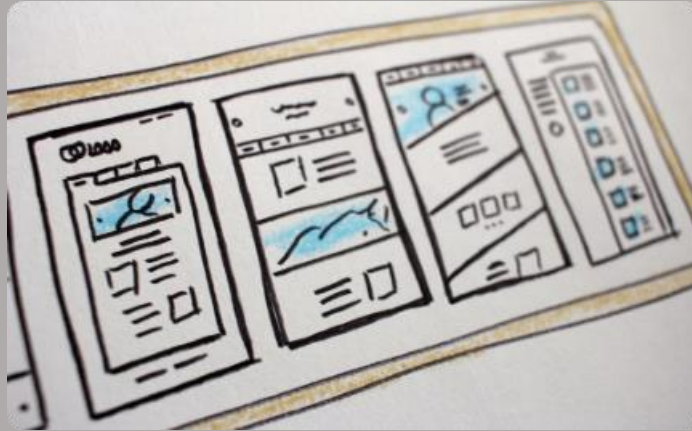
space-between



space-around

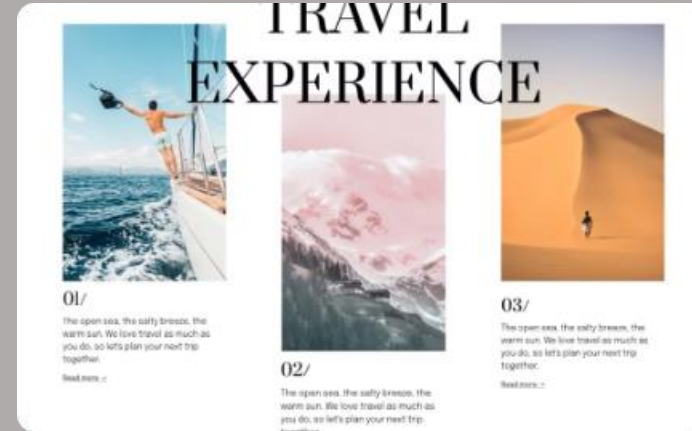


Ejemplos prácticos de Flexbox



Barra de Navegación

Distribuir enlaces de navegación de manera flexible y alinearlos horizontal o verticalmente.



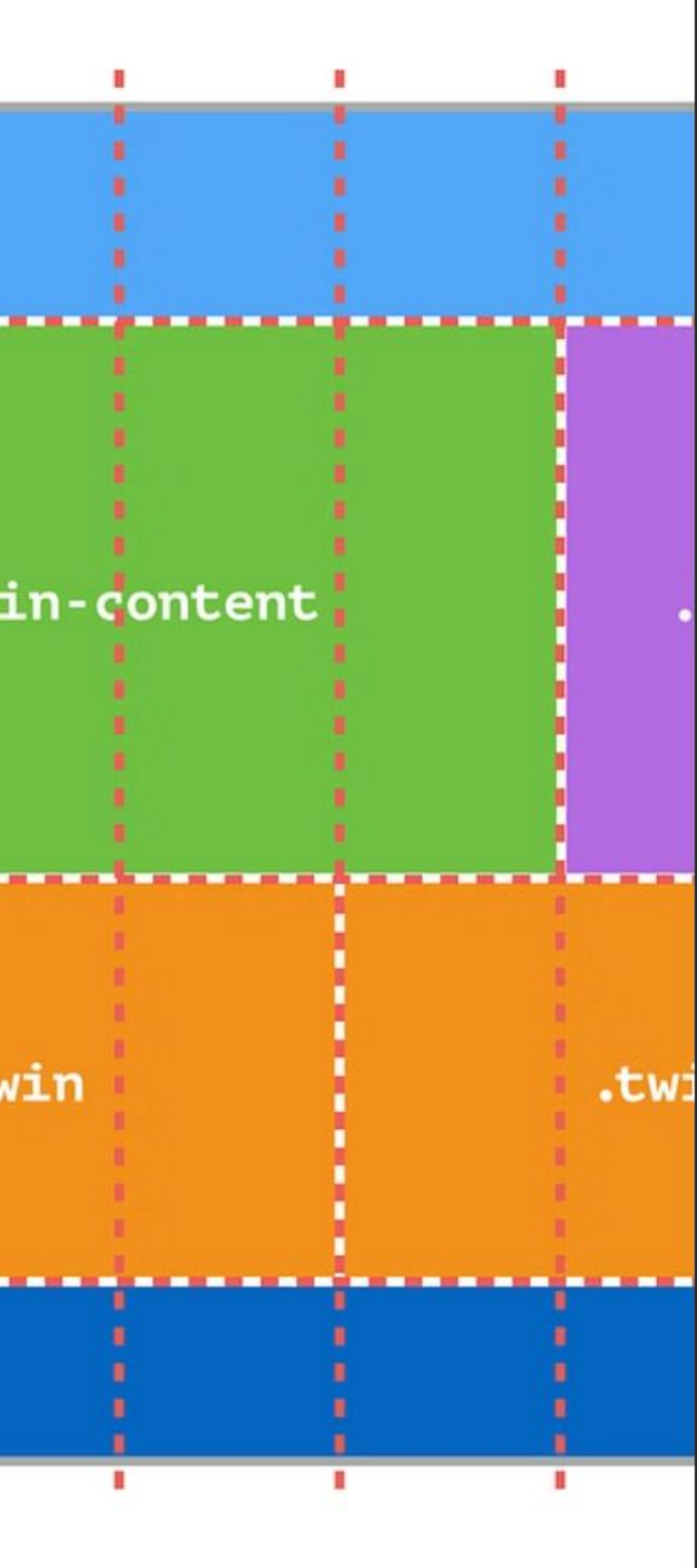
Diseño de Tarjetas

Crear un diseño de tarjetas adaptable y con elementos de diferente tamaño.



Layouts Centrados

Centrar elementos de manera vertical y horizontal utilizando las propiedades de Flexbox.



Conceptos clave de Grid

1

Contenedor Grid

El contenedor "display: grid" define una estructura de filas y columnas.

2

Unidades de Grid

Grid utiliza unidades de medida como fr, % o px para definir el tamaño de las filas y columnas.

3

Posicionamiento

Los elementos hijos pueden ser posicionados y alineados dentro de la estructura de cuadrícula.

Entendiendo el Sistema de Grid en CSS

El **contenedor Grid**, definido con "display: grid", establece una estructura de filas y columnas para organizar los elementos hijos de manera sistemática y controlada.

Grid proporciona un conjunto de **unidades de medida flexibles**, como fr (fracción), %, o px, que permiten definir el **tamaño y posicionamiento** de las filas y columnas con gran precisión. Esto brinda un control sin precedentes sobre la distribución de los elementos en la página.

Además, Grid facilita el **posicionamiento y alineación** de los elementos hijos dentro de la estructura de cuadrícula, simplificando la creación de diseños complejos y altamente personalizados.

En resumen, Grid es una herramienta poderosa que revoluciona la forma en que organizamos y distribuimos los contenidos en nuestras páginas web, ofreciendo una mayor flexibilidad y control sobre el diseño final.

LOS TRES PILARES DE CSS

Conocer estos conceptos
te hará dominar este
lenguaje



Prof. Alvaro Felipe 
@alvarofelipe

HERENCIA

Los hijos heredan estilos de sus padres, así no necesitas escribir estilos para todo.

```
body {  
  color: blue  
}  
  
h1 {  
  color: blue  
  /*No es necesario*/  
}
```

No es necesario
definir un color de
texto para los
títulos pues son
heredados de body



ESPECIFICIDAD

Cuando hay conflictos de estilos el navegador aplica solo el de mayor especificidad.

```
.parrafo {  
  color: blue  
}  
  
.article .parrafo {  
  color: red  
}
```

El párrafo es rojo
por que el
segundo selector
es más específico



CASCADA

Todo estilo sobrescribe a uno anterior. La C de CSS significa Cascada

```
.parrafo {  
  color: blue  
}  
  
.parrafo {  
  color: red  
}
```

El párrafo es rojo
por que
sobrescribe a los
anteriores estilos



Understanding CSS Grid Positioning

CSS Grid is a powerful layout system that allows you to precisely control the positioning and sizing of elements on a web page. At the core of Grid are a few key concepts:

Grid Container: The parent element that establishes the grid structure, defined with `display: grid;`

Grid Items: The child elements that will be placed inside the grid container.

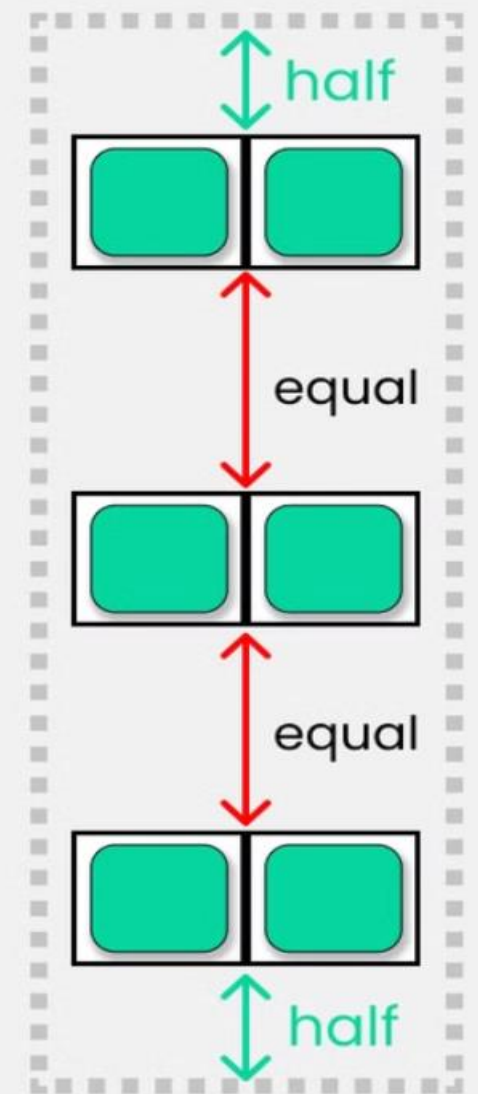
Grid Lines: Invisible lines that define the rows and columns of the grid. These can be referenced to position items.

Grid Tracks: The space between adjacent grid lines, representing the rows and columns of the grid.

With these building blocks, you can use various grid properties to control the layout, such as `grid-template-columns`, `grid-template-rows`, `grid-column`, and `grid-row`. Let's look at a step-by-step example...

content ->

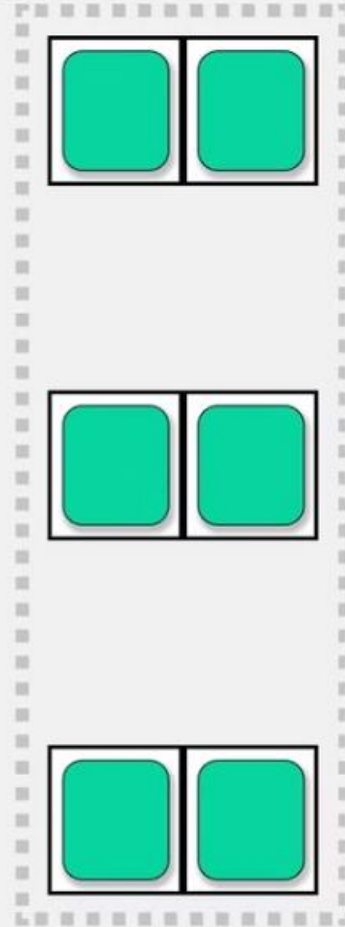
grid container



space-around

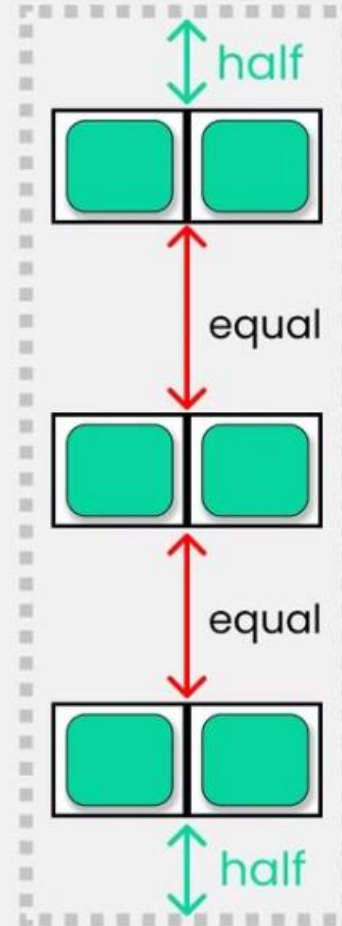
align-content -> Y Axis

grid container



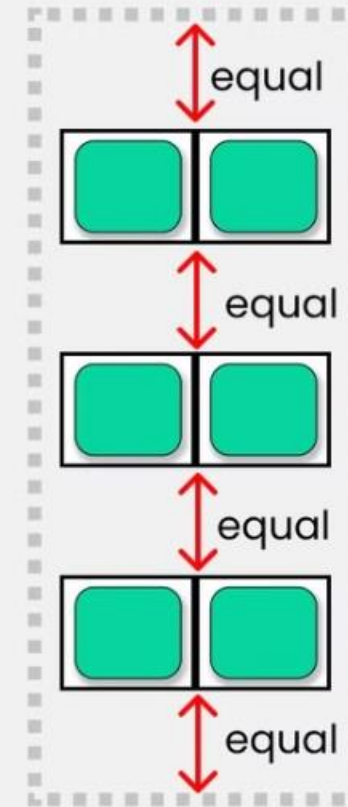
space-between

grid container



space-around

grid container

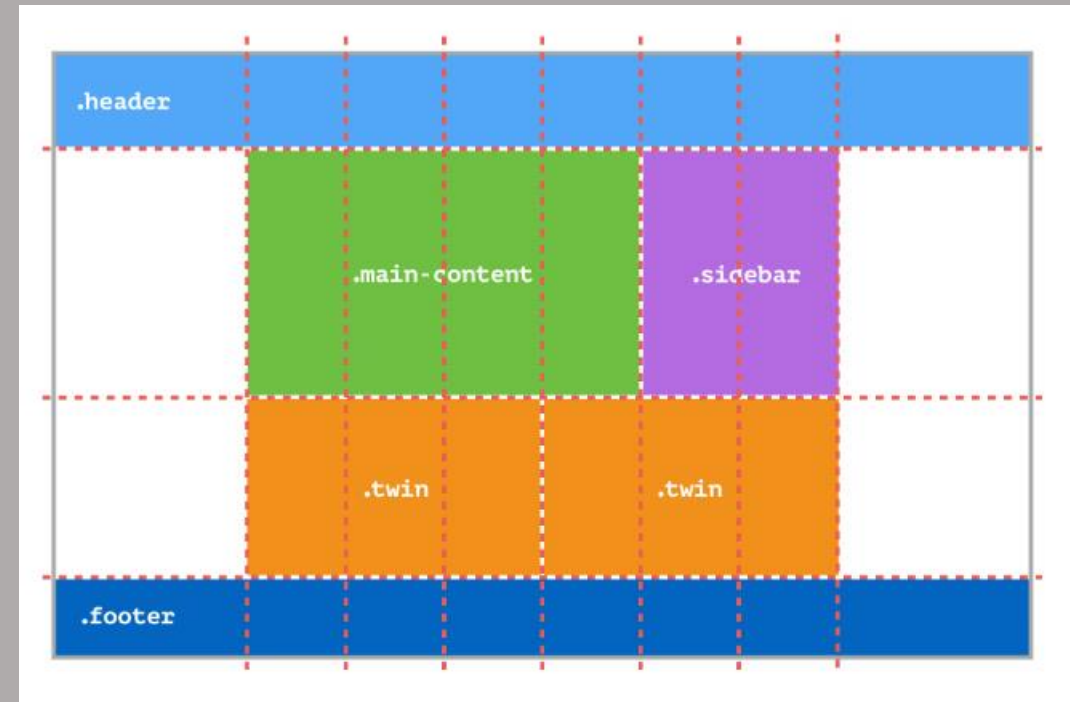


space-evenly

Conceptos clave de Grid

El contenedor Grid ("display: grid") define una estructura de filas y columnas para organizar los elementos hijos.

Grid utiliza unidades de medida como fr, % o px para definir el tamaño de las filas y columnas, brindando mayor control sobre el layout.



Los elementos hijos pueden ser posicionados y alineados de manera precisa dentro de la estructura de cuadrícula.

Objetivos y características de Grid

Objetivos de Grid

Ofrecer un sistema de layout bidimensional más avanzado y poderoso que Flexbox.

Características clave

Definir filas y columnas, colocar elementos en celdas específicas, combinar celdas.

Aplicaciones comunes

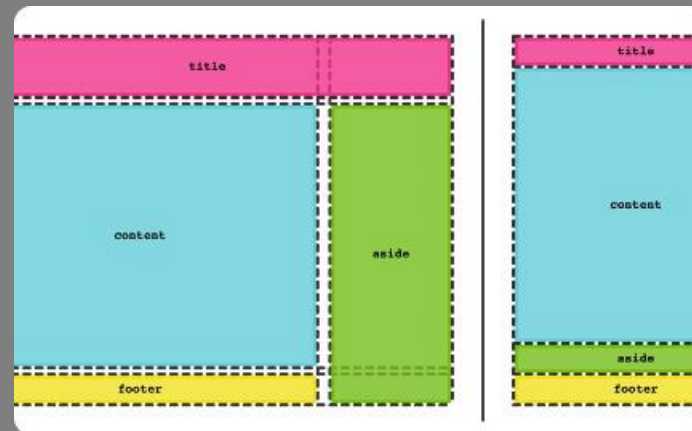
Diseños de grillas, layouts de periódico, diseños web complejos y aplicaciones responsivas.

Ejemplos prácticos de Grid



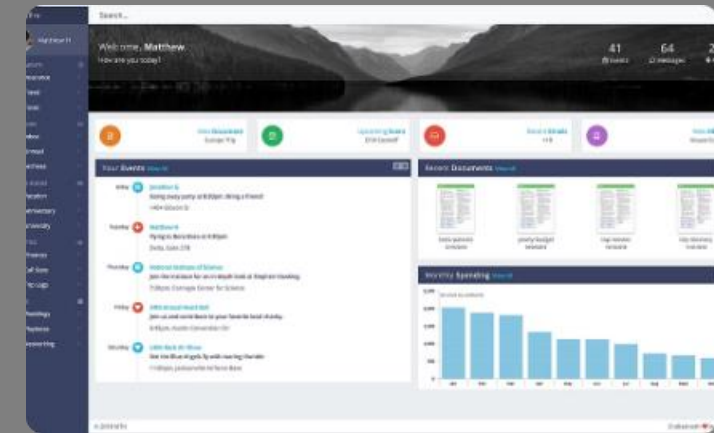
Diseño de Revista

Utilizar Grid para crear un diseño de revista con secciones y elementos de diferentes tamaños.



Diseño Responsivo

Adaptar el layout de Grid a diferentes tamaños de pantalla para una experiencia fluida.



Panel de Control

Organizar elementos de un dashboard o panel de administración utilizando la cuadrícula de Grid.

Conclusión y próximos pasos

Flexbox vs. Grid

Conocer las diferencias, fortalezas y casos de uso de cada sistema de layout.

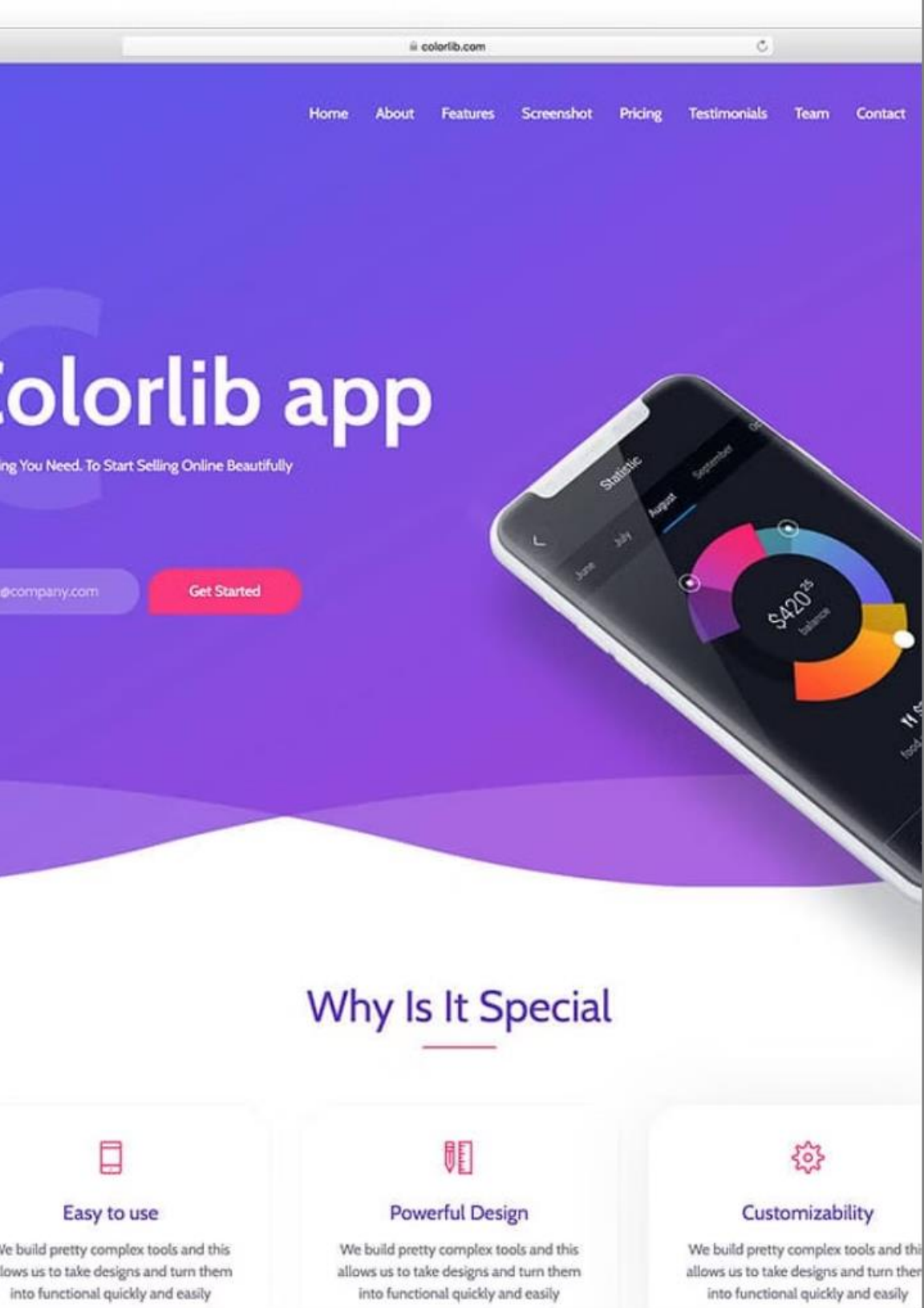
Próximos pasos

Practicar, experimentar y profundizar en el dominio de Flexbox y Grid.

Recursos recomendados

Libros, tutoriales y cursos que ayuden a mejorar tus habilidades de maquetación.
Realizar ejercitaciones para conocer y practicar.
Descubrir fortalezas y debilidades del aprendizaje





¡Gracias por tu atención!

Ahora tienes una sólida comprensión de los sistemas de layout Flexbox y Grid. Recuerda seguir practicando y profundizando en estos conceptos para dominar el diseño de layouts web.

abril 3, 2024

Las 3 grandes novedades que ha dejado el 2023 en CSS

<https://www.digitaljump.com.ar/novedades-css-que-dejo-el-2023/>

Anidamiento CSS

Nuevos valores para la propiedad "text-wrap"

Nueva pseudoclase: ":has()"