



Una guía al diseño responsive

Box modeling

Todos los elementos en una página tienen un tamaño y un espacio entre ellos, cada contenedor actúa como caja y el navegador los interpreta para acomodarlos uno al lado del otro.

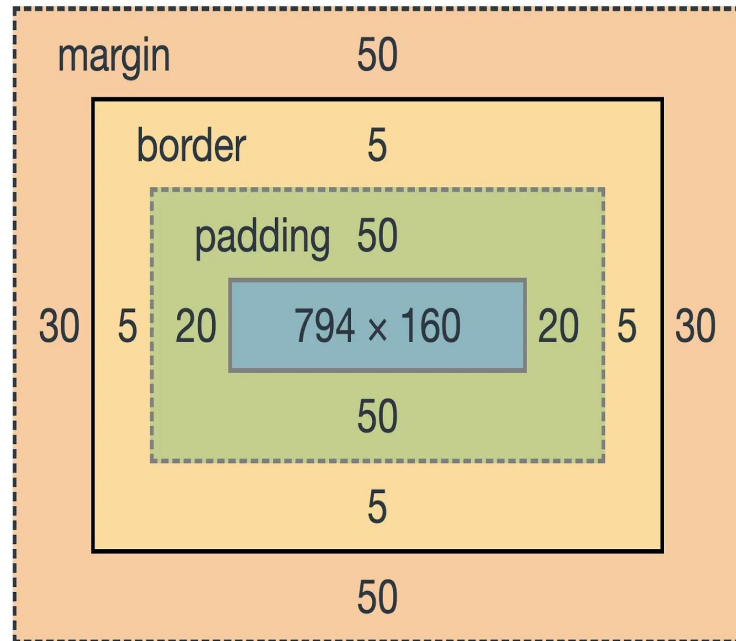
1. **content:** Todo lo que tiene adentro la caja
2. **padding:** Es el espacio que hay entre el contenido y el borde de la caja, este aumenta el tamaño visible de la caja
3. **border:** El borde que encapsula el contenido y el padding
4. **margin:** este es el espacio que hay entre la caja y otra caja o elemento, y no afecta el tamaño visible de la caja

Shorthand: (padding/margin)

margin: Top Right Bottom Left;

margin: Top Right/Left Bottom;

margin: Top/Bottom Right/Left;



Qué es “responsive”?

El **diseño web responsive** consiste en permitir que un sitio web **se adapte automáticamente** a diferentes tamaños de pantalla, como celulares, tablets o computadoras.

No se trata solo de que la página se encoja, sino que **los elementos cambien de tamaño, posición o distribución** para ofrecer una buena experiencia de usuario en cualquier resolución de pantalla.



¿Por qué es importante?

- El 60% del tráfico web viene de dispositivos móviles. Esto significa que, si una página no se ve bien en un celular, la mayoría de los usuarios tendrán una mala experiencia.
- Mejora la experiencia del usuario.
- Es clave para el SEO (Search Engine Optimization, Google prioriza sitios responsive).

Herramientas para el Diseño Responsive

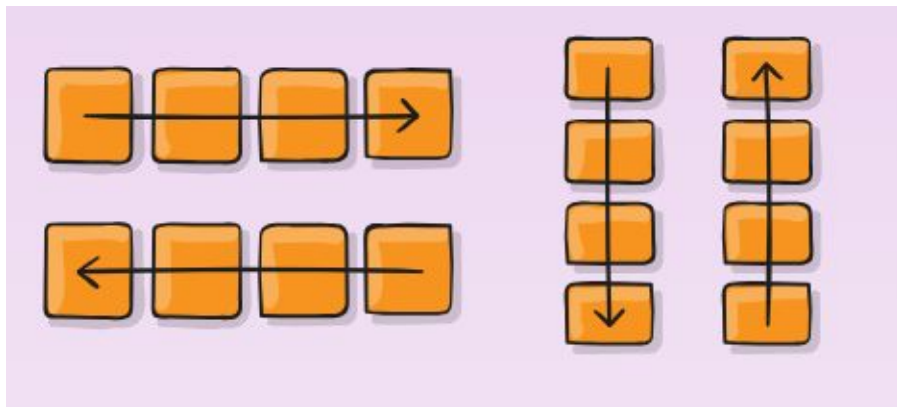
1. **Flexbox:** Sistema de diseño unidimensional, Ideal para alinear, distribuir y centrar elementos dentro de un contenedor.
2. **CSS Grid:** Sistema de diseño bidimensional (filas y columnas). Se usa para construir estructuras más complejas como el layout principal de una página.
3. **Media Queries:** Sistema transversal. Permite aplicar estilos CSS solo cuando la pantalla cumple cierta condición.

Flex: Información relevante

Flex es un posible valor de la propiedad CSS display (`display: flex;`). Este nos da en un contenedor una propiedad responsive. este a su vez va a tener efecto sobre sus hijos o elementos encapsulados inmediatamente después



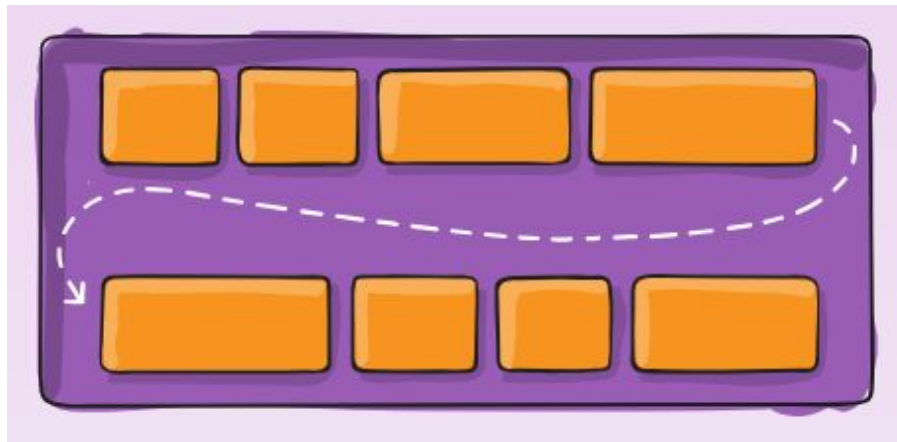
Propiedades importantes:



flex-direction: Esta propiedad nos permite ajustar el contenido (items) a uno de los ejes de la pantalla, al eje X (row: si queremos que se lea de izquierda a derecha, row-reverse: si queremos que se lea de derecha a izquierda), o si queremos que se ajuste al eje Y (column, column-reverse).

Propiedades importantes:

flex-wrap: por defecto, la propiedad flex tratará de acomodar todos sus hijos (items) dentro de una sola línea, esta propiedad nos da control de si queremos que sean elementos que tan pronto toquen el borde de su contenedor bajen a una segunda línea.



Propiedades importantes

`justify-content`: Esta propiedad define como queremos que ajusten los elementos en el eje que determinamos con la propiedad `flex-direction`, hay 6 posibles valores:

1. `flex-start`: se acomoda todo desde el inicio de la propiedad `flex-direction`.
2. `flex-end`: se acomoda todo desde el final de la propiedad `flex-direction`.
3. `center`: se acomoda todo desde el centro del eje determinado por el `flex-direction`.
4. `space-between`: se acomoda el primer objeto al inicio, el último al final y con el espacio en medio, se calculan las distancias que se usarán como margen entre hijos (items).
5. `space-around`: se le da un margen a cada hijo (item), cuando todos tienen un margen, se acomodan el primer elemento al inicio y el último al final.
6. `space-evenly`: se asegura que el espacio entre márgenes sea igual para cada uno de los elementos y se acomoda el primero al inicio y el último al final.

justify-content:

flex-start



flex-end



center



space-between



space-around



space-evenly



Otras propiedades importantes

align-content y align-items: ambos son para trabajar en el eje que con flex-direction no fue seleccionado como el eje principal.

Tiempo de Practicar:

Práctica:

[Un juego para aprender CSS flexbox](#)

Mas teoria:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

¿Flexbox o Grid? ¿Cuál usar?

Característica	Flexbox	Grid
Dirección principal:	Una sola (fila o columna)	Dos (filas y columnas)
Ideal para:	Elementos en línea o componentes	Estructuras completas de páginas
Control	Limitado	completo
Alineacion	Mejor	Buena

Grid

<https://cssgridgarden.com/#es>

