

EJEMPLO DE DISEÑO RESPONSIVE CON FLEXBOX

La caja de herramientas que llevamos vistas para maquetación no está nada mal. Pero si te fijas, el posicionamiento de elementos con CSS se convierte en la mayoría de ocasiones en un sopesar los efectos colaterales que cada técnica empleada genera hasta ver la que mejor nos encaja. De ahí que la demanda del mercado para disponer de más y mejores tecnologías no haya cesado. Al igual que se investiga con técnicas quirúrgicas buscando ser más efectivas y con menos efectos secundarios para el paciente.

Algo parecido les va a pasar a los maquetadores cuando empiezan a estar hasta el gorro de la propiedad ya conocida de CSS como es **float**, al tener que enfrentarse con ciertas distribuciones de elementos mucho más complicadas: desde el control preciso de las dimensiones verticales de la rejilla y el modo de distribuirse automáticamente al cambiar las resoluciones, hasta poder cambiar de distribución por completo una página sólo con un pequeño cambio de CSS, sin tocar el contenido.

Para dar cobertura a estos casos y muchos otros de maquetación compleja, la W3C diseñó unas tecnologías como **Flexbox** y **Grid**. Ambas tecnologías no siendo excluyentes entre sí, más bien se pueden complementar porque, de hecho, cuando así ocurre se les puede sacar más potencia al combinarse para lograr los mejores resultados. En el siguiente ejemplo práctico puedes comprobarlo [<https://www.silocreativo.com/flexbox-vs-css-grid-ejemplo-practico/>].

Existen por ahí cantos de sirena al estilo de esas publicidades que te venden aprender algo sin esfuerzo como puede ser Bulma [<https://bulma.io/>] donde proclaman eso de "hágaselo *responsive* sin tener ni idea de CSS". Lo dejo ahí para atrevidos.

Primero con Flexbox

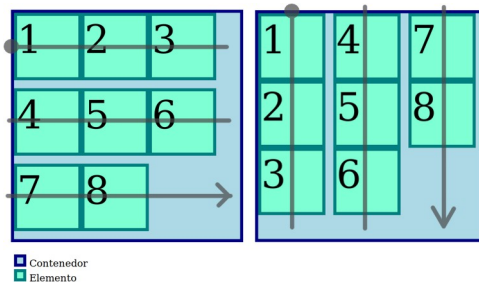
Flexbox es una tecnología que proviene de CSS3 que nos permite acceder a un modelo de cajas flexibles que nos facilita la ordenación de cualquier tipo de contenedores, gracias a que incorpora más opciones de control para la maquetación. Ello posibilita no tener que hacer tantos cambios en los estilos consiguiendo un diseño más ajustable a todo tipo de dispositivos con sólo realizar algún que otro cambio para que el sitio se presente lo más perfecto posible en cualquier resolución de pantalla.

De ahí que el 50% del aprendizaje de esta tecnología pasa por conocer bien sus fundamentos. La propiedad fundamental que convierte un contenedor en gestor unidimensional de elementos hijos con Flexbox es [**display: flex;**]. A partir de ahí queda indicarle a dicho gestor tanto en qué dirección van ir [**flex-direction: row/column;**] como si les permite saltar a la siguiente fila/columna o rebosar el contenedor [**flex-wrap: wrap/nowrap;**].

En 15 segundos te resume gráficamente las propiedades Flexbox
<https://youtu.be/tg9wa7jYnsw>

Este ejemplo sería con
flex-direction: row; y
flex-wrap: wrap;

Propiedad resumen:
flex-flow: row wrap;



Este ejemplo sería con
flex-direction: column; y
flex-wrap: wrap;

Propiedad resumen:
flex-flow: column wrap;

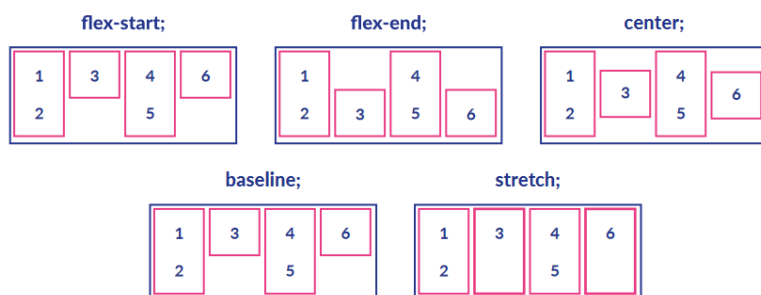
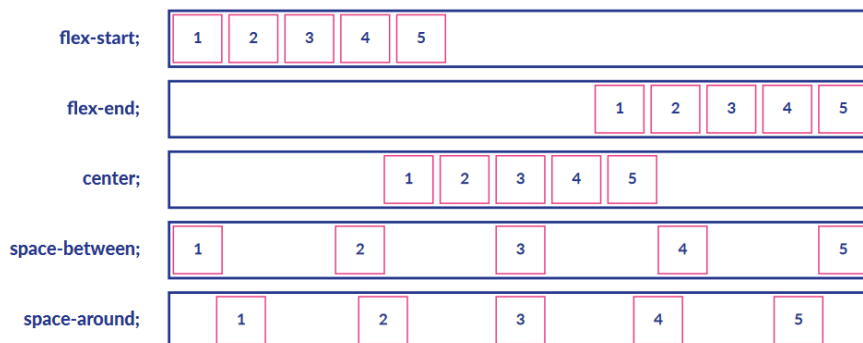
El código **1.FB_sus_ejes.html** lo presenta de una forma muy didáctica.

Nos queda afinar algo más puesto que aún no somos capaces de establecer cómo queremos que los elementos hijo del contenedor se organicen de forma automática dentro del contenedor. Para ello se suben al carro otras tres propiedades como son [**justify-content: flex-start/flex-end/center/space-between/space-around**], [**align-items: flex-start/flex-end/center/stretch/space-between/space-around**] y [**align-content: flex-start/flex-end/center/baseline/stretch**].

justify-content

Define la alineación a lo largo del eje principal establecido (Hor o Ver), distribuyendo el espacio libre que vaya quedando.

Magnífica demo en
[<https://codepen.io/team/css-tricks/pen/zzJMGJ>]



align-items

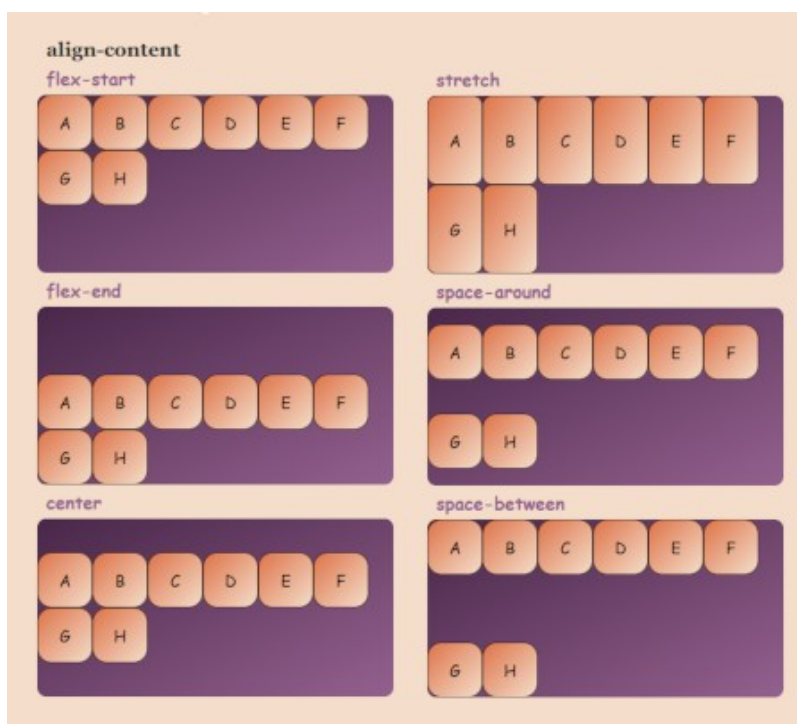
Define el comportamiento por defecto de los elementos sobre el eje transversal (perpendicular al principal), distribuyendo el espacio libre que vaya quedando.

Magnífica demo en
[https://www.youtube.com/watch?v=_ERtW0z3FF0]

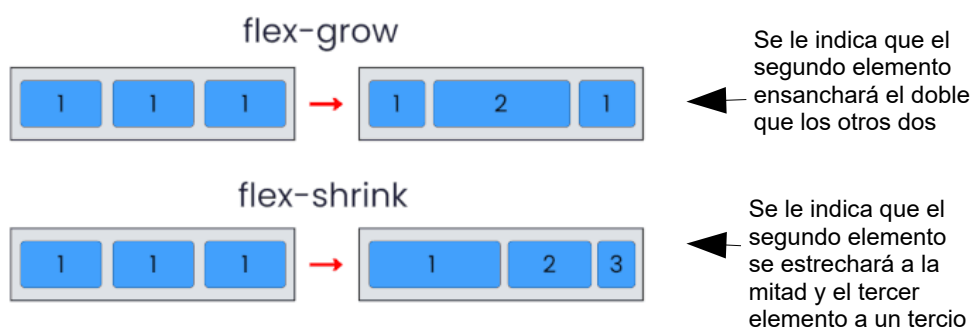
align-content

Define el comportamiento de las líneas ajustándolas dentro del contenedor cuando queda espacio extra en el eje transversal.

Magnífica demo en
[<https://www.youtube.com/watch?v=GShRBWz8jIM>]



Por último, está una capacidad que pueden adquirir los elementos hijo de un contenedor de ensanchar o adelgazar de tamaño en función del espacio restante (cuando exista) y los parámetros suministrados. Se trata de las propiedades [**flex-grow: n°**] y [**flex-shrink: n°**]. El número que acompaña a estas propiedades viene a indicar cuántas veces puede ensanchar (**flex-grow**) o adelgazar (**flex-shrink**) respecto al resto de elementos que le acompañan.



Para ir entendiendo cómo se aplica una hoja de estilos basada en Flexbox, ahí va una muestra simplona de contenedor con dos columnas. En este caso, tras convertir el contenedor en flexible (**display: flex;**) y no indicar nada sobre la dirección del eje principal lo que por defecto lo va a disponer en fila los elementos hijo (**col1, col2**).

Seguidamente prepara dichos elementos de forma que siempre que el contenedor sea lo suficientemente ancho, el segundo de ellos (2ª columna) ocupará el doble de espacio que el primero.

| Código HTML5 | Hoja de estilos |
|---|---|
| <pre><body> <div class="contenedor"> <div class="col1"><p>Texto que va en la columna 1</p></div> <div class="col2"><p>Texto que va en la columna 2</p></div> </div> </body></pre> | <pre>.contenedor { display: flex; } /* Por defecto la dirección es row */ .col1 { flex: 1; margin-right: 5%; } .col2 { flex: 2; }</pre> |

Ejemplos interactivos reunidos (en lengua portuguesa)
<https://origamid.com/projetos/flexbox-guia-completo/>

La secuencia de códigos de ejemplo que acompaña a esta batería tiene por objeto que pruebes cada una e identifiques en su código las propiedades de Flexbox que se han ido exponiendo asta aquí para asimilar su funcionamiento. Una buena manera de hacerlo es utilizando las **Herramientas de Desarrollador** del navegador (Firefox y Chrome) para ver cómo afecta cada propiedad al diseño confeccionado.


Por último, dejar constancia de una dirección web [<https://philipwalton.github.io/solved-by-flexbox/>] en la que grandes quebraderos de cabeza para maquetadores CSS vieron su solución de la mano de Flexbox.

Tablero de Formación Certificada



Acabó en Diocesis-Arriaga (Vitoria) sus estudios de **Formación Profesional en Informática** en 1985 y posteriormente en la Universidad de Sevilla los de **Ingeniero Técnico en Informática de Gestión**

1985 / 1990



Especialista en Internet y sus Aplicaciones en el Mundo de la Educación y de la Empresa por la Universidad Nacional de Educación a Distancia (UNED)

1996



Coordinador del Proyecto Europeo Comenius "Intercambio de Experiencias Educativas en Informática" SAFA Ntra. Sra. de los Reyes (Sevilla), La Providence (Amiens-FR) y St. Ignatius College (Enfield-ING)

1996-1999



Curso sobre el **Lenguaje de Marcas XML** a través del portal de Internet ALMAGESTO perteneciente al Grupo Eidos

2001



Experto Profesional en **E-Learning**: Educación y Formación por Internet por la Universidad Nacional de Educación a Distancia (UNED)

2007



Master en Desarrollo de Aplicaciones Empresariales en **J2EE** por Sun Microsystems y Formación Digital

2008



Curso **Metodologías y Herramientas Orientadas al Aprendizaje Activo** del Alumnado de FP por impartido por Daniel Irazola (Maristas-Durango) enmarcado en la XXVI Escuela de Verano SAFA

2008



Programador Certificado en Java SCJP5 por la empresa Sun Microsystems con una puntuación de 86%

2009



Gestión del **E-Learning en Moodle**: Instalación, Administración y Uso Avanzado impartido por Tadel Formación, impulsado y patrocinado por el FSE y el MICyT

2009




Curso **MY OXFORD ENGLISH** cuyos 10 niveles son compatibles con la certificación B2 diseñada por Oxford University Press

2013




Experto en **Análisis Forense Informático y Pericial** por la Asociación de Peritos Judiciales Tecnológicos de Andalucía

2019



Programa **Avanzado en Python** por el Instituto Tecnológico Telefónica para el desarrollo de APIs JSON en DJANGO y FLASK

2020




Cursos sobre **Java, Básico de Android, Kotlin, Javascript, Typescript y C# (.NET)** por la plataforma CODECADEMY

2019-2021



Cursos sobre **Laravel5, Colecciones Java, Java y XML, Clean Code y Javascript** por la plataforma OPENWEBINARS

2019-2021



Especialista en **Maquetación Responsive** con **HTML5, Flexbox, CSS Grid y Bootstrap** por la plataforma campusMVP

2021