Curso de Flexbox CSS - Posicionamiento de los elementos.

Clase 1

Justify-content: center; Hace que todo el contenido se centre en el medio dentro de la misma función de esta propiedad está left y right.

Justify-content: flex-start Hace que todos los elementos se vayan al inicio y flex-end hace que se vayan al final, es mejor usar justify-content: flex-end en vez de right y flex\_start en lugar de left en nuevos desarrollos web para garantizar la compatibilidad con todos los navegadores modernos.

Justify-content: flex-end; Pone todo el espacio al lado izquierdo de la pantalla, lanzando el contenido hacia la derecha.

Justify-content: space-between; Hace que los elementos tengan espacios entre ellos, pero el primer elemento ocupa el borde del contenedor y el ultimo ocupa el final del borde del contenedor, por lo que si tenemos dos elementos entonces uno ira a la derecha del todo y otro a la izquierda.

Justify-content: space-around; Hace que los elementos tengan un espacio entre ellos, pero el primer elemento comenzara a la mitad del espacio que tendrán el resto de los elementos entre ellos y el último elemento terminará con un espacio a su derecha que será la mitad del resto de todos los espacios.

Justify-content: space-evenly; Esta propiedad hace que todos los elementos tengan el mismo espacio entre ellos, desde la parte izquierda del primer elemento como el resto de elementos y el ultimo tendrán el mismo espacio entre ellos.

Align-items: center; La propiedad align-items hace que los elementos dentro del elemento padre que está con flex se alineen verticalmente, por lo tanto, para que ellos se alineen en el centro podemos utilizar align-items: center, poniendo esta propiedad en el elemento padre de los elementos que queremos alinear, en este caso header.

Clase 2

Flex-direction: La propiedad CSS **flex-direction** especifica cómo colocar los objetos flexibles en el contenedor flexible definiendo el eje principal y la dirección (normal o al revés). Algunas de las propiedades o valores que se pueden agregar son:

* row (usado por defecto)
* row-reverse
* column
* column-reverse

Flex-wrap: La propiedad **flex-wrap** de CSS especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas. Si la cobertura (wrap) está permitida, esta propiedad también te permite controlar la dirección en la cual serán apilados los elementos. Los valores aceptados serán los siguientes:

* nowrap
* wrap
* wrap-reverse

Clase 3

¿Por qué puede llegar a ser complicado utilizar justify-content: space/between-around para el grid?

1. Es complicado usarlas porque ellas ponen comportamientos que no son buenos para grids, por ejemplo, si usamos space-around y en la última fila del grid tenemos sólo dos elementos en lugar de cuatro, el espaciado entre ellos será equilibrado. Es complicado usarlas porque ponen comportamientos que no son adecuados para grids. Tanto space-between cuanto space-around harían que la última línea quedara muy fuera del patrón, en caso de que esa tuviera menos elementos que el resto de las líneas.
2. Es complicado usarlos porque ponen comportamientos que no son adecuados para grids, por ejemplo, si usamos space-between y en la última línea del grid tenemos sólo dos ítems en lugar de cuatro, el espaciado entre ellos no quedará equilibrado. Es complicado usarlas porque ponen comportamientos que no son adecuados para grids. Tanto space-between cuanto space-around harían que la última línea quedara súper rara, en caso de que esa tuviera menos elementos de que el total de elementos por línea.

Clase 4

Flex-order: Permite ordenar los flex ítems con otra posición. La propiedad CSS **order** especifica el orden utilizado para disponer los elementos en su contenedor flexible. Los elementos estarán dispuestos en orden ascendente según el valor de order. Los elementos con el mismo valor de order se dispondrán en el orden en el cual aparecen en el código fuente.

**Recordar el uso de mediaqueries para la visualización de los elementos en móviles (responsive).**

**(CSS-Tricks.com (Documentación relevante sobre flexbox)).**

Flex-grow: La propiedad **flex-grow** de CSS especifica el factor de crecimiento de un elemento flexible (que tiene asignado display:flex), en su dirección principal. El factor de crecimiento especifica qué cantidad del espacio restante dentro del contenedor flexible, debería ocupar el item en cuestión.

La dirección principal puede ser la altura o el ancho del elemento, dependiendo del valor de flex-direction.

Ejemplo: .box {flex-grow:1;}

Flex-shrink: La propiedad CSS **flex-shrink** especifica el factor de contracción de un flex item. Los flex items se encogerán para llenar el contenedor de acuerdo a su número flex-shrink , cuando el tamaño por defecto de los flex items sea mayor al de su contenedor flex container.

Ejemplo: .box {flex-shrink:1;}