

Y aquí es donde comienza lo bueno. Bueno, todo es necesario para hacer un buen desarrollo, pero hay algunas herramientas que son fundamentales. Hablaremos de posicionamiento.

Antiguamente con "float", como vimos en la clase pasada, era cómo se hacía y es muy difícil lograr un control absoluto, ¿no?, tal vez en una resolución única, pero hoy día accedemos a una misma página web, en una pantalla, con un monitor extra grande, con un monitor más chico, con una laptop, con una tablet o hasta con un teléfono, ¿cómo hacer que el contenido se adapte a todas esas resoluciones, sin tener que hacer una página web para cada monitor? Lo cual es un incordio, además de que es sumamente impráctico y costosísimo, bien, para esto hablaremos de el posicionamiento. Hablaremos de las herramientas modernas de diseño adaptable, hablaremos de "flex box", hablaremos de "grid". Comenzamos con el modelo de caja flexible.

Bien, vamos a aprender cuál es el principio básico de "flex box", del sistema de caja flexible, que es la evolución en la cuestión del posicionamiento en el CSS moderno, junto con "grid", que veremos un poquitito más adelante.

¿Qué estamos viendo en pantalla? Bien, a la derecha pues, una simple pieza de código "html" que no tiene otra cosa que un "div", con la clase "contenedor". Vamos a quitarlo por el momento, porque no está cumpliendo ninguna función, simplemente tenemos aquí uno, dos, tres, cuatro "divs", cuatro secciones con un número dentro. Simplemente para identificarlas en nuestro archivo de CSS. Vamos a borrar por el momento esta línea y vamos a concentrarnos en nuestros "divs". Simplemente, tiene un ancho de 200 píxeles, un alto y hemos alineado el texto al centro. Claro, para poder ver un poco mejor esto, vamos a agregarles un color de fondo, por caso, lo vamos a copiar y lo vamos a poner aquí, ahora sí. Cada uno de los "div", como pueden ver ustedes aquí, tiene asociada a una clase. En este caso, "item one", o "ítem uno, dos, tres y cuatro", lo hice simplemente para darle un color distinto a cada uno, esto es para que ustedes puedan ver en pantalla cómo van fluyendo los elementos, mayormente. Claro, también podríamos indicar que cada uno de estos "div"s tengan también un "font size" un poco más grande, un poco más, inclusive, para poder apreciar mejor los números.

Bueno, ahora, ¿cómo podemos hacer que estas cajas se ordenen según nuestras preferencias? Aquí es donde entra el concepto fundamental de "flex box". Cada uno de los ítems que queremos modificar, digamos, que queremos alinear en la pantalla de algún modo particular, debe estar contenido en un contenedor flexible, puede ser cualquier elemento, puede ser un "div", puede ser una etiqueta semántica como "section", como "article", como "navbar", no importa, cualquier elemento que pueda contener a otros. En este caso, usaré simplemente un "div" y voy a ponerle a ese "div" una clase, el nombre es "convencional", puede llamarse de cualquier modo, manera, suele ponerse, para este tipo de enseñanzas, "flex container" o contenedor flexible o simplemente "container", para abreviar, okey. Este contenedor tiene que envolver, es un "wrapper", bueno, un contenedor tiene que contener o envolver a los demás elementos. Perfecto, ahora en pantalla ustedes no van ningún cambio, es lógico, que no vean ninguno, pero si vamos a nuestro CSS, ahora vamos a poder indicar que nuestra clase container, es decir, el "div" que contiene esta clase, será alcanzado por esta propiedad.

La propiedad fundamental que pone en marcha el sistema de contenedores flexibles o de "flex box", es la propiedad "display flex". Al indicar esta propiedad sobre el elemento padre, sobre la caja contenedora, noten cómo cambia en pantalla. Lo que ocurre con los "divs", es que los "divs" están ahora alineados dentro del contenedor. Bueno, es cierto que también ha cambiado su tamaño y algunas cuestiones, ya les voy a explicar por qué. Para explicarlo, mejor hacerlo en el código, vamos a indicar un "background color", para el contenedor de algún color, bien, puede ser este.

Bien y no lo están viendo de fondo, ¿por qué?, bueno, esto tiene que ver, no con "flex box", sino con una cuestión de CSS. Aquí hemos indicado que los "div", todos los "div", tienen un ancho de 200 píxeles y una altura de 150. Okey, esto alcanzaba a cada uno de nuestros "div", antes de que creásemos el contenedor flexible que los envuelve. Ahora, el contenedor flexible, también es un "div", por lo tanto, está siendo alcanzado por esta propiedad, es decir, tiene 200 píxeles de ancho.

El contenedor flexible intentará acomodar a todos sus hijos en el espacio que tenga disponible, por eso, ahora sus hijos ya no tienen los 200 píxeles de ancho. Tienen que conformarse cada uno, bueno, supongo que con 50 píxeles, ¿verdad? Porque son 200 píxeles para cualquier "div", en este caso, para el contenedor flexible, dividido entre cuatro hijos. Esto podemos solucionarlo con, bueno, pues, siendo más específicos, en la regla de selección. Vamos a indicar que estos 200 píxeles, por 150 de ancho, sólo se apliquen a los "div" que estén dentro de otro "div", okey. Entonces, ahora lo que está ocurriendo es lo siguiente: solamente estos cuatro "div" de aquí, tienen 200 anchos, 200 píxeles de ancho, mientras que el contenedor flexible, bueno, pues tiene todo el ancho de la pantalla, porque no se le ha especificado que tenga ninguno. Entonces, ¿ven?, si pongo toda la pantalla aquí disponible, noten que este es el color de fondo del contenedor, para eso le puse un color de fondo, para que sea un poquito más autoexplicativo. Ahora, para que esto mejore aún más en la explicación podríamos, por ejemplo, también al contenedor darle, darle un ancho, perdón, un alto, un ancho ya tiene. Vamos a darle, pues, aquí estamos siendo un poco arbitrarios, 800 píxeles de alto. Bueno, ahora vamos a ver cómo se acomodan los elementos dentro de un contenedor flexible. Como ustedes habrán notado lo primero que ocurrió es que se pusieron uno junto a otro, en fila. Recuerden lo que ocurría cuando no teníamos un contenedor flexible que lo se envolviera. Vamos a comentarlo aquí en el "html", vamos a quitar del flujo de nuestro documento el contenedor, okey, ya no tenemos un contenedor flexible. Bien, están uno sobre el otro, uno sobre el otro, okey. Ahora, ponemos nuevamente en función nuestro contenedor flexible, vamos a intentar un poco el código. Nuestro contenedor flexible tiene 800 píxeles de alto, por, bueno, pues como no le indicamos ancho tiene toda la pantalla para su dominio y cada uno de estos elementos se ha ordenado dentro del contenedor flexible, uno junto al otro. Ésta es la posición por defecto que ocupa cada uno en fila, veremos mucho más adelante. Ahora vamos con algunas propiedades básicas, ¿cómo podemos ordenar esa fila?, bien, si achicamos un poquito aquí, hacemos 100 píxeles, lo vamos a ver aún mejor. Perdón, no es este el valor que quiero cambiar, en realidad, el valor que quiero cambiar es el ancho, no el alto. Bueno, le puse mil, en vez de 100. Ahora sí, okey, o 150, sí, estoy un poco indeciso, disculpen. Okey, correcto, ahora, ¿cómo podemos hacer que estos elementos que están en la fila se muevan a lo largo del contenedor? Bien, podríamos indicar en el contenedor padre, en el contenedor padre, no en los hijos, okey, podríamos indicar la propiedad "justified content", que va a ordenar mis elementos en el eje principal, en el eje central, okey. Entonces, vamos a indicar, por ejemplo, que queremos que estén al centro, bien, y se mueven al centro. También podemos decir que estén al final, con

la propiedad "flex end". Bueno, la propiedad "flex start" es lo mismo que no ponerla ¿verdad?, porque es la propiedad por defecto para el "justified content", para la propiedad ya "justified content" es el valor por defecto "flex start". También podríamos indicar que haya espacios en medio, "space between" o que haya espacios alrededor de los contenedores, con "space around" o podríamos, inclusive, decir que quiero que haya la misma cantidad de espacio entre cada una de las cajas. Para eso tenemos el valor "space evenly".

Muy bien, ahí tenemos, entonces, algunas posibilidades de posicionamiento. Otra vez, si hubiésemos elegido al centro, por caso, y quisiéramos separar cada una de las cajas, bien, podríamos "hackear", digamos, este "flex box" poniendo aquí una propiedad "margin" en cada una de las cajas interiores, pero no sería necesario, porque de hecho "flex box" tiene una propiedad para hacer eso, que es row gap y colonga es decir digamos el espacio entre filas o el espacio entre columnas aquí lo que quiero indicar es el espacio entre columnas de hechos filas tengo una sola por lo tanto la propiedad "row gap" no haría nada, pero si la propiedad "column gap", seteada en, bueno, pues, un valor arbitrario, 10 píxeles por ejemplo. Y ahí tenemos un espacio. ¿Quieren un poco más? Vamos a poner un "rem".

Okey y ahí tenemos el espacio entre cada una de estas, de estos "div" ¿verdad?, de nuestros "div" con la propiedad "column gap". Okey. Ahora, ¿cómo podemos hacer para que esto se mueva en la otra dimensión? Es decir, en el, en el "cross axis" aquí, tenemos el "main axis", que en este caso es horizontal. Luego podemos cambiarlo, pero por defecto los elementos en un contenedor flexible, fluyen en una fila "row", por lo tanto, esta es la "main axis" o el eje principal y el "cross axis", en cruz o eje secundario, es en este caso, el vertical. ¿Cómo podemos moverlo allí? Bueno, para moverlo allí tenemos la propiedad, siempre estamos hablando, recuerden, de propiedades del contenedor padre, del contenedor flexible, "align items", y aquí podríamos indicar "center" también y van a ir al centro del contenedor. Me imagino que ustedes, ya están ahí rascándose vuestras cabezas, diciendo "¡ah!, pero esto es muy potente", sí, efectivamente, "flex box" es súper potente y nos da un control muy amplio de nuestro diseño. Podemos ir seleccionando cada artículo, cada sección, cada galería de imágenes, lo que sea, y darle un posicionamiento muy, muy fino, con mucho detalle, con unos pocos comandos, en realidad. Okey, por supuesto, "align items" maneja también las propiedades "flex start", que esta es la propiedad por defecto, mejor dicho, el valor por defecto de la propiedad excedente, okey y además también santa perfecto hasta ahí entonces tenemos las herramientas de posicionamiento básicas de "flex box", no por básicas menos importantes, de hecho, éstas son las que hay que dominar ciento por ciento antes de entrar con otro tipo de detalles.

Bien, por supuesto tenemos también la posibilidad de, por ejemplo, tener mucho más que cuatro elementos y ahí, ya empezaremos a ver otros conceptos. ¿Qué ocurriría si aquí, en vez de tener, bueno, pues no sé, cuatro, tuviésemos, aquí tengo, 20? Vamos a copiarlos, 20. Okey, aquí vemos cómo se acomodan todos en una fila. Dijimos que, por defecto, "flex box" trabaja en filas, aunque eso luego lo vamos a cambiar, si es que lo necesitamos, pero no tiene ustedes que ya estos "div" interiores no tienen los 150 píxeles de ancho que le indicamos, ¿por qué?, porque una fila no tiene ese espacio. Si ampliamos un poco aquí la pantalla, veremos que van creciendo y se ajustan de acuerdo al tamaño disponible, pero nunca llegan a tener el tamaño especificado y cuando la pantalla tiene menos espacio que lo que necesitan para mostrarse los "div", bueno pues, nuestro diseño se rompe. Para evitar esto y para que cada uno de los de estas cajas mantenga el tamaño que le indicamos, vamos a indicar que cuando haya más elementos que el espacio disponible en una fila, se

Cree una nueva fila debajo y así sucesivamente, hasta acomodar todo el contenido. Esto podemos lograrlo con la propiedad “flex wrap” que por defecto está en “no wrap” y poniéndola en precisamente “wrap”, okey. Les dije que esto era potente ¿verdad? y sí, es muy potente. Bien, por supuesto estas propiedades pueden revertirse, es decir, pueden ordenarse al revés. Okey, fíjense que aquí estamos con “wrap reverse” y estamos empezando desde abajo, leyendo siempre de izquierda a derecha, ¿cierto? Uno, dos, tres, cuatro, cinco, seis, siete, bla, bla, bla, etcétera, etcétera.

Bien, tal vez ustedes, en algún momento, encuentren una utilidad para la posibilidad del “wrap reverse”, sepan que está allí por si necesitan utilizarla. Muy bien vamos a eliminar ahora algunos de estos elementos, no todos, para ver otras propiedades muy interesantes. Podríamos aquí hacer hasta 15 filas, ¿les parece?, por ejemplo, okey, perdón, 15 filas no, quise decir 15 elementos. Bien, vamos a ver, ahora entran en juego por ejemplo la propiedad online content no la hemos visto hasta ahora pero ahora podemos verla porque no la vimos antes porque a diferencia de “align items”, “align content” solamente puede aplicarse, o solamente sus valores tienen un reflejo en nuestro diseño, cuando tenemos más de una fila. Es decir, cuando la propiedad “flex wrap” está activada, sino, no tiene ningún efecto. Aquí, por ejemplo, podríamos indicar, vamos a comentar el “align items”, okey, ahí no tenemos ninguna, ninguna indicación específica y vamos a indicar que queremos alinear el contenido y también querían, queríamos enviarlo, por ejemplo, al centro. Bien, fíjense que funciona en otro eje igual que la propiedad ya “justify content” lo hace en el eje principal, ¿correcto? Bien, entonces, ahí tenemos “center”, también podremos indicar “flex end” y va a ir todo el contenido abajo. Noten que todas las filas se mueven en conjunto ¿verdad? También podríamos indicar “space around”, “space between”, “space evenly”, ¿bien? Y ahí pueden ver ustedes entonces cómo funciona de la misma manera que funciona “justify content”. Ahora qué ocurre si “flex wrap” la comentamos o, miren, la ponemos en “no wrap”, que es el valor por defecto. Bueno, ya no se está aplicando “align content”, porque como les indiqué, solamente se aplica cuando hay más de una fila, es decir, cuando “flex wrap” está activada o está con su valor “wrap”, ¿okey? Si se aplicaría, por ejemplo, esta propiedad “align items” sí va a aplicarse, perfecto. Ahora vamos a poner nuevamente “flex wrap” para que nuestro diseño se ajuste de esta forma y aquí, por ejemplo, vamos a eliminar el “align items” y vamos a dejar el “align content”. Muy bien, ustedes van a notar, por ejemplo, alguna diferencia si ponemos “align content” en “center”, este va a ser el resultado y si ponemos “align items”, el resultado es distinto, ¿notan ustedes?, okey.

Aquí les recomiendo que practiquen usted es bastante para notar sobre la práctica exactamente cuál es la propiedad que desean aplicar. Aquí, por ejemplo, podríamos hacer que cada una de estas cajas sea un poquito más, más bajita y al tener más espacio de contenedor, vamos a ser más conscientes de los cambios. Cómo se aplican y de la diferencia, por ejemplo, entre “align items” con la posición “center” y “align content” con la posición “center”.

Vamos a ahora hacer al revés, dejamos “align content” en vez de “align items” y todos se van al centro. Ahora ustedes podrían decir, con total razón, “este diseño me sirve para lo que quiero hacer”, pero claro, el problema es que me quedan pegadas y no quiero, bueno, recuerdan que les dije que teníamos “column gap” y se está aplicando, saben que cuando hay más de una fila, por supuesto podemos utilizar “row gap”, “row gap” y también podríamos darle un “rem” u otro valor, no importa, es indistinto.

Bien y aquí podemos ver entonces cómo fluyen nuestros elementos. Tenemos un poco más de “flex box”, pero vamos a dejarlo para la próxima clase.

