
*** CSS Parte 3***

__ Conceptos Basicos Responsive __

Se usa para adaptar una pagina a diferentes resoluciones en pantalla ya sea movil, tablet, lapto o pc de escritorio con el resposive hacemos que la parte visual de la pagina se comporte diferente dependiendo de a que resolucion se encuentre y de esta manera sus elementos se acomoden de forma diferente en el espacio de la pagina.

Se basa en adaptar la pagina de resoluciones grandes a pequeñas.

Mobile - First : El Mobile - first por su parte se encarga de adaptar la pagina pero de resoluciones pequeñas a resoluciones mas grandes, es decir de resolucion de movil a una resolucion de escritorio o cualquier otra resolucion que sea mas grande que la de escritorio.

Para trabajar con responsive design se usa una propiedad llamada media Query, los mediaQuery son en los que especificamos la resolución en la que se aplicará el cambio de la página dependiendo de la resolución, los media Query se encargan de detectar la resolución en que se encuentra la página y aplicar esos cambios la forma de aplicar los media Query es ("@media screen and(resolución que debe detectar)").

__ FlexBox __

El uso de las cajas flexibles se emplea para acomodar los elementos en la página y sustituir a las tablas.

flex necesita de dos cosas que son un flex-container y un flex-item.

a.flex-container : Es el contenedor donde se encuentra todos los items que se desean posicionar con flex, cuando le damos display flex al contenedor se comporta como bloque ese contenedor los cambios se notan adentro del contenedor donde se necesita que los elementos que están dentro sean flexibles y de esta manera poder acomodarlos.

b.flex-items : Son los elementos que están dentro del contenedor que se le aplicó el display flex pero solo los elementos hijos del que posee el display flex si los elementos hijos tienen otros elementos dentro de ellos a estos no les afecta el display flex a los que si afecta directamente el display flex son a los hijos directos del contenedor haciéndolos flexibles para poder movilizarlos dentro del contenedor.

_ Ejes del flex box _

FlexBox posee dos ejes, así como en un plano hay los ejes (x, y) en flexbox tenemos los ejes (main axis(x) y cross axis(y)), los 2 vienen siendo a la dirección a la que apuntan por ejemplo.

main axis(x) apunta en el eje horizontal y va de izquierda a derecha que viene siendo (main-start(izq) y main-end(der)).

cross axis(y) apunta en el eje vertical y va de arriba hacia abajo que viene siendo (cross-start(arr) y cross-end(aba)).

Con el flex-box lo que hacemos es invertir las direcciones en las que apunta el main axis y el cross axis haciendo que se inviertan las posiciones en las que apuntan por ejemplo que main axis apunte en vertical y el cross axis apunte en horizontal y dependiendo hacia donde sea la direccion a la que apuntas de donde sale sera el start y hacia donde va sera el end de cada uno (main y cross).

___ Display Flex ___

Cuando aplicamos Display flex las cajas dentro del elemento con la propiedad flex se adaptan de acorde a su contenido y se posicionan una a lado de otra pero sin el problema de que un contenedor hijo sea mas alto que el otro por que este tenga mas o menos contenido, los contenedores hijos se adaptaran con el mismo tamaño.

Si un contenedor hijo tienes mas contenido que otro se adapta el contenido para que sigan teniendo la misma altura.

Cuando se usa flex si no se le da un width al item no sera visible dentro del contenedor esto es una particularidad de flex.

-- Propiedades de Flex --

_ Flex Direction _

Es una propiedad que se le aplica al contenedor pero afecta a los items y se usa para modificar la direccion del "main axis".

1.row : Es el valor que trae por defecto y hace que los items se comporten como si estuvieran en una fila.

2.row-reverse : Cambia la direccion en el mismo eje pero ahora apuntara de derecha hacia la izquierda invierte la fila como si fuera un espejo.

3.column : Hace que los items se comporten como si estuvieran en una columna, es decir el "main axis" corre en la direccion que corria el cross axis.

4.column-reverse : Cambia la direccion en el mismo eje pero ahora apuntara de abajo hacia arriba invierte la columna como si fuera un espejo.

`_ flex-wrap _`

1.flex-nowrap : Es el valor por defecto y es la ausencia de la propiedad.

2.wrap : Hace que al reducir el espacio del contenedor los items se acomoden y bajen a la siguiente linea cuando el espacio no es suficiente para su tamaño y para que no pierda el ancho que se le dio al item cuando llegue al espacio limite donde los items es uno a lado de otro y ya no halla mas espacio se pasa a la linea de abajo el item que este es la equina y asi ningun item perdera su ancho.

3.wrap-reverse : Funciona casi igual al flex-wrap pero en vez de irse para abajo se va hacia a arriba la caja de la esquina si no hay espacio suficiente para el tamaño de todos los items.

__ Flex-Flow __

Es un acortador de las propiedades flex-direction y flex-wrap en donde puedes incluir valores para ambas propiedades.

__ Justify-Content __

Trabaja en el main axis y se usa para posicionar los items dentro del contenedor, es una propiedad que se le aplica al contenedor pero afecta los items.

1.center : Se usa para centrar los items dentro del contenedor.

2.space-around : Funciona como el margin-auto dividiendo el espacio que hay en el contenedor entre los items, pero lo unico es que el space-around solo centra en un solo eje y el margin-auto centra en los dos y se usa para darle un margen automatico a todos los items dentro del contenedor y que todos items tengan el mismo espacio de separacion que se le da por ambos lados, excepto en las esquinas, en el principio y al final los margenes son mas pequeños.

3.space-between : Hace que todos los items esten separados lo mas posibles entre ellos a la misma distancia lo mas posible dentro del espacio del contenedor y los items de las esquinas solo se le da el margen del lado donde halla que separarlo de otro item del lado que esta pegado del contenedor no se le otorga margen.

4.space-evenly : Funciona parecido a space-around solo que esta propiedad le da un margen especifico a todos los items para que sea igual de los dos lados tanto de izquierda como de derecha incluso en los lados de las equinas que estan pegadas al inicio y final del contenedor.

__ Aling-Items __

Se usa solo cuando hay una sola linea de items que son flex-item y se usa para mover los items en el cross axis del contenedor.

1.stretch : Es la propiedad por defecto asi que cuando se esta empleando esta propiedad y no se le otorga un height especifico al item ocupa todo el alto del contenedor.

2.flex-start : Se usa para mover los items hacia el principio del cross-axis, es decir el punto de partida hacia donde este apuntado el cross-axis desde donde arranca.

Cuando se aplica flex-start el item si no se le otorga un height especifico solo ocupara un alto del tamaño de su contenido.

3.center : Se usa para centrar los items dentro del contenedor en el eje cross-axis.

4.flex-end : Funciona casi igual al flex-start solo que este mueve los objetos hacia el punto final de donde apunta el cross-axis, es decir el punto de llegada.

5.Baseline : Funciona parecido a flex-start, alinea los items en el contenedor segun la base del contenido de los items del contenedor.

Su uso mas comun es cuando hay un wrap-reverse y se desea que los item se que vallan pasando a otra linea pasen encima de los otros, entonces se combinan Baseline con el wrap-reverse para esto mandara los items para abajo como si se tratara de un flex-end.

__ Aling-Content __

Se usa cuando hay mas de una linea, despues de la segunda linea deja de funcionar el align-item y es hay donde debemos de aplicar el align-Content para que se haga cargo de poder aplicar las propiedades a las lineas que se vallan creando dentro del contenedor.

Aling-Content funciona con las misma propiedades que Aling-Items solo que estas se aplican para mas lineas.

1.stretch : Es la propiedad por defecto asi que cuando se esta empleando esta propiedad y no se le otorga un heigth especifico al item ocupa todo el alto del contenedor.

2.flex-start : Se usa para mover los items hacien el principio del cross-axis, es decir el pundo de partida hacia donde este apuntado el cross-axis desde donde arranca.

Cuando se aplica flex-start el item si no se le otorga un heigth especifico solo ocupara un alto del tamaño de su contenido.

3.center : Se usa para centrar los items dentro del contenedor en el eje cross-axis.

4.flex-end : Funciona casi igual al flex-start solo que este mueve los objetos hacia el punto final de donde apunta el cross-axis, es decir el punto de llegada.

5. Baseline : Funciona parecido a flex-start, alinea los items en el contenedor segun la base del contenido de los items del contenedor.

Su uso mas comun es cuando hay un wrap-reverse y se desea que los item se que vallan pasando a otra linea pasen encima de los otros, entonces se combinan Baseline con el wrap-reverse para esto mandara los items para abajo como si se tratara de un flex-end.

-- Propiedades de los Items --

Son propiedades que se le dan directamente a los items de esta manera podemos ajustar o modificar el item sin necesidad de aplicarle las propiedades al contenedor.

Estas propiedades se usan para aplicarle cambios a los items de manera individual, de esta forma podemos escoger un solo item y aplicarle los cambios a ese item en especifico y no de manera grupal con todos los items del contenedor como con las propiedades (Justify-Content - Aling-items - Aling-Content).

El margin se comporta de forma diferente cuando se usa flex funciona alreves, esto quiere decir que si le decimos que se mueva a la izquierda ira a la derecha, si le decimos que valla hacia arriba ira hacia abajo.

Estas propiedades solo se pueden aplicar a los items si su contenedor es flex.

1. align-self : Se usa para mover un item especifico en el cross-axis el item con align-self se movera independientemente de los otros items del contenedor.

Su valores tambien son (stretch - flex-start - flex-end - Baseline - center)

2.flex-grow : Toma el espacio que sobra en el contenedor y lo reparte entre las cajas asi crecen en tamaño las cajas, se puede hacer que lo reparta en partes iguales para cada caja o repartir de forma desigual, es decir dar mas a una caja que a otra.

Si se aplica el flex-grow al contenedor la reparticion del espacio sobrante sera equitativa pero si se desea que una caja tenga mas espacio que otra se le debe otorgar directamente la propiedad flex-grow al item que se quiere que se quede con mas espacio que los demas.

Individualmente se puede dividir el espacio entre las cajas dandole valores al flex-grow en que cantidad de espacio del que sobra quieres que tenga cada caja.

3.flex-shrink : Se usa para que una caja ceda mas espacio que otra cuando se este haciendo el contenedor mas pequeño, la caja con flex-shrink dependiendo del valor que se le de cuando ya no halla suficiente espacio para mantener el ancho de todas las cajas que halla en una linea y no se aplico la propiedad flex-wrap entonces no se moveran las cajas y se haran mas pequeñas de lo normal entonces es hay donde flex-shrink hace que la caja con esta propiedad ceda mas espacio que las demas cajas haciendo a esta mas pequeña y las otras conservaran mas tamaño.

Al momento de volver a su tamaño original de ancho donde ya halla espacio suficiente para todas las cajas volveran todas al mismo tiempo al ancho que se les otorgo inicialmente sin importar que tengan flex-shrink.

Tambien con esta propiedad se puede hacer no solo que ceda mas espacio si que tambien lo contrario que ceda menos espacio que las demas cajas.

4.flex-basis : Funciona como si fuera un width, se usa para darle un ancho a los items pero el css toma a flex-basis como mas importante que el width, es decir si se aplican los 2 tanto el width como el flex-basis se le dara prioridad al flex-basis.

5.flex : Es un acortador de las propiedades (flex-grow - flex-shrink - flex-basis) y empieza a funcionar a partir de un solo parametro que se le pase que es el flex-grow.

6.order : Funciona como el z-index pero en el eje en el que apunta el main-axis, esto quiere decir que el que posea el valor mas alto estara al final de donde apunte el main-axis.

Si se cambian la direccion donde apunta el main-axis cambiara la posicion en la que se posiciona el item con el valor mas alto.