**VIDEO 01=** FLEXBOX es uno de los dos grandes módulos de css que tenemos actualmente para hacer maquetación en páginas web. El otro módulo de maquetación es GRIP. Flexbox tiene alrededor de 11 o 12 propiedades de maquetación principales, unas se aplican al elemento hijo y otras al padre. TIPS= **GUIA COMPLETA DE FLEXBOX EN CSS TRICKS**. Es importante conocer flexbox porque los grandes framework de css como lo son **bootstrapCDN y foundation** trabajan como base con fleXbox. TIPS= Descargar extensión en el VSC llamada flexbox cheatsheet, que te ayuda muchísimo ya que es como un resumen de todas las propiedades de flexbox y que hacen, ***se abre usando control+shift+p .***

**VIDEO 02=** Aunque pareciera que flexbox nos da columnas y filas y que Bootstrap y foundation generan su grip con flexbox realmente no es así, flexbox es un sistema unidimensional vamos a tener o filas o columnas, pero no ambas. Esa estructural bidimensional la tenemos es con grip. Esta terminología básica que se ve en la imagen es el diagrama oficial que hizo la gente de la w3c sobre flexbox. Entonces vamos a tener nuestro contenedor que es el flex container, los elementos hijos que es a los que se les aplica el comportamiento de flexbox, llamados flexitem. Y vamos a tener dos ejes el eje principal llamado main axis y el eje transversal llamado cross axis, que en la mayoría de los casos el eje principal es x y el eje transversal es y pero hay una propiedad que nos permite cambiar esto y hacer que sea al revés. El tamaño del eje transversal (cross size) es definido por y y el tamaño del eje principal (main size) seria definido por x.Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Empecemos por la primera propiedad que es:

**DISPLAY:** Esta propiedad es la que activa las propiedades de flexbox. Tenemos dos posibles valores para esta propiedad una es **FLEX BLOCK Y EL OTRO ES INLINE-FLEX.**

Si usamos **DISPLAY: FLEX** el comportamiento por defecto de esta caja es que los elementos hijos se van a alinear en fila de forma horizontal uno al lado del otro. Si hubiera otra caja con la misma propiedad la va a aplicar abajo como en cajas en bloque. Display flex es una caja flexible que trabaja en bloque.

Si usamos el valor **DISPLAY: INLINE-FLEX** Los contenedores que son flexbox se van a comportar como elementos en línea, es decir van a ocupar solamente el ancho que necesiten su contenido, en este caso sus tres elementos hijos.

**VIDEO 03=** La siguiente propiedad es FLEX-DIRECTION: Tenemos 4 valores por defecto para esta propiedad.

**(row)=>** ubica los elementos ordenados en fila

**(row reverse)=>** invierte los elementos

**Column=**> columna

**Column reverse=>** invierte el orden de la columna y pasan a estar abajo del contenedor

Cuando usamos flex-direction el valor por defecto es row así no la escribamos aplica ese valor.

Diagrama

Descripción generada automáticamente con confianza media

La siguiente propiedad es la propiedad FLEX-WRAP: Esta propiedad traducida significa envoltorio, es la que dice si los elementos se comportan alineados en una sola línea o van a considerar el ancho que tengan cada elemento y van a hacer alusión como si tuviéramos varias filas o varias columnas dependiendo la dirección. **El valor por defecto es nowrap=>** es decir no envuelvas, es que todos sus hijos se alineen en una sola fila o en una sola columna dependiendo de la dirección que tenga predeterminada en la propiedad flex-direction.

Forma, Cuadrado

Descripción generada automáticamente con confianza mediaUn dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

Si nuestro elemento item en el caso del ejemplo tuviera ancho y alto, y va a tratar siempre de mantener en una sola línea o columna a todos los elementos y si reduzco la pantalla igual no se van a pasar hacia abajo, sino que se reduce el tamaño sin respetar el ancho que debe tener.

El otro valor que tiene esta propiedad **es WRAP=>** Va a envolver los hijos, y va a considerar el tamaño el ancho y el alto del contenedor padre y si no logran entrar todos los hijos en una misma fila o columna allí el los va a envolver y va a empezar a formar varias filas

Tabla

Descripción generada automáticamente

Esta propiedad es la que nos hace pensar que tenemos filas y columnas, pero la realidad es **que flexbox es un sistema unidireccional** que solo nos da filas o columnas, y esta apariencia de que tenemos dilas y columnas nos la da la propiedad flexwrap con el valor wrap que los envuelve formando filas o columnas dependiendo del ancho, alto y de la dirección que tenga.

Adicionalmente tenemos **el valor wrap-reverse=>** que los envuelve, pero en el sentido contrario.

**Esto de la propiedad flex-direction y flex-wrap** es lo que se conoce como flujo de la caja, que es como se van a comportar los elementos adentro de la caja. Hay una combinación de esas dos propiedades que es lo que se **llama FLEX-FLOW al cual podemos ponerle dos valores EL PRIMER VALOR SERIA PARA LA DIRECCION Y EL SEGUNDO EL ENVOLTORIO. Es un shortcut de flexdirection y flexwrap**



**VIDEO 04=** En este video nos explican 3 propiedades que son las que nos permiten alinear los elementos, ósea como distribuirlos dentro de la caja flexbox.

**JUSTIFY-CONTENT=>** Define la alineación de los elementos hijos en el eje principal que puede ser x o y el que determina cual es el eje principal sea x (filas) o y (columnas) es flex-direction. El valor por defecto es **flex-start.** Pero también justify content admite estos 4 valores:

**Flex-start=>** Los elementos se alinean a la izquierda de la caja pegados como si fuera un Flow left.

**Center=>** Se centran en el medio de la caja. Se centran en el alto

**Flex-end=>** Con este se alinean a la derecha como si fuera un Flow rigth. En el caso de las columnas son todos los elementos abajo. En el caso de las columnas son todos los elementos arriba

**Space-between=>** ignora las orillas de la caja contenedora y el espacio que sobre lo reparte entre los elementos hijos. No considera las orillas igual que en filas en columnas

**Space-around=>** Considera las orillas, pero solo a la mitad. Lo mismo para las columnas

**Space-envenly=>** Vino con grip css y le da le mismo valor a las orillas que a los elementos internos. Lo mismo para las columnas.

Todo esto de la alineación aplica solo si me sobra espacio en la caja padre, si no me sobra espacio no se alinean y no aplica esta propiedad.

Interfaz de usuario gráfica, Icono

Descripción generada automáticamente

**VIDEO 05=** En este video se nos habla de cómo alinear en el eje transversal es decir el eje sobrante. Para ello tenemos dos propiedades **ALIGN-ITEMS Y ALIGN CONTENS.**

En el caso de **ALIGN-ITEMS=>** Los valores por defecto son

**Flex-start=>** los hijos se alinean arriba

**Flex-end=>** los hijos se alinean abajo

**Center=>** en el medio

**Stretch=>** Es el valor por defecto. Significa que los hijos de una caja flexbox en el eje transversal es decir el que no es el principal, se van a estirar al tamaño del contenedor padre

**Baseline=>** Todos los textos se ubican al mismo nivel sobre una línea imaginaria independientemente de su tamaño.

Esta propiedad trabaja por cada línea que tengamos. **ALIGN ITEMS FUNCIONA PARA ALINEAR LOS HIJOS EN EL EJE TRANSVERSAL LLAMADO CROSS AXIS. CUANDO EL EJE PRINCIAL ES FILA EL ALINEA EN COLUMNAS Y CUANDO EL EJE PRINCIPAL ES COLUMNAS ALINEA EN FILAS.**

Interfaz de usuario gráfica

Descripción generada automáticamente

**LA DIFERENCIA** ENTRE JUSTIFY-CONTENT Y ALIGN-ITEMS=> ES QUE JUSTIFY CONTENT TRABAJA LA ALINEACION DE LOS HIJOS EN EL EJE PRINCIPAL EN CONJUNTO, MIENTRAS QUE ALIGN ITEMS TRABAJA LA ALINEACION DE LOS HIJOS EN EL EJE TRANSVERSAL PERO DE MANER INDEPENDIENTE CADA FILA O CALA COLUMNA.

Si queremos alinear los elementos hijos en el eje transversal como un todo tenemos otra propiedad llamada **ALIGN-CONTENT=>** Se parece mucho a justify content pero la diferencia es que en vez de hacerlo en el eje principal lo hace en el eje transversal. (cross axis). No funciona cuando los hijos están en una sola línea. Tiene todos estos valores esa propiedad**. Hay que colocar el valor de wrap ya que no trabaja en una sola fila o columna. Y de esta manera tendremos mas de una fila o columna. VALOR POR DEFECTO FLEX-START**

Aunque en el cuadro no este especificado también admite el valor de align-evenly. Y RECORDAR QUE CUANDO FLEX-WRAP ES NO-WRAP NO FUNCIONA ALIGN CONTENT SINO ALING ITEM. SOLO FUNCIONA CON WRAP O WRAP-REVERSE

Interfaz de usuario gráfica

Descripción generada automáticamente

**VIDEO 06=** En este video se empiezan a ver tres propiedades que afectan a los hijos.

**FLEX-GROW => FACTOR DE CRECIMIENTO /VALOR POR DEFECTO 0.** Cuando la caja flexbox tenga espacio sobrante, los hijos van a poder aprovechar ese espacio sobrante porque flexgrow es la habilidad o el factor de crecimiento, esta propiedad los dota de poder crecer más y aprovechar ese espacio sobrante, no se aceptan valores negativos.

**FLEX-SHRINK=> FACTOR DE REDUCCION /VALOR POR DEFECTO 1.** Cuando la caja flexbox es muy pequeña y no tiene espacio sobrante, es la habilidad que tienen los elementos hijos de encogerse o reducirse, no se aceptan valores negativos.

**FLEX-BASIS=> VALOR POR DEFECTO AUTO.** Es el tamano del elemento hijo dentro de la línea de la caja flexbox. Si la caja flexbox tiene dirección de fila, flex basisc representa width. Si la caja flexbox tiene dirección de columna, flex-basisc representa el height.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si aplicamos un factor de crecimiento flex-grow 1 se aprovecha proporcionalmente en iguales partes el espacio sobrante. Pero si queremos que uno solo de los ítems crezca mas proporcionalmente a los otros se le aplica solo a ese ítem un factor 2.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**VIDEO 07= FLEX-SHRINK=> FACTOR DE REDUCCION /VALOR POR DEFECTO 1.** Cuando la caja flexbox es muy pequeña y no tiene espacio sobrante, es la habilidad que tienen los elementos hijos de encogerse o reducirse, no se aceptan valores negativos.

Texto

Descripción generada automáticamente

**VIDEO 08= FLEX-BASIS=> VALOR POR DEFECTO AUTO.** Es el tamaño del elemento hijo dentro de la línea de la caja flexbox. Si la caja flexbox tiene dirección de fila, flex basisc representa width. Si la caja flexbox tiene dirección de columna, flex-basisc representa el height. Es decir, cuando la dirección de la caja sean fils flex-basisc representa el ancho. Y cuando la dirección de la caja es columnas flex-basisc representa el alto.

**TENEMOS UN SHORTHAND DE LAS TRES PORPIEDADES:**

**La propiedad se llama FLEX: 0 1 AUTO**

flex-grow: 0;

    flex-shrink: 1;

    flex-basis: auto;

**Estas tres las podemos abreviar en la propiedad flex**

 flex: 0 1 auto;

**sí lo colocamos así si crecerá al espacio proporcional y se reducira cuando lo necesitemos**

    flex: 1 1 auto;

**VIDEO 09= ORDEN Y ALINEACION DE HIJOS**.

**ORDER=>** Esta propiedad nos va a controlar el orden que tendrán los elementos hijos en la caja flexbox. **Valor por defecto cero (0).** Se aceptan valores positivos y negativos. Un valor menor siempre ira antes que un valor mayor.

**ALIGN-SELF=>** Esta propiedad se aplica a los hijos a diferencia de ALIG-ITEMS QUE SE APLICA AL CONTENEDOR. Toma los mismos valores que align ítems, **su valor por defecto es stretch.** Ella va alinear al elemento al que se le aplique en el eje transversal y no el principal (croos axis).

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**VIDEO 10=** En este video se nos explica cómo usar flexbox para la maquetación y responsive en el desarrollo web. Al día de hoy existen diferentes frameworks (estructura), que se utilizan para agilizar esto entre los más conocidos están Bootstrap y fundation. Pero para entenderlos lo primero es saber **que es un framework=>** Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

Utilizar frameworks puede simplificar (y mucho) una tarea o proceso, de ahí que se trate de una de las herramientas habituales que manejan los Digital Workers, porque les ayuda a ser más ágiles y productivos. Actualmente estos framework trabajan con flexbox y grip.

Uno de los atractivos de estos framework es como podemos hacer filas y columnas.

USANDO BOOTSTRAP SIMPLIFICAMOS EL CODIGO Y PODEMOS HACER ESTO SIN UTILIZAR CSS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

SIN BOOTSTRAP LO HARIAMOS DE ESTA MANERA. AGREGANDO LAS MEDIA QUARY YA QUE CON BOOTSTRAP LO TIENE INCLUIDO, PERO SIN BOOTSTRAP HAY QUE AGREGARLAS AL FINAL DEL CSS

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**VIDEO 11= EJERCICIO 01**

COMO LOGRAR UN PIE DE PAGINA Y UNA CABECERA DE PAGINA CON FLEXBOX SIN COMPLICARNOS CON POSITION.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**EJERCICIO 02** MENU CON ITEMS Y LOGO

**TIPS=>** PARA QUITAR LAS VINETAS DE LOS ITEMS DE UNA LISTA USAMOS

.menuul {

    list-style-type: none;

}

Con align item centre la caja y con text align centre el texto.

.header {

    display: flex;

    flex-direction: column;

    align-items: center;

    text-align: center;

}

El navegador a las li les da un margin por defecto y un padding por defecto hay que restaurarlos a cero para que se vean centrados

Acá le estoy diciendo que cuando llegue a un ancho de 1024 header pase de columna a fila

@media screen and (min-width:1024px) {

    .header {

        display: flex;

        flex-direction: row;

    }

Si yo quiero que el menú este de un lado y el logo de otro puedo aplicar esta propiedad para que el espacio sobrante lo reparta entre ambos. Justify-content

 @media screen and (min-width:1024px) {

        .header {

            display: flex;

            flex-direction: row;

            justify-content: space-between;

        }

Así usando media quari se logra una cabecera vertical para móvil y horizontal para web.

Forma, Rectángulo

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**EJERCICIO 3=>** PONER IMAGEN DE FONDO DEL CONTENIDO

**TIPS=>** PAGINA DE DONDE PUEDO TRAER IMAGEN LOS ULTIMOS VALORES NUMERICOS CORRESPONDEN AL ANCHO Y ALTO Y LA ULTIMA PALABRA ES LA CATEGORIA

background-image: url("https://placeimg.com/1980/720/animals");

colocando el tamaño de letra en unidades de vw hace que crezcan con la pantalla

.hero-imageh1 {

        font-size: 5vw;

    }

Esto nos ayuda a centrar en todos los ejes lo que queremos

.hero-imagediv {

        background-color: rgba(0, 0, 0, .75);

        min-width: inherit;

        display: flex;

        justify-content: center;

        align-items: center;

        text-align: center;

    }

Gato parado junto a un caballo

Descripción generada automáticamente con confianza media

**VIDEO 12=**

**EJERCICIO**