FLEXBOX

1. Introduccion a flexbox

Implementación de herramienta CSS Flexbox Cheatsheet en visual studio code, tambien, la guias de https://jonmircha.com/flexbox y https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Conceptos basicos

- **Contenedor Padre (*Flex Container*).**
- **Elementos Hijos (*Flex Items*).**
- **Eje Principal (*Main Axis*)= la mitad horizontal del contenedor (eje x)**
- **Eje Transversal (*Cross Axis*)= la mitad vertical del contenedor (eje y)**
- **Ancho del contenedor (main size).**

inicio del ancho (main start)

fin del ancho (main end)

- **Alto del contenedor (cross size)**

inicio del alto (cross start)

fin del alto (cross end)

Flexbox: es un módulo de diseño CSS que facilita la distribución de espacio y alineación de elementos dentro de un contenedor.

- Flexbox trabaja en una sola dirección (horizontal o vertical) para distribuir los elementos (Es unidimensional).
- Habilita un contexto flexible para todos sus elementos secundarios directos.

Elementos Flex:

Los elementos hijos directos del contenedor flex, se pueden manipular para modificar su tamaño, posición y alineación.

Ventajas:

Permite una mayor flexibilidad en la alineación y distribución de elementos.

Simplifica la creación de diseños response

Flexbox se activa con dos valores en la propiedad display:

a) display: flex; el comportamiento por defectos de los elementos hijos, es que se pondrán en fila, y el contenedor por defecto trabaja como bloque ocupando todo el ancho disponible (caja flexible en bloque)

b) display: inline-flex; el comportamiento por defectos de los elementos hijos, es que se pondrán en fila, y el contenedor funcionara como un elemento en línea.(caja flexible en línea)

Es decir, si hay dos contenedores Flex y queremos ponerlos uno al lado del otro, usamos **inline-flex**

2. Flujo de caja de flexbox

recordar que flexbox es unidimensional es decir o se trabaja en fila o en columna

a) **flex-direction:** es la dirección dentro de la caja, en fila o en columna define el valor en que trabajara la caja, por defecto trabaja en fila

tiene 4 valores

row: elementos en fila y de lado izquierdo de la caja (es el valor por defecto)

row-reverse: invierte el orden de los elementos y los coloca a la derecha de la caja

column: ordena los elementos en columnas y en la parte superior de la caja

colum-reverse: ordena los elementos en columnas invirtiendo su posición y colocándolos en la parte inferior de la caja

b) **flex-wrap:** es a propiedad que envuelve los elementos hijos los alinea según la dirección del flex-direction.

tiene 3 valores

wrap: envuelve los elementos dentro del contenedor padre

wrap-reverse: envuelve los elementos de manera inversa a su posición.

nowrap: valor por defecto, donde no envuelve los elementos hijos y los alinea segun el flex-direction

Existe un shordhad que engloba estas dos propiedades:

flex-flow: flex-direction flex-wrap

Nota: el tamaño de los elementos hijos ancho, alto, van estar en función de la dirección del flujo de caja (flex-direction), es decir, si están en fila (row "valor por defecto") el ancho se ajustará al contenido y el alto expandirá ocupando todo el alto disponible (en caso que no estén declarados).

Si están en columnas, ocurre lo contrario el alto se ajustará en función del contendió y el ancho se expandirá ocupando todo el espacio disponible, porque los ejes cambian (*Main Axis (x), Cross Axis* (y).

Ejemplo:

HTML

```
<body>
  <div class="container flex-1">
    <div class="items elemento-1">items-1</div>
    <div class="items elemento-2">items-2</div>
    <div class="items elemento-3">items-3</div>
    <div class="items elemento-4">items-4</div>
  </div>
   <div class="container flex-2">
    <div class="items elemento-a">items-A</div>
    <div class="items elemento-b">items-B</div>
    <div class="items elemento-a">items-C</div>
    <div class="items elemento-b">items-D</div>
  </div>
  <div class="container flex-3">
    <div class="items hijosflex-1">element-1</div>
    <div class="items hijosflex-2">element-2</div>
    <div class="items hijosflex-3">element-3</div>
    <div class="items hijosflex-4">element-4</div>
```

```
    <div class="items hijosflex-5">element-5</div>
    <div class="items hijosflex-6">element-6</div>
    <div class="items hijosflex-7">element-7</div>
    <div class="items hijosflex-8">element-8</div>
    <div class="items hijosflex-9">element-9</div>
    <div class="items hijosflex-10">element-10</div>
  </div>
</body>
```

CSS

```
*{box-sizing: border-box;}
.container{
    display: flex; /* se activa flexbox*/
    height: 30vh;
    gap: 5px;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8);}
.items{
    width:10%;
    height: 20%;
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px; }
.flex-1{
    flex-direction:row-reverse; /* se establece si se trabaja en fila o columna*/
    flex-wrap:wrap-reverse; } /* se establece si se envuelve */
.flex-2{
    flex-direction:column;  /*row / row-reverse / column / column-reverse */
    flex-wrap:wrap-reverse; /*nowrap / wrap / wrap-reverse */ }
.flex-3{ flex-flow: column-reverse wrap; }
```

3. Alineación en el Eje X, eje principal Main Axis

justify-content: propiedad que alinea y da espacios a los elementos de la caja flex, en función del flujo de la caja determinada por flex-direction--ojo por defecto trabaja definiendo la alineación de los hijos en el eje principal **(*Main axis*).**

posee 6 valores

flex-star: valor por defecto, que alinea los elementos en la parte superior izquierda

center: alinea los elementos en la parte central

flex-end: alinea los elementos en la parte superior-derecha

space-between: alinea los elementos ocupando los extremos de la caja y el espacio sobrante lo reparte entre las cajas internas

space-around: Al primer y último elemento solo se le asignará la mitad del espacio y los elementos internos tendrán el mismo espacio entre sí.

space-evenly: asigna el espacio vacío entre los elementos de manera igual.

MOVERA LOS ELEMENTOS SIEMPRE Y CUADO TENGAN ESPACIO DISPONIBLE

Nota: en caso de que se cambien flex-direction a columna cambiara a eje transversal (y) Cross Axis, y se moverá en función de esté.

Ejemplo con el anterior HTML:

En este caso de ejemplo veremos eje principal MAIN AXIS (X) determinado por defecto por (row)

CSS

```
.container{
    display: flex;
    flex-flow: row wrap;
justify-content:space-around;  /* flex-star / center/ flex-end /space-between / space-around / space-evenly*/
    height: 30vh;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8);}
```

```
.items{
    width:10%;
    height: 20%;
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px;}
```

NOTA: TODOS ELEMENTOS SI ALINEARAN EN EL CONTENER SIEMPRE Y CUANDO TENGAN ESPACIO--OJO

En este caso veremos CROSS AXIS (Y) determinado por column

CSS

```
.container{
    display: flex;
    flex-flow: column wrap-reverse; /* EJE PRINCIPAL DETERMINADO Y */
    justify-content:space-around; /* flex-star / center/ flex-end /space-between / space-around / space-evenly*/
    height: 30vh;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8);}

.items{
    width:10%;
    height: 20%;
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px;}
```

4. Alineación del Eje y, Cross Axis (y)

align-items: propiedad define el comportamiento predeterminado de cómo se disponen los elementos a lo largo del eje transversal (eje y) Cross Axis (y) (y perpendicularmente al eje principal (x) main axis).

posee 5 valores

- **stretch:** (valor predeterminado) Estira los elementos en el eje trasversal (y), cabe destacar, que funciona según el flujo de dirección determinado (flex-direction) para llenar estirarse en el contenedor, siempre y cuando no tengan un ancho o alto declarado.

ejemplo:

Sí tiene un flujo de dirección en fila **(row)** se determina el eje principal(x), y los elementos se estirarán en función del alto.

SÍ tiene un flujo de dirección en columna **(column)** se determina el eje trasversal(y), y los elementos se estirarán en función del ancho

- **flex-start:** los elementos se colocan en su parte inicial, se posicionan en la parte superior izquierda (cross start) del eje trasversal (y) Cross axis
- **flex-end:** los elementos se colocan en la parte inferior izquierda (cross end) del eje transversal (y), y si no tienen medidas declaradas se ajusta al tamaño del contenido.
- **center:** los elementos están centrados en el eje transversal(y), y si no tienen medidas declaradas se ajusta al tamaño del contenido.
- **baseline:** los elementos flexibles están alineados tal como se alinean sus líneas base, es decir, si un elemento tiene un contenido con base diferente a los demás, estos se ajustaran a la base de alineación de ese contendió, para que estén todos parejos en una misma línea de contenido.

Nota: el valor baseline no funciona cuando se trabaja el flujo de dirección en columna, solo funciona en fila

Ejemplo css:

caso con flujo de direccion por defecto row main axis eje principal eje (x):

```
.container{
    display: flex;
    flex-flow: row wrap; /* flujo determinado main axis eje x */
    justify-content: flex-start; /* se movera main axis eje x */
align-items: baseline;  /*main cross eje y, valores: stretch, flex-start, flex-end,
center, baseline */
    height: 30vh;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8);}
.items{
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px;}
.elemento-3{
    display: flex;
    align-items: center;
    width:10%;
    height: 25%;
   justify-content: center; }
.elemento-b{
    width:10%;
    height: 20%; }
.hijosflex-10{
    width:10%;
    height: 30%; }
```

Caso con flujo de dirección en columna eje transversal (y) cross axis:

```
.container{
    display: flex;
    flex-flow: column wrap; /* flujo determinado Cross Axis eje y */
    justify-content: flex-start; /* cross axis eje y ahora se movera en ese eje */
align-items: baseline; /* Cross Axis eje X, valores: stretch, flex-start, flex-end,
center, baseline */
    height: 30vh;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8); }
.items{
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px; }
.elemento-3{
    display: flex;
justify-content: center
   align-items: center;
    width:10%;
    height: 25%; }
.elemento-b{
    width:10%;
    height: 20%; }
.hijosflex-10{
    width:10%;
    height: 30%;}
```

align-content: es la propiedad que ajusta las líneas dentro de un contenedor flex cuando hay espacio extra en el eje transversal(y), es decir, trabaja por defecto en el eje transversal Cross axis, agrupando y moviendo todo el contenido en ese eje, como un todo, tiene los mismos valores que justify-content.(flex-star / center/ flex-end /space-between / space-around / space-evenly)

Nota: align-content funciona automáticamente con wrap o wrap-reverse, porque estos valores envuelven el contenido y automáticamente general varias líneas en el flujo de dirección.

Por lo tanto no funciona con nowrap porque solo tiene una línea de dirección, es decir, esta propiedad no tiene efecto en cajas flexibles de una sola línea.-- > ojo

Ejemplo:

```
.container{
    display: flex;
    flex-flow: row wrap; /* flujo determinado main Axis eje x */
    justify-content: flex-start; /* main axis eje x*/
align-content: center;  /*cross axis eje y /    /*flex-star / center/ flex-end /space-between / space-around / space-evenly*/
  height: 30vh;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8); }
 .items{
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px; }
```

5. Propiedades de los elementos hijos.

Factor de Crecimiento

- **flex-grow:** es la habilidad o el factor de crecer de los elementos hijos, Cuando la caja *flexbox* tenga espacio sobrante, valor por defecto es 0, NO se aceptan valores negativos.

  Los valores determinan el factor de crecimiento del elemento hijo, es decir crece estableciendo hasta cuanto creces en función de tu tamaño inicial, sin embargo, ocupará todo el ancho disponible.

nota: el elemento crece según el flujo de dirección determinado**--ojo**

ejemplo:

```
.container{
   display: flex;
   flex-flow: row wrap; /* flujo determinado main Axis eje x */
   justify-content: flex-start; /* main axis eje x*/
align-content: center; /*cross axis eje y /
  height: 30vh;
   margin-bottom: 10px;
   background-color: rgb(155 155 155 /0.8); }

.items{
flex-grow: 1; /* el valor fija el factor de crecimiento del elemento hijo, es decir crece una vez tu tamaño inicial, sin embargo ocupará todo el ancho disponible*/
   font-size:large;
   text-align: center;
   border-style: solid;
   border-radius: 10px; }
```

Factor de Reducción

- **flex-shrink:** es la habilidad o el factor de encogerse, cuando la caja *flexbox* NO tenga espacio sobrante, valor por defecto es 1(es decir, se

reduce el elemento una sola vez de su tamaño inicial), **NO** se aceptan valores negativos.

nota: el elemento se reduce según el flujo de dirección determinado**--ojo**

ejemplo:

```
.container{
   display: flex;
   flex-flow: row wrap; /* flujo determinado main Axis eje x */
   justify-content: flex-start; /* main axis eje x*/
align-content: center;  /*cross axis eje y /
  height: 30vh;
   margin-bottom: 10px;
   background-color: rgb(155 155 155 /0.8); }

.items{
flex-grow: 1;
flex-shrink: 2; /* fija el valor de reducirse hasta 2 veces su valor inicial*/
   font-size:large;
   text-align: center;
   border-style: solid;
   border-radius: 10px; }
```

Tamaño inicial de los elementos hijos.

- **flex-basis:** establece el tamaño inicial de un elemento flexible antes de que crezca o se reduzca, cabe mencionar, que el tamaño del elemento hijo se establece dentro del flujo de línea de la caja *flexbox (es decir segun su flex-direction)*.
- Si la caja *flexbox* tiene dirección de fila, *flex-basis* representa el *width*.
- Si la caja *flexbox* tiene dirección de columna, *flex-basis* representa el *height*.
- Valor por defecto *auto*.

nota: si flex-basis esta declarado, las propiedades de medida (width o heigth), no funcionan según el flujo de dirección principal determinado.---- > **ojo**

flex: es un shordhand que engloba las 3 propiedades **flex-grow, flex-shrink, flex-basis** en ese mismo orden.

Ejemplo:

```
.container{
    display: flex;
    flex-flow: row wrap;
    justify-content: flex-start;
    align-items: baseline;
    height: 30vh;
    margin-bottom: 10px;
    background-color: rgb(155 155 155 /0.8); }

.items{
    font-size:large;
    text-align: center;
    border-style: solid;
    border-radius: 10px;
    flex: 1 2 200px; }
```

nota: a pesar de tener un factor de crecimiento de 1 (donde aprovecha el espacio disponible y se expande todo el ancho del contenedor), y un ancho 200px (aplicado por basis) esos 200px representan el tamaño inicial del elemento antes del factor de crecimiento.

6. Orden y Alineación de hijos flexbox (order y align-self)

order: Representa el orden que tendrán los elementos hijos en la caja *flexbox*.

- Valor por defecto 0.
- Se aceptan valores positivos y negativos.
- Un valor menor siempre irá antes que un valor mayor.

align-self: Sobre escribe el valor de la propiedad *align-items* sólo para el elemento hijo especificado y por lo tanto, usa los mismos valores.