**CSS Flexbox**

Creamos nuestro HTML y CSS base para poder comenzar a estudiar las propiedades de Flexbox. Dichas propiedades serán estilizadas en el HTML para no mezclarnos.

    <div class="container">

      <div class="box box-1">1</div>

      <div class="box box-2">2</div>

      <div class="box box-3">3</div>

      <div class="box box-4">4</div>

      <div class="box box-5">5</div>

    </div>

body {font-family: Verdana, Geneva,

Tahoma, sans-serif; background: "f3f3f3"}

.container {border: 5px solid red}

.box {padding: 30px; text-align: center;

font-size: 35px; color: #f3f3f3}

.box-1 {background: #581f18}

.box-2 {background: #f0a202}

.box-3 {background: #d95d39}

.box-4 {background: #202c59}

.box-5 {background: #51cb20}

* Esta es la BASE.

**Flexbox**

Tendremos un Parten **(Flex-Container)** y los Children **(Flex-Items)**. En nuestro ejemplo tenemos un Flex-Container y 5 Flex-Items.

Para comenzar:

.container {display: flex}

* Flex 🡪 It is a BLOCK element.

.container {display: inline-flex}

* Inline-Flex 🡪 It is an Inline

Properties :

Generalmente las podemos separar en dos categorìas :

* Affect all Children (flex)
* Affect Individual Child (flex item)

Defaults en **Display: Flex :**

* Flex-Direction : Row
* Flex-Wrap : No-Wrap
* Justify-Content : Flex-Start
* Align-Items : Stretch

**All**

**Flex-Direction :**

Row 🡪 Default one (1 to 5).

Row-Reverse 🡪 Change direction, 5 to 1.

Column 🡪 Volvemos al Original (default sin habilitar Flexbox).

Column-Reverse  🡪 Original dado vuelta.

**Flex-Wrap :**

Para demostrar Wrap crearemos 5 nuevos divs sin importancia, para luego ser borrados.

**IF** our Flex Container is **ABLE** to fix boxes, Everything will be as before. The moment it is **NOT ABLE**, the user will need to scroll.

No-Wrap  🡪 Default

Wrap 🡪 When theres not enough space : It starts new line. (G, 1 to 5)

Wrap-Reverse 🡪 When there is NO space : Starts new line reversed. (G, 5 to 1 when NO SPACE).

Ej : **Nav Bar** When the screen is smaller we want to see all the items. We would use WRAP. (if not they would need to scroll)

**Justify-Content :**

Responsible for Horizontal Alignment.

Flex-Start 🡪 Default

Flex-End 🡪 Las boxes comienzan del otro lado (derecha) y de 5 to 1.

Center 🡪 Pone todo la centro sin espacios.Solo bordes Izq y Der.

Space-Between 🡪 1 and 5 Tocan las puntas (Izq y Der), luego harbá espacio entre medio de los items.

Space-Around 🡪 Crea margenes al rededor de todos los items. (L-item-R,L-item-R, etc.)

Space-Evenly 🡪 Crea margenes parejos para todos los lados.

* If we start a new line(wrap) : All Lines will be applied with the Justfy-Content chosen.

**Align-Items :**

Risponisble for Vertical Alignment. Necesitamos aclarar previamente un **Height** a la Parent.

Stretch 🡪 Estira los elementos para cubrir todo el Height (Default)

Flex-Start 🡪 Los elementos se alinean al TOP del container

Flex-End 🡪 Tocan el borde de abajo

Center 🡪 Se alinea Verticalmente al centro.

Baseline 🡪 Se alinea dependiendo el CONTENIDO (podrán sobrar bordes arriba o abajo) EJ : Alinear el menu a la misma altura que el logo.

**\*\*Mini Project : Hero Components\*\***