**CSS - FLEXBOX**

El módulo FLEX ( caja flexible) facilita el diseño de una estructura de diseño flexible y receptiva sin usar flotación o posicionamiento.

Vaya probando todas las propiedades del módulo FLEX en su Visual Code. Genere un documento HTML5 vacío, y dentro de su body vaya escribiendo los códigos html (color azul); en el head vincule su archivo html a una hoja de estilos CSS, son los códigos color rojo. Cuando están mezclados ambos el color es verde. Prueben, ejerciten, con las diferentes posibilidades que se mencionan en este texto.

Todo comienza definiendo una caja, o sea un contenedor, el cual diremos que es FLEXIBLE, por ejemplo en nuestro archivo CSS, podemos crear una clase

.flex-container {

display: flex;

background-color: DodgerBlue;

}

luego en el archivo HTML5 tendremos que tener un DIV, al cual le asignaremos esa clase….

<div class="flex-container">

</div>

dentro de ese div vamos a ubicar otros divs que por ser hijos del flex-container, serán flexibles.

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

</div>

Ya estamos en condiciones de manipular esos tres divs, por ejemplo….asignando estas propiedades:

.flex-container > div {

background-color: #f1f1f1;

margin: 10px;

padding: 20px;

font-size: 30px;

}

**PROPIEDAD FLEX-DIRECTION**

Vamos a ver primeramente la propiedad flex-direction que define la forma en que la caja contenedora apilará a sus hijos….

La propiedad flex-direction puede adoptar 4 valores : row, column, row-reverse o column-reverse

el código del “padre” o contenedor, quedaría así…

.flex-container {

display: flex;

flex-direction: column;

}

prueben ir cambiando **column** por los otros valores y vayan viendo como se acomodan los hijos.

**PROPIEDAD FLEX-WRAP**

Esta propiedad define si los elementos flexibles (hijos) deben ajustarse o no. O sea, si se mantienen en una fila y se van achicando de acuerdo al espacio disponible o bien se va formando otra fila.

Pongan varios div hijos así se ve mejor el efecto que produce…

.flex-container {

display: flex;

flex-wrap: wrap;

background-color: Blue;

}

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

<div>4</div>

<div>5</div>

<div>6</div>

<div>7</div>

<div>8</div>

<div>9</div>

<div>10</div>

<div>11</div>

<div>12</div>

</div>

Los posibles valores para esta propiedad son wrap, nowrap, y wrap-reverse.

**PROPIEDAD FLEX-FLOW**

Sirve para definir conjuntamente las dos propiedades anteriores o sea flex-direction y flex-wrap.

Por ejemplo…..  
.flex-container {

display: flex;

flex-flow: row wrap;

}

**PROPIEDAD JUSTIFY-CONTENT**

Es usada para alinear los elementos hijos dentro del padre. Prueben los diferentes valores, cambiando center por flex-start , flex-end, space-around y space-between y miren los resultados.

.flex-container {

display: flex;

justify-content: center;

}

**PROPIEDAD ALIGN-ITEMS**  
Se utiliza para alinear los elementos hijos flexibles.

.flex-container {

display: flex;

height: 200px;

align-items: center;

}

el código anterior ubica los elementos en el centro del padre. En lugar de center, los valores pueden ser flex-start, flex-end, strech, baseline. Prueben que pasa con cada uno de ellos.

**PROPIEDAD ALIGN-CONTENT**

Se usa para alinear las líneas flexibles. Les dejo el código html y el css de un ejemplo, luego vayan variando, las posibilidades son space-between, space-around, strech, center, flex-start, flex-end.

.flex-container {

display: flex;

height: 600px;

flex-wrap: wrap;

align-content: space-between;

background-color: DodgerBlue;

}

.flex-container > div {

background-color: #f1f1f1;

width: 100px;

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

}

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

<div>4</div>

<div>5</div>

<div>6</div>

<div>7</div>

<div>8</div>

<div>9</div>

<div>10</div>

<div>11</div>

<div>12</div>

</div>

**CENTRADO PERFECTO, COMO RESOLVER UN PROBLEMA COMUN**

Establezca las propiedades justify-contenty align-items en el *centro,* y el elemento flexible estará perfectamente centrado

.flex-container {

display: flex;

height: 300px;

justify-content: center;

align-items: center;

}

**ELEMENTOS SECUNDARIOS O CHILDS ITEMS**

Los elementos secundarios directos de un contenedor (padre) flexible se convierten automáticamente en elementos (hijos) flexibles.

Estos elementos pueden ser manipulados con estas propiedades…

order

flex-grow

flex-shrink

flex-basis

flex

align-self

**PROPIEDAD ORDER (ORDEN)**

La propiedad ORDER puede cambiar el orden de los elementos...así:

<div class="flex-container">

<div style="order: 3">1</div>

<div style="order: 2">2</div>

<div style="order: 4">3</div>

<div style="order: 1">4</div>

</div>

**PROPIEDAD FLEX-GROW (CRECIMIENTO FLEXIBLE)**

El siguiente ejemplo hace que el tercer elemento flexible crezca ocho veces más rápido que los otros elementos flexibles:

<div class="flex-container">

<div style="flex-grow: 1">1</div>

<div style="flex-grow: 1">2</div>

<div style="flex-grow: 8">3</div>

</div>

**PROPIEDAD FLEX-SHRINK**

La propiedad Flex-shrink especifica cuánto se encogerá un elemento flexible en relación con el resto de los elementos flexibles.

El valor debe ser un número y el valor predeterminado es 1.

El siguiente ejemplo no permite que el tercer elemento flexible se encoja tanto como los otros elementos flexibles

<div class="flex-container">

<div>1</div>

<div>2</div>

<div style="flex-shrink: 0">3</div>

<div>4</div>

<div>5</div>

<div>6</div>

<div>7</div>

<div>8</div>

<div>9</div>

<div>10</div>

</div>

**PROPIEDAD FLEX-BASIS**

Esta propiedad especifica la longitud inicial de un elemento flexible.

En el siguiente ejemplo se establece la longitud inicial del tercer elemento flexible en 200 píxeles

<div class="flex-container">

<div>1</div>

<div>2</div>

<div style="flex-basis: 200px">3</div>

<div>4</div>

</div>

**PROPIEDAD FLEX**

La propiedad FLEX es una propiedad abreviada para las propiedades flex-grow, flex-shrinky y flex-basis.

Un ejemplo de como se escribe esto es flex: 0 0 200px o sea 0(cero) para grow (no puede crecer), 0 (cero) para shrink (no se puede encoger) y 200 px de tamaño inicial.

En el siguiente ejemplo el tercer elemento flexible no se pueda crecer (0), no se pueda encoger (0) y tenga una longitud inicial de 200 píxeles:

<div class="flex-container">

<div>1</div>

<div>2</div>

<div style="flex: 0 0 200px">3</div>

<div>4</div>

</div>

**PROPIEDAD ALIGN-SELF**

La propiedad **align-self** especifica la alineación del elemento seleccionado dentro del contenedor flexible.

La propiedad **align-self** anula la alineación predeterminada establecida por la align-items del contenedor .

Veamos DOS ejemplos, donde usamos un contenedor de 200 píxeles de alto para demostrar mejor la propiedad align-self.

**Ejemplo 1:**  
Alinea el tercer elemento flexible en el medio del contenedor

<style>

.flex-container {

display: flex;

height: 200px;

background-color: #f1f1f1;

}

.flex-container > div {

background-color: DodgerBlue;

color: white;

width: 100px;

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

}

</style>

<div class="flex-container">

<div>1</div>

<div>2</div>

<div style="align-self: center">3</div>

<div>4</div>

</div>

**Ejemplo 2:**Alinea el segundo elemento flexible en la parte superior del contenedor y el tercer elemento flexible en la parte inferior del contenedor

<style>

.flex-container {

display: flex;

height: 200px;

background-color: #f1f1f1;

}

.flex-container > div {

background-color: DodgerBlue;

color: white;

width: 100px;

margin: 10px;

text-align: center;

line-height: 75px;

font-size: 30px;

}

</style>

<div class="flex-container">

<div>1</div>

<div style="align-self: flex-start">2</div>

<div style="align-self: flex-end">3</div>

<div>4</div>

</div>