

# Flexbox

## Contenido

Flexbox.....	2
Definir flexbox:.....	2
Propiedades para contenedores: .....	2
Propiedades para items: .....	5
Compatibilidad con navegadores:.....	5
Web para probar propiedades de flexbox:.....	6
Referencias:.....	6

## Flexbox

Flexbox es un módulo de css, para crear diseños flexibles.

Establece una serie de propiedades que nos permiten posicionar los elementos HTML, tanto horizontal como verticalmente con mayor facilidad que con los flotados.

Requiere que establezcamos un elemento contenedor y dentro de ese elemento contenedor, una serie de ítems que se adaptan, ajustando sus tamaños y disposición en función del espacio disponible de la caja contenedora.

Podemos incluso, desde el css, alterar el orden de los elementos y modificar la forma en que los elementos se ubican en pantalla (sin modificar el html).

### Definir flexbox:

Para poder utilizar flexbox, necesitamos crear un elemento contenedor (padre) y una serie de elementos hijos. Como el siguiente ejemplo:

```
<section class="contenedor">
    <article class="item1">1</article>
        <article class="item2">2</article>
        <article class="item3">3</article>
</section>
```

Al elemento contenedor le agregamos la propiedad display, con el valor flex, que establece que se trata de una caja flexible.

```
.contenedor{
    display: flex;
}
```

### Propiedades para contenedores:

#### 1.- flex-direction

Esta propiedad establece de qué forma los elementos hijos del contenedor, se van a posicionar.

```
.contenedor{  
    flex-direction: row;  
}
```

Tiene 4 valores posibles:

a) **flex-direction: row;**

Ordena los elementos horizontalmente de izquierda a derecha en una fila.

b) **flex-direction: row-reverse;**

Ordena los elementos horizontalmente de derecha a izquierda en una fila e invierte el orden de los mismos.

c) **flex-direction:column;**

Ordena los elementos verticalmente de arriba abajo en una columna.

d) **flex-direction:column-reverse;**

Ordena los elementos verticalmente de abajo arriba en una columna.

## 2.- **flex-wrap**

Redistribuye los elementos para que entren una o más líneas.

a) **flex-wrap: no-wrap;**

Redistribuye los elementos para que entren en una sola línea, achicándolos si es necesario.

b) **flex-wrap: wrap;**

Redistribuye los elementos en varias líneas de izquierda a derecha y de arriba a abajo.

c) **flex-wrap: wrap-reverse;**

Redistribuye los elementos en varias líneas de derecha a izquierda y de abajo a arriba.

## 3.- **justify-content**

Distribuye la posición de los elementos y la disposición que tendrán en el eje **horizontal**.

a) **Justify-content: flex-start;**

Este es el valor por defecto, alinea los elementos a la izquierda.

**b) Justify-content: flex-end;**

Alinea los elementos a la derecha.

**c) Justify-content: center;**

Centra horizontalmente los elementos.

**d) Justify-content: space-around;**

Distribuye los elementos, dejando el mismo espacio entre ellos.

**e) Justify-content: space-between;**

Distribuye los elementos ubicando el primero pegado a la izquierda del contenedor (sin espacio hacia la izquierda) y el último sin espacio a la derecha.

#### 4.- align-items

Distribuye la posición de los elementos y la disposición que tendrán en el eje **vertical**.

**a) align-items: stretch;**

Los elementos hijos, ocupan todo el ancho (o alto) del elemento contenedor.

**b) align-items: flex-start;**

Los elementos se ubican desde el inicio del borde superior del contenedor.

**c) align-items: flex-end;**

Los elementos se ubican desde el inicio del borde inferior del contenedor.

**d) align-items: center;**

Los elementos se ubican centrados verticalmente.

**e) align-items: baseline;**

Alinea los elementos tomando como línea base la altura del texto.

#### 5.- align-content

Permite distribuir verticalmente los elementos cuando hay varias filas.

## Propiedades para items:

### 1.-order

Con esta propiedad, podemos cambiar el orden de cualquier elemento hijo. Por ejemplo:

```
.item1{  
    order:3;  
}
```

### 2.-flex-grow

Es el crecimiento que va a tener un elemento en relación a los demás elementos hermanos. El valor por defecto es 1, si algún elemento tiene un valor de flex-grow mayor, el resto de los elementos, se redistribuyen en ancho.

Por ejemplo:

```
.item2{  
    flex-grow:2;  
}
```

### 3.-flex-shrink

Esto define la posibilidad de que un elemento flexible se contraiga si es necesario.

### 4.-flex-basis

Sirve para establecer un valor inicial de ancho o de alto de los elementos, previo a que ese ancho o alto sea modificado por las otras propiedades.

```
.item1{  
    flex-basis:30em;  
}
```

### 5.-align-self

Esta propiedad permite alinear un elemento en forma individual.

Los valores posibles son:

auto | flex-start | flex-end | center | baseline | stretch

## Compatibilidad con navegadores:

Como flexbox aún no es compatible con todos los navegadores, debemos agregar los prefijos -ms-, -webkit- a las propiedades de flexbox.

Podemos ver qué navegadores son compatibles en el sitio Can I Use:

<http://caniuse.com/#search=flexbox>

## Web para probar propiedades de flexbox:

<https://demos.scotch.io/visual-guide-to-css3-flexbox-flexbox-playground/demos/>

## Referencias:

- W3 Schools Css3 Flexbox

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

- A visual Guide to flexbox properties

Autor: Dimitar Stojanov

<https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties>

- Css-Tricks- A Complete Guide to Flexbox

Autor: Chris Coyier

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

- Understanding Flexbox: Everything you need to know

Autor: Ohans Emmanuel

<https://medium.freecodecamp.com/understanding-flexbox-everything-you-need-to-know-b4013d4dc9af#.mmfc1uaov>