



# Flexbox

El módulo flexbox de **CSS** es una nueva forma de organizar, distribuir y alinear el layout de elementos **css**, permitiéndoles incluso ocupar espacio de contenedores con tamaño desconocido o variable (de ahí lo "flexible").

La idea básica de flexbox es darle al contenedor la capacidad de distribuir su espacio sobrante entre cada uno de los items que contiene, permitiéndoles crecer o achicarse según corresponda el caso.

## display: flex

Es la propiedad que le vamos a agregar al contenedor para volverlo un contenedor flexible.

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

## flex-direction

Establece la dirección principal en la que se van a colocar los elementos dentro del contenedor.

### row

- Posiciona los elementos a lo largo del eje **X**, en fila, horizontalmente:

```
.container {  
  display: flex;  
  flex-direction: row;  
}
```

### row-reverse

- Posiciona todos los elementos a lo largo del eje **X** pero de atrás hacia adelante



• Posiciona todos los elementos a lo largo del eje **X**, pero de atrás hacia adelante.

```
.container {  
  display: flex;  
  flex-direction: row-reverse;  
}
```

## column

- Posiciona todos los elementos a lo largo del eje **Y**, verticalmente, en columna.

```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

## column-reverse

- Posiciona todos los elementos a lo largo del eje **Y**, pero de atrás hacia adelante

```
.container {  
  display: flex;  
  flex-direction: column-reverse;  
}
```

## Ejercicio 1

## order

Es una propiedad que se le da a los elementos **dentro** de un contenedor flexible para cambiar el orden en que se ubican.

- Se indica simplemente con un número.
- Por defecto todos tienen orden 0.
- Si dos elementos tienen el mismo orden, se distribuyen según ubicación en el css.



```
.item-2 {
```

```
    order: 3;
  }
.item-3 {
  order: 2;
}
```

## Ejercicio 2

### flex-basis

- Especifica el tamaño inicial del item, antes de que el espacio restante sea distribuido.
- Cuando se omite se pone en 0.
- Si se utiliza la palabra **auto**, tomará como referencia el contenido del elemento, ya sea el alto o el ancho, dependiendo de la orientación del contenedor.

```
.item-1 {
  flex-basis: 400px;
}
.item-2 {
  flex-basis: 100px;
}
```

## Ejercicio 3

### flex-grow

- Indica cuánto más espacio a tomar del espacio libre restante va a ocupar un elemento con respecto a otro.
- Por defecto es 0, es decir, no va a tomar ningún espacio libre.
- Si todos los elementos tienen un valor de 1, el espacio a ocupar se distribuye equitativamente entre ellos. Si uno tiene valor de 2, tomará el doble de éstos.

```
.item-2 {
  flex-grow: 1;
}
.item-3 {
  flex-grow: 3;
```



}

## Ejercicio 4

### flex-shrink

- Indica hasta cuánto puede reducirse un elemento con respecto a otros.
- Por defecto su valor es 1.
- Si todos los elementos tienen un valor de 1, a medida que el contenedor se achica, se reducirán equitativamente.
- Si uno de los elementos tiene un valor de 2, éste se reducirá la mitad de éstos.

```
.item-2 {  
    flex-shrink: 2;  
}  
.item-3 {  
    flex-shrink: 3;  
}
```

## Ejercicio 5

### flex-wrap

- Le indica al contenedor si puede distribuir los elementos en más de una línea a medida que se éste se va a chicando.
- El valor por defecto es **no-wrap**.

```
.container {  
    flex-wrap: wrap;  
}  
  
.container {  
    flex-wrap: wrap-reverse;  
}
```



## Ejercicio 6

# justify-content

Le indica al contenedor cómo distribuir los elementos cuando hay espacio libre.

## flex-start

- Posiciona los elementos al comienzo del contenedor (por defecto)

```
.container {  
  justify-content: flex-start;  
}
```

## flex-end

- Posiciona los elementos al final del contenedor.

```
.container {  
  justify-content: flex-end;  
}
```

## center

- Posiciona los elementos en el medio del contenedor.

```
.container {  
  justify-content: center;  
}
```

## space-around

- Les da a todos los elementos la misma distancia de separación a ambos lados.
- Los items de los extremos están más cerca del borde que de los items, porque entre el borde y ellos hay una distancia de separación, y entre ellos y los items, una por cada uno

```
.container {  
  justify-content: space-around;  
}
```



```
justify-content: space-around;  
}
```

### space-between

- Posiciona un elemento al principio y otro al final del contenedor, y distribuye los restantes entre estos dos, con espacio equitativo entre ellos.

```
.container {  
  justify-content: space-between;  
}
```

### space-evenly

- Posiciona los elementos de forma tal que el espacio entre ellos y entre los bordes es el mismo.

```
.container {  
  justify-content: space-evenly;  
}
```

## Ejercicio 7

## align-items

Define como los items se alinean en el eje perpendicular al eje del contenedor.

### flex-start

- Posiciona los elementos al principio de la línea perpendicular al eje del contenedor.

```
.container {  
  align-items: flex-start;  
}
```

### flex-end

Posiciona los elementos al final de la línea perpendicular al eje del contenedor.



```
.container {  
    align-items: flex-end;  
}
```

## center

- Posiciona los elementos en el medio de la línea perpendicular al eje del contenedor.

```
.container {  
    align-items: center;  
}
```

## stretch

- Posiciona los elementos para que llenen todo el espacio posible a través de la línea perpendicular al eje del contenedor (por defecto)

```
.container {  
    align-items: stretch;  
}
```

## baseline

- Los elementos se distribuyen de forma tal que la base de línea del contenido de ellos queda alineado.

```
.container {  
    align-items: baseline;  
}
```

## align-content

Funciona como **justify-content**, pero distribuyendo el espacio que hay a lo largo del eje perpendicular al eje contenedor. Debe usarse en combinación con **flex-wrap**, ya que no funciona cuando el contenedor tiene una sola línea de elementos.



## flex-start

```
.container {  
  align-content: flex-start;  
}
```

## flex-end

```
.container {  
  align-content: flex-end;  
}
```

## center

```
.container {  
  align-content: center;  
}
```

## space-around

```
.container {  
  align-content: space-around;  
}
```

## space-between

```
.container {  
  align-content: space-between;  
}
```


## stretch

```
.container {  
  align-content: stretch;  
}
```





## Ejercicio 8

 [Edit this page](#)

Previous

« **Media Queries**

Next

**Sass »**

display: flex

flex-direction

order

flex-basis

flex-grow

flex-shrink

flex-wrap

justify-content

align-items

align-content

### Docs

[Frontend](#)

### Community

[Discord](#)

### Social

[Linkedin](#)

[Instagram](#)



Facebook

Copyright © 2021 Ada Itw.

