

# Estructura básica de Flexbox

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree



**Flexbox** propone una estructura basada en el uso de un **contenedor padre** (*flex-container*) y sus **elementos hijos** (*flex-items*).



# Trabajar con Flexbox

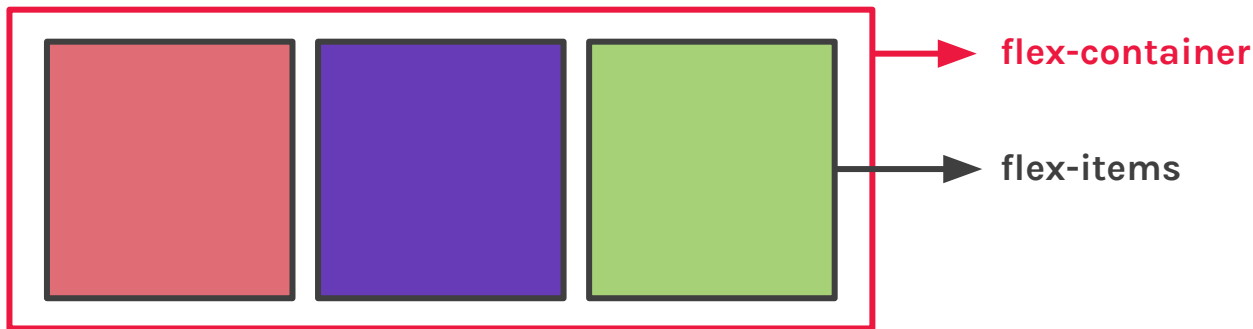
Para empezar a trabajar con Flexbox tenemos que definir un **flex-container**. Para eso usamos la propiedad `display` con el valor `flex`. De esta forma, estamos habilitando un **contexto flex**, para trabajar con los **hijos directos** del **elemento**. La propiedad `display` también puede recibir el valor `inline-flex`.

CSS

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

# Estructura básica

Cuando hablamos de un **flex-container**, hablamos de un elemento HTML que **contiene** a uno o más elementos. A estos elementos anidados los llamamos **flex-items**. En el **flex-container** es en donde configuramos la mayoría de las propiedades flex.



# flex-wrap

Por defecto, los elementos hijos de un contenedor flex van a tratar de **entrar** todos en una **misma línea**.



Para aclararle al contenedor que **debe respetar** el **ancho** definido de sus **hijos** usamos la propiedad `flex-wrap` con el valor `wrap`.



# flex-wrap

Como dijimos, la propiedad `flex-wrap` con el valor `wrap` permitirá que los ítems tomen el ancho definido y que los que no entren en la línea, caigan a la siguiente. `flex-wrap` también puede recibir los valores `nowrap` y `wrap-reverse`.

css

```
.contenedor-padre {  
  display: flex;  
  flex-wrap: wrap;  
}
```

# flex-items

Un **flex-item**, a su vez, puede convertirse en un **flex-container**. Para eso, solo hace falta asignarle la regla `display: flex`, para que así sus **elementos hijos** pasen a ser **flex-items**.

css

```
.elemento-hijo {  
    display: flex;  
}
```

DigitalHouse>  
Coding School