

Con la propiedad flex hacemos que todos los elementos hijo se coloquen uno al lado del otro
Con esta propiedad siempre debemos tener un contenedor

```
<div class="flex-container">  
  <div class="flex-hijo">A</div>  
  <div class="flex-hijo">A</div>  
  <div class="flex-hijo">A</div>  
</div>
```

```
.flex-container {  
  display: flex;  
}  
  
.flex-hijo {  
  width: 150px;  
  height: 100px;  
  background-color: brown;  
  margin: 10px;  
}
```

```
flex-direction: row;
```

```
flex-direction: row-reverse;
```

```
flex-direction: column;
```

```
flex-direction: column-reverse;
```

Por defecto .- se colocan en línea

Se invierten las posiciones

Se colocan en columnas

Se invierten las posiciones

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

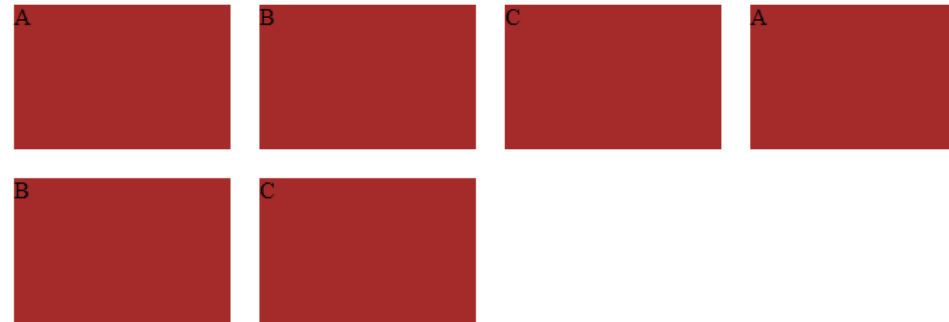
```
.flex-container {  
  display: flex;  
  flex-direction: column-reverse;  
}
```

La propiedad flex-wrap .- nos permite controlar cuantos elementos estarán en una fila
Esto dependerá del tamaño de los mismos



```
<div class="flex-container">  
  <div class="flex-hijo">A</div>  
  <div class="flex-hijo">B</div>  
  <div class="flex-hijo">C</div>  
  <div class="flex-hijo">A</div>  
  <div class="flex-hijo">B</div>  
  <div class="flex-hijo">C</div>  
</div>
```

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



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



La propiedad flex-start no permite controlar la posición de los hijos con respecto al padre en horizontal

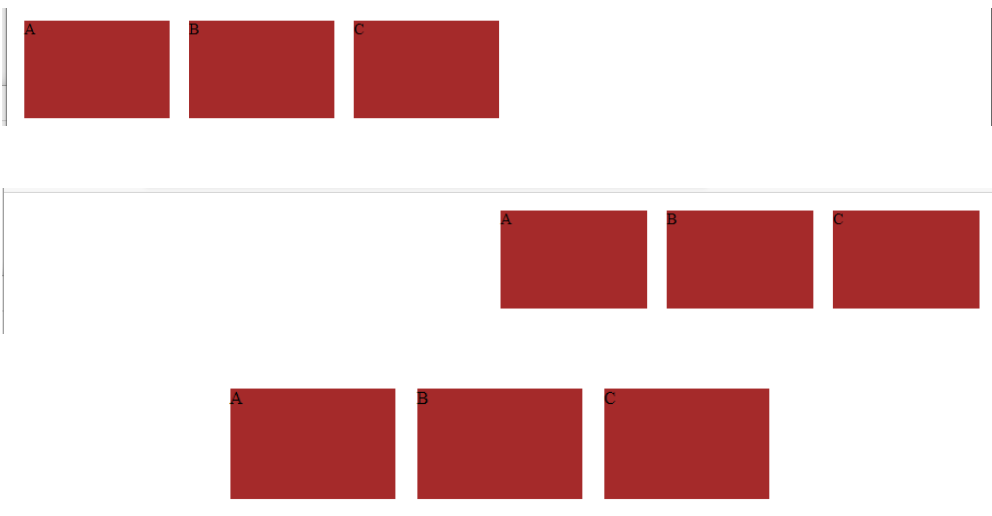
Justify-content: flex-end / flex-start

```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
  flex-direction: row;  
  justify-content: flex-end;  
}
```

Estas dos propiedades nos permiten distribuir
Los elementos dentro del padre

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

Podemos colocar los hijos al principio del padre , al final o al centro



```
justify-content: flex-start;
```

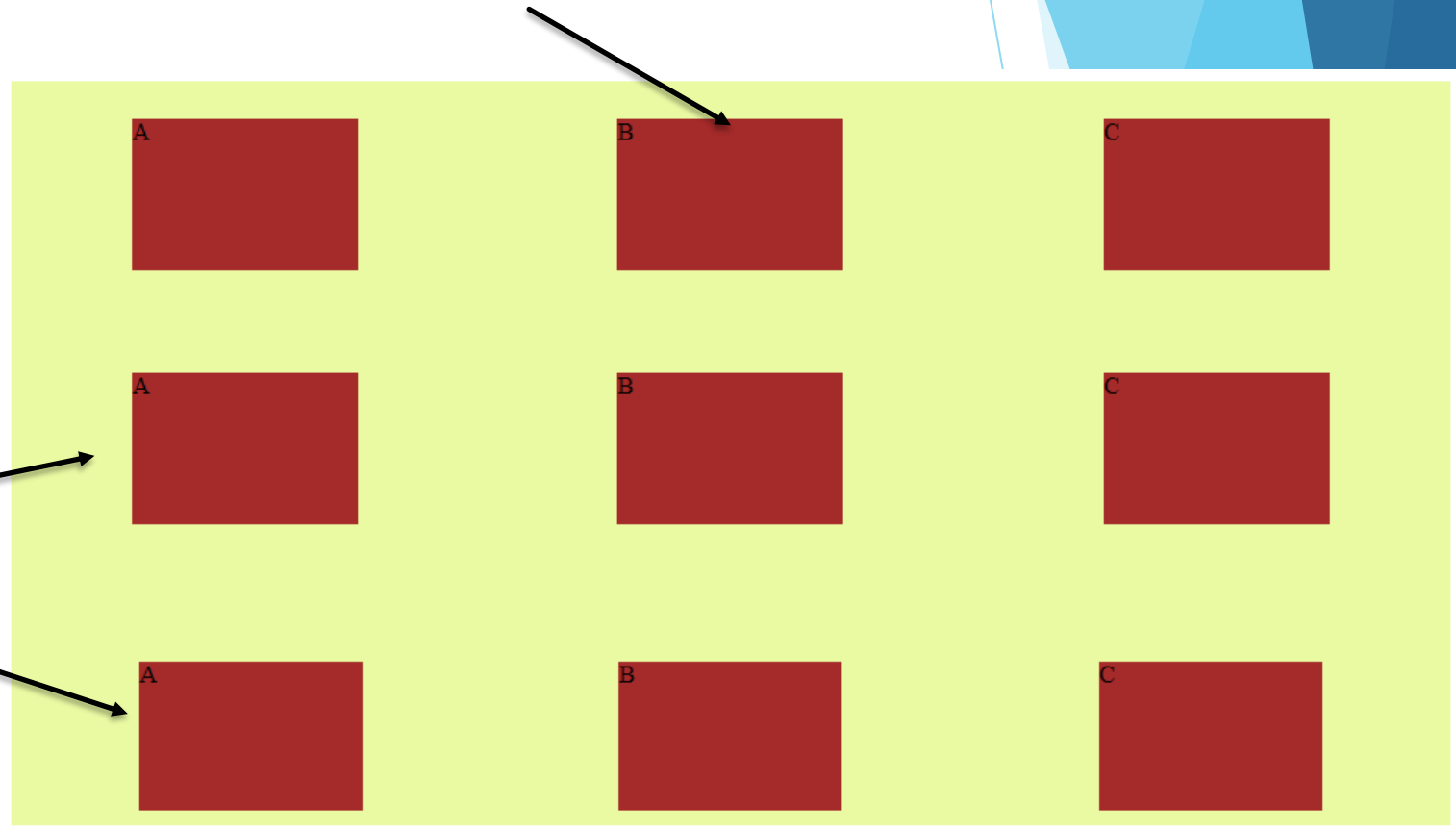
```
justify-content: flex-end;
```

```
justify-content: center;
```

La propiedad align-items nos permite controlar la posición de los hijos con respecto al padre en vertical

Por defecto esta en align-items: flex-start

```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
  height: 500px;  
  flex-direction: row;  
  justify-content: space-around;  
  align-items: center;  
  align-items: flex-end;  
  background-color: #eafaa3;  
}
```

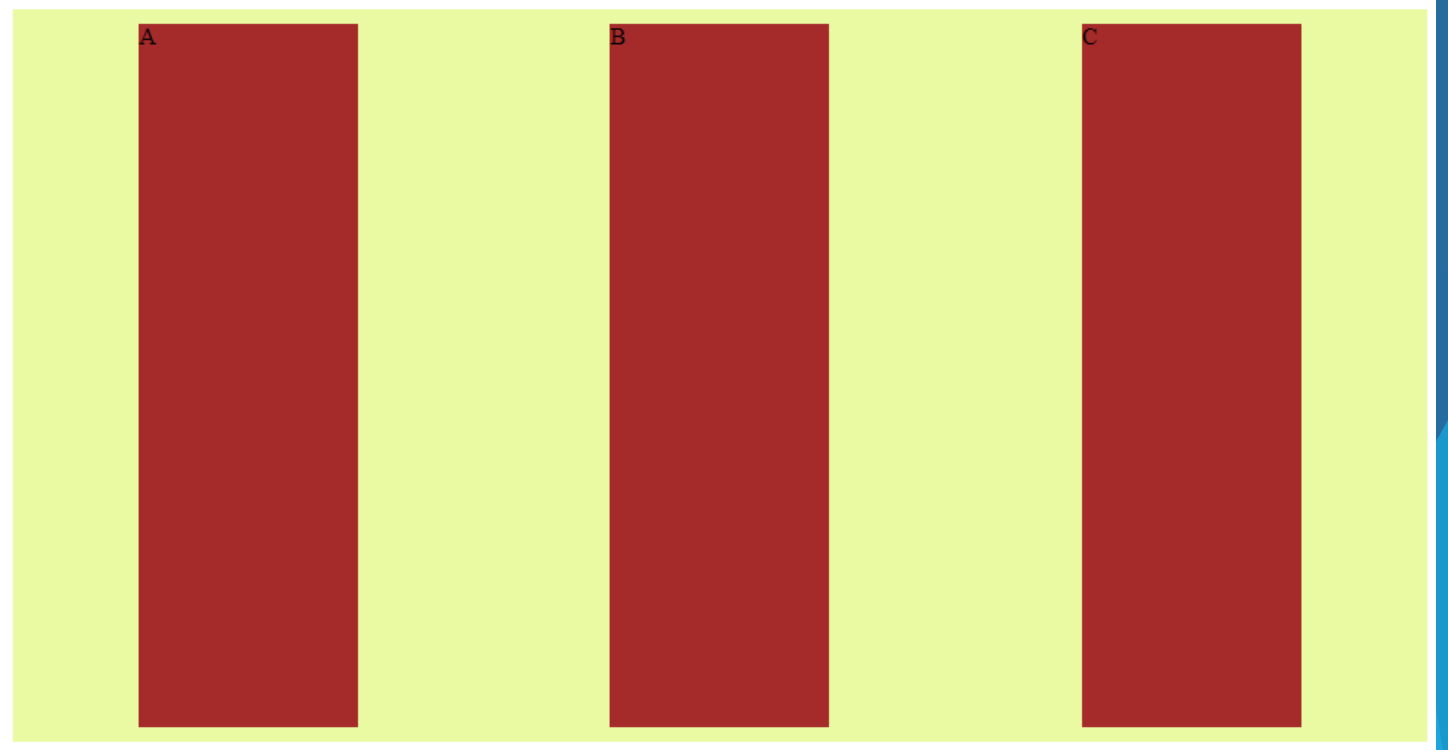


`align-items: stretch; .-`

Si le quitamos la altura a los hijos, estos se adaptaran al tamaño del padre

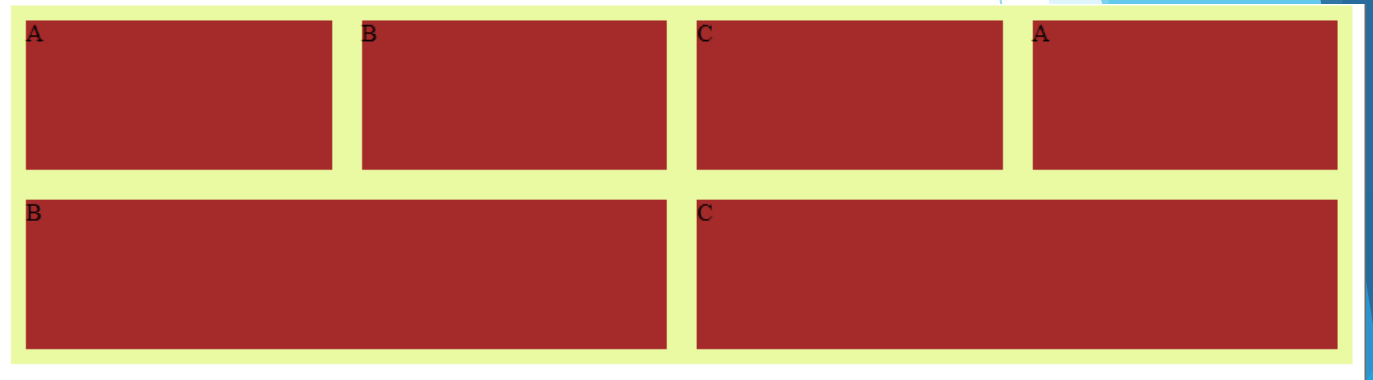
```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
  height: 500px;  
  flex-direction: row;  
  justify-content: space-around;  
  align-items: stretch;  
  background-color: #eafaa3;  
}
```

```
.flex-hijo {  
  width: 150px;  
  height: 100px;  
  background-color: brown;  
  margin: 10px;  
}
```



`flex-grow: 1;` // ocupan el 100% del contenedor y se reparten entre los hijos
`flex-shrink: 1;` / por defecto, la forma en que decrecen las cajas - 0 no decrece
`flex-basis: auto;` / ancho mínimo definido
Flex: 1 1 auto; conseguimos que se adapta automáticamente los anchos

```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
  flex-direction: row;  
  align-content: flex-start;  
  flex-wrap: wrap;  
  background-color: #eafaa3;  
}  
.flex-hijo {  
  width: 200px;  
  background-color: brown;  
  margin: 10px;  
  height: 100px;  
  flex-grow: 1; //  
}
```



[\(599\) Curso de HTML y CSS. 13.8. Flexbox: tamaños. flex-grow, flex-shrink y flex-basis - YouTube](#)
[Flexy Boxes – CSS flexbox playground and code generation tool \(the-echoplex.net\)](#)