

HTML + CSS



Episodio X

Flexbox

Posicionamiento
== Flex ==

Para qué podemos usar FlexBox?

Para qué usamos la flotación (float)?

- ⊙ Alineación
- ⊙ Dirección
- ⊙ Orden
- ⊙ Tamaño

Posicionamiento

== Flex ==

display: flex;

Consiste en un elemento padre con varios elementos hijos los cuales se van a acomodar de forma horizontal o vertical.

Tomando como referente nuestra caja madre.

Ej:

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

Flexbox

== justify-content ==

Justify-content.

Nos va a permitir acomodar los elementos según esté definido el **MAIN-AXIS** (**flex-direction**).

Si *flex-direction = row* => **MAIN-AXIS** es horizontal

Si *flex-direction = column* => **MAIN-AXIS** es vertical

flex-start: Coloca elementos al lado izquierdo del contenedor.

flex-end: Coloca elementos al lado derecho del contenedor.

center: Coloca elementos en el centro del contenedor.

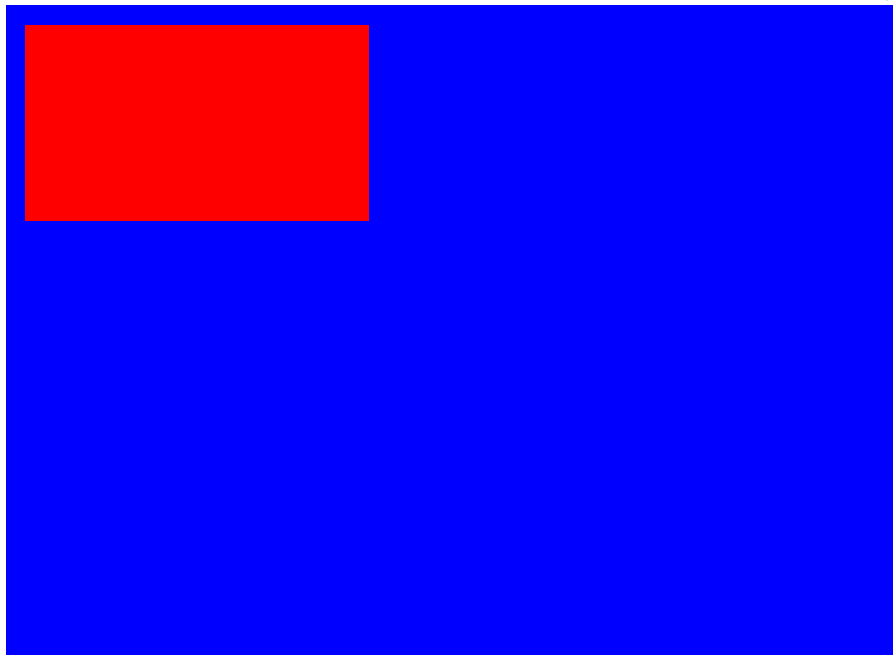
space-between: Coloca elementos con la misma distancia entre ellos.

space-around: Coloca elementos con la misma separación alrededor de ellos.

Flexbox

== justify-content ==

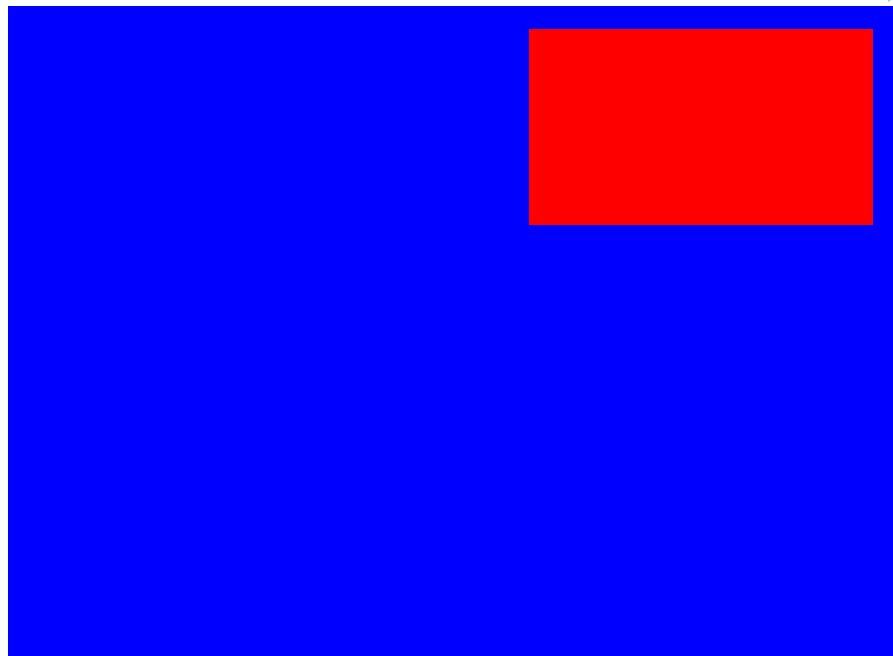
```
.caja-azul {  
  display: flex;  
  justify-content: flex-start;  
}
```



Flexbox

== justify-content ==

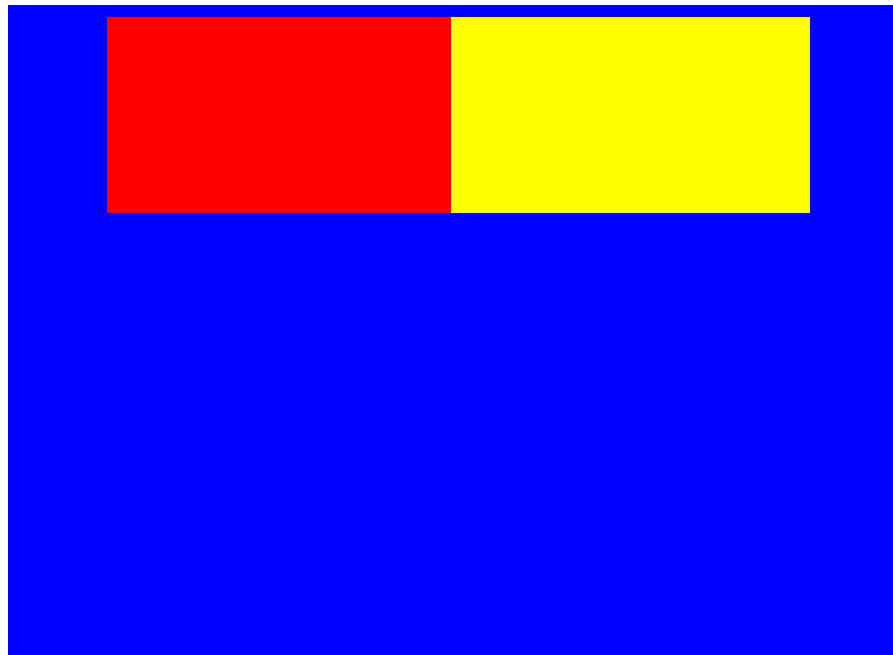
```
.caja-azul {  
  display: flex;  
  justify-content: flex-end;  
}
```



Flexbox

== justify-content ==

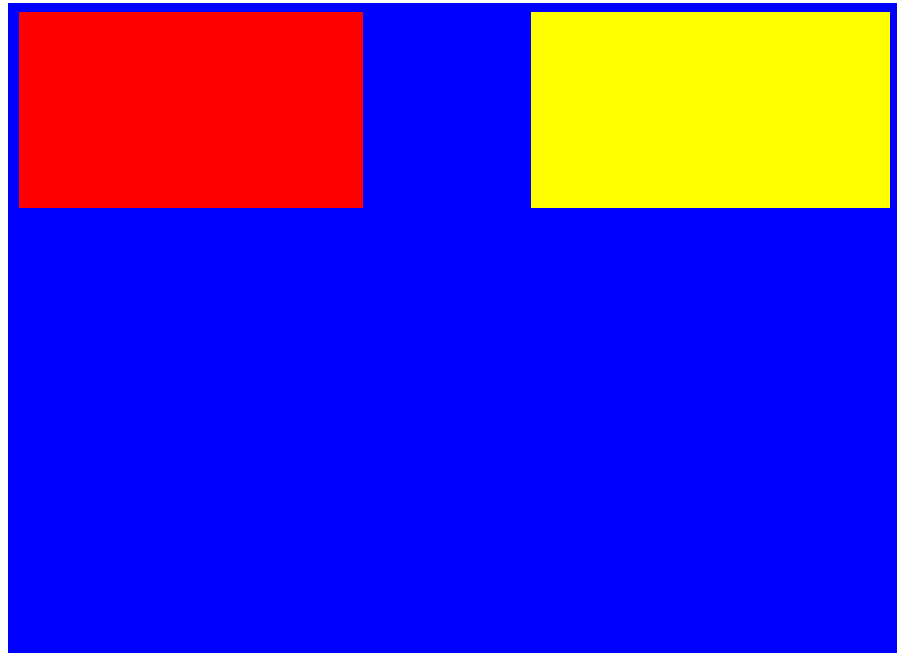
```
.caja-azul {  
  display: flex;  
  justify-content: center;  
}
```



Flexbox

== justify-content ==

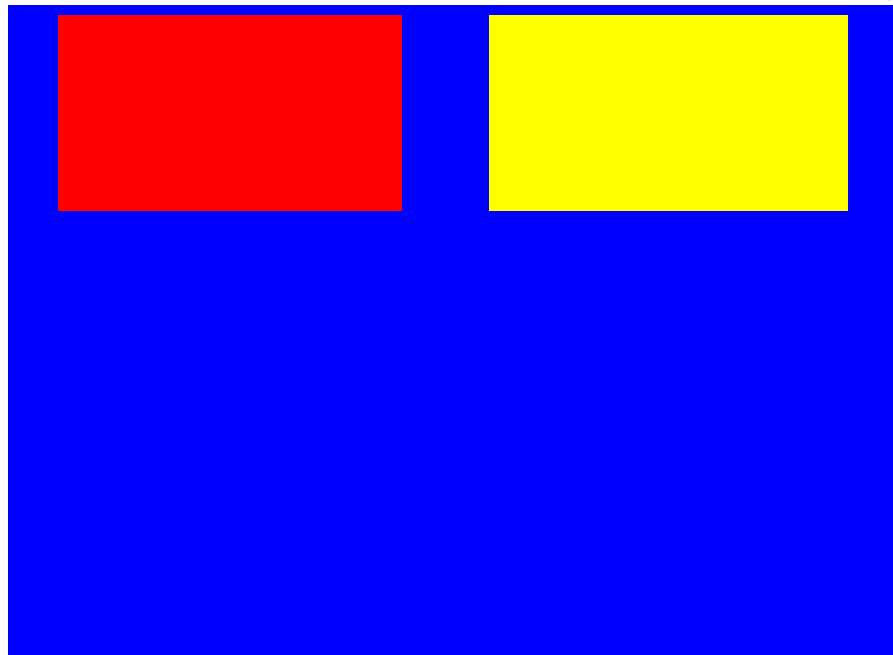
```
.caja-azul {  
  display: flex;  
  justify-content: space-between;  
}
```



Flexbox

== justify-content ==

```
.caja-azul {  
  display: flex;  
  justify-content: space-around;  
}
```





Practiquemos

Realizar los puntos de la guía de ejercicios.

Flexbox

== flex-direction ==

Flex-direction.

row: Elementos son colocados en la misma dirección del texto. (default)

row-reverse: Elementos son colocados en la dirección opuesta al texto.

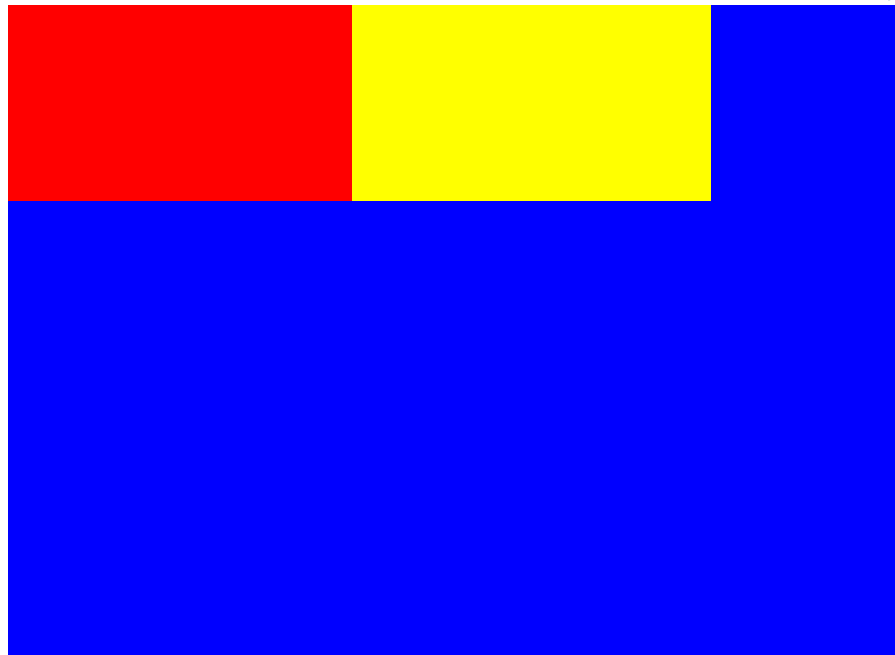
column: Elementos se colocan de arriba hacia abajo.

column-reverse: Elementos se colocan de abajo hacia arriba.

Flexbox

== flex-direction ==

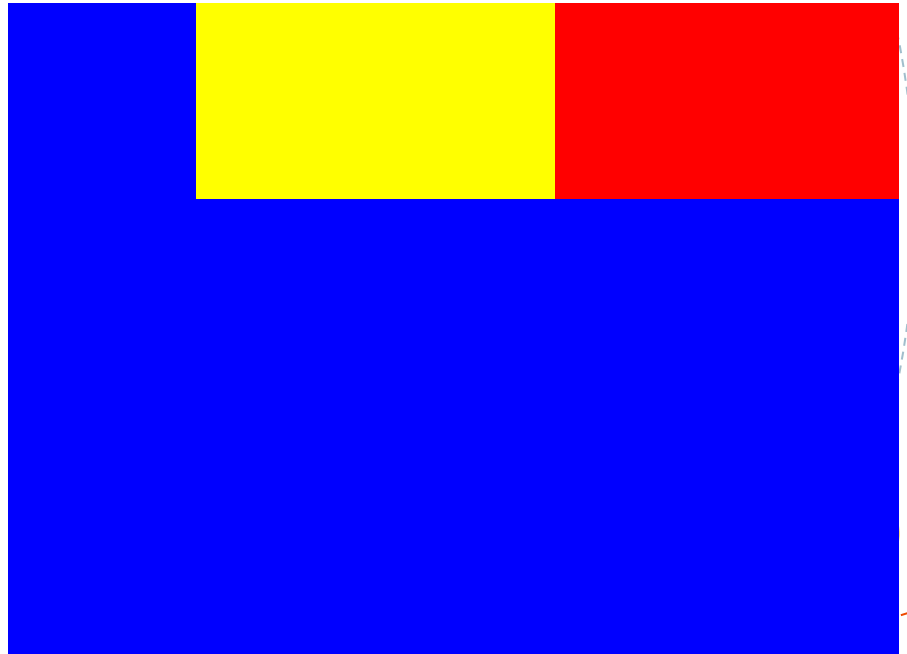
```
.caja-azul {  
  display: flex;  
  flex-direction: row;  
}
```



Flexbox

== flex-direction ==

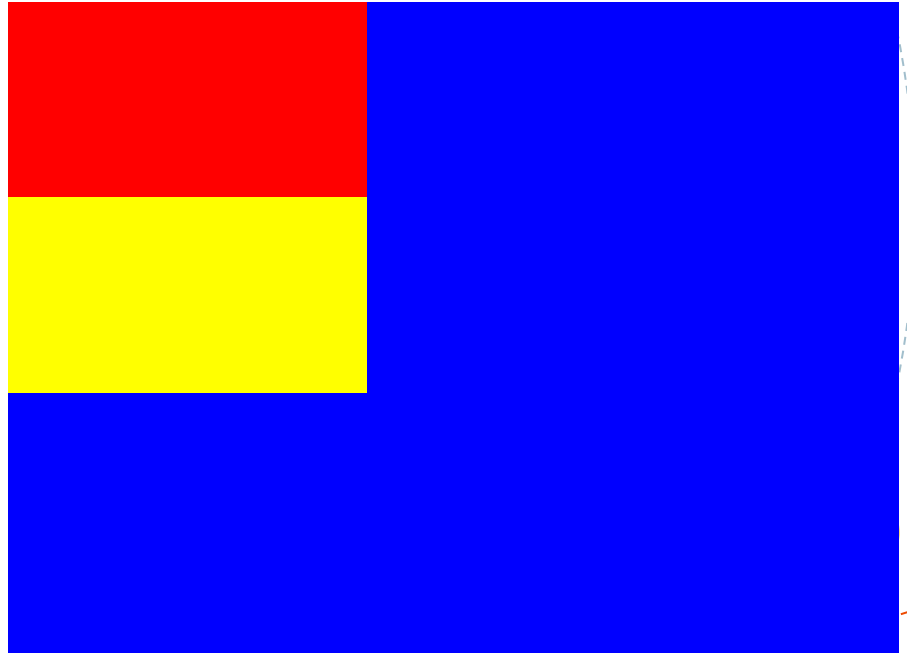
```
.caja-azul {  
  display: flex;  
  flex-direction: row-reverse;  
}
```



Flexbox

== flex-direction ==

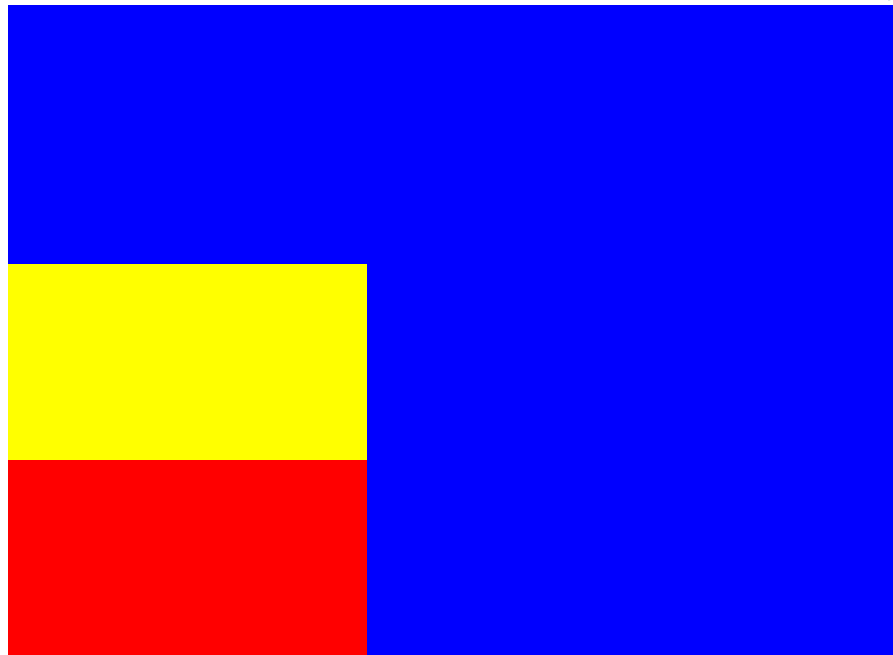
```
.caja-azul {  
  display: flex;  
  flex-direction: column;  
}
```



Flexbox

== flex-direction ==

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



Flexbox

== align-items ==

Align-items.

Nos va a permitir alinear los elementos de forma vertical u horizontal en un contenedor sobre el **CROSS-AXIS**.
Depende de cómo esté definida la propiedad **flex-direction**

Para el caso de **flex-direction: row** los "mueve" verticalmente

flex-start: Alinea elementos a la parte superior del contenedor.

flex-end: Alinea elementos a la parte inferior del contenedor.

center: Alinea elementos en el centro del contenedor.

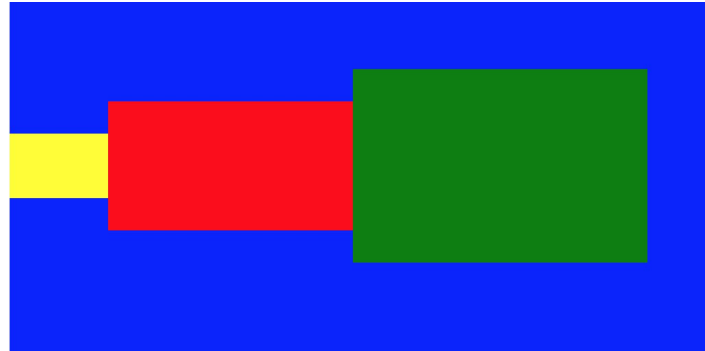
baseline: Muestra elementos en la línea base del contenedor

stretch: Elementos se estiran para ajustarse al contenedor. (salvo que tengan alto definido!!)

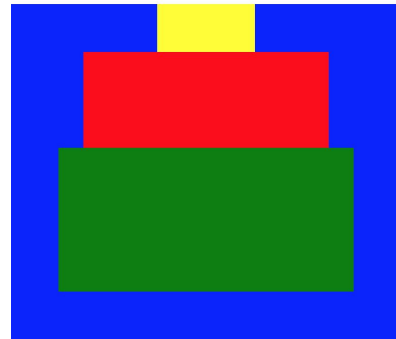
Flexbox

== align-items ==

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



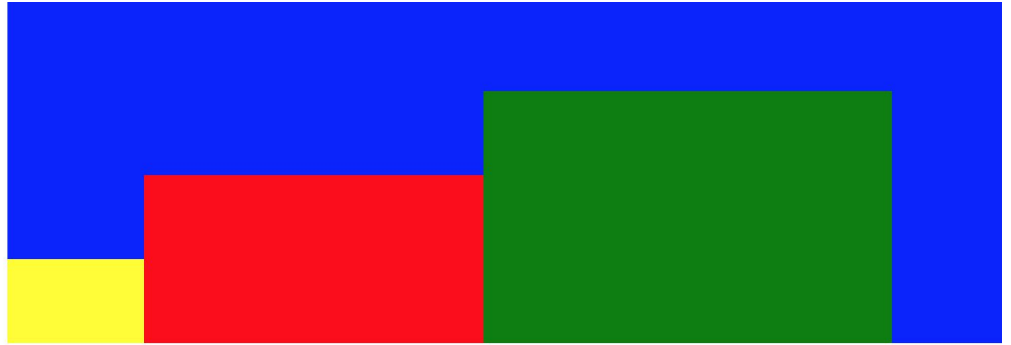
```
.container {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}
```



Flexbox

== align-items ==

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





Practiquemos

Realizar los puntos de la guía de ejercicios.



Posicionamiento == Order ==

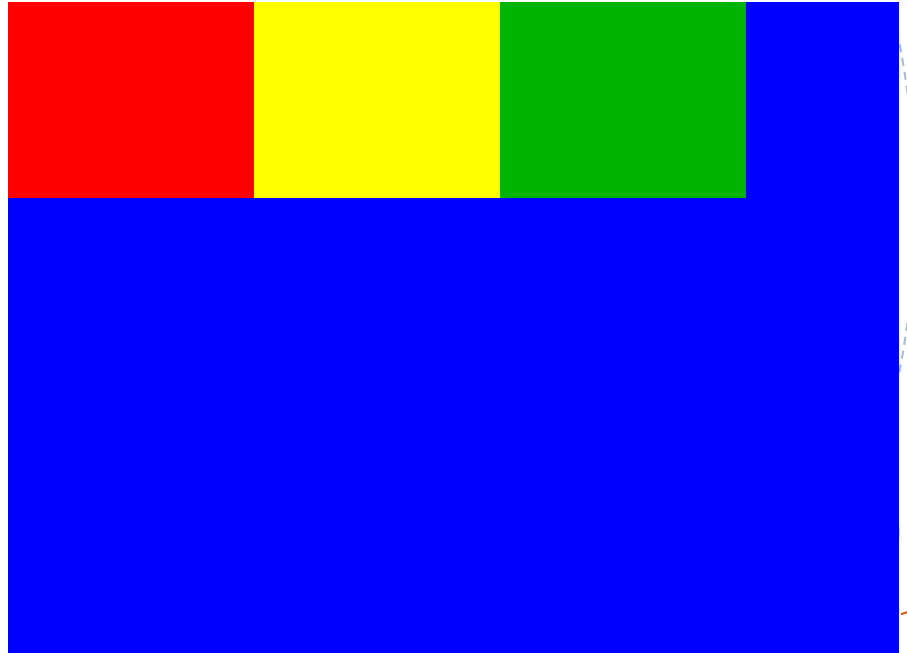
Order

Podemos aplicar una propiedad `order` a elementos individuales. Por defecto, los elementos tienen un valor 0, pero nosotros podemos usar esta propiedad para establecerlo a un número entero positivo o negativo.

Flexbox

== Order ==

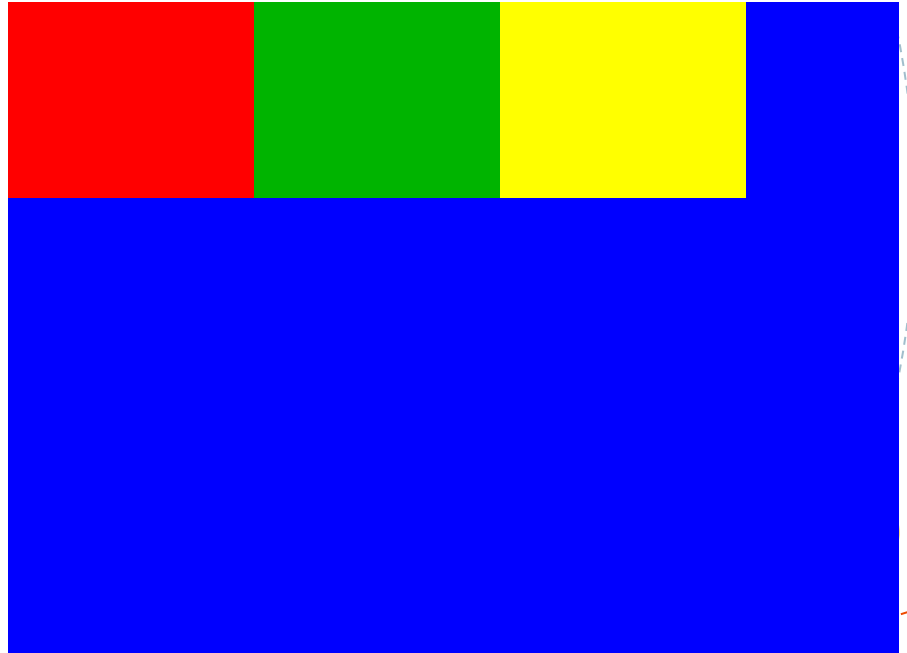
```
.caja-azul {  
  display: flex;  
}  
.caja amarilla{  
  order: 0;  
}
```



Flexbox

== Order ==

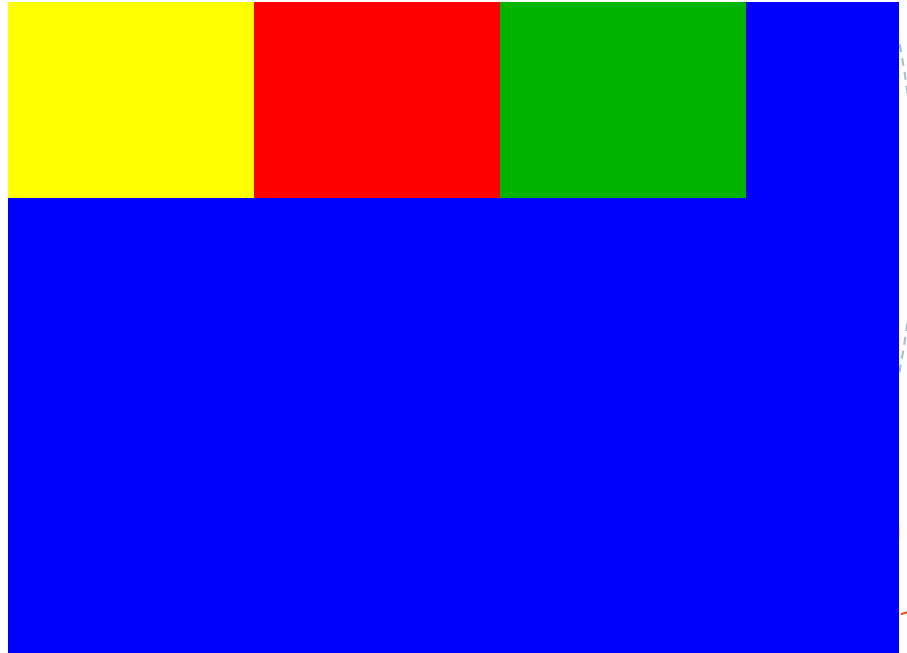
```
.caja-azul {  
  display: flex;  
}  
.caja amarilla{  
  order: 1;  
}
```



Flexbox

== Order ==

```
.caja-azul {  
  display: flex;  
}  
.caja amarilla{  
  order: -1;  
}
```





Practiquemos

Realizar los puntos de la guía de ejercicios.

Flexbox

== flex-wrap ==

Flex-wrap

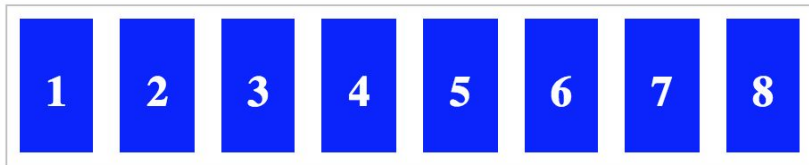
Propiedad que especifica si los elementos están forzados a estar en una línea o pueden estar en múltiples líneas. También tiene un control de dirección que invierte la dirección de las filas.

nowrap: Cada elemento se ajusta en una sola línea. (**default** | **no importan los anchos**)

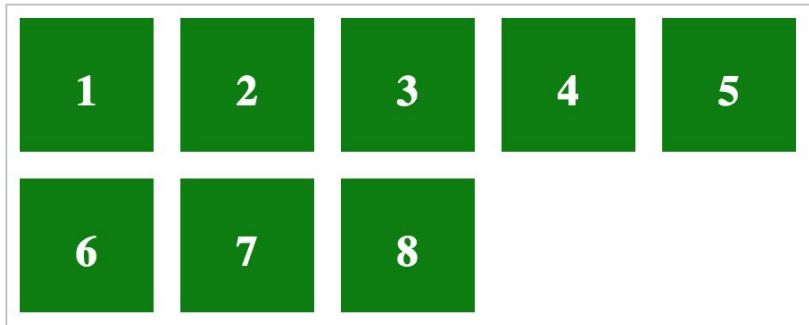
wrap: los elementos se envuelven alrededor de líneas adicionales. *Es decir permite múltiples líneas.* (**respetar los anchos**)

wrap-reverse: Los elementos se envuelven alrededor de líneas adicionales en reversa.

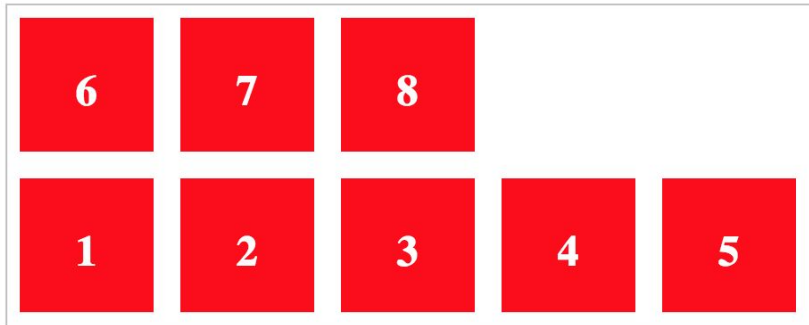
```
.container {  
  display: flex;  
  flex-wrap: nowrap;  
}
```



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



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





Practiquemos

Realizar los puntos de la guía de ejercicios.

Flexbox

== align-content ==

Align-content

Sirve para establecer como múltiples líneas están separadas una de otra.

Similar a justify-content pero VERTICAL.

Esta propiedad acepta los siguientes valores:

flex-start: Las líneas se posicionan en la parte superior del contenedor.

flex-end: Las líneas se posicionan en la parte inferior del contenedor.

center: Las líneas se posicionan en el centro (verticalmente hablando) del contenedor.

space-between: Las líneas se muestran con la misma distancia entre ellas.

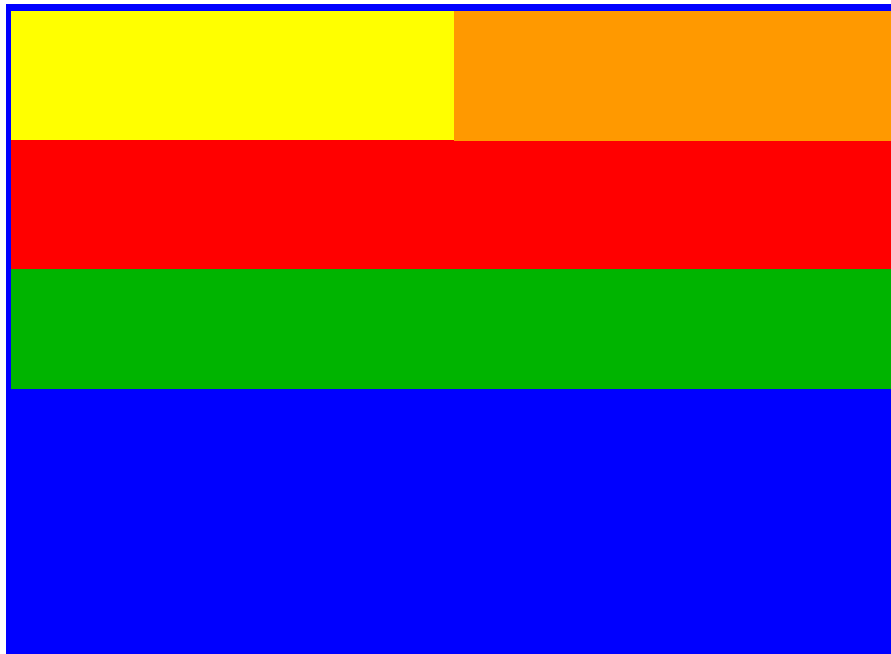
space-around: Las líneas se muestran con la misma separación alrededor de ellas.

stretch: Las líneas se estiran para ajustarse al contenedor.

Flexbox

== align-content ==

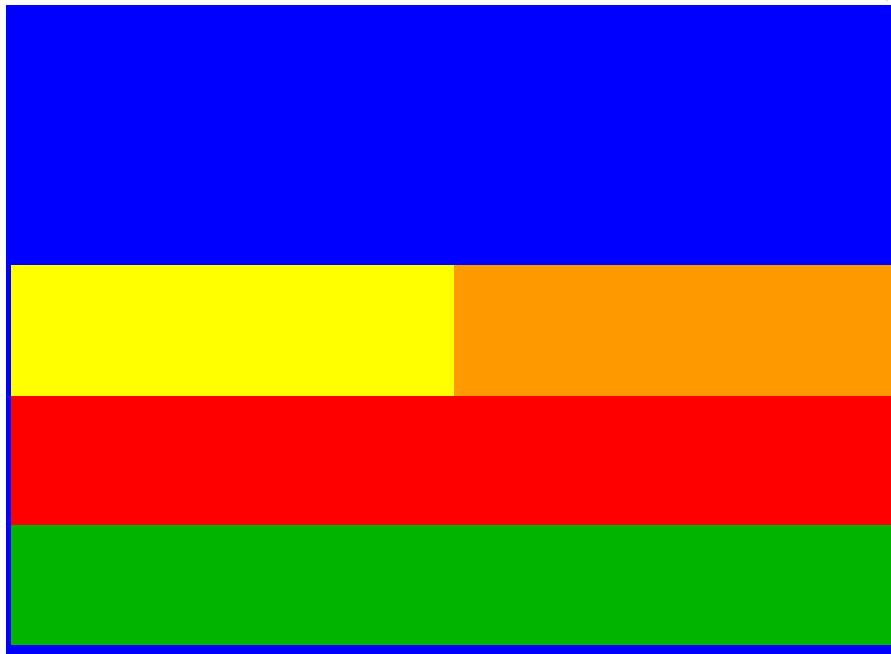
```
.caja-azul {  
  display: flex;  
  align-content: flex-start;  
}
```



Flexbox

== align-content ==

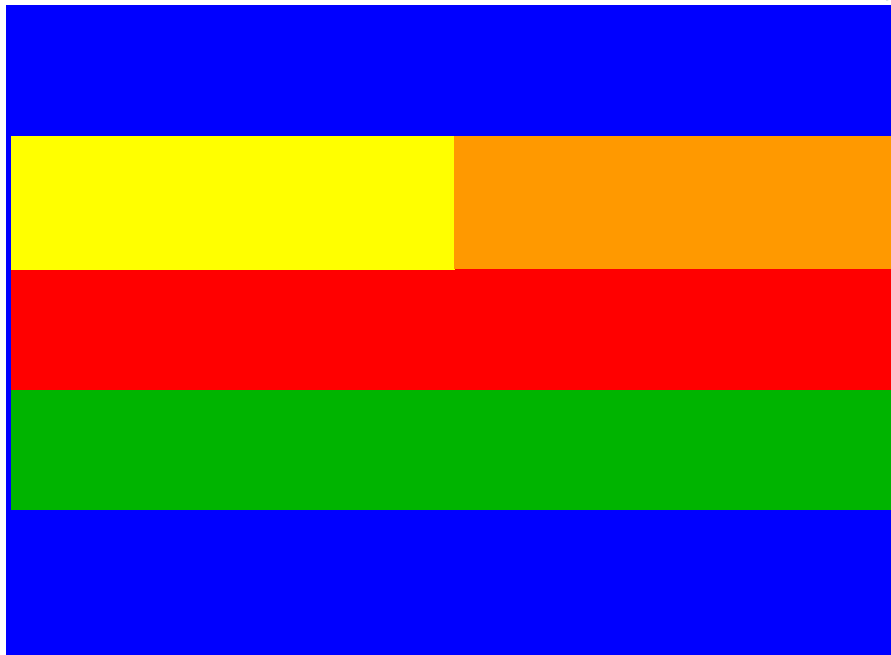
```
.caja-azul {  
  display: flex;  
  align-content: flex-end;  
}
```



Flexbox

== align-content ==

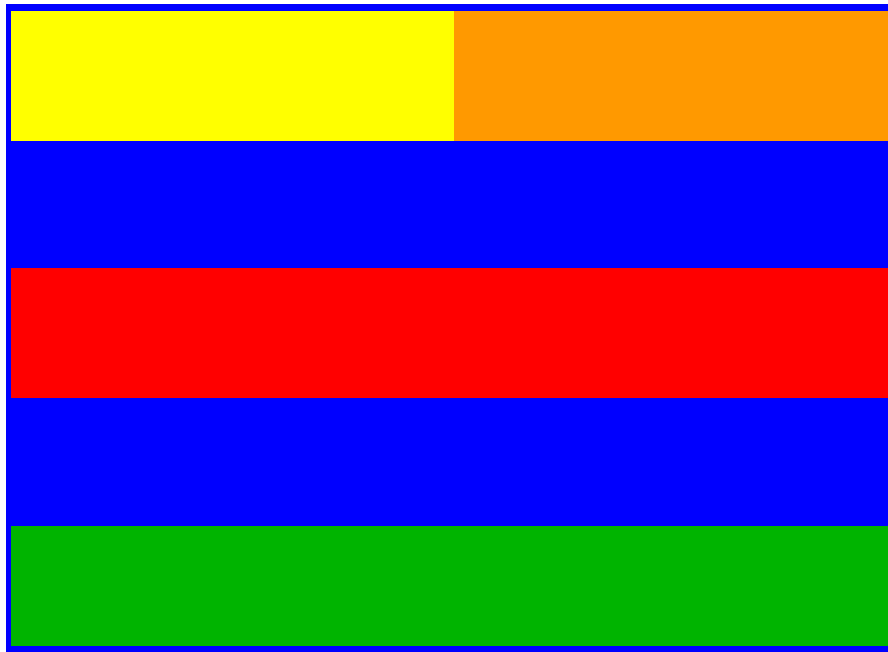
```
.caja-azul {  
  display: flex;  
  align-content: center;  
}
```



Flexbox

== align-content ==

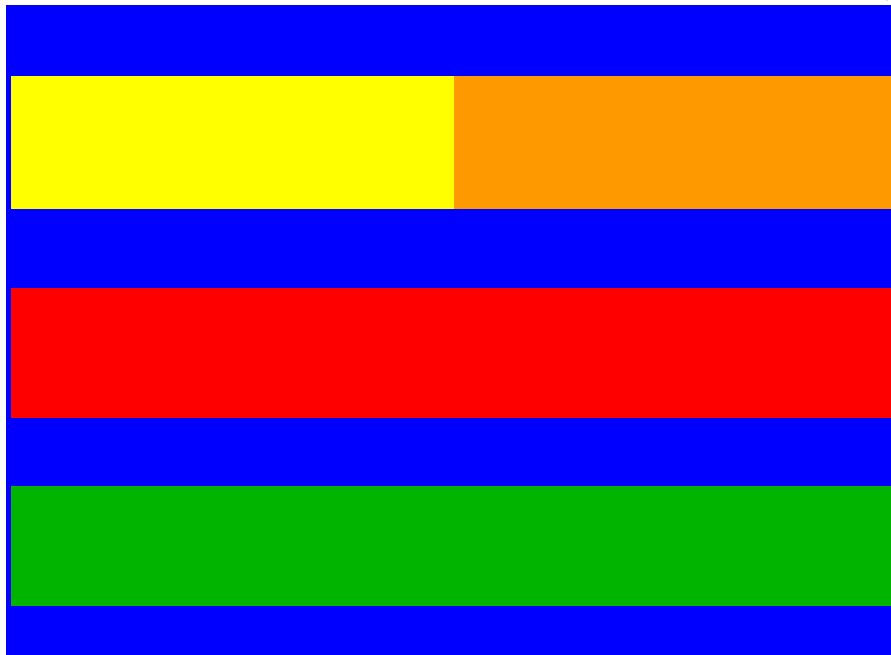
```
.caja-azul {  
  display: flex;  
  align-content: space-between;  
}
```



Flexbox

== align-content ==

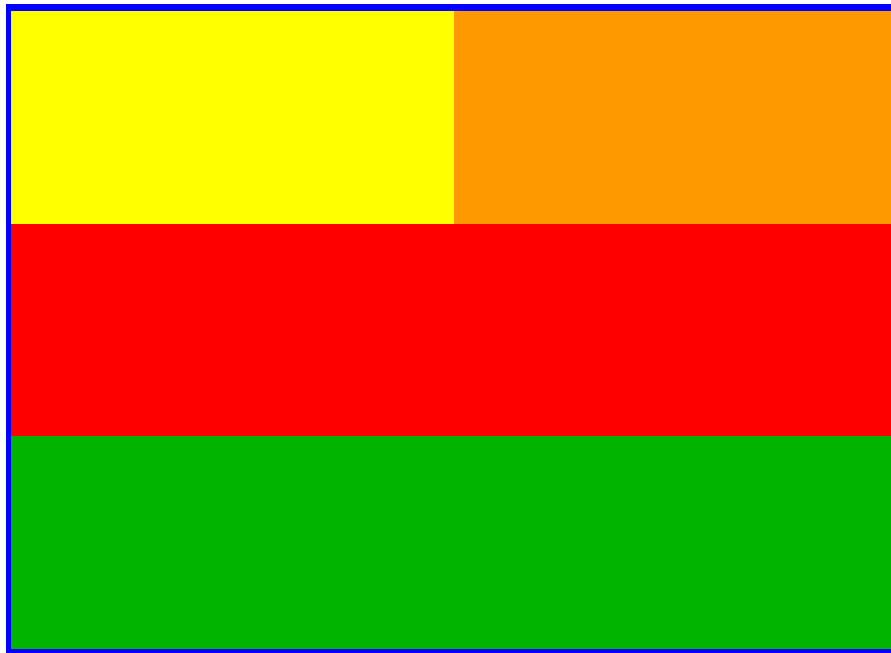
```
.caja-azul {  
  display: flex;  
  align-content: space-around;  
}
```



Flexbox

== align-content ==

```
.caja-azul {  
  display: flex;  
  align-content: stretch;  
}
```



Flexbox

== flex-basis ==

Flex-basis

Especifica el tamaño inicial del elemento flexible, antes de que cualquier espacio disponible se distribuya de acuerdo con los factores de flexión

number: Una unidad de largo, o porcentaje que especifica el largo inicial de un elemento.

auto: Valor default. El largo es igual al largo del elemento con flex. Si el ítem no tiene un largo determinado, el largo va a ser acorde a su contenedor.

initial: Setea esta propiedad a su valor default.

inherit: Hereda las propiedad de su padre.

Atentil: cambia el comportamiento según la propiedad flex-wrap

Flexbox

== flex-basis ==

```
.caja-azul {  
  display: flex;  
}  
.caja-amarilla{  
  flex-basis:30%;  
}  
.caja-roja{  
  flex-basis:70%;  
}
```





Practiquemos

Realizar los puntos de la guía de ejercicios.



Gracias!



Suerte!!!

ericardo.bianchi@gmail.com

