

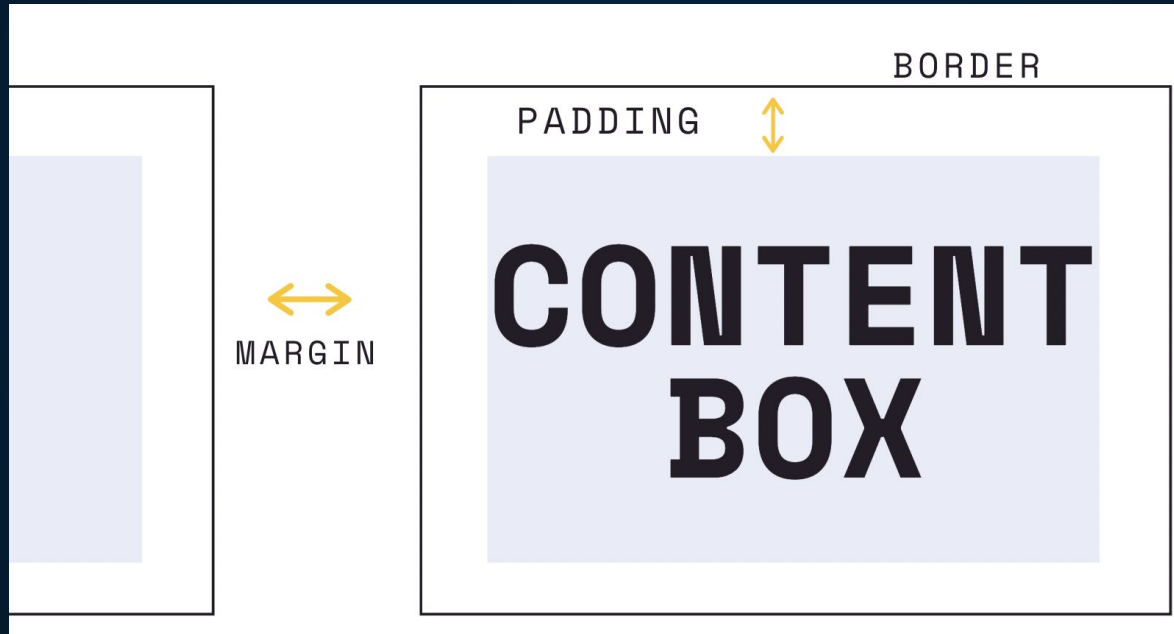


# Ing. Erick Gutiérrez



# Propiedad: Border y Padding

---



# Propiedad: Border

---

## BORDER-WIDTH

Controls the thickness of the border.

## BORDER-COLOR

Controls the...color of the border

## BORDER-STYLE

Controls the line style - dashed, solid, etc.

# Propiedad: Padding

---

## Propiedades individuales

```
PADDING-LEFT  
PADDING-RIGHT  
PADDING-BOTTOM  
PADDING-TOP
```

O puedes modificar todas las anteriores...

**Aplicar a todos los lados**

```
padding: 10px
```

**vertical | horizontal**

```
padding: 5px 10px;
```

**top | horizontal | bottom**

```
padding: 1px 2px 2px;
```

**top | right | bottom | left**

```
padding: 5px 1px 0 2px;
```

# Propiedad: Margin

---

## Propiedades individuales

MARGIN-LEFT

MARGIN-RIGHT

MARGIN-BOTTOM

MARGIN-TOP

O puedes modificar todas las anteriores...

**Aplicar a todos los lados**

`margin: 10px`

`vertical | horizontal`

`margin: 5px 10px;`

`top | horizontal | bottom`

`margin: 1px 2px 2px;`

`top | right | bottom | left`

`margin: 5px 1px 0 2px;`

# Propiedad: Display

---

## Elemento: Inline



Los elementos en línea encajan junto con otros elementos.

Los elementos a nivel de bloque ocupan un "bloque" completo de espacio



## Elemento: Block

# Propiedad: Display

---

- **block:** hace que el comportamiento del elemento sea como un bloque. (width, margin & padding son respetados)
- **inline:** el elemento se renderizará en línea con otros elementos. (width & height son ignorados)
- **inline-block:** el elemento tendrá un comportamiento mezcla entre los dos anteriores, que ahora voy a describir. (width, height, margin & padding son respetados)

# Unidades en CSS

---

## Relativas:

- EM
- REM
- VH
- VW
- %
- ... y más

## Absolutas:

- PX
- PT
- CM
- IN
- MM



# Unidades absolutas

---

**Px - Por mucho la unidad absoluta más utilizada:**

**1px no es necesariamente igual al ancho exacto de un pixel.**

**No es recomendado para sitios responsivos.**

# Unidades relativas

---

**Porcentaje**, es siempre relativo a otro valor:

En ocasiones, es un valor tomado desde el padre y otras veces es un valor tomado desde el elemento en sí.

**width: 50%** - Mitad del ancho del elemento padre.

**line-height: 50%** - Mitad del tamaño de la fuente del elemento en sí.

# Unidades relativas

---

**Em's:**

**Con font-size, 1em es igual al font-size del padre.  
2em's es el doble del tamaño del font-size del padre.**

**Con otras propiedades, 1em es igual al font-size computado del elemento en sí.**

# Unidades relativas

---

**rem:**

**Relativo a la raíz del elemento html (font-size).**

**Si la raíz contiene font-size 20px, 1 rem es siempre 20px, 2 rem 40px y así sucesivamente...**

# Introducción a flexbox

---

# Lo básico de Flexbox

---

## ¿Qué es?

Flexbox es un método de diseño unidimensional para colocar elementos en filas y/o columnas.

## ¿Cuánto tiempo se ha utilizado?

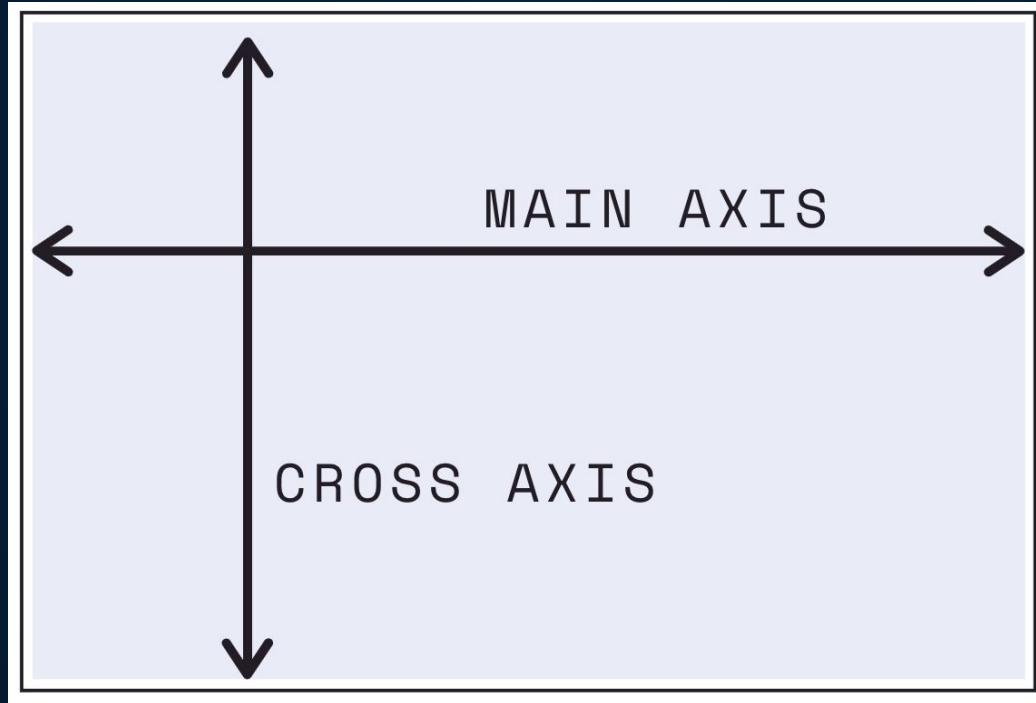
Flexbox es una reciente adición a CSS, se incluyó para resolver frustraciones de diseño.

## ¿Por qué 'flex'?

Nos permite distribuir espacio de forma dinámica entre los elementos de tamaño desconocido, de ahí el término 'flex'.

# Modelo: Flex

---



# FLEX DIRECTION

---



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



# FLEX DIRECTION

---



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

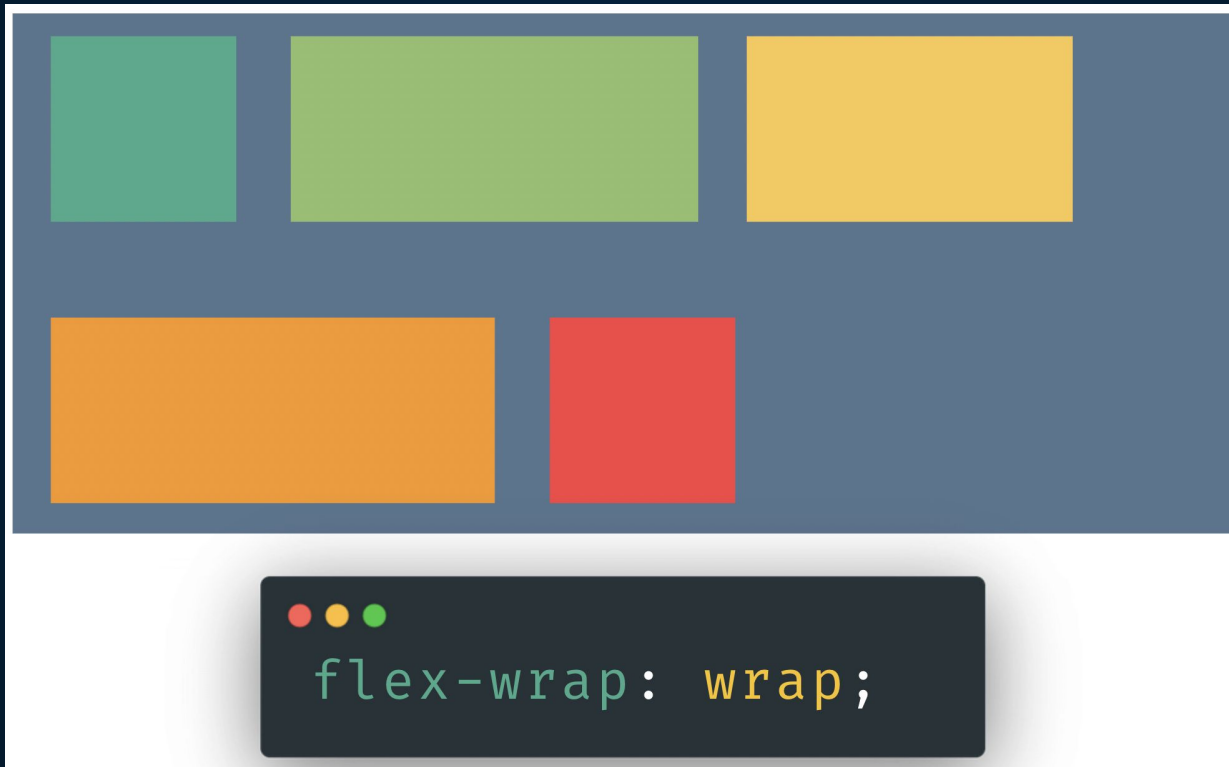
# FLEX DIRECTION

---



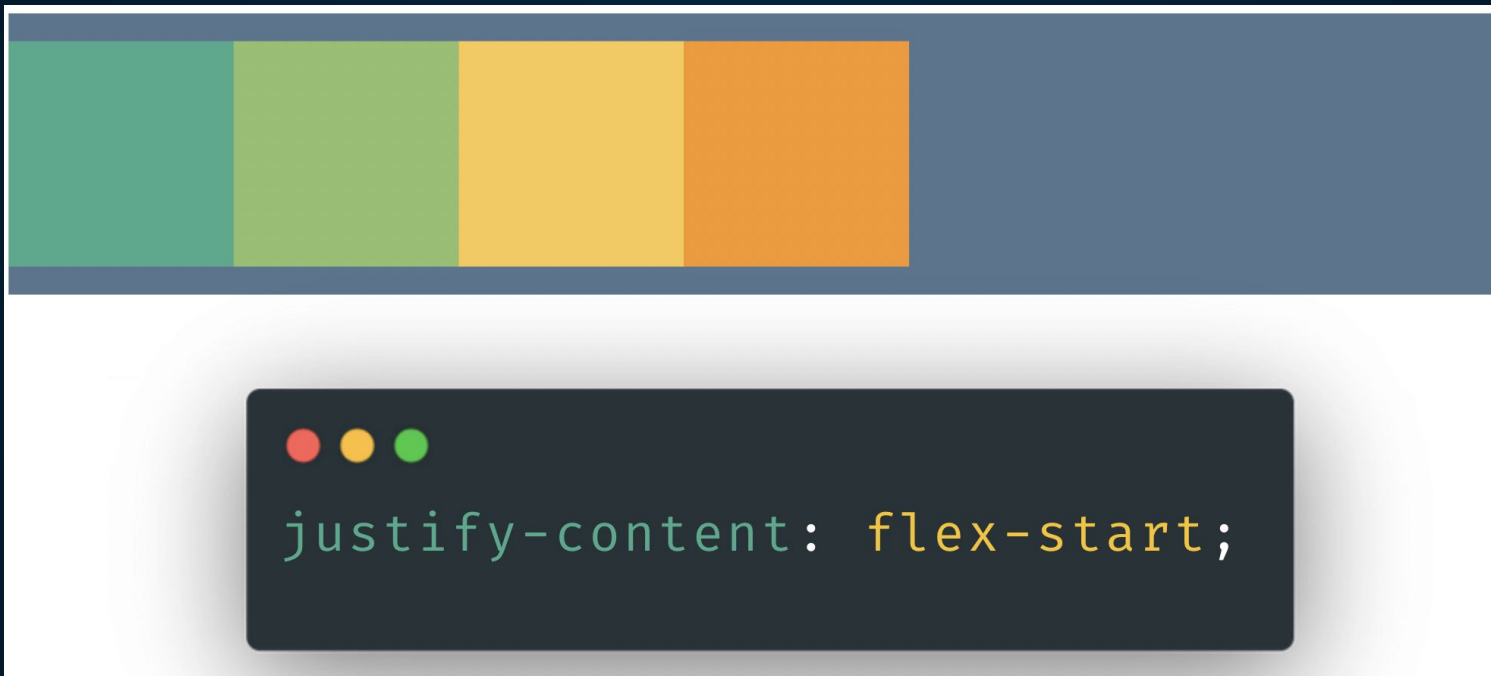
# FLEX WRAP

---



# JUSTIFY CONTENT

---



# JUSTIFY CONTENT

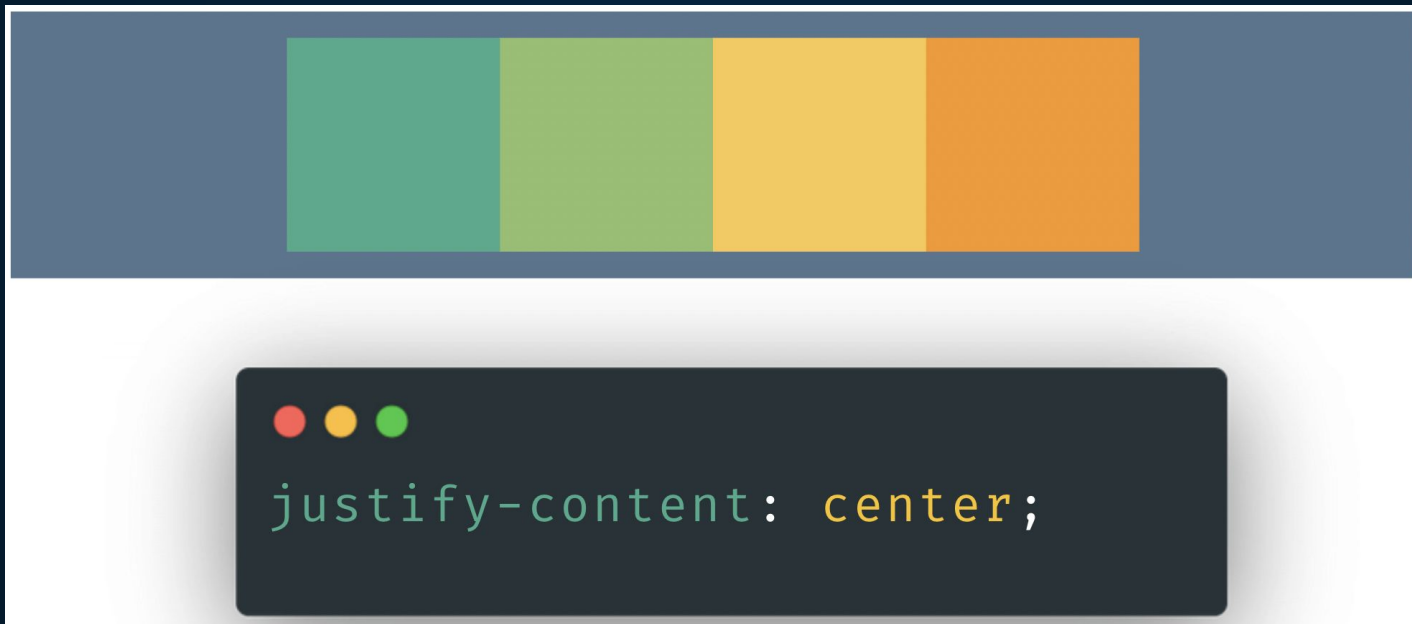
---



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

# JUSTIFY CONTENT

---



# JUSTIFY CONTENT

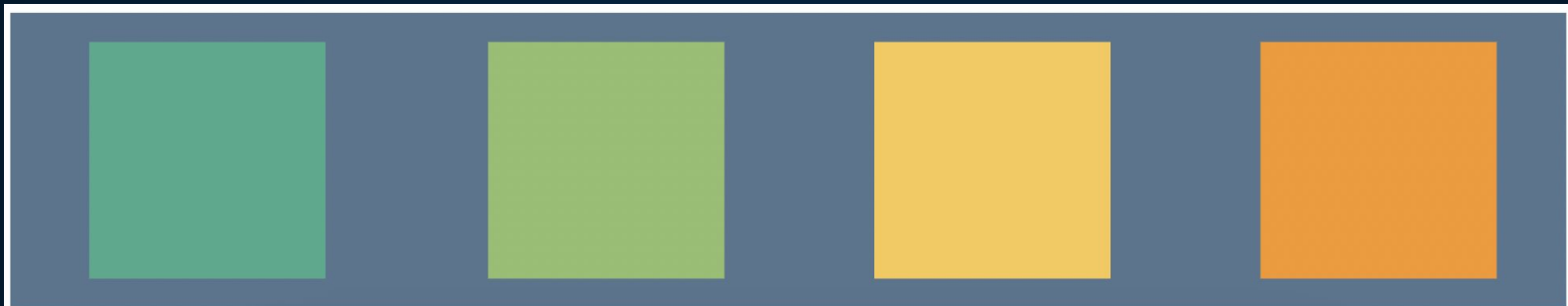
---



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

# JUSTIFY CONTENT

---

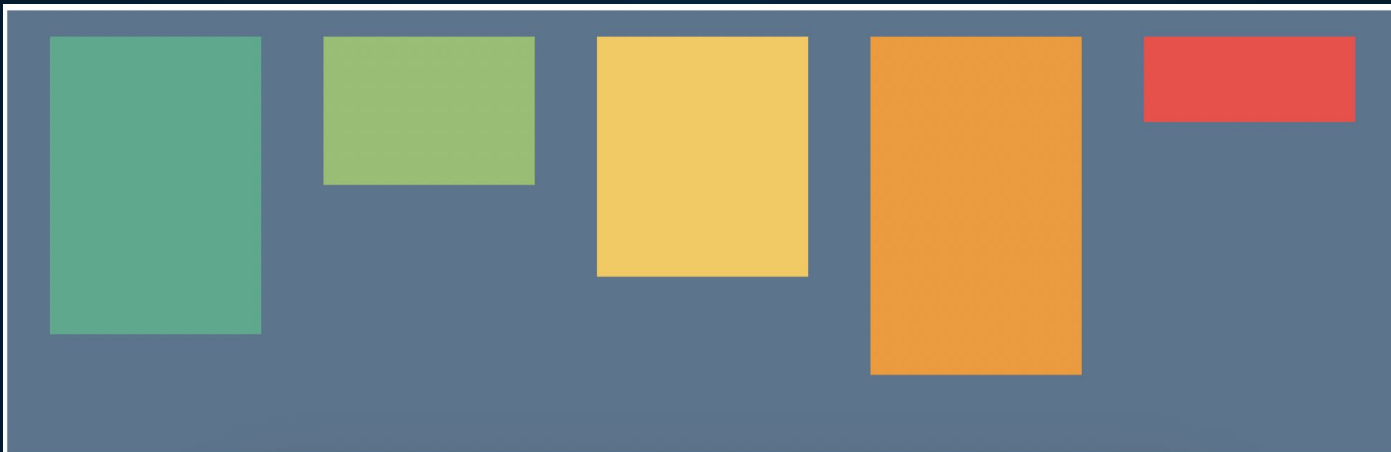


```
justify-content: space-around;
```



# ALIGN ITEMS

---



```
align-items: flex-start;
```

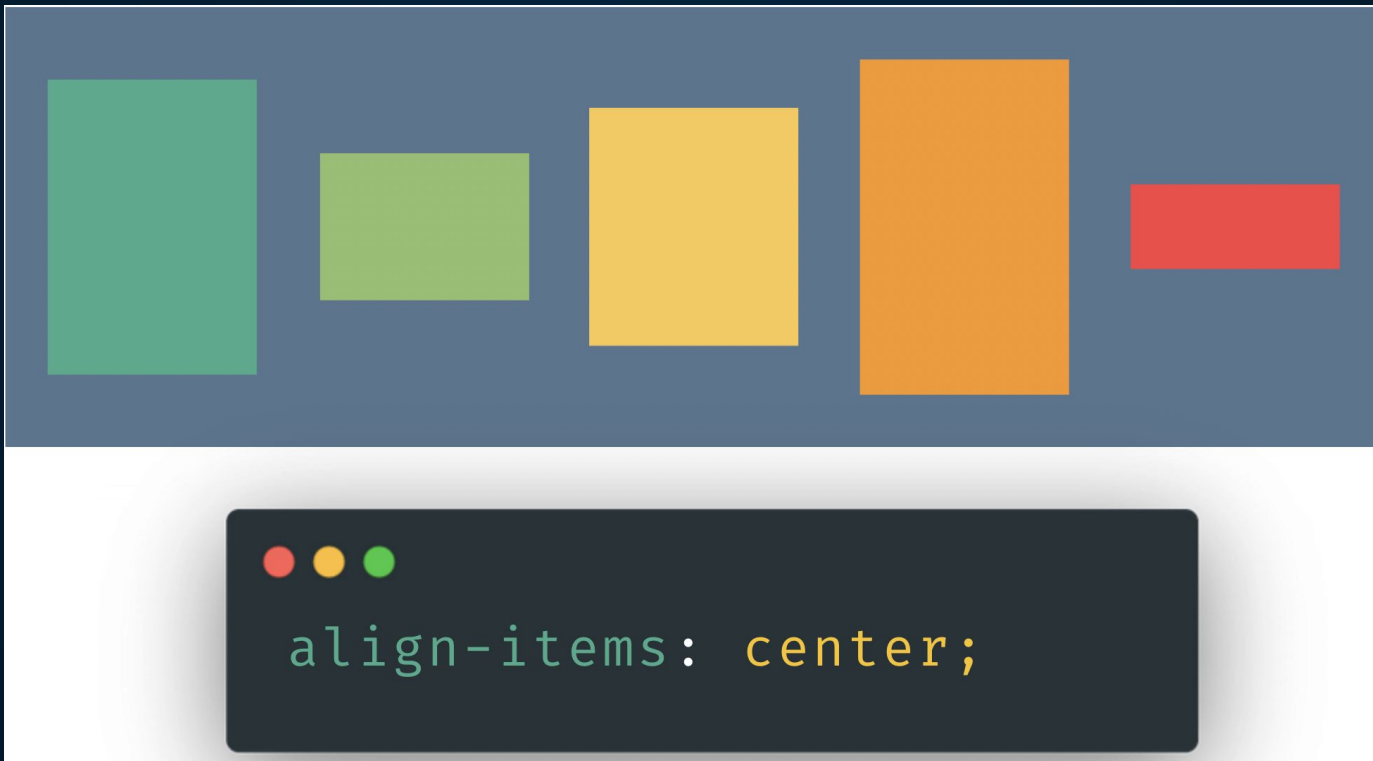
# ALIGN ITEMS

---



# ALIGN ITEMS

---



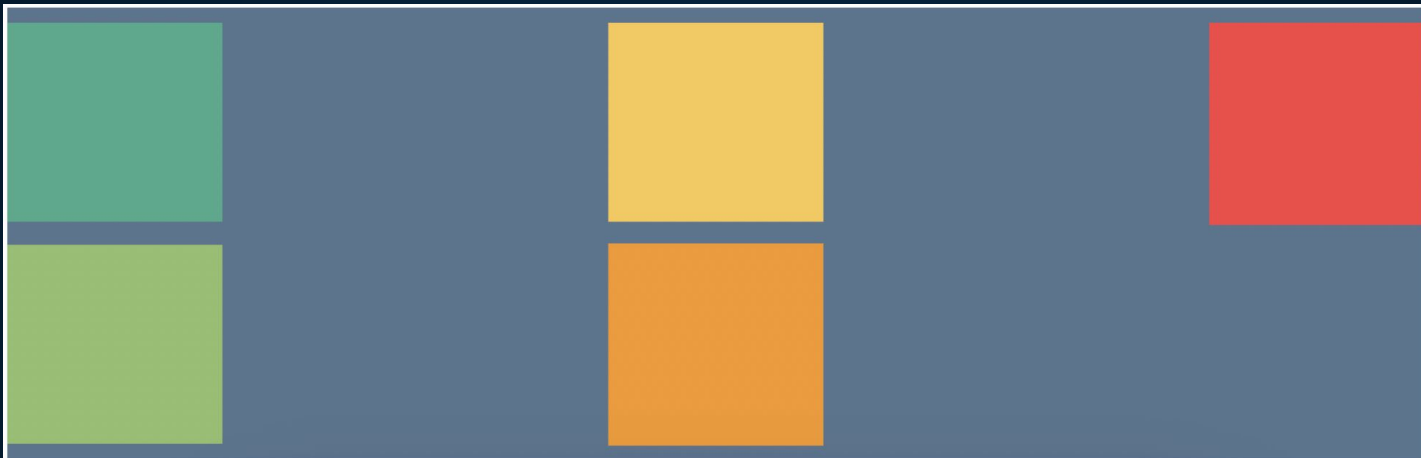
# ALIGN ITEMS

---



# ALIGN ITEMS

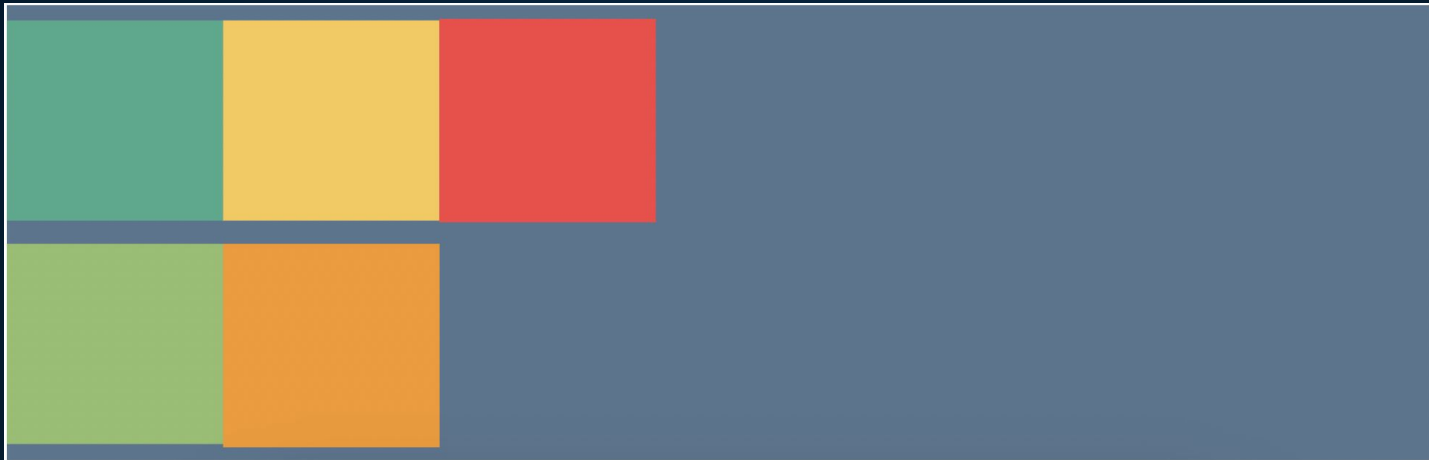
---



```
align-content:space-between;
```

# ALIGN ITEMS

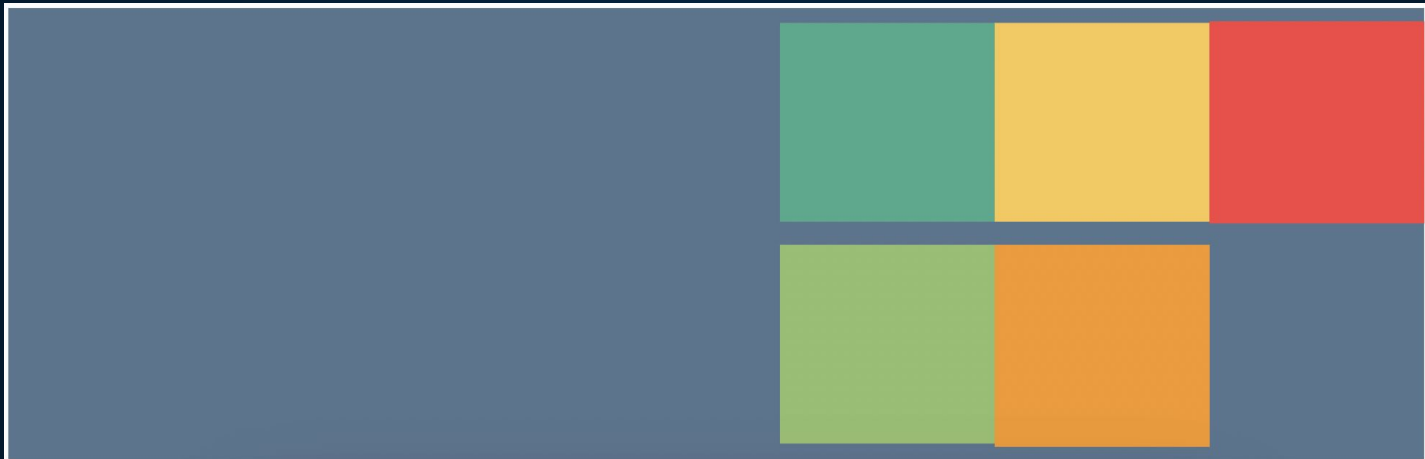
---



```
align-content: flex-start;
```

# ALIGN ITEMS

---



```
align-content: flex-end;
```

# ALIGN ITEMS

---

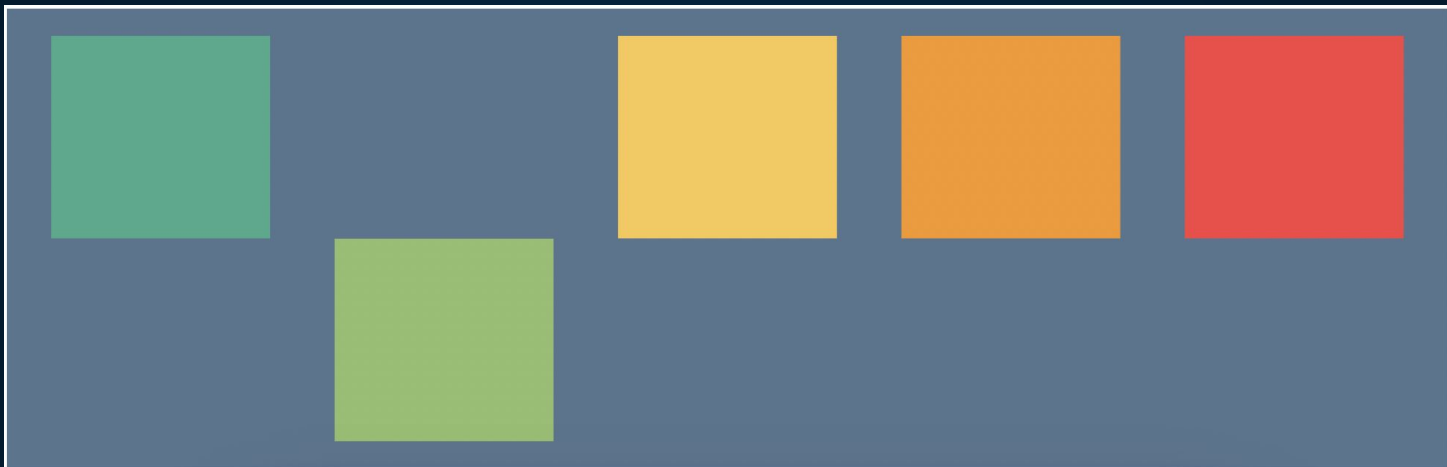


```
align-content:center;
```



# ALIGN SELF

---



```
align-self: flex-end;
```

# Propiedades: Flex-sizing

---

- **Flex-basis**
- **Flex-grow**
- **Flex-shrink**

# Flex-basis

---

La propiedad flex-basis, como su nombre lo indica, define el **main-size base** para un **flex-item**. Tamaño base principal (main size base) significa que no necesariamente ese será su tamaño al dibujarse por el navegador, pero que será un punto de partida para calcular el tamaño final.

En otras palabras, si el **main axis** es horizontal (predeterminado), flex-basis será equivalente a width; y si el **main axis** es vertical, flex-basis será equivalente a height.

## Flex-basis

---

Ten en cuenta que el tamaño definido por flex-basis es, como su nombre lo dice, el tamaño **base**. Es decir, que podrá variar (crecer o encogerse), según los valores de flex-grow y flex-shrink que veremos más adelante.

## Flex-grow

---

La propiedad flex-grow, como su nombre lo indica, controla que tanto crecerá el flex-item para rellenar el espacio disponible. Su valor solo puede ser un número entero (no negativo).

Esta propiedad solo se aplica si el espacio disponible es positivo (el tamaño del contenedor es mayor a la suma de los flex-items y sus márgenes).

## ¿Qué significa el valor de flex-grow?

---

En flex-grow:  $x$  el numero  $x$  indica cuantas unidades crecerá el item para calcular su tamaño final. Por ejemplo, flex-grow: 3 significa que el item crecerá 3 unidades.

## Flex-shrink

---

Si el espacio disponible es negativo (el tamaño del contenedor es menor a la suma de los tamaños de los items), de forma predeterminada los items se encogen en proporciones iguales para caber en una sola línea.

Con la propiedad flex-shrink se controla cómo se encogerán los elementos.

# Flex

---

Es el shorthand (propiedad abreviada) para flex-grow, flex-shrink y flex-basis (en ese orden específico).

*Entonces:*

```
.item { flex: 2 1 200px; }
```

=

```
.item { flex-grow: 2; flex-shrink: 1; flex-basis: 200px; }
```

El valor predeterminado de flex (si no se especifica) es: **flex: 0 1 auto;**

Es decir que por defecto los items no van a crecer pero si reducirse de manera proporcional para ajustarse al espacio. Además, que flex-basis se establece en auto (respecto a su contenido).