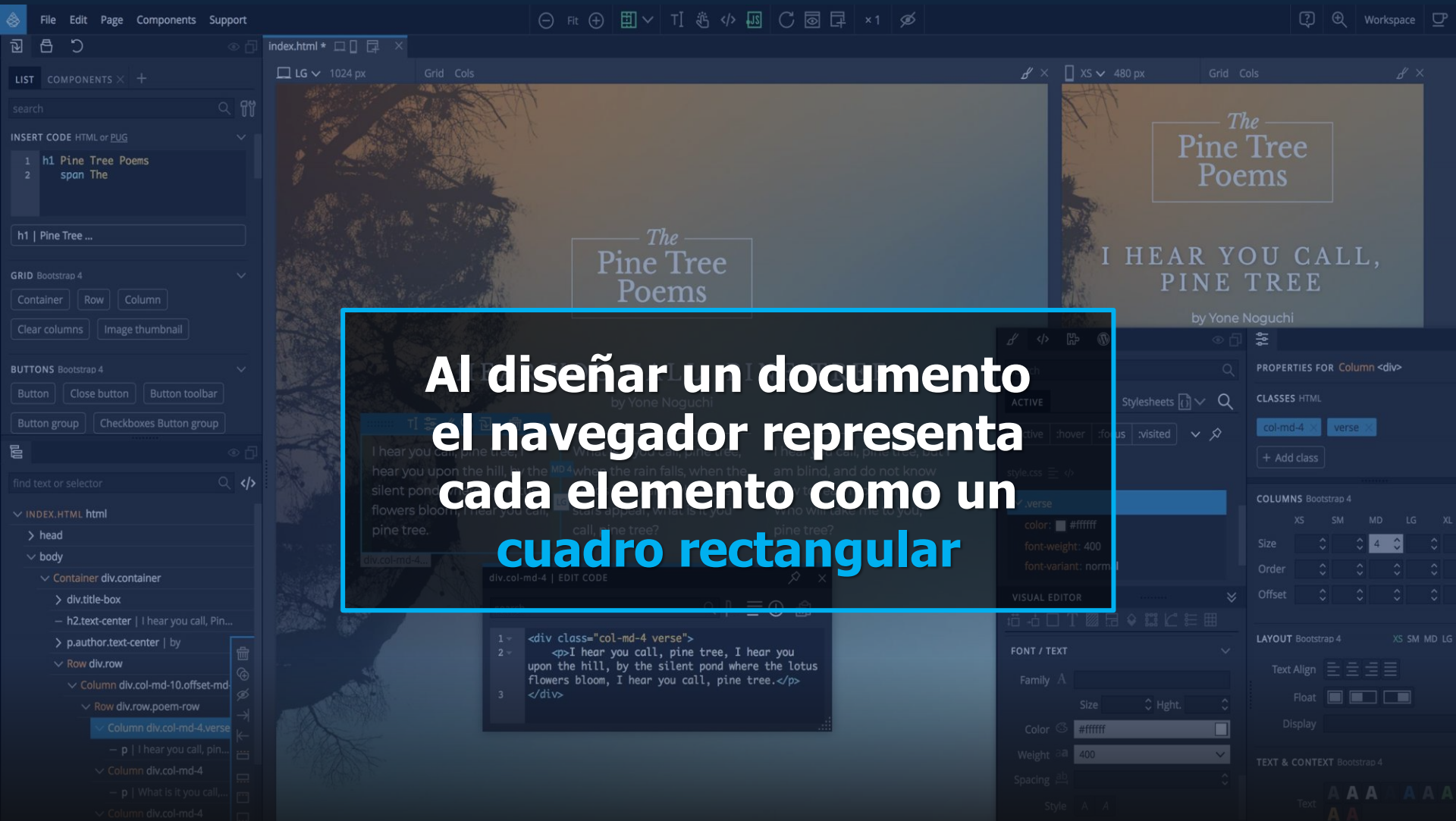


SESIÓN  
**02** | **ESTRUCTURA**



# **01** | **BOX MODEL**





# Box Model

El modelo de caja CSS es una **caja 2D** que **envuelve** cada elemento HTML.

**Ancho**

**Margen**

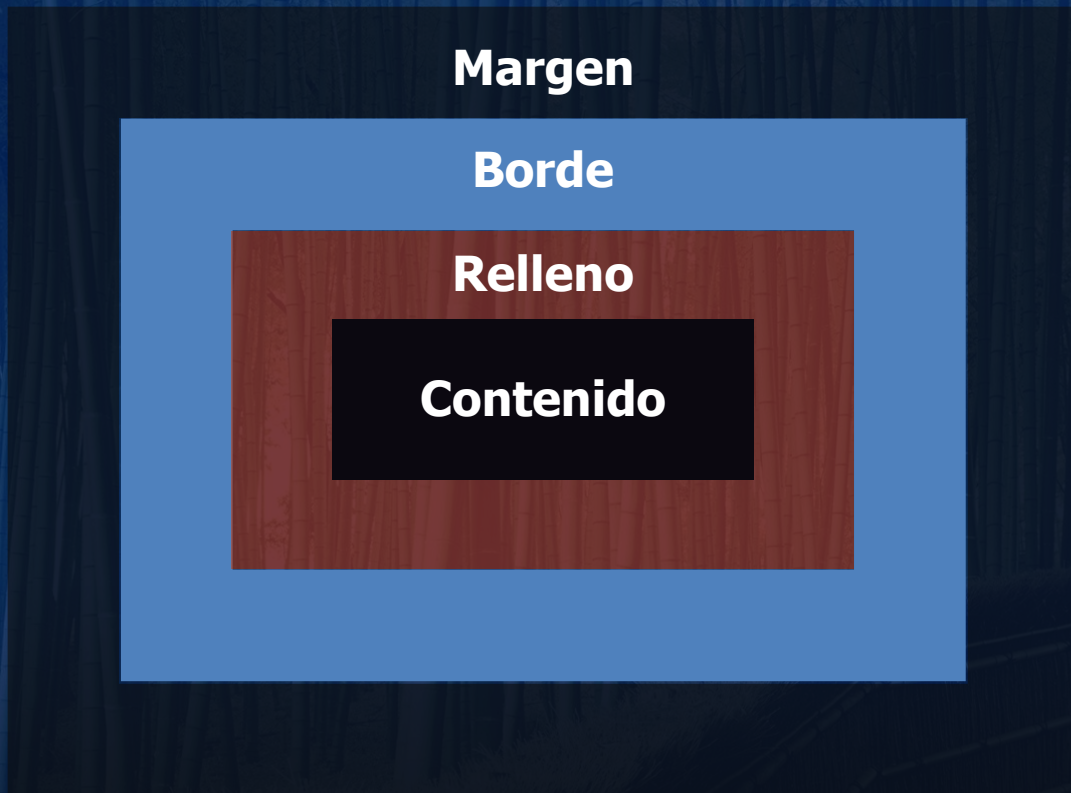
**Borde**

**Relleno**

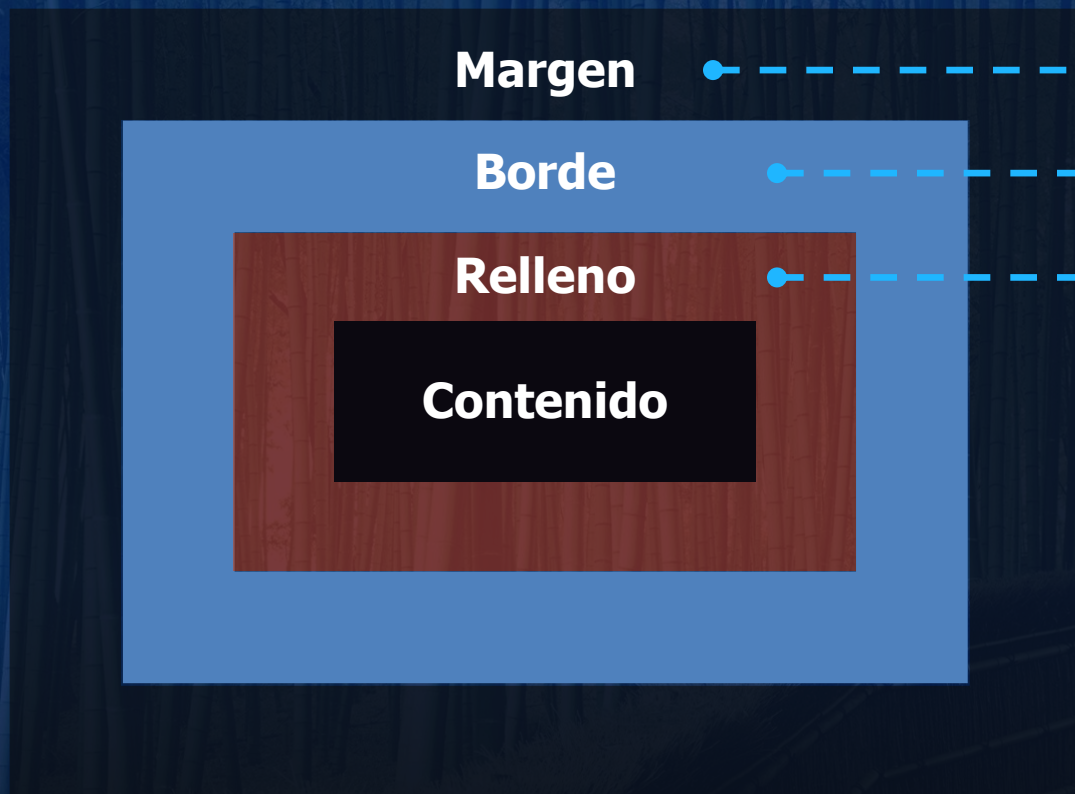
**Contenido**

**Altura**

**Box Model**







**Margen**

**Borde**

**Relleno**

**Contenido**

**margin**

**border**

**padding**

## Box Model

Ancho ● —> **width**

Altura ● —> **height**

## Box Model • Ancho y altura

<b>width</b>	<b>Anchura</b>
<b>Valores</b>	<medida>   <porcentaje>   auto
<b>Valor inicial</b>	auto
<b>Descripción</b>	Establece la anchura de un elemento. No admite valores negativos

<b>height</b>	<b>Altura</b>
<b>Valores</b>	<medida>   <porcentaje>   auto
<b>Valor inicial</b>	auto
<b>Descripción</b>	Establece la altura de un elemento. No admite valores negativos

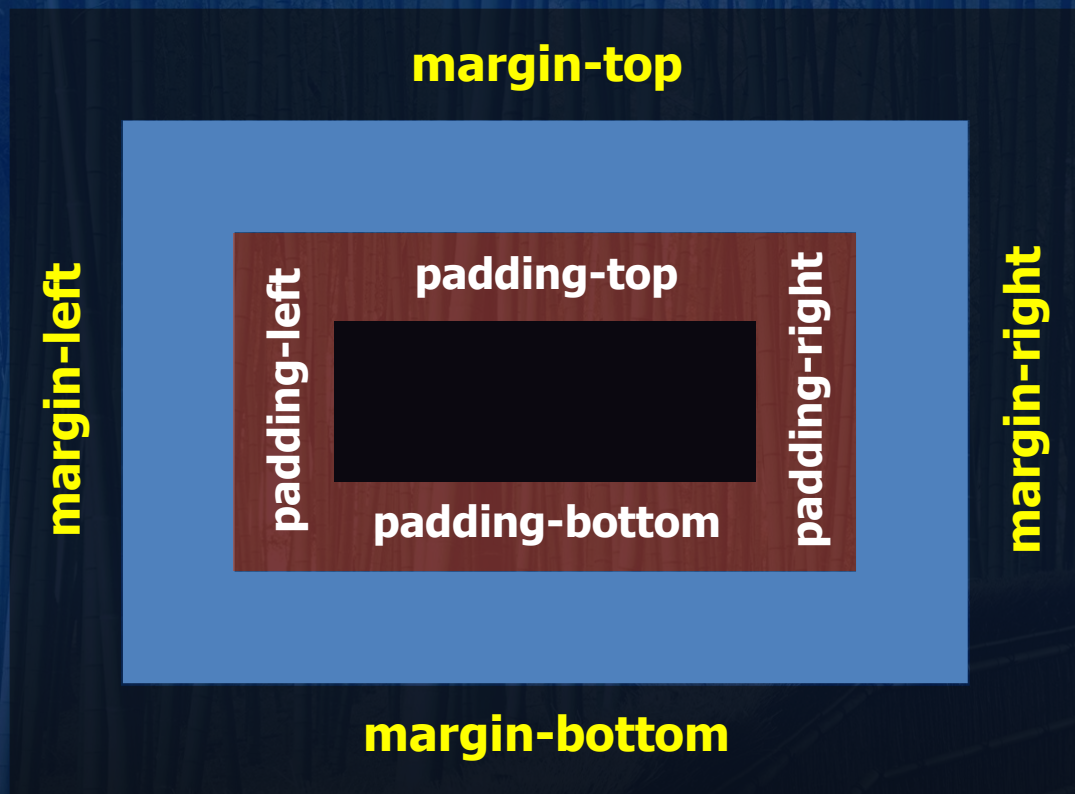
# Box Model • Margen y relleno

<b>margin-top</b> <b>margin-right</b> <b>margin-bottom</b> <b>margin-left</b>	<b>Margen superior</b> <b>Margen derecho</b> <b>Margen inferior</b> <b>Margen izquierdo</b>
<b>Valores</b>	<medida>   <porcentaje>   auto
<b>Valor inicial</b>	0
<b>Descripción</b>	Establece cada uno de los márgenes horizontales y verticales de un elemento



## Box Model • Margen y relleno

<b>padding-top</b> <b>padding-right</b> <b>padding-bottom</b> <b>padding-left</b>	<b>Relleno superior</b> <b>Relleno derecho</b> <b>Relleno inferior</b> <b>Relleno izquierdo</b>
<b>Valores</b>	<medida>   <porcentaje>   auto
<b>Valor inicial</b>	0
<b>Descripción</b>	Establece cada uno de los rellenos horizontales y verticales de un elemento



## Box Model

Orden:  
Sentido horario

```
div img {  
    margin-top: 20%;  
    margin-right: 20%;  
    margin-bottom: 20%;  
    margin-left: 20%;  
    padding-top: 20%;  
    padding-right: 20%;  
    padding-bottom: 20%;  
    padding-left: 20%;  
}
```

**¿No es esto  
mucho código?**





<b>margin</b>	<b>Margen</b>
<b>Valores</b>	margin-top   margin-right   margin-bottom   margin-left
<b>Descripción</b>	Establece todas las propiedades del margen del elemento.

<b>padding</b>	<b>Relleno</b>
<b>Valores</b>	padding-top   padding-right   padding-bottom   padding-left
<b>Descripción</b>	Establece todas las propiedades del relleno del elemento.

```
div img {  
  margin: 3% 1% 2% 4%;  
  padding: 4% 3% 1% 2%;  
}
```



**4 valores**

top | right | bottom | left

---

```
div img {  
  margin: 2% 1% ;  
  padding: 4% 6% ;  
}
```



**2 valores**

top-bottom | right-left

---

```
div img {  
  margin: 1%;  
  padding: 1%;  
}
```



**1 valor**

all



<b>border-width</b>	<b>Grosor del borde</b>
<b>Valores</b>	<medida>   thin   medium   thick
<b>Valor inicial</b>	medium
<b>Descripción</b>	Establece la anchura de los 4 bordes de los elementos

<b>border-color</b>	<b>Color del borde</b>
<b>Valores</b>	<color>   transparent
<b>Valor inicial</b>	black
<b>Descripción</b>	Establece el color de los 4 bordes de los elementos

<b>border-style</b>	<b>Estilo del borde</b>
<b>Valores</b>	none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset
<b>Valor inicial</b>	none
<b>Descripción</b>	Establece el estilo de los 4 bordes de los elementos

<b>border</b>	<b>Borde</b>
<b>Valores</b>	<medida_borde>    <color_borde>    <estilo_borde>
<b>Descripción</b>	Establece el estilo completo de todos los bordes de los elementos

## 02 | FLEXBOX

A man and a woman are shown in profile, working together at a desk with two computer monitors. The scene is dimly lit, with a desk lamp providing light. The background is blurred, showing office shelves. The entire image has a blue color overlay.

# FlexBox

Sistema de elementos flexibles CSS, que permite una mecánica más **potente, limpia y personalizable** de estructura y contenidos, los cuales **se adaptan automáticamente al dispositivo**.

# Contenedor

## Elemento padre

Elemento que tendrá **en su interior** cada uno de los **ítems flexibles**.

Se le asigna las **propiedades** CSS necesarias.

w3schools/CSS FlexBox



Contenedor

Elemento padre



```
.flex-container {
```

```
  display: flex;
```

```
}
```

Conceptos • Elementos básicos

# Ítem

## Elemento hijo

**Contenidos** flexibles en el **interior** de un contenedor flex.

Hereda las **propiedades** CSS del elemento padre.

w3schools/CSS FlexBox





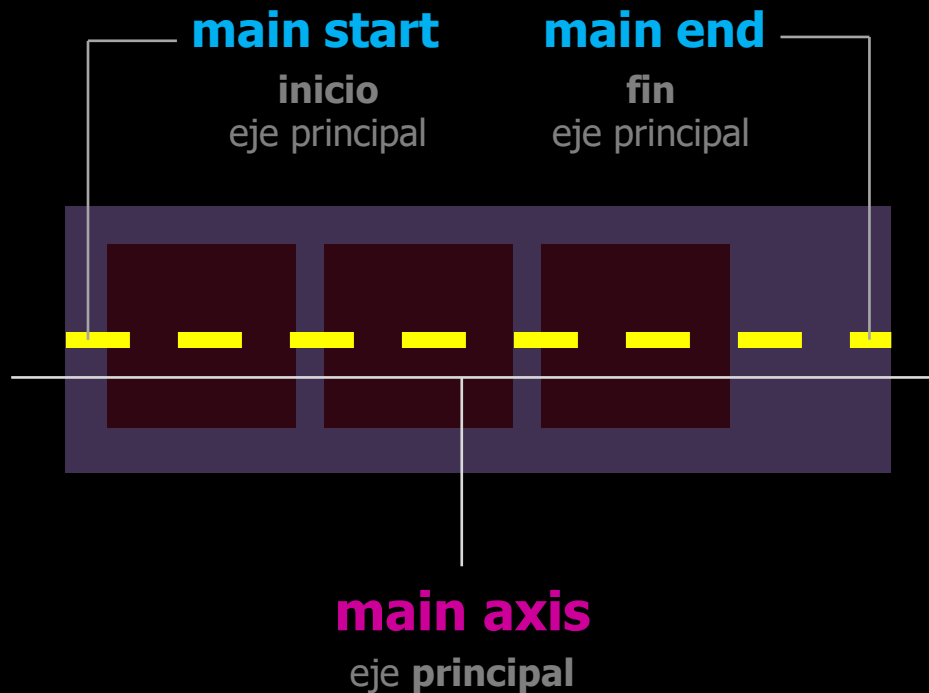
# Main Axis

## Eje principal

Los contenedores flexibles tendrán una orientación principal específica.

Por defecto, es **horizontal** (en fila).

w3schools/CSS FlexBox



# Cross Axis

## Eje secundario

Orientación **perpendicular** a la principal. Si la principal es en horizontal, la secundaria será en vertical, y viceversa.

w3schools/CSS FlexBox

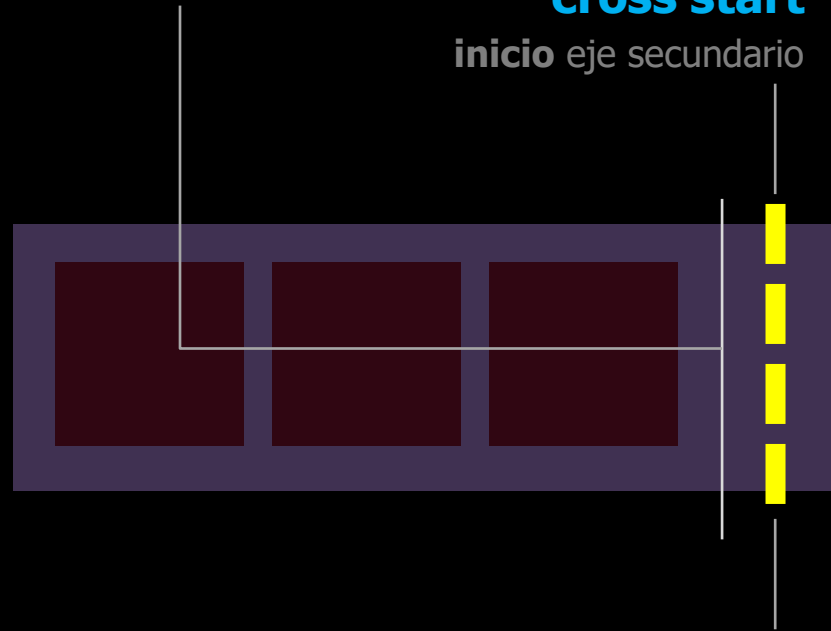


**cross axis**

eje **secundario**

**cross start**

**inicio** eje secundario



**cross end**

**fin** eje secundario

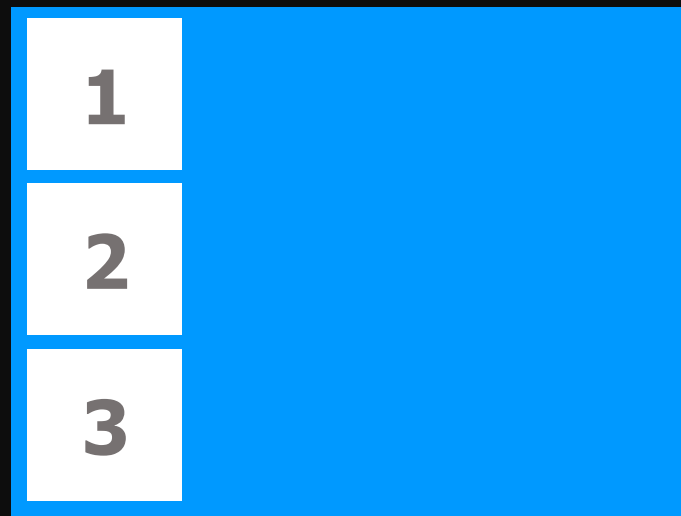
## **02** | **CONTENEDORES**

Propiedades • Contenedores flex

# flex-direction

Define en qué **dirección** el contenedor quiere **apilar los elementos flexibles**, en base al **eje principal**.

w3schools/CSS FlexBox Container



Propiedad • flex-direction

Valor

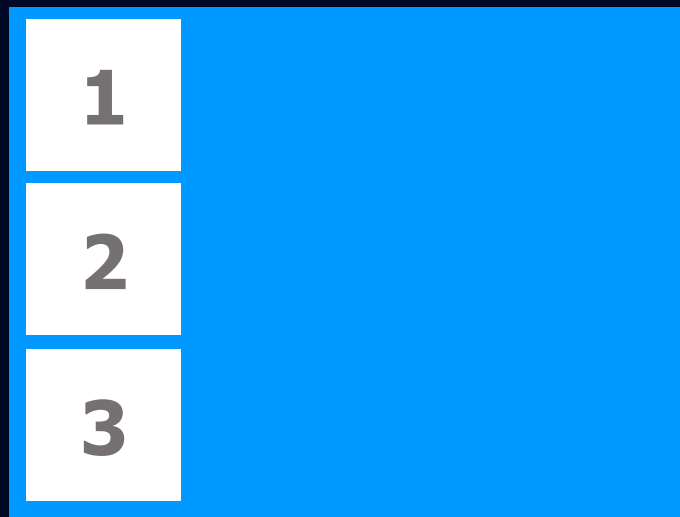
# column

**Apila** los elementos flexibles **verticalmente** (de arriba a abajo).

w3schools/CSS FlexBox Container



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



Propiedad • flex-direction

Valor

# column-reverse

**Apila** los elementos flexibles **verticalmente pero en orden inverso** (de abajo a arriba).

w3schools/CSS FlexBox Container



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





Propiedad • flex-direction

Valor

# row

**Apila** los elementos flexibles **horizontalmente** (de izquierda a derecha).

w3schools/CSS FlexBox Container



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



Propiedad • flex-direction

Valor

# row-reverse

**Apila** los elementos flexibles **horizontalmente pero en orden inverso** (de derecha a izquierda).

w3schools/CSS FlexBox Container



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

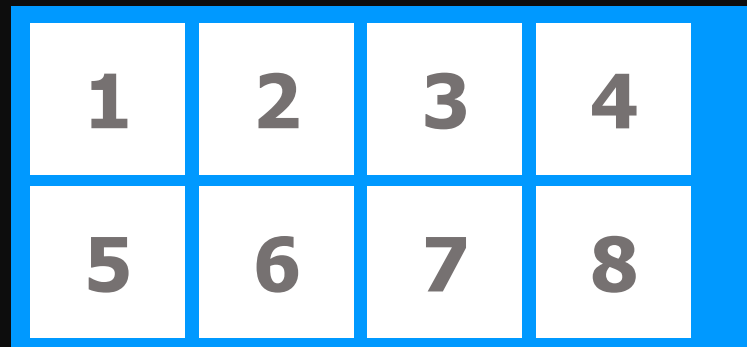


Propiedades • Contenedores flex

## flex-wrap

Especifica si los elementos flexibles **deben ajustarse o no al espacio**. Es decir, **evita o permite el desbordamiento**.

w3schools/CSS FlexBox Container



Propiedad • flex-wrap

Valor

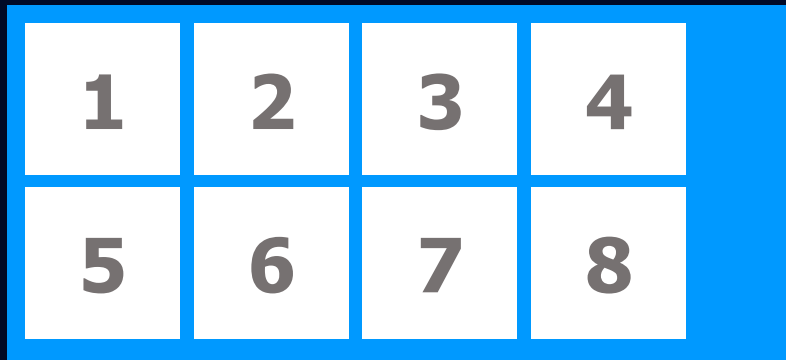
# wrap

Especifica que los elementos flexibles **se ajustarán** si es necesario. Establece los ítems en modo **multilínea** (permite que se **desborde** el contenedor).

w3schools/CSS FlexBox Container



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



Propiedad • flex-wrap

Valor

# nowrap

Especifica que los elementos flexibles **no se ajustarán** (valor predeterminado).

Establece los ítems en **una sola línea (no permite** que se **desborde** el contenedor).

w3schools/CSS FlexBox Container



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



Propiedad • flex-wrap

Valor

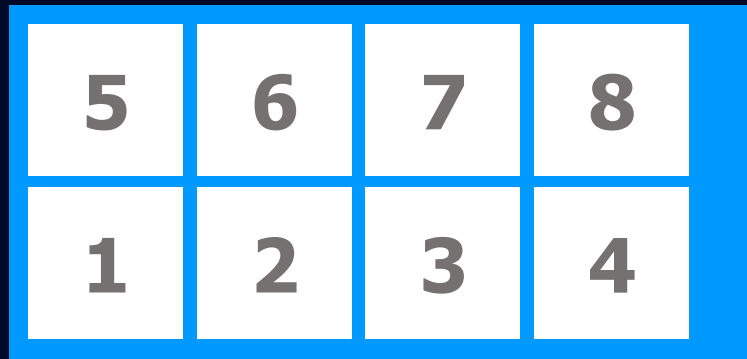
## wrap-reverse

Especifica que los elementos flexibles **se ajustarán** si es necesario, es decir, **multilínea en orden inverso**.

w3schools/CSS FlexBox Container



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



Propiedades • Contenedores flex

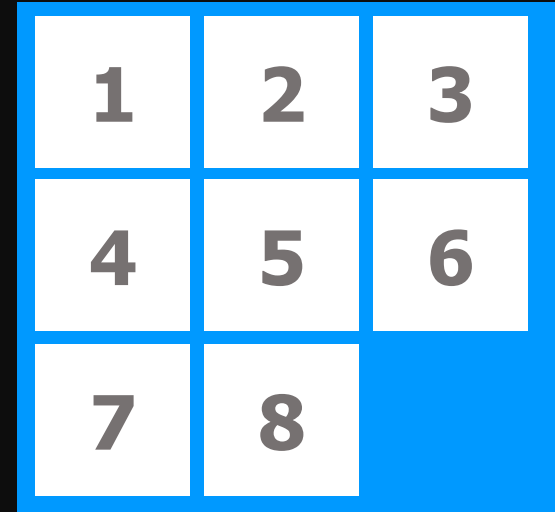
## flex-flow

**Shorthand** (propiedad abreviada) para establecer las propiedades **flex-direction** y **flex-wrap**.

w3schools/CSS FlexBox Container



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

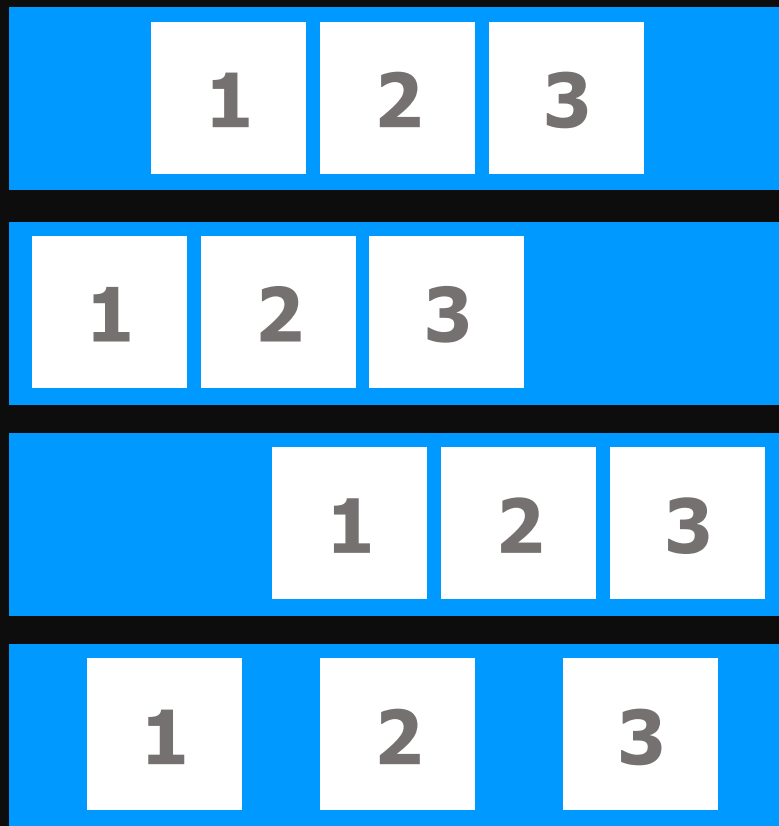


Propiedades • Contenedores flex

## justify-content

Propiedad que se usa para **alinear** los elementos flexibles del **eje principal** (por defecto, el horizontal).

w3schools/CSS FlexBox Container





Propiedad • justify-content

Valor

# center

---

Alinea los elementos flexibles en el **centro del eje principal del contenedor**.

w3schools/CSS FlexBox Container



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



Propiedad • justify-content

Valor

# flex-start

---

Alinea los elementos flexibles **al comienzo del eje principal del contenedor** (valor predeterminado).

w3schools/CSS FlexBox Container



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



Propiedad • justify-content

Valor

# flex-end

---

Alinea los elementos flexibles **al final del eje principal del contenedor.**

w3schools/CSS FlexBox Container



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



Propiedad • justify-content

Valor

# space-around

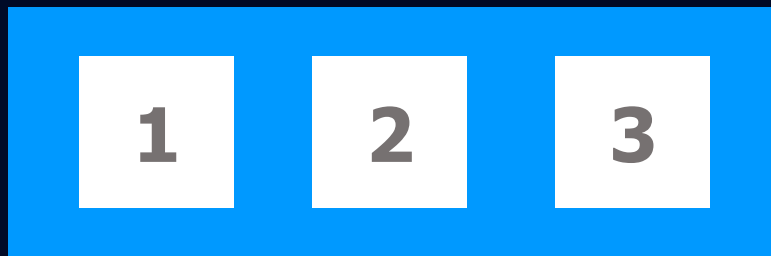
---

Muestra los elementos flexibles **dejando el mismo espacio entre ellos.**

w3schools/CSS FlexBox Container



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



Propiedad • justify-content

Valor

# space-between

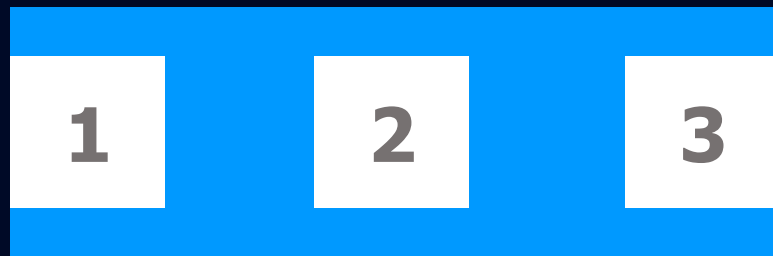
---

Muestra los elementos flexibles con **el máximo espacio entre los ítems** para separarlos.

w3schools/CSS FlexBox Container



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

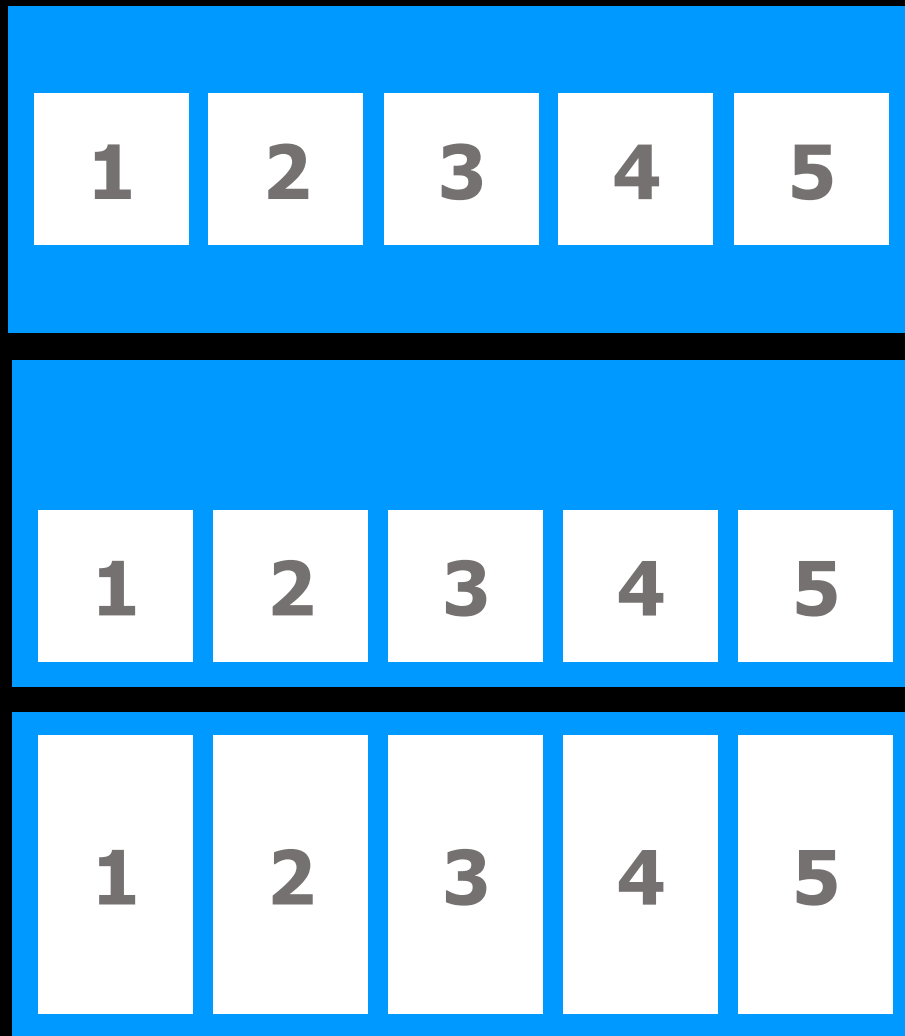


Propiedades • Contenedores flex

## align-items

Propiedad al tratar con un contenedor flex **de una línea de contenidos**: contenedor en el que los ítems **caben en el ancho disponible**, y el **eje secundario no se divide** en múltiples líneas.

w3schools/CSS FlexBox Container



Propiedad • align-items

Valor

# center

Alinea los elementos flexibles  
en el **medio** del **eje**  
**secundario** del contenedor.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: center;  
}
```

1

2

3

4

5

Propiedad • align-items

Valor

# flex-start

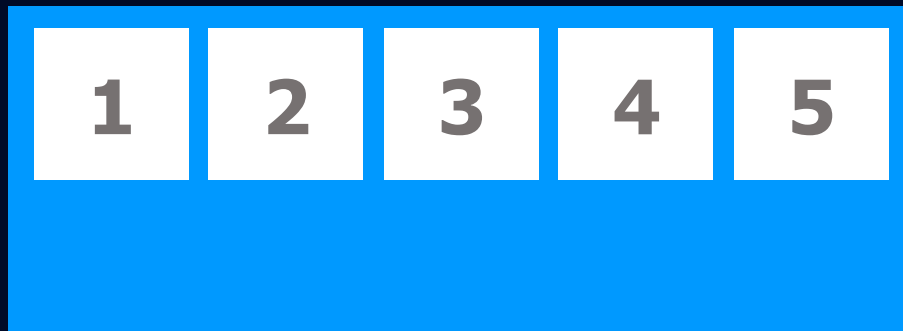
---

Alinea los elementos flexibles en el **inicio (parte superior)** del **eje secundario** del contenedor.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: flex-start;  
}
```





Propiedad • align-items

Valor

# flex-end

Alinea los elementos flexibles en el **final (parte inferior)** del **eje secundario** del contenedor.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: flex-end;  
}
```

1

2

3

4

5

Propiedad • align-items

Valor

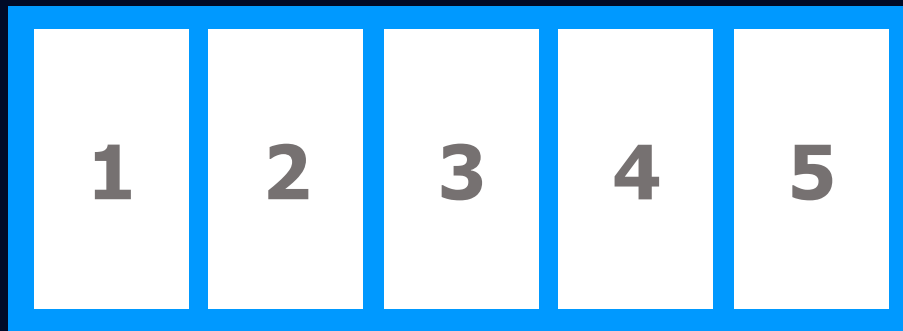
# stretch

**Estira** los elementos flexibles **para llenar** el contenedor de forma **equitativa** (valor predeterminado).

w3schools/CSS FlexBox Container



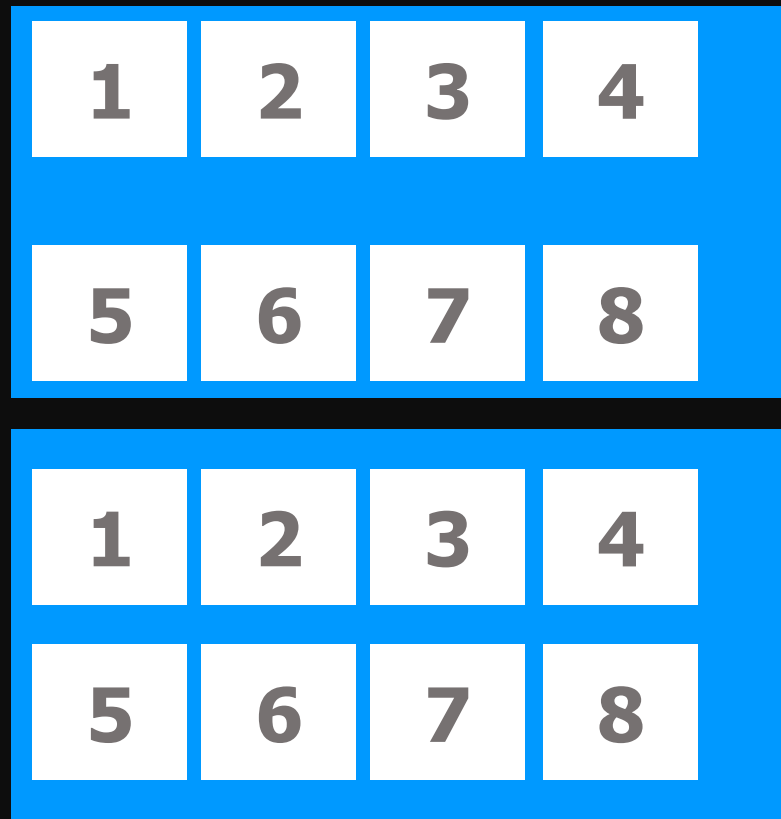
```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: stretch;  
}
```



# align-content

Propiedad al tratar con un contenedor flex **multilínea**: contenedor en el que los ítems **no caben en el ancho disponible**, y el **eje secundario se divide** en múltiples líneas.

w3schools/CSS FlexBox Container



Propiedad • align-content

Valor

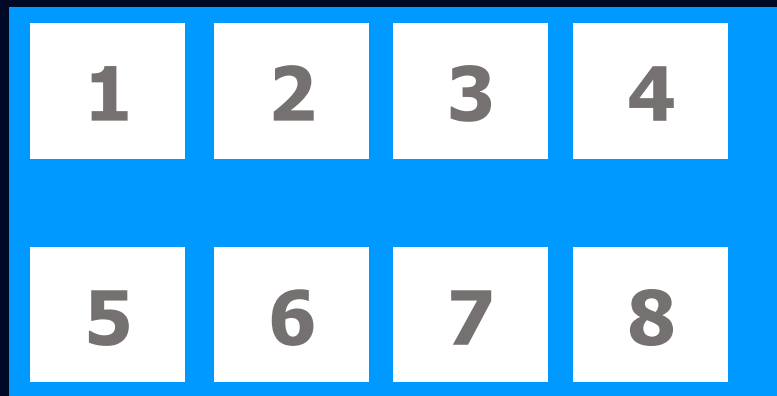
# space-between

**Distribuye** los ítems desde el inicio hasta el final, **con el máximo espacio** entre ellas.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: space-between;  
}
```



Propiedad • align-content

Valor

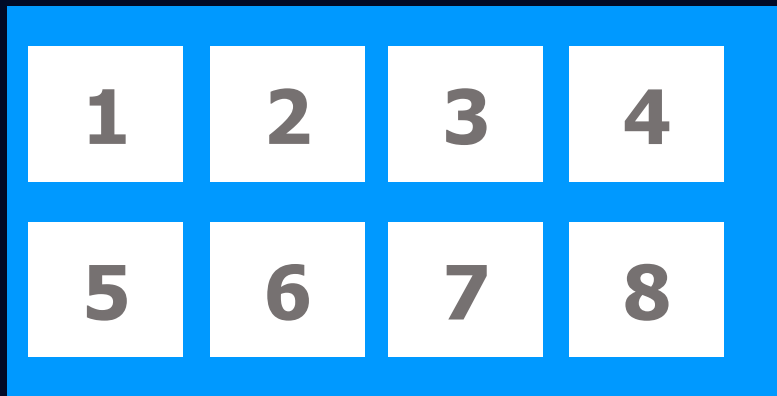
# space-around

**Distribuye** los ítems **con el mismo espacio** en los lados de cada uno de ellos.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: space-around;  
}
```



Propiedad • align-content

Valor

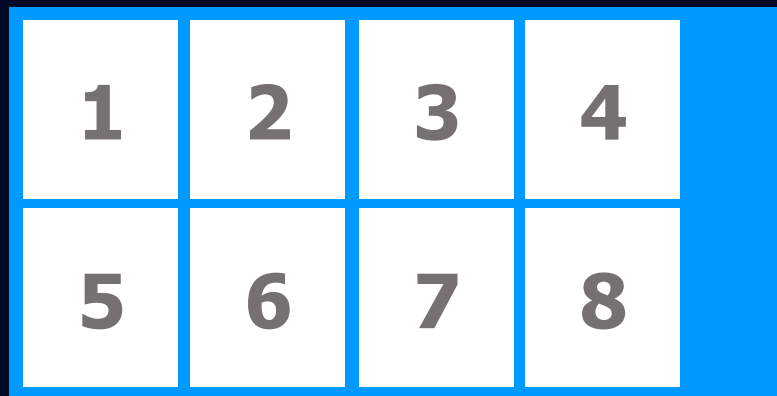
# stretch

Alinea los ítems **estirándolos** de modo que cubran desde el inicio hasta el final del contenedor.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: stretch;  
}
```



Propiedad • align-content

Valor

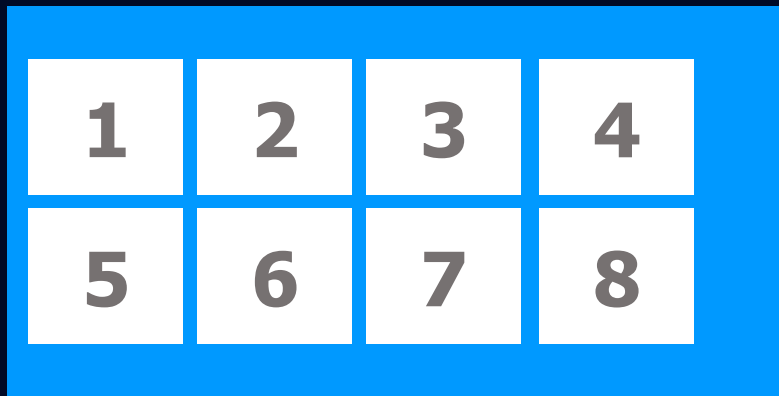
# center

Alinea los ítems **al centro del eje secundario.**

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: center;  
}
```



Propiedad • align-content

Valor

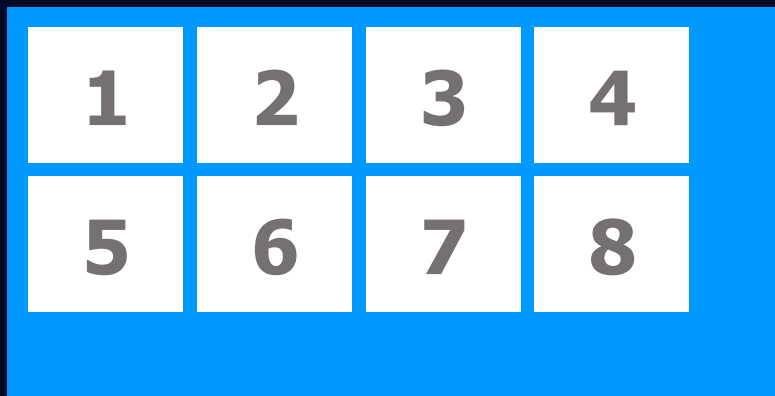
# flex-start

Agrupar los ítems **al inicio del eje secundario.**

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: flex-start;  
}
```





Propiedad • align-content

Valor

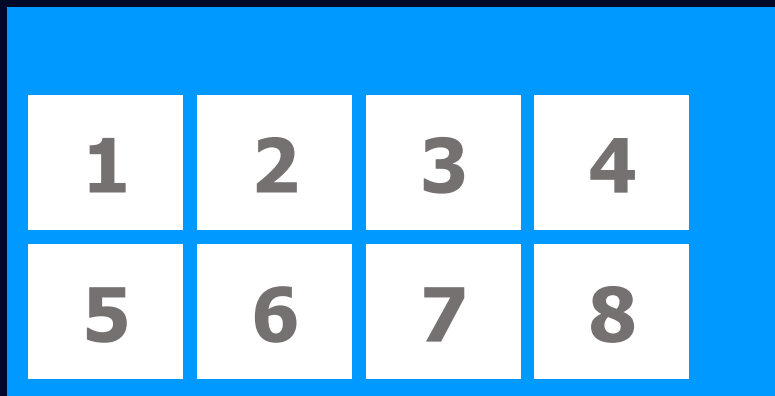
# flex-end

Agrupar los ítems **al final del eje secundario**.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 400px;  
  flex-wrap: wrap;  
  align-content: flex-start;  
}
```



Propiedades • Contenedores flex

# Centrado perfecto

---

Estableciendo las propiedades **justify-content** y **align-items** en **center**, hace que el elemento flexible estará perfectamente centrado.

w3schools/CSS FlexBox Container



```
.flex-container {  
  display: flex;  
  height: 300px;  
  justify-content: center;  
  align-items: center;  
}
```

1

## 03 | ITEMS

Propiedades • Elementos flex

# flex-basis

Especifica la **longitud inicial de un elemento flexible**, antes de aplicar una variación.

w3schools/CSS FlexBox Items



```
div .c-2 {  
  display: flex;  
  flex-basis: 200px;  
}
```



# flex-grow

Especifica **cuánto crecerá** un elemento flexible **en relación con el resto** de los elementos flexibles (en el caso de que no tengan un ancho específico).



```
div .c-1 {
```

```
  display: flex;
```

```
  flex-grow: 1;
```

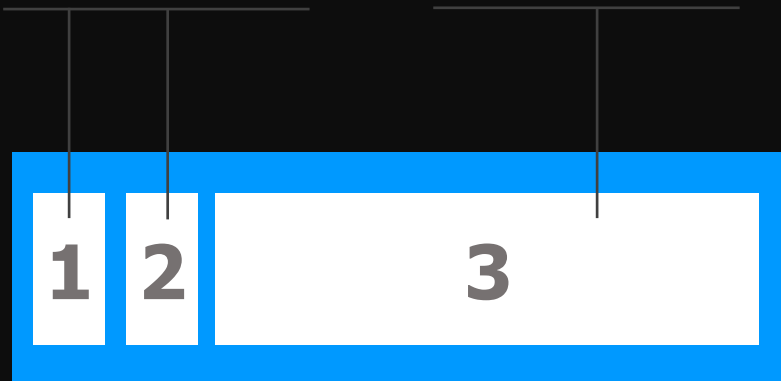
```
}
```

```
div .c-2 {
```

```
  display: flex;
```

```
  flex-grow: 8;
```

```
}
```



# order

Especifica el **orden de aparición** de los elementos flexibles. El primer elemento flexible en el código no tiene que aparecer como el primer elemento en el diseño.



```
div .c-1 {
```

```
  display: flex;
```

```
  order: 1;
```

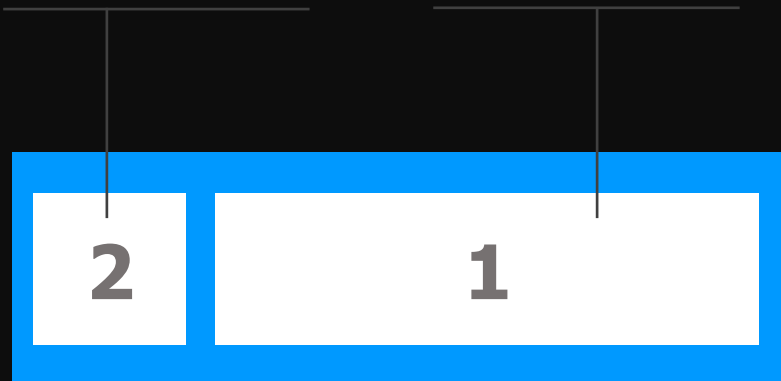
```
}
```

```
div .c-2 {
```

```
  display: flex;
```

```
  order: 2;
```

```
}
```



# **CONCLUSIONES**



1

## **Box model = CSS**

**Esquema para todo elemento del website**





2

## Box model

margin, border, padding



**3**

## **Shorthand**

**Ahorro de código, mayor eficacia**



4

## **Flexbox = responsive**

**Flujo de elementos de acuerdo al espacio disponible en dispositivo**





5

**Contenedor = elemento padre**  
Estilos CSS influye en estilos hijo



6

**Ítem = elemento hijo**

**Estilos CSS influidos por elemento padre**

**ISIL/**