

EJERCICIO 09

Ejercicio 1: Alineación básica con `justify-content` y `align-items`

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
</div>

/* CSS */
.container {
  display: flex;
  justify-content: space-around;
  align-items: center;
  height: 200px;
  border: 1px solid black;
}

.box {
  width: 50px;
  height: 50px;
  background-color: lightblue;
}
```

Instrucción para el alumno: Dibuja cómo se distribuyen los tres cuadros (`box`) dentro del contenedor (`container`) y explica la diferencia en el espacio alrededor de cada cuadro.

Ejercicio 2: Dirección y alineación con **flex-direction** y **align-items**

```
<!-- HTML -->
<div class="container">
  <div class="box">A</div>
  <div class="box">B</div>
  <div class="box">C</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-direction: column;
  align-items: flex-end;
  height: 200px;
  border: 1px solid black;
}

.box {
  width: 60px;
  height: 40px;
  background-color: lightcoral;
}
```

Instrucción para el alumno: Dibuja cómo se muestran los cuadros A, B y C dentro del contenedor, uno debajo del otro, alineados a la derecha.

Ejercicio 3: Distribución en múltiples líneas con **flex-wrap**

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-wrap: wrap;
  width: 150px;
  border: 1px solid black;
}

.box {
  width: 60px;
  height: 60px;
  background-color: lightgreen;
  margin: 5px;
}
```

Instrucción para el alumno: Dibuja cómo los cuadros se distribuyen en varias filas. ¿Cuántas filas se crean?

Ejercicio 4: Usar **flex-grow** para expandir elementos

```
<!-- HTML -->
<div class="container">
  <div class="box">A</div>
  <div class="box">B</div>
  <div class="box">C</div>
</div>

/* CSS */
.container {
  display: flex;
  border: 1px solid black;
}

.box {
  height: 50px;
  background-color: lightyellow;
}

.box:nth-child(1) { flex-grow: 1; }
.box:nth-child(2) { flex-grow: 2; }
.box:nth-child(3) { flex-grow: 1; }
```

Instrucción para el alumno: Dibuja los cuadros en el contenedor y muestra cómo se distribuye el espacio según el valor de **flex-grow**.

Ejercicio 5: Cuadrícula flexible con `align-content`

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-wrap: wrap;
  align-content: space-between;
  height: 300px;
  border: 1px solid black;
}

.box {
  width: 80px;
  height: 80px;
  background-color: lightpink;
}
```

Instrucción para el alumno: Dibuja cómo se distribuyen los cuadros en filas dentro del contenedor. Asegúrate de representar el espacio entre filas según el valor de `align-content`.

Ejercicio 6: Cambiar el orden de elementos con **order**

```
<!-- HTML -->
<div class="container">
  <div class="box">A</div>
  <div class="box">B</div>
  <div class="box">C</div>
</div>

/* CSS */
.container {
  display: flex;
  border: 1px solid black;
}

.box {
  width: 50px;
  height: 50px;
  background-color: lightblue;
}

.box:nth-child(1) { order: 3; }
.box:nth-child(2) { order: 1; }
.box:nth-child(3) { order: 2; }
```

Instrucción para el alumno: Dibuja la disposición de los cuadros en el contenedor teniendo en cuenta el orden indicado en el CSS.

Ejercicio 7: Distribuir espacio con **space-between** y **space-around**

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
</div>

/* CSS */
.container {
  display: flex;
```

```
    justify-content: space-between;
    align-items: center;
    height: 150px;
    border: 1px solid black;
}

.box {
    width: 40px;
    height: 40px;
    background-color: lightcoral;
}
```

Instrucción para el alumno: Dibuja la distribución de los tres elementos cuando están espaciados con `space-between`.

Ejercicio 8: Alineación de elementos con `align-self`

```
<!-- HTML -->
<div class="container">
  <div class="box">A</div>
  <div class="box">B</div>
  <div class="box">C</div>
</div>

/* CSS */
.container {
  display: flex;
  align-items: center;
  height: 200px;
  border: 1px solid black;
}

.box {
  width: 50px;
  height: 50px;
  background-color: lightblue;
}

.box:nth-child(2) { align-self: flex-start; }
.box:nth-child(3) { align-self: flex-end; }
```

Instrucción para el alumno: Dibuja los cuadros dentro del contenedor mostrando cómo cada uno se alinea según su propiedad `align-self`.

Ejercicio 9: Uso de **flex-basis** y **flex-grow**

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
</div>

/* CSS */
.container {
  display: flex;
  border: 1px solid black;
}

.box {
  flex-basis: 100px;
  flex-grow: 1;
  background-color: lightgreen;
}
```

Instrucción para el alumno: Dibuja la distribución de los elementos dentro del contenedor y observa cómo **flex-grow** ajusta el tamaño según el espacio disponible.

Ejercicio 10: Flexibilidad con `flex-shrink`

```
<!-- HTML -->
<div class="container">
  <div class="box">A</div>
  <div class="box">B</div>
  <div class="box">C</div>
</div>

/* CSS */
.container {
  display: flex;
  width: 150px;
  border: 1px solid black;
}

.box {
  width: 100px;
  height: 50px;
  background-color: lightyellow;
}

.box:nth-child(1) { flex-shrink: 0; }
.box:nth-child(2) { flex-shrink: 1; }
.box:nth-child(3) { flex-shrink: 2; }
```

Instrucción para el alumno: Dibuja cómo se ajustan los tamaños de los cuadros en el contenedor según el valor de `flex-shrink`.

Ejercicio 11: flex-direction y justify-content

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-direction: column;
  justify-content: space-around;
  height: 300px;
  border: 1px solid black;
}

.box {
  width: 60px;
  height: 60px;
  background-color: lightpink;
}
```

Instrucción para el alumno: Dibuja cómo se distribuyen los cuadros en el contenedor en una columna espaciada con `space-around`.

Ejercicio 12: Diferentes alineaciones en múltiples líneas

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-wrap: wrap;
  align-content: space-around;
  height: 250px;
  border: 1px solid black;
}

.box {
  width: 70px;
  height: 70px;
  background-color: lightcyan;
}
```

Instrucción para el alumno: Dibuja la distribución de los elementos en filas dentro del contenedor y cómo se distribuyen en el espacio usando `align-content: space-around`.

Ejercicio 13: Combinación de **order** y **align-self**

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
</div>

/* CSS */
.container {
  display: flex;
  height: 150px;
  border: 1px solid black;
}

.box {
  width: 60px;
  height: 60px;
  background-color: lightgray;
}

.box:nth-child(1) { order: 3; align-self: flex-end; }
.box:nth-child(2) { order: 1; align-self: center; }
.box:nth-child(3) { order: 2; align-self: flex-start; }
```

Instrucción para el alumno: Dibuja cómo se distribuyen y alinean los cuadros en el contenedor según el **order** y **align-self**.

Ejercicio 14: Distribución con **flex-flow**

```
<!-- HTML -->
<div class="container">
  <div class="box">A</div>
  <div class="box">B</div>
  <div class="box">C</div>
  <div class="box">D</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-flow: row wrap;
  width: 200px;
  height: 200px;
  border: 1px solid black;
}

.box {
  width: 100px;
  height: 100px;
  background-color: lightsteelblue;
}
```

Instrucción para el alumno: Dibuja la distribución de los cuadros y explica cómo **flex-flow** permite una combinación de **flex-direction** y **flex-wrap**.

Ejercicio 15: Alineación compleja con `align-content` y `justify-content`

```
<!-- HTML -->
<div class="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
</div>

/* CSS */
.container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  align-content: space-evenly;
  height: 300px;
  border: 1px solid black;
}

.box {
  width: 60px;
  height: 60px;
  background-color: lightcoral;
  margin: 5px;
}
```

Instrucción para el alumno: Dibuja cómo se distribuyen los cuadros en el contenedor y muestra el efecto de `justify-content: center` y `align-content: space-evenly` en la organización del espacio.