



Flexbox



Flexbox es una tecnología de diseño en CSS que se utiliza para crear diseños flexibles y responsivos. Con Flexbox, los elementos dentro de un contenedor pueden ser alineados y distribuidos de manera eficiente, independientemente del tamaño del contenedor o de la cantidad de contenido que contenga. Flexbox también permite la gestión de dimensiones automáticas y la optimización de la distribución del espacio, lo que facilita el diseño de elementos adaptables y escalables en diferentes dispositivos y tamaños de pantalla. En resumen, Flexbox es una herramienta poderosa para los diseñadores web que les permite crear diseños flexibles, responsivos y de fácil mantenimiento.

Para comenzar a usar el modelo Flexbox, primero debe definir un contenedor flexible.

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-
scale=1.0">

  <title>Document</title>

  <style>

    .flex-container {
      background-color: #ddd;
    }

    .flex-container>div {
      background-color: #333;
      margin: 10px;

      padding: 20px;
      font-size: 30px;
      color: #ccc;
      width: 150px;
      height: 150px;
    }

  </style>

</head>
```

Elemento padre (container)

Como hemos visto, este es un contenedor flexible (el área lightgray) con tres elementos flexibles (darkgray):

```
<div class="flex-container">

  <div>1</div>

  <div>2</div>

  <div>3</div>

</div>
```



El contenedor flexible se vuelve flexible al establecer la propiedad **display** en flex:

```
.flex-container {
  display: flex;
  background-color: #ddd;
}
```



Las propiedades del contenedor flexible son:

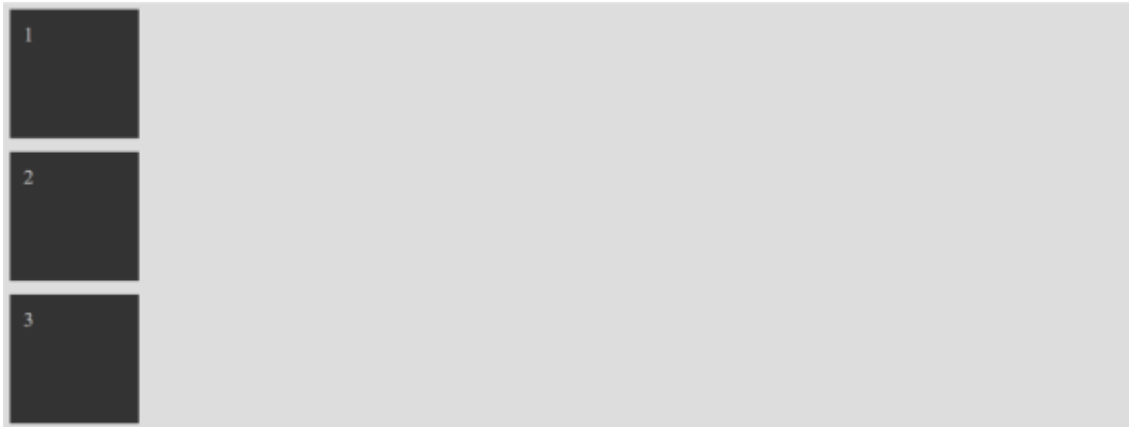
- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

La propiedad de dirección flexible

La propiedad `flex-direction` define en qué dirección el contenedor quiere apilar los elementos flexibles.

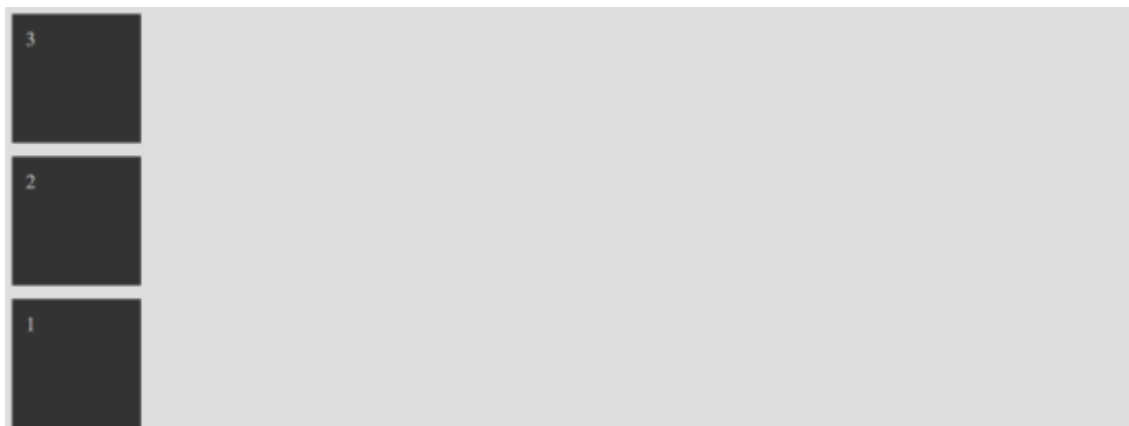
El valor `column` apila los elementos flexibles verticalmente (de arriba a abajo):

```
.flex-container {  
    display: flex;  
    flex-direction: column;  
    background-color: #ddd;  
}
```



El valor `column-reverse` apila los elementos flexibles verticalmente (pero de abajo hacia arriba):

```
.flex-container {  
    display: flex;  
    flex-direction: column-reverse;  
    background-color: #ddd;  
}
```



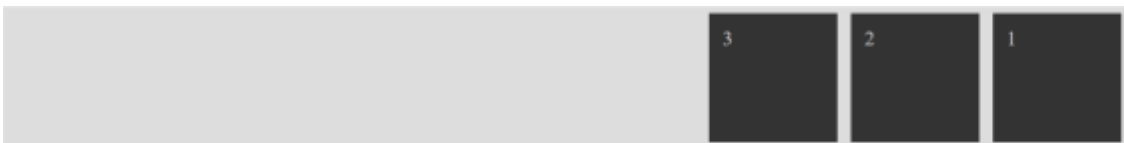
El valor row apila los elementos flexibles horizontalmente (de izquierda a derecha):

```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    background-color: #ddd;  
}
```



El valor row-reverse apila los elementos flexibles horizontalmente (pero de derecha a izquierda):

```
.flex-container {  
    display: flex;  
    flex-direction: row-reverse;  
    background-color: #ddd;  
}
```



La propiedad de envoltura flexible

La flex-wrap especifica si los elementos flexibles deben ajustarse o no.

Los ejemplos a continuación tienen 15 elementos flexibles para demostrar mejor la flex-wrap .

```
<div class="flex-container">

  <div>1</div>

  <div>2</div>

  <div>3</div>

  <div>4</div>

  <div>5</div>

  <div>6</div>

  <div>7</div>

  <div>8</div>

  <div>9</div>

  <div>10</div>

  <div>11</div>

  <div>12</div>

  <div>13</div>

  <div>14</div>

  <div>15</div>

</div>
```

El wrap especifica que los elementos flexibles se ajustarán si es necesario:

```
.flex-container {

  display: flex;
  flex-wrap: wrap;

  background-color: #ddd;

}
```



El valor nowrap especifica que los elementos flexibles no se ajustarán (esto es predeterminado)

```
.flex-container {  
    display: flex;  
    flex-wrap: nowrap;  
    background-color: #ddd;  
}
```



El wrap-reverse especifica que los elementos flexibles se ajustarán si es necesario, en orden inverso:

```
.flex-container {  
    display: flex;  
    flex-wrap: wrap-reverse;  
    background-color: #ddd;  
}
```



La propiedad de flujo flexible

La flex-flow es una propiedad abreviada para configurar las propiedades flex-direction y flex-wrap.

```
.flex-container {  
    display: flex;  
    flex-flow: row wrap;  
    background-color: #ddd;  
}
```



La propiedad de justificar el contenido

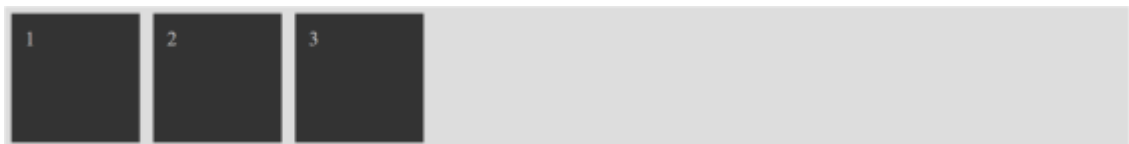
La `justify-content` se utiliza para alinear los elementos flexibles:

```
.flex-container {  
    display: flex;  
    justify-content: center;  
    background-color: #ddd;  
}
```



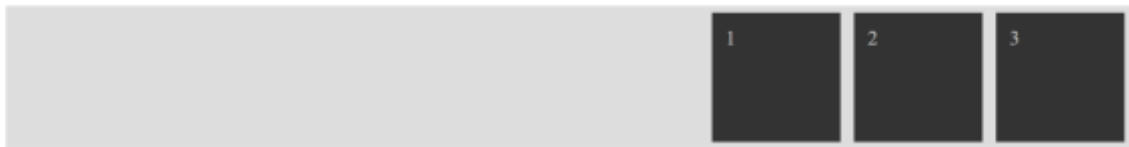
El `flex-start` alinea los elementos flexibles al comienzo del contenedor (esto es predeterminado):

```
.flex-container {  
    display: flex;  
    justify-content: flex-start;  
    background-color: #ddd;  
}
```



El `flex-end` alinea los elementos flexibles al final del contenedor:

```
.flex-container {  
    display: flex;  
    justify-content: flex-end;  
    background-color: #ddd;  
}
```



El space-around muestra los elementos flexibles con espacio antes, entre y después de las líneas:

```
.flex-container {  
    display: flex;  
    justify-content: space-around;  
    background-color: #ddd;  
}
```



El space-between muestra los elementos flexibles con espacio entre líneas:

```
.flex-container {  
    display: flex;  
    justify-content: space-between;  
    background-color: #ddd;  
}
```



La propiedad de elementos de alineación

La align-items se utiliza para alinear los elementos flexibles.

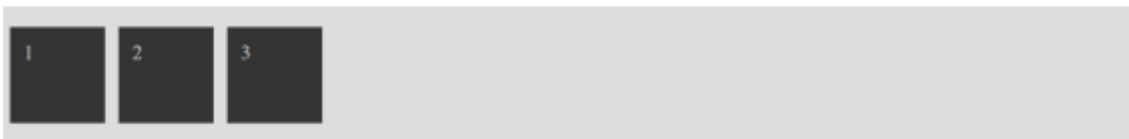
```
<style>

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

    background-color: #ddd;
  }

  .flex-container>div {
    background-color: #333;
    margin: 10px;

    padding: 20px;
    font-size: 30px;
    color: #ccc;
    width: 100px;
    height: 100px;
  }
</style>
```



El flex-start alinea los elementos flexibles en la parte superior del contenedor:

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

El flex-end alinea los elementos flexibles en la parte inferior del contenedor:

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

El stretch estira los elementos flexibles para llenar el contenedor (esto es predeterminado):

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

El baseline alinea los elementos flexibles como se alinean sus líneas base:

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



Nota: el ejemplo utiliza un tamaño de fuente diferente para demostrar que los elementos se alinean con la línea de base del texto:

```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {
  display: flex;
  height: 200px;

  align-items: baseline;
  background-color: #ddd;
}

.flex-container > div {
  background-color: #333;
  width: 100px;

  margin: 10px;

  text-align: center;
  line-height: 75px;
  font-size: 30px;
  color: #ddd;
}

</style>

</head>

<body>

<h1>The align-items Property</h1>

<p>The "align-items: baseline;" aligns the flex items such as their
baselines aligns:</p>

<div class="flex-container">

  <div><h1>1</h1></div>

  <div><h6>2</h6></div>

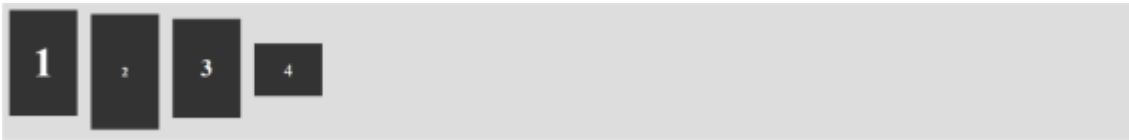
  <div><h3>3</h3></div>

  <div><small>4</small></div>

</div>

</body>

</html>
```



La propiedad de alineación de contenido

La align-content se utiliza para alinear las líneas flexibles.

```
<div class="flex-container">
```

```
<div>1</div>
```

```
<div>2</div>
```

```
<div>3</div>
```

```
<div>4</div>
```

```
<div>5</div>
```

```
<div>6</div>
```

```
<div>7</div>
```

```
<div>8</div>
```

```
<div>9</div>
```

```
<div>10</div>
```

```
<div>11</div>
```

```
<div>12</div>
```

```
<div>13</div>
```

```
<div>14</div>
```

```
<div>15</div>
```

```
</div>
```

En estos ejemplos, usamos un contenedor de 600 píxeles de alto, con la flex-wrap establecida en wrap, para demostrar mejor la align-content.

El space-between muestra las líneas flexibles con el mismo espacio entre ellas:

```
.flex-container {  
    display: flex;  
    height: 600px;  
    flex-wrap: wrap;  
  
    align-content: space-between;  
  
    background-color: #ddd;  
}
```



El space-around muestra las líneas flexibles con espacio antes, entre y después de ellas:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: space-around;  
}
```

El stretch estira las líneas flexibles para ocupar el espacio restante (esto es predeterminado):

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: stretch;  
}
```

Las centerpantallas de valor muestran las líneas flexibles en el medio del contenedor:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: center;  
}
```

El flex-start muestra las líneas flexibles al comienzo del contenedor:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: flex-start;  
}
```

El flex-end muestra las líneas flexibles al final del contenedor:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: flex-end;  
}
```


Centrado perfecto

En el siguiente ejemplo resolveremos un problema de estilo muy común: el centrado perfecto.

```
<div class="flex-container">

  <div>1</div>

</div>
```

```
<style>

  .flex-container {
    display: flex;
    height: 600px;

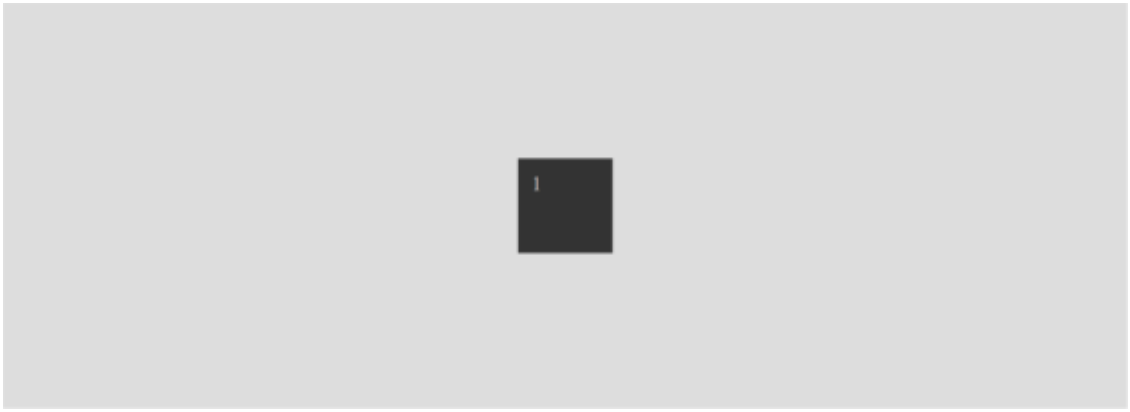
    justify-content: center;
    align-items: center;

    background-color: #ddd;
  }

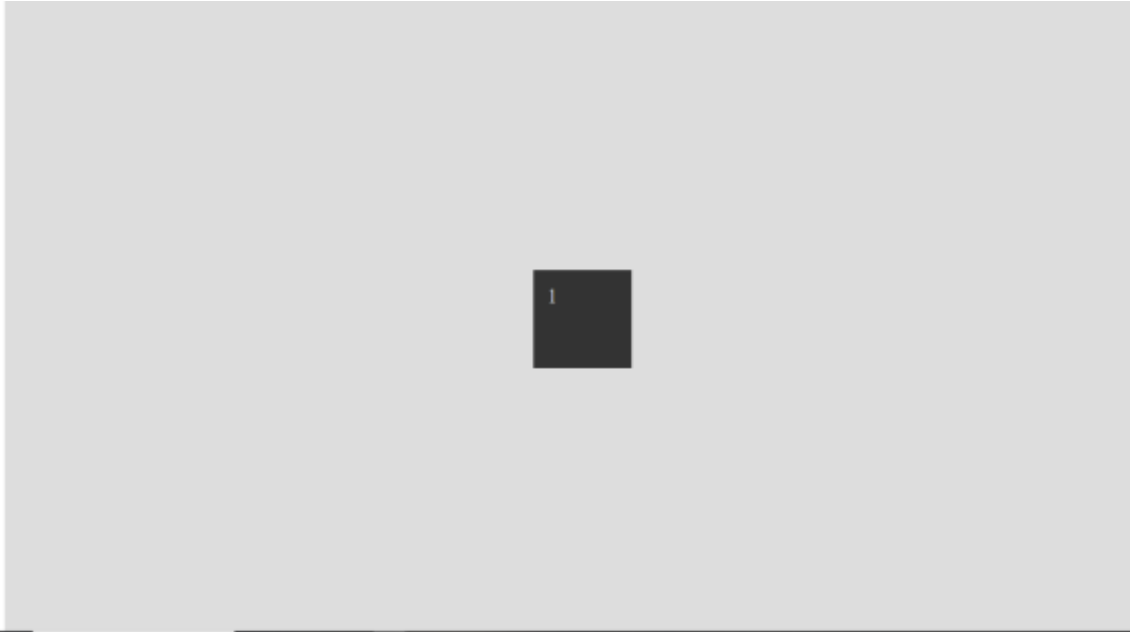
  .flex-container>div {
    background-color: #333;
    margin: 10px;

    padding: 20px;
    font-size: 30px;
    color: #ccc;
    width: 100px;
    height: 100px;
  }

</style>
```



```
.flex-container {  
    display: flex;  
    height: 100vh;  
    justify-content: center;  
  
    background-color: #ddd;  
}
```



Docs_ Generalidades Curiosidades