

## Flexbox

Existen dos tipos de contenedores flexibles:

de bloque y en-línea.

La propiedad `display` permite definir el tipo de contenedor flexible:

- `flex`: el contenedor se comporta como un bloque, es decir, ocupa todo el espacio horizontal disponible.
- `inline-flex`: el contenedor se comporta como un elemento en-línea (ocupa sólo el espacio horizontal necesario y pueden colocarse unos detrás de otros en la misma línea).

### Dirección de los elementos flexibles: **flex-direction**

La propiedad **flex-direction** define la dirección principal, es decir, la dirección en la que se colocan los elementos flexibles en los contenedores.

Los posibles valores de la propiedad `flex-direction` son:

- `row`: los elementos se colocan horizontalmente de izquierda a derecha.
- `row-reverse`: los elementos se colocan horizontalmente de derecha a izquierda
- `column`: los elementos se colocan verticalmente de arriba abajo
- `column-reverse`: los elementos se colocan verticalmente de abajo arriba.

### Ajuste de línea de los elementos flexibles: **flex-wrap**

La propiedad `flex-wrap` establece si los elementos se colocan en una sola línea o en varias.

Los posibles valores de la propiedad `flex-wrap` son:

- `no-wrap`: los elementos se colocan en una sola línea.
- `wrap`: los elementos se colocan en varias líneas si no caben en una sola línea
- `wrap-reverse`: los elementos se colocan en varias líneas si no caben en una sola línea, pero las líneas se muestran de abajo arriba

La propiedad compuesta `flex-flow`

La propiedad compuesta `flex-flow` permite establecer simultáneamente las dos propiedades `flex-direction` y `flex-wrap`.

Como no hay ambigüedad, el orden de los valores de `flex-direction` y `flex-wrap` no es importante

### Orden de los elementos flexibles: **order**

Los elementos flexibles se muestran en el mismo orden en el que se encuentran en el código fuente de la página web, pero la propiedad `order` permite cambiar ese orden. A la propiedad **order** se le pueden dar valores enteros positivos o negativos. El valor predeterminado de un elemento es 0.

En el ejemplo siguiente se ha dado la clase "primero" al cuarto párrafo, por lo que este se muestra en primer lugar.

Si varios elementos flexibles tienen el mismo valor de la propiedad `order`, se muestran en el mismo orden en que se encuentran en el código fuente.

En el ejemplo siguiente se ha dado la clase "primero" al segundo y al cuarto párrafo, por lo que estos se muestran en primer lugar. El segundo se muestra antes que el cuarto porque el segundo está antes en el código fuente de la página web.

### Tamaño inicial de los elementos flexibles: **flex-basis**.

El tamaño de los elementos flexibles viene determinado en principio por su contenido. Si hay espacio suficiente en la dirección principal, los elementos se ensanchan para mostrar todo su contenido en una sola línea:

```
div { display: flex; }
```

```
p { border: black 1px solid; margin: 2px; }
```

Si no hay espacio suficiente en la dirección principal, el espacio disponible se distribuye de forma desigual entre todos los elementos, dando más espacio a los elementos con más contenido, de manera que el tamaño en la dirección secundaria sea el menor posible:

```
div { display: flex; }
```

```
p { border: black 1px solid; margin: 2px; }
```

La propiedad flex-basis establece el tamaño inicial del elemento, antes de que se reparta en su caso el espacio libre.

Los posibles valores de la propiedad flex-basis son:

- content: el tamaño inicial viene determinado por el contenido del elemento.
- auto: el tamaño inicial viene determinado por las propiedades width y/o height.
- tamaño: indica el tamaño inicial exacto.

### Factor de expansión de los elementos flexibles: **flex-grow**

La propiedad flex-grow hace que los elementos crezcan hasta ocupar todo el espacio disponible en la dirección principal.

```
div { display: flex; }
```

```
p { flex-grow: 1; border: black 1px solid; margin: 2px; }
```

La propiedad flex-grow toma valores enteros que indican la proporción en que se reparte el espacio.

Si todos los elementos toman el mismo valor, el espacio se reparte a partes iguales.

Si hay valores distintos, el espacio se reparte de forma proporcional a esos valores

### Factor de compresión de los elementos flexibles: **flex-shrink**

La propiedad flex-shrink, que toma valores enteros, indica la proporción en que se estrechan los elementos cuando su contenido no cabe en el espacio disponible en la dirección principal.

```
div { display: flex; }
```

```
p { flex-shrink: 1; border: black 1px solid; margin: 2px; }
```

Si todos los elementos toman el mismo valor, el espacio se reparte a partes iguales.

Si hay valores distintos, la reducción es proporcional a esos valores (cuanto mayor es el valor, más se reduce el elemento).

### La propiedad compuesta **flex**

La propiedad compuesta flex permite establecer simultáneamente las tres propiedades flex-grow, flex-shrink y flex-basis. Como puede haber ambigüedad, los valores se interpretan en ese orden.

```
div { display: flex; }
```

```
p { flex: initial; border: black 1px solid; margin: 2px; }
```

Se recomienda utilizar la propiedad compuesta flex en vez de las propiedades específicas flex-grow, flex-shrink y flex-basis, para asegurar que se definen las tres propiedades.

La propiedad compuesta flex admite cuatro valores básicos, que corresponden a los usos más comunes de las cajas flexibles:

- **initial:** El valor flex: initial es equivalente a flex: 0 1 auto. Así, el tamaño inicial de los elementos depende de su contenido (o de las propiedades width y height que se puedan aplicar a los elementos), los elementos no crecen si hay espacio libre, pero se estrechan si no hay espacio suficiente, de manera que el tamaño vertical final sea lo más reducido posible.
- **Auto:** El valor flex: auto es equivalente a flex: 1 1 auto. Así, el tamaño inicial de los elementos depende de su contenido (o de las propiedades width y height que se puedan aplicar a los elementos), pero los elementos crecen de manera uniforme si hay espacio libre y se estrechan si no hay espacio suficiente, de manera que el tamaño vertical final sea lo más reducido posible.
- **None:** El valor flex: none es equivalente a flex: 0 0 auto. Así, el tamaño inicial de los elementos depende de su contenido (o de las propiedades width y height que se puedan aplicar a los elementos), y los elementos ni crecen si hay espacio libre ni se estrechan si no hay espacio suficiente.
- **Número:** El valor flex: número es equivalente a flex: número 1 0. Así, el tamaño inicial de los elementos es proporcional al número indicado, y los elementos crecen de forma uniforme si hay espacio libre y se estrechan de forma uniforme si no hay espacio suficiente.

### Márgenes automáticos en los elementos flexibles: margin: **auto**

Con la propiedad margin con el valor auto, se asigna todo el espacio disponible al lado correspondiente.

En el ejemplo siguiente se asigna la clase "derecha" al cuarto párrafo, por lo que el espacio disponible se asigna a su lado izquierdo y el cuarto párrafo queda completamente alineado a la derecha:

```
div { display: flex; }
```

```
p { border: black 1px solid; margin: 2px; }
```

```
p.derecha { margin-left: auto; }
```

### Alineación en la dirección principal: **justify-content**

La propiedad justify-content establece la forma en que se reparte el espacio libre disponible en la dirección principal.

Los posibles valores de la propiedad justify-content son:

- flex-start: los elementos se sitúan al principio de la dirección principal, es decir, el espacio disponible se sitúa al final

```
div { display: flex; justify-content: flex-start; }
```

```
p { border: black 1px solid; margin: 2px; }
```

- flex-end: los elementos se sitúan al final de la dirección principal, es decir, el espacio disponible se sitúa al principio.
- center: los elementos se sitúan en el centro de la dirección principal, es decir, el espacio disponible se reparte entre el principio y el final
- space-between: el espacio disponible se reparte entre los elementos
- space-around: el espacio disponible se reparte entre los elementos y en los extremos.

#### Alineación en la dirección secundaria (una línea): align-items

La propiedad align-items establece la forma en que se alinean los elementos en la dirección secundaria.

Los posibles valores de la propiedad align-items son:

- flex-start: los elementos se sitúan al principio de la dirección secundaria (es decir, arriba si la dirección secundaria es vertical).

```
div { display: flex; align-items: flex-start; }
```

```
p { border: black 1px solid; margin: 2px; }
```

- flex-end: los elementos se sitúan al final de la dirección secundaria (es decir, abajo si la dirección secundaria es vertical)
- center: los elementos se sitúan en el centro de la dirección secundaria (es decir, en medio si la dirección secundaria es vertical).
- stretch: todos los elementos toman el mismo tamaño en la dirección secundaria.
- baseline: los elementos se alinean en la dirección secundaria según la primera línea de texto. Para distinguirlo del valor flex-start, en el último párrafo del ejemplo siguiente se ha aumentado el tamaño del texto.

#### Alineación individual en la dirección secundaria: align-self

La propiedad align-self permite que un elemento tenga una alineación en la dirección secundaria distinta de la establecida con align-items.

Los posibles valores de la propiedad align-self son los mismos que los de la propiedad align-items:

- flex-start: el elemento se sitúa al principio de la dirección secundaria (es decir, arriba si la dirección secundaria es vertical).

```
div { display: flex; align-items: flex-start; }
```

```
p { border: black 1px solid; margin: 2px; }
```

- flex-end: el elemento se sitúa al final de la dirección secundaria (es decir, abajo si la dirección secundaria es vertical).
- center: el elemento se sitúa en el centro de la dirección secundaria (es decir, en medio si la dirección secundaria es vertical).
- stretch: el elemento toma el tamaño del mayor elemento en la dirección secundaria
- baseline: los elementos se alinean en la dirección secundaria según la primera línea de texto. Para distinguirlo del valor flex-start, la propiedad se ha aplicado a dos párrafos y en uno de ellos se ha aumentado el tamaño del texto.

#### Alineación en la dirección secundaria (varias líneas): **align-content**

La propiedad align-content establece la manera en que se reparte el espacio sobrante en la dirección secundaria.

Normalmente, el espacio ocupado en la dirección secundaria por un elemento flexible depende de su contenido, por lo que no suele haber espacio sobrante. Por ello, en los ejemplos siguientes el espacio disponible en la dirección secundaria se ha establecido con la propiedad height, de manera que haya espacio sobrante.

Los posibles valores de la propiedad align-content son:

- flex-start: los elementos se sitúan al principio de la dirección secundaria, es decir, el espacio disponible se sitúa al final.

```
div { display: flex; align-items: flex-end; }
```

```
p { border: black 1px solid; margin: 2px; }
```

```
p.especial { align-self: flex-start; border-color: blue; }
```

- flex-end: los elementos se sitúan al final de la dirección secundaria, es decir, el espacio disponible se sitúa al principio
- center: los elementos se sitúan en el centro de la dirección secundaria, es decir, el espacio disponible se reparte entre el principio y el final
- space-between: el espacio disponible se reparte entre los elementos
- space-around: el espacio disponible se reparte entre los elementos y en los extremos
- stretch: los elementos aumentan su tamaño en la dirección secundaria de manera que ocupan todo el espacio disponible