

Flexible Box

Capítulo 14

Flexible Box Layout. Introducción

- La especificación oficial,
<https://www.w3.org/TR/css-flexbox-1/> (link a especificación más actual)
<https://www.w3.org/TR/2018/CR-css-flexbox-1-20181119/> (la actual)
nos dice:

The specification describes a CSS box model optimized for user interface design. In the flex layout model, the children of a flex container can be laid out in any direction, and can “flex” their sizes, either growing to fill unused space or shrinking to avoid overflowing the parent. Both horizontal and vertical alignment of the children can be easily manipulated. Nesting of these boxes (horizontal inside vertical or vertical inside horizontal) can be used to build layouts in two dimensions. CSS is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, etc.

Flexible Box Layout. Introducción

CSS 2.1 definió cuatro modos de diseño o *layout* (entiendiendo por tal, algoritmos que determinan el tamaño y la posición de las cajas en función de sus relaciones con sus cajas hermanas y padres antecesoras):

- Diseño de bloques, diseñado para diseñar o maquetar documentos.
- Diseño en línea, diseñado para diseñar o maquetar texto
- Diseño de tabla, diseñado para diseñar o maquetar datos 2D en un formato tabular.
- Diseño posicionado, diseñado para un posicionamiento explícito sin tener en cuenta otros elementos en el documento.

Flexible Box Layout. Visión general

Este módulo de CSS introduce un nuevo modo de diseño, diseño flexible, que está diseñado para diseñar aplicaciones y páginas web más complejas.

El diseño flexible es superficialmente similar al diseño de bloques. Carece de muchas de las propiedades más complejas centradas en texto o documento que se pueden utilizar en el diseño de bloques, como *float* y *columns*. A cambio, obtiene herramientas sencillas y eficaces para distribuir espacio y alinear contenido de maneras que las aplicaciones web y las páginas web complejas a menudo necesitan.


Flexible Box Layout. Visión general

El contenido de un contenedor flexible:

- se puede colocar en cualquier dirección de flujo (hacia la izquierda, hacia la derecha, hacia abajo, ¡o incluso hacia arriba!)
- puede tener su orden de visualización invertido o reorganizado en la capa de estilo (es decir, el orden visual puede ser independiente del origen y el orden de voz)
- puede colocarse linealmente a lo largo de un solo eje (principal) o desarrollarse en varias líneas a lo largo de otro eje secundario (cruz)
- pueden "flexibilizar" sus tamaños para responder al espacio disponible
pueden alinearse con respecto a su contenedor o entre sí en el secundario (cruz)

Flexible Box Layout. La compatibilidad de los navegadores

- En el sitio Can I Use, podemos ver que este módulo lo reconocen todos los navegadores modernos, excepto Microsoft Internet Explorer v11. Por lo tanto, muchos sitios web ya lo están utilizando.

Flexible Box Layout Module  - CR

Global 82.33% + 14.72% = 97.05%
unprefixed: 81.46% + 4.44% = 85.91%

Method of positioning elements in horizontal or vertical stacks.
Support includes the support for the all properties prefixed with **flex** as well as **display: flex**, **display: inline-flex**, **align-content**, **align-items**, **align-self**, **justify-content** and **order**.

[Current aligned](#) [Usage relative](#) [Show all](#)

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			49						
		47	51					4.4	
8	13	48	52	9.1		9.3		4.4.4	
11	14	49	53	10	40	10	all	52	53
		50	54	TP	41				
		51	55		42				
		52	56						

[Notes](#) [Known issues \(9\)](#) [Resources \(12\)](#) [Feedback](#)

Most partial support refers to supporting an **older version** of the specification or an **older syntax**.

⁴ Partial support is due to large amount of bugs present (see known issues)

Flexible Box Layout. Principio de uso



- El principio del uso de cajas flexibles es el siguiente: basta con definir un contenedor primario como caja flexible para que todos sus elementos, incluidos los elementos secundarios, adopten su comportamiento. Luego, para cada caja secundaria, podemos determinar las propiedades de visualización deseadas.
- El elemento estructural básico es un contenedor que proporciona el contexto de caja flexible (*flex-container* en inglés).
- Todos los elementos secundarios de este contenedor principal heredan de este contexto y serán elementos flexibles (*flex-items* en inglés).

Flexible Box Layout. Principio de uso

Tendremos:

- Un contenedor principal flexible proporcionado por la propiedad **display: flex**.
- Todos los elementos secundarios directos que serán elementos flexibles.
- A continuación, podremos gestionar la distribución de los elementos secundarios, gestionar las alineaciones y el centrado, optar por reorganizar la presentación de los elementos y gestionar los espacios.

Flexible Box Layout. Contexto flexible

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Flexbox</title>
  <meta charset="UTF-8" />
  <style>
    #caja-flexible {
      display: flex;
      border: solid 1px #000;
    }
  </style>
</head>
<body>
  <div id="caja-flexible">
    <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.</p>
    <p>Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.</p>
    <p>Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.</p>
  </div>
</body>
</html>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Flexible Box Layout. Contexto flexible

Los párrafos secundarios, y por tanto los elementos flexibles, se adaptan automáticamente a la anchura de la ventana del navegador. Tenemos un diseño flexible.

La propiedad **display: flex** actúa sobre el contenedor como una presentación de tipo **block**, como con las técnicas de diseño «tradicionales».

La propiedad **display: inline-flex** actúa sobre el contenedor como una presentación de tipo **inline**. Es decir, el valor inline-flex hace que el contenedor flexible sea un elemento "en línea" dentro del elemento "padre" al que pertenezca.

Flexible Box Layout. Distribución de los elementos flexibles

- Los elementos flexibles se distribuyen horizontalmente en el contenedor --> los párrafos aparecen unos al lado de otros.
- Con la propiedad **flex-direction** podemos cambiarlo:
- **row**: los elementos se distribuyen en fila, es el valor por defecto.
- **row-reverse**: misma distribución, pero los elementos se muestran en el sentido inverso de su declaración en el código.
- **column**: los elementos se distribuyen en columnas, unos debajo de otros.
- **column-reverse**: misma distribución en columnas, pero los elementos se muestran en el sentido inverso de su declaración en el código.



Flexible Box Layout. Distribución de los elementos flexibles

```
<meta charset="UTF-8" />
<style>
  #caja-flexible {
    display: flex;
    flex-direction: column;
    border: solid 1px #000;
  }
</style>
</head>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Flexible Box Layout. Cambio de fila.

Propiedad flex-wrap



Podemos decidir qué elementos flexibles se deben trasladar cuando no hay suficiente espacio en el contenedor.

La propiedad **flex-wrap**, declarada para el contenedor flexible, es la que determina el comportamiento de los elementos flexibles:

- **nowrap**: los elementos no pasan a la siguiente fila y se reduce su anchura para mostrarlos. Este es el valor por defecto.
- **wrap**: los elementos pasan a la siguiente fila y conservan su anchura.
- **wrap-reverse**: los elementos pasan a la siguiente fila, pero en sentido inverso al de su declaración.

Flexible Box Layout. Cambio de fila.

Propiedad flex-wrap

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Flexbox</title>
  <meta charset="UTF-8" />
  <style>
    #caja-flexible {
      display: flex;
      flex-direction: row;
      flex-wrap: wrap;
      border: solid 1px #000;
    }
    p {
      width: 300px;
    }
  </style>
</head>
<body>
  <div id="caja-flexible">
    <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.</p>
    <p>Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.</p>
    <p>Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.</p>
    <p>Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Maecenas sed diam eget risus varius blandit sit amet non magna.</p>
    <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Sed posuere consectetur est at lobortis.</p>
  </div>
</body>
</html>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Duis mollis, est non commodo luctus, nisi erat Vivamus sagittis lacus vel augue laoreet porttitor ligula, eget lacinia odio sem nec elit. rutrum faucibus dolor auctor. Sed posuere Maecenas sed diam eget risus varius blandit consectetur est at lobortis. sit amet non magna.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Maecenas sed diam eget risus varius blandit sit amet non magna.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Sed posuere consectetur est at lobortis.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Maecenas sed diam eget risus varius blandit sit amet non magna.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Sed posuere consectetur est at lobortis.

Flexible Box Layout. Sintaxis corta

La propiedad CSS flex-flow es una propiedad atajo para las propiedades individuales flex-direction y flex-wrap.

flex-flow: row wrap equivale a **flex-direction: row** y **flex-wrap: wrap**

Otro ejemplo:

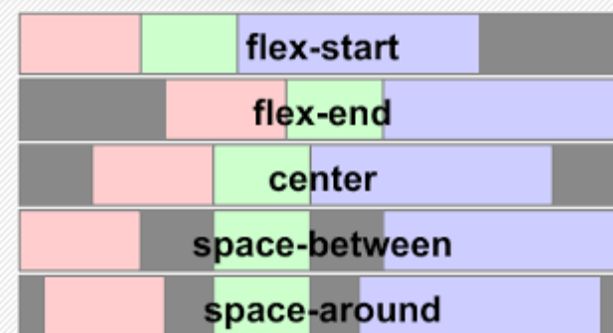
flex-flow: column-reverse wrap;

¿Qué haría esto?

Flexible Box Layout. Justificación horizontal en las filas

La propiedad `justify-content` es la que define esta justificación. Esta propiedad acepta cinco valores:

- **flex-start** permite el posicionamiento de los elementos a la izquierda del contenedor.
- **flex-end** permite el posicionamiento de los elementos a la derecha del contenedor.
- **center** permite el posicionamiento centrado de los elementos.
- **space-between** añade un espacio idéntico entre los elementos. El primer elemento está alineado a la izquierda del contenedor, y el último, a la derecha.
- **space-around** espacia de forma regular los elementos que no están alineados a la izquierda y a la derecha del contenedor.



Flexible Box Layout. Justificación vertical en las columnas

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Flexbox</title>
  <meta charset="UTF-8" />
  <style>
    #caja-flexible {
      display: flex;
      flex-direction: column;
      justify-content: space-around;
      border: solid 1px #000;
      height: 150px;
    }
    p {
      margin: 0;
      background-color: #eee;
    }
  </style>
</head>
<body>
<div id="caja-flexible">
  <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.</p>
  <p>Integer posuere erat a ante venenatis dapibus posuere velit aliquet.</p>
  <p>Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.</p>
  <p>Duis mollis, est non commodo luctus, nisi erat porttitor ligula.</p>
  <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.</p>
</div>
</body>
</html>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.

Duis mollis, est non commodo luctus, nisi erat porttitor ligula.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.

Flexible Box Layout. Justificación vertical en las columnas

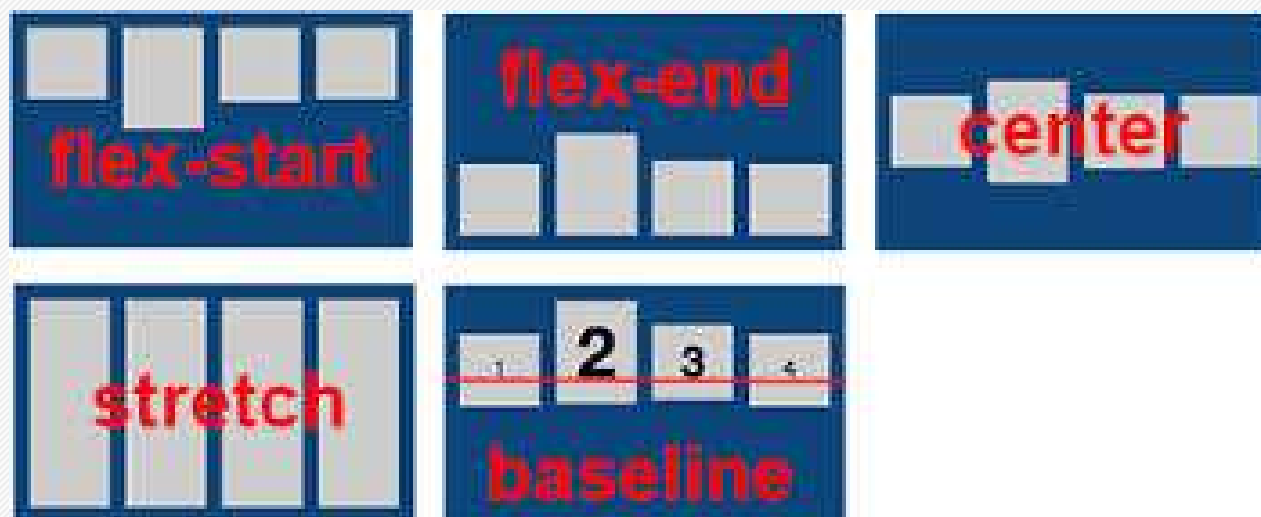
```
#caja-flexible {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  border: solid 1px #000;  
  height: 150px;  
}
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.
Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.
Duis mollis, est non commodo luctus, nisi erat porttitor ligula.
Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.

Flexible Box Layout. Alineación vertical en las filas

La propiedad **flex-direction: row** proporciona el eje principal horizontal.

La propiedad **align-items** actúa sobre el eje secundario que es el opuesto al eje principal; por lo tanto, al eje vertical.



Flexible Box Layout. Alineación vertical en las filas

La propiedad **flex-direction: row** acepta cinco valores:

- **stretch**: los elementos se amplían verticalmente para ocupar toda la altura disponible en el contenedor. Este es el valor por defecto.
- **flex-start**: los elementos se colocan en la parte superior del contenedor.
- **flex-end**: los elementos se colocan en la parte inferior del contenedor.
- **flex-center**: los elementos se colocan en el centro del contenedor.
- **baseline**: los elementos se alinean sobre la línea de base del texto.

Flexible Box Layout. Alineación vertical en las filas

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Flexbox</title>
  <meta charset="UTF-8" />
  <style>
    #caja-flexible {
      display: flex;
      flex-direction: row;
      align-items: center;
      border: solid 1px #000;
      height: 150px;
    }
    p {
      margin: 0;
      background-color: #eee;
    }
  </style>
</head>
<body>
  <div id="caja-flexible">
    <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus
dolor auctor.</p>
    <p>Integer posuere erat a ante venenatis dapibus posuere velit
aliquet.</p>
    <p>Donec sed odio dui. Cras mattis consectetur purus sit amet
fermentum.</p>
  </div>
</body>
</html>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.	Integer posuere erat a ante venenatis dapibus posuere velit aliquet.	Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.
------------------------------------------------------------------------	----------------------------------------------------------------------	-----------------------------------------------------------------------

Flexible Box Layout. Alineación vertical en las filas

```
yle>  
#caja-flexible {  
  display: flex;  
  flex-direction: row;  
  align-items: stretch;  
  border: solid 1px #000;  
  height: 150px;  
}  
p {  
  margin: 0;
```

Vivamus sagittis lacus vel augue laoreet Integer posuere erat a ante venenatis
rutrum faucibus dolor auctor. dapibus posuere velit aliquet. Donec sed odio dui. Cras mattis
consectetur purus sit amet fermentum.

Flexible Box Layout. Alineación vertical en las filas

```
>  
#caja-flexible {  
  display: flex;  
  flex-direction: row;  
  align-items: flex-start;  
  border: solid 1px #000;  
  height: 150px;  
}  
p {
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.	Integer posuere erat a ante venenatis dapibus posuere velit aliquet.	Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.
------------------------------------------------------------------------	----------------------------------------------------------------------	-----------------------------------------------------------------------

Flexible Box Layout. Alineación vertical en las filas

```
charset="UTF-8" />
<div>
  #caja-flexible {
    display: flex;
    flex-direction: row;
    align-items: flex-end;
    border: solid 1px #000;
    height: 150px;
  }
  p {
    margin: 0;
    background-color: #eee;
  }
</div>
```

Vivamus sagittis lacus vel augue laoreet Integer posuere erat a ante venenatis Donec sed odio dui. Cras mattis
rutrum faucibus dolor auctor. dapibus posuere velit aliquet. consectetur purus sit amet fermentum.

Flexible Box Layout. Alineación horizontal en las columnas

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Flexbox</title>
  <meta charset="UTF-8" />
  <style>
    #caja-flexible {
      display: flex;
      flex-direction: column;
      align-items: flex-start;
      border: solid 1px #000;
      height: 150px;
    }
    p {
      margin: 0;
      background-color: #eee;
    }
  </style>
</head>
<body>
<div id="caja-flexible">
  <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.</p>
  <p>Integer posuere erat a ante venenatis dapibus posuere velit aliquet.</p>
  <p>Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.</p>
</div>
</body>
</html>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.
Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.

Flexible Box Layout. Alineación horizontal en las columnas

```
<title>flexbox</title>
<meta charset="UTF-8" />
<style>
  #caja-flexible {
    display: flex;
    flex-direction: column;
    align-items: center;
    border: solid 1px #000;
    height: 150px;
  }
  p {
    margin: 0;
    background-color: #eee;
  }
</style>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.
Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.

Flexible Box Layout. Alineación horizontal en las columnas

```
e>  
#caja-flexible {  
  display: flex;  
  flex-direction: column;  
  align-items: flex-end;  
  border: solid 1px #000;  
  height: 150px;  
}
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.
Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Donec sed odio dui. Cras mattis consectetur purus sit amet fermentum.

Ejercicio

Crear mediante el uso de flexbox un catálogo de productos en el que veamos una serie de artículos, con una foto para cada uno, una descripción debajo y un botón que diga “compra”. La página deberá tener header y footer.