

A thick dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'LENGUAJES DE MARCAS'. In the bottom-left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

LENGUAJES DE MARCAS

POSICIONAMIENTO FLEXBOX

ÍNDICE

INTRODUCCIÓN A FLEXBOX	1
A. ¿QUÉ ES FLEXBOX?	1
B. CONTEXTO DE USO.....	1
ALGUNAS PROPIEDADES FLEX.....	4
A. FLEX-DIRECTION	4
B. ORDER.....	5
C. FLEX WRAP	6
D. JUSTIFY CONTENT.....	6
E. ALIGN ITEMS.....	7

INTRODUCCIÓN A FLEXBOX

A. ¿QUÉ ES FLEXBOX?

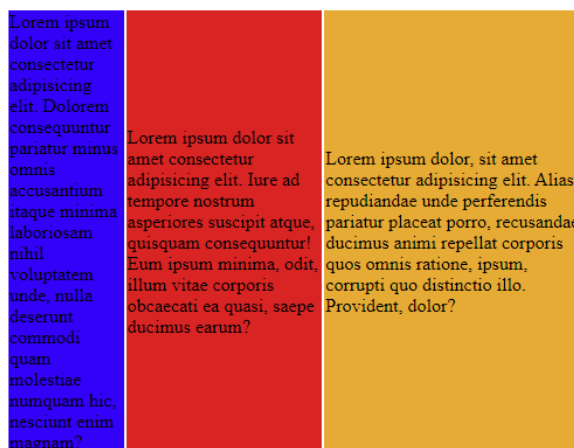
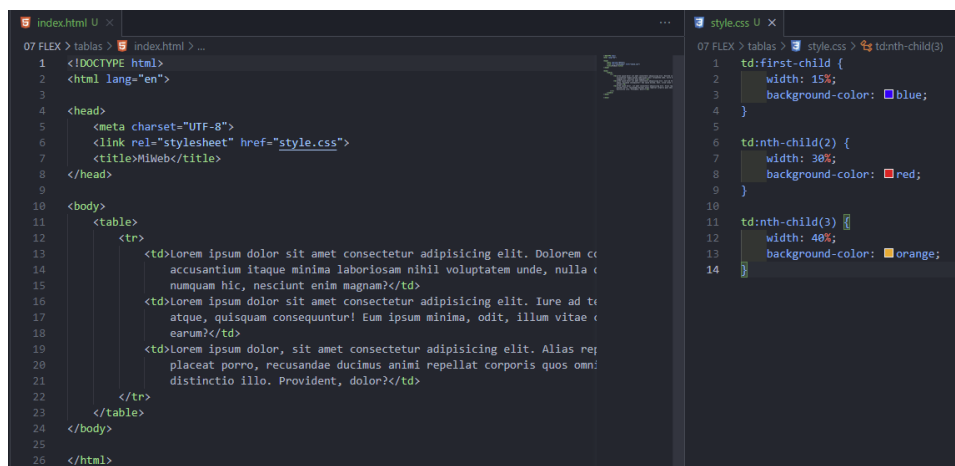
Flexbox es un modelo de diseño de CSS que permite la creación de diseños flexibles y responsivos en páginas web a través de un conjunto de propiedades y comportamientos que permiten distribuir y alinear elementos en un contenedor sin importar su tamaño o estructura.

Se puede controlar la dirección de los elementos, su orden, su alineación, su tamaño y la distribución del espacio disponible, lo que facilita la creación de diseños adaptables a diferentes dispositivos y pantallas.

B. CONTEXTO DE USO

El diseño y maquetación web se inspiró originalmente en los artículos de los periódicos y en las revistas.

Originalmente se utilizaba un diseño a través de una tabla para provocar ese diseño en forma de columnas y de espacio dentro de una página.



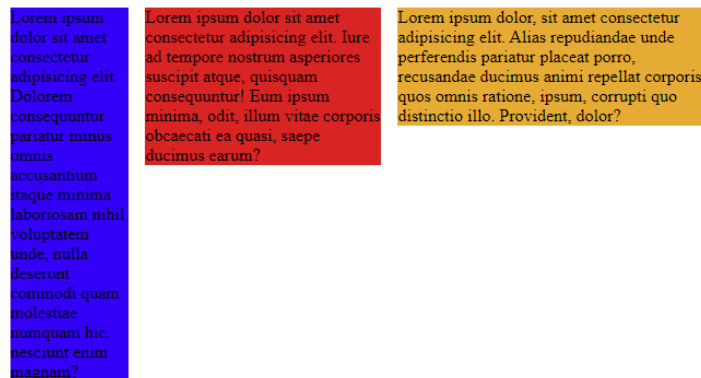
Pero esto obviamente trae muchos problemas y limitaciones de diseño. Por ello una vez que se dejaron de utilizar las tablas como elemento de diseño, se pasó a utilizar el diseño flotante a través del uso de contenedores (div, section, article...) y de la propiedad CSS float.

```

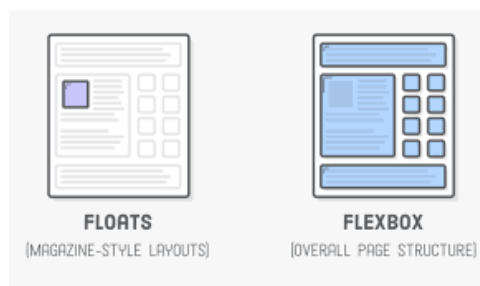
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="style.css">
7   <title>MiWeb</title>
8 </head>
9
10 <body>
11   <section>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorem consi
12     accusantium itaque minima laboriosam nihil voluptatem unde, nulla deserunt
13     nesciunt enim magnam</section>
14   <section>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure ad temp
15     quisquam consequuntur! Eum ipsum minima, odit, illum vitae corporis obaeci
16   </section>
17   <section>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Alias repud
18     placeat porro, recusandae ducimus animi repellat corporis quos omnis ratio
19     illo. Provident, dolor?</section>
20 </body>
21
22 </html>

style.css
1 section:first-of-type {
2   float: left;
3   width: 15%;
4   margin-right: 15px;
5   background-color: blue;
6 }
7
8 section:nth-of-type(2) {
9   float: left;
10  width: 30%;
11  margin-right: 15px;
12  background-color: red;
13 }
14
15 section:nth-of-type(3) {
16   float: left;
17   width: 40%;
18   background-color: orange;
19 }

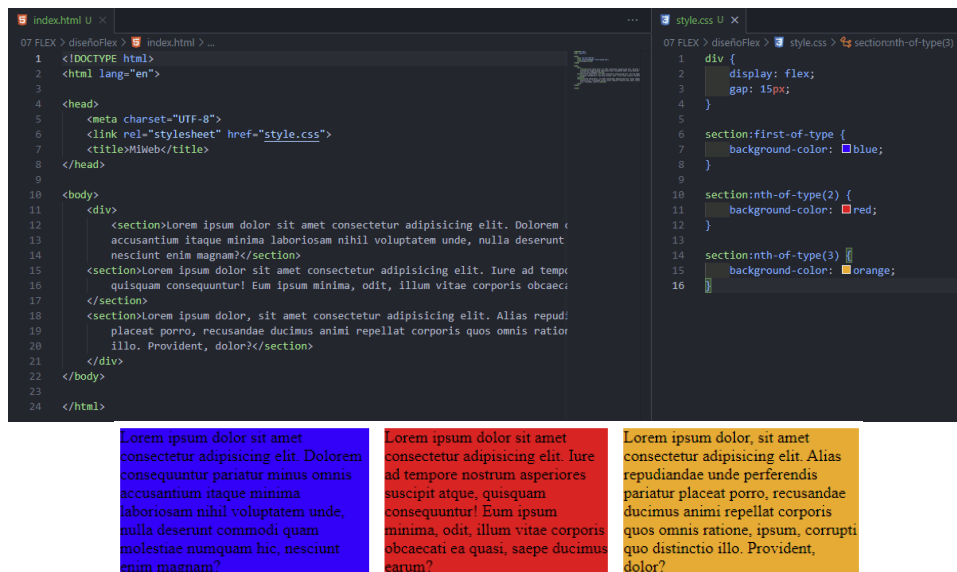
```



A pesar de que el posicionamiento flotante es una herramienta muy potente, es recomendable limitar su uso a flotar imágenes a derecha o izquierda para que un bloque de texto las envuelva pero para el diseño general es preferible usar otro tipo de herramientas como flexbox, grid o bootstrap ya que están diseñadas específicamente para la creación de una estructura general de la página y ya, dentro de cada uno de los bloques de la estructura, usar las propiedades float y clear para un mejor diseño.



Para utilizar flexbox, lo primero que tenemos que hacer es encerrar todos los bloques que queremos afectar dentro de un contenedor padre y, en el CSS, utilizar la propiedad display con el valor flex afectando a dicho contenedor padre. Además, podemos usar la propiedad gap para marcar un margen entre los bloques.



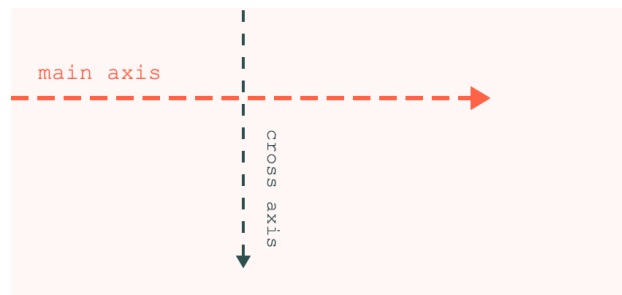
Hay que tener en cuenta que cuando usamos flexbox los valores predeterminados de visualización son ignorados, es decir, los párrafos dejarán de ser elementos en bloque, las imágenes dejarán de ser elementos en línea... y todos pasarán a ser elementos en flex y el ancho de cada uno de los elementos dentro del contenedor se basará, por lo general, en el tamaño del contenido.

ALGUNAS PROPIEDADES FLEX

A. FLEX-DIRECTION

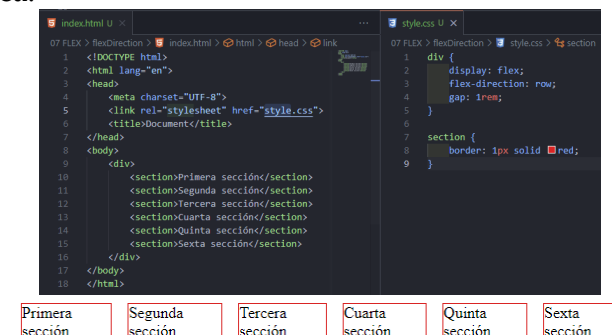
Se utiliza para establecer la dirección principal de los elementos flexibles en un contenedor flex. Es decir, va a afectar a cómo se distribuyen y alinean los elementos dentro del contenedor.

Para entender esta propiedad tenemos que pensar que un contenedor flex tiene 2 ejes, el eje principal (que por defecto va de izquierda a derecha) y el eje transversal (que por defecto va de arriba abajo).

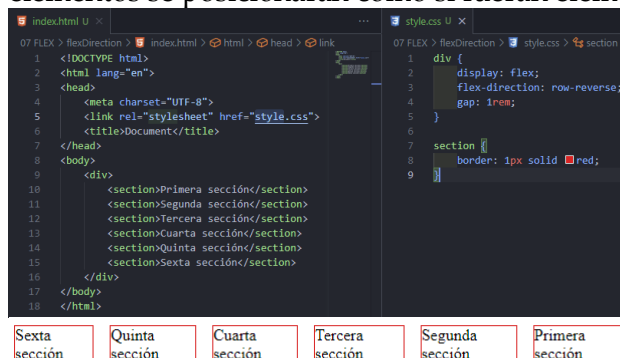


Esta propiedad tiene los siguientes valores:

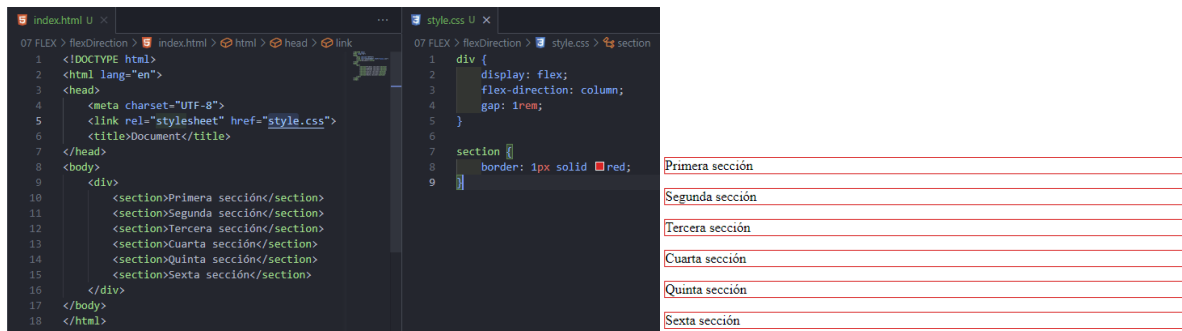
- **row** (valor por defecto). El eje principal va de izquierda a derecha y el eje transversal de arriba abajo. Los elementos se posicionarán como si fueran elementos en línea.



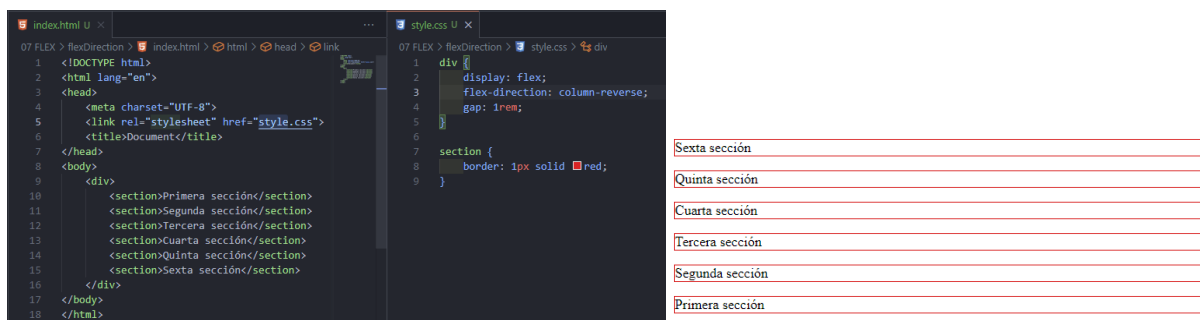
- **row-reverse**. El eje principal va a ir de derecha a izquierda (de forma que el primer elemento se situará a la derecha y el último a la izquierda) y el eje transversal de arriba abajo. Los elementos se posicionarán como si fueran elementos en línea.



- **column.** El eje principal va a ir de arriba abajo y el eje transversal de izquierda a derecha. Los elementos se van a posicionar como si fueran elementos en bloque.



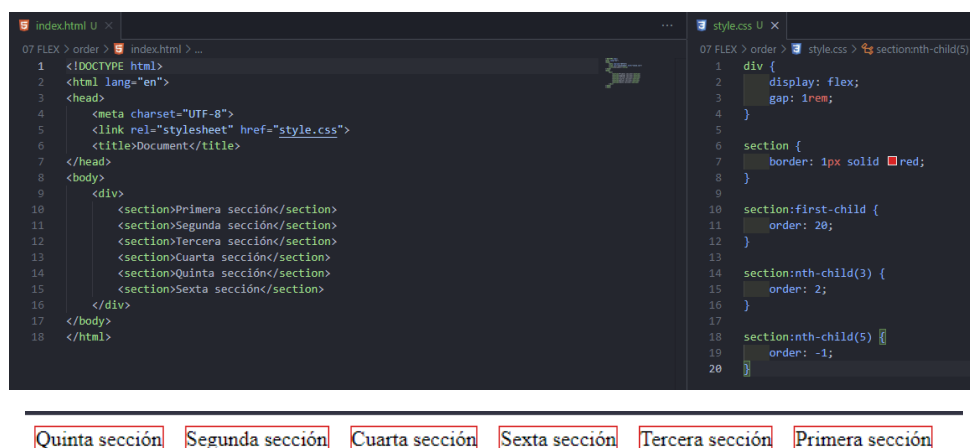
- **column-reverse.** El eje principal va a ir de abajo hacia arriba (de forma que el primer elemento será el último en aparecer y el último elemento será el primero en aparecer) y el eje transversal va de izquierda a derecha.



B. ORDER

La propiedad order nos va a permitir decidir el orden que deben ocupar los elementos dentro de un contenedor flexible. El valor por defecto es 0.

El valor asignador a order no indica la posición como tal, se pueden usar valores por encima del número de elementos y también números negativos. Simplemente determina un valor que, posteriormente, sirve para ordenar los elementos.

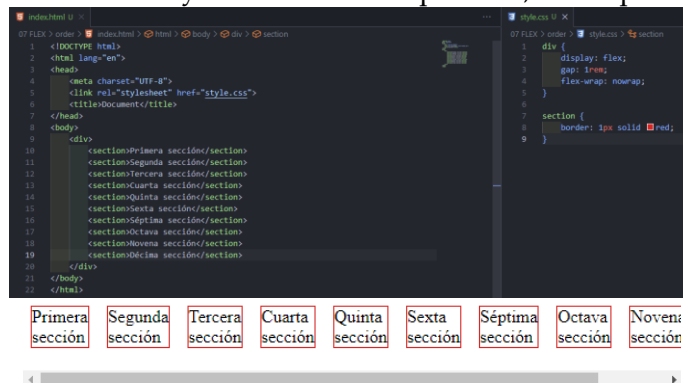


C. FLEX WRAP

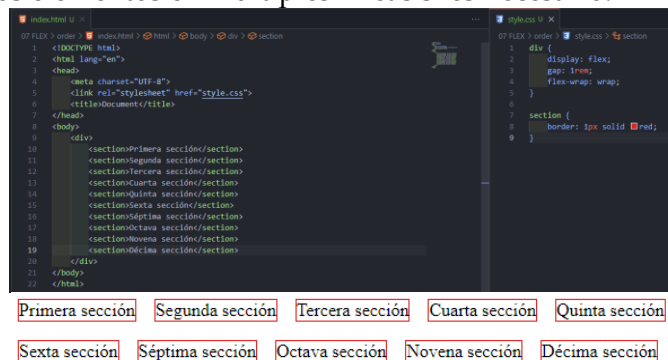
Esta propiedad define si los elementos flexibles pueden utilizar una única línea o varias.

Tiene dos valores:

- **nowrap** (valor por defecto). Alinea todos los elementos en una única línea. Si el número de elementos es mayor al ancho de la pantalla, estos quedarán ocultos.



- **wrap**. Alinea los elementos en múltiples líneas si es necesario.

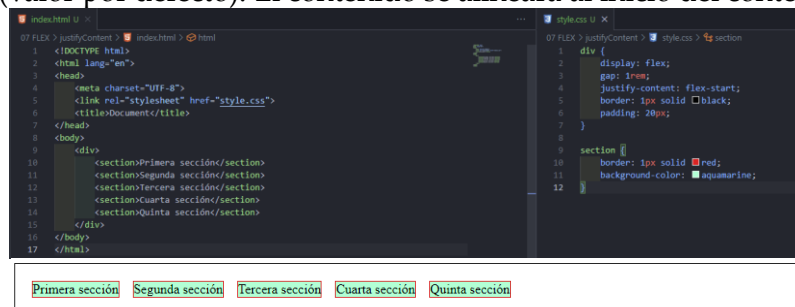


D. JUSTIFY CONTENT

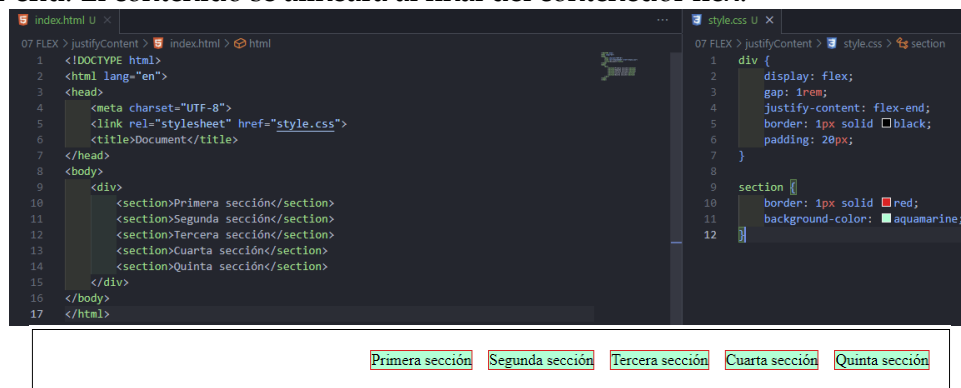
Establece la distribución dentro del eje principal de los elementos en el interior de un contenedor flex.

Tiene varios valores posibles, pero cabe destacar:

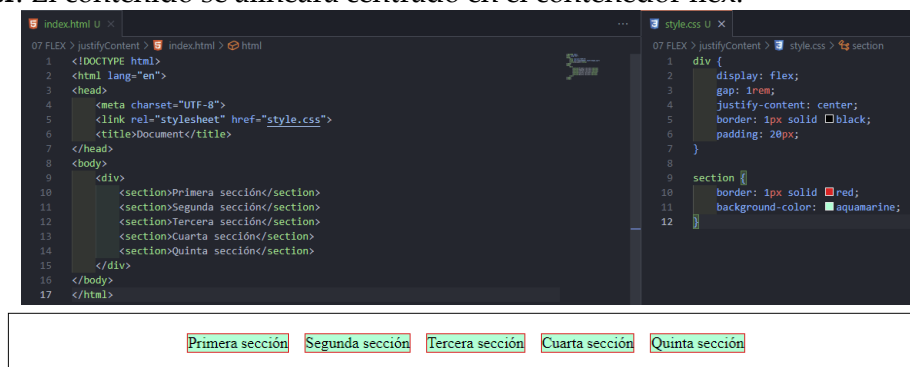
- **flex-start** (valor por defecto). El contenido se alinearán al inicio del contenedor flex.



- **flex-end.** El contenido se alineará al final del contenedor flex.



- **center.** El contenido se alineará centrado en el contenedor flex.



- **space-between.** Permite que el contenido se alinee a lo largo del contenedor dejando un espaciado entremedias.

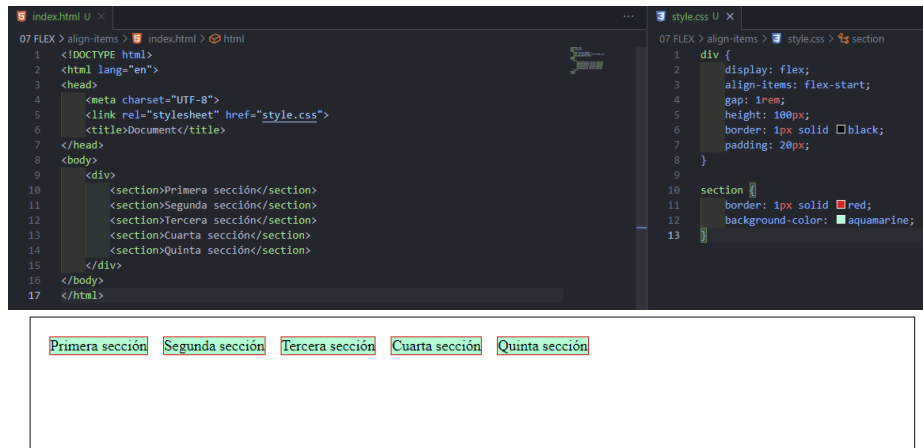


E. ALIGN ITEMS

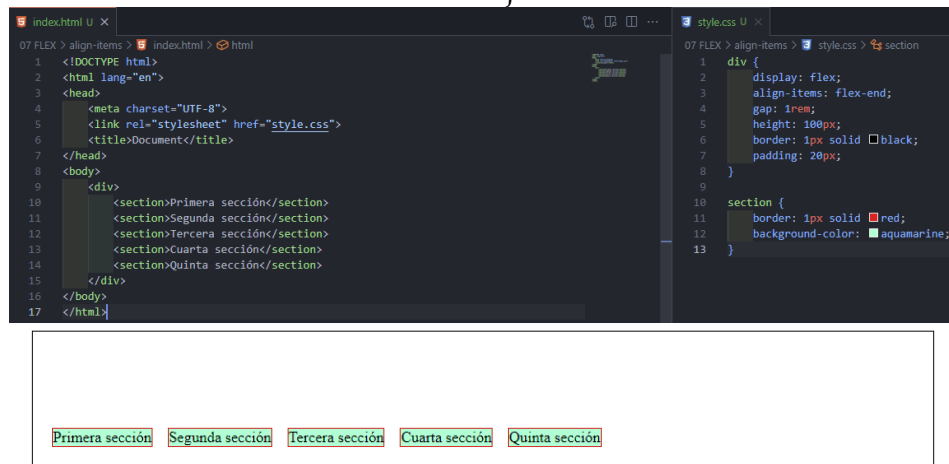
En este caso se va a afectar la distribución de los elementos a lo largo del eje transversal.

Tiene varios valores posibles:

- **flex-start** (valor por defecto). Alinea los elementos al inicio del eje transversal.



- **flex-end.** Alinea los elementos al final del eje transversal.



- **center.** Alinea los elementos en el centro del eje transversal.

