FlexBox

Introducción

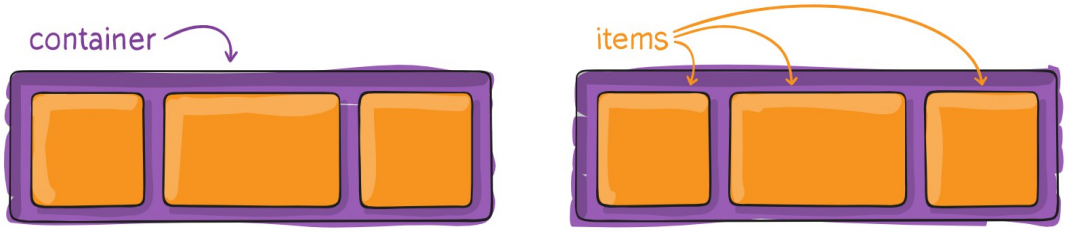
El módulo de diseño de FlexBox (caja flexible) tiene como objetivo proporcionar una forma más eficiente de diseñar, alinear y distribuir espacio entre los elementos en un contenedor, incluso cuando su tamaño es desconocido y / o dinámico (de aquí la palabra "flex").

La idea principal del diseño FlexBox es darle al contenedor la capacidad de alterar el ancho / alto de sus items (y el orden) para llenar mejor el espacio disponible (principalmente para acomodarlo a todo tipo de dispositivos de visualización y tamaños de pantalla). Un contenedor FlexBox expande los elementos para llenar el espacio libre disponible o los reduce para evitar el desbordamiento.

Lo más importante es que el diseño FlexBox es "agnóstico" con los diseños regulares (elementos de bloque que están basados en la orientación vertical y elementos de línea que están basados en la orientación horizontal). Estos diseños funcionan bien pero carecen de la "flexibilidad" necesaria para utilizarlas en aplicaciones grandes o complejas (especialmente cuando se trata de cambiar la orientación, cambiar el tamaño, estirar, encoger, etc.).

Conceptos básicos y terminología

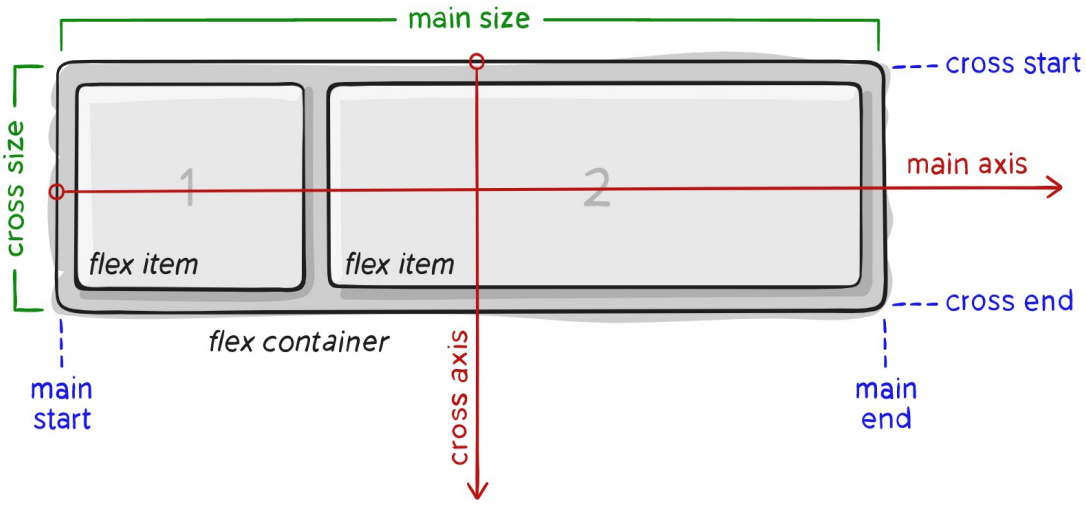
Flexbox es un módulo completo y no simplemente una propiedad CSS. Esto quiere decir que FlexBox incluye su propio conjunto completo de propiedades. Algunas de estas propiedades son propias del contenedor (elemento padre, conocido como "contenedor flex"), mientras que otras son propiedades de los elementos "hijo" (también llamados "flex items").



El diseño "regular" se basa en direcciones de flujo en bloque y en línea. El diseño flexible se basa en "direcciones de flujo flexible".

Los elementos se colocarán siguiendo el eje principal o **main axis** (desde el **main-start** hasta el **main-end**) o se colocarán siguendo el eje transversal o **cross axis** (desde el **cross-start** hasta el **cross-end**).

Vamos a explicar la idea principal del diseño flexible con el siguiente gráfico:



eje principal o main axis: el eje principal de un contenedor flexible. Es el eje primario a lo largo del cual se disponen los elementos flexibles (items "hijo"). No tiene que ser necesariamente horizontal; depende de la propiedad "flex-direction" que se explicará más adelante.

main-start | main-end: los elementos flexibles (items "hijo") se colocan dentro del contenedor flex comenzando desde main-start y yendo hacia main-end.

tamaño principal o main size: es el ancho o la altura de un elemento flexible dependiendo de la propiedad "flex-direction".

eje transversal o cross axis: es el eje perpendicular al eje principal o main axis. Su dirección depende de la dirección del eje principal o main axis.

cross-start | cross-end: los flex items (elementos "hijo") van rellenando líneas en el contenedor comenzando en cross-start del contenedor flexible y yendo hacia el cross-end.

cross size: es el ancho o el alto de un elemento flexible dependiendo de la propiedad "flex-direction".

Propiedades para el “parent” o Flex Container

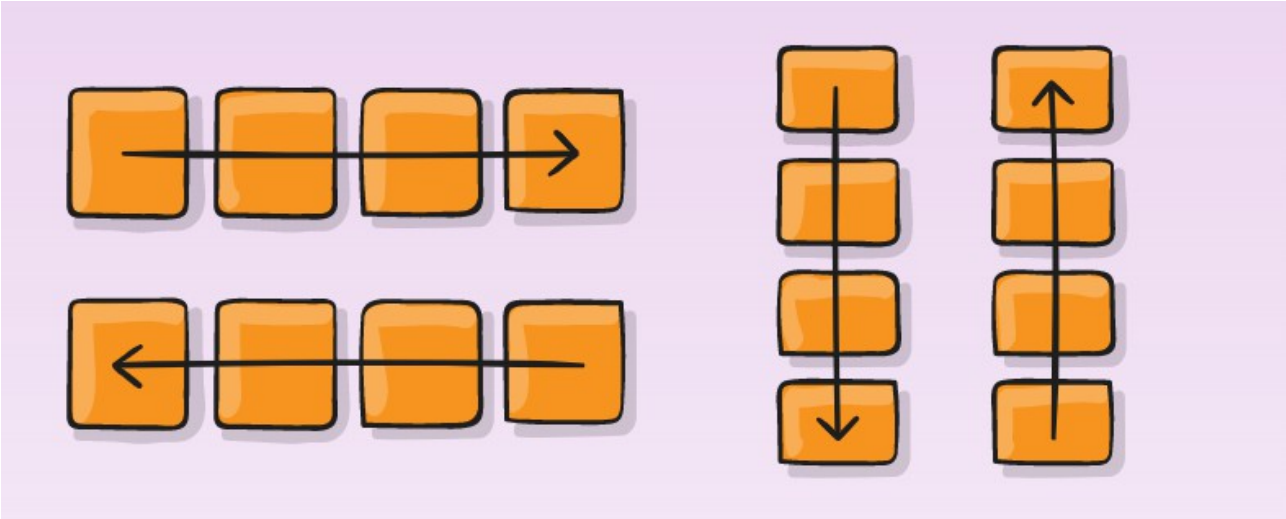
display

Define un contenedor flexible. En línea o en bloque dependiendo del valor dado. Permite un contexto flexible para todos sus hijos directos.

.container { display: flex; ** o inline-flex **/ }

“display: inline-flex” no hace que los flex items (“hijos”) se comporten como elementos en línea. El que se comportará como un elemento en línea será el “parent”, es decir el “container”.

flex-direction

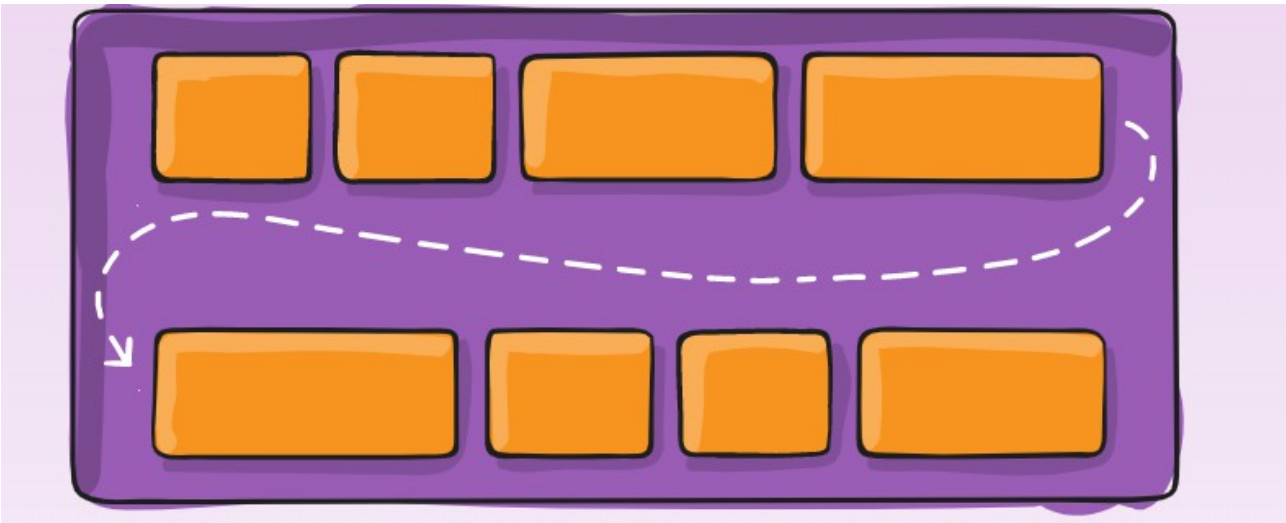


Establece el eje principal (main axis), definiendo así la dirección en la que se colocan los elementos flexibles en el contenedor flexible. Flexbox es un concepto de diseño de una sola dirección. Debes pensar en los elementos flexibles (items "hijo") como distribuidos en filas horizontales o columnas verticales.

```
.container {
  flex-direction: row; /*o "row-reverse" o "column" o "column-reverse"*/
}
```

row: es la propiedad por defecto (de izquierda a derecha).
row-reverse: de derecha a izquierda.
column: de arriba hacia abajo.
column-reverse: de abajo hacia arriba.

flex-wrap



De forma predeterminada, todos los elementos flexibles (flex items o elementos “hijo”) intentarán encajar en una línea. Podemos cambiar esto respetando el ancho fijo establecido en los elementos “hijo” haciendolos saltar de línea.

```
.container{
  flex-wrap: nowrap; /* o "wrap" o "wrap-reverse"*/
}
```

nowrap: es la propiedad por defecto. Todos los flex items se colocarán en 1 línea.
wrap: se respeta el ancho establecido en los flex items y se colocarán en múltiples líneas de arriba hacia abajo si fuera necesario.
wrap-reverse: los flex items se colocarán en múltiples líneas de abajo hacia arriba.

flex-flow

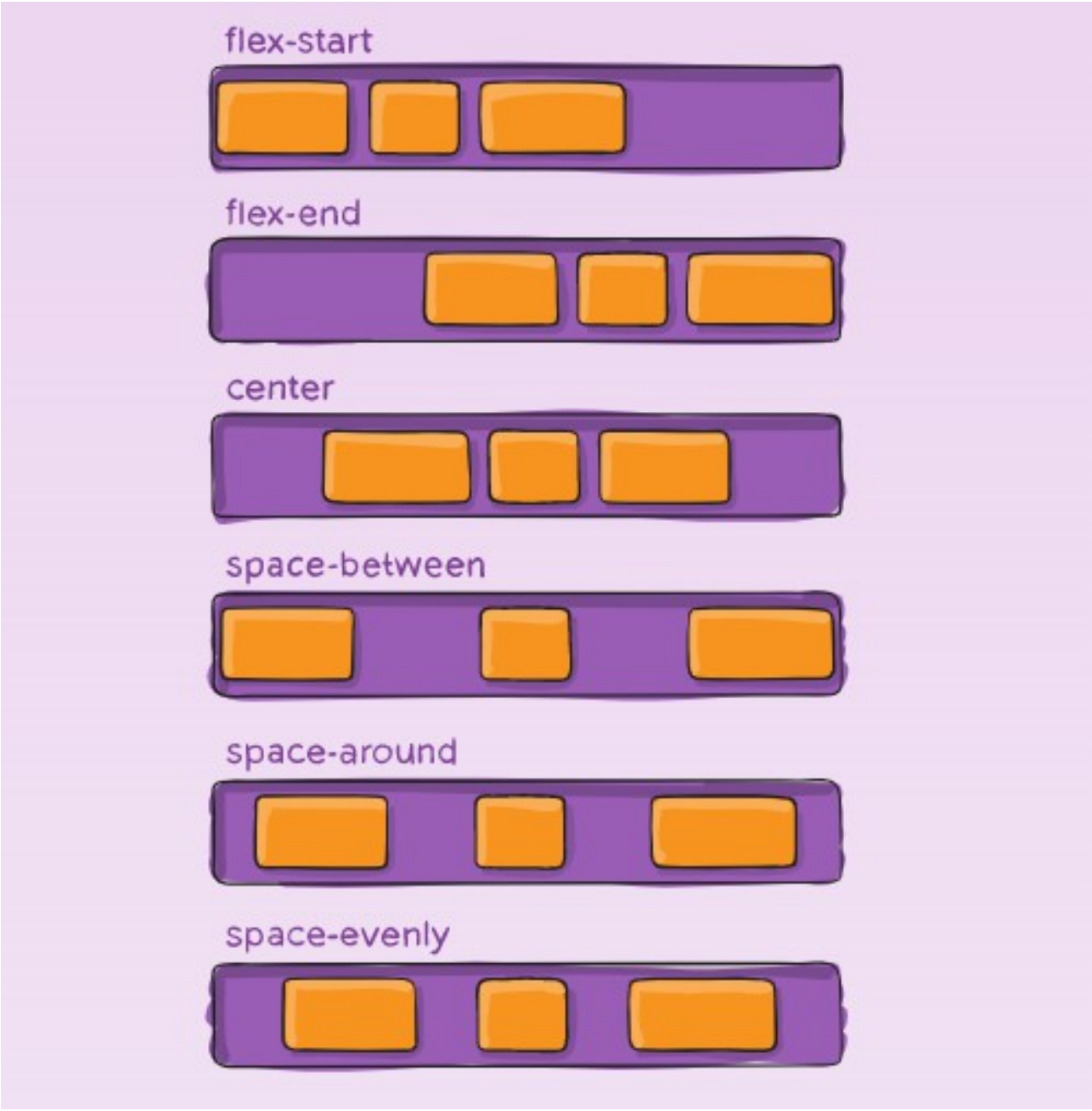
Es un “atajo” (shorthand) para aplicar “flex-flow” y “flex-wrap” conjuntamente. Su valor por defecto es: row nowrap.

Veamos un ejemplo:

```
.container{
  flex-flow: column wrap;
}
```

justify-content

Define la disposición de los elementos hijo a través del eje principal (main axis).



Ayuda a distribuir las sobras de espacio libre adicional cuando todos los elementos hijo de una línea tienen un ancho determinado o son flexibles pero han alcanzado su tamaño máximo. También ejerce cierto control sobre la alineación de los elementos cuando desbordan la línea.

```
.container{
    justify-content: flex-start; /* o "flex-end" o "center" o “space-between”...etc*/
}
```

flex-start: es la propiedad por defecto. Los items se aglutinan desde el inicio de “flex-direction”.

flex-end: los items se aglutinan desde el final de “flex-direction”.

center: los items se aglutinan en el centro.

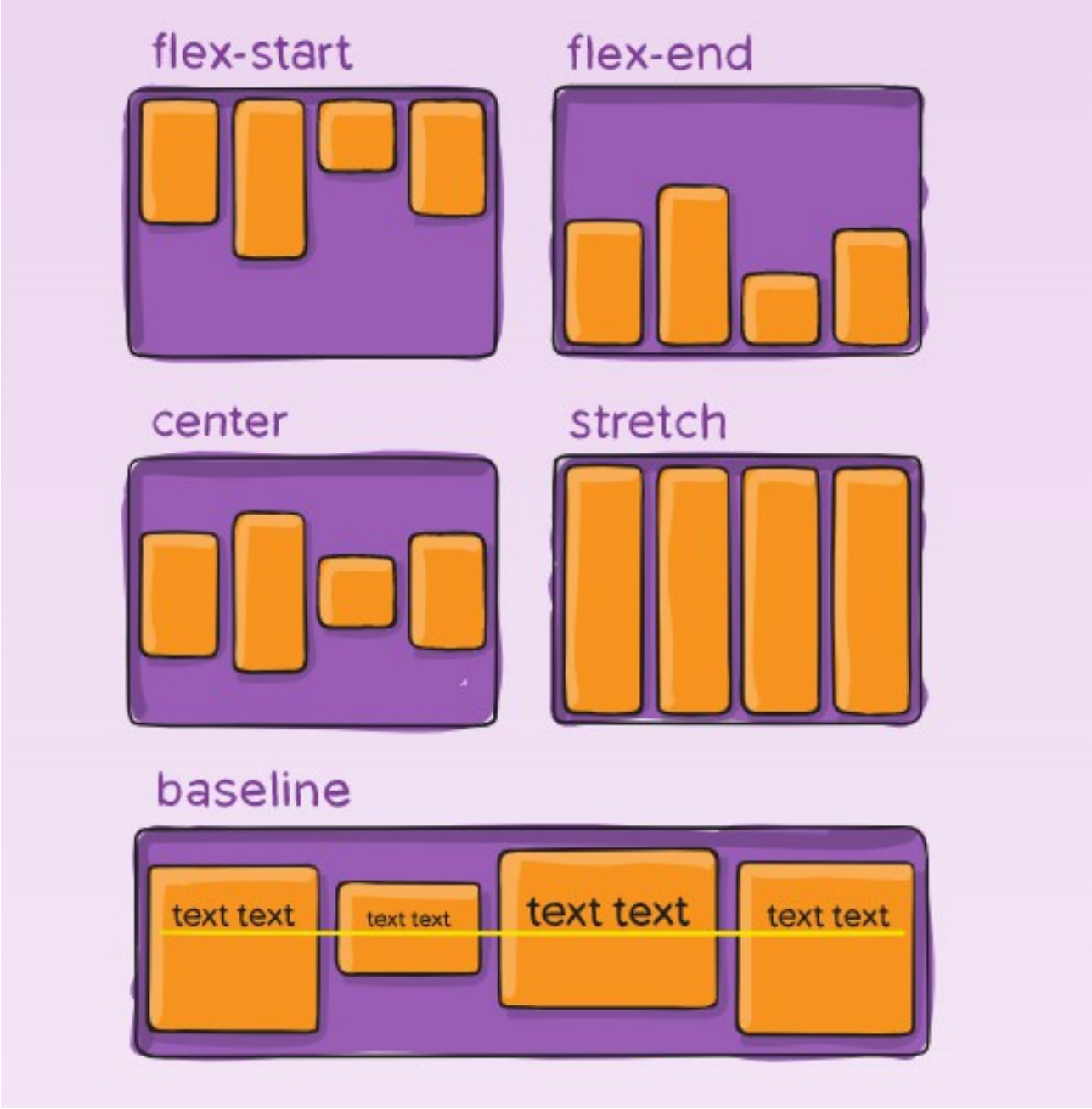
space-between: El primer item se sitúa en el comienzo de la línea y el último item al final. Los demás items se distribuyen, entre el primer y último item, equidistantes a lo largo de toda la linea.

space-around: los elementos se distribuyen uniformemente en la línea con igual espacio a su alrededor. Hay que tener en cuenta que visualmente los espacios no son iguales, ya que todos los elementos tienen el mismo espacio en ambos lados. El primer elemento tendrá una unidad de espacio contra el borde del contenedor, pero dos unidades de espacio entre el siguiente elemento porque el siguiente elemento tiene su propio espacio que se aplica.

space-evenly: los elementos se distribuyen de modo que el espacio entre dos elementos (y el espacio hasta los bordes) sea igual.

align-items

Define cómo se disponen los elementos flexibles a lo largo del eje transversal (cross-axis). Podemos pensar que esta propiedad es similar a la anteriormente estudiada "justify-content" pero para el eje transversal (perpendicular al eje principal).



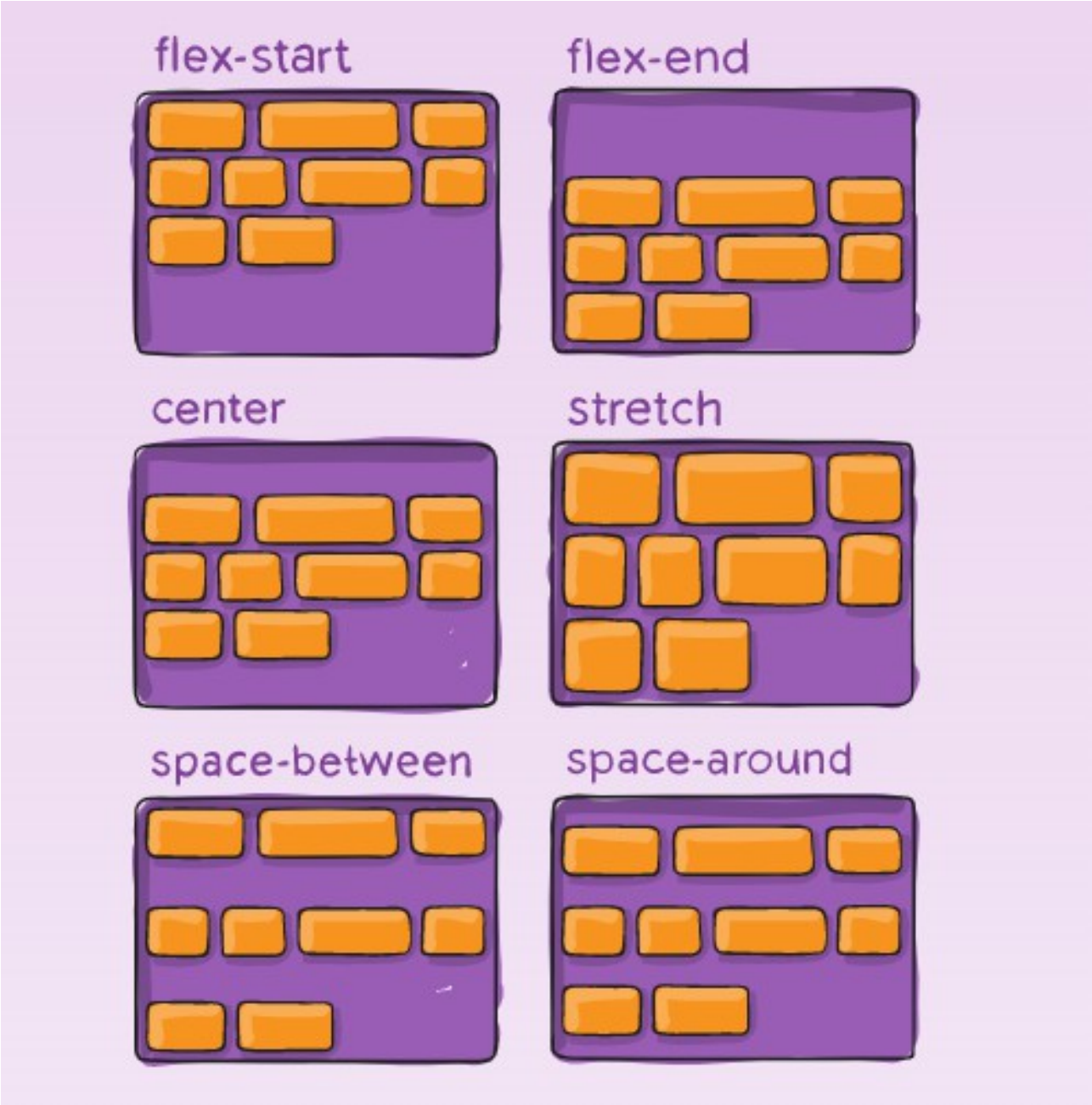
.container{
align-items: flex-start; /* *o "flex-end" o "center" o “stretch”...etc/**
}

stretch: es la propiedad por defecto. Los items se “estiran” hasta rellenar el contenedor.

flex-start: los items se aglutinan desde el principio del “cross-axis”.
flex-end: los items se aglutinan desde el final del “cross-axis”.
center: los items se aglutinan en el centro del “cross-axis”.
baseline: los items se alinean en relación a su baseline.

align-content

Actúa alineando las líneas de un contenedor flexible cuando hay espacio adicional en el eje transversal, de forma similar a cómo “justify-content” alinea elementos individuales dentro del eje principal.
Nota: esta propiedad no tiene efecto cuando solo hay una línea de elementos flexibles.



```
.container{
    align-content: flex-start; /* o "flex-end" o "center" o “stretch”...etc*/
}
```

stretch: es la propiedad por defecto. las líneas se “estiran” hasta ocupar el espacio restante

flex-start: los elementos se aglutinan en líneas al principio del contenedor.

flex-end: los elementos se aglutinan en líneas al final del contenedor.

center: los elementos se aglutinan en líneas que quedan centradas en el contenedor.

space-between: las líneas se distribuyen uniformemente. La primera línea está al comienzo del contenedor mientras que la última está al final.

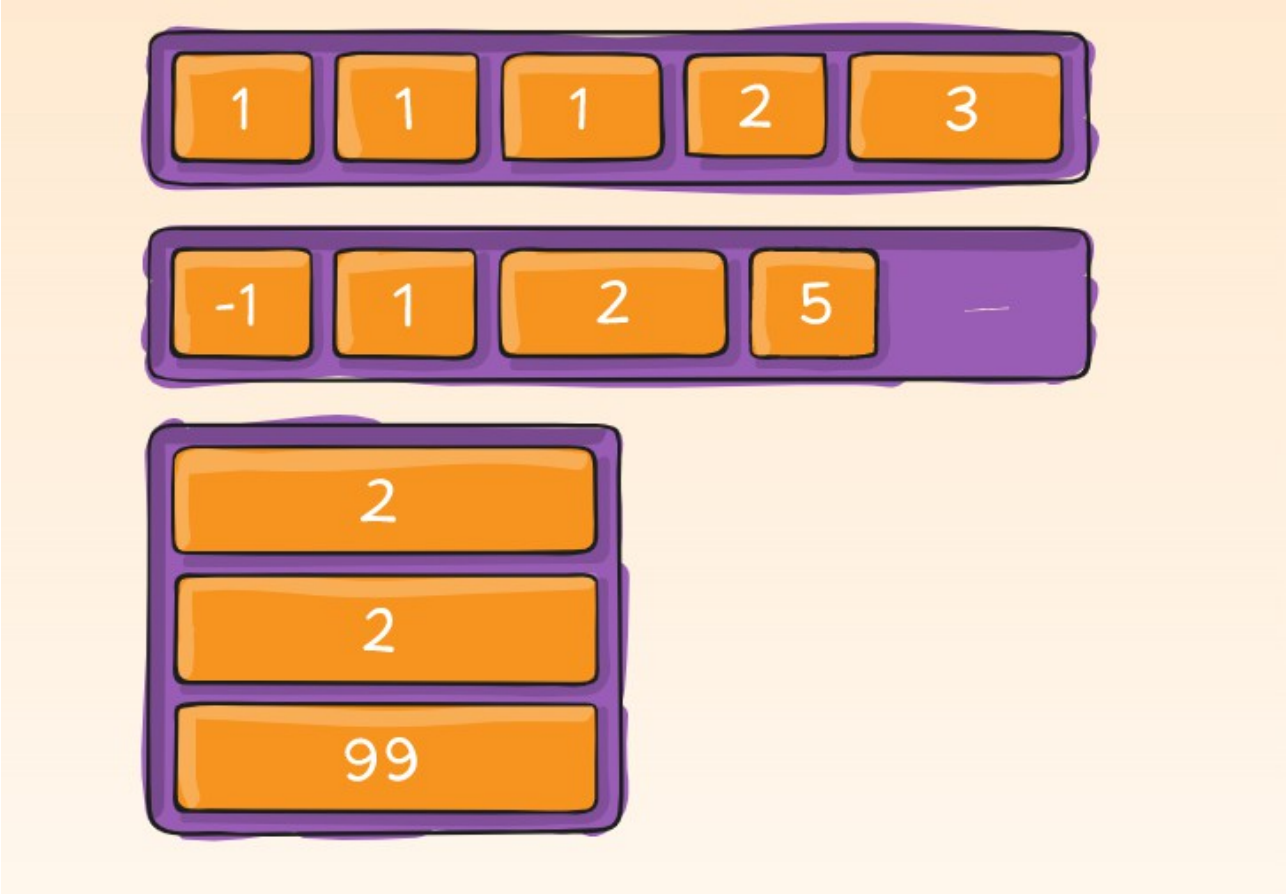
space-around: los elementos se aglutinan en líneas que se distribuyen uniformemente con igual espacio alrededor de cada línea.

space-evenly: los elementos se aglutinan en líneas que se distribuyen uniformemente con igual espacio a su alrededor.

Propiedades para los elementos “hijo” o Flex Items

order

Por defecto, los elementos flexibles se presentan en el orden de origen. Sin embargo, la propiedad de “order” controla el orden en que aparecen en el contenedor flexible.

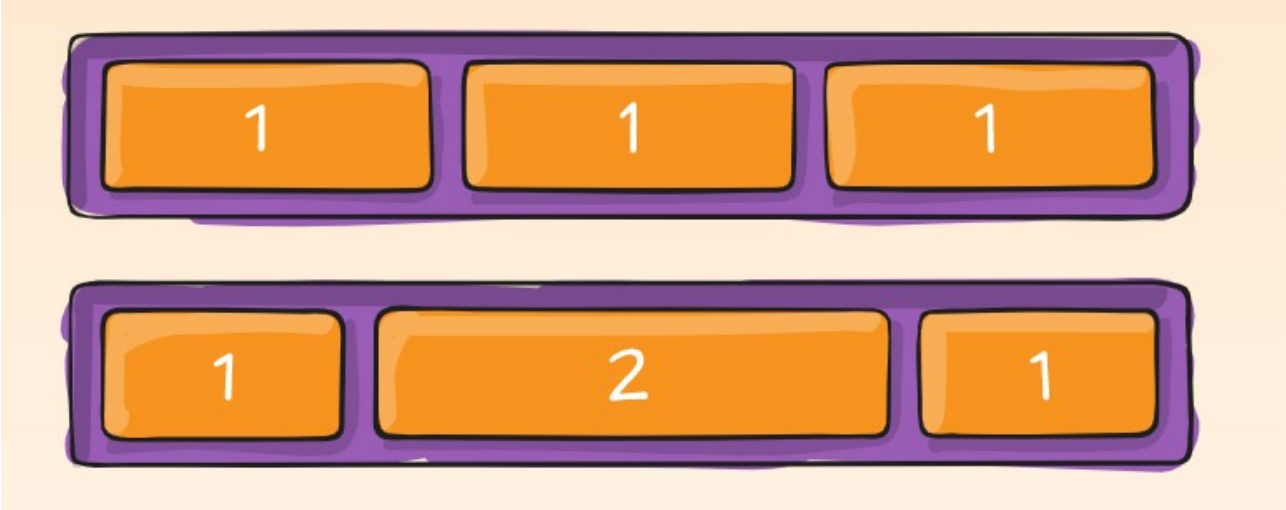



```
.item-hijo{
    order: 4; /* un número entero. Por defecto es el 0*/
}
```

flex-grow

Define la capacidad de un elemento flexible de crecer si es necesario. Acepta un valor sin unidades que sirve como proporción. Dicta qué cantidad de espacio disponible dentro del contenedor flexible debe ocupar el item.

Si todos los items tienen establecido un “flex-grow” de 1, el espacio del contenedor se distribuirá por igual a todos ellos. Si uno de los items tiene un valor de 2, ocuparía el doble de espacio que los demás (o al menos lo intentará).




```
.item-hijo{
    flex-grow: 2; /* un número entero. Por defecto es el 0. Los números negativos son inválidados*/
}
```

flex-shrink

Define la capacidad de un elemento flexible de encogerse si es necesario.

```
.item-hijo{
    flex-shrink: 2; /* un número entero. Por defecto es el 0. Los números negativos son inválidados*/
}
```

flex-basis

Define el tamaño predeterminado de un item . Puede ser una longitud (por ejemplo, 20%, 5rem, etc.)

```
.item-hijo{
    flex-basis: 2rem; /* o “auto” . En general una longitud*/
}
```

flex

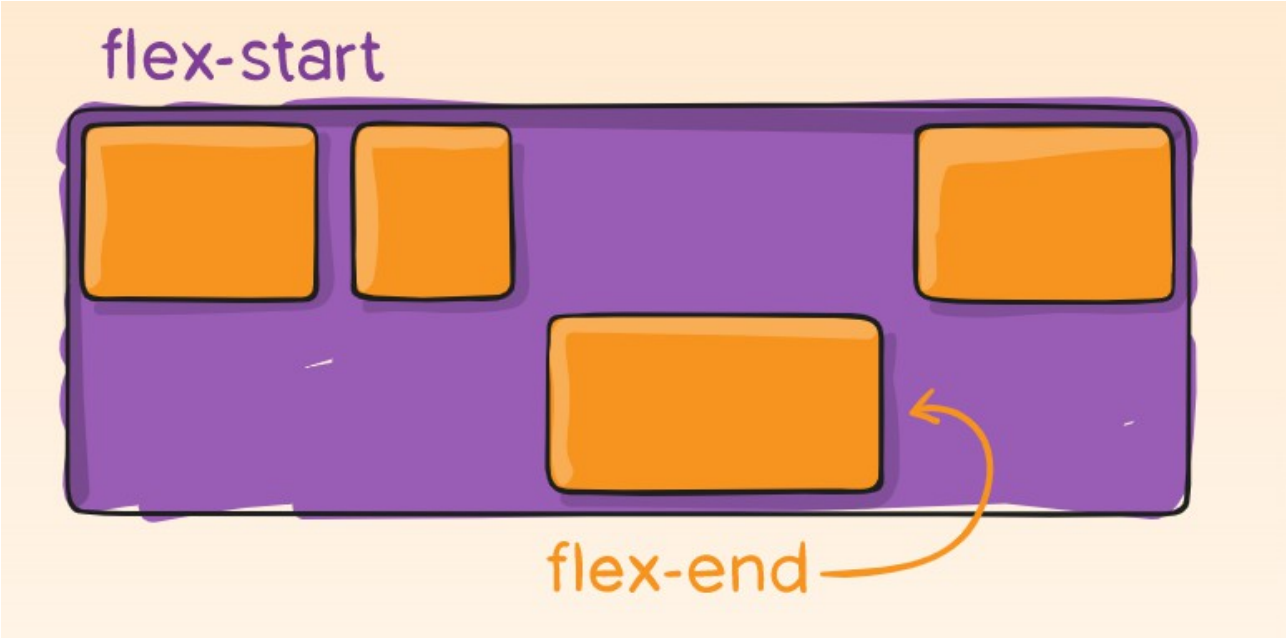
Es un atajo (shorthand) de “flex-grow”, “flex-shrink” y “flex-basis” combinados. El segundo y tercer parámetro (“flex-shrink” y “flex-basis”) son opcionales.

```
.item-hijo{
    flex: 2 2 3rem;
}
```

align-self

Permite que la alineación predeterminada (o la especificada con “align-items”) se anule para los elementos flexibles individuales.

Consulta la explicación de “align-items”, estudiada previamente, para comprender los valores disponibles.



```
.item-hijo{
    align-self: flex-start; /* o "auto"  o "flex-end" o "stretch"...etc.*/
}
```

Nota: la propiedad CSS "vertical-align" no tiene ningún efecto sobre un flex item.