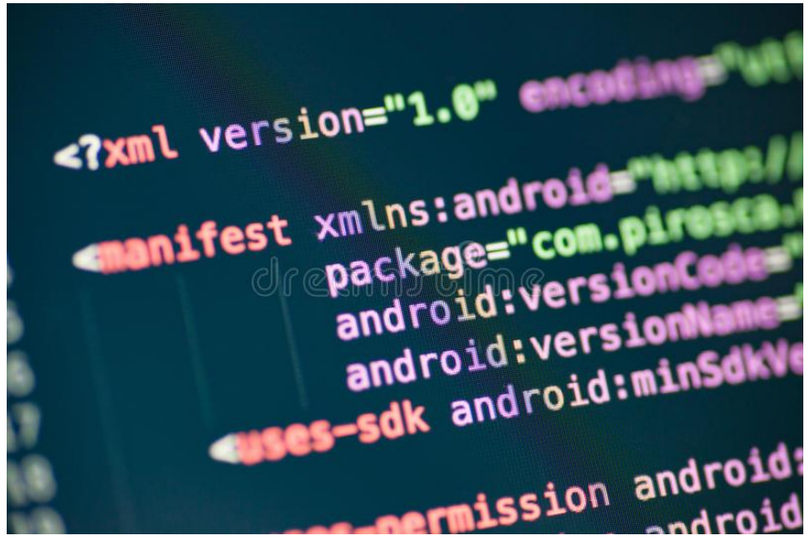
Lenguajes de marcas y sistemas de gestión de la información



UT03 – CSS

8 – Float – Flex

IES Clara del Rey – Madrid

Float

La propiedad "float" permite determinar como un elemento debe "flotar" o "empujarse" a un lado u otro.

Puede tomar los valores:

- none: no flota. Se representa como corresponda en el flujo de página. Es el valor por defecto.
- right: se desplaza a la derecha de su contenedor
- left: se desplaza a la izquierda de su contenedor

El caso de uso más habitual es flotar las imágenes dentro de un párrafo a la izquierda o la derecha.

También permite "apilar" elementos a un lado o al otro.

Float

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae

. Mauris ante
esent convallis
Nunc sagittis
.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum di

Float Next To Each Other

In this example, the three divs will float next to each other.

Div 1 Div 2 Div 3

Clear

Clear permite "romper" el flotado. Puede tomar los siguientes valores:

- right: dejarán de flotar los elementos que flotaban a la derecha
- left: dejarán de flotar los elementos que flotaban a la izquierda
- right: dejarán de flotar todos los elementos, ya sea que flotaran a la izquierda o a la derecha.

Dejar de flotar no significa que no floten los anteriores al elemento con "clear". Significa que ese elemento "reinicia el flotado". Si no tiene float, se comportará de forma natural, y si lo tiene, volverá a empezar a flotar. En este último caso sería como insertar un salto de línea.

Float / Clear para maquetación

Hace años, antes de que aparecieran mejores sistemas (flex y grid), se usaba float para realizar la maquetación de los diseños.

Esto, hoy día no es correcto. No debe usarse, en general, para maquetación, salvo casos concretos.

Como se ha comentado, el caso de imágenes flotadas junto a párrafos es uno de los más habituales, sobre todo si:

- Se usan imágenes con fondos transparentes.
- Se combina con propiedades para "rodear" la imagen con texto, ocupando las partes transparentes: shape-outside, shape-margin y shape-threshold.

Consultar MDN para ejemplos de uso de shape-outside y otras propiedades relacionadas.

Flex

Flex (y grid) son dos formas de posicionar / organizar elementos dentro de un contenedor.

Antes de flex se usaba float y position para organizar elementos, pero era un sistema rudimentario, con gran cantidad de problemas asociados, y que no respondía del todo bien a la variedad de navegadores y dispositivos.

Flex, también llamado flexbox, está pensado para organizar elementos en una dimensión, esto es, en una fila o en una columna.

Aunque puede responder bien para formatos en "tabla", si queremos hacer eso, es mejor usar grid.

Flex – Utilidad

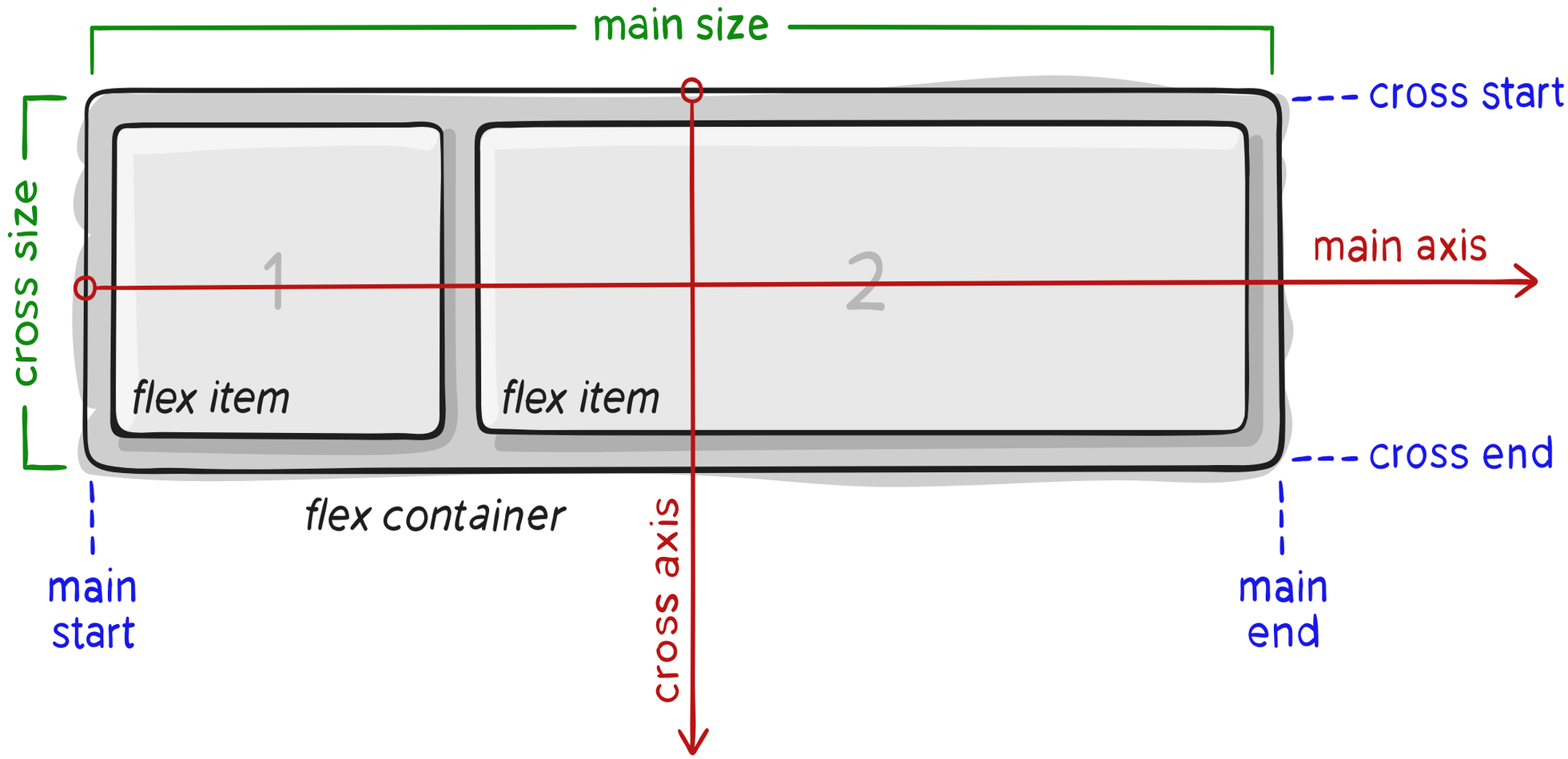
Viene a hacer más fáciles tareas que con float / position son bastantes complicadas, aunque parezcan básicas, como:

- Centrar verticalmente un elemento dentro de su contenedor
- Centrar horizontalmente un elemento dentro de su contenedor
- Hacer que todos los elementos dentro de un contenedor ocupen el mismo espacio
- Hacer que el espaciado entre elementos sea siempre el mismo
- Que varias columnas contiguas tengan el mismo alto independientemente de que tengan más o menos contenido

Flex – Elementos que lo forman

- Flex container: contenedor (elemento externo) donde vamos a utilizar flexbox.
- Flex ítems: elementos secundarios (dentro del flex container) que queremos alinear / posicionar.
- Para utilizar Flexbox hay que aplicar propiedades CSS en el flex contaniner, y casi siempre en los flex itemsç
- Eje principal y transversal. Flex está diseñado para trabajar en una dimensión, pero esta puede ser horizontal o vertical, y puede trabajar en el sentido de escritura o en el contrario. Los ejes determinan como se distribuyen los elementos.

Flex – Elementos que lo forman



Flex – Elementos que lo forman

- Eje principal (main axis): a lo largo de él se colocan los flex ítems. Usando flex-direction podemos definir la dirección (vertical/horizontal) y sentido (izquierda/derecha/arriba/abajo).
- Main-start y main-end: los flex ítems se colocan en el contenedor empezando por el lado start, dirigiéndose hacia el lado end. Ya no se utiliza left o right, ni top o bottom.
- Tamaño principal (main size): El ancho o alto de un flex container, dependiendo de la dirección del contenedor, es el tamaño principal del ítem. La propiedad de tamaño principal de un flex ítem puede ser tanto width como height, dependiendo de cuál esté en la dirección principal.

Flex – Elementos que lo forman

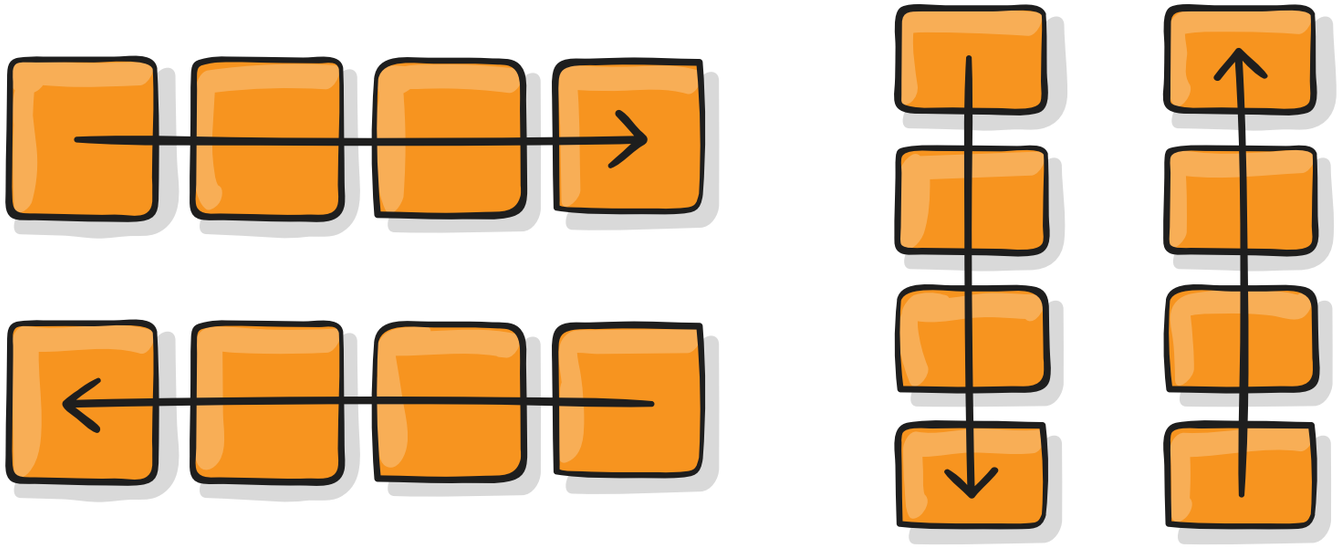
- Eje transversal (cross axis): El eje perpendicular al eje principal se llama eje transversal. Su dirección depende de la dirección del eje principal.
- Cross-start | cross-end: Líneas flex se llenan con ítems y se agregan al contenedor, comenzando desde el lado cross start del flex container hacia el lado cross end.
- Tamaño transversal (cross size): El ancho o alto de un flex ítem, dependiendo de lo que haya en la dimensión transversal, es el cross size del ítem. La propiedad cross size puede ser el ancho o el alto del ítem, lo que se encuentre en la transversal.

Flex – Propiedades del container

- display:
  - flex, inline-flex
  - Ambas opciones hacen que el contenedor sea un flex container.
  - La diferencia está en cómo se comporta el contenedor respecto a lo que le rodea:
    - "flex" hace que el contenedor sea un elemento en bloque, y ocupe todo el ancho disponible, provoque salto de línea, etc.
    - "inline-flex" hace que sea un elemento en línea
  - Internamente, dentro del contenedor, los elementos se distribuirán siguiendo la estructura flex.

Flex – Propiedades del container

- flex-direction:
  - row, row-reverse, column, column-reverse
  - establece el eje principal (dirección y sentido)



Flex – Propiedades del container

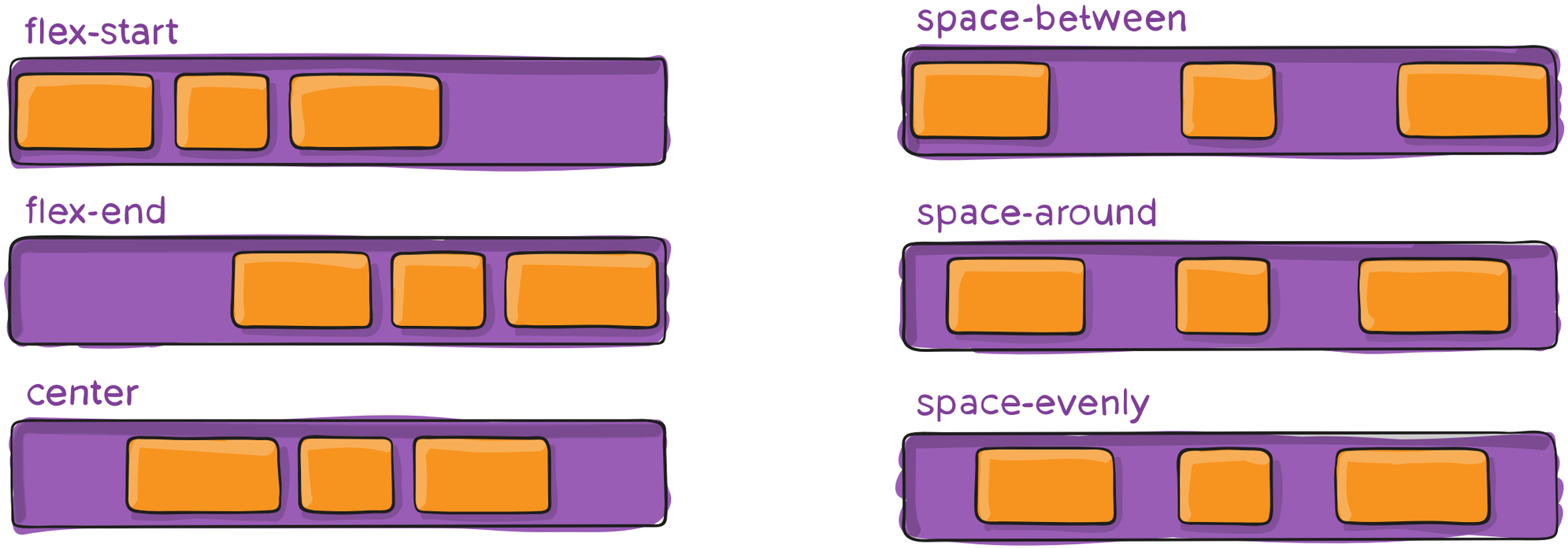
- flex-wrap:
  - nowrap, wrap, wrap-reverse
  - Por defecto todos los flex ítems se colocan en una sola fila o columna (nowrap).
  - Esto permite que los ítems pasen a la siguiente línea cuando sea necesario.
  - La diferencia entre wrap y wrap-reverse es que el primero llena la primera línea y añade en la dirección del eje transversal, y el segundo añade líneas en sentido contrario al eje transversal, es decir, "antes" de la primera línea.

Flex – Propiedades del container

- flex-flow:
  - Es una abreviatura para flex-direction y flex-wrap. Ej: flex-flow: row wrap
- justify-content:
  - flex-start, flex-end, center, space-between, space-around, space-evenly
  - Alineación de los items respecto al eje principal
  - Ayuda a distribuir el espacio libre en el contenedor

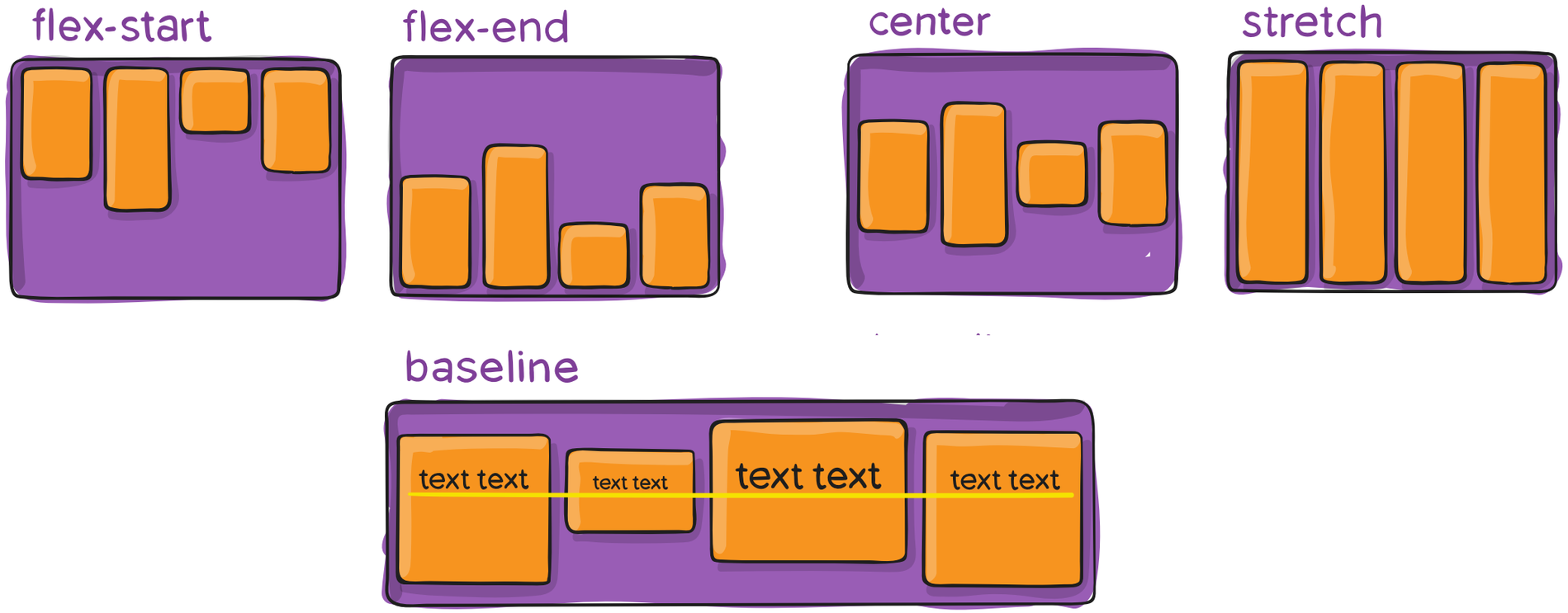
Flex – Propiedades del container

- justify-content



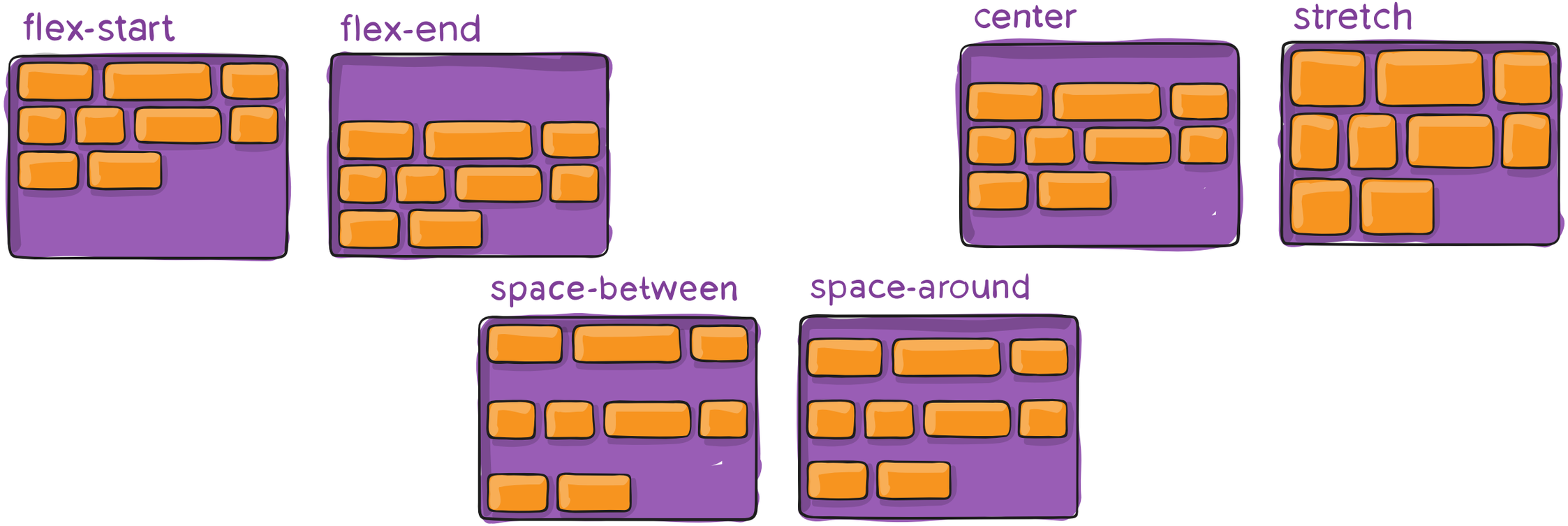
Flex – Propiedades del container

- align-items:
  - stretch, flex-start, flex-end, center, baseline
  - Comportamiento respecto al eje transversal (cross axis)



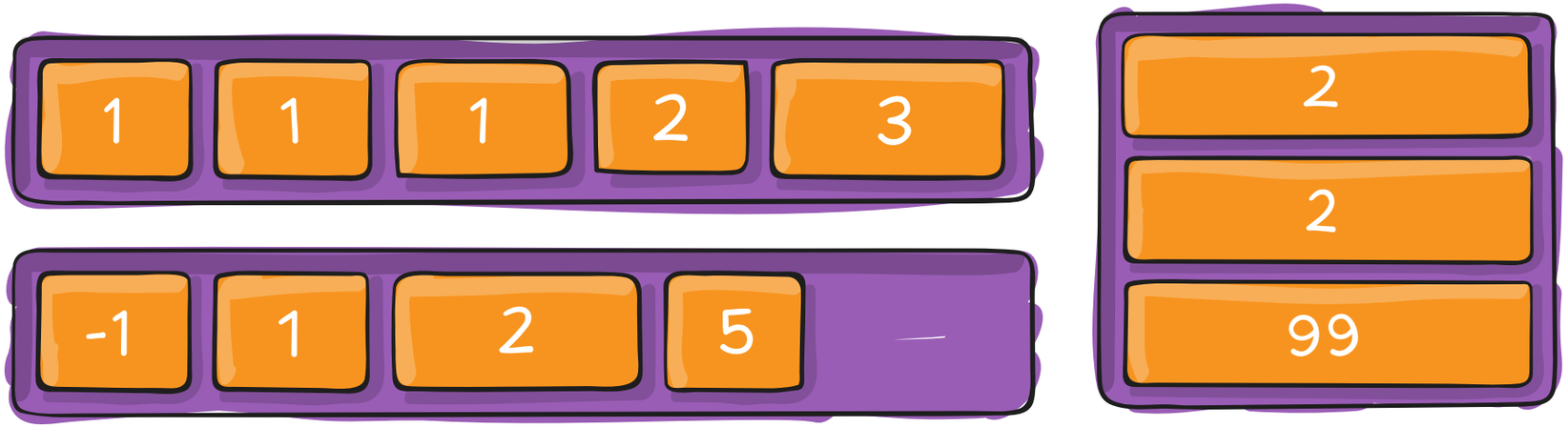
Flex – Propiedades del container

- align-content:
  - flex-start, flex-end, center, space-between, space-around, stretch
  - Alineación de los items respecto al eje tranversal, ayuda a distribuir el espacio libre en el contenedor



Flex – Propiedades de los items

- order:
  - Determina el orden del elemento.
  - Por defecto es 0: se respeta el orden de los elementos.
  - Si se usa un número distinto de cero, permite mover uno o varios elementos el elemento al inicio o al final (según el sentido del eje principal).

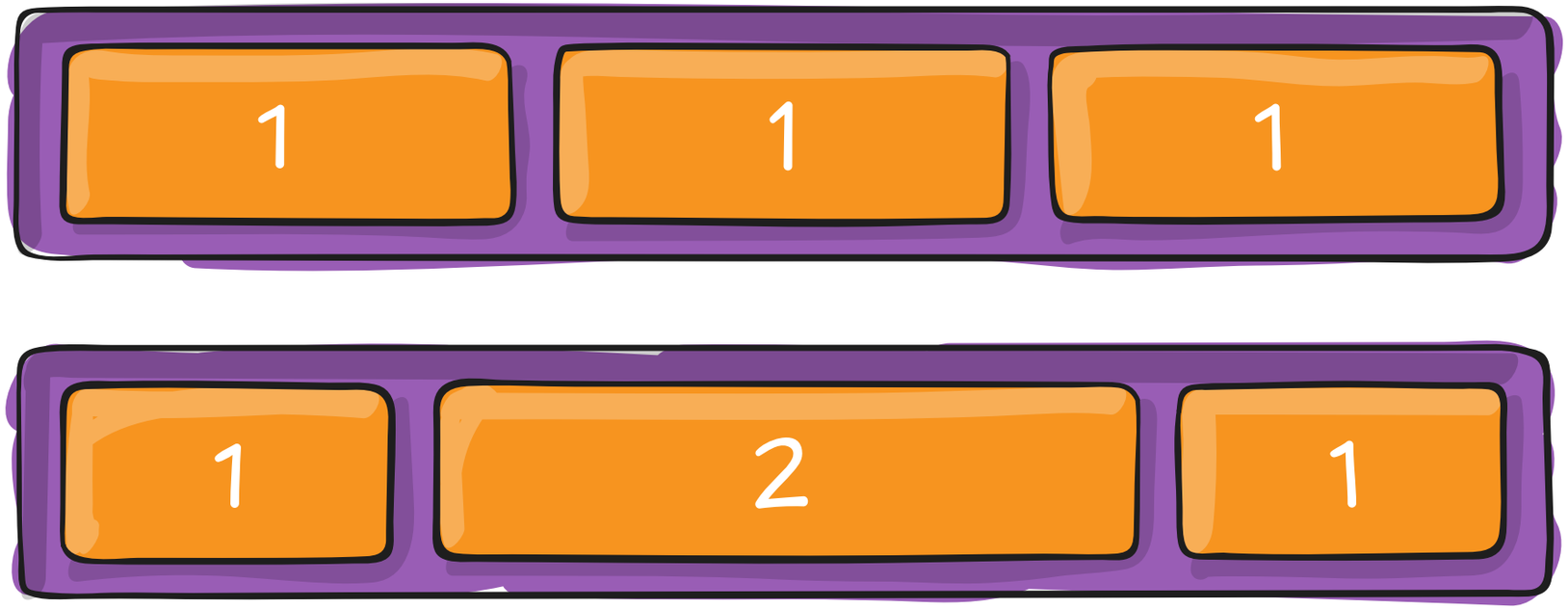


Flex – Propiedades de los items

- flex-basis
  - Define el tamaño base de los flex ítems antes de distribuirlos.
  - Se puede establecer un tamaño inicial (fijo, porcentaje, etc.)
  - Si no se especifica, es como si valiera "content", que asume el contenido como tamaño base.
- flex-grow / flex-shrink:
  - Define si un elemento debe ser del tamaño base o tiene que ser más grande / pequeño
  - Es un valor numérico (sin unidades) que se usa para calcular la proporción. Si es 2, usará el doble / la mitad de espacio que el resto, y así con el 3, 4, 5, etc.

Flex – Propiedades de los items

- flex-basis "content" con flex-grow:

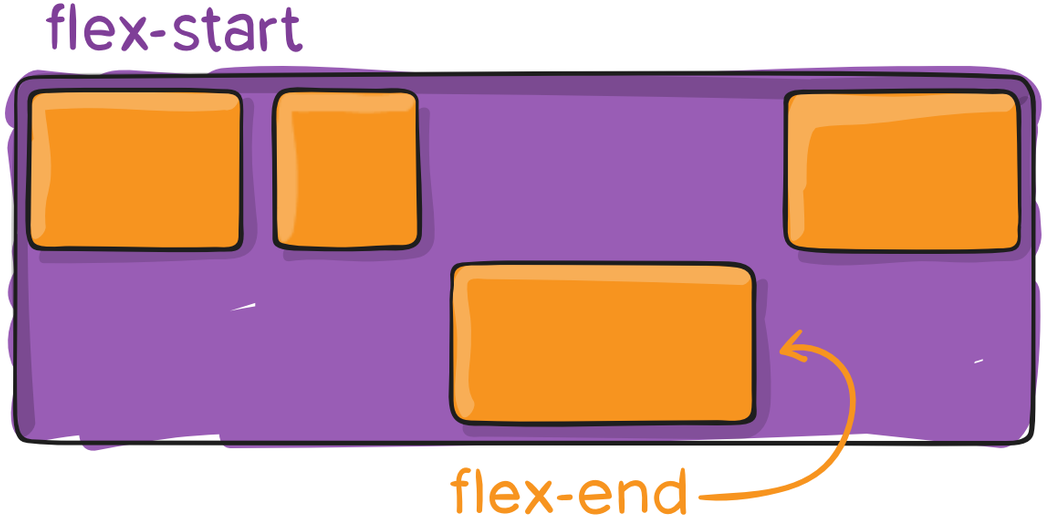


Flex – Propiedades de los items

- flex:
  - Atajo para las propiedades flex-grow, flex-shrink y flex basis.
  - La segunda y tercera (flex-shrink y flex basis) son opcionales.
  - Por defecto es “0 1 auto”.
  - Se recomienda usarlo en lugar de usarlas independientemente.

Flex – Propiedades de los items

- align-self:
  - auto, flex-start, flex-end, center , baseline, stretch
  - Sobrescribe la alineación establecida por “align-items“ en el contenedor.
  - Permite alinear un item de forma diferente al resto



Flex – Consideraciones finales

- CSS solo ve la jerarquía de contenedor-item
- No aplicará propiedades Flex a elementos que no estén directamente relacionados, es decir, elementos descendientes que estén dentro de los flex-items
- Para que las propiedades de los ítems funcionen, el contenedor debe tener propiedad display: flex / inline-flex.
- Las propiedades float, clear y vertical-align no tienen ningún efecto en flex-items.

Recursos y juego de práctica

- Referencia y tutoriales:
  https://lenguajecss.com/css/maquetacion-y-colocacion/flex/
  https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
  https://www.w3schools.com/csS/css3_flexbox.asp
  https://www.tutorialrepublic.com/css-tutorial/css3-flexible-box-layouts.php
- Práctica online:
  https://flexboxfroggy.com
  http://www.flexboxdefense.com
  https://codingfantasy.com/games/flexboxadventure/play
  https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox_skills