FLEXBOX

What's a flexbox?

FLEXBOX

Topics

- Main Concepts
- Flex Container Properties
- Flex Items

Flexbox

Well, technically, the flexible box module

Essentially, it's a tool in css that provides us a more efficient (easier) way to lay out, align, and distribute the space of our items in a container.

In more technical, MDN terms, Flexbox is a one-dimensional layout method for laying out items in rows or columns. Items flex to fill additional space and shrink to fit into smaller spaces. This article explains all the fundamentals.

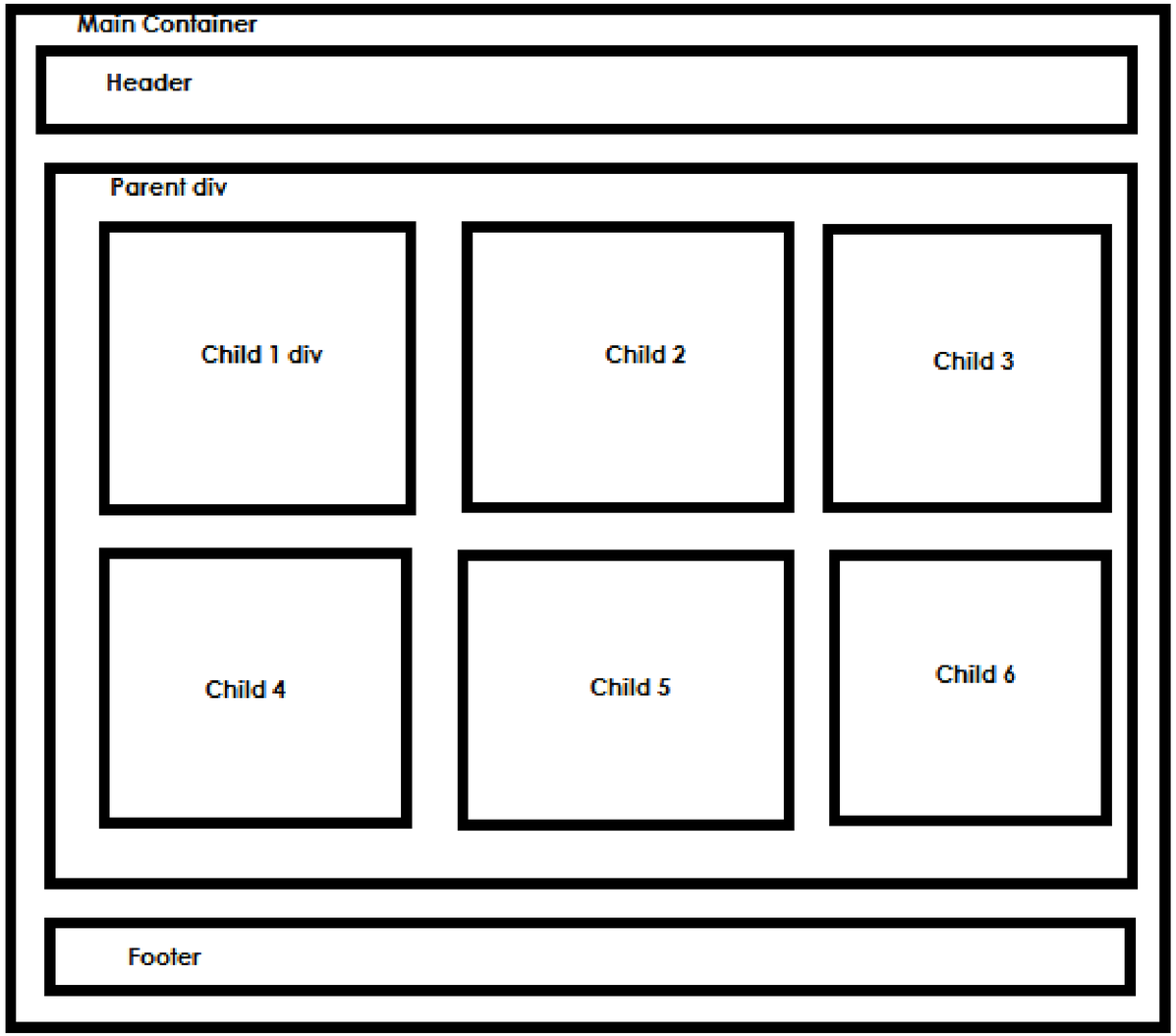
Why Flexbox?

- It provides an easy and clean way to arrange items withing a container - i.e. create layouts.
- NO FLOATS
- Responsive and Mobile Friendly
- Positioning child elements is easier.
- Ability to order elements without editing source HTML.
- And more...

The Flexbox Concept

- Ability to alter item width/height to fit it's containers available space
- Direction Agnostic
- Built for 1 dimensional / small scale layouts

CSS BOXES

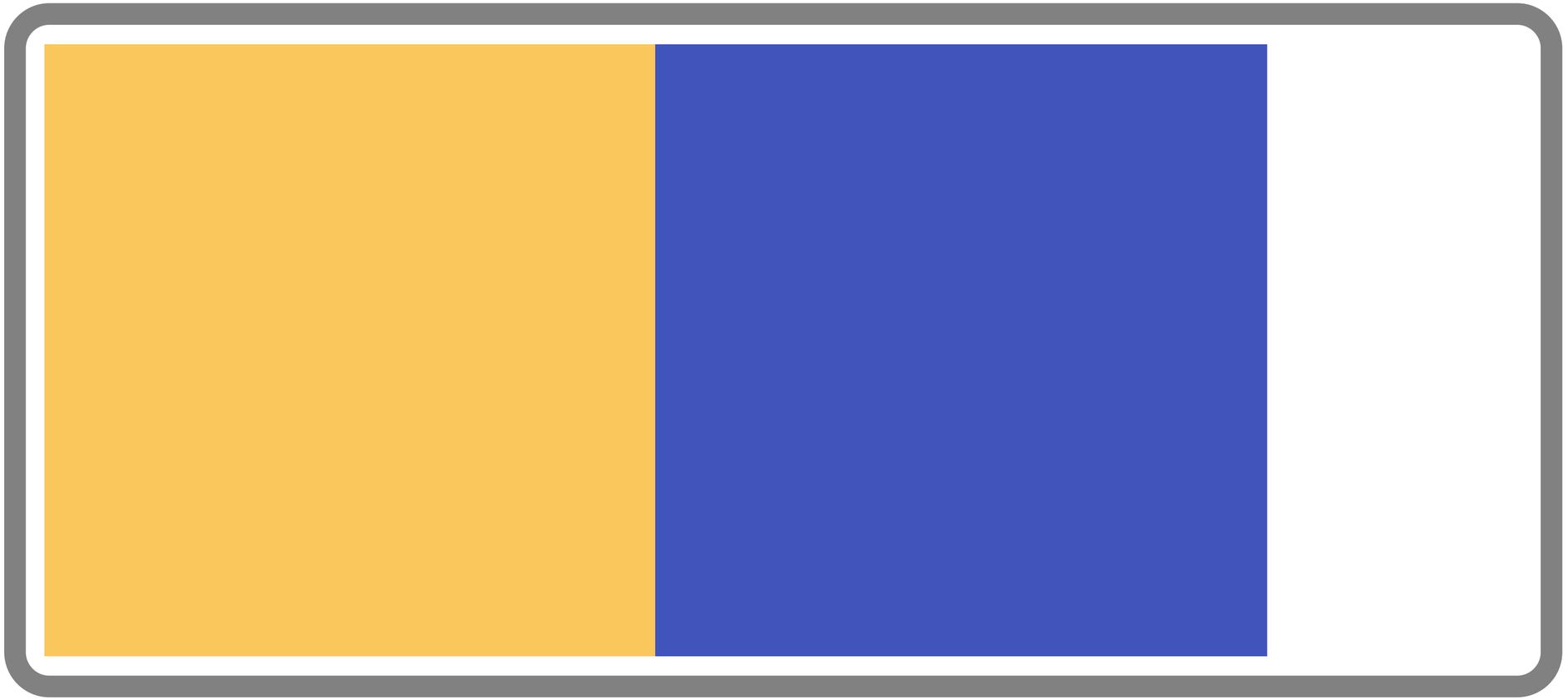


On our site, all of our content lives in boxes.

Every Element in CSS has a box around it. Be it a span, h1, or p element.

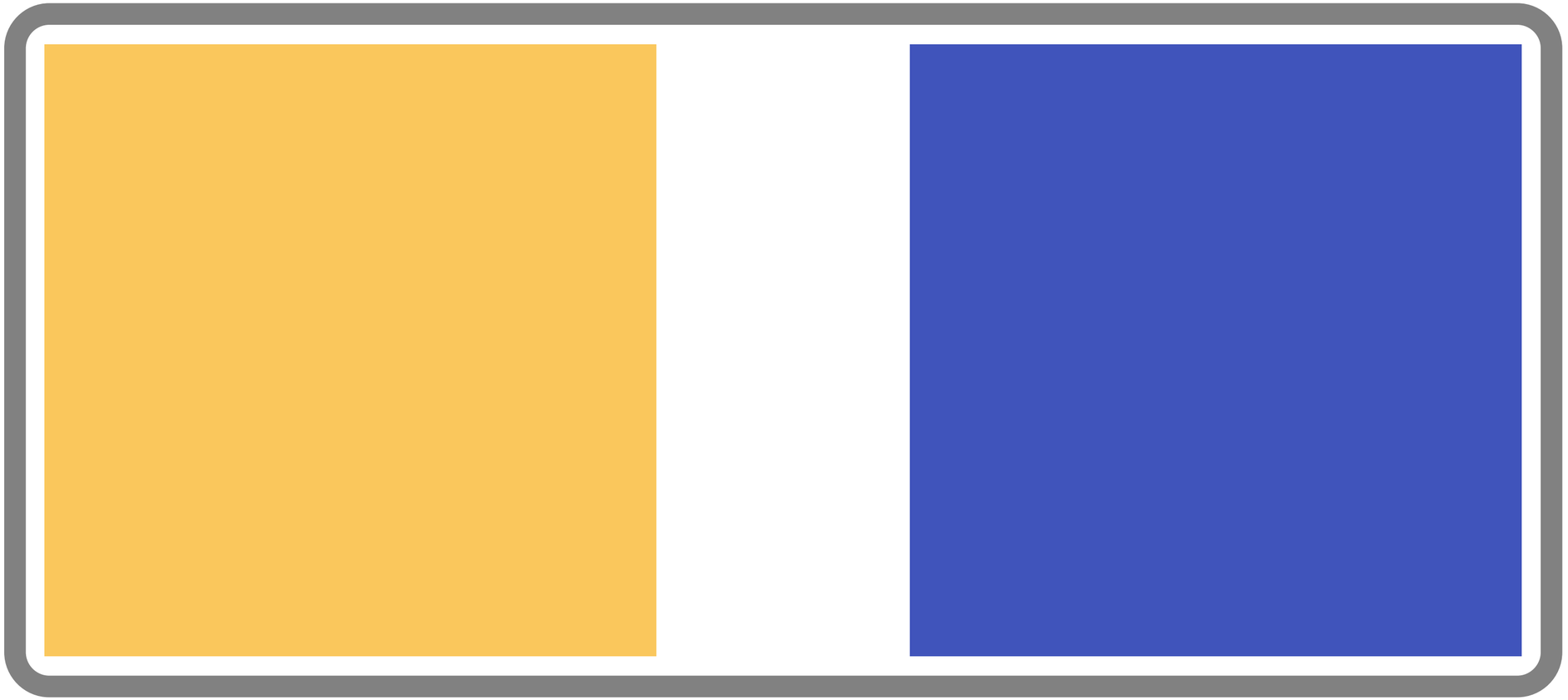
Flex Container

Row



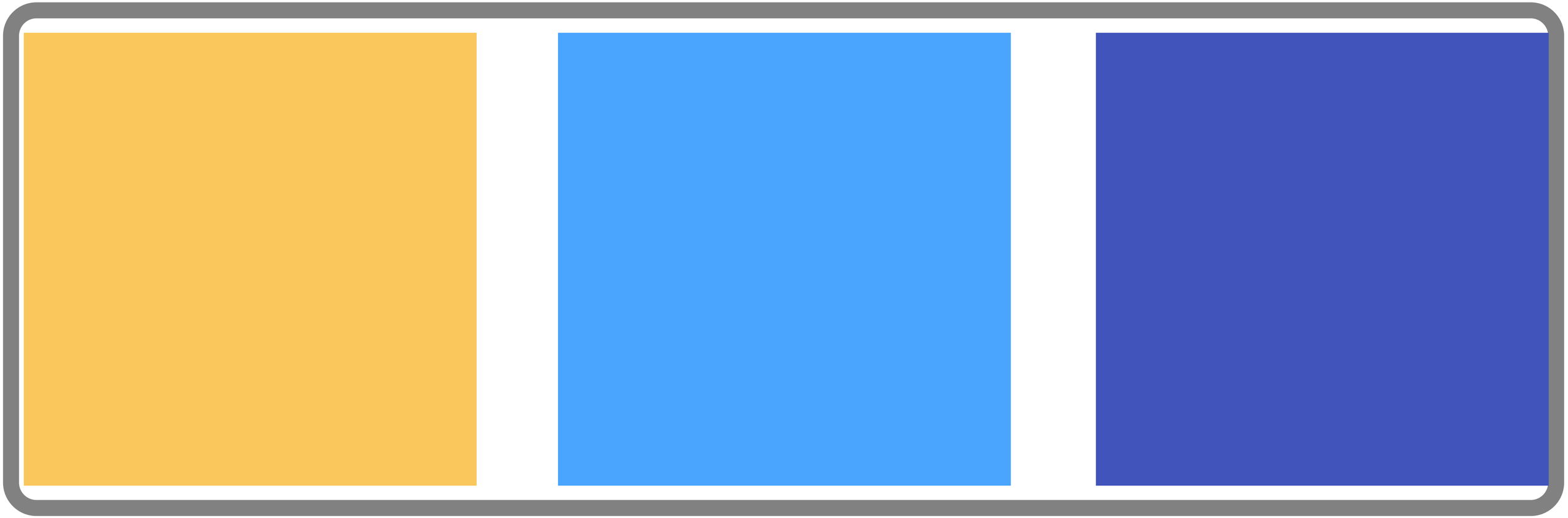
Flex Container

Row



Flex Container

Row



Flex Container

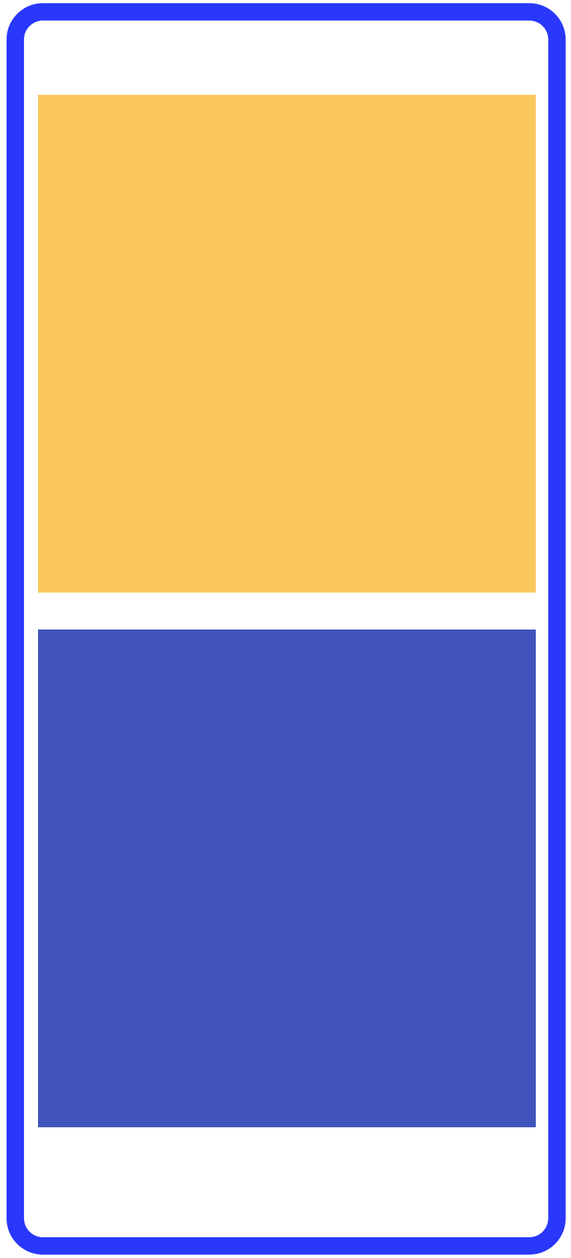
Row

Flex Container

Row

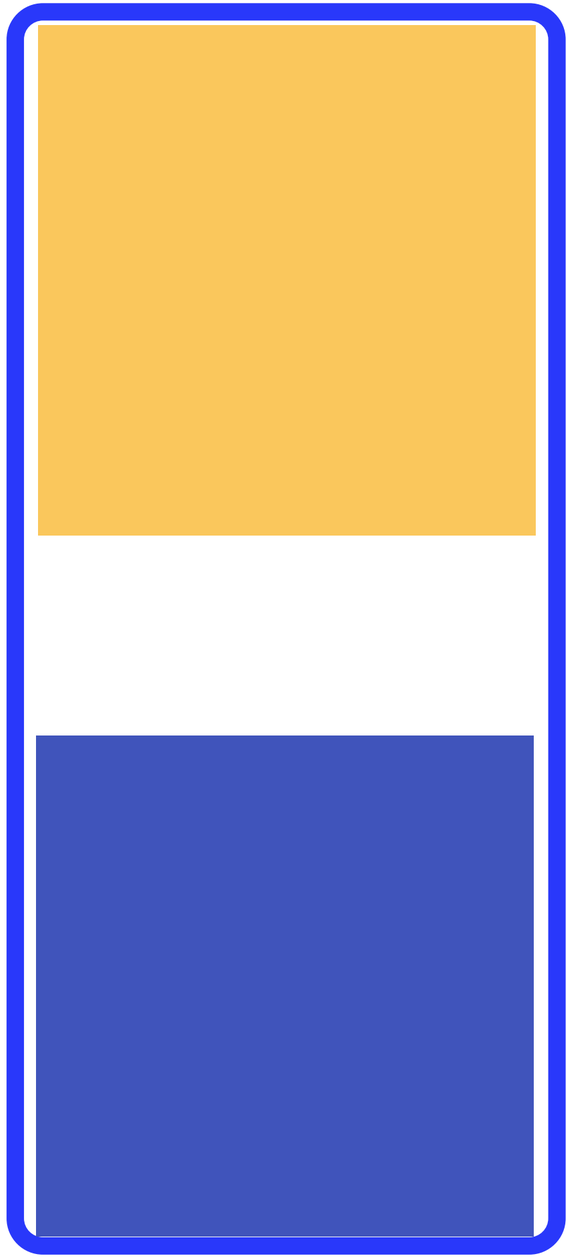
Flex Container

Column



Flex Container

Column

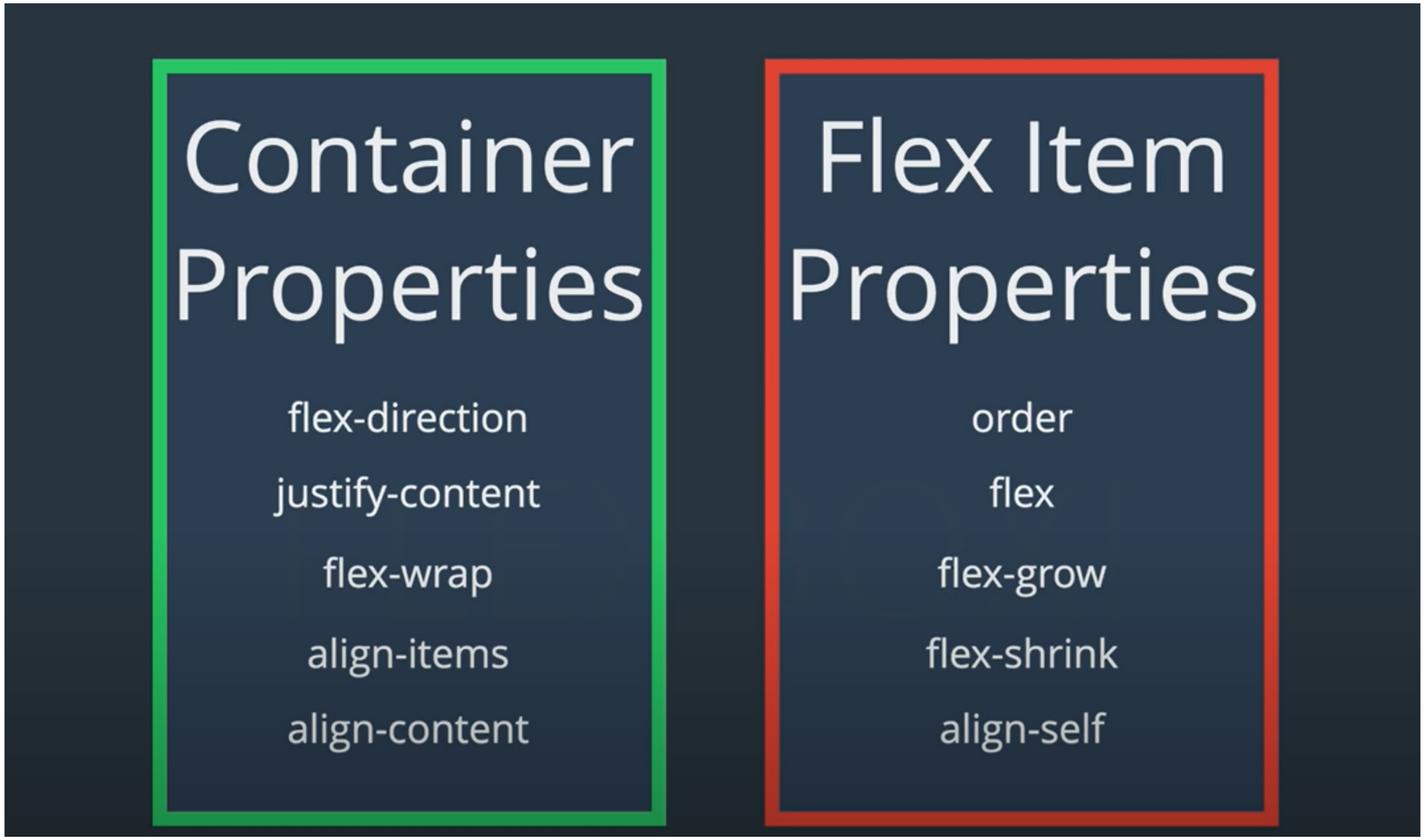


And all you need is:

Display: flex;

Well, there is some other stuff...

Flex Container & Item Properties



Main Axis

- **main axis** – The main axis of a flex container is the primary axis along which flex items are laid out. Beware, it is not necessarily horizontal; it depends on the flex-direction property (see below).

- **main-start | main-end** – The flex items are placed within the container starting from main-start and going to main-end.

- **main size** – A flex item's width or height, whichever is in the main dimension, is the item's main size. The flex item's main size property is either the 'width' or 'height' property, whichever is in the main dimension.

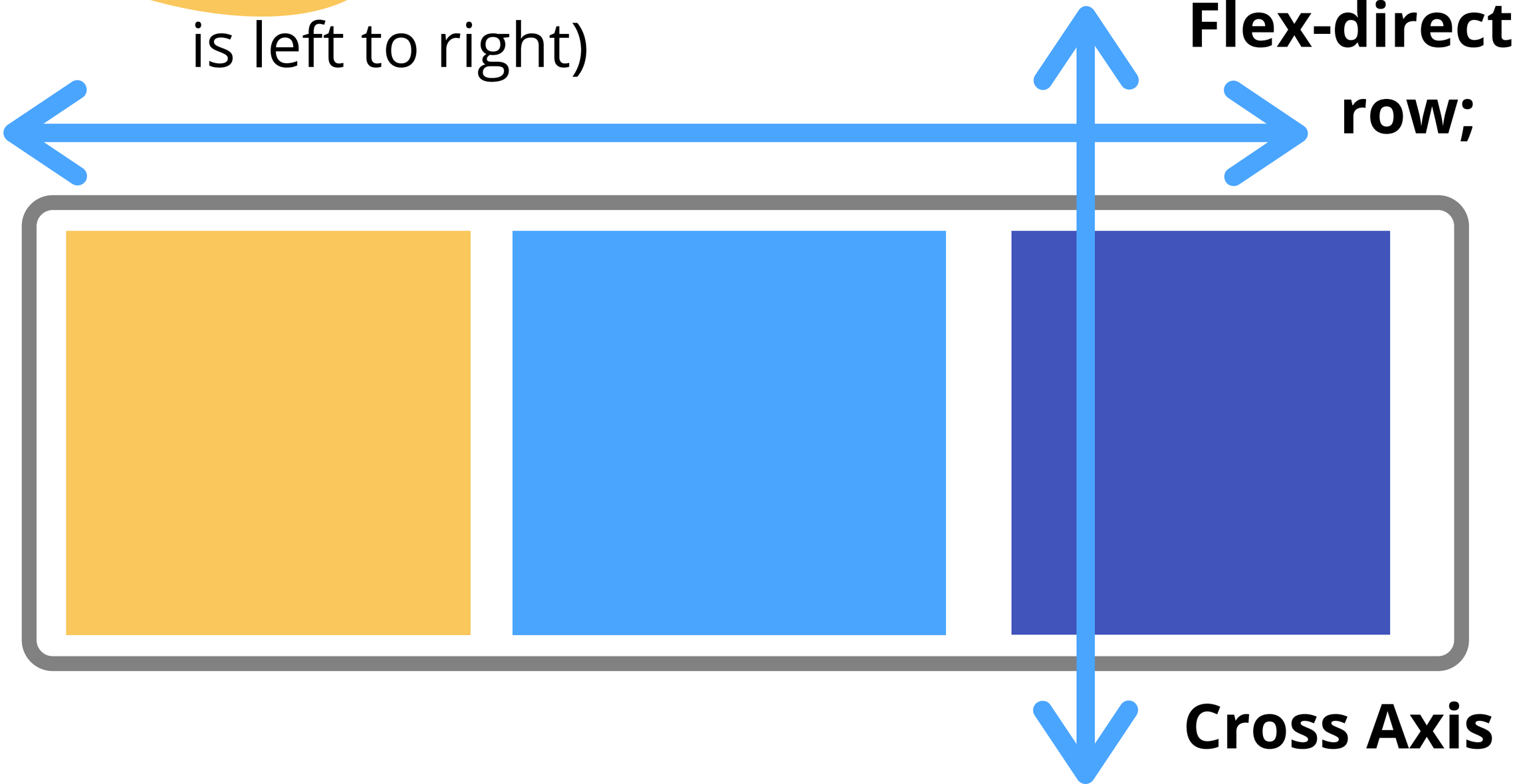
Cross Axis

- **cross axis** – The axis perpendicular to the main axis is called the cross axis. Its direction depends on the main axis direction.
-
- **cross-start | cross-end** – Flex lines are filled with items and placed into the container starting on the cross-start side of the flex container and going toward the cross-end side.
-
- **cross size** – The width or height of a flex item, whichever is in the cross dimension, is the item's cross size. The cross size property is whichever of 'width' or 'height' that is in the cross dimension.

Flex Container

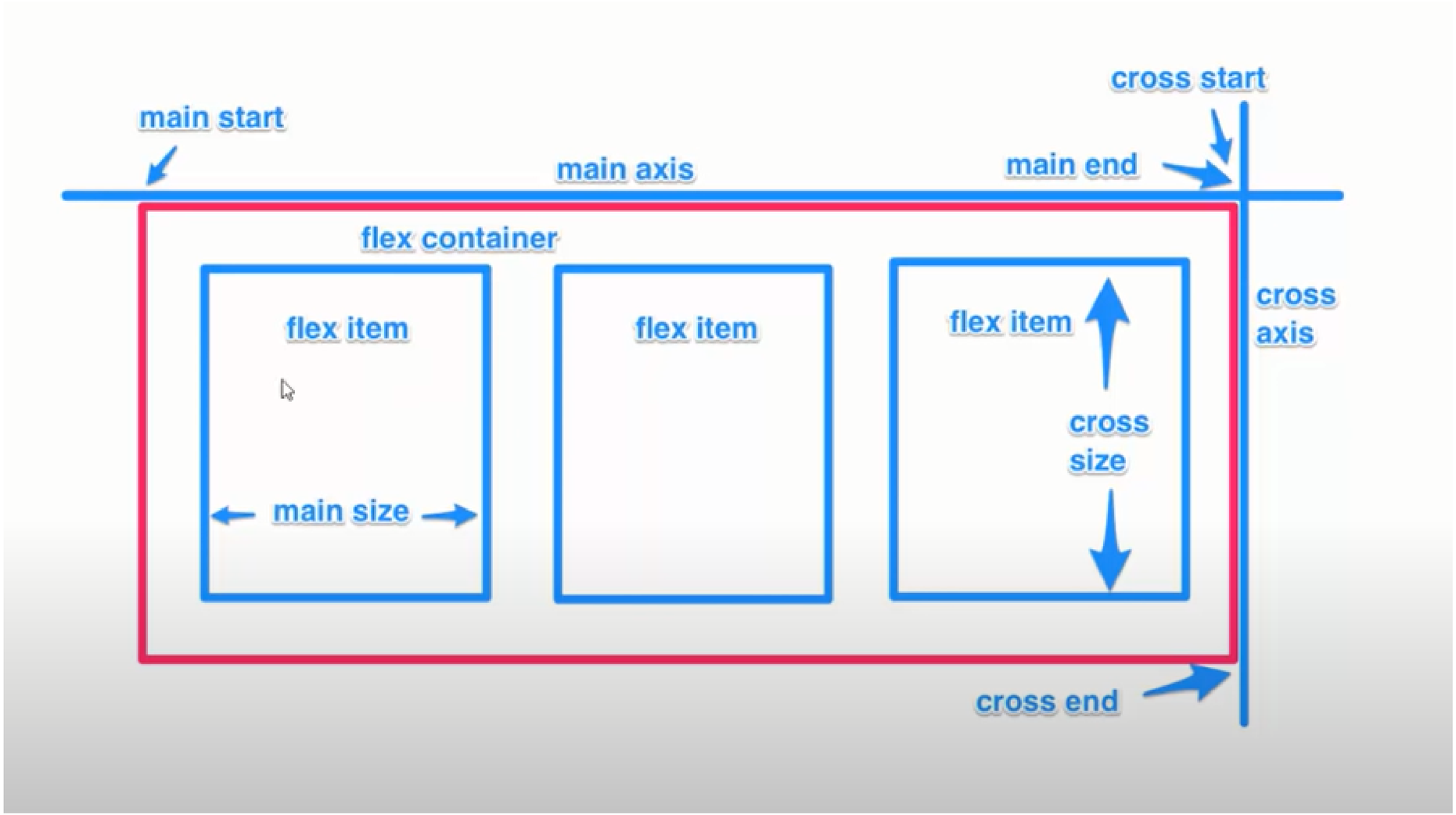
Main Axis (by default is left to right)

Flex-direction: row;



Cross Axis

Main/Cross Axis



Container Properites

Container Properties:

display: flexbox | inline-flex;

flex-direction: row | row-reverse | column | column-reverse;

flex-wrap: nowrap | wrap | wrap-reverse;

flex-flow: <‘flex-direction’> || <‘flex-wrap’>

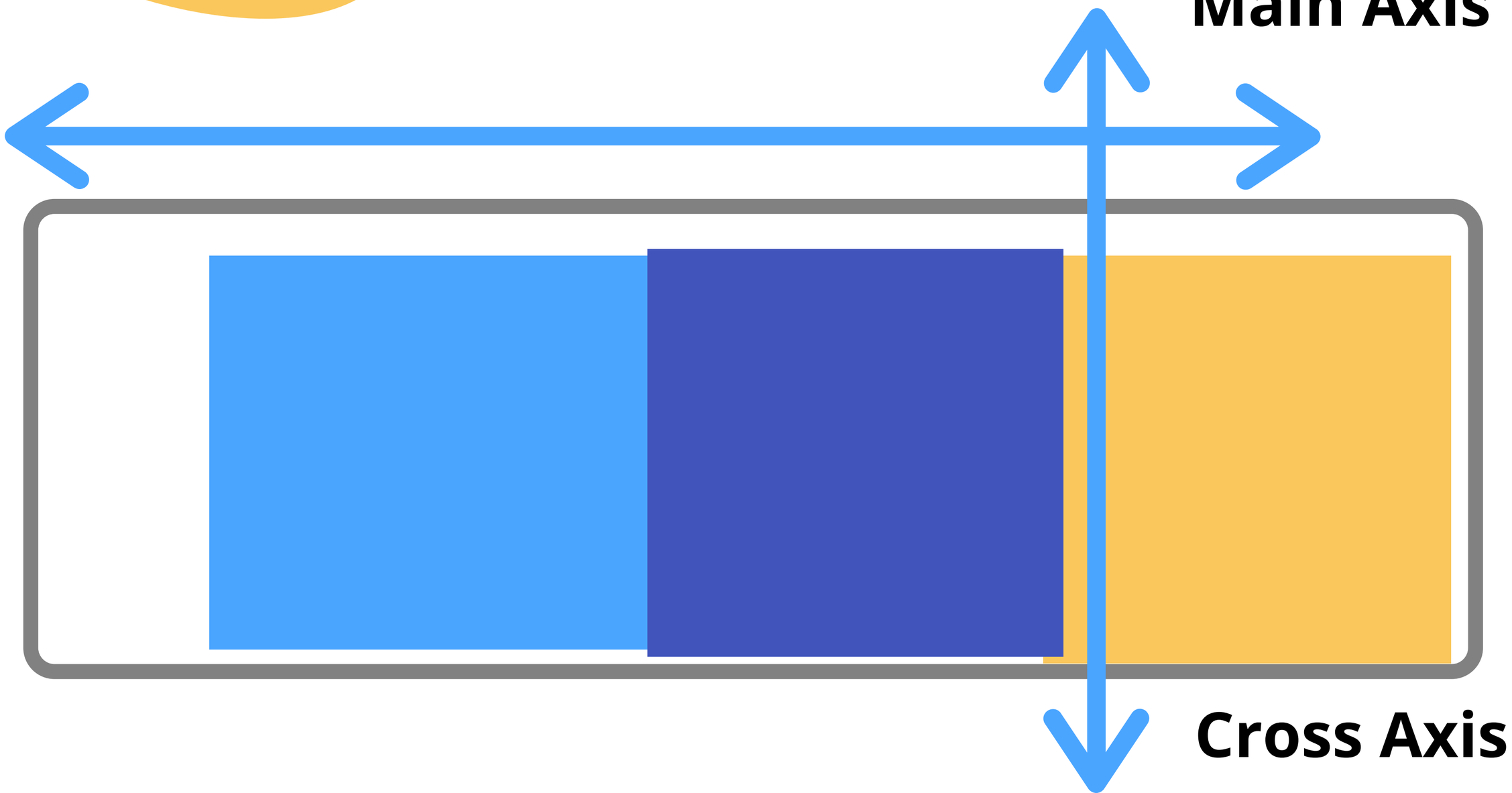
justify-content: flex-start | flex-end | center | space-between | space-around;

align-items: flex-start | flex-end | center | baseline | stretch;

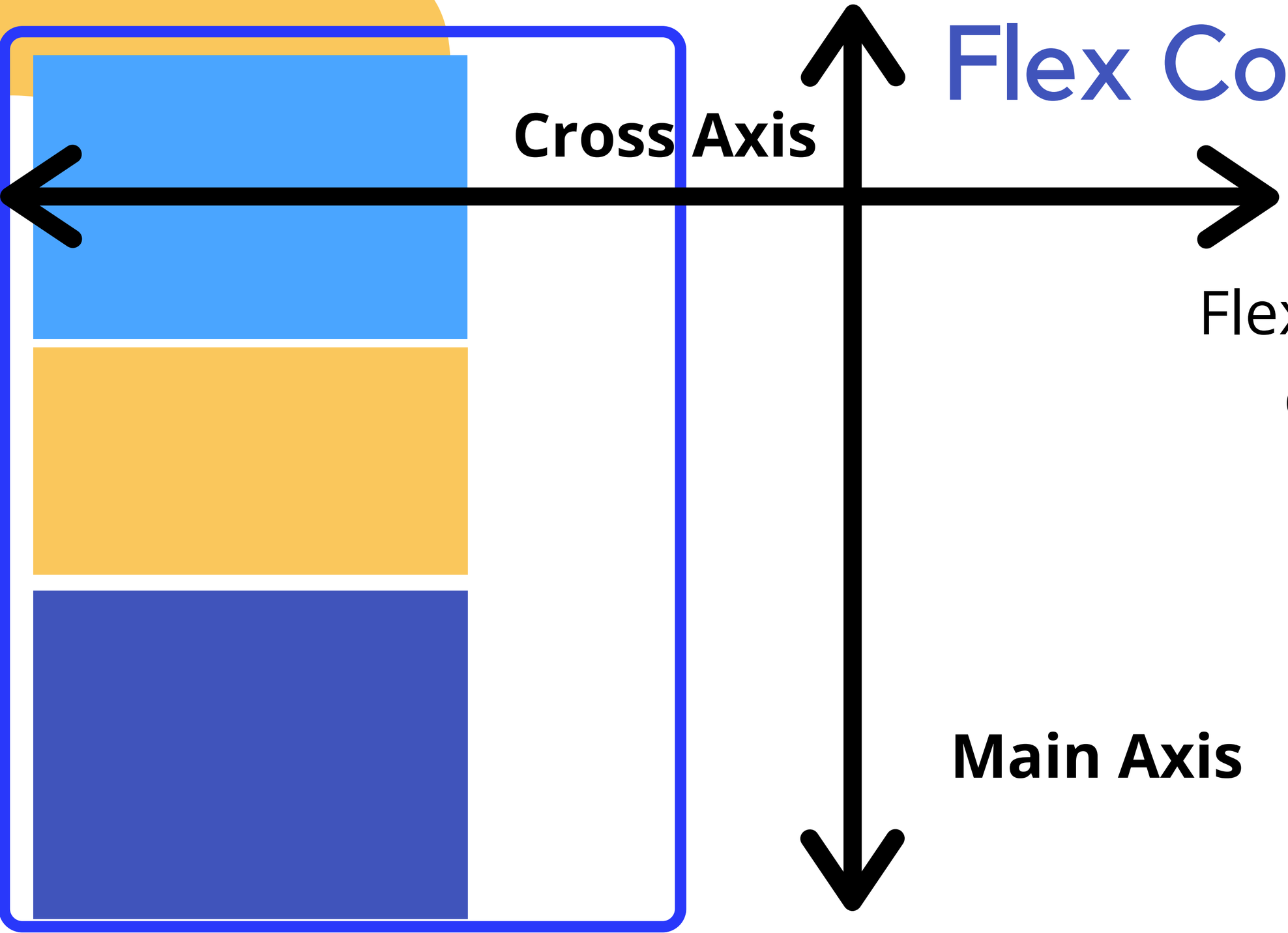
align-content: flex-start | flex-end | center | space-between | space-around | stretch;

Flex Container

Flex-direction: row-reverse;

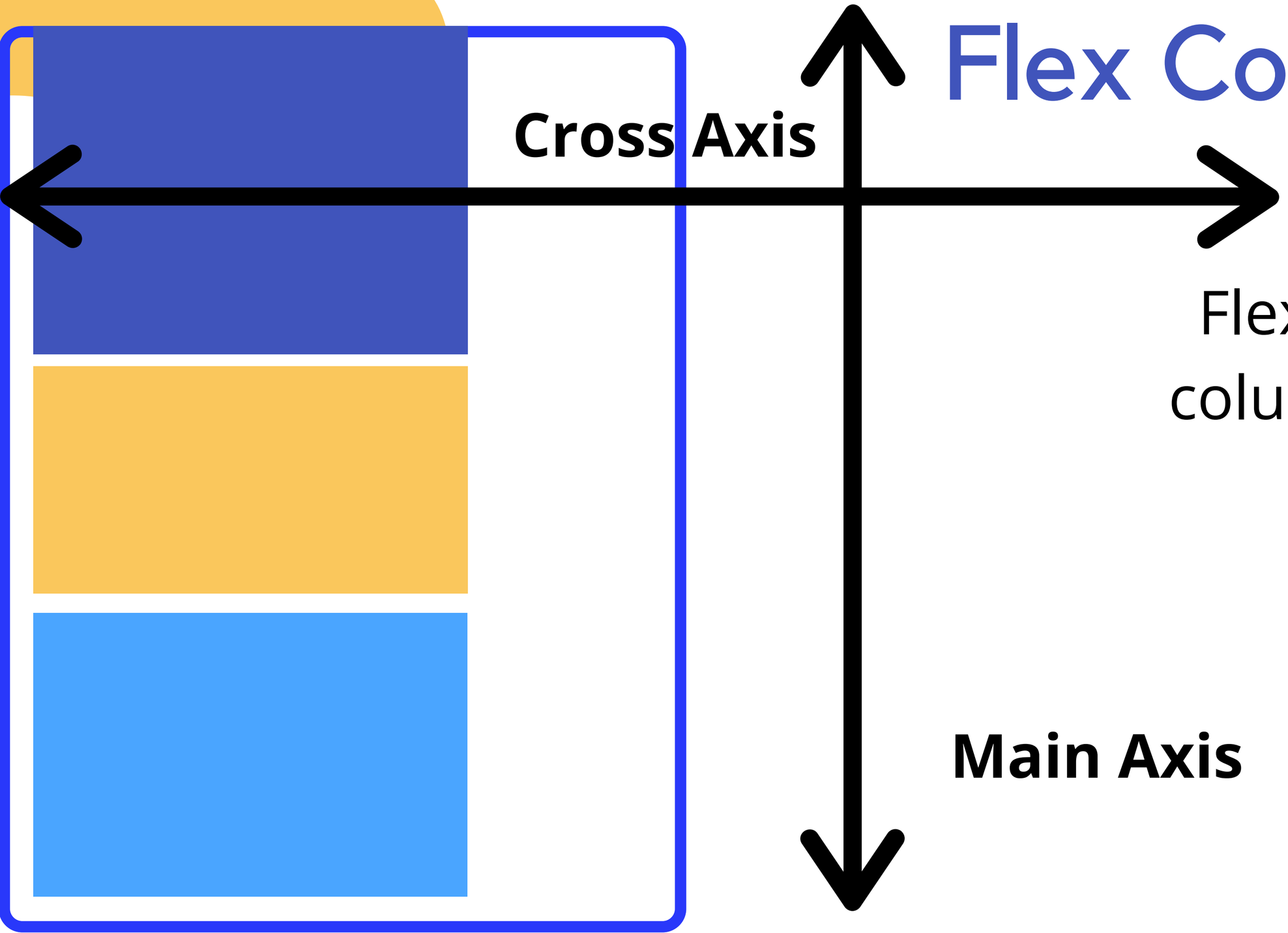


Flex Container



Flex-direction:
column;

Flex Container



Flex-direction:
column-reverse;

Flex-Wrap

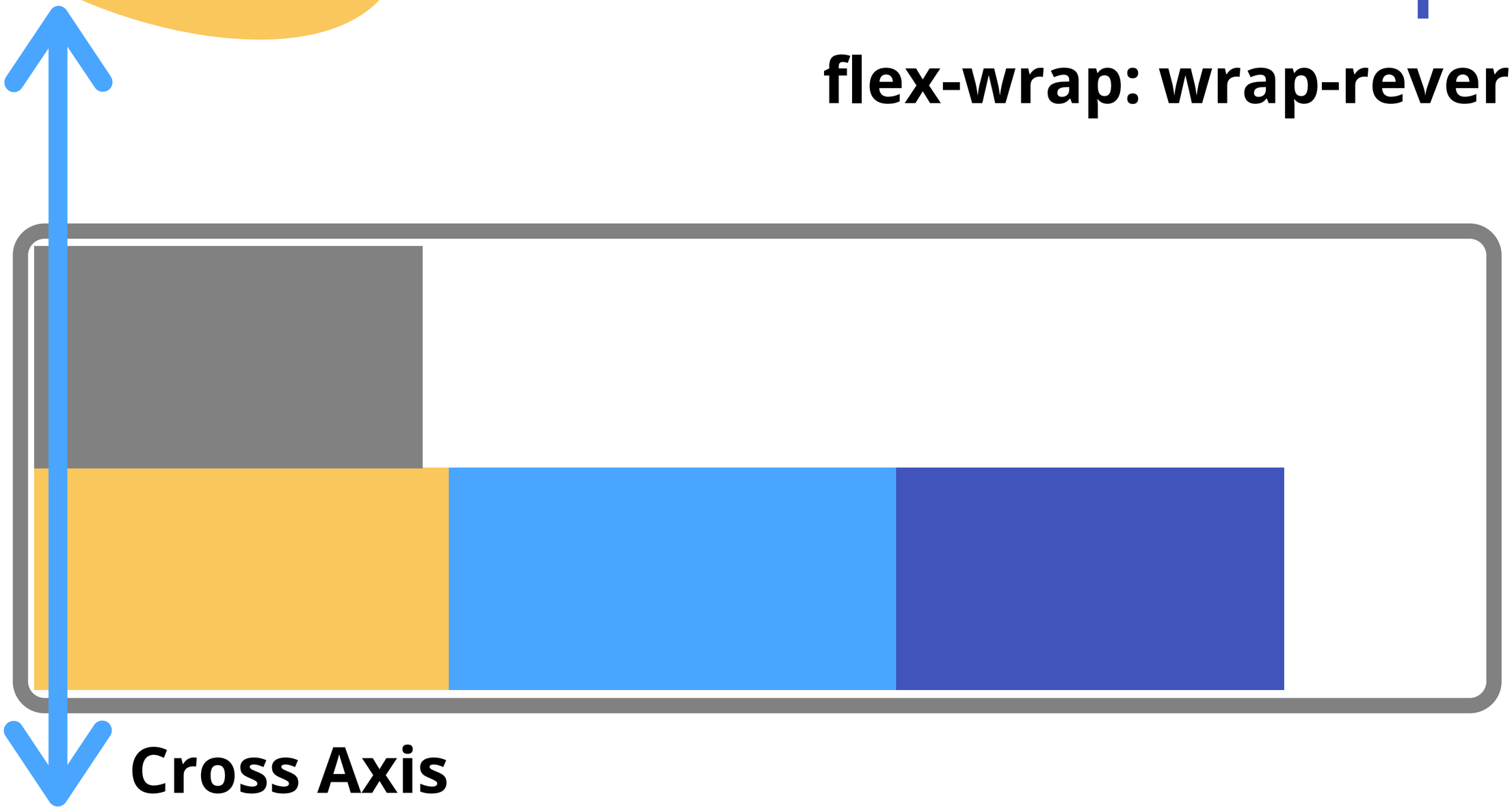
flex-wrap: wrap > nowrap > wrap-reverse;

This is how we can specify whether items should be forced onto a single line or wrapped into multiple lines;

Flex Wrap

flex-wrap: wrap-reverse;

wrap-reverse will reverse the start/end of the cross axis



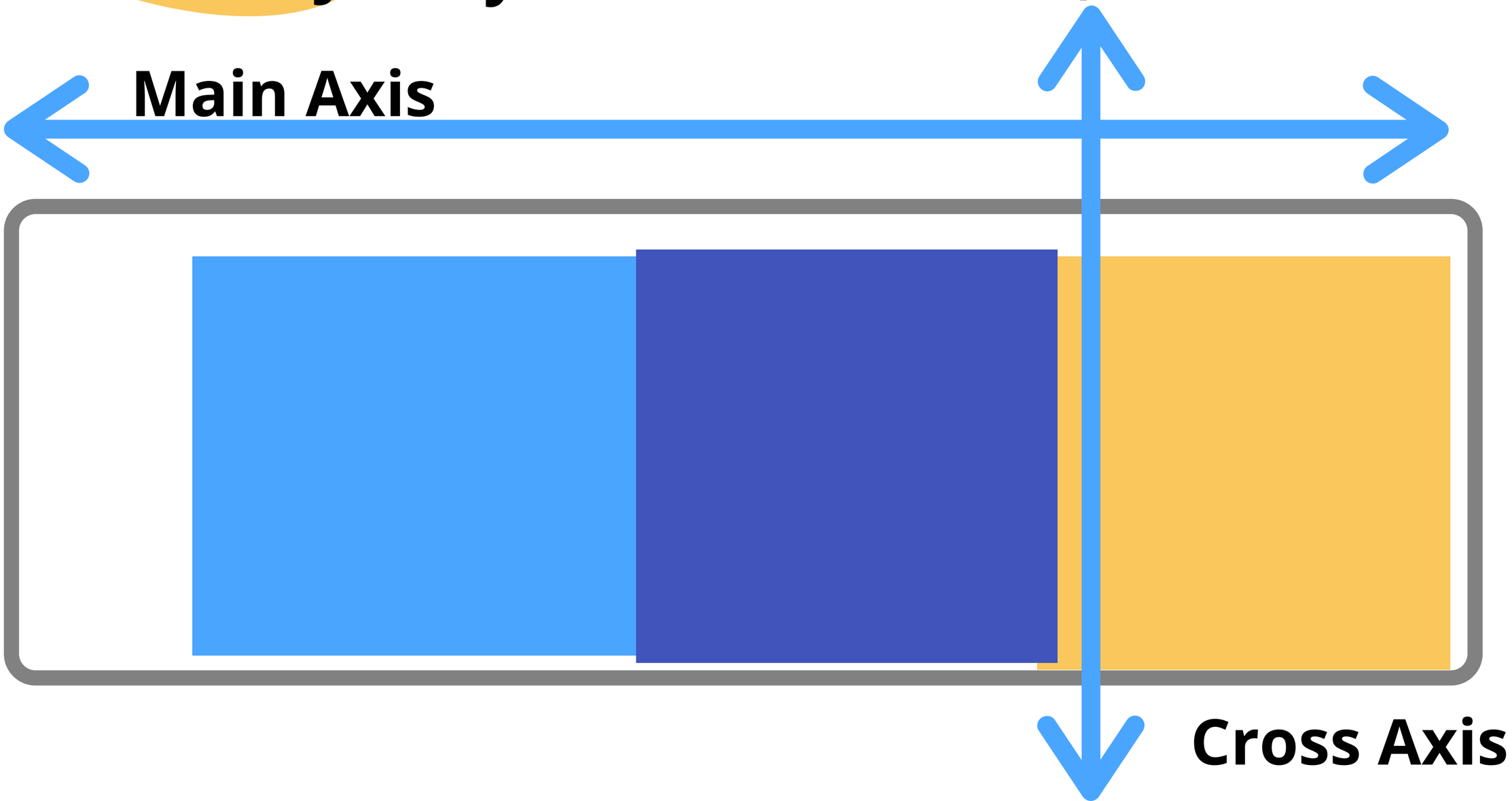
Justify-content

Justify-content: flex-start > flex-end > center > space-between > space-around;

We can use this to define how much space is distributed between items in a flex container **along the main axis**

Justify-Content

Justify-content: flex-end;



Item Properties

order: <integer>;

flex-grow: <number>; /* default 0 */

flex-shrink: <number>; /* default 1 */

flex-basis: <length> | auto; /* default auto */

flex: none | [<'flex-grow'> <'flex-shrink'>? || <'flex-basis'>]

align-self: auto | flex-start | flex-end | center | baseline | stretch;

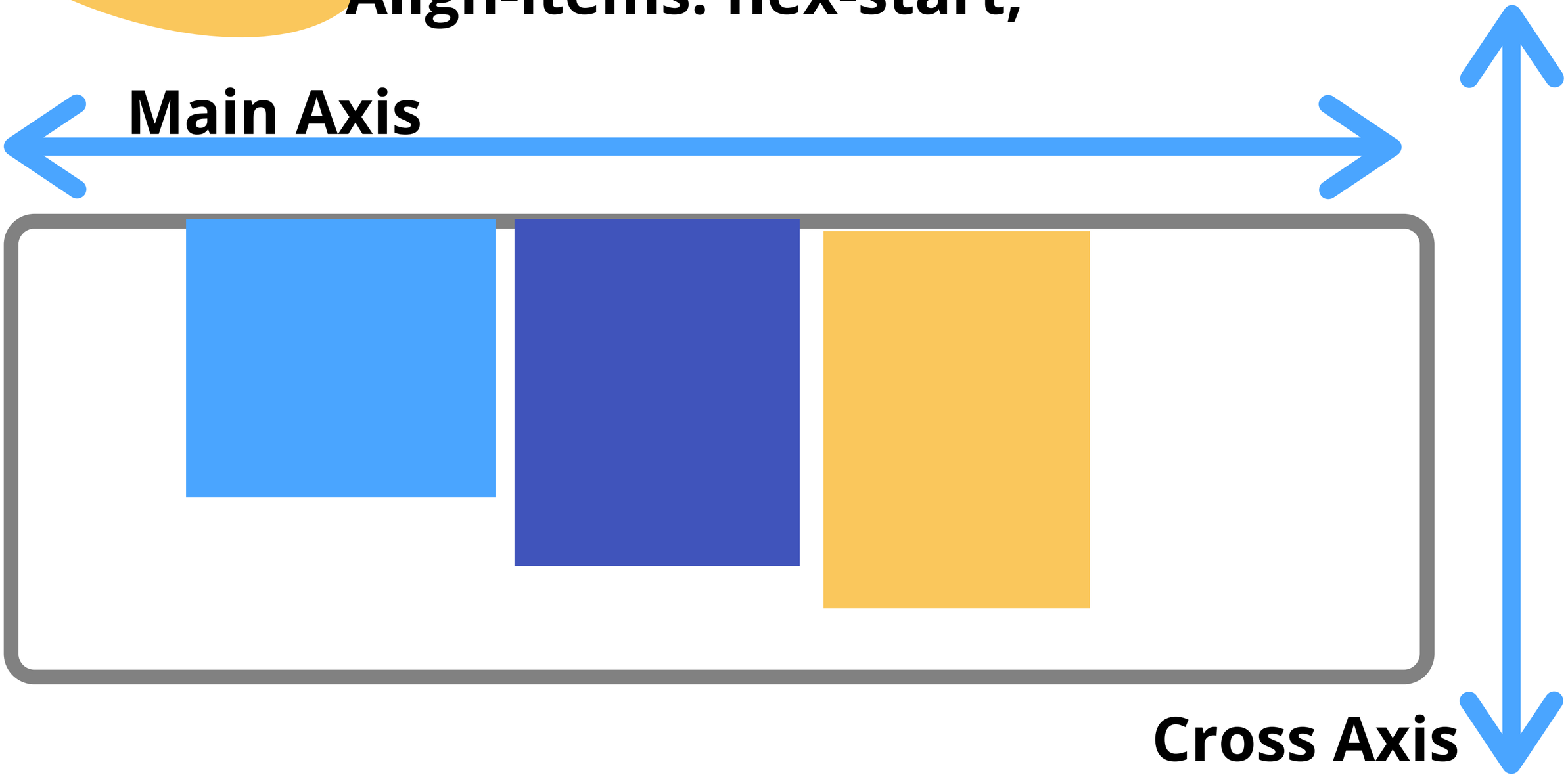
Align-Items

align-items: flex-start > flex-end > center > stretch > baseline

Defines how much space is distributed between items in a flex container **along the cross axis**

Align-items

Align-items: flex-start;



Align-Content

align-content: flex-start > flex-end > center > stretch > baseline

Defines how space is distributed **BETWEEN ROWS** in a flex container along the **CROSS AXIS**

Align-Self

align-content: flex-start > flex-end > center > stretch > baseline

Defines how space is distributed **BETWEEN ROWS** in a flex container along the **CROSS AXIS**

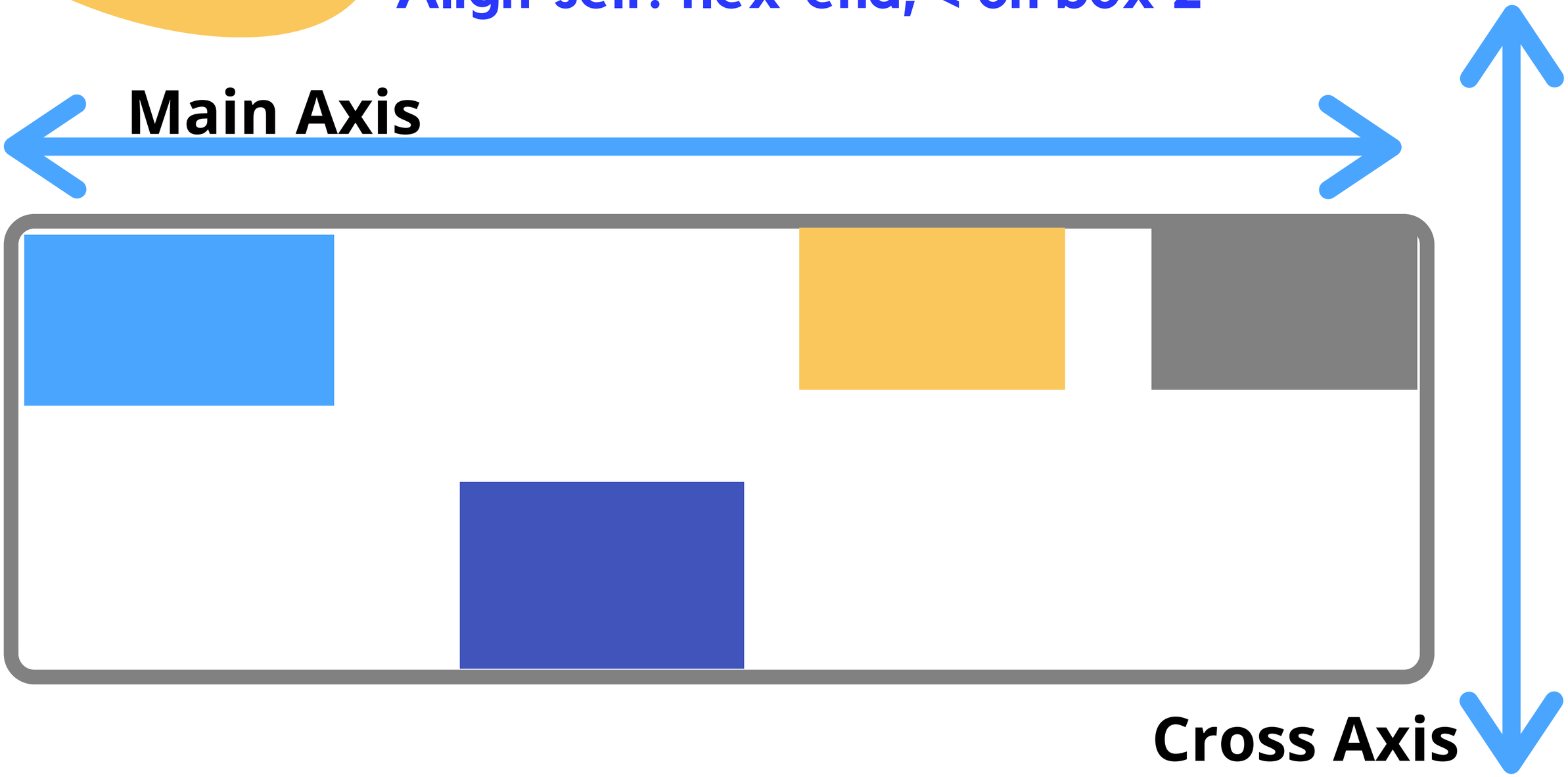
Item Properties

Align-Self

Allows you to override align-items on individual flex items

Align-items: flex-start; < on Container

Align-self: flex-end; < on box 2



Flex

Defines how a flex item will grow or shrink to fit the available space in a container.

(shorthand for 3 properties, actually)

flex-grow

flex-shrink

flex-basis

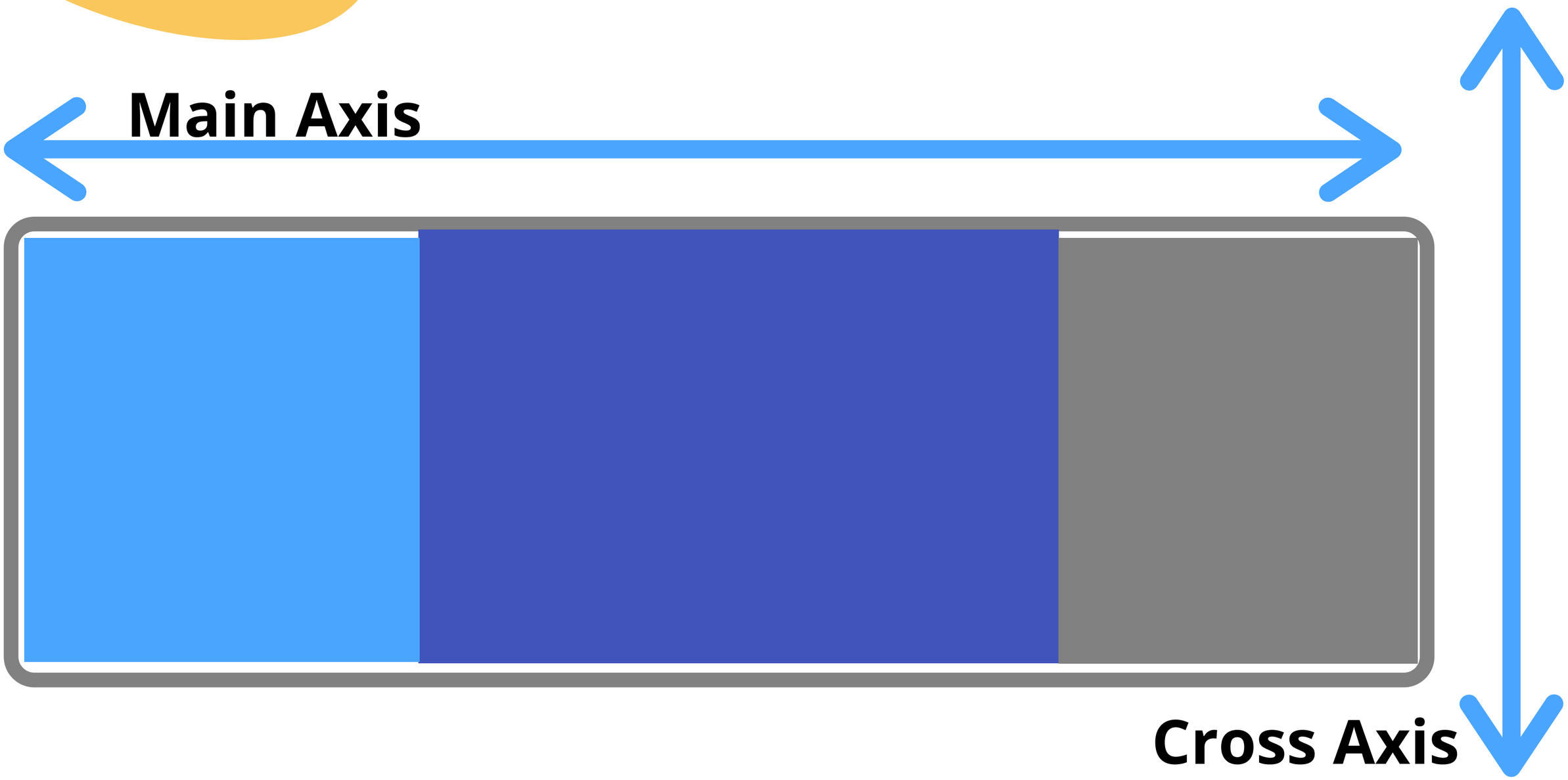
Flex Basis

Kind of like width but not quite

Specifies the ideal size of a flex item before it's placed into a flex container.

Order

Specifiese the order used to lay out items in their container


Main Axis
Cross Axis