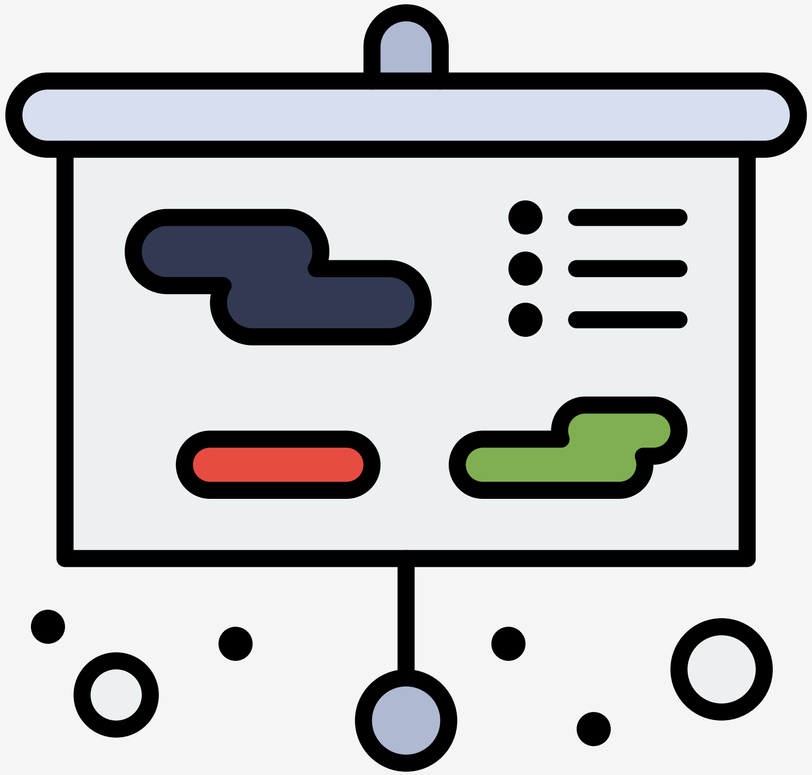
RESPONSIVE
WEB DESIGN II

Lecture 5

TODAY'S TOPICS



- Flexbox
- Exercise: Flexblocks

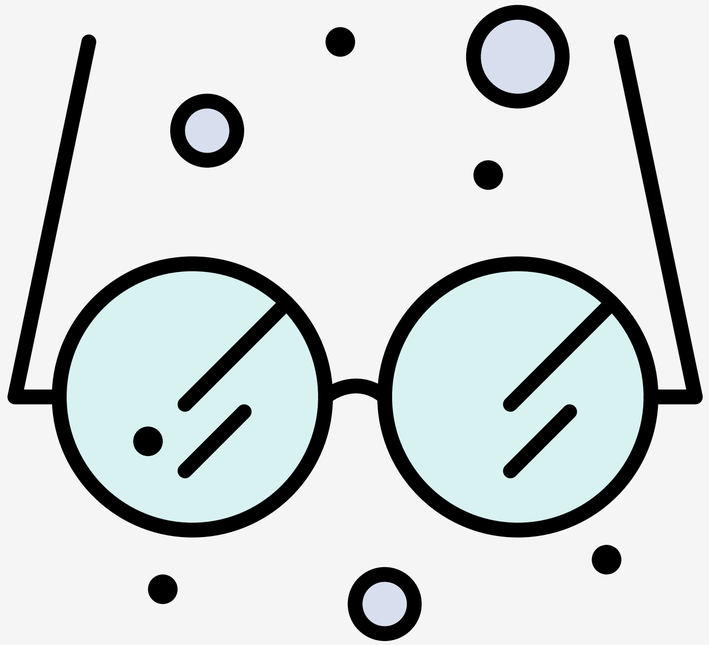
ANNOUNCEMENTS

- Sign-in Sheet
- Recordings

QUESTIONS

FLEXBOX

HISTORY WEB DESIGN LAYOUTS

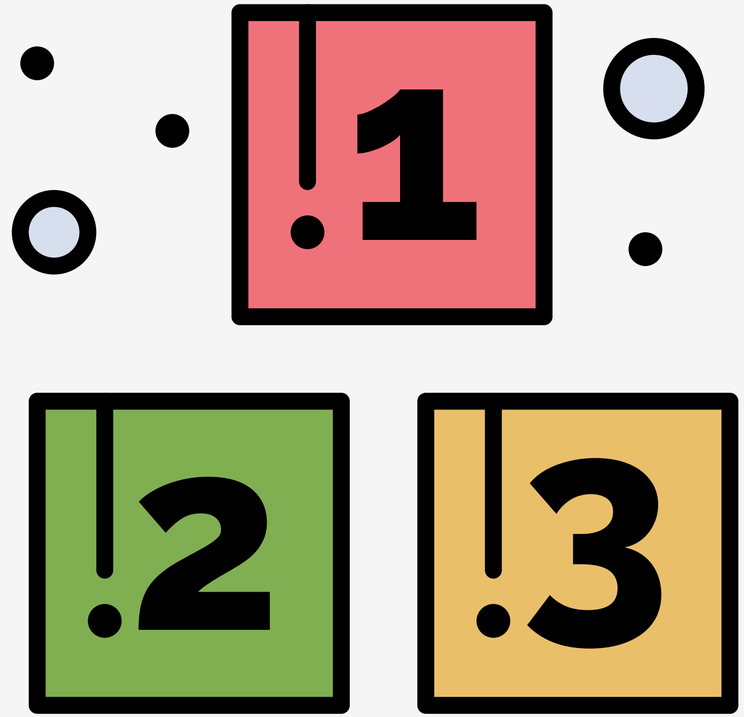


- No Layout (1990 - 1995)
- Table Layouts (1995 - 2004)
- Tableless Layouts (2001 - 2012)
- Responsive Layouts (2010 - Today)
- Grid Layouts (2011 - Today)
- Liquid Layouts (2014 - Today)

DISPLAY

- Sets the display type of an element
- Two categories: Outer and Inner
- Outer display types affect the element itself (e.g. `inline`, `block`)
- Inner display types affect how the element's children (e.g. `flex`, `grid`)

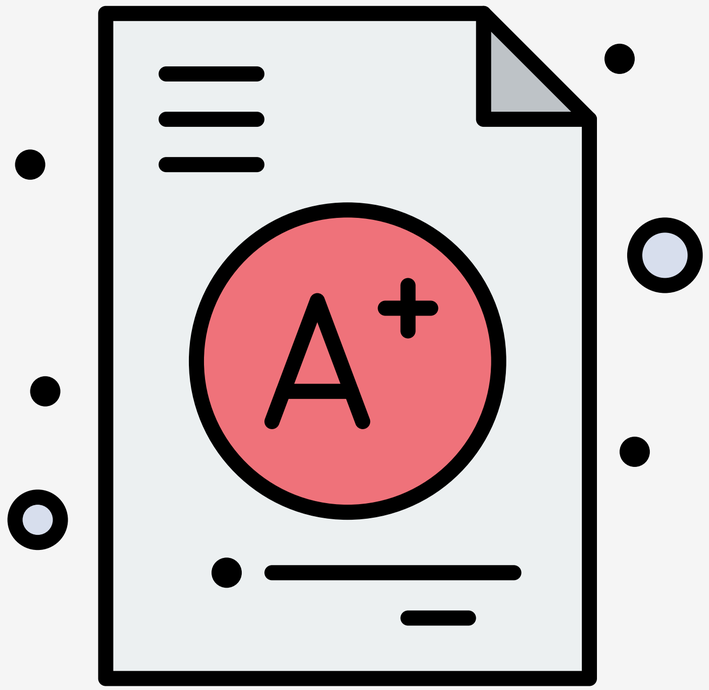
FLEXBOX



- Flexbox defines a box model optimized for layouts in one dimension
- The `display: flex` rule creates flexbox container
- Elements directly inside a flexbox container are affected

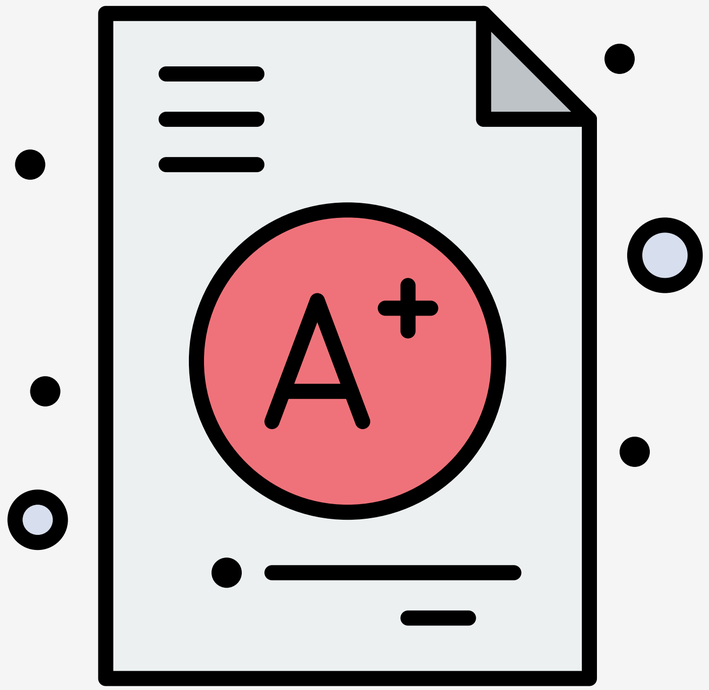
FLEXBOX CONTAINER

FLEXBOX CONTAINER

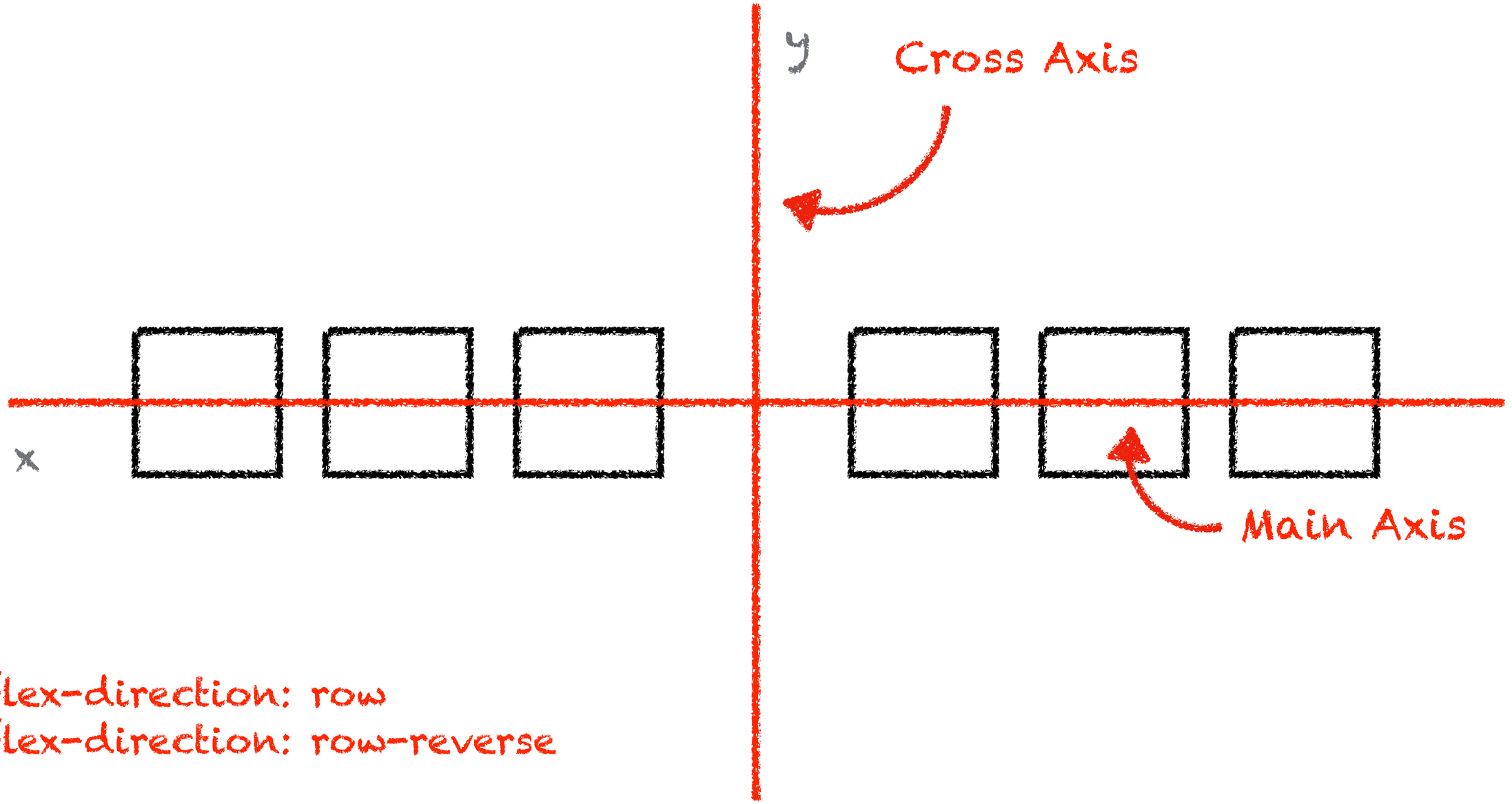


- Works with two axis: main and cross
- The main axis which direction the children will flow and is defined by `flex-direction`
- The cross axis is alway perpendicular to the main axis

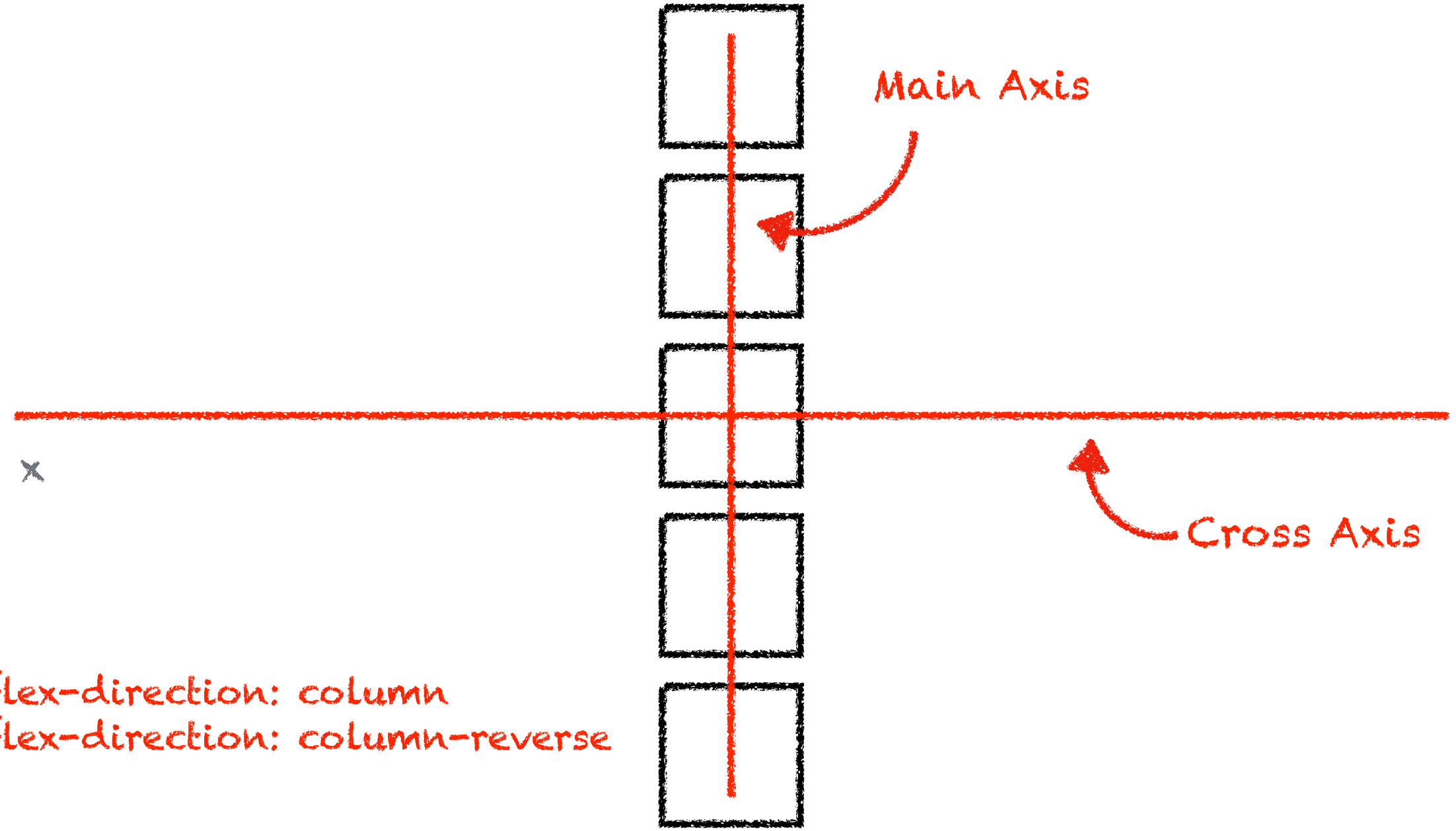
FLEX-DIRECTION



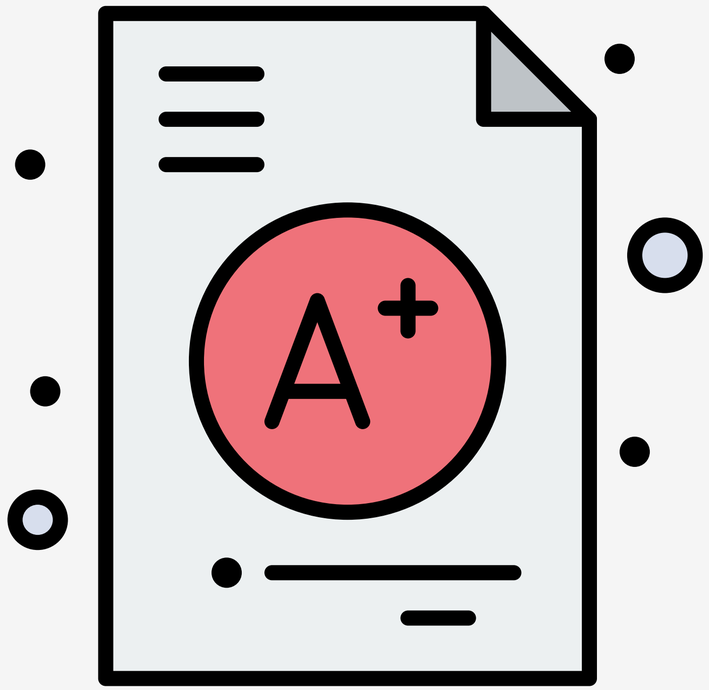
- The `flex-direction` property defines the main axis
- The children of the flexbox container flow along the main axis
- The `flex-direction` values:
  - `row`
  - `row-reverse`
  - `column`
  - `column-reverse`



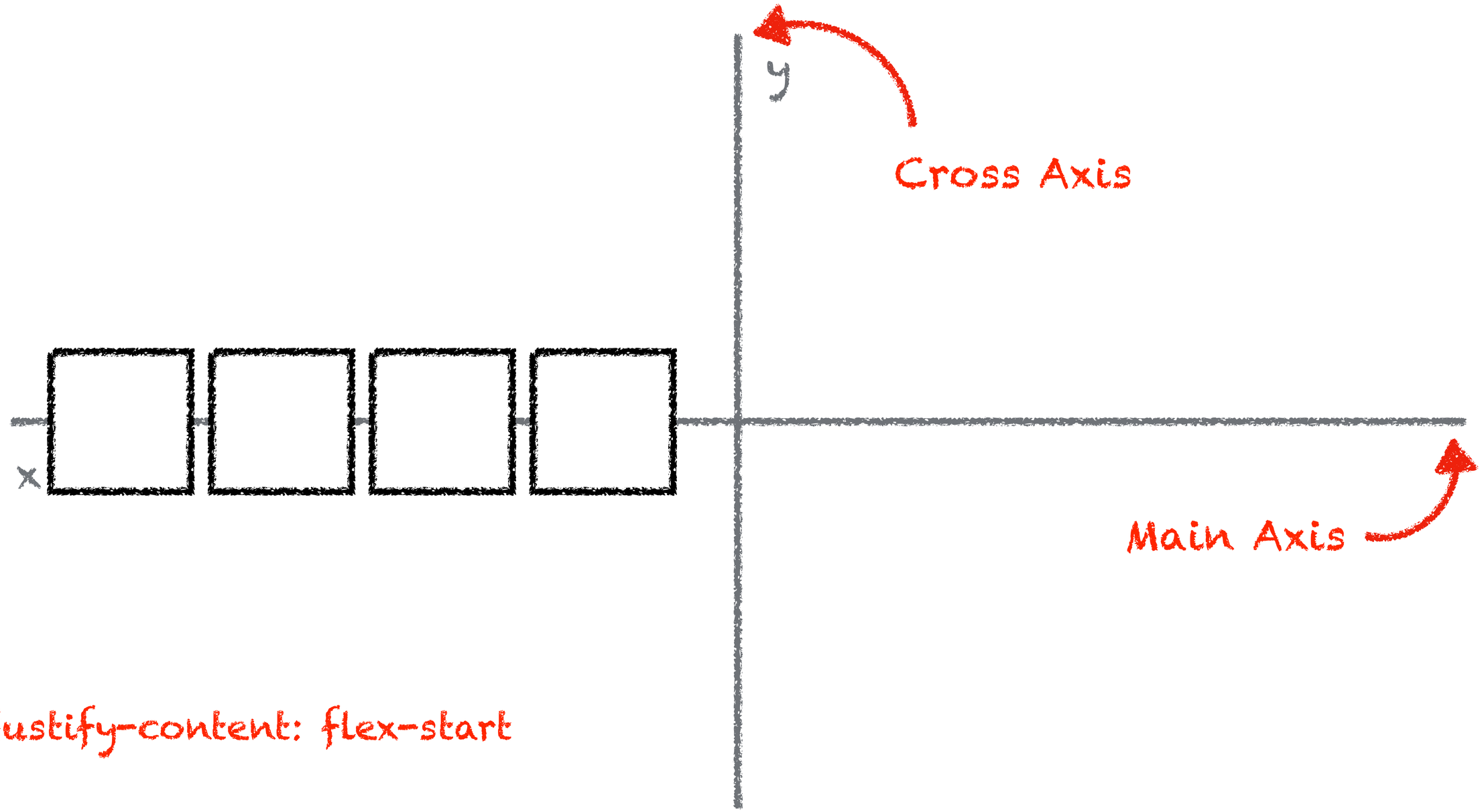
flex-direction: row
flex-direction: row-reverse


Main Axis
Cross Axis
flex-direction: column
flex-direction: column-reverse

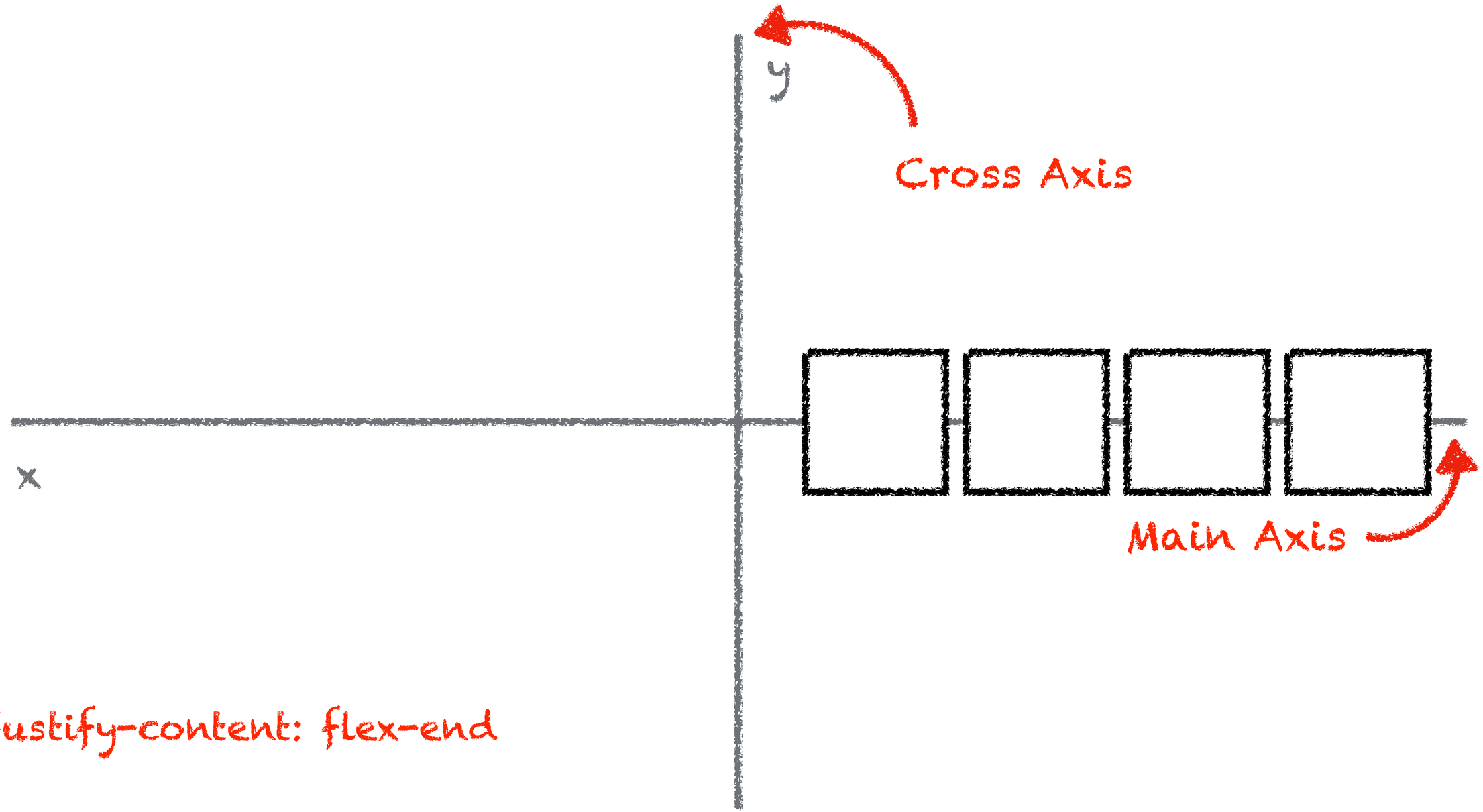
JUSTIFY-CONTENT



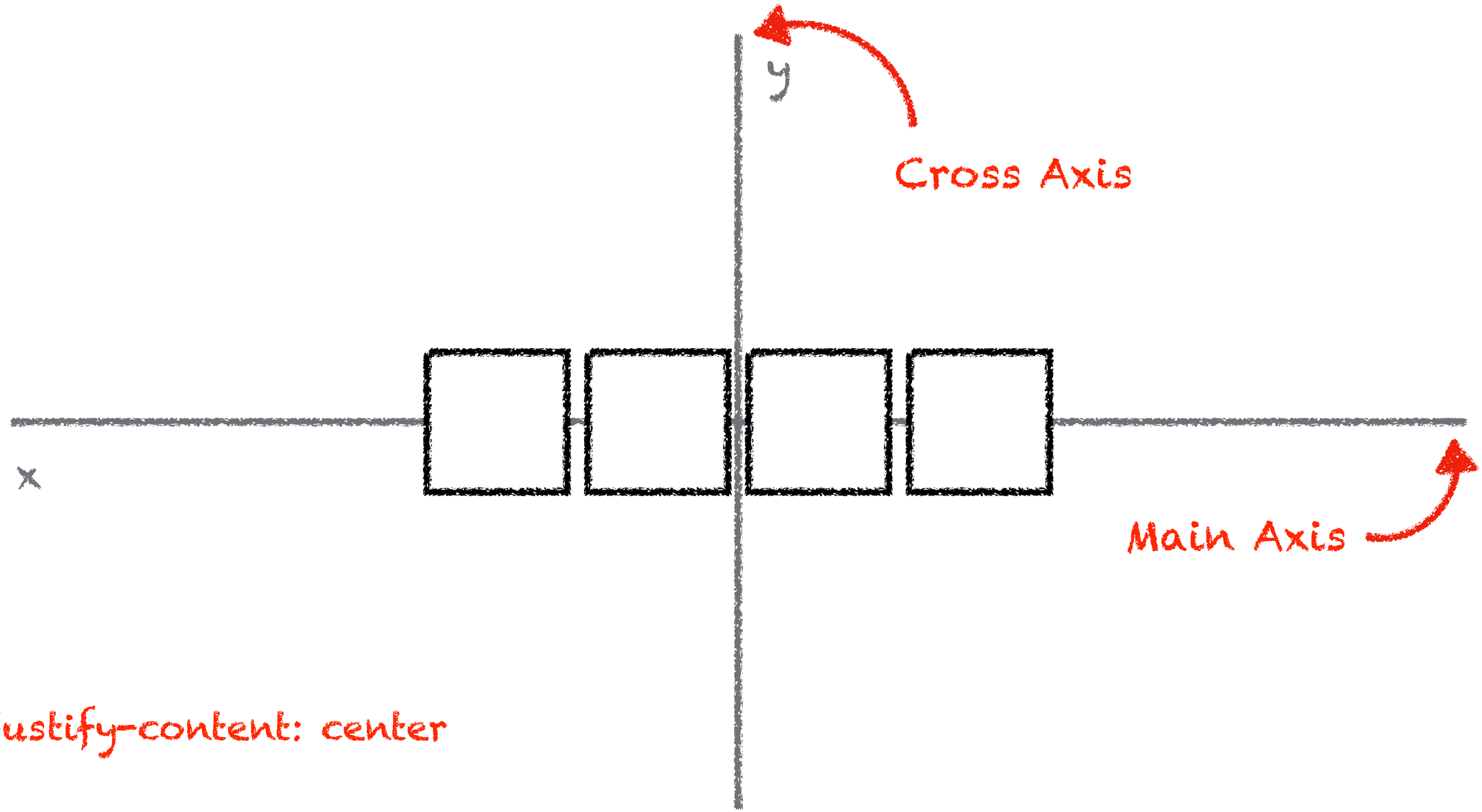
- Used to set how the children are distributed along the main axis
- The values for `justify-content` are:
  - `flex-start` (default)
  - `flex-end`
  - `center`
  - `space-between`
  - `space-around`
  - `space-evenly`



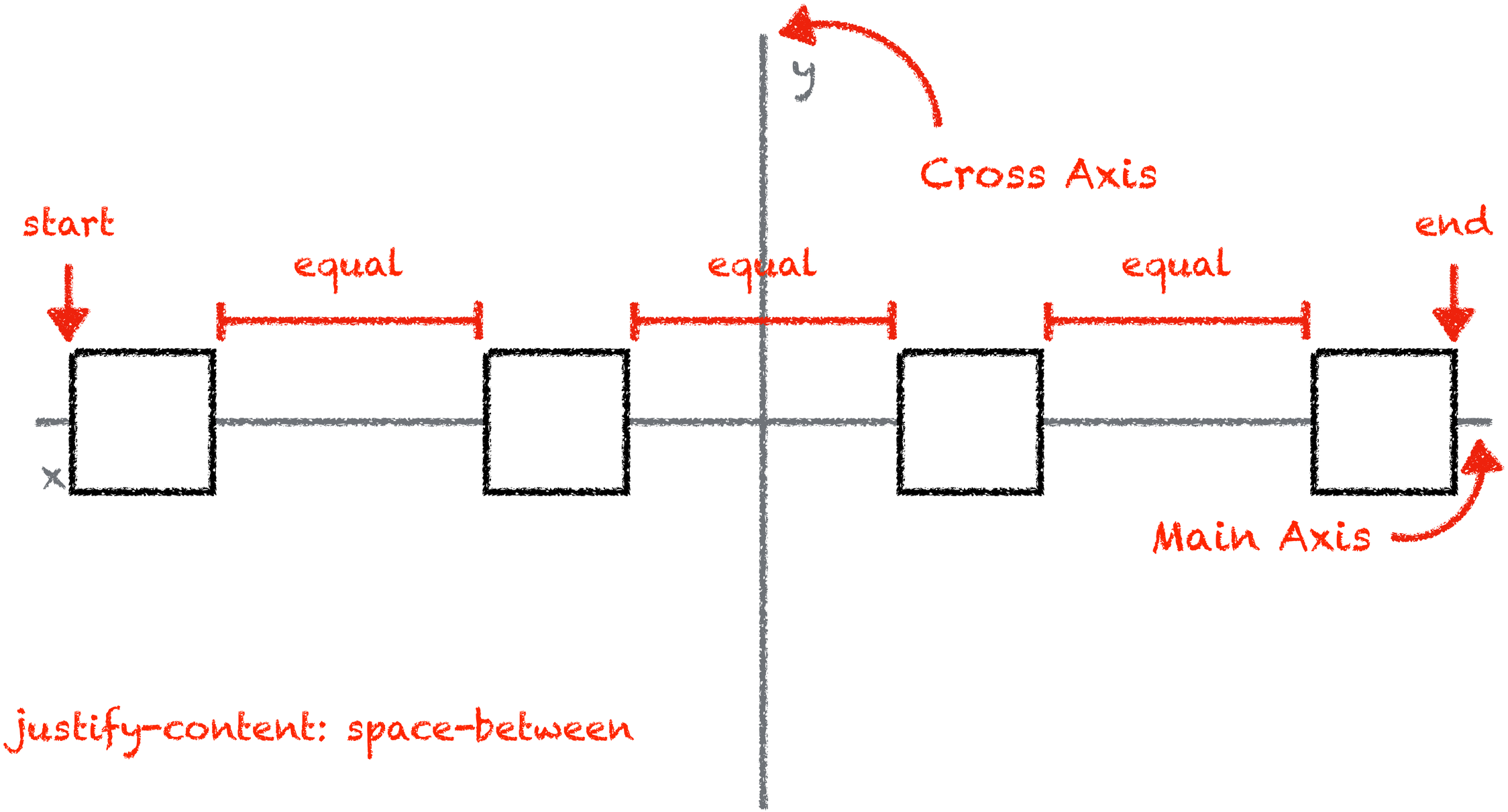
justify-content: flex-start



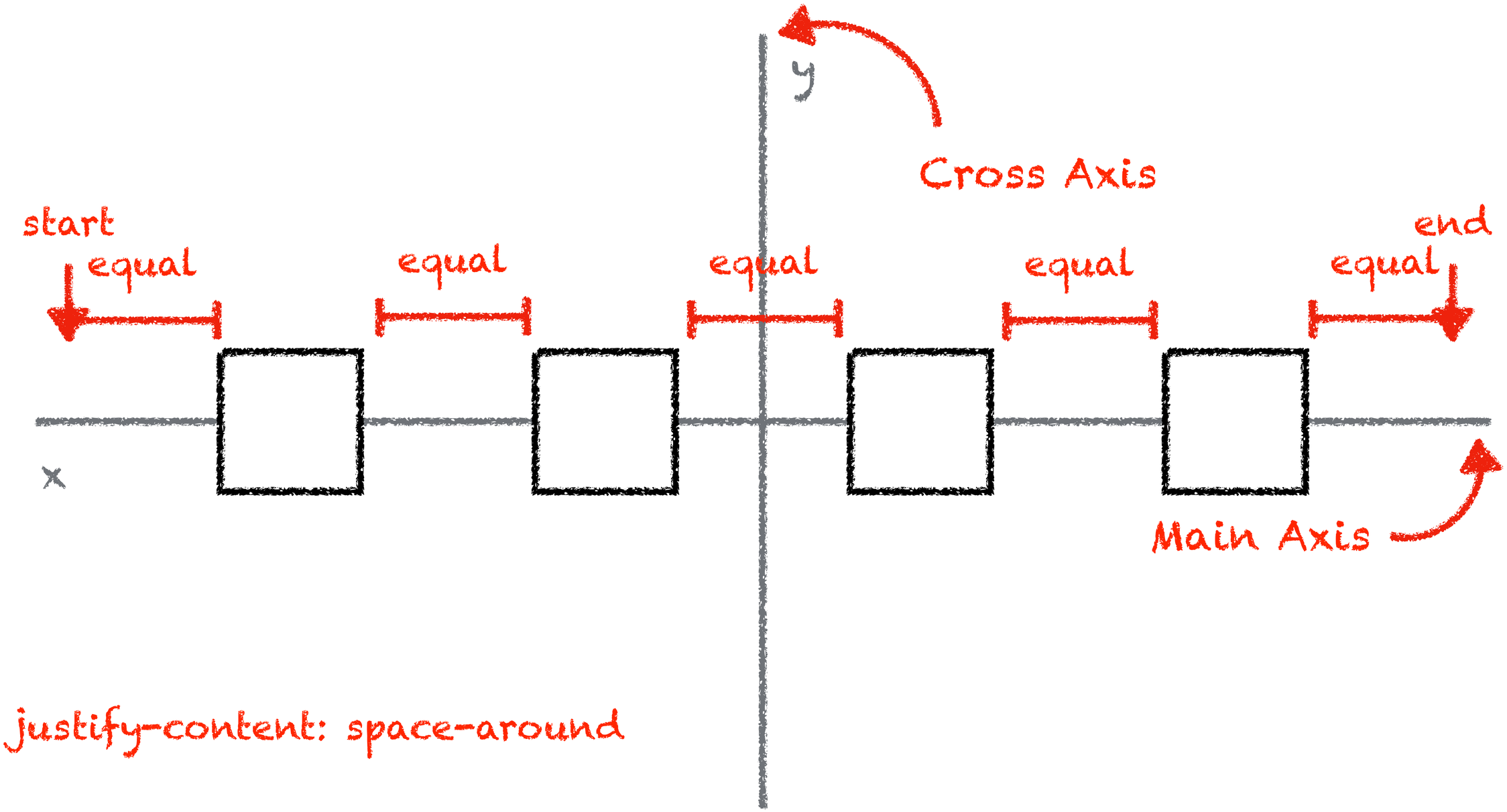
justify-content: flex-end



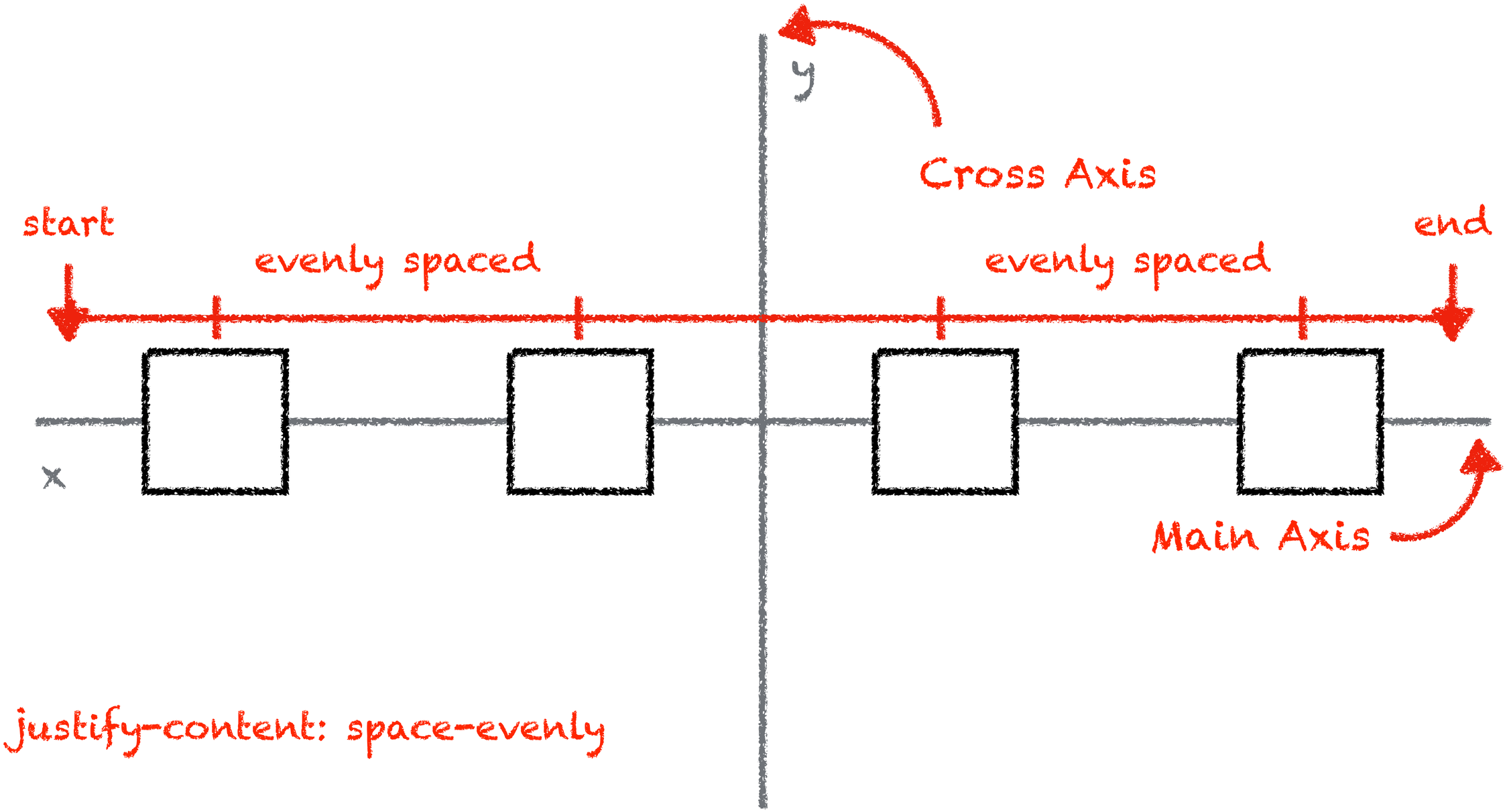
justify-content: center



justify-content: space-between

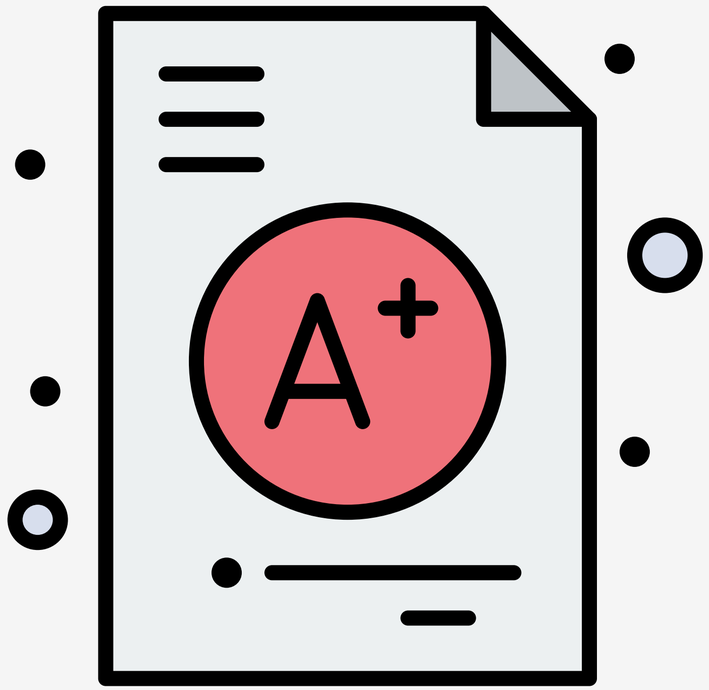


justify-content: space-around

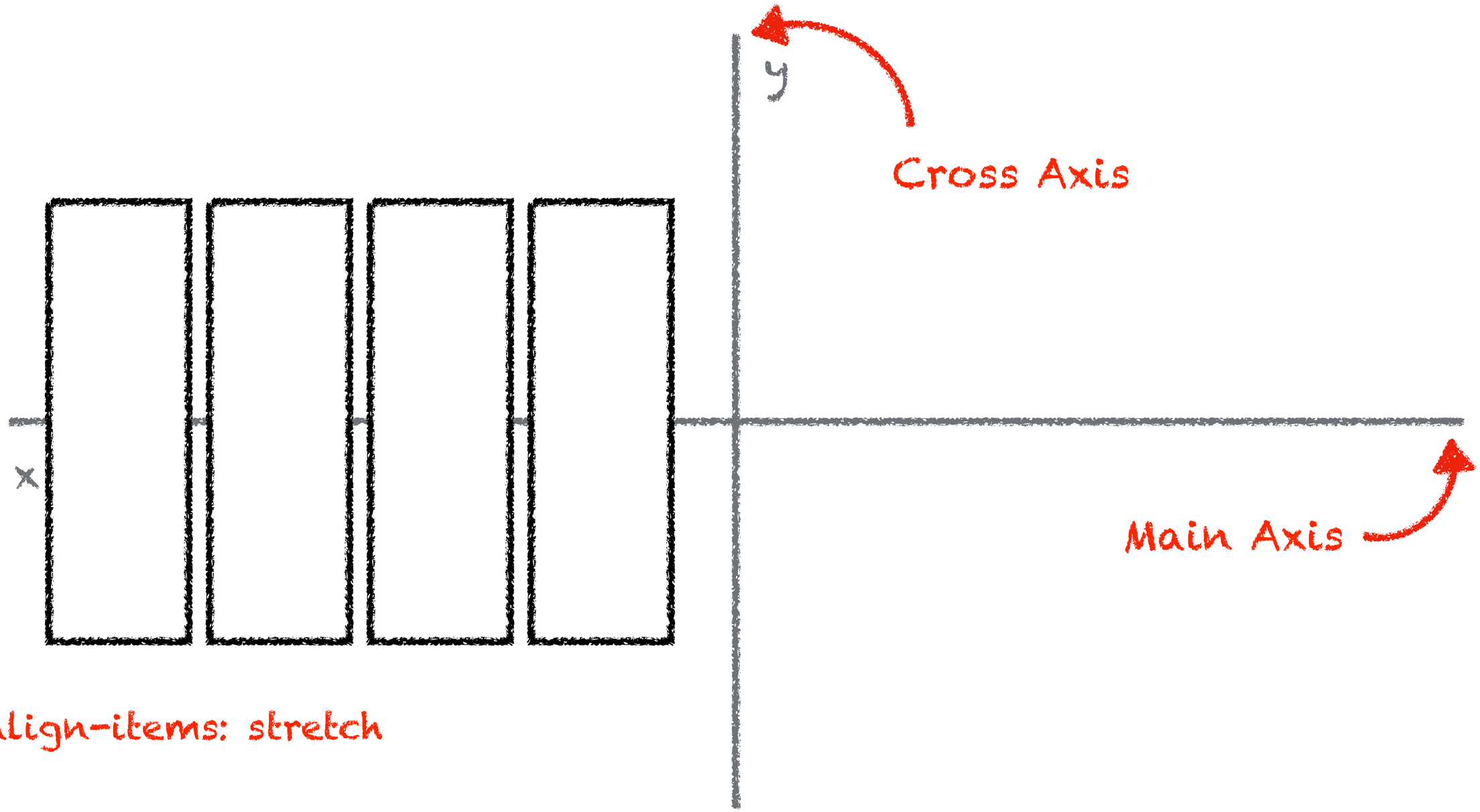


justify-content: space-evenly

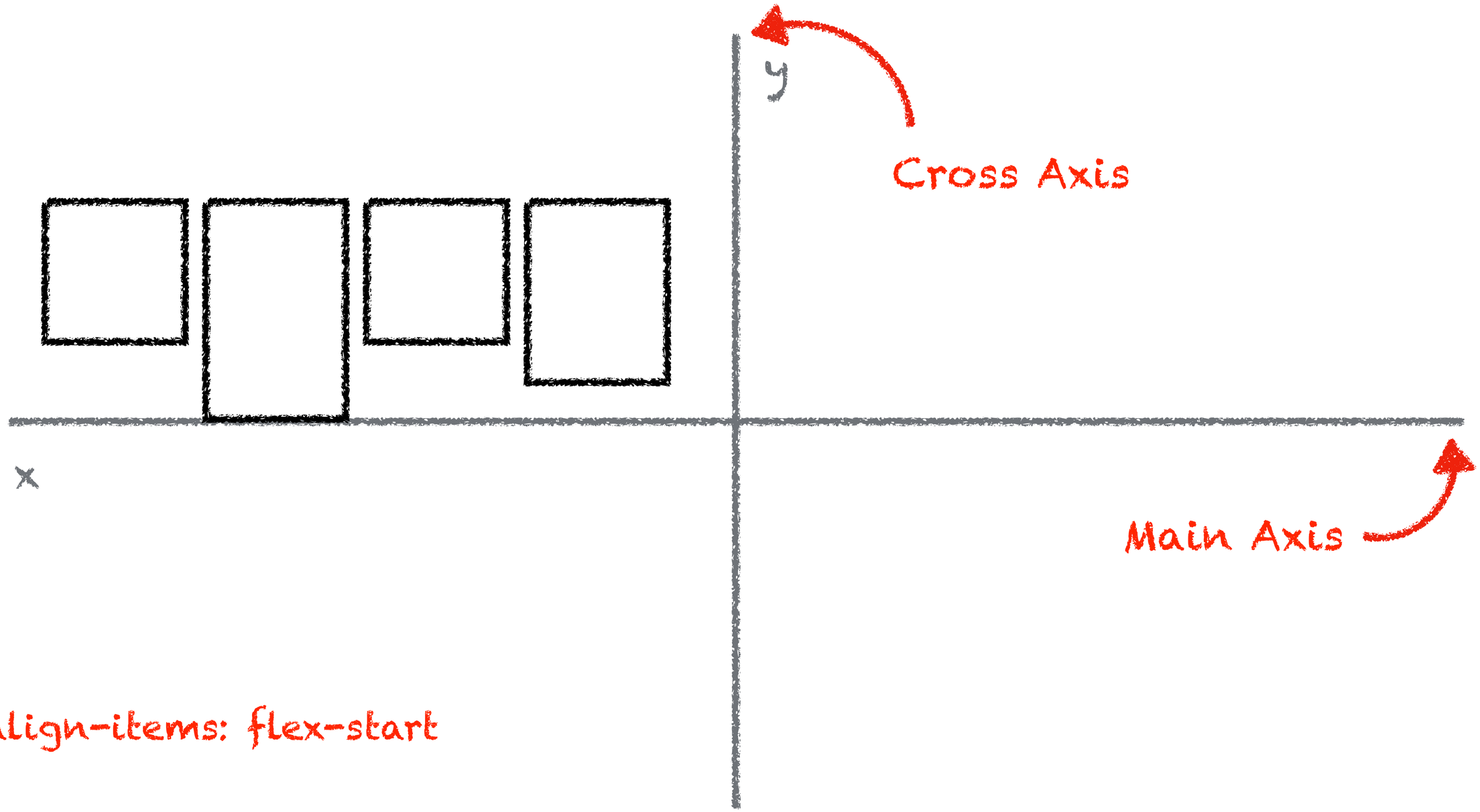
ALIGN-ITEMS



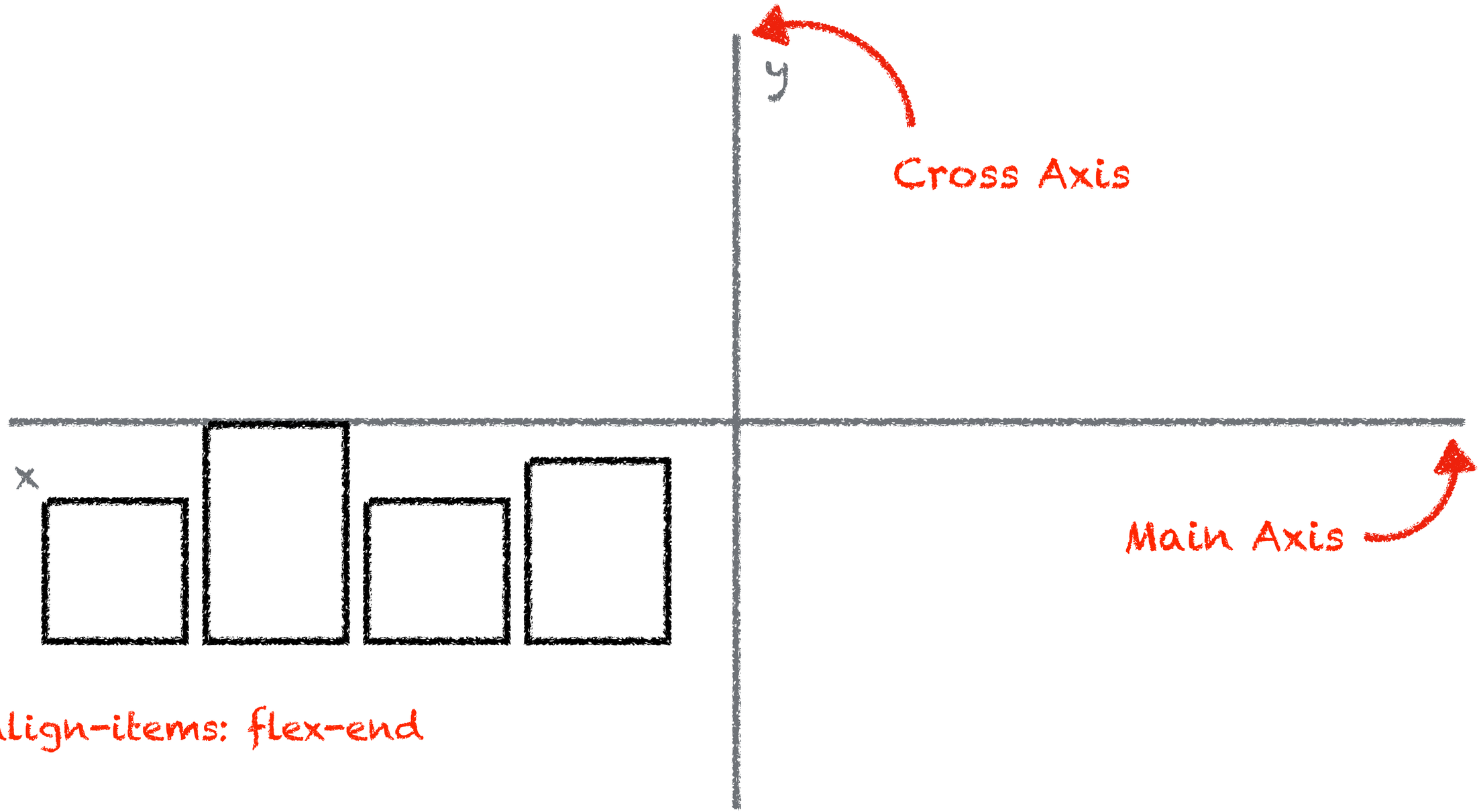
- Used to set how the children are laid along the cross axis
- The values for `align-items` are:
  - `stretch` (default)
  - `flex-start`
  - `flex-end`
  - `center`
  - `baseline`



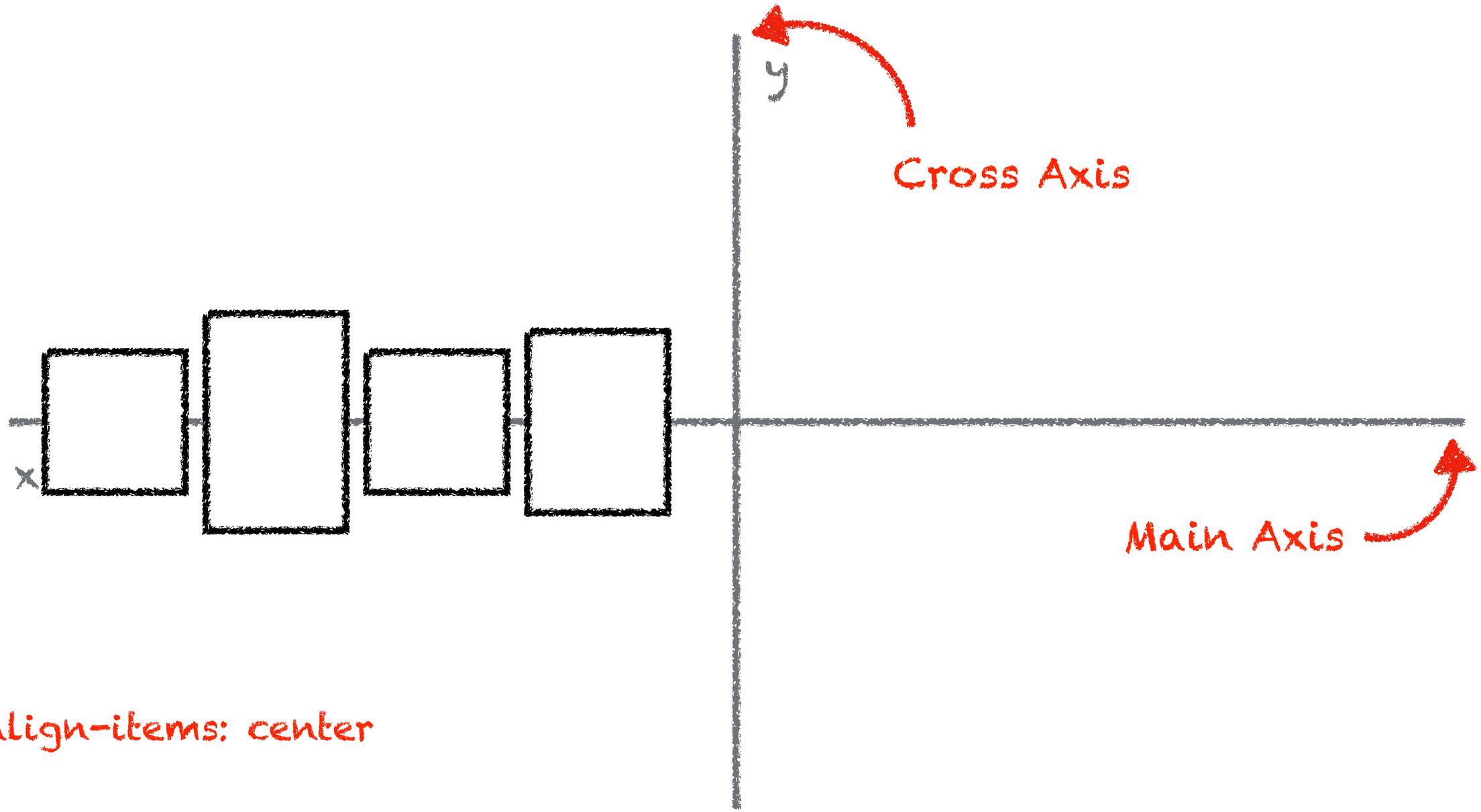
align-items: stretch



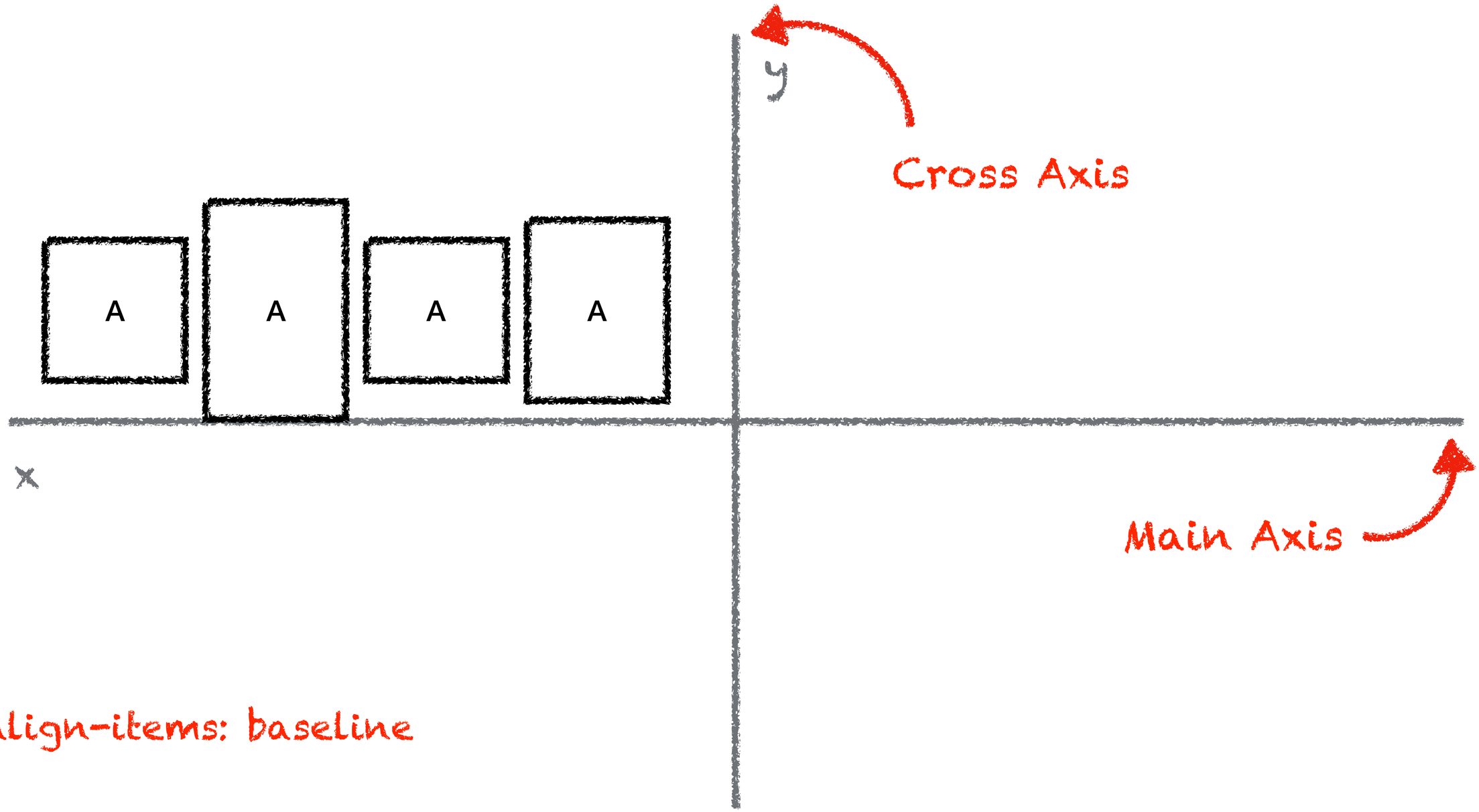
align-items: flex-start



align-items: flex-end

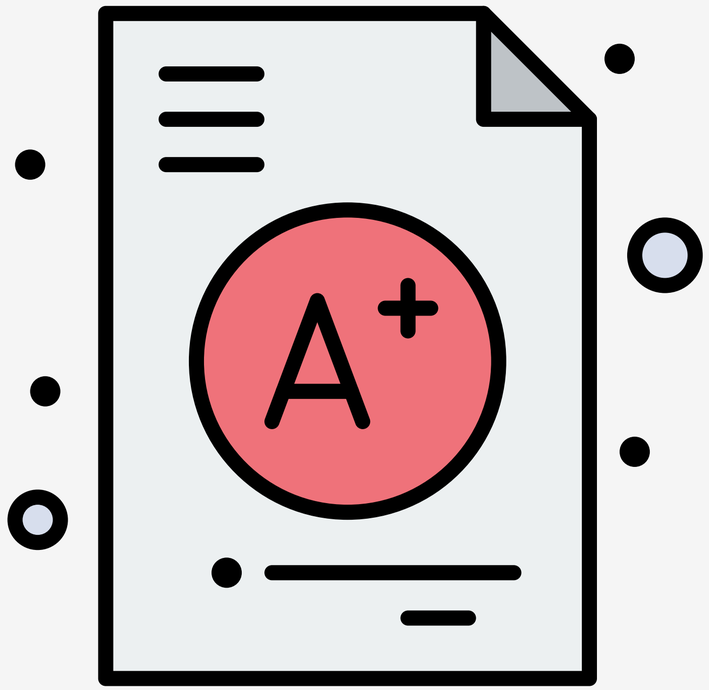


align-items: center

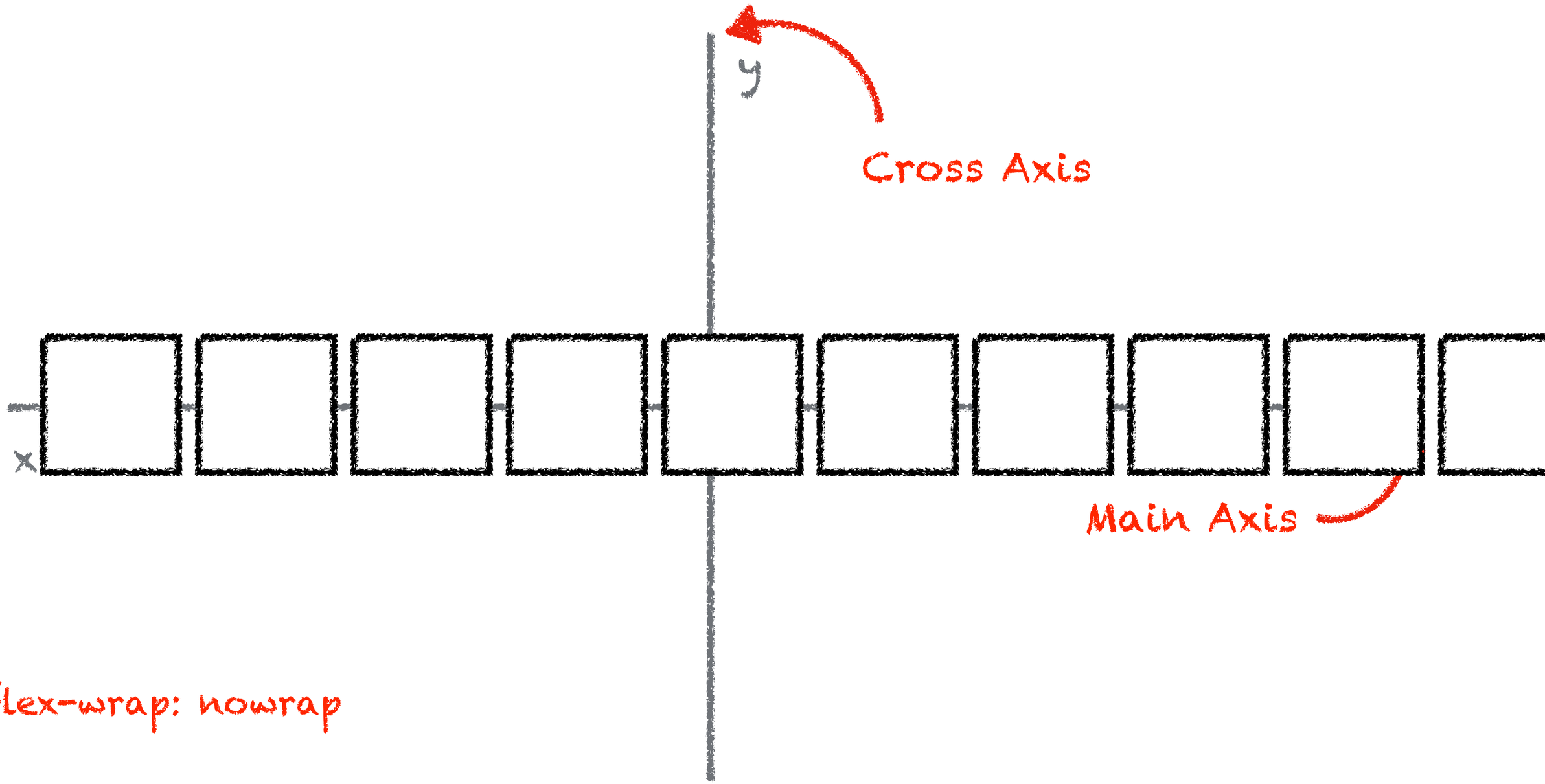


align-items: baseline

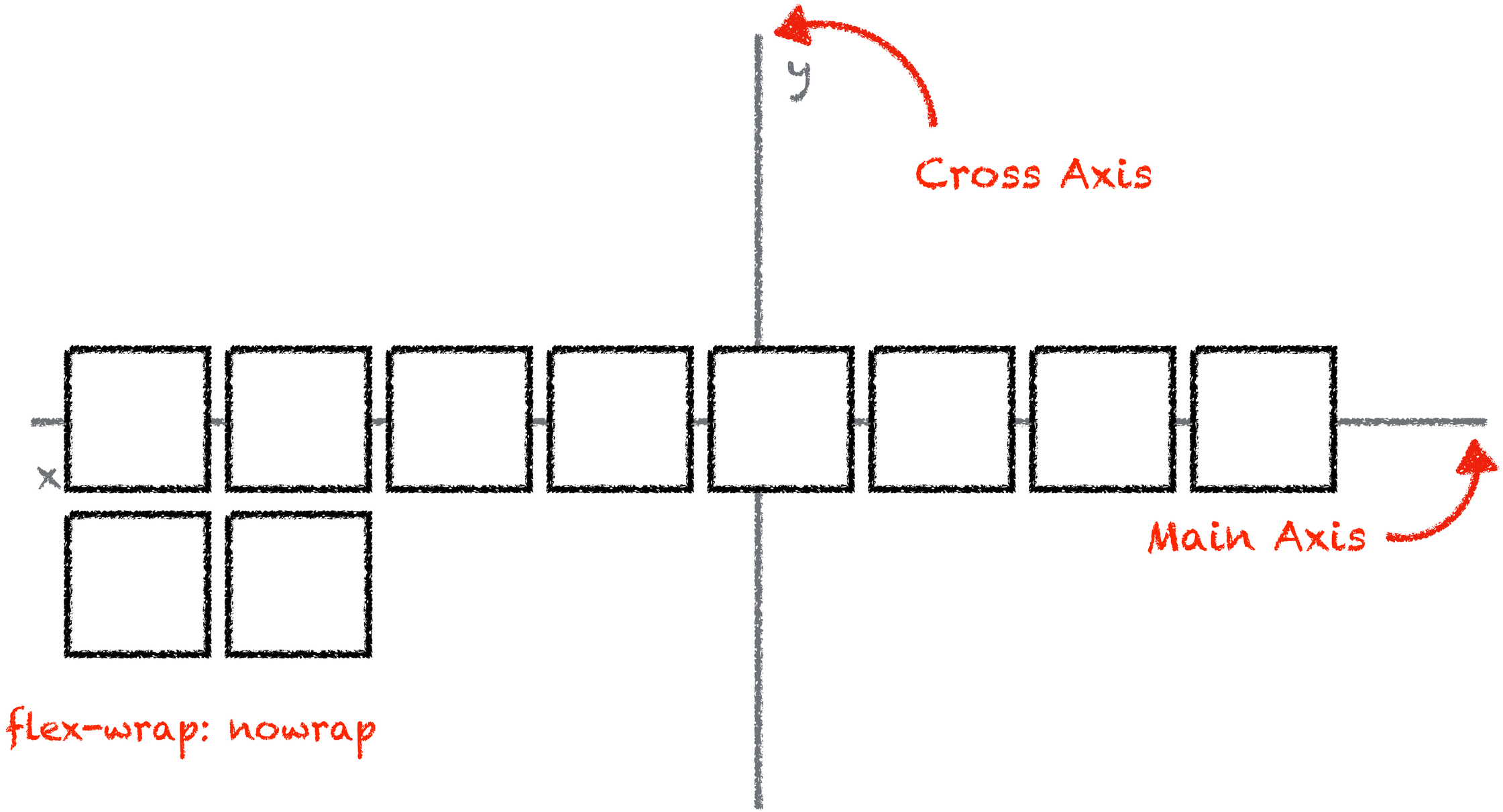
FLEX-WRAP



- By default, children of a flexbox container will remain on a single line
- The `flex-wrap` property controls how children will wrap
- The values for `flex-wrap` are:
  - nowrap (default)
  - wrap
  - wrap-reverse



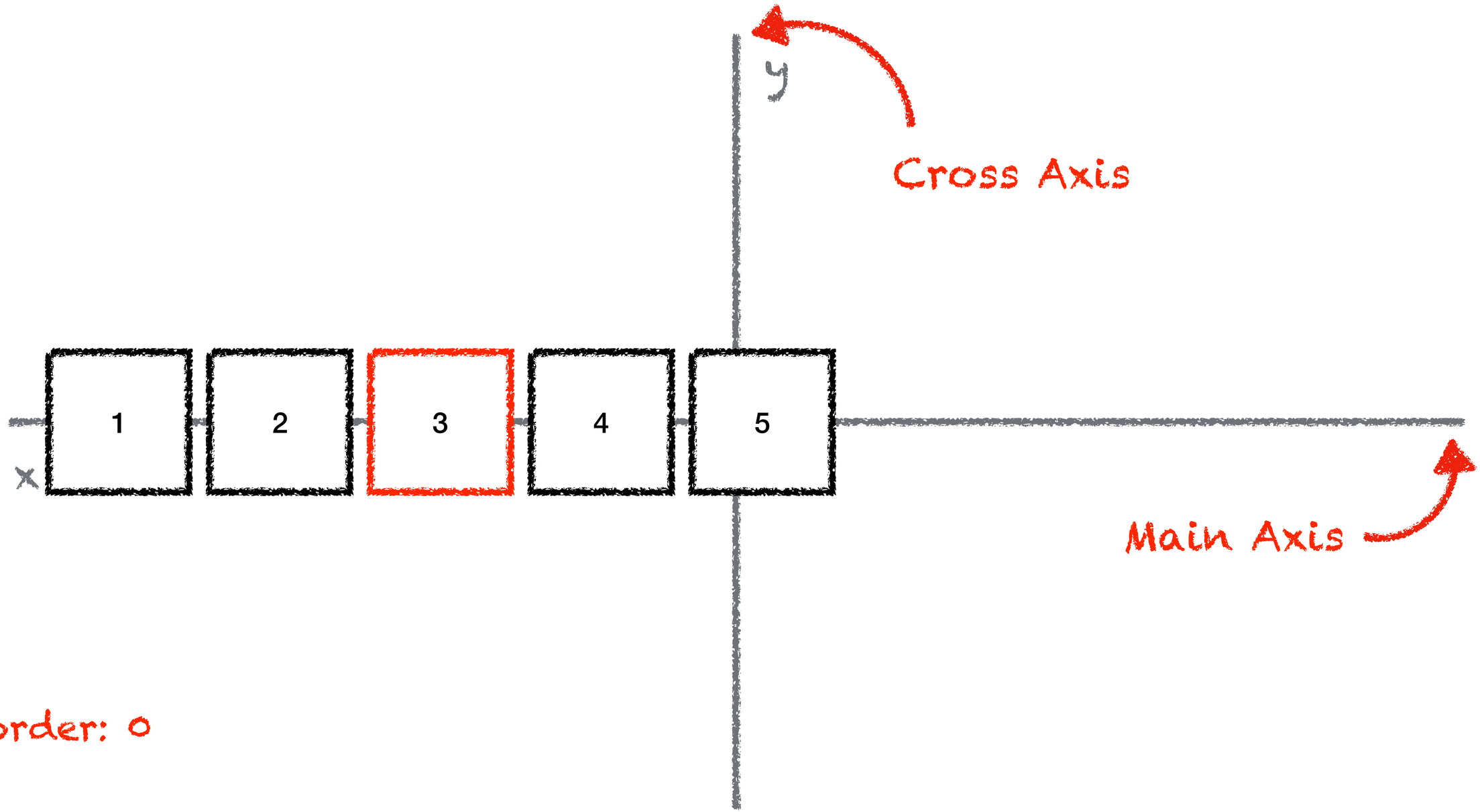
flex-wrap: nowrap

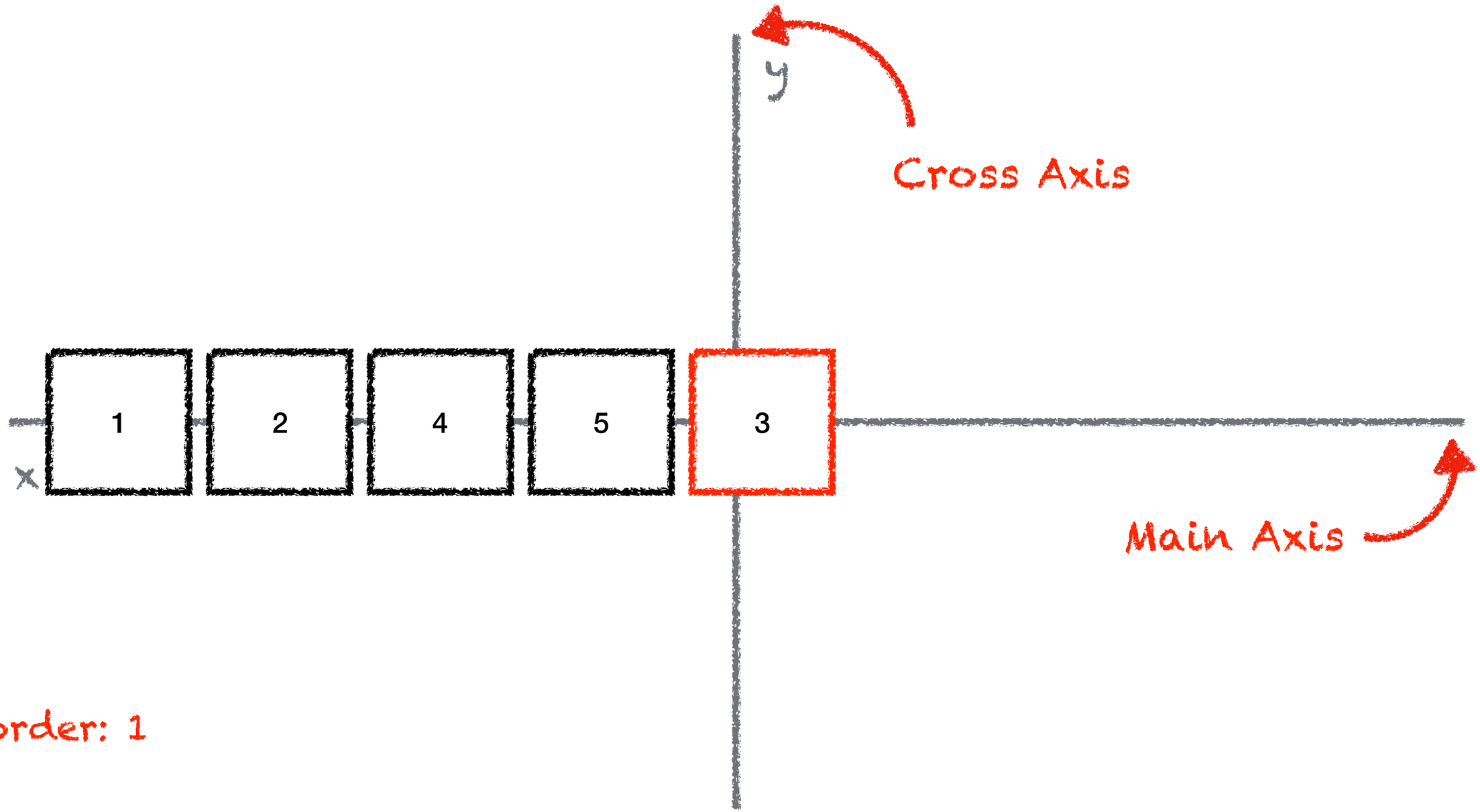

y
Cross Axis
x
Main Axis
flex-wrap: nowrap

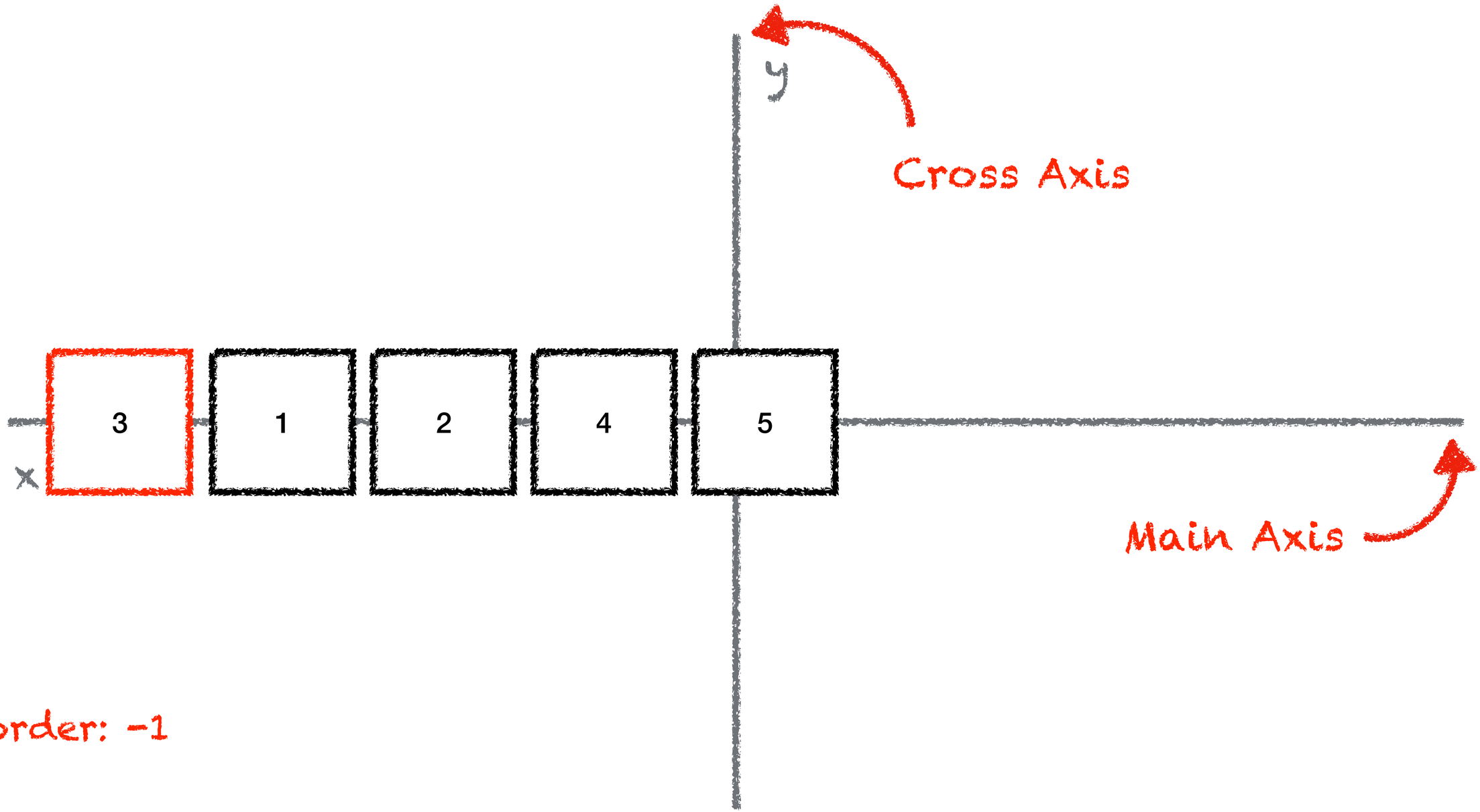
FLEXBOX ITEMS

ORDER

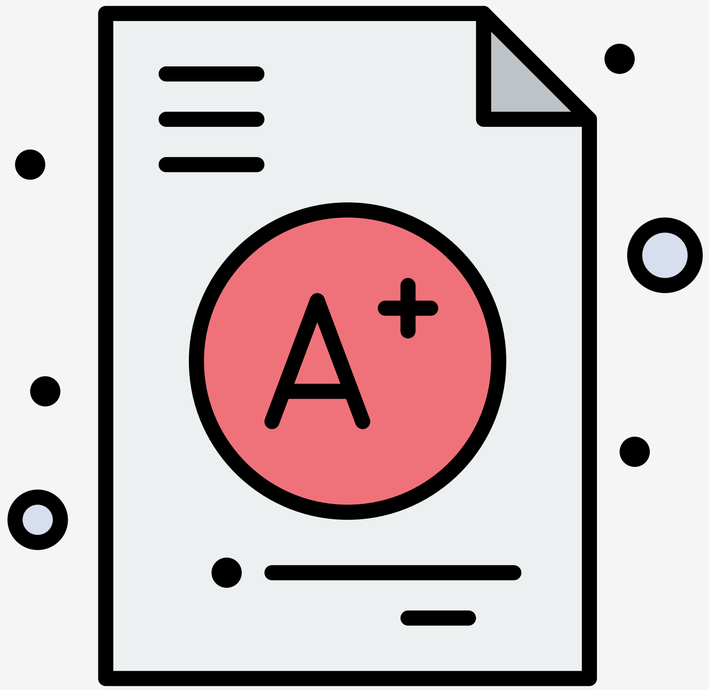
- By default, children of a flexbox container will be displayed in source order
- The `order` property alters that order
- The `order` property takes any positive or negative integer
- The default value is `0`
- Elements will be place in order by their order value


y
Cross Axis
1
2
3
4
5
x
Main Axis
order: 0

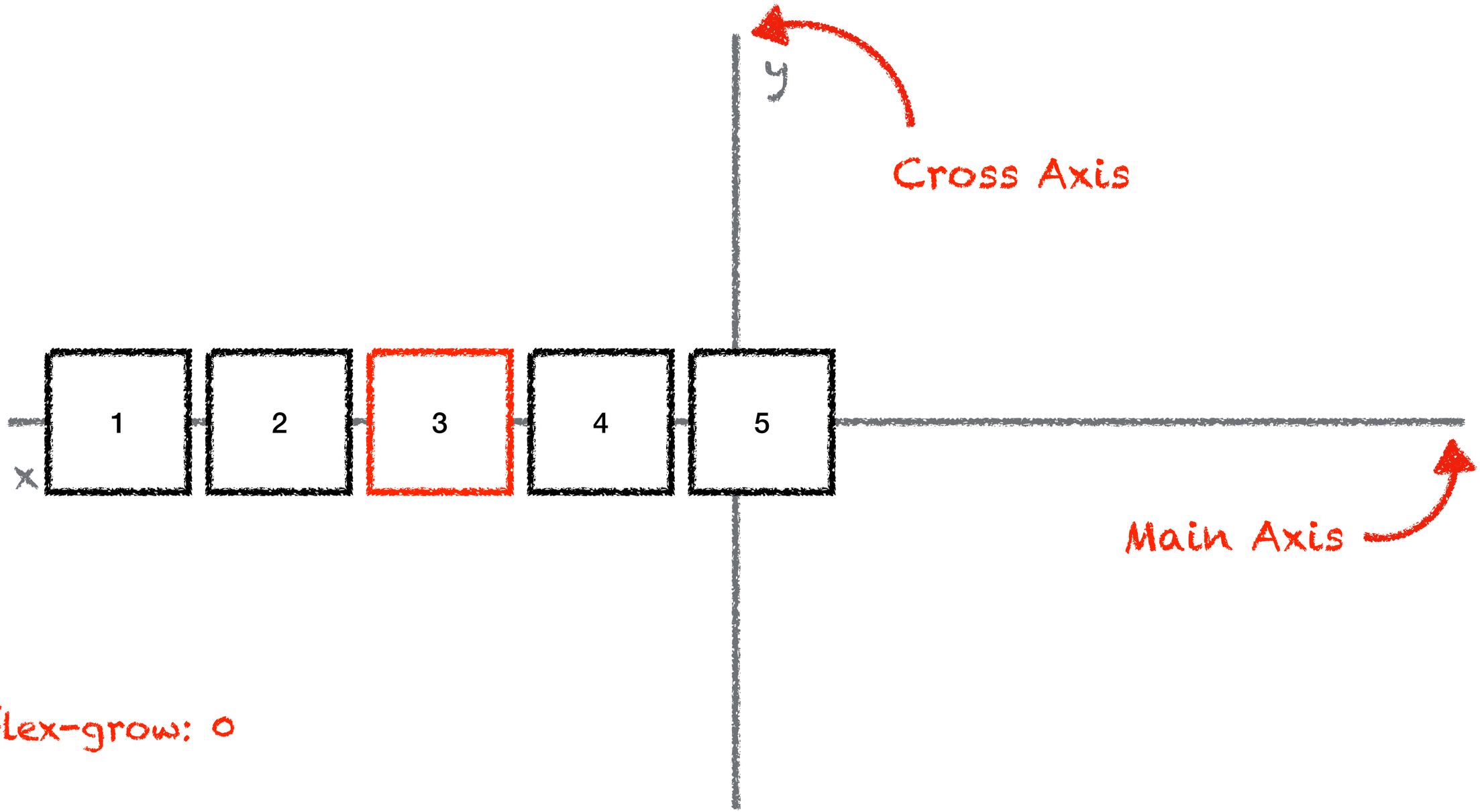

y
Cross Axis
1
2
4
5
3
x
Main Axis
order: 1


y
Cross Axis
3
1
2
4
5
x
Main Axis
order: -1

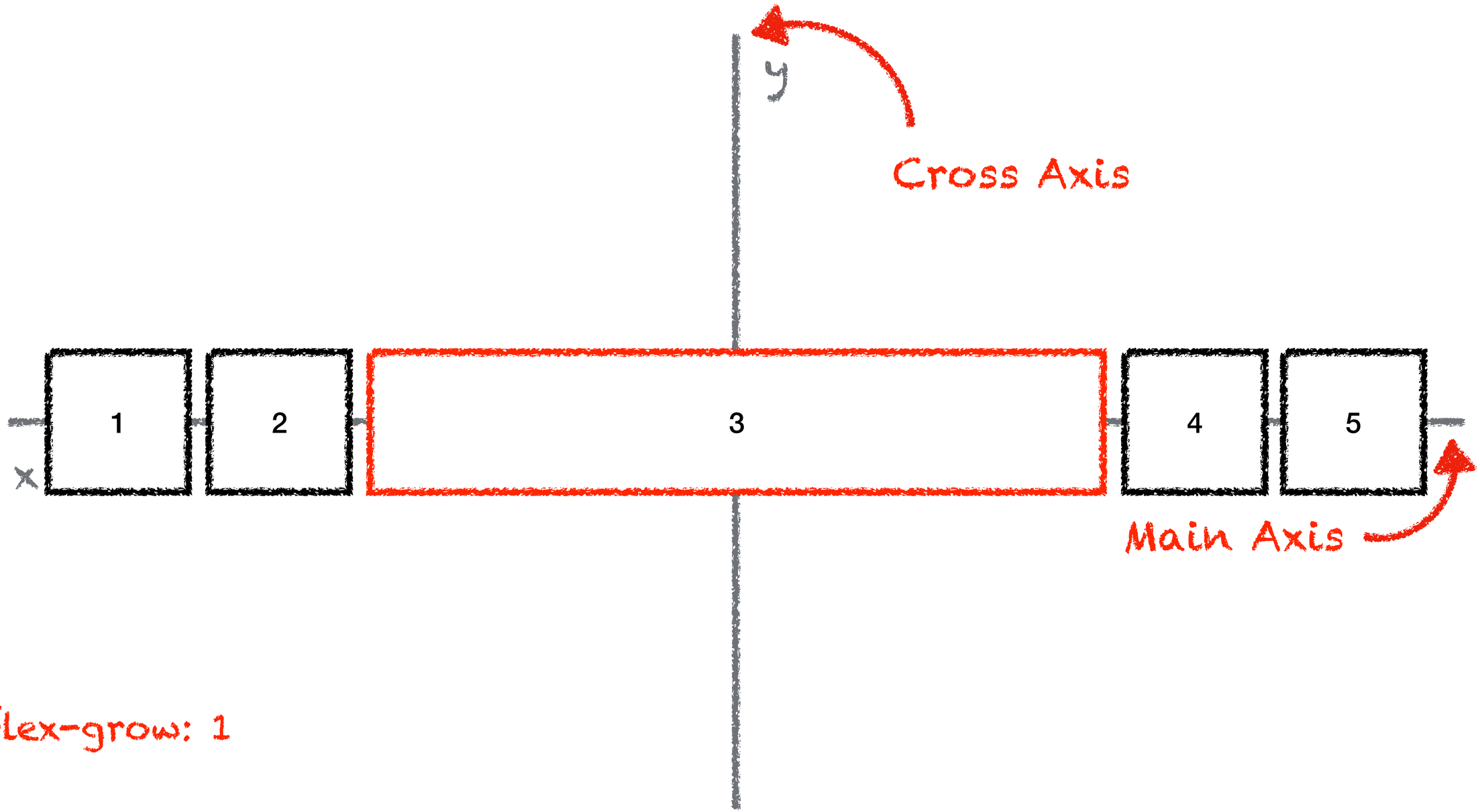
FLEX-GROW



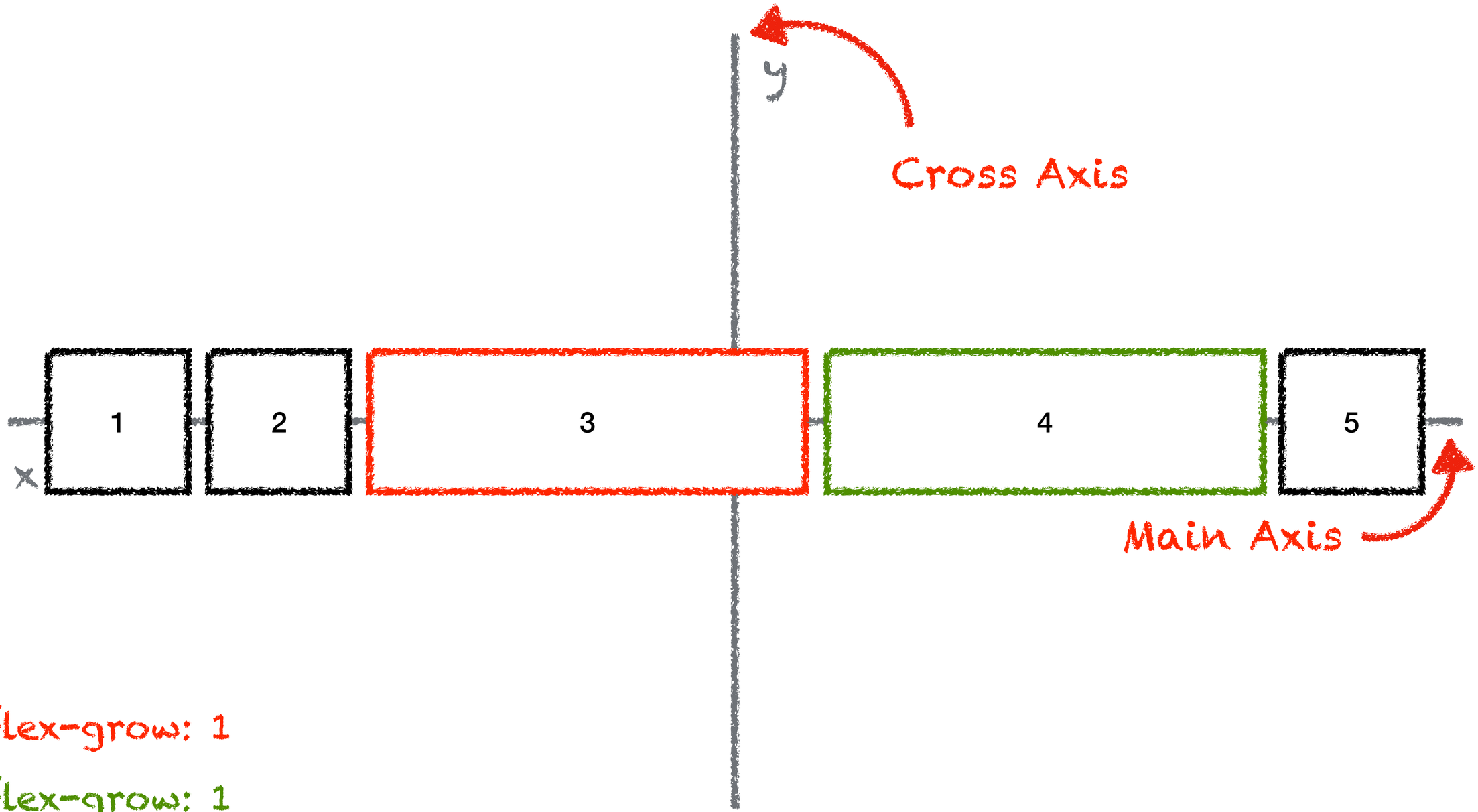
- By default, children will only as much space as necessary on the main axis
- The `flex-grow` property make the child "grow" to fill any available space
- The `flex-grow` property will take any positive integer, with `0` being the default.
- Elements with a higher `flex-grow` value will take more of the available space

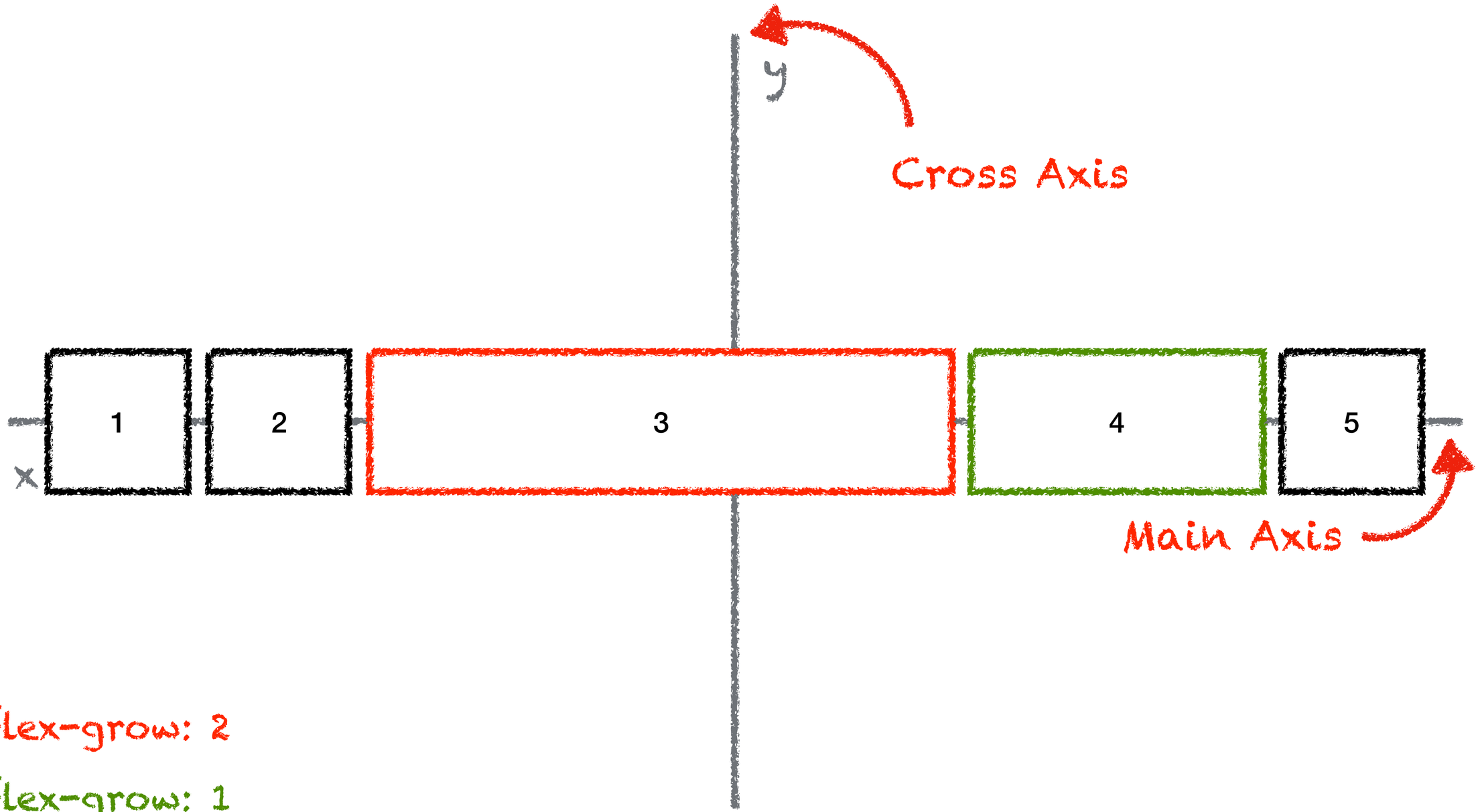


flex-grow: 0

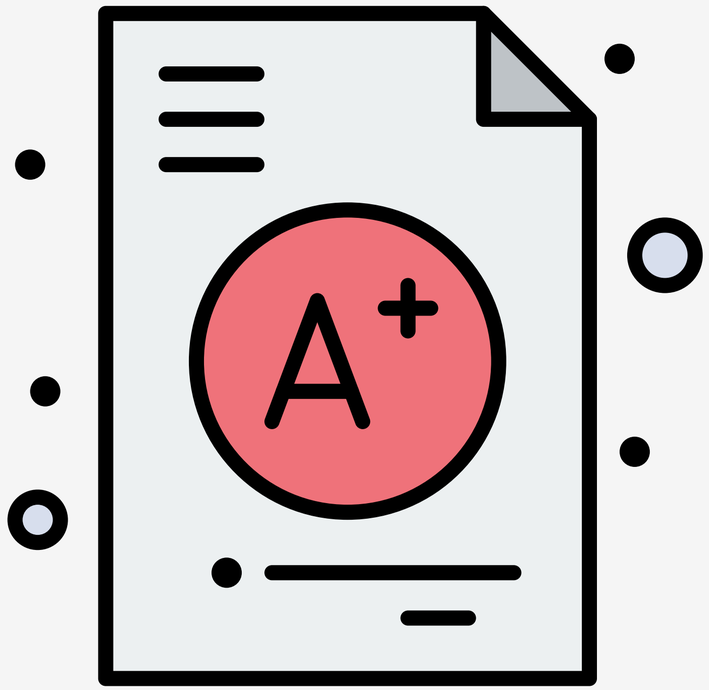


flex-grow: 1

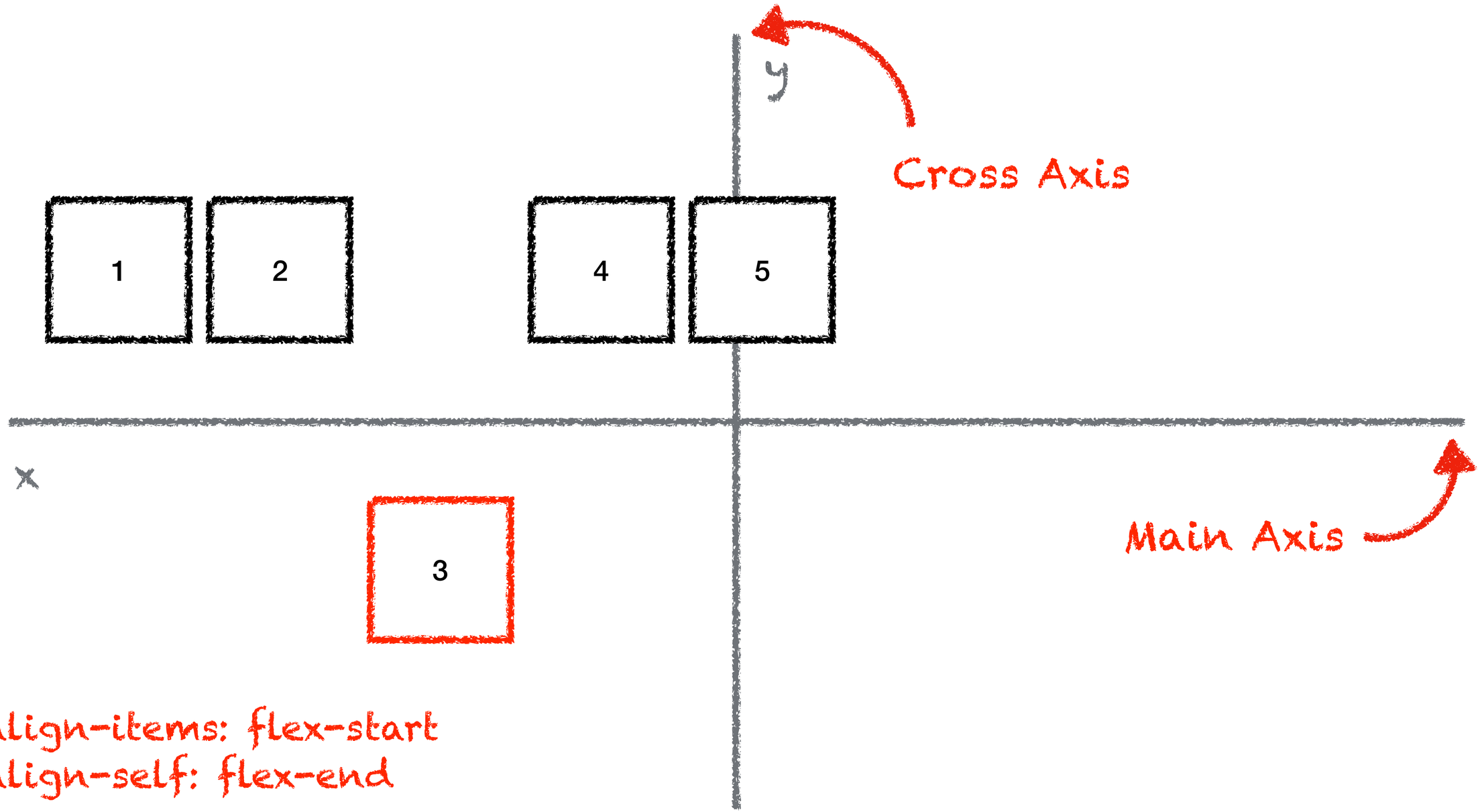

y
Cross Axis
1
2
3
4
5
x
Main Axis
flex-grow: 1
flex-grow: 1


y
Cross Axis
1
2
3
4
5
x
Main Axis
flex-grow: 2
flex-grow: 1

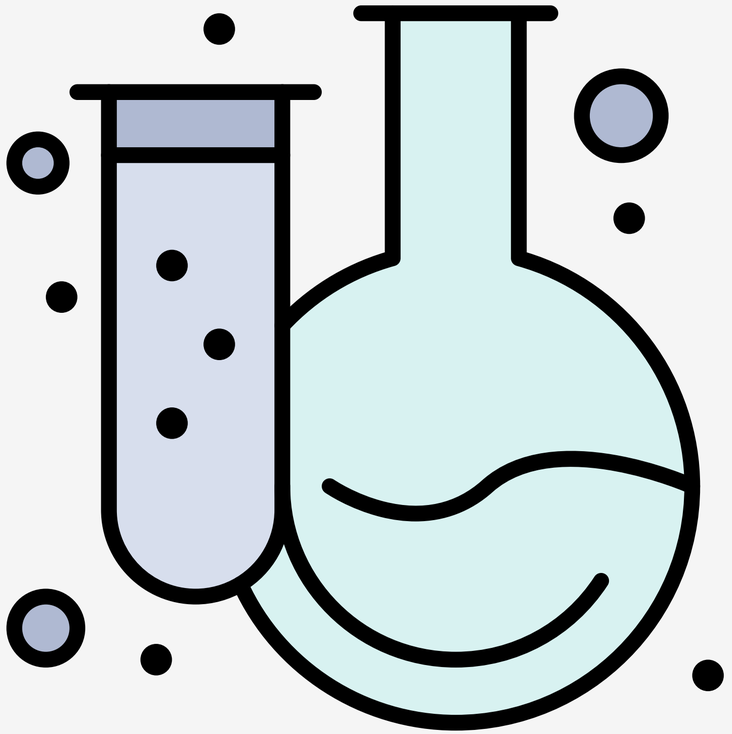
ALIGN-SELF



- The `align-self` property overrides the `align-items` property of the flex container
- The align-self property has all the same values as align-items

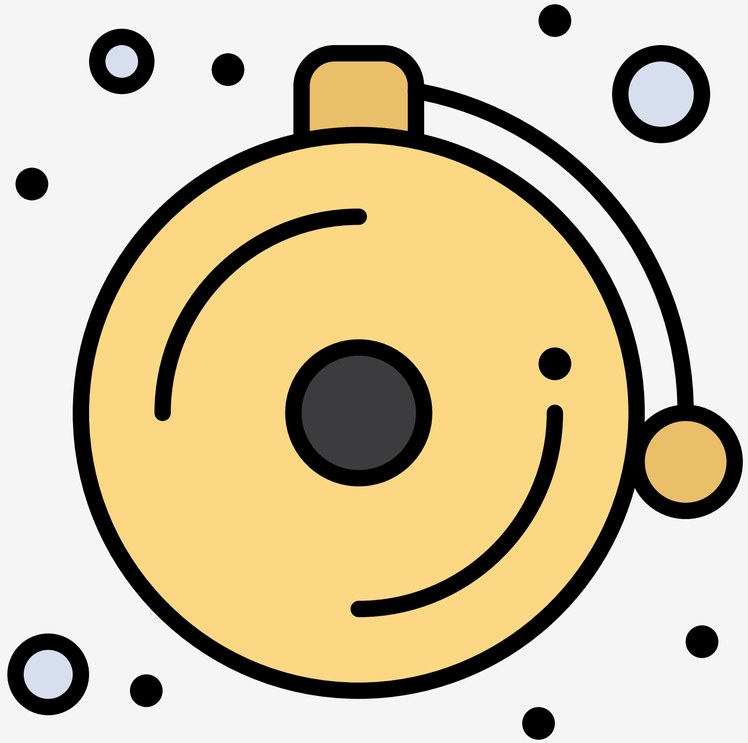

y
Cross Axis
1
2
4
5
x
3
Main Axis
align-items: flex-start
align-self: flex-end

FLEXBLOCKS



- *FORK THE PEN!*
- Use flexbox to stack the blocks as shown
- Use the properties suggested for each stack
- Do *NOT* change the HTML
- Submit the URL to your pen
- *DUE:* **Tue. Jan 21 @ 11:59 PM**

NEXT TIME...



- Flexbox Demonstration
- **Exercise:** Landing Page