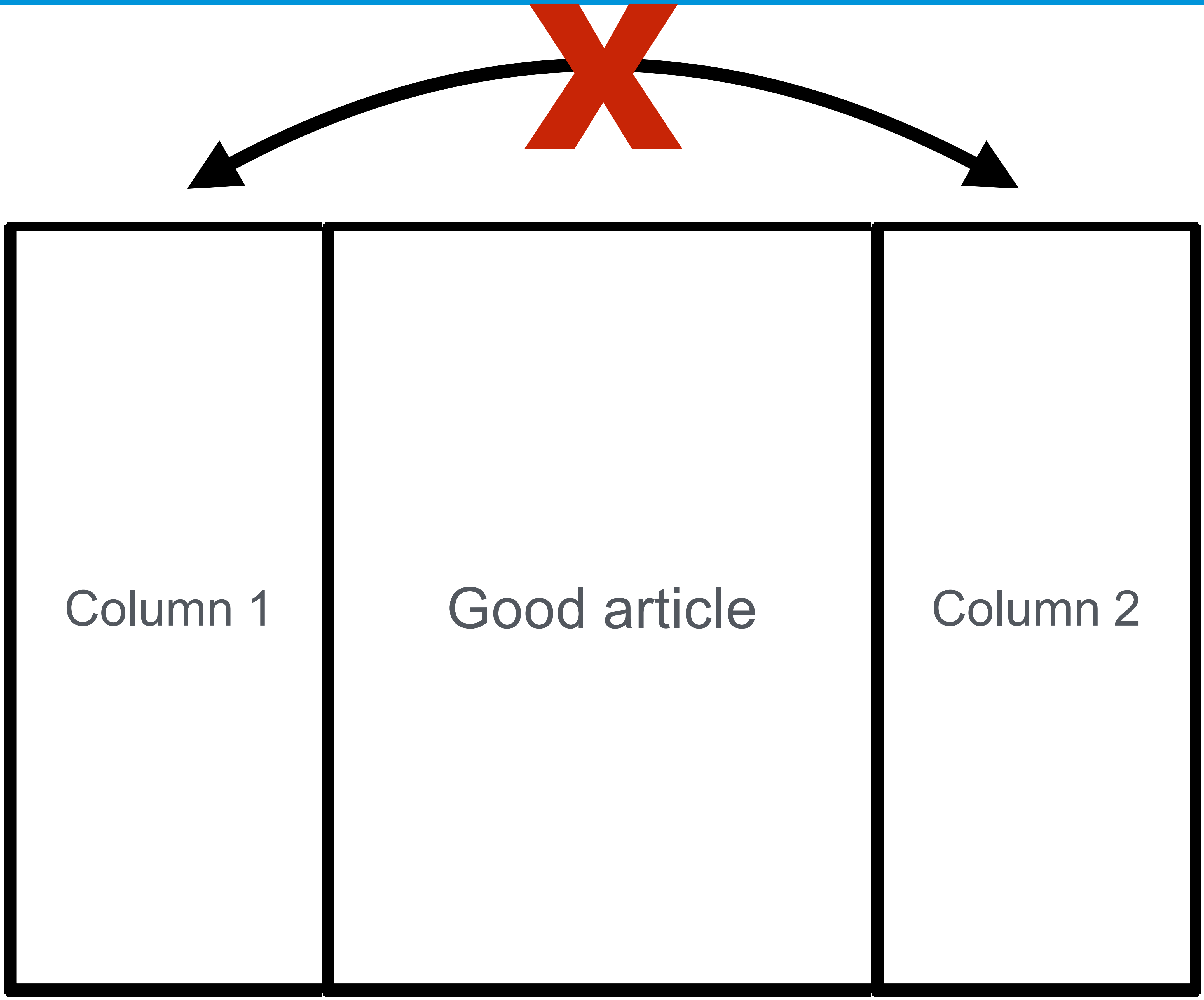
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Layout techniques

Flexbox and Grid

Table

Tables made history. They changed the way how we show and structure content in the websites, but…

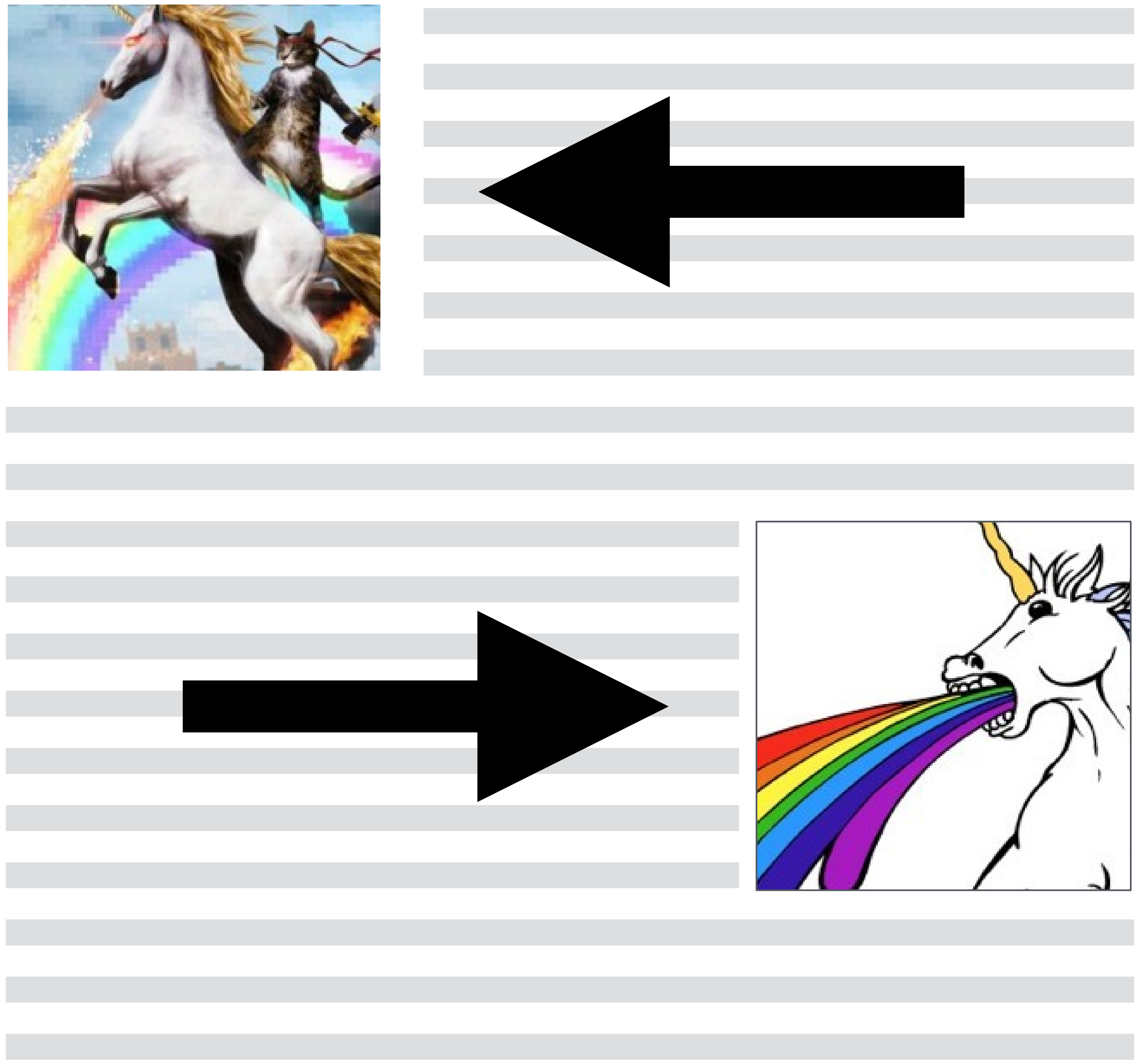


No sematic…

Float

Float give us some flexibility

but…



Floats affect other elements

Forcing you to use other properties and techniques to solve some problems:

clearfix, overflow, faux columns, double margins etc…

Float depends on the structure

You need to put the HTML elements in right place and order to make this right

How to solve the problem of structuring layouts?

Solutions to different problems

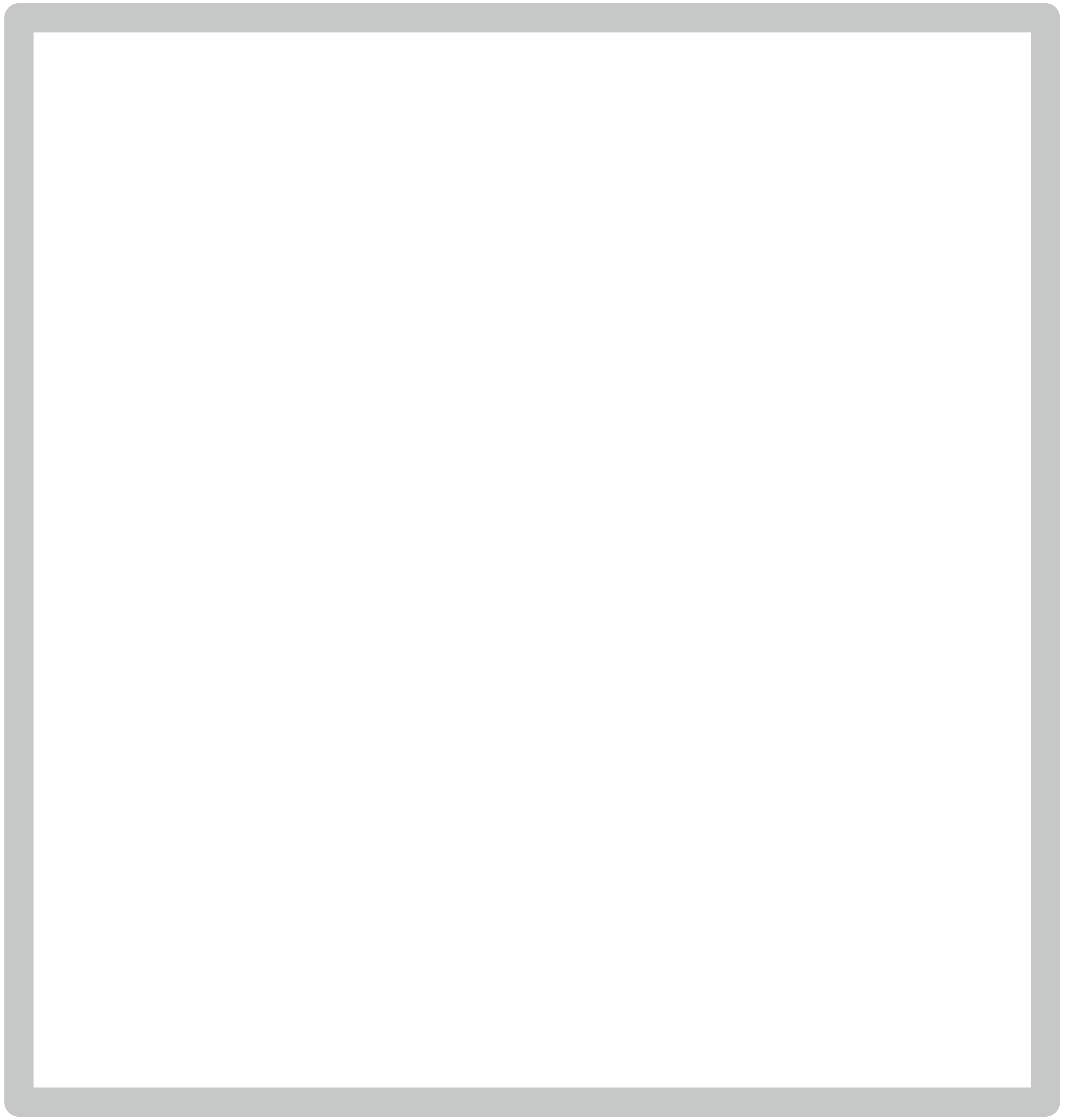
- Grid Layout
  - to structure parent elements
- Flexbox
  - to control the structure of child elements

Grid Layout

- This CSS module defines a two-dimensional grid-based layout system, optimized for user interface design
- In the grid layout model, the children of a grid container can be positioned into arbitrary slots in a flexible or fixed predefined layout grid

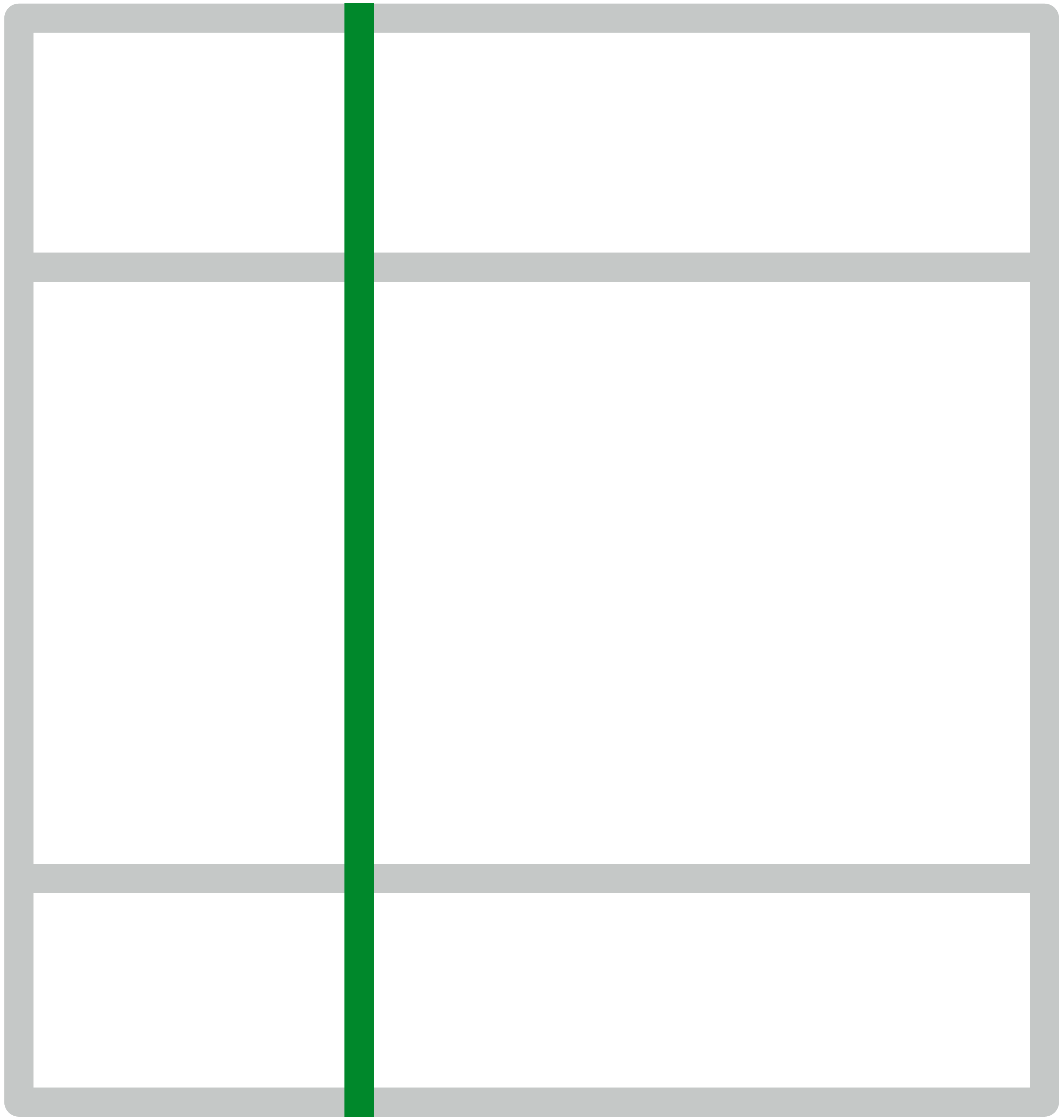
GRID TERMINNOLOGY

Grid container



A grid container establishes a new grid formatting context for its contents.

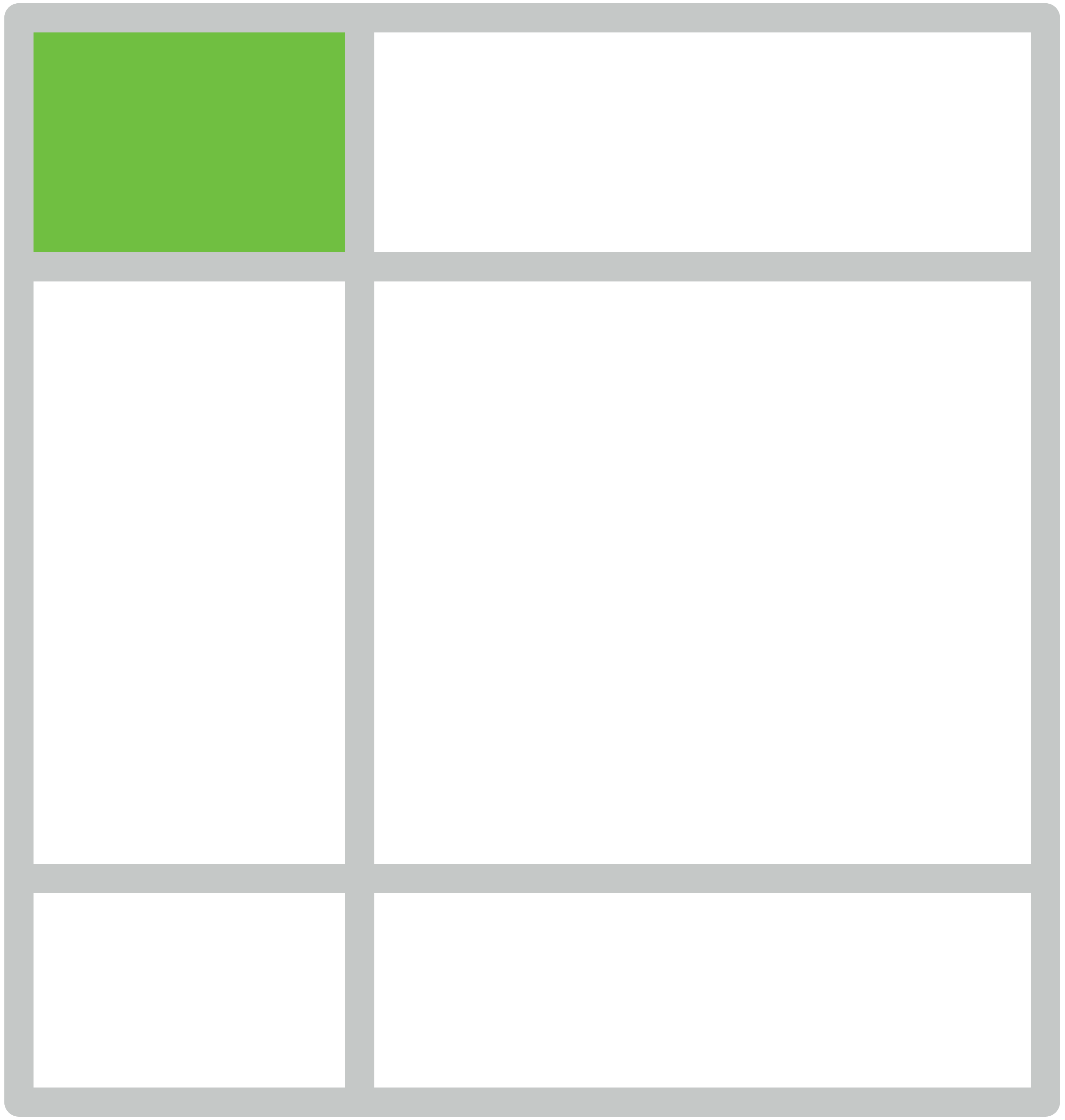
Grid lines



Grid lines are horizontal or vertical lines between grid cells. They can be named or referred by numbers.

The highlighted line in the image is the column line 2.

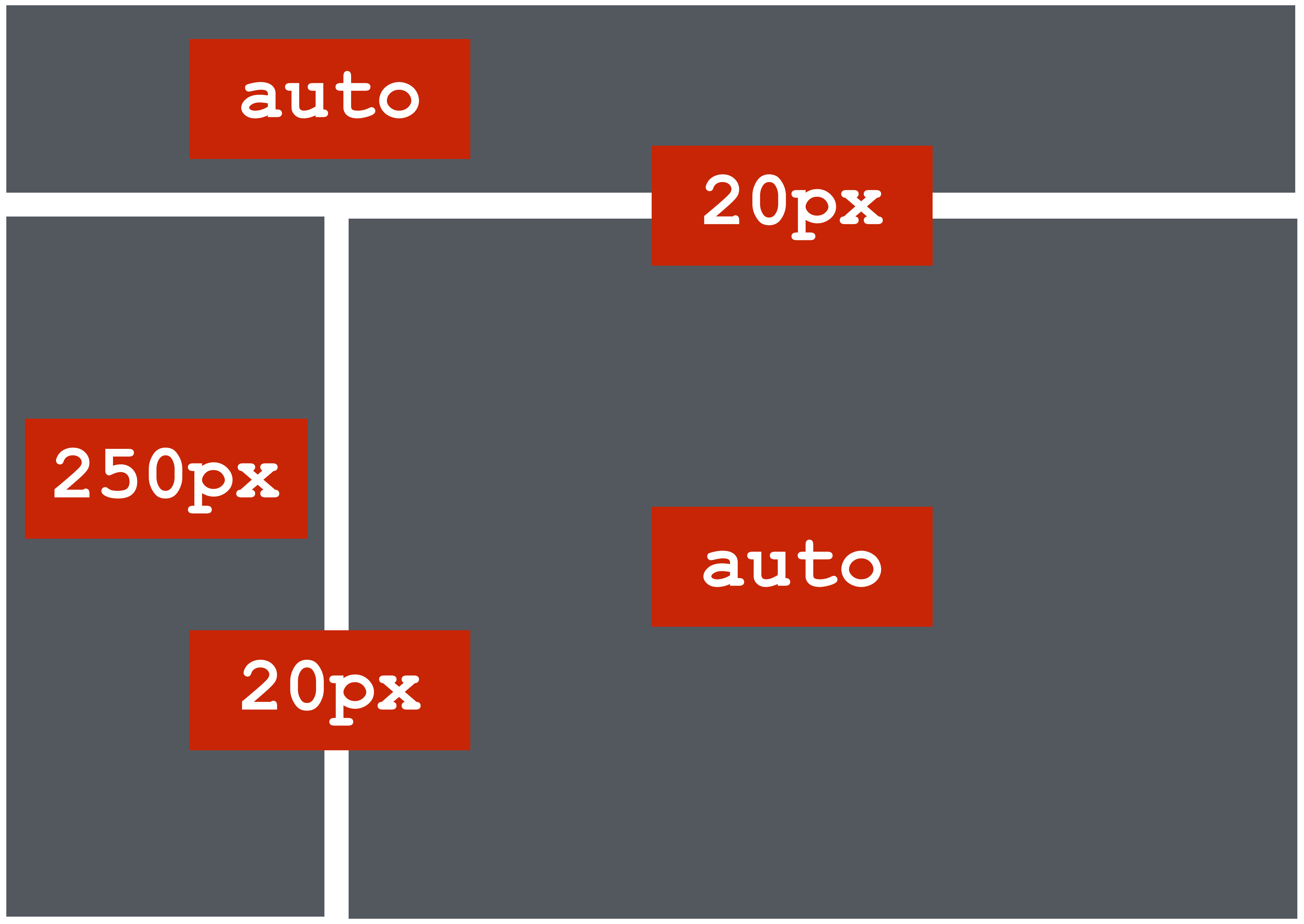
Grid cell

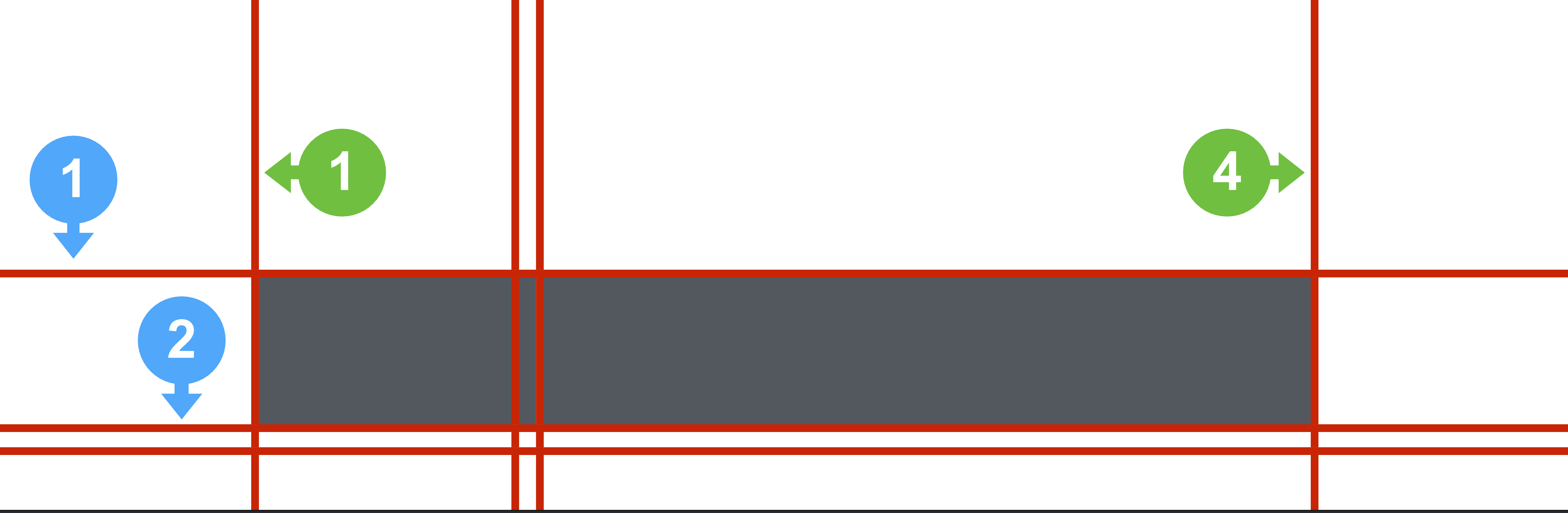


It is the **space between two adjacent row and two adjacent column grid lines**.

It is the smallest unit of the grid that can be referenced when positioning grid items

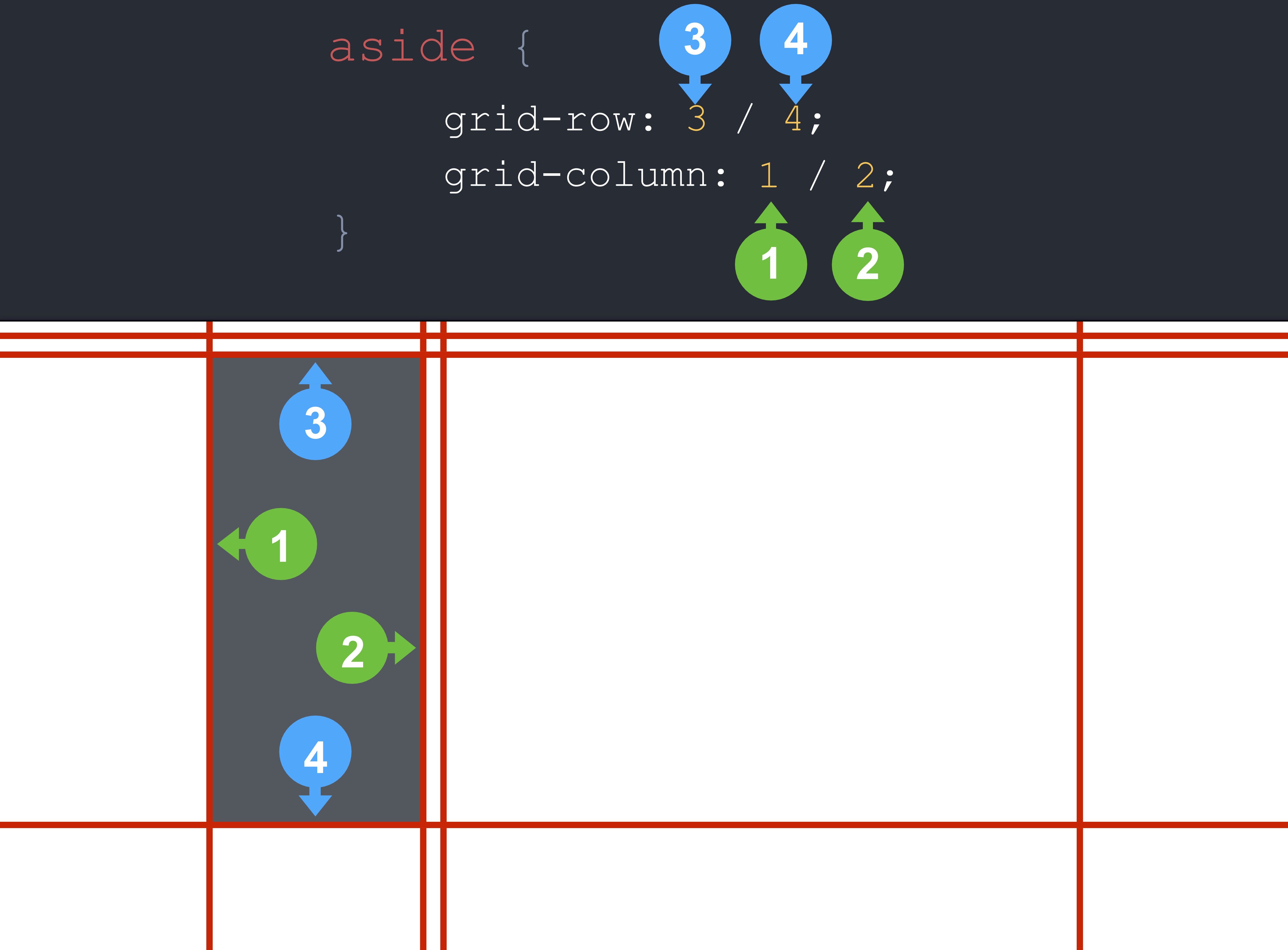
```
.main {

  /* Enable the grid space */
  display: grid;

  grid-template-rows: auto 20px auto;
  grid-template-columns: 250px 20px auto;

}
```

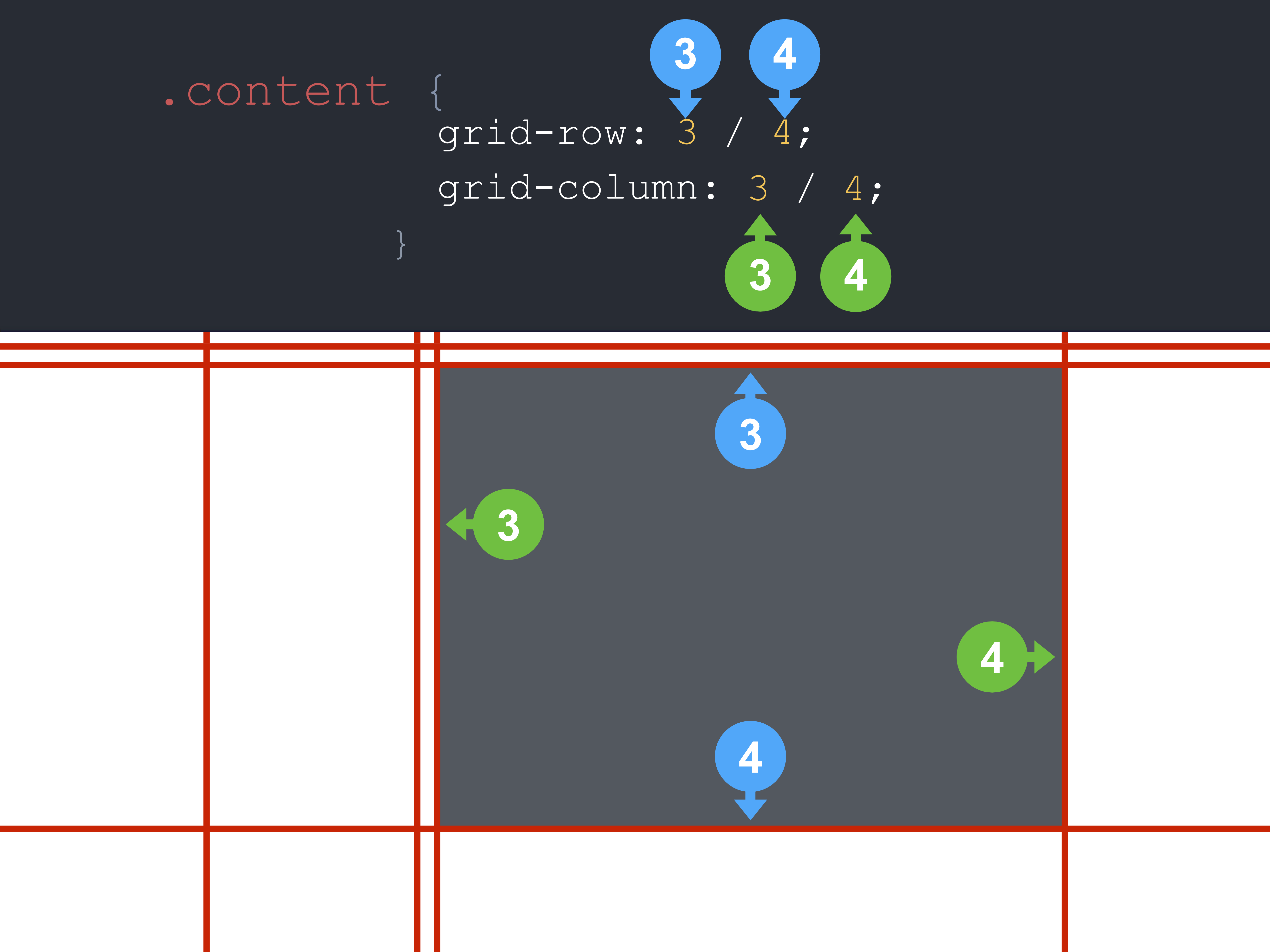

auto
20px
250px
auto
20px



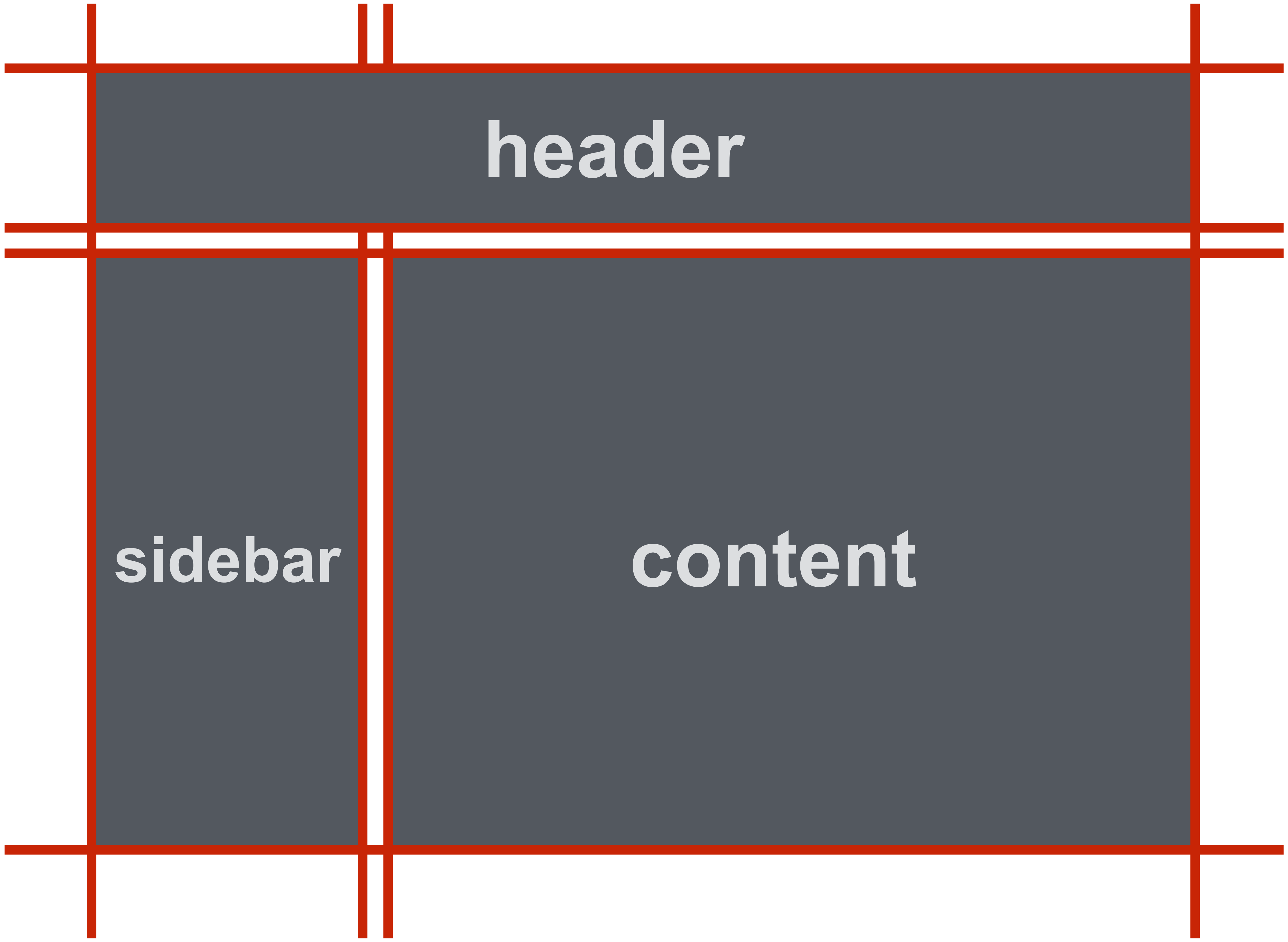
```
header {
  grid-row: 1 / 2;
  grid-column: 1 / 4;
}
```

```
aside {
        grid-row: 3 / 4;
        grid-column: 1 / 2;
}
```



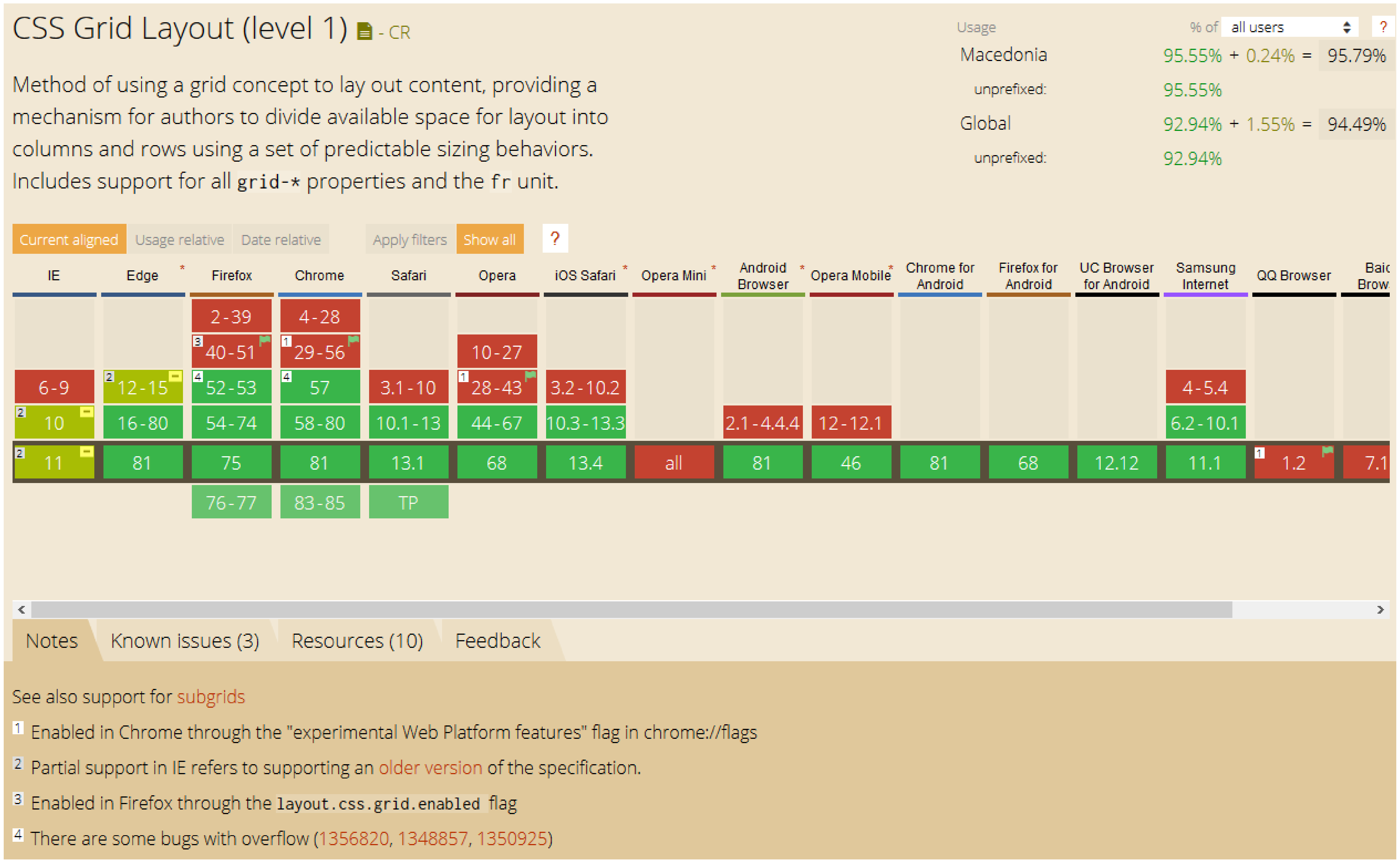


```
.content {
        grid-row: 3 / 4;
        grid-column: 3 / 4;
      }
```


header
sidebar
content

```
.main {

  display: grid;
  grid-template-rows: auto 20px auto;
  grid-template-columns: 250px 20px auto;

  grid-template-areas: “header header header"
                       “. . .”
                       “sidebar . article”

}
```

```css
header {
  grid-area: header;
}

aside {
  grid-area: sidebar;
}

article {
  grid-area: article;
}
```



CSS Grid Layout (level 1) - CR

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all `grid-*` properties and the `fr` unit.

Usage	% of all users			
Macedonia	95.55%	+ 0.24%	=	95.79%
unprefixed:	95.55%			
Global	92.94%	+ 1.55%	=	94.49%
unprefixed:	92.94%			

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baic Brow
		2-39	4-28												
		[3] 40-51	[1] 29-56		10-27										
6-9	[2] 12-15	[4] 52-53	[4] 57	3.1-10	[1] 28-43	3.2-10.2							4-5.4		
[2] 10	16-80	54-74	58-80	10.1-13	44-67	10.3-13.3		2.1-4.4.4	12-12.1				6.2-10.1		
[2] 11	81	75	81	13.1	68	13.4	all	81	46	81	68	12.12	11.1	[1] 1.2	7.1
		76-77	83-85	TP											

Notes | Known issues (3) | Resources (10) | Feedback

See also support for subgrids

[1] Enabled in Chrome through the "experimental Web Platform features" flag in chrome://flags

[2] Partial support in IE refers to supporting an older version of the specification.

[3] Enabled in Firefox through the `layout.css.grid.enabled` flag

[4] There are some bugs with overflow (1356820, 1348857, 1350925)

Source https://caniuse.com/#feat=css-grid

Grid Examples

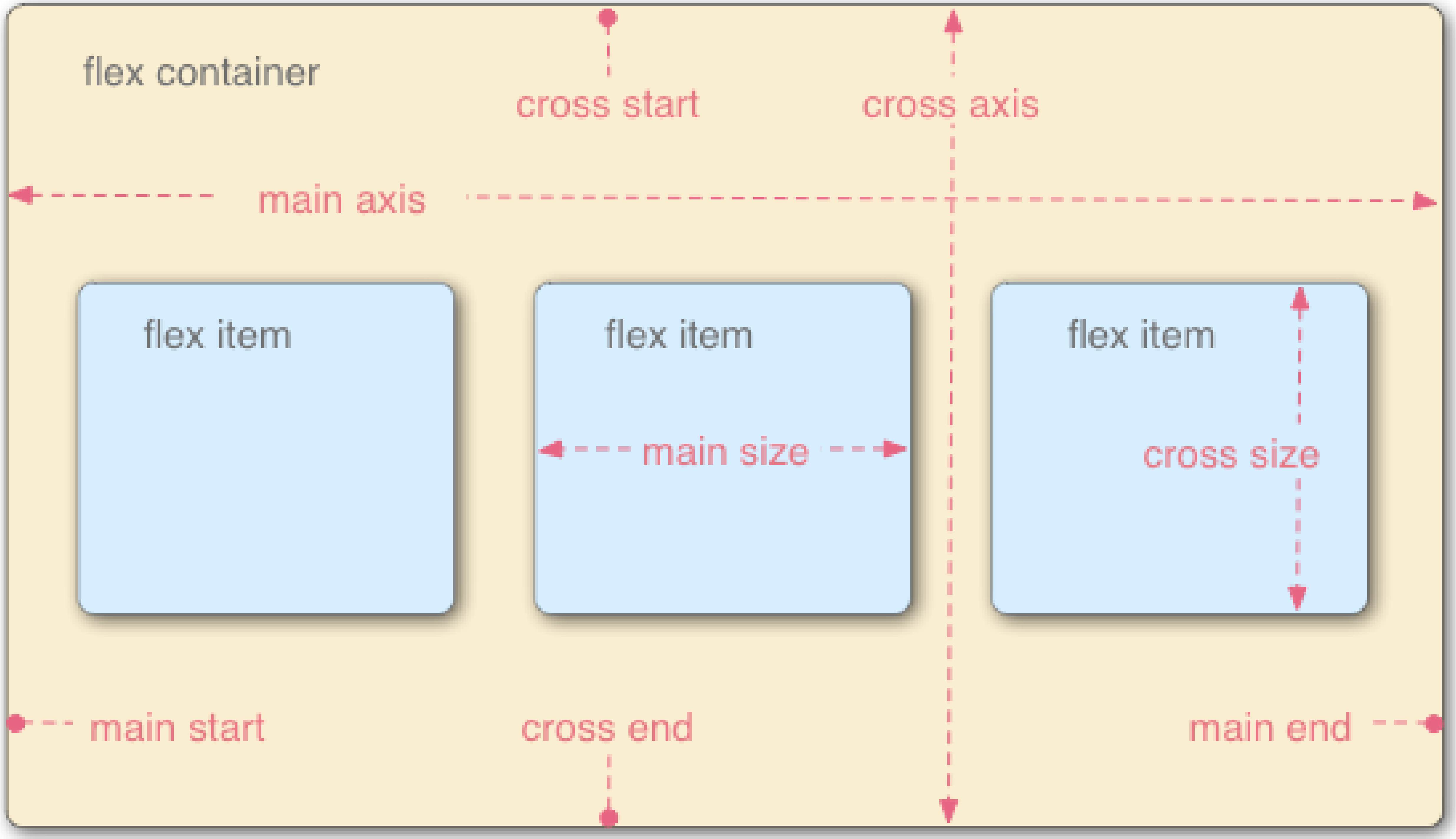
- More grid examples
  - http://gridbyexample.com

Flexbox

- Flexbox define how the child elemnets will <u>fill the blank space available</u> of parent element.

Flex Container

- First, we need to know the context where the flex items will work.
- This parent element where the flex items will work is called Flex Container.

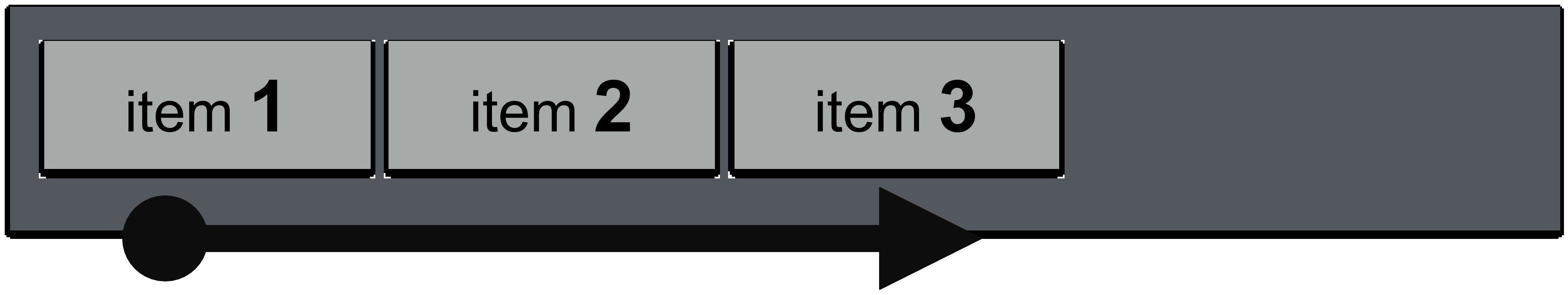


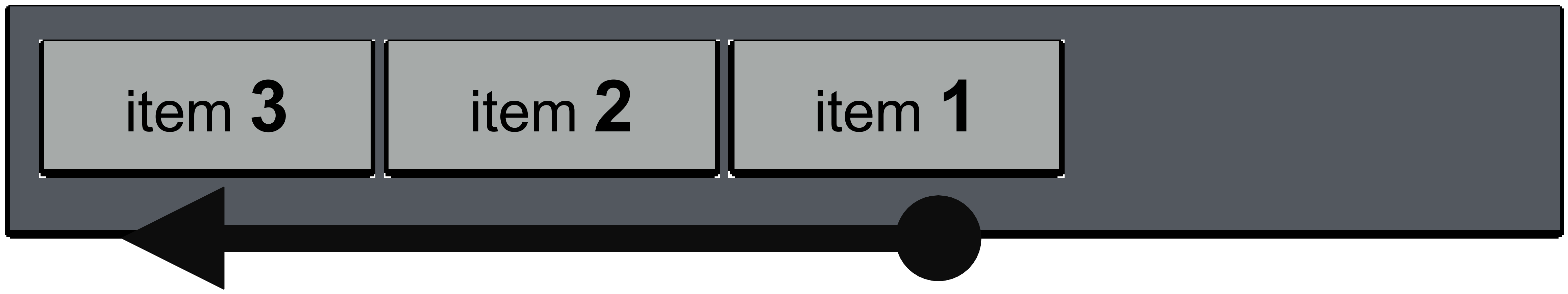
https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

flex-direction

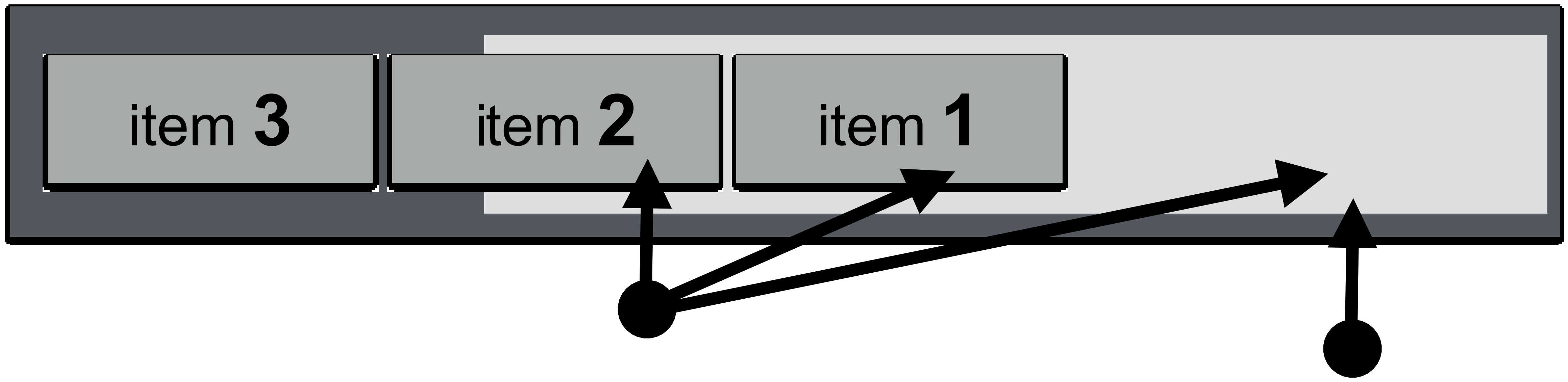
Define flow of the flex items placed in flex container.

flex-direction



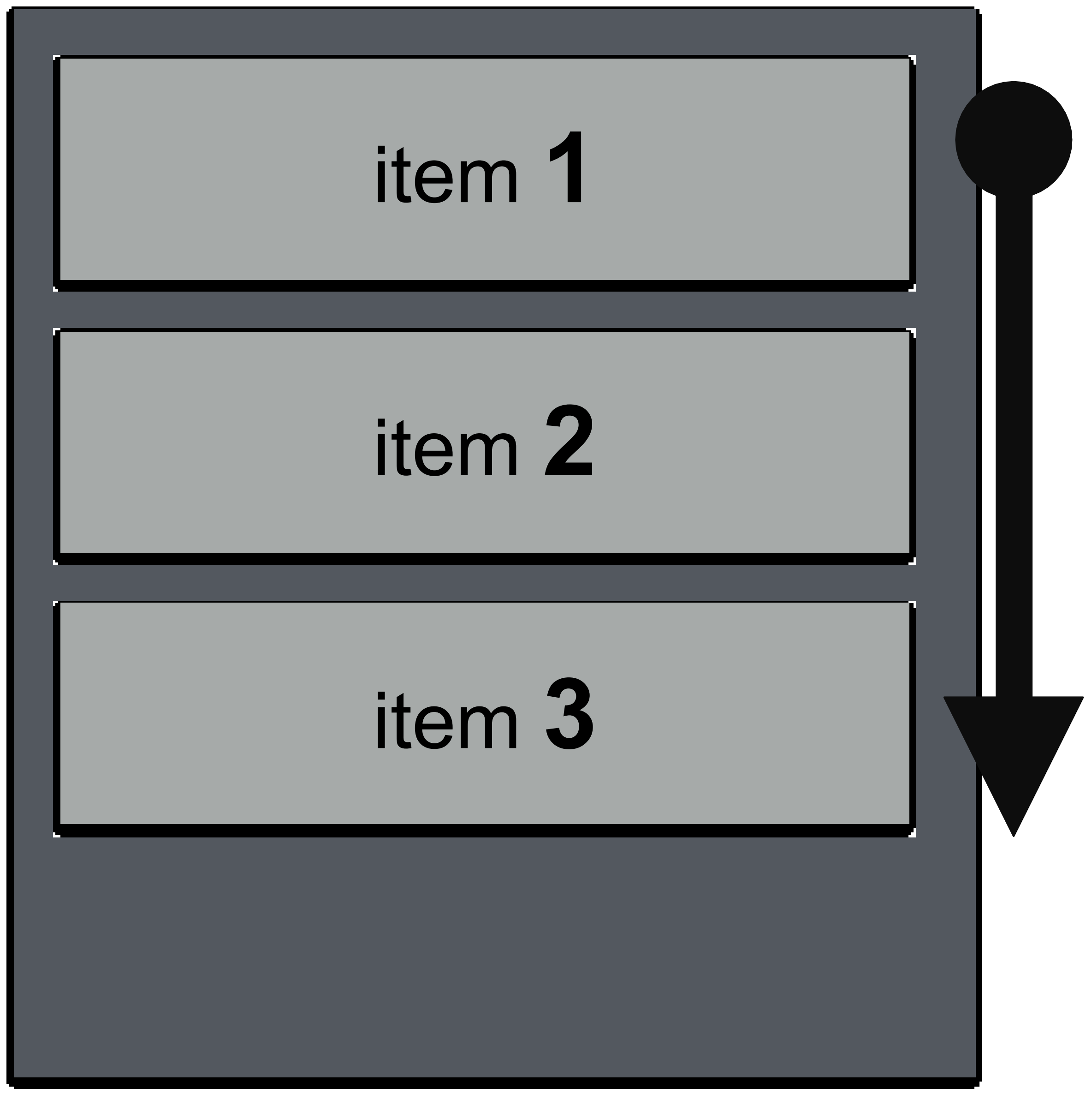


with float...

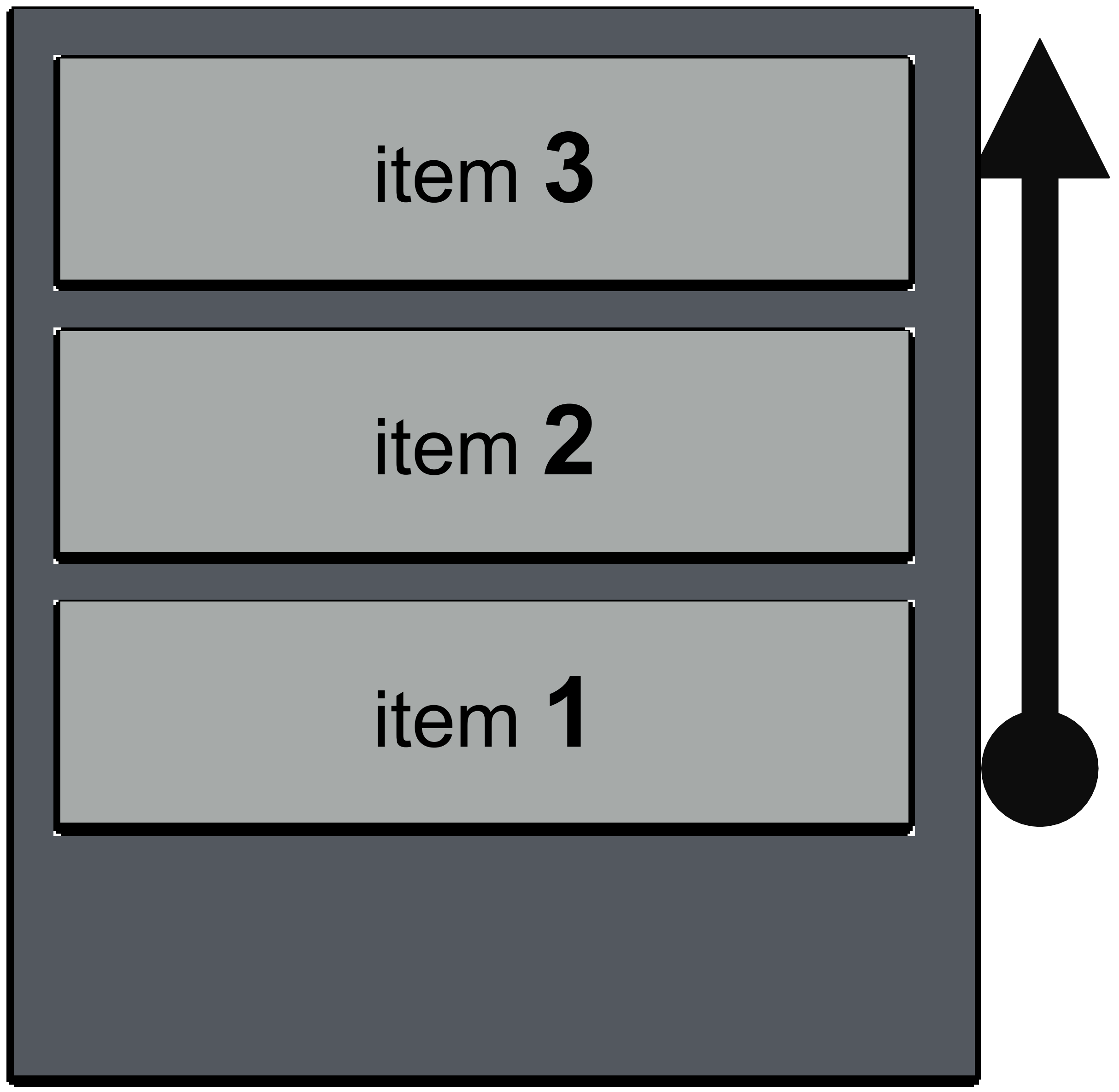


flex-direction

column



column-reverse

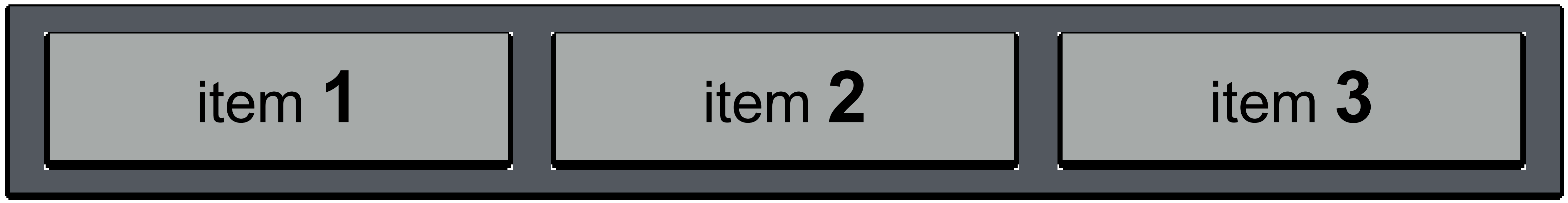


flex-wrap

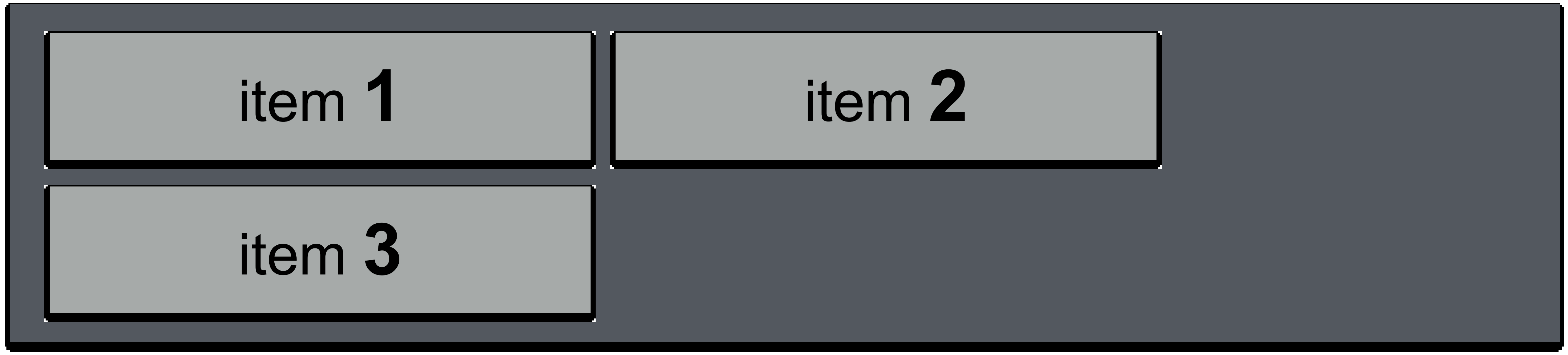
- Define if the flex items will break onto multiple lines if their width are larger than width of container.

nowrap

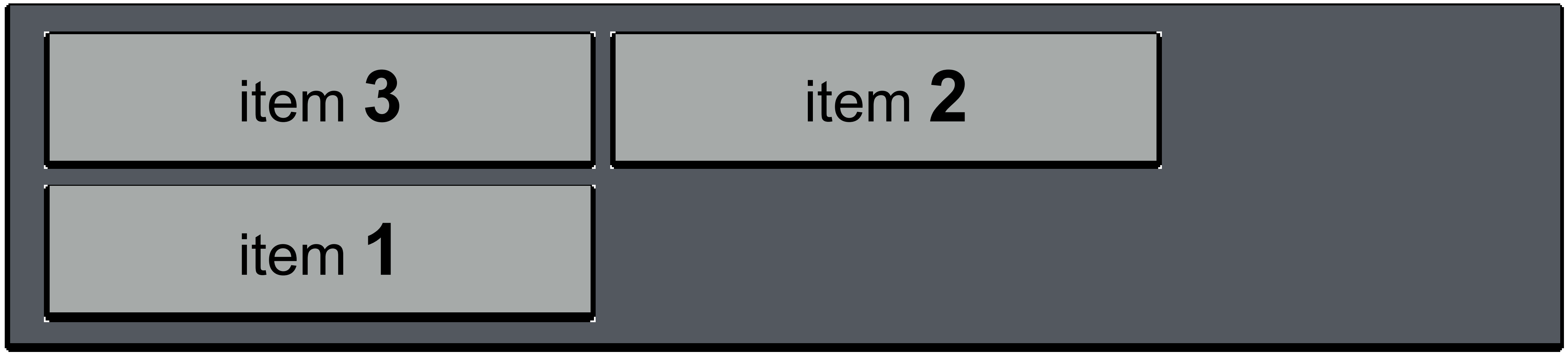
default



wrap



wrap-reverse



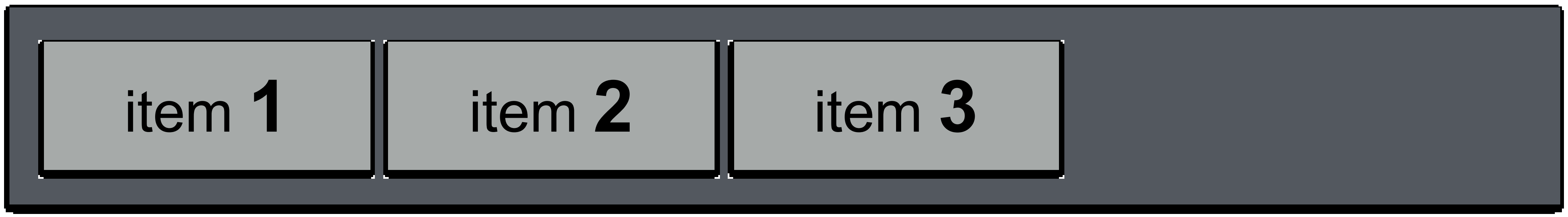
justify-content

Determine align of flex items in main-axis (**horizontal line**).

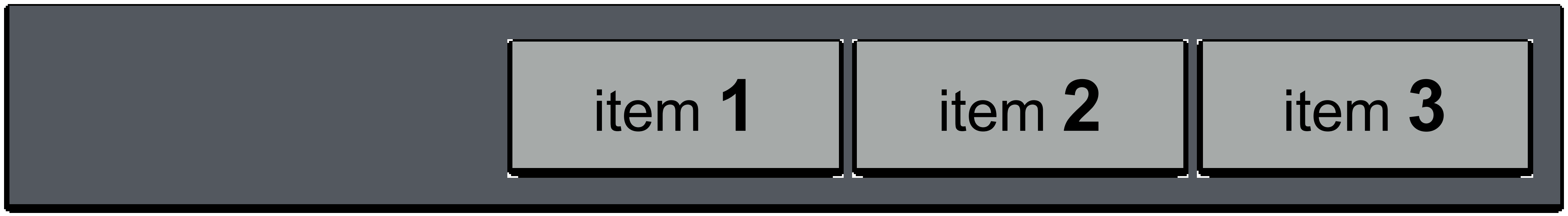
justify-content

flex-start

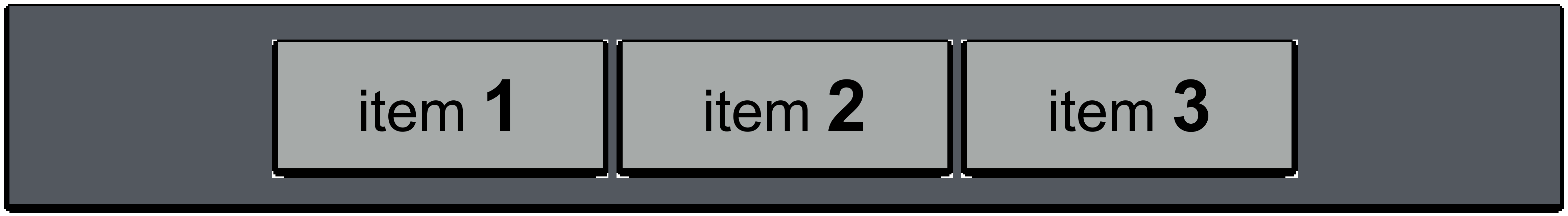
default



flex-end

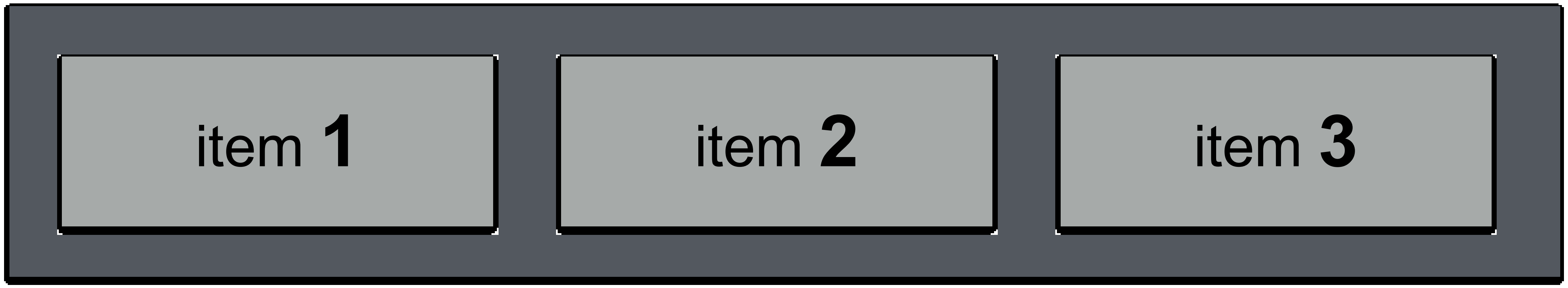


center

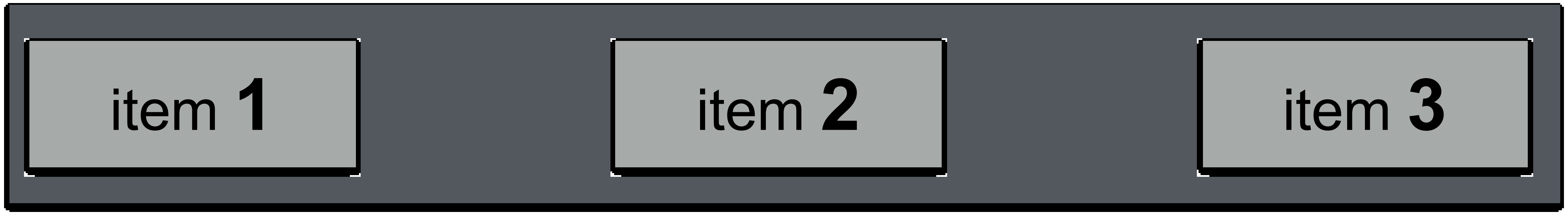


justify-content

space-around



space-between



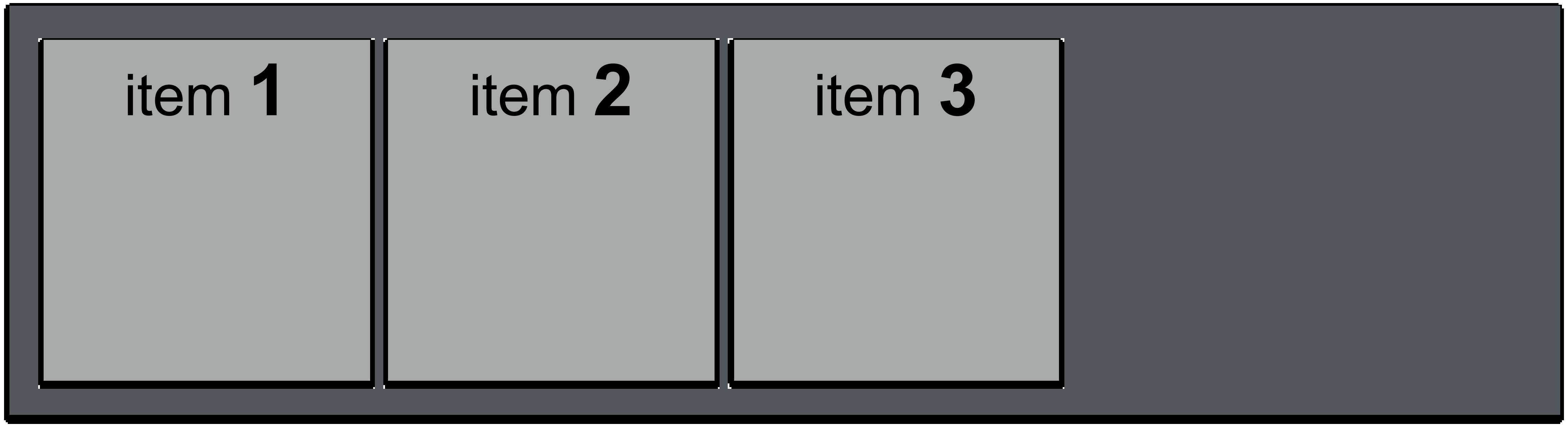
align-items

Determine align of flex items in cross-axis (**vertical line**).

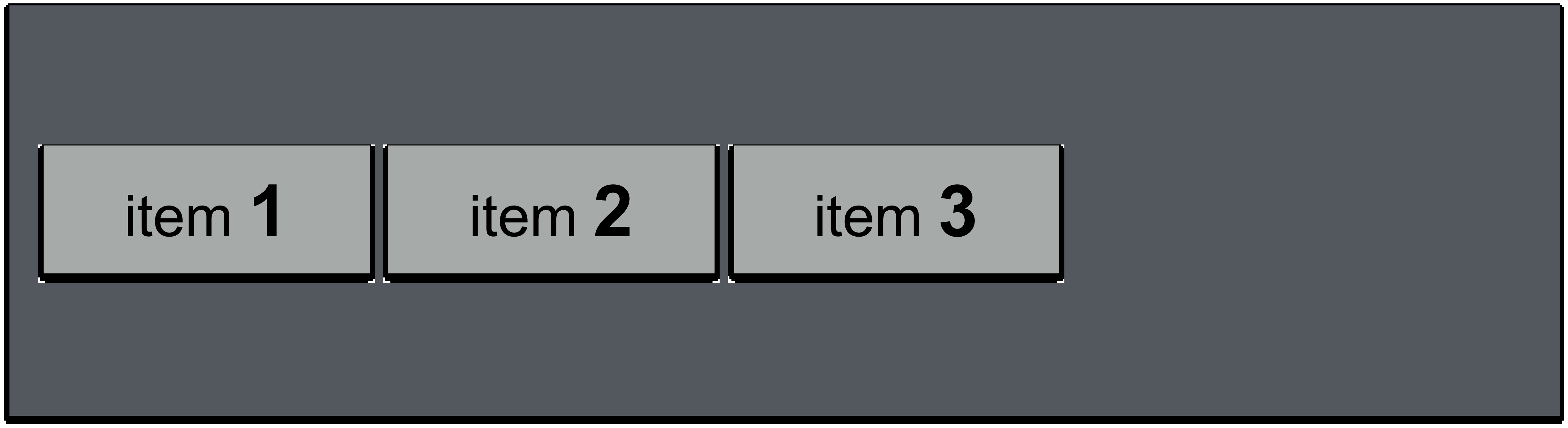
align-items

stretch

default

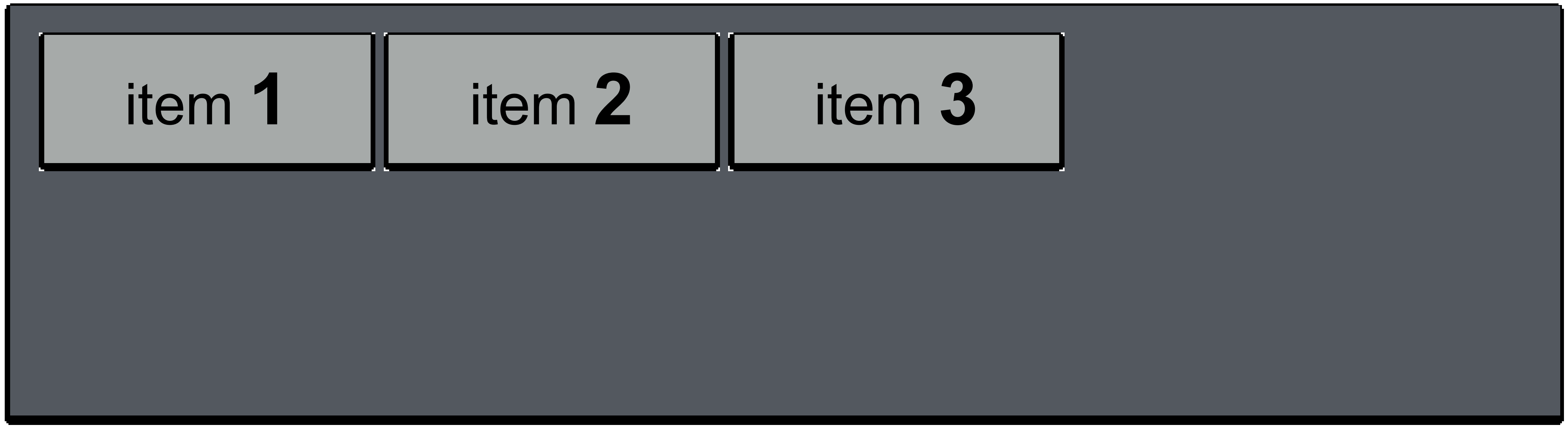


center

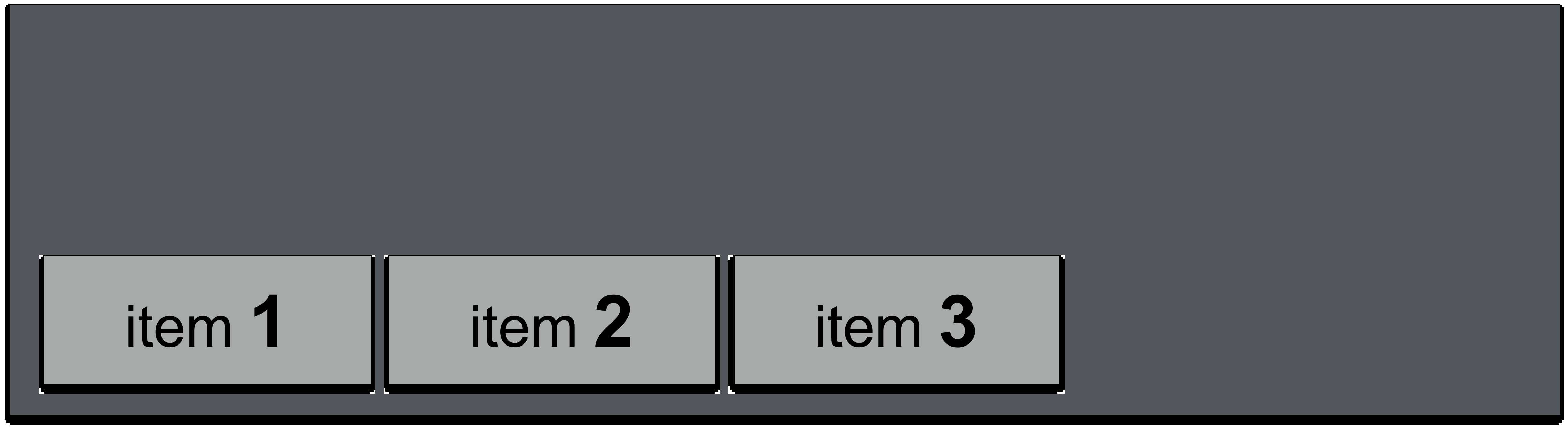


align-items

flex-start

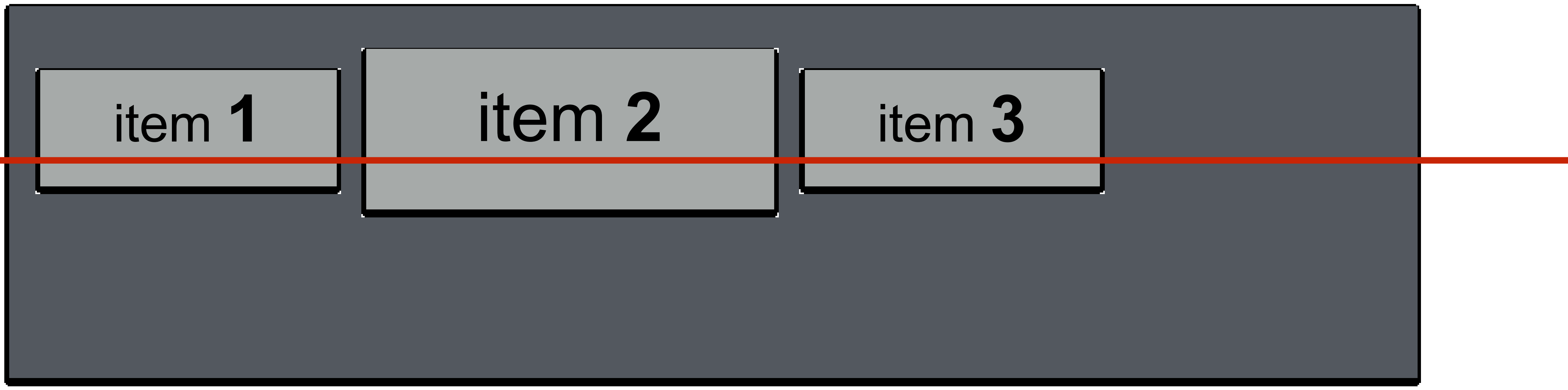


flex-end



align-items

baseline

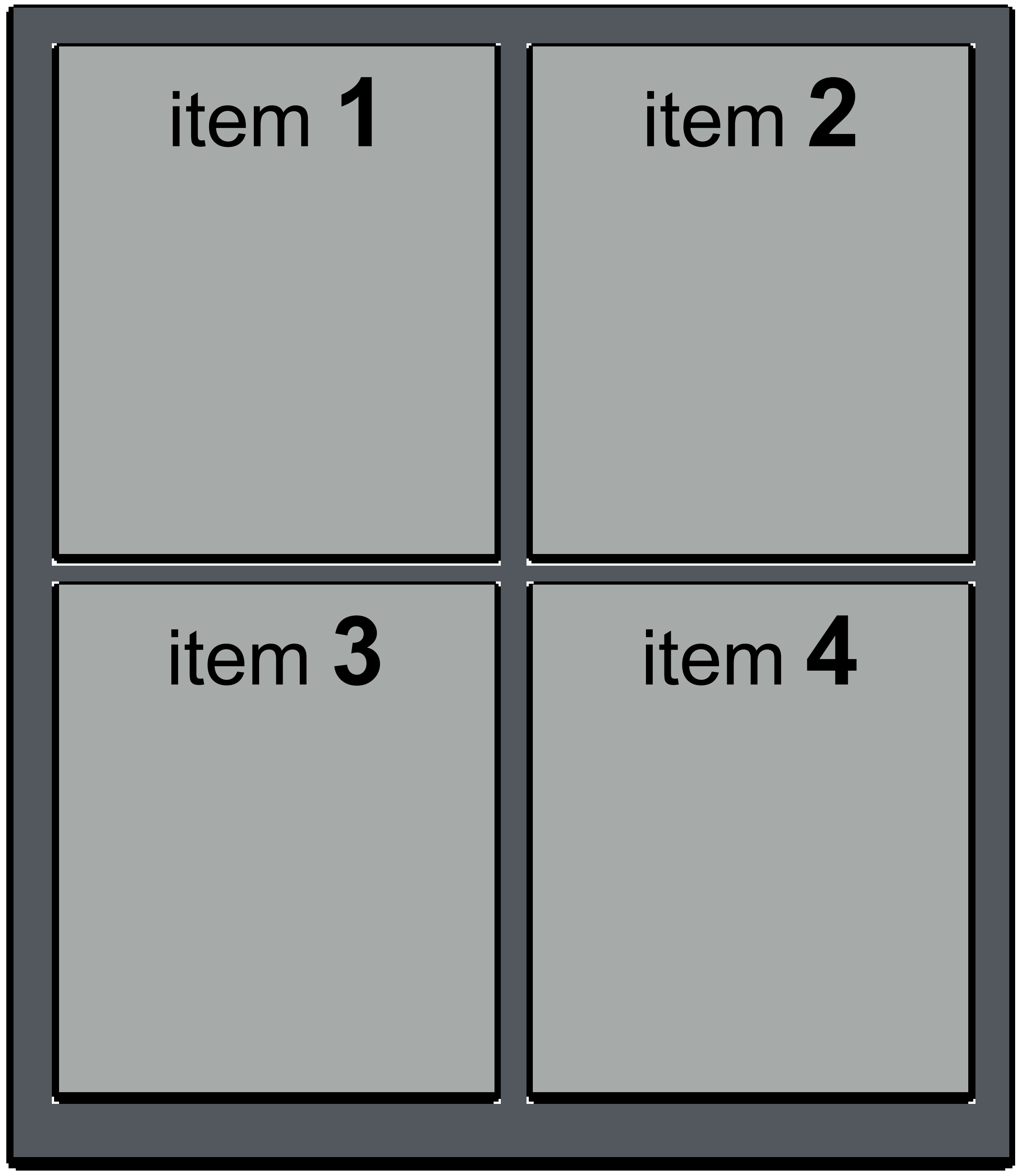


align-content

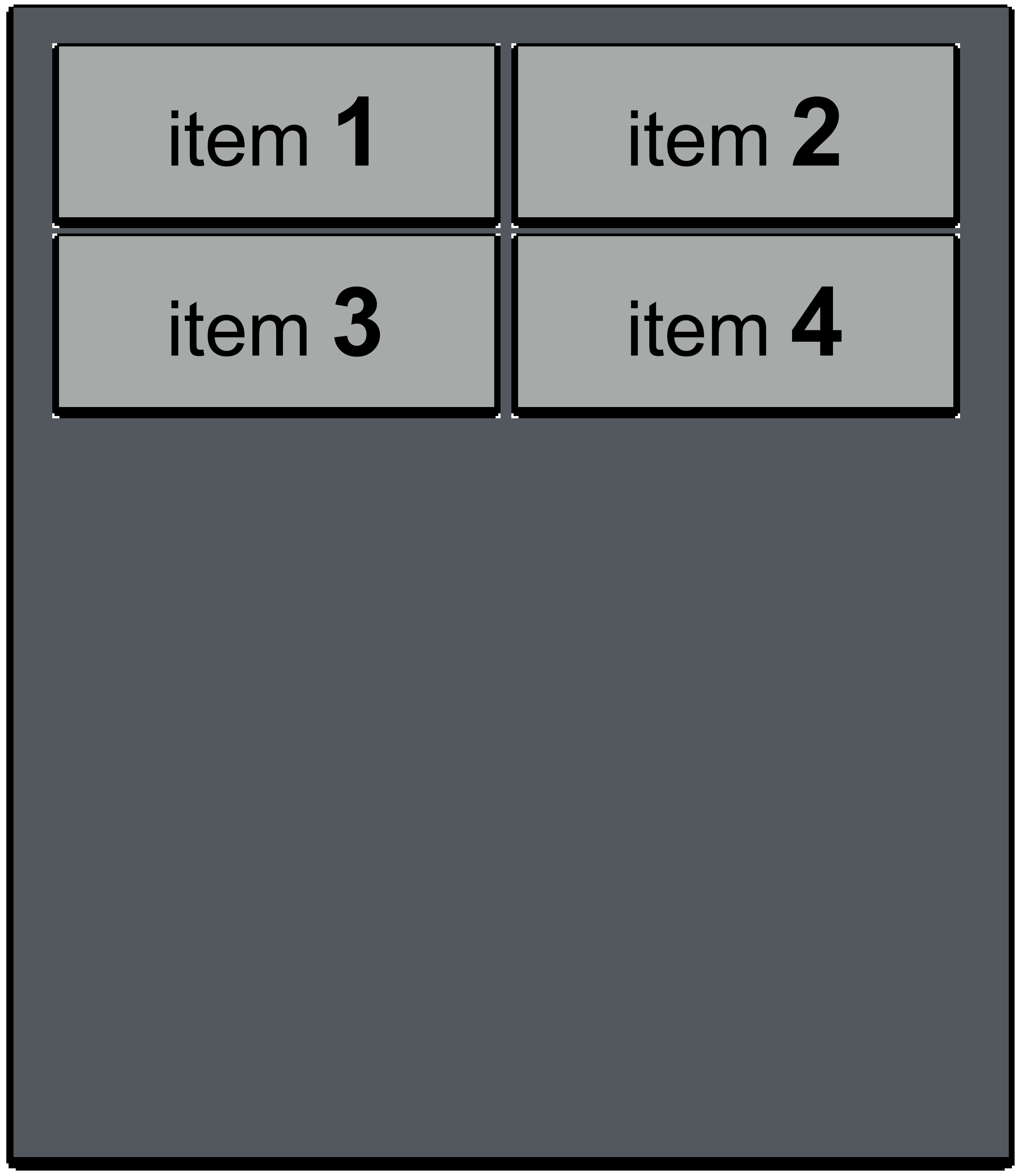
Align flex items with extra space on the **cross-axis,** within the flex container **when there are multiple lines.**

align-content

stretch

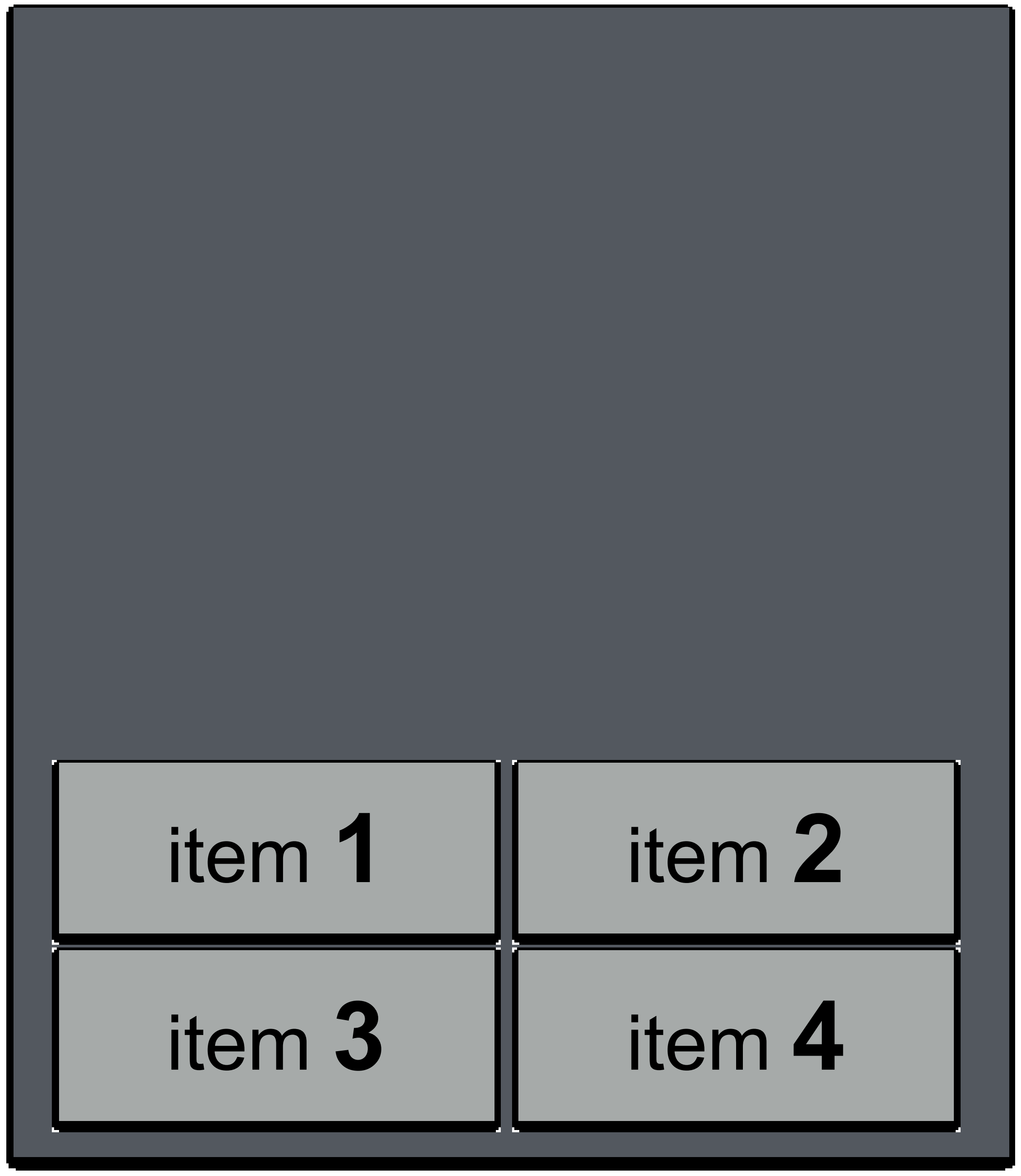


flex-start

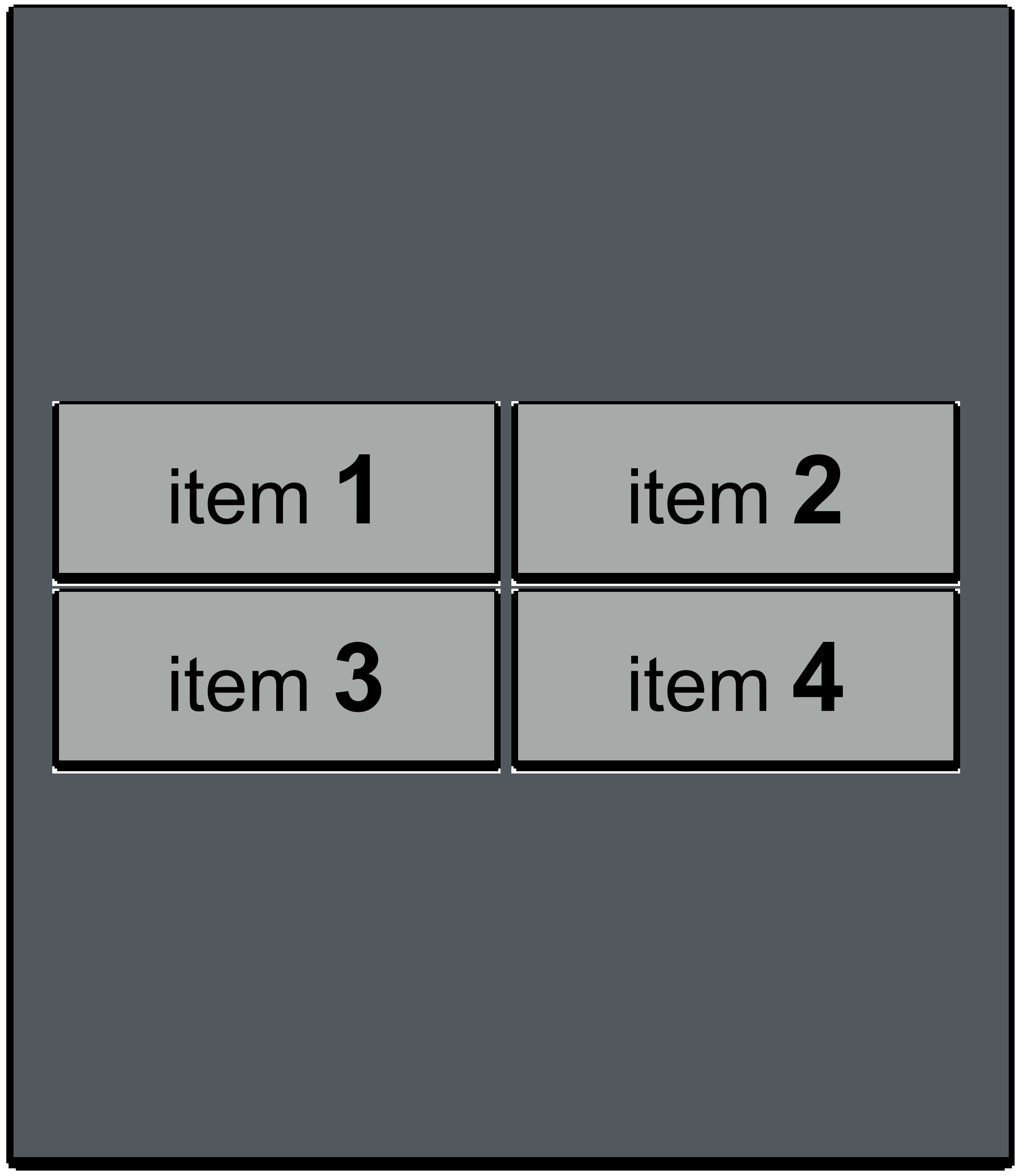


align-content

flex-end

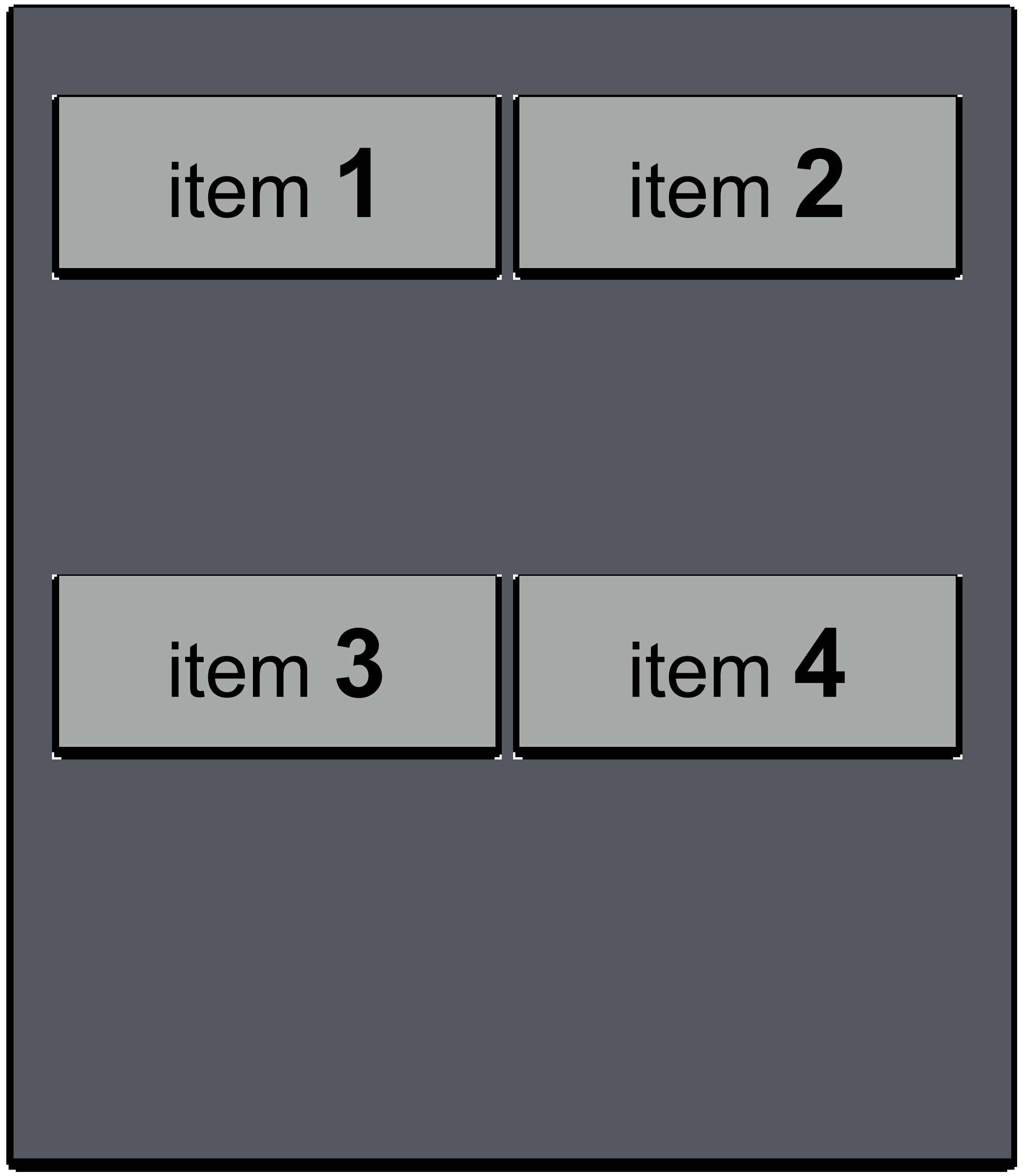


center

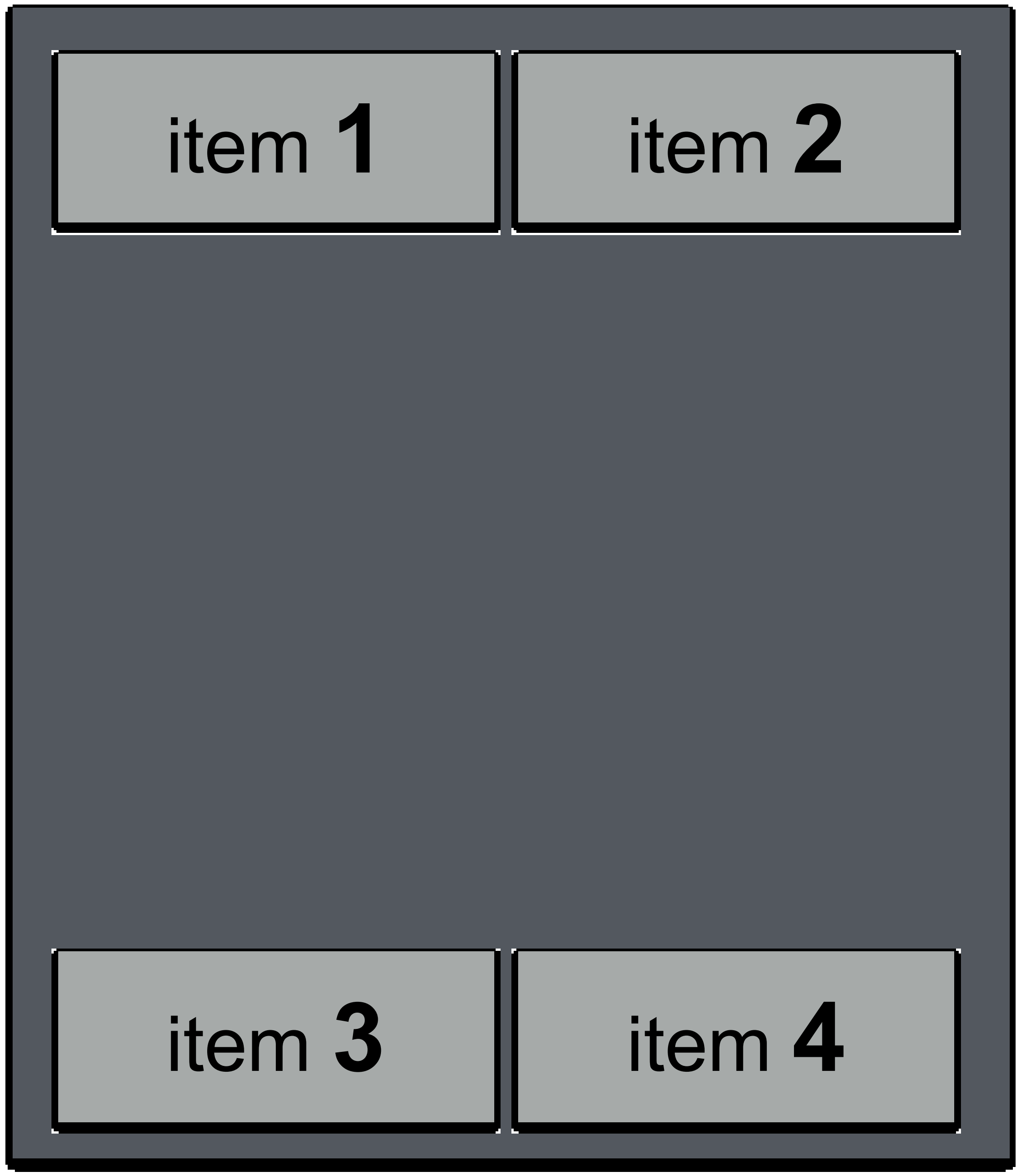


align-content

space-around



space-between

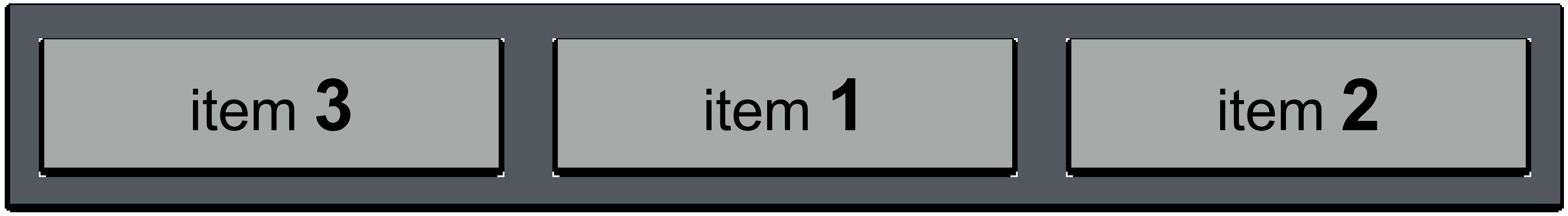


Flex items

order

Reorders appearance HTML elements.

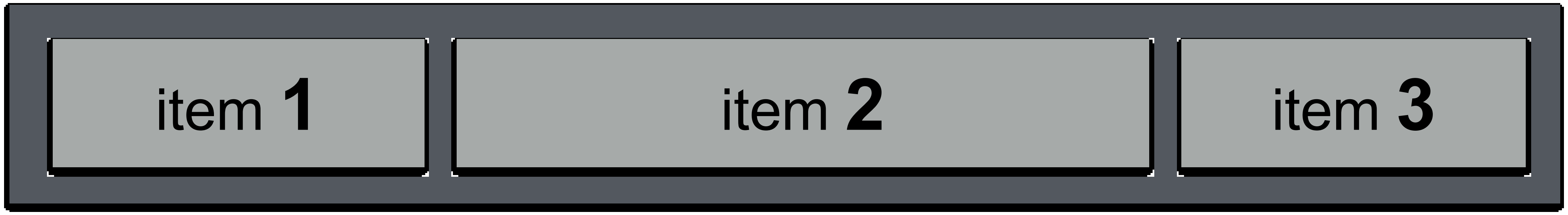
```css
.item1 { order: 2; }    .item2 { order: 3; }    .item3 { order: 1; }
```



flex-grow

Define how much the item will take of available space. The value serves as a proportion. If all elements have 1 of value, all elements will have same width. If one element have 2 of value, that element will have the double of size.

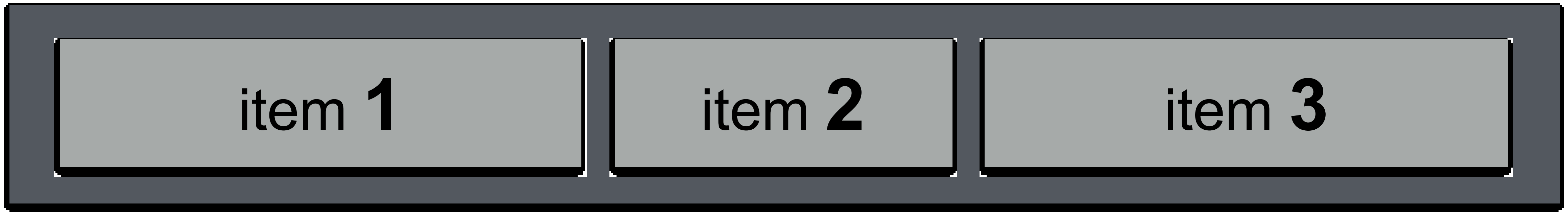
```
.item2   { flex-grow: 2; }
```



flex-shrink

Define how much the item will shrink.

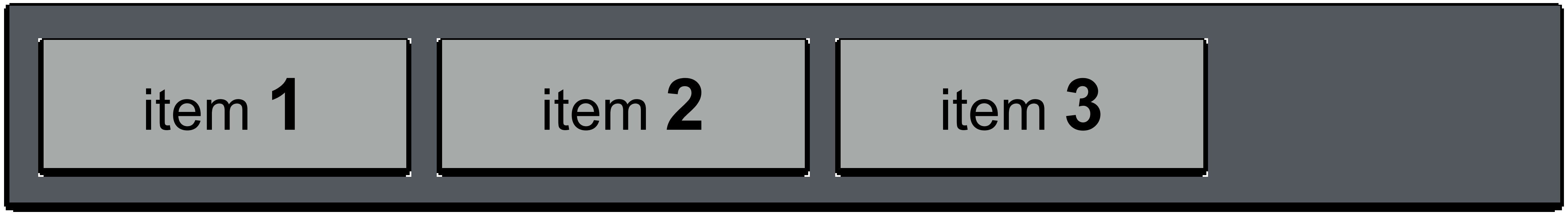
```
.item2  { flex-shrink: 2; }
```



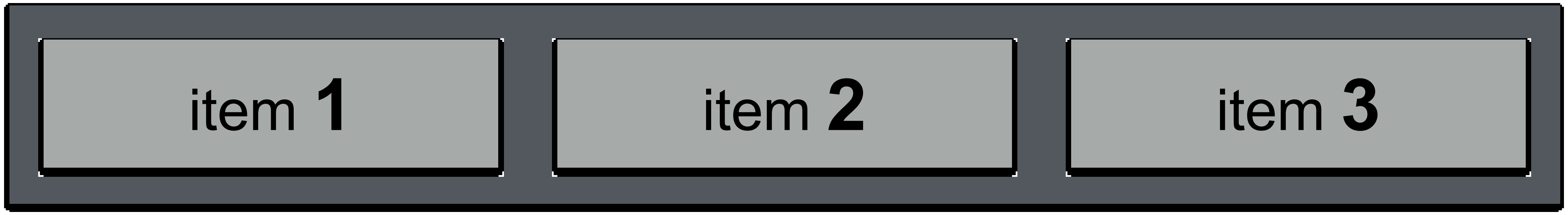
flex-basis

Define the width of elements. This specifies the initial length of a flex item.

```
.item   { flex-basis: 100px; }
```



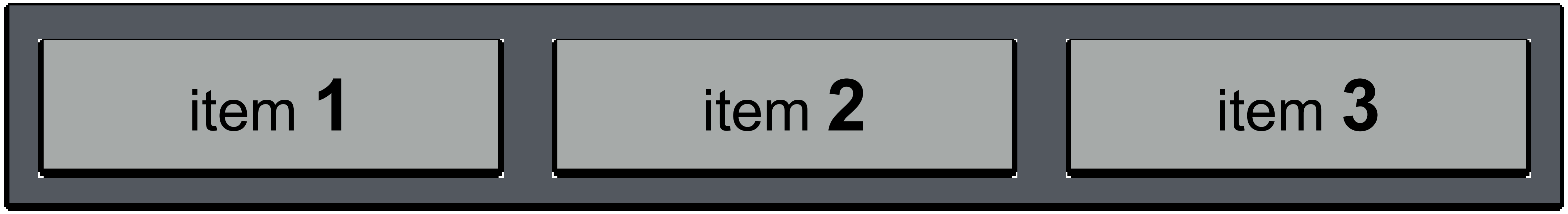
```
.item { flex-basis: 100%; }
```



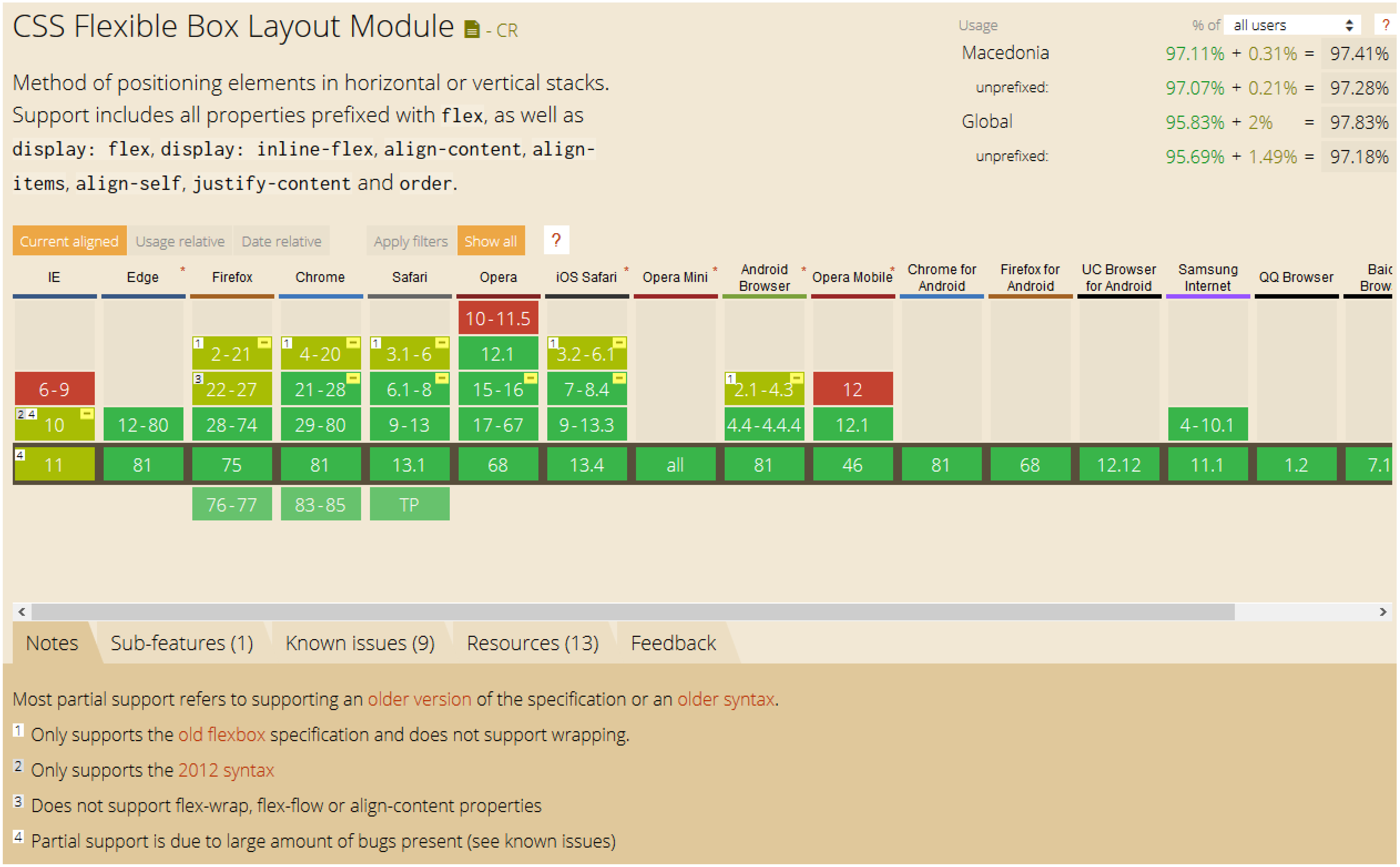
flex

Shorthand to combine all properties.

```
.item   { flex: 1; }
```



```
.item {
 flex-grow: 1;
 flex-shrink:1;
 flex-basis: auto;
}
```



CSS Flexible Box Layout Module - CR

Method of positioning elements in horizontal or vertical stacks. Support includes all properties prefixed with `flex`, as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Usage % of all users ?

Macedonia	97.11%	+ 0.31%	=	97.41%
unprefixed:	97.07%	+ 0.21%	=	97.28%
Global	95.83%	+ 2%	=	97.83%
unprefixed:	95.69%	+ 1.49%	=	97.18%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baic Brow
					10-11.5										
		[1] 2-21	[1] 4-20	[1] 3.1-6	12.1	[1] 3.2-6.1									
6-9		[3] 22-27	21-28	6.1-8	15-16	7-8.4		[1] 2.1-4.3	12						
[2] [4] 10	12-80	28-74	29-80	9-13	17-67	9-13.3		4.4-4.4.4	12.1				4-10.1		
[4] 11	81	75	81	13.1	68	13.4	all	81	46	81	68	12.12	11.1	1.2	7.1
		76-77	83-85	TP											

Most partial support refers to supporting an older version of the specification or an older syntax.

[1] Only supports the old flexbox specification and does not support wrapping.

[2] Only supports the 2012 syntax

[3] Does not support flex-wrap, flex-flow or align-content properties

[4] Partial support is due to large amount of bugs present (see known issues)

Source https://caniuse.com/#feat=flexbox

Questions?