avionschool

Lesson 3.0 Responsive Web Design

BATCH 4

NOVEMBER 16, 2020

Article Tribute Page Presentation

Whether it's a full-fledged dynamic SPA, or a CMS-reliant website of static pages, users of today will probably want to see how the web app looks like on a mobile screen.

The number of smartphone users in the world is 3.5 billion - 44.87% of the world's population

Philippines spends 5 hours and 11 minutes on mobile internet per day - more than any other country in the world

- SPA - Single Page Application
- CMS - Content Management System

How do we support all these different screen sizes?

Responsive Web Design

- makes your web page look good on all devices, not a program or JavaScript, uses only HTML and CSS to resize, hide, shrink, enlarge, or move the content to make it look good on any screen, designing for the best experience for all users - W3Schools

- the practice of building a website suitable to work on every device and every screen size, no matter how large or small, mobile or desktop, focusing around providing an intuitive and gratifying experience for everyone. - learn.shayhowe.com

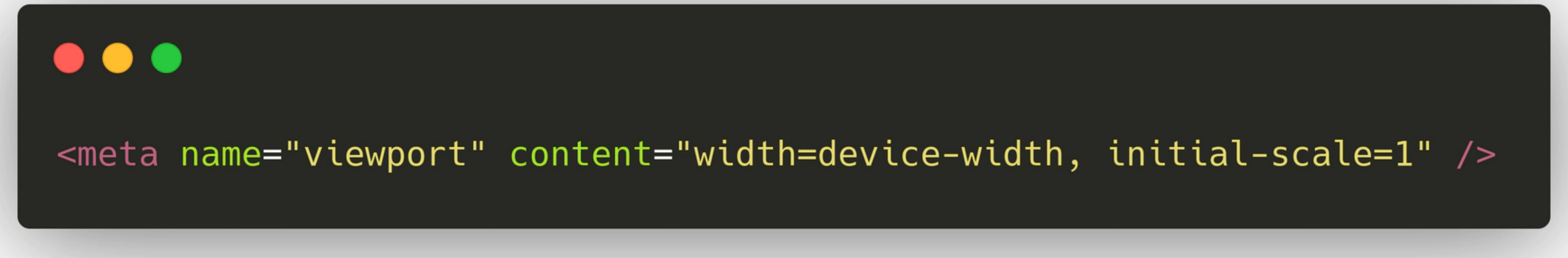
- approach that suggests that design and development should respond to the user's behavior and environment based on screen size, platform and orientation. - Smashing Magazine

- Rather than tailoring disconnected designs to each of an ever-increasing number of web devices, we can treat them as facets of the same experience. We can design for an optimal viewing experience, but embed standards-based technologies into our designs to make them not only more flexible, but more adaptive to the media that renders them. - Ethan Marcotte

Responsive Web Design

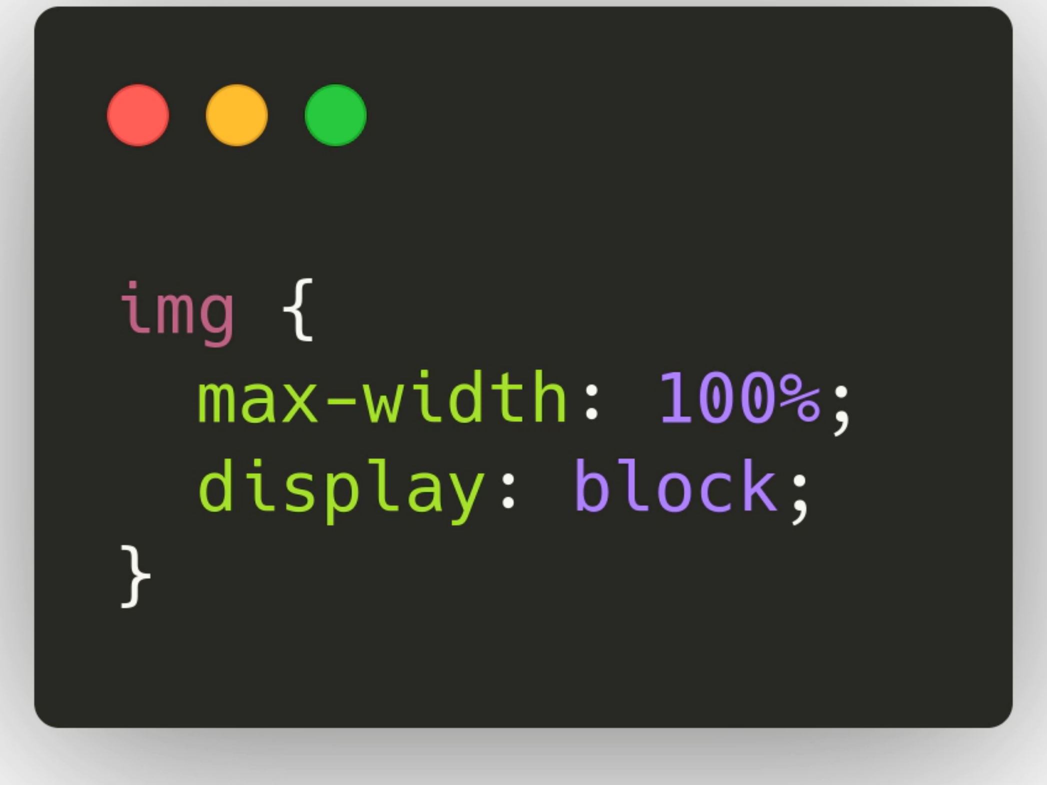
1. Flexible Images
2. Fluid Layout
3. Media Queries

Setting the Viewport



```html
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Flexible Images



```css
img {
  max-width: 100%;
  display: block;
}
```

Optimizing Font

1. 1em = 12pt = 16px = 100%
2. target / context = result

Fluid Layouts

Flexbox

DEFINITION

- A CSS3 layout mode that provides an efficient way to align and distribute space among items in a container, even when their size is unknown and/or dynamic.

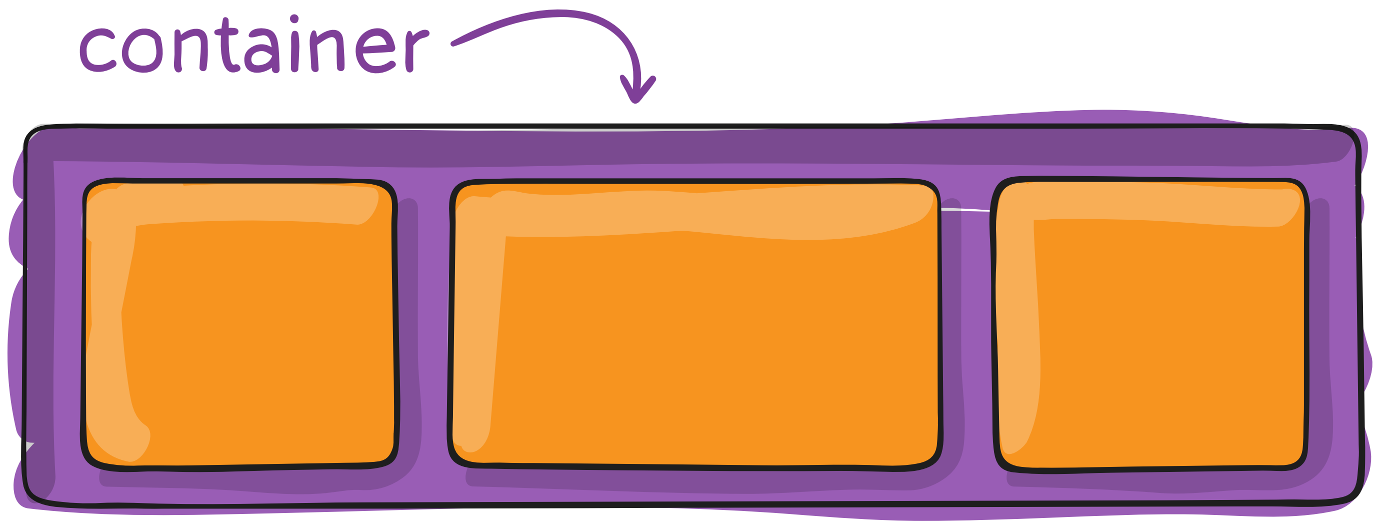
ADVANTAGES

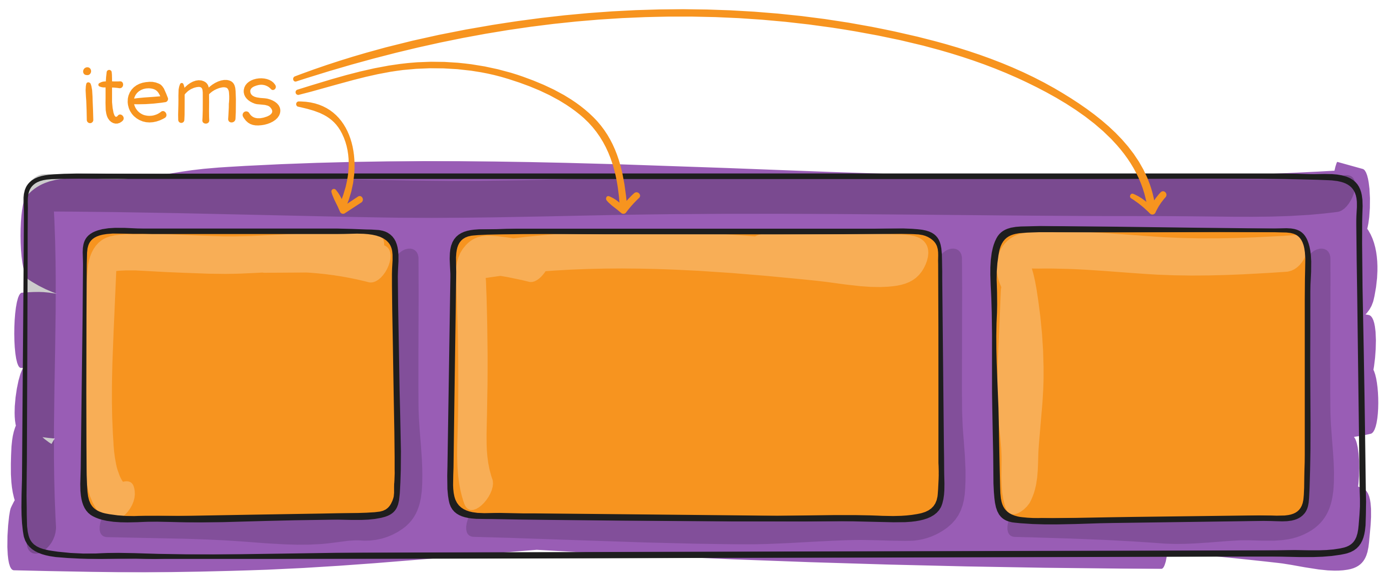
- Responsive and mobile-friendly
- Positioning child elements is much easier
- Flex container's margins do not collapse with the margins of its contents.
- Order of elements can easily be changed without editing the source HTML

Basics & Terminology

DEFINITIONS

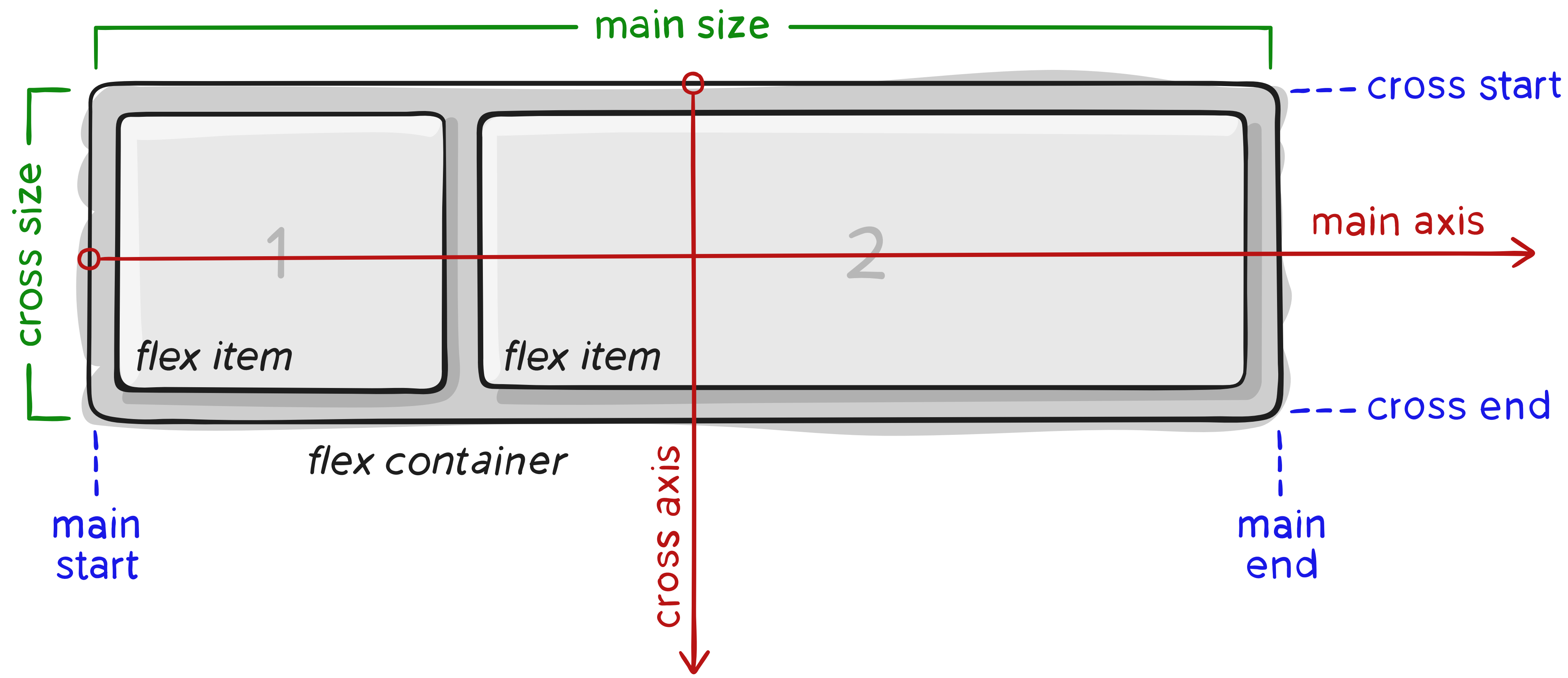
1. Container (Parent Element; Flex Container)
2. Children (Child Element; Flex Items)





Basics & Terminology

FLEX-FLOW DIRECTIONS



Items are laid out **two ways**:

1. Main axis (Left to right)
2. Cross axis (Top to bottom)

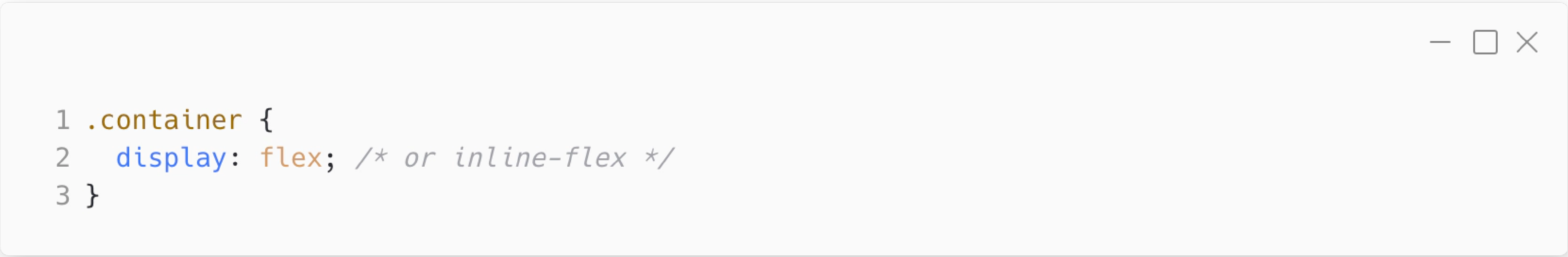
Properties for the Parent (flex container)

display

This enables a flex context for all its direct children.

We can define the flex container's type in two ways:

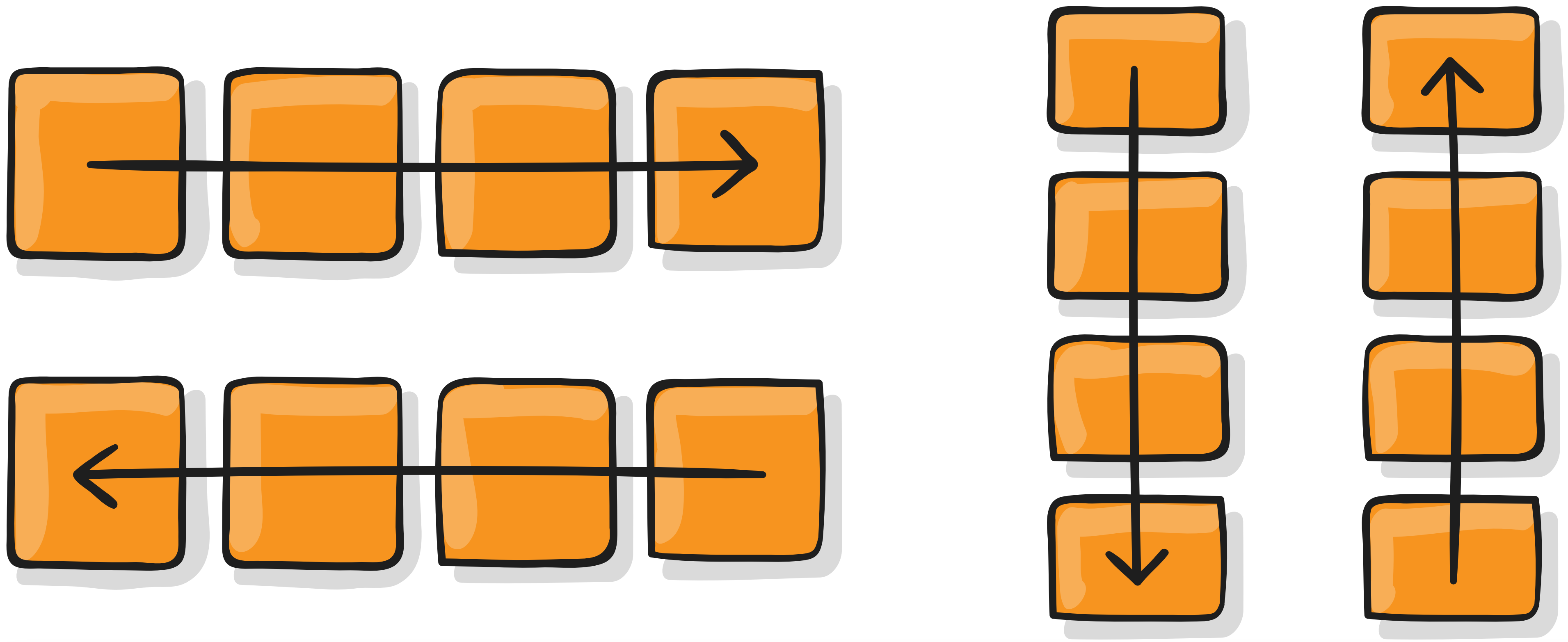
1. inline
2. block

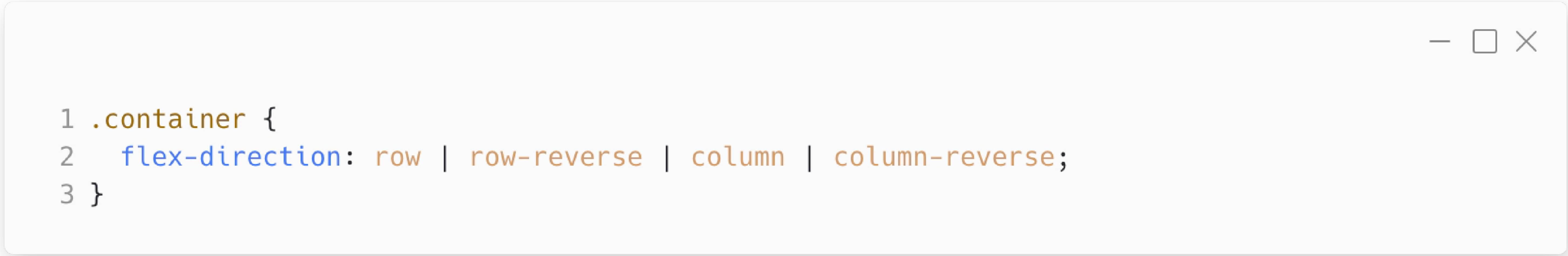


```css
.container {
  display: flex; /* or inline-flex */
}
```

Properties for the Parent (flex container)

flex-direction

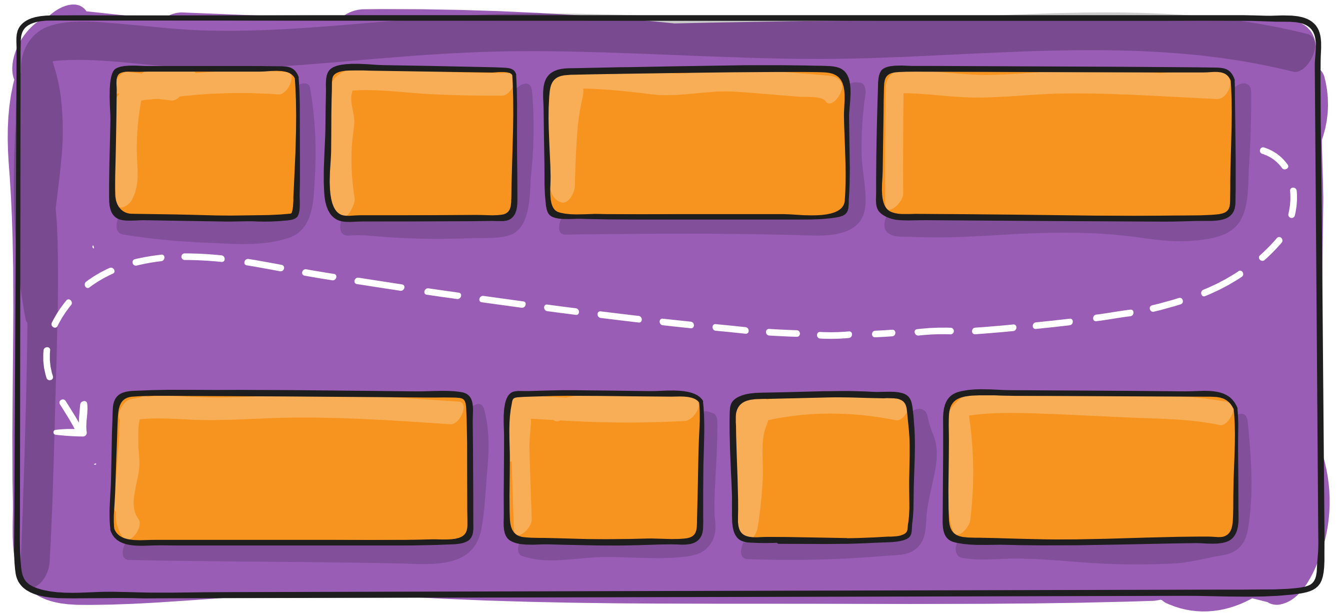


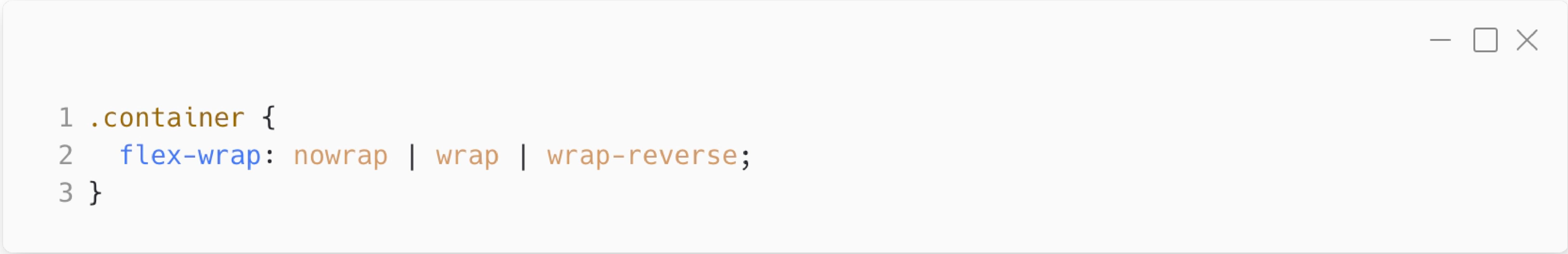


```css
.container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```

Properties for the Parent (flex container)

flex-wrap

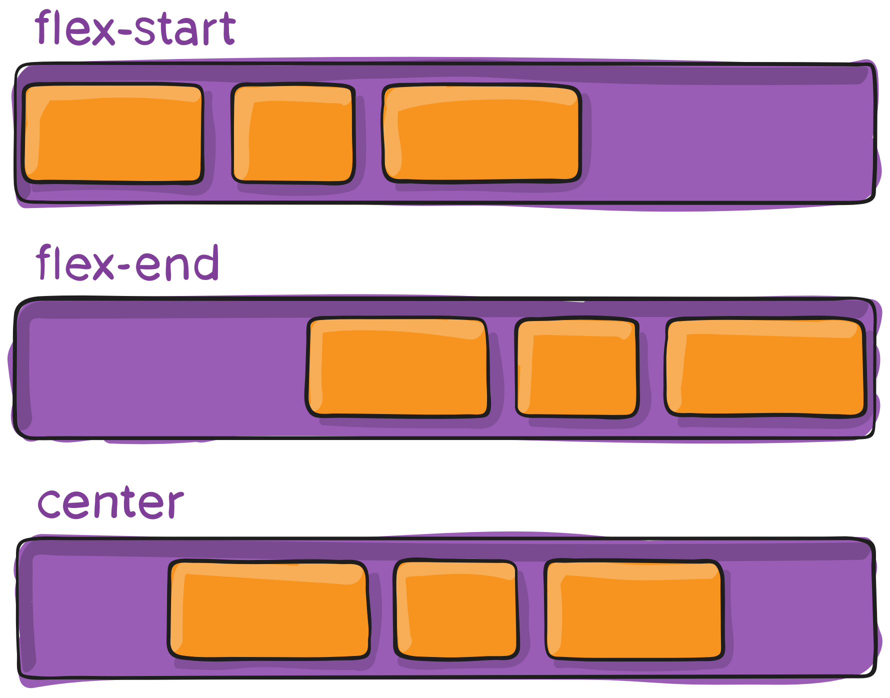


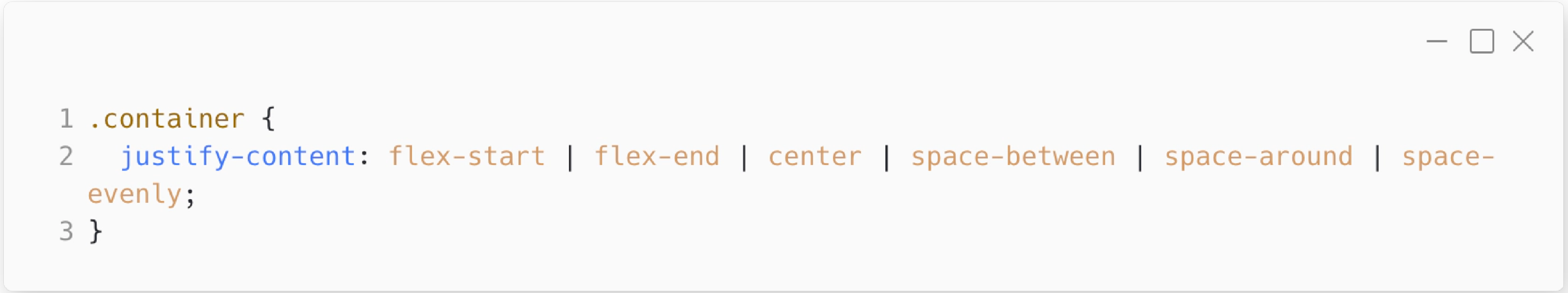


```css
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

Properties for the Parent (flex container)

justify-content

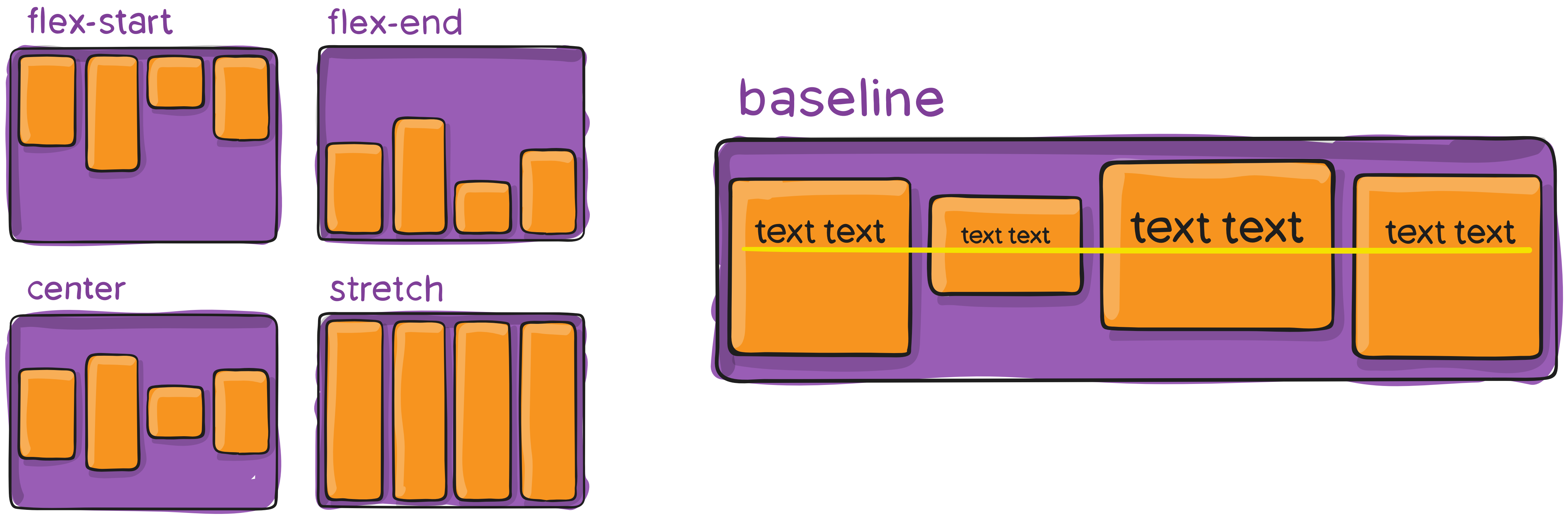


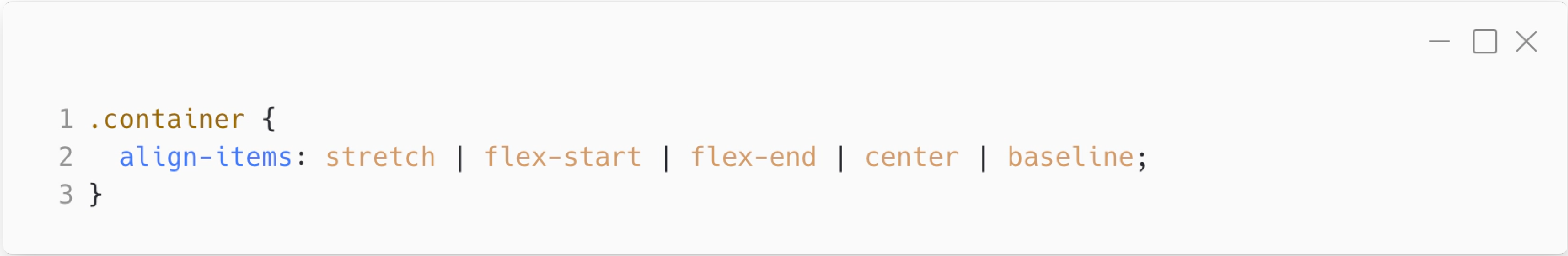


```
.container {
  justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;
}
```

Properties for the Parent (flex container)

align-items

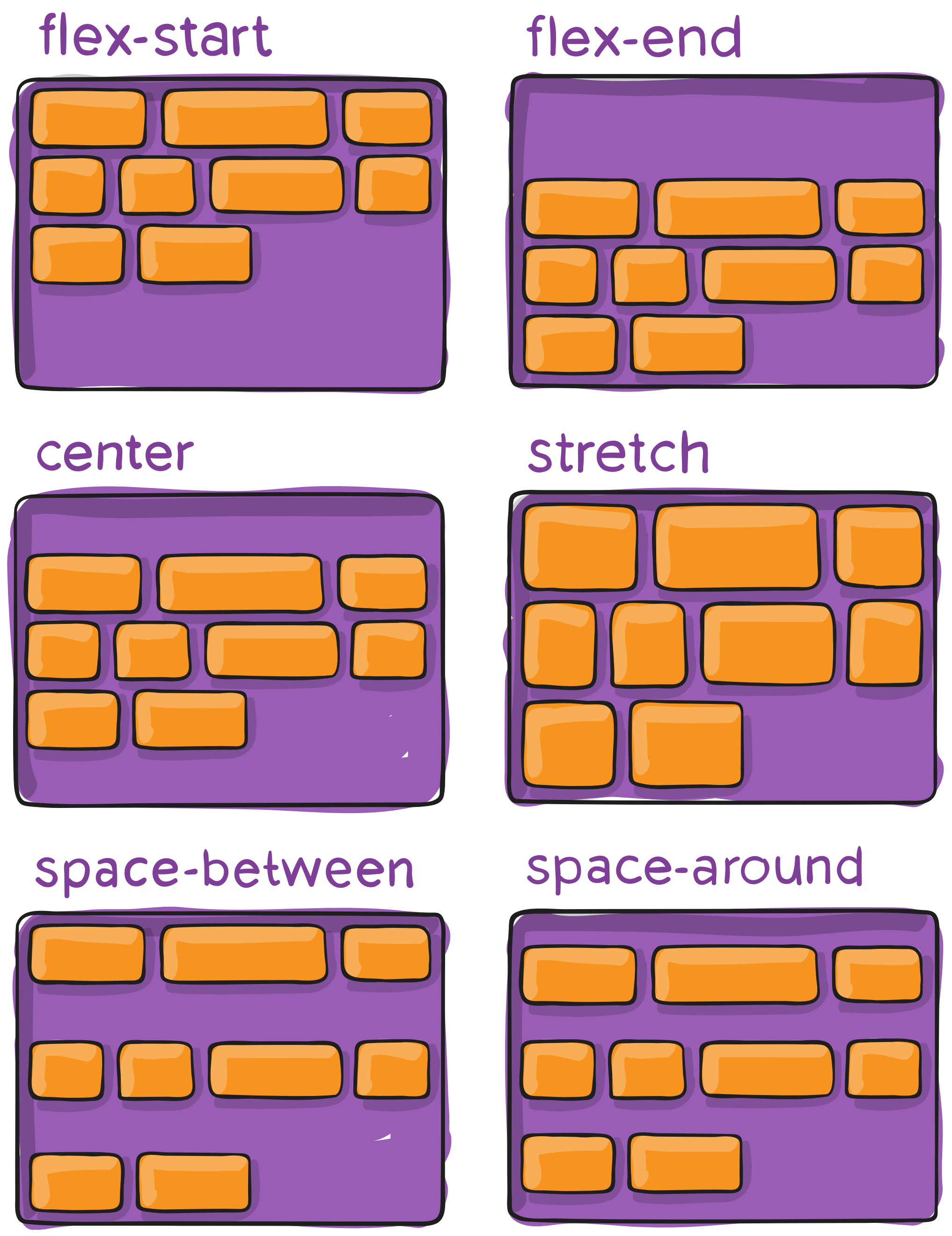




```
.container {
  align-items: stretch | flex-start | flex-end | center | baseline;
}
```

Properties for the Parent (flex container)

align-content



flex-start

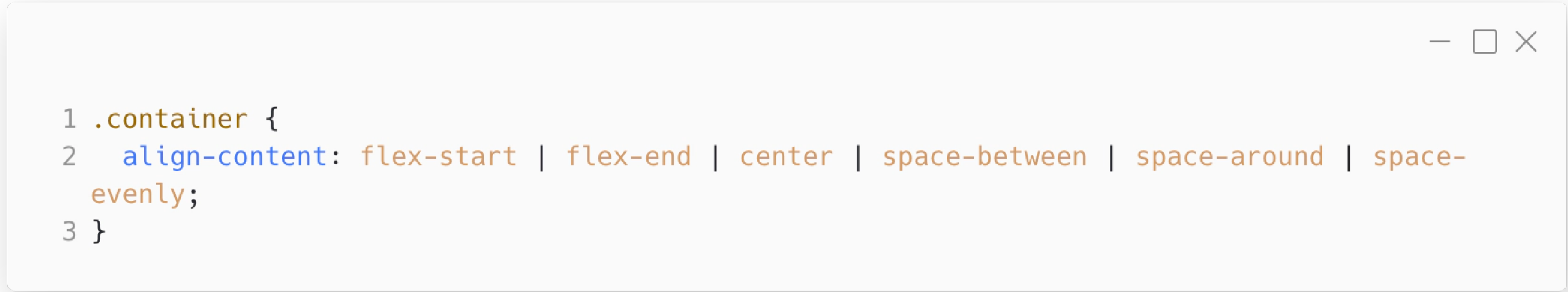
flex-end

center

stretch

space-between

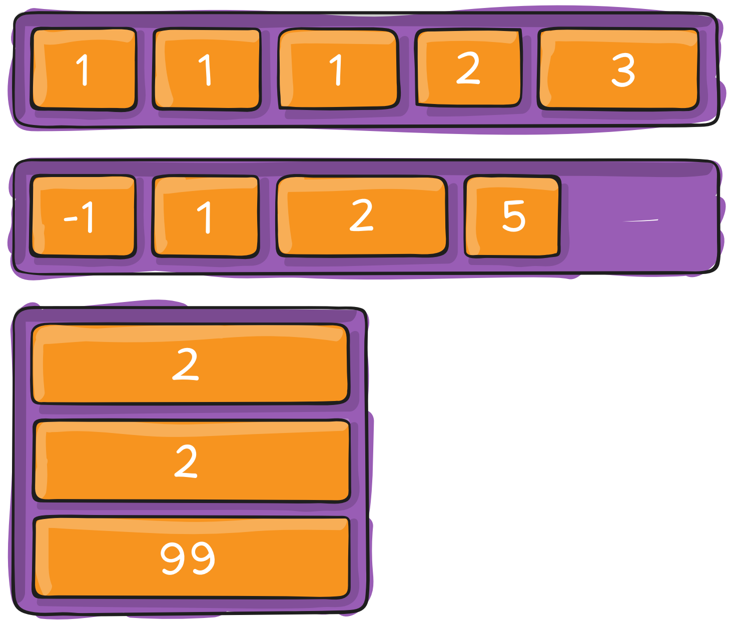
space-around

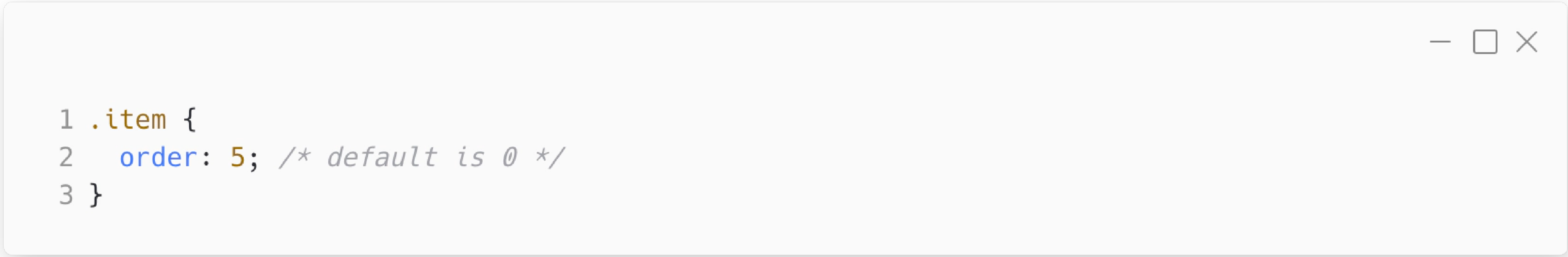


```css
.container {
  align-content: flex-start | flex-end | center | space-between | space-around | space-evenly;
}
```

Properties for the Children (flex items)

order

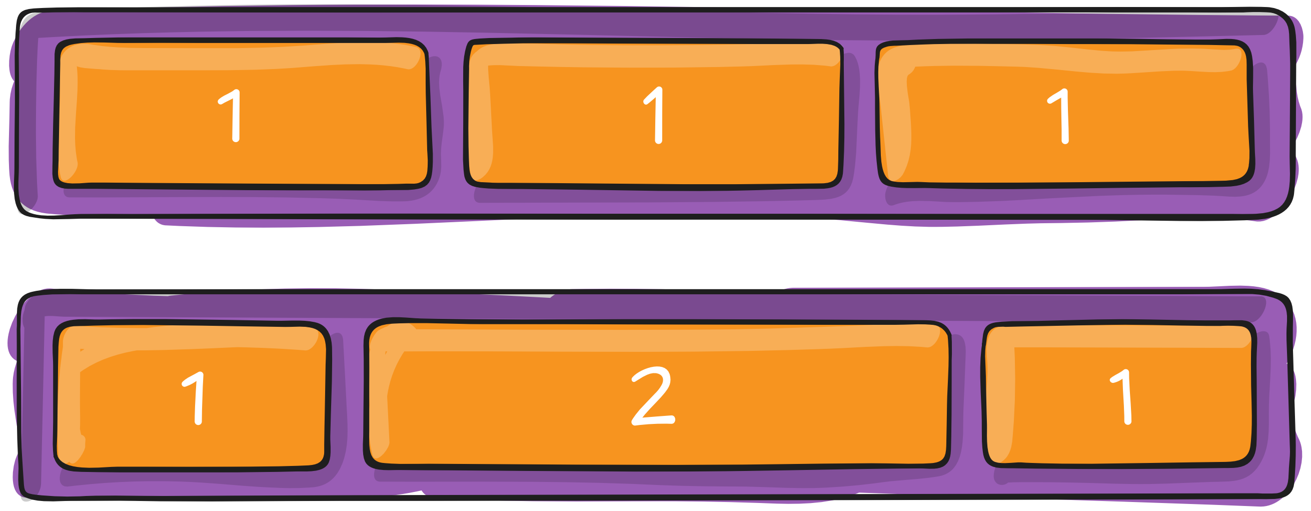


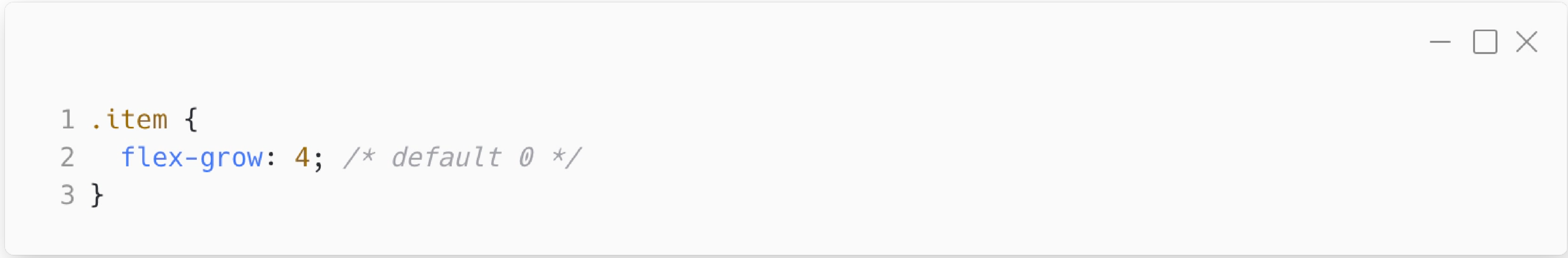


```
.item {
  order: 5; /* default is 0 */
}
```

Properties for the Children (flex items)

flex-grow

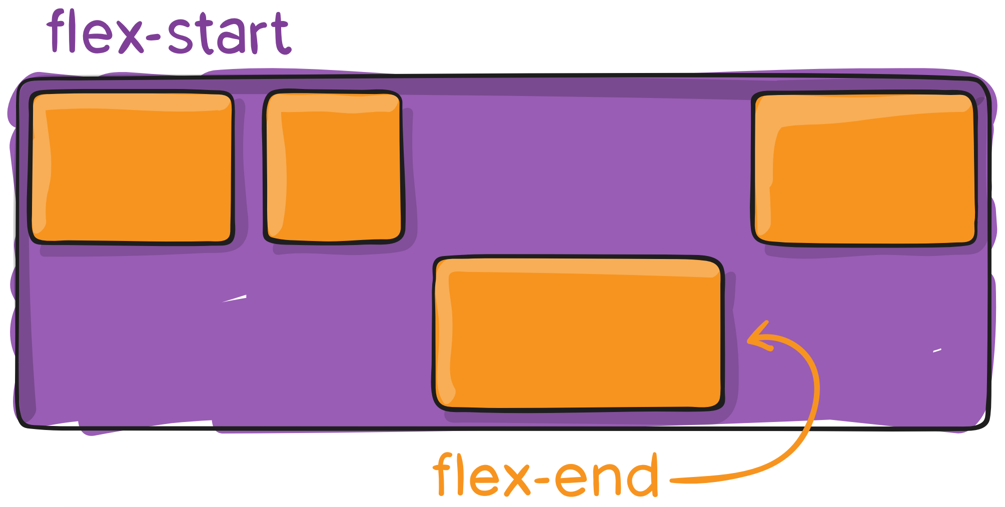


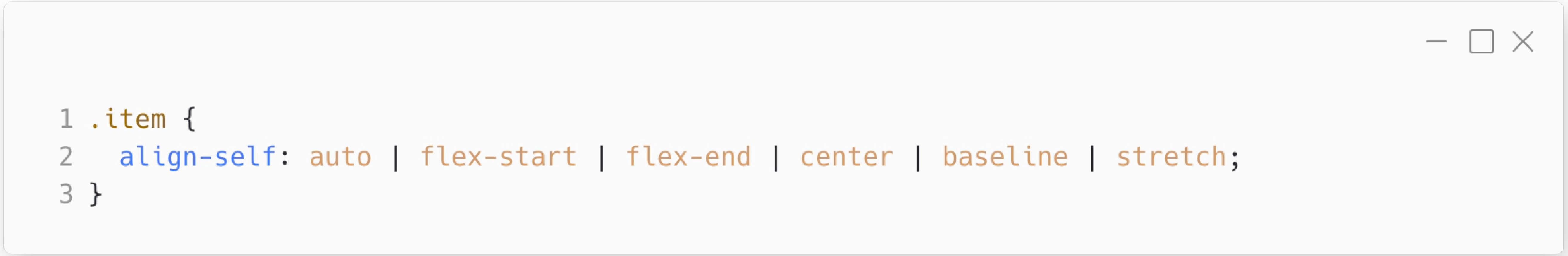


```
.item {
  flex-grow: 4; /* default 0 */
}
```

Properties for the Children (flex items)

align-self





```css
.item {
  align-self: auto | flex-start | flex-end | center | baseline | stretch;
}
```

We don't have to line everything up.

Sometimes what we need is just to make things fit nicely.

Activity

Make a responsive chessboard using flexbox!

References

- https://www.bankmycell.com/blog/how-many-phones-are-in-the-world
- https://www.d8aspring.com/eye-on-asia/is-the-social-media-capital-of-the-world-ready-for-a-life-online
- https://business.inquirer.net/293426/netflix-to-adjust-video-quality-as-covid-19-quarantine-kicks-up-internet-usage
- https://web.dev/responsive-web-design-basics/
- https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/
- https://learn.shayhowe.com/advanced-html-css/responsive-web-design/
- https://alistapart.com/article/responsive-web-design/
- https://kyleschaeffer.com/css-font-size-em-vs-px-vs-pt-vs-percent