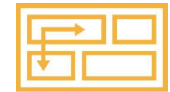
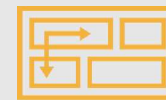


WHAT IS CSS GRID?

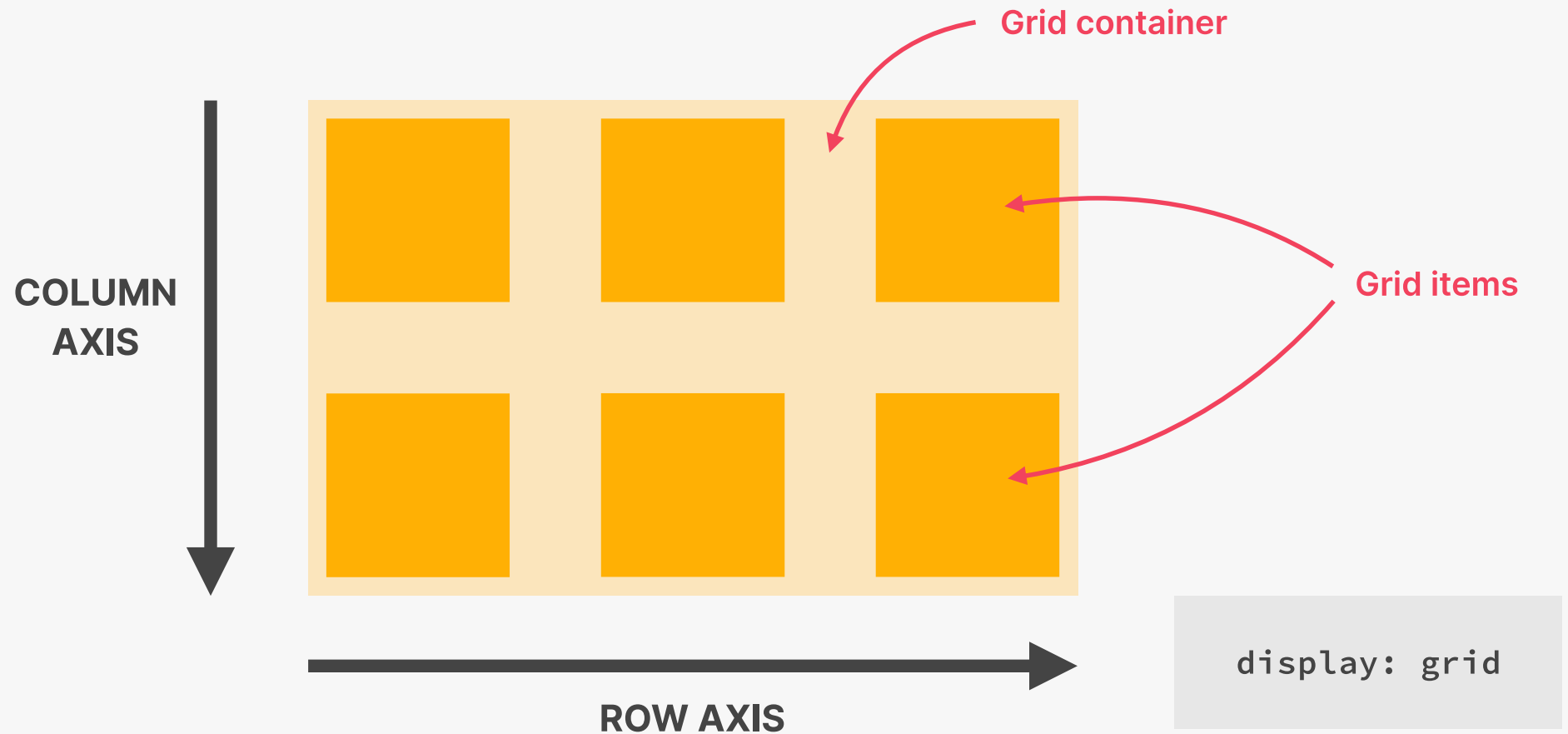
CSS GRID



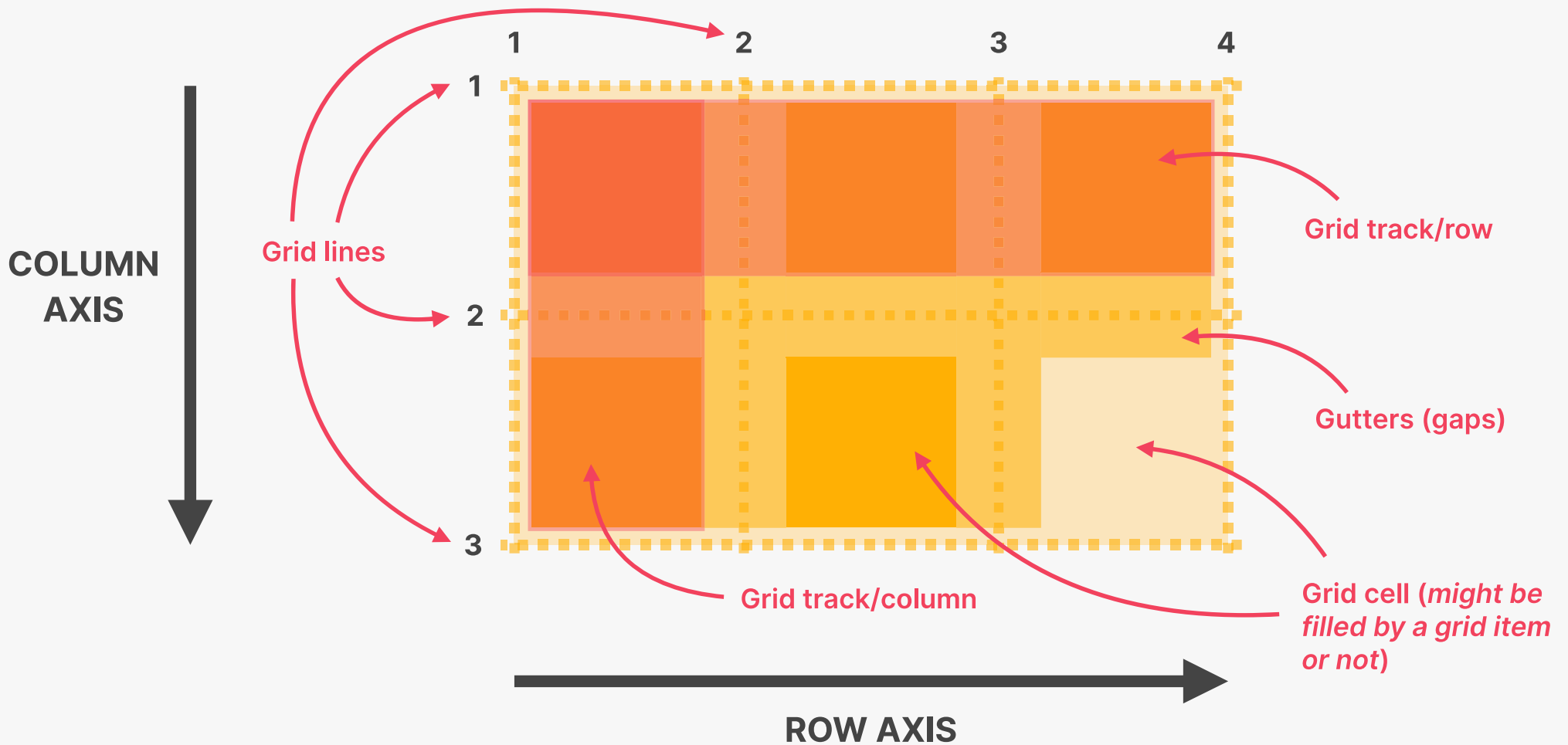
- 👉 CSS Grid is a set of **CSS properties** for **building 2-dimensional layouts**
- 👉 The main idea behind CSS Grid is that we **divide a container element into rows and columns** that can be filled with its child elements
- 👉 In two-dimensional contexts, CSS Grid allows us to write **less nested HTML** and **easier-to-read CSS**
- 👉 CSS Grid is **not meant to replace flexbox**! Instead, they work perfectly together. Need a **1D** layout? Use flexbox. Need a **2D** layout? Use CSS Grid.



BASIC CSS GRID TERMINOLOGY



MORE CSS GRID TERMINOLOGY



GRID CONTAINER

1

`grid-template-rows: <track size>*`
`grid-template-columns: <track size>*`

👉 To establish the grid **row and column tracks**. One length unit for each track. Any unit can be used, new **fr** fills unused space

2

`row-gap: 0 | <length>`
`column-gap: 0 | <length>`] `gap: 0 | <length>`

👉 To **create empty space** between tracks

3

`justify-items: stretch | start | center | end`
`align-items: stretch | start | center | end`

👉 To align items inside rows / columns (**horizontally / vertically**)

4

`justify-content: start | start | center | end | ...`
`align-content: start | start | center | end | ...`

👉 To align entire **grid inside grid container**. Only applies if container is larger than the grid

GRID ITEMS

1

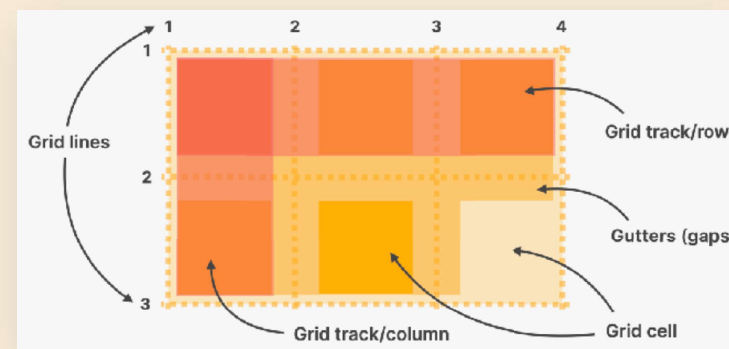
`grid-column: <start line> / <end line> | span <number>`
`grid-row: <start line> / <end line> | span <number>`

👉 To **place a grid item** into a specific cell, based on line numbers. `span` keyword can be used to span an item across more cells

2

`justify-self: stretch | start | center | end`
`align-self: stretch | start | center | end`

👉 To **overwrite** `justify-items` / `align-items` for single items



👉 This list of CSS Grid properties is not exhaustive, but enough to get started.

