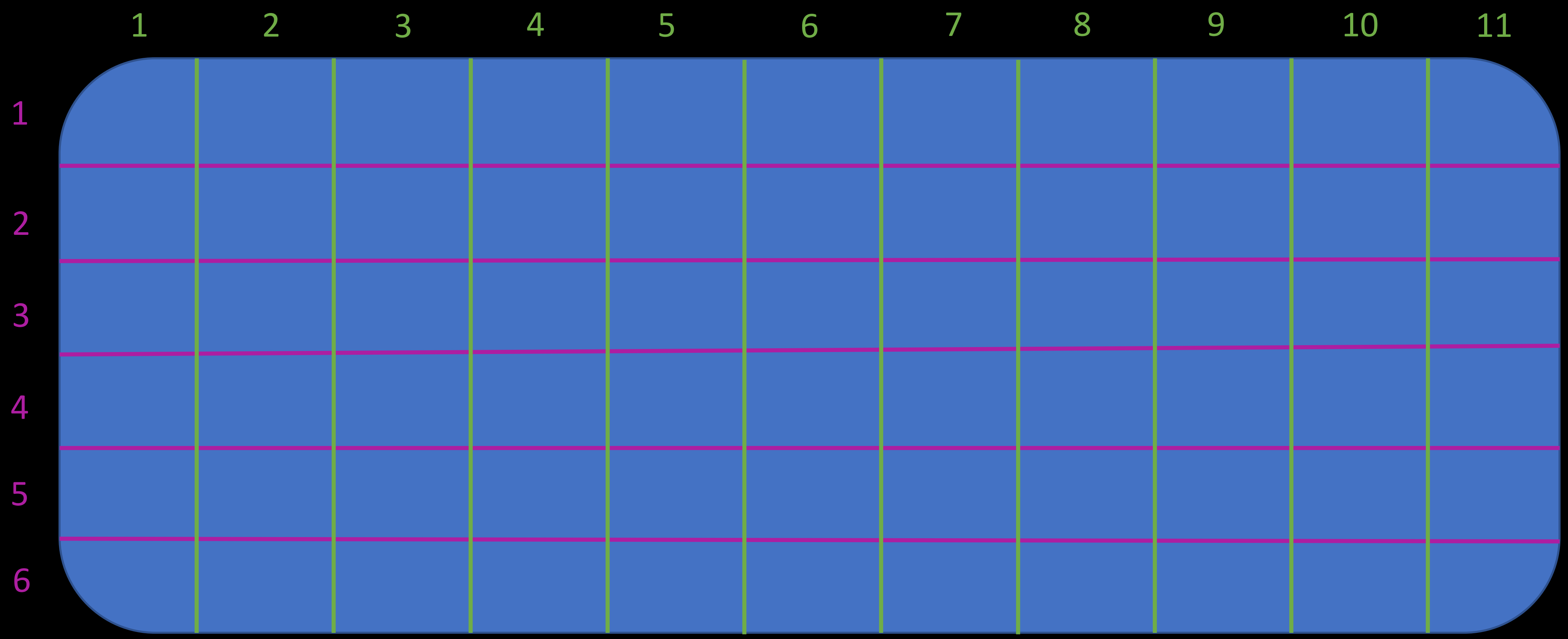
CSS Grid

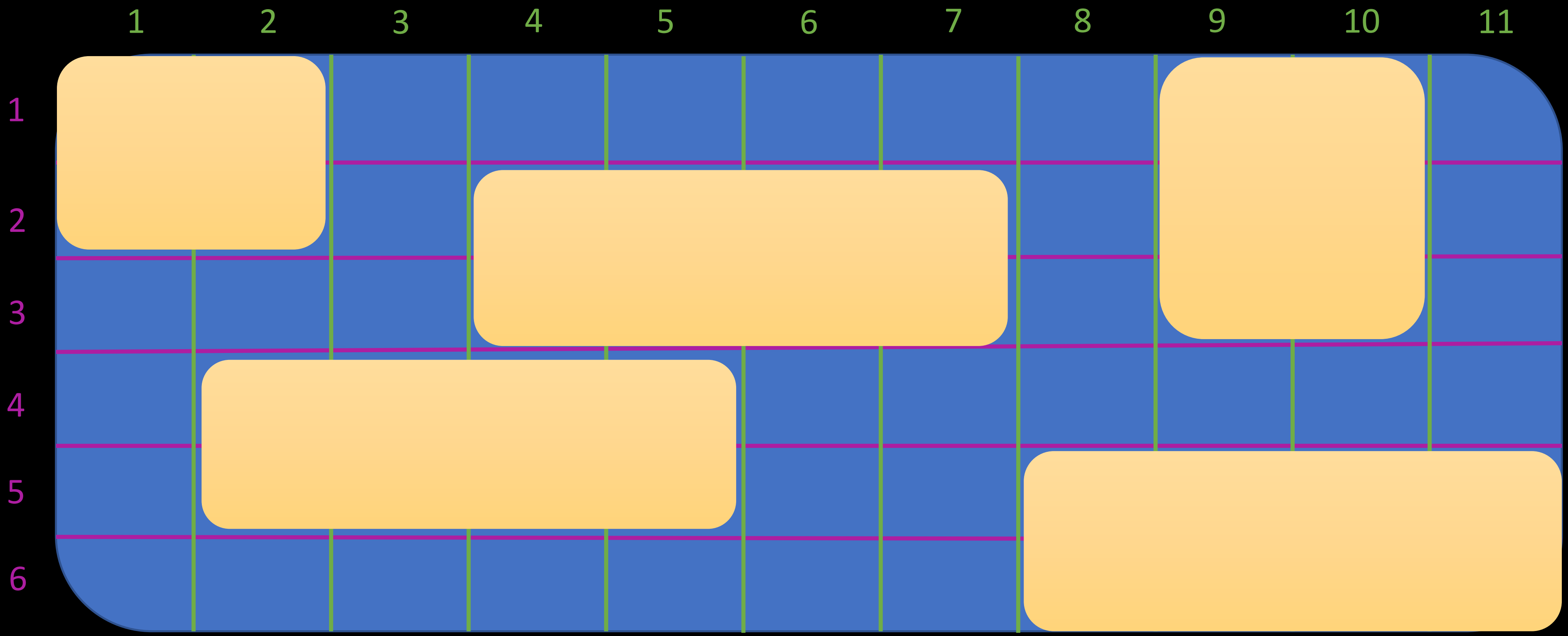
A system of laying out items across a set of vertical and horizontal lines

- You can turn an element into a grid container using **display: grid;**
- Now we can slice and dice our grid and place the items anywhere on it

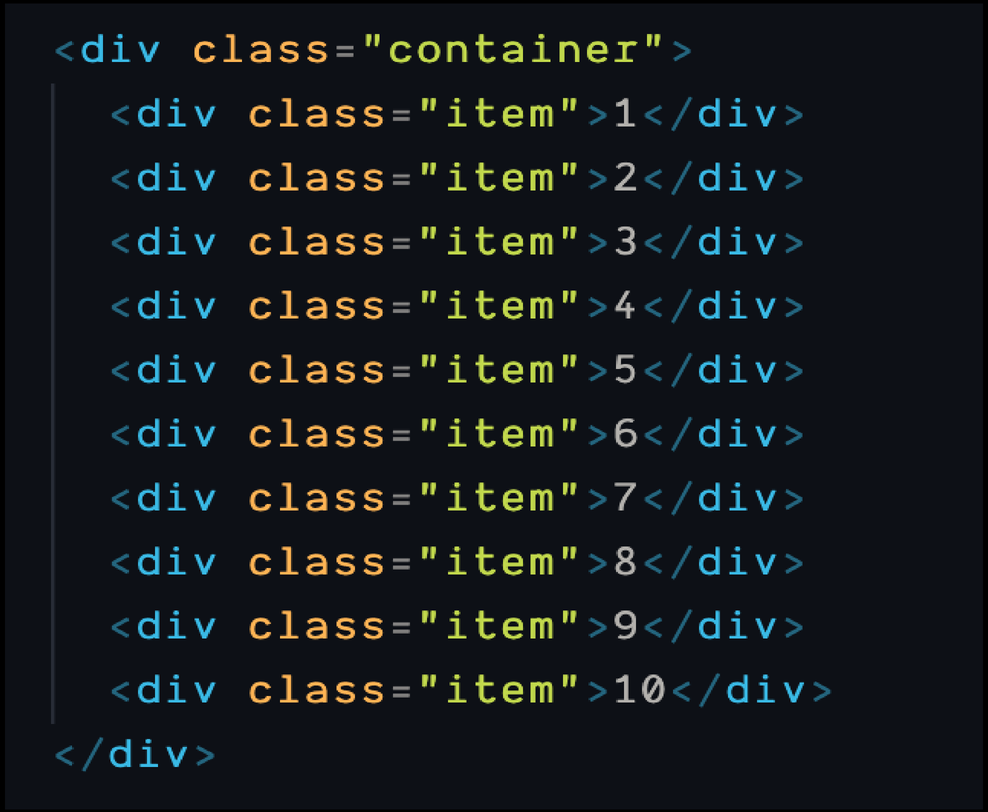


A system of laying out items across a set of vertical and horizontal lines

- You can turn an element into a grid container using **display: grid;**
- Now we can slice and dice our grid and place the items anywhere on it

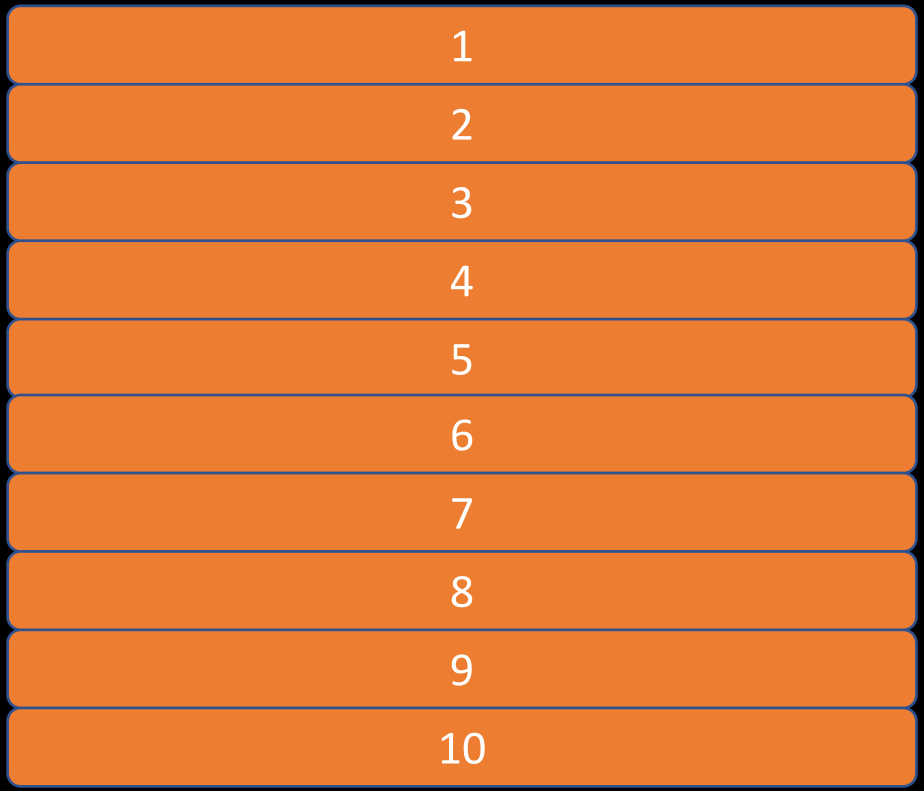


Given the following example



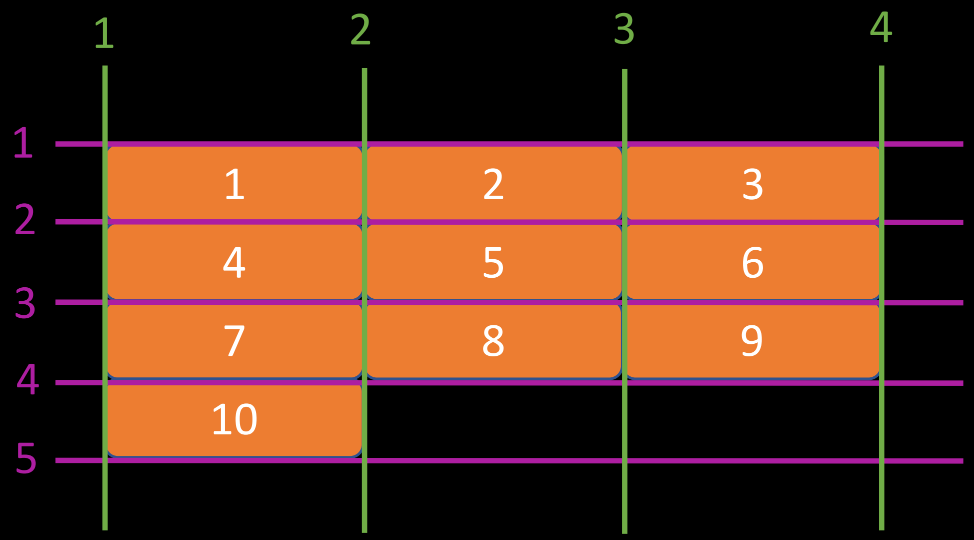
```html
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
  <div class="item">6</div>
  <div class="item">7</div>
  <div class="item">8</div>
  <div class="item">9</div>
  <div class="item">10</div>
</div>
```

Add **display: grid;**



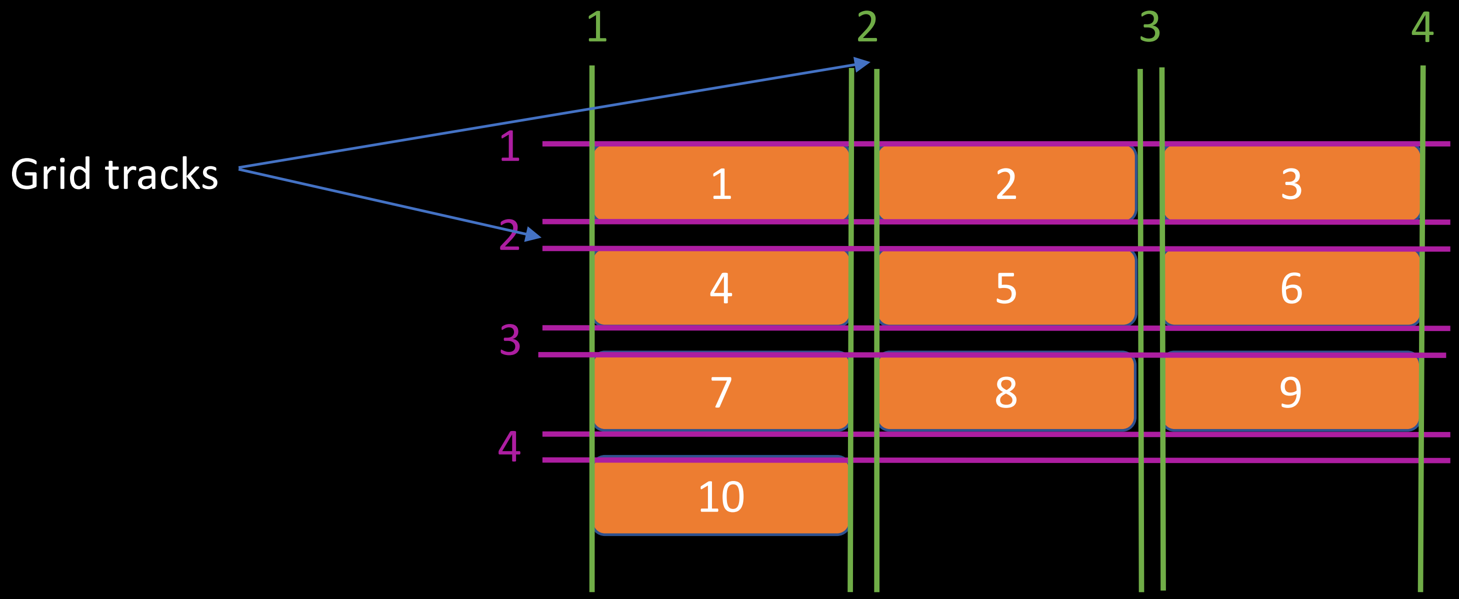
Explicitly defining the **grid-template-columns**

- Specify **grid-template-columns: 100px 100px 100px** on the grid container (explicit grid)
- The rows will be automatically adjusted based on the content (implicit grid)



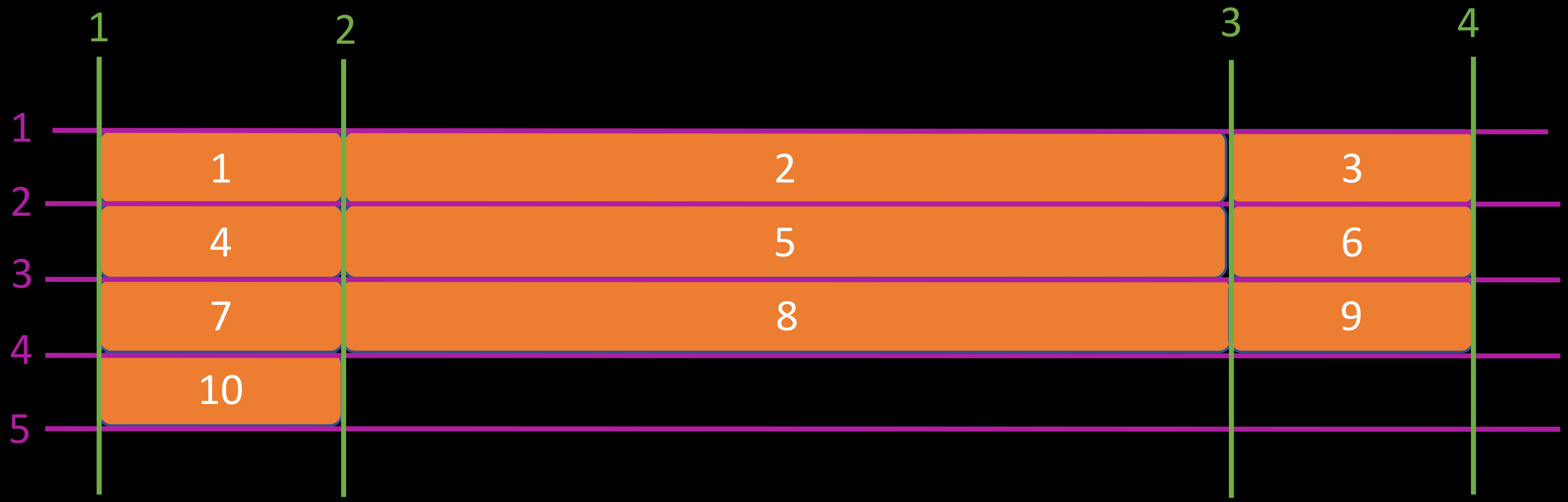
Let's add some spacing between them

- Specify **grid-gap: 20px** on the grid container



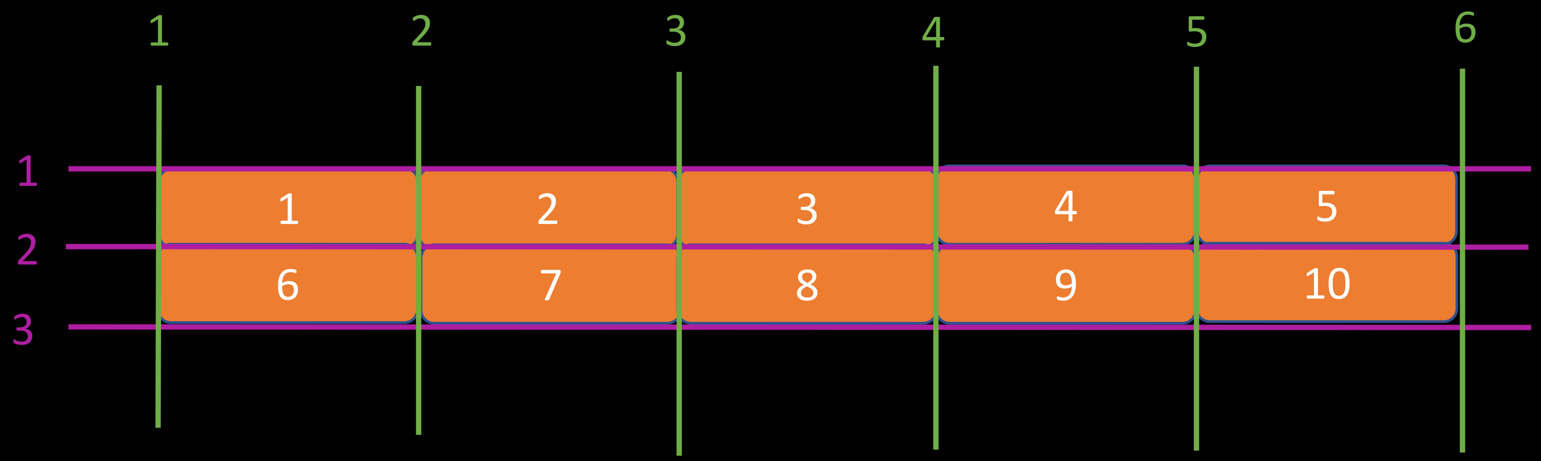
Playing with the **grid-template-columns**

- Specify **grid-template-columns: 100px auto 100px** on the grid container
- auto will stretch the middle column to the whole available space



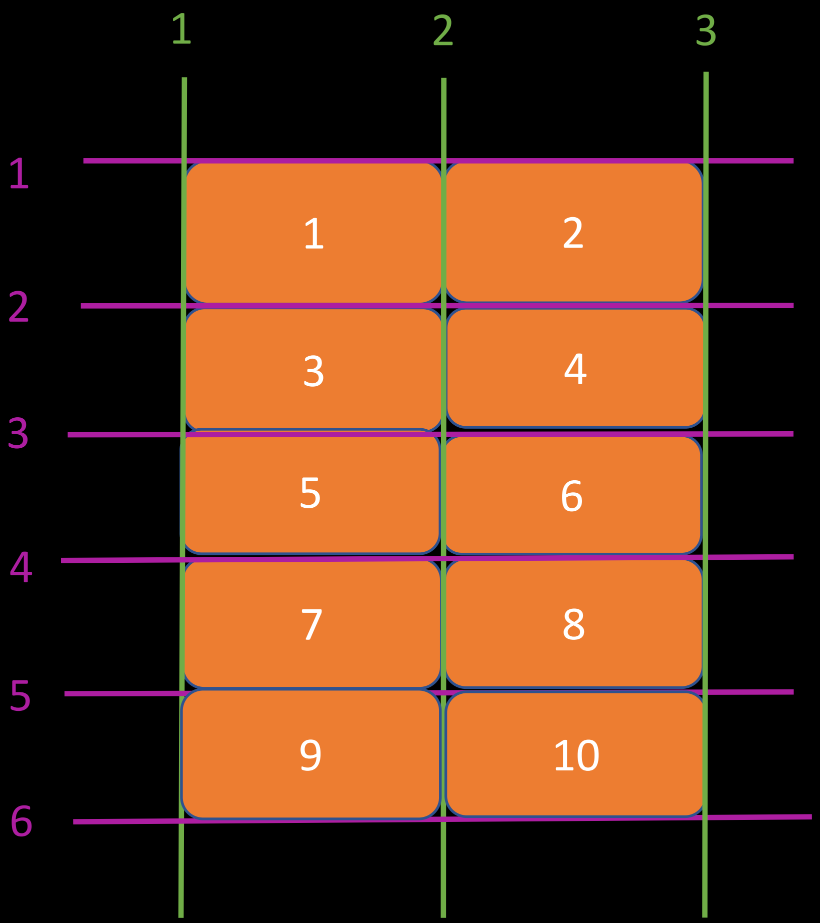
Playing with the **grid-template-columns**

- Specify **grid-template-columns: repeat(5, 100px)** on the grid container



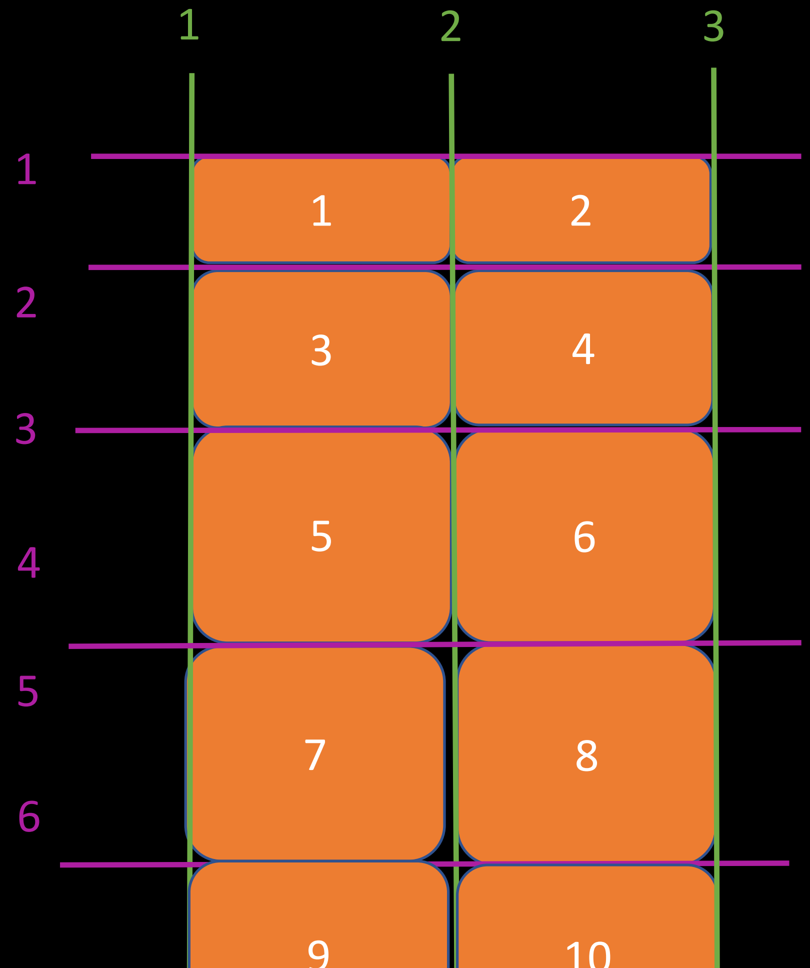
Playing with the **grid-template-rows**

- Specify **grid-template-rows: repeat(5, 50px)** on the grid container (explicit)



Styling implicit grid

- Specify **grid-template-rows: 40px 100px** on the grid container (explicit)
- Specify **grid-template-columns: 100px 100px** (explicit);
- Specify **grid-auto-rows: 150px;** (will affect implicit rows)

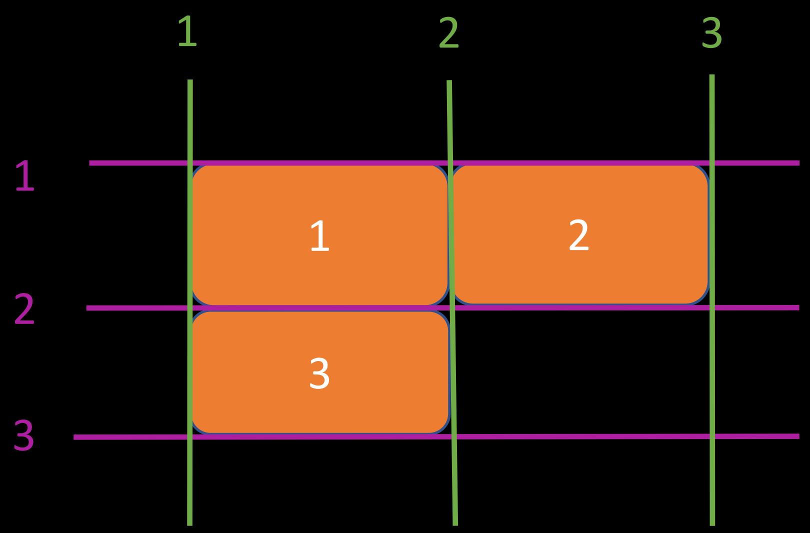


- When we define a certain number of columns, the extra elements will be converted into a new row

- For implicit columns we can use **grid-auto-flow**

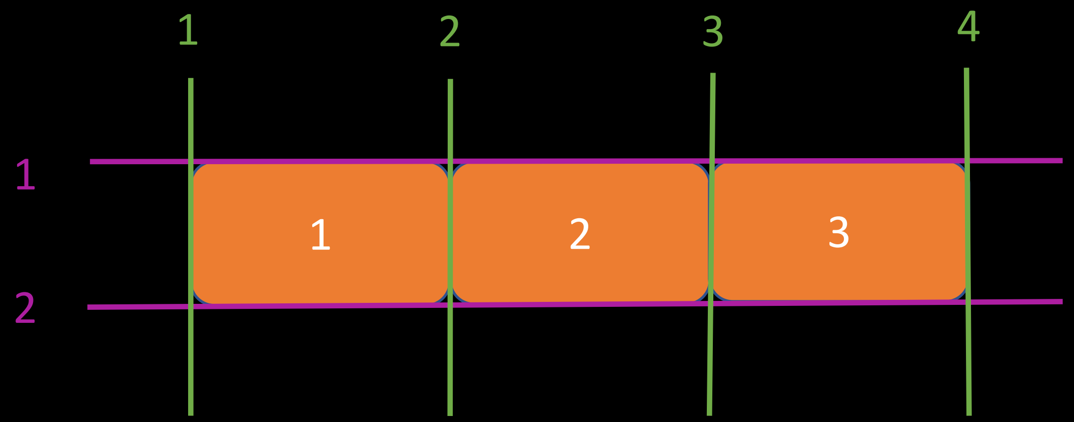
Playing with the grid-auto-flow

- Specify **grid-template-columns: 100px 100px;** on the grid container (explicit)
- 3rd element will be moved into a new implicit row



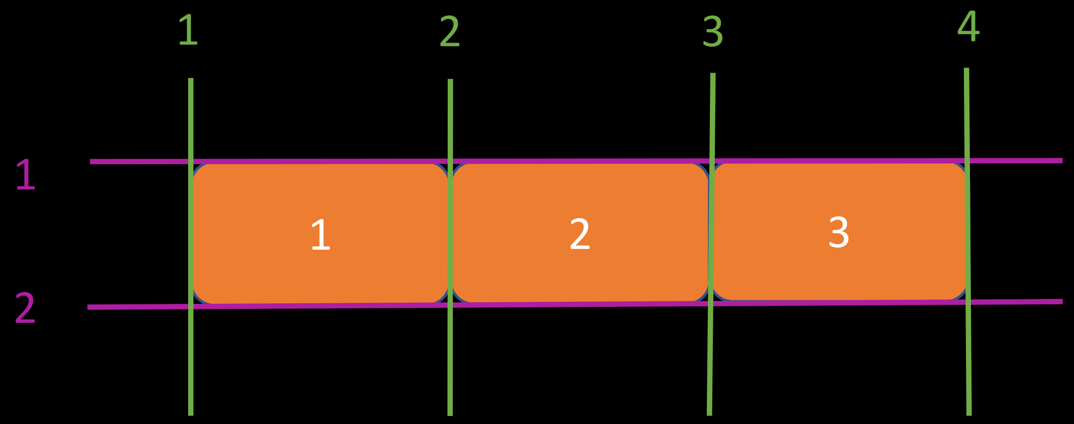
Playing with the grid-auto-flow

- Specify **grid-template-columns: 100px 100px;** on the grid container (explicit)
- Set **grid-auto-flow: column;** (defaults to row) // similar to flex-direction



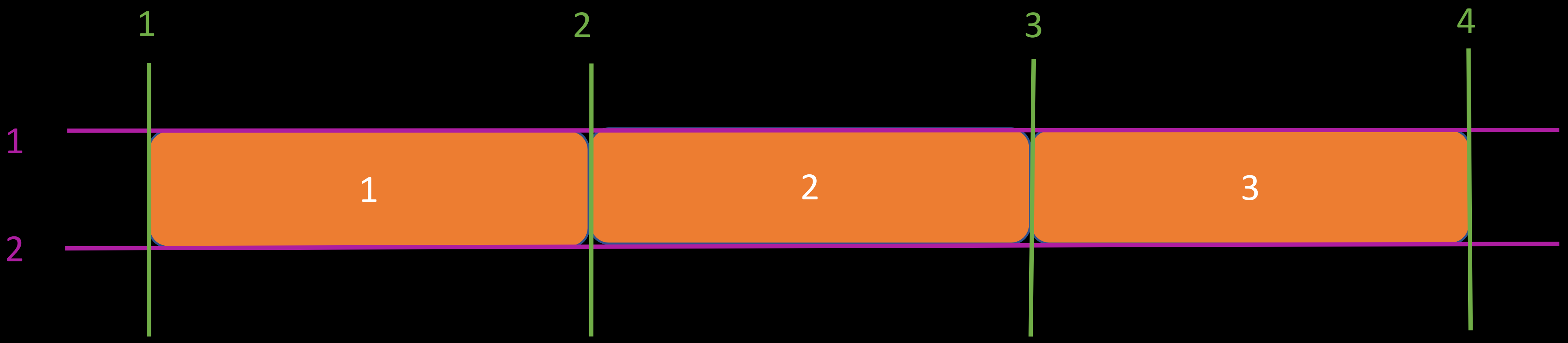
Playing with the grid-auto-flow

- Specify **grid-template-columns: 100px 100px;** on the grid container (explicit)
- Set **grid-auto-flow: column;** (defaults to row) // similar to flex-direction



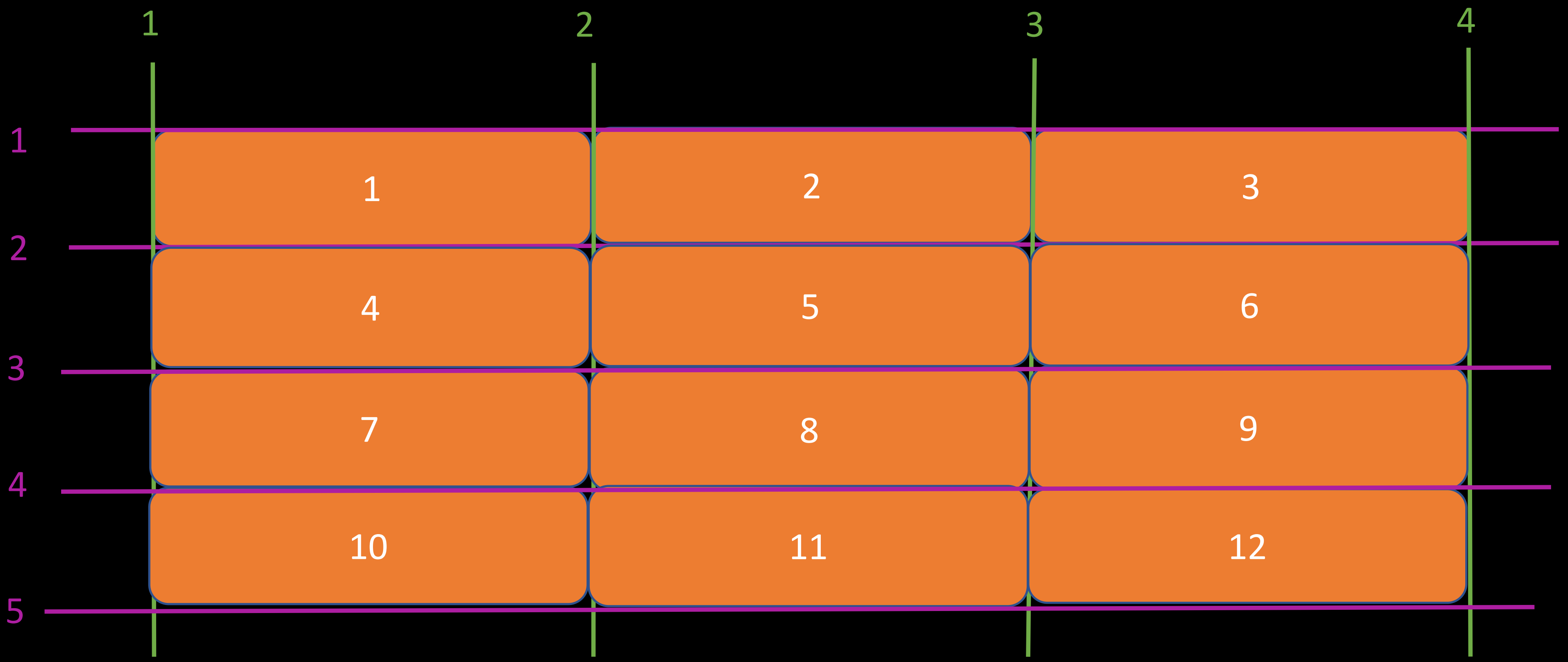
Playing with **fr** units

- Specify **grid-template-columns: 1fr 1fr 1fr // repeat(3, 1fr);** the grid items will take up their space
- + 1 fraction of the available free space



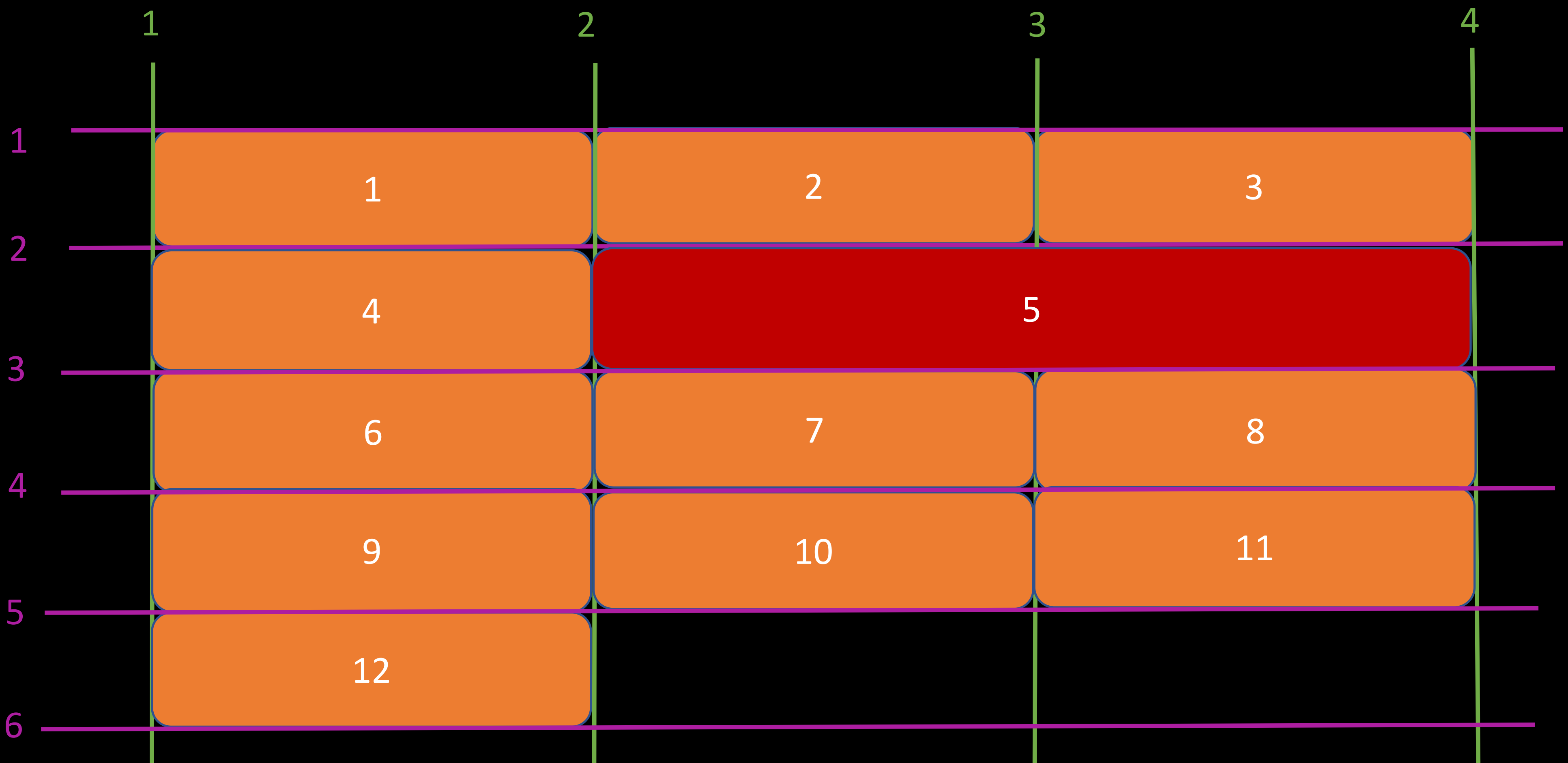
Sizing items

- Set **grid-template-columns: repeat(3, 1fr)**



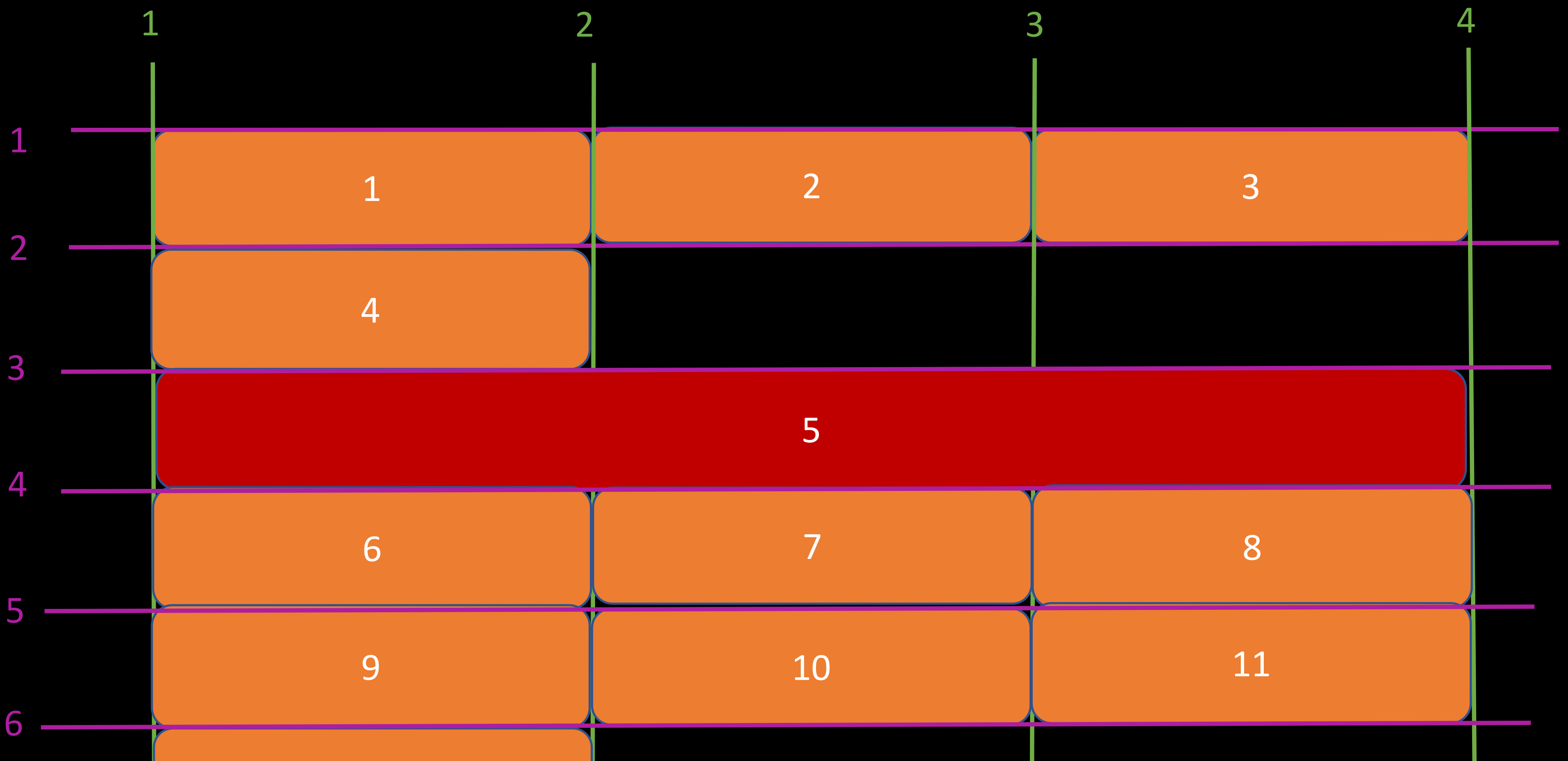
Sizing items

- Set **grid-column: span 2;**



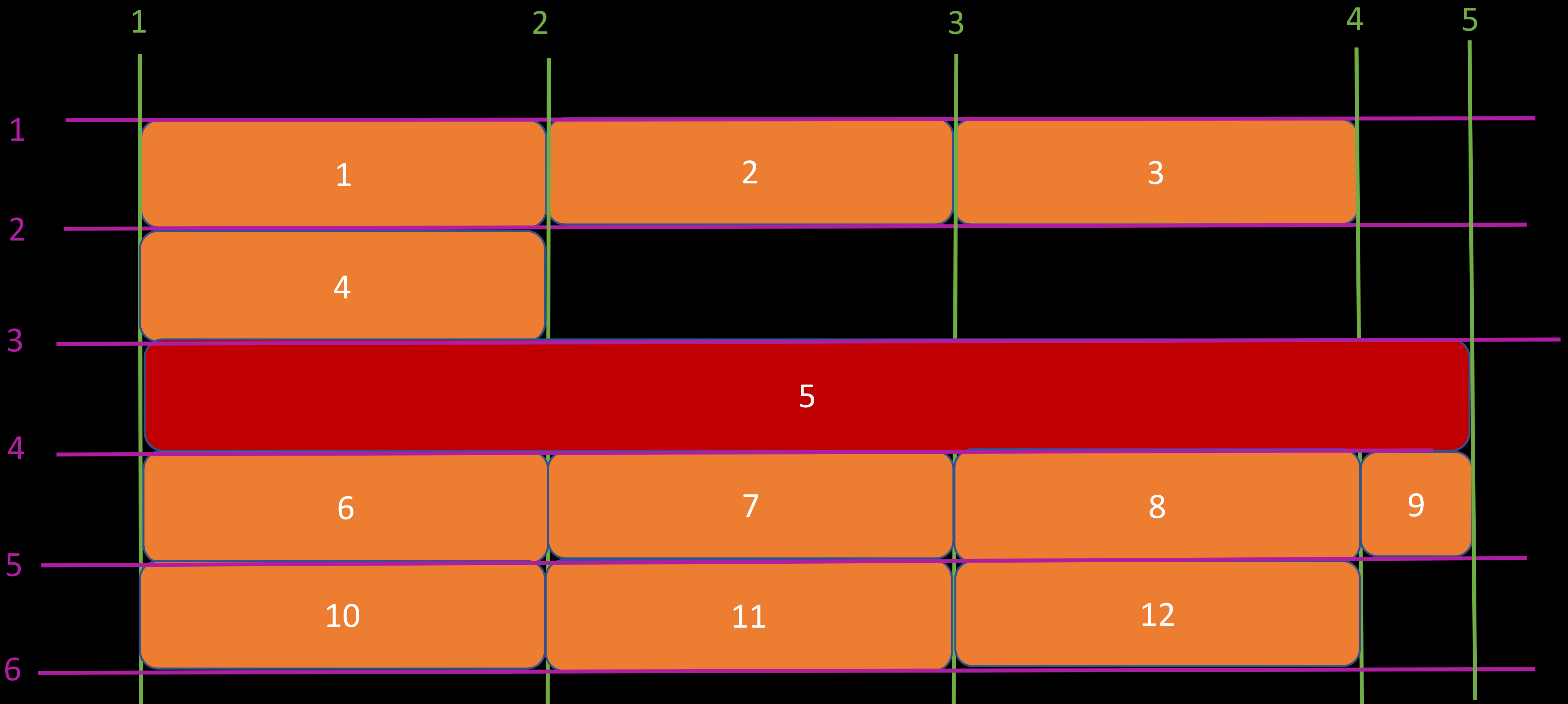
Sizing items

- Set **grid-column: span 3;**



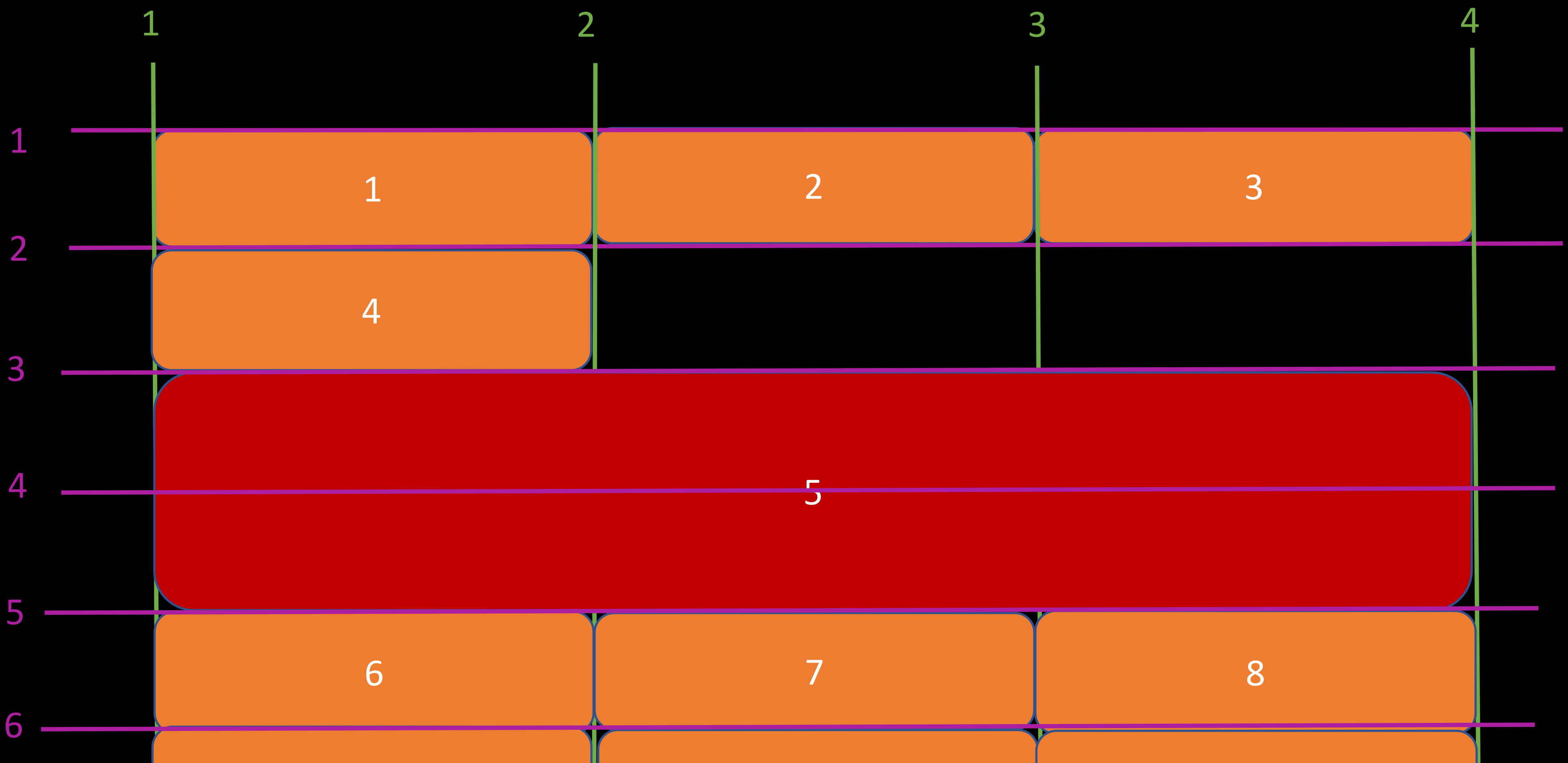
Sizing items

- Set **grid-column: span 4;** (forces the grid to get larger – creates implicit column)



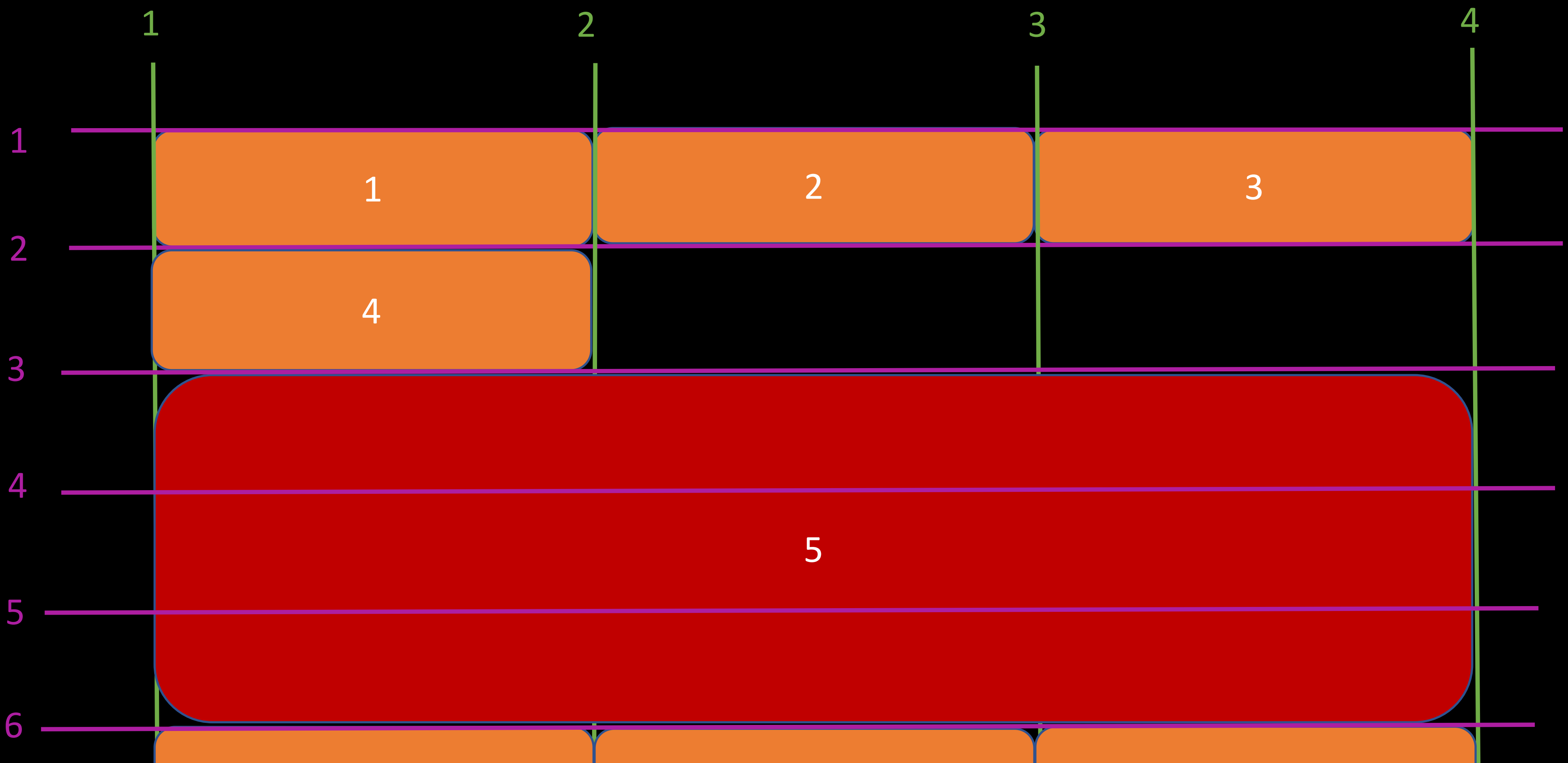
Placing grid items

- ~~Set **grid-column: span 2;**~~ // shorthand for: grid column start, grid column end;
- Set **grid-column-start: 1; grid-column-end: 4;** shorthand: **grid-column: 1 / -1**
- Set **grid-row: span 2;**



Placing grid items

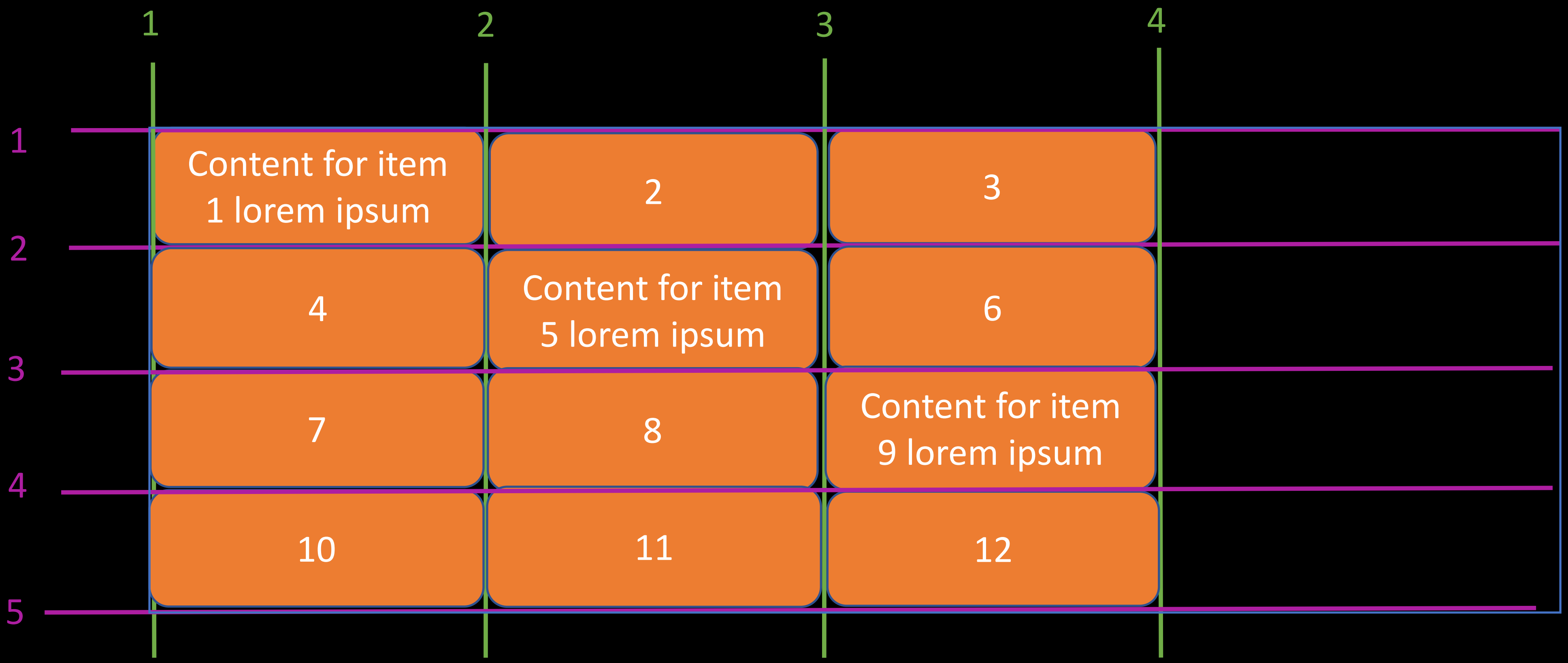
- Set **grid-column: 1 / -1**
- Set **grid-row: 3 / span 3;**



Exercise

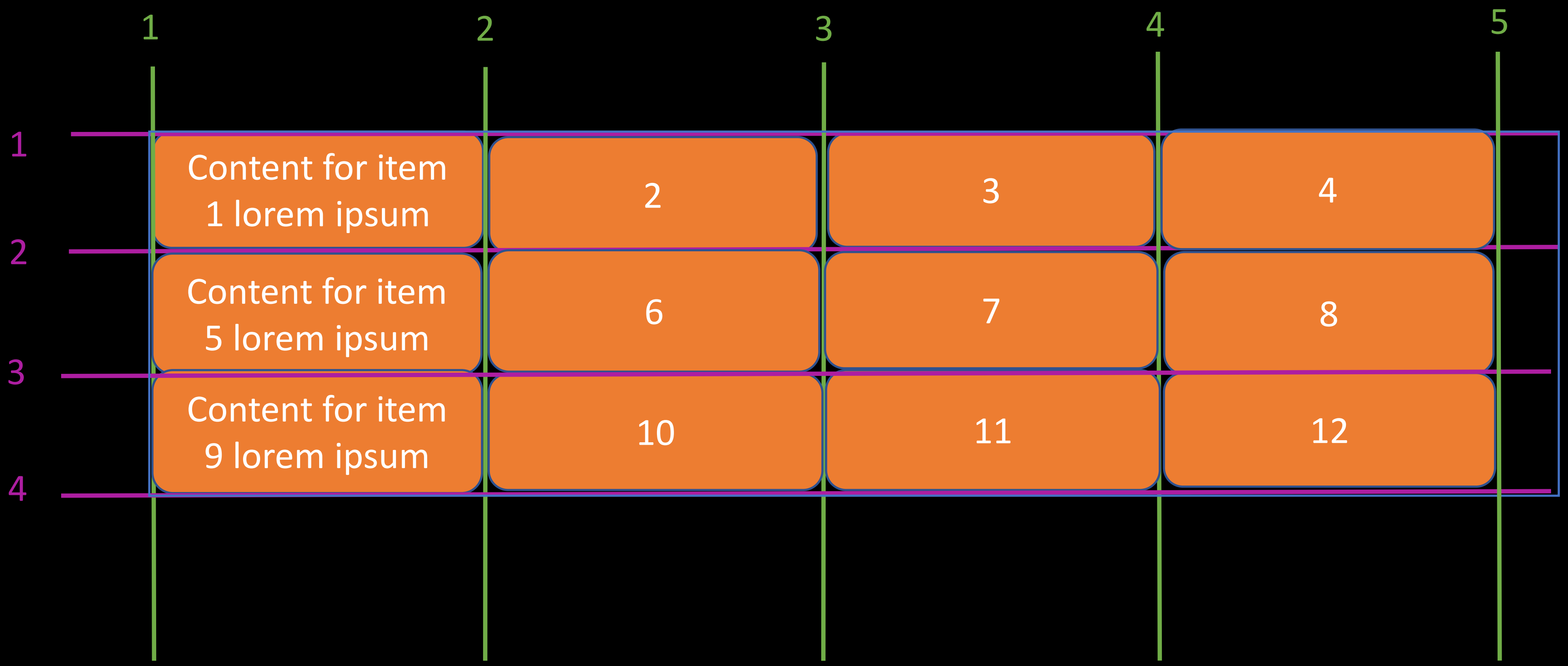
Auto fit and auto fill

- The items content might vary so we need a dynamic way to scale this
- Set **grid-template-columns: repeat(3, 300px)**


1
2
3
4
1
2
3
4
5
Content for item 1 lorem ipsum
2
3
4
Content for item 5 lorem ipsum
6
7
8
Content for item 9 lorem ipsum
10
11
12

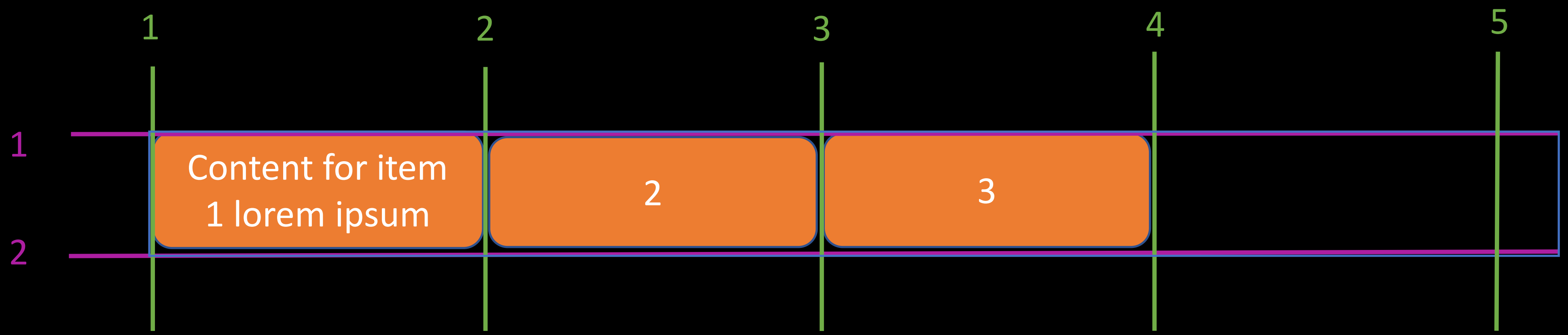
Auto fit and auto fill

- Set **grid-template-columns: repeat(auto-fill, 300px)**
- Automatically fits columns based on the container width


1
2
3
4
5
1
2
3
4
Content for item 1 lorem ipsum
2
3
4
Content for item 5 lorem ipsum
6
7
8
Content for item 9 lorem ipsum
10
11
12

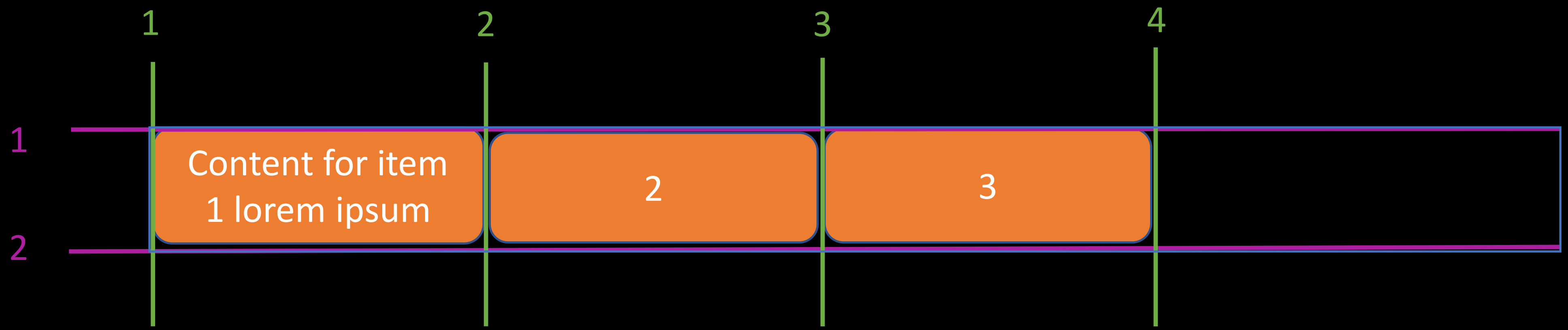
Auto fit and auto fill

- Set **grid-template-columns: repeat(auto-fill, 300px)**
- Use case: you want to move item 3 in the 4 / 5



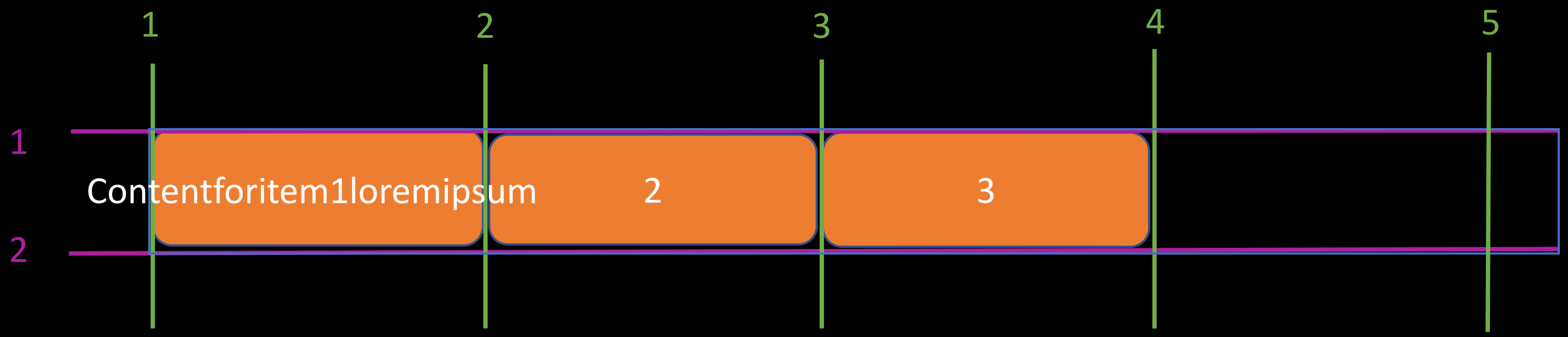
Auto fit and auto fit

- Set **grid-template-columns: repeat(auto-fit, 300px)**
- Won’t create that extra column space



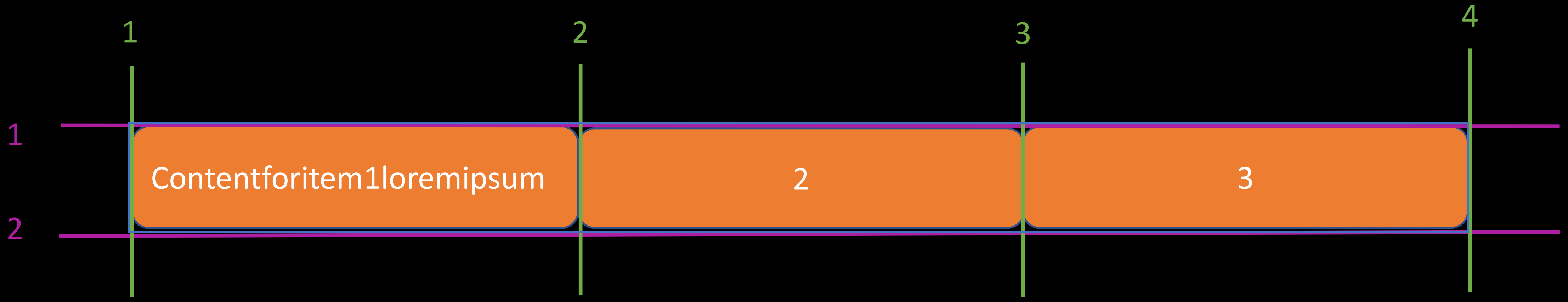
Using minmax

- Set **grid-template-columns: repeat(auto-fit, 300px)**

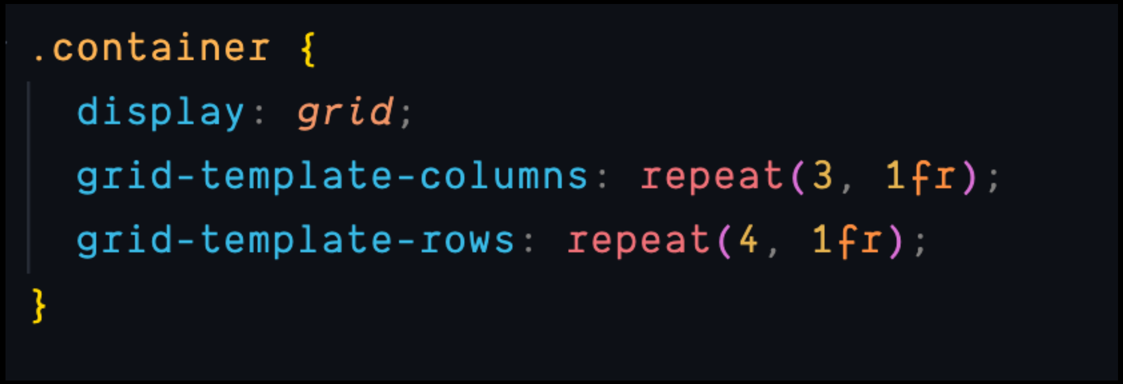


Using minmax

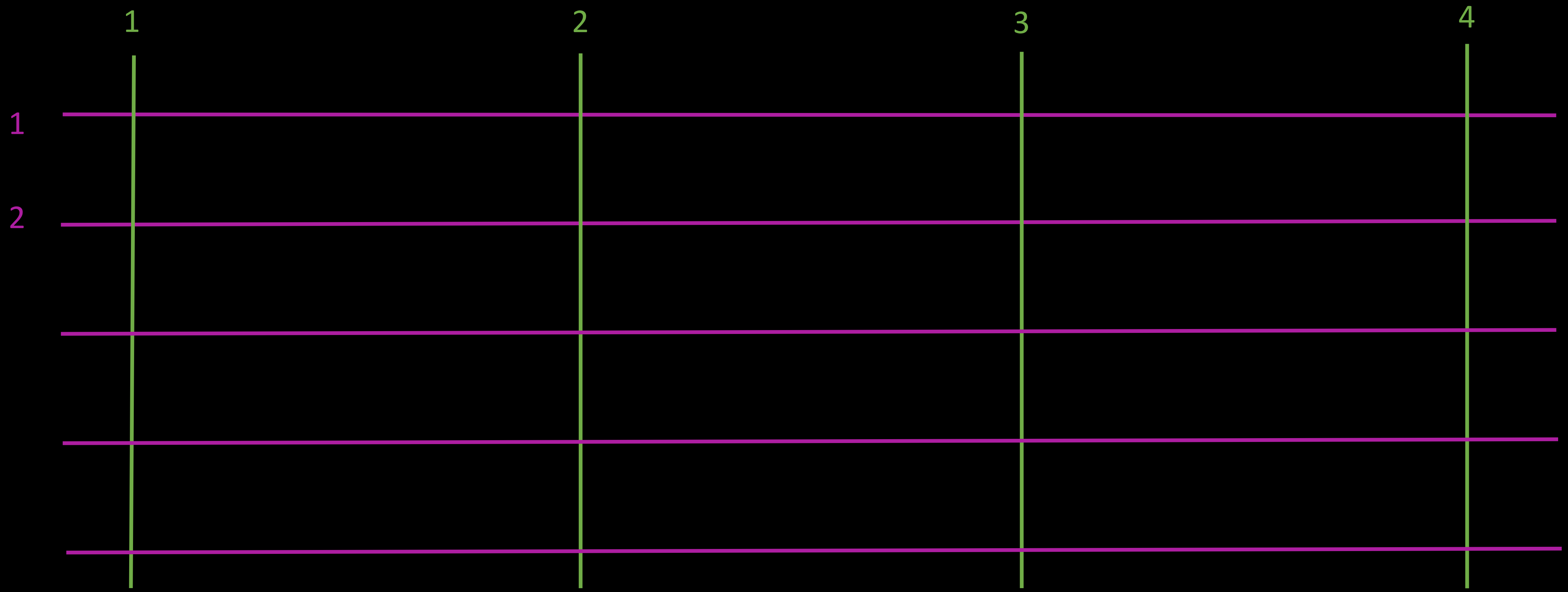
- Set **grid-template-columns: repeat(auto-fit, minmax(150px, 1fr))**



Grid template areas

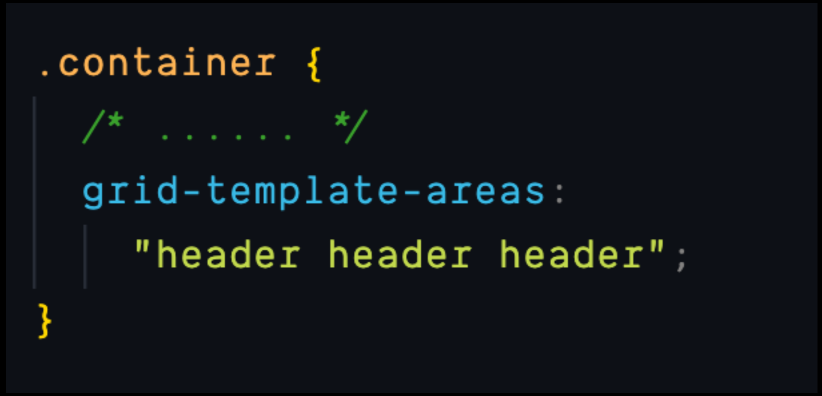


```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(4, 1fr);
}
```

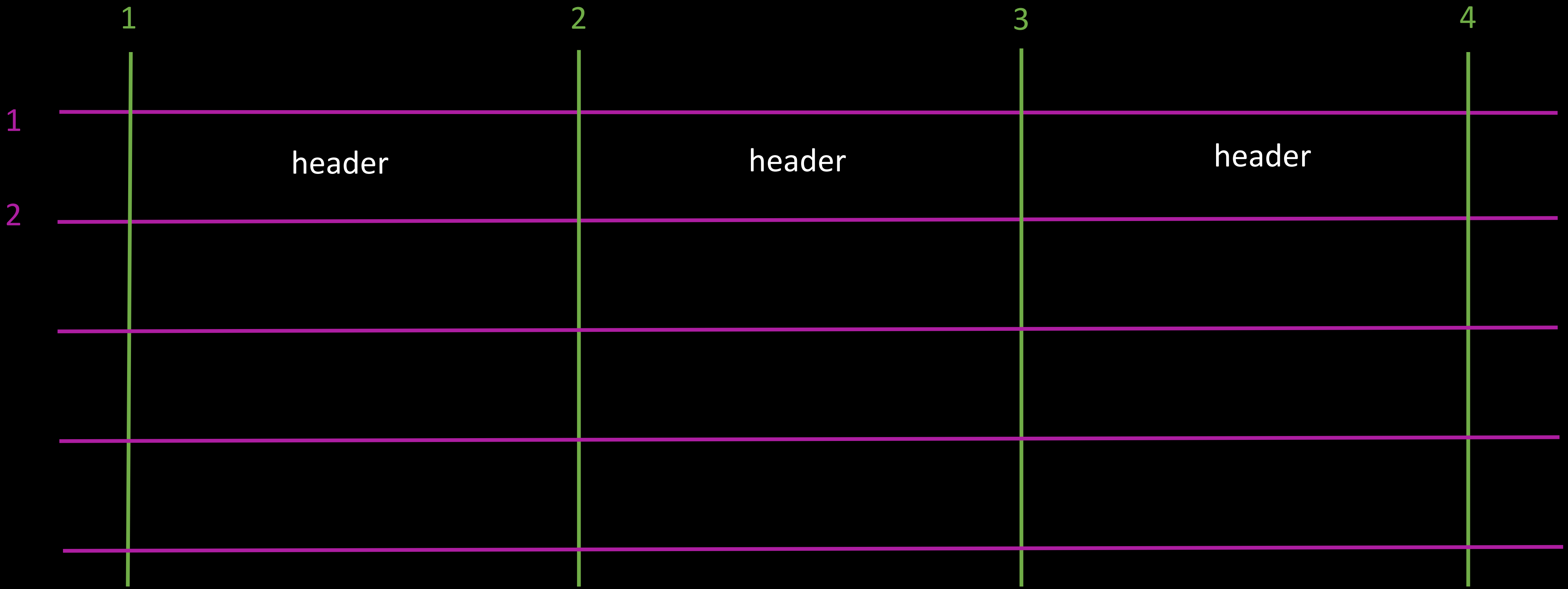


Grid template areas

- Add header area using grid-template-areas

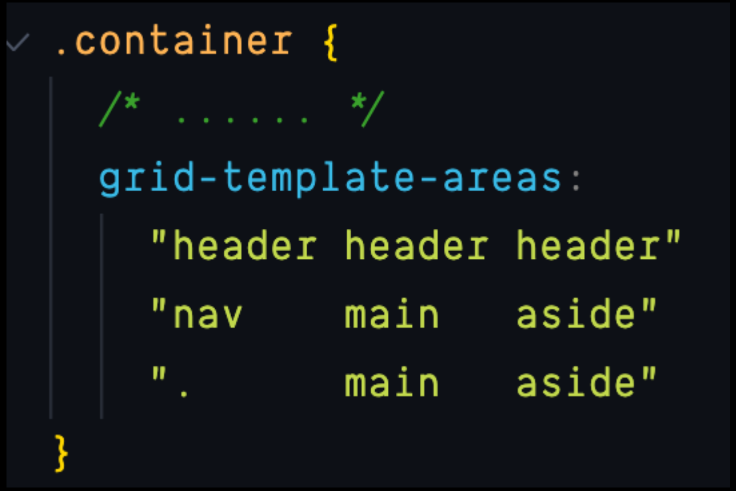


```
.container {
  /* ...... */
  grid-template-areas:
    "header header header";
}
```

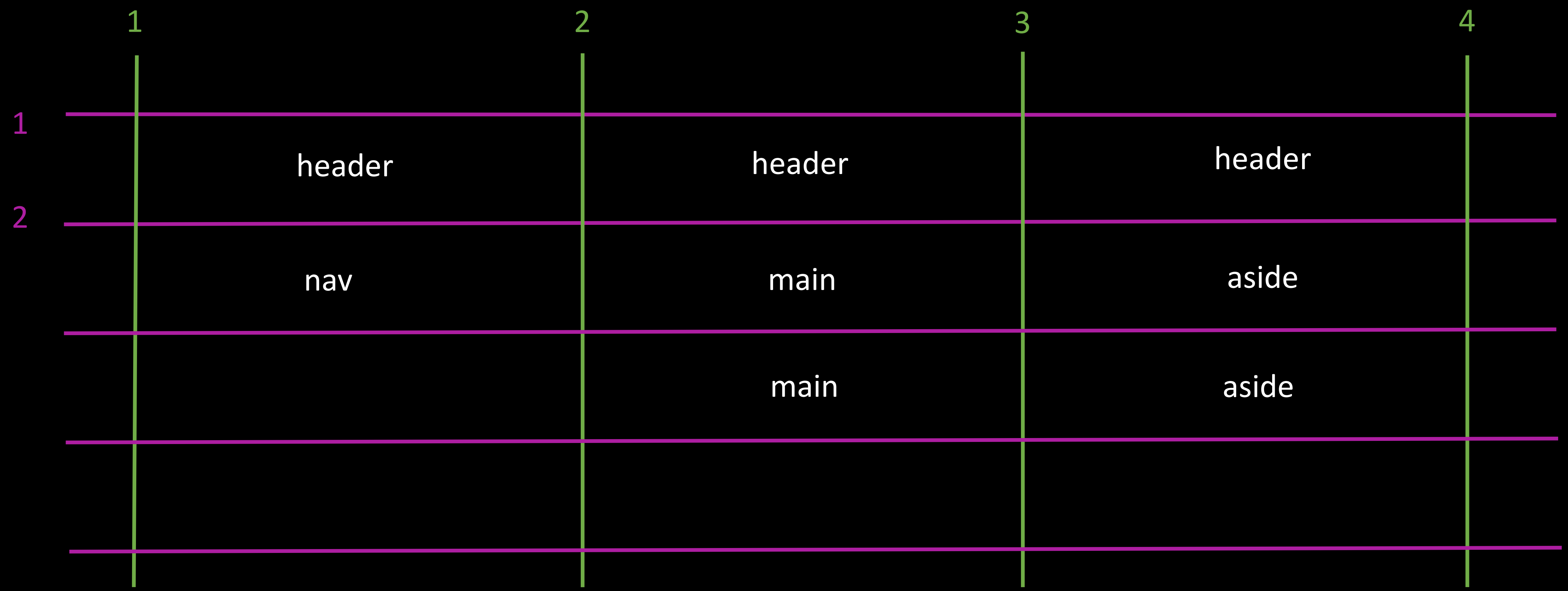


Grid template areas

- Add nav, main and side area using grid-template-areas

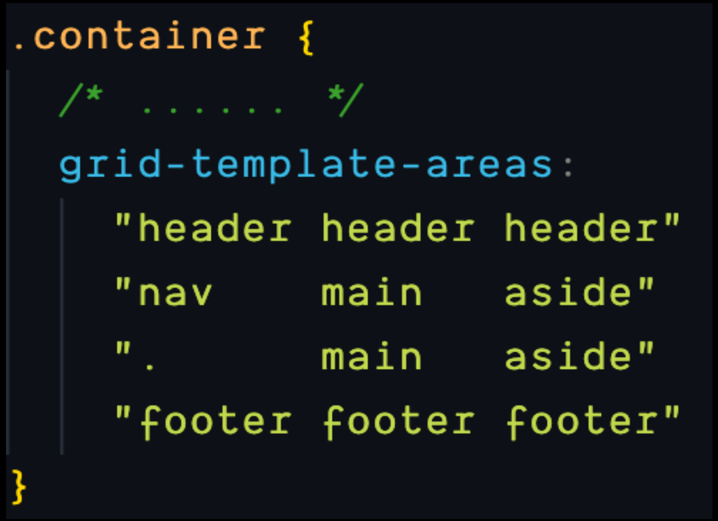


```
.container {
  /* ...... */
  grid-template-areas:
    "header header header"
    "nav    main   aside"
    ".      main   aside"
}
```

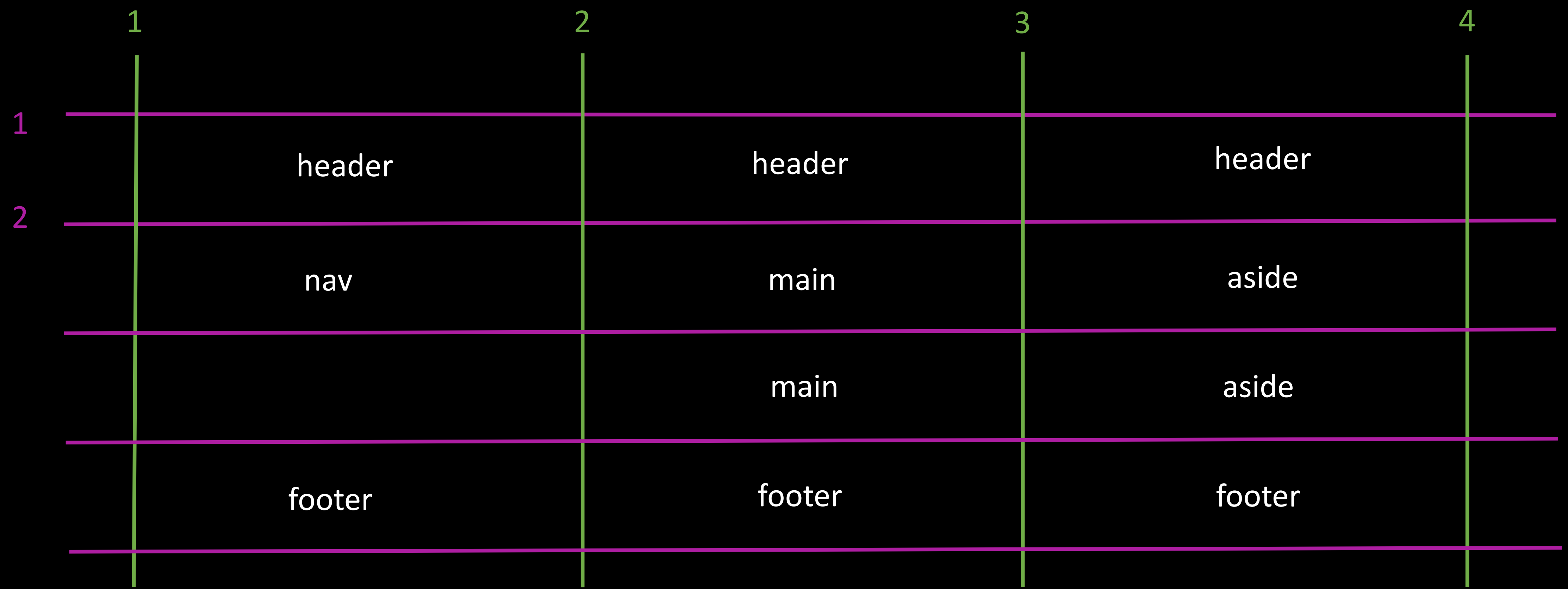


Grid template areas

- Add nav, main and side area using grid-template-areas



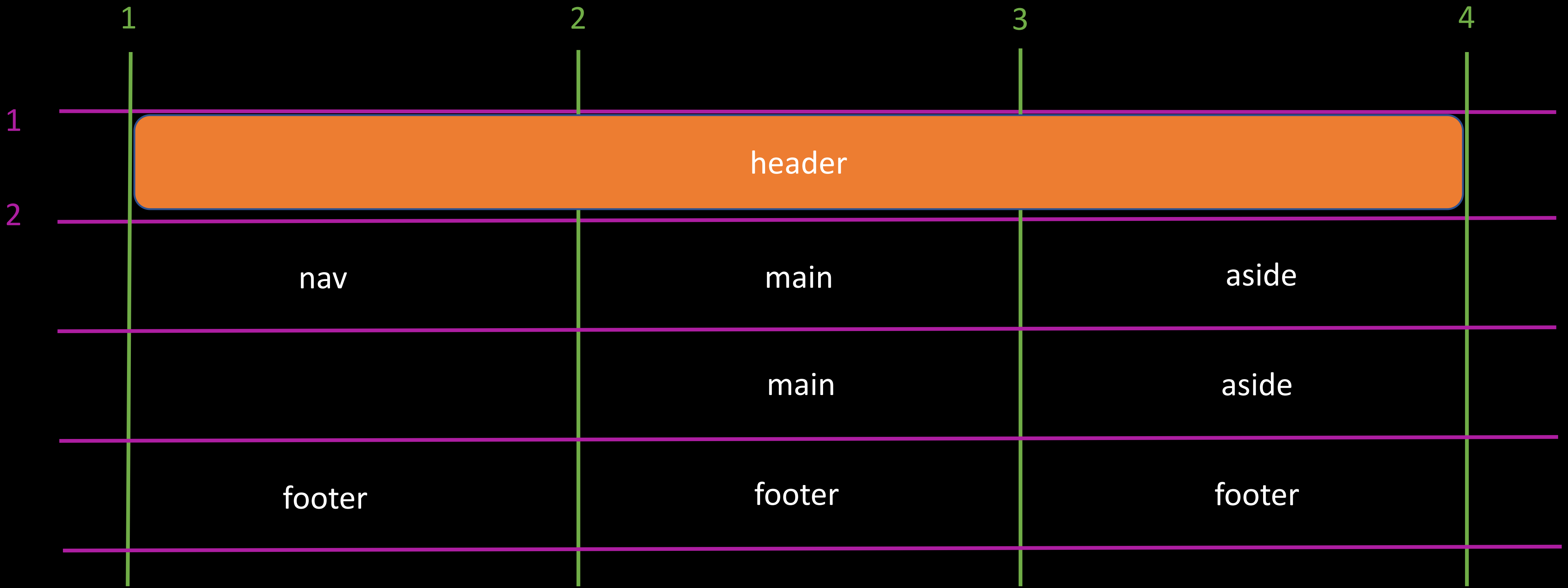
```css
.container {
  /* ...... */
  grid-template-areas:
    "header header header"
    "nav    main   aside"
    ".      main   aside"
    "footer footer footer"
}
```



Grid template areas

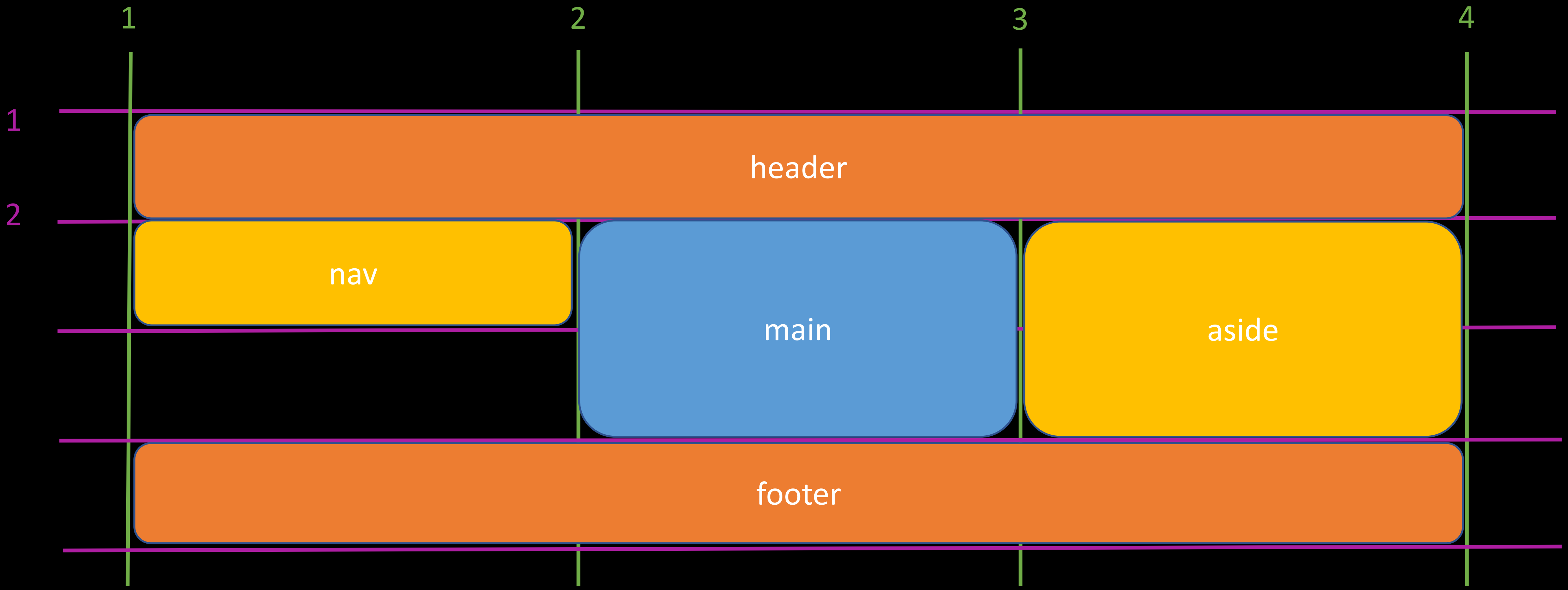
- Speify **grid-area** on the grid item

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



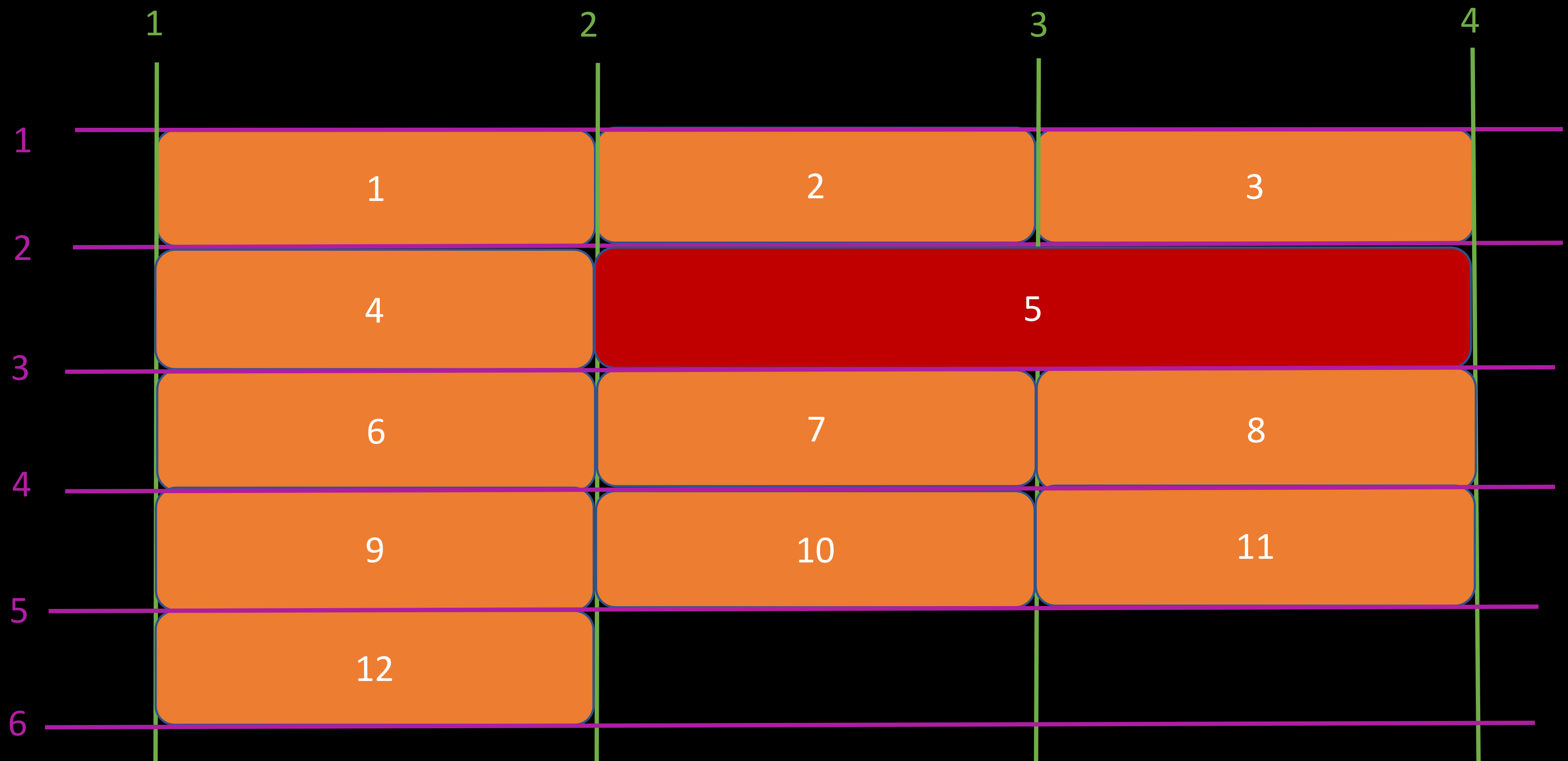
Grid template areas

- Speify **grid-area** on the grid item



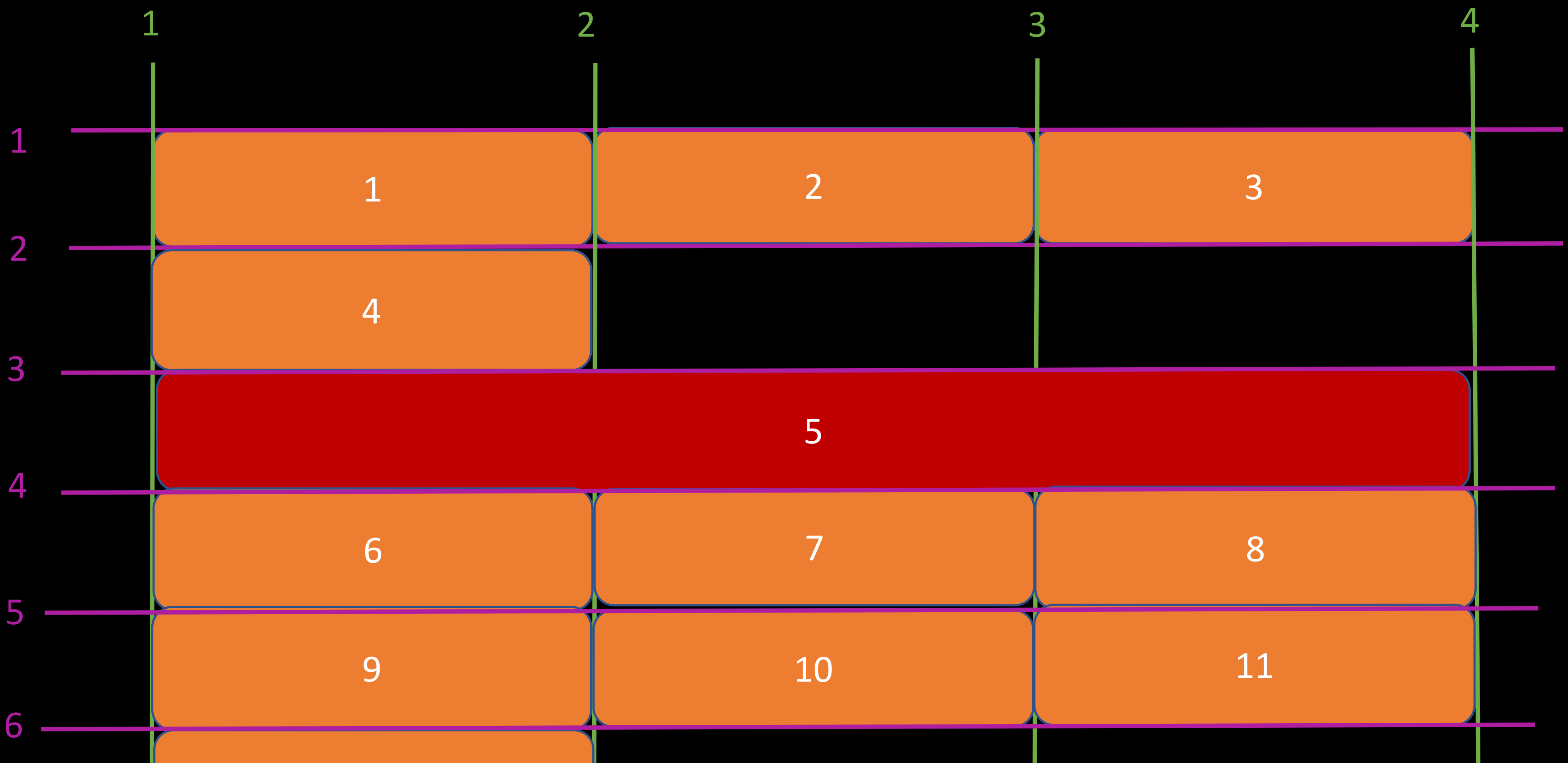
Auto flow: dense

- Set **grid-column: span 2;**



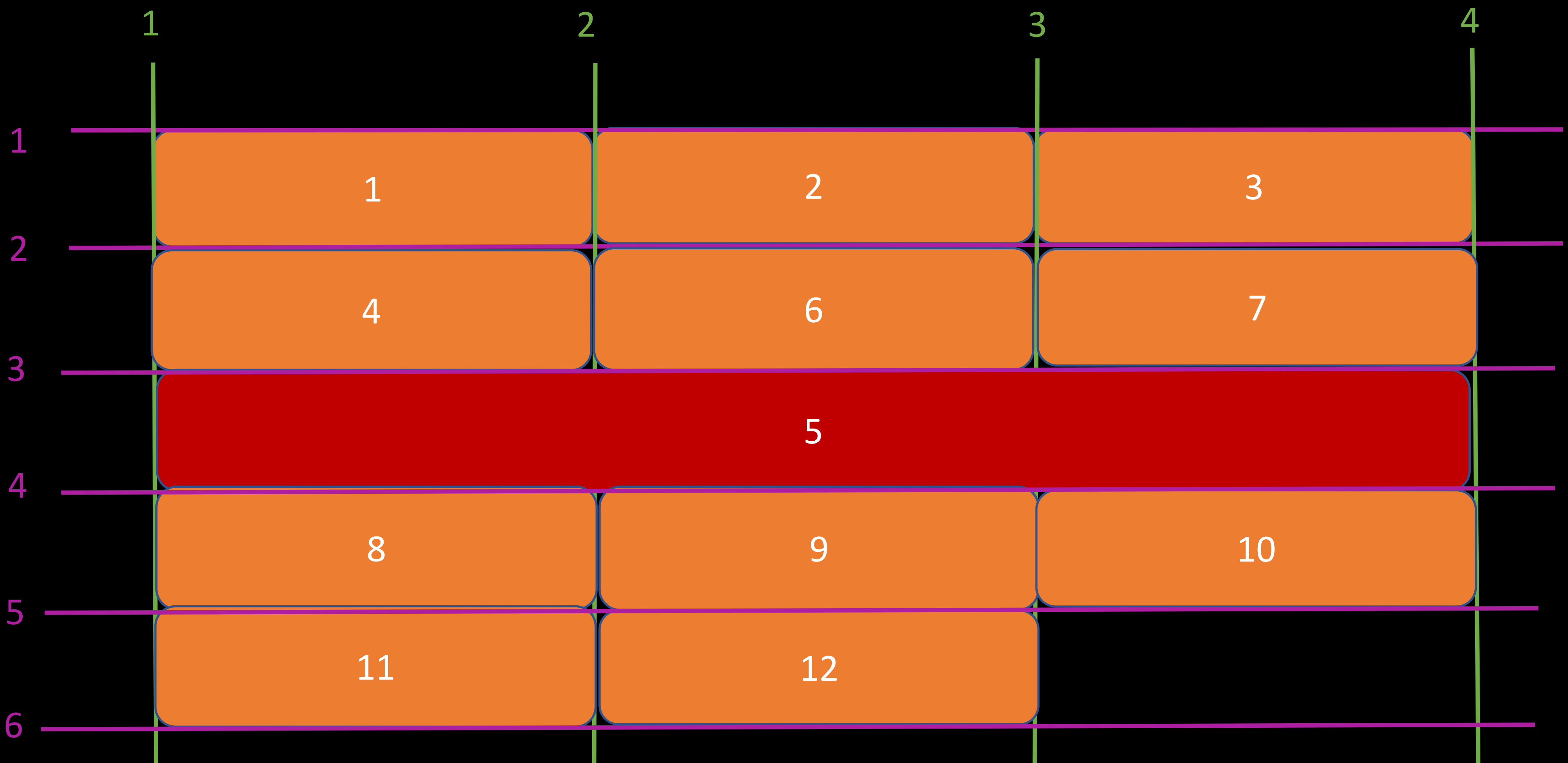
Auto flow: dense

- Set **grid-column: span 3;**



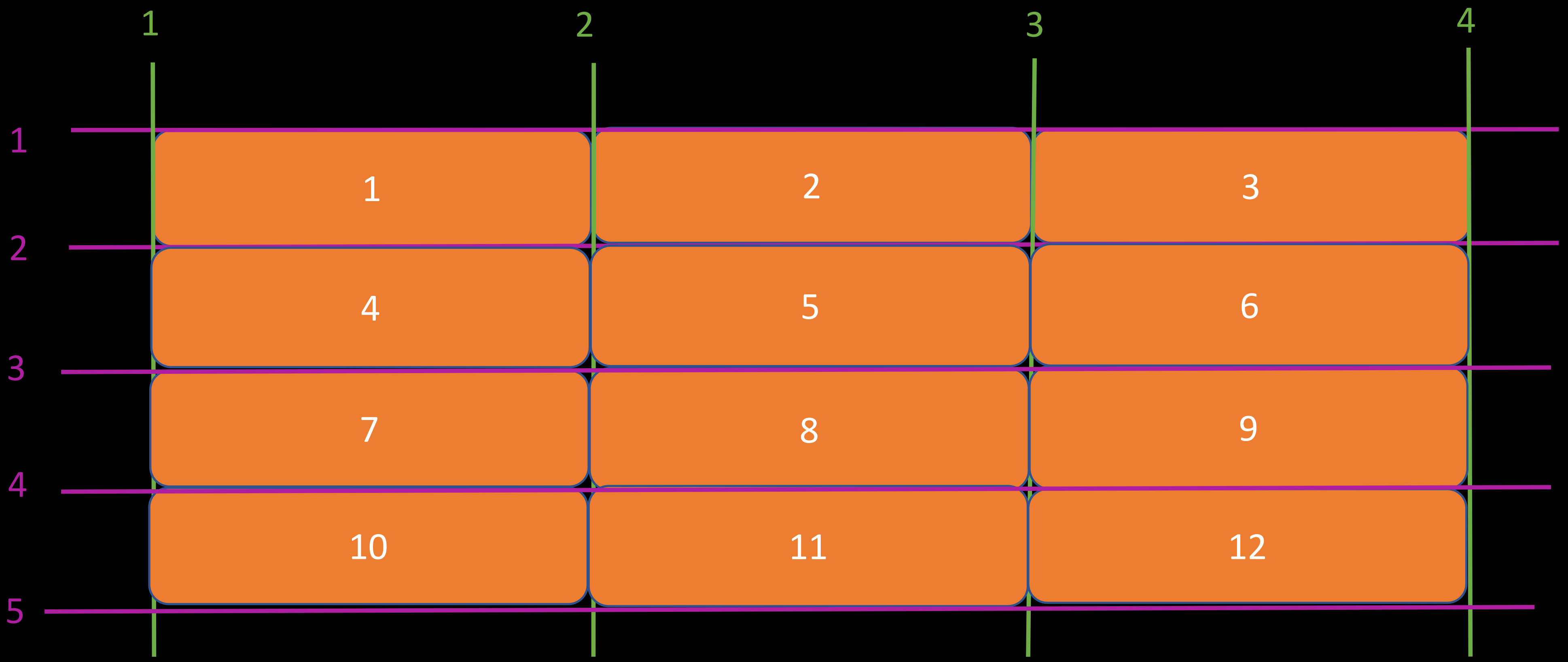
Auto flow: dense

- **Set grid-auto-flow: dense;** on the grid container



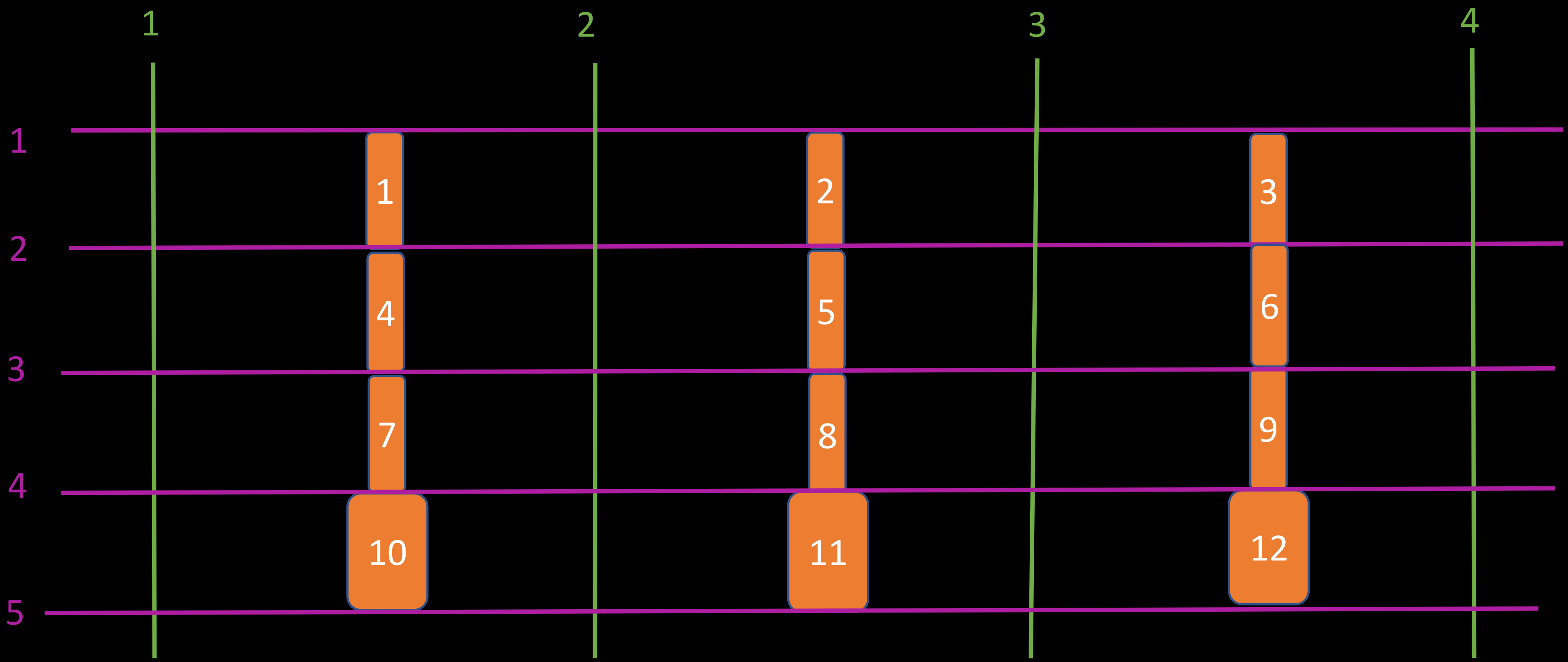
Aligning items: justify-items

- Set **justify-items: stretch** (default)



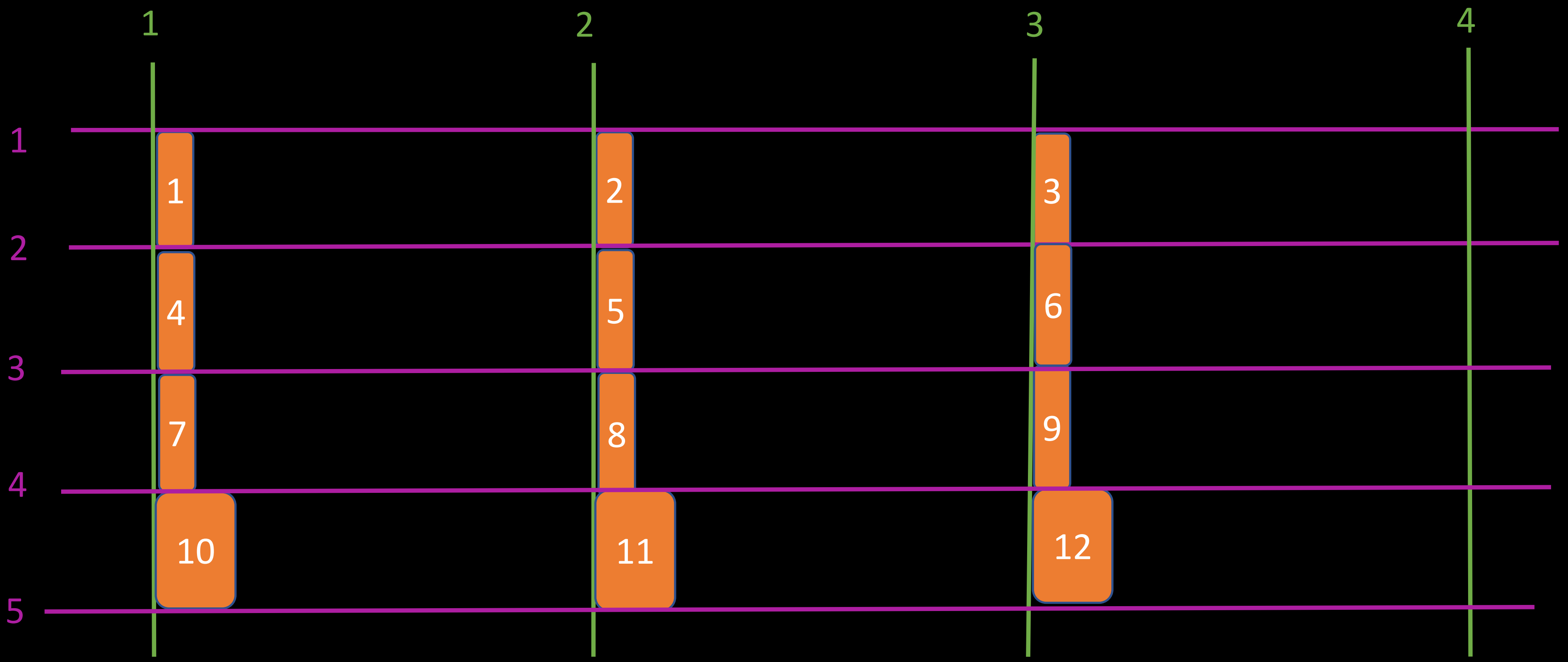
Aligning items: justify-items

- Set **justify-items: center**



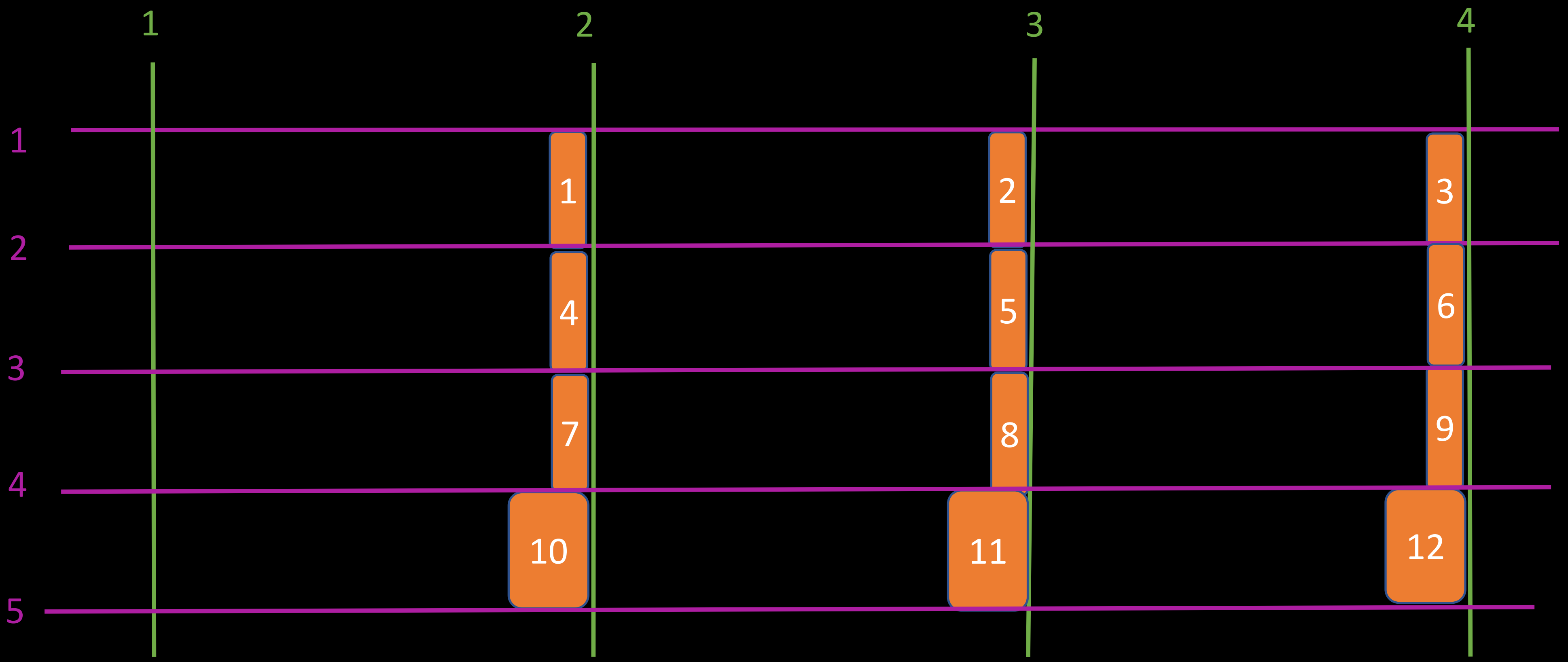
Aligning items: justify-items

- Set **justify-items: start**



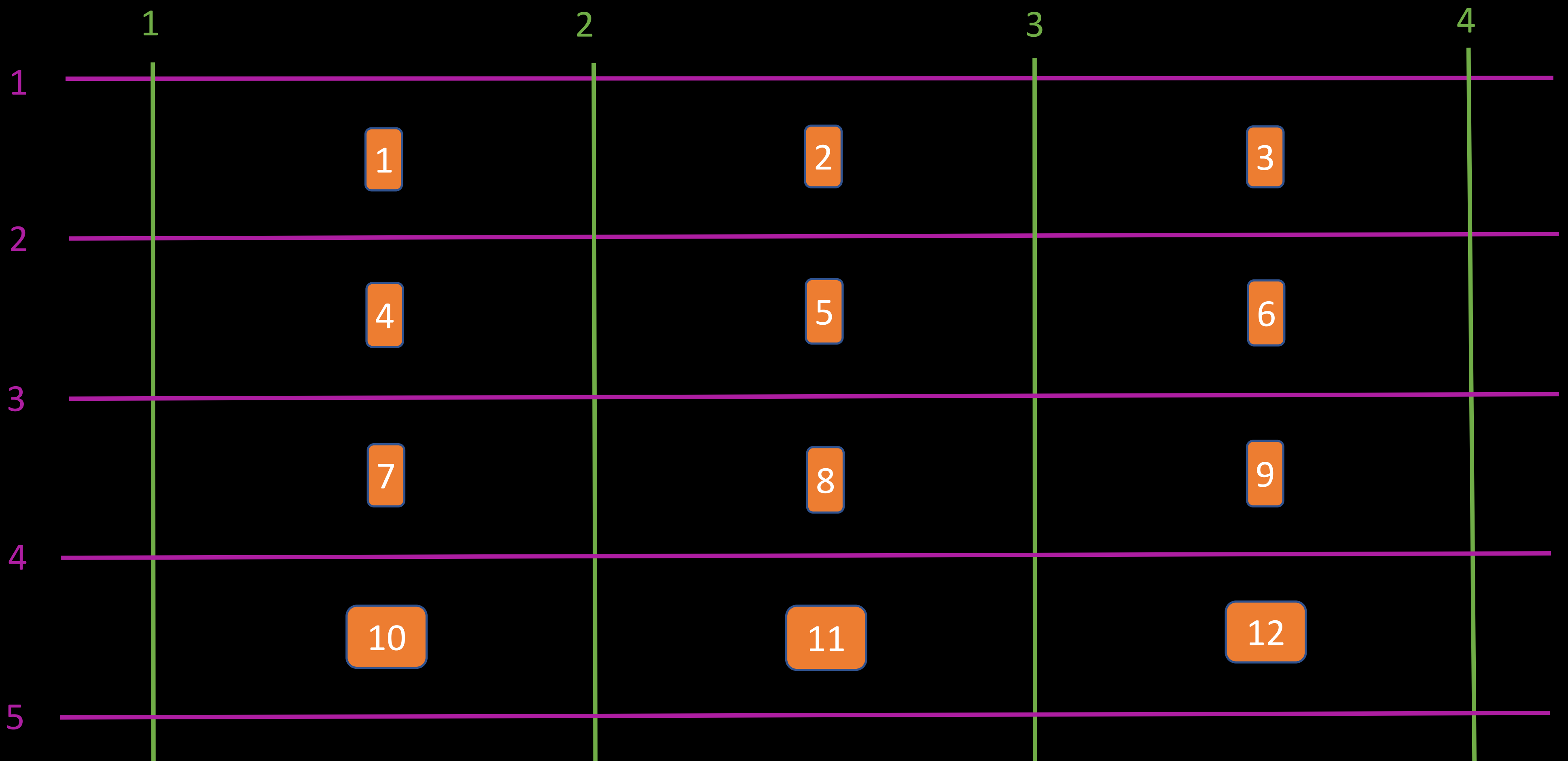
Aligning items: justify-items

- Set **justify-items: end**



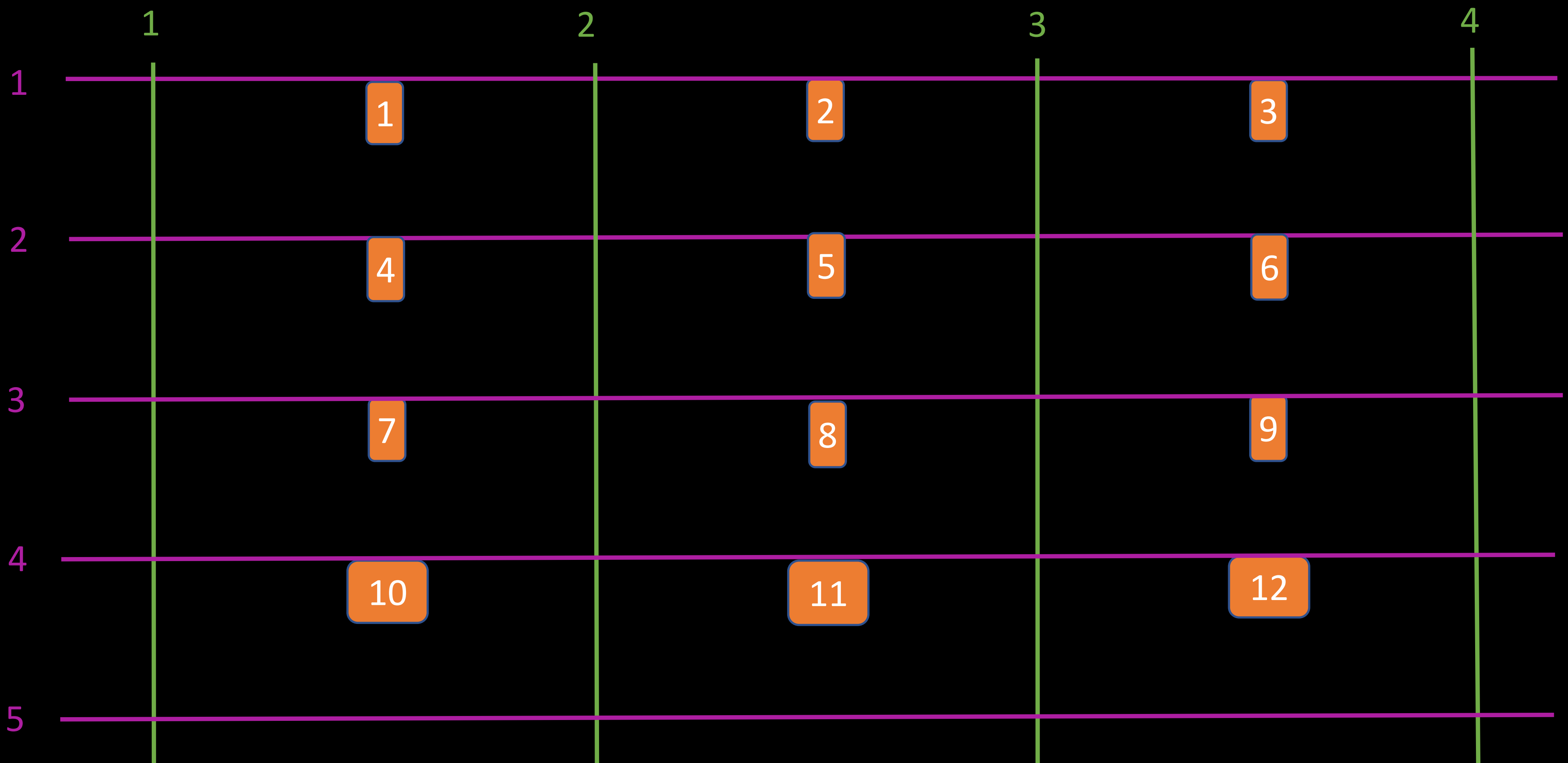
Aligning items: align-items

- grid-template-rows needs to be set so the row have a bigger height than the content
- Set **grid-template-rows: repeat(4, 100px)**
- Set **justify-items: center;**
- Set **align-items: center;**



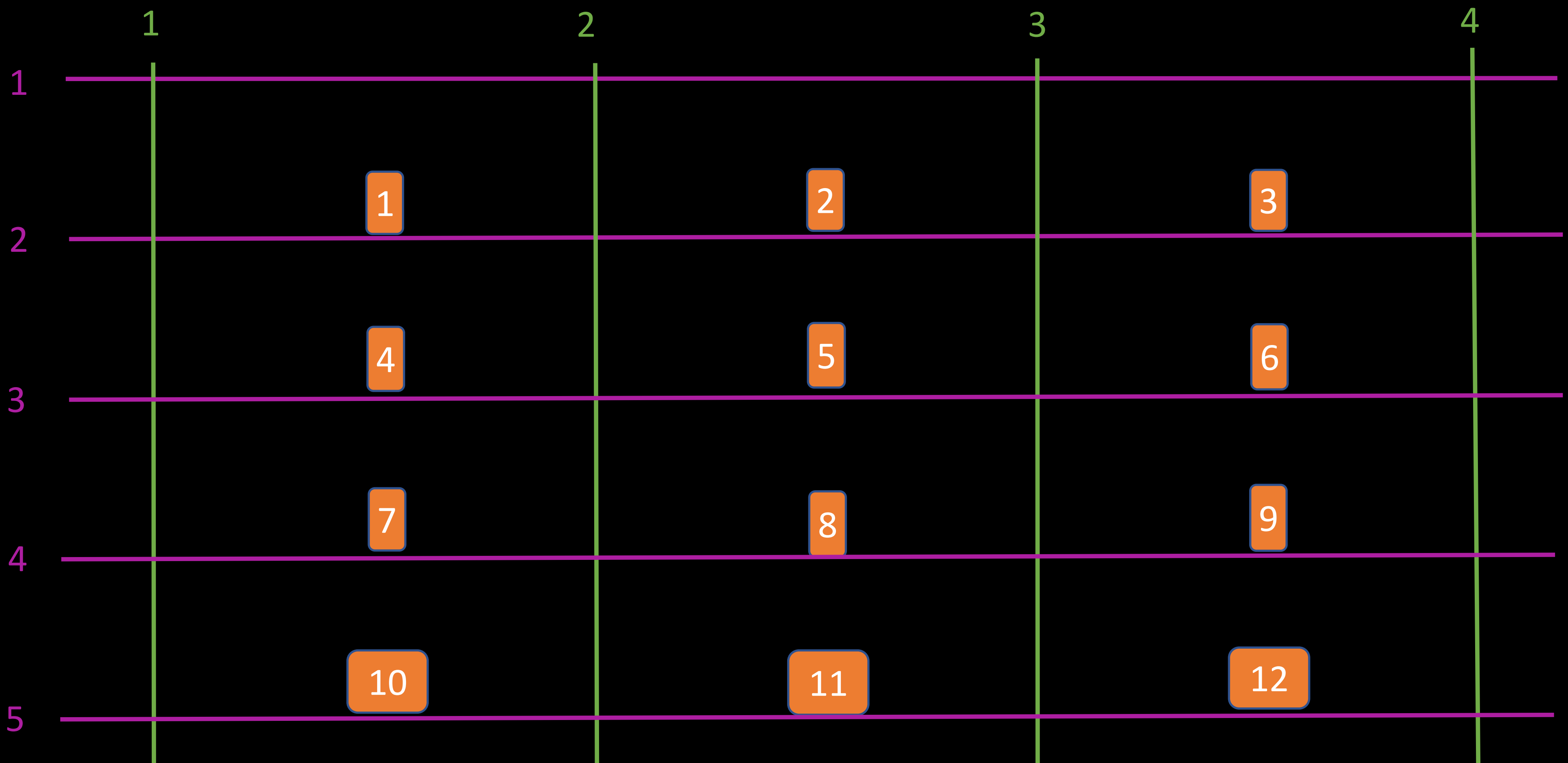
Aligning items: align-items

- Set **align-items: start;**



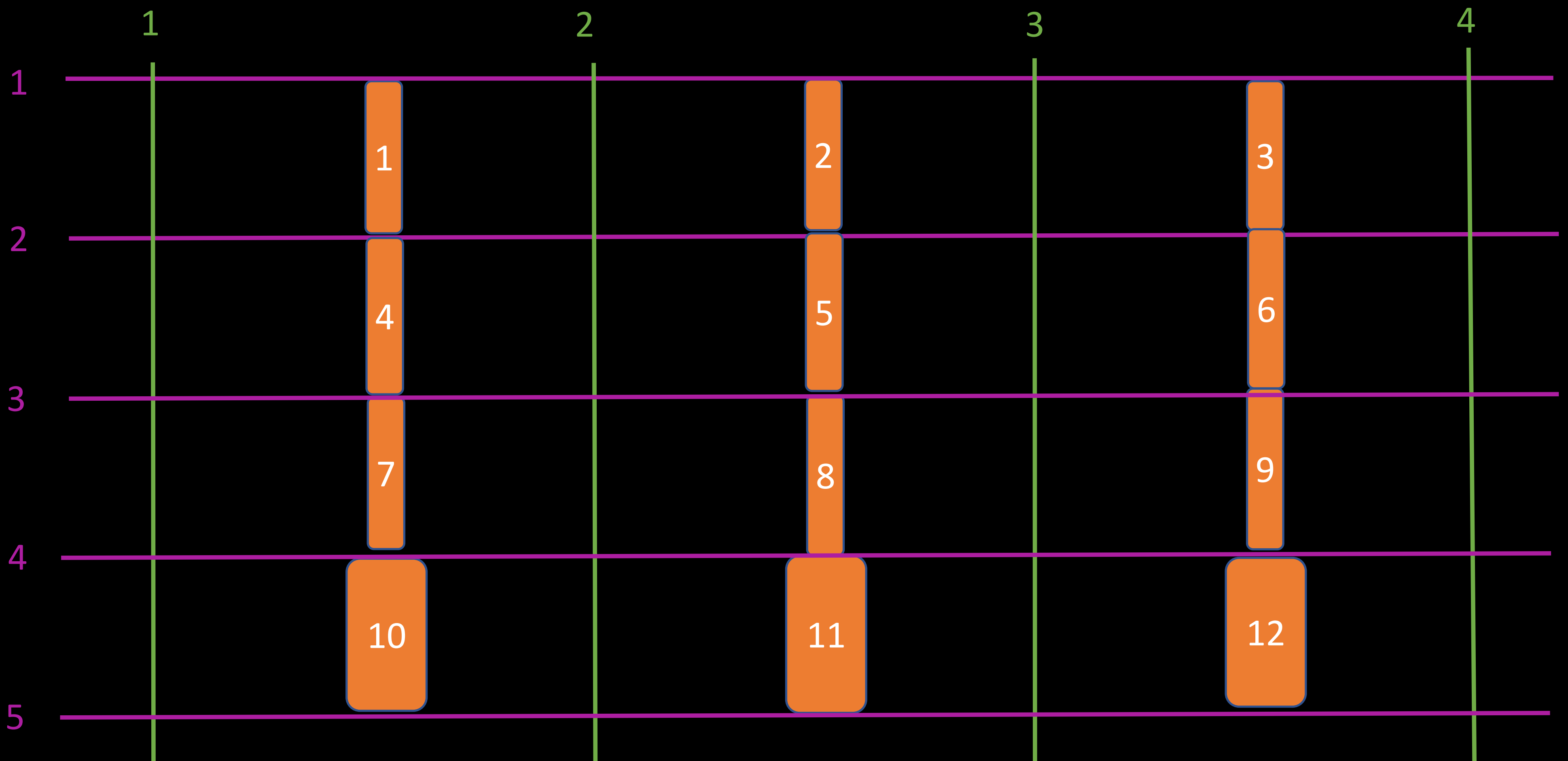
Aligning items: align-items

- Set **align-items: end;**



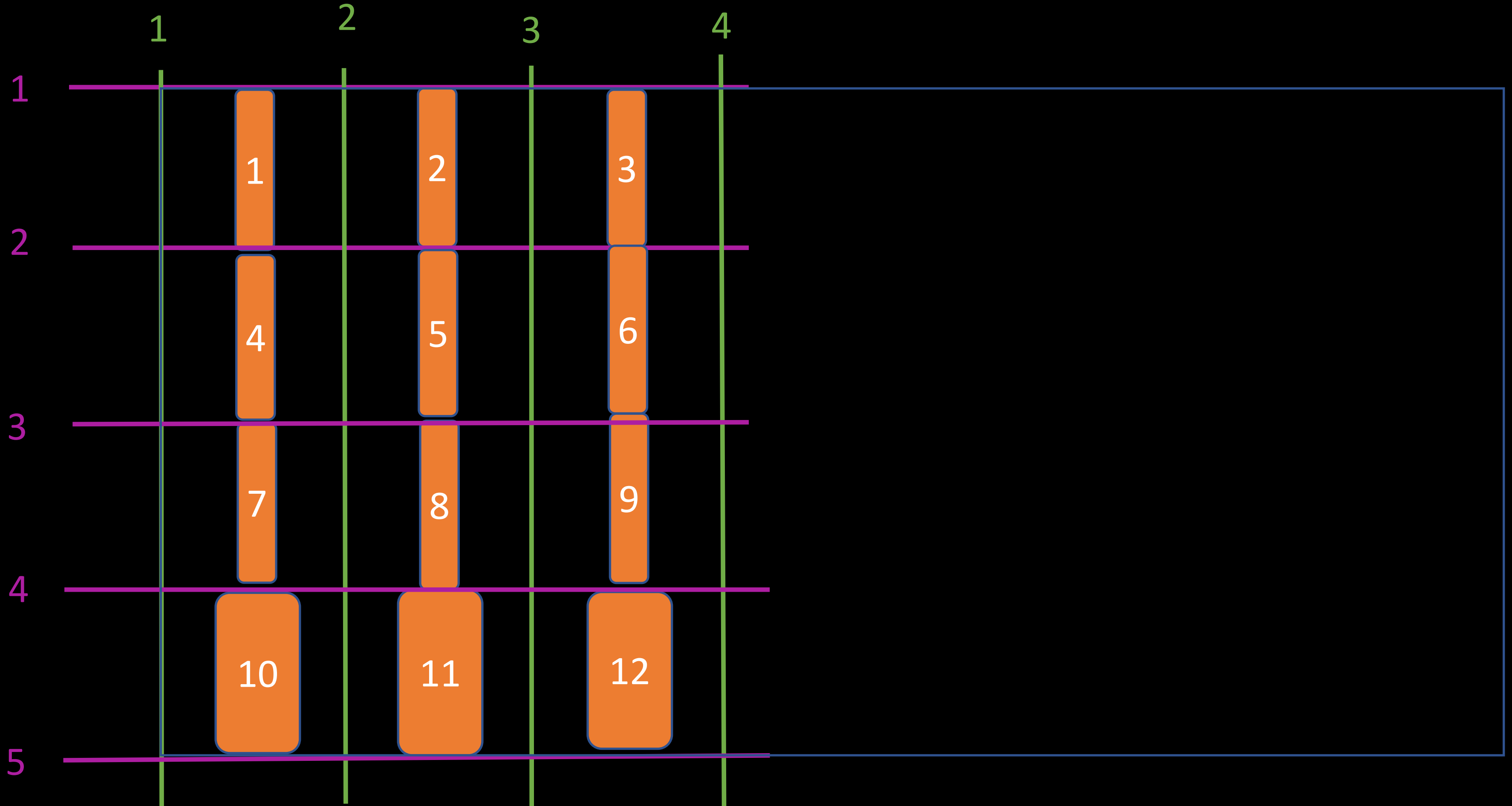
Aligning items: align-items

- Set **align-items: stretch;**



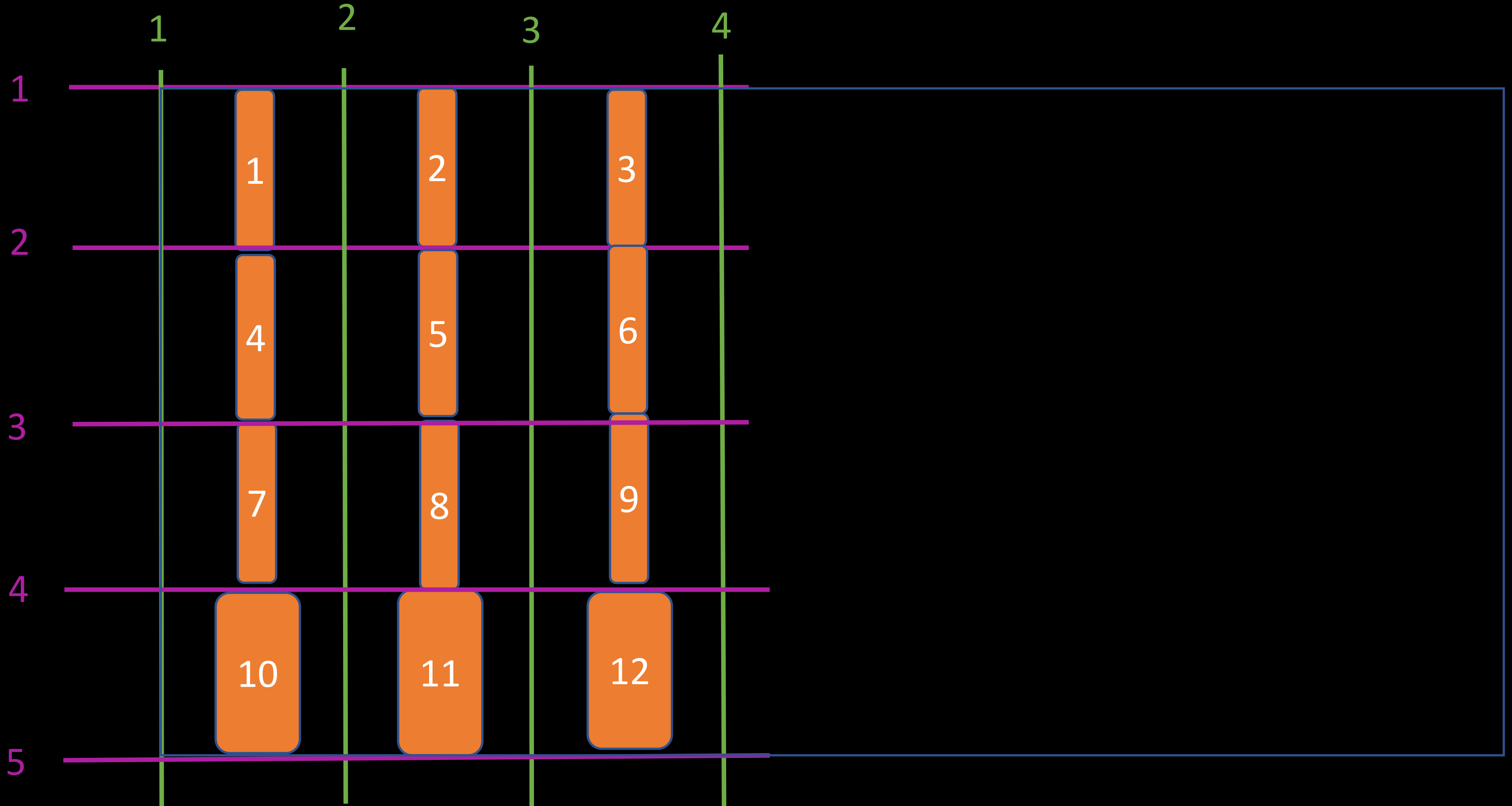
Justify content

- What do we do with all that extra space?
- Set **grid-template-columns: repeat(3, 50)**



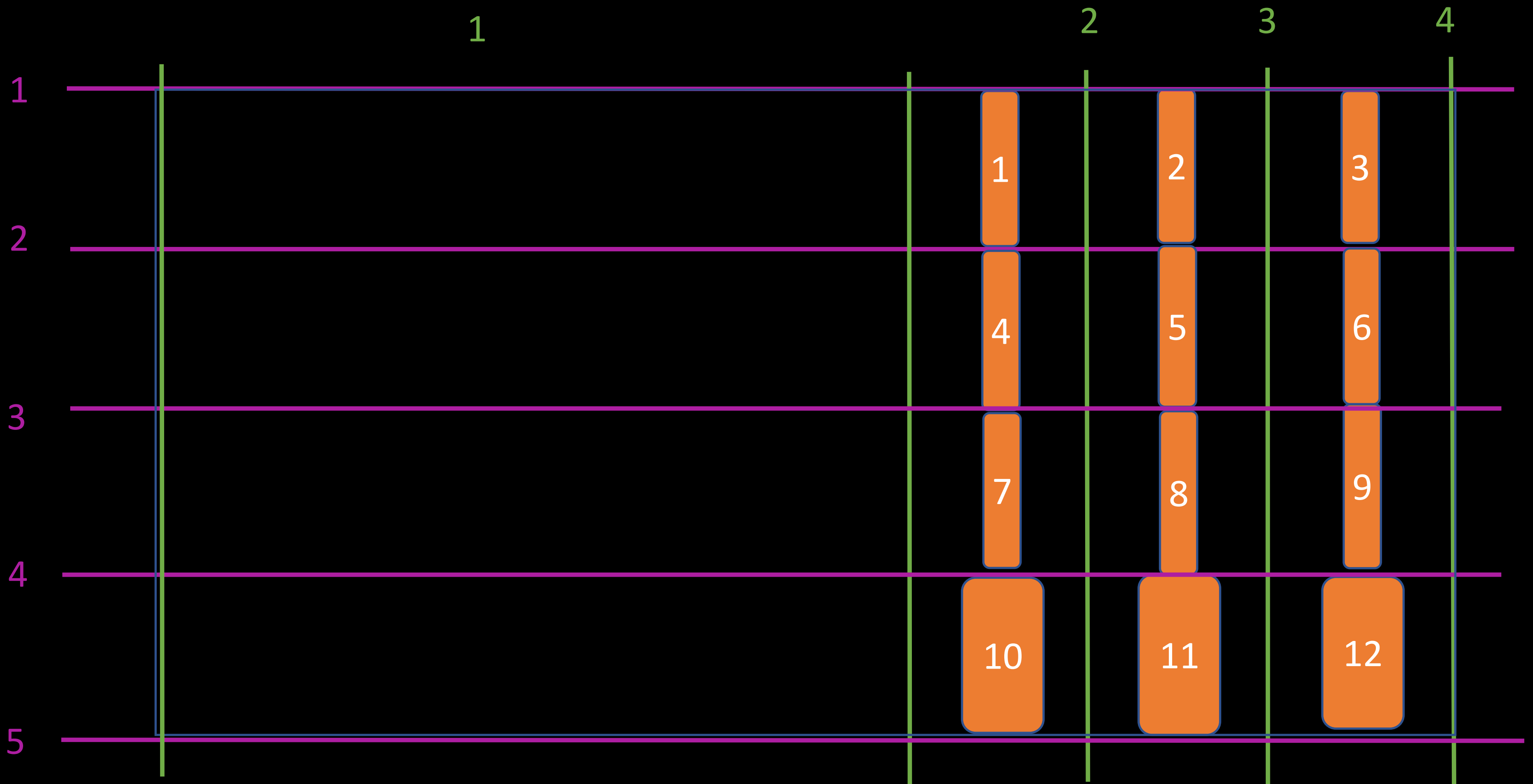
Justify content

- Set **justify-content: start (default)**



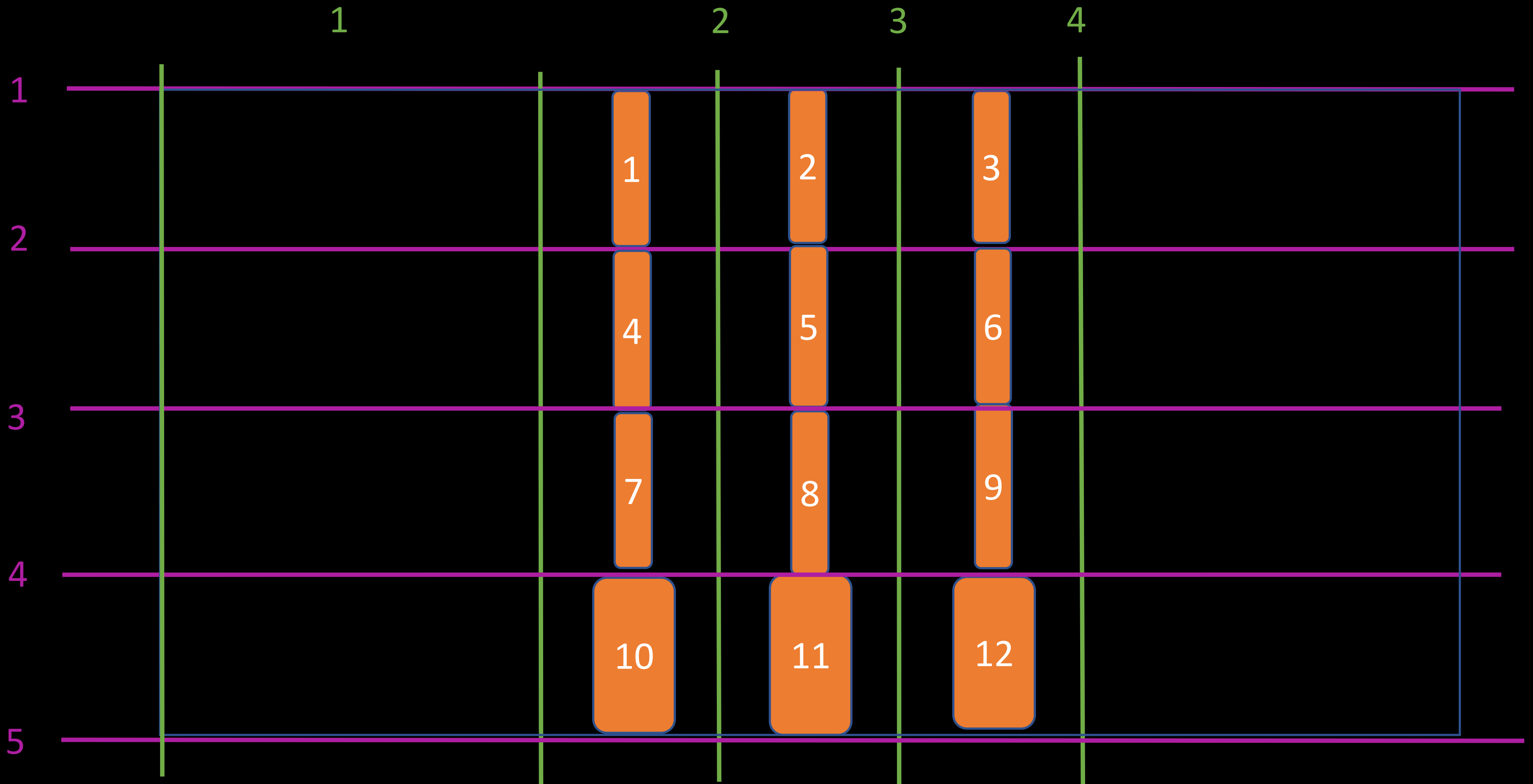
Justify content

- Set **justify-content: end(default)**



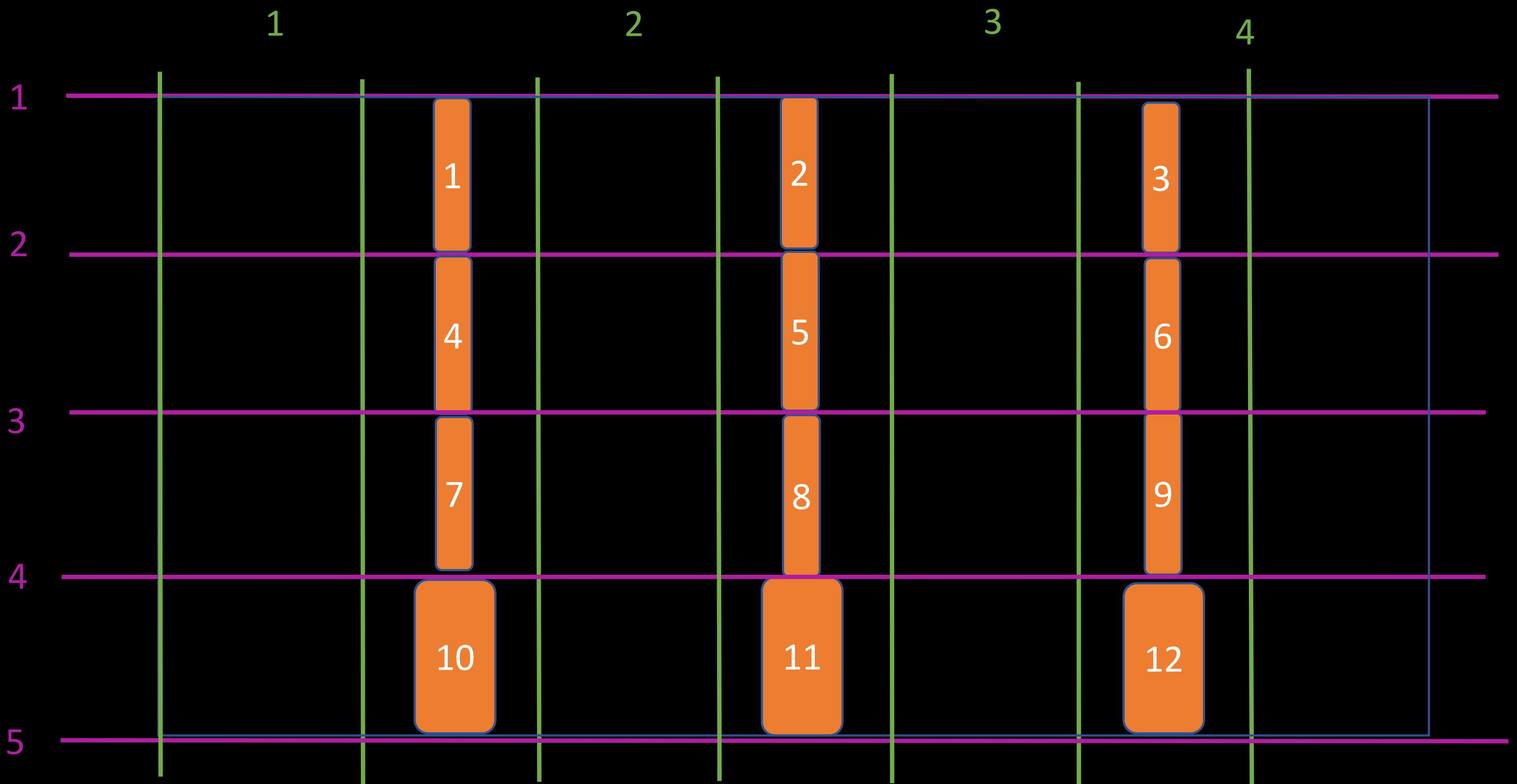
Justify content

- Set **justify-content: center**



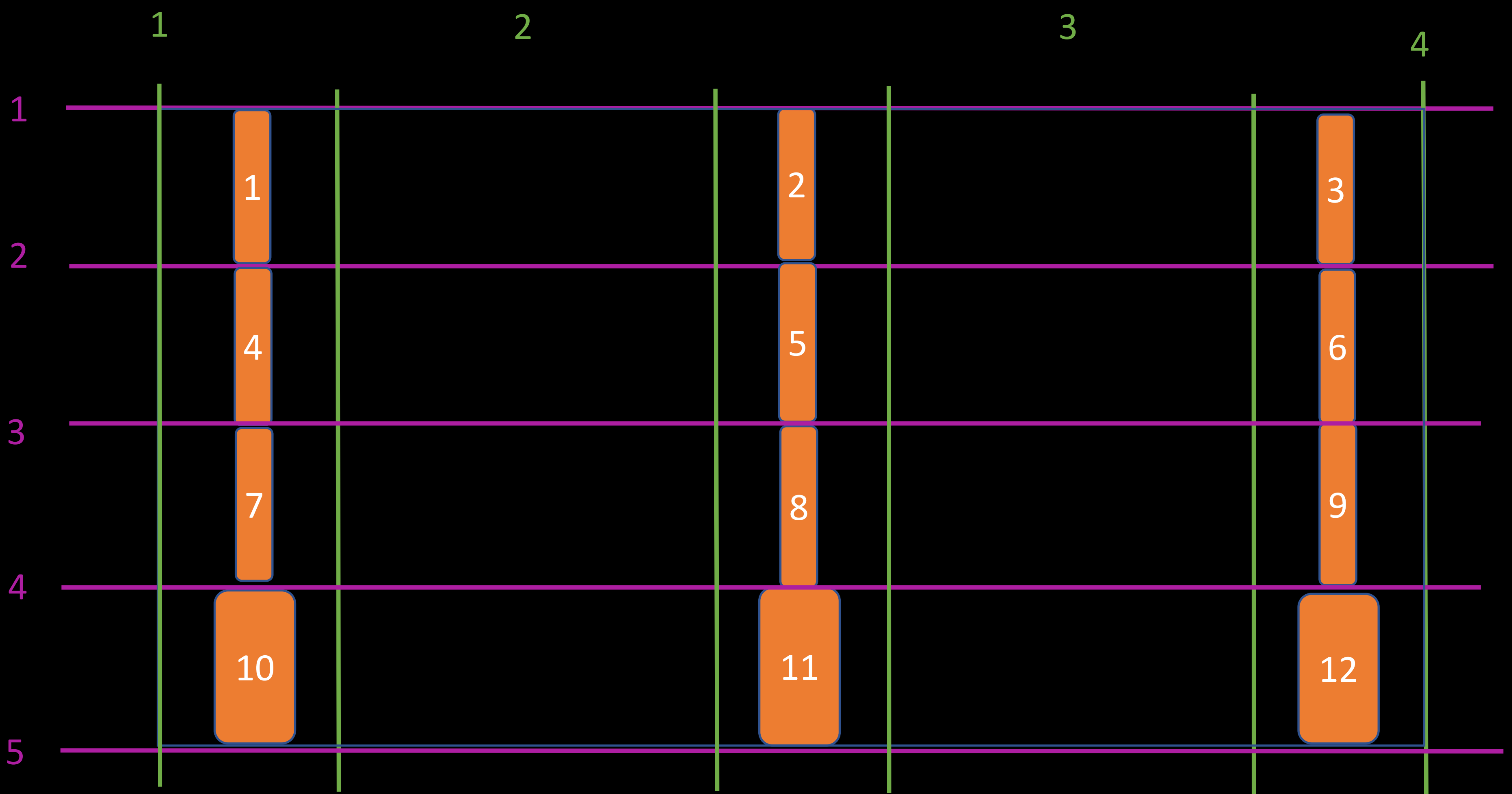
Justify content

- Set **justify-content: space-around**



Justify content

- Set **justify-content: space-between**



Align items

- Similar to justify content but affects the y axis. 😁
- Same values as on justify-content.
- Virtually unused since very rarely we have fixed height grids.

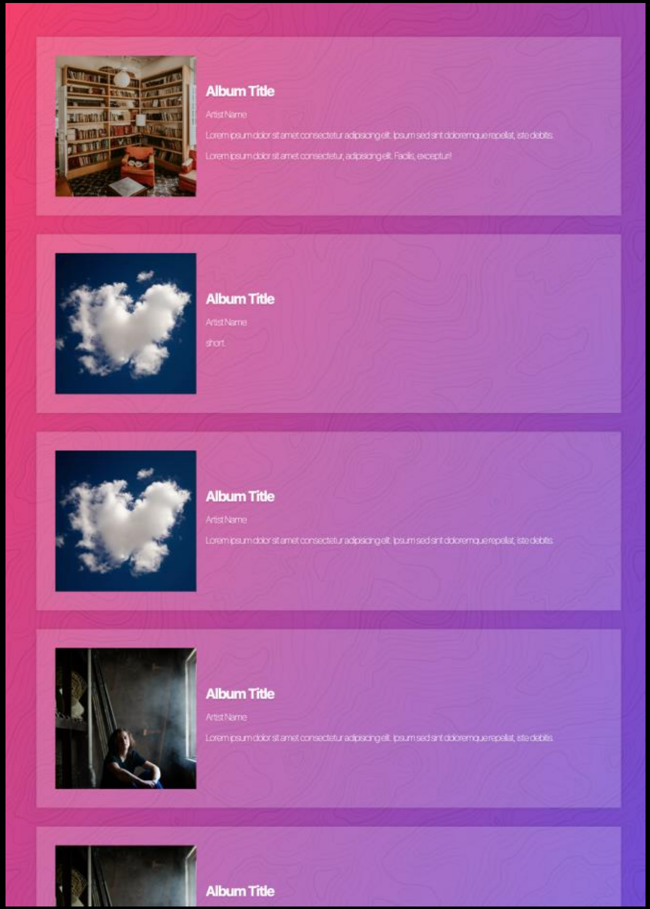
Justify self and align self

- Same values as justify-content and align-items.
- Modify each grid item on a case by case basis.

Exercise

- Implement Holy Grail layout.

Implement the following



Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Facilis, excepturi!

Album Title

Artist Name

short.

Album Title

Artist Name

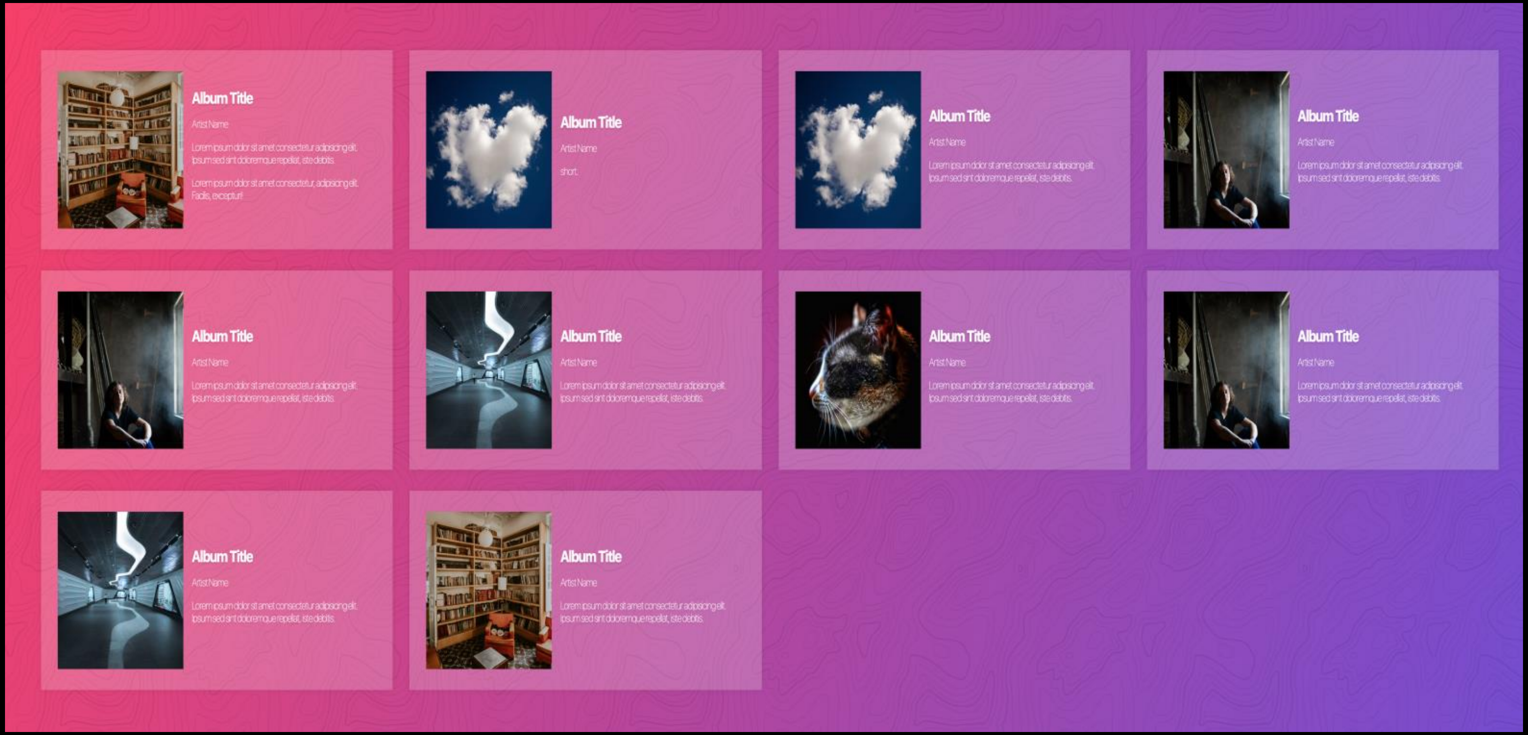
Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title



Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Facilis, excepturi!

Album Title

Artist Name

short.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

Artist Name

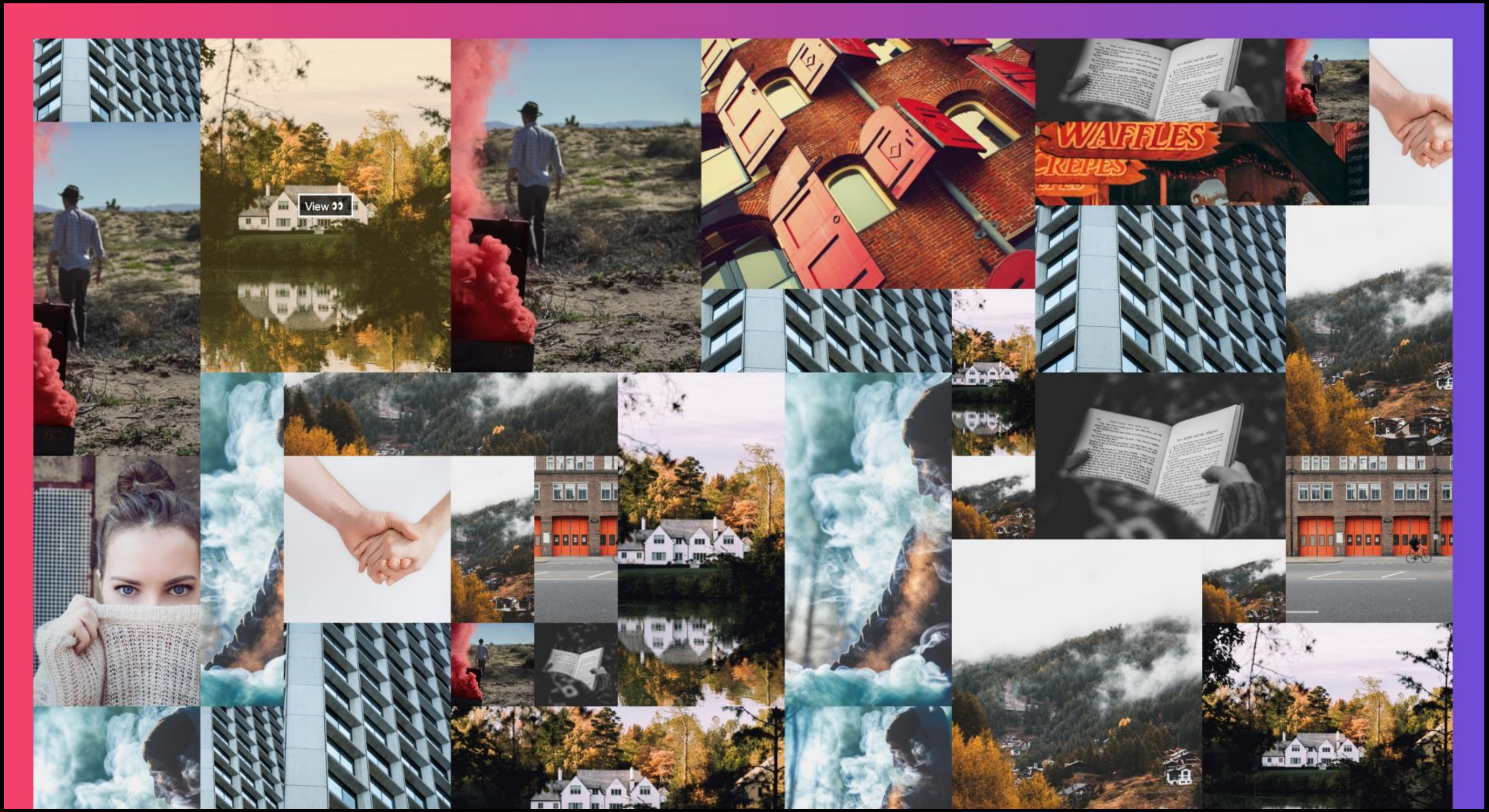
Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Album Title

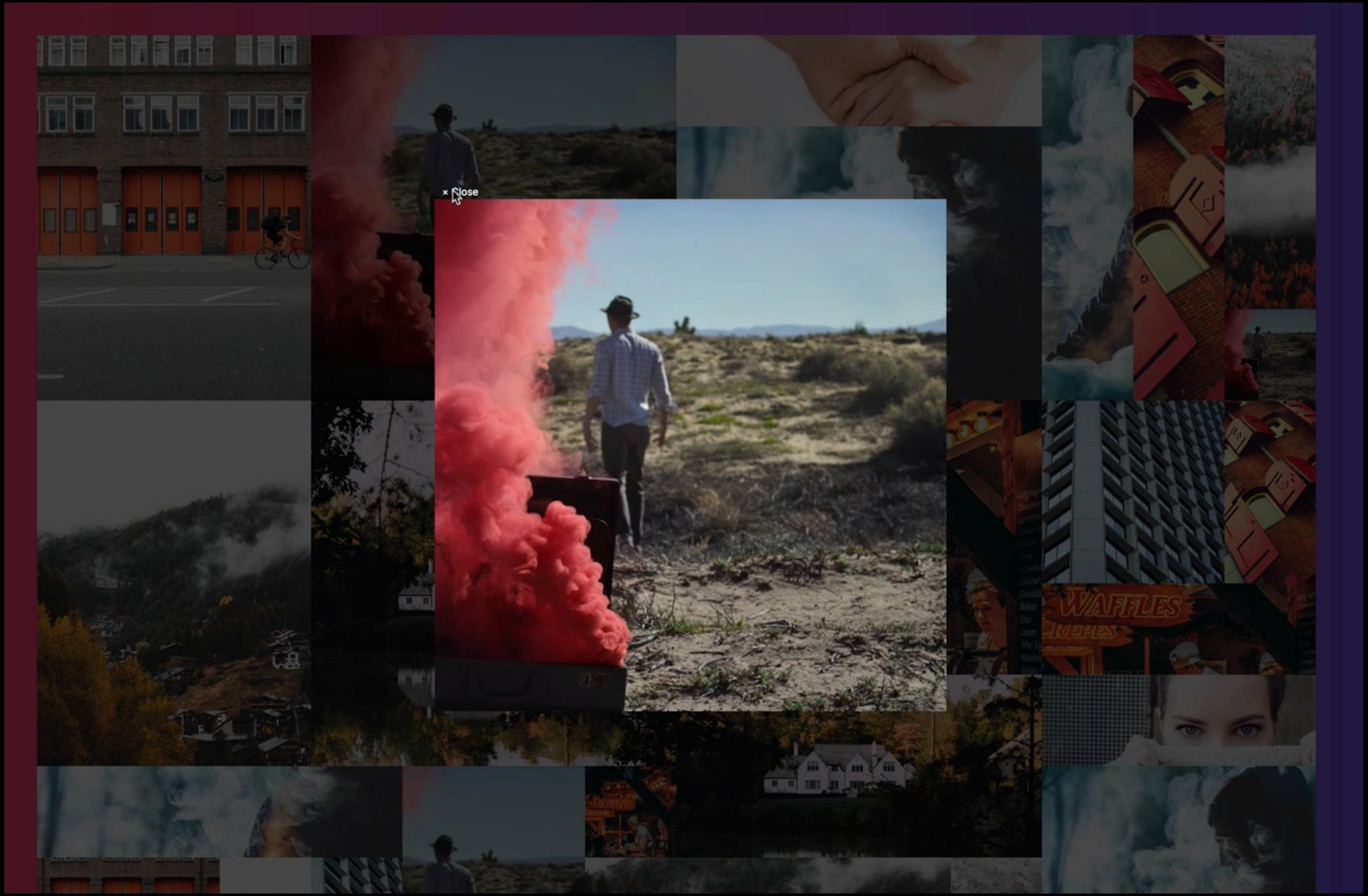
Artist Name

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsum sed sint doloremque repellat, iste debitis.

Implement the following image gallery 😁



Implement the following image gallery 😁


× Close
WAFFLES
CREPES