CSS Flex

BCA III NAST

Terminology


```
<div class="container">
    <div>Item 1</div>
    <div>Item 2</div>
    <div>Item 3</div>
</div>
```

Flex Container

Flex Items

Flexbox axes

Main Axis

Cross Axis

Flex Container

Flex Container Properties

- display
- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

display

Create either a block level or inline level flex container.

- flex
- inline-flex

flex-direction

Sets the direction of the main axis.

- row
- row-reverse
- column
- column-reverse

flex-wrap

Control the wrapping of flex items within the container.

- nowrap
- wrap
- wrap-reverse

flex-flow

Shorthand for flex direction and flex wrap.

flex-flow: <flex-direction> <flex-wrap>

justify-content

Align items and distribute any extra spacing in the parent container.

The alignment is always along the main axis.

- flex-start
- flex-end
- center
- space-between
- space-around
- space-evenly

align-items

Align items along the cross axis.

- flex-start
- flex-end
- center
- baseline
- stretch

align-content

Aligns lines of content along the cross axis and distribute any extra spacing in the parent container.

- flex-start
- flex-end
- center
- space-between
- space-around
- stretch

Flex Items

Flex item properties

- order
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

order

Control the order of items in the flex container.

Integer value

flex-grow

Dictates what amount of the available space inside the flex container the item should take up.

Relative to the other items in the container.

Default value is 0 – items do not grow.

flex-grow value of 1 - flex items grow evenly.

flex-shrink

Dictates the shrink factor of the flex items when the default size of flex items is larger than the flex container.

Relative to the other items in the container.

Default value is 1.

flex-basis

Set the initial size of a flex item.

Pixels, percentages or relative units.

Default value is auto.

flex

Short hand for flex grow, flex shrink and flex basis.

```
.item {
    flex-grow: 2;
    flex-shrink: 5;
    flex-basis: 200px;
}
```

```
.item {
    flex: 2 5 200px;
}
```

flex: <flex-grow> <flex-shrink> <flex-basis>

Default

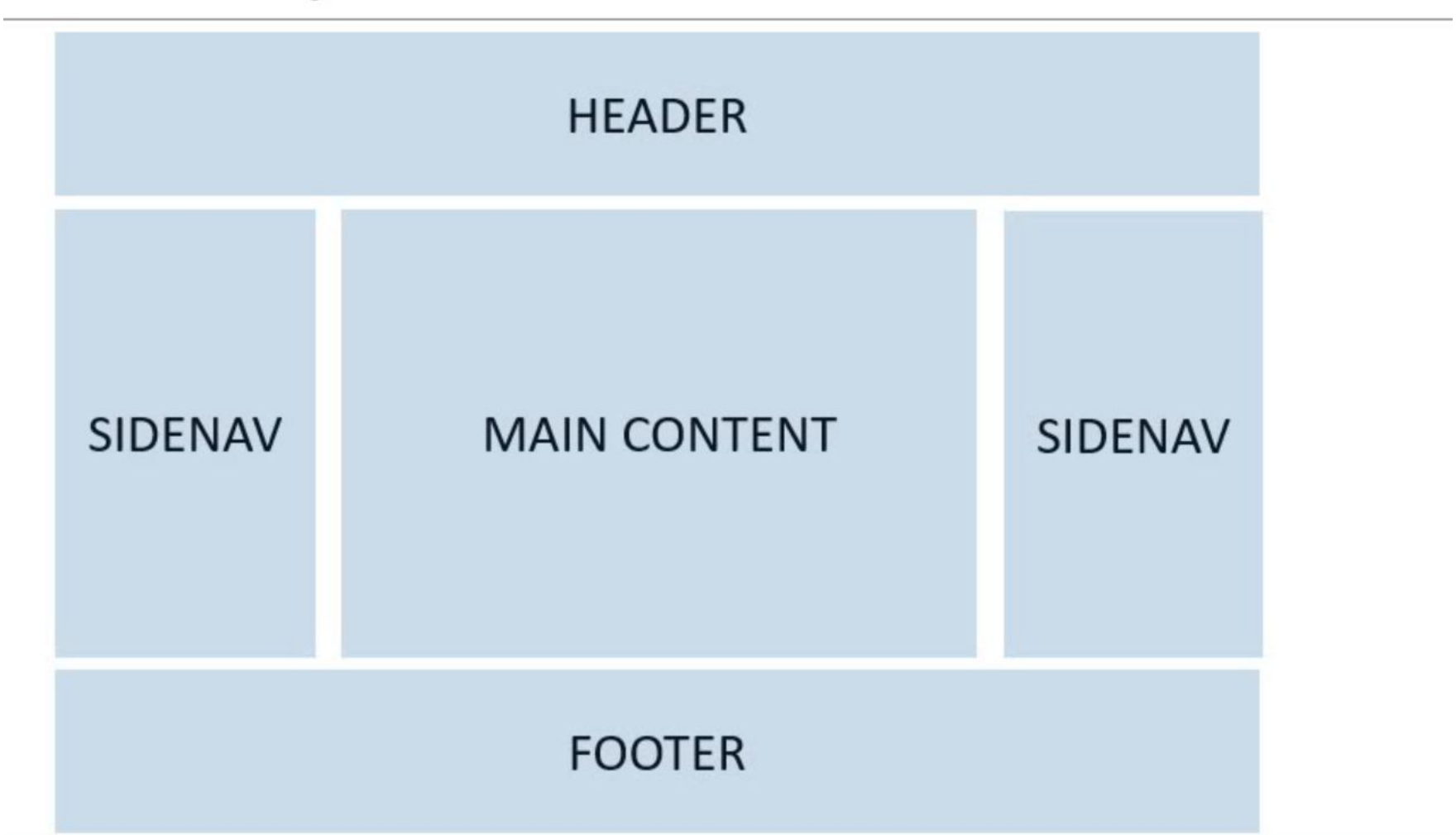
```
.item {
    flex: 0 1 auto;
}
```

align-self

Align the items individually.

Values like auto, flex-start, flex-end, center and stretch.

Overrides the align-items value of the flex container.

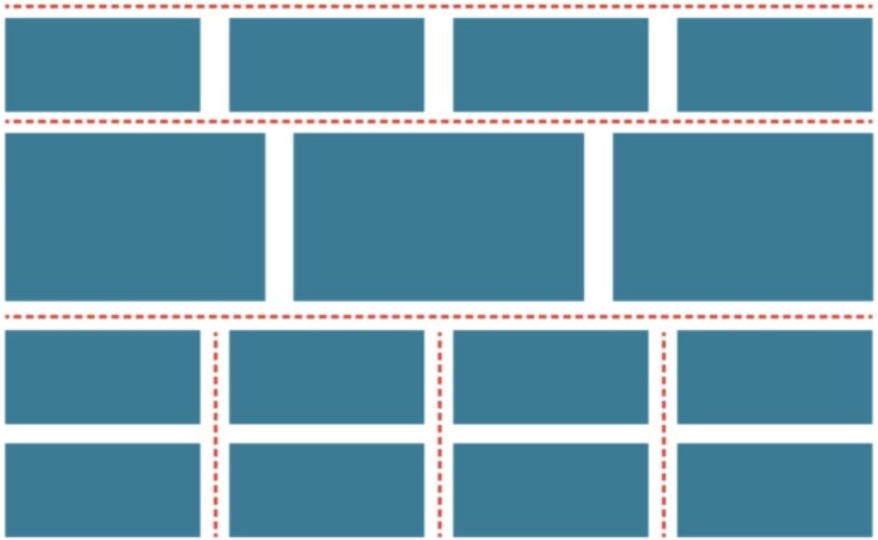

HEADER
SIDENAV
MAIN CONTENT
SIDENAV
FOOTER

CSS Grid

BCA III NAST

flexbox

One dimensional grid



CSS grid

two dimensional grid

flexbox

.wrapper

.content-one

.content-two

.aside

CSS grid

Less markup

.content-one

.content-two

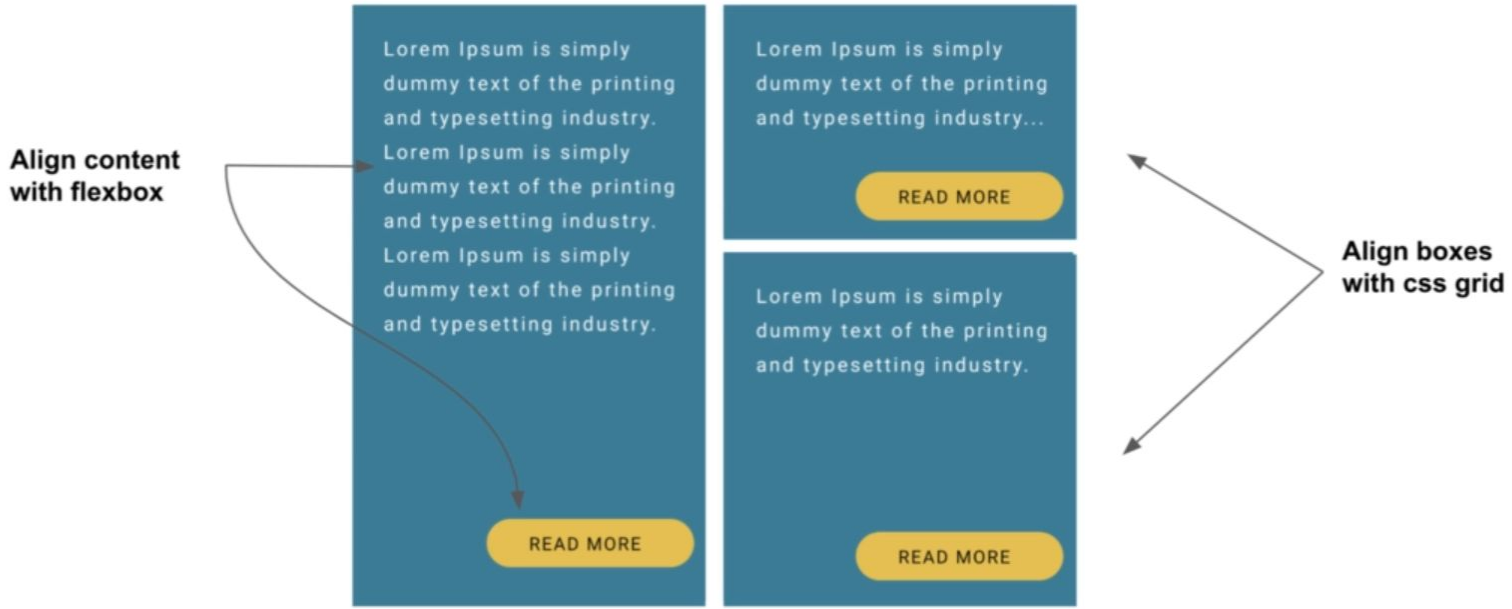
.aside

flexbox

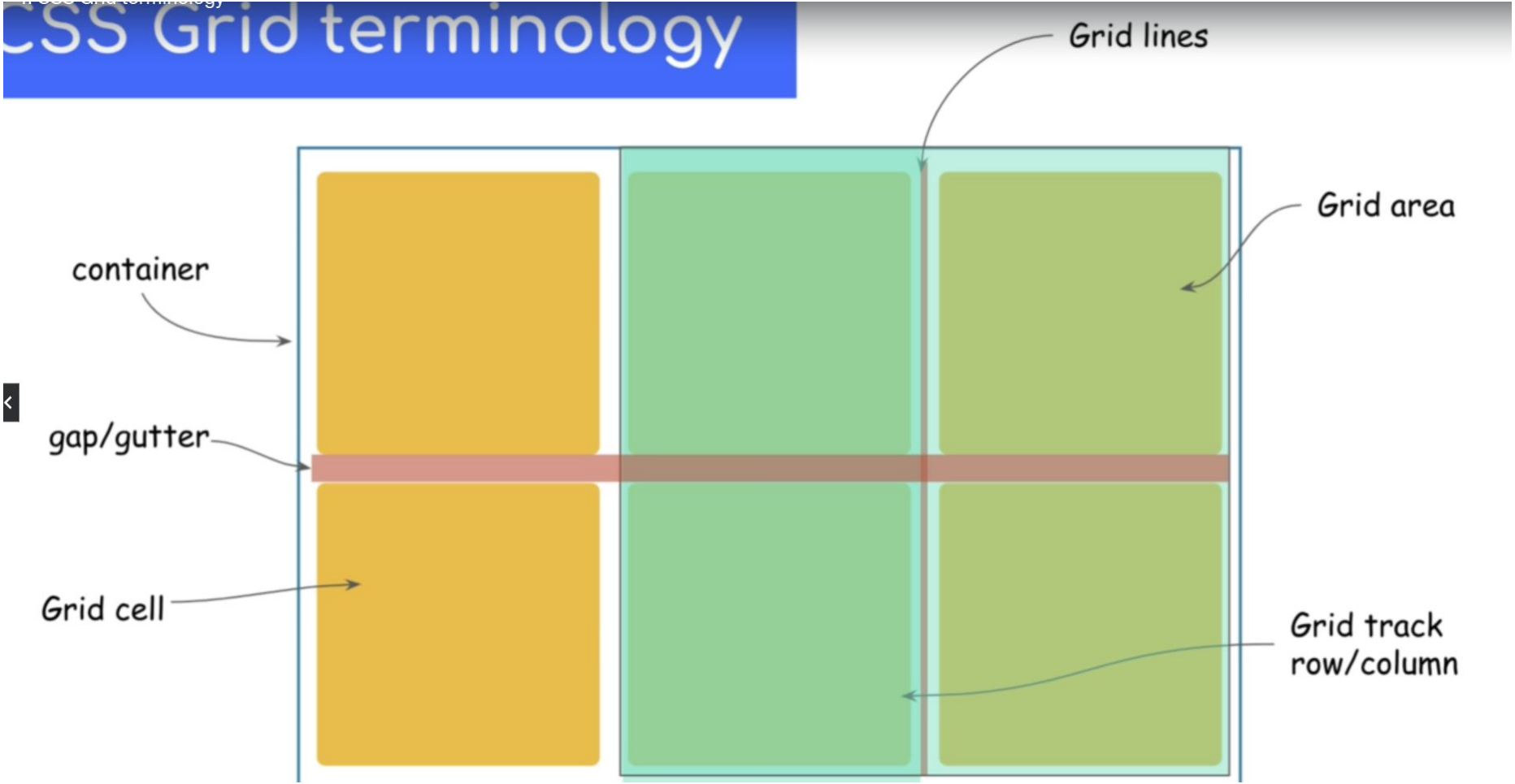
Better for aligning the content that inside the layout items

CSS grid

For adjusting the layout



CSS Grid terminology



Grid Container

Grid Container Properties

- Display
- Grid-template-rows
- Grid-template-columns
- Grid-template-areas
- gap
- align-items
- Justify-items

Display

Display: creates either block level or inline level grid container

1. grid
2. inline-grid

Grid-template-rows

Define the number and size of rows:

1. fraction
2. Percentage
3. Px, or any other units

Grid-template-columns

Define the number and size of columns:

1. fraction
2. Percentage
3. Px, or any other units

Grid-template

Define the number and size of rows and columns:

1. fraction
2. Percentage
3. Px, or any other units

grid-template: 20px 30px/repeat(5, 1fr);

Row-gap and column-gap

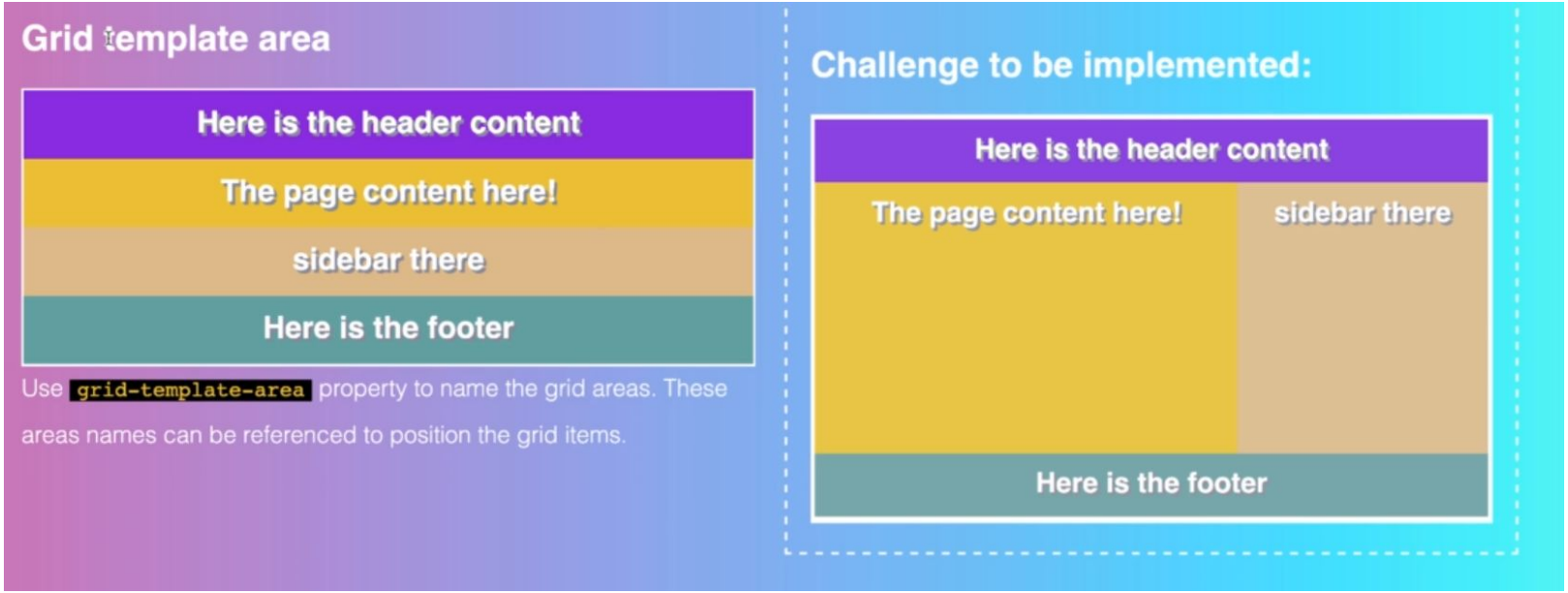
Sets the spacing between rows and column:

row-gap: 20px

column-gap: 10px;

gap : 20px 10px;

Grid template area



Grid Item

Positioning items

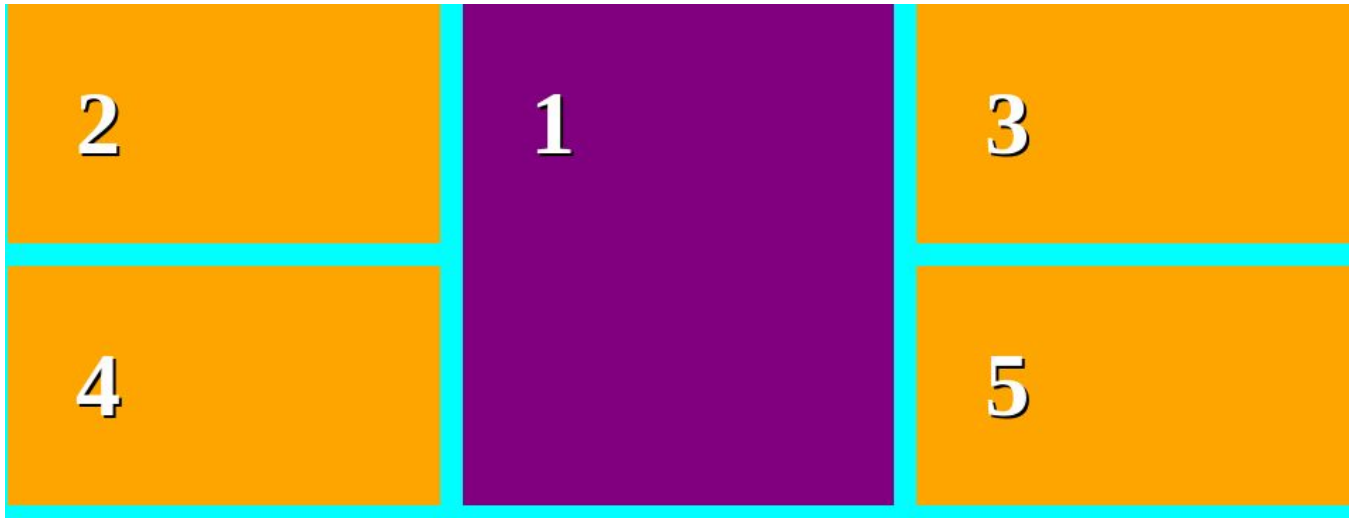
Positioning items with **line numbers**:

Grid-column-start:
Grid-column-end
Grid-row-start:
Grid-row-end:

In short

Grid-column:
Grid-row:

Positioning items

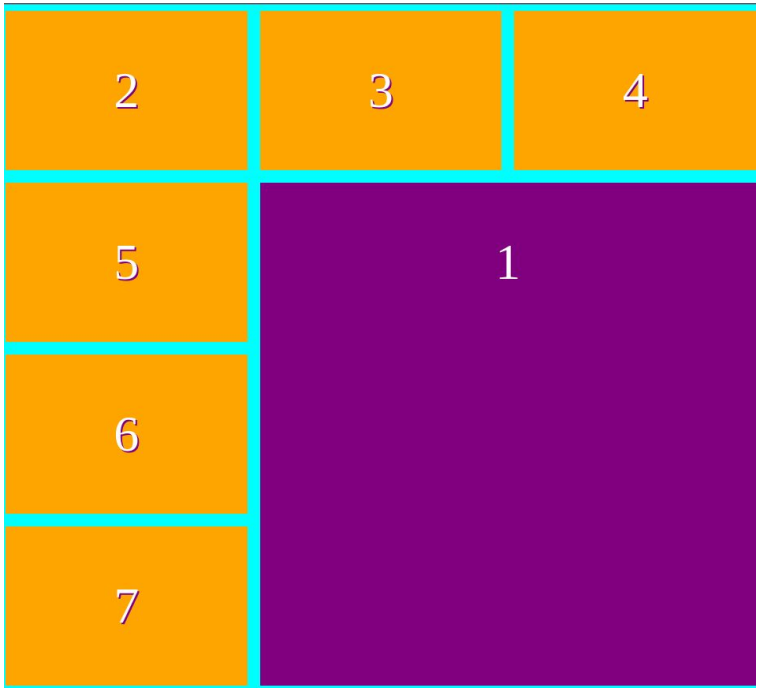


Positioning items

Positioning items with **span**:

Use *Span* keyword with in *grid-column* and *grid-row* to define the item position and size.

Positioning items



Filling empty cells automatically

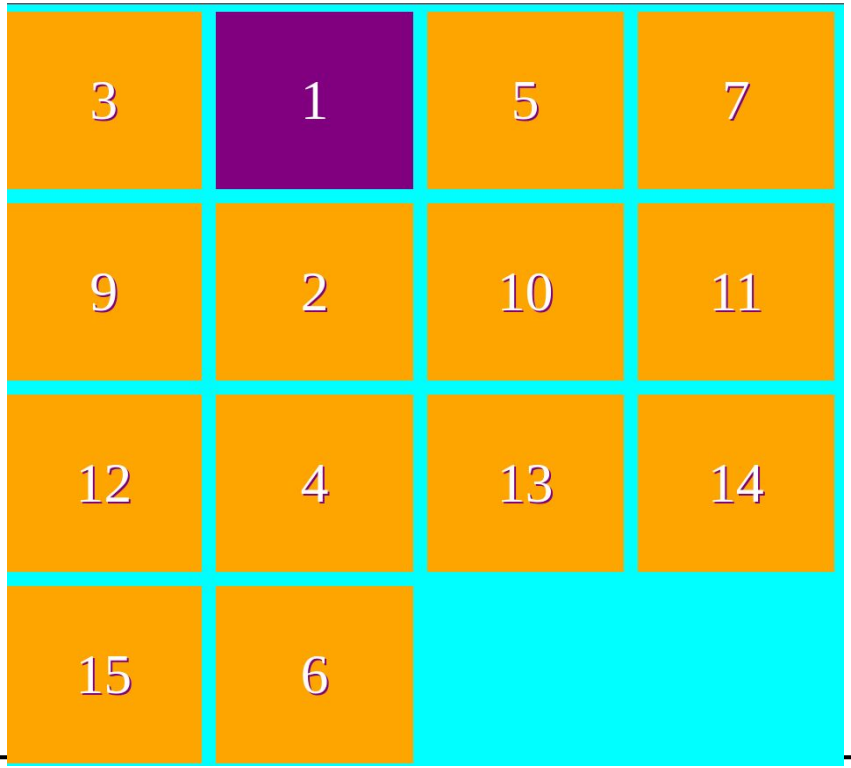
Auto-flow:

Once defining the grid for container. By default the items flow in a rows.

Using *grid-auto-flow* can control the flow of items.

Grid-auto-flow: row
Grid-auto-flow: column
Grid-auto-flow: dense
Grid-auto-flow: row dense
Grid-auto-flow: column dense

Filling empty cells automatically



Thanks