



# LAYOUTING

MIT FLEXBOX & CSS GRID

VON TIMO KORINTH

---



# HI! ICH BIN TIMO KORINTH

**job.** UX Entwickler, Trainer & Autor **personal.** Verheiratet, Vater &  
BVB Fan **hobbies.** Nerd Zeug **where.** Dortmund, Deutschland **social.**



@timokorinth



timo-korinth



timokorinth.de

CENTER



# DER HEILIGE GRAL

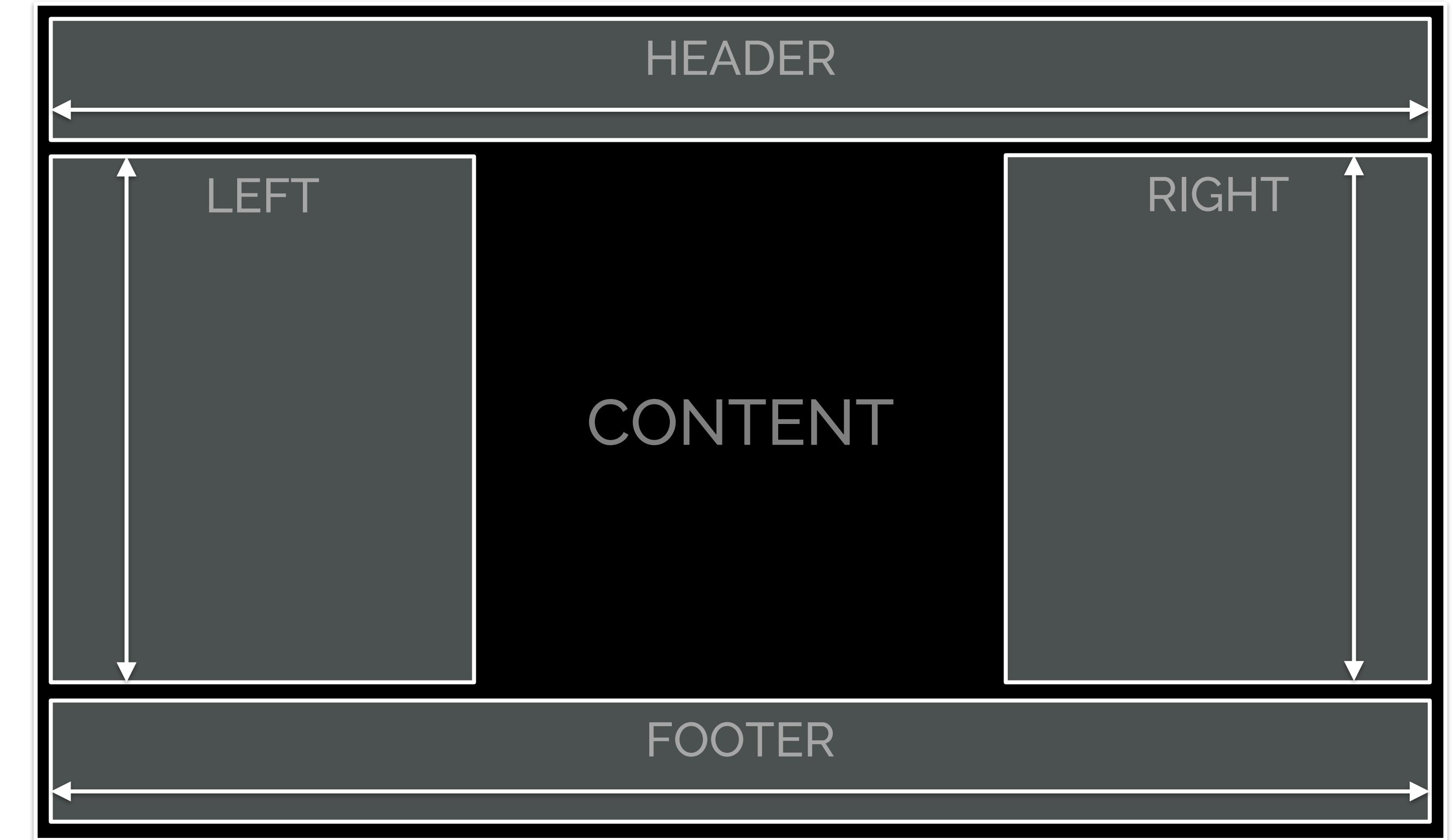
Der Webentwicklung

**VOLLE BREITE**

**GLEICH HOCH**

**CODE REIHENFOLGE**

**SO WENIG CODE WIE MÖGLICH**



# APPLIKATIONS-LAYOUT

Meist genutzt:

**BLOCK & FLOAT**

**ABSOLUTE POSITIONIERUNG**

**FESTE GRÖßen**

**JAVASCRIPT / JQUERY**

**CSS HACKS / TRICKS**



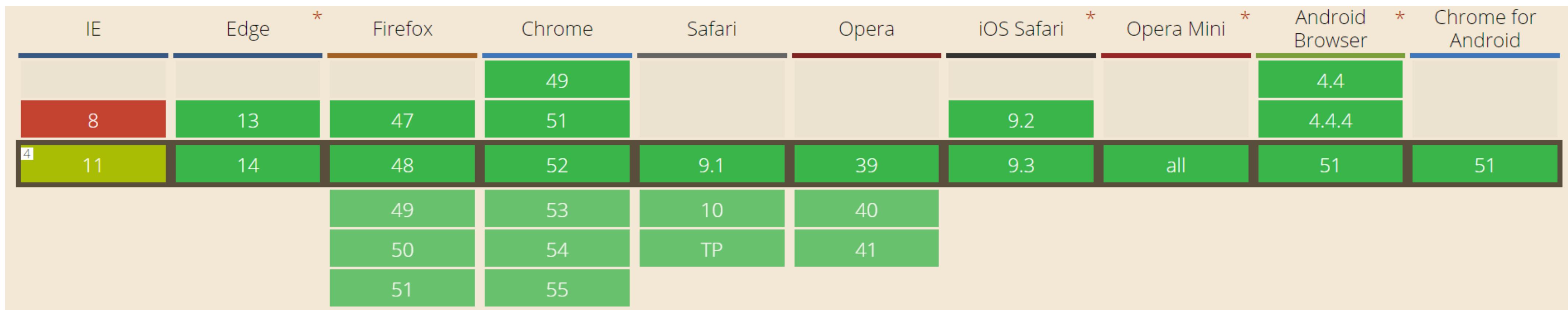
**FLEXBOX**

# FLEXBOX

The Flexbox Layout (Flexible Box) **module** (currently a W3C Last Call Working Draft) aims at providing a more **efficient** way to lay out, **align and distribute** space among **items in a container**, even when their **size is unknown** and/or **dynamic** (thus the word "flex").

# FLEXBOX

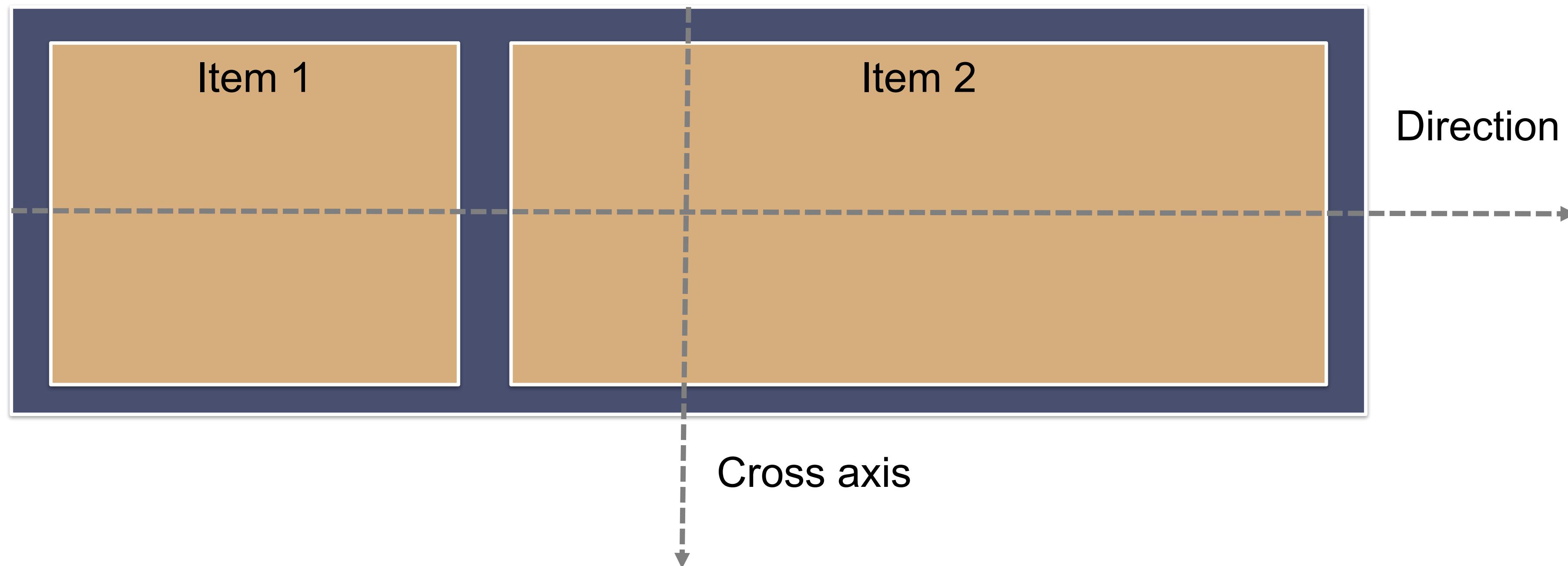
## Browser Support



<http://caniuse.com/#feat=flexbox>

# FLEXBOX

Container



# FLEXBOX

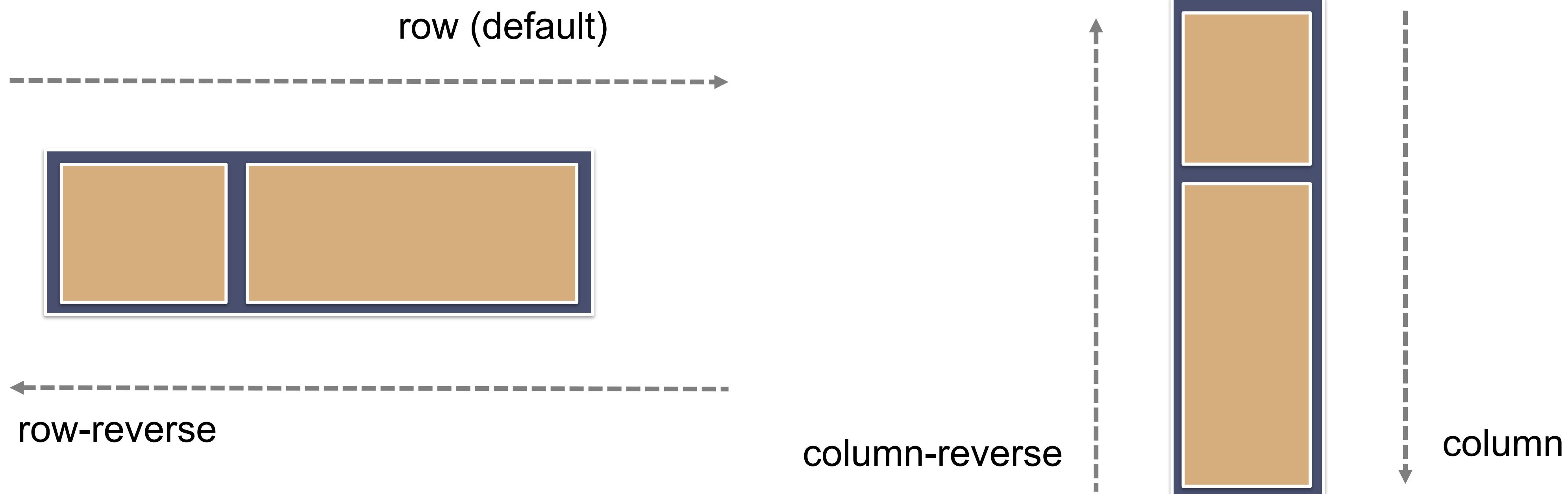
**display: flex;**

Container



# FLEXBOX

**flex-direction:** row | row-reverse | column | column-reverse;

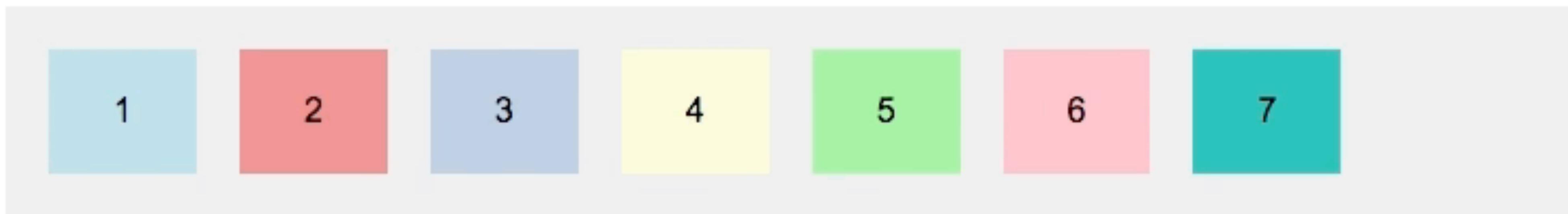


# FLEXBOX

**flex-grow: <number>; //default: 0**

# FLEXBOX

**flex-grow: 0;**



# FLEXBOX

**flex-grow: 1;**



# FLEXBOX

```
flex-grow: 1;  
.item3 { flex-grow: 2; }
```



# FLEXBOX

```
flex-grow: 0;  
.item3 { flex-grow: 1; }
```



# FLEXBOX

**flex-shrink: <number>; //default: 1**

# FLEXBOX

**flex-shrink: 1;**



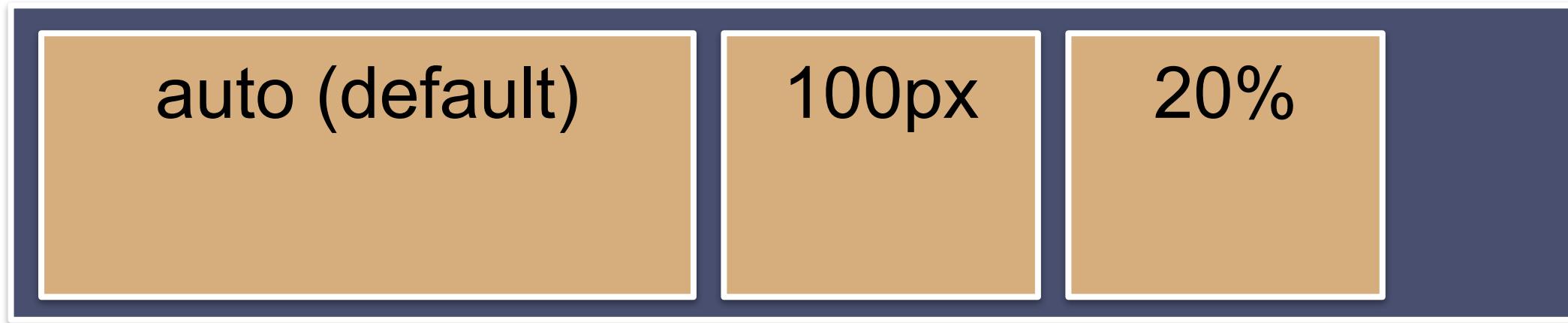
# FLEXBOX

```
flex-shrink: 1;  
.item3 { flex-shrink: 0; }
```



# FLEXBOX

**flex-basis: <length> | auto;**



# FLEXBOX

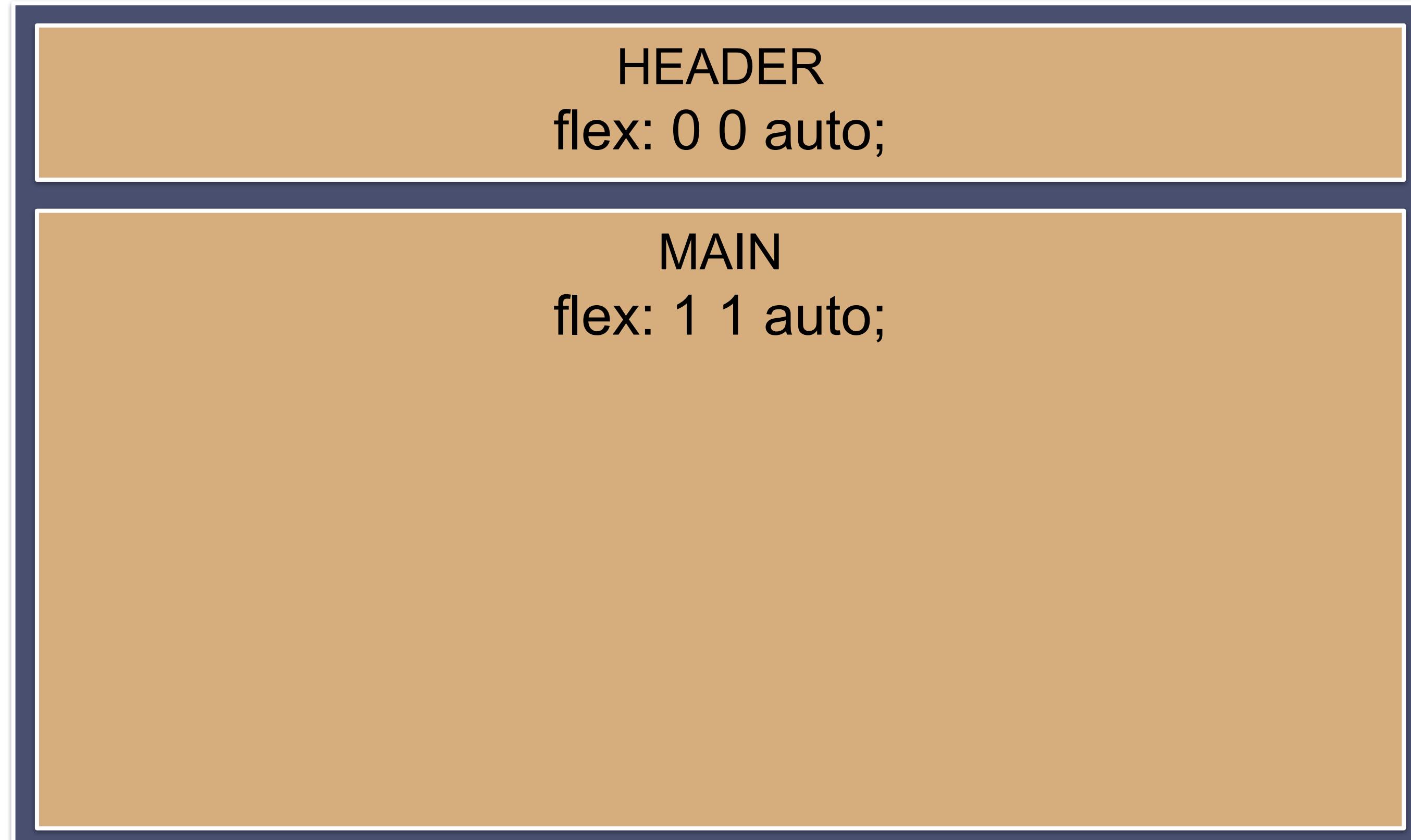
Kurzschreibweise:

**flex: <grow> <shrink> <basis>;**

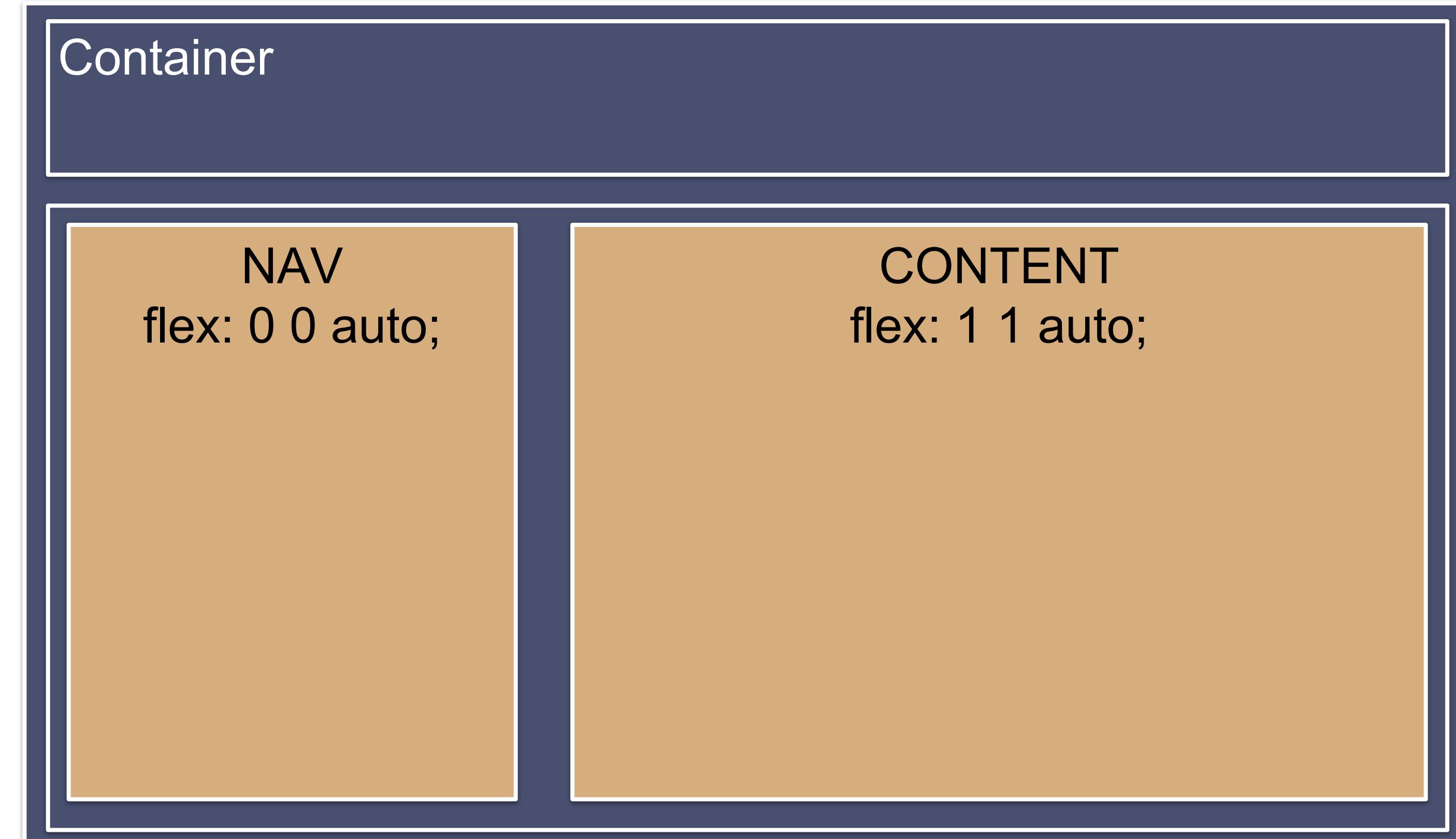
# FLEXBOX

Container

`flex-direction: column;`



# FLEXBOX



# BROWSER PREFIXE

**display: flex;**

**display: -webkit-box;**

**display: -webkit-flex;**

**display: -ms-flexbox;**

Ist eigentlich ...

# AUTOPREFIXER

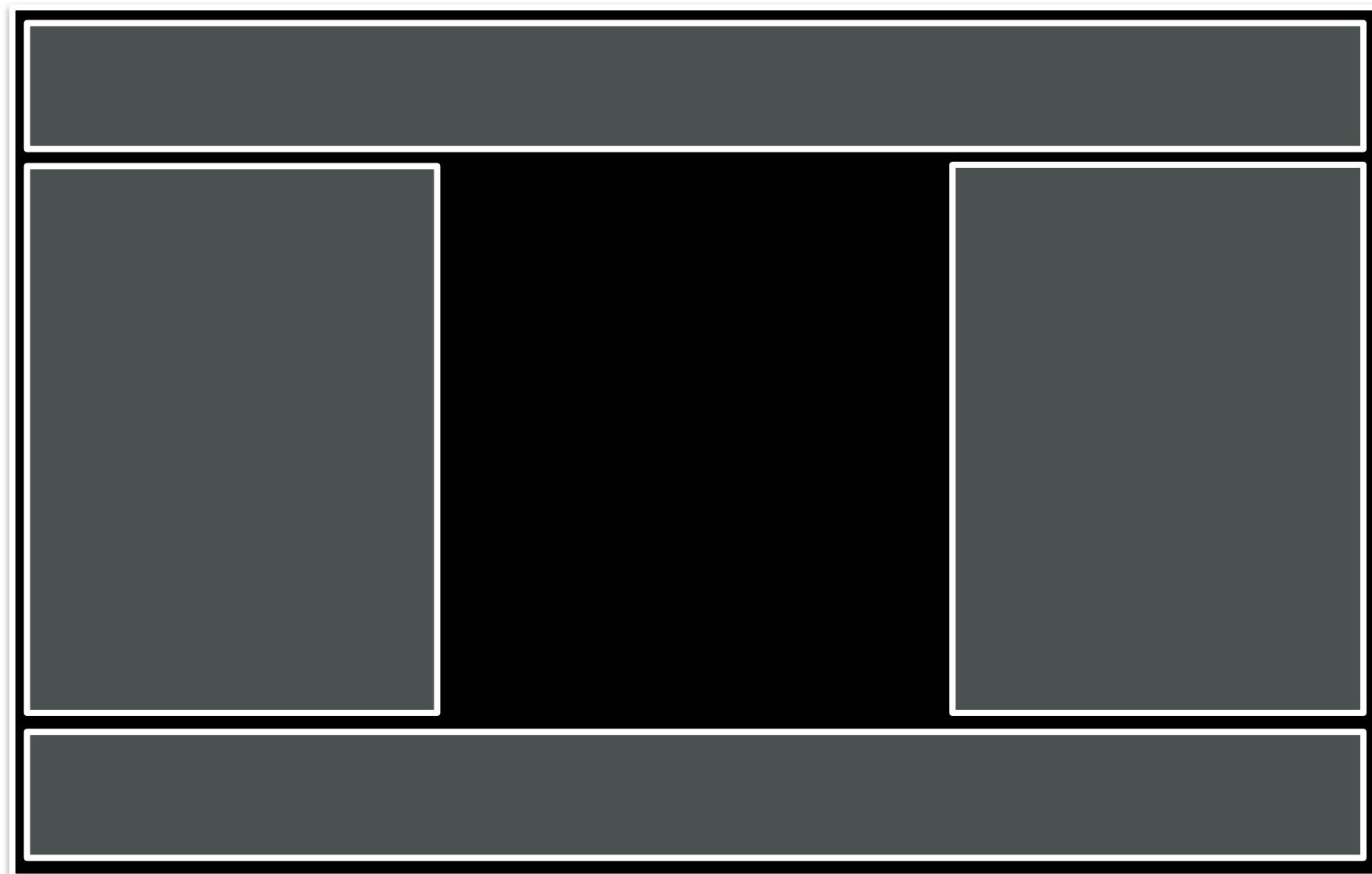
```
.flexDisplay {  
    display: flex;  
    display: -webkit-box;  
    display: -webkit-flex;  
    display: -ms-flexbox;  
}
```



<https://github.com/postcss/autoprefixer>

# FLEXBOX

Der heilige Gral

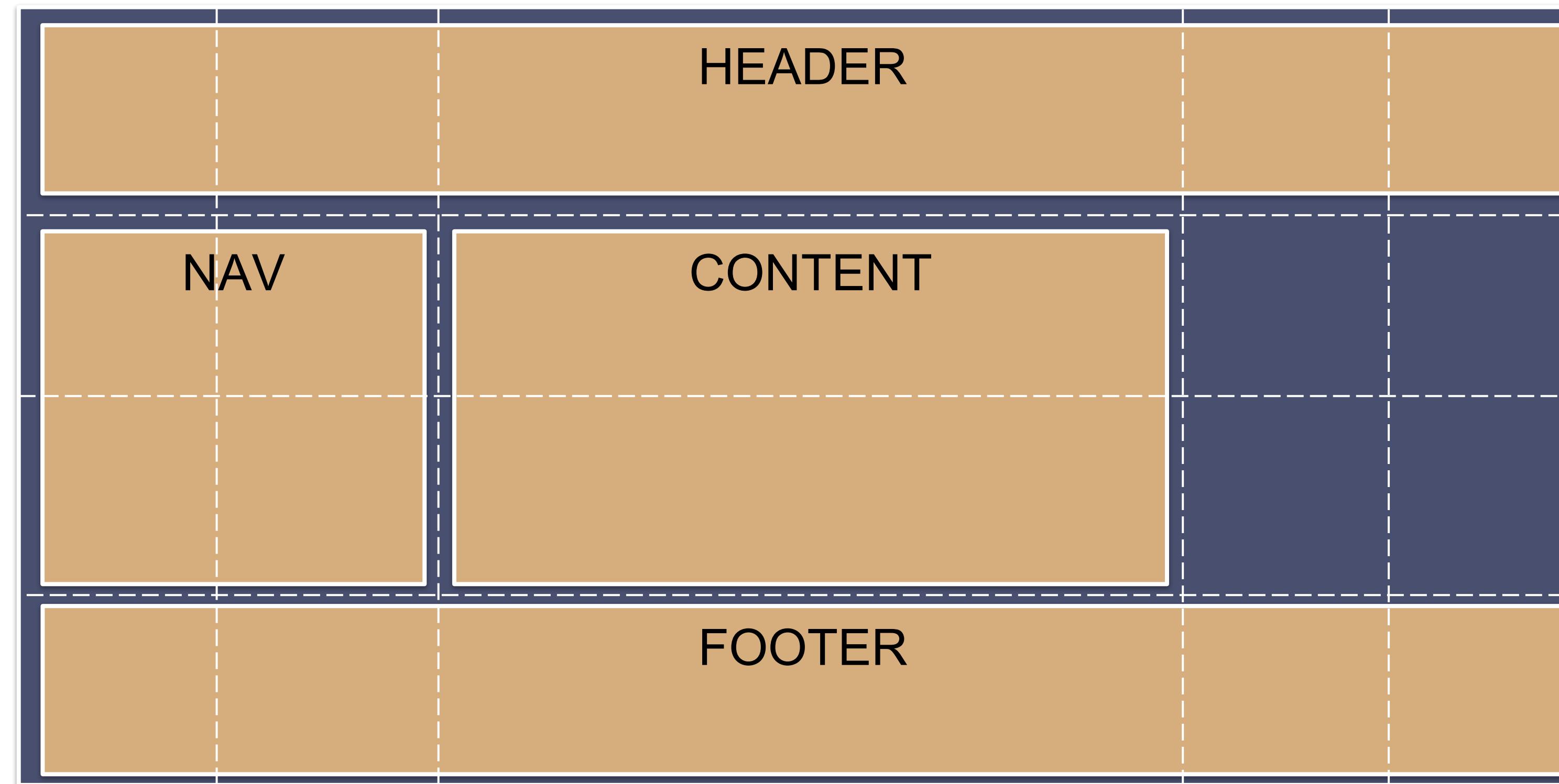


DEMO



**CSS GRID**

# GRID LAYOUT

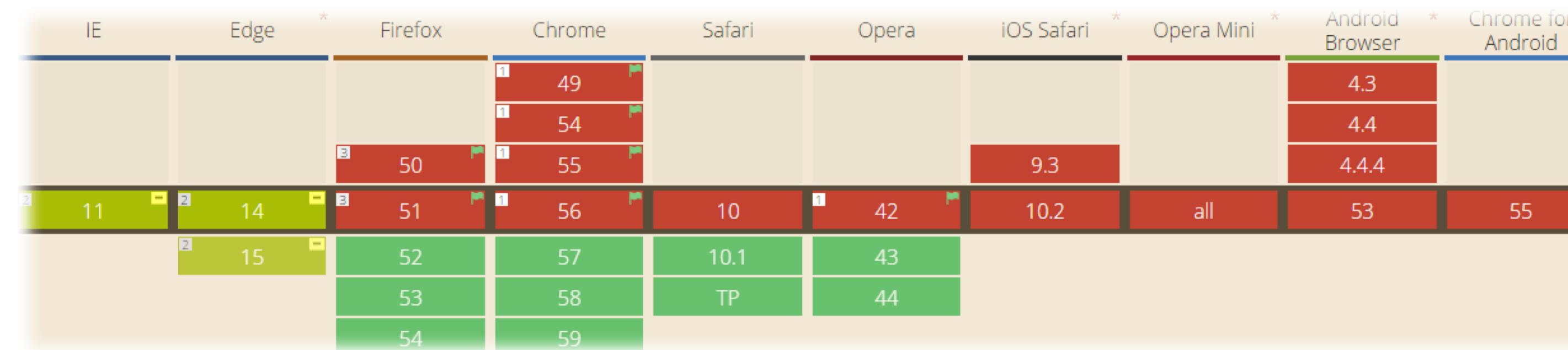
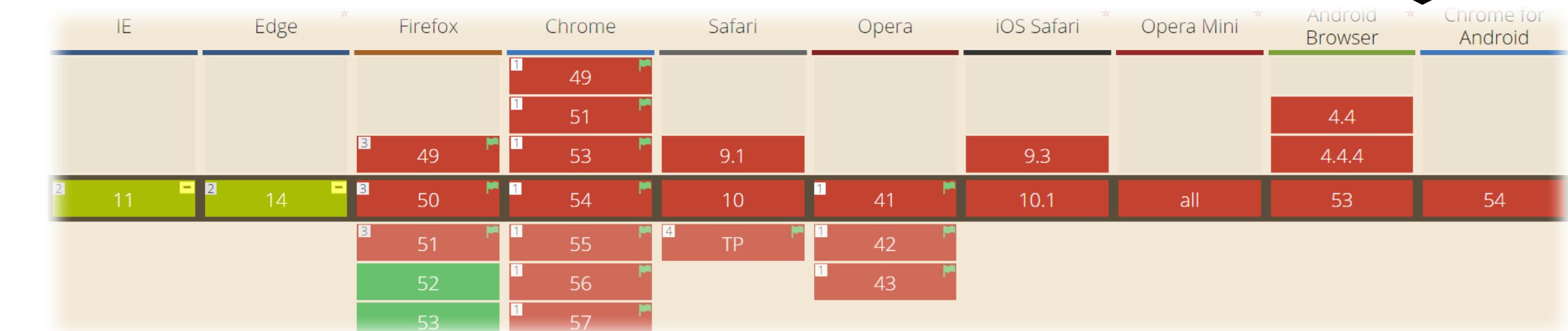


# GRID LAYOUT

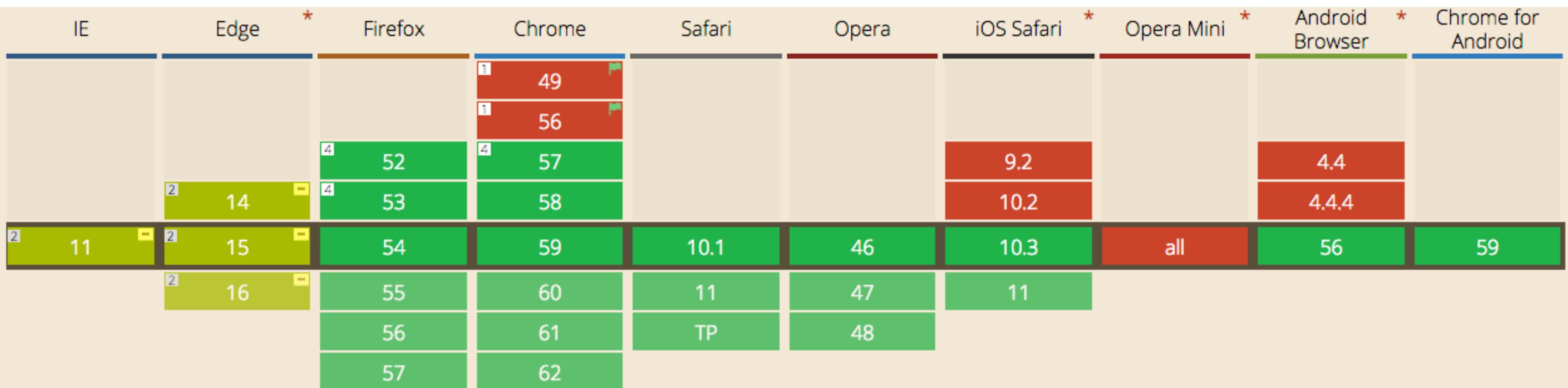
An important thing to understand about Grid is that it's not ready to be used in production yet.

It's currently a W3C Working Draft and isn't supported correctly in any browsers by default. Internet Explorer 10 and 11 support it, but it's an old implementation with outdated syntax.

# Browser Support



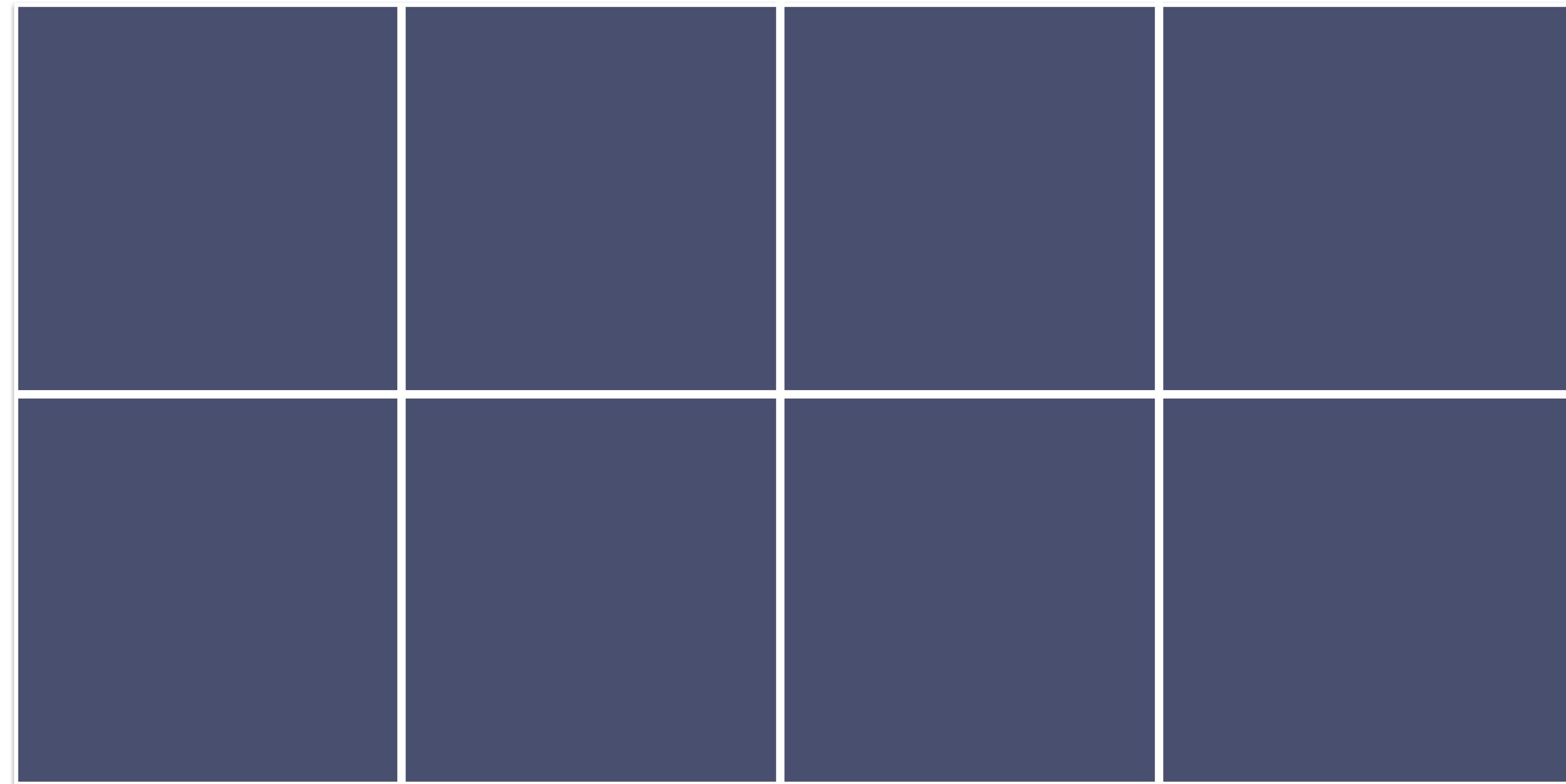
<http://caniuse.com/#feat=css-grid>



# GRID LAYOUT

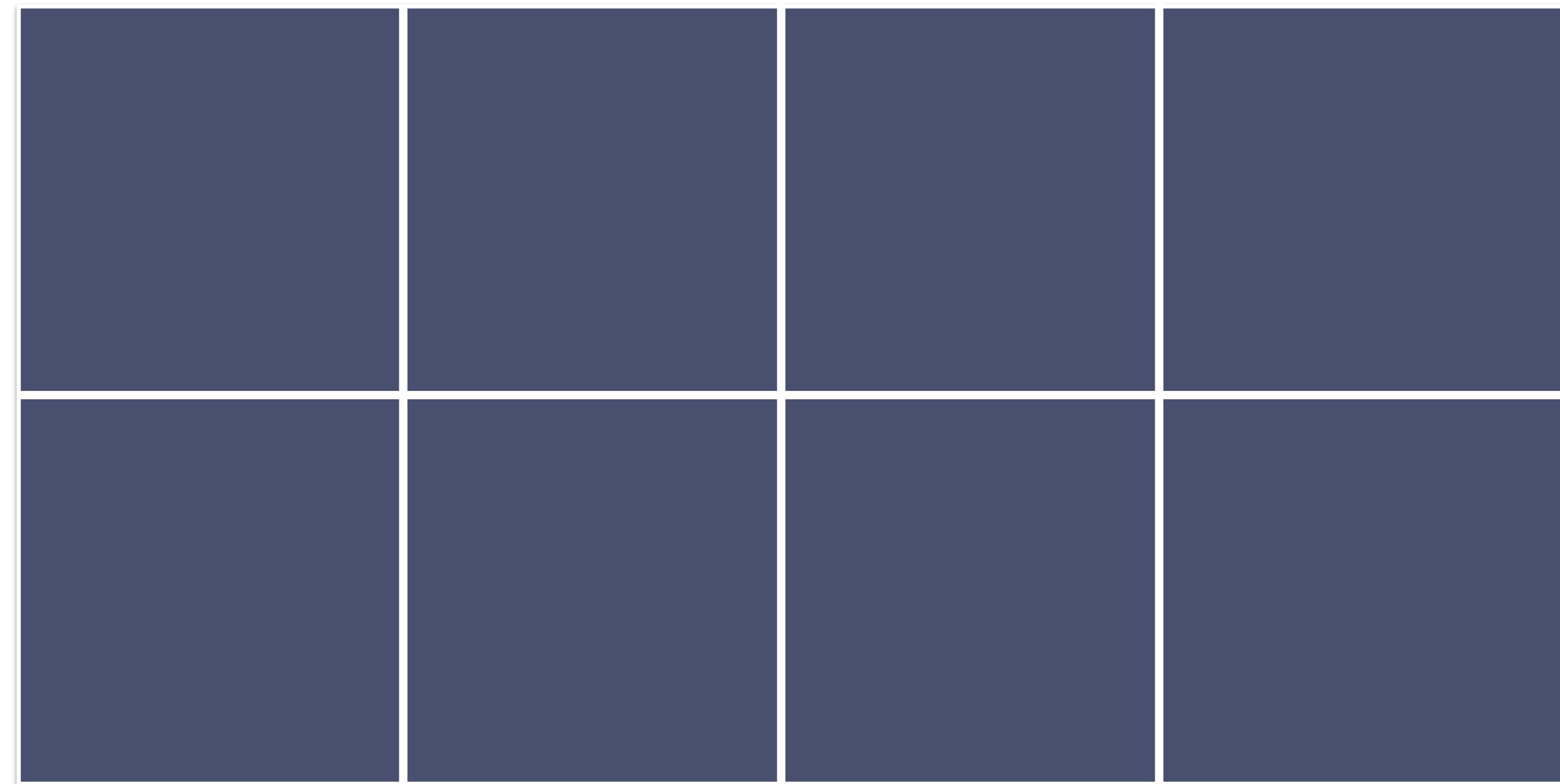
Grid Container

`display: grid;`



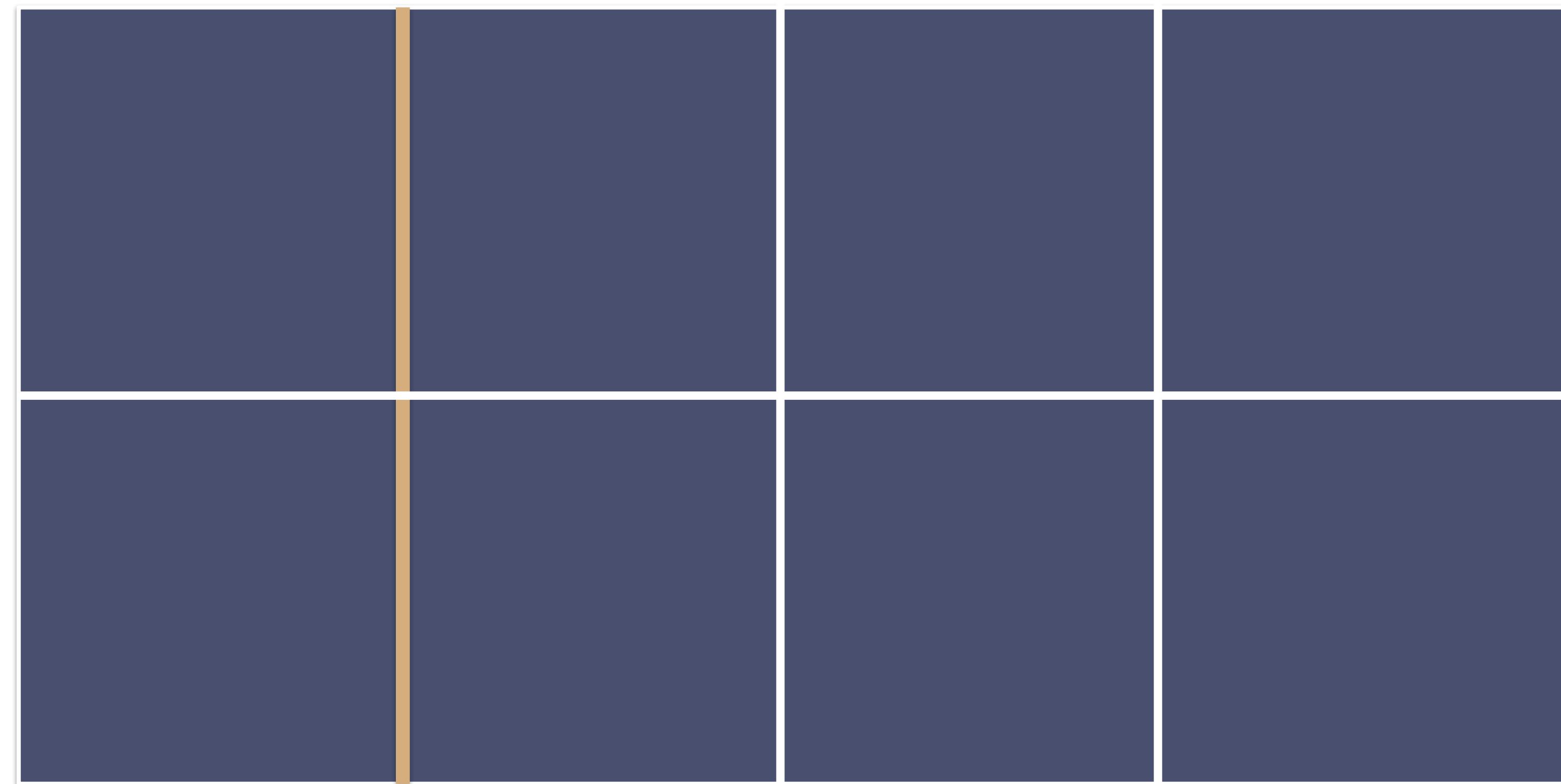
# GRID LAYOUT

Grid Items: Nur direkte Kinder des Conainers



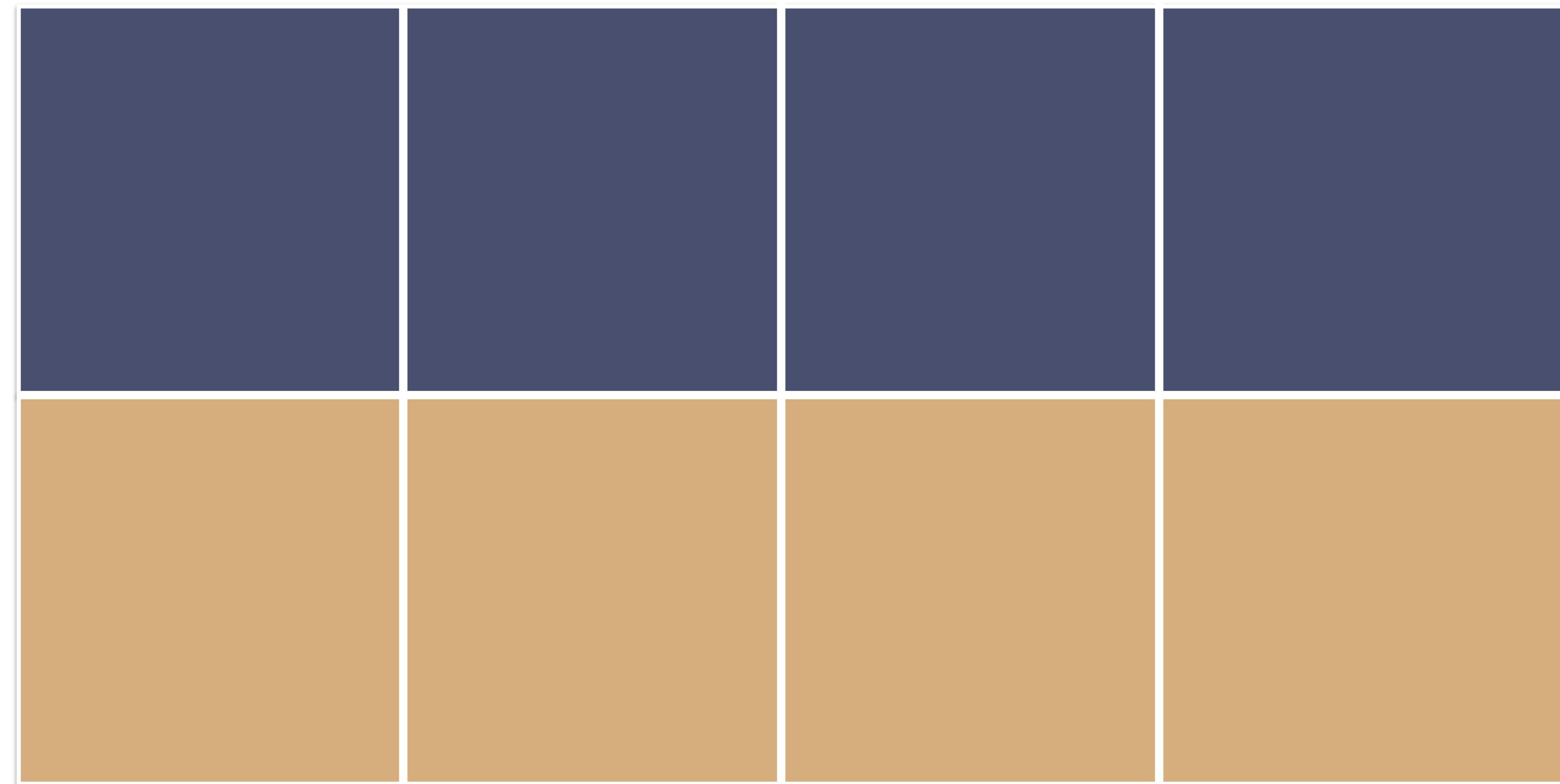
# GRID LAYOUT

Grid Line



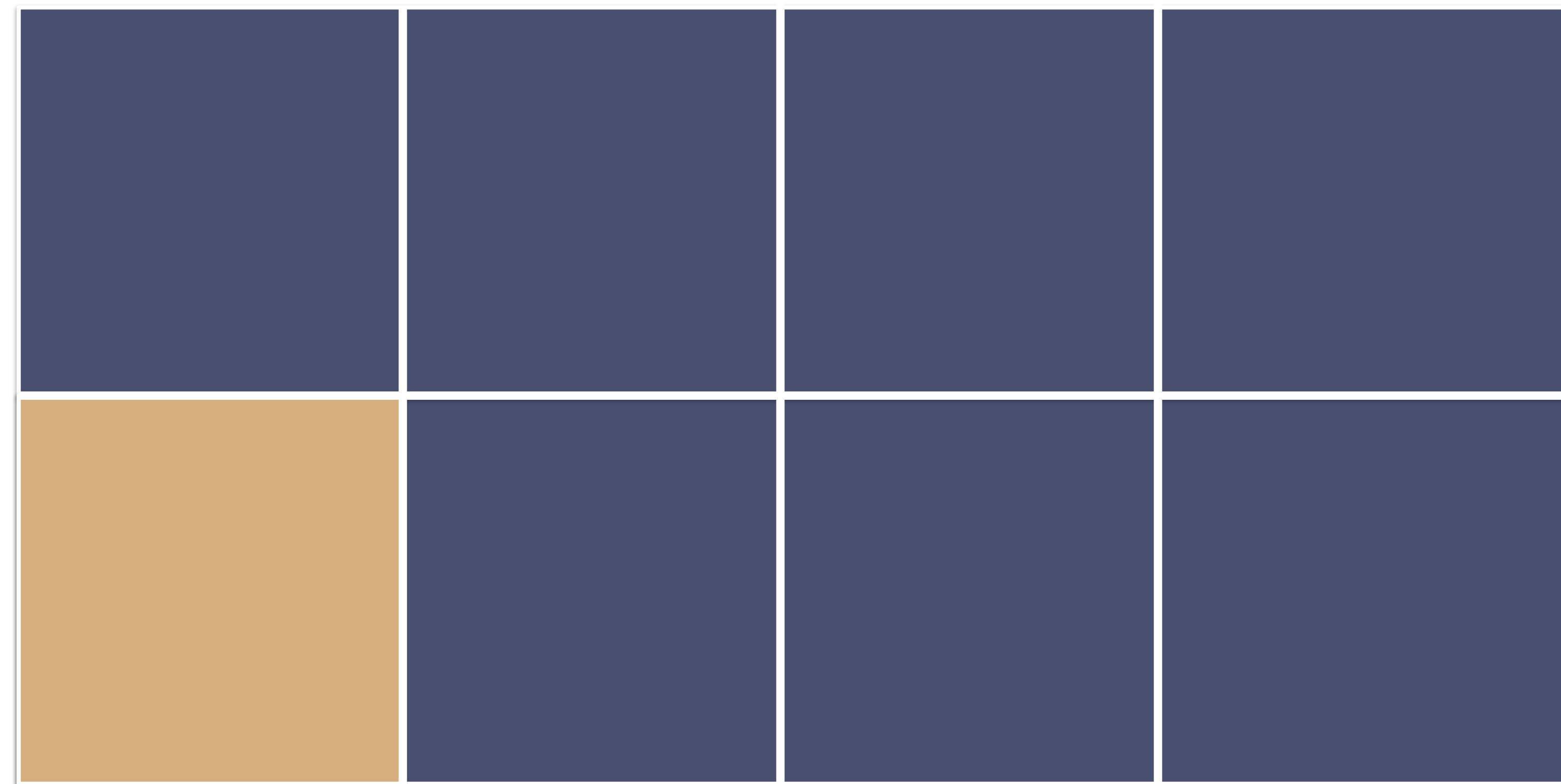
# GRID LAYOUT

Grid Track



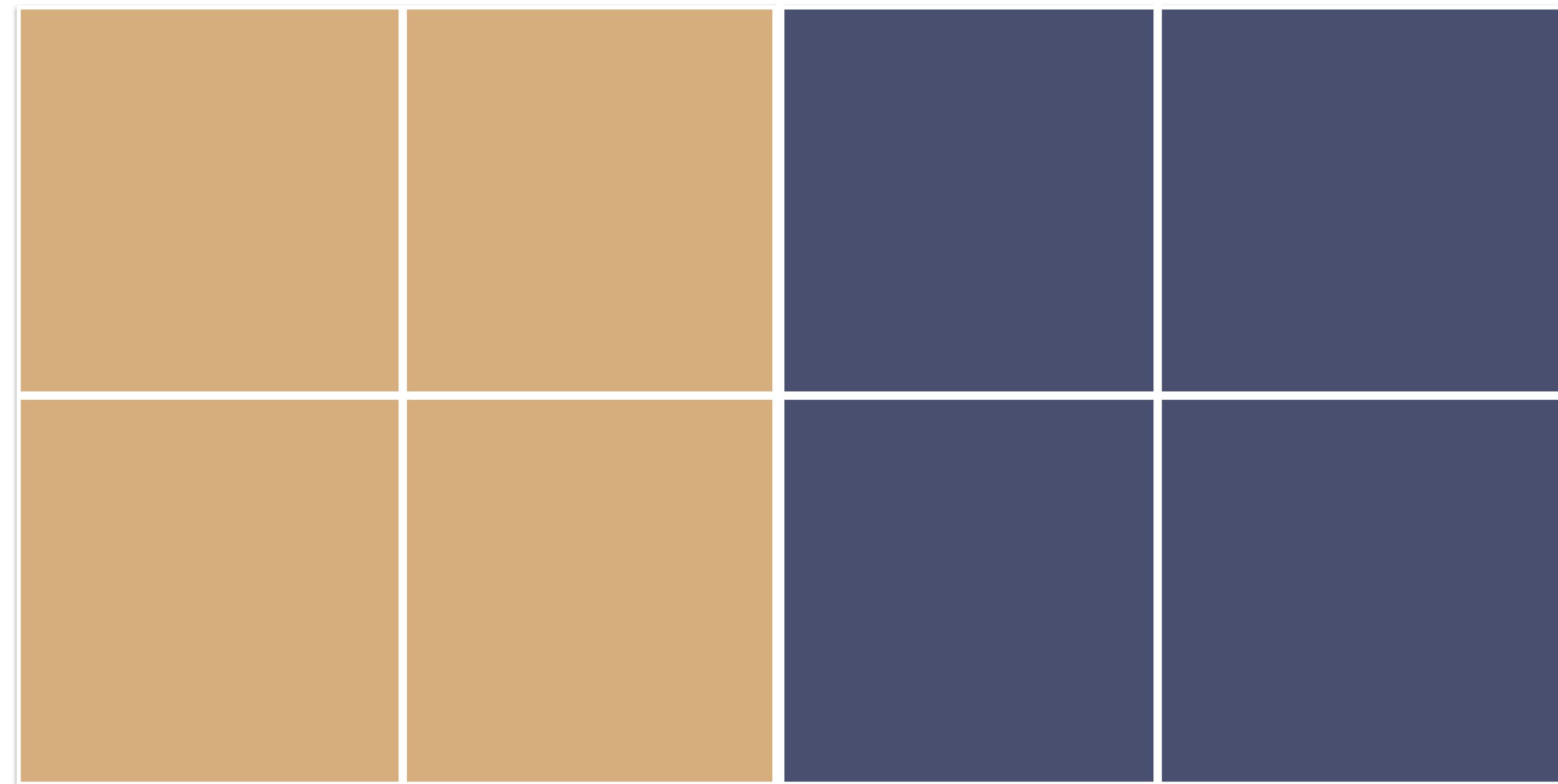
# GRID LAYOUT

Grid Cell



# GRID LAYOUT

Grid Area

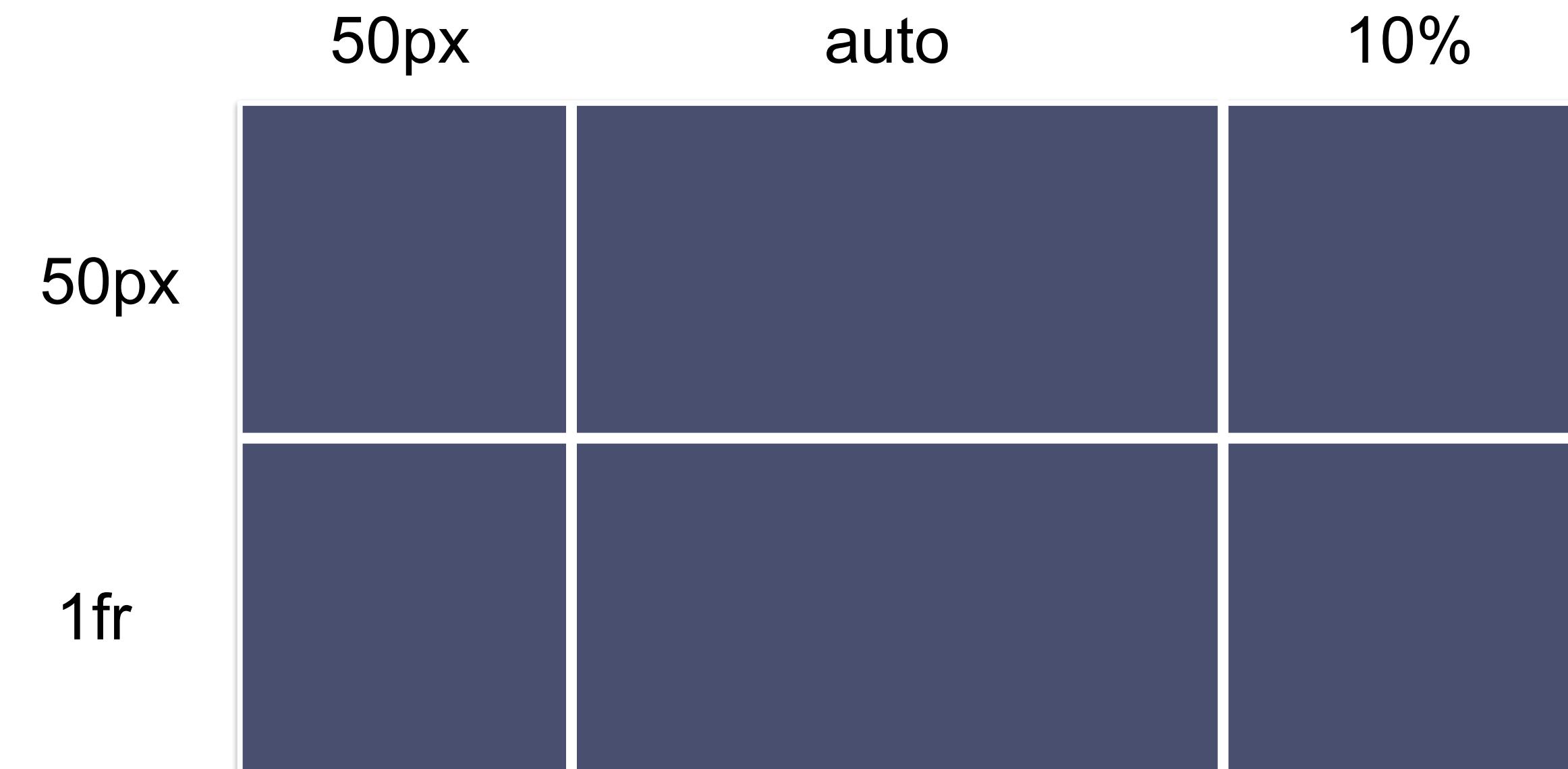


# GRID CONTAINER

**grid-template-columns:** <track-size> ... | <line-name> <track-size> ...;

**grid-template-rows:** <track-size> ... | <line-name> <track-size> ...;

Shorthand: **grid-template**

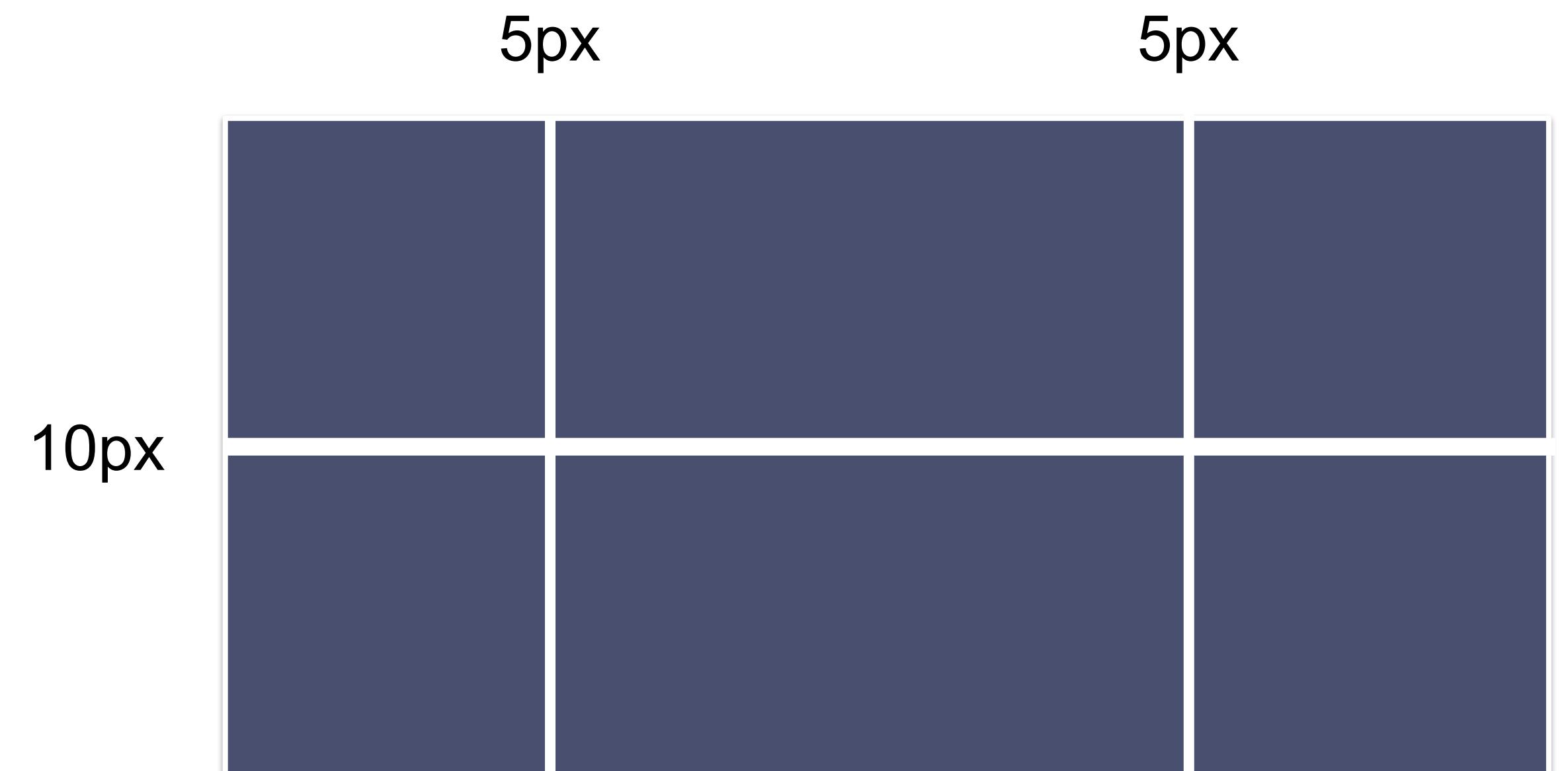


# GRID CONTAINER

**grid-column-gap: <line-size>;**

**grid-row-gap: <line-size>;**

Shorthand: **grid-gap**

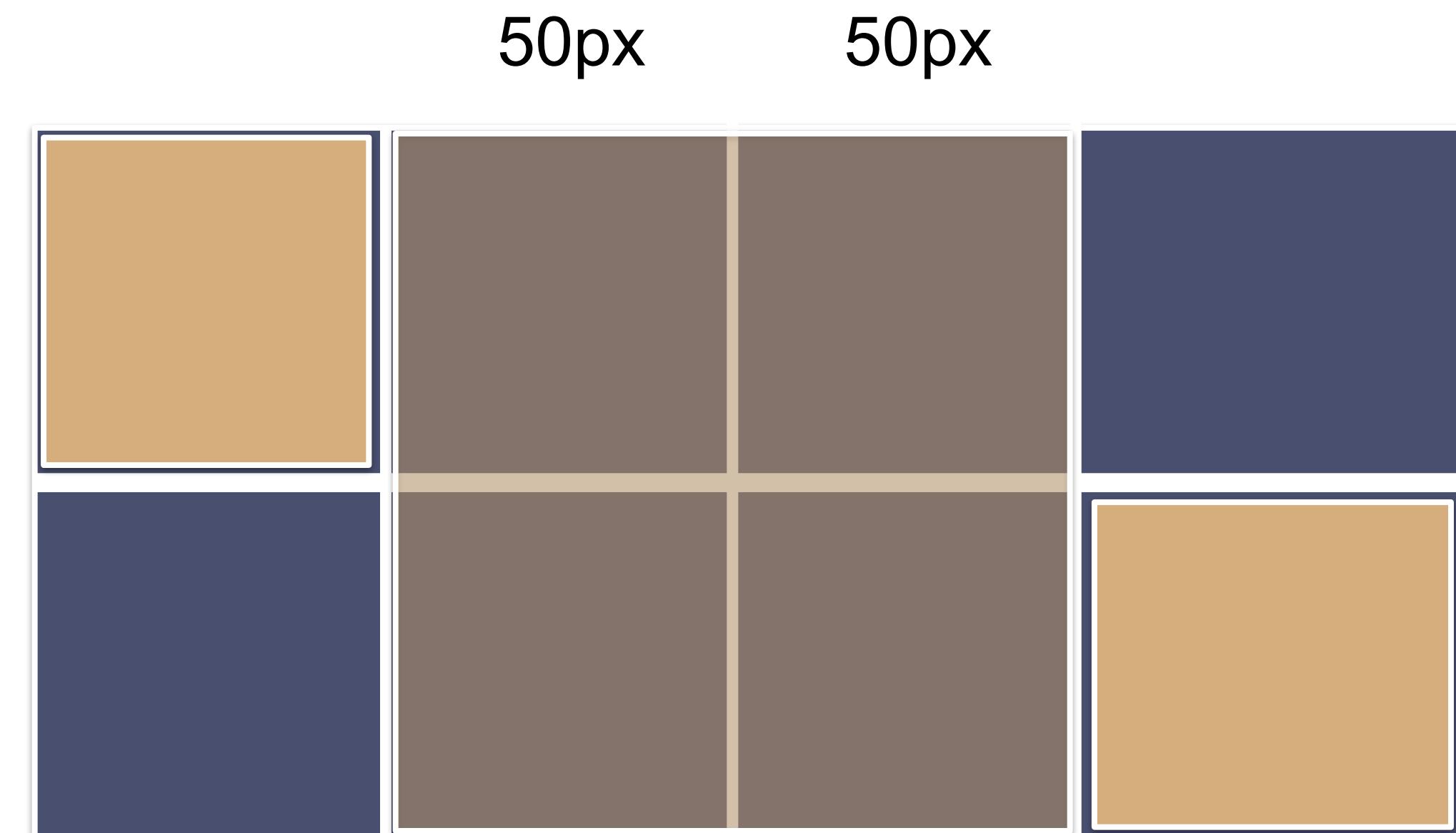


Nicht am Rand!

# GRID CONTAINER

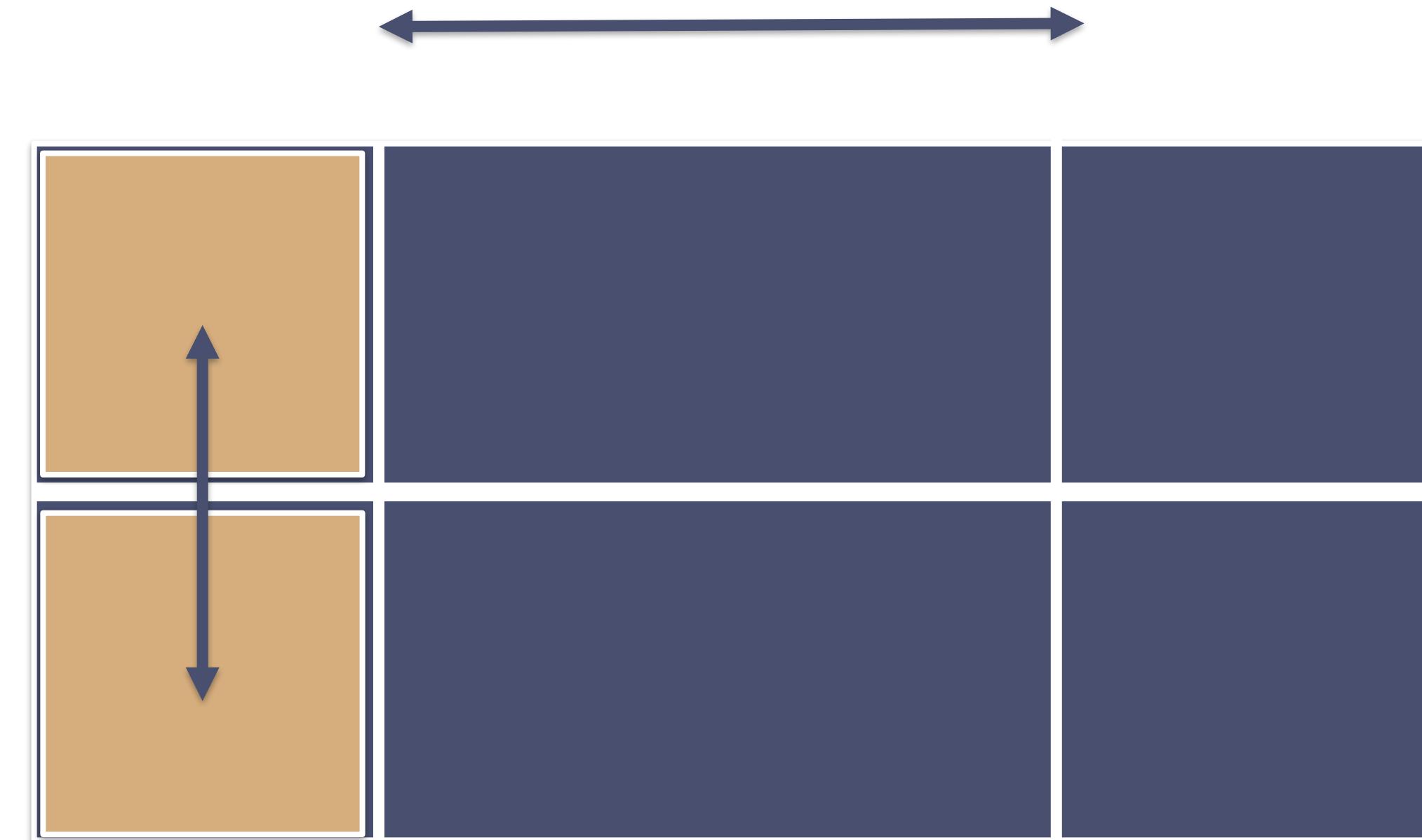
**grid-auto-columns: <track-size> ...;**

**grid-auto-rows: <track-size> ...;**



# GRID CONTAINER

**grid-auto-flow:** row | column | row dense | column dense;



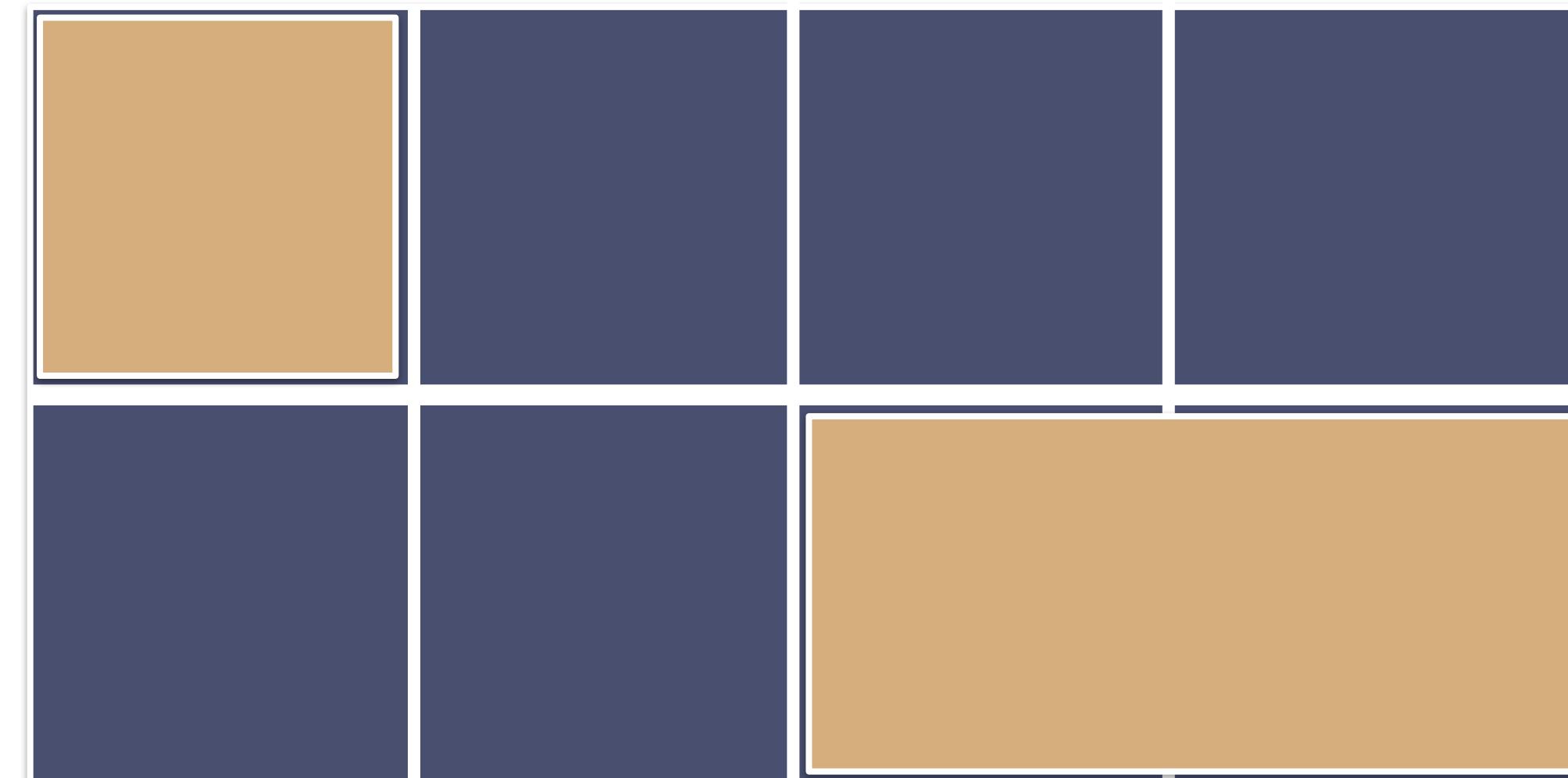
# GRID ITEMS

**grid-column-start, grid-column-end: <number> | span <number>;**

**grid-row-start, grid-row-end: <number> | span <number>;**

Shorthand: **grid-column, grid-row**

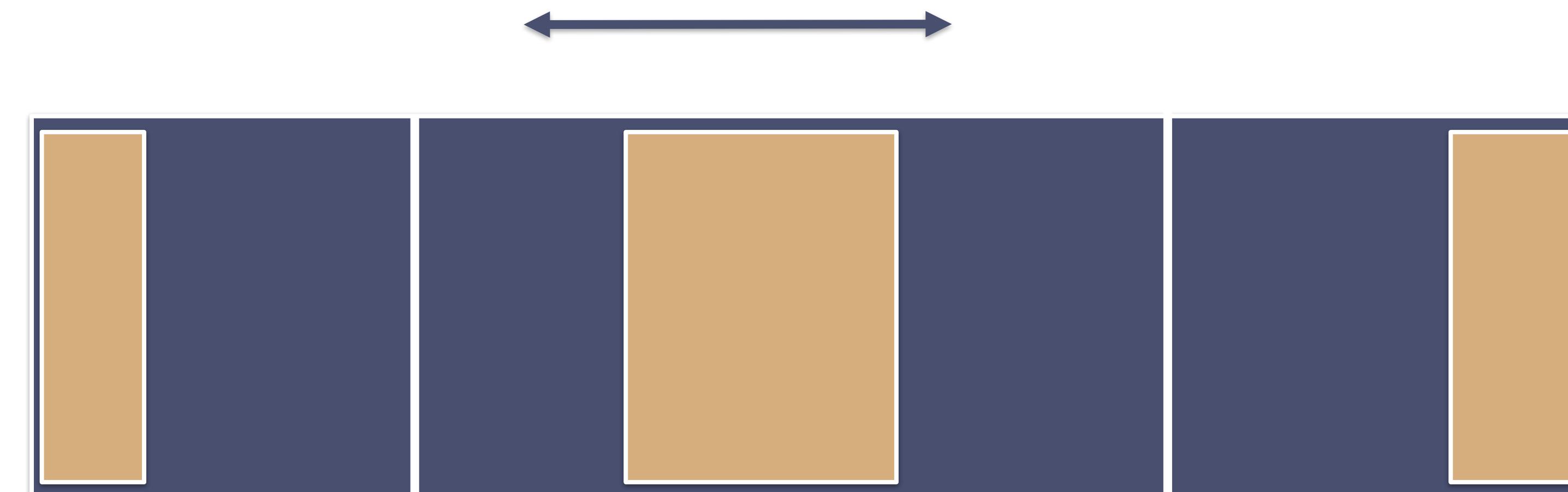
```
grid-column-start: 1;  
grid-column-end: 2;  
grid-row-start: 1;  
grid-row-end: 2;
```



```
grid-column-start: 3;  
grid-column-end: span 2;  
grid-row-start: 2;  
grid-row-end: span 1;
```

# GRID ITEMS

**justify-self:** start | end | center | stretch;



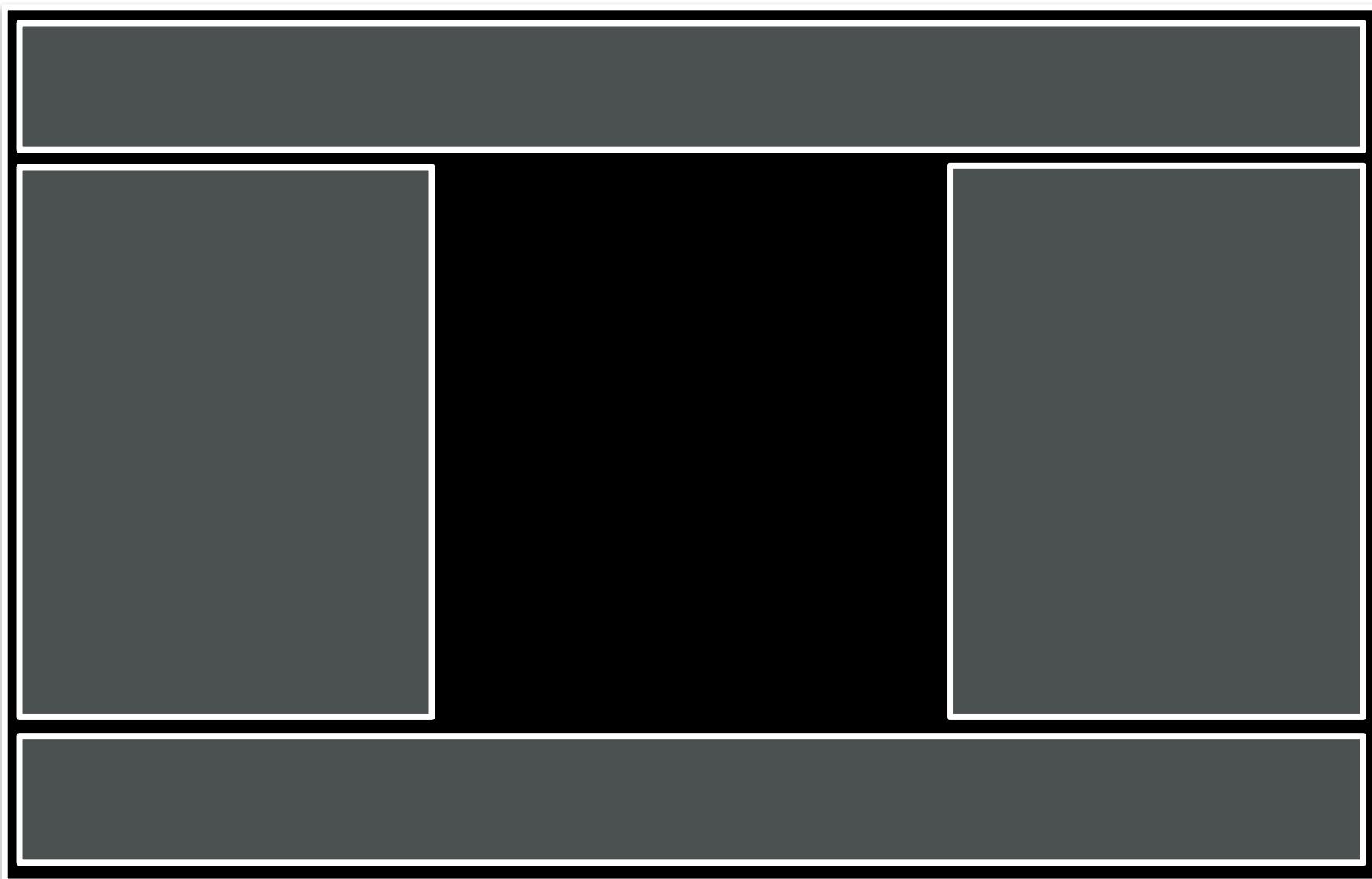
# GRID ITEMS

**align-self: start | end | center | stretch;**



# CSS GRID

Der heilige Gral



DEMO

# FLEXBOX VS. GRID

## Flexbox

1 Dimension

Alien Syntax

Browser Support -> Gut

Community -> Alt und Weise

...

## Grid

2 Dimensionen

Intuitives Grid Modell

Browser Support -> Läuft.

Community -> Frisch

...