

Building Complex Layouts with CSS Grid

“Intrinsic Web Design”

Jen Simmons



moz://a

Why not just use flexbox?

“Flexbox is great for taking a bunch of oddly-sized things and returning the most reasonable layout for those things”

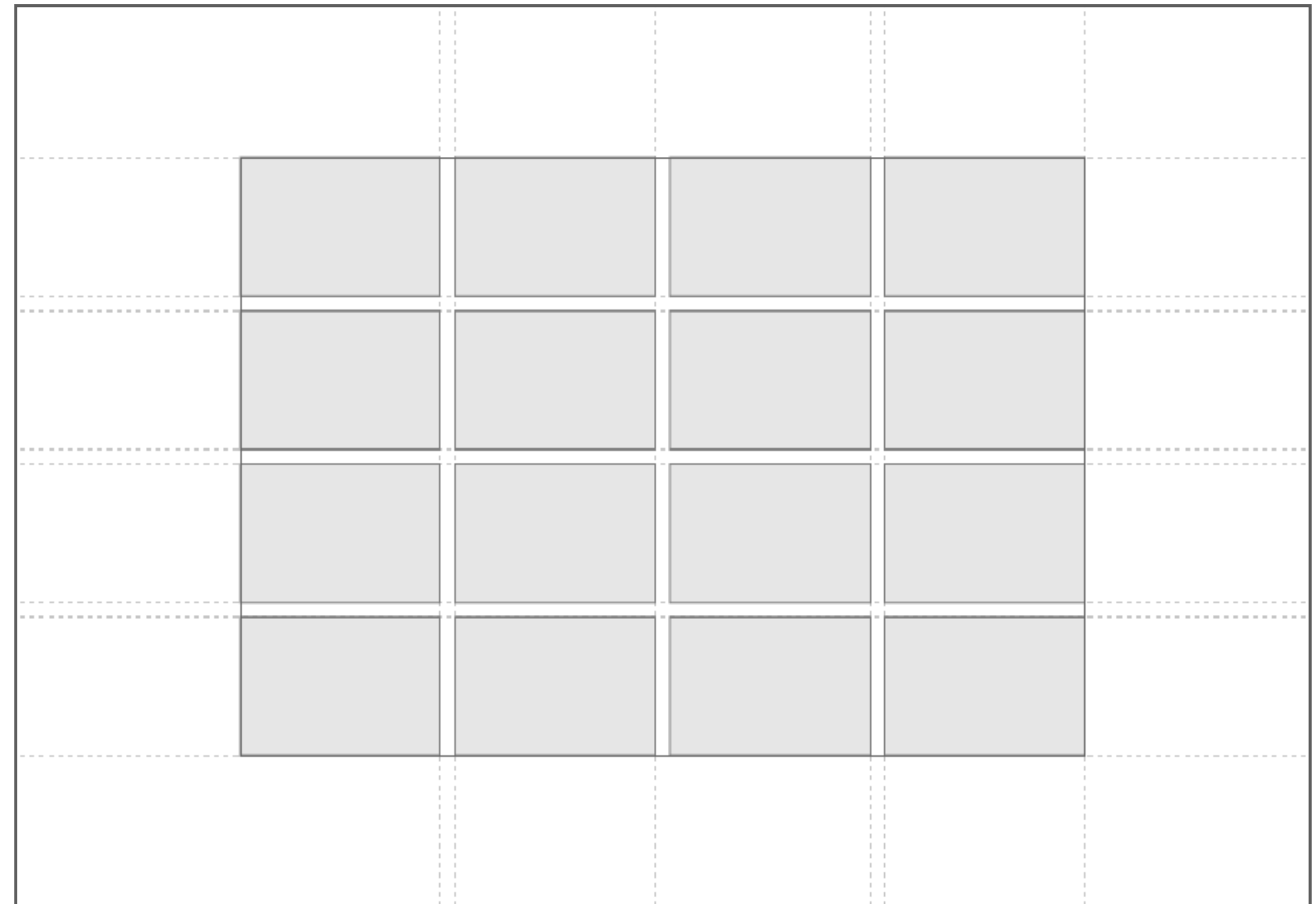
Rachel Andrew

Complete control over 2D layout

Grid terminology

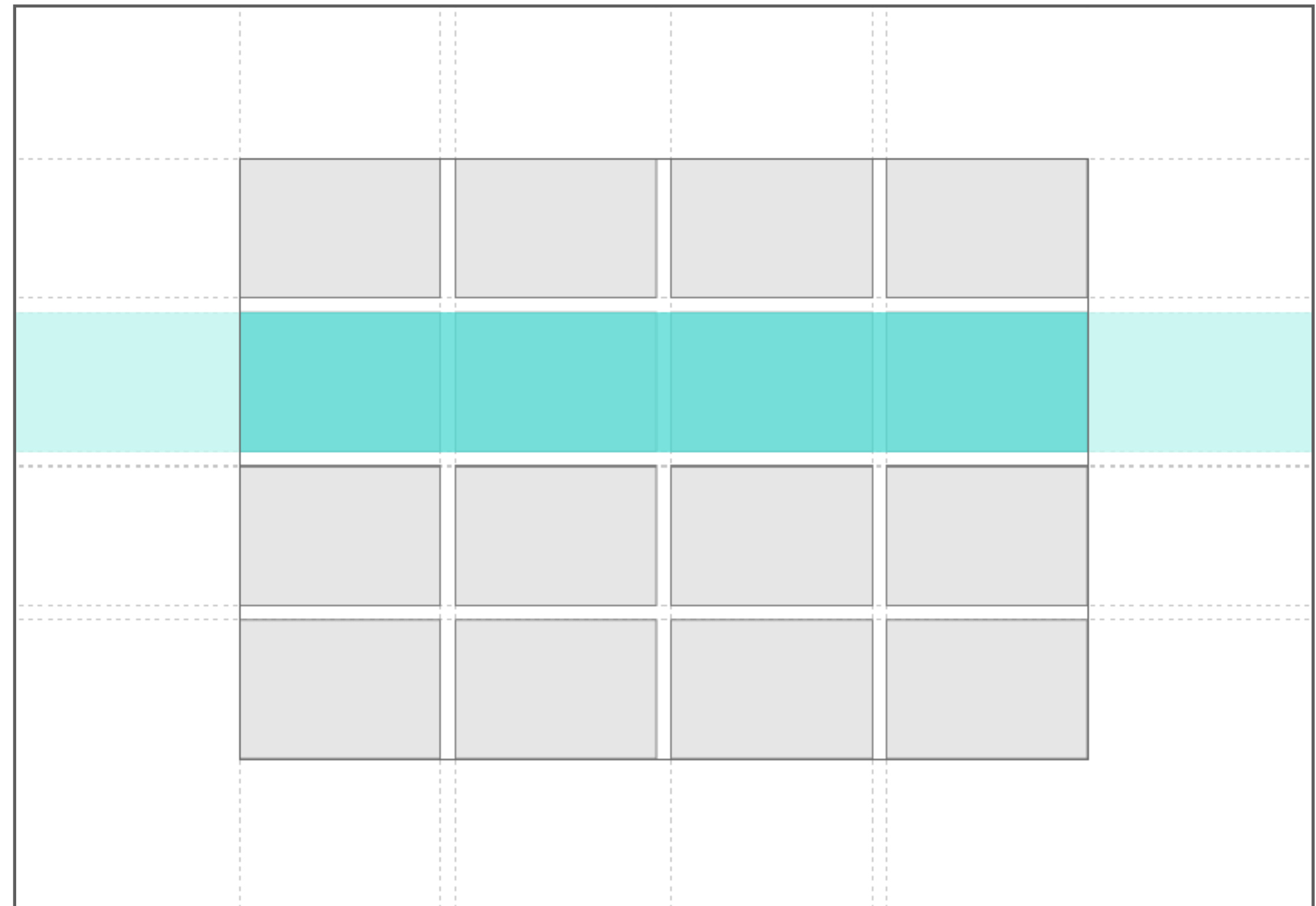
Grid

```
display: grid;
```



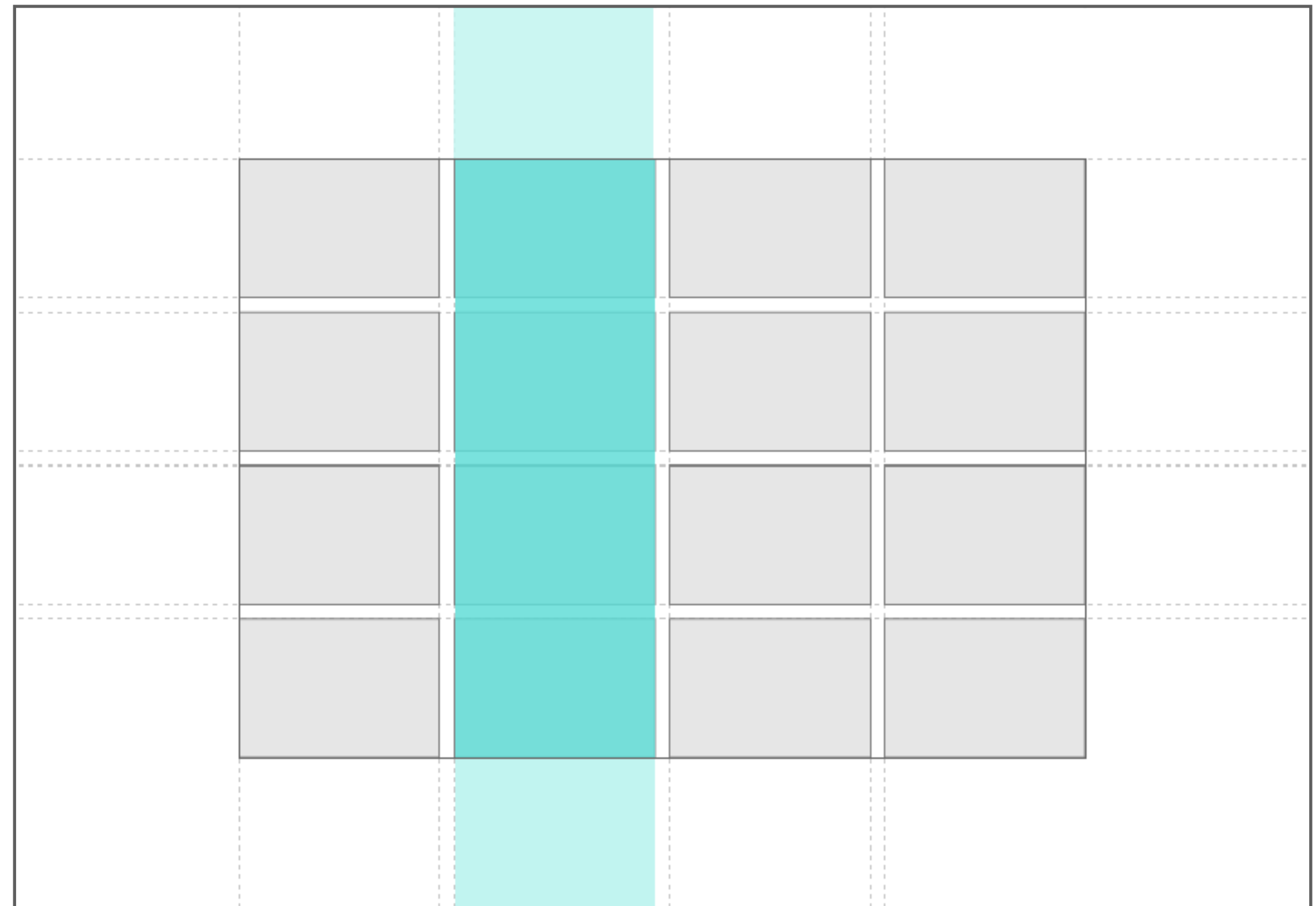
Tracks

```
grid-template-rows
```

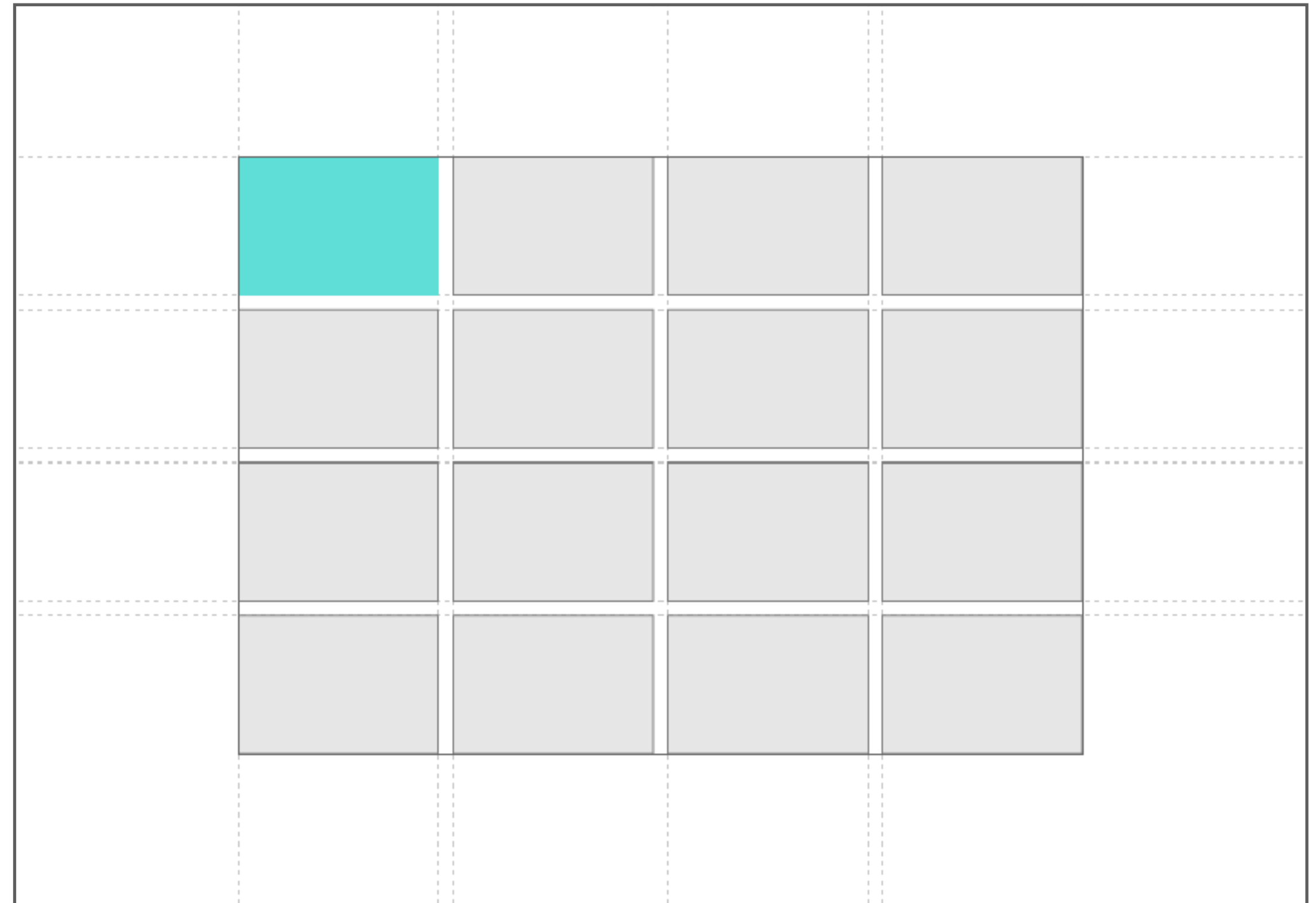


Tracks

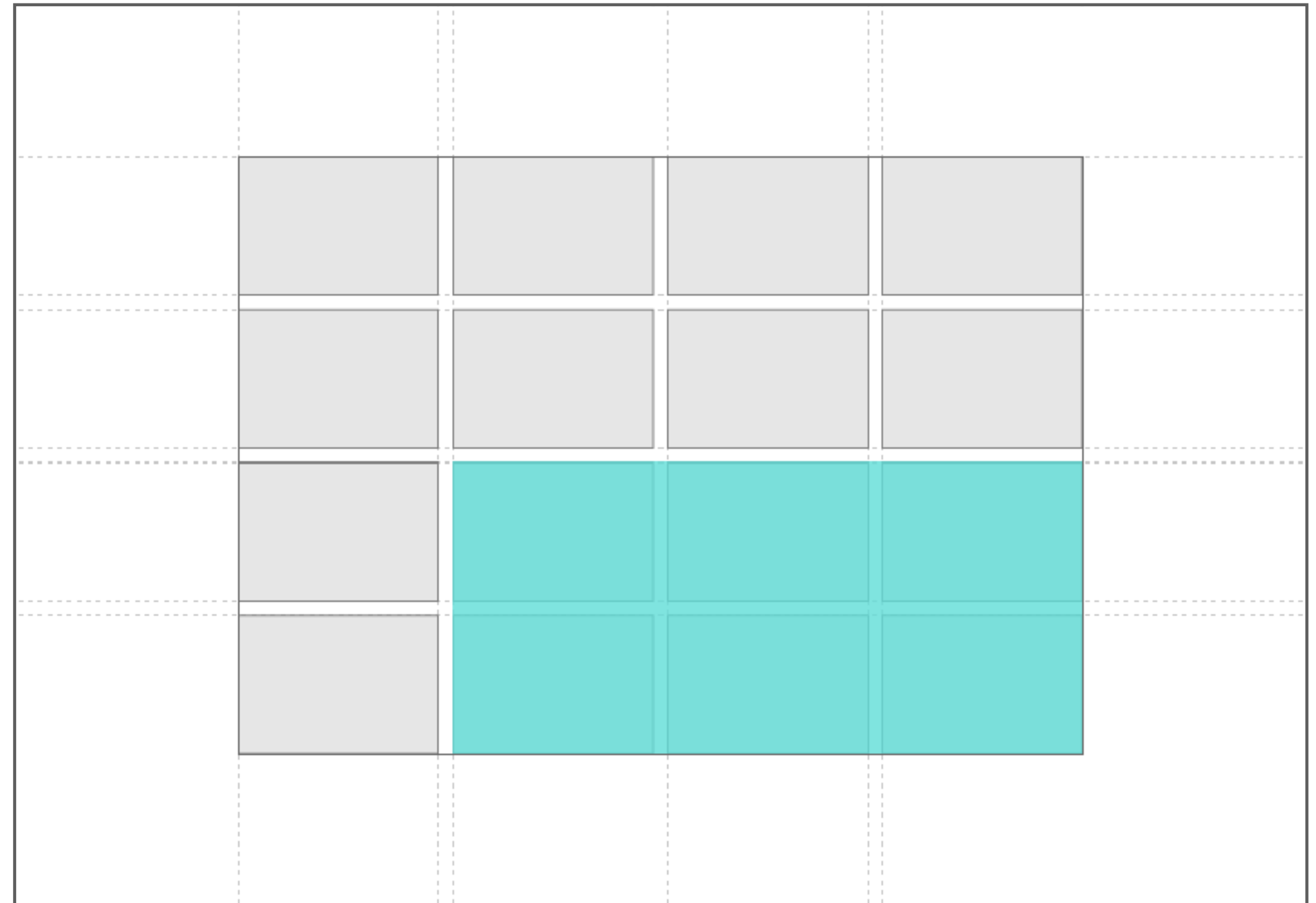
```
grid-template-columns
```



Cells

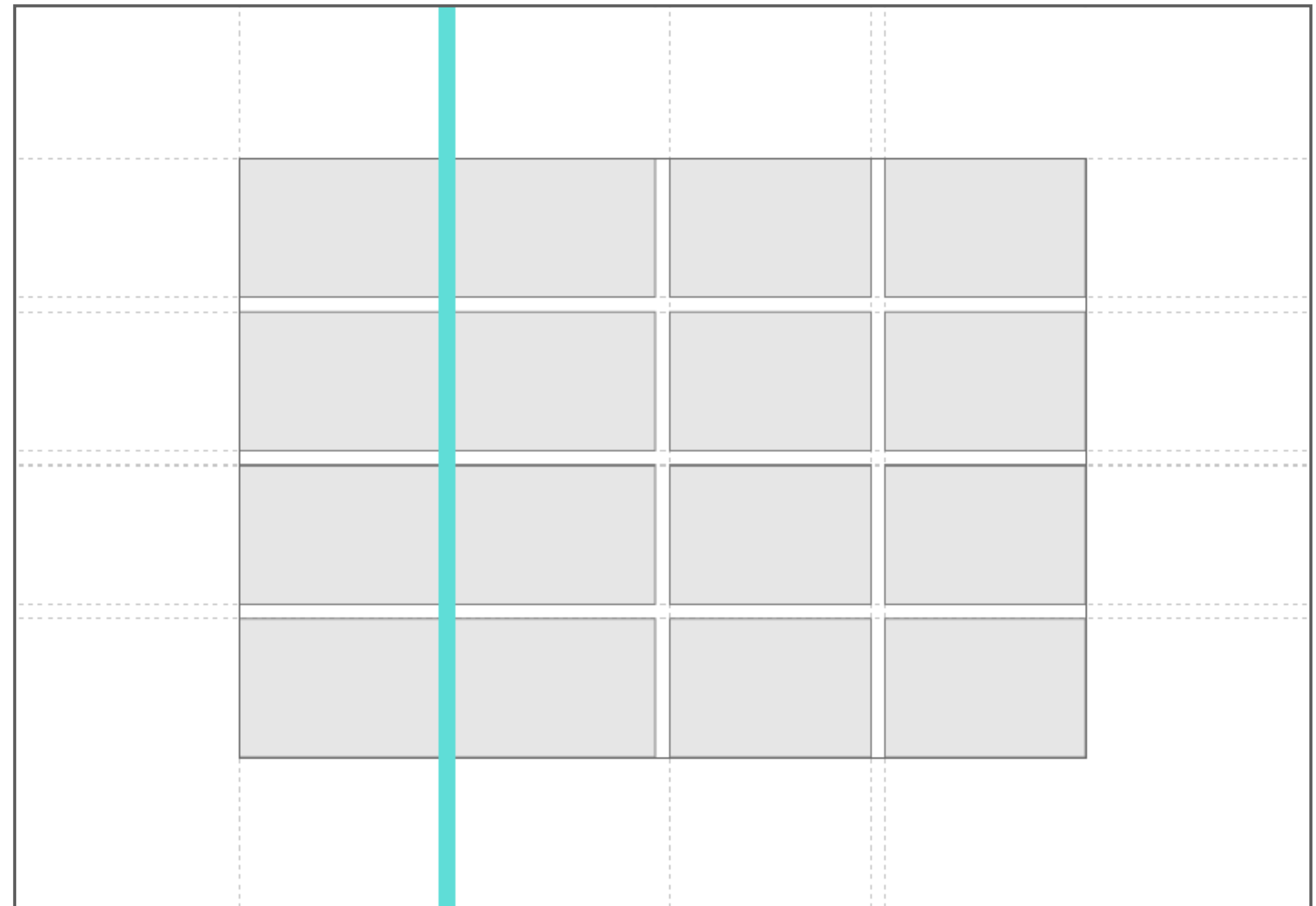


Grid areas



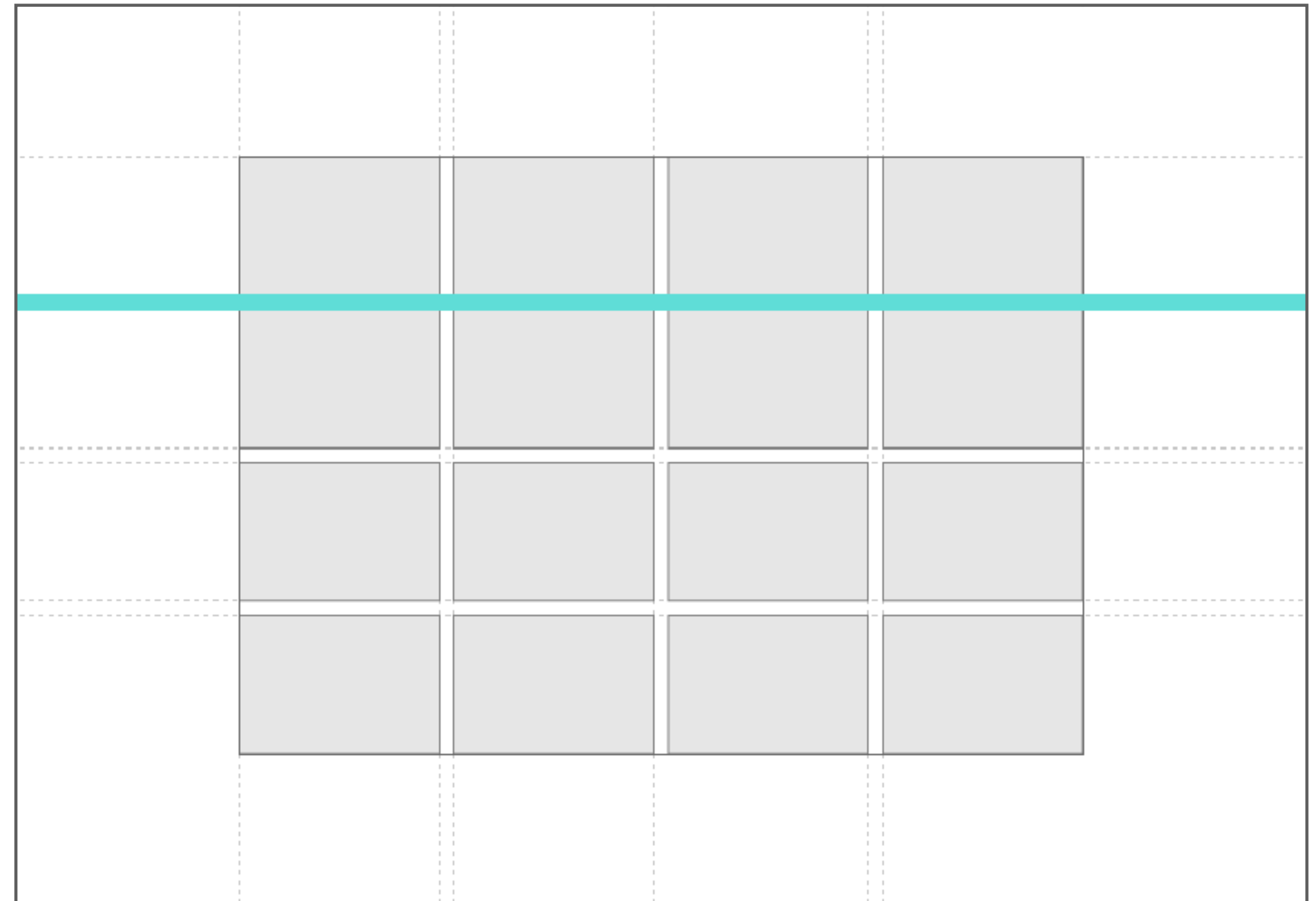
Gutters

`column-gap`
(Formerly `grid-column-gap`)



Gutters

`row-gap`
(Formerly `grid-row-gap`)



Defining a grid

CSS

```
.grid {  
  display: grid;  
}
```

codepen.io/michellebarker/pen/eLZwVg

CSS

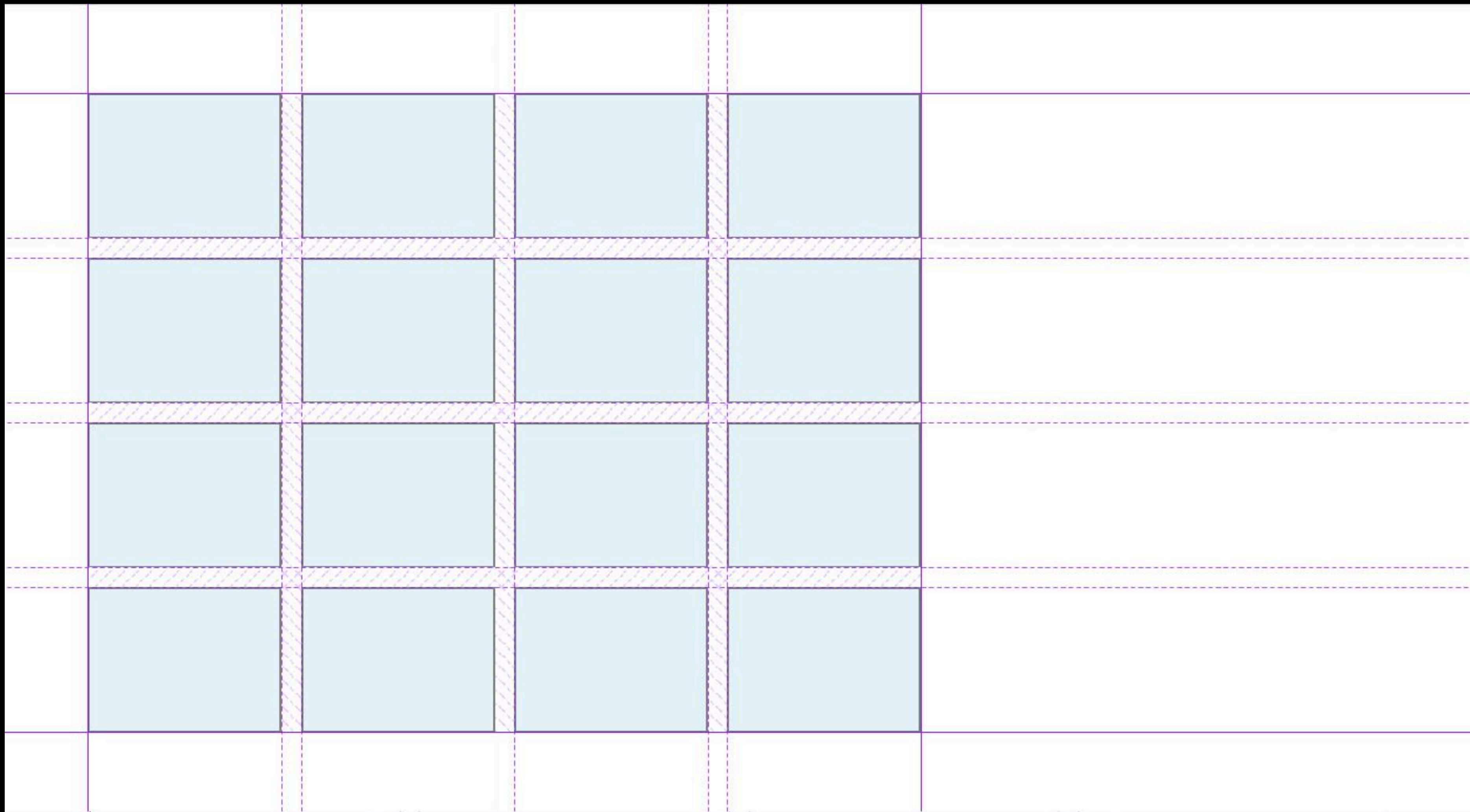
```
.grid {  
  display: grid;  
  grid-template-columns: 200px 200px 200px 200px;  
  grid-template-rows: 150px 150px 150px 150px;  
}
```

codepen.io/michellebarker/pen/eLZwVg

CSS

```
.grid {  
  display: grid;  
  grid-template-columns: 200px 200px 200px 200px;  
  grid-template-rows: 150px 150px 150px 150px;  
  column-gap: 20px;  
  row-gap: 20px;  
}
```

codepen.io/michellebarker/pen/eLZwVg



codepen.io/michellebarker/pen/eLZwVg

@mbarker_84

CSS

repeat()

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 200px);  
  grid-template-rows: repeat(4, 150px);  
  grid-gap: 20px;  
}
```

codepen.io/michellebarker/pen/eLZwVg

CSS

Shorthand

```
.grid {  
  display: grid;  
  grid-template: repeat(4, 150px) / repeat(4, 200px);  
}
```

codepen.io/michellebarker/pen/eLZwVg

CSS

Shorthand

```
.grid {  
  display: grid;  
  grid-template: repeat(4, 150px) / repeat(4, 200px);  
  gap: 20px; // row-gap / column-gap  
}
```

codepen.io/michellebarker/pen/eLZwVg

Fraction units (fr)

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-template-rows: repeat(4, 150px);  
  gap: 20px;  
}
```


Fr = fraction units

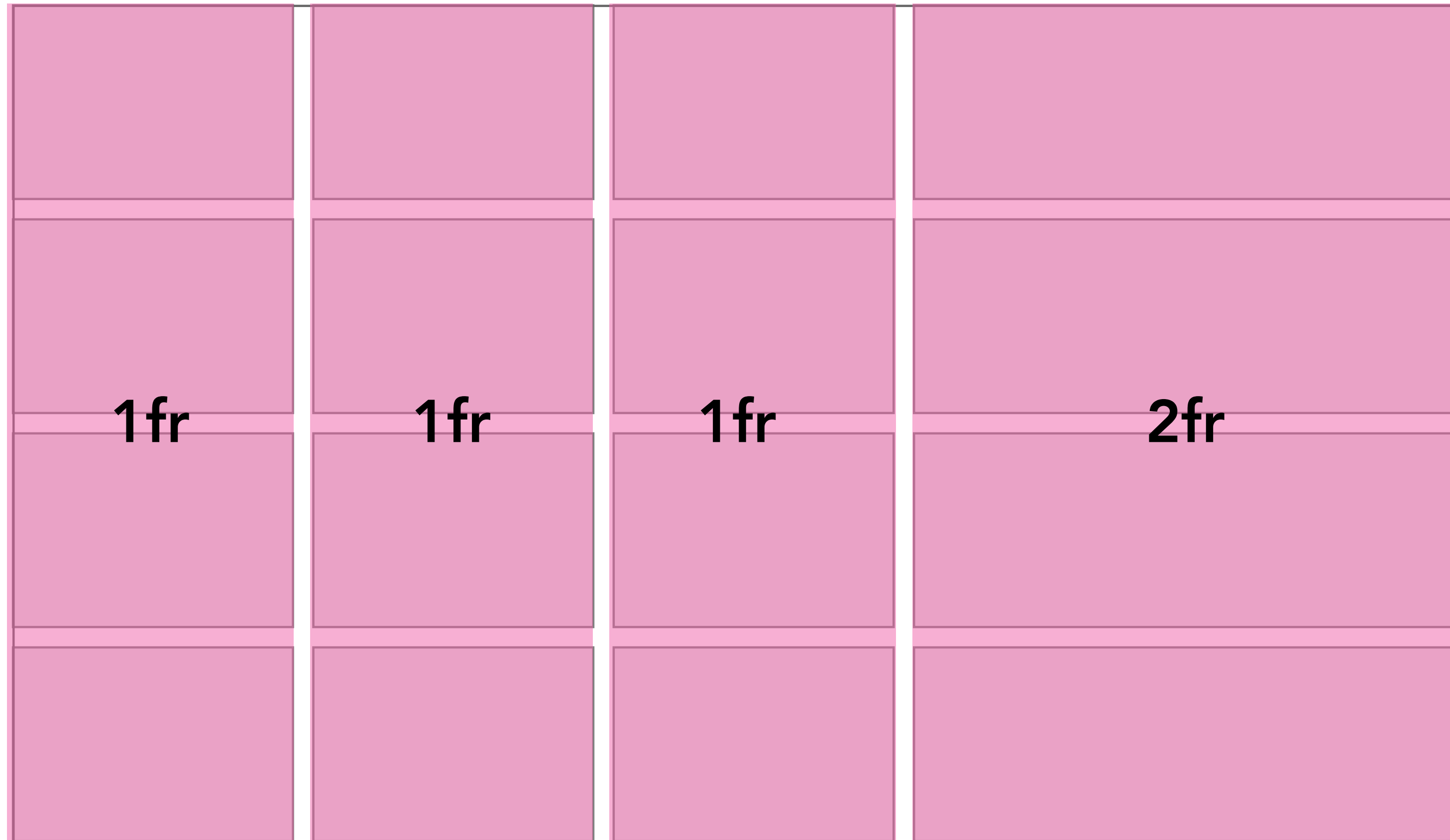
**a fraction of the leftover space in the
grid container**

CSS

Fraction units (fr)

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-template-rows: repeat(4, 150px);  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/eLZwVg

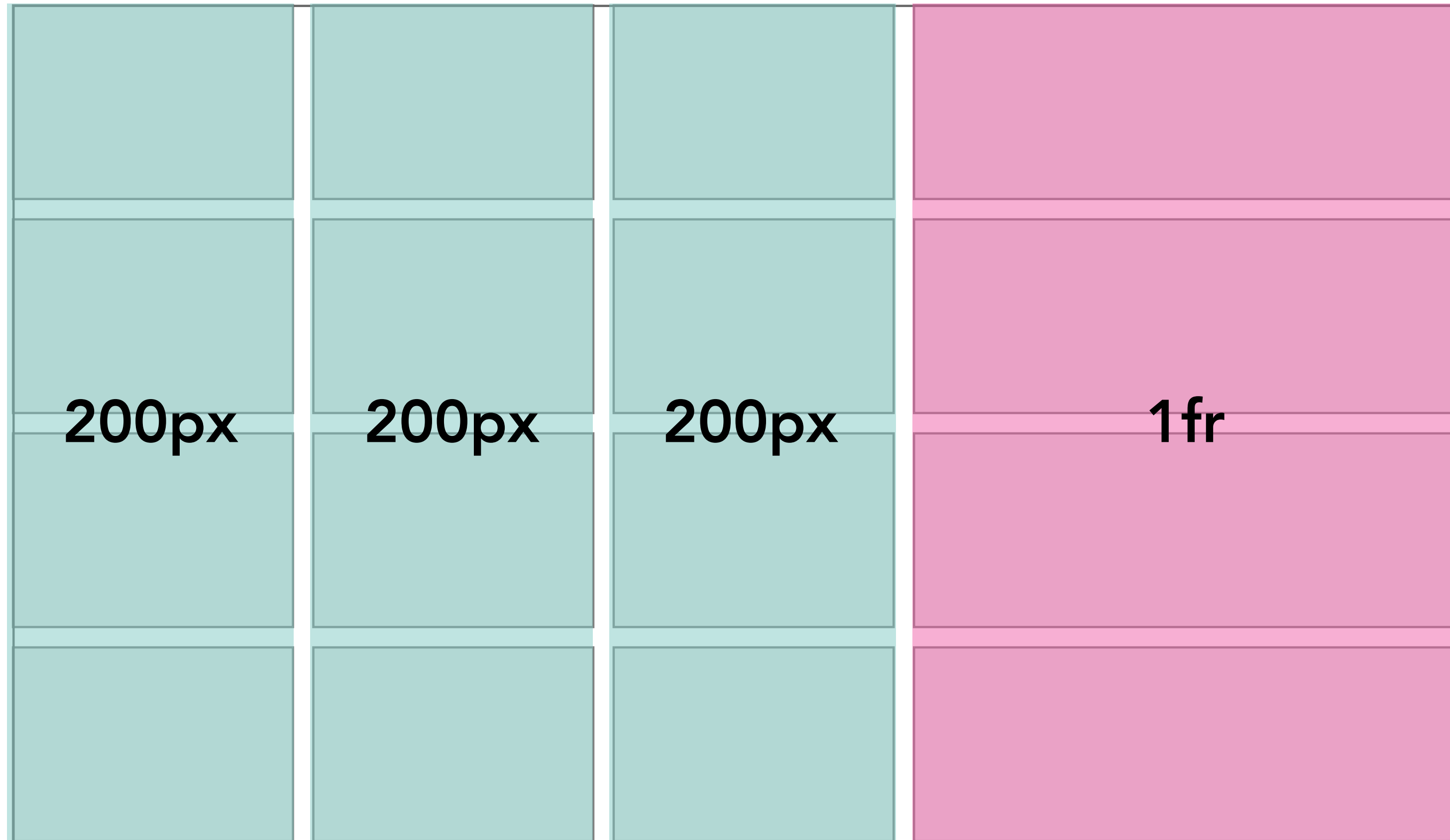


CSS

Fraction units (fr)

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(3, 200px) 1fr;  
  grid-template-rows: repeat(4, 150px);  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/eLZwVg



codepen.io/michellebarker/pen/wEdjqX

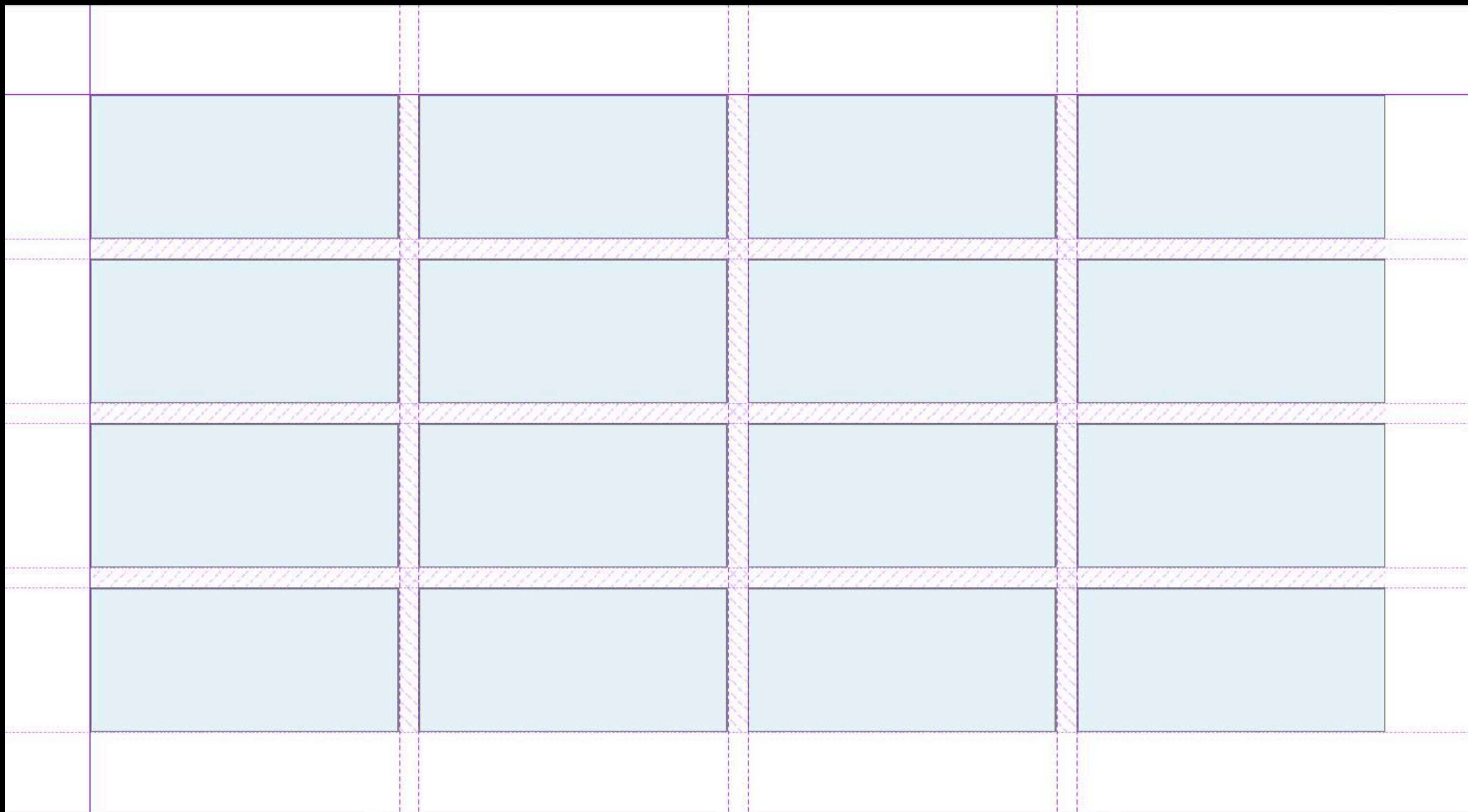
Implicit vs Explicit tracks

Implicit tracks

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-auto-rows: 150px;  
  gap: 20px;  
}
```

Implicit tracks

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-template-rows: repeat(4, 150px);  
  grid-auto-rows: 150px;  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/eLZwVg

@mbarker_84

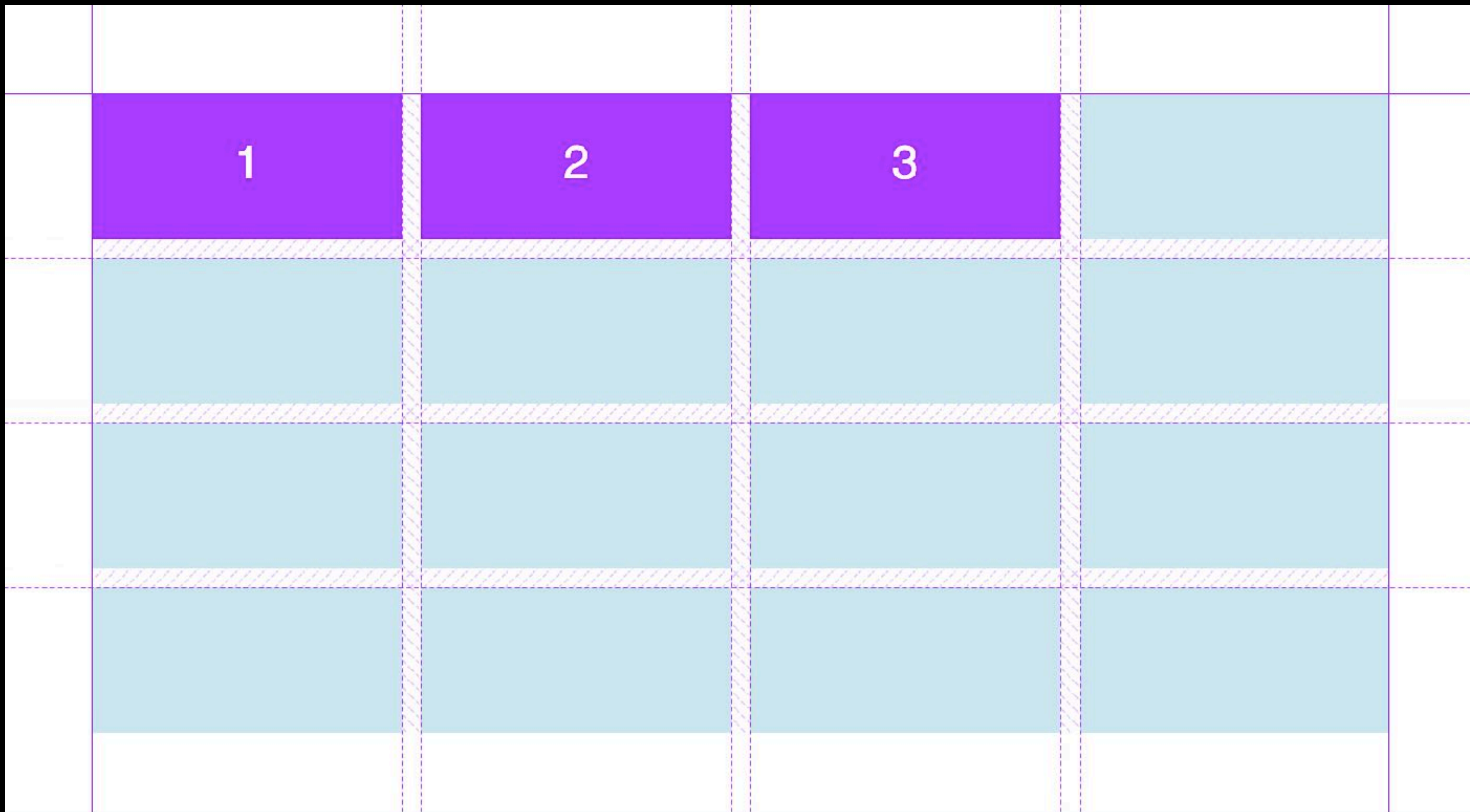
Placing items

HTML

Auto-placement

```
<div class="grid">  
  <div class="grid__item">1</div>  
  <div class="grid__item">2</div>  
  <div class="grid__item">3</div>  
</div>
```

codepen.io/michellebarker/pen/gdrVXP



codepen.io/michellebarker/pen/gdrVXP



My Awesome Grid

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

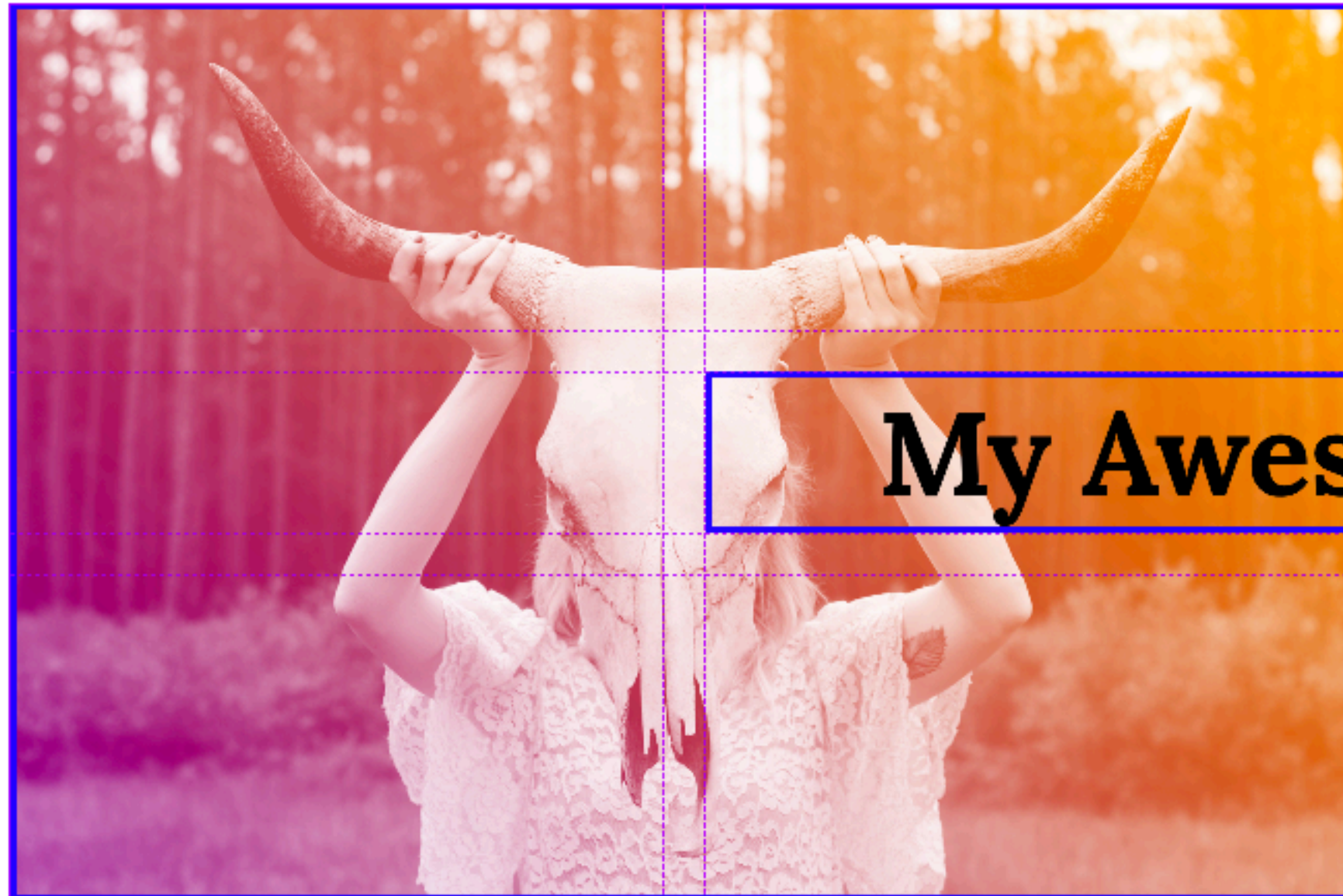


My Awesome Grid

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Placing items by grid line

	1	2	3	4	5
1					
2					
3					
4					
5					



My Awesome Grid

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

CSS

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-template-rows: 1fr auto 1fr;  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/MRrBwa

CSS

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-template-rows: 1fr auto 1fr;  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/MRrBwa

CSS

Placing items by grid line

```
.grid__image {  
  grid-column-start: 1;  
  grid-column-end: 3;  
  grid-row-start: 1;  
  grid-row-end: 5;  
}
```

codepen.io/michellebarker/pen/MRrBwa

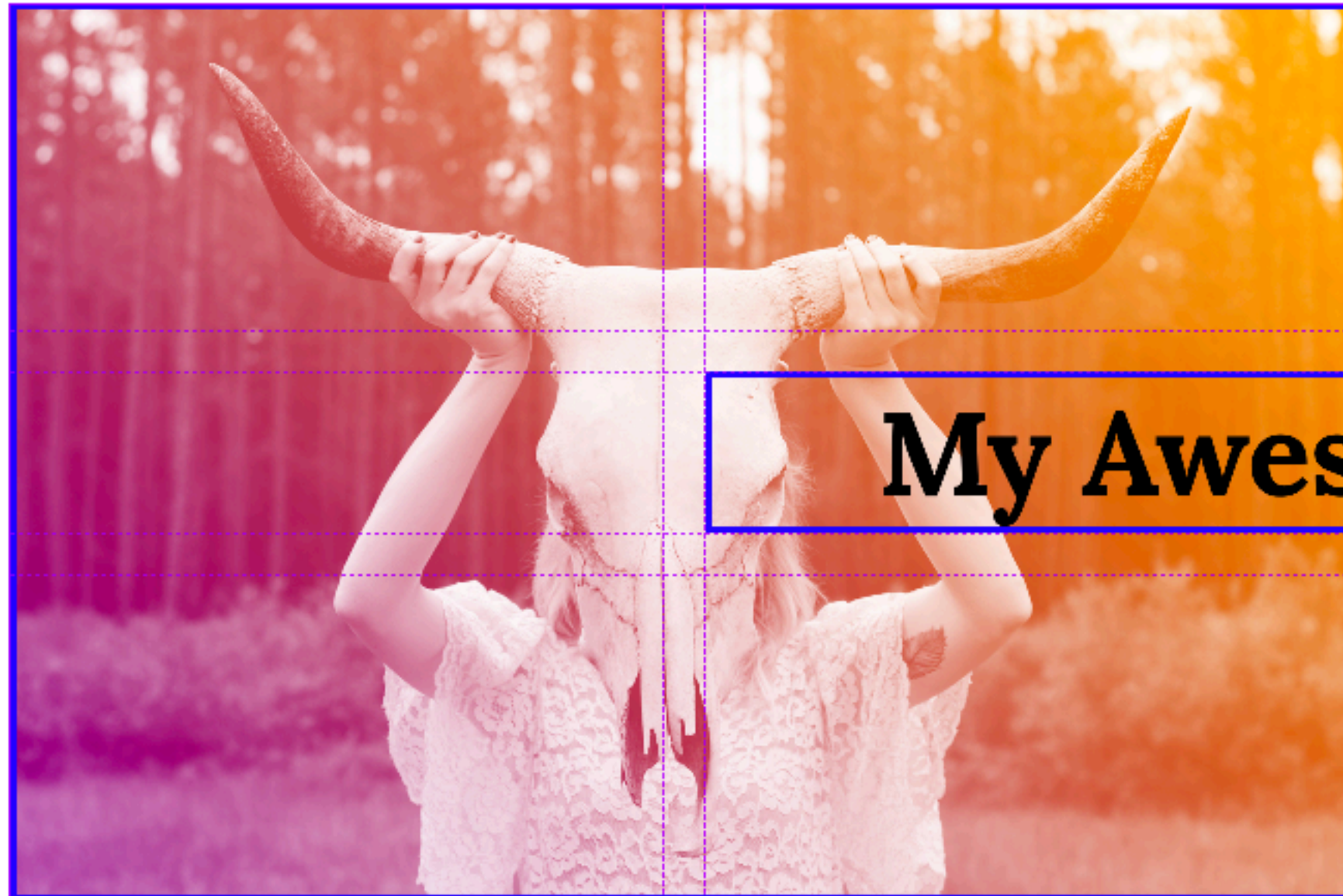
CSS

Placing items by grid line

```
.grid_heading {  
  grid-column-start: 2;  
  grid-column-end: 4;  
  grid-row-start: 2;  
}
```

```
.grid_text {  
  grid-column-start: 4;  
  grid-row-start: 1;  
}
```

codepen.io/michellebarker/pen/YOqmjP



My Awesome Grid

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

CSS

```
.grid__heading {  
  grid-column: 2 / 4;  
  grid-row: 2;  
}  
  
.grid__image {  
  grid-column: 1 / 3;  
  grid-row: 1 / 4;  
}  
  
.grid__text {  
  grid-column: 4;  
}
```

codepen.io/michellebarker/pen/MRrBwa

CSS

```
.grid__heading {  
  grid-column: 2 / 4;  
  grid-row: 2;  
}
```

```
.grid__image {
```

```
  grid-column: 1 / span 2;  
  grid-row: 1 / span 3;
```

```
}
```

```
.grid__text {  
  grid-column: 4;  
}
```

codepen.io/michellebarker/pen/MRrBwa

CSS

Start line / end line

```
.grid__image {  
  grid-column: 1 / 3;  
  grid-row: 1 / 4;  
}
```

CSS

Start line / span

```
.grid__image {  
  grid-column: 1 / span 2;  
  grid-row: 1 / 4;  
}
```

CSS

Span / end line

```
.grid__image {  
  grid-column: span 2 / 3;  
  grid-row: 1 / 4;  
}
```

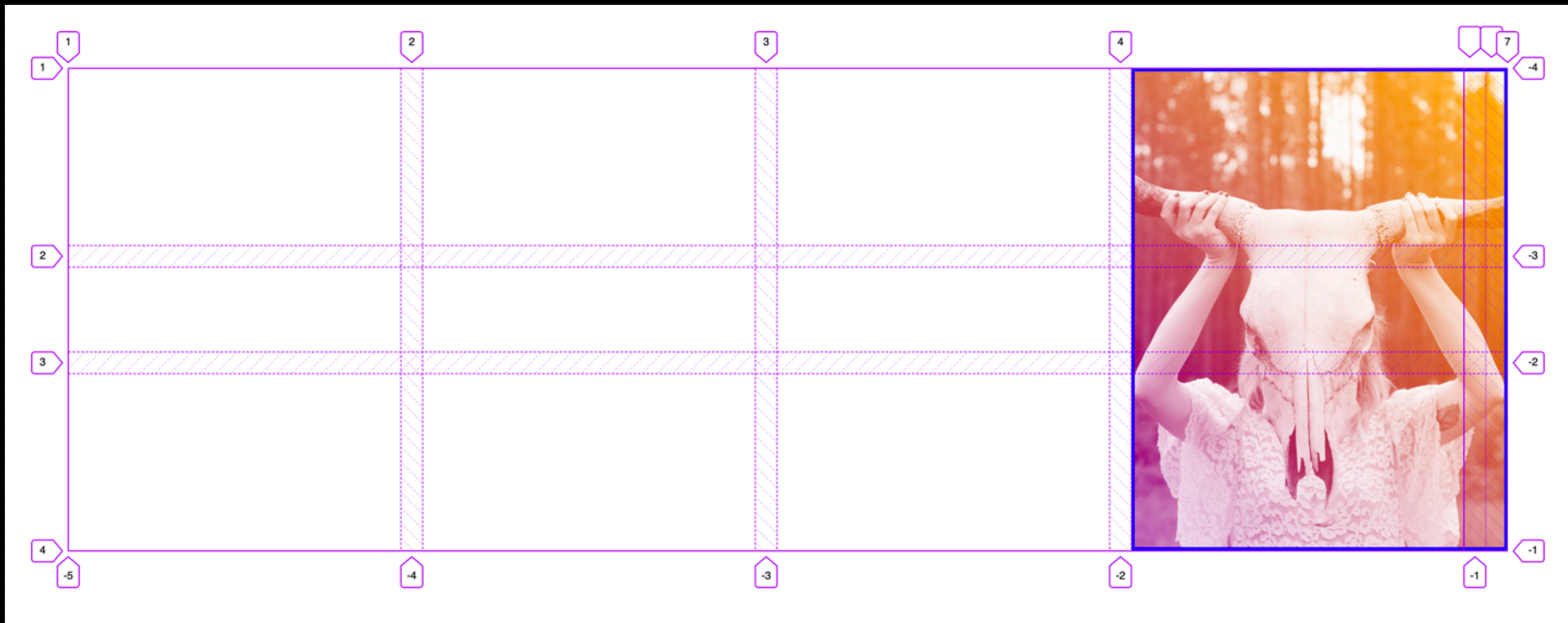
CSS



Implicit tracks

```
.grid__image {  
  grid-column: 3 / span 3;  
  grid-row: 1 / 4;  
}
```

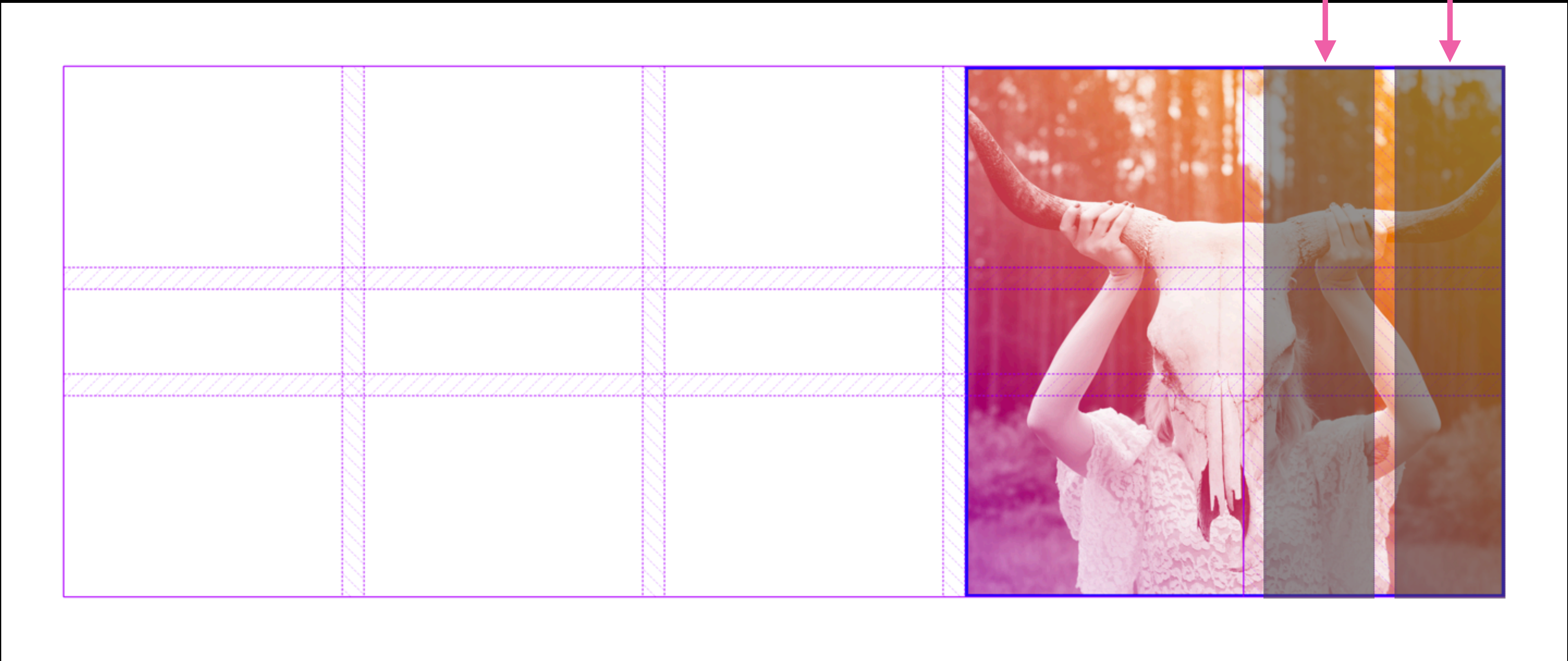
codepen.io/michellebarker/pen/dLJqLY



codepen.io/michellebarker/pen/dLJqLY

@mbarker_84

```
grid-auto-columns: 100px;
```



codepen.io/michellebarker/pen/dLJqLY

CSS

Naming grid lines

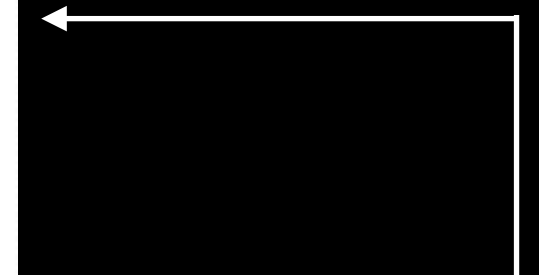
```
.grid {  
  display: grid;  
  
  grid-template-columns:  
    1fr [heading-start] repeat(2, 1fr) [heading-end] 1fr;  
  grid-template-rows:  
    1fr [heading-start] auto [heading-end] 1fr;  
  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/VNyEeK

heading-start

heading-end

heading-start



heading-end



CSS

```
.grid_heading {  
  grid-area: heading;  
}
```

codepen.io/michellebarker/pen/VNyEeK

CSS

Naming grid lines

```
.grid {  
  display: grid;  
  grid-template-columns:  
    [image-start] 1fr [heading-start] 1fr [image-end]  
    1fr [heading-end text-start] 1fr [text-end];  
  grid-template-rows:  
    [image-start text-start] 1fr  
    [text-end heading-start] auto [heading-end]  
    1fr [image-end];  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/VNyEeK

CSS

grid-template-areas

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-template-rows: 1fr auto 1fr;  
  
  grid-template-areas: "i i . t"  
                      "i i . ."  
                      "i i . .";  
  
  gap: 20px;  
}
```

codepen.io/michellebarker/pen/MRQLrL

CSS

```
.grid__image {  
  grid-area: i;  
}
```

```
.grid__text {  
  grid-area: t;  
}
```

codepen.io/michellebarker/pen/VNyEeK

CSS

```
.grid__heading {  
  grid-column: 2 / -2;  
  grid-row: 2;  
}  
  
.grid__image {  
  grid-area: i;  
}  
  
.grid__text {  
  grid-area: t;  
}
```

codepen.io/michellebarker/pen/MRQLrL

CSS

```
.grid__heading {  
  grid-column: 2 / -2;  
  grid-row: 2;  
}  
  
.grid__image {  
  grid-area: i;  
}  
  
.grid__text {  
  grid-area: t;  
}
```

codepen.io/michellebarker/pen/MRQLrL

-5



-4



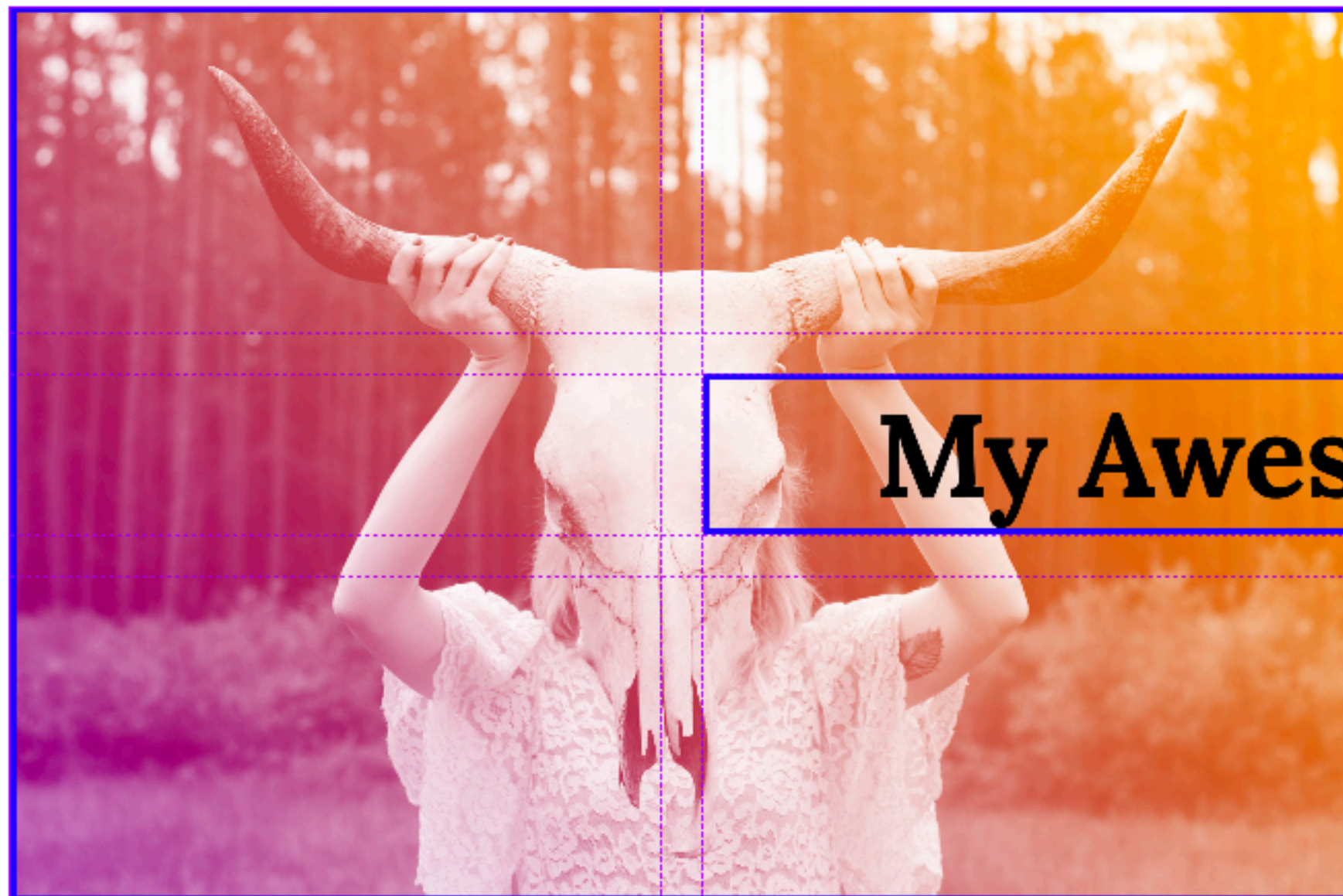
-3



-2



-1



My Awesome Grid

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

CSS

Alignment

```
.grid__text {  
  grid-area: t;  
  align-self: flex-start;  
}
```

codepen.io/michellebarker/pen/MRQLrL

Alignment



```
align-self: flex-end;
```



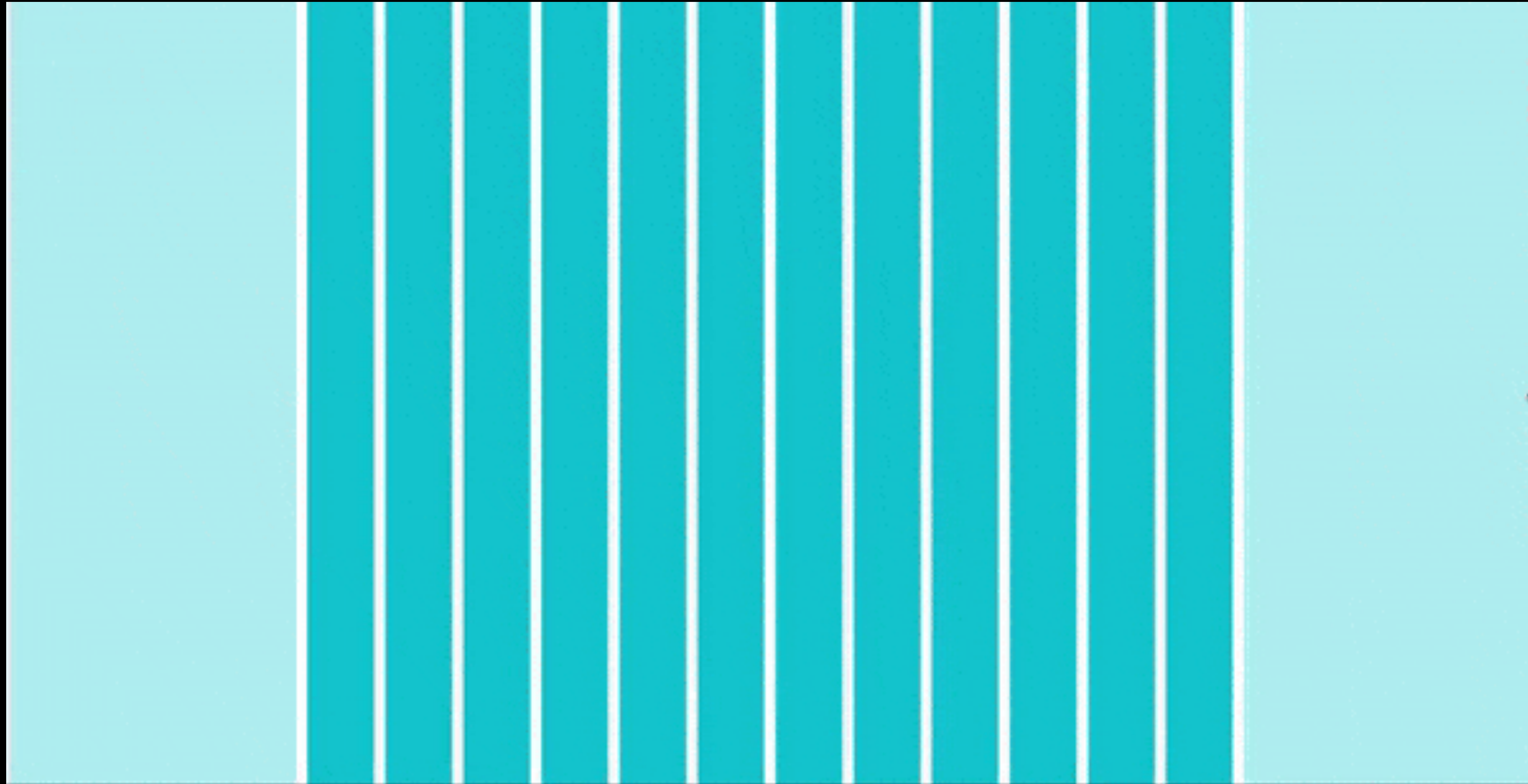
minmax()

minmax(20px, 1fr)

Minimum size

Maximum size

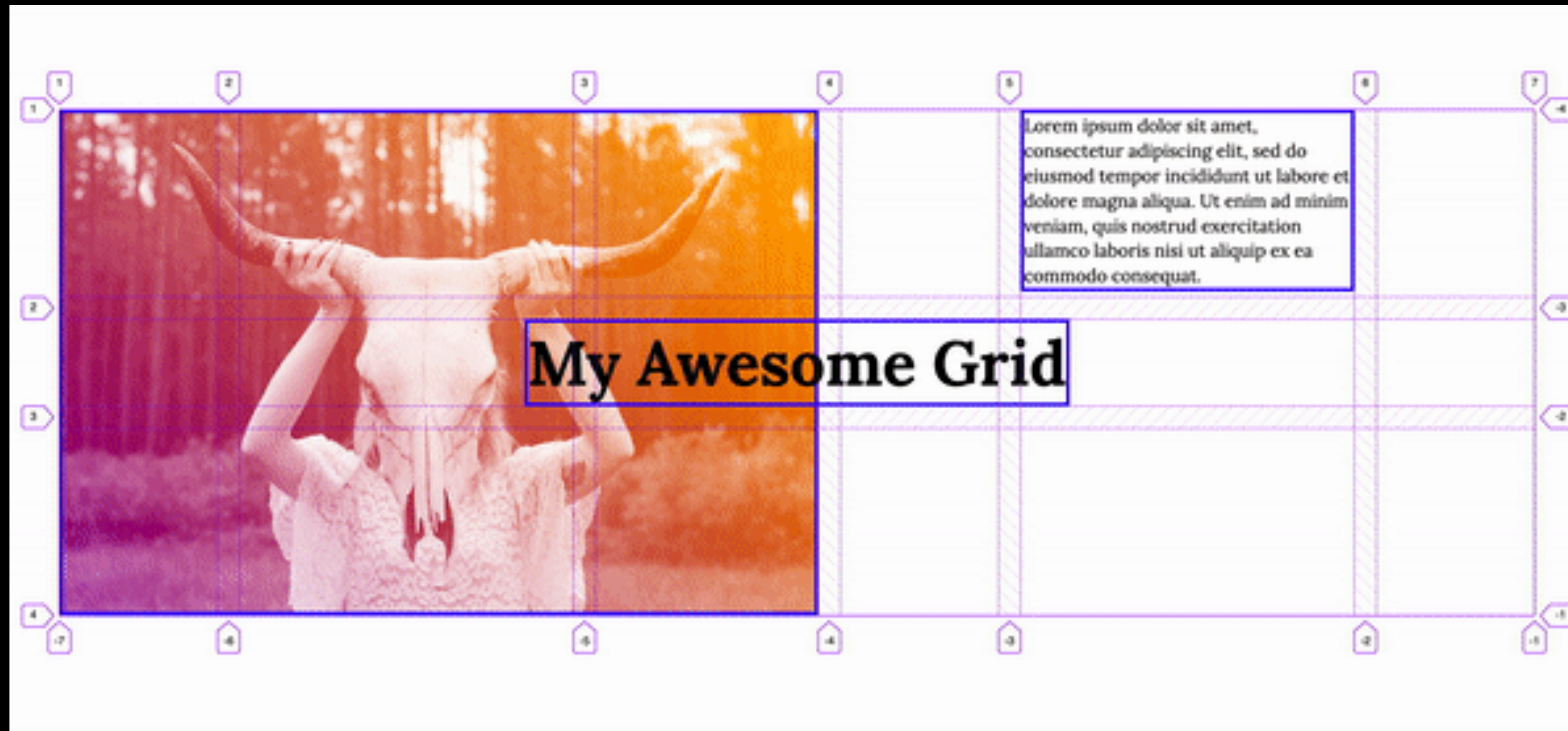
`minmax(20px, 1fr)`



`minmax(0, 60px)`

codepen.io/michellebarker/pen/wYGMPQ

```
grid-template-columns:
minmax(20px, 1fr) minmax(auto, 300px) minmax(0, 200px)
minmax(0, 1fr) minmax(auto, 300px) minmax(20px, 1fr);
```



codepen.io/michellebarker/live/NmzJKv

Resources

Grid by Example (Rachel Andrew)

gridbyexample.com

Layout Land (Jen Simmons)

layout.land

CSS Grid Playground (MDN)

mozilladevelopers.github.io/playground/css-grid

CSS { In Real Life }

css-irl.info

Thank you for listening

@mbarker_84 / @CSSInRealLife