Super-powered Layouts

with

CSS Grid + CSS Variables

CSS Grid

(CSS Grid Layout Module Level 1)

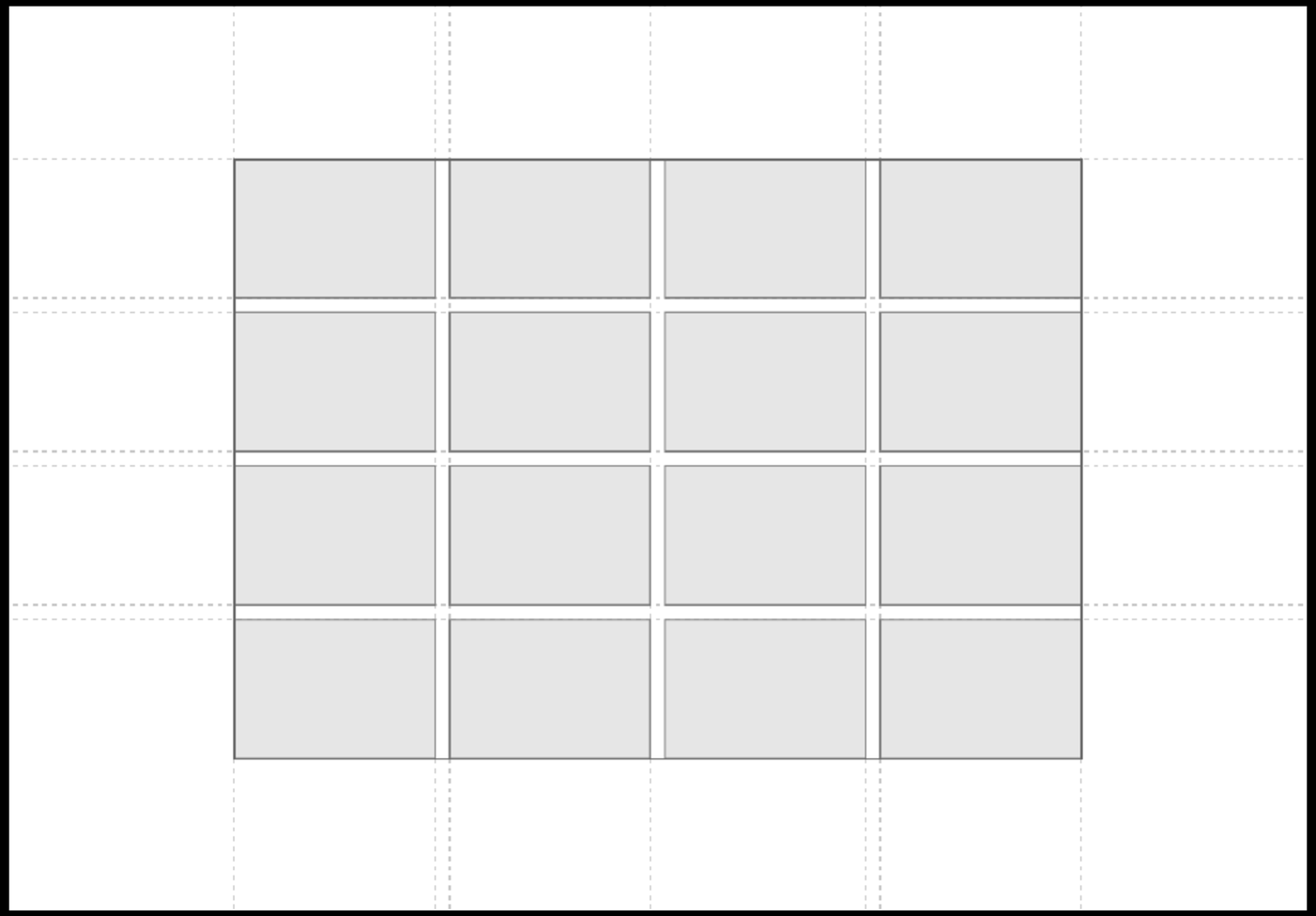
Why not just use flexbox?

Intrinsic Web Design

Grid terminology

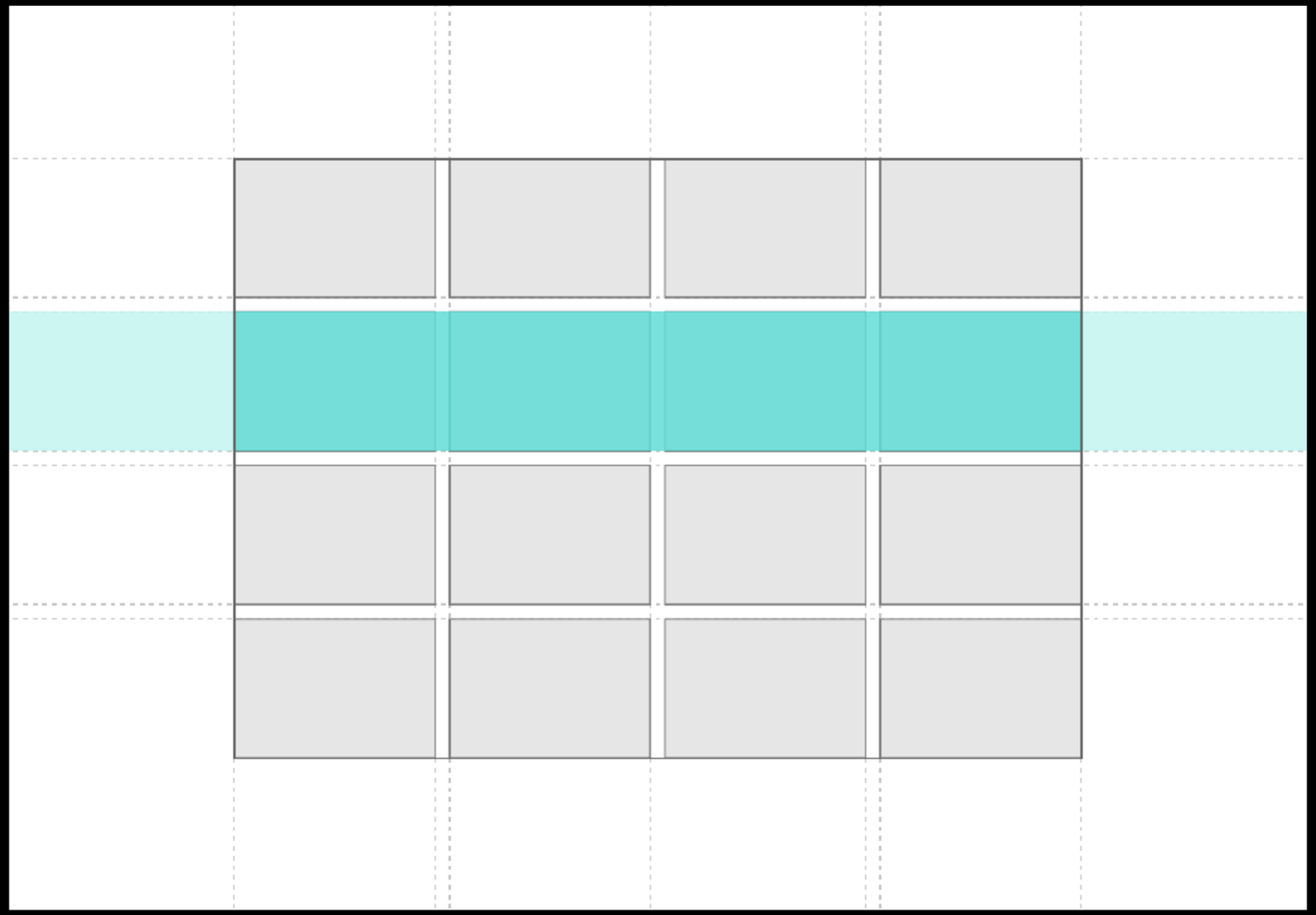
Grid

```
display: grid;
```



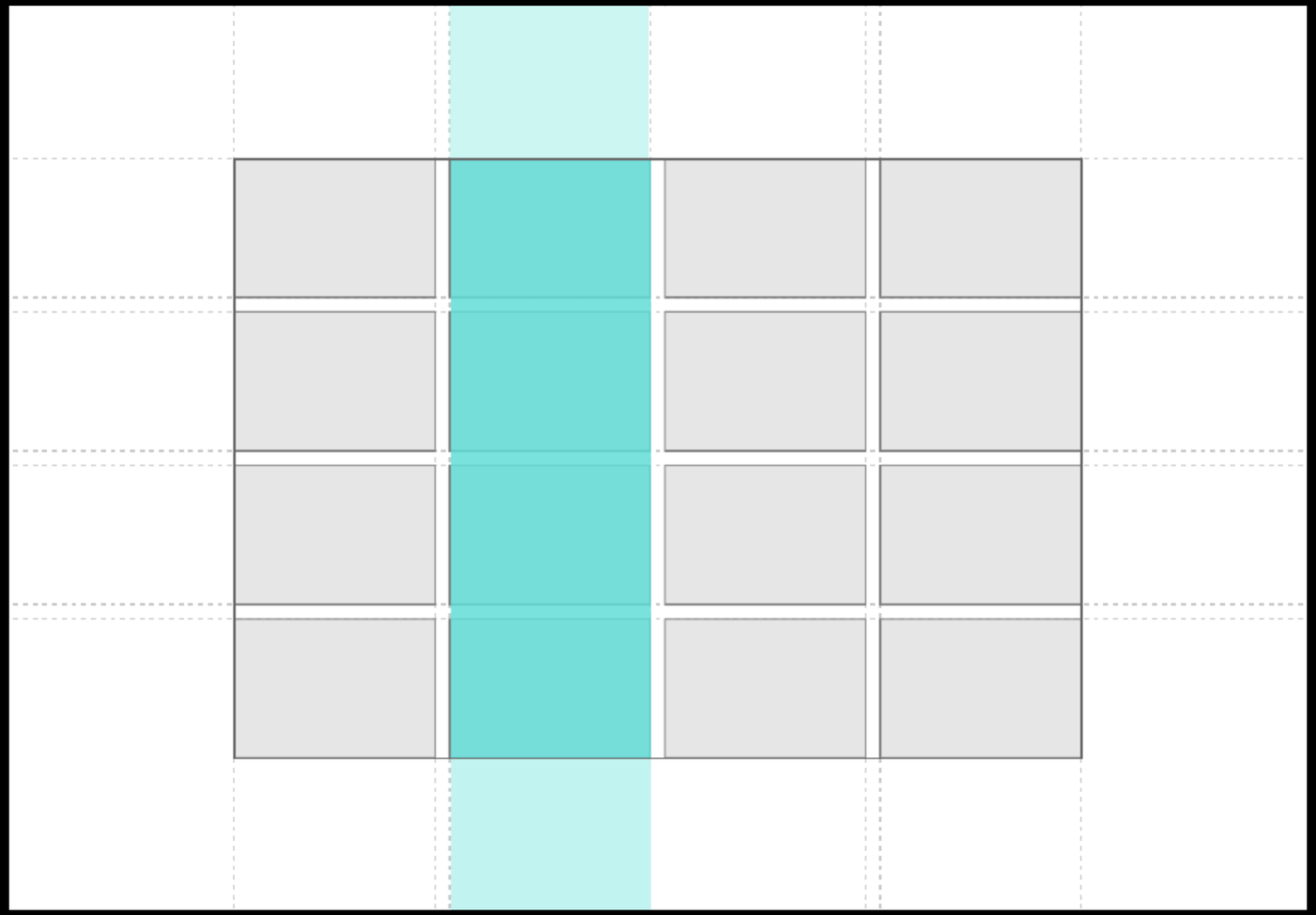
Tracks

`grid-template-rows`

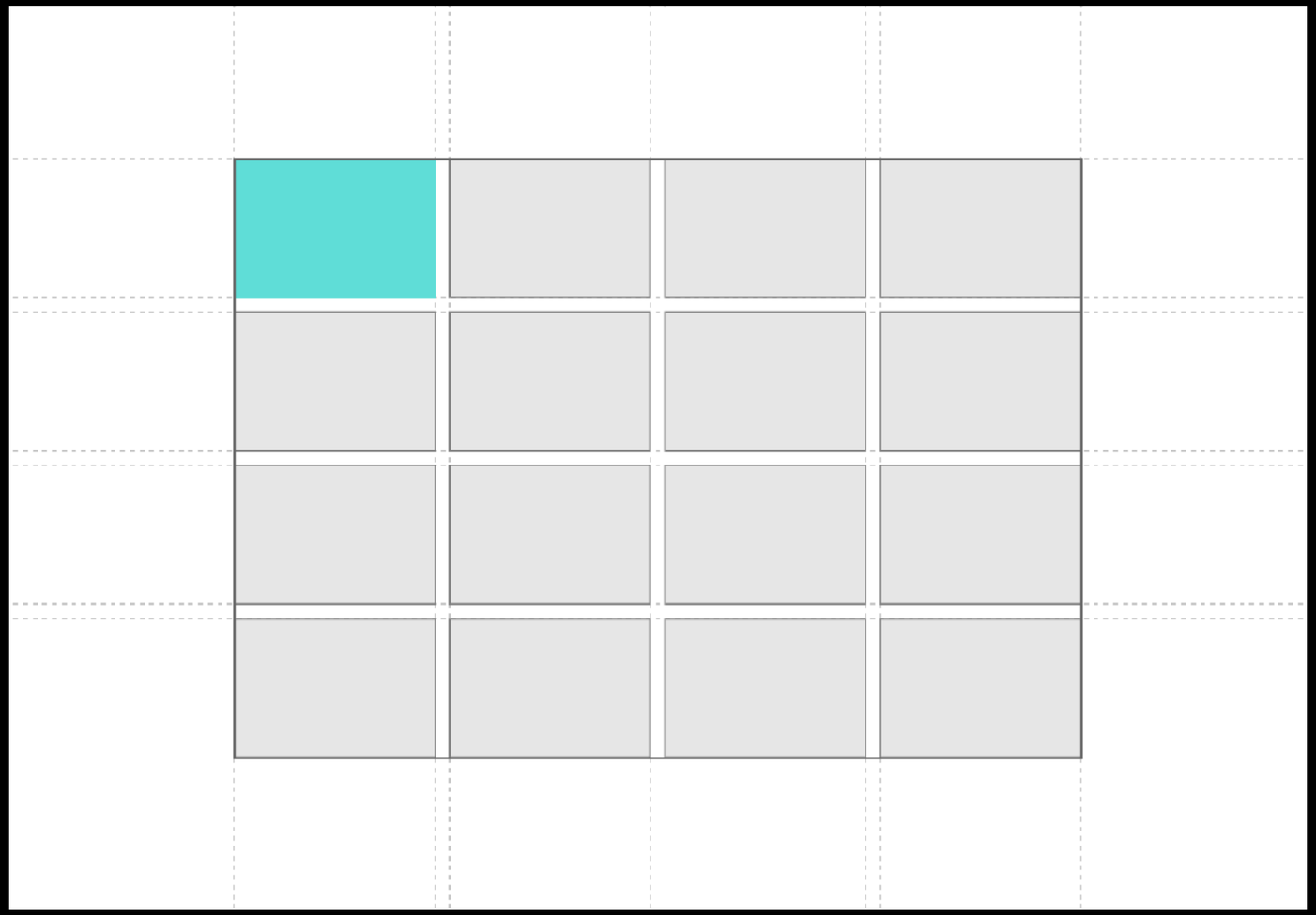


Tracks

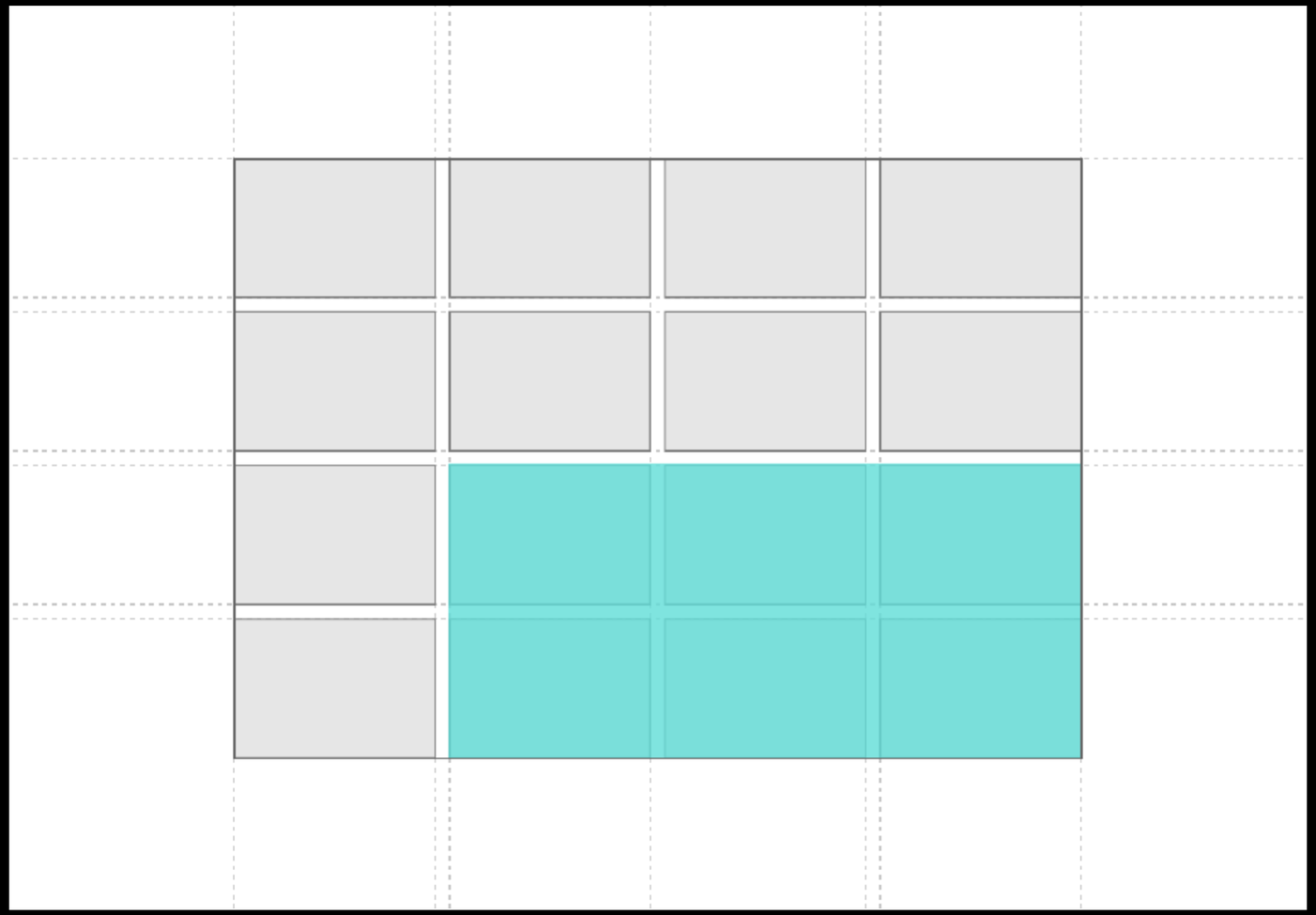
`grid-template-columns`



Cells

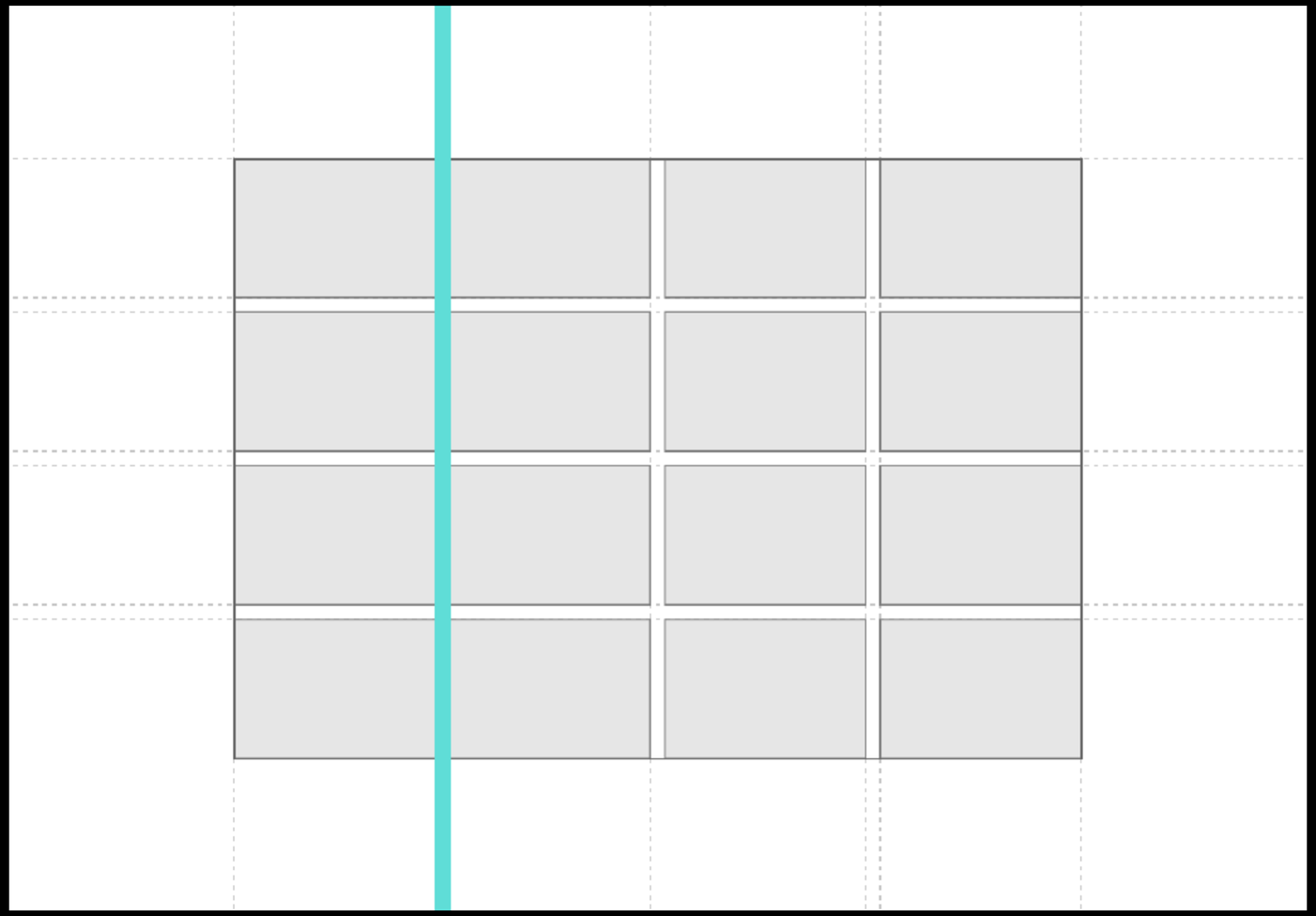


Grid areas



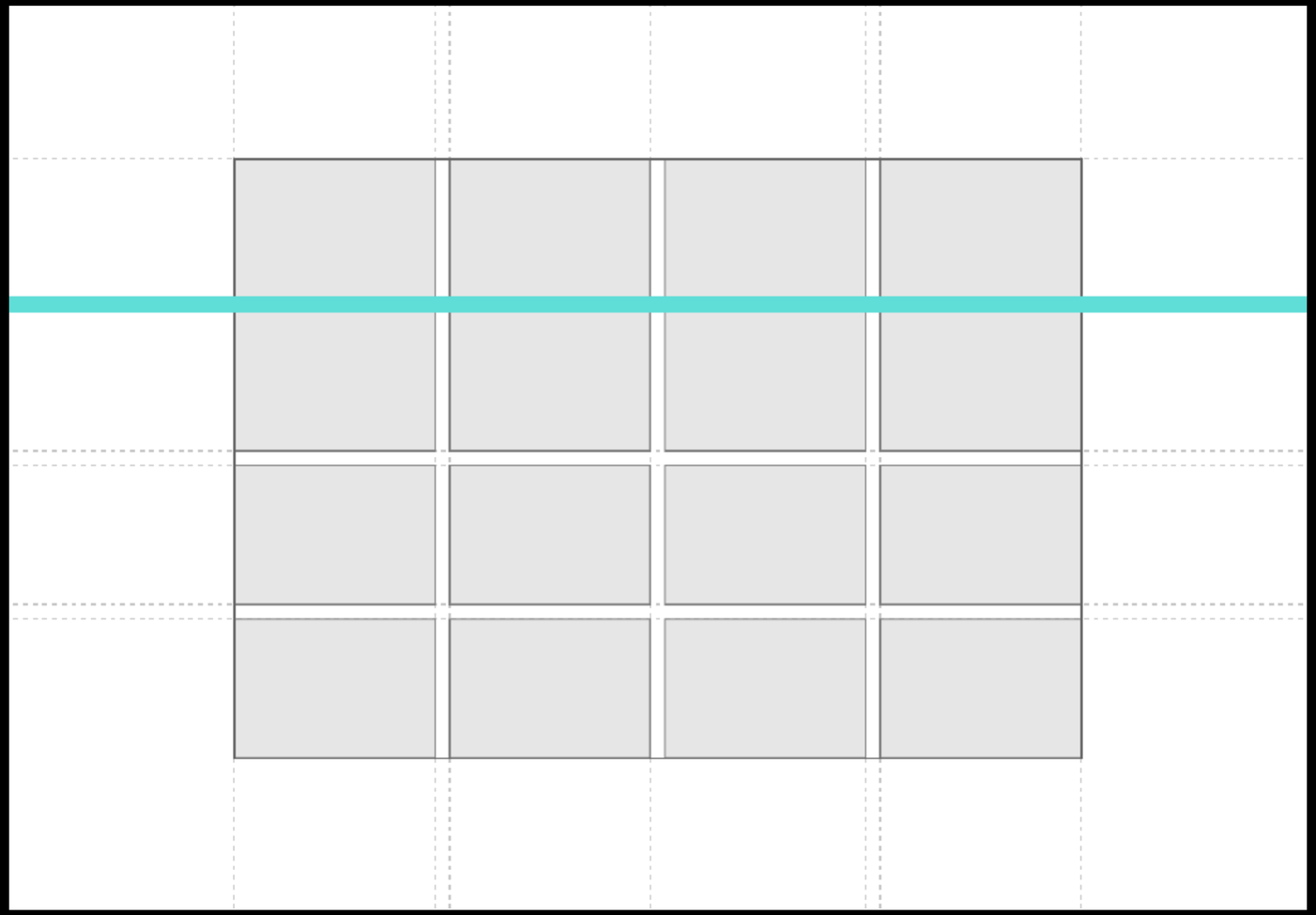
Gutters

`grid-column-gap`



Gutters

`grid-row-gap`



Defining a grid

On the container:

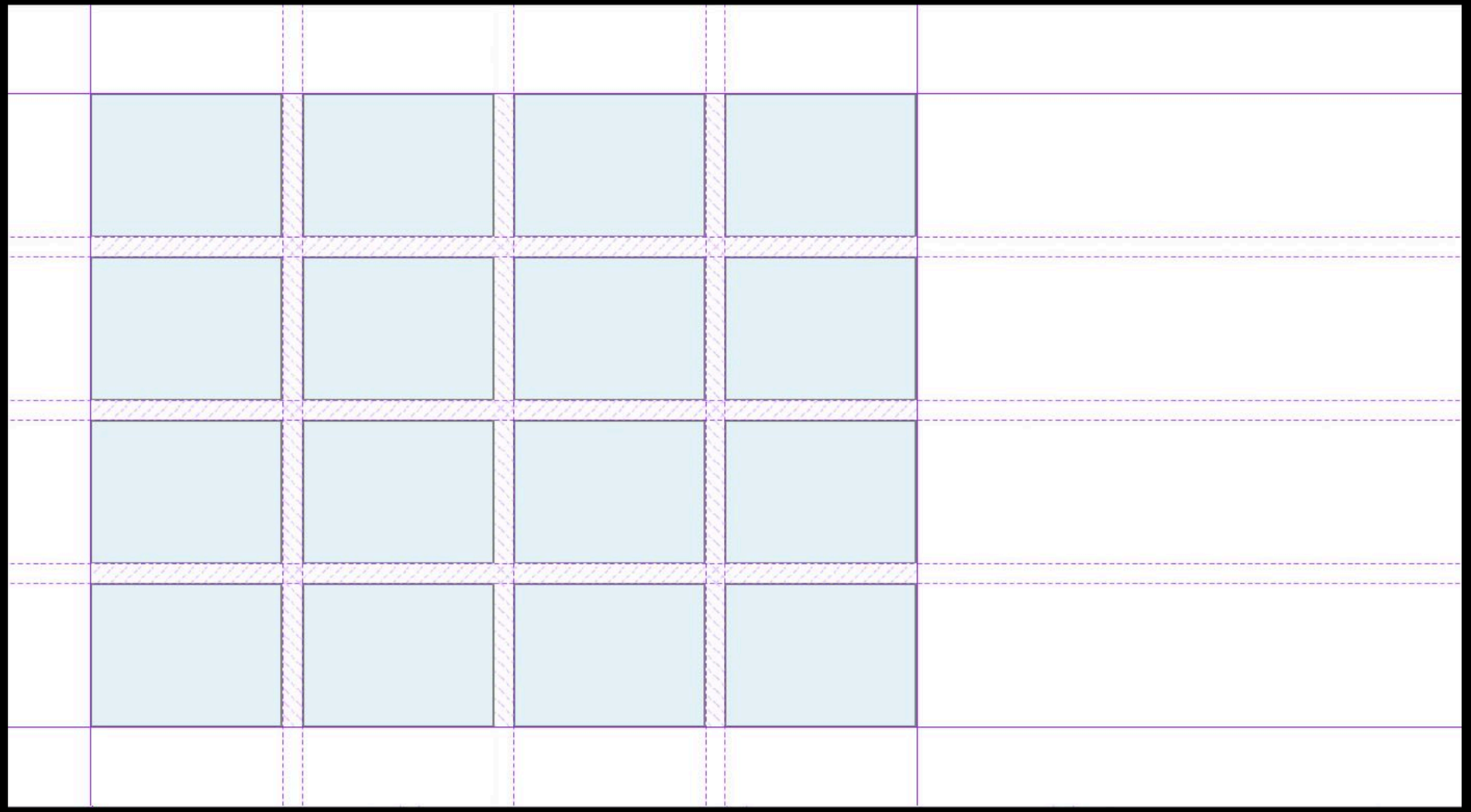
```
.grid {
  display: grid;
  grid-template-columns: 200px 200px 200px 200px;
  grid-template-rows: 150px 150px 150px 150px;
}
```

codepen.io/michellebarker/pen/eLZwVg

On the container:

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

codepen.io/michellebarker/pen/eLZwVg



codepen.io/michellebarker/pen/eLZwVg

repeat()

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

codepen.io/michellebarker/pen/eLZwVg

Shorthand

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

codepen.io/michellebarker/pen/eLZwVg

Shorthand

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

codepen.io/michellebarker/pen/eLZwVg

Flexible units (fr)

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

codepen.io/michellebarker/pen/eLZwVg

Fr

a fraction of the leftover space in the grid container

Flexible units (fr)

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

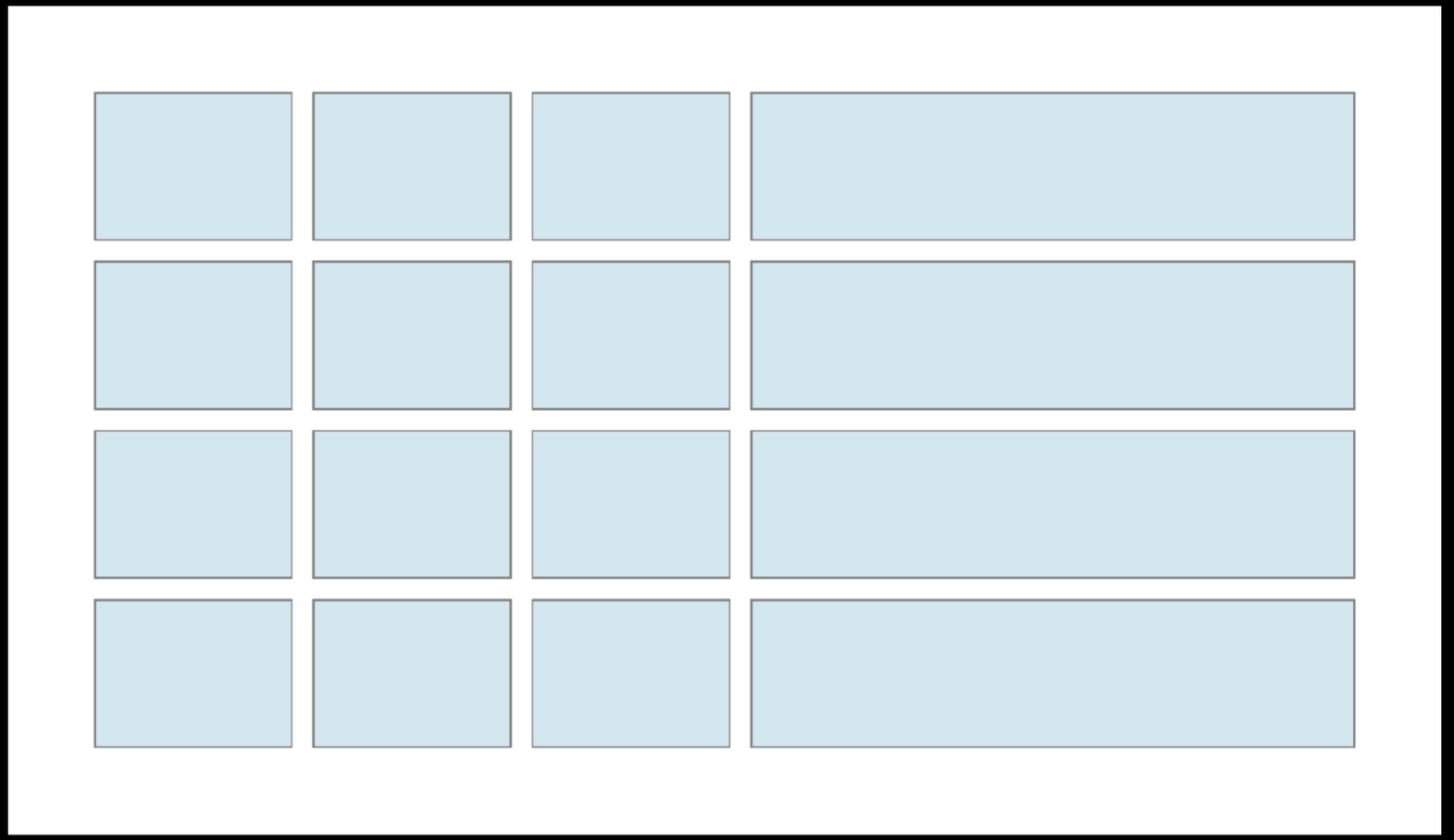
codepen.io/michellebarker/pen/eLZwVg

Flexible units (fr)

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

codepen.io/michellebarker/pen/wEdjqX

Flexible units (fr)



codepen.io/michellebarker/pen/wEdjqX

Implicit tracks

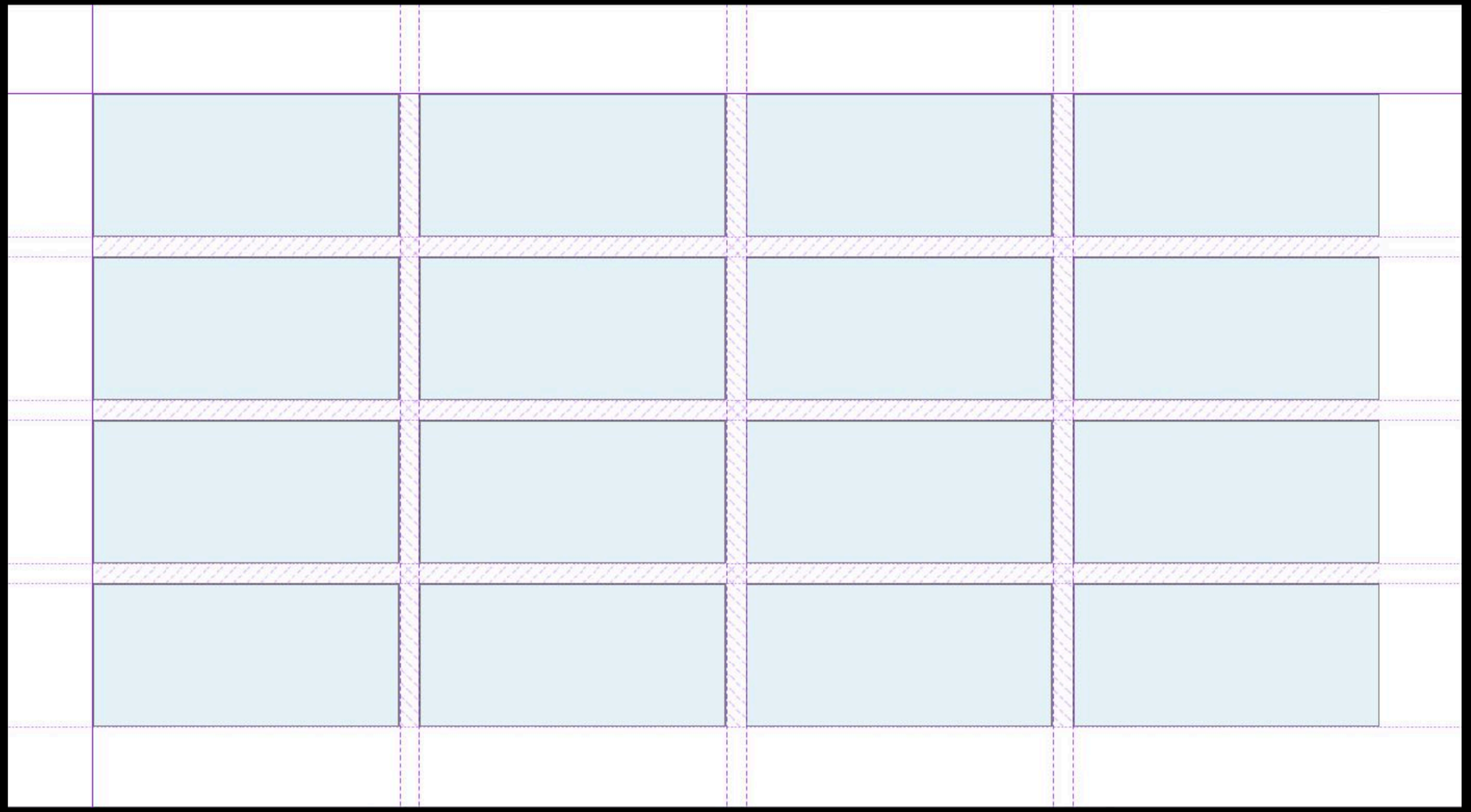
```
.grid {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-auto-rows: 150px;
  grid-gap: 20px;
}
```

codepen.io/michellebarker/pen/eLZwVg

Implicit tracks

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

codepen.io/michellebarker/pen/eLZwVg



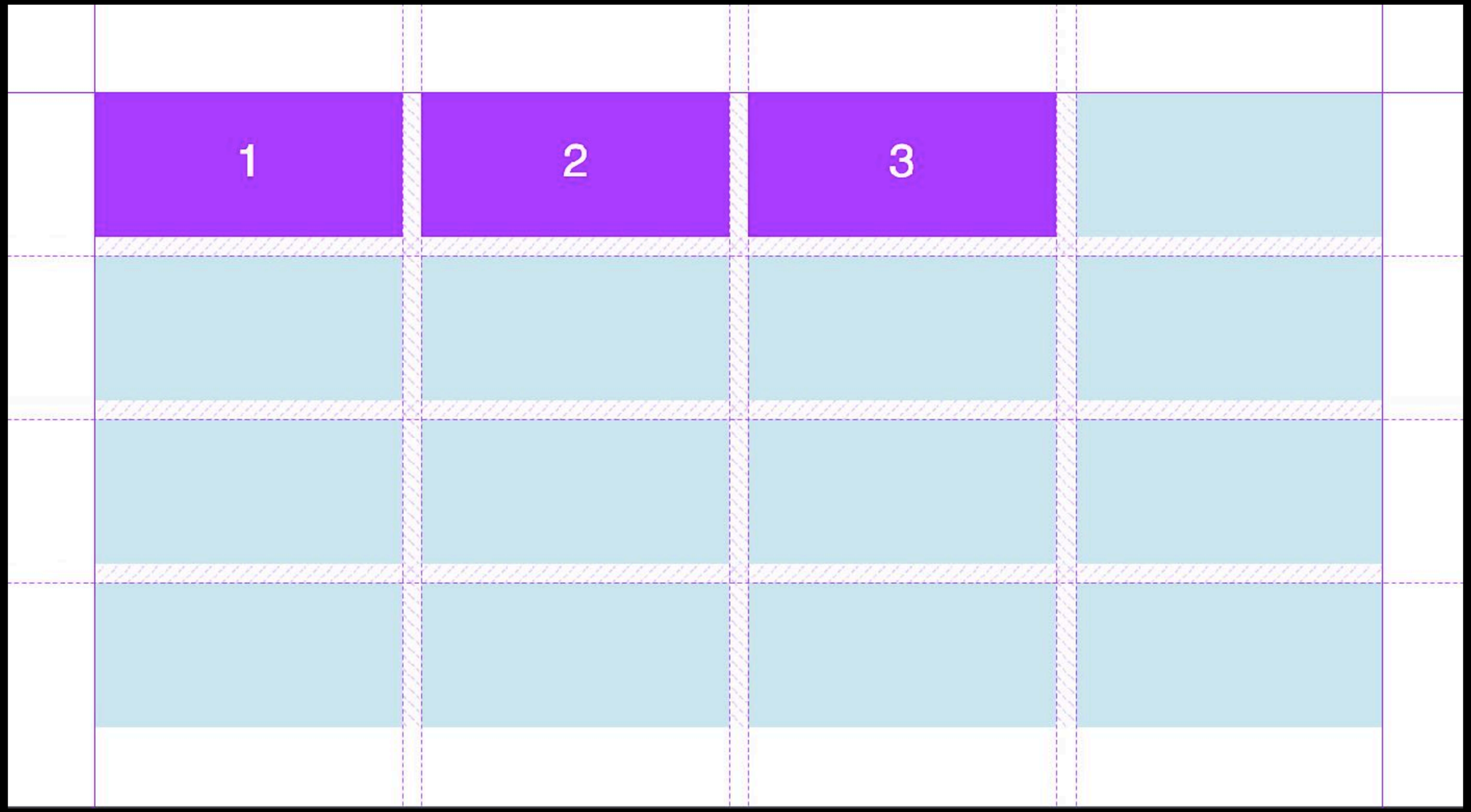
codepen.io/michellebarker/pen/eLZwVg

Placing items

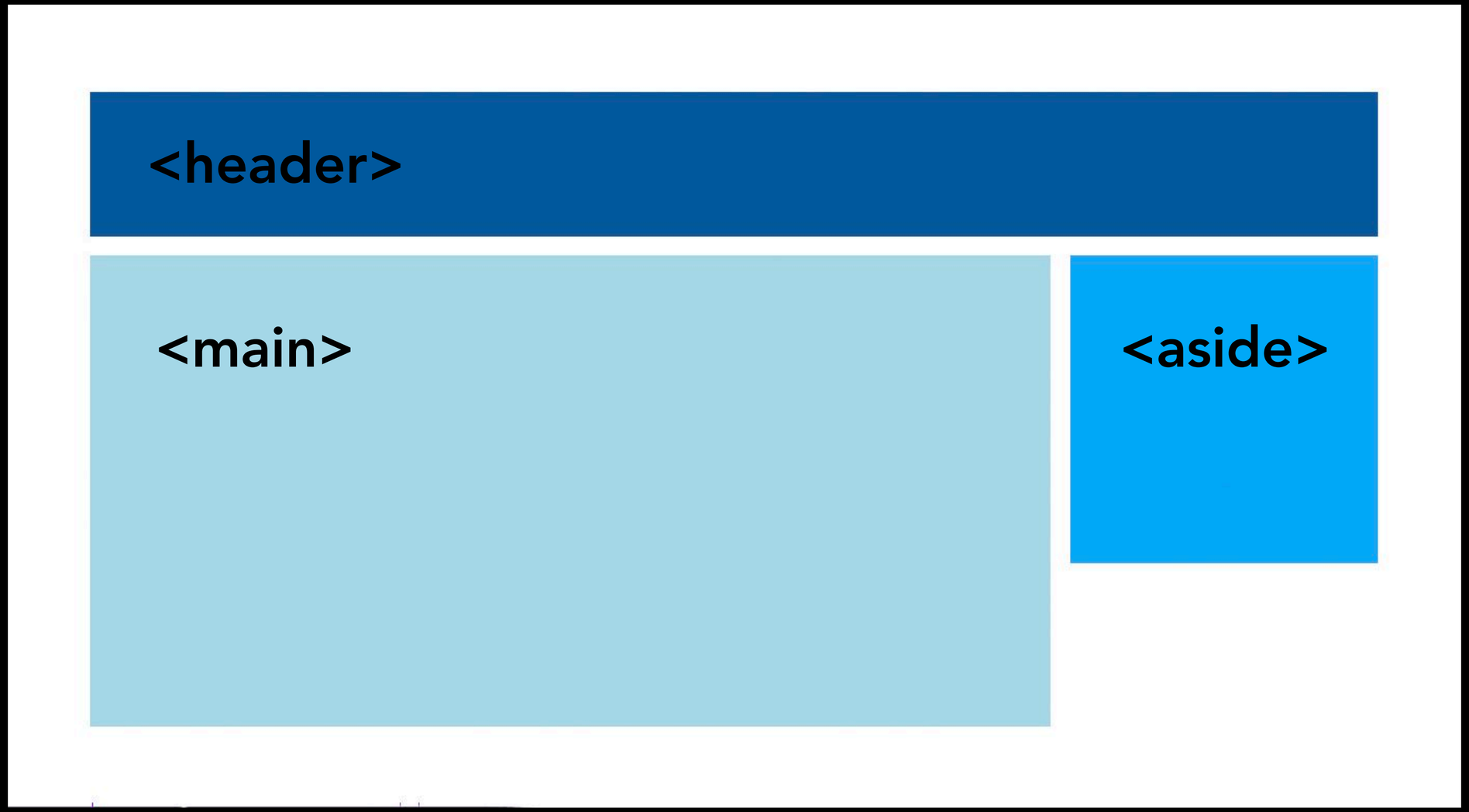
Auto-placement

```
<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



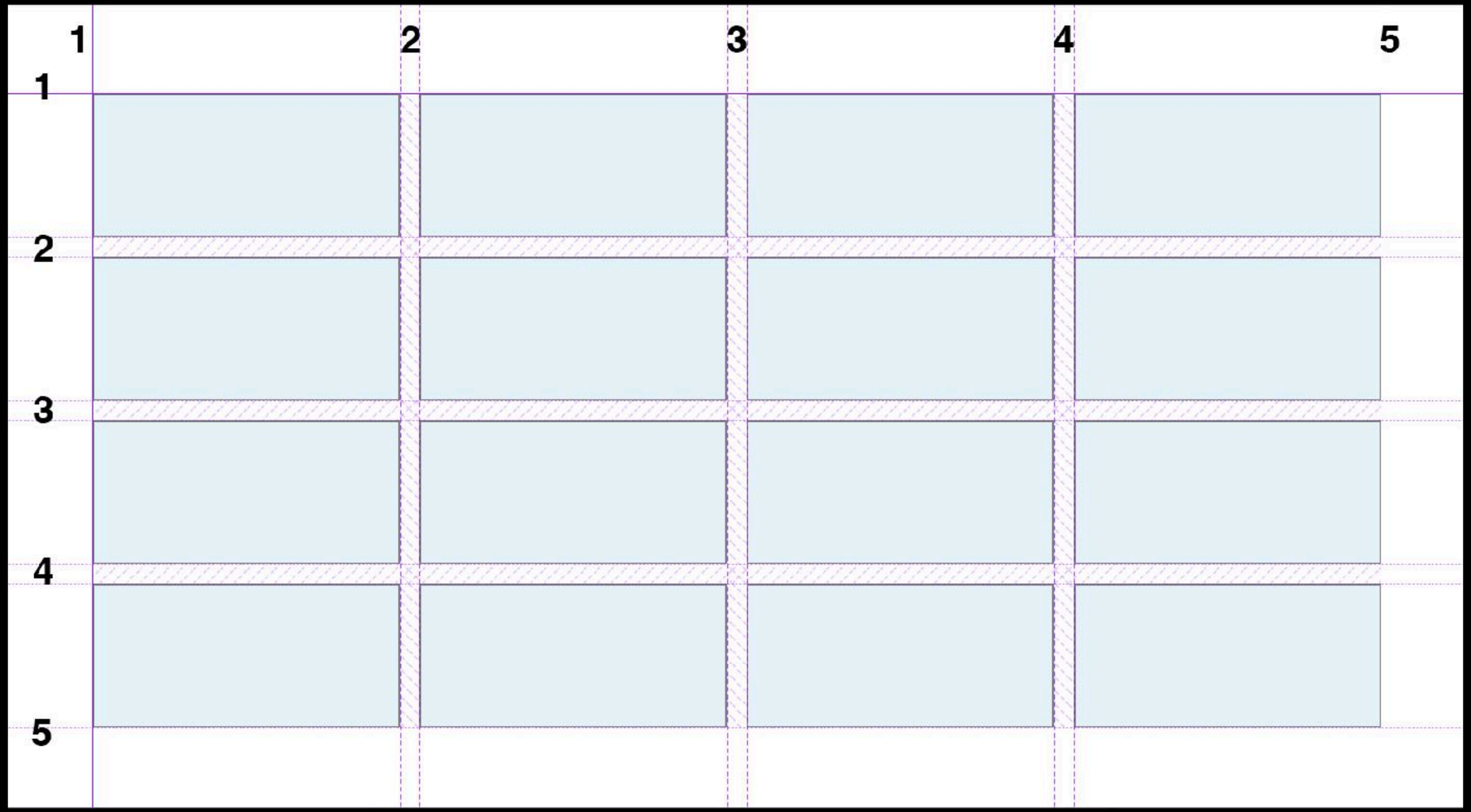
codepen.io/michellebarker/pen/gdrVXP



codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

```
<div class="grid">
  <header></header>
  <main></main>
  <aside></aside>
</div>
```

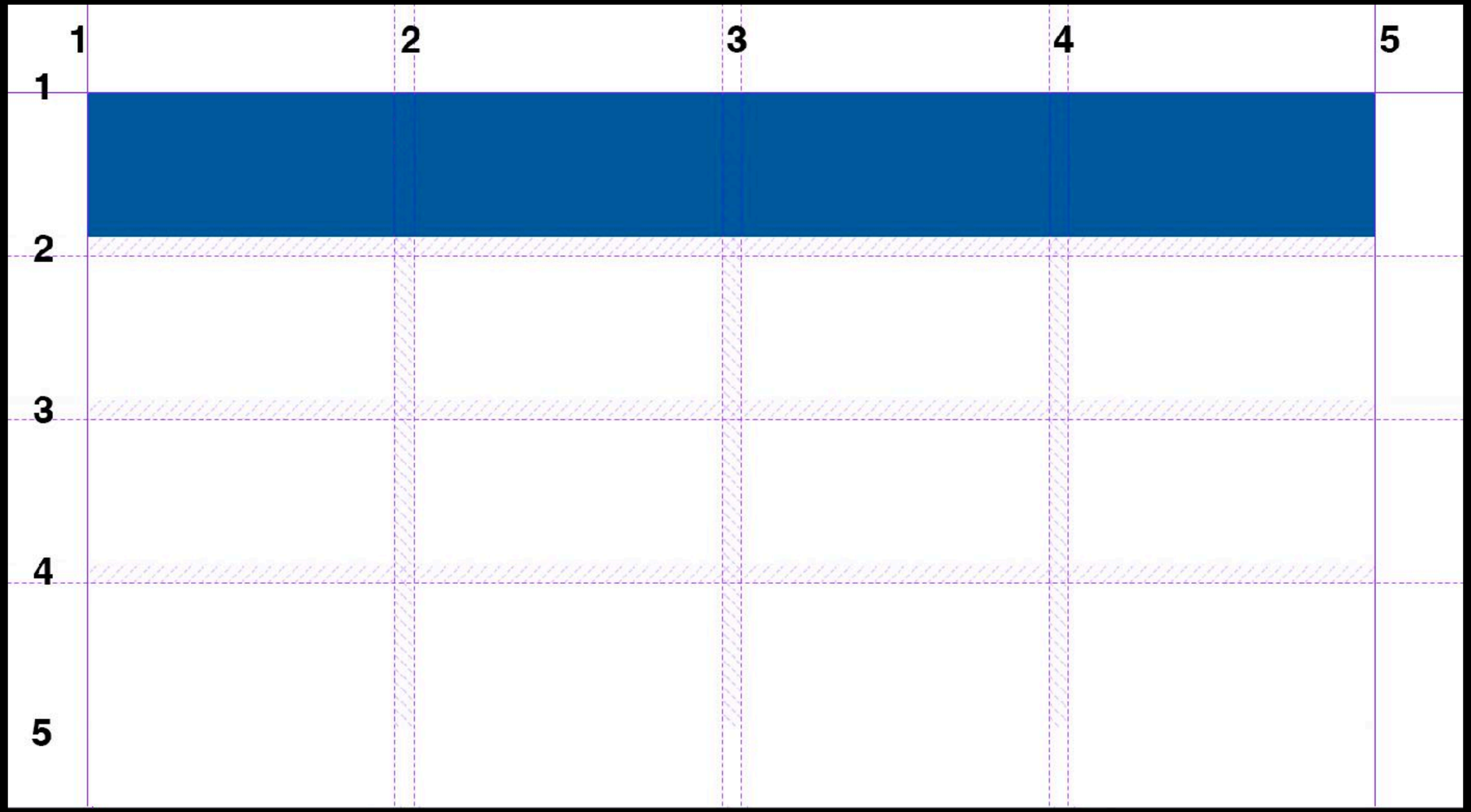


codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

```
header {
  grid-column-start: 1;
  grid-column-end: 5;
}
```

codepen.io/michellebarker/pen/YOqmjP

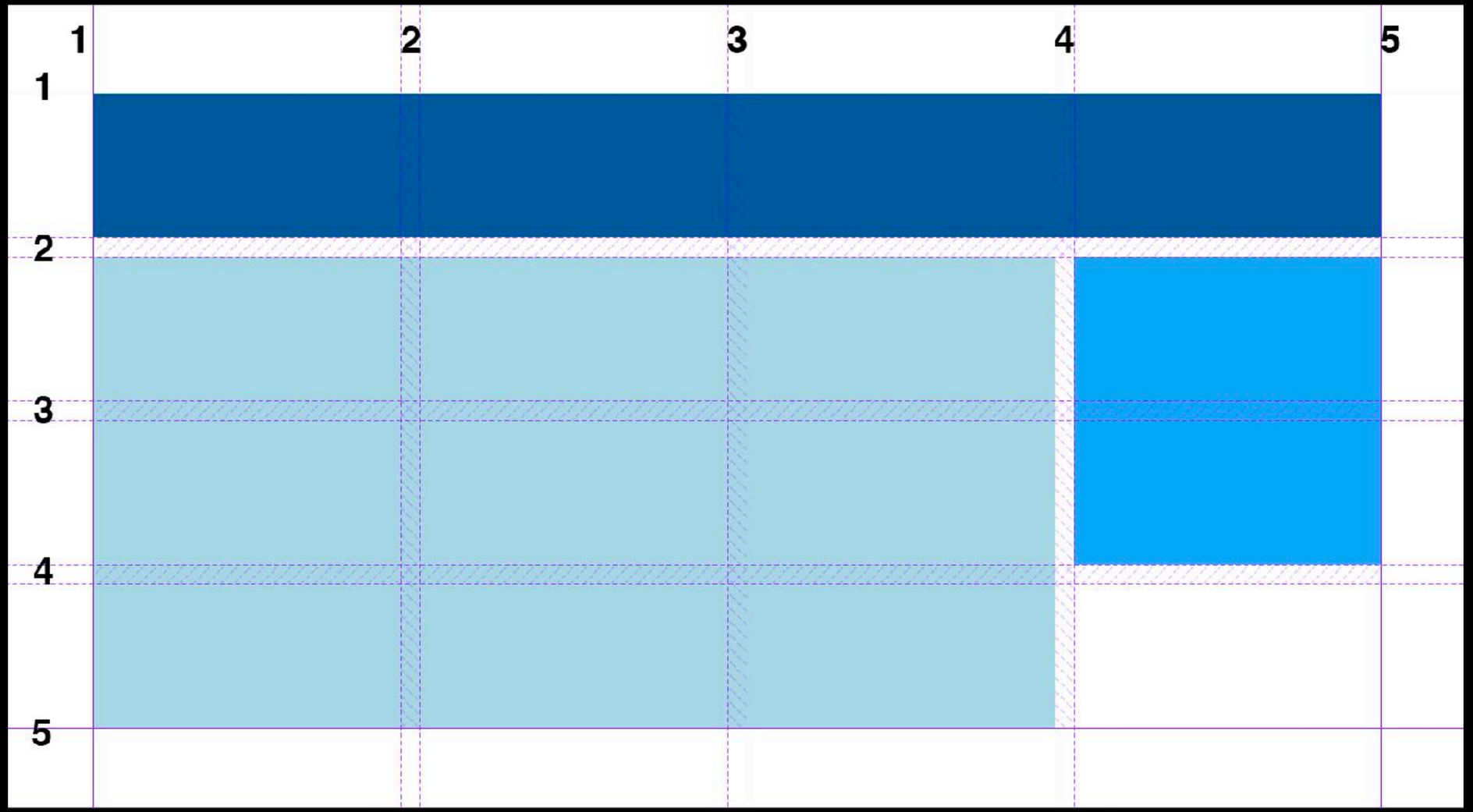


codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

```
main {
  grid-column-start: 1;
  grid-column-end: 4;
  grid-row-start: 2;
  grid-row-end: 5;
}

aside {
  grid-column-start: 4;
  grid-column-end: 5;
  grid-row-start: 2;
  grid-row-end: 4;
}
```

codepen.io/michellebarker/pen/YOqmjP

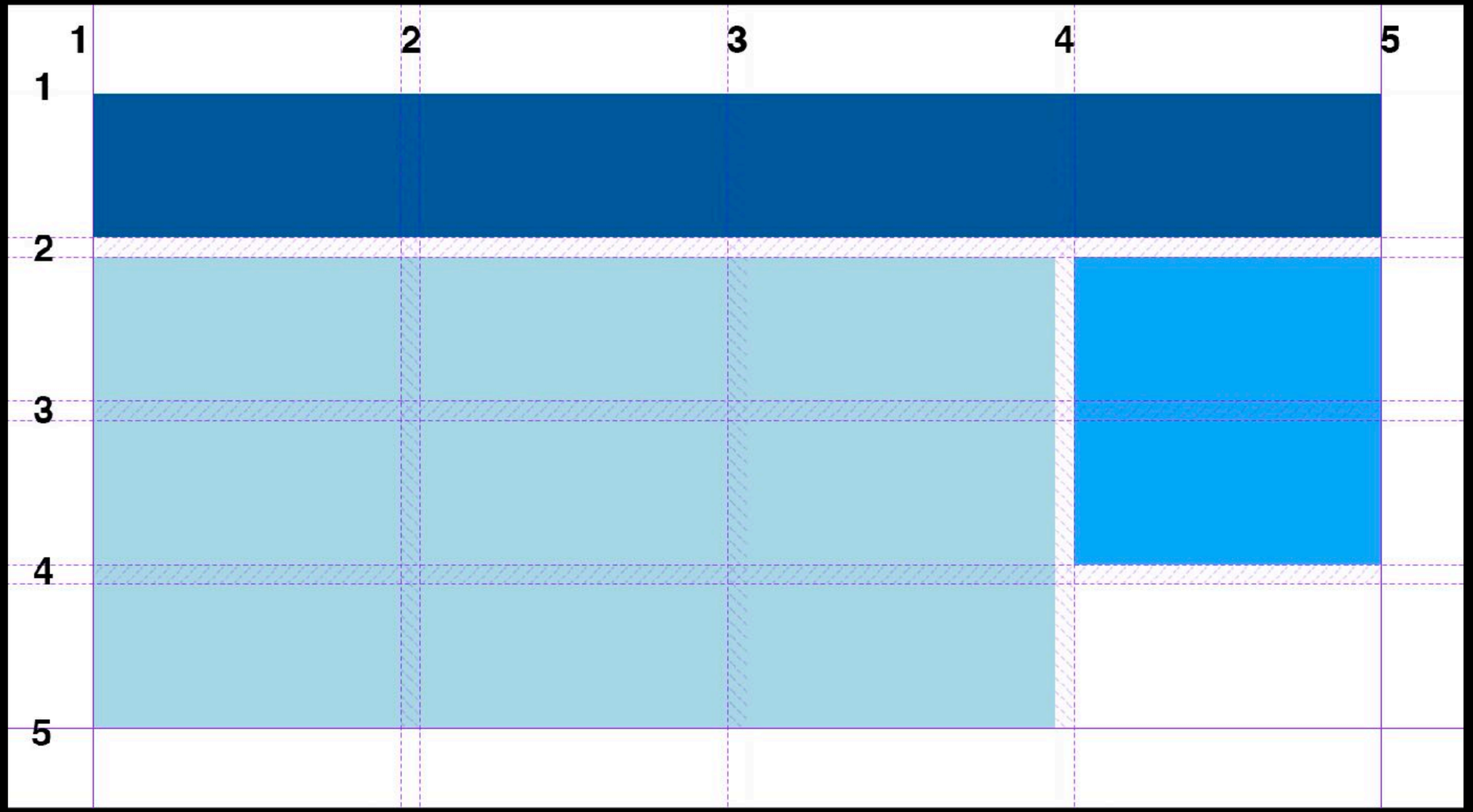


codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

```
header {
  grid-column: 1 / 5;
}

main {
  grid-column: 1 / 4;
  grid-row: 2 / 5;
}

aside {
  // grid-column: 4 / 5; - unnecessary as will be
auto-placed
  grid-row: span 2;
}
```

codepen.io/michellebarker/pen/YOqmjP



codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

Start line / end line

```
main {
  grid-column: 1 / 4;
  grid-row: 2 / 4;
}
```

codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

Span (with auto-placement)

```
main {
  grid-column: span 3;
  grid-row: 2 / 4;
}
```

codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

Start line / span

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

codepen.io/michellebarker/pen/YOQLLZ

Placing items by grid line

Span / end line

```
main {
  grid-column: span 3 / 4;
  grid-row: 2 / 4;
}
```

codepen.io/michellebarker/pen/YOqmjP

Placing items by grid line

Negative grid lines

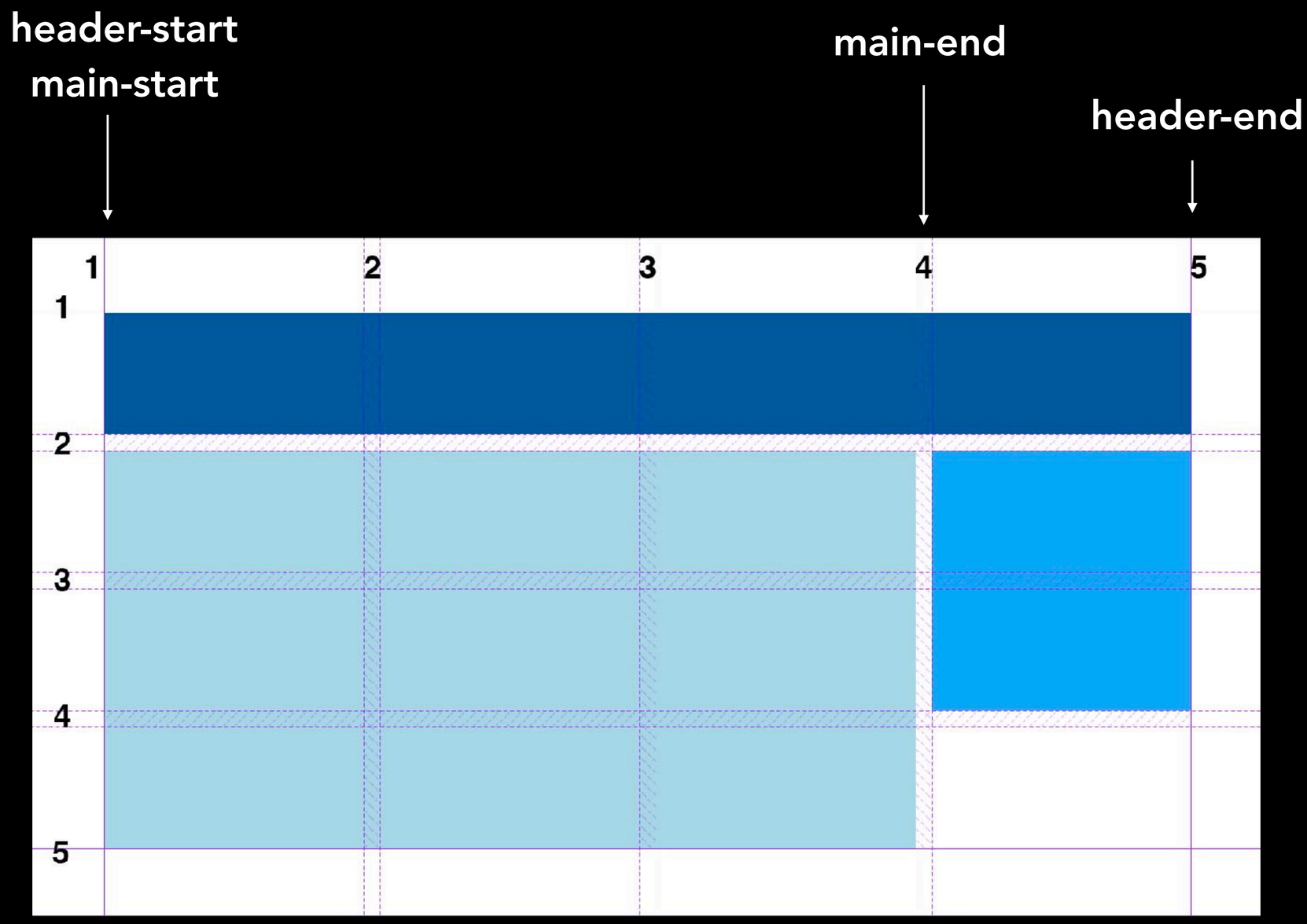
```
main {
  grid-column: 1 / -2;
  grid-row: 2 / 4;
}
```

codepen.io/michellebarker/pen/ReLYwp

Naming lines

```
.grid {
  display: grid;
  grid-template-columns:
   [header-start main-start]
   repeat(3, 1fr)
   [main-end] 1fr [header-end];
  grid-auto-rows: 150px;
}
```

codepen.io/michellebarker/pen/OobJaa



codepen.io/michellebarker/pen/OobJaa

Grid areas

```
header {
  grid-column: header;
}

main {
  grid-column: main
  grid-row: span 3;
}

aside {
  grid-row: span 2;
}
```

codepen.io/michellebarker/pen/rqZevv

grid-template-areas

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

codepen.io/michellebarker/pen/OobJqr

grid-template-areas

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

main {
  grid-area: m;
}

aside {
  grid-area: a;
}
```

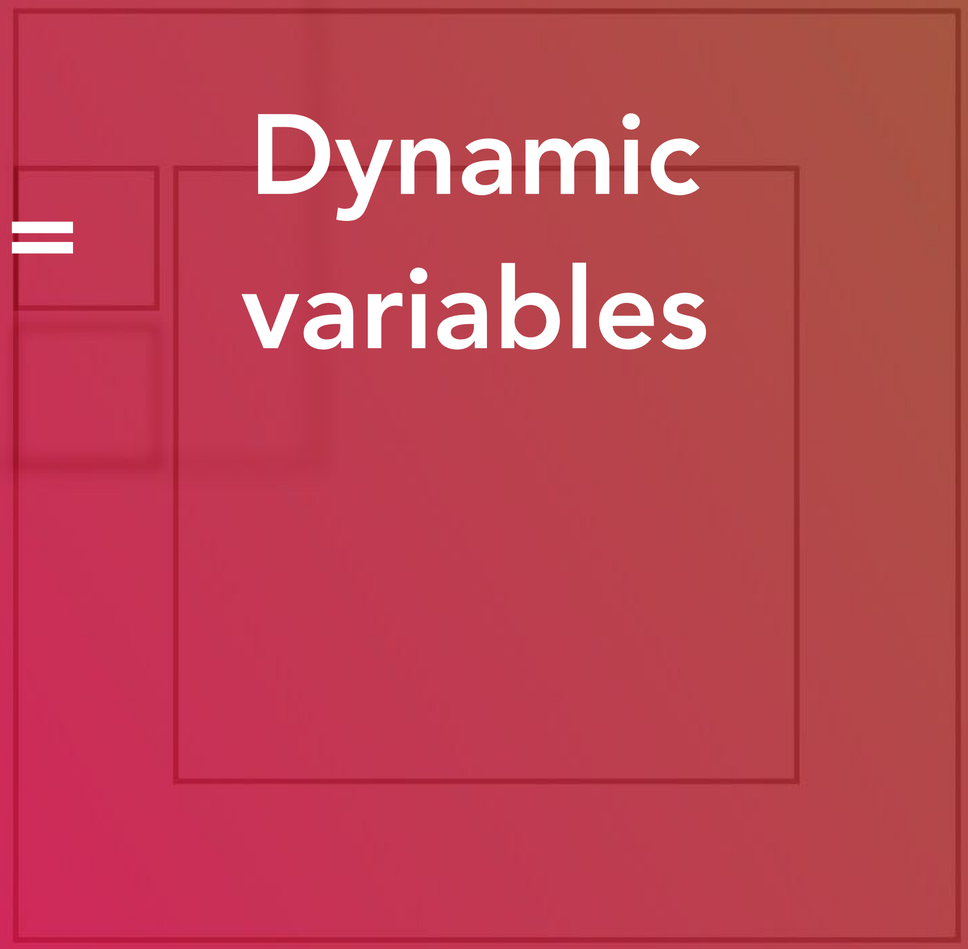
codepen.io/michellebarker/pen/OobJqr

CSS Variables

(a.k.a. Custom Properties)

CSS Variables != Preprocessor Variables

CSS Variables == Dynamic variables



Defining variables

```
:root {
  --bgColor: red;
}
```

Using variables

```
.my-component {
  background-color: var(--bgColor);
}
```

Defaults

```
.my-component {
  background-color: var(--bgColor, orange);
}

.my-component:last-child {
  --bgColor: red;
}
```

codepen.io/michellebarker/pen/MXXQRq

Defaults

```
.box {
  background-color: var(--bgColor, orange);
}

.purple {
  --bgColor: rebeccaPurple;
}
```

codepen.io/michellebarker/pen/rKKdBP

Defaults

```
<div class="box"></div>
```

```
.box {
  background-color:
   orange;
}
```

```
<div class="purple">
  <div class="box"></div>
</div>
```

```
.box {
  background-color:
   rebeccaPurple;
}
```

codepen.io/michellebarker/pen/rKKdBP

CSS Variables
+
CSS Grid

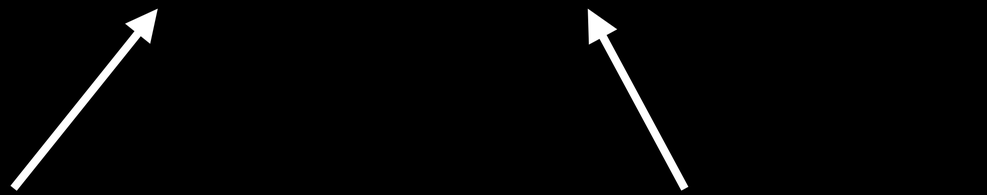
Managing complex grid declarations

Managing complex grid declarations

```
.grid {
  display: grid;
  grid-template-columns:
    [outer-start] minmax(20px, 1fr) [wrapper-start]
    repeat(24, minmax(0, 30px))
    [wrapper-end] minmax(20px, 1fr) [outer-end];
  grid-template-rows:
    [outer-start] 1fr [text-top-end] 40px
    [heading-start] auto [heading-end]
    40px [text-bottom-start] 1fr [outer-end];
  grid-gap: 0 20px;
}
```

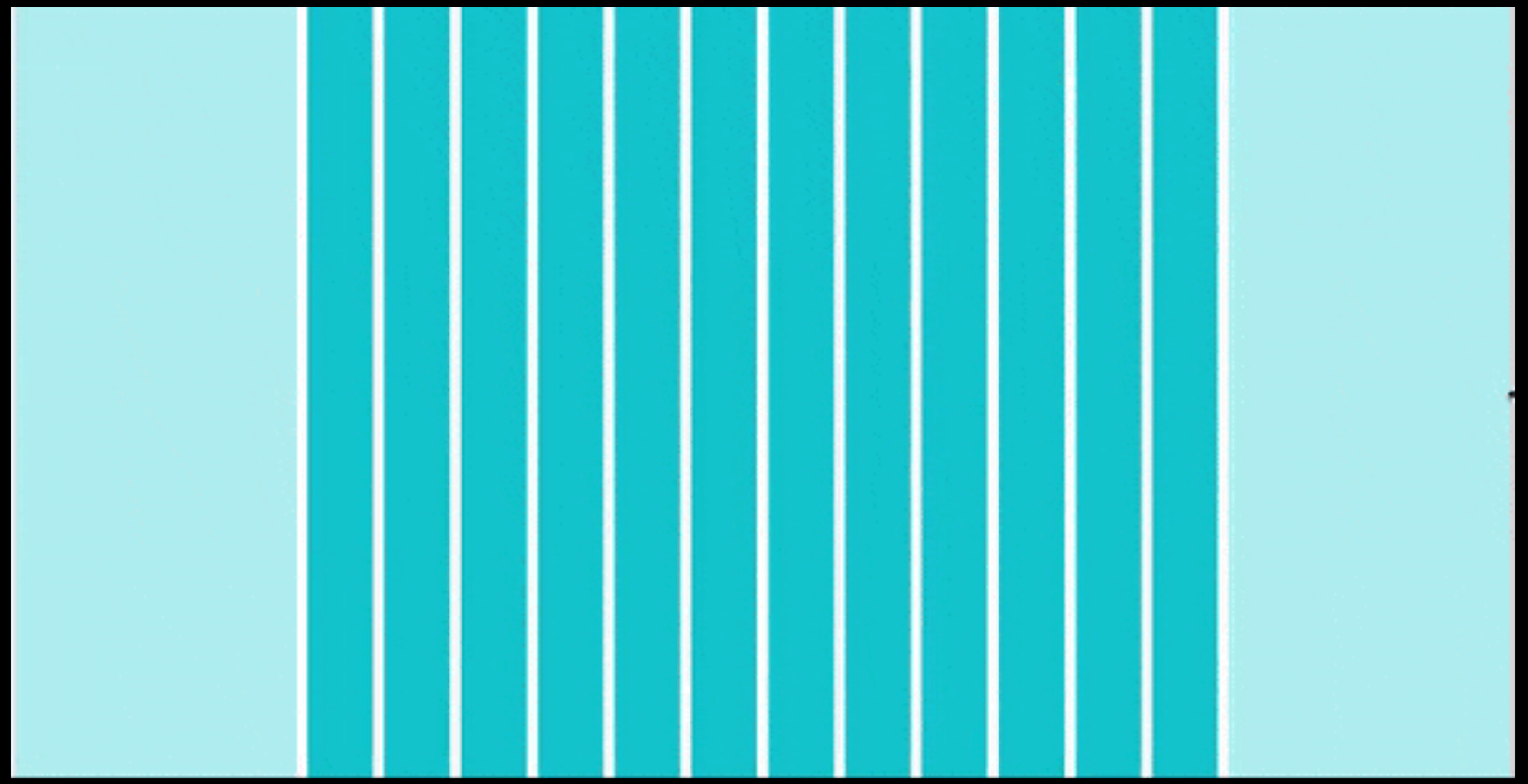
minmax()

```
minmax(20px, 1fr)
```



minmax()

```
minmax(20px, 1fr)
```



```
minmax(0, 60px)
```

codepen.io/michellebarker/pen/wYGMPQ

Managing complex grid declarations

```
.grid {
  display: grid;
  grid-template-columns:
    [outer-start] minmax(20px, 1fr) [wrapper-start]
    repeat(24, minmax(0, 30px))
    [wrapper-end] minmax(20px, 1fr) [outer-end];
  grid-template-rows:
    [outer-start] 1fr [text-top-end] 40px
    [heading-start] auto [heading-end]
    40px [text-bottom-start] 1fr [outer-end];
  grid-gap: 0 20px;
}
```

Managing complex grid declarations

```
:root {
  --colWidth: 30px;
  --gutter: 20px;

  @media (min-width: 1600px) {
    --colWidth: 50px;
    --gutter: 30px;
  }
}
```

Managing complex grid declarations

```
.grid {
  display: grid;
  grid-template-columns:
    [outer-start] minmax(20px, 1fr) [wrapper-start]
    repeat(24, minmax(0, var(--colWidth)))
    [wrapper-end] minmax(20px, 1fr) [outer-end];
  grid-template-rows:
    [outer-start] 1fr [text-top-end] 40px
    [heading-start] auto [heading-end]
    40px [text-bottom-start] 1fr [outer-end];
  grid-gap: 0 var(--gutter);
}
```

codepen.io/michellebarker/pen/NzYbVO

Managing component variants

Managing component variants

```
<article class="grid">
  <figure class="grid__img"></figure>
  <div class="grid__text"></div>
</article>
```

codepen.io/michellebarker/pen/XBPMZZ

Managing component variants

codepen.io/michellebarker/pen/XBPMZZ

Managing component variants

```
.grid {
  display: grid;
  grid-template-columns:
    minmax(20px, 1fr)
    repeat(12, minmax(0, 70px))
    minmax(20px, 1fr);
  grid-gap: 20px;
}
```

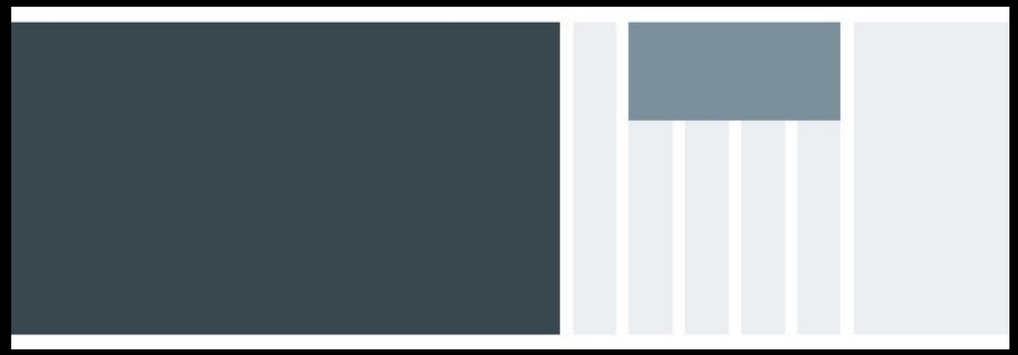
codepen.io/michellebarker/pen/XBPMZZ

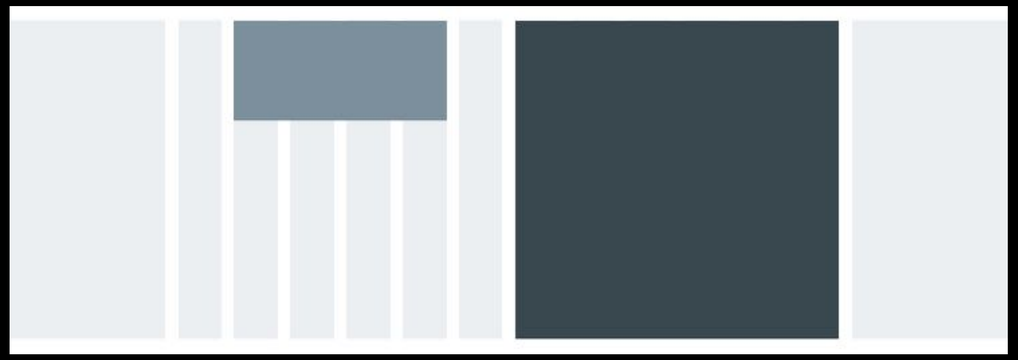
Managing component variants

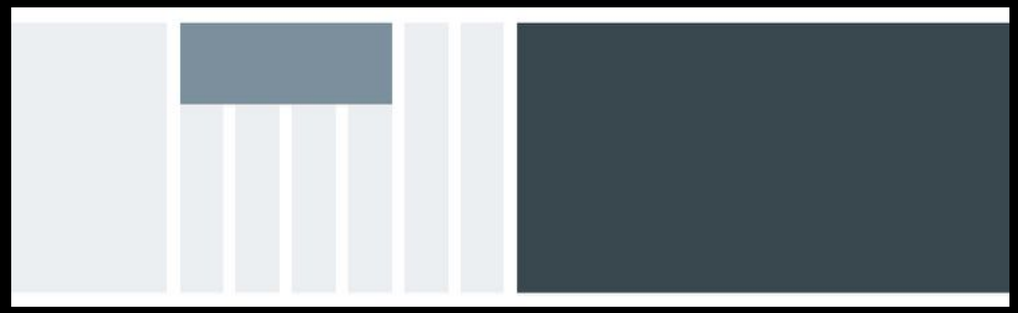
```
.grid__img {
  grid-column: 2 / 8;
}

.grid__text {
  grid-column: 10 / span 4;
}
```

codepen.io/michellebarker/pen/XBPMZZ

Managing component variants







codepen.io/michellebarker/pen/XBPMZZ

Managing component variants

```
.grid__img {
  grid-column:
    var(--imgStart, 2) / var(--imgEnd, 8);
  grid-row: 1;
}

.grid__text {
  grid-column:
    var(--textStart, 10) / span 4;
  grid-row: 1;
}
```

codepen.io/michellebarker/pen/XBPMZZ

Managing component variants

```
.grid--2 {
  --textStart: 3;
  --imgStart: 8;
  --imgEnd: -2;
}

.grid--3 {
  --imgStart: 1;
}

.grid--4 {
  --textStart: 2;
  --imgStart: 8;
  --imgEnd: -1;
}
```

codepen.io/michellebarker/pen/XBPMZZ

CSS Variables
+
Javascript

Get property value:

```
getComputedStyle(element).getPropertyValue("--myVariable")
```

(Returns a string)

Set property value:

```
element.style.setProperty("--myVariable", 4)
```

Demo

CSS Grid background generator

codepen.io/michellebarker/pen/xabrLv

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

The Future of Grid...

Thank you for listening

@mbarker_84 / @CSSInRealLife