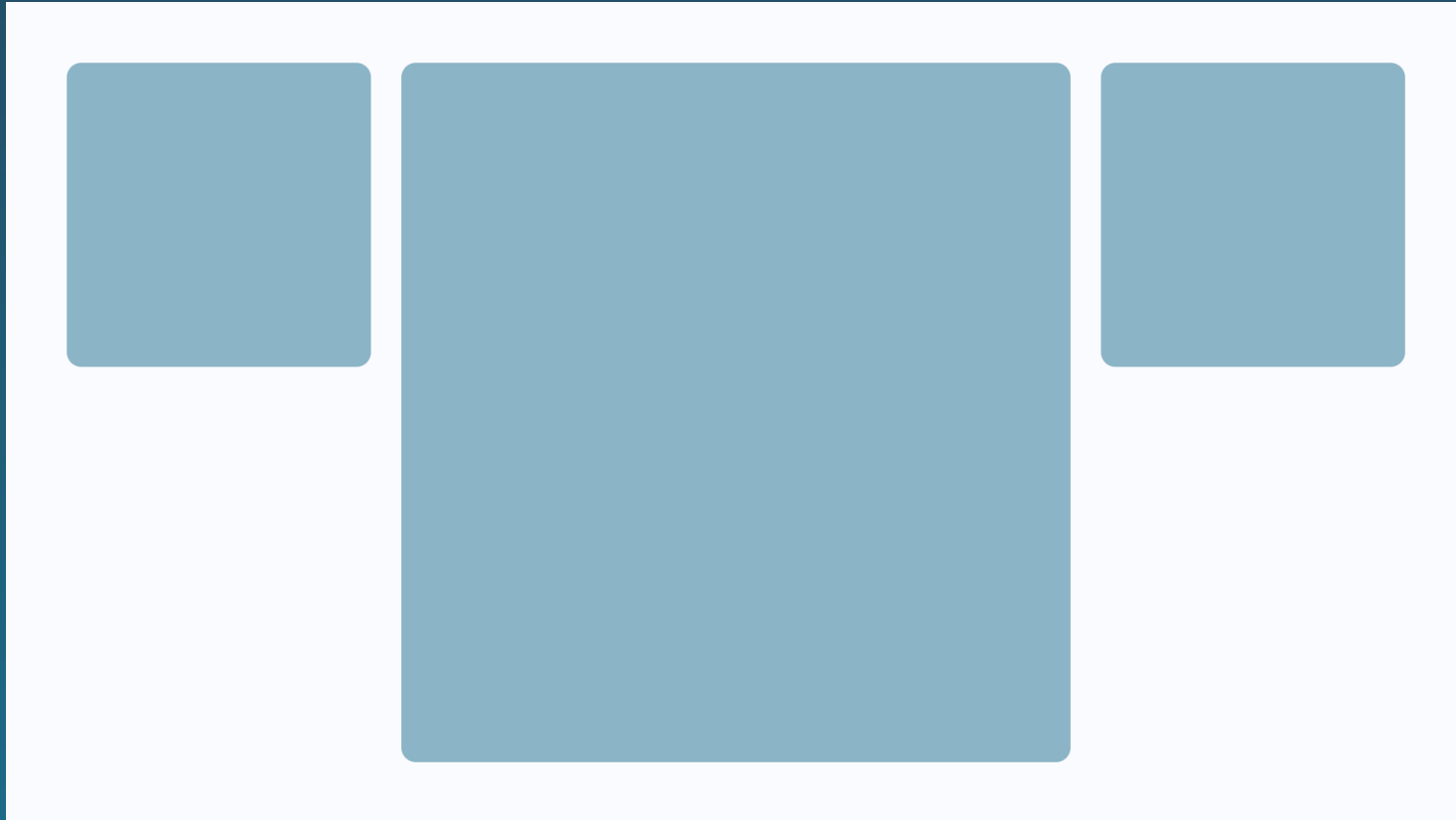


CSS Layout from the inside out

Rachel Andrew, State of the Browser, October, 2021

The Holy Grail

The Holy Grail

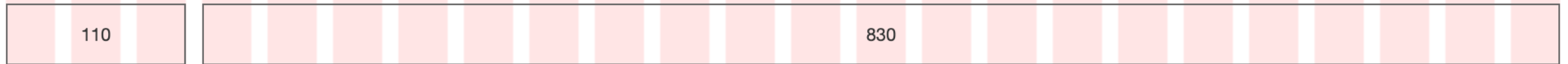
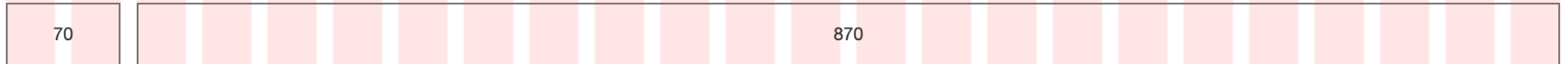


target / context = result

Fluid Grids, Ethan Marcotte, A List Apart, 2009.

960 Grid System

24 Column Grid



A float-based grid

Size things with % and push them around to look like a grid.

This is box a.

This is box b.

This is box c.
With two lines of text.

```
.box {  
  float: left;  
}  
  
.a { width: 25%; }  
  
.b { width: 16.66%; }  
  
.c {width: 33.33%; }
```

The biggest problems in (float-based) web design?

Clearing floats and getting equal height columns.

A flex-based grid

Size things with % and push them around to look like a grid (but with equal height columns)

This is box a.

This is box b.

This is box c.
With two lines of text.

```
.wrapper {  
  display: flex;  
}  
  
.box {  
  flex: none;  
}  
  
.a { width: 25%; }  
  
.b { width: 16.66%; }  
  
.c {width: 33.33%; }
```

The biggest problem in (flex-based) web design?

Flexbox being weird and hard.

This is box a.

This is box b.

This is box c.
With two lines of text.

```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(12,minmax(0,1fr));  
}  
  
.a { grid-column: auto / span 3; }  
  
.b { grid-column: auto / span 2; }  
  
.c {grid-column: auto / span 4; }
```

Has grid made flexbox redundant?

Cat

Pleasing fungus beetle

Pigeon

Rhinoceros

Tasselled Wobbegong

Elk

Elephant

Cat

Pleasing
fungus
beetle

Pigeon

Rhinoceros

Tasselled
Wobbegong

Elk

Elephant

Cat

Pleasing fungus beetle

Pigeon

Rhinoceros

Tasselled Wobbegong

Elk

Elephant

Cat

Pleasing
fungus
beetle

Pigeon

Rhinoceros

Tasselled
Wobbegong

Elk

Elephant

```
.grid {  
  display: grid;  
  grid-template-columns:  
    repeat(6, minmax(0, 1fr));  
}
```

Cat

Pleasing
fungus beetle

Pigeon

Rhinoceros

Tasselled
Wobbegong

Elk

Elephant

$\min\max(0, 1fr)$

Cat

Pleasing
fungus
beetle

Pigeon

Rhinoceros



Elk

Elephant

`minmax(auto, 1fr)`

Heading decorations

Heading decorations

```
h1 {  
  display: grid;  
  grid-template-columns: 1fr auto 1fr;  
}  
  
h1::before, h1::after {  
  content: "";  
  border-top: 1px solid var(--charcoal);  
  align-self: center;  
}
```


Aspect Ratios

The “padding hack”

<https://css-tricks.com/aspect-ratio-boxes/>

aspect-ratio: 16/9;

16x9

4x3

16x9

4x3

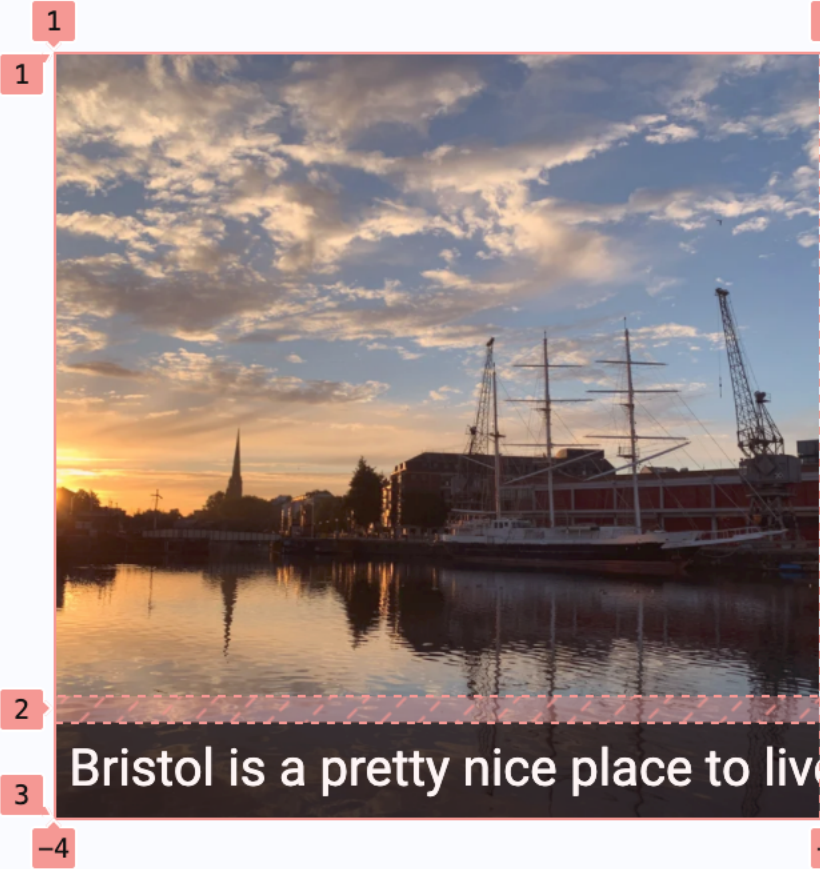
```
.wrapper {  
  display: grid;  
  grid-template-columns: 200px 350px;  
}
```

```
.box16x9 {  
  aspect-ratio: 16/9;  
}
```

```
.box4x3 {  
  aspect-ratio: 4/3;  
}
```



Bristol is a pretty nice place to live.



1fr



1fr



1fr

CSS Containment

This box and children are independent from the rest of the layout.


```
.sidebar {  
  contain: layout;  
}
```

Container Queries

Design components that can react to the available space in their container.



Bristol is a pretty nice place to live.



Bristol is a pretty nice place to live.

Try out Container Queries

In `chrome://flags` enable “CSS Container Queries”.

```
.sidebar {  
  container-type: inline-size;  
}
```

Using container-type creates a containment context.

This also turns on layout, style, and inline-size containment, in the same way as using the contain property does.

```
.sidebar {  
  container-type: inline-size;  
  container-name: sidebar;  
}
```

```
@container (min-width: 400px) {  
  .card {  
    display: grid;  
    grid-template-columns: 1fr 2fr;  
  }  
}
```



```
@container sidebar (min-width: 400px) {  
  .card {  
    display: grid;  
    grid-template-columns: 1fr 2fr;  
  }  
}
```

One

Two

Three

Four

One

Two

Three

Four

```
.wrapper > div {  
  container-type: inline-size;  
}  
  
@container (min-width: 220px) {  
  .box {  
    background-color: var(--blue-sapphire);  
  }  
}
```



Bristol is a pretty nice place to live.

```
.fig-panel {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
}
```



Bristol is a pretty nice place to live.



```
@container (min-width: 700px) {  
  grid-template-columns: repeat(3, 1fr);  
}
```

This seems so simple!
Why did it take so long?

RenderingNG deep-dive: LayoutNG

Published on Friday, October 8, 2021



Ian Kilpatrick
Layout tech lead



Koji Ishi
Layout engineer

This post is a part of a series on the Chromium rendering engine. Check out the rest of the series to learn more about [RenderingNG](#), [the architecture](#), [key data structures](#), and [VideoNG](#).

I'm Ian Kilpatrick, an engineering lead on the Blink layout team, along with Koji Ishii. Before working on the Blink team, I was a front-end engineer (before Google had the role of "front-end engineer"), building features within Google Docs, Drive, and Gmail. After around five years in that role I took a large gamble switching to the Blink team, effectively learning C++ on the job, and attempting to ramp up on the massively complex Blink codebase. Even today, I only understand a relatively small portion of it. I'm grateful for the time given to me during this period. I was comforted by the fact a lot of "recovering front-end engineers" made the transition to being a "browser engineer" before me.

Table of contents

[A 30,000 foot view of layout engine architectures](#)

Types of layout bugs

[Correctness](#)

[Under-invalidation](#)

[Hysteresis](#)

[Over-invalidation and performance](#)

[In summary](#)

```
.sidebar {  
  container: inline-size / sidebar;  
}
```



```
.feature {  
  container-type: block-size;  
  container-name: feature;  
}
```

Black bean chili dip with tomato and basil zesty
tofu pad thai parsley street style Thai basil tacos
chili guacamole one bowl elderberry Italian
linguine puttanesca mediterranean dragon fruit
peanut butter crunch blackberries broccoli
lemonade zest green bowl veggie burgers eating
together tabasco pepper couscous bruschetta
four-layer citrusy miso dressing oranges.

Black bean chili dip with tomato and basil zesty
tofu pad thai parsley street style Thai basil tacos
chili guacamole one bowl elderberry Italian
linguine puttanesca mediterranean dragon fruit
peanut butter crunch blackberries broccoli
lemonade zest green bowl veggie burgers eating
together tabasco pepper couscous bruschetta
four-layer citrusy miso dressing oranges.

Container Queries are coming

But they probably aren't the solution to all your problems.

Remember that flexbox is designed for flexible things

A bunch of different sized items, within an unknown container size,
dealt with as a group.

Use intrinsic and range sizing methods with grid

`minmax()`, `min-content`, `max-content`, `auto`, and `fit-content()`

But, do get excited for Container Queries!

Try it out! If you spot any issues or think of a potential enhancement, let the
CSS Working Group know:

Thank you!

Find me @rachelandrew

Slides at: <https://noti.st/rachelandrew>