LET'S LEARN

CSSGRID

Hi there!

I’m Tim Smith

The Bold Report

ANYTHIN’GOES

BLOC

Agenda

Agenda

1. Problems we’ve had with layout

Agenda

1. Problems we’ve had with layout
2. What is CSS Grid and how does it solve these problems?

Agenda

1. Problems we’ve had with layout
2. What is CSS Grid and how does it solve these problems?
3. Why do I need to learn this?

Agenda

1. Problems we've had with layout
2. What is CSS Grid and how does it solve these problems?
3. Why do I need to learn this?
4. Resources

A History of **Layout Issues**

Floats

Floats Suck

CSS Frameworks

CSS Frameworks

```html
<div class="small-12 medium-6 large-8"></div>
```

~~Separation of Concerns~~

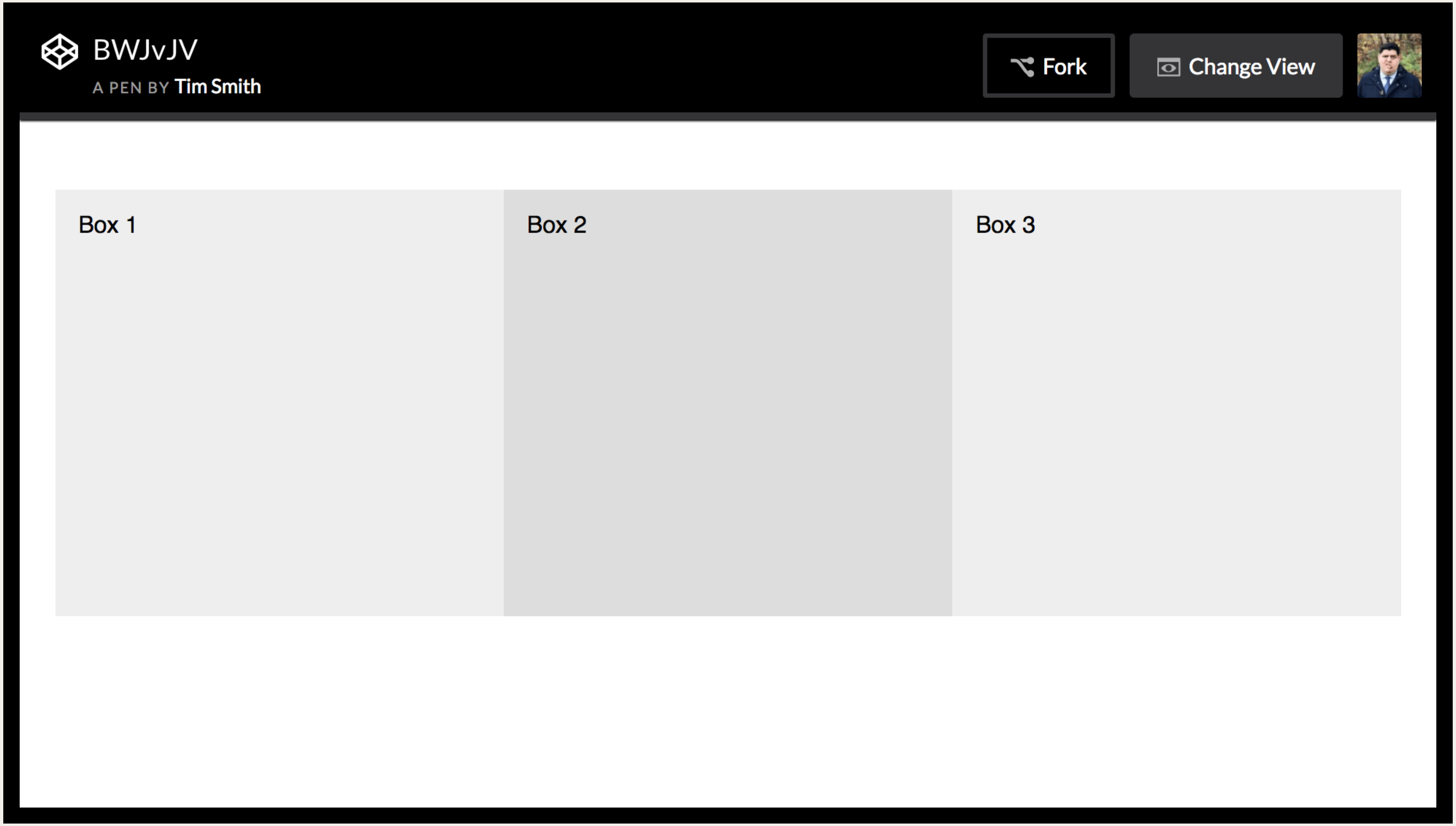
Floats

Aren't Bad

They’re Not for

Complex Layouts

Why?



Box 1

Box 2

Box 3

Flexbox

Flexbox is Flexible

Main Column and Sidebar Layout

```
<aside>Aside</aside>
<main>Main Content</main>
```

Main Column and Sidebar Layout

```
<aside>Aside</aside>
<main>Main Content</main>
```

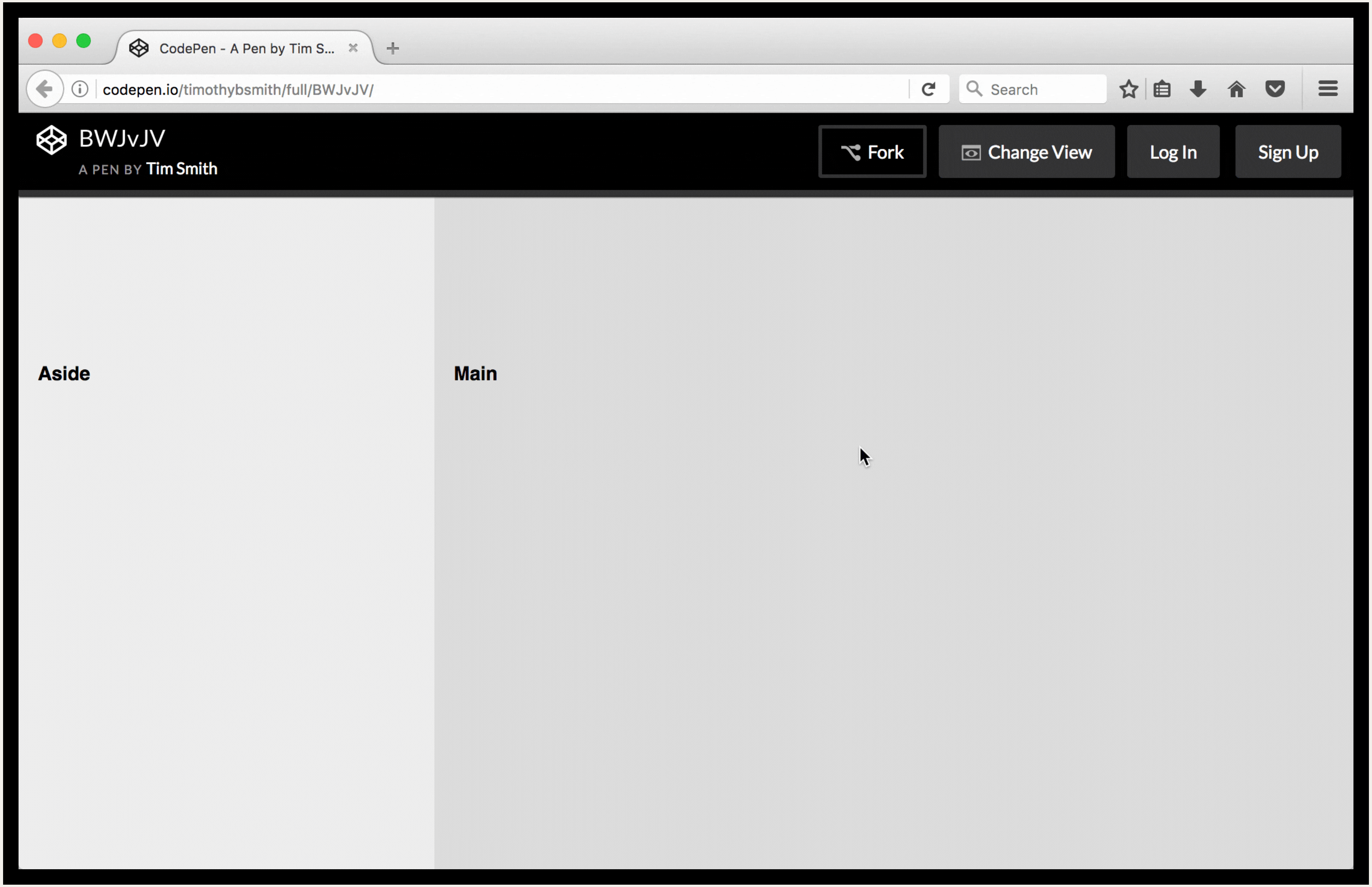
And our CSS.

Main Column and Sidebar Layout

```
<aside>Aside</aside>
<main>Main Content</main>
```

And our CSS.

```
body {
  display: flex;
}

aside {
  background-color: #eee;
  flex: 1 1 30%;
}

main {
  background-color: #ddd;
  flex: 1 1 70%;
}
```



The two columns are completely flexible.

Flexbox Card Layout

```
<ul class="boxes">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
</ul>
```

Flexbox Card Layout

```
<ul class="boxes">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
</ul>
```

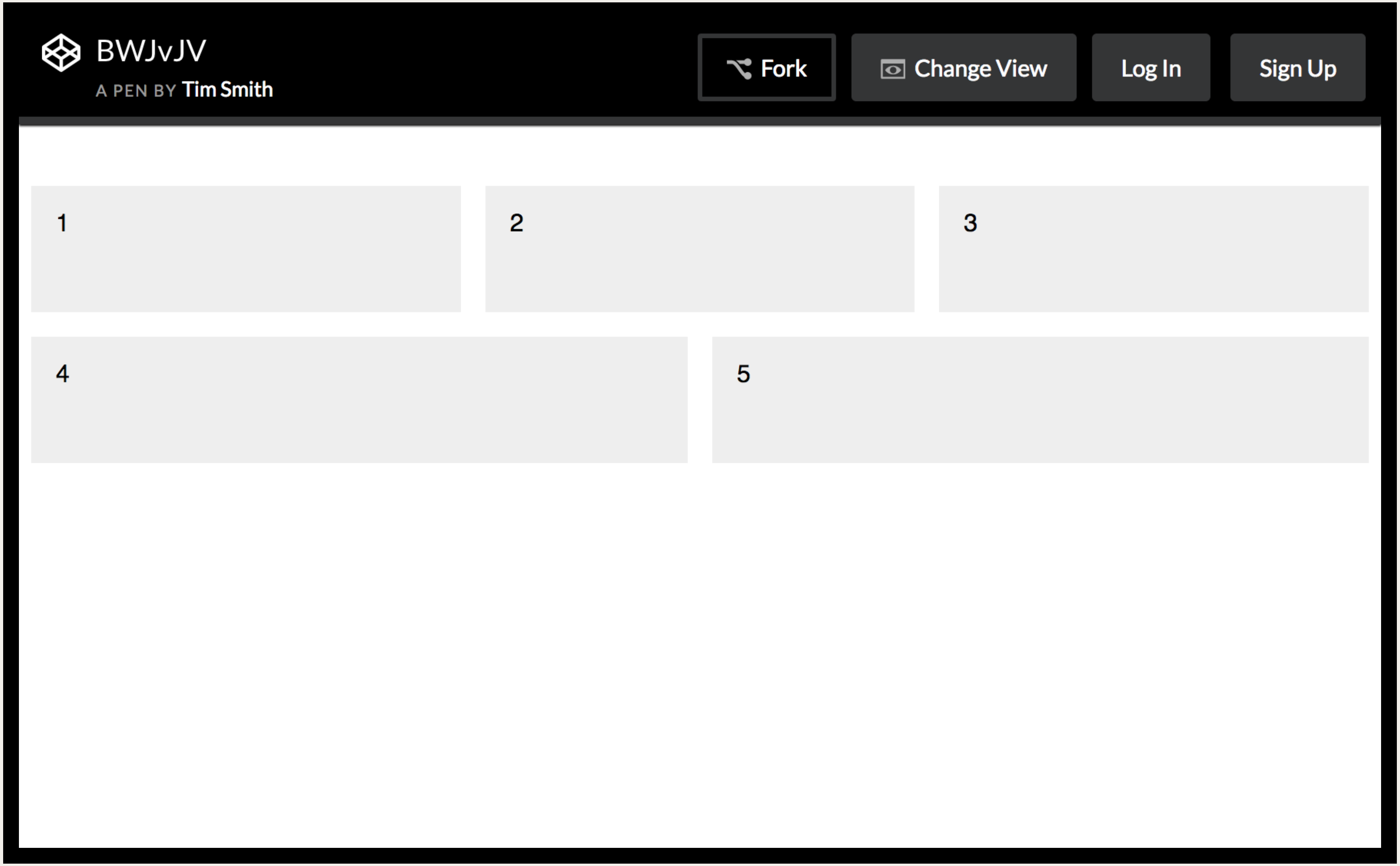
And now our CSS.

Flexbox Card Layout

```
<ul class="boxes">
  <li>1</li>
  <li>2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
</ul>
```

And now our CSS.

```
.boxes {
  display: flex;
  flex-wrap: wrap;
}

.boxes li {
  flex: 1 1 30%;
  background: #eee;
  margin: 0 .5rem 1rem;
}
```



This is not what I want!

Math

You Need Stuff Like This

```css
li:nth-child(2n+2) {
  margin-right: 0;
}
```

You Need Stuff Like This

```css
li:nth-child(2n+2) {
  margin-right: 0;
}
```

Oh wait! What about bigger screens?

You Need Stuff Like This

```css
li:nth-child(2n+2) {
  margin-right: 0;
}
```

Oh wait! What about bigger screens?

```css
@media screen and (min-width: 40em) {
  li:nth-child(2n+2) {
    margin-right: 1.5rem;
  }
  li:nth-child(3n+3) {
    margin-right: 0;
  }
}
```

Flexbox

Is Awesome

Vertical Align!

We're

Using Flexbox Wrong

CSS Grid

to the Rescue

CSS Grid is Not a **Float Killer**

CSS Grid is Not a **Flexbox Killer**

You Don't Need

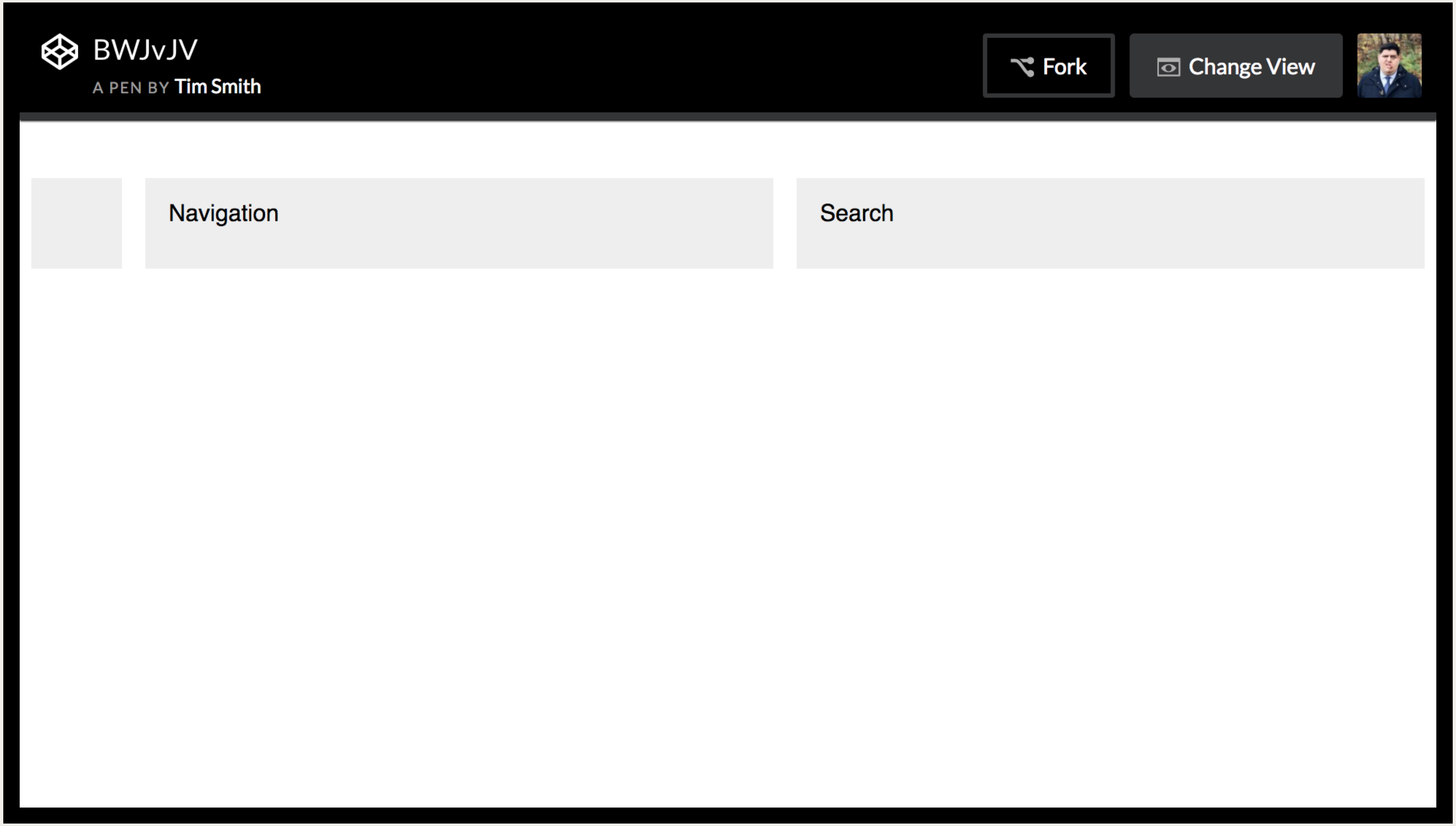
a Framework

CSS Grid

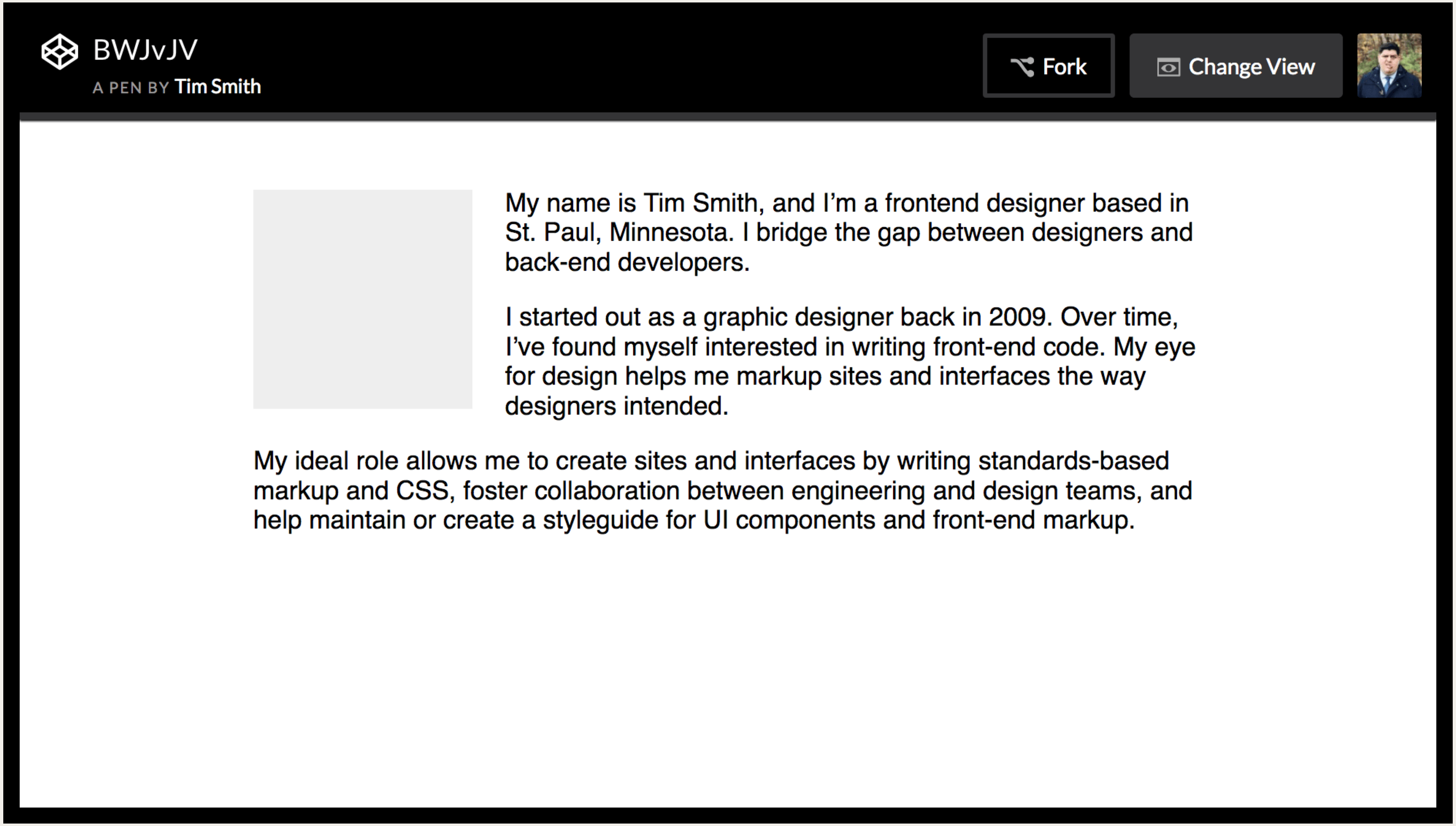
Is the Framework

CSS Grid is

Friendly



Great Use of Flexbox



My name is Tim Smith, and I'm a frontend designer based in St. Paul, Minnesota. I bridge the gap between designers and back-end developers.

I started out as a graphic designer back in 2009. Over time, I've found myself interested in writing front-end code. My eye for design helps me markup sites and interfaces the way designers intended.

My ideal role allows me to create sites and interfaces by writing standards-based markup and CSS, foster collaboration between engineering and design teams, and help maintain or create a styleguide for UI components and front-end markup.

Great Use of Floats

CSS Grid

What Is It Good For?

> “My super simple theory at the moment — if you are putting widths on flex items, you are doing it wrong. Use Grid. (Let’s see if that holds.) —Jen Simmons

CSS Grid

Basics

```
display: grid;
```

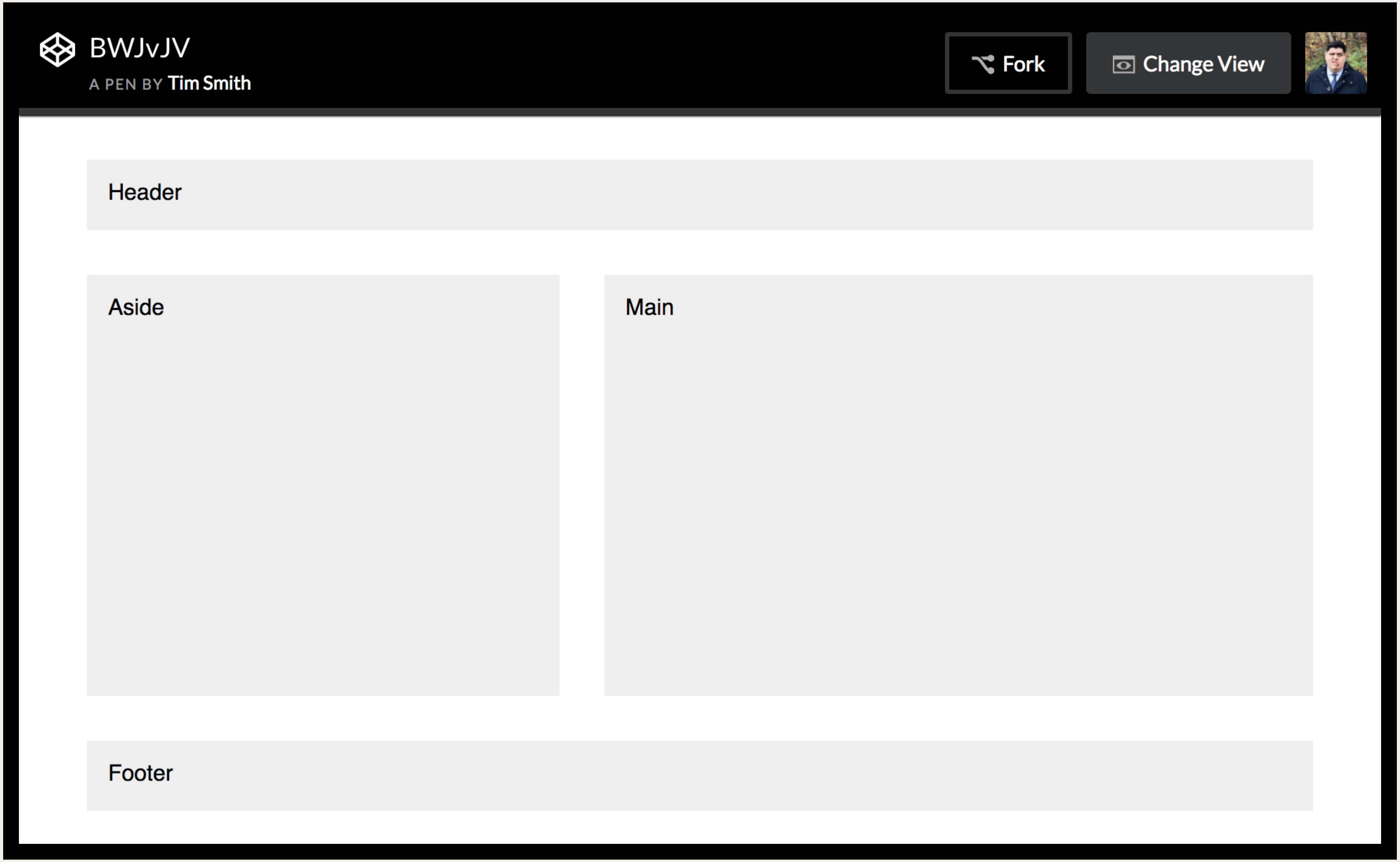
`grid-template-columns`

fr

`grid-gap`

Let’s See It in Action

Holy Grail Layout



Header

Aside

Main

Footer

Set Up

```html
<header>Header</header>
<aside>Aside</aside>
<main>Main Content</main>
<footer>Footer</footer>
```

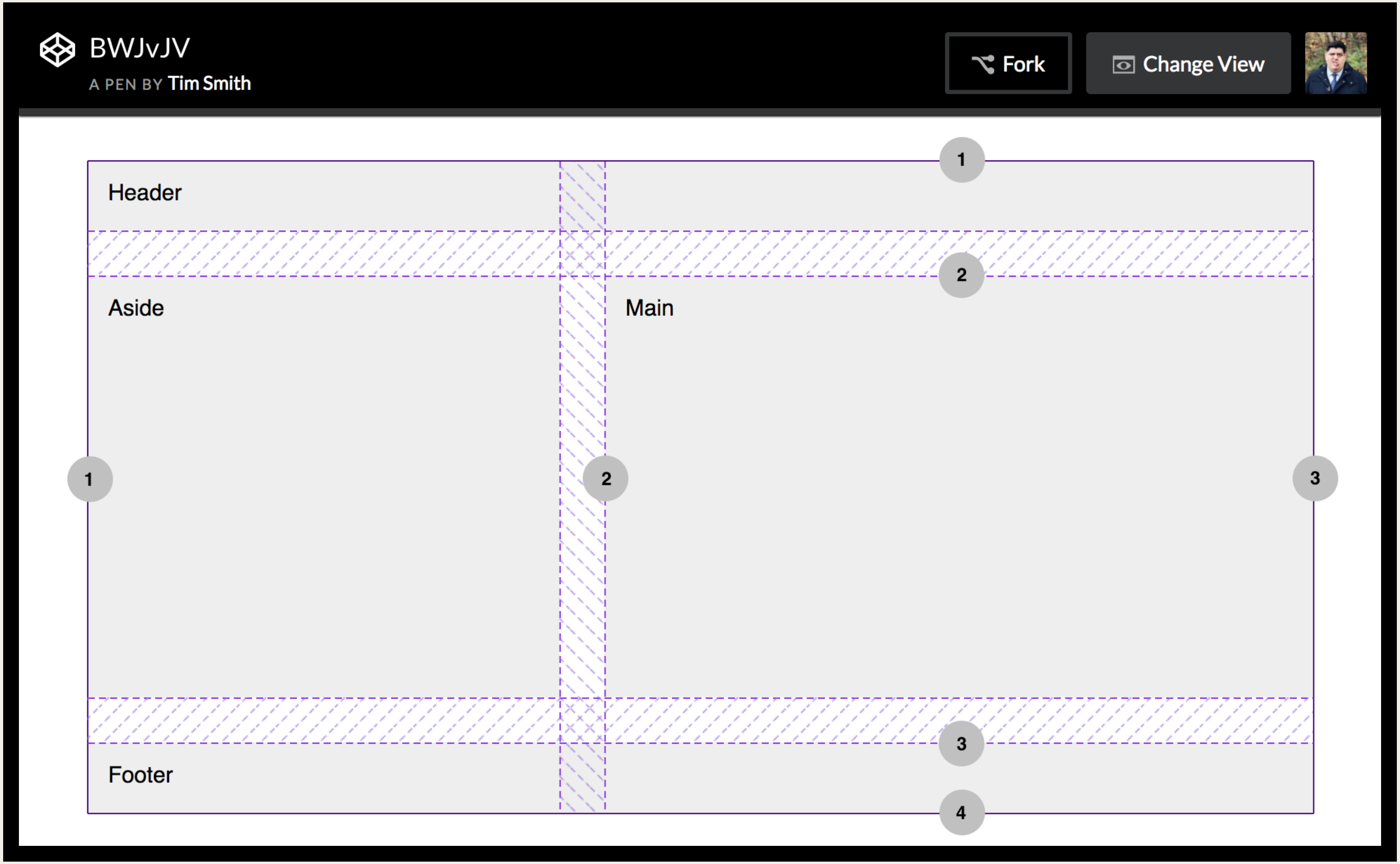
Set Up

```html
<header>Header</header>
<aside>Aside</aside>
<main>Main Content</main>
<footer>Footer</footer>
```

```css
body {
  display: grid;
  grid-template-columns: 2fr 3fr;
  grid-gap: 2rem;
}
```

Let’s Place Our Items

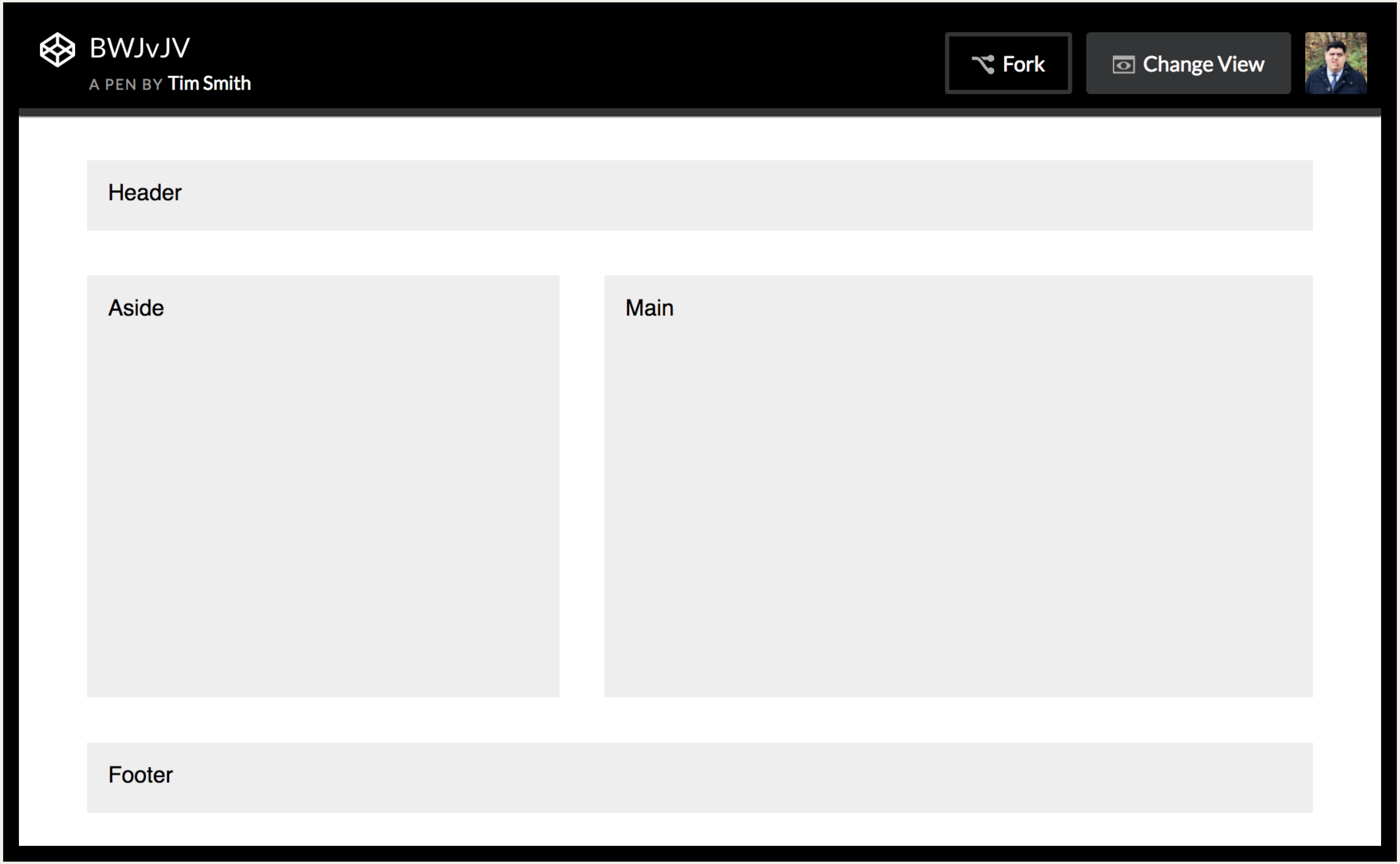
```
header {
  grid-column: 1 / span 2;
}

aside {
  grid-column-end: span 1;
  grid-row: 2;
}

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

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


BWJvJV
A PEN BY Tim Smith
Fork
Change View
Header
Aside
Main
Footer
1
2
3
4
1
2
3

Grid Lines

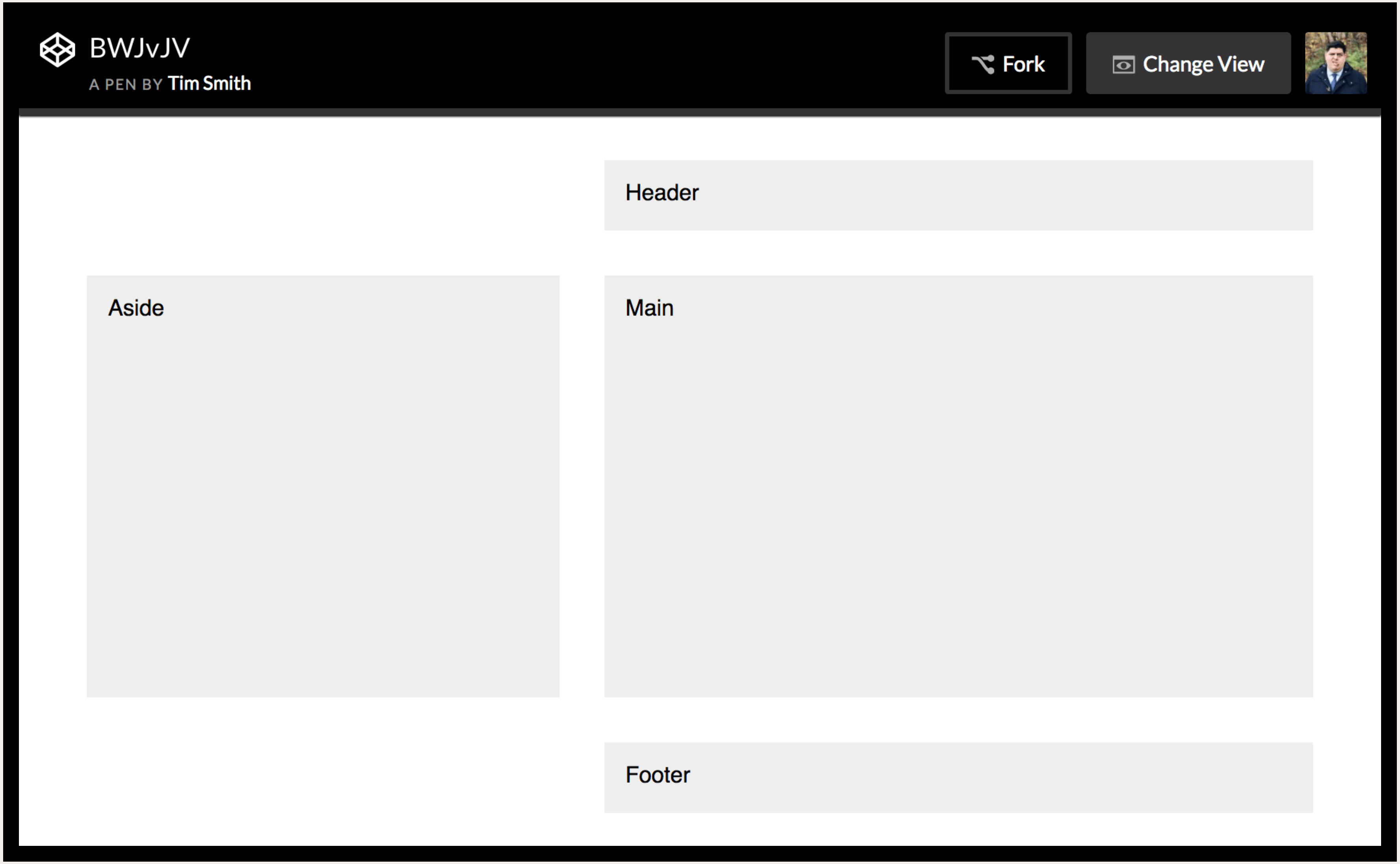
Rewrite with Grid Areas

```css
body {
  display: grid;
  grid-template-columns: 2fr 3fr;
  grid-gap: 2rem;
  grid-template-areas:
    "header header"
    "aside main"
    "footer footer";
}

header {
  grid-area: header;
}
aside {
  grid-area: aside;
}
main {
  grid-area: main;
}
footer {
  grid-area: footer;
}
```


BWJvJV
A PEN BY Tim Smith
Fork
Change View
Header
Aside
Main
Footer

What If?

```
body {
  grid-template-areas:
    ". header"
    "aside main"
    ". footer";
}
```

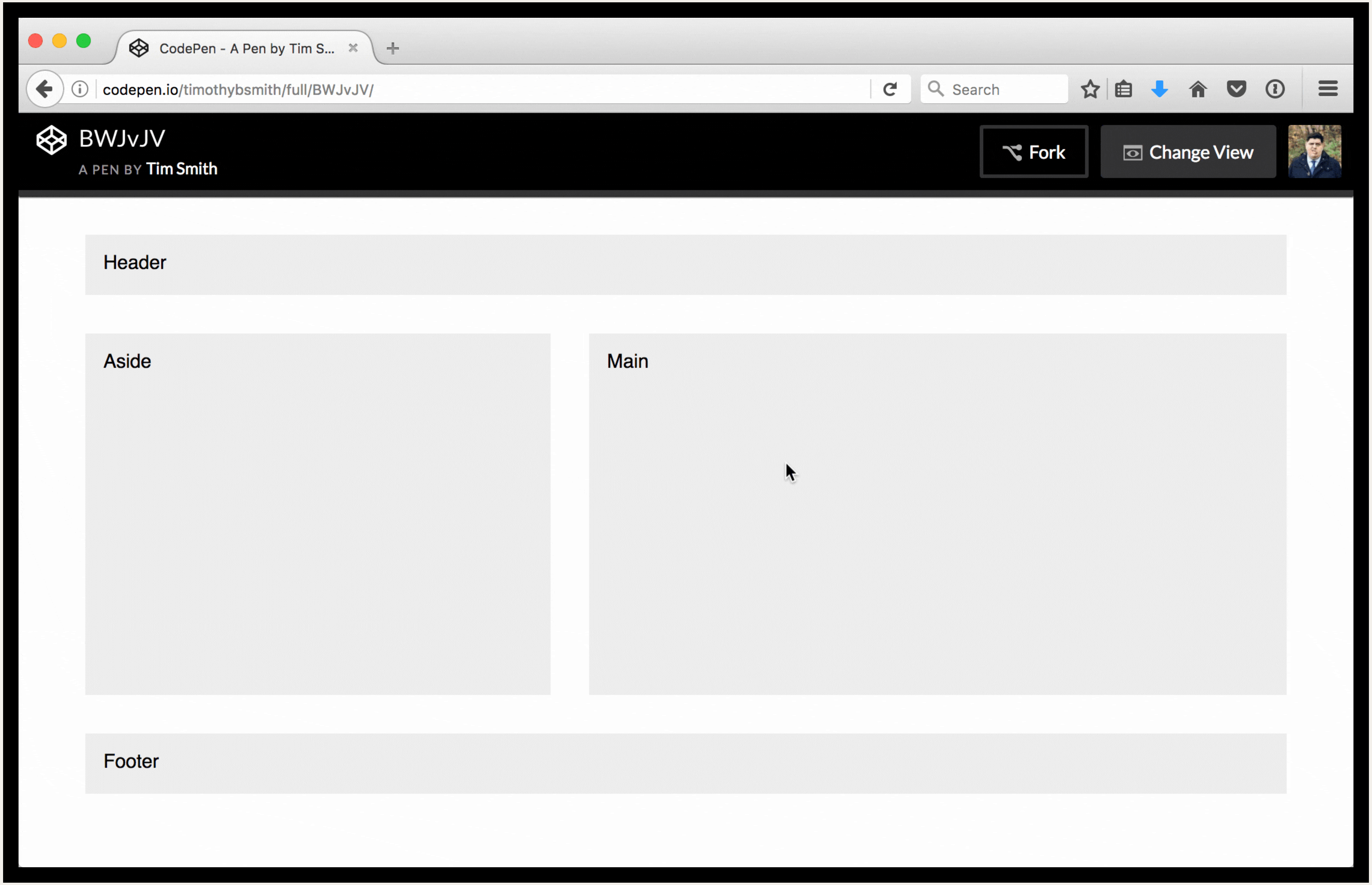


BWJvJV

A PEN BY Tim Smith

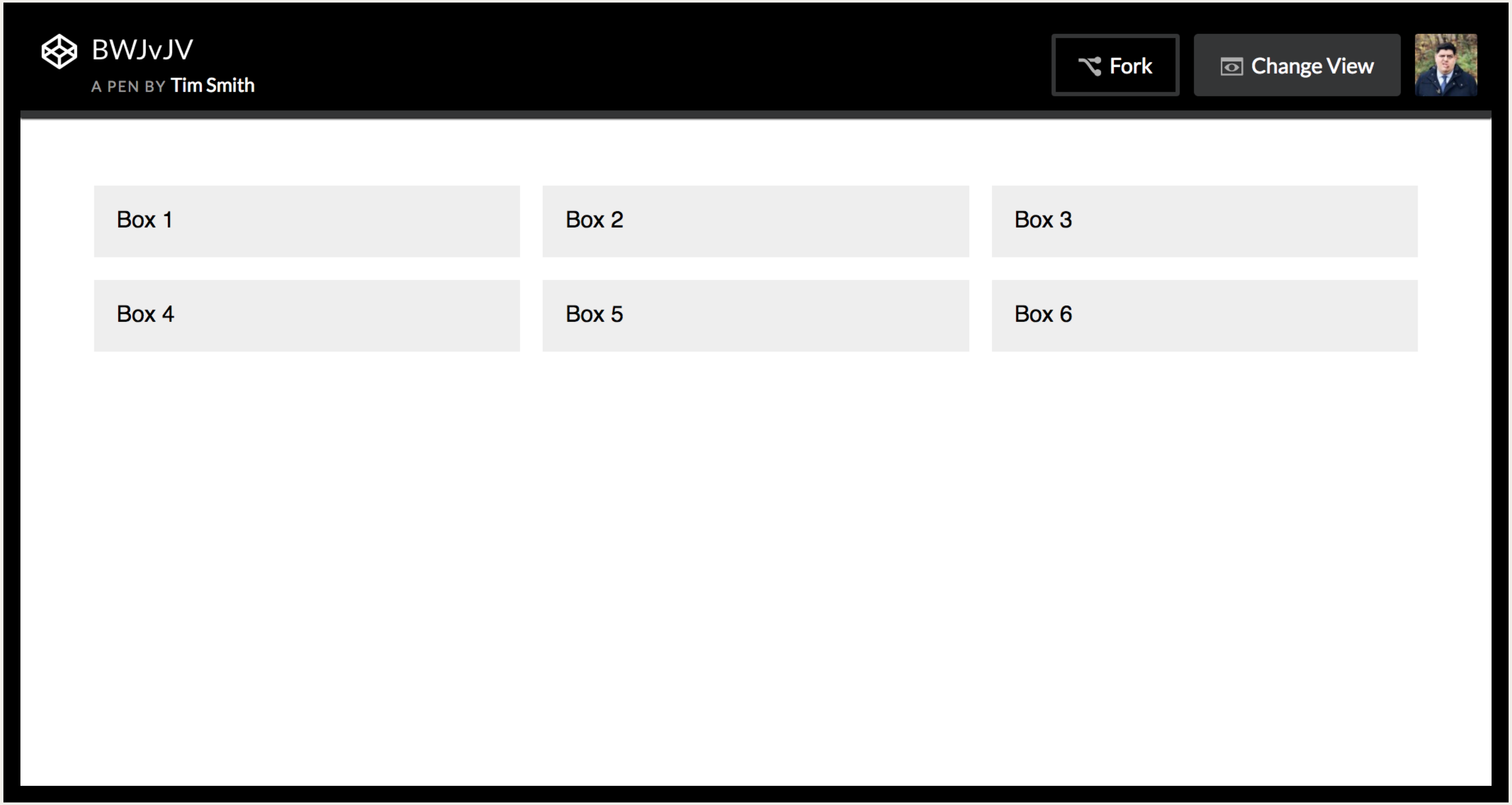
Easily Responsive

```css
body {
  display: grid;
  grid-gap: 2rem;
  grid-template-areas:
    "header"
    "main"
    "aside"
    "footer";
}

@media and screen (min-width: 40em) {
  body {
    grid-template-columns: 2fr 3fr;
    grid-template-areas:
      "header header"
      "aside main"
      "footer footer";
  }
}
```



CodePen - A Pen by Tim S...
codepen.io/timothybsmith/full/BWJvJV/
Search
BWJvJV
A PEN BY Tim Smith
Fork
Change View
Header
Aside
Main
Footer

Card Layout



BWJvJV

A PEN BY Tim Smith

Box 1	Box 2	Box 3
Box 4	Box 5	Box 6

Setup

```
<ul class="boxes">
  <li>Box 1</li>
  <li>Box 2</li>
  <li>Box 3</li>
  <li>Box 4</li>
  <li>Box 5</li>
  <li>Box 6</li>
</ul>
```

Setup

```
<ul class="boxes">
  <li>Box 1</li>
  <li>Box 2</li>
  <li>Box 3</li>
  <li>Box 4</li>
  <li>Box 5</li>
  <li>Box 6</li>
</ul>
```

Now let’s get these puppies on a Grid.

Setup

```html
<ul class="boxes">
  <li>Box 1</li>
  <li>Box 2</li>
  <li>Box 3</li>
  <li>Box 4</li>
  <li>Box 5</li>
  <li>Box 6</li>
</ul>
```

Now let’s get these puppies on a Grid.

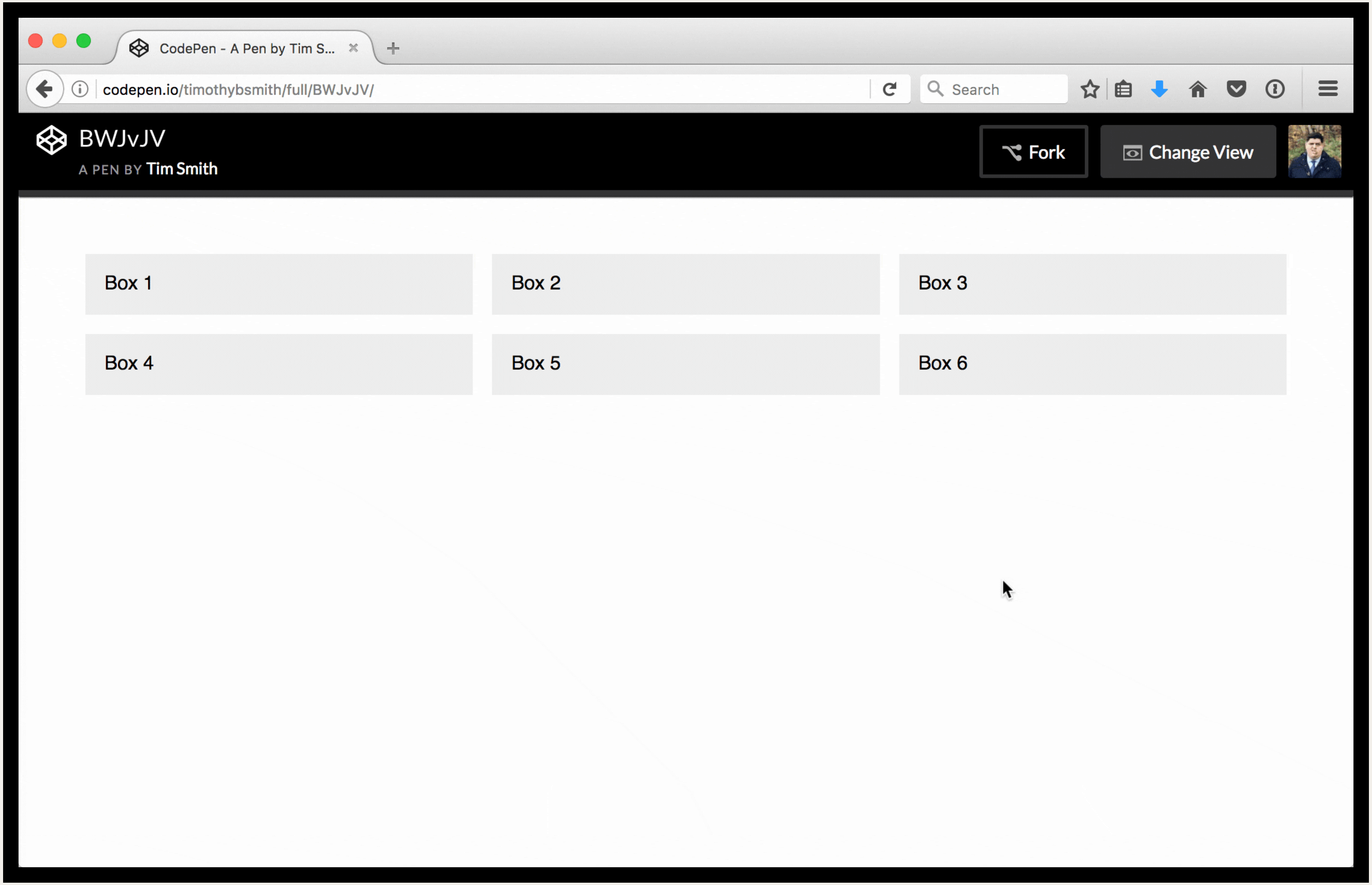
```css
.boxes {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  grid-gap: 1rem;
}
```

Whoa!

Hold Up.

Let’s Dissect This

```css
.boxes {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  grid-gap: 1rem;
}
```



CodePen - A Pen by Tim S...
codepen.io/timothybsmith/full/BWJvJV/
Search
BWJvJV
A PEN BY Tim Smith
Fork
Change View

Box 1

Box 2

Box 3

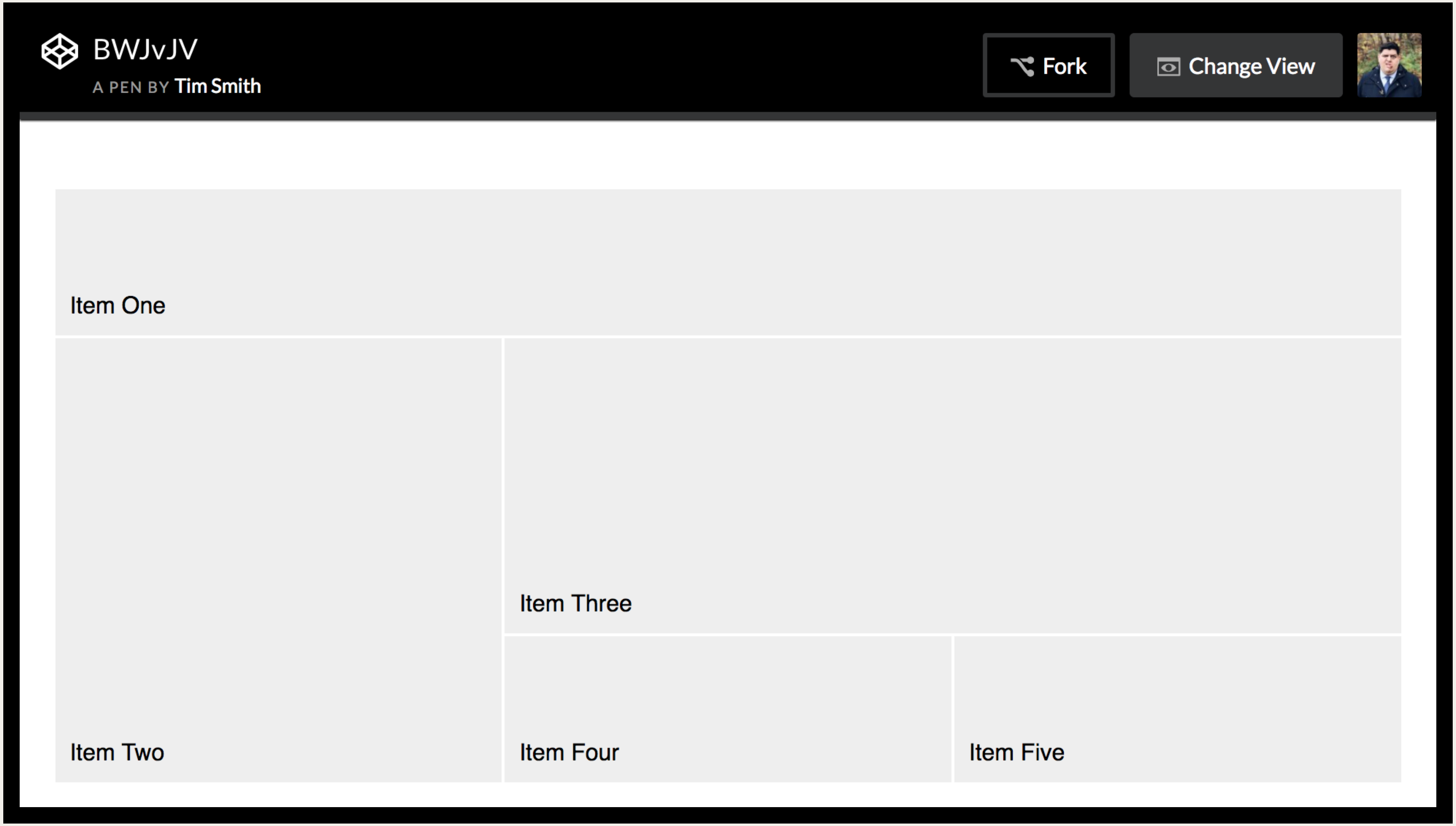
Box 4

Box 5

Box 6

Mind. Blown.

Special Portfolio-Inspired Layout



Setup

```
<ul class="portfolio">
  <li>Item One</li>
  <li>Item Two</li>
  <li>Item Three</li>
  <li>Item Four</li>
  <li>Item Five</li>
</ul>
```

Setup

```
<ul class="portfolio">
  <li>Item One</li>
  <li>Item Two</li>
  <li>Item Three</li>
  <li>Item Four</li>
  <li>Item Five</li>
</ul>
```

And setup our Grid container.

Setup

```html
<ul class="portfolio">
  <li>Item One</li>
  <li>Item Two</li>
  <li>Item Three</li>
  <li>Item Four</li>
  <li>Item Five</li>
</ul>
```

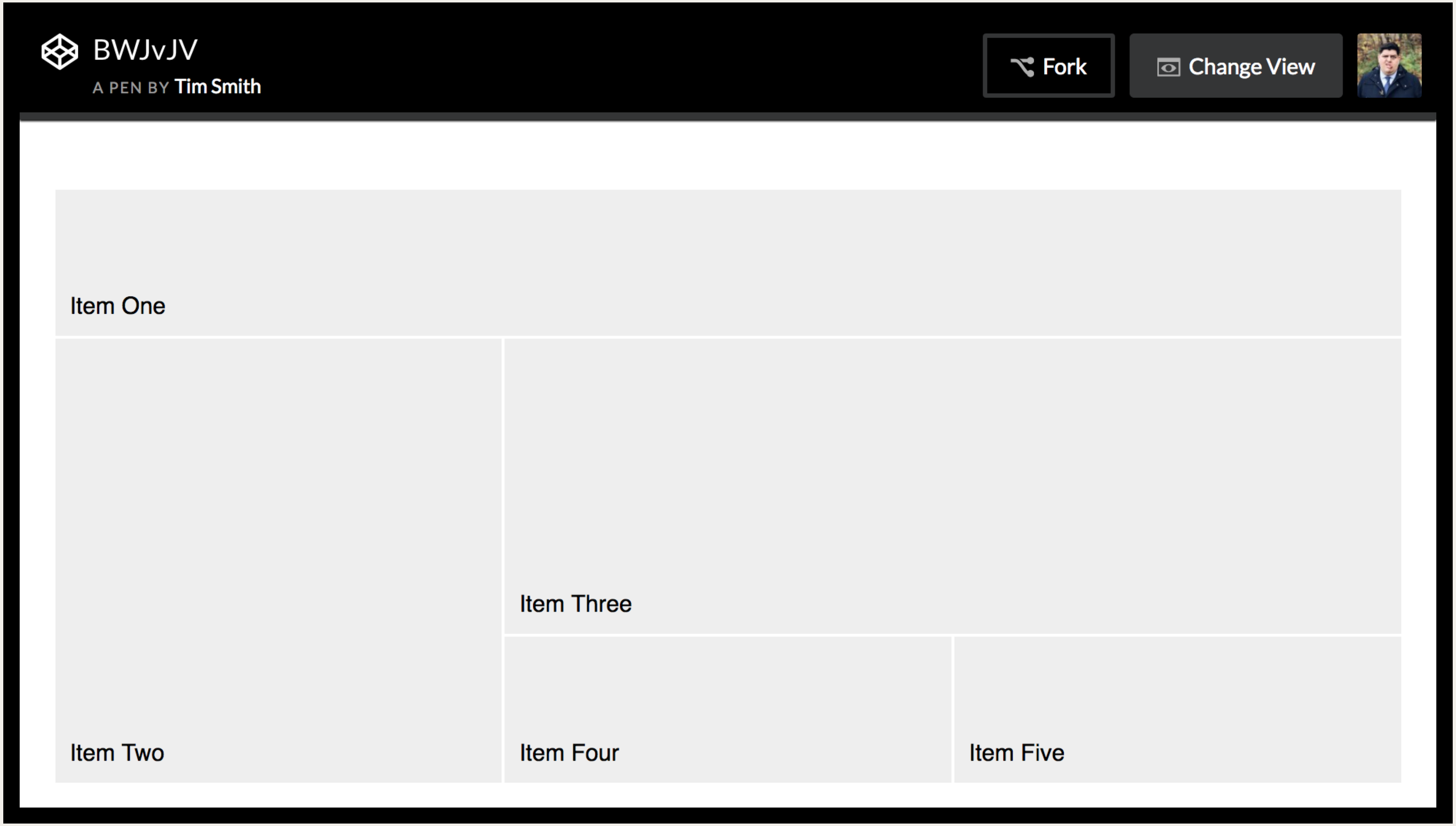
And setup our Grid container.

```css
.portfolio {
  display: grid;
  grid-auto-rows: minmax(100px, auto);
  grid-gap: 2px;
  grid-template-columns: repeat(6, 1fr);
}
```

grid-auto-rows

Let’s Place Our Items

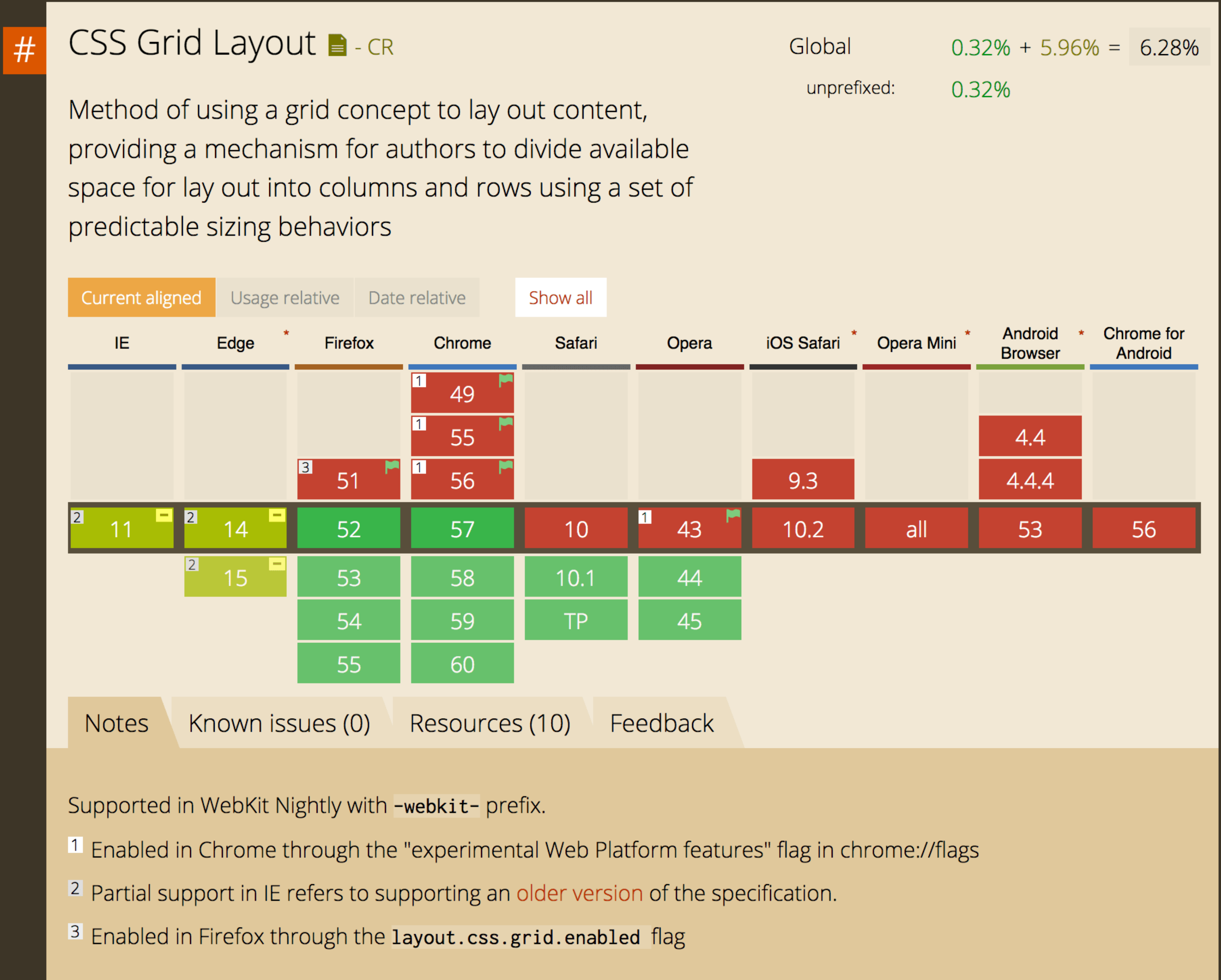
```css
.portfolio li:nth-child(1) {
  grid-column: 1 / -1;
}

.portfolio li:nth-child(2) {
  grid-column: 1 / span 2;
  grid-row-end: span 3;
}

.portfolio li:nth-child(3) {
  grid-column: 3 / span 4;
  grid-row-end: span 2;
}

.portfolio li:nth-child(4),
.portfolio li:nth-child(5) {
  grid-column: span 2;
}
```


BWJvJV
A PEN BY Tim Smith
Fork
Change View
Item One
Item Two
Item Three
Item Four
Item Five

Holy Jack City,

Batman

The Power of CSS Grid



CSS Grid Layout - CR

Global 0.32% + 5.96% = 6.28%

unprefixed: 0.32%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for lay out into columns and rows using a set of predictable sizing behaviors

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			🟥 49 [1]						
			🟥 55 [1]					🟥 4.4	
		🟥 51 [3]	🟥 56 [1]			🟥 9.3		🟥 4.4.4	
🟨 11 [2]	🟨 14 [2]	🟩 52	🟩 57	🟥 10	🟥 43 [1]	🟥 10.2	🟥 all	🟥 53	🟥 56
	🟨 15 [2]	🟩 53	🟩 58	🟩 10.1	🟩 44				
		🟩 54	🟩 59	🟩 TP	🟩 45				
		🟩 55	🟩 60						

Supported in WebKit Nightly with `-webkit-` prefix.

[1] Enabled in Chrome through the "experimental Web Platform features" flag in chrome://flags

[2] Partial support in IE refers to supporting an older version of the specification.

[3] Enabled in Firefox through the `layout.css.grid.enabled` flag

Why Now?

It’s

Our Job

Simpler

Code

New

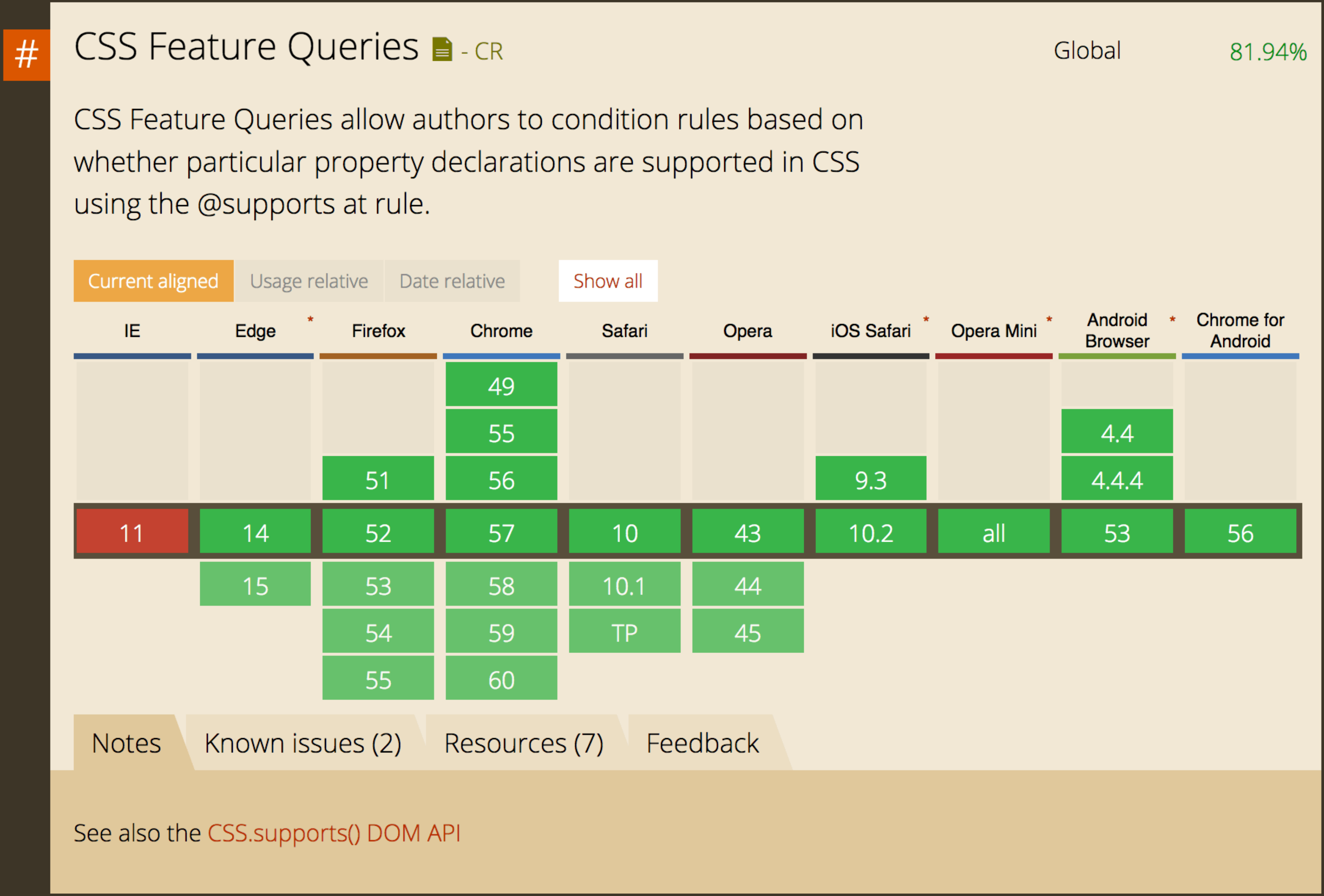
Amazing Layouts

I Delayed Learning Flexbox **and Regret It**

Feature Queries

Like This

```
@supports (display:grid) {
  /* Grid CSS in here */
}

@supports not (display:grid) {
  /* Fallback CSS here */
}
```



CSS Feature Queries - CR

CSS Feature Queries allow authors to condition rules based on whether particular property declarations are supported in CSS using the @supports at rule.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			55					4.4	
		51	56			9.3		4.4.4	
11	14	52	57	10	43	10.2	all	53	56
	15	53	58	10.1	44				
		54	59	TP	45				
		55	60						

See also the CSS.supports() DOM API

Browsers

Ignore CSS

They Don’t Understand

What Did We Learn?

1 CSS Grid Is Awesome

2 CSS Grid Is Friendly

3

Use CSS Grid Today

Only the Beginning

Rachel Andrew

gridbyexample.com

Jen Simmons

labs.jensimmons.com

Mozilla Developer Network

developer.mozilla.org

Questions?

Thanks!

ttimsmith.com · @smithtimmytim