Your 2019 CSS Layout Toolkit

@rachelandrew @oredev

Rachel Andrew

Co-founder Perch CMS and Notist

Editor in Chief at Smashing Magazine

W3C Invited Expert to the CSS Working Group, co-editor Multicol and Page Floats

@rachelandrew | https://rachelandrew.co.uk

Slides & Code

https://noti.st/rachelandrew/es94DQ

A CSS Layout System

Real layout for the first time!

Components of CSS Layout

Flow Layout, Grid, Flexbox, Multiple-column Layout

Components of CSS Layout

CSS Shapes, Transforms, Scroll Snapping, Variable Fonts

Components of CSS Layout

Writing Modes, Logical Properties & Values, Alignment, Sizing

Components of CSS Layout

Media Queries, Feature Queries

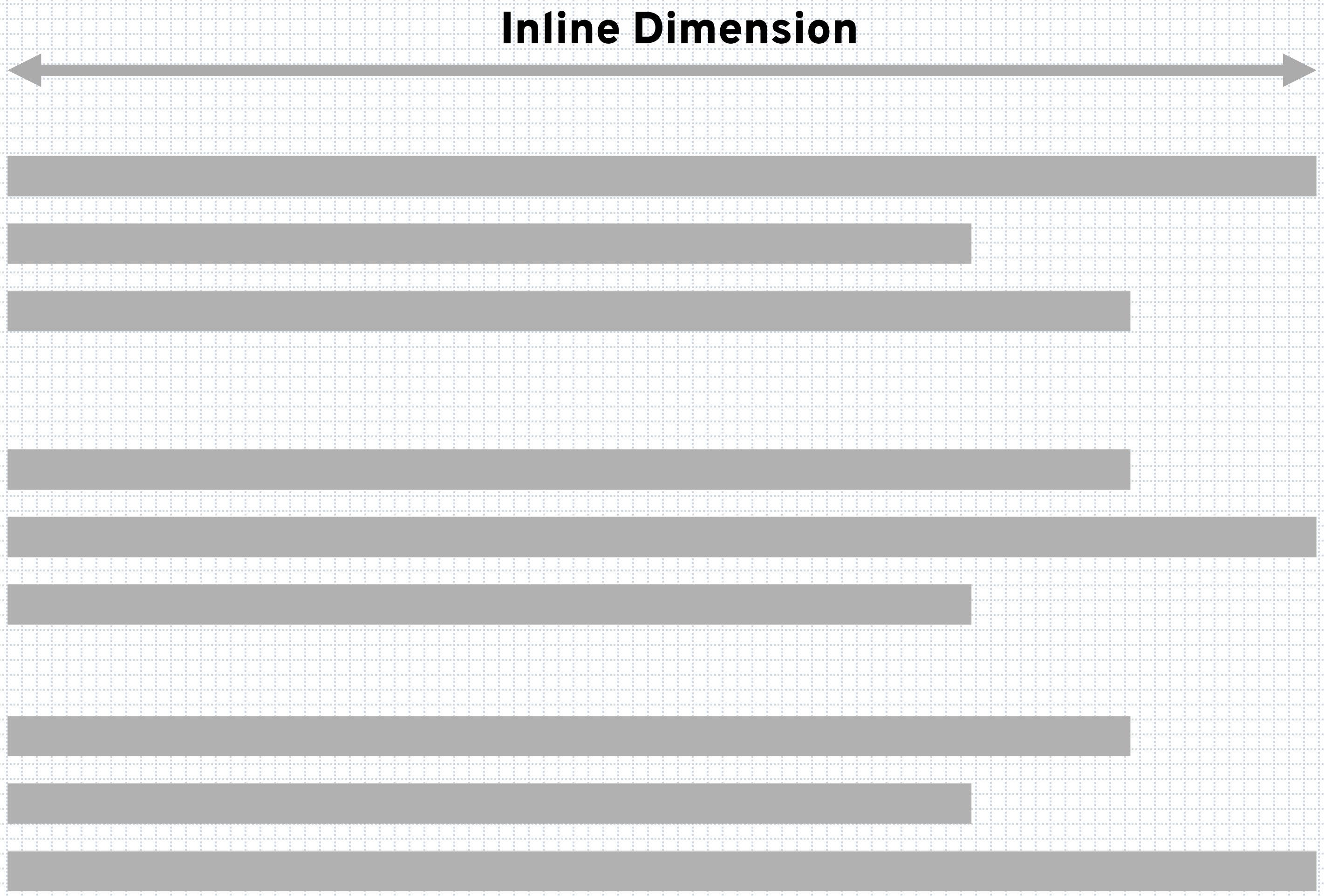
Look past the headline specs

Understanding the things that tie together our new layout

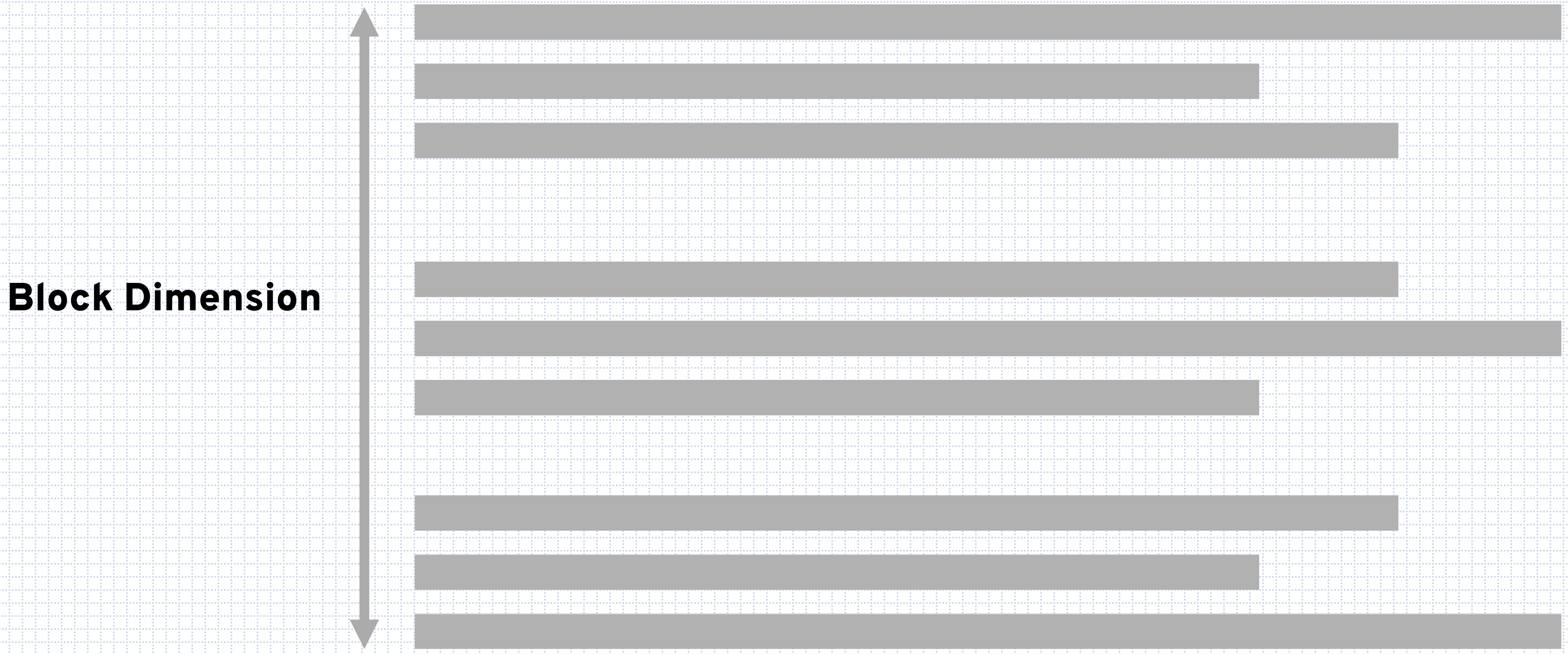
Writing Modes

Horizontal or Vertical

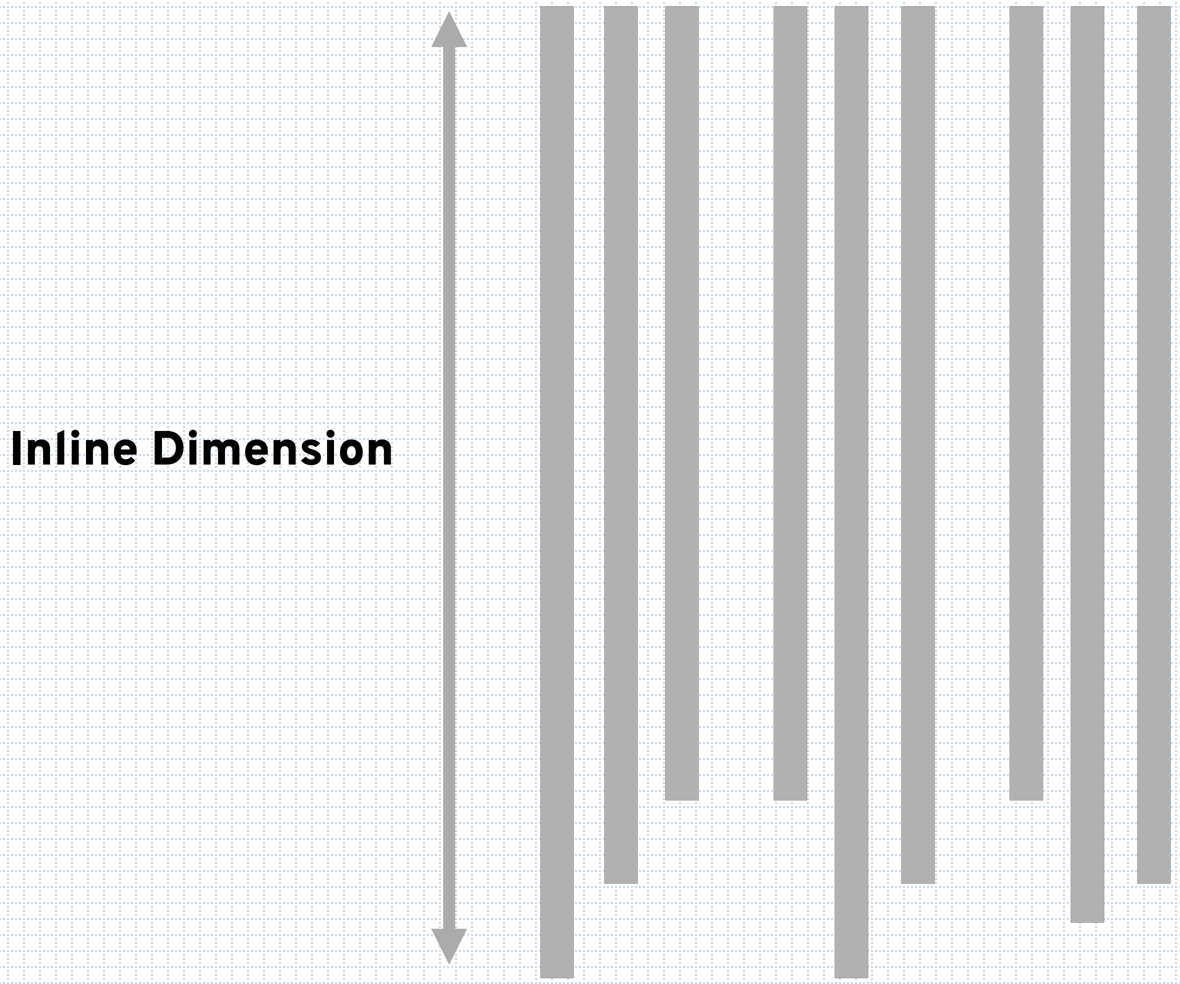
Horizontal Writing Mode



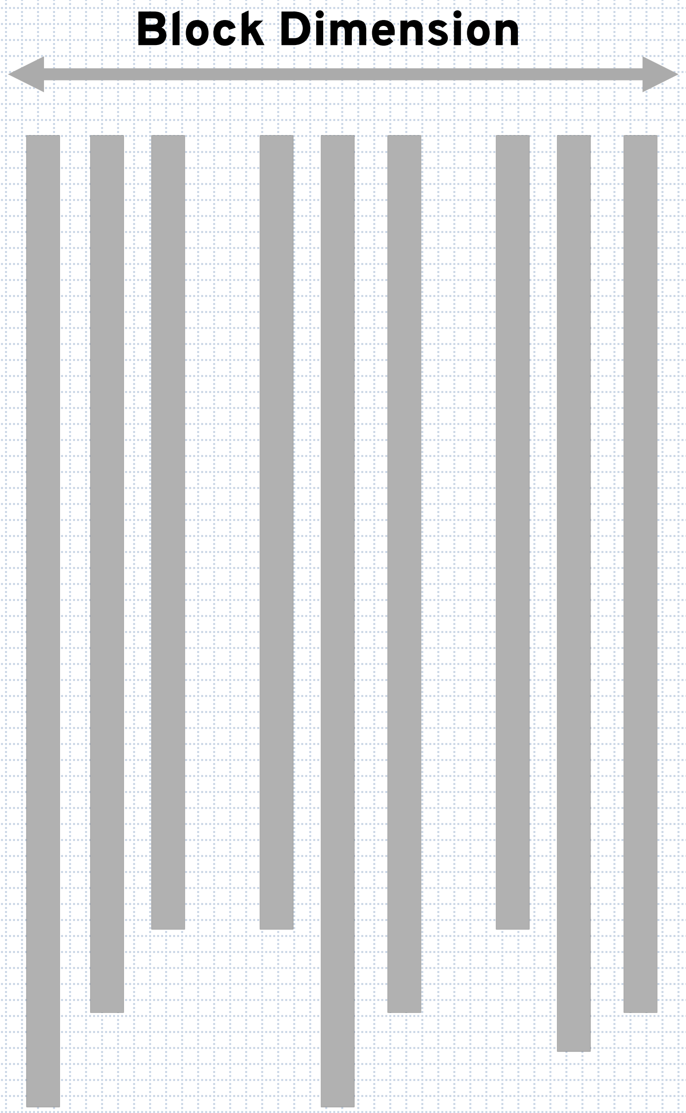
Horizontal Writing Mode



Vertical Writing Mode

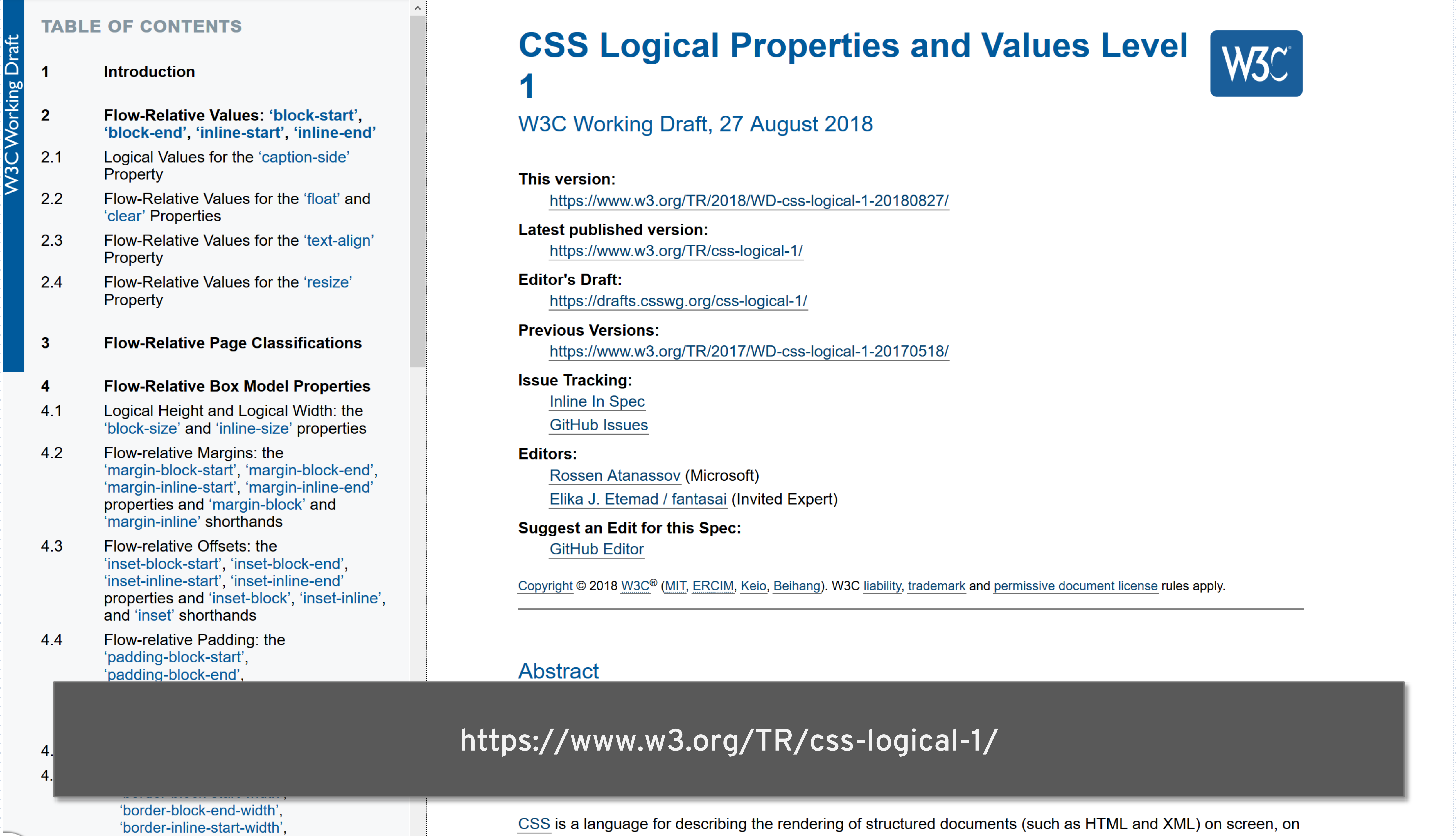


Vertical Writing Mode



Logical Properties & Values

Writing-mode relative equivalents of physical properties.



W3C Working Draft

TABLE OF CONTENTS

CSS Logical Properties and Values Level 1

W3C Working Draft, 27 August 2018

This version:
https://www.w3.org/TR/2018/WD-css-logical-1-20180827/

Latest published version:
https://www.w3.org/TR/css-logical-1/

Editor's Draft:
https://drafts.csswg.org/css-logical-1/

Previous Versions:
https://www.w3.org/TR/2017/WD-css-logical-1-20170518/

Issue Tracking:
Inline In Spec
GitHub Issues

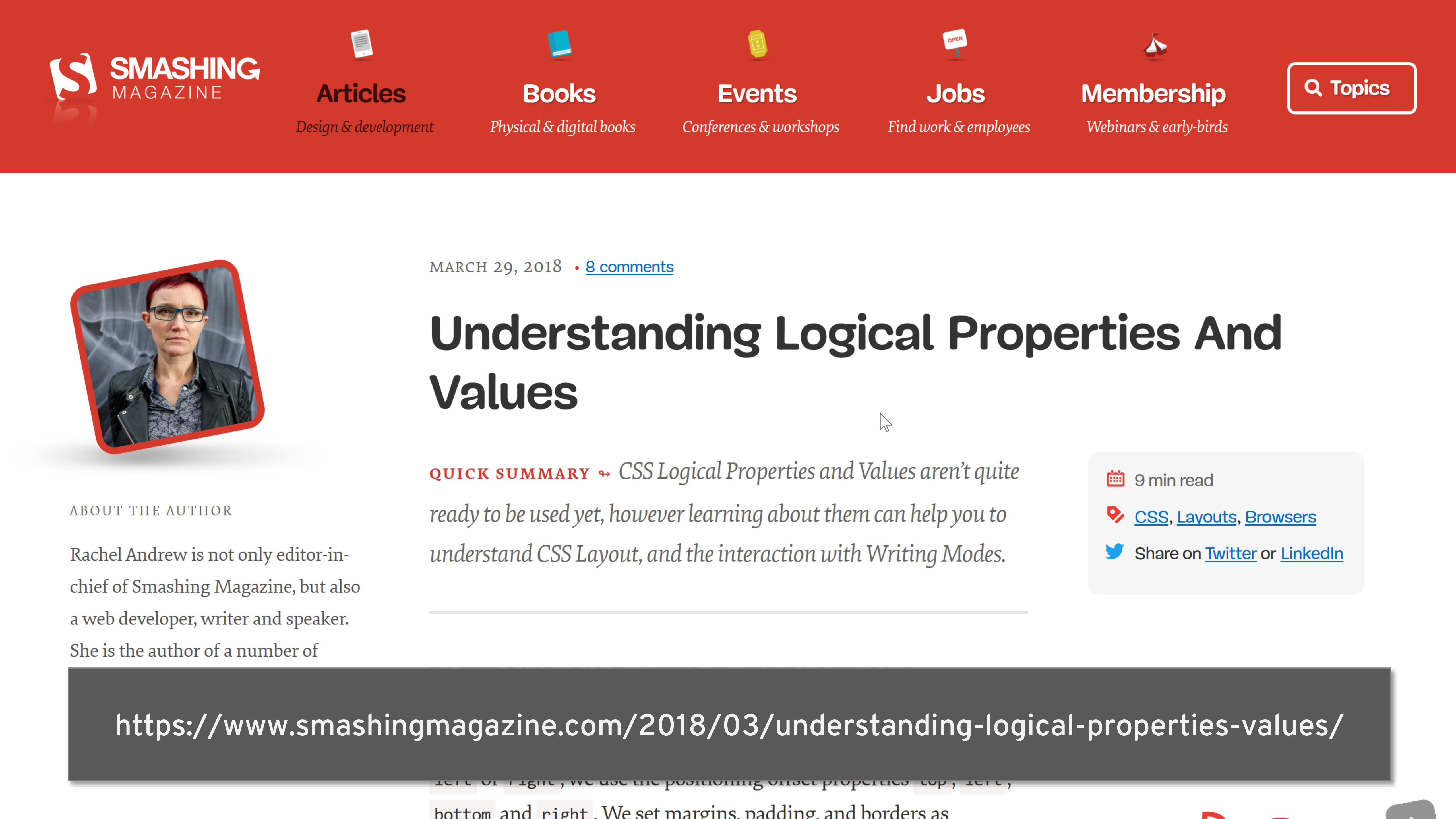
Editors:
Rossen Atanassov (Microsoft)
Elika J. Etemad / fantasai (Invited Expert)

Suggest an Edit for this Spec:
GitHub Editor

Abstract

https://www.w3.org/TR/css-logical-1/

CSS is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on



MARCH 29, 2018 • 8 comments

Understanding Logical Properties And Values

ABOUT THE AUTHOR

Rachel Andrew is not only editor-in-chief of Smashing Magazine, but also a web developer, writer and speaker. She is the author of a number of

QUICK SUMMARY ↬ *CSS Logical Properties and Values aren't quite ready to be used yet, however learning about them can help you to understand CSS Layout, and the interaction with Writing Modes.*

9 min read

CSS, Layouts, Browsers

Share on Twitter or LinkedIn

`left` or `right`, we use the positioning offset properties `top`, `left`,

`bottom` and `right`. We set margins, padding, and borders as

https://www.smashingmagazine.com/2018/03/understanding-logical-properties-values/

Box Alignment

Consistent alignment across layout methods.

Align or Justify?

align-content
align-self
align-items

justify-content
justify-self
justify-items

In Grid Layout

align-content
align-self
align-items

Block Axis

justify-content
justify-self
justify-items

Inline Axis

Distributing space

align-content and justify-content

In Flex Layout

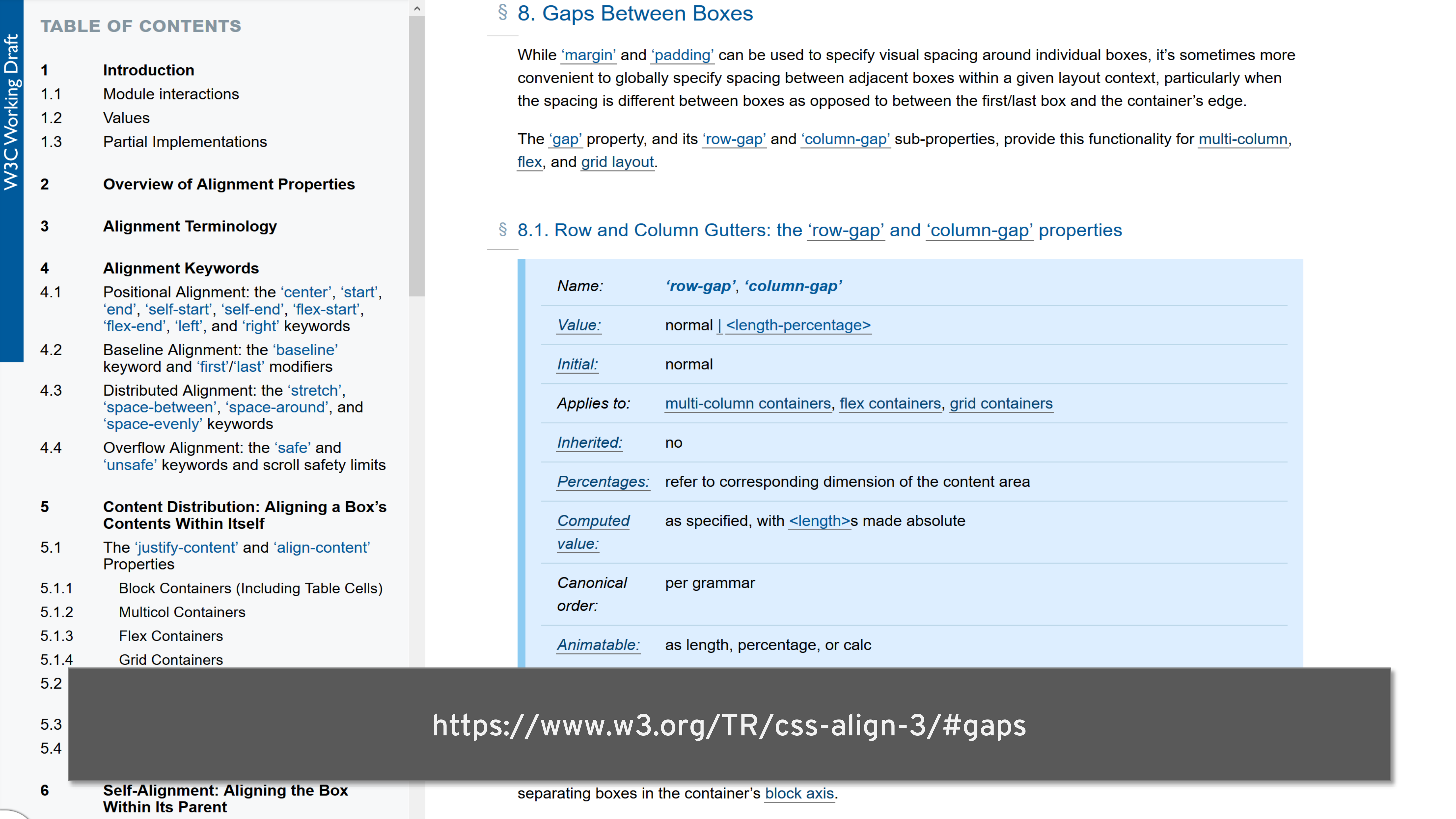
align-content **align-self** **align-items**	$\}$ ***Cross*** **Axis**
justify-content	***Main*** **Axis**

Alignment in Block Layout

No browser implementation as yet …

Gaps

row-gap, column-gap, gap



W3C Working Draft

TABLE OF CONTENTS

§ 8. Gaps Between Boxes

While ‘margin’ and ‘padding’ can be used to specify visual spacing around individual boxes, it’s sometimes more convenient to globally specify spacing between adjacent boxes within a given layout context, particularly when the spacing is different between boxes as opposed to between the first/last box and the container’s edge.

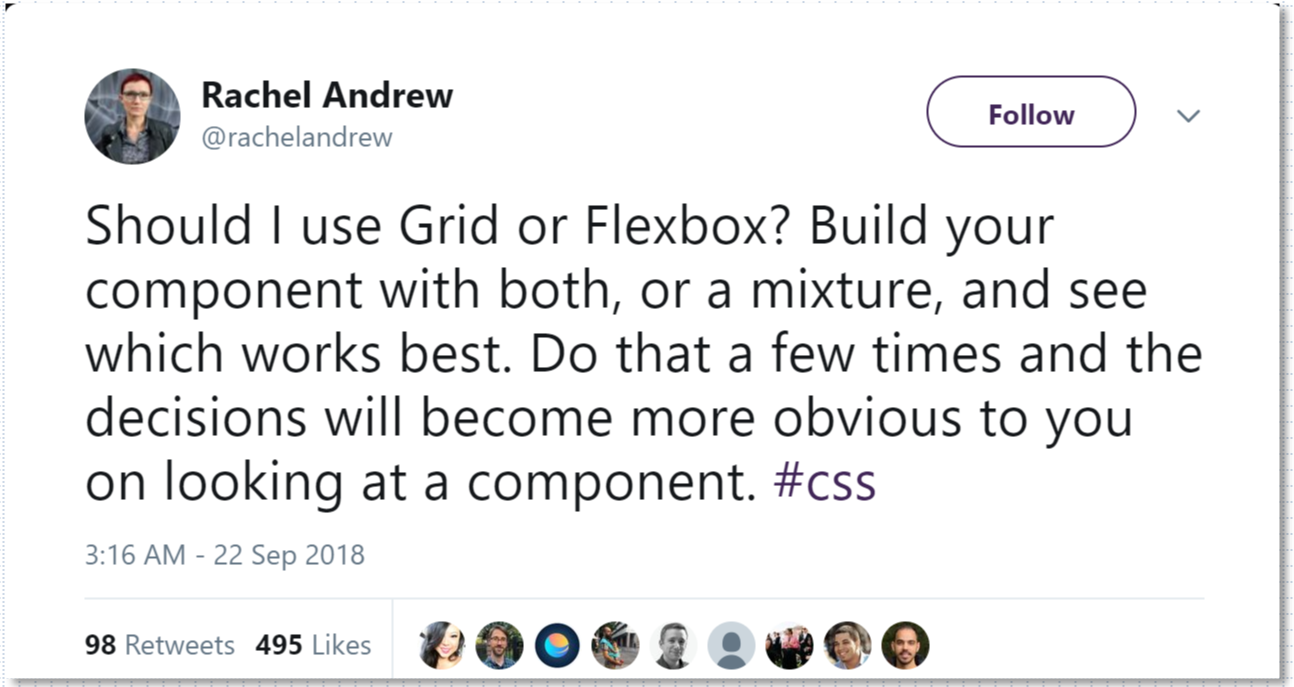
The ‘gap’ property, and its ‘row-gap’ and ‘column-gap’ sub-properties, provide this functionality for multi-column, flex, and grid layout.

§ 8.1. Row and Column Gutters: the ‘row-gap’ and ‘column-gap’ properties

Name:	***‘row-gap’, ‘column-gap’***
Value:	normal | <length-percentage>
Initial:	normal
Applies to:	multi-column containers, flex containers, grid containers
Inherited:	no
Percentages:	refer to corresponding dimension of the content area
Computed value:	as specified, with <length>s made absolute
Canonical order:	per grammar
Animatable:	as length, percentage, or calc

https://www.w3.org/TR/css-align-3/#gaps

separating boxes in the container’s block axis.


Rachel Andrew
@rachelandrew

Should I use Grid or Flexbox? Build your component with both, or a mixture, and see which works best. Do that a few times and the decisions will become more obvious to you on looking at a component. #css

3:16 AM - 22 Sep 2018

98 Retweets 495 Likes

Sizing

How big should this item be?

JANUARY 16, 2018 • 9 comments

How Big Is That Box? Understanding Sizing In CSS Layout

QUICK SUMMARY ↬ *When starting to use Flexbox and Grid, it can be frustrating to find that we sometimes don't get the layout we expect. Often this is due to the way sizing is calculated in these new layout methods. In this article, I try to explain exactly how big that box is, and how it got to be that size!*

22 min read

CSS, Layouts, Browsers

Share on Twitter or LinkedIn

ABOUT THE AUTHOR

Rachel Andrew is not only editor-in-chief of Smashing Magazine, but also a web developer, writer and speaker. She is the author of a number of books, including … More about

https://www.smashingmagazine.com/2018/01/understanding-sizing-css-layout/

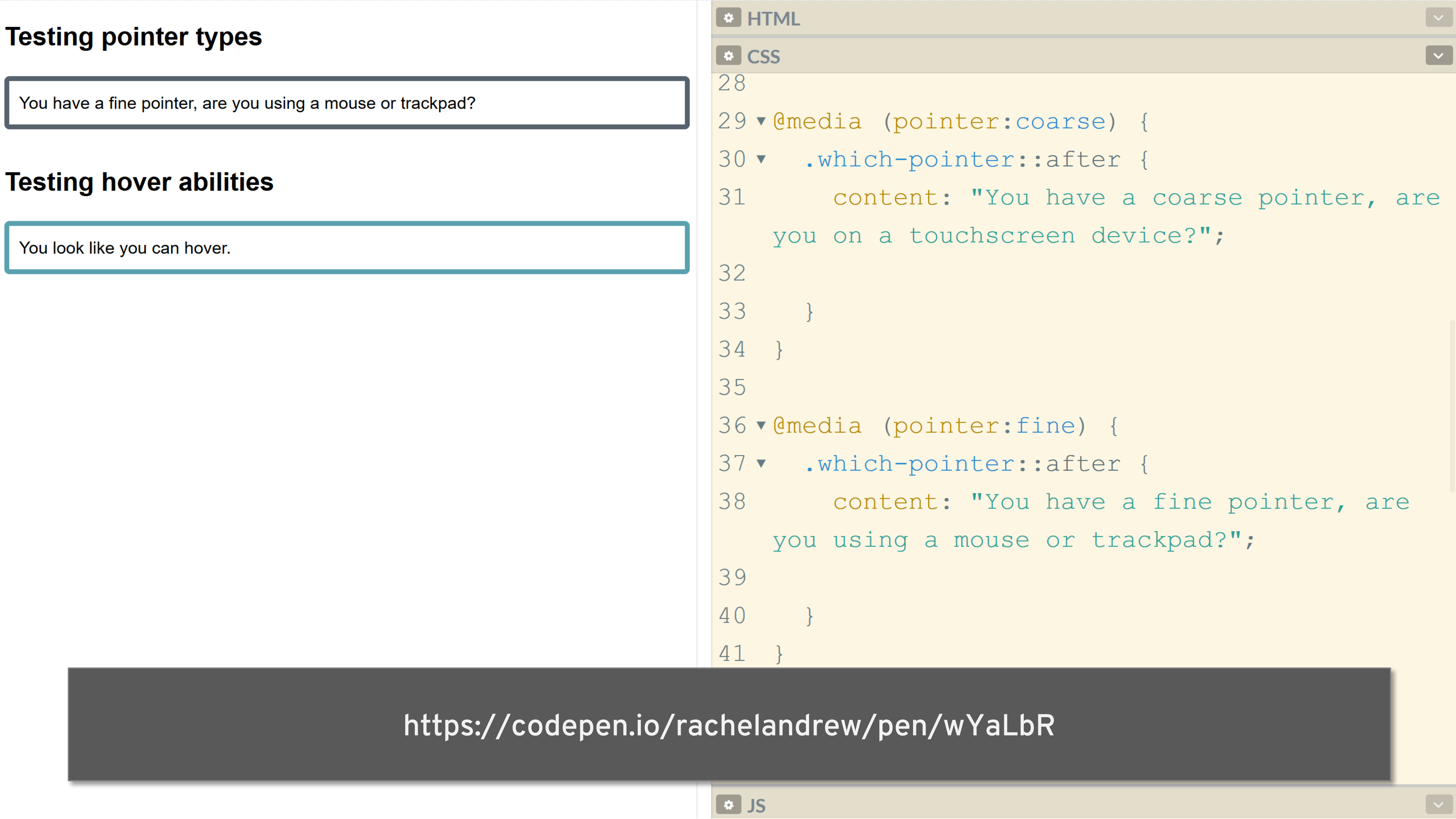
inside grid and flex items. Quite often this *just works*, and we get the

Media Queries

Level 4 Media Queries

Media Features

What features does this environment have?



Testing pointer types

You have a fine pointer, are you using a mouse or trackpad?

Testing hover abilities

You look like you can hover.

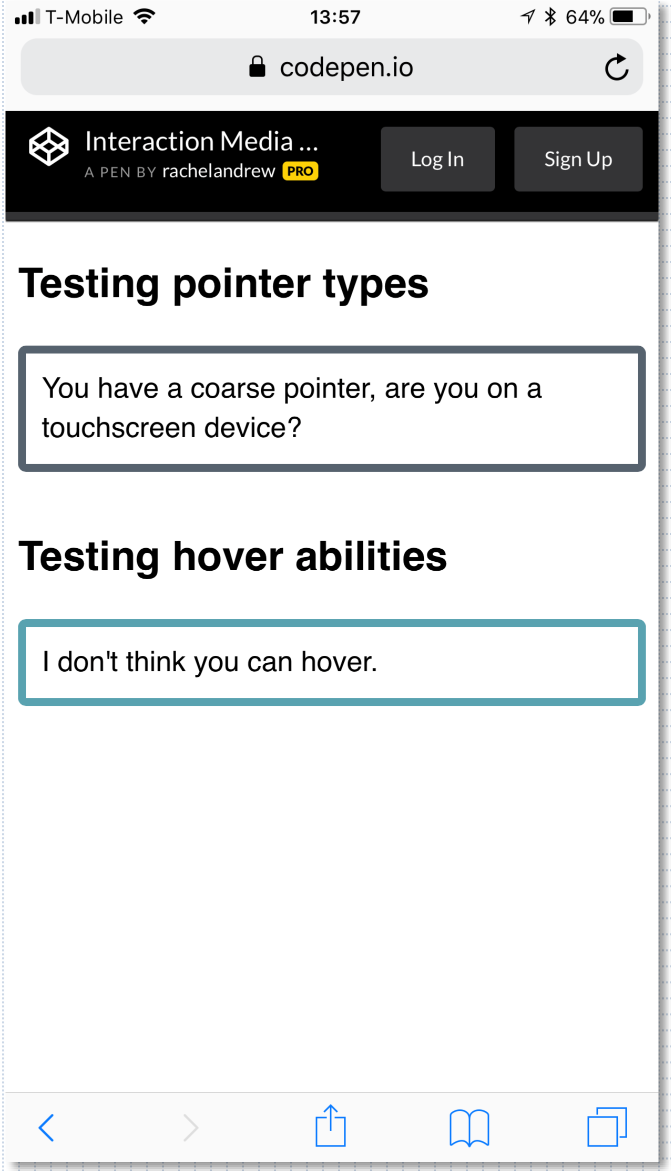
HTML

CSS

```css

@media (pointer:coarse) {
  .which-pointer::after {
    content: "You have a coarse pointer, are you on a touchscreen device?";

  }
}

@media (pointer:fine) {
  .which-pointer::after {
    content: "You have a fine pointer, are you using a mouse or trackpad?";

  }
}
```

JS

https://codepen.io/rachelandrew/pen/wYaLbR

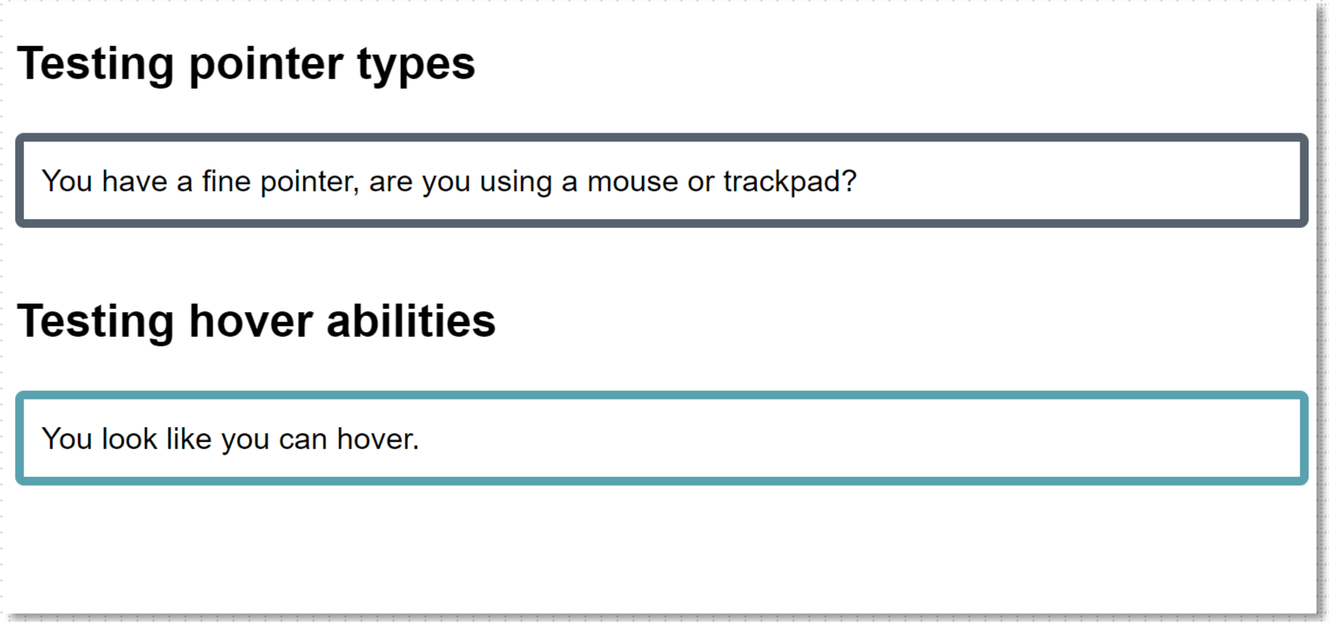


Testing pointer types

You have a coarse pointer, are you on a touchscreen device?

Testing hover abilities

I don't think you can hover.



Testing pointer types

You have a fine pointer, are you using a mouse or trackpad?

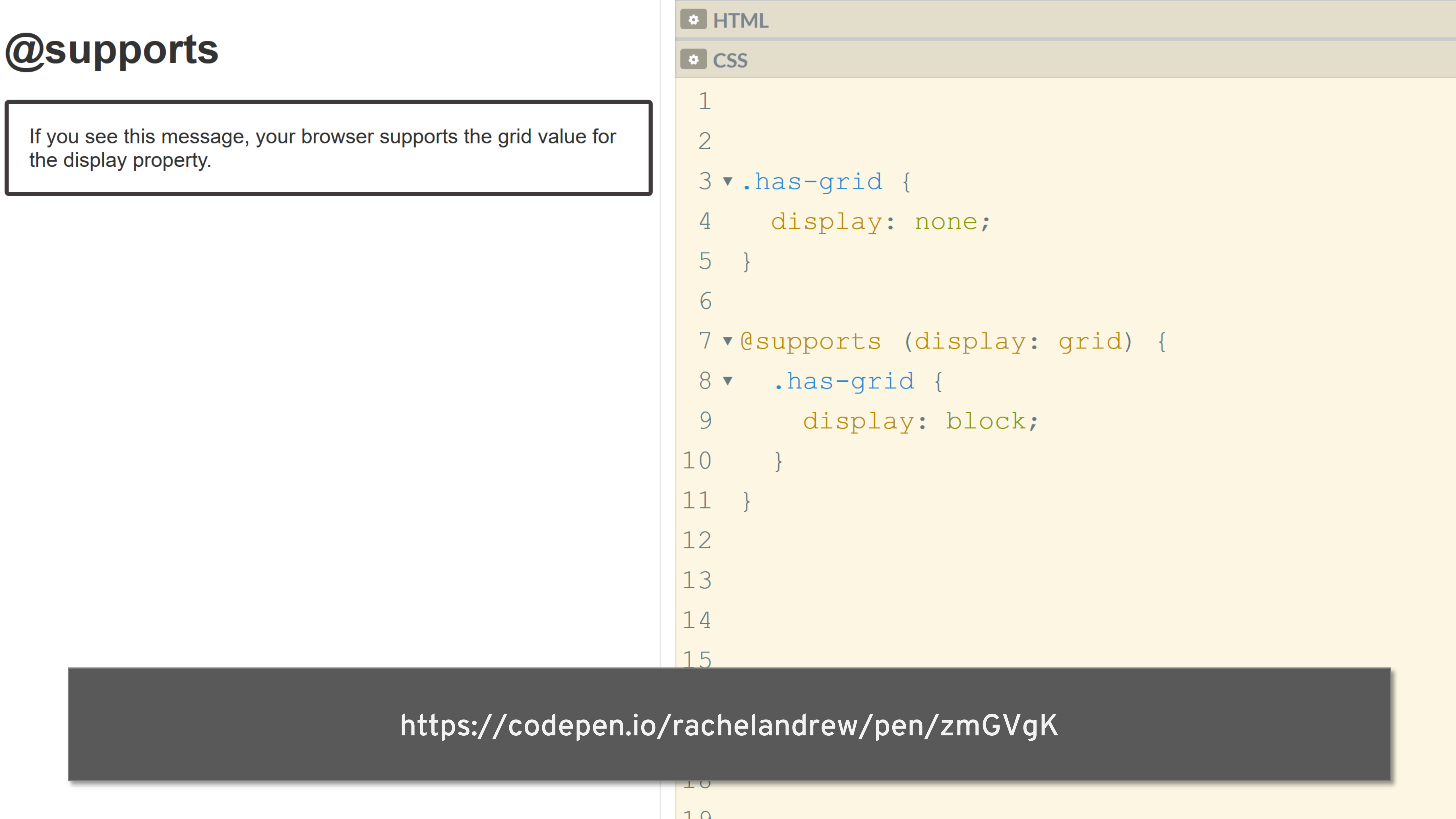
Testing hover abilities

You look like you can hover.

Feature Queries

Hey browser! Do you speak grid?

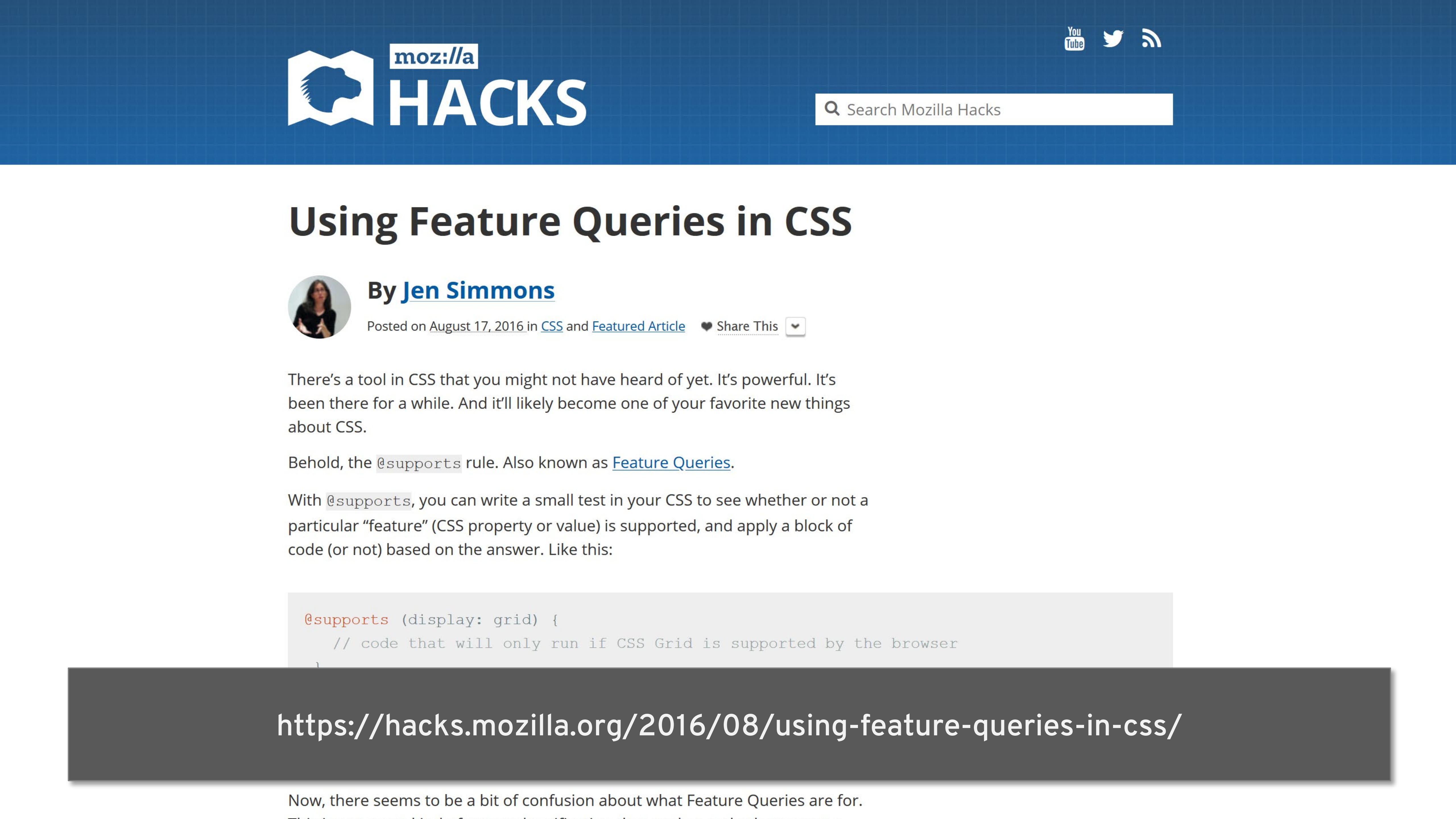
@supports



If you see this message, your browser supports the grid value for the display property.

```
.has-grid {
  display: none;
}

@supports (display: grid) {
  .has-grid {
    display: block;
  }
}
```

https://codepen.io/rachelandrew/pen/zmGVgK



Using Feature Queries in CSS

By Jen Simmons

Posted on August 17, 2016 in CSS and Featured Article ❤ Share This

There's a tool in CSS that you might not have heard of yet. It's powerful. It's been there for a while. And it'll likely become one of your favorite new things about CSS.

Behold, the `@supports` rule. Also known as Feature Queries.

With `@supports`, you can write a small test in your CSS to see whether or not a particular "feature" (CSS property or value) is supported, and apply a block of code (or not) based on the answer. Like this:

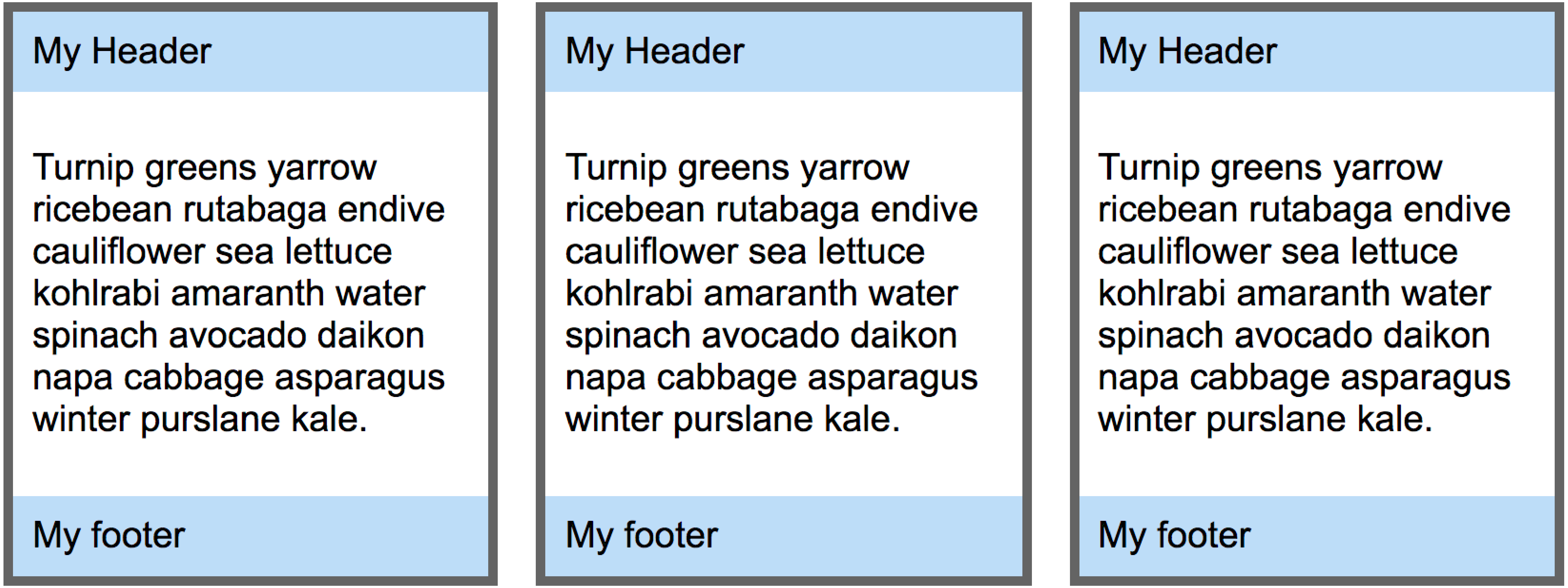
```
@supports (display: grid) {
   // code that will only run if CSS Grid is supported by the browser
```

https://hacks.mozilla.org/2016/08/using-feature-queries-in-css/

Now, there seems to be a bit of confusion about what Feature Queries are for.

Subgrid

What’s coming in CSS Grid Level 2?



My Header

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

My footer

My Header

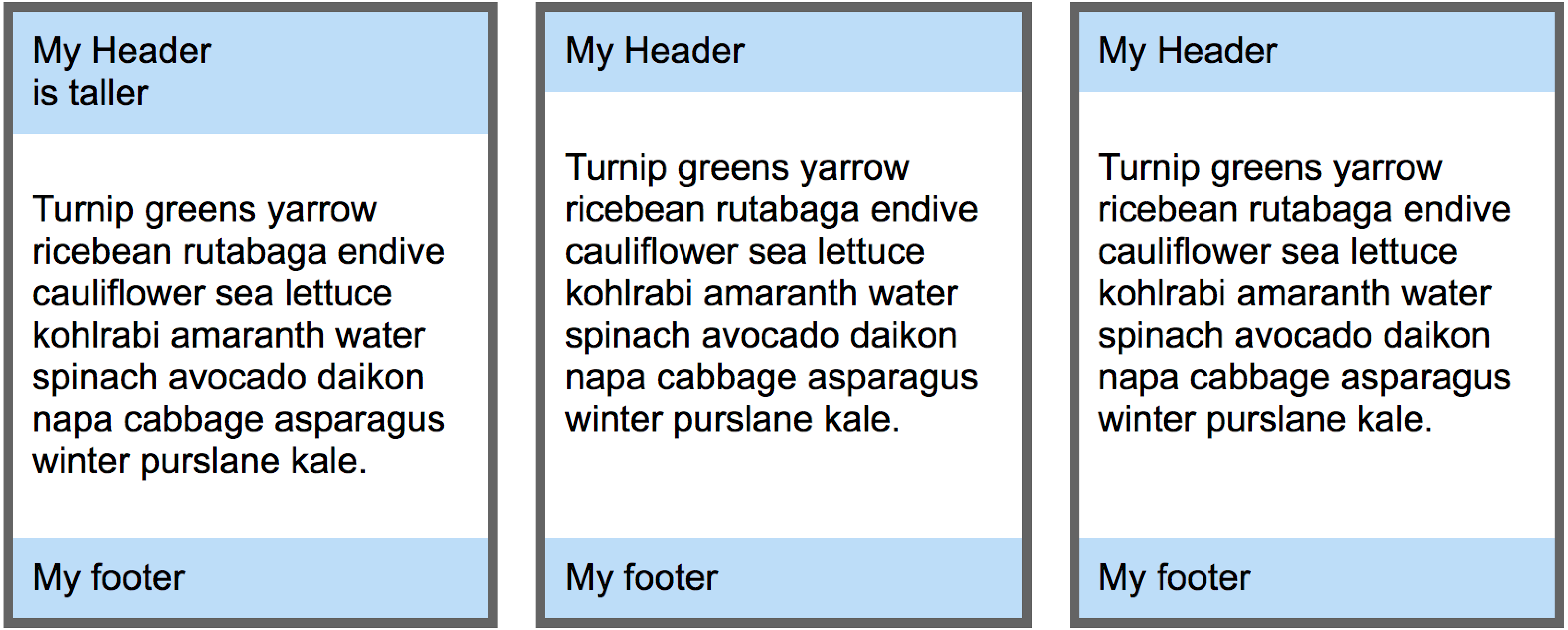
Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

My footer

My Header

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

My footer



My Header
is taller

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

My footer

My Header

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

My footer

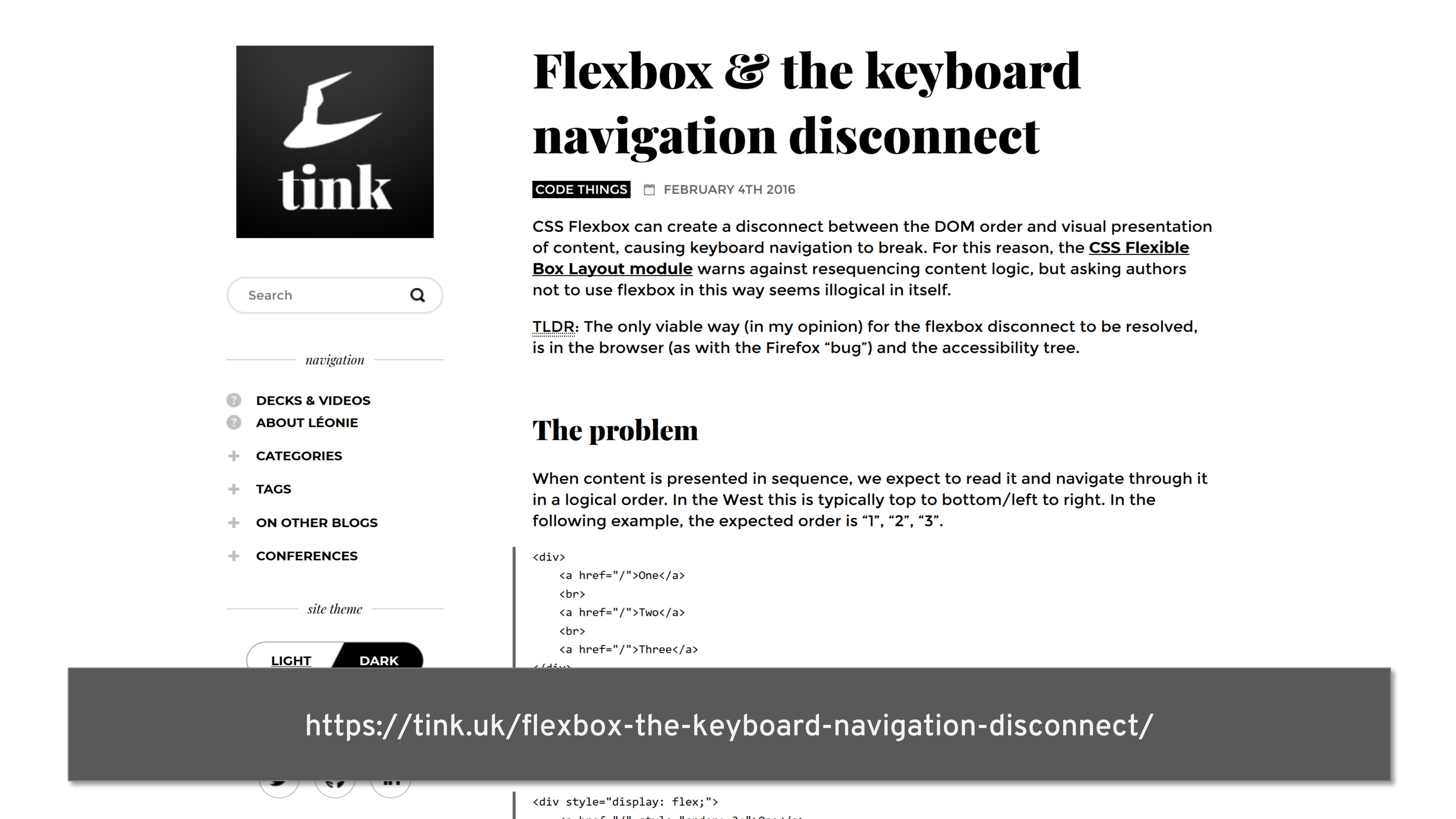
My Header

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

My footer

New Challenges

The potential for new accessibility issues.



Flexbox & the keyboard navigation disconnect

CODE THINGS FEBRUARY 4TH 2016

CSS Flexbox can create a disconnect between the DOM order and visual presentation of content, causing keyboard navigation to break. For this reason, the **CSS Flexible Box Layout module** warns against resequencing content logic, but asking authors not to use flexbox in this way seems illogical in itself.

TLDR: The only viable way (in my opinion) for the flexbox disconnect to be resolved, is in the browser (as with the Firefox "bug") and the accessibility tree.

The problem

When content is presented in sequence, we expect to read it and navigate through it in a logical order. In the West this is typically top to bottom/left to right. In the following example, the expected order is "1", "2", "3".

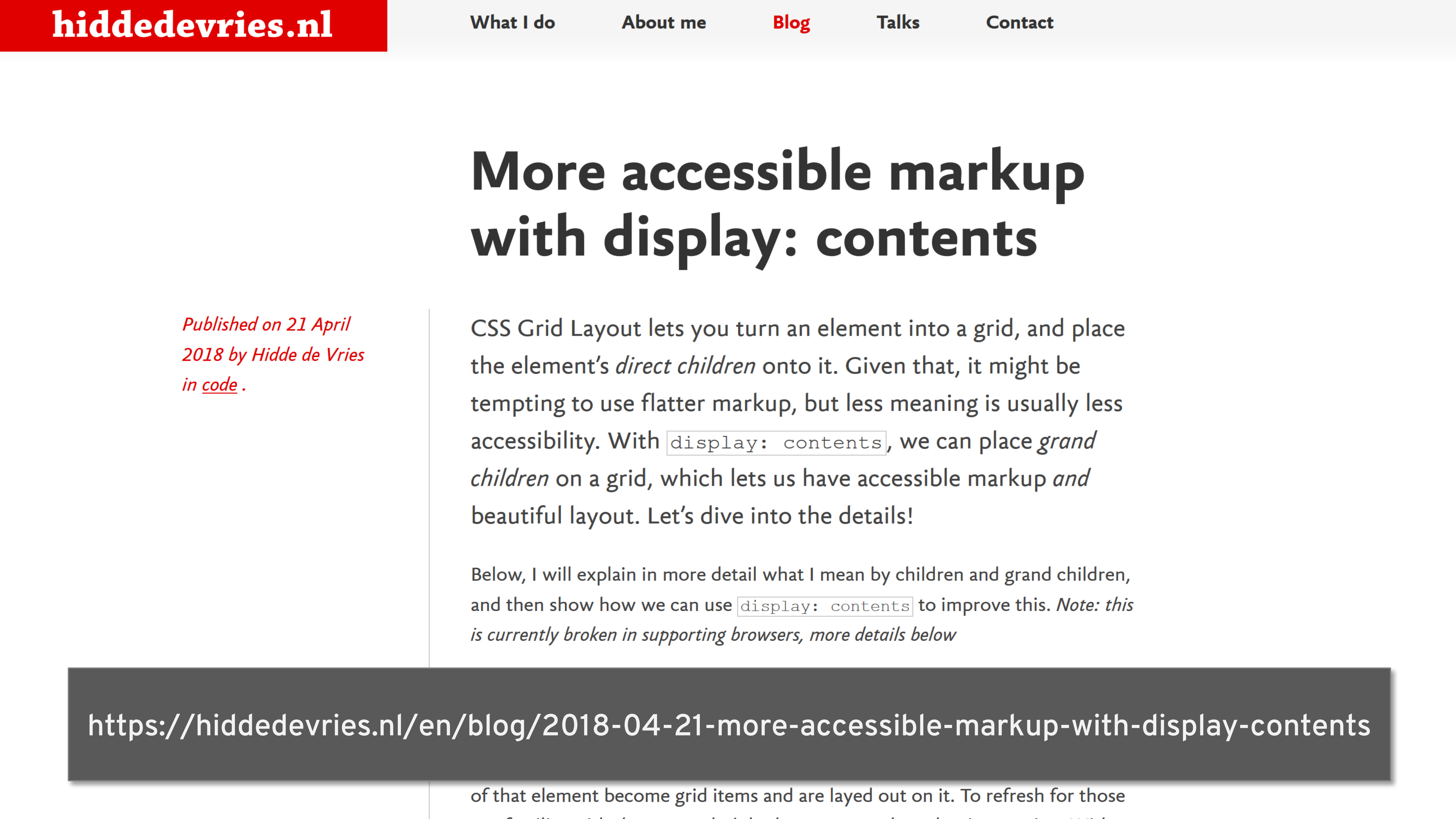
```
<div>
    <a href="/">One</a>
    <br>
    <a href="/">Two</a>
    <br>
    <a href="/">Three</a>
```

https://tink.uk/flexbox-the-keyboard-navigation-disconnect/

```
<div style="display: flex;">
```

display: contents

A cautionary tale.


hiddedevries.nl

What I do About me Blog Talks Contact

More accessible markup with display: contents

Published on 21 April 2018 by Hidde de Vries in code .

CSS Grid Layout lets you turn an element into a grid, and place the element's *direct children* onto it. Given that, it might be tempting to use flatter markup, but less meaning is usually less accessibility. With `display: contents`, we can place *grand children* on a grid, which lets us have accessible markup *and* beautiful layout. Let's dive into the details!

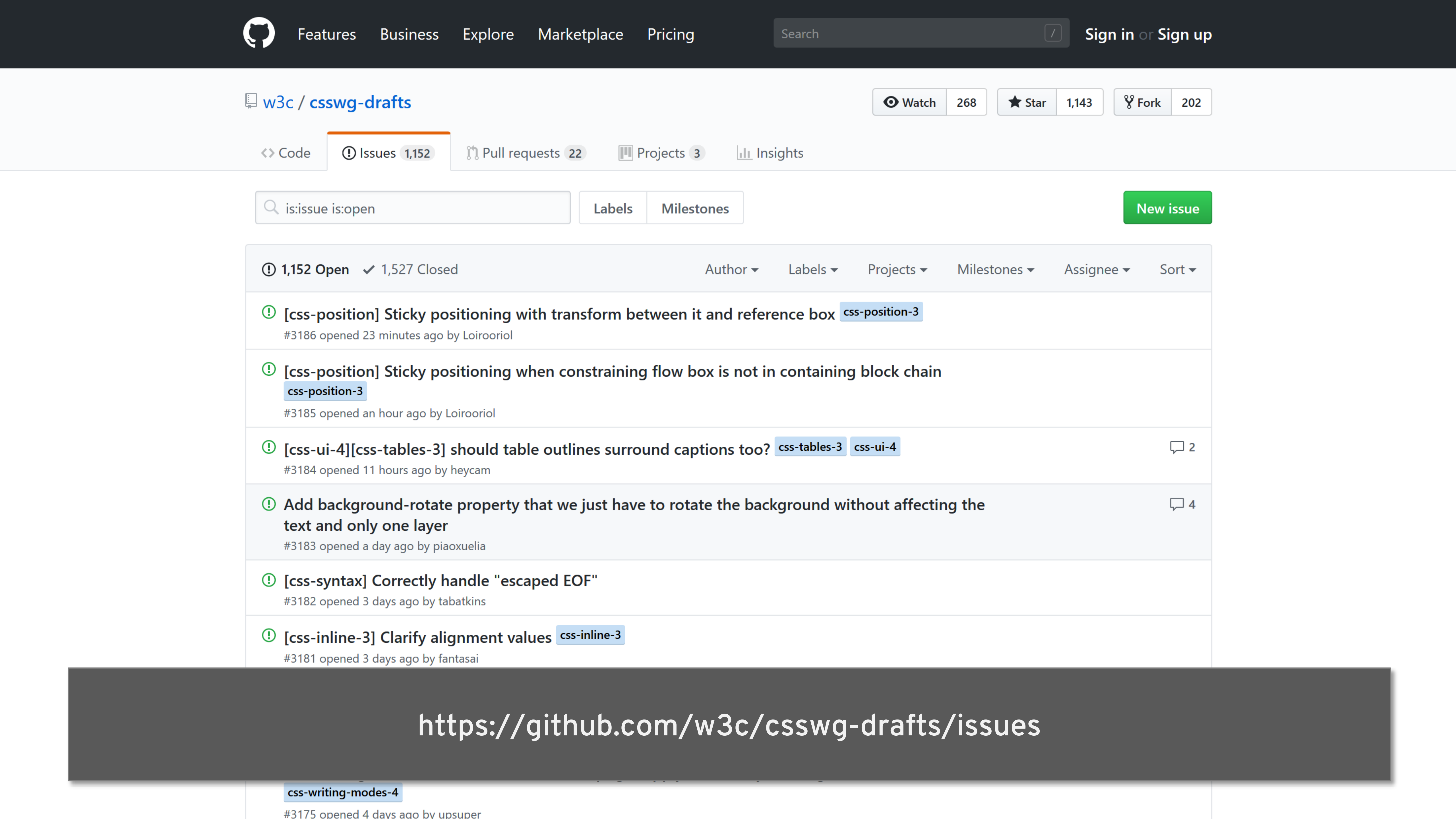
Below, I will explain in more detail what I mean by children and grand children, and then show how we can use `display: contents` to improve this. *Note: this is currently broken in supporting browsers, more details below*

https://hiddedevries.nl/en/blog/2018-04-21-more-accessible-markup-with-display-contents

of that element become grid items and are layed out on it. To refresh for those

What’s next?

What do you need?



w3c / csswg-drafts

is:issue is:open

1,152 Open ✓ 1,527 Closed

[css-position] Sticky positioning with transform between it and reference box css-position-3
#3186 opened 23 minutes ago by Loirooriol

[css-position] Sticky positioning when constraining flow box is not in containing block chain css-position-3
#3185 opened an hour ago by Loirooriol

[css-ui-4][css-tables-3] should table outlines surround captions too? css-tables-3 css-ui-4 2
#3184 opened 11 hours ago by heycam

Add background-rotate property that we just have to rotate the background without affecting the text and only one layer 4
#3183 opened a day ago by piaoxuelia

[css-syntax] Correctly handle "escaped EOF"
#3182 opened 3 days ago by tabatkins

[css-inline-3] Clarify alignment values css-inline-3
#3181 opened 3 days ago by fantasai

css-writing-modes-4
#3175 opened 4 days ago by upsuper

https://github.com/w3c/csswg-drafts/issues

Firefox 62

CSS Shapes, Shape Path Editor, Variable Fonts

https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Releases/62

Firefox 63

gap, row-gap and column-gap in Flex Layout

https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Releases/63

Chrome 69

Scroll Snap, display cutouts, conic gradients

https://developers.google.com/web/updates/2018/09/nic69

Edge HTML 17

Variable fonts

https://blogs.windows.com/msedgedev/2018/04/30/edgehtml-17-april-2018-update/

Thank you

https://noti.st/rachelandrew/es94DQ