

IMPROVING WEBSITE PERFORMANCE WITH CSS CONTAINMENT

MANUEL REGO CASASNOVAS (@regocas)

CSSconf EU / 31 May 2019 (Berlin)



ABOUT ME

Igalia Web Platform Team

Web engines hacker working on Chromium/Blink and
Safari/WebKit

Member of **CSS Working Group** since 2017



igalia



CSS CONTAINMENT



contain

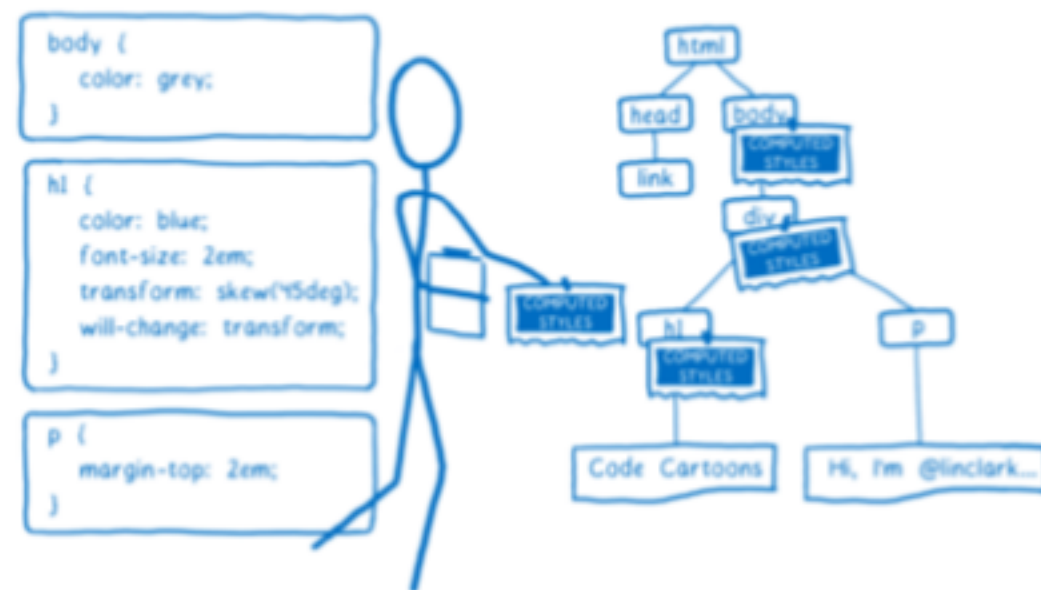


- **layout**
- **paint**
- **size**
- **style**

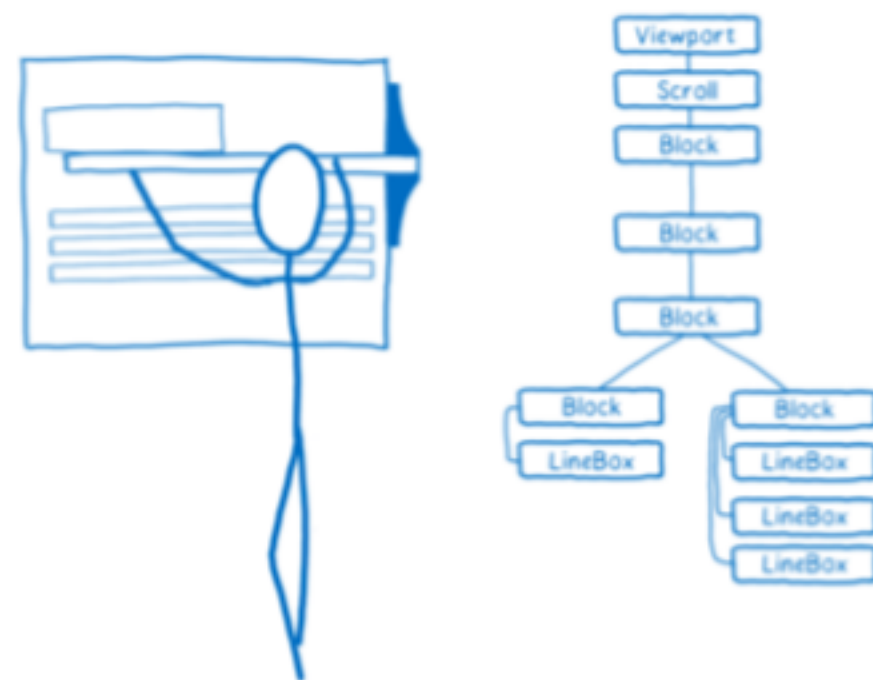


RENDERING PIPELINE

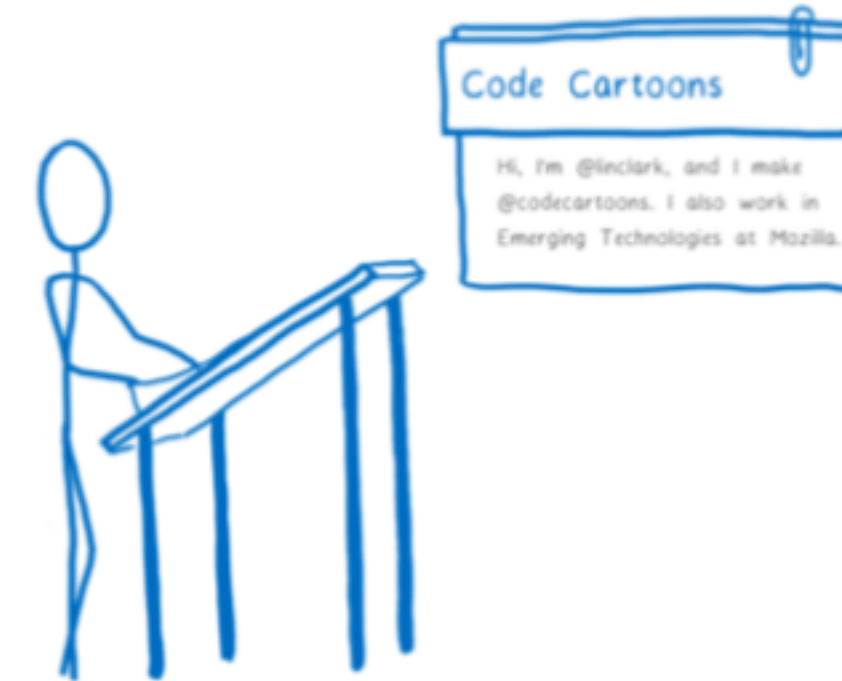
STYLE



LAYOUT



PAINT



COMPOSITE & RENDER



TABLE OF CONTENTS

1	Introduction
2	Strong Containment: the 'contain' property
3	Types of Containment
3.1	Size Containment
3.2	Layout Containment
3.3	Style Containment
3.4	Paint Containment
4	Privacy and Security Considerations
	Appendix A. Changes
	Changes from the Candidate Recommendation of 08 November 2018
	Changes from the Candidate Recommendation of 24 May 2018
	Changes from the Candidate Recommendation of 8 August 2017
	Changes from the Working Draft of 19 April 2017
	Changes from the First Public Working Draft of 21 February 2017
	Conformance
	Document conventions
	Conformance classes
	Requirements for Responsible Implementation of CSS

CSS Containment Module Level 1

Editor's Draft, 9 May 2019

▶ 146 Tests ×



► Specification Metadata

[Copyright](#) © 2019 [W3C](#)® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

CSS CONTAINMENT

This CSS module describes the 'contain' property, which indicates that the element's subtree is independent of the rest of the page. This enables heavy optimizations by user agents when used well.

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

Status of this document

This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress.

[GitHub Issues](#) are preferred for discussion of this specification. When filing an issue, please put the text "css-contain" in the title, preferably like this: "[css-contain] ...summary of comment...". All issues and comments are [archived](#), and there is also a [historical archive](#).

INTRODUCTION

- Improve rendering performance by **isolating subtrees**
- Changes inside that subtree don't get propagated outside



INTRODUCTION

"contain property doesn't block things going into the subtree, it blocks things leaking out"

by [Tab Atkins](#)



contain PROPERTY

- layout
- paint
- size
- style



[illegible]

DESCRIPTION

- Identify independent parts of the website in complex DOM and tag them so the browser can avoid some unneeded work
- Browser engines implement optimizations based on css-contain information
- Using it implies some restrictions



TYPES OF CONTAINMENT



LAYOUT CONTAINMENT

- **layout:** The internal layout is isolated from the rest of the page



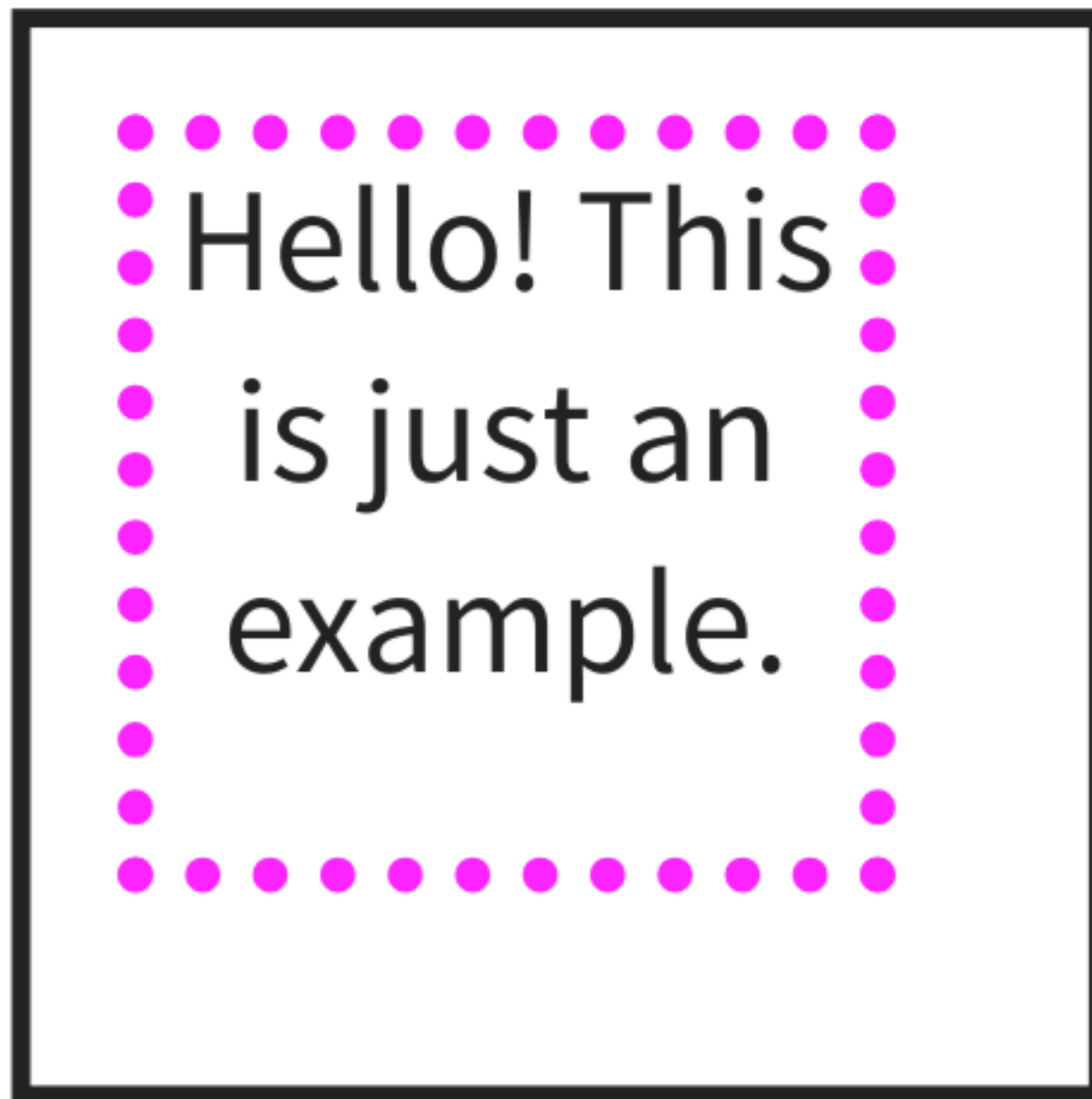
LAYOUT CONTAINMENT

- Any overflow is treated as *ink overflow*
- Containing block for abspos/fixed positioned elements
- Stacking context
- Independent formatting context
- No baseline

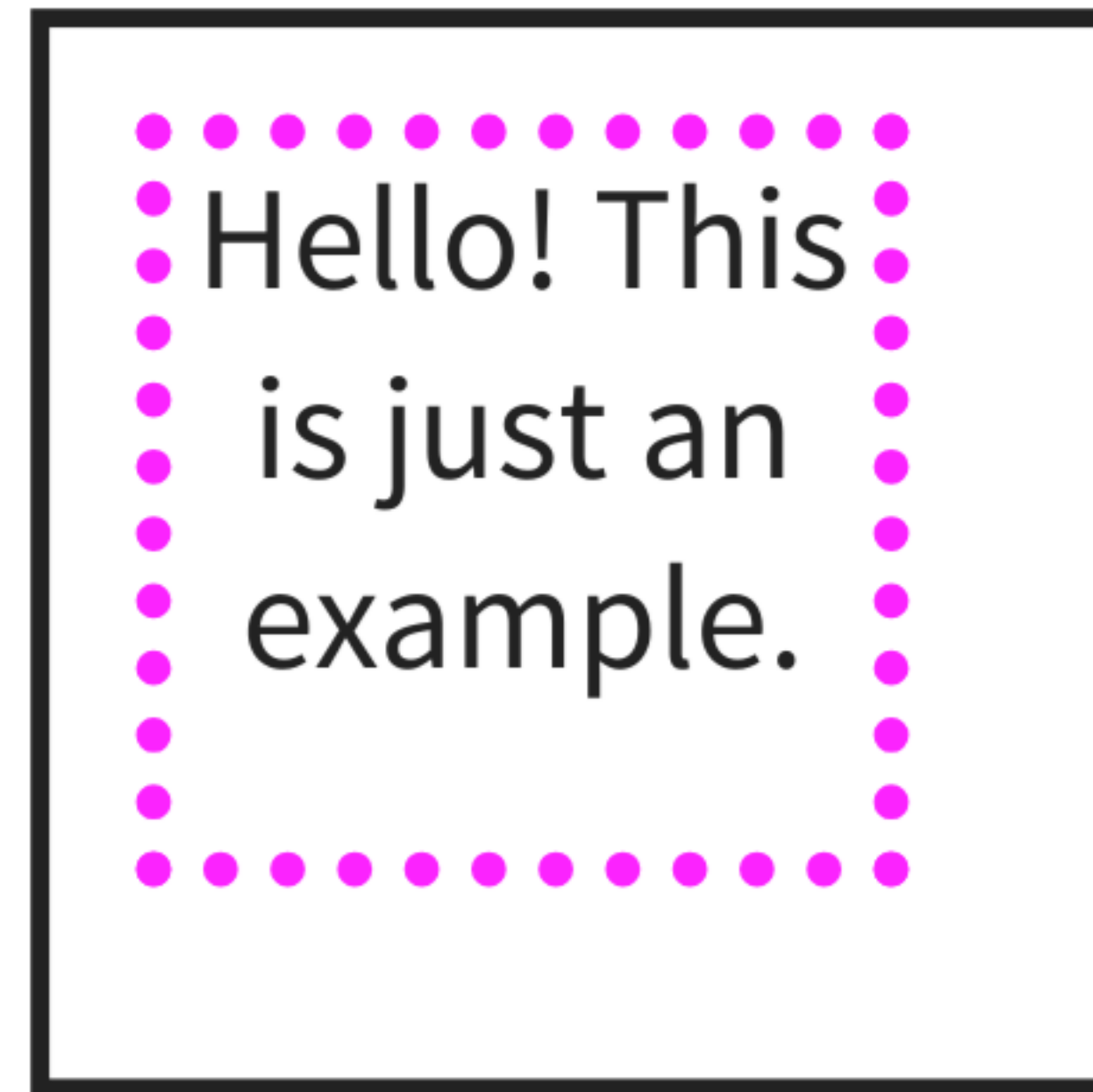


EXAMPLE: INK OVERFLOW

contain: none

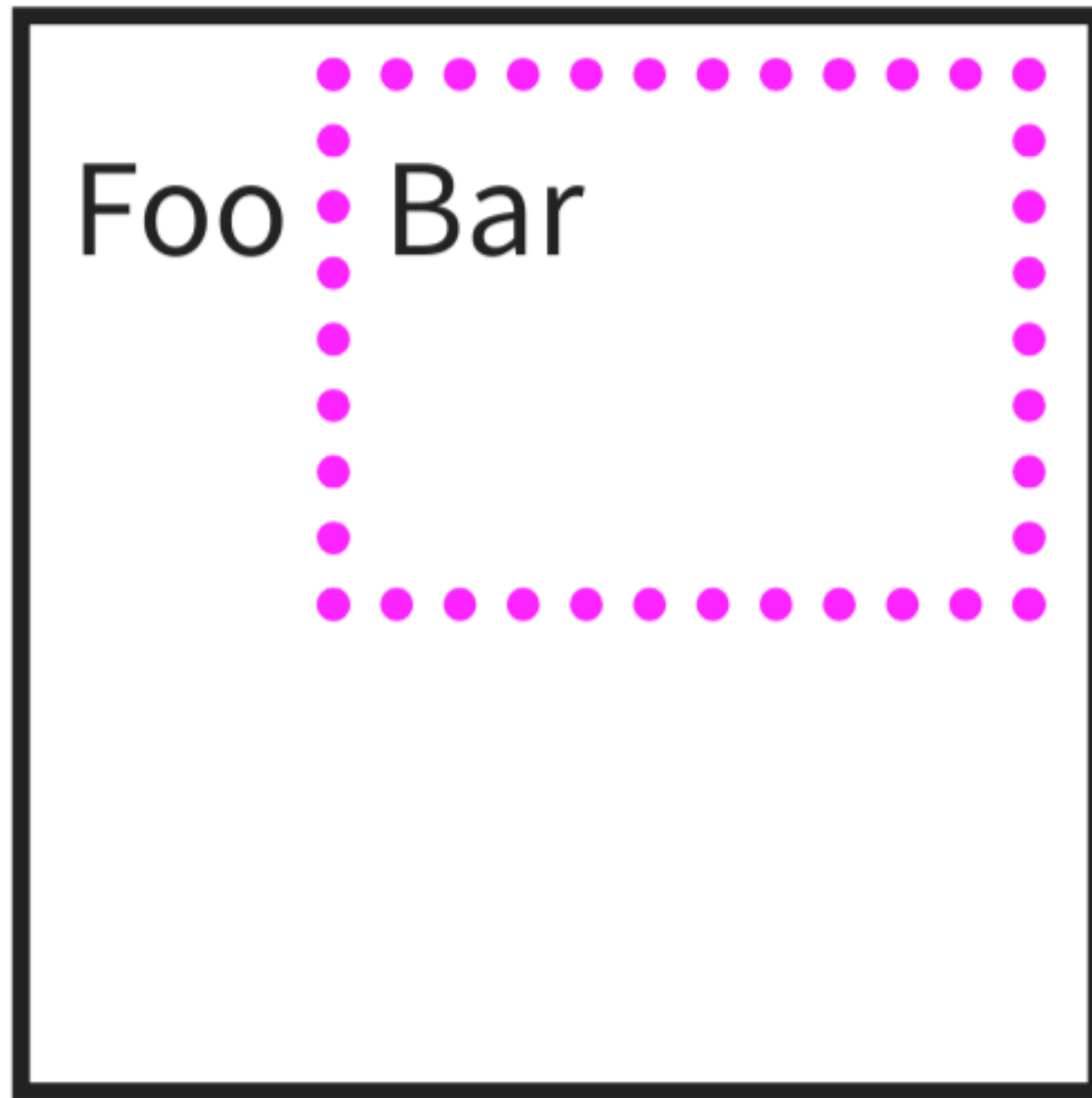


contain: layout

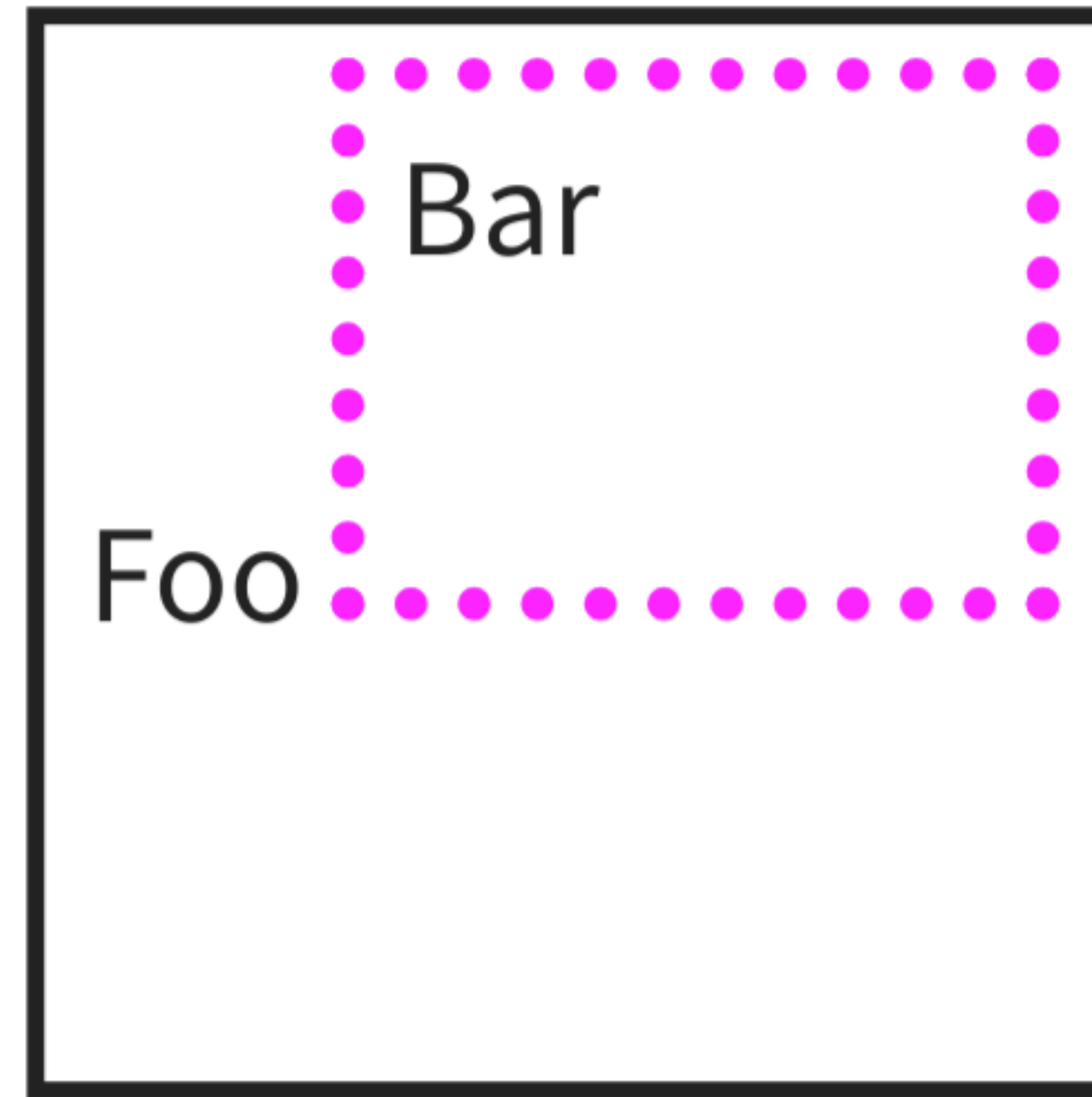


EXAMPLE: BASELINE

contain: none



contain: layout



PAINT CONTAINMENT

- **paint**: Descendants cannot be displayed outside the bounds of the element (similar to `overflow: hidden`)



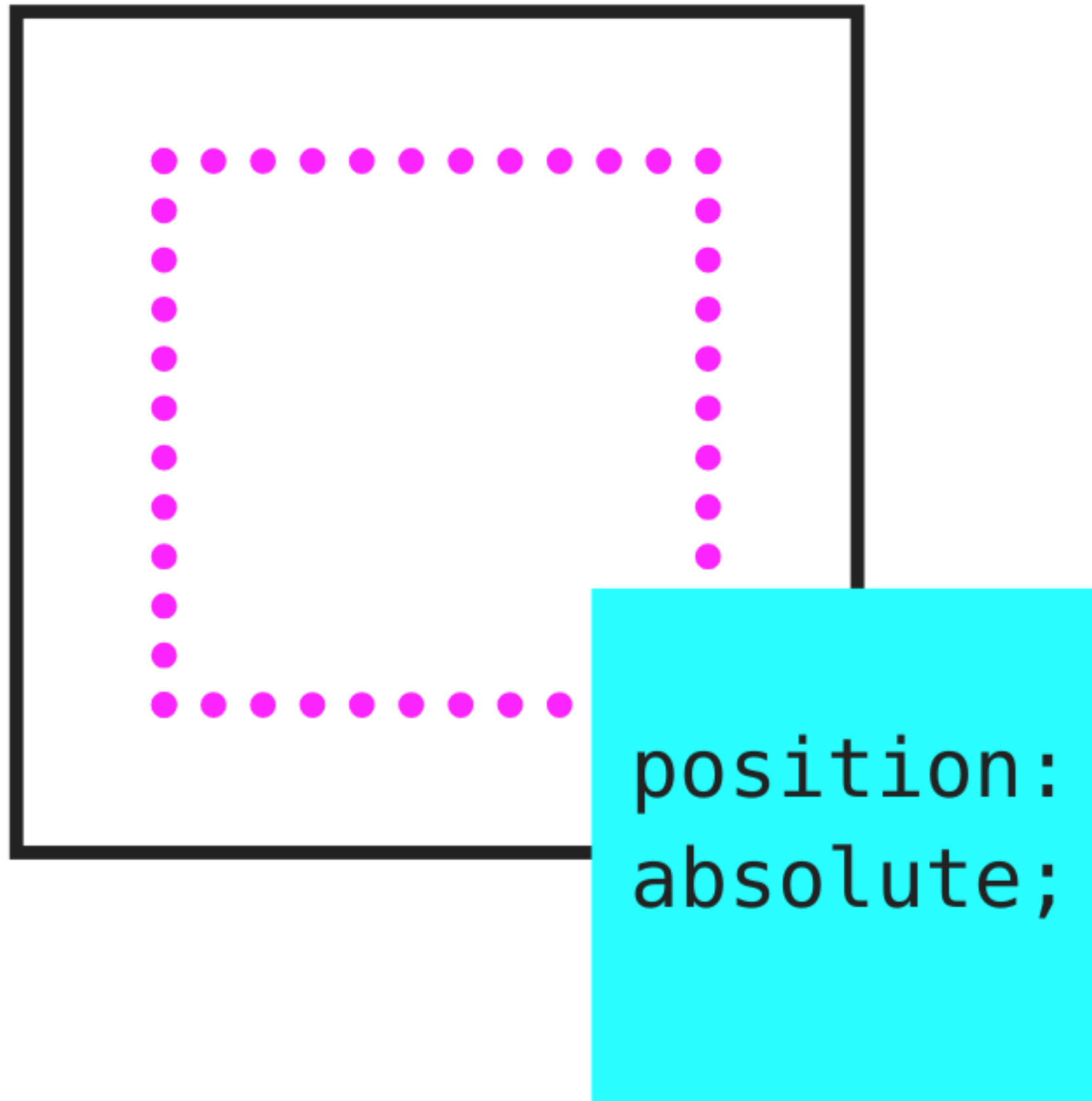
PAINT CONTAINMENT

- Contents are clipped
- Containing block for abspos/fixed positioned elements
- Stacking context
- Independent formatting context

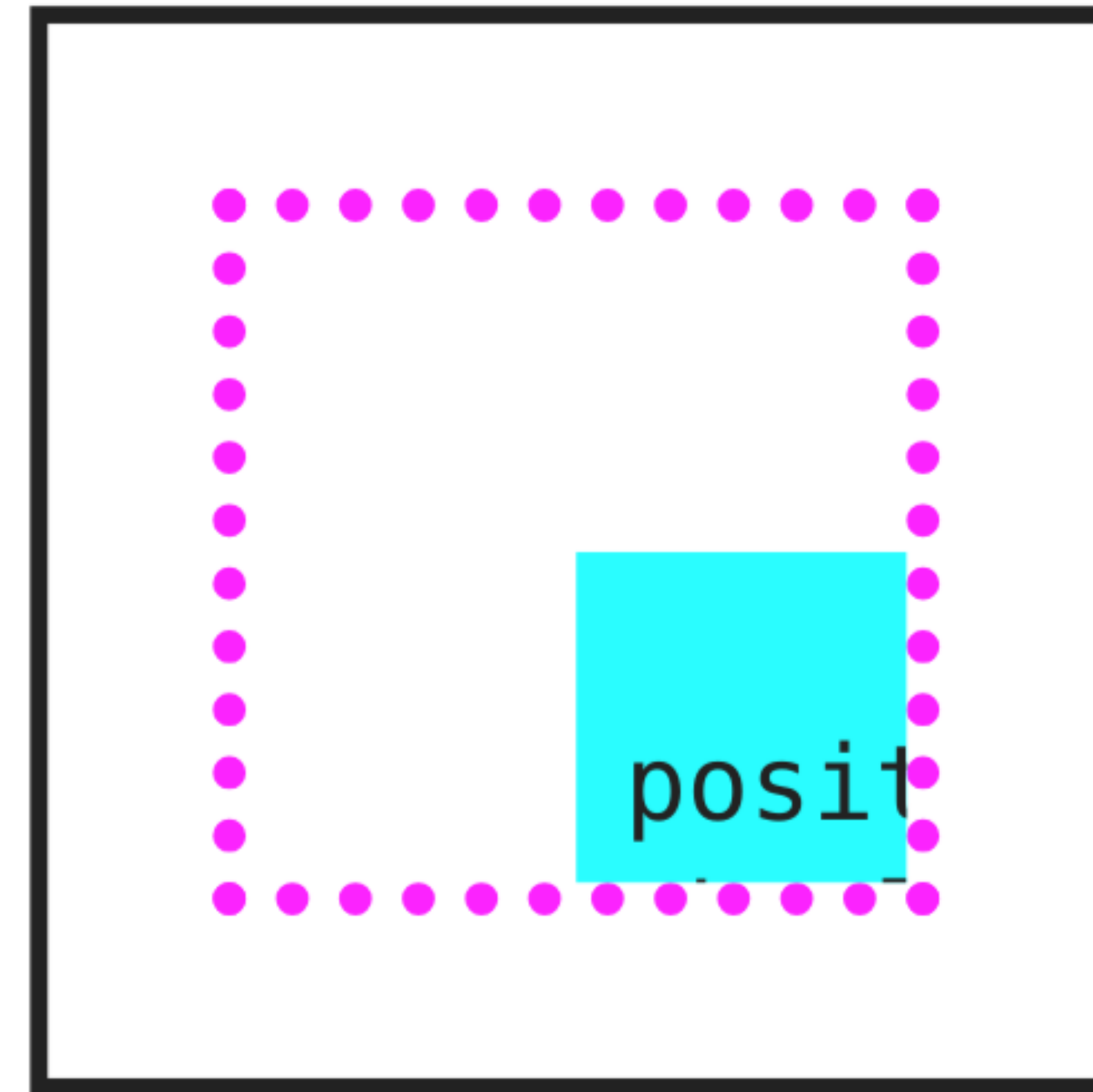


EXAMPLE: CLIPPING AND ABSPOS

`contain: none`



`contain: paint`



SIZE CONTAINMENT

- **size:** Size of the element can be computed without checking its children



SIZE CONTAINMENT

- Treated as having no contents
- Monolithic



EXAMPLE: SIZE CONTAINMENT

`contain: none`



`contain: size`



STYLE CONTAINMENT

- **style**: counters and quotes cannot escape this element (*at risk*)



STYLE CONTAINMENT

- Counters and quotes scoped to the subtree
- Create new counters



EXAMPLE: COUNTERS

```
.counter {  
  counter-increment: n;  
}  
  
pre::after {  
  content: counters(n, ".");  
}
```

```
<div class="counter">  
  <pre>Before: </pre>  
  <div style="contain: style;">  
    <div class="counter"></div>  
    <pre>Inside: </pre>  
  </div>  
  <pre>After: </pre>  
</div>
```



EXAMPLE: COUNTERS

```
.counter {  
  counter-increment: n;  
}  
  
pre::after {  
  content: counters(n, ".");  
}
```

```
<div class="counter">  
  <pre>Before: </pre>  
  <div style="contain: style;">  
    <div class="counter"></div>  
    <pre>Inside: </pre>  
  </div>  
  <pre>After: </pre>  
</div>
```

Before: 1

Inside: 1.1

After: 1



EXAMPLE QUOTES

```
body {  
  quotes: "«" "»" "<" ">";  
}  
.open-quote:before {  
  content: open-quote;  
}  
.close-quote:after {  
  content: close-quote;  
}
```

```
<span class="open-quote">  
  foo  
  <span style="contain: style;">  
    <span class="open-quote">  
      bar  
    </span>  
  </span>  
</span>  
  
<span class="close-quote"></span>
```



EXAMPLE QUOTES

```
body {  
  quotes: "«" "»" "‹" "›";  
}  
.open-quote:before {  
  content: open-quote;  
}  
.close-quote:after {  
  content: close-quote;  
}
```

```
<span class="open-quote">  
  foo  
  <span style="contain: style;">  
    <span class="open-quote">  
      bar  
    </span>  
  </span>  
</span>  
  
<span class="close-quote"></span>
```

« foo ‹ bar »



ALIASES

- **content:** layout, paint and ~~style~~
- **strict:** layout, paint, size and ~~style~~



STATUS



STATUS

- Not "new". Chrome shipping it since M52 (July 2016)
- During 2018 spec has been pushed (by [Florian Rivoal](#)) and test suite has been completed (by [G rard Talbot](#))
- Chromium implementation has been fixed to align with the spec (by Igalia)



STATUS

- Firefox is working on the implementation too (currently behind a runtime flag `layout.css.contain.enabled`)
- Chromium has implemented some optimizations based on `css-contain`



Can I use

? ⚙ Settings

x | Feature: CSS Containment

CSS Containment 📄 - CR

Usage % of all users ⬆ ?
Global 70.35%

The CSS `contain` property lets developers limit the scope of the browser's styles, layout and paint work for faster and more efficient rendering.

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	U f	
		2 - 40	4 - 50		10 - 37										
		2 41 - 51	1 51		1 38 - 39										
6 - 10	12 - 17	2 52 - 65	52 - 73	3.1 - 12	40 - 57	3.2 - 12.1		2.1 - 4.4.4	7	12 - 12.1			10		
11	18	2 66	74	12.1	58	12.2	all	67	10	46	74	2 66	11		
	75	2 67 - 68	75 - 77	TP											

Source: <https://caniuse.com/css-containment>

EXAMPLES



EXAMPLE

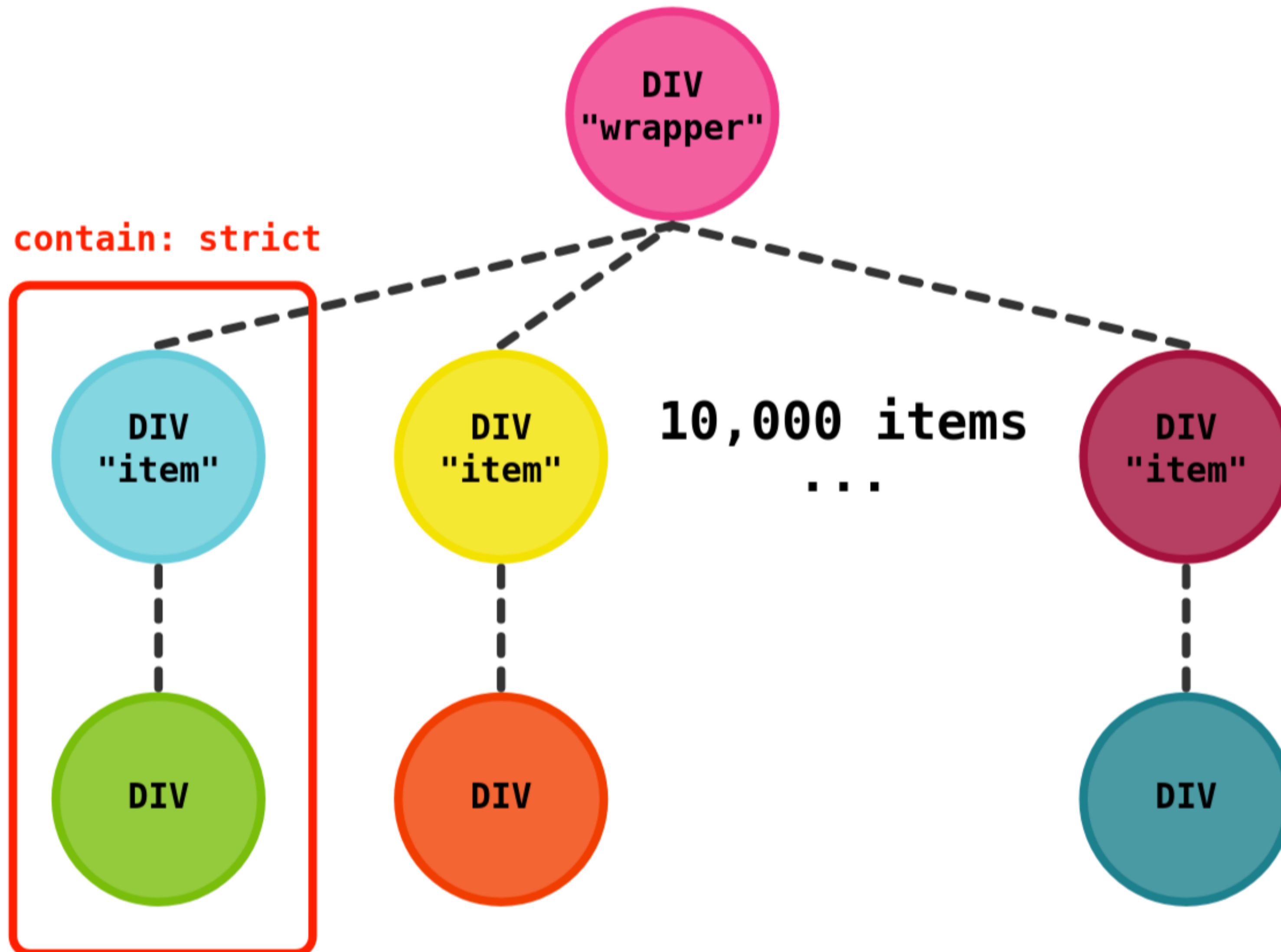
- 10,000 DOM elements:

```
<div class="item">  
  <div>Lorem ipsum...</div>  
</div>
```

- First item's child is modified via `textContent` attribute



DOM Tree



CSS Containment Example

Hello World!

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

Lorem ipsum...

FIRST RUN (contain: none)

=====

- ```

1) Change "targetInner" content... Time (ms): 5.610000000160653
2) Change "targetInner" content... Time (ms): 6.65000000026339
3) Change "targetInner" content... Time (ms): 6.124999999883585
4) Change "targetInner" content... Time (ms): 8.015000000341388
5) Change "targetInner" content... Time (ms): 6.8099999998594285
6) Change "targetInner" content... Time (ms): 8.17499999937427
7) Change "targetInner" content... Time (ms): 5.804999999782012
8) Change "targetInner" content... Time (ms): 5.529999999907886
9) Change "targetInner" content... Time (ms): 5.59999999958527
10) Change "targetInner" content... Time (ms): 5.245000000286382

```

```
==> AVERAGE (ms): 6.356500000038068
```

SECOND RUN (contain: strict)

=====

- ```
1) Change "targetInner" content... Time (ms): 0.33000000030369847
2) Change "targetInner" content... Time (ms): 0.180000000000029104
3) Change "targetInner" content... Time (ms): 0.17500000012660166
4) Change "targetInner" content... Time (ms): 0.1749999996718543
5) Change "targetInner" content... Time (ms): 0.1749999996718543
6) Change "targetInner" content... Time (ms): 0.1700000002529123
7) Change "targetInner" content... Time (ms): 0.1649999992447556
8) Change "targetInner" content... Time (ms): 0.16999999979816494
9) Change "targetInner" content... Time (ms): 0.1649999992447556
10) Change "targetInner" content... Time (ms): 0.16999999979816494
```

```
==> AVERAGE (ms): 0.1874999999472493
```

PERFORMANCE IMPROVEMENT: 97% (34x)



CSS Containment Example

10000 items

12x



Title 0

BYE



Title 1

Lorem ipsum...



Title 2

Lorem ipsum...



Title 3

Lorem ipsum...

Chromium

CSS Containment Example

10000 items

2703x



Title 0

BYE



Title 1

Lorem ipsum...



Title 2

Lorem ipsum...



Title 3

Lorem ipsum...

Firefox



OPTIMIZATIONS

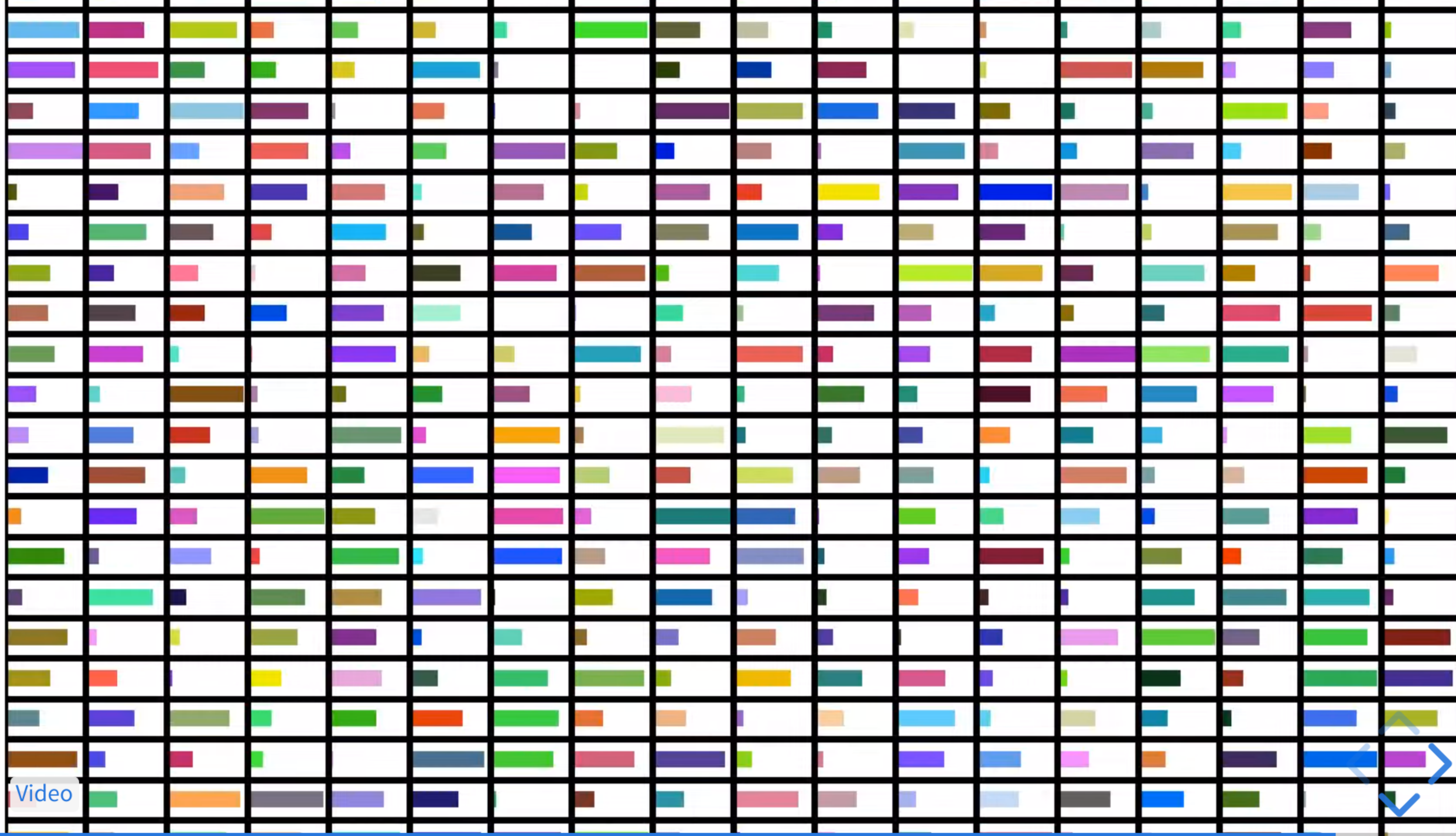


39.28	57.45	60.04	58.96	15.48	75.43	20.57	11.88	80.43	26.84	0.18	23.82	81.88	86.40	19.21
81.33	57.77	59.10	28.86	68.32	94.17	31.67	5.16	66.78	0.32	80.58	49.15	2.92	4.69	57.30
19.16	7.23	65.68	18.87	58.48	87.16	40.29	5.50	95.26	6.36	55.38	93.74	84.59	38.40	23.79
17.82	49.56	13.65	26.80	71.39	6.43	17.22	85.58	33.21	53.54	69.93	72.66	51.78	98.77	92.47
98.07	10.45	35.37	37.00	82.49	27.62	88.40	71.99	75.23	1.23	59.71	88.73	87.70	36.36	39.00
92.09	61.82	49.00	46.83	63.04	47.98	3.54	37.21	57.28	51.29	9.47	9.65	26.74	8.42	55.79
45.09	21.88	50.09	33.18	77.30	10.69	9.94	76.13	51.94	78.77	72.56	80.93	96.63	72.06	66.82
55.74	26.14	0.89	73.41	13.06	37.45	73.18	83.41	47.11	30.51	5.63	23.91	86.45	43.45	4.80
88.41	12.57	74.41	16.29	64.99	92.57	67.34	76.74	56.14	49.66	72.90	43.48	10.42	76.55	52.00
44.70	66.97	97.03	57.83	8.56	57.48	74.05	17.99	64.34	67.46	97.58	5.88	54.97	59.23	53.22
25.89	18.68	20.93	37.74	72.85	65.20	94.03	86.17	96.76	22.97	35.60	72.65	99.26	56.41	35.90
52.53	76.56	43.52	89.72	33.38	1.96	2.63	42.08	22.74	41.61	99.21	38.10	58.64	35.97	46.26
18.40	46.34	41.71	45.64	76.23	80.08	55.74	52.67	83.55	54.62	63.33	77.07	85.91	19.85	13.96
93.30	34.22	87.99	62.64	58.74	55.34	49.89	82.96	92.23	12.89	2.62	88.76	51.16	44.26	92.40
33.69	6.26	96.60	97.71	63.39	20.56	28.43	52.52	95.85	18.80	42.29	28.44	40.71	42.29	16.65
21.71	26.11	32.65	23.41	92.14	94.24	79.45	34.27	7.18	26.14	5.90	69.11	75.00	51.56	53.64
48.38	65.96	96.62	99.26	23.96	88.85	32.46	41.61	57.11	46.60	92.43	11.10	70.88	32.70	2.50

OPTIMIZATIONS

- Change TextNode contents using data attribute
- Avoid layout time when replacing one line by another one
- Using `contain: layout size`
- Stress test: 10,000 elements
4x faster





Video



OPTIMIZATIONS

- Big DOM tree, each cell creating its own layer
- Avoid traversing the whole layer tree when something changes inside one of these elements
- Using `contain: strict`
- Stress test: 40,000 elements
1.1x faster



FUTURE



FUTURE

- Better browsers support (Firefox shipping soon, Safari?)
- More optimizations based on css-contain
- New features based on css-contain:
e.g. [Display Locking](#)



THANKS

- Twitter: [@regocas](#)
- Mail: rego@igalia.com
- Blog: <https://blogs.igalia.com/mrego/>

