{codemotion}

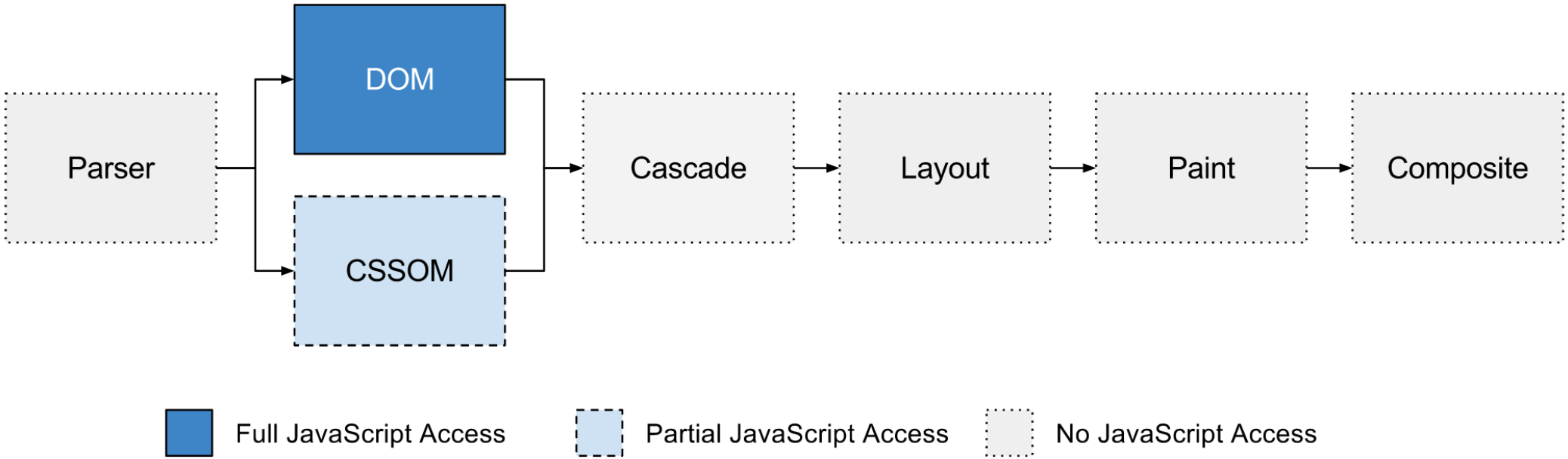
Houdini

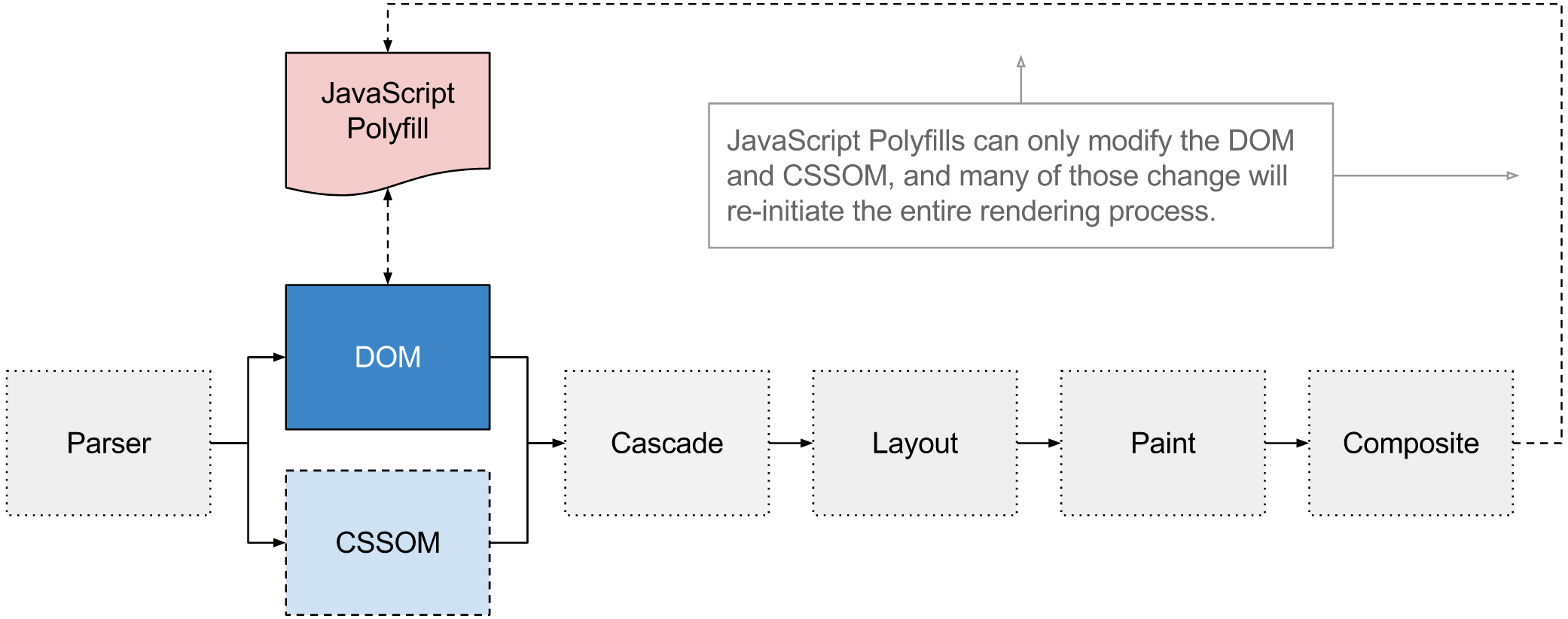
programming in CSS

Madrid | Nov 30 - Dec 1, 2018

Lucas Benjamin

CSS
Ruvim Noga


DOM
Parser
CSSOM
Cascade
Layout
Paint
Composite
Full JavaScript Access
Partial JavaScript Access
No JavaScript Access

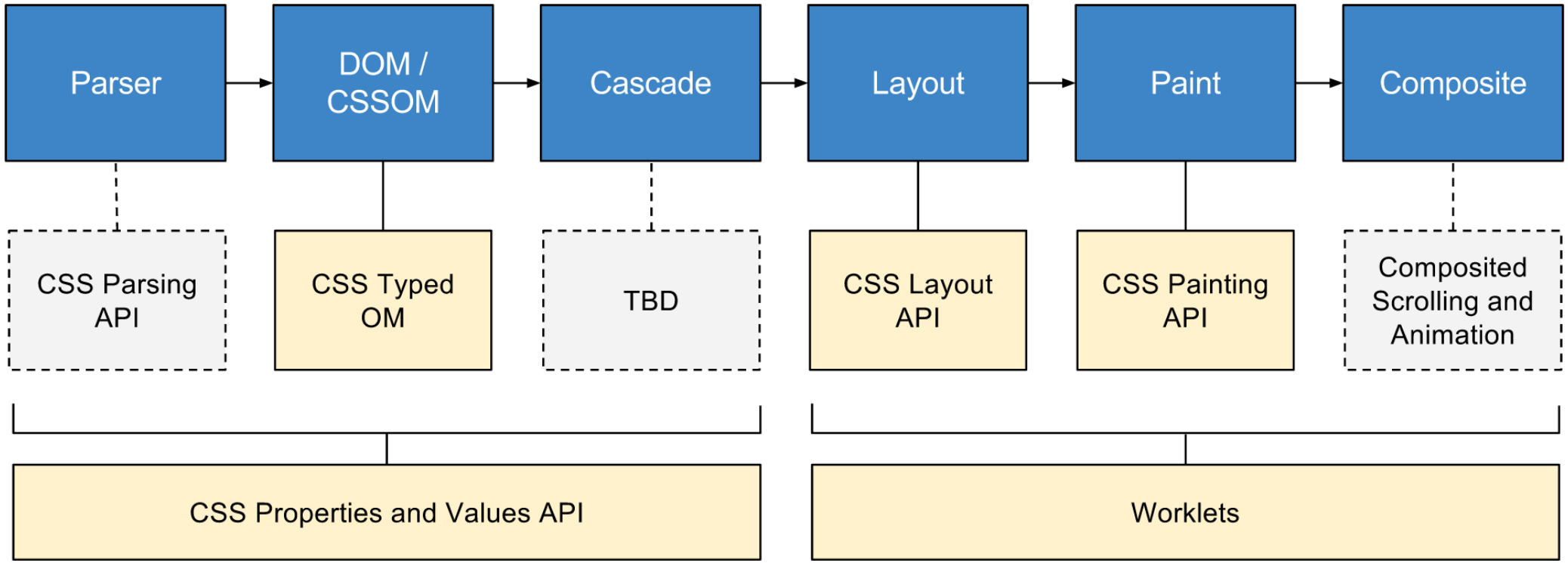

JavaScript Polyfill
JavaScript Polyfills can only modify the DOM and CSSOM, and many of those change will re-initiate the entire rendering process.
DOM
Parser
CSSOM
Cascade
Layout
Paint
Composite

CSS Houdini

Lucas Benjamin

The objective of the CSS-TAG Houdini Task Force (CSS Houdini) is to jointly develop features that explain the “magic” of Styling and Layout on the web. - CSS Houdini

.CSS-Houdini { }


Parser
DOM / CSSOM
Cascade
Layout
Paint
Composite
CSS Parsing API
CSS Typed OM
TBD
CSS Layout API
CSS Painting API
Composited Scrolling and Animation
CSS Properties and Values API
Worklets

.CSS-Houdini { }

Box Tree API

Parser API

Layout API

Paint API

Properties & Values API

Worklets

Typed OM API

Animation Worklet

Font Metrics API

Extends CSS

.Extends-CSS .Worklets { }

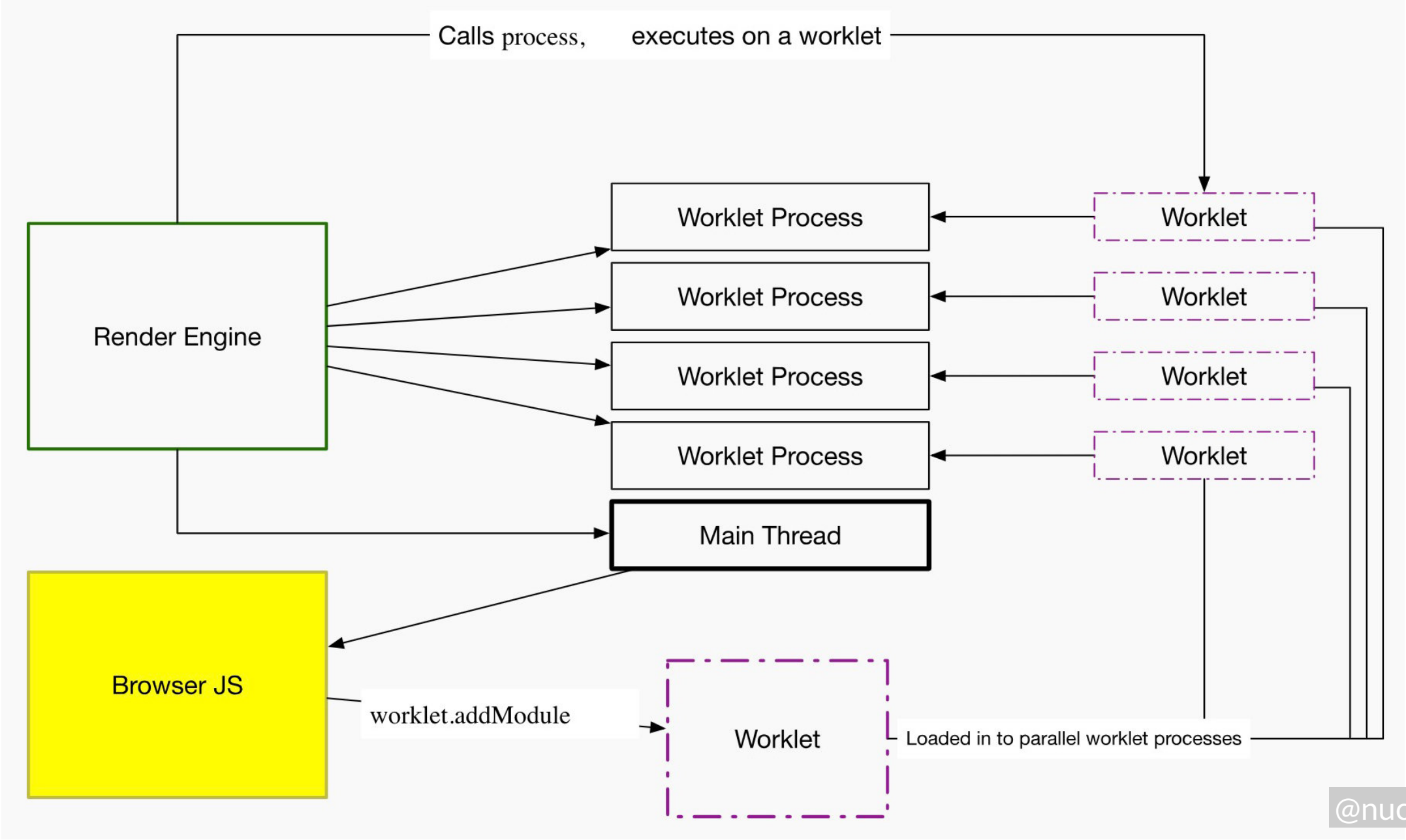
Worklet connects to the browser engine

Is a Javascript Module

Without control over execution

It has no impact on performance

.Extends-CSS .Worklets { }



.Extends-CSS .Worklets { }

js

```
if ('paintWorklet' in CSS) {
    CSS.paintWorklet.addModule('PlaceholderBoxPainter.js');
}
```

worklet

```
class PlaceholderBoxPainter {
    paint(ctx, size) {
        // Magic 🎩
    }
}

registerPaint('placeholder-box', PlaceholderBoxPainter);
```

Paint API

.Extends-CSS .Paint-API{ }

js

```
if ('paintWorklet' in CSS) {
    CSS.paintWorklet.addModule('PlaceholderBoxPainter.js');
}
```

CSS

```
@supports (background: paint(slanted-bg)) {
  .el span {
    background: paint(slanted-bg);
  }
}
```

.Extends-CSS .Paint-API{ }

worklet

```
class SlantedBg {
  constructor () {
    this.hue = 110;
  }
  paint (ctx, geom, props, args) {
    // draw random colors
    ctx.fillStyle = 'hsl(' + this.hue + ', 100%, 50%)';
    this.hue += 10;

    ...
  }
}

registerPaint('slanted-bg', SlantedBg)
```

.Extends-CSS .Paint-API{ }

worklet

```
registerPaint('slanted-bg', class {
  constructor () {
    this.hue = 110;
  }
  paint (ctx, geom, props, args) {
    // draw random colors
    ctx.fillStyle = 'hsl(' + this.hue + ', 100%, 50%)';
    this.hue += 10;

    ...
  }
})
```

Houdini-Paint-API-Circles

https://codesandbox.io/embed/yqm0y9v42v?fontsize=16

Houdini-Paint-API-Circles-props

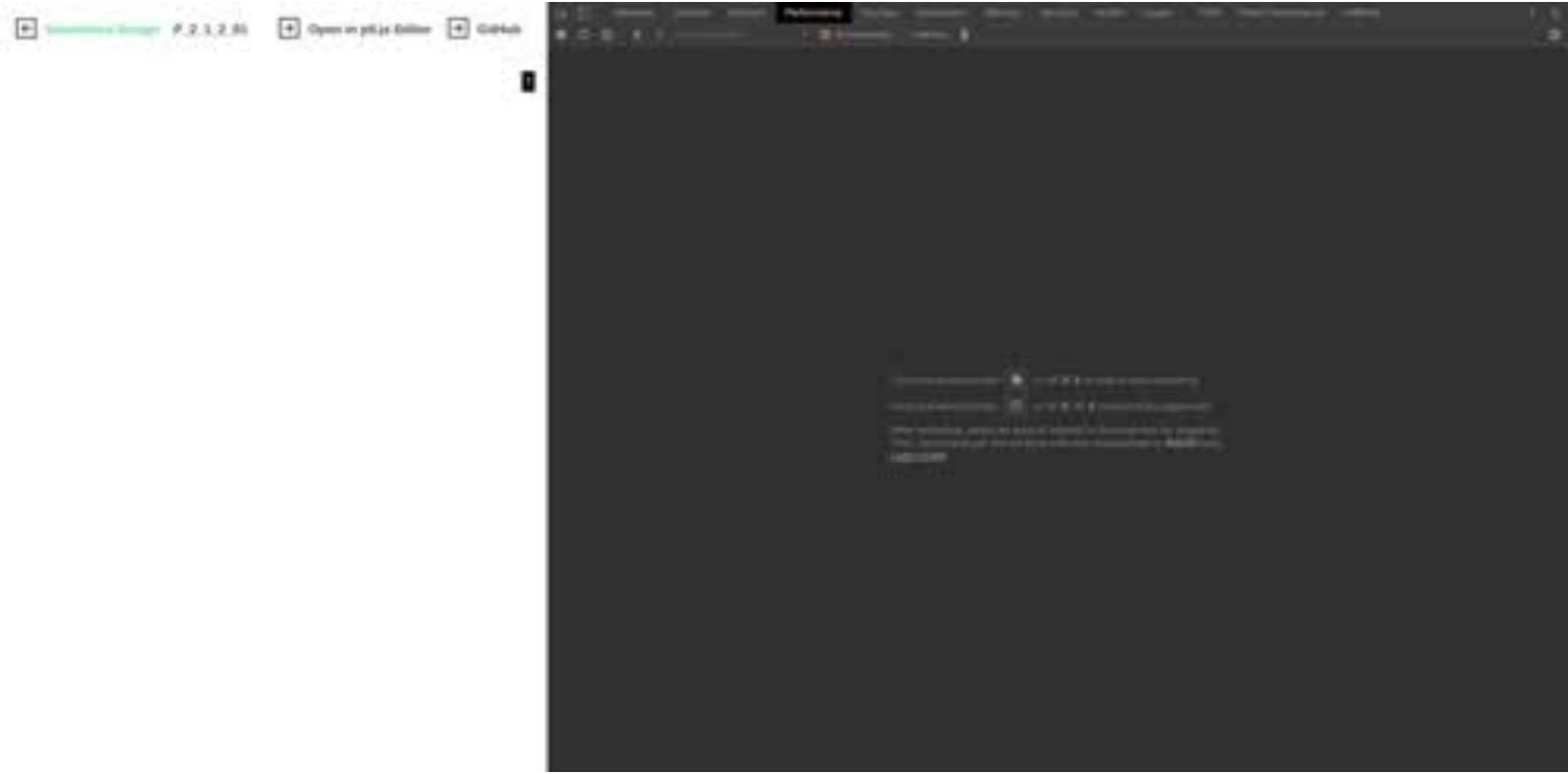
https://codesandbox.io/embed/r74pzln91q

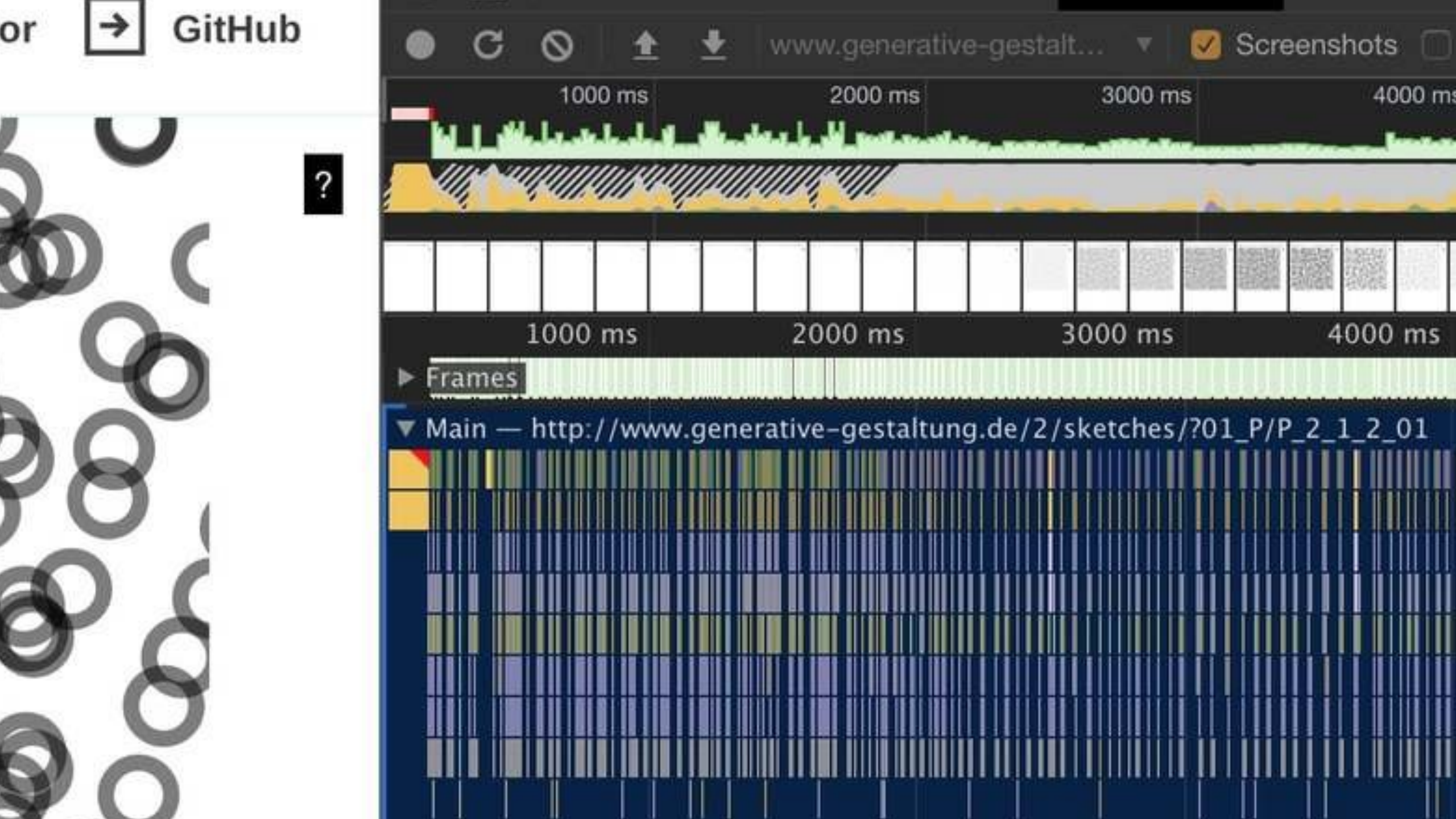
Houdini-Paint-API-Circles-args

https://codesandbox.io/embed/044vkz1k6w

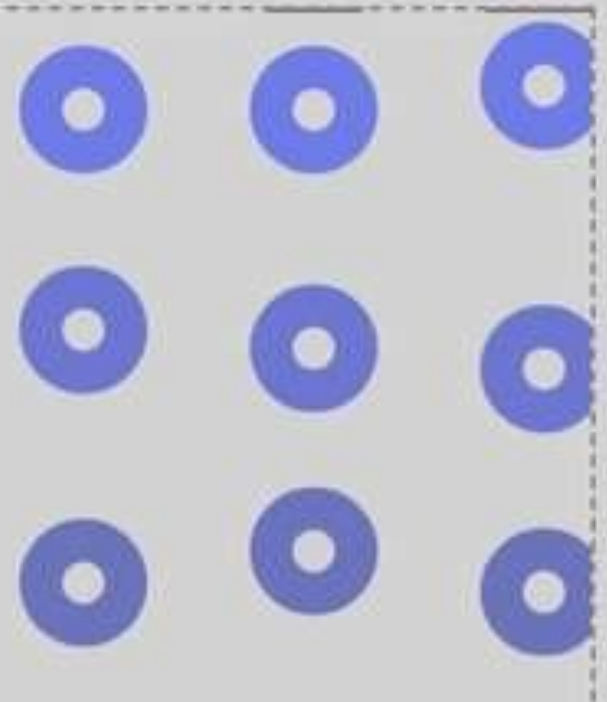
Houdini-Paint-API-Circles-props-JS

https://codesandbox.io/embed/zxyzr4l61l




or
GitHub
?
www.generative-gestalt...
Screenshots
1000 ms
2000 ms
3000 ms
4000 ms
Frames
Main — http://www.generative-gestaltung.de/2/sketches/?01_P/P_2_1_2_01

ouse



127.0.0.1 #1

Screenshots

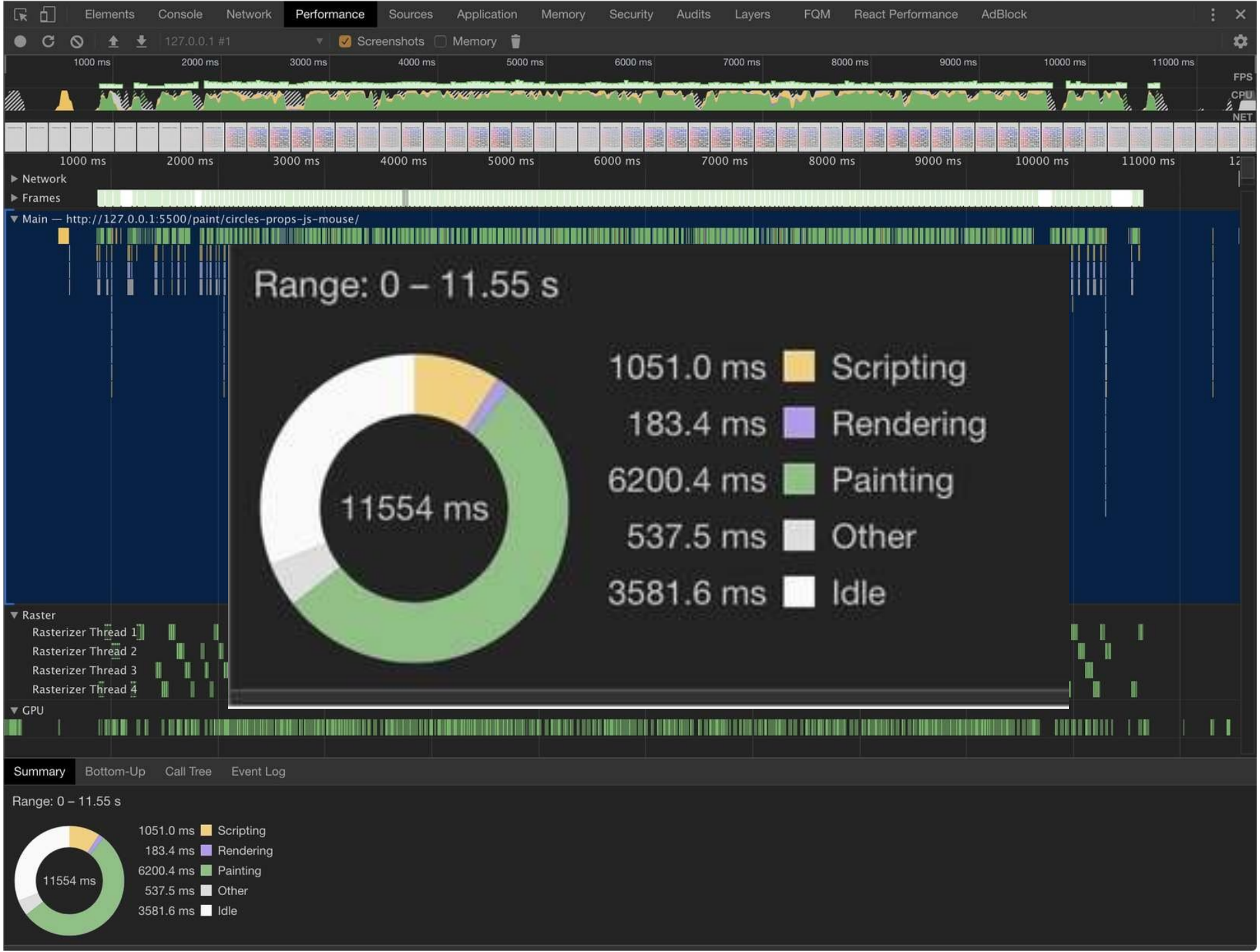
1000 ms 2000 ms 3000 ms 4000 ms

1000 ms 2000 ms 3000 ms 4000 ms

Network

Frames

Main — http://127.0.0.1:5500/paint/circles-props-js-mouse/


Elements
Console
Network
Performance
Sources
Application
Memory
Security
Audits
Layers
FQM
React Performance
AdBlock
127.0.0.1 #1
Screenshots
Memory
1000 ms
2000 ms
3000 ms
4000 ms
5000 ms
6000 ms
7000 ms
8000 ms
9000 ms
10000 ms
11000 ms
FPS
CPU
NET
Network
Frames
Main — http://127.0.0.1:5500/paint/circles-props-js-mouse/
Range: 0 – 11.55 s
11554 ms
1051.0 ms Scripting
183.4 ms Rendering
6200.4 ms Painting
537.5 ms Other
3581.6 ms Idle
Raster
Rasterizer Thread 1
Rasterizer Thread 2
Rasterizer Thread 3
Rasterizer Thread 4
GPU
Summary
Bottom-Up
Call Tree
Event Log
Range: 0 – 11.55 s
11554 ms
1051.0 ms Scripting
183.4 ms Rendering
6200.4 ms Painting
537.5 ms Other
3581.6 ms Idle

Properties & Values API

.Extends-CSS .Properties-Values-API { }

js

```
CSS.registerProperty({
  name: '--line-with',
  syntax: '<number>',
  inherits: false,
  initialValue: 1
});
```

CSS Properties Syntax

'<length>' accepts length values.
'<length> | <percentage>' accepts lengths, percentages, percentage calc expressions, and length calc expressions, but not calc expressions containing a combination of length and percentage values.
'<length-percentage>' accepts all values that "<length> | <percentage>" would accept, as well as calc expressions containing a combination of both length..
'big | bigger | BIGGER' accepts the ident "big", or "bigger", or "BIGGER".
'<length>+' accepts a space-separated list of length values.

CSS Properties Syntax

'<length>' accepts length values.
'<length> | <percentage>' accepts lengths, percentage
expressions, and length calc expressions, but not calc e
combination of length and percentage values.
'<length-percentage>' accepts all values that "<length
accept, as well as calc expressions containing a combin
'big | bigger | BIGGER' accepts the ident "big", or "bigge
'<length>+' accepts a space-separated list of length va

'<length>'
'<number>'
'<percentage>'
'<length-percentage>'
'<color>'
'<image>'
'<url>'
'<integer>'
'<angle>'
'<time>'
'<resolution>'
'<transform-function>'
'<custom-ident>'

[css-properties-values-api/#supported-syntax-strings]()

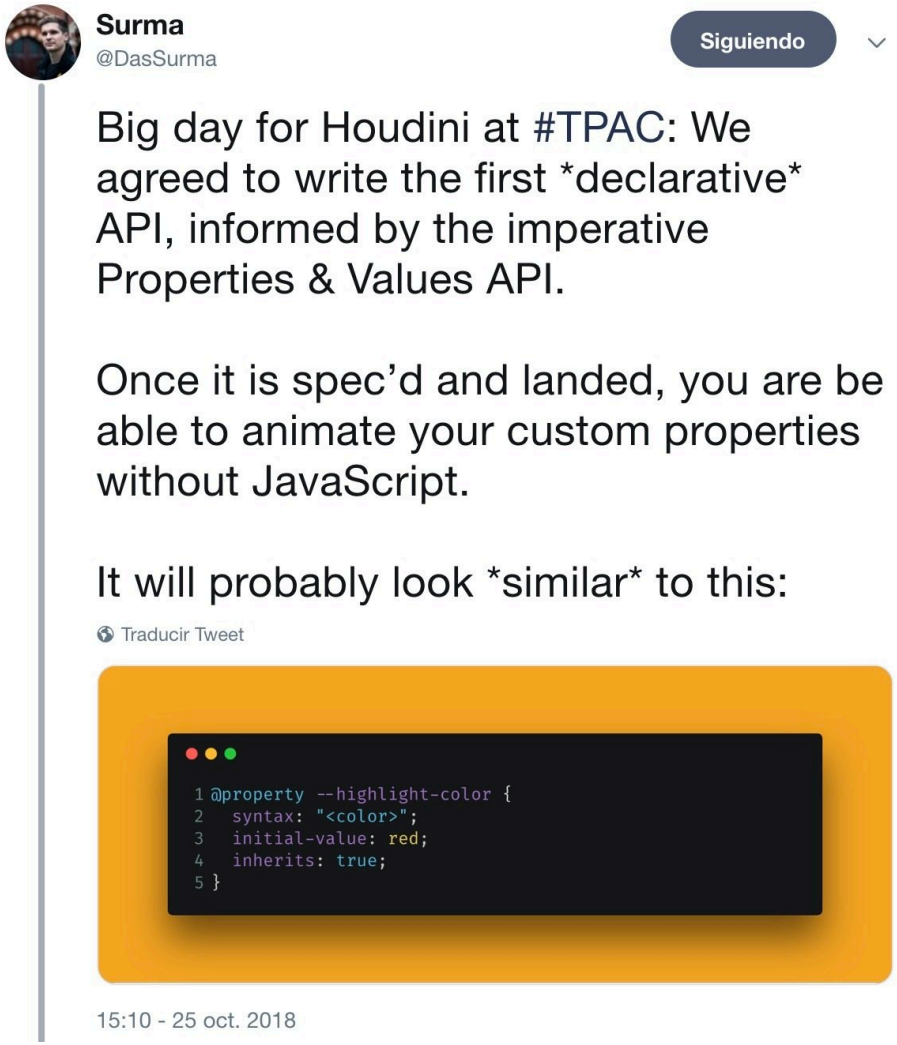
.Extends-CSS .Properties-Values-API { }

https://codesandbox.io/embed/92znj0j9ow

.Extends-CSS .Properties-Values-API { }

//codepen.io/thebabydino/embed/MrZvNm/?height=265&theme-id=light&default-tab=css,result

.Extends-CSS .Properties-Values-API { }



.Extends-CSS .Properties-Values-API { }

```
1 @property --highlight-color {
2   syntax: "<color>";
3   initial-value: red;
4   inherits: true;
5 }
```

15:10 - 25 oct. 2018

.Extends-CSS .Properties-Values-API { }

PostCSS

input

```
@property --theme {
  syntax: '<color>+';
  initial-value: #fff;
  inherits: true;
}
```

output

```
if ("registerProperty" in CSS)
  CSS.registerProperty({
    name: "--theme",
    syntax: "<color>+",
    initialValue: "#fff",
    inherits: true
  });
}
```

postcss-register-custom-props (Vitalii Bobrov)

.Extends-CSS .Properties-Values-API { }

PostCSS

input

```
@property --highlight-color {
  inherits: true;
  initial-value: red;
  syntax: "<color>";
}

@property --gap-spacing {
  inherits: false;
  initial-value: 1em;
  syntax: "<length-percentage>";
}
```

output

```
[
  {
    "name": "--highlight-color",
    "inherits": true,
    "initialValue": "red",
    "syntax": "<color>"
  },
  {
    "name": "--gap-spacing",
    "initialValue": "1em",
    "syntax": "<length-percentage>"
  }
]
```

postcss-register-property (Jonathan Neal)

.Extends-CSS .Properties-Values-API { }

PostCSS

input

```
@property --highlight-color {
  inherit
  initial
  syntax:
}

@property
  inherit
  initial
  syntax:
}
```

output

```
[
  {
    "name": "--highlight-color",
    ...
    ...ntage>"
]
```

js

```
import properties from './styles.css.properties.json';

if (window.CSS && CSS.registerProperty) {
  for (const descriptor of properties) {
    CSS.registerProperty(descriptor);
  }
}
```

postcss-register-property (Jonathan Neal)

Typed OM API

.Extends-CSS .Typed-OM-API { }

The **Type OM API** is an extension of the existing CSS object model (CSSOM) that exposes CSS values as typed JavaScript objects, rather than a simple string of text as they are today.

With **Type OM API**, CSS values are now members of a new base class CSSStyleValue.

CSSPositionValue	CSSKeywordValue	CSSImageValue
CSSUnitValue	CSSMathValue	CSSTransformValue

.Extends-CSS .Typed-OM-API { }

CSSOM

```
el.style.opacity = 0.5;
window.getComputedStyle(el).opacity === "0.5" // Strings!
```

Typed OM

```
el.attributeStyleMap.set('opacity', 0.5);
el.computedStyleMap().get('opacity').value // 0.5
```

Get the radius from properties #1

Merged **una** merged 1 commit into `una:master` from `nucliweb:patch-1` 24 days ago

Conversation 3 | Commits 1 | Checks 0 | Files changed 1

nucliweb commented 24 days ago — Contributor

You need to get the value of the CSSUnitValue object `properties.get('--extra-scallopRadius').value`.
I think that this it's happening because you're doing a Math operation with this var in the next lines.

`Get the radius from properties` Verified aac8eb6

surma commented 24 days ago

You were faster than me, **@nucliweb** :D

This is correct. `properties.get()` gives you a `CSSUnitValue`, which has both `.value` and `.unit` so it can model stuff like `4px`, `8em`, `9%` etc.

@nucliweb

Get the radius from properties #1

Merged **una** merged 1 commit into `una:master` from `nucliweb:patch-1` 24 days ago

```
6   6
7   7       paint(ctx, size, properties) {
8   -         // why doesn't this work?
9   -         // const radius = properties.get('--extra-scallopRadius')
10  -         const radius = 10
    8   +     const radius = properties.get('--extra-scallopRadius').value
11  9         const scallopWeight = properties.get('--extra-scallopWeight')
12  10        const color = properties.get('--extra-scallopColor')
13  11        const height = size.height
@@ -69,4 +67,4 @@ if (typeof registerPaint !== 'undefined') {
69  67      }
```

This is correct. `properties.get()` gives you a `CSSUnitValue`, which has both `.value` and `.unit` so it can model stuff like `4px`, `8em`, `9%` etc.

Nested expressions

```
calc(1px - 2 * 3em)
```

```
new CSSMathSum(
  CSS.px(1),
  new CSSMathNegate(
    new CSSMathProduct(2, CSS.em(3))
  )
);
```

attributeStyleMap

```
el.attributeStyleMap.set('font-size', CSS.em(2));
el.attributeStyleMap.get('font-size');
// CSSUnitValue { value: 2, unit: 'em' }


el.attributeStyleMap.set('opacity', CSS.number(.5));
el.attributeStyleMap.get('opacity');
// CSSUnitValue { value: 0.5, unit: 'number' }
```

.Extends-CSS .Typed-OM-API { }

CSS

```
.foo {
  background-position: center bottom 10px;
  transform: translateX(1em) rotate(50deg) skewX(10deg);
  vertical-align: baseline;
  width: calc(100% - 3em);
}
```

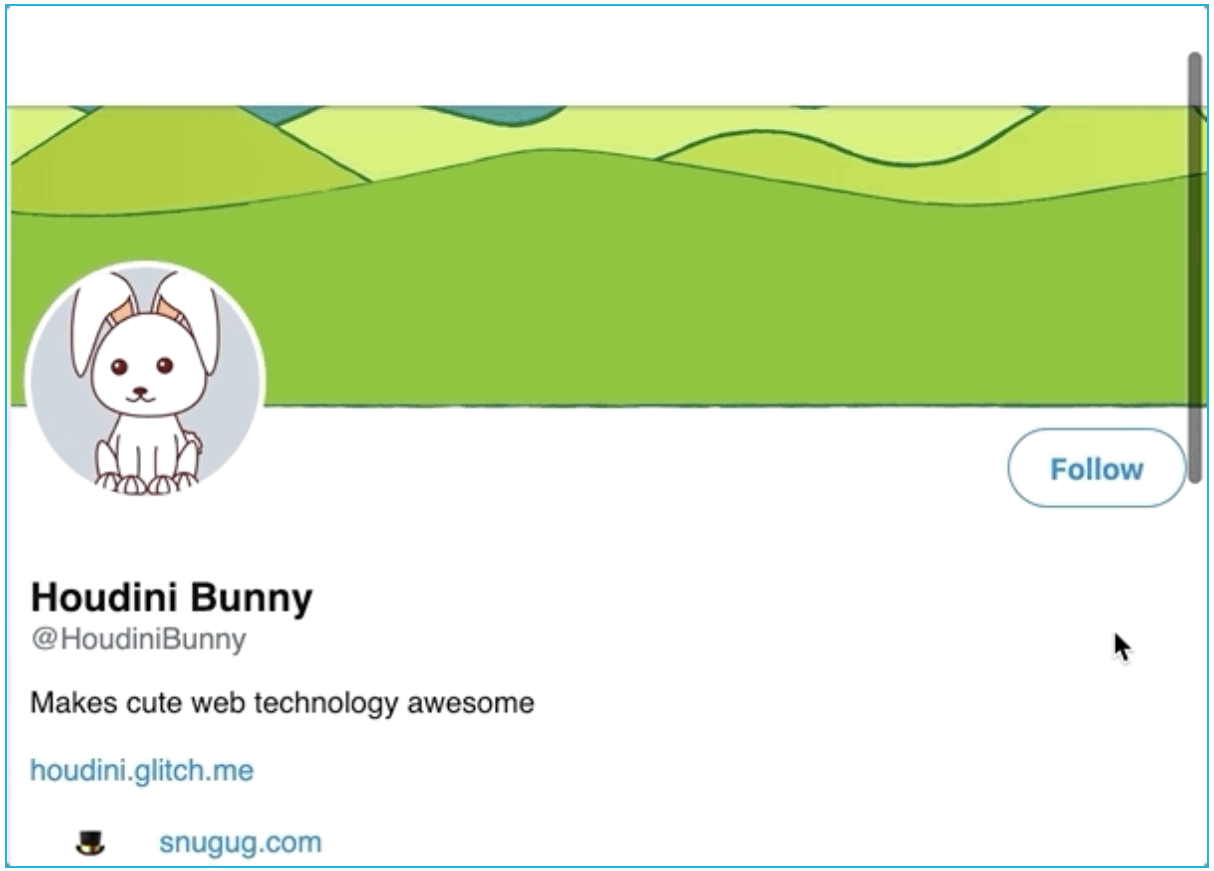
.Extends-CSS .Typed-OM-API { }

computedStyleMap

```
const cs = document.querySelector('.foo').computedStyleMap();

cs.get('vertical-align');
// CSSKeywordValue {
//  value: 'baseline',
// }

cs.get('background-position').x;
// CSSUnitValue {
//   value: 50,
//   unit: 'percent',
// }

cs.get('background-position').y;
```

Animation Worklet

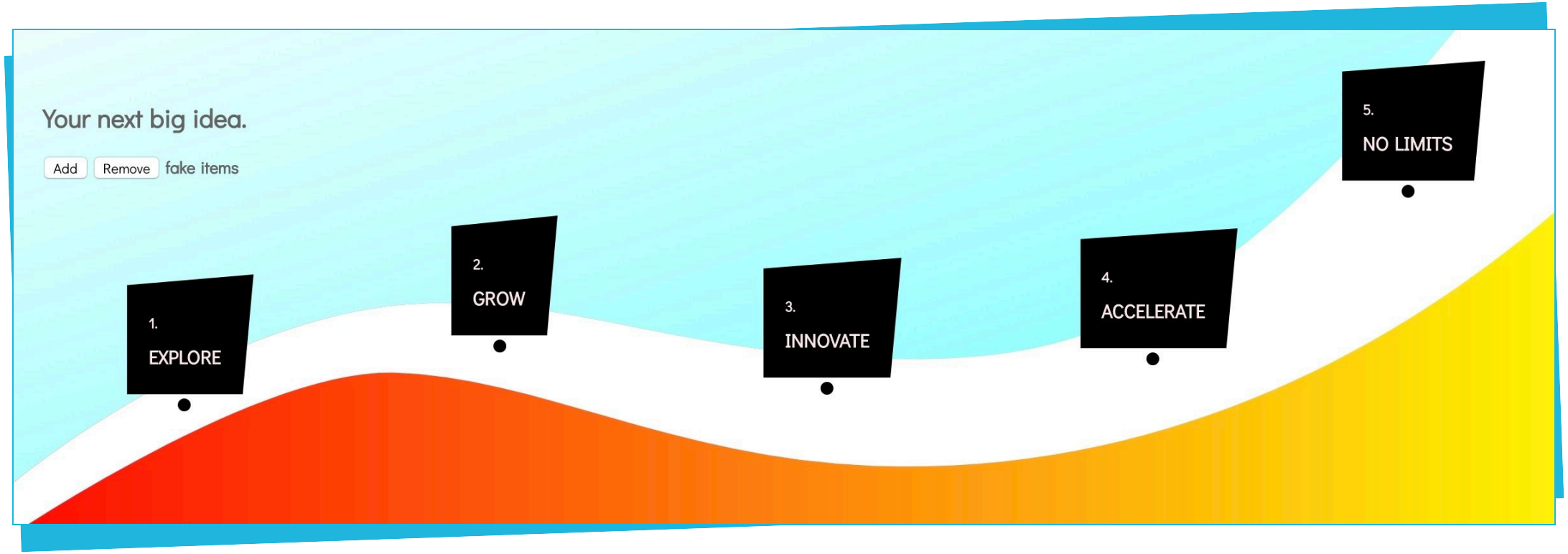
.Extends-CSS Animation-Worlet { }



[houdini.glitch.me/animation]()

Layout API

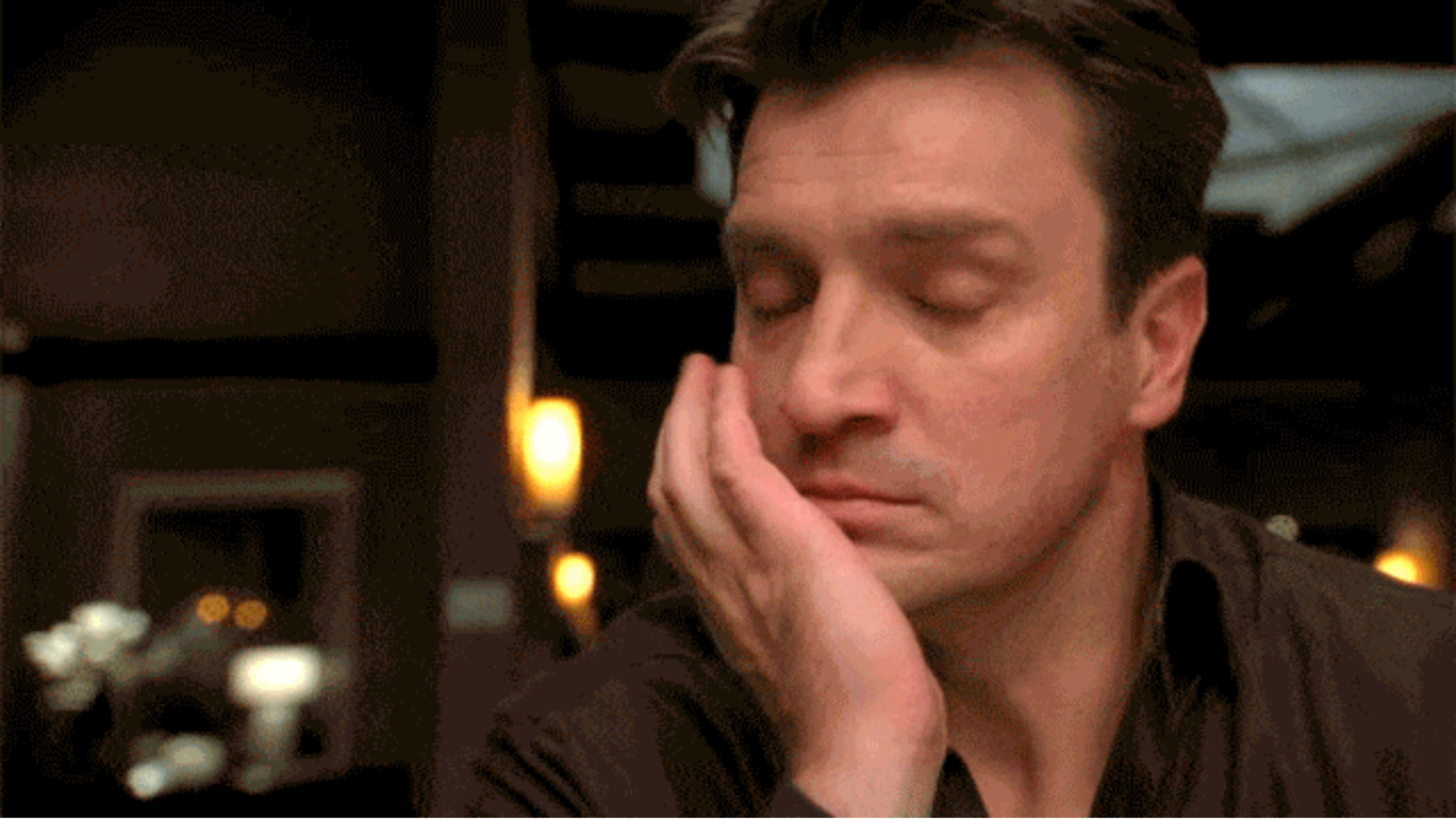
.Extends-CSS .Layout-API {}



Support
Christoph Keil

Is Houdini ready yet?

	Apple Safari	Google Chrome	Microsoft Edge	Mozilla Firefox	Opera	Samsung Internet	Spec
Layout API (Explainer)	No signal	Partial support (Canary) Details	No signal	Intent to implement Details	No signal	No signal	First Public Working Draft Spec
Paint API (Explainer Demos Article)	In Development Details	Shipped (Chrome 65) Details	No signal	Intent to implement (Servo) Details	Shipped (Opera 52) Details	Intent to implement	Candidate Recommendation Spec
Parser API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec
Properties & Values API (Demos)	Partial support (Safari TP 67) Details	Partial support (Canary) Details	No signal	In Development Details	No signal	No signal	Working Draft Spec
AnimationWorklet (Explainer Demos Article)	No signal	Partial support (Canary) Details	No signal	No signal	No signal	No signal	First Public Working Draft Spec
Typed OM (Explainer Article)	Intent to implement Details	Shipped (Chrome 66) Details	No signal	Intent to implement Details	Shipped (Opera 53) Details	Intent to implement	Working Draft Spec
Font Metrics API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec



Conclusions

Ruvim Noga

.Conclusions { }

Polyfills

This is not for me

Free will

Houdini is **NOT** like a Babel for CSS

.Conclusions { }

Hi Stoyan.

PostCSS + PostCSS Preset Env s like Babel for CSS, because is a transpiler to add support all the new features from Ecma TC39 in JS and CSS-TAG Houdini Task Force in CSS.

I think that CSS Houdini is more than transpiler, we have a new APIs to work **inside** the CSS engine.

CSS Houdini Postcss

.Conclusions { }

Hi Stoyan.

PostCSS + PostCSS Preset Env s like Babel for CSS, because is a transpiler to add support all the new features from Ecma TC39 in JS and CSS-TAG Houdini Task Force in CSS.

I think that CSS Houdini is more than transpiler, we have a new APIs to work **inside** the CSS engine.

CSS Houdini Postcss

I see another talk...

CSS
is
Awesome

Resources

Ricardo Gomez Angel

.Resources { }

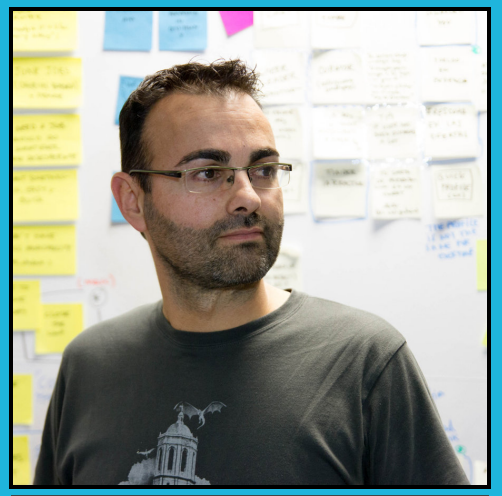
How Browsers Work: Behind the scenes of modern web browsers

.Awesome-CSS-Houdini { }

⭐ are welcome 😊

Houdini samples

.Hello {



@nucliweb

Joan León

Developer Advocate 🥑
at @SchibstedSpain

Co-Founder & UI Engineers
at @sublime_codes

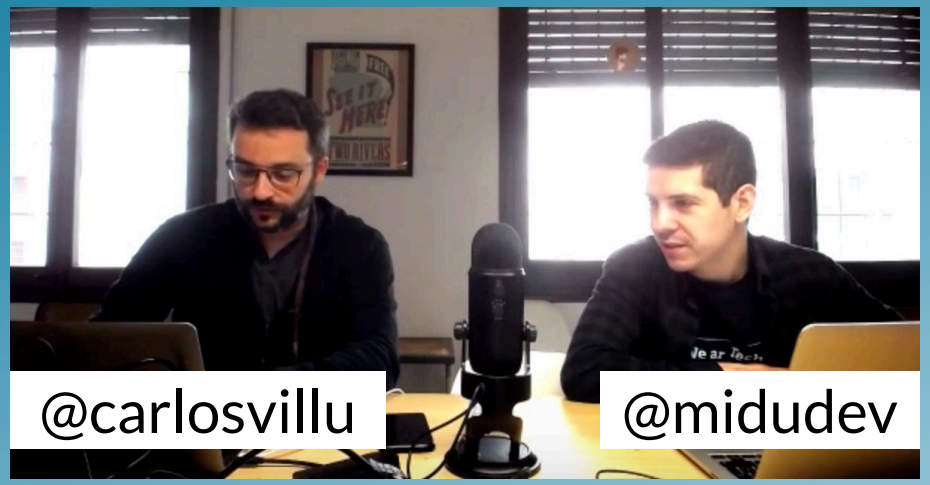
#CSS
#Animation
#PostCSS
#SVG
#Javascript

}

We are a **big family** of digital brands

SUI Components

@suiengineers



SCHIBSTED
SPAIN

```
.Talk::after {

    content: "Any question?";

}
```

.Thanks {

@nucliweb

```
*::after {
  content: " CSS is Awesome 🤘 ";
}
```

}