Houdini

...and the future of the web

Who Are These Jokers?

Ian Kilpatrick: TL of Houdini in Chrome. Member of CSSWG and Houdini TF. Primary editor of Paint API, Layout API, Worklets.

Eddy Mead: TL of Style team in Chrome. Lead implementor of Typed CSSOM.

Shane Stephens: Sydney Chrome uberTL. Member of CSSWG and co-chair of Houdini TF. Primary editor of Typed OM, Properties and Values.

The Extensible Web Manifesto

“To enable libraries to do more, browser vendors should provide new low-level capabilities that expose the possibilities of the underlying platform as closely as possible.”

Houdini

The extensible web, applied to CSS.

- Build new CSS primitives without waiting for a specification committee
- Achieve better interop, faster
- “Scratch that itch” - adjust features to work the way you want them to

Houdini is...

Extensible CSS

Custom Elements, but for style

Access to the layout & paint stack

As you listen to this talk...

...keep in mind that each of the APIs we talk about both

solve an immediate problem,

and

contribute towards the greater goal of extensible CSS.

Houdini Specifications

- Specification group is the Houdini Task Force
  - CSSWG + TAG
- New specifications being incubated through WICG
- Most specs at https://drafts.css-houdini.org/

Houdini Specifications

We’re on github!

- https://github.com/w3c/css-houdini-drafts
- https://github.com/WICG/CSS-Parser-API
- https://github.com/WICG/ResizeObserver
- https://github.com/WICG/IntersectionObserver

Houdini Specifications

- Github used for
  - Specification drafting
  - Issue tracking
  - *(coming soon?)* Polyfills
- **Non-member contributions are welcome**
  - Use cases
  - Feature discussion
  - Specification edits

Specification Roll Call

We have **4 specifications** that are ready for experimental implementation:

- **Typed OM:** *high performance object access to CSSOM*
- **Properties & Values:** *typed custom properties*
- **Worklets:** *synchronous, isolated JavaScript execution*
- **Paint API:** *DIY rendering*

Specification Roll Call

We have **3 specification areas** with a degree of forward progress:

- **Layout API:** *DIY layout*
- **Font Metrics:** *Programmatic access to font information*
- **Paint API L2:** *DIY rendering. But better!*

Specification Progress

We have **2 areas of investigation** underway:

- **Scrolling:** *currently includes compositor worker, scroll customization, declarative scroll-linked animations, and more*
- **Parsing:** *extend or replace the CSS syntax*

Implementation status

In **Blink** we have partial experimental implementations of **Typed OM**, **Worklets** and **Paint API**. We intend to start working on **Properties and Values** soon.

We also have an experimental implementation of **Compositor Worklet**.

Firefox has begun implementation of **Properties and Values**.

Typed OM

Because *anything* is better than Strings

The problem

```
someElement.style.boxShadow = '0px 0px 8px 4px ' + 'rgb(' + r +
',' + g + ',' + b + ')';
```

```
someElement.style.transform = 'scale(' + scaleValue + ') rotate('
+ rotationValue + 'deg) translate(' + translateValue + 'px)';
```

Don't build strings

```
someElement.styleMap.set(
       'transform',
       new CSSTransformValue([
         new CSSScale(2, 2),
         new CSSRotation(90),
         new CSSTranslation(new CSSSimpleLength(3, "px"))
    ]);
```

The problem

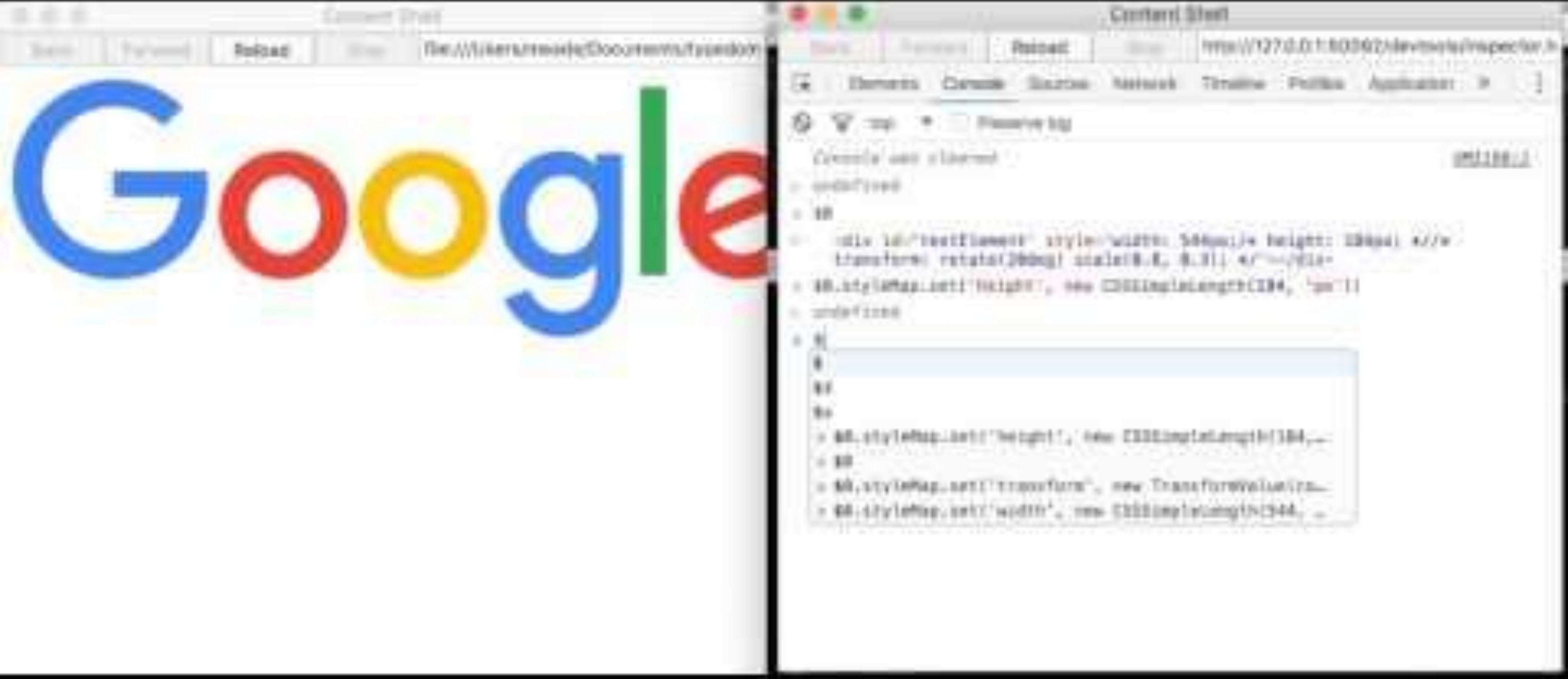
```
var transform = someElement.style.transform;
// 'scale(2) rotate(90deg)';


// Need to update the rotation value somehow...
var newTransform = someLibrary.parseTransform???
```

Don't parse strings

```
var v = someElement.styleMap.get('transform')
// CSSTransformValue {cssString: "scale(2) rotate(90deg)"}

var components = [...v.values()];
components[1] = new CSSRotation(components[1].angle + 20);
someElement.styleMap.set('transform',
       new CSSTransformValue(components));
```

Demo

https://github.com/wilddamon/typedom-demos



```
Console was cleared                                          VM118:1
< undefined
> $0
<   <div id="testElement" style="width: 500px;/* height: 184px; */ /*
    transform: rotate(20deg) scale(0.8, 0.3); */">…</div>
> $0.styleMap.set('height', new CSSSimpleLength(184, 'px'))
< undefined
> $
  $
  $$
  $x
  > $0.styleMap.set('height', new CSSSimpleLength(184,…
  > $0
  > $0.styleMap.set('transform', new TransformValue([ro…
  > $0.styleMap.set('width', new CSSSimpleLength(344, …
```

Layout API

Let authors write next gen-layouts:

- Constraint-based layouts
- MathML
- KnuthPlass
- etc..

Layout API

https://github.com/w3c/css-houdini-drafts/tree/master/css-layout-api

Fragment

BreakToken

????

ConstraintSpace

Exclusions

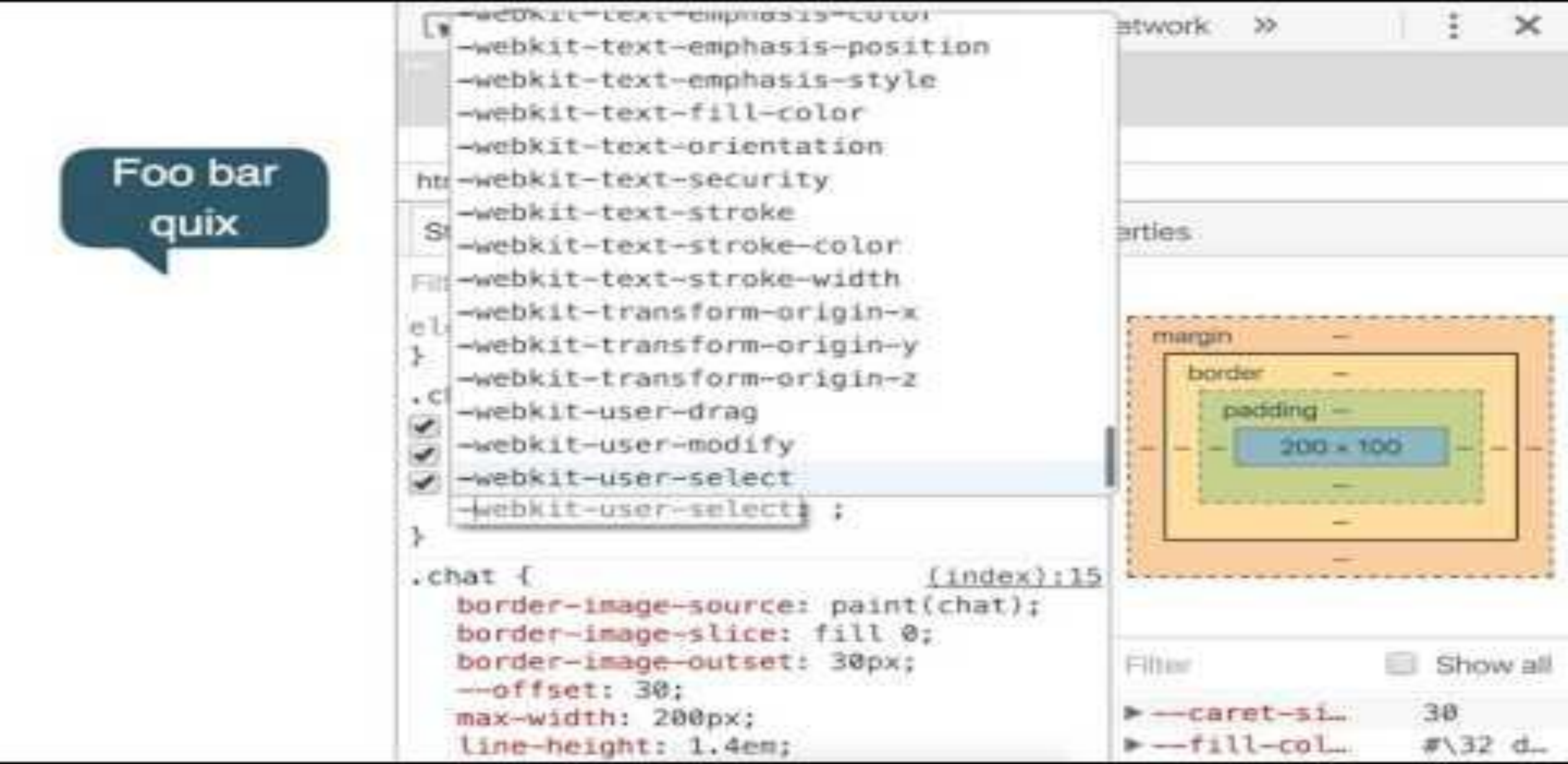
????

Paint API

Layout-sized canvases anywhere you can use an image.

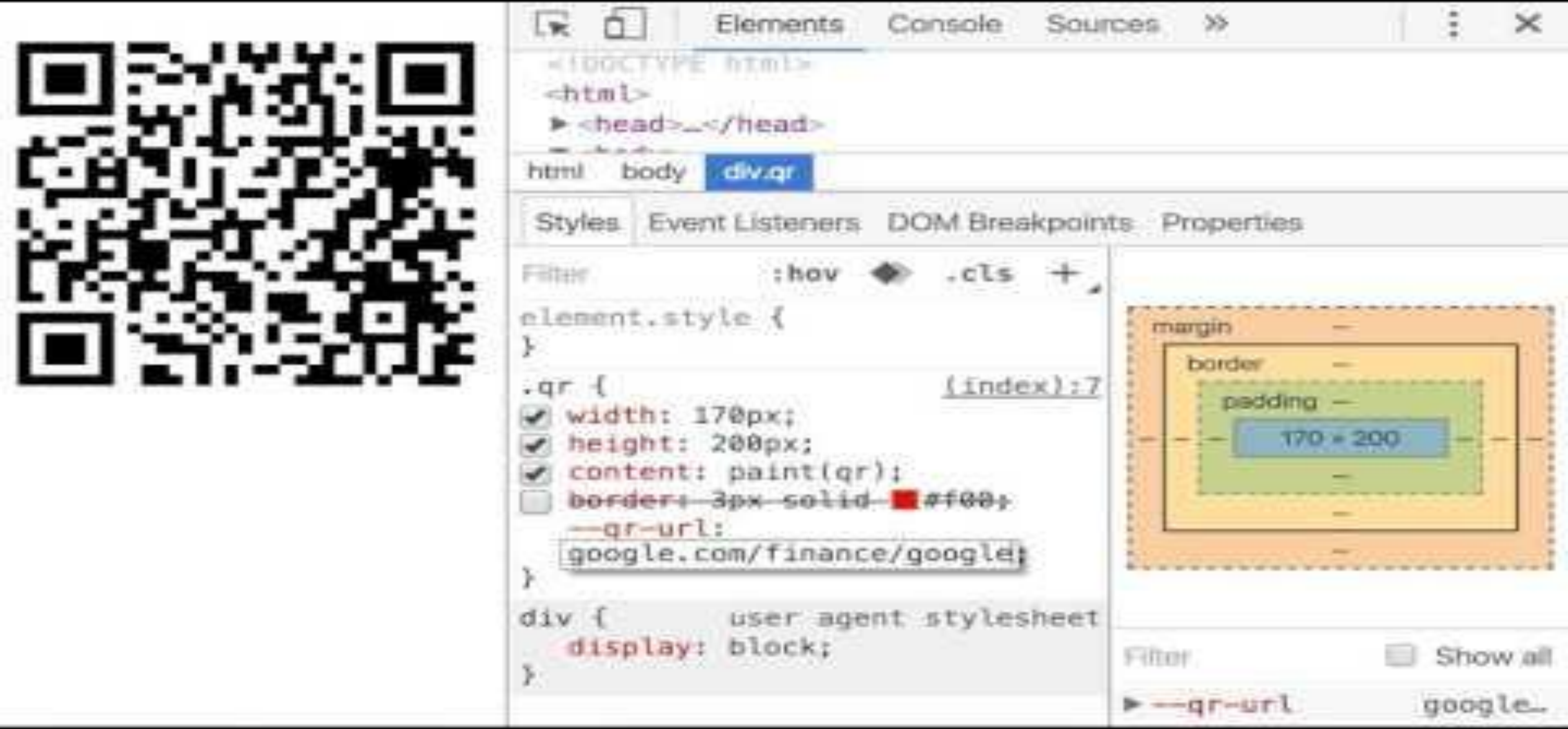
Paint API (v1)

```
registerPaint('circle', class {
    static get inputProperties() { return ['--circle-color']; }
    paint(ctx, geom, properties) {
        ctx.fillStyle = properties.get('--circle-color');

        const x = geom.width / 2;
        const y = geom.height / 2;
        const radius = Math.min(x, y);

        ctx.beginPath();
        ctx.arc(x, y, radius, 0, 2 * Math.PI, false);
        ctx.fill();
    }
});
```



```
-webkit-text-emphasis-color
-webkit-text-emphasis-position
-webkit-text-emphasis-style
-webkit-text-fill-color
-webkit-text-orientation
-webkit-text-security
-webkit-text-stroke
-webkit-text-stroke-color
-webkit-text-stroke-width
-webkit-transform-origin-x
-webkit-transform-origin-y
-webkit-transform-origin-z
-webkit-user-drag
-webkit-user-modify
-webkit-user-select
-webkit-user-select ;
}
.chat {                                  (index):15
  border-image-source: paint(chat);
  border-image-slice: fill 0;
  border-image-outset: 30px;
  --offset: 30;
  max-width: 200px;
  line-height: 1.4em;
```

```
▶ --caret-si…     30
▶ --fill-col…     #\32 d…
```



```
<!DOCTYPE html>
<html>
  ▶ <head>…</head>
```

```
element.style {
}
.qr {                                      (index):7
☑ width: 170px;
☑ height: 200px;
☑ content: paint(qr);
□ ~~border: 3px solid 🟥#f00;~~
  --qr-url:
  google.com/finance/google;
}
div {              user agent stylesheet
  display: block;
}
```

```
▶ --qr-url                        google…
```

Paint API (soon™)

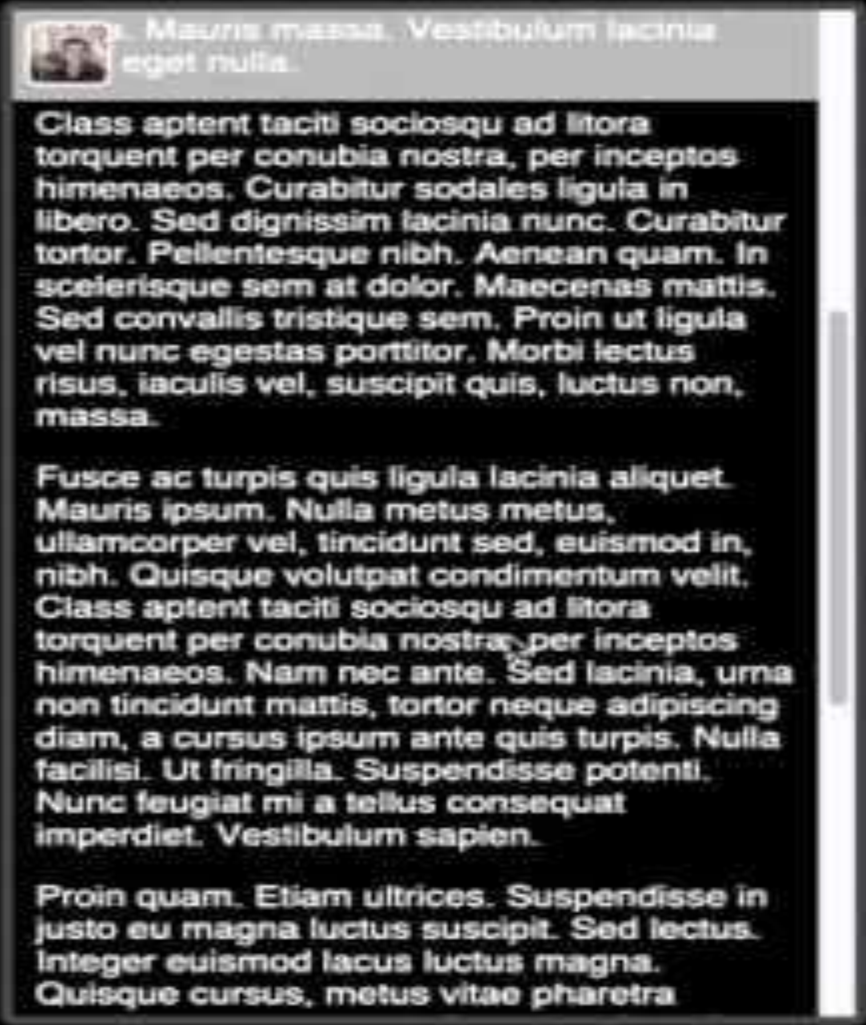
- Image/font/filter support, e.g.

```
.class {
  --image: url(img.png);
  --filter: blur(3px);
  --font-face: “my-font”
}
```

```
paint(ctx, geom, map) {
  ctx.drawImage(map.get(‘--image’), ...);
  ctx.filter = map.get(‘--filter’);
  ctx.font = map.get(‘--font-face’);
}
```

Compositor Worklet

Almost code complete on trunk!!!



. Mauris massa. Vestibulum lacinia eget nulla.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa.

Fusce ac turpis quis ligula lacinia aliquet. Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam nec ante. Sed lacinia, urna non tincidunt mattis, tortor neque adipiscing diam, a cursus ipsum ante quis turpis. Nulla facilisi. Ut fringilla. Suspendisse potenti. Nunc feugiat mi a tellus consequat imperdiet. Vestibulum sapien.

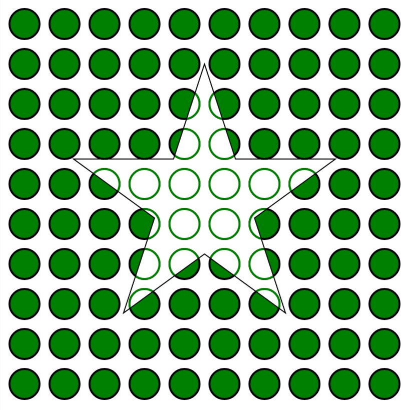
Proin quam. Etiam ultrices. Suspendisse in justo eu magna luctus suscipit. Sed lectus. Integer euismod lacus luctus magna. Quisque cursus, metus vitae pharetra

Covert

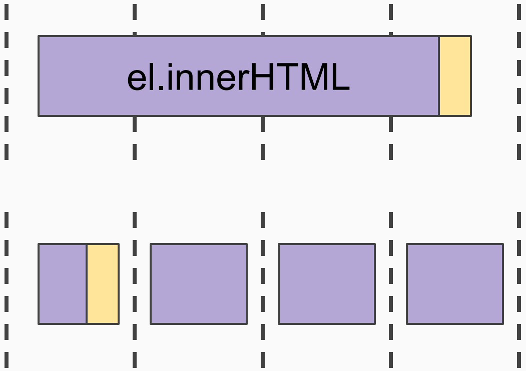
3-day experiment to replicate (parts of) SVG at user-level

Breakout session planned.

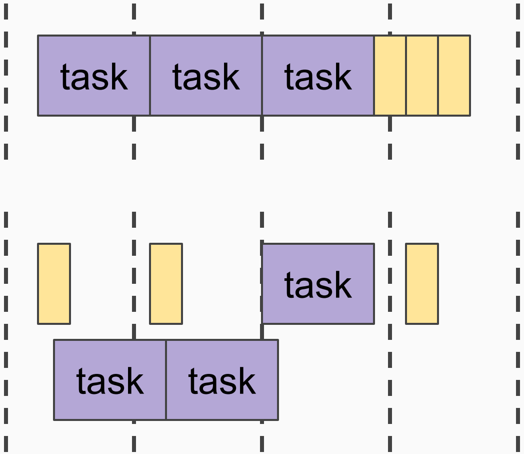
```
<cvt-texture name=foreground>
  <cvt-repeat name=x value="20,60..380">
    <cvt-repeat name=y value="20,60..380">
      <cvt-circle class=fg-circle center-left="{x}"
                  center-top="{y}"></cvt-circle>
    </cvt-repeat>
  </cvt-repeat>
</cvt-texture>
<cvt-shape class='star' path="..."
     stroke="black" line-width="1" texture=foreground></cvt-shape>
```



... and one more thing



asyncAppend



task worklets

… and one more thing

- Async-Append
- Task-Worklets
- ResizeObserver
- DOM-Read/Write-Batching
- ???

Treating the main thread with respect

Fin.