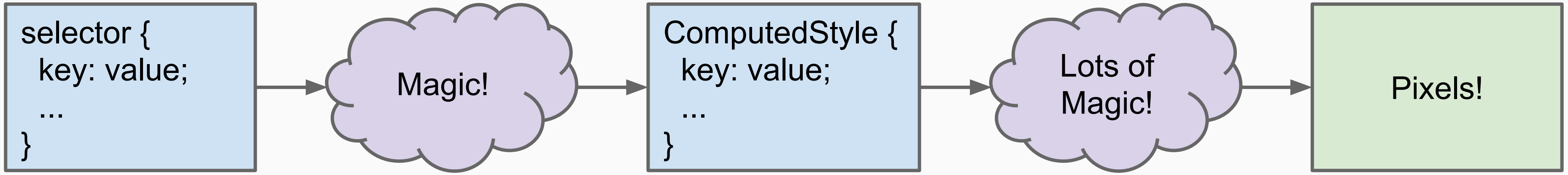
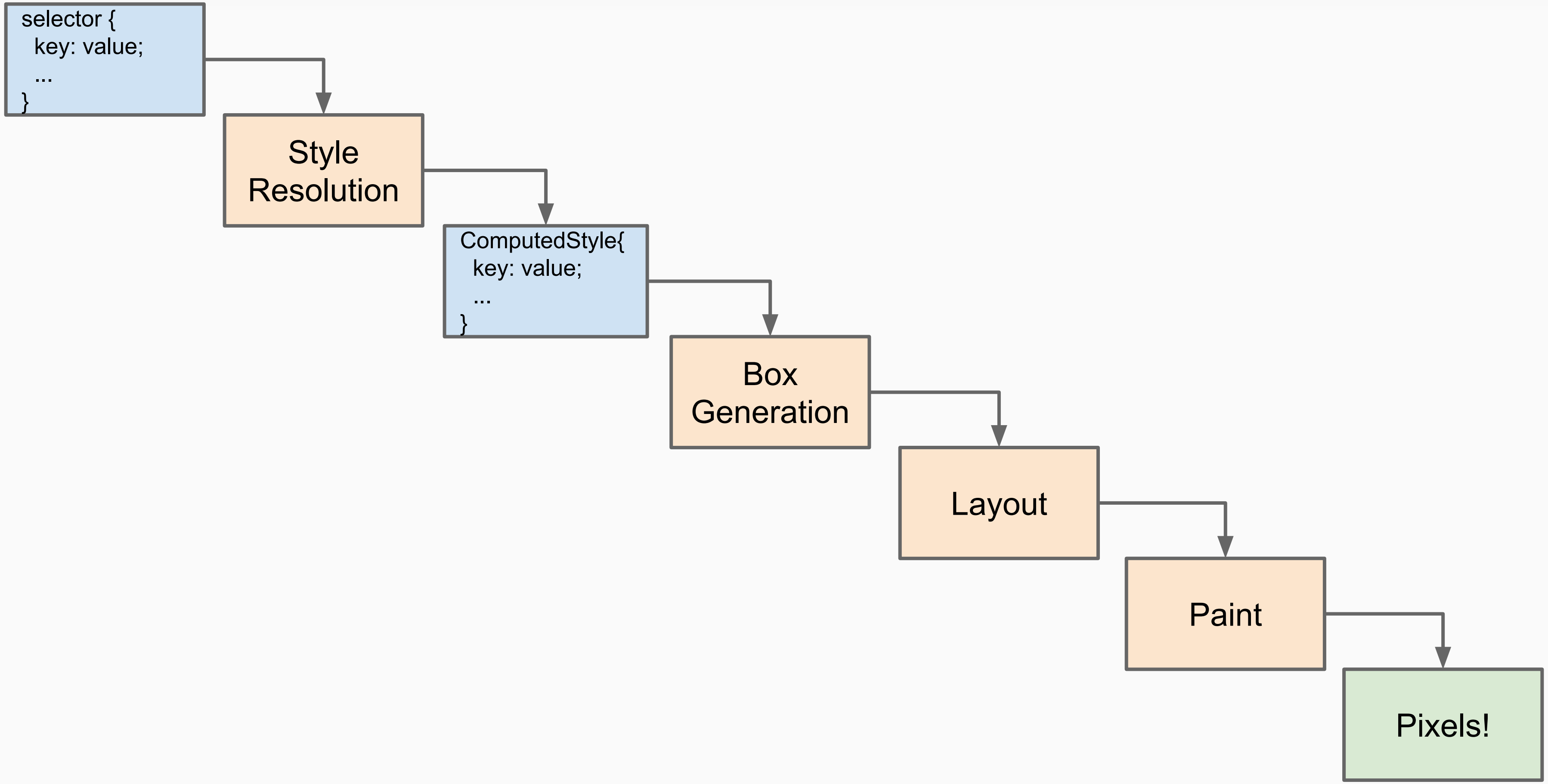
HoudiniTF

ikilpatrick@

“Explaining the Rendering Engine”



“Explaining the Rendering Engine”



Motivation

Web Authors

"It takes years for requested features to be implemented in browsers"

=> Give web authors the same power as browser vendors for adding functionality to CSS

Spec Authors

"There are too many new ideas and not enough specification authors to capture them"

=> make it possible for web developers to "directly specify" features

Browser Vendors

"The CSSWG is drowning us in new features"

=> Provide a layered API for adding features on top of a tiny core

Motivation

Users

- New ideas on the web inevitably cost users in performance
  - Material Design ripples - div creation/removal for every ripple
  - flexbox polyfills - expensive absolute positioning (300ms layouts - on desktop!)

- Create “performance foothills” rather than a “performance cliff”

Style

Custom Properties

https://drafts.css-houdini.org/css-properties-values-api/

```
.className {
  --my-scale: 1;
  --my-scale: ‘foo’; // No type checks!
  transform: scale(var(--my-scale));
}


document.registerProperty({
  name: ‘--my-scale’,
  syntax: ‘<number>’, // Validate types!
  inherits: false, // Choose inheritance!
  initialValue: ‘1’
};
```

Style

Typed CSS OM

https://drafts.css-houdini.org/css-typed-om/

```
// Actual code.
style.transform =
  'translate3d('+dx+'px,'+dy+'px,0)';
-------------------------------------


function checkHighContrastMode() {
  var c1 = 'rgb(30, 40, 50)';
  div.style.color = c1;
  var c2 = getComputedStyle(div).color;
  if (isValidColor(c1)&&isValidColor(c2)) {
    var parsedColor1 = libParse(c1);
    var parsedColor2 = libParse(c2);
    return parsedColor1.hex ==
      parsedColor2.hex;
  }
  return false;
}
```

Style

Typed CSS OM

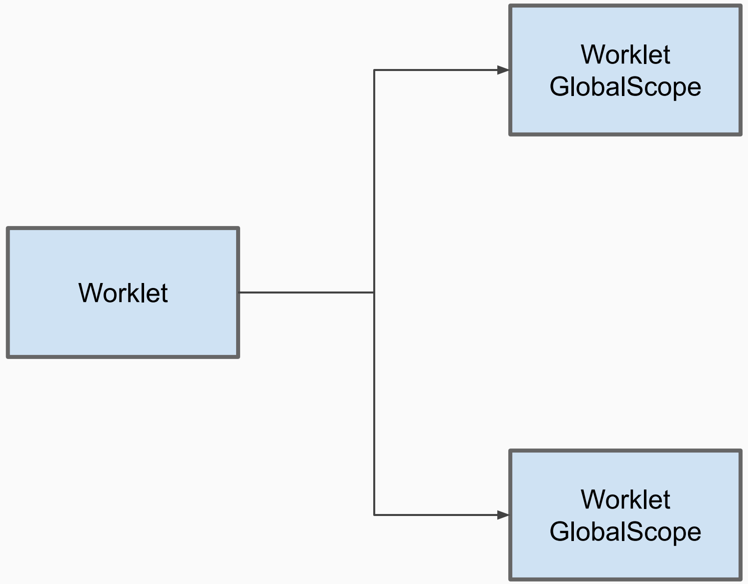
https://drafts.css-houdini.org/css-typed-om/

```
// New code.
function checkHighContrastMode() {
  var c1 = new ColorValue(30, 40, 50);
  div.styleMap.set(‘color’, c1);
  var c2 =
    getComputedStyleMap(div).get(‘color’);
  return c1.rgb == c2.rgb; // \o/
}
```

Infrastructure

Worklets

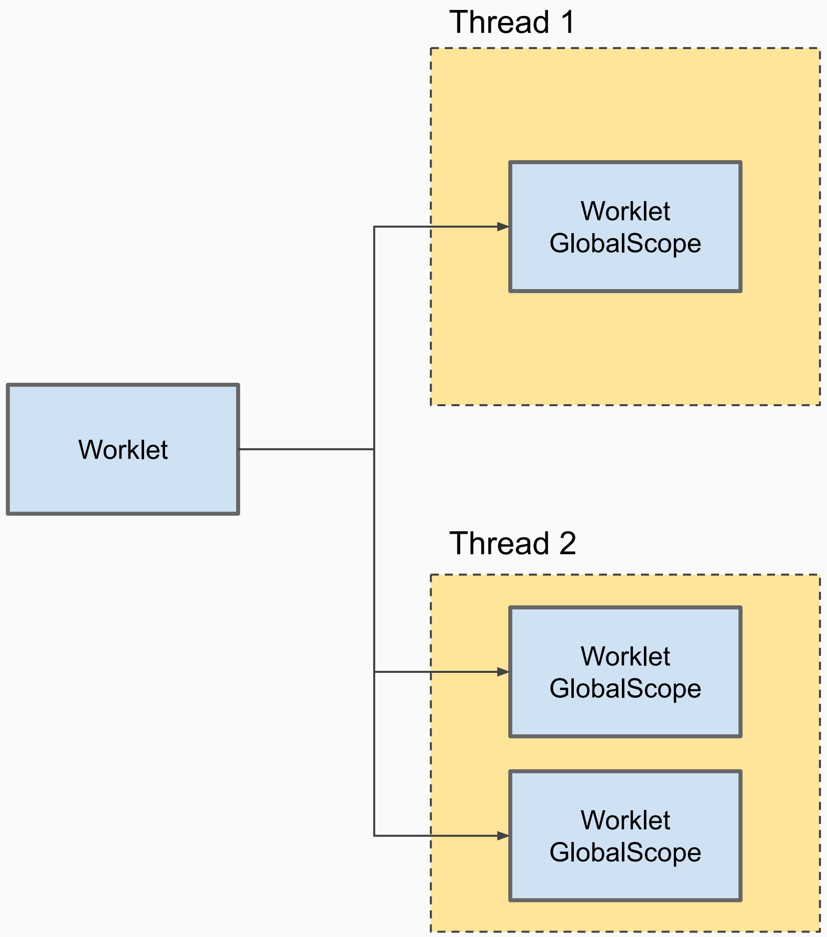
https://drafts.css-houdini.org/worklets/



Infrastructure

Worklets

https://drafts.css-houdini.org/worklets/



Style
Apply Hook

https://drafts.css-houdini.org/css-properties-values-api/

```
// Main Javascript.
window.styleWorklet.importScripts(
  'flexbox-style.js');

// LayoutGlobalScope Javascript.
registerApplyHook('my-flexbox', class {
  apply() { … }
}, {
  inputProperties: [ … ];
  outputProperties: [ … ];
});
```

Layout

CSS Layout API

https://drafts.css-houdini.org/css-layout-api/

```
// Main Javascript.
window.layoutWorklet.importScripts(
  ‘flexbox-layout.js’);

// LayoutGlobalScope Javascript.
registerLayout(‘my-flexbox’, class {
  minContent() { }
  maxContent() { }
  layout() { }
}, {
  inputProperties: [‘--flex-order’];
});
```

Line Layout

CSS Line Layout API

(*no specification yet*)

```
// Main Javascript.
window.layoutWorklet.importScripts(
  ‘flexbox-layout.js’);

// LayoutGlobalScope Javascript.
registerLineLayout(‘ruby-ruby’, class {
  minMaxContent() { }
  layout() {
    measureText(...);
    renderLineBox(...);
  }
}, {
  inputProperties: [‘--ruby-type’];
});
```

Paint

CSS Paint API

https://drafts.css-houdini.org/css-paint-api/

```
// Main Javascript.
window.paintWorker.importScripts(
  ‘new-borders.js’);

// PaintGlobalScope Javascript.
registerPaint(‘fancy-border’, class {
  paint(ctx, geom, styleMap) { }
  overflow() { }
}, {
  inputProperties: [‘--fancy-border’];
});
---------------
.className {
  border-image: paint(‘fancy-border’);
}
```

Compositing

Async Style

https://drafts.css-houdini.org/css-async-style/

```
// Main Javascript.
worker = new
CompositorWorker("scroll-animation.js");
scroller = new CompositorProxy(scrollElement,
    ['scrollTop']);
worker.postMessage([scroller]);

// CompositorWorker Javascript.
self.onMessage = function(e) {
  scroller = e.data[0];
  requestAnimationFrame(tick);
}

tick = function(ts) {
  scroller.scrollTop = OMGButterfliesFunction(ts);
  requestAnimationFrame(tick);
}
```

DEMOS!