

Making a Silky Smooth Web.

Paul Lewis // SmashingConf 2015

A close-up portrait of a man with light brown hair and a well-groomed, reddish-brown beard and mustache. He is wearing black-rimmed glasses and a dark blue beanie with three horizontal stripes. He is looking directly at the camera with a neutral expression. The background is a plain, light-colored wall.

+aerotwist
@aerotwist



Web Perf Today.





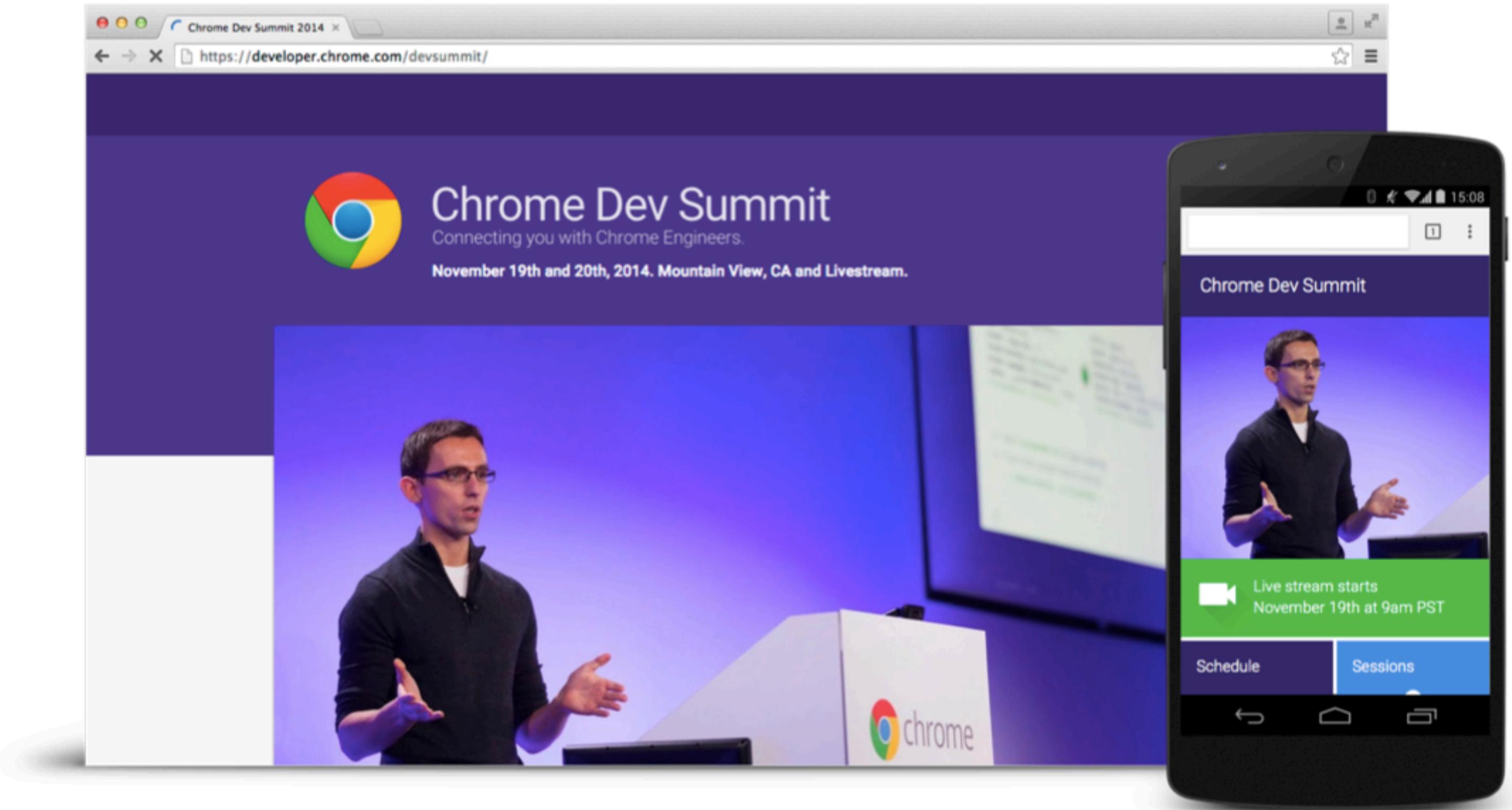
WOW!

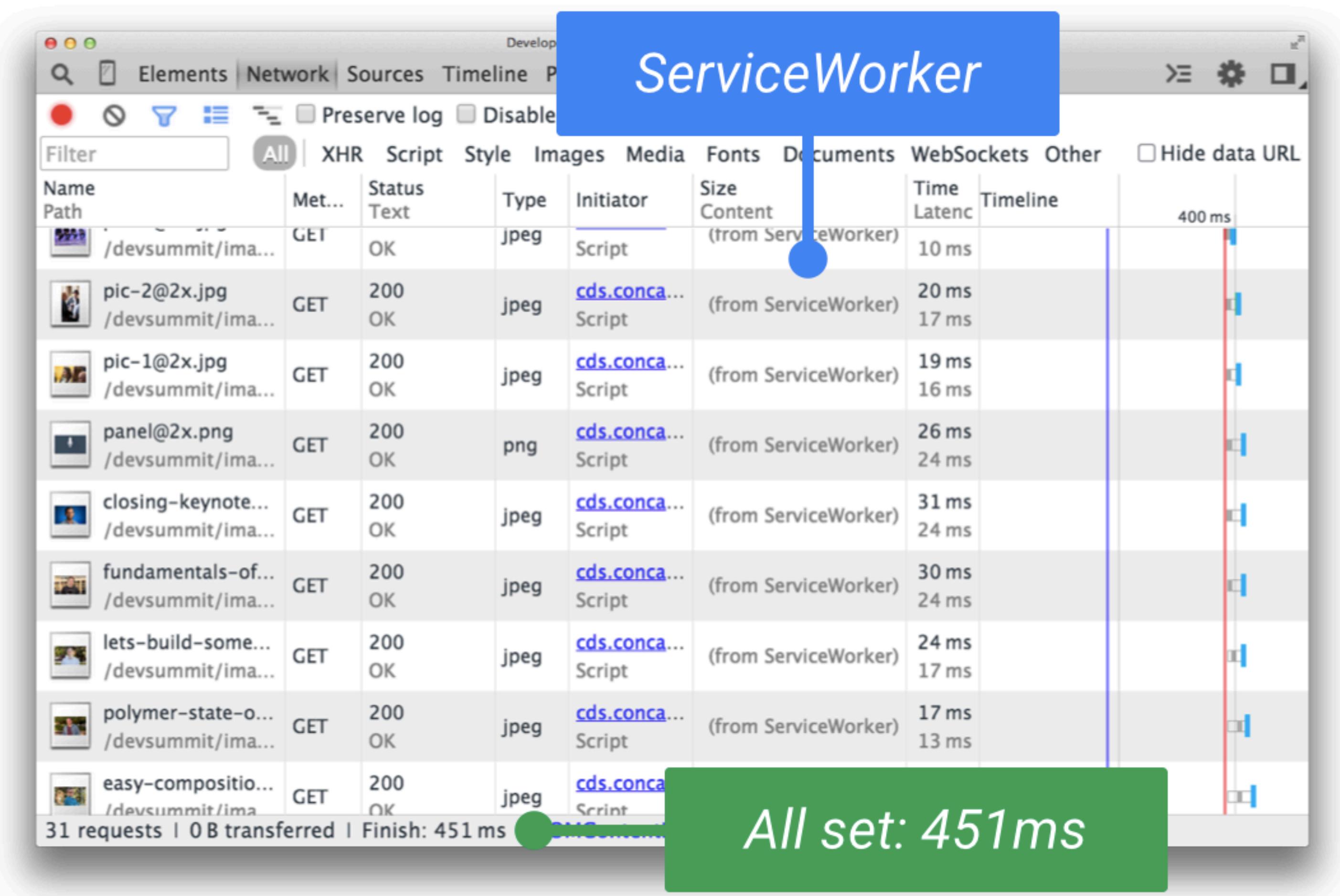
ONLY

5MB

(GZIPPED)







Chrome Browser - Google X

https://play.google.com/store/apps/details?id=com.android.chrome

Google play

Search

Sign In

Apps

Categories ▾ Home Top Charts New Releases

My apps

Shop

Games

Editors' Choice

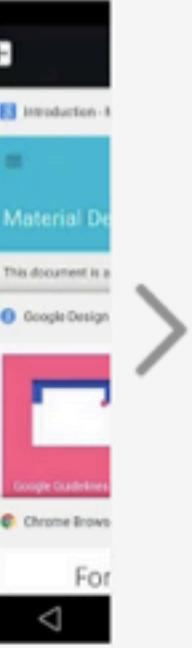
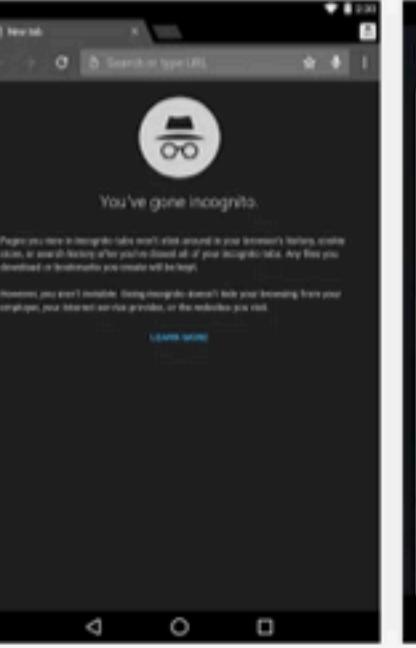
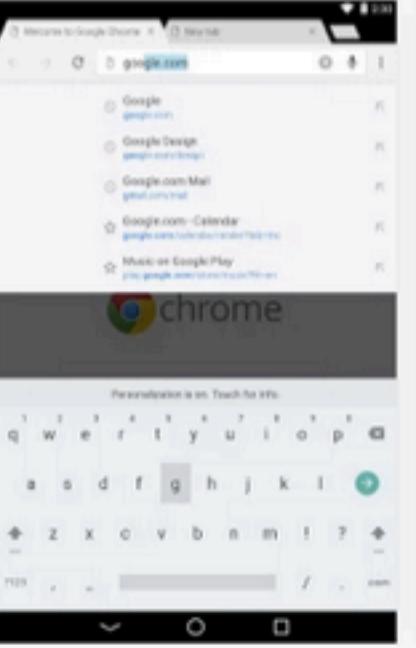
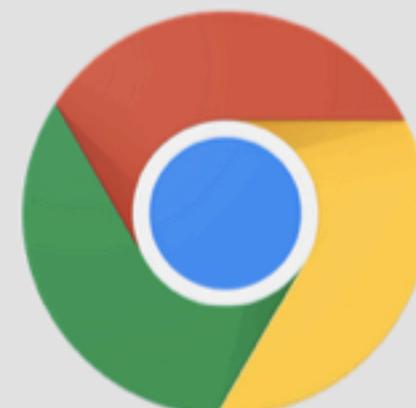
Chrome Browser - Google

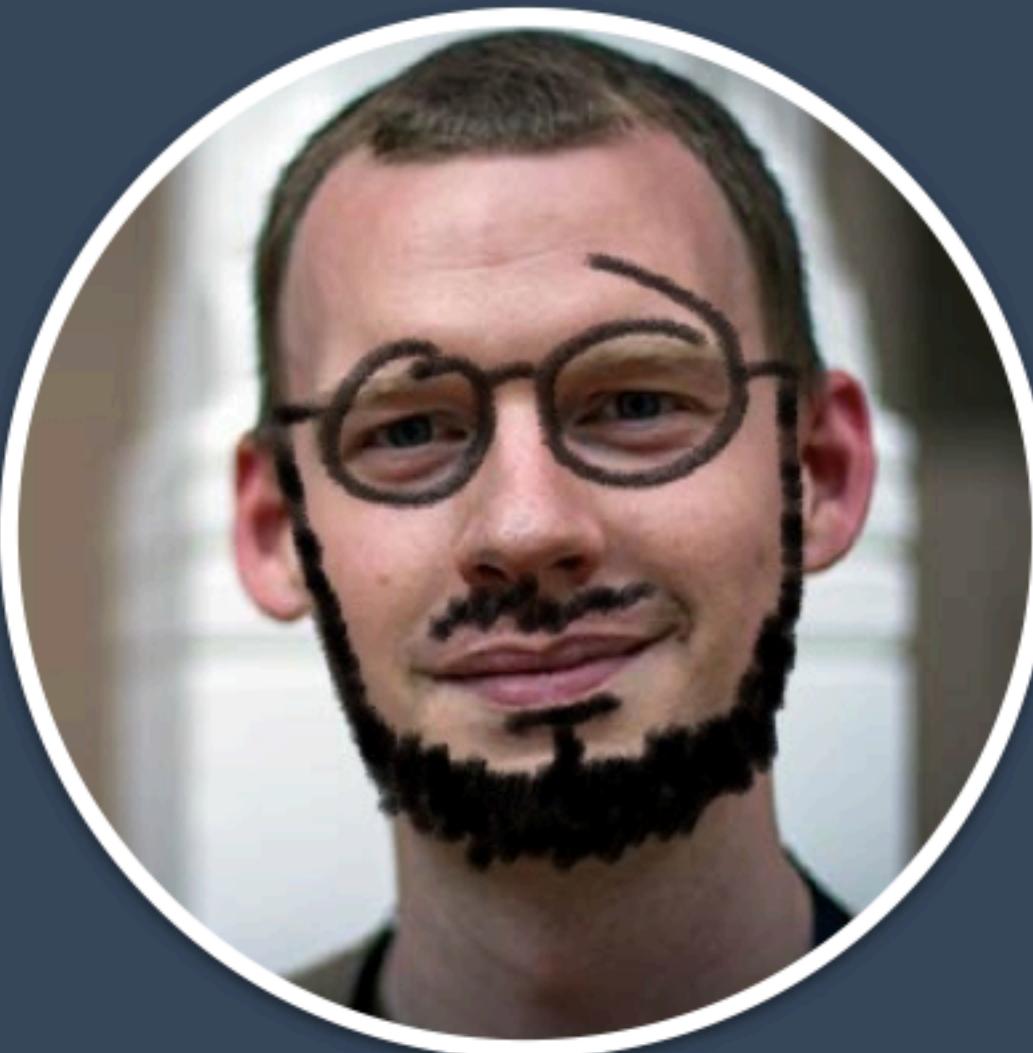
Google Inc. - March 11, 2015
Communication

Install Add to Wishlist

★★★★★ (2,789,016) g+ +2523581 Recommend this on Google

Top Developer

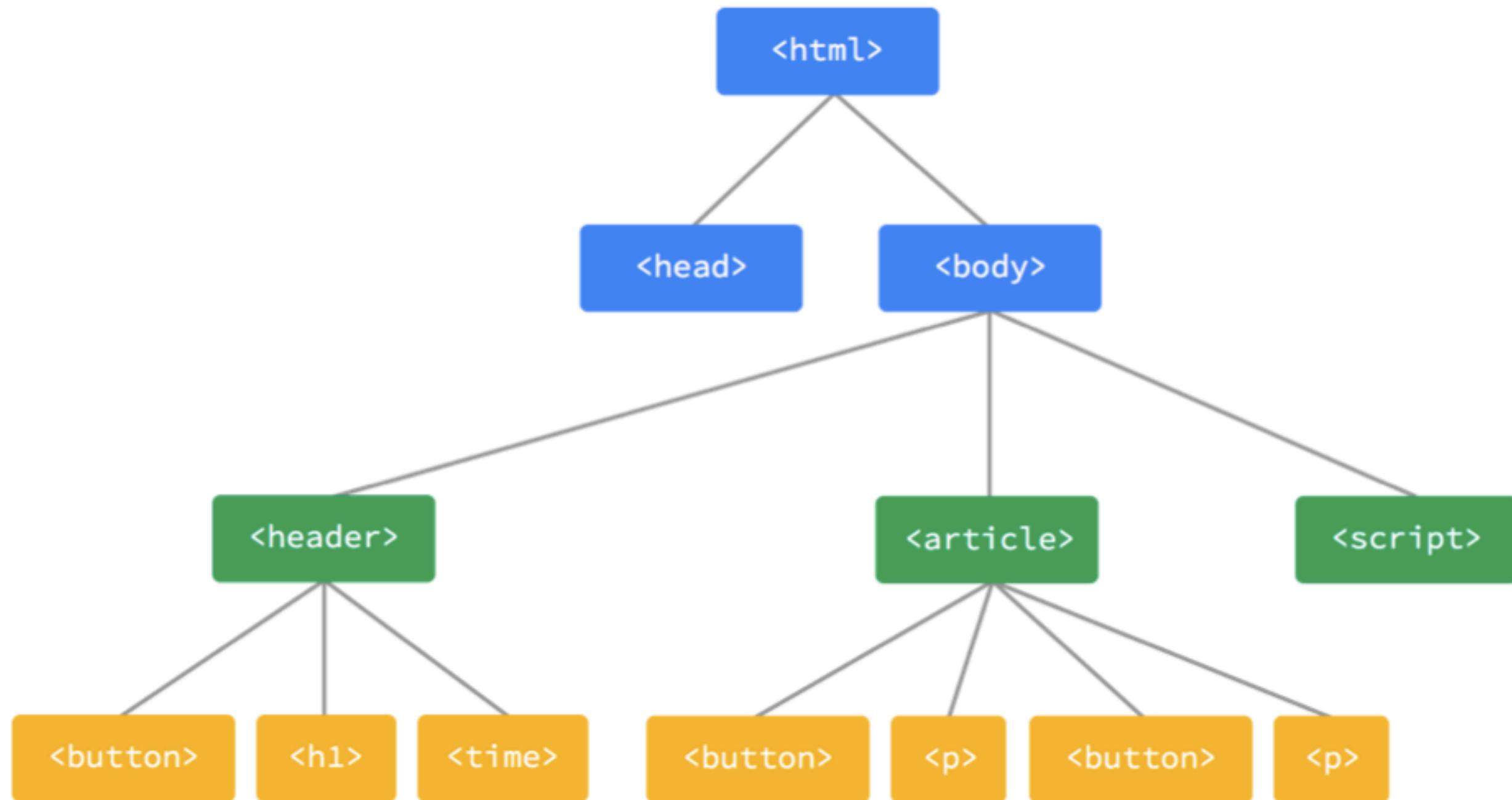




Modern Progressive Enhancement

16:40 today with Jack Archibungle

Making frames.



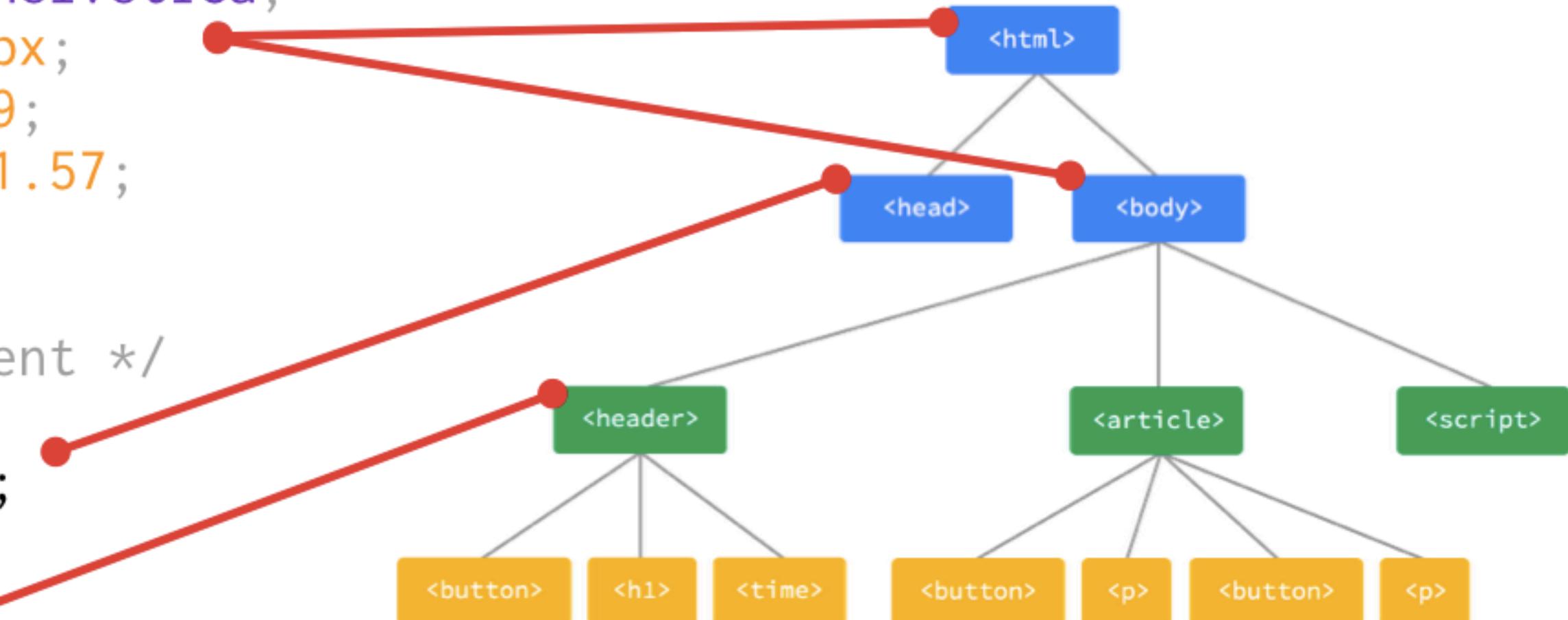
Recalculate Style.

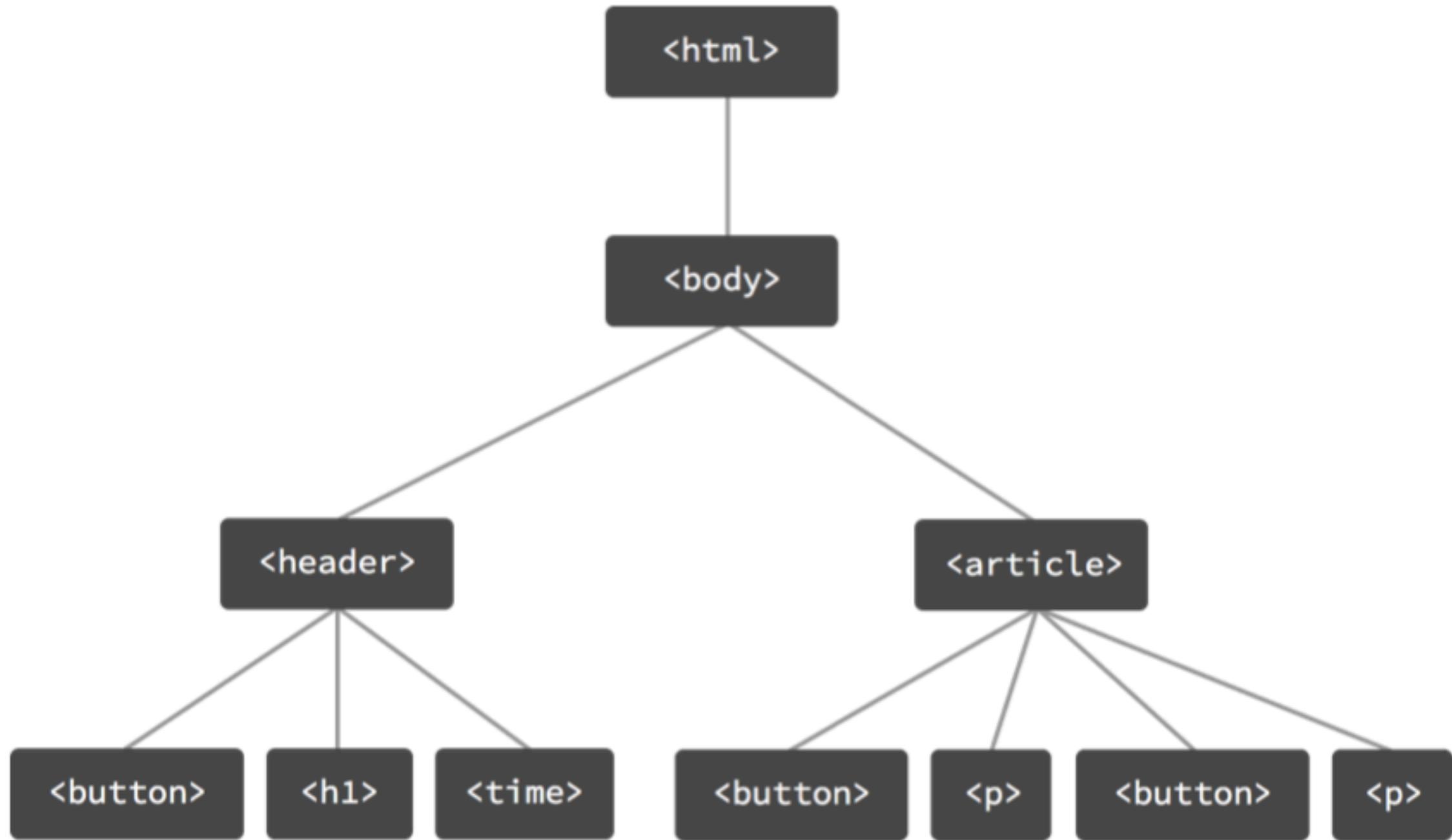
```
html, body {  
    margin: 0;  
    padding: 0;  
    font-family: Helvetica;  
    font-size: 14px;  
    color: #595959;  
    line-height: 1.57;  
}  
  
/* From user agent */
```

```
head {  
    display: none;  
}
```

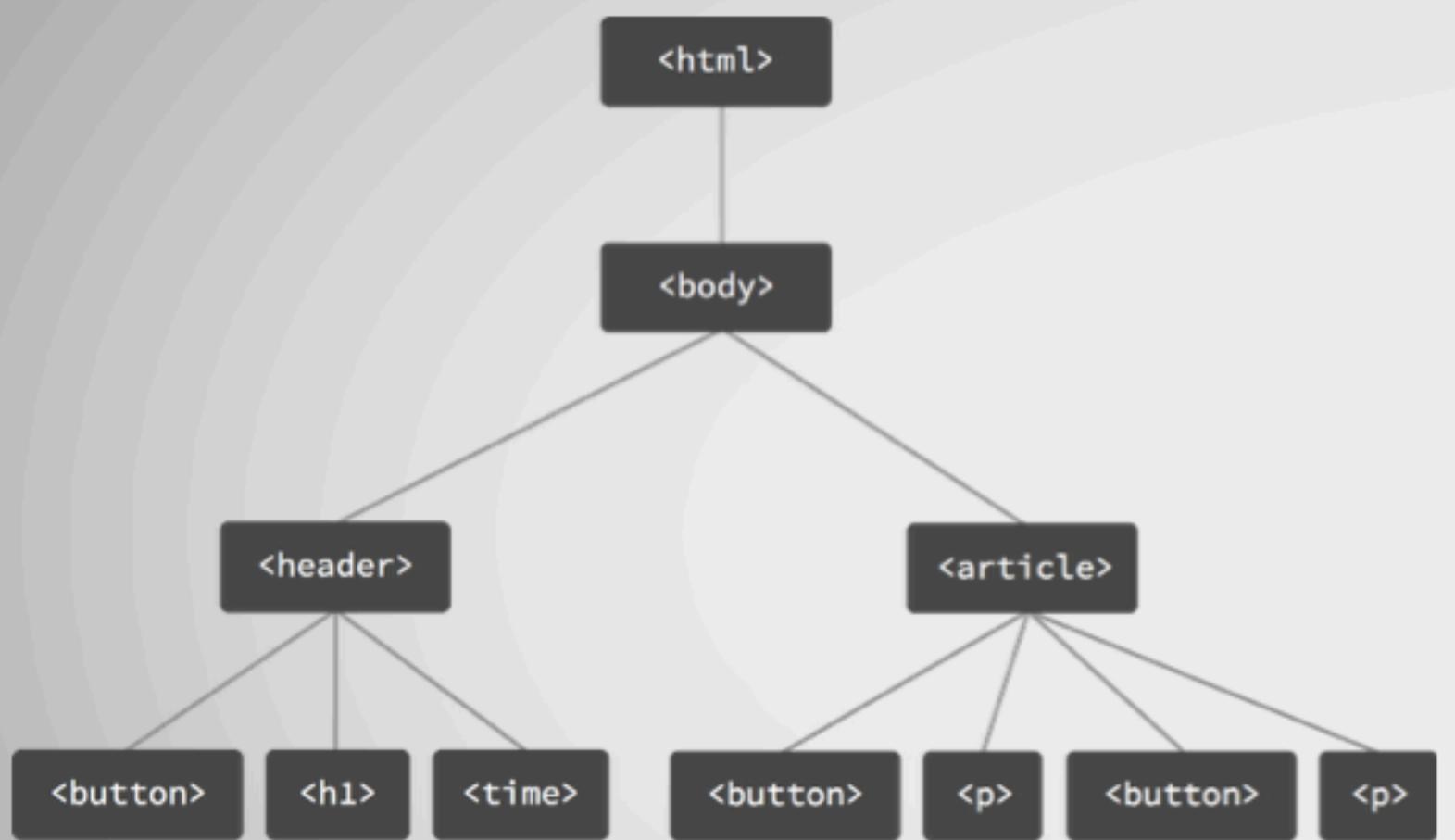
```
header {  
    height: 200px;  
    background: url(fadey-header.gif);  
}
```

```
.highlight {  
    font-size: 14px;  
    word-wrap: break-word;  
}
```





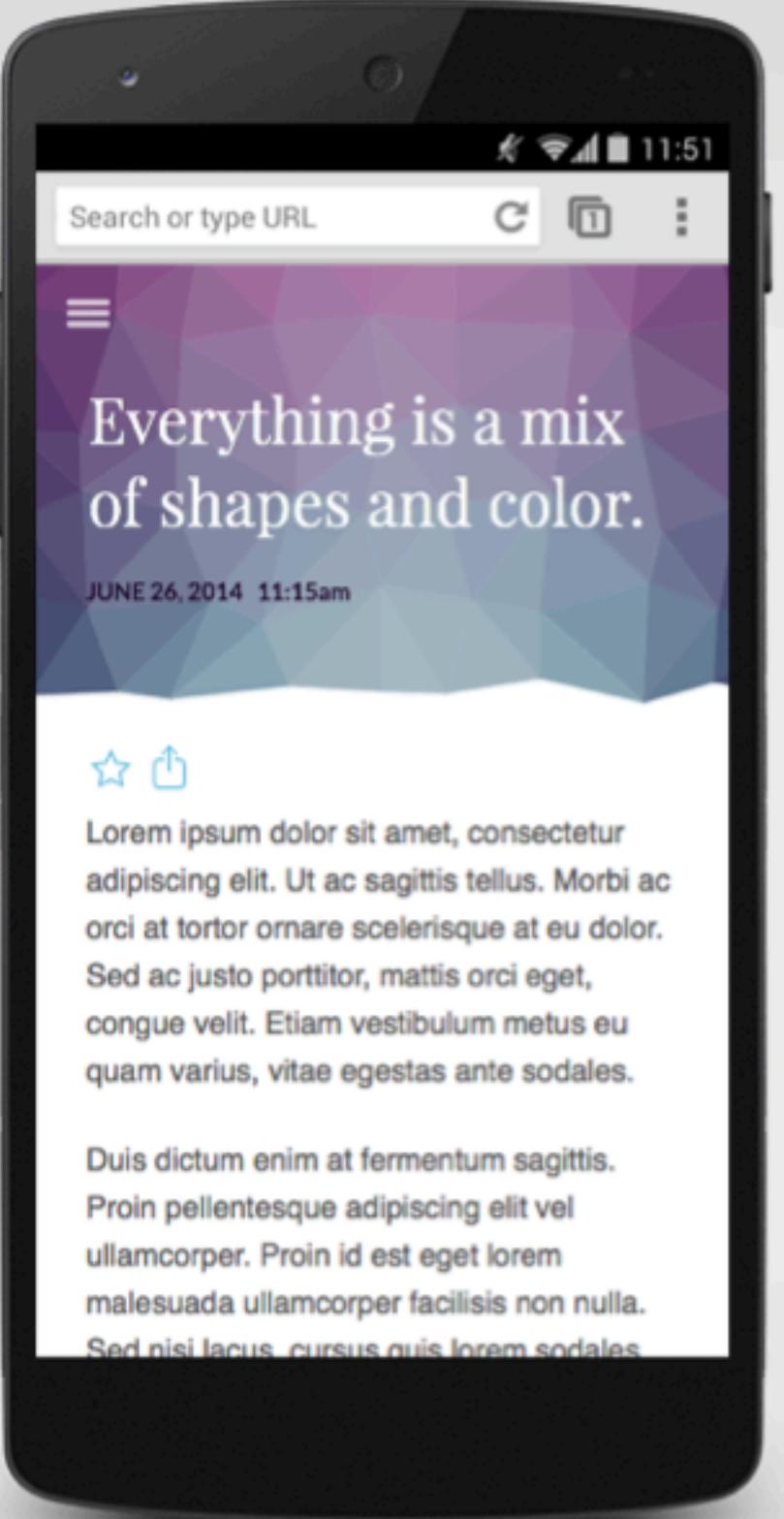
Layout.



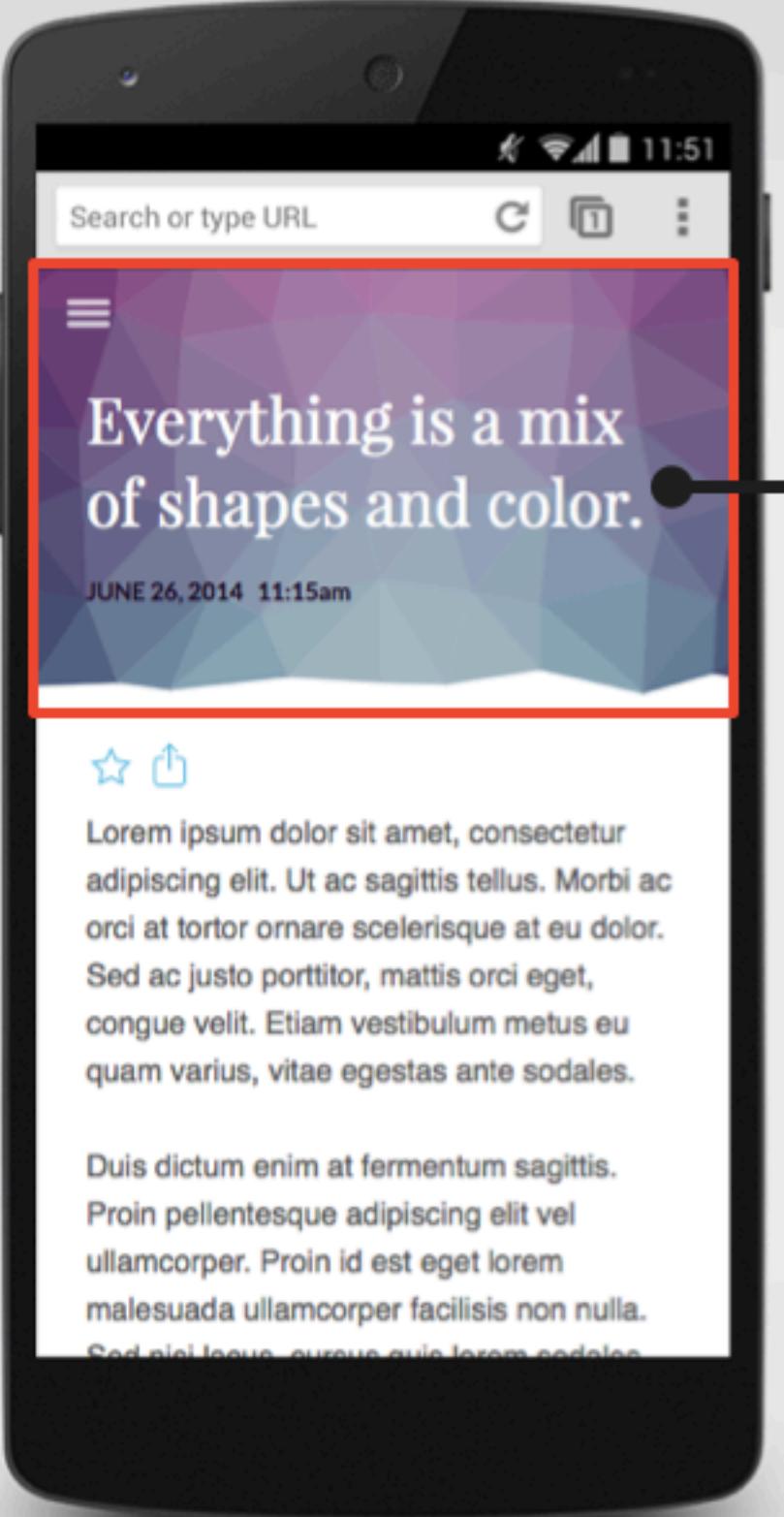


Paint.

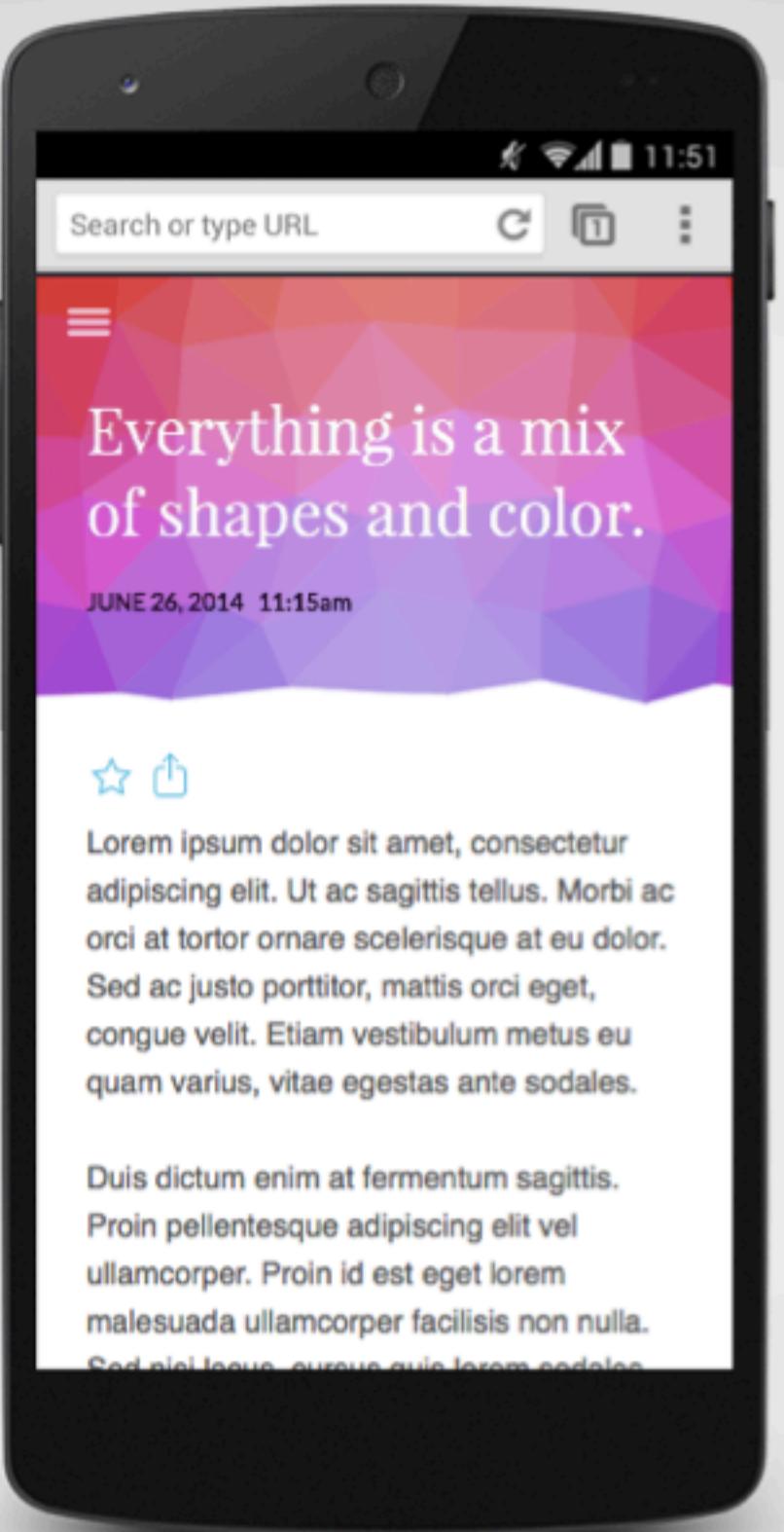




Composite.







We can do better...

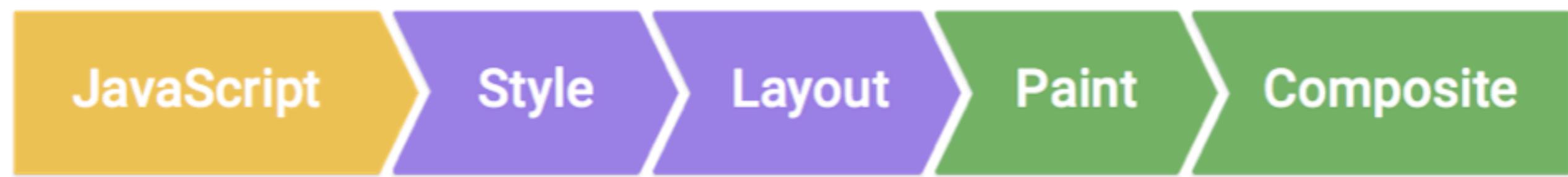
Everything is a mix
of shapes and color.

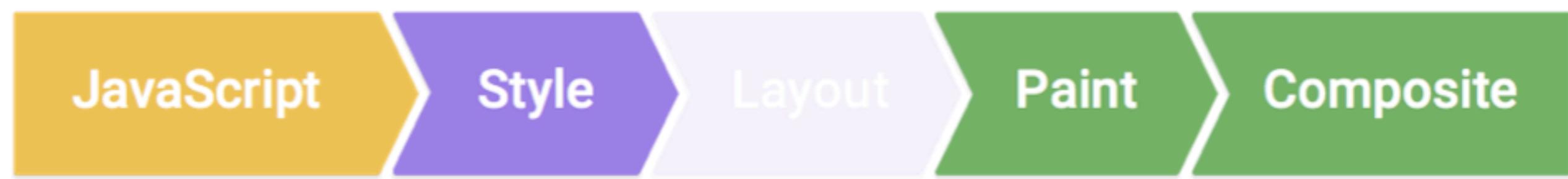
JUNE 26, 2014 11:15am

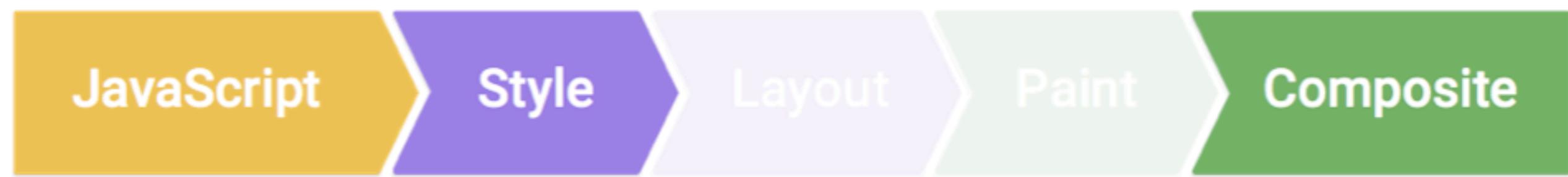
☆

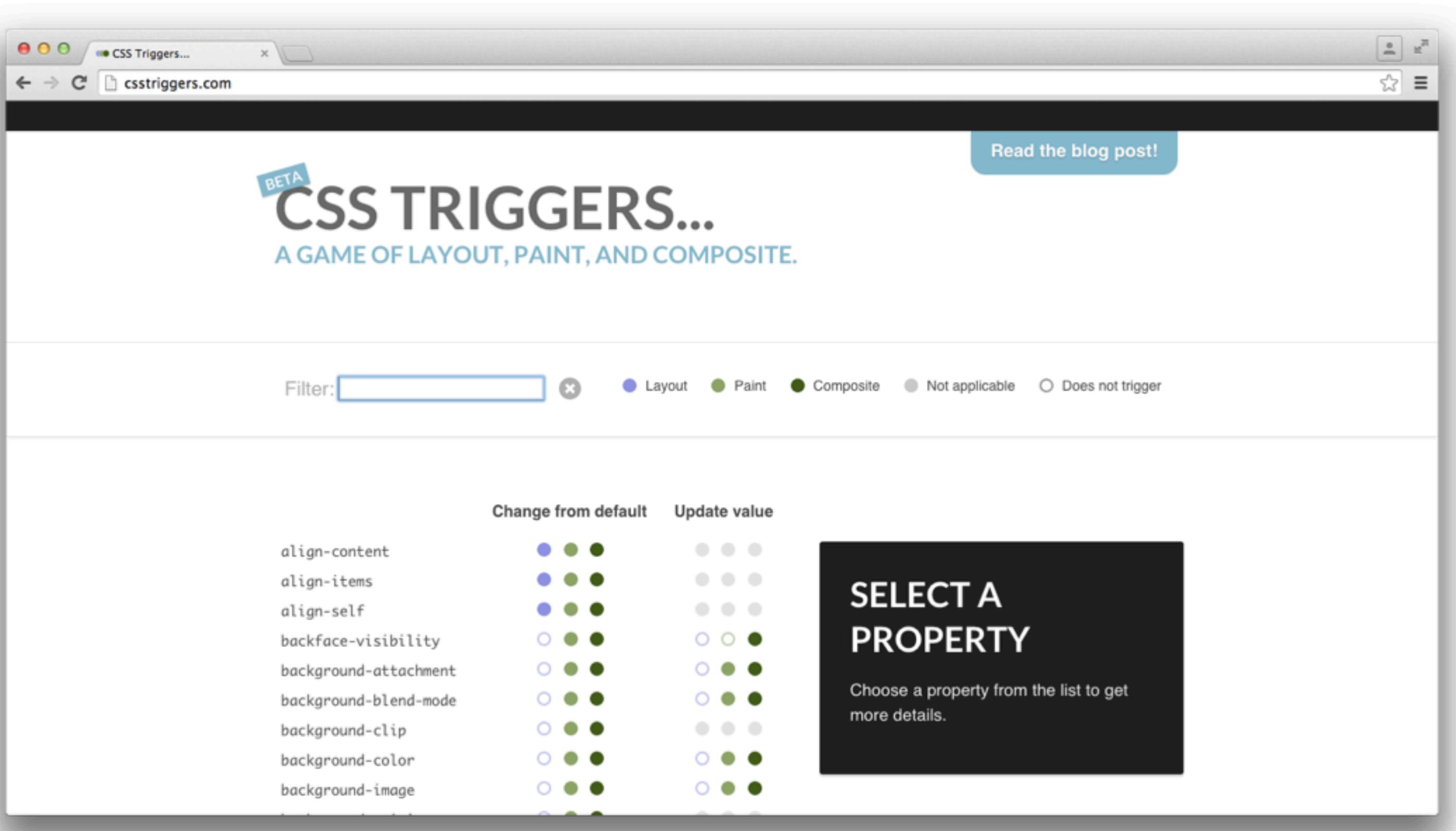
lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac sagittis tellus. Morbi ac orci at tortor ornare scelerisque at eu dolor. Sed ac justo porttitor, mattis orci eget, congue velit. Etiam vestibulum metus eu quam varius, vitae egestas ante sodales.

Duis dictum enim at fermentum sagittis. Proin pellentesque adipiscing elit vel ullamcorper. Proin id est eget lorem malesuada ullamcorper facilisis non nulla.









CSS Triggers... csstriggers.com

BETA

CSS TRIGGERS...

A GAME OF LAYOUT, PAINT, AND COMPOSITE.

Read the blog post!

Filter: Layout Paint Composite Not applicable Does not trigger

left

Change from default Update value

| | | | | | |
|-----------------------|--------|-------|-----------|----------------|------------------|
| align-content | Layout | Paint | Composite | Not applicable | Does not trigger |
| align-items | Layout | Paint | Composite | Not applicable | Does not trigger |
| align-self | Layout | Paint | Composite | Not applicable | Does not trigger |
| backface-visibility | Layout | Paint | Composite | Not applicable | Does not trigger |
| background-attachment | Layout | Paint | Composite | Not applicable | Does not trigger |
| background-blend-mode | Layout | Paint | Composite | Not applicable | Does not trigger |
| background-clip | Layout | Paint | Composite | Not applicable | Does not trigger |
| background-color | Layout | Paint | Composite | Not applicable | Does not trigger |
| background-image | Layout | Paint | Composite | Not applicable | Does not trigger |

SELECT A PROPERTY

Choose a property from the list to get more details.

The screenshot shows a web browser window with the URL 'csstriggers.com'. At the top, there's a header with the text 'BETA' and 'CSS TRIGGERS...', followed by 'A GAME OF LAYOUT, PAINT, AND COMPOSITE.' Below this is a 'Read the blog post!' button. The main area features a search bar with the word 'left' typed into it. To the right of the search bar is a legend with colored circles: blue for 'Layout', green for 'Paint', and dark green for 'Composite'. Below the search bar are two buttons: 'Change from default' and 'Update value'. A table lists various CSS properties with their corresponding triggers: align-content, align-items, align-self, backface-visibility, background-attachment, background-blend-mode, background-clip, background-color, and background-image. Each property has three columns of circles representing the three types of triggers. To the right of the table is a large gray box with the text 'SELECT A PROPERTY' and the instruction 'Choose a property from the list to get more details.'

The frame game.

FOOLSS

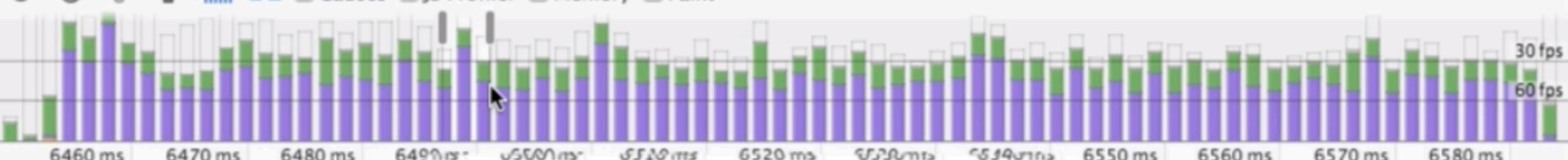
NOT

RULES

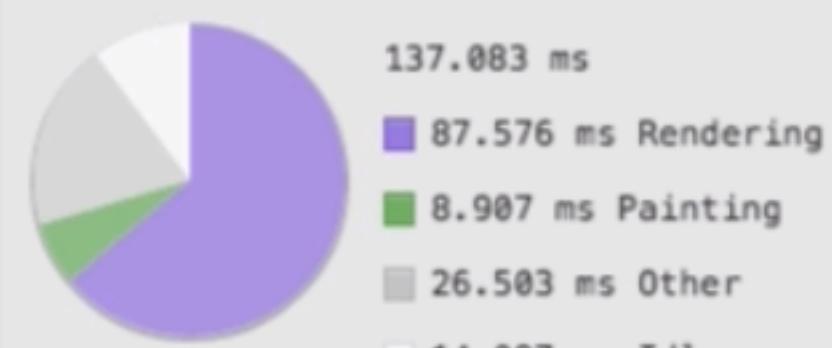
Elements Network Sources Timeline Profiles Resources Audits Console

>≡ ⚙️ 🖼

● ⚡ ⚡ Causes JS Profiler Memory Paint

**Main Thread****CompositorTileWorker1/20503****Details**

Range 6.45 s – 6.59 s

Aggregated TimeConsole Search Emulation **Rendering**

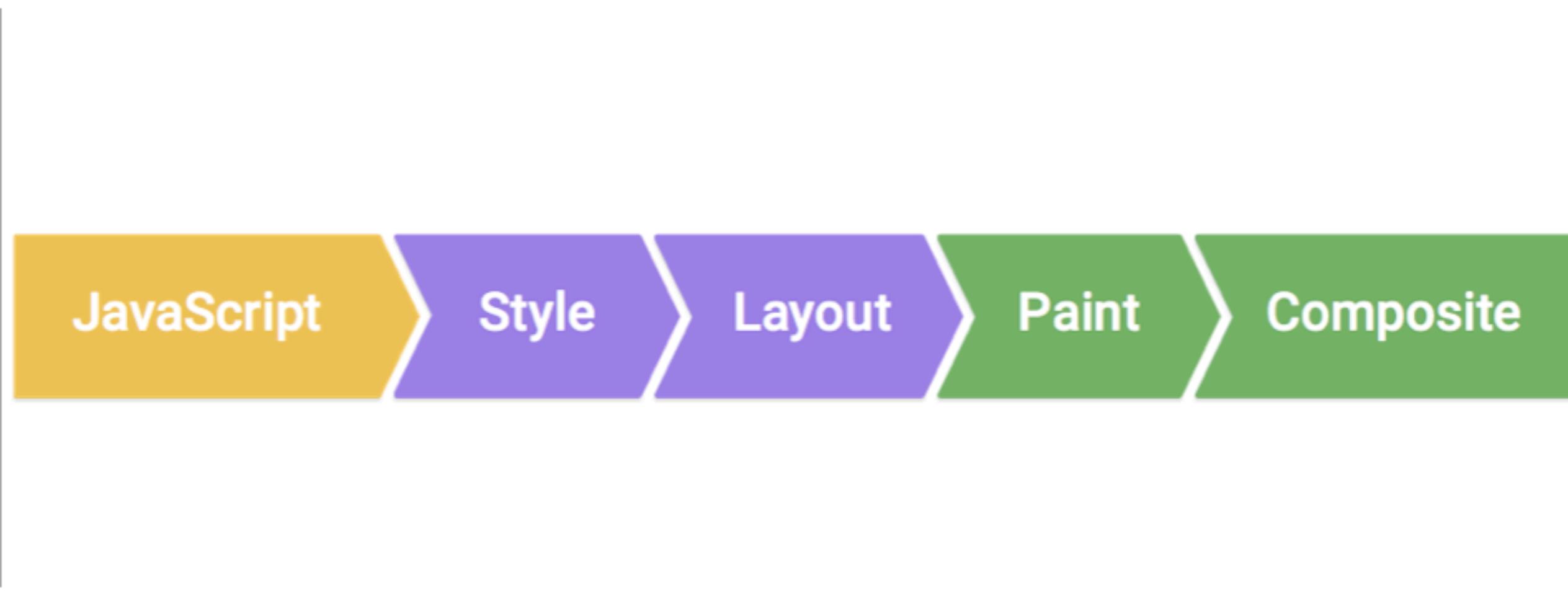
- Show paint rectangles
- Show composited layer borders
- Show FPS meter
- Enable continuous page repainting
- Show potential scroll bottlenecks

JS

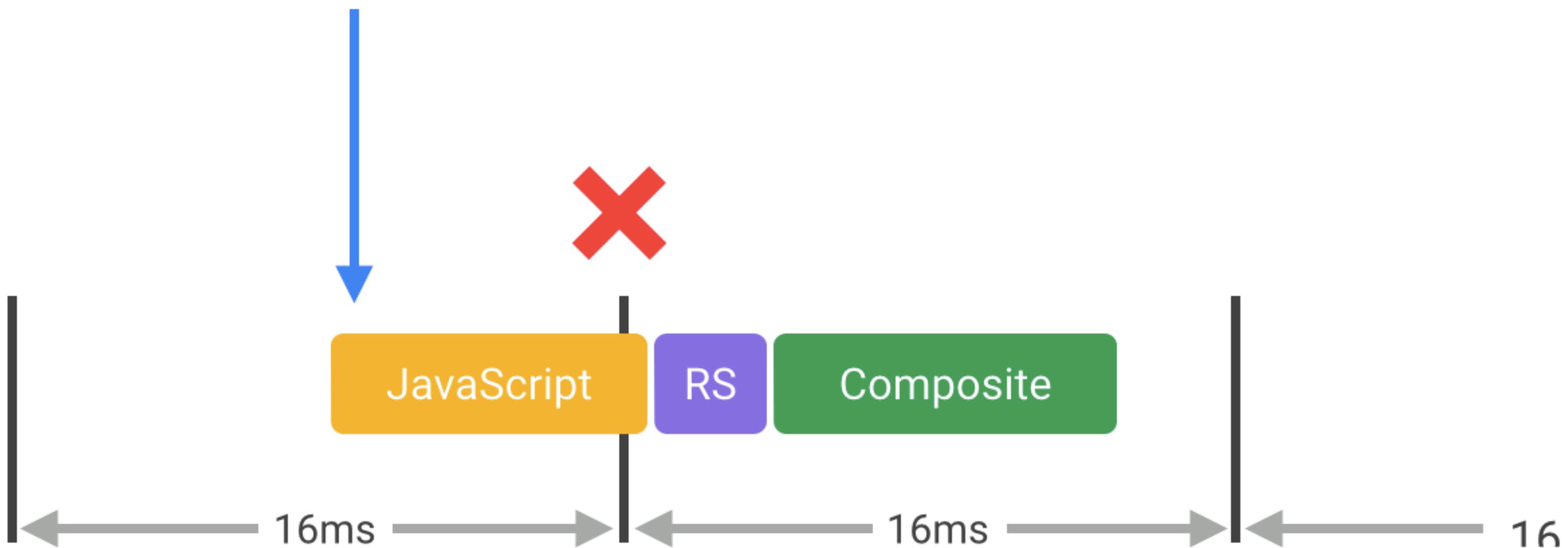
Badly-timed Scripts.

Start

End



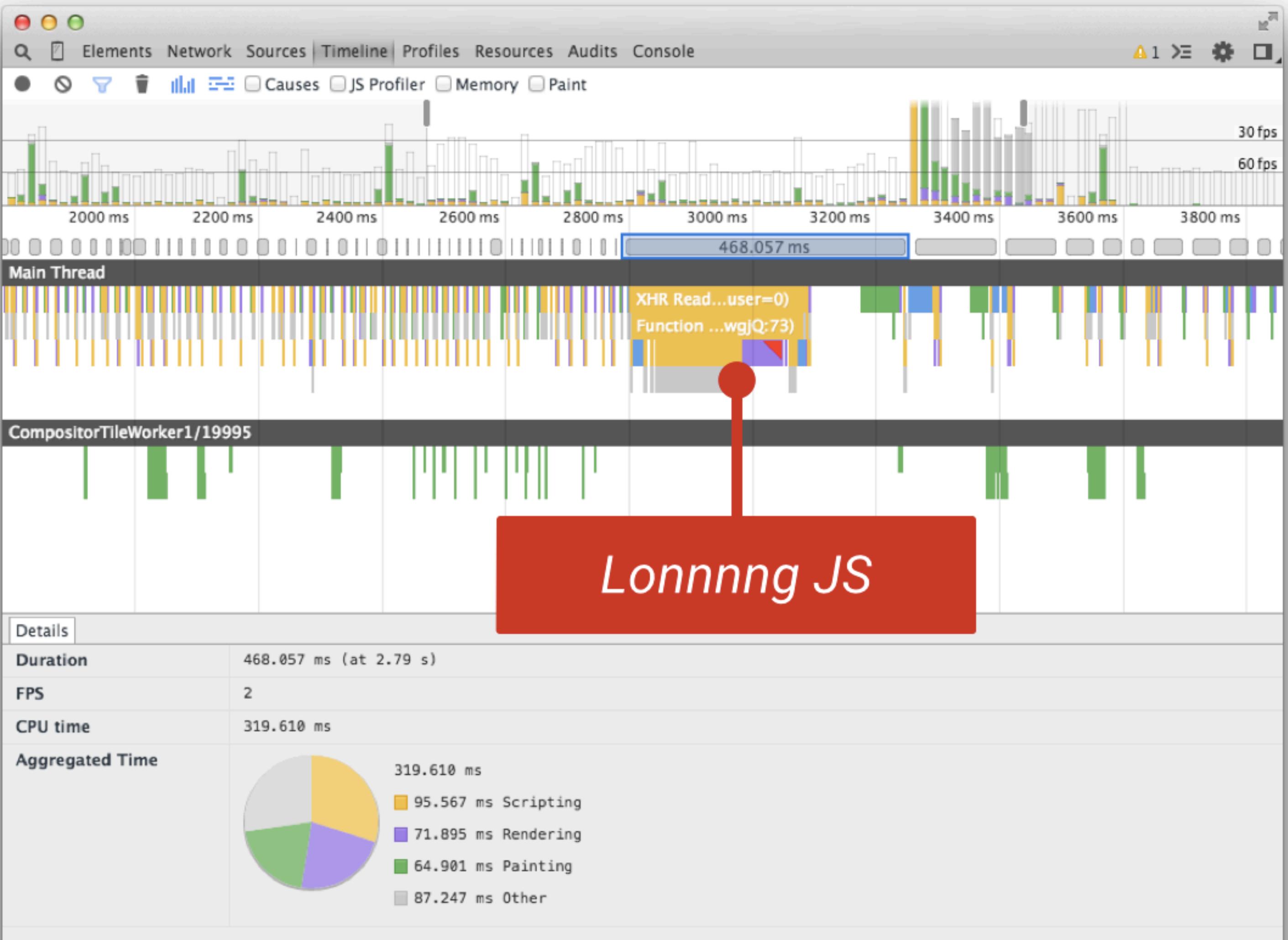
setTimeout fires



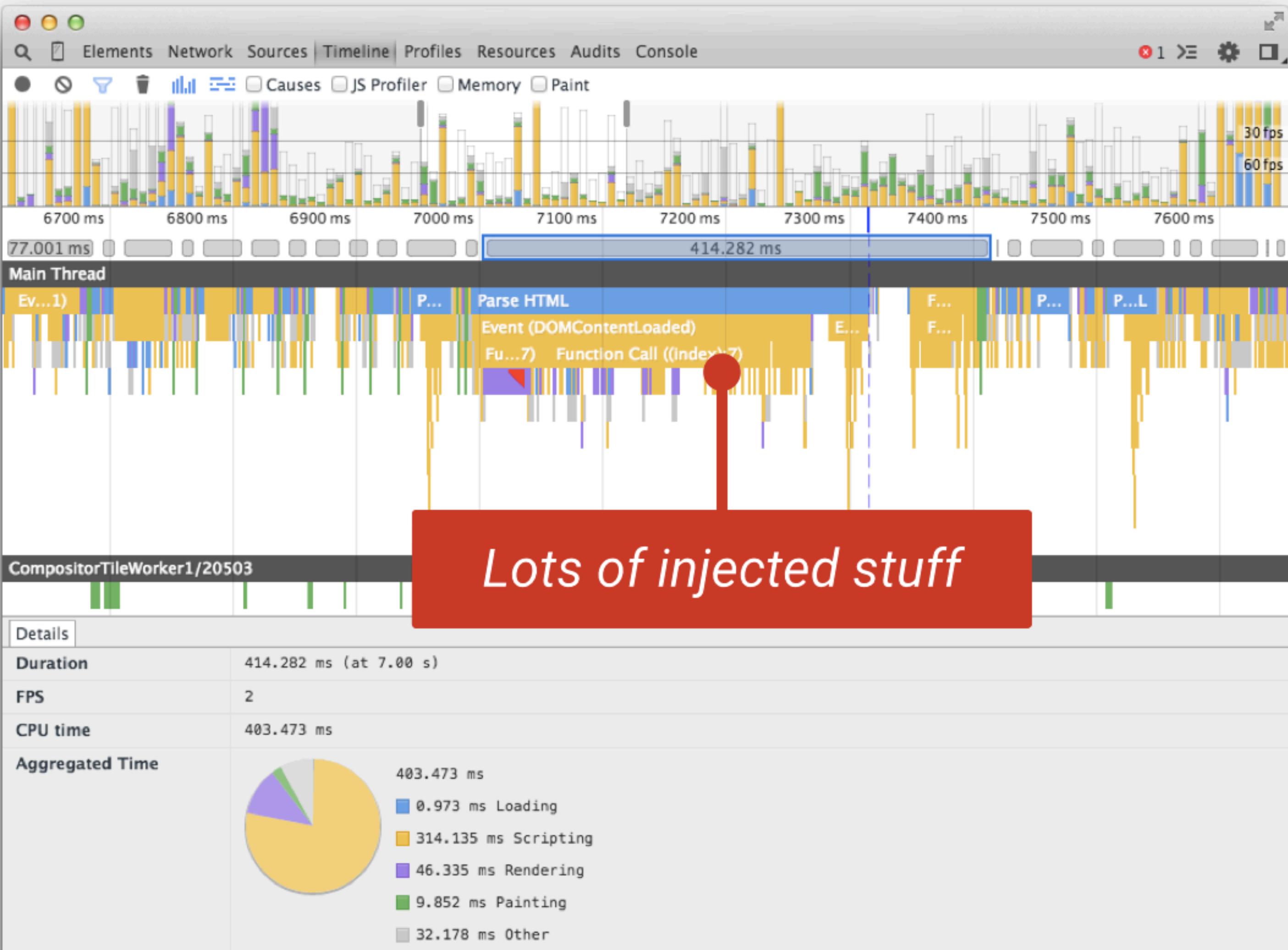


```
requestAnimationFrame(  
    totesRadAnimation  
);
```

Long-running Scripts.



```
var dataSortWorker =  
  new Worker("sort-worker.js");  
  
dataSortWorker.postMessage(dataToSort);  
  
// The main thread is now free to  
// continue working on other things...  
  
dataSortWorker.addEventListener('message',  
  function(evt) {  
    var sortedData = e.data;  
    // Update data on screen in a rAF...  
  });
```



```
var taskList = breakTasksDown(monsterTaskList);
requestAnimationFrame(processTaskList);

function processTaskList(taskStartTime) {
    var taskFinishTime;
    do {
        var nextTask = taskList.pop();

        processTask(nextTask);

        // Go again if there's enough time...
        taskFinishTime = window.performance.now();
    } while (taskFinishTime - taskStartTime < 3);

    if (taskList.length > 0)
        requestAnimationFrame(processTaskList);
}
```

“*But workers don’t have DOM access.*”

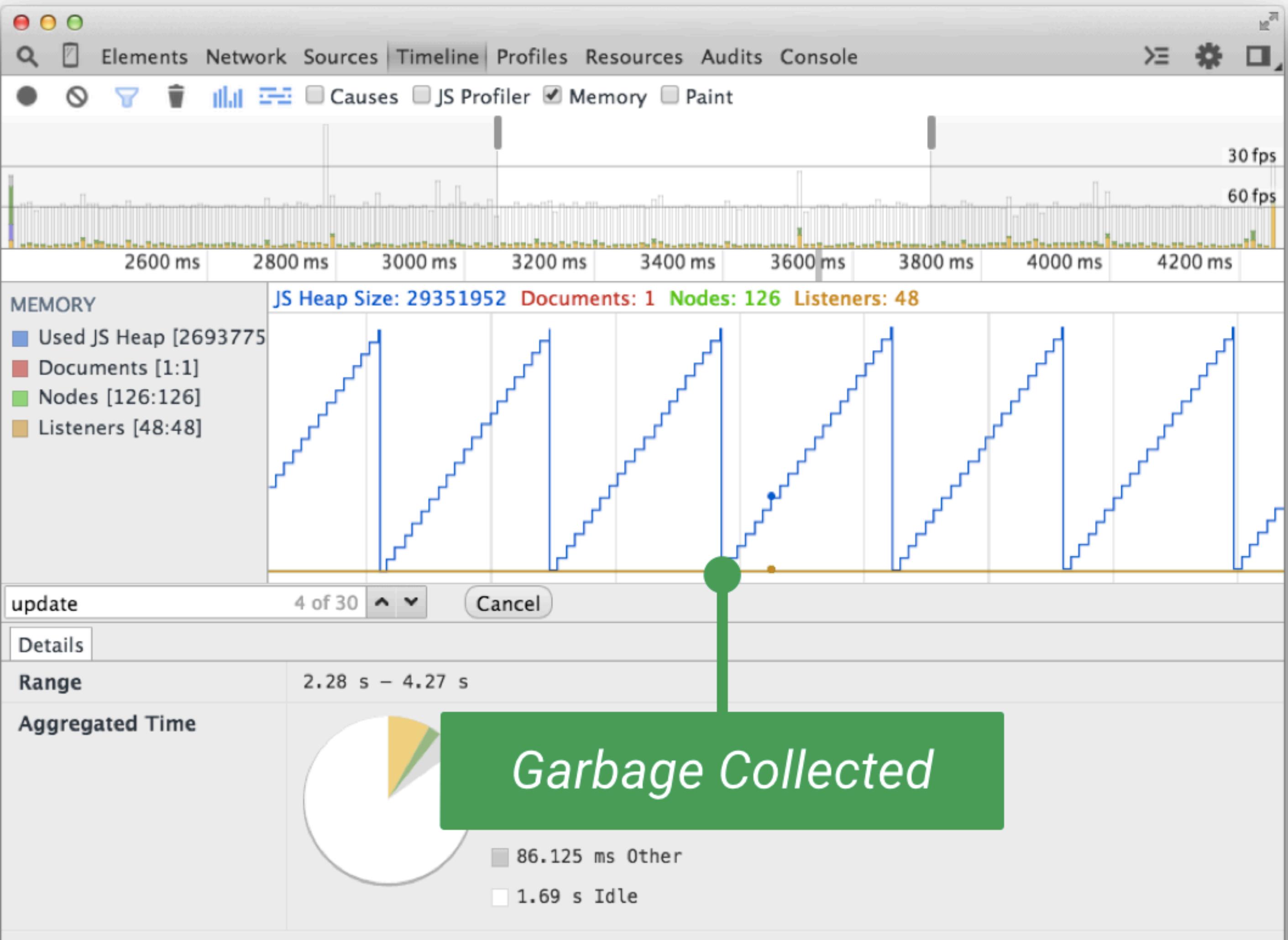
3-4ms per frame
in Animations

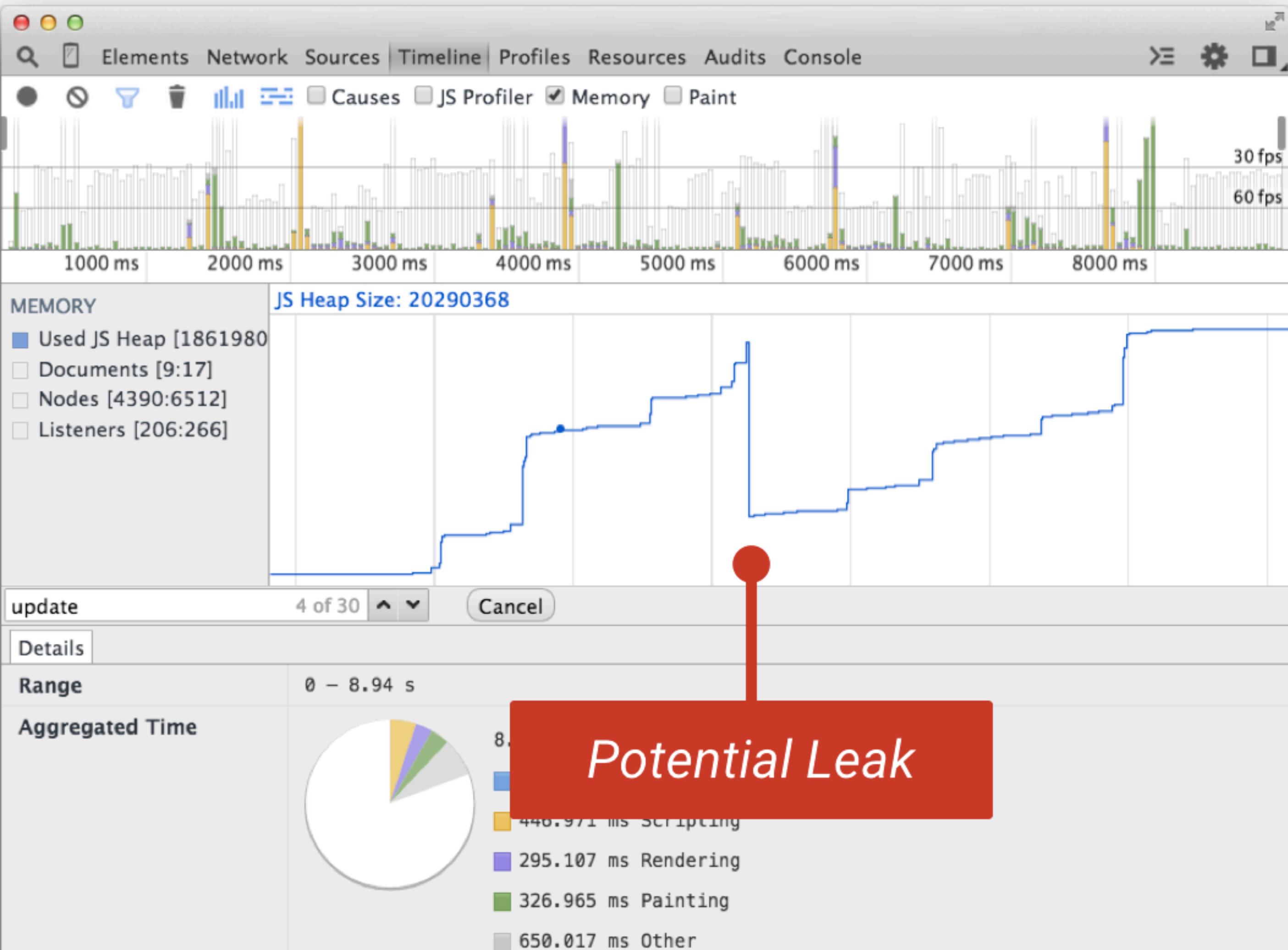
JS

spell check no likey



Garbagey Scripts.

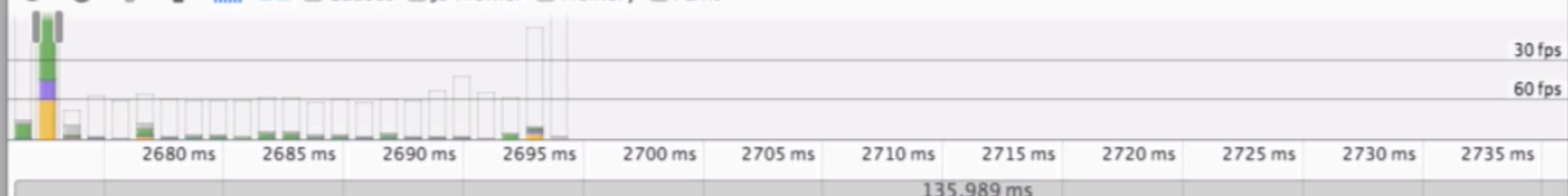






Addy Osmani

JavaScript Memory Masterclass

Causes JS Profiler Memory Paint**Main Thread**

Animation Frame Fired (1)

Function Call (cds.min.js:32)

Rec...yle

Com

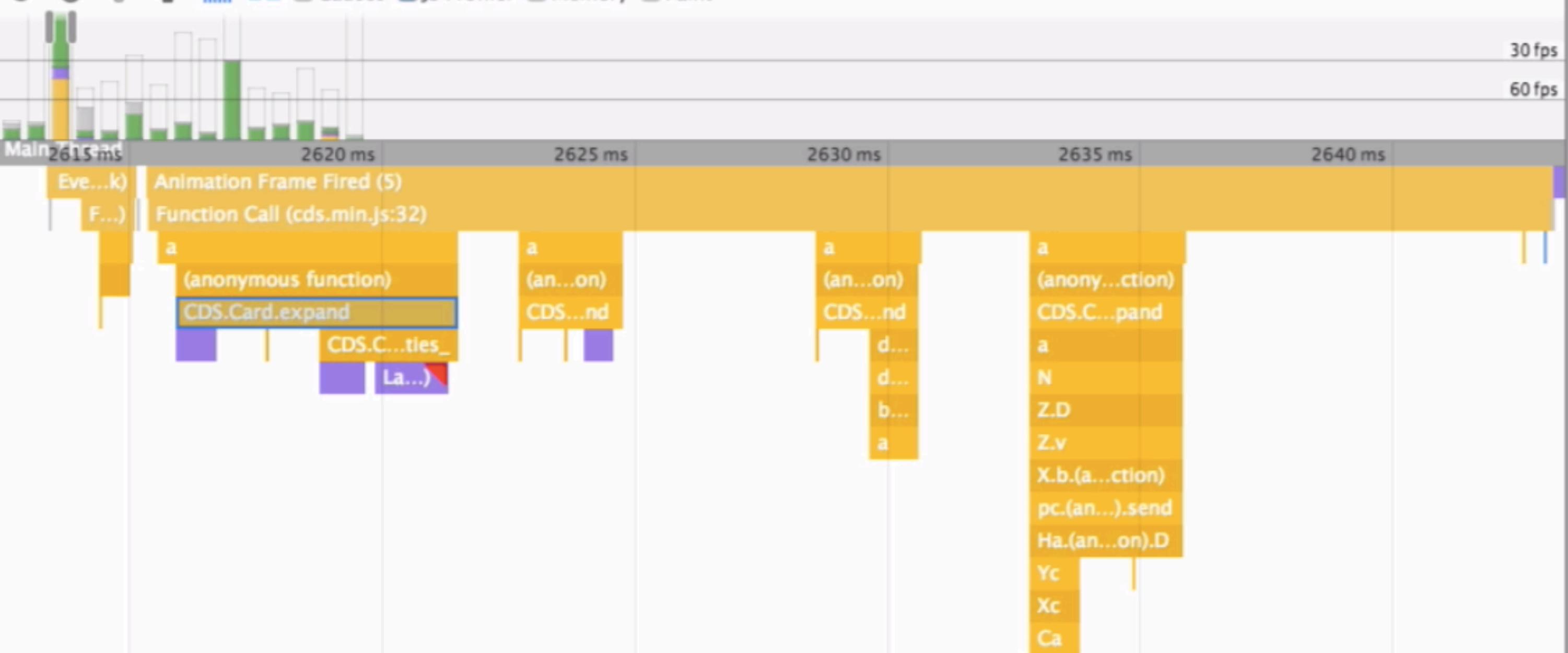
CompositorTileWorker1/20503**Details****Type** Function Call**Self Time** 16.168 ms**Start Time** 2.70 s**Location** [cds.min.js:32](#)**Aggregated Time**

19.051 ms

16.168 ms Scripting (Self)

2.883 ms Rendering

We can do better...

Causes JS Profiler Memory Paint


CompositorTileWorker1/20503

Details

| | |
|------|----------|
| Type | JS Frame |
|------|----------|

| | |
|-----------|----------|
| Self Time | 2.043 ms |
|-----------|----------|

| | |
|------------|--------|
| Start Time | 2.62 s |
|------------|--------|

| | |
|---------|---------------------------------|
| Details | CDS.Card.expand |
|---------|---------------------------------|

| | |
|-----------------|--|
| Aggregated Time | |
|-----------------|--|

5.591 ms

2.043 ms Scripting (Self)

0.407 ms Scripting (Children)



Styles.

Which is fastest in Blink today?

1 `div.box:not(:empty):last-of-type .title`

2 `.box--last > .title-container > .title`

3 `.box:nth-last-child(-n+1) .title`

Which is fastest in Blink today?



`div.box:not(:empty):last-of-type .title`



`.box--last > .title-container > .title`



`.box:nth-last-child(-n+1) .title`

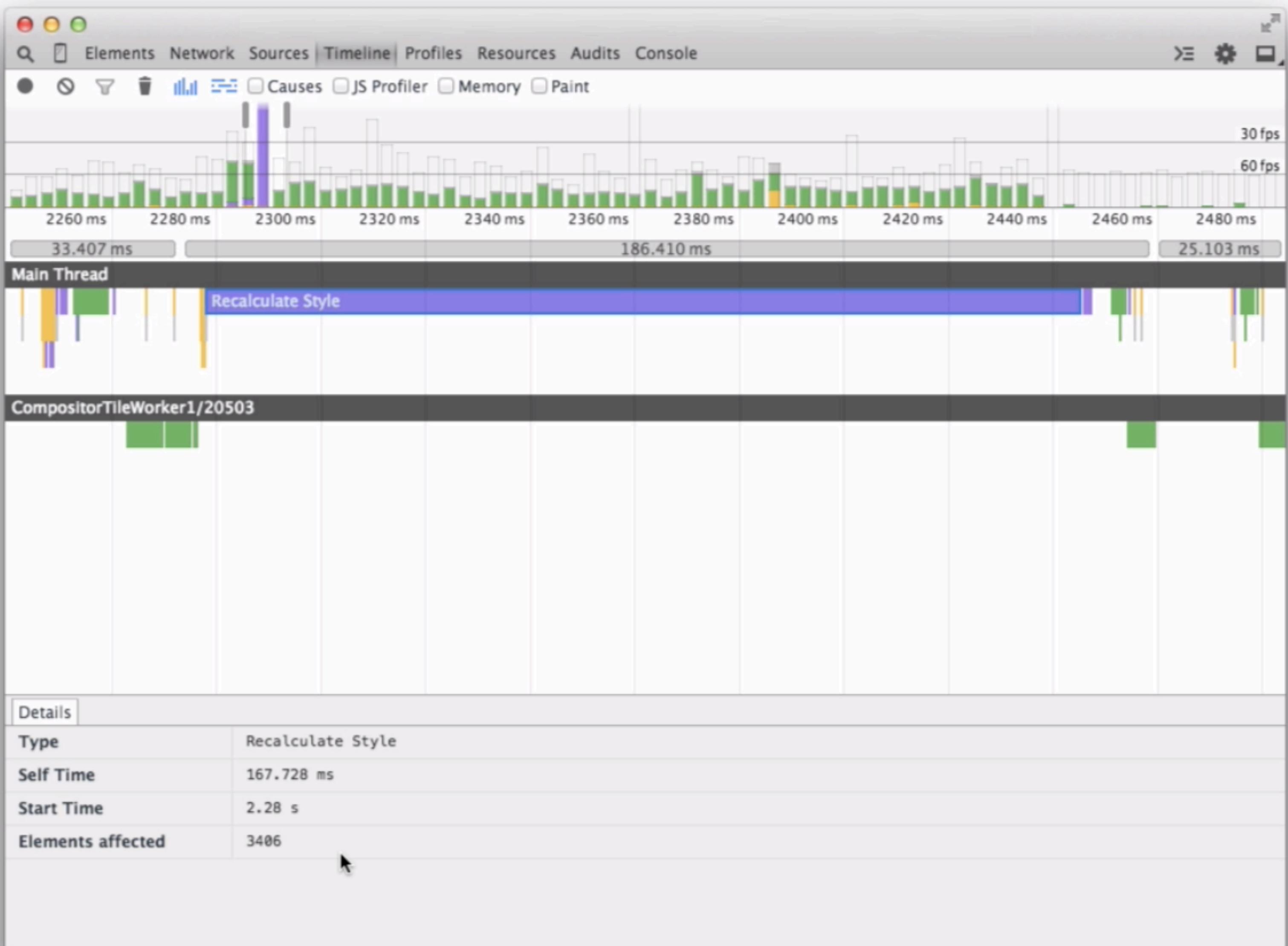


2 orders of magnitude slower!

```
// Assume many divs...
var divs = document
    .querySelectorAll('div');

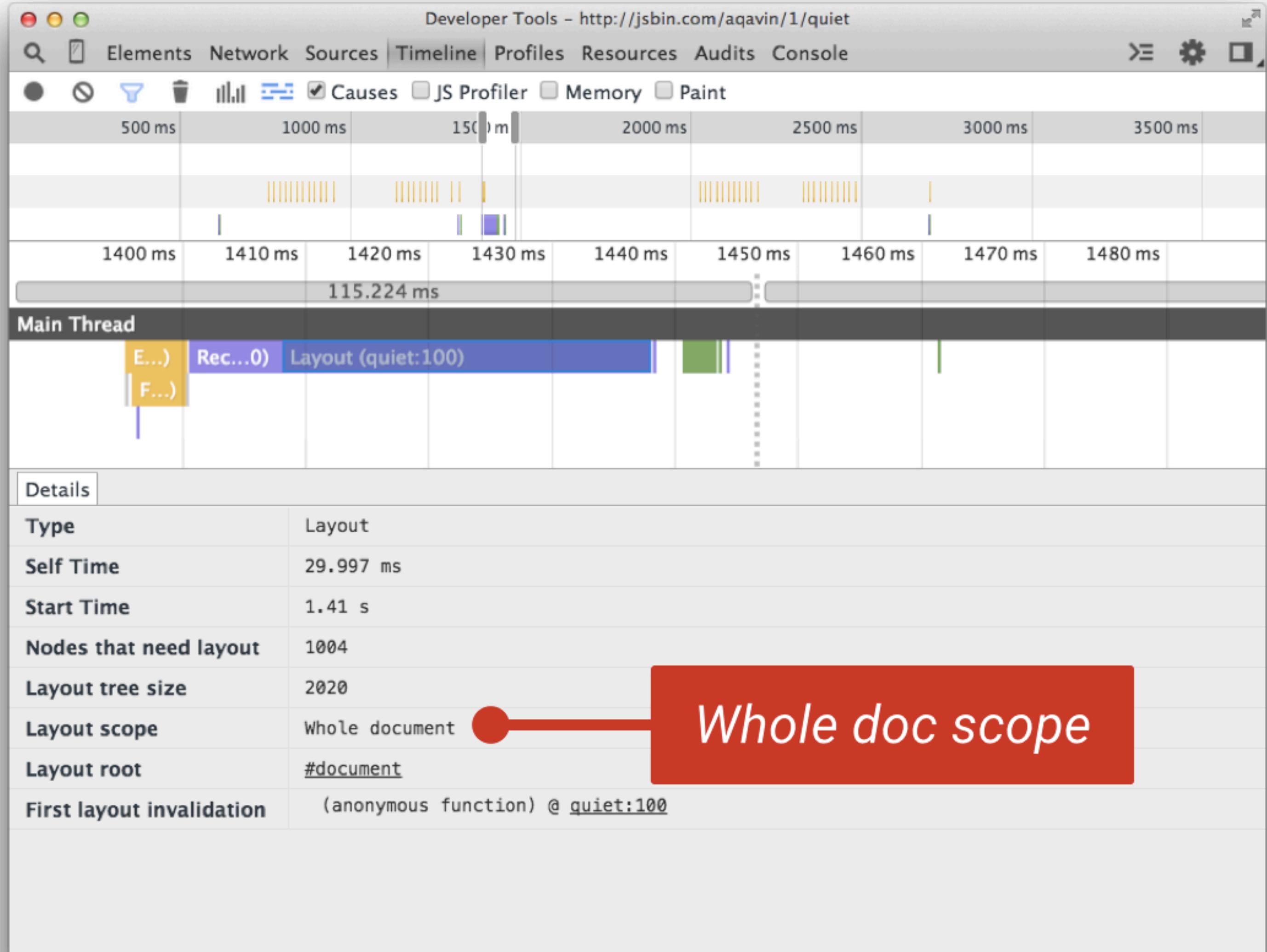
for (var i = 0; i < divs.length; i++) {
    divs[i].classList.toggle('woopwoop');
    divs[i].style.color = 'palegoldenrod';
}

body.classList.add('norvanmunkeh');
```

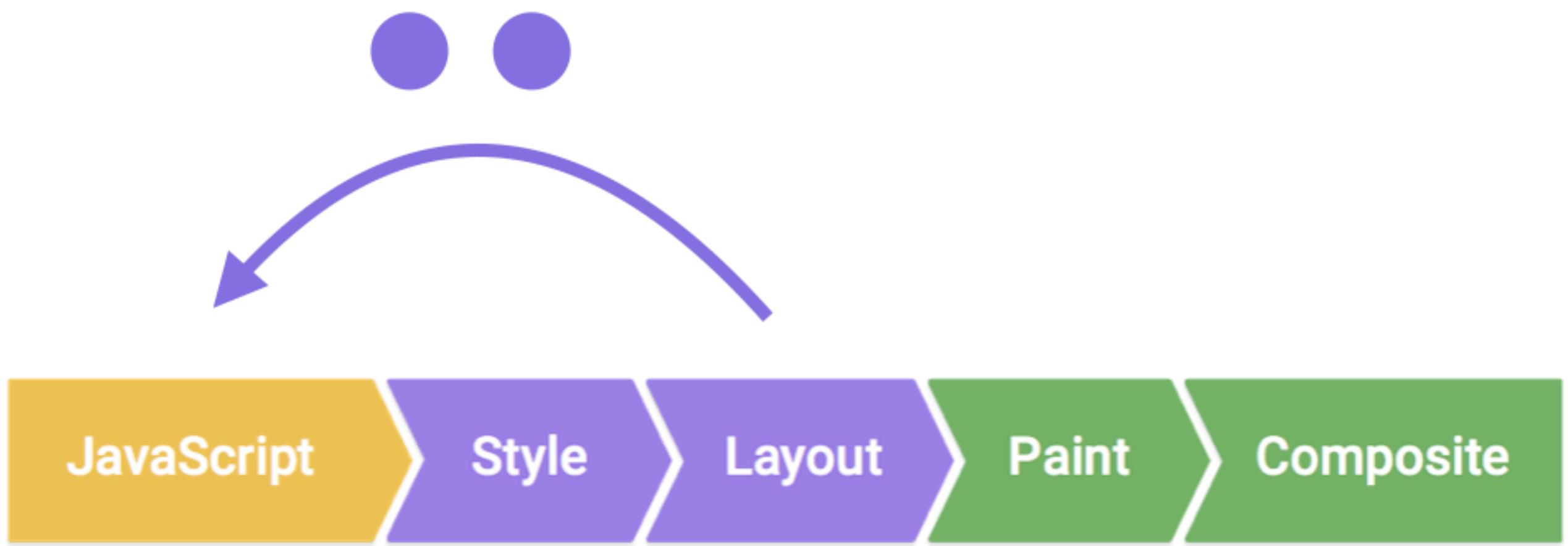


1. Did I limit the scope enough?
2. Are the selectors fast?

Layout.

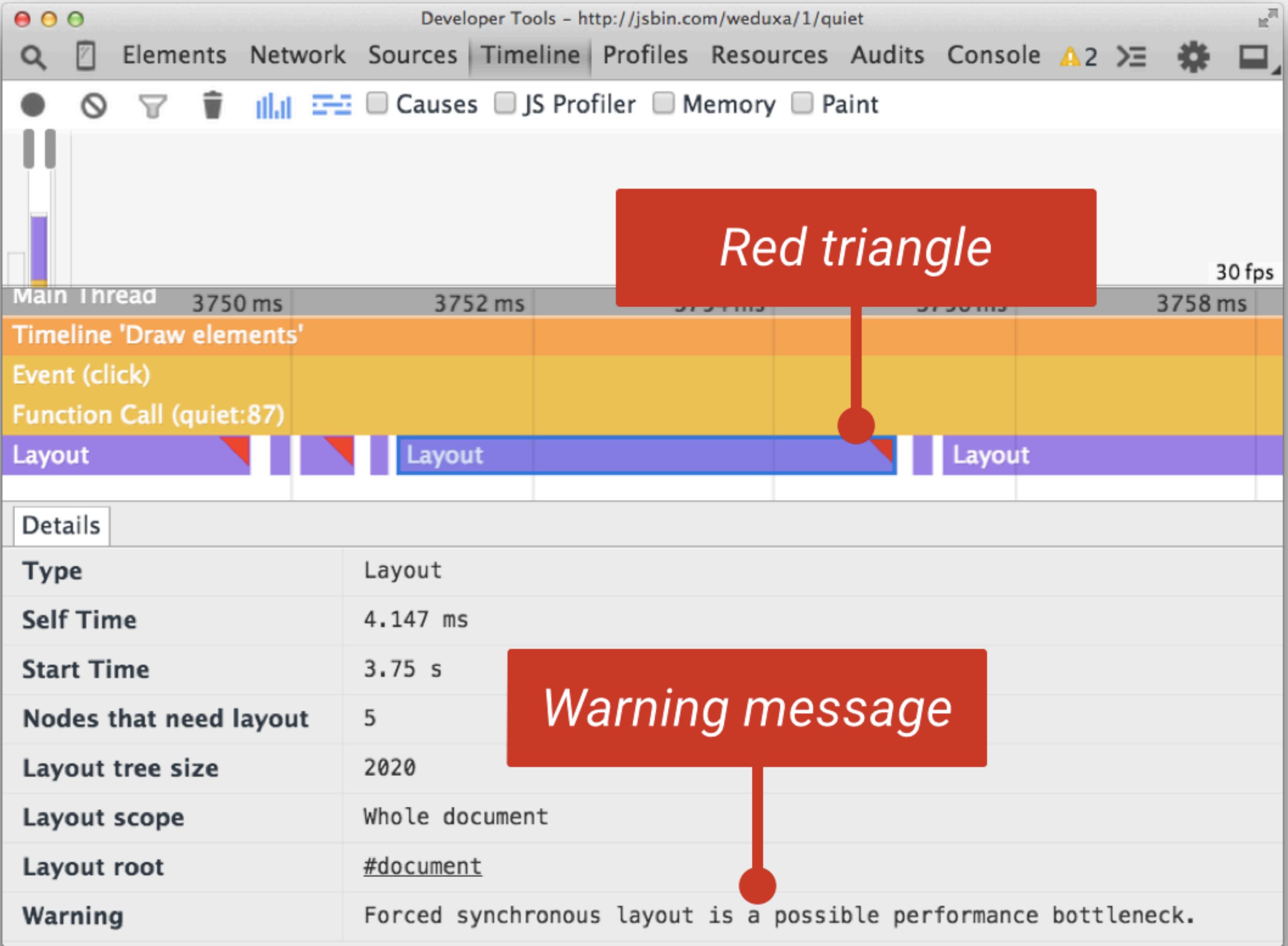


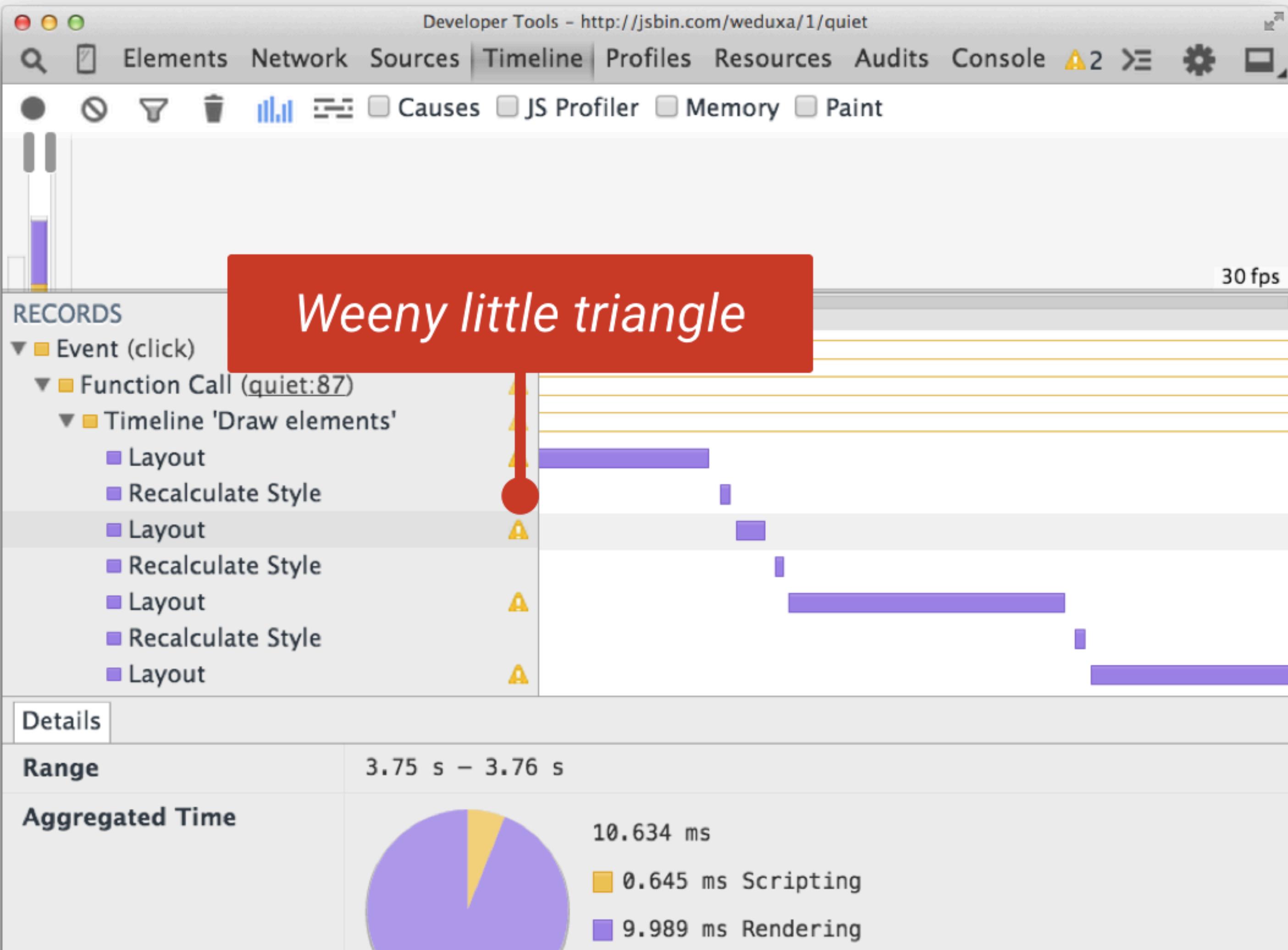
clientHeight, clientLeft,
clientTop, clientWidth, focus(),
getBoundingClientRect(),
getClientRects(), innerText,
offsetHeight, offsetLeft,
offsetParent, offsetTop,
offsetWidth, outerText,
scrollByLines(), scrollByPages(),
scrollHeight, scrollIntoView(),
scrollIntoViewIfNeeded(),
scrollLeft, scrollTop, scrollWidth





```
// Set the width.  
el.style.width = '200px';  
  
// Then ask for it. Bad.  
var width = el.offsetWidth;
```





Read
then
write



```
// Get the last known width.  
var width = el.offsetWidth;  
  
// Then update it.  
el.style.width = '200px';
```

It gets worse...



```
// Paragraphs should match the div width.  
for (var p = 0; p < paragraphs.length; p++) {  
    var para = paragraphs[p];  
  
    // Read a layout property...  
    var width = div.offsetWidth;  
  
    // Then invalidate layouts with writes.  
    para.style.width = width + 'px';  
}
```

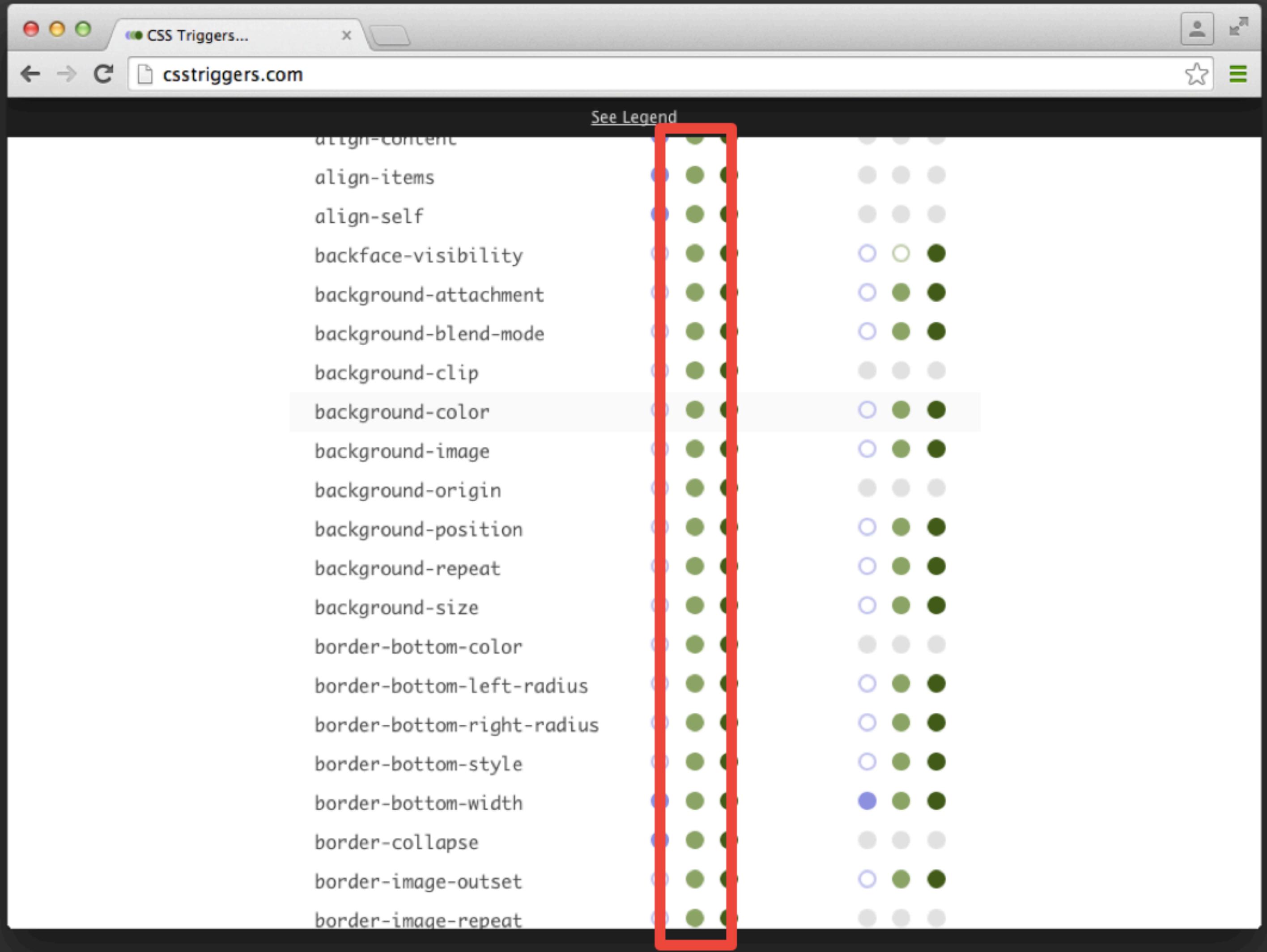


```
// Read a layout property...
var width = div.offsetWidth;

// Paragraphs should match the div width.
for (var p = 0; p < paragraphs.length; p++) {
  var para = paragraphs[p];

  // Then invalidate layouts with writes.
  para.style.width = width + 'px';
}
```

Paint.





PARALL DEMO

A TITLE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eget purus erat, sit amet volutpat felis. Sed et lectus nisl, id ultricies felis. Vivamus non ante augue, sit amet pellentesque magna. Donec eu dapibus sapien. Morbi in ipsum vel massa hendrerit porttitor. Morbi leo magna,



Show paint rectangles



Show FPS meter

Enable continuous page repainting

Show potential scroll bottlenecks

Parallax Demo - DOM + AI X

www.html5rocks.com/static/demos/parallax/demo-1a/demo.html

Elements Network Sources Timeline »

<top frame> ▾ Preserve log

Console Search Emulation **Rendering**

Show paint rectangles

Show composited layer borders

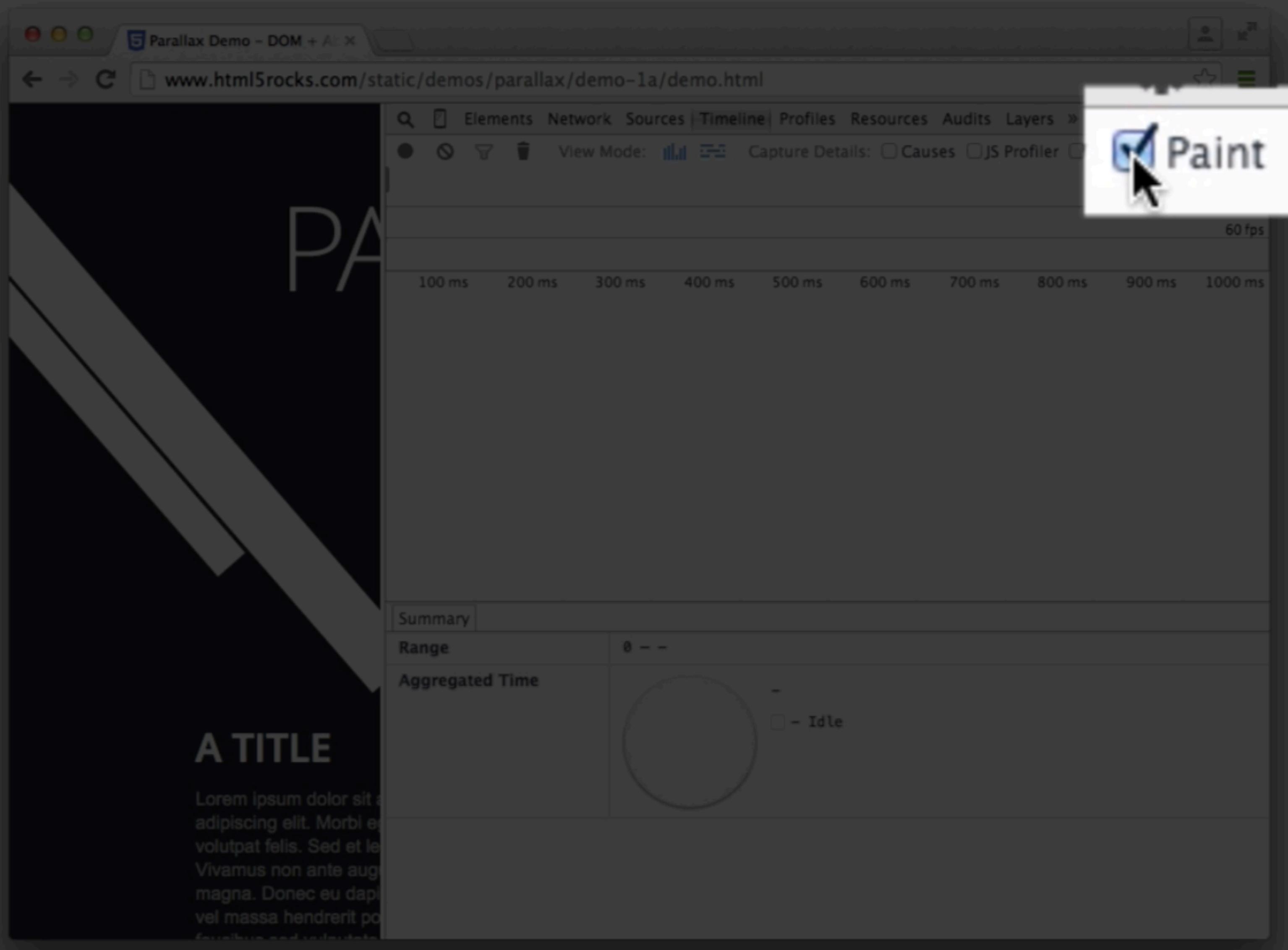
Show FPS meter

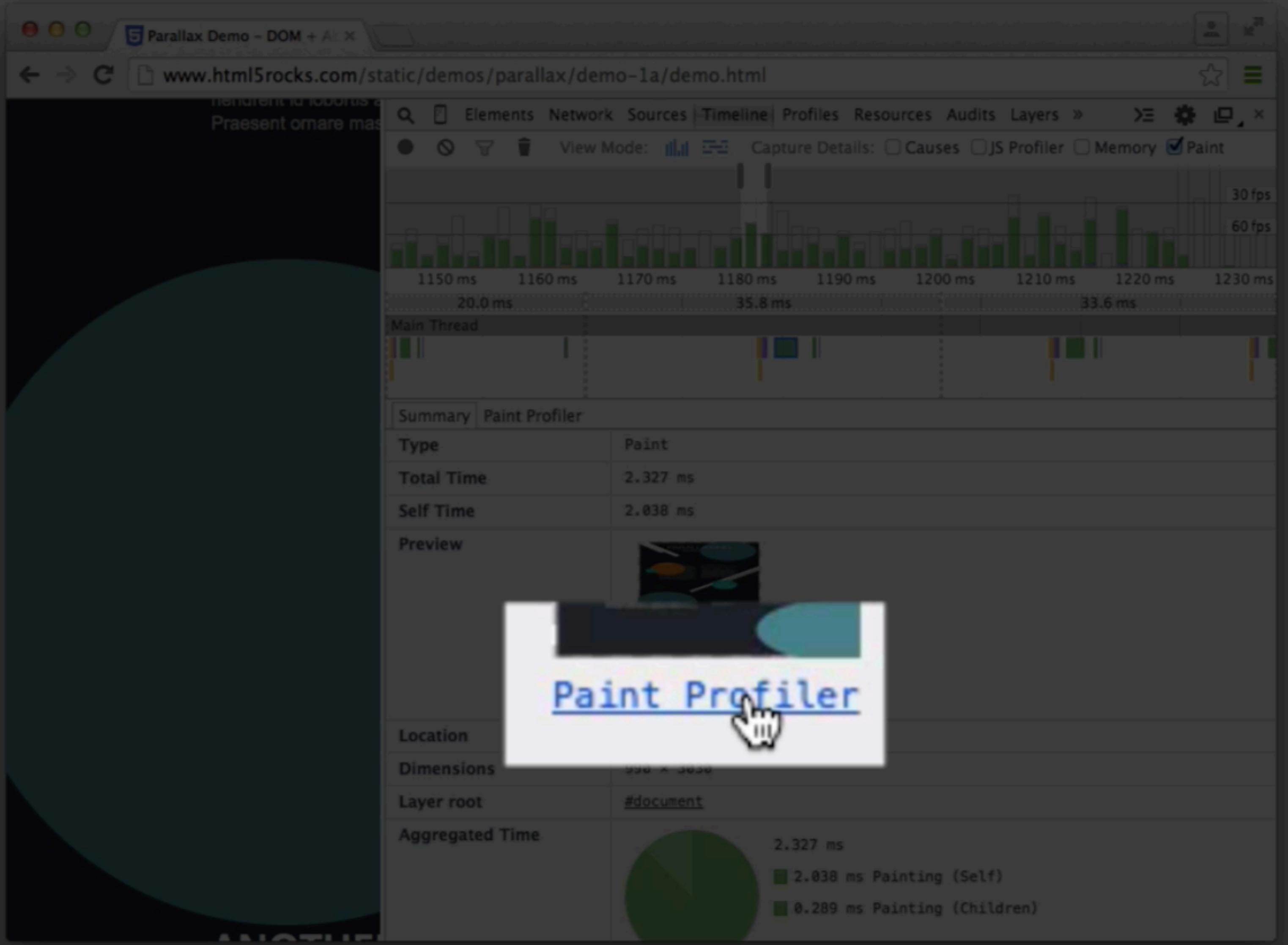
Enable continuous page repainting

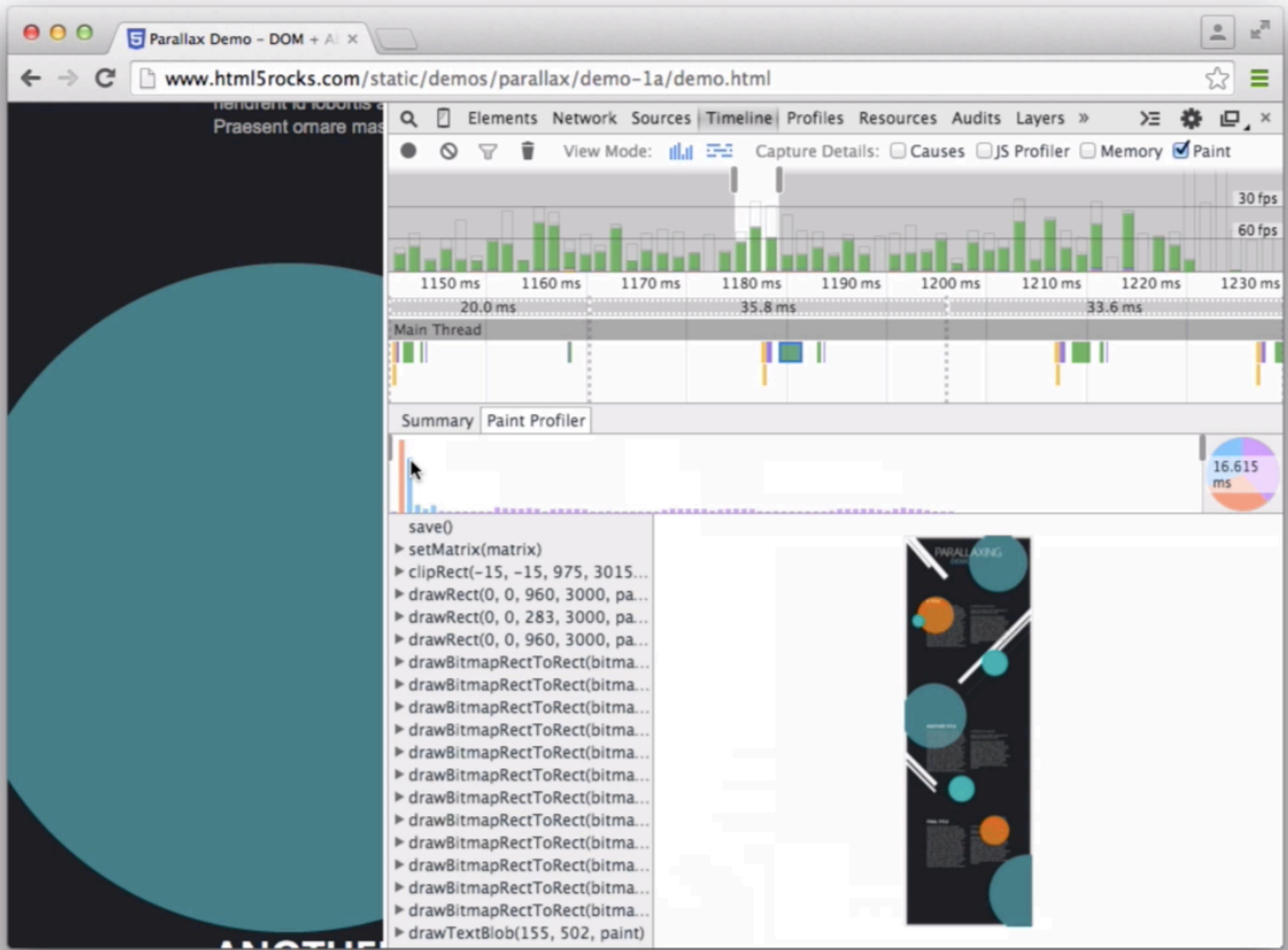
Show potential scroll bottlenecks

The screenshot shows a web browser window displaying a parallax demo from html5rocks.com. The page features a dark green background with two large, semi-transparent circles: one orange and one teal. A block of placeholder text is visible on the left side. The Chrome DevTools rendering panel is open on the right, specifically the 'Rendering' tab. It displays several options for visualizing the rendering process: 'Show paint rectangles' (checked), 'Show composited layer borders' (checked), 'Show FPS meter' (unchecked), 'Enable continuous page repainting' (unchecked), and 'Show potential scroll bottlenecks' (unchecked). The 'Elements' tab is currently selected in the main DevTools header.

We can do better...







1. Can I reduce the paint area?



```
.moving-thing {  
will-change: transform;  
}
```

```
.moving-thing-retro {  
transform: translateZ(0);  
}
```

```
* {  
  will-change: transform;  
  transform: translateZ(0);  
}
```

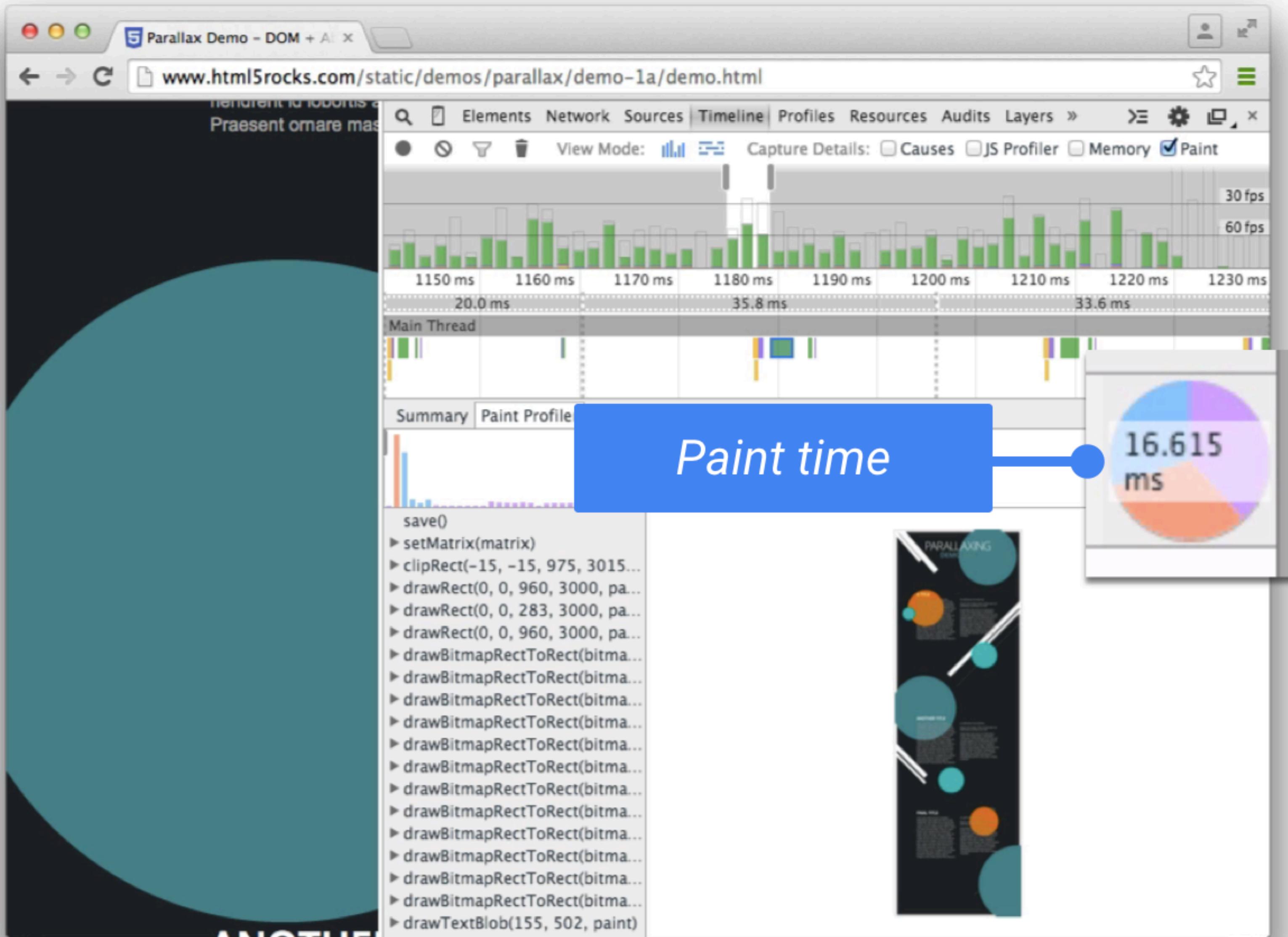
NO.

```
* {  
willz-  
tancez(∅);
```

* {

- ⚠ Will-change memory consumption is too high. Surface area covers 6098969 pixels, budget is the document surface area multiplied by 3 (623824 pixels). All occurrences of will-change in the document are ignored when over budget.

1. Can I reduce the paint area?
2. Can I reduce paint complexity?

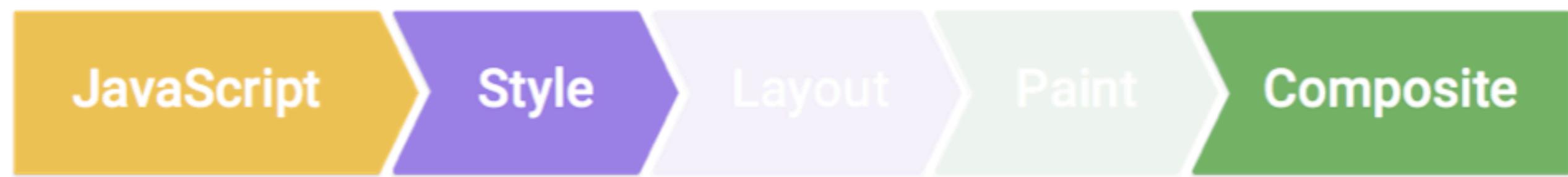


Composite.



| | |
|-----------------|--|
| Position | <code>transform: translate(npx, npx);</code> |
| Scale | <code>transform: scale(n);</code> |
| Rotation | <code>transform: rotate(ndeg);</code> |
| Skew | <code>transform: skew(X Y)(ndeg);</code> |
| Matrix | <code>transform: matrix(3d)(...);</code> |
| Opacity | <code>opacity: 0...1;</code> |

(The element will need to be on its own compositor layer.)



Parallax Demo - rAF + DO

www.html5rocks.com/static/demos/parallax/demo-2/demo.html

The screenshot shows a web page titled "Parallax Demo - rAF + DO" from the URL "www.html5rocks.com/static/demos/parallax/demo-2/demo.html". The page displays a parallax effect where a large teal circle moves across the screen, revealing different background colors (black, orange, green) and text content. The "Elements" tab is selected in the DevTools sidebar. The "Rendering" panel is open, showing the "Paint Profiler" tab. The timeline at the bottom indicates a FPS of 30. The main content area shows several green and yellow rectangular "paint rectangles" overlaid on the page, indicating the areas that were repainted during the frame. A large white arrow points diagonally across the page, and a teal circle is positioned in the upper left quadrant. The "Composited Layer Borders" feature is enabled, showing thin green lines around some elements. The "Console" tab is also visible in the DevTools interface.

magna. Donec eu dapibus sapien. Morbi in ipsum vel massa hendrerit porttitor. Morbi leo magna, faucibus sed vulputate lacinia, adipiscing a massa. Mauris vitae sem neque, at volutpat nisi. Nullam turpis est, porttitor et sagittis ultricies, volutpat sit amet velit. Donec euismod, tellus sit amet commodo molestie, nunc turpis tincidunt tortor, varius consequat massa eros sed ipsum. Morbi eleifend risus sed mauris elementum in ultrices tortor fermentum. Nunc auctor, mi nec tempor facilisis, nunc orci egestas elit, eu aliquet purus eros vitae libero. Aliquam turpis lectus, hendrerit id lobortis at, faucibus eget dui. Praesent ornare massa placerat nulla vestibulum

Suspendisse vitae tu ornare. Proin tempu condimentum. Maur vulputate egestas. P sodales facilisis, lac bibendum tortor ante interdum posuere fe pharetra. Nullam nisi pellentesque ac, her pellentesque est sit venenatis felis hend consequat.

30 fps

60 fps

200 ms 400 ms 600 ms 800 ms 1000 ms

Summary Paint Profiler

Range 0 --

Aggregated Time

-

- Idle

Console Search Emulation **Rendering**

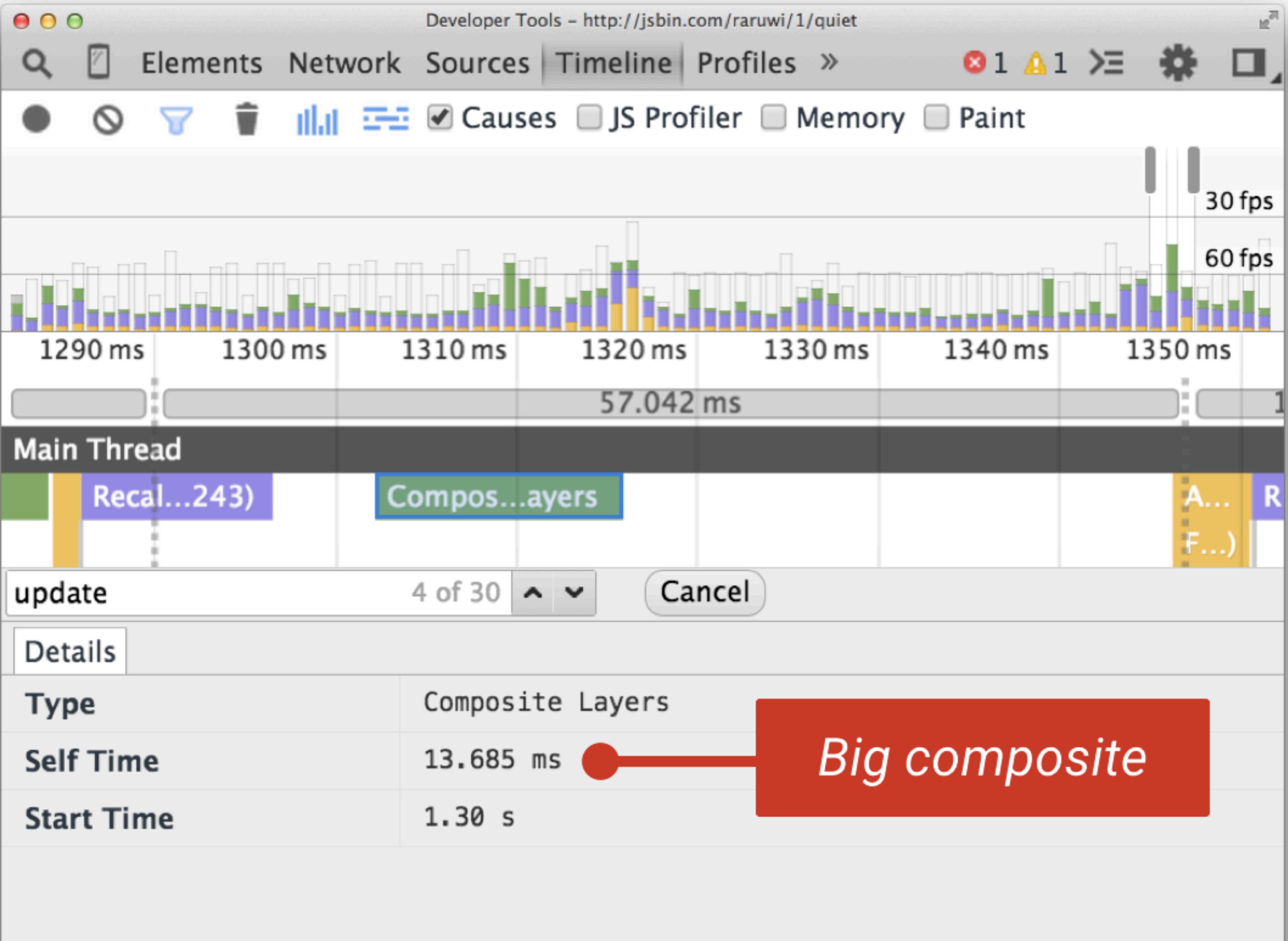
Show paint rectangles

Show composited layer borders

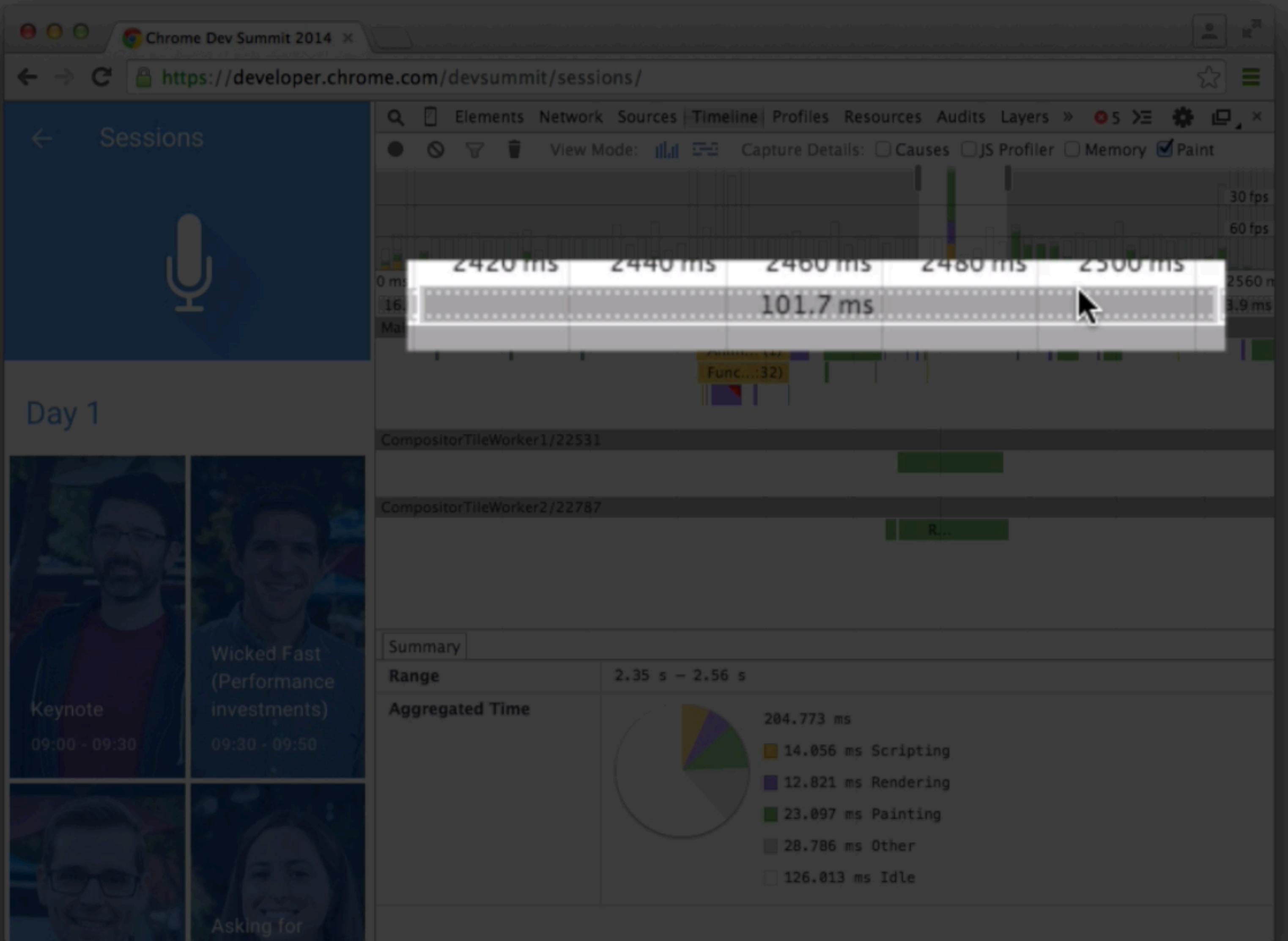
Show FPS meter

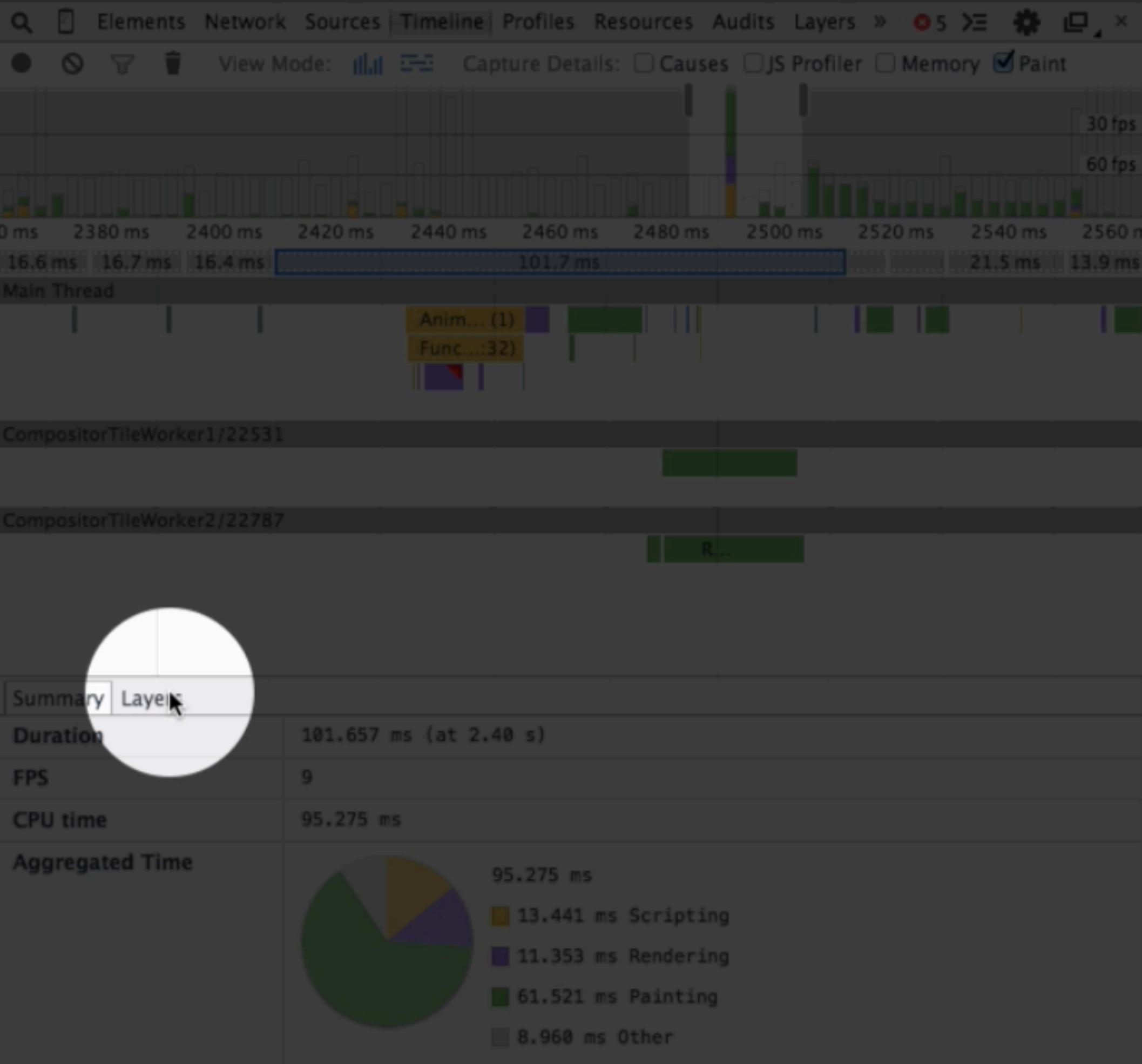
Enable continuous page repainting

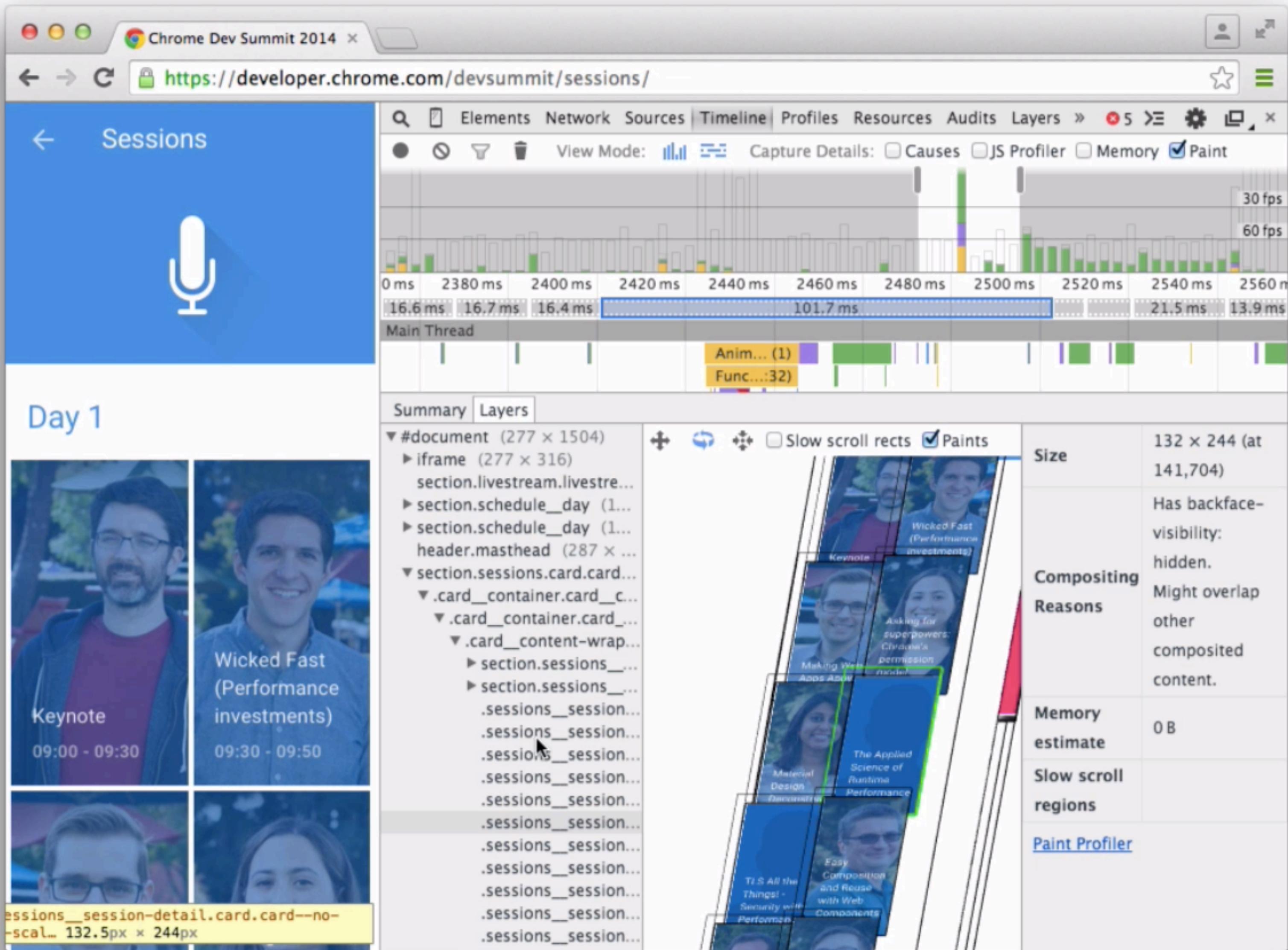
Show potential scroll bottlenecks

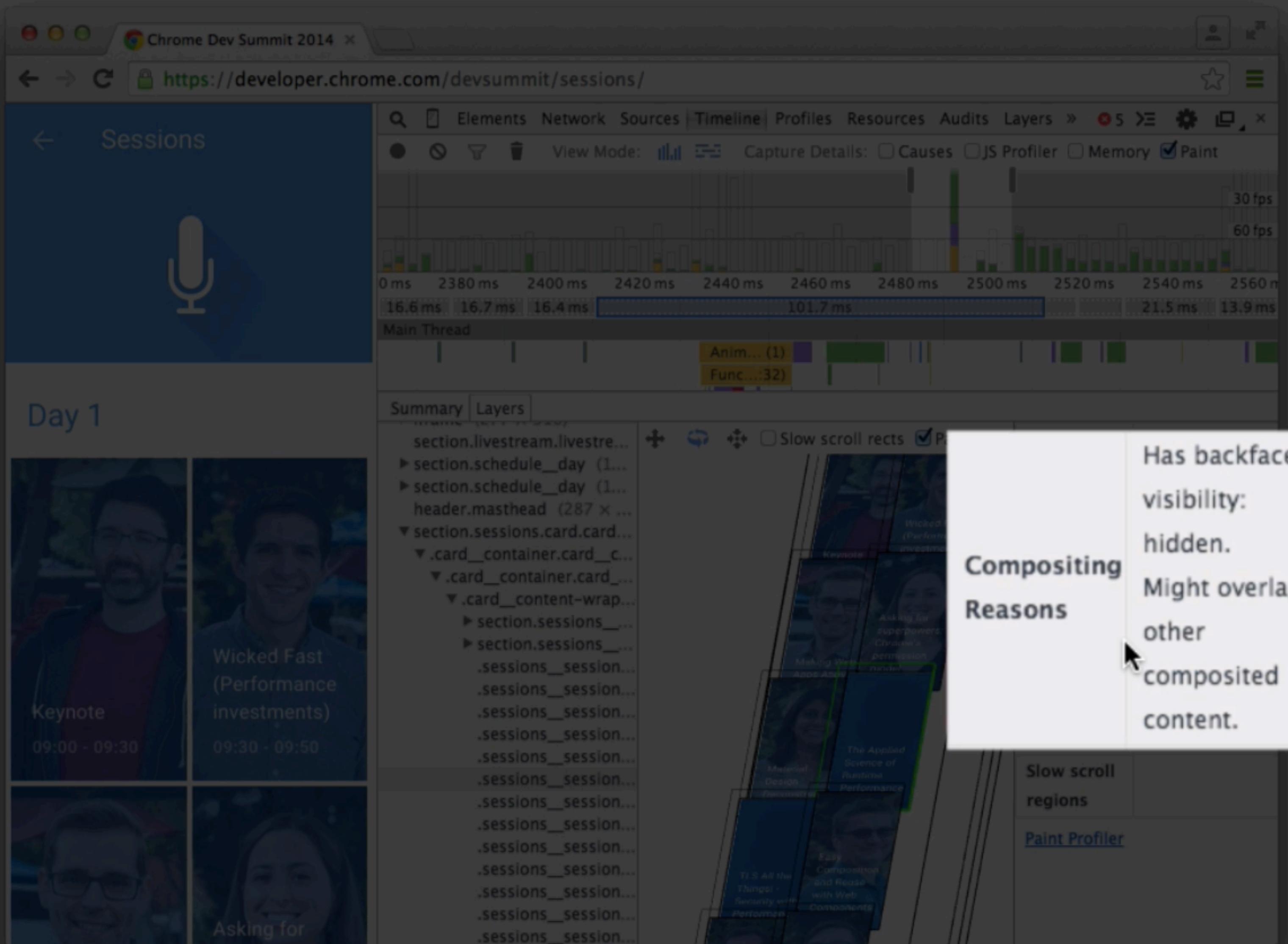


We can do better...







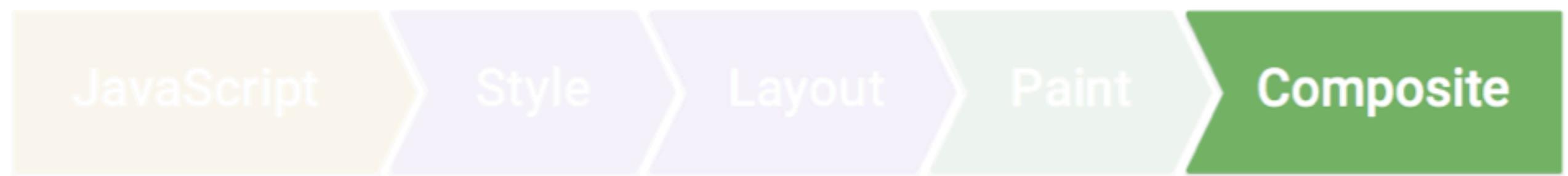


Bonus Round:
Touch and Input.

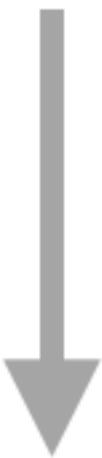
User touch Commit to GPU



Main Thread



User touch



Commit to GPU



Compositor Thread

touchmove



Main Thread

onTouchMove



← 50ms →

User touch

Commit to GPU

Scrolling blocked!

Compositor Thread

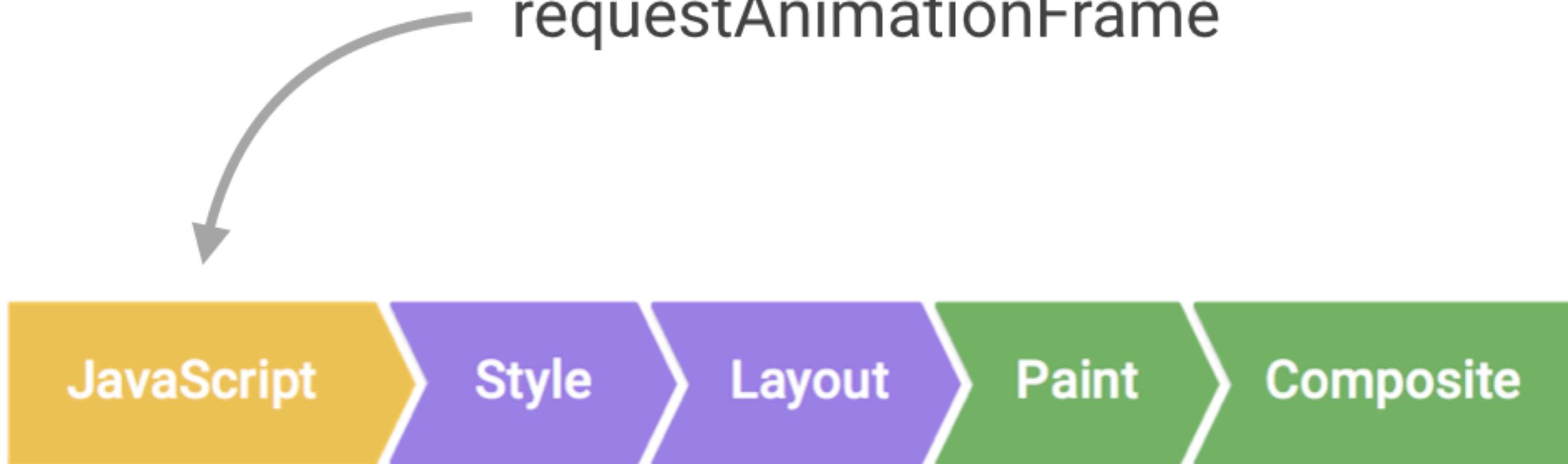
touchmove

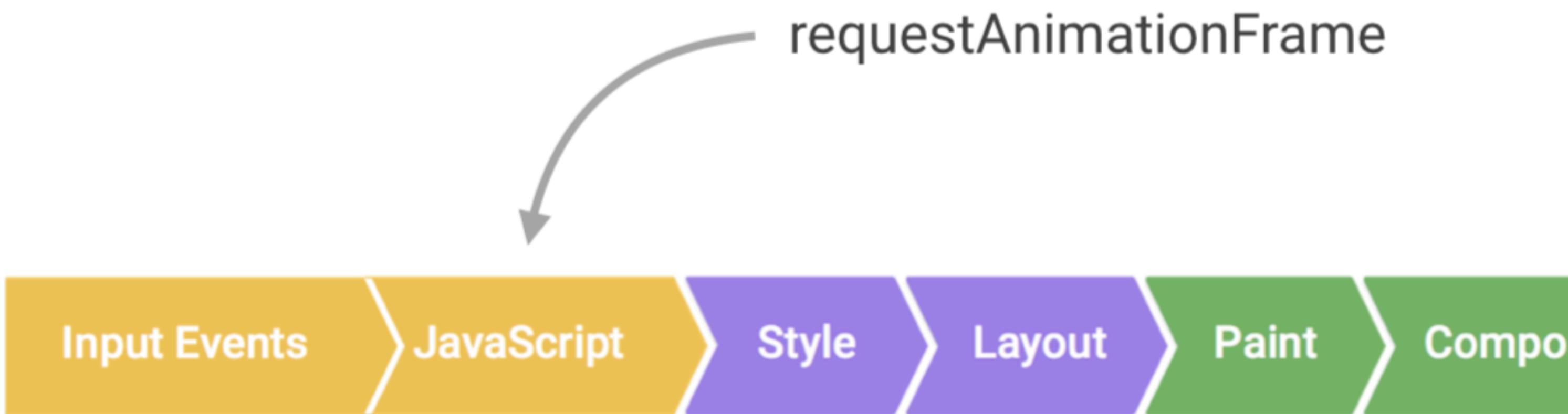
Main Thread

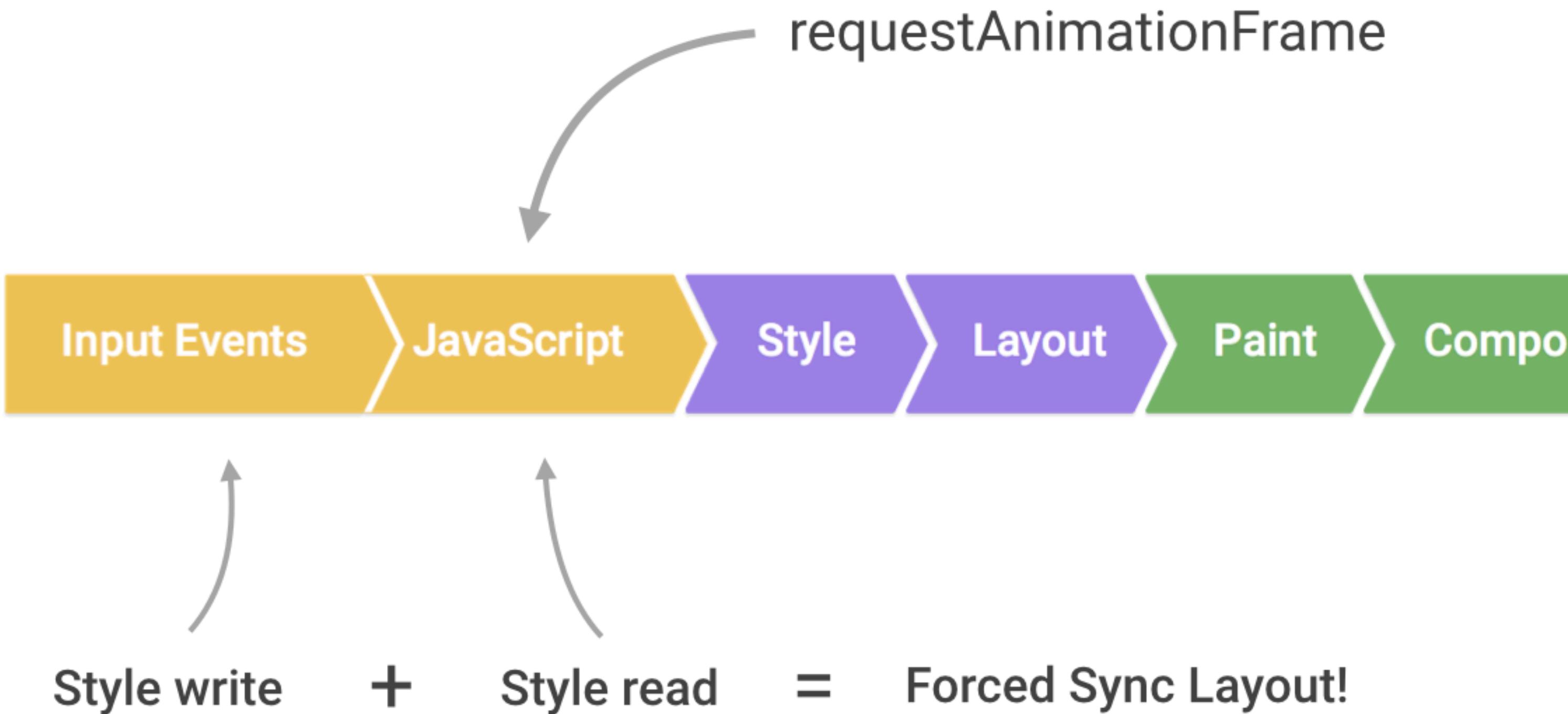
onTouchMove

← → 50ms

requestAnimationFrame









```
function onScroll (evt) {  
    // Store the scroll value for laterz.  
    lastScrollY = window.scrollY;  
  
    // Prevent multiple rAF callbacks.  
    if (scheduledAnimationFrame)  
        return;  
  
    scheduledAnimationFrame = true;  
    requestAnimationFrame(readAndUpdatePage);  
}  
  
window.addEventListener('scroll', onScroll);
```

Web Perf Next.

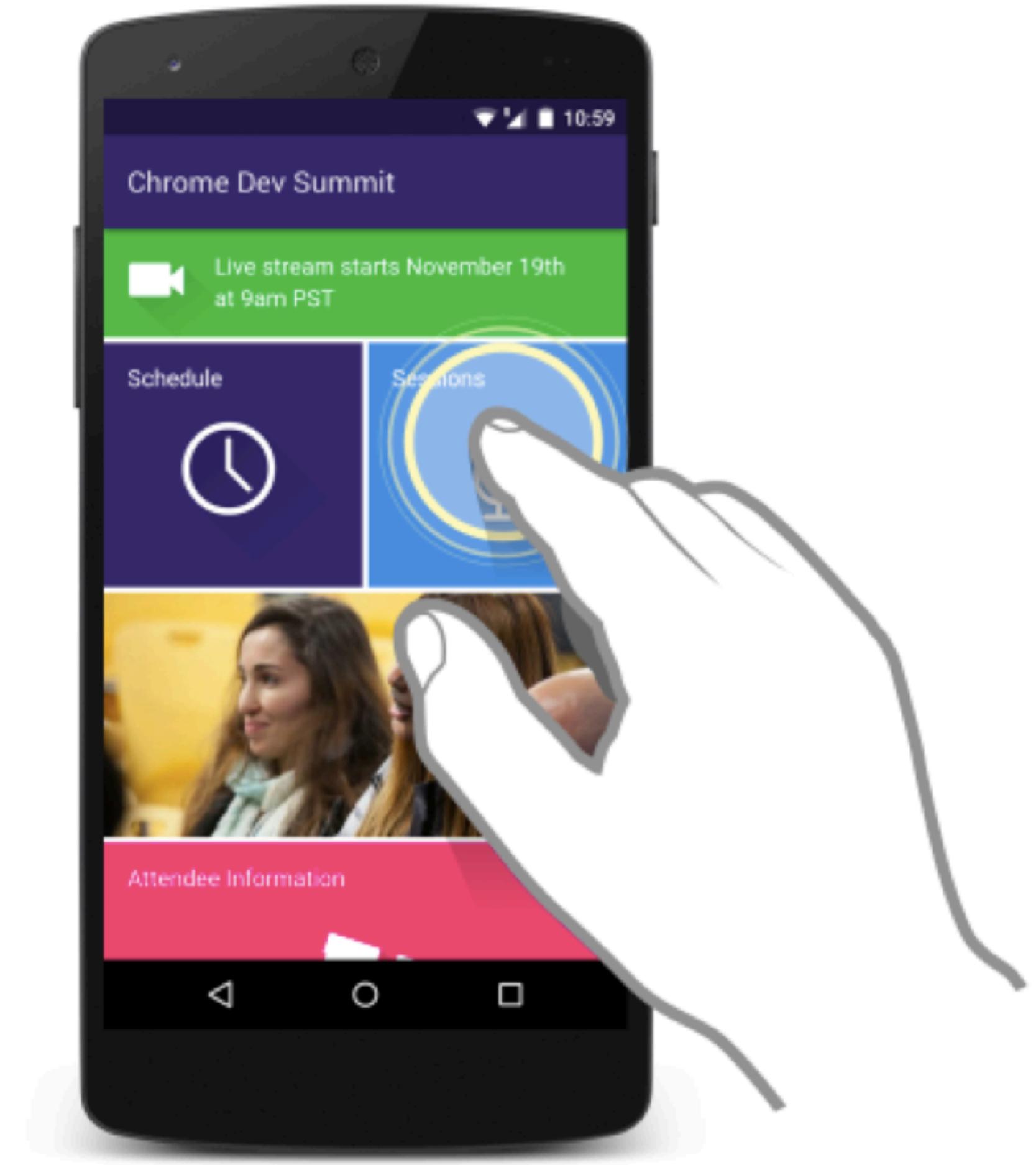
RAIL

Response

Animation

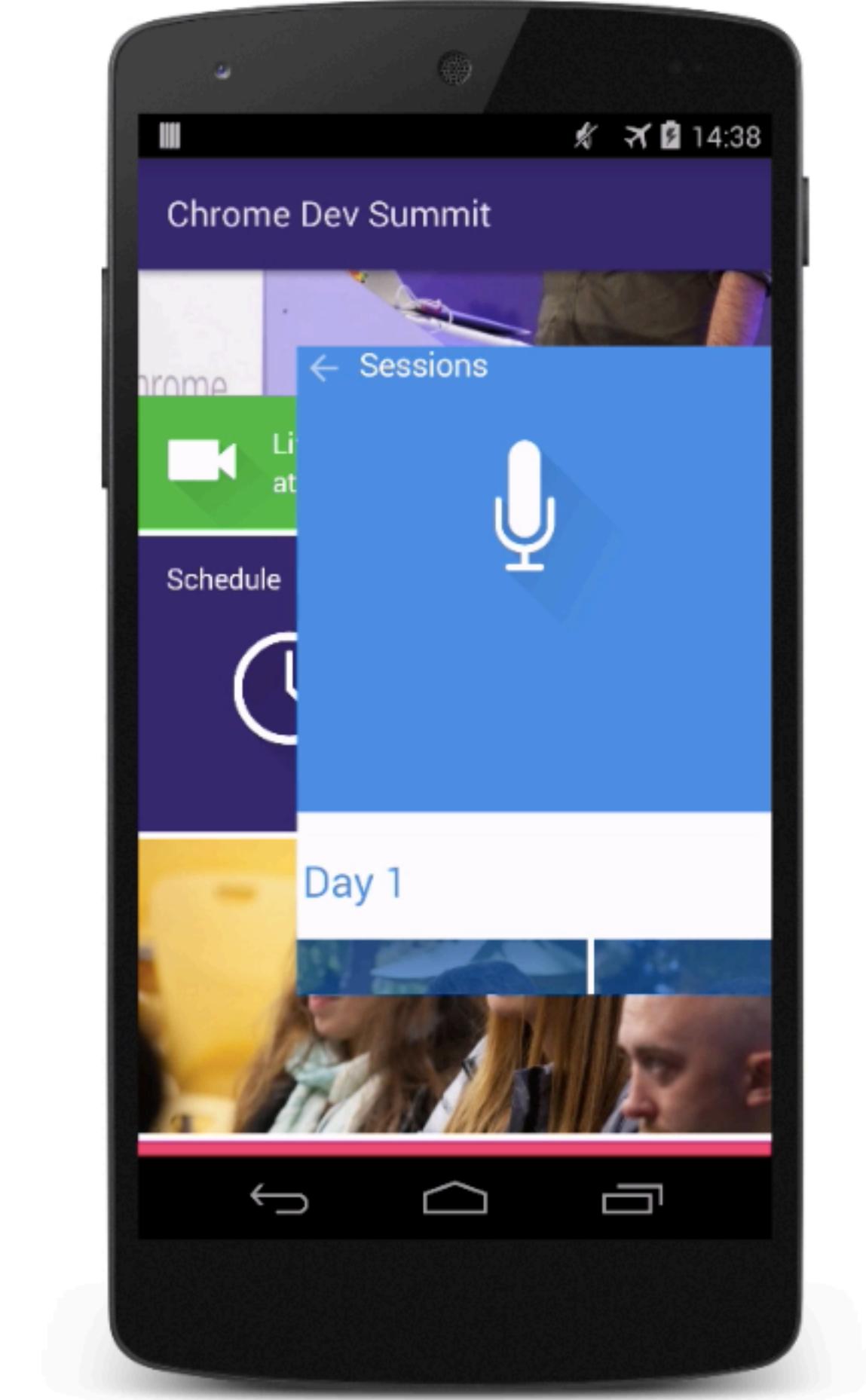
Idle

Load



RESPONSE

100ms



ANIMATION

16ms

ANIMATION

10ms



IDLE

50ms



LOAD

1000ms

Response

100ms

Animation

10ms

Idle

50ms

Load

1000ms

Pipeline, meet RAIL.

| | Response | Animation | Idle | Load |
|--------------------------|----------|-----------|-------------|--------|
| Threshold | 100ms | 10ms | 50ms chunks | 1000ms |
| Asset load / parse | Avoid | Avoid | Unknown | 400ms |
| JS: Parse | Avoid | Avoid | Unknown | 30ms |
| JS: Execute | 15ms | 3ms | Unknown | 60ms |
| JS: GC | Avoid | Avoid | Unknown | 20ms |
| Blink: Style Calcs | 10ms | 1ms | Unknown | 25ms |
| Blink: Layout | 15ms | 3ms | Unknown | 90ms |
| Blink: Layer Management | 10ms | 2ms | Unknown | 10ms |
| Blink: Paint | 5ms | Avoid | Unknown | 20ms |
| Compositor: Rasterize | 30ms | Avoid | Unknown | 100ms |
| Compositor: Image Decode | Avoid | Avoid | Unknown | 180ms |
| Compositor: Image Resize | Avoid | Avoid | Unknown | 55ms |
| Composite | 10ms | 2ms | Unknown | 10ms |

Closing thoughts.

-  **Performance is more than page load.**
-  **Service Worker changes everything.**
-  **Tools, not rules.**
-  **Profile, then optimize.**
-  **Prioritize the user's needs. Use RAIL.**

Performance matters.

Thank you!