Debugging animation performance

Renato Ruk - front end dev @degordian

Performance is a feature

Bad performance on high end devices is a bug

UI thread

- single thread for JavaScript execution and user interface updates
- web workers
- fast JavaScript is the prerequisite for performance

Frame budget

- 60 frames per second
- 1000ms / 60 = ~16ms per frame
- due to browser overhead, all work needs to be done in 10-12ms to avoid jank

The pixel pipeline

1. JavaScript / CSS change
2. Style recalculation
3. Layout
4. Paint
5. Compositing

JS / CSS change

- changing the DOM structure (appending elements, adding or modifying attributes)
- moving an element with an animation library
- CSS animation or transition

Style recalculation

- process of determining CSS property values after the DOM changed or transition or animation changed a style value
- figuring out which CSS rules apply to which elements based on matching selectors

```
<html>
<head>
    <style>
        .header {
            position: relative;
        }

        .header--fixed {
            position: fixed;
        }
    </style>
</head>
<body>


<div class="wrapper">
    <header class="header">Header content</header>
</div>

<script>

    const header = document.querySelector('.header');
    header.classList.add('header--fixed');

</script>
</body>
</html>
```

Style recalculation

Layout

- calculating how much space does an element take on the screen and what is its position
- one element can affect others

Layout

- content
- padding
- border
- margin
- top, right, bottom, left

Paint

- process of determining which color to paint each pixel
- drawing is often done into multiple smaller rectangles called layers
- triggered by any property that isn’t transform or opacity

Paint

- background-color
- border-style
- box-shadow

Compositing

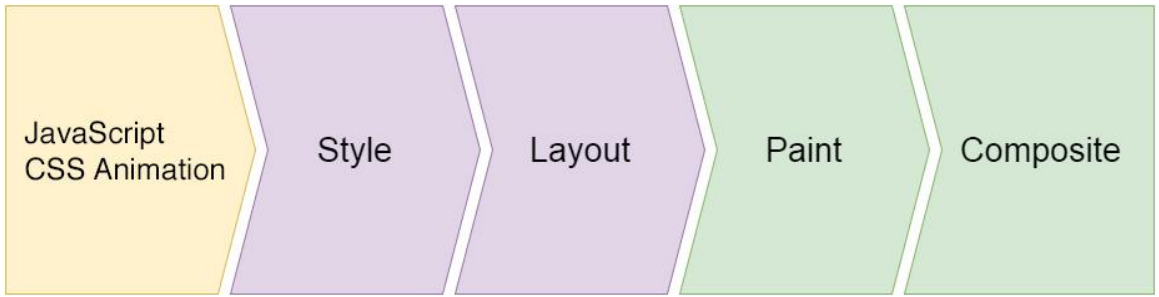
- layered bitmaps are uploaded to the GPU and combined
- overlapping layers - need to be drawn in the correct order to avoid appearing on top of another incorrectly

Compositing

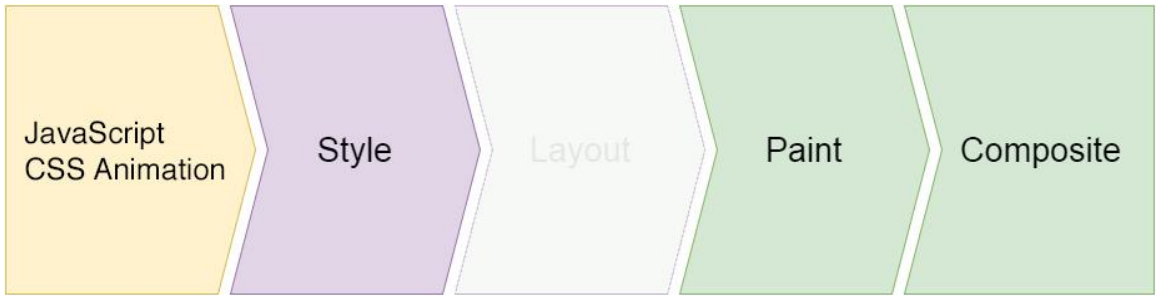
- transform
- opacity

Pipeline flows

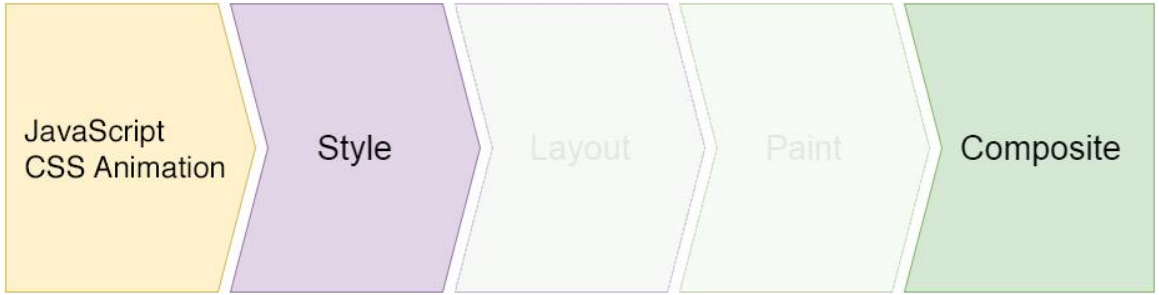
- complete cycle
- no layout
- **no layout and no paint - cheapest**



Complete cycle



No layout



No layout and paint

Promoting elements

- manually move elements that you plan to animate to a separate compositor layer
- will-change or translateZ (translate3d)
- don’t overdo it as it requires additional VRAM and management

Recap

- keep JavaScript execution fast
- avoid animating expensive properties
- promote elements which you plan to animate

Additional links

- The Layer Model
- High Performance Animations
- Simplifying Paint Complexity
- CSS Triggers

Thank you