GOSIM

GOSIM HANGZHOU 2025

A Dive Into the servo Layout System

Oriol Brufau / Martin Robinson

Servo @ Igalia

What is a browser engine?

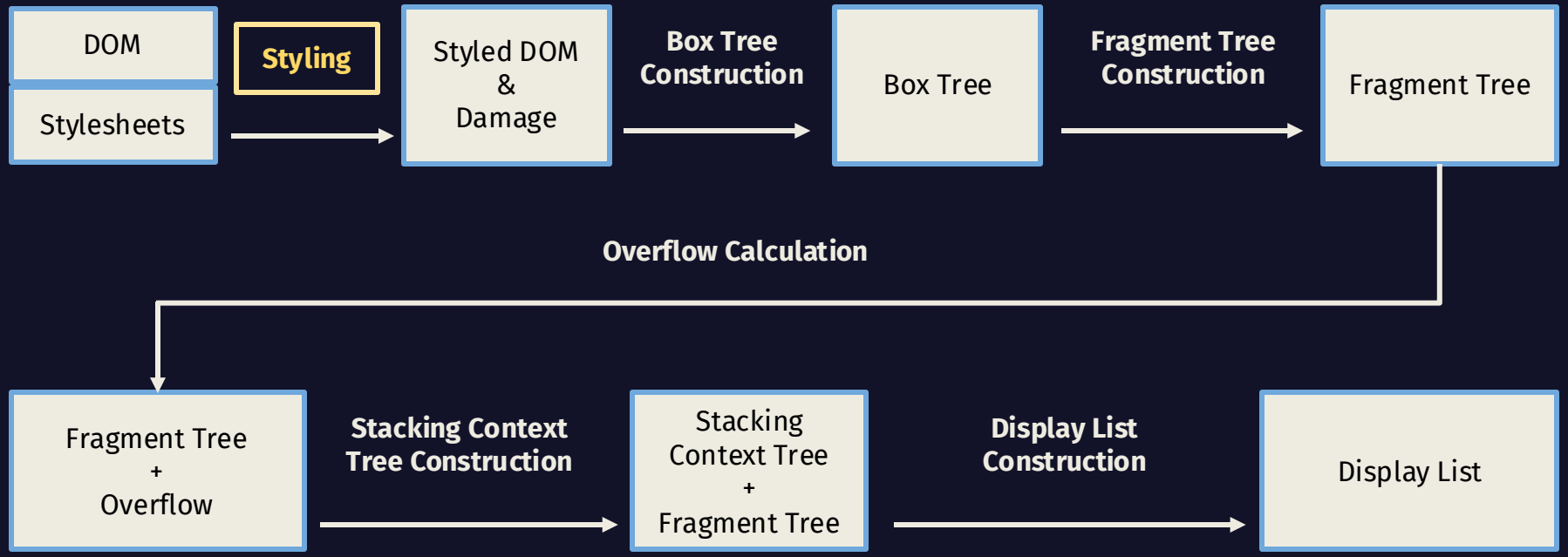
Servo

- New web browser engine
- Written in Rust: memory safety & concurrency
- JavaScript engine: SpiderMonkey
- Multi-process and single-process modes
- Layout done in parallel, but DOM is mainly synchronous

Layout

- Combine CSS & DOM into different layout trees
- Necessary to render page contents
- Expensive
  - Parallelize when possible
  - Preserve work within and between layouts
- Some DOM operations force layout
  - `Element.clientLeft`
  - `Element.scrollTo()`
  - `Window.getComputedStyle()`

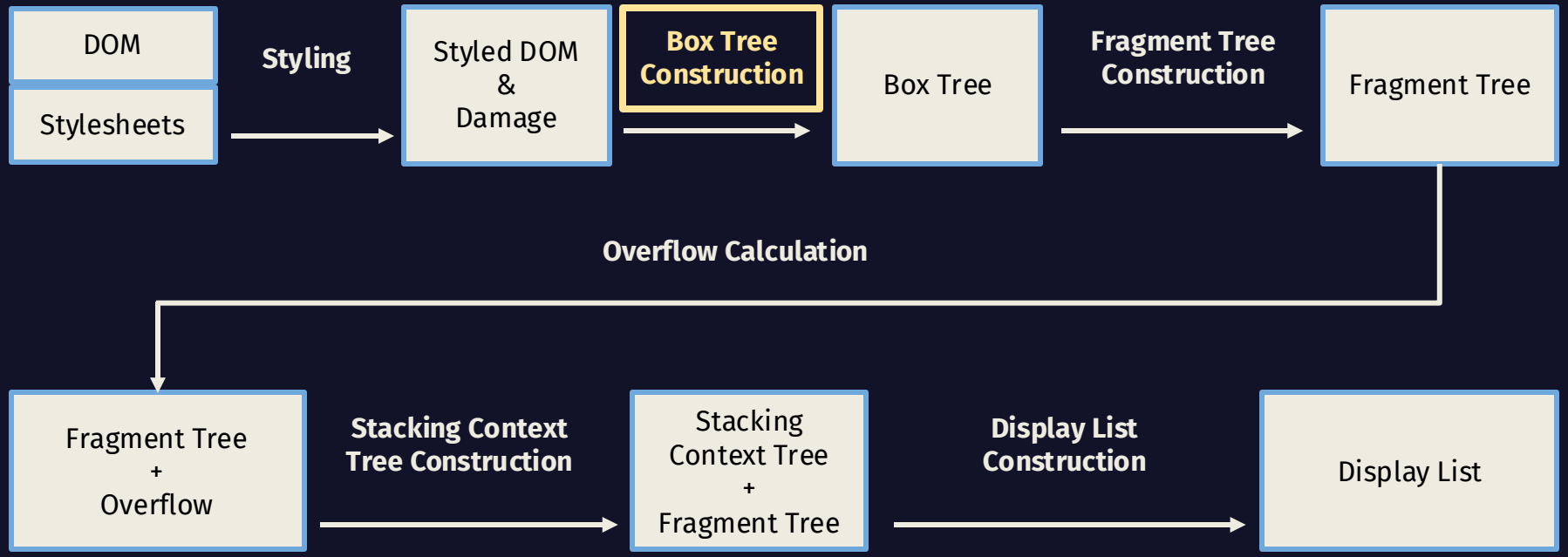
Layout Pipeline



Styling

- Parsing done ahead-of-time before layout
- Computes the value of CSS properties on each element
- Selector matching and CSS cascade
  - Specificity, layers, origins, `!important`
- Style sharing
- Changes to a node's style apply *damage* to nodes

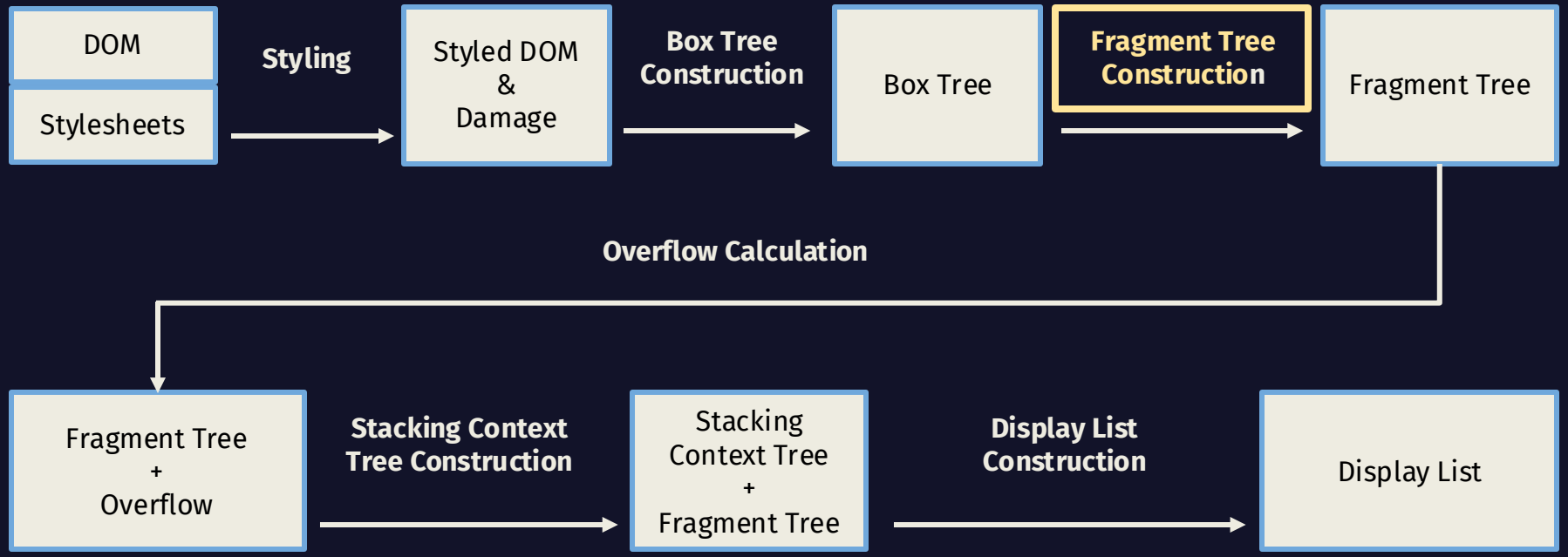
Layout Pipeline



Box Tree Construction

- **Box Tree:** High-level representation of layout
- **Boxes:**
  - Formatting contexts (`display` mode)
  - Formatting context contents
    - flex-level item, Inline-level item, etc
    - Items can contain other formatting contexts
  - Others: absolutes, floats (flow only), list item markers
- Traverse DOM and combine with style
- Style can produce pseudo-elements / anonymous boxes

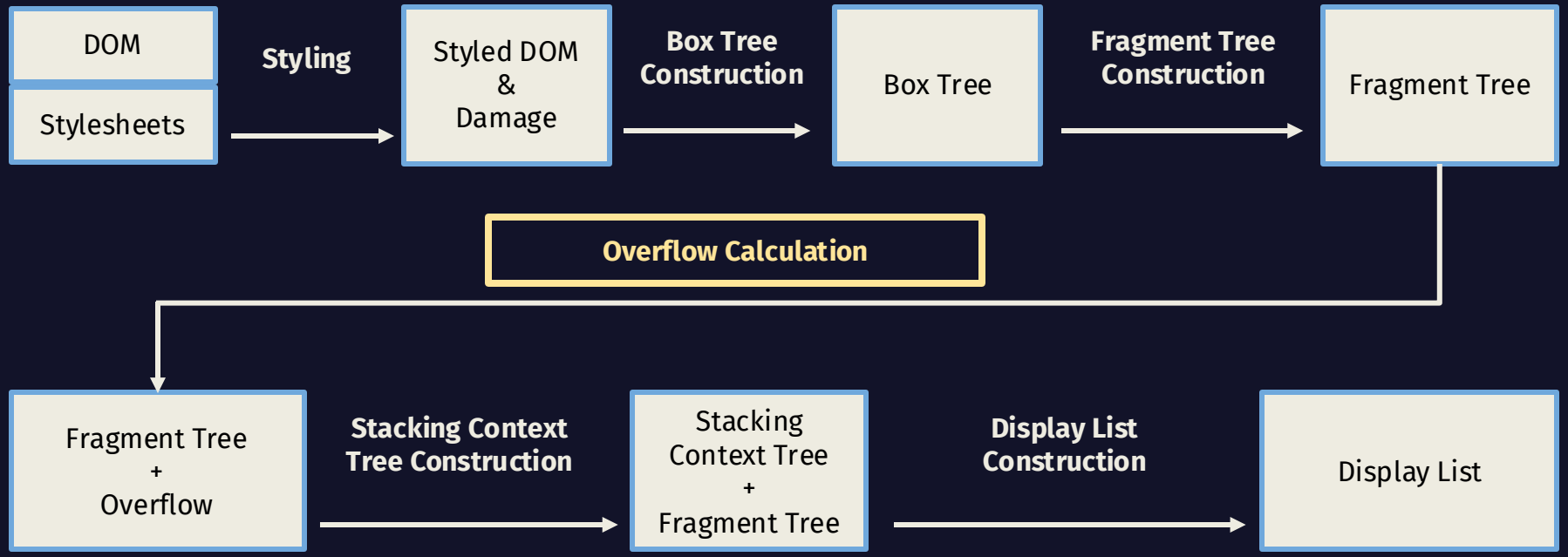
Layout Pipeline



Fragment Tree Construction

- **Fragment Tree:** Tree of size and position of box tree nodes, split into fragments
- **Fragments:**
  - Boxes can have multiple fragments
  - Includes style, rectangle, position, margin, border, and padding
  - Relative to containing block
- Walk the box tree and lay out boxes
- Absolutes move up the tree (hoist) to their containing blocks

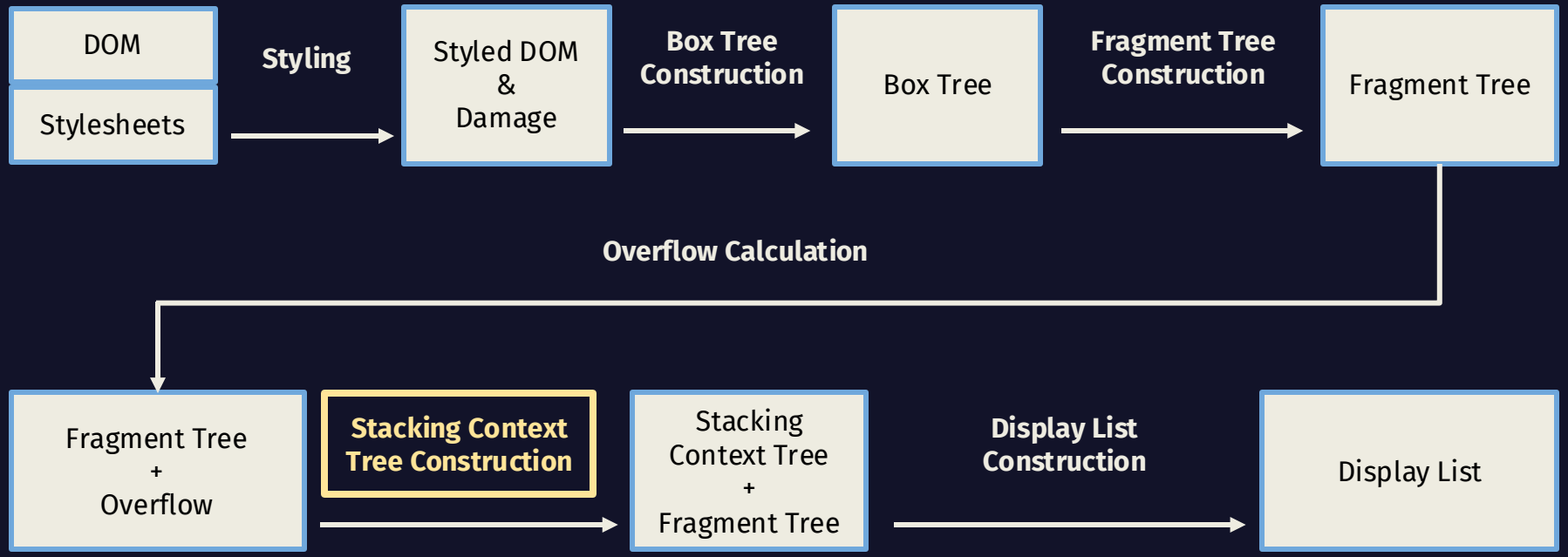
Layout Pipeline



Overflow Calculation

- **Overflow**: When contents of a fragment extend beyond boundaries of parent fragment
- Calculated by accumulating overflow of each fragment in parent
- Minor layout phase
- Determines size of scroll containers and position of sticky nodes

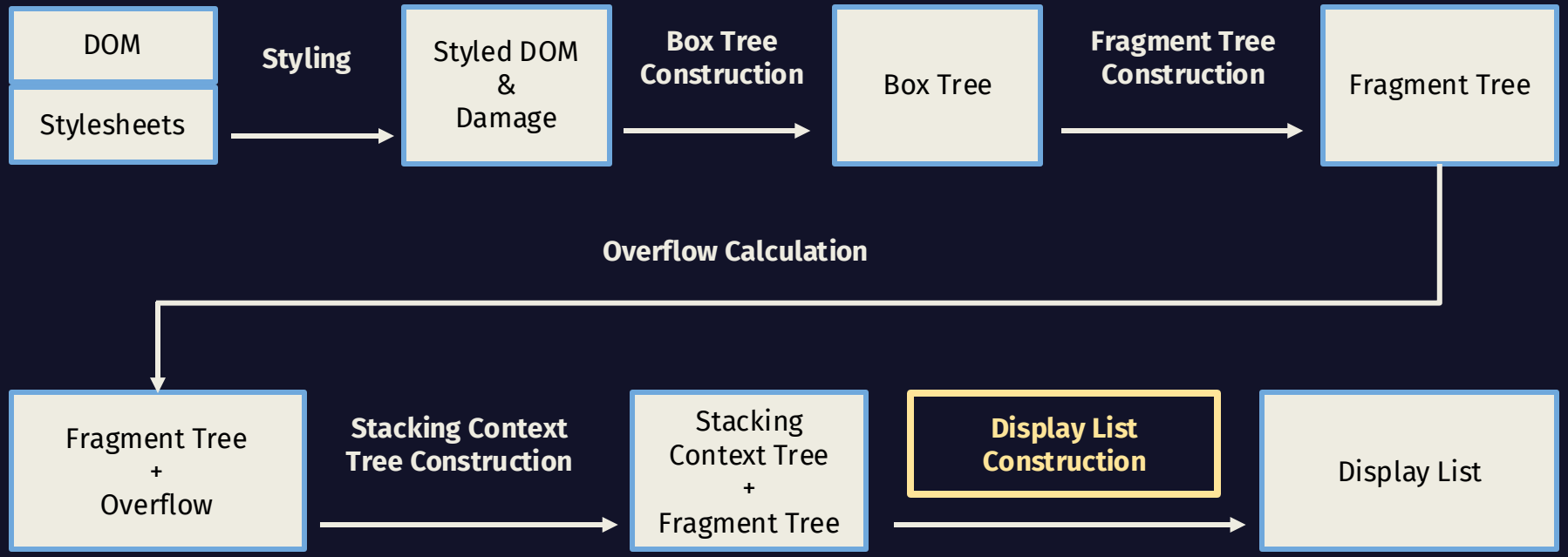
Layout Pipeline



Stacking Context Tree Construction

- **Stacking Context Tree:** Transformation of the fragment tree that splits fragments into layers and order them for painting.
- Walk the fragment tree:
  - Split fragments into their layers
  - Sort fragment pieces according to specification
  - Determine final scroll container dimensions and transforms
  - Painting order determined by style
- Fragment layers: backgrounds, borders, content

Layout Pipeline



Display List Construction

- **Display List:** List of CSS display items
  - Borders, rectangles, text, drop shadows
  - A tree of clips
  - A tree of transforms and scroll nodes (spatial tree)
- Walk fragment tree in stacking context tree order:
  - Create display items for each fragment layer
  - Based on style / size & position of Fragment
  - Prepare all images, video frames, clips, etc.
- Display list is serialized and sent to the renderer for rasterization

Rebuild only the damaged part of the tree

Changing Display Mode

```
<div>Untouched</div>
<div id="container" style="display: block">
    <div>inner item</div>
</div>



<script>
    container.style.display = "flex";
</script>
```

Skip *unnecessary* layout phases

Background Color

```
<div id=”container” style=”background: pink”>
    Really complicated layout
</div>



<script>
    container.style.background = "blue";
</script>
```

Transformation

```
<div
    id=”container”
    style=”transform: translateX(10px);”
>
    Really complicated layout
</div>



<script>
    container.style.transform = "translateX(20px)";
</script>
```

Transformation

```
<div id=”container”>
    Really complicated layout
</div>



<script>
    container.style.transform = "translateX(20px)";
</script>
```

Changing z-index

```
<div id=”container”
     style=”position: relative; z-index: -10;”>
</div>



<script>
    container.style.zIndex = "10";
</script>
```

Can we do

less work?

Skip layout entirely

Animated Images

Contact

- Try an experimental build of Servo
- ~ Mastodon: https://floss.social/@servo
- ~ GitHub: https://github.com/servo
- ~ Chat: servo.zulipchat.com
- ~ Email: join@servo.org

Thank You