Container Queries

Rune Lillesveen

Google

2022-05-19

Introduction

@media queries

- Responsive design based on viewport size
- Example:

```
@media (width >= 600px) {
  #my-component { display: flex; }
}
```

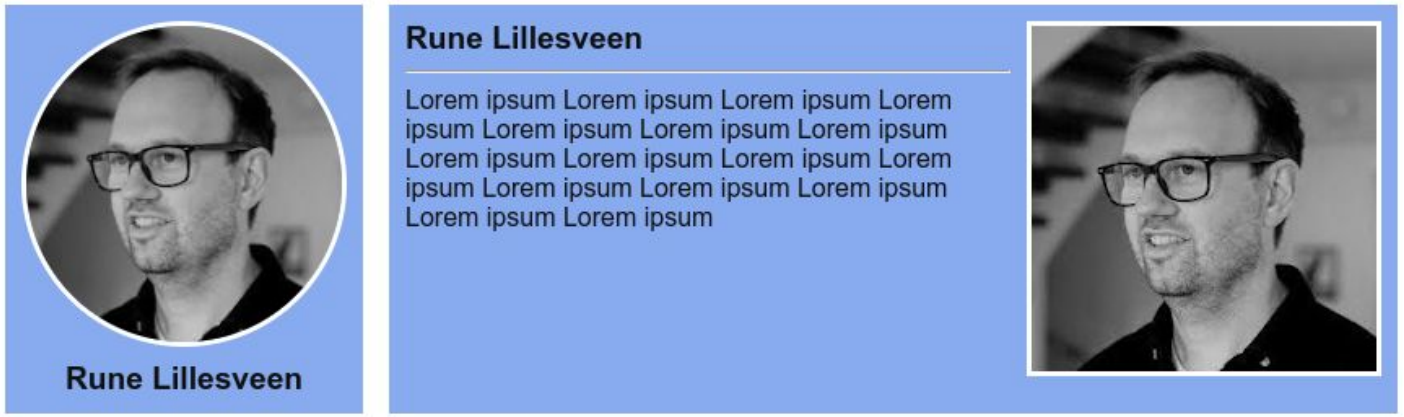
@container queries

- Highly requested feature from developers
- Responsive design based on component size
- Example:

```
@container card (width >= 400px) {
  .portrait { float: right; }
}
```

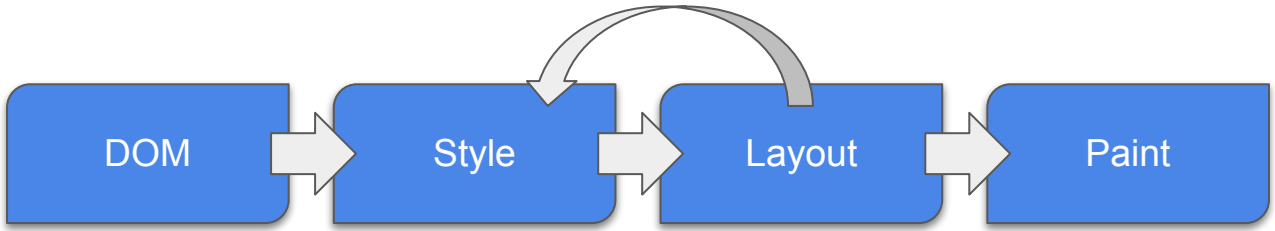
@container queries

- Example rendering



The rendering pipeline

Container queries introduce a loop in the pipeline



Specification

Containment - the inherent problem

```
<style>
  #container {
    container-type: inline-size;
    width: fit-content;
    font-size: 100px;
    border: 1px solid currentColor;
  }
  @container (width >= 100px) {
    #inner { font-size: 10px; }
  }
</style>
<div id="container">
  <div id="inner">TEXT</div>
</div>
```

Containment - the inherent problem

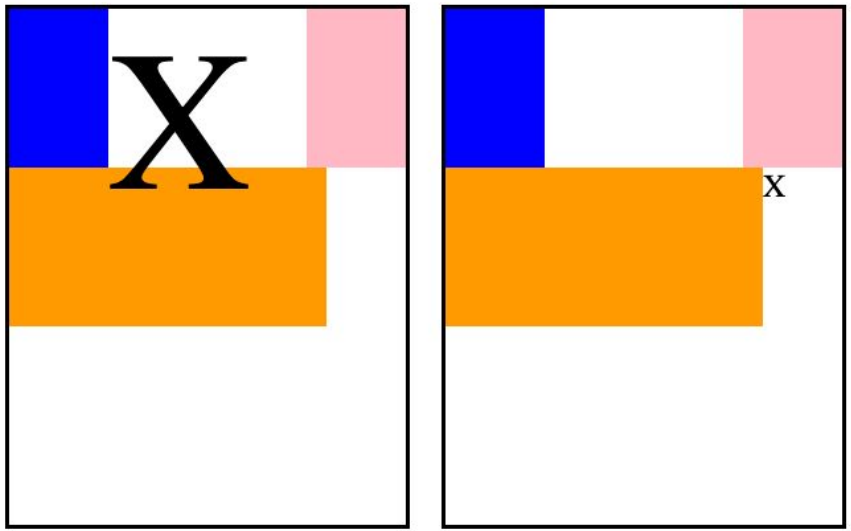
- Apply containment for container-type: size
  - Size containment
  - Layout containment
  - Style containment

Containment - inline-size

- Size containment too restrictive
- contain: inline-size
- container-type: inline-size
- Container with size containment and auto height overflowing container:

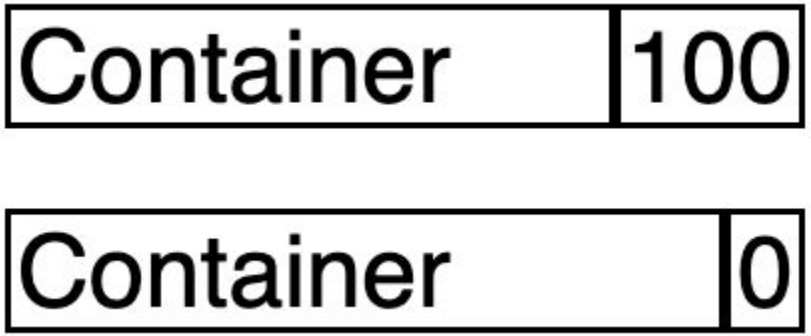
Inline size containment

- Always moving layout forwards
- Never return to previously attempted opportunities



Containment - style

- Effect of counter-increment contained to a subtree
- Counters affecting intrinsic size outside container for flex items:



Specification Summary

- Introducing size containment in one dimension
  - contain: inline-size
- Establishing a query container with a type and a name:
  - container-type: size / inline-size
  - container-name: <custom-ident>
- Apply size, layout, and style containment for size container types
- Query ancestor containers via @container rule:

```
@container my-component (width >= 600px) {
  .figure { float: right; }
}
```

Implementation

Interleaving Layout and Style

- ResizeObserver - full style and layout iterations
  - Need to limit iterations
  - Performance issues
- Approach
  - Skip style recalc of container subtrees and resume from layout when possible
  - Support for LayoutNG only
  - Mark ComputedStyles as depending on container queries
  - Invalidate style for container query changes during layout
  - Set up container as style recalc root

Implementation Challenges

- Change invariant contracts between style and layout
- Marking layout dirty during layout
- Allow style recalc during layout
- Allow box tree modifications during layout
- LayoutNG not finished - handle legacy trees

Implementation - Style

- Style recalc as usual until a size container is reached
- Skip recalc for size container subtrees which will be visited for subsequent layout
- Subtrees will be left style-dirty with state stored on the container for resuming recalc during layout
- Containers store a query evaluator that keeps track of evaluation changes
- Container descendants depending on size container queries tagged for invalidation when size queries change

Implementation - Layout

- Only works with LayoutNG
- Invoke style recalc from NGBlockNode::Layout
  - Due to query changes
  - Resuming skipped style recalc
- Stop-gap solutions for legacy engine fallback
  - Resume style recalc building legacy subtrees
  - Avoid skipping style recalc in legacy subtrees

Implementation - Layout

- Implemented inline-size containment

```
#component { contain: inline-size; }
```

- Improved auto scrollbar implementation
  - Non-overlay scrollbars
  - Containers inside scrollable containers have different size based on scrollbar presence
  - Improved predictability regardless of previous layout state

Implementation - Animations

- Computed style may have multiple updates due to multiple layout passes or not skipping style recalc
- Could lead to starting animations too early
- Moved the animations update to after layout

Implementation - getComputedStyle()

- Normally only depends on style recalc
- Resolved values like width depends on layout
- Size container queries cause style to depend on layout
- Introduced a LayoutUpgrade concept
- Upgrade for getComputedStyle() if ancestor chain has size query dependencies

Shipping and onwards

Shipping Plan

- Intent to Ship size queries for M105 sent - 3 LGTMs
- Also shipping container relative units
- Blocked on shipping LayoutNG Table fragmentation
- Will ship without printing support (uses legacy layout)
- Cross-engine support
  - WebKit implementation in progress
  - Gecko implementation in progress
- Polyfill work in progress

Onwards

- Support for style() container queries
  - Querying computed style on ancestor elements
  - Mixed signals from other vendors
  - Custom properties vs standard properties
  - OKR to implement for custom properties in Q2
- Element.matchContainer()
  - Similar to window.matchMedia() for media queries
  - Not specified yet

Questions?