

Find deterministic sizing rules for free-form documents

TPAC2025 Breakout session

<https://github.com/w3c/tpac2025-breakouts/issues/58>

<https://github.com/w3c/webextensions/issues/692>

Presentation: <https://apps.jeurissen.co/tpac2025-free-form.pdf>

Code of Ethics and Professional Conduct <https://www.w3.org/Consortium/cepc/>

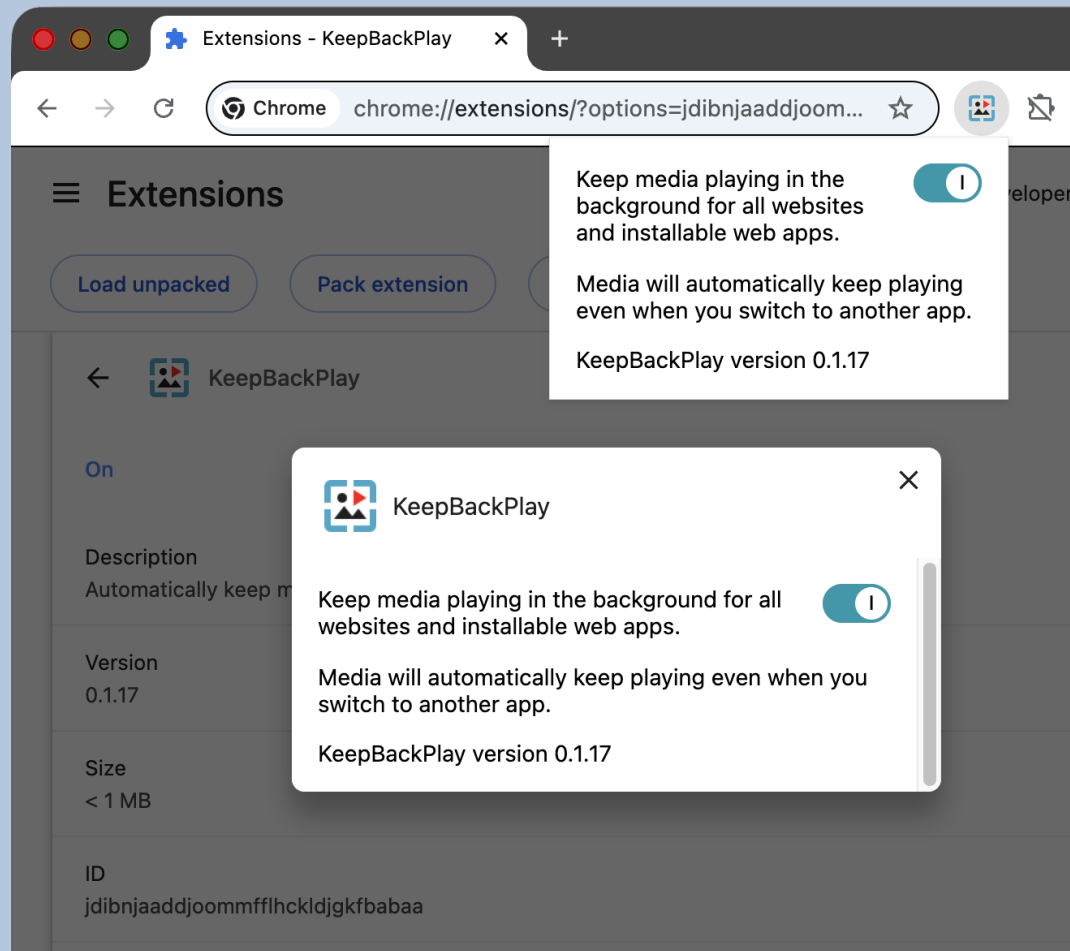
Antitrust and Competition Guidance <https://www.w3.org/Consortium/Legal/2017/antitrust-guidance>

Welcome

- Goals:
 - Elaborate on web extensions
 - Make progress on auto-sizing frames
 - Determine approach and next steps

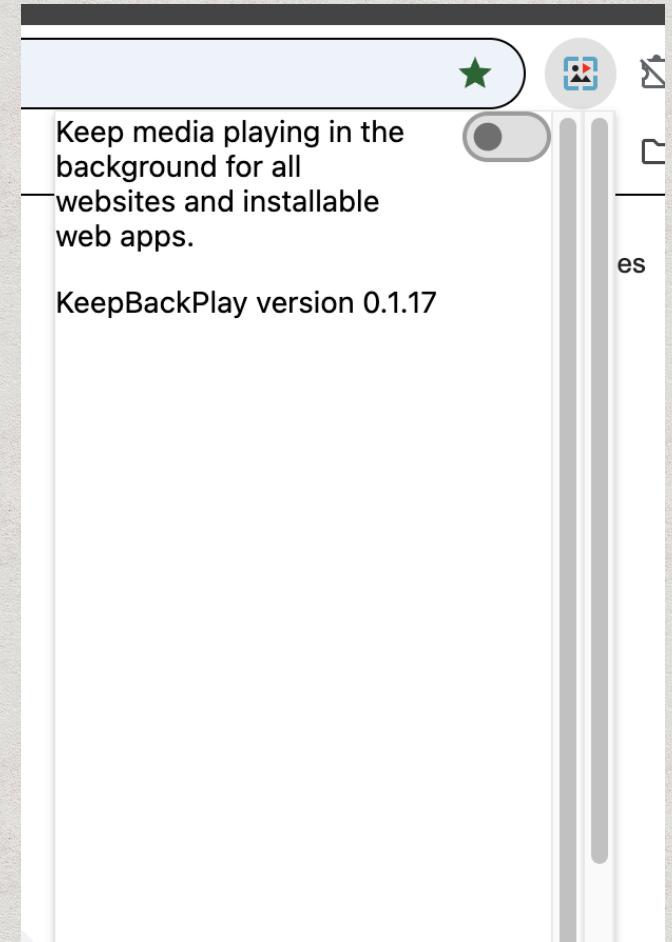
Current web extension model

- Guess developer size intentions based on html/body size and/or content
- Similar to proposed Contain-intrinsic-size: from-element



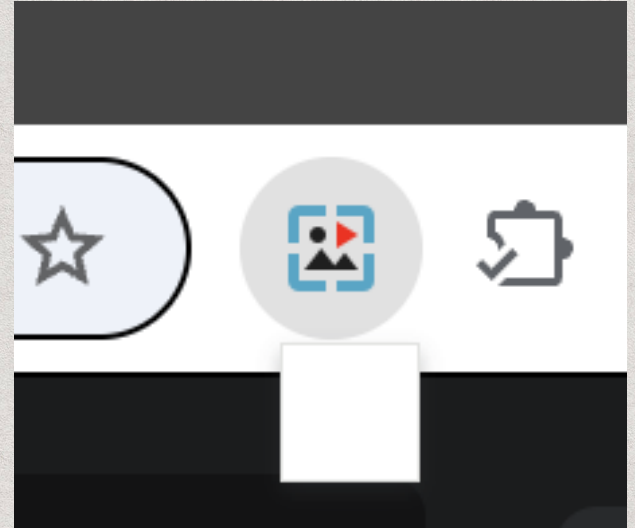
Issues with extension model (1)

- Double scrollbar if taller than browser limit



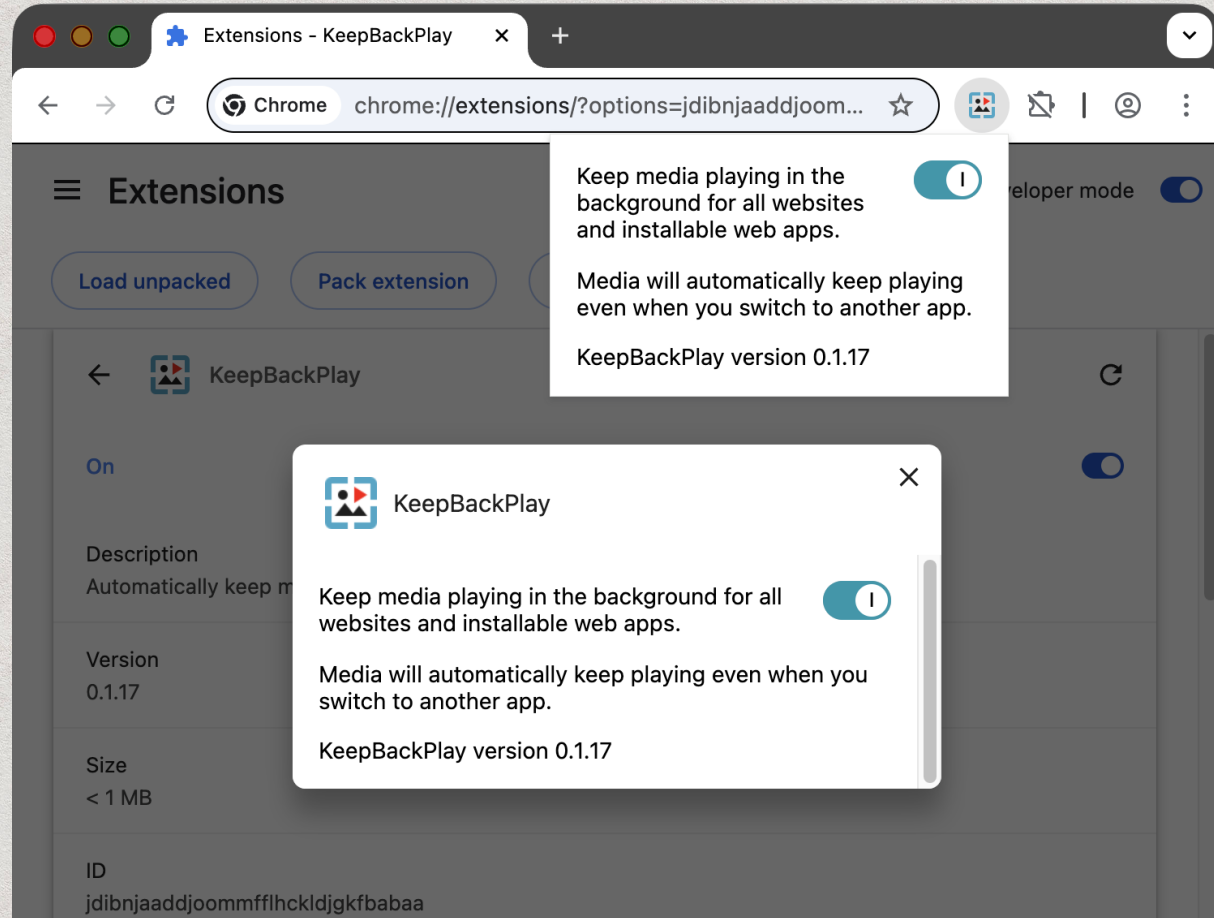
Issues with extension model (2)

- 100% size with dynamic loading



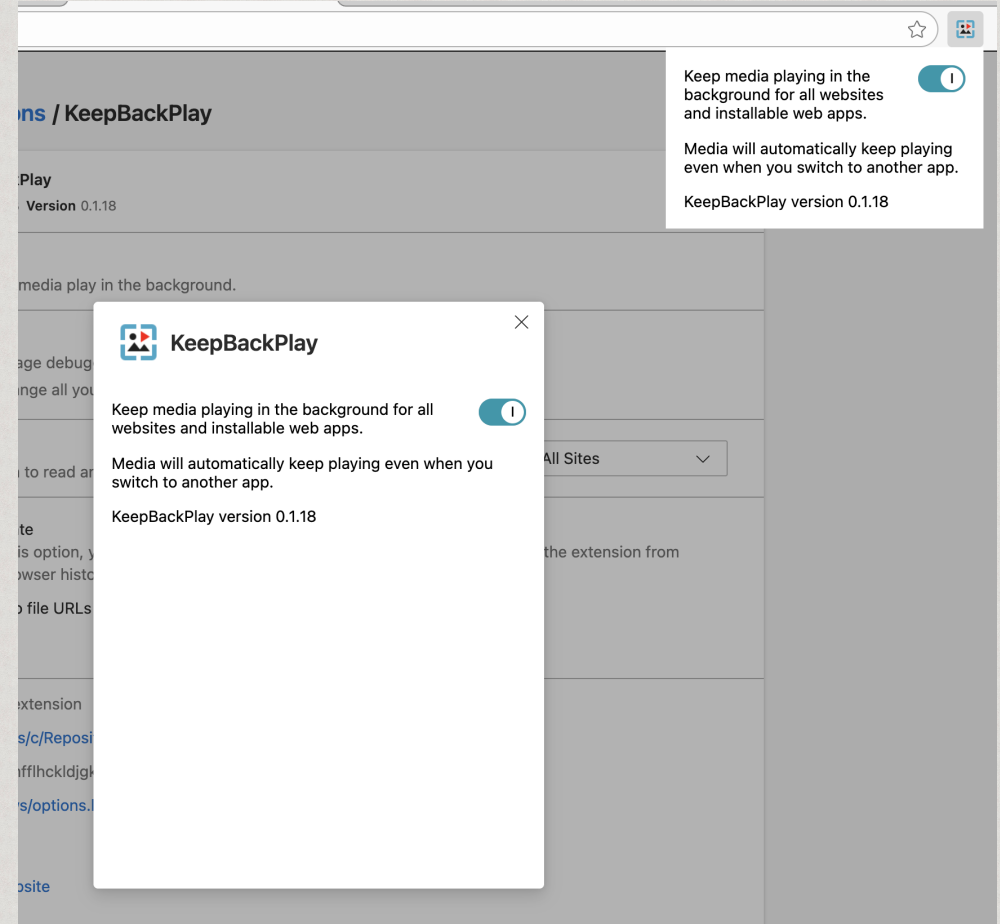
Issues with extension model (3)

- Browser inconsistencies: Chrome



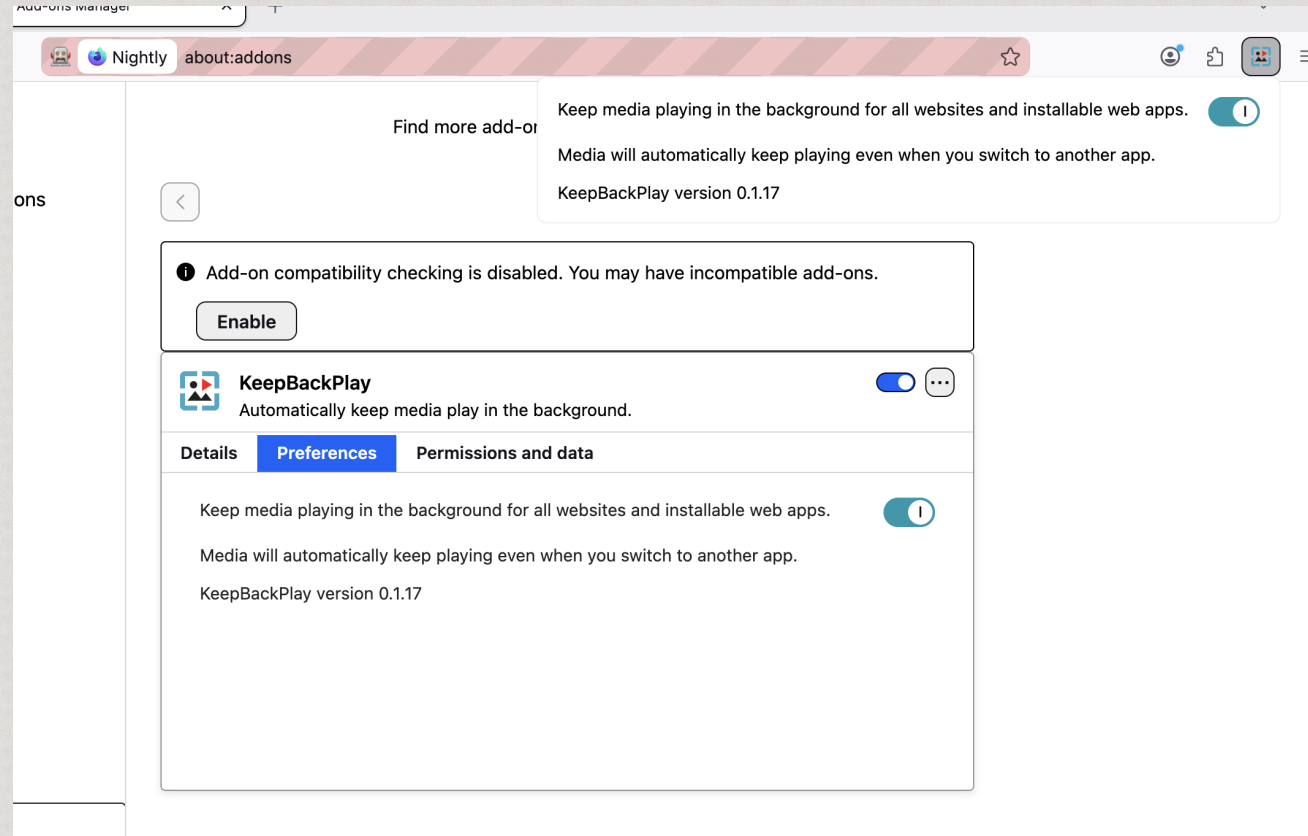
Issues with extension model (3)

- Browser inconsistencies:
Edge



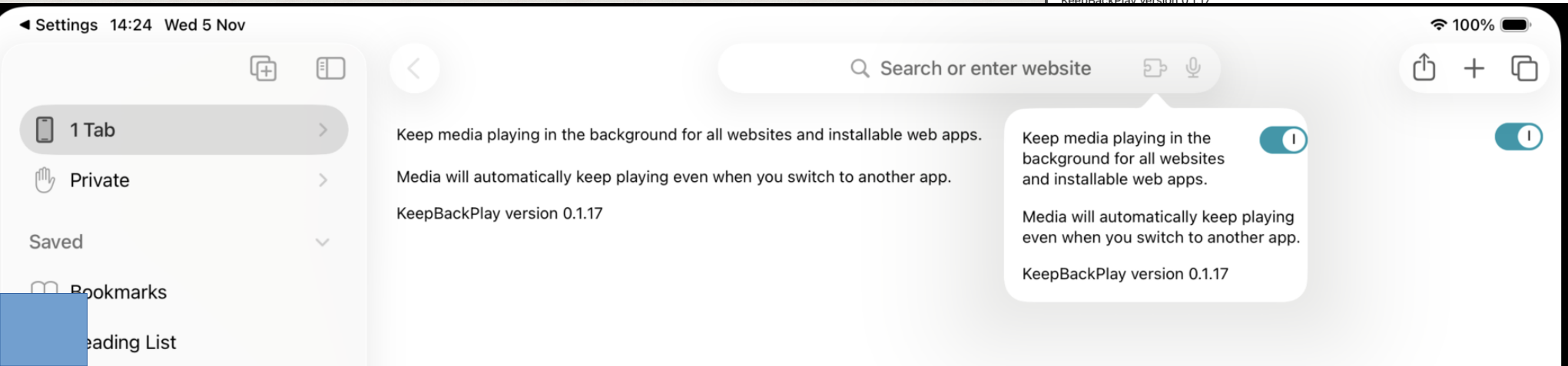
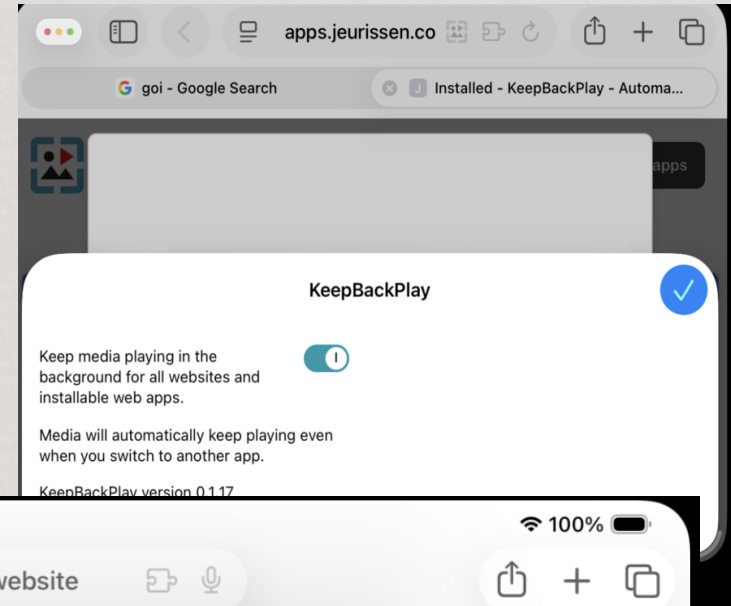
Issues with extension model (3)

- Browser inconsistencies: Firefox



Issues with extension model (3)

- Browser inconsistencies:
Safari



Issues with extension model (4)

- Hacks to work both as popup and full page

```
html,  
body {  
  width: 600px;  
  height: 596px;  
}
```

```
@media (width >= 617px), (height >= 601px) {  
  html,  
  body {  
    min-width: 100%;  
    max-width: 100%;  
    height: 100%;  
    min-height: 596px;  
  }  
}
```

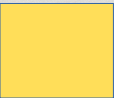
```
@media (width <= 599px), (height <= 595px), (width >= 617px), (height >= 601px) {  
  html.loaded,  
  html.loaded > body {  
    min-width: 100%;  
    max-width: 100%;  
    height: 100%;  
    min-height: 100%;  
  }  
}
```

New challenges

1. Host is king
2. Lifetime changes
3. Backwards compatible
4. Sizing before load
5. One dimension flexibility

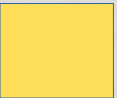
Challenge 1: Host is king

- Browsers want the final say over sizing
- min/max document host should overrule min/max document author
- Document authors may want to opt-in to the ideal size of the browser but with constraints by specifying min/max



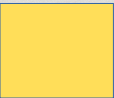
Challenge 2: lifetime changes

- Extensions do not know if their document will be used as flexible popup or as full page document
- With iPad window sizing, flexible size can be morphed into fixed size and reverse



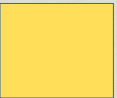
Challenge 3: Backwards compatible

- Feature detection for new sizing features
- Different need for opt-in



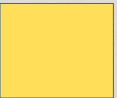
Challenge 4: Sizing before load

- Browsers wait for load before showing popup



Challenge 5: One dimension flexibility

- Bottom panel
 - Fixed width
 - Self-determined height
- Sidebar
 - Self-determined width
 - Fixed height
- Note: in practice, all browsers set fixed sizing for both



How can we address these?

- media queries
- @selector
- meta tags

Some potential solutions (0)

- Existing proposals
 - <https://github.com/w3c/csswg-drafts/issues/1771>
 - <https://github.com/domenic/cooperatively-sized-iframe>

**Contain-intrinsic-size:
from-element**

```
<html  
  requestedwidth="480"  
  requestedheight="320">
```

Some potential solutions (1)

- Allow host documents to query for auto sizing conditions

Addresses:

- one direction
- backwards compatibility
- lifetime changes

```
@media (autosize: horizontal) {  
  html {  
    min-width: 300px;  
    max-width: 600px;  
  }  
}
```

Some potential solutions (2)

- Using environment variables

Addresses:

- Double scrollbars
- lifetime changes

```
html, body {  
  max-height:  
    env(auto-size-max-height, 600px);  
  
  max-width:  
    env(auto-size-max-width, 800px);  
}
```

Some potential solutions (3)

Like @page, specify freeform properties. Which will only be used in freeform mode

Addresses:

- one direction
- backwards compatibility
- lifetime changes
- Opt-in

Benefits:

- Does not cause issues with html/body sizing
- Allows document author to conditionally opt-into host preference size declaratively

```
@freeform {
```

- min-width: 200px;
- width: fit-content;
- max-width: 500px;
- }

Some potential solutions (4)

- Like color-scheme, allow specifying both as meta and as css value

Addresses:

- Waiting on load
- opt-in

```
<meta rel="viewport" "freeform-  
min-width=200,freeform-max-  
width=400">
```

```
<meta rel="freeform" "min-  
width=200,max-width=400">
```

Discussion and next steps

- Which approach would work best?
- Can freeform documents in extensions be solved together with iframe auto-sizing?
- Could the webExtension ecosystem be used as test ground?
- Naming bikeshedding

<https://github.com/w3c/tpac2025-breakouts/issues/58>

<https://github.com/w3c/webextensions/issues/692>