

# Ready for developer testing: Scoped view transitions



Bramus



(<https://twitter.com/bramus>)



(<https://github.com/bramus>)



(<https://frontend.social/@bramus>)



(<https://bsky.app/profile/bramus.us>)



(<https://www.bram.us/>)

Published: Sep 10, 2025

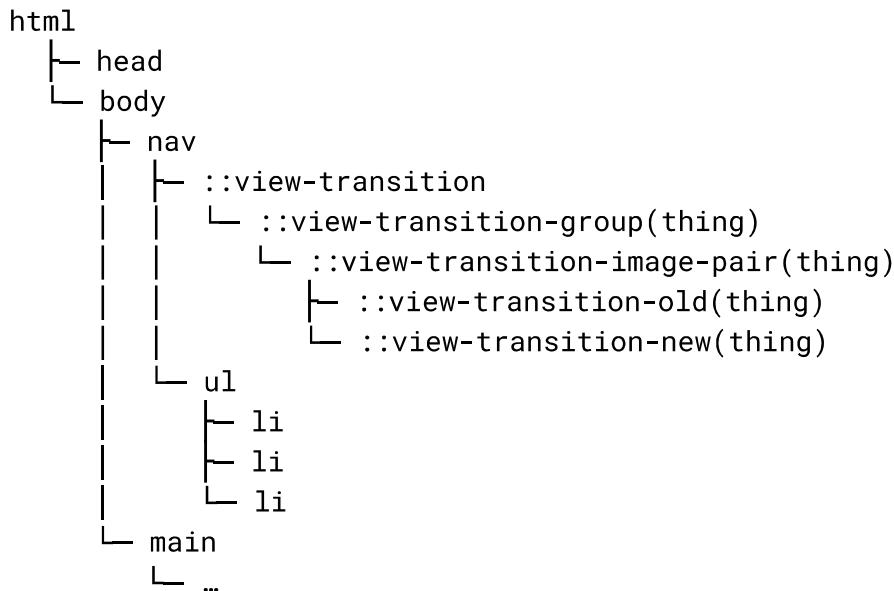
Scoped view transitions is a proposed extension to the [View Transition API](#) (/docs/web-platform/view-transitions) which lets you start a view transition on a subtree of the DOM. The feature is ready for testing from Chrome 140 with the "Experimental Web Platform features" flag enabled in `chrome://flags`. We're actively looking for developer feedback.

## Quick intro

To start a scoped view transition, call `element.startViewTransition()` (instead of `document.startViewTransition()`) on any `HTMLElement` with `contain: layout` applied to it.

```
document.querySelector('nav').startViewTransition({
  callback: () => {
    // ... code that manipulates the contents of <nav>
  },
})
```

The result is a view transition scoped to the subtree of your chosen `HTMLElement`, with the resulting `::view-transition` pseudo-tree getting injected onto that `HTMLElement`. The `HTMLElement` acts as a container for the transition animations.



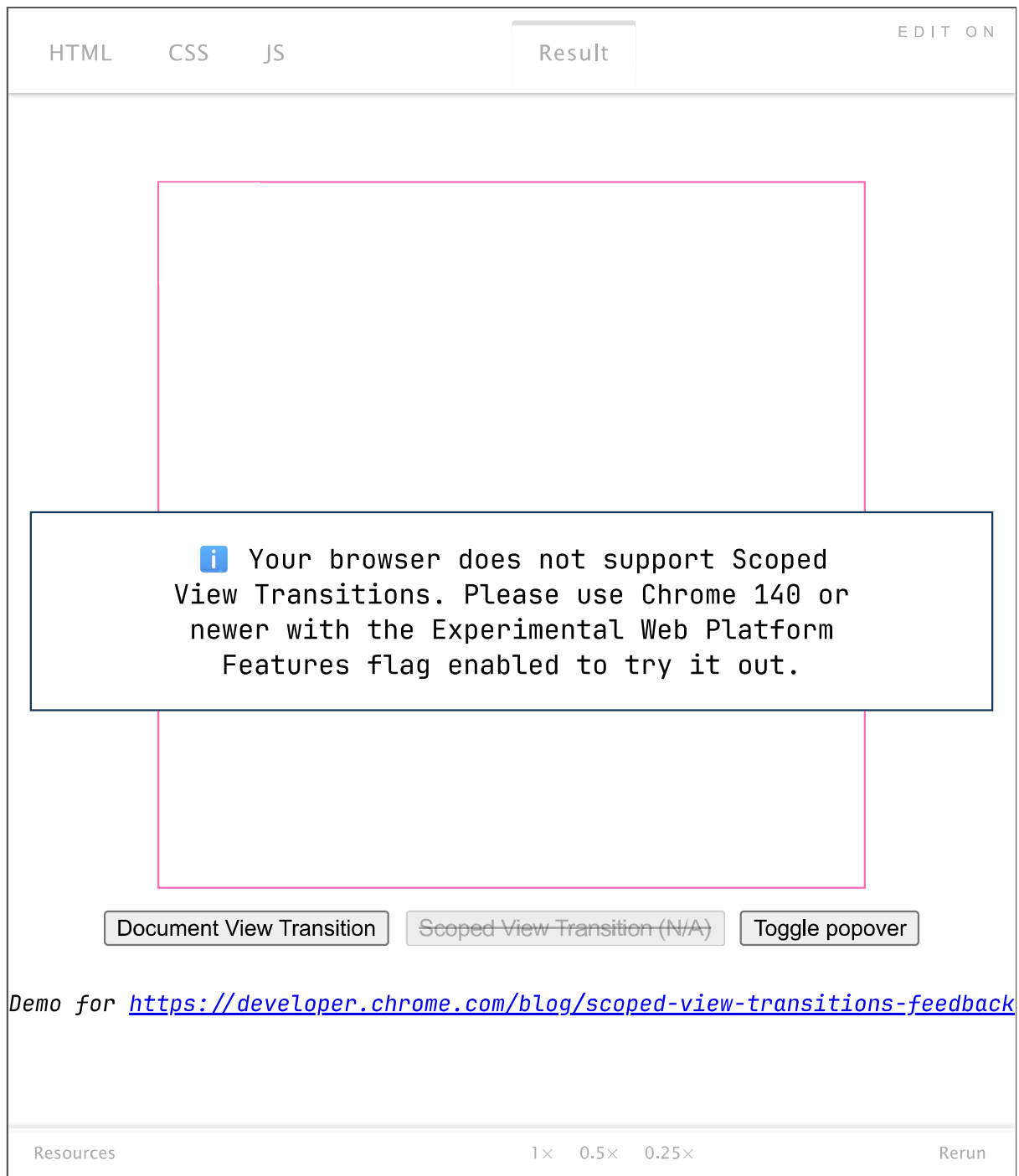
Scoped transitions allow multiple view transitions to run at the same time (as long as they have different transition roots). Pointer events are prevented on only that subtree ([which you can re-enable](https://www.bram.us/2025/01/29/view-transitions-page-interactivity/) (<https://www.bram.us/2025/01/29/view-transitions-page-interactivity/>)), rather than the whole document. In addition, they allow elements outside of the transition root to be painted on top of the scoped view transition.

Learn all about scoped view transitions in [the scoped view transitions explainer](https://github.com/WICG/view-transitions/blob/main/scoped-transitions.md) (<https://github.com/WICG/view-transitions/blob/main/scoped-transitions.md>).

## Demo

In the following demo, you can move a dot within its container by clicking one of the buttons. This can be done with document-scoped view transitions or element-scoped view transitions, allowing you to see how they behave differently.

[Live Demo](#)[Demo Recording...](#)  
(#live-demo)



With scoped view transitions, the dot remains clipped by its parent during the transition. Furthermore the dot does not paint over the popover when shown.

## Feedback wanted

We're interested in feedback from the web developer community about:

- The shape of the Scoped View Transitions API.

- Use-cases where the feature works well or didn't work as expected.
- How the questions about self-participating scopes ([https://docs.google.com/document/d/1l\\_38aS2\\_W1iKT35wClSkvoJ8JYDPCmuVuhPmowSG6pU/eidit?tab=t.0#heading=h.50yc5154aelq](https://docs.google.com/document/d/1l_38aS2_W1iKT35wClSkvoJ8JYDPCmuVuhPmowSG6pU/eidit?tab=t.0#heading=h.50yc5154aelq)) should be resolved.

Share your feedback by commenting on CSS WG issue #9890

(<https://github.com/w3c/csswg-drafts/issues/9890>). Familiarize yourself with the known issues (<https://github.com/WICG/view-transitions/blob/main/scoped-transitions.md#known-issues>) and constraints (<https://github.com/WICG/view-transitions/blob/main/scoped-transitions.md#constraints>) before leaving feedback.

If you run into a bug, file a Chromium bug (<https://crbug.com/new>) to let us know.

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the Apache 2.0 License (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the Google Developers Site Policies (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-09-10 UTC.