

# *View Transitions: Cranking it up to 11 (Safely, Maybe)*

**@bramus – 2026.05.07 (All Day Hey!)**

# Bramus



 @bram.us

 @bramus@front-end.social

~~@bramus~~



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

W3C Editor's Draft

CSS View Transitions Module

drafts.csswg.org/css-view-transitions-1/


Work

## TABLE OF CONTENTS

- 1 Introduction**
  - 1.1 Separating Visual Transitions from DOM Updates
  - 1.2 View Transition Customization
  - 1.3 View Transition Lifecycle
  - 1.4 Transitions as an enhancement
  - 1.5 Rendering Model
  - 1.6 Examples
- 2 CSS properties**
  - 2.1 Tagging Individually Transitioning Subtrees: the 'view-transition-name' property
    - 2.1.1 Rendering Consolidation
- 3 Pseudo-elements**
  - 3.1 Pseudo-element Trees
  - 3.2 View Transition Pseudo-elements
    - 3.2.1 Named View Transition Pseudo-elements
    - 3.2.2 View Transition Tree Root: the '::view-transition' pseudo-element
    - 3.2.3 View Transition Named Subtree Root: the '::view-transition-group()' pseudo-element
    - 3.2.4 View Transition Image Pair Isolation: the '::view-transition-image-pair()' pseudo-element
    - 3.2.5 View Transition Old State Image: the '::view-transition-old()' pseudo-element
    - 3.2.6 View Transition New State Image: the '::view-transition-new()' pseudo-element
- 4 View Transition Layout**
  - 4.1 The Snapshot Containing Block
  - 4.2 View Transition Painting Order
- 5 User Agent Stylesheet**
- 6 API**
  - 6.1 Additions to Document
    - 6.1.1 startViewTransition() Method Steps
  - 6.2 The ViewTransition interface
    - 6.2.1 skipTransition() Method Steps

# CSS View Transitions Module Level 1

Editor's Draft, 16 February 2025



▼ More details about this document

**This version:**  
<https://drafts.csswg.org/css-view-transitions-1/>

**Latest published version:**  
<https://www.w3.org/TR/css-view-transitions-1/>

**Implementation Report:**  
<https://wpt.fyi/results/css/css-view-transitions>

**Feedback:**  
[CSSWG Issues Repository](#)

**Editors:**  
[Tab Atkins-Bittner](#) (Google)  
[Jake Archibald](#) (Google)  
[Khushal Sagar](#) (Google)

**Suggest an Edit for this Spec:**  
[GitHub Editor](#)

Copyright © 2025 World Wide Web Consortium. W3C® liability, trademark and permissive document license rules apply.

## Abstract

This module defines the View Transition API, along with associated properties and pseudo-elements, which allows developers to create animated visual transitions representing changes in the document state.

CSS is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, etc.

## Status of this document

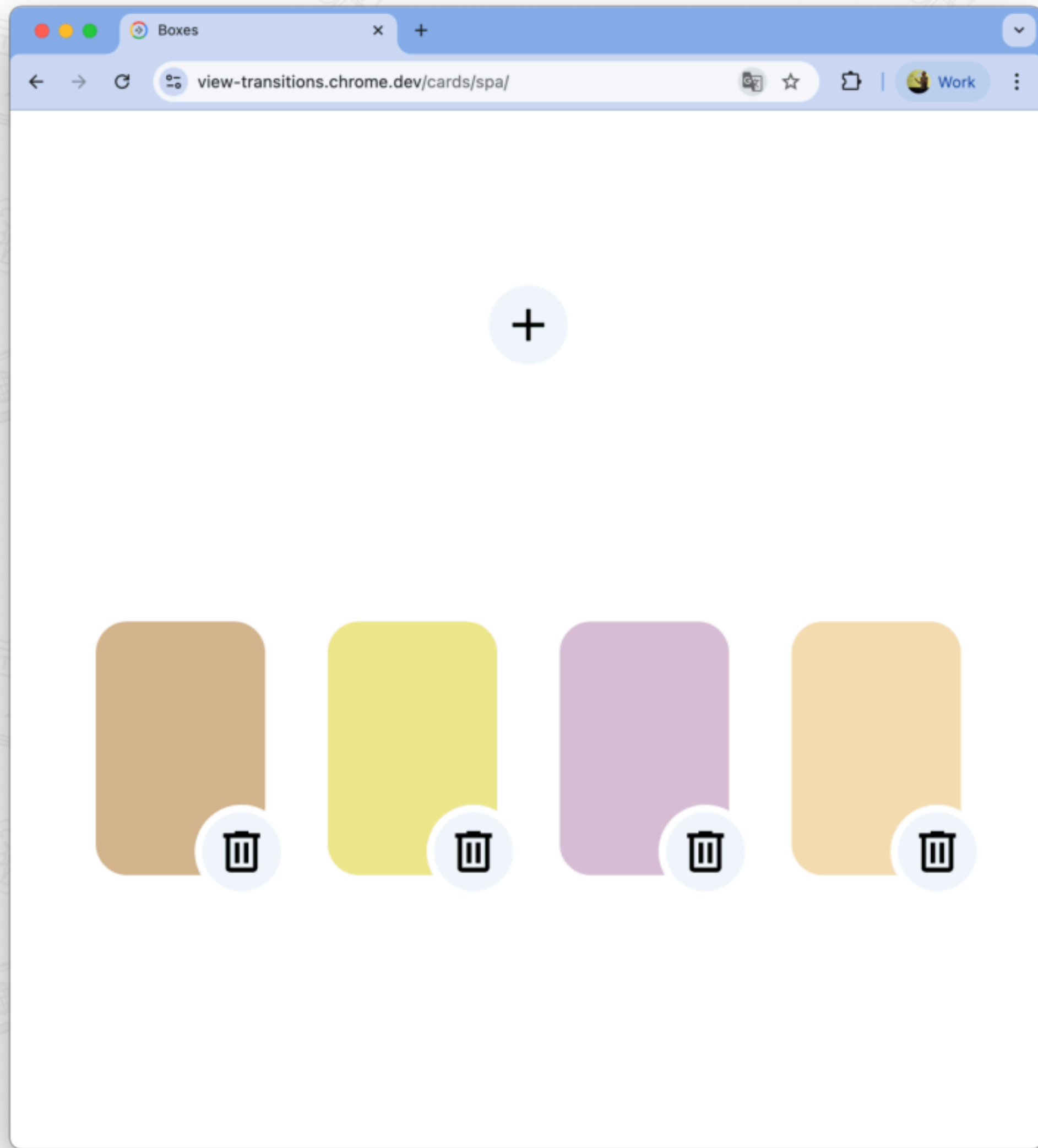
This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress.

Please send feedback by [filing issues in GitHub](#) (preferred), including the spec code "css-view-transitions" in the title, like this: "[css-view-transitions] [summary of comment](#)". All issues and comments are [archived](#).

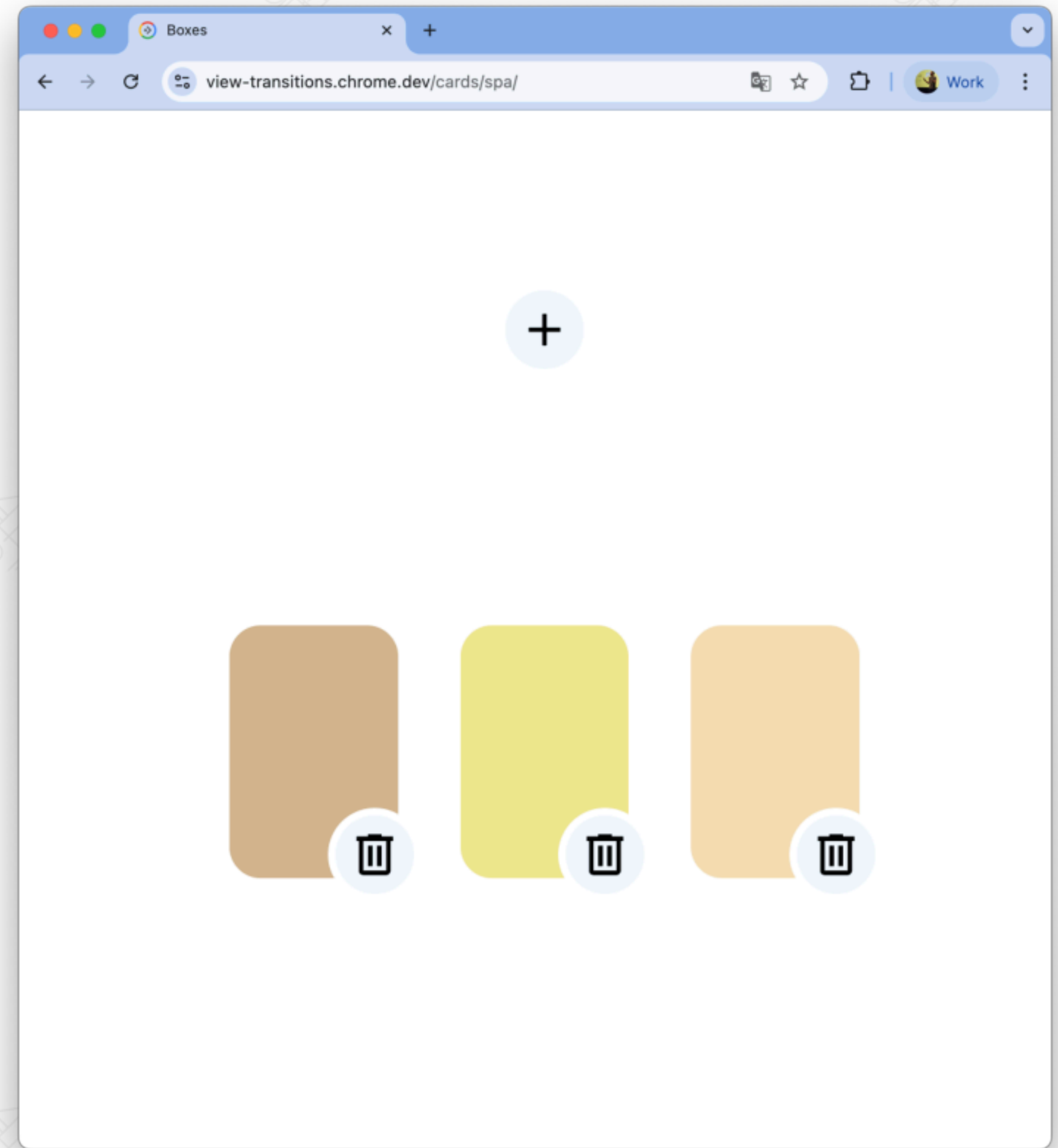
<https://drafts.csswg.org/css-view-transitions-1/>

***“The View Transition API gives you the power to create seamless visual transitions between different views on your website.”***

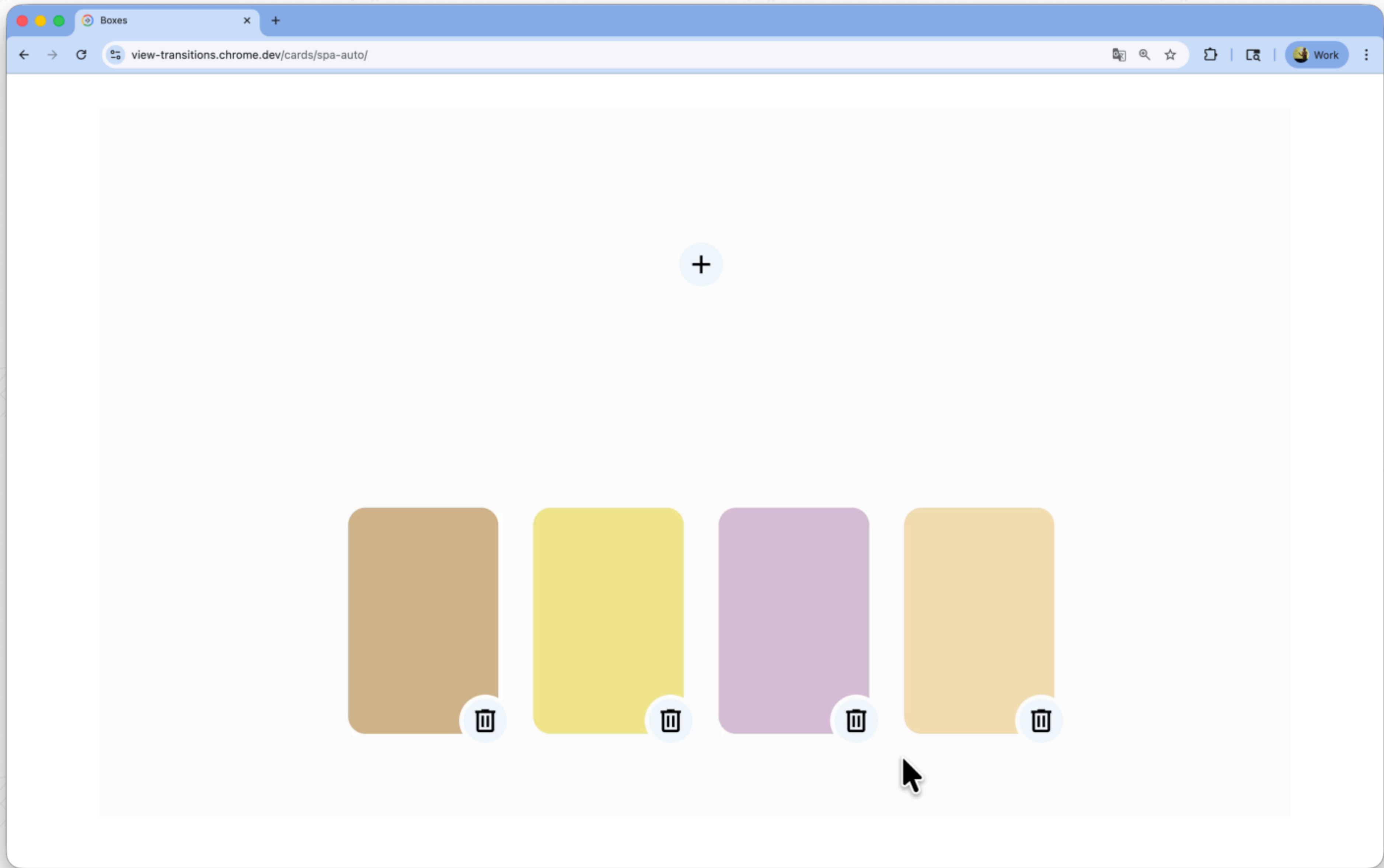
<https://goo.gle/view-transitions>



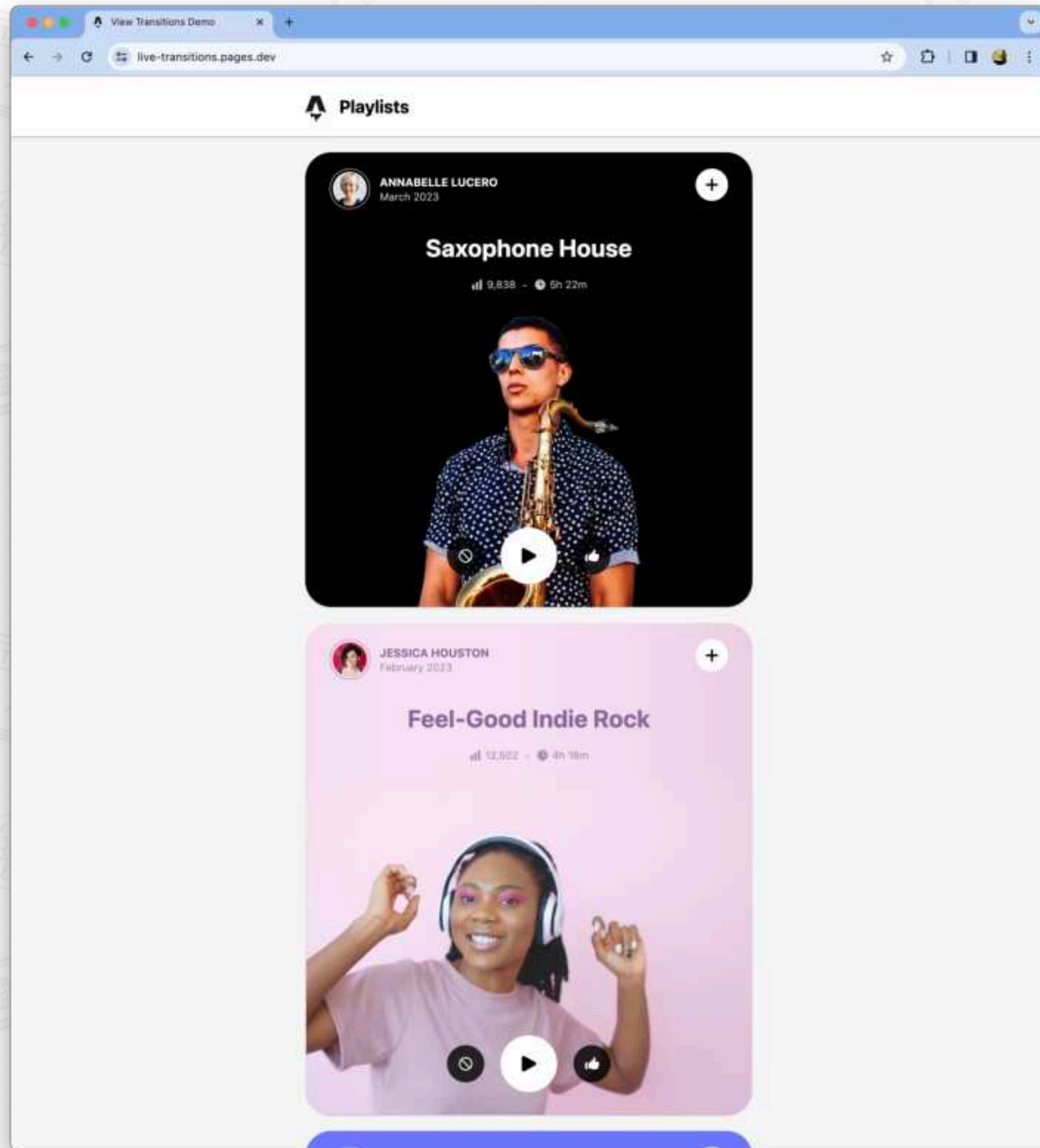
4 items



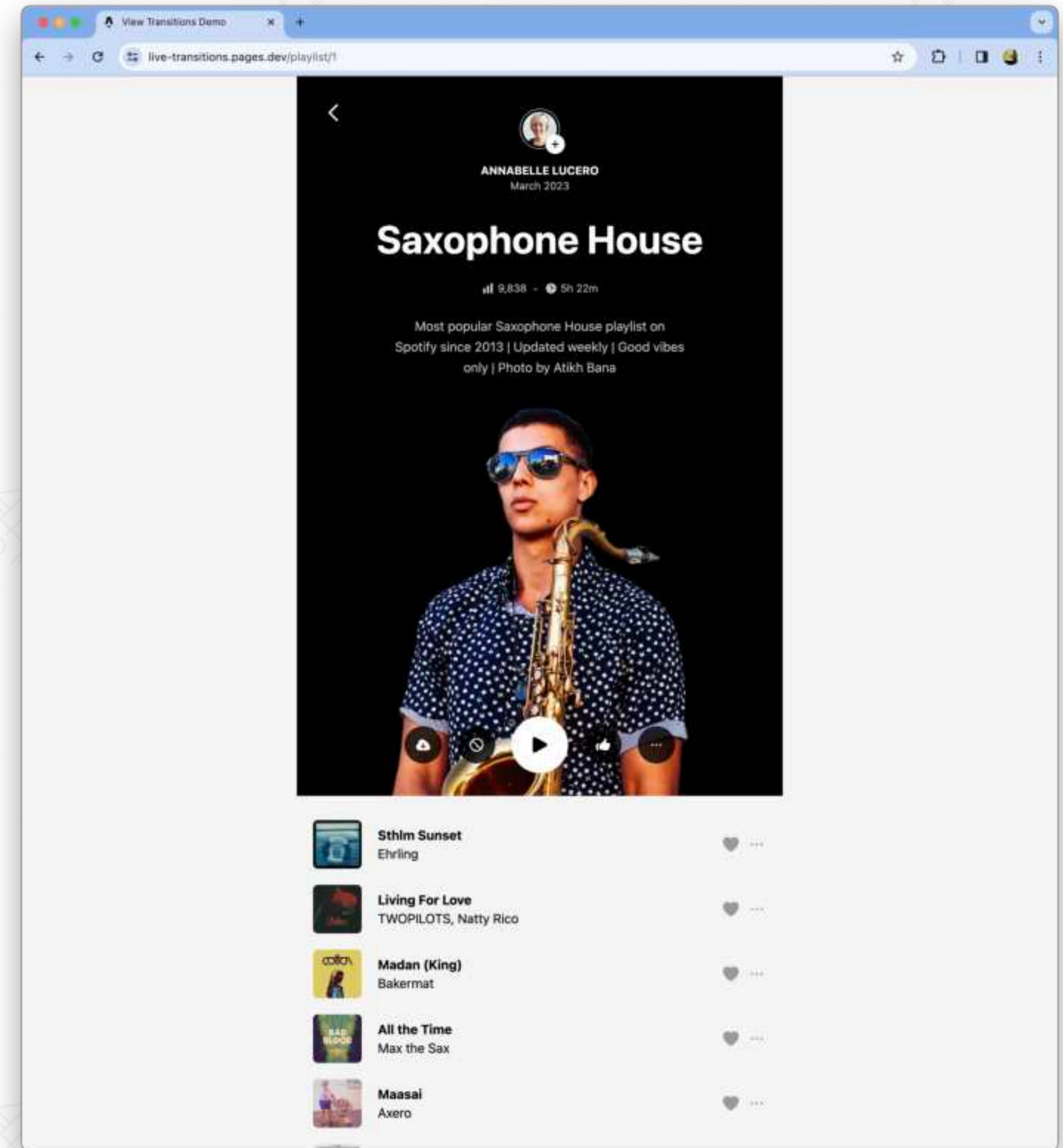
3 items



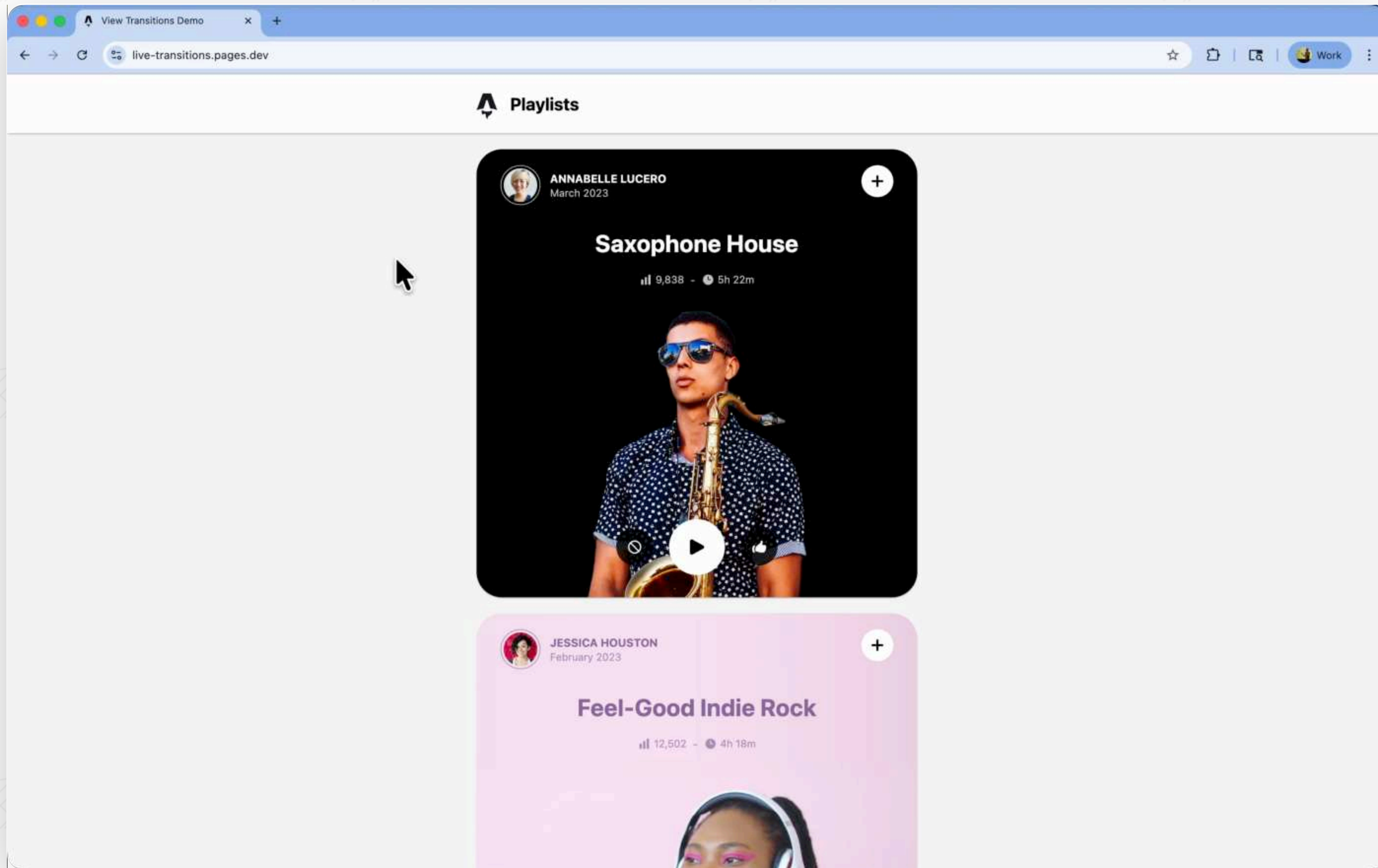
<https://view-transitions.chrome.dev/cards/spa/>



Overview



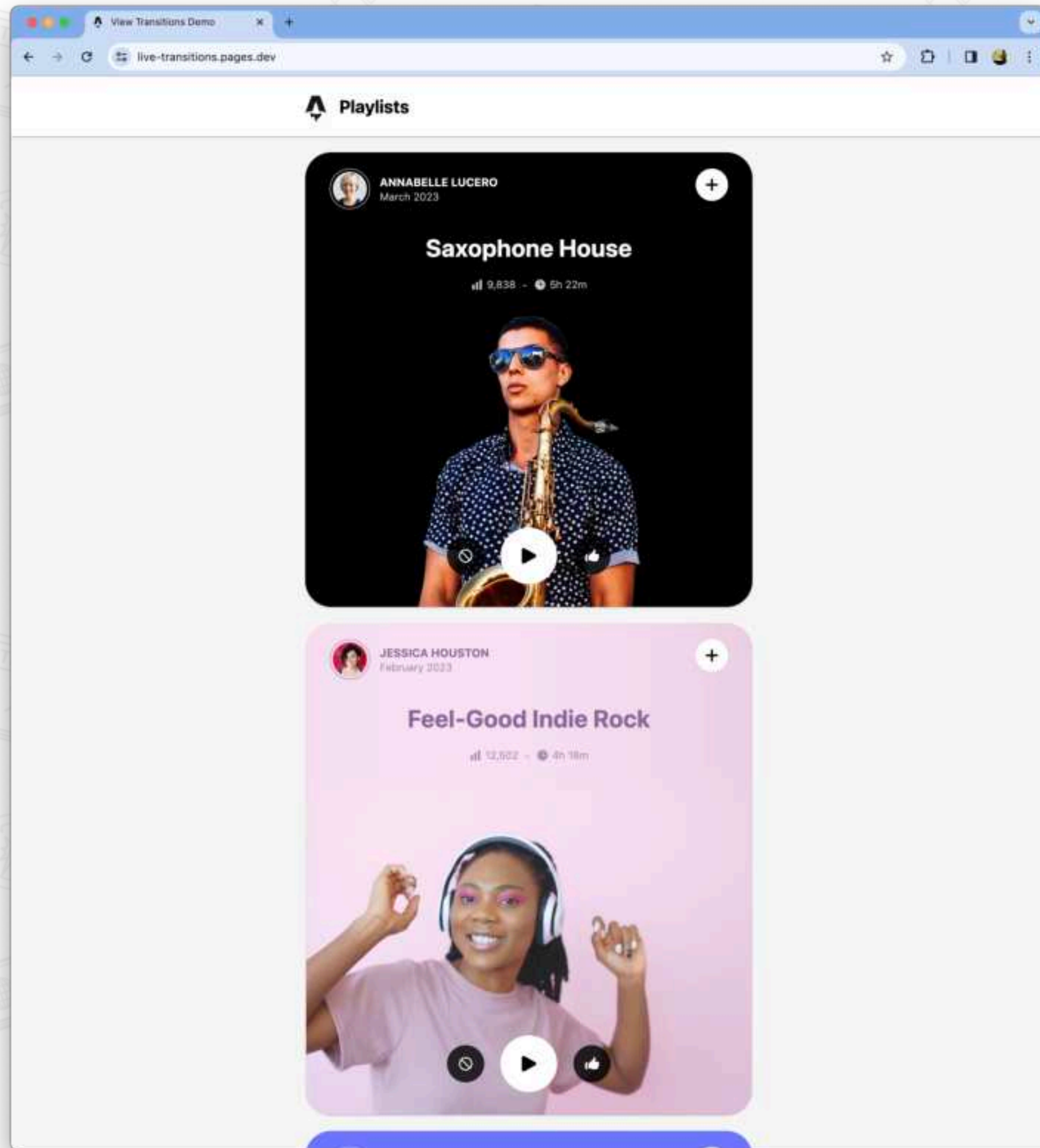
Detail



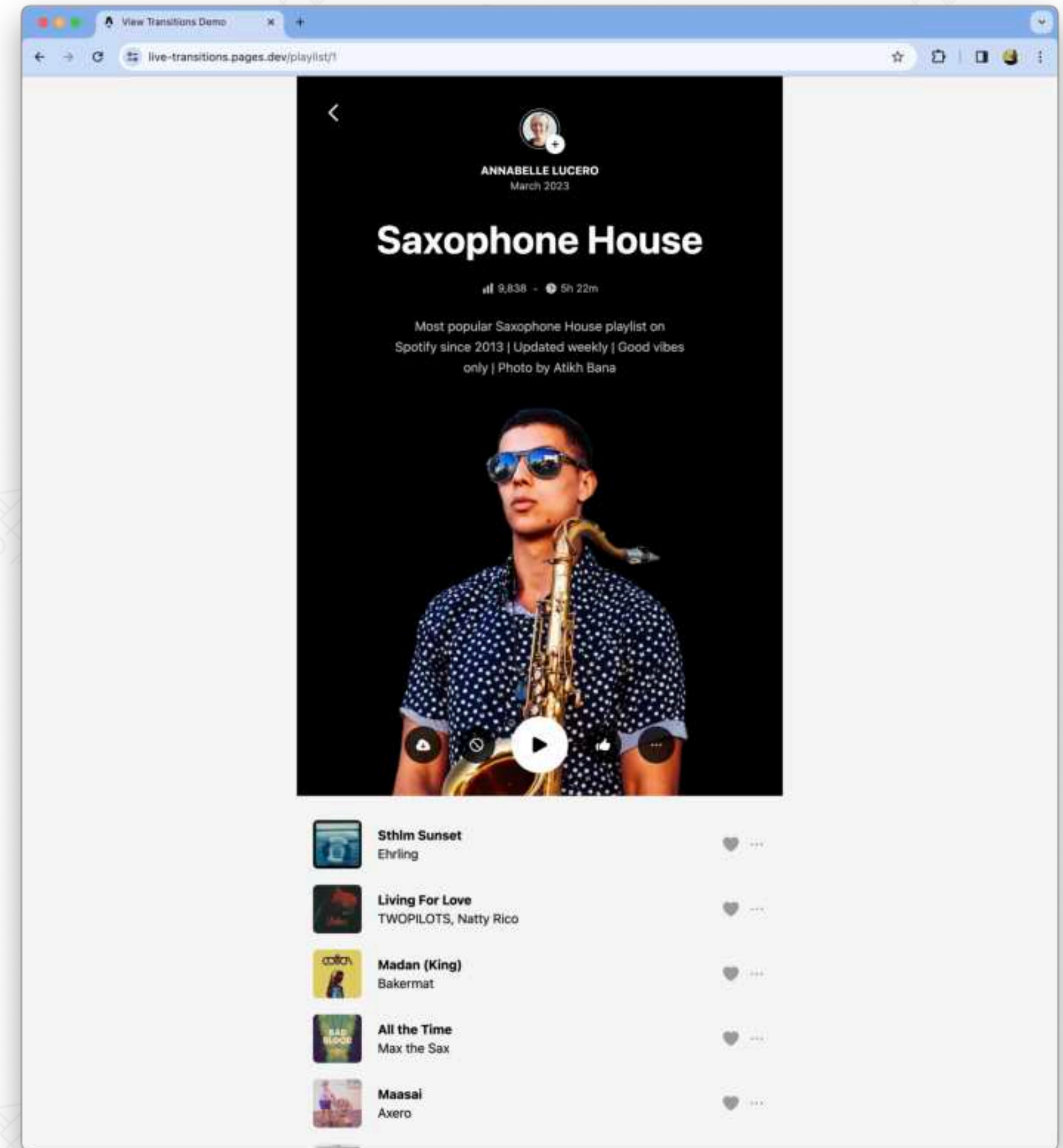
<https://live-transitions.pages.dev/> by [Maxi Ferreira](#)

# 1. Identify and name elements to transition

CSS `view-transition-name` property

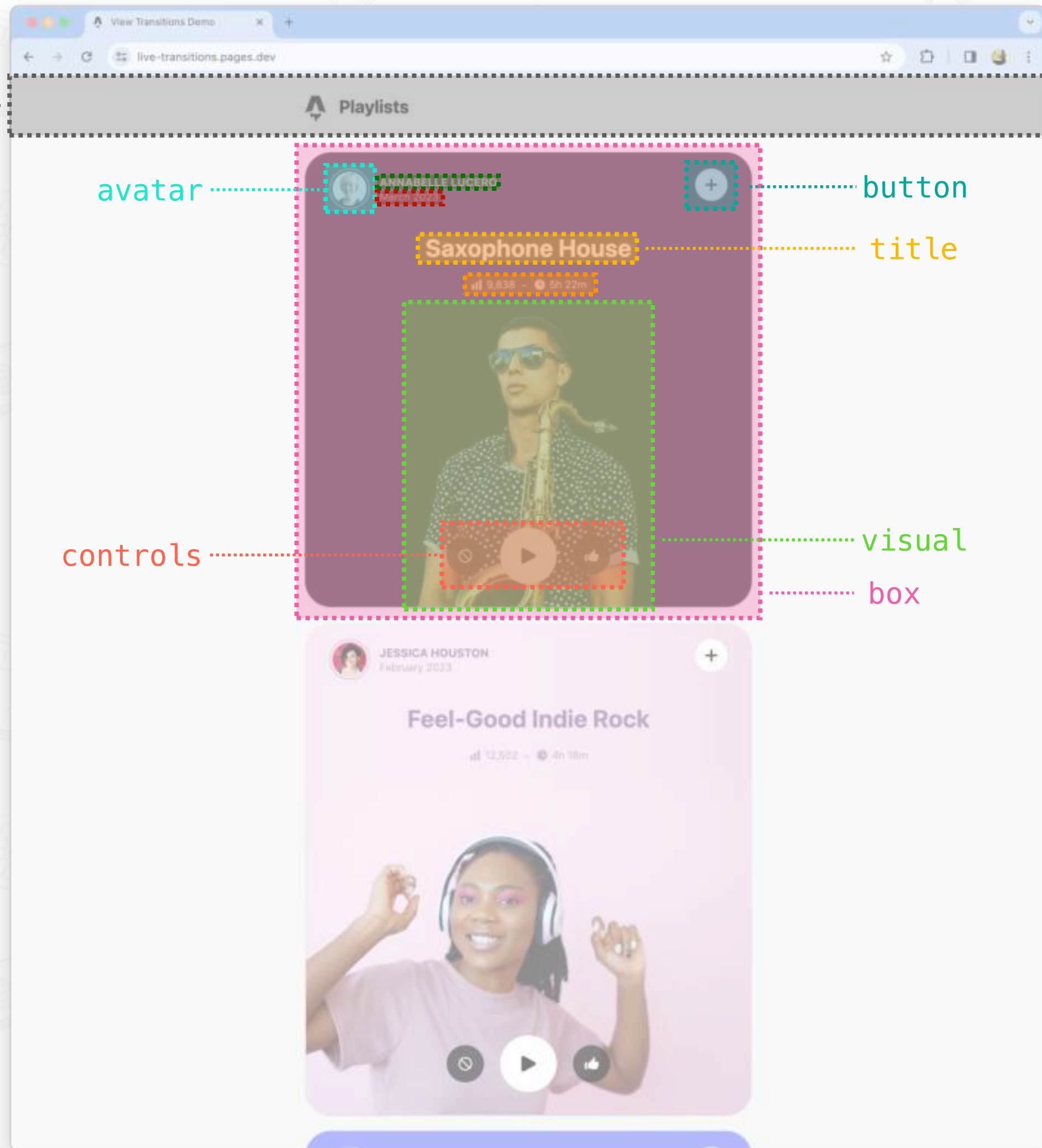


Overview

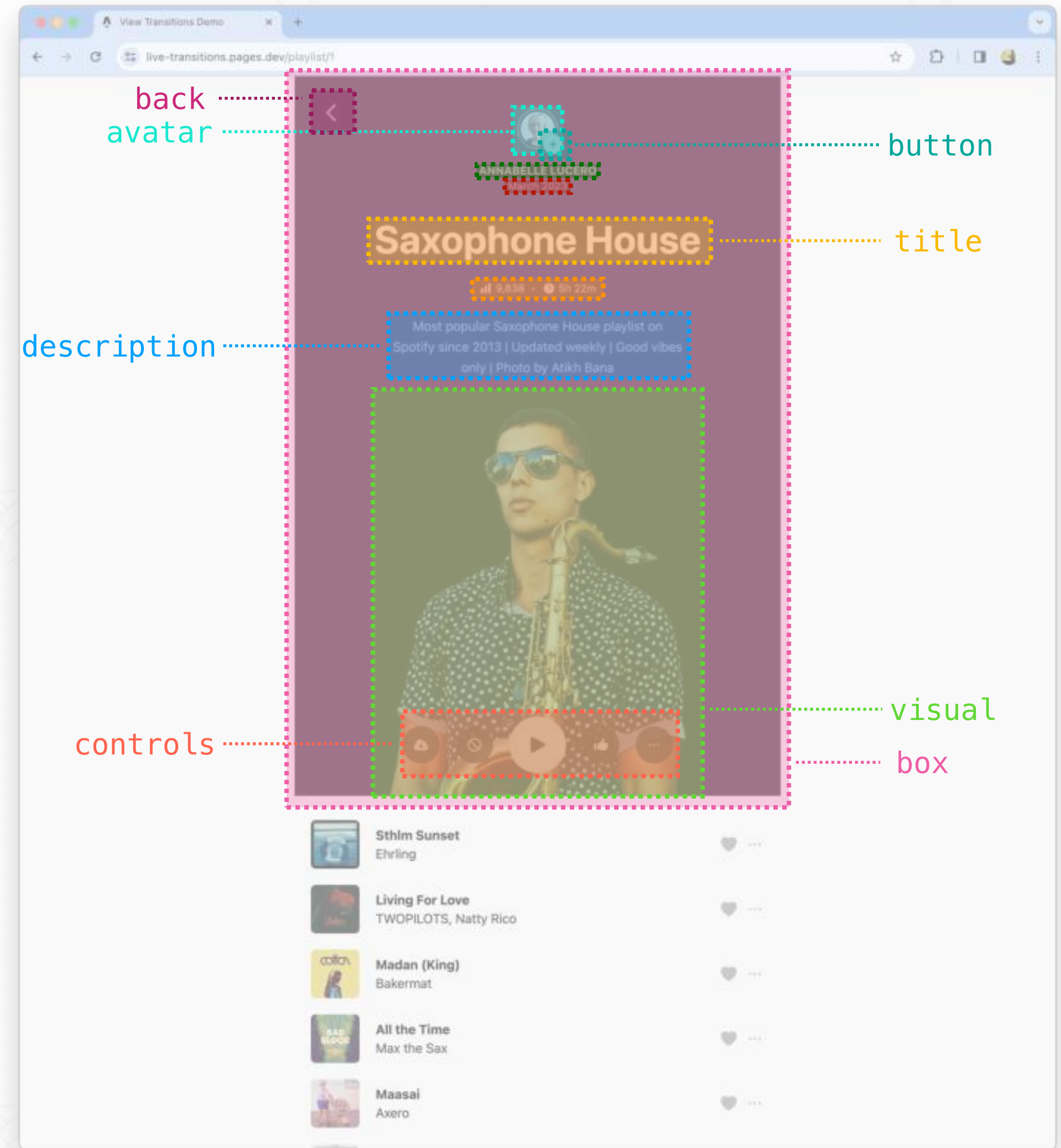


Detail

topbar



Overview



Detail

```
.card {  
  view-transition-name: box;  
}
```

```
.overview .card h2 {  
  view-transition-name: title;  
}
```

```
.detail .card h1 {  
  view-transition-name: title;  
}
```

# 2. Trigger the View Transition

Same-Document or Cross-Document



111



144



18

```
$link.addEventListener('click', e => {  
  if (!document.startViewTransition) {  
    showPlaylist(e.currentTarget);  
    return;  
  }  
  
  document.startViewTransition(() => {  
    showPlaylist(e.currentTarget);  
  });  
});
```



126



WIP

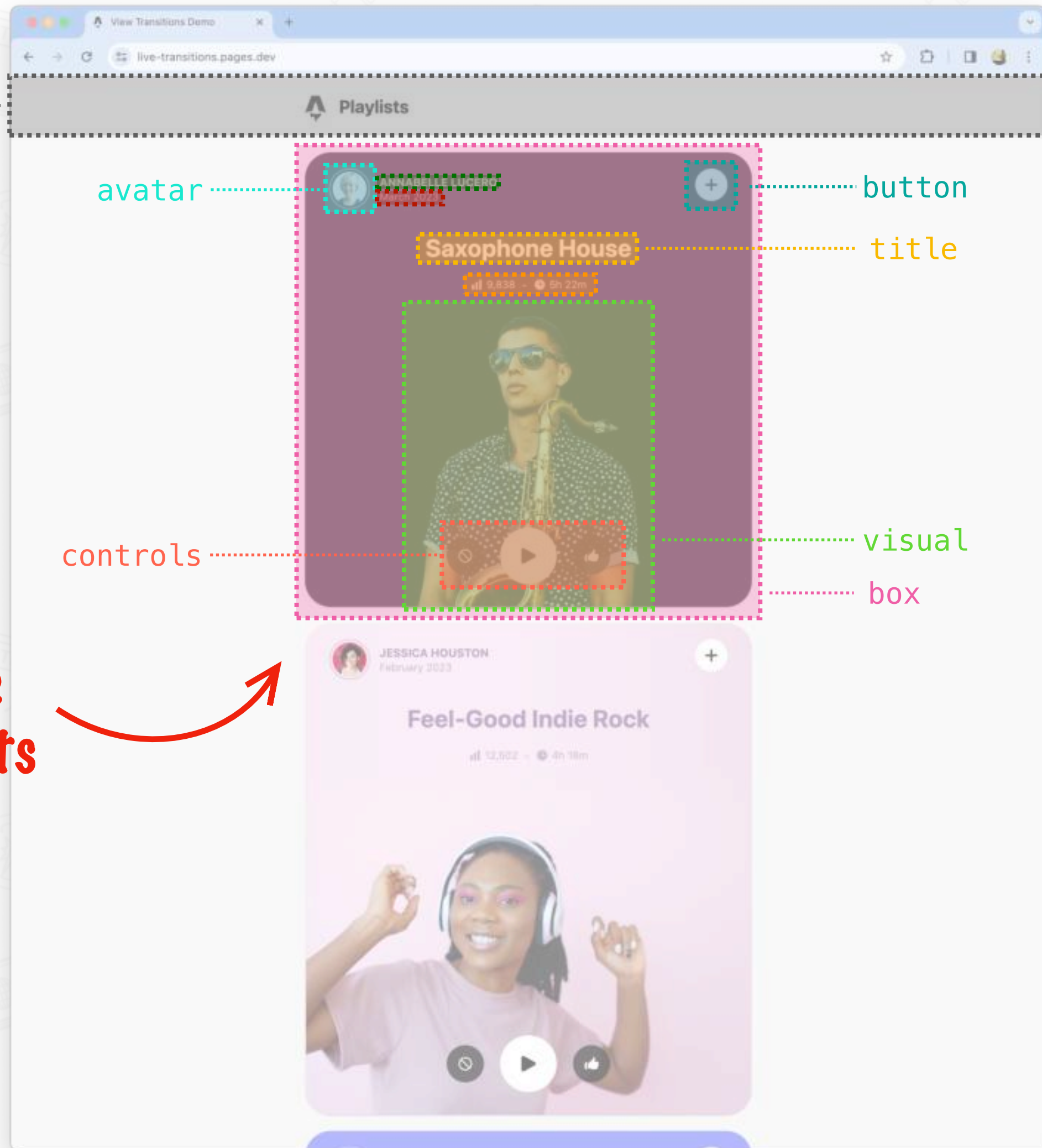


18.2

```
<a href="/playlists/2">Show Playlist</a>  
<form action="/process" method="POST"></form>
```

```
@view-transition {  
  navigation: auto;  
}
```

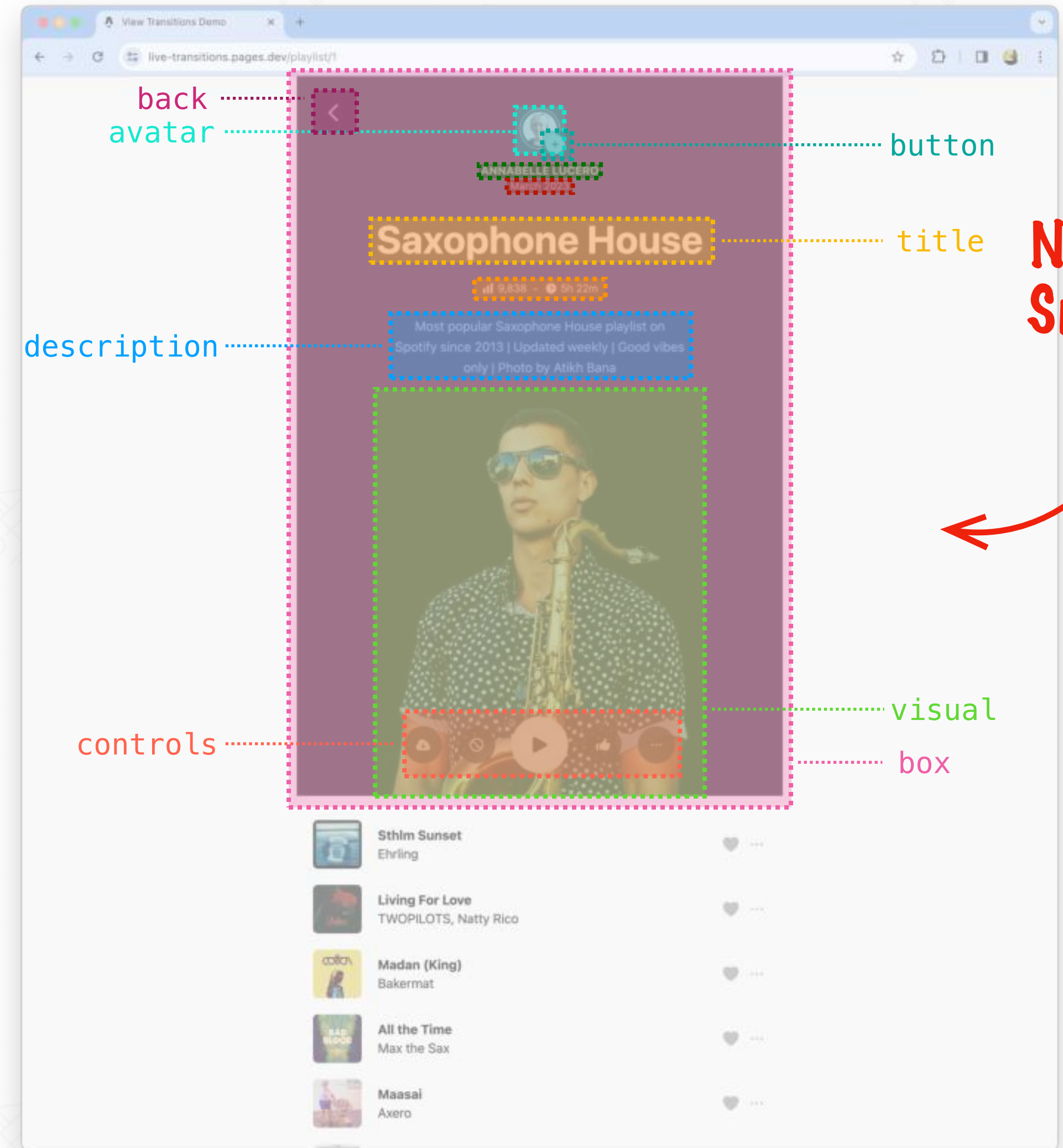
topbar



Old state  
Snapshots



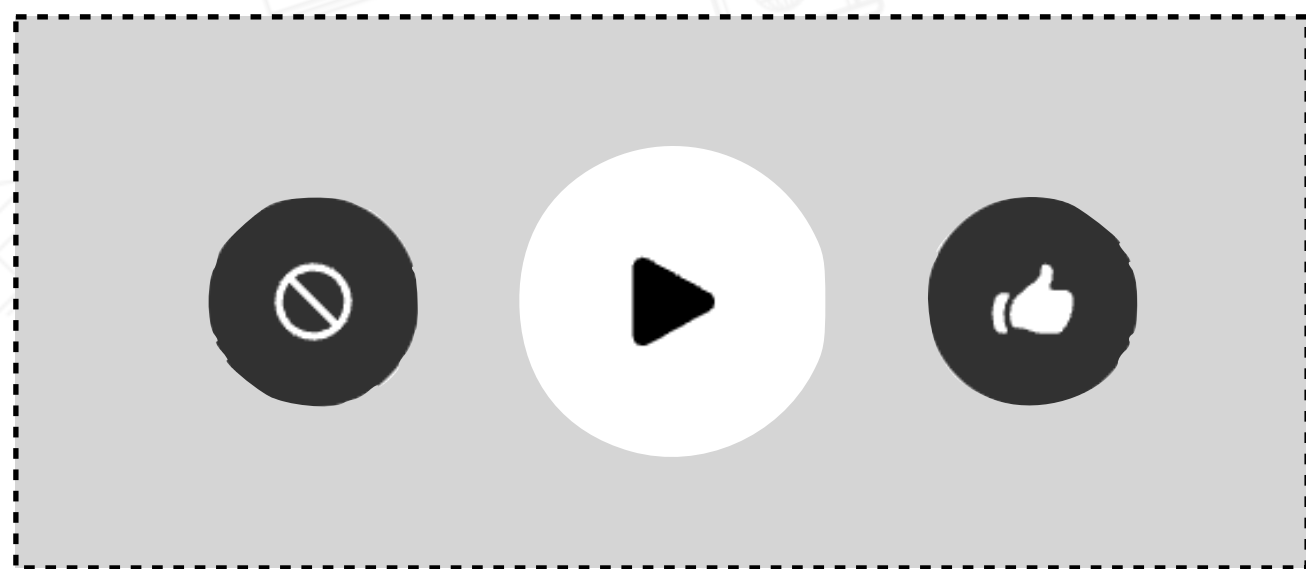
Overview



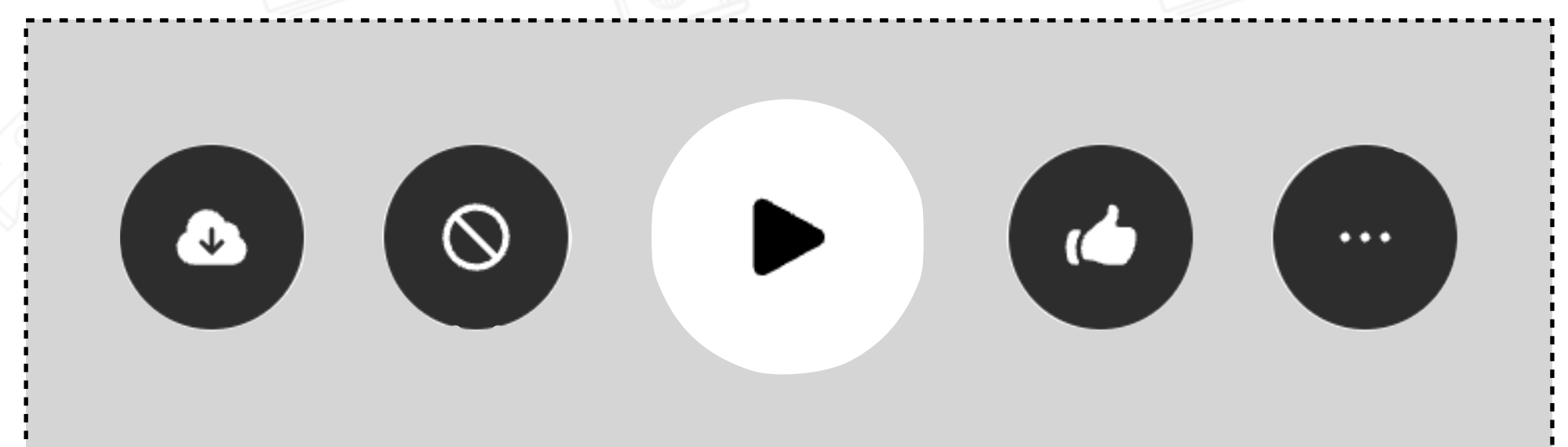
New state  
Snapshots



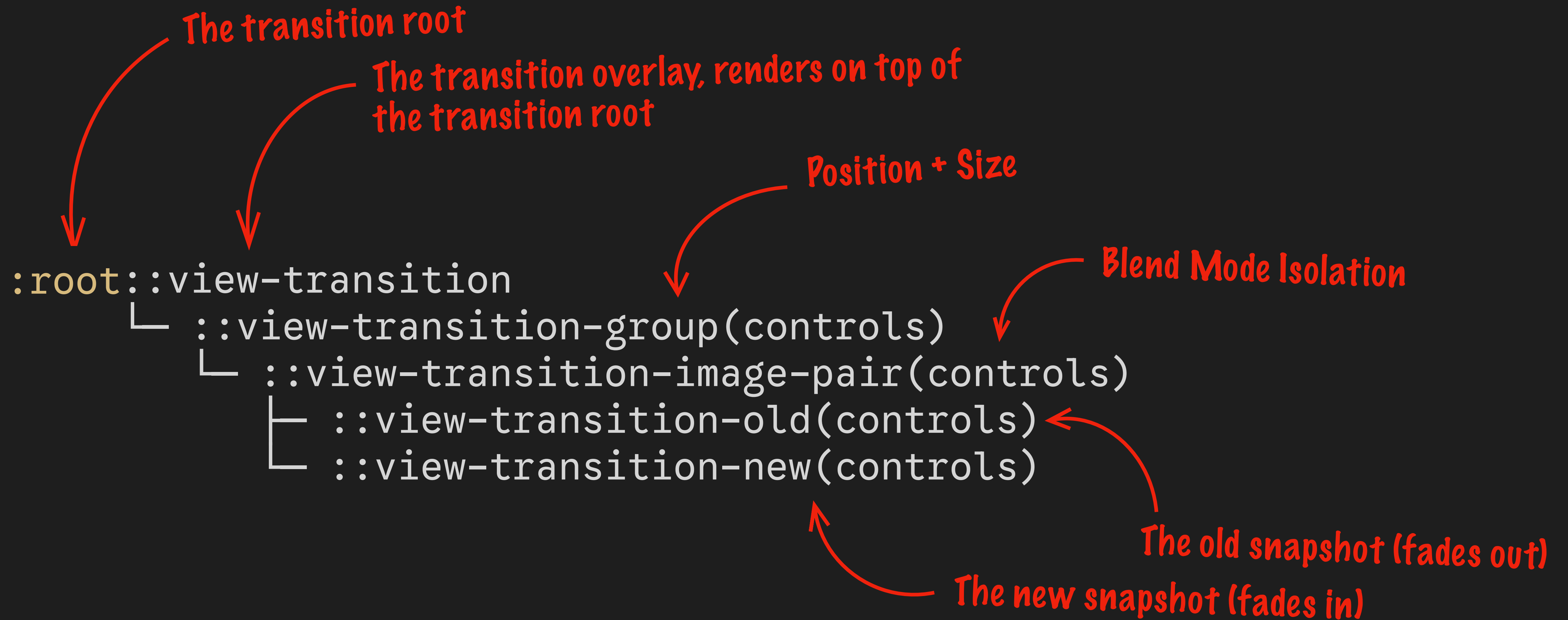
Detail

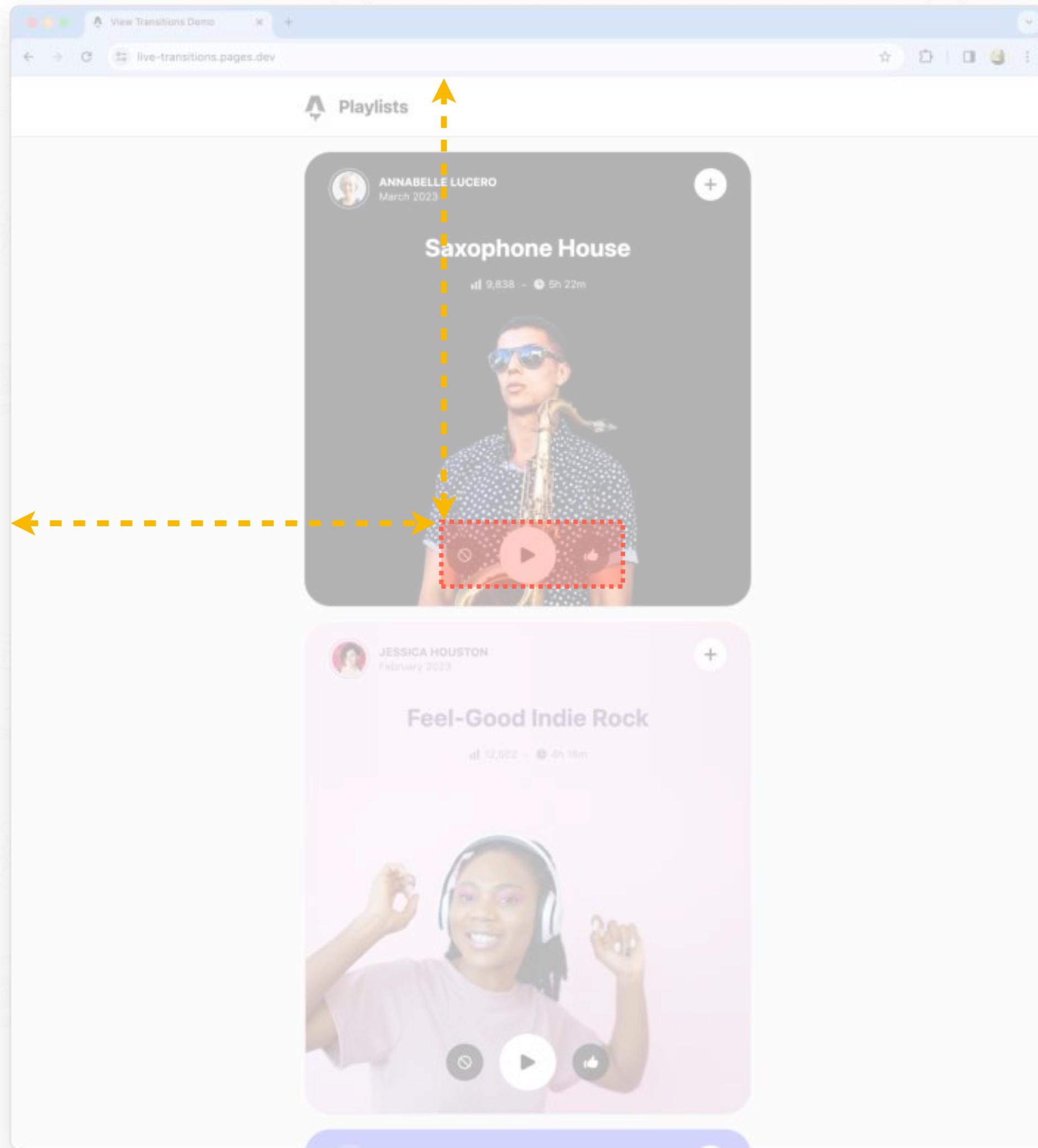


`::view-transition-old(controls)`

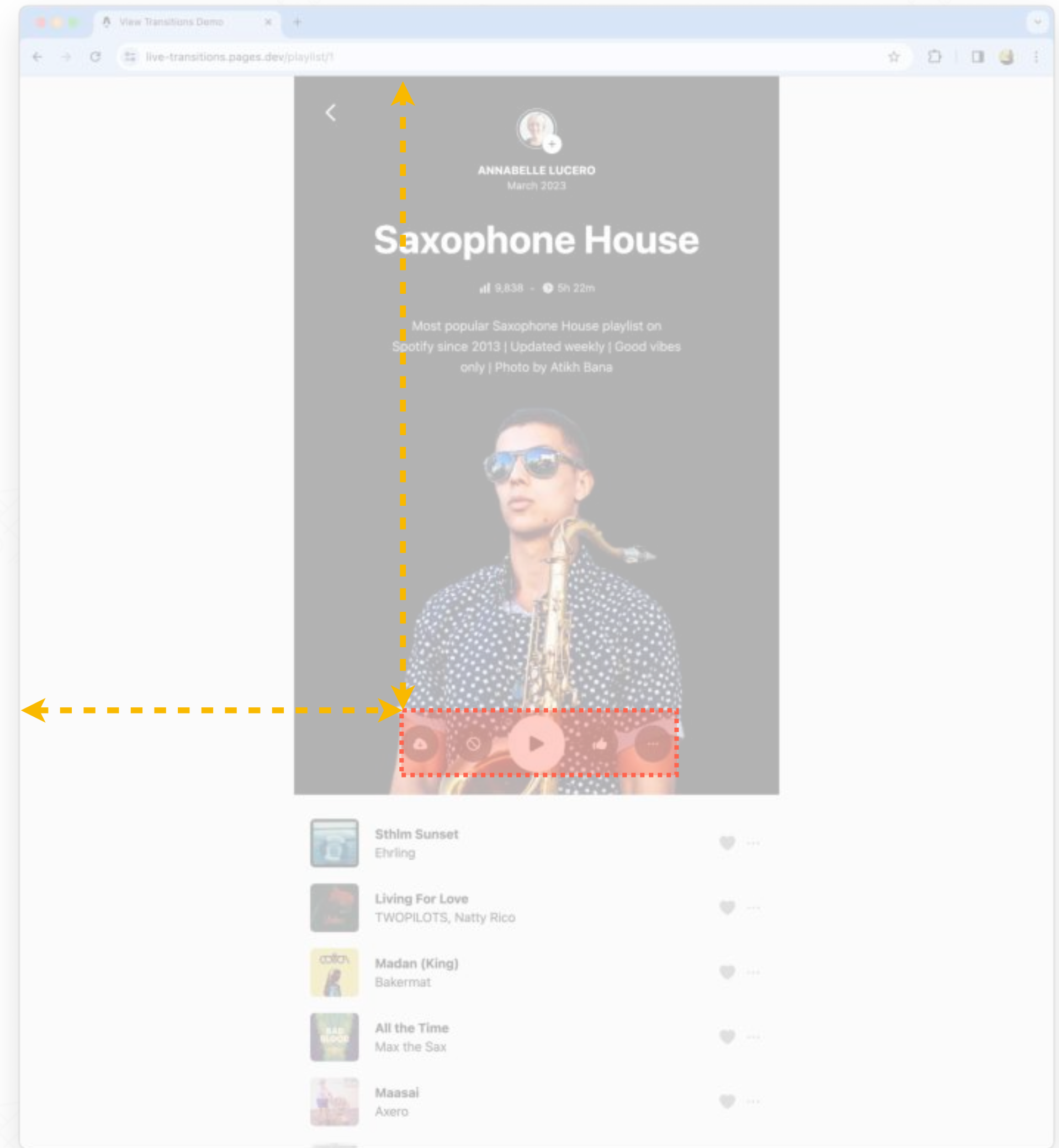


`::view-transition-new(controls)`





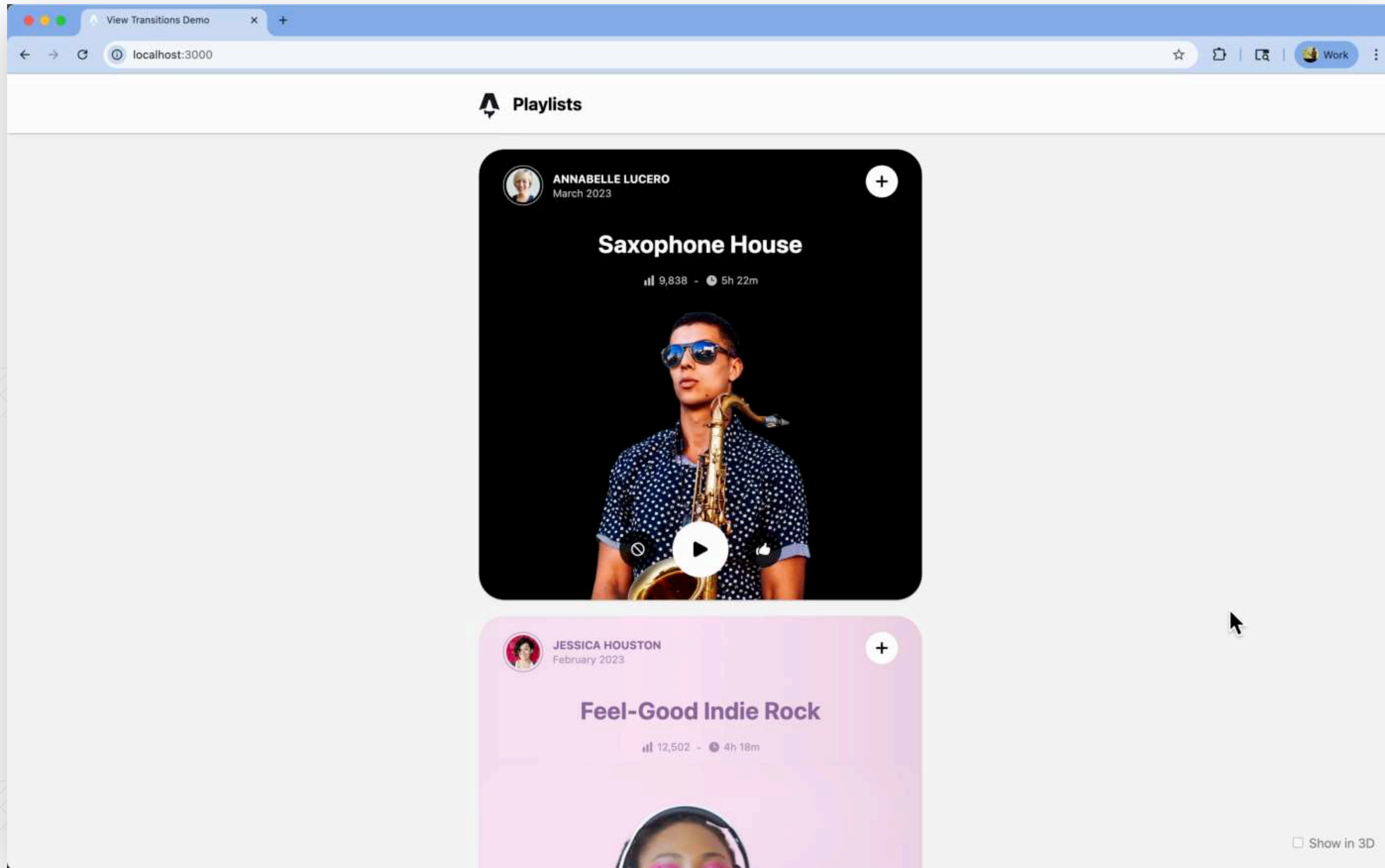
Overview



Detail

`:root::view-transition`

```
├─ ::view-transition-group(card)
│   └─ ::view-transition-image-pair(card)
│       ├── ::view-transition-old(card)
│       └─ ::view-transition-new(card)
├─ ::view-transition-group(visual)
│   └─ ::view-transition-image-pair(visual)
│       ├── ::view-transition-old(visual)
│       └─ ::view-transition-new(visual)
├─ ::view-transition-group(topbar)
│   └─ ::view-transition-image-pair(topbar)
│       └─ ::view-transition-old(topbar)
├─ ::view-transition-group(avatar)
│   └─ ::view-transition-image-pair(avatar)
│       ├── ::view-transition-old(avatar)
│       └─ ::view-transition-new(avatar)
└─ :
```



<https://live-transitions.pages.dev/> by [Maxi Ferreira](#)

# 3. Customize the animations

*(Optional)*

```
::view-transition {
  position: absolute; inset: 0;
}
::view-transition-group($NAME) {
  animation-name: -ua-view-transition-group-anim-$NAME;
}
::view-transition-image-pair(*) {
  isolation: isolate;
}
::view-transition-old(*) {
  animation-name: -ua-view-transition-fade-out;
}
::view-transition-new(*) {
  animation-name: -ua-view-transition-fade-in;
}
```

**Move the group**

**Fade out the old snapshots**

**Fade in the new snapshots**

*User-Agent Generated CSS (partial)*

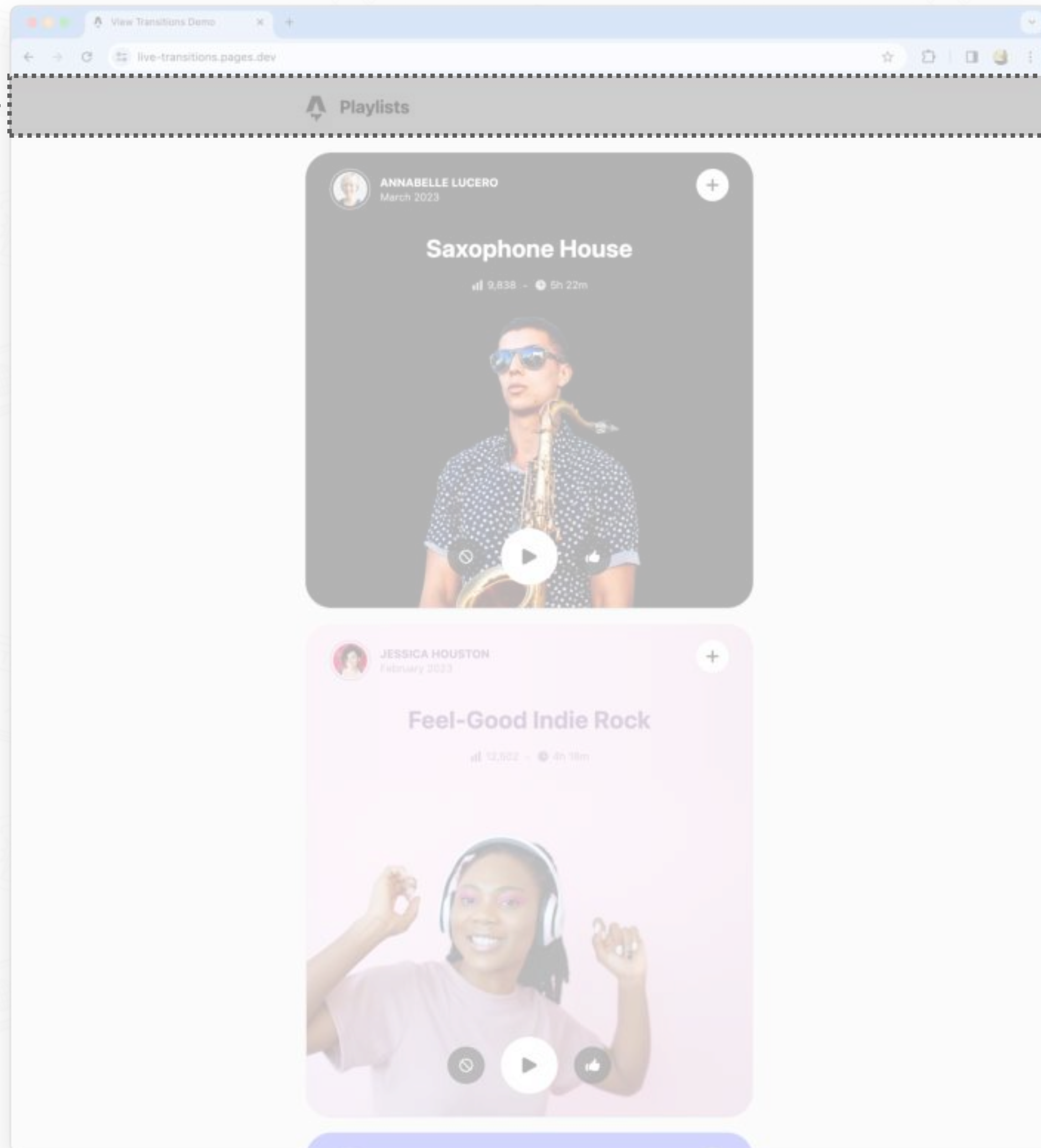


CSS

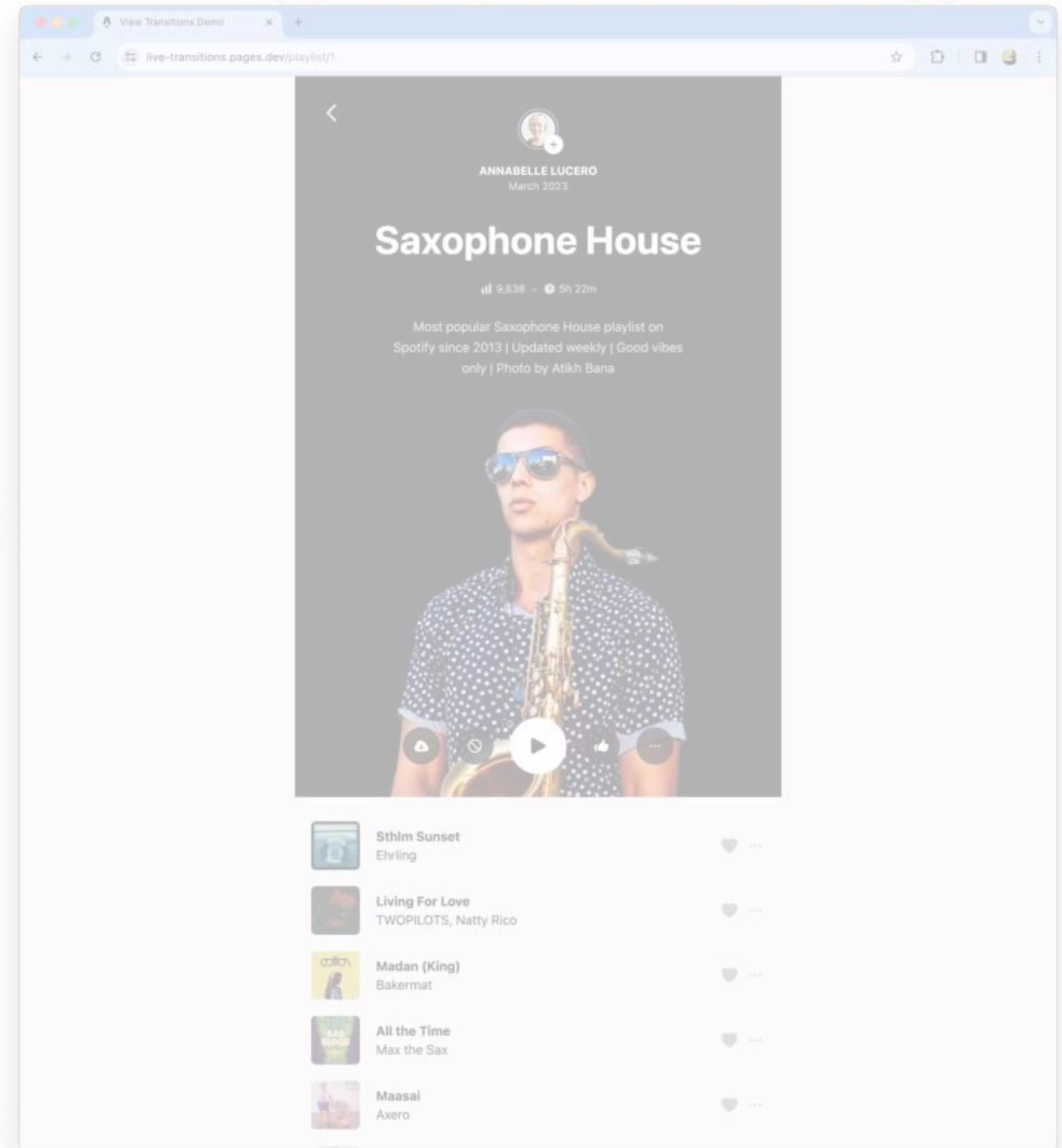
It's a UNIX system. I know this.

```
::view-transition-group(*) {  
  animation-duration: 2s;  
}
```

topbar



Overview



Detail

**:root::view-transition**

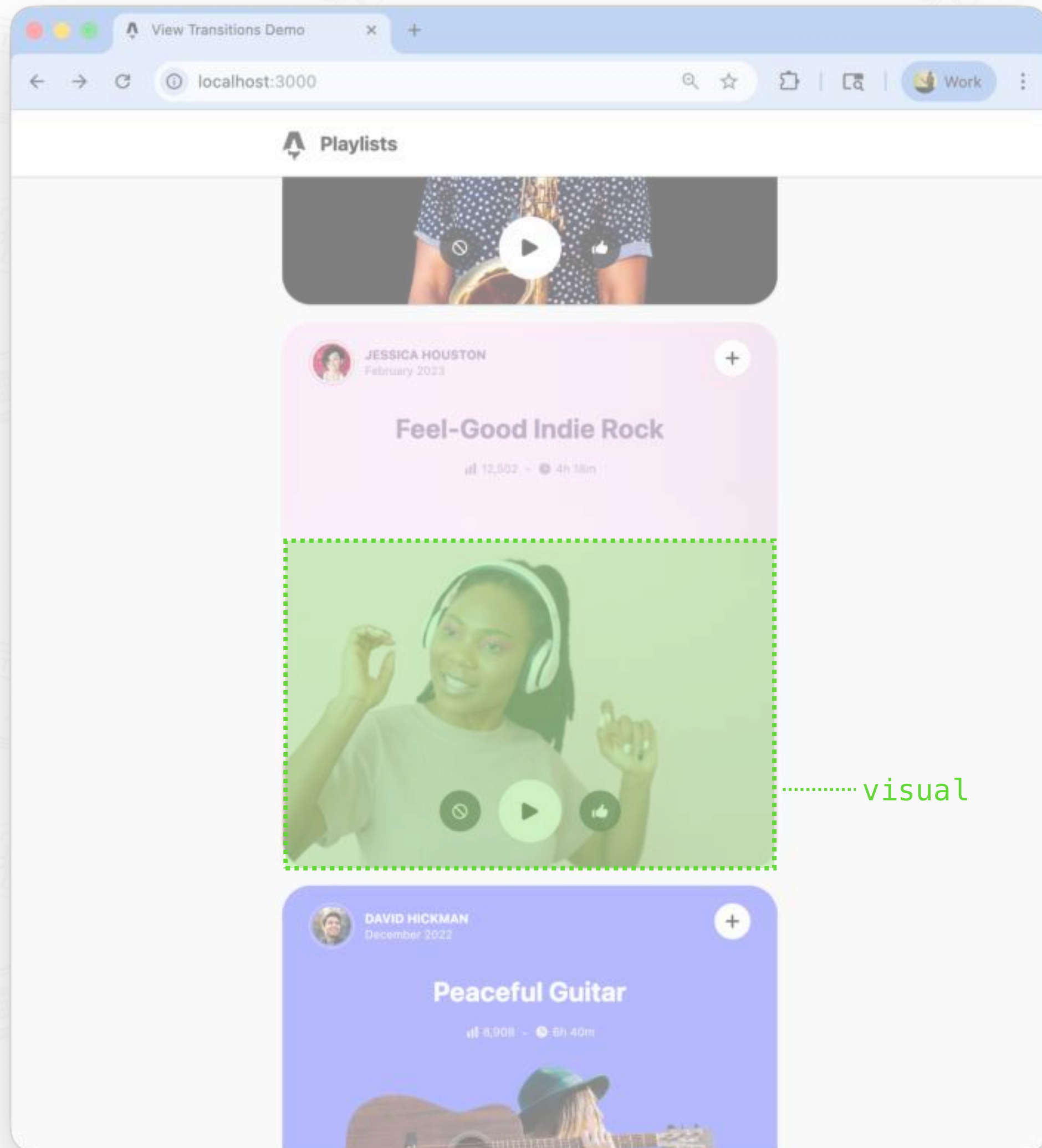
```
├─ ::view-transition-group(card)
  │   └─ ::view-transition-image-pair(card)
  │       ├── ::view-transition-old(card)
  │       └─ ::view-transition-new(card)
├─ ::view-transition-group(visual)
  │   └─ ::view-transition-image-pair(visual)
  │       ├── ::view-transition-old(visual)
  │       └─ ::view-transition-new(visual)
├─ ::view-transition-group(topbar)
  │   └─ ::view-transition-image-pair(topbar)
  │       └─ ::view-transition-old(topbar)
├─ ::view-transition-group(avatar)
  │   └─ ::view-transition-image-pair(avatar)
  │       ├── ::view-transition-old(avatar)
  │       └─ ::view-transition-new(avatar)
└─ :
```

**There is no new snapshot  
for this captured element**

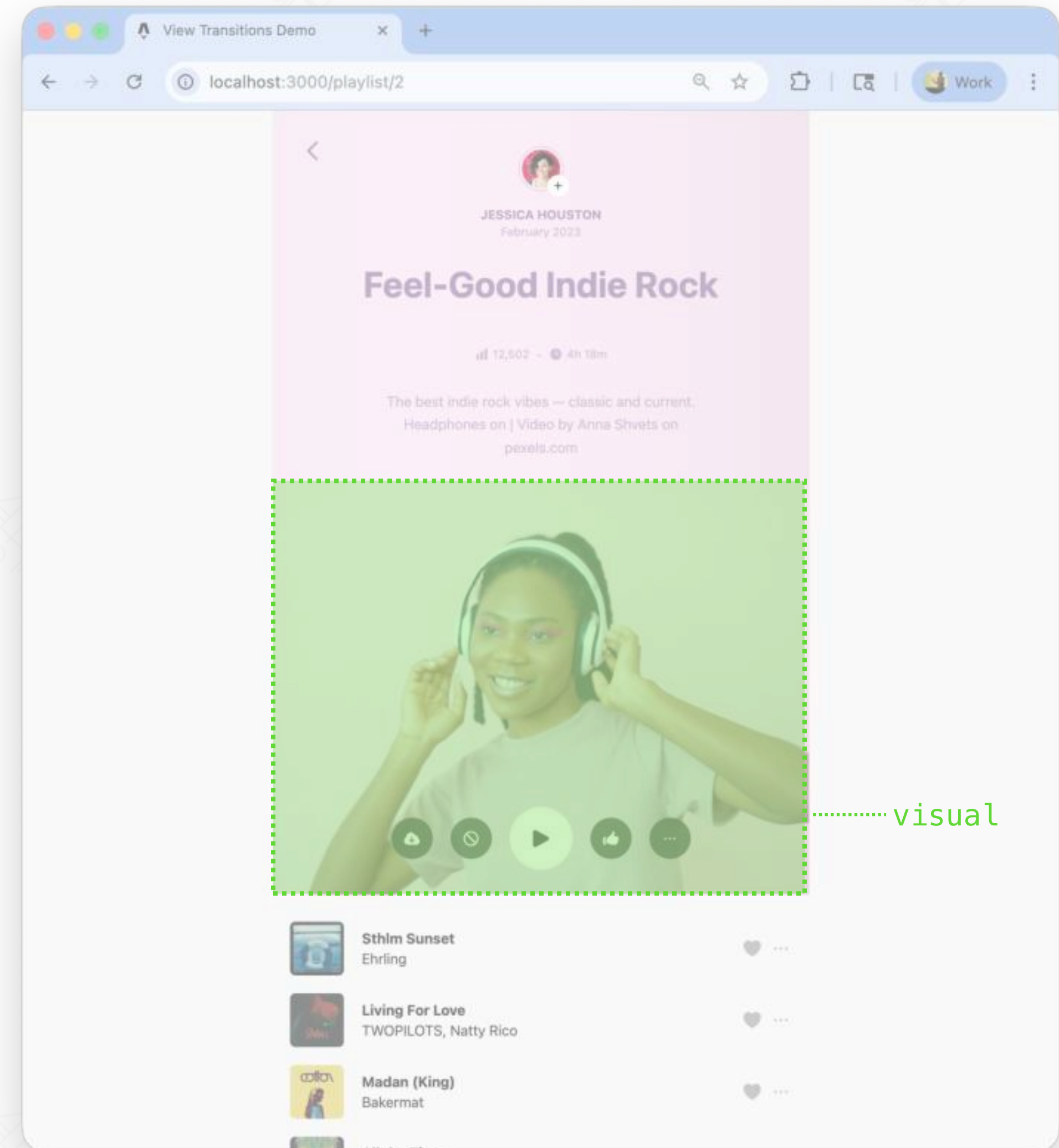


```
::view-transition-new(topbar):only-child {  
  animation: .25s cubic-bezier(.4,0,.2,1) both slide-down;  
}  
  
::view-transition-old(topbar):only-child {  
  animation: .25s cubic-bezier(.4,0,.2,1) both slide-up;  
}
```

*Custom CSS (Snippet)*



Overview



Detail

The old snapshot is stale



```
::view-transition-old(visual)
```

The new snapshot is live



```
::view-transition-new(visual)
```

Don't bother showing this one



```
::view-transition-old(visual)
```

Don't do the default fade in



```
::view-transition-new(visual)
```

```
::view-transition-old(visual) {  
    display: none; /* Hide the old snapshot */  
}  
  
::view-transition-new(visual) {  
    animation: none; /* Don't fade-in the new snapshot */  
}
```

# View Transitions

tl;dr

1. Identify and Name Elements
2. Trigger the View Transition
  - Snapshots Before
  - Pause Rendering
  - Update DOM
  - Snapshots After
  - Pseudo-tree construction
3. Customize the animations (*optional*)
  - Resume Rendering
  - Run the animations

Animate CSS Grid with View Tr × +

codepen.io/bramus/full/ZEqqqOX

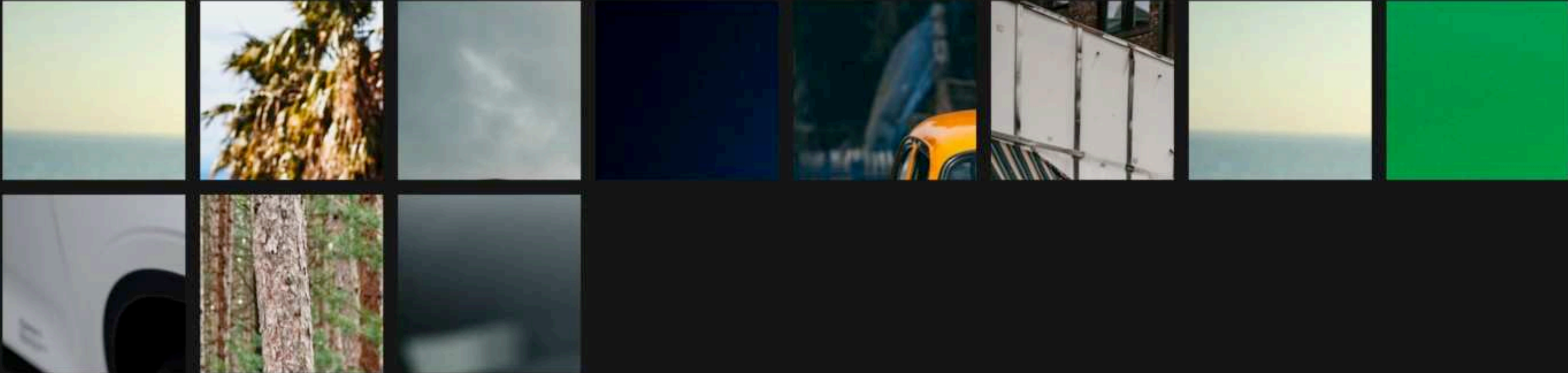
A Pen by Bramus **PRO**

### Animate CSS Grid with View Transitions (now with expanding squares ... and delays!)

This pen uses [animate-css-grid](#) [the View Transition API](#)

Updated version of [this pen](#), originally adapted from [this pen](#) by Alex

click a card to toggle the `grid-column` and `grid-row` properties on the card



<https://codepen.io/bramus/full/ZEqqqOX>

View Transitions like IsotopeJS x

codepen.io/argyleink/full/VwBKjwj

A Pen by Adam Argyle PRO + Follow

View Transitions like IsotopeJS

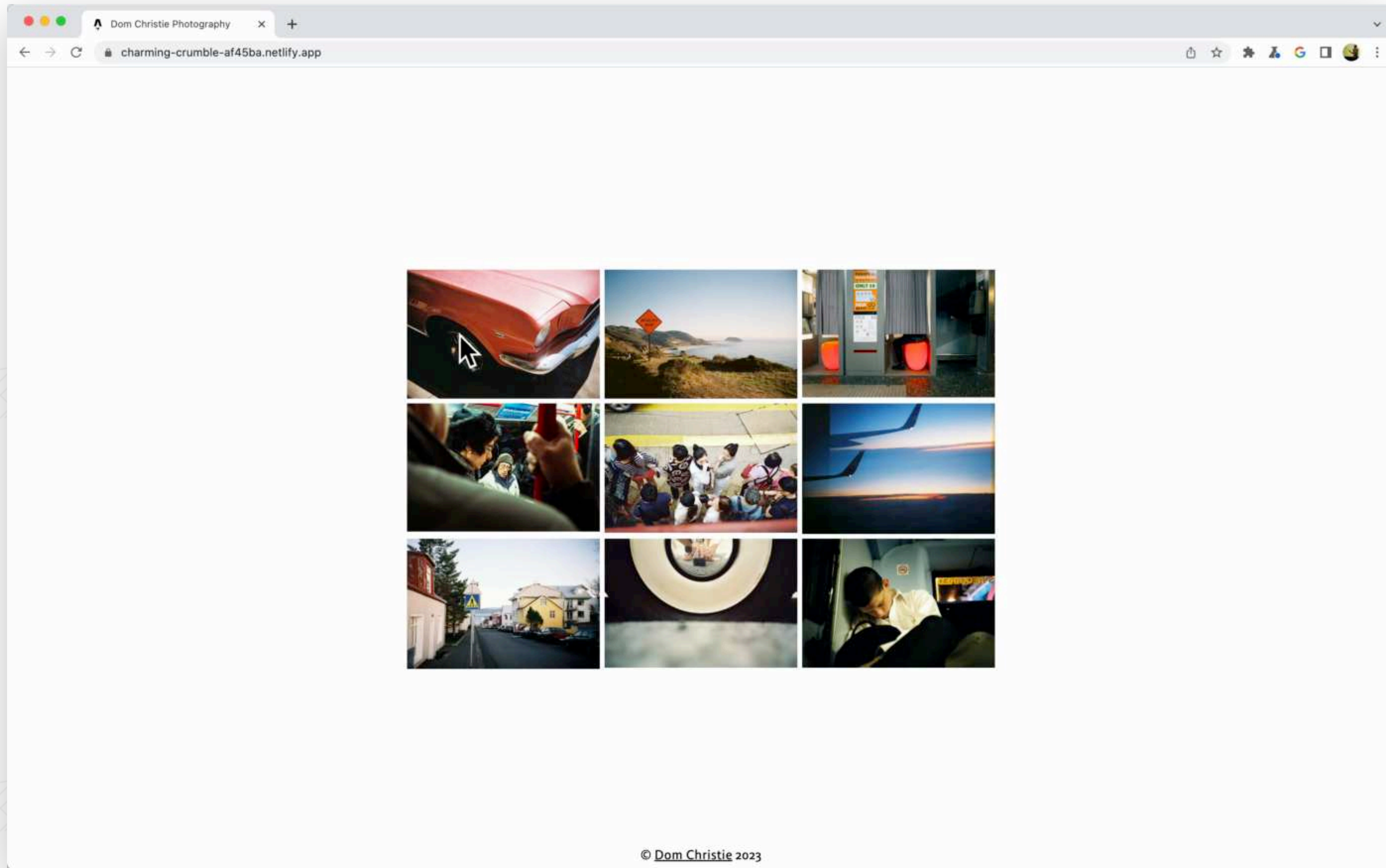
View Source Code

Filter:  Show All  Metal  Transition  -lum

Sort:  Name  Symbol  Number

51 <b>Sb</b> Antimony 121.76	18 <b>Ar</b> Argon 39.948	83 <b>Bi</b> Bismuth 208.980	48 <b>Cd</b> Cadmium 112.411	20 <b>Ca</b> Calcium 40.078	27 <b>Co</b> Cobalt 58.933	79 <b>Au</b> Gold 196.967	82 <b>Pb</b> Lead 207.2	80 <b>Hg</b> Mercury 200.59	7 <b>N</b> Nitrogen 14.007
94 <b>Pu</b> Plutonium 244	19 <b>K</b> Potassium 39.0983	75 <b>Re</b> Rhenium 186.207	11 <b>Na</b> Sodium 22.99	52 <b>Te</b> Tellurium 127.6	81 <b>Tl</b> Thallium 204.383	92 <b>U</b> Uranium 238.029	70 <b>Yb</b> Ytterbium 173.054		

<https://codepen.io/argyleink/full/VwBKjwj>



<https://charming-crumble-af45ba.netlify.app/> by [Dom Christie](#)

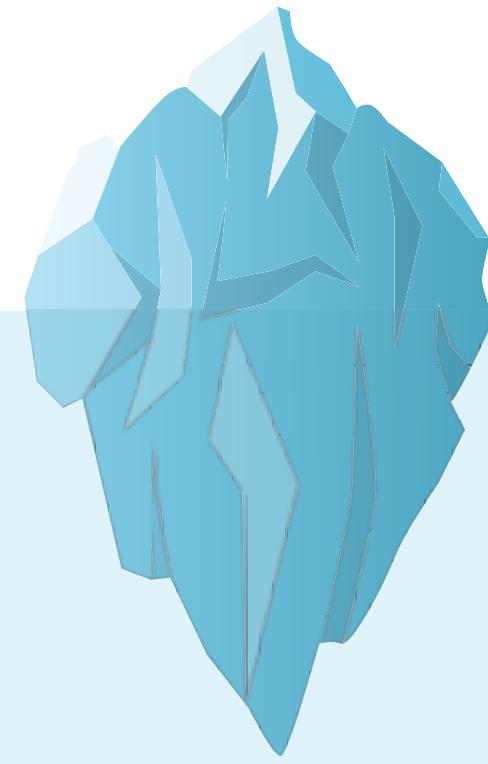
HTTP 203

http203-playlist.netlify.app

All Bramus Cassie Ada Surma Paul

<p>Scroll-linked animations HTTP 203</p> <p>2022-10-25</p>	<p>How rotate(0) fixed my transform animation HTTP 203</p> <p>2022-10-11</p>	<p>A love letter to DOMPoint and DOMMatrix HTTP 203</p> <p>2022-09-27</p>	<p>It's viewports all the way down HTTP 203</p> <p>2022-09-13</p>	<p>Magic tricks with the HTML parser HTTP 203</p> <p>2022-08-30</p>
<p>DOM ready events considered harmful HTTP 203</p> <p>2022-07-26</p>	<p>Demystifying SVG paths HTTP 203</p> <p>2022-07-12</p>	<p>FLIPing Animations HTTP 203</p> <p>2022-06-28</p>	<p>The History Navigation API HTTP 203</p> <p>2022-06-07</p>	<p>Samsung Internet HTTP 203</p> <p>2022-05-24</p>
<p>Is .CSS a bad idea? HTTP 203</p> <p>2022-05-03</p>	<p>Which key was pressed? HTTP 203</p> <p>2022-03-29</p>	<p>Building VR and AR experiences using HTML HTTP 203</p> <p>2022-03-15</p>	<p>Cross-fading DOM Elements HTTP 203</p> <p>2022-01-04</p>	<p>Generating your color palette in CSS HTTP 203</p> <p>2021-12-27</p>
<p>The ZOMBIE Dom HTTP 203</p>	<p>Bringing bounce and elastic easing to CSS HTTP 203</p>	<p>The gamepad API HTTP 203</p>	<p>Are SPAs better than MPAs? HTTP 203</p>	<p>Avoiding layout shift by putting the CSS in charge HTTP 203</p>

<https://http203-playlist.netlify.app/>



Active Types

The snapshotting process

Handle changes in aspect-ratio

Debug with Chrome DevTools

Promises

Dedupe animation styles with view-transition-class

...

Enter/Exit Only Transitions

Scoped Transitions

Framework integration

Construction of the pseudo tree

Animate the pseudos with JavaScript


Nested VT Groups

Navigation API

Skip a transition

Supercharge Web UX with View Transitions (2024.10.18 @ React Brussels)

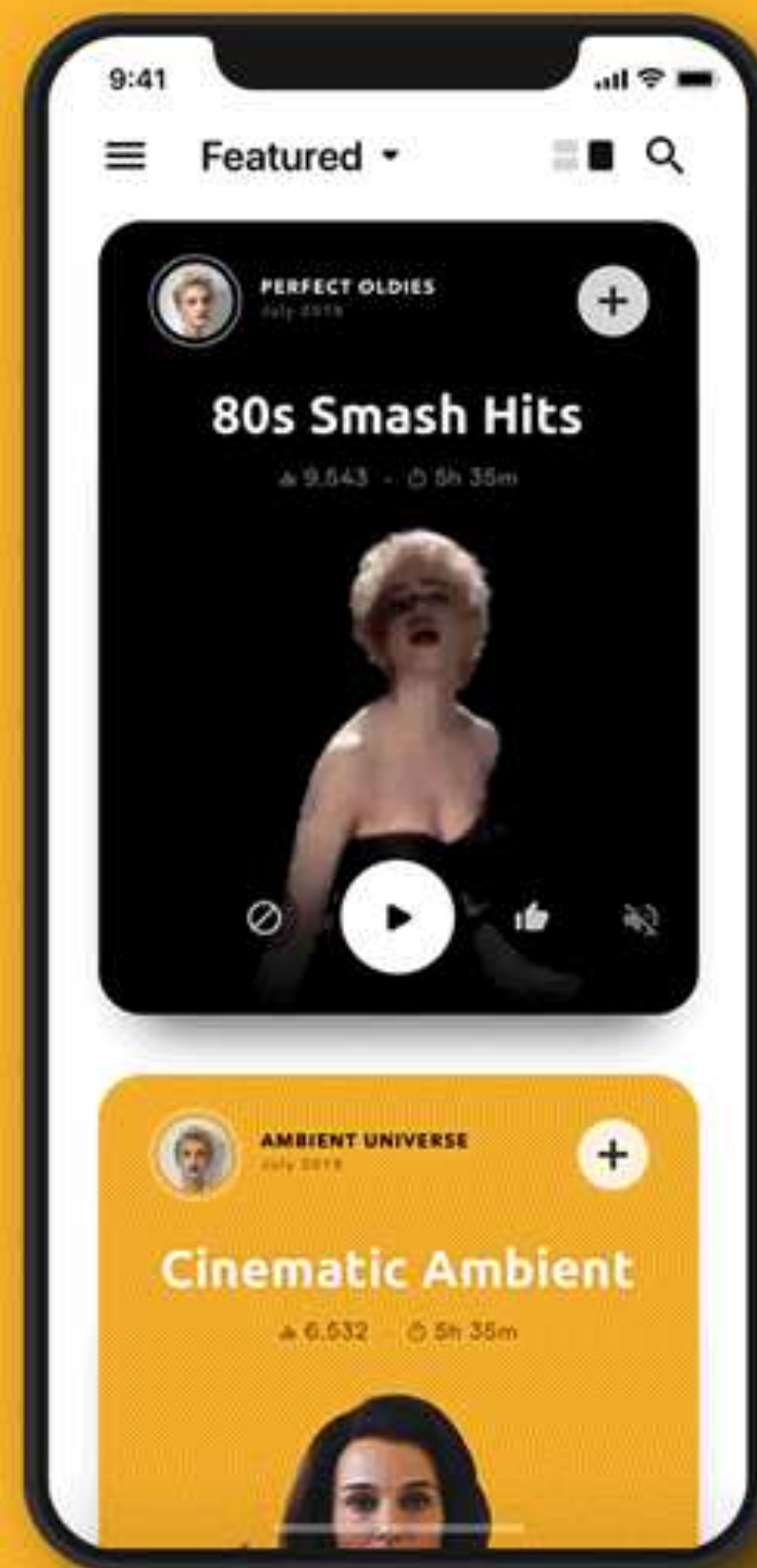
November 17, 2024 · Leave a comment



*Me, on stage.*

**INTRO** → Last month I spoke at [React Brussels](#) and gave a talk "Supercharge Web UX with View Transitions"

<https://brm.us/view-transitions-intro>

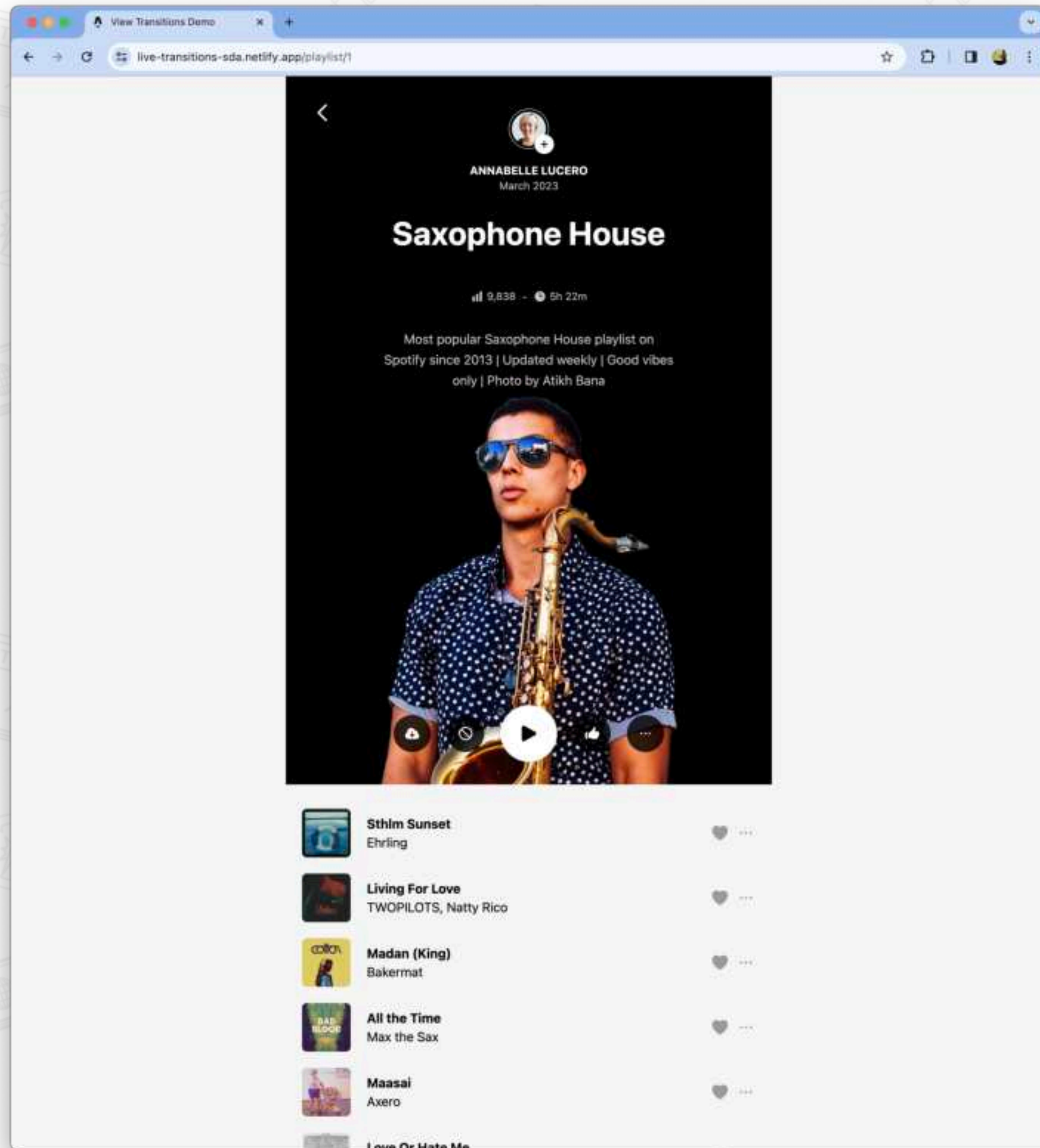


*Music Playlist App Interaction by Ehsan Rahimi*  
<https://dribbble.com/ehsancinematic>

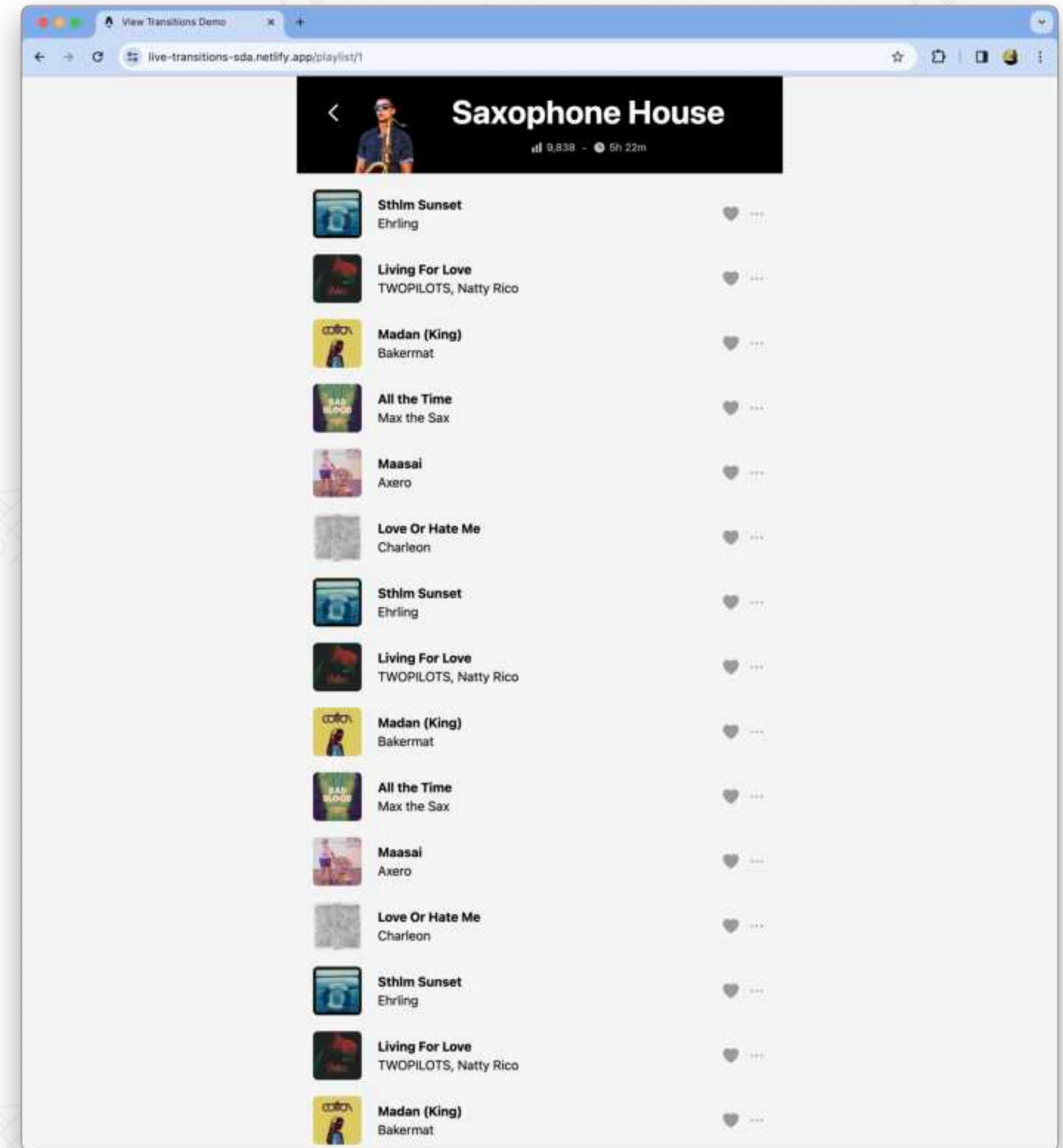


***“That’s a transition  
between two states!”***

*Me, on a bright day*



Large



Small

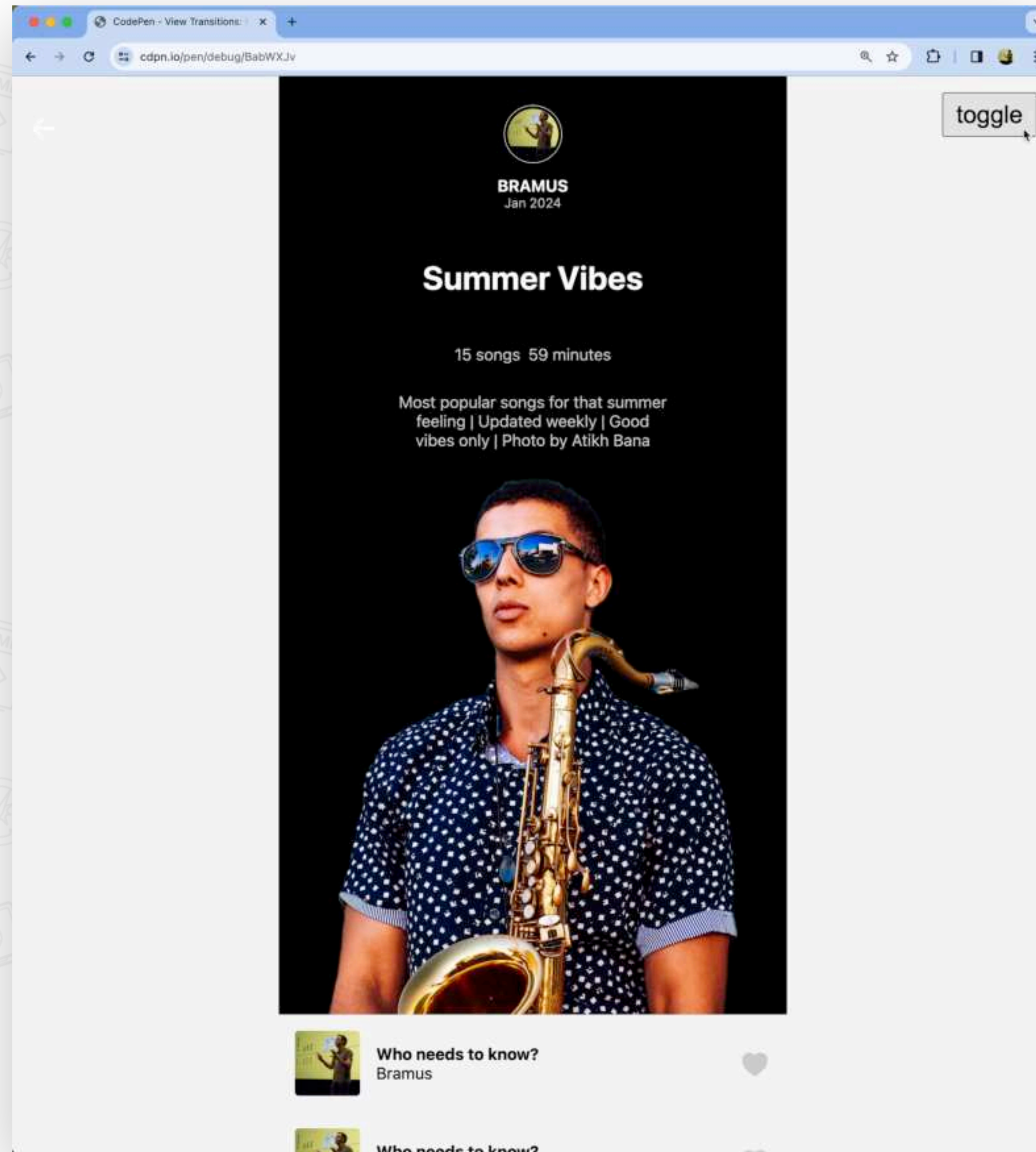
```
.card {  
  display: grid;  
  grid-template:  
    'meta' auto  
    'title' 100px  
    'moremeta' auto  
    'description' 80px  
    'image' auto / auto;  
  
  /* ... */  
}
```

```
.card.small {  
  grid-template:  
    'image title' 1fr  
    'image moremeta' 1fr / 80px auto;  
  
  /* ... */  
}
```

```
const toggleCard = () => {
  if (!document.startViewTransition) {
    $card.classList.toggle('small');
    return;
  }

  document.startViewTransition(() => {
    $card.classList.toggle('small');
  });
}

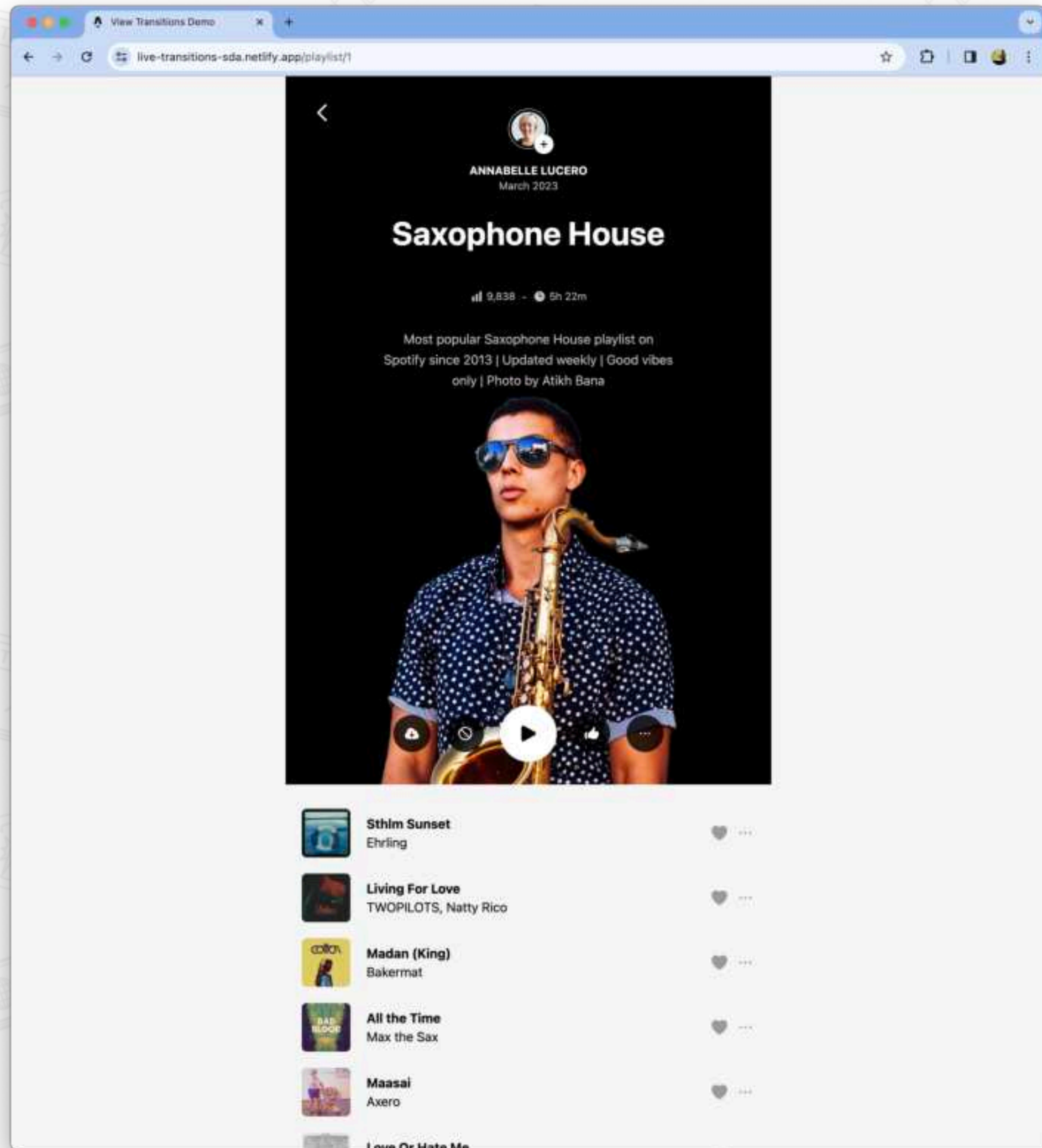
$button.addEventListener('click', async e => {
  e.preventDefault();
  toggleCard();
});
```



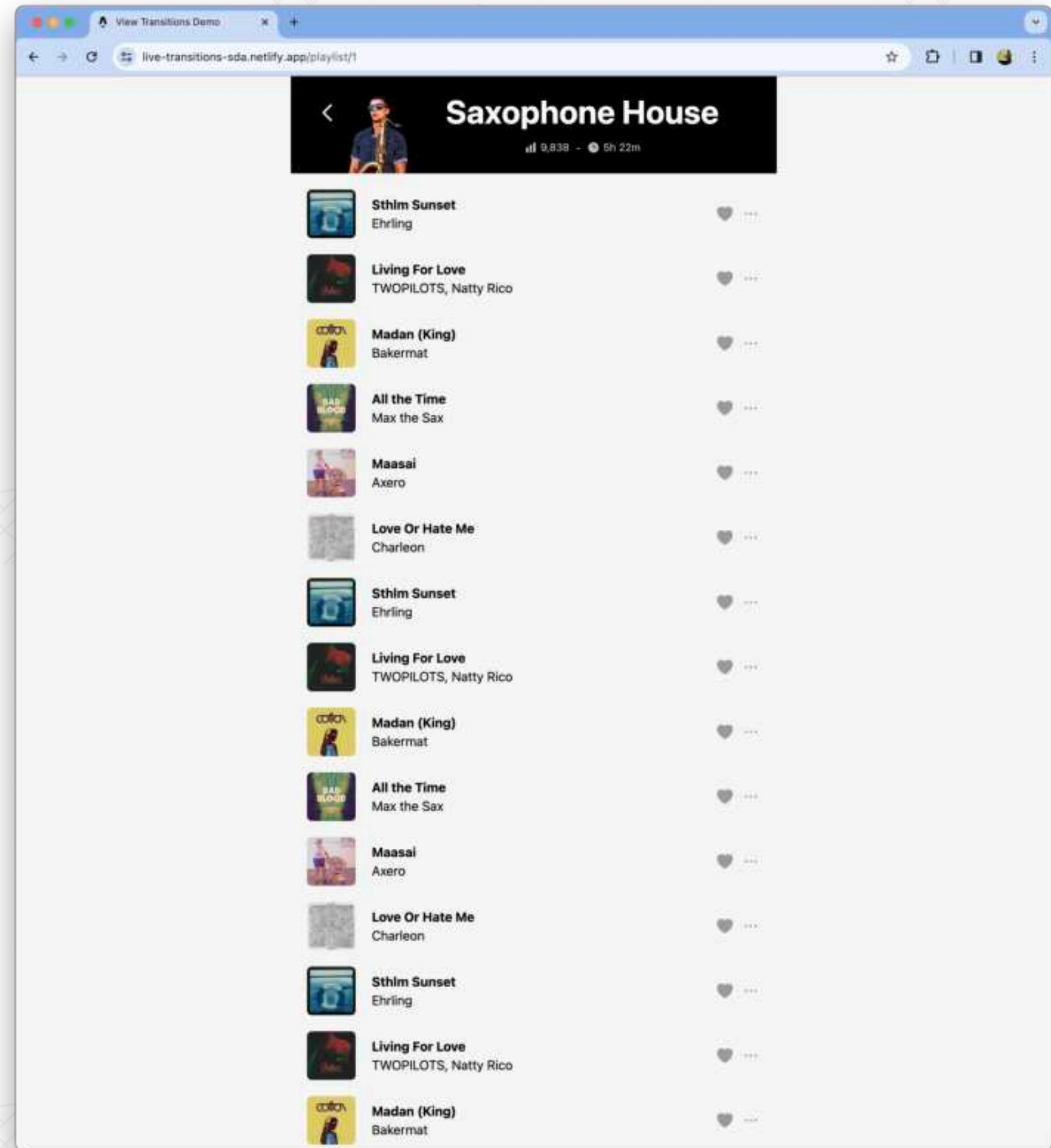
<https://codepen.io/bramus/full/BabWXJv>

# 1. Identify and name elements to transition

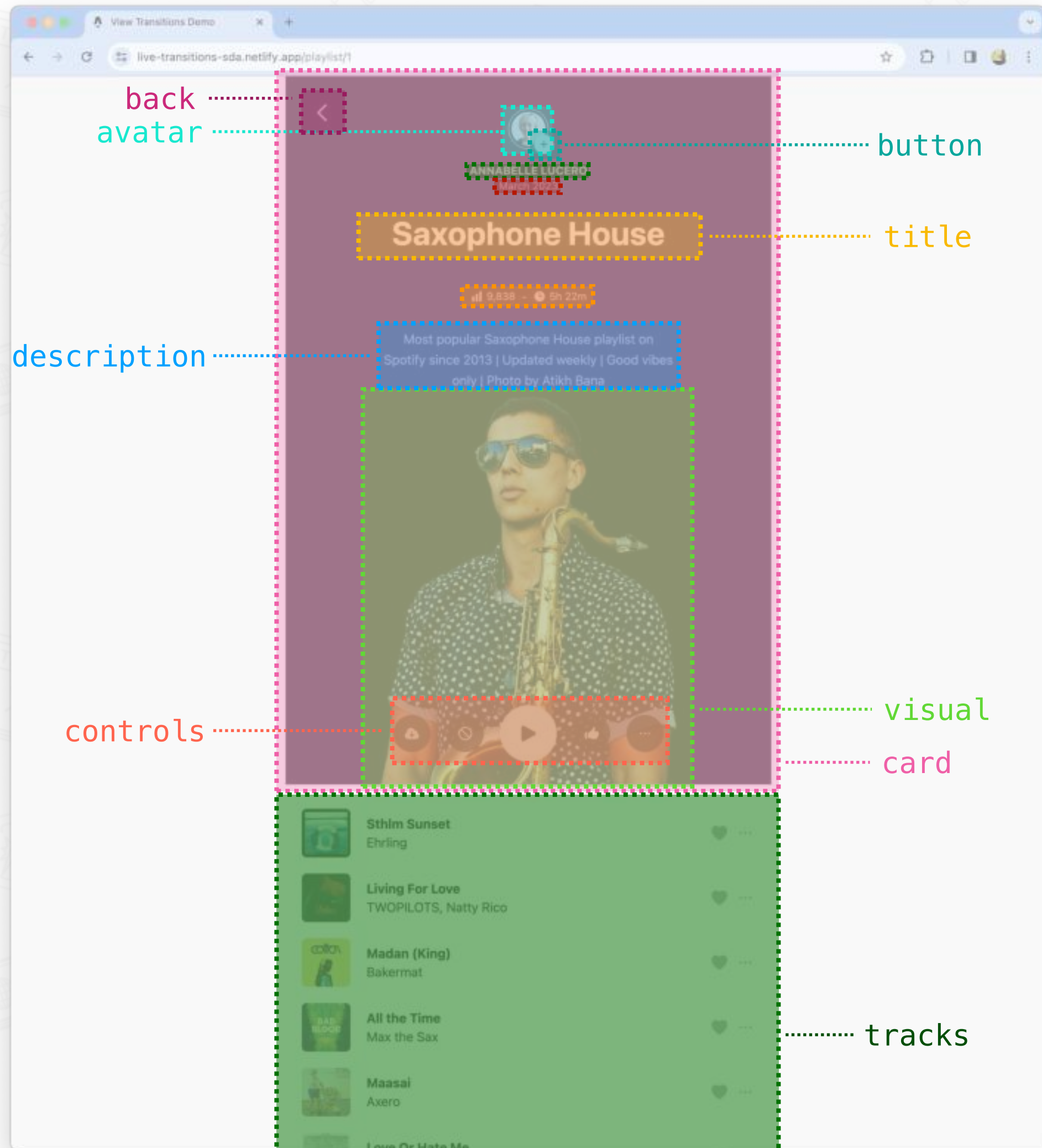
CSS `view-transition-name` property



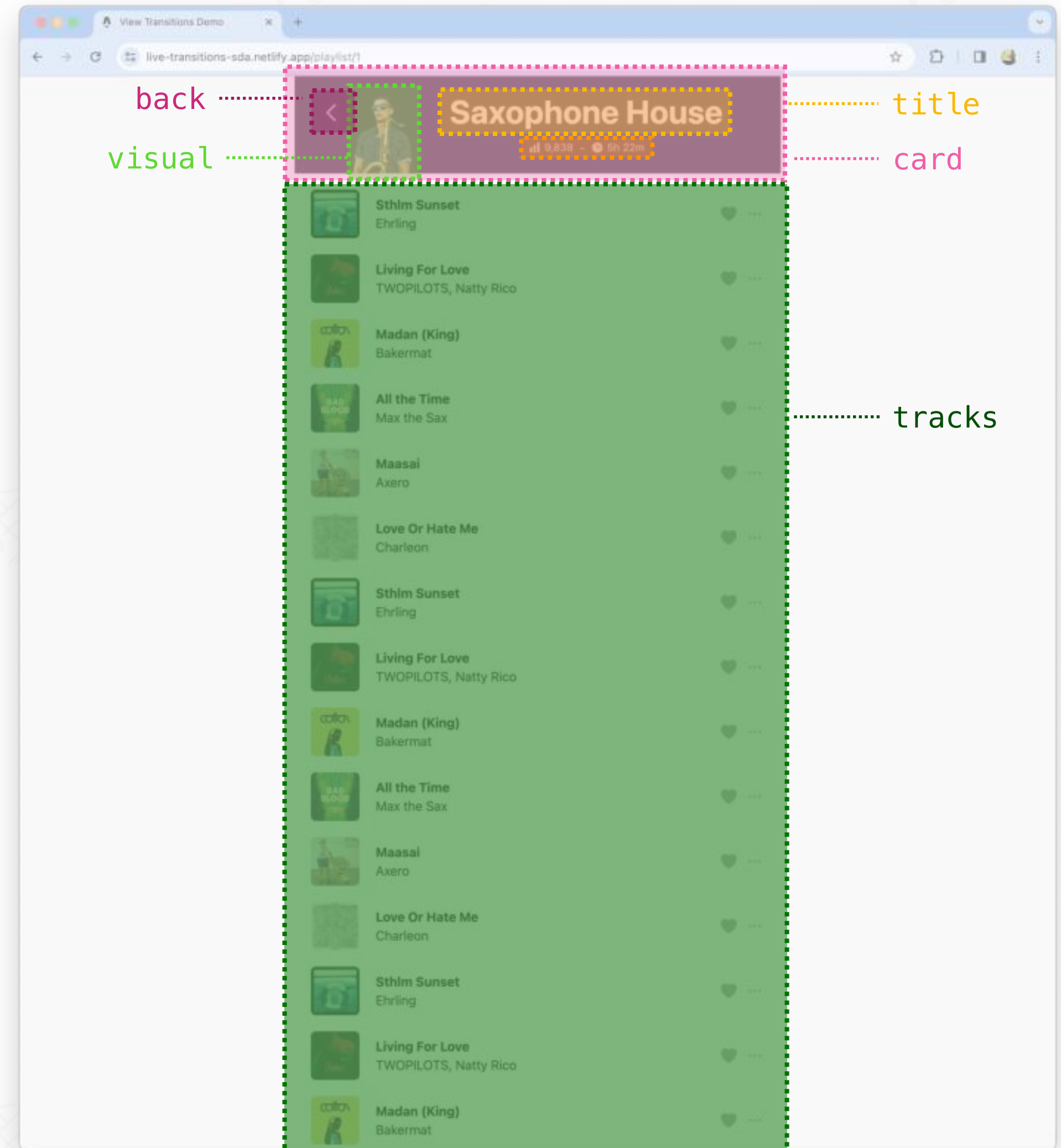
Large



Small



Large



Small

# 2. Trigger the View Transition

Same-Document or Cross-Document

```
const toggleCard = () => {
  if (!document.startViewTransition) {
    $card.classList.toggle('small');
    return;
  }

  document.startViewTransition(() => {
    $card.classList.toggle('small');
  });
}

$button.addEventListener('click', async e => {
  e.preventDefault();
  toggleCard();
});
```

# 3. Customize the animations

*(Optional)*

The image shows a web browser window displaying a music playlist page and its developer tools. The page content includes a back arrow, a profile picture for 'BRAMUS Jan 2024', the title 'Summer Vibes', and the text '15 songs 59 minutes'. Below this is a description: 'Most popular songs for that summer feeling | Updated weekly | Good vibes only | Photo by Atikh Bana'. A large image of a man playing a saxophone is featured. At the bottom, there are two song entries, each with a small profile picture and the text 'Who needs to know? Bramus'. The developer tools are open, showing the 'Elements' panel with the following HTML structure:

```
<!DOCTYPE html>
<html lang="en">
  <::view-transition
  <head> </head>
  <body>
    <main> </main>
    <button>toggle</button>
    <script id="rendered-js"> </script>
  </body>
</html>
```

The 'Animations' panel shows a timeline of various CSS transitions and delays:

- `::view-transition::view-transition-group(cover)` with `-ua-view-transition-group-anim-cover` (0-100%)
- `::view-transition-image-pair::view-transition-old(cover)` with `-ua-view-transition-fade-out` (0-100%)
- `::view-transition-image-pair::view-transition-old(cover)` with `-ua-mix-blend-mode-plus-lighter` (0-100%)
- `::view-transition::view-transition-group(title)` with `-ua-view-transition-group-anim-title` (0-100%)
- `::view-transition::view-transition-group(moremeta)` with `-ua-view-transition-group-anim-moremeta` (0-100%)
- `::view-transition-image-pair::view-transition-old(description)` with `slide-up` (0-100%)
- `::view-transition-image-pair::view-transition-old(description)` with `fade-out` (0-100%)
- `::view-transition-image-pair::view-transition-new(cover)` with `-ua-view-transition-fade-in` (0-100%)
- `::view-transition-image-pair::view-transition-new(cover)` with `-ua-mix-blend-mode-plus-lighter` (0-100%)
- `::view-transition-image-pair::view-transition-old(meta)` with `fade-out` (0-100%)
- `::view-transition::view-transition-group(tracks)` with `-ua-view-transition-group-anim-tracks` (0-100%)
- `::view-transition::view-transition-group(card)` with `-ua-view-transition-group-anim-card` (0-100%)

<https://codepen.io/bramus/full/BabWXJv>

but elements in that card only for a certain part of that entire duration. To achieve this, the durations and delays are expressed as fractions, which are then used in a calculation to get the actual duration in seconds.

```
*/  
::view-transition {  
  --vt-base-duration: 1s;  
  
  --vt-description-duration: 0.5;  
  --vt-description-delay: 0;  
  
  --vt-moremeta-duration: 0.65;  
  --vt-moremeta-delay: 0.2;  
  
  --vt-title-duration: 0.6;  
  --vt-title-delay: 0.2;  
  
  --vt-meta-duration: 0.5;  
  --vt-meta-delay: 0.3;  
}  
  
/* Apply base duration to all */  
::view-transition-group(*) {  
  animation-duration: var(--vt-base-duration);  
}
```

```
    --vt-meta-duration: 0.5;
    --vt-meta-delay: 0.3;
}

/* Apply base duration to all */
::view-transition-group(*) {
    animation-duration: var(--vt-base-duration);
}

/* Also inherit the delay from the group onto the child pseudos */
::view-transition-image-pair(*),
::view-transition-new(*),
::view-transition-old(*) {
    animation-delay: inherit;
}

/* Allow cursor to send events to underlying page while a VT is running */
::view-transition {
    pointer-events: none;
}

/* Some keyframes to use */
@keyframes slide-up { to { translate: 0 -100%; }}
@keyframes slide-down { from { translate: 0 -100%; }}
```

```
        animation-duration: var(--vt-base-duration);
    }

    /* Also inherit the delay from the group onto the child pseudos */
    ::view-transition-image-pair(*),
    ::view-transition-new(*),
    ::view-transition-old(*) {
        animation-delay: inherit;
    }

    /* Allow cursor to send events to underlying page while a VT is running */
    ::view-transition {
        pointer-events: none;
    }

    /* Some keyframes to use */
    @keyframes slide-up { to { translate: 0 -100%; }}
    @keyframes slide-down { from { translate: 0 -100%; }}
    @keyframes fade-out { to { opacity: 0; }}
    @keyframes fade-in { from { opacity: 0; }}

    /* Capture all these individual elements instead. Also, don't capture the root. */
    :root {
        view-transition-name: none;
    }
}
```

```
::view-transition-old(*) {
  animation-delay: inherit;
}

/* Allow cursor to send events to underlying page while a VT is running */
::view-transition {
  pointer-events: none;
}

/* Some keyframes to use */
@keyframes slide-up { to { translate: 0 -100%; }}
@keyframes slide-down { from { translate: 0 -100%; }}
@keyframes fade-out { to { opacity: 0; }}
@keyframes fade-in { from { opacity: 0; }}

/* Capture all these individual elements instead. Also, don't capture the root. */
:root {
  view-transition-name: none;
}
.card {
  view-transition-name: card;
}
.meta {
  view-transition-name: meta;
}
```

```
/* Capture all these individual elements instead. Also, don't capture the root. */
:root {
    view-transition-name: none;
}
.card {
    view-transition-name: card;
}
.meta {
    view-transition-name: meta;
}
.title {
    view-transition-name: title;
}
.moremeta {
    view-transition-name: moremeta;
}
.description {
    view-transition-name: description;
}
.cover {
    view-transition-name: cover;
}
.tracks {
    view-transition-name: tracks;
}
```

```
        view-transition-name: description;
    }
    .cover {
        view-transition-name: cover;
    }
    .tracks {
        view-transition-name: tracks;
    }

    /* The card itself should just shrink, not fade */
    ::view-transition-group(card) {
        overflow: clip;
    }
    ::view-transition-new(card),
    ::view-transition-old(card) {
        animation-name: none;
    }

    /* The title, moremeta, and tracks remain the same. Therefore, don't fade but
    immediately use the new snapshot */
    ::view-transition-new(title),
    ::view-transition-new(moremeta),
    ::view-transition-new(tracks) {
        animation-name: none;
    }
}
```

```
    overflow: clip;
  }
  ::view-transition-new(card),
  ::view-transition-old(card) {
    animation-name: none;
  }
}
```

*/\* The title, moremeta, and tracks remain the same. Therefore, don't fade but immediately use the new snapshot \*/*

```
  ::view-transition-new(title),
  ::view-transition-new(moremeta),
  ::view-transition-new(tracks) {
    animation-name: none;
  }
  ::view-transition-old(title),
  ::view-transition-old(moremeta),
  ::view-transition-old(tracks) {
    display: none;
  }
}
```

*/\* Slide and fade description. \*/*

```
  ::view-transition-old(description):only-child {
    animation-duration: calc(var(--vt-base-duration) * var(--vt-description-duration));
    animation-delay: calc(var(--vt-base-duration) * var(--vt-description-delay));
    animation-name: slide-up fade-out;
```

```
}  
::view-transition-old(title),  
::view-transition-old(moremeta),  
::view-transition-old(tracks) {  
    display: none;  
}  
  
/* Slide and fade description. */  
::view-transition-old(description):only-child {  
    animation-duration: calc(var(--vt-base-duration) * var(--vt-description-duration));  
    animation-delay: calc(var(--vt-base-duration) * var(--vt-description-delay));  
    animation-name: slide-up, fade-out;  
}  
::view-transition-new(description):only-child {  
    animation-duration: calc(var(--vt-base-duration) * var(--vt-description-duration));  
    animation-delay: calc(var(--vt-base-duration) * (1 - (var(--vt-description-delay) +  
var(--vt-description-duration))));  
    animation-name: slide-down, fade-in;  
}  
  
/* Set timing for various components */  
::view-transition-group(moremeta) {  
    animation-duration: calc(var(--vt-base-duration) * var(--vt-moremeta-duration));  
    animation-delay: calc(var(--vt-base-duration) * var(--vt-moremeta-delay));  
}
```


```
}
  :root:not(:has(div.small))::view-transition-group(meta) {
    animation-delay: calc(var(--vt-base-duration) * (1 - (var(--vt-meta-delay) + var(--vt-meta-duration))));
  }
  ::view-transition-old(meta):only-child {
    animation-name: fade-out;
  }
  ::view-transition-new(meta):only-child {
    animation-name: fade-in;
  }

  ::view-transition-group(tracks) {
    z-index: -1;
  }
}
```

CodePen - View Transitions: x +

cdpn.io/pen/debug/BabWXJv


toggle


  
**BRAMUS**  
Jan 2024


## Summer Vibes

15 songs 59 minutes

Most popular songs for that summer feeling | Updated weekly | Good vibes only | Photo by Atikh Bana

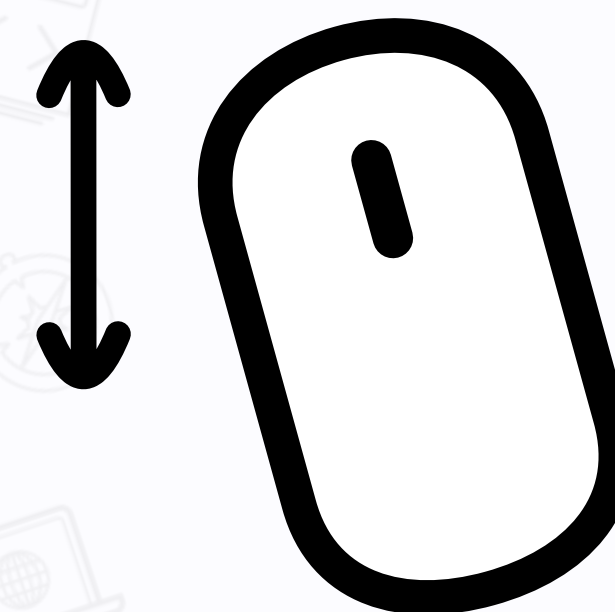


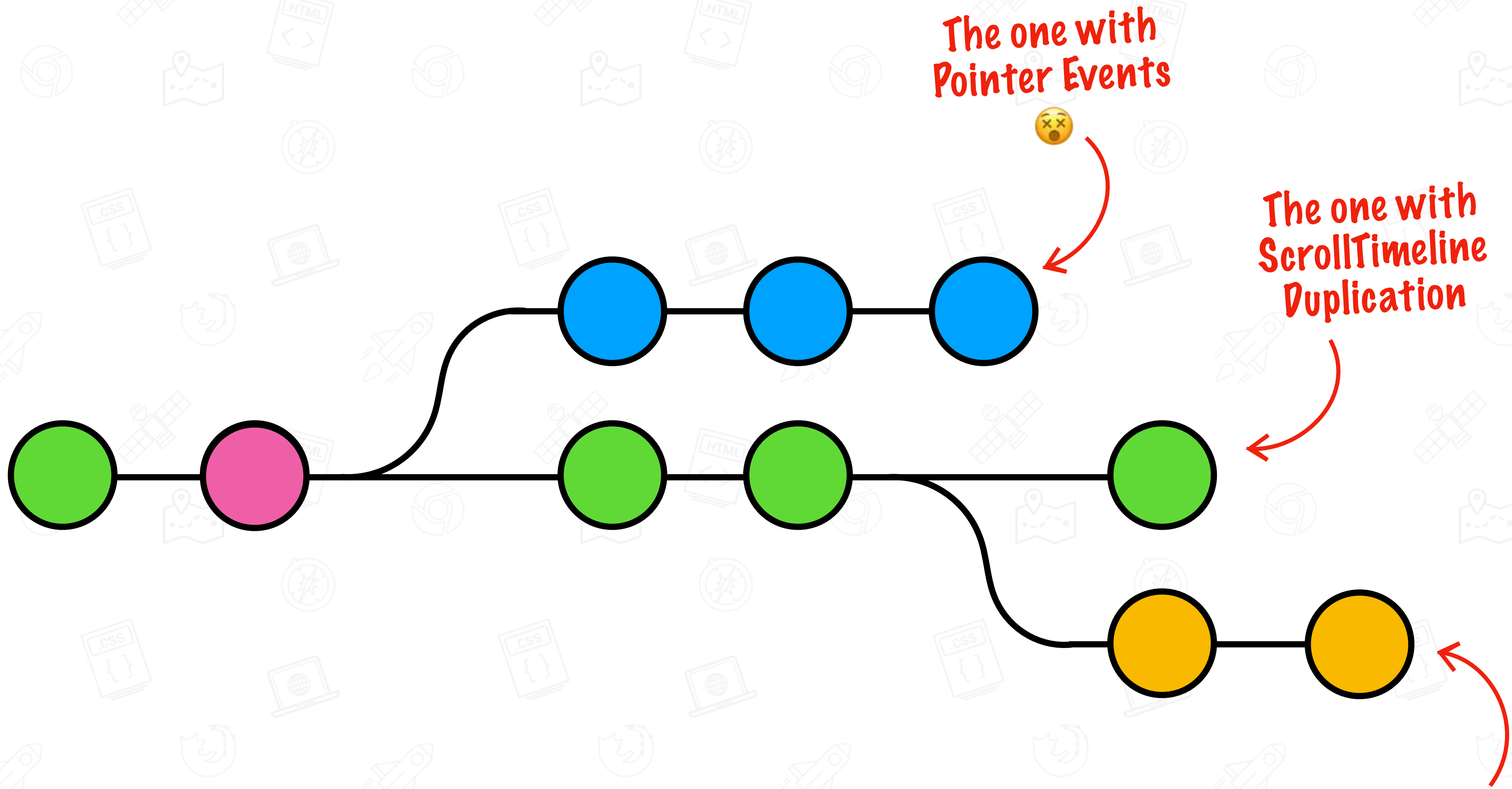
 Who needs to know?  
Bramus

 Who needs to know?

<https://codepen.io/bramus/full/BabWXJv>

***“How can I make this  
scroll-driven?”***





**The one with  
Pointer Events**

**The one with  
ScrollTimeline  
Duplication**

**The one that listens to  
the scroll event**

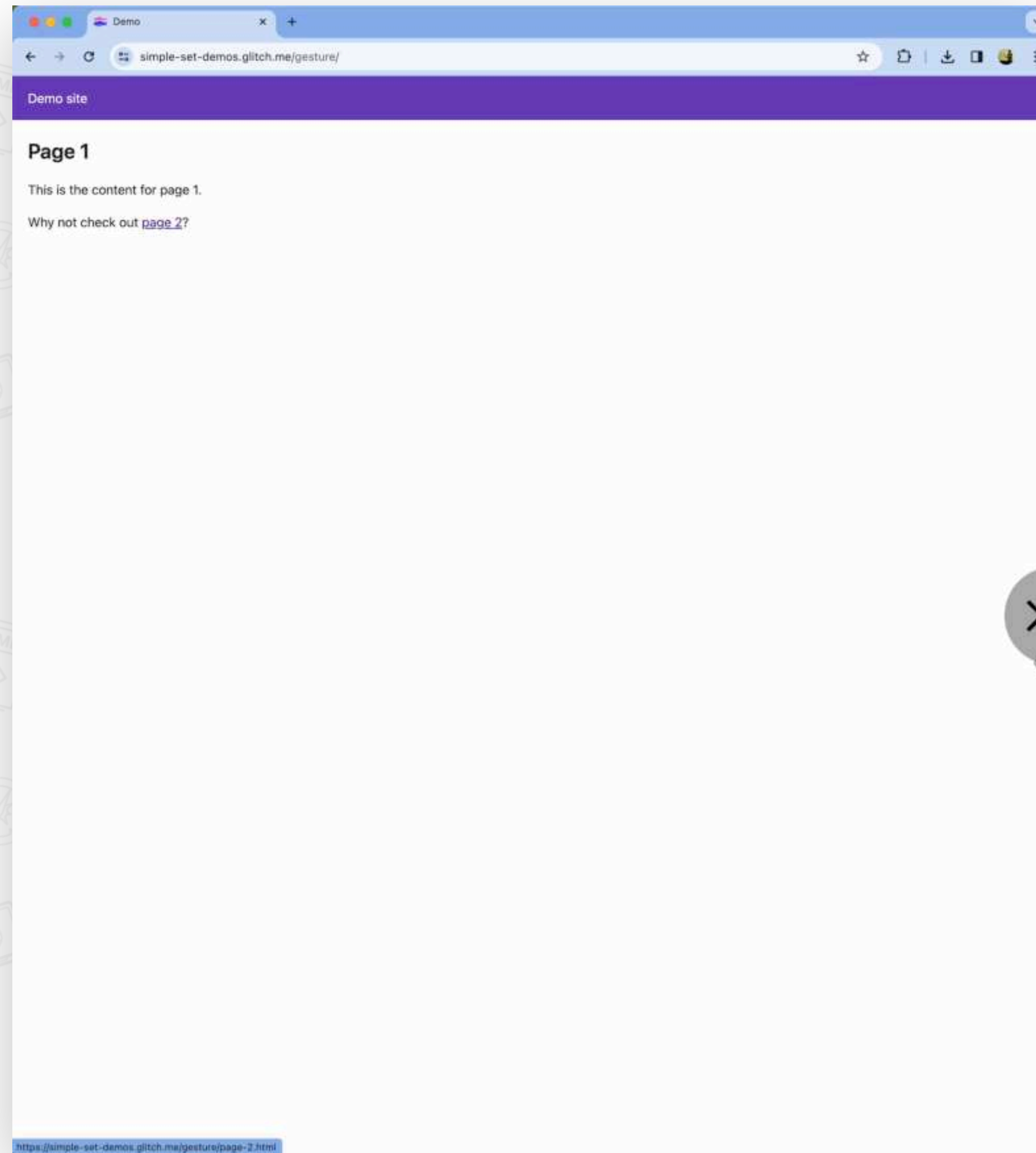
```
await activeViewTransition.ready;
```

```
activeAnimations = document.getAnimations().filter((anim) =>  
    anim.effect.target === document.documentElement &&  
    anim.effect.pseudoElement?.startsWith("::view-transition")  
);
```

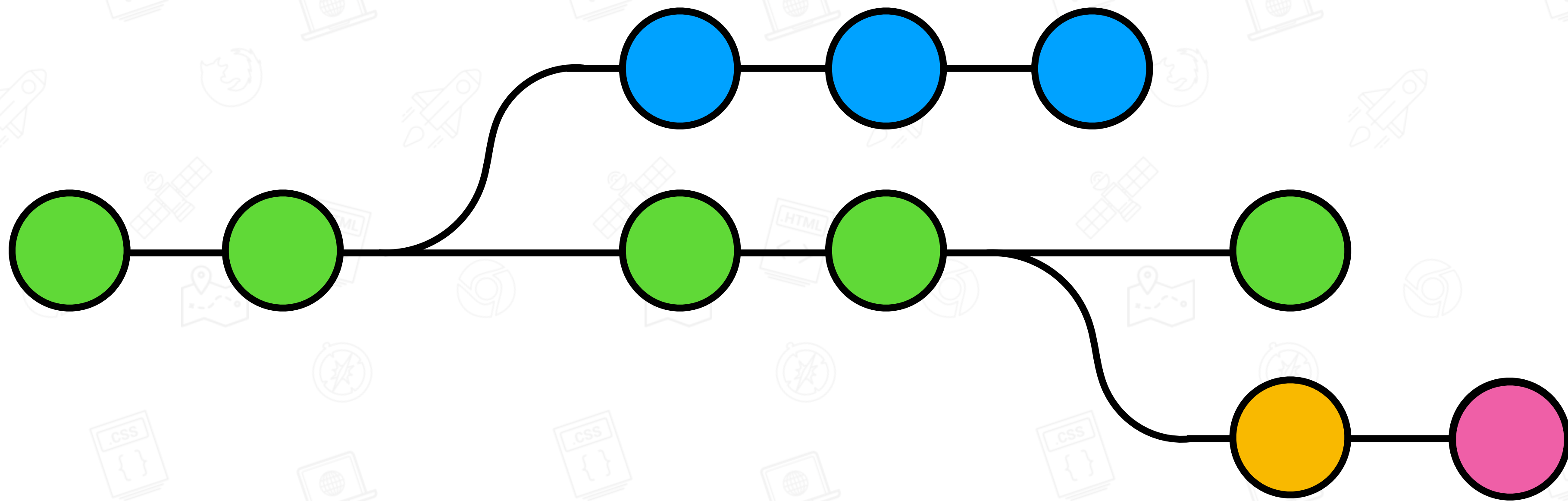
```
const transition = document.startViewTransition(() => {  
    // ...  
});  
  
// Wait for update callback to be done  
await transition.updateCallbackDone;  
console.log('updateCallbackDone');  
  
// Wait for the pseudo-elements + animations to be created  
await transition.ready;  
console.log('ready');  
  
// Wait for view transition to be done  
await transition.finished;  
console.log('finished');
```

```
await activeViewTransition.ready;
```

```
activeAnimations = document.getAnimations().filter((anim) =>  
    anim.effect.target === document.documentElement &&  
    anim.effect.pseudoElement?.startsWith("::view-transition")  
);
```



<https://simple-set-demos.glitch.me/gesture/> by Jake Archibald



**The one that listens to the scroll event**

```
const checkScrollPosition = async () => {
  if (
    (document.documentElement.scrollTop > triggerRange.start) &&
    (document.documentElement.scrollTop < triggerRange.end)
  ) {
    if (!activeViewTransition) {
      startViewTransitionAndPauseTheAnimations();
    } else {
      updateAnimations();
    }
  }
  else {
    if (activeViewTransition) {
      activeViewTransition.skipTransition();
    }
  }
}
window.addEventListener('scroll', checkScrollPosition);
```

```
const startViewTransitionAndPauseTheAnimations = async () => {
  const isReverse = document.querySelector('.small') ? true : false;

  activeViewTransition = document.startViewTransition(() => {
    document.querySelector('.card').classList.toggle('small');
  });

  await activeViewTransition.ready;
  activeAnimations = document.getAnimations().filter((anim) =>
    anim.effect.target === document.documentElement &&
    anim.effect.pseudoElement?.startsWith("::view-transition")
  );

  for (const anim of activeAnimations) {
    if (isReverse) anim.reverse();
    anim.pause();
  }

  // .. (cleanup)
};
```

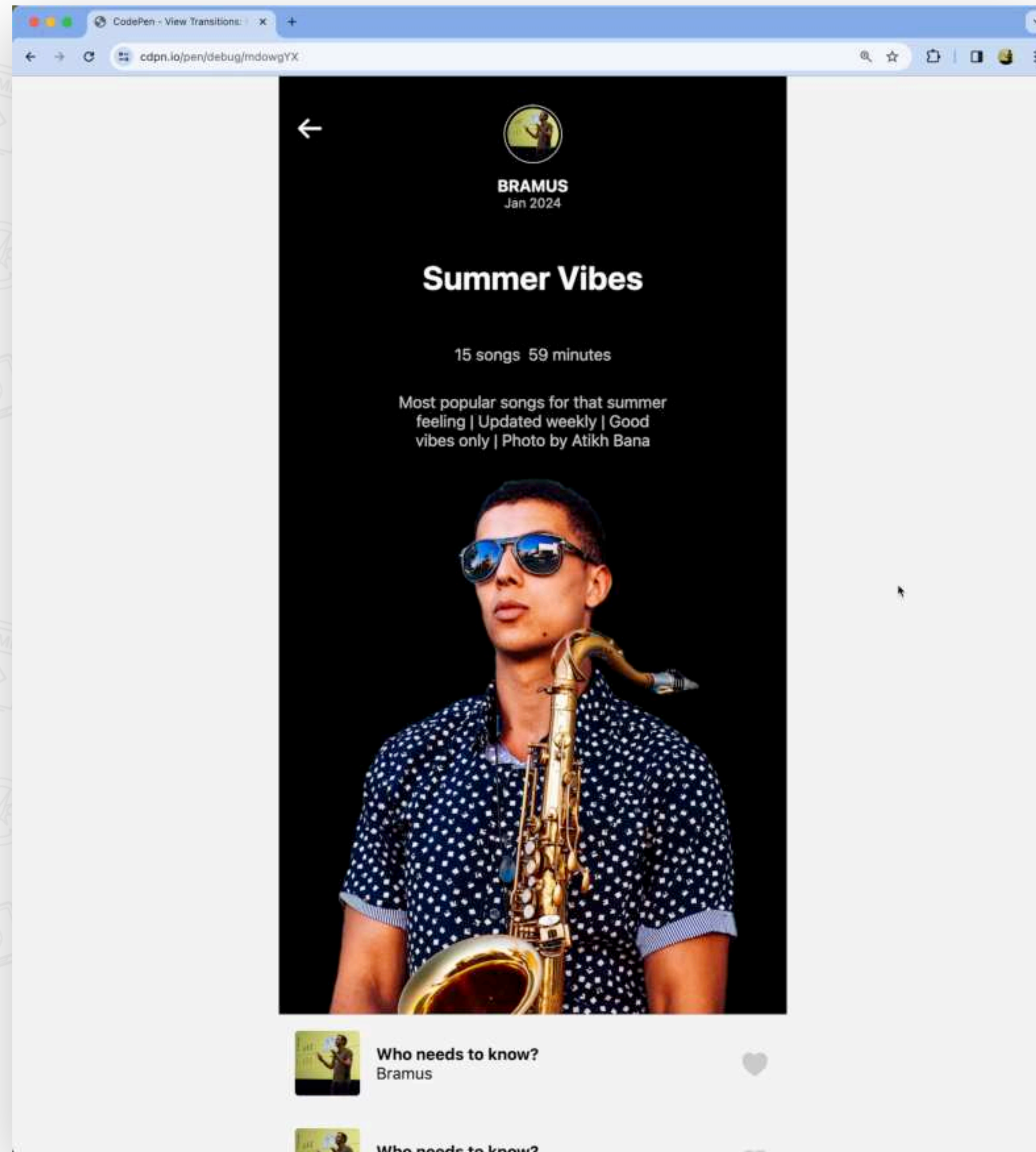
```
const updateAnimations = () => {
  if (!activeAnimations.length) return;

  const scrollProgress = currentScrollDistance / targetScrollDistance;

  const currentTime = clamp(0, scrollProgress * baseDuration, baseDuration);

  for (const animation of activeAnimations) {
    if (animation.playbackRate === -1) {
      animation.currentTime = baseDuration - currentTime;
    } else {
      animation.currentTime = currentTime;
    }
  }
}
```

```
const startViewTransitionAndPauseTheAnimations = async () => {  
  // ... (start VT + pause animations)  
  
  await activeViewTransition.finished;  
  if (document.documentElement.scrollTop > triggerRange.start) {  
    document.querySelector('.card').classList.add('small');  
  } else {  
    document.querySelector('.card').classList.remove('small');  
  }  
  
  activeViewTransition = null;  
  
};
```



<https://codepen.io/bramus/full/mdowgYX>

```
activeAnimations = document.getAnimations().filter((anim) =>
  anim.effect.target === document.documentElement &&
  anim.effect.pseudoElement?.startsWith("::view-transition")
);
```

<https://brm.us/view-transition-getanimations>

***“The keyframes are not performant enough!”***

*Your colleague with DevTools*

Regular and Better performin... x +

codepen.io/bramus/full/MYgNVay

Bramus PRO Animation CSS ...

Elements Console Sources Performance Memory >> 40 1

codepen.io #1 Screenshots Memory

Dim 3rd parties

1,000 ms 2,000 ms 3,000 ms 4,000 ms 5,000 ms 6,000 ms 7,000 ms CPU NET

1,000 ms 4,500 ms 5,000 ms 5,500 ms 6,000 ms

Animations

- ua-view-transition-group-anim-box
- ua-view-transition-fade-in
- ua-mix-blend-mode-plus-lighter
- ua-view-transition-fade-out
- ua-mix-blend-mode-plus-lighter

Interactions

- Main — https://codepen.io/bramus/full/oNQNQdb
- Frame — https://cdpn.io/bramus/fullpage/oNQNQdb?nocache=true&view=fullpage

Summary Bottom-up Call tree Event log

Duration 2.00 s

Animating -ua-view-transition-group-anim-box

Related node ::view-transition-group(box)

**Compositing failed** Unsupported CSS properties: height & width

Trace event

```
{ "args": { "data": { "beginEvent": { "args": { "data": { "displayName": "-ua-view-transition-group-anim-box", "id": "264",
```

<https://codepen.io/bramus/full/oNQNQdb>

```
[
  {
    "offset": 0,
    "easing": "ease",
    "composite": "auto",
    "height": "100px",
    "width": "100px",
    "backdropFilter": "none",
    "transform": "matrix(1, 0, 0, 1, 0, 0)",
  },
  {
    "offset": 1,
    "easing": "ease",
    "composite": "replace",
    "height": "100px",
    "width": "100px",
    "backdropFilter": "none",
    "transform": "matrix(1, 0, 0, 1, 0, 0)",
  }
]
```

```
[
  {
    "offset": 0,
    "easing": "ease",
    "composite": "auto",
    "height": "100px",
    "width": "100px",
    "backdropFilter": "none",
    "transform": "matrix(1, 0, 0, 1, 0, 0)",
  },
  {
    "offset": 1,
    "easing": "ease",
    "composite": "replace",
    "height": "100px",
    "width": "100px",
    "backdropFilter": "none",
    "transform": "matrix(1, 0, 0, 1, 0, 0)",
  }
]
```

🤩 Chrome has optimizations in place for when these don't change.

```
const keyframes = boxGroupAnimation.effect.getKeyframes();
```

```
delete keyframes[0].width;
```

```
delete keyframes[1].width;
```

```
delete keyframes[0].height;
```

```
delete keyframes[1].height;
```

```
boxGroupAnimation.effect.setKeyframes(keyframes);
```

**⚠ Only works when  
the width and height  
remain the same**

**🐛 Returns incorrect  
keyframes in Chrome <137**

**[crbug/387030974](https://crbug.com/387030974)**

*Quick Fix*

***“Let’s roll our own FLIP”***

*Me, going down the rabbit hole*

```
// Trigger VT
const t = document.startViewTransition(moveTheBox);
await t.ready;

// Get the group's animation
const boxGroupAnimation = getAnimation("::view-transition-group(box)");

// Get the keyframes
const boxGroupKeyframes = boxGroupAnimation.effect.getKeyframes();
```

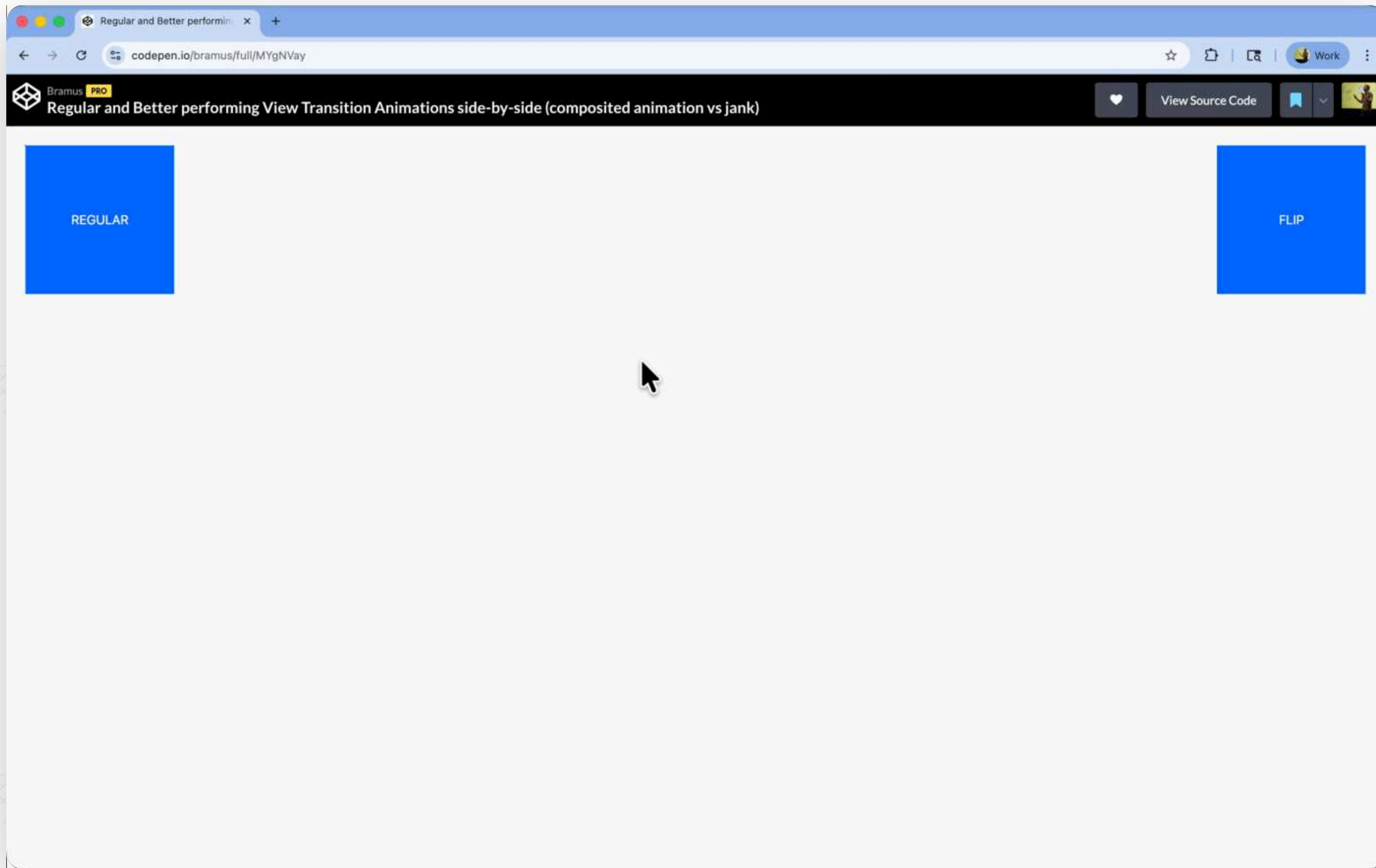
```
[
  {
    "offset": 0,
    "easing": "ease",
    "composite": "auto",
    "height": "100px",
    "width": "100px",
    "backdropFilter": "none",
    "transform": "matrix(1, 0, 0, 1, 0, 0)",
  },
  {
    "offset": 1,
    "easing": "ease",
    "composite": "replace",
    "height": "100px",
    "width": "100px",
    "backdropFilter": "none",
    "transform": "matrix(1, 0, 0, 1, 0, 0)",
  }
]
```

```
// Build rect to represent the old position + size
// based off of the "from" transform value
const oldMatrix = new DOMMatrix(boxGroupKeyframes[0].transform);
const rectBefore = {
  width: boxGroupKeyframes[0].width.split('px')[0],
  height: boxGroupKeyframes[0].height.split('px')[0],
  left: oldMatrix.e,
  top: oldMatrix.f,
};
```

```
// Build rect to represent the new position + size
// based off of the “to” transform value
// ⚠ Needs workaround for Chrome < 137
const newMatrix = new DOMMatrix(boxGroupKeyframes[1].transform);
const rectAfter = {
  width: boxGroupKeyframes[1].width.split('px')[0],
  height: boxGroupKeyframes[1].height.split('px')[0],
  left: newMatrix.e,
  top: newMatrix.f,
};
```

```
const transform = [  
  `translate(${rectBefore.left}px,${rectBefore.top}px)  
  scaleX(${rectBefore.width / rectAfter.width})  
  scaleY(${rectBefore.height / rectAfter.height})`,  
  
  `translate(${rectAfter.left}px,${rectAfter.top}px)  
  scaleX(1)  
  scaleY(1)`,  
];
```

```
const flipKeyframes = {  
  transform,  
  transformOrigin: ['0% 0%', '0% 0%'],  
  easing: "ease",  
};
```



<https://codepen.io/bramus/full/MYgNVay>

```
// Build rect to represent the new position + size
// based off of the “to” transform value
// ⚠ Needs workaround for Chrome < 137
const newMatrix = new DOMMatrix(boxGroupKeyframes[1].transform);
const rectAfter = {
  width: boxGroupKeyframes[1].width.split('px')[0],
  height: boxGroupKeyframes[1].height.split('px')[0],
  left: newMatrix.e,
  top: newMatrix.f,
};
```



```
if (isBuggyChromium(boxGroupKeyframes)) {
  boxGroupAnimation.currentTime =
    boxGroupAnimation.effect.getTiming().duration;

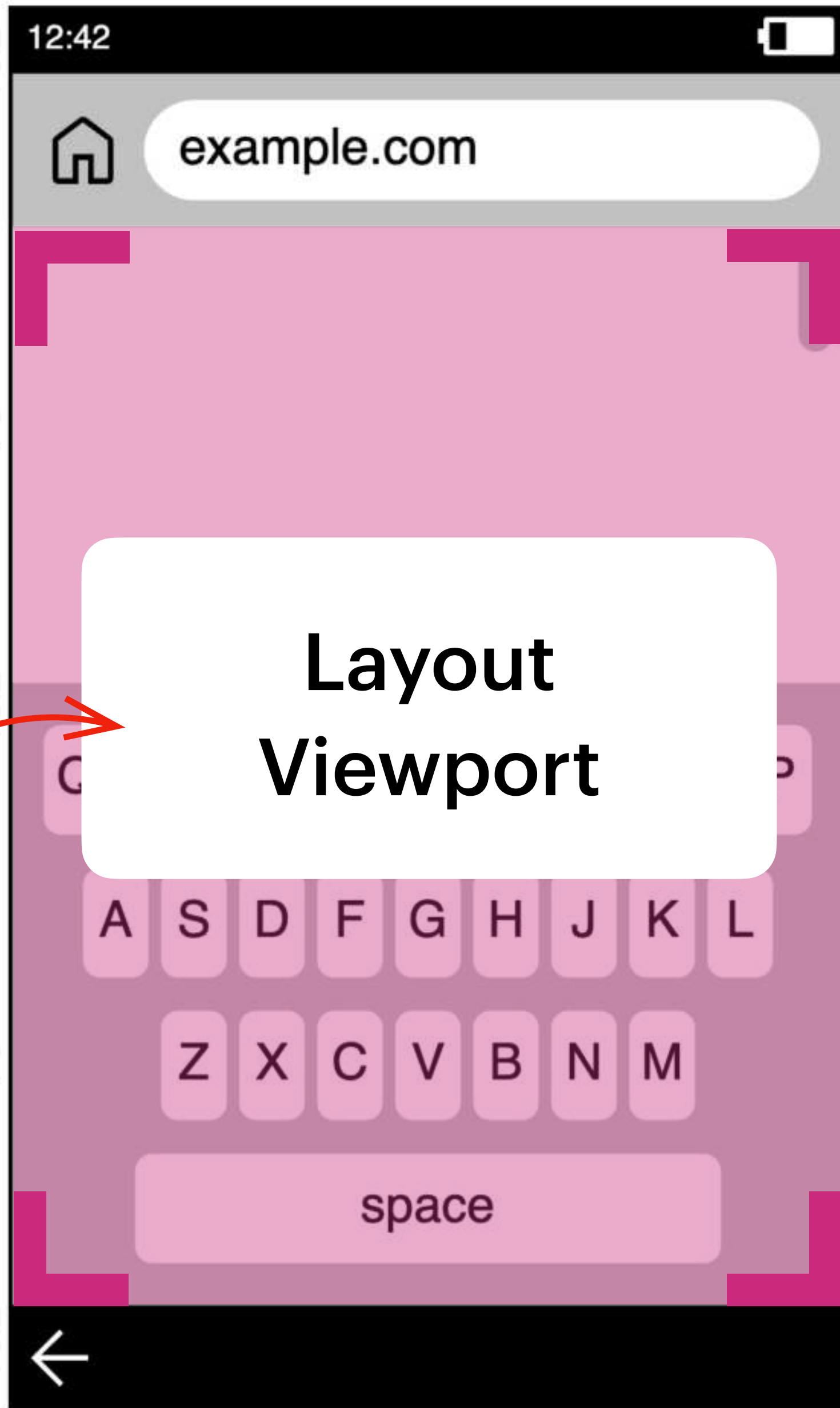
  const newStyles = window.getComputedStyle(
    document.documentElement, '::view-transition-group(box)');
  const newMatrix = new DOMMatrix(newStyles.transform);

  rectAfter = {
    width: newStyles.width.split('px')[0],
    height: newStyles.height.split('px')[0],
    left: newMatrix.e,
    top: newMatrix.f,
  };

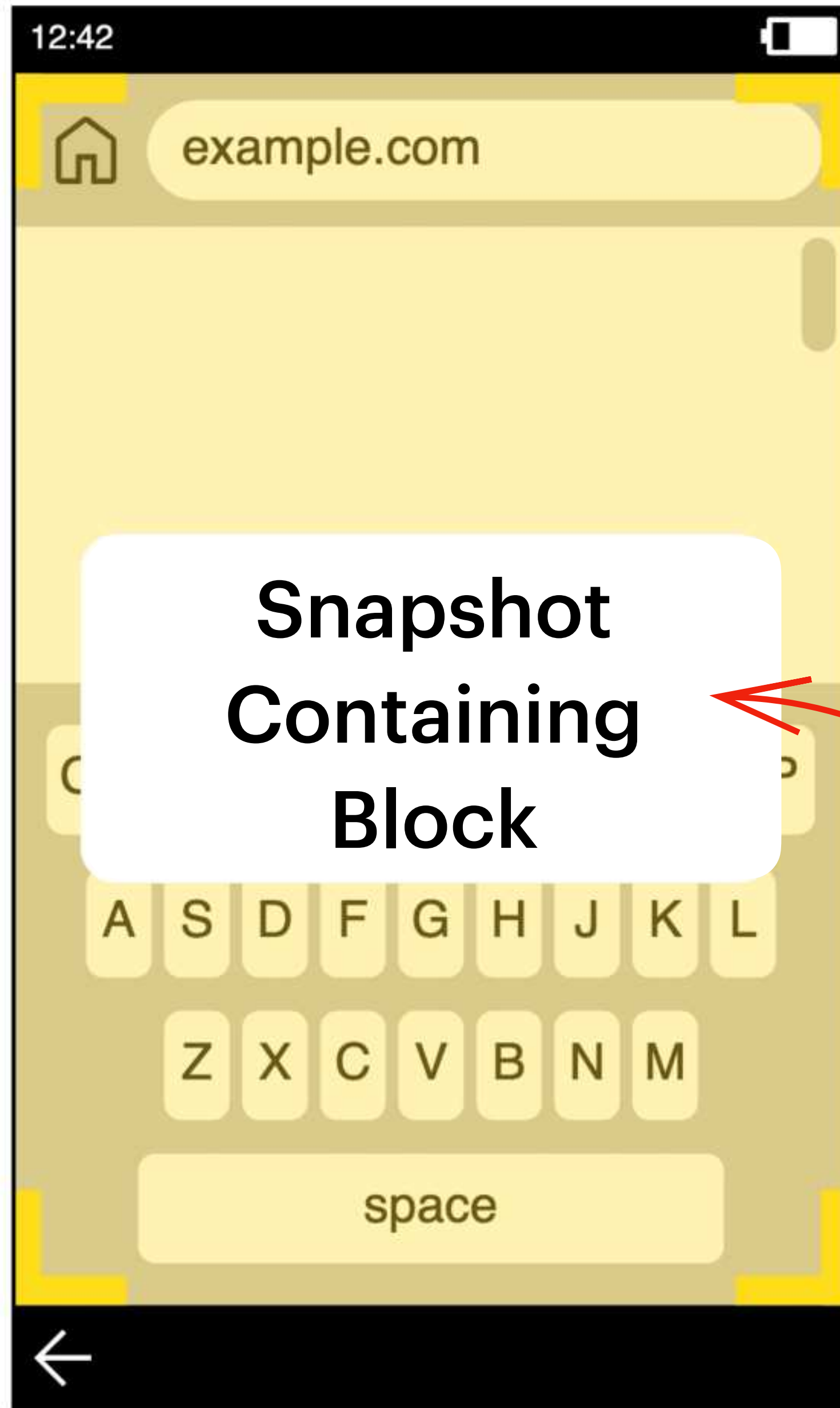
  boxGroupAnimation.currentTime = 0;
}
```

***“Why not just do a  
getBoundingClientRect?”***

*Critical you*



**Used by  
the DOM  
and Layout**



**Used by  
View Transitions**

View Transitions Applied: More

← → ↻ [bram.us/2025/02/07/view-transitions-applied-more-performant-view-transition-group-animations/](https://bram.us/2025/02/07/view-transitions-applied-more-performant-view-transition-group-animations/) ☆ 🗄️ Work ⋮

Blog ▾ Talks About 📄 RSS ▾ Search

# Bram.us

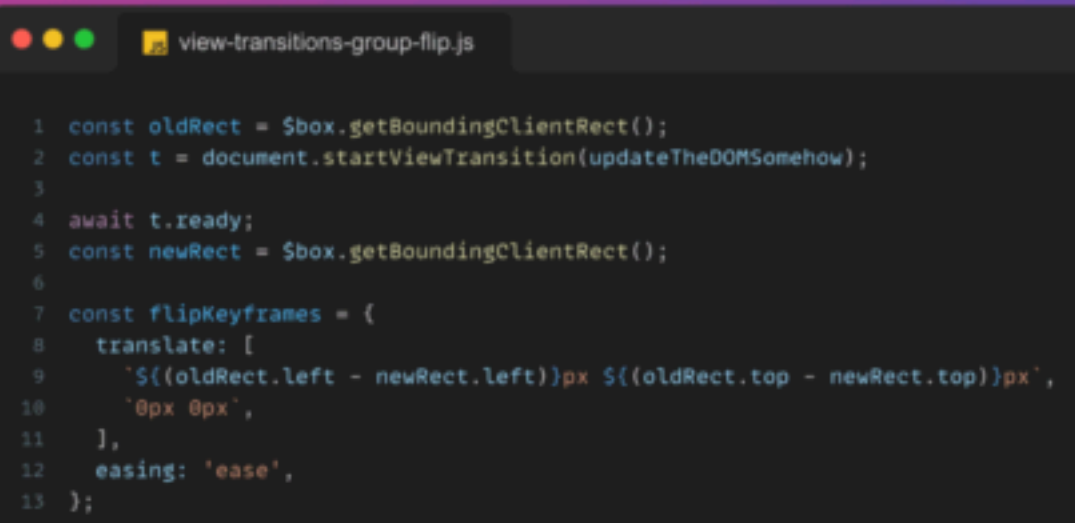
A rather geeky/technical weblog, est. 2001, by Bramus

Careers in tech are changing rapidly. Check out the latest AI job postings on Authentic Jobs!

ADS VIA CARBON

## ★ View Transitions Applied: More performant `::view-transition-group(*)` animations

🕒 February 7, 2025 🗨️ Leave a comment



```
1 const oldRect = $box.getBoundingClientRect();
2 const t = document.startViewTransition(updateTheDOMSomehow);
3
4 await t.ready;
5 const newRect = $box.getBoundingClientRect();
6
7 const flipKeyframes = (
8   translate: [
9     `${(oldRect.left - newRect.left)}px ${oldRect.top - newRect.top}px`,
10    `0px 0px`,
11  ],
12   easing: 'ease',
13 );
```

<https://brm.us/view-transitions-keyframes>

View Transitions Applied: De... x +

← → ↻ [bram.us/2025/03/04/view-transitions-snapshot-containing-block/](https://bram.us/2025/03/04/view-transitions-snapshot-containing-block/) ☆ 🗄️ | Work ⋮

Blog ▾ Talks About 📄 RSS ▾ Search

# Bram.us

A rather geeky/technical weblog, est. 2001, by Bramus

Get Unblocked, Ship Fast. Boost productivity with AI. Fewer blockers, faster shipping, 100 PRs free. ADS VIA CARBON

## ★ View Transitions Applied: Dealing with the Snapshot Containing Block

🕒 March 4, 2025 🔗 <https://brm.us/snapshot-containing-block> 💬 Leave a comment

```
const boxGroupAnimation = vtAnimations.find((anim) => {
  return anim.effect.pseudoElement === '::view-transition-group(box)';
});

// Get old position + size by extracting the data from the keyframes
const boxGroupKeyframes = boxGroupAnimation.effect.getKeyframes();
const oldMatrix = new DOMMatrix(boxGroupKeyframes[0].transform);

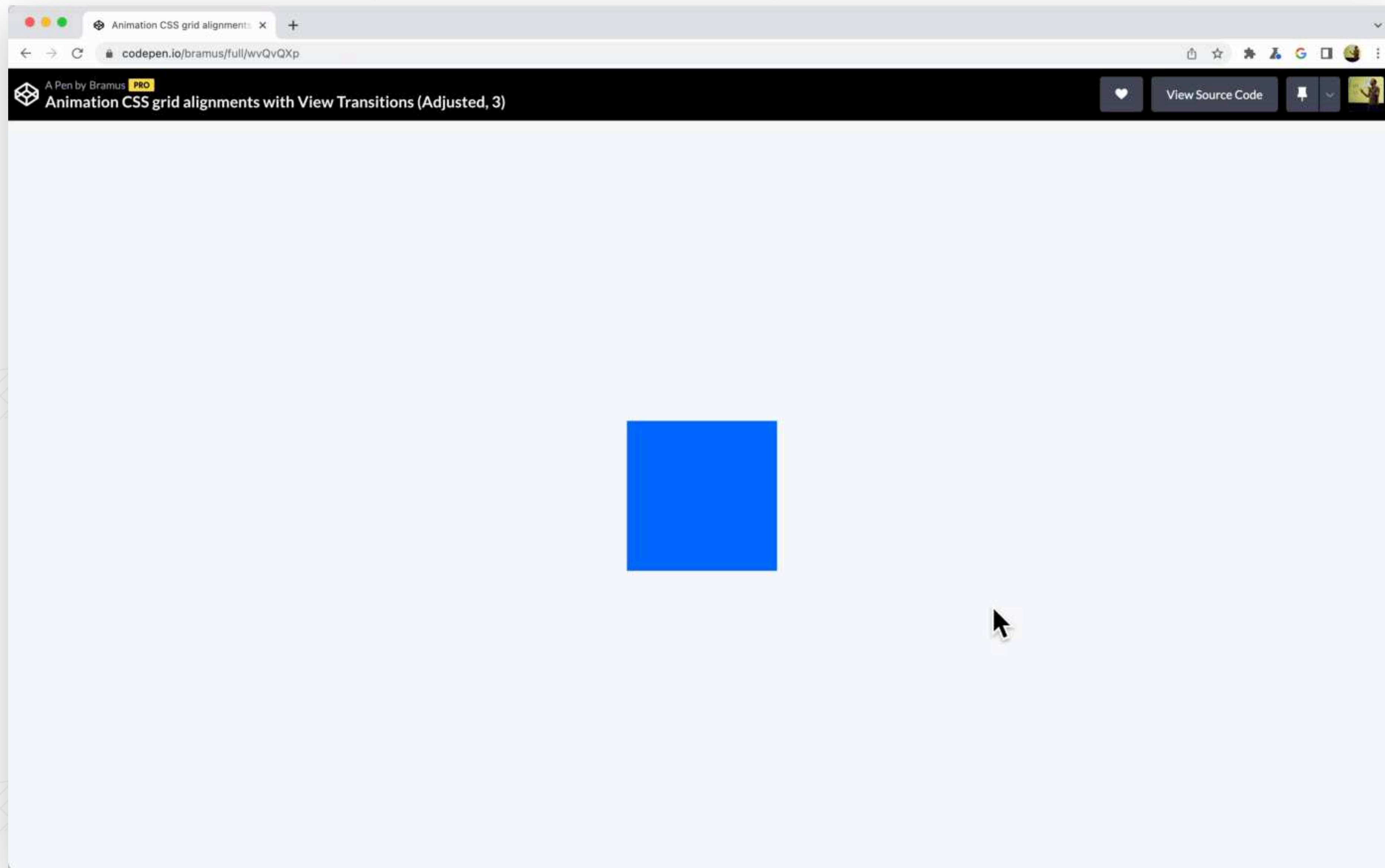
const rectBefore = {
  width: boxGroupKeyframes[0].width.split('px')[0],
  height: boxGroupKeyframes[0].height.split('px')[0],
}
```

<https://brm.us/snapshot-containing-block>

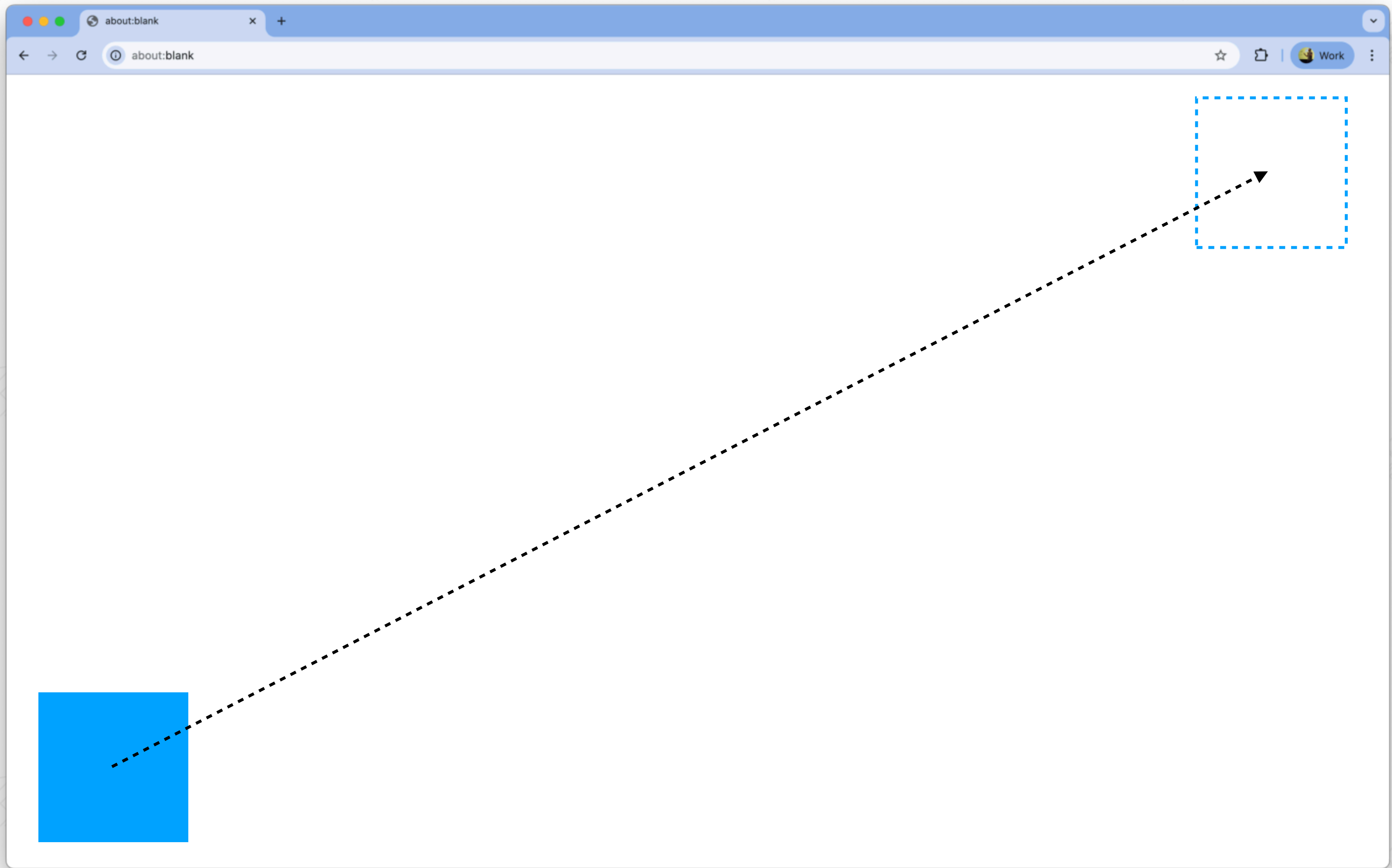


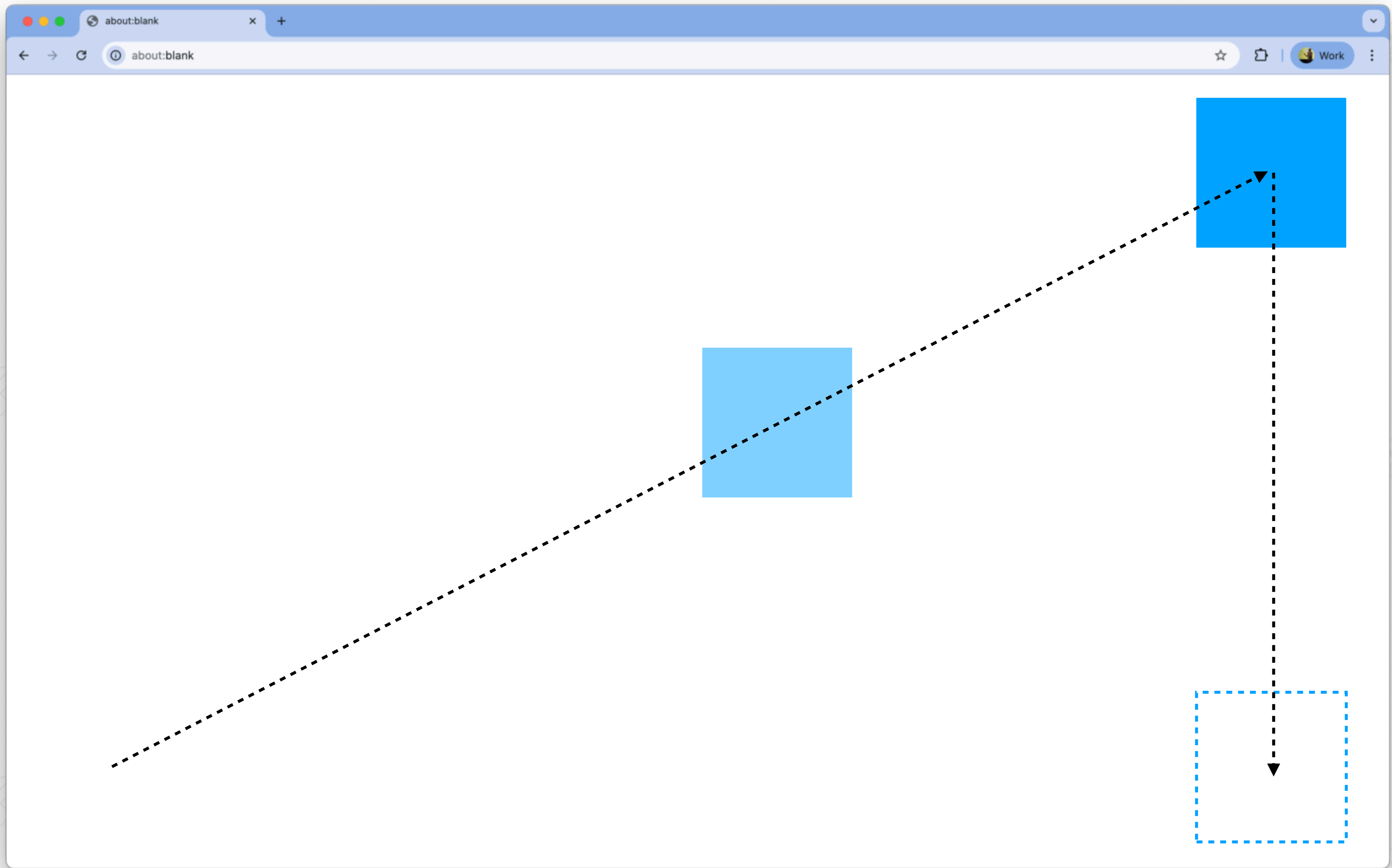
***“Interrupted  
View Transitions  
skip to the end”***

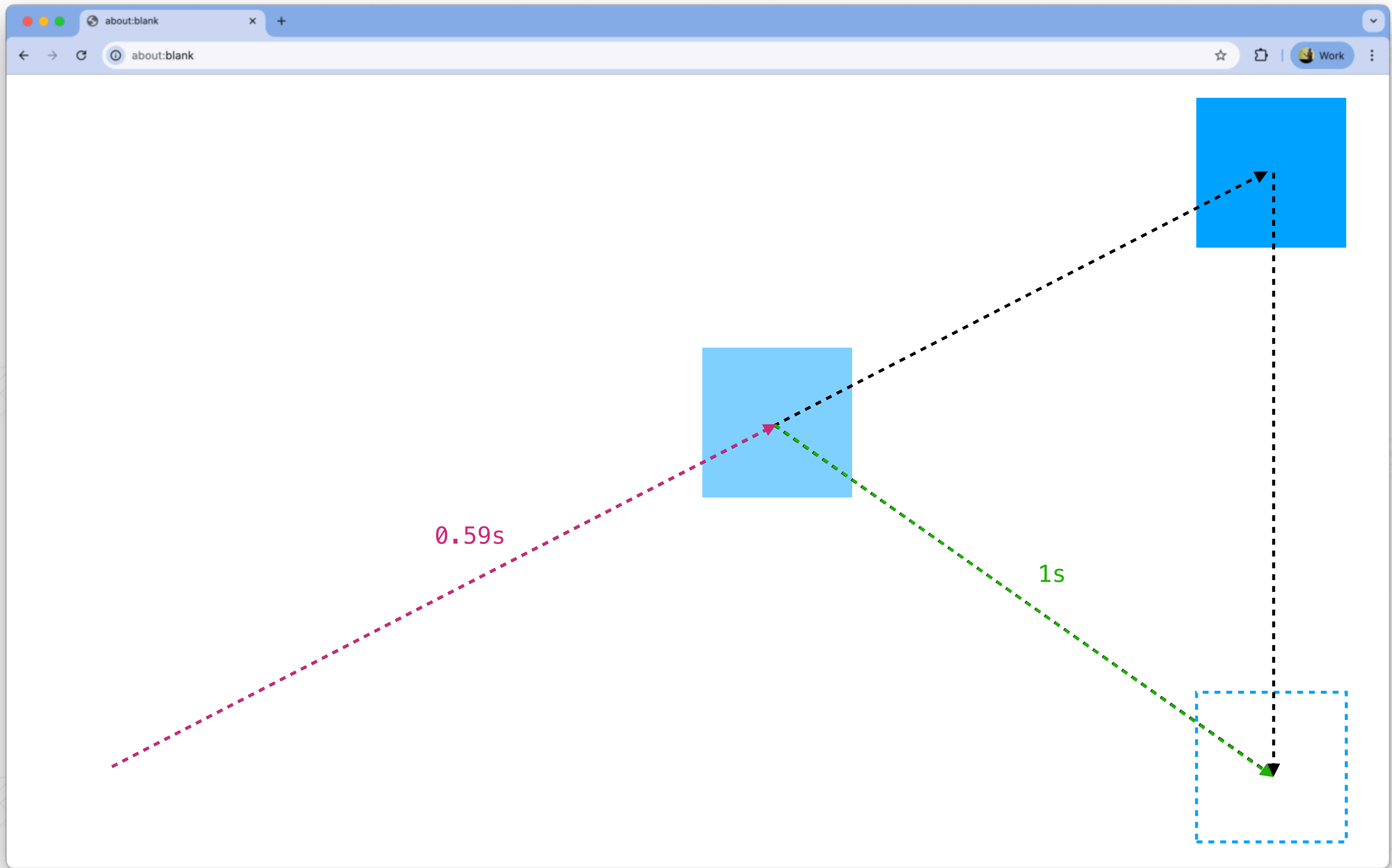
*A rage-clicking you*



<https://codepen.io/bramus/full/wvQvQXp>







WIP

Retargetable View Transitions, Redu X

https://cdpn.io/pen/debug/

Inspector Console Debugger

Search HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <div class="clickedbox" inert="" style="top: 245px; left: 635px; width: 79px;">
    </div>
    <div class="box" style="top: 245px; left: 635px; width: 79px; background-color: rgb(122, 162, 120);">
    </div>
    <div class="fake-vt-group" inert="">
    </div>
    <p>
    </p>
    <script id="rendered-js">
    </script>
  </body>
</html>
```

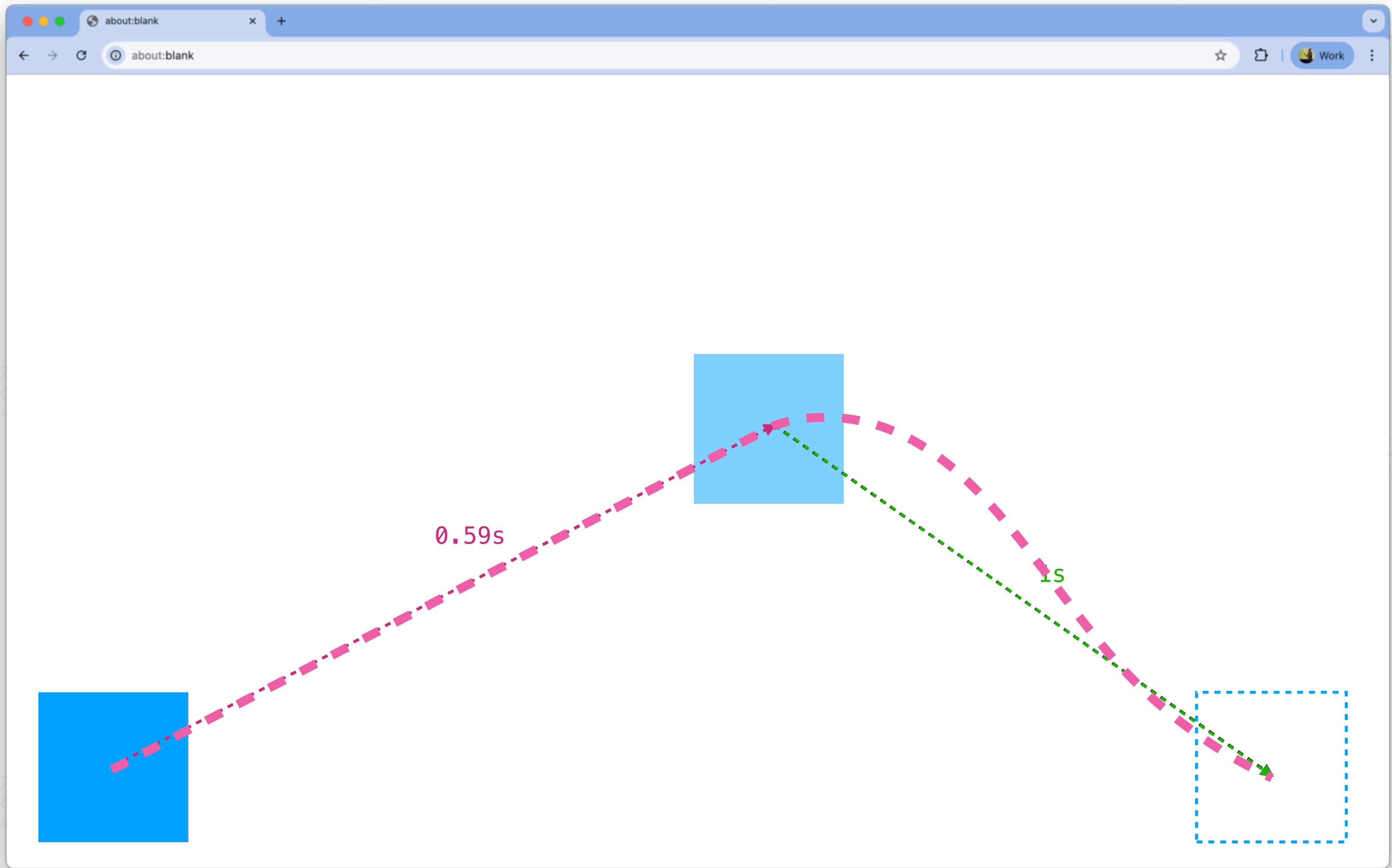
html > body

Rules Layout Computed Changes Compatibility Fonts Anir

Filter Styles :hov .cls +

```
element :: {
}
body :: {
  display: # grid;
  place-content: center;
}
html, body :: {
  margin: 0;
  padding: 0;
  height: 100%;
}
```

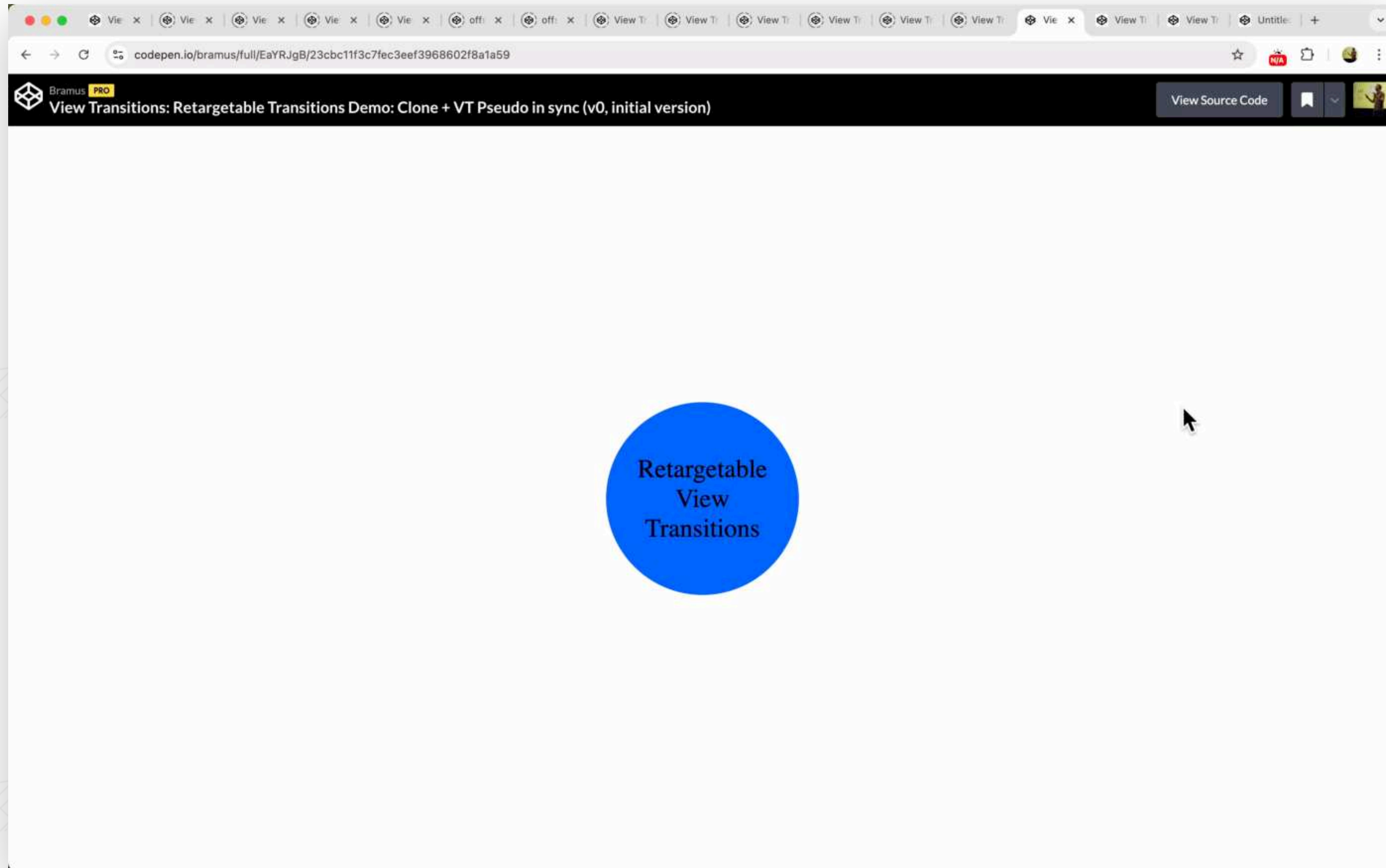
Click anywhere on the page to move the dot to that location



```
const $svg = createLineSVG(...intermediaryPositions, rectAfter);

const $path = $svg.querySelector('#line_path');
const pathData = $path.getAttribute('d');
pathLength = $path.getTotalLength();

const keyframes = {
  easing: "linear",
  visibility: ['visible', 'visible'],
  offsetPath: [`path("${pathData}")`, `path("${pathData}")`],
  offsetRotate: ['0deg', '0deg'],
  offsetAnchor: ['50%', '50%'],
  offsetDistance: ['0%', '100%'],
};
```




<https://codepen.io/bramus/full/YPKjyVM/8879924fa74779c696b4fe415d391751>

Post by @kvndy.bsky.social

bsky.app/profile/kvndy.bsky.social/post/3ly3vxj7uq22q

Post

 **kvndy**  
@kvndy.bsky.social


For interruption and retargeting of in-progress animation, the solution is additive and relative. View Transitions give CSS Animation behavior. 🤖  
FLIP gives CSS Transition behavior. 🤖 Relative Animation blends animations together 🤖 (not updated for mobile yet)  
[codepen.io/kvndy/pen/OJ...](https://codepen.io/kvndy/pen/OJ...)

4:52 PM · Sep 5, 2025

1 like

1 comment

Write your reply

 **Andrew Roberts** @ajr.sh · 7mo  
artisanal

thanks for the explainer and example code link!! TIL

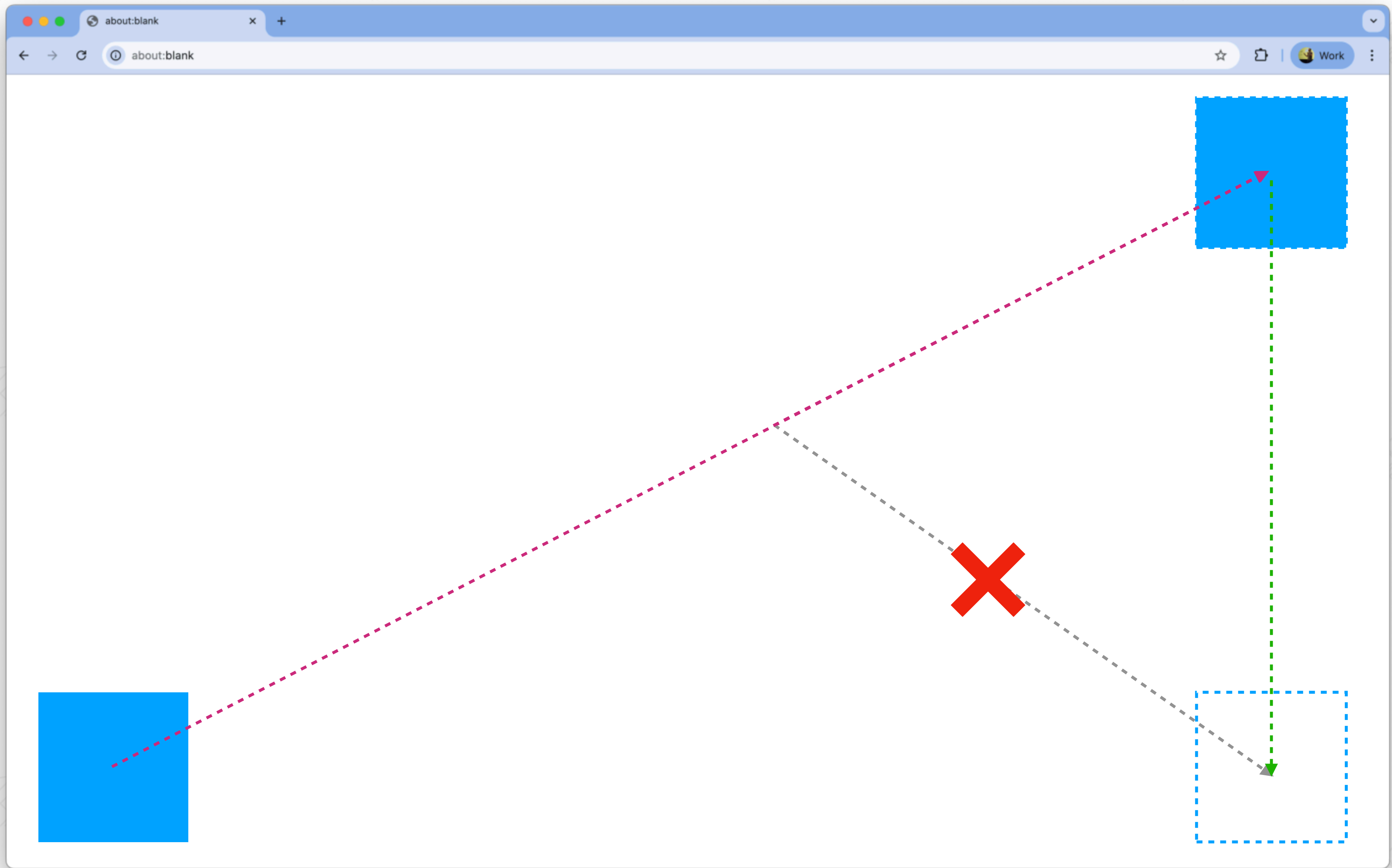
1 like

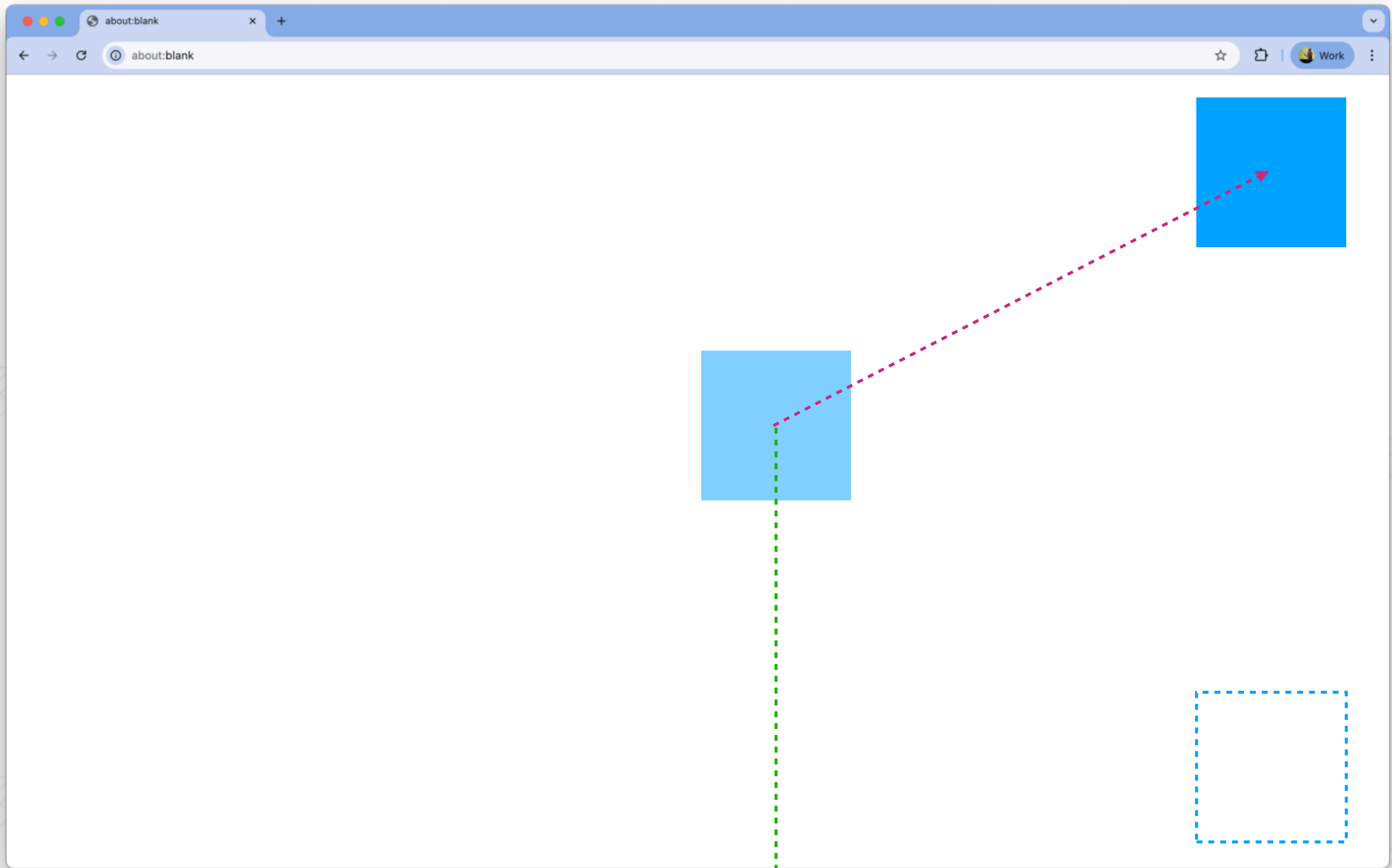
Search

Following  
Discover  
Popular With Friends  
OnlyPosts  
More feeds

Feedback · Privacy · Terms · Help

<https://bsky.app/profile/kvndy.bsky.social/post/3ly3vxj7uq22q>





WIP

Retargetable View Transitions, Redu X

https://cdpn.io/pen/debug/

Inspector Console

Search HTML

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
  </head>  
  <body>  
  </body>  
</html>
```

Click anywhere on the page to move the dot to that location

html > body

Rules Layout Computed Changes Compatibility For

Filter Styles :hov .cls +

```
element :: {  
}  
body :: { inline:28  
  display: # grid;  
  place-content: center;  
}  
html, body :: { inline:19  
  margin: 0;  
  padding: 0;  
  height: 100%;
```

```
const delta = `translate3d(
  ${old_discrete_x - new_discrete_x}px,
  ${old_discrete_y - new_discrete_y}px,
  0px
)`;

const keyframes = [
  { transform: delta },
  { transform: 'translate3d(0px ,0px, 0px)' }
];

const effect = new KeyframeEffect($element, keyframes, {
  duration: duration * 1000,
  easing: easing,
  fill: "backwards",
  composite: "accumulate"
});

const animation = new Animation(effect, document.timeline);
```

<https://codepen.io/bramus/full/NPrNrgY/75a434c85ee0409356ecb321218624d7>

WIP

The image shows a web browser window with the URL `https://cdpn.io/pen/debug/`. The page content consists of a horizontal container with six rounded rectangular buttons labeled 'A', 'B', 'C', 'D', 'E', and 'F'. Below this container is a 'SHUFFLE' button. The browser's developer tools are open on the right side, showing the HTML structure and CSS styles.

**HTML Structure:**

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

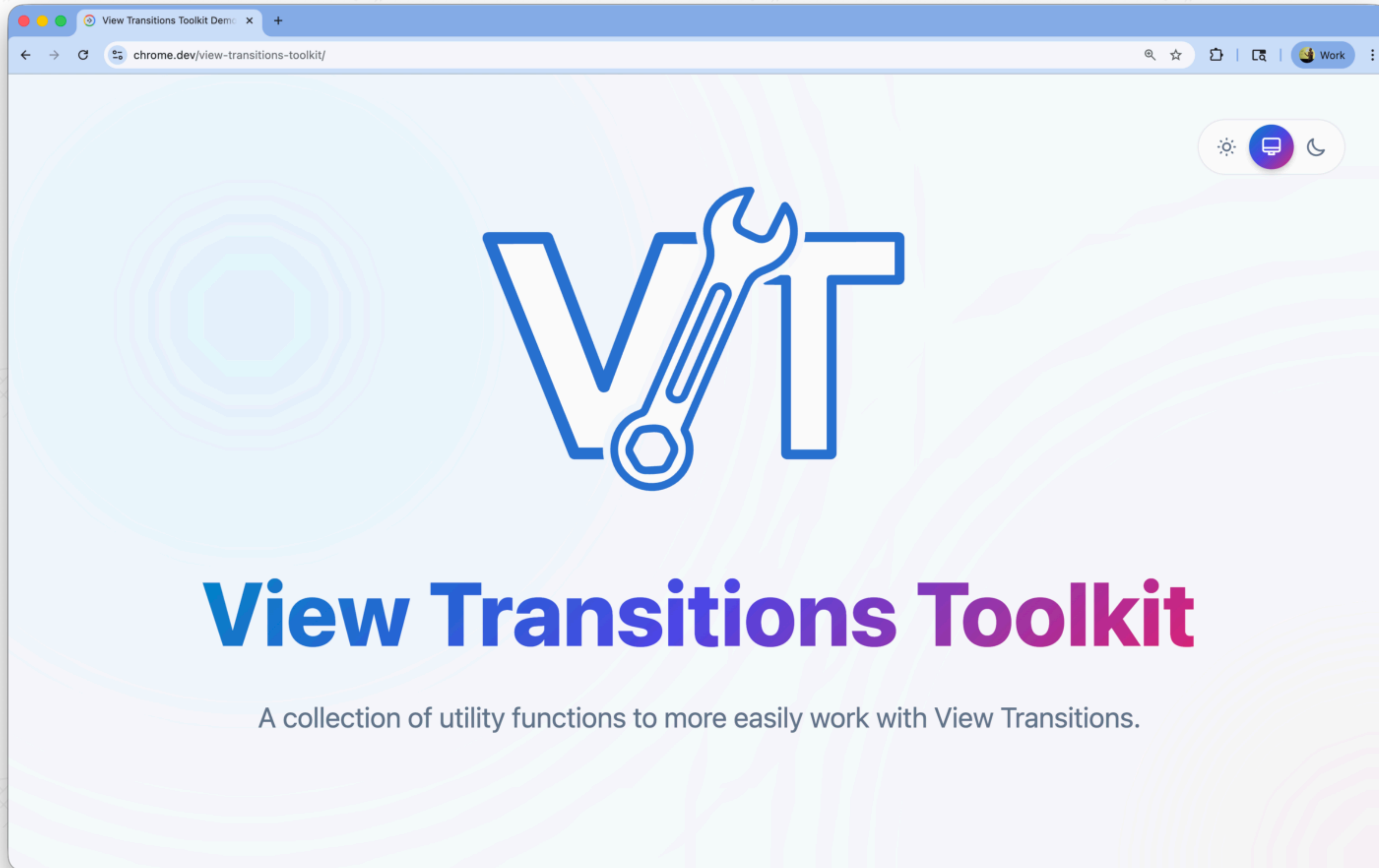
**CSS Styles:**

```
html {
  view-transition-name: none;
}

html {
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
  Helvetica, Arial, sans-serif, "Apple Color Emoji", "Segoe UI
  Emoji", "Segoe UI Symbol";
}

html, body {
  margin: 0;
  padding: 0;
  height: 100%;
}

element attributes style {
  -x-lang:-;
}
```



<https://chrome.dev/view-transitions-toolkit>

The screenshot shows a web browser window with the address bar at `chrome.dev/view-transitions-toolkit/get-animations/`. The main content area is split into two parts. On the left is a demo window with a light blue background and a central blue square containing the text "Click the document!". On the right is the documentation page, which includes a "Back" button, a title "Animations: getAnimations", a descriptive paragraph, a "The Code" section with a code block, a "Functions Used" section with another code block, and a link to "documentation".

**Animations: getAnimations**

The `getAnimations` function from the toolkit is a wrapper around `document.getAnimations()` that returns only the animations related to View Transitions. You can further filter the animations by providing a `view-transition-name` and/or a `ViewTransitionPart`.

**The Code**

```
const t = document.startViewTransition(() => {
  randomize();
});

await t.ready;

const animations = getAnimations(t, "box");
```

**Functions Used**

This demo uses `getAnimations` to read animation details during a transition:

```
import { getAnimations } from "view-transitions-toolkit/animations";
```

Read more in the [documentation](#).

<https://chrome.dev/view-transitions-toolkit>

```
import { getAnimations } from "view-transitions-toolkit/animations";

const activeViewTransition = document.startViewTransition(...);
await activeViewTransition.ready;

activeAnimations = getAnimations(activeViewTransition);
```

Playback Control Demo - View x +

chrome.dev/view-transitions-toolkit/playback-control/

/demo

BOX

Start Pause Resume

Scrub Progress 0.00

← Back

### Playback Control

Control the playback of a View Transition. Use utilities to pause, resume, or scrub all animations linked to a View Transition simultaneously.

#### The Code

The demo binds click and input events to the `pause`, `resume`, and `scrub` functions:

```
document.getElementById("pause").addEventListener("click", () => {
  pause(t);
});

document.getElementById("resume").addEventListener("click", () => {
  resume(t);
});

document.getElementById("scrub").addEventListener("input", (e) => {
  scrub(t, e.target.value);
});
```

Once the View Transition is paused, use the slider to scrub through the animation, or click resume to let it finish.

#### Functions Used

This demo uses `pause`, `resume`, and `scrub` to synchronize View Transition playback with a range slider or custom timeline:

```
import { pause, resume, scrub } from "view-transitions-toolkit/playback-control";
```

<https://chrome.dev/view-transitions-toolkit>

chrome.dev/view-transitions-toolkit/scroll-driven-view-transition/

← Back

## Scroll-Driven View Transition

When scrolling the page, the header card will toggle between the large and small state. This is achieved by starting a View Transition that is immediately paused and then scrubbed based on the scroll position.

### The Code

Between two scroll positions a View Transition is started and, once started, immediately paused and scrubbed based on the scroll position.

```
const startViewTransition = async () => {
  document.startViewTransition(() => {
    document.querySelector(".card").classList.toggle("small");
  });
  await document.activeViewTransition.ready;
  pause(document.activeViewTransition);
};

const updateAnimations = () => {
  scrub(document.activeViewTransition, scrollProgress);
};
```

When scrolling back the animations need to play in reverse. This is achieved by reversing the animations and scrubbing from 1 to 0.

```
if (isReverse) {
  for (const anim of getAnimations(document.activeViewTransition)) {
    anim.reverse();
  }
}
```

<https://chrome.dev/view-transitions-toolkit>

```
import { pause, scrub }
  from "view-transitions-toolkit/playback-control";

const startViewTransition = async () => {
  document.startViewTransition(() => {
    document.querySelector(".card").classList.toggle("small");
  });
  await document.activeViewTransition.ready;
  pause(document.activeViewTransition);
};

const updateAnimations = () => {
  const scrollProgress = /* ... */;
  scrub(document.activeViewTransition, scrollProgress);
};
```

Optimize Demo - View Transitions

chrome.dev/view-transitions-toolkit/optimize/

/demo

REGULAR

OPTIMIZED

← Back

### Animations: Optimize

Optimize `::view-transition-group` animations using performant `transform`-based animations instead of width and height changes.

#### The Code

When you click the page, the demo triggers a View Transition and optimizes the "box-optimized" group. This optimized animation is a `transform`-based animation that animates the `scale` property instead of width and height. The "box-normal" group animation is not optimized.

```
document.body.addEventListener("click", async (e) => {
  const t = document.startViewTransition(() => {
    mutateTheDOM();
  });
  await t.ready;
  optimizeGroupAnimations(t, "box-optimized");
});
```

In addition to optimizing the `box-optimized` group, the code also artificially blocks the main thread while the View Transition is still running.

Because the optimized animation runs on the compositor, it will not be subject to jank, unlike the default width/height animations.

#### Functions Used

This demo uses `optimizeGroupAnimations` to replace width/height animations with performant `transform`-based animations:

<https://chrome.dev/view-transitions-toolkit/>

```
import {  
  optimizeGroupAnimations  
} from "view-transitions-toolkit/animations";  
  
const t = document.startViewTransition(() => { ... });  
await t.ready;  
  
optimizeGroupAnimations(t, "box");
```



← MORE  
THIS WAY

A neon sign is mounted on a dark wall. The sign features a white arrow pointing to the left, followed by the words "MORE" and "THIS WAY" in a white, sans-serif font. The sign is illuminated from within, creating a bright glow. The background is dark, and the sign is framed by a dark border.

✓ Re-enable  
pointer-events

***“The document is not  
interactive while a VT runs”***

✓ Use Element-Scoped  
View Transitions

*You, clicking around*

```
document.startViewTransition(...);
```

*View Transition*

```
$list .startViewTransition(...);
```

*View Transition*



147



X



X

```
$list .startViewTransition(...);
```

*Element-Scoped View Transition*



147

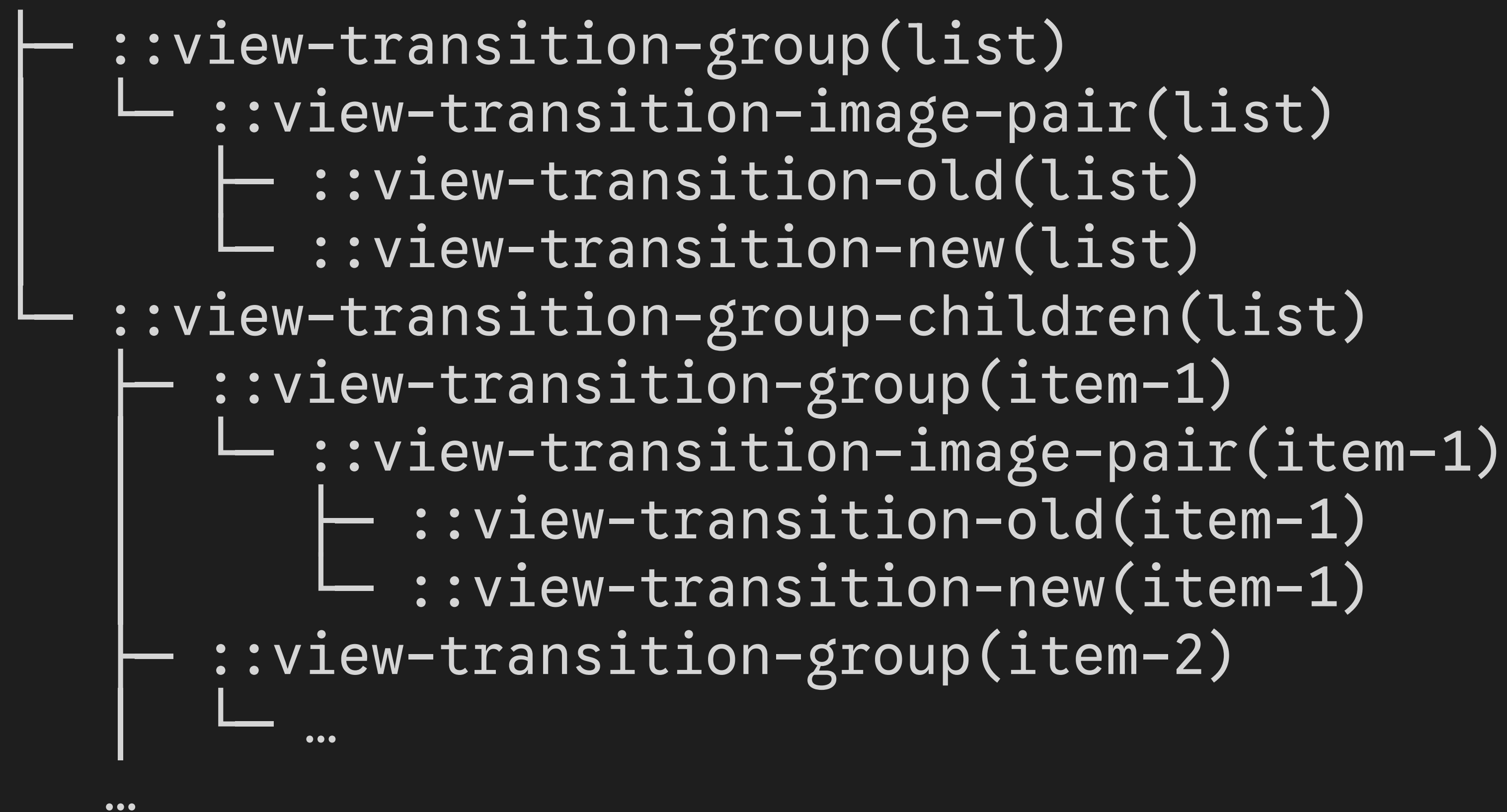


X



X

`#list ::view-transition`



*Element-Scoped View Transition Pseudo Tree*

Concurrent and Nested  
View Transitions! 😁



147



X



X

album 1

album 2

album 3

REORDER



album 6

album 7

album 8

REORDER

Run concurrent and nested view transitions with element-scoped view transitions

developer.chrome.com/docs/css-ui/view-transitions/element-scoped-view-transitions?hl=en

chrome for developers Docs Case studies Blog New in Chrome Search English Sign in

Docs > CSS and UI

Home > Docs > CSS and UI Was this helpful?

# Run concurrent and nested view transitions with element-scoped view transitions

Bramus

Published: Mar 27, 2026

Element-scoped view transitions let multiple view transitions run simultaneously, allow ongoing view transitions to be nested within another, and resolve `z-index` issues you might encounter with document-scoped view transitions—all while keeping the rest of the page interactive. Read this guide to learn how to use them.


Re-imagine the web with Scoped View Transitions  
Chrome for Developers re-imagined

Where to?  
Argentina

<https://brm.us/scoped-transitions>

***“Can you polyfill it?”***

*Your colleague who thinks websites need to look the same in every browser*



***No, but ...***

*The short answer*


W3C Editor's Draft

CSS View Transitions Module

drafts.csswg.org/css-view-transitions-1/

# CSS View Transitions Module Level 1

Editor's Draft, 2 May 2025



▼ More details about this document

**This version:**  
<https://drafts.csswg.org/css-view-transitions-1/>

**Latest published version:**  
<https://www.w3.org/TR/css-view-transitions-1/>

**Implementation Report:**  
<https://wpt.fyi/results/css/css-view-transitions>

**Feedback:**  
[CSSWG Issues Repository](#)

**Editors:**  
[Tab Atkins-Bittner](#) (Google)  
Jake Archibald (Google)  
Khushal Sagar (Google)

**Suggest an Edit for this Spec:**  
[GitHub Editor](#)

Copyright © 2025 World Wide Web Consortium. W3C® liability, trademark and permissive document license rules apply.

## Abstract

This module defines the View Transition API, along with associated properties and pseudo-elements, which allows developers to create animated visual transitions representing changes in the document state.

[CSS](#) is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, etc.

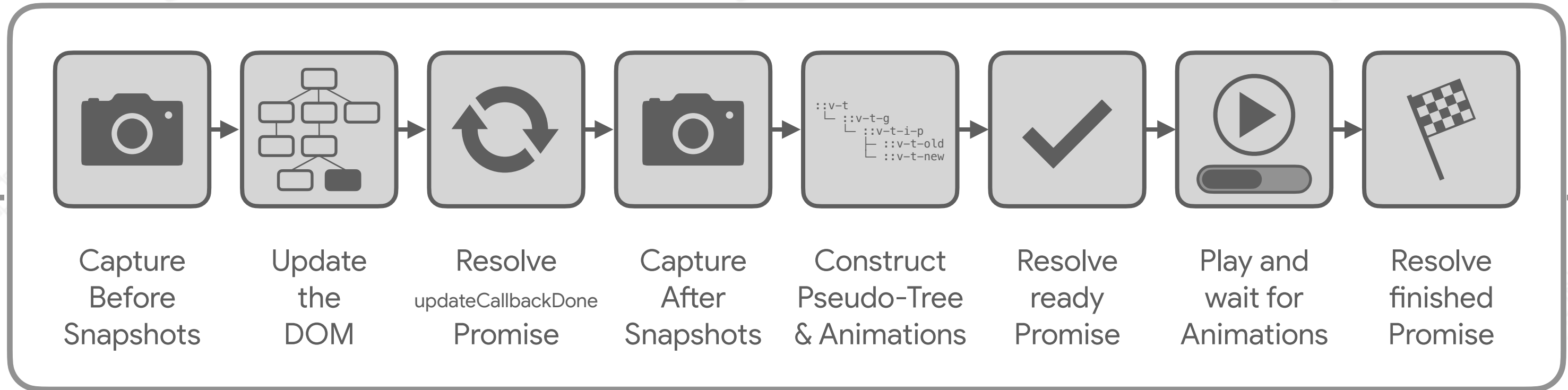
## TABLE OF CONTENTS

- 1 Introduction
  - 1.1 Separating Visual Transitions from DOM Updates
  - 1.2 View Transition Customization
  - 1.3 View Transition Lifecycle
  - 1.4 Transitions as an enhancement
  - 1.5 Rendering Model
  - 1.6 Examples
- 2 CSS properties
  - 2.1 Tagging Individually Transitioning Subtrees: the 'view-transition-name' property
    - 2.1.1 Rendering Consolidation
- 3 Pseudo-elements
  - 3.1 Pseudo-element Trees
  - 3.2 View Transition Pseudo-elements
    - 3.2.1 Named View Transition Pseudo-elements
    - 3.2.2 View Transition Tree Root: the '::view-transition' pseudo-element
    - 3.2.3 View Transition Named Subtree Root: the '::view-transition-group()' pseudo-element
    - 3.2.4 View Transition Image Pair Isolation: the '::view-transition-image-pair()' pseudo-element
    - 3.2.5 View Transition Old State Image: the '::view-transition-old()' pseudo-element
    - 3.2.6 View Transition New State Image: the '::view-transition-new()' pseudo-element
- View Transition Layout
  - 1 The Snapshot Containing Block
  - 2 View Transition Painting Order

<https://drafts.csswg.org/css-view-transitions-1/>



STATE A



STATE B

github.com/bramus/view-transitions-js

bramus / view-transitions-js

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

view-transitions-js Public

Sponsor Pin Unwatch 1 Fork 0 Star 5

main 1 Branch 0 Tags

Go to file Add file Code

bramus Remove non-spec compliant setup phase 3b322d4 · last month 2 Commits

demo	First Commit	last month
src	Remove non-spec compliant setup phase	last month
LICENSE	First Commit	last month
README.md	First Commit	last month

README MIT license

## ViewTransitions.js

A JavaScript implementation of [css-view-transitions-1](#) (May 2, 2025 Snapshot).

⚠ DO NOT USE THIS IN PRODUCTION. THIS IS AN EXPERIMENT, NOT A POLYFILL.

### NOTES/CAVEATS/LIMITATIONS

- This library captures the snapshots to a `<canvas>` using <https://html2canvas.hertzen.com/> which comes with limitations:
  - Limited CSS support – See <https://html2canvas.hertzen.com/features>
  - No way to capture ink overflow
  - External images are not captured (unless you do some special `crossorigin` attribute things)
- Because I also use html2canvas to capture the new snapshot, the new snapshot is not a live one. (if only we had `element()` in all browsers ... that would allow that!)

About

A JavaScript implementation of `css-view-transitions-1`

javascript css view-transitions view-transition-api

Readme MIT license Activity 5 stars 1 watching 0 forks

Sponsor this project

bramus Bramus

Sponsor

Learn more about GitHub Sponsors

Languages

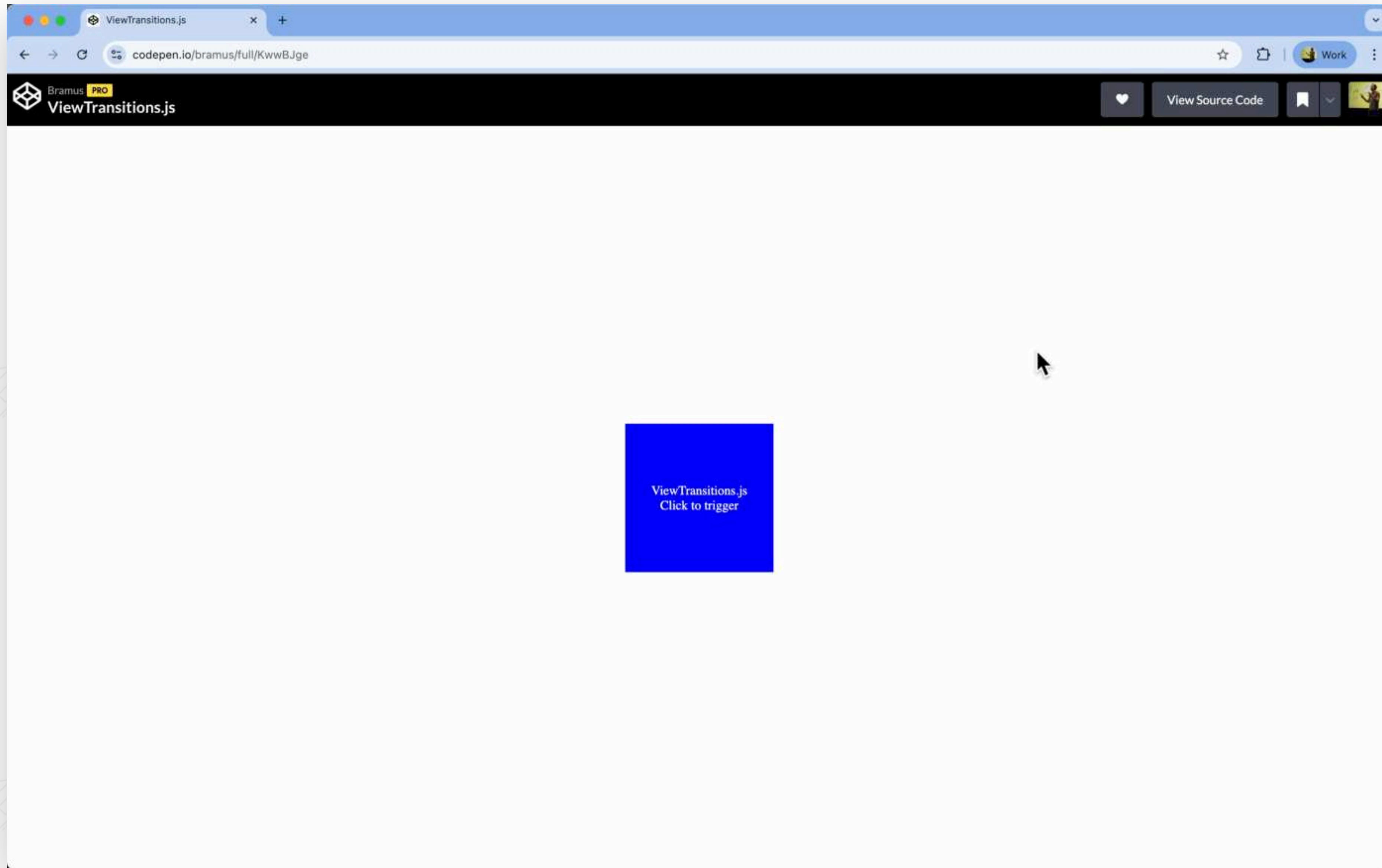
JavaScript 96.6% CSS 3.4%

Suggested workflows

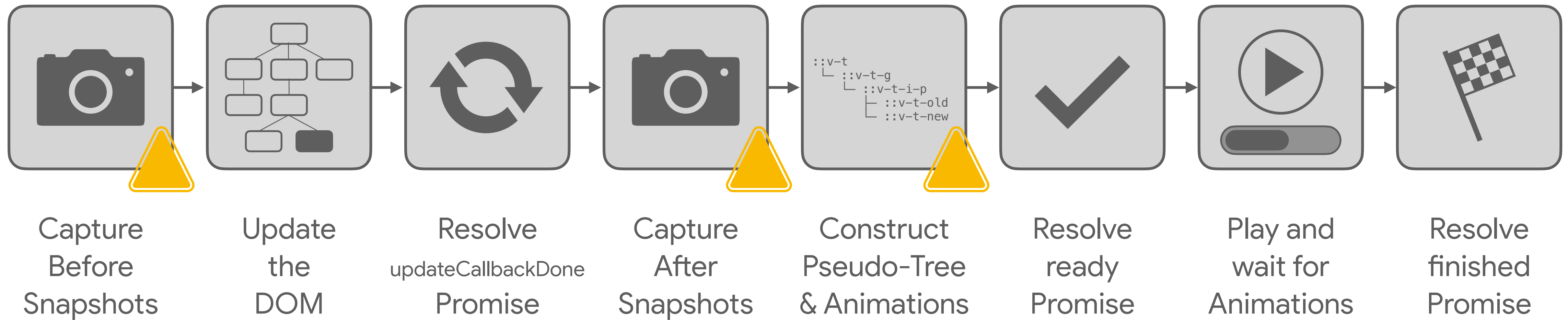
Based on your tech stack

Publish Node.js Package to GitHub Configure

<https://github.com/bramus/view-transitions-js>



<https://codepen.io/bramus/pen/KwwBJge>



<https://github.com/bramus/view-transitions-js>



***DO NOT USE THIS IN  
PRODUCTION. THIS IS AN  
EXPERIMENT, NOT A POLYFILL.***

*Me, trying to warn you*

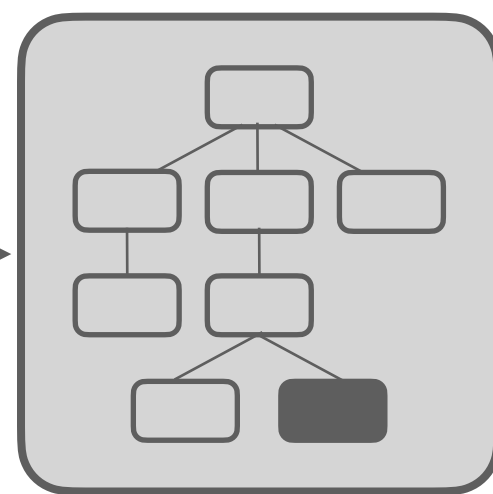


***But what if I only polyfill  
the API surface and don't do  
the animation bits?***

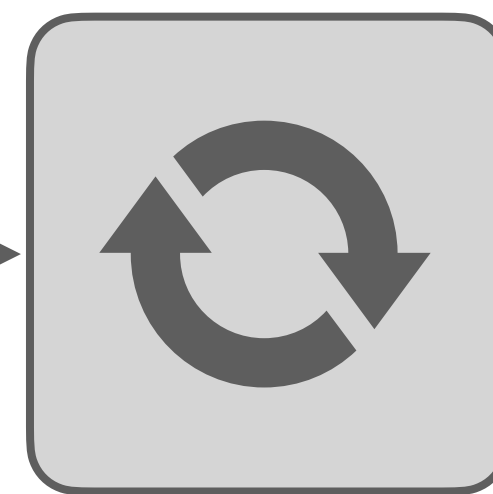
*Me, a few weeks later*



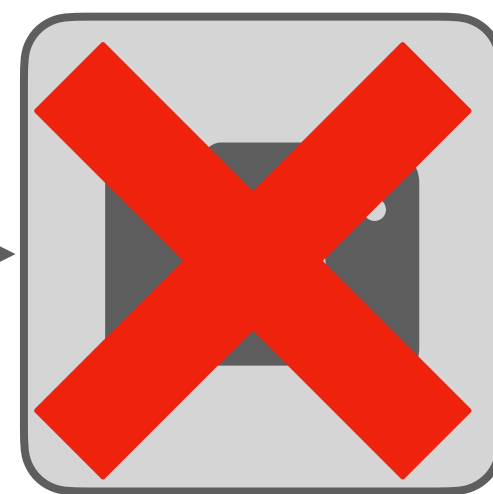
Capture  
Before  
Snapshots



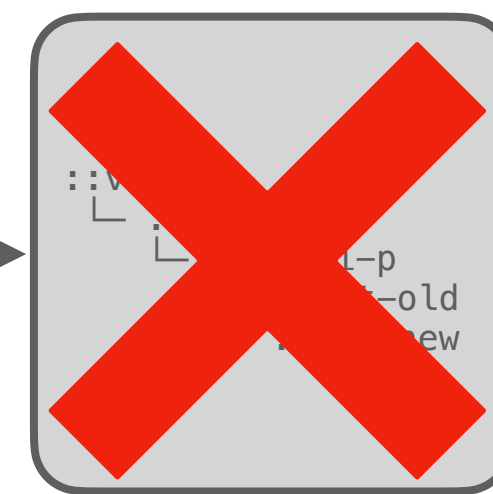
Update  
the  
DOM



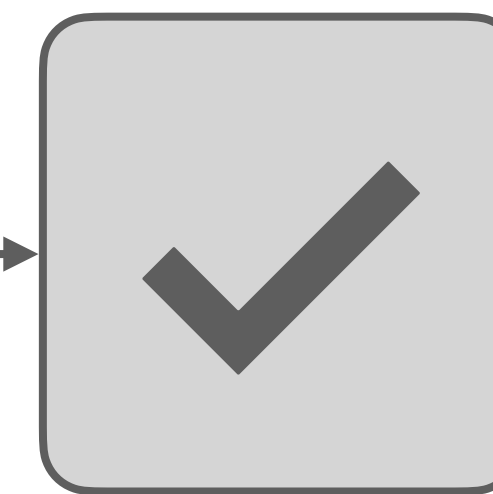
Resolve  
`updateCallbackDone`  
Promise



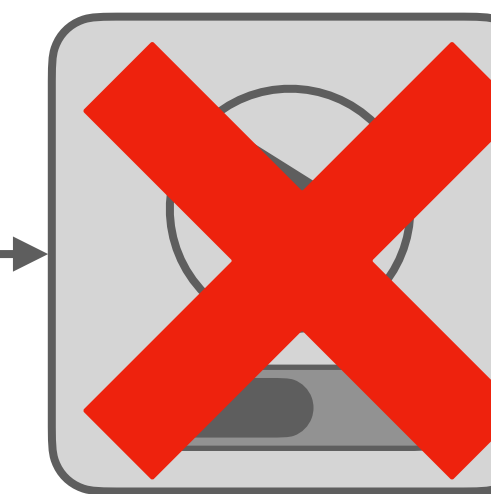
Capture  
After  
Snapshots



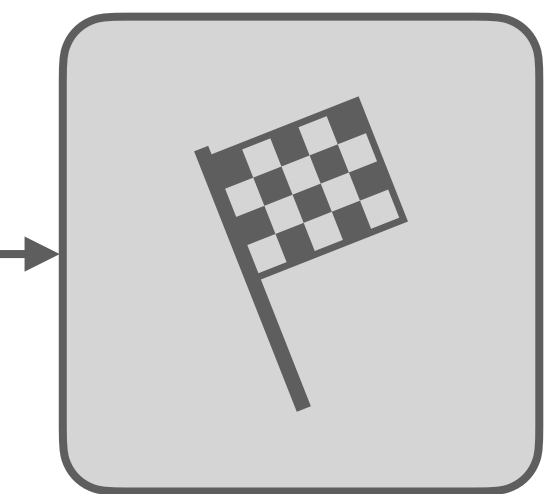
Construct  
Pseudo-Tree  
& Animations



Resolve  
ready  
Promise



Play and  
wait for  
Animations



Resolve  
finished  
Promise

view-transitions-mock - npm x +

npmjs.com/package/view-transitions-mock

## view-transitions-mock TS

1.1.0 • Public • Published a month ago

[Readme](#) [Code](#) Beta [1 Dependency](#) [0 Dependents](#) [3 Versions](#) [Settings](#)

### View Transitions Mock

Mock support for Same-Document View Transitions in browsers with no support.

### Overview

View Transitions Mock is a robust JavaScript implementation of Same-Document View Transitions (including View Transition Types). It is not a polyfill, as it doesn't replicate the pseudo-tree or the animations from View Transitions. Instead, it mocks support for `document.startViewTransition`, `document.activeViewTransition`, `ViewTransition.transitionRoot` and `ViewTransitionTypeSet`. This allows you to write modern, standard-compliant Same-Document View Transitions code for *any* browser, including those without support for `document.startViewTransition` or View Transition Types.

Once **registered**, stop sprinkling `if (document.startViewTransition)` guards across your codebase and, instead, safely call the API, handle its promises, and manage View Transition Types as if they were natively supported.

- Without `view-transitions-mock`:

```
document.querySelector('button').addEventListener('click', async () => {
  if (document.startViewTransition && ("types" in ViewTransition?.prototype)) {
    document.querySelector('#thing').style.viewTransitionName = 'the-third';
    const t = document.startViewTransition({
      update: updateTheDOM,
      types: ['slide', 'from-left']
    });
    await t.finished;
    document.querySelector('#thing').style.viewTransitionName = '';
  } else {
    updateTheDOM();
  }
});
```

### Install

```
> npm i view-transitions-mock
```

### Repository

[github.com/GoogleChromeLabs/view-transitions-mock](https://github.com/GoogleChromeLabs/view-transitions-mock)

### Homepage

[github.com/GoogleChromeLabs/view-transitions-mock#readme](https://github.com/GoogleChromeLabs/view-transitions-mock#readme)

### Weekly Downloads

9


Version	License
1.1.0	Apache-2.0

Unpacked Size	Total Files
72.4 kB	6

### Last publish

a month ago

### Collaborators



<https://www.npmjs.com/package/view-transitions-mock>

```
$link.addEventListener('click', e => {  
  if (!document.startViewTransition) {  
    showPlaylist(e.currentTarget);  
    return;  
  }  
  
  document.startViewTransition(() => {  
    showPlaylist(e.currentTarget);  
  });  
});
```

*Same-Document View Transition for Single-Page Applications*

```
import { register } from "view-transitions-mock";
register();

$link.addEventListener('click', e => {
  document.startViewTransition(() => {
    showPlaylist(e.currentTarget);
  });
});
```



***“Internet Explorer  
did it first”***

*The Gen X (and older) folks in the room*

Introduction to Filters and Tra x +

learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/platform-apis/ms532847(v=vs.85)?redirectedfrom=MSDN#interpage-transitions

## Interpage Transitions

Interpage transitions enable you to provide effects for the entire window as a Web page is loaded or exited. When a transition is applied to a page, it creates an interpage transition where the page's entire content becomes the object of the filter. You can apply an interpage transition to a page when it is loaded or exited using the same transitions described in the Transitions section above. Just as programs such as Microsoft PowerPoint enable transitions between slides, you can provide wipes and fades, and create custom transitions when the content on a Web page changes.

Transitions are implemented with [meta](#) tags placed in the [head](#) section of Web pages. The **meta** tag specifies the type of transition, as well as whether the transition should occur as the following page is loaded or as the current page is exited.

### Interpage Transition Syntax

The syntax for interpage transitions consists of two parts: when the transition should play, and what kind of interpage transition to use. The following two examples show how to apply interpage transitions upon loading and exiting a page.

```
<META http-equiv="Page-Enter"
CONTENT="progid:DXImageTransform.Microsoft.Blinds(Duration=4)" />

<META http-equiv="Page-Exit"
CONTENT="progid:DXImageTransform.Microsoft.Slide(Duration=2.500,slidestyle='HIDE
```

The first **meta** tag causes the **Blinds** transition to play when the user loads the page, lasting four seconds; the second **meta** tag causes a **Slide** transition to play when the user exits the page, lasting 2.5 seconds, written as 2 seconds and 500 milliseconds.

**In this article**

- Creating Multimedia Effects with Visual Filters and Transitions
- Defining Visual Filters
- Scripting Filters
- Filter Design Considerations
- Visual Filter Scenarios
- Transitions
- Interpage Transitions**
- Transition Design Considerations
- Transition Scenarios

[Show 3 more](#)

<https://brm.us/interpage-transitions>

***“Just like Microsoft PowerPoint enables transitions between slides, you can provide wipes and fades [...] when the content on a Web page changes.”***

<https://brm.us/interpage-transitions>

Influential Computers - The RevealTrans HTML Meta Tag - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Address Links

# INFLUENTIAL COMPUTERS LTD.

IT Consultancy Internet Services Website Design Helpdesk Contact About Shopping

## The RevealTrans HTML Meta Tag

The table below lists the various HTML RevealTrans META Tag statements that you may like to use in your site. The tag can be set to function at Page-Enter, Page-Exit, Site-Enter and Site-Exit. The demonstration links below are set to action on Page-Exit, so click the link to move to the demonstration page, then click the Back button to see the effect and return to this page.

These effects require Internet Explorer 4 or later.

Effect	Meta Tag Format
<a href="#">Box In</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=0)">
<a href="#">Box Out</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=1)">
<a href="#">Circle In</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=2)">
<a href="#">Circle Out</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=3)">
<a href="#">Wipe Up</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=4)">
<a href="#">Wipe Down</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=5)">
<a href="#">Wipe Right</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=6)">
<a href="#">Wipe Left</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=7)">
<a href="#">Vertical Blinds</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=8)">
<a href="#">Horizontal Blinds</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=9)">
<a href="#">Checkerboard Across</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=10)">
<a href="#">Checkerboard Down</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=11)">
<a href="#">Random Dissolve</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=12)">
<a href="#">Split Vertical In</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=13)">
<a href="#">Split Vertical Out</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=14)">
<a href="#">Split Horizontal In</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=15)">
<a href="#">Split Horizontal Out</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=16)">
<a href="#">Strips Left Down</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=17)">
<a href="#">Strips Left Up</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=18)">
<a href="#">Strips Right Down</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=19)">
<a href="#">Strips Right Up</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=20)">
<a href="#">Random Bars Horizontal</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=21)">
<a href="#">Random Bars Vertical</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=22)">
<a href="#">Random</a>	<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=23)">
<a href="#">Blend</a>	<meta http-equiv="page-exit" content="blendtrans(duration=3)">

Internet

<https://www.influentialcomputers.com/meta-reveal-demo.asp> (IE6)

```
<meta http-equiv="Page-Enter" content="revealTrans(Duration=0.5,Transition=23)">  
<meta http-equiv="Page-Exit" content="revealTrans(Duration=0.5,Transition=23)">
```

# Interpage Transitions Effects

0 = box-in	5 = wipe-down	10 = checkerboard across	15 = split-horizontal-in	20 = strips-right-up
1 = box-out	6 = wipe-right	11 = checkerboard down	16 = split-horizontal-out	21 = random-bars-horizontal
2 = circle-in	7 = wipe-left	12 = random dissolve	17 = strips-left-down	22 = random-bars-vertical
3 = circle-out	8 = vertical-blinds	13 = split-vertical-in	18 = strips-left-up	23 = random effect
4 = wipe-up	9 = horizontal-blinds	14 = split-vertical-out	19 = strips-right-down	

<https://brm.us/reveal-trans-filter>



***“Let’s bring  
the oldskool back”***

*Me, procrastinating.*

github.com/bramus/ie-page-transitions

bramus / ie-page-transitions

Code Issues 11 Pull requests Discussions Actions Projects Security Insights Settings

ie-page-transitions Public

Sponsor Pin Unwatch 1 Fork 1 Star 20

main 1 Branch 4 Tags

Go to file Add file Code

bramus Update CDN options 88f8bd7 · 4 months ago 44 Commits

demo	Add option to set the backdrop color	4 months ago
src	Add option to set the backdrop color	4 months ago
.gitignore	Add Netlify	last year
LICENSE	Add LICENSE	7 months ago
README.md	Update CDN options	4 months ago
netlify.toml	Redirect netlify deploy to main domain	6 months ago
package-lock.json	0.0.1	5 months ago
package.json	0.0.1	5 months ago

README Apache-2.0 license

## ie-page-transitions

Bringing back Internet Explorer's Page Transitions thanks to the View Transition API

Source GitHub npm v0.0.1 license Apache-2.0 demo Website

### Internet Explorer's Page Transitions

Microsoft Internet Explorer 4.0 and above featured a proprietary feature called [Interpage Transitions](#). It allowed you to specify [a transition effect](#) to apply to the entire window as a Web page was loaded or exited.

About

Bringing back Internet Explorer's Page Transitions thanks to the View Transition API

[page-transitions.style](#)

javascript css view-transitions view-transition-api

Readme Apache-2.0 license Activity 20 stars 1 watching 1 fork

Releases

4 tags

Create a new release

Sponsor this project

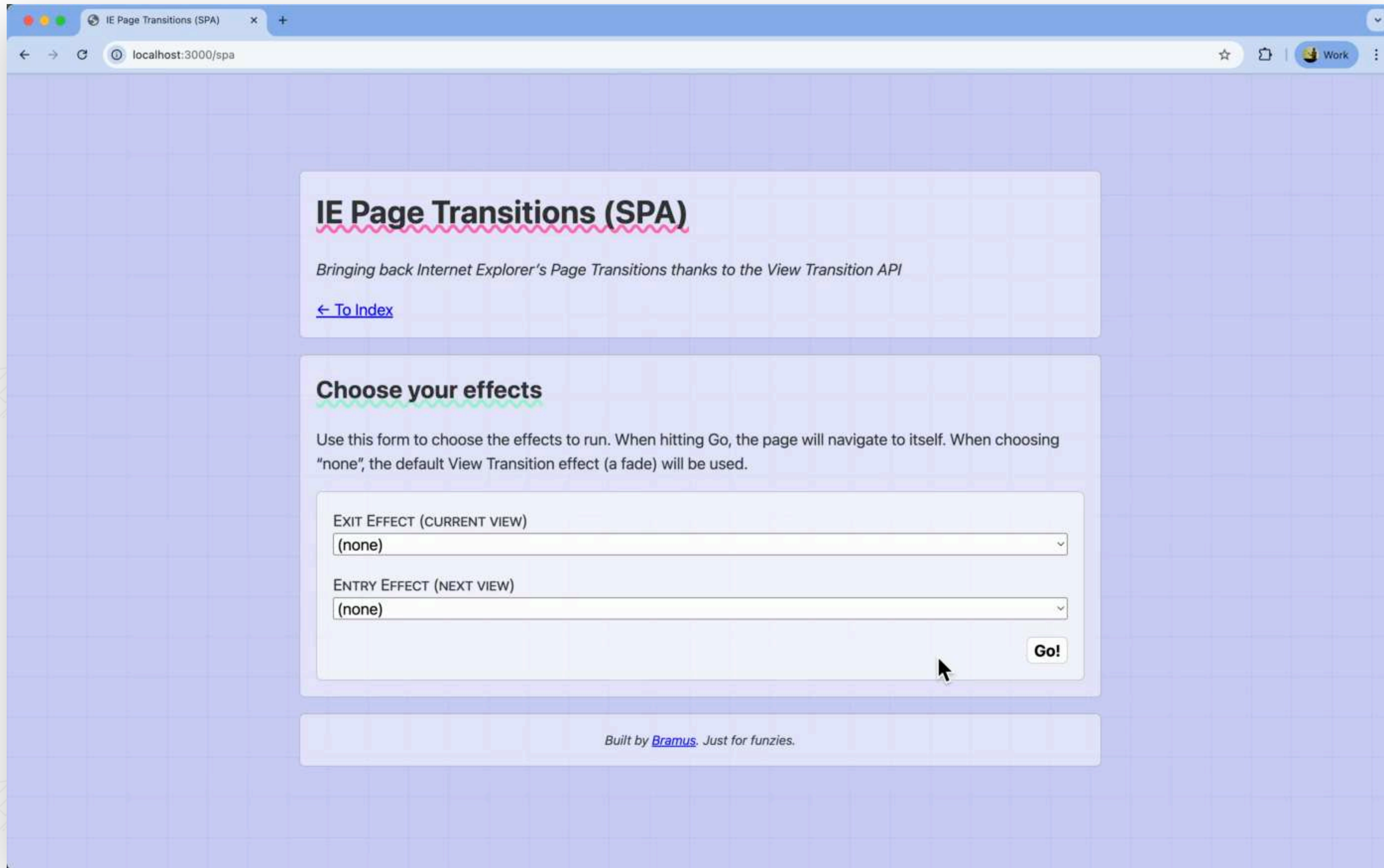
bramus Bramus

Sponsor

Learn more about GitHub Sponsors

Packages

<https://github.com/bramus/ie-page-transitions>



<https://page-transitions.style/>



126



WIP



18.2



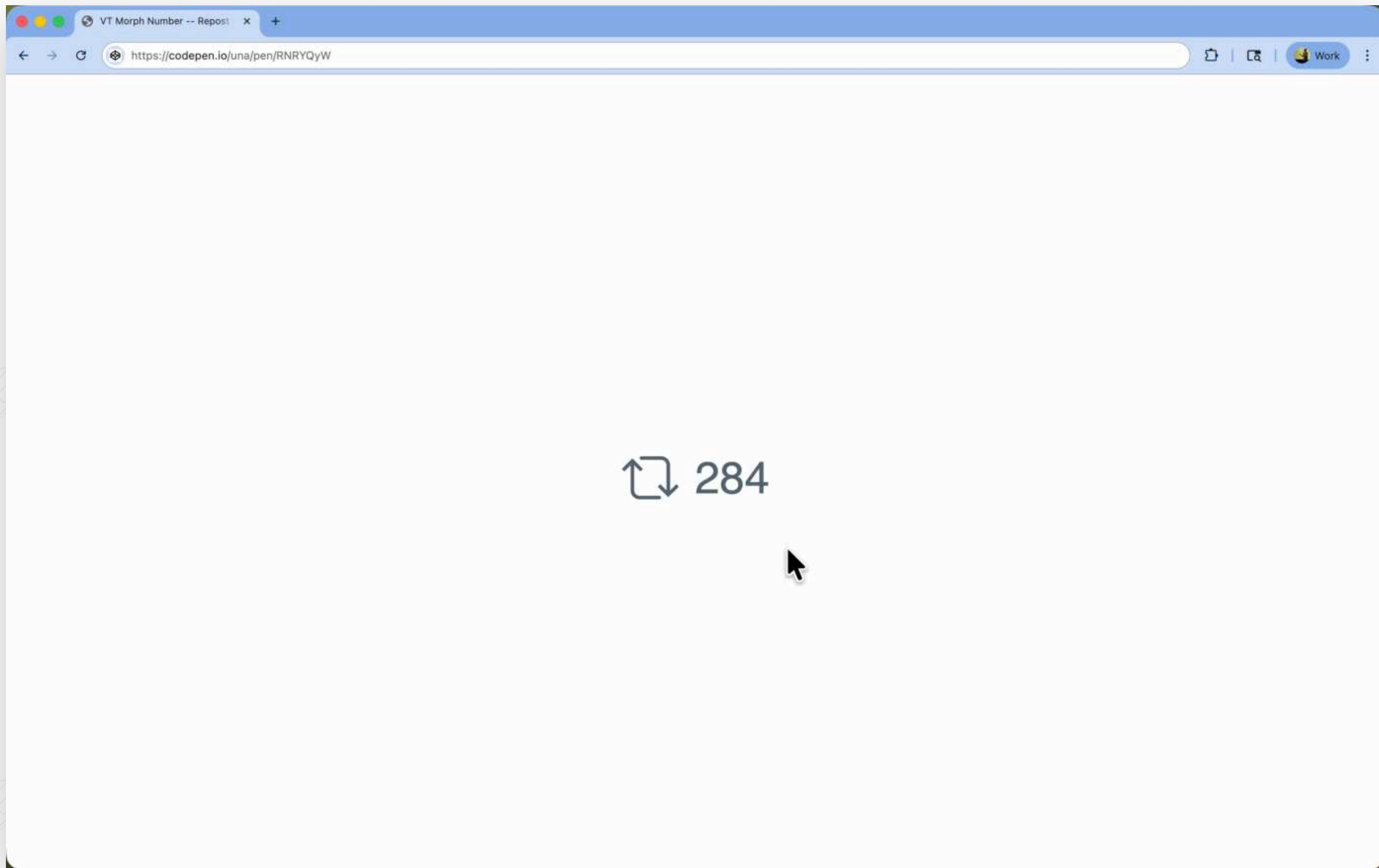
5.5 - 8

```
<meta http-equiv="Page-Enter" content="revealTrans(Duration=0.5,Transition=23)">
<meta http-equiv="Page-Exit" content="revealTrans(Duration=0.5,Transition=23)">

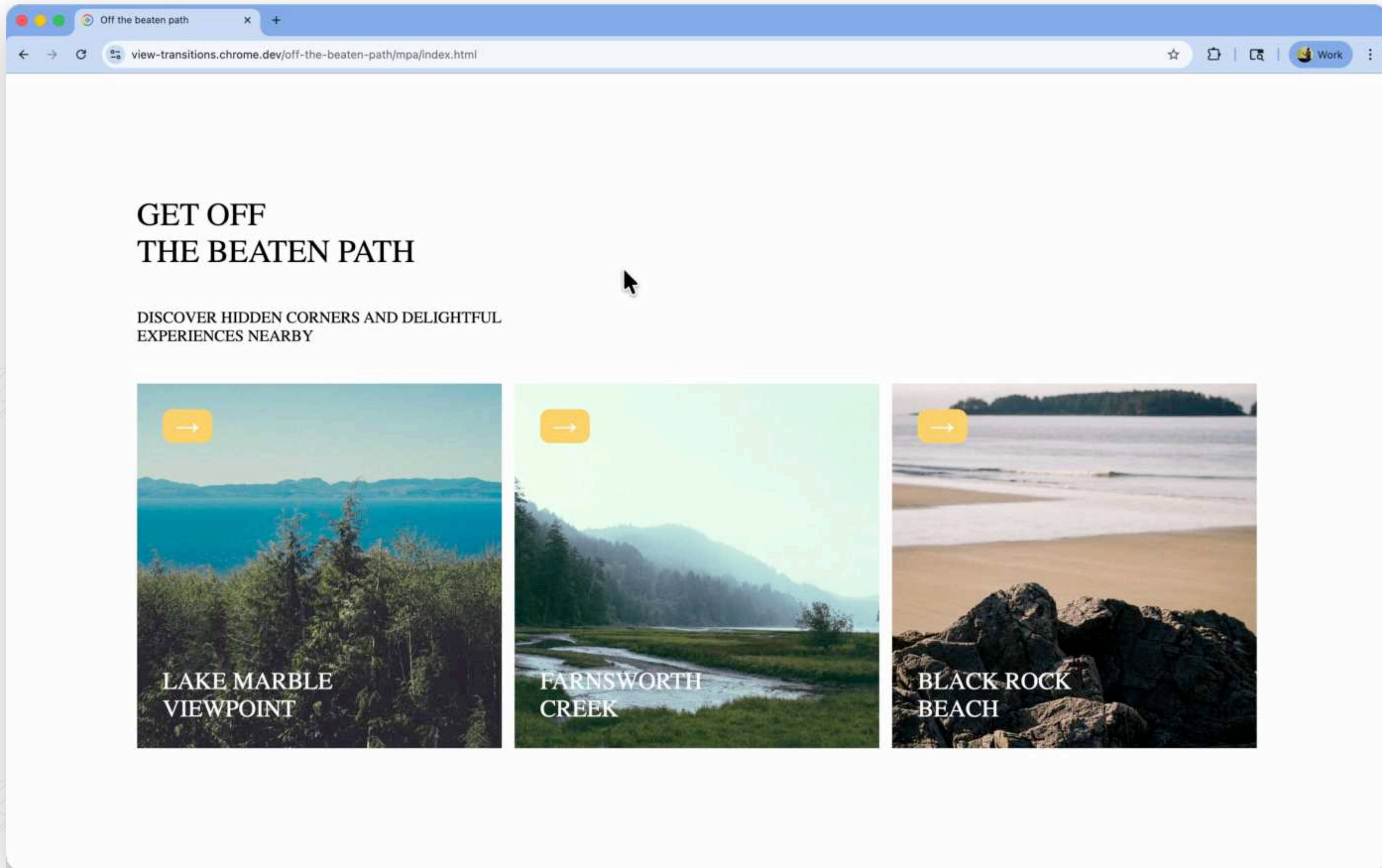
<style>
  @view-transition {
    navigation: auto;
  }
</style>

<link rel="stylesheet" href="ie-page-transitions.css">
<script src="ie-page-transitions.mpa.js" type="module" blocking="render"></script>
```

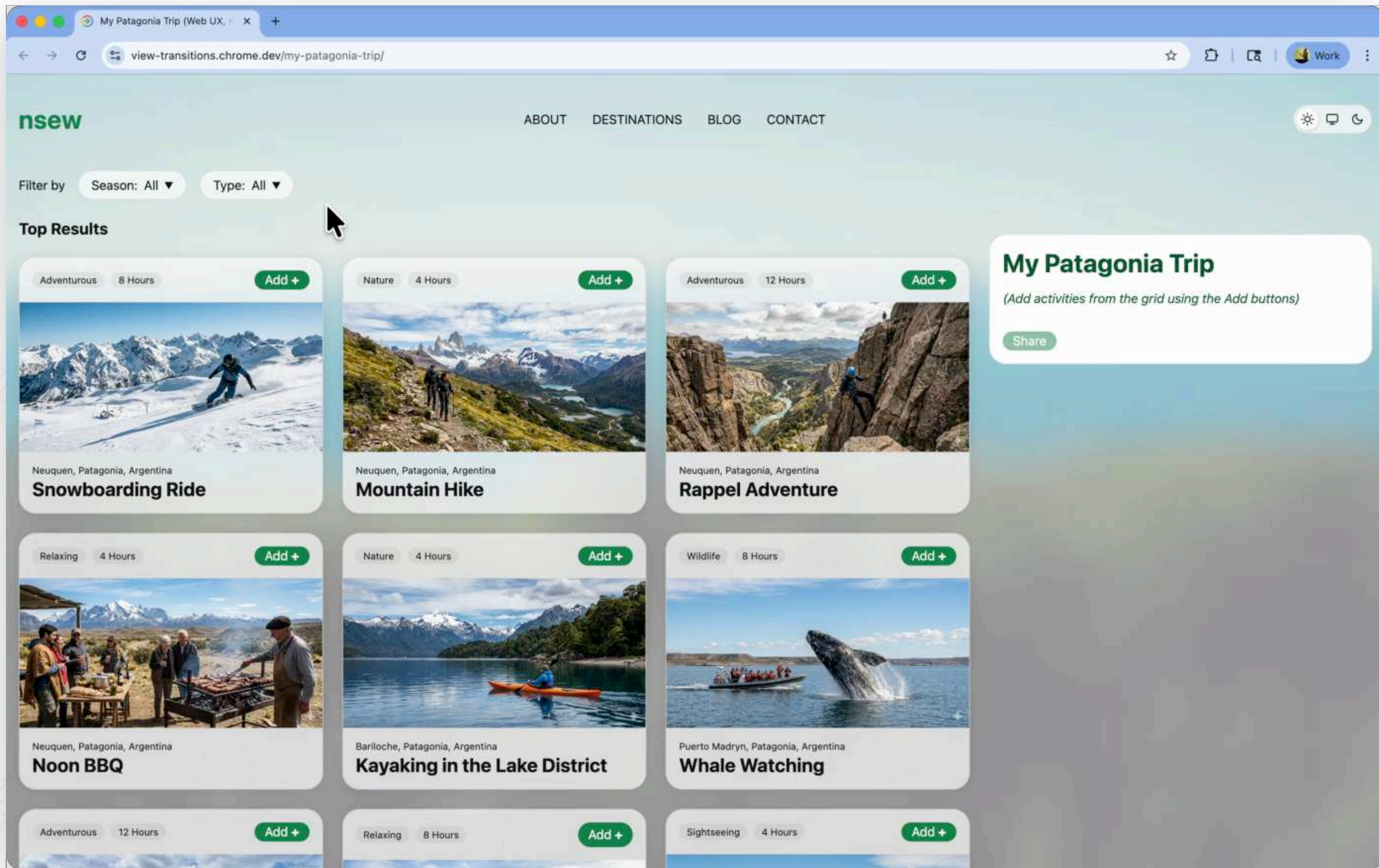




<https://codepen.io/una/pen/RNRYQyW>



<https://view-transitions.chrome.dev/off-the-beaten-path/mpa/>



<https://view-transitions.chrome.dev/my-patagonia-trip/>

# Thanks!



 @bram.us

 @bramus@front-end.social

~~@bramus~~



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