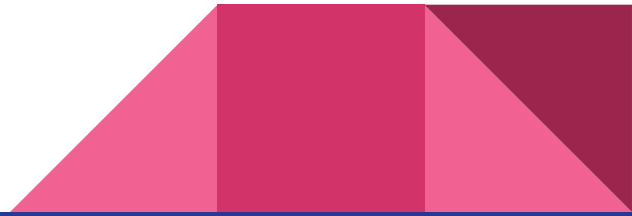
Desktop View Transitions

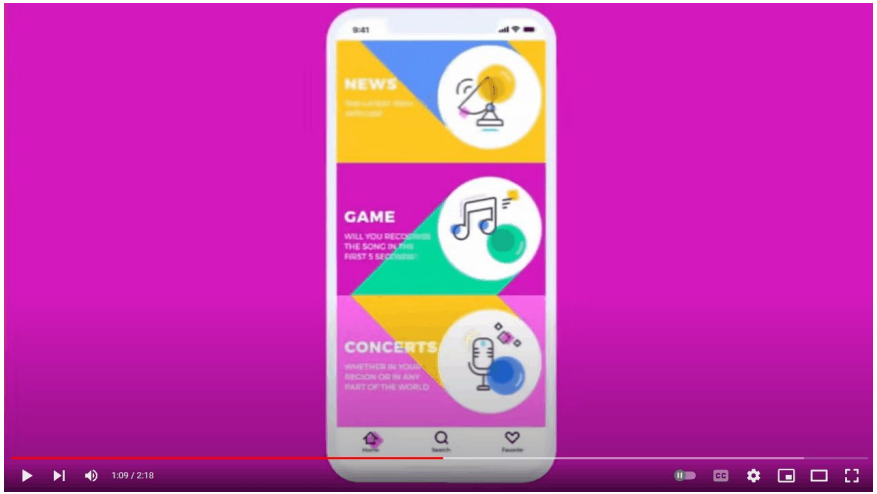
by Edwin Choi

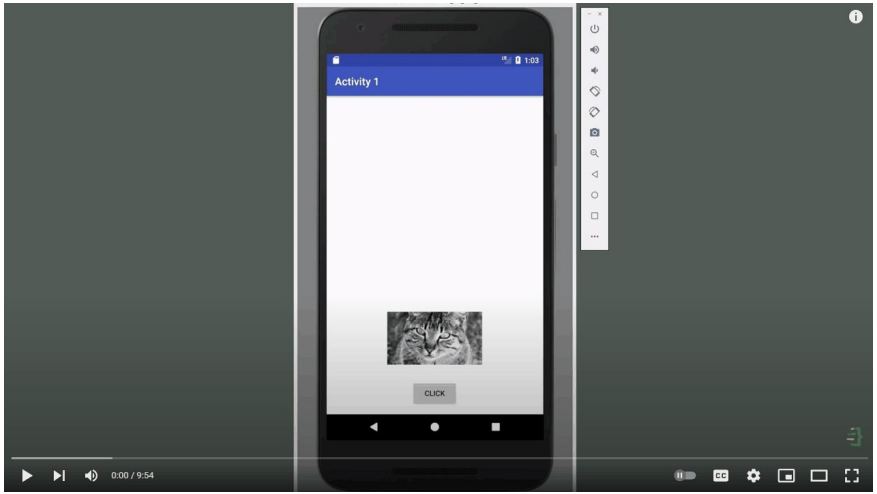
Main Points

- Mobile View Transitions
- Page Transitions on Desktop Views
- Examples
- How it works
- Drawbacks
- Closing Thoughts



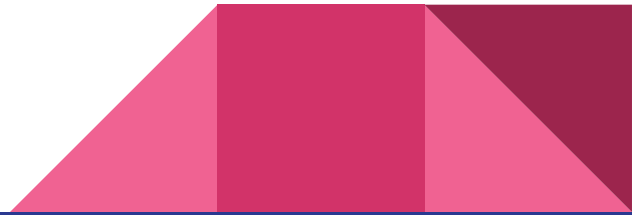
Mobile View Transitions





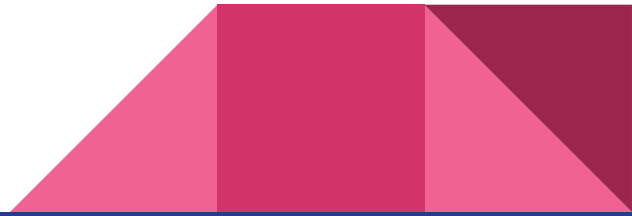
Benefits of VT on Mobile Views

- Synchronizes very well with touch-based navigation.
- Helps to retain user focus on pages or elements related to each other.
- Can also work in tandem with data fetching.
- Outside of that, they provide lots of aesthetic value.



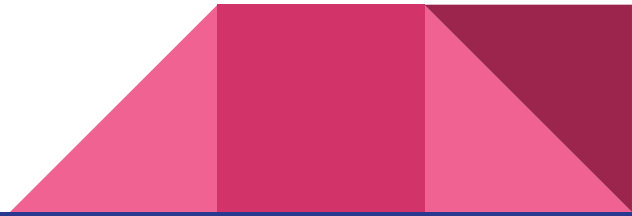
Page Transitions on Desktop Apps

- Possible, but normally very difficult due to the nature of state transitions and how they are handled (ie. React.js)
- However, by using the View Transitions API, one is able to circumvent these issues through the use of state “snapshotting”

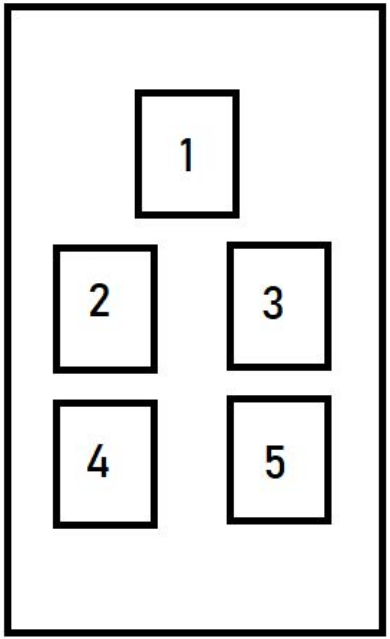


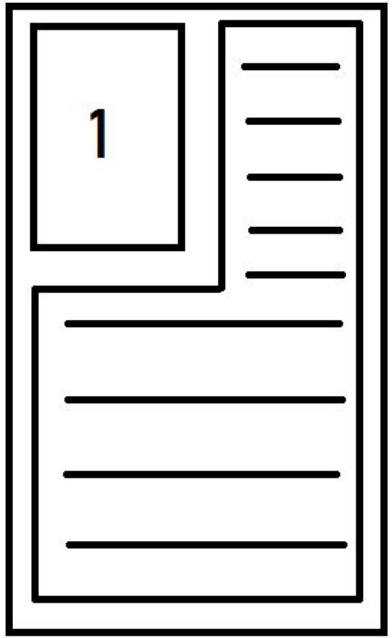
Examples

- Cross-Fade: https://simple-set-demos.glitch.me/3-shared-axis/
- Slow Cross-Fade: https://simple-set-demos.glitch.me/2-slow-cross-fade/
- HTTP203: https://http203-playlist.netlify.app/

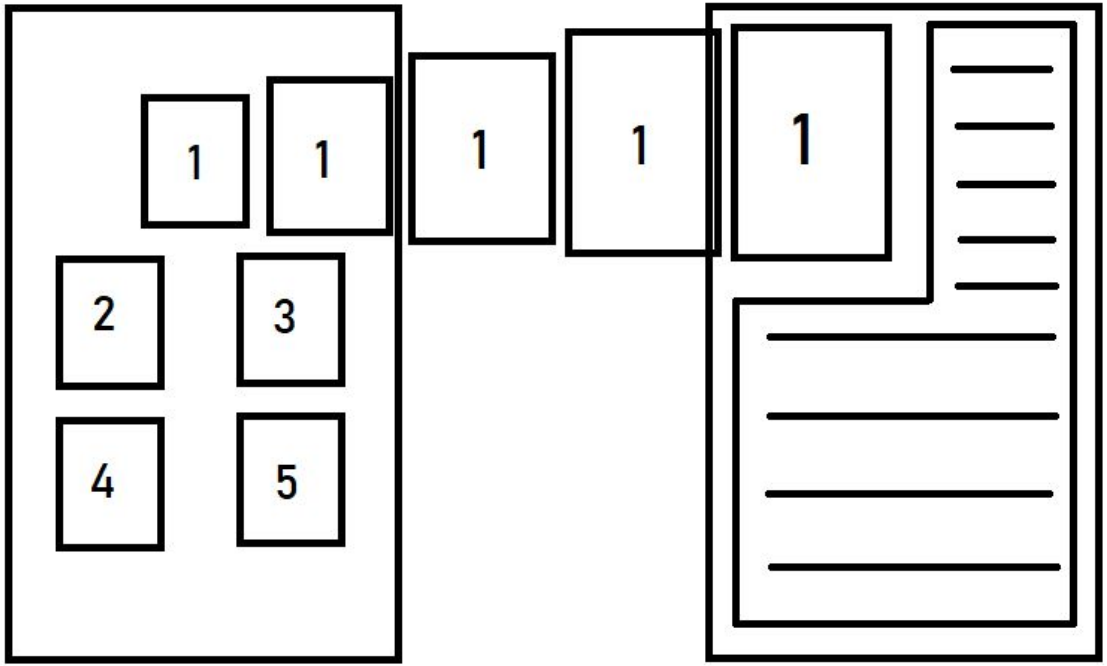


Standard Page Navigation





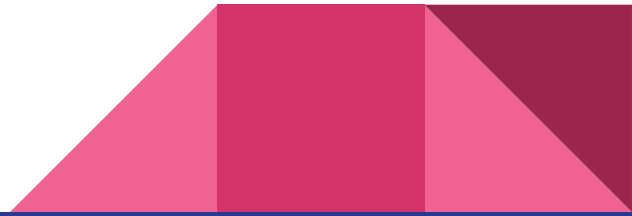
Page/Element Transitions



Drawbacks

- The view transition API is currently only compatible with Google Chrome 111+ and Microsoft Bing.
- However, you can integrate failsafes to account for different runtime environments, so that page loads happen normally if the API is not recognized. (ie. Progressive Enhancement)

- https://caniuse.com/



Can I use view transition ? Settings

12 results found

CSS property: view-transition-name

Usage % of all users ?

Global 53.86%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE ⚠ *
4-110					
111	12-110	3.1-16.3	2-110	10-94	6-10
112	111	16.4	111	95	11
113-115		16.5	112-113		
		TP			

Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
	3.2-16.3	4-19.0		12-12.1		2.1-4.4.4				2.5
111	16.4	20	all	73	13.4	111	110	13.1	13.18	3.1
	16.5									

Notes | Test on a real browser | Feedback

This feature is **experimental**. Use caution before using in production.

See full reference on MDN Web Docs.

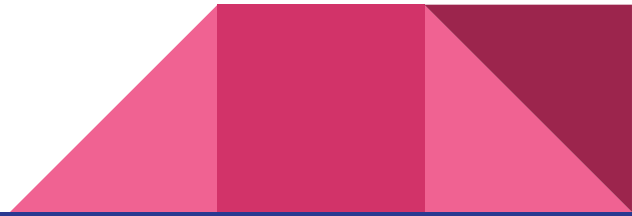
Support data for this feature provided by:

MDN browser-compat-data

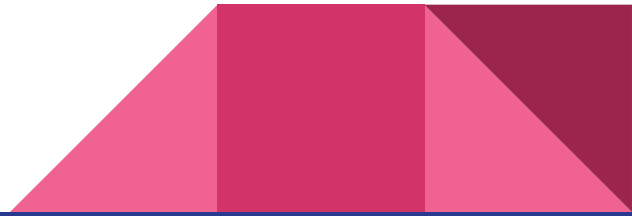
React.js Integration

- The View Transitions API can work alongside React, as long as one is using the `flushSync` import to force synchronous state changes.

- Otherwise, React Router is also another viable option, since its `useSyncExternalStore` API also performs synchronous rendering.



Closing Thoughts



Cited Works

Jake Archibald's Breakdown of View Transitions

https://developer.chrome.com/docs/web-platform/view-transitions/

https://www.youtube.com/watch?v=JCJUPJ_zDQ4

Malcolm Kee's guide to using View Transitions in React

https://malcolmkee.com/blog/view-transition-api-in-react-app/