To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Sign In Get started

You have **1** free member-only story left this month. Sign up for Medium and get an extra one

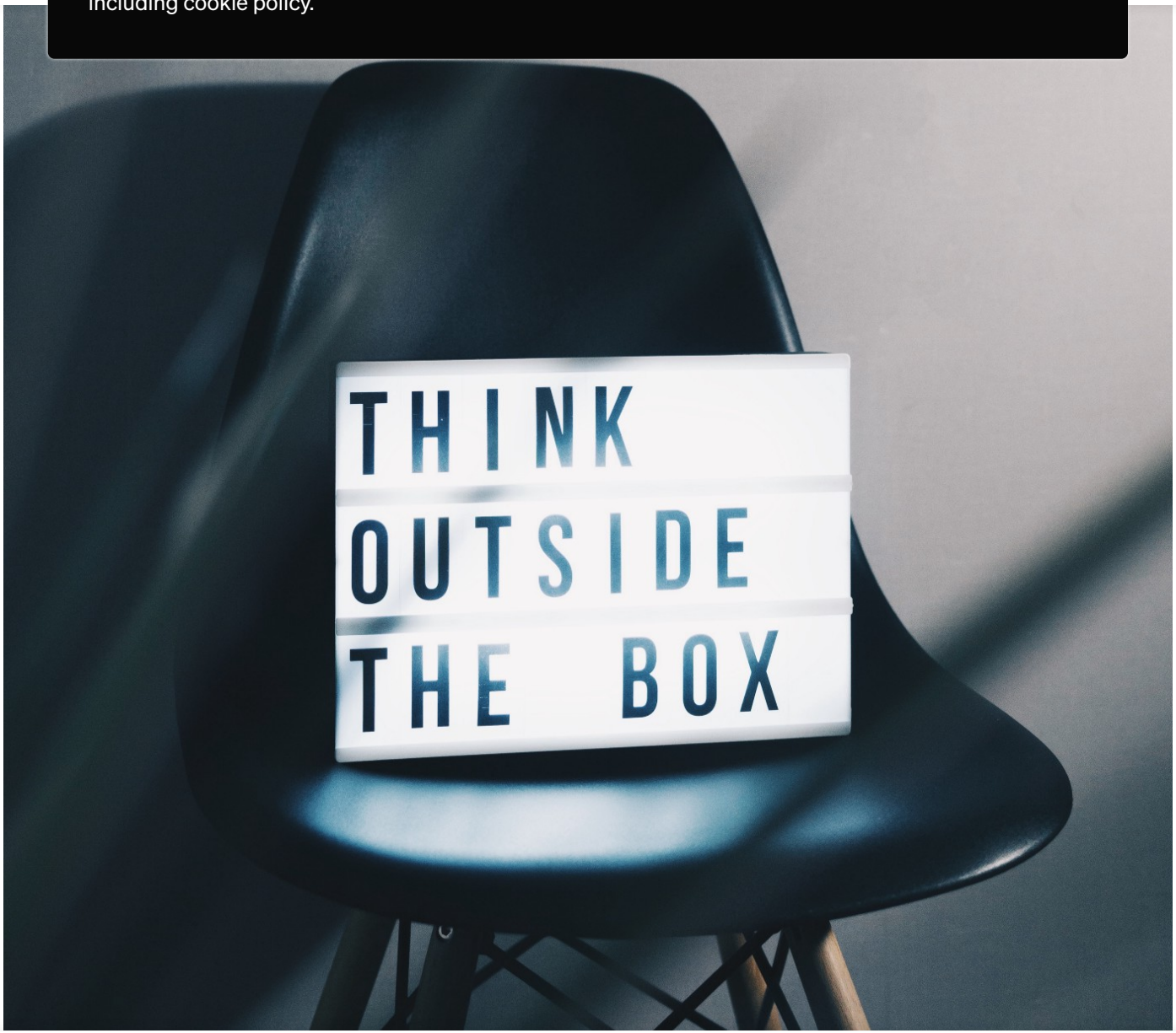
João Miguel Cunha Follow

Apr 10, 2018 · 5 min read ★ · Listen

Amazing React animation with react-pose

Dead easy animation with similar API to styled-components

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.



To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

> *No*
> ***upgrade if you haven't already***

Animation in React has always been a fuzzy subject. Many people don't even know how to start looking for something to help them build cool and fun animations for their web projects.

Of course there are a few packages that help us, like:

- react-motion — A very good and reliable one, I've used it and will probably use again eventually;
- react-animated — Never used because it always felt confusing to me but mostly because react-motion filled all my animation needs, but heard great things about this animation lib;
- react-transition-group — The "classic", probably the first one I tried when starting out with animations.

All these packages are great in their own way and they work! But I was looking for something new and something that just felt right. That is when I found **react-**

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Sign In Get started

Bu

You're absolutely right. CSS animations are great and you can do so many things with them. And please **USE** them! Js animations are useful in their own way.

- Use **css** animations if you're doing simple and self-contained animations. You can keep the state in your Javascript and the animations on your css.
- Use **javascript** animations if you want absolute fine-grain control over your animation. For example pausing animations or even changing animation values mid way like stiffness, damping and acceleration.

Read more about it here!

react-pose

This library is amazing. You can use it to create animated and interactive components that you can reuse throughout your React website.

> Declarative motion system for HTML, SVG & React

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Ins

```
$ npm install react-pose --save

// or

$ yarn add react-pose
```

And import it just you would **styled** from **styled-components**

```
import posed from "react-pose";
```

Let's go through some examples so you can learn how to use it!

I'll use styled-components to style my examples.

First example — Simple scale on hover

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

unl
poses when creating a posed Component.

```
const Square = posed.div({
  idle: { scale: 1 },
  hovered: { scale: 1.5 }
});
```

How we created a posed Component

As you can see we have the **idle** pose where the scale of the div is 1. We also define a **hovered** pose where we increase the size by 50% making it 1.5.

We apply these poses by defining a state of the component and setting the

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.


```
<Square
    pose={this.state.hovering ? "hovered" : "idle"}
    onMouseEnter={() => this.setState({ hovering: true })}
    onMouseLeave={() => this.setState({ hovering: false })}
/>
```

Rendering the square and setting the pose depending on the state

It’s this easy! 😃 💯 Try it out in the example below!

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Sign In

Get started

Second example — Draggable div with wavy and background color animation

In this example we can see how we can make a div become draggable on the **x** axis and alternate it's position and background color.

Let it inspire you in the example below!

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Sign In

Get started

Th

Can you imagine an inbox? Messages coming in and out. It would be cool for the new ones to animate to the top because they're the most recent messages, uh, and the new ones can fade in to the inbox.

Easy! This is all you need!

Our **Item** Component will be incoming or existing messages. When entering, the opacity is 1 and when leaving, the opacity is 0. The animation for the order of the messages is handled by our **PoseGroup** Component. All you really need is an unique key so that pose can handle it for you.

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.


```
  padding: 15px;
  list-style-type: none;
  margin: 5px 0px 5px 0px;
  border: 1px solid #e3e3e3;
`;

const ItemList = ({ items }) => (
  <ul>
    <PoseGroup>
      {items.map(item => (
       <StyledItem key={item.id}>
         {item.text}
       </StyledItem>
      ))}
    </PoseGroup>
  </ul>
```

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

See

List with re-ordering animation and fadeIn, fadeOut

Fourth example — Applying FLIP Technique

> *The FLIP technique, fully explained here, is a way of animating expensive, layout-breaking animations like width and top by using quick transforms*

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

For on of elements, pose handles those changes for you.

> *Pose will measure the size and position of the element before and after, and animate from one to the other using transform properties instead.*

With just this, you can have a simple clickable div become a fullscreen modal.

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.


```
const WonkyModal = posed.div({
  fullscreen: {
    width: "100vh",
    height: "100vh",
    transition: tween
  },
  idle: {
    width: 100,
    height: 100,
    transition: tween
  }
});
```

WonkyModal code

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

cas

doing this with js you can even apply some transformations to your image so that it properly adjusts to fullscreen.

Applying FLIP technique

Final example — Customizing transitions

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Sign In

Get started

Wi

transition time for both poses independently.

```
const Square = posed.div({
  idle: {
    scale: 1,
    opacity: 0.5,
    transition: props => tween({ ...props, duration: 2000 })
  },
  hovered: {
    scale: 2,
    opacity: 1,
    transition: props => tween({ ...props, duration: 1000 })
  }
});
```

When I hover the Square I want it to scale to 2 and the opacity to 1 and, all of this should be done in 1 second. When I stop hovering the square and it goes back into **idle** pose, I want it to go back to normal in 2 seconds.

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Customizing transitions

Conclusion

I've shown you some examples of what you can do. Now it's your turn to try this out!

- Their docs are very good!

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

If y are down below in the responses! 💯 💯

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Sign In

Get started

. . .

If you want to stay updated on everything I write please feel free to Follow me!

João Miguel Cunha — Medium

Read writing from João Miguel Cunha on Medium. Full-Stack @ Wiremaze SA. ❤ JS and React. Every day, João Miguel Cunha...

medium.com

And if you want to see more stuff from me I can recommend these posts:

Using toolkits to jumpstart React development 🛠️

Setup a React application in just a few seconds 📦

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

L

Still experimental but we can start building fun stuff 😃

medium.com