

With Hooks

Preferred for performance and ergonomics

```
import { useSpring, animated } from 'react-spring';

function Box({ x, y }) {
  const style = useSpring({
    transform: `translate(${x}px, ${y}px)`,
  });

  return (
    <animated.div style={style}>
      Moving box!
    </animated.div>
  );
}
```

With Render Props

Flexible, easier data access

```
import { Spring } from 'react-spring/renderprops';

function Box({ x, y }) {
  return (
    <Spring
      to={{ transform: `translate(${x}px, ${y}px)` }}
    >
      {style => (
        <div style={style}>
          Moving box!
        </div>
      )}
    </Spring>
  );
}
```

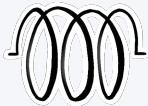
Configuration

Change the mechanics of the animation



mass

Heavier items
move more slowly



tension

Stiff springs release
more energy



friction

High friction
dampens the effect

Interpolation

Direct access to useSpring values. Escape hatch.

```
const spring = useSpring({
  progress: isOpen ? 1 : 0,
});

return (
  <animated.div
    style={{
      background: spring.progress.interpolate(p =>
        `rgba(210, 57, 77, ${p})`
      ),
    }}
  >
  </>
);
```

React Spring

Cheatsheet for v8.0.27

Orchestration

Coordinating between multiple elements

useTrail

Stagger the same transition across multiple elements

```
function Trail({ x, y }) {
  const trail = useTrail(4, {
    transform: `translate(${x}px, ${y}px)`
  });

  return trail.map(style => (
    <animated.div style={style} />
  ));
}
```

useChain

Sequence multiple springs however you'd like!

```
function Chain({ x, y }) {
  const firstRef = React.useRef();
  const spring1 = useSpring({
    transform: `translate(${x}px, ${y}px)`,
    ref: firstRef
  });

  const secondRef = React.useRef();
  const spring2 = useSpring({
    transform: `translate(${x}px, ${y}px)`,
    ref: secondRef
  });

  useChain([firstRef, secondRef]);

  /* Use springs normally */
}
```

Simulating keyframes

Multi-step animation on mount

```
const style = useSpring({
  from: { opacity: 0 },
  to: [
    { opacity: 1 },
    { opacity: 0 },
  ],
});
```

Callbacks

Sequencing with other parts of your React app

```
const style = useSpring({
  transform: `translate(${x}px, ${y}px)`,
  onRest: ({ transform }) =>
    // Handle the end of the transition
});
```

Protip: Tweak config.precision for more accurate timing