

ANIMATING the VIRTUAL DOM

Sarah Drasner
@sarah_edo



Sarah Drasner
@sarah_edo

SR. CLOUD DEVELOPER ADVOCATE



Microsoft



FIBER

Priority-based scheduling of updates
Animation update preferred to data update
Animation becomes a first-class citizen

“We’ve evolved to perform actions that flow more or less seamlessly.

“We aren’t wired to deal with the fits and starts of human-computer interaction.”

-Tammy Everts

Sensory memory: Your occipital lobe (AKA “the memory store”) works in 100ms bursts.

source

REACT ENCAPSULATES STATE
THAT'S CHANGING

ANIMATION TIES THESE TWO
STATES TOGETHER

USERS CAN THEN FOLLOW THE
TRAJECTORY OF THE CHANGE AS IF IT
WERE A

STORY

THE "SO WHAT" FACTOR

User attention span is short.

2 SECONDS

until dropoff

Amazon has discovered that for every one second delay, conversions dropped by 7%. If you sell \$100k per day, that's an annual loss of \$2.5m.

Walmart has found that it gains 1% revenue increase for every 100ms of improvement.

Source

START WITH THE END



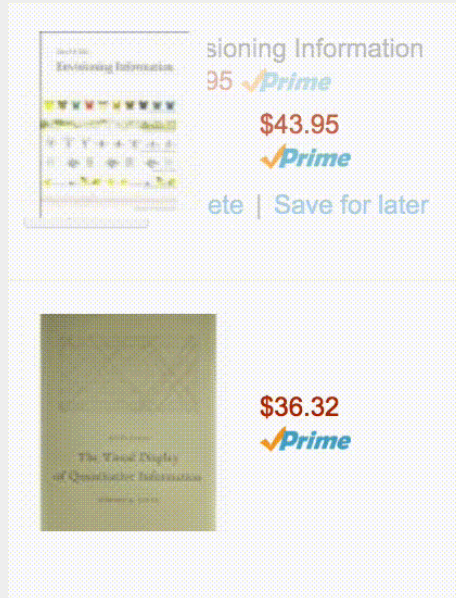
Also helps track what validation looks like:
Google PMs are encouraged to find ONE goal.

FILL IN THE SPACE BETWEEN

Aligns the entire team to the same goal
Reduces friction



WITHOUT TRANSITIONS



GAIN UNDERSTANDING

Spatial or otherwise



Codrops

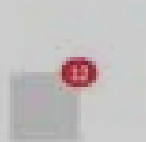
W | Workflo Components

Shared components for the entire org



SHARE

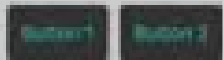
Badge



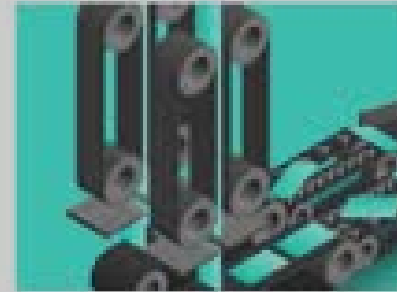
Button



ButtonGroup



CardImage



ContentCard



FloatingButton

PERCIEVED PERFORMANCE

- Humans over-estimate passive waits by 36% - **Eli Fitch** and Richard Larson, MIT
- Viget study: wait twice as long for custom loader
- Over 4 seconds: **HORROR**
- Uncertain waits are longer than known, finite waits
- Disney world/Airports- entertainment while waiting
- People want to get started- doctor exam rooms

Goran Rakic from **Loader Collection**

Entire filesize: 6KB!

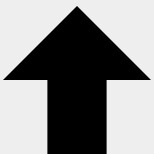
SVG (basic support) - REC

Global

97.04% + 0.75% = 97.79%

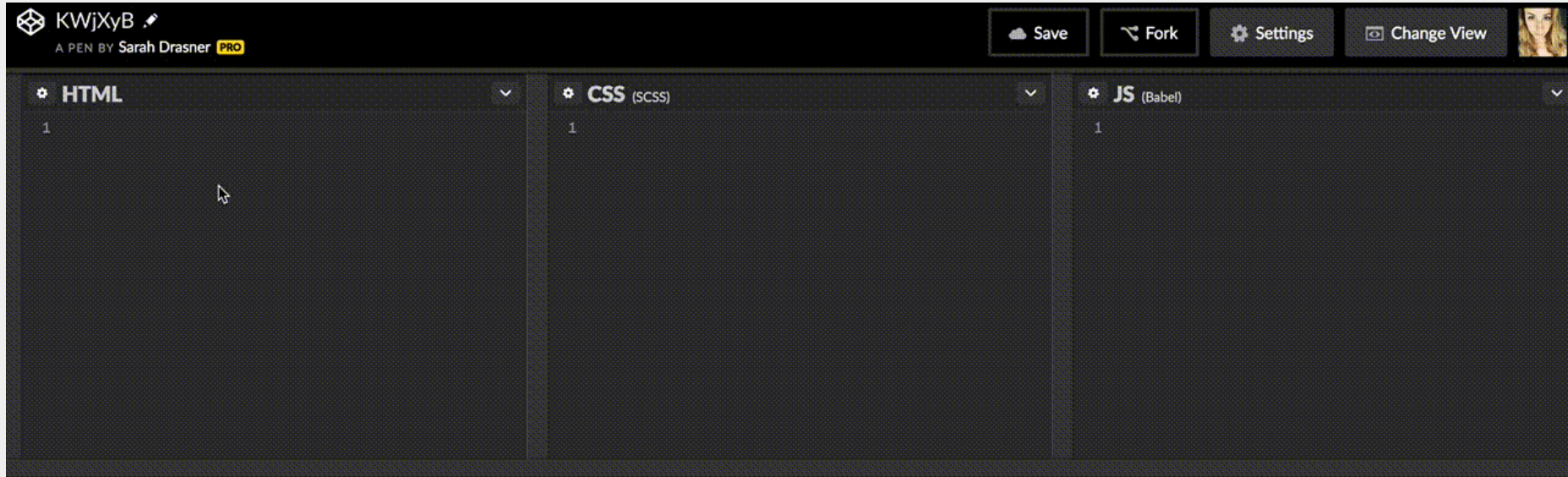
Method of displaying basic Vector Graphics features using the embed or object elements. Refers to the SVG 1.1 spec.

Current aligned	Usage relative	Date relative	Show all						
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49					¹ 4.3	
			54					4.4	
		50	55			9.3		4.4.4	
² 11	² 14	51	56	10	42	10.2	all	53	56
	² 15	52	57	10.1	43				
		53	58	TP	44				
		54	59						



This actually goes
back to IE9

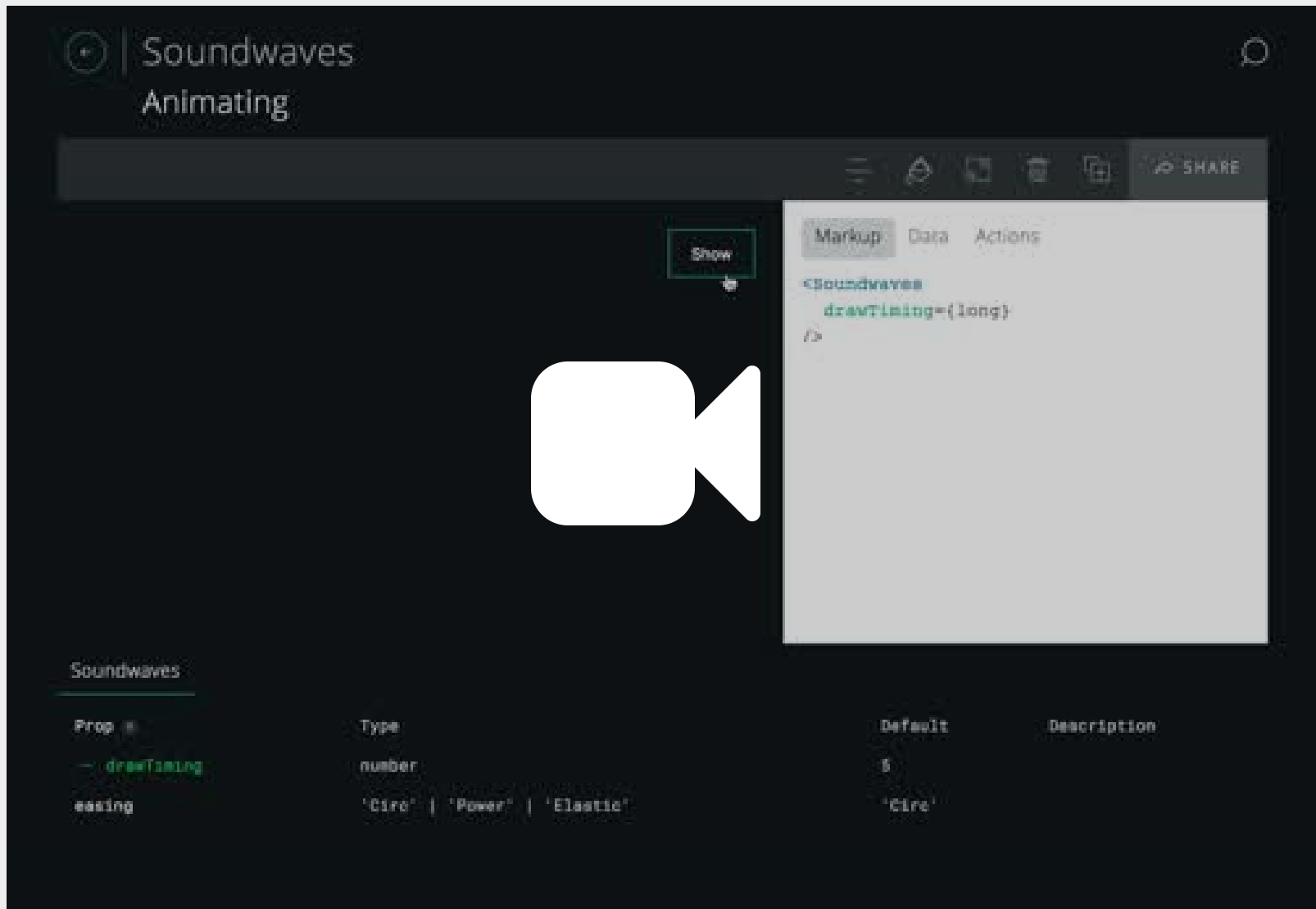
WHAT DOES THE "SCALABLE" MEAN?



You don't have to care about **positioning**,
even for responsive

DECLARATIVE ANIMATIONS

TIMING

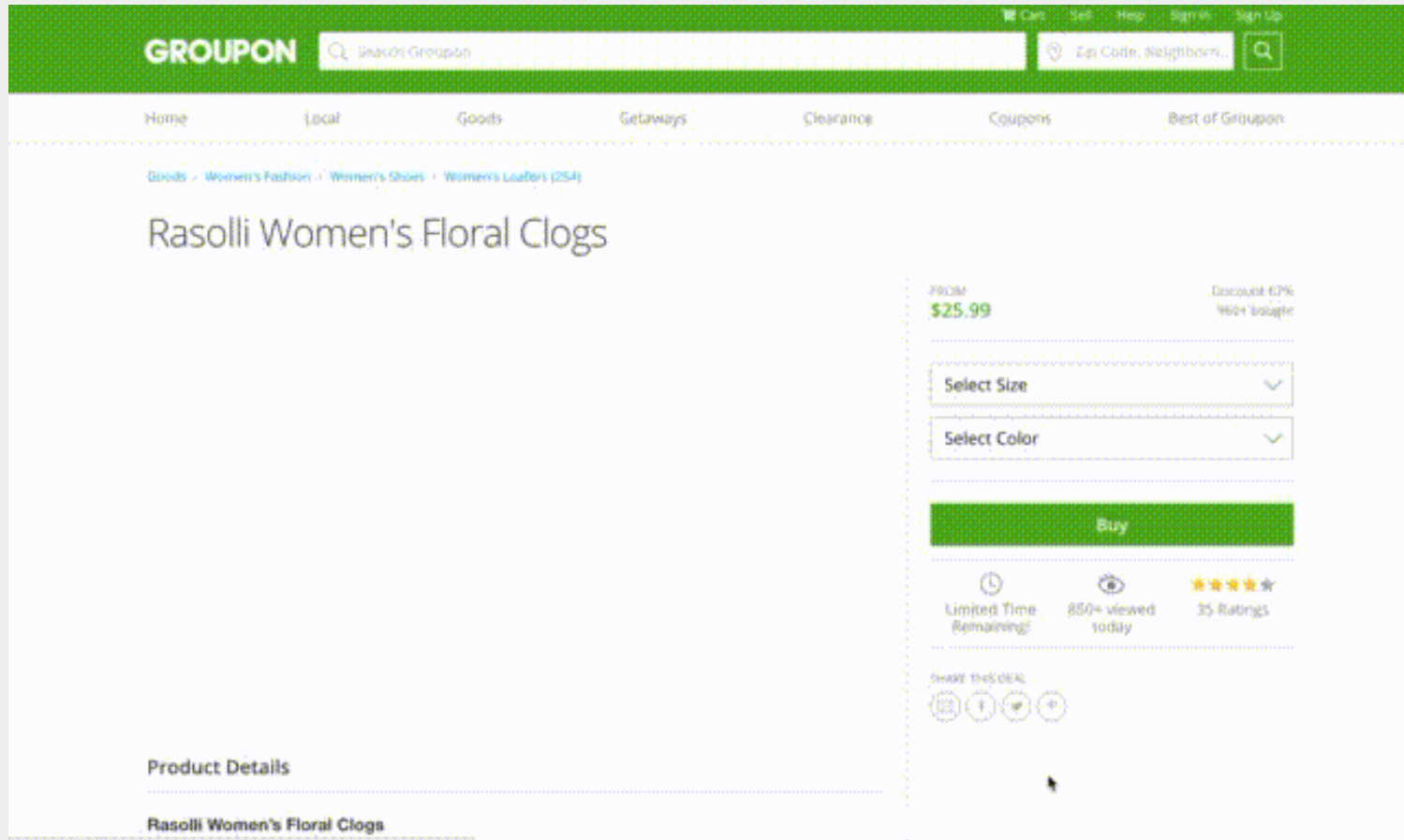


TIMING

```
<TransitionGroup>
  { this.state.shouldShowSoundwaves &&
    <Soundwaves
      outTime={0.5}
      drawTiming={5}
      elTime={1}
      easing="Circ"
    />
  }
</TransitionGroup>
```

TIMING

Consider how things look to your user

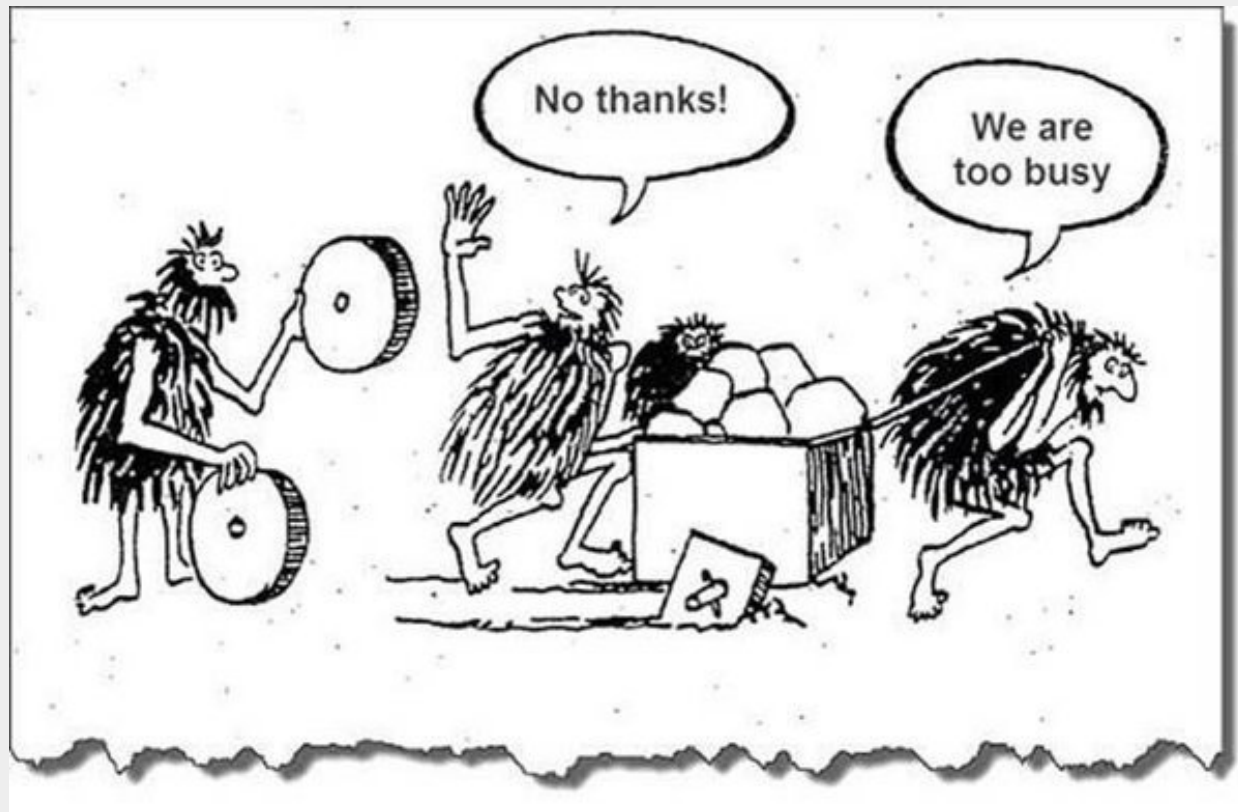


Impacts your SEO with Google



CREATE BEAUTIFUL DEFAULTS

Keep animation consistent and reusable
the way you do with text, layout




```
const Eases = {
  entrance: {
    animationTimingFunction: `cubic-bezier(0.39, 0.575, 0.565, 1)`,
  },
  entranceEmphasis: {
    animationTimingFunction: `cubic-bezier(0.175, 0.885, 0.32, 1.275)`,
  },
  exit: {
    animationTimingFunction: `cubic-bezier(0.47, 0, 0.745, 0.715)`,
  },
  exitEmphasis: {
    animationTimingFunction: `cubic-bezier(0.6, -0.28, 0.735, 0.045)`,
  },
}
```

h1, h2, h3, h4, **h5 is body copy**

t1, t2, t3, t4, **t5 is typical timing**

```
const Timing = {  
  t1: {  
    animationDuration: 0.1,  
  },  
  t2: {  
    animationDuration: 0.15,  
  },  
  t3: {  
    animationDuration: 0.2,  
  }...  
}
```

ANIMATION ITERATION



Animation



Animates children using tunable parameters

SHARE

Appear In/Out



Trigger

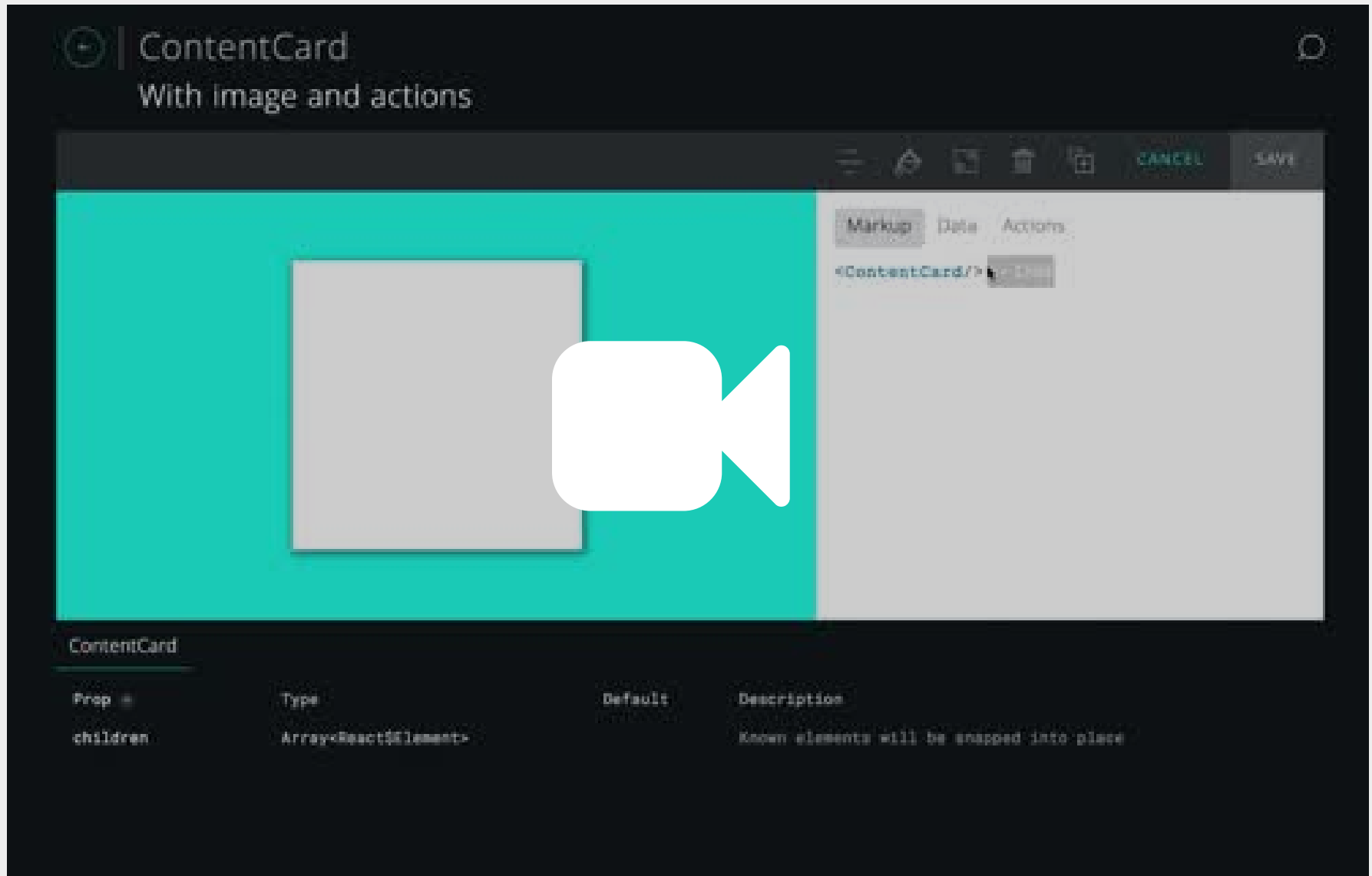
Enter/Exit clockwise

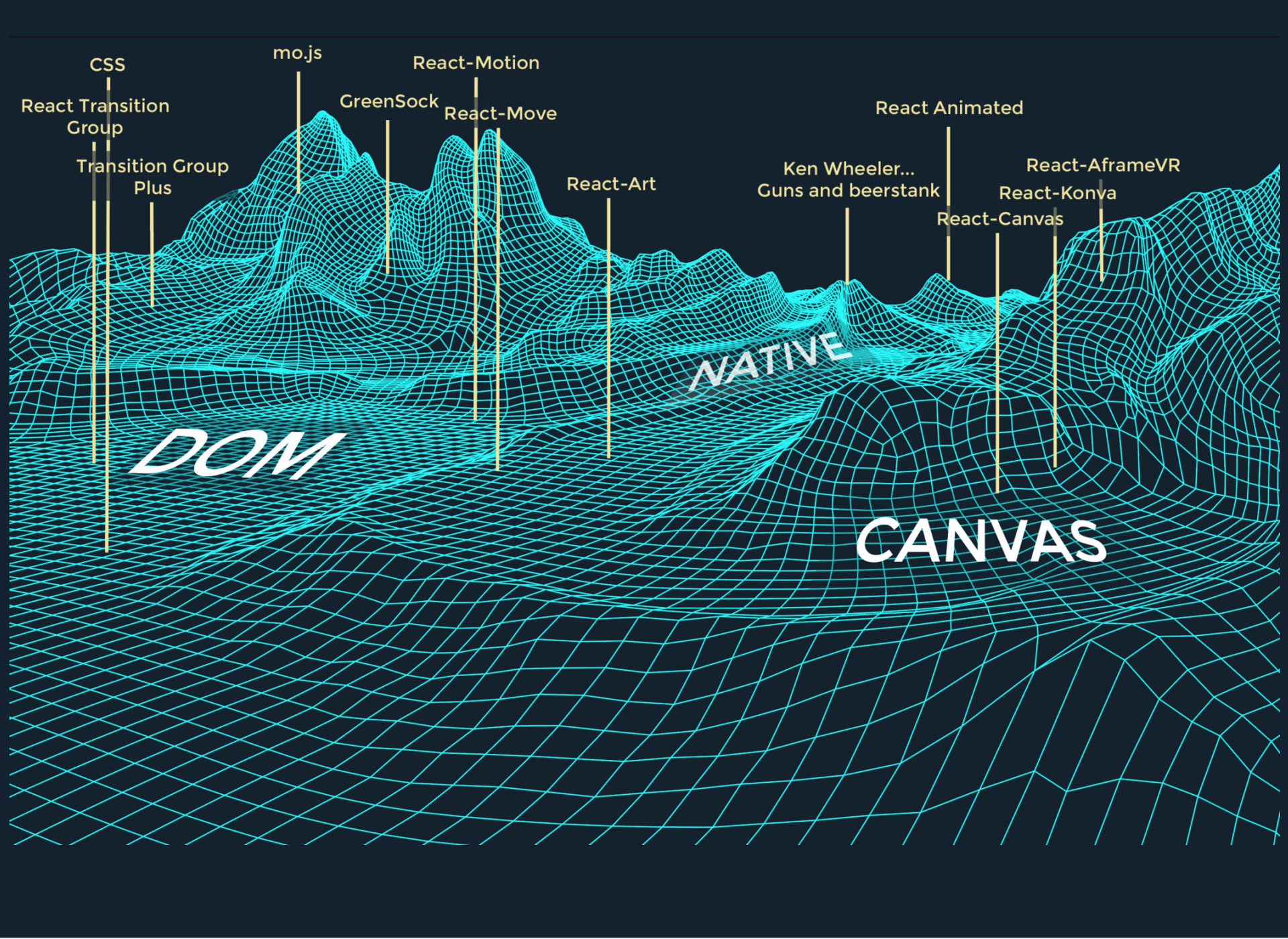


Enter/Exit from left

Trigger

COMPOSITE COMPONENTS, INCLUDING ANIMATION





CSS

mo.js

React-Motion

React Transition Group

GreenSock

React-Move

React Animated

Transition Group Plus

React-Art

Ken Wheeler... Guns and beerstank

React-AframeVR

React-Konva

React-Canvas

DOM

NATIVE

CANVAS

MO.JS

```
• JS  
1 const burst = new mojs.Burst().play();
```

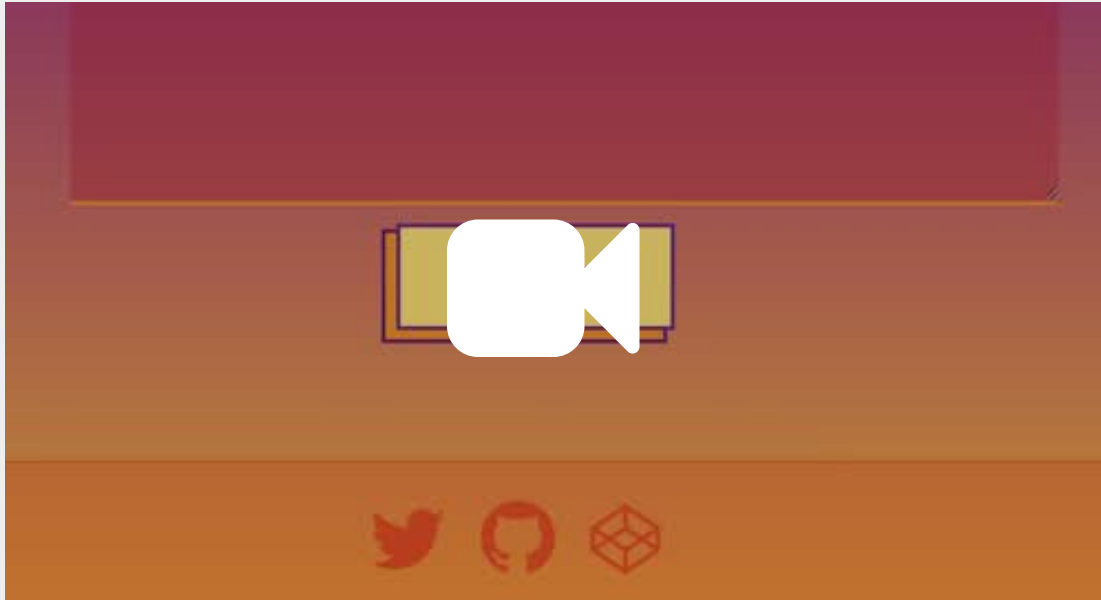


MO.JS

```
HTML
CSS (SCSS)
JS (Babel)

10
11 const zigzag = new mojs.Shape({
12   shape:      'zigzag',
13   points:      12,
14   radius:      25,
15   radiusY:     50,
16   top:         pos.row1,
17   left:        pos.col1,
18   fill:        'none',
19   stroke:      color1,
20   isShowStart: true,
21 });
22
23 const cross = new mojs.Shape({
24   shape:      'cross',
25   radius:      20,
26   radiusY:     30,
27   top:         pos.row1,
28   left:        pos.col2,
29   stroke:      color2, //no fill for this one
30   isShowStart: true,
31   y:          -25
32 });
33
```





```
const burst = new mojs.Burst({
  left: 0, top: 0,
  zIndex: 10000,
  radius: { 0: 100 },
  count: 5,
  children: {
    shape: 'cross',
    stroke: { 'cyan' : 'yellow' },
    angle: { 360: 0 },
    duration: 2000,
    delay: 'stagger(0, 100)'
  }
});
```


MO.JS

This **pen**.



```
class ButtonGroup extends React.Component {
  render() {
    return (<div></div>);
  },

  shouldComponentUpdate() {
    this.props.isPlaying && this._burst.replay();
    return false;
  },

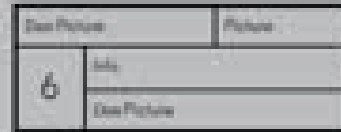
  componentDidMount () {
    const dog1 = new mojs.Burst({
      ...dog_opts,
      children: {
        ...dog_child_opts,
        shape:      'dogbase',
        fill:        '#E89221'
      }
    });

    ...
  }
}
```

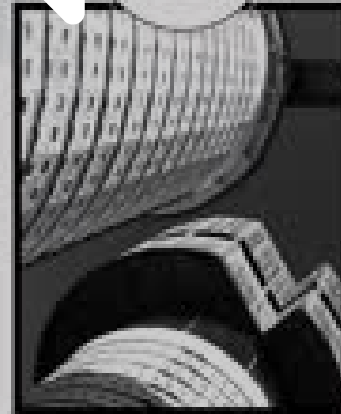
REACT-MOTION

This pen





Name: _____

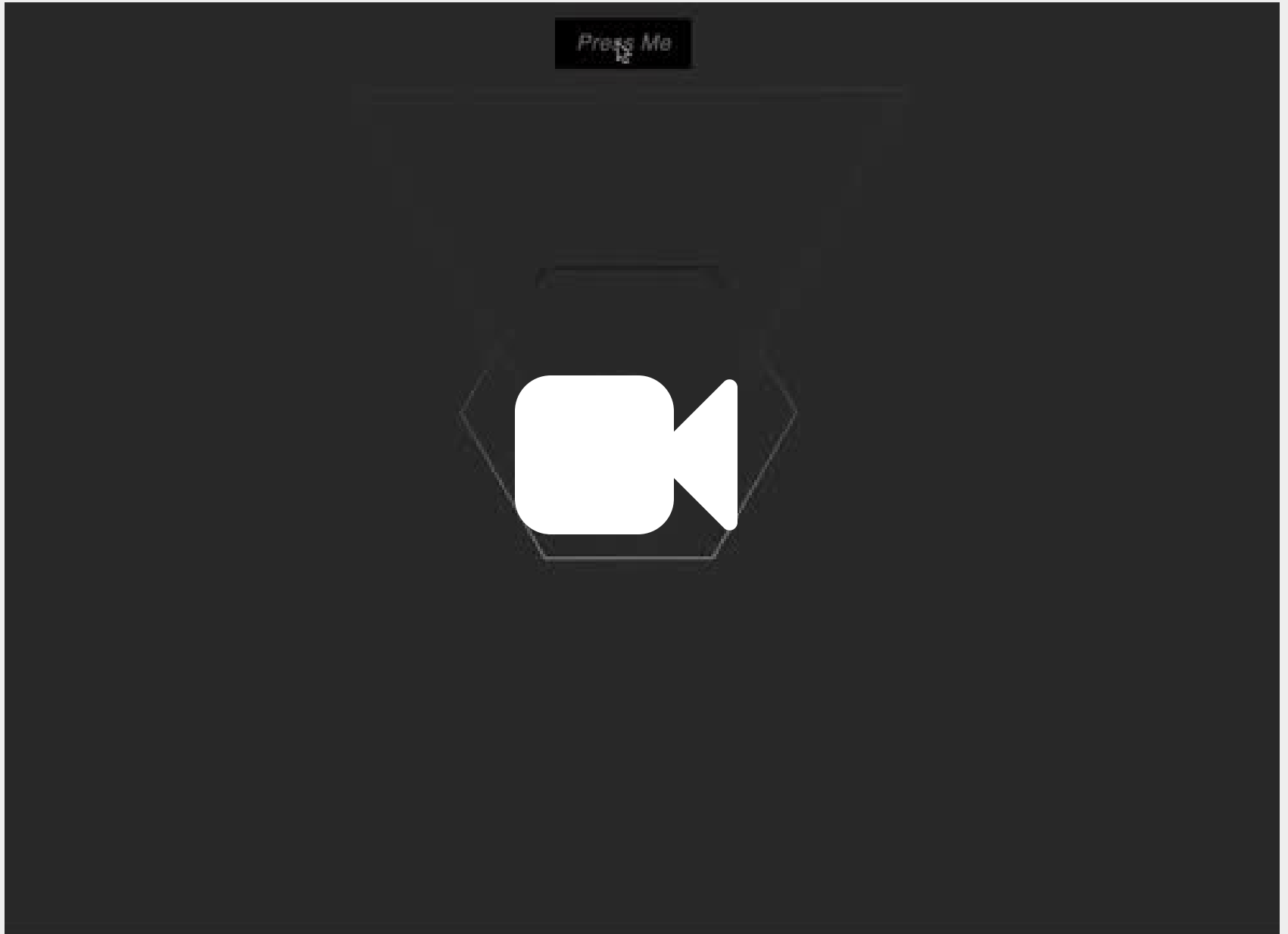


100

```
getStyles(prevStyles) {  
  const endValue = prevStyles.map((_, i) => {  
    let staggerStiff = 100, staggerDamp = 19;  
    return i === 0  
      ? { opacity: spring(this.state.open ? 0 : 1, {stiffness: staggerStiff, dampin  
      : { opacity: spring(this.state.open ? 0 : 1, {stiffness: (staggerStiff - (i *  
  });  
  return endValue;  
},
```

```
<StaggeredMotion
  defaultStyles = {arr}
  styles={this.getStyles}>
  {circ =>
    <g fill={pathColor} className="cPath">
      {circ.map(({opacity}, i) =>
        <path
          key={i}
          d={pathData[i]}
          style={{
            opacity: opacity
          }} />
      )}
    </g>
  }
</StaggeredMotion>
```

This pen.



DRAWN SVG

Done with stroke-dasharray and stroke-dashoffset

- Path or shape has a stroke
- The stroke is dashed
- Use JS to .getTotalLength()
- Dasharray the whole length of the shape
- Animate dashoffset

```
@keyframes dash {  
  50% {  
    stroke-dashoffset: 0;  
  }  
  100% {  
    stroke-dashoffset: -274;  
  }  
}
```



```

<Motion style={{
  //designate all of the differences in interpolated values in these ternary opera
  ...
  dash: spring(this.state.compact ? 0 : 200),
  ...
}}>

  {/* make sure the values are passed below*/}
  ({..., dash, ...}) =>
  <svg viewBox="0 0 803.9 738.1" aria-labelledby="title">
    <title>React-Motion</title>

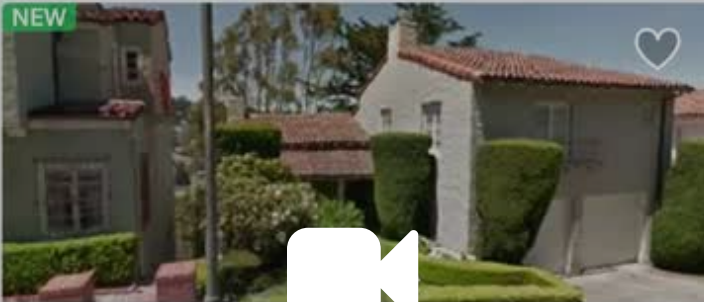
    ...
    <g style={{ strokeDashoffset: `${dash}` }}
      className="react-letters" data-name="react motion letters">
        <path className="cls-5" d="M178.4,247a2.2,2.2,0,1,1-3.5,2.6l-6.5-8.7h-8.6
        ...
      </g>
    </svg>
  }
</Motion>

```

San Francisco, CA



Filters



\$2,395,000

3bd 2ba 3,115 sqft

60 Miraloma Dr

Saint Francis Wood, San Francisco, CA

NEW CONSTRUCTION



CORRECT TOOLS FOR THE JOB

MY RECOMMENDATIONS:

CSS-in-JS/Styles

- Small sequences and simple interactions
- Once you get more than 3... switch to:

GSAP (GreenSock)

- Great for sequencing and complex movement
- Cross-browser consistency

React-Motion

- Great for single movements that you'd like to look realistic

- [React-Tween](#) similar to React-Motion: time instead of spring
- [React-Move](#) similar to React-Motion: small, flexible
- [Snap.svg](#) is more like jQuery for SVG
- [Web Animations API](#) looks great, still waiting on support
- [Velocity-React](#) is similar to GSAP with less bells and whistles
- [Mo.js](#) spins up shapes declaratively and is still in beta

CORRECT TOOLS FOR THE JOB

GSAP (GreenSock)



IT'S **VERY TOUGH** TO FIND SOMETHING ANOTHER TOOL CAN DO THAT GSAP CAN'T DO...

BUT THERE ARE PLENTY OF THINGS GSAP CAN DO THAT
OTHER TOOLS CAN'T.

- Transforms
- Cross-browser inconsistencies
- Morphing
- Sequencing and chaining

NOT ALL ARE CREATED EQUAL

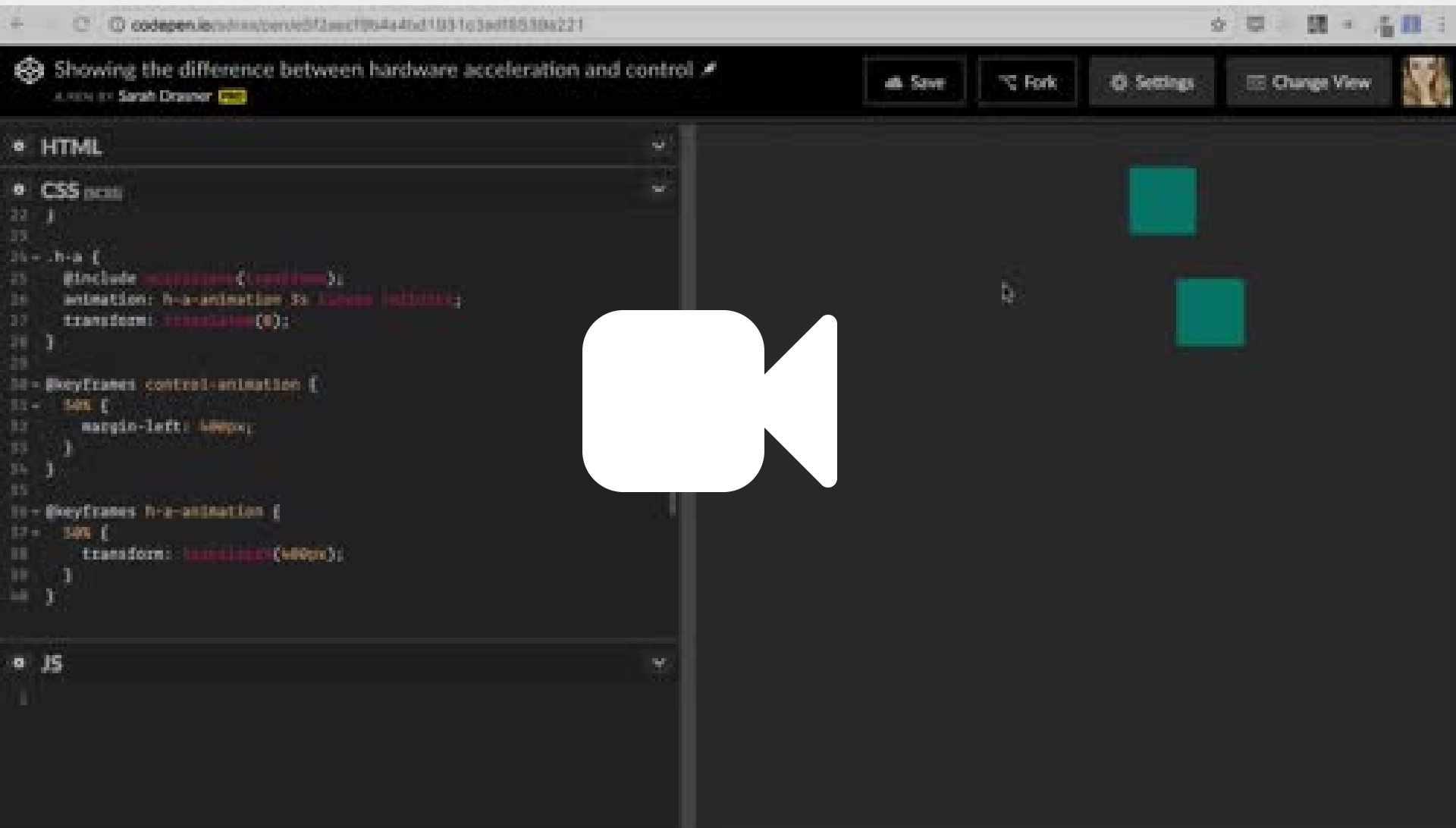
- Opacity
- Transforms
- Hardware Acceleration

```
const accelerate = {  
  transform: 'translateZ(0)',  
  backface-visibility: 'hidden',  
  perspective: '1000px',  
}
```

Resources:

Debugging Keyframe Animations

High Performance Animations



STACKING TRANSFORMS

```
/*          3rd          2nd          1st          */  
transform: rotate(90deg) translateX(30px) scale(1.5);
```


STACKING TRANSFORMS

womp womp :(

```
@keyframes foo {  
  30% {  
    transform: rotateY(360deg) scale(1.23) translateY(-14px)  
  }  
  65% {  
    transform: rotateY(-360deg) scale(1.5) translateY(-30px)  
  }  
  90% {  
    transform: rotateY(-102deg) scale(0.75) translateY(10px)  
  }  
}
```

GREENSOCK

```
TweenMax.to('.thing', 2, {  
  x: 20,  
  y: 100,  
  scale: 20,  
  skew: 2  
})
```

Don't have to write transform: translateX... etc either.
Or hardware accelerate.

Not beholden to the spec



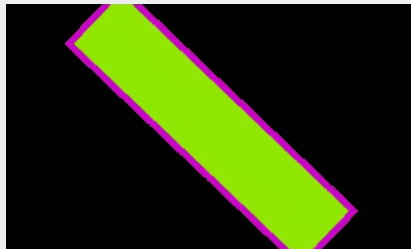
*React-Move/React-Motion/React-Tween can handle
matrix transforms, it's just not very intuitive

Animate the unanimatible, like SVG filters

SOLVES CROSS-BROWSER INCONSISTENCIES

Bad transform origin bug on rotation, soon
to be solved in Firefox.

More in this [CSS-Tricks article](#).



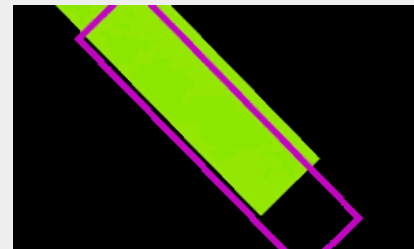
Chrome



IE



Firefox



Safari (zoomed)

SOLVES OTHER TRANSFORM-ORIGIN ISSUES

This pen.

`smoothOrigin:true`
element doesn't jump when origin changes.

`smoothOrigin:false`
element jumps when origin changes.

UP

UP

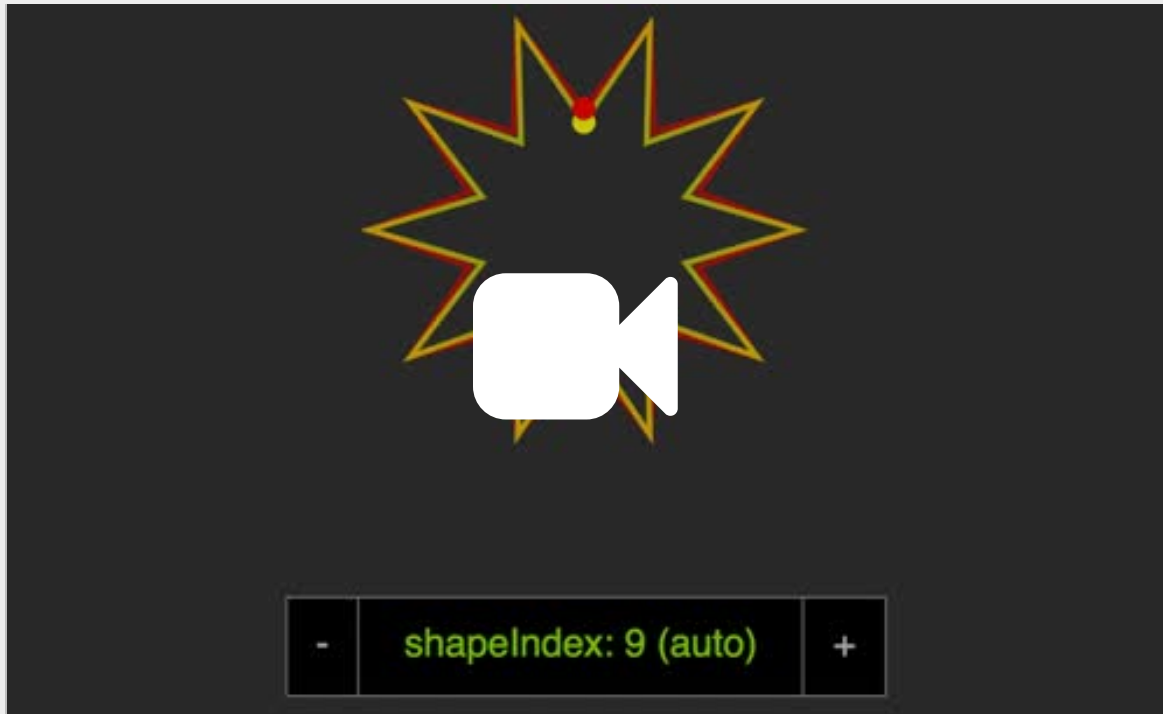
RESTART

Initial scale of the boxes is `scale:2` with their transform origin in the upper left corner, `transformOrigin:"left top"`. The red circles indicate the transformOrigin position.

`smoothOrigin:false` is actually the "proper" implementation based on the SVG and CSS specs, but `smoothOrigin:true` looks more natural and smooth.

MORPHSVG

Edit this.



SEQUENCES & CHAINING

Easily spaghetti:
Delays for chaining,
Callbacks

TIMELINE

- stack tweens
- change their placement in time
- group them into scenes
- add relative labels
- animate the scenes!
- reuse the scenes, play backwards, pause

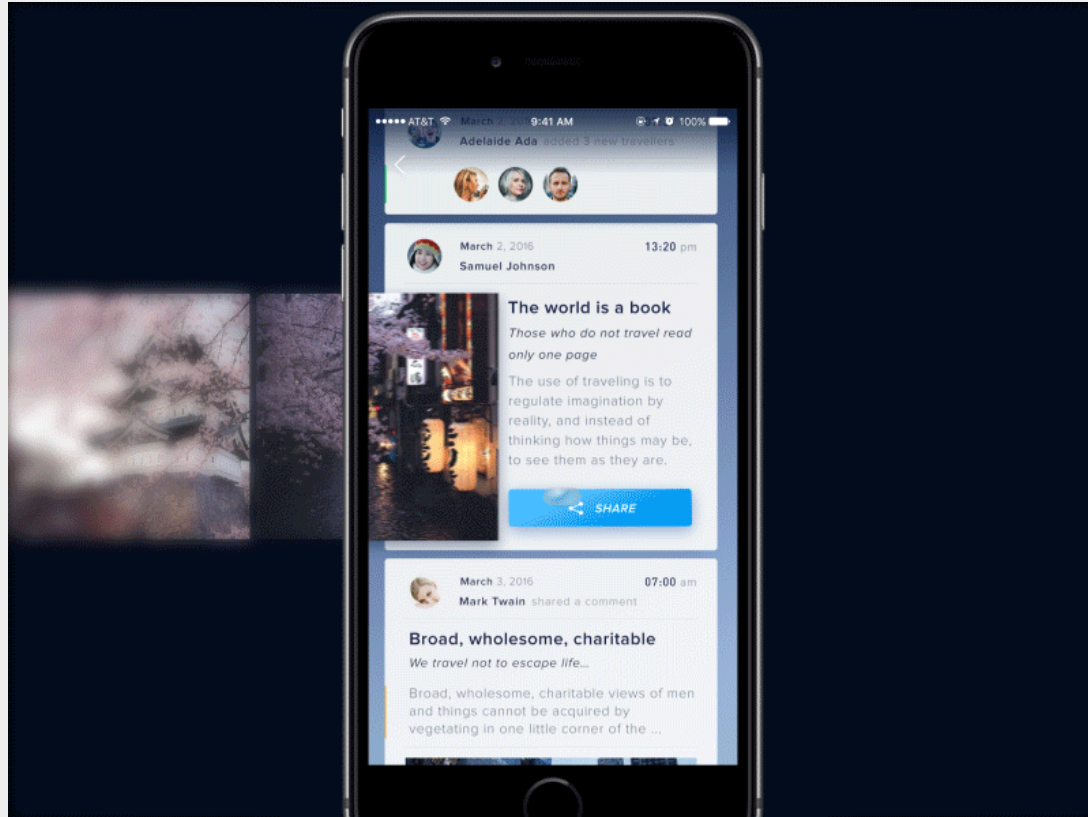
```
//instantiate a TimelineLite
const tl = new TimelineMax();

//adds a tween at the beginning
tl.to(el1, 0.5, {x:100, opacity:0})

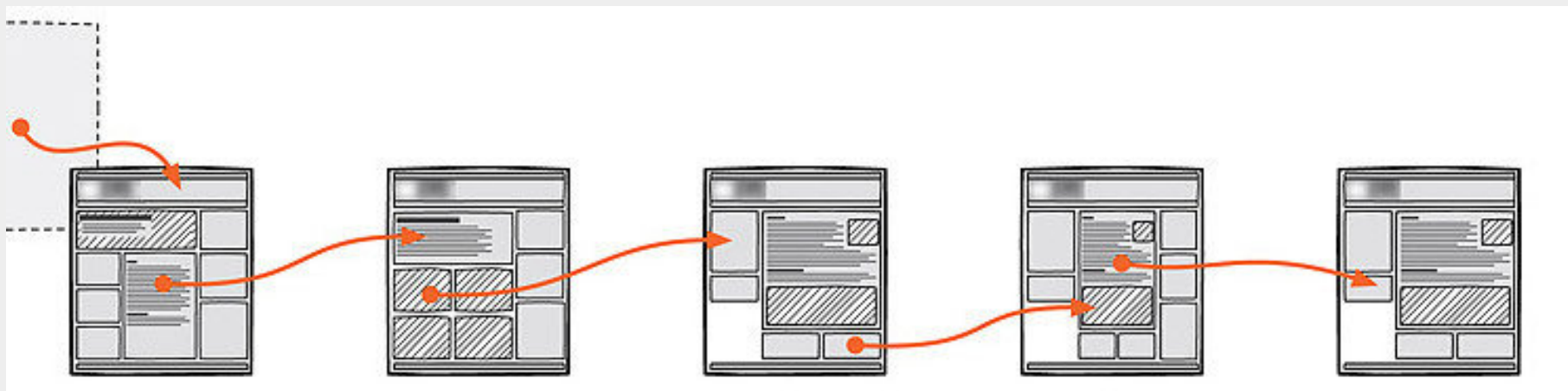
//adds another tween immediately after
tl.to(el2, 0.5, {y:-100, opacity:0})

//schedule next tween 0.5 seconds after
tl.to(el3, 0.5, {scale:.5}, "+=0.5")
```

END TO END



Leo Leung



REACT TRANSITION
GROUP **PLUS**
&&
GREENSOCK

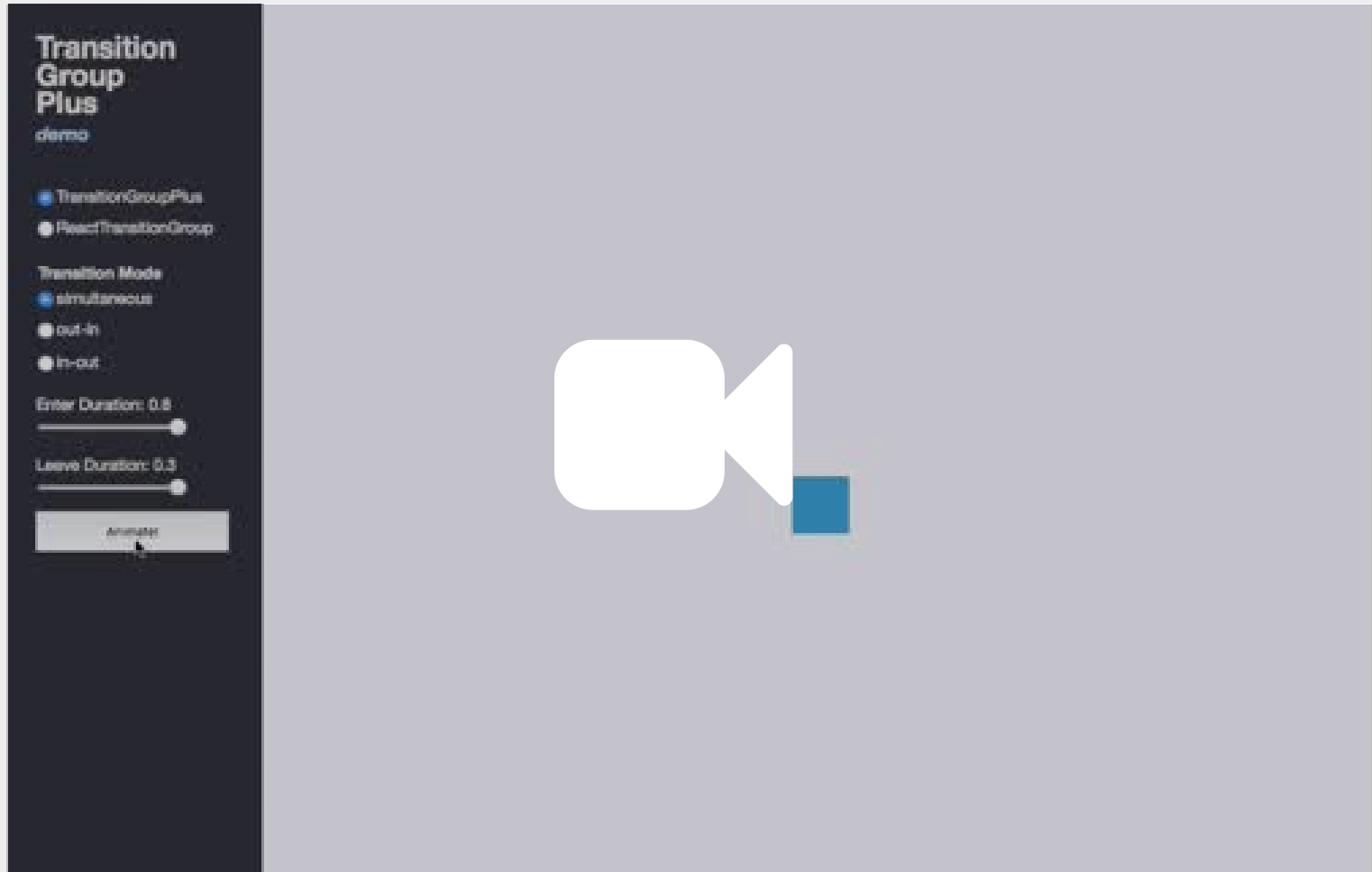
==



TRANSITIONS

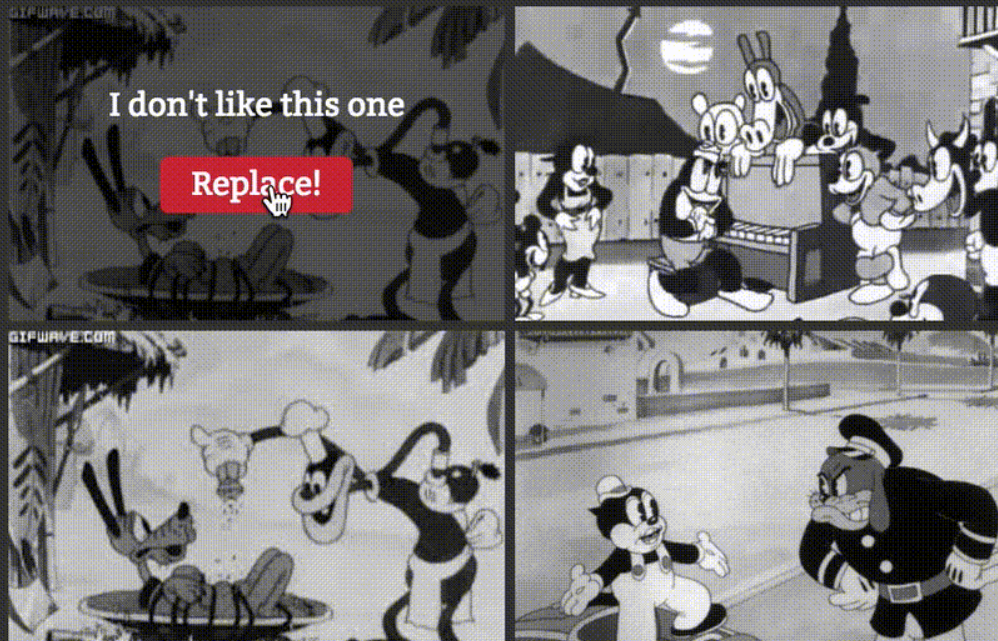
REACT TRANSITION GROUP PLUS

Don't reinvent the wheel / Avoid callback hell



In-out modes

Hover us.



Without in-out modes

Hover us.



END TO END

Encapsulate what is changing



STATE-DRIVEN ANIMATIONS

```
toggleShape() {  
  if (this.state.screen === 0) {  
    this.animFire(this.state.splitText)  
  } ...  
  this.setState({  
    screen: (this.state.screen + 1) % 3  
  })  
}
```

STATE-DRIVEN ANIMATIONS

```
animFire(splitText) {  
  const tl = new TimelineMax,  
    stA = 'start';
```

```
  TweenMax.set([this.g1.childNodes, this.g2.childNodes], {  
    clearProps: 'svgOrigin'  
  })
```

```
  tl.add('start')  
  tl.staggerFromTo(this.g1.childNodes, dur, {  
    drawSVG: '68% 100%'  
  }, {  
    drawSVG: '27.75% 0%',  
    ease: Back.easeOut  
  }, stD, stA)  
  ...  
}
```



OSS Version

```
componentWillEnter(callback) {  
  const { drawTiming, elTime, easing } = this.props,  
        elTime2 = elTime * 2;  
  
  TweenMax.fromTo(this.box, elTime2, {  
    opacity: 0,  
    drawSVG: '50% 50%'  
  }, {  
    opacity: 1,  
    drawSVG: '100%',  
    ease: `${easing}.easeOut`  
  })  
  ...  
}
```

More!

Draggable

Motion Along a Path

Custom Easing

Physics/ThrowProps

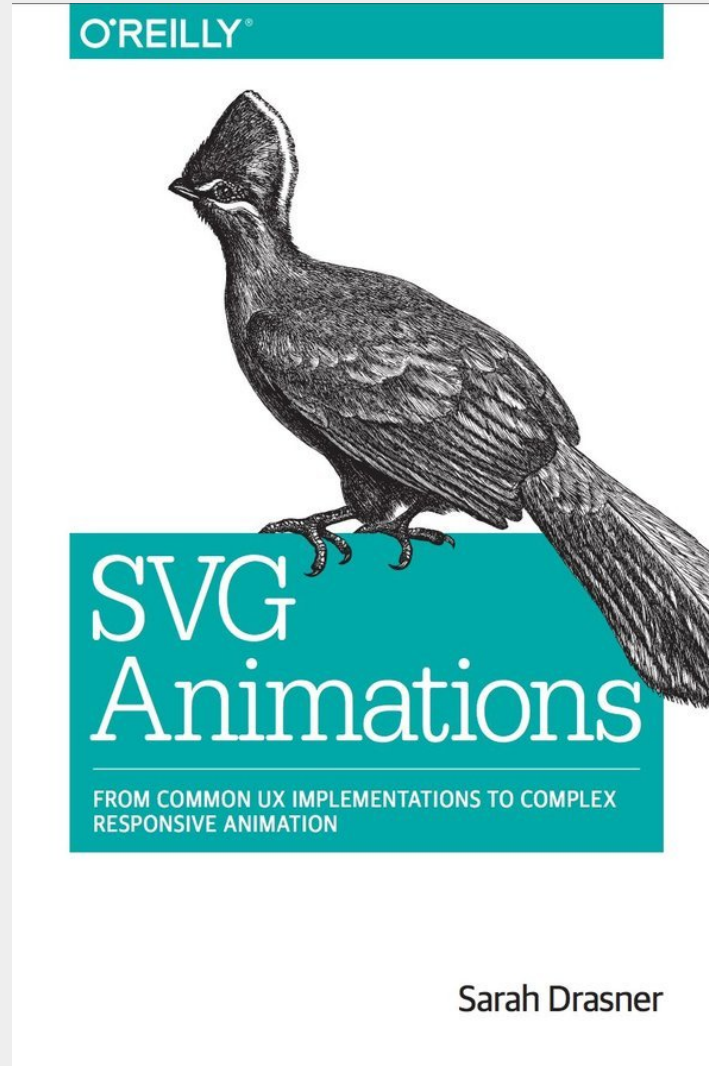
THE SKY'S THE LIMIT!

REACT ENCAPSULATES STATE
THAT'S CHANGING

ANIMATION TIES THESE TWO
STATES TOGETHER



SVG ANIMATIONS





WEB ANIMATION WORKSHOPS

with Val Head

THANK YOU!

@sarah_edo on twitter