Practical Web Animation

Lisi Linhart



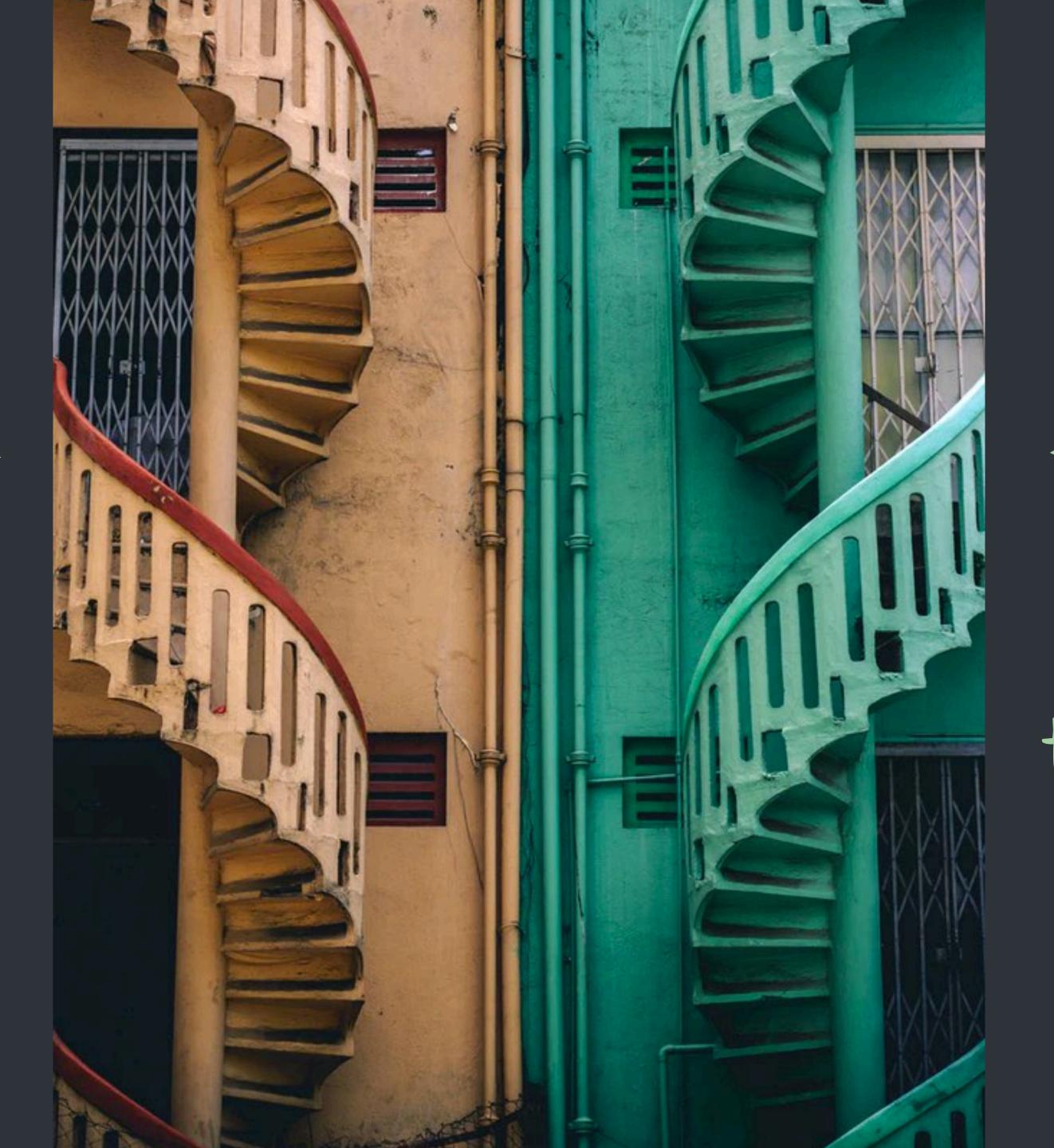




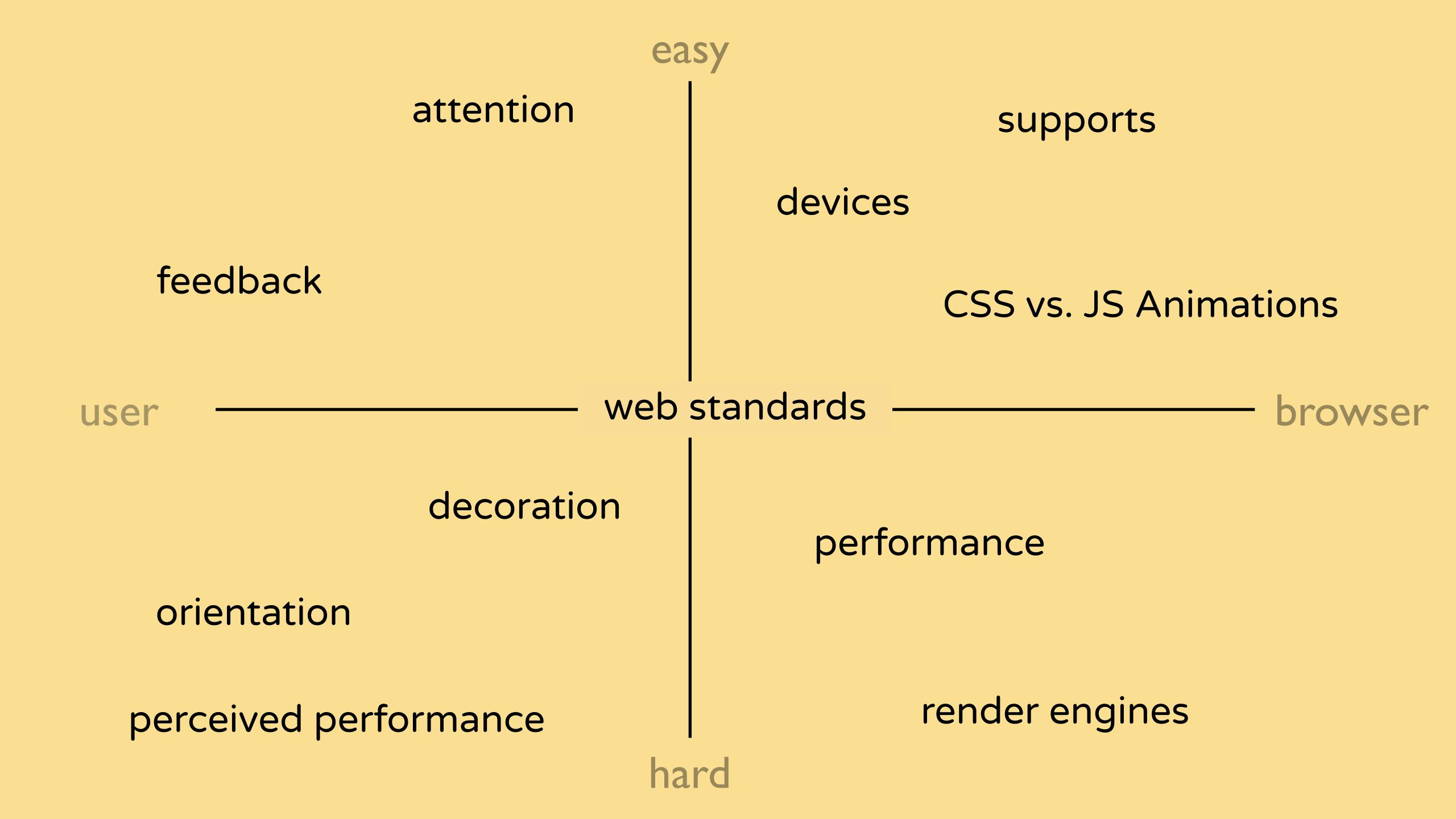


the user

the user

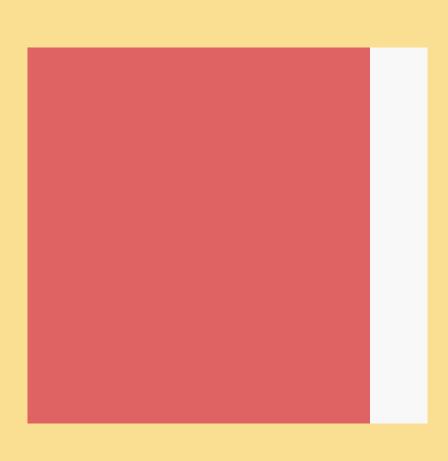




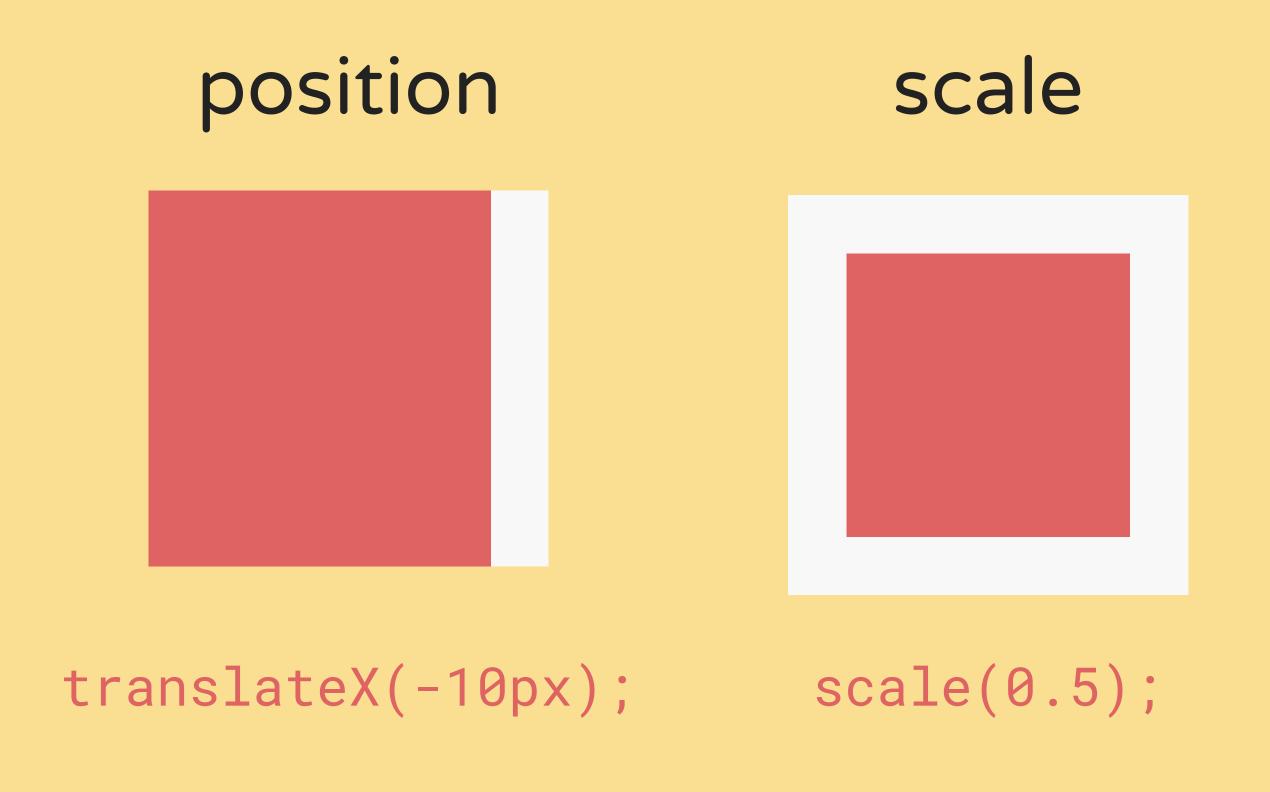


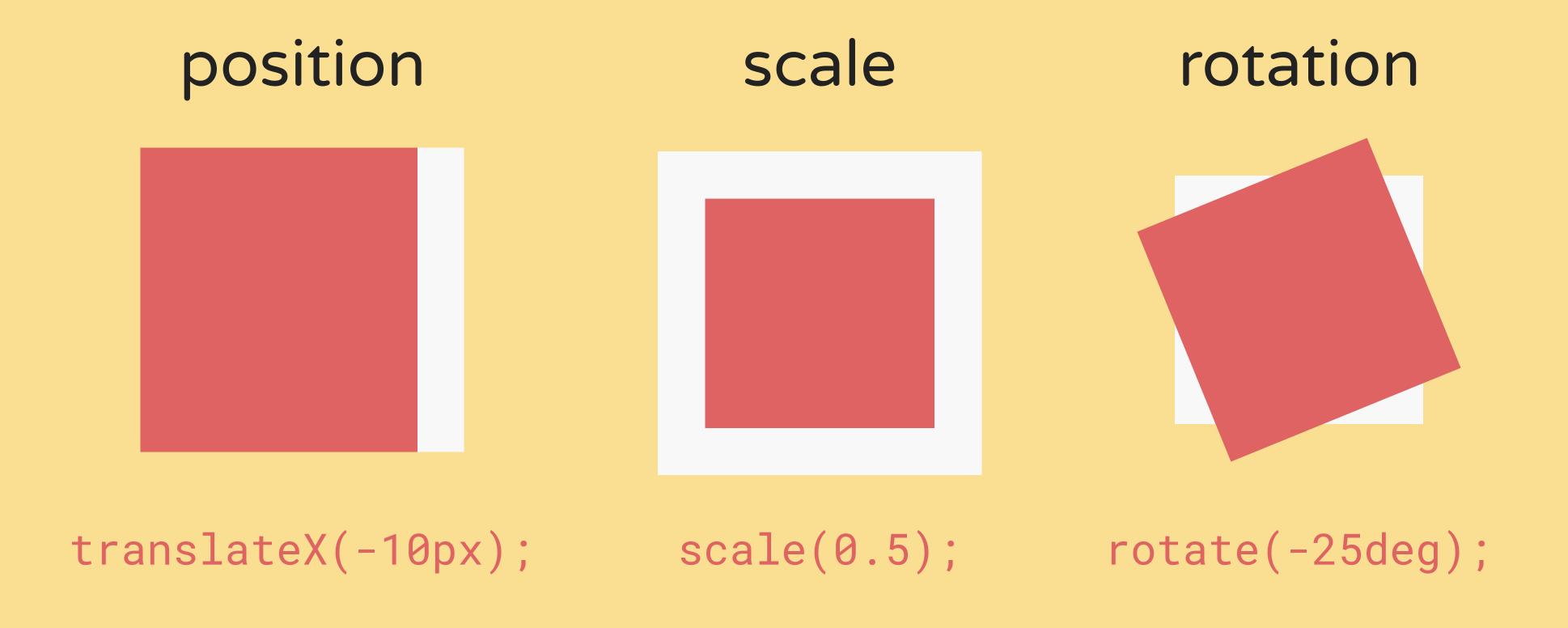
Animation with the Browser

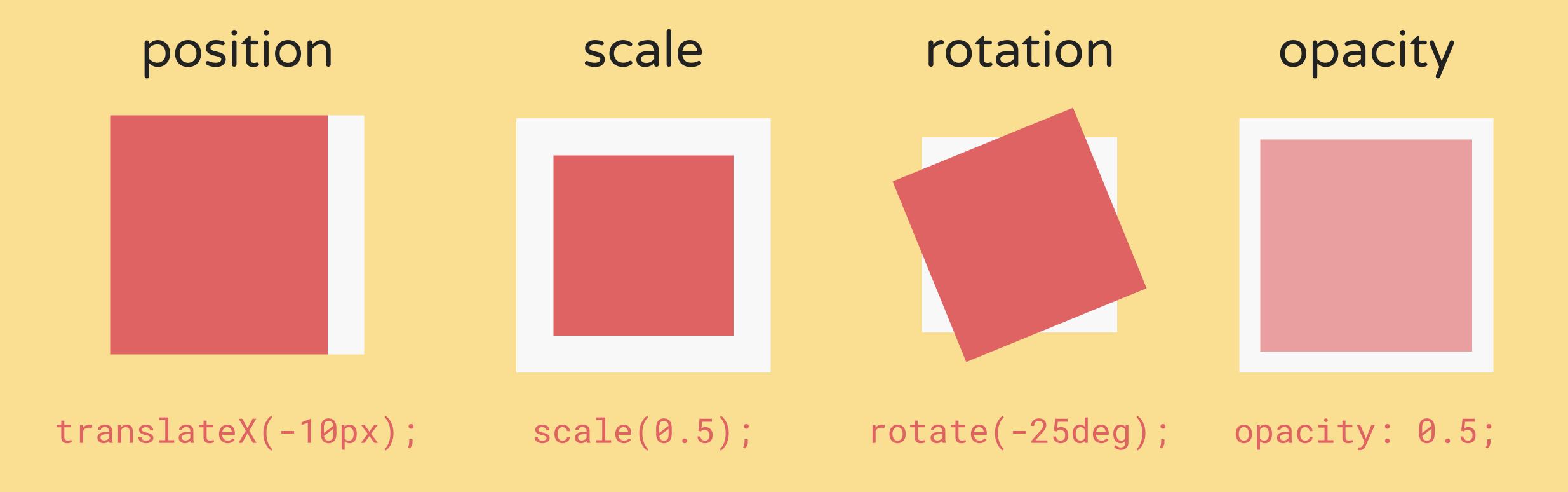
position



translateX(-10px);







order matters

```
.foo {
    transform: translateX(100px);
    transform: scale(0.5);
    transform: rotate(45deg);
    transform: skewX(15deg);
    transform: rotate(5deg) scale(1.1, 1.1) translate(-20%, 30px);
    opacity: 0.5;
}
```

transform properties can be chained

transform: translateX(66vmin) rotate(.125turn) scale(.5) translateY(66vmin)



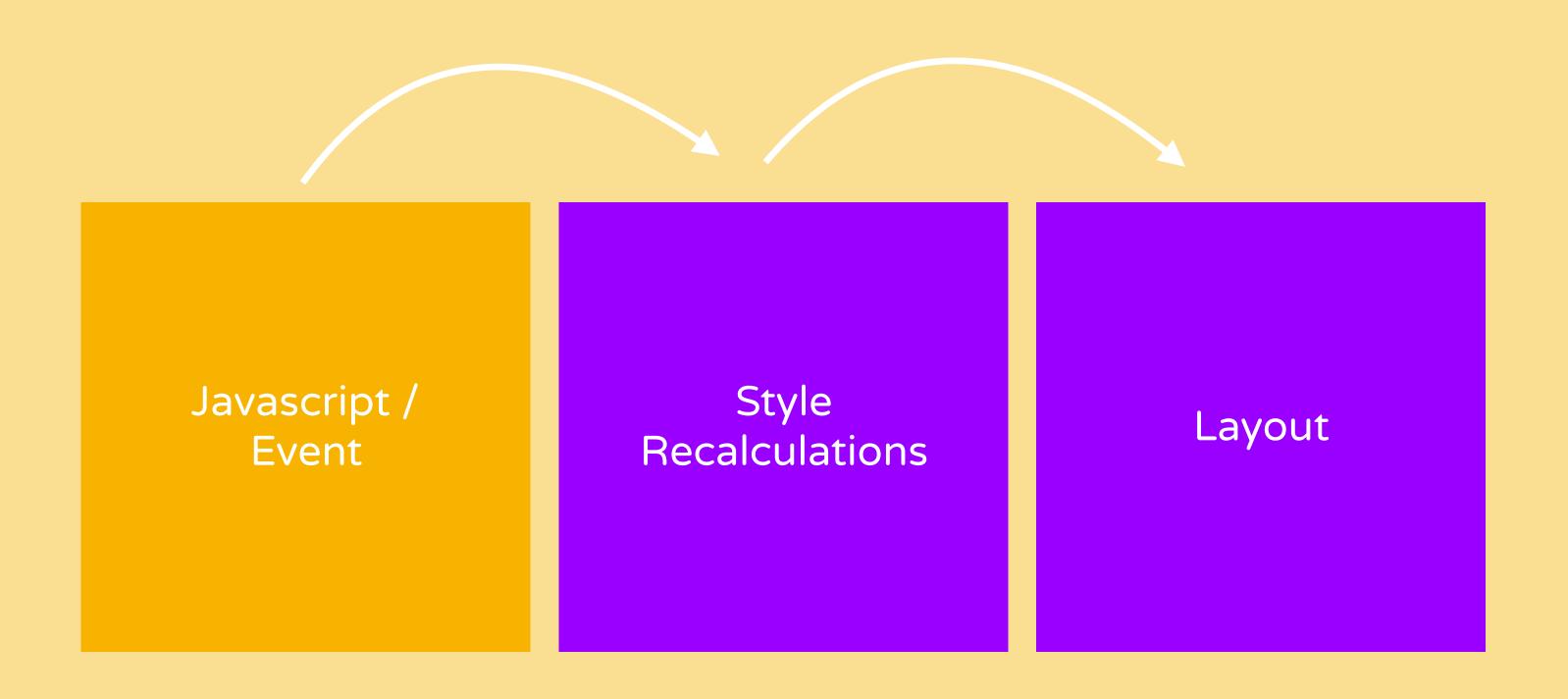
this is not performant

```
.menu {
 left: 100%;
 transition: all 0.3s ease-in-out;
.menu.open {
 left: 0%;
```

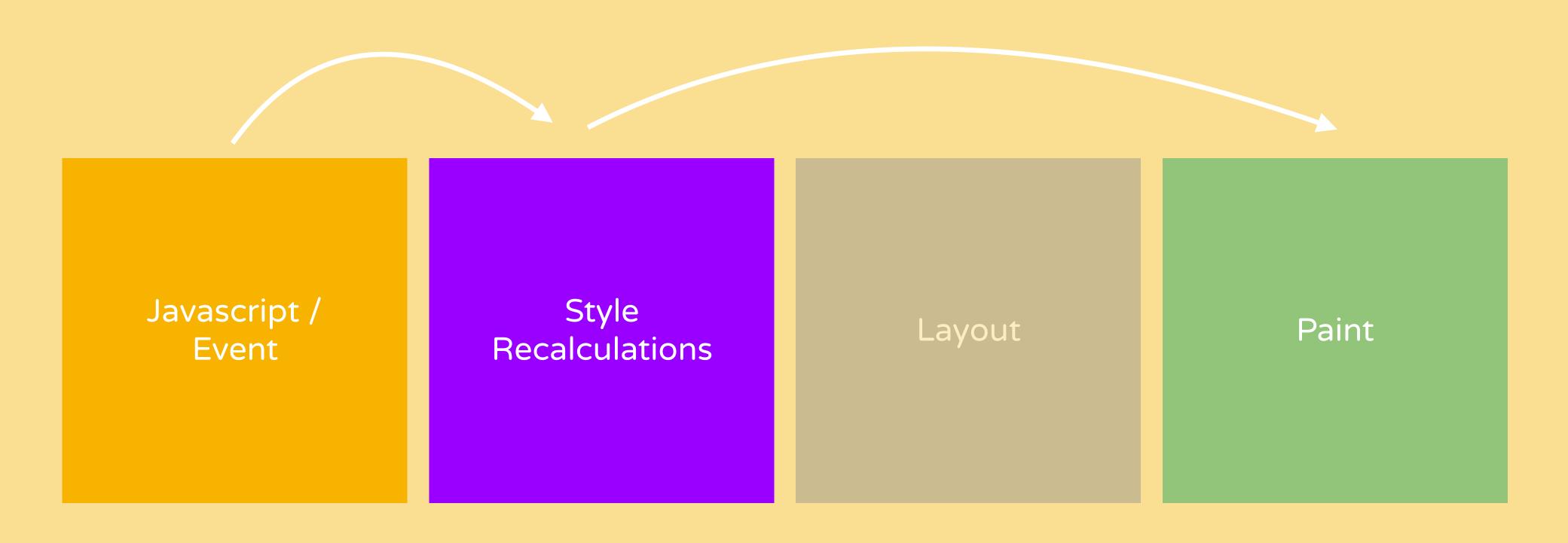
use the transform property L

```
.menu {
 transform: translateX(100%);
 transition: transform 0.3s ease-in-out;
.menu.open {
 transform: translateX(0%);
```

Javascript /
Event

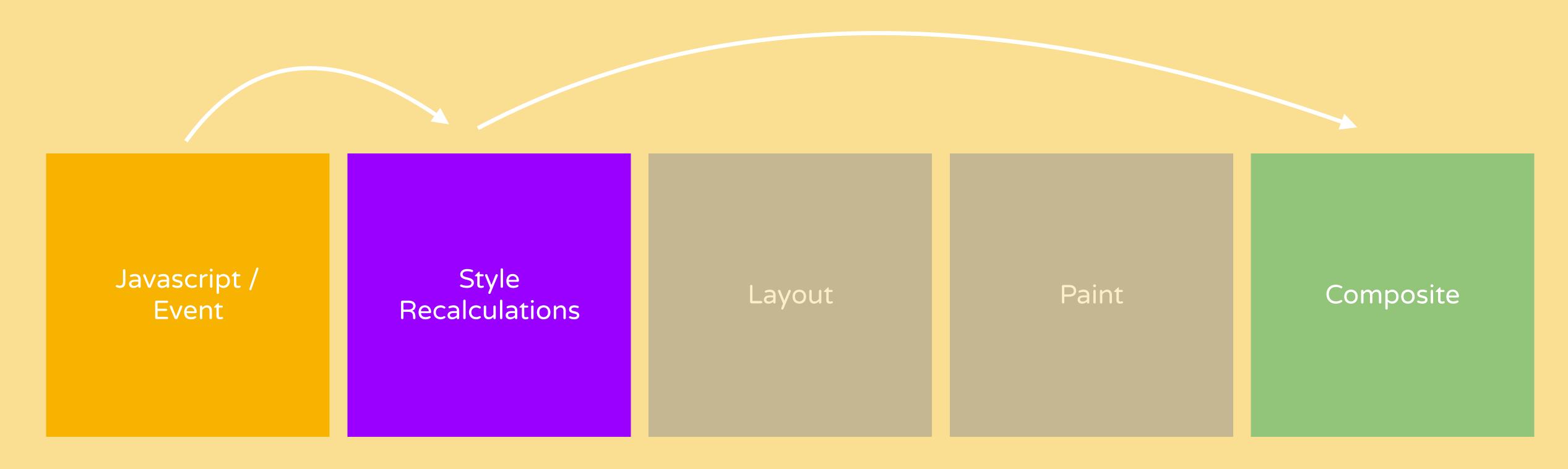


width,
margin,
left,



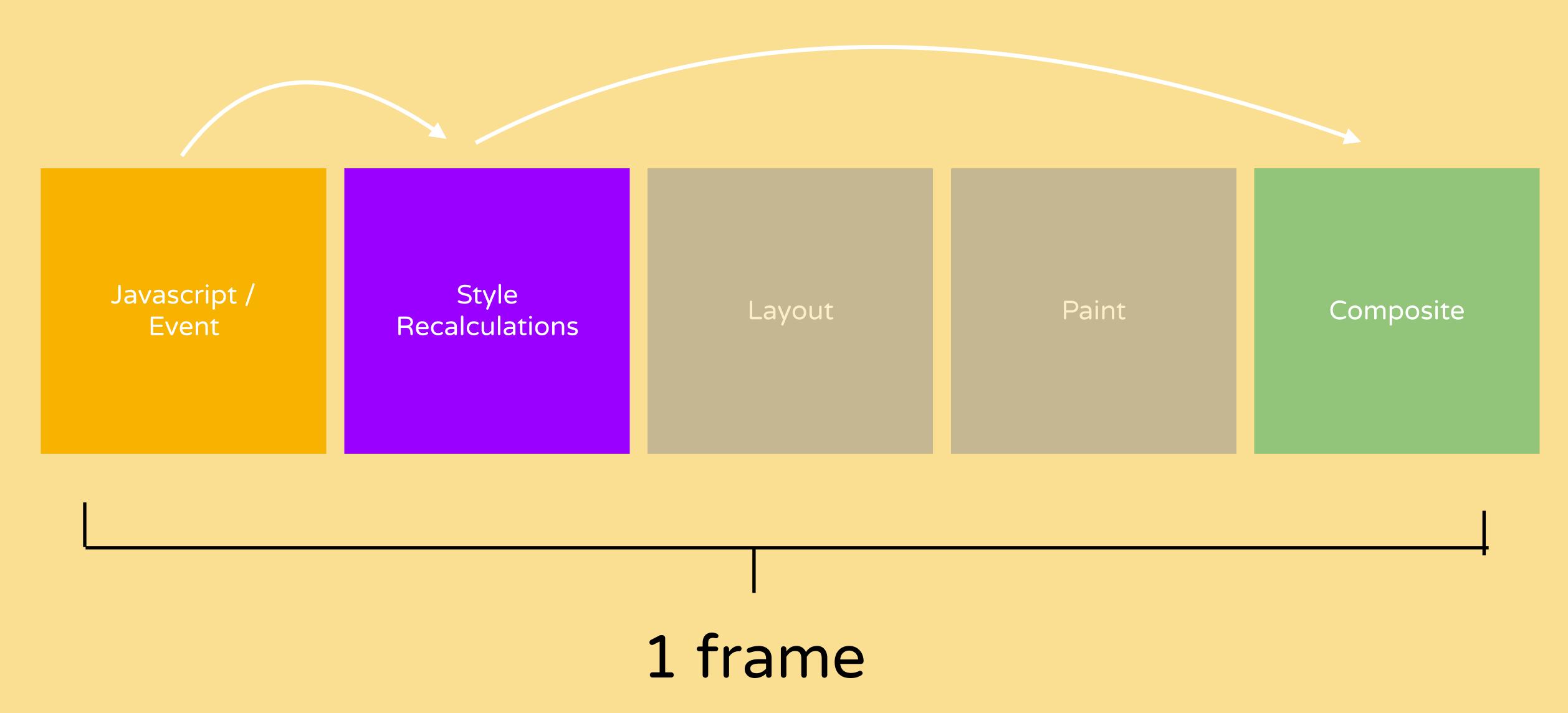
background,
box-shadow,
outline,

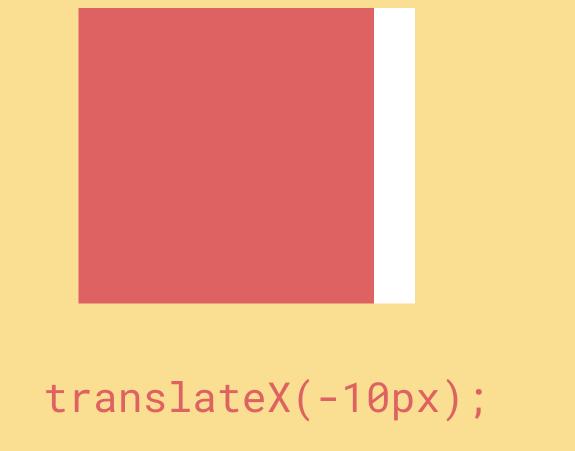
• •

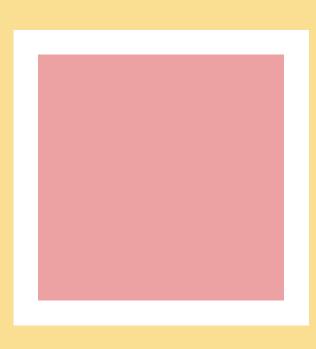


transform, opacity

• •

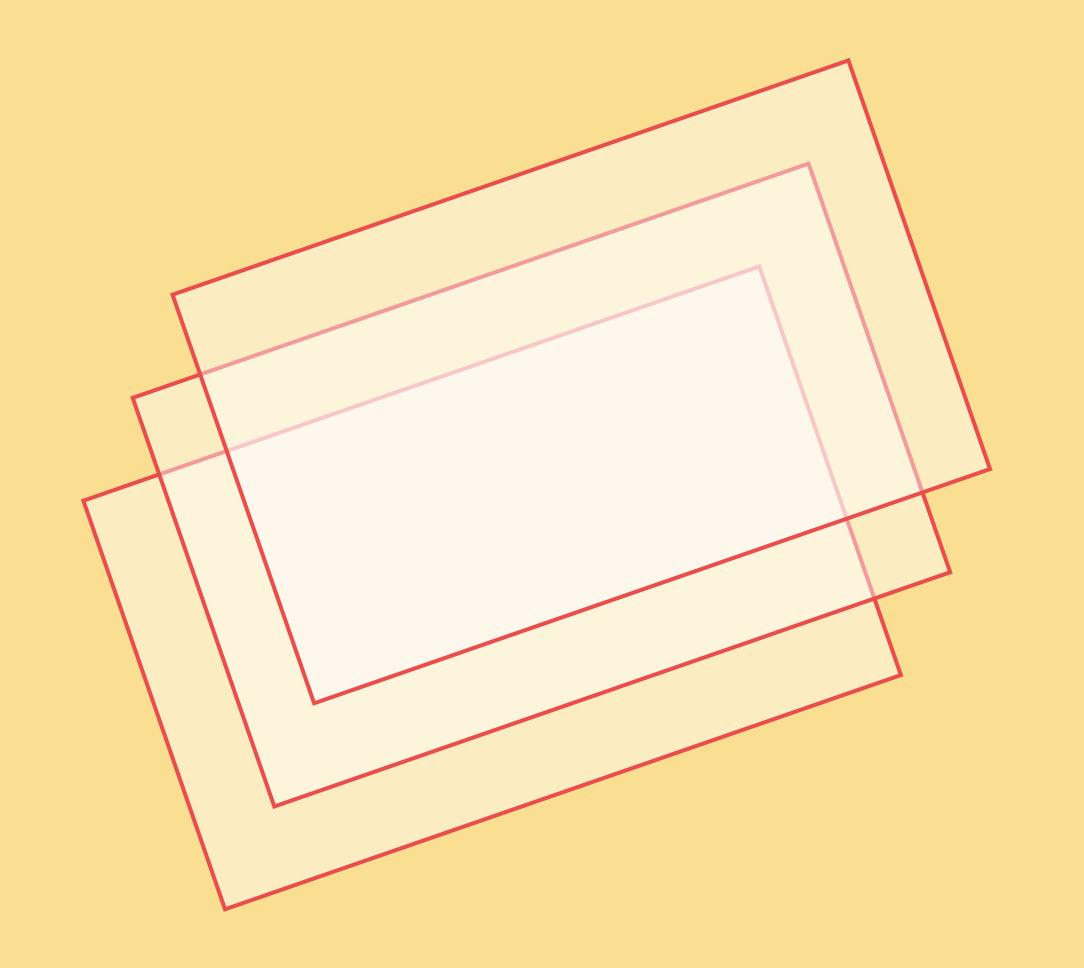






opacity: 0.5;

try to limit yourself to CSS transform
& opacity for most animations



promote elements to new layers if animated often & constantly

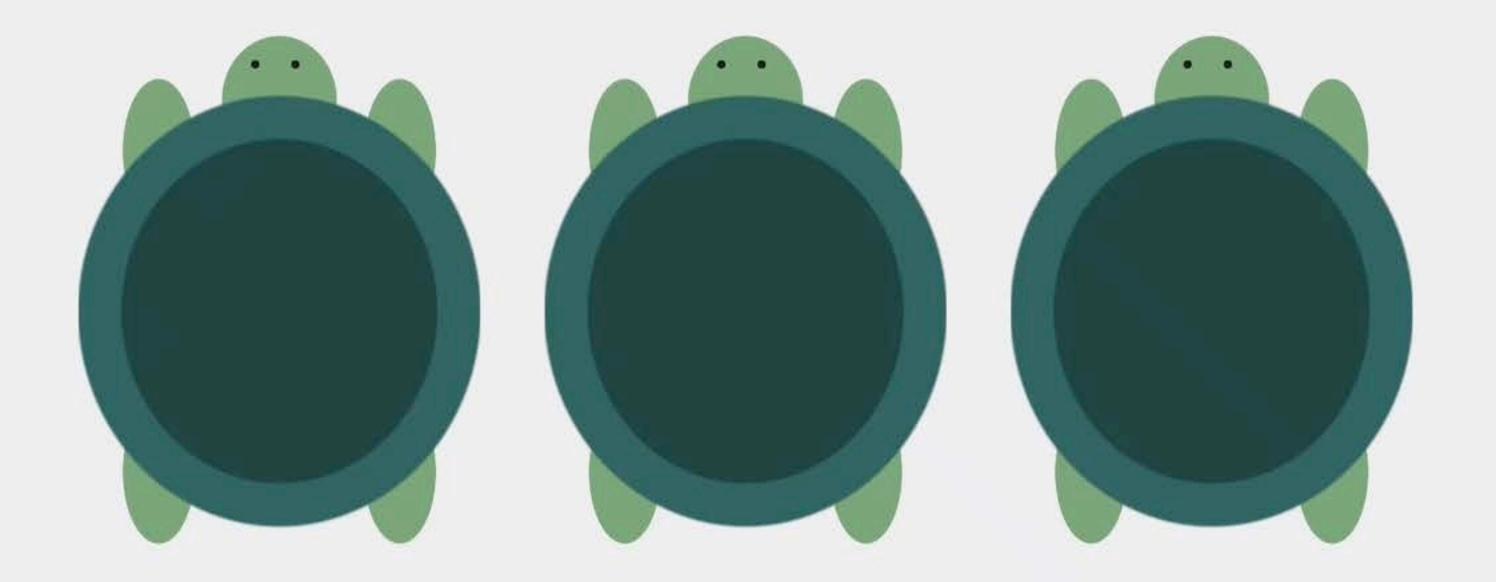
Creating new layers

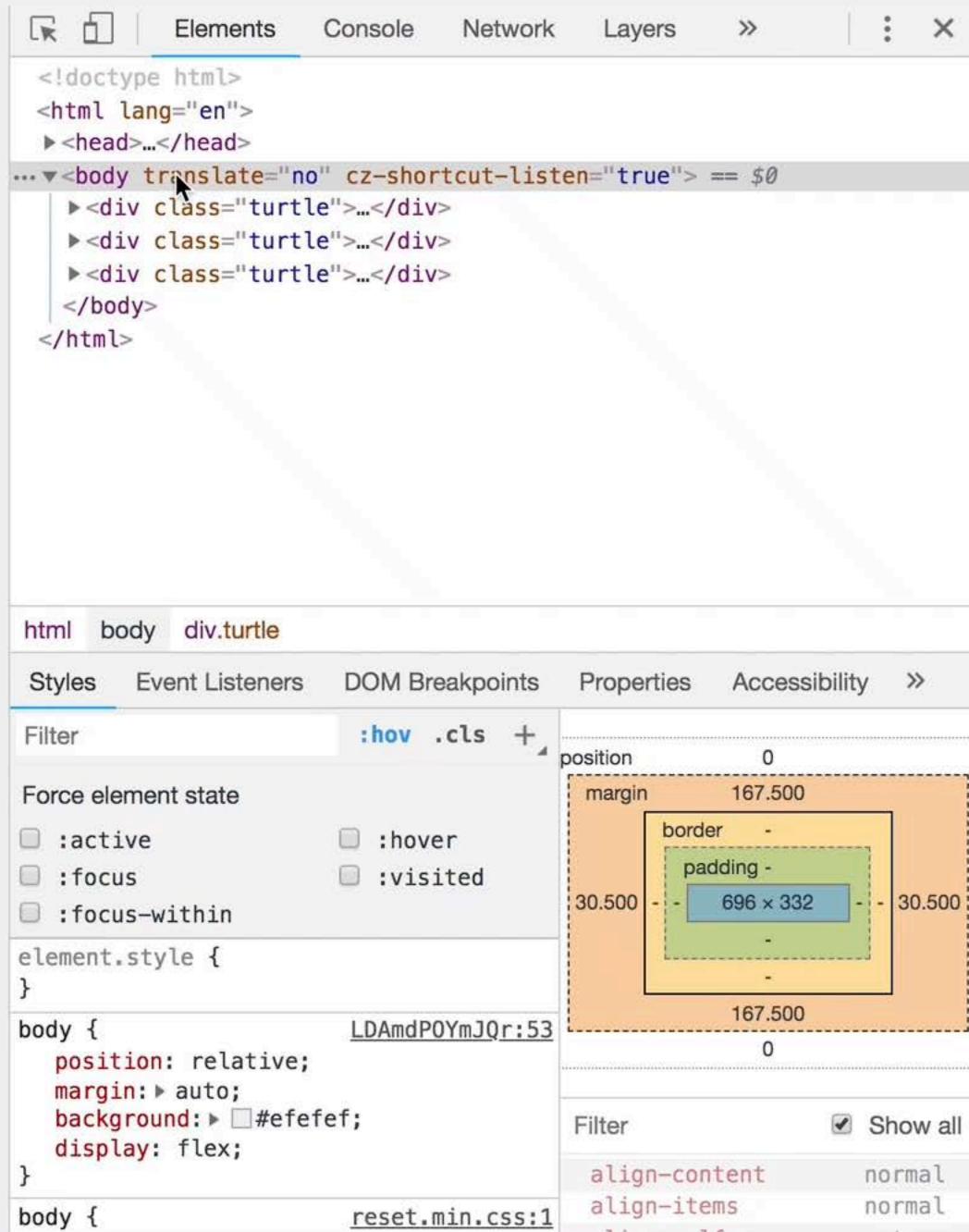
```
.foo {
  will-change: transform;
  transform: translateZ(0);
  backface-visibility: hidden;
}
```

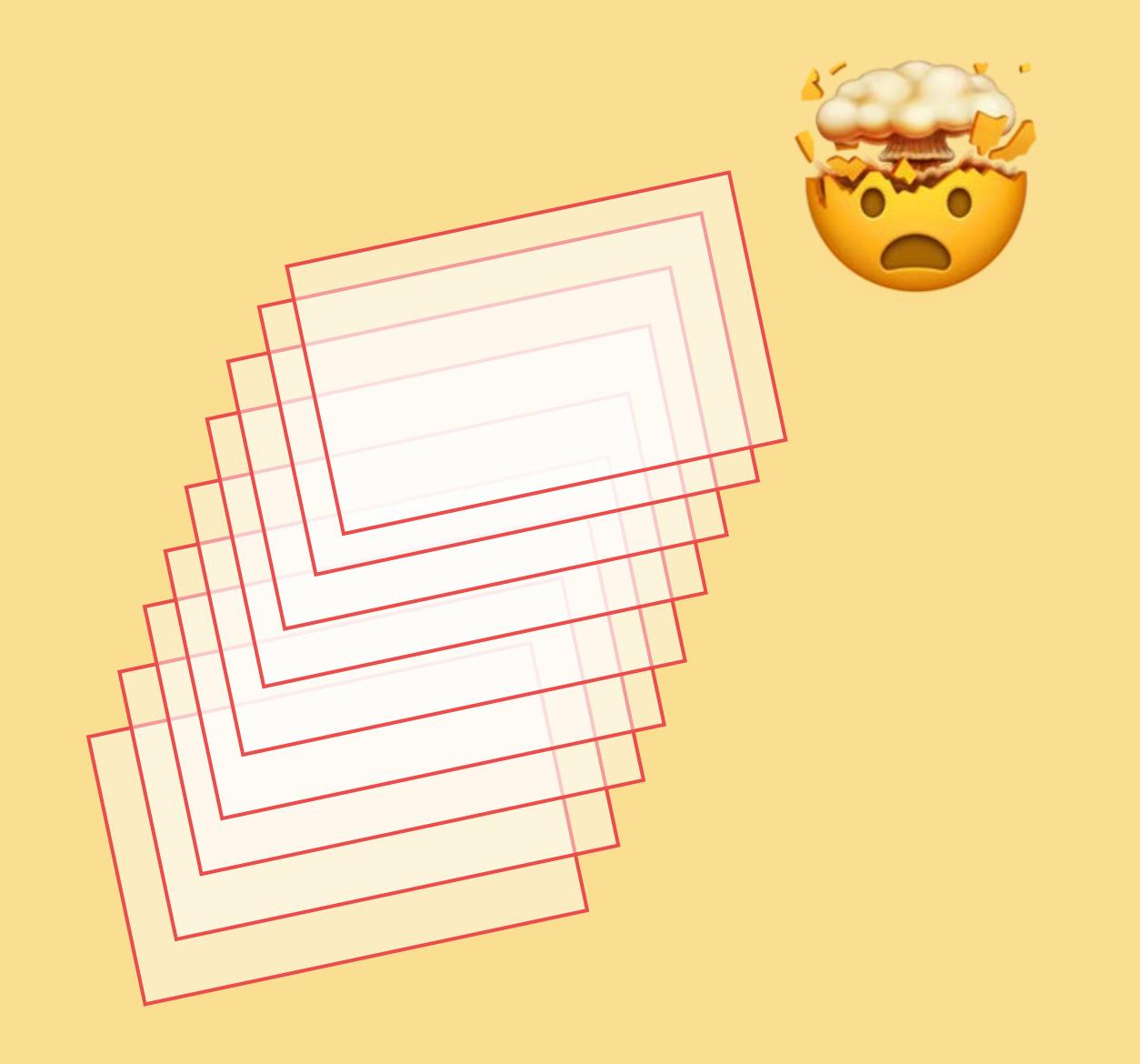
Creating new layers

```
.foo {
  will-change: transform;
  transform: translateZ(0);
  backface-visibility: hidden;
}
```

- 1. will-change property
- 2. 3D Transform
- 3. animated 2D transforms
- 4. being on top of a compositing layers
- 5. animated CSS filters



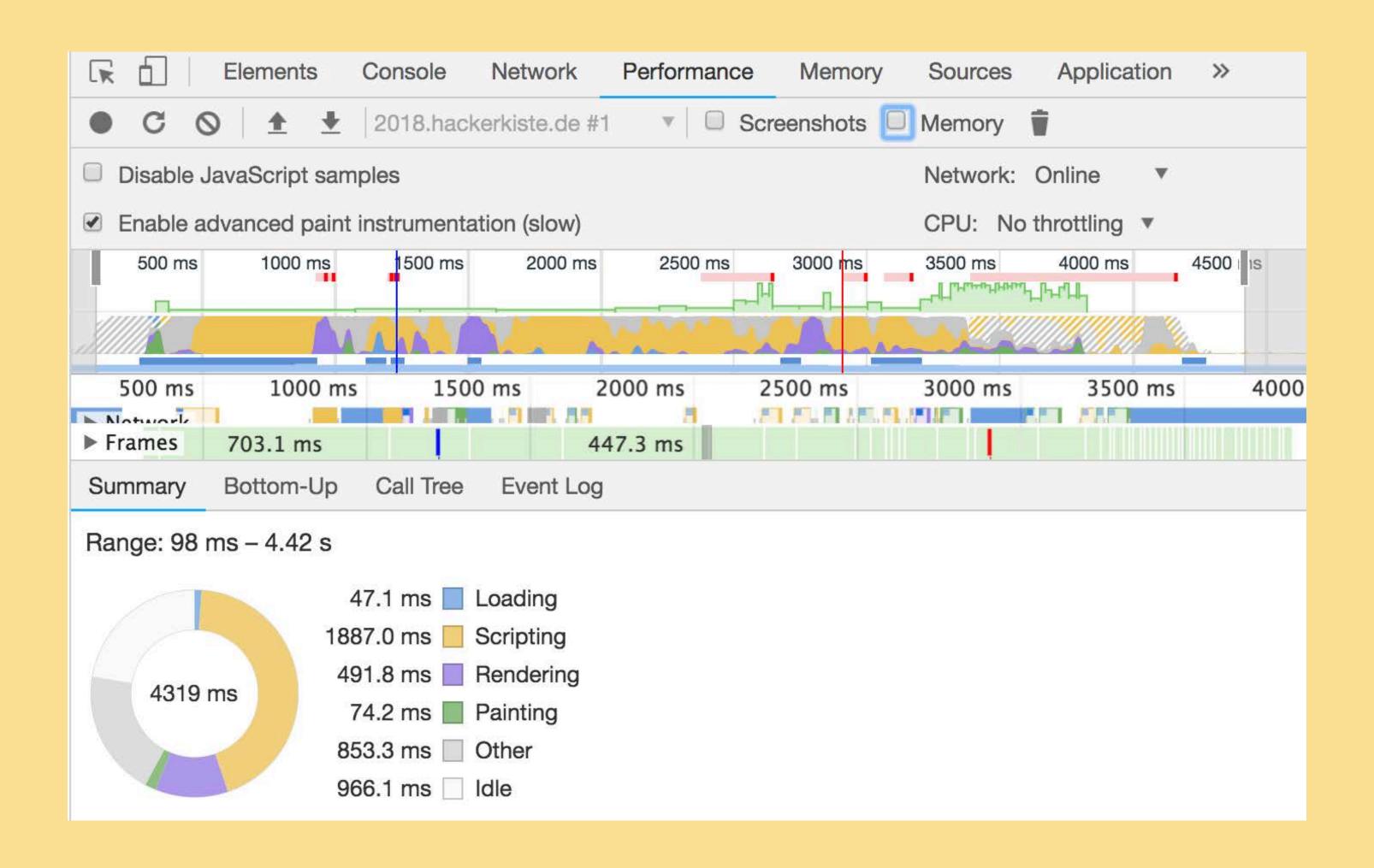


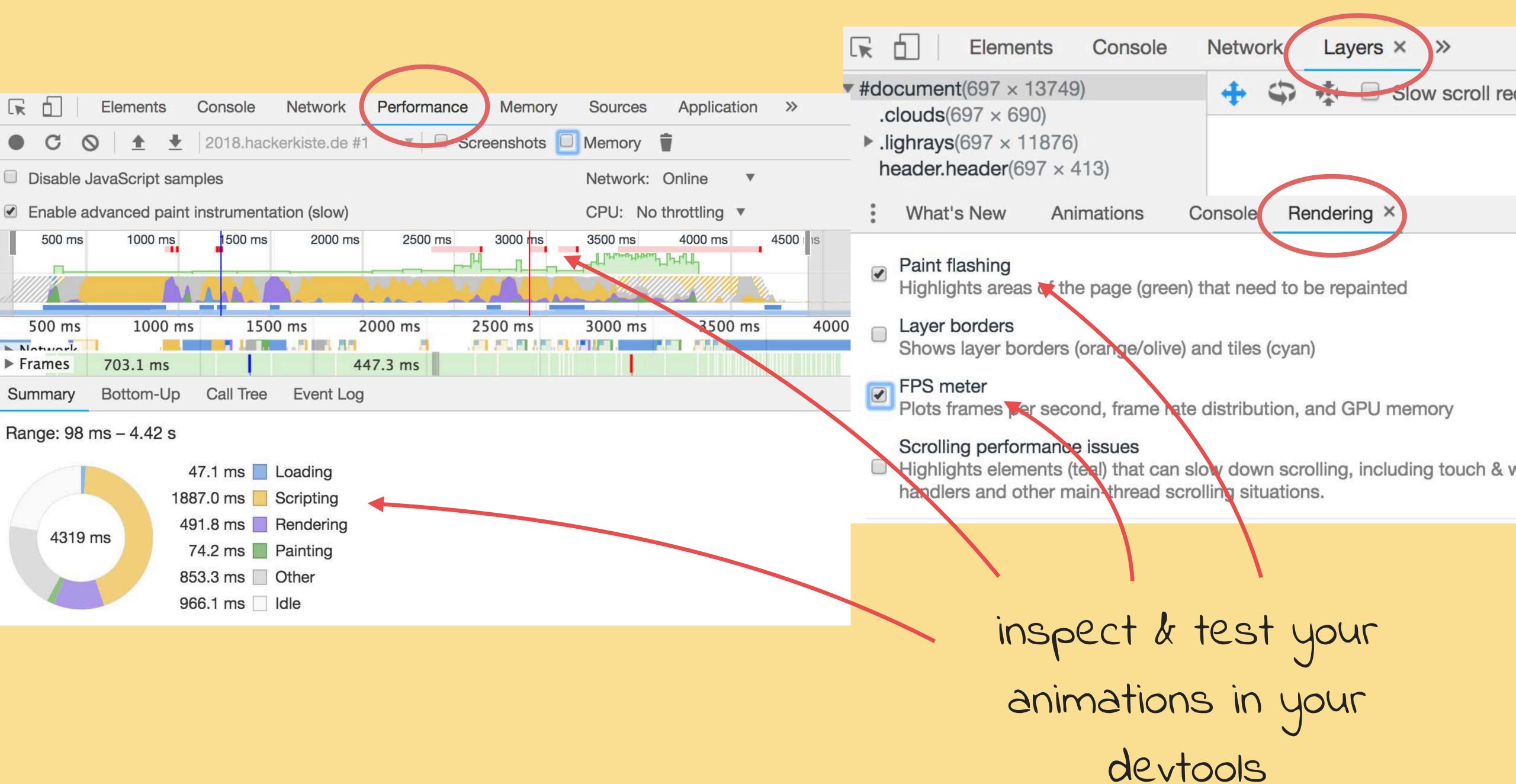


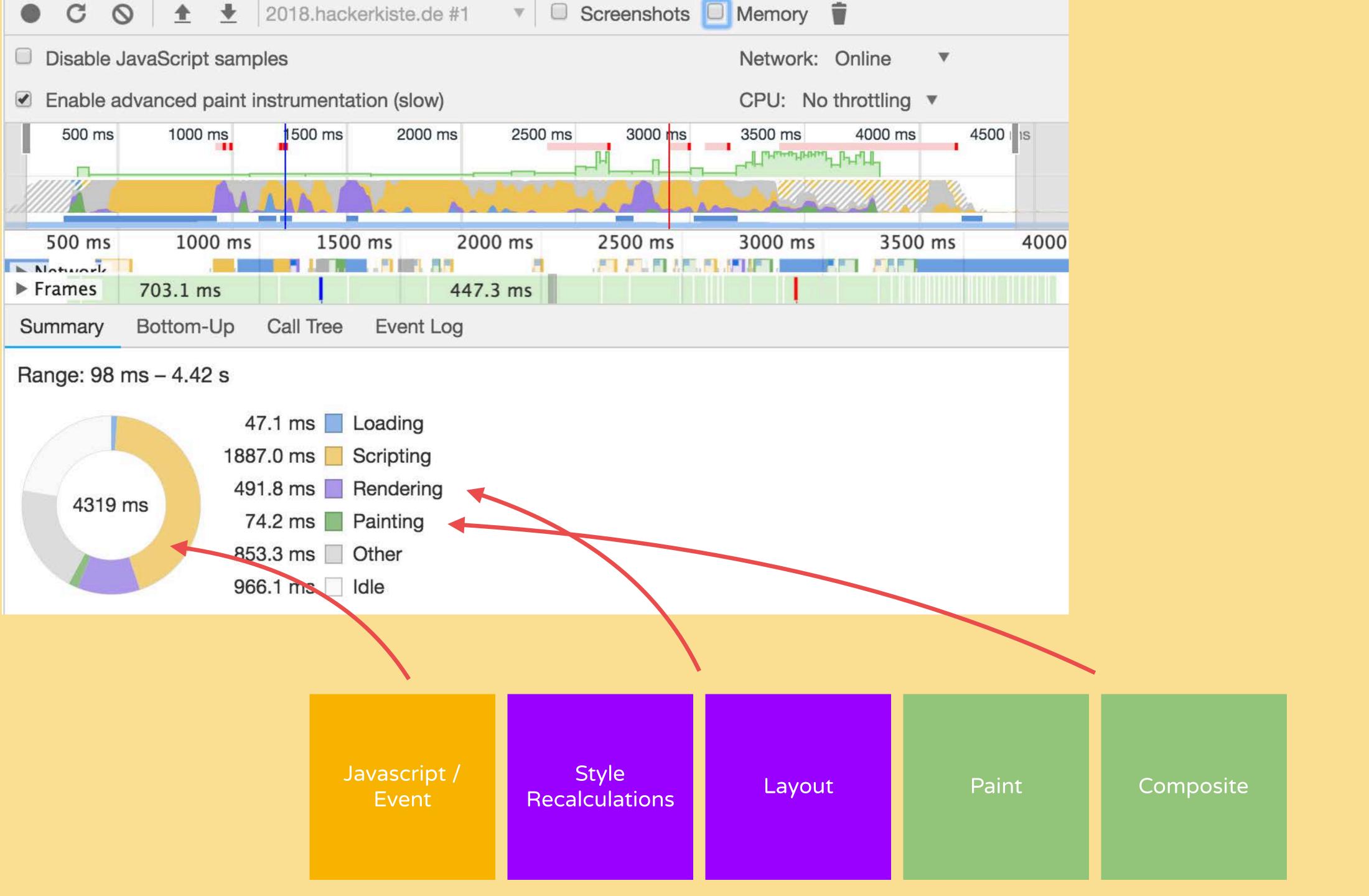
don't create too many layers

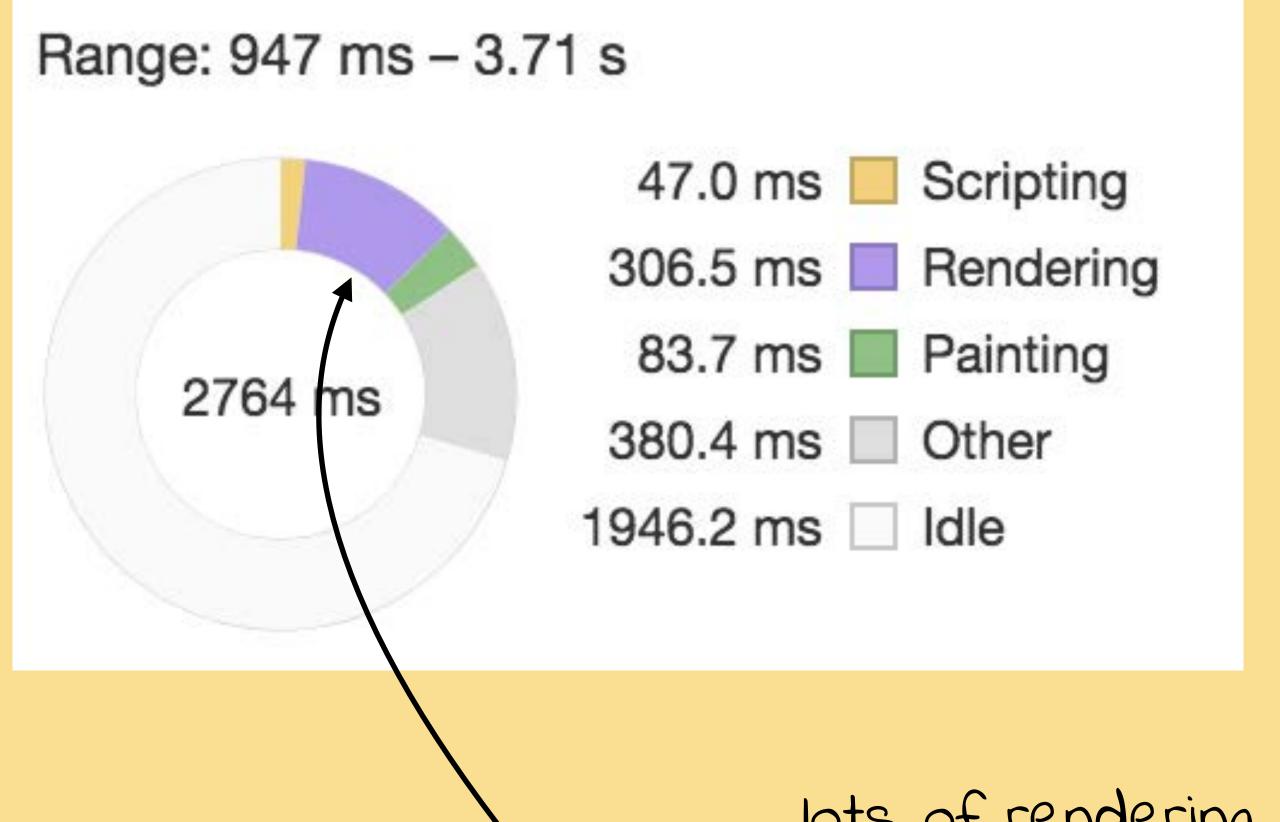
remove them if the animation is finished

DevTools









lots of rendering, maybe we animated something that triggers Layout

Animation for the User

ORIENTATION & TRANSITIONS



ORIENTATION & TRANSITIONS

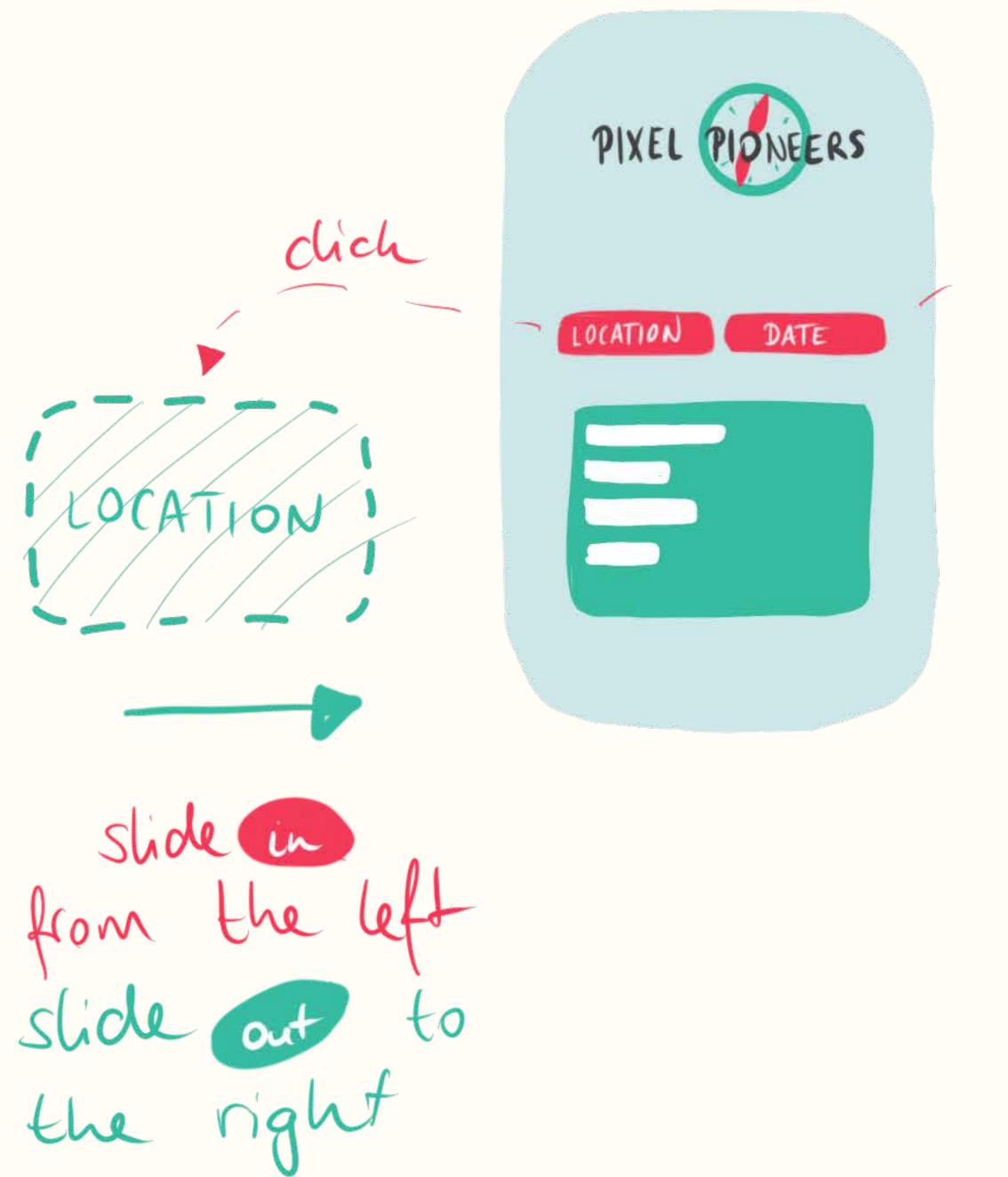
- help with reorientation in the user
- explain relationships between different information spaces

ORIENTATION & TRANSITIONS

Example: Vuels



codepen.io/lisilinhart/full/pmYBva



clich stide in from the right stide out to the left

```
new Vue({
    el: "#app",
    data: {
       view: "location"
    },
});
```

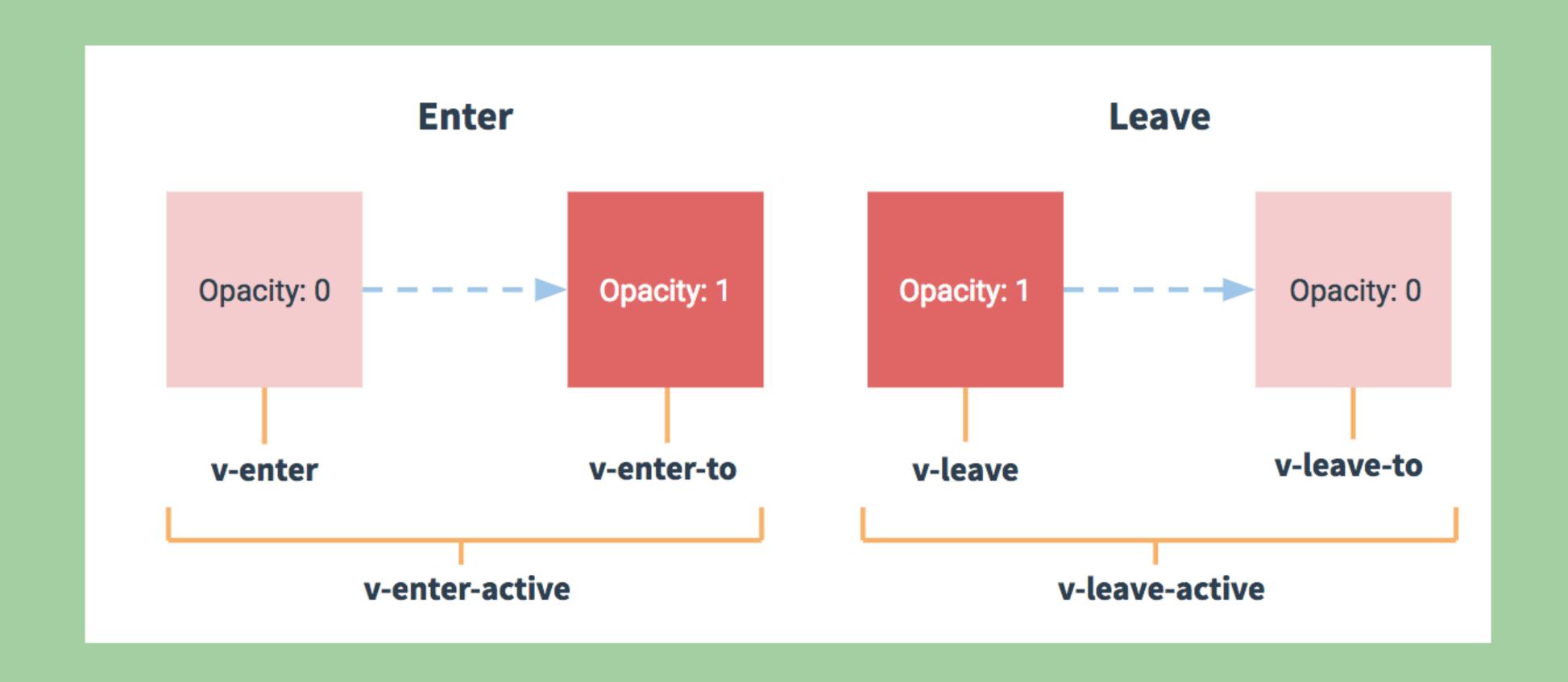
view is toggled when a button is clicked

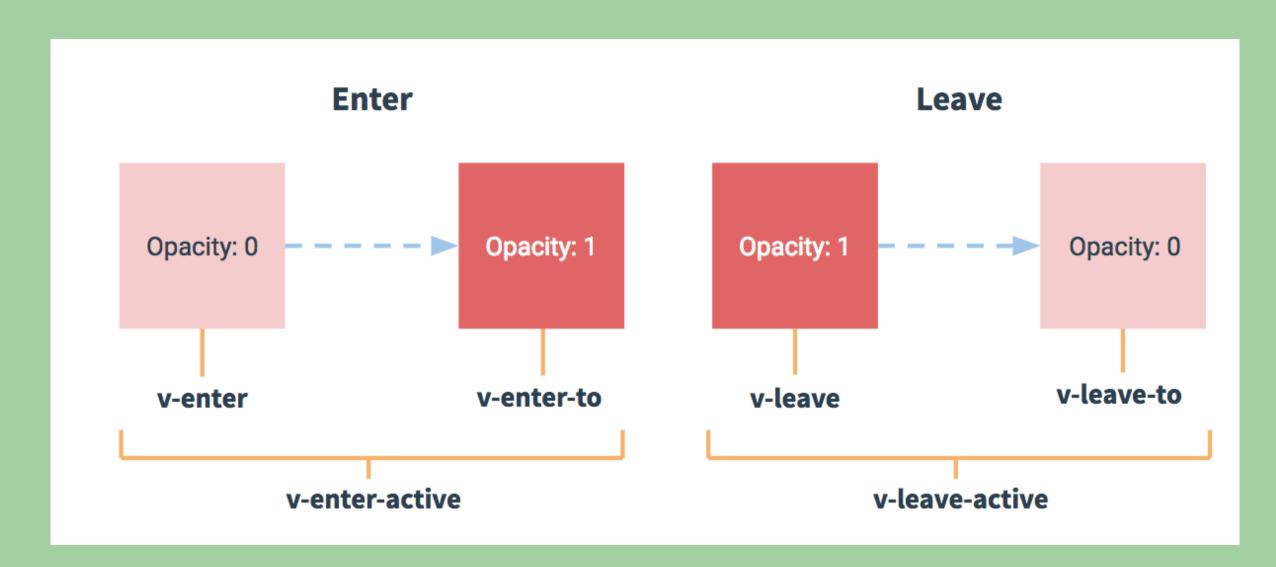
```
new Vue({
    el: "#app",
    data: {
       view: "location"
    },
});
```

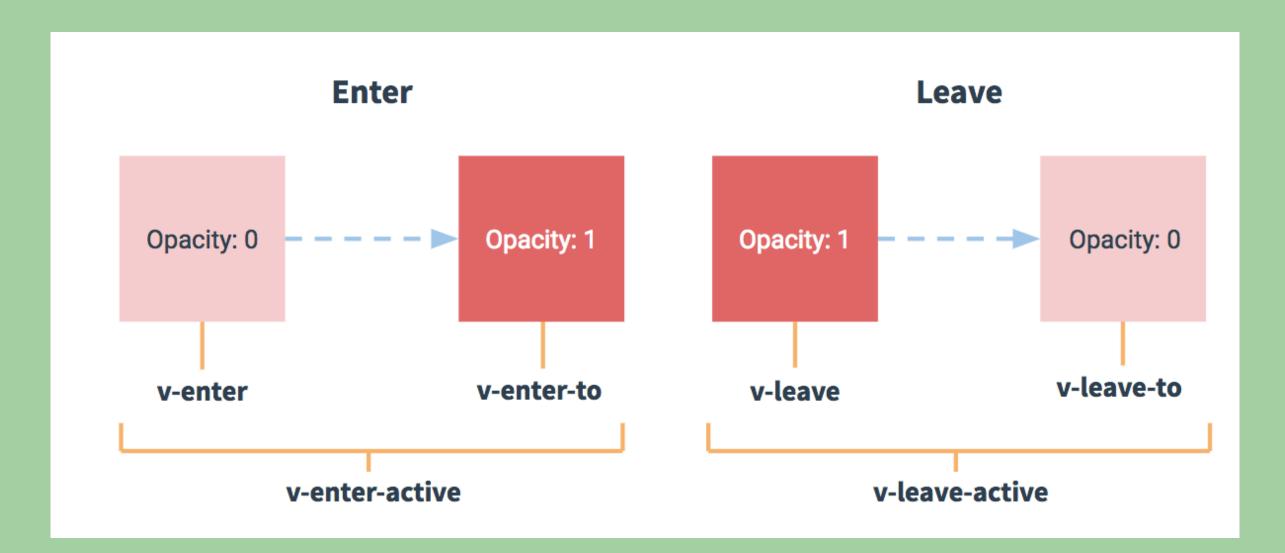
view is toggled when a button is clicked

```
<transition :name="transition" mode="out-in">
   <main v-if="view == 'index'" key="index">
     A one-day conference of practical and inspiring design
     and front-end talks, featuring eight world-class speakers,
      preceded by a workshop day.
   </main>
   <main v-if="view == 'location'" key="location">M Shed, Bristol, UK</main>
   <main v-if="view == 'date'" key="date">7th June</main>
</transition>
```

component is transitioned in if view is the correct value

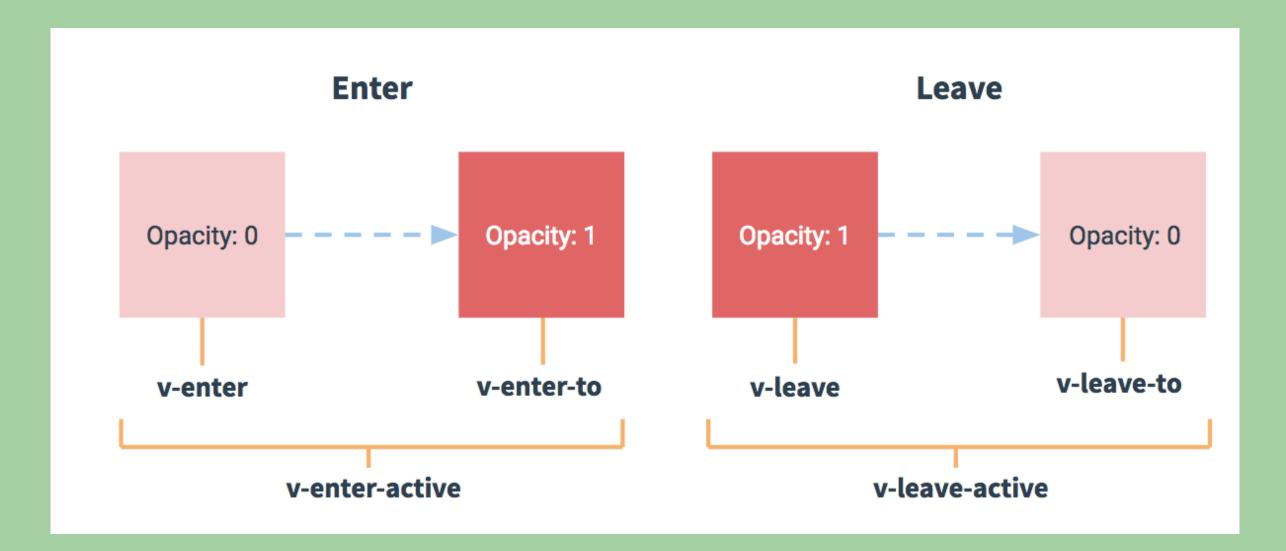






```
.main {
    opacity: 1;
    transition: transform .4s, opacity .4s;
    transition-timing-function: var(--circ-out);
}
```

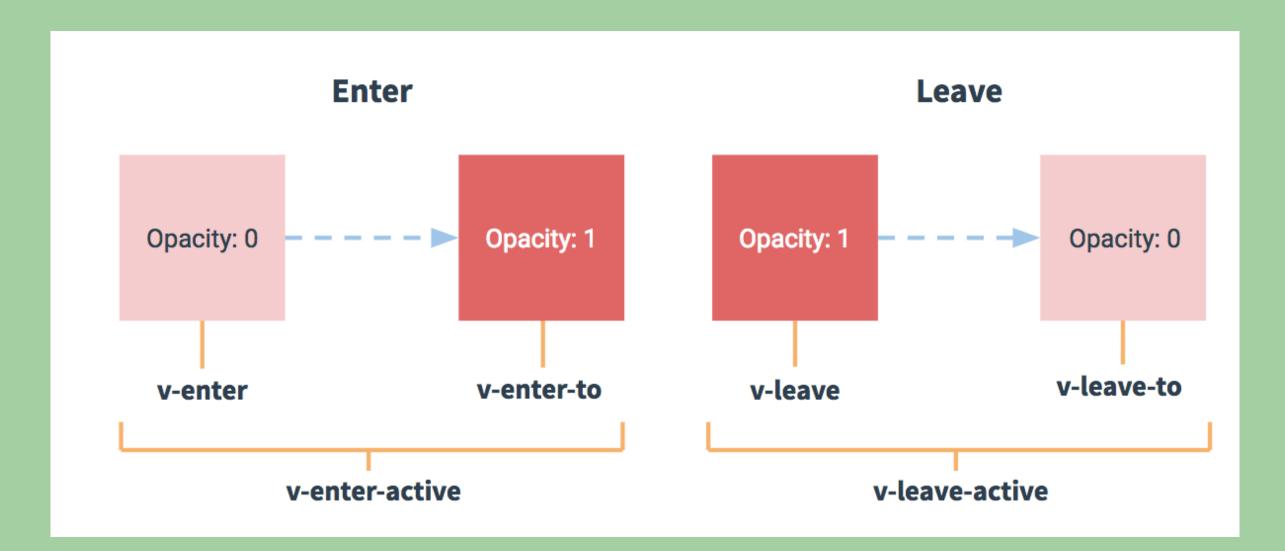
define default state and the properties you want to transition



```
.main {
    opacity: 1;
    transition: transform .4s, opacity .4s;
    transition-timing-function: var(--circ-out);
}

.slide-left-enter {
    opacity: 0;
    transform: translateX(-50px);
    transition-timing-function: var(--c.rc-out);
}
```

define how the element should animate when it'



define how the element should animate when it's leaving

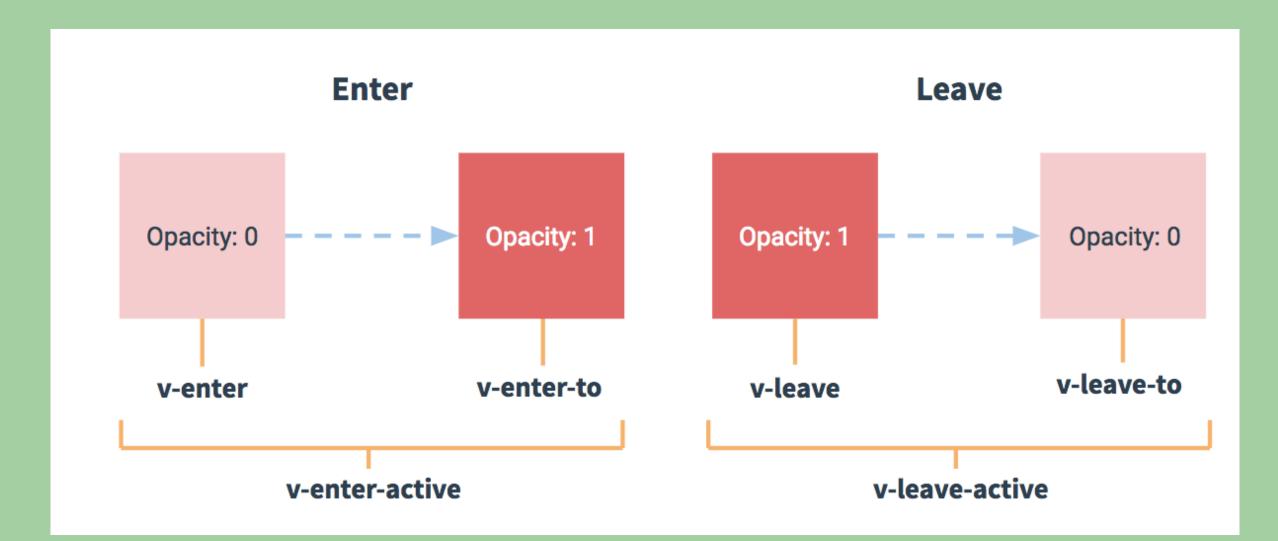
```
.main {
    opacity: 1;
    transition: transform .4s, opacity .4s;
    transition-timing-function: var(--circ-out);
}

.slide-left-enter {
    opacity: 0;
    transform: translateX(-50px);
    transition-timing-function: var(--circ-out);
}
```

```
.slide-left-leave-to {
  opacity: 0;
  transform: translateX(50px);
  transition-timing-function: var(--circ-in);
}
```

opacity: 0;

transform: translateX(50px);



define different timing functions for entering and leaving

```
.main {
   opacity: 1;
   transition: transform .4s, opacity .4s;
   transition-timing-function: var(--circ-out);
.slide-left-enter {
  opacity: 0;
  transform: translateX(-50px);
transition-timing-function: var(--circ-out);
 slide-left-leave-to {
```

transition-timing-function: var(--circ-in);

defines dynamic transitions depending on the state



codepen.io/lisilinhart/full/pmYBva

"Human perception of time is fluid, and can be manipulated in purposeful and productive ways."

Chris Harrison

Faster Progress Bars: Manipulating Perceived Duration with Visual Augmentations

Chris Harrison, Zhiquan Yeo, Scott E. Hudson

Faster Progress Bars: Manipulating Perceived Duration with Visual Augmentations

animated progress bars reduced the perceived duration among the participants by 11%

0. seconds feels instantaneous

second user will feel the pause

5-10 seconds struggle to maintain the user's attention

The page is loading

activity & progress indication

The page is loading

Here is a skeleton screen of what content is being loaded

activity & progress indication

Lazy loading & hinting

The page is loading

Here is a skeleton screen of what content is being loaded

While you're waiting, here is some more information on how the process works

activity & progress indication

Lazy loading & hinting

using idle times to inform the user

Example:

web Animations API

WEBANIMATIONS API

```
const el = document.querySelector('.foo');
el.animate(<keyframes>, <timing object>);
```

keyframes object or array: stages of the animation

timings object:
how it's animated

(e.g. speed, delay, ...)

WAAPI KEYFRAMES

keyframes object keyframes array

```
{ // from
  opacity: 0,
  color: "#fff"
{ // to
  opacity: 1,
  color: "#000"
```

WAAPI TIMINGS OBJECT

```
duration: 2000, // milliseconds
  iterations: 1, // or Infinity
  direction: 'normal', // 'alternate', 'reverse', 'alternate-reverse'
  fill: 'forwards', // 'backwards', 'both', 'none', 'auto'
  delay: 0, // milliseconds
  endDelay: 0, // milliseconds
  easing: 'linear', // 'ease', 'ease-in-out', 'ease-in', ...
}
```

how fast should my animation run, how often should it run, what should happen after it's finished?

CSS VS WAAPI

```
.el {
   animation: stretch 500ms Os 2 alternate ease-in-out;
}

@keyframes stretch {
   80% {
     transform: scaleX(.9);
   }
   100% {
     transform: scaleX(1.2);
   }
}
```

CSS Animation

CSS VS WAAPI

```
.el {
   animation: stretch 500ms 0s 2 alternate ease-in-out;
}

@keyframes stretch {
   80% {
     transform: scaleX(.9);
   }
   100% {
     transform: scaleX(1.2);
   }
}
```

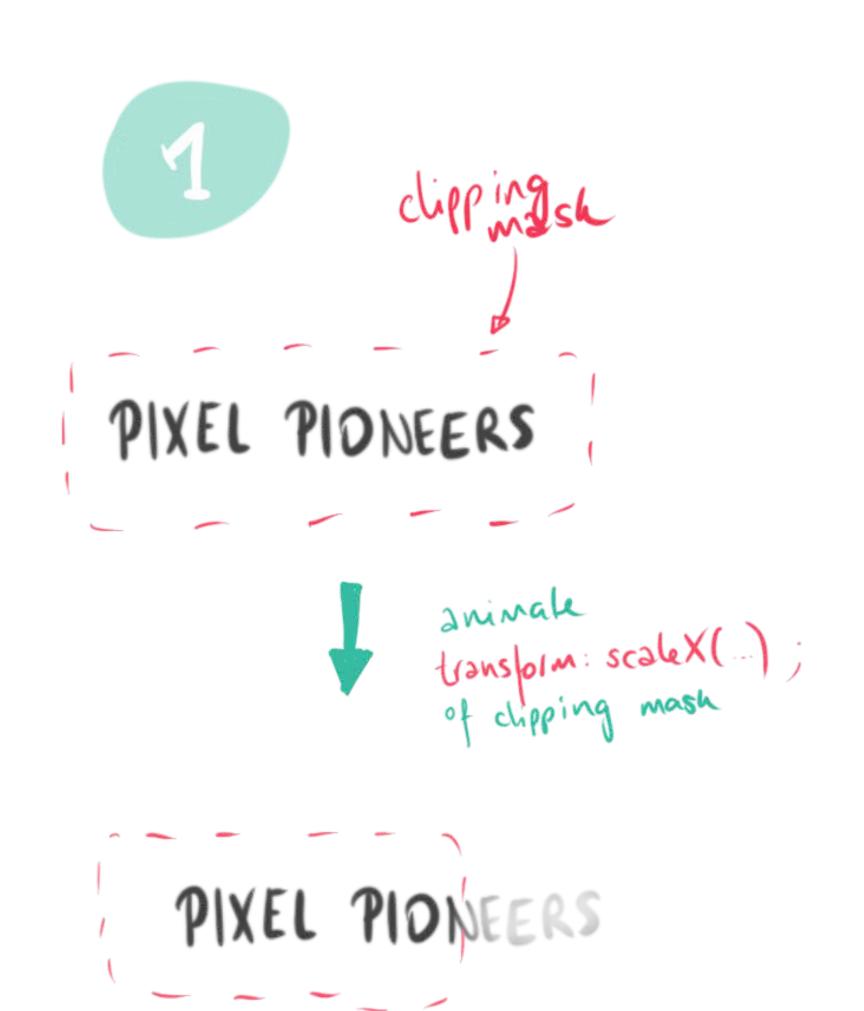
```
const animation = el.animate([
    { transform: 'scaleX(1)', offset: 0 },
    { transform: 'scaleX(.9)', offset: .8 },
    { transform: 'scaleX(1.2)', offset: 1 }
    ], {
      duration: 500,
      iterations: 2,
      direction: 'alternate',
      easing: 'ease-in-out'
    });
```

CSS Animation

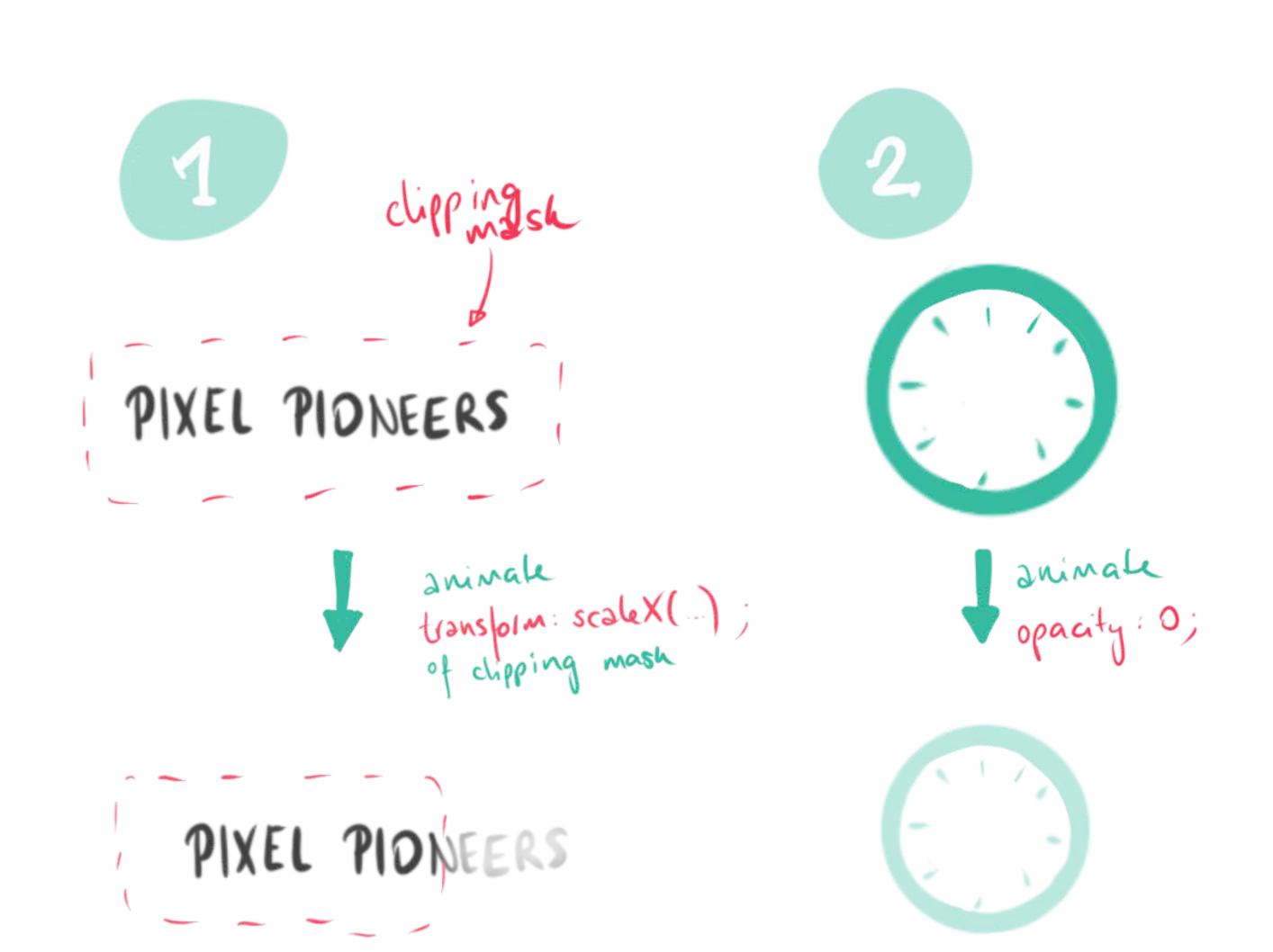
Waapi Animation



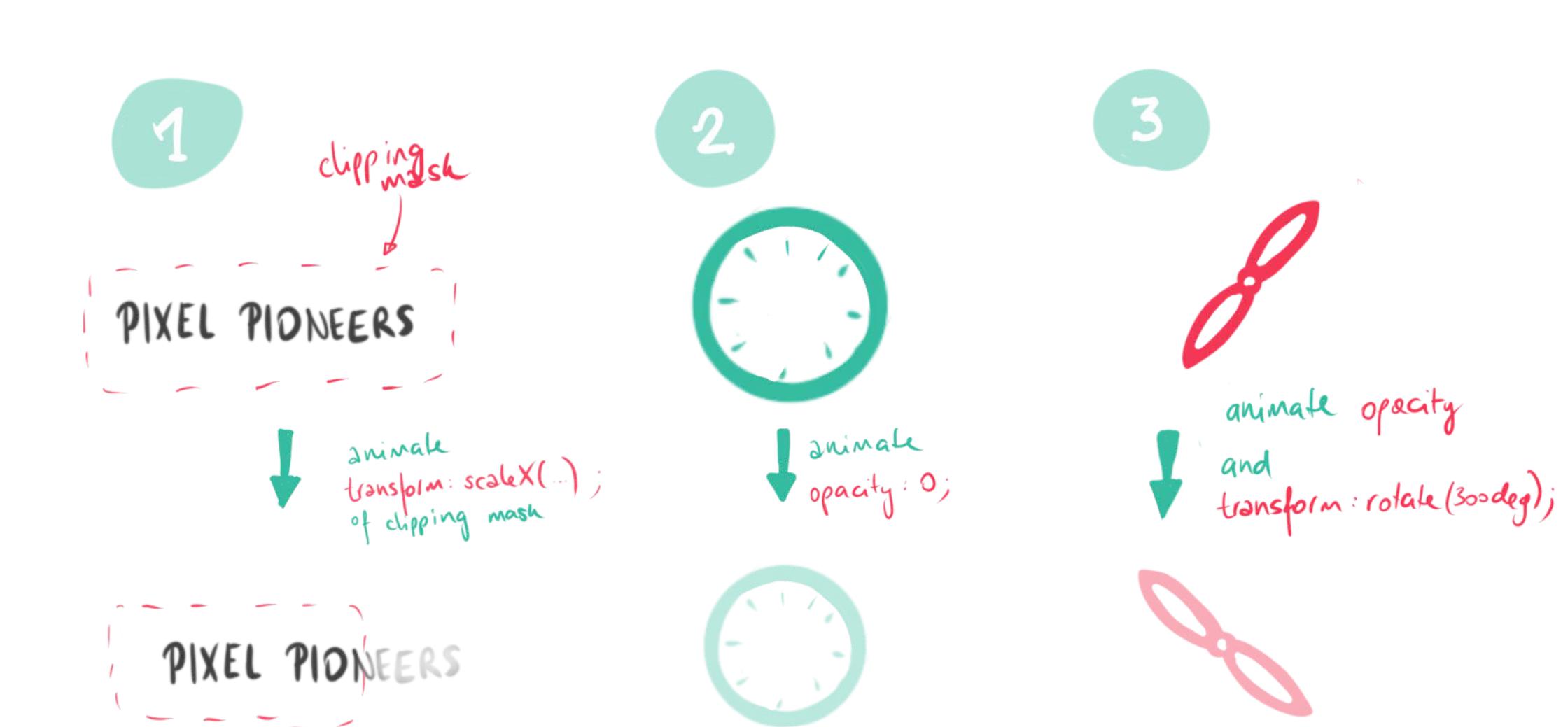












WAAPI EXAMPLE

```
<svg viewBox="0 0 946.99 370.43" class="logo">
 <title>Pixel Pioneers</title>
                                                              width & height of view box
 <defs>
   <clipPath id="clip">
     <rect id="clip-rect" x="0" y="0" width="946.99" height="370.43"></rect>
   </clipPath>
 </defs>
 <g class="logo__compass" transform="translate(532.821, 211.1785)">
 </g>
 <g class="logo__text" clip-path="url(#clip)" fill="#231f20">
                                                                    clip-path for text in
 </g>
</svg>
                                                                     order to animate
                                                                   transform: scale X(0)
```

WAAPI EXAMPLE

```
<svg viewBox="0 0 946.99 370.43" class="logo">
 <title>Pixel Pioneers</title>
 <defs>
   <clipPath id="clip">
      <rect id="clip-rect" x="0" y="0" width="946.99" height="370.43"></rect>
   </clipPath>
 </defs>
 <g class="logo__compass" transform="translate(532.821, 211.1785)">
 </g>
 <g class="logo__text" clip-path="url(#clip)" fill="#231f20">
  </g>
</svg>
```

text with applied clip-path

WAAPI EXAMPLE

```
const clipPath = document.querySelector('#clip-rect');
const needle = document.querySelector('.compass__needle');
const compass = document.querySelector('.compass__outside');
```

getting the elements

```
const clipPath = document.querySelector('#clip-rect');
const needle = document.querySelector('.compass__needle');
const compass = document.querySelector('.compass__outside');
```

```
const easingSineIn = 'cubic-bezier(0.47, 0, 0.745, 0.715)';
const easingSineOut = 'cubic-bezier(0.39, 0.575, 0.565, 1)';
```

getting the elements

defining some custom easing

```
const clipPath = document.querySelector('#clip-rect');
const needle = document.querySelector('.compass__needle');
const compass = document.querySelector('.compass__outside');
const easingSineIn = 'cubic-bezier(0.47, 0, 0.745, 0.715)';
const easingSineOut = 'cubic-bezier(0.39, 0.575, 0.565, 1)';
```

getting the elements

```
const duration = {
   duration: 2000,
   easing: easingSineOut,
   iterations: 'Infinity',
   direction: 'alternate',
```

defining some custom easing

```
const clipPath = document.querySelector('#clip-rect');
                                                          calling the animations
const needle = document.querySelector('.compass__needle');
const compass = document.querySelector('.compass__outside');
const duration = {
    duration: 2000,
    easing: easingSineOut,
    iterations: 'Infinity',
    direction: 'alternate',
};
                   clipPath.animate([
                     { transform: scaleX(0) }, // 0%
                     { transform: scalex(1) } // 100%
                   ], duration);
```

all animated properties different easing on need to be in every keyframe last keyframe

```
needle.animate([
   { opacity: 0, transform: 'rotate(0)', offset: 0 },
   { opacity: 0, transform: 'rotate(0)', offset: 0.5 },
   { opacity: 1, transform: 'rotate(380deg)', offset: 1, easing: easingSineIn }
], duration);
```

the animation for the needle starts at 50%



WAAPI CONTROLS

.animate() returns an animation object

```
document
    .querySelector('body')
    .addEventListener('click', () => {
        needleAnimation.playbackRate += 0.1;
    })
```

speeding up
the animations
when we click the body

WAAPI CONTROLS

```
const animation = element.animate(/* animation */);
console.log(animation.playState); //"running"
animation.pause(); //"paused"
animation.play(); //"running"
animation.cancel(); //"idle"... jump to original state
animation.finish(); //"finished"...jump to end state
animation.reverse(); // play animation in reverse
animation.playbackRate = 1.5; // play faster
```

WAAPI CALLBACKS

```
const animation = element.animate(/* animation */);
animation.onfinish = function() {
  console.log("Animation finished");
};
animation.oncancel = animation.effect.target.remove();
```

WAAPI ACCESSIBILITY

```
:root {
   --animation-timing: 0.2s;
                                            CSS variables
.btn {
 transition: transform var(--animation-timing) ease-out;
@media (prefers-reduced-motion: reduce) {
  :root {
    --animation-timing: 0;
```

WAAPI ACCESSIBILITY

```
const mediaQuery = window.matchMedia('(prefers-reduced-motion: reduce)');

mediaQuery.addEventListener('change', () => {
    document.getAnimations().forEach(
        function (animation) {
        animation.cancel();
      }
    );
});
```

cancelling animations in Javascript

WHY THE WEB ANIMATIONS API?

- API provided by the browser
- no extra Javascript code like for libraries
- can render animations over the compositor thread (no JS stalling)
- choreographed & chained animations
- a native & more powerful alternative to CSS animations

Web Animations API - wd

Lets you create animations that are run in the browser, as well as inspect and manipulate animations created through declarative means like CSS.

| Current aligned Usage relative Date relative | | | Apply filters | Show all | ? | | | | | | | | | |
|--|-------|----------------------|---------------|--------------------|-----------------|----------------------|--------------|----------------------|-----------------------|---------------|-----------------------|------------------------|-----------|------|
| IE | Edge | Firefox | Chrome | Safari | Opera | iOS Safari * | Opera Mini * | Android * Browser | Blackberry Browser | Opera Mobile* | Chrome for Android | Firefox for Android | IE Mobile | UC I |
| | | 2-32 | | | | | | | | | | | | |
| | | 33-46 | 4-35 | | 10-22 | | | | | | | | | |
| | | ³ 47 - 58 | 36-38 | 3.1 - 10.1 | | 3.2-10.3 | | | | | | | | |
| 6-10 | 12-17 | ² 59-66 | 39-73 | ⁴ 11-12 | 26-57 | ⁴ 11-12.1 | | 2.1 - 4.4.4 | 7 | 12-12.1 | | | 10 | |
| 11 | 18 | 67 | 74 | 12.1 | ² 58 | 12.2 | all | 2 67 | 10 | 2 46 | 74 | 66 | 11 | . 2 |
| | 75 | 68-69 | 75-77 | 13-TP | | 13 | | | | | | | | |

ATTENTION & FEEDBACK

"Feedback indicates causation between two or more events, often used to connect a user's interaction with the interface's reaction"

Rachel Nabors - Animation at Work

I recognised that you clicked this button

giving a reaction to user input

I recognised that you clicked this button

You entered some information, but it's not correct

giving a reaction to user input

cause & effect of the users input

I recognised that you clicked this button

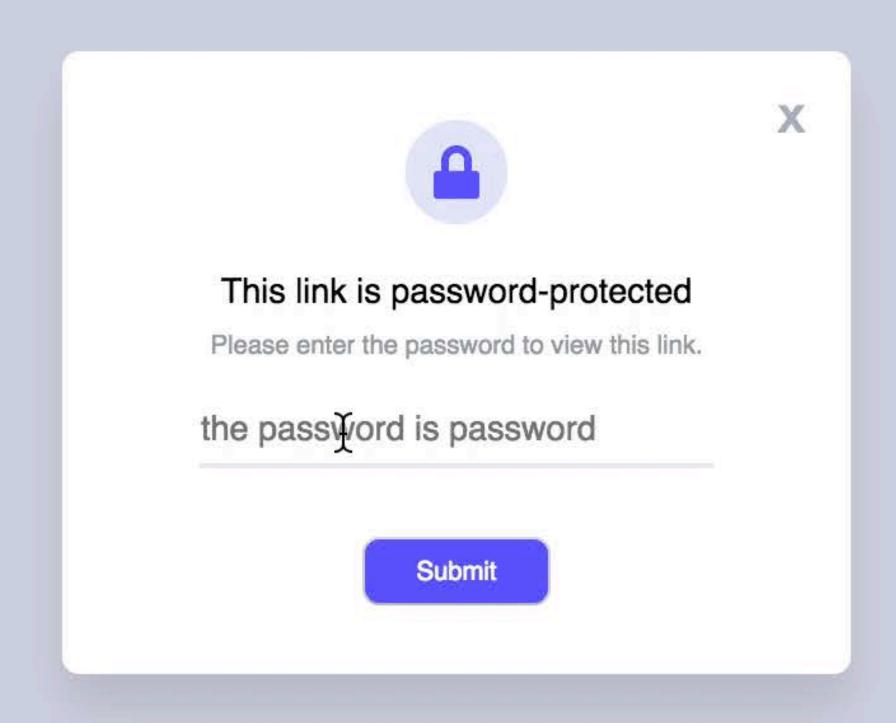
You entered some information, but it's not correct

You submitted the form, but our system is still processing it

giving a reaction to user input

cause & effect of the users input

making system activity visible



To get to the next step, click this prominent item

showing possible actions

To get to the next step, click this prominent item

We created this explanatory animation in order for you to better understand our process

showing possible actions

teaching the user about a process, storytelling

To get to the next step, click this prominent item

We created this explanatory animation in order for you to better understand our process

You had to read a lot of information, so we added this kitten animation to entertain you

showing possible actions

teaching the user about a process, storytelling

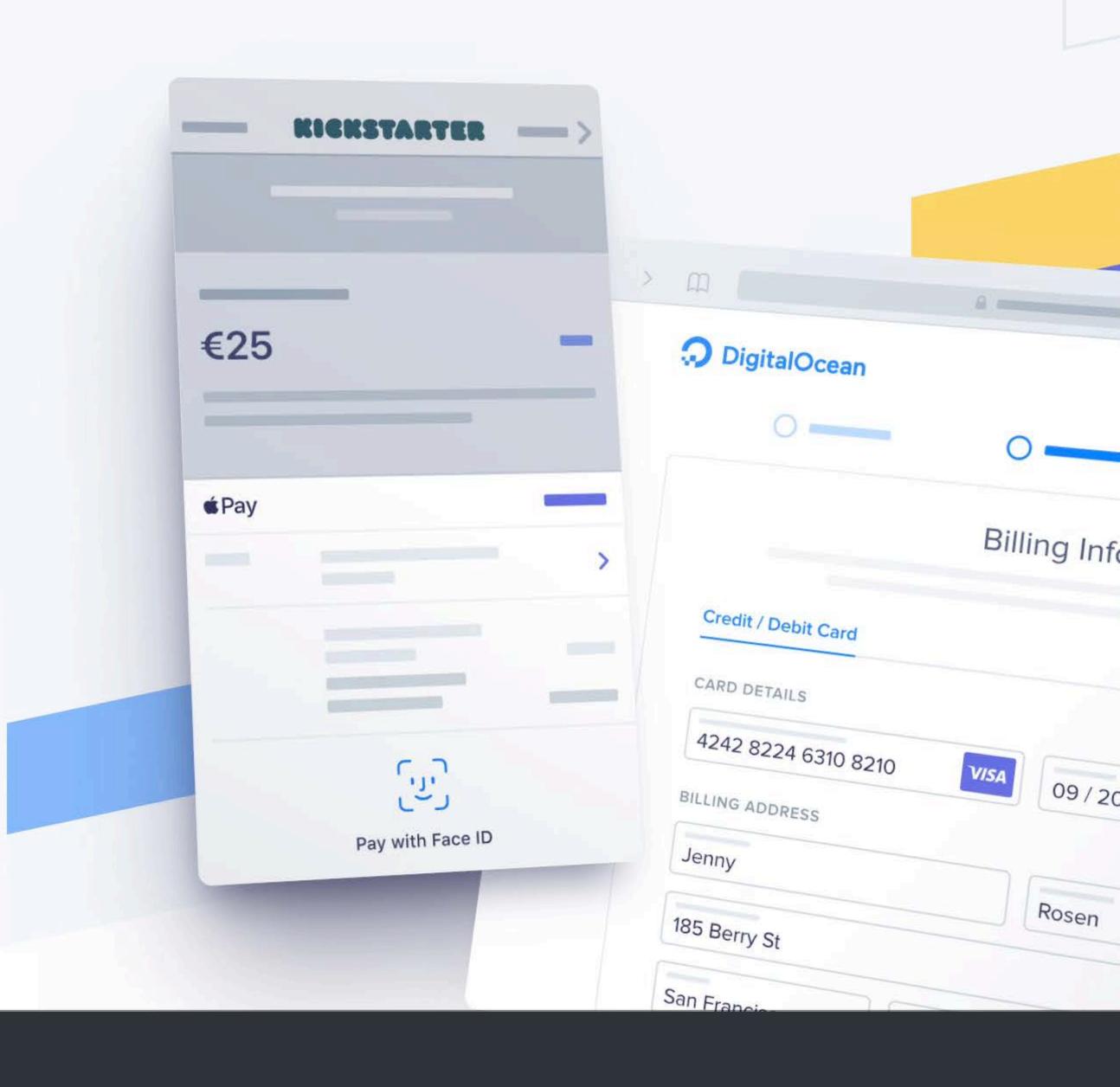
adding fun and reinforcing branding

A complete payments platform engineered for growth

Whether you're looking to bill customers on a recurring basis, set up a marketplace, or simply accept payments, do it all with a fully integrated, global platform that can support online and in-person payments.

CREATE ACCOUNT

EXPLORE FEATURES



ATTENTION & FEEDBACK

Reactive Animation

"A reactive animation is one involving discrete changes, due to events."

Conal Elliott



Conal Elliott

read here

Recent

March 2018 Paper: The simple essence of automatic differentiation

February 2017 Paper: Compiling to categories

February 2017 Paper: Generic parallel functional programming

June 2016 Joined Target, working in data sciences and optimization, continuing to explore elegant and principled techniques from math and programming language theory for

building fast, correct, and beautiful software.

January 2015 Sadly, Tabula closed down.

October 2011 I'm working at <u>Tabula</u> and having a great time I'm helping with chip specification and working out how to compile Haskell to hardware.

May 2009 Paper: Push-pull functional reactive programming

March 2009 Paper: Beautiful differentiation

March 2009 Paper: <u>Denotational design with type class morphisms</u>

January 2008 New technical blog

November 2007 Google tech talk: Tangible Functional Programming: a modern marriage of usability and composability

October 2007 The paper *Functional Reactive Animation*, co-authored with Paul Hudak, was awarded as the most influential paper of ICFP '97.

```
const element = document.querySelector('.foo');
element.style.setProperty('--x', '10%');
element.style.setProperty('--rotation', '30deg');
```

setting a CSS variable via JS

```
const element = document.querySelector('.foo');
element.style.setProperty('--x', '10%');
element.style.setProperty('--rotation', '30deg');
```

using the set variables in your CSS

```
.foo {
    transform: translateX(var(--x)) rotate(var(--rotation));
}
```

```
const element = document.querySelector('.foo');
element.style.setProperty('--x', '10%');
element.style.setProperty('--rotation', '30deg');
```

CSS variables are inherited from their parent, for better performance set them at the most specific level

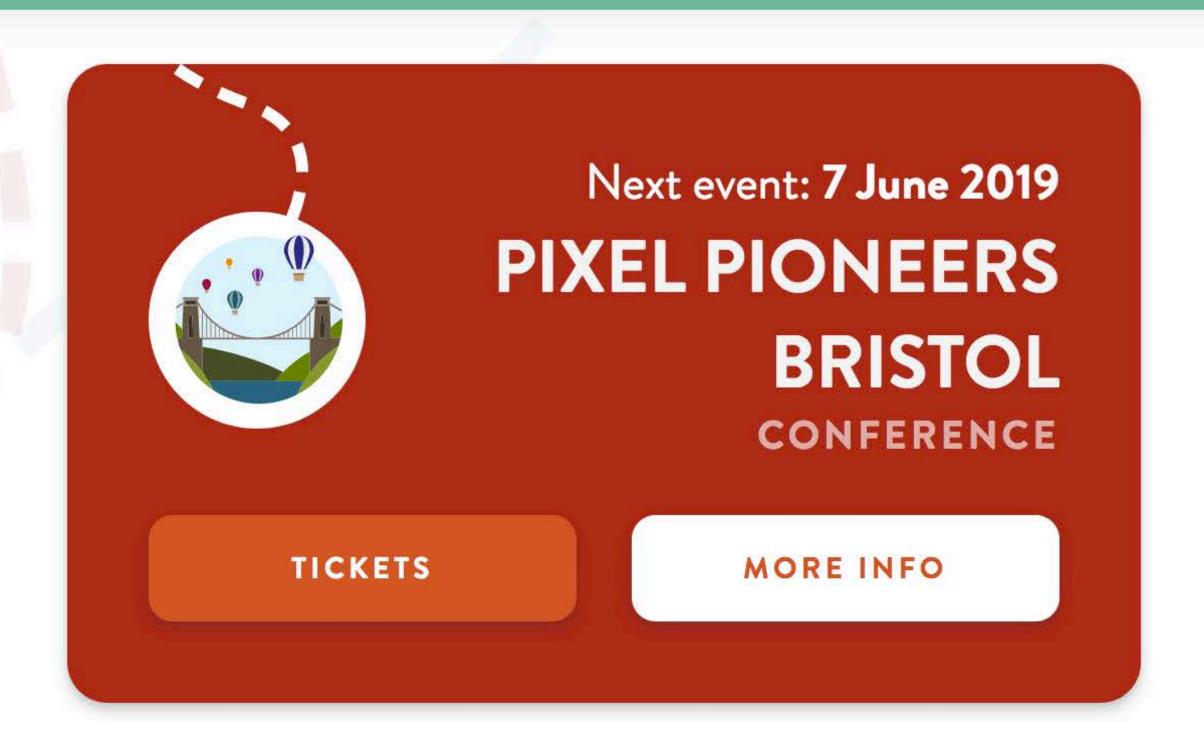


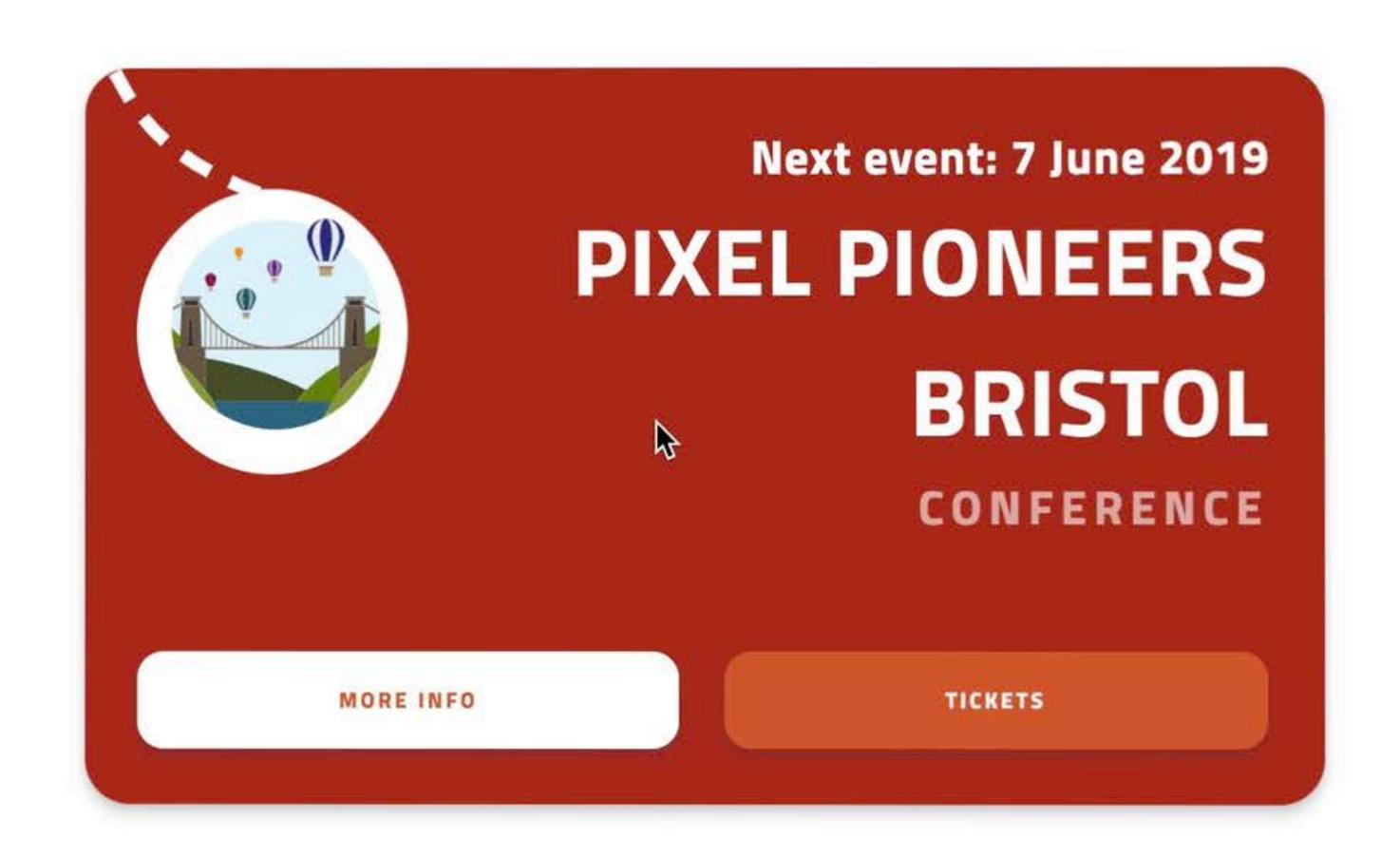
Conferences, workshops, and advice for a better web

Pixel Pioneers is a series of affordable events bringing world-class advice from the brightest minds in digital design and front-end development to the UK.

Organised by Oliver Lindberg, former editor of net magazine with more than 15 years experience in the web industry, the events are tailored to the web's vibrant community.

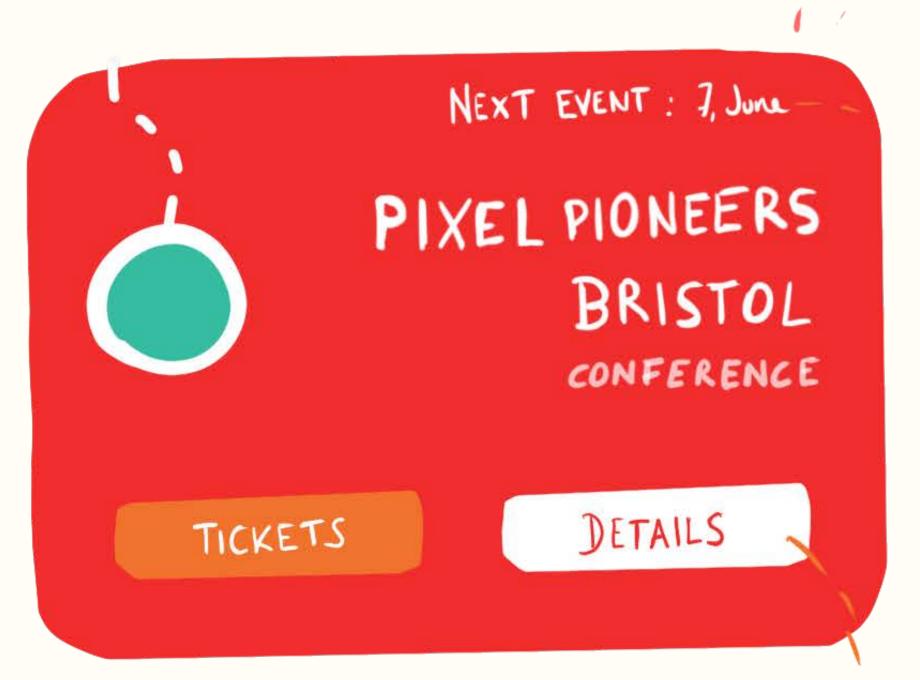
Thank you for your amazing support. We're back in Bristol on 7 June 2019!





codepen.io/lisilinhart/full/oROmWB

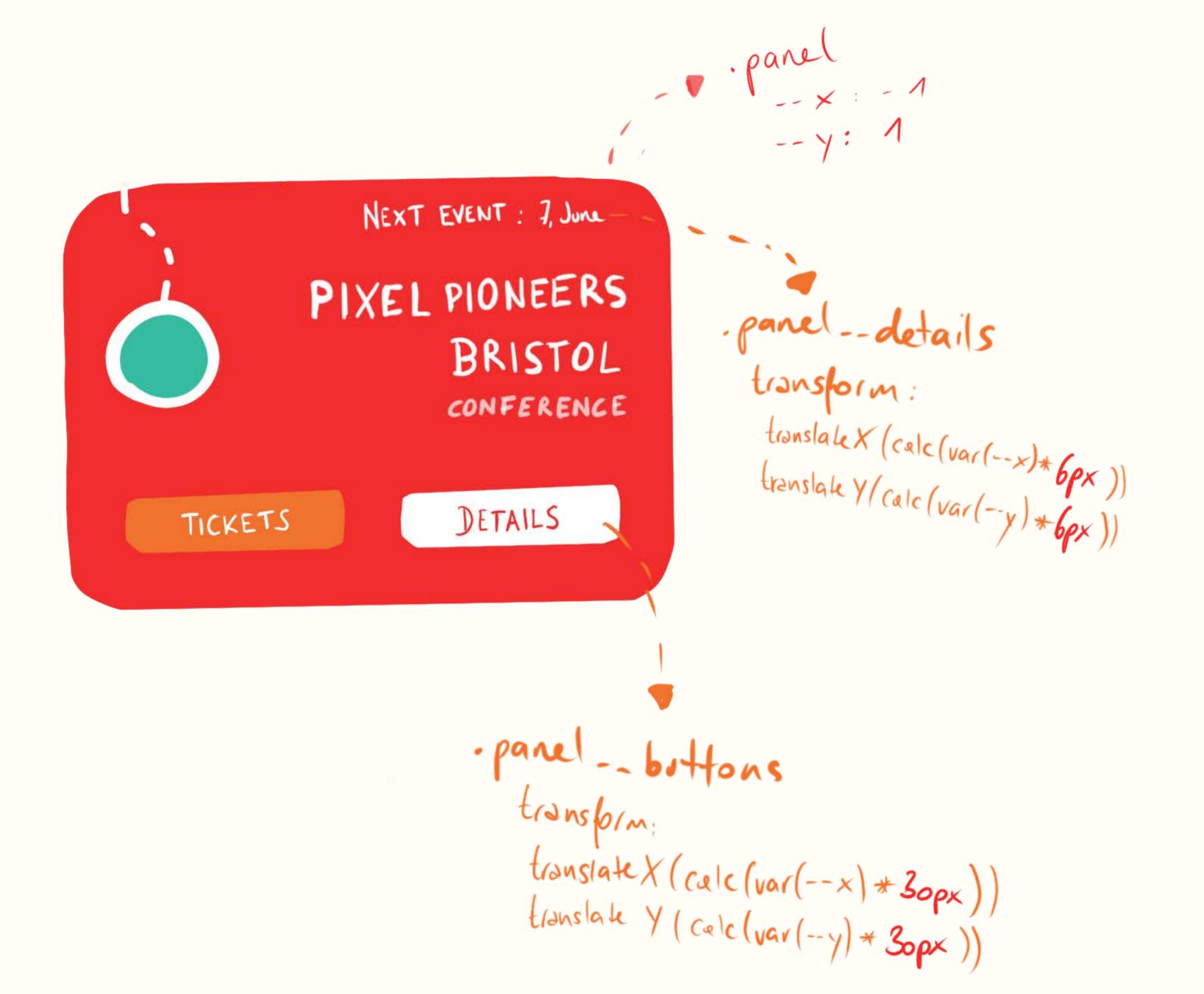
---x:-1
---y:1



NEXT EVENT : 7, June transform:

translate X (calc (var (--x)*6px))

translate Y/(calc (var (--y)*6px)) PIXEL PIONEERS BRISTOL CONFERENCE DETAILS TICKETS



```
let x, y, rAF;
const panel = document.querySelector('.panel');
```

define variables for changes and requestAnimationFrame

```
let x, y, rAF;
const panel = document.querySelector('.panel');
```

```
document.addEventListener('mousemove', (e) => {
    x = (e.clientX / window.innerWidth) - 0.5) * 2;
    y = (e.clientY / window.innerHeight) - 0.5) * 2;

// update the DOM performantly with rAF
    rAF = rAF || requestAnimationFrame(update);
});
```

define EventListener and calculate X & Y from event values that are pixels

```
let x, y, rAF;
const panel = document.querySelector('.panel');

document.addEventListener('mousemove', 10) => {
    x = (e.clientX / window.innerWidth) - 0.5) * 2;
    y = (e.clientY / window.innerHeight) - 0.5) * 2;

// update the DOM performantly with rAF
    rAF = rAF || requestAnimationFrame(update);
});
```

values need to be between 1 and -1 for easier usage with CSS calc()

```
let x, y, rAF;
const panel = document.querySelector('.panel');

document.addEventListener('mousemove', (e) => {
    x = (e.clientX / window.innerWidth) - 0.5) * 2;
    y = (e.clientY / window.innerHeight) - 0.5) * 2;

// update the DOM performantly with rAF
    rAF = rAF || requestAnimationFrame(update);
});
```

update Dom inside requestAnimationFrame

```
let x, y, rAF;
const panel = document.querySelector('.panel');
document.addEventListener('mousemove', (e) => {
 x = (e.clientX / window.innerWidth) - 0.5) * 2;
 y = (e.clientY / window.innerHeight) - 0.5) * 2;
 // update the DOM performantly with rAF
 rAF = rAF | requestAnimationFrame(update);
});
```

```
function update(){
  panel.style.setProperty('--x', x );
  panel.style.setProperty('--y', y );
  rAF = null;
}
```

set CSS variables and reset requestAnimationFrame when done

```
.panel {
    transform: matrix3d(
        1, 0, 0, calc(var(--x) * 0.0002),
        0, 1, 0, calc(var(--y) * 0.0002),
        0, 0, 1, 1,
        0, 0, 0, 1);

    transition: transform 0.3s ease-out;
}
```

rotate panel according to the variables

```
.panel {
    transform: matrix3d(
        1, 0, 0, calc(var(--x) * 0.0002),
        0, 1, 0, calc(var(--y) * 0.0002),
        0, 0, 1, 1,
        0, 0, 0, 1);

    transition: transform 0.3s ease-out;
}
```

transition the transform so the card gets animated smoothly

MATRIX 3D ????

THE MATRIX RESOLUTIONS

I imagine that right now, you're feeling a bit like Alice. Hmm? Transforming down the rabbit hole?



This is your last chance. After this, there is no turning back. You take the blue pill—the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill—you stay in Wonderland and I show you how deep the rabbit-hole goes.

The Red Pill The Blue Pill

MATRIX 3D

```
.translation-matrix {
   transform: matrix3d(
        1, 0, 0, <tx>,
        0, 1, 0, <ty>,
        0, 0, 1, <tz>,
        0, 0, 0, 1);
}
```

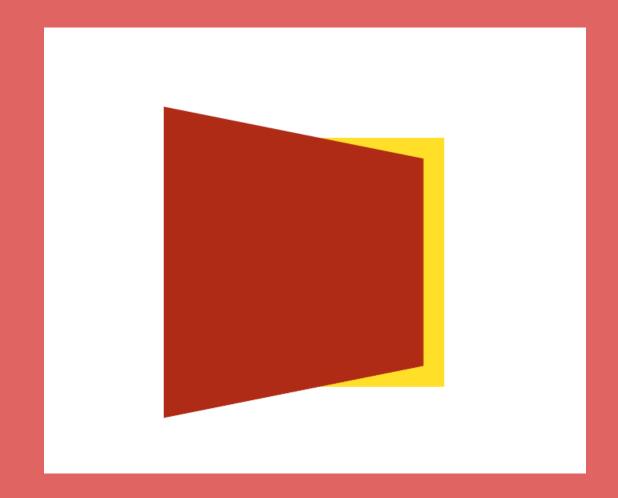
```
.rotate-x-matrix {
   transform: matrix3d(
      1, 0, 0, 1,
      0, cosX, -sinX, 1,
      0, sinX, cosX, 1,
      0, 0, 0, 1);
}
```

```
.rotate-y-matrix {
    transform: matrix3d(
        cosX, 0, -sinX, 1),
        0, 1, 0, 1),
        sinX, 0, cosX), 1)
        0, 0, 0, 1);
}
```

The matrix3d() function is an alternative to the three dimensional transform functions rotate3d(), rotateX(), rotateY(), rotateZ(), translate3d(), translateZ(), scale3d(), scaleZ(), and perspective().

MATRIX 3D

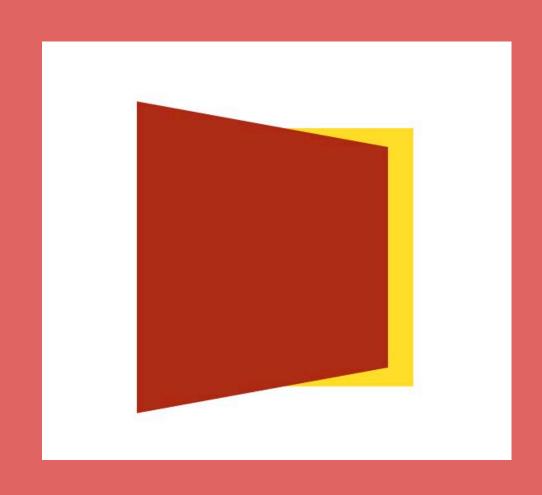
```
.panel {
    transform: matrix3d(
        1, 0, 0, 0.002,
        0, 1, 0, 1,
        0, 0, 1, 1,
        0, 0, 0, 1);
}
```



```
needs perspective
on parent

.parent {
    perspective: 200px;
}

.panel {
    transform: rotateY(30deg);
}
```



MATRIX 3D

```
.panel {
    transform: matrix3d(
        1, 0, 0, 0.002,
        0, 1, 0, 1,
        0, 0, 1, 1,
        0, 0, 0, 1);
    }
```

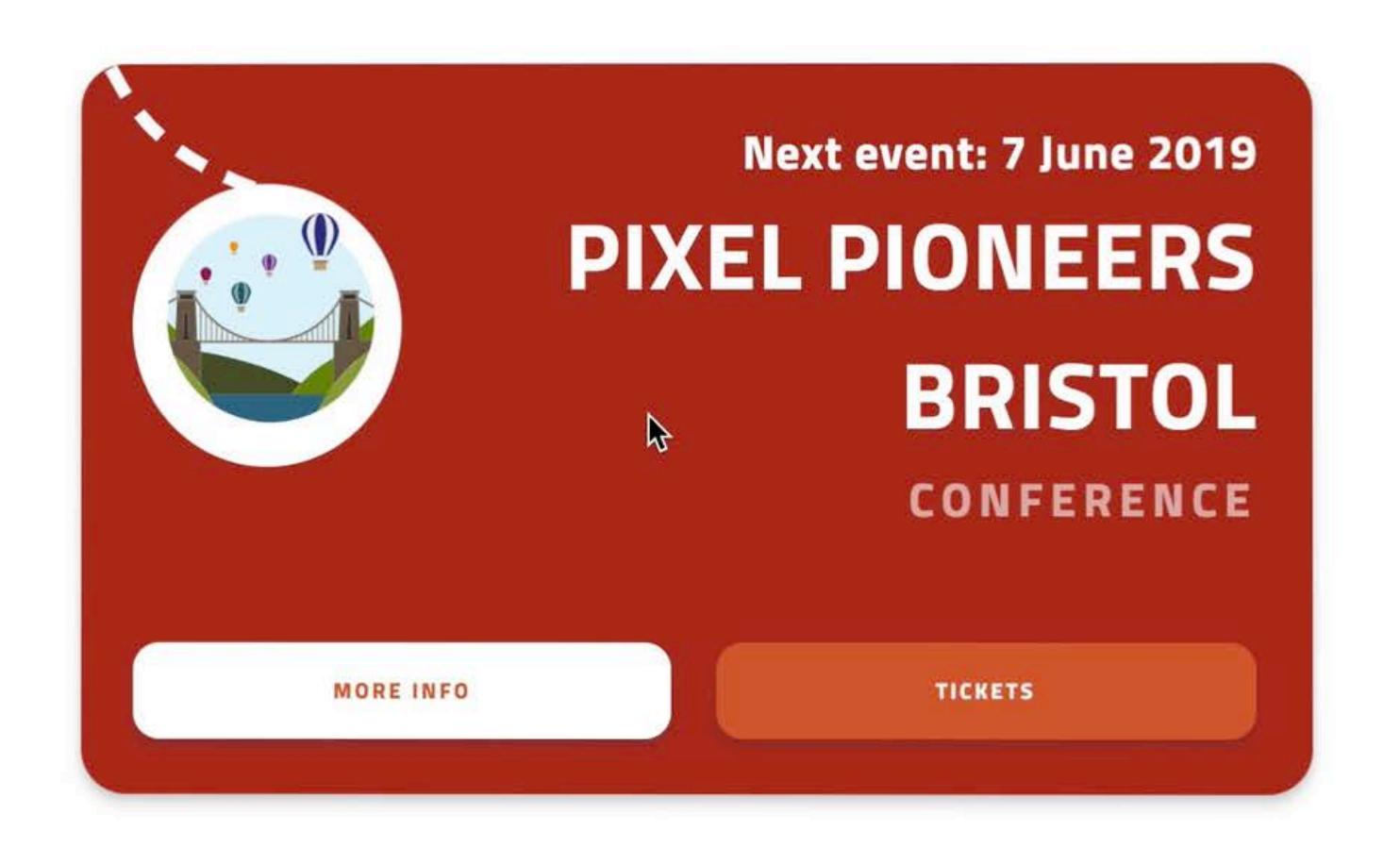
```
.panel {
    transform: rotateY(30deg);
}
without perspective
```

```
.panel__details {
   transform: translate(
     calc(var(--x) * 4px),
     calc(var(--y) * 4px)
.panel__icon {
   transform: translate(
     calc(var(--x) * 30px),
     calc(var(--y) * 30px)
```

difference in translation creates a 3D parallax effect

```
.panel__details {
   transform: translate(
     calc(var(--x) * 4px),
     calc(var(--y) * 4px)
.panel__icon {
   transform: translate(
     calc(var(--x) * 30px),
     calc(var(--y) * 30px)
```

higher values look like they're closer

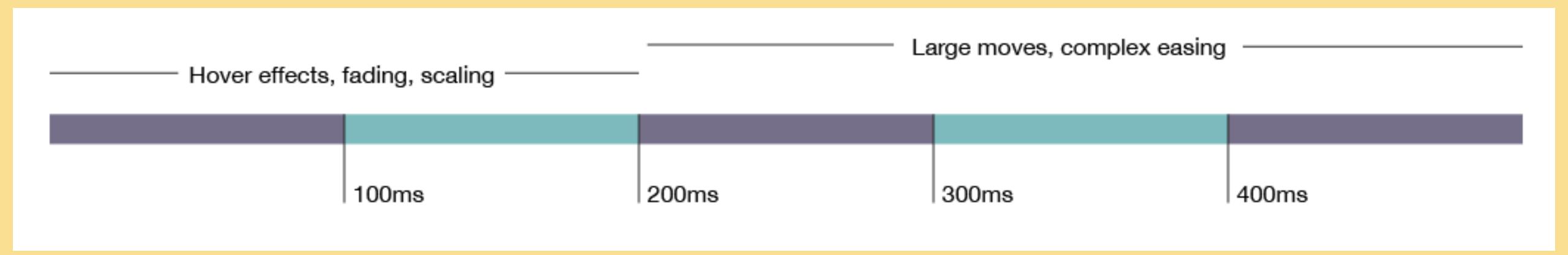


codepen.io/lisilinhart/full/oROmWB

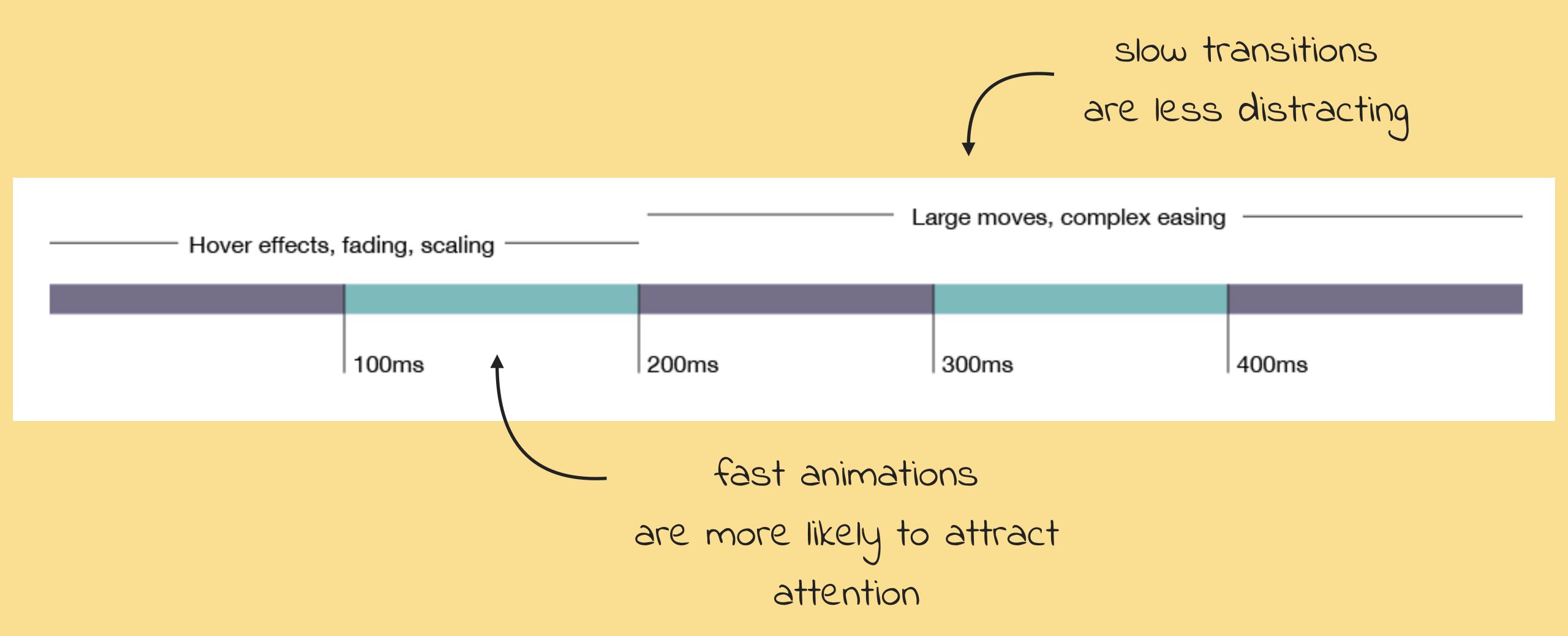
- easily debuggable
- no excessive DOM manipulation
- DOM node independent
- great if you animate multiple child elements
- great for reactive & physics animation
- Transform: Individual Properties



TIMING



TIMING

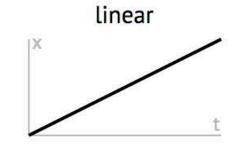


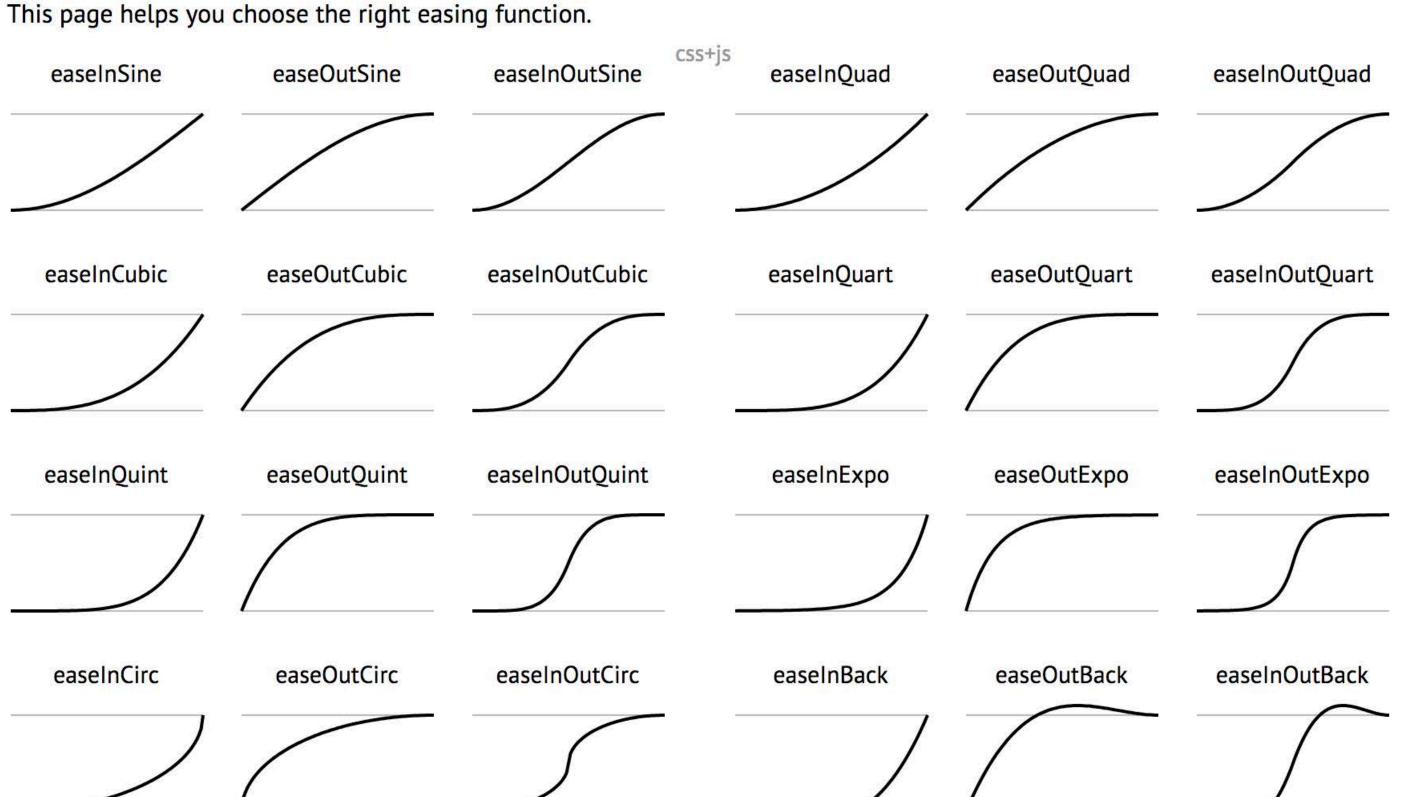
valhead.com/2016/05/05/how-fast-should-your-ui-animations-be/

EASINGS.NET

Easing functions specify the rate of change of a parameter over time.

Objects in real life don't just start and stop instantly, and almost never move at a constant speed. When we open a drawer, we first move it quickly, and slow it down as it comes out. Drop something on the floor, and it will first accelerate downwards, and then bounce back up after hitting the floor.

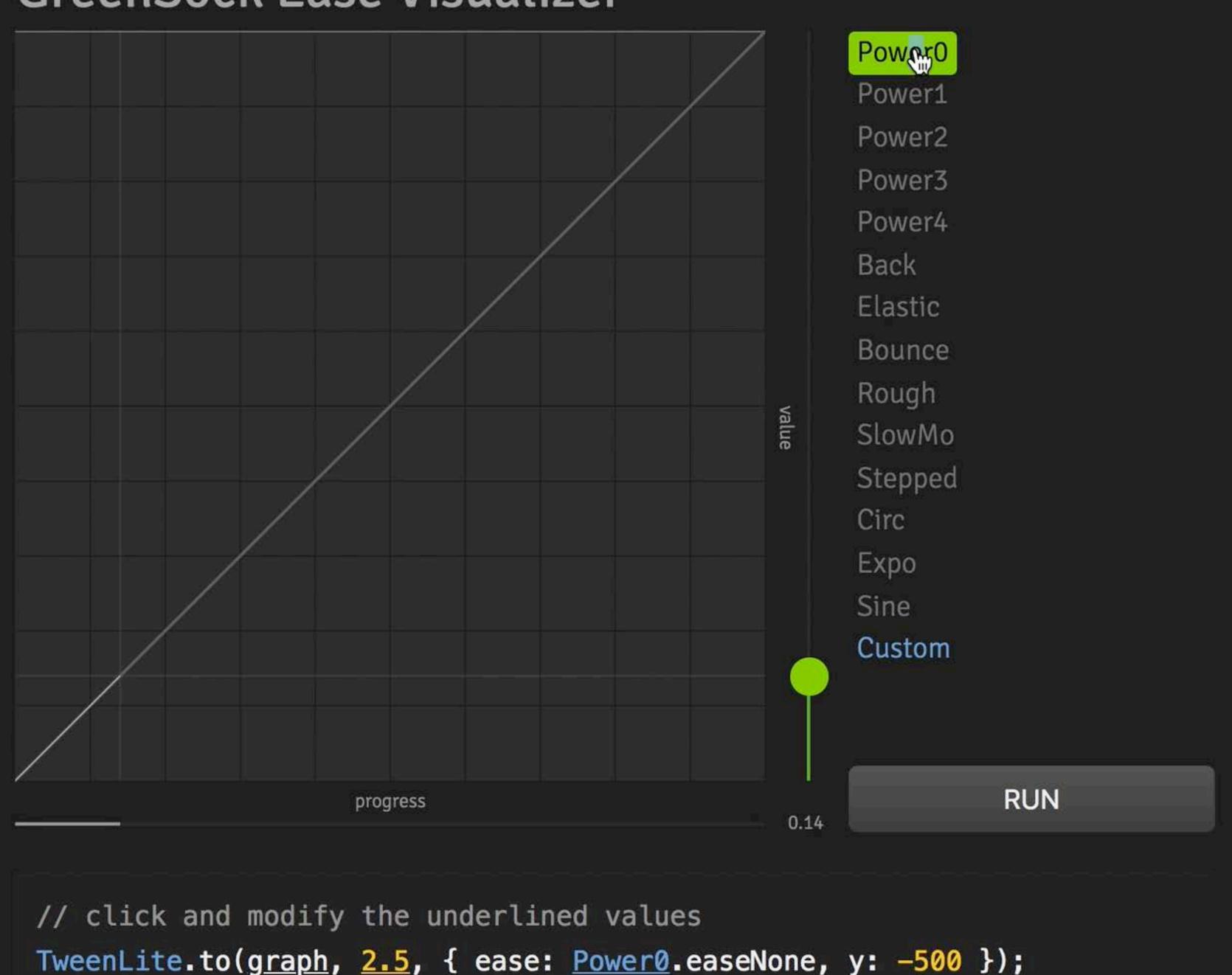




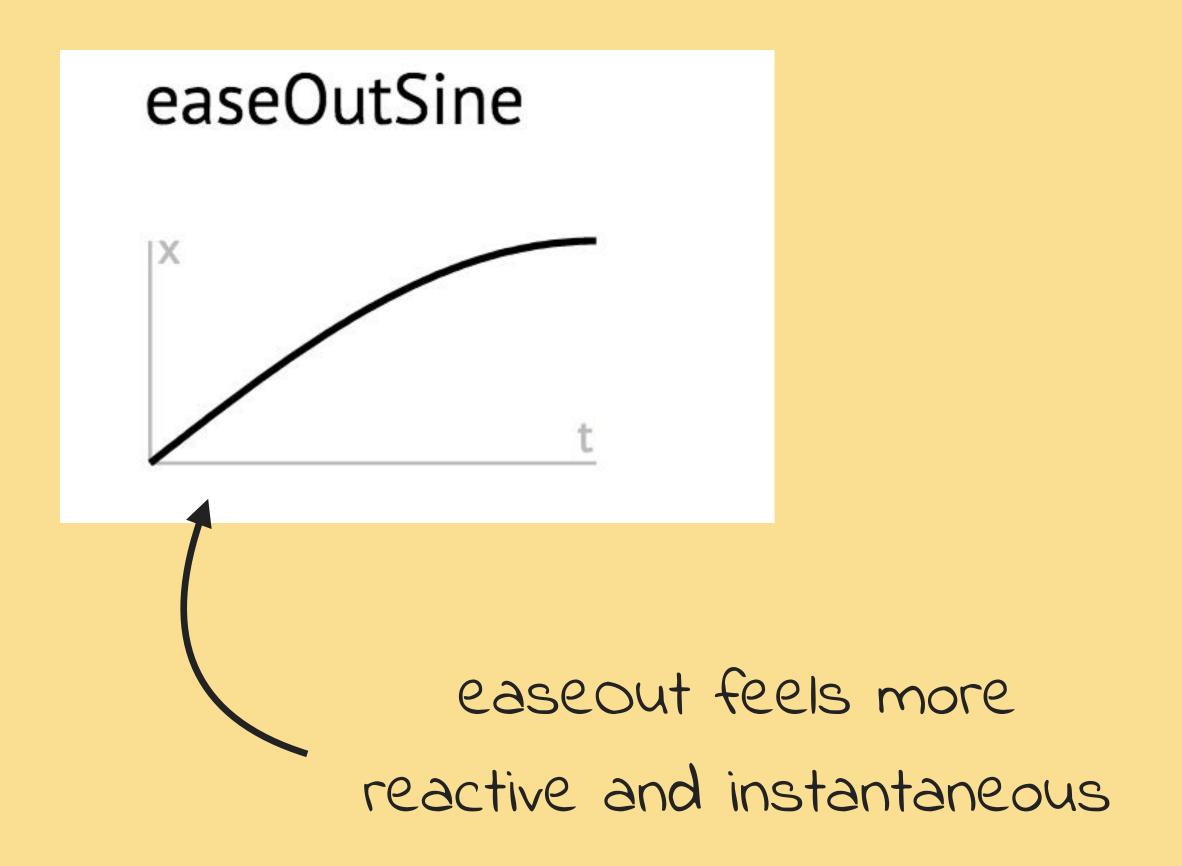
EASINGS.NET

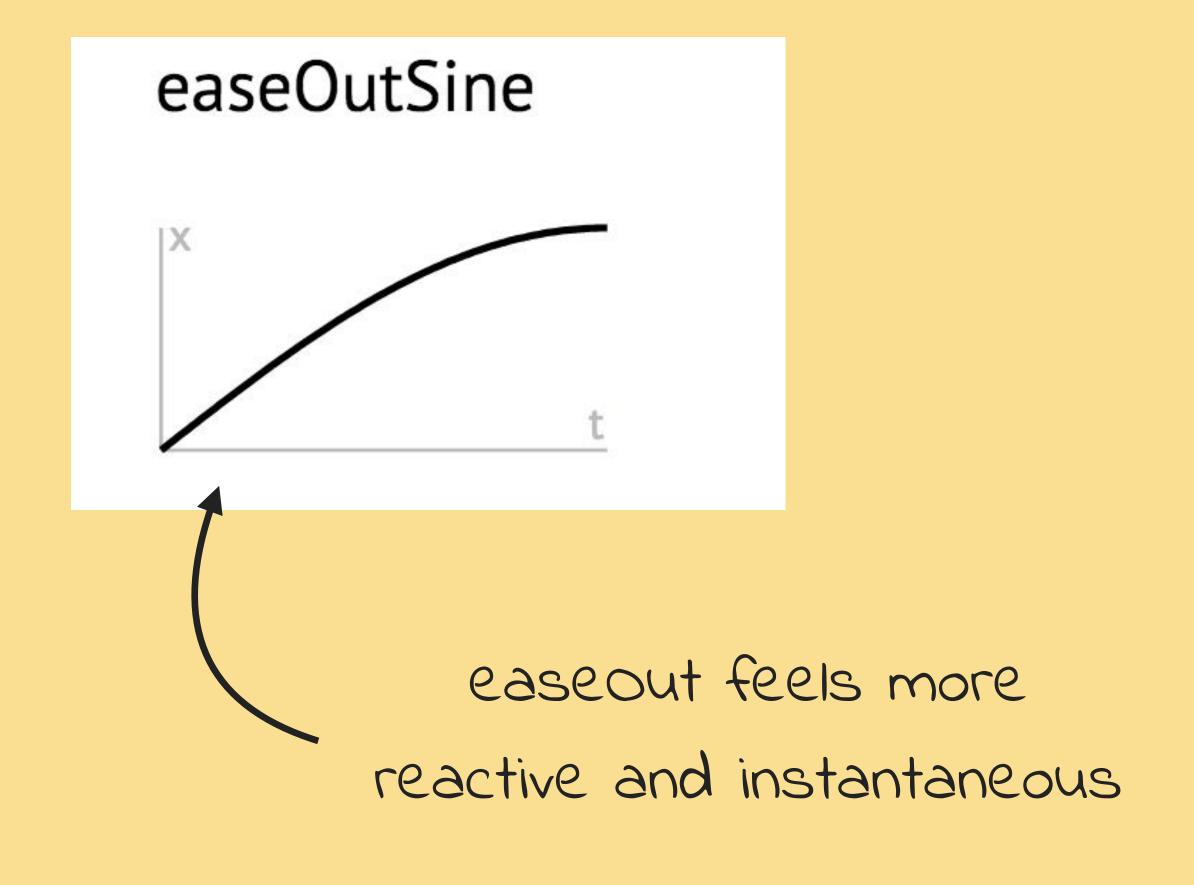


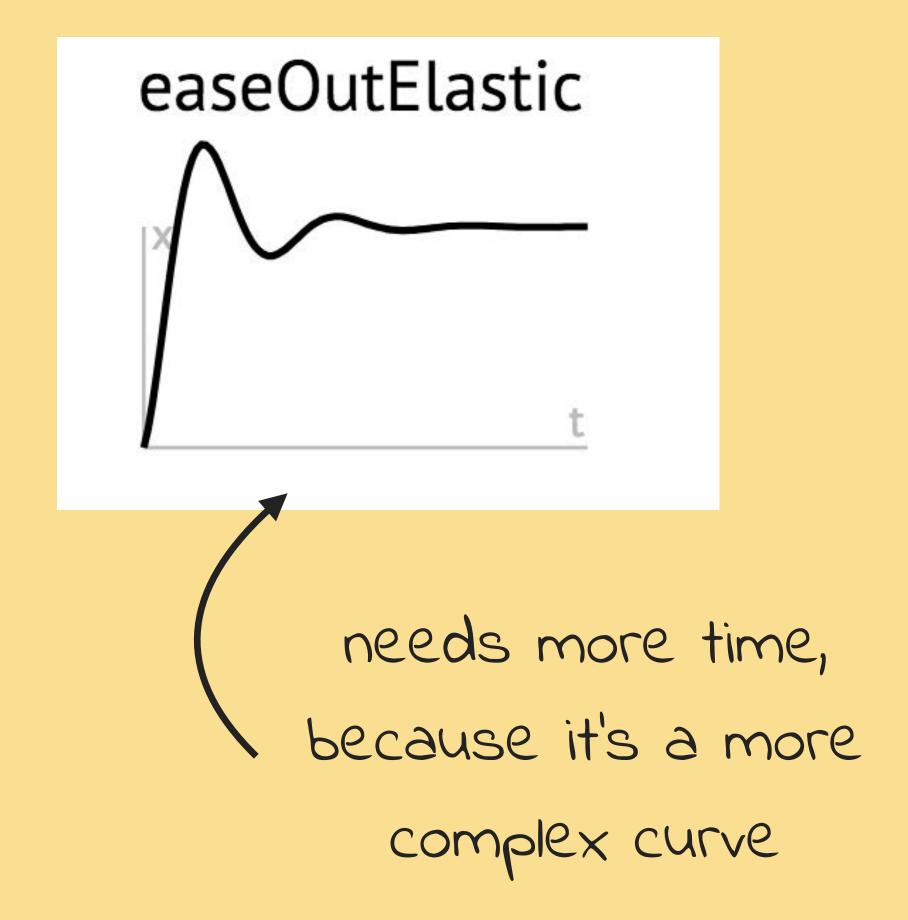
GreenSock Ease Visualizer

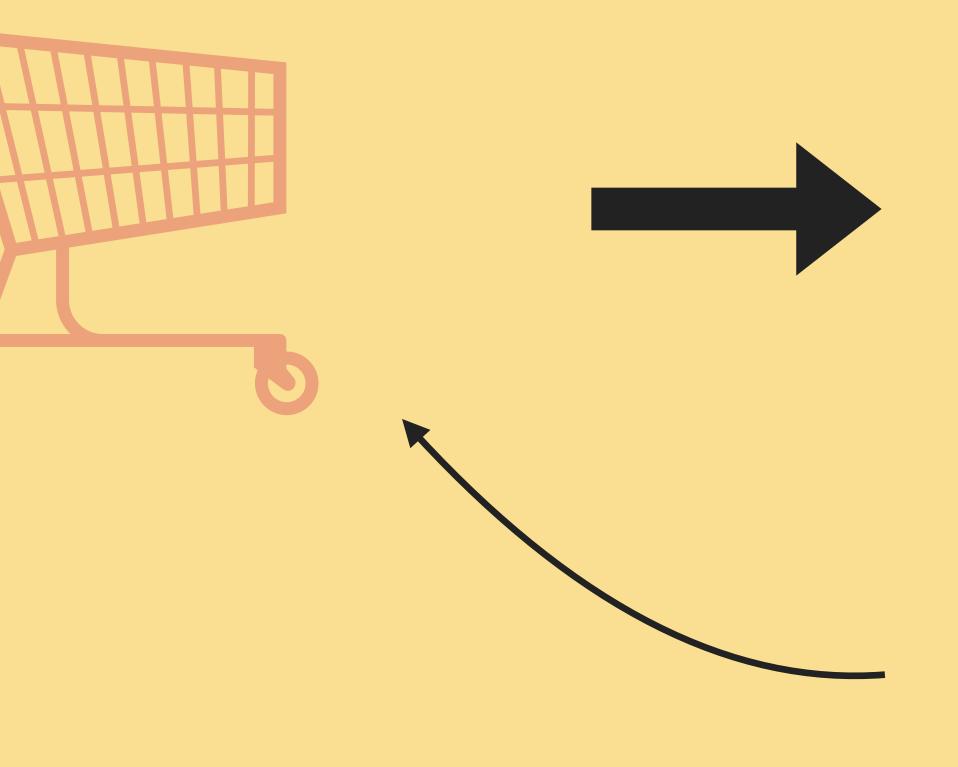


TweenLite.to(graph, 2.5, { ease: Power0.easeNone, y: -500 });



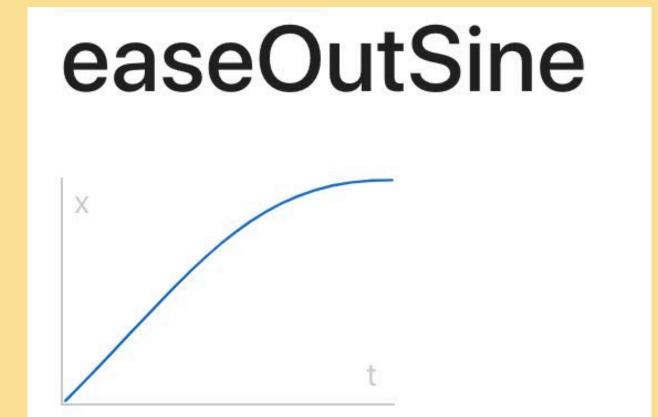








when an element is moving into the screen it should start quick and slow down towards the center







easeInSine



when an element is moving out of the screen it should start slow and speed up towards the end

Opportunities for Animation

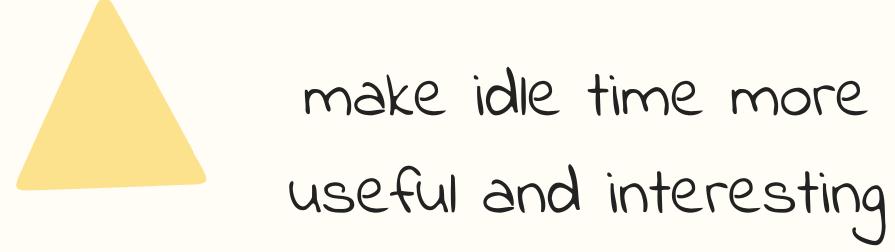




guide the user in a process

Opportunities for

Animation

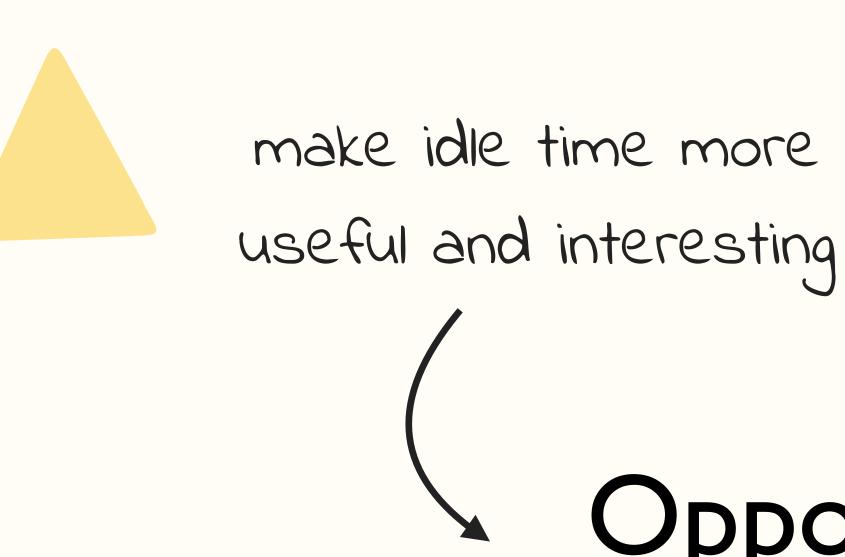


guide the user in a process

Opportunities for

Animation

explain relationships in the information space



guide the user in a process

Opportunities for

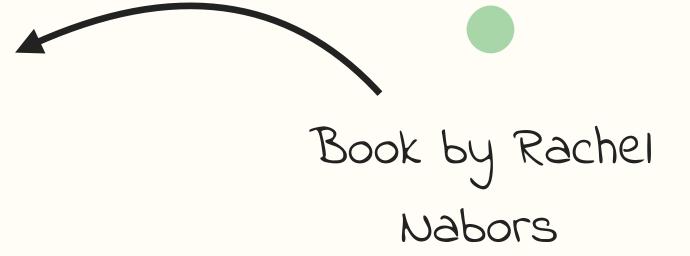
Animation

get the users attention

explain relationships in the information space



Animation At Work - A Book Apart



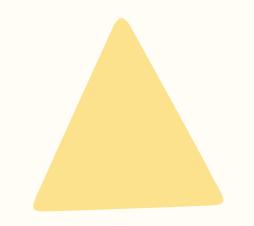


Animation At Work - A Book Apart

Twitch Channel by David Khourshid & Stephen Shaw

keyframe.rs

Book by Rachel Nabors



Animation At Work - A Book Apart

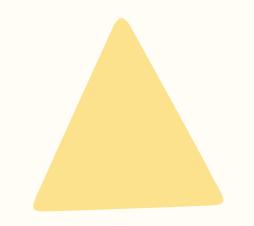
Twitch Channel by David Khourshid & Stephen Shaw

keyframe.rs

animationatwork.slack.com

Book by Rachel Nabors

Slack Channel to get feedback & inspiration



Animation At Work - A Book Apart

Twitch Channel by David Khourshid &

Stephen Shaw

basics to know about animation

keyframe.rs

animationatwork.slack.com

developers.google.com/web/ fundamentals/design-and-ux/animations/ Book by Rachel Nabors

Slack Channel to get feedback & inspiration



Animation At Work - A Book Apart

Twitch Channel by David Khourshid &

Stephen Shaw

basics to know about animation

keyframe.rs

animationatwork.slack.com

developers.google.com/web/ fundamentals/design-and-ux/animations/

uianimationnewsletter.com

Book by Rachel Nabors

Slack Channel to get feedback & inspiration

inspiration for UI animation

"Users should only notice your animation if you need to attract their attention in that moment. Otherwise, micro-interactions and other transitions should be so seamless, users don't even notice that there is animation"

Heather Daggett - Animation At Work

Animate for the user, don't create obstacles,