

# Animation Essentials

*CSS Animations and Transitions*

with [@RachelNabors](#) & Tuna P. Katz



Get started with “SMASHINGROCKS” at [goo.gl/ZL0SVd](https://goo.gl/ZL0SVd)

@RachelNabors RachelNabors.com



seventeen



hella  
emo



Goth.



doing  
what she  
loves for

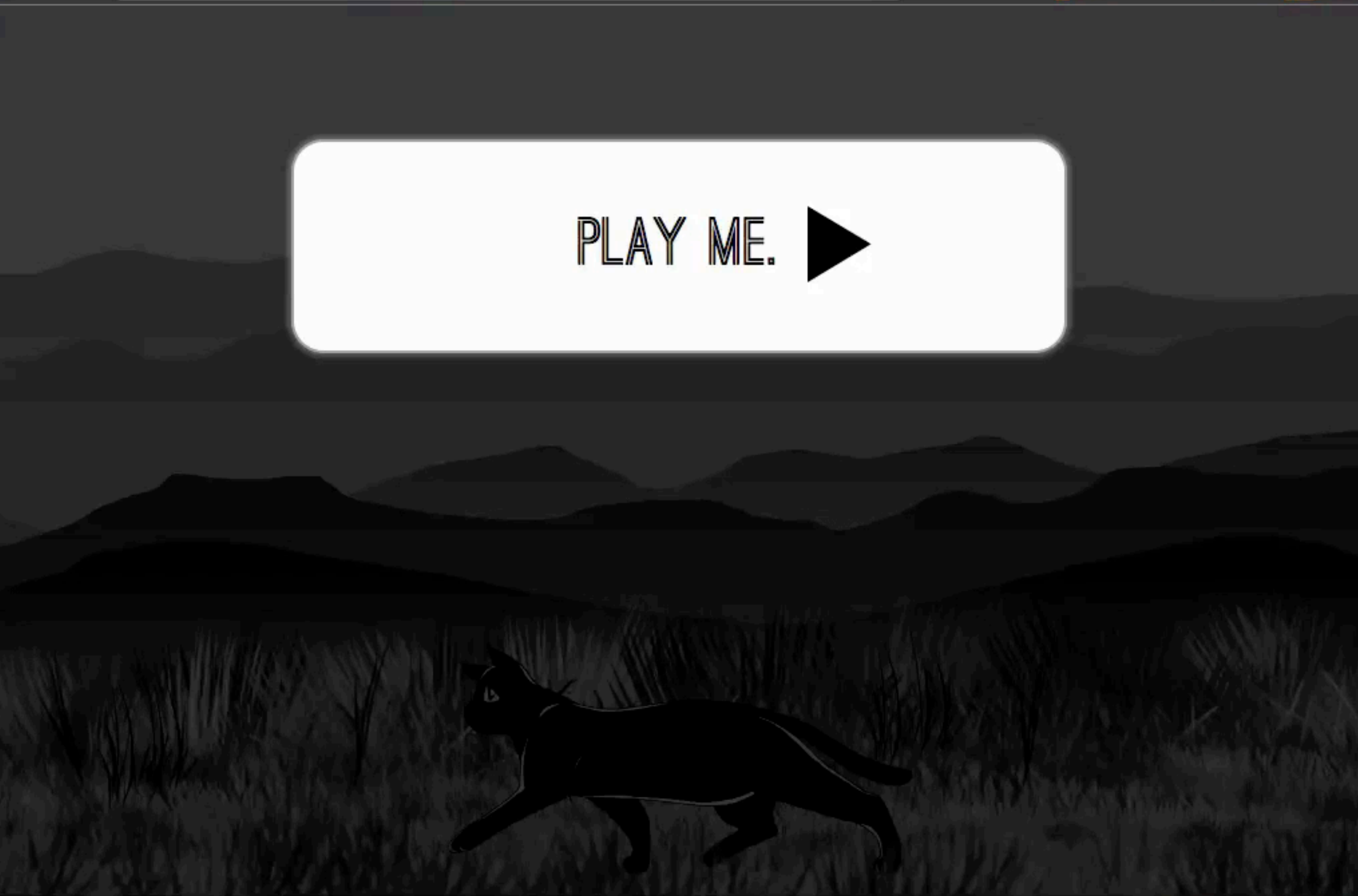


##

hordes of  
fan girls



PLAY ME. ▶







*WebAnimationWeekly.com*

*(It's the best.)*

# You will learn...

- CSS Transitions
- CSS Animations
- Sequencing Animations
- How to Design Performant Animations
- And some handy browser tools for working with animations

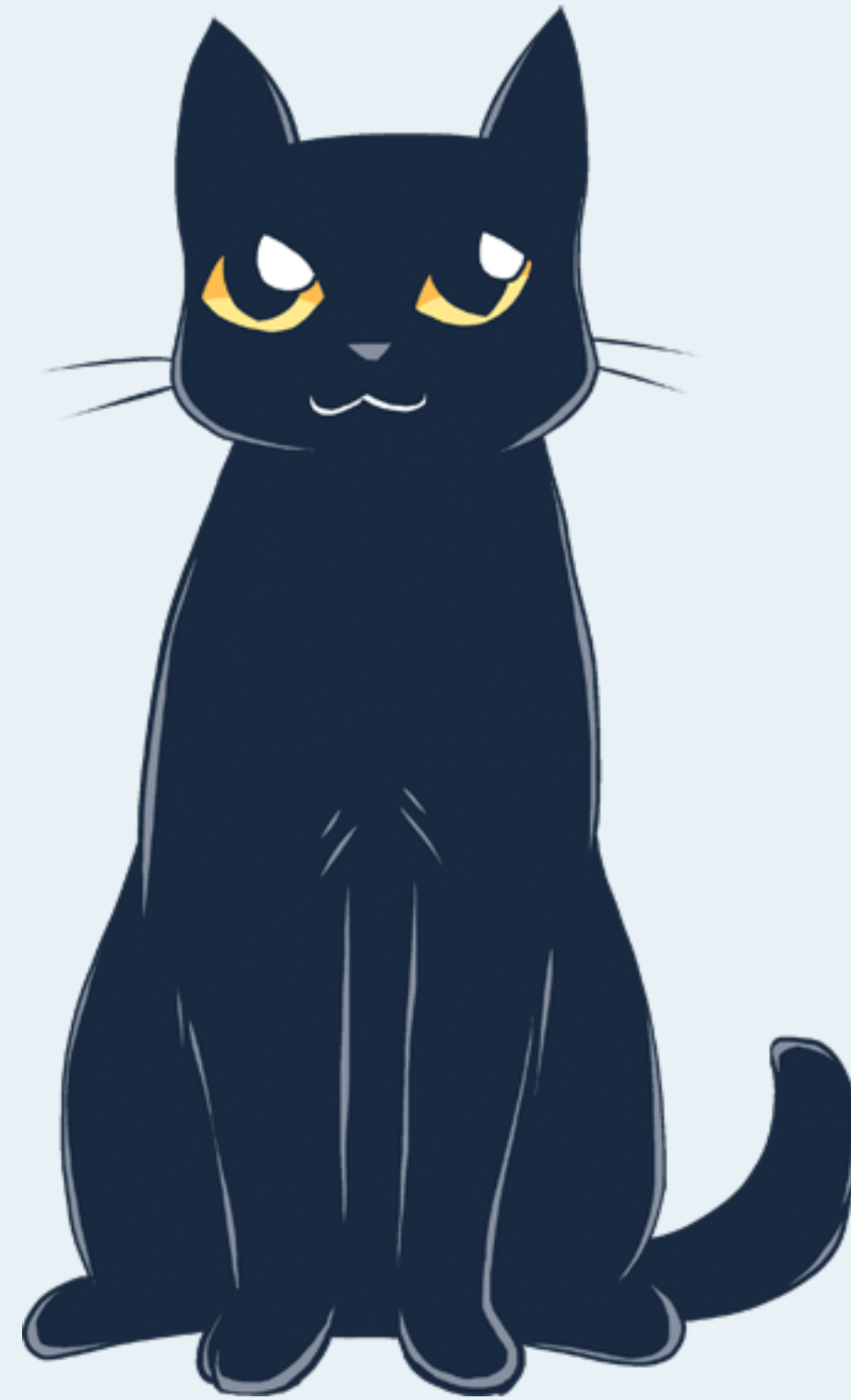
# For your convenience...

All of the exercises use:

- jQuery
- and -prefix-free

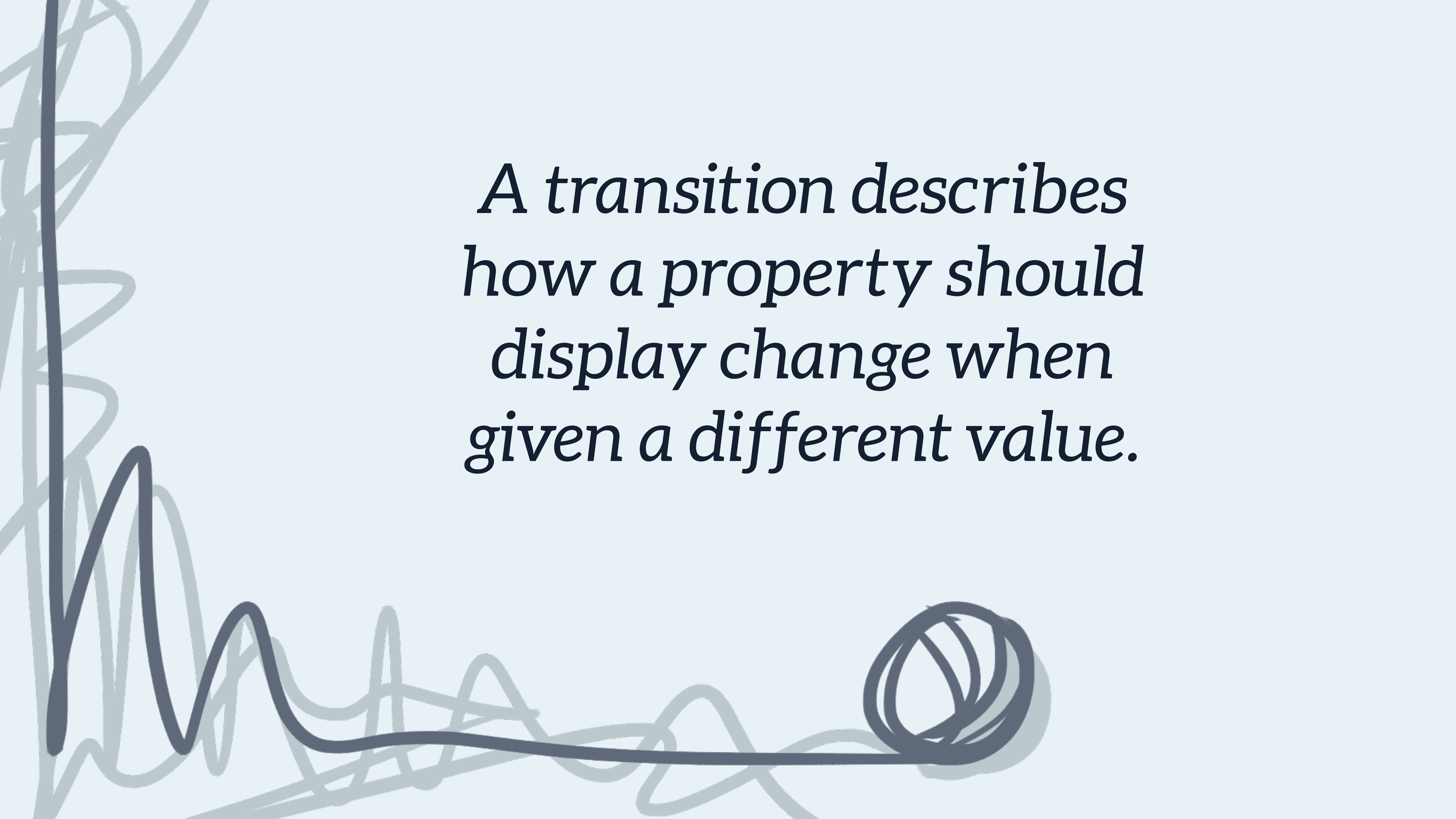


# Our Teacher's Assistant Today



# CSS Transitions





*A transition describes how a property should display change when given a different value.*

```
transition: color 2s;
```

```
transition-property: color;  
transition-duration: 2s;
```

# Anatomy of a transition

- **transition-property** the property you want to transition. *(Only some properties are transitionable, see [goo.gl/Ttk1S2](http://goo.gl/Ttk1S2))*
- **transition-duration** in seconds or milliseconds: 4s or 4000ms
- **transition-timing-function** “cushioning” for the transition, **optional**: defaults to ease
- **transition-delay** the number of milli/seconds to delay the transition before firing it, **optional**

so this must  
be the delay



**transition:** color 2s 100ms;



duration always  
comes first

# Transitioning Multiple Properties





transition-property:

~~all;~~

Don't do it!!

**transition-property: color, transform;**

**transition-duration: 2s, 300ms;**

**transition-delay: 0, 1s;**

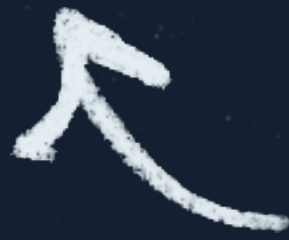


*redundant, bleh*

**transition:**

**color 2s,**

**transform 300ms 1s;**



separate sets of values  
with commas

*exercise*  
**Rolling a Ball**



Start coding! [cdpn.io/fGFDj](https://cdpn.io/fGFDj)

Try transitioning its color, too.


Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors

# Duration



# Three timeframes for user attention

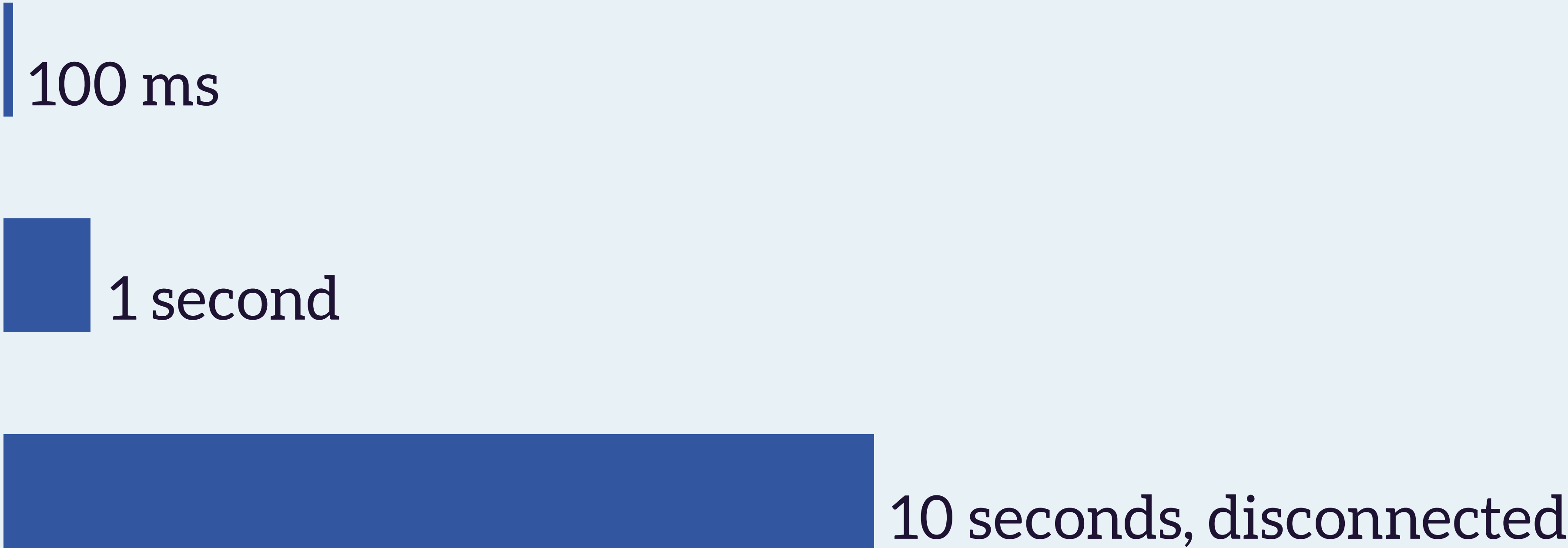


100 ms, instantaneous



1 second, still connected

# Three timeframes for user attention





## SPEAKING SCHEDULE

Mar 26

SFHTML5, San Francisco

May 9-13

Beyond Tellerand, Düsseldorf & DOM in Motion workshop

## ONLINE WORKSHOPS

Apr 2 - UIE

Improve UX with Animation

Jun 30 - Smashing Online

Animation Essentials: CSS Animations and Transitions

Open Enrollment - Learn with Rachel Nabors

Practical Cartooning for Technical Folk

## ANIMATION PROJECTS



The Black Brick Road to Oz #47

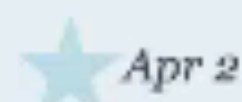


Alice in Videoland

xmix this ~

# WEB ANIMATION AND MOTION DESIGN TRAINING AND WORKSHOPS

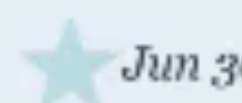
## ONLINE WORKSHOPS



Apr 2

UIE

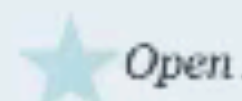
[Improve UX with Animation](#)



Jun 30

Smashing Online

[Animation Essentials: CSS Animations and Transitions](#)



Open Enrollment

Learn with Rachel Nabors

[Practical Cartooning for Technical Folk](#)

AVAILABLE FOR TRAINING IN HOUSE AND AT CONFERENCES

[rachelnabors.com](http://rachelnabors.com)

TABLE OF CONTENTS

COURSES  
SPEAKING  
TRAINING  
WRITING  
ABOUT

STAY IN TOUCH!

MAILING LIST  
EMAIL





“However long your pre-production  
animation, halve its duration...  
then halve it again.”

*-Studio animation rule of thumb*



Rachel Nabors



TABLE OF CONTENTS

COURSES  
SPEAKING  
TRAINING  
WRITING  
ABOUT

STAY IN TOUCH!

MAILING LIST  
EMAIL



### SPEAKING SCHEDULE

Mar 26

SFHTML5, San Francisco

May 9-13

Beyond Tellerand, Düsseldorf & DOM in Motion workshop

### ONLINE WORKSHOPS

Apr 2 - UIE

Improve UX with Animation

Jun 30 - Smashing Online

Animation Essentials: CSS Animations and Transitions

Open Enrollment - Learn with Rachel Nabors

Practical Cartooning for Technical Folk

### ANIMATION PROJECTS



The Black Brick Road to Oz #47



Alice in Videoland



Xmix this ~

# WEB ANIMATION AND MOTION DESIGN TRAINING AND WORKSHOPS

## ONLINE WORKSHOPS

★ Apr 2  
UIE

[Improve UX with Animation](#)

★ Jun 30

Smashing Online

[Animation Essentials: CSS Animations and Transitions](#)

★ Open Enrollment

Learn with Rachel Nabors

[Practical Cartooning for Technical Folk](#)

## AVAILABLE FOR TRAINING IN HOUSE AND AT CONFERENCES

I teach from a place of creative coding and my rich history as an award-winning cartoonist. As a web designer turned front-end developer, I connect the silos of design, UX, and programming wherever I teach. This empathy for all parts of the human equation required for building a better web lets me engage all



*250~300 ms:  
sweet spot for  
many animations*

*faster != better*





TABLE OF CONTENTS

COURSES  
SPEAKING  
TRAINING  
WRITING  
ABOUT

STAY IN TOUCH!

MAILING LIST  
EMAIL



### SPEAKING SCHEDULE

Mar 26

SFHTML5, San Francisco

May 9-13

Beyond Tellerand, Düsseldorf & DOM in Motion workshop

### ONLINE WORKSHOPS

Apr 2 - UIE

Improve UX with Animation

Jun 30 - Smashing Online

Animation Essentials: CSS Animations and Transitions

Open Enrollment - Learn with

Rachel Nabors

Practical Cartooning for Technical Folk

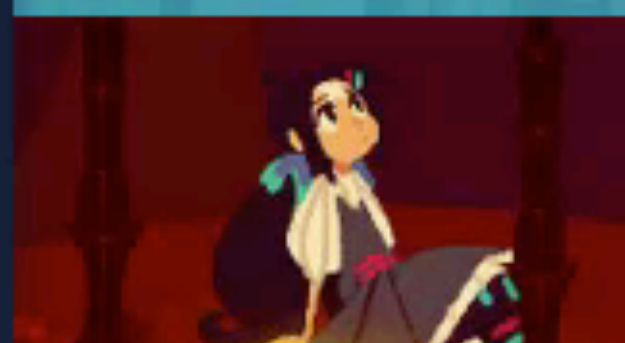
### ANIMATION PROJECTS



The Black Brick Road to Oz #47



Alice in Videoland



Xmix this 2

# WEB ANIMATION AND MOTION DESIGN TRAINING AND WORKSHOPS

## ONLINE WORKSHOPS

★ Apr 2  
UIE

[Improve UX with Animation](#)

★ Jun 30

Smashing Online

[Animation Essentials: CSS Animations and Transitions](#)


★ Open Enrollment

Learn with Rachel Nabors

[Practical Cartooning for Technical Folk](#)

## AVAILABLE FOR TRAINING IN HOUSE AND AT CONFERENCES

I teach from a place of creative coding and my rich history as an award-winning cartoonist. As a web designer turned front-end developer, I connect the silos of design, UX, and programming wherever I teach. This empathy for all parts of the human equation required for building a better web lets me engage all



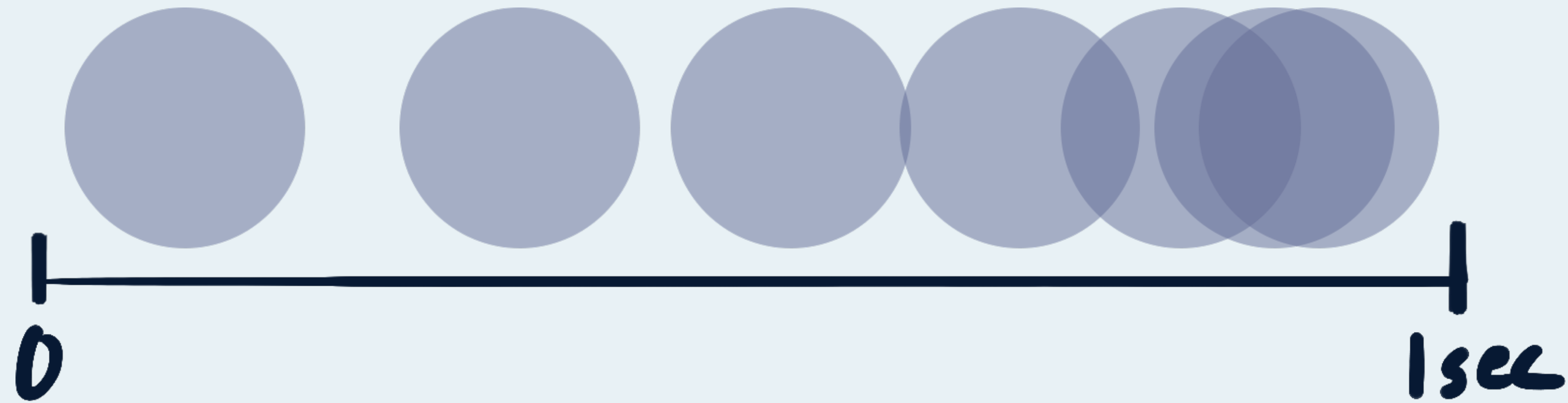
*You can slow down  
transitions with  
Dev Tools to get a  
good look at 'em.*

# Timing Functions

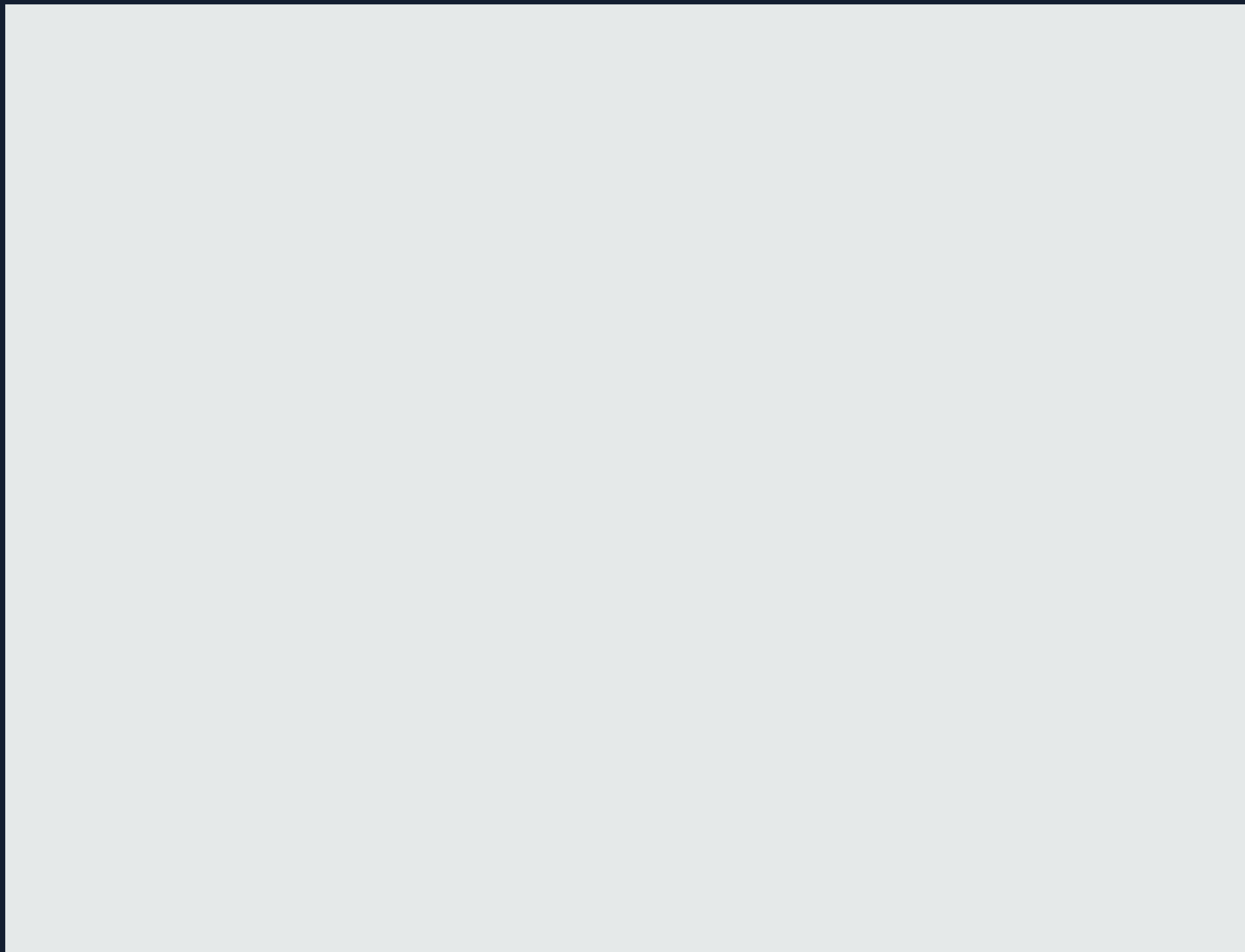


# *Easing*

Easing, also known as “cushioning” in studio animation, describes an animation’s rate of change over time.









# Timing Function Values

- linear
- ease-in
- ease-out
- ease-in-out
- steps (more on this later)



*Replace “ease”  
with “slow” to better  
remember the different  
timing effects.*



*What you  
really want:*  
**cubic-bezier**

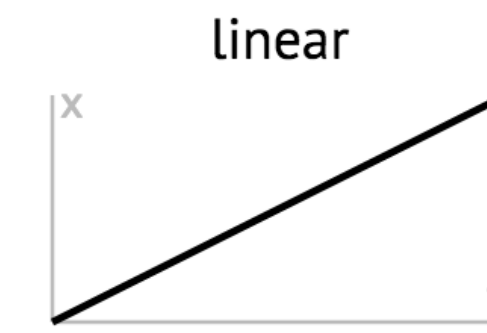
Make your own at [cubic-bezier.com](https://cubic-bezier.com)  
or edit with the Chrome/Firefox  
Dev Tools!

*Loading...*

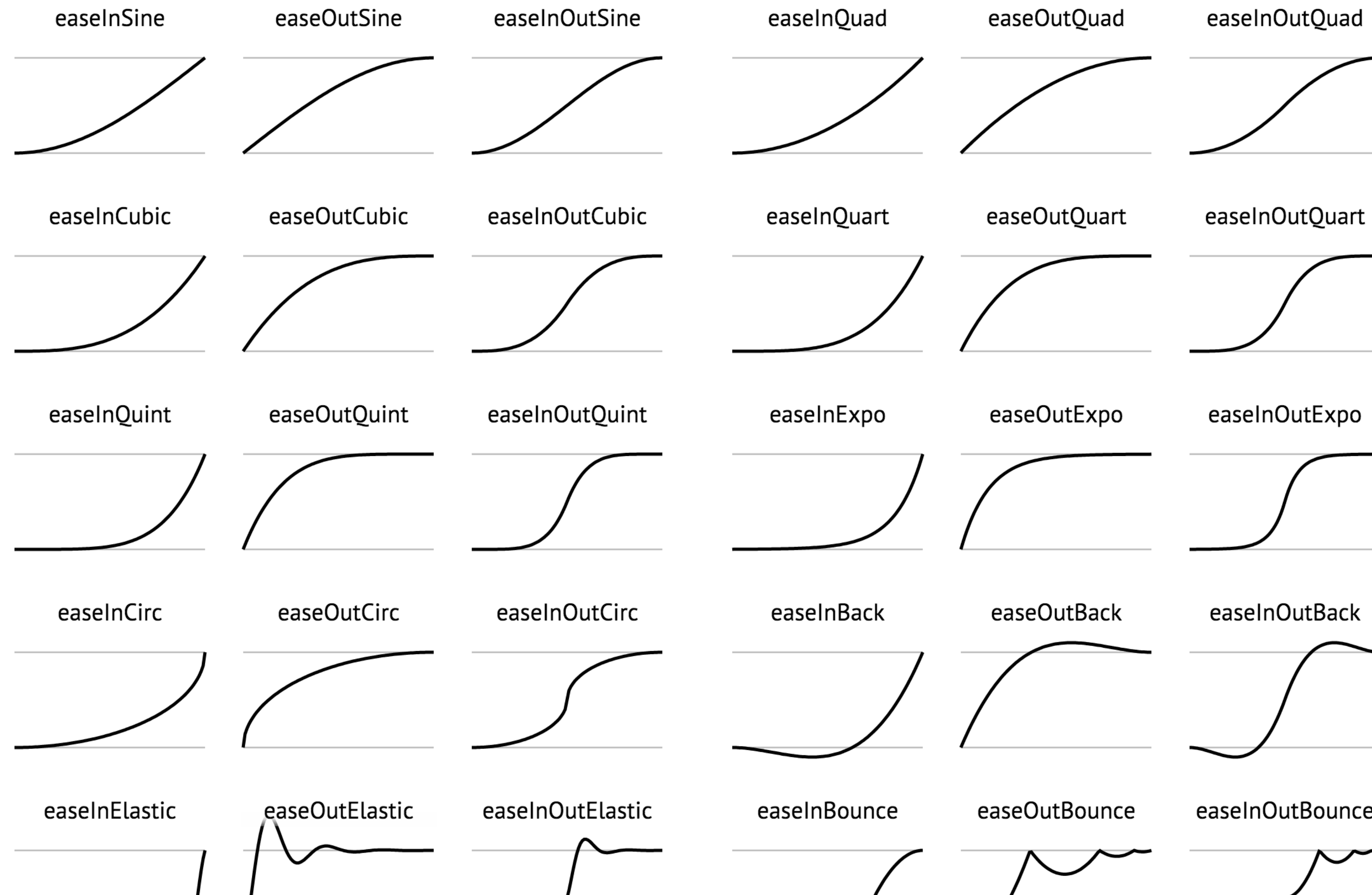
**Easing functions** specify the rate of change of a parameter over time.

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.

This page helps you choose the right easing function.



css+js



js

*exercise*  
Applying Physics to the Ball



Start coding! [cdpn.io/LVEdXg](https://cdpn.io/LVEdXg)

Experiment with different easing and durations to find the combination that feels “right” to you.

Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		36						
		37						
		38					4.1	
8	31	39					4.3	
9	35	40	7				4.4	
10	36	41	7.1		7.1		4.4.4	
11	37	42	8	27	8.3	8	40	42
TP	38	43		28				
	39	44		29				
	40	45						

Any thing but IE 9 and lower & Opera Mini  
[caniuse.com/#feat=css-transitions](http://caniuse.com/#feat=css-transitions)



# CSS Transitions rock because...

- **Single fire** If you only want something to happen once.
- **Granularity** If you would only animate one or two properties in a given state

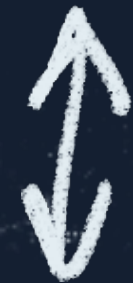


Questions?

# CSS Animations



```
.animated-thing {  
  animation: black-to-white 1s linear 1;  
}
```



```
@keyframes black-to-white {  
  0% { background: #000; }  
  100% { background: #fff; }  
}
```

*number of  
times to run*

```
.animated-thing {  
  animation:  
    $name  
    $duration  
    $timing-function (optional)  
    $animation-delay (optional)  
    $iteration-count;  
}
```

# Long form animation properties

- **animation-name:** The name of the keyframe block you want to use.
- **animation-duration:** How long the animations takes to go from 0% to 100%
- **animation-timing-function:** Like **transition-timing-function**
- **animation-delay:** The number of seconds to delay the animation instead of playing right away
- **animation-iteration-count:** The number of times you want to go from 0% to 100%; use **infinite** to never stop. Defaults to 1.

```
.animated-thing {  
  animation:  
    $name  
    $duration  
    $direction  
    $fill-mode  
    $play-state;  
}
```

# Long form advanced animation properties

- **animation-direction:** defaults to **normal** but can be set to **alternate**, **reverse**, **alternate-reverse**
- **animation-fill-mode:** Defaults to **backwards**, reverting to pre-animation. Can be set to **forwards**, assuming the end result of the animation, or **both** (a freakish combination?).
- **animation-play-state:** defaults to **running** but can be set to **paused**.



*exercise*  
**Wag the Cat**



Start coding! [cdpn.io/AfDBF](https://cdpn.io/AfDBF)

Play with Tuna's tail! Change the default advanced CSS animation properties.

Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors

# The `@keyframes` block



```
@keyframes black-to-white {
```

```
  0% {
```

```
    background: #000;
```

```
    color: #fff;
```

```
  }
```

```
  100% {
```

```
    background: #fff;
```

```
    color: #000;
```

```
  }
```

```
}
```

```
@keyframes black-to-white {  
  from {  
    background: #000;  
    color: #fff;  
  }  
  to {  
    background: #fff;  
    color: #000;  
  }  
}
```

```
@keyframes black-to-red {  
  0% { color: #000; }  
  50% { color: red; }  
  100% { color: #000; }  
}
```

```
@keyframes black-to-red {  
  0%, 100% { color: #000; }  
  50% { color: red; }  
}
```

```
@keyframes black-to-red {  
  0%, 100% {  
    animation-timing-function:  
    linear;  
  }  
  50% { color: red; }  
}
```

```
.animated-thing {  
  animation:  
    black-to-white 1s linear 1,  
    black-to-red 2s ease-out infinite 2s;  
}
```

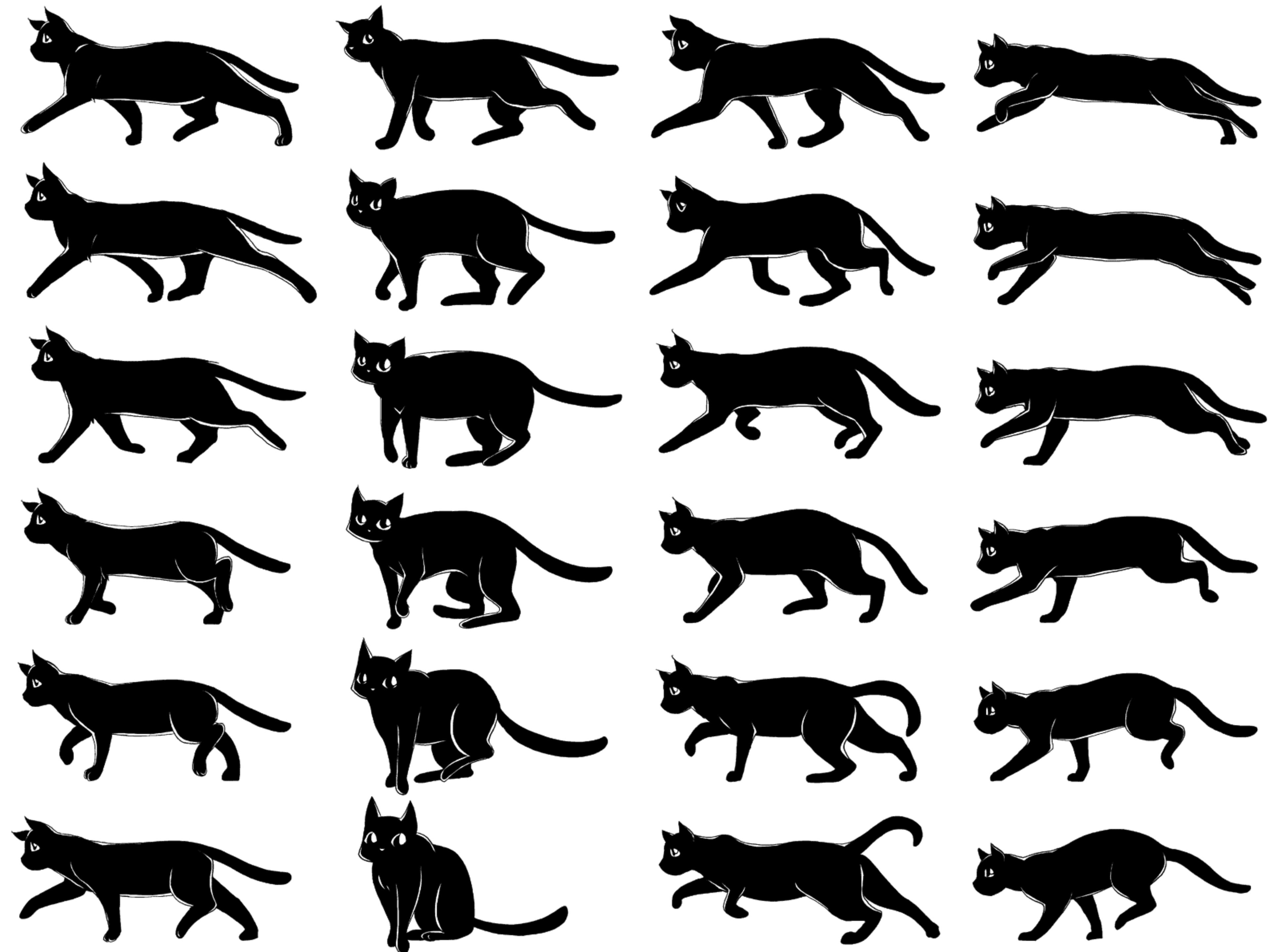


# Sprite animation with CSS





*used with the permission of Steven E. Gordon*



# Meet `steps()`

- **`steps(x)`** is a timing function...
- ...splits a block of keyframes into **`x`** equal steps, then hops between them.
- The [documentation](#) by Tab Atkins
- How it works, plus gotchas: [cdpn.io/zeFqy](https://cdpn.io/zeFqy)

*exercise*

# Making a Walk-Cycle



Start coding! [cdpn.io/cdqga](https://cdpn.io/cdqga)

The sprite: [stash.rachelnabors.com/animation-workshop/sprite\\_catwalk.png](https://stash.rachelnabors.com/animation-workshop/sprite_catwalk.png)

Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors



I

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		36						
		37						
		38					4.1	
8	31	39					4.3	
9	35	40	7				4.4	
10	36	41	7.1		7.1		4.4.4	
11	37	42	8	27	8.3	8	40	42
TP	38	43		28				
	39	44		29				
	40	45						

Not available on IE 9 and lower & Opera Mini.

Chrome, Safari and Android all require **-webkit-** at this time.

Consult [caniuse.com/#feat=css-animation](http://caniuse.com/#feat=css-animation)

# CSS Animations

- **Looping** Can loop infinitely
- **Self starting** Doesn't require trigger like **transition**
- **Repeating** You can set how many time it repeats
- **Alternating** Can alternate between the end state and start state
- **Grouping** Each animation can change a number of properties





Questions?

# Sequencing



```
.sequenced-thing {  
  animation:  
    sequence1 1s 2,  
    sequence2 3s 2s 1 forwards;  
}
```

↑  
the 2nd time  
measurement = delay

*exercise*  
**Sitting Tuna Down**



Start coding! [cdpn.io/Djcya](https://cdpn.io/Djcya)

Using `animation-delay`, start a sitting animation after the walking animation.

Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors

*But stacking  
isn't the only  
way, nor the  
most reliable.*



9:41 AM 100%

Search Private Done

X Rachel Nabors: Award-winning cartoonist turned digital storyteller. - Rachel Nabors, a...








TABLE OF CONTENTS

COURSES  
SPEAKING  
TRAINING  
WRITING  
ABOUT

STAY IN TOUCH!

MAILING LIST  
EMAIL





Loading...

**SPEAKING SCHEDULE**

*May 9-13*  
Beyond Tellerand, Düsseldorf & DOM in Motion workshop

*May 26-27*  
At the Front, Copenhagen & Motion: in Design and Development workshop

*Jun 24-26*  
Code, Melbourne & DOM in Motion workshop

*Jul 8*

*Jul 8*

**FEATURED WRITINGS**

Five Ways to Animate Responsibly Français *24 Ways*

The State of the Animation 2014  
*Smashing Mag*


Web Animation at Work *A List Apart*

Alice in Videoland: Designing an interactive HTML5 storybook  
*Adobe Inspire*


Flashless Animation

Flashless Animation *24 Ways*

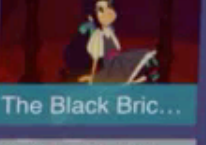
**ANIMATION PROJECTS**




The Black Bric...



Alice in Videola...



The Black Bric...



The Black Bric...

☁ Rachel Nabors's iPhone

---

TweenMax / GSAP Cheat codes list — Me...

---

Web Animations

---

Lazy Nezumi Pro - Mouse and Pen Smoot...

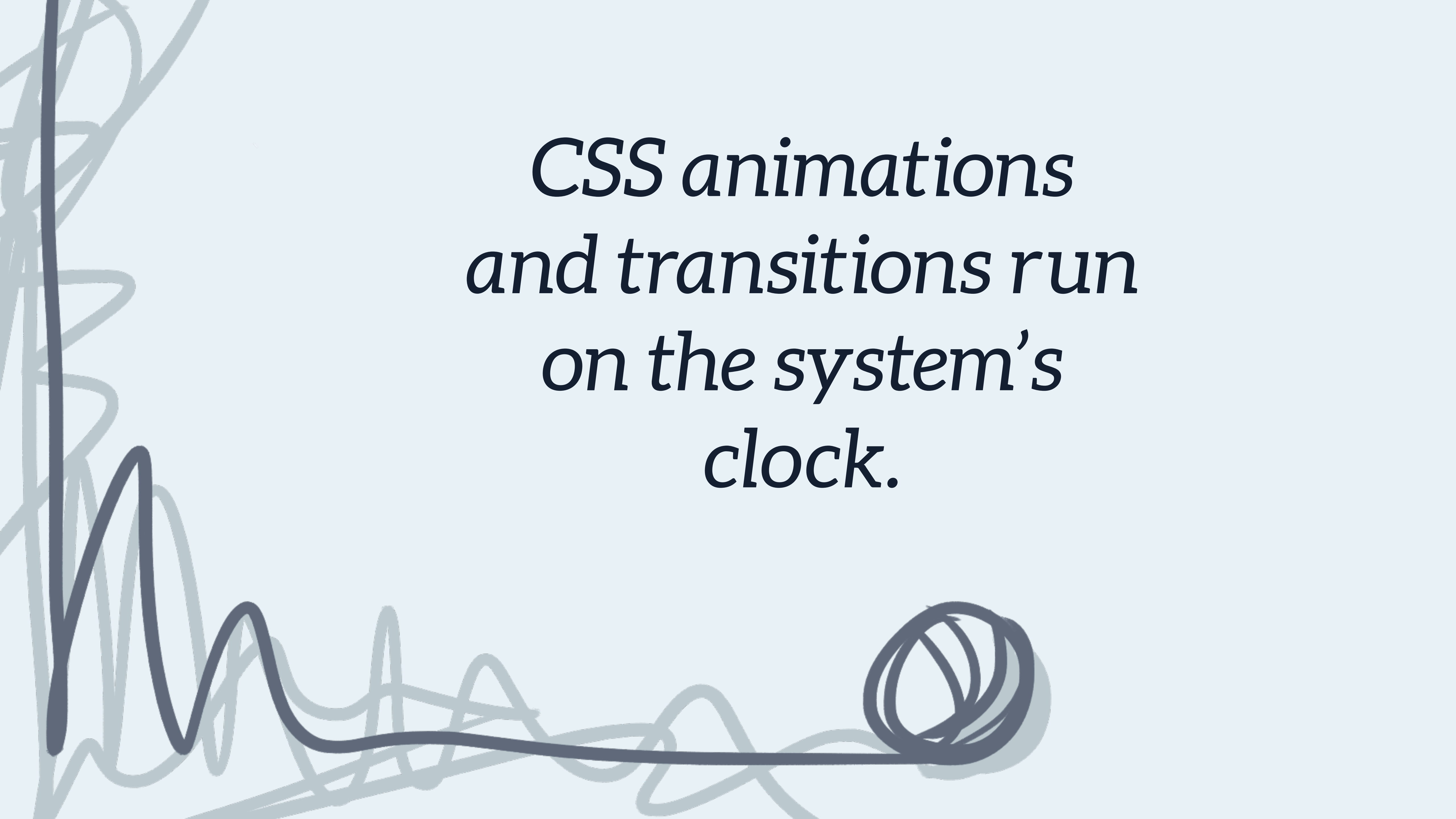
☁ Rachel's MacBook Pro

---


Rendering Performance — Web Fundame...

---

Elmoji



*CSS animations  
and transitions run  
on the system's  
clock.*



*We're gonna  
need a more  
robust technique.*



# Chaining Animations with Event Listeners



# Four must-have JavaScript event listeners

- **animationstart**
- **animationend**
- **animationiteration**
- **transitionend**

*They will fire for  
every animation/  
transition on the  
element.*



# Browser prefixes requires (of course)

- **webkitAnimationIteration** for Webkit
- **MSAnimationIteration** for IE 10
- Everyone else is **animationiteration**

*exercise*

## Sitting Tuna Down with event listeners



Start coding! [cdpn.io/pliqj](https://cdpn.io/pliqj)

Using `animationend`, add the `.sit` class after Tuna's done walking

Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors



Questions?

# Performance



BETA

# CSS TRIGGERS...

A GAME OF LAYOUT, PAINT, AND COMPOSITE.

[Read the blog post!](#)

Filter:



Layout

Paint

Composite

Not applicable


Does not trigger

	Change from default	Update value
align-content	<input checked="" type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>
align-items	<input checked="" type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>
align-self	<input checked="" type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>
backface-visibility	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/> <input checked="" type="radio"/>
background-attachment	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>
background-blend-mode	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>
background-clip	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>
background-color	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>
background-image	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>
background-origin	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/> <input type="radio"/>
background-position	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>
background-repeat	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>
background-size	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/>

## SELECT A PROPERTY

Choose a property from the list to get more details.



The background features a vertical line on the left side and a horizontal scribble at the bottom. The scribble consists of several overlapping, wavy lines in shades of blue and grey, extending across the width of the page.

*The two CSS  
properties that  
don't don't trigger  
layout or repaints:*

**opacity**

**transform**

# Handy alternatives

- **transform: scale()** instead of **width** or **height**
- **transform: translate()** instead of **position**
- **opacity** instead of **z-index** or **visibility: hidden**

*exercise*  
Walk the cat... again!



Start coding! [cdpn.io/QbwXry](https://cdpn.io/QbwXry)

Find a way to walk the cat without using expensive background positioning.

Keep up at:

Want to share? Save & tweet codepen link to @rachelnabors

**will-change**



# Hardware acceleration

AKA “kicking it to the GPU”:

```
.resource-sink {  
    transform: translateZ(0);  
}
```

Please don't do this.

*Hacks are not  
future-  
friendly.*




```
.in-view .animated-thing {  
  will-change: transform;  
}
```

```
.in-view .animated-thing:hover {  
  transform: translateY(5%);  
}
```



*Don't slap*  
**will-change**  
*on everything.*

*When we*  
**will-change**  
*everything, we*  
*optimize nothing.*





IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		36						
		37						
		38						
8	<sup>1</sup> 31	39					4.1	
9	<sup>1</sup> 35	40	7				4.3	
10	36	41	7.1				4.4	
		42					4.4.4	
11	37	42	8	27	7.1	8.3	8	40
TP	38	43		28				
	39	44		29				
	40	45						

[caniuse.com/#feat=will-change](http://caniuse.com/#feat=will-change)

# Chrome Dev Tools



# Chrome's FPS Meter

1. Open your Developer Tools.
2. Press the Escape Key to bring up a console window.
3. Choose the Rendering tab.
4. Check the Show FPS meter option.

# Paint Rectangles

To enable Chrome's paint rectangles:

1. Open your Developer Tools.
2. Press the Escape Key to bring up a console window.
3. Choose the Rendering tab.
4. Check show paint rectangles under Rendering

# The Timeline Tool

To use Chrome's timeline tool:

1. Open your Developer Tools.
2. Go to the Timeline tab.
3. Press the "record circle."
4. Do stuff.
5. Press the "record circle" again to stop.
6. Inspect!

*Best perf tip ever:  
Don't animate it if  
it isn't visible.*





Questions?

# the end

*Let's be animation friends!*

*@RachelNabors & RachelNabors.com WebAnimationWeekly.com*

