Eve Weinberg Follow
Present Futurist. Interaction Designer II @ frog design. Formerly ran a motion graphics studio. School: NYU's ITP & WashU. http://work.evejweinberg.com/work.htm
Jan 25, 2017 · 8 min read

Web Animation—A comprehensive overview

```
Contents:

1. BASIC INTRO – css and html

2. JAVASCRIPT LIBRARIES – start with GSAP and mo.js

3. PLUGINS – smaller libraries/toolkits to know about

4. WORKFLOW – a few resources for efficiency

5. INSPIRATION – people to follow, websites to subscribe to

6. DESIGN PRINCIPLES – read up on when/how to use animation

7. MORE ?!  – AE plugins and misc articles
```

. . .

This article is for people who are familiar with front end code and want a thorough overview of the existing methods, libraries, and resources for simple 2D web animation.

This is NOT covering advanced things like WebGL animations using GLSL shaders, physics libraries, robust game libraries like pixie.js, 3D libraries like Three.js, or huge platforms like p5.js. It's about DIY, writing code from scratch.

. . .

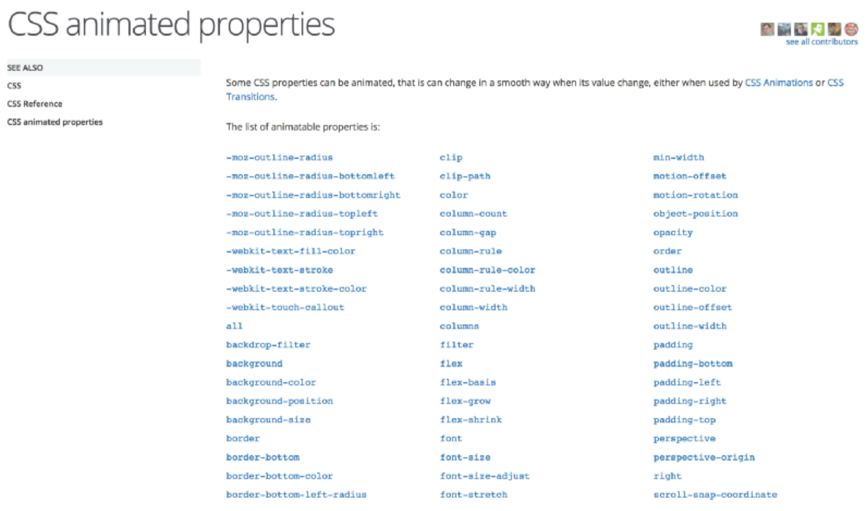
1 . Basic Introduction

There are three languages you need to know—**javascript**, **html**, and **css** (scss or sass are even better, but css will do). If you know only css and html, that's fine; javascript you can pick up easily as you need it on a per-library basis. In fact, here is an entire article on 'CSS vs Javascript' for animation. CSS and html can accomplish incredibly robust animations and interactions, so let's focus on that first.

METHOD 1: CSS ANIMATIONS

Note: you can do everything mentioned here in css, but if you start using this in your professional workflow, take a timeout and learn scss or sass because with those languages you can use variables which saves tons of time.

First, here is a list of all of the properties that can be animated using CSS.



CSS animated properties

see all contributors

SEE ALSO

CSS

CSS Reference

CSS animated properties

Some CSS properties can be animated, that is can change in a smooth way when its value change, either when used by CSS Animations or CSS Transitions.

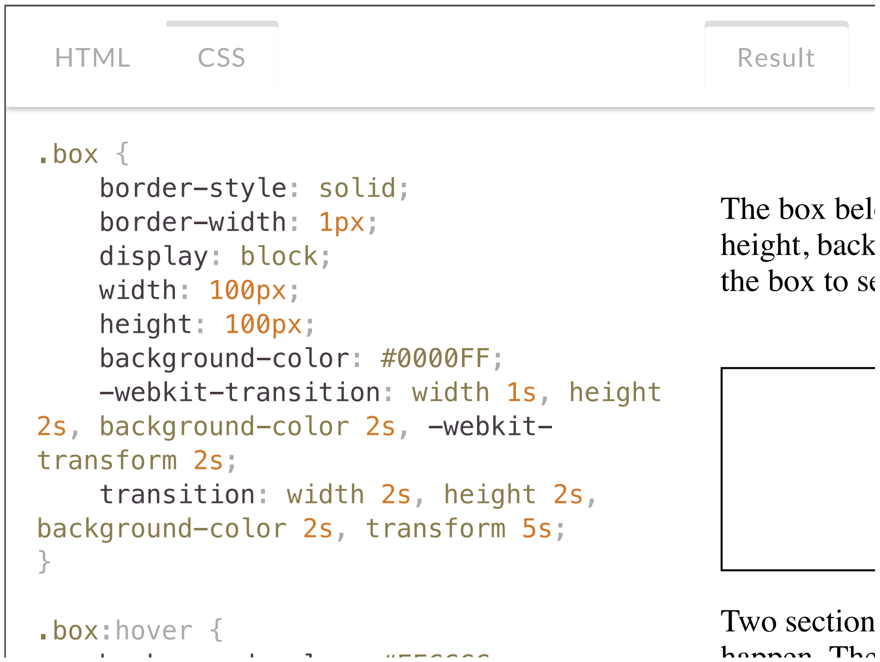
The list of animatable properties is:

-moz-outline-radius	clip	min-width
-moz-outline-radius-bottomleft	clip-path	motion-offset
-moz-outline-radius-bottomright	color	motion-rotation
-moz-outline-radius-topleft	column-count	object-position
-moz-outline-radius-topright	column-gap	opacity
-webkit-text-fill-color	column-rule	order
-webkit-text-stroke	column-rule-color	outline
-webkit-text-stroke-color	column-rule-width	outline-color
-webkit-touch-callout	column-width	outline-offset
all	columns	outline-width
backdrop-filter	filter	padding
background	flex	padding-bottom
background-color	flex-basis	padding-left
background-position	flex-grow	padding-right
background-size	flex-shrink	padding-top
border	font	perspective
border-bottom	font-size	perspective-origin
border-bottom-color	font-size-adjust	right
border-bottom-left-radius	font-stretch	scroll-snap-coordinate

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties

For almost any of these properties (there are always caveats and edge cases), we can animate them in CSS with one of two methods—transitions or keyframes.

Here is an example of using transition (look at the CSS tab):

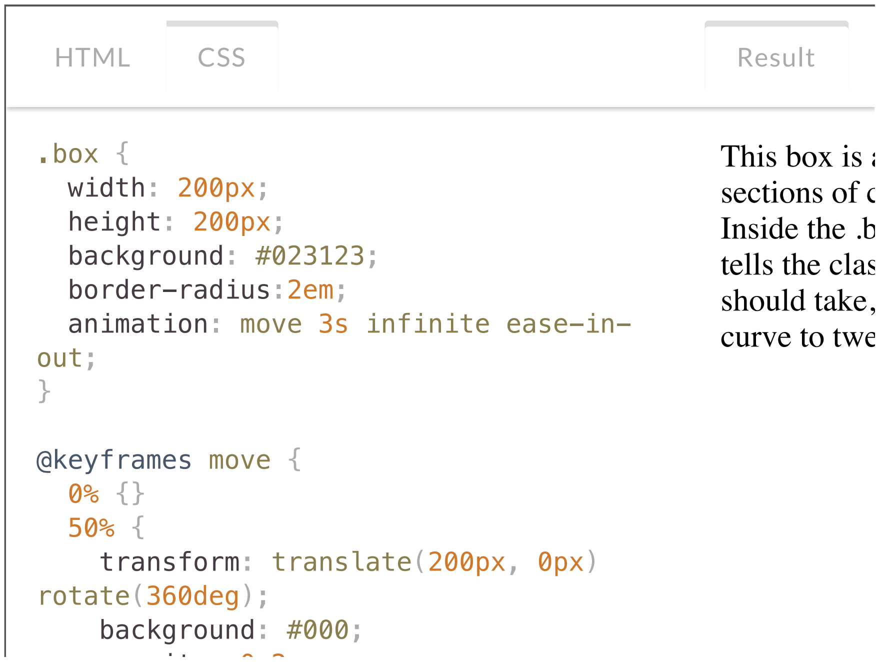


```
.box {
    border-style: solid;
    border-width: 1px;
    display: block;
    width: 100px;
    height: 100px;
    background-color: #0000FF;
    -webkit-transition: width 1s, height
2s, background-color 2s, -webkit-
transform 2s;
    transition: width 2s, height 2s,
background-color 2s, transform 5s;
}

.box:hover {
```

The box bel
height, back
the box to se

Two section

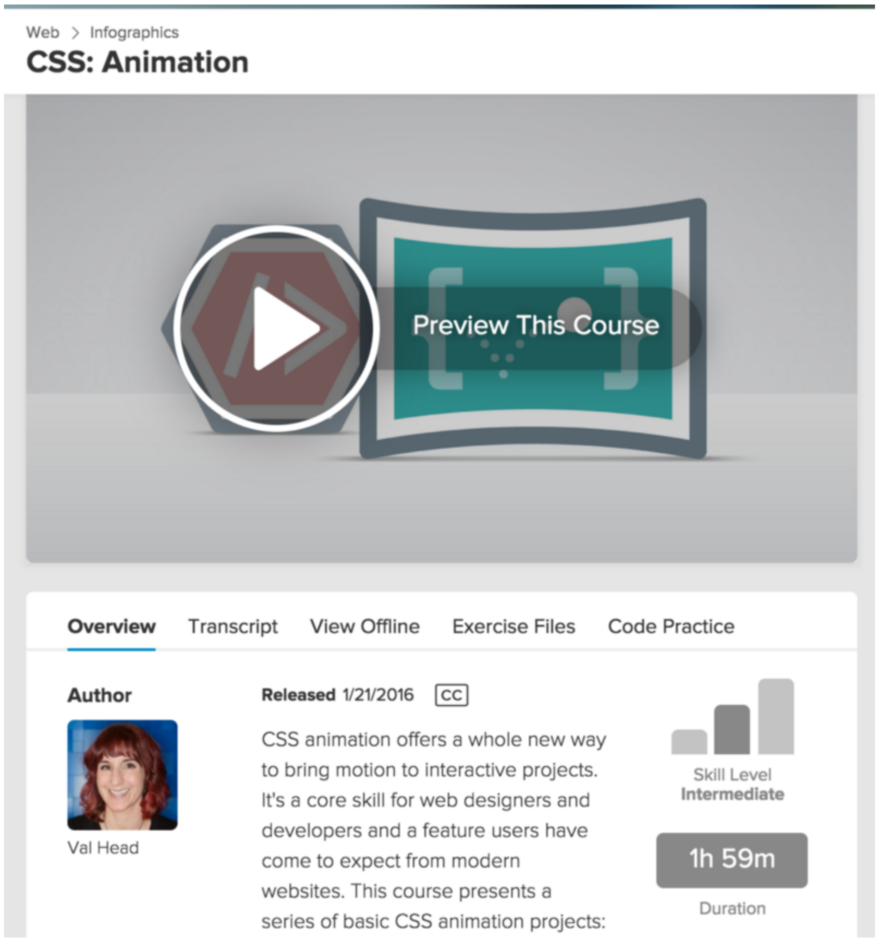
Here is an example of using keyframes (look at the CSS tab):



```
.box {
  width: 200px;
  height: 200px;
  background: #023123;
  border-radius:2em;
  animation: move 3s infinite ease-in-
out;
}

@keyframes move {
  0% {}
  50% {
    transform: translate(200px, 0px)
rotate(360deg);
    background: #000;
```

This box is
sections of
Inside the .b
tells the clas
should take,
curve to twe

If you're intrigued by these two examples, the best way to learn CSS animation is from Val Head on Lynda:


Web > Infographics

CSS: Animation

Author

Val Head

Released 1/21/2016 CC

CSS animation offers a whole new way to bring motion to interactive projects. It's a core skill for web designers and developers and a feature users have come to expect from modern websites. This course presents a series of basic CSS animation projects:

1h 59m

https://www.lynda.com/CSS-tutorials/CSS-Animation/439683-2.html

Another great reference is this Codrops page , where they cover parameters, functions, datatypes, and more!

CSS Reference

An extensive CSS reference with all the important properties and info to learn CSS from the basics

Pseudo-class	Property	Function
Data Type	Concept	@rule

https://tympanus.net/codrops/css_reference/

Or if you learn better by clicking around existing examples, here are 50 examples of css parameters (some are animated, some are just properties that you might not have realized could be set in css)

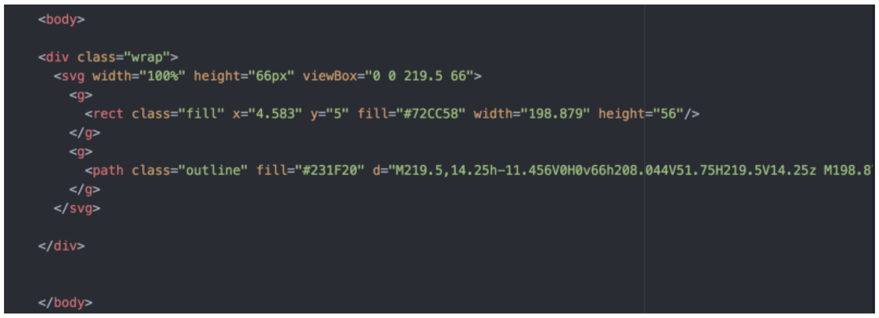
If you're old school, here's a book!

And if you're ready for sass, here's their website. I suggest using Lynda again to learn this in detail.

. . .

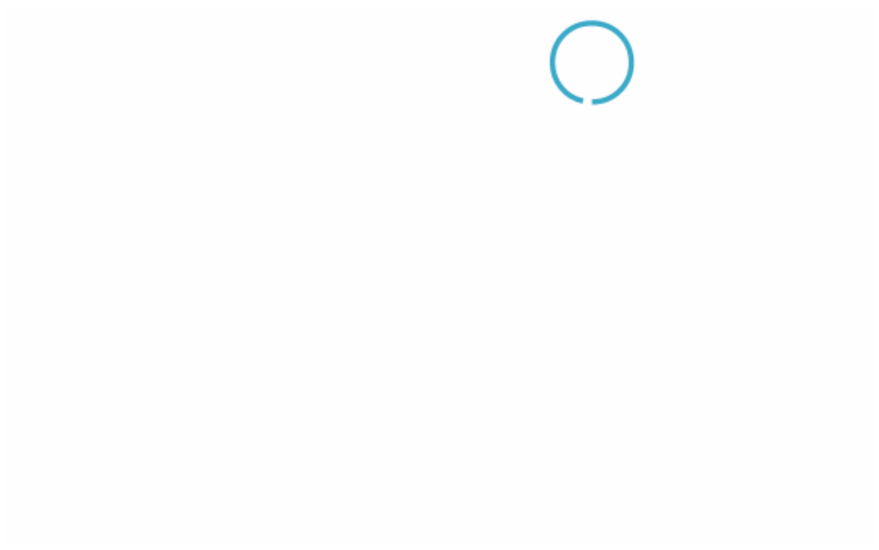
METHOD 2: SVG

SVGs are scalable vector graphics and they are easily to manipulate with css or simple javascript. The workflow for SVG is to build your file in illustrator, or download an SVG from somewhere like the noun project, then copy paste the generated code into your html like this:



```
<body>

<div class="wrap">
  <svg width="100%" height="66px" viewBox="0 0 219.5 66">
    <g>
      <rect class="fill" x="4.583" y="5" fill="#72CC58" width="198.879" height="56"/>
    </g>
    <g>
      <path class="outline" fill="#231F20" d="M219.5,14.25h-11.456V0H0v66h208.044V51.75H219.5V14.25z M198.8
    </g>
  </svg>

</div>


</body>
```

The most common animations on svg elements is to animate the stroke. Like this!

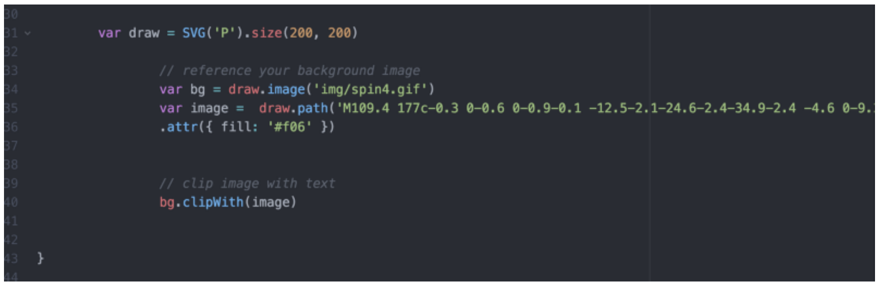


For that you would use the CSS parameters ‘stroke-dasharray’ and ‘stroke-dashoffset’ like this:

```
.outline {
  stroke-dasharray: 0;
  stroke-dashoffset: 0;
  animation: offsetStrokes 4s .5s ease-in-out  forwards
infinite;


}

@keyframes offsetStrokes {
  to{
   stroke-dasharray: 505;
   stroke-dashoffset: 405;
  }
}
```

Here’s a great overview of how the ‘stroke-dasharray’ and ‘stroke-dashoffset’ work. If you start using this technique a lot, it might be time to look at a javascript library like vivus.

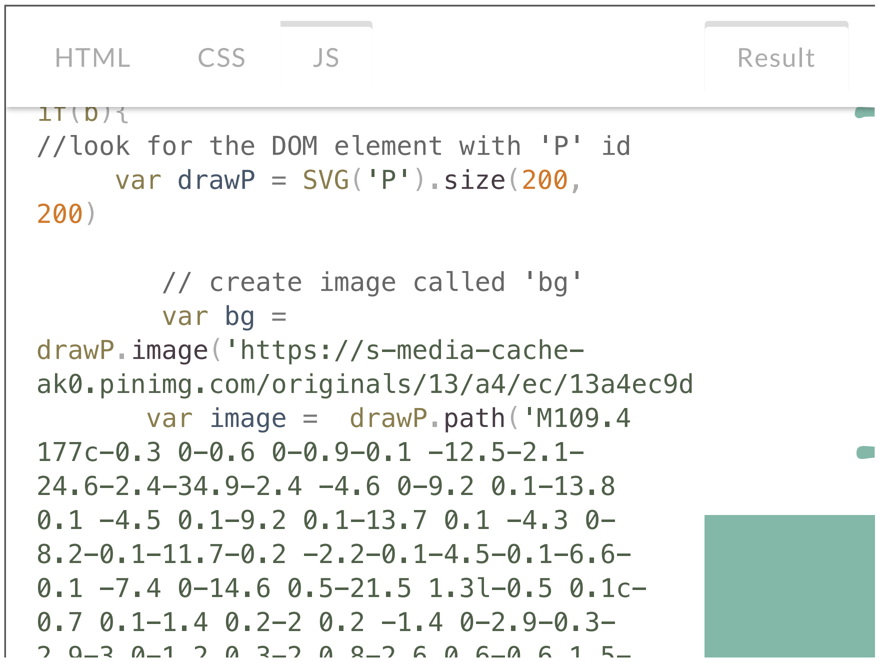
For more interesting and complex SVG animation ideas, look at the library http://svgjs.com/ Here is a code snippet of how to instantiate an svg with that library:



```
        var draw = SVG('P').size(200, 200)

                // reference your background image
                var bg = draw.image('img/spin4.gif')
                var image =  draw.path('M109.4 177c-0.3 0-0.6 0-0.9-0.1 -12.5-2.1-24.6-2.4-34.9-2.4 -4.6 0-9.
                .attr({ fill: '#f06' })


                // clip image with text
                bg.clipWith(image)


}
```

Notice that in both html and javascript we use the ‘path’ attribute.

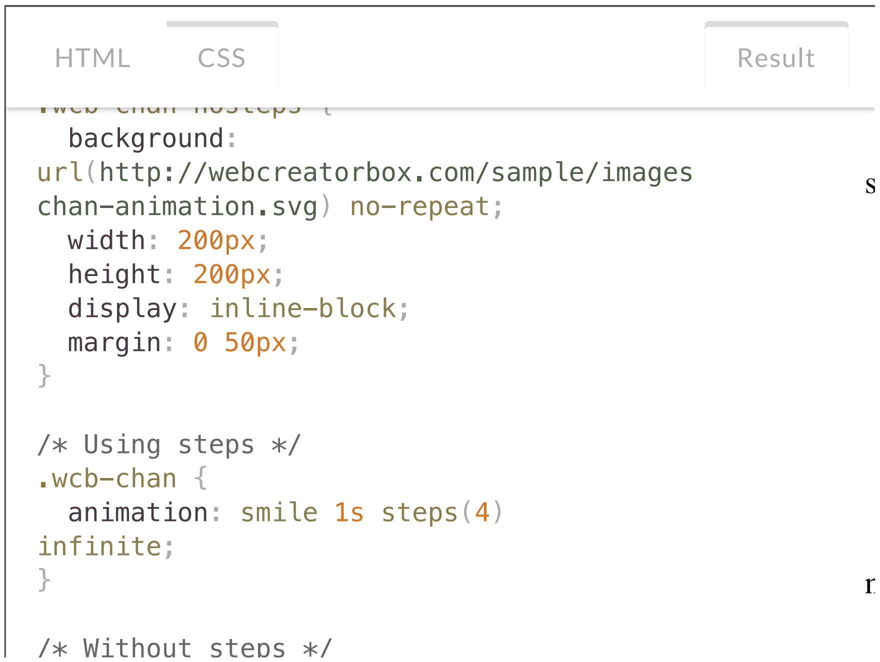
Svg.js is a very robust library and I highly recommend it. Here is a small example I made showing another feature it offers—masking type or an svg path with an image:



```
if(b){
//look for the DOM element with 'P' id
     var drawP = SVG('P').size(200,
200)

        // create image called 'bg'
        var bg =
drawP.image('https://s-media-cache-
ak0.pinimg.com/originals/13/a4/ec/13a4ec9d
       var image =  drawP.path('M109.4
177c-0.3 0-0.6 0-0.9-0.1 -12.5-2.1-
24.6-2.4-34.9-2.4 -4.6 0-9.2 0.1-13.8
0.1 -4.5 0.1-9.2 0.1-13.7 0.1 -4.3 0-
8.2-0.1-11.7-0.2 -2.2-0.1-4.5-0.1-6.6-
0.1 -7.4 0-14.6 0.5-21.5 1.3l-0.5 0.1c-
0.7 0.1-1.4 0.2-2 0.2 -1.4 0-2.9-0.3-
```

METHOD 3: SPRITE SHEETS

If you’re a traditional animator and just want different ways to get your animations on the web, try sprite sheets. This is a process where you export the frames of your animation, then compile them all onto one large file, equally spaced. Then, in css you tell the DOM element how far to scoot around the document and at what frame rate:



```
  background:
url(http://webcreatorbox.com/sample/images
chan-animation.svg) no-repeat;
  width: 200px;
  height: 200px;
  display: inline-block;
  margin: 0 50px;
}

/* Using steps */
.wcb-chan {
  animation: smile 1s steps(4)
infinite;
}

/* Without steps */
```

Here's a whole medium article on sprite sheets, if you want to dig into it. And if you're dead-set on animating and having the code generated for you, there are plugins for AfterEffects (one listed in section 7) or you can try the desktop application Tumult Hype 3, although be warned—the code generated by these tools has been referred to as 'garbage', 'heavy', 'messy', and 'wtf' by many developers I have worked with. This is more of a prototyping route and not for production.

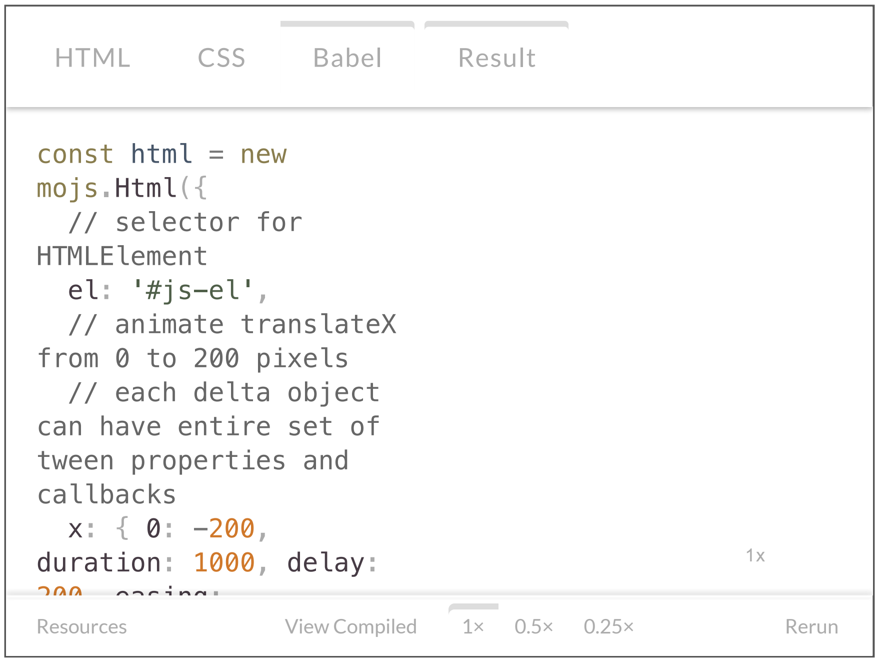
. . .

2. Javascript Libraries

For more complex animations and interactions, you'll want javascript. There are two libraries I would start with (and a thousands more out there). I recommend greensock for manipulating DOM elements, and mo.js for making animations from scratch in javascript.

Greensock (GSAP) can be used in conjunction with anything, you will not regret taking the time to learn this. I even use it to tween my camera moves in Three.js. Design firms like the ones in the Awwwards and dribble communities use GSAP to manually create motion design chains where CSS animations fall short. The best way to get to know this library is to look at the documentation and start with this free online quick course.

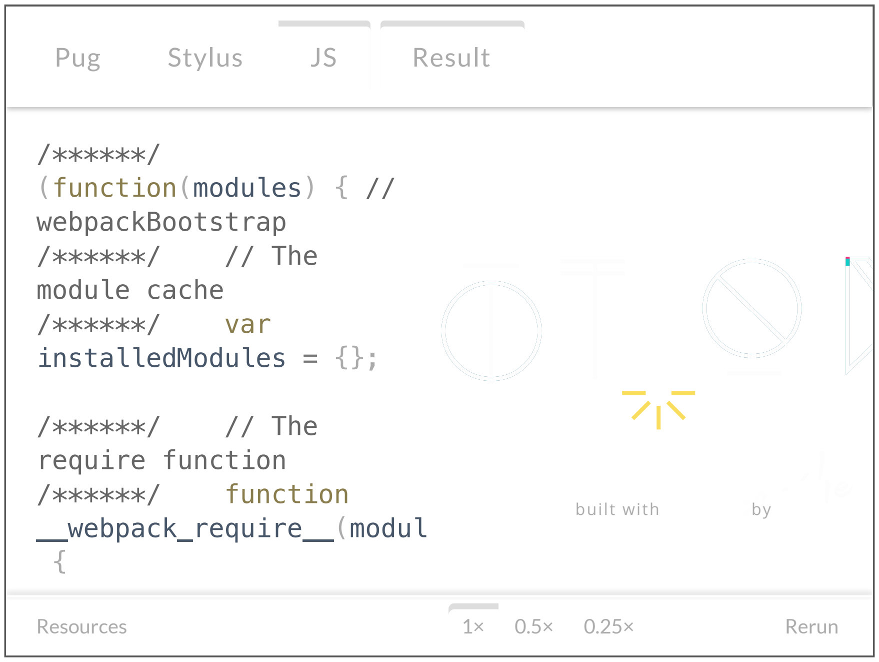
Mo.js is more akin to Flash. Keeping with the theme of using boxes, here is how to animate a box in mo.js:



```
const html = new
mojs.Html({
  // selector for
HTMLElement
  el: '#js-el',
  // animate translateX
from 0 to 200 pixels
  // each delta object
can have entire set of
tween properties and
callbacks
  x: { 0: -200,
duration: 1000, delay:
200, easing:
```

http://codepen.io/sol0mka/pen/ZpvLzX/

And here's what happens when you go pro and really understand mo.js:



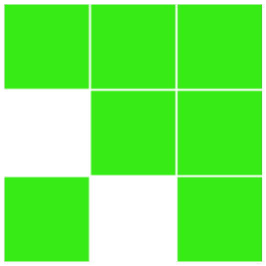
```
/******/
(function(modules) { //
webpackBootstrap
/******/    // The
module cache
/******/    var
installedModules = {};

/******/    // The
require function
/******/    function
__webpack_require__(modul
 {
```

http://codepen.io/sol0mka/pen/ogOYJj

Read here what the hacker news community has to say about it. And if you like this, dig into the documentation. It's well designed :)

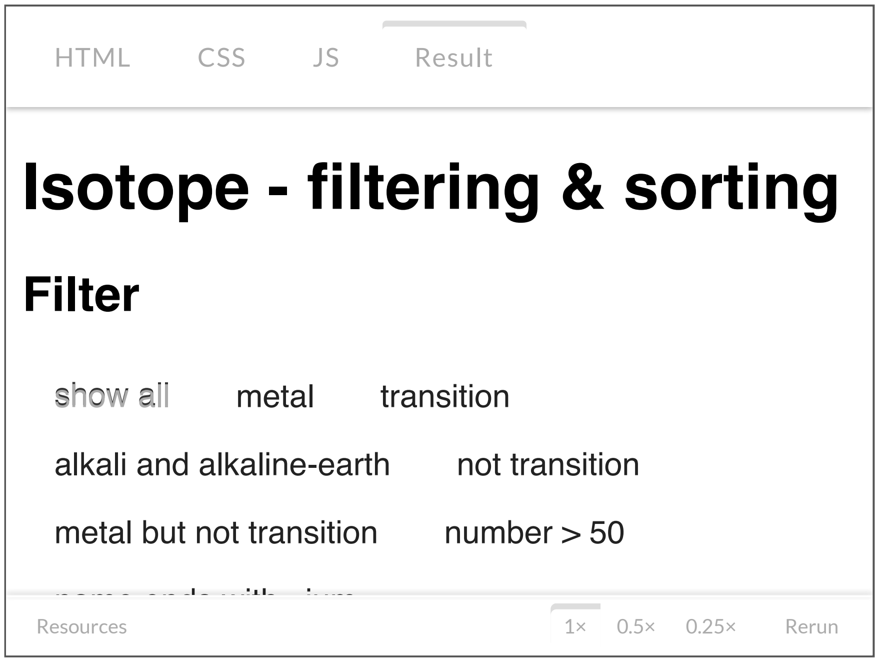
Other popular libraries (that I have had very little experience with) are paper.js, velocity.js, and two.js. Then of course there's jQuery which is very easy to use, with it's .animate() method and using it with the add-on library called jQueryUI is very powerful.

. . .



3. More plugins

Plugins, toolkits, toolbelts, frameworks. . .whatever you prefer to call them, I wanted to mention a few more resources for finding smaller packaged animations aside from the two libraries above. These plugins usually serve one purpose per plugin. David Desandro makes some great ones, such as isotope:



Isotope - filtering & sorting

Filter

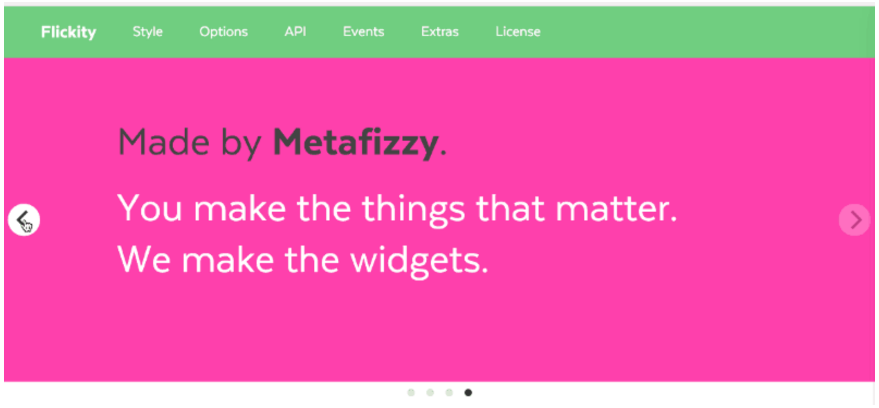
show all metal transition

alkali and alkaline-earth not transition

metal but not transition number > 50

http://codepen.io/evejweinberg/pen/ggXMrJ

…and flickity:

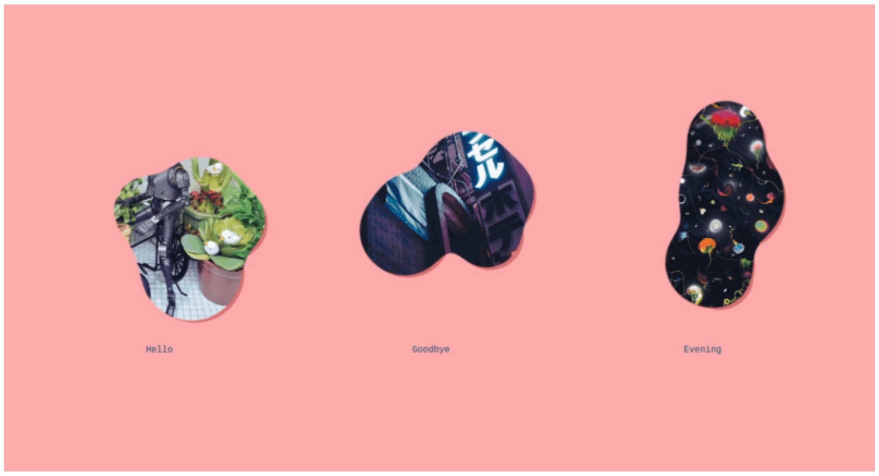


Made by Metafizzy.

You make the things that matter.
We make the widgets.

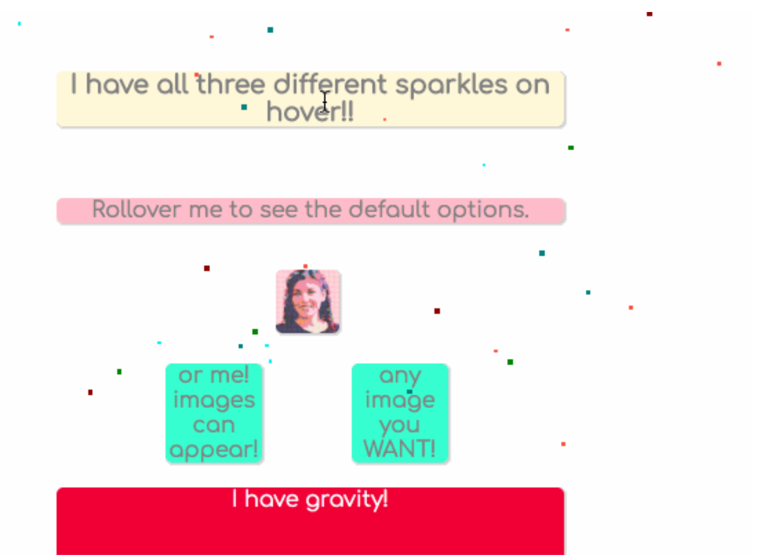
http://flickity.metafizzy.co/

Francis Tseng made this one with svg and javascript so you can make these cool blobs on hover:



https://github.com/frnsys/svg_blobs

I made my own jQuery plugin recently. It makes particles on hover for any object you choose. You can use that if you like! It's called SparkleHover. Here is a gif of it in action:


I have all three different sparkles on hover!!
Rollover me to see the default options.
or me! images can appear!
any image you WANT!
I have gravity!

https://github.com/evejweinberg/sparkleHover

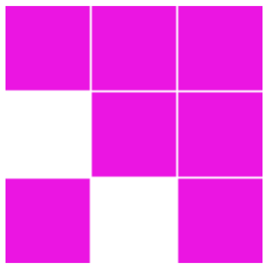
And then a quick search on github for "javascript animations", produces many results. Tip—sort by stars or forks.

Vivus—for drawing on the lines of svg files

Granim- for animated gradients

TweenJS

· · ·



4. Resources for better workflow and efficiency

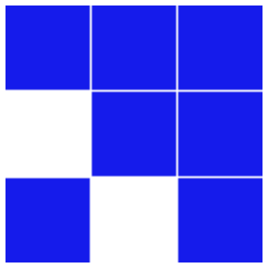
1. Worried about runtime performance? Use this chart

2. Worried about cross-browser compatibility? DON'T! Just compile with autoprefixer this when you're ready to deploy.

3. Don't want to write bezier curves from scratch? Get the code to all of them here

4. Need to clean up your messy svg code before adding to you site? Do it here

. . .



5. Inspiration!

WHO TO FOLLOW:

1. Val Head
2. Rachel Nabors
3. Sarah Drasner

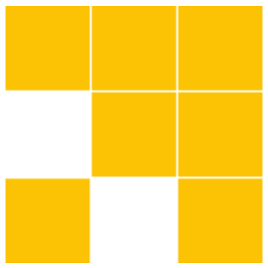
4. Sara Soueidan
5. Ana Tudor
6. Rachel Smith

⇧ ⇧ ⇧*Who runs the world? Girls!!* ⇧ ⇧ ⇧

WEBSITES FOR DAILY INSPIRATION AND LEARNING:

1. Codrops—a blog with an overwhelming amount of resources for learning and implementing code
2. Codepen—a playground where you can write code, play with other people's code, fork things, save things, share things.
3. CSS Tricks—awesome resource, awesome blog for learning and getting ideas that you can implement.
4. Awwwards—super advances and inspiring websites
5. FWA Awards—even MORE inspiring websites, on average a bit more complex than awwwards

. . .



. . .

6. Philosophy + Design Principles

Again, Val Head to the rescue. Here is her book, "Designing Interface Animation".

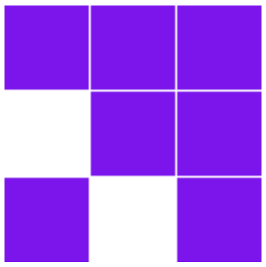
And here is an article she wrote: http://alistapart.com/article/designing-interface-animation

And Sarah Drasner gives amazing lectures, here are her slides from a very in-depth one about motion workflow, branding, and concept: http://slides.com/sdrasner/style-guide-anim#/

Then check out the philosophies and principles of these companies:

1. Google Material design principles on motion
2. Salesforce principles of motion UI

· · ·



· · ·

7. More?! Here are some more misc articles

Not sure if this article helped you? Here is another article, just like it. Sometimes it's good to hear the same material in two different people's voices: https://desandro.github.io/motion-emotion/

And another article: http://www.hongkiat.com/blog/ux-motion-designer-freebies/

For After Effect lovers, try this plugin for AE called bodymovin. It's free and easy to install from the Adobe app store. I installed and demoed it and it seemed very easy to use. Please let me know if you use it! Here are some demos to give you some ideas. And here is a blog post by the creator.

· · ·

Thank you for reading!!

Eve Weinberg

Self Portrait created using photogrammetry // www.NeverOddorEven.tv