

Animation on the Web

Vivian Chen

Before I started

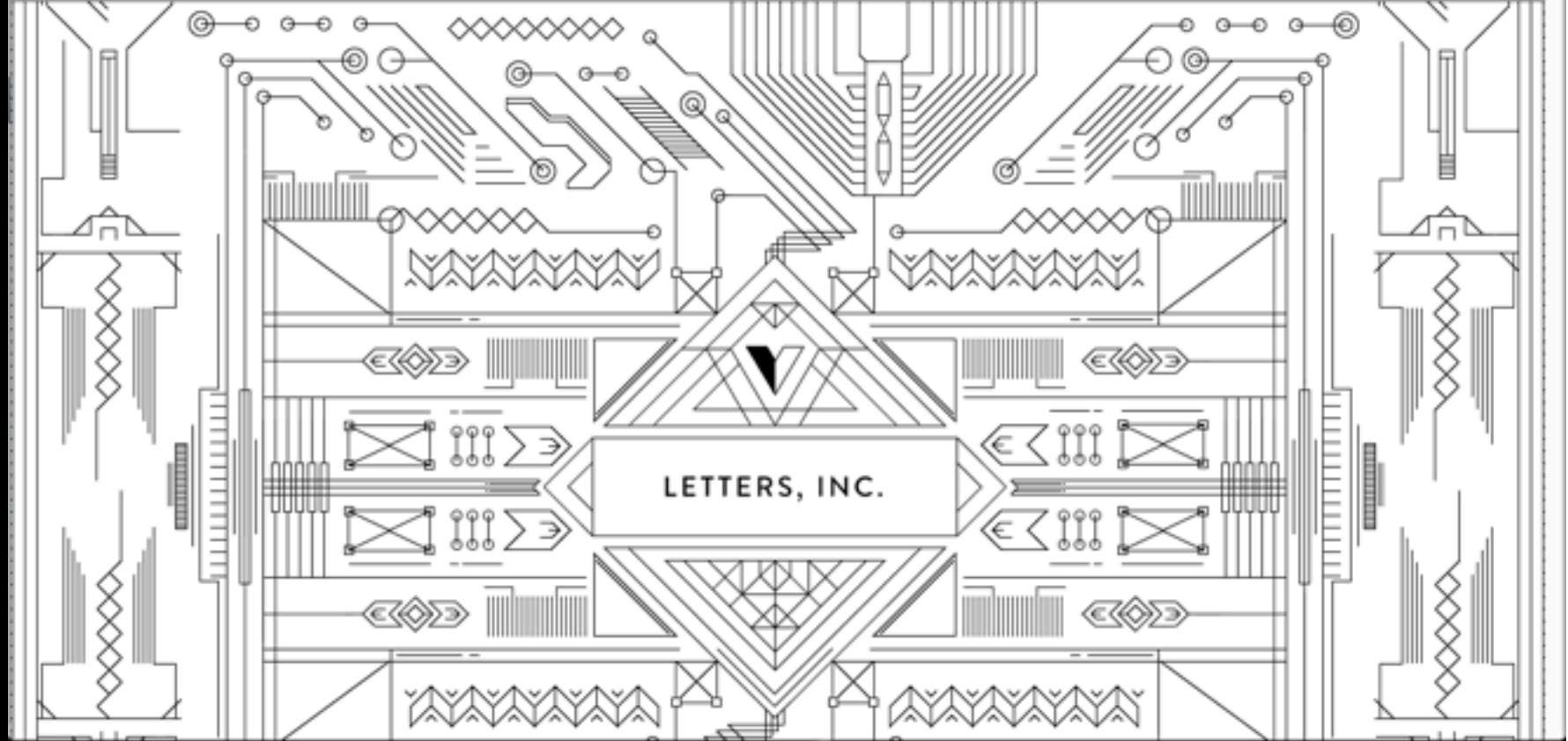
I've been really intrigued by web technology for what and how the content can be displayed, and the two main factors are animation and interaction.

Before taking this class, I have already been trying to figure out what are the different types of animation/interaction out there and seeking for ways to actually make them.

As a result, I decided to create some animation on the web for this project.

Research

Precedents



Like Share

u* The team has been blown away by your
and talk about our site! We never expected
00 hits and to say thank you we've decided
h our game this week with a \$10,000 prize!
ne developers, here's how we built the site.
ODG

START
Scroll to explore



<http://letters-inc.jp/>

<http://gardenstudio.com.br/>

<http://www.ohdeergames.com>

ScrollMagic

I wanted to make the animation interactive.
The first thing that came to my mind if Parallax
Scrolling effect and other effect controlled by
scrolling.

ScrollMagic is the solution here.

Yet, soon I realized that I have to make the
animation first, while ScrollMagic provides just
triggers and control of the animation.

I decided to come back later when I have my
animation made.



CDN Bower NPM Download

```
<script src="//cdnjs.cloudflare.com/ajax/libs/ScrollMagic/2.0.1/ScrollMagic
<script src="//cdnjs.cloudflare.com/ajax/libs/ScrollMagic/2.0.1/plugins/add
```

<http://janpaepke.github.io/ScrollMagic/>

Canvas

Animation with canvas in html5 was the technology that really surprised me before I dug into web development. The most interesting piece is the "Tearable Cloth". The simulation of gravity and texture is impressive.

The excitement that html5 canvas has brought me led me to do research on Paper.js, raphael.js and Processing.js, which are the most popular-used Javascript library for manipulating elements on canvas. Although the way of drawing graph is very similar to Processing that was taught in the class, I still found it too complicated.

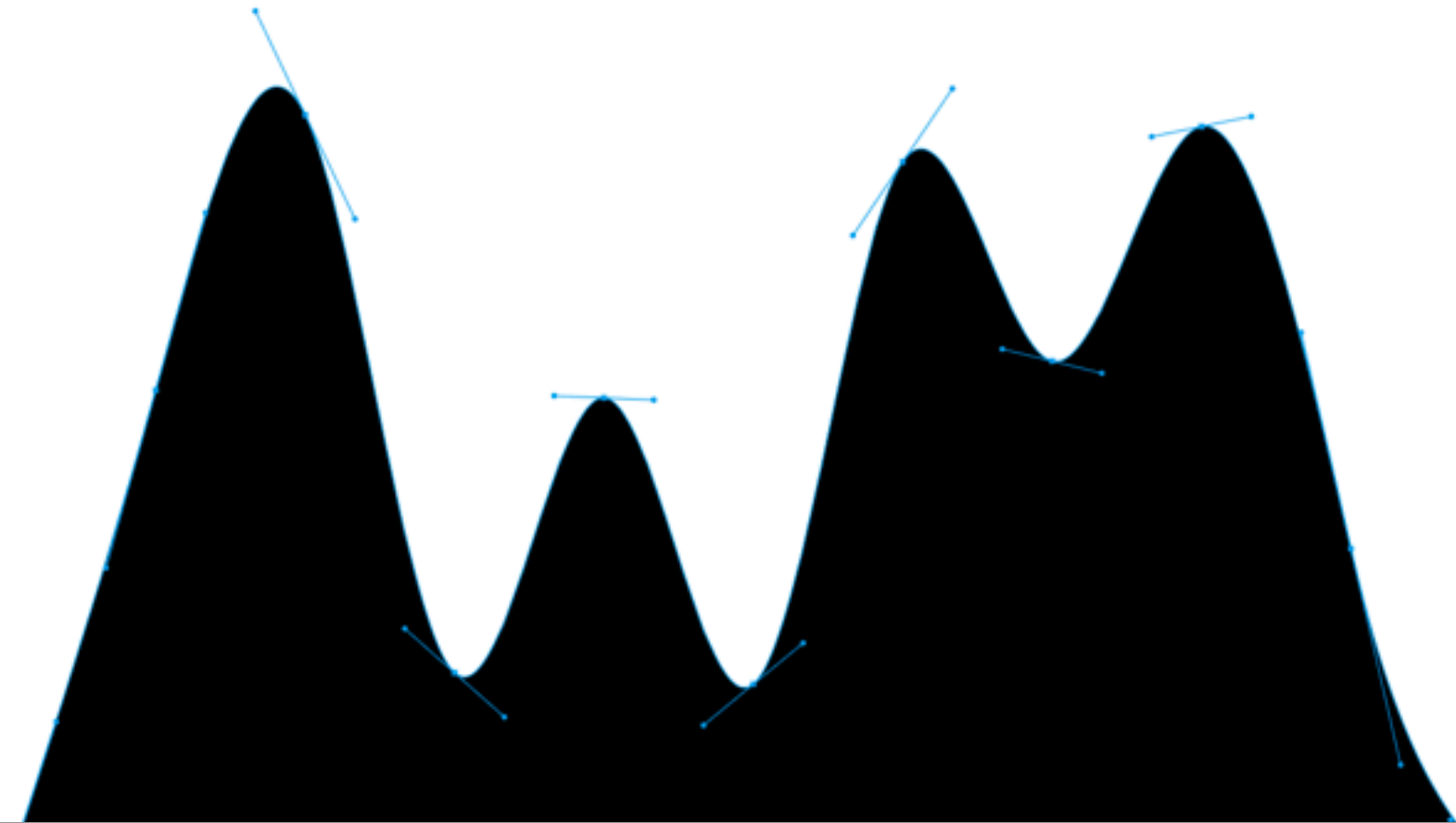
Hopefully I will get back to it soon when I accumulate enough knowledge about html5, canvas and Javascript.

<http://codepen.io/gordonnl/full/byouf/>

<http://paperjs.org/>

Paper.js

About
Features
Examples
Showcase
Tutorials
Reference
Sketch
Download
Donation
License
Mailing List
Follow on Twitter
Watch on Github



Raphaël—JavaScript Library

? What is it?

Raphaël is a small JavaScript library that should simplify your work with vector graphics on the web. If you want to create your own specific chart or image crop and rotate widget, for example, you can achieve it simply and easily with this library.

Raphaël ['ræfɛrɛl] uses the SVG W3C Recommendation and VML as a base for creating graphics. This means every graphical object you create is also a DOM object, so you can attach JavaScript event handlers or modify them later. Raphaël's goal is to provide an adapter that will make drawing vector art compatible cross-browser and easy.

Raphaël currently supports Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ and Internet Explorer 6.0+.

Download v. 2.1.2 (91 Kb)

Our recommendation is to GZIP it. It will help to reduce file size. You can download [uncompressed source \(299 Kb\)](#) as well.

Documentation

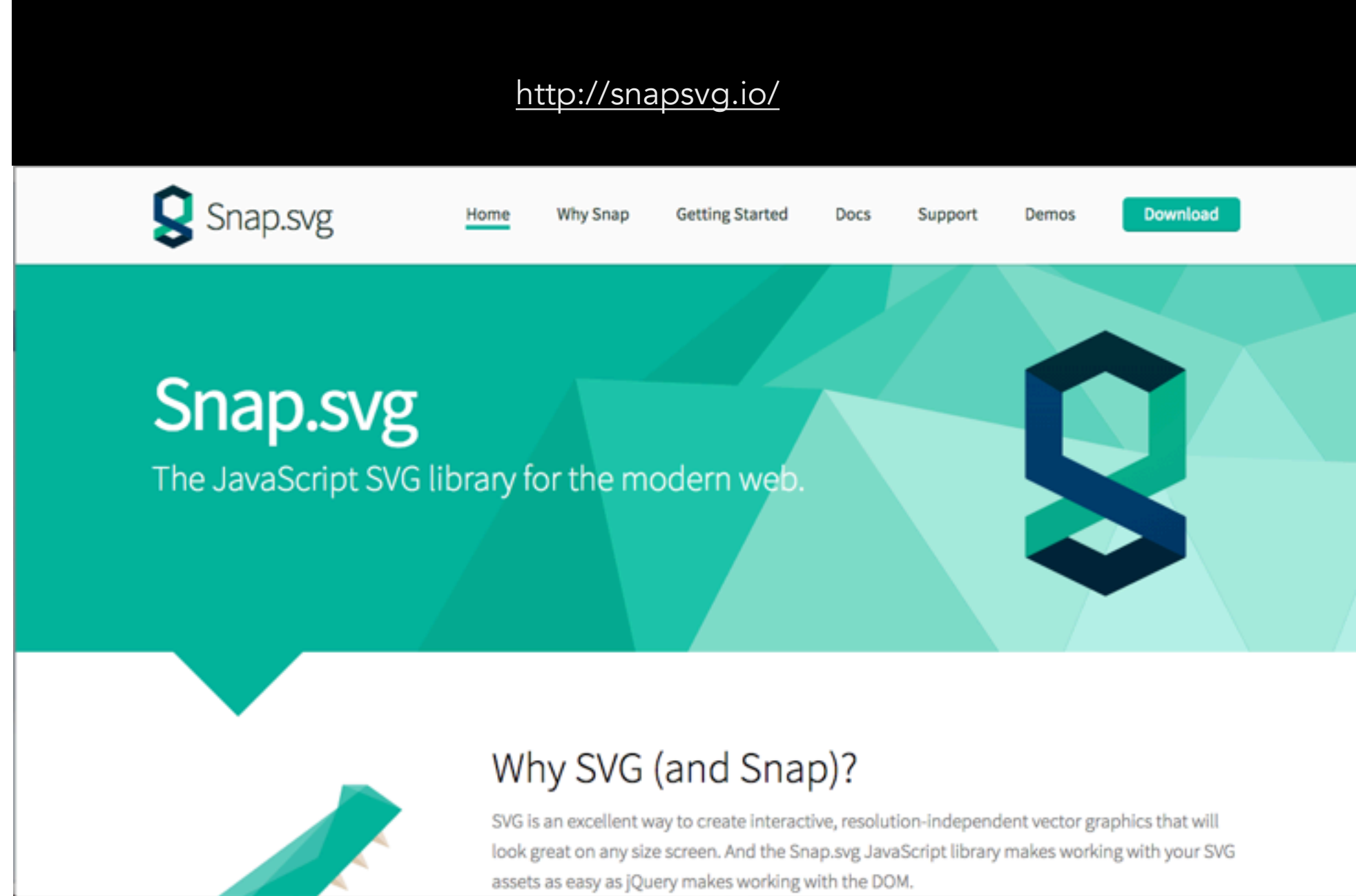
Discussion Group

Twitter

Snap.svg

Snap.svg is a Javascript library that animates and draws SVGs. In the beginning I was really interested in using it, but the more information I read about SVGs, the more I thought that it's too complicated for this project, since the time was tight.

Apart from that, SVG animation are light and flexible so I might try this later.



Kathleen Spurr

<http://codepen.io/ghepting/pen/xnezB>

jQuery

jQuery might be the first thing that comes to mind when animation on the web is mentioned. However, I know that codes of jQuery are fairly basic, which is ideal for small animation and more about interaction. It will cost me a lot of time if I want to use it to achieve the ideal animation for me.

CSS 3 animation

CSS3 animation is great and fast, but it will have browser compatibility issues.

I chose not to use it also because I'm not familiar with CSS3, and I know it's going to be hard to maintain.

The image is a composite of two screenshots. The top screenshot shows the jQuery API documentation for the `.animate()` method. It features the jQuery logo at the top left, a navigation bar with links like 'Download', 'API Documentation', 'Blog', 'Plugins', and 'Browser Support', and a search bar. On the left is a sidebar with a tree view of API categories. The main content area is titled `.animate()` and includes a description: 'Perform a custom animation of a set of CSS properties.' Below this, it lists parameters: `properties` (Type: PlainObject), `duration` (default: 400, Type: Number or String), and `easing` (default: 'swing', Type: String). The bottom screenshot shows the 'CSS Animate Beta' web application. It has a central canvas with a blue square labeled 'Drag' and a dashed crosshair. To the right is a 'Keyframes' panel with settings for 'Duration' (4 sec), 'Repeat' (1 times), 'Timing function' (linear), and 'Rotation point'. At the bottom is a 'Timeline' panel.

<http://api.jquery.com/animate/>

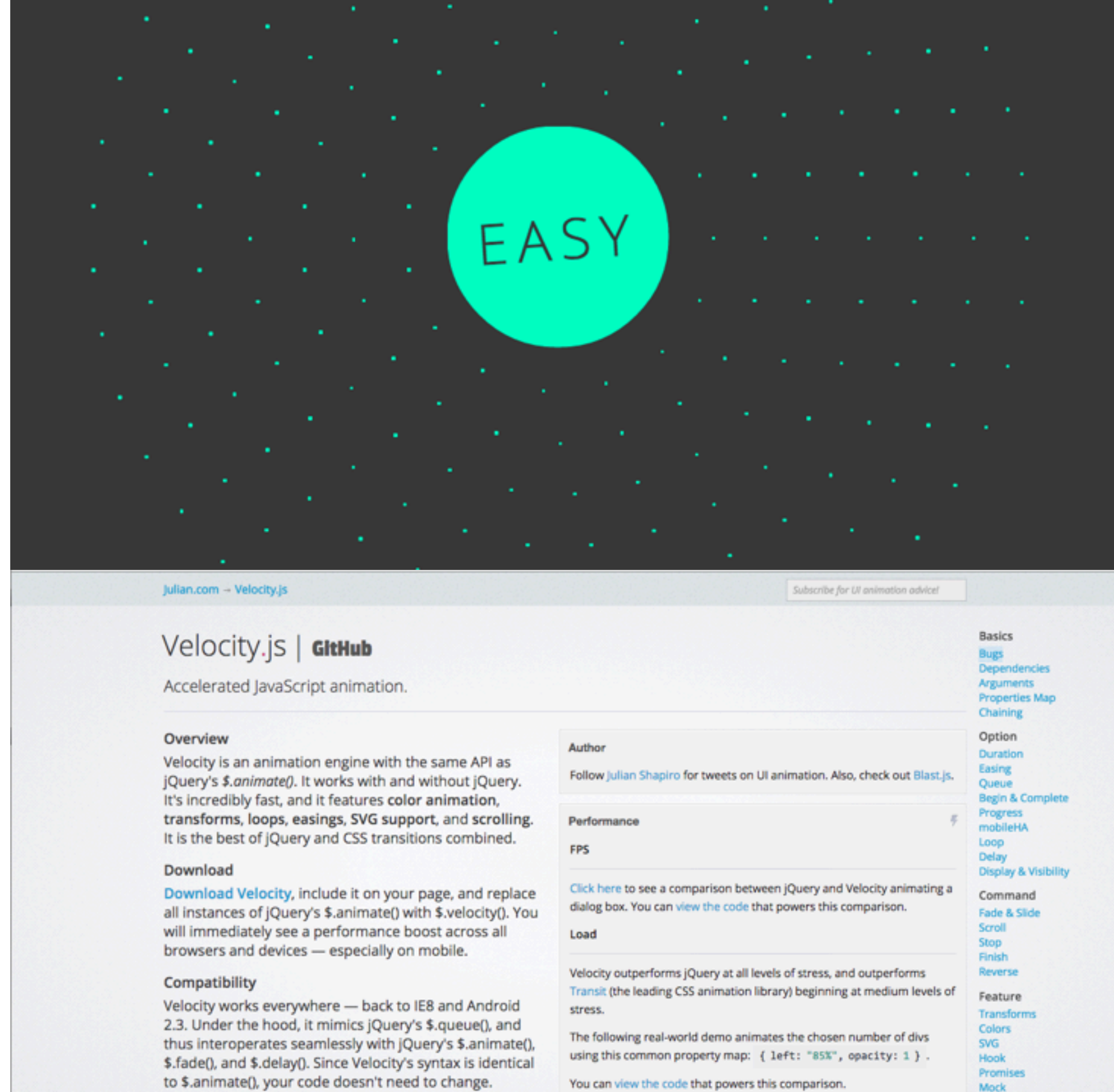
<http://cssanimate.com/>

Velocity.js

Velocity.js seems really hard at the first glance. The documentation confused me. However, the animation is beautiful, and it is clear to see that the library can be used to design complex and detailed animation.

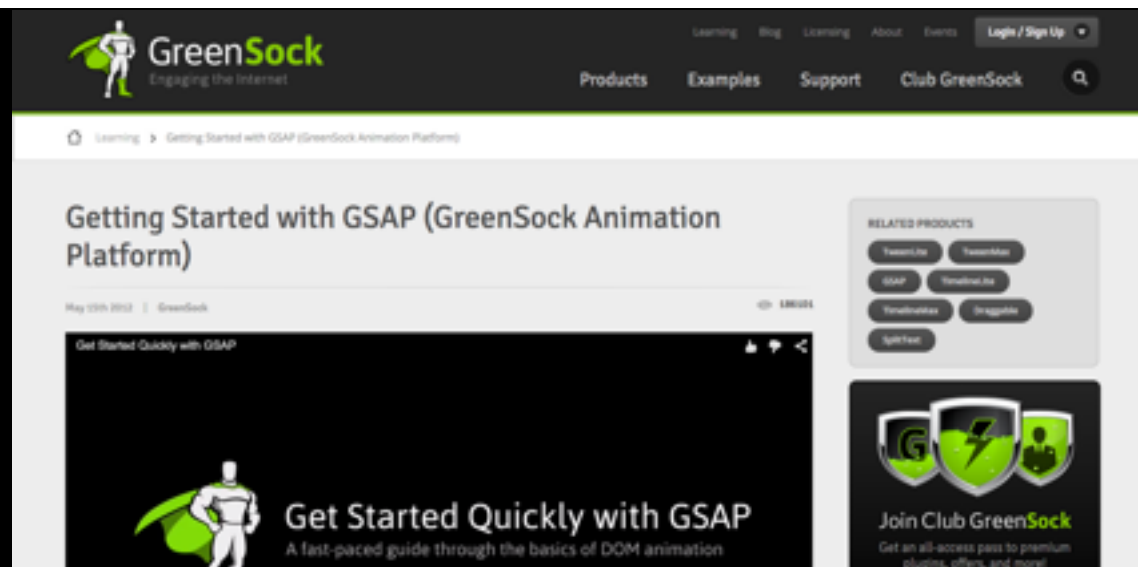
<http://codepen.io/sol0mka/full/kzyjJ/>

<http://julian.com/research/velocity/>



GSAP

When I'm doing research on how animation on the web could be done, I saw this beautiful demo. This demo led me to the introduction of GSAP, which is a Javascript library. I watched the tutorial and was amazed by how simple it is. Here I found an article, "Animating without jQuery" that compares GSAP, Velocity.js and jQuery. The article is a worth reading.



<http://codepen.io/gordonnl/full/byouf/>

<http://greensock.com/get-started-js>

<http://www.smashingmagazine.com/2014/09/04/animating-without-jquery/>

The Process of Making

Some thoughts

During the process, I found GSAP really easy to learn. It's forgiving, simple and clear. It turned out that the thing the challenged me the most is CSS instead of Javascript. In order to adapt with my poor CSS skill, I was forced to revise my design for several times.

One of the problem was my workflow. I didn't do the layout with CSS first before I implement the animation. I have been designing while the development at the same time. It should be easier if I have everything positioned well.


```

1 <!DOCTYPE html>
2 <html>
3 <head>
4
5
6
7 <script src="http://cdnjs.cloudflare.
  com/ajax/libs/gsap/1.16.0/TweenMax.
  min.js"></script>
8 <link rel="stylesheet" type="text/css"
  href="animation.css">
9
10 <script src="animation.js"></script>
11
12 </head>
13 <body>
14
15   <div class="container">
16
17     <div id="butterfly"></div>
18     <div id="restart"><a href="#"
19       >RESTART</a></div>
20     <div class="namewrapper">
21       <div class="letters">
22         <div class="letter">
23           <div class="linecircle">
24             <div class="butterfly">
25               <div class="restart">
26                 <div class="namewrapper">
27                   <div class="letters">
28                     <div class="letter">
29                       <div class="linecircle">
30                         <div class="butterfly">
31                         </div>
32                       </div>
33                     </div>
34                   </div>
35                 </div>
36               </div>
37             </div>
38           </div>
39         </div>
40       </div>
41     </div>
42   </div>
43 </body>
44 </html>
45

```

```

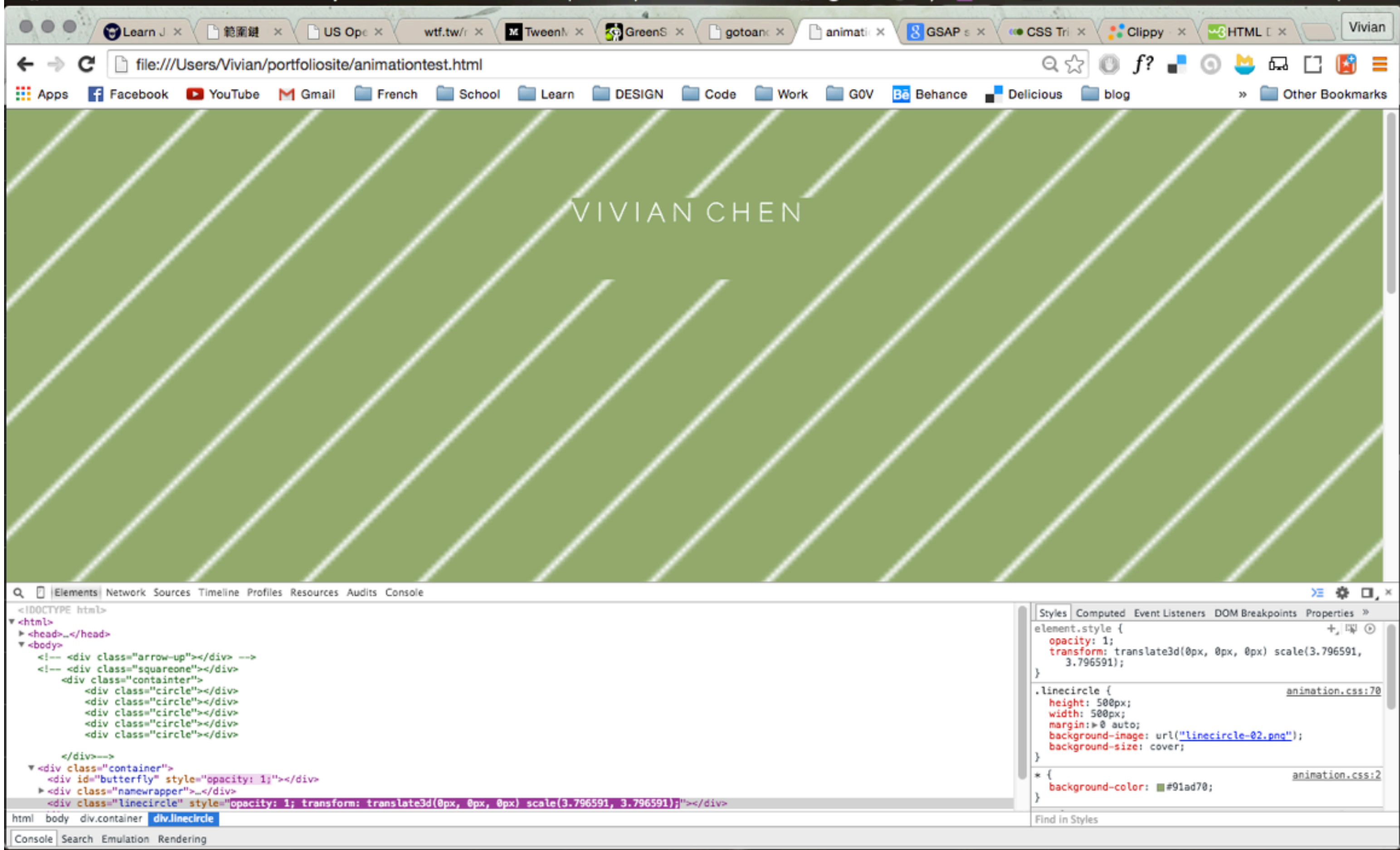
1 @import url(http://fonts.googleapis.com/css?fami
2 * {
3   background-color: #66Bab7;
4 }
5 body{
6   overflow: hidden;
7 }
8
9
10 .letter{
11   color: white;
12   display: inline-block;
13 }
14
15 .letters{
16   font-size: 200%;
17   text-align: center;
18   margin: 0 auto;
19   z-index: 20;
20   font-family: "Raleway Dots";
21   position: absolute;
22 }
23
24 .namewrapper{
25   text-align: center;
26   z-index: 20;
27   position: relative;
28   margin: 0 auto;
29   height: 80px;
30   width: 300px;
31   z-index: 10;
32 }
33
34 #butterfly{
35   background-image: url("butterfly-03.png");
36   background-size: contain;
37   background-repeat: no-repeat;
38   height: 150px;
39   width: 220px;
40   display: block;
41   margin: 0 auto;
42   z-index: 10;
43 }
44
45 .container{
46   height: 720px;
47 }
48

```

```

1
2 window.onload = function(){
3
4   var tl1 = new TimelineLite();
5
6   tl1.staggerFrom(".letter", 1, {scale: 0}, 0.1);
7   tl1.from("#butterfly", 0.5, {opacity: 0});
8
9
10  document.getElementById("restart").onclick = func
11  document.getElementById("restart").onmouseover = f
12    TweenMax.to("#restart", 0.1, {opacity: 1, borderSt
13    .to("a:link", 0.1, {textDecoration: "underline"})
14  };
15  document.getElementById("restart").onmouseout = fu
16
17
18  tl1.from(".linecircle", 2, {
19    scale: 0,
20    opacity: 0,
21    rotation: 500,
22    ease: Bounce.easeOut
23  }, 0.1);
24
25  tl1.to(".linecircle", 0.5, {rotation: 45}).to(".linec
26
27  tl1.to(".letter", 1, {fontSize: 110, margin: 'auto'}, "-
28
29  tl1.to(".linecircle", 1, {
30    opacity: 0,
31    scale: 0,
32    zIndex: 0,
33    ease: Bounce.easeOut
34  }, "end");
35
36  tl1.to(".letter", 1, {scale: 0, display: 'none'}, "end")
37  tl1.to("#butterfly", 1, {y: 500}, "end");
38
39  tl1.to("#restart", 1, {y: 260, opacity: 0}, "end")
40    .to("#restart", 1, {display: "inherit", opacity: 0})
41    .to("#restart", 2, {opacity: 0.5}, "-=0.5");
42
43  var tl2 = new TimelineLite();
44  tl2.to("#restart", 1, {background-color: "#01ed70", "-=0.5"});
45

```



VIVI
ANC
HEN





Final Design

<http://vivianyachen.ca/isma309/>

A large circle with diagonal white lines on a green background.

The End