# Animation on the Web

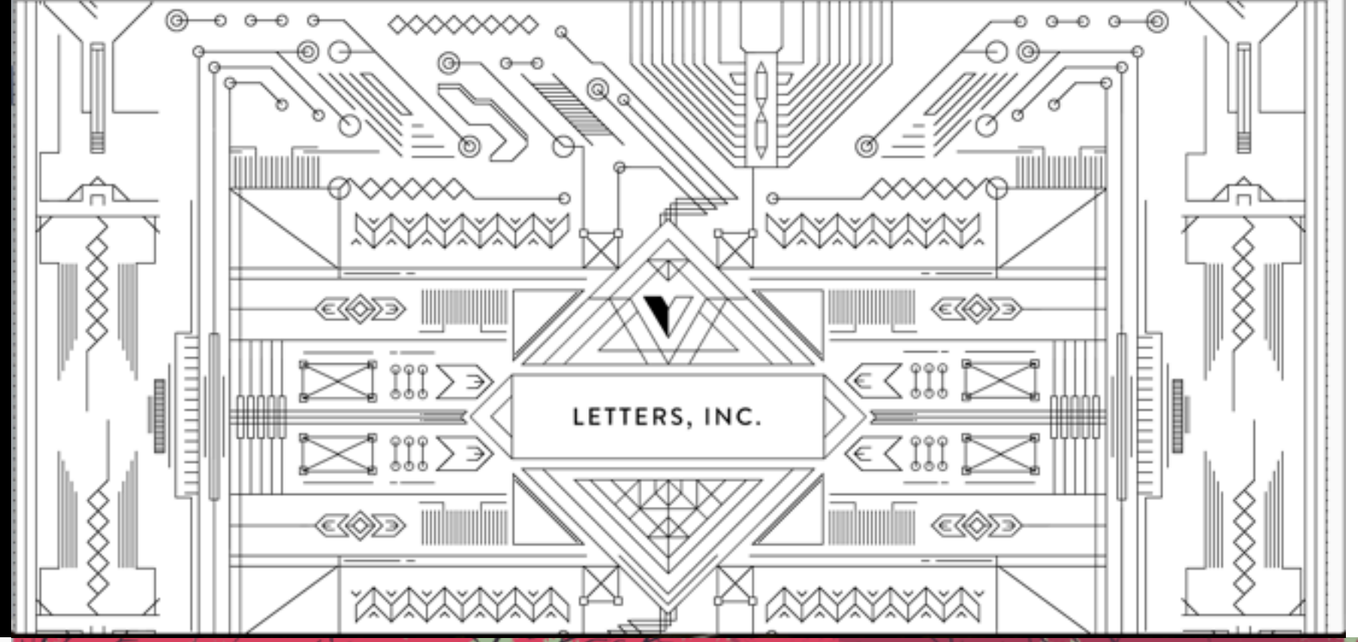
Vivian Chen

# 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






http://letters-inc.jp/

http://gardenestudio.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.

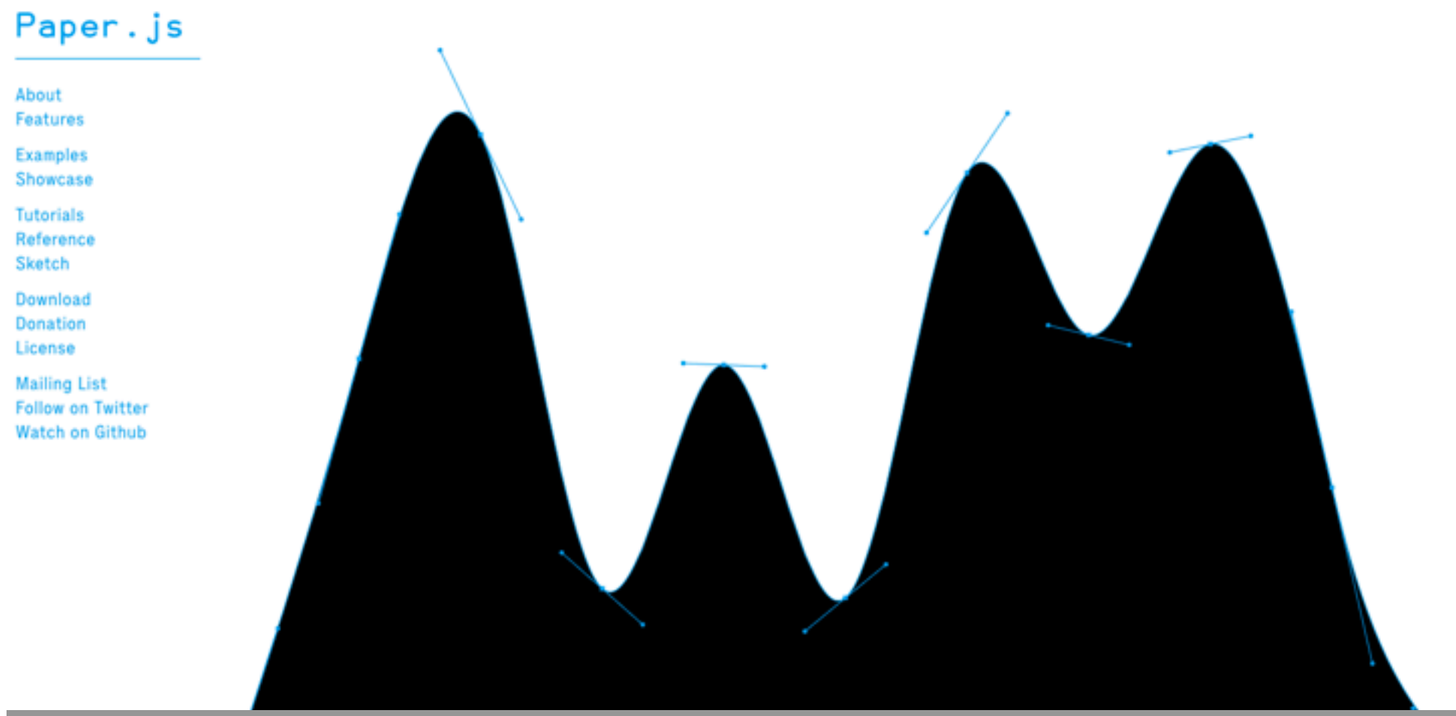

http://janpaepke.github.io/ScrollMagic/

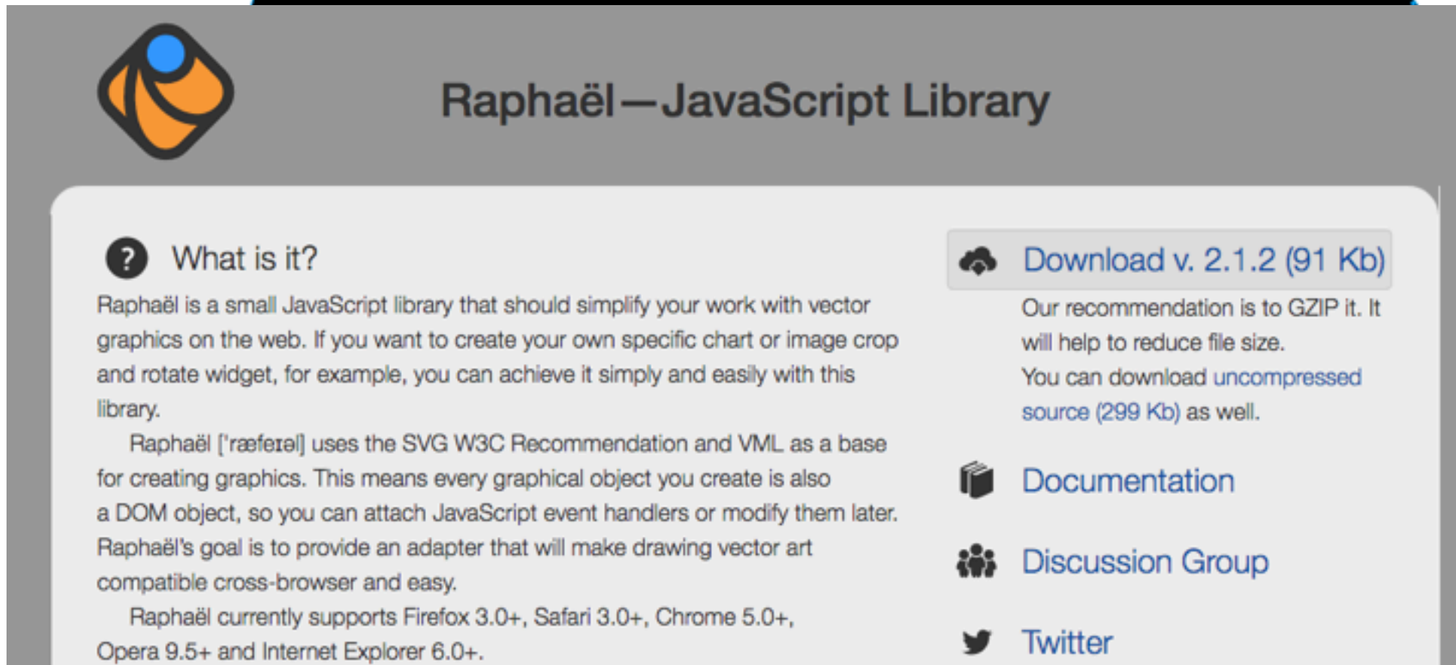
# Canvas

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 Clothe”. 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/

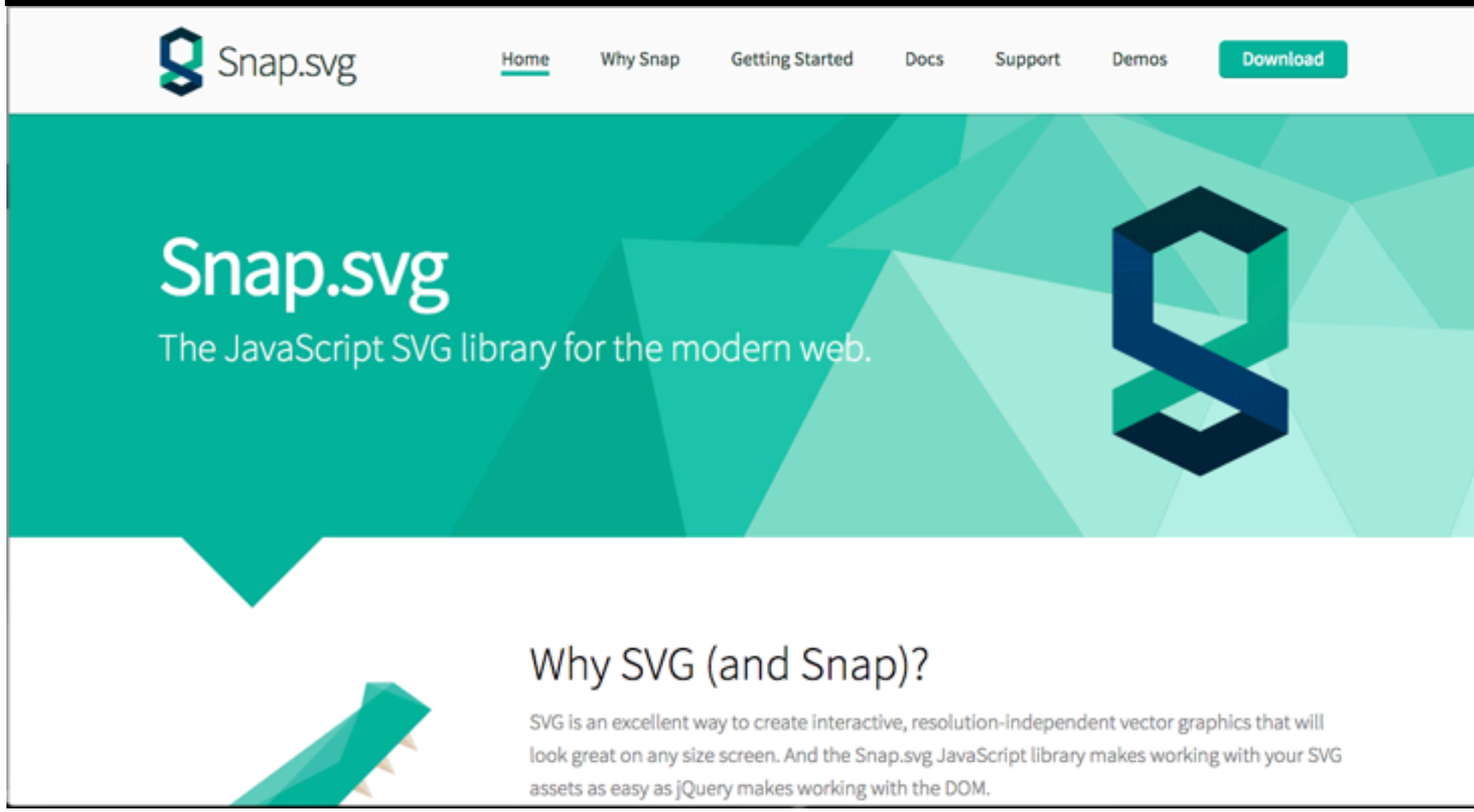


# Snap.svg

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.

http://snapsvg.io/



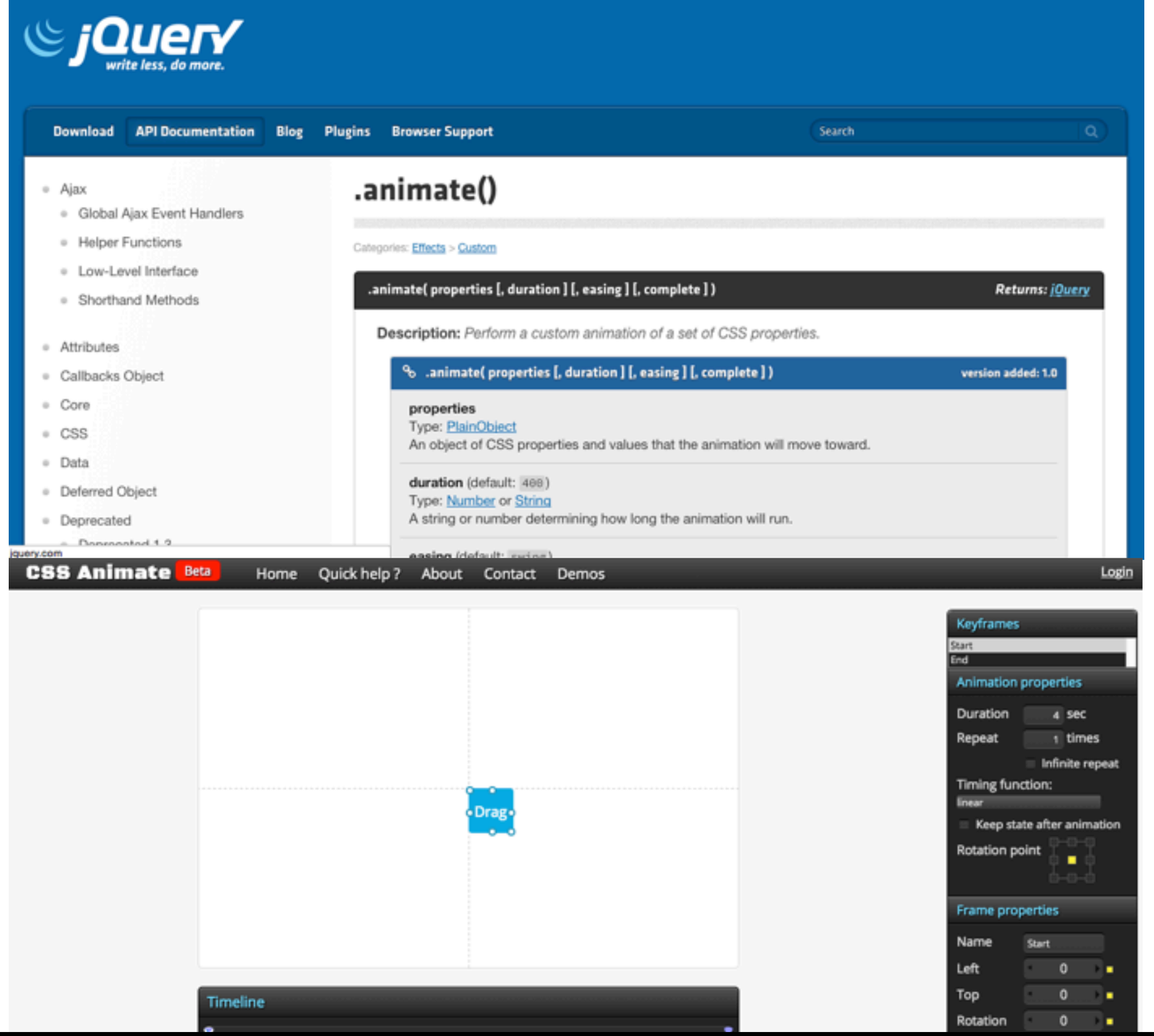

http://codepen.io/ghepting/pen/xnezB

# 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 us 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.




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

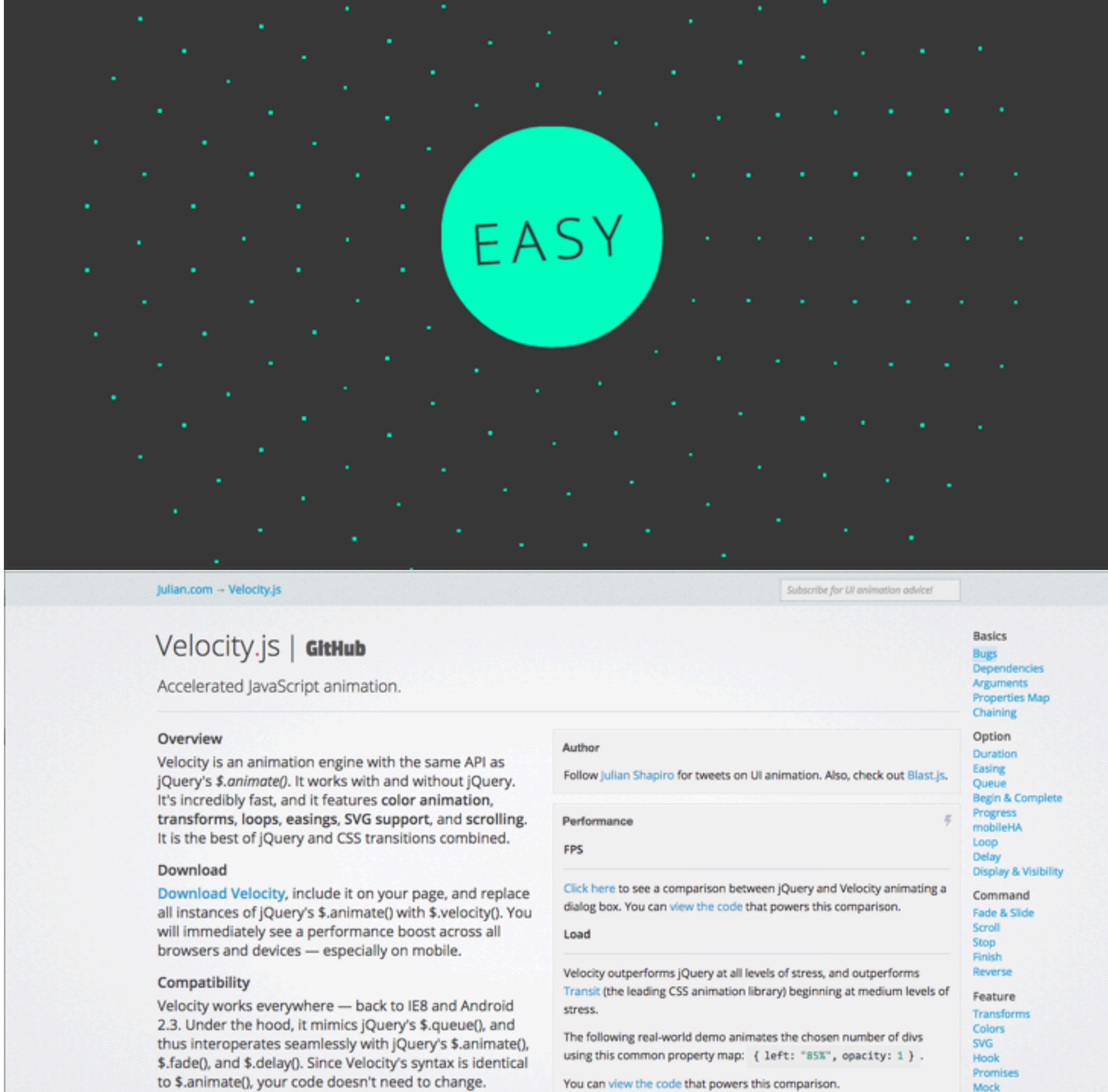
http://cssanimate.com/

# 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.

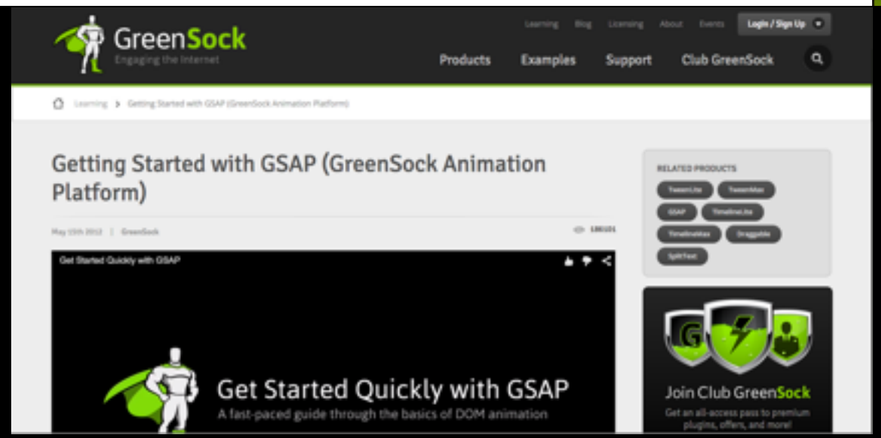
http://codepen.io/sol0mka/full/kzyjJ/

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.

# tutorials

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

https://greensock.com/jump-start-js

http://greensock.com/docs/#/HTML5/GSAP/

index.html | animationtest.html

```html
<!DOCTYPE html>
<html>
<head>



<script src="http://cdnjs.cloudflare.com/ajax/libs/gsap/1.16.0/TweenMax.min.js"></script>
<link rel="stylesheet" type="text/css" href="animation.css">

<script src="animation.js"></script>

</head>
<body>

    <div class="container">

        <div id="butterfly"></div>
        <div id="restart"><a href="#">RESTART</a></div>
        <div class="namewrapper">
                <div class="letters">
                ...
                </div>
        </div>

        <div class="linecircle"></div>

    </div>

</body>


</html>
```

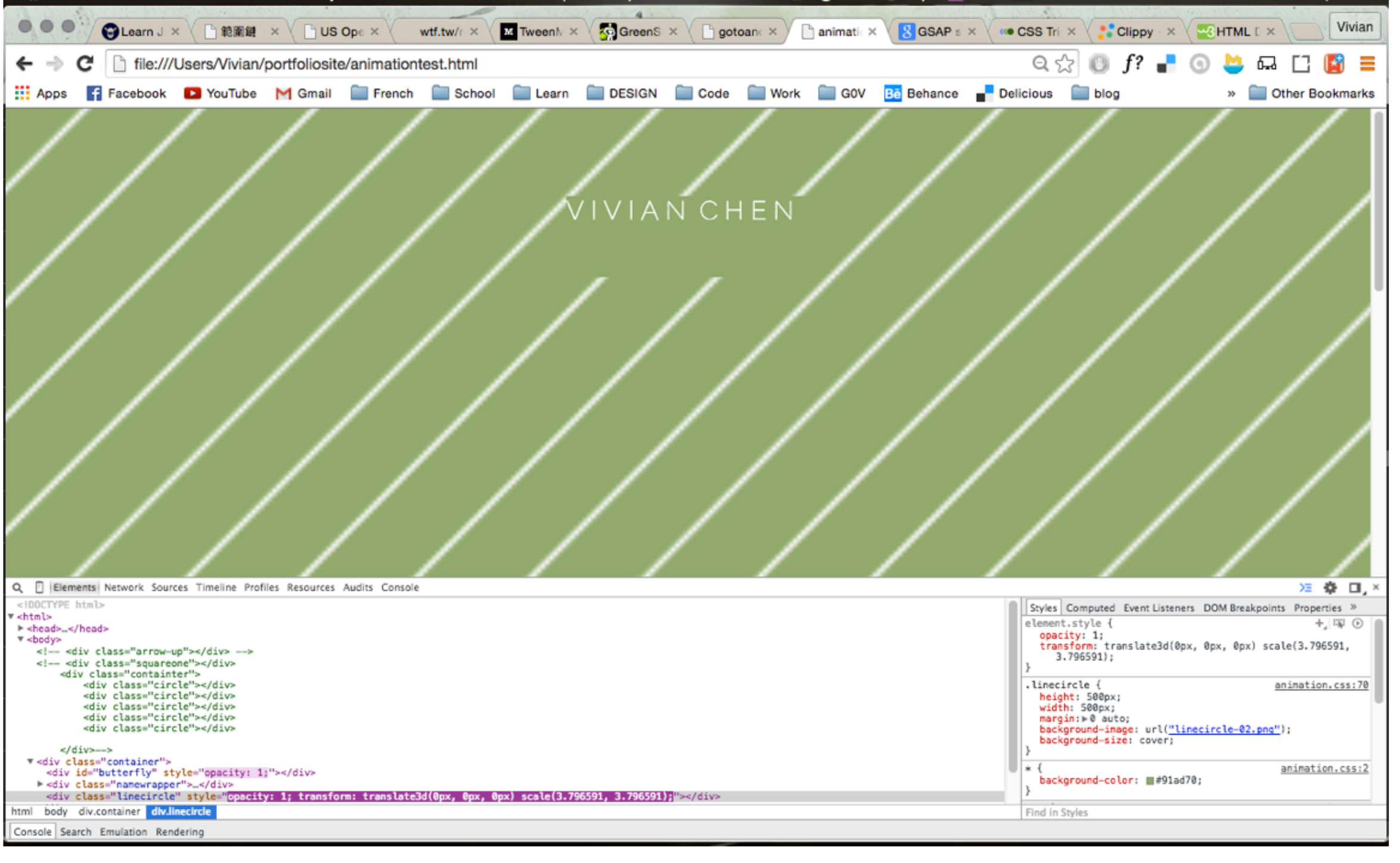
animation.css

```css
@import url(http://fonts.googleapis.com/css?fami
* {
    background-color:#66Bab7
}
body{
    overflow:hidden;
}


.letter{
    color:white;
    display:inline-block;

}
.letters{
    font-size: 200%;
    text-align: center;
    margin:0 auto;
    z-index: 20;
    font-family: "Raleway Dots";
    position:absolute;
}
.namewrapper{
    text-align: center;
    z-index: 20;
    position:relative;
    margin:0 auto;
    height:80px;
    width:300px;
    z-index: 10;
}
#butterfly{
    background-image: url("butterfly-03.png");
    background-size:contain;
    background-repeat: no-repeat;
    height:150px;
    width:220px;
    display: block;
    margin: 0 auto;
    z-index: 10;
}
.container{
    height:720px;
```

animation.js

```js
window.onload = function(){

 var tl1 = new TimelineLite();

 tl1.staggerFrom(".letter",1,{scale:0},0.1);
 tl1.from("#butterfly", 0.5 ,{opacity:0});


 document.getElementById("restart").onclick = func
document.getElementById("restart").onmouseover = f
    TweenMax.to("#restart",0.1,{opacity:1,borderSt
    .to("a:link",0.1,{textDecoration:"underline"})
};
document.getElementById("restart").onmouseout = fu


tl1.from(".linecircle",2,{
    scale:0,
    opacity:0,
    rotation:500,
    ease:Bounce.easeOut
},0.1);

tl1.to(".linecircle",0.5,{rotation:45}).to(".linec

tl1.to(".letter",1,{fontSize:110,margin:'auto'},"-

tl1.to(".linecircle",1,{
    opacity:0,
    scale:0,
    zIndex:0,
    ease:Bounce.easeOut
},"end");

tl1.to(".letter",1,{scale:0,display:'none'},"end")
tl1.to("#butterfly",1 ,{y:500},"end");

tl1.to("#restart",1,{y:260,opacity:0},"end")
.to("#restart",1,{display:"inherit",opacity:0})
.to("#restart",2,{opacity:0.5},"-=0.5");

var tl2 = new TimelineLite();
```

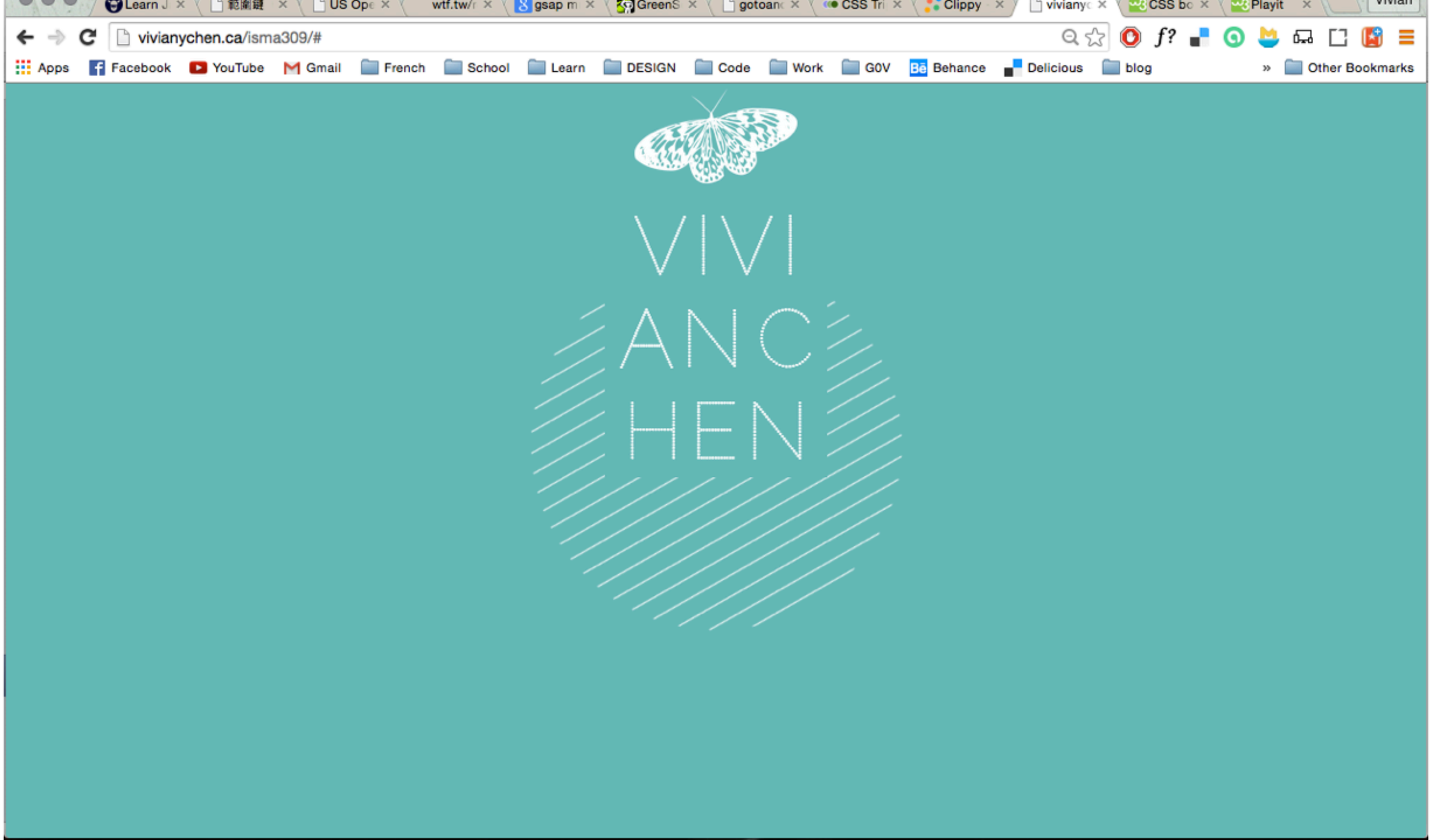
Find | Find Prev | Find All


Learn J
範圍鍵
US Op
wtf.tw/r
TweenM
GreenS
gotoan
animati
GSAP s
CSS Tri
Clippy
HTML
Vivian
file:///Users/Vivian/portfoliosite/animationtest.html
Apps
Facebook
YouTube
Gmail
French
School
Learn
DESIGN
Code
Work
G0V
Behance
Delicious
blog
Other Bookmarks
VIVIAN CHEN
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<!DOCTYPE html>
<html>
<head>…</head>
<body>
<!-- <div class="arrow-up"></div> -->
<!-- 

-->

…

html
body
div.container
div.linecircle
Styles
Computed
Event Listeners
DOM Breakpoints
Properties
element.style {
opacity: 1;
transform: translate3d(0px, 0px, 0px) scale(3.796591, 3.796591);
}
.linecircle {
animation.css:70
height: 500px;
width: 500px;
margin:▶0 auto;
background-image: url("linecircle-02.png");
background-size: cover;
}
* {
animation.css:2
background-color: #91ad70;
}
Find in Styles
Console
Search
Emulation
Rendering


vivianychen.ca/isma309/#
Apps
Facebook
YouTube
Gmail
French
School
Learn
DESIGN
Code
Work
G0V
Behance
Delicious
blog
Other Bookmarks
VIVI
ANC
HEN

# Final Design

http://vivianychen.ca/isma309/


The End