Data Visualization in the Browser

Rob Larsen

DATE

Download

What we're going to talk about

- About me
- SVG & Canvas
- Libraries like Raphael, D3, and Highcharts

Me

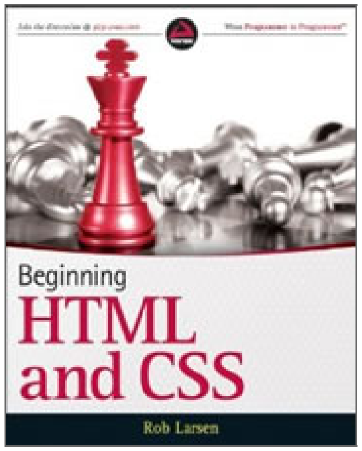
Work

- I've been making web sites since 1997
- I've been primarily focused on HTML, CSS & JavaScript that whole time
- Formerly an agency guy/consultant. Lots of big brand stuff (Adidas, Motorola, Samsung, etc.)
- Nowadays I'm the client & I come with with 100% more suits

Seriously, **suits**

Work

- I'm roblarsen on Github
- I'm **@robreact** on Twitter
- I have a blog @ **htmlcssjavascript.com**
- I wrote this book. It just came out.



Art

Data Visualization

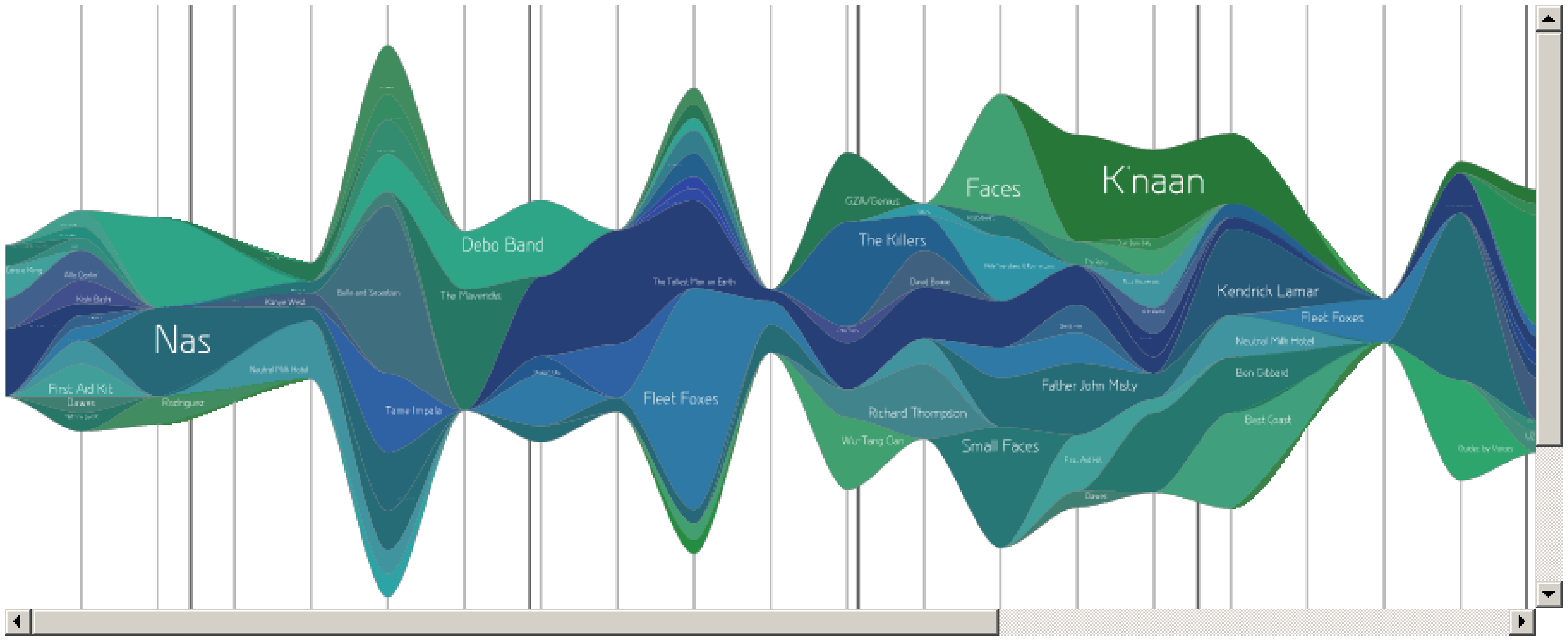
Rhymes with crufty? **Photo from Andreas_MB on Flickr**

Basically

Taking this

```
{"username": "rob_react", "weeks": {"1152446400": {"Pink": 2, "The Decemberists": 3, "Jerry Jeff Walker
": 7, "The Clash": 2, "Jim Carroll Band": 2, "Big Audio Dynamite II": 1, "The Beatles": 12, "Cheap Tric
k": 1, "Rod Stewart": 3, "Ninja Tunes": 1, "Nas": 1, "Boogie Down Productions": 1, "Everclear": 8, "DAN
GERDOOM": 2, "Paul Simon": 1, "Macy Gray": 1, "Jem": 1, "Terror Danjah Ft. Riko, Bruza, D Double E And
Hyper": 1, "Kelly Willis": 1, "Herman's Hermits": 1, "Tool": 1, "Sound Tracks": 1, "The Cure": 1, "Bell
e and Sebastian": 9, "Tom Waits": 2, "Nick Drake": 1, "Jammer Ft. Wiley, D Double E, Kano and Goodz": 2
, "Oasis": 4, "Acid House Kings": 2, "Ralph Stanley & Friends": 1, "Joe Jackson": 1, "The Libertines":
7, "Wilco": 1, "Sublime": 5, "The Cardigans": 1, "Bill Hicks": 3, "Harry Nilsson": 1, "Operation Ivy":
1, "Bob Dylan": 11, "Joni Mitchell": 1, "Led Zeppelin": 1, "50 Cent": 1, "Black Crowes & Wilco": 1, "Ka
no": 1, "Mason Jennings": 7, "Dave Matthews Band": 1, "The White Stripes": 1, "Michael Penn": 4, "Ralph
 Stanley": 1, "Kirsty MacColl / The Pogues": 1, "Billy Bragg & Wilco": 2, "Misfits": 3, "The Pretenders
": 2, "The Pogues": 12, "Sangue Misto": 1, "Rufus": 1, "The Beach Boys": 2, "Marvin Gaye": 2, "The Stro
kes": 3, "The Streets": 5, "The Stanley Brothers": 5, "Franz Ferdinand": 1, "Dave Matthews": 1, "My Mor
ning Jacket": 1
```

And turning it into this



Generated with the awesome **http://lastgraph3.aeracode.org/**

The Technologies

Scalable Vector Graphics (SVG)

SVG has had a long and strange journey to the world of "emerging technologies." SVG was actually defined in 1999, so it’s not exactly the new kid on the block.

Still, it took a long while to catch on, so I don't get bent out of shape when people lump it in with "HTML5 and Friends."

Vector graphics mixed up with XML.

When created in the context of an HTML document SVG elements are just slightly funky DOM elements. That means properties are stored as part of the regular DOM and access to individual elements is available using traditional DOM access methods like `document.geElementById` and `document.getElementsByTagName`; or, if you only speak jQuery `$()`

It can also be styled with CSS.

SVG can also be output from a vector based drawing program and used as the `src` of an `<img>` element or as the background of an element as defined by CSS.

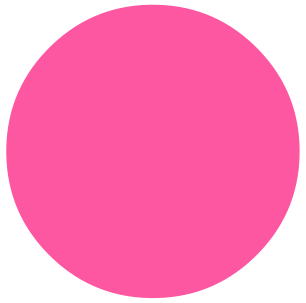
Vector means: it scales!

HTML
HTML
HTML
HTML
HTML

HTML

5

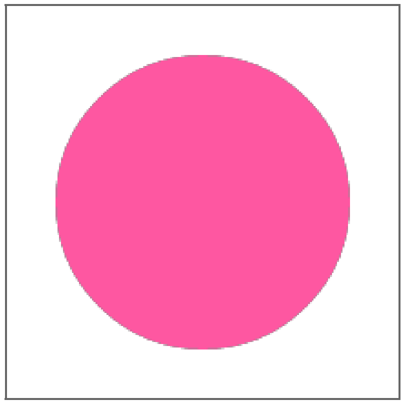
It looks like this



Viewing Source

```
<svg width="200" height="200"
    viewPort="0 0 200 200" version="1.1"

    xmlns="http://www.w3.org/2000/svg">

    <circle cx="100" cy="100" r="75" fill="#fe57a1"/>

</svg>
```

It can also look like this



Viewing Source

```
<img src="img/circle.svg">
```

Support

5.5 6 7 8 9 10

2 3 3.5 3.6 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

3.1 3.2 4 5 5.1 6

9 9.5-9.6 10.0-10.1 10.5 10.6 11 11.1 11.5 11.6 12 12.1 12.5

3.2 4.0-4.1 4.2-4.3 5.0-5.1 6

5.0-7.0

2.1 2.2 2.3 3 4 4.1 4.2

7 10

10 11 11.1 11.5 12 12.1

0

0

data from caniuse.com

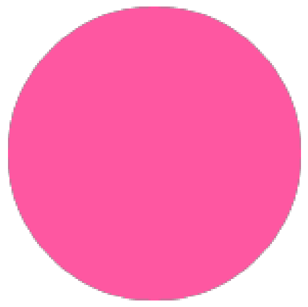
Canvas

The canvas element and associated API started life as an Apple extension to HTML. From there it blossomed into one of the early stars of the HTML5 era.

The canvas element provides a scriptable interface for drawing two-dimensional images in the browser. Think... dynamic PNGs. Even without full browser support on the desktop, developers have embraced canvas fully.

It's an extremely low level API. This means you can sometimes do more work but you have complete, pixel level control (*cue diabolical laughter*). It's been used for everything from high traffic visualizations to game engines, a popular system for delivering custom fonts, and a port of the Processing programming language into JavaScript.

It looks like this



Viewing Source

```
<canvas id="circle" width="800" height="300"></canvas>
```

```
var ctx = document.getElementById("circle").getContext("2d");
ctx.beginPath();
ctx.arc(400, 150, 75, 0, Math.PI*2, true);
ctx.fillStyle = "#fe57a1";
ctx.closePath();
ctx.fill();
```

Did I say low level?

I sure did. Obviously, `arc` is all you need when you have `Math.PI*2` :)

```
ctx.arc(400, 150, 75, 0, Math.PI*2, true);
```

Support

5.5 6 7 8 9 10

2 3 3.5 3.6 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

3.1 3.2 4 5 5.1 6

9 9.5-9.6 10.0-10.1 10.5 10.6 11 11.1 11.5 11.6 12 12.1 12.5

3.2 4.0-4.1 4.2-4.3 5.0-5.1 6

5.0-7.0

2.1 2.2 2.3 3 4 4.1 4.2

7 10

10 11 11.1 11.5 12 12.1

0

0

data from caniuse.com

Comparing Canvas & SVG

At a High Level

Generally... Go SVG when dealing with a limited number of elements, interactivity and limited animation requirements. It's more familiar to the average front end engineer and it's much easier to implement in an accessible way.

Canvas is better for applications that require intense animation, real-time interaction and/or **many** more elements. SVG elements are DOM elements so thousands of them is kind of scary. Thousands of "elements" in the canvas is still just the one element being manipulated.

Of course, things can bog down with Canvas too.

Scaling matters

If different resolutions and form factors are the kind of thing that keeps you up at night, make nice with SVG. SVG is like a miracle cure for what ails you.

Your Mileage May Vary

This issue could be a whole presentation all by itself. In lieu of that, here are a couple of articles which cover this question in greater depth:

- **How To Choose Between SVG and Canvas**
- **How to Choose Between Canvas and SVG for your Site**

In Practice

These are the tools and approaches I've used over the past few years and can speak about with some authority. While these aren't the only solutions to these problems, these are ones that have been successful for me.

SVG: Highcharts

“Highcharts is a charting library written in pure JavaScript, offering intuitive, interactive charts to your web site or web application. Highcharts currently supports line, spline, area, areaspline, column, bar, pie, scatter, angular gauges, arearange, areasplinerange, columnrange and polar chart types.”

SVG: Highstock

“Highstock lets you create stock or general timeline charts in pure JavaScript, including sophisticated navigation options like a small navigator series, preset date ranges, date picker, scrolling and panning.”

Commercial products, but... awesome.

- Pluses:
  - Super powerful
  - Well Documented
  - Old IE support via VML
  - Rock solid
- Minuses:
  - Not free

Viewing Source

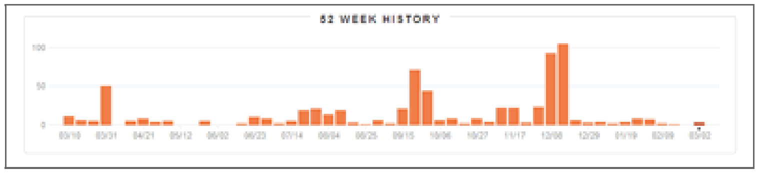
```
var seriesOptions = [],
   yAxisOptions = [],
   seriesCounter = 0,
   names = ['AAPL', 'BBRY'],
   colors = Highcharts.getOptions().colors;
$.each(names, function(i, name) {
  $.getJSON('../data/'+names[i].toLowerCase()+'.json', function(data) {
     for (var j = 0, len = data.length; j < len; j++) {
      data[j][0] = Date.parse( data[j][0] ).getTime();
    }
     data = data.reverse();
    seriesOptions[i] = {
       name: name,
       data: data
    };
    seriesCounter++;
    if (seriesCounter == names.length) {
      createChart( seriesOptions );
    }
  });
});
function createChart(seriesOptions){
var chart = new Highcharts.StockChart({
    chart: {
        renderTo: 'chart'
    },
    yAxis: {
      labels: {
```

SVG: D3

"D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation."

data from the **Hubway Data Visualization Challenge**

- Pluses:
  - Super powerful
  - Under active development



  - Many examples
- Minuses:
  - No support for old IE

View Source

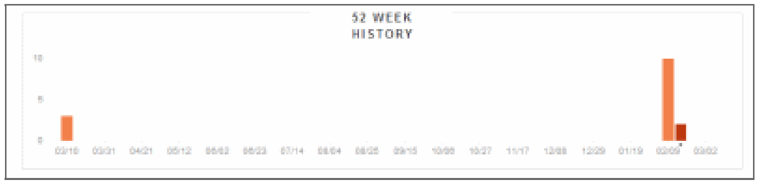
```
var chord = d3.layout.chord()
    .padding(.05)
    .sortSubgroups(d3.descending)
    .matrix(matrix);
var width = 900,
    height = 500,
    innerRadius = Math.min(width, height) * .35,
    outerRadius = innerRadius * 1.1;

var fill = d3.scale.ordinal()
             .domain(d3.range(4))
             .range(["#336699", "#99ccff", "#6699cc", "#0066cc"]);

var svg = d3.select("body").append("svg")
            .attr("width", width)
            .attr("height", height)
            .append("g")
            .attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");

svg.append("g").selectAll("path")
   .data(chord.groups)
   .enter().append("path")
   .style("fill", function(d) { return fill(d.index); })
   .style("stroke", function(d) { return fill(d.index); })
   .attr("d", d3.svg.arc().innerRadius(innerRadius).outerRadius(outerRadius))
   .on("mouseover", fade(.1))
   .on("mouseout", fade(1));
```

SVG: Raphaël

“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.”

Background photo from stintje on flickr

- Pluses:
  - Fun, intuitive API
  - Built-in OLD IE support
  - Lower Level API
- Minuses:
  - Lower Level API
  - Hopefully not dead, just resting:



View Source

```
var paper = Raphael(0, 0, 900, 550);
var alpe = paper.path("M71.5,538.641l29.937-15.665l30.633-14.271l30.981-16.014 l31.329-14.9
68l30.633-12.88l30.981-10.791l29.937-10.443l32.026-12.184l29.936-11.835l30.285-11.488l30.63
3-10.442l30.981-12.88 l29.937-11.488l30.633-13.924l30.285-10.792L562.672,335l29.937-11.835l
31.329-9.747l29.937-12.88l32.374-11.487l30.285-12.879  l30.285-11.835l29.938-12.184l29.937-
11.836l31.678-10.095L868.307,225h13.576  L886,538 L71.5,538")
    .attr('fill','url(../slides/img/alpe.jpg)');

function climb( name, time, hex, endpos){
  var rider = paper.text(10, 10, name)
       .attr("fill", hex)
       .attr("font-size", "14px");
  rider.animateAlong({
    path: "M71.5,538.641l29.937-15.665l30.633-14.271l30.981-16.014 l31.329-14.968l30.633-12
.88l30.981-10.791l29.937-10.443l32.026-12.184l29.936-11.835l30.285-11.488l30.633-10.442l30.
981-12.88 l29.937-11.488l30.633-13.924l30.285-10.792L562.672,335l29.937-11.835l31.329-9.747
l29.937-12.88l32.374-11.487l30.285-12.879  l30.285-11.835l29.938-12.184l29.937-11.836l31.67
8-10.095L868.307,225h13.576",
    rotate: false,
    duration: time,
    easing: 'ease-out',
    debug: false
    },function(){ rider.animate({x:endpos.x,y:endpos.y}, 1000)
  });
}

climb("Pantani 1997 37.35", 3735, "#06c", {x:100,y:30});
climb("Lance 2004 37.36", 3736, "#cc0", {x:100,y:60});
```

SVG: Other Libraries

- **SVG.js**
- **Polymaps**

Canvas: Core Canvas

“This specification defines the 2D Context, Level 2 for the HTML canvas element. The 2D Context provides objects, methods, and properties to draw and manipulate graphics on a canvas drawing surface.”

- Pluses:
  - Not a library
  - POWERFUL
- Minuses:
  - Lower level API

View Source

```
function draw(data) {
  ctx.fillRect(0,0,900,300);
  var len = data.length,
      width = 900/len,
      gradient;
      gradient = ctx.createLinearGradient(0, 0, 0, 300);
      gradient.addColorStop(".5", "#003366");
      gradient.addColorStop("1.0", "#ccff00");
      ctx.fillStyle = gradient;
  for (var i=0; i<len; i++) {
      ctx.fillRect(i,300,width,-data[i]);
  }
}
```

Canvas: Canvas.JS

> *“CanvasJS is a small helper library for the canvas 2d API. The goal is to extend and enhance the basic API while still remaining familiar”*

I'm partially to blame

- Pluses:
  - Familiar API, just enhanced
  - Tiny (4k gzipped)
  - It's got a really nice personality
  - Chaining
- Minuses:
  - Full of youthful exuberance
  - Remains low level

View Source

```
function draw(data) {
  ctx.reset();
  var len = data.length,
      width = 900/len,
      fill;
  for (var i=0; i<len; i = i + 10) {
    var dataLen = data[i];
      for (var j = 0; j < dataLen; j = j+10){
        fill = "rgb("+j+","+(255-j)+",255)";
        ctx.fillCircle({
          x : i,
          y : 300 - j,
          radius : 4,
          fillStyle : fill
        });
      }
   }
}
```

Canvas: JavaSCript InfoVis Toolkit

“The JavaScript InfoVis Toolkit provides tools for creating Interactive Data Visualizations for the Web.”

- Pluses:
  - Strong foundation
  - Straightforward API
- Minuses:
  - Out of the box visualizations are skeletal

View Source

```
icicle = new $jit.Icicle({
    injectInto: 'infovis',
    animate: animate,
    offset: 1,
    cushion: false,
    constrained: true,
    levelsToShow: 4,
    Tips: {
      enable: true,
      type: 'Native',
      offsetX: 20,
      offsetY: 20,
      onShow: function(tip, node){
        var count = 0;
        node.eachSubnode(function(){
          count++;
        });
        tip.innerHTML = "<div class=\"tip-title\"><b>Name:</b> "
            + node.name + "</div><div class=\"tip-text\">" + count
            + " children</div>";
      }
    },
    Events: {
      enable: true,
      onClick: function(node){
        if (node) {
          icicle.tips.hide();
          icicle.enter(node);
```

Canvas: Other Libraries

- **sigma.js**
- **paper.js**
- **Peity**
- **arbor.js**
- **envision**
- **CanvasQuery**

Bonus: Google Maps

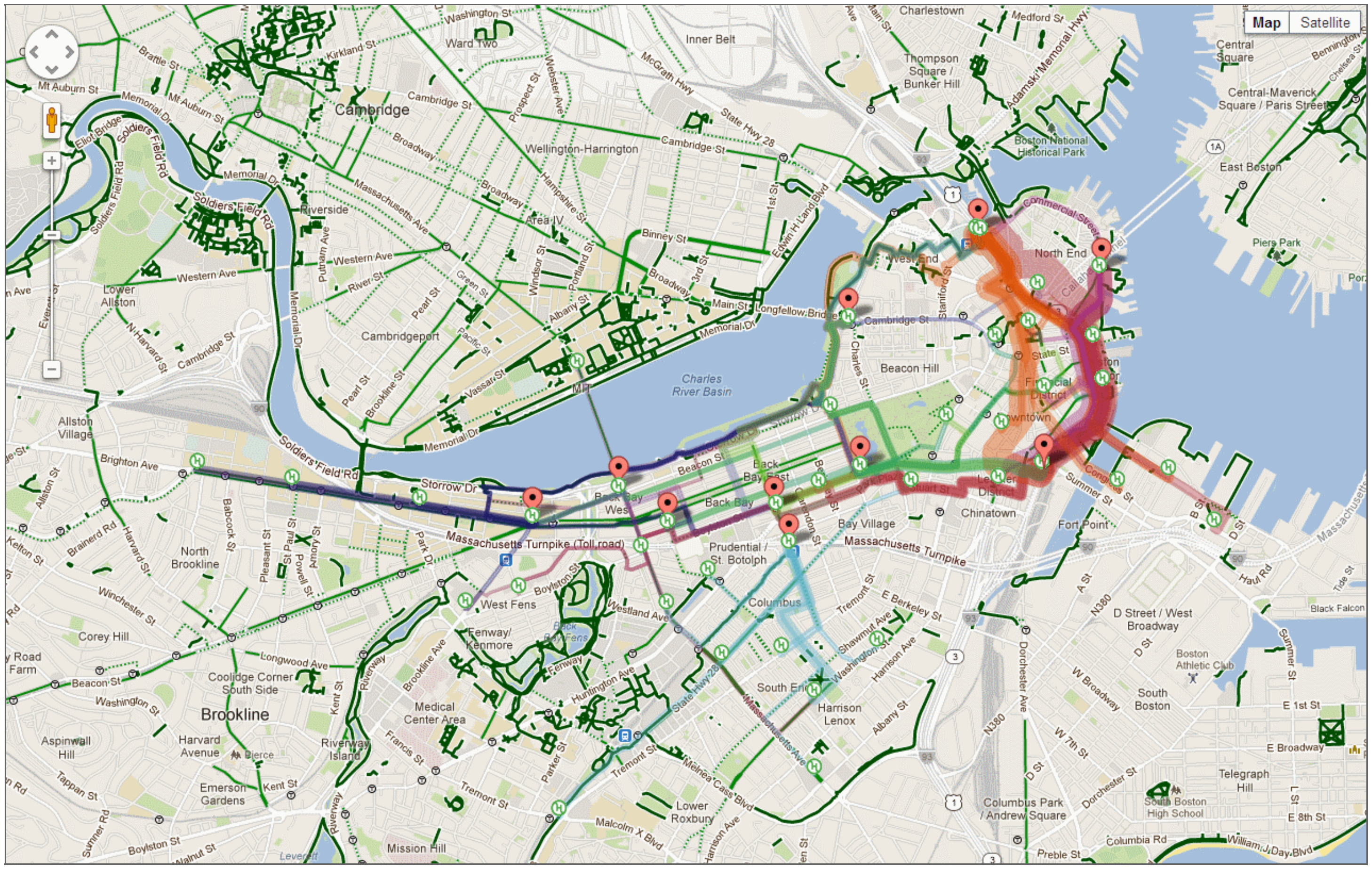
“

The Google Maps Javascript API lets you embed Google Maps in your own web pages. Version 3 of this API is especially designed to be faster and more applicable to mobile devices, as well as traditional desktop browser applications.

The API provides a number of utilities for manipulating maps (just like on the http://maps.google.com web page) and adding content to the map through a variety of services, allowing you to create robust maps applications on your website.

”

The Maps API is already rich, adding a layer of data on top of it opens up a wide range of possibilities


Map
Satellite
Cambridge
Charles River Basin
Beacon Hill
North End
Back Bay
Chinatown
South End
Brookline
Fenway/Kenmore
East Boston
Charlestown
South Boston
Massachusetts Turnpike (Toll road)
Storrow Dr
Memorial Dr

- Pluses:
  - Powerful, well documented API
- Minuses:
  - If you get really popular it costs $$

View Source

```
;(function(window, document, $, undefined) {
  "use strict";
  if(window.HW === undefined) {
    window.HW = {};
  };
  var HW = window.HW;
  HW.common = {
    init: function() {
      $("#map").height($(window).height());
      var GM = google.maps,
        defaultPosition = new GM.LatLng(42.3520, -71.0560),
        mapOptions = {
          zoom: 12,
          center: defaultPosition,
          mapTypeId: GM.MapTypeId.ROADMAP
        },
        map = new GM.Map(document.getElementById('map'), mapOptions),
        geocoder = new GM.Geocoder(),
        position, endPosition, marker, polylineOptions, DD, DS;
      $("#map").data("map", map);
      createMarker(0);
      var i = 1, timer;
      function render(){
        if (i == 10){
          clearInterval(timer);
          return;
        }
        createMarker(i);
```

Thanks!