

# **D3.jsで オープンデータを可視化しよう**

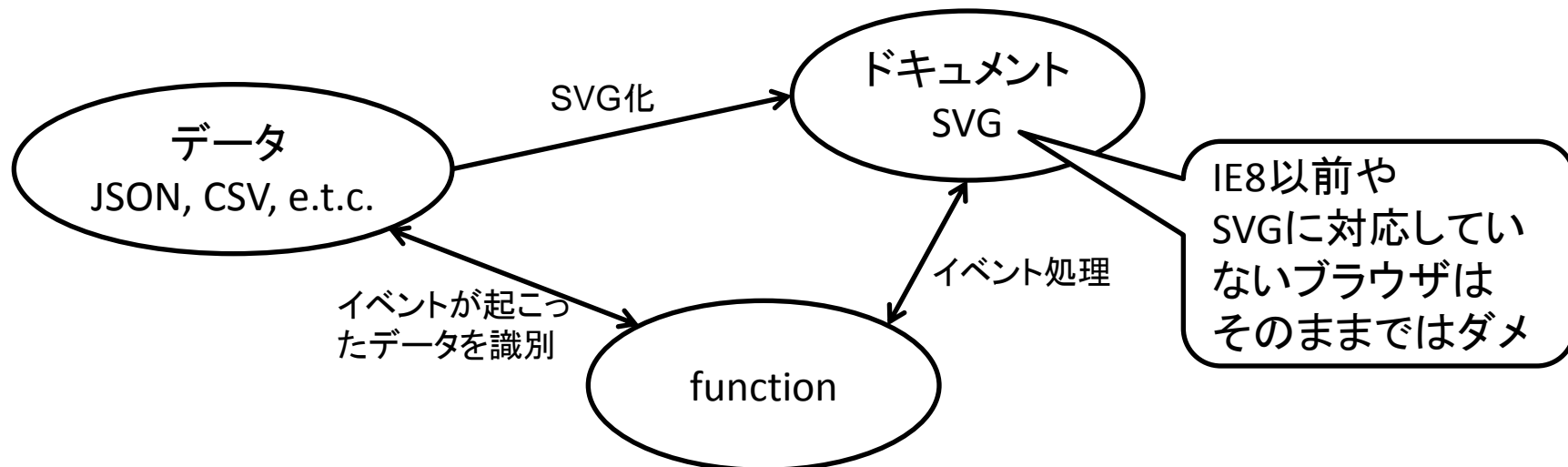
**2014年5月20日**

**先端IT活用推進コンソーシアム  
クラウド・テクノロジー活用部会  
荒本 道隆**

**D3.js** (またはD3:Data-Driven Documents、旧:Protovis<sup>[1]</sup>)は、2011年に開発が始まった<sup>[2]</sup>[ウェブブラウザ](#)で動的コンテンツを描画する[JavaScriptライブラリ](#)である。[World Wide Web Consortium](#)準拠の[データ可視化](#)ツールとして、[Scalable Vector Graphics](#) (SVG)、JavaScript、[HTML5](#)、[Cascading Style Sheets](#)を最大限に活用している。その他多くのライブラリとは対照的に、最終的に出力された結果に視覚的な調整ができる。<sup>[3]</sup>

ウィキペディアより

- データをドキュメント化して、関係も保持



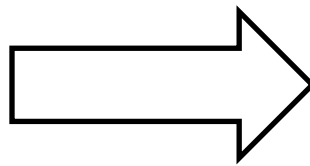
- 使えそうなデータを発見
  - オープンデータ・API、社内システムのデータ・API
  - 必要に応じてフォーマット変換
- そのデータに合った表現方法を探す
  - D3.js Examples
    - <https://github.com/mbostock/d3/wiki/Gallery>
- サンプルのソースコードを流用
- あっという間に完成

慣れればこんな感じ(きっと)

# 本日の目標

- オープンデータを可視化する
  - 2つのデータをマッシュアップ
  - 地図と雨量:どちらも生活の中で身近なデータ
- 地図を中心にビジュアライズ
  - D3.js は、地図描画の機能も充実
  - 緯度・経度の扱いに注意

完成イメージ  
さらに機能を追加して  
『シズツシュ』を目指す



<http://sipos.shizuoka2.jp/sipos/index.html>

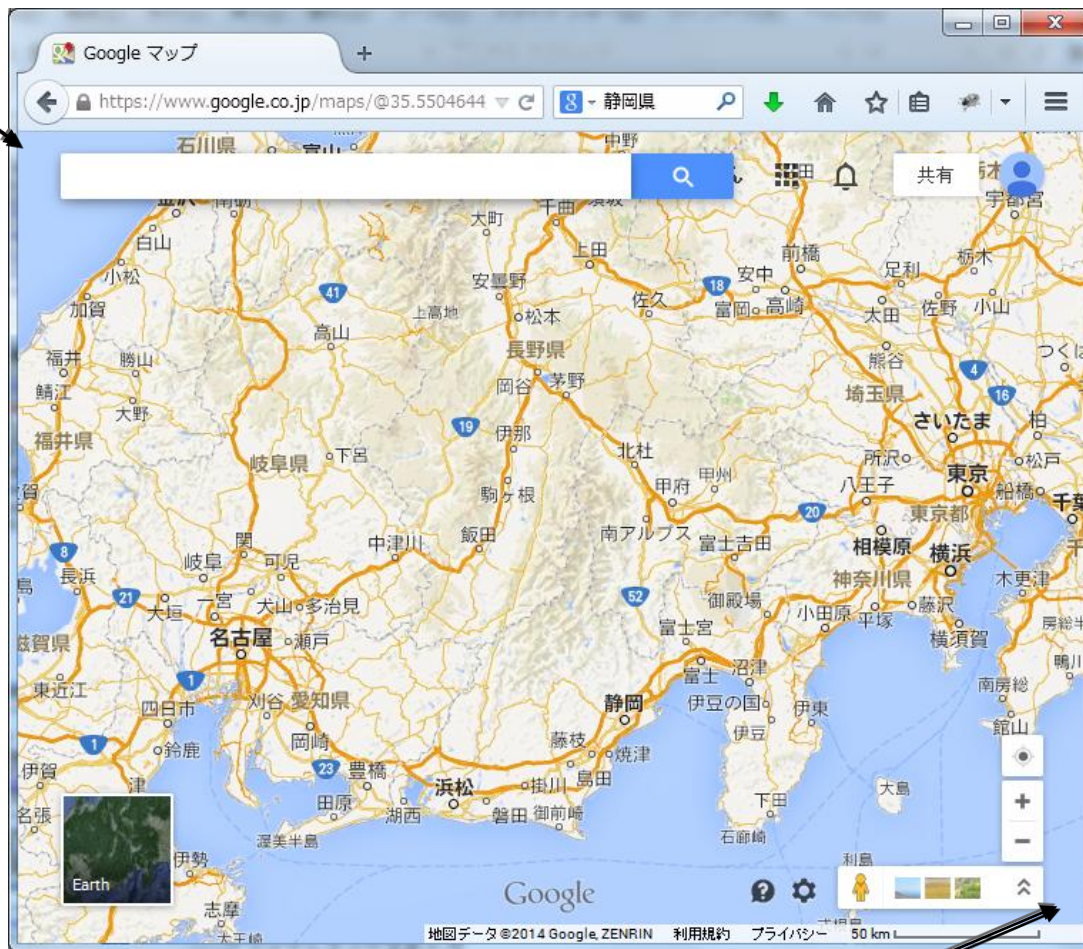
# 緯度・経度を扱う上での注意

0, 0  
136.0895, 36.7678

画面

画面内に描画するため  
・画面の原点と、緯度・経度の原点が違う  
・表示スケールの変更

緯度・経度



900, 600  
140.1379, 34.3596

- ふじのくにオープンデータ
  - <http://open-data.pref.shizuoka.jp/>
  - 静岡県のリアルタイム河川情報
    - 10分ごとに、過去24時間分取れる
  - 24時間以内に雨が降っていないと、何も表示されない
    - 何月何日に雨が降ったか確認
      - <http://weather.yahoo.co.jp/weather/jp/past/22/5010.html?c=2014&m=3>
      - 2014/4/28, 4/29, 5/5, ...
- 4/26以降、10分ごとにwgetして蓄積しておいた
  - 雨量データ
    - <http://aramoto.sakura.ne.jp/shizuoka2/Rain/>
  - 水位データ
    - <http://aramoto.sakura.ne.jp/shizuoka2/WaterLevel/>
  - 観測局情報
    - <http://aramoto.sakura.ne.jp/shizuoka2/>
      - 文字コードを変換済み



- 地図データ

- D3.js の geo パッケージで読めるデータ形式が必要
  - GeoJSON: 地理情報用に規格されたJSONデータ形式
  - TopoJSON: GeoJSONの拡張形式。D3.jsではプラグインが必要
    - 冗長性を排するので、データサイズが20%程度になる
- Shape形式で入手し、GeoJSONに変換
  - 入手元: 国土交通省、国土数値情報
    - <http://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N03.html>
    - 静岡: N03-130401\_22\_GML.zip
  - このサイトでShapeを10%に圧縮(3,285KB→364KB)
    - <http://www.mapshaper.org/>
    - このサイトで直接GeoJSONをダウンロードしても、うまく表示できなかった
  - Shape→GeoJSONに変換
    - <http://qiita.com/sawamur@github/items/ec32237bcbaba94108d>
- 変換結果(GeoJSON形式、654KB)
  - [http://aramoto.sakura.ne.jp/shizuoka2/gis/shizuoka\\_utf8.json](http://aramoto.sakura.ne.jp/shizuoka2/gis/shizuoka_utf8.json)

- JavaScriptからオープンデータにアクセスするために
  - － ファイルの位置関係が重要
    - JavaScript: ローカルで編集し、そのままブラウザで表示
      - － file:///C:/Users/.....
    - 参照したいデータファイル: ネット上にある
      - － <http://aramoto.sakura.ne.jp/shizuoka2/>.....
  - － JavaScriptとデータファイルが別ドメインの場合
    - そのままでは、クロスドメイン制約にひっかかる
      - － **実際の挙動: 通信がうまくいった後に、ブラウザがエラーを返す**
    - 回避方法
      - － データファイルを置いているWebサーバ側で、HTTPヘッダに追加
        - » Header append Access-Control-Allow-Origin: \*
      - － 詳しくは、『ajax access control』でググってください
      - － ちなみに、「ふじのくにオープンデータ」はこのヘッダが付いていない
        - » 直接は、アプリは利用できるけど、JavaScriptは利用できない
  - － ネットが遅い人は、事前に全ファイルをダウンロード
    - Firefoxのみ有効
    - 同ドメインへのアクセスとなるので、何も問題は起こらない



- D3.js をダウンロード
  - <http://d3js.org/>
    - d3.zip をダウンロードし、d3.min.js を取り出す
    - js ディレクトリを作成し、格納
- jQuery をダウンロード
  - データファイルへのアクセスをAjaxで行う&色々と便利
  - <http://jquery.com/download/>
    - [Download the compressed, production jQuery 1.11.1](#)
    - 同jsディレクトリに格納
- 必要な人は、データファイルを事前にダウンロード
- テキストエディタ
  - 使い慣れた物が一番
- ブラウザ
  - [F-12]を押せば、開発ツールが起動する
  - Firefox: firebugを入れ、「ツール」→「Web開発」→「Firebug」→「Firebugを開く」
  - Safari: 「環境設定」→「詳細」→「メニューバーに”開発”メニューを表示」
  - Chrome: 「ツール」→「デベロッパーツール」
    - ローカルだとうまく動作しない、ネット上のデータを参照すればOK
  - IE(9以降): 「F12 開発者ツール」
    - ローカルだとうまく動作しない

- セレクタ (W3C Selectorsを参照)
  - `d3.select("#hoge")` → `<xxx id="hoge">` を対象
  - `d3.select(".hoge")` → `<xxx class="hoge">` を対象
  - `d3.select("hoge")` → `<hoge>` を対象
- セレクション
  - `selectAll()`, `enter()`, `exit()`
  - 繰り返し処理が楽に書ける
    - `d3.select("#TEXT1").selectAll("p").style("color", "fuchsia");`
- 動的プロパティ
  - `svg.selectAll(".node").data(datas).text(function(d) { return d.title; });`
- データの結合
  - 更新
    - `svg.selectAll(".node").data(datas).text("hogehoge");`
  - 追加
    - `svg.selectAll(".node").data(datas).enter().text("piyopiyo");`
  - 削除
    - `svg.selectAll(".node").data(datas).exit().remove();`
- アニメーション
  - `d3.select("#TEXT4").transition().style("background-color", "black");`

# 以降は、ソースコードを見ながら解説

- D3.jsでデータとSVGが連動するサンプル
  - sample1\_1.html
  - sample1\_2.html
- D3.jsで地図を描画するサンプル
  - sample2.html
- jQueryでデータを取得するサンプル
  - sample3.html
- 取得したデータをD3.jsで描画する
  - sample4.html
    - 緯度・経度の扱いに注意
- 地図に重ねてデータを描画する
  - sample5.html

# 時間の許す限り、拡張してみよう

- 拡張案
  - 雨量に合わせて、色を変える
    - 淡→濃
    - 青→黄→赤
  - 60分雨量を使用する
    - 10分雨量だと、差が分かりにくい
  - 4/30 0000→2350までアニメーションで表示
    - 日も変えられるようにする
  - 地図の投影法を変える
    - サンプルはメルカトル図法を使用
  - 付加情報の表示
    - clickかmouseoverで、地名・住所・雨量などを重ねて表示
    - 文字をもっと見易くする(白抜き、など)
  - 雨量に合わせて地図に色を塗る
  - 雨量の代わりに水位を描画