

D3.jsを使った データビジュアライズ勉強会

2014年7月17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

- 19:30～20:10
 - オープンデータから表示用ファイルを準備
- 20:10～21:30
 - D3.jsで可視化しよう

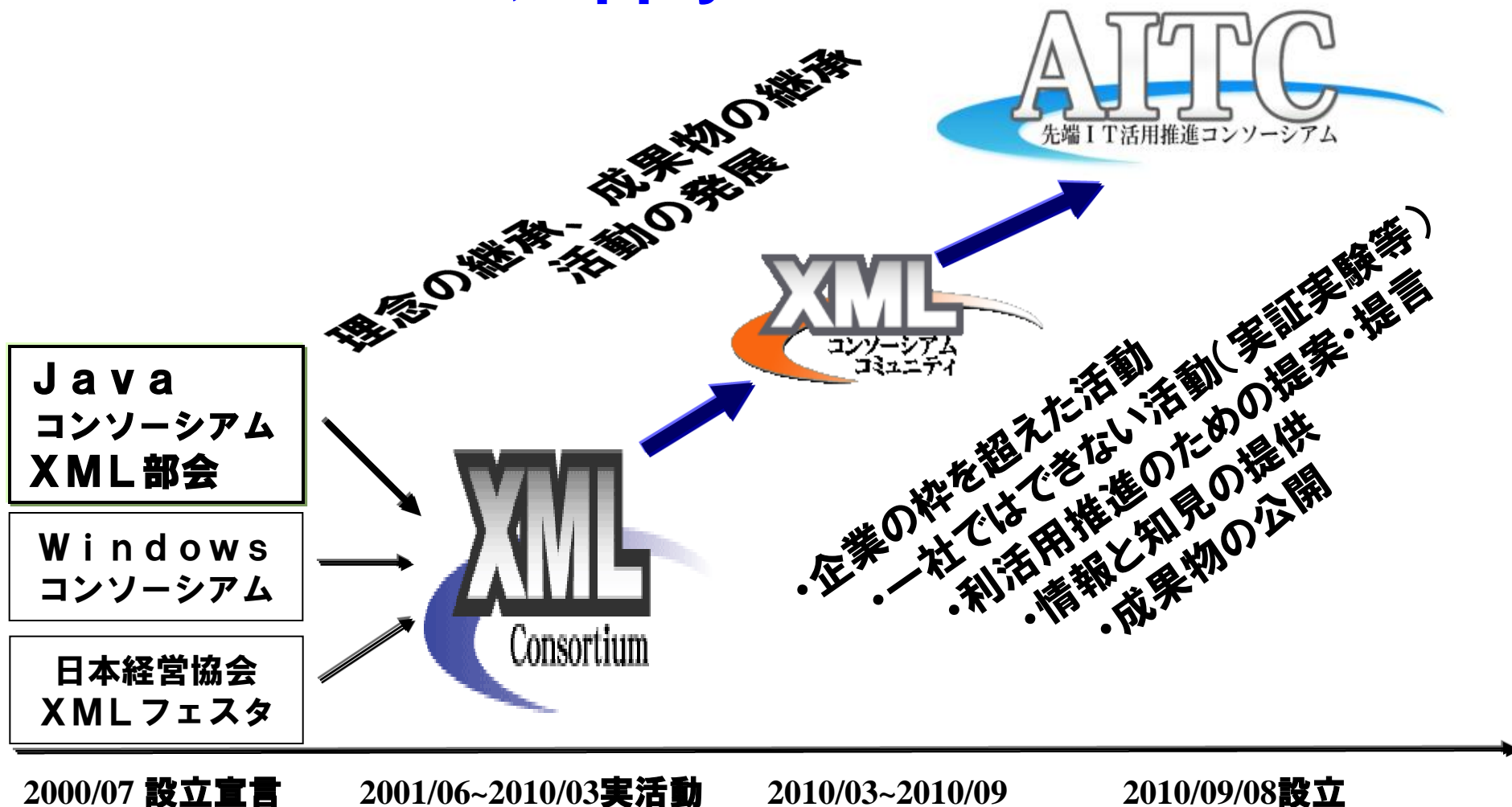
- D3.jsを使ってデータを可視化してみる
 - 数値を眺めているだけでは見えなかったものが、可視化によって何かが見えてきたらステキ
- 本日の内容を別の方法でも実現できるけど
 - Google Maps API
 - <https://developers.google.com/maps/?hl=ja>
 - Microsoft Excel の Power View
 - <http://office.microsoft.com/ja-jp/excel-help/HA102899553.aspx>
- D3.jsで実装することで
 - 思い通りの表現・拡張ができる
 - SVG (Scalable Vector Graphics) のスキルも自然に身に付く
 - 地図以外にも活用できる
 - ほとんどローカルで出来るので
 - 業務データなどでも安心してトライする事ができる

A large, abstract blue graphic with white and yellow accents is positioned on the left side of the slide. It consists of several curved, overlapping shapes that create a sense of motion and depth. The colors range from light blue to dark blue, with white and yellow highlights.

ちょっとだけ 先端IT活用推進コンソーシアム AITC ご紹介

先端IT活用推進コンソーシアム

Advanced IT Consortium
to Evaluate, Apply and Drive



AITCとは：

企業における先端ITの活用および

先端ITエキスパート技術者の育成を目的とし、

もって、社会に貢献することを目指す非営利団体

設立： 2010年9月8日（会期： ～2016年8月31日）

会長： 鶴保 征城（IPA顧問、HAL校長）

会員： 法人会員＆個人事業主、個人会員、学会会員

特別会員（産業技術総合研究所、気象庁、
消防研究センター、防災科学技術研究所）

顧問： 稲見 昌彦（慶応義塾大学大学院 教授）

和泉 憲明（産業技術総合研究所 上級主任研究員）

萩野 達也（慶応義塾大学 教授）

橋田 浩一（東京大学大学院 情報理工学系研究科 教授）

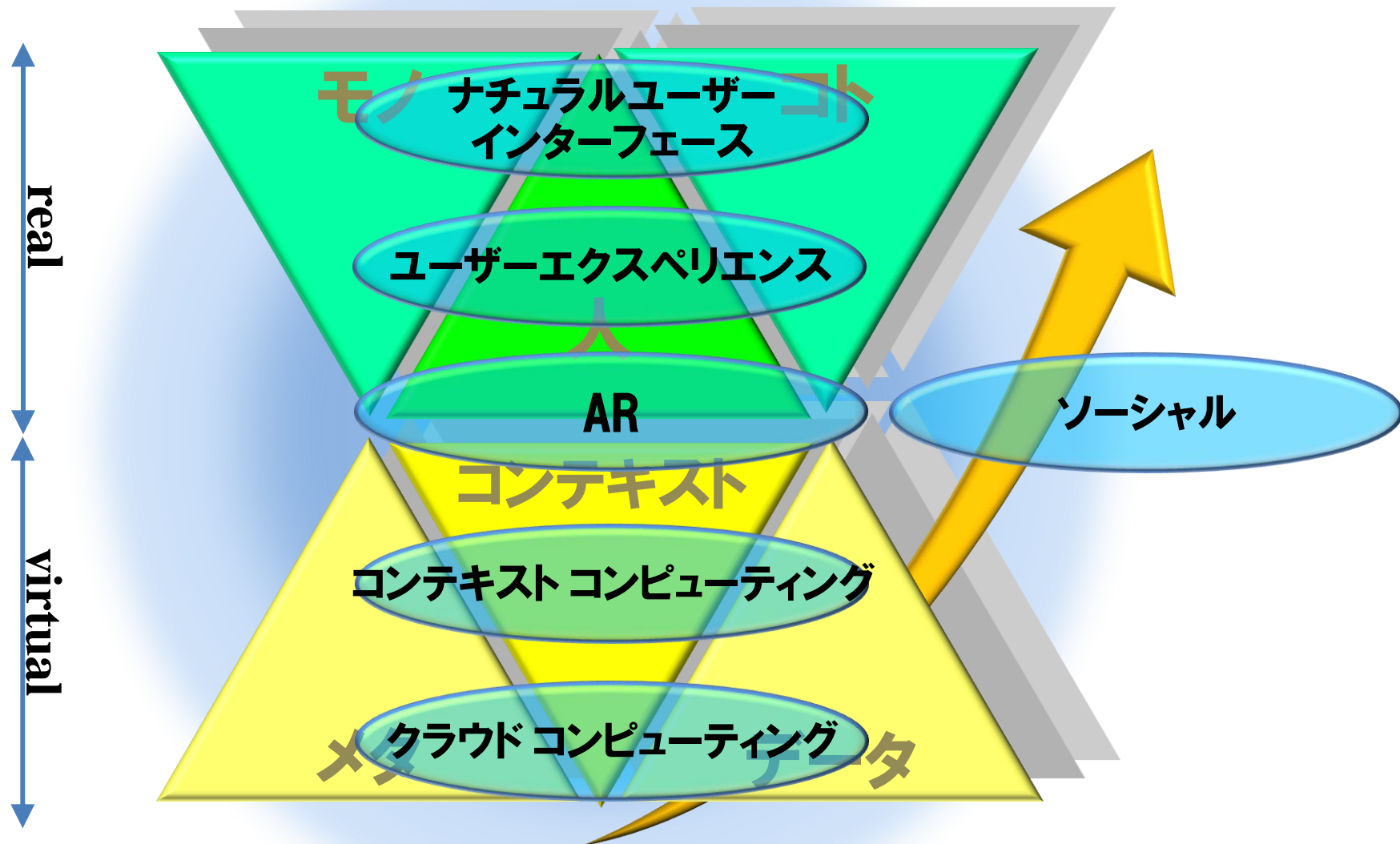
丸山 不二夫（早稲田大学大学院 客員教授）

山本 修一郎（名古屋大学大学院 教授）

BizAR顧問：三淵 啓自（デジタルハリウッド大学大学院 教授）

川田 十夢（AR三兄弟 長男）

第4期活動対象分野



活動の種類と関係性

部会のシーズ

セミナー・勉強会

部会へ
フィードバック

部会

AITC
オープンラボ

部会の
ノウハウを展開

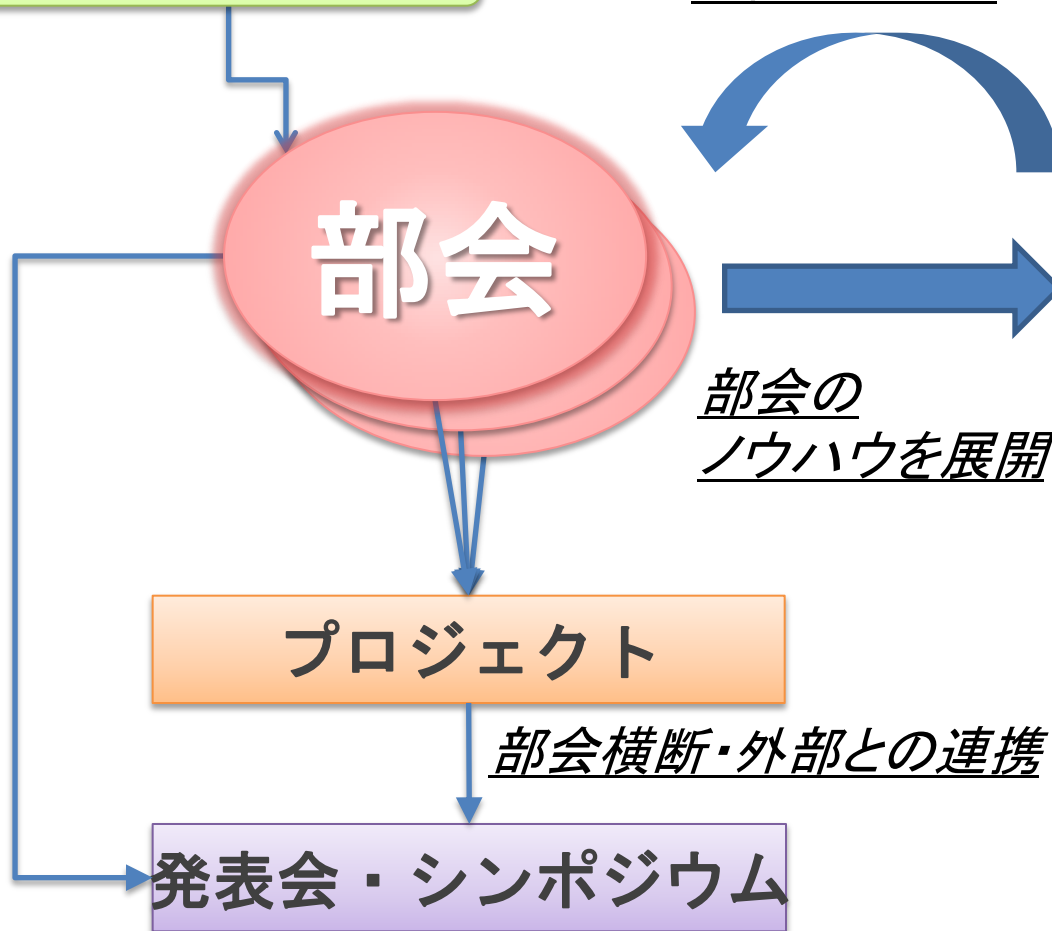
月1回程度の開催
スキルと知見の伝播

プロジェクト

部会横断・外部との連携

発表会・シンポジウム

部会・プロジェクトの成果を発信



オープンデータから 表示用ファイルを準備

- 横浜オープンデータポータル
 - <http://data.yokohamaopendata.jp/>
- ヨコハマ・アート・LOD 

SPARQLによるクエリが可能

 - http://fp.yafjp.org/yokohama_art_lod
- よこはまオープンデータカタログ(試行版)
 - <http://www.city.yokohama.lg.jp/seisaku/seisaku/opendata/catalog.html>
 - 2014年7月14日オープン

オープンデータじゃないけど

- 横浜市統計情報ポータルサイト
 - <http://www.city.yokohama.lg.jp/ex/stat/>

その他のオープンデータ

- Open DATA METI
 - <http://datameti.go.jp/data/ja/dataset>
- データカタログサイト（試行版）
 - <http://www.data.go.jp/>
- オープンデータプラットフォーム
 - <http://odp.jig.jp/>
- 気象庁XML公開サイト
 - <http://api.aitc.jp/>
- 鯖江市
 - <http://www.city.sabae.fukui.jp/pageview.html?id=12768>



SPARQLによるクエリが可能

- オープンデータプラットフォーム <http://odp.jig.jp/>
 - SPARQLで横浜市に関する登録を検索
 - <http://sparql.odp.jig.jp/sparql.html>

```
select * where {  
  ?id ?t <http://odp.jig.jp/odp/1.0#OpenDataCity> .  
  ?id ?p ?o .  
  filter(regex(?o, "横浜市"))  
} limit 10
```

- 神奈川県横浜市 : <http://odp.jig.jp/jp/jig/opendatacity/13>
- 神奈川県横浜市金沢区 : <http://odp.jig.jp/jp/jig/opendatacity/5>

敷居はかなり高いけど、慣れたら「RDBを公開してくれている」感覚

データを準備する上での注意－1

- 文字コード
 - CSVはShift-JISで公開されているものが多い
 - JavaScriptで読み込む場合
 - とりあえず「UTF-8」に変換すれば、トラブルは起こりにくい
- ライセンス
 - データを公開しているサイトの利用規約を参照
 - http://data.yokohamaopendata.jp/dataset/kanazawa_kz-opendata_2013_35-child-care-event
 - リソース→ライセンスを参照

データを準備する上での注意－2

- 他サイト上のデータを直接参照したい
 - － 常に最新データを表示できる
 - 今日やるようなデータ加工をJavaScriptで実装
 - － CORS (Cross-Origin Resource Sharing)
 - データの公開サイトが、他サイトから直接参照を許可しているか？
 - 確認方法
 - － HTTPヘッダに「Access-Control-Allow-Origin: *」があればOK
- 開発時に、全部ローカルでやる場合は
 - － Chromeは起動オプション「--allow-file-access-from-files」
 - － Firefox, Safariは、特に何もしなくてもOK
 - － IEはうまく動作しない
 - ファイルアクセス部分にjQueryを使う事で回避可能

使用するデータを選択

- 使用するデータの選択条件
 - 住所もしくは緯度経度と、数値があるもの
 - 時間とともに変化すると、さらに面白い
- 今回使用するデータ
 - 横浜市総務局 防災関連データ
 - <http://www.city.yokohama.lg.jp/somu/org/kikikanri/data/>
→「地域防災拠点」※注:XMLファイル
クリエイティブコモンズ「表示」(CC BY)
 - 二次著作物を作成する場合は、利用するデータの出典(データのタイトルと著作権者名など)の表示をお願いします。
- 使用するツール
 - Excel で編集を行います
 - ExcelとJavaScriptに自信があれば、他データに挑戦してもOK
 - Excelを持っていない人は
 - Sample ディレクトリ以下からコピーしてください

- XML→CSVに変換する
 - － Excel に対象URLを開く
 - 「XMLテーブルとして開く」を選択
 - 1回目はちょっと時間がかかります
 - － 「名前を付けて保存」でCSVを選択
 - ファイル名は「shelter.csv」
- CSVを加工
 - － Excelを再起動し、「shelter.csv」を開きなおす
 - － フルの住所のカラムをCとDの間に追加
 - タイトルは「住所」
 - ="神奈川県横浜市" & B2 & C2
 - － **他のデータを使う場合も**
 - CSVとして正しい形式で、住所が入っていれば、だいたいOK

データ準備－1－最終形

shelter.csv - Microsoft Excel

	A	B	C	D	E	F	G	H
	Name	Ward	Address	住所	Notes			
1	Name	Ward	Address	住所	Notes			
2	生麦小学校	鶴見区	生麦四丁目15番1号	神奈川県横浜市鶴見区生麦四丁目15番1号	被災した住民の避難生活の場所、情報受任			
3	豊岡小学校	鶴見区	豊岡町27番地1	神奈川県横浜市鶴見区豊岡町27番地1	被災した住民の避難生活の場所、情報受任			
4	鶴見小学校	鶴見区	鶴見中央三丁目19番1号	神奈川県横浜市鶴見区鶴見中央三丁目19番1号	被災した住民の避難生活の場所、情報受任			
5	潮田小学校	鶴見区	向井町3丁目82番地1	神奈川県横浜市鶴見区向井町3丁目82番地1	被災した住民の避難生活の場所、情報受任			
6	下野谷小学校	鶴見区	下野谷町2丁目49番地	神奈川県横浜市鶴見区下野谷町2丁目49番地	被災した住民の避難生活の場所、情報受任			
7	市場小学校	鶴見区	元宮一丁目13番1号	神奈川県横浜市鶴見区元宮一丁目13番1号	被災した住民の避難生活の場所、情報受任			
8	平安小学校	鶴見区	平安町2丁目9番地1	神奈川県横浜市鶴見区平安町2丁目9番地1	被災した住民の避難生活の場所、情報受任			
9	末吉小学校	鶴見区	上末吉一丁目9番1号	神奈川県横浜市鶴見区上末吉一丁目9番1号	被災した住民の避難生活の場所、情報受任			
10	上末吉小学校	鶴見区	上末吉五丁目24番1号	神奈川県横浜市鶴見区上末吉五丁目24番1号	被災した住民の避難生活の場所、情報受任			
11	下末吉小学校	鶴見区	下末吉二丁目25番6号	神奈川県横浜市鶴見区下末吉二丁目25番6号	被災した住民の避難生活の場所、情報受任			
12	旭小学校	鶴見区	北寺尾四丁目25番1号	神奈川県横浜市鶴見区北寺尾四丁目25番1号	被災した住民の避難生活の場所、情報受任			
13	東台小学校	鶴見区	東寺尾東台12番地1	神奈川県横浜市鶴見区東寺尾東台12番地1	被災した住民の避難生活の場所、情報受任			
14	岸谷小学校	鶴見区	岸谷一丁目6番1号	神奈川県横浜市鶴見区岸谷一丁目6番1号	被災した住民の避難生活の場所、情報受任			
15	矢向小学校	鶴見区	矢向三丁目8番1号	神奈川県横浜市鶴見区矢向三丁目8番1号	被災した住民の避難生活の場所、情報受任			
16	入船小学校	鶴見区	浜町1丁目1番地1	神奈川県横浜市鶴見区浜町1丁目1番地1	被災した住民の避難生活の場所、情報受任			
17	寺尾小学校	鶴見区	東寺尾五丁目19番1号	神奈川県横浜市鶴見区東寺尾五丁目19番1号	被災した住民の避難生活の場所、情報受任			
18	汐入小学校	鶴見区	汐入町2丁目36番地	神奈川県横浜市鶴見区汐入町2丁目36番地	被災した住民の避難生活の場所、情報受任			
19	馬場小学校	鶴見区	馬場七丁目20番1号	神奈川県横浜市鶴見区馬場七丁目20番1号	被災した住民の避難生活の場所、情報受任			
20	駒岡小学校	鶴見区	駒岡三丁目14番1号	神奈川県横浜市鶴見区駒岡三丁目14番1号	被災した住民の避難生活の場所、情報受任			
21	獅子ヶ谷小学校	鶴見区	獅子ヶ谷一丁目19番1号	神奈川県横浜市鶴見区獅子ヶ谷一丁目19番1号	被災した住民の避難生活の場所、情報受任			
22	上寺尾小学校	鶴見区	馬場三丁目21番21号	神奈川県横浜市鶴見区馬場三丁目21番21号	被災した住民の避難生活の場所、情報受任			
23	新鶴見小学校	鶴見区	江ヶ崎町2番地1	神奈川県横浜市鶴見区江ヶ崎町2番地1	被災した住民の避難生活の場所、情報受任			
24	市場中学校	鶴見区	市場下町1番地1	神奈川県横浜市鶴見区市場下町1番地1	被災した住民の避難生活の場所、情報受任			
25	午向中学校	鶴見区	午向一丁目8番24号	神奈川県横浜市鶴見区午向一丁目8番24号	被災した住民の避難生活の場所、情報受任			

- 住所を緯度経度に変換する
 - <http://newspat.csis.u-tokyo.ac.jp/geocode/>
 - 「今すぐサービスを利用する」をクリック
 - 各パラメータを設定
 - 住所を含むカラム番号: 4
 - 変換したいファイル名: shelter.csv
 - 「送信」を押すと、変換結果のCSVが落ちてくる
 - ダウンロード後、Excelで開いて内容を確認
 - 取得した緯度経度をGoogleMapsで確認
 - 「fy fx」の順で検索

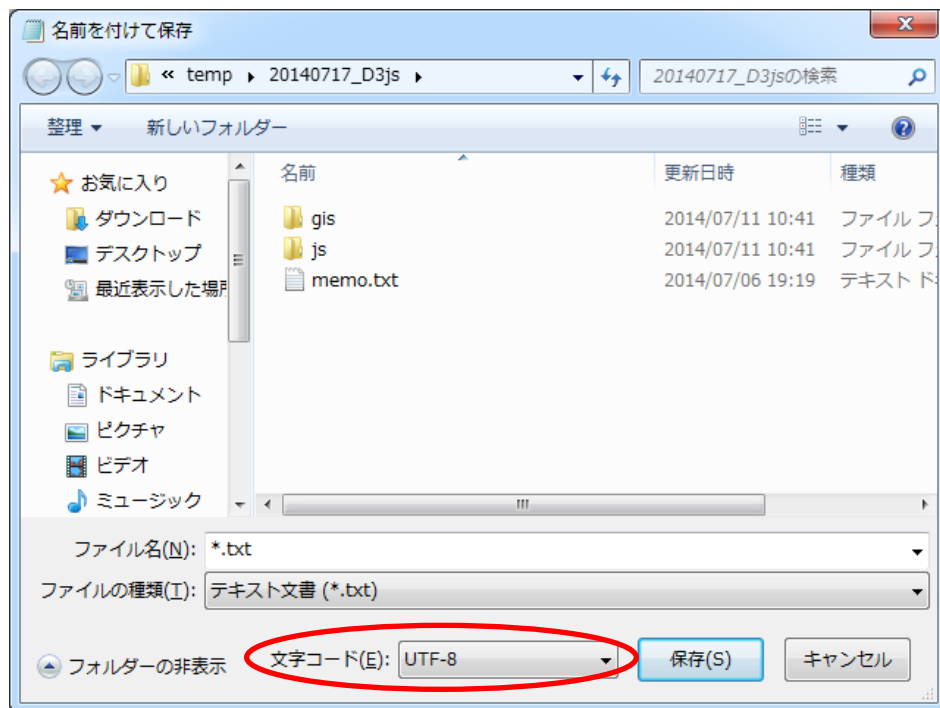


The screenshot shows the 'CSV Geocoding Service' web page. The title is 'CSVアドレスマッチングサービス'. Below the title, it says 'Geocoding service for CSV formatted file on WWW, powered by SPAT'. The main form is titled 'パラメータ設定' (Parameter Setting). It contains several input fields and dropdown menus: '対象範囲' (Target Range) is set to '全国市区レベル(経緯度・世界測地系)'; '住所を含むカラム番号' (Column number containing address) is set to '4'; '入力ファイルの漢字コード' (Character code of input file) is set to '自動設定' (Automatic setting); '出力ファイルの漢字コード' (Character code of output file) is set to '入力ファイルと同じ' (Same as input file); 'マッチングオプション' (Matching options) has a checkbox for 'x,yを反転' (Reverse x,y) which is unchecked, and a dropdown for '部分一致を' (Partial match) set to '探す' (Search); '変換したいファイル名' (File name to be converted) is set to 'shelter.csv'. At the bottom of the form are buttons for '送信' (Send) and 'クリア' (Clear). A link '説明に戻る' (Return to explanation) is at the bottom left.

データ準備－3

- 文字コード変換

- テキストエディタで開き、文字コードを「UTF-8」に変更し保存
- ファイル名は、「shelter_utf8.csv」を指定



Windowsのメモ帳
「ファイル」→「名前を付けて保存」



Macのテキストエディット
「テキストエディット」→「環境設定」

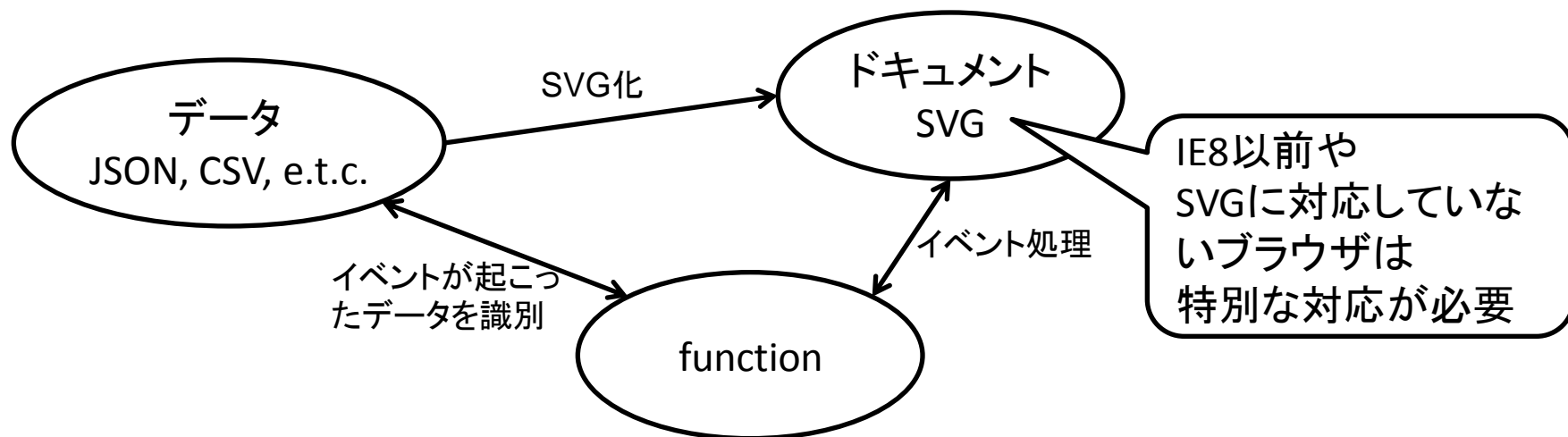
- 他のデータに対しても、同様の手順を行う
 - －「住所を含むカラム番号」に注意
- 改行が正しく表示されない場合
 - －住所→緯度経度変換時に、UNIX改行に変わった
 - －心配なら、別のエディタでCR+LFに変換
 - 変換しなくても、特に問題はない

D3.jsで可視化しよう

D3.js (またはD3:Data-Driven Documents、旧:Protovis^[1]) は、2011年に開発が始まった^[2][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。^[3]

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
 - SVGを効率良く生成するためのライブラリです

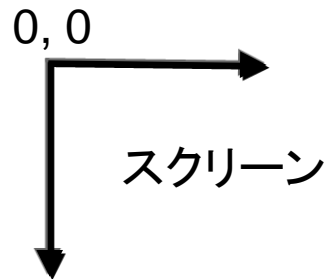


- 今回は地図を軸に可視化してみる
 - D3.js は、地図描画の機能が充実
- 注意：地図は楽だけど、グラフを描くのは意外に面倒
 - 必要な部品がそろっているだけ
 - D3.js Examples → 各グラフを描くためのソースを参照
 - <https://github.com/mbostock/d3/wiki/Gallery>
 - 参考URL <http://postd.cc/what-d3js-is-not/>
- 参考書籍
 - エンジニアのための データ可視化[実践]入門
～D3.jsによるWebの可視化
 - <http://www.amazon.co.jp/dp/4774163260>
 - D3.js はほとんど出てこないけど、「可視化」の理解が深まります

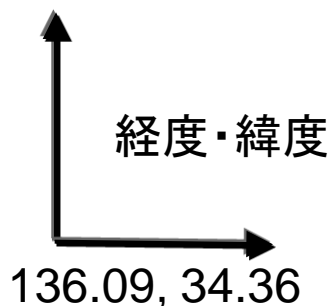
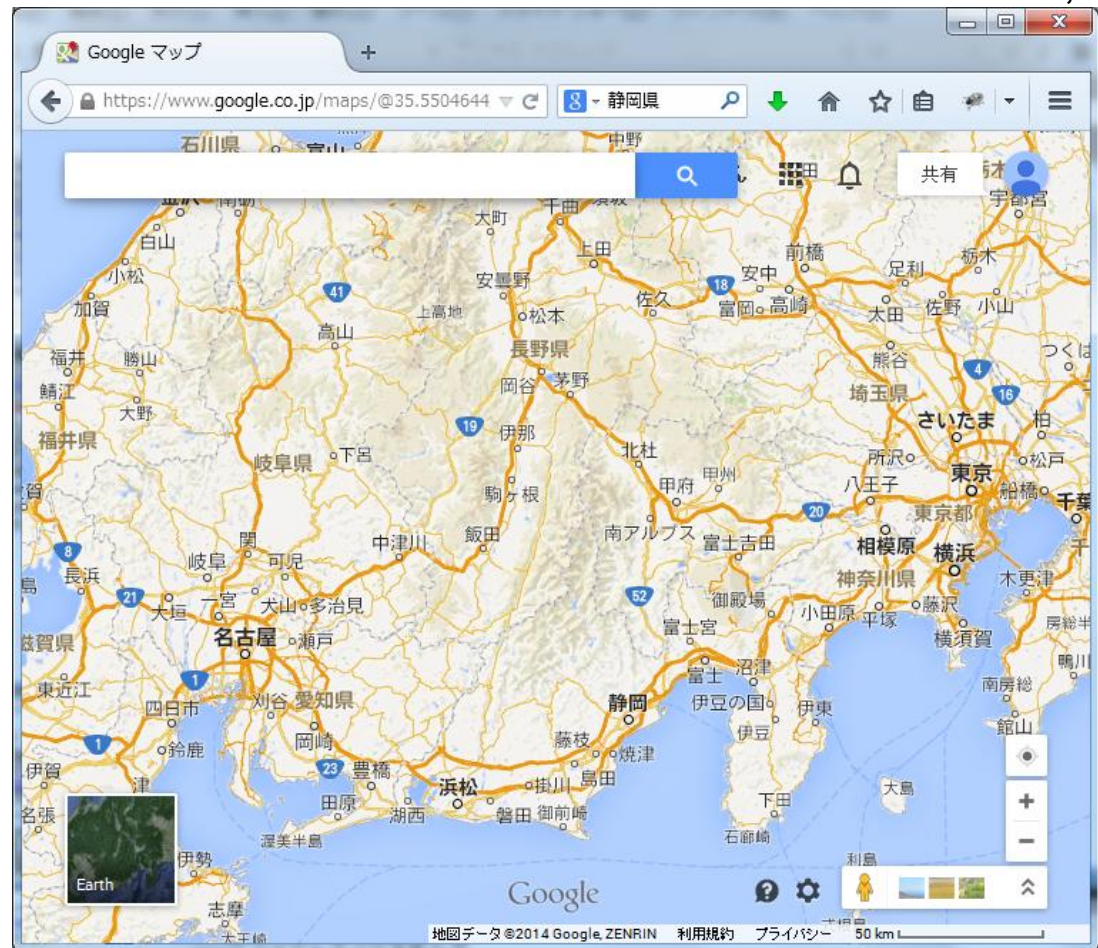
緯度・経度を扱う上での注意

0 , 0
136.09, 36.77

900 , 0
140.18, 36.77



画面内に描画するため
・スクリーンの原点と、緯度・経度の原点が違う
・表示スケールの調整



0 , 600
136.09, 34.36

900 , 600
140.18, 34.36

開発環境の準備

- 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
 - 今回は、すでに入っています
- テキストエディタ
 - お気に入りのエディタでOK
- ブラウザ
 - [F-12]を押せば、開発ツールが起動する
 - Chrome: 「ツール」→「デベロッパーツール」
 - ローカルデータファイルを参照するためには、起動オプションを追加
「--allow-file-access-from-files」
 - Firefox + firebug: 「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」
 - 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
 - データにアクセスしている部分をjQueryにすれば、動くらしい

- 地図データ

- D3.js の geo パッケージで読める形式が必要

- 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形式の地図を入手し、GeoJSON形式に変換

- 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>
 - 神奈川県: N03-130401_14_GML.zip

- このサイトで10%に圧縮 + GeoJSONに変換

- <http://www.mapshaper.org/>
 - 10%というのはShapefileでの比 (1,191KB → 131KB)
 - Shapefile → GeoJSONに変換すると大きくなる (131KB → 514KB)

- 変換結果 (GeoJSON形式、514KB)

- http://cloud.aitc.jp/20140717_D3js/gis/kanagawa.json

- セクタ(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

```
signal = [  
  { "cx": 100, "cy": 100, "color": "#0000ff", "title": "青", },  
  { "cx": 200, "cy": 100, "color": "#ffff00", "title": "黄", },  
  { "cx": 300, "cy": 100, "color": "#ff0000", "title": "赤", },  
];
```

- セクション

- selectAll(), enter(), exit()
- 繰り返し処理が楽に書ける

- d3.select("#TEXT1").selectAll("p").style("color", "#000000");

- 動的プロパティ

- svg.selectAll(".node").data(signal).text(function(d) { return d.title; });
- svg.selectAll(".node").data(signal).text(function(d, i) { return i; });

- データの結合

- 更新: svg.selectAll(".node").data(signal).text("hogehoge");
- 追加: svg.selectAll(".node").data(signal).enter().append("text").text("piyopiyo");
- 削除: svg.selectAll(".node").data(signal).exit().remove();

- アニメーション

- d3.select("#TEXT4").transition().style("background-color", "black");

その他の今回使用するメソッド

- ファイルの読み込み
 - d3.json(), d3.text(), d3.csv()
- 地図描画
 - d3.geo.mercator()
 - d3.geo.path().projection(projection);

SVG (Scalable Vector Graphics) について

- Wikipedia

- http://ja.wikipedia.org/wiki/Scalable_Vector_Graphics

Scalable Vector Graphics (スケーラブル・ベクター・グラフィックス、**SVG**) は、[XML](#)をベースとした、2次元[ベクターイメージ](#)用の[画像形式](#)の1つである。アニメーションやユーザインタラクションもサポートしている。SVGの仕様は[W3C](#)によって開発され、[オープン標準](#)として勧告されている。

- SVG仕様

- http://www.hcn.zaq.ne.jp/___/SVG11-2nd/

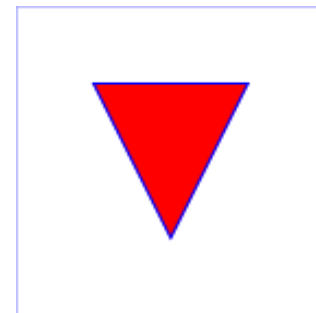
- 使用上の注意

- 順番通りに上に重ねて描画
- 対応していないブラウザだと、何も表示されない
- ブラウザによっては、微妙に見え方が違う

- 代表的な図形

- パス

```
<path d="M 100 100 L 300 100 L 200 300 z" fill="red" stroke="blue" stroke-width="3" />
```



- 円

```
<circle cx="600" cy="200" r="100" fill="red" stroke="blue" stroke-width="10" />
```

- 線

```
<line x1="100" y1="300" x2="300" y2="100" stroke="green" stroke-width="5" />
```

- テキスト

```
<text x="250" y="150" font-family="Verdana" font-size="55" fill="blue" >
```

Hello, out there

```
</text>
```

以降は、サンプルコードを見ながら解説

- http://cloud.aitc.jp/20140717_D3js/
- D3.jsでデータとSVGが連動するサンプル
 - step1_1.html + js, step1_2.html + js
- D3.jsで地図を描画する
 - step2.html + js
- D3.jsでデータを取得する
 - step3.html + js
- 取得したデータをsvgで可視化する
 - step4.html + js
 - 十分な量のデータがあれば、地図無しでもいいけるかも
- 地図に重ねてデータを描画する
 - step5.html + js 金沢区内公園を描画する

step5.js を改造し、「地域防災拠点」に対応

```
35 ////////////////////////////////////////////////////
36 // 金沢区内公園 http://www.city.yokohama.lg.jp/kanazawa/kz-opendata/
37 var csvfile = "./23-kz-park_utf8.csv";
38 d3.csv(csvfile, function(csv) {
39     points = csv;
40     svg = d3.select("svg"); // <svg>を選択
41
42     svg.selectAll(".node") // <circle class="node">を選択
43     .data(points).enter() // データの増分を対象
44     .append("circle") // svgのcircleを追加
45     .attr("class", "node")
46     .attr("cx", function(d) { return projection([d.経度, d.緯度])[0]; }) // 円の中心
47     .attr("cy", function(d) { return projection([d.経度, d.緯度])[1]; }) // 円の中心
48     .attr("r", function(d) { return 5; }) // 円の半径
49     .style("stroke", function(d) { return "#000000"; }) // 枠線の色
50     .style("stroke-width", "1px") // 枠線の太さ
51     .style("fill", function(d) { // 塗りつぶしの色
52         if (d.名称.indexOf("公園")>=0){
53             return "white";
54         } else {
55             return "red";
56         }
57     })
58     .on('click', function(d) {
59         console.log(d);
60         address = [{Name:d.名称, 緯度:d.緯度, 経度:d.経度, }];
61         svg.selectAll(".address")
62         .data(address)
63         .attr("x",function(d) { return projection([d.経度, d.緯度])[0]; })
64         .attr("y",function(d) { return projection([d.経度, d.緯度])[1]; })
65         .text(function(d) { return d.Name; })
66         ;
67     });
68 ;
69
70 // 住所を表示する入れ物を作っておく
71 address = [{Name:"ダミー", }];
72 svg.selectAll(".address")
73 .data(address).enter()
74 .append("text")
75 .attr("class", "address")
76 .attr("x",function(d) { return 0; })
77 .attr("y",function(d) { return 0; })
78 .style("font", "8pt sans-serif")
79 .text(function(d) { return d.Name; })
80 ;
81 });
82 });
83 }
84
```

ファイル名を「./shelter_utf8.csv」にし、
csvファイルと同ディレクトリにコピー

csvのタイトルに合わせる

とりあえず固定で「white」を返す

csvのタイトルに合わせる

- Nameによって、色を変える
 - 小学校、中学校、その他
- 他の情報も合わせて表示
 - 津波非難施設、応急給水拠点、帰宅困難者一時滞在施設
- アニメーション
 - D3.js の更新＋アニメーション機能
 - .transition().duration(ミリ秒).変化後のスタイル()
 - JavaScriptのsetInterval()
 - 静岡県雨量を使った例
 - http://cloud.aitc.jp/20140627_D3js/sample6.html
- 自分の現在位置を表示
 - HTML5のGeolocation API

- 付加情報を表示
 - clickかmouseoverで、詳細情報を表示
 - 地名・住所・関連情報など
 - 文字をもっと見易くする(白抜き文字、など)
 - <http://www.slideshare.net/kadoppe/inline-svg/53>
- 他のデータも合わせて表示
 - 他のオープンデータ
 - 社内の業務データなど
- アニメーション
 - 「もう1つ軸を足したい」時など
- 拡大／縮小
 - 特定の区を詳細に見たい
 - 倍率に合わせて、円の大きさや文字サイズを調整

- 時間があれば、紹介
 - 「地域防災拠点」を使った実装例
 - step5example.js
 - 横浜市統計情報ポータルサイトのデータを使った例
 - <http://www.city.yokohama.lg.jp/ex/stat/index2.html#24>
 - 「テレビ放送受信契約数」平成18年から24年の推移
 - ライセンス
 - <http://www.city.yokohama.lg.jp/front/aboutweb.html>
 - > 私的使用のための複製や引用など著作権法上認められた場合を除き、無断で複製・転用をすることはできません。
 - 横浜市に勉強会での使用について問い合わせた結果
 - > 「横浜市統計書」からの出典を明らかにしていただければ、
 - > 著作権法上認められた場合に該当すると判断します。