情報整理のための Google Map API入門

日紫喜　光良

プロジェクトI/II

2016.9.23

今日の目標

- Google Map APIを用いて、Google Mapの地図上にマーカーを表示する
  - HTML
    - Webページの構造を宣言する。
      - 地図を表示する場所を宣言する。
  - Javascriptプログラミング
    - Webページの地図表示箇所上に、地図を描く。
    - マーカーオブジェクトを生成して、地図上に置く。
    - 複数のマーカーの位置データをJSON形式のテキストファイルから読み込む。
    - 地図上に、複数のマーカーオブジェクトを置く。
    - マーカーに吹き出し（インフォウィンドウ）をつける。
- JSON形式によるデータの表現方法を学ぶ
  - Key-value pair（“項目名”:”その値”）の集合
- データ収集から表示形式までの効率的な方法を考える。
  - 住所から緯度と経度を得る
  - 衛星写真上で位置を微調整する
  - JSON形式へのデータ変換

必要なもの

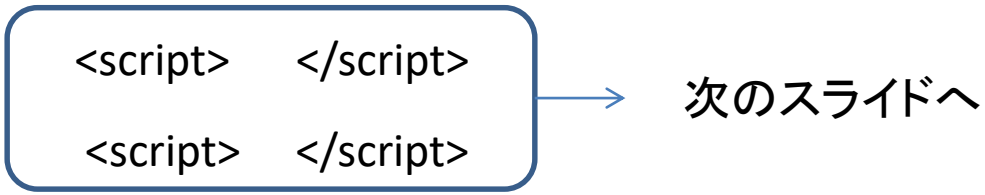
- ブラウザ（Firefoxを推奨）
  - PC上のJSONファイルをブラウザ上で動くJavascriptプログラムに読み込ませるので、Firefoxを推奨する。Chromeではこれができない。
  - Developer toolを使うと、デバッグに便利
- テキストエディタ（秀丸, mi等）

注意

- 次以降のスライドでのサンプルコードをそのままコピペすると、全角文字（具体的には引用符）のためにエラーになる場合があります。
  - それに注意してコピペしてください。

HTMLでWebページの構造を作る

```
<!DOCTYPE html>
<meta charset="UTF-8" />
 <html>
 <head>
```




```
 <style>
 #map {width: 500px; height: 380px; border: 1px solid #666; float: left;}
 </style>
 </head>
  <body>
  <div id="map"></div>
 </body>
 </html>
```

Javascriptで地図を表示する(1)

```
<head>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false&lan
guage=ja"></script>
  <script>
```

次のスライド

```
  </script>
 <style>
 #map {width: 500px; height: 380px; border: 1px solid #666;
 float: left;}
 </style>
</head>
```

Javascriptで地図を表示する(2)

```
var c_lat, c_lng;
var map;

c_lat  = 35.1814;
c_lng =  136.9064;
window.onload  =  function(){
        draw_map(c_lat, c_lng);
}
function draw_map(lat,lng){
   var op = {
      zoom:13,
      center:new google.maps.LatLng(lat,lng),
      mapTypeId:google.maps.MapTypeId.ROADMAP
   };
   map = new
google.maps.Map(document.getElementById("map"),op);
}
```

地図の上にマーカーを置く(1)

<script type="text/javascript“ src="http://maps.google.com/maps/api/js?sensor=false&language=ja"></script>

```
<script>
var c_lat, c_lng, t_lat, t_lng;
var map;
var marker;
c_lat  = 35.1814;
c_lng =  136.9064;
t_lat = 35.1584;
t_lng = 136.9217;
 window.onload =  function(){
            draw_map(c_lat, c_lng);
            placeMarker(map, t_lat, t_lng);
 }
 function draw_map(lat,lng){ … }
 function placeMarker(map, lat, lng){ … }

<script>
```

地図の上にマーカーを置く(2)

```
function placeMarker(map, lat, lng){
   var latLng = new google.maps.LatLng(lat, lng);
   var label = "Nagoya University Hospital";
   marker = new google.maps.Marker({
      position:latLng,map:map,title:label
    });
}
```

マーカーのその他のプロパティの例

- icon
  - １．マーカーアイコンを変更するために用いる
    - URL（または相対パス）を指定すれば、画像がそのまま出る。
    - google.maps.Icon()を用いると、マーカーのサイズや伸縮性を指定できる。
  - ２．マーカーアイコンの種類を変更するためにさらにiconのプロパティを用いる
    - 例えば、path:google.maps.SymbolPath.CIRCLE, scale:数式で、マーカーの数値に応じた直径の円盤を表示できる。

Javascriptを構成するもの

- オブジェクト（object）
  - プロパティ（property）をもつ。
  - ”プロパティ”:”値”をコンマ（,）で結合 ⟶ JSON形式が踏襲
- 変数（variable）
  - 変数のスコープ（scope）
- 演算子（operator）
  - 例：代入（assignment）記号（=）、等号（==）
- 式（expression）
- 文（statement）
  - 最後にセミコロン
  - 制御フローを表現するために、if (条件式) {…} else {…}
- 関数（function）
  - 引数（argument）、返り値（return value）
  - 定義するには　function *function_name*(*argments*){*statements*}
  - 呼び出すには　*function_name*(*arguments*);
  - 無名関数　function(){ … }

Javascriptプログラムを書くために

- 必要な定数、変数を先に定義しておく。
- window.onloadプロパティに、Webページを表示し終わった後の動作を記述する。
  - 動作の記述には、関数を用いる。
- プログラムを関数に分けて書くと、読みやすい。
- 変数は、varをつけて定義すれば、スコープは関数の中だけ。つけなければ、プログラム全体。

まとめ：Google Map APIを使うには（１）

- まずHTMLでページの構造を作る。
  - とくに、地図を載せる 要素のid属性と、スタイルの定義が必要。
- <head>要素の中の<script>要素に、地図を描くためのJavascriptプログラムを書く。
  - 別の<script>要素に、Google Maps APIのプログラム本体（ライブラリ）が置いてある場所のURLを書く必要がある。
- まず地図を描く。その上にマーカーを置く。
- 地図は、google.maps.Mapで描く。
- google.maps.Mapは地図を乗せる要素とオプションを要求する。
  - 要素はdocument.getElementById(要素名)で与える。
  - オプションには、最低限**zoom**, **center**, **mapTypeId**を。
- マーカーはgoogle.maps.Markerで作る。
  - **position**: google.maps.LatLng(緯度,経度)で定義されるオブジェクト
  - **map**: 地図オブジェクト
  - **title**: 文字列

マーカー情報をJSONで表現

JSON: Javascript Object Notation

```
{“hospital”:[
  {“name”:”Nagoya University Hospital”,
    ”lat”:35.1584, ”lng”:136.9217},
  {“name”:”Chukyo Hospital”,
    ”lat”:35.1100, “lng”:136.9020},
  {“name”:”Nagoyadaiichi Red Cross Hospital”,
    ”lat”:35.1718, ”lng”:136.8624}
]}
```

UTF-8形式で保存

hosp.json

ひとつのオブジェクトは
{“プロパティ1”:”値1”,”プロパティ2”:”値2”,…,”プロパティN”:”値N”}

オブジェクトの配列は、
[オブジェクト1, オブジェクト2, … , オブジェクトM]

JSON形式のファイルを読み込む

```
<script type="text/javascript“ src="http://maps.google.com/maps/api/js?sensor=false&language=ja">
</script>
<script type='text/javascript'
      src='http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js'>
</script>
 <script>
  var c_lat, c_lng, t_lat, t_lng; ...   ;t_lng = 136.9217;
  var hosp_data;
  $.getJSON(“hosp.json", function(data){
        hosp_data = data;
  });
 window.onload =  function(){
        draw_map(c_lat, c_lng);
        plot_sites(hosp_data);
 }

 function draw_map(lat,lng){ … }
 function plot_sites(data){ … }
 <script>
```

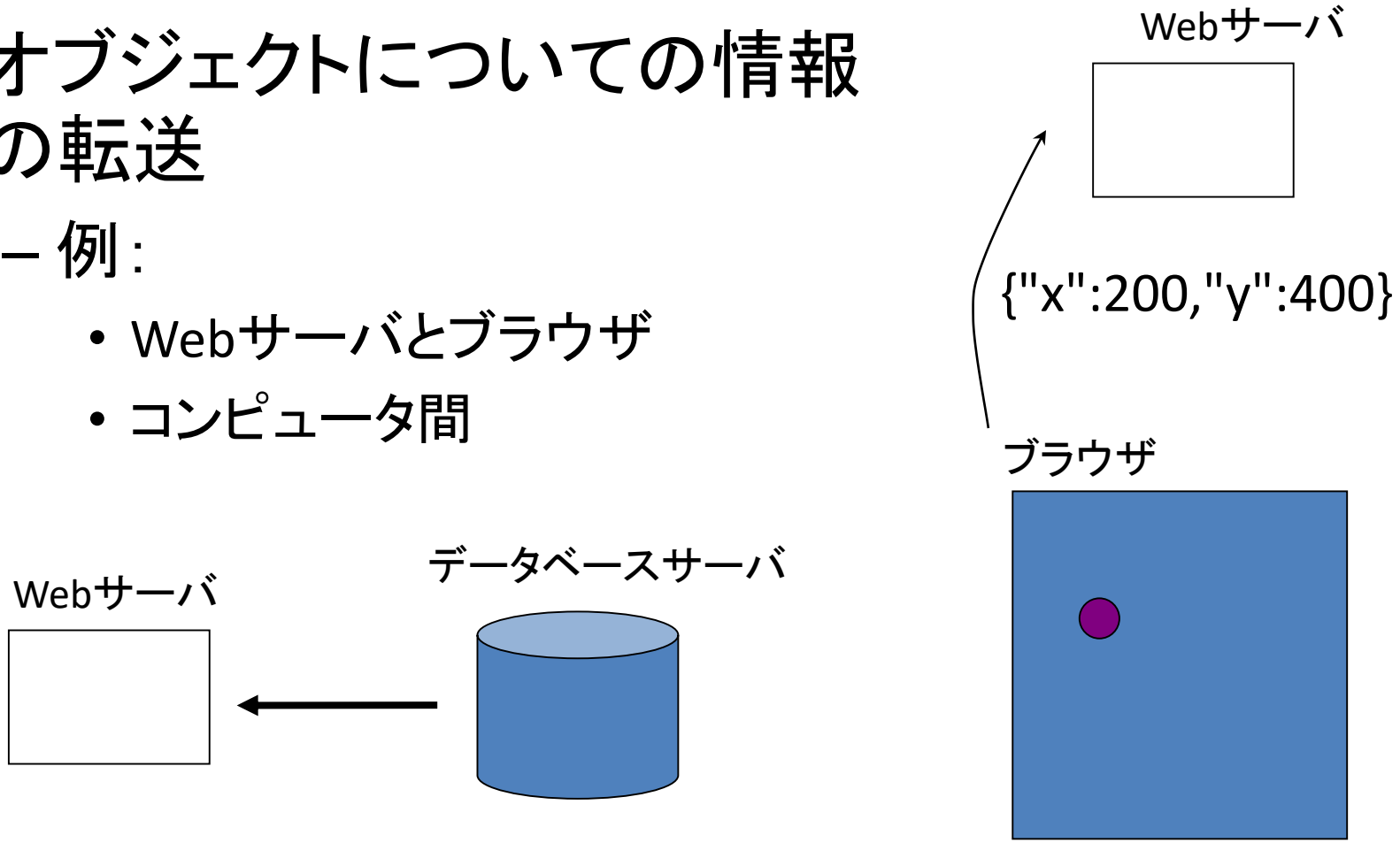
次スライドへ

複数のマーカーを表示する

```
function plot_sites(data){
    var len = data.hospital.length;
    for (var i=0; i < len; i=(i+1)){
      var site = data.hospital[i];
      var lat = site.lat;
      var lng = site.lng;
      var name = site.name;
      var pos = new google.maps.LatLng(lat,lng);
      var marker_prop  =  {position:pos, map:map, title:name};
      var marker = new google.maps.Marker(marker_prop);
   }
}
```

JSONの目的

- オブジェクトについての情報の転送
  - 例:
    - Webサーバとブラウザ
    - コンピュータ間



{"id":"123456","name":"pochi", "species":"dog"}

JSONの特徴（１）

- テキストで書かれている
  - 人が読める
- １つのオブジェクトは{}で囲まれている
- 1つのオブジェクトは、プロパティと値のペアを1つ以上含む。
  - 複数のペアはコンマ（,）で並べる
- 値の型はさまざまなものが可能
  - 文字列、数値、配列、オブジェクト
- プロパティも値もダブルクォーテーション（"）で囲む。
  - ここが、単なるJavascriptのオブジェクトの表現と異なる。
  - 値が数値のときは、ダブルクォーテーションで囲まない
  - 値が配列やオブジェクトのときなども、囲まない。

JSON形式で構造をもつデータを表現できる

```
{ "pets":[

    {"id":“01","name":"pochi", "species":"dog",
   "favorites":{"food":"meat","place":"garden"}},

    {"id":“02","name":"tama", "species":"cat",
   "favorites":{"food":"tuna","place":"kitchen"}},

 ]}
```

マーカーに吹き出しをつける

```
{“hospital”:[
  {“name”:”Nagoya University Hospital”,
    ”lat”:35.1584, ”lng”:136.9217, ”ward”:”Showa”,
    ”url”:” http://www.med.nagoya-u.ac.jp/hospital/”},
  {“name”:”Chukyo Hospital”,
    ”lat”:35.1100, “lng”:136.9020, “ward”:”Minami”,
    “url”:”http://chukyo.jcho.go.jp”},
  {“name”:”Nagoyadaiichi Red Cross Hospital”,
    ”lat”:35.1718, ”lng”:136.8624, “ward”:”Nakamura”,
    ”url”:”http://www.nagoya-1st.jrc.or.jp/”}
]}
```

マーカーに対して吹き出しを付ける

```
function plot_sites(data){
    var len = data.hospital.length;
    for (var i=0; i < len; i=(i+1)){
      var site = data.hospital[i];
      var lat = site.lat;
      var lng = site.lng;
      var name = site.name;
      var url = site.url;
      var pos = new google.maps.LatLng(lat,lng);
      var marker_prop  =  {position:pos, map:map, title:name};
      var marker = new google.maps.Marker(marker_prop);
      marker = add_InfoWindow(marker,url);
  }
}
```

吹き出しのイベントドリブンな動作を定義する

```
function add_InfoWindow(marker,url){
    var infowindow = new google.maps.InfoWindow();
    google.maps.event.addListener(marker, 'click', function() {
        infowindow.setContent(url);
        infowindow.open(map,marker);
    });
    google.maps.event.addListener(map, 'click', function() {
        infowindow.close();
    });
  return marker;
}
```

まとめ：Google Map APIを使うには（2）

- JSON形式のファイルを読み込むには、jQueryライブラリのgetJSONを使うと簡単である。
  - ファイルのURLと、読み込んだ後の動作を与える。
- 吹き出しは、google.maps.InfoWindowで作る。
  - setContent, open, close等のメソッドはよく使う。
- 吹き出しの出没は、マーカーや地図のイベント（クリック等）によって起動される。
  - google.map.event.addListner
  - どのオブジェクトのイベントか、イベントの種類、イベントが発生させる動作

地理データの取得

課題：　住所を緯度・経度に変換する。
　　より正確な微調整を行なう（例：〇棟⇒◇棟）。

ツールを開発した。

http://hishikilab.sakura.ne.jp/project2013/adrs2pos_02.html

- 入力した住所を、緯度・経度に変換（google.maps.Geocoderを利用）
- その場所を中心とする衛星写真を表示、そこにマーカーを立てる。
- マーカーを移動してボタンを押すと、新しい場所を書き出す。
- エクセルにコピー＆ペーストして記号で区切る

JSONデータの作成

例：

エクセル（CSV）データ

テキストエディタで開く

変換用Webページにコピー＆ペーストして、変換

テキストエディタでJSON形式のファイルに追加