

プロジェクト I / I I テーマ H (担当: 日紫喜光良)

目標

- 地図上に情報を表示することが解決に役立つような問題を見つけることができる。
- 問題解決のために必要な情報をグループメンバーと協力して収集できる。
- Google の地図 (Google Maps) 上に情報を表示することができる。
- 地理情報と組み合わせた情報管理のために、JSON形式のテキストファイルや Google Fusion Table を用いることができる。

作業環境: テキストエディタ、ならびに、HTML 5 対応でローカルコンピュータ上のファイルを読み込み可能な Web ブラウザ (例えば Firefox。Chrome は不可。)。必要に応じて、公開のための Web サイト (今回の実習ではなくてもできる)。

このテキストの使い方

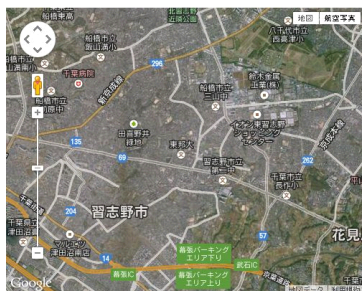
1. 最初からひとつおとり、サンプルプログラムを読みこめば、地図上で情報を管理するための基礎的な方法が習得できるはずである。
2. 自分のやりたいことを実現する方法を探すために: 「目的別のヒント」のタイトルと最初に提示された図をざっと眺めて、やりたいことにもっとも近いセクションを見つける。参照しているプログラムがあれば、それをよく読んで、自分の目的のために使えるか、使えたとすればどこをどう修正したらよいか考える。

このテキストが扱っていない重要関連事項の例

- A) マーカーの形をカスタマイズすること。例えば、パン屋の位置にはパンの形をしたマーカーを表示したい。
- B) 地理情報の収集作業の際に、情報を Web 上で入力できるようにする方法。
- C) エクセルや Google Spreadsheet の情報を JSON形式のテキストに変換する方法。
- D) JSONデータをテキストファイルではなくデータベースで保持する方法。
- E) ポリゴンデータの扱い: ダウンロード、Google Fusion Table への取り込みなど
- F) Google Fusion Table データを Google Map 地図の上に重ねる。

目的別のヒント

1. ある地点を中心とする地図を表示する。



ファイル名 : case_0.html

要点1. 表示される地図の定義は、<head>...</head> (ヘダー) 内のスクリプトである initialize() 関数に書く (名前は何でもいいが、多くの例は initialize() を使っている)。ブラウザがHTML文書を読み込む際に、この関数を起動するようにする。

```
<head>
...
<script>
    function initialize() {...var map=...;}
</script>
</head>
<body onload="initialize()">...</body>
```

要点2. 表示するためには、表示する場所が必要→body要素の中で、div要素を定義。id属性として名前が必要。

```
<div id="map_canvas" style="width:500px; height:400px"></div>
```

要点3. 地図オブジェクトの生成には、クラス google.maps.Map を用いる。

クラスを具体化するためには、(a) 画面上の表示場所と (b) 表示される区域の地理情報が必要

```
var myOptions = {
    zoom: 13,
    center: myLatLng,
    mapTypeId: google.maps.MapTypeId.HYBRID
}
var map = new
google.maps.Map(document.getElementById("map_canvas"),
myOptions);
```

その他の重要事項

1). 小文字で始まる単語 : オブジェクト、変数、プロパティ (属性)、関数、メソッド、またはパッケージ

```
オブジェクトの例 : document
変数の例 : myLatLng, myOptions, map
プロパティの例 : zoom, center, mapTypeId
関数の例 : initialize() {...,}
メソッドの例 : document.getElementById
パッケージの例 : google
```

2). 大文字で始まる単語：クラス

例：`google.maps.Map`, `google.maps.LatLng`

3). HTML ファイルの構造

`<!DOCTYPE html>` 文書型宣言

`<html> ...</html>` HTML 文書の開始・終了タグ

`<head> ...</head>` ヘッダ情報

`<body> ...</body>` ボディ情報

`<head>`要素と`<body>`要素から HTML 文書が構成される。(→右囲み「HTML ファイルの構造」)

HTML ファイルの構造

```
<!DOCTYPE html>
<html>
  <head> ヘッダ情報 </head>
  <body> ボディ情報 </body>
</html>
```

ヘッダに含まれるもの

A). `<meta 属性="">` 文書全体の属性を設定するためのタグ

属性設定の例：

```
<meta charset="UTF-8" />
//文字コードとして Unicode (UTF-8)を使用する。
```

B). `<script>...</script>`タグで囲まれた、Javascript のプログラムへのリンクまたはプログラム本体。

ライブラリへのリンクは必須。

例：Google Maps API へのリンク

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

ボディにはいくつかの`<div>...</div>`要素が含まれる。

参考：

Google Map を表示する - Google Maps Javascript API V3 ("Waste Of Time"
<http://www6.ocn.ne.jp/~wot/web/gmap/view01.html>)

2. ある地点を中心とする地図の上に、一つマーカを表示する。



ファイル名 : case_1_mod_3.html

概要 : 緯度と経度で定義される地図上の1点に、マーカを1個表示する。マーカにマウスを重ねる(マウスオーバーする)と名前が表示される。マーカをクリックするとウィンドウがポップアップして情報を表示する。

要点1. マーカの定義には、position プロパティ(属性)、map プロパティ、title プロパティが必要。position プロパティはマーカの座標(緯度と経度)を、map プロパティはマーカを乗せるマップオブジェクトを、そして、title プロパティはマーカの上にマウスを乗せた時に表示するラベル、を表す。

例 : myOptions で定義された map を div 要素 mapa に表示し、その上の座標 point に、マウスオーバーすると label を表示するマーカ marker を生成する。

```
var myOptions = {
  zoom: 13,
  center: new google.maps.LatLng(35.692781, 140.049514),
  mapTypeId: google.maps.MapTypeId.ROADMAP
};
map = new google.maps.Map(document.getElementById('mapa'),
myOptions);
var point = new google.maps.LatLng(35.692781, 140.049514);
var label = 'メディアセンター';
...(中略)
var marker = new google.maps.Marker({position: point, map: map,
title: label});
```

要点2. オブジェクト(例: マップやマーカ)にイベント(例: マウスを乗せたりクリックしたりすると情報を表示する)を追加するためには、google.maps.event.addListener メソッドを用いる。

例1 : マーカをクリックすると吹き出し(infowindow オブジェクト)を開く。

```
google.maps.event.addListener(marker, 'click', function() {
    infowindow.setContent(html);
    infowindow.open(map, marker);
});
```

例 2 : 地図をクリックすると吹き出しを閉じる。

```
google.maps.event.addListener(map, 'click', function() {
    infowindow.close();
});
```

要点 2. マーカーの吹き出しを生成するには、`google.maps.InfoWindow` クラスを用いる。

例 : 吹き出しオブジェクト `infowindow` を生成する。

```
var infowindow = new google.maps.InfoWindow();
```

参考 : 1. と同じ。

3. 地図上に複数のマーカーを表示する。



case_2.html の解説

概要 : 1つのマーカーを表示する関数を繰り返し利用して複数のマーカーを表示する。

4. マーカー情報を J S O N形式の外部ファイルから読み込む

case_3_org2.html (利用する J S O Nファイルは map_2.json) の解説

概要 :

マーカー情報を記録した J S O N形式のテキストファイルを jQuery ライブラリを用いて同じサイト上から読み込み、マーカー情報オブジェクトの配列を作る。その配列から 1つずつマーカー情報を読み出し、マーカーを生成し、直ちに、吹き出しをマーカーに関連付けて生成する。

要点 1. jQuery を利用するためには、ヘッダーで jQuery へのライブラリへのリンクを指示する必要がある。

例：

```
<script type="text/javascript"
src="http://code.jquery.com/jquery-1.5.2.js"></script>
```

要点2. jQuery の関数は、接頭語\$をつけることで区別する。

例：次のコードでは、\$(function() {...})と\$.ajax({...})がjQuery の関数。

```
$(function() {
$.ajax({
...
success:function(json_data) {
var data=jsonRequest(json_data);
initialize(data);
}
});
});
```

要点3. \$.ajax 関数を用いてHTTP非同期通信を行う。このプログラムでは、同じサーバー上のファイルを読み込む。この関数は送受信が可能で、多くのオプションがある。

例：(要点4参照)

要点4. \$.ajax 関数の success オプションに対して、ファイルを読み込んだ後の処理を記述できる。initialize 関数はここで呼び出される。initialize 関数には引数として、マーカ情報 data が必要である。

```
$(function() {
// JSON ファイル読み込み開始
$.ajax({
url:"map_2.json",
cache:false,
dataType:"json",
success:function(json_data) {
var data=jsonRequest(json_data);
initialize(data);
}
});
});
```

要点5. 読み込んだJSONオブジェクト json_data を、マーカ情報の配列である data に入れなおす。

例 :

```
function jsonRequest(json_data) {
    var data=[];
    if(json_data.Marker) {
        var n=json_data.Marker.length;
        for(var i=0;i<n;i++){
            data.push(json_data.Marker[i]);
        }
    }
    return data;
}
```

要点6. for 文を用いて data から順番に情報を読み出し、google.maps.Marker クラスからマーカーを生成する。吹き出しの付加のために別の attachMessage 関数を生成する。この関数に marker を引数として与える必要がある。

while ループの中で、上で作成した createMarker 関数を呼び出しても動作しなかった。おそらく変数のスコープの問題と思われるがよくわからない。

例 :

```
var i=data.length;
while(i-- >0){
    var dat=data[i];
    var obj={
        position:new google.maps.LatLng(dat.lat,dat.lng),
        map:map,
        title:dat.content,
    };
    var marker=new google.maps.Marker(obj);
    attachMessage(marker,map,dat.url);
}
```

参考

1) jQuery.ajax のオプションの解説 :

<http://semooh.jp/jquery/api/ajax/jquery.ajax/options/>

2) JSON形式のテキストファイルのフォーマットと利用方法

<http://www.ajaxtower.jp/googlemaps/gdownloadurl/index4.html>

例 :

```
{"Marker": [
```

```

    {"lat":"35.691228",
    "lng":"140.020476",
    "content":"JR 津田沼駅",
    "url":"http://www.jreast-timetable.jp/timetable/list1009.html"},
    /
    {"lat":"35.685894",
    "lng":"140.047990",
    "content":"京成大久保駅",
    "url":"http://ekikara.jp/newdata/station/12216041.htm"},
    {"lat":"35.689310",
    "lng":"140.050200",
    "content":"大久保十字路",
    "url":"http://www.keiseibus.co.jp/jikoku/bs_tt.php?key=04068_0
    1a"}
  ]}

```

5. マーカーを直接クリックする代わりにページ上のリストのアイテムをクリックすると、対応するマーカーに吹き出しが表示される。



case_4_1_mod.html の解説。(JSON ファイルは map_3.json)

要点 1. ページの body 要素の中にサイドバーのための div 要素 side_bar が追加された。

```
<div id="side bar"></div>
```

要点 2. プログラム中でマーカーをクリックする動作を行わせるには、

google.maps.trigger 関数を用いる。

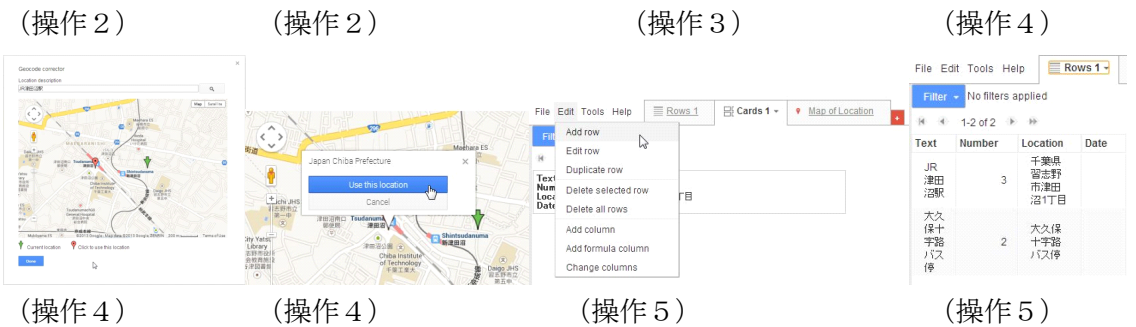
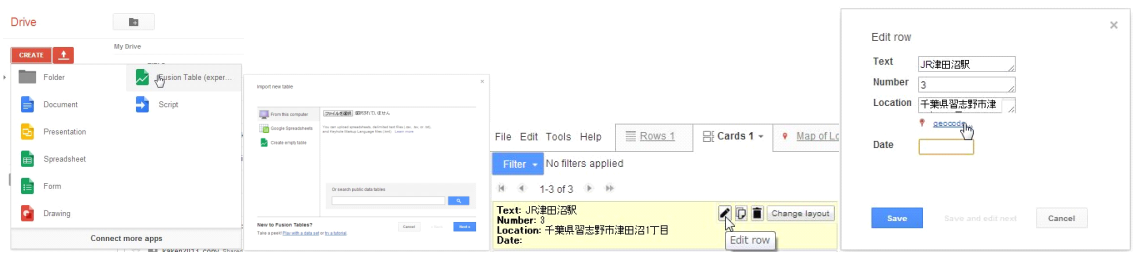
要点 3. クリックすると上記のマーカークリック操作をおこなうには、HTML でリンク先を Javascript のプログラムにすればよい。

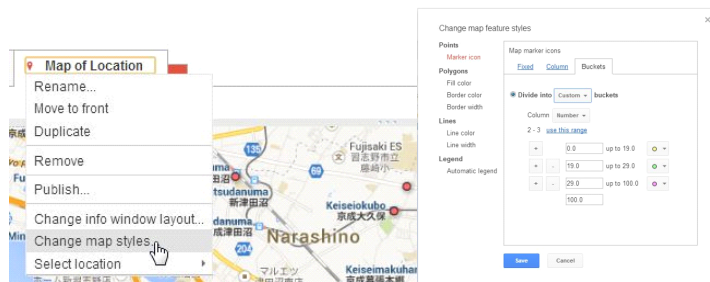
要点 4. 関数 initialize の中で呼び出される関数 myclick は、marker オブジェクトの配列 gmarkers を利用する。これを定義する位置をいろいろ試してみたが、関数 initialize の外で定義した場合だけうまく動作した。

プログラムのうち、要点2～4に該当する部分を示す。

```
function myclick(i) {
  google.maps.event.trigger(gmarkers[i], 'click');//要点2.
}
var gmarkers = [];//要点4.
function initialize(data){
  ...
  while (i<data_length){
    ...
    marker = new google.maps.marker(obj);
    gmarkers[i]=marker;
    side_bar +=
    '<a href="javascript:myclick('+i+')">'
    +dat.content+'</a><br>';
    //要点3.
    i++
  }
  ...
}
```

6. Google Fusion Tableにデータを住所とともに記録する





(操作 7)

(操作 7)

概要：Google Fusion Table は、スプレッドシートとしての機能に加えて、データの中の地理情報を用いてデータを地図上のマーカーとして表示することができる。地理情報として、緯度と経度のペアだけでなく、住所を利用できる。

準備：Google (Gmail) のアカウントを取得する。

操作 1. Google Drive を起動する。

操作 2. Google Fusion Table のテーブルを新規に作成するために、Create、次いで Fusion Table を選択する。テーブルを新規に作成する方法として 3 つのうちから "Create empty table" を選択して、テーブルを新規に作成する。

操作 3. Cards1 (名前は変更可能) のタブを選択してカードの形式にし、編集したいカードをクリックする。するといくつかアイコンが出現するので、その中から、鉛筆マークをクリックして編集する。最低でも、Text, Number, Location には入力する。

操作 4. Location に入力した住所と地図上の期待する位置とのズレを修正するために、まず、Geocoding リンクをクリックして地図を表示する。マーカーの位置がずれていれば、検索用テキストフィールドに別の地名を入力してマーカーを新たに表示し、選択し直す。

操作 5. 新たな行 (カード) を追加するために、まず、Rows1 (名前は変更可能) のタブを選択してテーブルの形式にし、メニューで Edit→Add row を選択して、新たな行を追加する。次に、操作 3, 4 によって、カードに情報を入力する。

操作 6. データを地図上に表示するために、"Map of Location" タブをクリックする。

操作 7. Number の値によってマーカーの色を変えたい。まず、"Map of Location" タブからメニューをプルダウンして、"Change map styles" を選ぶ。次に、"Bucket" タブをクリックして、数値の区切りと区切りの数を調節するためのパネルを開く。