

ソフトウェア開発 第 14 回目授業

平野 照比古

2015/11/16

Google Maps とは

Google Maps は Google が提供する地図サービスである。Google Maps JavaScript API のページ¹ には次のような記述がある。

Google Maps JavaScript API を使用すると、*Google マップ* をウェブページに埋め込むことができます。この *API* のバージョン 3 は、従来のパソコン用ブラウザ アプリケーションとしてだけでなく、携帯端末でも快適に動作するように設計されています。

この *API* では <http://maps.google.co.jp> サイトで使えるような地図を操作し、さまざまなサービスを介してコンテンツを地図に追加するための多数のユーティリティを提供しています。これを利用して、ウェブサイトにもパワフルな地図アプリケーションを作成できます。

JavaScript Maps API V3 は、誰でも自由にアクセスできるウェブサイトであれば、無料で利用できるサービスです。詳細については、利用規約をご覧ください。

¹<https://developers.google.com/maps/documentation/javascript/?hl=ja>

- この API を用いて地図を表示する例を与えている。
- API の仕様はデベロッパーガイドや API リファレンス（英語です）コードサンプルが参考になる。

Google Maps のオブジェクト

Google Maps で提供されるものの代表的なもの

- Map
地図を表示するためのオブジェクトである。
- Controls
地図をコントロールするオブジェクトである。ズームや、地図の移動を可能にするコントロールが含まれる。
- Overlays
地図上に表示されるオブジェクトである。折れ線、多角形、マーカーなどが含まれる。
- サービス
地図を利用する際に必要となるサービスである。緯度経度からその位置の情報（建物や住所）を得る Geocoder、ルート検索をする Direction サービスなどを含む。

GPS 機器で記録された移動記録を Google Maps 上に表示するサンプル

- GPS のログデータのリストを記述したファイルを読み込む
- その内容に基づいてルートを表示
- GPS のログデータのファイルの内容は各行に一つファイル名が書かれている。

20140412.gpx

20140922.gpx

20141128.gpx

20141128-plan.gpx

地図を表示するページのリスト (1)

URL で指定したオプションパラメーターの処理を行う関数を二つ定義

```
1  <?php
2  function setParam($name, $default, $min) {
3      $val = $default;
4      if(array_key_exists($name,$_GET)) {
5          $val = $_GET[$name];
6          if($val <$min) $val = $min;
7      }
8      return $val;
9  }
10 function setParamS($name, $default) {
11     $val =$default;
12     if(array_key_exists($name,$_GET)) {
13         $val = $_GET[$name];
14     }
15     return $val;
16 }
```

地図を表示するページのリスト(1)-解説

- 関数 `function setParam($name, $default, $min)` はパラメータが数値の場合を取り扱う。この関数は次の3つの引数をとる。
 - 一番目の引数 `$name` はパラメータの名称
 - 2番目の引数 `$default` はパラメータの値がなかった場合のデフォルト値
 - 3番目の引数 `$min` は設定するパラメータの最小値値をデフォルト値に設定したのち(3行目)、URLに指定した引数があるかチェックし(4行目)、あった場合にはその値が指定された最小値より大きいかわ調べ、小さい場合には指定された最小値に設定する(5行目から6行目)。
- 関数 `setParamS($name, $default)` は文字列のパラメータを設定するための関数である。この関数には設定される値のチェックはしない。

地図を表示するページのリスト (2)

```
17  $w = setParam("w",1000,200);  
18  $h = setParam("h",800,200);  
19  $hl =setParamS("hl","ja");
```

ここでは地図の横幅 (\$w)、高さ (\$h)、表示に使用する言語 (\$hl) などの値を設定している。

地図を表示するページのリスト (3)

```
20 $latMaxG=$latminG=35.486210;
21 $lonMaxG=$lonminG=139.341443;
22
23 $filenames = file("history.dat",FILE_IGNORE_NEW_LINES|FILE_SKIP_EMPTY_LINES)
24 Print <<<_EOL_
25 <!DOCTYPE html>
26 <html>
27 <head>
28 <title>My Trace</title>
29 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
30 <link rel="stylesheet" type="text/css" href="map.css"/>
31 <script src="http://maps.google.com/maps/api/js?sensor=false"
32       type="text/javascript" charset="UTF-8"></script>
33 <script src="jquery-1.11.2.min.js" type="text/javascript"
34       charset="UTF-8"></script>
35 <script src="map4.js" type="text/ecmascript"></script>
```

- 21 行目と 22 行目ではデフォルト地点の緯度 (\$latminG)、経度 (latminG) を定義 (この値は神奈川工科大学のもの)
- 24 行目では読み込む GPS ファイルのリストが入ったファイルを行単位で配列に格納
- 関数 `file()` の 2 番目の引数の定数は、行末の改行文字を取り除き (`FILE_IGNORE_NEW_LINES`)、かつ空行を読み飛ばす (`FILE_IGNORE_NEW_LINES`) を論理和 (`|`) でつないでいる。
- 25 行目から 46 行目では出力する HTML 文書の先頭部分をヒアドキュメントの形式で記述し、出力
- 32 行目から 33 行目では Google Maps のライブラリーを、34 行目から 35 行目では jQuery のライブラリーを読み込んでいる。
- 36 行目では 39 行目で呼び出される関数 `initialize()` などが定義されている JavaScript のファイルを読み込んでいる。

地図を表示するページのリスト (4)

```
36 <script type="text/ecmascript">
37 window.onload = function() {
38     initialize($latMaxG,$lonMaxG,$latminG,$lonminG);
39 }
40 </script>
41 </head>
42 <body>
43 <div id="map_canvas" style="width:{$w}px; height:{$h}px"><
44     <form id="filenames">
45 _EOL_;
```

- 38 行目から 40 行目ではこの HTML 文書での処理が終わった後の関数を定義。単純に initialize() を呼び出し。
- 44 行目では地図を表示するための<div>要素を定義している。この中では PHP の変数\w と\$h の値が展開されて埋め込まれる。

地図を表示するページのリスト (5)

```
46  for($i=0;$i<count($filenames); $i++) {  
47      print '<div class="fn"><div><input type="checkbox" id="file' . $i.  
48          '"/></div><div>' . $filenames[$i]. ' </div><div> </div></div>';  
49  }  
50  print '</form></div></body> </html>'  
51  ?>
```

47 行目から 50 行目では表示する GPX データを一覧できる、チェックボックス、ファイル名とこのパスの全体距離の 3 つが一つになった <div>要素を、ファイルの数だけ表示するようにしている。

地図を表示するページのリスト (6)—CSS ファイル

```
1 <style type="text/css">
2 #map_canvas {
3     position: relative;
4     float:left;
5 }
6 #filenames> div    {
7     float:left;
8     width:200px;
9 }
10 #filenames > div div:nth-child(1) {
11     width:20px;
12     float:left;
13 }
14 #filenames > div div:nth-child(2) {
15     width:80px;
16     float:left;
17 }
18
19 #filenames > div div:nth-child(3){
20     float:left;
21     width:100px;
22     text-align:right;
23 }
```

地図を表示するページのリスト (6)—CSS ファイル—解説

- 6 行目のセレクトは、id が filename である要素の直下の<div>要素を意味する。この中には GPX データの情報などを示す 3 つの<div>要素が含まれる。
- 10 行目のセレクトはチェックボックスを選択する。
- 14 行目のセレクトはファイル名を示す<div>要素を選択する。
- 19 行目のセレクトは GPX データの全体の距離を示す<div>要素を選択する。

地図を表示するページのリスト (7)

地図を表示するページで呼ばれる JavaScript ファイル
Ajax を用いてルートデータをサーバーに要求し、順にルートを表示するようになっている。

地図を表示するページのリスト (8)

地図を表示するための準備の部分

```
1 function initialize(latMax,lonMax,latmin,lonmin) {
2     var Clrs = [[255,0,0],[63,63,63],
3                 [0,0,255],[127,127,0],[127,0,127],[0,127,127]];
4     var GMap = google.maps;
5     var Routes =[];
6     var map = new GMap.Map($("#map_canvas")[0],
7         {center: new GMap.LatLng((latMax+latmin)/2,(lonMax+lonmin)/2),
8         mapTypeId: google.maps.MapTypeId.ROADMAP,
9         scaleControl:true,
10        scaleControlOptions:google.maps.ControlPosition.BOTTOM_LEFT
11    });
12    var latLngBounds = new GMap.LatLngBounds(
13        new GMap.LatLng(latmin,lonmin),
14        new GMap.LatLng(latMax,lonMax));
15    map.fitBounds(latLngBounds);
16 }
```


地図を表示するページのリスト (8)—解説

- 関数 `initialize()` の引数は表示すべき地図の範囲の南西部の緯度と経度、北東部の緯度と経度
- 2 行目から 3 行目では、GPS ログを示すルートの色の種類を与えている。数が増えたときは、この要素の数で割った余りの色となる。
- 6 行目から 11 行目で地図を表示している。なお、ここでは jQuery 関数で HTML 文書内の要素を得ている。jQuery オブジェクトに対して Google Maps の地図を割り当てる要素として使えないので引数を与えて、HTML 要素を得ている。
- 12 行目から 15 行目で 13 行目と 14 行目で与えられた緯度経度の地点を含む範囲 (`LatLngBounds`) を構成
- 15 行目でこの範囲を含むような最大のズームレベルにするメソッド `fitBounds()` を呼び出している。

地図を表示するページのリスト (9)

```
17 var InfoWindow = InfoWindow = new google.maps.InfoWindow({});
18   var Colors = [], C2 = [], i, j;
19   for(i=0;i<Clrs.length;i++) {
20     Colors[i] = "rgb("+Clrs[i].join(",")+")";
21     C2[i] = "rgb("+Math.floor((Clrs[i][0]+255*2)/3,1)+"",
22             Math.floor((Clrs[i][1]+255*2)/3,1)+"",
23             Math.floor((Clrs[i][2]+255*2)/3,1)+"")";
```

- 16 行目では地図上に情報を表示するための InfoWindow() を作成
- 18 行目から 23 行目では GPS ログのルートの一覧部分の背景色の設定をしている。パスの表示は重なった場合に備えて不透明度を 0.5 にしている (52 行目) ので、定義の色より明るくしている。

地図を表示するページのリスト (9)–GPS ログを地図上に表示する

ページがすぐに開かないことを避けるために Ajax を用いてデータを得ている。

```
24     var Files = $(".fn");
25     for(j = 0;j<Files.length;j++){
26         (function(j) {
27             jQuery.ajax({
28                 type:"GET",
29                 url: "./getData.php",
30                 data:"file="+$($("div",Files[j]))[1]).text(),
31                 dataType:"json",
32                 error: function(){alert("error");},
```

地図を表示するページのリスト (9)-解説

- 24 行目で HTML 内に表示されているルートを表示している要素を得る。
- 25 行目からそのファイル名をサーバーに渡して、GPS ログのデータを得て、地図上に表示する。
- 26 行目では関数を定義してその場で実行することで、Ajax の処理を行っている。これは、ループの制御変数 j (30 行目) が Ajax の通信が終了した段階 (47 行目など) で変化しているかもしれないことに対処するためである。
- 27 行目から 72 行目で jQuery による Ajax 関数の呼び出しを行っている。

地図を表示するページのリスト (10)

通信終了後に呼び出される関数

```
33      success: function(Data){
34          var ps, length, PP, P, LL, point, i,k;
35          ps = [];
36          length =0;
37          for(k=0;k<Data.route.length;k++){
38              P = Data.route[k];
39              LL = P.length;
40              ps.push(new GMap.LatLng(P[0][0], P[0][1]));
41              for(i=1;i<LL;i+=1) {
42                  point = new GMap.LatLng(P[i][0], P[i][1]);
43                  length += getDistance(P[i-1],P[i]);
44                  ps.push(point);
45              }
46          }
47      }
```

地図を表示するページのリスト (9)-解説

- サーバーから得たデータはすでに JavaScript のオブジェクトとなっているので、その `position` プロパティに緯度と経度の配列を要素とする配列が設定されている。
- 37 行目から 46 行目でルートを示す `PolyLine` オブジェクトに渡す配列に変換している。さらに、ルートの長さも計算している。

地図を表示するページのリスト (10)

表示すべき PolyLine を配列に格納

```
48     Routes[j] = [  
49         new GMap.Polyline({  
50             path:ps,  
51             strokeColor:Colors[j % Colors.length],  
52             strokeWeight:4,  
53             strokeOpacity:0.5,  
54             map:map  
55         }),true];
```

地図を表示するページのリスト (11)

```
56  $($("div",Files[j])[2]).text((length/1000).toFixed(2)+
57  $(Files[j]).css("backgroundColor",C2[j % Colors.length]
58  $("input",Files[j]).attr("checked",true);
59  var showId = (function(No){return function(E){
60      InfoWindow.setOptions({
61          content: "<div>"+ No+"</div>",
62          position: E.latLng});
63      InfoWindow.open(map);
64      };
65  })($($("div",Files[j])[1]).text());
66  latLngBounds.extend(
67      new GMap.LatLng(Data.position[0],Data.position[2]))
68  latLngBounds.extend(
69      new GMap.LatLng(Data.position[1], Data.position[3]
70  map.fitBounds(latLngBounds);
71  GMap.event.addListener(Routes[j][0],"click", showId)
72  }
73  });
```


地図を表示するページのリスト (10)-解説

- 55 行目では計算された距離をリスト内に表示し、背景色を定義
- 57 行目では対応するチェックボックスをにチェックを入れる。
- 58 行目から 64 行目では、ルート上をクリックしたときにそのファイル名を InfoWindow に表示する関数を定義。70 行目でイベントの登録を `addListener()` メソッドを用いてしている。
- サーバーからはルートが含まれる範囲の緯度経度が渡されているのでそれをもとに今までの範囲を `extend()` メソッドで拡大

地図を表示するページのリスト (11)

ファイルのリストのチェックボックスの値が変化したときに、ルートを表示のオンオフをする処理

```
76 document.getElementById("filenames").addEventListener("ch
77     function (E) {
78         var i = $(E.target).attr("id").substring(4)-0;
79         if(E.target.checked && !Routes[i][1]) {
80             Routes[i][0].setMap(map);
81             Routes[i][1] = true;
82         }
83         if(!E.target.checked && Routes[i][1]) {
84             Routes[i][0].setMap(null);
85             Routes[i][1] = false;
86         }
87     },false);
88 }
89
```

地図を表示するページのリスト (12)

緯度と経度で与えられた 2 点間の距離を求める関数

```
90 function getDistance(P1, P2) {  
91     var R = 6378137;  
92     var P1LatRad = P1[0]*Math.PI/180;  
93     var P1LngRad = P1[1]*Math.PI/180;  
94     var P2LatRad = P2[0]*Math.PI/180;  
95     var P2LngRad = P2[1]*Math.PI/180;  
96     var P1LatCos = Math.cos(P1LatRad);  
97     var P2LatCos = Math.cos(P2LatRad);  
98     var Xdiff = R*(P1LatCos*Math.cos(P1LngRad) - P2LatCos*Math.cos(P2LngRad));  
99     var Ydiff = R*(P1LatCos*Math.sin(P1LngRad) - P2LatCos*Math.sin(P2LngRad));  
00     var Zdiff = R*(Math.sin(P1LatRad) - Math.sin(P2LatRad));  
01     return Math.sqrt(Xdiff*Xdiff+Ydiff*Ydiff+Zdiff*Zdiff);  
02 }  
03
```

地図を表示するページのリスト (13)

GPS データの各地点の緯度と経度のリストを JSON 形式で返す

```
1 <?php
2 function SetDatafromFile($fin,$folder) {
3     $fn = "$folder$fin";
4     if(!file_exists($fn)) return;
5
6     $data = new DOMDocument();
7     $data->load($fn);
8     $trks = $data->getElementsByTagName("trk");
```

地図を表示するページのリスト (13)-解説

- GPX ファイルは XML 形式のファイルなので、構造を保持したままデータを扱うためには PHP の DOMDocument オブジェクトに読み込むのが簡単である。
- 6 行目で DOMDocument オブジェクトを作成し、7 行目でファイルを読み込んでいる。
- 残りの捜査は DOM の扱いと同様に `getElementsByTagName()` などを用いてデータを処理すればよい。
- GPX には複数の `<trk>` 要素があり、その中に位置情報を示す `trkpt` がある。

地図を表示するページのリスト (14)

```
9      $len = $trks->length;  
10     $trackdata = array();  
11     $lonMax = -181;  
12     $lonmin = 181;  
13     $latMax = -91;  
14     $latmin = 91;
```

地図を表示するページのリスト (15)

```
15 $cnt=0;
16 for($i=0;$i<$len;$i++){
17     $trk = $trks->item($i);
18     $trksegs = $trk->getElementsByTagName("trkpt");
19     $len2 = $trksegs->length;
20     if($len2 <10) continue;
21     $newtrk = array();
22     for($j = 0;$j<$len2;$j++) {
23         $trkseg = $trksegs->item($j);
24         $lat = $trkseg->getAttribute("lat");
25         $lon = $trkseg->getAttribute("lon");
26         if($lat > $latMax) $latMax = $lat;
27         if($lat < $latmin) $latmin = $lat;
28         if($lon > $lonMax) $lonMax = $lon;
29         if($lon < $lonmin) $lonmin = $lon;
30         array_push($newtrk,"[$lat,$lon]");
31     }
32     if($lonMax - $lonmin >0.001 ||$latMax - $latmin > 0.001) {
33         $cnt++;
34         array_push($trackdata,"[.implode(",",$newtrk)."]");
35     }
36 }
37 print "{\position\":[$latMax,$latmin,$lonMax,$lonmin],\route\":[.
38     implode(",",$trackdata).]}";
39 }
40 SetDatafromFile($_GET["file"],"");
41 }>
```

地図を表示するページのリスト (14)–解説

- trkpt 要素の属性に緯度 (lat) と経度 (lng) があるので、それを組み合わせた JSON 形式の配列 (の文字列) を配列に追加する (30 行目)。
- 途中でルート全体を含む緯度経度の範囲を求めている (26 行目から 29 行目)。
- これらのデータをまとめて出力する (37 目から 38 行目)

試験の内容

- 授業で行った簡単なプログラムや CSS の設定に関するもの
- プログラムを組む問題はなし
- いくつかの項目について論述する。
 - JavaScript の言語の特性に関する考察
 - jQuery のライブラリーを使うメリットとデメリット
 - Google Maps のライブラリの設計思想に関する意見
 - そのほか、この授業を通じて理解したことについての論述 (各自が設定してよい)