

外部から図形情報を読み込む

2-1-2-2 では円の位置を JavaScript に直接書き込んだが、その方法では円を描く位置が多くなった場合、JavaScript が大変長くなってしまふ。そこで、位置情報を記述したファイルを別に用意して読み込ませるという方法をとる。今回は、「data.json」というファイルに円の中心の緯度経度を記述している(JSON は Web アプリケーションでのデータのやり取りによく用いられ、人間も機械も読み取りやすい形式である)。おおよそ 2-2 と同じような配列になっているので、一応中身を確認してみよう。それでは、下のソースコードに従ってフォルダ「3-1」直下の「main.js」を書き換えてみよう。

main.js

```
'strict'

var map = null;

function initMap() {
  map = new google.maps.Map($('#map').get(0), {
    center: {lat: 43.068543, lng: 141.351128},
    zoom: 15
  });

  - // circle
  - var circle_coordinates = [
  -   {lat: 43.076490, lng: 141.333961},
  -   {lat: 43.072195, lng: 141.368723},
  -   {lat: 43.058400, lng: 141.337008},
  - ];
  + $.ajax({
  +   url: 'data.json',
  +   type: 'GET',
  +   dataType: 'json'
  + })
  + .done(function(data){
  +   console.log(data);
  +   render(data);
  + })
  + .fail(function(){
  +   console.log('error');
  + });
  +}

  - circle_coordinates.forEach(function(coordinate){
  +function render(data){
  + data.circle_coordinates.forEach(function(coordinate){
    var circle = new google.maps.Circle({
      map: map,
      center: coordinate,
      radius: 100
    });
  });
  });
}
```

ここでは、「Ajax」という処理方法を用いて、ページそのものを更新することなく、緯度経度のデータを別ファイルから取得している(プログラム 11~23 行目)。「url」にファイル名、「datatype」にファイル形式(この場合は JSON)を記述している。

上手くいくと、2-2 と同様、3 か所に円が描かれているはずである。時間があれば、「data.json」の緯度経度の値を変えて同じ場所にアップロードし、円の位置が変わるかどうか見てみよう。