地図を表示する

ブラウザに GoogleMap を表示するには、HTML、CSS、そして JavaScript が記述されている必要がある。HTML は Web ページの枠組みを作るもの、CSS は Web ページの装飾をするもの、そして JavaScript は Web ページに動きを加えたり複雑な処理を行えるようにしたりするものである。 今回の演習ではこれら 3 つがそれぞれ別のファイルに記述されているが、その内 HTML と CSS については時間の都合上説明を省き、JavaScript(拡張子が js のファイル)を中心として簡単に説明しながら書き換え、地図を表示させることを目指す。

本章では、指定した緯度経度を中心とする地図を表示させることが目標である。フォルダ「1-1」直下にある「main.js」というファイルには現状何も書かれていないはずであるので、まずはそのファイルに以下のように記述し、アップロードして地図を確認してみよう。

main.js

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

上手くいけば、次ページのように JR 札幌駅を中心とした Google マップが表示されるはずである。 このように、Google Maps API の google.maps.Map クラスを使うことで地図を表示させているが、 特にここでは緯度経度とズームレベルの指定の仕方を見ていく。これらの指定は、オプションで行っている。

地図の中心の緯度経度は、プログラム 5 行目の「center」というオプションで設定しており、「lat」は 緯度、「lng」は経度を意味する。単位は°(度)で、小数を使った表記(DEG 表記)を用いる。また、緯 度が正の数の時は北緯、経度が正の数の時は東経を表す。すなわち、ここでは北緯 43.068543°、 東経 141.351128°を中心にして Google マップを表示させている。

ズームレベルは、6 行目の「zoom」というオプションで設定しており、ここではズームレベル 15 を指 定している。

以上のことを確認するには、緯度経度やズームレベル(0~21)を適当にいじってみるとよいだろう。

2. playmyldig/gent140/essory X +	- 0 ×
← → C ① 保護れていない遺信 play.mylab.jp/guest145/resson/googlemapsap/sample-1-1/index.html	x 🗹 🍮 i
地図 加空写真 た20年点 北道会学大型建築の (地区の20年) (1975年) (197576)	J−E−♥ ^{ncerna} creeze di ♥
● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	法国公園 社会医療 三草合 クラ・
	面積グリーン 公開 ふしこ幼稚
● ヤマク目後 マクランド 211日日 211日 21日日	札幌苗穂郵使局 ♥
	「保育園 ♥ 香田メグミルク ♥ 路農と乳の歴史館
シロ注意 生のたまゆら 保護 ・ ・ ・ ・ ・ ・ ・ ・ ・	スタジオカティス 🛛 🥍
	P∓1818
■ 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	温泉 蔵ノ県 上1矢東
● 1200-12時時時、1201123-2010日 ● 12月2日 23月日 23月2日 23月日 23月2日 23月日 23月2日 23月日 23月2日 23月1日 23月2日 23月1日 23月3日 23月11日 23月11日	577 E
バラースリーンイヤダ 20% エスター	小◆ 札幌市立上白石小 ◆
	1 Q 第次9条 北海道礼院東嘉 Q
	5×7%
	♀ 札戦市立戦東中 □×9取員 +
	● 利用振行 市局の計りを報告する

本章の課題が上手くいけば、上のような地図が表示されるはずである。もし上手くいかない場合は、 Google Chrome のウィンドウ右上にある、縦に3つ並んだ点の部分(下図赤丸)をクリックし、「その 他のツール」>「デベロッパーツール」を開いて「Console」のタブをクリックすると、どの部分がエラ ーを起こしているか確認できる(大抵は単純な打ち間違いなので、よく確認してみよう)。

					_	o ×
					☆ 🗸	
北18荼通 ◎	学校法人札 学園 札幌:	幌大谷 大谷大 環	新しいタブ 新しいウィ シークレット	『(T) ンドウ(N) ト ウィンドウを開く(I	I) (Ctrl+T Ctrl+N Ctrl+Shift+N
北斗高 ビジネス専門 北海道スポーツ専門 学校2号館 吉田学師 ^編	11枚 学校 環状通 問 図 ・ 北15条東 89 現状通 札幌光星高 マ 北13条東 北13条東	主要醫練	履歴(H) ダウンロー ブックマー?	ド(D) 7(B)		► Ctrl+J
マ 画 副 13条東 の で 社会医療 母恋 天使 マ マ マ マ マ マ マ マ マ マ マ マ マ	東区役所前(幻→地・* 法人 病院 ○ 北10条束	幼保連携型認定 園せいめい 北11条東 札幌市立苗穂小	ズーム 印刷(P) キャスト(C 検索(F))	- 100%	+ E C Ctrl+P Ctrl+F
₹ (X 49) TEE [20]	名前を付けてページを保存(A)	Ctrl+S	その他のツ	ノール(L)		×
 スポーツデボ光 スポーツデボ光 スポーツデボ光 	ショートカットを作成 閲覧履歴を消去(C) 拡張機能(E) タスク マネージャ(T)	Ctrl+Shift+Delete Shift+Esc	編集 設定(S) ヘルプ(H)	切り取り(T)	⊐ピ–(C)	貼り付け(P) ト
● 273 第 第 中央郵便局	デベロッパー ツール(D) R庭教師の コーソー 札幌支店	Ctrl+Shift+l	終了(X) E 北4:	条東 泉 函館本線		千歲線
北5条束	JA北海道開 札幌厚4	字生理"♀──धाध्यध्य∞稼苗穂に	10001	1 古蓮眼前	进 /温	

地図に図形を表示する

続いては、Google マップ上に図形を表示させる。そのためには、1-1 で学んだ Google マップを表示させるクラスに加えて、図形を表示させるクラスを用いる。今回は、Google マップ上に円を表示させるクラスを使って、指定した緯度経度に円を表示させてみる。フォルダ「1-2」直下の「main.js」に は、以下のソースコードのうち黒字の部分(1-1 の「main.js」と同じ文)が記述されているので、その下の緑字の部分をその「main.js」に書き加えてアップロードし、地図の表示を確かめよう(今後、緑字の部分については「書き加え」を意味し、「+」より後の部分を書く。なお、「+」のみの行は空白行を加える)。

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_coordinate = {lat: 43.076490, lng: 141.333961};
+
  var circle = new google.maps.Circle({
    map: map,
    center: circle_coordinate,
    radius: 100
  });
```

上手くいけば、地図上の北大敷地内に黒い円が描かれているはずである(次ページ図)。

地図上に円を描くには、google.maps.Circleクラスを用いる(プログラム13行目)。14行目の「map」 オプションで円を表示させる Google マップを指定し、15 行目の「center」オプションで円の中心を 指定している。円の中心の緯度経度は、その前の 12 行目で「circle coordinate」という変数に格 納しているので、15 行目では変数名「circle coordinate」で指定する。16 行目の「radius」オプシ ョンは円の半径を実際の距離で指定するもので、m 単位で指定できる。ここでは実際の大きさで半 径 100mの円を描いていることになる。

時間があれば、円の場所や半径を変えてみよう。

なお、11 行目は円を描くことを明示しているコメントであり、プログラムの動作には関係ない (JavaScript では、1行のコメントには先頭にスラッシュを2つつける)。

2 plymylabjplyurit40/essn/; x +	-	Ø	×
そ ② ② 保護されていない通信 play.mylab.jp/guest145/lesson/googlemapsapi/sample-2-1/index.html	☆ 🔽	10	:
地図 加空写真 たのは、生品基大学大学医惑、 ・単語之 、ない、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、	Iーヒー 🖓 小 🤤 法国公園 👰	「「「」」	に 100000 二 二 二 二 二 二 二 二 二 二 二 二 二
	苗穂クリー 公開 札幌苗穂彩信 住保育園 ♥	-シ ふし E間 🗣) C 50 M
	雪田メグ 筋震と乳の スタジオカティ	ミ 史 招 マ 千歳線	101
	昌泉 蔵ノ湯 北1条束 10 泉水	10.7711 2.07	R Shite
したりシーシードには6 ・ 北東市会子単語 6 ・ エスエポス活動 1 北京北ポス活動 1 北京市会子 1 北京市会子 1 北京市会子 1 北京市会子 1 日本市会子 1 日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日	P P HLWEI	而立上白: 一類水	石小 C
	北海道礼祭 水7条	東高・	
		レダ油田 札幌5点	+

地図に複数の図形を表示する

2-1 では 1 か所に円を描いたが、今度は複数個所に円を描いていく。フォルダ「2-2」直下の 「main.js」について、以下の緑字の部分は 2-1 同様書き加え、赤字の部分は削除し(今後も同様、 当然行頭の「-」は記述しない)、JavaScript を完成させる。

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_coordinate = {lat: 43.076490, lng: 141.333961};
  var circle = new google.maps.Circle({
     map: map,
     center: circle_coordinate,
     radius: 100
  var circle_coordinates = [
+
    {lat: 43.076490, lng: 141.333961},
{lat: 43.072195, lng: 141.368723},
{lat: 43.058400, lng: 141.337008},
+
+
+
  ];
+
+
   circle coordinates.forEach(function(coordinate){
+
+
     var circle = new google.maps.Circle({
       map: map,
+
       center: coordinate,
+
       radius: 100
+
    });
+
  });
```

成功すると、3 か所に円が描かれた Google マップが表示されるはずである(次ページ図)。今回は 指定する円の中心を 3 か所に増やし、それらの緯度経度を「circle coordinates」という配列に格 納している(プログラム 12~16 行目)。

18 行目では、Google マップ上に円を描く動作を、配列「circle coordinates」の各緯度経度に対し て行うように指示している。円を描く動作そのものは、2-1 と同様である。余裕があれば、円を描く緯 度経度を変えたり円を増やしたりしてみてもよいだろう。

2. physynthilip/specific/filesovie X +	- a ×
← → C ① 保護されていない通信 play.mg/ab.jp/guest145/fesson/googlemapsapi/sample-2-2/index.html	☆ 🗹 🏐 :
	1-ヒー 〇 「Centrategeory 小 〇 二 二 法国公園 〇 二 単会のう 三 単会のう
日1点目 日1 日1点目 日1 日1 日1 日1 日1 日1 日1 日1 日1 日1	「商業クリーン」
	ふしこ幼稚 れ城苗穂彰使局 🗣
日本語を見た場合 日本語を見た場合 日本語を見た場合 日本語を見ていた時に 日本語を見たり 日本語を見たい 日本語を見た 日本語 日本語を見た 日本語を見た 日本語を見た 日本語を見た 日本語を見	1日日日 Q (11) (11) (11) (11) (11) (11) (11) (1
	略農と乳の歴史館
マルニー上日料製作用 ・ ・ ・ ・	干機線
	北日安東 日本川 日本 南水上町
	札幌翰水上
	ト♥ 札幌市立上白石小 ♥ Ⅰ♥ ♥ 5×98
	北海道扎慶東高 <table-cell></table-cell>
	● 礼間市立領東中
	コメダ酸剤 取り(成5)第 の一 の一 の一 の一 の一 の一 の一 の の の の の の の の の

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う。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。

```
'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。

MySQL から取得した緯度経度に円を描く

ここからは、MySQL を用いてデータベースにある位置情報を地図上にプロットしていく。ここでも 3-1 でやったように、Ajax を用いて外部からデータを取得することになるが、その際に同じフォルダ の JSON ファイルではなく、PHP を通じてデータベースから SQL で取得したデータを用いる。 今回はファストフードチェーン・マクドナルドの店舗の位置情報を取得し、プロットしていく。ここでは これまで同様に 1 つ目のソースコードに従って「main.js」を書き換えると共に、「api.php」というま っさらなファイルがあるので、2 つ目のソースコードを「api.php」に書き込んでいこう。

```
'strict'
var map = null;
function initMap() {
 map = new google.maps.Map($('#map').get(0), {
   center: {lat: 43.068543, lng: 141.351128},
   zoom: 15
   });
 url: 'data.json
url: 'api.php',
+
   type: 'GET',
dataType: 'json'
 })
  .done(function(data){
   console.log(data);
   render(data);
 })
  .fail(function(){
   console.log('error');
 });
}
function render(data){
  data.circle coordinates.forEach(function(coordinate){
+ data.forEach(function(geom){
   var circle = new google.maps.Circle({
     map: map,
_
      center: coordinate,
      center: {lat: geom.coordinates[1], lng: geom.coordinates[0]},
     radius: 100
   });
 });
```

api.php

```
<?php
$db = mysqli_connect('localhost', 'guest', 'Test@1234', 'opendata');
if(!$db){
       return;
}
sql = <<<EOS
SELECT ST_AsGeoJson((geom)) AS geojson
FROM mcdnald
EOS;
$stmt = $db->prepare($sql);
$stmt->execute();
$stmt->bind_result($geojson);
$json = [];
while(($row = $stmt->fetch()) != NULL){
       $json[] = json_decode($geojson);
}
echo json encode($json);
exit;
```

上手くいけば、図1のように札幌市中心部のマクドナルドの店舗位置がプロットされるはずである。 試しに地図を動かせば他の場所にあるマクドナルドもプロットされているし、ズームを引くと日本全 国のマクドナルドの店舗位置がプロットされているのが分かるだろう。デベロッパーツールの 「Network」タブを開きながら Google マップを開き、「api.php」をクリックすると、全国約3000店舗 のマクドナルドの位置情報を読み込んでいることが分かる(図2)。

今回用いた PHP は、JavaScript とは異なり、サーバーサイドで実行される言語である。PHP に ついては本筋から外れるので深くは触れないが、9~12 行目で MySQL を用い、データベースの 「mcdnald」というテーブルからマクドナルドの位置情報を GeoJSON という形式で取得している (GeoJSON は、JSON を基にして地理情報を扱えるようにした形式である)。これらの作業をサー バー側で行うことにより、クライアント側ではあたかも PHP がマクドナルドの位置情報を記述した GeoJSON ファイルであるかのように見える。一方、JavaScript 側では PHP で取得したマクドナ ルド店舗の位置情報をプロットする作業を行っている。先ほどの Ajax では「url」が「data.json」で あったが、今度は「api.php」になっている(プログラム 12 行目)。

なお、GeoJSON には「経度、緯度」の順番(x,y の順番)で格納されているので、円の中心を 「center」オプションで指定する際は「緯度は配列の2要素目、経度は配列の1要素目」というよう に指定しなければならないが、JavaScript では配列の要素番号が[0]から始まるため、プログラム 29行目のように記述する必要があるという点に注意が必要である。

ただし、今回のソースコードでは先ほど述べたように全国約 3000 店舗分のマクドナルドの店舗情

報を読み込んでいるので、データ量も大きくなり、表示にも時間がかかってしまう。次の 2-1 では、 その問題を解決する。



図 1. 札幌市中心部のマクドナルドの店舗位置をプロットした図。

		– 0 ×
		☆ 🔽 🍮 🗄
	Elements Console	Sources Network Performance Memory Application » 01
人札幌大谷 「 」 礼幌大谷大 「 」	● ◎ ■ ▼ ♀ View:	I Group by frame Preserve log Disable cache Offline No throttling
	Filter	Hide data URI's AU XHR IS CSS Imo Media Font Doc WS Manifest Other
	500 ms	1000 ms 1500 ms 2000 ms 2500 ms 3000 ms
(車) (9)		
也下鉄東豐線		
幼保連携型語	Name	× Headers Preview Response Cookies Timing
園せいる	index.html	<pre>▼[{type: "Point", coordinates: [141.292480489, 43.0772826618]},]</pre>
北11条束	main.css	▶ [0 99] ▶ [100 100]
し、札幌市立苗和	jquery.min.js	▶ [200 299]
北9条東	main.js	▶ [300 399]
ク光星店	js?key=AlzaSyDzkWLmgsO4pc	▶ [400 499] ▶ [500 599]
1/8冬車	defObject.json	▶ [600 699]
100 ***	common.js	▶ [700 799]
,8条通	util.js	▶ [800 899] ▶ [000 900]
	map.js	▶ [1000 1099]
16冬東	api.php	▶ [1100 1199]
	openhand_8_8.cur	▶ [1200 1299]
	onion.js	▶ [1200 1299] ▶ [1400 1499]
	ViewportInfoService.GetViewp	▶ [1500 1599]
日道厚生連 💡 出館本線	vt?pb=!1m5!1m4!1i15!2i2925(▶ [1600 1699]
苗穂!	vt?pb=!1m5!1m4!1i15!2i29249	▶ [1700 1799] ▶ [1800 1809]
北3条東	vt?pb=!1m5!1m4!1i15!2i29249	▶ [1900 1999]
北3条通	vt?pb=!1m5!1m4!1i15!2i2925(▶ [2000 2099]
	vt?pb=!1m5!1m4!1i15!2i29251	▶ [2100 2199]
北2条週	vt?pb=!1m5!1m4!1i15!2i29251	▶ [2200 2299] ▶ [2300 2399]
副志子国 文	vt?pb=!1m5!1m4!1i15!2i29251	▶ [2400 2499]
	vt?pb=!1m5!1m4!1i15!2i29250	▶ [2500 2599]
大通吏	vt?pb=!1m5!1m4!1i15!2i29249	▶ [2600 2699] ▶ [2700 2799]
1. The second second	vt?pb=!1m5!1m4!1i15!2i29248	▼ [2800 2887]
大迎	vt?pb=!1m5!1m4!1i15!2i2924{	▶ 2800: {type: "Point", coordinates: [131.565260338, 33.2123677235]}
札幌市役	vt?pb=!1m5!1m4!1i15!2i2924{	2801: {type: "Point", coordinates: [131.783968778, 33.1148919505]} > 2802. {type: "Deint", coordinates: [131.656736887, 33.1566706148]}
未来局子育て	vt?pb=!1m5!1m4!1i15!2i2924{	<pre>> 2803: {type: "Point", coordinates: [131.356720087, 33.1506790148]} > 2803: {type: "Point", coordinates: [131.341921393, 33.5268761357]}</pre>
公益社団法人北:	vt?pb=!1m5!1m4!1i15!2i29249	<pre>> 2804: {type: "Point", coordinates: [131.6580253, 33.195172319]}</pre>
勤労者医療協	vt?pb=!1m5!1m4!1i15!2i2925(▶ 2805: {type: "Point", coordinates: [131.614414182, 33.2460290198]}
	vt?pb=!1m5!1m4!1i15!2i29251	2806: {type: "Point", coordinates: [131.422/40/47, 31.896//9463/]} > 2807: {type: "Point", coordinates: [131.426278556, 31.926678964]}
	vt?pb=!1m4!1m3!1i15!2i29248	<pre>▶ 2808: {type: "Point", coordinates: [131.399681875, 31.9213040684]}</pre>
院開祖 志村鉄 翁居住の地跡	geometry.js	▶ 2809: {type: "Point", coordinates: [131.652784113, 32.4253892505]}
	poly.js	<pre>> 2810: {type: "Point", coordinates: [131.068503642, 31.7342277429]} > 2811: {type: "Point", coordinates: [131.454562003_31_9224380228]}</pre>
	transparent.png	<pre>> 2812: {type: "Point", coordinates: [131.379635041, 31.604391569]}</pre>
社会福祉法	controls.js	▶ 2813: {type: "Point", coordinates: [131.0794696, 31.7349070349]}
会 菊水す 呆	AuthonticationSonuice Authont	<pre>> 2814: {type: "Point", coordinates: [130.995221083, 31.9824544406]} > 2814: {type: "Point", coordinates: [331.454653644, 31.9824544406]}</pre>

図 2. 「api.php」の中身。全国 2887 店舗分の位置情報が読み込まれている。

表示範囲の経緯度のみを MySQL から取得

1-1 ではデータベースにあるマクドナルドの位置情報をプロットしたが、その際に取得したのは全国 約 3000 店舗分のマクドナルドの位置情報である。したがってデータサイズが大きくなり、その分表 示にも時間がかかってしまう。そこで、地図の表示範囲にある店舗の位置情報のみを取得してデー タサイズを抑え、地図の表示時間を短縮する。

```
'strict'
var map = null;
function initMap() {
 map = new google.maps.Map($('#map').get(0), {
    center: {lat: 43.068543, lng: 141.351128},
    zoom: 15
    });
  google.maps.event.addListener(map, 'idle', idleMaps);
+
+}
+function idleMaps()
+{
+
  var zoom = map.getZoom();
+
 var bounds = map.getBounds();
  var ne = bounds.getNorthEast();
  var sw = bounds.getSouthWest();
+
  var params = 'zoom=' + zoom + '&n=' + ne.lat() + '&s=' + sw.lat() +
'&e=' + ne.lng() + '&w=' + sw.lng();
+
 $.ajax({
    url: 'api.php',
    url: 'api.php?' + params,
    type: 'GET',
+
   dataType: 'json'
  })
  .done(function(data){
    console.log(data);
    render(data);
  })
  .fail(function(){
    console.log('error');
 });
}
function render(data){
 data.forEach(function(geom){
   var circle = new google.maps.Circle({
     map: map,
      center: {lat: geom.coordinates[1], lng: geom.coordinates[0]},
      radius: 100
    });
```

}); }

api.php

```
<?php
$db = mysqli_connect('localhost', 'guest', 'Test@1234', 'opendata');
if(!$db){
         return;
}
+$north = $_GET['n'];
+$west = $_GET['w'];
+$east = $_GET['e'];
+$south = $_GET['s'];
+$zoom = $_GET['zoom'];
+
+$geom = 'POLYGON((';
- $west . ' ' . $north .
+$geom .= $west . ' ' . $north .
+$geom .= $west . ' ' . $south .
                                             ;
                                         · · ·
+$geom .= $east . ' ' . $south .
+$geom .= $east . ' ' . $north .
                                             ;
                                          נ' נ'
+$geom .= $west . ' ' . $north;
+$geom .= '))';
$sql = <<<EOS
SELECT ST_AsGeoJson((geom)) AS geojson
FROM mcdnald
+WHERE ST_Intersects(geom, ST_GeomFromText(?, 4326, 'axis-order=long-
lat')) = True
EOS;
$stmt = $db->prepare($sql);
+$stmt->bind_param('s', $geom);
$stmt->execute();
$stmt->bind result($geojson);
$json = [];
while(($row = $stmt->fetch()) != NULL){
         $json[] = json_decode($geojson);
}
echo json_encode($json);
```

地図の表示範囲にある店舗の位置情報のみを取得するには、おおよそ次のような流れで行う。

① JavaScript で表示領域の東西南北の緯度経度を取得し、パラメータとして PHP に渡す

② PHP で、渡された東西南北の緯度経度を用いて四角形のポリゴン(面)データを作成する

③ 生成された四角形に含まれる位置情報データのみを取得するよう SQL に条件追加(PHP)

まず①について、表示領域の東西南北の緯度経度を取得するには表示領域の四隅の座標が分 かればよい。さらに言うと、対角に位置する2点の座標さえ分かれば四隅の座標も分かるので、東 西南北の座標を取得できる(図 1)。ここでは地図の表示領域を getBounds()で取得した後、北東 (右上)の座標を getNorthEast()で、南西(左下)の座標を getSouthWest()で取得し、そこから北 西(左上)と南東(右下)の座標を算出する、という処理を「main.js」で行っている(プログラム 17~19 行目)。こうして得られた表示領域の座標から東西南北の緯度経度をパラメータとして構築し、PHP に渡す。



図 1. 地図の表示領域における東西南北の緯度経度を取得する方法のイメージ。

東西南北の緯度経度を渡された「api.php」では、②と③の処理を行うことになる。②については、「api.php」の 9~21 行目において、渡された東西南北の緯度経度を用いて 4 つの点を作り、四角形のポリゴンデータを生成する。ここでは、北西(左上)の点から反時計回りに 4 点をつなぎ、最後に北西の点に戻るという形で、四角形のポリゴンを作っている(図 2)。



図 2. 東西南北の座標から四角形のポリゴンを作成するイメージ。

そして③では、生成した四角形に含まれるような店舗の位置情報を取得するよう SQL に条件を追加する。店舗が四角形の中に含まれるかを判断するためには、26 行目のように ST_Intersects を 用いて記述する(図 3)。ST_Intersects では 2 つの図形に共通の部分があれば「True」を返すの で、「api.php」26 行目のように「『ST_Intersects = True』となるマクドナルド店舗(を表す円の中 心)の位置情報を取得する」という条件を与えれば、地図の表示範囲内のマクドナルド店舗の位置 情報のみを取得することができるのである。



boolean ST_Intersects(geometry geomA, geometry geomB); 赤い四角形と交わる(True)のは太い円。

図 3. 店舗が生成した四角形の中に含まれるかを判断するイメージ。

以上のようにして、Google マップの表示領域内にあるマクドナルド店舗の位置情報のみを取得す ることができたので、後は再び JavaScript で 1-1 と同様に Ajax を使い、取得した位置情報をプ ロットする。

成功すれば、最初は 1-1 と同じ場所にマクドナルドの店舗位置がプロットされるだろう。しかし、デ ベロッパーツールの Network から api.php をみると、含まれているデータの数は数個しかないは ずである(図 4)。地図を動かしたりズームレベルを変えたりすれば、その度に上記の処理が行われ、 取得したマクドナルドの店舗の位置情報をプロットしてくれる。

表示時間の問題についてはこれで解決したが、もう一つ問題がある。それは、地図を動かしたりした際に円が上書きされ、どんどん濃くなることである(図5)。それについては、次の2-2で解決する。



図 4. デベロッパーツールの「Network」タブから PHP を見たところ。地図に表示されている 7 店 舗分の情報(赤枠部)のみ取得していることが分かる。



図 5. Google マップを数回動かしたときの図。円が何度も上書きされ、黒くなってしまっている。

円を上書きしないで再描画する

2-2 では、表示時間短縮やデータ量節減のために地図の表示領域内の店舗情報のみを取得して 地図に表示させた。その結果、地図を動かすたびに処理が行われて円が上書きされるので、円が どんどん濃くなってしまうという問題が発生する。これを解決するために、円を描きなおす際に一度 既に描いてある円を削除する処理を行う。今回は円を描く処理に関することなので、「main.js」の みを書き換える。

```
'strict'
var map = null;
+var objects = [];
function initMap() {
 map = new google.maps.Map($('#map').get(0), {
   center: {lat: 43.068543, lng: 141.351128},
   zoom: 15
   });
 google.maps.event.addListener(map, 'idle', idleMaps);
}
function idleMaps()
 var zoom = map.getZoom();
 var bounds = map.getBounds();
 var ne = bounds.getNorthEast();
 var sw = bounds.getSouthWest();
 var params = 'zoom=' + zoom + '&n=' + ne.lat() + '&s=' + sw.lat() + '&e='
+ ne.lng() + '&w=' + sw.lng();
  $.ajax({
   url: 'api.php?' + params,
   type: 'GET',
   dataType: 'json'
  })
  .done(function(data){
   console.log(data);
   render(data);
  })
  .fail(function(){
   console.log('error');
  });
}
function render(data){
+ objects.forEach(function(o){
   o.setMap(null);
+
+
 });
+
 data.forEach(function(geom){
```

```
var circle = new google.maps.Circle({
    map: map,
    center: {lat: geom.coordinates[1], lng: geom.coordinates[0]},
    radius: 100
    });
+
+
+ objects.push(circle);
});
```

ここでは、プログラム 39 行目で既に描いてある円を一度消去し、49 行目で再び円を描いている。 上手くいけば、何度地図を動かしたりズームレベルを変えたりしても円が濃くならなくなる。

追加演習

これまでの演習では、Google マップ上に自分で緯度経度を決めて円を描く方法と、外部から取得 した緯度経度に円を描く方法を学んだ。円の半径(radius)については実際の距離を m 単位で設 定できる。したがって、「〇〇駅から半径 500m の範囲を図示する」といったことも可能である。 そこで、札幌駅から半径 3km 圏内にあるマクドナルド店舗を知るため、追加演習として

「Google マップ上にマクドナルド店舗の位置情報をプロットし、なおかつ札幌駅から半径 3km の範囲を図示する」

ことにチャレンジしてみよう。これまでの演習と違って、正解のソースコード全文は書かずにヒントのみに留めておくので、ヒントを参考にしながら考えてみてほしい(もちろん質問したり調べたりするのは自由)。

ヒント

- 今回は外部データの取得方法を変えたりしないので、「main.js」のみ変更すればよい。変更 する元の「main.js」は、MySQL・2-2 と同じものである。
- 札幌駅を中心とした円を描画するのは、GoogleMapsAPI・2・1 でやったように、地図そのもの を描画するタイミングで行う。ちなみに、地図の中心は札幌駅である。
- 半径 3km の円がすっぽり入るよう、ズームレベルも変える。
- デフォルトの円の色は黒であるが、半径 3kmの円とマクドナルドのプロットが同じ色では見づらい。そこで、「strokeColor」オプションで円のフチの色が、「fillColor」オプションで円の塗りつぶしの色が変更できるので、上手く色を変えてみてほしい。また、店舗のプロットであれば円の中の塗りつぶしは不透明でもよいだろう。塗りつぶしの透明度を変えるには、「fillOpacity」オプションを使う。
- マクドナルドの位置情報のプロットについては、Google マップでよく見るピンを使う方法もある。この場合、マクドナルドの位置情報を描画するクラスを「Circle」から「Marker」に変える。「Marker」クラスでは、地図を指定する「map」オプションとピンの置き場所を指定する「position」オプションを設定する。

今回のヒントで紹介したクラスやオプションについての詳細は、「Google Maps JavaScript API の使い方まとめ - SYNCER」(<u>https://lab.syncer.jp/Web/API/Google_Maps/JavaScript/</u>)を見 るとよい。

参考資料

- FOSS4G Hokkaido 2019 ハンズオン「JavaScriptとMySQL で GIS を作ってみよう」スラ イド資料 http://bit.ly/foss4g-2019-mysql
- FOSS4G Hokkaido 2019 ハンズオン「JavaScriptとMySQL で GIS を作ってみよう」演習 資料
 http://bit.ly/git-web-gis
- 今さら聞けない! PHP とは【初心者向け】 TechAcademy マガジン <u>https://techacademy.jp/magazine/6618</u>
- JSON の紹介
 <u>https://www.json.org/json-ja.html</u>
- GeoJSON ·GIS 実習オープン教材
 <u>https://gis-oer.github.io/gitbook/book/materials/web_gis/GeoJSON/GeoJSON.html</u>