

プログラミング演習(JavaScript)②【発展】オブジェクトなど

■オブジェクト指向プログラミング

処理の流れに沿ってプログラムを作成する方法を手続き型プログラミングと言う。これに対して、関連するデータ(変数)とそれに対する操作(関数)をまとめてそれを一つのモノ(オブジェクト)として捉え、それをもとにプログラムを作成する方法をオブジェクトプログラミングという。オブジェクトを構成するデータをプロパティと呼び、オブジェクトに対する操作を記述した関数をメソッドと呼ぶ。

例えば、高校野球の選手には、年齢、背番号、打順、打率などのその人を構成するデータや、走る、打つ、ポジション(背番号)を変えるといった機能や操作(メソッド)があり、こうした選手に関するデータをまとめて「選手オブジェクト」というものを作ることができる。このオブジェクトが持つデータや機能(メソッド)をプロパティという。

■オブジェクトを利用する

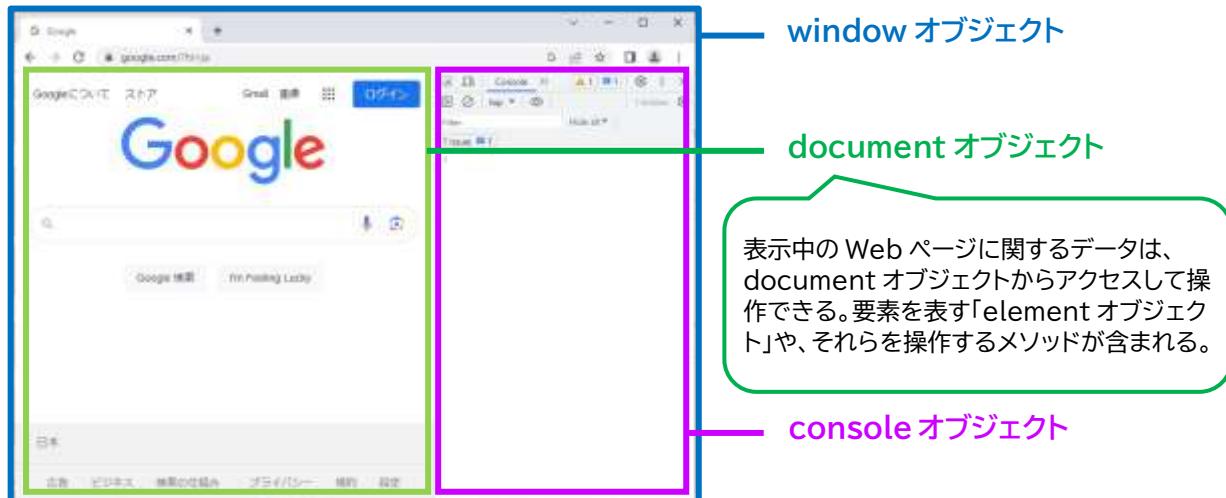
先述の「選手オブジェクト」は、自分に関するプロパティ(データや機能・操作)を持っていて、その中には「走る」といった操作(メソッド)も含まれる。こうしたオブジェクトの持つプロパティを利用するには「オブジェクト名.プロパティ名」という形でドット「.」でつないで記述する。例えば、選手に走ってほしい時は、「選手.走る」という具合にプログラムを書く。

また、選手オブジェクトのプロパティには「背番号」があり、これも操作ができるオブジェクトである。このようなオブジェクトの中にオブジェクトがある場合、ここでは選手オブジェクトの中に背番号オブジェクトある場合、背番号オブジェクトが持つ背番号を変更する「番号変更メソッド」を利用するとときは、「選手.背番号.番号変更(3)」というふうに書く。

■Web ページとオブジェクトの関係

JavaScript には、ブラウザに関するあらゆるデータをまとめた、ブラウザのウィンドウ自体を表す「window オブジェクト」が初めから用意されている。window オブジェクトを利用すると、Web ページの HTML や CSS の情報を取得したり、操作したりすることができる。

●window オブジェクトとブラウザの対応



なお、window.は省略でき、「document.write("Hello World!");」も「window.document.write("Hello World!");」も同じ結果になる。

参考:いちばんやさしい JavaScript の教本 第 2 版 岩田宇史著(インプレス)

6. クラス、オブジェクト

何かを作るとき、設計図が必要となる。この設計図にあたるものをクラスといい、クラスをもとに作られたものをオブジェクト、インスタンスといい、クラスをもとにしてオブジェクトを作ることを、オブジェクトの生成、インスタンス化という。

クラスを作るには、まずオブジェクト定義を行う。クラスの定義はコンストラクタ、ゲッタメソッド、セッタメソッド、メソッドなどからなる。

●コンストラクタ

クラス内で宣言される変数をフィールドという。「var オブジェクト名=new クラス名(引数)」で呼び出すことができる。

```
constructor(引数){  
    this.フィールド名 = 引数;  
}
```

●ゲッタメソッド

フィールドの値を取得できるメソッド。「オブジェクト名.プロパティ名」で呼び出すことができる。

```
get プロパティ名(){  
    return this.フィールド名;  
}
```

●セッタメソッド

フィールドに値を設定するためのメソッドである。「オブジェクト名.プロパティ名=値;」で設定できる。セッタメソッドはいつでもフィールドに値を設定できる。

```
set プロパティ名(引数){  
    this.フィールド名=引数;  
}
```

●メソッド

ここでは、普通のメソッドを定義する。メソッドは「オブジェクト名.メソッド名(引数)」で呼び出すことができる。

```
メソッド名(引数){  
    <ここにメソッドの処理を書く>  
    return 戻り値;  
}
```

senshu という名前のクラスに、name と seban というフィールド(キー)を設定し、

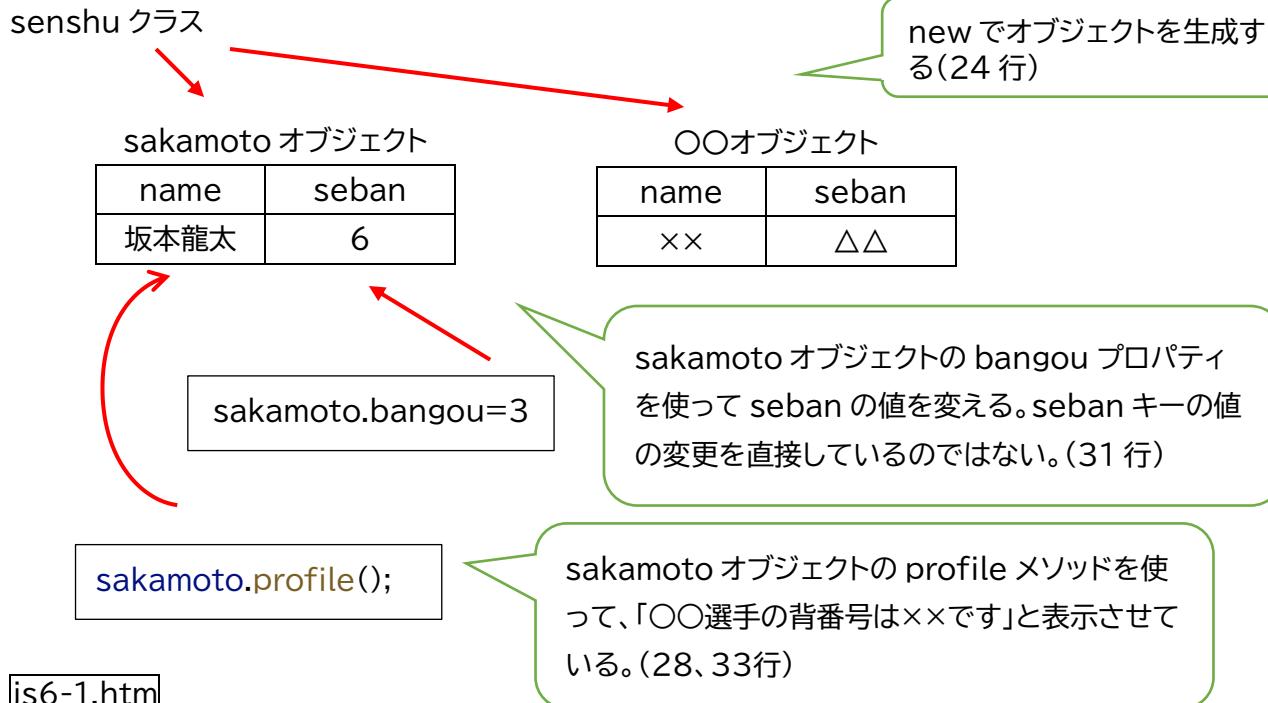
seban を変更したり、取得したりするのに、bangou プロパティを設定した。

sakamoto というオブジェクトを作り、name の値(バリュー)を坂本龍太、seban の値(バリュー)を 6 として、ここでは、sakamoto.bangou を使って、seban を表示させたり、入力した値を代入したりしている。

profile メソッドを作り、sakamoto.profile(); と書いた場合、

this.name(ここでは sakamoto.name の「坂本龍太」)選手の背番号は、

this.seban(ここでは sakamoto.seban の「6」) です。と表示されるようにしている。



js6-1.htm

```

009  class Senshu{
010    constructor(n,s){
011      this.name = n;
012      this.seban = s;
013    }
014    get bangou(){
015      return this.seban;
016    }
017    set bangou(s){
018      this.seban = s;
019    }
020    profile(){
021      alert(this.name + '選手の背番号は' + this.seban + 'です');
022    }
023  }
024  var sakamoto = new Senshu('坂本龍太', '6');
025  alert('現在の sakamoto.bangou を表示します');
026  alert(sakamoto.bangou);
027  alert('sakamoto.profile()を表示します');
028  sakamoto.profile();
029  var nb = Number(prompt('坂本選手の新しい背番号を入力してください', '')); 
030  alert('sakamoto.bangou に' + nb + 'を代入します');
031  sakamoto.bangou = nb;
032  alert('sakamoto.profile()を表示します');

```

コンストラクタ

ゲッタメソッド

セッタメソッド

メソッド

033	<code>sakamoto.profile();</code>
-----	----------------------------------

■組み込みオブジェクト(クラス)

あらかじめ組み込まれたオブジェクト。入力された値の平方根を求めるには、下記のように入力する。

```
var hei = Number(prompt('何の平方根を求めますか?', ''));  
alert(Math.sqrt(hei));
```

時刻を扱うときは、

```
var today = new Date();  
alert(today.toDateString());
```

■アラートで時刻を表示した後、HTML に時刻を「時:分:秒」という形で表示させる

js6-2.htm

008	<code><p id="time"></p></code>
009	<code><script></code>
010	<code>var today = new Date();</code>
011	<code>alert(today.toDateString());</code>
012	<code>var hour = today.getHours();</code>
013	<code>var min = today.getMinutes();</code>
014	<code>var sec = today.getSeconds();</code>
015	<code>time.innerHTML= hour+':'+min+':'+sec;</code>

■時刻の自動更新

js6-2-2.htm

008	<code><p id="time"></p></code>
009	<code><script></code>
010	<code>setInterval(function(){</code>
011	<code>var today = new Date();</code>
012	<code>var hour = today.getHours();</code>
013	<code>var min = today.getMinutes();</code>
014	<code>var sec = today.getSeconds();</code>
015	<code>time.innerHTML= hour+':'+min+':'+sec;</code>
016	<code>},1000);</code>

●ゲームを作る

10の倍数でボタンを押したら成功として、表示させる。そうでなかつたら失敗とする。

js6-3.htm

008	<code>0</code>
009	<code><button id='action_btn'>スタート</button></code>

```
010 <ul id="results"></ul>
011 <script>
012     var action_btn = document.getElementById('action_btn');
013     var counter = document.getElementById('counter');
014     var count=1;
015     var interval_id;
016     function start_game(){
017         var btn_text = action_btn.innerHTML;
018         if(btn_text == 'スタート'){
019             interval_id=setInterval(function(){
020                 count++;
021                 counter.innerHTML=count;
022             },200);
023             action_btn.innerHTML='ストップ';
024         }else if(btn_text == 'ストップ'){
025             clearInterval(interval_id);
026             action_btn.innerHTML='スタート';
027             var li = document.createElement('li');
028             if (count % 10 == 0){
029                 var text = document.createTextNode('カウント:' +count+ ' 結果: 成功');
030             } else {
031                 var text = document.createTextNode('カウント:' +count+ ' 結果: 失敗');
032             }
033             li.appendChild(text);
034             results.appendChild(li);
035             count=1;
036             counter.innerHTML=count;
037         }
038     }
039     action_btn.addEventListener('click',start_game);
```

7. 状態遷移図から作る

下図の状態遷移図からプログラムを作成する

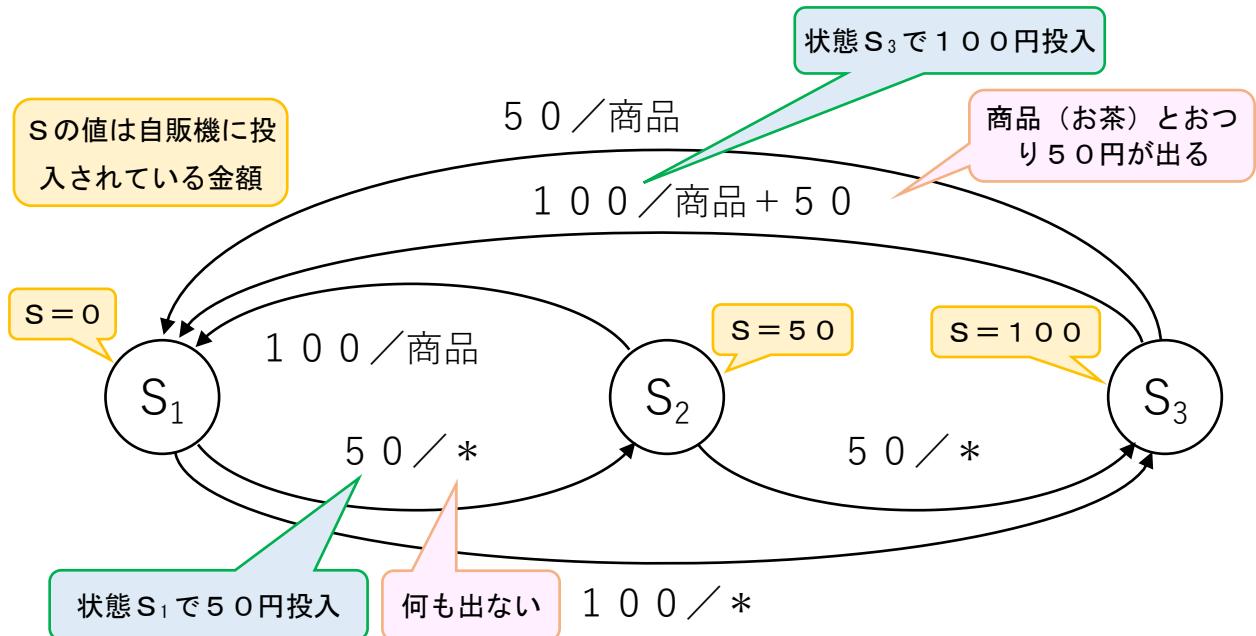


図 7-1 150 円のペットボトルを販売する自動販売機

js7-1.htm

```

007 <body bgcolor="#aaffaa">
008 
009 <script defer>
010 alert('いらっしゃいませ！');
011 alert('お茶のペットボトルの自動販売機です');
012 alert('金額は150円です');
013 alert('この自販機は50円と100円しか入れられません');
014 function jihanki(x){
015   if(s==0 && x == 50){
016     s=50; alert('投入金額 ' + s + '円');
017   }else if ( s == 50 && x == 50 || s == 0 && x == 100 ){
018     s = 100; alert('投入金額 ' + s + '円');
019   }else if ( s == 50 && x == 100 || s == 100 && x == 50 ){
020     s = 0; alert('ペットボトルが出てきました');
021   }else if ( s == 100 && x == 100){
022     s = 0; alert('ペットボトルとおつり 50 円が出てきました');
023   }
  
```

注釈:

- 累計 0 で入金 50 の時は累計 50
- 累計 50 で入金 50 の時または累計 0 で入金 100 の時は累計 100
- 累計 50 で入金 100 の時または累計 100 で入金 50 の時は累計 0 でペットボトル

```
024 }
025 var s=0;
026 while(true){
027     var i= Number(prompt('金額の入力(50 or 100 , 終了は0)', ''));
028     if (i == 50 || i == 100){
029         jihanki(i);
030     }else if( i==0 ){
031         alert('ご利用ありがとうございました');
032         break;
033     }
034 }
```

true にしかならないので、無限にループする

累計 100 で入金100の時はペットボトルとおつり50

i=0 にすると、無限ループから抜けられる

【問題】 このプログラムでは、画像が alert のメッセージよりの後に表示される。画像をメッセージより先に表示させるにはどうしたらよいか考え、プログラムを直せ。うまく表示されない場合は、なぜうまくいかないか考えよ。

8. オープンデータ(CSV ファイル)を利用する

気象庁の Web サイトからダウンロードした日別の最高気温のデータをファイルから読み込んで、日付と気温のデータだけにして、入力した気温以上を観測した年月日とその気温をブラウザに表示する

■ CSV ファイルの準備

ブラウザで、静岡県の過去の気象データを検索する。

キーワード「過去の気象データ・ダウンロード」

気象庁|過去の気象データ・ダウンロード のページをクリック



検索条件を設定する

- ・場所 静岡(県)→静岡(市)
- ・データの種類 日別値、気温タブの「日最高気温」をチェック
- ・期間 連続した期間で表示する 2016年4月1日 から 2021年3月31日
- ・表示オプション その他の都道府県名を格納(CSVファイルダウンロード実行時)にチェック

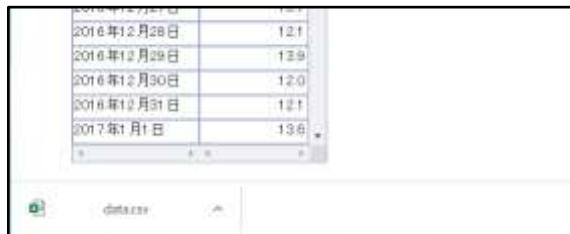
右側の画面に表示ボタンを押す



表示されたデータを確認し、右側の CSV ファイルをダウンロードボタンを押す



画面下にダウンロードされたファイルが表示されるので、右側の上向きの矢印を選択する



フォルダを開くを選択し、data.csv ファイルをコピーして、任意の場所に保存する



data.csv ファイルをダブルクリックすると Excel が起動し、ファイルを開く

	A	B	C	D	E
1	ダウンロードした時刻 : 2021/05/03 01:14:47				
2					
3		静岡	静岡	静岡	
4		静岡	静岡	静岡	
5	年月日	最高気温(°C)	最高気温(°C)	最高気温(°C)	
6					
7		品質情報	均質番号		
8	2016/4/1	16.7	8	1	
9	2016/4/2	16.2	8	1	
10	2016/4/3	19.5	8	1	
11	2016/4/4	21.2	8	1	
12	2016/4/5	17.8	8	1	
13	2016/4/6	20.5	8	1	
14	2016/4/7	21.1	8	1	
15	2016/4/8	21	8	1	

年月日と最高気温以外は消し、ファイル名前を付けて保存 「data2.csv」という名前で保存する

	A	B	C	D	E
1	2016/4/1	16.7			
2	2016/4/2	16.2			
3	2016/4/3	19.5			
4	2016/4/4	21.2			
5	2016/4/5	17.8			
6	2016/4/6	20.5			
7	2016/4/7	21.1			
8	2016/4/8	21			

●プログラムを作る

CSV ファイルは書く値を半角カンマで区切り、行末に改行コードの「¥r¥n」を付けて1行のデータとしている。この場合、1行ずつ配列に代入して行き、各行ごと半角カンマの位置で別の配列に代入すればよい。



js8-1.htm

```

008 <p><input type="file" id="file"></p>          ファイル選択領域の作成
009 <div id = "text"></div>                      矩形領域の作成
010 <script>
011     function fileChange(event){           fileChange 関数を定義。引数 event は発生
012         var file =event.target.files;      したイベント。下の2行が実行される
013         reader.readAsText(file[0]);        reader の readAsText メソッドを使ってファイルを文
014     }                                         字列として reader に読み込む。file[0] は選択された
015     function fileLoad(event){           ファイルを表す File オブジェクト
016         var t1 = parseFloat(prompt('気温[°C]', ''));  選択されたファイルは1つだけなので file[0] しかない
017         var txt = '';
018         var lines =reader.result.split('¥r¥n');    ファイル読み込み完了時に実行される fileLoad 関数を定義。
019         for (var i=0 ; i < lines.length - 1 ; i++){
020             var a= lines[i].split(',');           16~24行が処理される。event は関数で利用しない
021             var date = a[0];
022             var t2 = parseFloat(a[1]);          読み込み結果を取り出したものを入れた
023             if (t2 >= t1){                   reader.result
024                 txt = txt + date + ' : ' + a[1] + '<br>'  (reader の result プロパティ)
025             }                                         から、区切り文字を¥r¥n として、1ずつ
026         }                                         lines 配列の各要素に取り出される
027         contents.innerHTML = txt;           lines[i] の中の半角カンマを区切り文字とし
028     }                                         て、日付を a[0] に、気温を a[1] に代入する
029         var inputFile = document.getElementById('file');  大きさを比較できるように文字列 a[1] を実数
030         var contents = document.getElementById('text');    に変換して t2 に代入する
031         var reader = new FileReader();           t1 と t2 を比較して、t2 ≥ t1 だったら、txt
032         inputFile.addEventListener('change',fileChange);  にその行の値を足していく。
033         reader.addEventListener('load',fileLoad);        矩形領域の contents オブジェクトに txt を
034     </script>                                代入して、矩形領域に文字列 text を表示する
                                                
```

インスタンス化

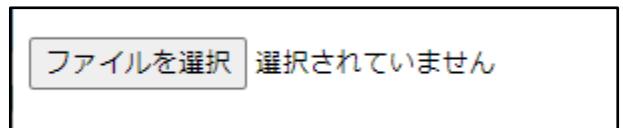
FlieReader クラスから FlieReader オブジェクトの reader を生成する

ファイルを選択した時に 11 行目の関数 fileChange を実行する

ファイルを読み込んだ時に 15 行目の関数 fileLoad を実行する

●プログラムの実行す

js_8-1.htm を開いて、ファイルを選択ボタンを押し、data2.csv を選択する



半角で 30 と入力する



指定した温度以上の年月日と最高温度が表示される



9. 連想配列

NoSQL でと呼ばれるデータベースがあり、そのうちのキーバリュー型データベースでは、キー(項目名)とバリュー(値)の組を書き込みキーを指定することでそのキーに紐づいたバリューを読み出せることが特徴である。JavaScript では、キーバリュー型のデータ構造は連想配列で扱う。

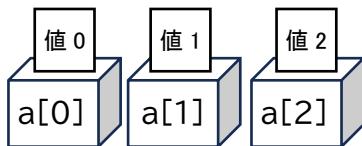


図 9-1 配列

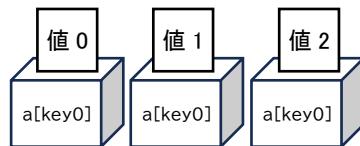


図 9-2 連想配列

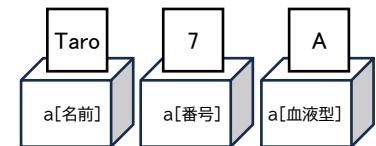


図 9-3 連想配列の例

表9-1 連想配列の扱い方

連想配列の宣言	<code>var 連想配列名 = {キー0:バリュー0, キー1:バリュー1, …};</code>
連想配列の参照	<code>連想配列名['キー名']</code> または <code>連想配列名.キー名</code>
値の取り出し	<code>for(変数 in 連想配列){ 処理 }</code>

ここで、連想配列を使って

名前:坂本龍太

背番号:3

部活動:ビジネス研究部

と表示するプログラムを作る

js9-1.htm

```
009 var a = {'名前':'坂本龍太',  
010     '背番号':'3',  
011     '部活動':'ビジネス研究部'};  
012 for (var key in a){  
013     document.write(key + ' : ' + a[key] + '<br>');  
014 }
```

連想配列 a の宣言
連想配列から、名前、背番号、部活動の順で取り出しながら、13 行を繰り返す
key とバリューの取り出し

なお、配列変数の宣言は下記のように、からの連想配列を作つてから代入してもよい。

```
009 var a = {};  
010 a['名前']='坂本龍太';  
011 a['背番号']='3';  
012 a['部活動']='ビジネス研究部';
```

ファイルを読み込んだ時に 15 行目の
関数 fileLoad を実行する

■二次元連想配列

9-1のキーを使った連想配列を複数作るために、二次元連想配列を使う

表9-2 二次元連想配列

i	key		
	名前	背番号	部活
0	坂本龍太	3	ビジネス研究部
1	松原忠史	7	剣道部
2	大城平三	2	ボランティア同好会

表9-3 二次元連想配列の扱い方

二次元連想配列の宣言	var 二次元連想配列名=[{キー00:バリュー00, キー01:バリュー01,…}, {キー10:バリュー10, キー11:バリュー11,…}, …];
二次元連想配列の参照	二次元連想配列名['キー名'] または 連想配列名.キー名
値の取り出し	<pre>for (var i=0; i < a.length; i++){ for(変数 in 二次元連想配列[i]){ 处理 } }</pre> <div style="border: 1px solid green; padding: 5px; margin-left: 200px;"> すべて取り出すために添え字を0からループさせている。一つだけ取り出すなら必要ない </div> <div style="border: 1px solid green; padding: 5px; margin-left: 200px;"> 添え字を付けて参照する </div>

js9-2.htm

```

009      var a = [];
010      a.push( {'名前':'坂本龍太',
011              '背番号':'3',
012              '部活動':'ビジネス研究部' } );
013      a.push( {'名前':'松原忠史',
014              '背番号':'7',
015              '部活動':'剣道部' } );
016      a.push( {'名前':'大城平三',
017              '背番号':'2',
018              '部活動':'ボランティア同好会' } );
019      for (var i=0; i < a.length; i++){
020          for (var key in a[i]){
021              document.write(key + ' : ' + a[i][key] + '<br>');
022          }
023          document.write('<br>');
024      }

```

【問題】次の表の二次元連想配列を表示させるプログラムを作れ

表9-4 二次元連想配列2

key				
i	名前	背番号	部活	ひとこと
0	坂本龍太	3	ビジネス研究部	視野を広げたい
1	松原忠史	7	剣道部	天誅
2	大城平三	2	ボランティア同好会	
3	梶谷城	8	歴史研究部	九州遠征したい
4	岡本綺太郎	6	化学部	試合の前に花火を上げたい

10.グラフを作る

グラフは、Canvas で自作することもできるが、ここでは、グラフ作成ライブラリを読み込んでグラフを描画する。

参考:あんこライブラリ:<https://anko.education/reference/plotly-js>
からくりブログ:<https://www.kara-kuri.jp/blog/?p=2721>

10-1 棒グラフ

表10-1 喫茶リンゴの1週間の売り上げ

日付・曜日		4/5(月)	4/7(水)	4/8(木)	4/9(金)	4/10(土)	4/11(日)
売り上げ(円)		8000	13000	10000	20000	35000	50000

js10-1.htm

```
008  <div id="chart"></div>
009  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
010  <script>
011  let trace1 = {
012    name : "リンゴ",
013    x: ['4/5(月)', '4/7(水)', '4/8(木)', '4/9(金)', '4/10(土)', '4/11(日)'],
014    y: [8000, 13000, 10000, 20000, 35000, 50000],
015    type: 'bar',
016    width: 0.5
017  };
018  let data = [trace1];
019  let layout = {
020    hovermode: 'closest',
021    title: '喫茶リンゴの売上',
```

```

022  xaxis: {
023      title: '日付'
024  },
025  yaxis: {
026      title: '売上金額'
027  }
028 };
029 Plotly.newPlot('chart', data, layout);
030 </script>

```

10-2 折れ線グラフ

type を'scatter'にして、2つの店舗のデータを変数 data に入れて出力する

表10-2 喫茶リンゴとブティック島崎の年別売り上げ推移(月ベース)

年		2019	2020	2021	2022	2023
売り上げ(万円)	リンゴ	85	30	41	39	40
	島崎	120	85	180	200	205

js10-2.htm

```

008  <div id="chart"></div>
009  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
010 <script>
011 let timestamp = [
012   '2019', '2020', '2021', '2022', '2023'
013 ];
014 let trace1 = {
015   name : "リンゴ",
016   x: timestamp,
017   y: [85, 30, 41, 39, 40],
018   type: 'scatter'
019 };
020 let trace2 = {
021   name : "島崎",
022   x: timestamp,
023   y: [120, 85, 180, 200, 210],
024   type: 'scatter'
025 };
026 let data = [trace1, trace2];

```

```

027 let layout = {
028   title: '年別売り上げ推移(月ベース)',
029   xaxis: {
030     title: {
031       text: '年',
032     },
033   },
034   yaxis: {
035     title: {
036       text: '売上金額',
037     }
038   }
039 };
040 Plotly.newPlot('chart', data, layout);
041 </script>

```

【問題】

2店舗のデータを読み、問題点をあげ、その問題を解決するための方法を考えよ。

■店舗情報

・1階:ブティック島崎（婦人洋品販売、販売代行(シャルル)、洋服のお直し）

　経営者:島崎若菜(42) アルバイト:彩(27) 手伝い 息子の島崎駿(草薙高校2年)

　メモ:アルバイトを正規の従業員として雇いたい気持ちがある

・2階:喫茶リンゴ(喫茶店とイベントの貸しスペース・ドラムセット使用可能、アルコール類提供可能)

　経営者:ハリー藤村(67) 手伝い 孫の藤村凜(草薙高校2年) 夜のイベント時:近所の老人たち

　メモ:コロナ禍以降、経営難に陥っている。深夜まで営業するのがつらくなっている。