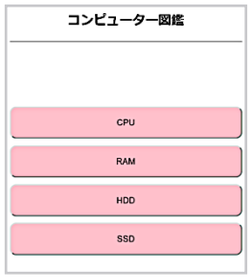
アプリプログラミングシート ~図鑑~

アプリの概要

　コンピュータ用語の図鑑アプリを作成する。このアプリは、トップページと各項目のページそれぞれのボタンを押すことでページ移動ができるアプリである。

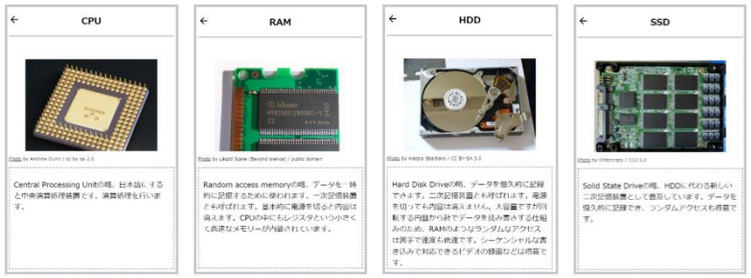
　使われているＨＴＭＬ、ＣＳＳでのページの作成方法とページ間を移動する方法（ハイパーリンク）について学ぶ。

トップページ



リンク

個別項目ページ

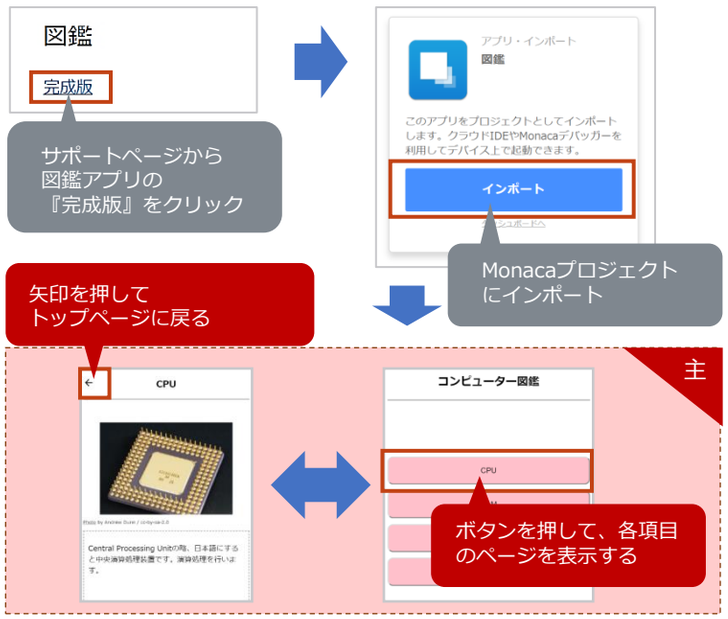


学習内容

【主】HTMLの基本的な構造とハイパーリンク(リンク)で別ページを表示する方法
【副】その他関連する書籍対応表（右表）

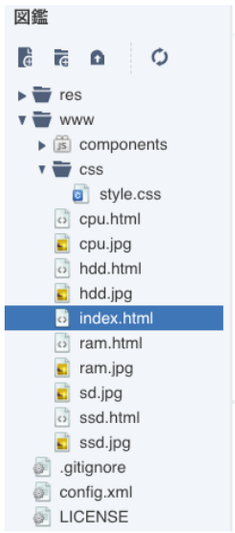
書籍	関連する内容
第1章 アプリ開発入門	Monacaで図鑑プロジェクトを編集する
第2章 HTML入門	ボタンはHTMLで用意されている
第3章 CSS入門	ボタンの色や配置を設定している

図鑑アプリを動かしてみよう



ファイル一覧

トップページを含むすべてのページは、ＨＴＭＬで作成されている。



index.htmlを読んでみよう

```
<body>
    <nav>
        <div> </div>
        <h1>コンピューター図鑑</h1>
        <div> </div>
    </nav>
    <div id="menu">
        <a href="cpu.html"><button>CPU</button></a>
        <a href="ram.html"><button>RAM</button></a>
        <a href="hdd.html"><button>HDD</button></a>
        <a href="ssd.html"><button>SSD</button></a>
    </div>
</body>
```

①ナビゲーション部分

②メニュー部分

③ボタンにリンクを設定

cpu.htmlを読んでみよう

```
<body>
    <nav>
        <a href="index.html">
          <div class="material-icons">
            arrow_back
          </div>
        </a>
        <h1>CPU</h1>
        <div></div>
    </nav>
    <div id="mount">
        <img src="cpu.jpg">
    </div>
    <p id="license">
      <a href="https://commons.wikimedia.org/wiki/File:Intel_80486DX2_bottom.jpg">Photo</a>
      by Andrew Dunn / cc-by-sa-2.0</p>
    <div id="explain">
        <p>
          Central Processing Unitの略、日本語にすると中央演算処理装置です。
          演算処理を行います。
        </p>
    </div>
</body>
```

④ CSSで arrow_back をアイコンに変換

⑤ CPUの画像を表示

カスタマイズ①
トップページの項目と新しいページを追加をしてみよう

カスタマイズの概要

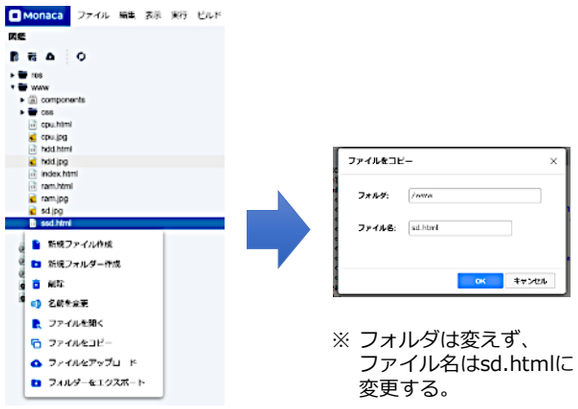
例：「SD Card」のページを追加してみよう

変更点

- sd.htmlページを作成する
  ※ 適当なファイル(例ではssd.html)をコピーし、その内容を変更して作成する。
- index.htmlにボタンを追加する

ファイルのコピー方法

ファイル一覧で、コピー元ファイルを右クリックすることでコピーできる。



※ フォルダは変えず、ファイル名はsd.htmlに変更する。

index.htmlの変更例

```
<div id="menu">
    <a href="cpu.html"><button>CPU</button></a>
    <a href="ram.html"><button>RAM</button></a>
    <a href="hdd.html"><button>HDD</button></a>
    <a href="ssd.html"><button>SSD</button></a>
    <a href="sd.html"><button>SD Card</button></a>
</div>
```

ボタンとリンクの追加

新しい項目ページ作成時の変更箇所と変更例

```
<nav>
    <a href="index.html"><div class="material-icons">arrow_back</div></a>
    <h1>SSD</h1>
    <div></div>
</nav>
<div id="mount">
    <img src="ssd.jpg">
</div>
<p id="license">
    <a href="https://commons.wikimedia.org/wiki/File:Embedded_World_2014_SSD.jpg">
      Photo
    </a>
    by Ordercrazy / CC0 1.0
</p>
<div id="explain">
    <p>
      Solid State Driveの略、HDDに代わる新しい二次記憶装置として普及しています。
      データを恒久的に記録でき、ランダムアクセスも得意です。
    </p>
</div>
```

h1(大見出し)の変更

大見出しの変更例
<h1>SD Card</h1>

画像の変更

画像の変更例
<img src=SD.jpg">

ライセンス表記の変更

ライセンス表記の変更例

Photo

by Rstoplabe14 / public domain

※今回の画像は、著作権が放棄された画像（パブリックドメイン）のため、ライセンス表記は削除可。

説明文の変更

説明文の表記例
挿抜を容易に行えるカード型の補助記憶装置です。

⇒ **他にも自分でページを追加してみよう。**

カスタマイズ②
図鑑アプリの配色を変更してみよう

カスタマイズの概要

ＣＳＳを使ってトップページの配色を変更する。

変更点

- 背景色を『cyan』に変更
- ボタンの背景色を『red』に変更
- ボタンの文字色を『white』に変更

css/style.cssの変更例

```
body{
    background-color: cyan;
}

#menu button {
    width:100%;
    height:50px;
    padding:10px;
    margin-bottom:10px;
    border-radius:10px;
    color: white;
    background-color: red;
}
```

全体(bodyタグ内)の背景色を変更

ボタンの文字色を変更

ボタンの背景色を変更

※ 色は好きな色に変更して構いません。

カラーコードによる色指定

色の表現方法は、「red」や「blue」などの色の名称を指定する方法のほかに、カラーコードと呼ばれる色の指定方法がある。

カラーコードによる例

ff00ff

赤の強さ　緑の強さ　青の強さ

赤がff（255）、緑が00（0）、青がff（255）なので、
光の三原色の■赤と■青を混ぜた色（■マゼンタ）が色指定されます。

※16進数等、詳しくは書籍を参考

アプリプログラミングシート ~おみくじ~

アプリの概要

ボタンを押すと、**「大吉」「中吉」「凶」**などの結果をランダムに表示するアプリを作成する。

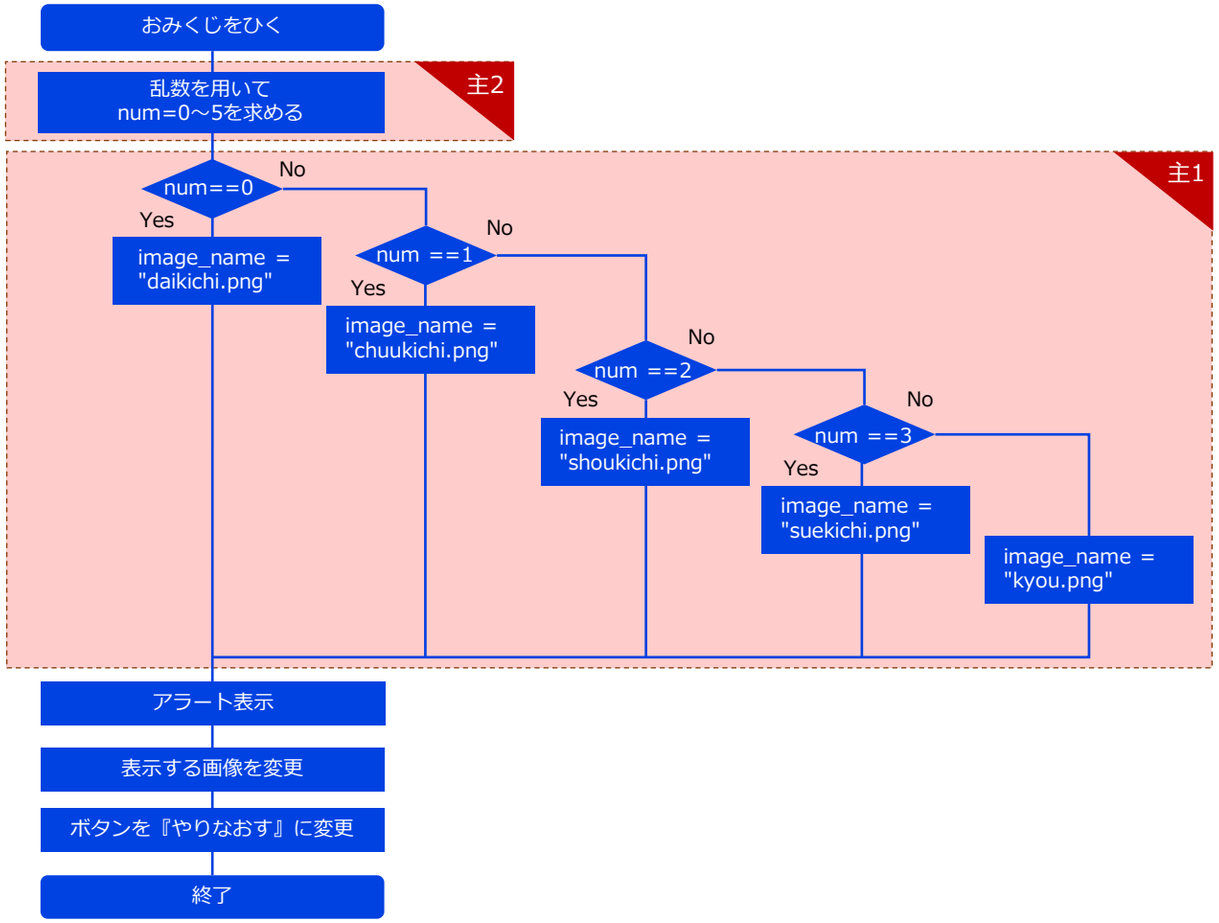
このアプリを題材として、ランダム値を取得する方法と、条件に応じて画像を差し替える方法を学ぶ。

学習内容

【主1】条件による分岐の方法
【主2】乱数の取得
【副】その他関連する内容(書籍対応表)

書籍	関連する内容
第1章 アプリ開発入門	Monacaでおみくじプロジェクトを編集する
第2章 HTML入門	画像やボタンはHTMLで用意されている
第3章 CSS入門	画像やボタンを中央寄せに整えている 背景画像を表示している
第4章 JavaScript入門	乱数や画像名を変数に代入している
第5章 条件分岐	乱数の値に応じて利用する画像を切り替えている
第6章 関数	全体をplay()関数としてまとめている
第7章 イベント	onclickイベントでplay()関数を呼び出している
第8章 DOM	画像やボタンの文字を書き換えている

おみくじのフローチャート(簡易)を確認しよう



おみくじアプリを動かしてみよう

プログラムを読んでみよう

```
<script>
    function play() {
        // 0～4の範囲のランダムな値を得る
        var num = Math.floor(Math.random() * 5);

        // ランダム値に応じて表示する画像を変える
        var image_name;
        if (num == 0) {
            image_name = "daikichi.png";
        } else if (num == 1) {
            image_name = "chuukichi.png";
        } else if (num == 2) {
            image_name = "shoukichi.png";
        } else if (num == 3) {
            image_name = "suekichi.png";
        } else {
            image_name = "kyou.png";
        }

        alert("おみくじが出ました！さて結果は？");
        // 画像と文字列の差し替え
        document.getElementById("omikuji").src = "images/" + image_name;
        document.getElementById("playBtn").innerHTML = "やりなおす";
    }
</script>
```

① 乱数を用いて基となる値を求める

② ①の値に応じて変数に画像を代入する

③ 代入した画像の表示とボタン名の変更

カスタマイズ① おみくじの結果を追加してみよう

カスタマイズの概要

おみくじの結果に**「大凶」**を加える。

変更点

- 乱数で求められる値の範囲を1つ増やす。
- 増やした値によって”daikyou.png”が表示されるようにする。

JavaScriptの変更

フローチャートで考えた変更をプログラムに反映する。

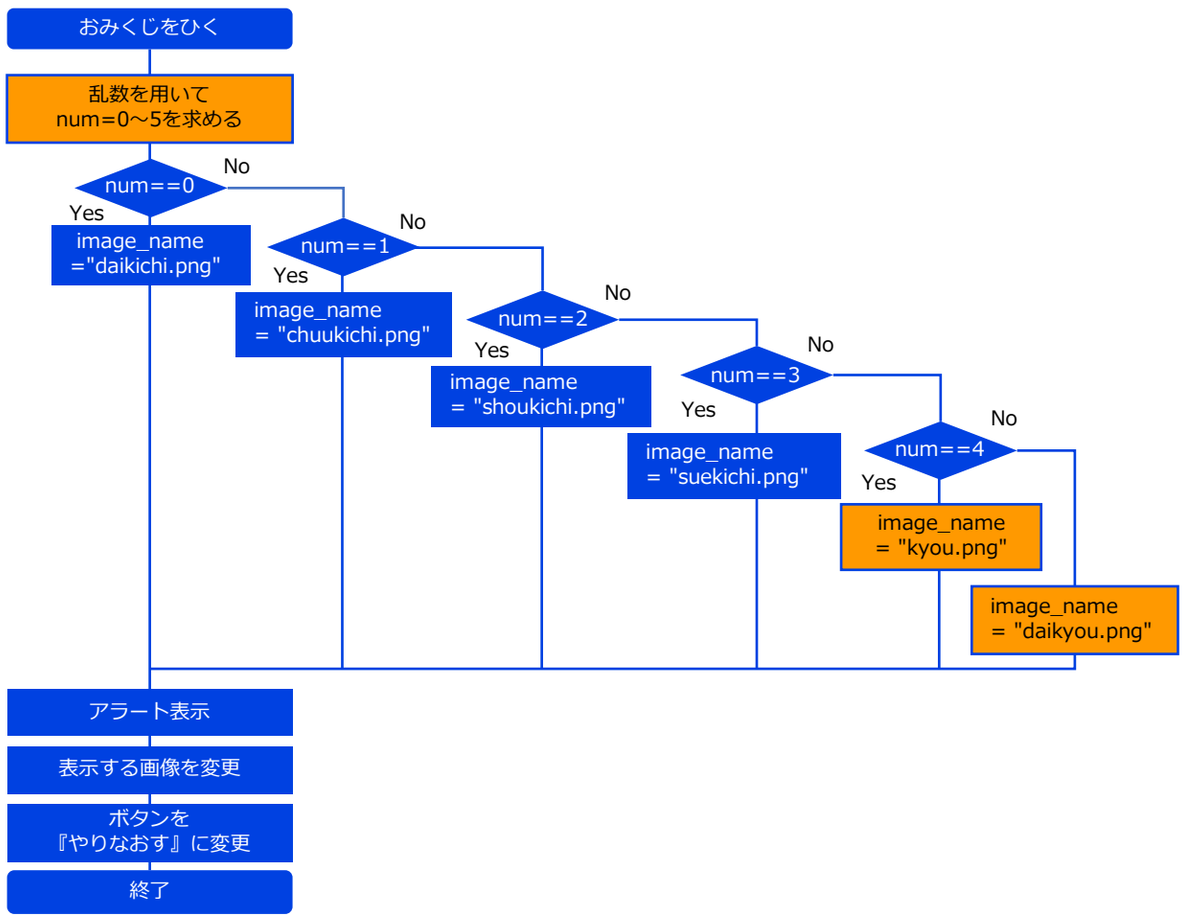
```
<script>
    function play() {
        // 0～5の範囲のランダムな値を得る
        var num = Math.floor(Math.random() * 6);

        // ランダム値に応じて表示する画像を変える
        var image_name;
        if (num == 0) {
            image_name = "daikichi.png";
        } else if (num == 1) {
            image_name = "chuukichi.png";
        } else if (num == 2) {
            image_name = "shoukichi.png";
        } else if (num == 3) {
            image_name = "suekichi.png";
        } else if (num == 4) {
            image_name = "kyou.png";
        } else {
            image_name = "daikyou.png";
        }

        alert("おみくじが出ました！さて結果は？");
        // 画像と文字列の差し替え
        document.getElementById("omikuji").src = "images/" + image_name;
        document.getElementById("playBtn").innerHTML = "やりなおす";
    }
</script>
```

変更点

凶の条件の変更

大凶の追加

フローチャート(簡易)で考えよう



カスタマイズ② おみくじに一言メッセージを追加してみよう

カスタマイズの概要

おみくじの結果に一言メッセージを加える。

メッセージを追加

変更点 【考えてみよう！】

-
-

JavaScriptの変更

```
// ランダム値に応じて表示する画像を変える
var image_name;
var message;
if (num == 0) {
    image_name = "daikichi.png";
    message = "大吉です";
} else if (num == 1) {
```

変数messageを追加

messageへ代入
(分岐ごとに)

【中略】

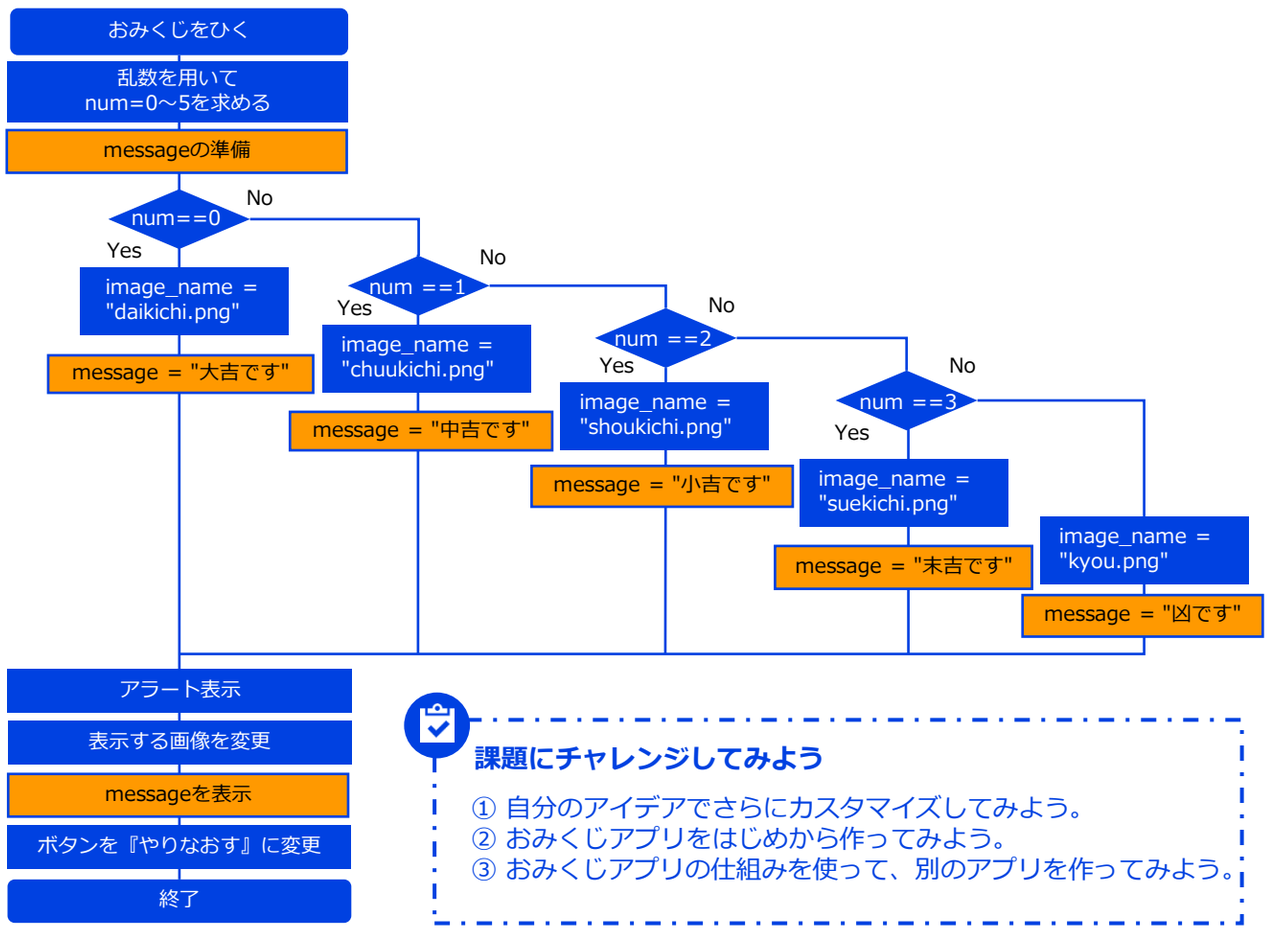
```
// 画像と文字列の差し替え
document.getElementById("omikuji").src = "images/" + image_name;
document.getElementById("message").innerHTML = message;
document.getElementById("playBtn").innerHTML = "やりなおす";
```

messageを表示

HTMLの変更

```
<body>
    <img src="images/omikuji-box.png" id="omikuji">
    <p id="message"></p>
    <button id="playBtn" onclick="play()">おみくじをひく<
</body>
```

メッセージを表示するため<p>タグを追加し、idをmessageにする



課題にチャレンジしてみよう

① 自分のアイデアでさらにカスタマイズしてみよう。
② おみくじアプリをはじめから作ってみよう。
③ おみくじアプリの仕組みを使って、別のアプリを作ってみよう。

アプリプログラミングシート ~貯金シミュレーションアプリ~

アプリの概要

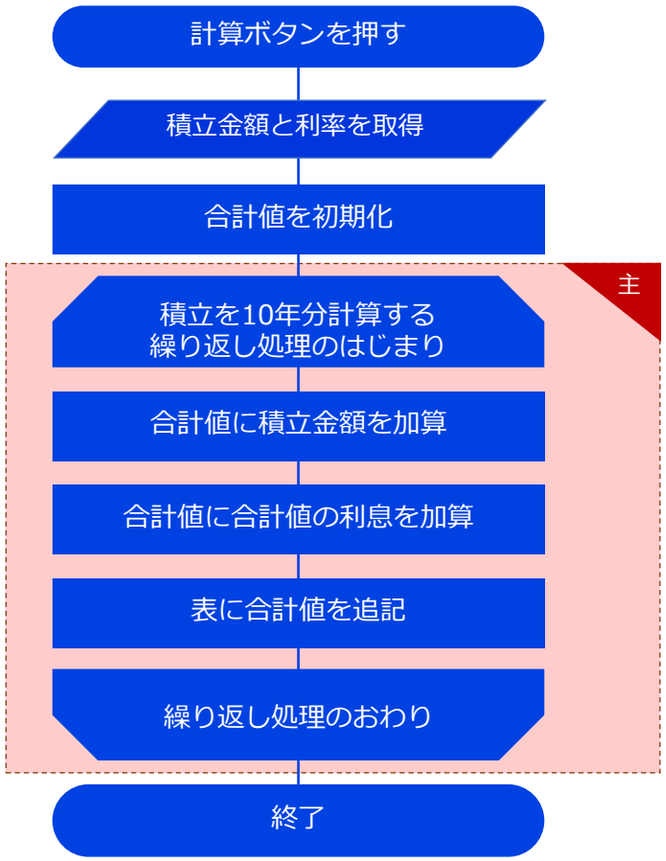
フォームに毎年積み立てる金額と利率を入力し計算ボタンを押すと、１年毎の貯金額の合計が表示されるアプリを作成する。

学習内容

【主】繰り返しの処理
【副】フォームの値を変数に取得
変数の値をHTMLに反映
その他関連する内容(書籍対応表)

書籍	関連する内容
第1章 アプリ開発入門	Monacaで貯金シュミレーションプロジェクトを編集する
第2章 HTML入門	ボタンはHTMLで用意されている
第3章 CSS入門	文章の表示領域を整えている
第4章 JavaScript入門	毎年の積立金額や利息、合計値を変数に代入している
第6章 関数	全体をcalcTest()関数としてまとめている
第7章 イベント	onclickイベントで calcInterest()関数を呼び出している
第9章 フォーム	毎年の金額や利息の入力欄を用意している
第10章 いろいろな演算子	合計値の計算で加算代入(+=)を行ったり 除算・乗算で利息の計算を行っている
第12章 繰り返し	積立を10年分計算する反復処理をfor文で行っている

複利計算のフローチャート(簡易)を確認しよう



貯金シミュレーションアプリを動かしてみよう

プログラムを読んでみよう

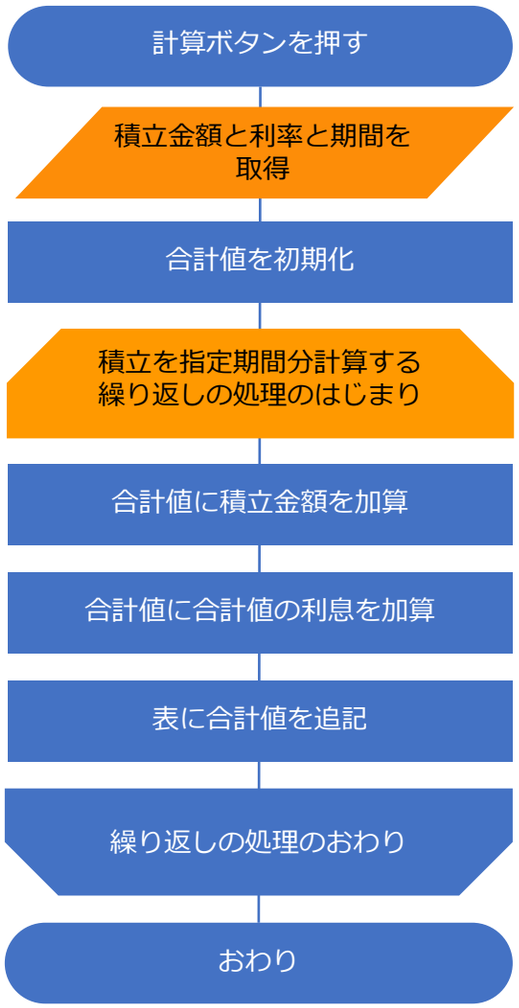
```
<script>
    function calcInterest() {                                   ① 関数を定義
        //HTML初期化
        document.getElementById("result").innerHTML = "";

        //HTMLからJavaScriptの変数に
        var addition = parseInt(document.getElementById("addition").value);
        var rate = document.getElementById("rate").value;       ②積み立てる金額と利率を取得

        //変数初期化
        var amount = 0;                                         ③ 合計値を初期化

        //1年ずつ計算
        for (var i = 1; i <= 10; i++){                          ④ 10回、繰り返しの処理を行う
          amount += addition;
          amount += amount * rate / 100;                        ⑤ 合計値に積立金額と合計値の利息を加算

          var tr = document.createElement("tr");

          var th = document.createElement("th");
          th.innerHTML = i;
          tr.appendChild(th);

          var td = document.createElement("td");
          td.innerHTML = amount;
          tr.appendChild(td);

          document.getElementById("result").appendChild(tr);    ⑥ 表に合計値を追記

        }                                                       ⑦ 繰り返し処理の終わり
    }
</script>
```

カスタマイズ① 積立の年数を変更できるようにしよう

カスタマイズの概要

積み立てる期間を10年ではなく、指定できるようにする。

フローチャートで考えよう



HTMLの変更

```
<h2>積立複利の計算</h2>
<div class="interest">
毎年の金額:<input type="number" value="10000" id="addition" placeholder="金額">円<br/>
利息:<input type="number" value="2" id="rate" placeholder="利息(%)">%<br/>
期間:<input type="number" value="3" id="year" placeholder="年">年<br/>
<button onclick="calcInterest()" >計算</button>
```

「期間」を入力するフォームの追加

JavaScriptの変更

```
function calcInterest() {
    //初期化
    document.getElementById("result").innerHTML = "";

    //HTMLからJavaScriptの変数に
    var addition = parseInt(document.getElementById("addition").value);
    var rate = document.getElementById("rate").value;
    var year = parseInt(document.getElementById("year").value);

    //1年ずつ計算
    amount = 0;
    for (var i = 1; i <= year; i++){
      amount += addition;
      amount += amount * rate / 100;

      var tr = document.createElement("tr");

      var th = document.createElement("th");
      th.innerHTML = i;
      tr.appendChild(th);

      var td = document.createElement("td");
      td.innerHTML = amount;
      tr.appendChild(td);

      document.getElementById("result").appendChild(tr);
    }
}
```

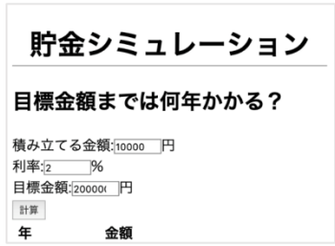
「期間」に入力された値を取得

「期間」に入力された値の回数繰り返す

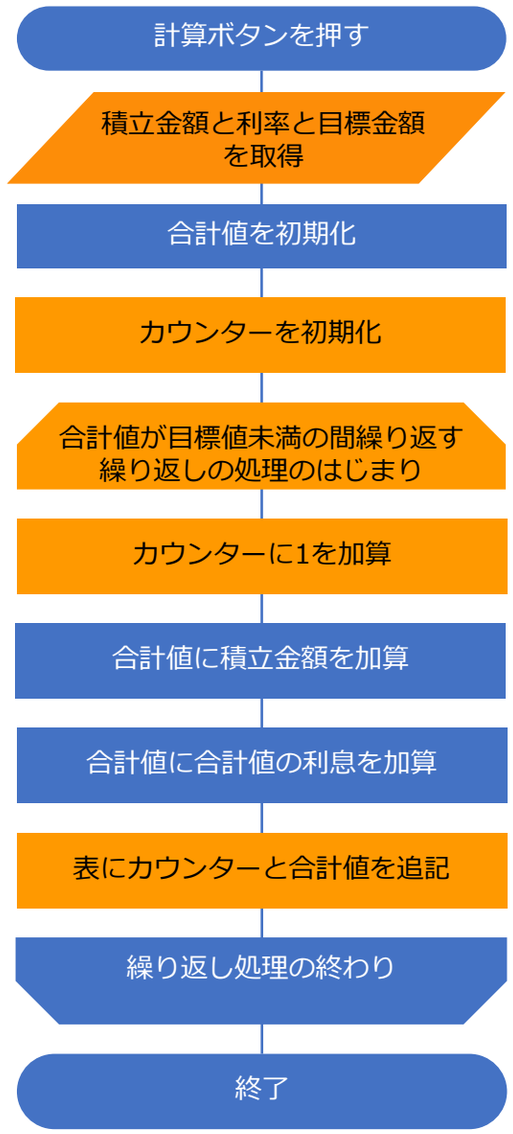
カスタマイズ② 目標金額までの計算にしよう

カスタマイズの概要

while文を使用して、目標金額に達するために必要な年数を求める



フローチャートで考えよう



HTMLの変更

```
<h2>◯円になるのは何年?積立複利の計算</h2>
<div class="interest">
毎年の金額:<input type="number" value="10000" id="addition" placeholder="金額">円<br/>
利息:<input type="number" value="2" id="rate" placeholder="利息(%)">%<br/>
目標金額:<input type="number" value="200000" id="target">円<br/>
<button onclick="calcInterest()" >計算</button>
```

目標金額の入力欄を追加

JavaScriptの変更

```
function calcInterest() {
    //HTML初期化
    document.getElementById("result").innerHTML = "";

    //HTMLからJavaScriptの変数に
    var target = parseInt(document.getElementById("target").value);
    var addition = parseInt(document.getElementById("addition").value);
    var rate = document.getElementById("rate").value;

    // 変数初期化
    var amount = 0;
    var i = 0;

    //1年ずつ計算
    while (amount < target){
      i++;
      amount += addition;
      amount += amount * rate / 100;

      var tr = document.createElement("tr");

      var th = document.createElement("th");
      th.innerHTML = i;
      tr.appendChild(th);

      var td = document.createElement("td");
      td.innerHTML = amount;
      tr.appendChild(td);

      document.getElementById("result").appendChild(tr);
    }
}
```

目標金額を取得

while文に変更
目標金額と合計値を比較

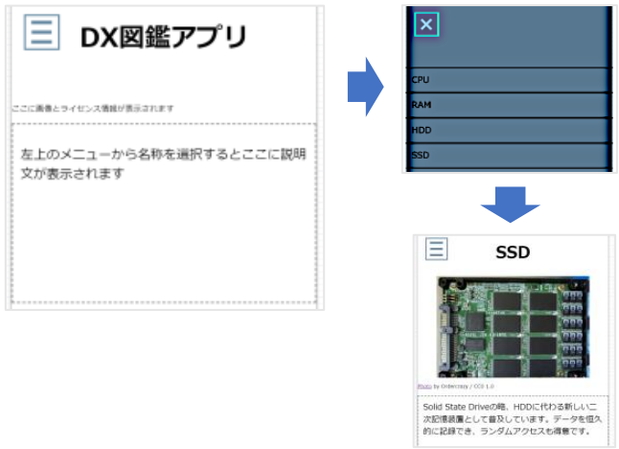
カウンターに1を加算

課題にチャレンジしてみよう

①毎年の利息と積み立てた金額（元金）を分けて表示してみよう。
②身の回りの計算をプログラムで求めてみよう。

アプリプログラミングシート ~DX図鑑~

アプリの概要



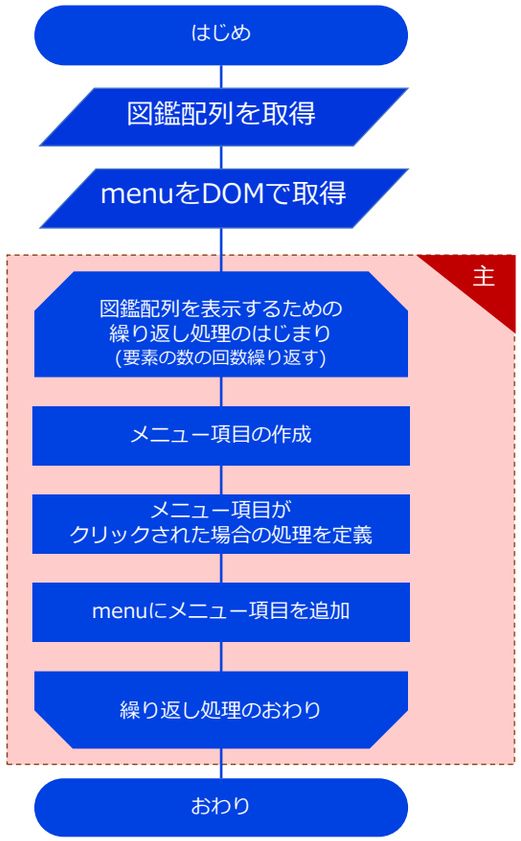
メニューボタン( )を押すと、メニューの項目が表示される。
項目を選ぶと画像や説明文が表示される。

学習内容

【主】繰り返しの処理と配列
【副】 その他関連する内容(書籍対応表)

書籍	関連する内容
第1章 アプリ開発入門	MonacaでDX図鑑プロジェクトを編集する
第2章 HTML入門	画像やボタンはHTMLで用意されている
第3章 CSS入門	画像や文章の表示領域を整えている
第4章 JavaScript入門	図鑑配列を変数に代入している
第6章 関数	全体をonLoad()関数としてまとめている
第7章 イベント	onloadイベントでonLoad()関数を呼び出している
第8章 DOM	メニューがクリックされると 図鑑の画像や文章が書き換わる
第11章 配列	図鑑の内容を配列形式でまとめている
第12章 繰り返し	配列の内容をDOMでli要素として 追加する処理を繰り返しで行っている

DX図鑑のフローチャート(簡易)を確認しよう



メニューの仕組みについて

このアプリのメニューは、インターネット上で公開され、無料で自由に使用できる『Pure Drawer』というツールで作成されている。(※1)
『Pure Drawer』自体はCSSだけで記述されており、HTML側に特定のタグを記述するだけで典型的なデザインのメニューを実装することができる。
CSSの内容はcssフォルダ内の『pure-drawer.css 』ファイルで確認できる。

参考：http://mac81.github.io/pure-drawer/
※1　MITライセンスという制限がほぼ無いライセンス

DX図鑑を動かしてみよう

プログラムを読んでみよう

```
<script src="data.js"></script>                                  ① 図鑑配列を取得
<script>
    function onLoad(){
        var menu = document.getElementById("menu");

        for (var i = 0;i < data.length;i++) {                    ② 繰り返し処理のはじまり
                                                                    (配列の要素数分繰り返す)
            var li = document.createElement("li");
            li.innerHTML =  data[i].title;                       ③ メニュー項目の作成
            li.dataset.index = i;
            li.onclick = function (event) {
                key = event.target.dataset.index;

                document.getElementById('pure-toggle-left').checked = false;
                document.getElementById('title').innerHTML = data[key].title;
                document.getElementById('photo').src = data[key].image;
                document.getElementById('explain').innerHTML = data[key].explain;
                document.getElementById('license').innerHTML = data[key].license;
            }                                                    ④ メニューの項目がクリックされたときの処理
            menu.appendChild(li);                                ⑤ メニュー項目を追加
        }                                                        ⑥ 繰り返し処理のおわり
    }
</script>
```

カスタマイズ① DX図鑑に項目を追加してみよう

カスタマイズの概要

DX図鑑の内容「**SD Card**」を加えてみよう。

変更点

- 配列 data.js にSDカードの情報を追加

画像の探し方

画像を利用する際には著作権を考慮する必要があり、インターネット上で公開されている画像であっても自由に使えるわけではない。
作品に組み込める画像の入手先として、例えば『Wikimedia Commons』などがあり、条件付きで自由に利用可能(クリエイティブ・コモンズ・ライセンス)な画像や著作権が放棄された(パブリックドメイン)画像が公開されている。

引用：
https://commons.wikimedia.org/wiki/File:SD_Cards.JPG

JavaScriptの変更

要素を追記しよう

```
1  var data = [
2      {
3          title: 'CPU',
4          explain: 'Central Processing Unitの略、日本語にすると中央演算処理装置です。演算処理を行います。',
5          image: 'cpu.jpg',
6          license: '<a
           href="https://commons.wikimedia.org/wiki/File:Intel_80486DX2_bottom.jpg">Photo</a> by
           Andrew Dunn / cc-by-sa-2.0',
7      },
8      {
9          title: 'RAM',
                                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
25         license: '<a
           commons.wikimedia.org/wiki/File:Embedded_World_2014_SSD.jpg">Photo</a>
           by Ordercrazy / CC0 1.0',
26     },
27     {
28         title: 'SDカード',
29         explain: '挿抜を容易に行えるカード型の補助記憶装置です。',
30         image: 'sd.jpg',
31         license: '<a href="https://commons.wikimedia.org/wiki/File:SD_Cards.JPG">Photo</a> by
           Rstoplabe14 / public domain',
32     },
33 ];
```

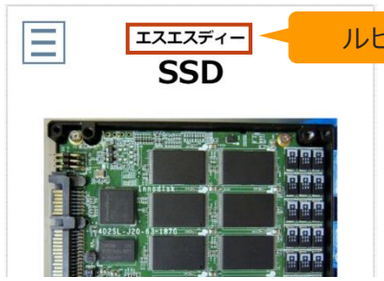
カンマ注意

図鑑情報を入力

カスタマイズ② DX図鑑のタイトルにルビを追加してみよう

カスタマイズの概要

タイトルにルビを加えてみよう



JavaScriptの変更

ruby付きでタイトルを出力するプログラム

```
document.getElementById('title').innerHTML =
    "<ruby>" + data[key].title + "<rt>"
    + data[key].ruby + "</rt>" + "</ruby>";
```

※ 実際には1行で記述して構わない。
※ しっかりプログラムを読んでみよう！

配列の変更

data.jsの各要素にruby情報(ルビの情報)を加える

```
1  var data = [
2      {
3          title: 'CPU',
4          ruby: 'シーピーユー',
5          explain: 'Central Processing Unitの略、日本語にすると中央演算処理装置です。演算処理を行います。
           ',
6          image: 'cpu.jpg',
7          license: '<a
           href="https://commons.wikimedia.org/wiki/File:Intel_80486DX2_bottom.jpg">Photo</a>
           by Andrew Dunn / cc-by-sa-2.0',
8      },
9      {
10         title: 'RAM',
11         ruby: 'ラム',
           explain: 'Random Access Memory の略。データを一時的に記憶するために使われます。主記憶装置とも呼
                                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
29         license: '<a
           href="https://commons.wikimedia.org/wiki/File:Embedded_World_2014_SSD.jpg">Photo</a>
           by Ordercrazy / CC0 1.0',
30     },
31     {
32         title: 'SDカード',
33         ruby: 'エスディーカード',
34         explain: '挿抜を容易に行えるカード型の補助記憶装置です。',
35         image: 'sd.jpg',
36         license: '<a href="https://commons.wikimedia.org/wiki/File:SD_Cards.JPG">Photo</a> by
           Rstoplabe14 / public domain',
37     },
38 ];
```

ruby(ルビ) 追加

課題にチャレンジしてみよう

①オリジナルの図鑑アプリを考えてみよう。
②配列と繰り返しを活用したプログラムの例を調べてみよう。
③著作権について調べてみよう。

アプリプログラミングシート ~地図アプリ(基本編)~

アプリの概要

和歌山城を中心とした地図が表示され、下には自分の現在値に関する情報が表示されるアプリを作成する。和歌山城やその周辺にマーカーが表示される。マーカーの位置や表示内容はプログラムで変更できるようにする。

学習内容

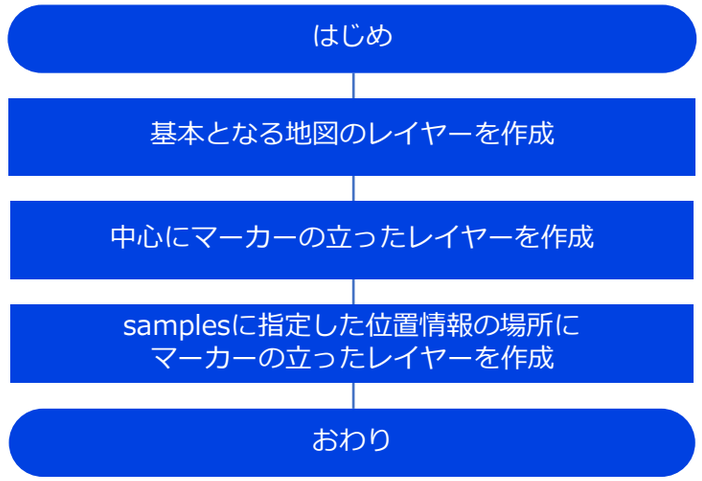
【主】関数の使用とセンサーによる情報の取得
【副】その他関連する内容(書籍対応表)

書籍	関連する内容
第1章 アプリ開発入門	Monacaで地図アプリプロジェクトを編集する
第2章 HTML入門	画像やボタンはHTMLで用意されている
第3章 CSS入門	地図エリアのサイズを設定している
第4章 JavaScript入門	地図やセンサーの値を変数で扱いる
第5章 条件分岐	マーカーがクリックされたときの処理で扱いる
第6章 関数	センサーの値が取れたときの処理を関数化している
第7章 イベント	センサー準備完了イベント(deviceready)で センサー処理を呼び出している
第8章 DOM	センサーの値をフォームに書き出している
第9章 フォーム	センサーの値を表示するためのフォームを用意している
第10章 いろいろな演算子	センサーの値を計算する
第11章 配列	マーカーの緯度経度や名前の情報を配列で扱う
第12章 繰り返し	複数のマーカー情報を地図にマッピングする際に扱う

地図アプリを動かしてみよう

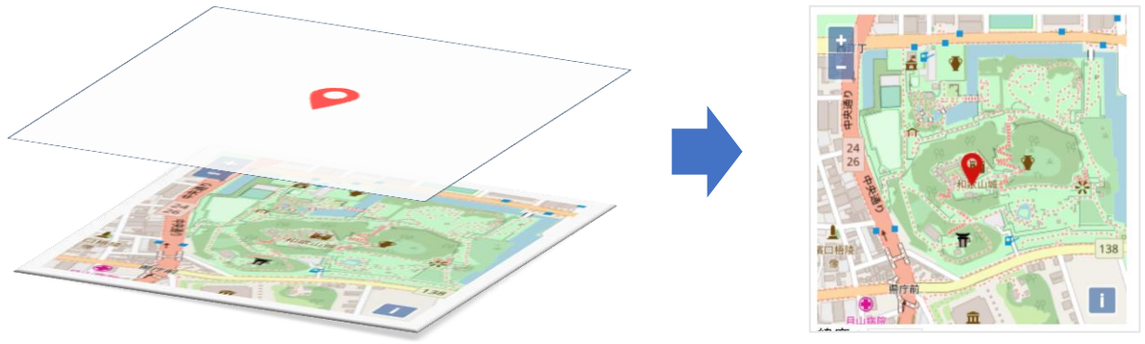
タブレット端末で確認

地図アプリのフローチャート(簡易)を確認しよう



レイヤーとは

レイヤーとは「層」を意味する単語である。基本となる地図レイヤーを用意し、その上にマーカー等を書き込んだ透明なレイヤーを重ねることで自分たちの作りたい地図を画面上で表現することができる。



プログラムを読んでみよう(app.js)

```
// HTMLの読み込みが終わったときに実行する処理
function onLoad() {
    makeMap(osmTile,center);                          ① 基本となる地図を作成

    map.addLayer(makeCenterMarkerLayer(center));      ② 中心にマーカーのあるレイヤーを追加

    map.addLayer(makeGuideMarkerLayer(samples));

}
```

カスタマイズ① 地図の中心と拡大率を変えてみよう

カスタマイズの概要

地図の中心を自分たちの学校にする。

変更点

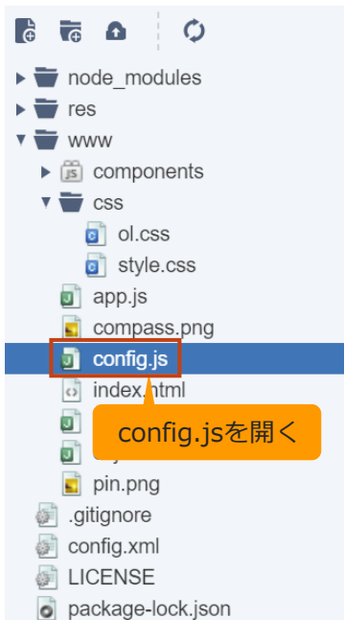
- 自分たちの学校の緯度経度を調べる。
- 地図の中心の緯度経度を変更する。
- 見やすい拡大率に変更する。

緯度経度を調べよう

国土地理院地図 https://maps.gsi.go.jp/ を
使い調べたい地点を表示する。
中心点の緯度/経度が下段に表示される。
(表示されない場合 ∧ をクリック)

JavaScriptの変更

調べた緯度経度と見やすい拡大率にプログラムを変更してみよう。



カスタマイズ② 地磁気センサーで地図を回転させよう

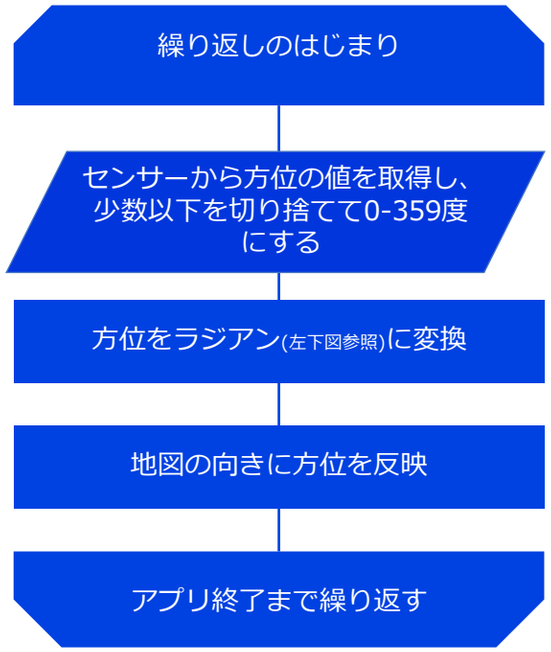
カスタマイズの概要

タブレットの地磁気センサーの値を
地図の方位に反映させる。

変更点

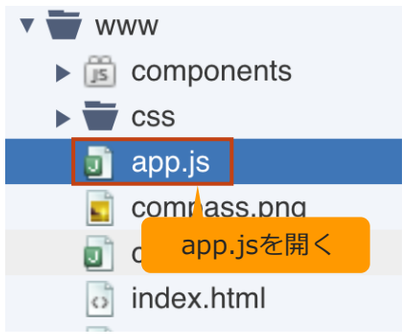
- 地磁気センサーの値を取得する。
- 取得した値を地図オブジェクトが
認識できるよう変換する。

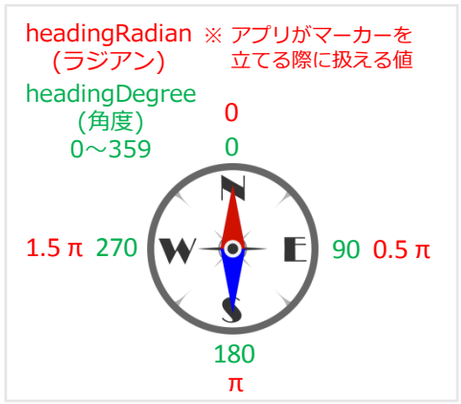
フローチャートで考えよう

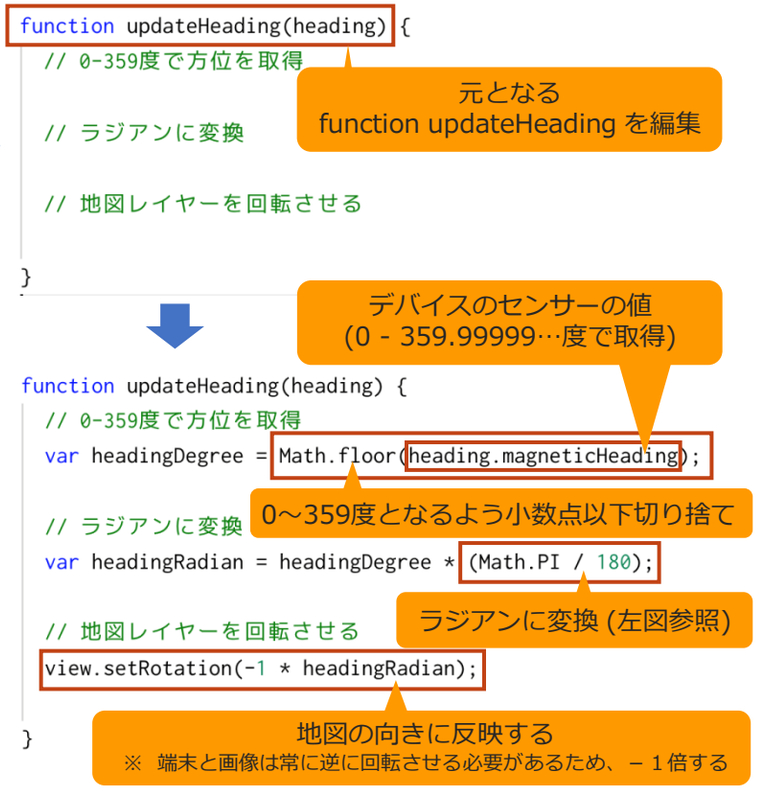


JavaScriptの変更

App.js の updateHeading 関数を動くようにしよう。







アプリプログラミングシート　~地図アプリ(オープンデータ編)~

アプリの概要

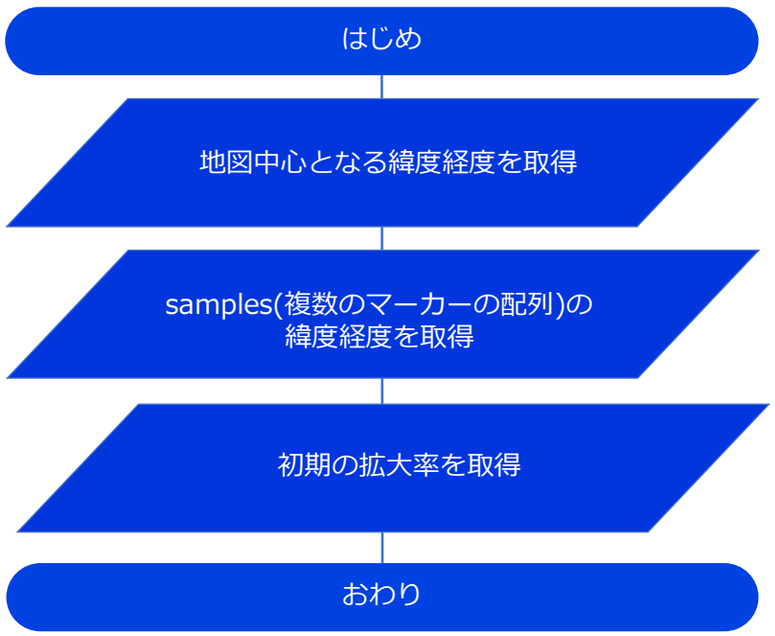
地図アプリ

自分の現在地の情報

緯度：35.6586
経度：139.7086
誤差：65 m
方位：0 度

和歌山城を中心とした地図が表示されるアプリを作成する。和歌山城やその周辺にマーカーが表示される。マーカーの位置や表示内容はプログラムで変更できるようにする。

地図アプリのフローチャート(簡易)を確認しよう



学習内容

【主】配列と関数およびオープンデータの活用
【副】その他関連する内容(書籍対応表)

書籍	関連する内容
第1章 アプリ開発入門	Monacaで地図アプリプロジェクトを編集する
第2章 HTML入門	画像やボタンはHTMLで用意されている
第3章 CSS入門	地図エリアのサイズを設定している
第4章 JavaScript入門	地図やセンサーの値を変数で扱う
第5章 条件分岐	マーカーがクリックされたときの処理で扱う
第6章 関数	センサーの値が取れたときの処理を関数化している
第7章 イベント	センサー準備完了イベント(deviceready)で センサー処理を呼び出している
第8章 DOM	センサーの値をフォームに書き出している
第9章 フォーム	センサーの値を表示するためのフォームを用意している
第10章 いろいろな演算子	センサーの値を計算する
第11章 配列	マーカーの緯度経度や名前の情報を配列で扱う
第12章 繰り返し	複数のマーカー情報を地図にマッピングする際に扱う

プログラムを読んでみよう(config.js)

config.js*

```
var center = [34.2276,135.1714, "和歌山城天守閣"];

var samples = [
    [34.2275,135.1724, "sample1"],
    [34.2275,135.1704, "sample2"],
    [34.2295,135.1714, "sample3"],
    [34.2255,135.1714, "sample4"],
];

// マーカーの画像を設定
var style = new ol.style.Style({
    image: new ol.style.Icon({
        src: 'pin.png',
    })
});
// 拡大率を設定
var zoom = 16;
// コンパスのタイマーIDを用意
var compassWatchId;
var geolocationWatchId;
```

① 中心となる位置を取得

② 複数のマーカーの位置を配列で取得

③ 拡大率を決める

オープンデータについて

オープンデータは、国や地方公共団体が公開する行政情報等のデータです。コンピュータで扱いやすいデータ形式で、二次利用が可能なルールで提供される。近年、経済活性化や官民協働等に活用するために提供の拡大が進められている。

例：和歌山県・福祉のまちづくりマップに掲載された公共施設一覧(csv)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	名称	住所	住所1	住所2	住所3	電話	FAX	利用時間	階数	定休日	駐車場	敷地内の	誘導施設	案内設備	昇降設備	トイレ	公衆電話	その他の	備考	カテゴリー	経度	緯度	分類
2	会津川河	田辺市秋	和歌山県	田辺市	秋津川791－1			終日			◎一般駐車場	有り：3台								その他	135.3749	33.73589	その他
3	かつらぎ	伊都郡か	和歌山県	伊都郡か	丁ノ町2530			9：００－	地上１階							◎障害者（車椅子）用便所		有り：１箇所		その他	135.5172	34.29542	その他
4	ふれあい	伊都郡か	和歌山県	伊都郡か	志賀1341	0736-26-(	0736-26-(	営業時間	地上２階	木曜日	◎一般駐車	◎施設入口	◎誘導ブロック	有り		◎障害者（車椅子）用便所		有り	◎車椅子用	その他	135.5034	34.2218	その他
5	JAかつら	伊都郡か	和歌山県	伊都郡か	大谷字大(	0736-22-(	0736-22-(	8:30～17:(	地上２階	土・日曜日	◎一般駐車	◎施設入口	◎誘導ブロック	有	◎１階（	◎障害者（車椅子）用	◎キャッシ	◎駐車場(		その他	135.4932	34.29232	その他
6	かつらぎ	伊都郡か	和歌山県	伊都郡か	妙寺1471	0736-22-(	0736-22-(	9:00～17:(	地上１階	なし	◎一般駐車	◎施設入口	◎誘導ブロ	◎一般案内	有り	◎障害者	◎難聴者用	◎車椅子の	◎障害者(	その他	135.5193	34.31218	その他
7	喫茶軽食	橋本市高	和歌山県	橋本市	高野口町(	0736-44-(	0736-44-(	7:00～18:(	地上３階	（店舗は１	◎一般駐車	◎施設入口	への車椅子利用：可 ◎自			◎障害者	◎車椅子用	電話　有り：１箇所		その他	135.576	34.31114	その他
8	高野山観	伊都郡高	和歌山県	伊都郡高	高野山600	0736-56-(	0736-56-(	8:30～16:(	地上２階	12/29～1	◎一般駐車	◎施設入口	への車椅子	◎一般案内	有り			◎車椅子の	◎従業員(	その他	135.5867	34.21319	その他
9	中の橋会	伊都郡高	和歌山県	伊都郡高	高野山４	0736-56-3713		9:00～17:(	地上２階	なし		◎施設入口	への車椅子利用：可			椅子用便所なし			◎１、２階	その他	135.6046	34.2163	その他
10	中の橋会	伊都郡高	和歌山県	伊都郡高	高野山４	0736-56-3713		9:00～17:(	地上２階	なし		◎施設入口	への車椅子利用：可			◎障害者（車椅子）用便所	有	◎新中の		その他	135.6046	34.2163	その他

カスタマイズ① 複数のスポットにマーカーを立てよう

カスタマイズの概要

自分が紹介したい地点の緯度経度を調べ、地図に反映させる。

変更点

- 表示されているマーカーの位置を変える。

samples 変数について

1つの地点の情報は下記の配列の形で表される。

地点1の情報

```
[34.2275,135.1724, "sample1"],
```

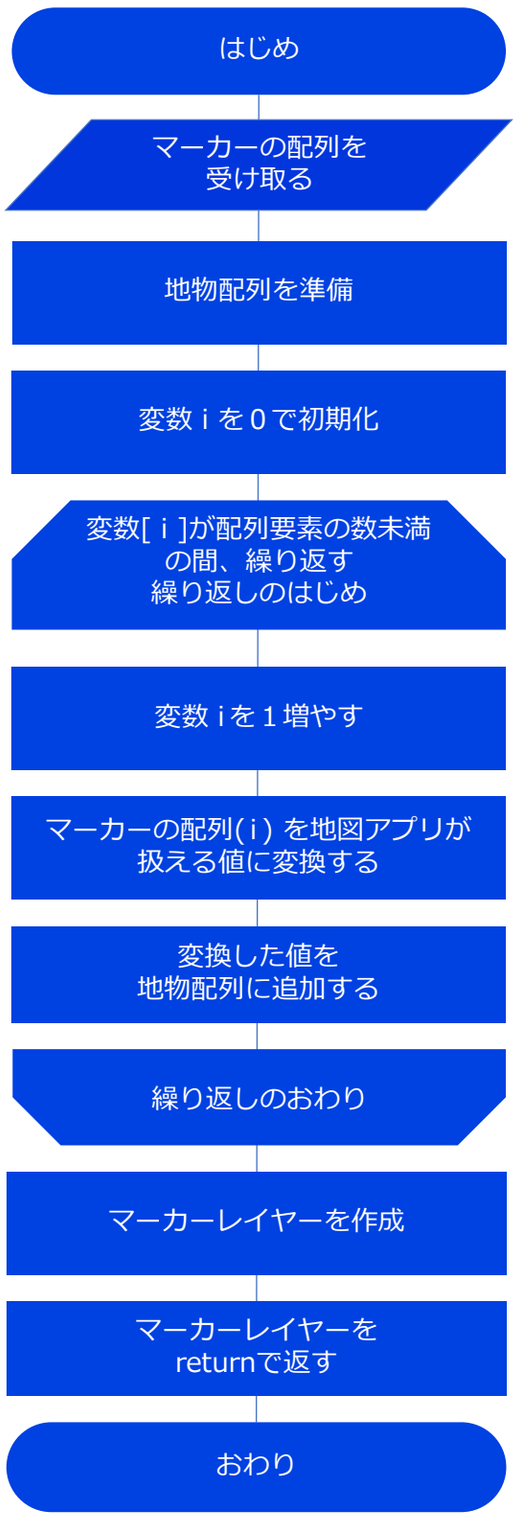
緯度　経度　名称

地点1-4の情報

```
var samples = [
    [34.2275,135.1724, "sample1"],
    [34.2275,135.1704, "sample2"],
    [34.2295,135.1714, "sample3"],
    [34.2255,135.1714, "sample4"],
];
```

地点1　地点2　地点3　地点4

マーカーレイヤー作成のフローチャート（簡易）を確認しよう



プログラムを読んでみよう(map.js)

```
// markerLayerにガイドマーカー描画する関数
function makeGuideMarkerLayer(samples){
    var features = [];          ① 地物配列を定義

    for (var i = 0;i < samples.length;i++) {
        var feature = array2feature(samples[i]);
        features.push(feature);
    }                           ②マーカーの配列[ i ]を地図アプリ上で扱える値に変換し、地物配列に追加する処理の繰り返し

    // markerLayerを作成する
    var markerLayer = new ol.layer.Vector({
        source: new ol.source.Vector({
            features: features
        })
    });

    return markerLayer;         ③ マーカーレイヤーを返す
}
```

JavaScriptの変更

app.js の updateHeading 関数を動くようにしよう

カスタマイズ② オープンデータからマーカーを立てよう

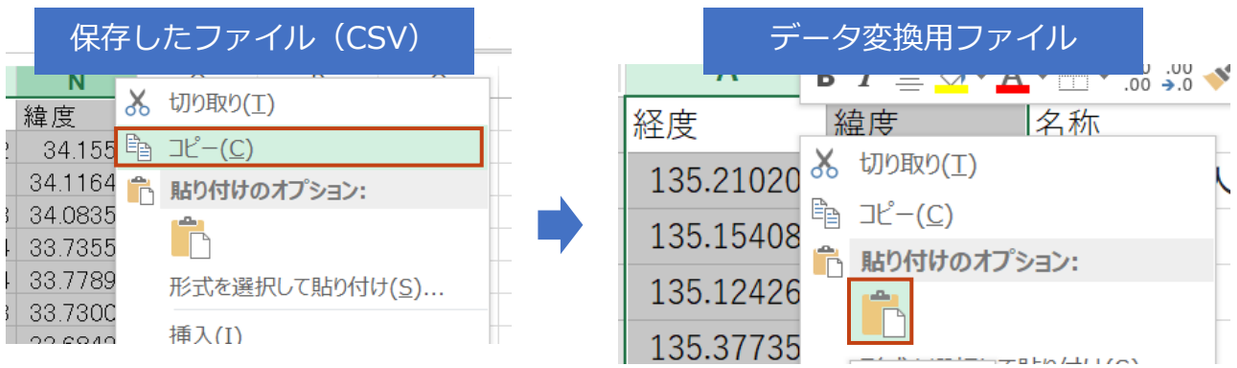
オープンデータの入手・変換

オープンデータの入手

和歌山県オープンデータ　https://github.com/wakayama-pref-org を開く

JavaScript形式への変換

1. 保存したファイルとオープンデータ変換用Excelファイルを Excelで開く。
2. ２つのファイルを並べて表示する。
3. 保存したファイル(CSV)の緯度の列をコピーし、オープンデータ変換用Excelファイルの緯度の列にペーストする。



4. 同様に経度と名称の列についてもコピー&ペーストを行う。
5. 変換用ファイルのD列にJavaScriptの配列形式　[緯度, 経度, “名称”],　が表示される。

A	B	C	D
経度	緯度	名称	JavaScript
135.210202	34.155398	海南市海	[34.15539799892, 135.21020217448, "海南市海南保健福祉センター"],
135.154083	34.116433	海南市下	[34.1164330272992, 135.154083152278, "海南市下津保健福祉センター"],
135.124267	34.083553	有田市文	[34.0835530353026, 135.124266930366, "有田市文化福祉センター"],
135.377357	33.7355586	田辺市総	[33.7355586406906, 135.377357307104, "田辺市総合センター"],

JavaScriptの変更

変換した緯度経度名称を プログラム(config.js)に反映してみよう

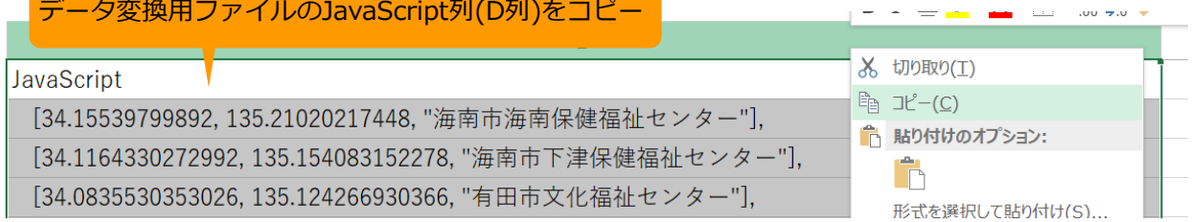

```
var samples = [
    [34.2275,135.1724, "sample1"],
    [34.2275,135.1704, "sample2"],
    [34.2295,135.1714, "sample3"],
    [34.2255,135.1714, "sample4"],
];
```

samplesの配列の中の値をすべて削除

```
var samples = [
];
```

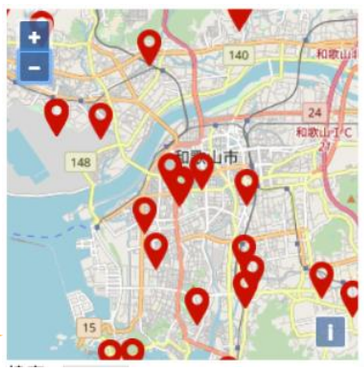
全体を囲う [ ]; は残す

データ変換用ファイルのJavaScript列(D列)をコピー




```
var center = [34.2276,135.1714, "和歌山城天守閣"];
var samples = [
    [34.15539799892, 135.21020217448, "海南市海南保健福祉センター"],
    [34.1164330272992, 135.154083152278, "海南市下津保健福祉センター"],
    [34.0835530353026, 135.124266930366, "有田市文化福祉センター"],
    [33.7355586406906, 135.377357307104, "田辺市総合センター"],
    [33.7789901499148, 135.852424364267, "杉の郷"],
```

samplesの配列に貼り付ける



地図に反映されることを確認

課題にチャレンジしてみよう

①他のオープンデータから地図を作ってみよう (緯度経度情報を含むものを選ぶ)。
②自分だけの（緯度経度情報の）リストを作って地図に反映してみよう。