

## 第3章 Geolocation APIの実装

### Web応用

#### 第13回 GPSを使った位置情報の取得

## 第3章

# Geolocation APIの実装

## 第3章 学習目標

位置情報を取得するための機能の実装について理解できる。

## 1. geolocation

前章のファイル「sample13-2.html」を引き続き使用しましょう。

geolocation APIは、利用者の位置情報を扱うAPIで、JavaScriptで操作できます。  
スマホやPCなどのブラウザがサポートしています。

### geolocation

navigatorオブジェクトにある、geolocationオブジェクトをチェックして、使用できるかどうかを調べます。  
(①の部分)

- if文の中には「navigator.geolocation」しかありませんが、「これが存在すれば条件を満たす」という条件式です。
- 結果をdiv要素(id「msg」)に出力しています。

### ■ サンプル

```
30 <script>
31   //① 使用できるかのチェック
32   if(navigator.geolocation){
33     document.getElementById("msg").innerHTML="使用できます";
34   }else{
35     document.getElementById("msg").innerHTML="使用できません";
36   }
37
38   //② Android用オプション値
```

## 2. .getCurrentPosition()

## 現在地を取得

関数「getMap()」を起動し、現在地の値を取得します。

- 取得するには、geolocationオブジェクトの「**.getCurrentPosition()**」メソッドを使用します。このメソッドは起動すると1回のみ位置情報を取得します。
- このメソッドは、位置情報の取得に成功したら、第1引数、失敗したら第2引数のプログラムを起動させます。ここでは「seiko」、「shippai」としました。メソッド内ではプログラム名に丸括弧「()」はつけない決まりです。

位置情報を取得するための記述は次のとおりです。

### ■ サンプル

```
38 //ⓂAndroid用オプション値
39 var option={enableHighAccuracy:true,timeout:10000,maximumAge:0};
40
41 //Ⓜ取得関数起動
42 function getMap(){
43     navigator.geolocation.getCurrentPosition(seiko,shippai,option);
44 }
```

### 補足

- ②：第3引数「option」は、Android用のオプション値。
  - 「enableHighAccuracy」は、「高精度の位置情報取得するか＝GPSを使用するか」であり、値はtrue。
  - 「timeout」は、「取得タイムアウト時間（ミリ秒）」であり、10000。
  - 「maximumAge」は、「位置情報の有効期限（ミリ秒）」であり、0。
  - タイムアウトエラーが頻繁に発生するようなら、timeoutを長めに設定。
  - Androidの端末については、機種によって個体差があるようで、位置情報を取得しにくい場合があります。その場合は、他のスマートフォン端末やPCで確認してください。

## 3. coords

### 1. 緯度経度などの情報を取得

現在地の情報から、緯度経度高度を取得します。

位置情報の値の取得に成功した場合に起動する関数「seiko」を定義します。(④)

- 位置情報のデータ値はseiko()の第1引数に格納されます。(ここでは引数「position」)
- 緯度経度などはposition内の「coords」に格納され、緯度・経度・高度はそれぞれ、「coords」内の「latitude」「longitude」「altitude」に格納されています。(④-1)

### ■ サンプル

```

46 //④成功したときの関数
47 function seiko(position){
48     //④-1 値を取得
49     var ido = position.coords.latitude;
50     var keido = position.coords.longitude;
51     var kodo = position.coords.altitude;
52
53     //④-2 値を表示
54     document.getElementById("ido").innerHTML ="緯度:"+ido;
55     document.getElementById("keido").innerHTML ="経度:"+keido;
56     document.getElementById("kodo").innerHTML ="高度:"+kodo;
57
58     //④-3 地図を描画
59
60     //④-4 マーカを表示
61
62 }

```

## 2. 失敗したらエラー情報を取得

位置情報の取得に失敗した場合には、関数「shippai」が起動します。

- 失敗したデータは引数「err」に格納され、「err」内のcodeにエラーコード、messageにエラーメッセージが入ります。

### ■ サンプル

```

64 //⑤ 失敗
65 function shippai(err){
66     alert("失敗しました。エラーコードは"+err.code+"。メッセージは"+err.message+"です。");
67 }

```

もう一息。頑張りましょう。

## 練習問題1

### 問題

【クイズ】 択一選択（即解答表示）

Geolocation APIで緯度を取得するコードはどれですか。

- ☐ position.coords.longitude
- ☐ position.coords.altitude
- ☐ position.coords.latitude

## 練習問題1の解説

正解は

`position.coords.latitude;`

です。

経度は「`position.coords.longitude`」、高度は「`position.coords.altitude`」ですね。

## 第3章 まとめ

位置情報を取得するための機能の実装について理解できた。

## 第3章 終わり

Web応用

第13回 GPSを使った位置情報の取得

# 第3章

# Geolocation APIの実装

# 終わり