

★Geolocation API

Geolocation API



■Geolocation APIとは

Geolocation APIは、ユーザーの位置情報を扱うためのAPIです。Geolocation APIの登場以前にも携帯端末などで位置情報を扱うことは出来ましたが、各携帯キャリア独自の規格であったためキャリアごとの仕様に合わせて開発する必要がありました。

Geolocation APIは、ウェブの標準化団体であるW3Cが仕様策定を進める規格であり、JavaScriptで位置情報を取得できるように標準化されています。すでに、Firefox・Google Chrome・Safariなどの一般的なブラウザでサポートされており、スマートフォンのようなGPS対応の携帯端末向けのウェブサイトだけではなく、一般的なブラウザで閲覧するいわゆるPCサイトでもユーザーの位置情報を利用したコンテンツを提供することが可能になっています。

とにかくサンプルをしてみる

■どんなしくみ？

Geolocation APIでは、無線LAN・WiFi・携帯電話基地局・GPS・IPアドレスなどから位置情報を取得します。そのため、ユーザーのインターネット接続環境次第で、取得できる位置情報の精度や取得に掛かる時間が異なります。

例えば、GPS対応の携帯端末から接続しているユーザーの位置情報は比較的正確に取得できるでしょうが、一般的なインターネットサービスプロバイダから接続しているユーザーの位置情報は大まかにしか取得できないかもしれません。

■セキュリティとプライバシー

Geolocation APIは、ユーザーのセキュリティとプライバシーに配慮した仕様となっています。位置情報をリクエストすれば必ず取得できるわけではなく、ユーザーの許可が必要となります。ユーザーはブラウザの確認ウィンドウで、位置情報の利用を許可するかどうかを選択できるようになっています。

iPhoneの場合



Firefoxの場合



※メモ：上の画面で[常に許可しない]を選択した場合、それを解除するには、ウェブページ上で右クリック → [ページの情報を表示] → [サイト別設定] → [位置情報の送信] の [毎回確認する] にチェックします。（Firefoxの場合）

[スクリーンショットを確認](#)

■ユーザー環境でGeolocation APIが利用できるかどうかチェック

ユーザー環境でGeolocation APIが利用できるかどうかは、navigator.geolocationを参照することでチェックすることができます。Geolocation APIを利用できる環境と利用できない環境で処理を振り分けても良いでしょう。

JavaScriptソース

```
if (navigator.geolocation) {
  //Geolocation APIを利用できる環境向けの処理
} else {
  //Geolocation APIを利用できない環境向けの処理
}
```

■位置情報の取得に関する3つのメソッド

Geolocation APIでユーザーの位置情報の取得をする場合、現在のところ使用できるのは以下の3つのメソッドです。 [このページで紹介するサンプル](#)では、 `getCurrentPosition()`メソッドでユーザーの現在の位置情報を一回だけ取得していますが、 制作するアプリケーションの内容に応じて必要なメソッド・属性を使用します。

`getCurrentPosition()` …… ユーザーの現在の位置情報を一回だけ取得する
`watchPosition()` …… ユーザーの位置情報を定期的に監視する
`clearWatch()` …… `watchPosition()`による位置情報の監視をクリアする

[Geolocation APIのメソッド・属性の一覧](#)

`getCurrentPosition()`メソッドには、3つの引数を指定することができます。最初の引数は位置情報の取得に成功した場合のコールバック関数、2番目の引数は位置情報の取得ができなかった場合のコールバック関数、3番目の引数は位置情報の取得の詳細を指定する追加オプションです。

最初の引数は必須ですが、2番目と3番目の引数は省略することができます。

JavaScriptソース

```
//ユーザーの現在の位置情報を取得
navigator.geolocation.getCurrentPosition(
  successCallback, errorCallback
);
```

■ 位置情報の取得の詳細を指定する追加オプション

上のJavaScriptソースでは3番目の引数を省略していますが、指定すればより詳細に位置情報を制御することができます。指定可能な追加オプションは、現在のところ以下の3つです。

`enableHighAccuracy` …… より精度の高い位置情報を取得する (true, false)
`timeout` …… タイムアウトまでの時間 (単位: ミリ秒)
`maximumAge` …… 位置情報の有効期限 (単位: ミリ秒)

[Geolocation APIのメソッド・属性の一覧](#)

JavaScriptソース

```
//ユーザーの現在の位置情報を取得 (位置情報の精度を高くする・6秒でタイムアウト・有効期限10分)
navigator.geolocation.getCurrentPosition(
  successCallback, errorCallback, {enableHighAccuracy:true; timeout:6000; maximumAge:600000;}
);
```

■ 位置情報の取得に成功した場合の処理

`getCurrentPosition()`メソッドで位置情報を取得すると、コールバック関数にPositionというオブジェクトが渡されます。位置情報の取得に成功した場合のコールバック関数内では、Positionオブジェクトの`coords`属性から緯度・経度・高度などの値を取得します。

尚、現在のところ、Positionオブジェクトの`coords`属性から取得できるのは以下の位置情報ですが、Geolocation APIの将来のバージョンでは、`coords`属性だけではなく、例えば、通りの番号などを取得できるような新しい属性が追加されるかもしれません。

属性名	値の内容	値の単位
latitude	緯度 (-180～180)	度
longitude	経度 (-90～90)	度
altitude	高度	m
accuracy	緯度・経度の誤差	m
altitudeAccuracy	高度の誤差	m
heading	方角 (0～360)	度
speed	速度	m/秒

以下のサンプルでは、Positionオブジェクトのcoords属性から取得した緯度・経度・高度などの値を、ウェブページ上に表示するテキスト（HTMLソース）に置き換えています。

JavaScriptソース

```

/***** ユーザーの現在の位置情報を取得 *****/
function successCallback(position) {
  var gl_text = "緯度：" + position.coords.latitude + "<br>";
  gl_text += "経度：" + position.coords.longitude + "<br>";
  gl_text += "高度：" + position.coords.altitude + "<br>";
  gl_text += "緯度・経度の誤差：" + position.coords.accuracy + "<br>";
  gl_text += "高度の誤差：" + position.coords.altitudeAccuracy + "<br>";
  gl_text += "方角：" + position.coords.heading + "<br>";
  gl_text += "速度：" + position.coords.speed + "<br>";
  document.getElementById("show_result").innerHTML = gl_text;
}

```

■ 位置情報の取得ができなかった場合の処理

Geolocation APIでユーザーの位置情報をリクエストしても、ユーザー側の許可設定・接続環境・電波状況などによって位置情報が取得できない場合があります。このような場合に備えて、あらかじめエラー処理をしておくほうが良いでしょう。

位置情報の取得時のエラーコードは、code属性で取得することができます。code属性では1, 2, 3のいずれかの数値が返ります。それぞれの意味は以下の通りです。

エラーコード値	キーワード値	エラー内容
1	PERMISSION_DENIED	位置情報の利用が許可されていない（ユーザーが位置情報の利用を許可しなかった場合など）
2	POSITION_UNAVAILABLE	デバイスの位置が判定できない（プロバイダが内部エラーとなっている場合など）
3	TIMEOUT	タイムアウト（timeout属性で指定された時間内にPositionオブジェクトを取得できなかった場合など）

エラー内容の詳細は、message属性で取得することができます。もし、不明なエラーが発生した場合には、message属性でエラーメッセージを取得することでその詳細が分かるかもしれません。ただし、message属性は主にデバッグ用であり、アプリケーションのユーザーインターフェイスには直接使用すべきではないとされています。

JavaScriptソース

```

/***** 位置情報が取得できない場合 *****/
function errorCallback(error) {
    var err_msg = "";
    switch(error.code)
    {
        case 1:
            err_msg = "位置情報の利用が許可されていません";
            break;
        case 2:
            err_msg = "デバイスの位置が判定できません";
            break;
        case 3:
            err_msg = "タイムアウトしました";
            break;
    }
    document.getElementById("show_result").innerHTML = err_msg;
    //デバッグ用→ document.getElementById("show_result").innerHTML = error.message;
}

```

■Geolocationのサンプル ソース全文

HTML + JavaScriptソース

```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset=utf-8>
<script>
//ユーザーの現在の位置情報を取得
navigator.geolocation.getCurrentPosition(successCallback, errorCallback);

/***** ユーザーの現在の位置情報を取得 *****/
function successCallback(position) {
    var gl_text = "緯度：" + position.coords.latitude + "<br>";
    gl_text += "経度：" + position.coords.longitude + "<br>";
    gl_text += "高度：" + position.coords.altitude + "<br>";
    gl_text += "緯度・経度の誤差：" + position.coords.accuracy + "<br>";
    gl_text += "高度の誤差：" + position.coords.altitudeAccuracy + "<br>";
    gl_text += "方角：" + position.coords.heading + "<br>";
    gl_text += "速度：" + position.coords.speed + "<br>";
    document.getElementById("show_result").innerHTML = gl_text;
}

/***** 位置情報が取得できない場合 *****/
function errorCallback(error) {
    var err_msg = "";
    switch(error.code)
    {
        case 1:
            err_msg = "位置情報の利用が許可されていません";
            break;
        case 2:
            err_msg = "デバイスの位置が判定できません";
            break;
        case 3:
            err_msg = "タイムアウトしました";
            break;
    }
    document.getElementById("show_result").innerHTML = err_msg;
    //デバッグ用→ document.getElementById("show_result").innerHTML = error.message;
}
</script>
<title>Geolocation API サンプル</title>
</head>
<body>
<p>あなたの現在位置</p>
<div id="show_result"></div>
</body>
</html>

```

[サンプルを別画面で開く](#)**■ Geolocation APIのメソッド・属性**

Geolocation APIには、以下のメソッド・属性が用意されています。

位置情報の取得	位置情報取得の詳細オプション
位置情報の取得項目	エラー内容

位置情報の取得 Geolocation interface

getCurrentPosition() …… ユーザーの現在の位置情報を一回だけ取得する

watchPosition() …… ユーザーの位置情報を定期的に監視する

clearWatch() …… watchPosition()による位置情報の監視をクリアする

http://www.w3.org/TR/geolocation-API/#geolocation_interface

[▲メソッド・属性の先頭へ](#)

位置情報取得の詳細オプション PositionOptions interface

enableHighAccuracy …… より精度の高い位置情報を取得する (true, false)

timeout …… タイムアウトまでの時間 (単位: ミリ秒)

maximumAge …… 位置情報の有効期限 (単位: ミリ秒)

http://www.w3.org/TR/geolocation-API/#position_options_interface

[▲メソッド・属性の先頭へ](#)

位置情報の取得項目 Coordinates interface

latitude …… 緯度 (-180~180、単位: 度)

longitude …… 経度 (-90~90、単位: 度)

altitude …… 高度 (単位: m)

accuracy …… 緯度・経度の誤差 (単位: m)

altitudeAccuracy …… 高度の誤差 (単位: m)

heading …… 方角 (0~360、単位: 度)

speed …… 速度 (単位: m/秒)

http://www.w3.org/TR/geolocation-API/#coordinates_interface

[▲メソッド・属性の先頭へ](#)

エラー内容 PositionError interface

code …… エラーコードを返す (1 = Geolocation APIの利用が許可されない, 2 = デバイスの位置が判定できない, 3 = タイムアウト)

message …… エラー内容の詳細を返す

http://www.w3.org/TR/geolocation-API/#position_error_interface