

# 「Googleマップを現在地からの経路検索で開くリンク」ってなんだ

---

と思われることもありそうなので少し具体的に説明をば。

ウェブサイトを作っているとよくGoogleマップの埋め込みとその傍らにGoogleマップを直接開くリンクテキストなどを置いておくことがあります。

このとき特定の住所を開くだけではなく、現在地からその住所への経路検索をした状態で開きたい！という要望がありました。

経路検索をした状態というのは↓こんな感じです。

宮城から東京への経路

この「宮城から」のところを「現在地から」で開きたい。というわけです。

調べたところGoogleマップの公式ではそういったサービスを提供していないようでしたので、Geolocation APIを使って経路検索した状態のURLを作ることにしました。

## ということで、Geolocation APIの使い方を見てみる

---

みんなだいすき（かはわからないけど）MDNさんの[Geolocation の利用](#)によると、Geolocation APIは `navigator.geolocation` を通じて提供され、ユーザーの現在位置を取得するには `getCurrentPosition()` メソッドを呼び出すそうです。

※Chromeはバージョン50からhttpでの現在地の取得を禁止しているので、大多数のブラウザに対応するにはhttps化する必要があります。

```
navigator.geolocation.getCurrentPosition(success, error, options)
```

`getCurrentPosition()` の第 1 引数は、現在位置の取得に成功したときのコールバック関数を、第 2 引数は失敗したときのコールバック関数を実行します。

つまり、現在位置の取得に成功した結果を受け取りたい場合には、第 1 引数に関数を指定してやればその関数の引数として結果が受け取れます。

```
navigator.geolocation.getCurrentPosition(function(position) {  
    console.log(position);  
});
```

位置情報は `position.coords.latitude` に緯度、`position.coords.longitude` に経度が格納されています。

実際に使う際には、即時関数などの中でブラウザがGeolocation APIに対応しているかどうかの判定を加えて使います。

```
(function(){  
    if( 'geolocation' in navigator == false) return;  
  
    navigator.geolocation.getCurrentPosition(function(position) {  
        console.log(position);  
    });  
})();
```

エラー時やオプションを設定する場合にはこのようなコードになります。

```
(function(){  
    if( 'geolocation' in navigator == false) return;  
  
    navigator.geolocation.getCurrentPosition(function(position) {  
        console.log(position);  
  
    }, function(error) {
```

```
// ブラウザがGeolocation APIに対応しているが、使えない場合とかエラーが起きたときの処理
```

```
}, {  
  timeout: 10000, // 取得タイムアウトまでの時間（ミリ秒）  
  maximumAge: 10000, // 位置情報の有効期限（ミリ秒）  
  enableHighAccuracy: true // より精度の高い位置情報を取得するか（true/false）  
});  
}());
```

## ちなみに

---

`getCurrentPosition()` の他にも現在位置を取得するメソッドとして `watchPosition()` がありますが、こちらは現在位置が更新されるたびにコールバック関数を実行してくれるものなので、よりリアルタイムな位置情報が必要なときにはこちらを使います。

引数の設定が `getCurrentPosition()` と同じなのは覚えやすくていいですね。

## 現在位置の取得方法がわかったので、経路検索のURLを作って設定する

---

現在位置が取得できたら後は簡単です。

jQueryとか使ってa要素の `href` を経路検索のURLに書き換えます。

経路検索のURLは「<https://www.google.co.jp/maps/dir/{緯度},{経度}/{目的地}/>」となっているため、`{緯度}{経度}`をGeolocation APIで取得した情報を指定し、`{目的地}`のところは予め設定した住所をURLエンコードして指定します。

```
<a href="https://goo.gl/maps/gsdryZYLwvD2" target="_blank" class="btn-googlemap">大きな地図で見る</a>
```

```
$(function() {  
  // Geolocation APIに対応していなかったら処理を止める  
  if ( 'geolocation' in navigator == false ) return;
```

```
var address = '北海道稚内市';
var btn = $('btn-googlemap');

// リンクを書き換える
var setLink = function(lat, lng) {
    // Google mapの経路検索のURLを作る
    var link = 'https://www.google.co.jp/maps/dir/' + lat + ',' + lng + '/' + encodeURIComponent

    btn.attr('href', link);
};

navigator.geolocation.getCurrentPosition(function(position) {
    var coords = position.coords;
    setLink(coords.latitude, coords.longitude);
});
});
```



今回は href に予めGoogle mapで住所のリンクを設定しておき、Geolocation APIが使えない場合にはその住所が開くようにしました。

しかし、使える場合でもクリックされたタイミングで現在位置を取得していないこともありえるため、取得できていなかったら簡単なローディングアニメーションを表示しておき、取得できたタイミングで location.href や window.open を使って開くようにするとユーザーには親切かなと思います。

## できたもの

---

(このブログがhttps化してないので) [CODEPENでサンプル](#)用意してみました。スマホで見た方が位置情報が正確なので楽しいです。