

このドキュメントについて

<http://www.google.com/apis/maps/documentation/>

を日本語訳したものです。

株式会社 sus4 の野村亮之が翻訳を行いました。

翻訳された内容に不備がある場合は、aki-nomura@sus4.co.jp までご連絡ください。

GoogleMaps API 仕様書

グーグルマップ API を利用して、あなたの運営しているウェブページにグーグルマップを表示させることが可能になります。API を利用するには、最初に [sign up for an API key](#) で API キーを取得してください。そして、下記の説明にしたがってコードを記述します。

API 自体がリリースされたばかりであるので、バグが残っている可能性があり、また、ドキュメントの不足があるかもしれません。これらのバグやドキュメンテーションの不足を解決するために、[join the Maps API discussion group](#) に参加して、API についてフィードバックをしたり、議論をしたりしてください。

グーグルマップの “ Hello World ”

API を使い始めるのにもっとも簡単な方法は、下のサンプル例を見ることです。下のサンプルページは、300x300 の Palo Alto の中心街を表示するスクリプトです。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Google Maps JavaScript API Example - simple</title>
    <script src="http://maps.google.com/maps?file=api&v=1&key=abcdefg"
type="text/javascript"></script>
  </head>
  <body>
    <div id="map" style="width: 300px; height: 300px"></div>
    <script type="text/javascript">
      //

      if (GBrowserIsCompatible()) {
        var map = new GMap(document.getElementById("map"));
        map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);
      }

      //]]&gt;</pre></div>
```

```
</script>
</body>

</html>
```

[download this example](#) からサンプルコードをダウンロードし、編集して遊んでみるすることができます。自分のウェブサイトでこのスクリプトを実行する場合には、“ abcdefg ” の部分を [Maps API key](#) で取得した API キーに置き換える必要があります。

上記のスクリプトのサンプルから分かるように、グーグルマップを利用する場合には、一行の JavaScriptURL(<http://maps.google.com/maps?file=api&v=1>)をページ内に記述して、グーグルマップを自分のウェブサイトで行うのに必要なシンボル（関数やクラス群）を読み込んでいます。グーグルマップ API を利用するには、取得した API キーを変数として記述した URL を script 要素に記述する必要があります。以下は、その例です：

```
<script src="http://maps.google.com/maps?file=api&v=1&key=abcdefg"
type="text/javascript"></script>
```

グーグルマップ API から読み込んだメインのクラスは GMap で、GMap はページに単一の地図を表示するクラスです。GMap クラスを使用して、1 ページ内に複数の地図を表示することもできます。地図を表示する場合は div 要素の中に埋め込むことが通例です。

地図を操作するメソッドや地図上にアイコンや文字列をプロットするためのメソッドの詳細は、順に説明していきます。

ブラウザ互換性

グーグルマップはすべてのブラウザで動作するわけではありません。現段階で、最新のバージョンの Firefox/Mozilla, IE 5.5+, Safari 1.2+ と sort of supports Opera で動作の保証がされています。IE 5.0 は動作の保証はされていません。もしも互換性のないブラウザでアクセスしてきたユーザーに対して、グーグルマップを表示せず違う動作（警告を出力したり、他のスクリプトを実行させたい場合）には、グーグルマップ API は、グローバルメソッドである ([GBrowserIsCompatible\(\)](#))を提供していますので、それを利用して互換性があるブラウザかどうかを判別することもできます。しかし、このメソッドはすべてのブラウザに対して正しく動作するとは限りません。

グーグルマップ実行時に読み込むスクリプト <http://maps.google.com/maps?file=api&v=1> は、ほとんどのブラウザでエラーを出力せずに実行することができますので、互換性をチェックしないで読み込みを行っても問題はありません。

[GBrowserIsCompatible\(\)](#) についてのサンプル例は上記のスクリプトサンプルを確認してください。そして、古いブラウザでエラーを出力しないこともあわせて確認して導入してください。通常のアプリケーション上の動作では問題ないですが、スクリプトを記述する前にサンプルで確認をすることをお奨めします。

XHTML と VML

地図を埋め込むページは、standards-compliant（標準互換性のある）XHTML で記述することをお奨めします。ページ内の 1 行目に記述されている DOCTYPE をブラウザが認識できれば、そのページの描画は“標準互換性モード”として実行されます。標準互換性モードで実行される場合は、ブラウザをまたがって、厳密な動作や描画を行うことができます（どのブラウザでも同じ表現・動作をさせることができることを意味します）。

同様に、地図上に線を読み込みたい（描画したい）場合は、IE ブラウザ上で XHTML としてページを実行し、VML 名前空間を宣言します。以下はサンプル例です。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
  <head>
    <style type="text/css">
      v\*: * {
        behavior:url(#default#VML);
      }
    </style>
    <script src="http://maps.google.com/maps?file=api&v=1&key=abcdefg"
type="text/javascript"></script>
  </head>
```

詳しくは、[Microsoft's VML workshop](#) を確認してください。

API Updates

<http://maps.google.com/maps?file=api&v=1> は、第 1 版（Version 1）の API を参照します。将来、API に重要な変更が加えられた場合には、API のバージョンをあげる予定です。バージョンアップがされた場合には、[Google Code](#) や [Maps API discussion group](#) で告知をする予定です。

古いバージョンの API と新しいバージョンの API は、数ヶ月間は並行で運用されますが、古いバージョンは使用停止になっていきます。

グーグルマップチームは、API に対しての細かいバグフィックスや機能強化を裏で行います。これらのバグフィックスは性能の強化に対してのみ実行されるはずですが、バグフィックス後の予期せぬ事態により API 等が正しく動作しない可能性もあります。その場合は、速やかに [Maps API discussion group](#) へ報告をしてください。

ジオコーディング¹、ルーティング、その他

グーグルマップには、現在のところジオコーディングやルーティングサービスは実装されていません。しかしながら、インターネット上には、ジオコーディングを行うためのさまざまな [free geocoders](#) が公開されています。

1 住所の文字情報を解析して、自動的に緯度経度を付け、地図上にポインティングする事をジオコーディングと呼びます。データを地図上にプロットすることで、分布状況を視覚的にとらえる事が出来るほか、その他の統計データを組み合わせてその地域エリアの潜在性を分析することが可能です。

<http://www.mapinfo.jp/products/data/geocoding/>より引用

サンプル例

基本

Palo Alto の中心街についてのマップを作成します。

```
var map = new GMap(document.getElementById("map"));
```

```
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);
```

[View example \(simple.html\)](#) (サンプル例を参照)

地図の動作とアニメーション

メソッド `recenterOrPanToLatLng` は、表示領域の継続的な平行移動を行わせるためのメソッドです。lat/lng (座標) が現在の地図の表示領域の中にある場合は、表示領域の継続的な平行移動をします。そうでない場合は、移動は行われません。

[View example \(animate.html\)](#) (サンプル例を参照)

コントローラーを地図に追加する

メソッド `addControl` を使って地図上にコントローラーを追加することができます。以下のサンプル例では、ビルドイン関数である `GSmallMapControl` と `GMapTypeControl` を追加して、地図の平行移動・ズームイン・ズームアウト、そして地図のサテライトモードと地図モードの切り替えを行うことができますようにしています。

```
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);
```

[View example \(controls.html\)](#) (サンプル例を参照)

イベントリスナー

`GEvent.addListener` によって、イベントリスナーを登録することができます。下のサンプル例では、ユーザーがドラッグして地図を動かした後に地図の中心座標を取得して、div 要素に出力しています。

```
var map = new GMap(document.getElementById("map"));
GEvent.addListener(map, "moveend", function() {
    var center = map.getCenterLatLng();
    var latLngStr = '(' + center.y + ', ' + center.x + ')';
    document.getElementById("message").innerHTML = latLngStr;
});
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);
```

[View example \(event.html\)](#) (サンプル参照)

情報ウィンドウを開く

以下のサンプル例では、“Hello world” を地図の中心に、情報ウィンドウを使って表示しています。情報ウィンドウは、通常ではマーカーの上に展開されますが、地図上のどこにでも表示することができます。

```
var map = new GMap(document.getElementById("map"));
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);
map.openInfoWindow(map.getCenterLatLng(),
    document.createTextNode("Hello world"));
```

[View example \(infowindow.html\)](#) (サンプル参照)

マップオーバーレイ（地図上に情報を重ねる）

マップオーバーレイ機能を使用して、マーカーと線をランダムにそれぞれ 10 回描画しています。

```
// Center the map on Palo Alto
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);

// Add 10 random markers in the map viewport using the default icon
var bounds = map.getBoundsLatLng();
var width = bounds.maxX - bounds.minX;
var height = bounds.maxY - bounds.minY;
for (var i = 0; i < 10; i++) {
    var point = new GPoint(bounds.minX + width * Math.random(),
                           bounds.minY + height * Math.random());
    var marker = new GMarker(point);
    map.addOverlay(marker);
}

// Add a polyline with 4 random points. Sort the points by longitude so that
// the line does not intersect itself.
var points = [];
for (var i = 0; i < 5; i++) {
    points.push(new GPoint(bounds.minX + width * Math.random(),
                           bounds.minY + height * Math.random()));
}
points.sort(function(p1, p2) { return p1.x - p2.x; });
map.addOverlay(new GPolyline(points));
```

[View example \(overlay.html\)](#)（サンプル参照）

クリック操作

次のサンプル例では、地図をクリックしたときにその座標に新しいマーカーを作成する方法を紹介しています。マーカーをクリックすると、地図上からマーカーが削除されます。

```
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);

GEvent.addListener(map, 'click', function(overlay, point) {
    if (overlay) {
```

```
    map.removeOverlay(overlay);
  } else if (point) {
    map.addOverlay(new GMarker(point));
  }
});
```

[View example \(click.html\)](#) (サンプル参照)

情報ウィンドウをマーカー上に配置

次のサンプル例は、マーカー上をクリックした時のイベントリスナーを用意し、マーカーをクリックしたときに任意の情報ウィンドウを表示する方法です。関数の機能を利用して、任意の文字列をそれぞれのマーカーに対して表示させることを実現しています。

```
// Center the map on Palo Alto
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);

// Creates a marker whose info window displays the given number
function createMarker(point, number) {
  var marker = new GMarker(point);

  // Show this marker's index in the info window when it is clicked
  var html = "Marker #<b>" + number + "</b>";
  GEvent.addListener(marker, "click", function() {
    marker.openInfoWindowHtml(html);
  });

  return marker;
}

// Add 10 random markers in the map viewport
var bounds = map.getBoundsLatLng();
var width = bounds.maxX - bounds.minX;
var height = bounds.maxY - bounds.minY;
for (var i = 0; i < 10; i++) {
  var point = new GPoint(bounds.minX + width * Math.random(),
    bounds.minY + height * Math.random());
  var marker = createMarker(point, i + 1);
  map.addOverlay(marker);
}
```

[View example \(markerinfowindow.html\)](#) (サンプル参照)

アイコンを新規作成

以下のサンプル例では、[Google Ride Finder](#) で使用されているミニマーカーを地図上に表示しています。作成したアイコンを使用するには、表面に使用する画像、影面に使用する画像、そして各アイコンの中心座標 (ポイント) と情報ウィンドウを表示する座標 (ポイント) をそれぞれ定義します。

```
// Create our "tiny" marker icon
var icon = new GIcon();
icon.image = "http://labs.google.com/ridefinder/images/mm_20_red.png";
icon.shadow = "http://labs.google.com/ridefinder/images/mm_20_shadow.png";
icon.iconSize = new GSize(12, 20);
icon.shadowSize = new GSize(22, 20);
icon.iconAnchor = new GPoint(6, 20);
icon.infoWindowAnchor = new GPoint(5, 1);

// Center the map on Palo Alto
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);

// Creates one of our tiny markers at the given point
function createMarker(point) {
  var marker = new GMarker(point, icon);
  map.addOverlay(marker);
  GEvent.addListener(marker, "click", function() {
    marker.openInfoWindowHtml("You clicked me!");
  });
}

// Place the icons randomly in the map viewport
var bounds = map.getBoundsLatLng();
var width = bounds.maxX - bounds.minX;
var height = bounds.maxY - bounds.minY;
for (var i = 0; i < 10; i++) {
  createMarker(new GPoint(bounds.minX + width * Math.random(),
    bounds.minY + height * Math.random()));
}
```

[View example \(icon.html\)](#) (サンプル参照)

アイコンクラスを利用する

多くの場合、地図上に表示する表面の画像は違いますが、同じ形 (中心座標等) や影の画像を使用することがあると思い

ます。アイコンの表示を効率化するために `GIcon` クラス (コンストラクタ) を使用すれば、使用するときコンストラクタで定義したプロパティを呼び出し、地図上に表示することができます。

```
// Create a base icon for all of our markers that specifies the shadow, icon
// dimensions, etc.
var baseIcon = new GIcon();
baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png";
baseIcon.iconSize = new GSize(20, 34);
baseIcon.shadowSize = new GSize(37, 34);
baseIcon.iconAnchor = new GPoint(9, 34);
baseIcon.infoWindowAnchor = new GPoint(9, 2);
baseIcon.infoShadowAnchor = new GPoint(18, 25);

// Center the map on Palo Alto
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);

// Creates a marker whose info window displays the letter corresponding to
// the given index
function createMarker(point, index) {
    // Create a lettered icon for this point using our icon class from above
    var letter = String.fromCharCode("A".charCodeAt(0) + index);
    var icon = new GIcon(baseIcon);
    icon.image = "http://www.google.com/mapfiles/marker" + letter + ".png";
    var marker = new GMarker(point, icon);

    // Show this marker's index in the info window when it is clicked
    var html = "Marker <b>" + letter + "</b>";
    GEvent.addListener(marker, "click", function() {
        marker.openInfoWindowHtml(html);
    });

    return marker;
}

// Add 10 random markers in the map viewport
var bounds = map.getBoundsLatLng();
var width = bounds.maxX - bounds.minX;
var height = bounds.maxY - bounds.minY;
```

```

for (var i = 0; i < 10; i++) {
    var point = new GPoint(bounds.minX + width * Math.random(),
                           bounds.minY + height * Math.random());
    var marker = createMarker(point, i);
    map.addOverlay(marker);
}

```

[View example \(iconclass.html\)](#) (サンプル参照)

XML と AJAX (Asynchronous JavaScript + XML) をグーグルマップで利用する

以下のサンプル例では、表示したい座標 (lat/lng) をリストとして持っている XML を静的に読み込み、その XML の読み込みが終了した時点で、XML を検証して座標リストを展開し、マーカーを表示しています。

```

// Center the map on Palo Alto
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.centerAndZoom(new GPoint(-122.141944, 37.441944), 4);

// Download the data in data.xml and load it on the map. The format we
// expect is:
// <markers>
//   <marker lat="37.441" lng="-122.141"/>
//   <marker lat="37.322" lng="-121.213"/>
// </markers>
var request = GXmlHttp.create();
request.open("GET", "data.xml", true);
request.onreadystatechange = function() {
    if (request.readyState == 4) {
        var xmlDoc = request.responseXML;
        var markers = xmlDoc.documentElement.getElementsByTagName("marker");
        for (var i = 0; i < markers.length; i++) {
            var point = new GPoint(parseFloat(markers[i].getAttribute("lng")),
                                     parseFloat(markers[i].getAttribute("lat")));
            var marker = new GMarker(point);
            map.addOverlay(marker);
        }
    }
}
request.send(null);

```

[View example \(async.html\)](#) (サンプル参照)