Google API Expert
が解説する

Google Maps API
プログラミングガイド

Google API Expert
勝又 雅史・古簇 一浩
石丸 健太郎・安藤 幸央［共著］

完全対応
「Maps JavaScript API V3」

スマホ時代に特化した
Googleマップ活用の決定版！

Maps JavaScript API
Static Maps API
Geocoding API
Directions API
Places API
Google Maps API Premier
Fusion Tables
Latitude API
Prediction API
HTML5
jQuery Mobile
Google App Engine

ダウンロードして使える
サンプルコード付き!

インプレスジャパン

WARNING

はじめにお読みください

Google API Expert

Google Maps API

プログラミングガイド

Google API Expert　勝又 雅史・古簇 一浩・石丸 健太郎・安藤 幸央 [共著]

インプレスジャパン

本書で使用したサンプルプログラムのソースは以下のURLからダウンロード可能です。

http://www.impressjapan.jp/books/3116

＊ 本書の内容は、2011年9月の情報に基づいています。記載したURLやサービス内容などは、予告なく変更される可能性があります。

＊ 本書の内容によって生じる直接的または間接的被害について、著者ならびに弊社では一切の責任を負いかねます。

＊ 本書中の社名、製品・サービス名などは、一般に各社の商標、または登録商標です。本文中に©、®、™は表示していません。

はじめに

Google Maps APIは、2005年6月に初めてversion 1がリリースされました。それからAjaxやHTML5、CSS3など、JavaScriptやブラウザを取り巻く環境の変化、そしてAndroid、iPhoneといったスマートフォンの普及、Google App EngineやAmazon Web Servicesといったクラウドの登場など、大きな技術革新がありました。2011年現在、Google Maps APIはversion 3まで進化し、そして今も新しい機能を作り続けています。

本書では、そのGoogle Maps APIを、Google Maps API Expert 4人が2011年9月現在の最新版にあわせて紹介します。

私が担当した章では、過去に開発してきた案件の中で実際にリクエストされてきたものや、Webサイトやグループで質問されることの多かったものを中心に取り上げています。たとえば情報ウィンドウ（吹き出し）のデザインを変更する、というのは、デザイナーが描いたデザインに合わせなくてはならないときの例です。本書で紹介しているのはあくまでも簡略化した例ですが、基本となる部分は押さえてあります。

この本が実際の開発現場で少しでも参考になれば幸いです。

勝又 雅史

Googleマップが出た当時はAPIが用意されていませんでした。仕方なく解析してGoogleマップを表示するプログラムを作成しました。その後、APIが整備されより使いやすく高性能なものになっていきました。現在はパソコンだけでなくモバイル環境でも活用できるまでになり

ました。またWebページで使用するだけでなくモバイルアプリケーションとして作成することもできます。

本書がGoogleマップをどのように活用するか、そのヒントや手助けになれば幸いです。

古簗 一浩

Googleマップが世に出てからもう何年が過ぎるでしょう。数多くのGoogle APIの中でも老舗中の老舗となった今でこそ最新技術の話題に上ることの少ないMaps APIですが、その利用数は今でもダントツ上位に位置するそうです。そして実はHTML5やAndroidの技術とも非常に親和性が高く、近年のスマートフォンやタブレットの隆盛も相まって、より手軽に、より身近に、地図を使ったサービスを扱える環境が整いつつあるといった状況ではないでしょうか。本書で扱うさまざまな事例が、あなたにとってピリリとパンチの効いた「アイデア・スパイス」となれば幸いです。

石丸 健太郎

リチャード・ソウル ワーマンの地図帳はとても見やすく利用しやすいものでした。紙面の1ページ分の縮尺が、ちょうど車で1日分の走行距離で描かれ、距離感を把握できるようになっていました。また、限られた紙面をめくりながら、見たい地域が紙の端になったときに、次にどのページを見ればよいのか、すぐにわかる仕組みも考えられていた

のです。しかし、とてもわかりやすいながらも「地図を読み取る能力」は必要でした。迷子になる人はどんなに使いやすい地図があっても迷子になっていたのです。

そしてGoogleマップの登場は大変な驚きでした。それまでは上下左右にボタンを押しながら見るWebの地図が一般的で、それでも十分有益だったのですが、Googleマップではシームレスに地図を見ることができ、地図というもの自体の認識が大きく変わりました。

全世界分の衛星写真が見られることも驚きでした。GPSと連動して自分が今いる場所や、目的地までの経路もわかります（太平洋を泳いで渡るという経路が表示されることもありますが...）。

さらに、資料としての地図ではなく、さまざまな情報がまとめられ、利用者の目的のための地図として、Googleマップの見やすく、利用しやすいデザインも、多くの地図サービスに影響を与えています。

Googleマップの登場で、地図には限りがなく、地表は全部つながっているものだという認識が強くなりました。生まれたときからインターネットや携帯電話が使えるデジタルネイティブ世代がよく話題になりますが、さらにGoogleマップ登場前と後では、位置や空間を把握する人間の能力さえも違ってくるかもしれません。人間が地球上で生きて、移動する限り（もしかしたら宇宙へ進出したとしても？）、地図のない生活は考えられません。Googleマップ（Maps API）によってますます便利なサービスが作られ、多くの人が快適に移動し、場所を探し、迷子にならずになってほしいものです。

安藤 幸央

対象読者

本書は、Google Maps APIを使って実際にWebシステムを開発している方、または趣味などで実際にGoogle Maps APIを使っている方を想定しています。

各章では、JavaScriptやAjaxをある程度使いこなすことができ、Geolocation API、Android/iPhoneなどのスマートフォンのプログラミングや、PhoneGap、Google App Engine、FusionTablesなどの技術について、いくつか知っている、またはとても興味があるという方、もしくはそれらを使って仕事をしている、という方を対象にしています。

入門書としては『Google Maps APIプログラミング入門』が幅広くカバーしていますのでそちらをお勧めします。本書は入門書では物足りない！という方にオススメです。

動作環境

Google Maps API JavaScript version3が動作する環境は次の通りです。

- IE 7.0 ～（Windows）
- Firefox 3.0 ～（Windows、Mac OS X、Linux）
- Safari 4 ～（Mac OS X、iOS）
- Chrome（Windows, Mac OS X, Linux）
- Android
- BlackBerry 6 ～
- Dolfin 2.0（Samsung Bada）

第1章・第2章・第3章

JavaScriptのコードは主にGoogle Chrome（Mac OS X）/Safari 4（Mac OS X）で動作確認しています。標準的なJavaScriptの書き方をしているため、基本的にはそのほかの環境でも問題なく動くと思われます。

第4章

- iPhone/iPad（iOS 4.3以降）
- Android 2.3以降

※ 各キャリアのケータイ端末は多岐にわたるため、必ずしも動作するとは限りません。

- ブラウザはFirefox
- OSはMacOS X/Windows 7を利用

第5章

5-1

Android SDK 2.3でWebViewを使った開発を行っています。

5-2 〜

- iPhone/iPad（iOS 4.3以降）
- Android 2.2以降
- MacOS X（開発環境は10.6以降）/Windows XP
- PhoneGap、NimbleKit、Corona

第6章

Google App Engine SDK 1.5.4/Python 2.5（Mac OS X）を使用しています。

Contents

第1章　イントロダクション

第2章　Hello World

第3章 Maps APIの活用

第5章　モバイルフレームワーク

第6章　Google App Engineとの連携

Google Maps API
プログラミングガイド
by Google API Experts

Chapter 1

イントロダクション

［第 1 章］

第 1 章では、Google Maps API version3 の特徴について紹介しています。今まで使われていたGoogle Maps API version2 とどう違うのか、というところを中心に紹介しています。また一般的によく使われる開発環境・デバッグツールについても紹介しています。

勝又 雅史

1-1 Google Maps APIとは

Google Maps APIとは、Google社が自社の地図サービスであるGoogleマップ（http://maps.googleapis.com）の機能の一部を、外部の開発者向けに使えるようにしたサービスです。Google Maps APIを使うことで、Googleマップを使った地図を一般のホームページ内に埋め込むことができます。Google Maps APIを使わなくても、Googleマップの地図を一般のホームページに埋め込めますが、APIを使うと、デザインのカスタマイズや独自機能の追加、データベースとの連動などを行うことができます。

Googleマップの魅力の1つは、オンラインのブラウザさえあれば、世界地図から住宅地図レベルまで、世界中を手軽に見られることでしょう。Google Maps APIではそのGoogleマップと同じ地図を見ることができます。Google Maps APIは、Googleマップの機能の一部を、一般の開発者向けに利用できるようにしたサービスです。また、ルート検索やGoogleストリートビュー、航空写真、お店や建物などのランドマーク検索なども利用でき、データはほぼ最新のものが使用されています。国や都市にもよりますが、地図のデータも頻繁に更新されています。これだけの機能が揃っているのにもかかわらず、それらを基本的には無償で使うことができます。

Google Maps APIには無償版APIと有償版APIがありますが、主な違いは利用規約の制限と上限利用量です。無償版APIを使うためには、次の基本原則を理解する必要があります。

Googleマップ無償版API利用のための基本原則

① 一般的に公開されたWebサイトで使用しなければならない
② 誰でも無償で同じ機能が利用できる必要がある
③ 会員制サイトで使用する場合は、誰でも無償で登録でき、誰でも同じ地図の機能が利用できる状態でなければならない
④ 地図上のロゴや属性を改変したり、見えにくくしてはならない

無償版APIと有償版APIの詳細については、本書巻末の付録「Google Maps API Premierについて」を参照してください。

なお、1-3以降の解説において、特に断りのない限り「Maps API」は同名のAPIファミリーのうち、JavaScript版であるGoogle Maps JavaScript APIを指しています。

1-2 Google Maps APIファミリー

Google Maps APIには、現在５つのサービスがあります。

（1）JavaScript版（version2, version3）

JavaScriptを使ってWebサイトに地図を埋め込むことができます。AndroidやiPhoneなどのスマートフォンでも動作します。なお、Google Maps API version2は、version3が正式版に採用されたことで廃止が決定しています。2013年５月でサポートが終了になりますので、それまでにversion3へ移行してください。

（2）Google Maps API for Flash

Flashを使ってFlashベースのWebサイトに地図を埋め込むことができます。Flash版のみ3D地図を表示することもできます。しかし、残念ながら2011年９月に廃止が発表されました。2014年９月でサポートが終了になりますので、それまでにGoogle Maps API version3（JavaScript版）へ移行してください。

（3）Google Maps Image APIs（Static Maps、Street View）

URLパラメータによるクエリだけで地図やストリートビューの画像を取得できるAPIです。携帯電話（フィーチャーフォン）のブラウザなどのJavaScriptが動作しない環境で使用できます。

（4）Google Earth API

Google Earth APIは、バーチャル地球儀ソフト「Google Earth」の機能をJavaScriptを使ってWebサイトに埋め込むためのAPIです。Google Earthの3Dの世界観をそのままWebサイトに埋め込むことができます。

（5）Google Maps Web Services

URLリクエストを使って、ジオコーディングやルート検索・場所検索などを利用できます。JSONまたはXML形式で結果を受け取ることができます。

本書では近年増えてきたスマートフォン向けサイトでのGoogle Maps APIの利用の需要に応えるために、（1）Google Maps API version3、（3）Google Static Maps API、（5）Google Maps Web Servicesを中心に紹介しています。

また、Google Maps APIファミリーのメンバーではありませんが、Google Maps APIと組み合わせのよいGoogle FusionTablesというサービスがあります。Google FusionTablesには、Googleの公開データベース上にデータを取り込むと、地図上に自動でデータをマッピングしてくれる機能があります。「3-5 Fusion Tables Quick Hands-on」「3-6 ポリゴンで都道府県を表示する」「3-7 Fusion Tablesで特定の位置を中心としたデータを表示する」で紹介します。

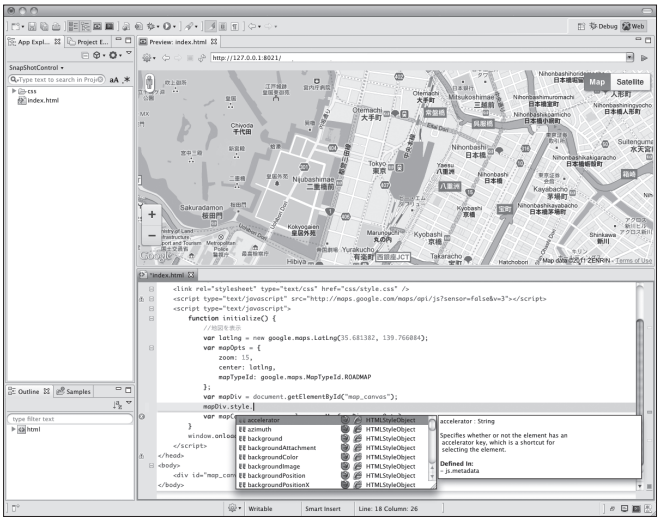
1-3 開発環境

Google Maps APIを使ったページを開発するときに便利な開発環境を紹介します。Google Maps APIはJavaScriptのライブラリなので、シンプルなテキストエディタでもよいのですが、開発環境を使いこなせれば開発工数が大幅に削減できます。

1-3-1 Aptana Studio 3.0（クロスプラットフォーム）

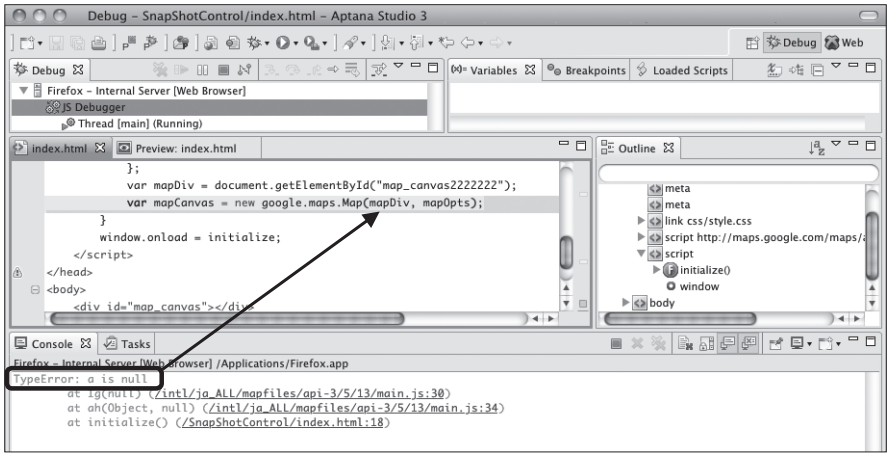
Aptana StudioはオープンソースのWebオーサリングツールです。Eclipseをベースとして開発されています。基本はHTMLコードエディタですが、Previewウィンドウを開きながらコーディングすると、コードの内容がすぐに反映されるので開発がスムーズに進みます（Previewウィンドウは内蔵のWebkitエンジンで描画されます）。また、コードアシスト機能、JavaScriptの文法チェック／フォーマッタなどの機能もあります。

Aptana Studio 3.0での開発画面



デバッグモードでは、ブレークポイントや変数の内容の確認などが行えます。エラーが発生したときにはスクリプトコードのトレースが行われるので、どのような流れでエラーが発生したのかがわかります。次の図では、index.htmlの18行目「var mapCanvas = new google.maps.Map(mapDiv, mapOpts);」の部分で「a is null」というエラーが発生しています。「a」は関数の第1引数のことです（「function(a,b)」のように書くため）。つまり「mapDiv = null」がエラーの原因であることがわかります。

JavaScriptのデバッグ画面

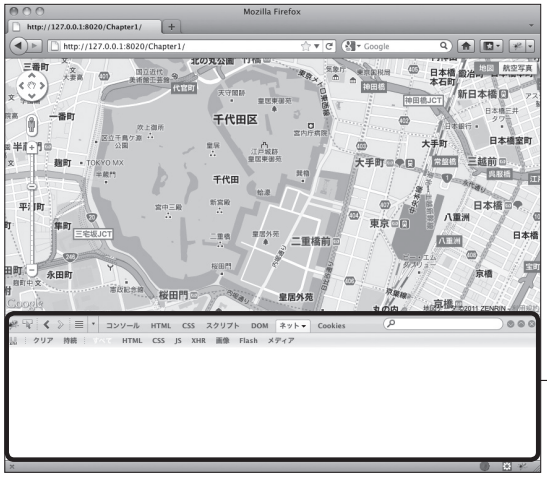


Aptana Studio 3はクロスプラットフォーム環境で動作するので、Mac OS XやWindowsでも動作します。また、Eclipseプラグインも提供しています。

1-3-2 Firebug (Windows/Mac OS X)

Firebugは Firefox用 JavaScriptデバッグツールです。コンソール、HTMLインスペクター、CSS／スクリプトデバッガ、DOMビューワー、ネットワークモニターの機能があります。JavaScriptでエラーが発生すると、コンソールにエラーの内容の詳細情報が表示されます。コンソールには、プログラムからconsole APIを通じて任意の情報を表示することもできます。

地図の下に表示されているのがFirebug



エラーが起きたときの画面

・Firebug console API のリファレンス
　http://getfirebug.com/wiki/index.php/Console_API

Firebug は Firefox 用のアドオンですが Firebug Lite を使うと、Firefox 以外のブラウザでも一部の機能を使うことができます。<head> 〜 </head> 間に

```
<script type="text/javascript" src="https://getfirebug.com/firebug-lite.js">
</script>
```

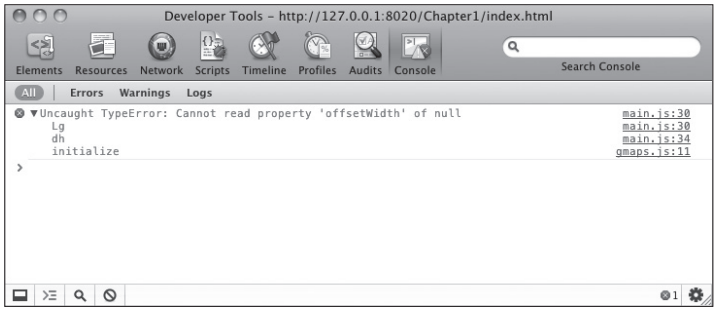
とするだけで利用できます。特にデバッグ機能の少ない Internet Explorer では有用です。

・Firebug Lite
　http://getfirebug.com/firebuglite

1-3-3　Chrome DevTools（Windows / Mac OS X）

Chrome DevTools は、Google Chrome に標準で搭載されているデバッグツールです。Firebug と同じく、Console、Script パネル、Resources（Web ページで読み込んだファイルや Cookie などの確認）、Network パネルの機能などがあります。また、メニューの「ツール → タスクマネージャー」を開くと、各ページでどれだけメモリが使用されているのかも確認できます。

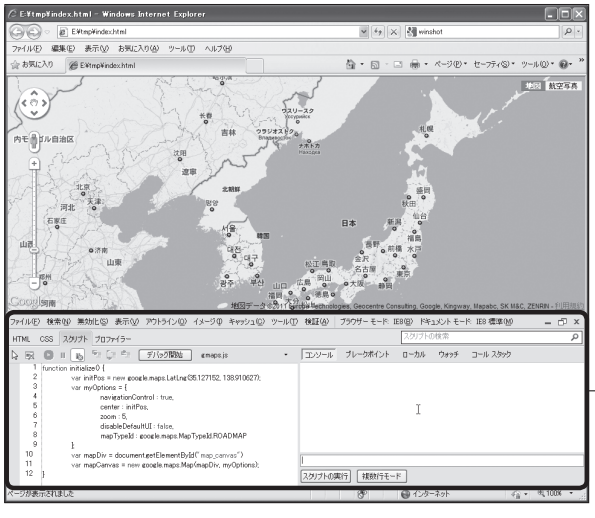
JavaScriptでエラーが発生したときの画面



1-3-4 Internet Explorer開発者ツール（IE8以上）

Internet Explorer 8から標準機能として搭載されたツールです。HTMLインスペクター、CSS編集、スクリプトのデバッグ、プロファイリングなどができます。Internet Explorer 6/7 の場合は、Internet Explorer Development Toolbarというプラグインをインストールすると、同様の機能が利用できます。

地図の下に表示されているのが開発者ツール

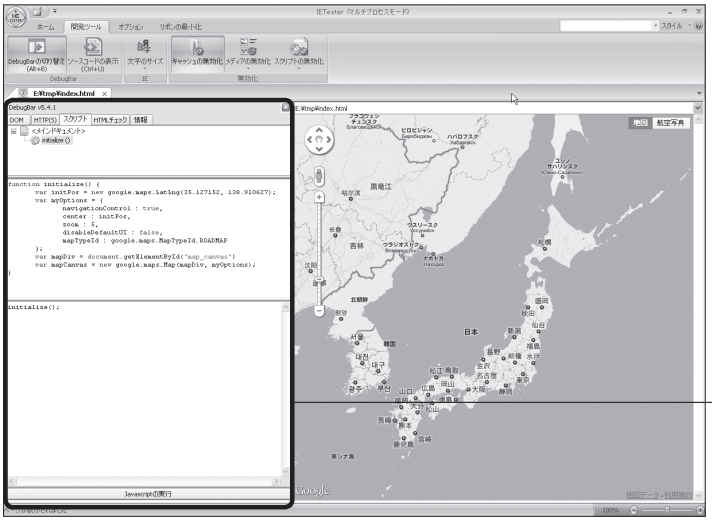


1-3-5 IETester

IETesterは、Internet Explorer 5.5～10まで対応しているInternet Explorerエミュレータです。各バージョンのレンダリングエンジン（Trident）を内蔵しています。複数の異なるバージョンのInternet Explorerの挙動を確認できるツールとして広く使用されています。DebugBarプラグインを

インストールすると、IETester で JavaScript のデバッグをすることが可能になります。

IETesterでのデバッグ画面



1-4 Google Maps API v2からv3への移行

Google Maps API version2から3では、基本的な設計から見直され、コーディングスタイルも大きく変化しました。

1-4-1 名前空間の統一

Google Maps APIの名前空間がすべて「google.maps.*」に統一されました。以下に、v2からv3のクラス変換表を示します。表でわかるように全機能の互換性を持っているわけではありません。ほかのライブラリと重複するような機能（GDownloadUrlなど）や、ほかの方法で実現できる機能（GControlなど）は移植されていません。

v2からv3の主なクラス変換表（2011年9月現在）（v3のgoogle.maps.名前空間は省略）

v2でのクラス名	v3でのクラス名
GMap2	Map
GMapOptions	MapOptions
GInfoWindow	InfoWindow
GInfoWindowTab	（該当なし）
GInfoWindowOptions	InfoWindowOptions
GMarker	Marker
GMarkerOptions	MarkerOptions
GPolyline	Polyline
GPolylineOptions	PolylineOptions
GPolyEditingOptions	（該当なし）
GPolyStyleOptions	PolylineOptions
GPolygon	Polygon
GPolygonOptions	PolygonOptions
GScreenOverlay	（該当なし）
GScreenPoint	（該当なし）
GScreenSize	（該当なし）
GGroundOverlay	GroundOverlay
GIcon	MarkerImage

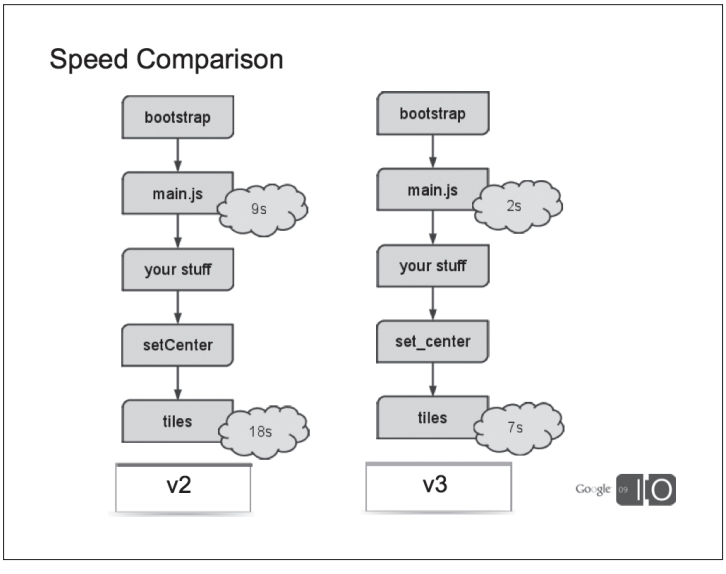
v2でのクラス名	v3でのクラス名
GPoint	Point
GSize	Size
GBounds	（該当なし）
GLatLng	LatLng
GLatLngBounds	LatLngBounds
GControl	（該当なし）
GTileLayerOptions	（該当なし）
GTileLayerOverlayOptions	（該当なし）
GEvent	event
GEventListener	event
GXmlHttp	（該当なし）
GXml	（該当なし）
GXslt	（該当なし）
GLog	（該当なし）
GDraggableObject	（該当なし）
GDraggableObjectOptions	（該当なし）
GGeoStatusCode	GeocoderStatus
GGeoAddressAccuracy	（該当なし）
GClientGeocoder	Geocoder
GGeocodeCache	（該当なし）
GFactualGeocodeCache	（該当なし）
GMarkerManager	（該当なし）
GMarkerManagerOptions	（該当なし）
GGeoXml	（該当なし）
GDownloadUrl	（該当なし）
GBrowserIsCompatible	（該当なし）
GDirections	DirectionsService
GDirectionsOptions	DirectionsOption
GTravelModes	DirectionsRequest
GRoute	DirectionsRoute
GStep	DirectionsStep
GTrafficOverlay	TrafficLayer
GTrafficOverlayOptions	（該当なし）

v2でのクラス名	v3でのクラス名
GAdsManager	AdUnit
GAdsManagerOptions	AdUnitOptions
GStreetviewPanorama	StreetViewPanorama
GStreetviewPanoramaOptions	StreetViewPanoramaOptions
GStreetviewOverlay	（該当なし）
GStreetviewClient	DirectionsService
GStreetviewClient.ReturnValues	StreetViewStatus
GStreetviewData	StreetViewPanoramaData
GStreetviewLocation	StreetViewLocation
GStreetviewLink	StreetViewLink
GPov	StreetViewPov
GStreetviewPanorama.ErrorValues	（該当なし）
GGoogleBarListingTypes	（該当なし）
GGoogleBarLinkTarget	（該当なし）
GGoogleBarResultList	（該当なし）
GMapPane	MapPanes
GOverlay	OverlayView
GControl	（該当なし）
GControlPosition	ControlPosition
GControlAnchor	ControlPosition
GMapTypeContro	MapTypeControlOptions から指定
GMenuMapTypeControl	MapTypeControlOptions から指定
GHierarchicalMapTypeControl	MapTypeControlOptions から指定
GMapType	MapType
GMapTypeOptions	（該当なし）
GLayer	OverlayView
GTileLayer	ImageMapType
GTileLayerOverlay	（該当なし）
GCopyrightCollection	（該当なし）
GCopyright	（該当なし）
GProjection	Projection
GMercatorProjection	MercatorProjection

1-4-2　パフォーマンスの向上

Google Maps API v3はスピード、特にモバイル端末（Android/iPhone/iPadなど）のブラウザにおける地図の表示スピードの向上に重点をおいて開発されました。Google Maps API v2ではライブラリのサイズが170kB（2011年9月現在）もあり、3G回線ではロードするだけでも時間がかかってしまいましたが、Google Maps API v3では遅延ロードを内部的に採用することにより最初のダウンロードはわずか35kB（2011年9月現在）です。その35kBのライブラリで地図を描画して、遅延してマーカーや情報ウィンドウなどの必要なライブラリを読み込んでいます。また、v3では必要のないコードはダウンロードしない仕組みになっています。

Google Maps API v2/v3の実行速度比較（Google I/O 2009 Maps APIs & Mobileより）



（2011年9月現在では、v3の「set_center」は「setCenter」に変更されています）

なお、現在は図中のパフォーマンスよりもさらに向上しています。この図はあくまでもv3のほうがv2よりもパフォーマンスに優れているということを理解していただくためのものです。

1-4-3　MVCモデルの導入

Google Maps API v3からMVC（Model View Control）モデルが導入されました。これによってデータ／表示処理／制御処理をそれぞれ独立してコーディングしやすくなりました。

それを支えているのが、KVO（Key Value Object）のbind（バインド）処理です。MVCObjectクラスは、Keyに関連したObjectを保存できます。そのvalueが変化したとき、(key)_changedイベントが発生します。bind処理はその値の変化を直接別のMVCObjectに関連付ける処理をします。

Google Maps API v3のクラスの多くは、このMVCObjectを継承しています。たとえば、地図の中心に、常に情報ウィンドウが表示されるプログラムを作るとします。InfoWindowには地図の中心座標が表示されます。

プログラムの実行画面

Google Maps API v2のコードでは以下のようになります。

Google Maps API v2を用いた場合のコード　sample　C01→1-4-3v2.html

```
var mapCanvas = new GMap2(mapDiv);
mapCanvas.setCenter(latlng, 15, G_NORMAL_MAP);
mapCanvas.openInfoWindow(latlng, "地図をドラッグ&ドロップしてください");
GEvent.addListener(mapCanvas, "move", function(){
  var center = mapCanvas.getCenter();
  mapCanvas.openInfoWindow(center, center.toUrlValue(4));
});
```

Google Maps API v3のコードでは次のようになります。

Google Maps API v3を用いた場合のコード　sample　C01→1-4-3v3.html

```
var mapCanvas = new google.maps.Map(mapDiv, mapOpts);
var info = new google.maps.InfoWindow({
  content : "地図をドラッグ&ドロップしてください"
});
```

```
info.open(mapCanvas);
info.bindTo("position", mapCanvas, "center");
info.position_changed = function() {
  info.setContent(info.get("position").toUrlValue(4));
};
```

それぞれのコードは処理の方法がまったく異なります。v2 の場合は、GMap2 クラスが openInfoWindow メソッドを持っているため、地図と情報ウィンドウが一体となってしまっています。一方、v3 のコードでは、情報ウィンドウを 1 つのオブジェクトとして生成し、地図の center プロパティと、情報ウィンドウの position プロパティを関連付けています（バインドしています）。地図の center プロパティが変化すると、情報ウィンドウの position プロパティは自動的に更新されます。つまり、地図と情報ウィンドウは完全に切り離されていて、MVCObject のバインドによって処理が伝達されているわけです。

NOTE ● MVCObject のサンプルは「2-4 サイドバー付きの地図を作る」で紹介しています。

1-4-4　そのほかの変更点

Google Maps API v3 ではそのほかに以下の変更点があります。

- Google Maps API v2 ではライブラリのロード時に指定が必要だった API キーが不要になった
- デフォルトの UI（ズームボタンやマップタイプコントロールなど）が自動的に画面サイズに合わせて表示されるようになった

1-4-5　コーディングスタイル

「1-4-3 MVC モデルの導入」でも少し触れましたが、Google Maps API v3 ではすべてのクラスが独立して、クラスによる依存がないように作られています。その結果、コーディングスタイルも大きく変化しました。

一番わかりやすい例としてはオーバーレイでしょう。たとえば、Google Maps API v2では、マーカーを地図に追加する処理は次のようなコードでした。

Google Maps API v2を用いた場合のコード　sample　C01→1-4-5v2.html

```
GEvent.addListener(mapCanvas, "addoverlay", function(overlay) {
  if (overlay.__proto__ == GMarker.prototype) {
    var latlng = overlay.getLatLng();
    mapCanvas.openInfoWindow(latlng, latlng.toUrlValue(4));
  }
});

GEvent.addListener(mapCanvas, "click", function(overlay, latlng) {
  if (overlay == null) {
    var marker = new GMarker(latlng);
    mapCanvas.addOverlay(marker);
  }
});
```

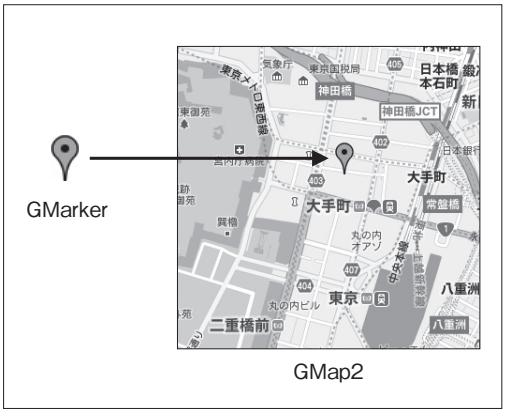
このコードでの注目すべき点は、GMap2のaddoverlayイベントです。このイベントは、地図上にオーバーレイが追加されたことを通知するイベントです。つまり、地図側（mapCanvas）でマーカーが追加されたことを知ることができたのです。

それに対して、Google Maps API v3では、addoverlayに該当するイベントはありません。なぜなら、地図とマーカーはそれぞれ独立しているため、地図側ではマーカーが地図に関連付いたことを知ることができないのです。

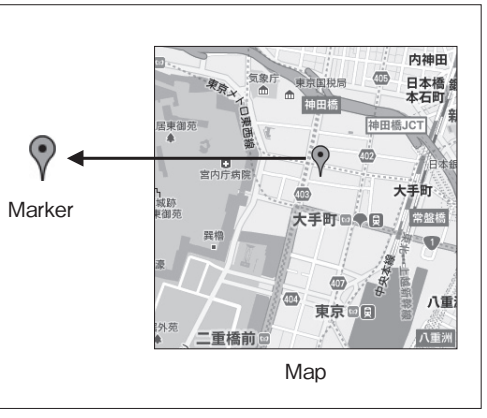
Google Maps API v3を用いた場合のコード　sample　C01→1-4-5v3.html

```
var info = new google.maps.InfoWindow();
google.maps.event.addListener(mapCanvas, "click", function(mouseEvent){
  var marker = new google.maps.Marker({
    position : mouseEvent.latLng,
    map: mapCanvas
  });
  info.setContent(mouseEvent.latLng.toUrlValue(4));
  info.setPosition(mouseEvent.latLng);
  info.open(mapCanvas);
});
```

マーカーを地図に関連付けるのがv2



マーカーに地図を関連付けるのがv3



これだけコーディングスタイルが変わっているので、v2 のコードはそのまま移植できないことが時々あります。そのような場合は、変更された部分だけ新しく設計しなおす必要があります。

Google Maps API
プログラミングガイド
by Google API Experts

Chapter

2

Hello World

［第２章］

第２章では、これからGoogle Maps API v3を使い始めるという方のために、簡単な地図を表示するサンプルを解説します。すでにGoogle Maps API v3を使ったことがある方は、そのまま第３章にお進みください。

勝又 雅史　2-1 ～ 2-4
勝又 雅史・古簇 一浩・石丸 健太郎　2-5

2-1　住所から緯度経度を調べる

そもそも地図を表示するのに基本となる位置情報をどうやって取得するのか、何の位置情報が必要なのか、というところから解説していきましょう。

まず、一言で「位置情報」といってもさまざまなものがあります。住所も位置情報ですし、「○○駅から3軒隣のラーメン屋」でも立派な位置情報です。

しかし、本書の中で「位置情報」といったら「緯度経度」だと思ってください。Google Maps API（以下、Maps API）では、メルカトル図法による地図で表示されます。メルカトル図法とは、球体の地球を平面図として表現するための投影法の1種です。Maps APIでは、メルカトル図法と世界測地系（WGS84）を使って緯度経度を用います。測地系とは、地球上の位置を緯度経度を用いて表すときの前提条件のことです。この2つを使ってGoogleマップの地図を表示します。

2-1-1　Googleマップを使って緯度経度を調べる

Googleマップ（http://maps.google.co.jp）を使って、住所から緯度経度を調べる手順を紹介します。わずか3ステップで緯度経度を調べることができます。

Googleマップにアクセスし、目的地の住所などを入力して検索する