

# SAPPORO CAMP Day02

---

JavaScript & API

# CONTENTS

---

- APIとは
- APIを使うための準備
- 現在地の取得
- 現在地を中心に地図を表示
- 現在地にピンを立てる
- 現在地にinfoboxを表示する
- 他のAPIの紹介

- 本日のゴール -

APIで動くアプリが手元にある！

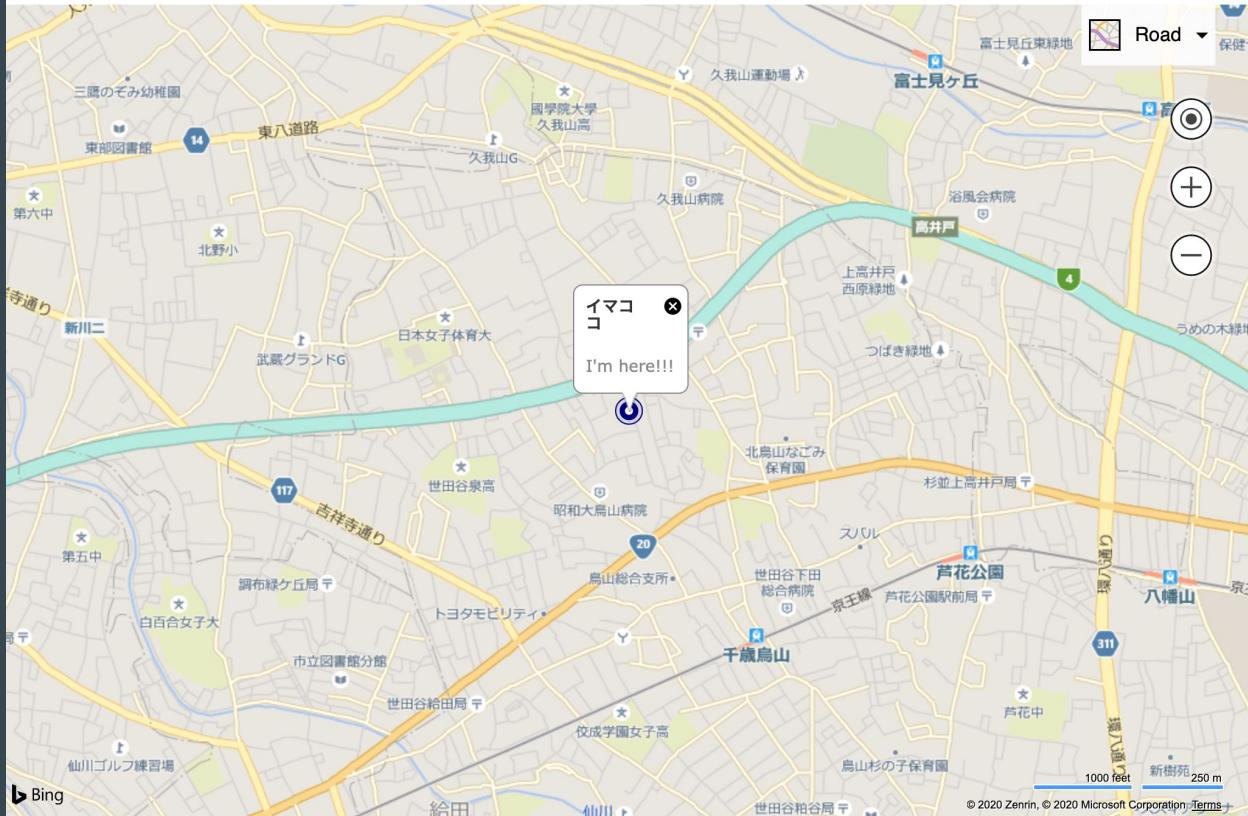
APIの仕組みを知る！

JavaScriptでできることの幅を知る！

# 本日実装するアプリケーション

マップアプリ！！

# オリジナルマップ



# APIとは？？

# Applocation Programming Interface

# APIとは？？

---

- Microsoftが提供する地図API
  - APIはアプリケーションの一部を公開したもの.
  - すごい人が作ったものを使わせてもらえるイメージ！
- 例
  - maps API :リクエストに応じてブラウザに地図を表示する
  - google books api :google booksのデータを返す

# APIとは？？

---

- 利点
  - 他社のデータを利用できる！
  - 自分で作らなくて良いので開発速度がアップ！
  - いろいろなデータを組み合わせてアプリケーションを実装できる！

スタートアップや個人開発に最適！

# APIとは？？

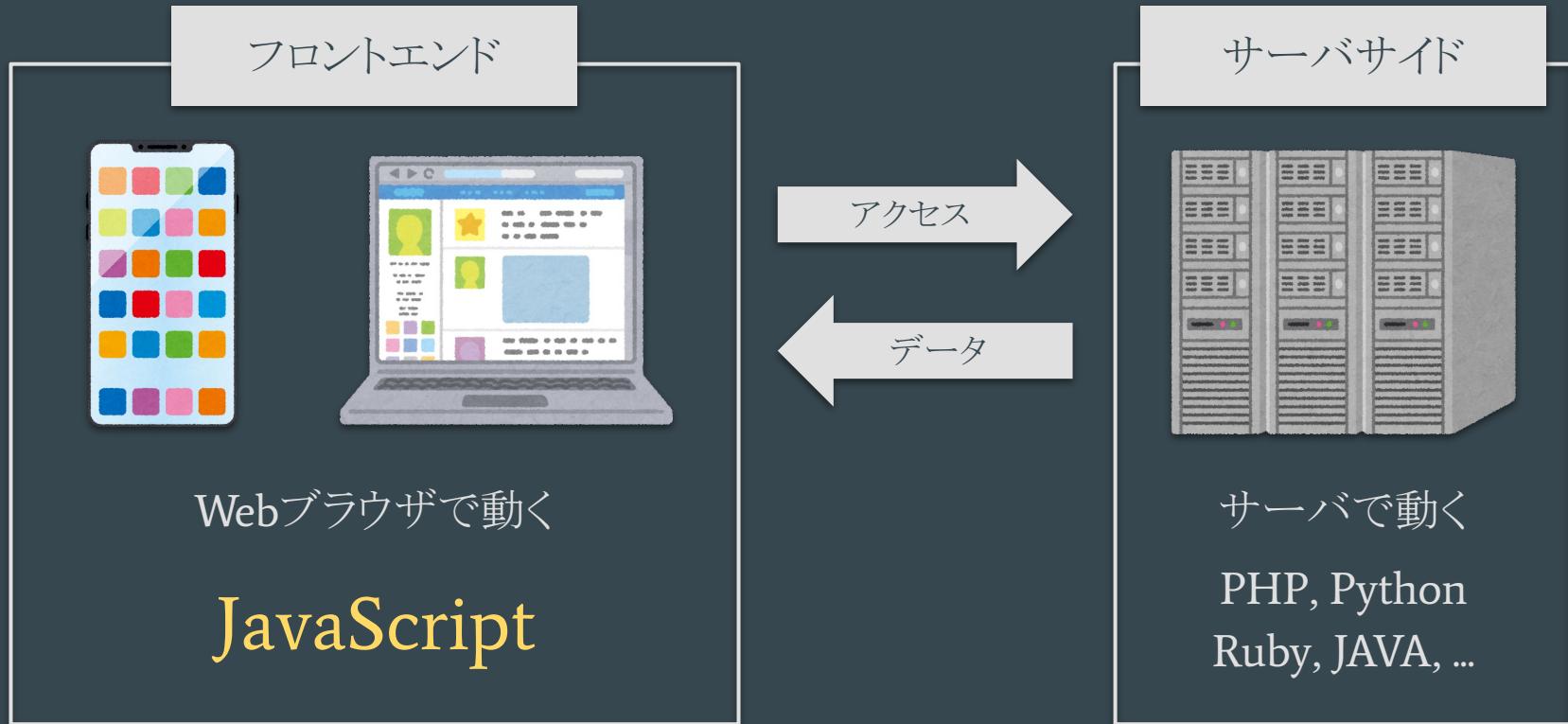
---

- 仕組み
  - 基本は「リクエスト」と「レスポンス」

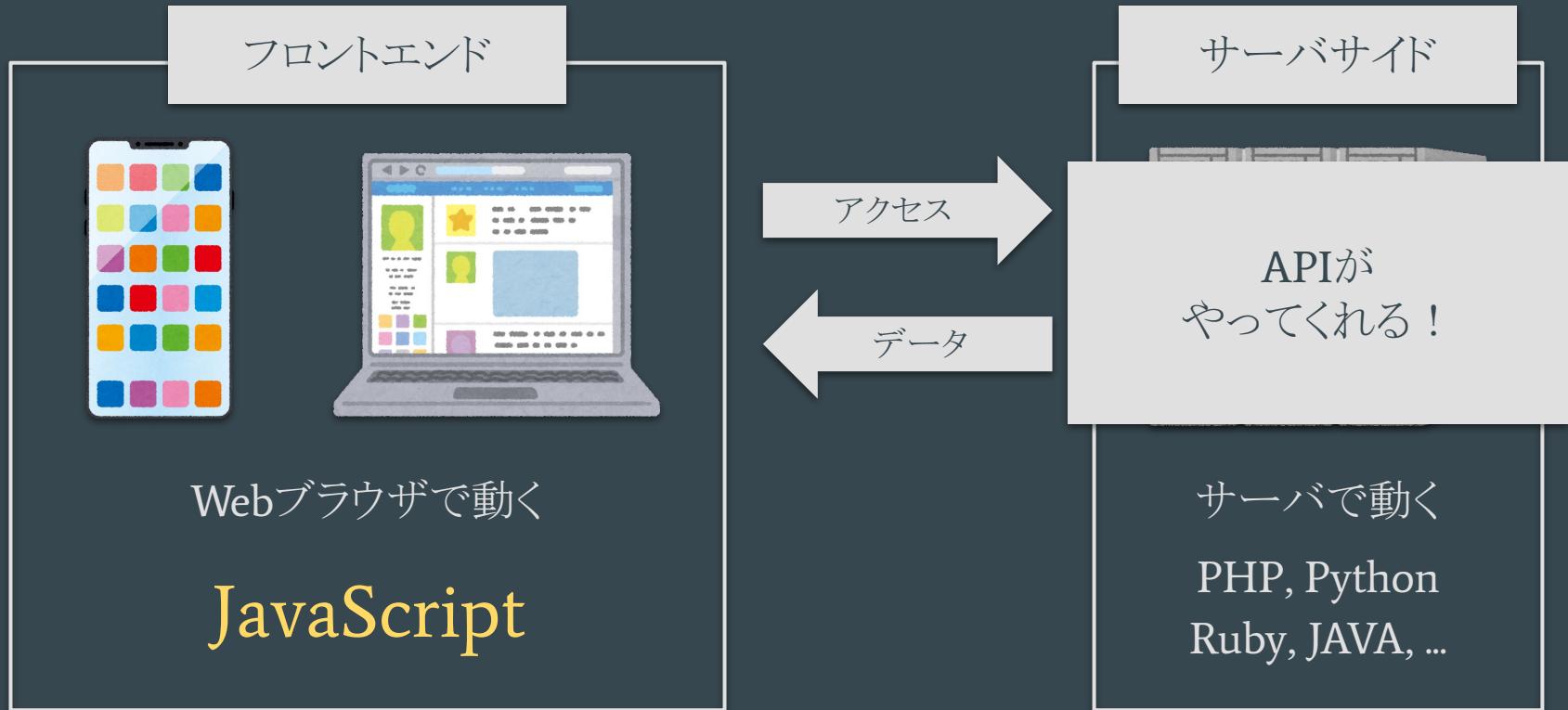


Webアプリケーションの構造そのもの！

# Webアプリケーションの構造

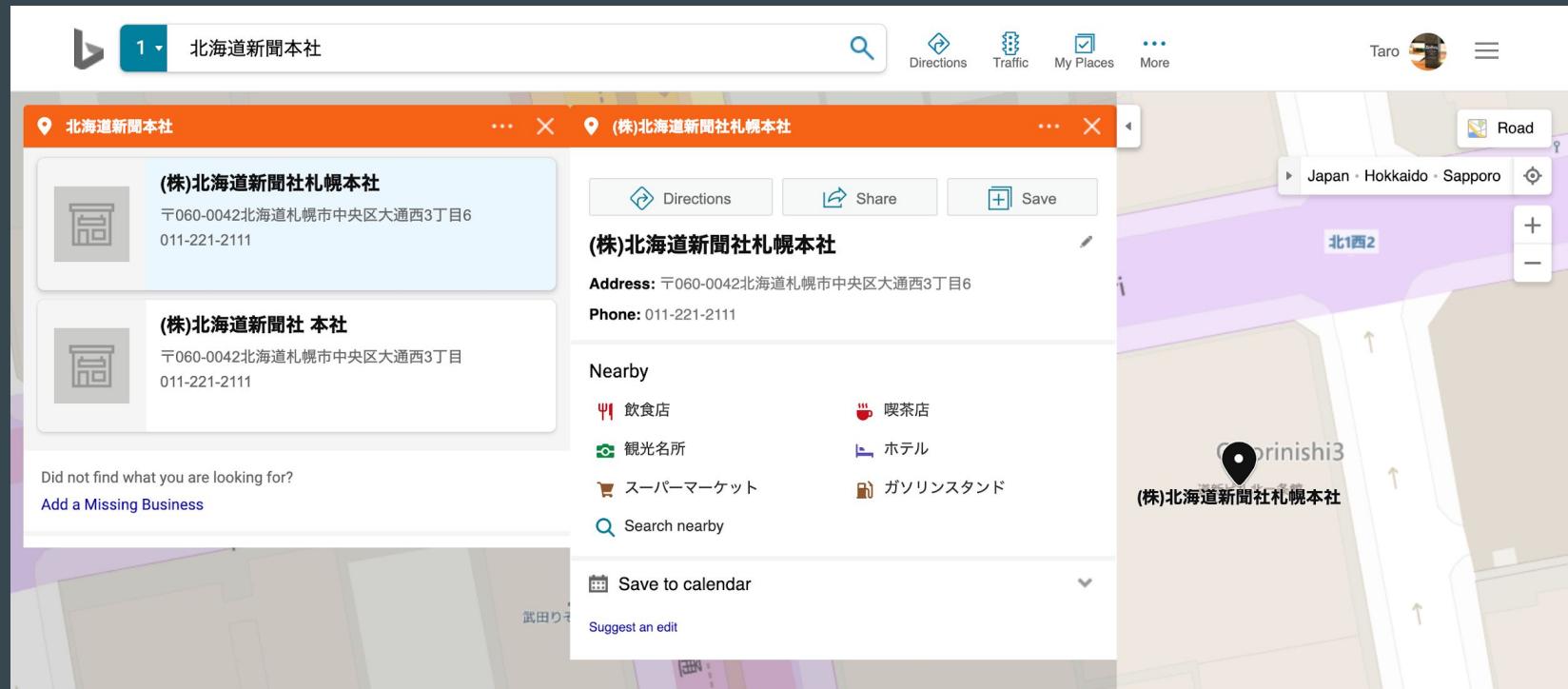


# Webアプリケーションの構造



# 本日使用するAPI

# bing maps API



# bing maps API

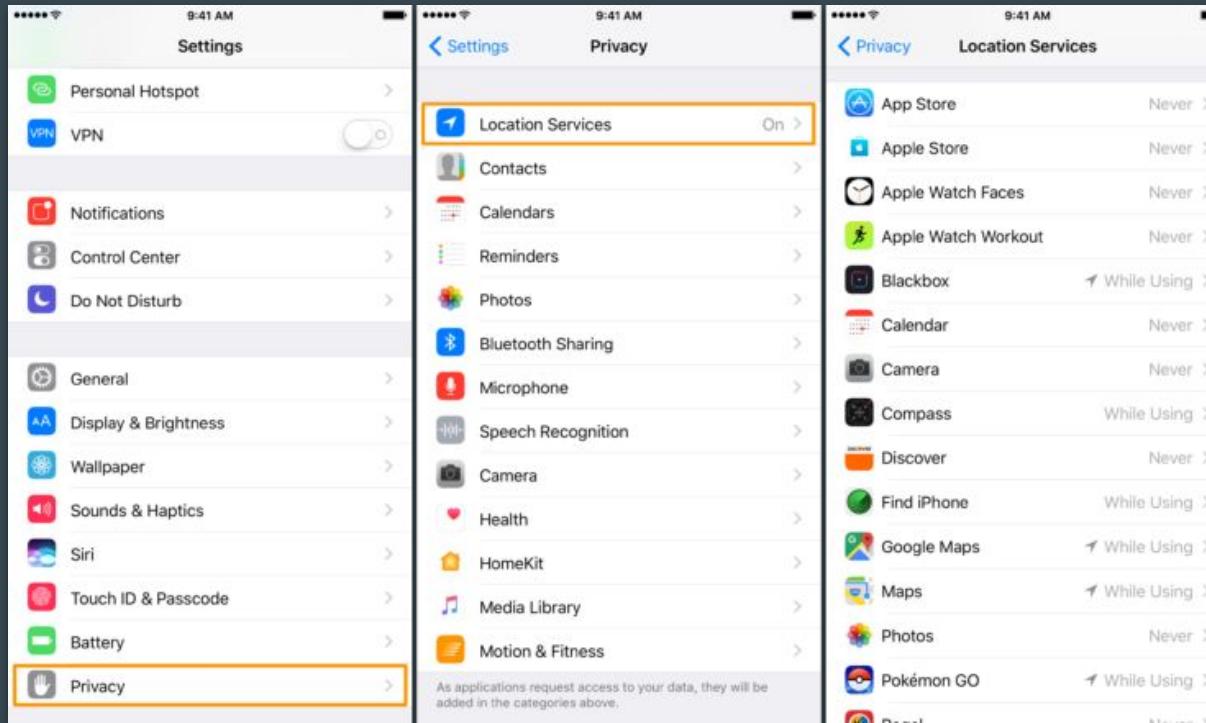
---

- Microsoftが提供する地図API
  - 位置情報を利用して様々な機能を提供する！
  - 地図の表示, ルート検索
  - ピンの設置, infobox, etc...
  - 住所から緯度経度を取得
  - 緯度経度から住所を取得
  - 無料

 位置情報 

# 位置情報の利用

- 世の中のサービスの8割程度が利用.. ! !



# 位置情報の利用

---

- 地図 / ルート検索
  - 位置情報とマップの組み合わせ
    - 今までではユーザの現在地がわからなかつた
  - 地図上に店舗データなどを表示
    - 住所だけではわかりにくい
  - どこに行けばよいかがわかる
    - 道を間違っても自動的に現在地からのルートが分かる

# 位置情報の利用

---

- 店舗のマーケティングリサーチ
  - 電波強度を測定
    - 強ければ店内
  - 滞在時間で顧客を分ける
    - 長ければ店内, 短ければ店外
  - 営業時間で調べる
    - 営業時間内なら顧客, 時間外は通行人

# 位置情報の利用

---

- 店内の販促ツール
  - 棚の近くにビーコンを設置
  - ユーザが近づくとLINEにクーポンを配信
    - ユーザにアプリ等のツールが必要ない
  - 購入レシートを送信するとポイントが当たる
    - 再度の来店を促す

# APIキーの取得

# APIキーとは？？

---

- APIキー

- APIにリクエストを送るためにはユニークな文字列が必要.
- 設定を行ったアプリケーションのみ動作させるため.
- (無断で使用されないように. . ! )
- この文字列をAPIキーを呼び、APIの種類問わず必須の場合が多い.

# Microsoftアカウントの取得

---

- 下記URLを参考に進めよう！
  - <https://kaoruya.org/blog/ms-account/#i-3>
  - 「既に持ってるメアドを使う場合」で進める.
  - 取得できたら一旦待機！
  - すでに持っている場合は既存のアカウントでOK.

<https://www.bingmapsportal.com>にアクセス！



Bing maps | Dev Center

Welcome

The Bing Maps Dev Center provides the tools and resources you need to develop with Bing Maps. You can store, access, and keep track of your store locations or other spatial data through our online data source management system. You will also receive important announcements around your Bing Maps account.

[Sign in](#)

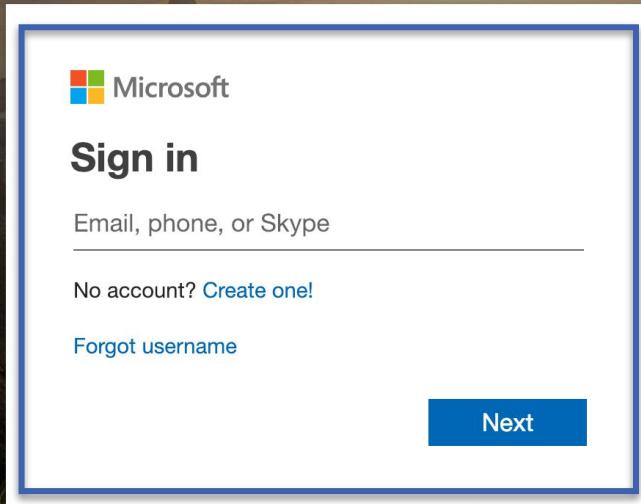
**First time Bing Maps developer?**

To get started developing with Bing Maps, you will need a Bing Maps key. To create a key:

1. [Sign in](#) to the Bing Maps Dev Center with your existing Microsoft Account or create a new one.
2. Or, if you're an Azure customer, you can add Bing Maps to your Azure subscription through the [Azure Marketplace](#).

# マイクロソフトのアカウントでログイン

---



# この画面(文章違っても)が出た人は一旦待機！！

The screenshot shows the Bing Maps Dev Center homepage. At the top, there's a navigation bar with links for 'My account', 'Data sources', 'Announcements', 'Contacts & Info', and a user account section '(macchinetta.xyz@gmail.com) Sign out'. Below the navigation, a large bold heading reads 'Announcement: No current service announcements.' followed by a subtext: 'To stay up to date on all the recent Bing Maps Platform releases and news, please visit the [Bing Maps Blog](#)'. A second bold heading 'Important reminder regarding Bing Maps service notifications:' is present, with a detailed explanatory text below it. At the bottom, there's a copyright notice and links to 'Privacy and Cookies', 'Legal', and 'Terms of Use'.

Bing maps | Dev Center

My account ▾ Data sources ▾ Announcements Contacts & Info (macchinetta.xyz@gmail.com) Sign out

## Announcement: No current service announcements.

To stay up to date on all the recent Bing Maps Platform releases and news, please visit the [Bing Maps Blog](#).

## Important reminder regarding Bing Maps service notifications:

To ensure your company receives important Bing Maps service notifications and announcements that may affect your service availability, please make sure you always keep your organization's email contacts up-to-date in the 'Account Details' section of the Bing Maps Dev Center. **We recommend having multiple email address contacts listed there and the use of a distribution group email address is a suggested best practice (i.e.: bingmaps@contoso.com).**

© 2019 - Microsoft Corporation. All rights reserved. [Privacy and Cookies](#) [Legal](#) [Terms of Use](#)

こちらの画面が出た人は「Yes」-> アカウント作成！

---

Bing maps | Dev Center

Do you want to use the account [REDACTED] for your new Bing Maps account? **Yes**

Or choose one of the following options:

[Sign in with another account](#) (This will sign you out of **hiro@hiros-dot.net**, so you can sign in with another account or create a new one.)

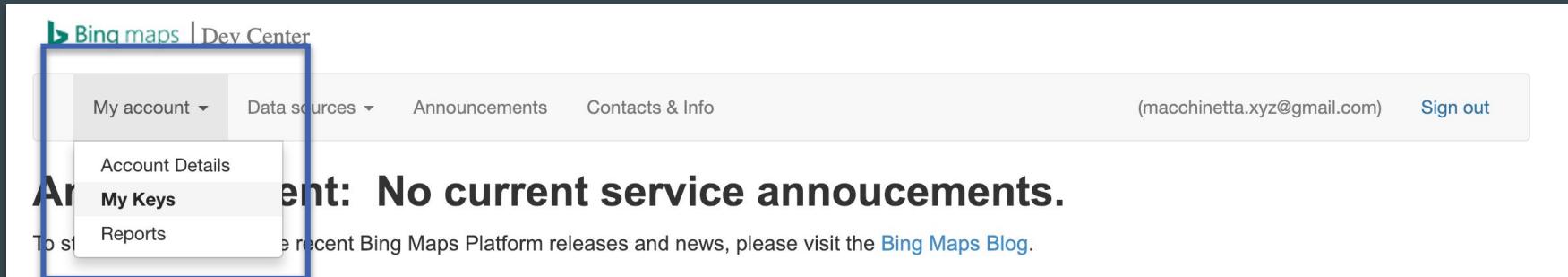
[Cancel, and think about it](#) (This will take you back to the Maps Dev Center home page.)

Yesをクリック



ここから全員共通

# 「My Keys」をクリック



The screenshot shows the Bing Maps Dev Center interface. At the top left is the 'Bing maps | Dev Center' logo. Below it is a navigation bar with 'My account', 'Data sources', 'Announcements', 'Contacts & Info', and user information '(macchinetta.xyz@gmail.com) Sign out'. A dropdown menu is open under 'My account' with options: 'Account Details', 'My Keys' (which is highlighted with a blue border), and 'Reports'. The main content area displays a message: 'Attention: No current service announcements.' It also encourages users to visit the 'Bing Maps Blog' for recent releases and news.

## Important reminder regarding Bing Maps service notifications:

To ensure your company receives important Bing Maps service notifications and announcements that may affect your service availability, please make sure you always keep your organization's email contacts up-to-date in the 'Account Details' section of the Bing Maps Dev Center. **We recommend having multiple email address contacts listed there and the use of a distribution group email address is a suggested best practice (i.e.: bingmaps@contoso.com).**

© 2019 - Microsoft Corporation. All rights reserved. [Privacy and Cookies](#) [Legal](#) [Terms of Use](#)

# アプリケーション名を入力

The screenshot shows the Bing Maps Dev Center interface. At the top, there's a navigation bar with links for 'My account', 'Data sources', 'Announcements', 'Contacts & Info', and a user account section '(macchinetta.xyz@gmail.com) Sign out'. Below the navigation, the title 'My keys' is displayed. A 'Create key' form is open, featuring fields for 'Application name' (with a red asterisk indicating it's required), 'Application URL', 'Key type' (with a dropdown menu showing 'Basic'), and 'Application type' (with a dropdown menu showing 'Dev/Test'). The 'Application name' field is highlighted with a blue border.

Bing maps | Dev Center

My account ▾ Data sources ▾ Announcements Contacts & Info (macchinetta.xyz@gmail.com) Sign out

## My keys

### Create key

Application name \*

Enter application name

Application URL

Enter application URL

Key type \*

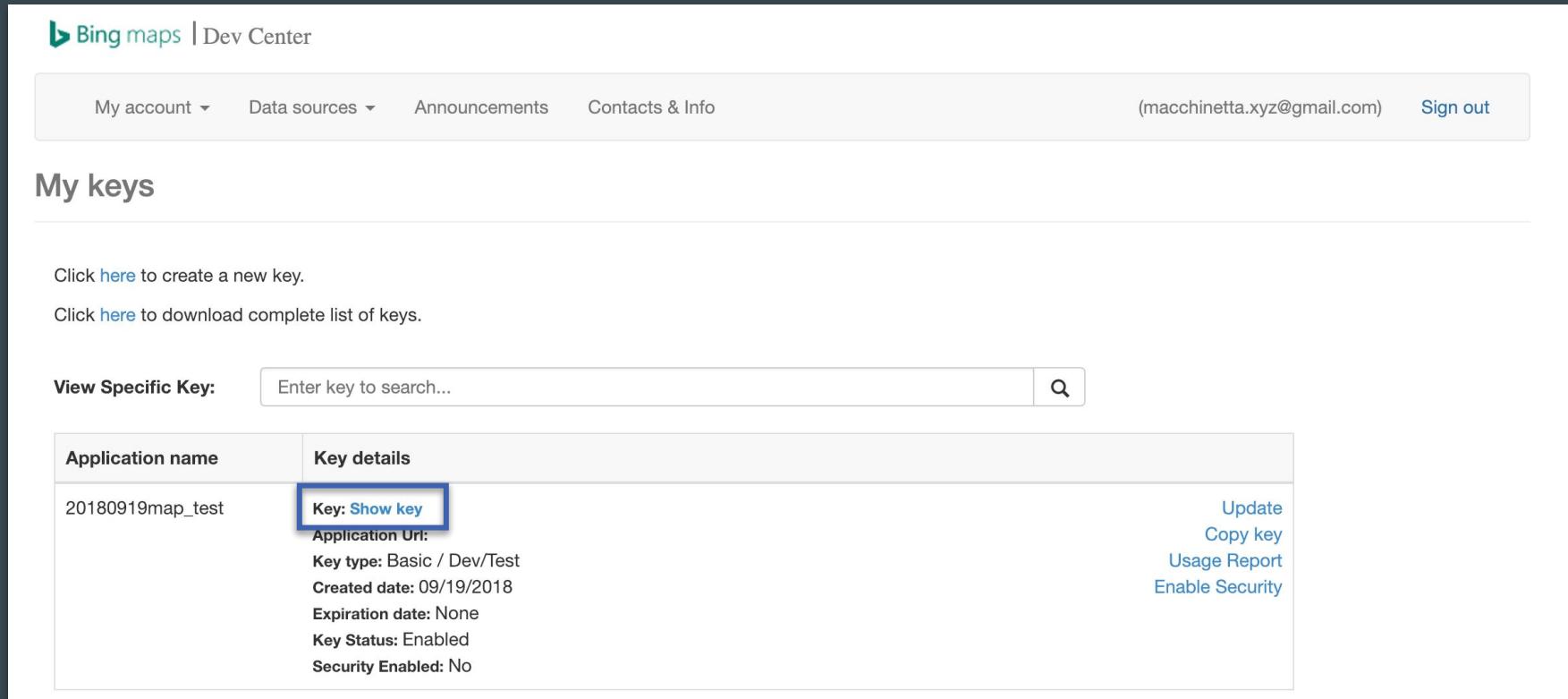
What's This

Basic

Application type \*

Dev/Test

# 「Show key」をクリックするとAPIキーが表示される



Bing maps | Dev Center

My account ▾ Data sources ▾ Announcements Contacts & Info (macchinetta.xyz@gmail.com) Sign out

## My keys

Click [here](#) to create a new key.  
Click [here](#) to download complete list of keys.

**View Specific Key:**

Application name	Key details	
20180919map_test	<p>Key: <a href="#">Show key</a> <span style="float: right;">Update Copy key Usage Report Enable Security</span></p> <p>Application Url: Key type: Basic / Dev/Test Created date: 09/19/2018 Expiration date: None Key Status: Enabled Security Enabled: No</p>	

# 現在地の取得

# 現在地の位置情報取得

---

- 種類
  - `getCurrentPosition` : 実行時に現在地を取得する.
  - `watchPosition` : 実行したら位置情報を取得し続ける.
  - (`watchPosition`) はバッテリー消費に注意！！
- オプション
  - `enableHighAccuracy` : 対応端末でGPSを使用
  - `maximumAge` : 指定時間以内であれば前回の値を用いる
  - `timeout` : タイムアウト時間を設定



時間の単位は  
「ミリ秒」

# 現在地の位置情報取得

---

```
<script>  
    // ()内には「成功時の関数」「失敗時の関数」「オプション」の順に記述する.  
  
    navigator.geolocation  
        .getCurrentPosition(showPosition, showError, option);  
  
</script>
```

# 現在地の位置情報取得

```
// 成功時の関数（位置情報を表示する）
// 関数については補足スライド参照

function showPosition(position) {
  const lat = position.coords.latitude;
  const lng = position.coords.longitude;
  console.log(lat, lng);
}
```

まずはconsoleに表示！！

# 現在地の位置情報取得

```
// 失敗時の関数（エラーを表示する）

function showError(error) {
    let e = '';
    if (error.code == 1) {
        e = '位置情報が許可されてません';
    } else if (error.code == 2) {
        e = '現在位置を特定できません';
    } else if (error.code == 3) {
        e = '位置情報を取得する前にタイムアウトになりました';
    }
    alert('error : ' + e);
}
```

エラーは  
3種類！

# 現在地の位置情報取得

---

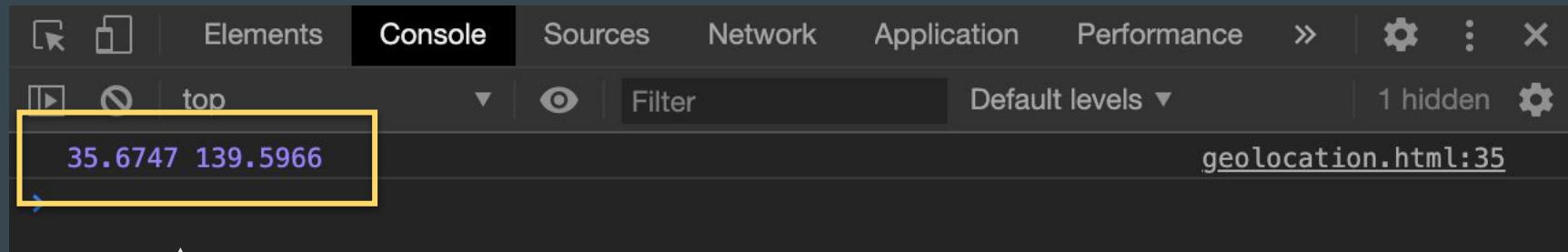
```
// オプション（スライド13枚めの内容）
// オブジェクトの形（補足スライド参照）

const option = {
  enableHighAccuracy: true,
  maximumAge: 20000,
  timeout: 1000000,
};

// この下に位置情報取得の処理（navigator.geolocation...）を記述しよう！
```

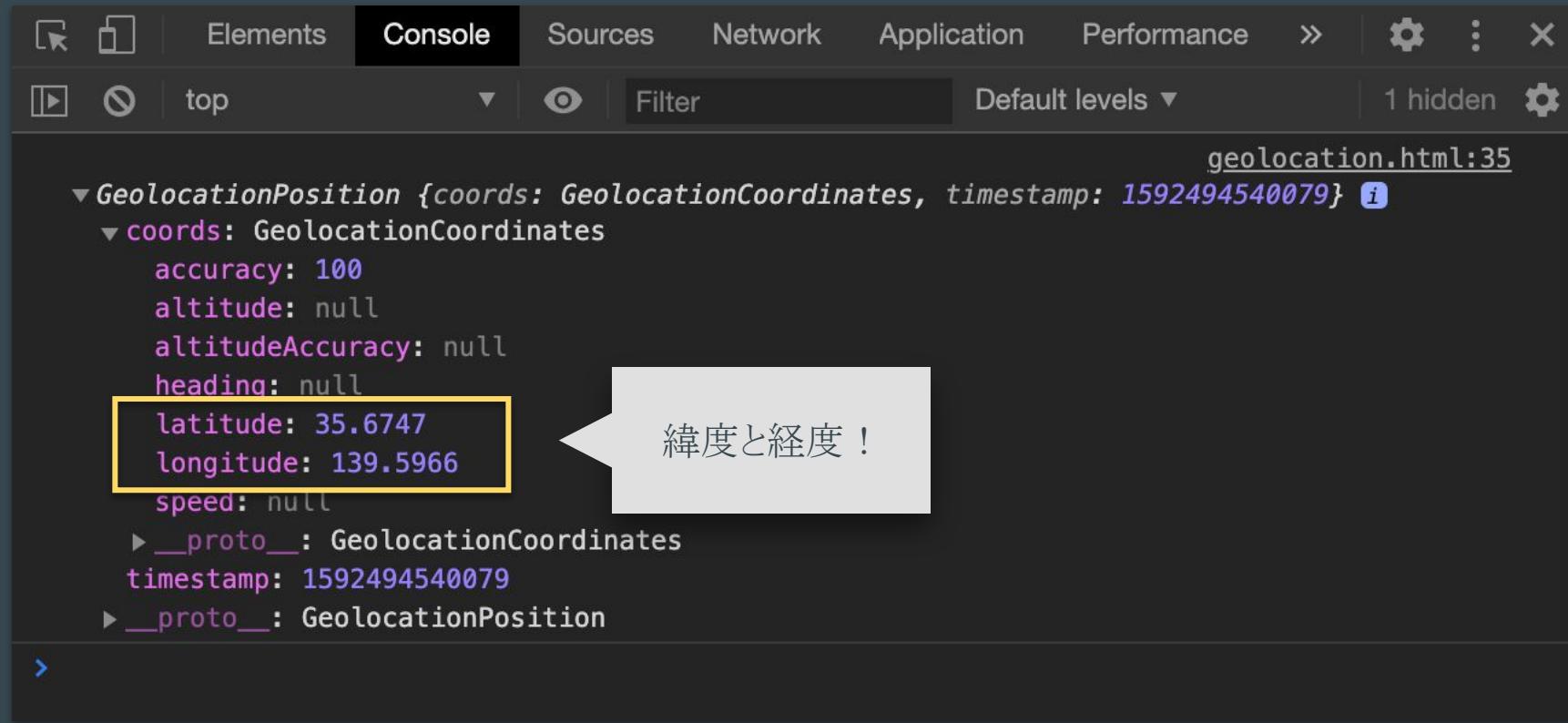
# 現在地を取得して表示しよう！

# 実行結果



緯度経度が表示されればOK！

# 位置情報のデータの詳細(positionの中身)



The screenshot shows the Chrome DevTools interface with the "Console" tab selected. A JSON object is displayed:

```
geolocation.html:35
▼ GeolocationPosition {coords: GeolocationCoordinates, timestamp: 1592494540079} ⓘ
  ▼ coords: GeolocationCoordinates
    accuracy: 100
    altitude: null
    altitudeAccuracy: null
    heading: null
    latitude: 35.6747
    longitude: 139.5966
    speed: null
  ► __proto__: GeolocationCoordinates
  timestamp: 1592494540079
  ► __proto__: GeolocationPosition
```

A yellow box highlights the "latitude" and "longitude" properties. An arrow points from this highlighted area to the text "緯度と経度！".

# 【補足】配列

```
// 複数の値に順番をつけてまとめて扱う方法。強い。  
// 順番を「index」と呼ぶ。「0」からスタート！
```

// 配列の定義

```
const arr = ['大吉', '中吉', '小吉', '凶', '大凶'];  
console.log(arr[0]); // 大吉
```



# 【補足】オブジェクト

// 配列と同様に複数の値を管理する方法  
// 配列の「index」に対して「key」「value」で管理.

// オブジェクトの定義  
const schedule = {  
 date: '2020/06/21',  
 day: 'Sun',  
 event: 'camp\_day02',  
};

// 値の取得  
console.log(schedule.event); // camp\_day02  
console.log(schedule['event']); // camp\_day02

2020/06/21	Sun	camp_day02
date	day	event

## 【補足】配列とオブジェクトの組み合わせ

```
// オブジェクトが複数あるときは配列と組み合わせる
// APIを使用すると結構出てくる形

// オブジェクトの配列
const schedules = [
  { date: '2020/06/19', day: 'Fri', event: 'meet_up' },
  { date: '2020/06/20', day: 'Sat', event: 'camp_day01' },
  { date: '2020/06/21', day: 'Sun', event: 'camp_day02' },
];

// 値の取得
console.log(schedules[0].date);          // 2020/06/19
console.log(schedules[1].event);          // camp_day01
```

# 【補足】関数(function)

```
// 関数とは記述した処理をまとめて名前をつけて使い回せるようにしたもの。  
// 一度処理を定義してしまえば、呼び出すだけで実行可能！
```

```
// 関数の定義  
function generate0to4() {  
    const randomNumber = Math.floor(Math.random() * 5);  
    console.log(randomNumber);  
    return randomNumber;  
}  
  
// 関数の実行  
const random = generate0to4();  
console.log(random); // 0から4のどれか
```

# マップを表示する！

# 必要なファイルを読み込む

```
<script src='https://www.bing.com/api/maps/mapcontrol?mkt=ja-jp&key=API_KEY'></script>
<script>
    // jsの処理
</script>
```

# 処理の流れ

---

- 位置情報の取得
  - navigator.geolocation...
  - 成功時の関数, 失敗時の関数, オプションは先程と同様.
  - (成功時の関数にマップ表示の処理を追加する)
- マップの表示
  - ブラウザに地図を表示する.
  - 中心座標の指定が必要.
  - (取得した緯度と経度を使用する)

# 準備

---

```
// マップ表示に使用する変数を用意しておく.  
// 失敗時の関数とオプションをコピーペーストしておく  
// (エラー時の警告とオプション内容は同様のため)  
  
let map;  
  
const option = {...};  
  
function showError(error) {...}
```

# 位置情報の取得

---

```
// getPosition()関数を定義する.  
// 外部ファイルの読み込みがあるため、読み込み終了後に実行するよう指定する.  
  
// 現在地を取得し、成功したら地図を表示する関数  
function getPosition() {  
    navigator.geolocation  
        .getCurrentPosition(mapsInit, showError, option);  
}  
  
// 他のファイルの読み込みが終わったら{}内を実行する  
window.onload = function () {  
    getPosition();  
}
```

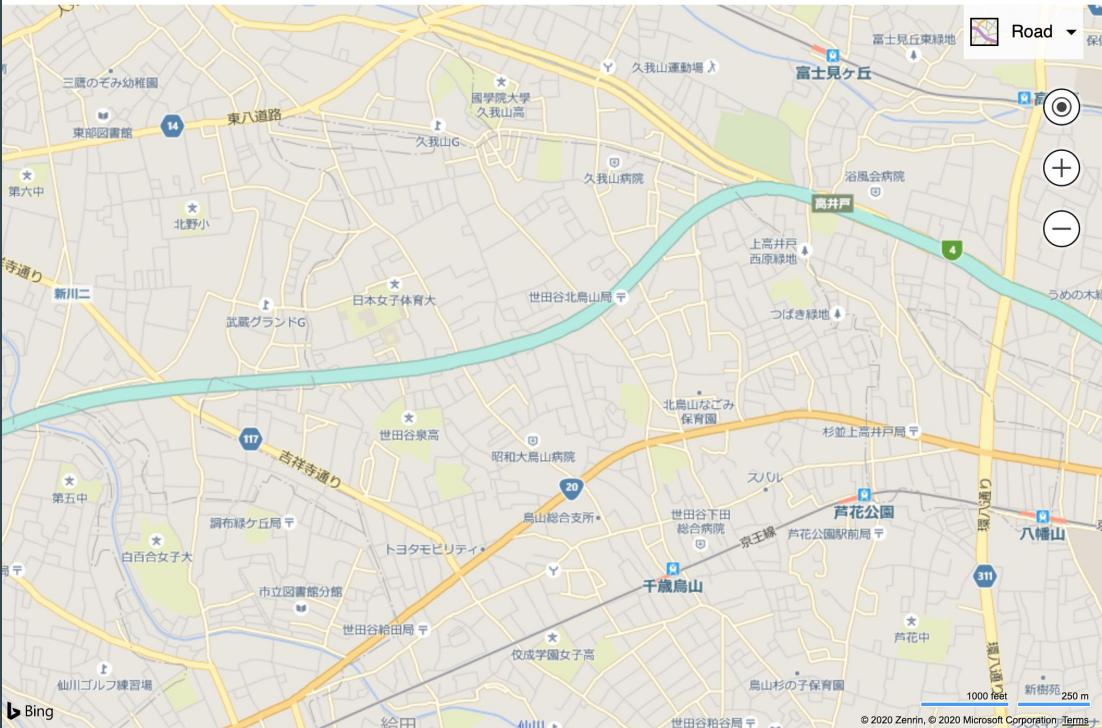
# 地図の表示

---

```
// 位置情報取得に成功したときの関数 (mapsInit) を定義する.  
// 「地図を表示したいHTML要素のid」「center」「zoom」を指定する.  
  
function mapsInit(position) {  
    const lat = position.coords.latitude;  
    const lng = position.coords.longitude;  
    map = new Microsoft.Maps.Map( '#map' , {  
        center: {  
            latitude: lat, longitude: lng  
        },  
        zoom: 15,  
    } );  
}
```

# 現在地が中心のmapが表示されればOK！

オリジナルマップ



ピンを立ててみる！

# 処理の流れ

---

- ピンの生成
  - ピンを立てる関数を定義する.
  - ピンの色などを指定する.
- マップ<sup>°</sup>上に設置
  - マップ表示後にピンを立てる関数を実行する.

# ピンの生成

---

```
// 緯度, 経度, 表示したいマップを入力するとピンを立てる関数を定義.  
// ピンの色と表示するよう設定.  
  
function pushPin(lat, lng, now) {  
    const location = new Microsoft.Maps.Location(lat, lng)  
    const pin = new Microsoft.Maps.Pushpin(location, {  
        color: 'navy',  
        visible: true,  
    });  
    now.entities.push(pin);  
};
```

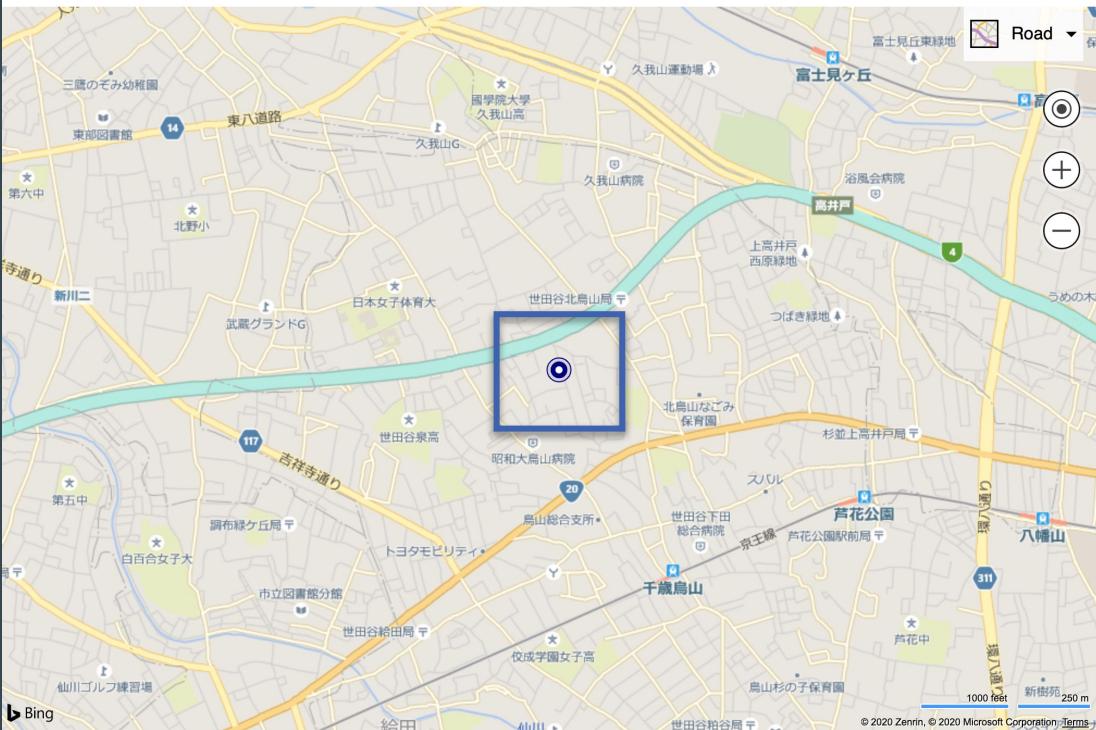
## マップ上に設置

---

```
// マップを表示する「mapsInit()」関数内でピンを設置する関数を実行.  
  
function mapsInit(position) {  
  const lat = position.coords.latitude;  
  const lng = position.coords.longitude;  
  // ...省略  
  pushPin(lat, lng, map);  
}
```

# 現在地にピンが立てばOK！

オリジナルマップ



infoboxを出してみる！

# 処理の流れ

---

- infoboxの生成
  - infoboxを生成する関数を定義する.
  - 表示内容を指定する.
- マップ上に設置
  - マップ表示後にinfoboxを生成する関数を実行する.

# infoboxの生成

---

```
// 緯度、経度、表示したいマップを入力するとinfoboxをつくる関数を定義.  
// 表示内容を設定する.  
  
function generateInfobox(lat, lng, now) {  
    const location = new Microsoft.Maps.Location(lat, lng)  
    const infobox = new Microsoft.Maps.Infobox(location, {  
        title: 'イマココ',  
        description: "I'm here!!!"  
    });  
    infobox.setMap(now);  
}
```

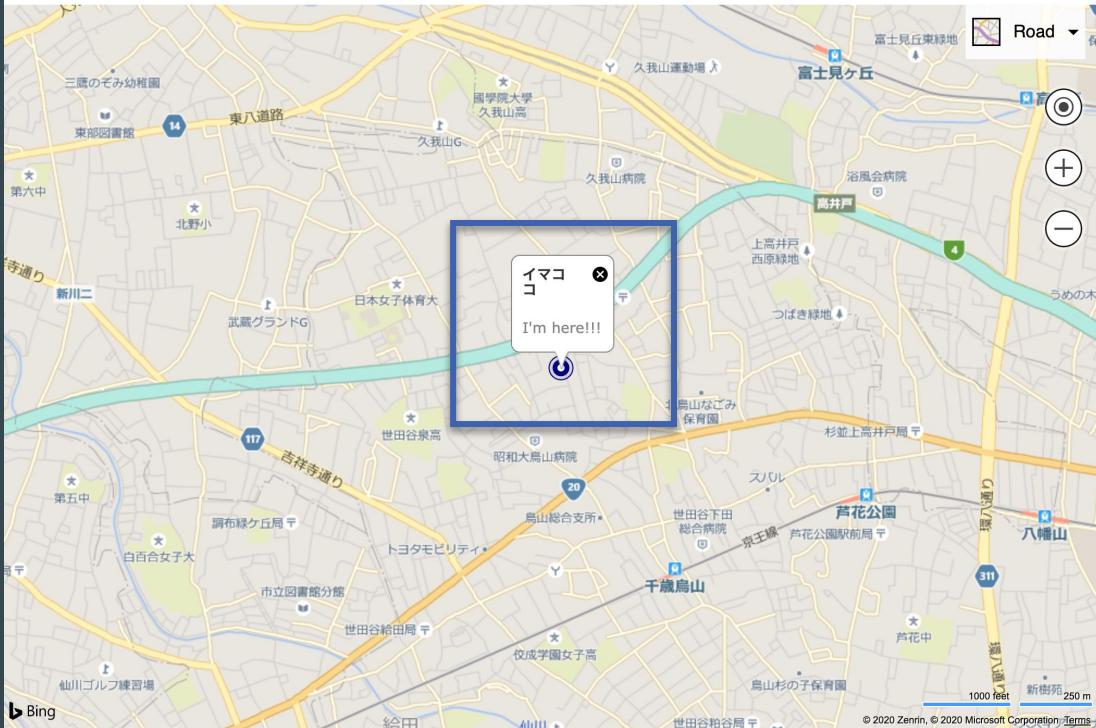
## マップ上に設置

---

```
// マップを表示する「mapsInit()」関数内でinfoboxを生成する関数を実行.  
  
function mapsInit(position) {  
    const lat = position.coords.latitude;  
    const lng = position.coords.longitude;  
    // ...省略  
    pushPin(lat, lng, map);  
    generateInfobox(lat, lng, map);  
}
```

# 現在地にinfoboxが出来ればOK！

オリジナルマップ



# 他にもできることたくさん．．！！

---

公式サンプル集(英語はGoogle翻訳！)

- <https://bingmapsv8samples.azurewebsites.net>

ジーズアカデミー学校長山崎先生のアプリ

- <https://mapapi.org>

# APIは複数のものを組み合わせよう！

---

例えば. . .

- レストラン検索(ぐるなびAPI)
  - レストランの位置情報(緯度, 経度)を取得できる.
  - マップ上にレストランの名前を入れたInfoboxを作成できる！
- 天気予報(「API 天気」でググるといくつかある)
  - 現在地を取得して、周辺の天気予報を取得できる.
  - 天気や気温に応じて、ぐるなびから周りのお店をレコメンド！

# 【おまけ】Google books API

# Google books API

---

- google books API
  - 今回は本を検索するAPIを使用しましょう！
  - ドキュメント:<https://developers.google.com/books/docs/overview>
- できること
  - google booksに登録されている本のjsonデータを取得できる！
  - タイトルや著者などで条件を指定できる！
  - (仕様をよく読みましょう！)
  - API keyが不要！！！

## 【参考】APIキーとは

---

- APIキー
  - APIにリクエストを送るためにはユニークな文字列が必要.
  - 設定を行ったアプリケーションのみ動作させるため.
  - (無断で使用されないように. . ! )
  - この文字列をAPIキーを呼び、APIの種類問わず必須の場合が多い.

# Google books API

---

- 準備
  - 下記のURLにリクエストしよう！
  - <https://www.googleapis.com/books/v1/volumes>
- パラメータの追加
  - 今回はタイトルに「`firebase`」が入っている本を指定して検索してみよう！
  - 上記URLの最後に「`?q=intitle:firebase`」を追加しよう！
- リクエストには`axios`ライブラリを使う！！
  - JavaScriptでhttp通信を行うためのライブラリ(読み込みが必要)
  - 最近のモダンなフレームワーク(`react`, `vue.js`など)でも利用されている

# Google books API

```
// APIにリクエストを送るときはaxios.get()を使用
axios.get(url)
  .then(function (response) {
    // リクエスト成功時の処理 (responseに結果が入っている)
    console.log(response);
  }).catch(function (error) {
    // リクエスト失敗時の処理 (errorにエラー内容が入っている)
    console.log(error);
  }).finally(function () {
    // 成功失敗に関わらず必ず実行
    console.log('done!');
  });
}
```

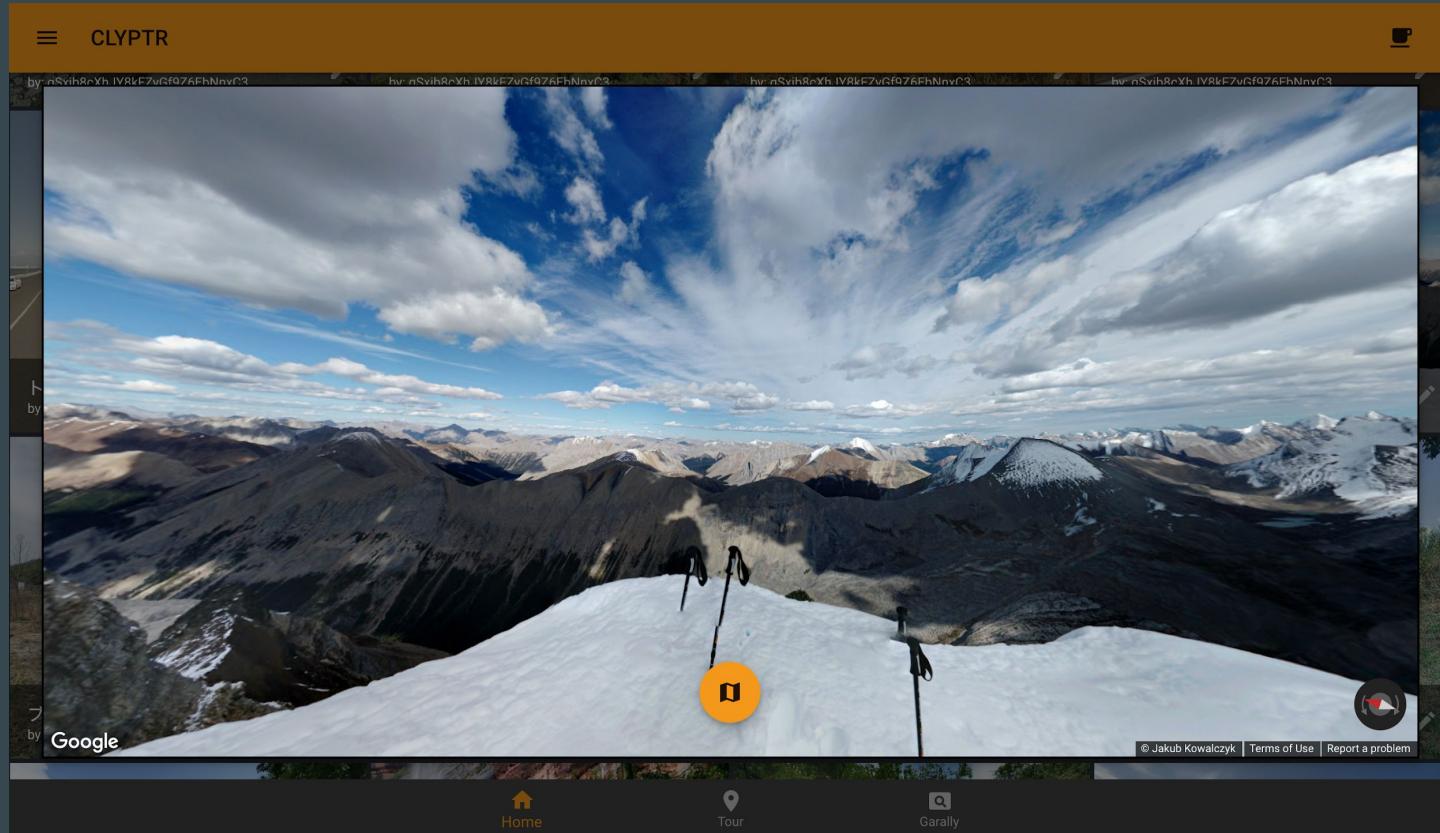
※ console.log()以外は決まっている！！

# 'response'の中身はこんな感じ！

```
booksapi.html:33
{
  data: {...}, status: 200, statusText: "", headers: {...}, config: {...}, ...
}
  ▶ config: {url: "https://www.googleapis.com/books/v1/volumes?q=intitle:j...
  ▶ data:
    ▶ items: Array(10)
      ▶ 0:
        ▶ accessInfo: {country: "JP", viewability: "PARTIAL", embeddable: ...
          etag: "vWwUyoZamvw"
          id: "KfBfAAAAQBAJ"
          kind: "books#volume"
        ▶ saleInfo: {country: "JP", saleability: "NOT_FOR_SALE", isEbook: ...
        ▶ searchInfo: {textSnippet: "電子版オリジナルの特別編集シリーズ『Web Des...
          selfLink: "https://www.googleapis.com/books/v1/volumes/KfBfAAAAQBAJ...
        ▶ volumeInfo:
          allowAnonLogging: true
        ▶ authors: ["Web Designing編集部"]
          canonicalVolumeLink: "https://books.google.com/books/about/Web...
          contentVersion: "preview-1.0.0"
          description: "電子版オリジナルの特別編集シリーズ『Web Designing Library』"
        ▶ imageLinks: {smallThumbnail: "http://books.google.com/books/com...
          infoLink: "http://books.google.co.jp/books?id=KfBfAAAAQBAJ&dq=...
          language: "ja"
          maturityRating: "NOT_MATURE"
        ▶ panelizationSummary: {containsEpubBubbles: false, containsImage...
          previewLink: "http://books.google.co.jp/books?id=KfBfAAAAQBAJ&...
          printType: "BOOK"
          publisher: "マイナビ出版"
        ▶ readingModes: {text: false, image: true}
          subtitle: ""
          title: "Web Designing Library #02 JavaScriptで学ぶプログラミング超...
        ▶ __proto__: Object
      ▶ 1: {kind: "books#volume", id: "Cb-jDwAAQBAJ", etag: "LMkdFx5NmDo", ...
      ▶ 2: {kind: "books#volume", id: "i2K3DwAAQBAJ", etag: "KZbZQ7cCrwY", ...
    
```

# 他のAPIの紹介

# ストリートビュー(Google)



# 画像認識API(Google)

---



```
{  
  "mid": "/m/07030",  
  "description": "Sushi",  
  "score": 0.84657246,  
  "topicality": 0.846572  
},
```

# 翻訳API(いろいろ)

---

プログラミング楽しい！



```
{  
    "translatedText":  
        "متعة البرمجة !",  
    "detectedSourceLanguage":  
        "ja"  
}
```

# レストラン検索API(ぐるなび)

検索条件を指定  
(例：緯度経度と電子マネー可)

latitude=43.0656865  
&longitude=141.3693016  
&e\_money=1



```
"rest": [  
  {  
    "@attributes": {  
      "order": 0  
    },  
    "id": "h308020",  
    "name": "ナガヤマ レスト",  
    "name_kana": "ナガヤマレスト",  
    "latitude": "43.065905",  
    "longitude": "141.364493",  
    "category": "和洋折衷喫茶",  
  },  
]
```

# API紹介サイト(<https://www.apibank.jp>)

API First, ランキング

ログイン | 会員登録 | ? Help API名を入力してください 

**APIbank**  
an **AOS** company

APIbankについて API一覧 APIを使う・作る APIblog

## SOBAプラットフォーム

APIを一元管理し、動作環境を提供するAPI開発プラットフォーム



76

# 課題

# 課題

---

- 「API」を使用したアプリケーションを実装！！
  - APIを組み合わせる！
  - もちろん他の技術を組み込んでもOK！
  - API自体を作ってもいいぞ..！
- アレンジの例
  - ゾンビと対戦するマップ
  - ワンタップで周辺のレストランを検索
  - セカイのどこかへストリートビュー
  - 政府の統計情報からマーケット分析
  - twitter bot, LINE bot, etc...

# プロダクトの考え方(一例)

---

まずこれを決める！

- 「誰が」「どうなるため」のもの？？
- (「自分が」もちろんOK！)

次はこれ！

- 上記を達成するにはどんな画面(見た目)が必要？？
- そのためにはどんな機能が必要？？
- 使ってみたい技術があるならどうやって使う？？

「コード」=「量」

2/14に発表会！！！  
最高のプロダクトを創ろう！！