

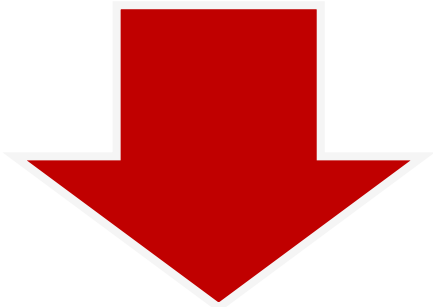
2019年第2回WXBCセミナー

APIセミナー

ビジネスソリューション事業部 システム部
土川 正彦

本日のコンセプト

ハレックスのAPIを使うと 気象情報が簡単に利用できることを体験してもらう



- ① サンプルプログラムの説明
- ② 自分でプログラミングして気象情報を操作

【持ち物】

●WiFi接続可能なノートPC

※WiFi接続できないとAPIで気象情報が取得できない・・・

※使用ブラウザ：GoogleChrome・Microsoft Edge
Internet Explorerは一部動作しない・・・

※文字コードは「UTF-8」 設定によっては文字化けします

●事前配布したアクセスキー

※セミナー用につき、二次配布や商用利用NG

※本日は、「WX2000-1xRr4fGm」を利用

本日は配布したアクセスキーは
使いません

【持ち物（任意）】

● Excel

※ グラフ表示、ピボットテーブル集計を利用

※ バージョンは2010以降を推奨

【資料ダウンロード】

<http://www.wxbc.jp/wp-content/uploads/tmp/api.zip>

ダウンロードは終わりましたか？

【その他】

- JavaScriptを利用してAPIで取得した
気象情報を操作して頂きます。
- 時間の関係から、気象に関わるご質問は
別途個別に打合せをさせていただきます。

※名刺を頂ければ、後日ご連絡させていただきます

本日より提供するAPI HalexMemory! (過去データ)

ハレックスでは未来予測に向けたサービスを提供している。

- ① 過去データ (HalexMemory!) ← 気象との因果関係 (傾向) を確認
- ② ピンポイントでの気象予測 (HalexDream!) ← 気象データを利用した未来予測
- ③ 気象用Webサイト + 気象条件によるアラートメール (HalexSmile!) ← すぐに気象を使いたい人向け

■ ハレックスのサービス

サービス		概要
気象	HalexDream! ※特許取得	<p>気象庁から大量かつ高頻度で発行されるデータをオンラインで高速リアルタイム処理することで、利用される方々によって扱い易いデータに編集し、緯度経度によって取り扱えるように1kmメッシュでデータを管理・処理するシステム</p> <p>なお、編集したデータは次の3つの手段によって提供</p> <ul style="list-style-type: none"> ① DreamAPI 緯度経度でパラメータ指定されジャストポイントの時系列データをレスポンスするサービス ② DreamFiles 緯度経度で指定されたジャストポイントの時系列データをファイルに格納して提供 ③ DreamAll 日本全国分の面展開したデータをファイルに格納して提供するサービス
	HalexMemory!	<p>過去の気象データをご利用頂くサービス</p> <p>日本全国1kmメッシュ×1時間間隔で管理したデータをCSVファイルによるオフライン提供やAPIによるサービスを実施</p> <p>※天気、温度、湿度、1時間降水量、風向風速・・・</p>
	HalexSmile!	<p>DreamAPIをフル活用した気象データ閲覧Webサイト</p> <p>地図上で指定した任意地点のデータ閲覧が可能で常時閲覧する地点は固定地点として登録が可能</p> <p>また条件設定することでアラートメール通知も可能</p>

第1回

本日はコレ

■ニーズの変化

【IoT・ビッグデータ・AIの活用によるパラダイムシフトが発生】

データのデジタル化、コンピュータ処理の高速化、データ分析技術の発達により課題やニーズの変化

	今まで	今後
課題	業務の見える化	未来予測
効果	業務効率化によるコスト削減	リスク回避・売上向上（プロフィット増大）

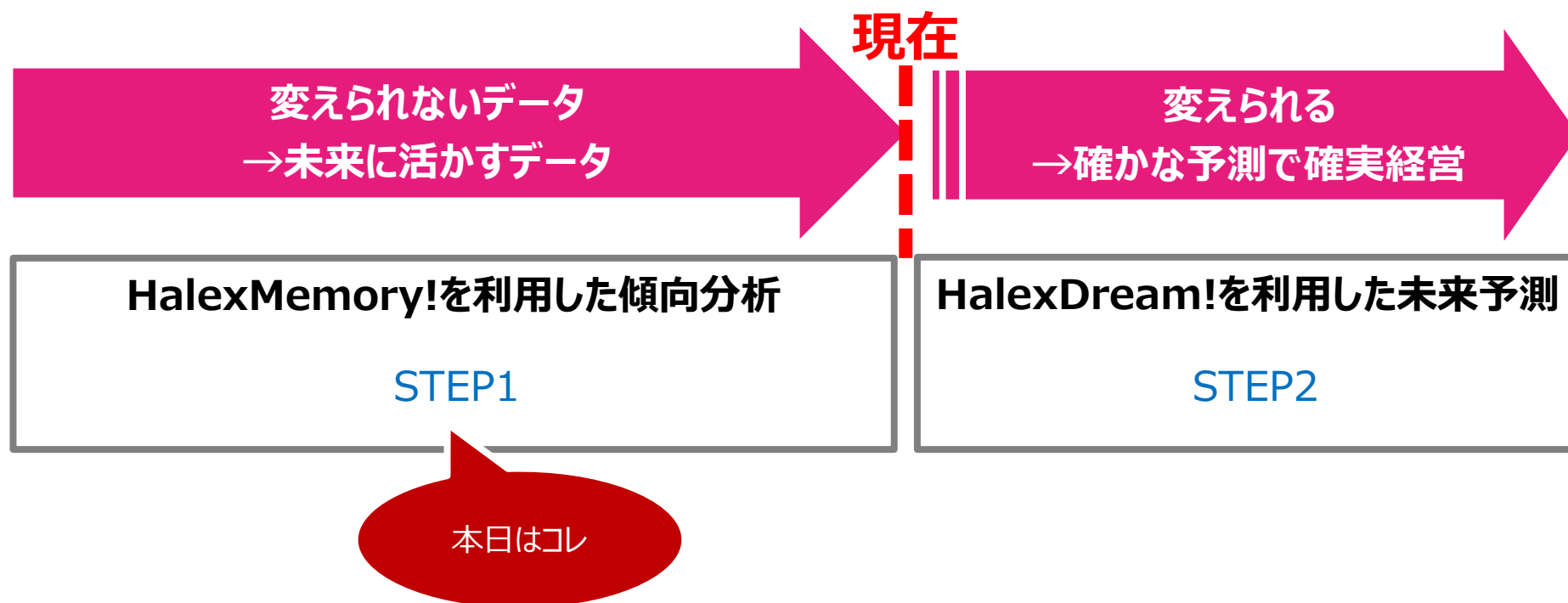
■気象情報利用への期待

気象は85%もの産業で影響を及ぼすと言われている。そのため、IoTやAI等の技術の進展により、農業、小売業、運輸業をはじめとする幅広い産業において気象データを利用した生産性の飛躍的向上が見込まれる。



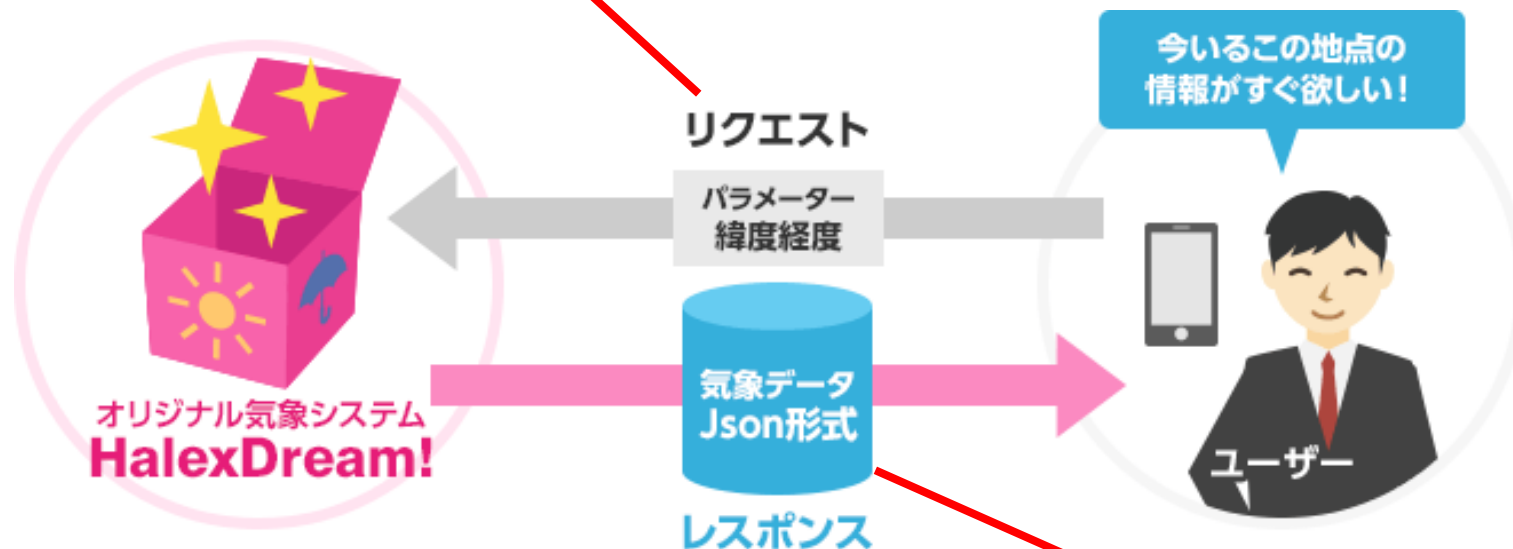
気象データの活用方法としては、「過去データによる傾向分析→未来予測」の手順が必要である。
気象データは他の情報と比べ「未来予測」に優れており、過去データ分析が完了すればすぐに実用化が期待できる。

■ 提供データごとの取り組み



ハレックスのAPIの特徴

http://demo.halex.co.jp/wimage/hpd?sid=jpwx-api&rem=wx&lat=[[緯度](#)]&lon=[[経度](#)]&key=[[アクセスキー](#)]



```
{
  "iniTime": "2019/07/11 16:00:00.000 JST",
  "param": {
    :
  },
  "sessionId": "F6DBF666691641A2E62C8BD0D5C9D014",
  "systemTime": "2019/07/11 15:34:55.813 JST",
  "weatherData": {
    "a": {
      "dy": {"mgtn": "0", "mgtx": "-1", "pb24": "60", "pb6_3": "50", "pb6_4": "60", "tn": "19", "tx": "23", "wd": "5", "ws": "2", "wt": "300", "wtt": "雨"},
      "h1-00": {"pr": "0.0", "te": "19", "wd": "2", "ws": "2", "wt": "100"},
      "h1-01": {"pr": "0.0", "te": "19", "wd": "2", "ws": "2", "wt": "100"},
      "h1-02": {"pr": "0.0", "te": "19", "wd": "2", "ws": "2", "wt": "100"},
      :
    }
  }
}
```

【URL】

[http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[データ取得開始\]&to=\[データ取得終了\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[データ取得開始]&to=[データ取得終了]&key=[アクセスキー])

※[]内の詳細は仕様書を参照

【提供内容】

アクセスキー	取得範囲	取得期間	提供期間
本日利用キー WX2000-1xRr4fGm	全国	2017.6～ 2018.9	セミナー時
各自配布したキー	大阪駅を中心に 半径100km		2019.11末


JSONとはJavaScript Object Notationの略で、XMLなどと同様のテキストベースのデータフォーマットです。

その名前の由来の通りJSONはJavaScriptのオブジェクト表記構文のサブセットとなっており、XMLと比べると簡潔に構造化されたデータを記述することができるため、記述が容易で人間が理解しやすいデータフォーマットです。


※今回のセミナーはJSONPを利用して記述しています


ご提供資料の説明


編集用プログラムの動作を確認するには、メニューの「カスタマイズページを開く」をクリック


 index.css


ファイル名拡張子が表示されている場合


 index.html


 MemoryAPI 簡易仕様書.pdf


 Sample3-1.html


 Sample3-2.html


 Sample3-3.html


 Sample3-4.html


 Sample4-1.html


 Sample4-2.html


 Sample4-3.html


 Sample4-4.html

 カスタマイズ用Sample4-1.html

 カスタマイズ用Sample4-2.html

 カスタマイズ用Sample4-3.html

 カスタマイズ用Sample4-4.html

 集計用シート.xlsx

セミナーのメニュー画面

API仕様書 ※重要

サンプルプログラム

編集用プログラム

例題④-5で利用

メニュー画面



セミナーのメニュー画面を開きましょう！！

- index.css
- index.html**
- MemoryAPI仕様書.pdf
- Sample3-1.html
- Sample3-2.html
- Sample3-3.html
- Sample3-4.html
- Sample4-1.html
- Sample4-2.html
- Sample4-3.html
- Sample4-4.html
- カスタマイズ用Sample4-1.html
- カスタマイズ用Sample4-2.html
- カスタマイズ用Sample4-3.html
- カスタマイズ用Sample4-4.html
- 集計用シート.xlsx

セミナーのメニュー画面

API仕様書 ※重要

サンプルプログラム

編集用プログラム

例題④-5で利用

メニュー画面



コレを起動
ブラウザはGoogleChromeまたは
MicrosoftEdge！！

GoogleChromeまたはMicrosoftEdge以外のブラウザで開いてしまう場合、
GoogleChromeまたはMicrosoftEdgeを起動後、「index.html」をドラッグ＆ドロップ！！



ハレックスAPIセミナー（2019年9月25日）

① HalexMemory!について

<https://www.halex.co.jp/service/index/memory.html>

② JSONについて

JSONとはJavaScript Object Notationの略で、軽量なデータ記述言語の1つです。

JSONはJavaScript専用のデータ形式では決してなく、様々なソフトウェアやプログラミング言語間におけるデータの受け渡しに使えるよう設計されています。

今回はWebブラウザを使用して情報を取得するためJSONPを使用します。

JSONPとは引数にJSONをとったコールバック関数のことで、主にWeb画面内のJavaScriptでJSONを使いたい場合に使用されます。

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）



ハレックスAPIセミナー（2019年9月25日）

① HalexMemory!について

弊社のサイト API説明 ※割愛

<https://www.halex.co.jp/service/index/memory.html>

② JSONについて

APIレスポンスであるJSONについて
※割愛

JSONとはJavaScript Object Notationの略で、軽量なデータ記述言語の1つです。

JSONはJavaScript専用のデータ形式では決してなく、様々なソフトウェアやプログラミング言語間におけるデータの受け渡しに使えるよう設計されています。

今回はWebブラウザを使用して情報を取得するためJSONPを使用します。

JSONPとは引数にJSONをとったコールバック関数のことで、主にWeb画面内のJavaScriptでJSONを使いたい場合に使用されます。

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）

API仕様書の説明



ハレックスAPIセミナー（2019年9月25日）

① HalexMemory!について

<https://www.halex.co.jp/service/index/memory.html>

② JSONについて

JSONとはJavaScript Object Notationの略で、軽量なデータ記述言語の1つです。

JSONはJavaScript専用のデータ形式では決してなく、様々なソフトウェアやプログラミング言語間におけるデータの受け渡しに使えるよう設計されています。

今回はWebブラウザを使用して情報を取得します。

JSONPとは引数にJSONをとったコードで、画面内のJavaScriptでJSONを使いたい場合に使用されます。

クリック

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）



HALEX MemoryAPI 過去データ 仕様書



株式会社 ハレックス
2019/8/28 ver1.0

1. 概要

- ・1kmメッシュ(3次メッシュ)で管理された気象情報を、緯度、経度、取得期間、粒度を指定することにより、該当メッシュの過去の気象情報をJSON形式または、JSONP形式でレスポンスします。

アクセスキーを設定

2. リクエスト

【リクエストURL】(例)

http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=34.702485&lon=135.495951&gran=5&from=20170801&to=20170831&key=xxxxxx

←リクエストURL

【リクエストパラメータ】

No	パラメータ	内容	記述および説明
1	sid	サービスID	JSON : analysis-service JSONP : analysis-p-service
2	lat	緯度	取得対象位置の緯度を指定(世界測地系・10進法)
3	lon	経度	取得対象位置の経度を指定(世界測地系・10進法)
4	rem	要素指定	all(全要素)固定
5	gran	データ取得粒度	取得したい時間粒度を指定 年 : 1, 月 : 2, 旬 : 3, 半旬 : 4, 日 : 5, 時 : 6
6	from	データ取得開始	取得したい期間の始めを指定します。※西暦・日本時間、以下同じ 年 : YYYY (例 : 2017年 2017) 月 : YYYYMM (例 : 2017年1月 201701) 旬 : YYYYMM[a,b,c] (例 : 上旬 YYYYMMa) 半旬 : YYYYMM[a1,a2,b1,b2,c1,c2] (例 : 中旬後半 YYYYMMb2) 日 : YYYYMMDD (例 : 2017年1月4日 20170104) 時 : YYYYMMDDhh (例 : 2017年1月4日9時 2017010409)
7	to	データ取得終了	取得したい期間の終わりを指定します。
8	func	JSONPコールバック関数名	コールバック関数名を指定します。

←リクエストパラメータ

【データ取得粒度毎の指定可能期間】

No	データ取得粒度	指定可能期間
1	年	5年
2	月	12か月
3	旬	18旬 (半年)
4	半旬	18半旬 (3か月)
5	日	62日
6	時	72時間

←データ取得粒度毎の指定可能な期間

提供範囲はP12参照

3. レスポンス

【レスポンスイメージ】

・特別データ指定

```
{
  "Data": {
    "2017010100": {
      "pr": "0.0", "te": "-2.1", "wd": "191.4", "ws": "2.2", "wt": "100",
      "2017010101": {
        "pr": "0.0", "te": "-2.7", "wd": "198.0", "ws": "2.0", "wt": "100",
        .
        .
        "2017010323": {
          "pr": "0.0", "te": "-0.8", "wd": "160.1", "ws": "2.8", "wt": "100"
        }
      }
    }
  },
  "param": {
    "from": "2017010100", "gran": "6", "lat": "35.55555", "lon": "138.95555", "rem": "all", "sid": "analysis-service", "to": "2017010323",
    "sessionId": "xxxxxx", "sid": "analysis-service", "systemTime": "2019/08/27 13:07:58.187 JST"
  }
}
```

データ部

情報部

←レスポンスイメージ

※年データ指定は割愛

【データ部変数説明】

No	変数	要素	単位	フォーマット	格納例
1	wt	天気	code	数値、3桁整数、コード番号(※1)	晴: 100等
2	te	気温	℃	数値、小数点第一位、0.1刻み、マイナスあり	-10℃: -10.0、1.5℃: 1.5
3	teAve	平均気温	℃	同上	同上
4	teMax	最高気温期間極値	℃	同上	同上
5	teMin	最低気温期間極値	℃	同上	同上
6	teMaxA	最高気温期間平均値	℃	同上	同上
7	teMinA	最低気温期間平均値	℃	同上	同上
8	pr	降水量合算値	mm	数値、小数点第一位、0.1刻み、0.0以上	無: 0.0、0.5mm: 0.5
9	prMaxH	時間降水量最大値	mm	同上	同上
10	prMaxD	日降水量最大値	mm	同上	同上
11	wd	風向	度	数値、小数点第一位、0.1刻み、0.0以上(※2)	静穏: 0.0、北の風: 360.0
12	ws	風速	m/s	数値、小数点第一位、0.1刻み、0.0以上	静穏: 0.0、風速1m/s: 1.0

←データ部の説明

(※1)主な天気コード

天気コード	予報内容
100	晴
200	曇
300	雨
400	雪
500	みぞれ

(※2)0.1度刻みの風向を8方位に変換する場合、下記テーブル参照

windDirection値	風向
0.0	静穏
337.6~360.0	北
0.1~ 22.5	
22.6~ 67.5	北東
67.6~112.5	東
112.6~157.5	南東
157.6~202.5	南
202.6~247.5	南西
247.6~292.5	西
292.6~337.5	北西

レスポンスデータ

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）

③-1 MemoryAPIを使ってみよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

ajaxを使って、JSONPでデータを取得します。

[サンプルページはこちら](#)

クリック

③-2 MemoryAPIから1日の最高気温を取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を表示します。

[サンプルページはこちら](#)

③-3 MemoryAPIから1日の最高気温を3日分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を日数分繰り返し処理（ループ）で表示します。

[サンプルページはこちら](#)

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

③-3を参考に取得期間と要素を変更します。

[サンプルページはこちら](#)

データ取得粒度は「日」です。



MemoryAPIからJSONで気象情報を取得

[トップ](#) > MemoryAPIからJSONで気象情報を取得

MemoryAPIからJSONで気象情報を取得

JavaScriptからMemoryAPIを使って、JSONを画面に表示します。

Ajaxを使って、JSONPでデータを取得します。

Ajaxとは「Asynchronous JavaScript + XML」（JavaScriptとXMLを使った非同期通信）の略です。

今回は本画面からAjaxを使って、JSONPの取得を行います。

緯度経度の設定

緯度 : 経度 : 期間from : 期間to :

※ 「/（スラッシュ）」なしの「YYYYMMDD」で入力してください。

実行

クリック

この緯度経度の
気象情報を取得
※初期値は大阪駅

株式会社ハレックス 問い合わせ先：03-5420-4311

緯度経度の設定

緯度 : 34.702485

経度 : 135.495951

期間from : 20170801 ~ 20170804

※ 「/ (スラッシュ)」なしの「YYYYMMDD」で入力してください。

実行

「Data」属性の下
に指定した日付
分の情報が格納

```
{
  "Data": {
    "20170801": {
      "pr": "5.0", "prMaxH": "2.0", "teAve": "30.3", "teMax": "34.7", "teMin": "27.5", "wd": "16.5", "ws": "6.5", "wt": "311"
    },
    "20170802": {
      "pr": "0.0", "prMaxH": "0.0", "teAve": "29.6", "teMax": "34.3", "teMin": "26.0", "wd": "112.3", "ws": "5.8", "wt": "101"
    },
    "20170803": {
      "pr": "0.0", "prMaxH": "0.0", "teAve": "29.4", "teMax": "34.3", "teMin": "26.2", "wd": "90.0", "ws": "5.9", "wt": "200"
    },
    "20170804": {
      "pr": "1.0", "prMaxH": "0.5", "teAve": "30.4", "teMax": "34.8", "teMin": "27.1", "wd": "50.8", "ws": "6.3", "wt": "214"
    }
  },
  "param": {
    "_": "1568204581617",
    "callback": "doJson",
    "from": "20170801",
    "gran": "5",
    "lat": "34.702485",
    "lon": "135.495951",
    "rem": "all",
    "sid": "analysis-p-service",
    "to": "20170804"
  },
  "sessionId": "hnojzym2eg4vgylwd2kyjna",
  "sid": "analysis-p-service",
  "systemTime": "2019/09/11 21:27:42.810 JST"
}
```

データ部

情報部

■ Dataの値

年(1) : yyyy
 月(2) : yyyy-mm
 旬(3) : yyyy-mm[a-c]
 半旬(4) : yyyy-mm[a-c][1-2]
 日(5) : yyyy-mm-dd
 時(6) : yyyy-mm-dd-hh

【データ部変数説明】

No	変数	要素	単位	ラベル
1	wt	天気	code	変数
2	te	気温	℃	変数
3	teAve	平均気温	℃	変数
4	teMax	最高気温期間極値	℃	変数
5	teMin	最低気温期間極値	℃	変数
6	teMaxA	最高気温期間平均値	℃	変数
7	teMinA	最低気温期間平均値	℃	変数
8	pr	降水量合算値	mm	変数
9	prMaxH	時間降水量最大値	mm	変数
10	prMaxD	日降水量最大値	mm	変数
11	wd	風向	度	変数
12	ws	風速	m/s	変数

JSONの構造

【基本形】

```
{"項目1":"値"}
```

キー

バリュー

【応用形①（2重構造）】

```
{"項目1":{"項目2":"値"}}
```

キー

バリュー

キー

バリュー

【応用形②（2重構造×複）】

```
{"項目1":{"項目2":"値"},  
{"項目2":"値"}}
```

キー

バリュー

キー

バリュー

サンプルプログラムの説明

③-2

③-2 MemoryAPIから1日の最高気温を取得しよう

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）

③-1 MemoryAPIを使ってJSONデータを取得しよう

JavaScriptからMemoryAPIを使って、JSONを画面に表示します。

ajaxを使って、JSONPでデータを取得します。

[サンプルページはこちら](#)

③-2 MemoryAPIから1日の最高気温を取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を表示します。

[サンプルページはこちら](#)

クリック

③-3 MemoryAPIから1日の最高気温を3日分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を日数分繰り返し処理（ループ）で表示します。

[サンプルページはこちら](#)

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

③-3を参考に取得期間と要素を変更します。

[サンプルページはこちら](#)



MemoryAPIから1日の最高気温を取得

[トップ](#) > MemoryAPIから1日の最高気温を取得

MemoryAPIから1日の最高気温を取得

JavaScriptからMemoryAPIを使って、JSONを取得します。
取得したJSONから指定した1日の気温を表示します。
最高気温は「teMax」となります。

緯度経度の設定

緯度 : 経度 :

指定日 : ※ 「/ (スラッシュ)」なしの「YYYYMMDD」で入力してください。

実行

この緯度経度、日付
の気象情報を取得
※初期値は大阪駅

株式会社ハレックス 問い合わせ先 : 03-5420-4311

③-2 プログラムの表示

セミナーのメニュー画面を開きましょう！！

index.css

index.html

MemoryAPI 簡易仕様書.pdf

Sample3-1.html

Sample3-2.html

Sample3-3.html

Sample3-4.html

Sample4-1.html

Sample4-2.html

Sample4-3.html

Sample4-4.html

カスタマイズ用Sample4-1.html

カスタマイズ用Sample4-2.html

カスタマイズ用Sample4-3.html

カスタマイズ用Sample4-4.html

集計用シート.xlsx

セミナーのメニュー画面

API仕様書 ※重要

サンプルプログラム

編集用プログラム

例題④-5で利用

コレをノートパッドで開く
※テキストエディタならOK

メニュー画面




```

Sample3-2.html - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<!doctype html>
<html lang="ja">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-Style-Type" content="text/css" />
<meta http-equiv="Content-Script-Type" content="text/javascript" />
<meta name="viewport" content="width=device-width" />
<title>1日の日まとめデータを取得</title>
<link rel="stylesheet" href="index.css" type="text/css" />
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/j
<script type="text/javascript" language="javascript">
//実行ボタン押下時の処理
function onClick() {
//APIキーの設定 ※配布されたAPIキーを入力してください
var api_key = "WX2000-1xRr4fGm";
//画面上の緯度経度を取得
var lat = document.forms.id_form1.lat.value;
var lon = document.forms.id_form1.lon.value;
//画面上の指定日を取得
var day_from = document.forms.id_form1.day_from.value;
//変数初期化
var out_txt = "";
//気象データの取得処理
$.ajax({
url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon=" + lon + "&gran=5&from=" + day_from + "&to=" + day_from + "&key=" + api_key,
dataType: "jsonp", //気象データの形式はJSONP
jsonpCallback: "doJson" //JSONPのコールバック名はデフォルトdoJson
})
//正常に気象データを取得できた場合の処理
.done(function(json) {
// [teMax] : 最高気温
out_txt = day_from + "の最高気温: " + json["Data"][day_from]["teMax"] + "℃";
})
//気象データの取得に失敗した場合の処理
.fail(function(json) {
alert("気象データの取得に失敗しました。");
out_txt = "失敗";
})
//最終処理
.always(function(json) {
var target = document.getElementById("output");
target.innerHTML = out_txt;
});
});
</script>
</head>

```

セミナー用のライセンスキー
セミナー終了後は「api_key」に
配布したライセンスキーを設定

③-2 MemoryAPIから1日の最高気温を取得しよう

```

1 <!doctype html>
2 <html lang="ja">
3
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <meta http-equiv="Content-Style-Type" content="text/css" />
7   <meta http-equiv="Content-Script-Type" content="text/javascript" />
8   <meta name="viewport" content="width=device-width" />
9   <title>1日の日まとめデータを取得</title>
10  <link rel="stylesheet" href="index.css" type="text/css" />
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
12  <script type="text/javascript" language="javascript">
13    //実行ボタン押下時の処理
14    function onClick() {
15      //APIキーの設定 ※配布されたAPIキーを入力してください
16      var api_key = "WX2000-1xRr4fGm";
17      //画面の緯度経度を取得
18      var lat = document.forms.id_form1.lat.value;
19      var lon = document.forms.id_form1.lon.value;
20      //画面の指定日を取得
21      var day_from = document.forms.id_form1.day_from.value;
22      //変数初期化
23      var out_txt = "";
24      //気象データの取得処理
25      $.ajax({
26        url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon=" + lon,
27        dataType: "jsonp", //気象データの形式はJSONP
28        jsonpCallback: "doJson" //JSONPのコールバック名はデフォルトdoJson
29      });
30      //正常に気象データを取得できた場合の処理
31      .done(function(json) {
32        //「teMax」：最高気温
33        out_txt = day_from + "の最高気温：" + json["Data"][day_from]["teMax"] + "℃";
34      });
35      //気象データの取得に失敗した場合の処理
36      .fail(function(json) {
37        alert("気象データの取得に失敗しました。");
38        out_txt = "失敗";
39      });
40      //最終処理
41      .always(function(json) {
42        var target = document.getElementById("output");
43        target.innerHTML = out_txt;
44      });
45    }
46  </script>
47 </head>
48

```

Ajaxとは、JavaScriptを
効率よく利用するためのフレームワーク

JavaScript

←要素を指定
["Data"]：(固定)
[day_from]：日付←画面より取得
["teMax"]：最高気温

←変数「output」を表示

③-2 MemoryAPIから1日の最高気温を取得しよう

```

49 <body>
50 <div id="outer">
51   <div id="header">
52     <a href="https://www.halex.co.jp/" target="_blank">
53       
54     </a>
55     <h1>MemoryAPIから1日の最高気温を取得</h1>
56   </div>
57   <ul class="breadcrumb">
58     <li>
59       <a href="index.html" itemprop="url">
60         <span itemprop="title">トップ</span>
61       </a>
62     </li>
63     <li>
64       <span itemprop="title">MemoryAPIから1日の最高気温を取得</span>
65     </li>
66   </ul>
67   <div id="main">
68     <div class="content">
69       <h2>MemoryAPIから1日の最高気温を取得</h2>
70       <p>
71         JavaScriptからMemoryAPIを使って、JSONを取得します。<br>
72         取得したJSONから指定した1日の気温を表示します。<br>
73         最高気温は「teMax」となります。
74       </p>
75       <h3>緯度経度の設定</h3>
76       <div>
77         <form name="form1" id="id_form1" action="#">
78           <label>
79             緯度:<input type="text" name="lat" id="lat" size="15" maxlength="15" value="34.702485" />
80             経度:<input type="text" name="lon" id="lon" size="15" maxlength="15" value="135.49595" />
81           </label>
82           <label>
83             指定日:<input type="text" name="day_from" id="day_from" size="15" maxlength="15" value="" />
84             ※「/ (スラッシュ)」なしの「YYYYMMDD」で入力してください。
85           </label>
86           <input type="button" value="実行" onclick="onButtonClick();" />
87         </form>
88         <div id="output"></div>
89       </div>
90     </div>
91   </div>
92 </div>
93 <div id="footer">
94   <div class="copy">株式会社ハレックス&nbsp;&nbsp;問い合わせ先: 03-5420-4311</div>
95 </div>
96 </div>
97 </body>
98 </html>
99

```

サンプルプログラムの説明

③-3

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）

③-1 MemoryAPIを使ってJSONデータを取得しよう

JavaScriptからMemoryAPIを使って、JSONを画面に表示します。

ajaxを使って、JSONPでデータを取得します。

[サンプルページはこちら](#)

③-2 MemoryAPIから1日の最高気温を取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を表示します。

[サンプルページはこちら](#)

③-3 MemoryAPIから1日の最高気温を3日分取得しよう

JavaScriptからMemoryAPIを使

取得したJSONの最高気温を3日間分（3日間分）で表示します。

[サンプルページはこちら](#)

クリック

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

③-3を参考に取得期間と要素を変更します。

[サンプルページはこちら](#)



MemoryAPIから1日の最高気温を3日分取得

[トップ](#) > MemoryAPIから1日の最高気温を3日分取得

MemoryAPIから1日の最高気温を3日分取得

JavaScriptからMemoryAPIを使って、JSONを取得します。
取得したJSONから指定した3日分の最高気温をループ処理を使用し表示します。

緯度経度の設定

緯度 : 経度 :

期間from : 期間to :

※「/（スラッシュ）」は「YYYYMMDD」で入力してください。

















この緯度経度、日付
の気象情報を取得
※初期値は大阪駅

実行

株式会社ハレックス 問い合わせ先 : 03-5420-4311

③-3 プログラムの表示

セミナーのメニュー画面を開きましょう！！

-  index.css
-  index.html
-  MemoryAPI 簡易仕様書.pdf
-  Sample3-1.html
-  Sample3-2.html
-  **Sample3-3.html**
-  Sample3-4.html
-  Sample4-1.html
-  Sample4-2.html
-  Sample4-3.html
-  Sample4-4.html
-  カスタマイズ用Sample4-1.html
-  カスタマイズ用Sample4-2.html
-  カスタマイズ用Sample4-3.html
-  カスタマイズ用Sample4-4.html
-  集計用シート.xlsx

セミナーのメニュー画面

API仕様書 ※重要

サンプルプログラム

編集用プログラム

例題④-5で利用

これをノートパッドで開く
※テキストエディタならOK

メニュー画面



③-3 MemoryAPIから1日の最高気温を3日分取得しよう

```

1 <!doctype html>
2 <html lang="ja">
3
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <meta http-equiv="Content-Style-Type" content="text/css" />
7   <meta http-equiv="Content-Script-Type" content="text/javascript" />
8   <meta name="viewport" content="width=device-width" />
9   <title>MemoryAPIから1日の最高気温を3日分取得</title>
10  <link rel="stylesheet" href="index.css" type="text/css" />
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">/script>
12  <script type="text/javascript" language="javascript">
13    //実行ボタン押下時の処理
14    function onClick() {
15      //APIキーの設定 ※配布されたAPIキーを入力してください
16      var api_key = "WX2000-1xRr4fGm";
17      //経度を取得
18      var lat = document.forms.id_form1.lat.value;
19      //経度を取得
20      var lon = document.forms.id_form1.lon.value;
21      //画面上の期間fromを取得
22      var day_from = document.forms.id_form1.day_from.value;
23      //画面上の期間toを取得
24      var day_to = document.forms.id_form1.day_to.value;
25      //変数初期化
26      var out_txt = "";
27      //ヘッダー部初期化
28      var out_th = "";
29      //データ部初期化
30      var out_td = "";
31      //気象データの取得処理
32      $.ajax({
33        url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon=" + lon,
34        dataType: "jsonp", //気象データの形式はJSONP
35        jsonpCallback: "doJson" //JSONPのコールバック名はデフォルトdoJson
36      });

```

Ajaxとは、JavaScriptを
効率よく利用するためのフレームワーク

←気象情報を取得したい地点の緯度経度を設定

←気象情報を取得したい日付を設定

←APIリクエストを設定

JavaScript

つづく

③-3 MemoryAPIから1日の最高気温を3日分取得しよう

```
31 //気象データの取得処理
32 $.ajax({
33   url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon="
34   dataType: "jsonp", //気象データの形式はJSONP
35   jsonpCallback: "doJson" //JSONPのコールバック名はデフォルト doJson
36 })
37 //正常に気象データを取得できた場合の処理
38 .done(function(json) {
39   //key(日付)の数だけループする
40   for (let key of Object.keys(json["Data"])) {
41     //ヘッダー部の設定
42     out_th = out_th + "<th>" + key + "</th>";
43     //「teMax」: 最高気温
44     //データ部の設定
45     out_td = out_td + "<td>" + json["Data"][key]["teMax"] + "</td>";
46   }
47   //表示するために整形
48   out_txt = "<table><tr>" + out_th + "</tr><tr>" + out_td + "</tr></table>";
49 }
50 //気象データの取得に失敗した場合の処理
51 .fail(function(json) {
52   alert("気象データの取得に失敗しました。");
53   out_txt = "失敗";
54 })
55 //最終処理
56 .always(function(json) {
57   var target = document.getElementById("output");
58   target.innerHTML = out_txt;
59 });
60
61 </script>
62 </head>
63
64
65 <body>
66   <div id="outer">
67     <div id="header">
68       <a href="https://www.halex.co.jp/" target="_blank">
69         
70       </a>
71       <h1>MemoryAPIから1日の最高気温を3日分取得</h1>
72     </div>
```

ループ処理

←日付分だけループ ※["Data"]配下の日付毎に処理

←表示用に編集 ※日付

←表示用に編集 ※最高気温

←表示用に編集 ※htmlのtableタグ

要素を指定

["Data"] : (固定)

[key] : 日付←APIで取得した日付

["teMax"] : 最高気温

←変数「output」を表示

つづく

JavaScript

③-3 MemoryAPIから1日の最高気温を3日分取得しよう

```

69 
70 </a>
71 <h1>MemoryAPIから1日の最高気温を3日分取得</h1>
72 </div>
73 <ul class="breadcrumb">
74 <li>
75 <a href="index.html" itemprop="url">
76 <span itemprop="title">トップ</span>
77 </a>
78 </li>
79 <li>
80 <span itemprop="title">MemoryAPIから1日の最高気温を3日分取得</span>
81 </li>
82 </ul>
83 <div id="main">
84 <div class="content">
85 <h2>MemoryAPIから1日の最高気温を3日分取得</h2>
86 <p>
87 JavaScriptからMemoryAPIを使って、JSONを取得します。<br>
88 取得したJSONから指定した3日分の最高気温をループ処理を使用し表示します。<br>
89 </p>
90 <h3>緯度経度の設定</h3>
91 <div>
92 <form name="form1" id="id_form1" action="#">
93 <label>
94 緯度:<input type="text" name="lat" id="lat" size="15" maxlength="15" value="34.70248
95 経度:<input type="text" name="lon" id="lon" size="15" maxlength="15" value="135.4956
96 </label>
97 <label>
98 期間from:<input type="text" name="day_from" id="day_from" size="15" maxlength="15"
99 期間to:<input type="text" name="day_to" id="day_to" size="15" maxlength="15" value="
100 ※「/ (スラッシュ)」なしの「YYYYMMDD」で入力してください。
101 </label>
102 <input type="button" value="実行" onclick="onButtonClick();" />
103 </form>
104 <div id="output"></div>
105 </div>
106 </div>
107 </div>
108 <div id="footer">
109 <div class="copy">株式会社ハレックス&nbsp;&nbsp;問い合わせ先：03-5420-4311</div>
110 </div>
111 </div>
112 </body>
113 <
114 </html> [EOF]

```

サンプルプログラムの説明

③-4

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

③ ハレックスのAPIを使ってみよう

仕様書：[MemoryAPI 簡易仕様書.pdf](#)

リクエストURL：[http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=\[緯度\]&lon=\[経度\]&gran=\[データ取得粒度\]&from=\[期間FROM\]&to=\[期間TO\]&key=\[アクセスキー\]](http://demo.halex.co.jp/wpast/hpd?sid=analysis-service&rem=all&lat=[緯度]&lon=[経度]&gran=[データ取得粒度]&from=[期間FROM]&to=[期間TO]&key=[アクセスキー])

[緯度][経度]：取得したい地点の緯度経度を設定してください。

[データ取得粒度]：取得したい時間粒度を設定してください。（日：5、時：6）

[期間FROM、TO]：取得したい期間の始めと終わりを設定してください。

[アクセスキー]：別途案内されているアクセスキーを設定してください。

（http,httpsでのリクエストが可能）

③-1 MemoryAPIを使ってJSONデータを取得しよう

JavaScriptからMemoryAPIを使って、JSONを画面に表示します。

ajaxを使って、JSONPでデータを取得します。

[サンプルページはこちら](#)

③-2 MemoryAPIから1日の最高気温を取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を表示します。

[サンプルページはこちら](#)

③-3 MemoryAPIから1日の最高気温を3日分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONの最高気温を日数分繰り返し処理（ループ）で表示します。

[サンプルページはこちら](#)

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

JavaScriptからMemoryAPIを使って、JSONを取得します。

③-3を参考に取得期間と要素を設定します。

[サンプルページはこちら](#)

クリック



MemoryAPIから1日の最低気温を1週間分取得

[トップ](#) > MemoryAPIから1日の最低気温を1週間分取得

MemoryAPIから1日の最低気温を1週間分取得

JavaScriptからMemoryAPIを使って、JSONを取得します。

取得したJSONから指定した1週間分の最低気温をループ処理を使用し表示します。

緯度経度の設定

緯度 : 経度 :

期間from : 期間to :

実行














この緯度経度、日付
の気象情報を取得
※初期値は大阪駅

※「/」（スラッシュ）を「-」（ハイフン）「YYYYMMDD」で入力してください。

株式会社ハレックス 問い合わせ先 : 03-5420-4311

③-4 プログラムの表示

セミナーのメニュー画面を開きましょう！！

-  index.css
-  index.html
-  MemoryAPI 簡易仕様書.pdf
-  Sample3-1.html
-  Sample3-2.html
-  Sample3-3.html
-  **Sample3-4.html**
-  Sample4-1.html
-  Sample4-2.html
-  Sample4-3.html
-  Sample4-4.html
-  カスタマイズ用Sample4-1.html
-  カスタマイズ用Sample4-2.html
-  カスタマイズ用Sample4-3.html
-  カスタマイズ用Sample4-4.html
-  集計用シート.xlsx

セミナーのメニュー画面

API仕様書 ※重要

サンプルプログラム

編集用プログラム

例題④-5で利用

コレをノートパッドで開く
※テキストエディタならOK

メニュー画面



**ただ、③-4を実現するには
③-3を 1 か所修正するだけ！！**

修正箇所を考えてみよう。

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

以下は③-3のサンプルプログラム

```
1 <!doctype html>
2 <html lang="ja">
3
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <meta http-equiv="Content-Style-Type" content="text/css" />
7   <meta http-equiv="Content-Script-Type" content="text/javascript" />
8   <meta name="viewport" content="width=device-width" />
9   <title>MemoryAPIから1日の最高気温を3日分取得</title>
10  <link rel="stylesheet" href="index.css" type="text/css" />
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
12  <script type="text/javascript" language="javascript">
13    //実行ボタン押下時の処理
14    function onClick() {
15      //APIキーの設定 ※配布されたAPIキーを入力してください
16      var api_key = "WX2000-1xRr4fGm";
17      //経度を取得
18      var lat = document.forms.id_form1.lat.value;
19      //経度を取得
20      var lon = document.forms.id_form1.lon.value;
21      //画面上の期間fromを取得
22      var day_from = document.forms.id_form1.day_from.value;
23      //画面上の期間toを取得
24      var day_to = document.forms.id_form1.day_to.value;
25      //変数初期化
26      var out_txt = "";
27      //ヘッダー部初期化
28      var out_th = "";
29      //データ部初期化
30      var out_td = "";
31      //気象データの取得処理
32      $.ajax({
33        url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon=" + lon + "&day_from=" + day_from + "&day_to=" + day_to,
34        dataType: "jsonp", //気象データの形式はJSONP
35        jsonpCallback: "doJson" //JSONPのコールバック名はデフォルトdoJson
36      })
```

Ajaxとは、JavaScriptを
効率よく利用するためのフレームワーク

←気象情報を取得したい地点の緯度経度を設定

←気象情報を取得したい日付を設定

←APIリクエストを設定

JavaScript

つづく

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

つづく

JavaScript

```
31 //気象データの取得処理
32 $.ajax({
33   url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon="
34   dataType: "jsonp", //気象データの形式はJSONP
35   jsonpCallback: "doJson" //JSONPのコールバック名はデフォルト doJson
36 })
37 //正常に気象データを取得できた場合の処理
38 .done(function(json) {
39   //key(日付)の数だけループする
40   for (let key of Object.keys(json["Data"])) { ←日付分だけループ ※["Data"]配下の日付毎に処理
41     //ヘッダー部の設定
42     out_th = out_th + "<th>" + key + "</th>"; ←表示用に編集 ※日付
43     //「teMax」：最高気温
44     //データ部の設定
45     out_td = out_td + "<td>" + json["Data"][key]["teMax"] + "</td>"; ←表示用に編集 ※最高気温
46   }
47   //表示するために整形
48   out_txt = "<table><tr>" + out_th + "</tr><tr>" + out_td + "</tr>" + "</table>"; ←表示用に編集 ※htmlのtableタグ
49   //データの取得に失敗した場合の処理
50   (function(json) {
51     alert("気象データの取得に失敗しました。");
52     out_txt = "失敗";
53   })(json);
54 }
55 //最終処理
56 .always(function(json) {
57   var target = document.getElementById("output");
58   target.innerHTML = out_txt;
59 });
60 }
61 </script>
62 </head>
63
64
65 <body>
66   <div id="outer">
67     <div id="header">
68       <a href="https://www.halex.co.jp/" target="_blank">
69         
70       </a>
71       <h1>MemoryAPIから1日の最高気温を3日分取得</h1>
72     </div>
```

ループ処理

日付処理はループ処理
なので修正不要

要素を指定
["Data"]：(固定)
[key]：日付←APIで取得
["teMax"]：最高気温

修正はココだけ
["teMax"]最高気温
→["teMin"]最低気温

←変数「output」を表示

③-4 MemoryAPIから1日の最低気温を1週間分取得しよう

```

69 
70 </a>
71 <h1>MemoryAPIから1日の最高気温を3日分取得</h1>
72 </div>
73 <ul class="breadcrumb">
74 <li>
75 <a href="index.html" itemprop="url">
76 <span itemprop="title">トップ</span>
77 </a>
78 </li>
79 <li>
80 <span itemprop="title">MemoryAPIから1日の最高気温を3日分取得</span>
81 </li>
82 </ul>
83 <div id="main">
84 <div class="content">
85 <h2>MemoryAPIから1日の最高気温を3日分取得</h2>
86 <p>
87 JavaScriptからMemoryAPIを使って、JSONを取得します。<br>
88 取得したJSONから指定した3日分の最高気温をループ処理を使用し表示します。<br>
89 </p>
90 <h3>緯度経度の設定</h3>
91 <div>
92 <form name="form1" id="id_form1" action="#">
93 <label>
94 緯度:<input type="text" name="lat" id="lat" size="15" maxlength="15" value="34.70248
95 経度:<input type="text" name="lon" id="lon" size="15" maxlength="15" value="135.4956
96 </label>
97 <label>
98 期間from:<input type="text" name="day_from" id="day_from" size="15" maxlength="15"
99 期間to:<input type="text" name="day_to" id="day_to" size="15" maxlength="15" value="
100 ※「/ (スラッシュ)」なしの「YYYYMMDD」で入力してください。
101 </label>
102 <input type="button" value="実行" onclick="onButtonClick();" />
103 </form>
104 <div id="output"></div>
105 </div>
106 </div>
107 </div>
108 <div id="footer">
109 <div class="copy">株式会社ハレックス&nbsp;&nbsp;問い合わせ先: 03-5420-4311</div>
110 </div>
111 </div>
112 </body>
113 <
114 </html> [EOF]

```

例題

資料の説明

編集用プログラムの動作を確認するには、メニューの「カスタマイズページを開く」をクリック

















- index.css
- index.html
- MemoryAPI 簡易仕様書.pdf
- Sample3-1.html
- Sample3-2.html
- Sample3-3.html
- Sample3-4.html
- Sample4-1.html
- Sample4-2.html
- Sample4-3.html
- Sample4-4.html
- カスタマイズ用Sample4-1.html
- カスタマイズ用Sample4-2.html
- カスタマイズ用Sample4-3.html
- カスタマイズ用Sample4-4.html
- 集計用シート.xlsx

④-1～④-4の
サンプルソース

④-1～④-4の
例題で編集するファイル
※内容は③-3のサンプルファイルと同じ

まずは例題④-1をやってみよう

編集したプログラムの動作を確認するには、メニューの「カスタマイズページを開く」をクリック

 index.css index.html MemoryAPI 簡易仕様 Sample3-1.html Sample3-2.html Sample3-3.html Sample3-4.html Sample4-1.html Sample4-2.html Sample4-3.html Sample4-4.html カスタマイズ用Sample4-1.html カスタマイズ用Sample4-2.html カスタマイズ用Sample4-3.html カスタマイズ用Sample4-4.html 集計用シート.xlsx②編集した内容の動作確認は
「カスタマイズページを開く」

①このファイルを編集

④ 例題

③-3をベースとしたカスタマイズ用ファイルをご用意しています。
編集はカスタマイズ用をご利用ください。

③-3 2017年7月の大阪駅の降水量合算値を取得しよう

2017年7月の大阪駅の降水量合算値を取得してください。

期間：2017年7月1日～2017年7月31日

緯度経度：緯度: 34.702485、経度: 135.495951

[カスタマイズページを開く](#)

[回答例はこちら](#)

③-2 2018年7月の大阪駅の降水量合算値を取得しよう

2018年7月の大阪駅の降水量合算値を取得してください。

期間：2018年7月1日～2018年7月31日

緯度経度：緯度: 34.702485、経度: 135.495951

[カスタマイズページを開く](#)

[回答例はこちら](#)

日本豪雨の雨量の多さが確認できます。

④-3 2017年の夏（7月～9月）に35度以上の日数を計算しよう

2017年の夏（7月～9月）の大阪駅の最高気温が35度以上となった日数を計算してください。

期間：2017年7月1日～2017年9月30日

緯度経度：緯度: 34.702485、経度: 135.495951

[カスタマイズページを開く](#)

[回答例はこちら](#)

④-4 2018年の夏（7月～9月）に35度以上の日数を計算しよう

2018年の夏（7月～9月）の大阪駅の最高気温が35度以上となった日数を計算してください。

期間：2018年7月1日～2018年9月30日

緯度経度：緯度: 34.702485、経度: 135.495951

[カスタマイズページを開く](#)

[回答例はこちら](#)

※④-3と④-4を比較すると201

同じ要領で④-4までやってみよう！！

例題④-5

④-5 気象情報とアイスの販売個数データを突合し、関係を調査しよう

次の期間の気象情報と、アイスの販売個数データの関係进行调查してください。

期間：2018年7月1日～2018年9月31日

アイスの販売個数データは[集計用シート.xlsx]を参照して下さい。

⑤ 注意事項

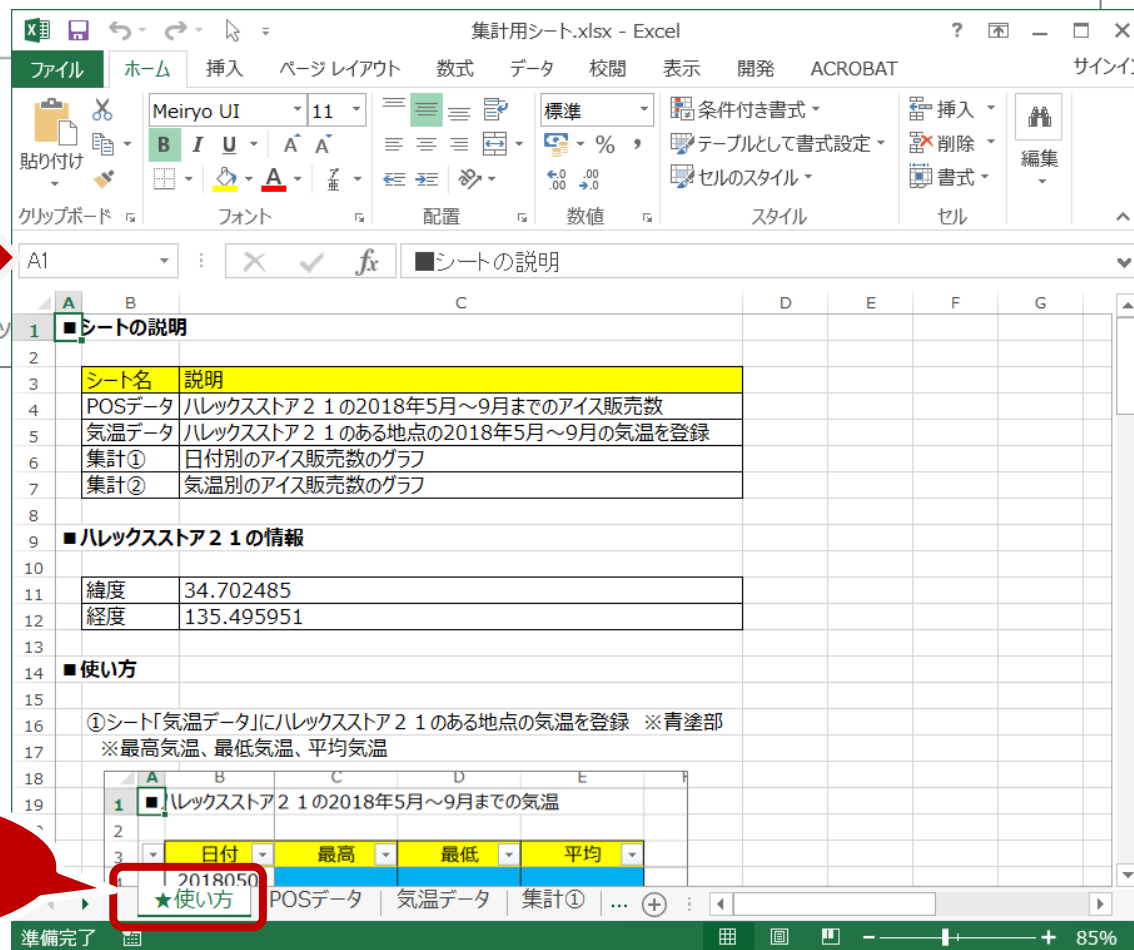
⑤-1 ライセンスに関して

今回配布したAPIキーはセミナー用とさせていただきます。
二次配布や商用利用は行わないでください。

①クリックし、
このファイルを開く

株式会社ハレックス

②シート「★使い方」
に沿って実施



集計用シート.xlsx - Excel

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 開発 ACROBAT サインイン

貼り付け クリップボード フォント 配置 数値 スタイル セル

A1 : X ✓ fx ■シートの説明

シート名	説明
POSデータ	ハレックスストア 2 1 の2018年5月～9月までのアイス販売数
気温データ	ハレックスストア 2 1 のある地点の2018年5月～9月の気温を登録
集計①	日付別のアイス販売数のグラフ
集計②	気温別のアイス販売数のグラフ

■ハレックスストア 2 1 の情報

緯度	34.702485
経度	135.495951

■使い方

①シート「気温データ」にハレックスストア 2 1 のある地点の気温を登録 ※青塗部
※最高気温、最低気温、平均気温

日付	最高	最低	平均
20180501			

★使い方 POSデータ 気温データ 集計① ... (+)

準備完了 85%

例題④-1解説

※例題④-2も同様

④-1 2017年7月の大阪駅の降水量合算値を取得しよう

```

1 <!doctype html>
2 <html lang="ja">
3
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <meta http-equiv="Content-Style-Type" content="text/css" />
7   <meta http-equiv="Content-Script-Type" content="text/javascript" />
8   <meta name="viewport" content="width=device-width" />
9   <title>2017年7月の大阪駅の降水量合算値を取得</title>
10  <link rel="stylesheet" href="index.css" type="text/css" />
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
12  <script type="text/javascript" language="javascript">
13    //実行ボタン押下時の処理
14    function onClick() {
15      //APIキーの設定 ※配布されたAPIキーを入力してください
16      var api_key = "WX2000-1xRr4fGm";
17      //緯度経度を取得
18      var lat = document.forms.id_form1.lat.value;
19      var lon = document.forms.id_form1.lon.value;
20      var day_from = document.forms.id_form1.day_from.value;
21      var day_to = document.forms.id_form1.day_to.value;
22      var out_txt = "";
23      //気象データの取得処理
24      $.ajax({
25        url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon=" + lon + "&day_from=" + day_from + "&day_to=" + day_to,
26        dataType: "jsonp", //気象データの形式はJSONP
27        jsonpCallback: "doJson", //JSONPのコールバック名はデフォルトdoJson
28      });
29      //正常に気象データを取得できた場合の処理
30      .done(function(json) {
31        //keyの数だけループする
32        for (let key of Object.keys(json["Data"])) {
33          out_txt = out_txt + "<tr><th>" + key + "</th>";
34          //「pr」: 降水量合算値
35          out_txt = out_txt + "<td>" + json["Data"][key]["pr"] + "</td></tr>";
36        }
37        out_txt = "<table class='mode1'>" + out_txt + "</table>";
38      });
39      //気象データの取得に失敗した場合の処理
40      .fail(function(json) {
41        alert("気象データの取得に失敗しました。");
42        out_txt = "失敗";
43      });
44      //最終処理
45      .always(function(json) {
46        var target = document.getElementById("output");
47        target.innerHTML = out_txt;
48      });
49    }
50  </script>
51 </head>

```

Ajaxとは、JavaScriptを
効率よく利用するためのフレームワーク

←ボタン押下イベントで実行

←アクセスキーを設定

←気象情報を取得したい地点の緯度経度、日付を設定

←APIリクエストを設定

ループ処理

←日付分だけループ ※["Data"]配下の日付毎に処理

←表示用に編集 ※日付

←表示用に編集 ※降水量合算値

←表示用に編集 ※htmlのtableタグ

要素を指定

["Data"] : (固定)

[key] : 日付←APIで取得した日付

["PR"] : 降水量合算値

←変数「output」を表示

JavaScript

④-1 2017年7月の大阪駅の降水量合算値を取得しよう

```

54 <body>
55   <div id="outer">
56     <div id="header">
57       <a href="https://www.halex.co.jp/" target="_blank">
58         
59       </a>
60       <h1>2017年7月の大阪駅の降水量合算値を取得</h1>
61     </div>
62     <ul class="breadcrumb">
63       <li>
64         <a href="index.html" itemprop="url">
65           <span itemprop="title">トップ</span>
66         </a>
67       </li>
68       <li>
69         <span itemprop="title">2017年7月の大阪駅の降水量合算値を取得</span>
70       </li>
71     </ul>
72     <div id="main">
73       <div class="content">
74         <h2>2017年7月の大阪駅の降水量合算値を取得</h2>
75         <p>
76           2017年7月の大阪駅の降水量合算値を取得して表示します。<br>
77           期間：2017年7月1日～2017年7月31日<br>
78           緯度経度：緯度：34.702485、経度：135.495951
79         </p>
80         <h3>緯度経度の設定</h3>
81         <div>
82           <form name="form1" id="id_form1" action="#">
83             <label>
84               緯度：<input type="text" name="lat" id="lat" size="15" maxlength="15" value="34.702485" />
85               経度：<input type="text" name="lon" id="lon" size="15" maxlength="15" value="135.495951" />
86             </label>
87             <label>
88               期間from：<input type="text" name="day_from" id="day_from" size="15" maxlength="15" value="201
89               期間to：<input type="text" name="day_to" id="day_to" size="15" maxlength="15" value="20170731"
90               ※「/（スラッシュ）」なしの「YYYYMMDD」で入力してください。
91             </label>
92             <input type="button" value="実行" onclick="onButtonClick();" />
93           </form>
94           <div id="output"></div>
95         </div>
96       </div>
97     </div>
98     <div id="footer">
99       <div class="copy">株式会社ハレックス&nbsp;&nbsp;問い合わせ先：03-5420-4311</div>
100     </div>
101   </div>
102 </body>
103
104 </html>[EOF]

```

例題④-3解説

※例題④-4も同様

④-3 2017年の夏（7月～9月）に35度以上の日数を計算しよう

```

3
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6   <meta http-equiv="Content-Style-Type" content="text/css" />
7   <meta http-equiv="Content-Script-Type" content="text/javascript" />
8   <meta name="viewport" content="width=device-width" />
9   <title>2017年の夏（7月～9月）の大阪駅の35度以上の日数を計算</title>
10  <link rel="stylesheet" href="index.css" type="text/css" />
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
12  <script type="text/javascript" language="javascript">
13    //実行ボタン押下時の処理
14    function onClick() {
15      //APIキーの設定 ※配布されたAPIキーを入力してください
16      var api_key = "WX2000-1xRr4fGm";
17      //緯度経度を取得
18      var lat = document.forms.id_form1.lat.value;
19      var lon = document.forms.id_form1.lon.value;
20      var day_from = document.forms.id_form1.day_from.value;
21      var day_to = document.forms.id_form1.day_to.value;
22      var cnt = 0;
23      var out_txt = "";
24      //気象データの取得処理
25      $.ajax({
26        url: "http://demo.halex.co.jp/wpast/hpd?sid=analysis-p-service&rem=all&lat=" + lat + "&lon=" + lon + "
27        dataType: "json", //気象データの形式はJSONP
28        jsonpCallback: "doJson" //JSONPのコールバック名はデフォルトdoJson
29      });
30      //正常に気象データを取得できた場合の処理
31      .done(function(json) {
32        //keyの数だけループする
33        for (let key of Object.keys(json["Data"])) {
34          //「teMax」：最高気温
35          out_txt = out_txt + "<td>" + json["Data"][key]["teMax"] + "</td></tr>";
36          if (parseFloat(json["Data"][key]["teMax"]) >= 35.0) {
37            cnt = cnt + 1;
38          }
39        }
40        out_txt = "2017年の夏（7月～9月）の35℃以上の日数：" + cnt + "日";
41      });
42      //気象データの取得に失敗した場合の処理
43      .fail(function(json) {
44        alert("気象データの取得に失敗しました。");
45        out_txt = "失敗";
46      });
47      //最終処理
48      .always(function(json) {
49        var target = document.getElementById("output");
50        target.innerHTML = out_txt;
51      });
52    }
53  </script>
54 </head>

```

Ajaxとは、JavaScriptを
効率よく利用するためのフレームワーク

←ボタン押下イベントで実行

←アクセスキーを設定

←気象情報を取得したい地点の緯度経度、日付を設定

←APIリクエストを設定

ループ処理

←日付分だけループ ※["Data"]配下の日付毎に処理

←最高気温が35度以上を判定

←表示用に編集

要素を指定

["Data"]：（固定）

[key]：日付←APIで取得した日付

["teMax"]：最高気温

←変数「output」を表示

JavaScript

④-3 2017年の夏（7月～9月）に35度以上の日数を計算しよう

```

57 <body>
58 <div id="outer">
59 <div id="header">
60 <a href="https://www.halex.co.jp/" target="_blank">
61 
62 </a>
63 <h1>2017年の夏（7月～9月）の大阪駅の35度以上の日数を計算</h1>
64 </div>
65 <ul class="breadcrumb">
66 <li>
67 <a href="index.html" itemprop="url">
68 <span itemprop="title">トップ</span>
69 </a>
70 </li>
71 <li>
72 <span itemprop="title">2017年の夏（7月～9月）の大阪駅の35度以上の日数を計算</span>
73 </li>
74 </ul>
75 <div id="main">
76 <div class="content">
77 <h2>2017年の夏（7月～9月）の大阪駅の35度以上の日数を計算</h2>
78 <p>
79 2017年の夏（7月～9月）の大阪駅の最高気温が35度以上となった日数を計算します。<br>
80 期間：2017年7月1日～2017年9月30日<br>
81 緯度経度：緯度：34.702485、経度：135.495951<br>
82 </p>
83 <h3>緯度経度の設定</h3>
84 <div>
85 <form name="form1" id="id_form1" action="#">
86 <label>
87 緯度：<input type="text" name="lat" id="lat" size="15" maxlength="15" value="34.702485" />
88 経度：<input type="text" name="lon" id="lon" size="15" maxlength="15" value="135.495951" />
89 </label>
90 <label>
91 期間from：<input type="text" name="day_from" id="day_from" size="15" maxlength="15" value="201
92 期間to：<input type="text" name="day_to" id="day_to" size="15" maxlength="15" value="20170930"
93 ※「/（スラッシュ）」なしの「YYYYMMDD」で入力してください。
94 </label>
95 <input type="button" value="実行" onclick="onButtonClick();" />
96 </form>
97 <div id="output"></div>
98 </div>
99 </div>
100 <div id="footer">
101 <div class="copy">株式会社ハレックス &nbsp;&nbsp;問い合わせ先：03-5420-4311</div>
102 </div>
103 </div>
104 </body>
105 </html>

```