

2 スライドショーが完成した

ここまでのコードが記入できたら、ファイルを上書き保存して、index.htmlをブラウザで確認してみてください。もしも上手く表示できない場合は、コンソールにエラーが表示されていないか確認するとい



slickのより詳細な設定方法は、公式サイト (<http://kenwheeler.github.io/slick>) の「setting」で確認できます。



👍 ワンポイント jQueryを利用しない場合もある

jQueryは多機能で汎用的なライブラリですが、意図的に利用しない場合もあります。例えば、高速な処理が必要な場合には、多機能なjQueryを避けて、より軽量なライブラリを利用したり、ライブラリを利用せずにJavaScriptを記述したりする場合があります。また、jQueryと併用できない他のライブラリやフレームワー

クを採用する場合もあります。jQueryが広く用いられていることは間違いないですが、利用できない場合は、JavaScriptそのものの知識が重要になってきます。特にプロを目指している人であれば、両方の書き方を覚えておくことがベストです。

👍 ワンポイント プラグイン／ライブラリの活用にも基本が大切

プラグインやライブラリを利用する際に注意したいのが、変数や関数などの名前の重複です。プラグインやライブラリもプログラムなので、当然JavaScriptの変数や関数を利用しています。そのため、プラグインやライブラリを利用していると、気づかぬうちに変数名や関数名が重

複してしまって、うまく動かなくなるということが起こります。こうしたときに不具合を解消するためには、やはり基本的なJavaScriptの知識が欠かせません。プラグイン／ライブラリを活用して効率よくプログラムを作るためにも、基本が大切なのです。

Chapter

11

Web APIの 基本を学ぼう

この章では「Web API」を利用するために必要な基礎知識を学びます。Web APIを利用すると、GoogleやFacebookなどの外部サービスと連携したWebアプリケーションを制作することができます。



Web APIとは何かを
知しましょうこのレッスンの
ポイント

Webサービスの提供者は、自社のWebサービスをより活用してもらうように「Web API」という仕組みを使ってさまざまな機能を提供しています。Web APIをうまく活用すれば、高機能なサービスをも簡単に作成できるようになります。

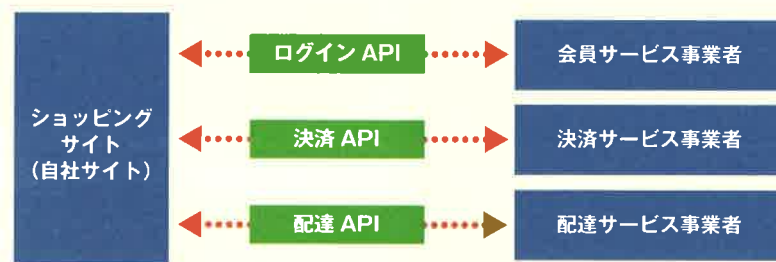
→ Web APIのメリットを知ろう

Web API (Application Programming Interface) とは、Webを通じて利用することのできるサービスと、そのサービスを利用するためのルールのことです。

Web APIは自分で作成できるだけでなく、多くのサービス事業者によって、公開・提供されています。例えば、ショッピングサイトを作ろうとすると「ログ

イン」「決済」「配達」など、さまざまなサービスが必要になりますが、すべてのサービスを自分たちで提供することはとても大変です。そこで、サービス事業者が提供するWeb APIを用いて、既存のサービスと連携していけば、ずっと簡単に目的のサービスを提供できるようになります。

▶ Web APIの利用例



Web APIは、Webを通じてサービスを利用・公開する際に欠かせない仕組みです。



→ Web APIの注意点

Web APIは非常に便利ですが、データのやりとりを行うので、信用できるサービス提供者のもののみ利用してください。また、途中で利用方法が変更さ

れたり、サービスが終了してしまう場合もないとはいえません。事前に利用規約をよく確認して利用してください。

→ 無料で利用できるWeb APIもたくさんある

Web APIを探すには、検索エンジンで「Web API」と検索すれば、たくさんのWeb APIを見つけることができるでしょう。Web APIには、利用料が有料の

ものと、無料のものが存在します。下の表は、無料（一部有料含む）で使用することのできる有名なWeb APIの一部をまとめたものです。

▶ 代表的なWeb API

API名	公式ドキュメント
政府統計の総合窓口	http://www.e-stat.go.jp/api
図書館検索 カーリル	http://calil.jp/doc/api.html
天気情報 (livedoor)	http://weather.livedoor.com/weather_hacks
郵便番号検索API	http://zip.cgis.biz
Instagram	https://www.instagram.com/developer
Twitter	https://dev.twitter.com/docs
Google	https://developers.google.com
Facebook	https://developers.facebook.com
Yahoo	http://developer.yahoo.co.jp
Wikipedia	https://www.mediawiki.org/wiki/API:Main_page/ja

※利用量や機能によっては有料

このリストはほんの一部です。現在では、あらゆるものといっていいくらい多くのサービスがWeb APIで提供されています。



Lesson
60

[基本的な仕組み]

Web APIの仕組みを
知りましょうこのレッスンの
ポイント

Web APIの概要はイメージできましたか？ここではWeb APIがどのような仕組みで実現されているのか、より具体的な例を挙げながら、プログラムの動きを確認していきます。基本的にはWebページを表示するときと同じく、URLを送って結果のデータを受け取る流れです。

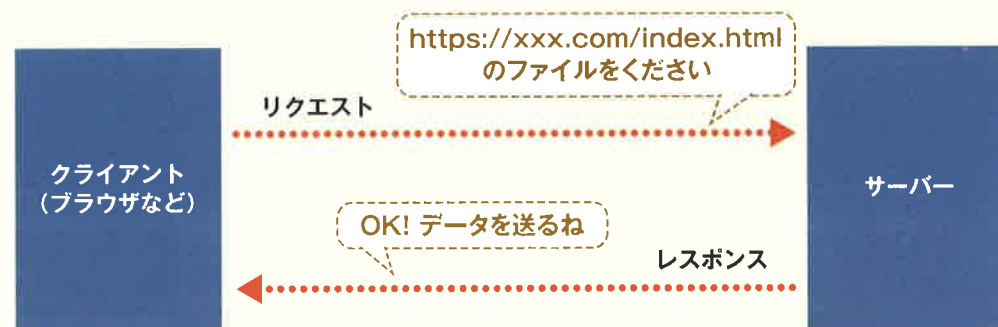
→ Web APIの基本は「リクエスト」と「レスポンス」

Webの世界では、URLを使ってほしい情報を「リクエスト」して、その「レスポンス」として結果を受け取ります。

Web APIも基本的に同じ仕組みです。ですからWeb APIを使用する際は「どのようなURLでリクエストを

すればいいか」「レスポンスをどのようなデータ形式で受け取ることができるのか」「受け取ったデータをどのように活用すればいいのか」といったことが大切になってきます。

▶ リクエストとレスポンスの流れ



Webの世界では、サービスを提供する端末を「サーバー」といい、サービスを受ける端末を「クライアント」といいます。リクエスト(要求)しているのが「クライアント」、レスポンス(応答)を返すのが「サーバー」ともいえます。



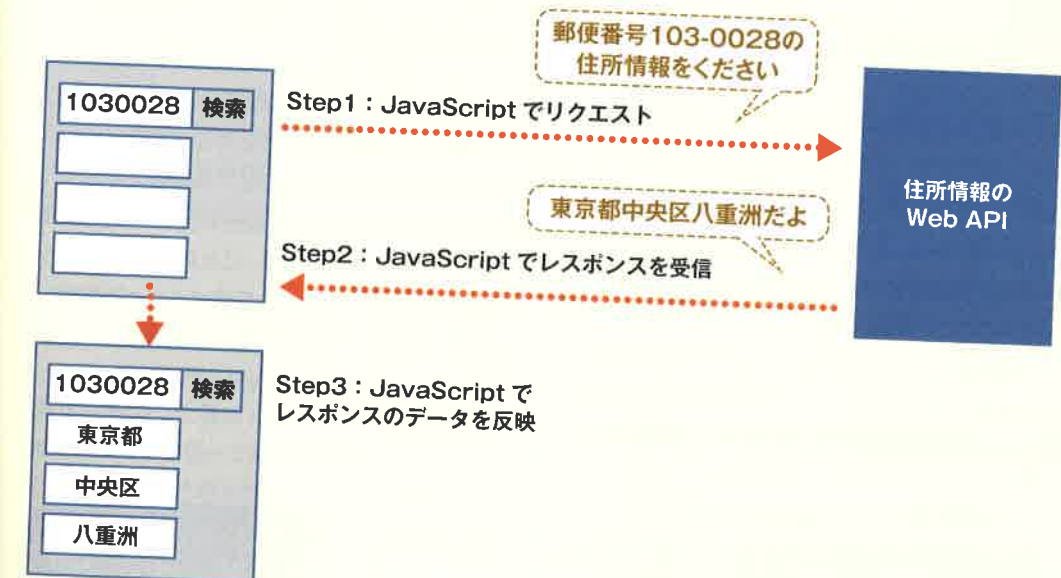
→ 実例を見ながら理解を深めよう

皆さんは郵便番号を入れると自動で住所を入力してくれるフォームを目にしたことはありませんか？この機能は、住所情報を提供してくれるWeb APIを利用することで実現できます。実際に仕組みを見てみましょう。

まず、フォームに郵便番号を入力して検索ボタンをクリックすると、JavaScriptを使ってWeb APIに住所

情報をリクエストするようにします(Step1)。次に、Web APIのレスポンスで提供された住所情報を受信します(Step2)。そして最後に、受信した情報を元に、JavaScriptを使ってフォームに住所情報を反映させます(Step3)。以降のレッスンでは、このデータをやりとりするプログラムを具体的に学んでいきましょう。

▶ 郵便番号から住所を自動入力する仕組み



JavaScriptを使うと、Webページを切り替えることなくサーバーと通信することができます。



Lesson 61 [Ajax] Ajaxについて理解しましょう



このレッスンのポイント

このレッスンではWeb APIを利用する際によく用いる「Ajax(エイジャックス)」というプログラミング手法を学びます。Ajaxを使うと、現在開いているWebページを再読み込みすることなくWeb APIを利用したり、その結果をWebページに反映させることができます。

➡ 操作性の高いWebアプリを実現する「Ajax」

従来のWebアプリケーションでは、サーバーにリクエストを送信することで処理を実行し、レスポンスをWebページとして受け取ることで処理結果を確認していました。でもこの方法では、処理を行うたびにWebページの再読み込みが必要なので、軽快に動作させることが困難でした。

そこで登場したのが「Ajax (Asynchronous JavaScript + XML)」というプログラミング手法です。Ajaxでは、JavaScriptを用いることで、Webページ

全体の再読み込みを行うことなくサーバーと通信し、その結果を画面に反映させることができます。例えば地図アプリのGoogle Mapsでは、画面をドラッグすればそれまで表示されていなかった地図の情報もちゃんと表示してくれますよね。これは、必要になった情報をAjaxでサーバーから取得することで実現しています。先のレッスンで登場した郵便番号による住所の自動入力も、Ajaxを利用しています。

▶ Ajaxを有名にしたGoogle Maps



画面をドラッグすると、ページの再読み込みなしで新たな地図が表示される

Ajaxは「Google Maps」に採用され、その高い操作性から一気に有名になりました。いまでは必須の技術となっています。



➡ Ajaxを実装する

AjaxはJavaScriptだけで記述できますが、ブラウザ間で細かい動作の違いがあるので、ブラウザ間の違いを吸収してくれるjQueryを使って利用することをおすすめします。jQueryを使ってAjaxの処理を記述すると、以下の基本構文のようになります。

\$.ajaxメソッドの引数には、オブジェクトの形式でパラメータを指定します。「url」にはリクエスト先のURL (Web APIなど) を指定します。dataTypeは、レスポンスとして受け取るデータの形式を指定します。

▶ jQueryを利用したAjaxの基本構文

```
$.ajax({
  url: リクエスト先のURL,
  dataType: レスポンスのデータ形式,
  // ...
}).done(function(data) {
  // 通信が成功したときの処理
}).fail(function(data) {
  // 通信が失敗したときの処理
});
```

➡ 通信後の処理を記述する

通信に成功した場合の処理は、\$.ajax()に続けてdoneメソッドを使って記述します。メソッドの引数に実行したい処理を関数で指定します。このとき、関数の引数 (data) にはレスポンスで得られたデータが自動的に格納されます。

Ajaxではサーバーとやりとりしているので、通信状況やサーバーの状態によっては、通信に失敗する場合があります。

通信に失敗した場合の処理はfailメソッドを使って記述します。利用方法はdoneメソッドと同様です。

AjaxとWeb APIの関係は理解できましたか？ Ajaxが利用できるようになると、JavaScriptがぐっと楽しくなりますよ！



62 JSONについて理解しましょう



このレッスンのポイント

Web APIの多くは「XML(エックスエムエル)」や「JSON(ジェイソン)」と呼ばれる通信に適した形式でデータを提供します。Web APIの利用には、こうしたデータ形式の理解が欠かせません。このレッスンでは、近年主流となりつつあるJSONの扱い方について学びましょう。

→ Web APIで利用される代表的なデータ形式「JSON」

クライアントとサーバー間で情報をやりとりするためには、お互いに理解できるデータ形式が必要です。Web APIをはじめ、さまざまなプログラムで広く利用されているのが、JSON (JavaScript Object Notation) というデータ形式です。

実は、このJSONは、JavaScript由来のデータ形式です。JavaScriptには、データをまとめて扱える「オブジェクト」のデータ型がありましたよね。JSONは、このJavaScriptのオブジェクトを、保存や通信に適した形式に変換したものです。

▶ Web APIでよく利用されるデータ形式

名称	特徴
JSON	軽量でJavaScriptとの相性が非常にいい
XML	マークアップで汎用的なデータ構造を作れるが、データ量はJSONより大きくなる

JavaScriptからWeb APIを利用する際は、JSONで通信することが多くなっています。



→ JSONの書き方

JSONの記述方法はJavaScriptの配列やオブジェクトと同じですが、プロパティ名はダブルクォーテーション「"」で囲んだ文字列にしなければならないというルールがあります。

▶ JSON形式で郵便番号と住所情報をまとめたデータ

```
{
  "results": [
    {
      "address1": "東京都",
      "address2": "中央区",
      "address3": "八重洲",
      "zipcode": "1030028"
    },
    {
      "address1": "東京都",
      "address2": "渋谷区",
      "address3": "渋谷",
      "zipcode": "1500002"
    },
    ...
  ]
}
```

1件目の住所情報

2件目の住所情報

すでに配列とオブジェクトについて学んだ皆さんなら、なじみ深い形式ですね。



👍 ワンポイント JSONP

Webページは通常、そのWebページが設置されているドメイン以外のサーバーと通信をすることができません。そのため、この問題に対策を行っていないWeb APIは、JavaScriptから直接利用することができません。こうした問題を解決

する技術に「JSONP (ジェイソンピー: JSON with padding)」があります。JSONPに対応したWeb APIであれば、データもJSONで提供され、ドメインをまたいでスムーズに利用できます。



Web APIで郵便番号から住所を取得してみましょう



このレッスンのポイント

郵便番号を入れると自動で住所を入力してくれるフォームを目にしたことはありませんか？今回は郵便番号から住所情報を提供してくれるWeb API「zipcloud」を利用して、この機能を作成します。実習を通じて、Web APIに必要な技術を復習しましょう。

→ ゴールイメージの確認

今回は、郵便番号を入力して[検索]ボタンをクリックすると自動で住所が入力されるフォームを作成します(以下図参照)。プログラムの流れとしては、まず郵便番号を入力して検索ボタンをクリックする

と、Web API (zipcloud) に住所情報がリクエストされます。その結果をレスポンスとして受け取り、住所情報を取り出して、HTMLに反映します。

①郵便番号を入れて、検索ボタンを押す



②Web API (zipcloud) に住所情報がリクエストされる



③ Web API (zipcloud) からレスポンスを受け取る



④ レスポンスをHTMLに反映する

住所検索

郵便番号 1500002

都道府県

市区町村

住所

郵便番号を入力して
[検索]ボタンをクリック

住所検索

郵便番号 1500002

都道府県 東京都

市区町村 渋谷区

住所 渋谷

住所が自動入力される

→ zipcloud(郵便番号検索API)の使い方

zipcloud(郵便番号検索API)は株式会社アイビスにより提供されているWeb APIで、無料で利用することができます(2017年2月現在)。利用の際は、公式サイト(<http://zipcloud.ibsnet.co.jp/doc/api>)で最新の利用規約を確認してください。

APIの利用方法も公式サイトに掲載されています。以下では、基本的な利用方法を紹介します。なお、Googleで「zipcloud」と検索すると同名のマルウェアの記事も表示されますが、このAPIとは関係ないので、安心してください。

APIを利用する際は、必ず利用規約を確認しましょう。



▶ リクエストの構文

// 郵便番号のハイフン「-」はあってもなくてもOK

<http://zipcloud.ibsnet.co.jp/api/search?zipcode=郵便番号>

▶ レスポンスパラメータ

フィールド名	項目名	備考
status	ステータス	正常時は 200、エラー発生時にはエラーコードが返される
message	メッセージ	エラー発生時に、エラーの内容が返される
results	---	検索結果が複数存在する場合は、以下の項目が配列として返される---
zipcode	郵便番号	7桁の郵便番号。ハイフンなし
prefcode	都道府県コード	JIS X 0401 に定められた2桁の都道府県コード
address1	都道府県名	
address2	市区町村名	
address3	町域名	
kana1	都道府県名カナ	
kana2	市区町村名カナ	
kana3	町域名カナ	

出典: <http://zipcloud.ibsnet.co.jp/doc/api>

入力フォームを作る

1 HTMLファイルを編集する

10/zipcode/practice/index.html

それでは今回も外観となる部分から作成していきましょう。このレッスンのindex.htmlファイルをBracketsで開いて、フォームを表示する以下のコードを記述して上書き保存してください①。

なお、このレッスンのファイルも、jQueryの読み込み処理があらかじめ記載されています。各入力項目(input要素)には、後でJavaScriptから操作しやすいようid属性を付与しています。

1 フォームを作成

```
008 <body>
009   <h3>住所検索</h3>
010   <div>
011     <p>
012       <label>郵便番号<input_id="zipcode"_type="text"_size="10"_max
013       length="8"></label>
014       <button_id="btn">検索</button>
015     </p>
016     <p><label>都道府県<input_id="prefecture"_type="text"_size="10"></label></p>
017     <p><label>市区町村<input_id="city"_type="text"_size="10"></label></p>
018     <p><label>住所<input_id="address"_type="text"_size="10"></label></p>
019   </div>
020   <script_src="js/jquery-3.1.1.min.js"></script>
021   <script_src="js/app.js"></script>
022 </body>
```

住所検索

郵便番号 都道府県 市区町村 住所

外見はこの状態ができていればOKです。



Web APIの動作を確認する

1 動作とレスポンスの確認

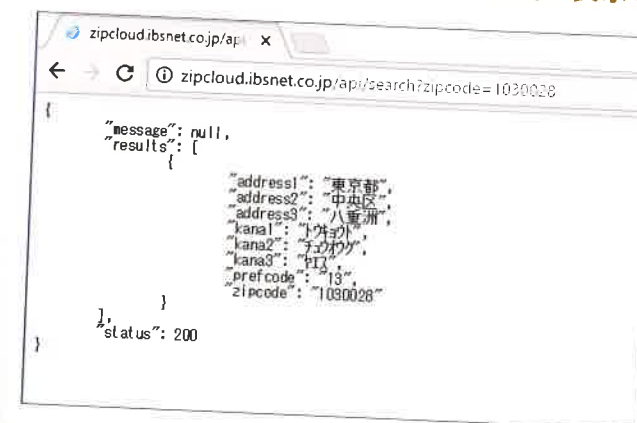
Web APIを利用したプログラムを記述する前に、Web APIが仕様どおりに利用できることを確認しておきましょう。以下の「リクエストの例」で指定されたURLをブラウザのアドレスバーに入力し、アクセスしてみてください。するとレスポンスのJSONがブラウザの画面に表示されるはずです。

こうして見るとWeb APIを利用した通信といっても、ブラウザでWebページを見るのと同じ変わりないことがわかります。これにさらにAjaxを組み合わせると、Webページ全体を更新することなく、一部だけを更新することができるのです。

▶ リクエストの例

http://zipcloud.ibsnet.co.jp/api/search?zipcode=1030028

▶ レスポンスのJSONがブラウザの画面に表示される



第三者の提供するWeb APIは仕様変更やサービス終了が起こる可能性をゼロにすることはできません。プログラムを開始する前に、提供状況を確認しましょう。



郵便番号から住所を取得する

1 検索ボタンに処理を関連付ける

10/zipcode/practice/js/app.js

ここからWeb APIを利用するためのJavaScriptのプログラムを記述していきます。このレッスンのapp.jsファイルをBracketsで開いて、以下のコードを記述してください。

HTML読み込みが完了した後にプログラムが実行されるようにするための処理を書きます①。以降のコードはすべてこの中に書いていきます。

検索ボタンを押したときに処理が実行されるように、検索ボタンのクリックイベントに処理を関連付けましょう。検索ボタンはHTMLを記述する際、id="btn"を指定していたので「\$('#btn)」で取得できます。これにonメソッドを使用してクリックイベントが発生したときの処理を記述していきます②。

```
001 $(function())_{
002   $('#btn').on('click',_function())_{
003     //今後、ここにクリックされたときの処理を記述する
004   };
005 };
```

① 読み込み完了後に実行

② クリックイベントに登録

2 住所情報をリクエストする(Ajax処理)

続いて、Ajaxを使用してWeb APIに住所情報をリクエストする処理を記述していきます。まずは\$.ajaxメソッドに、必要なパラメータを設定していきます①。「url」には、Web APIにリクエストするURLを指定します。今回は郵便番号の情報を付与するため、

「\$('#zipcode').val()」で、フォームに入力されている郵便番号を取得し、結合します②。「dataType」はJSONでデータを取得できるように「json」を指定します③。

```
002   $('#btn').on('click',_function())_{
003     //入力された郵便番号でWebAPIに住所情報をリクエスト
004     $.ajax({
005       url: "http://zipcloud.ibsnet.co.jp/api/search?zipcode="+$('#zipcode').val(),
006       dataType: 'json',
007     });
008   };
```

① \$.ajaxを追加

② urlを指定

③ dataTypeを指定

3 通信に成功した場合の処理を書く(Ajax処理)

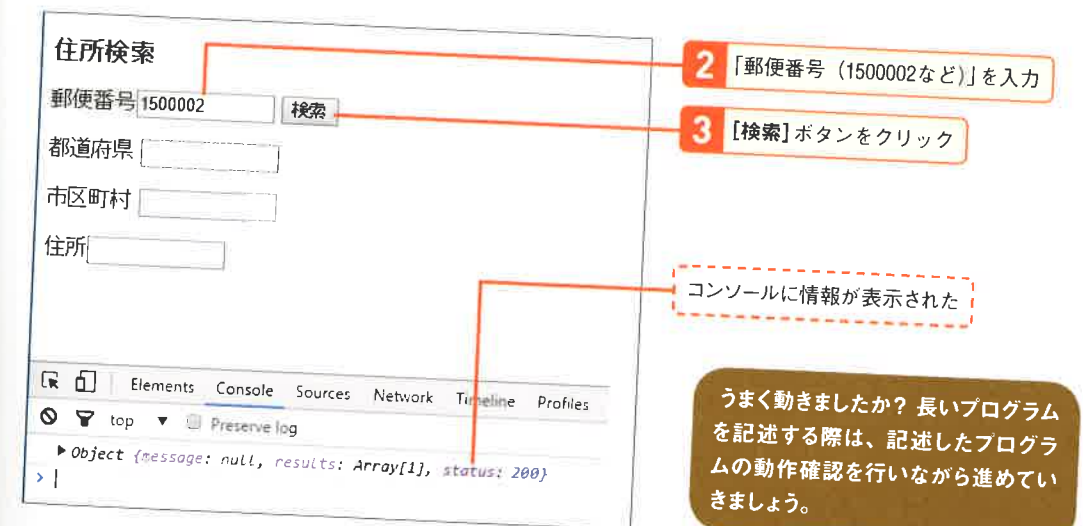
続いて、通信に成功した場合の処理をdoneメソッドを使って記述していきます。

doneメソッドを利用する際に引数を指定すると、その引数から取得したデータにアクセスできます。引数の名前は自由に付けることができますが、今回は「data」という名前にしましょう。これで取得したデータに、引数で指定した変数「data」からアクセスでき

るようになります。dataの中身を確認するため、console.logメソッドで出力する処理を記述しておきましょう①。以下のコードが記述したらファイルを上書き保存し、ブラウザでindex.htmlを開いて動作を確認してください。dataの中身がコンソールに表示されていれば問題なく動作しています②③④。

```
004     $.ajax({
005       url: "http://zipcloud.ibsnet.co.jp/api/search?zipcode="+$('#zipcode').val(),
006       dataType: 'json',
007     }).done(function(data)_{
008       console.log(data); //取得できているかの確認用。後で消します。
009     });
```

① コンソールに表示



② 「郵便番号 (1500002など)」を入力

③ 「検索」ボタンをクリック

コンソールに情報が表示された

うまく動きましたか？ 長いプログラムを記述する際は、記述したプログラムの動作確認を行いながら進めていきましょう。



NEXT PAGE → 237

4 エラーが起きた際の処理を書く (Ajax処理)

通信に成功しても、郵便番号が正しく指定されていなければ住所を取得することはできません。このように、問題が起きた場合の処理も記述しておきましょう。動作確認用に記述していた「console.log(data);」の行を削除して、「data.results」が取得できなかった場合の条件分岐を追記します①。

また、ネットワーク経路に障害があるなどの理由で通信自体に失敗した場合も住所を取得することはできません。通信に失敗した場合の処理はfailメソッドを使って記述することができます②。いずれの場合も、うまくデータを取得できなかった場合にはメッセージを表示するようにしておきましょう。

```
007 .....}.done(function(data){
008 .....if_(data.results){
009 .....// データが取得できたときの処理を書く
010 .....} else {
011 .....alert('該当するデータが見つかりませんでした');
012 .....}
013 .....}.fail(function(data){
014 .....alert('通信に失敗しました');
015 .....});
```

① 取得できなかった場合の条件分岐

② failメソッドを追加

▶ 動作確認: 結果が取得できない場合

このページの内容:
該当するデータが見つかりませんでした

OK

何らかの問題で結果が取得できなかった場合は、このメッセージが表示される

▶ 動作確認: 通信に失敗した時

このページの内容:
通信に失敗しました

OK

インターネットにつながっていない場合などは、このメッセージが表示される

ユーザーに安心して利用してもらうには、エラーに備えたプログラムが重要です。



○ 住所情報をHTMLに反映する

1 JSONからデータを取り出す関数を作る

10/zipcode/practice/js/app.js

最後に、Web APIから取得した情報をHTMLに反映する処理を記述しましょう。

取得したJSONから住所情報を取り出して、HTMLに反映するsetData関数を作成します①。引数とし

て取得したJSONの「results」のフィールドにある住所情報を受け取ると、HTMLの各input要素の値として設定するようにします②。

```
017 .....
018 .....// データ取得が成功したときの処理
019 .....function setData(data){
020 .....//取得したデータを各HTML要素に代入
021 .....$('#prefecture').val(data.address1); // 都道府県名
022 .....$('#city').val(data.address2); // 市区町村名
023 .....$('#address').val(data.address3); // 町域名
024 .....}
025 .....};
```

① 関数を定義

② input要素に設定

2 関数を呼び出す

関数ができあがったら、doneメソッド内のデータの取得処理が完了した部分でsetData関数を呼び出すようにします①。setData関数の引数には住所の情報を指定する必要があります。P235に記載された

レスポンスのJSONを確認すると、住所の情報は、resultsフィールドの配列に入っていることがわかります。そこで引数に「data.results[0]」を指定して、住所情報だけが渡るようにしています。

```
007 .....}.done(function(data){
008 .....if_(data.results){
009 .....setData(data.results[0]);
010 .....} else {
011 .....alert('該当するデータが見つかりませんでした');
012 .....}
013 .....}.fail(function(data){
014 .....alert('通信に失敗しました');
015 .....});
```

① 関数を呼び出す

3 プログラムが完成した

プログラムが完成したら、内容を上書き保存して、... う①②。
index.htmlをブラウザで開いて動作を確認しましょ

住所検索

郵便番号

都道府県

市区町村

住所

1 郵便番号を入力

2 [検索] ボタンをクリック

住所が自動入力される

住所検索

郵便番号

都道府県

市区町村

住所

うまく動作しましたか？ うまく動作しない場合は、サンプルファイルの完成品と違いを見比べてみましょう。



Chapter

12

YouTubeの 動画ギャラリー を作ろう

この章では、学習の総まとめとして、Googleが提供するYouTube Data API (v3) を使った、ビデオギャラリーを作成します。いままでに学んだことを生かしてWebサイトを完成させましょう。

