

第3クールSeason1

# WebサービスAPI勉強会

「Javascriptの文法ワークとAPI」

## 1

## Season1のゴールは？

Season1のゴールはJavascriptを使ってAPIからデータを取得し、それをブラウザに表示する『最低限』のことだけを実施します。

Javascriptもまだあまりよくわかっていない、HTMLのタグの中に挿入すれば良い、食わず嫌いだ・・・etc.そのような方でもほんの少しだけJavascriptを使ってみて、APIを触るところまで進めていきます。

一見、無謀のようなチャレンジですが、この勉強会で集中して一個ずつこなしていけば、マスター出来ると思います。

そして、この勉強会が終わってからがこのSeason1の本番です。今日の成果物を元に、自分なりのアレンジをして応用すること、自分で試してみることによって、本当の勉強が始まるのです。

今日はそのスタートラインに過ぎません。集中して、勉強していきましょう。

※前回は毎回の勉強会をPart.1、Part2と区切っていましたが、今回からはSeason1、Season2と区切るようにしました。  
これは海外ドラマにハマりすぎの主催者の好みであり、特に意図されたものではありません。

が、表向きの理由としては

「次回までの勉強会の間、勉強会当日にやったことを復習し、自分なりに応用、カスタマイズして失敗や成功を繰り返す、勉強会コミュニティドラマ」

という位置づけにしています(笑)。

## ワーク

①事前にテキストの全体の見出しだけでも見て、全体像を把握しておきましょう。出来れば、完璧に理解できなくとも読んでおく当日のわからないところが明確に出来ます。

(今後の勉強会でも共通です)

②今日の勉強会が終わったらやることを決めてここにメモしておく。

例：復習、自分の好みの色のパーツを作る・・・等

*memo*

## 2

## JavaScriptとは？

JavascriptとはWebでよく使用されるスクリプト(プログラム)言語のひとつです。

▽WikipediaによるJavaScriptの記述

<http://ja.wikipedia.org/wiki/JavaScript>

PHPやPerlなどと比べて大きく違うのがウェブブラウザ上で動くということです。それとは違いPHPやPerlなどはサーバ上で動くスクリプト言語です。

Javascriptはウェブブラウザ上で動くのでサーバだけでは出来ないことが実現出来ます。例えば

- ・あるポイントをマウスオーバーしたときに表示内容を変える
- ・現在の時間をリアルタイムに表示させる(カウントさせる)
- ・ブラウザのサイズを大きくしたり小さくしたりできる

などが出来ます。さらにそれらを応用して

- ・Google マップ

<http://maps.google.co.jp/>

- ・Apple iPodサイト

<http://www.apple.com/jp/ipod/>

などのAjax利用サイトがあります。

※Ajaxについては第4クールでやります。

## ワーク

①Ajaxサイトを見て、ソースを見てみましょう。

→Flashが使われていないことを確認する。Flashを使わなくてもHTMLとJavascriptだけでこれだけ動きのあるページを作ることができる、ということを確認する。

②Javascriptを勉強することの再確認。JavaScriptやAjaxで動きのあるサイトを作ることができることの再確認。

## 3

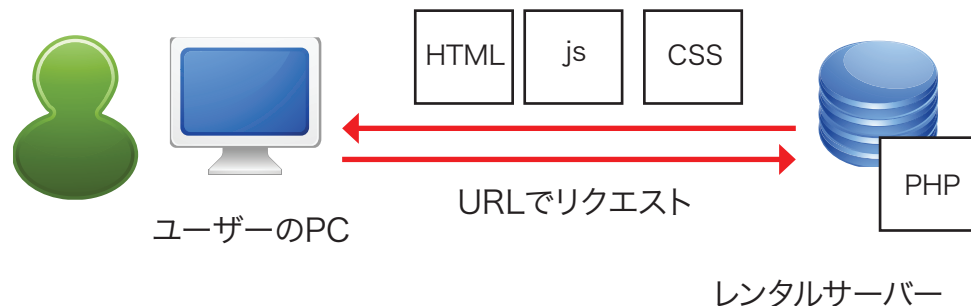
## サーバーサイドとクライアントサイドの違い

Javascriptはクライアントサイドスクリプトに分類されます。

クライアントサイドスクリプトとは、サーバーサイドスクリプトの対義語であり、サーバーに対してのクライアント、つまり、インターネット上であれば、Webブラウザ(ユーザが使用するパソコンのブラウザ)を指します。

Webで扱うサーバサイドスクリプトはPHP、Perl、Rubyなど種類も多いです。現状のWebサイトの多くはこれらの言語とJavascriptを組み合わせたものがほとんどです。ブログやWebサービスなどもほとんどがこのパターンに当てはまります。

注意すべきことは、ユーザーがページをリクエストしてから、帰ってくるまでにどういった順番でスクリプトが実行されるか、ということです。これを理解していないと、組み合わせたサイトを作れませんので、理解しておきましょう。



## ワーク

①ここでクイズです。

PHPで動かすプログラムはクライアント(ユーザのPC)ではHTML、JavaScript、CSS(スタイルシート)のどの形になって、ブラウザ上での表示に影響することが出来るでしょうか？

- a.HTML
- b.JavaScript
- c.CSS
- d.全部

## 4

## HTML中のJavaScriptの書き方

JavaScriptは基本的にはHTMLの中に書きます。下記のようにHTMLの<script>タグで挟んで記述します。

```
<script type="text/javascript">
//
document.write('&lt;link rel="stylesheet" href="http:
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="38 464 457 578" data-label="Text"><p>また、スタイルシートと同じように外部ファイル化することが出来ます。外部ファイルは拡張子を「.js」とし、JavaScriptのみを記述します。外部ファイルにした.jsファイルを表示したいページで読み込むには下記のような記述をします。</p></div><div data-bbox="38 614 444 670" data-label="Text"><pre>&lt;script type='text/javascript' src='http://musilog.net/wp-
includes/js/prototype.js?ver=1.6.1'&gt;&lt;/script&gt;</pre></div><div data-bbox="38 705 458 789" data-label="Text"><p>JavaScriptはこのように2種類の書き方がありますが、汎用的なものは外部ファイル化し、そのページ特有のものはHTMLに直接書き込むなどをして使い分けていきましょう。</p></div><div data-bbox="545 127 602 153" data-label="Section-Header">ワーク</div><div data-bbox="534 192 960 246" data-label="Text"><p>①自分のブログやサイトのソースを見て、JavaScriptがどのように書かれているか確認してみましょう。</p></div><div data-bbox="548 252 906 306" data-label="Text"><p>例：外部ファイル化されているか？直接書かれているなら、どんな書き方をされているか。</p></div><div data-bbox="47 963 62 986" data-label="Page-Footer">8</div><div data-bbox="121 961 372 980" data-label="Page-Footer">Copyright (c) 2010 Takashi Wakimura All Rights Reserved.</div><div data-bbox="617 961 876 980" data-label="Page-Footer">Copyright (c) 2010 Takashi Wakimura All Rights Reserved.</div><div data-bbox="940 963 957 986" data-label="Page-Footer">9</div>
```

## 5

## JavaScriptを書き初めてみよう

それでは、ここからはいよいよ実践でJavaScriptを書き始めていきます。今回はHTMLファイルの中に直接書いていきます。  
(外部ファイル化しません)

サンプルファイルをダウンロード・解凍し、XAMPPのhtdocsの下にseason1フォルダごとに入れてください。templateフォルダにテンプレートファイル(元ファイル)が入っているので、それをworkフォルダにコピーして作業をしていきます。もし、失敗してわからなくなっても、templateフォルダからコピーしてやり直せば良いだけです。

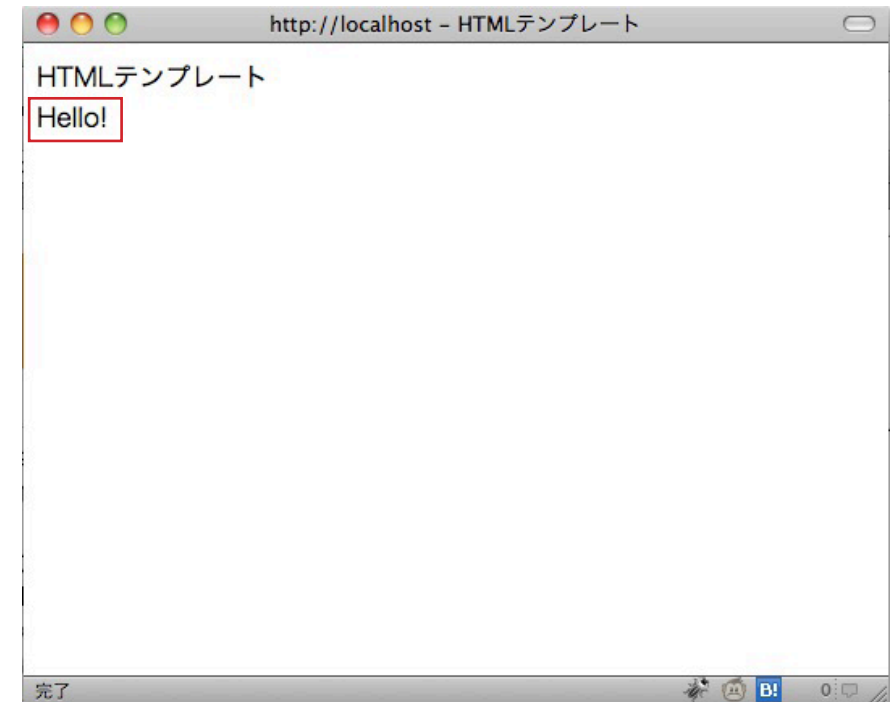
さて、first.htmlをコピーしてworkフォルダに入れます。下記記述を入れてみましょう。

```
<script type="text/javascript">
<!--
document.write("Hello!");
//-->
</script>
```

※テキストをコピペした場合、PDFファイルの都合上半角スペースや全角が混じっている場合がありますので、きちんと確認しながらコピペするか手打ちするようにしてください。

## ワーク

①左記のようにJavaScriptを入れて、表示を確認してみましょう。



※XAMPPのインストール・起動はオリエンテーションで済んでいる前提ですが、導入していない場合はファイルをダブルクリックしてブラウザで表示させてみてください。

確認URL

<http://localhost/season1/work/first.html>

## 6

## JavaScriptの書き方確認

前ページのように書くことによって、JavaScriptが動作することがわかりました。これはブラウザによって、このJavaScriptが解釈されて実行された、とも言えます。

それではその書き方について確認していきたいと思います。

```
<script type="text/javascript">
<!--
スクリプトの中身
//-->
</script>
```

HTMLの中ではscriptタグを上記のように書き、スクリプトの中身を挟んでやります。コメントタグが入っていますが、ここでは詳細の解説は省略して、とにかくこの赤い文字を必ず書いて、その間にスクリプトを書いていくと覚えてください。

さて、document.writeというのはどうやら、文字を出力するものようです。これについては次のページで学びましょう。

## ワーク

①同じfirst.htmlの”HTML文中に”もうひとつ同じように左記のようにJavascriptを書き、document.write("Hello!");の代わりに好きな文字を表示させるように書いてみてください。

②document.writeで出力する内容にHTMLタグも使えます。<b>タグなどで文字を装飾してみてください。

## 【参考】

文字列を示す引用符は「" "」と「' '」と二種類使えます。HTMLタグ中にある「" "」があればスクリプトが文字列の範囲の解釈を誤るので、文字列を囲うほうを「' '」、HTML中の引用符は「" "」とします。

※別途エスケープという方法もありますが、ここでは省きます。

※スクリプトの中身は原則半角文字、スペース等でエディタ上見分けがつかない全角スペースを入れないように気をつけましょう。

document.writeの””記号の中身だけ全角文字や全角スペースが使えます。

## 7

## ナビゲータオブジェクトとメソッド

Javascriptにはブラウザの部品を”オブジェクト”として表すことができます。例えばブラウザのウィンドウやボタンや画像や履歴などの部品です。これらのオブジェクトをまとめて”ナビゲータオブジェクト”と言います。

ナビゲータオブジェクトの値を変更したり、調べたりしてブラウザ上で動的に動くサイトを実現します。

```
document.write("Hello!");
```

さきほど実行したavaScriptはこのような形でした。これはdocumentというオブジェクトに対してwriteという”メソッド”を動作させています。

writeは文字を書き出すメソッドです。

このようにオブジェクトとメソッドをピリオドでつなげて、括弧の中身の値(文字列)を与えれば、documentオブジェクトに、つまりそのHTMLファイルのその部分に文字を書き出すことになります。

## ワーク

①「ナビゲータオブジェクト」で検索していくつかのサイトを見てみる。

→階層構造になっていることをチェックする

→windowオブジェクトは省略できる

→それぞれのオブジェクトでどんなことができそうかチェックする

例: JavaScriptのページ/ナビゲーターオブジェクト

<http://bit.ly/aixgsA>



## 8

## 変数とは？

変数とは例えて言うと、値(文字列)などを格納できる空間的なメモリみたいなものです。もっと例えれば、ハコのようなものです。

そのハコには計算が可能な数値が入る場合もあれば、文字列が入る場合もあります。そして場合によってはそのハコの中に仕切り板をつけて、一番目の棚と二番目の棚に違う文字を入れられます。そのような変数のことを配列変数と呼びます。

それでは、早速変数を使ってみましょう。

ここでの課題は変数に文字列を記憶させ(一般的に代入と言います)、その文字列をdocument.writeで書き出してみましょう。

変数は次のような形で代入することができます。

```
var hako1 = "ハコ1";
var hako2 = "ハコ2";
```

そして、この変数を出力するときは変数を値として指定してwriteメソッドを実行させれば良いことになります。

```
document.write(hako1);
```

変数のhako1の内容がブラウザ上に表示されます。

## ワーク

①同じようにscriptタグを入れて、変数に入れて文字を出力してみましょう。

```
<script type="text/javascript">
<!--
var hako1 = "ハコ1";
var hako2 = "ハコ2";
document.write(hako1);
//-->
</script>
```

```
document.write("hako1");
```

とすると、変数hako1の値が表示されるのではなく、文字列としての「hako1」が表示されます。「” ”」で囲まれた部分は文字列として処理されるからです。

②変数名を変えて表示が変わることも確認しましょう。

```
document.write(hako2);
```

確認URL

<http://localhost/season1/work/first.html>

## 9

## 関数(function)とは？

スクリプトを書き続けていくうちによく使うものや何かのアクションがあってから呼び出したいものがあります。そういった処理の流れのスクリプトを名前を付けて定義したものを関数と言います。

関数の定義は下記のように行ないます。

```
function 関数名(引数) {
  スクリプト
}
```

functionで定義されている範囲のスクリプトはHTMLのソース上、その場では実行されません。関数が定義されるだけです。呼び出されたタイミングで初めてその関数の中身が実行されます。

また、関数を呼び出すときに値を引数として渡すことにより呼び出し元で作られた値によって違う動作をする関数を作することもできます。

呼び出し方はいろんなパターンがありますが、単純に下記のように書いて、呼び出す事もできます。

関数名(引数);

## ワーク

①新しいテンプレートファイルfunction.htmlをworkフォルダにコピーして編集作業を始めます。

関数を作成し、HTML文中で変数を定義して、それを引数として関数を呼び出してみましょう。

a.関数の作成:<head>タグの中で定義してみましょう。

```
function hyouji(hikisu) {
  document.write(hikisu);
}
```

b.変数に代入、関数の呼び出し:本文中で呼び出しましょう。

```
var hako1 = "ハコ1";
var hako2 = "ハコ2";
hyouji(hako1);
```

※通常関数の定義は<head>タグ内で行い、関数を呼び出すのは本文中で呼び出します。これは、関数の定義をする前に関数を呼び出してしまう可能性があるからです。

※当然、scriptタグなどで囲ってください。

確認URL

<http://localhost/season1/work/function.html>

## 10

## 繰り返し処理

プログラミングの世界ではある一定の基準をもとに繰り返し処理することがよくあります。

例えば、商品検索APIなどを使うときに、キーワードをAPIへ投げれば10個のデータが返ってきます。その返ってきたデータを1個の商品を表示する処理を10回繰り返して表示します。

これは10個分の処理を書き続けるよりも合理的です。

```
for (i=1;i<=10;i++) {
  document.write(i+"という数字を表示したよ！<br>");
  var kakezan = i*5;
  document.write("5倍すると"+kakezan+"だよ！<br>");
}
```

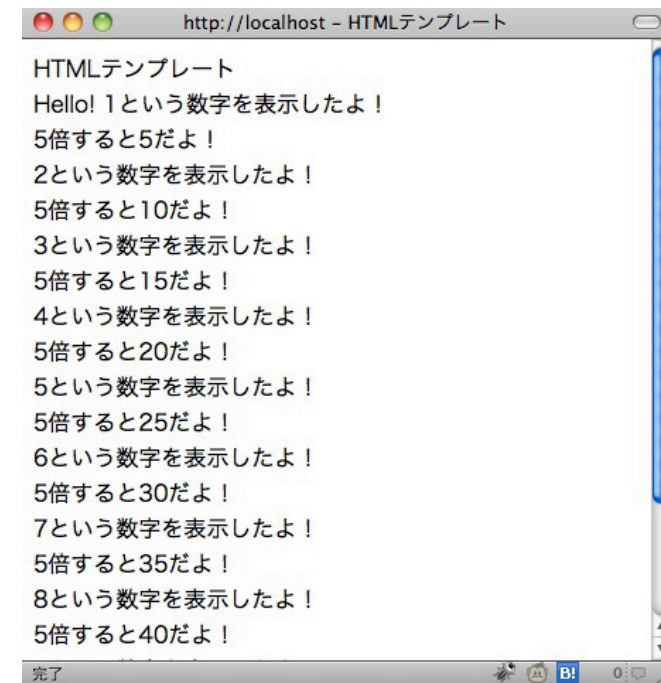
繰り返し処理するときには、変数に代入しながらカウントし、繰り返しています。そのため、何回繰り返したかもその変数を出力してみるとわかります。

```
for (初期値;条件式;増減値) {
  スクリプト
}
```

## ワーク

①新しいテンプレートファイルfor.htmlをworkフォルダにコピーして編集作業を始めます。

左記のようなスクリプトを入れて実行してみてください。また、条件式を変えてみて、どのように変わるか実行してみてください。



確認URL

<http://localhost/season1/work/for.html>

## 11

## 1 時間目のまとめ

1 時間目はAPIを扱うにあたっての最低限のJavaScriptの知識を最低限にしてまとめました。最低限ということは、活用していくためにはこれ以上の知識も要求されるし、さらには1 時間目でやった土台がしっかりと固まっていないと応用もできません。

1 時間目の内容はそのスクリプトの解説は氷山の一角と例えても良いぐらい、断片的な情報です。現在はJavaScriptの解説書やグーグル大先生などで情報を集められますので、それで少しずつ、知識の肉を付けてください。

現在は骨だけ、というよりも骨粗鬆症の骨だけのようなJavaScriptの知識です。復習時にワークをもう一度やってみたり、自分で変数名や関数名を変えたり、値を変えたりして練習を繰り返してみてください。エラーが出たりうまくいかないことも多いでしょう。

その問題解決をして、初めて実力になります。

1 時間目ではプログラミング、スクリプト言語の骨格とも言える、文法、変数、関数、繰り返し処理のごく一部を取り扱いました。逆に言えば、これらの概念さえ理解すれば、徐々に力をつけて伸ばしていくことができます。

ぜひトライアンドエラーで練習を積み重ねていきましょう。

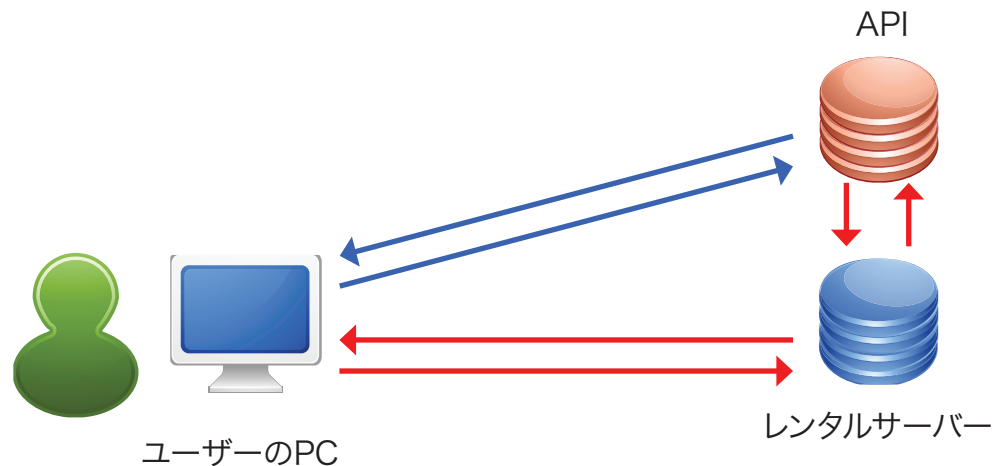
## ワーク

- ①これまでのワークを繰り返してみる
- ②JavaScriptについてよくまとまっているサイトを探すが、一冊「JavaScript辞典」などのような書籍を買って手元に置いておく。

## 12

## Web APIの使い方の基本

APIの使い方の基本はリクエストURLを組み立てて、レスポンスを受け取り、それをブラウザで表示させるスクリプトを書くことです。



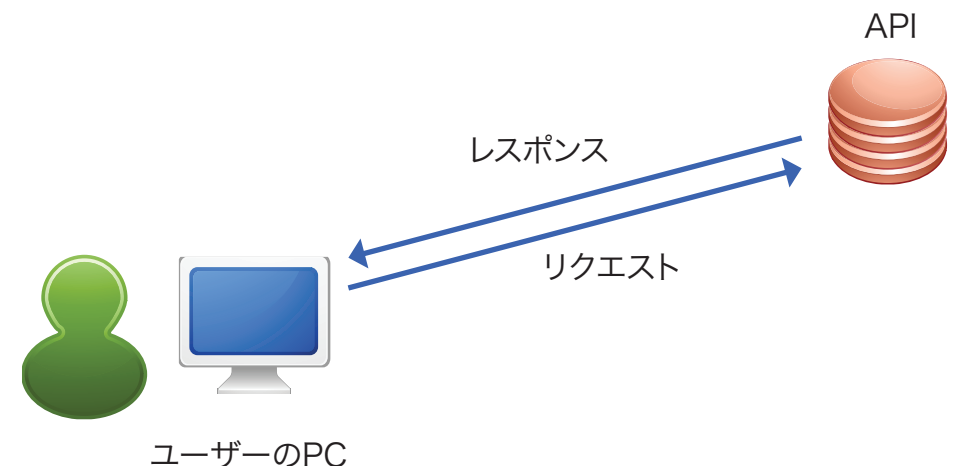
PHPとJavaScriptではデータ通信経路の違い、スクリプトが実行される場所の違いがありますが、リクエストを行い、レスポンスを受け取るという流れは変わりません。

JavaScriptではユーザのブラウザから直接APIへリクエストを行います。また、返ってきたデータを元にJavaScriptでブラウザ上に表示させます。

リクエストURLとはAPIの基本URL(ベースポイント)にパラメータを付けて組み立てます。パラメータはAPIによって異なりますが、主にAPIキーや検索キーワード、ソート順、取得件数、アフィリエイトIDなどです。

それに対してレスポンスは、その検索結果に関する情報概要と検索結果のデータのセットを返します。データのセットとは商品検索であれば、商品名、商品説明、リンク先、商品画像、価格などがそれに該当します。

これらの内容はそれぞれのAPIごとに定められたルールがあります。そのルールに従って、翻訳作業を行うことがプログラミングに求められます。



## 12

## リクエストとレスポンスの翻訳

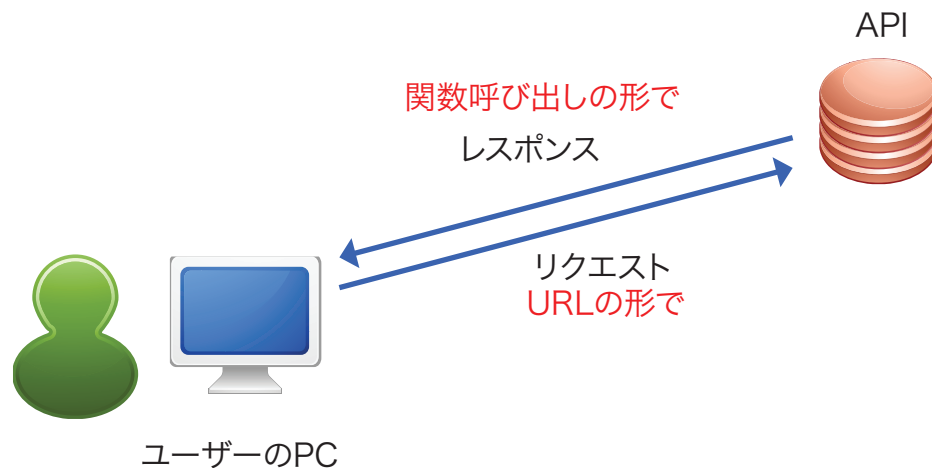
つまり、リクエスト内容とレスポンスの翻訳作業が肝となります。

リクエストはURLにパラメータを組み立てた形で組み立てます。パラメータにはリクエストするための条件を含めます。

一方レスポンスは今回はJavaScriptでJSON形式を扱う場合について説明します。レスポンスはJavaScriptの関数呼び出しの形返ってきます。

これに対してJavascriptであらかじめそれを解釈しブラウザに表示する関数を作っておきます。

ちょうどP.18の関数の話を思い出してみてください。変数に入るものがAPIから返ってきたデータに置き換わったのと同じ流れです。



## ワーク

①p.18の関数とは?の部分を読み直しておく。

特に関数を定義していること。そして関数から引数でデータを受渡しているところの流れを再度確認。今後のAPIではその引数の中に入るデータの量が数文字ではなくかなりの文字数であったり、形式が異なるものだったりするので、見た目で混乱しないように、その基本的な流れを再確認しましょう。



## 13

## 楽天ウェブサービスを使う

JavaScriptからAPIを扱えるもので、アフィリエイトにもつながるものとして楽天市場商品検索などの楽天ウェブサービスを使ってみようと思います。楽天ウェブサービスとは楽天のAPIを提供するシステムの総称です。開発者は無料で使うことができます。

## ▽楽天ウェブサービス

<http://webservice.rakuten.co.jp/>

楽天ウェブサービスを使うには楽天市場に会員登録が必要です。また、楽天ウェブサービスを初めてお使いの方は楽天ウェブサービスのページで、登録する必要があります。

APIを使用するにあたって、下記二つの項目が必要です。これらはリクエストURLのパラメータとして使います。

- ・デベロッパーID
- ・アフィリエイトID

なお、現在デベロッパーIDの確認には文字入力とメール認証の二つが必要になっており、10分ぐらいかかるので事前に準備しておくことが望ましいです。そして忘れないようにするためにもきちんと書き留めておくことが大切です。

また、新たに他社APIを使用する時も同様ですが、規約を一度読んでおいてください。禁止事項を知らずに違反になるケースがあります。

## ワーク

①自分の楽天デベロッパーIDと楽天アフィリエイトIDをテキストエディタなどでコピペでメモって保存しておきましょう。(いつでも参照できるように)。

②時間のある時で良いので規約を必ず読んでおきましょう。

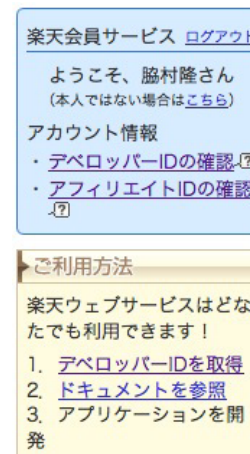
## ▽楽天ウェブサービス規約

<http://webservice.rakuten.co.jp/rule/>

③クレジット表示についても読んでおきましょう

## ▽クレジット表示

<http://webservice.rakuten.co.jp/credit/>



## 14

## APIリファレンスの見方

各APIの仕様書・リファレンスはWebもしくはPDF形式でのダウンロードなどで用意されています。楽天ウェブサービスの場合はWebで確認できます。

どのAPIも基本的には

- ・リクエストURL(ベースURL)
- ・リクエストパラメータ(入力パラメータ)
- ・レスポンスフィールド(出力パラメータ)

この3要素で仕様書が構成されています。ただし提供事業者によって呼び名は少し違いがありますので、ご注意ください。

それぞれフィールドが用意されていて、それぞれのフィールドを指定することにより、その条件でリクエストできます。デベロッパーID/APIキーなどは必須項目になっていることが多いです。

また返ってきたフィールドのデータは何のデータかも、ここでわかります。まずは仕様書に目を慣らすという意味も含めて一通り目を通してみましょう。

## ワーク

①楽天商品検索API (version:2010-09-15)でAPIのリファレンスに目を通してみましょう。ご一緒に見ていきましょう。

<http://webservice.rakuten.co.jp/api/itemsearch/>

②自分がAPIでサイトを作るとしたら、こんな項目を表示させたい、というイメージを作る

*memo*



## 15

## 日本語文字はURLエンコードする

よく使われる商品検索APIですが、半角英数字だけでなく、日本語・全角文字で検索することのほうが多いでしょう。これら全角文字はURLに含めることができないので、半角英数字の特殊な形態に変換して、パラメータとして組み上げる必要があります。

これをURLエンコードと言います。この処理は今後のマルチバイト文字(全角文字、日本語のひらがなやカタカナや記号など)の処理に必須です。また、その反対の元に戻すことをURLデコードと言います。

ここからのワークでは参加者それぞれがキーワードを思い浮かべそれぞれの商品検索結果画面を出していきます。ですので、まわり方と表示検索結果数が違ったりする可能性があります。

## ワーク

①楽天市場商品検索で必ず引っかけりそうな商品名をURLエンコードして、テキストエディタ等でもメモしておく(後で使います)。

この次に、実際にAPIにアクセスするときに使います。

※レアものすぎる商品とか検索に引っかからないワードだとプログラムが正しくても商品が表示されない可能性があります。

▽サンプルファイル付属URLエンコードツール

<http://localhost/season1/tools/urlencodetool/urlencode.html>

(PHPを使用しているので必ずXAMPPでApacheを起動している必要があります。)

※XAMPPを入れられていない方は「URLエンコード」等でググって「UTF-8」でエンコードしてくれるサービスを提供しているサイトがあるので、そこを利用しましょう。文字コード「UTF-8」を決して指定するのを忘れないように。

<http://www.tagindex.com/tool/url.html>

こういった、サイトはググればいくつか出てきますが、自分専用にカスタマイズして持っておくのも良いでしょう。

例: ショッピングモールの検索アフィリエイトリンク作成等

## 16

## リクエストURLの組み立て方

リクエストURLは、リファレンスを見ながら少しずつ組み立てていきます。ひとつずつ項目を確認し、まずは必須なところから埋めていきます(パラメータの順番は問いません)。

ベースURLとパラメータの間は「?」で繋げて、その後がパラメータであることを示します。またパラメータを複数つなげるときは「&」でつなげていきます。

各パラメータは「developerId=xxxxx」とフィールド名に「=」でつなげて値を指定します。

▽楽天商品検索API (version:2010-09-15)

<http://webservice.rakuten.co.jp/api/itemsearch/>

今回JavaScriptでAPIにアクセスするのにREST形式ではなくJSONP形式を選択します。REST形式はPHPで良く使いましたが、JavaScriptで扱いやすいJSONP形式を使います。

まず、ベースURLは  
[http://api.rakuten.co.jp/rws/3.0/](http://api.rakuten.co.jp/rws/3.0/json?[parameter]=[value]...)  
[json?\[parameter\]=\[value\]...](http://api.rakuten.co.jp/rws/3.0/json?[parameter]=[value]...)

とし、callBack=callbackFuncというパラメータを付けることでJSONP形式で扱う手続きができます。

その他のパラメータをワークで順次指定していきましょう。

## ワーク

①最低限のパラメータの指定をして、リクエストURLを組み立てていきます。テキストエディタに書き込んでいきましょう。

まずはベースURLの確認。このあと「?」でパラメータを繋げます。  
<http://api.rakuten.co.jp/rws/3.0/json?>

必須なパラメータは

developerId=【あなたのデベロッパID】

operation=ItemSearch

keyword=【さきほどURLエンコードした検索キーワード】

version=2010-09-15

callBack=callbackFunc

です。

アフィリエイトリンクを作成されたい方は

affiliateId=【あなたのアフィリエイトID】

とします。

余裕があれば、他のパラメータも追記してみてください。  
ここでのリクエストURLは後ほどまた使います。



## 18

## レスポンスの解読方法

レスポンスを凝視していると一定の法則が見つかります。PHPでREST形式でAPIを利用したときはXML形式でブラウザ上で表示させてもクリックしてタグを閉じたりしたりして階層構造が分かりやすかったものです。しかし、今回はブラウザ上でも単なる文字列としか見えません。

そこで、下記ルールで読んでみると、なんとなく読めてきます。

"affiliateRate":1・・・アフィリエイトレートが数値1である  
カンマ「,」でデータが繋がられている

"mediumImageUrl":"http://thumbnail.image.rakuten.  
co.jp/@0\_mall/e-besya/cabinet/a-kusu0/039-0002-1.  
jpg?\_ex=128x128"・・・中ぐらいの画像URLがそのアドレスである。

({"Body":{"ItemSearch":{"Items":{"Item":  
階層構造になっている。複数のデータセットがある。

フッターのほうを見ると検索した条件などの情報が読める。ここでは、完璧にこのデータを解読できることではなく、階層構造を持っていることと、どのような書式でデータが入っていることだけを理解する。

## ワーク

①レスポンスを解読して、画像URLをブラウザに入れて直接表示してみたり、リンクURLを直接入力して、アクセスしてみたりして、返ってきたデータを確認してみよう。

- ・商品画像URLでアクセスして画像がでるか？
- ・アフィリエイトリンクでアクセスしてショップが出るか？

## 19

## 商品検索結果表示

では、早速レスポンスも確認し、それが関数呼び出しの形になっていることを確認したので、画面に検索結果を表示する関数を作成したいと思います。

templateフォルダのsearch.htmlをworkフォルダに移動してください。

すでに関数などは書き込まれています。確認していきましょう。

まず、リクエストURLを組み立てたものは本文中の  
`<script type="text/javascript" src="【あなたが作ったリクエストURL】">`  
 という部分に入れます。

するとこの場所で先程のブラウザで表示された内容が読み込まれ、関数が実行されることになります。この関数は引数として、先程のブラウザで見たとおりの多数の商品情報を含んでいます。

今度呼び出される側の関数はheadタグ内で定義しています。この関数はjsonpdataという変数で引数を受け取り、この変数から商品データを抜き出して、複数の商品を一個ずつ表示する流れになっています。

参考：公開API活用ガイド

<http://www.amazon.co.jp/dp/4777515362/>

## ワーク

①さきほど自分で組み立てたリクエストURLをこのsearch.html内に組み込んで、ブラウザで表示を確認してください。



確認URL

<http://localhost/season1/work/search.html>



## 20

## コールバック関数とは

コールバック関数とは電話を後でかけなおしてもらうという意味のコールバックと意味は似ています。折り返しの電話をかけてもらうようなものです。

ここではAPIへアクセスする際に、どの関数に検索結果のデータを引数で渡せばよいかをリクエスト時に指定します。レスポンスではその指定された関数名で引数に検索結果を格納してその関数を呼び出します。

先程はコールバック関数に  
callBack=callbackFunc  
と指定しましたが、任意の関数名を指定することも可能です。

1ページで複数のコールバック関数が必要になるときは複数のコールバック関数を作成しておくといいでしょう。

▽コールバック関数とは【callback function】 - 意味/解説/説明/

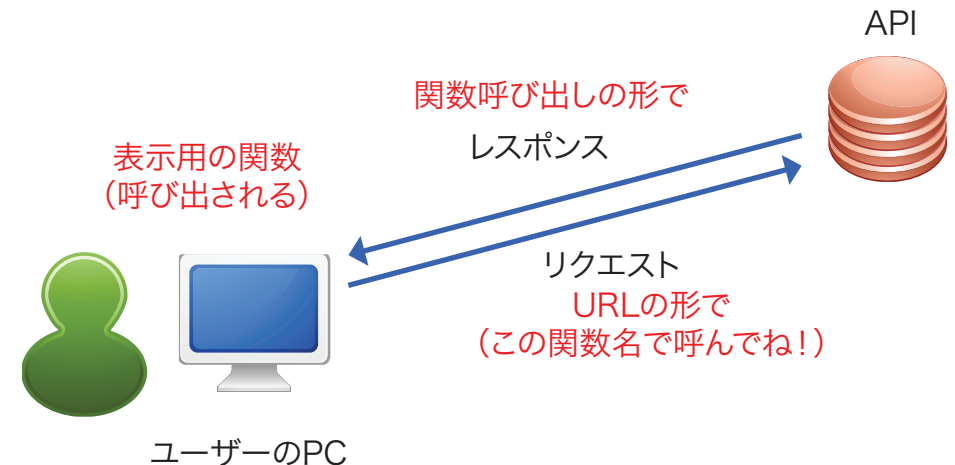
定義：IT用語辞典:

<http://e-words.jp/w/E382B3E383BCE383ABE38390E38383E382AFE996A2E695B0.html>

## ワーク

①先程のテンプレートでコールバック関数を自分で命名して書き換えて実行してみましょう。

書き換える場所は2箇所あります。



確認URL

<http://localhost/season1/work/search.html>

## 21

## 表示項目を増やすには？

表示項目を増やすにはリファレンスのレスポンスを参考にしながら、コールバック関数の中身をカスタマイズしていくことによって実現します。

例えば、コールバック関数のdocument.writeの中で次の項目を表示させたい箇所に挿入すると販売価格を追加することが出来ます。

```
items[i].itemPrice + "円<br/>\n"
```

文字列を結合するときには「+」を使います。

補足ですが、変数の中身、つまり値(文字列)を出力するときはその変数名をそのまま記述し、単純にそのまま出力する文字列は「” ”」で囲みます。初めての方は混乱しがちなので、ここを整理して記述していきます。

## ワーク

①表示させたい項目を増やしてみよう

例：

- ・商品説明文
- ・レビュー件数
- ・レビュー平均
- ・店舗名

## 22

## レイアウトを調整するには？

表示されるHTMLのイメージをつかみながら、document.writeの中に組み込んでいきます。

まずはどんな表示イメージにしたいか、をHTMLやCSSで組み立てます。(CSSは外部ファイル化していても良いが、HTMLファイルで読み込むこと)

一度仮の画像やテキストで組み立ててから、そのHTMLの文字列をdocument.writeに入れて表示を確認します。



## ワーク

- ①先程作ったsearch.htmlをコピーしてそれをベースに始めます。  
自分好みのレイアウトに調整してみましょう。  
→各自でチャレンジしてみてください(3時間目のワークなどで)



## 24

## ブログパーツとして利用する

PHPを使わずにJavascriptだけでAPIへアクセスし、商品検索結果を表示することができました。つまり、ブログパーツを配信するために特別にPHPが動くサーバーやプログラムを組む必要なくブログパーツを作成することも実現出来ます。

第1クールや第2クール参加者はPHPでJavascriptを生成したり、PHPをおくためのレンタルサーバを用意したりと手間がかかっていましたが、わずかなJavaScriptで同様なこと可能になります。

リクエストURLの工夫次第で面白いブログパーツが作成出来るので、考えてみてください。

(売れ筋順、レビュー数順、ランキング等)

APIを使ったブログパーツですので、最新の情報が常に表示されるので更新せずとも、最新情報が出続けるのがメリットです。

## ※注意事項

すべてのブログサービスに対応しているわけではありません。

楽天のJSONPを使っていると気づかれるとデベロッパーIDが流出する危険性があります。

## ワーク

①自分のブログに試しに入れてみる。

- ・サイドバー
- ・記事下広告

## データ版リリース

しました。予習などにお役立てして聞くことが出来ます。

## ブログパーツ

なまはげチョコ8個入×3袋〔個包装/アソートチョコレート/スイートビター・ミルク・ホワイト/クール便・冷蔵(冬期は常温でお届けします)〕※お届けまでに2週間ほどかかる場合がございます

1260円



ご当地ケロロ軍曹★東北なまはげ★根付ストラップ【あす楽対応】【楽ギフ\_包装選択】【楽ギフ\_メッセージ入り】02P12oct10

448円



JA秋田みなみなまはげしょうゆ 赤鬼 1000ml

326円

## 1詳細&amp;参加申込フォー

1申込フォームを設置しました。

をご参照ください。

[cool/](#)

## 25

## リンクビルダーとして活用する

意外と知られていないのですが、ブログサービスのリッチテキストエディタでは、ブラウザで表示された内容をそのままコピー出来ます。

つまりJavaScriptなどで出力されたHTMLをコピーすると、そのHTMLとまったく同じものが張り付けられます。画像もテキストもアフィリエイトリンクもです。

これを応用して、自分専用のリンクビルダーを作成しましょう。

自分専用ビルダーのメリットは

- ①自分の思い通りの表示項目
- ②自分の思い通りのデザイン・レイアウト
- ③自分の思い通りの検索条件
- ④いちいちログインしなくていい
- ⑤すぐリンク作成出来る

といった感じで、自分のブログアフィリエイトを効率化するツールを作っていくことも出来るのです。

しかもJavaScriptのみなのでレンタルサーバーもXAMPPも立ち上げ図にブラウザだけで動作するので、個人用ツールとしてつくっておくのも良いでしょう。

#### ※注意事項

すべてのブログサービス、ブラウザに対応しているわけではありません。

## ワーク

- ①自分がよく使いそうなリンクビルダーを作ってみましょう  
(自動更新ではなく記事と同時にネタとして取得したいもの)
  - ・今週のメンズファッションランキング
  - ・今週の売れ筋レギンスをご紹介!
  - ・今週のお取り寄せグルメレビュー点数高得点ランキング!

このようにAPIはブログアフィリエイトでも記事を書くネタを拾うだけでなく、アフィリエイトリンクを作る一連の流れまでも効率化するという仕組みを作ることも出来ます。

Webサイトというより、アプリケーションを作る感覚です。

※ブログサービスがリッチテキストエディタに対応していない場合は、ちょっとした改修でタグを書き出すテキストエリアを作ることが出来ます。

そのタグをコピーするという手もあります。