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、自分で使っ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と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

1 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。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。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。

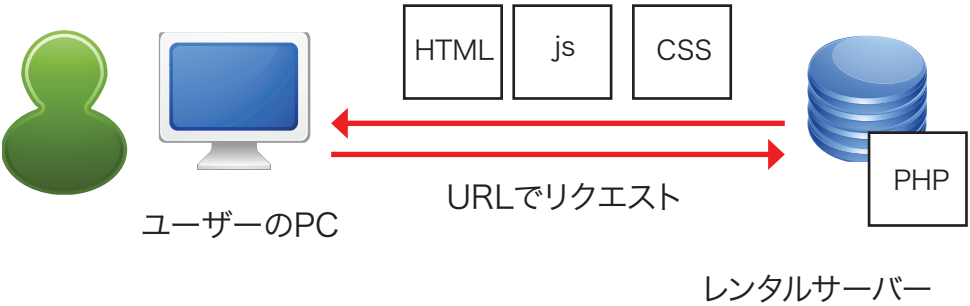
1 サーバーサイドとクライアントサイドの違い

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

1 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">
//<![CDATA[
document.write('<link rel="stylesheet" href="http:
//]]>
</script>
```

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。

<script type='text/javascript' src='http://musilog.net/wp-includes/js/prototype.js?ver=1.6.1'></script>

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使い分けていきましょう。

ワーク

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。

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。

1 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。

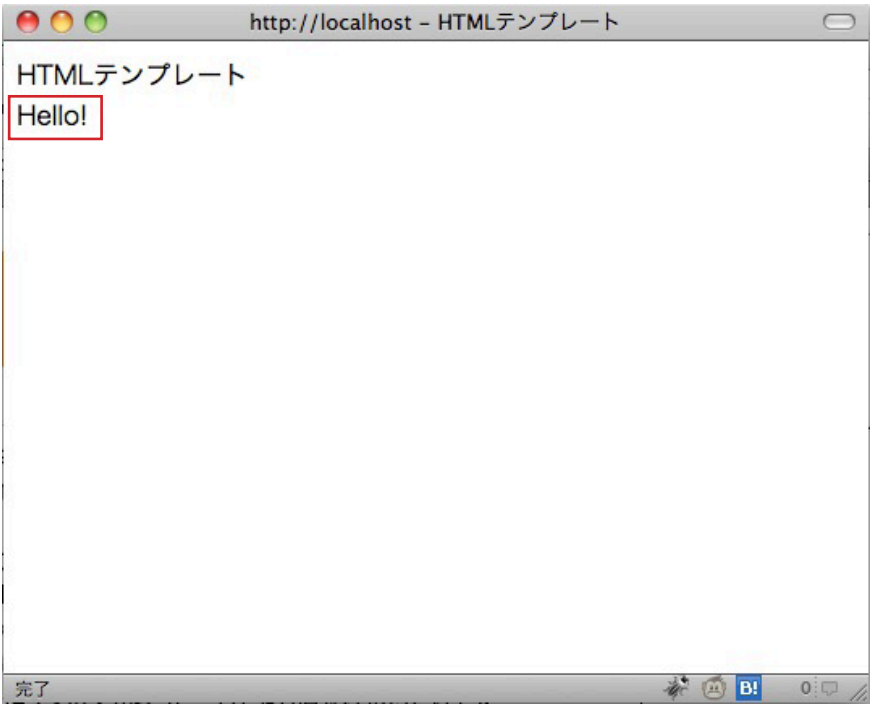
さて、first.htmlをコピーしてworkフォルダに入れます。下記記述を入れてみましょう。

```
<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。



確認URL
http://localhost/season1/work/first.html

1 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。

※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。

確認URL
http://localhost/season1/work/first.html

1 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”として表すことが出来ます。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

1 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字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);
```

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ます。

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

1 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。呼び出されたタイミングで実行されます。

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ることもできます。

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

1 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。

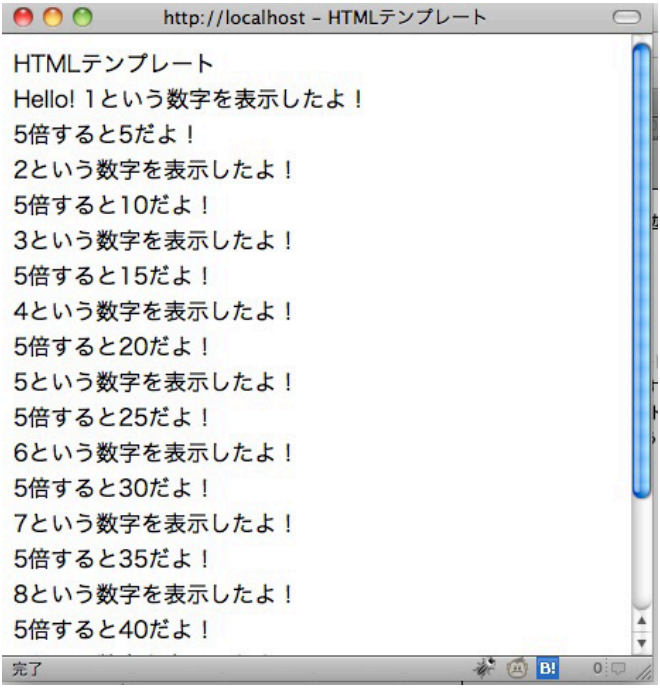
```
for (i=1;i<=10;i++) {
document.write(i+"という数字を表示したよ！<br>");
var kakezan = i*5;
document.write("5倍すると"+kakezan+"だよ！<br>");
}
```

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。
先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

1 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。

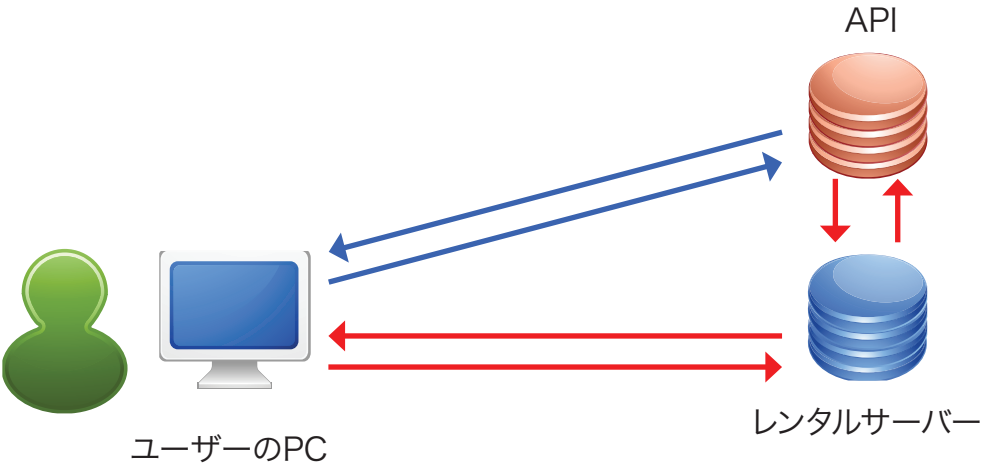
ぜひトライアンドエラーで練習を積み重ねていきましょう。

ワーク

①これまでのワークを繰り返してみる

1 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。



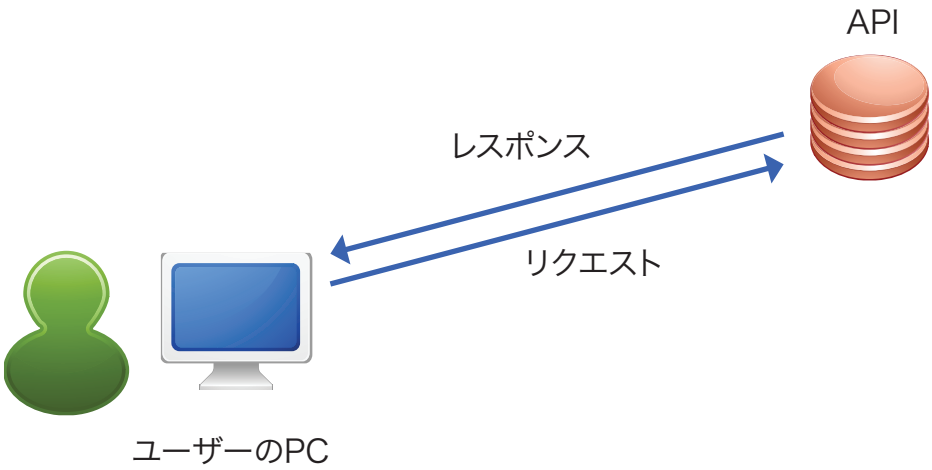
PHPとJavaScriptではデータ通信経路の違い、スクリプトが実行される場所のが違いがありますが、リクエストを行い、レスポンスを受け取るという流れは代わりません。

JavaScriptではユーザのブラウザから直接APIへリクエストを行ないます。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ることです。

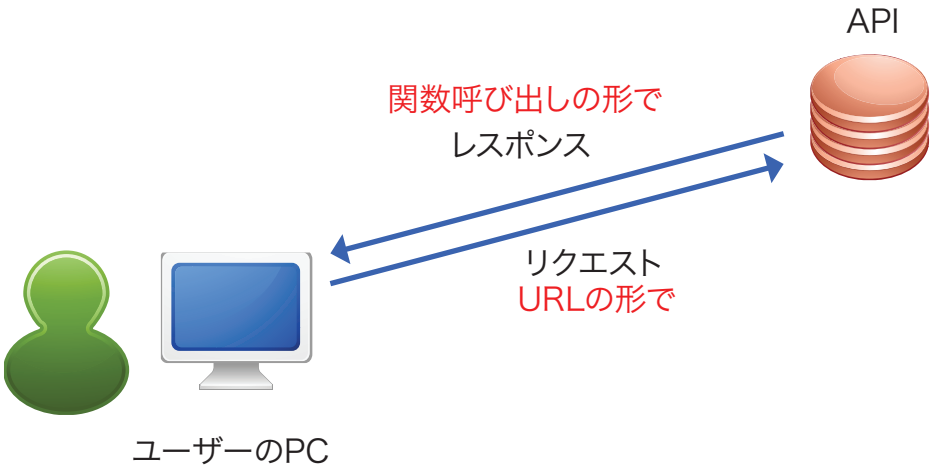


1 リクエストとレスポンスの翻訳

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JSONP形式を扱う場合について説明します。JavaScriptの関数呼び出しの形返ってきます。これをJavascriptであらかじめそれを解釈する関数を作っておき、呼び出されるように準備します。呼び出された関数はそれを解釈してHTMLとしてブラウザに出力できるように組み立てていきます。



ワーク

①p.18の関数とは？の部分を再読しておく。

　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。

1 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

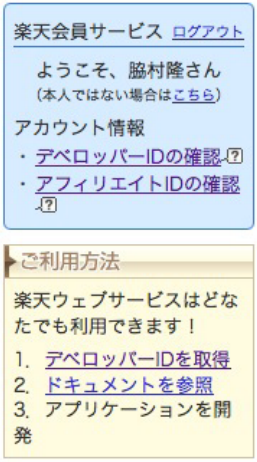
①自分の楽天デベロッパーIDと楽天アフィリエイトIDをテキストエディタなどでコピペでメモって保存しておきましょう。(いつでも参照できるように)。

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/



1 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）

と呼び名に微妙な違いはあるものの、この3要素で仕様書が構成されています。

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

1 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。

ワーク

①楽天市場商品検索で必ず引っかかりそうな商品名を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。）

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

1 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生きます。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/
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?

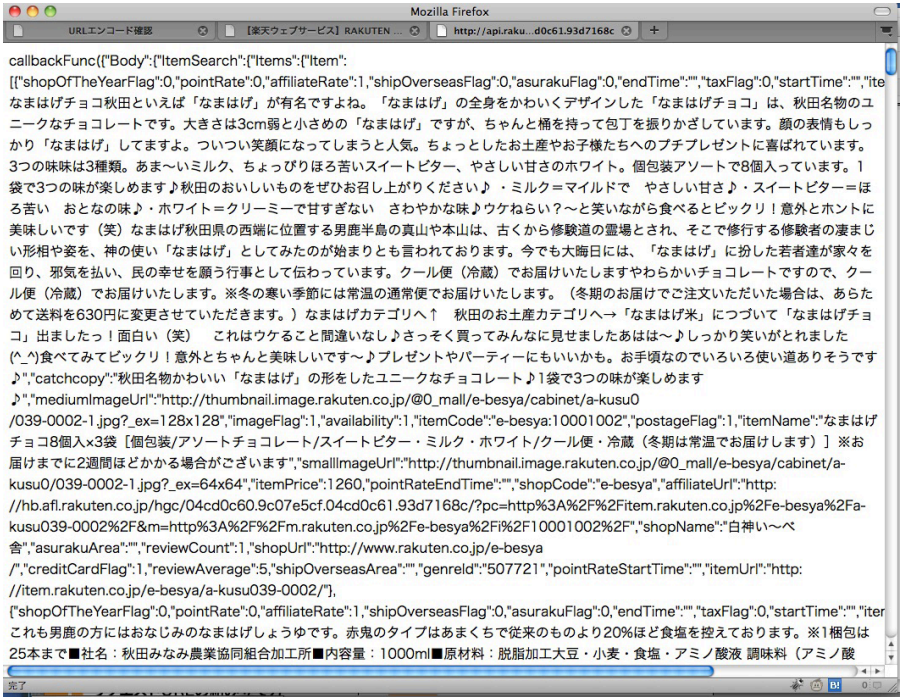
必須なパラメータは
developerId=【あなたのデベロッパID】
operation=ItemSearch
keyword=【さきほどURLエンコードした検索キーワード】
version=2010-09-15
callBack=callbackFunc
です。
アフィリエイトリンクを作成されたい方は
affiliateId=【あなたのアフィリエイトID】
とします。

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。

1 ブラウザ上でリクエストする

リクエストURLが組みあがったところで、それが問題なく正しいものかどうかを確認剃る方法として、ブラウザでそのURLでアクセスする方法があります。

ブラウザ上でリクエストすると、生のレスポンスを見ることができます。文字列が複雑に見えますが、よく見れば、商品検索結果やリンクURLなどの文字列を発見することが出来るでしょう。



ワーク

①自分が検索キーワードに意図する結果が出ているか確認する。

②長文で構造が把握出来ないが、callbackFunc(・・・で始まる関数呼び出しの形になっており、引数としてこれらの大量のデータが含まれていると確認する。
※つまりこれが引数となる関数を呼び出そうとしている。

1 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「,」でデータが繋げられている

"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。

1 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

1 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良いでしょう。

▽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。

1 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"
```

ワーク

①表示させたい項目を増やしてみよう

1 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。

ワーク

①自分好みのレイアウトに調整してみましょう。

1 フォームに入力した内容の反映

未完

ワーク

ああああ
ああああ
ああああ

1 出力されたHTMLの利用方法

未完

ワーク

あああああ
あああああ
あああああ