

実践！ワークショップ・PHPでWebサービスAPI勉強会 ＜応用編＞

2009/05/17

ムジログ wackey

<http://musilog.net/>



目次

- 事前準備・宿題
(できれば1週間前までに提出していただければ添削いたします)
 - common.phpのアップグレード(必要な方のみ)
 - 検索フォーム設置ページと検索結果ページのデザイン
 - UTF-8へ文字コードを変換する方法
 - マッシュアップの企画案を考える
- 1時間目 PHP文法スキルアップ
 - 変数の復習
 - 配列変数とは?
 - 配列変数ワークショップ
 - 繰り返し処理について(foreach)復習
 - Y!ショッピングで検索する方法の復習
 - Y!ショッピングのデータを配列変数に格納
 - 配列変数に格納されたデータを表示する
 - 配列変数の番号を指定して表示してみる
 - キャッシュ(PEAR Cache/Lite)の利用概要説明(なぜ必要か?)
 - キャッシュ(PEAR Cache/Lite)の利用(ワーク)
 - PEAR Cache/Liteがインストールされていないサーバーの場合の対処方法
 - 条件分岐(if~else)取得データが0件でない場合
- 2時間目 YAHOO!ショッピング+楽天市場の商品横断検索をしやすい順に並べてみよう!
 - 価格比較に向けた横断検索の考え方
 - 向いている検索、向いてない検索
 - 複数ECサイトのリクエストURLの組み立て
 - キャッシュ(PEAR Cache/Lite)を使う
 - レスポンスフィールドの調整(SimpleXMLが使えるように「:」削除。VC、楽天対応)
 - 違うECサイト間の共通項目を配列変数かする
 - 共通項目を配列変数へ代入→一旦そのまま表示
 - asort関数を使って安い順番に並び替える
 - 検索フォームを設置する
 - TIPS解説:画像が同じ大きさに見えるようにCSS処理する
- 課題演習
 - 自分オリジナルのデザインでオリジナルな企画の横断検索サイトを作る!
- ネット帳
 - 宿題で作成したHTMLファイルにPHPを埋め込んで動かす方法
 - 条件分岐(if~else)でこんなことができる
 - 商品検索人気キーワードランキングを表示する
 - 商品検索人気キーワードにリンクを貼り検索させる
 - 対象検索サイト・ASPを増やす
 - 送料の有無を表示をする
 - 値段に「,(コンマ)」を入れて金額を見やすくする
 - レビュー点数やレビュー記事ページへのリンクを張る
 - レビュー数or点数順に並び替える(考え中)
 - JANコードを出力してJANコードを元に検索し価格比較する

3時間目～4時間目は実際に課題演習 & フリーQ&A + 発表・解説の予定



common.phpのアップグレード

- 最後の課題演習までYAHOO!と楽天のみで行かれる方は基本編で使ったcommon.phpをそのまま使ってください。対応サービスが増えない限りは基本編のcommon.phpでもこの勉強会での互換性があります。
- 今回の応用編の課題演習でバリューコマース、リンクシェア、アクセストレードを追加される方は下記ページから、最新のcommon.phpをダウンロードし、以前の情報を転記して新しいcommon.phpを準備してください。
- http://musilog.net/webdesign/web-service/web_service_api_sample_config.php
- 既知のバグ:リクルートWebサービス入れるの忘れました。すみません<m(__)m>(位置情報編で復活させます)



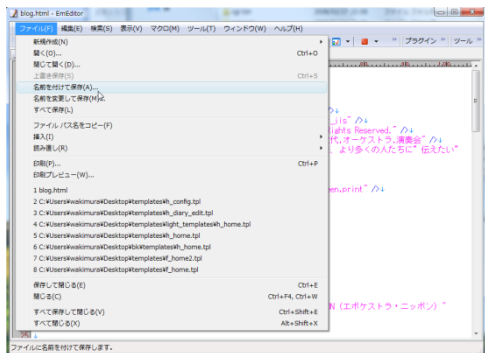
検索フォーム設置ページと検索結果ページのデザイン、HTML作成

- 応用編の課題演習ではオリジナルな横断検索ページを作成していただきます。
- 検索フォーム設置ページをform.html(トップページにされる場合はindex.html)、検索結果ページをkekka.htmlとしてデザインしてHTML作成してください。検索結果ページは実際に商品が2, 3個検索された結果イメージをダミーに入れてみてください。
- ホームページビルダーを使っても、DreamWeaverを使っても、デザインテンプレート配布サイトからダウンロードしてカスタマイズしたもので、すでにご自身が作成されたサイトのデザインでも基本編の「ぐるっと居酒屋」をパクっても(これが無難かも?)なんでも結構です。オリジナルなものを目指してください!
- テンプレート作成後、文字コードがUTF-8で無い場合はUTF-8へ変換してください(次ページで解説)。UTF-8で作らないと文字化けで苦しみます。
- 出来上がったHTMLで問題が無いか、チェックしますので、可能であれば1週間前までに基本編と同じFTPでouyou/html/というディレクトリがあるのでそこで表示が確認できるようにイメージファイル、css、jsファイル一式まとめてアップしてください。

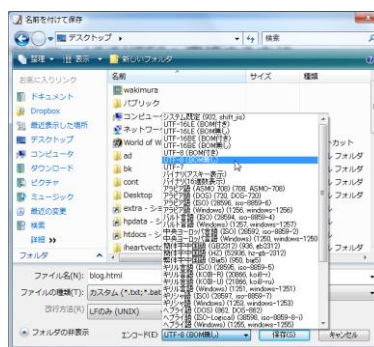


HTML、css、jsファイルをUTF-8へ変換する方法

- ①EmEditorなどでHTMLファイルを開き文字コードを変えて保存する



ファイル→名前を付けて保存
(今のファイルに上書きで良い)



保存先を決め(今のファイルの上書きで良い)
改行コードを「LFのみ(UNIX)」にし、エンコードを「UTF-8(BOM無し)」
「保存」ボタンを押し、上書き保存します。

- ②METAタグなどshift-jisなど他の文字コードになっている部分をutf-8へ変更する(css、jsも)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=shift_jis" />  
<meta http-equiv="Content-Script-Type" content="text/html; charset=shift_jis" />  
<meta name="Copyright" content="&copy; 2007-2008 EPOCHESTRA NIPPON All Rights  
Reserved" />
```

上書き保存すれば、それで完了。そのあとはFTPでアップして添削を受けてください。

マッシュアップの企画案を考える

- 課題演習で実施する自分のオリジナルなデザイン、オリジナルな切り口の横断検索サイトを作ります。その企画を考えてください。
- 質問票をテキストファイルにて添付しますので、埋めてご提出ください。
- 例えば「レディースファッションを扱っているショップに絞った横断検索サイト」「100円で買えるものの横断検索サイト」など
- 機能を追加して「人気キーワードで横断検索」「関連検索キーワードを自動的に追加してさらに横断検索」など
(基本編のスキルの応用で十分できますよ！)
- いただいた企画案を元に、できそうかできなさそうか、こうしたらどうか？という回答をさせていただければ、と思います。



1時間目 PHP文法スキルアップ

- 基本編で学んだことの再確認とその応用へ進みます。
- 基本編ではシンプルに行った商品検索表示を配列変数を経由して表示させます。
- 配列変数を経由することで、異なるECやASPのデータを配列変数で一元管理し、加工・編集が容易になります。



変数の復習

上から順番

```
13 echo "アイスコーヒーが"
14 $a = "気温は";↓
15 echo $a;↓「気温は」と出力
16 $a= 28;↓ 数値として上書き
17 echo $a;↓「28」と出力される
18 echo "度です";↓
19 ?><br />↓
```

文法

\$変数名 = “文字列”;・・・文字列として代入
(\$変数名 = 数値;・・・数値として代入)

同じページだけでしか存在出来ない。
(他のページへ引き継げない)
上書きが可能 = 上書きされる

文字列や数字を自由に出し入れできる箱と考える。
(そのページだけでしか記憶出来ない一時的なメモリ)

変数名に使える文字の種類

- ・a~z
- ・A~Z
- ・0~9
- ・_(アンダーバー)



YAHOO!のリクエストURLの
変数名を
\$ywsurl=・・・とする、等

これから複雑なWebサービスAPIを利用したアプリ
を作るには変数名の管理が重要。
意図せずに上書きされたり、変数名のスペル名を
書き間違えたり・・・変数名を見てわかる名づけを
しておきましょう。

Webアプリ開発のネタ帳

変数には大きく分けて文字列と数値を出し入れするものがある。そのうち数値は電卓のように計算が可能。つまり、ECサイトの検索結果から3個買ったらか何円とか、3人でわけたら1人あたりいくら、とか計算できる。また異なるECサイトの複数の商品を選択して合計金額概算(送料無視)を出すことも可能。



配列変数とは？

変数

\$shopname



\$shopname="ヤフー"

1個しかデータが入らない

配列変数

\$shopname

[0]
[1]
[2]
[3]
[4]
[5]
[6]

\$shopname[0]="ヤフー"
\$shopname[1]="楽天"
\$shopname[2]="Amazon"
\$shopname[3]="ベルーナ"
\$shopname[4]="nissen"
\$shopname[5]="DMM"
\$shopname[6]="Apple store"

同じ変数名の中にラベルをつけて
複数のデータを扱うことができる

ひとまどりのデータを配列変数にまとめて
入れることにより、処理しやすくなる。つまり...

配列変数とはつまり表にしてデータを整理しやすくするようなもの

	A	B	C	D
1		title	price	linkurl
2	0	新型インフルエンザ(豚インフ	3,990	http://hb.afl.rakuten.co.jp/hgc/04cd0c60.9c07e5cf.04cd0c6
3	1	新型(豚)インフルエンザウィ	63,000	http://ck.jp.ap.valuecommerce.com/servlet/referral?sid=2424
4	2	『オゾン発生機 ピュレア N!	29,400	http://ck.jp.ap.valuecommerce.com/servlet/referral?sid=2424
5	3	新型インフルエンザ(豚インフ	913	http://hb.afl.rakuten.co.jp/hgc/04cd0c60.9c07e5cf.04cd0c6
6	4	新型インフルエンザ対策に!	11,700	http://ck.jp.ap.valuecommerce.com/servlet/referral?sid=2424
7				

キーとなる配列番号
(ラベル=今日は「数字」)

商品名
\$title[]

価格
\$price[]

リンクURL
\$linkurl[]

APIから返されたデータをそのままの順番で表示するだけでなく、配列変数に格納することにより加工が可能

複数ECサイトから商品リストを同じ配列変数に順番に格納しやすい順番で並び替えることにより全ECサイトでの検索結果の安い順表示が可能になる。

Webアプリ開発のネタ帳

APIから帰ってくる値で希望通りのソート方法が無い場合はこのように一旦配列変数に入れ、アイウエオ順にするとか(あまり意味がない・・・)、安い順且つ送料無しを上位表示に優先させるなど、順番を加工して表示することが可能

配列変数文法ワークショップ

① 配列変数の代入方法と表示方法

```
<?php↓
$item[0] = "サル";↓
$item[2] = "チンパンジー";↓
$item[1] = "ゴリラ";↓
?>↓
↓
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">↓
↓
<head>↓
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">↓
<title>配列変数表示テスト用サンプル</title>↓
<meta http-equiv="Content-Style-Type" content="text/css">↓
</head>↓
↓
<body>↓
$itemの配列変数を0から2まで順番に表示する。<br />↓
0の中は<?php echo $item[0]; ?><br />↓
1の中は<?php echo $item[1]; ?><br />↓
2の中は<?php echo $item[2]; ?><br />↓
↓
```

文法

\$変数名[数字] = 文字列;
配列番号を指定して文字列を代入

echo \$変数名[数字];
配列番号を指定して取りだして出力

作業ファイル: hairetsuhensu_a.php

② 配列変数の代入方法で数字を省略する方法

```
<?php↓
$item[] = "サル";↓
$item[] = "チンパンジー";↓
$item[] = "ゴリラ";↓
?>↓
↓
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">↓
↓
<head>↓
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">↓
<title>配列変数表示テスト用サンプル</title>↓
<meta http-equiv="Content-Style-Type" content="text/css">↓
</head>↓
↓
<body>↓
$itemの配列変数を0から2まで順番に表示する。<br />↓
0の中は<?php echo $item[0]; ?><br />↓
1の中は<?php echo $item[1]; ?><br />↓
2の中は<?php echo $item[2]; ?><br />↓
↓
```

文法

\$変数名[] = 文字列;
配列番号を省略して文字列を代入
すると配列番号0から順番に代入される

作業ファイル: hairetsuhensu_b.php

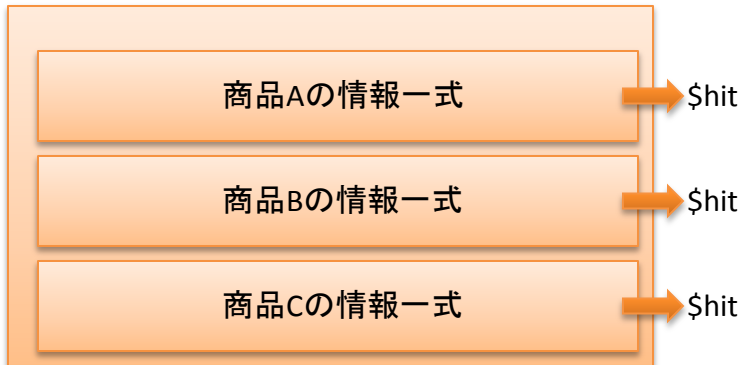
ワーク

配列変数に代入する中身の情報を書き換えたり、配列番号を書き換えたりして上記の文法を理解する

作業ファイル: hairetsuhensu_a.php hairetsuhensu_b.php

繰り返し処理について(foreach)復習

\$hits



※商品の個数分繰り返される

文法

```
foreach ($hits as $shit) {  
    echo h($shit->Name)."<br />";  
    echo h($shit->Url)."<br />";  
    echo h($shit->Image->Medium)."<br />";  
    echo h($shit->Description)."<br />";  
}
```

XMLデータを一つずつ分解して取り出すのも
配列に格納されたものを分解して取り出すのもforeach

\$price



※商品の個数分繰り返される

文法

```
foreach ($price as $key => $value) {  
    echo $itemname[$key];  
    echo $imgurl[$key];  
    echo $price[$key];  
    echo $description[$key];  
}
```

配列変数も同じように一個ずつ取り出していくことができる

やや乱暴な説明ですが上記の例では値段をキーとして
例えば安い順に配列番号を取り出し、それぞれの情報を出力している

Y!ショッピングで商品検索する方法の復習

①材料を揃える

- ・Webサービス(API)を利用するためのアプリケーションID取得
- ・アフィリエイトする場合は提携とアフィリエイトコード取得
- ・リクエストURL(ベースURL)の確認
- ・リクエストパラメータの確認(どんな条件で検索できるか)
- ・レスポンスフィールドの確認(どんな結果を得られるか)

②設定ファイル(例:common.php)の作成

- ・WebアプリケーションIDなどサイト内で共通して使うものを保存

③リクエストURLを組み立ててみる

④取得したデータをHTML化する

ソースを見て流れを確認し、
どんな方法だったか思いだしてみよう

Y!ショッピングのデータを一旦配列変数に格納

```
$hits = $xml->Result->Hit;
...
<?php foreach ($hits as $hit) { ?> ←商品一個ずつ取り出して無くなるまで繰り返す
<h2><a href="<?php echo h($hit->Url); ?>"><?php echo h($hit->Name); ?></a></h2>
<?php } ?>
```

基本編では商品情報の詳細(タイトル、リンクなど部品取り出し)はHTML表示部分で処理していた

```
$hits = $xml->Result->Hit;
foreach ($hits as $hit) { ←商品一個ずつ取り出して無くなるまで繰り返す
$itemname[] = h($hit->Name);
$linkurl[] = h($hit->Url);
$imgurl[] = h($hit->Image->Medium);
$price[] = h($hit->Price);
$description[] = h($hit->Description);
}
...
<?php foreach ($hits as $hit) { ?>
<h2><a href="<?php echo h($hit->Url); ?>"><?php echo h($hit->Name); ?></a></h2>
<?php } ?>
```

商品名やリンク先URLなどのそれぞれの部品を取り出し
それぞれ配列変数に一旦格納する
(ここではよくつかわれそうなものを上げています)

配列変数に一旦格納することにより、
取り込んだデータを加工してから表示できる

配列変数に格納されたデータを表示する

```
<p><?php foreach ($hits as $hit) { ?>
<div class="Item">
<h2><a href="<?php echo h($hit->Url); ?>"><?php echo h($hit->Name); ?></a></h2>
<p><a href="<?php echo h($hit->Url); ?>"></a><?php echo
h($hit->Description); ?></p>
<br clear="all" />
</div>
<?php } ?>
</p>
```

基本編では\$hitsに格納されていたものを\$hit(商品単位)に分解していたがすでに分解されているので、その分解されたものをひとつずつ取り出していく

```
<p><?php foreach ($price as $key => $value) { ?> ←今回値段を軸として並び替えて表示するので$priceを使っています。
<div class="Item">
<h2><a href="<?php echo $linkurl[$key]; ?>"><?php echo $itemname[$key]; ?></a></h2>
<p><a href="<?php echo $linkurl[$key]; ?>"></a> <?php echo
$price[$key]; ?>円<br /> <?php echo $description[$key]; ?></p>
<br clear="all" />
</div>
<?php } ?>
</p>
```

※いろんな書き方がありますが、今後のためこの表記で統一します。

それぞれの配列変数に入れられたデータを
一個ずつ取りだしてそれぞれを表示させる

完成ファイル: yahoo_hairetsu.php

配列変数の番号を指定して表示してみるワーク

ワーク

- ①3番目のもの、とか5番目のものとか表示したいアイテムを1個決めてください
- ②現在表示されている商品の名前や写真を覚えてください。
- ③それだけ表示してください

やり方は「<?php foreach (**\$price as \$key => \$value**) { ?>」と「<?php } ?>」の繰り返し処理のタグを外し、それぞれの配列変数の変数で指定されている\$**key**の部分を表示したいアイテムの配列番号にするだけです(が落とし穴に注意)。

配列変数やプログラミングが初めての方が
ハマりやすいワナに気をつけよう

作業ファイル: yahoo_hairetsu.php



条件分岐(if~else)取得データが0件でない場合

キャッシュのところでも出てきたif~else。
これは条件に応じて動作を変えるものであり、今後のWebアプリケーションを作成するときにはぜひ、覚えておきたい。
例として、リクエストしたデータの検索結果が0だった場合の条件分岐文を追加説明として書いておきます。

```
$xml = simplexml_load_string ($urlBuff);
```

```
if ($xml["totalResultsReturned"] != 0) { //検索件数が0件でない場合、変数$hitsに検索結果を格納します。  
$hits = $xml->Result->Hit; ←検索結果が0件の時だけ実行される。0件の時に実行されたらエラーになってしまう  
}
```

!= …左と右の内容がイコールで無い
== …左と右の内容がイコールである

\$xml["totalResultsReturned"]って？

```
<ResultSet xsi:schemaLocation="urn:yahoo:jp:itemSearch http://shopping.yahooapis.jp/ShoppingWebService/V1/itemSearch.xsd"  
totalResultsAvailable="13437" totalResultsReturned="20" firstResultPosition="1">  
<Result>  
</Result>  
</ResultSet>
```

レスポンスフィールド

商品検索Web APIのスキーマの記述は、以下のURLをご覧ください。
<http://shopping.yahooapis.jp/ShoppingWebService/V1/itemSearch.xsd>

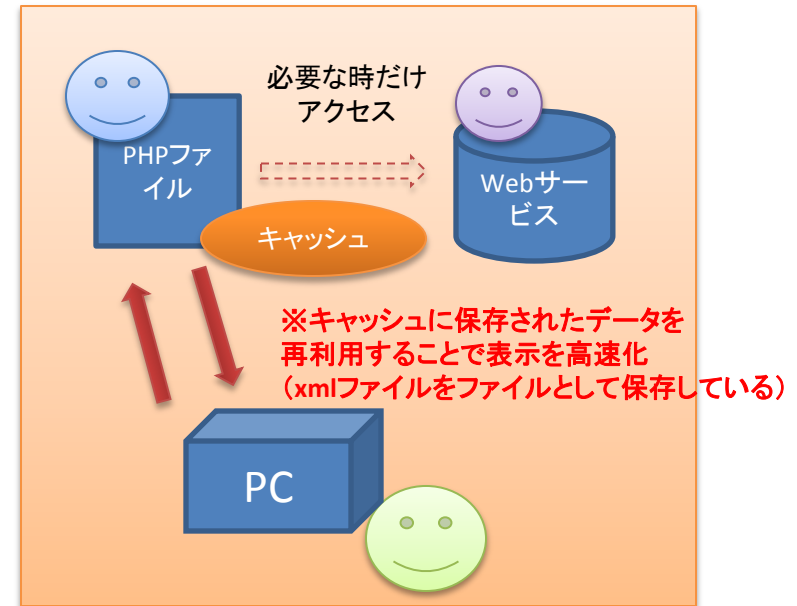
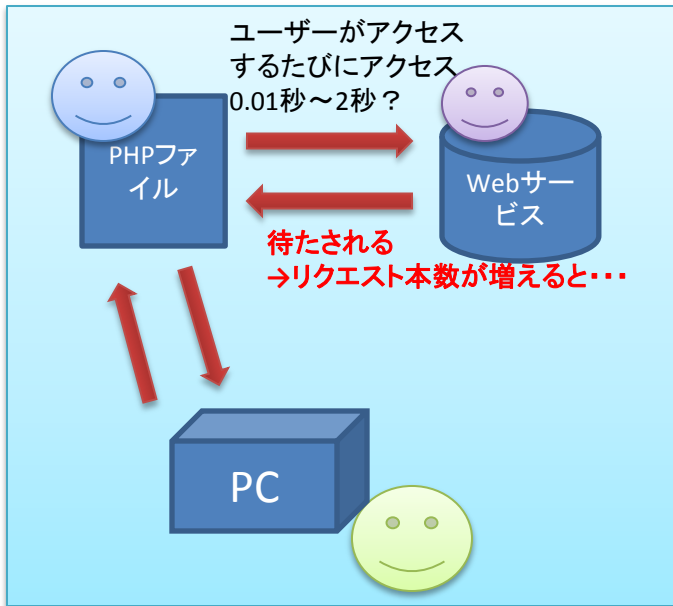
フィールド	説明
ResultSet	クエリレスポンスのすべてを含み、次の属性を持ちます。 <ul style="list-style-type: none">totalResultsAvailable: 検索数+1件数totalResultsReturned: 返された値の個数firstResultPosition: 最初のデータが何個目にあたるか(最初=1)

\$xml["totalResultsReturned"]は数字ではないが、XMLファイルを取り出すときに名前前で指定して取り出す配列変数

Webアプリ開発のネタ帳

送料のフラグを見て条件分岐し、「送料アリ」「送料無し」の表示をすとか、「送料無し」のみ表示とかが出来る。
検索結果は〇〇件です、と表示できる。
ページ番号を解釈して「次のページへ」と検索結果を深掘りさせることができる

キャッシュ(PEAR Cache/Lite)の利用概要説明(なぜ必要か?)



1. キャッシュに保存されたデータを再利用することで表示を高速化
2. Webサービスにアクセスする負荷を低減するだけでなく
Webサービスへのアクセス回数を減らし規約違反に抵触する危険性を防ぐ
3. Webサービスが重くなったとき(夜間など)に左右されにくいサイトを作る

キャッシュを利用することにより無駄なリクエスト減らす
負荷軽減および表示速度の高速化が可能

キャッシュ(PEAR Cache/Lite)の利用(ワーク)

ワーク

コピペしながら解説します。みなさんも一緒に手を動かしながら流れを覚えてください(以後、基本はコピペです)

• 準備編

- キャッシュさせるフォルダ(cache)を作る(workフォルダ以下に作成済です)
- そのフォルダを書き込み可能にする(Windowsならcacheフォルダのプロパティ読み込み専用のチェックを外す。Webサーバ上ならパーミッションを777など適切な値にする)
- キャッシュフォルダ名の指定とキャッシュ時間を決める(common.phpに書き込み済み)

• PHPプログラム編

- require_once 'Cache/Lite.php';を追記
- キャッシュID(ファイル名みたいなもの)を設定。ユニークな名前にするためにはリクエストURLがユニークになるので、それを指定する
- new Cache_Liteでキャッシュの利用を可能にする
- キャッシュの有無を判断し条件分岐。有ればcacheフォルダにあるファイルを読みこむ。無ければリクエストURLをもとにAPIにアクセスしデータを取得。同時にxmlデータをcacheフォルダへ保存。
- Simpleloadxmlで今まで直接商品データを読みこんでいたが、一旦保存されたxmlデータを分解して、変数に代入
- 以下、同じ作業でOK

作業ファイル:yahoo_hairetsu.php 完成ファイル:yahoo_cache.php



PEAR Cache/Liteがインストールされていないサーバーの場合の対処

- サーバー管理者にお願いする
→サーバーのポリシーによっては入れてくれない
- 自分で入れる(PEARからダウンロードする)
- パスを通して使う。PHPファイルの頭に下記のように入れる
例:

```
set_include_path(get_include_path() .  
PATH_SEPARATOR . "../PEAR");
```
- ※時間の余裕のあるときに詳細解説文章を作りますが、そこそこPHP利用実績のあるレンタルサーバーなら大丈夫だと思います(CORESERVERでは確認済)。

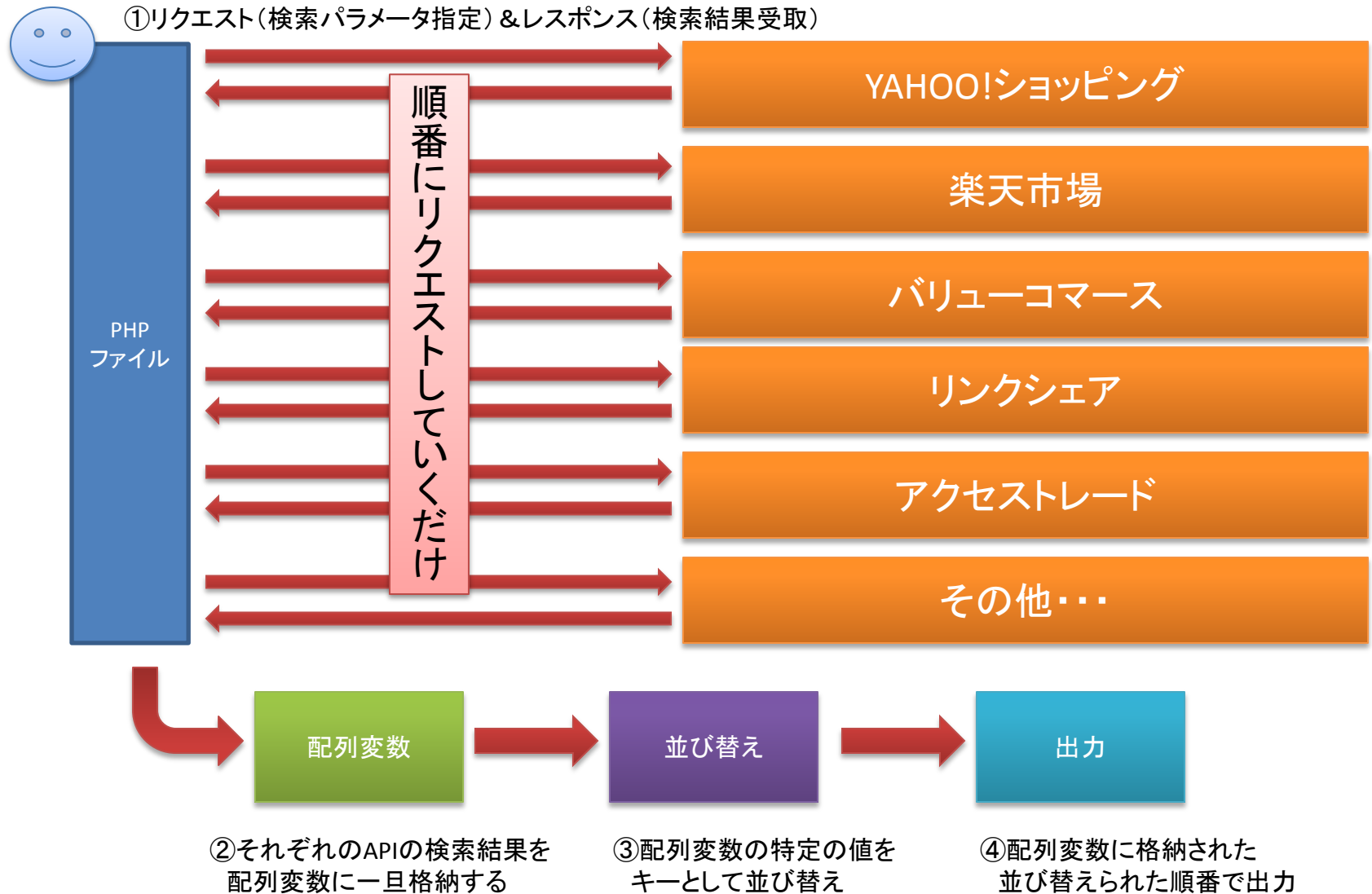


2時間目 Y!+ 楽天の商品横断検索で安い順に並べてみよう!

- 1時間目でYAHOO!ショッピングで一旦商品名などの部品を配列変数に入れて格納しました。2時間目では同じ配列変数に楽天市場商品検索の結果も追加して入れます。
- 同じ配列変数に一旦格納したので、「値段」という軸で改めて並べ替えます。



価格比較に向けた横断検索の考え方



複数ECサイトのリクエストURLの組み立て

原則：変数名を区別しやすいように書き換える、決める。

基本的に1時間目で行ったYAHOO!ショッピング検索結果を元に進めます。

これに楽天を追加するには

楽天のリクエストURL作成→リクエスト→キャッシュ処理→配列変数に格納

を追加するだけで良いですが、処理が複雑になると混乱するので、それぞれの変数名をそれぞれのAPIの名称がわかるようにつけ直します。

	YAHOO!ショッピング	楽天市場	リンクシェア	バリューコマース	アクセストレード
接頭辞(頭につける文字)	yws	rws	ls	vc	at
リクエストURL	\$ywsurl	\$rwsurl	\$lsurl	\$vcurl	\$aturl
キャッシュ用変数	\$ywsCache	\$rwsCache	\$lsCache	\$vcCache	\$atCache
XMLファイル一時格納変数	\$ywsBuff	\$rwsBuff	\$lsBuff	\$vcBuff	\$atBuff

ワーク①

1時間目で配列変数に格納するところまで作ったファイルを元に作業します。

まずYAHOO!用のものと区別できるように\$url→\$ywsurl、\$urlCache→\$ywsCache、\$urlBuff→\$ywsBuffと変更変更して正しく動くか確認すること。

完成ファイル:oudan1.php

ワーク②

- ① \$ywsurlを定義しているところの前の行にコメントを入れる「//ここからYAHOO!処理」
- ② 「//ここからYAHOO!処理」から配列変数に格納指定しているところをコピーしてその処理の次に入れる
- ③ コピーしたところに「//ここから楽天処理」と書き換え、それぞれの変数をywsからrwsに書き換えます。
- ④ 楽天商品検索リクエストURLを組み立てます(\$rwsurl)、一旦エラーが出ますが、\$rwsurlをechoで表示させてリクエストURLが正しいか確認してください(次のページへ続く) 完成ファイル:oudan2.php



レスポンスフィールドの調整(つづき)

```
<Arg key="keyword" value="生キャラメル">
</Arg>
<Status>Success</Status>
<StatusMsg/>
</header:Header>
-<Body>
  -<itemSearch:ItemSearch>
    <count>897</count>
```

楽天市場やバリューコマースのWebサービスではXMLファイルの中に「:」を含んだ要素名がある。これが入っているだけでsimpleXMLが処理できなくなります。そのため、XMLファイルの中の「:」を外すおまじないを入れる。(単純に「:」削除だとURL自体もおかしくなってしまうので注意)

ワーク③

simpleXMLで処理する前におまじない設置(「:」削除)

```
$rwsBuff = str_replace('header:Header', 'headerHeader', $rwsBuff);
```

```
$rwsBuff = str_replace('itemSearch:ItemSearch', 'itemSearchItemSearch', $rwsBuff);
```

ワーク④

(if文無かったことにする。すいません、ちょっと今回パスもしくはあとまわし)

配列変数へ書き込む処理を書き込む

表示を確認する。Y!→楽天の順番に表示されています(それぞれデフォルトの表示順序、おすすめ順などで表示されています)

完成ファイル:oudan3.php

simpleXMLで処理できないXMLファイルとして
「:」が要素名に含まれているものがある

asort関数を使って安い順番に並び替える

配列変数\$priceに入れてある値段情報の数値を軸としてデータの格納順番を安い順番に入れ変えます(配列番号もろとも入れ替わります)

文法

```
asort($price);
```

asort で昇順並び替え
小さい値から大きい値の順番 例:安いもの順

arsort で降順並び替え
大きい値から小さい値の順番 例:レビュー件数多い順

ワーク⑤

asort(\$price);を楽天の配列変数が格納し終わった場所に挿入。
楽天とヤフー合わせて、Yahoo!ショッピングと楽天の垣根を超えて安い順番に並んでいるか確認。

注意・表示する部分と同じ配列変数を軸とすること

今回値段順を前提に作っていますが、仮に\$reviewcountの多い順から並べるとしたら、この部分の修正以外にForeachのキーとなる配列変数の変更が必要(\$priceを\$reviewcountに変更)。

完成ファイル:oudan4.php

配列変数に格納すれば、関数ひとつで
思い通りの順番に並び替えて表示できる！

検索フォームを設置する

基本編でやったようにフォームを作成し、それで検索してみるものを作ろう

たとえば

○検索フォーム

```
<form action="oudankekka.php" method="get">  
<input type="text" name="word" value="" />  
<input type="submit" value="商品検索" />  
</form>
```

検索結果を表示するページ
(検索パラメータを受け取るPHPファイル名)

○検索結果側

```
$word = "生キャラメル";だったのを  
$word = $_GET["word"];でフォームに入力された値を受け取る。
```

完成ファイル: oudanform.php oudankekka.php

基本編の復習。
検索条件を複数にするにはパラメータを増やせば良い

- 自分オリジナルのデザインでオリジナルな企画の横断検索サイトを作る！

- より、本格的に見せるためのネタや小技などをまとめて書いておきます。
(企業秘密以外)
演習での作成や今後の作成にお役立てください。

宿題で作成したHTMLファイルにPHPを埋め込んで動かす方法

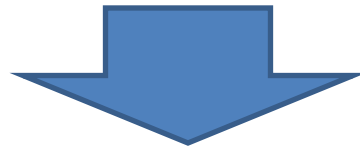
- 前提としてUTF-8で作られているか、p.4の文字コードUTF-8への変換を行ってください
- それぞれの拡張子をphpにします
- 検索結果ページはp.25で作った結果ページのコードをコピーで埋め込みます。
まずhtmlの先頭より上にp.25で作ったものと同じようにPHPコードを埋め込みます。
また結果を表示させたい部分も同じようにp.25で作ったコードを埋めます(とりあえず)
- 検索フォームページはp.25で作ったformタグ間を入れたい場所にコピーで入れます



値段に「,(コンマ)」を入れて金額を見やすくする

- 「1000円」ではなく「1,000円」

```
<?php echo $price[$key]; ?>円
```



```
<?php echo number_format($price[$key]); ?>円
```

number_format();関数を使うことにより
数字を読みやすくする

APIから取り出す部品を増やす(送料の有無表示を追加する)

2時間目では下記部品のみAPIからデータを取り出した。

```
$itemname[]  
$linkurl[]  
$imgurl[]  
$price[]  
$description[]
```

これに送料を\$souryou[]加える

○ヤフーの場合

```
$souryou[] = h($hit->Shipping->Name);
```

○楽天の場合(次ページ参照)

```
if ($hit->postageFlag == 0) {  
    $souryou[] = "送料無料";  
} else {  
    $souryou[] = "送料別";  
}
```

○送料情報が取り出せないAPIの場合

```
$souryou[] = "";
```

←送料が取り出せないAPIの場合は中身が「からっぽ(null)」でも良いので配列変数に何か入れる
※入れないと変数ごとの配列の数が異なり、存在しない配列番号の配列変数を呼び出しエラーとなる

表示方法はHTML中の表示させたい場所に

```
<?php echo $souryou[$key]; ?>  
と入れる
```

条件分岐(if~else)でこんなことができる

- 送料の有無が「0」か「1」でレスポンスが帰ってくる楽天市場検索の場合
(前提として各APIで\$souryouという配列変数を追加)

```
if ($hit->postageFlag == 0) {  
    $souryou[] = "送料無料"; ←もし送料パラメータpostageFlagが0ならば・・・  
} else {  
    $souryou[] = "送料別"; ←じゃなければ  
}
```

<?php echo \$souryou[\$key]; ?>で送料情報を表示することができる

配列変数に入れる前に入っているデータにより
表示したい内容を配列変数に入れる(違いの吸収)

画像サイズバラバラ対策

ECサイトやAPIにより画像サイズはまちまち。
仮に160px × 160pxに固定しているかのように見せかけるためには
160px × 160pxの固定サイズ枠付きボックスを作成し、その背景にセンタリング表示する。



120px × 160pxなど縦横比が違うものでもセンタリングすれば
そのサイズの写真と認識する



160px × 160pxちょうどの場合にはジャストフィットサイズ



160px × 160pxを超えるサイズは背景にセンタリングすることにより
その画像の中央部分がトリミングされたかのように見える。
(実際はトリミングも拡大縮小も行われていない)

対象検索サイト・ASPを増やす

- ヤフーのみから楽天を増やしたようにコピペで同じことを繰り返すだけ
- その時に変数名の書き換えとそのAPI向けのリクエストURLの組み立てが必要となる
- 以下、増やしたいだけ繰り返す(5個以下推奨)



商品検索人気キーワードランキングを表示する



レビュー点数やレビュー記事ページへのリンクを張る

- レビュー点数 (`$reviewcount[]`) とレビュー平均点 (`$reviewavr[]`) レビュー記事URL (`$reviewurl[]`) を追記
- ヤフーの場合
`$reviewnum[] = h($hit->Review->Count);`
`$reviewavr[] = h($hit->Review->Rate);`
`$reviewurl[] = h($hit->Review->Url);`
- 楽天の場合
`$reviewnum[] = h($hit->reviewCount);`
`$reviewavr[] = h($hit->reviewAverage);`
`$reviewurl[] = "";` ←レビューURLが存在しないため空を代入



レビュー数or点数順に並び替える

- `arsort($reviewcount);`
で並び替える
- 表示部分の繰り返し開始部分
`<?php foreach ($price as $key => $value) { ?>`
を
`<?php foreach ($reviewcount as $key =>`
`$value) { ?>`
とする。

