



CMP実習2

DB+PHP+XML/JSON+JavaScript

中村, 宮下, 齊藤, 菊池



PHPとJavaScript連携

- サーバとクライアントをどうやって繋げるか？
- PHPとJavaScript間の情報のやりとりを行う

JavaScript

JSON/XML

PHP

DB

- 簡易的なWeb APIを作ろう！
 - PHPにGETで情報を送り込むことでXMLまたはJSONを出力する
 - AjaxでPHPに対して要求する
 - JavaScriptでXMLまたはJSONを処理する



Web API とは？

- Web API
 - Web上でアクセス可能なAPI
 - 様々な情報にアクセスすることが可能
 - 何かの緯度経度, キーワード検索結果, 画像検索結果, 商品検索, 書籍検索, ブックマーク数, 地図, 形態素解析, アニメ検索, Facebook, Twitter, メールなどなど
 - 一般的なWeb APIではURLで情報を取得

<http://example.jp/search?query=test&area=10&...>

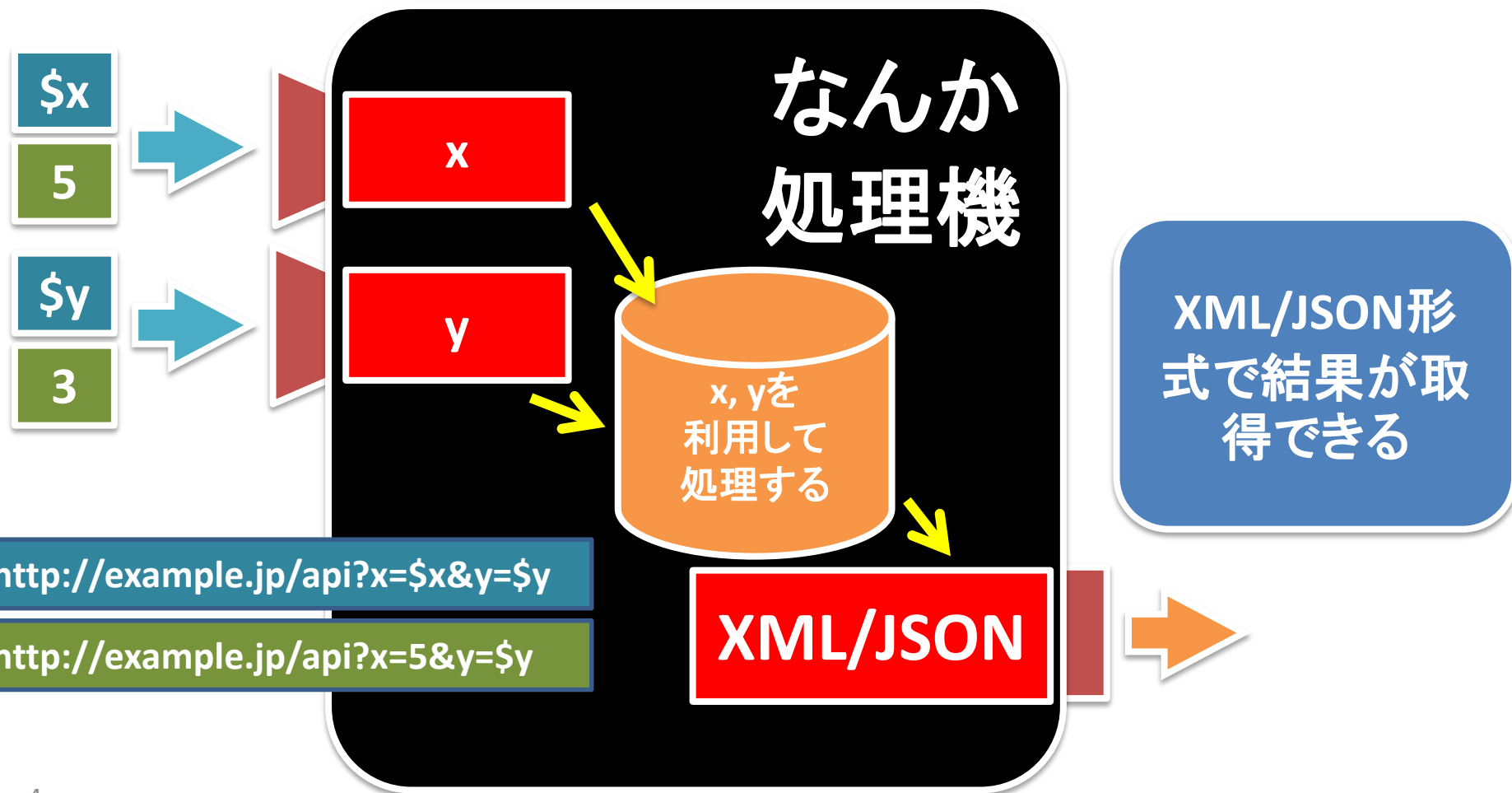
ベースURL

query=test

area=10

Web APIの内部処理

- 複数の引数を受け取ることが可能





演習

- いつものサイトに下記DBを置きました
 - Mt100u.sqlite (UTF-8版)
 - Mt100s.sqlite (Shift JIS版)
- sqlite3 Mt100u.sqlite で中身を見てみよう
 - create table mountain (id integer, name text, yomi text, height integer, pref text, city text); でテーブルが作成されています
- 100名山のテーブル
 - 出典：日本の百名山
<http://yamadarake.web.fc2.com/yama/list/nihon100.html>



単にテーブルで表示

```
<?php
// 文字コードに応じて u または s を切り替えて下さい
$db = new PDO("sqlite:Mt100s.sqlite");
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);

$rows = $db->query("select * from mountain order by height desc");
echo "<table>";
while( $cols = $rows->fetch() ){
    echo "<tr><td>";
    echo $cols["name"];
    echo "</td><td>";
    echo $cols["height"];
    echo "</td></tr>";
}
echo "</table>";
?>
```

[return_html.php](#)



XML化

```
<?php
// 文字コードに応じて u または s を切り替えて下さい
$db = new PDO("sqlite:Mt100u.sqlite");
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);

// 文字コードに応じて UTF-8 または Shift_JIS を切り替えて下さい
$rootNode = new SimpleXMLElement(
    "<?xml version='1.0' encoding='UTF-8'?><items></items>" );

$rows = $db->query("select * from mountain order by height desc");
while( $cols = $rows->fetch() ){
    $itemNode = $rootNode->addChild("item");
    $itemNode->addChild("name", $cols["name"] );
    $itemNode->addChild("height", $cols["height"] );
}

// XMLとして出力する
echo $rootNode->asXML();
?>
```



JSON化

```
<?php
// JSONの場合はutf-8のみ
$db = new PDO( "sqlite:Mt100u.sqlite" );
$db->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING );

$rows = $db->query( "select * from mountain order by height desc" );
/items = array();
while( $cols = $rows->fetch() ){
    // arrayの中身を増やせば色々な情報を出力可能
    // ポイントは「名前 => 値」とすること
    $items [] = array( "name"=>$cols["name"], "height"=>$cols["height"] );
}

// JSONで出力するという事をヘッダで明示する
header( 'Content-type: application/json' );
echo json_encode($items);
?>
```

[return_json.php](#)



演習

- XMLおよびJSONで出力する結果に、百名山の読み (yomi) と県 (pref) の情報を追加しましょう
- PHPで出力したXMLまたはJSONをJavaScriptで処理して表示してみよう！



APIにするってどういうこと？

- 基本的にGET/POSTメソッドで何かの情報を送りつけて、その結果をXMLかJSONで返せばOKということ！
- (例)
 - type=json なら json で, type=xml なら xml で返す
 - pref=キーワード で都道府県名で検索
 - method=view で単純に表示, method=search で検索して表示, method=insert で登録
 - limit=10 で10件, limit=100 で100件取得
 - sort=desc なら降順で, sort=asc なら昇順で取得



[http:// ... /return_xml?pref=長野](http://.../return_xml?pref=長野)

```

<?php
// 文字コードに応じて u または s を切り替えて下さい
$db = new PDO("sqlite:Mt100u.sqlite");
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
if( isset( $_GET["pref"] ) ){
    $query = "select * from mountain where pref like '%" . $_GET["pref"] . "%'";
}
// 文字コードに応じて UTF-8 または Shift_JIS を切り替えて下さい
$rootNode = new SimpleXMLElement(
    "<?xml version='1.0' encoding='UTF-8'?><items></items>" );

$rows = $db->query( $query );
while( $cols = $rows->fetch() ){
    $itemNode = $rootNode->addChild("item");
    $itemNode->addChild("name", $cols["name"] );
    $itemNode->addChild("height", $cols["height"] );
    $itemNode->addChild("pref", $cols["pref"] );
}

// XMLとして出力する
echo $rootNode->asXML();
?>
    
```

return_xml.php



演習

- `type` で `json` と `xml` を切り替える事ができるようにしてみよう
- `hsort` で高さの降順, 昇順で並び替えることが出来るようにしてみよう
- `from`と`to`で何`m`以上, 何`m`以下という条件を指定できるようにしてみよう
- `limit` で表示件数を指定するようにしてみよう



演習

- テキストボックスを用意し，文字が入力される度にその山の場所がどこにあるかという文字を含む結果を取得および表示するプログラムを作成せよ
- テキストボックスを用意し，数値が入力される度にその値より大きな山だけ表示するプログラムを作成せよ



文字でキーワード検索する例

```
<html>
<head> <script src="../../lib/jquery-2.1.1.min.js"></script>
<script>
$(function(){
$("#pref").on("keyup",function(){
var reqURL = "api.php?search=pref&pref=" + encodeURIComponent($("#pref").val());
console.log( reqURL );
$.ajax({
url: reqURL,
dataType: "json",
success: function(data){
console.log( data );
$("#results").html("");
$("#results").append("<table>");
$("#results").append("<tr><th width=80px>山名</th><th width=80px>高さ</th><th width=80px>都道府県</th></tr>");
for( var i=0; i<data.length; i++ ){
$("#results").append("<tr><td>" + data[i].name + "</td>" + "<td>" + data[i].height
+ "</td>" + "<td>" + data[i].pref + "</td></tr>");
}
$("#results").append("</table>");
}
})
});
</script>
</head>
<body>
<input type=text id=pref>
<div id="results"> </div>
</body>
</html>
```

search.html



```
<?php
// JSONの場合はutf-8のみ
$db = new PDO("sqlite:Mt100u.sqlite");
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
$items = array();

if( isset( $_GET["pref"] ) && !empty( $_GET["pref"] ) && $_GET["search"] === "pref" ){
    $query = "select * from mountain where pref like '%" . $_GET["pref"] . "%'";
    $rows = $db->query( $query );
    while( $cols = $rows->fetch() ){
        $items [] = array( "name"=>$cols["name"], "height"=>$cols["height"], "pref"=>$cols["pref"] );
    }
}

// jsonで出力するという事をヘッダで明示する
header('Content-type: application/json');
echo json_encode($items);
?>
```

api.php



API化のメリット

- しっかりしたAPIさえ用意しておけば, 色々なものから活用することが可能
- すべての結果を取得するのではなく部分的に取得して利用
- Ajaxを使った画面遷移のないページを作ろうと思った時には必須
- ブラウザ拡張などを実装するときには必須!