

1 Ajax-->重点&难点

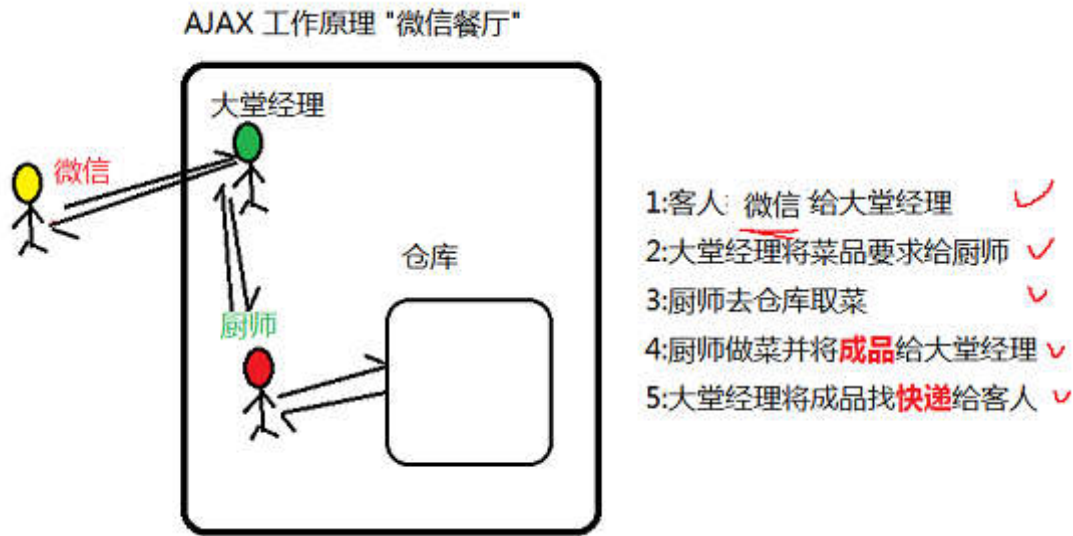
1.1:快速入门 ajax

添加用户示例

a:创建表:在数据库创建用户表 `huimaiche user`

b:创建php: `add_user.php` (获取参数/创建sql/返回添加成功/失败)

c:创建 `html:add_user.html`



```
<?php
//1: 获取参数 uname upwd
@$uname = $_REQUEST['uname'] or die('用户名是必须的');
@$upwd = $_REQUEST['upwd'] or die('密码是必须的');
//2: 创建与数据库连接
//3: 设置编码
require("init.php");
//4: 创建sql 并且发送结果 INSERT ...
$sql = "INSERT INTO user VALUES(null,'$uname','$upwd')";
$result = mysqli_query($conn,$sql);
//5: 判断并且返回成功或者失败
if($result===true){
    echo "添加成功";
}else{
    echo "添加失败";
}
?>
```

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <h1>ajax 添加用户</h1>
```

```

<form action="#">
  用户名<input type="text" id="uname" /><br/>
  密码<input type="password" id="upwd" /><br/>
  <input id="btn" type="button" value="添加" />
</form>
<script>
  // *1: 获取btn按钮
  // *2: 为其绑定点击事件
  btn.onclick = function(){
    // 2.0: 获取用户名和密码
    var u = uname.value;
    var p = upwd.value;
    // 2.1: 创建对象XHR 16:35--16:45
    var xhr = new XMLHttpRequest();
    // 2.2: 绑定事件onreadystatechange
    xhr.onreadystatechange = function(){
      alert(xhr.responseText);
    }
    xhr.open( 'GET',
      'add_user.php?uname='+u+'&upwd='+p, true);
    // 2.4: 发送请求
    xhr.send(null);
  }
</script>
</body>
</html>

```

1.2 标准语法: XHR 发送请求并且接收返回数据(重点)

(1) 创建XHR对象

```
var xhr = new XMLHttpRequest();
```

(2) 为XHR绑定事件，状态改变事件

```

xhr.onreadystatechange = function(){
  alert(xhr.responseText); // 接收服务器程序返回数据
}

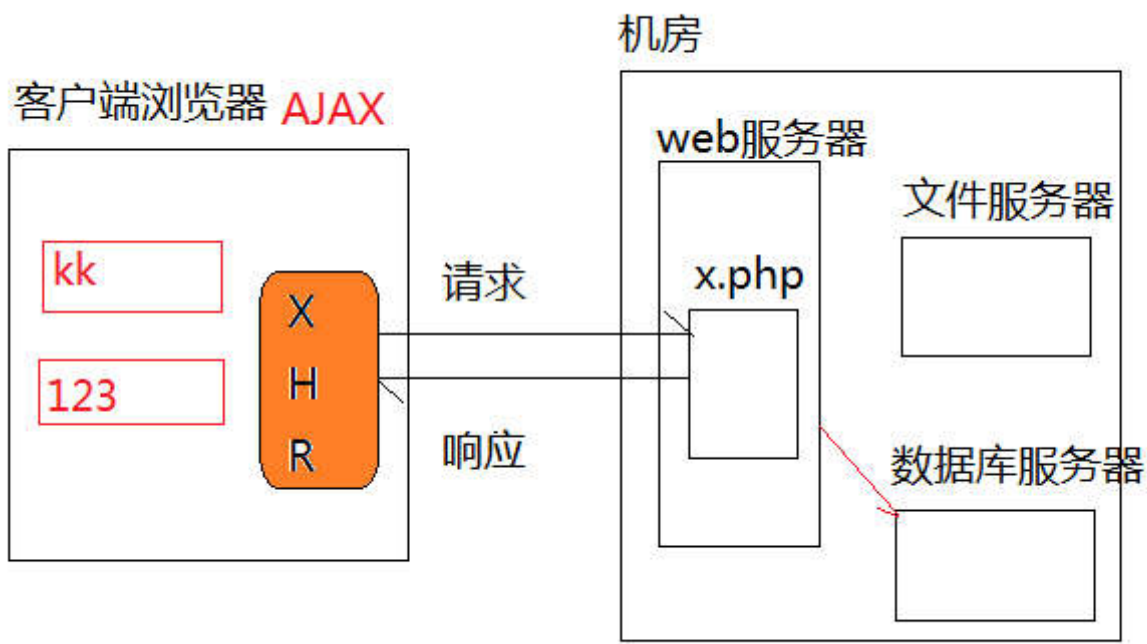
```

(3) 创建一个与服务器程序连接

```
xhr.open(请求方法, 请求地址, 是否为异步请求)
```

(4) 发送请求

```
xhr.send(null);
```



1.3:概述

2002年左右，由Google在搜索引擎，提出GoogleSuggest 自动提示，提出AJAX概念(Asynchronous JavaScript And XML)

AJAX=HTML+CSS+JS+DOM+XML+HTTP

作用:无刷新，无提交，无跳转情况下网页内容局部更新.

应用场合:聊天室，股票走势图，搜索建议...

示例一:验证用户名是否存在(注册其中一个功能)

a:创建php existsuname.php (获取参数一个 uname;查询;'该用户名已存在',或者'该用户名可以使用')

b:创建html userreg.html (用户名失去焦点自动触发ajax)

1.4 test

1:编写sql:sohu.sql 库sohu 表user(uid,uname,upwd) 插入三行记录

```

#sohu.sql
#1:创建库
CREATE DATABASE sohu CHARSET=utf8;
USE sohu;
#2:创建表
CREATE TABLE user(
  uid INT PRIMARY KEY AUTO_INCREMENT,
  uname VARCHAR(20),
  upwd VARCHAR(32)
);
#3:添加3条记录
INSERT INTO user VALUES(null,'tom','123');
  
```

```
INSERT INTO user VALUES(null,'jerry','123');
INSERT INTO user VALUES(null,'kk','123');
```

2:编写 login.php 接收客户端提交的uname和upwd，验证用户名密码是否正确，返回"login-succ"或者"login-err"

```
init.php
<?php
//1: 创建连接
$conn = mysqli_connect('127.0.0.1','root','','sohu');
//2: 设置编码
mysqli_query($conn,"SET NAMES UTF8");
?>
```

```
login.php
<?php
//1: 获取参数 uname upwd
@$uname = $_REQUEST["uname"]or die("用户名是必须的");
@$upwd = $_REQUEST["upwd"]or die("密码是必须的");
//2: 创建数据库连接
//3: 设置编码
require("init.php");
//4: 创建sql, 并且发送sql
$sql = "SELECT * FROM user WHERE ";
$sql .= " uname='$uname' AND upwd='$upwd'";
$result = mysqli_query($conn,$sql);
$row = mysqli_fetch_assoc($result);
//5: 判断
if($row===null){
    echo "login-err";
}else{
    echo "login-succ";
}
//6: 返回值 Login-succ Login-err
?>
```

3:编写login.html,输入用户名和密码后，点击"登录"按钮,通过AJAX 发送AJAX请求，若服务器返回"login-succ" 显示"登录成功，欢迎回来"，若服务器返回"login-err" 显示"登录失败"

```
login.html
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
<h1>ajax 用户登录</h1>
<form action="#">
    用户名<input type="text" id="uname" /><br />
    密码<input type="password" id="upwd" /><br />
    <input id="btn1" type="button" value="登录"/><br />
</form>
<script>
    //1: 获取登录按钮
```

```

//2: 绑定点击事件
btn1.onclick = function(){
    //2.1: 获取用户名和密码
    var u = uname.value;
    var p = upwd.value;
    //2.2: 创建对象XHR
    var xhr = new XMLHttpRequest();
    //2.3: 绑定事件
    xhr.onreadystatechange = function(){
        var input = xhr.responseText;
        alert("|"+input+"|");
        if(input==='login-succ'){
            alert('登录成功');
        }else{
            alert("登录失败");
        }
    }
    //2.4: 创建连接
    xhr.open('GET', 'login.php?uname='+u+'&pwd='+p, true);
    //2.5: 发送请求
    xhr.send(null);
}
</script>
</body>
</html>

```

#建议:php程序不要添加多余空格和回车

2 ajax 常用对象属性及处理不同数据格式 (JSON-重点&难点)

2.1 REVIEW

HTTP协议:作用(传送网页中数据 html/css/js/img/mp3..)

请求消息

[请求方式:GET/POST/PUT/HEAD/DELETE/CONNECT/TRACE/OPTIONS]

响应消息

[响应状态码] 100 200 300 400 500

[响应主体类型]text/plain;text/html;

GET/POST区别

AJAX 概述与工作流程

AJAX(Asynchronous JavaScript And XML)异步javascript and xml

核心对象:XHR 作用发送请求并且接收服务器返回数据

AJAX开发流程(背)

(1)创建对象 XHR var xhr = new XMLHttpRequest();

(2)绑定事件

```

xhr.onreadystatechange = function(){
    xhr.responseText;
}

```

```
(3)创建服务器连接 xhr.open('GET','x.php',true);
(4)发送请求      xhr.send(null);
```

2.2 同步异步(工作方式)

同步请求:前面的请求方式均属于同步(5 GET 1 POST)

异步请求:AJAX

同步工作方式 (打电话)

客户端	服务器
填写表单	等待
点击提交发送(等待)	操作
接受数据	等待

异步工作方式(微信)

客户端	服务器
填写表单	操作
点击提交发送(操作)	操作
接收数据	操作

2.3 AJAX常见对象、属性、方法、事件

2.3.1 对象 XMLHttpRequest (作用:发送请求并且接收服务器返回数据)

```
if(window.XMLHttpRequest){
    var xhr = new XMLHttpRequest();
}else{
    var xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
```

2.3.2属性

readyState 表示当前XHR对象状态,不能赋值,可取值 其值会随着请求-响应过程进程变化.

- 0 UNSENT 请求消息尚未发送
- 1 OPENED XHR已经创建与WEB服务器连
- 2 HEADER_RECEIVED XHR已经接收到响应消息头部
- 3 LOADING XHR正在收到响应消息主体
- 4 DONE XHR 接收完成响应消息主体

```
<?php
    echo "hello";
?>
```

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
<h1>ajax 理解XHR状态</h1>
<script>
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function(){
        //alert("状态"+xhr.readyState);
        if(xhr.readyState===4 && xhr.status===200){
            alert(xhr.responseText);
        }
    }
    xhr.open('GET', '01.php', true);
    xhr.send(null);
</script>
</body>
</html>
```

XHR 状态改变图



- status 服务器响应状态码 200
- responseText 保存响应消息主体(文本)
- responseXML 保存响应消息主体(XML)

2.3.3方法

`open(method,url,isAsync);`

- 创建(打开)一个服务器连接
- isAsync:发送请求方是异步(true)同步(false)
- send();发送请求主体
 - 如果请求方式GET 参数 null
 - 如果请求方式POST 参数 自己拼接

2.3.4事件：当XHR状态改变时会触发事件

onreadystatechange()

2.4 AJAX接收服务器返回不同格式的数据

- text/plain 纯文本--了解
- text/html 文本与网页--了解
- application/javascript JS代码--了解
- application/json json格式数据--熟练掌握
- application/xml xml 格式数据--了解

2.5 test

功能:当当网一些功能模块!!!!!!

1. 添加图书
2. 搜索提示

[c]

```
<li>css大全</li>
<li>css指南</li>
```

3. 显示图书列表

2.6 Do it

1. 添加图书

- i. 创建sql文件dangdang.sql
 - a. 创建库 dangdang
 - b. 创建表 dd_book
 - c. bid 编号 bname 名称 pic 图片
 - d. price 价格 pubdate 上架时间
 - e. 添加10条记录(中文)css 2本 js3本 ..

#:创建库 dangdang

#:创建表 dd_book

#bid 编号 bname 名称 pic 图片

#price 价格 pubdate 上架时间

```
CREATE DATABASE dangdang CHARSET=UTF8;
```

```
USE dangdang;
```

```
CREATE TABLE dd_book(
  bid INT PRIMARY KEY AUTO_INCREMENT,
  bname VARCHAR(50),
  pic VARCHAR(100),
  price DECIMAL(10,2),
  pubdate DATETIME
);
```

```
INSERT INTO dd_book VALUES(null, 'css指南', '1.jpg', 100, now());
```



```

INSERT INTO dd_book VALUES(null,'css大全','2.jpg',100,now());
INSERT INTO dd_book VALUES(null,'js大全','3.jpg',90,now());
INSERT INTO dd_book VALUES(null,'js指南','4.jpg',100,now());
INSERT INTO dd_book VALUES(null,'js入门','5.jpg',100,now());
INSERT INTO dd_book VALUES(null,'js入门1','6.jpg',30,now());
INSERT INTO dd_book VALUES(null,'js入门2','7.jpg',100,now());
INSERT INTO dd_book VALUES(null,'js入门3','8.jpg',35,now());
INSERT INTO dd_book VALUES(null,'js入门4','9.jpg',100,now());
INSERT INTO dd_book VALUES(null,'js入门5','10.jpg',100,now());

```

ii. 创建php程序 add_book.php 接收用户提交参数

bname,pic,price,pubdate

并且添加数据库 返回值

图名不在 -2 图片不在 -3 价格不在 -4 上架不在 -5 sql出错 -1

正确将新添加记录 id返回

```
<?php
```

```
//1: 获取参数返回值数值
```

```
//die("-2");
```

```
//第一个功能: 停止当前php执行
```

```
//第二个功能: 输出 -2
```

```
@$bname = $_REQUEST['bname']or die('-2');
```

```
@$pic = $_REQUEST['pic']or die('-3');
```

```
@$price = $_REQUEST['price']or die('-4');
```

```
@$pubdate = $_REQUEST['pubdate']or die('-5');
```

```
//2: 创建一个数据库连接
```

```
//3: 设置编码
```

```
require("init.php");
```

```
//4: 创建sql语句, 并且发送
```

```
$sql = "INSERT INTO dd_book VALUES(null,'$bname','$pic',$price,'$pubdate')";//14:14--14:19
```

```
$result = mysqli_query($conn,$sql);
```

```
//5: 判断 sql 出错 -1
```

```
// 正确 返回当前记录id
```

```
if($result===false){
```

```
    echo '-1';
```

```
}else{
```

```
    $id = mysqli_insert_id($conn);
```

```
    echo $id;
```

```
}
```

```
?>
```

iii. 创建html程序 add_book.html 表单AJAX发送请求

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
</head>
```

```
    <body>
```

```
        <h1>ajax 添加图书信息</h1>
```

```
        <form action="#">
```

```
            新书名称<input id="bname" /><br/>
```

```
            书籍图片<input id="pic" /><br/>
```

```
            书籍单价<input id="price" /><br/>
```

```
            出版日期<input id="pubdate" /><br/>
```

```
            <input id="btn1" type="button" value="添加"/><br/>
```

```
        </form>
```

```

<script>
//14:23--14:30
//1: 获取添加按钮
//2: 为按钮绑定点击事件
    btn1.onclick = function(){
        //2.1: 获取用户输入 4个值
        var b = bname.value;
        var p = pic.value;
        var r = price.value;
        var d = pubdate.value;
        //2.2: 创建XHR对象
        var xhr = new XMLHttpRequest();
        //2.3: 为XHR绑定事件(接收服务器返回数据)
        xhr.onreadystatechange = function(){
            if(xhr.readyState===4&&xhr.status===200){
                var num = parseInt(xhr.responseText);
                if(num>0){
                    alert("新记录添加成功!编号为"+num);
                    bname.value = '';
                    pic.value = '';
                    price.value = '';
                    pubdate.value = '';
                }else{
                    alert("新记录添加失败! 错误代码是"+num);
                }
            }
        }
        /*** 发起POST请求
        //a: open('POST');
        //b: 设置请求消息头
        //c: 发送数据
        //2.4: 打开一个与服务器连接
        xhr.open('POST', 'add_book.php', true);
        //2.5: 设置请求消息头
        xhr.setRequestHeader('Content-Type',
            'application/x-www-form-urlencoded');
        //2.6: 发送请求数据
        xhr.send(`bname=${b}&pic=${p}&price=${r}&pubdate=${d}`);
        }
    }
</script>
</body>
</html>

```

2. 搜索提示

- i. 创建php程序 search.php 接收用户提交参数 kw {关键字} j js c css 查询数据库中dd_book.bname 中是否有包含该字 母记录; #模糊查询 % 通配符代表 0位或多位字符 SELECT * FROM dd_book WHERE bname LIKE '%j%'

注意事项:

#如果搜索kw不存在 ' 关键字不存在 '

#如果数据正确

' js大全 '

js指南 '

...

;

2. 创建html程序 search.html

3. 显示图书列表(text/html,json+xml) book_list.php 查询dd_book表中所有记录, \$rows 将记录转换....输出 book_list.html 创建一个表格,table thead tbody 当网页加载完成自动发送ajax请求book_list.php 获取所有tr保存在tbody中

1. 完成当当图书网列表显示

2. 完成当当图书网删除指定图片功能

i. 创建php

del_book.php获取用户提交的参数bid删除数据库指定记录

ii. book_list.html为每个删除按钮绑定点击事件发送AJAX请求del_book.php删除指定记录后, 将当前行(tr)删除.

3 day03

3.1 REVIEW

AJAX开发流程(面试题)

(1)创建一个XHR `var xhr = new XMLHttpRequest();`

(2)绑定事件 `xhr.onreadystatechange = function(){};`

事件触发时机: 当对象XHR状态改变触发

0-1 1-2 2-3 3-4 4次

2.1:二次判断 `if(xhr.readyState===4&&xhr.status===200)`

2.2: `xhr.responseText`

(3)打开连接 `xhr.open('GET', 'x.php', true);`

`true`:是否是异步请求

(4)发送请求 `xhr.send(null);`

AJAX对象、属性、方法、事件

`XMLHttpRequest`

`readyState`

`status`

`responseText`

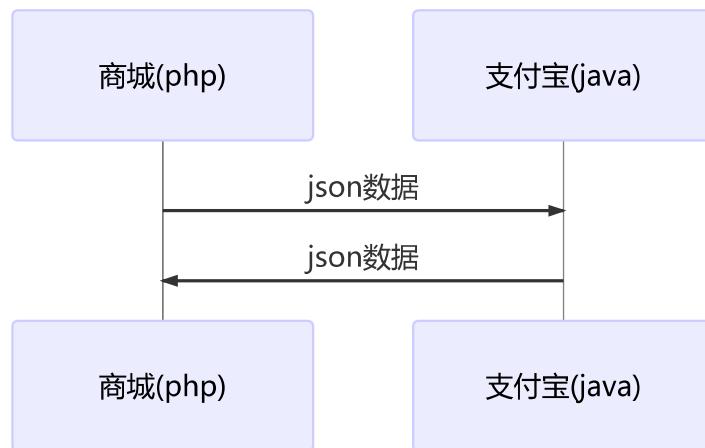
`responseXML`

`open()`

`send();`

`onreadystatechange()`

3.2 原生:AJAX 处理不同数据格式(JSON/JAVASCRIPT/XML)



1. JSON：是字符串数据格式，用于描述数据

2. JSON(JavaScript Object Notation)是一种轻量级的数据交换格式，易于阅读和编写，语法基于JS，但目前已经被各种语言支持.成为一种“异构系统交互数据的标准”

3. JSON: 语法要求

- i. 一个JSON字符串有且只有一个根
- ii. JSON中可以表示数字，BOOL,NULL,字符串（字符串必须是双引号）
- iii. 数组中可以包含多个值，使用逗号分隔
- iv. 对象中可以包含多个键和值，使用逗号分隔，键和值之间用冒号分隔，键必须用双引号.

4. PHP-AJAX-JSON开发流程

- i. 服务器php:将数组转换JSON字符串发送 服务端: array--->json--->输出 \$arr = ...;//手工创建数组，或者从数据中查询数组 \$inputJson = json_encode(\$arr); echo \$inputJson;
- ii. 客户端js:将JSON字符串转JS对象 客户端:json字符串-->JS对象->拼接创建DOM var obj = JSON.parse(xhr.responseText);

5. JQuery中的AJAX

- i. \$('选择器').load(URL,[data],fn);
- ii. \$.get(URL,[data],fn);
- iii. \$.post(URL,[data],fn);
- iv. \$.getJSON(URL,[data],fn);
- v. \$.getScript(URL,[data],fn);
- vi. \$.ajax(URL,[data],fn); 万能AJAX

vii. 标准语法:

```

$.ajax({
    type: 'GET',           // 请求方式 POST/DELETE/PUB/HEAD
    url: 'x.php',          // 请求程序地址(必须)
    data: 'k1=v1&k2=v2'    // 请求提交服务器数据
    beforeSend: fn,        // 请求消息发送之前回调方法

```

```

    success:fn,           // 响应完成并且响应成功回调方法    200
    error:fn,             // 响应完成并且出错回调方法        404 500..
    complete:fn           // 响应完成后回调(无论成功失败)
  });
#注意事项:请求参数

```

viii. 二种写法: 拼接字符串参数:data:'id='+id; //参数数量少可以使用 对象方式参数: data: {cname:"tom",age:10}

3.3 EXERCISE

3.3.1

```

book_list_json.php
//1: 修改响应主体类型application/json
//2: 创建数据库连接
//3: 设置编码
//4: 创建SQL语句, 并且发送sql
//  查询dd_book表中所有内容
//5: 抓取所有记录
//6: 转json字符串  php 函数 json_encode($arr)
//7: 发送json字符串

```

```

book_list_json.html
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
<h1>ajax json 显示图书列表信息</h1>
<table border="1" width="89%">
  <thead>
    <tr>
      <td>图书编号</td>
      <td>图书名称</td>
      <td>图书价格</td>
    </tr>
  </thead>
  <tbody id="tb1">
  </tbody>
</table>
<script>
//1: 页面元素加载成功后发送AJAX请求
//2: 接收服务器返回JSON字符串转JS对象
//a: 创建XHR 对象
//b: 绑定事件(判断2次 4 200)
//b.1 (判断2次 4 200)
//b.2 将json字符串转JS对象
//b.3: 拼接字符串创建DOM元素
//b.4: 创建空字符串
//b.5: 循环内获取
//for 结束 11:27--11:30
//b.6: 保存字符串

```

```

//c:打开连接
//d:发送请求
//3:拼接字符串创建DOM元素
</script>
</body>

```

3.3.2

```

<?php
//功能:AJAX 处理数据格式javascript
//服务器php: 输出 javascript程序
//1:修改响应主体内容 application/javascript
//2:输出程序??
//  获取客户端首选语言
//  如果首选语言前二个字符 zh 输出 你好
//  ..                               en 输出 hello
//  ..                               ja 输出 ..
//客户端js: 接收 eval()

//0:修改响应主体内容 js

//?1:获取所有用户请求头部信息

//?2:获取其中- 首选语言

//?3:截取前二个字母

//?4:判断如果前二个字母 zh 输出 alert('你好');
//  ..                               en 输出 alert('hello');
//

?>

```

3.3.3

功能模块:味多美蛋糕

1:创建sql文件 weiduomei.sql

2:创建库 weiduomei utf-8

3:创建二张表

dm_cake_type 蛋糕类别表

(编号,类别名称,该类别下蛋糕数量)

1 巧克力 3

2 芝士 2

3 水果 2

dm_cake 蛋糕信息表

(编号,名称, 价格,图片,所属类别编号)

1 黑巧克力 3000 1.jpg 1

添加7条记录

4:添加蛋糕

add_cake.php 15:35--15:50

接收客户端提交dname/pic/price/tid

保存入数据库

返回信息 json字符串

`$_REQUEST['pic']` or

```
die('{"code":-2,"msg":"图片是必须的"}');
echo '{"code":1,"msg":"添加成功"}'
add_cake.html
为客户提供四个输入框,一个button按钮,点击后
发送ajax请求
data:{dname:n,pic:p,price:d,tid:t}
```

5:查询某类别下所有蛋糕

select_cake.php 16:50--17:05

- 1.1:接收客户端传送参数 tid 类别编号
- 1.2:查询数据库中指定类别编号下所有蛋糕
- 1.3:返回json

select_cake.html

创建下拉列表

```
<select id="tid">
    <option value="1">巧克力</option>
    <option value="2">芝士</option>
    <option value="3">水果</option>
</select>
<table>...</table>
```

为列表绑定事件 onchange

- 1:获取当前列表value tid
- 2:发送ajax请求 GET select_cake.php
- 3:并且获取返回数据 json 拼接字符串保存
tbody [蛋糕图片,蛋糕名称]