复习：

AJAX

数据库服务器 —— SQL

Web服务器 —— PHP

HTTP协议 —— 请求/响应

原生AJAX

jQuery中的AJAX

HTML5

Bootstrap

今日目标：

(1)复习数据库服务器

(2)复习PHP Web服务器

(3)原生AJAX概述

(4)使用AJAX发起异步请求——重点&难点

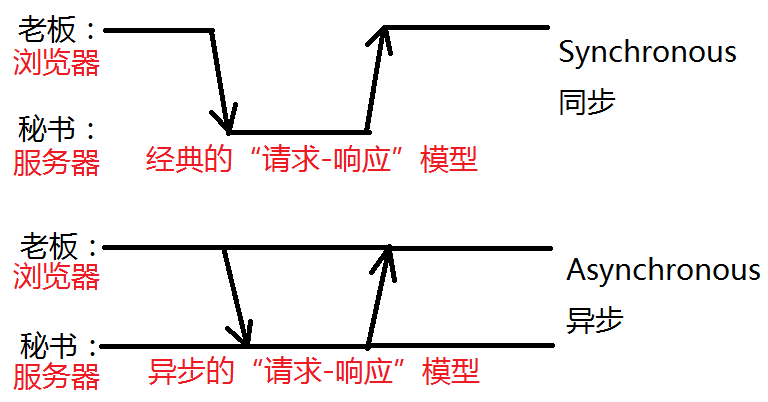
练习1：编写一个.sql脚本中，创建数据库名bbs，创建一个表user保存注册用户的信息(uid、uname、upwd、regTime)，插入三行记录，tom、mary、king，查询出所有的记录行

练习2：编写一个checkuname.php，接收客户端提交的uname数据，执行数据库查询，向客户端输出该用户名是否已经存在，如果已经存在，则向客户端输出cunzai，否则输出bucunzai

1.AJAX概述

Asynchronous Javascript And Xml，异步的JS和XML。

2001年，Google在搜索引擎应用中为了提高客户端的浏览体验，提出了AJAX的概念——Google Suggest。实现的目标：在无刷新无提交的页面内容局部的更新。涉及到的技术：HTML、CSS、JS、DOM、XML、HTTP —— 纯客户端技术的大集合。



浏览器发起的请求消息的类型：

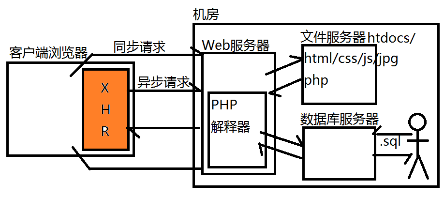
**(1)同步请求：**地址栏输入URL、超链接跳转、表单提交...

**(2)异步请求：**使用AJAX发起

提示：浏览器发起异步请求，依靠的是底层的XMLHttpRequest对象

|  |
| --- |
| 面试题：谈一下对AJAX理解 |

|  |
| --- |
| XHR对象的创建——注意浏览器兼容性  var xhr = null;  if(window.XMLHttpRequest){ //新IE，其它浏览器  xhr = new XMLHttpRequest();  }else { //老IE  xhr = new ActiveXObject('Microsoft.XMLHTTP');  } |

****

2.XMLHttpRequest对象的常用成员：

注意：XHR对象的用途：异步的发起HTTP请求消息，并接收服务器返回的响应消息。



**成员属性：**

readyState：int，初始值为0；表示当前请求-响应进行到哪个步骤了，可取值0 - 1 - 2 - 3 - 4

0： UNSENT 请求消息尚未发送

1： OPENED XHR已经打开了到服务器的连接

2： HEADERS\_RECEIVED XHR已经接收到了响应消息起始行和头部

3： LOADING XHR正在加载响应消息的主体

4： DONE XHR已经接收完成响应消息

responseText：初始值为""，readyState值变为3开始才有响应消息主体文本

responseXML：初始值为null，readyState值变为3开始才有响应消息主体内容——XMLDOM树对象

status：初始值为0，保存着响应状态码，readyState值变为2开始才有值

statusText：初始值为""，保存着响应消息中的原因短句，readyState值变为2开始才有值

**成员事件：**

onreadystatechange：每次readyState值发生改变就触发

**成员方法：**

open(method, url)：打开到服务器的连接

send()：发送请求消息

setRequestHeader(name,value)：设置请求消息头部

getAllResponseHeaders()：获取所有的响应消息头部

getResponseHeader(name)：获取某个响应消息头部

使用XHR对象发起请求的步骤——重点：

(1)创建XHR对象

var xhr = new XMLHttpResponse();

(2)监听XHR的就绪状态改变事件

xhr.onreadystatechange = function(){}

(3)连接到服务器

xhr.open(method, url, isAsync);

(4)发起请求消息

xhr.send(data);

提示：AJAX错误都是“看不见摸不着的”，必须通过Network中查看请求-响应消息的内容来纠错。

3.使用XHR发起异步的HTTP GET请求

应用示例：当输入框失去焦点时，异步验证用户名是否已经存在

实现步骤：

(1)创建必需的数据库结构 bbs.user(uid,uname,upwd)

(2)创建服务器端页面，checkuname.php，接收客户端提交的uname，返回cunzai或bucunzai

(3)编写客户端页面，待输入框失去焦点时，发起异步HTTP GET请求，根据服务器端返回的查询结果，给用户以提示

核心代码：

(1)创建XHR对象

var xhr = new XMLHttpResponse();

(2)监听XHR的就绪状态改变事件

xhr.onreadystatechange = function(){

if(xhr.readyState===4){

if(xhr.status===200){

doResponse(xhr);

}else {

alert('响应完成但有问题');

}

}

}

(3)连接到服务器

xhr.open('GET', 'x.php?k=v&k=v', true);

(4)发起请求消息

xhr.send(null);

4.使用XHR发起异步的HTTP POST请求

应用示例：弹出模态对话框，进行用户登录验证

实现步骤：

(1)创建必需的数据库结构 bbs.user(uid,uname,upwd)

(2)创建服务器端页面，login.php，接收客户端提交的uname和upwd，返回succ或err

(3)编写客户端页面，用户点击“登录”按钮，弹出“模态对话框”，用户输入登录信息，点击“提交”按钮异步GET/POST请求，提交给服务器端PHP，验证是否正确

核心代码：

(1)创建XHR对象

var xhr = new XMLHttpResponse();

(2)监听XHR的就绪状态改变事件

xhr.onreadystatechange = function(){

if(xhr.readyState===4){

if(xhr.status===200){

doResponse(xhr);

}else {

alert('响应完成但有问题');

}

}

}

(3)连接到服务器

xhr.open('POST', 'x.php ', true);

(3.5)修改请求消息头部

xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');

(4)发起请求消息

xhr.send('k=v&k=v');

|  |
| --- |
| ES6新特性： **` `**字符串——（1）其中可以包含 ${ } 变量占位符。（2）其中的文本内容可以换行  var str = `<li>AA</li>  <li>BB</li>`; |

练习内容：

(1)使用XHR对象发起异步的HTTP请求——成绩录入系统

<h1>成绩录入系统</h1>

学生姓名： <input id="stuName">

语文成绩： <input id="chinese">

数学成绩： <input id="math">

<input type="button" value="保存成绩">

要求：点击“保存成绩”按钮，发起异步HTTP请求，把数据提交给服务器端的PHP页面，INSERT到MySQL数据库，返回'succ'或'err'提示文字。客户端弹出一个alert("保存成功/失败")。用户点击确定后，清空输入框内容，继续录入下一个学生成绩。

实现步骤：

(1)创建tarena.sql，创建score表，包含sid、stuName、chinese、math四列，提交给数据库服务器执行

(2)创建score\_add.php，接收客户端提交的stuName / chinese / math数据，执行INSERT，把数据插入到数据库，返回succ或err

(3)创建score\_add.html，显示三个输入框及提交按钮，一点击发起异步POST请求