**十八 -1、javascript-AJAX**

**一、AJAX认识**

**AJAX全称：Asynchronous JavaScript and XML（异步JavaScript和XML) ；**

【注】XML 、JSON都是字符串，作用都是用来传输数据

**1.XML 标签名是自定义、可以存储任意类型的数据**

用户大型的门户网站的数据交互 （腾讯新闻、网易新闻、新浪、凤凰网）

**2.JSON 数据传输的格式 超过90%**

数据处理 通过数据结构处理数据

数据的传输 字符串是最方便

**二、AJAX下载数据**

**1.创建一个ajax对象**

（1）打开浏览器（**兼容问题：**在IE6及IE6以下不支持ajax；

new ActiveXObject("Microsoft.XMLHTTP"））

//var xhr = new XMLHttpRequest();

通过if解决：

var xhr = null;

if(window.XMLHttpRequest){

xhr = new XMLHttpRequest();

}else{

xhr = new ActiveXObject("Microsoft.XMLHTTP");

}

（2）在地址栏输入地址

xhr.open("get", "1.txt", true);

（3）提交

xhr.send();

（4）等待服务器返回内容

xhr.onreadystatechange = function(){

if(xhr.readyState == 4){

alert(xhr.responseText);

}

}



**三、try\_catch**

**1.try\_catch 排除异常，捕捉异常**

【格式】

try{

我们想要执行的代码

}catch(error){

当try里面执行异常的时候，会直接跳转到catch来进行补救

}

【原理】

1、当try中的代码执行正常的时候，catch中的代码不会去运行。

2、当try中的代码执行异常的时候，会去立即执行catch中的代码进行补救。并且catch中的error会包含，try中异常的信息。

try{

alert("代码之前");

alert(a);

alert("代码之后");

}catch(error){

alert(error); //ReferenceError: a is not defined

alert("补救的代码");

}

**2.try\_throw\_catch**

【作用】用的最多的地方就是代码调试。

【格式】

try{

throw new Error("自定义错误信息");

}catch(error){

}

throw 手动抛出异常的。如果遇到了throw，就直接去catch中执行代码。

try{

alert("代码前");

throw new Error("我是错误信息");

alert("代码后");

}catch(error){

alert(error);

alert("我是补救代码");

}

**3.用try\_catch解决ajax兼容问题**

var xhr = null;

try{

xhr = new XMLHttpRequest();

}catch(error){

xhr = new ActiveXObject("Microsoft.XMLHTTP");

}

**四、ajax\_open方法**

·· xhr.open("get", "1.txt", true);

参数：

1、请求方式

get post

2、请求地址 url

3、是否异步

true 异步 false 同步

(1)同步：阻塞操作，后一个程序执行，必须等到前一个程序执行完成以后。

(2)异步：非阻塞操作，后一个程序的执行和前面程序的执行结果没有任何关系。

进程 线程 阮一峰

**五、onreadystatechange事件**

onreadystatechange 事件

当readyState(准备状态)属性发生变化的时候，会触发该事件

0(初始化) 还没有调用open方法之前

1(载入) 调用send方法以后

2(载入完成) 已经收到所有相应的内容

3(解析) 正在解析

4(解析完成) 解析完成

xhr.onreadystatechange = function(){

alert(xhr.readyState);

if(xhr.readyState == 4){

alert(xhr.responseText);

}

}

**六、ajax\_status**

200 OK

404 Not found

400

AJAX状态码说明

1\*\*：请求收到，继续处理

2\*\*：操作成功收到，分析、接受

3\*\*：完成此请求必须进一步处理

4\*\*：请求包含一个错误语法或不能完成

5\*\*：服务器执行一个完全有效请求失败

100——客户必须继续发出请求

101——客户要求服务器根据请求转换HTTP协议版本

**200——交易成功**

201——提示知道新文件的URL

202——接受和处理、但处理未完成

203——返回信息不确定或不完整

204——请求收到，但返回信息为空

205——服务器完成了请求，用户代理必须复位当前已经浏览过的文件

206——服务器已经完成了部分用户的GET请求

300——请求的资源可在多处得到

301——删除请求数据

302——在其他地址发现了请求数据

303——建议客户访问其他URL或访问方式

304——客户端已经执行了GET，但文件未变化

305——请求的资源必须从服务器指定的地址得到

306——前一版本HTTP中使用的代码，现行版本中不再使用

307——申明请求的资源临时性删除

**400——错误请求，如语法错误**

401——请求授权失败

402——保留有效ChargeTo头响应

403——请求不允许

**404——没有发现文件、查询或URl**

405——用户在Request-Line字段定义的方法不允许

406——根据用户发送的Accept拖，请求资源不可访问

407——类似401，用户必须首先在代理服务器上得到授权

408——客户端没有在用户指定的饿时间内完成请求

409——对当前资源状态，请求不能完成

410——服务器上不再有此资源且无进一步的参考地址

411——服务器拒绝用户定义的Content-Length属性请求

412——一个或多个请求头字段在当前请求中错误

413——请求的资源大于服务器允许的大小

414——请求的资源URL长于服务器允许的长度

415——请求资源不支持请求项目格式

416——请求中包含Range请求头字段，在当前请求资源范围内没有range指示值，请求也不包含If-Range请求头字段

417——服务器不满足请求Expect头字段指定的期望值，如果是代理服务器，可能是下一级服务器不能满足请求

500——服务器产生内部错误

501——服务器不支持请求的函数

502——服务器暂时不可用，有时是为了防止发生系统过载

503——服务器过载或暂停维修

504——关口过载，服务器使用另一个关口或服务来响应用户，等待时间设定值较长

505——服务器不支持或拒绝支请求头中指定的HTTP版本

**七、表单\_GET/POST**

form 表单里有属性

1、action 填写的是url 如果点击表单中的submit按钮，就会直接跳转到这个url

2、method 请求的方法 默认是GET请求

**1.GET**

【数据传输】是将所有的数据，通过键值对的方式拼接在url的后面，然后传输给服务器。

<http://localhost/code11/AJAX/code/1.get.php?username=xxxxxxxx&age=18&password=123456>

GET请求

数据直接拼接在url后面

第二个参数 url? + 数据

【补充1】：中文 编码函数：encodeURI(中文)

【补充2】：缓存 为了避免造成缓存 可以在url后面拼接时间戳（当前时刻的毫秒数）

xhr.open("get", "code/1.get.php?username=" + encodeURI("李四") + "&age=18&password=123456&" + new Date().getTime(), true);

【优点】方便

【缺点】1、不安全

2、不能进行大文件的传输，不能上传

3、不能超过2kb

**2.POST**

【注】设置提交格式 enctype="application/x-www-form-urlencoded"

【数据传输】通过浏览器内部完成

【注】数据是通过send去发送 数据要当做参数写在send方法里。

【注】要去设置提交格式，写在send方法之前。

xhr.open("post", "code/1.post.php", true);

xhr.setRequestHeader("content-type", "application/x-www-form-urlencoded"); //声明发送的数据类型

//<3>提交

xhr.send("username=李四&age=18&password=123456");

【优点】1、安全

2、理论上来讲，可以传输任意大的数据，可以进行上传操作

【缺点】稍微不方便

**八、封装ajax**

//下载数据

//将处理数据的函数，写成说明书，交给这个下载完以后。

function ajax(method, url, data, success){

//1、创建ajax对象

var xhr = null;

try{

xhr = new XMLHttpRequest();

}catch(error){

xhr = new ActiveXObject("Microsoft.XMLHTTP");

}

if(method == "get" && data){

url += "?" + data;

}

xhr.open(method, url, true);

if(method == "get"){

xhr.send();

}else{

xhr.setRequestHeader("content-type", "application/x-www-form-urlencoded"); //声明发送的数据类型

xhr.send(data);

}

//<4>等待服务器返回内容

xhr.onreadystatechange = function(){

if(xhr.readyState == 4){

if(xhr.status == 200){

//alert(xhr.responseText);

//拿到数据以后

if(success){

//【3】

success(xhr.responseText); //回调函数。

}

}

}

}

}

**//调用**

window.onload = function(){

var aInputs = document.getElementsByTagName("input");

aInputs[0].onclick = function(){

//【1】

ajax("get", "code/1.get.php", "username=xxxx&age=18&password=123456", function(data){

//【4】

//data是下载到的数据

alert("这是GET下载到的数据" + data);

});

}

aInputs[1].onclick = function(){

ajax("post", "code/1.post.php", "username=xxxx&age=18&password=123456", function(data){

alert("这是POST下载到的数据" + data);

});

}

}

<body>

<input type = "button" value = "GET请求" />

<input type = "button" value = "POST请求" />

</body>



**九、JSON对象**

**alert(JSON); //object JSON**

**JSON的两个方法：**

（1）JSON.stringify() 将数据结构转成字符串

var arr = ["One", "Two", "Three"];

var str = JSON.stringify(arr);

// alert(arr);

// alert(str); //["One","Two","Three"]

====================================

var obj = {

name: "xxx",

age: 18,

sex: "男"

}

var str = JSON.stringify(obj);

// alert(str); //{"name":"xxx","age":18,"sex":"男"}

（2）JSON.parse() 将字符串转成数据结构

var str = '["One","Two","Three"]';

var arr = JSON.parse(str);

// alert(arr);//One,Two,Three

======================================

var str = '{"name":"xxx","age":18,"sex":"男"}';

var obj = JSON.parse(str);

alert(obj.name);//xxx

下载数据(字符串) -> 将字符串转成对应的数据结构

传输数据(数据结构) -> 将数据结构转成对应结构的字符串

**十、getList**

window.onload = function(){

var oBtn = document.getElementById("btn");

oBtn.onclick = function(){

ajax("get", "code/getList.php", "", function(data){

alert(data);

//通过对下载到的数据的格式进行分析，进行对应的转换

/\*var arr = JSON.parse(data);

alert(arr);\*/

var obj = JSON.parse(data);

alert(obj.username);

})

}

}