day06

三.ajax

4.代参数的get方法

①服务器

②ajax代码

|  |
| --- |
| xhr.open("get",url,true)  url="/demo/get\_login?uname="+$uname+"&upwd="+$upwd |
| function login(){  //获取页面上用户的用户名和密码  var $uname=uname.value;  var $upwd=upwd.value;  //使用ajax访问服务器，并接受请求  //1.创建xhr异步对象  var xhr=new XMLHttpRequest();  //4.绑定监听，接受响应  xhr.onreadystatechange=function(){  if(xhr.readyState==4&&xhr.status==200){  var result=xhr.responseText;  alert(result);  }  }  //2.创建请求，打开连接  xhr.open("get","/demo/get\_login?uname="+$uname+"&upwd="+$upwd,true);  //3.发送请求  xhr.send();  } |

5.restful规则的接口

|  |
| --- |
| restful接口定义规则  /接口名称/:参数值&:参数值  注意，使用restful不能把非空验证放到后台。  而要把非空验证在前端完成 |

6.post传参

|  |
| --- |
| 注意  1.xhr.send(formdata)必须带着请求主体发送  请求主体就是一个字符串  var formdata="uname="+$uname+"&upwd="+$upwd;  2.在xhr.open和xhr.send之间，设置请求头信息  把content-type设置为可以发送特殊字符  xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded"); |

7.json解析

|  |
| --- |
| json是以js对象，或者js对象数组为表现形式字符串  json串  JavaScript object notation  js 对象 表象方式  json的来源  1.可以手写json串  var str="{'uid':123}";  2.当服务器操作数据库之后，得到的result，在传递给前端之后自动就变成一个json |

①json解析

|  |
| --- |
| 把json字符串转换成js对象/js对象数组，这个行为就叫json解析  做json解析的目的，因为要对数据，使用js的api进行处理 |

②json字符串的格式

|  |
| --- |
| 1.json中用一对{}来表示一个对象  2.json中所有的属性名称，必须使用双引号括起来  使用单引号，不会发生错误，但是建议使用双引号  3.json中的属性值，如果是字符串，也要带双引号  4.整个json是一个字符串，所以最外层要是用单引号包裹 |

③解析语法

|  |
| --- |
| var arr=JSON.parse(result);  arr就是一个js的对象数组，可以使用js的数组操作模式了 |

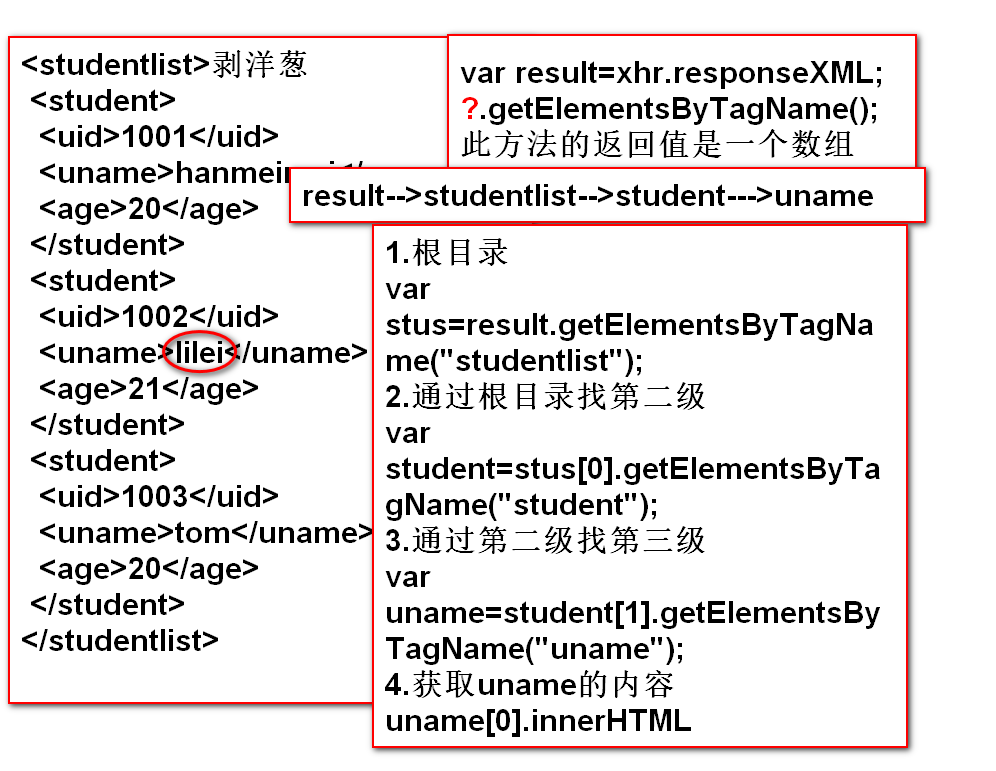
8.xml解析

|  |
| --- |
| eXtensible markup language  可拓展的 标记 语言  所有的标签，属性都是自己定义  xml就是做数据传递的，不用于数据展示 |

①语法

|  |
| --- |
| 1.首行做版本声明  <?xml version="1.0" encoding="utf-8"?>  2.xml标记，都是自己命名，只有双标记，没有单标记  3.xml标记，严格区分大小写，开始标记和结束标记必须一致  4.每个xml文档，有且只有一对根标记 |

②xml解析



四.ajax项目(使用restful风格)

1.login模块(后台接口，前台页面)

接口名称 /v1/login/:uname&:upwd

req.params.uname

错误总结

1.数据库没有开启

|  |
| --- |
|  |

2.缺少括号

|  |
| --- |
| 出乎意料的结尾，login这个方法未定义  说明login方法中缺少} |

3.符号错误

|  |
| --- |
|  |

作业：

1.userlist接口，完成用户在表格加载(json解析)

2.再完成一遍xml解析

3.再完成一遍restful，get方法登陆