```
全称: "Asynchronous Javascript And XML" (异步javaScript和XML)
                                      AJAX是一种用于创建快速动态网页的技术。通过在后台与服务器进行少
                    简单介绍:
                                   量数据交换,AJAX可以使网页实现异步更新
                                   用途: 可以在不重新加载整个网页的情况下, 对网页的某部分进行更新
                                                                                              var xhr = new XMLHttpRequest();
                                                      高级浏览器,包括火狐、chrome、opera,ie7以上:
                                                                            var xhr = new ActiveXObject("Microsoft.XMLHTTP");
                                                                            var xhr = ActiveXObject("Microsoft.XMLHTTP");
                                     1、创建ajax对象
                                                                            var xhr = new ActiveXObject("Msxml2.XMLHTTP.3.0");
                                                                            var xhr = new ActiveXObject("Msxml2.XMLHTTP.5.0");
                                                                            var xhr = new ActiveXObject("Msxml2.XMLHTTP.6.0");
                                                                            //IE维护的最高版本
                                                      ie7以下的垃圾浏览器:
                                                        使用XMLHttpRequest对象的open创建请求
                                    2、创建HTTP请求
                                                       如果是get请求,参数需要附加到url里
                                                       参数详情见: ajax.md
                                                   - 如果是get请求,send参数可以为空;
- 如果为post请求,参数必须在send参数中传递,参数格式和url中设置的一样
- 如果为post方式需要在send前,设置请求共信息;
                                                            xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded")
                                                    //get方式
                                                    xhr.send(null);
                                    3、发送请求
                                                    //post方式
                                                    xhr.setRequestHeader('content-type', 'application/x-www-form-urlencoded');
                                                    xhr.send('username=tom&pwd=123');
                                                          说明: 向服务器发送请求的目的是为了获得服务器端的数据,要获得服务器端返回的数据,需要监
                                                         听XMLHttpRequest的状态,每当XMLHttpRequest状态发生改变时,会触发事件onreadystatechange,我们可以编写onreadystatechange事件处理函数,获取服务器返回的数据
                                                                                                                        - 0: 请求未初始化
                                                                                                                       - 1:服务器连接已建立
                                                                                                                       - 2: 请求已 接收
                                                                                                                       - 3: 请求处理中
                                                                                                                       - 4: 请求已完成, 且响应已就绪
                                                         1、XMLHttpRequest对象的状态,可以通过属性readyState获取,公有以下5种状态:
                    Aiax的使用:
Ajax简介
                                                                                                                                         XMLHttpRequest的status属性用于判断http的状态, 200表示访问成功
                                                         2、当readyState值为4时,就可以获取返回数据了,但返回的数据是成功还是失败,需要根据http的状态码确定
                                     4、监听Ajax状态变化
                                                         xhr.onreadystatechange = function () {
                                                              if (xhr.readyState == 4) {
                                                            if (xhr.status == 200) {
                                                             alert(xhr.responseText);
                                                       可以通过XMLHttpRequest对象的responseText获取数据,现在已经不再使用responseXML获取数据了
                                                                               xhr.onreadystatechange = function () {
                                                                               if (4 == this.readyState && 200 == this.status) {
                                                                                //1. 通过eval将json字符串转换为对象
                                                                                 // var obj = eval('('+xhr.responseText+')');
                                                                                  // console.log(obj);
                                                                                  //2 使用json.parse将json字符串转换为对象
                                                                                 obj = JSON.parse(this.responseText);
                                                                                  console.log(obj);
                                                                               };
                                     5、获取响应数据
                                                       客户端获取json数据示例:
                                                                                   //json字符串 ==> json对象
                                                                                   var obj = JSON.parse(json字符串);
                                                                                   var obj = eval('(' + json字符串 + ')');
                                                                                   //json对象转json串
                                                                                   var str=obj.toJSONString();
                                                       JSON字符串和JSON对象的转换:
                                                                                   var str=JSON.stringify(obj);
                                              ajax请求数据时,客户端js代码必须等服务器返回请求的结果后再继续执行
                                   同步:
                                              注意: 同步请求是onreadystatechange事件处理绑定必须在open前, 否则错误
                    同步与异步
                                              ajax请求数据时,客户端js代码不必等领ajax请求的结果,可以继续执行,啥时候ajax请求返回数据在进行处理。
```