day05

一.DOM操作

1.获取元素的对象

|  |
| --- |
| 直接使用这个元素的id值 |

2.获取/修改元素的内容(只有双标签叫内容)

|  |
| --- |
| d1.innerHTML  可以获取/修改内容文本  还可以动态的添加元素 |

3.获取/设置表单控件的数据

|  |
| --- |
| 1.使用input控件的id代表控件对象  2.通过对象的value属性，获取/设置控件的值  <input id="uname" type="text">  uname.value="1000"; |

4.HTML元素的事件

|  |
| --- |
| 允许通过用户的行为来激发的操作就是事件  常用的事件 onclick 用户点击元素，激发某项操作 |
| 文本框/密码框，失去焦点事件 onblur  文本框/密码框,获得焦点事件 onfocus |
| body的事件 onload  当页面加载完成，马上要激发的操作 |

二.AJAX

|  |
| --- |
| 同步 Synchronous  在一个任务进行过程中，不能开启其它的任务。  同步访问：浏览器在向服务器发送请求时，浏览器只能等待服务器的响应，不能做其它事情。  出现场合：  1.地址栏输入网址并跳转  2.a标记跳转  3.submit表单的提交 |
| 异步Asynchronous  在一个任务开启的同时，可以开启其它任务。  异步的访问：浏览器在向服务器发送请求时，不耽误用户在网页上的其它操作  出现的场合  1.用户名的重复验证  2.聊天室  3.搜索建议(百度，京东，淘宝)  4.股票走势图 |

1.什么是AJAX

|  |
| --- |
| Asynchronous JavaScript And Xml  异步的 js 和 xml  本质：使用js提供的异步对象(XMLHttpRequest---xhr)  异步的向服务器发送请求，  并接收响应回来的数据  ajax中，可以无刷新效果，去更改页面的布局结构和内容 |

2.使用Ajax步骤

3.创建异步对象

|  |
| --- |
| 标准创建 var xhr=new XMLHttpRequest()  IE8以下的创建  var xhr=new ActiveXObject("Microsoft.XMLHttp"); |
| 如何判断用户的浏览器版本？  使用window.XMLHttpRequest属性  如果是高版本，这个属性有值  如果浏览器不支持标准创建，这个属性为null  if(window.XMLHttpRequest){  var xhr=new XMLHttpRequest();  }else{  var xhr=new ActiveXObject("Microsoft.XMLHttp");  } |

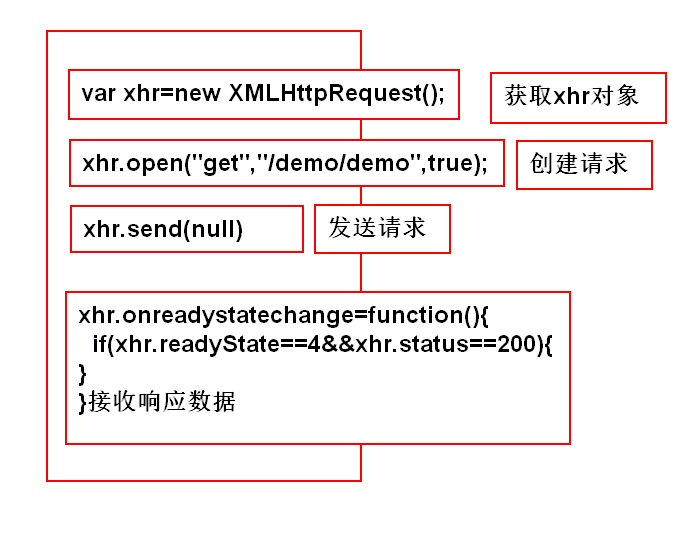
4.异步对象的常用属性和方法

|  |
| --- |
| readystate 表示xhr对象的请求状态  取值 0-4 表示着5个状态  0：请求尚未初始化  1.打开服务器连接，正在发送请求  2.接收响应头  3.接收响应主体  4.接收响应数据成功  当readystate的值为4的时候，才表示所有的数据接收完毕 |
| onreadystatechange 当xhr的readystate属性发生改变时触发 |
| status 表示服务器的响应状态码  只有在status=200，表示服务器已经正确处理了请求 |

使用ajax的步骤

|  |
| --- |
| 1.创建xhr对象  var xhr=new XMLHttpRequest(); |
| 2.绑定监听  xhr.onreadystatechange=function(){  if(xhr.readyState==4 && xhr.status==200){  var result=xhr.responseText;  }  } |
| 3.打开链接，创建请求  xhr.open(method,url,true);  method必须是字符串 "get" "post"  url="127.0.0.1:3000/list?uname="+name+"&upwd="+pwd; |
| 4.xhr.send(); |

01\_ajax.html练习、



|  |
| --- |
| 服务器部分步骤  1.在xz文件夹中，把routes文件夹中的user.js复制。改名为demo.js.  2.创建接口ajaxDemo  //1.ajaxDemo  router.get("/ajaxDemo",(req,res)=>{  res.send("这是我的第一个ajax");  });  3.在app.js中把demo.js导入  const demo=require("./routes/demo.js");  server.use("/demo",demo);  4.启动服务器，在地址栏输入网址，验证  http://127.0.0.1:3000/demo/ajaxdemo  5.在xz中创建文件夹myex.并在app.js中托管静态资源  server.use(express.static('myex'));  6.在myex中创建文件01\_ajax.html中编写ajax代码。 |
| 前端代码  <input type="button" value="ajax" onclick="getMsg()">  <script>  function getMsg(){//尝试传递参数  //1.创建xhr对象  var xhr=new XMLHttpRequest();  //2.绑定监听器  xhr.onreadystatechange=function(){  if(xhr.readyState==4&&xhr.status==200){  var result=xhr.responseText;  console.log(result);  }  }  //3.打开链接，创建请求  xhr.open("get","/demo/ajaxdemo",true);  //4.发送请求  xhr.send();  }  </script> |

5.使用get发送带参数的请求

|  |
| --- |
| url="demo/login?uname=value1&upwd=value2" |

练习：

|  |
| --- |
| 把01\_ajax.html以及对应接口/ajaxDemo，删除，整体重新敲一遍  把02\_login.html以及对应接口/login，删除，整体重新敲一遍 |

作业

|  |
| --- |
| 1.在demo.js中，创建一个接口userlist  在接口中查询出用户表所有的数据，并响应给前端页面  select \* from xz\_user  创建一个网页03\_getList.html.  网页中有一个按钮，点击后，获取userlist接口的数据 |
| 2.在demo.js中，创建接口checkUname.模拟学子商城的注册需求。  文本框失去焦点时，使用ajax连接checkUname.查询数据库是否存在该用户名  如果存在，响应给页面"用户名被占用"  如果不存在，响应给页面"用户名可用"  创建网页04\_register.html。一个文本框，一个密码框，文本框失去焦点的时候调用连接服务器的ajax方法 |

ajax错误总结

|  |
| --- |
| 404：检查url是否正确，要访问的资源是不是在url指向的位置(html位置，和路由)  5XX: 检查接口中的代码是否正确 |
| show is not defined show()方法未定义 |
| Uncaught SyntaxError: Unexpected token { 标点符号，或者括号错误 |
| Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, https.跨域错误，说明你的html是本地打开 |