

## Ajax 项目整理

服务器(app.js)、路由(mypro.js)、连接池(pool.js)、静态资源(.html)

### 一、登录

接收前端传递过来的用户名和密码  
将用户名和密码取数据库查询  
并给出提示 登录成功/用户名密码错误  
1.myPro.js 中写 login 接口  
2.myPro 文件夹中创建文件 01\_login.html  
在网页中添加用户名和密码输入框以及按钮。  
点击按钮是异步的向后端发送请求，完成登录(post 方法)

#### 大概流程:

在静态页面中写一个用户名框/密码框/登录按钮;

当用户名框失去焦点时，判断用户名框的 value 是否为空，返回一句话;

当密码为空，失去焦点，执行函数;

点击登录按钮，触发函数，在请求主体中写上要提交到服务器上的数据;

路由中，接收上一步传过来的数据并连接数据库查询是否存在;

将查询结果返回给 ajax 中的 result;

做一些处理将结果返回给静态页面;

### 二、获取用户列表

要求:

在 myPro.js 定义一个接口 userlist

查询所有的用户数据，以表格方式响应给前端

在 myPro 文件夹中创建网页 02\_list.html

在 body 的 onload 事件中绑定方法 get\_list()

```
function get_list(){
  var xhr=new XMLHttpRequest();
  xhr.onreadystatechange=function (){
    if(xhr.readyState==4&&xhr.status==200){
      var result=xhr.responseText;
      var arr=JSON.parse(result);
      console.log(arr);
    }
  }
  xhr.open("get","/mypro/list",true);
  xhr.send(null);
}
```

上面 console.log(arr)出来以下的数组;

▼ (4) [{}], [{}], [{}], [{}]

```
▶ 0: {uid: 9, uname: "张爱玲", upwd: "666666", email: "ailin@qq.com", phone: "18123564202", ...}
▶ 1: {uid: 20, uname: "lili", upwd: "123456", email: "56843@qq.com", phone: "13026547989", ...}
▶ 2: {uid: 21, uname: "leifld", upwd: "123456", email: "1236598@qq.com", phone: "134682486533", ...}
▶ 3: {uid: 24, uname: "rose", upwd: "123456", email: "6666@qq.com", phone: "17032658456", ...}
length: 4
__proto__: Array(0)
```

#### 大概流程:

①页面加载完成后自动调用函数，所以 get\_list()放在 body 里面的 onload

②创建小黄人对象，发送请求

③在路由中，没有获取数据，直接查询数据库

④将查询的结果返回给 ajax

⑤结果是一个 json 串，转换为数组。数组里面的元素是对象，通过遍历数组元素获取每一个值，拼接到一起

⑥插入到表格 tbody 中

```
function fn(){
  var xhr=new XMLHttpRequest();
  xhr.onreadystatechange=function(){
    if(xhr.readyState==4&&xhr.status==200){
      var result=xhr.responseText;
      var arr=JSON.parse(result);
      var html="";
      for(var i=0;i<arr.length;i++){
        var gender="";
        if(arr[i].gender==0){
          gender="女";
        }else if(arr[i].gender==1){
          gender="男";
        }else {
          gender="不详"
        }
        html+="|<td>"+arr[i].uname+"</td><td>"+arr[i].email+"</td><td>"+arr[i].phone+"</td><td>"+arr[i].user_name+"</td><td>"+gender+"</td><td><a href='javascript:deleteUser(\""+arr[i].uid+"')'>删除</a><a href='#'>修改</a></td></tr>";
      }
      d1.innerHTML=html;
    }
  }
  xhr.open("get","/mypro/list",true)
  xhr.send(null);
}
|  |

```

#### 注意事项:

①获取数据是 json 串，需要使用 JSON.parse()转义为数组;

②循环遍历拼接内容到表格中，需要先初始化一个空字符串来接收这些内容，点击链接，执行 js 代码

③循环结束后，把拼接完成的表格通过 dom 操作添加到 tbody 中

#### 错误:

①从数据库中获得数据的格式是 json 串，要进行数据格式的转换

②拼接字符串时，双引号里面在需要引号，要用单引号

```
✖ Uncaught SyntaxError: userlist.html:3
  Unexpected string

✖ ▶Uncaught ReferenceError: fn is not defined
  at onload (userlist.html:7)
```

### 三、删除

当点击删除链接时，触发一个函数；

给这个函数传一个 uid 参数，点击删除时，传一个实参(arr[i].uid)；

把这个实参传递给"/delete"路由中，接收这个数据，并且根据 uid 的值链接数据库进行删除

把删除的结果返回给 ajax 中的.responseText；将 result 返回给页面中；

路由中注意事项：

```
router.get("/delete", (req, res) => {
  var obj = req.query;
  console.log(obj);
  var $uid = obj.uid;
  console.log($uid);
  pool.query("delete from xz_user where uid=?", [$uid], (err, result) => {
    if (err) throw err;
    if (result.affectedRows > 0) {
      res.send("1");
    } else {
      res.send("0");
    }
  });
});
```

① 第一个 console 是从 url 中传递过来的数据，是一个对象格式

② 第二个 console 是根据这个对象的 uid 属性得到确切的 uid 值，根据这个 uid 值删除数据；

Html 中需要注意：

```
function deleteUser(uid) {
  console.log(uid);
  var xhr = new XMLHttpRequest();
  xhr.onreadystatechange = function () {
    if (xhr.readyState == 4 && xhr.status == 200) {
      var result = xhr.responseText;
      if (result == 1) {
        alert("删除成功");
        fn();
      } else {
        alert("删除失败");
      }
    }
  };
  xhr.open("get", "/mypro/delete?uid=" + uid, true);
  xhr.send(null);
}
```

① 注意参数传递 uid，两个位置

② 当删除成功后，直接调用函数 fn(); 重新获取一次用户列表；

### 四、更新

当点击修改链接后，跳转到页面 update.html(将文件的路径放在 href 里面并且附带参数 uid)；

页面加载完成就出现 uid 下的所有内容，所以这个函数需要放在 body 里面；

此函数的作用是根据 uid 去查询数据库，获取信息，并将所有的信息都填充到 input 里；(如何获取 url 里面的参数？请看后面)

点击修改按钮：调用函数，将修改后的所有信息更新到数据库中，返回 result；

修改成功后：使用这个代码跳转 location.href='http://127.0.0.1:3000/userlist.html';

如何获取 url 里面的参数？

```
var urlParams = new URLSearchParams(
  location.search);
```

```
var uid = urlParams.get("uid");
```

(js 高级里面会讲，先记住)

### 五、注册

点击注册按钮，将所有的内容插入到数据表中

进行用户名验证，密码验证，注册验证。

3 个函数 2 个路由。

## AJAX 面试题

### 1、什么是 AJAX，为什么要使用 Ajax（请谈一下你对 Ajax 的认识）

#### ①什么是 ajax:

AJAX 是“**Asynchronous JavaScript and XML**”的缩写。他是指一种创建交互式网页应用的网页开发技术。

#### ②Ajax 包含下列技术:

a 基于 web 标准 (standards-based presentation) XHTML+CSS 的表示;

b 使用 DOM (Document Object Model) 进行动态显示及交互; quansan 使用 XML 和 XSLT 进行数据交换及相关操作;

c.使用 XMLHttpRequest 进行异步数据查询、检索;

d.使用 JavaScript 将所有的东西绑定在一起。

#### ④为什么要用 ajax:

Ajax 应用程序的优势在于:

1. 通过异步模式, 提升了用户体验
2. 优化了浏览器和服务器之间的传输, 减少不必要的数据往返, 减少了带宽占用
3. Ajax 引擎在客户端运行, 承担了一部分本来由服务器承担的工作, 从而减少了大用户量下的服务器负载。

### 2、AJAX 最大的特点是什么

Ajax 可以实现动态不刷新 (局部刷新) 就是能在不更新整个页面的前提下维护数据。这使得 Web 应用程序更为迅捷地回应用户动作, 并避免了在网络上发送那些没有改变过的信息。

### 3、请介绍一下 XMLHttpRequest 对象。

Ajax 的核心是 JavaScript 对象 ----- XMLHttpRequest, 该对象在 Internet Explorer 5 中首次引入, 它是一种支持异步请求的技术。简而言之, XMLHttpRequest 使您可以使用 JavaScript 向服务器提出请求并处理响应, 而不阻塞用户。通过 XMLHttpRequest 对象, Web 开发人员可以在页面加载以后进行页面的局部更新。

### 4、AJAX 技术体系的组成部分有哪些

HTML, css, dom, xml, XMLHttpRequest, javascript

### 5、AJAX 应用和传统 Web 应用有什么不同

在传统的 Javascript 编程中, 如果想得到服务器端数据库或文件上的信息, 或者发送客户端信息到服务器, 需要建立一个 HTML form 然后 GET 或者 POST 数据到服务器端。用户需要点击 "Submit" 按钮来发送或者接受数据信息, 然后等待服务器响应请求, 页面重新加载。

因为服务器每次都会返回一个新的页面, 所以传统的 web 应用有可能很慢而且用户交互不友好。

使用 AJAX 技术, 就可以使 Javascript 通过 XMLHttpRequest 对象直接与服务器进行交互。

通过 HTTP Request, 一个 web 页面可以发送一个请求到 web 服务器并且接受 web 服务器返回的信息 (不用重新加载页面), 展示给用户的还是通一个页面, 用户感觉页面刷新, 也看不到到 Javascript 后台进行的发送请求和接受响应。

### 6、AJAX 请求总共有多少种 CALLBACK

Ajax 请求总共有八种 Callback

onSuccess  
onFailure  
onUninitialized  
onLoading  
onLoaded  
onInteractive  
onComplete  
onException

### 7.Ajax 和 javascript 的区别

javascript 是一种在浏览器端执行的脚本语言, Ajax 是一种创建交互式网页应用的开发技术, 它是利用了一系列相关的技术其中就包括 javascript。

Javascript 是由网景公司开发的一种脚本语言, 它和 sun 公司的 java 语言是没有任何关系的, 它们相似的名称只是一种行销策略。在一般的 web 开发中, javascript 是在浏览器端执行的, 我们可以用 javascript 控制浏览器的行为和内容。

在 Ajax 应用中信息是如何在浏览器和服务端之间传递的

通过 XML 数据或者字符串

## 8、在浏览器端如何得到服务器端响应的 XML 数据

XMLHttpRequest 对象的 responseXML 属性

## 9、XMLHttpRequest 对象在 IE 和 Firefox 中创建方式有没有不同

有，IE 中通过 new ActiveXObject() 得到，Firefox 中通过 new XMLHttpRequest() 得到

## 10、介绍一下 XMLHttpRequest 对象的常用方法和属性

open(“method”, “URL”) 建立对服务器的调用，第一个参数是 HTTP 请求 方式可以为 GET, POST 或任何服务器所支持的您想调用的方式。

第二个参数是请求页面的 URL。

send() 方法，发送具体请求

abort() 方法，停止当前请求

readyState 属性 请求的状态 有 5 个可取值 0=未初始化，1=正在加载

2=以加载，3=交互中，4=完成

responseText 属性 服务器的响应，表示为一个串

responseXML 属性 服务器的响应，表示为 XML

status 服务器的 HTTP 状态码，200 对应 ok 400 对应 not found

## 11、什么是 XML

XML 是扩展标记语言，能够用一系列简单的标记描述数据

## 12、XML 的解析方式

常用的用 dom 解析和 sax 解析。dom 解析是一次性读取 xml 文件并将其构造为 DOM 对象供程序使用，优点是操作方便，但是比较耗内存。Sax 是按事件驱动的方式解析的，占用内存少，但是编程复杂

## 13、你采用的是什么框架（架包）

这题是必问的，一般也是最开始就会问到。

在 java 中比较流行的有 dojo, Prototype, JQuery, Dwr, extjs 等等

## 14、如果熟悉某种 ajax 框架，他可能会问到怎样在程序中使用这种框架

DWR 框架介绍

DWR(DirectWeb Remoting)是一个 WEB 远程调用框架。利用这个框架可以让 AJAX 开发变得很简单。利用 DWR 可以在客户端利用

JavaScript 直接调用服务端的 Java 方法并返回值给 JavaScript 就好像直接本地客户端调用一样 (DWR 根据 Java 类来动态生成 JavaScript 代码)。

DWR 的实现原理是通过反射，将 java 翻译成 javascript，然后利用回调机制，从而实现了 javascript 调用 Java 代码

## 15、介绍一下 Prototype 的 \$() 函数，\$F() 函数，\$A() 函数都是什么作用

\$() 方法是在 DOM 中使用过于频繁的 document.getElementById() 方法的一个便利的简写，就像这个 DOM 方法一样，这个方法返回参数传入的 id 的那个元素。

\$F() 函数是另一个大受欢迎的“快捷键”，它能用于返回任何表单输入控件的值，比如 textbox, drop-down list。这个方法也能用元素 id 或元素本身做为参数。

\$A() 函数能把它接收到的单个的参数转换成一个 Array 对象。

## 16、介绍一下 XMLHttpRequest 对象

通过 XMLHttpRequest 对象，Web 开发人员可以在页面加载以后进行页面的局部更新。

AJAX 开始流行始于 Google 在 2005 年使用的 “Google Suggest”。

“Google Suggest” 就是使用 XMLHttpRequest 对象来创建动态 Web 接口：当用户开始输入 google 的搜索框，Javascript 发送用户输入的字符到服务器，然后服务器返回一个建议列表。

XMLHttpRequest 对象在 IE5.0+, Safari 1.2, Mozilla 1.0/Firefox, Opera 8+ 和 NetScape7 开始被支持。

## 17、AJAX 的全称是什么？介绍一下 AJAX？

AJAX 的全称是 Asynchronous JavaScript And XML。

AJAX 是 2005 年由 Google 发起并流行起来的编程方法，AJAX 不是一个新的编程语言，但是它是一个使用已有标准的新的编程技术。

使用 AJAX 可以创建更好，更快，更用户界面友好的 Web 应用。

AJAX 技术基于 Javascript 和 HTTP Request。

## 18、Ajax 主要包含了哪些技术？

Ajax (Asynchronous JavaScript + XML) 定义基于 web 标准(standards-based presentation) XHTML+CSS 的表示；

使用 DOM (Document Object Model) 进行动态显示及交互；

使用 XML 和 XSLT 进行数据交换及相关操作；

使用 XMLHttpRequest 进行异步数据查询、检索；

使用 JavaScript 将所有的东西绑定在一起。英文参见 Ajax 的提出者 Jesse James Garrett 的原文, 原文题目 (Ajax: A New Approach to Web Applications)。

类似于 DHTML 或 LAMP, AJAX 不是指一种单一的技术, 而是有机地利用了一系列相关的技术。事实上, 一些基于 AJAX 的“派生/合成”式 (derivative/composite) 的技术正在出现, 如 “AFLAX”。

AJAX 的应用使用支持以上技术的 web 浏览器作为运行平台。这些浏览器目前包括: Mozilla、Firefox、Internet Explorer、Opera、Konqueror 及 Safari。但是 Opera 不支持 XSL 格式对象, 也不支持 XSLT。

## 19、AJAX 都有哪些优点和缺点？

1、最大的一点是页面无刷新, 用户的体验非常好。

2、使用异步方式与服务器通信, 具有更加迅速的响应能力。

3、可以把以前一些服务器负担的工作转嫁到客户端, 利用客户端闲置的能力来处理, 减轻服务器和带宽的负担, 节约空间和宽带租用成本。并且减轻服务器的负担, ajax 的原则是“按需取数据”, 可以最大程度的减少冗余请求, 和响应对服务器造成的负担。

4、基于标准化的并被广泛支持的技术, 不需要下载插件或者小程序。

ajax 的缺点

1、ajax 不支持浏览器 back 按钮。

2、安全问题 AJAX 暴露了与服务器交互的细节。

3、对搜索引擎的支持比较弱。

4、破坏了程序的异常机制。

5、不容易调试。

# CSSday01

## 一.CSS 概述

### 1.什么是 css

Cascading Style Sheets 层叠样式表，级联样式表，简称样式表
---

### 2.作用

设置 HTML 网页中的元素样式
------------------

### 3.CSS 和 HTML 的关系

HTML：负责网页结构的搭建，内容的展示 CSS：负责网页的修饰
-------------------------------------

### 4.CSS 与 HTML 的属性使用原则

W3C 建议我们尽量使用 css 的方式来取代 html 的属性 css 样式： 1.样式代码高度的重用 2.提高可维护性
--

## 二.CSS 的语法规范

### 1.使用 CSS 的方法

1.内联样式 将 css 的样式定义在 HTML 标签的 style 属性中 <any style="样式声明; 样式声明"></any> 样式声明 样式属性:值 组成 多个样式声明之间用分号分隔
内联样式使用的非常少(禁止使用) 内联样式不能重用 内联样式优先级最高 项目中一般很少使用内联样式，只在学习和测试的时候使用

### 常用的 css 属性

color：设置文本的颜色 取值：合法颜色值(颜色的英文，#a~f*3 #abc #aaf #afa # faa #aff #faf) font-size：设置字体的大小 background:设置背景颜色
2.内部样式 在 <head> </head> 标 签 中 ， 添 加 <style> <style>标签，在其内部定义样式规则 <style> 样式规则 </style> 样式规则：由选择器和样式声明组成

选择器{ 样式声明; 样式声明; } span{ color:#afa; background:#faf; }
内部样式在本页面可以 重用，但是其他页面不能重用 项目中使用不多，但是学习和测试中大量使用
3.外部样式 独立创建一个新文件*.css。 可以在任何网页中引用 <head> <link rel="stylesheet" href="css 文件的 url"> </head> 外部样式是项目中必须使用的样式写法。

### 练习

01_ex.html 中，写 3 个标签 h1--->内联样式 h2--->内部样式 h3--->外部样式 字体颜色，字号大小，背景颜色
--

### 2.CSS 样式的特性

1.继承性 大部分的 css 效果是可以直接被子元素继承的 必须是层级结构
2.层叠性 可以为一个元素定义多个样式规则 规则中属性不冲突(不重复)，可以同时作用到当前元素上
3.优先级 当样式声明冲突时，按照样式规则优先级去应用 默认的优先级，从高到低： 1.内联样式(行内样式)优先级最高 2.内部样式、外部样式，遵循就近原则 3.浏览器默认样式最低
4.调整优先级 ! important 规则 放在属性值之后，与值之间用空格隔开 作用是调整优先级，让没有 important 属性不能覆盖他

## 练习

02\_ex.html 中添加一个 p 元素 lorem, 用内部样式设置文字颜色为蓝色  
字体大小为 24px, 用外部样式设置字体大小为 40px, 字体颜色为红色, 将外部样式引入 html。  
查看页面效果, 然后改变内部样与外部样式引入的位置, 再次查看页面效果、  
使用 !important 调整样式的优先级

## 三.基础选择器 (重点\*\*\*\*\*)

### 1.选择器的作用

规范了页面中哪些元素能够使用定义好的样式  
为了匹配元素

### 2.基础选择器详解

#### 1.通用选择器

\*{样式声明}

\*效率极低, 尽量少用

\*{margin:0;padding:0} 外边距, 内边距清 0

#### 2.元素选择器(标签选择器)

设置页面中一类元素的默认样式

标签名称{样式声明} 例如: p{color:red}

#### 3.类选择器

定义页面上某类元素的公共样式, 谁想用我, 就引用我的 class 名

元素通过 class 属性来引用该样式

.类名{样式的声明}

<any class="类名">

注意: "."不能省略

类名不能以数字开头

类名不能包含特殊字符(可以包含 - \_)

#### 类选择器的特殊用法

##### 1.多类选择器

让一个元素引用多个类选择器

<p class="font\_blue font\_size24">烤生蚝</p>

##### 2.分类选择器

将元素选择器和类型选择器联合使用

元素选择器.类选择器{}

可以匹配到, 指向了某个类的, 具体某个元素

<span class="font\_blue">水煮牛肉</span>

span.font\_blue{background:#faf;}

作用:

1.指向更精确

2.优先级变高

## 练习

03\_ex.html, 添加 div 和 p, 内容随意。  
用类选择器为所有的元素设置字体颜色为红色  
用分类选择器设置 p 元素的背景为黄色

### 4.id 选择器

设置指定元素的样式, 专属定制, 为元素设置 id 值, 使用 #id 值作为选择器, 编写专属样式

### 5.群组选择器

将多个选择器放在一起, 一块定义某公共样式  
选择器 1, 选择器 2, 选择器 3...{公共样式声明}

例: #content,p.mycolor,span{color:red}

### 6.后代选择器

通过元素的后代关系匹配元素

后代: 一层或者多层的嵌套关系

语法: 选择器 1 选择器 2 选择器 3{样式声明}

### 7.子代选择器

通过元素的子代关系匹配元素(一层嵌套)

选择器 1 > 选择器 2 > 选择器 3{样式声明}

## 练习

使用子代和后代选择器, 把三个 a 设置成三种不同字体颜色

```
<div>
  <a></a>
  <p>11111
    <span>
      <a></a>
    </span>
  </p>
</div> <a></a> </div>
</div>
```

### 8.伪类选择器

匹配元素的不同状态下的样式

#### 1.链接伪类

:link 元素尚未访问时的状态样式

:visited 元素访问过后的状态样式

#a1:link{color:red;}

#a1:visited{color:yellow;}

#### 2.动态伪类

:active 匹配元素被激活时的状态

#btn:active{color:purple;}

:focus 匹配元素获取焦点时的状态

#txt:focus{background:#aff;}

:hover 鼠标悬停在元素时的状态

.img:hover{width:100px;}

## 练习

1.04\_ex 添加一个 a 标记, 内容随意, 设置以下样式

- 1.未被访问之前为黑色, 没有下划线 text-decoration:none;
- 2.鼠标悬停时, 文本颜色为红色
- 3.被激活时, 文本颜色为 yellow
- 4.访问过后, 文本颜色 orange

2.页面中添加一个 input 输入框, 默认字体为灰色 gray

字体斜体 font-style:italic;

- 1.被激活时, 字体喝了脉动, 非斜体 (font-style:normal)
- 2.获取焦点时, 文本为红色

### 9.选择器的权值

权值: 标识当前选择器的重要程度, 权值越大优先级越高

!important > 1000

内联样式 1000

id 选择器 100

类和伪类选择器 10

元素选择器 1

\*选择器 0

继承的样式无权值 无

总结:

1.当一个选择器中含有多个选择器时, 需要将所有的选择器的权值进行相加。

然后再进行比较, 权值高的优先显示

注意: 选择器的权值计算不会超过其最大数量级 (1W 个 #id 也不会到 1000)

2.群组选择器的权值, 单独计算, 不会相加

3.样式后面加 !important 该样式会获取最高优先级

内联样式不能添加 !important

4.如果权值相同, 使用就近原则

## 四.尺寸和边框

### 1.尺寸属性

改变元素的宽高

width

最大宽度 max-width

最小宽度 min-width

height

最大高度 max-height

最小高度 min-height

## 附加知识点: 单位

### 1.尺寸单位

px 像素

in 英寸 1in=2.54cm

pt 磅值, 多数用它表示字体大小  
1pt=1/72in

cm mm 厘米, 毫米

em 相对单位, 相对父元素的倍数

rem 相对单位, 相对根元素的倍数(html)

% 相对单位, 相对父元素的百分之多少

### 2 颜色单位

1.颜色的英文单词 red blue yellow black orange pink purple

2.#rrggbb RGB 光学三元色 6 个 16 进制  
#006699

3.#aabbcc--->#abc #f00 #0f0 #00f  
#fff #000 #333

4.rgb(r,g,b); r:0~255 g:0~255 b:0~255  
rgb(255,255,255);白色

5.rgb(r%,g%,b%) 此变形式使用极少

6.rgba(r,g,b,alpha) alpha 透明度

## 周末作业:

### 1.ajax 项目

登录

注册

删除

修改

列表显示

2.css 的 3 种使用方式, 理论背

3.css 的特点, 理论

4.css 的基础选择器 8 种

5.尺寸 rem em 颜色