day04

ajax的异步加载

一.HTTP协议

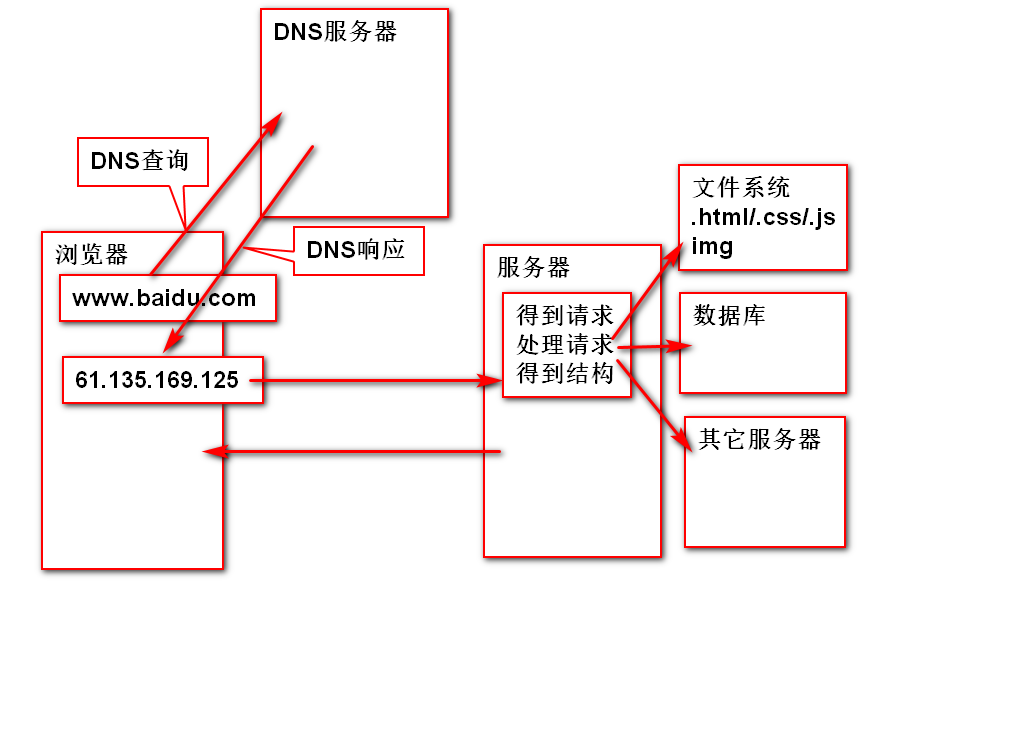
1.url

|  |  |  |
| --- | --- | --- |
| 结构：协议+主机名称+目录结构+文件名称  http://www.codeboy.com/admin/user\_list.html  url完整结构  <scheme>://<user>:<pwd>@<host>:<port>/<path>;<params>?<query>#<frag> | | |
| scheme 方案、协议，以哪种方式获取浏览器的资源  不区分大小写，常见协议，http/https/ftp | | |
| 常见的协议 | 默认端口号 | 基本作用 |
| ftp | 21 | 文件的上传下载 |
| SSH | 22 | 安全的远程登录 |
| TELNet | 23 | 远程登录 |
| SMTP | 25 | 邮件传输协议 |
| DNS | 53 | 域名解析 |
| HTTP | 80 | 超文本传输 |
| POP3 | 110 | 邮件接收 |
| HTTPS | 443 | 加密的传输http |
| <scheme>://<user>:<pwd>@<host>:<port>/<path>;<params>?<query>#<frag>  user：用户名  pwd：密码  host：主机名 localhost 127.0.0.1  port：端口号  path：资源在服务器上存放的位置  params: 参数，服务器端使用  query: 查询字符串 ?unmae=lilei&upw=1234567  frag:锚点 | | |

2.HTTP协议

|  |
| --- |
| HTTP:HyperText Transfer Protocol 超文本传输协议  规范了数据是如何打包及传递的 |
| http的历史 现在使用的版本是http/1.1 不需要设置 |

3.详解



|  |
| --- |
| HTTP协议规定，客户端与服务器端传递的数据块Message（消息/报文）。必须符合特定的格式，这个Message才能被客户端和服务器彼此理解    1.请求消息(请求起始行，请求头，请求主体)  浏览器要发送给服务器的数据  2.响应消息(响应起始行，响应头，响应主体)  服务器发送给浏览器的数据 |

**Request请求消息**

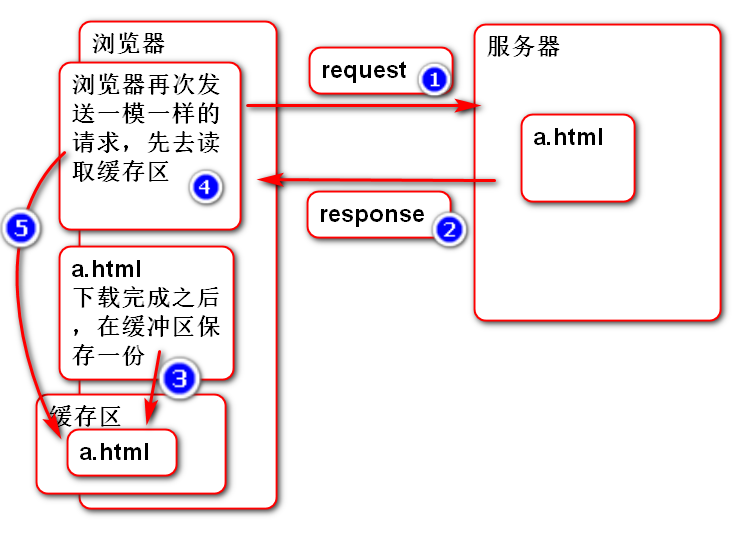
|  |
| --- |
| 客户端要发送给服务器的数据  由三部分组成 请求起始行，请求头，请求主体 |
| 1.请求起始行  -请求方法  -请求url  -协议版本 |
| 请求方法  get 客户端向服务器要数据的时候使用  靠地址栏传递查询字符串，无请求主体  post 客户端向服务器提交数据的时候使用  隐式发送。有请求主体(form data)  delete 表示客户端可以删除服务器上的内容(一般禁用)  put 表示客户端可以往服务器上放数据(一般禁用)  connect 测试连接  trace 追踪请求路径  option 选项  head 表示客户端只回去响应的消息头 |
| url 访问的地址链接  协议版本 http/1.1 |
| 2.请求头 Request Headers  host:告诉服务器，请求的是哪一个主机  Connection: keep-alive 告诉服务器进行持久连接  User-Agent:告诉服务器自己(浏览器)的类型  Accept-Encoding:gzip,deflate,告诉服务器，自己能接收的压缩文件类型  Accept-Language:告诉服务器自己能接收的自然语言的类型  Referer:引用/推荐人，告诉服务器请求来自哪个网页 |
| 3.请求主体 Form data |

**Response响应消息**

|  |
| --- |
| 服务器端发送给客户端的数据，  由三步组成，响应起始行，响应头，响应主体 |
| 1. 响应起始行  协议版本 http/1.1  响应状态码  原因短句，对状态码的解释说明 |
| 响应状态码  告诉客户端浏览器，服务器的响应状态是什么  取值：  1XX:100-199 提示信息  2XX:成功响应 200 ok  3xx：301 永久重定向 www.sun.com  302 临时重定向  304 请求未被修改，还是原来的请求，命中缓存  4XX：客户端请求的错误  404 NOT Found请求资源不存在  403 Forbidden 权限不够  405 Method not Allowed 请求方法不被允许  5XX：服务器运行错误 |
| 原因短句，就是对状态码的解释 |
| 2.响应头  Date 告诉浏览器，服务器的响应时间，格林威治时间 +8h是北京时间  Connection:keep-alive 告诉浏览器已经启动了持久连接  Content-Type:响应主体的类型  text/html 响应回来的数据是html文本  text/plain 普通文本  text/css样式文件  application/javascript js文件  image/png(jpeg.gif) 图片  application/json 响应回来的是json格式的字符串  application/xml 响应回来的是xml格式的字符串 |
| 3.响应主体 响应回来的数据  network中，在header的右边第二个 |

**缓存**

1.什么是缓存



2.缓存的优点

|  |
| --- |
| 1.减少冗余的数据传输，节省客户端的流量  2.节省服务器带宽  3.降低了对服务资源的消耗和运行需求  4.降低了远距离传输而造成的延迟加载 |
| 缓存的原理    1.请求--无缓存--访问服务器--存缓存--客户端得到数据  2.请求--有缓存--够新鲜--命中缓存--客户端得到数据  3.请求--有缓存--不新鲜--访问服务器看是否过期--没过期--更新缓存的新鲜度--客户端得到数据  4.请求--有缓存--不新鲜--访问服务器看是否过期--已过期--访问服务器--存缓存--客户端得到数据 |

3.与缓存相关的消息头

|  |
| --- |
| Cache-Control:http1.1的用法  从服务器将文档传到客户端之时起，  此文档处于新鲜的秒数，这个秒数是一个相对时间  语法：0 不缓存  max-age= 新鲜的秒数  Expires http1.0用法  指定过期的确切时间，是一个具体的时间点 |

4.网页如何添加缓存

|  |
| --- |
| <meta http-equiv="消息头" content="消息头对应的值">  ex  <meta http-equiv=" Cache-Control " content="max-age=3600"> |

5.Http性能优化

|  |
| --- |
| 1.http连接的过程  发起请求-->建立连接-->处理请求-->访问资源-->构建响应-->发送响应-->记录日志 |
| 2.http连接性能优化的思路  2.1减少请求次数  2.2减少连接创建次数(开启持久连接)  2.3提高服务器端的运行速度  2.4尽可能减少响应数据的长度 |
| 3.安全的HTTP协议  HTTPS，安全版本的http  SSL：为数据通信提供了安全支持  1.客户端发送请求消息时，在SSL层加密  服务器接收到加密文件，在SSL层解密  得到请求明文，对请求进行处理  2.服务器发送响应消息时，在SSL层加密  客户端接收到加密文件，在SSL层解密  得到响应明文，解析相应内容 |

二.DOM操作

1.完整的javaScript操作的组成

|  |
| --- |
| 1.js核心：ECMA Script ECMA--欧洲计算机制造联合会  2.DOM：Document Object Model 文档对象模型  让js动态的操作页面上的元素  3.BOM：Browser Object model 浏览器对象模型  让js动态的操作浏览器 |
| .DOM，这个阶段，获取页面上的元素，改变这个元素的内容 |

2.简单DOM操作

|  |
| --- |
| 1.要获取的元素一定要有id  <div id="d1">这是测试的文本！！</div> |
| 2.在js中，用id获取对象  document.getElementById("d1");  H5支持简写方式  d1就是div对象 |
| 3.获取/设置元素的内容 innerHTML  d1.innerHTML=...  var str=d1.innerHTML;  innerHTML指的是，双标签，开关标签之间的所有内容。  如果开关标签之间有其它元素,会得到该元素的文本字符串  如果设置innerHTML时，添加了新标签，在语法正确的情况下，新标签也会在页面上渲染 |
| 4.事件 用户与页面交互行为，调用什么js代码 |

作业：

复习nodejs接口写法

|  |
| --- |
| 在网页中创建一个div，内容随意  添加一个按钮，点击是获取div中文本并打印在控制台  再添加一个按钮，点击是将div中文本内容修改 |
| 在网页中创建一个div。内容随意  添加一个按钮，在div中动态生成一个3\*3的，内容随意的表格 |

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 |