1. 常见的HTTP状态码你了解多少？描述一下状态码的含义。
   1. 200 ：请求成功
   2. 301 ：永久移动。请求的资源已被永久的移动到新URI，返回信息会包括新的URI，浏览器会自动定向到新URI。今后任何新的请求都应使用新的URI代替
   3. 302 ：临时移动。与301类似。但资源只是临时被移动。客户端应继续使用原有URI
   4. 303：查看其它地址。与301类似。使用GET和POST请求查看
   5. 403：服务器理解请求客户端的请求，但是拒绝执行此请求
   6. 404：服务器无法根据客户端的请求找到资源（网页）。通过此代码，网站设计人员可设置"您所请求的资源无法找到"的个性页面
   7. 503：由于超载或系统维护，服务器暂时的无法处理客户端的请求。延时的长度可包含在服务器的Retry-After头信息中
2. 什么情况下会遇到跨域，描述一下前端常见处理跨域的几种方式。并封装一个jsonp原理

跨域：协议，域名，端口这三个，有任意一个不一样就算跨域  
处理跨域：1.后端配合我们进行跨域。

Pan.baidu.com===zhidao.baidu.com

(1)JSONP（正常情况下，返回的数据都是JSON格式，JSONP是一种特殊的格式）

(2)后端设置属性Acess-Control-Allow-Origin以支持跨域

2.后端不配和我们进行跨域

（3）iframe(只能显示，不能控制)

<iframe src="http://www.baidu.com"></iframe>

（4）通过后端代理（自己的后端）

1. Web前端应该从哪些方面来优化网站性能

 1. 减少 HTTP请求数浏览器端存储有哪些，并描述他们的区别：(1). 从设计实现层面简化页面，(2). 合理设置 HTTP缓存，(3). 资源合并与压缩，(4). CSS Sprites，(5). Inline Images，(6). Lazy Load Images

1. 将外部脚本置底（将脚本内容在页面信息内容加载后再加载）
2. 异步执行 inline脚本(其实原理和上面是一样，保证脚本在页面内容后面加载。)
3. Lazy Load Javascript（只有在需要加载的时候加载，在一般情况下并不加载信息内容。）
4. 异步请求 Callback（就是将一些行为样式提取出来，慢慢的加载信息的内容）
5. 说get和post请求

 1.是基于什么前提的？如果什么前提都没有，不使用任何规范，只考虑语法和理论上的HTTP协议

GET和POST几乎没有什么区别，只有名字不一样。

2.如果是基于RFC规范的。

(1)理论上的（Specification）：GET和POST是具有相同语法的，但是有不同的语义。GET是用来获取数据的，POST是用来发送数据的，其他方面没有区别。

(2)实现上的(Implementation):各个浏览器，就是这个规范的实现者。

常见的那些不同：1）GET数据在url是可见的，post请求是不显示在url里面的。url并不一定在地址栏里显示

2）GET对长度是有限制的，POST长度是无限的。

3）GET请求的数据可以收藏为书签，POST请求到的数据不可以收藏为书签

4）GET请求后，按后退按钮，刷新按钮无影响，post数据会被重新提交

5）GET的编码类型：application/x-www-form-url,POST的编码类型：有很多种。Encodeapplication/x-www-form-urlencoded,multipart/form-data

6)GET的历史参数会被保留在浏览器里，POST是不会被保存到浏览器中的

7）GET只允许ASCII编码，POST是没有编码限制的，甚至是允许发二进制的。

8）GET与POST相比，GET的安全性较差,因为所发的数据是url的一部分。

1. 一个页面从输入 URL 到页面加载完的过程中都发生了什么事情?

比如：输入的是1123.xxx

1.看浏览器的缓存

2.本机host：127.0.0.1 localhost；0.0.0.0表示不知道IP

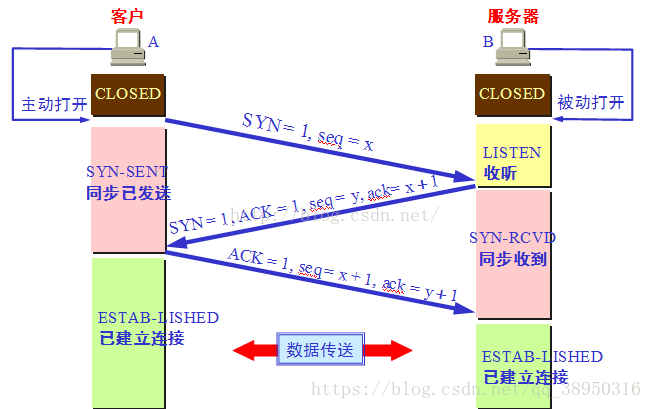
3.家里路由器

4.上级路由，城市的LDNS服务器

5.继续向上级的DNS服务器找，只到找到gDNS服务器

1. 请描述TCP三次握手与四次挥手？

三次握手：

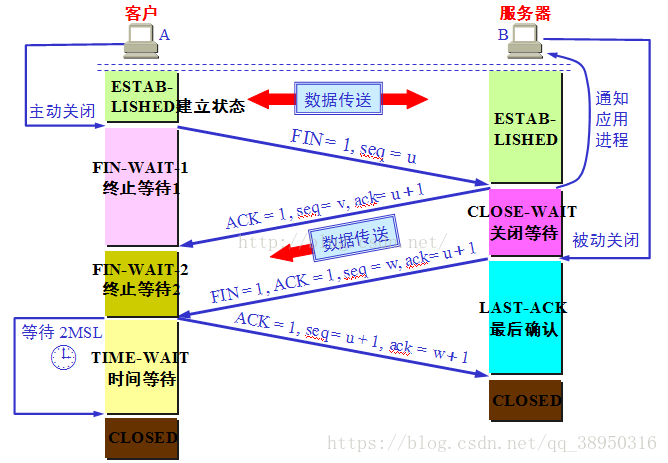


第一次握手：建立连接时，客户端发送syn包（syn=j）到服务器，并进入SYN\_SENT状态，等待服务器确认；SYN：同步序列编号（Synchronize Sequence Numbers）。

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK(ack=k+1），此包发送完毕，客户端和服务器进入ESTABLISHED（TCP连接成功）状态，完成三次握手。

四次挥手：

  
1）客户端进程发出连接释放报文，并且停止发送数据。释放数据报文首部，FIN=1，其序列号为seq=u（等于前面已经传送过来的数据的最后一个字节的序号加1），此时，客户端进入FIN-WAIT-1（终止等待1）状态。 TCP规定，FIN报文段即使不携带数据，也要消耗一个序号。

2）服务器收到连接释放报文，发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号seq=v，此时，服务端就进入了CLOSE-WAIT（关闭等待）状态。TCP服务器通知高层的应用进程，客户端向服务器的方向就释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。

3）客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的数据）。

4）服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，ack=u+1，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq=w，此时，服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。

5）客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是seq=u+1，此时，客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2∗∗MSL（最长报文段寿命）的时间后，当客户端撤销相应的TCB后，才进入CLOSED状态。

6）服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的TCP连接。可以看到，服务器结束TCP连接的时间要比客户端早一些。

1. TCP与UDP的区别

|  |  |  |
| --- | --- | --- |
|  | UDP | TCP |
| 是否连接 | 无连接 | 面向连接 |
| 是否可靠 | 不可靠传输，不使用流量控制和拥塞控制 | 可靠传输，使用流量控制和拥塞控制 |
| 连接对象个数 | 支持一对一，一对多，多对一和多对多交互通信 | 只能是一对一通信 |
| 传输方式 | 面向报文 | 面向字节流 |
| 首部开销 | 首部开销小，仅8字节 | 首部最小20字节，最大60字节 |
| 适用场景 | 适用于实时应用（IP电话、视频会议、直播等） | 适用于要求可靠传输的应用，例如文件传输 |
| TCP向上层提供面向连接的可靠服务 ，UDP向上层提供无连接不可靠服务。  虽然 UDP 并没有 TCP 传输来的准确，但是也能在很多实时性要求高的地方有所作为  对数据准确性要求高，速度可以相对较慢的，可以选用TCP | | |

1. http是什么?

HTTP是一个简单的请求-响应bai协议，它通常运du行在TCP之上。它zhi指定了客户端可能发送给服务器dao什么样的消息以及得到什么样的响应。请求和响应消息的头以ASCII码形式给出；而消息内容则具有一个类似MIME的格式。

1. http与https的区别

“https协议需要到CA申请证书，一般免费证书较少，因而需要一定费用。http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl/tls加密传输协议。”

1. 重排（回流）和重绘是什么?什么情况下会触发重排和重绘。

 DOM的变化影响到了元素的几何属性（宽高）,浏览器重新计算元素的几何属性，其他元素

的几何属性和位置也会受到影响，浏览器需要重新构造渲染树，这个过程称为重排，浏览器将受

到影响的部分

重新绘制到屏幕上的过程称为重绘。引起重排的原因有

     1.添加或者删除可见的DOM元素，

     2.元素位置、尺寸、内容改变，

     3.浏览器页面初始化，

     4.浏览器窗尺寸改变，重排一定重绘，重绘不一定重排

减少重绘和重排的方法：

    1.不在布局信息改变时做 DOM 查询

    2.使用 cssText 或者 className 一次性改变属性

    3.使用 fragment

    4.对于多次重排的元素，如动画，使用绝对定位脱离文档流，让他的改变不影响到其他元素