1. 在浏览器地址栏输入一个 URL 后回车，背后发生了什么

先会从域名解析服务器中解析域名对应的ip，由本机的hosts开始，到路由器，一层层的查找ip

找到ip后，会构造http包，通过tcp协议传输

1. 网络七层架构

* 物理层 比特流传输
* 数据链路层，以帧为单位传输数据，包含了mac地址的封装和解封等
* 网络层，处理ip地址的封装和解封，协议代表有ip、ipx、rip、ospf
* 传输层，端到端的数据传输，把上层数据分组封装，包含五元组信息，用ip协议发送，个人理解，ip协议是主机和主机间数据传输，数据传输后分发给哪个应用，就是由传输层的协力来处理，五元组信息中包含了源端口和目标端口
* 会话层 (抽象),其实传输层已经能完全的传输数据了，数据的处理和使用，这个就是业务上的事情了，所有后面的三层其实可以理解为业务上的分层，会话层可以用http的session来理解，用来判断是同一个会话，处理会话的连接，数据发送，关闭，协议有rpc协议
* 表现层，就是一些数据的处理，最后处理成易于给用户看到的数据的层面，比如说ssl，处理数据的加密解密，常见的协议有 ASCII、SSL/TLS
* 应用层，常见的协议有 HTTP，FTP，TELNET、SMTP 等。简单理解就是能让用户直接使用的层面，网页访问，文件共享，邮件发送

1. http报文组成

请求行：get /dir/html HTTP/1.1

请求头：Host:www.test.com

请求体: name=test&age=12

请求行中包含了请求方法、uri、协议版本

1. post请求发送两次

这个如果是简单请求，只会发送一次，复杂请求，会先发送一个option请求，再发post请求

简单请求：

请求方法是HEAD、GET、POST

只包含简单HTTP请求头，即：

·Accept,

·Accept-Language,

·Content-Language,

·Content-Type并且值是 application/x-www-form-urlencoded, multipart/form-data, 或者 text/plain之一的（忽略参数）。tcp连接三次握手和四次挥手

1. 跨域

跨域是指不符合浏览器的同源策略的资源访问

同源策略：域名、协议、端口，这三个都相同才算同源，子域名不同也是算不同源

1. 如何解决跨域问题

* JSONP

Script标签src不受同源策略影响，并且加载后会直接执行，所以可以把接口放在Script标签的src上，直接返回一段js方法调用代码，入参就是要返回的数据，就可以直接访问js方法，把数据传递过去

正常接口数据:{“result”:[1,2,3]}

定义个js方法:get\_data(data);

JSONP 接口返回数据get\_data({“result”:[1,2,3]})，这段放在Script 标签下就能直接调用get\_data方法传递数据过去



这段代码就是jsonp的实现，构造一个script标签，把请求地址放到src上，请求返回的数据用getData包装着，就能直接执行已定义好的getData方法，把数据传递过去

Jsonp方式的缺点是，只能用在get方法上

* CORS

Access-Control-Allow-Origin响应头返回前端的协议域名端口对应的url即可，浏览器会判断为可跨域

* 反向代理

可以通过反向代理，让前后端代理到同一个域名地址下，在浏览器看来就是不跨域的。

* Websocket

这个不受同源策略影响

跨域的话会有很多其他的问题，比如cookie的传递等

1. 协议五元组
2. tcp滑动窗口
3. tcp协议和udp协议区别
4. 协议1.1和1.2的区别
5. 请求体格式/http参数传递几种格式

multipart/form-data、x-www-form-urlencode、json、querystring几种传递参数方式

1. 各种请求方法的含义，特殊请求方法OPTIONS、PATCH
2. restfull标准
3. http常用状态码及含义
4. http常用请求头（以及项目中使用到的请求头）
5. nginx七层转发和lvs五层转发
6. 跨域问题、原理（几种 使用js 代理 vue可以使用代理进行跨域访问 发布时候可以使用nginx代理来处理跨域访问）
7. 消息边界
8. Tcp四次挥手讲解一下过程，最后一次ack如果客户端没接收到会怎么办

没接收到，到了指定时间(2分钟，这个是系统配置的一个网络时间)，会自动把状态转换为关闭

1. 对于socket编程，accept方法是干什么的，在三次握手中属于第几次

Accept是接收已经握手成功的socket的，所以是在第三次握手之后

Connect是进行socket连接的，是在主动连接端，所以是在第二次握手之后就返回。