# 前端http知识点

## 1、前端从url到页面显示的过程

参考：https://www.cnblogs.com/xianyulaodi/p/6547807.html

1. 输入网址
2. Dns解析
3. 建立tcp链接
4. 客户端发送http请求
5. 服务器处理请求
6. 服务器响应请求
7. 浏览器展示HTML
8. 浏览器发送请求获取其它在html中的资源
9. 输入网址：当我们开始在浏览器中输入网址是，浏览器其实已经在智能匹配可能得到的url。他会从历史记录、书签等地方找到已经输入的字符串可能对应的url，然后废除智能提示。
10. Dns解析：  
    请求一旦发起，浏览器首先要做的事情就是解析这个域名，一般来说浏览器会首先查看本地硬盘的hosts文件，看看其中有没有和这个域名对应的规则，如果有的话就直接使用hosts文件中对应的ip地址

如果本地hosts文件中没有找对应的ip地址，浏览器会发出一个DNS请求到本地DNS服务器。本地DNS服务器一般都是你的网络接入商提供，比如电信，移动等  
查询你输入的网址的Dns请求到达本地dns服务器之后，本地dns服务器会首先查询它的缓存记录，如果缓存记录中有此条记录，就可以直接返回结果，此过程是递归的方式进行查询。如果没有，本地dns服务器还要向dns根服务器进行查询

根服务器没有记录具体的域名和ip地址的对应关系，而是告诉本地Dns服务器，你可以到域服务器上去查询，并给出域服务器的地址，这种过程是迭代进行的。

本地DNS服务器继续向域服务器发出请求，在这个例子中请求的对象是.com域服务器.com.服务器收到请求后，也不会直接返回域名和IP地址的对应关系。而是告诉DNS服务器你的域名解析服务器的地址。

最后DNS服务器向域名的解析服务发出请求，这时就能收到一个域名和ip地址的对应关系，本地DNS服务器不仅要把ip地址返回给用户电脑，还要把这个对应关系保存在缓存中，以备下次别的用户查询时，可以直接返回结果，加快网络访问

（扩展）

什么事dns

Dns是因特网上作为域名和ip地址相互映射的一个分布式数据库，能够使用户更方便的访问互联网，而不用去记住能够被机器直接读取的ip数串，通过主机、域名，最终得到该主机名对应的ip地址的过程叫做域名解析  
DNS查询的两种方式：递归查询和迭代查询

递归查询：当局部DNS服务器自己不能回答客户机的DNS查询时，他需要向其它DNS服务器进行查询，局部DNS服务器自己负责向其它DNS服务器进行查询，一般是先向该域名的根服务器进行查询，再有根服务器一级级的向下查询，最后得到的查询结果返回给局部DNS服务器再有局部DNS服务器返回给客户端

迭代查询：局部DNS服务器不是自己向其它Dns服务器进行查询，而是把把能解析该域名的其它DNS服务器的地址返回给客户端DNS程序，客户端DNS程序在继续向这些Dns服务器进行查询，知道查询出结果为止。  
DNS域名称空间的组织方式：我们在前面说的根DNS服务器，域DNS服务器，这些都是域名称空间的组织方式。按其功能命名空间中用来描述DNS域名称的五个类别（如下例）



DNS负载均衡

当一个网站有足够多的用户的时候，假如每次请求的资源都位于一台服务器的上面，那么这台机器随时可能会崩掉，处理方法就是用DNS负载均衡技术，它的原理实在DNS服务器中为一个主机名配置多个ip地址，在应答DNS查询时，DNS服务器对每个查询将以DNS文件中主机记录的ip地址安顺序返回不同的返回结果，将客户端的访问引导到不同的机器上去，使得不同的客户端访问不同的服务器，从而达到负载均衡的目的

1. 浏览器向web服务器发送一个http请求

拿到域名对应的ip地址后，浏览器会以一个随机端口（1024<端口<65535）向服务器的web程序（常用的有httpd、nginx）80端口发起tcp链接请求。这个链接请求到达服务器端后（这中间通过各种路由设备，局域网内除外），进入到网卡，然后进入到内核的TCP/IP协议栈（用于识别该链接请求，解封包，一层一层的剥开），还有可能要经过Netfilter防火墙（属于内核的模块）的过滤，最终达到web程序，最终简历tcp/ip链接

建立了tcp链接之后，发起一个http请求。一个典型的http request header 一般需要包含请求的方法，例如get，post等客户端向服务器发起请求的时候会带一些信息，请求方法，请求头，请求正文

1. 浏览器显示html：

在浏览器没有完整接收全部HTML问件时，他就已经开始显示这个页面了，浏览器是如何把页面呈现在屏幕上的呐？不同的浏览器可能解析的不一样，下面是webkit的渲染过程。  
解析html以构建dom树->构render树->布局render树->绘制render树，  
浏览器在解析html文件时，会自上而下加载，并在加载过程中进行解析渲染。在解析过程中，如果遇到请求外部资源时，如图片，链接等请求过程是异步的，并不会影响html文档进行加载。  
解析过程中浏览器会首先解析HTML文件构建dom树，然后解析css文件构建渲染树，等到渲染树构建完成后，浏览器开始布局渲染树并将其绘制到屏幕上，涉及两个概念：回流（reflow）和重绘（repain）

DOM节点中的各个元素都是以盒模型的形式存在，这些都需要浏览器去计算其位置和大小等，这个过程称为回流;当盒模型的位置,大小以及其他属性，如颜色,字体,等确定下来之后，浏览器便开始绘制内容，这个过程称为重绘。

　　页面在首次加载时必然会经历reflow和repain。reflow和repain过程是非常消耗性能的，尤其是在移动设备上，它会破坏用户体验，有时会造成页面卡顿。所以我们应该尽可能少的减少reflow和repain。

当文档加载过程中遇到js文件，html文档会挂起渲染（加载解析渲染同步）的线程，不仅要等待文档中js文件加载完毕，还要等待解析执行完毕，才可以恢复html文档的渲染线程。因为JS有可能会修改DOM，最为经典的document.write，这意味着，在JS执行完成前，后续所有资源的下载可能是没有必要的，这是js阻塞后续资源下载的根本原因。所以我明平时的代码中，js是放在html文档末尾的。

JS的解析是由浏览器中的JS解析引擎完成的，比如谷歌的是V8。JS是单线程运行，也就是说，在同一个时间内只能做一件事，所有的任务都需要排队，前一个任务结束，后一个任务才能开始。但是又存在某些任务比较耗时，如IO读写等，所以需要一种机制可以先执行排在后面的任务，这就是：同步任务(synchronous)和异步任务(asynchronous)。

　　JS的执行机制就可以看做是一个主线程加上一个任务队列(task queue)。同步任务就是放在主线程上执行的任务，异步任务是放在任务队列中的任务。所有的同步任务在主线程上执行，形成一个执行栈;异步任务有了运行结果就会在任务队列中放置一个事件；脚本运行时先依次运行执行栈，然后会从任务队列里提取事件，运行任务队列中的任务，这个过程是不断重复的，所以又叫做事件循环(Event loop)。具体的过程可以看我这篇文章：点击这里

## 2、http协议

参考：<https://blog.csdn.net/ahou2468/article/details/105791871/>

https://blog.csdn.net/weixin\_41957484/article/details/105180285

### 1、http协议是什么

http协议是一种网络传输协议，要实现信息传递，信息收发双方需要遵循某一致的标准和规范，http协议就是这样一种规范。http是基于tcp/ip协议簇来传递数据的

## 2、http请求的三次握手四次挥手

1、第一次握手：客户端向服务端发起建立链接的请求。

2、第二次握手：服务端接收客户端的链接请求，向客户端发送确认收到，并且向客户端发送建立链接请求

3、第三次握手：客户端收到服务器端的确认信息和链接请求，向服务器端发送确认收到，此时链接建立

三次握手详解：

1. 客户端发送syn=1，随机产生seq number=1234567的数据包到服务器，服务器由SYN=1知道客户端要求建立联机（客户端：我要链接你）
2. 服务器端收到请求后要确认联机信息，向A发送ack number=（客户端seq+1），syn=1，ack=1，随机产生seq=7654321的包（服务器：好的你来连吧）
3. 客户端收到后检查ack number是否正确，即第一次发送的seq number+1，以及位码ack是否为1，若正确，客户端会再发送ack number = （服务器的seq+1），ack=1，服务器收到后确认seq值与ack=1则链接建立成功（客户端：好的，我来了）
4. 第一次挥手：客户端向服务器端发起关闭链接请求
5. 第二次挥手：服务器端接收到关闭链接的请求，向客户端发送确认收到。但此时服务器不能像接受到连接请求后同时向客户端发送确认收到和连接请求一样同时向客户端发送确认接收和关闭请求，因为此时的服务器端有可能还有数据未响应完成，不能即可与客户端断开连接，需要一个等待时间处理，所以此时客户端处于FIN\_WA\_2状态
6. 第三次挥手：服务器端处理完数据可以断开链接了，向客户端发送断开链接请求
7. 第四次挥手：客户端接受到服务器端的断开连接请求，向服务器端发送确认收到，服务器端断开链接，客户端断开与服务器端的链接。

为什么TIME\_WAIT状态需要经过2MSL（最大报文段生存时间），才能返回到close状态

## https

基于HTTP协议，通过SSL或TLS提供加密处理数据、验证对方身份以及数据完整性保护

通过抓包可以看到数据不是明文传输，而且HTTPS有如下特点：

1. 内容加密：采用混合加密技术，中间者无法直接查看明文内容
2. 验证身份：通过证书认证客户端访问的是自己的服务器
3. 保护数据完整性：防止传输的内容被中间人冒充或者篡改