**WEB面试题**

**目录**

1. 计算机网络体系结构 ....................................................................................................................................... 3
2. ISO/OSI网络体系结构
3. 七层网络通信协议
4. 网络通信的模式 ............................................................................................................................................... 6
5. B/S （Browser/Server，[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)/[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)模式）
6. C/S （Client/Server，客户端/[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)模式）
7. B/S和C/S的区别
8. TCP和UDP协议 ............................................................................................................................................ 8
9. TCP
10. UDP
11. 数据包
12. http协议 .......................................................................................................................................................... 10
13. 工作原理
14. 请求报文格式
15. 响应报文格式
16. 状态码
17. 输入url后的加载过程 ................................................................................................................................... 13
18. 浏览器内部工作原理 ...................................................................................................................................... 15
19. 浏览器的主要组件
20. 渲染引擎及工作流
21. 重排与重绘 ...................................................................................................................................................... 16
22. 页面性能优化 .................................................................................................................................................. 16
23. 移动端性能优化 .............................................................................................................................................. 17
24. 为什么利用多个域名来存储网站资源会更有效？ .................................................................................... 18
25. 网页重构 ........................................................................................................................................................... 19
26. SEO .................................................................................................................................................................... 21
27. W3C标准 ......................................................................................................................................................... 21
28. 浏览器内核与兼容性 .................................................................................................................................... 21
29. 内核类别
30. 内核的主要功能
31. 什么是微格式？在前端构建中应该考虑微格式吗？ ................................................................................ 22
32. WEB应用从服务器主动推送Data到客户端有那些方式？ ................................................................... 23
33. 常见web安全及维护管理 ............................................................................................................................ 23
34. sql注入原理及防范
35. XSS原理及防范
36. CSRF原理及防范
37. 什么样的前端代码是好的? ........................................................................................................................... 25

### 一. 计算机网络体系结构

### ISO/OSI网络体系结构

国际标准化组织ISO(International Standards Organization)在80年代提出的开放系统互联参考模型OSI(Open System Interconnection)，这个模型将计算机网络通信协议分为七层。这个模型是一个定义异构计算机连接标准的框架结构，其具有如下特点：

1. 网络中异构的每个节点均有相同的层次，相同层次具有相同的功能。
2. 同一节点内相邻层次之间通过接口通信。
3. 相邻层次间接口定义原语操作，由低层向高层提供服务。
4. 不同节点的相同层次之间的通信由该层次的协议管理，
5. 每层次完成对该层所定义的功能，修改本层次功能不影响其它层、
6. 仅在最低层进行直接数据传送。
7. 定义的是抽象结构，并非具体实现的描述。

b. 七层网络通信协议

1. **物理层(Physical Layer)**

物理层建立在物理通信介质的基础上，作为系统和通信介质的接口，用来实现数据链路实体间透明的比特 (bit) 流传输。只有该层为真实物理通信，其它各层为虚拟通信。物理层实际上是设备之间的物理接口，物理层传输协议主要用于控制传输媒体。

1. **数据链路层(Data Link Layer)**

数据链路层为网络层相邻实体间提供传送数据的功能和过程；提供数据流链路控制；检测和校正物理链路的差错。物理层不考虑位流传输的结构，而数据链路层主要职责是控制相邻系统之间的物理链路，传送数据以帧为单位，规定字符编码、信息格式，约定接收和发送过程，在一帧数据开头和结尾附加特殊二进制编码作为帧界识别符，以及发送端处理接收端送回的确认帧，保证数据帧传输和接收的正确性，以及发送和接收速度的匹配，流量控制等。

1. **网络层(Net Work Layer)**

广域网络一般都划分为通信子网和资源子网，物理层、数据链路层和网络层组成通信子网，网络层是通信子网的最高层，完成对通信子网的运行控制。网络层和传输层的界面，既是层间的接口，又是通信子网和用户主机组成的资源子网的界限，网络层利用本层和数据链路层、物理层两层的功能向传输层提供服务。

数据链路层的任务是在相邻两个节点间实现透明的无差错的帧级信息的传送，而网络层则要在通信子网内把报文分组从源节点传送到目标节点。在网络层的支持下，两个终端系统的传输实体之间要进行通信，只需把要交换的数据交给它们的网络层便可实现。至于网络层如何利用数据链路层的资源来提供网络连接，对传输层是透明的。

网络层控制分组传送操作，即路由选择，拥塞控制、网络互连等功能，根据传输层的要求来选择服务质量，向传输层报告未恢复的差错。网络层传输的信息以报文分组为单位，它将来自源的报文转换成包文，并经路径选择算法确定路径送往目的地。网络层协议用于实现这种传送中涉及的中继节点路由选择、子网内的信息流量控制以及差错处理等。

1. **传输层(Transport Layer)**

从传输层向上的会话层、表示层、应用层都属于端一端的主机协议层。传输层是网络体系结构中最核心的一层，传输层将实际使用的通信子网与高层应用分开。从这层开始，各层通信全部是在源与目标主机上的各进程间进行的，通信双方可能经过多个中间节点。传输层为源主机和目标主机之间提供性能可靠、价格合理的数据传输。具体实现上是在网络层的基础上再增添一层软件，使之能屏蔽掉各类通信子网的差异，向用户提供一个通用接口，使用户进程通过该接口，方便地使用网络资源并进行通信。

1. **会话层(Session Layer)**

会话是指两个用户进程之间的一次完整通信。会话层提供不同系统间两个进程建立、维护和结束会话连接的功能；提供交叉会话的管理功能，有一路交叉、两路交叉和两路同时会话的3种数据流方向控制模式。会话层是用户连接到网络的接口。

1. **表示层(Presentation Layer)**

表示层的目的是处理信息传送中数据表示的问题。由于不同厂家的计算机产品常使用不同的信息表示标准，例如在字符编码、数值表示、字符等方面存在着差异。如果不解决信息表示上的差异，通信的用户之间就不能互相识别。因此，表示层要完成信息表示格式转换，转换可以在发送前，也可以在接收后，也可以要求双方都转换为某标准的数据表示格式。所以表示层的主要功能是完成被传输数据表示的解释工作，包括数据转换、数据加密和数据压缩等。表示层协议主要功能有：为用户提供执行会话层服务原语的手段；提供描述负载数据结构的方法；管理当前所需的数据结构集和完成数据的内部与外部格式之间的转换。例如，确定所使用的字符集、数据编码以及数据在屏幕和打印机上显示的方法等。表示层提供了标准应用接口所需要的表示形式。

1. **应用层(Application Layer)**

应用层作为用户访问网络的接口层，给应用进程提供了访问OSI环境的手段。

应用进程借助于应用实体 (AE)、实用协议和表示服务来交换信息，应用层的作用是在实现应用进程相互通信的同时，完成一系列业务处理所需的服务功能。

### 二. 网络通信的模式

1. B/S （Browser/Server，[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)/[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)模式）

B/S结构（Browser/Server，[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)/[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)模式），是[WEB](http://baike.baidu.com/item/WEB" \t "http://baike.baidu.com/item/_blank)兴起后的一种网络结构模式，WEB浏览器是[客户端](http://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF" \t "http://baike.baidu.com/item/_blank)最主要的[应用软件](http://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6" \t "http://baike.baidu.com/item/_blank)。这种模式统一了[客户端](http://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF" \t "http://baike.baidu.com/item/_blank)，将系统功能实现的核心部分集中到[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)上，简化了系统的开发、维护和使用。客户机上只要安装一个[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)，[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)安装[SQL Server](http://baike.baidu.com/item/SQL Server" \t "http://baike.baidu.com/item/_blank)、[Oracle](http://baike.baidu.com/item/Oracle" \t "http://baike.baidu.com/item/_blank)、MYSQL等数据库。[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)通过Web Server同数据库进行数据交互。

1. C/S （Client/Server，客户端/[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)模式）

C/S是[Client/Server](http://baike.baidu.com/item/Client/Server" \t "http://baike.baidu.com/item/_blank)的缩写。[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)通常采用高性能的PC、工作站或[小型机](http://baike.baidu.com/item/%E5%B0%8F%E5%9E%8B%E6%9C%BA" \t "http://baike.baidu.com/item/_blank)，并采用大型[数据库系统](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/item/_blank)，如Oracle、Sybase、Informix或 SQL Server。[客户端](http://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF" \t "http://baike.baidu.com/item/_blank)需要安装专用的客户端软件。

1. B/S和C/S的区别

**1．硬件环境不同：**

C/S 一般建立在专用的网络上，小范围里的网络环境，[局域网](http://baike.baidu.com/item/%E5%B1%80%E5%9F%9F%E7%BD%91" \t "http://baike.baidu.com/item/_blank)之间再通过专门[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "http://baike.baidu.com/item/_blank)提供连接和数据交换服务。

B/S 建立在[广域网](http://baike.baidu.com/item/%E5%B9%BF%E5%9F%9F%E7%BD%91" \t "http://baike.baidu.com/item/_blank)之上的，不必是专门的网络硬件环境，例如电话上网，租用设备. 信息自己管理. 有比C/S更强的适应范围，一般只要有[操作系统](http://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/item/_blank)和[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)就行。

**2．对安全要求不同**

C/S 一般面向相对固定的用户群，对[信息安全](http://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E5%AE%89%E5%85%A8" \t "http://baike.baidu.com/item/_blank)的控制能力很强。 一般高度机密的信息系统采用C/S 结构适宜。可以通过B/S发布部分可公开信息。

B/S 建立在[广域网](http://baike.baidu.com/item/%E5%B9%BF%E5%9F%9F%E7%BD%91" \t "http://baike.baidu.com/item/_blank)之上， 对安全的控制能力相对弱， 可能面向不可知的用户。

**3．对程序架构不同**

C/S 程序可以更加注重流程， 可以对权限多层次校验， 对系统运行速度可以较少考虑。

B/S 对安全以及访问速度的多重的考虑，建立在需要更加优化的基础之上. 比C/S有更高的要求 B/S结构的程序架构是发展的趋势，从MS的.Net系列的BizTalk 2000 Exchange 2000等，全面支持网络的[构件](http://baike.baidu.com/item/%E6%9E%84%E4%BB%B6" \t "http://baike.baidu.com/item/_blank)搭建的系统。SUN 和IBM推JavaBean [构件技术](http://baike.baidu.com/item/%E6%9E%84%E4%BB%B6%E6%8A%80%E6%9C%AF" \t "http://baike.baidu.com/item/_blank)等，使 B/S更加成熟.。

**4．[软件重用](http://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E9%87%8D%E7%94%A8" \t "http://baike.baidu.com/item/_blank)不同**

C/S 程序可以不可避免的整体性考虑， [构件](http://baike.baidu.com/item/%E6%9E%84%E4%BB%B6" \t "http://baike.baidu.com/item/_blank)的重用性不如在B/S要求下的构件的重用性好。

B/S 的多重结构，要求构件相对独立的功能， 能够相对较好的重用，就如买来的餐桌可以再利用，而不是做在墙上的石头桌子。

**5．[系统维护](http://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E7%BB%B4%E6%8A%A4" \t "http://baike.baidu.com/item/_blank)不同**

C/S 程序由于整体性，必须整体考察，处理出现的问题以及系统升级、升级难、 可能是再做一个全新的系统。

B/S 构件组成，方便构件个别的更换，实现系统的无缝升级. [系统维护](http://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E7%BB%B4%E6%8A%A4" \t "http://baike.baidu.com/item/_blank)开销减到最小.用户从网上自己下载安装就可以实现升级。

**6．处理问题不同**

C/S 程序可以处理用户面固定，并且在相同区域，安全要求高，需求与[操作系统](http://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/item/_blank)相关. 应该都是相同的系统。

B/S 建立在[广域网](http://baike.baidu.com/item/%E5%B9%BF%E5%9F%9F%E7%BD%91" \t "http://baike.baidu.com/item/_blank)上， 面向不同的用户群，分散地域， 这是C/S无法作到的。与[操作系统](http://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/item/_blank)平台关系最小。

**7．[用户接口](http://baike.baidu.com/item/%E7%94%A8%E6%88%B7%E6%8E%A5%E5%8F%A3" \t "http://baike.baidu.com/item/_blank)不同 7**

C/S 多是建立的Window平台上，表现方法有限，对[程序员](http://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E5%91%98" \t "http://baike.baidu.com/item/_blank)普遍要求较高。

B/S 建立在[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8" \t "http://baike.baidu.com/item/_blank)上， 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低，减低开发成本。

**8．信息流不同**

C/S 程序一般是典型的[中央集权](http://baike.baidu.com/item/%E4%B8%AD%E5%A4%AE%E9%9B%86%E6%9D%83" \t "http://baike.baidu.com/item/_blank)的机械式处理，交互性相对低。

B/S 信息流向可变化，B-B B-C B-G等信息、流向的变化，更像交易中心。

### 三. TCP和UDP协议

* 1. TCP —— 面向连接的TCP协议

TCP（Transmission Control Protocol，[传输控制协议](http://baike.baidu.com/view/544903.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)）是基于连接的协议，也就是说，在正式收发数据前，必须和对方建立可靠的连接。一个TCP连接必须要经过三次“对话”才能建立起来，其中的过程非常复杂，我们这里只做简单、形象的介绍，你只要做到能够理解这个过程即可。我们来看看这三次对话的简单过程：[主机](http://baike.baidu.com/view/23880.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)A向主机B发出连接请求数据包：“我想给你发数据，可以吗？”，这是第一次对话；主机B向主机A发送同意连接和要求同步（同步就是两台主机一个在发送，一个在接收，协调工作）的数据包：“可以，你什么时候发？”，这是第二次对话；主机A再发出一个数据包确认主机B的要求同步：“我现在就发，你接着吧！”，这是第三次对话。三次“对话”的目的是使数据包的发送和接收同步，经过三次“对话”之后，[主机](http://baike.baidu.com/view/23880.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)A才向主机B正式发送数据。

TCP协议能为应用程序提供可靠的通信连接，使一台计算机发出的[字节流](http://baike.baidu.com/view/1645587.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)无差错地发往网络上的其他计算机，对可靠性要求高的[数据通信系统](http://baike.baidu.com/view/660332.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)往往使用TCP协议传输数据。

TCP支持的应用协议主要有：Telnet、FTP、SMTP等；

* 1. UDP —— 面向非连接的UDP协议

“面向非连接”就是在正式通信前不必与对方先建立连接，不管对方状态就直接发送。与手机短信非常相似：你在发短信的时候，只需要输入对方手机号就OK了。

UDP适用于一次只传送少量数据、对可靠性要求不高的应用环境。比如，我们经常使用“ping”命令来测试两台[主机](http://baike.baidu.com/view/23880.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)之间TCP/IP通信是否正常，其实“ping”命令的原理就是向对方主机发送ICMP数据包，然后对方主机确认收到数据包，如果数据包是否到达的消息及时反馈回来，那么网络就是通的。例如，在默认状态下，一次“ping”操作发送4个数据包（如图所示）。大家可以看到，发送的数据包数量是4包，收到的也是4包（因为对方主机收到后会发回一个确认收到的数据包）。这充分说明了UDP协议是[面向非连接](http://baike.baidu.com/view/2077319.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)的协议，没有建立连接的过程。正因为UDP协议没有连接的过程，所以它的通信效率高；但也正因为如此，它的可靠性不如TCP协议高。QQ就使用UDP发消息，因此有时会出现收不到消息的情况。

UDP支持的[应用层](http://baike.baidu.com/view/239619.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)协议主要有：NFS（[网络文件系统](http://baike.baidu.com/view/544339.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)）、SNMP（[简单网络管理协议](http://baike.baidu.com/view/875800.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)）、DNS（主域名称系统）、TFTP（通用[文件传输协议](http://baike.baidu.com/view/103832.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)）等。

* 1. 数据包

对于网络管理的网络安全具有至关重要的意义。比如，[防火墙](http://baike.baidu.com/view/3067.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)的作用本质就是检测网络中的[数据包](http://baike.baidu.com/view/25880.htm" \t "http://baike.baidu.com/item/TCP%2FUDP%E5%8D%8F%E8%AE%AE/_blank)，判断其是否违反了预先设置的规则，如果违反就加以阻止。下图就是某杀毒软件个人版防火墙软件设置规则的界面。细心的读者会发现，图中的“协议”栏中有“TCP”、“UDP”等名词。

### http协议

HTTP（HyperText Transport Protocol）是超文本传输协议的缩写，它用于传送WWW方式的数据，关于HTTP协议的详细内容请参考RFC2616。HTTP协议采用了请求/响应模型。客户端向服务器发送一个请求，请求头包含请求的方法、URL、协议版本、以及包含请求修饰符、客户信息和内容的类似于MIME的消息结构。服务器以一个状态行作为响应，响应的内容包括消息协议的版本，成功或者错误编码加上包含服务器信息、实体元信息以及可能的实体内容。

通常HTTP消息包括客户机向服务器的请求消息和服务器向客户机的响应消息。这两种类型的消息由一个起始行，一个或者多个头域，一个指示头域结束的空行和可选的消息体组成。HTTP的头域包括通用头，请求头，响应头和实体头四个部分。每个头域由一个域名，冒号（:）和域值三部分组成。域名是大小写无关的，域值前可以添加任何数量的空格符，头域可以被扩展为多行，在每行开始处，使用至少一个空格或制表符。

1. 工作原理

一次HTTP操作称为一个事务，其工作过程可分为四步：

1.首先客户机与服务器需要建立连接。只要单击某个超级链接，HTTP的工作就开始了。

2.建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议[版本号](http://baike.baidu.com/item/%E7%89%88%E6%9C%AC%E5%8F%B7" \t "http://baike.baidu.com/item/_blank)，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。

3.服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

4.客户端接收服务器所返回的信息通过浏览器显示在用户的显示屏上，然后客户机与服务器断开连接。

如果在以上过程中的某一步出现错误，那么产生错误的信息将返回到客户端，由显示屏输出。对于用户来说，这些过程是由HTTP自己完成的，用户只要用鼠标点击，等待信息显示就可以了。

许多HTTP通讯是由一个用户代理初始化的并且包括一个申请在源服务器上资源的请求。最简单的情况可能是在用户代理和服务器之间通过一个单独的连接来完成。在Internet上，HTTP通讯通常发生在TCP/IP连接之上。缺省端口是TCP 80，但其它的端口也是可用的。但这并不预示着HTTP协议在Internet或其它网络的其它协议之上才能完成。HTTP只预示着一个可靠的传输。

这个过程就好像我们打电话订货一样，我们可以打电话给商家，告诉他我们需要什么规格的商品，然后商家再告诉我们什么商品有货，什么商品缺货。这些，我们是通过电话线用电话联系（HTTP是通过TCP/IP），当然我们也可以通过传真，只要商家那边也有传真。

1. 请求报文格式

请求行 － 通用信息头 － 请求头 － 实体头 － 报文主体

典型的请求消息：

Host: download.\*\*\*\*\*\*\*.de

Accept: \*/\*

Pragma: no-cache

Cache-Control: no-cache

User-Agent: Mozilla/4.04[en](Win95;I;Nav)

Range: bytes=554554-

1. 响应报文格式

状态行 － 通用信息头 － 响应头 － 实体头 － 报文主体

典型的响应消息：

HTTP/1.0 200 OK

Date:Mon,31Dec200104:25:57GMT

Server:Apache/1.3.14(Unix)

Content-type:text/html

Last-modified:Tue,17Apr200106:46:28GMT

Etag:"a030f020ac7c01:1e9f"

Content-length:39725426

Content-range:bytes55\*\*\*\*\*\*/40279980

1. 报头
2. 通用头

Cache-Control、Connection、Date、Pragma、Transfer-Encoding、Upgrade、Via

1. 请求头

Accept、Accept-Charset、Accept-Encoding、Accept-Language、Authorization、From、Host、If-Modified-Since、If-Match、If-None-Match、If-Range、If-Range、If-Unmodified-Since、Max-Forwards、Proxy-Authorization、Range、Referer、User-Agent

1. 响应头

Age、Location、Proxy-Authenticate、Public、Retry-After、Server、Vary、Warning、WWW-Authenticate

1. 实体头

Allow、Content-Base、Content-Encoding、Content-Language、Content-Length、Content-Location、Content-MD5、Content-Range、Content-Type、Etag、Expires、Last-Modified、extension-header。extension-header

1. 状态码

1xx:信息响应类，表示接收到请求并且继续处理

2xx:处理成功响应类，表示动作被成功接收、理解和接受

3xx:[重定向](http://baike.baidu.com/item/%E9%87%8D%E5%AE%9A%E5%90%91" \t "http://baike.baidu.com/item/_blank)响应类，为了完成指定的动作，必须接受进一步处理

4xx:客户端错误，客户请求包含语法错误或者是不能正确执行

5xx:服务端错误，服务器不能正确执行一个正确的请求

### 输入url后的加载过程

**第一步： 查找域名对应IP地址**

这一步包括 DNS 具体的查找过程，包括：浏览器缓存->系统缓存->路由器缓存...  
(1) 浏览器搜索自己的 DNS 缓存（维护一张域名与 IP 地址的对应表）；  
(2) 搜索操作系统中的 DNS 缓存（维护一张域名与 IP 地址的对应表）；  
(3) 搜索操作系统的 hosts 文件（ Windows 环境下，维护一张域名与 IP 地址的对应表）；  
(4) 操作系统将域名发送至 LDNS（本地区域名服务器），LDNS 查询 自己的 DNS 缓存（一般查找成功率在 80% 左右），查找成功则返回结果，失败则发起一个迭代 DNS 解析请求：

① LDNS 向 Root Name Server （根域名服务器，如 com、net、org等的解析的顶级域名服务器的地址）发起请求，此处，Root Name Server 返回 com 域的顶级域名服务器的地址；

② LDNS 向 com 域的顶级域名服务器发起请求，返回 baidu.com 域名服务器地址；

③ LDNS 向 baidu.com 域名服务器发起请求，得到 www.baidu.com 的 IP 地址；

(5) LDNS 将得到的 IP 地址返回给操作系统，同时自己也将 IP 地址缓存起来；

(6) 操作系统将 IP 地址返回给浏览器，同时自己也将 IP 地址缓存起来；

**第二步： 建立连接(TCP的三次握手)**

(1) 主机向服务器发送一个建立连接的请求；

(2) 服务器接到请求后发送同意连接的信号；

(3) 主机接到同意连接的信号后，再次向服务器发送了确认信号 ;

注意：这里的三次握手中主机两次向服务器发送确认，第二次是为了防止已失效的连接请求报文段传至服务器导致错误。

**第三步：构建网页**

(1) 浏览器根据 URL 内容生成 HTTP 请求，请求中包含请求文件的位置、请求文件的方式等等；

(2) 服务器接到请求后，会根据 HTTP 请求中的内容来决定如何获取相应的 HTML 文件；

(3) 服务器将得到的 HTML 文件发送给浏览器；

(4) 在浏览器还没有完全接收 HTML 文件时便开始渲染、显示网页；

(5) 在执行 HTML 中代码时，根据需要，浏览器会继续请求图片、音频、视频、CSS、JS等文件，过程同请求 HTML ；

浏览器渲染展示网页过程

HTML代码转化为DOM(DOM Tree)

CSS代码转化成CSSOM（CSS Object Model）

结合DOM和CSSOM，生成一棵渲染树（包含每个节点的视觉信息）(Render Tree)

生成布局（layout），将所有渲染树的所有节点进行平面合成

将布局绘制（paint）在屏幕上

**第四步：断开连接(TCP的四次挥手)**

(1) 主机向服务器发送一个断开连接的请求；

(2) 服务器接到请求后发送确认收到请求的信号；(此时服务器可能还有数据要发送至主机)

(3) 服务器向主机发送断开通知；(此时服务器确认没有要向主机发送的数据)

(4) 主机接到断开通知后断开连接并反馈一个确认信号，服务器收到确认信号后断开连接；

注意：这里的四次挥手中服务器两次向主机发送消息，第一次是回复主机已收到断开的请求，第二次是向主机确认是否断开，确保数据传输完毕。

### 浏览器内部工作原理

1. 浏览器的主要组件

1. 用户界面 － 包括地址栏、后退/前进按钮、书签目录等，也就是你所看到的除了用来显示你所请求页面的主窗口之外的其他部分。

　　2. 浏览器引擎 － 用来查询及操作渲染引擎的接口。

　　3. 渲染引擎 － 用来显示请求的内容，例如，如果请求内容为html，它负责解析html及css，并将解析后的结果显示出来。

　　4. 网络 － 用来完成网络调用，例如http请求，它具有平台无关的接口，可以在不同平台上工作。

　　5. UI后端 － 用来绘制类似组合选择框及对话框等基本组件，具有不特定于某个平台的通用接口，底层使用操作系统的用户接口。

　　6. JS解释器 － 用来解释执行JS代码。

7. 数据存储 － 属于持久层，浏览器需要在硬盘中保存类似cookie的各种数据，HTML5定义了web database技术，这是一种轻量级完整的客户端存储技术。

1. 渲染引擎及工作流

渲染引擎的职责就是渲染，即在浏览器窗口中显示所请求的内容。默认情况下，渲染引擎可以显示html、xml文档及图片，它也可以借助插件（一种浏览器扩展）显示其他类型数据，例如使用PDF阅读器插件，可以显示PDF格式，将由专门一章讲解插件及扩展，这里只讨论渲染引擎最主要的用途——显示应用了CSS之后的html及图片。

渲染主流程

1：渲染引擎首先通过网络获得所请求文档的内容，通常以8K分块的方式完成

2：解析html以构建dom树 -> 构建render树 -> 布局render树 -> 绘制render树

渲染引擎开始解析html，并将标签转化为内容树中的dom节点。接着，它解析外部CSS文件及style标签中的样式信息。这些样式信息以及html中的可见性指令将被用来构建另一棵树——render树。

Render树由一些包含有颜色和大小等属性的矩形组成，它们将被按照正确的顺序显示到屏幕上。

Render树构建好了之后，将会执行布局过程，它将确定每个节点在屏幕上的确切坐标。再下一步就是绘制，即遍历render树，并使用UI后端层绘制每个节点。

值得注意的是，这个过程是逐步完成的，为了更好的用户体验，渲染引擎将会尽可能早的将内容呈现到屏幕上，并不会等到所有的html都解析完成之后再去构建和布局render树。它是解析完一部分内容就显示一部分内容，同时，可能还在通过网络下载其余内容。

### 重排与重绘

**重绘：**一个元素外观的改变所触发的浏览器行为，例如改变visibility、outline、背景色等属性。浏览器会根据元素的新属性重新绘制，使元素呈现新的外观。

**重排：**是更明显的一种改变，可以理解为渲染树需要重新计算。

### 页面性能优化

**缓存利用：**缓存Ajax，使用CDN，使用外部js和css文件以便缓存，添加Expires头，服务端配置Etag，减少DNS查找等

**请求数量：**合并样式和脚本，使用css图片精灵，初始首屏之外的图片资源按需加载，静态资源延迟加载。

**请求带宽：**压缩文件，开启GZIP，

**代码层面的优化:**

1. 避免使用高级选择器，通配选择器。
2. 少用全局变量
3. 用innerHTML代替DOM操作，减少DOM操作次数，优化javascript性能
4. 用setTimeout来避免页面失去响应
5. 缓存DOM节点查找的结果
6. 避免使用css表达式
7. 避免全局查询
8. 避免使用with(with会创建自己的作用域，会增加作用域链长度)
9. 多个变量声明合并
10. 避免图片和iFrame等的空Src。空Src会重新加载当前页面，影响速度和效率
11. 尽量避免写在HTML标签中写Style属性

### 移动端性能优化

1. 尽量使用css3动画，可以用transform: translateZ(0)来开启硬件加速。
2. 适当使用touch事件代替click事件。
3. 避免使用css3渐变阴影效果。
4. 不滥用Float。Float在渲染时计算量比较大，尽量减少使用
5. 不滥用Web字体。Web字体需要下载，解析，重绘当前页面，尽量减少使用。
6. 合理使用requestAnimationFrame动画代替setTimeout
7. CSS中的属性（CSS3 transitions、CSS3 3D transforms、Opacity、Canvas、WebGL、Video）会触发GPU渲染，请合理使用。过渡使用会引发手机耗电增加
8. PC端的在移动端同样适用

### 为什么利用多个域名来存储网站资源会更有效？

1. 静态内容和动态内容分服务器存放，使用不同的服务器处理请求。处理动态内容的只处理动态内容，不处理别的，提高效率，这样使得CDN（内容分发网络）缓存更方便突破浏览器并发限制 (你随便挑一个 G家的 url: https://lh4.googleusercontent.com//IMG\_20120603\_163233.jpg, 把前面的 lh4换成 lh3,lh6 啥的，都照样能够访问，像地图之类的需要大量并发下载图片的站点，这个非常重要。)
2. 跨域不会传cookie,节省宽带；举例说一下：   
   twitter 的主站 [http://twitter.com](http://twitter.com/" \t "http://blog.csdn.net/xujie_0311/article/details/_blank) ，用户的每次访问，都会带上自己的cookie ，挺大的。假如twitter 的图片放在主站域名下，那么用户每次访问图片时，request header 里就会带有自己的cookie ，header 里的cookie 还不能压缩，而图片是不需要知道用户的cookie 的，所以这部分带宽就白白浪费了。
3. 写主站程序时，set-cookie 也不要set 到图片的域名上。   
   在小流量的网站，这个cookie 其实节省不了多少带宽，当流量如facebook twitter 时，节省下来就很可观了。

关于多域名，也不是越多越好，虽然服务器端可以做泛解释，浏览器做dns解释也是耗时间的，而且太多域名，如果要走 https的话，还有要多买证书和部署的问题

### 网页重构

概述：

在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。也就是说是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

对于传统的网站来说重构通常是：

1. 表格(table)布局改为DIV+CSS
2. 使网站前端兼容于现代浏览器(针对于不合规范的CSS、如对IE6有效的)
3. 针对于SEO进行优化

深层次的网站重构应该考虑的方面：

1. 减少代码间的耦合
2. 让代码保持弹性
3. 严格按规范编写代码
4. 设计可扩展的API
5. 代替旧有的框架、语言
6. 增强用户体验
7. 通常来说对于速度的优化也包含在重构中
8. 压缩JS、CSS、image等前端资源(通常是由服务器来解决)
9. 程序的性能优化(如数据读写)
10. 采用CDN来加速资源加载
11. 对于JS DOM的优化
12. HTTP服务器的文件缓存

### SEO

1. 了解搜索引擎如何抓取网页和如何索引网页

你需要知道一些搜索引擎的基本工作原理，各个搜索引擎之间的区别，搜索机器人（SE robot 或叫 web crawler）如何进行工作，搜索引擎如何对搜索结果进行排序等等。

1. Meta标签优化

主要包括主题（Title)，网站描述(Description)，和关键词（Keywords）。还有一些其它的隐藏文字比如Author（作者），Category（目录），Language（编码语种）等。

1. 如何选取关键词并在网页中放置关键词

搜索就得用关键词。关键词分析和选择是SEO最重要的工作之一。首先要给网站确定主关键词（一般在5个上下），然后针对这些关键词进行优化，包括关键词密度（Density），相关度（Relavancy），突出性（Prominency）等等。

1. 了解主要的搜索引擎

　　虽然搜索引擎有很多，但是对网站流量起决定作用的就那么几个。比如英文的主要有Google，Yahoo，Bing等；中文的有百度，搜狗，有道等。不同的搜索引擎对页面的抓取和索引、排序的规则都不一样。还要了解各搜索门户和搜索引擎之间的关系，比如AOL网页搜索用的是Google的搜索技术，MSN用的是Bing的技术。

1. 主要的互联网目录

Open Directory自身不是搜索引擎，而是一个大型的网站目录，他和搜索引擎的主要区别是网站内容的收集方式不同。目录是人工编辑的，主要收录网站主页；搜索引擎是自动收集的，除了主页外还抓取大量的内容页面。

1. 按点击付费的搜索引擎

搜索引擎也需要生存，随着互联网商务的越来越成熟，收费的搜索引擎也开始大行其道。最典型的有Overture和百度，当然也包括Google的广告项目Google Adwords。越来越多的人通过搜索引擎的点击广告来定位商业网站，这里面也大有优化和排名的学问，你得学会用最少的广告投入获得最多的点击。

1. 搜索引擎登录

网站做完了以后，别躺在那里等着客人从天而降。要让别人找到你，最简单的办法就是将网站提交（submit）到搜索引擎。如果你的是商业网站，主要的搜索引擎和目录都会要求你付费来获得收录（比如Yahoo要299美元），但是好消息是（至少到目前为止）最大的搜索引擎Google目前还是免费，而且它主宰着60％以上的搜索市场。

1. 链接交换和链接广泛度（Link Popularity）

网页内容都是以超文本（Hypertext）的方式来互相链接的，网站之间也是如此。除了搜索引擎以外，人们也每天通过不同网站之间的链接来Surfing（“冲浪”）。其它网站到你的网站的链接越多，你也就会获得更多的访问量。更重要的是，你的网站的外部链接数越多，会被搜索引擎认为它的重要性越大，从而给你更高的排名。

1. 合理的标签使用

### W3C标准

W3C也称万维网联盟（World Wide Web Consortium）主要负责Web技术领域的标准起草和发布

网页主要由三部分组成：结构（Structure）、表现（Presentation）和行为（Behavior）；

对应的标准也分三方面：

结构层：[XHTML](http://baike.baidu.com/item/XHTML" \t "http://baike.baidu.com/_blank)、[XML](http://baike.baidu.com/item/XML" \t "http://baike.baidu.com/_blank)

表现层：CSS

行为层：DOM（文档对象模型）、ECMA[Script](http://baike.baidu.com/item/Script" \t "http://baike.baidu.com/_blank)等。

### 浏览器内核与兼容性

1. .内核类别
2. IE： trident内核
3. Firefox： gecko内核
4. Safari： webkit内核
5. Opera： 以前是presto内核，Opera现已改用Google Chrome的Blink内核
6. Chrome： Blink(基于webkit，[Google与Opera Software共同开发](http://baike.baidu.com/view/1369399.htm" \l "1_5" \t "http://www.mamicode.com/_blank))
7. 内核的主要功能

主要分成两部分：

1. **渲染引擎：**

负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

1. **JS引擎：**

解析和执行javascript来实现网页的动态效果。

注：最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

### 什么是微格式？在前端构建中应该考虑微格式吗？

微格式（Microformats）是一种让机器可读的语义化XHTML词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。（应用范例：豆瓣，有兴趣自行google）

### WEB应用从服务器主动推送Data到客户端有那些方式？

一：需要在浏览器端安装插件，基于套接口传送信息，或是使用 RMI、CORBA 进行远程调用；

二：无须浏览器安装任何插件、基于 HTTP 长连接。

1. 基于 AJAX 的长轮询（long-polling）方式，服务器Hold一段时间后再返回信息；
2. Commet：基于HTTP长连接的服务器推送技术
3. HTTP Streaming，通过iframe和<script>标签完成数据的传输；
4. HTML5新引入的WebSocket，可以实现服务器主动发送数据至网页端，它和HTTP一样，是一个基于HTTP的应用层协议，跑的是TCP，所以本质上还是个长连接，双向通信，意味着服务器端和客户端可以同时发送并响应请求，而不再像HTTP的请求和响应
5. SSE（Server-Send Event）：服务器推送数据新方式

### 常见web安全及维护管理

1. sql注入原理及防范

**原理：**

sql注入原理就是通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。

**防范：**

1. 永远不要信任用户的输入，要对用户的输入进行校验，可以通过正则表达式，或限制长度，对单引号和双"-"进行转换等。
2. 永远不要使用动态拼装SQL，可以使用参数化的SQL或者直接使用存储过程进行数据查询存取。
3. 永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。
4. 不要把机密信息明文存放，请加密或者hash掉密码和敏感的信息。
5. XSS原理及防范

**原理**

Xss(cross-site scripting)攻击指的是攻击者往Web页面里插入恶意 html标签或者[JavaScript](http://lib.csdn.net/base/javascript" \o "JavaScript知识库" \t "http://blog.csdn.net/kebi007/article/details/_blank)代码。比如：攻击者在论坛中放一个看似安全的链接，骗取用户点击后，窃取cookie中的用户私密信息；或者攻击者在论坛中加一个恶意表单，当用户提交表单的时候，却把信息传送到攻击者的服务器中，而不是用户原本以为的信任站点。

**防范**

1. 首先代码里对用户输入的地方和变量都需要仔细检查长度和对”<”,”>”,”;”,”’”等字符做过滤；其次任何内容写到页面之前都必须加以encode，避免不小心把html tag 弄出来。这一个层面做好，至少可以堵住超过一半的XSS 攻击。
2. 避免直接在cookie 中泄露用户隐私，例如email、密码等等。
3. 通过使cookie 和系统ip 绑定来降低cookie 泄露后的危险。这样攻击者得到的cookie 没有实际价值，不可能拿来重放。
4. 如果网站不需要再浏览器端对cookie 进行操作，可以在Set-Cookie 末尾加上HttpOnly 来防止javascript 代码直接获取cookie 。
5. 尽量采用POST 而非GET 提交表单
6. CSRF原理及防御

**原理**

XSS是获取信息，不需要提前知道其他用户页面的代码和数据包。CSRF是代替用户完成指定的动作，需要知道其他用户页面的代码和数据包。

要完成一次CSRF攻击，受害者必须依次完成两个步骤：

1. 登录受信任网站A，并在本地生成Cookie。
2. 在不登出A的情况下，访问危险网站B。

**防范**

服务端的CSRF方式方法很多样，但总的思想都是一致的，就是在客户端页面增加伪随机数。

### 什么样的前端代码是好的?

1. 高复用
2. 低耦合
3. 好维护
4. 好扩展