# HTTP协议

把专栏的学习由“被动地听”，转变为“主动地学”，实现“学以致用

URI：即统一资源标识符，作为互联网上资源的唯一身份；

HTML：即超文本标记语言，描述超文本文档；

HTTP：即超文本传输协议，用来传输超文本。

<https://www.ruanyifeng.com/blog/2016/08/http.html>

# 历史[推动HTTP发展的原动力：用户需求推动技术发展]

* HTTP 协议始于三十年前蒂姆·伯纳斯 - 李的一篇论文；
* HTTP/0.9 是个简单的文本协议，只能获取文本资源；
* HTTP/1.0 确立了大部分现在使用的技术，但它不是正式标准；
* HTTP/1.1 是目前互联网上使用最广泛的协议，功能也非常完善；
* HTTP/2 基于 Google 的 SPDY 协议，注重性能改善，但还未普及；HTTP/3 基于 Google 的 QUIC 协议，是将来的发展方向。

HTTP/0.9

* 只允许用“GET”
* 服务器只能回应HTML格式的字符串，不能回应别的格式。
* 服务器发送完毕，就关闭TCP连接。

HTTP/1.0

* 增加了 HEAD、POST 等新方法；
* 增加了响应状态码，标记可能的错误原因；
* 引入了协议版本号概念；
* 引入了 HTTP Header（头部）的概念，
* 让 HTTP 处理请求和响应更加灵活；传输的数据不再仅限于文本。

HTTP /1.1

* 增加了 PUT、DELETE 等新的方法；增加了缓存管理和控制；
* 明确了连接管理，允许持久连接；
* 允许响应数据分块（chunked），利于传输大文件；
* 强制要求 Host 头，让互联网主机托管成为可能。

HTTP/2.0

* 二进制协议，不再是纯文本；
* 可发起多个请求，废弃了 1.1 里的管道；
* 使用专用算法压缩头部，减少数据传输量；
* 允许服务器主动向客户端推送数据；
* 增强了安全性，“事实上”要求加密通信。

HTTP/3

Google 又发明了一个新的协议，叫做 QUIC

# HTTP是什么

**定义：HTTP 是一个在计算机世界里专门在两点之间传输文字、图片、音频、视频等超文本数据的约定和规范”。**

**超文本**：超文本”，就是“超越了普通文本的文本”，它是文字、图片、音频和视频等的混合体，最关键的是含有“超链接”，能够从一个“超文本”跳跃到另一个“超文本”，形成复杂的非线性、网状的结构关系。HTML 了，它本身只是纯文字文件，但内部用很多标签定义了对图片、音频、视频等的链接，再经过浏览器的解释，呈现在我们面前的就是一个含有多种视听信息的页面。

传输 ：

* HTTP 协议是一个“双向协议”。就是把一堆东西从 A 点搬到 B 点，或者从 B 点搬到 A 点，即“A<===>B
* 数据虽然是在 A 和 B 之间传输，但并没有限制只有 A 和 B 这两个角色，允许中间有“中转”或者“接力”。传输方式就从“A<===>B”，变成了“A<=>X<=>Y<=>Z<=>B

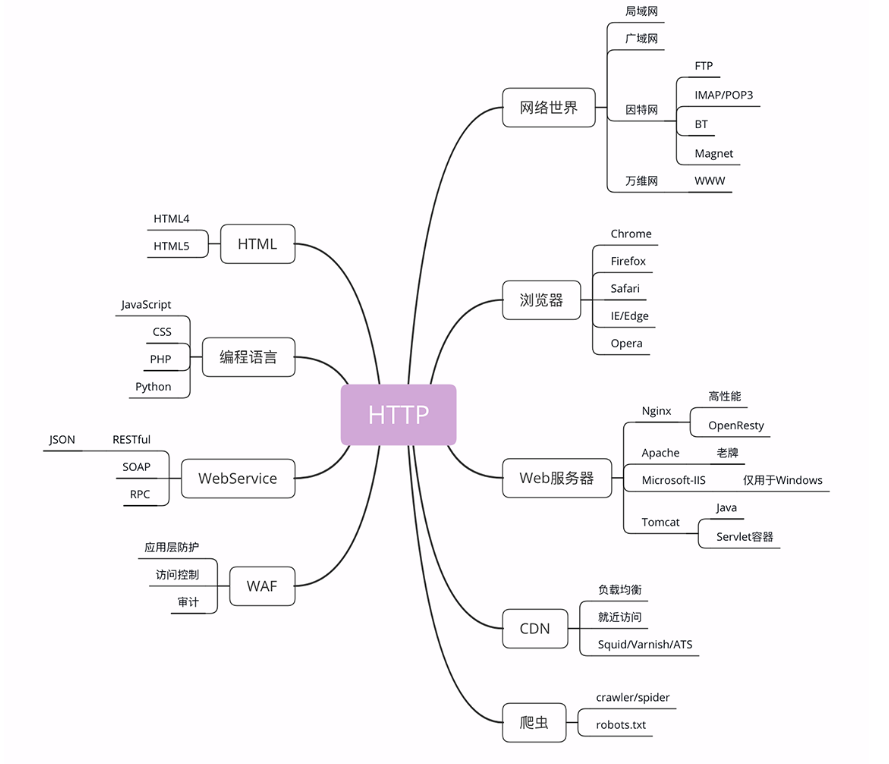
协议（多个参与者之间的约定）

* HTTP 是一个用在计算机世界里的协议，它确立了一种计算机之间交流通信的规范，以及相关的各种控制和错误处理方式。
* HTTP 专门用来在两点之间传输数据，不能用于广播、寻址或路由。
* HTTP 传输的是文字、图片、音频、视频等超文本数据。
* HTTP 是构建互联网的重要基础技术，它没有实体，依赖许多其他的技术来实现，但同时许多技术也都依赖于它。

【Question】HTTP不是什么？

* HTTP 不是互联网。
* HTTP 不是编程语言。
* HTTP 不是 HTML（HTML 是超文本的载体，是一种标记语言）
* HTTP 不是一个孤立的协议。
  + 在互联网世界里，HTTP 通常跑在 **TCP/IP** 协议栈之上，
  + 依靠 **IP** 协议实现寻址和路由、
  + TCP 协议实现可靠数据传输、
  + DNS 协议实现域名查找、
  + SSL/TLS 协议实现安全通信。
  + 此外，还有一些协议依赖于 HTTP，例如 WebSocket、HTTPDNS 等。
  + 这些协议相互交织，构成了一个协议网，而 HTTP 则处于中心地位。

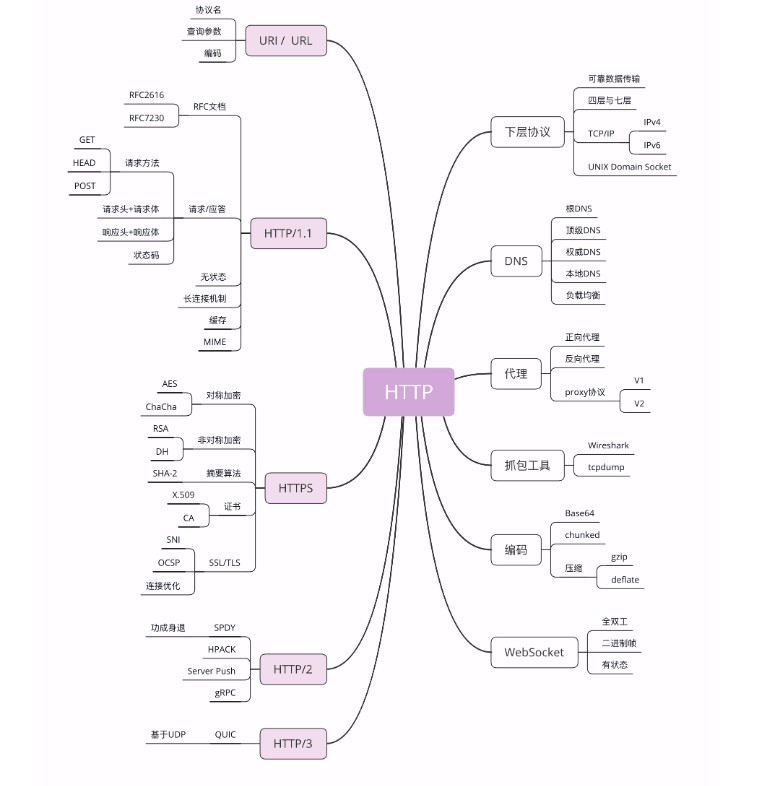
# HTTP相关概念



* 互联网上绝大部分资源都使用 HTTP 协议传输；
* 浏览器是 HTTP 协议里的请求方，即 User Agent；
* 服务器是 HTTP 协议里的应答方，常用的有 Apache 和 Nginx；
* CDN 位于浏览器和服务器之间，主要起到缓存加速的作用；

CDN，全称是“Content Delivery Network”，翻译过来就是“内容分发网络”。它应用了 HTTP 协议里的缓存和代理技术，代替源站响应客户端的请求。

* 爬虫是另一类 User Agent，是自动访问网络资源的程序。



* TCP/IP 是网络世界最常用的协议，HTTP 通常运行在 TCP/IP 提供的可靠传输基础上；
  + 四层模型：应用层、传输层、网际层、链接层
  + TCP/IP 协议实际上是一系列网络通信协议的统称
  + IP 协议是“Internet Protocol”的缩写，主要目的是解决寻址和路由问题，以及如何在两点间传送数据包。
* DNS 域名是 IP 地址的等价替代，需要用域名解析实现到 IP 地址的映射；URI 是用来标记互联网上资源的一个名字，由“协议名 + 主机名 + 路径”构成，俗称 URL；
  + 级别从左到右逐级升高
  + URI（Uniform Resource Identifier），中文名称是 统一资源标识符，使用它就能够唯一地标记互联网上资源
  + URL（Uniform Resource Locator）， 统一资源定位符，也就是我们俗称的“网址”，它实际上是 URI 的一个子集
  + <http://nginx.org/en/download.html>

你可以看到，URI 主要有三个基本的部分构成：

* + - 协议名：即访问该资源应当使用的协议，在这里是“http”；
    - 主机名：即互联网上主机的标记，可以是域名或 IP 地址，在这里是“nginx.org”；
    - 路径：即资源在主机上的位置，使用“/”分隔多级目录，在这里是“/en/download.html”。
* HTTPS 相当于“HTTP+SSL/TLS+TCP/IP”，为 HTTP 套了一个安全的外壳；
  + SSL/TLS它是一个负责加密通信的安全协议
  + HTTPS： “HTTP over SSL/TLS”，也就是运行在 SSL/TLS 协议上的 HTTP
  + SSL 的全称是“Secure Socket Layer”，由网景公司发明，当发展到 3.0 时被标准化，改名为 TLS，即“Transport Layer Security”，但由于历史的原因还是有很多人称之为 SSL/TLS，或者直接简称为 SSL。
* 代理是 HTTP 传输过程中的“中转站”，可以实现缓存加速、负载均衡等功能。
  + CDN，实际上就是一种代理，它代替源站服务器响应客户端的请求，通常扮演着透明代理和反向代理的角色。

# TCP/IP

它是 HTTP 协议的下层协议，负责具体的数据传输工作。并且还特别说了，TCP/IP 协议是一个“有层次的协议栈”

* TCP/IP 分为四层，核心是二层的 IP 和三层的 TCP，HTTP 在第四层；
* OSI 分为七层，基本对应 TCP/IP，TCP 在第四层，HTTP 在第七层；
* OSI 可以映射到 TCP/IP，但这期间一、五、六层消失了；
* 日常交流的时候我们通常使用 OSI 模型，用四层、七层等术语；
* HTTP 利用 TCP/IP 协议栈逐层打包再拆包，实现了数据传输，但下面的细节并不可见。

网络模型图

